



RESUMEN

El proyecto consta de cuatro tarjetas: una de amplificación, almacenamiento, visualización, sistema de alimentación. Todas ellas se acoplan mediante una tarjeta denominada “tarjeta madre” (tarjeta base). El geófono de tres componentes es un elemento externo al sistema y se lo utiliza para sensar los movimientos de la tierra en las direcciones: vertical, N-S y E-O.

La tarjeta amplificadora se encarga de tomar las señales provenientes del geófono, amplificándolas y controla la ganancia del amplificador operacional de instrumentación (AO INA114P) mediante un multiplexor (4050). Luego se pasa a través de un filtro paso bajo (LPF) cuya frecuencia de corte esta alrededor de los 500 Hz. Para finalmente acondicionar la señal a niveles entre 0 y 5V.

El PIC18F2553 es el encargado de la digitalización de las señales del geófono, cuyos canales se muestrean a razón de 200Hz cada uno. Además tiene implementado un mecanismo para detección de eventos sísmicos.

El propósito del almacenador (PIC18f452) es guardar los datos que envía el PIC2553 en la tarjeta SD. Los datos son guardados en formato ASCII utilizando las librerías de MikroC Pro para el manejo de los archivos en FAT16. Además de incluirse la hora y fecha en que sucedió el evento, mediante la lectura del RTC.

Por último se incluye un sistema de visualización de señales, cuya función es graficar el canal vertical del geófono.



INDICE

| | |
|--|-----------|
| RESUMEN..... | 1 |
| INTRODUCCION..... | 9 |
| CAPITULO 1: ANÁLISIS DE LOS EQUIPO EXISTENTES..... | 10 |
| INTRODUCCIÓN..... | 10 |
| 1.1 Descripción del funcionamiento de la red de sismógrafos | 11 |
| 1.1.1 Módulos existentes..... | 11 |
| 1.1.2 Fuentes de alimentación | 11 |
| 1.1.3 Funcionamiento del convertor de tensión de +12 a – 12..... | 11 |
| 1.1.4 Controlador de carga..... | 12 |
| 1.1.5 Amplificador de ganancia variable..... | 12 |
| 1.1.6 Mezclador..... | 12 |
| 1.1.6.1 Funcionamiento | 12 |
| 1.1.6.2 Especificaciones..... | 12 |
| 1.1.7 Sistema de transmisión..... | 13 |
| 1.1.8 Estaciones sensoras | 13 |
| Sistema de alimentación | 13 |
| Sistema de toma de información: | 13 |
| Sistema de acoplamiento de señal:..... | 13 |
| Sistema de transmisión: | 13 |
| Estaciones Repetidoras:..... | 13 |
| 1.2 Geófonos..... | 14 |
| Introducción..... | 14 |
| 1.2.1 Principio de Funcionamiento..... | 14 |
| 1.2.2 Ecuación diferencial del geófono..... | 15 |
| 1.2.4 Geófonos de una componente: vertical y horizontal..... | 18 |
| 1.2.5 Geófonos de tres componentes..... | 21 |
| 1.3 Digitalización de señales | 22 |
| 1.3.1 Origen de las ondas sísmicas | 22 |
| 1.3.2 Frecuencia de muestreo para un ADC..... | 23 |



| | |
|---|-----------|
| 1.3.3 Resolución. | 25 |
| 1.3.4 Rango dinámico | 25 |
| CAPITULO 2: UTILITARIOS PARA EL DISEÑO | 28 |
| 2.1 Proteus | 28 |
| 2.1.1 Interfaz de usuario..... | 28 |
| ISIS..... | 28 |
| 2.1.2 Captura de esquemas | 32 |
| 2.2 Mikroc | 33 |
| 2.2.1 Interfaz de usuario..... | 33 |
| 2.2.2 Manejo de Interrupciones en Mikroc | 38 |
| 2.2.3 Librerías del Mikroc Pro. | 44 |
| 2.2.4 Interacción con el ISIS de Proteus. | 45 |
| 2.3 Eagle | 47 |
| 2.3.1 Panel de control | 47 |
| 2.3.2 Esquemas | 51 |
| 2.3.3 PCB..... | 55 |
| CAPITULO 3: DISEÑO DE TARJETAS | 59 |
| 3.1 Sistema de almacenamiento de datos y temporización | 59 |
| 3.1.1 SD/MMC con SPI, uso de librerías de Mikroc Pro 3.2..... | 59 |
| 3.1.1.1 Protocolo de comunicación SPI..... | 59 |
| 3.1.1.2 Comunicación entre periféricos por SPI | 60 |
| 3.1.1.3 Tarjeta SD | 63 |
| 3.1.1.4 Funcionamiento de la memoria en modo spi..... | 66 |
| <i>Registros del PIC18F452 en modo SPI.....</i> | <i>66</i> |
| Maestro (PIC18F452)..... | 68 |
| Esclavo (memoria SD) | 68 |
| Transmisión..... | 68 |
| 3.1.1.4.1 Comunicación SPI con la tarjeta de memoria..... | 69 |
| 3.1.1.4.2 Uso de las librerías del módulo SPI del PIC18F452..... | 72 |
| Tabla 3.7. SPI1_Init_Advanced:..... | 72 |
| Tabla 3.8. SPI1_Read:..... | 73 |



| | |
|--|------------|
| Tabla 3.9. SPI1_Write: | 73 |
| 3.1.1.5 Uso de librerías de mikroc pro 3.2 para la sdcard | 73 |
| 3.1.1.5.1 Tabla de asignación de archivos(FAT16)..... | 73 |
| 3.1.1.5.2 Funciones para trabajar con FAT16..... | 75 |
| 3.1.1. 6 Conexión de la memoria SD con el pic18f452..... | 81 |
| 3.1.2Reloj de tiempo real, comunicación con GPS con el protocolo NMEA... 82 | |
| 3.1.2 .1 RTC (Reloj en Tiempo Real) | 82 |
| 3.1.2.1.1 Circuito integrado DS1307 | 82 |
| 3.1.2.2 Receptor GPS | 84 |
| 3.1.2.2.1 Medidas de posicionamiento global | 84 |
| 3.1.2.2.2 El GPS MN5010HS | 85 |
| 3.1.2.2.3 Recepción de datos..... | 86 |
| 3.2 Sistema de comunicaciones..... | 87 |
| 3.2.1Implementación de drivers para manejo de modem telefónico convencional | 87 |
| 3.2.1.1 Modem 73k224BL | 88 |
| 3.2.1.1.1 Descripción de los pines | 90 |
| 3.2.1.1.2 Descripción de los registros internos del modem | 93 |
| 3.2.1.1.3 Consideraciones generales para diseño | 95 |
| 3.2.1.1.4 Transformador de aislamiento..... | 96 |
| 3.2.1.1.5 Circuito detección de llamada | 96 |
| 3.3Despliegue en interfaz grafico..... | 97 |
| 3.3.1 Pantalla táctil..... | 97 |
| 3.3.2 Principio de funcionamiento | 98 |
| 3.3.3 Determinación de coordenadas de la pantalla táctil..... | 99 |
| 3.3.4 Conexión al Pic18f452 | 100 |
| 3.3.5 Uso de librerías de MikroC Pro 3.2 para la pantalla táctil..... | 101 |
| 3.4Integración de las partes..... | 102 |
| 3.4.1 Objetivos. | 102 |
| 3.4.2 Funcionamiento general..... | 103 |
| 3.4.2.1Sistema de muestreo y amplificación | 104 |
| 3.4.2.2Sistema de almacenamiento y georeferenciacion | 105 |



| | |
|--|------------|
| 3.4.2.3 Sistema de visualización | 106 |
| 3.4.3 Diagramas de flujo. | 108 |
| 3.4.3.1 Sistema de muestreo y amplificación | 108 |
| 3.4.3.2 Sistema de almacenamiento y georeferenciación | 110 |
| 3.4.3.3 Sistema de visualización | 112 |
| CAPITULO 4: CARACTERISTICAS Y ALGORITMOS DE LOS MICROPROCESADORES | 114 |
| 4.1 Características generales de los microprocesadores | 114 |
| 4.1.1 Características pic 18f452 y 18f2553 | 114 |
| 4.1.2 Oscilador modo hs + pll..... | 115 |
| 4.1.3 Organización de la memoria | 118 |
| 4.1.4 Diagrama de bloques pic18f452..... | 120 |
| 4.1.5 Diagrama de bloques pic18f2553..... | 121 |
| 4.2 Algoritmos en los microprocesadores | 122 |
| 4.2.1 Algoritmo del sistema de almacenamiento y georeferenciación..... | 122 |
| 4.2.1.1 Configuración de frecuencia de trabajo del microprocesador..... | 122 |
| 4.2.1.2 Frecuencia de muestreo de las señales del Geófono..... | 123 |
| 4.2.1.3 Configuración de la velocidad del bus SPI | 124 |
| 4.2.1.4 Algoritmo | 126 |
| 4.2.2 Algoritmo del sistema de visualización..... | 149 |
| 4.2.2.1 Configuración de frecuencia de trabajo del microprocesador..... | 149 |
| 4.2.2.2 Configuración de la velocidad del bus SPI | 150 |
| 4.2.2.3 Algoritmo | 151 |
| 4.2.3 Algoritmo del sistema de muestreo y detección de eventos..... | 157 |
| 4.2.3.1 Configuración de frecuencia de trabajo del microprocesador..... | 157 |
| 4.2.3.2 Configuración de la velocidad del bus SPI | 158 |
| 4.2.3.3 Algoritmo | 158 |
| 4.2.4 Descripción del algoritmo del controlador de carga..... | 164 |
| CAPITULO 5: IMPLEMENTACIÓN..... | 167 |
| 5.1 Pruebas finales del equipo. | 167 |
| 5.1.1 Amplificación de la señal proveniente del geófono. | 167 |



| | |
|---|------------|
| 5.1.1.1 Señal del Geófono..... | 167 |
| 5.1.1.2 Señal de entrada al PIC2553..... | 168 |
| 5.1.2Lectura del GPS y funcionamiento RTC..... | 168 |
| 5.1.3Detección de un evento, transmisión de datos hacia el almacenador.. | 169 |
| 5.1.4Visualización en la pantalla GLCD | 171 |
| 5.1.5Transmisión por el modem telefónico..... | 172 |
| 5.2 Manual de funcionamiento del equipo construido. | 173 |
| 5.2.1 Controlador de Carga..... | 173 |
| 5.2.1.1 Descripción general..... | 173 |
| 5.2.1.2 Funcionamiento | 173 |
| 5.2.1.3 Especificaciones del controlador de carga | 174 |
| 5.2.1.4 Descripción de los pines de la tarjeta del controlador de carga.... | 174 |
| 5.2.1.5 Descripción de los elementos del controlador de carga | 174 |
| 5.2.2 Almacenador | 181 |
| 5.2.2.1 Descripción general..... | 181 |
| 5.2.2.2 Funcionamiento | 181 |
| 5.2.2.3 Especificaciones del almacenador | 181 |
| 5.2.2.4 Descripción de los pines de la tarjeta del almacenador..... | 182 |
| 5.2.3 Tarjeta base | 185 |
| 5.2.3.1 Funcionamiento | 185 |
| 5.2.3.2 Especificaciones..... | 185 |
| CONCLUSIONES Y RECOMENDACIONES | 187 |
| BIBLIOGRAFIA | 190 |



GLOSARIO

A

A/D Analógico-Digital

ADC Conversor analógico-digital, es un dispositivo electrónico generalmente en circuito integrado, convierte un voltaje analógico a digital.

ASCII (American Standard Code of Information Interchange - Estándar Americano para Intercambio de Información). Es un código largo que define caracteres alfanuméricos para compatibilizar procesadores de texto y programas de comunicaciones.

B

BUS Es un sistema digital que transfiere datos entre los componentes de una computadora o entre computadoras. Está formado por cables o pistas en un circuito impreso.

G

GEOFONOS Transductores de desplazamiento, velocidad o aceleración que convierten el movimiento del suelo en una señal eléctrica.

GPS El GPS (Global Positioning System: sistema de posicionamiento global) es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión.

I

I2C Bus de comunicaciones en serie.

M

MMC/SD Tarjeta de memoria o tarjeta de memoria flash es un dispositivo de almacenamiento que conserva la información que le ha sido almacenada de forma correcta aun con la pérdida de energía, es decir, es una memoria no volátil.

**N**

NMEA Es una especificación combinada eléctrica y de datos entre aparatos electrónicos marinos y, también, más generalmente, receptores GPS. Es un medio a través del cual los instrumentos marítimos y también la mayoría de los receptores GPS pueden comunicarse los unos con otros.

P

PIC Programmable Interrupt Controller (Controlador programable de interrupciones usado especialmente en los IBM PCs.)

PLL Phase Locked Loop, circuito de sincronización de fase

R

RTC Un reloj en tiempo real (RTC) es un reloj de computadora (más a menudo en forma de un circuito integrado) que realiza un seguimiento de la hora y fecha actual.

RS-232 Interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos), aunque existen otras en las que también se utiliza la interfaz RS-232.

RSA Red Sísmica del Austro

S

SPI Estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier electrónica digital que acepte un flujo de bits serie regulado por un reloj.

V

VCD Tensión de corriente continua.



INTRODUCCION

La Red Sísmica del Austro (RSA) cuenta con varias estaciones de monitoreo sísmico ubicadas en distintos puntos de la región. Algunos de sus equipos de registro continuo son de tecnología analógica, causando que el procesamiento de los datos sea ineficiente e inadecuado, debido a que en la actualidad de toda la información grabada muy poco son sismos y el resto información inútil. Entonces surge la necesidad de tecnologías modernas, como lo es la digital, que permitirá el almacenamiento selectivo de los datos, para su posterior tratamiento.

Es por esto que se ha visto la necesidad de construir un equipo que cumpla con las características de: detección de eventos, almacenamiento, y visualización de señales, garantizando el tratamiento adecuado de los sismos.

Una vez que las señales del geófono son digitalizadas y discriminadas mediante el proceso de detección de eventos. Estas se almacenan en la tarjeta SD/MMC utilizando el formato FAT16, lo que hace que tales eventos sean tratados desde cualesquier computador que soporta dicho formato.



CAPITULO 1: ANÁLISIS DE LOS EQUIPOS EXISTENTES

INTRODUCCIÓN.

La Red Sísmica del Austro cuenta con 8 estaciones de monitoreo sísmico para la región Austral, distribuidas en distintos puntos. Las señales sísmicas de cada estación de monitoreo son tomadas por sensores denominados geófonos, luego estas son amplificadas y moduladas para la transmisión de la información hacia la estación central de recopilación de información ubicado en el local de la RSA, en el segundo piso de la facultad de Ingeniería de la Universidad de Cuenca.

Los geófonos (denominados también sismómetros) actuales, son elementos transductores que transforman la velocidad del movimiento del suelo a señales eléctricas. Anteriormente estos dispositivos carecían de un sistema de tratamiento de señales, pero en la actualidad se incluyen modernos elementos digitales en el procesamiento, almacenamiento, y transmisión de datos. Con esto se consiguen mejorar; el rango dinámico, resolución, y más parámetros de los geófonos.

El movimiento del suelo se produce en todas las direcciones, es por esto que las estaciones sísmicas requieren de un geófono vertical y como mínimo de dos horizontales para el registro sísmico. Típicamente la ubicación de los geófonos horizontales es uno en la dirección N-S y el otro en la E-O. La diferencia entre un vertical u horizontal radica en la forma como se coloque la masa de libre oscilación respecto del soporte del instrumento, pero el principio de funcionamiento es el mismo para los dos.

1.1 DESCRIPCIÓN DEL FUNCIONAMIENTO DE LA RED DE SISMÓGRAFOS

1.1.1 Módulos existentes

Cada estación de monitoreo cuenta con un módulo de acondicionamiento de señales el mismo que está formado por tarjetas electrónicas:

- controlador de carga/protector de baterías,
- amplificador,
- modulador y mezclador.
- posee también un panel solar o una fuente de 13V para recargar las baterías.
- los radios transmisores y receptores así como las antenas.

1.1.2 Fuentes de alimentación

Las fuentes de poder para los distintos módulos están basados en los integrados LM7805, LM3.3, LM317L que regulan tensión a -12, 3.3, -12, 5 VDC.

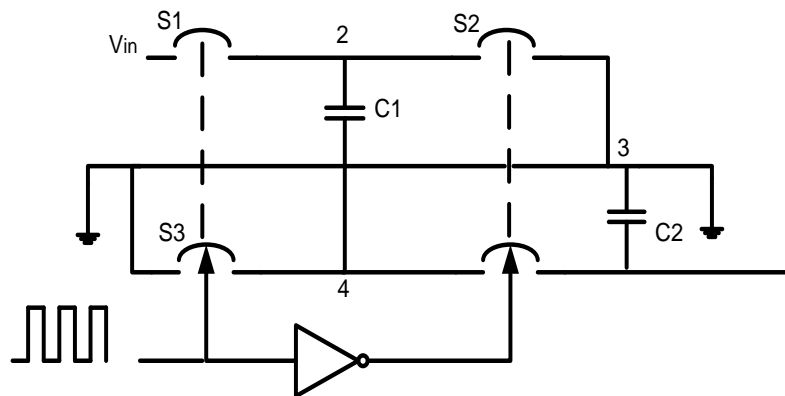


Fig. 1.1. Esquema de funcionamiento del conversor negativo

1.1.3 Funcionamiento del conversor de tensión de +12 a - 12.

En el primer ciclo del reloj, los switches S1 y S3 se cierran lo que permite que el condensador C1 se cargue. En la segunda mitad del ciclo, se abren los switches S1 y S3, se cierran S2 y S4 haciendo que el condensador cargado C1 entregue corriente a la salida (en este momento se produce la inversión de la tensión remanente en el condensador). Entonces este conversor obedece al principio de carga y descarga de un capacitor.



La fuente de alimentación para este módulo está basada en el chip 7662, el cual se utiliza para invertir la polaridad de la tensión. EL regulador LM33 proporciona una tensión de 3.3 y el LM7805 una de 5 VDC.

1.1.4 Controlador de carga

Este módulo permite el control de la recarga de la batería a través de un panel solar o su vez de una fuente de 13 VDC conectada a la red eléctrica de 110V, además realiza el control de la alimentación a los demás equipos. En caso de no existir una adecuada recarga de la batería. El controlador de carga no permite el funcionamiento de los demás elementos, sino hasta el instante en que se consiga un nivel óptimo en la batería.

1.1.5 Amplificador de ganancia variable

El circuito está basado en el amplificador de instrumentación INA114D, y se describe con mayor detalle en el capítulo cuatro.

1.1.6 Mezclador

1.1.6.1 Funcionamiento

Las señales de los canales de entrada que van hacia el modulo del amplificador donde son filtradas y amplificadas mediante amplificadores operacionales TL084. Cada una de las salidas del amplificador pasan a través de un dip-switch, el mismo que sirve para habilitarlas o deshabilitarlas. Luego todas estas señales se suman mediante el integrado TL081.

1.1.6.2 Especificaciones

Cuatro canales de entrada para señales moduladas de geófono y un canal extra para señales de información.

Un canal de salida

Tensión de alimentación 12V.

Tensiones interna de manipulación +/- 12V.

1.1.7 Sistema de transmisión.

El sistema de transmisión y recepción de la información que se registra en la Red sísmica del Austro está compuesto por varias etapas que se describe a continuación.

1.1.8 Estaciones sensoras

Sistema de alimentación:

A los módulos empleados para la recepción, amplificación y transmisión de la información, se les suministra energía eléctrica a través de un banco de baterías. Las mismas que pueden ser recargadas por un panel solar o a través de la alimentación de la red eléctrica de 120 VC-A, para su recarga se emplea una fuente regulada de ± 12 VDC.

Sistema de toma de información:

Los geófonos son los encargados de la toma de información sísmica.

Sistema de acoplamiento de señal:

La señal que llega desde el geófono es amplificada, acondicionan para luego ser transmitida.

Este sistema está formado de una fuente de alimentación, amplificador y un modulador.

Sistema de transmisión:

Formado por radios analógicos para el enlace.

Estaciones Repetidoras:

Sistema de Multiplexación:

El módulo de multiplexación tiene la finalidad de mezclar varios canales de entrada hacia un solo canal de salida, de forma que con un mismo enlace se puede usar varios canales de transmisión. En cada estación de monitoreo se tiene previsto como un máximo de cuatro canales de entrada a un módulo de multiplexación y tenga un solo canal de salida.

Los canales de entrada hacia el módulo de multiplexación son filtrados y amplificados mediante amplificadores operaciones (TL084).

1.2 GEÓFONOS.

Introducción.

En el registro y estudio de los movimientos de la tierra (sismología), se utilizan dispositivos transductores denominados sismómetros, los mismos que son elementos mecánicos, que transforman la velocidad del movimiento del suelo en señales eléctricas.

Existen sismómetros que se utilizan en la superficie de la tierra, y otros en el agua, denominados GEOFONOS e HIDROFONOS respectivamente. Durante este estudio se hará referencia únicamente a los geófonos.

El primer geófono o sistema de detección de sismos, se construyó en China más o menos por el año 132 d.C. este consistía de una vasija de bronce rodeada de ocho cabezas de dragones. Los mismos que en su boca tenían colocadas unas esferas, que estaban acopladas mediante varillas de bronce a un sistema pendular en el interior de la vasija. Además en alineación con los dragones estaban ocho ranas.

Cuando se presentaba un sismo, el sistema pendular oscilaba en la dirección del evento, haciendo que una de las esferas caiga en la boca de la rana. De este modo se sabía que había existido una vibración de la tierra, además de la dirección de la misma. Pero no se tenía referencia alguna, del instante en que se presentó el sismo, o peor aún de la forma y de la magnitud de este. Por lo que el principal limitante de este sistema era la falta de un sismograma (registra la forma de la onda sísmica referenciada en tiempo). Este defecto, se ha ido superando poco a poco con la llegada de las nuevas tecnologías y con la aplicación de nuevas técnicas de detección.

1.2.1 Principio de Funcionamiento.

Un sismógrafo consta de: una masa de libre oscilación (de masa “m”), un resorte (con una constante de rigidez “K”), un sistema de amortiguamiento (constante de amortiguamiento “c”), y un módulo de soporte en donde se encuentra fijado el resorte al que esta acoplado la masa. Este módulo de soporte esta sobre la superficie en movimiento (SUELO).

El punto en reposo se logra aprovechando el principio de la inercia por el cual, todo cuerpo ofrece resistencia al movimiento o al cambio de su velocidad cuando se le aplica una fuerza. Este hecho da un punto de referencia inicialmente estático (masa en suspensión).

Con la presencia de las ondas sísmicas, todo el mecanismo del sismógrafo se mueve conjuntamente con el suelo, excepto la masa en reposo, que permanece estática momentáneamente aprovechando la tracción y contracción del resorte. Esto da un punto de referencia a partir del que se pueden medir los movimientos oscilatorios. Pero a medida que transcurre el tiempo la masa comienza a oscilar levemente haciendo que la información registrada no responda al movimiento real, y por ende presentándose errores en los registros. Por este motivo se ve necesaria la inclusión de un sistema de amortiguamiento, garantizando así la existencia de un punto de referencia como tal.

El sistema de amortiguamiento antiguo utilizaba aceite, luego fueron evolucionando, hasta que en la actualidad se utilizan imanes o bobinas. Análogamente el sismograma era realizado por un lápiz acoplado a la masa de libre oscilación, el mismo que registraba las ondas sísmicas en un tambor que giraba a tiempo constante. Hoy en día los sismogramas son hechos por sistemas digitales y cuyo referenciamiento de tiempo es calibrado por relojes Satélites o GPS. Logrando mayor calidad y confiabilidad en los datos.

1.2.2 Ecuación diferencial del geófono.

Si se considera la Fig.1.2 .

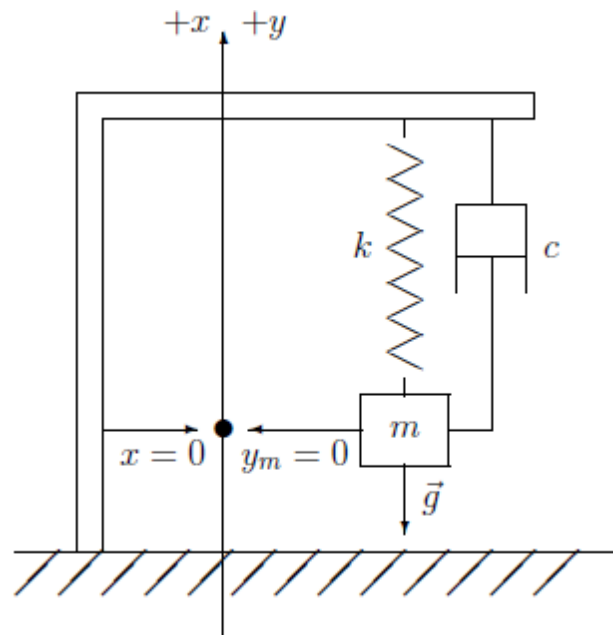


Fig.1.2 Modelo interno de un geófono

Dónde:



$x(t)$, posición del soporte o del suelo, respecto del eje de coordenadas.

$y_m(t)$, posición de la masa de libre oscilación, respecto del eje de coordenadas.

$y(t) = y_m(t) - x(t)$, posición relativa de la masa en suspensión, respecto del soporte del geófono o la tierra.

$\dot{y}_m(t)$, velocidad del movimiento de la masa.

$\ddot{y}_m(t)$, aceleración del movimiento de la masa.

$\dot{x}(t)$, velocidad del movimiento de la tierra o del soporte.

k , constante de rigidez del resorte.

c , constante de amortiguamiento.

m , masa del elemento en suspensión.

Los parámetros: m , c , k , son los que definen las características principales del sistema. Como su frecuencia natural (f_0, ω_0, T_0) , constante de amortiguamiento crítico (ξ) , determinan, el rango de frecuencias o ancho de banda para el cual el sismómetro está diseñado. Como es lógico con la calibración de tales parámetros (seteo de la sensibilidad del dispositivo) se pueden lograr distintas respuestas de frecuencia, logrando sentir varios tipos de sismos.

Si las condiciones iniciales para $t \leq 0$: $x(t) = y_m(t) = 0$; el suelo se encuentra en reposo y la masa en estado de equilibrio. Cuando se presenta un movimiento el soporte del geófono oscila junto con la tierra, pero la masa de equilibrio no. Está presente una resistencia al movimiento:

$$m * \ddot{y}_m(t)$$

También en ese instante se produce un desplazamiento en el resorte proporcional a la fuerza:

$$-k(y_m(t) - x(t))$$

Análogamente para el elemento amortiguador:

$$-c(\dot{y}_m(t) - \dot{x}(t))$$

En base a la segunda ley de Newton, se plantea la ecuación del movimiento de la masa desde su posición de equilibrio:

$$m * \ddot{y}_m(t) + k(y_m(t) - x(t)) + c(\dot{y}_m(t) - \dot{x}(t)) = 0$$

Reemplazando: $y(t) = y_m(t) - x(t)$, $2\beta = c/m$, $\omega_0^2 = k/m$, sumando $-m\ddot{x}(t)$, y dividiendo para m a los dos lados de la ecuación se tiene:

$$\ddot{y}(t) + 2\beta\dot{y}(t) + \omega_0^2 y(t) = -\ddot{x}(t) \quad (1)$$

El sismómetro solo mide el desplazamiento relativo de la masa, más no el absoluto respecto del eje de coordenadas. Para la solución de la ecuación se pueden utilizar dos métodos, la transformada de Fourier y la transformada de Laplace.

Transformada de Fourier:

La mayoría de funciones pueden ser representadas mediante una suma de armónicos de distintas frecuencias. Considerando este hecho se obtiene la solución:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{\omega^2}{\omega_0^2 - \omega^2 + j2\beta\omega}$$

En forma exponencial:

$$H(\omega) = \frac{\omega^2}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}} e^{j\varphi}$$

Dónde:

$$\varphi(\omega) = \arctan \frac{-2\beta\omega}{\omega_0^2 - \omega^2}$$

La constante de amortiguamiento puede ser definida como: $\xi = \beta/\omega_0$, su valor típico está entre 0.6 – 0.7. Estos valores dan como resultado una respuesta más plana en la zona de la frecuencia natural del sismómetro. Con lo que:

$$|H(\omega)| = \frac{\omega^2}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\xi^2\omega_0^2\omega^2}} = \frac{\omega^2/\omega_0^2}{\sqrt{(1 - \omega^2/\omega_0^2)^2 + 4\xi^2\omega^2/\omega_0^2}}$$
$$\phi(\omega) = \arctan \frac{-2\xi\omega/\omega_0^2}{1 - \omega^2/\omega_0^2}$$

Transformada de Laplace:

Análogamente, tomando la transformada de Laplace tanto de la función de entrada ($X(s)$) como de la salida ($Y(s)$), y aplicando la propiedad de diferenciación de la transformada:

$$\ddot{y} + 2\beta\dot{y} + \omega_0^2 y = -\ddot{x}$$

$$(s^2 + 2\beta s + \omega_0^2)Y(s) - (s + 2\beta)y(0) - \dot{y}(0) = -s^2 X(s) + sx(0) + \dot{x}(0)$$

Un caso típico en $t = 0$, el sismómetro y la masa se encuentran en reposo: $x(0) = \dot{x}(0) = 0$, $y(0) = \dot{y}(0) = 0$. La función de transferencia es:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{-s^2}{s^2 + 2\beta s + \omega_0^2}$$

Se puede observar como la función de entrada (movimiento del suelo) es afectada por la función de transferencia del sismómetro. Como en todo dispositivo transductor la señal original es alterada por la función de transferencia del elemento. Además de que la multiplicación en el dominio de la frecuencia corresponde a la convolución en el dominio del tiempo. Un caso especial es cuando la entrada es una función impulso, donde la respuesta del geófono sería igual a la salida.

En la actualidad la mayoría de los sismómetros miden la variación de la velocidad de la masa, teniendo como entrada el movimiento del suelo. Esto se debe a que el movimiento de la masa junto con la bobina causa la inducción de un voltaje variable en sus terminales. Entonces la función de transferencia entre el desplazamiento del suelo y la velocidad del sismómetro es:

$$\frac{sY(s)}{X(s)} = \frac{-s^3}{s^2 + 2\beta s + \omega_0^2}$$

Un aspecto importante es la calibración del sismómetro. Que únicamente consta de la aplicación de una señal de corriente de frecuencia conocida o de frecuencia cero a una bobina de calibración. Esto producirá una fuerza sobre la masa en suspensión, causando que esta obtenga una aceleración proporcional a la corriente aplicada.

1.2.4 Geófonos de una componente: vertical y horizontal.

Existen distintos tipos de sensores que detectan los movimientos del suelo, las diferencias principales están en su ancho de banda y la amplitud que estos son capaces de detectar. Los sismómetros (sensores de alta ganancia) pueden registrar la velocidad de los movimientos: cercanos y de pequeña magnitud, terremotos a grandes distancias de la estación (típicamente hasta 1 cm/s).

Mientras los acelerómetros (sensores de baja ganancia) están diseñados para registrar terremotos de gran magnitud, que típicamente que alcanzan valores hasta 2000 cm/s^2 . Claramente se puede notar que la diferencia entre los dos dispositivos radica en que; el primero es un transductor de la velocidad, mientras que el otro es uno de la aceleración del movimiento del suelo.

El geófono puede capturar los movimientos verticales u horizontales (véase Fig.1.2), dependiendo de la posición de; la masa, el resorte, y el soporte. De ahí su denominación (geófono vertical u horizontal).

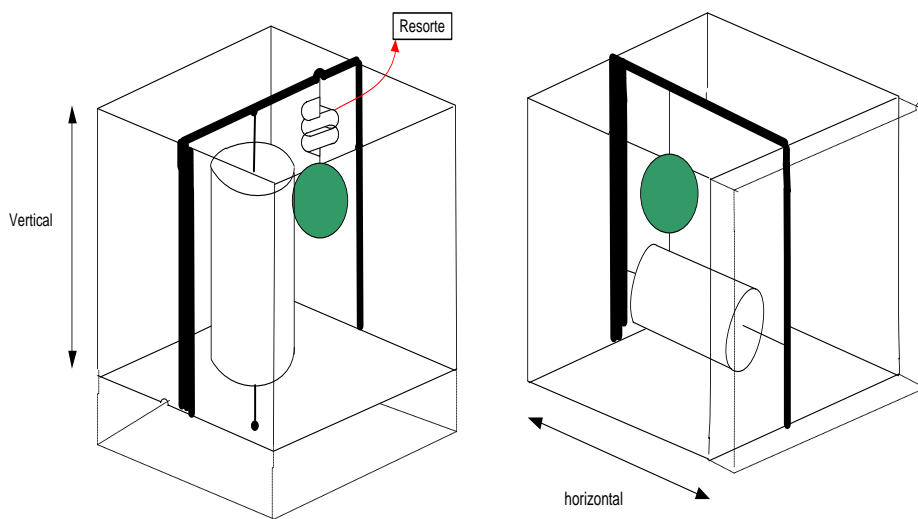


Fig.1.3 Geófono vertical y horizontal

Los geófonos se diferencian en los distintos materiales que se utilizan para su construcción, en base a esto se tienen los siguientes tipos; Electromagnético, de reluctancia variable, de capacidad, y Piezoeléctrico o tipo de presión. Pero los más utilizados son los primeros. Los sismómetros electromagnéticos típicos utilizan un sistema análogo al mostrado en la Fig.1.4.

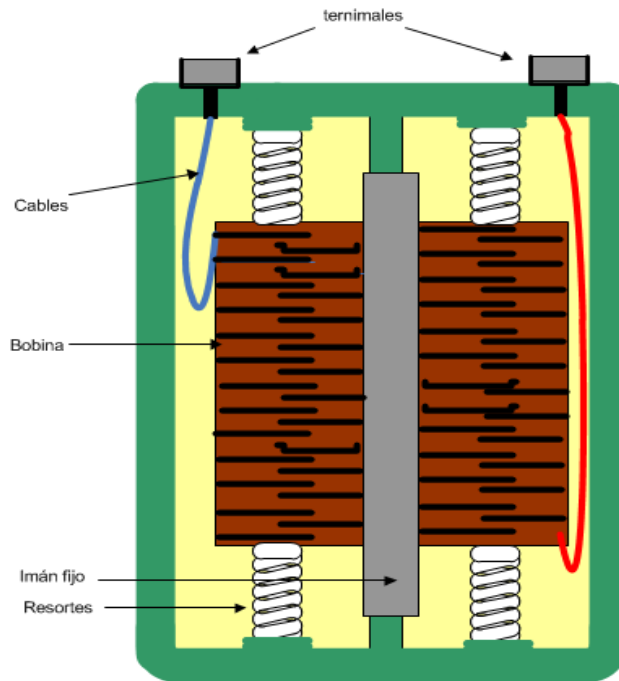


Fig.1.4 Sismómetros electromagnéticos

La frecuencia natural del sistema es:

$$\omega_0 = \sqrt{\frac{k}{m} - \frac{c^2}{4m^2}}$$

La masa de libre oscilación es la bobina acoplada a los resortes, la parte fija corresponde al imán permanente y a todos los elementos auxiliares que sirven de soporte. Como la bobina se encuentra dentro de un campo magnético permanente, el movimiento relativo entre la bobina y el soporte (cuando se produce un evento sísmico) produce la inducción de una FEM en la bobina, generándose un voltaje a través de sus terminales.

$$E = -L \frac{dH}{dt}$$

Dónde:

L es la inductancia de la bobina.

E es la FEM inducida.

H es la porción del campo magnético del imán permanente ocupado por la bobina.

Suponiendo que $H = H_0x$, donde x es la porción de la bobina presente dentro del entrehierro:

$$E = -LH_0 \frac{dx}{dt}$$

Claramente se observa que el voltaje generado es proporcional a la velocidad del movimiento de la masa.

Para mejorar el rango dinámico de los sensores, se incluye un sistema de retroalimentación forzada (sismómetros de retroalimentación forzada). Estos a diferencia de los sismómetros estándar, tienen un transductor de desplazamiento que mide el movimiento de la masa. Además de un elemento electromecánico encargado de producir la fuerza que amortigua el movimiento de masa respecto del soporte del sismómetro, esta fuerza corresponde a la aceleración de la tierra.

El rango dinámico de estos sismómetros es determinado por el sistema electrónico de retroalimentación forzada, más no por el mecánico del sismómetro. Logrando con esto un incremento de 100 dB respecto de los sistemas convencionales.

Los avances tecnológicos han dado lugar a la creación de mejores sistemas de detección de sismos. Esto trae consigo, el mejoramiento de algunos de los parámetros de los sismómetros entre los cuales podemos mencionar: aumento del ancho de banda de respuesta, aumento del rango dinámico, discriminación de eventos, etc.

1.2.5 Geófonos de tres componentes.

Los movimientos de la tierra se producen en todas las direcciones, motivo por el cual también se necesitan elementos sensores que registren el movimiento no solo en la dirección vertical, sino además en al menos dos direcciones horizontales, típicamente N-S y E-O. Esto puede lograrse de distintas formas, una de las cuales es; utilizar dos geófonos horizontales perpendicularmente colocados



más uno vertical, o en su defecto un solo geófono que tenga integrado los tres. Los instrumentos que sensan el movimiento pueden ser diferentes dependiendo del tipo de señal que se quiera registrar, y de su ubicación en el espectro de frecuencia. Los transductores típicos más usados son los geófonos y los acelerómetros. Por lo que ambos pueden estar ubicados en una misma estación, dependiendo de los requerimientos de esta.

Las tres señales analógicas de los transductores son tomadas para su tratamiento; acondicionamiento, conversión analógica-digital, algún tipo de procesamiento digital, almacenamiento local, y transmisión hacia la estación base. Este sistema total se denomina “sismógrafo”.

1.3 DIGITALIZACION DE SEÑALES

1.3.1 Origen de las ondas sísmicas

Las ondas sísmicas son originadas por la liberación de energía, en el momento que se produce una ruptura profunda dentro de la tierra (epicentro). Luego se propagan a través de la corteza terrestre, hasta llegar al hipocentro (superficie de la tierra). Y es en ese momento que el transductor siente el movimiento y se disparan sus mecanismos de registro. Entonces se puede notar que existe un retardo y atenuación en las ondas registradas, respecto de las iniciales. Es por este motivo que la propagación a través de la corteza terrestre se puede considerar como un filtro, el mismo que tiene una función de transferencia específica.

Denominemos que las ondas sísmicas tienen una función de tiempo $S(t)$, y que el filtro de la corteza terrestre una $P(t)$. Pero además los sismómetros tienen por defecto una función de transferencia adicional a las otras, la misma que depende del tipo de elemento, y de sus características intrínsecas. A la función del elemento de medición la denominaremos $I(t)$.

Como resultado de estos efectos, se tendrá una función final que es el resultado de la convolución de todas las funciones involucradas, además del ruido de la superficie, y las señales pequeñas de otros sismos que también deben ser incluidas a esta función, por lo que:

$$R(t) = S(t) * P(t) * I(t)$$



Cuando las señales del sismo son muy débiles, debido a que este es muy pequeño o que su epicentro está demasiado alejado de la estación de registro. Se corre el riesgo de que las ondas de energía sean comparables a las del ruido implícito en los transductores, y por esta razón se pierda esta información.

Las señales físicas en forma natural se presentan en forma analógica, pero en la mayoría de los procesos de tratamiento de estas, se utiliza un proceso de muestreo para digitalizarlas. Por lo que se presentan 4 dominios: tiempo continuo, frecuencia continua, tiempo discreto y frecuencia discreta. Y las respectivas técnicas matemáticas para pasar de un dominio a otro. Como por ejemplo: la transformada de Fourier, utilizada en el cambio del dominio de tiempo al de la frecuencia.

Ahora bien, también se debería incluir el ruido de los conversores A/D, para el caso de los sistemas modernos que utilizan este tipo de elementos en el tratamiento de las señales. La función de transferencia de los transductores y el efecto de la digitalización es lo único que se conoce más no así las funciones $S(t)$ y $I(t)$. Para la recuperación de algunas de las propiedades características de la señal $S(t)$, se pueden utilizar diferentes técnicas de deconvolución, obteniendo así una función muy parecida a su forma original. Estos procesos se pueden realizar en el dominio del tiempo como en el de la frecuencia.

1.3.2 Frecuencia de muestreo para un ADC

Para la conversión se debe poner cuidado en la selección del tiempo de muestreo y en la amplitud. Si la amplitud de la señal analógica, a la entrada del conversor, coincide con las máximas designadas para el A/D, entonces se introduce ruido por efecto de saturación. El periodo de muestreo se selecciona de acuerdo al teorema de Nyquist:

$$f_s = 2 * f_m \text{ *Mínima frecuencia de muestreo según Nyquist.*}$$

$$f_s = 2 * f_m * n \text{ *Mínima frecuencia de muestreo según Nyquist, para varios canales.*}$$

Dónde: f_m máxima frecuencia de entre las señales.

f_s frecuencia de muestreo del dispositivo conversor.

Número de canales a muestrear.

El periodo de muestreo debe ser entonces:

$$T_x = \frac{1}{f_s}$$

Que para el caso de varios canales a muestrear, este periodo es el correspondiente al del reloj del ADC. Este periodo nos garantiza la correcta cuantificación de todas las señales. Pero se debe tener en cuenta que el sobre muestreo puede ocasionar una disminución del ancho de banda de determinado canal, para el caso en que se requiera transmisión de los datos.

Un efecto conocido que se produce en la digitalización es el “aliasing” que es debido a una representación muy pobre de la señal original, por la selección de una f_s por debajo de la mínima de Nyquist. Esto se minimiza colocando un filtro paso-bajo (“filtro anti aliasing”) cuya frecuencia de corte es igual a la crítica, eliminando todas las frecuencias indeseables por encima de la máxima de muestreo. La frecuencia de corte es de 500 Hz, debido a que este sistema en un futuro, en vez de geófonos utilizará acelerómetros cuya respuesta en frecuencia es más elevada, típicamente hasta 1/3 de la natural.

En la actualidad los sistemas de adquisición y tratamiento de señales sísmicas tienen un dataloger. Que normalmente consta de: un ADC y una especie de computadora digital para tratamiento de señales, un dispositivo de almacenamiento local, y un interfaz hacia una red central.

El ADC toma la señal analógica proveniente del sismómetro y la convierte en una de tipo digital, es decir la guarda en un determinado número de bits, esto dependiendo del rango del conversor. Por lo general los digitalizadores más utilizados tienen un rango de 12 a 24 bits. Esto nos da como resultado un número de niveles o cuentas (niveles de cuantificación) en los registros digitales, que oscilan entre un mínimo y un máximo.

En el ancho de banda del sismómetro, en donde la respuesta es plana, la transformación entre la velocidad o aceleración de la tierra a los niveles de

cuantificación se la realiza mediante una constante (ganancia del sismómetro). Esto lógicamente dependiendo si se trata de un geófono o un acelerómetro.

1.3.3 Resolución.

En la digitalización el ADC escoge entre todos los niveles de cuantificación el entero más cercano al valor analógico presente, pero casi nunca este es igual a la magnitud real de la señal original. Esto trae como consecuencia un error de redondeo, en el registro de los datos.

$\sigma_e = n_i/\sqrt{12}$ Donde n_i es el valor más pequeño (primer nivel de cuantificación).

El error de cuantificación se puede disminuir notablemente añadiendo más bits por cada muestra. Ya que aumentara el número de niveles, se reducirán los espaciamentos entre estos, y por ende se disminuirá este error.

Por definición, la mayor varianza de x viene dada por:

$$\sigma_e^2 = \int_a^b (x - m_e)^2 p(x) dx$$

Ahora, con el correcto redondeo: $a=-1/2$, $b=1/2$, y $m_e=0$, y además considerando que $p(x)$ entre a y b tiene una distribución equiprobable, entonces:

$$p(x) = \frac{1}{a - b} = \frac{1}{\frac{1}{2} + \frac{1}{2}} = 1$$

$$\sigma_e^2 = \int_a^b (x)^2 dx = 1/12$$

1.3.4 Rango dinámico

La principal especificación que se debe tener en cuenta al momento de escoger un sismómetro es su rango dinámico. El cual define la razón entre el máximo valor y el mínimo, que pueden ser medidos por el dispositivo; está dado por la relación:

$$DR = \frac{V_{max}}{V_{min}}$$

En el contexto de la instrumentación sísmica, es usualmente la medida voltaje de salida para un sismómetro, o valor a ser digitalizado.

Este es un indicador que normalmente viene especificado en dB:

$$DR = 20 * \log\left(\frac{V_{max}}{V_{min}}\right) [dB]$$

El suelo tiene un rango dinámico de:

$$DR = 20 * \log\left(\frac{10^{10}}{1}\right)$$

$$DR = 10 * 20 * \log(10)$$

$$DR = 200 \text{ dB}$$

Similarmente, para sismos de magnitud m se el rango dinámico es equivalente a:

$$DR = (20 * m) \text{ dB}$$

Donde los 10^{10} corresponden a una zona con actividad sísmica muy grande (terremotos con una intensidad de orden de magnitud de 10). Y 1 el mínimo valor, al ruido presente en zonas inactivas, o de sismos muy pequeños. Por lo tanto sismógrafos deben ser capaces de capturar todo este rango, garantizando la fidelidad de los registros sísmicos.

Modernos sismógrafos tienen grabadores de registros digitales, los cuales convierten el voltaje analógico del sismómetro en salida de conteo digital. Entonces el voltaje salida del sismómetro es cuantificado y digitalizado, un bit se utiliza para representar el signo de la señal, y los restantes para la magnitud. Aquí el rango dinámico nominal viene dado en función del número de bits a emplear en la codificación. Por lo que se establece:

$$DR_{dB} = 20 * \log(2^{n-1}) = (n - 1) * 6.02$$

Donde, n es el número de bits del digitalizador.

Por ejemplo, para un número de 8 bits se tiene un rango dinámico de 42 dB. En la tabla 1.1 se presentan las especificaciones correspondientes a un número de bits específico.

| #bits | Rango dinámico | DR[dB] | Orden de Magnitud |
|-------|----------------|----------|-------------------|
| 8 | 256/2 | 42.1442 | 2.10720997 |
| 12 | 4096/2 | 66.2266 | 3.311329952 |
| 16 | 65536/2 | 90.309 | 4.515449935 |
| 20 | 1048576/2 | 114.3914 | 5.719569918 |
| 24 | 16777216/2 | 138.4738 | 6.9236899 |

Tabla 1.1 Rango dinámico



CAPITULO 2: UTILITARIOS PARA EL DISEÑO

2.1 PROTEUS

Proteus es un entorno diseñado para la simulación, construcción de equipos electrónicos en todas sus etapas. Proteus Sistema Virtual de Modelado (VSM) combina el modo mixto en la simulación de circuito de SPICE, componentes animados y modelos completos de diseños basados en microprocesadores para facilitar la simulación de microcontroladores.

El software Proteus es una aplicación CAD (Aplicación de diseño) que está compuesto de tres módulos, ISIS (Intelligent Schematic Input System) módulo de captura de esquemas, ARES (Advanced Routing Modelling) módulo para realizar circuitos impresos y el VSM módulo virtual de simulación.

2.1.1 Interfaz de usuario

ISIS

Al iniciar ISIS, se presenta la ventana de diseño de la figura 2.1 en la que se aprecian dos zonas, en la izquierda un visor del plano del proyecto, debajo la ventana para mostrar dispositivos y a la derecha la zona de trabajo propiamente dicha, en la que el diseñador realizara los circuitos electrónicos con sus respectivos componentes, posteriormente podrá ejecutar un modelo virtual que simula el proyecto electrónico.

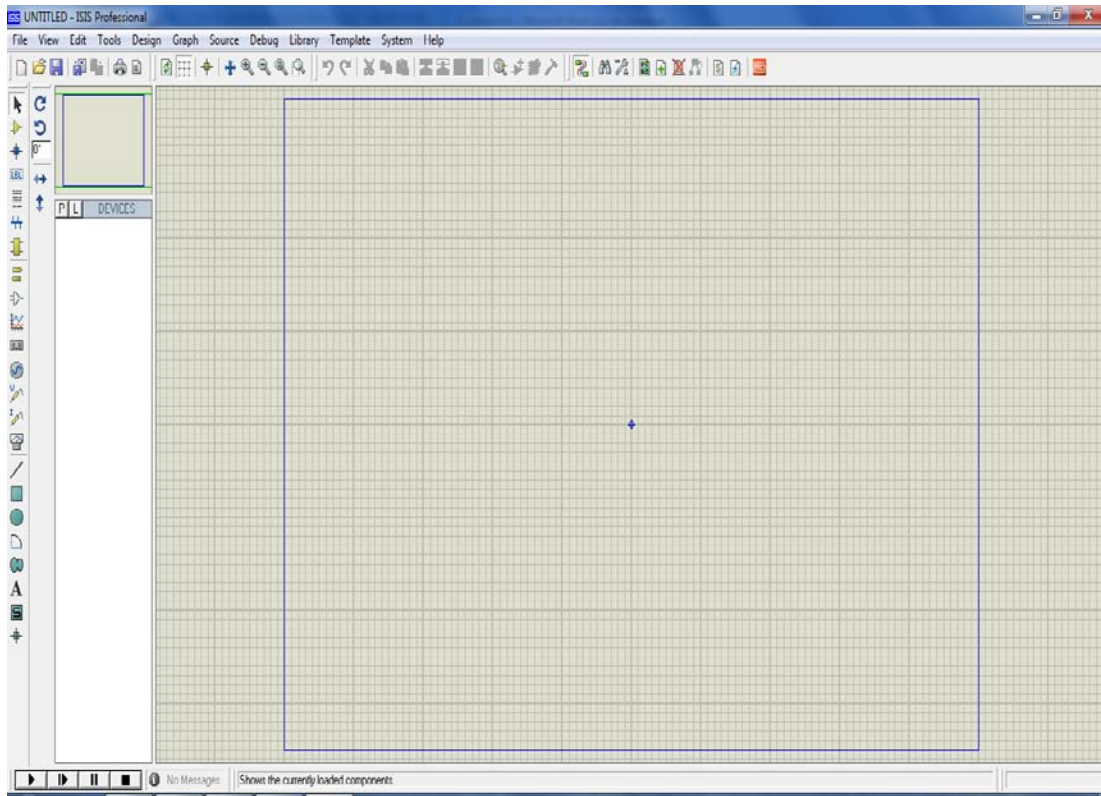


Figura 2.1 Ventana del módulo ISIS

En la columna de la izquierda se da un doble-click sobre el botón **P** que nos antepone la ventana de librerías, donde se elige el dispositivo que se requiere para la simulación, eligiendo otros componentes del mismo modo, lo que enviará el dispositivo seleccionado a la ventana principal. Proteus incorpora una librería de más de 6.000 modelos de dispositivos digitales y analógicos.

ARES

Proteus VSM tiene la capacidad de pasar el diseño de ISIS a un programa integrado llamado ARES (figura 2.2) en el cual se puede llevar a cabo el desarrollo de placas de circuitos impresos.

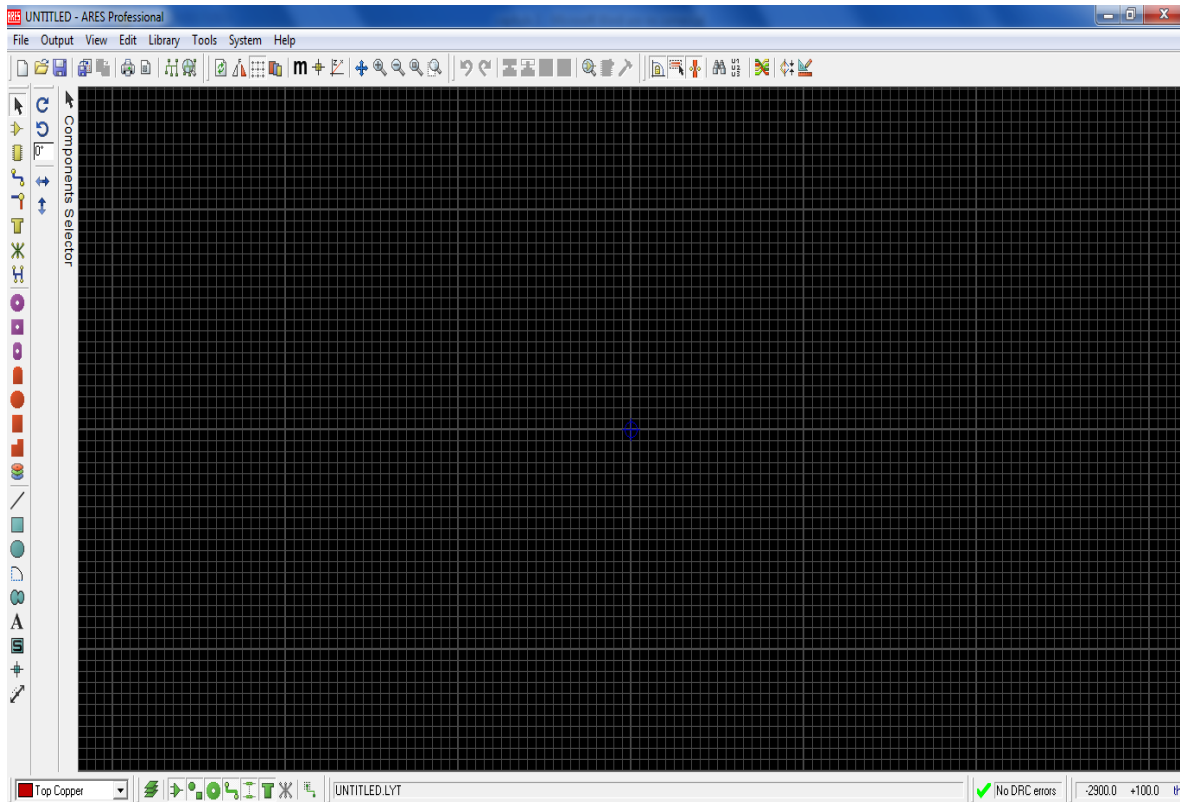


Figura 2.2 Ventana del módulo Ares

VSM

Virtual System Modelling, es el módulo de simulación.



Figura 2.3 Ventana del módulo VSM

ISIS es el módulo de captura y simulación de esquemas, tiene varios objetivos:

- Simular el circuito para verificar el correcto funcionamiento.
- Obtener una buena representación gráfica del circuito electrónico.
- Generar una placa de circuito impreso del esquema capturado en Ares.

En la figura 2.4 se presenta el entorno laboral de ISIS donde se puede apreciar sus componentes.

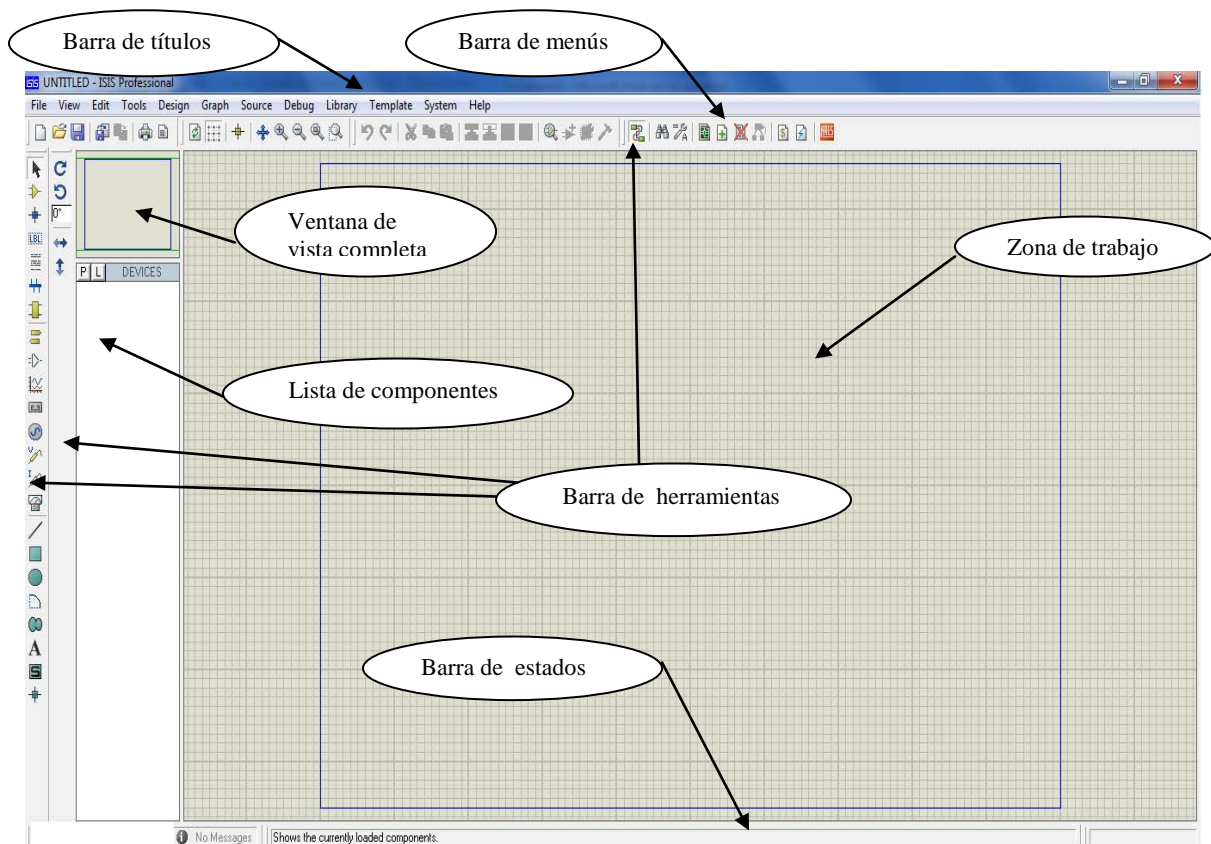


Figura 2.4 Componentes del módulo ISISI

Barra de título:

En ella se muestra el icono del programa, el nombre del fichero abierto y la leyenda ISIS Profesional.

Barra de menús:

Permite el acceso a las opciones del programa, algunas opciones están disponibles en los iconos de las barras de herramientas.

Barra de herramientas:

Son numerosas, el usuario las puede colocar en distintas posiciones de los bordes de la ventana y son de suma importancia, ya que con ellos se puede seleccionar la opción indicada para realizar el diseño.

**Barra de estado:**

En esta barra se muestran mensajes informativos acerca de las opciones del menú, el estado de los componentes en las simulaciones. Indica las coordenadas del cursor, las unidades son en milésimas de pulgada.

Zona de trabajo:

Lugar donde se creara el diseño.

Ventana de vista completa:

Aquí se muestra una visión global del diseño, y mediante el puntero se puede seleccionar que zona del diseño estará visible en la ventana de edición.

Lista de componentes:

En esta lista aparecen todos los componentes, terminales, pines, generadores, microcontroladores, resistencias, etc. Que se requieren para la construcción de un esquema, esta ventana dispone de 2 botones que nos permite acceder a las librerías de componentes incluidas en ISIS.

2.1.2 Captura de esquemas**Uso del módulo ISIS**

Una vez iniciado el módulo ISIS se debe seleccionar el componente que se va a utilizar. Para esto se da doble-click sobre el icono P que abrirá una ventana como la que se muestra en la figura 2.5, en la ventana desplegada se observan tres columnas con distintas funciones cada una. En la columna de la izquierda se selecciona el dispositivo a utilizar, en la mitad las características, marca en la columna de la derecha los pines y forma para la placa del circuito impreso.

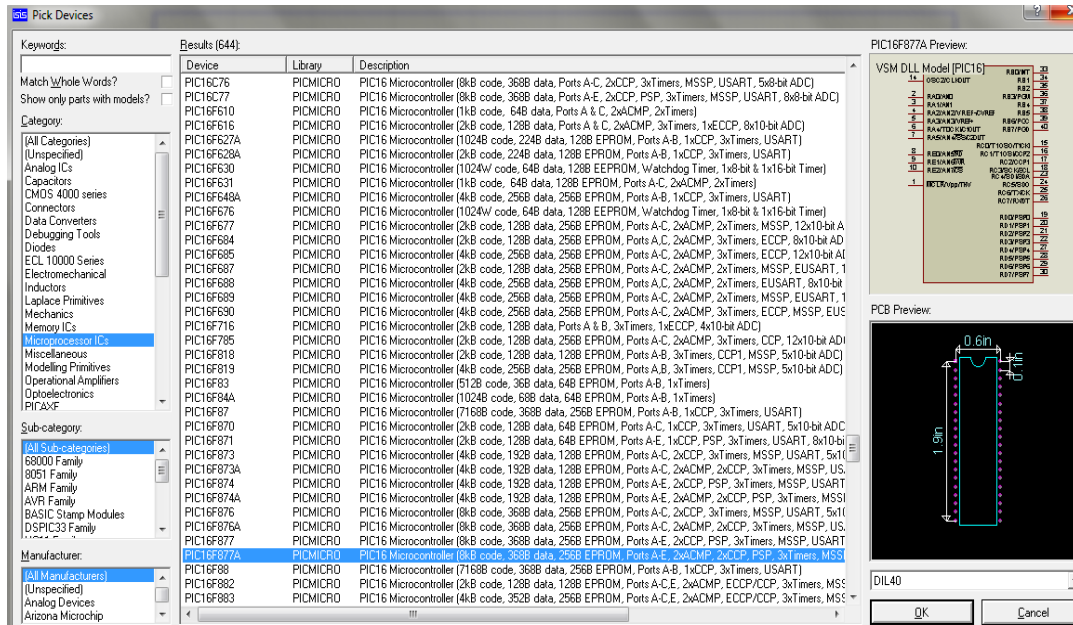


Figura 2.5 Lista de componentes que posee Proteus

2.2 MIKROC

MikroC PRO for PIC 3.2 organiza las aplicaciones de los proyectos que consisten en un solo fichero del proyecto (fichero con extensión .mcppi) o en uno o más ficheros fuentes (ficheros con extensión .c). Los ficheros fuentes son denominados cabeceras en el lenguaje de programación mikroC for pic 3.2.

Un fichero de proyecto contiene lo siguiente:

- Nombre del proyecto y la descripción opcional;
- Dispositivo destino (tipo de microcontrolador) utilizado;
- Frecuencia de reloj del microcontrolador;
- Lista de ficheros fuentes de proyecto;
- Ficheros binarios (*.mcl); y
- Otros ficheros (.hex).

2.2.1 Interfaz de usuario

Antes de crear un nuevo proyecto, es necesario hacer lo siguiente:

Paso 1: Instalar el compilador

Paso 2: Iniciar el compilador

Al iniciar el compilador mikroC PRO for PIC 3.2 realizando doble clic en el icono apropiado aparecerá mikroC PRO for PIC IDE 3.2 en la pantalla.

Para crear un nuevo proyecto en Mikroc.

En la barra de menú en Project se selecciona la opción New Project muestra en la Figura 2.6

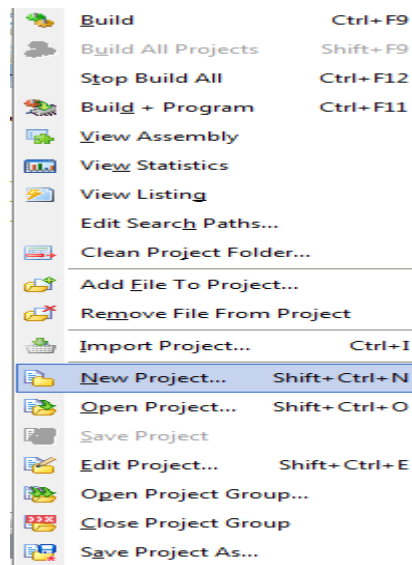


Figura 2.6

A continuación aparece la ventana de la figura 2.7 donde se selecciona el tipo de pic a utilizar, se hace doble Click en Next.

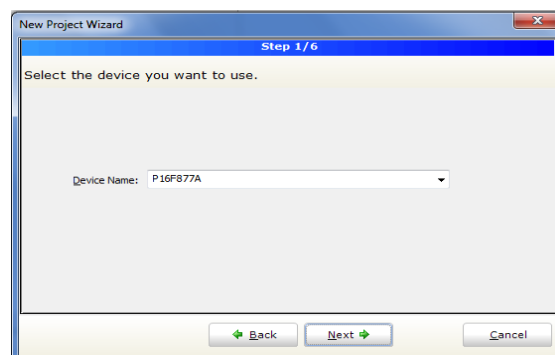


Figura 2.7 Selección del tipo de Pic

En la siguiente ventana la frecuencia del reloj del PIC en Megahercios, se hace doble Click en Next

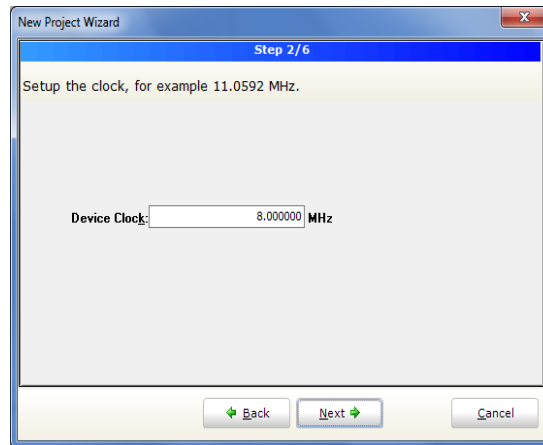


Figura 2.8

El nombre del proyecto y la ruta donde se guardara, se hace doble Click en Next

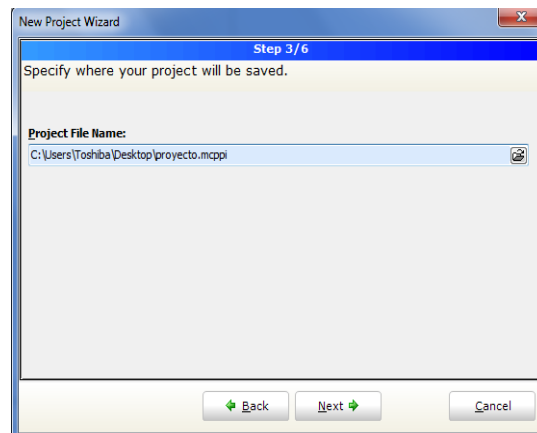


Figura 2.9

Si el nuevo proyecto consiste en varios ficheros fuentes, se necesita especificarlos y hacer click en el botón **Add** para incluirlos. En la mayoría de los casos no hay ficheros fuentes adicionales.

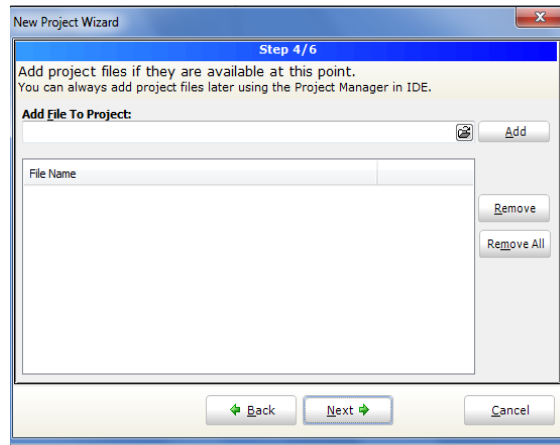


Figura 2.10

En el siguiente paso se selecciona las librerías que se van a utilizar para crear el programa. Por lo general se selecciona cargar todas las librerías por defecto.

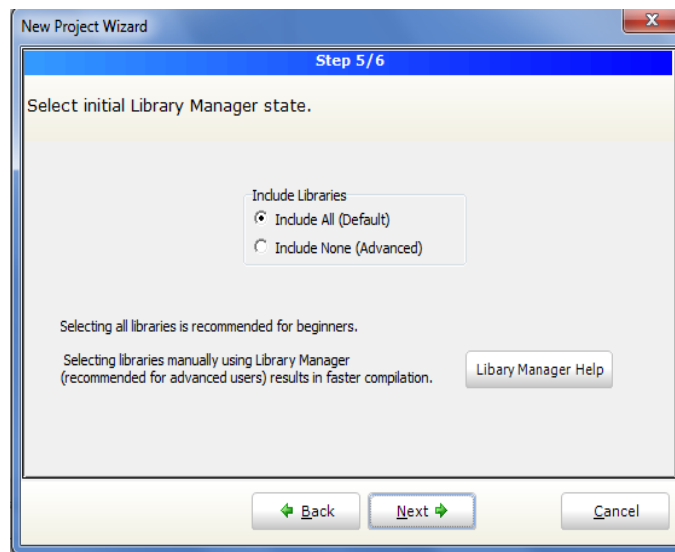


Figura 2.11

Por último es necesario confirmar todas las opciones seleccionadas anteriormente, dando doble-click en finish.

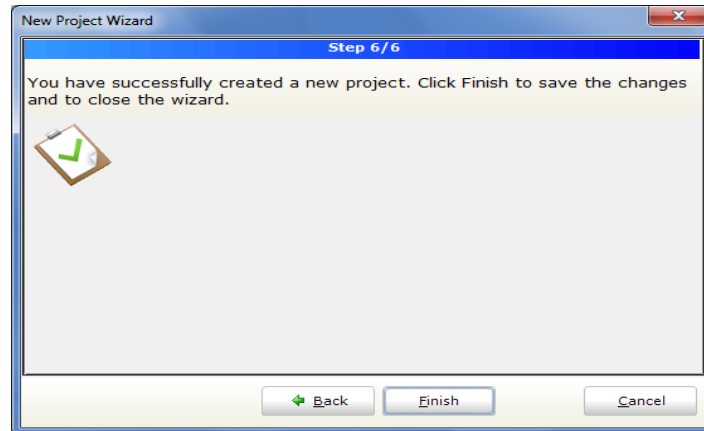


Figura 2.12

Después de crear el proyecto, aparecerá una ventana en blanco donde se deberá escribir el programa fuente en lenguaje C para luego compilarlo.

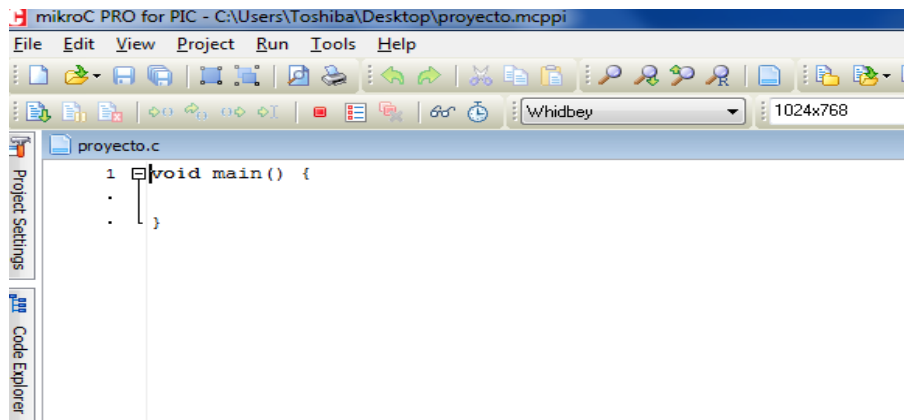


Figura 2.13 Ventana de Mikroc Pro 3.1

Una vez que se ha creado el programa es necesario compilarlo en un código de programa hexadecimal (.hex) para esto se seleccione una de las opciones de compilación del menú **Project**:

- Para crear un fichero HEX seleccione la opción Build (Ctrl+F9) del menú Project o hacer click en el icono Build de la barra de herramientas.



- La opción Build All Projects (Shift+F9) compila todos los ficheros del proyecto, librerías (si hay código fuente para ellas) y los ficheros para el microcontrolador utilizado.

Todos los errores detectados durante la compilación aparecerán en la ventana Message. Si no hay errores, mikroC PRO for PIC generará ficheros de salida, en caso contrario indicas las líneas de programación en las que se ha producido el error.

El proceso de programación del Pic (18f452, 16f877A) se realiza con el programador Picstar Plus o PicKit2 de Microchip.

2.2.2 Manejo de Interrupciones en Mikroc

Una interrupción es un evento que requiere, detener la ejecución normal del programa y a continuación ejecutar un código de programación relacionada con el hecho causante de la interrupción. Las interrupciones pueden ser internas (por algún evento dentro del chip) o externas. Las interrupciones son necesarias en aplicaciones, donde se requiere la atención inmediata del Pic.

El PIC18F452 tiene varias fuentes de interrupciones internas como externas. Las fuentes para que se produzcan interrupciones es:

- Las interrupciones externas INT0, INT1, INT2. (puerto B)
- Sobre flujo del Timer0.

Las fuentes de interrupción periféricas:

- Parallel slave port read/write interrupt
- A/D conversion complete interrupt
- USART receive interrupt
- USART transmit interrupt
- Synchronous serial port interrupt
- CCP1 interrupt
- CCP2 interrupt
- TMR0 overflow interrupt
- TMR1 overflow interrupt
- TMR2 overflow interrupt
- Timer 3 overflow interrupt
- Comparator interrupt



EEPROM/FLASH write interrupt
 Low-voltage detect interrupt

Cada fuente de interrupción (excepto INT0) tiene tres bits para controlar su funcionamiento. Estos bits son:

- Un bit de bandera para indicar si una interrupción ha ocurrido. Este bit tiene un nombre que terminan en ...IF.
- Un bit para habilitar o deshabilitar la fuente de interrupción. Este bit tiene el nombre que termina en ...IE.
- Un bit para seleccionar si es de alta o baja prioridad. Este bit tiene el nombre que termina en ...IP.

Registro RCON

El bit superior del registro RCON, llamado IPEN, es usado para habilitar la interrupción de acuerdo a los niveles de prioridad, o para deshabilitar la interrupción.

Para que una interrupción sea aceptada por el Pic se deben cumplir las siguientes condiciones:

1. La interrupción debe estar habilitada en los bits de registro. Por ejemplo, si la fuente de interrupción es INT0, entonces INT0IE del registro INTCON debe ser establecido en 1.
2. La bandera de la interrupción debe ser puesta en cero. Por ejemplo, si el fuente de interrupción es INT0 pin de interrupción externa, INT0IF de registro INTCON se borra a 0.
3. La interrupción debe ser activada/desactivada con PEIE de INTCON, este bit se debe poner 1 si la fuente de la interrupción es externa.
4. En el bit GIE of INTCON habilita/deshabilita todas las interrupciones del Pic, para habilitar las interrupciones este bit debe estar en 1.

Registro RCON

| | | | | | | | |
|-------|-----|-----|-------|-----|-----|-------|-------|
| R/W-0 | U-0 | U-0 | R/W-1 | R-1 | R-1 | R/W-0 | R/W-0 |
| IPEN | - | - | RI | TO | PD | POR | BOR |



Bit 7 **IPEN** habilita y deshabilita interrupciones globales

1 = habilita las interrupciones por prioridad

0 = deshabilita las interrupciones por prioridad

Bit 6-5 **indeterminado** se lee cero

Bit 4 **RI** bandera de la instrucción reset

1= la instrucción reset no ha sido ejecutada

0=la instrucción reset ha sido ejecutada

Bit 3 **TO** bandera de perro guardián

1=después de un power-up, clwdr

0= a wdt ocurrió un reset

Bit 2 **PD** bandera de detección de apagado del procesador

1=después de un power-up, clwdr

0=debido a la ejecución de la instrucción sleep

Bit 1 **POR** bit de estado power-on reset

1= Un Power-on Reset no ha ocurrido

0=Un Power-on Reset ocurrió

Debe ser colocado en 1 después de que un Power onReset ha ocurrido)

Bit 0 **BOR** Brown-out

1= Un Brown-out Reset no ha ocurrido.

0= Un Brown-out Reset ocurrió.

Debe ser colocado en 1 después de que un Brown-out Reset ha ocurrido)

Interrupciones. Registro **INTCON**

| | | | | | | | |
|----------|-----------|--------|--------|-------|--------|--------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GIE/GIEH | PEIE/GIEL | TMROIE | INTOIE | RBIE | TMROIF | INTOIF | RBIF |

Bit 7 **GIE/GIEH**: Habilitador global de interrupciones.

Si IPEN = 0: (Bit 7 RCON)

1 = Habilita todas las interrupciones sin máscara

0 = Deshabilita todas las interrupciones

Si IPEN = 1: (Bit 7 RCON)

1 = Habilita todas las interrupciones de alta prioridad.

0 = Deshabilita todas las interrupciones.



Bit 6 **PEIE/GIEL**: Habilitador de Interrupciones de periféricos

Si IPEN = 0: (Bit 7 RCON)

1 = interrupciones de periféricos Habilita todas las sin máscara.

0 = Deshabilita todas las interrupciones de periféricos

Si IPEN = 1: (Bit 7 RCON)

1 = Habilita todas las interrupciones de periférico de baja prioridad.

0 = Deshabilita todas las periféricos de baja prioridad

Bit 5 **TMR0IE**: Interrupción de rebose de TMR0

1=habilita la interrupción por rebose de TMRO

0 = Deshabilita la interrupción por rebose de TMR0.

Bit 4 **INT0IE**: Interrupción Externa INT0

1 = Habilita la INT0

0 = Deshabilita la INT0

Bit 3 **RBIE**: Interrupción por cambio de estado en PORTB

1 = Habilita la interrupción por cambio de estado en PORTB

0 = Deshabilita la interrupción por cambio de estado en PORTB

Bit 2 **TMR0IF**: Bandera de Interrupción por rebose de TMR0

1 = El TMR0 se ha rebosado (se debe borrar por SW)

0 = TMR0 no se ha rebosado.

Bit **INT0IF**: Bandera de INT0

1 = Ha ocurrido INT0 (debe ser borrada por SW)

0 = INT0 no ha ocurrido.

Bit 0 **RBIF**: Bandera de interrupción por cambio de estado en PORTB.

1 = Ha ocurrido un cambio de estado en uno de los pinesRB4 a RB7

0 = No ha ocurrido un cambio de estado en RB4 a RB7.

Interrupciones. Registro **INTCON2**

| | | | | | | | |
|-------|---------|---------|---------|-----|--------|-----|-------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1 | U-0 | R/W-1 |
| RBPU | INTEDG0 | INTEDG1 | INTEDG2 | - | TMR01P | - | RBIP |

Bit 7 **RBPU**: Bit habilitador de las resistencias Pull-up.

1 = Todas las resistencias pull-ups deshabilitadas.

0 = Las resistencias pull-ups están habilitadas.

Bit 6 **INTEDG0**: Selección del flanco de la Interrupción Externa 0

1 = Flanco de subida.

0 = Flanco de bajada.



Bit 5 **INTEDG1**: Selección del flanco de la Interrupción Externa 1

1 = Flanco de subida.

0 = Flanco de bajada.

Bit 4 **INTEDG2** Selección del flanco de la Interrupción Externa 2

1 = Flanco de subida

0 = Flanco de bajada

Bit 3 Sin Implementar

Bit 2 **TMR0IP**: Bit de prioridad de la interrupción por rebose de TMR0

1 = Alta prioridad

0 = Baja prioridad

Bit 1 Sin implementar

Bit 0 **RBIP**: de prioridad de la interrupción de cambios de estado en PORTB

1 = Alta prioridad.

0 = Baja prioridad.

Interrupciones. Registro **INTCON3**

| R/W-1 | R/W-1 | U-0 | R/W-1 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|--------|--------|-----|--------|--------|-----|--------|--------|
| INT2IP | INT1IP | - | INT2IE | INT1IE | - | INT2IF | INT1IF |

Bit 7 **INT2IP**: Bit de prioridad de INT2

1 = Alta prioridad.

0 = Baja prioridad.

Bit **INT1IP**: Bit prioridad de INT1

1 = Alta prioridad.

0 = Baja prioridad.

Bit 5 Sin implementar

Bit 4 **INT2IE**: Bit habilitador de INT2

1 = Habilita INT2.

0 = Deshabilita INT2.

Bit 3 **INT1IE**: Bit habilitador de INT1

1 = Habilita INT1.

0 = Deshabilita INT1.

Bit 2 Sin implementar

Bit 1 **INT2IF**: Bandera que indica que ocurrió INT2

1 = Ha ocurrido INT2 (debe ser borrada por SW)

0 = No ha ocurrido INT2



Bit 0 **INT1IF**: Bandera que indica que ocurrió INT1
1 = Ha ocurrido INT1 (debe ser borrada por SW)
0 = No ha ocurrido INT1

TEMPORIZADOR TIMER 0

Este temporizador/contador tiene un registro de conteo de pulsos, que se denomina de la misma forma TMR0.

Es accesible desde código, se puede verificar su valor por programa, el conteo de este registro es incremental.

Cuando se desborda, se necesita recargar el valor de conteo, para evitar conteos erróneos.

Genera una interrupción cuando se desborda en modo 8 o bits16 bits.

El TIMER0 es controlado por el registro T0CON.

Interrupciones. Registro TOCOM

| | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|-------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| TMR0ON | TO8BIT | TOCS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 |

Bit 7 **TMR0ON**: Bit de control ON/OFF del Timer0

1 = Habilita el Timer0

0 = Detiene el Timer0

Bit 6 **TO8BIT**: Bit de selección de Timer0 en modo 8 bits o 16 bits

1 = Timer0 en modo 8 bits

0 = Timer0 en modo 16 bits

Bit 5 **TOCS**: Bit de selección de la fuente de reloj de Timer0.

1 = Transiciones en el pin T0CKI (RA4)

0 = Reloj interno de Instrucciones (FOSC/4)

Bit 4 **T0SE**: Bit de tipo de flanco para incremento de Timer0.

1 = Incremento flanco de bajada.

0 = Incremento por flanco de subida.

Bit 3 **PSA**: Bit de asignación del Pre escalador

1 = Pre escalador no asignado a Timer0.

0 = Pre escalador asignado a Timer0.



Bit 2-0 **T0PS2:T0PS0**: Bits de selección del pre escalador

111 = 1:256

110 1:128

101 = 1:64

100 = 1:32

011 = 1:16

010 = 1:8

001 = 1:4

000 = 1:2

Si no se asigna un preescalador a TMR0, el registro del reloj se incrementa cada ciclo de reloj. Además si el registro TMR0L es escrito, el conteo no se reanuda hasta dos ciclos de instrucción después.

La interrupción de TMR0, no puede sacar al procesador de un SLEEP, ya que en este caso el oscilador interno está parado.

REGISTROS ASOCIADOS A TIMER0

| | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|--------|----------|-----------|--------|--------|------|--------|--------|-------|
| TMROL | | | | | | | | |
| TMR0H | | | | | | | | |
| INTCON | GIE/GIEH | PEIE/GIEL | TMROIE | INT0IE | RBIE | TMROIF | INTOIF | RBIF |
| T0CON | TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 |
| TRISA | | | | | | | | |

El registro TMR0H se actualiza solo cuando se hace una lectura de TMR0L, esto con el fin de leer los 16 bit de TMR0. La parte alta de TMR0, solo se carga en el registro TMR0H cuando se hace una escritura en TMR0L, esto con el fin de transferir los 16 bits a la vez.

2.2.3 Librerías del Mikroc Pro.

Mikroc Pro 3.2 posee un amplio conjunto de librerías que facilitan la programación de los distintos Pics en lenguaje C. Algunas librerías usan variables que deben ser declaradas en el programa. Además dispone de Library Manager, ventana donde se puede visualizar todas las librerías disponibles, estas pueden ser incluidas en el proyecto seleccionándolas como se observa en la figura 2.13.

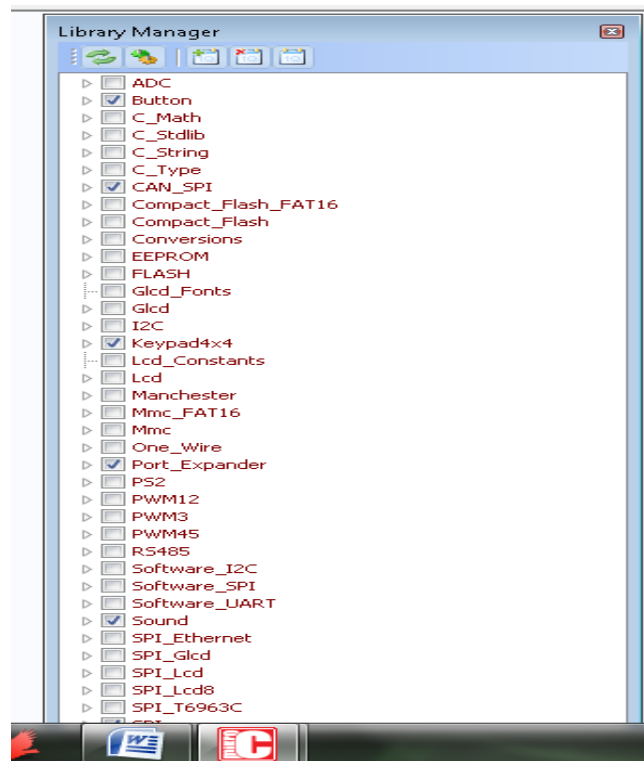


Figura 2.13. Library Manager

2.2.4 Interacción con el ISIS de Proteus.

Una vez que se tiene construido un circuito electrónico con Pics en ISIS de Proteus, y compilado el archivo hexadecimal en Mikoc Pro, es necesario realizar la simulación del esquema, para esto es necesario cargar el archivo hexadecimal en el pic correspondiente.

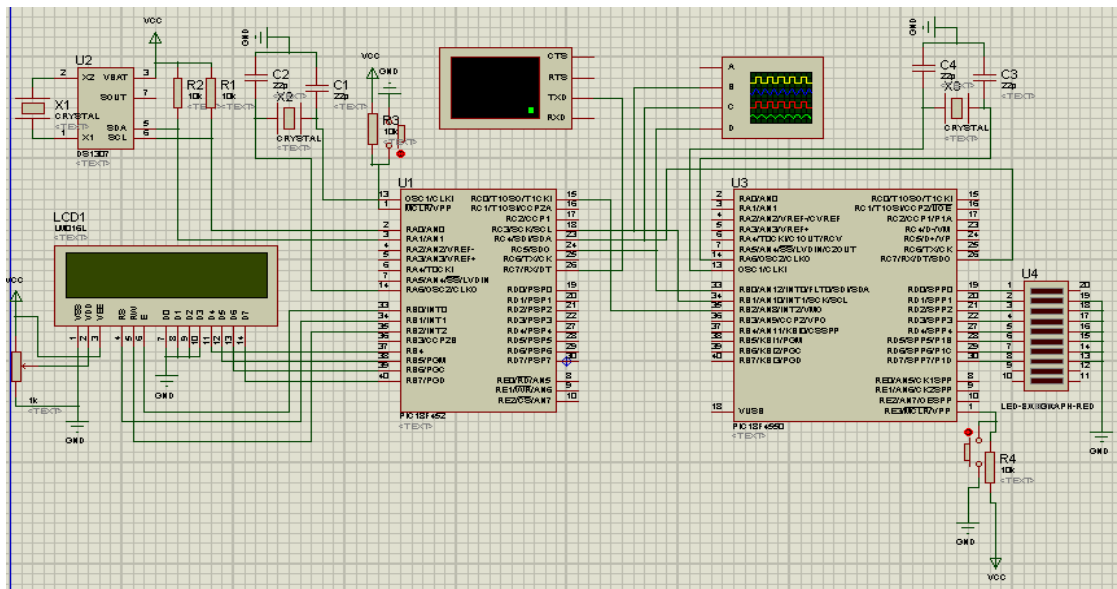


Figura 2.14. Esquema electrónico realizado en ISIS de Proteus

Para cargar el archivo hexadecimal es necesario dar doble-click sobre el pic, se visualiza la ventana de la figura 2.15.

En el casillero program file se coloca el directorio donde está el fichero hex por ejemplo **C:\misdocumentos\final1\final1.hex**, también se necesita colocar la frecuencia del cristal de cuarzo a la que estará conectado el pic, una vez que se ha llenado correctamente todos los casilleros de la ventana Edit Component se da doble-click en OK.

Por último se procede a simular el circuito.

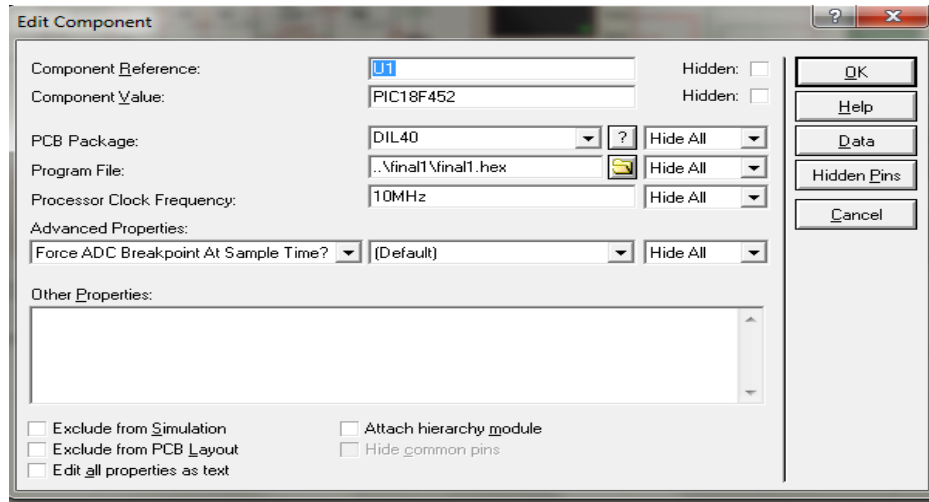


Figura 2.15. Editar componente

2.3 EAGLE

CadSoft EAGLE (Easily Aplicable Graphical Layout Editor) es un programa que permite desarrollar circuitos impresos. El programa consta de tres módulos:

- Diagramador,
- Editor de esquemas y
- Autorrouter.

2.3.1 Panel de control

El panel de control normalmente aparece después de iniciar EAGLE, este es el centro de control del programa. Todos los archivos de EAGLE se manejan aquí, y algunos ajustes básicos se pueden hacer en la configuración del programa.

La forma en la que está estructurado el panel de control proporciona una visión rápida y general de la librería, proyectos, entre otros elementos que forma parte del programa.

La primera vez que se le llama el panel de control aparecerá como se muestra la figura 2.16. Si se selecciona un objeto este aparecerá en forma de árbol, más información relevante se muestra en la parte derecha de la ventana relacionado al objeto seleccionado.

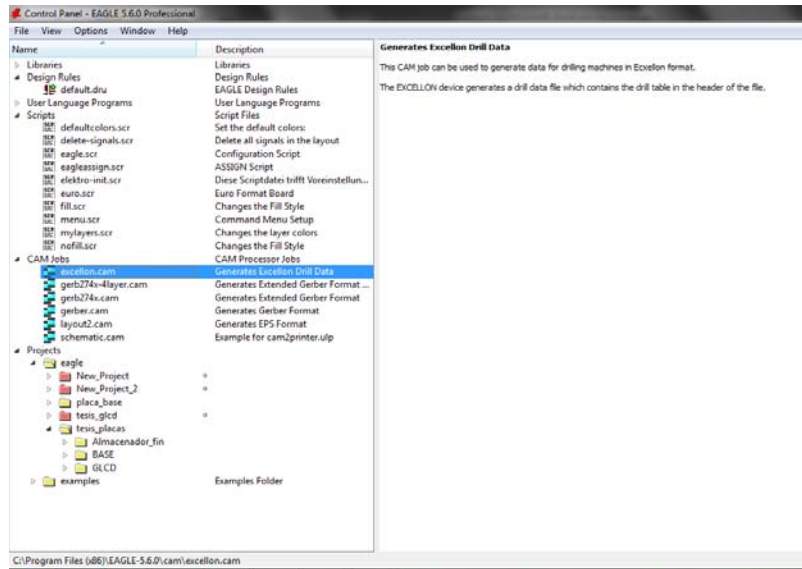


Figura 2.16. Vista del panel de control.

Biblioteca General (Library Summary)

La biblioteca de Eagle tiene la posibilidad de mostrar el contenido de las librerías para tener una visión muy rápida de los dispositivos disponibles.

Al hacer doble-click en library summary se abre en forma de árbol, se puede ver la colección de elementos disponibles para el diseño, en el campo descripción hay una breve detalle de cada elemento (figura 2.17).

El marcador verde atrás de la entrada de cada elemento de la biblioteca indica que está puede ser utilizado en el proyecto actual.

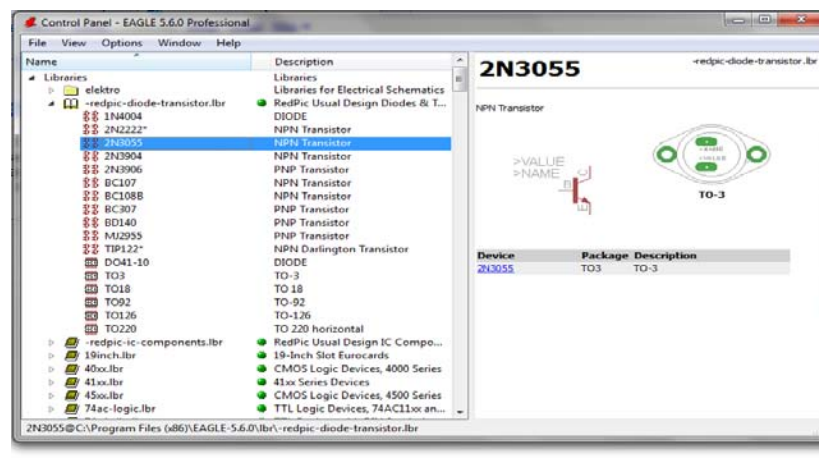


Figura 2.17. Library Summary

Reglas de diseño (Design Rules)

En Design rules se configura un conjunto de parámetros que regirá el proyecto actual.

La primera vez que se llama al programa la configuración para el diseño viene establecida por defecto. Si es necesario se ajusta los valores de acuerdo a las necesidades del proyecto. Cuando haya terminado los ajustes, se inicia la comprobación de errores haciendo clic en sobre el botón Aceptar.

El cuadro de diálogo **Reglas de diseño (Design Rules)** ofrece una gama de opciones diferentes que se pueden configurar a través de las pestañas. Las opciones incluyen:

File administra las reglas para el diseño.

Layers aquí se especifica el número de las láminas de cobre, la estructura de las capas.

Clearance distancia entre agujeros y las pistas, distancia entre pistas.

Distance Distancias desde el borde del tablero y entre agujeros.

Sizes Ancho mínimo de pista y diámetro mínimo del orificio.

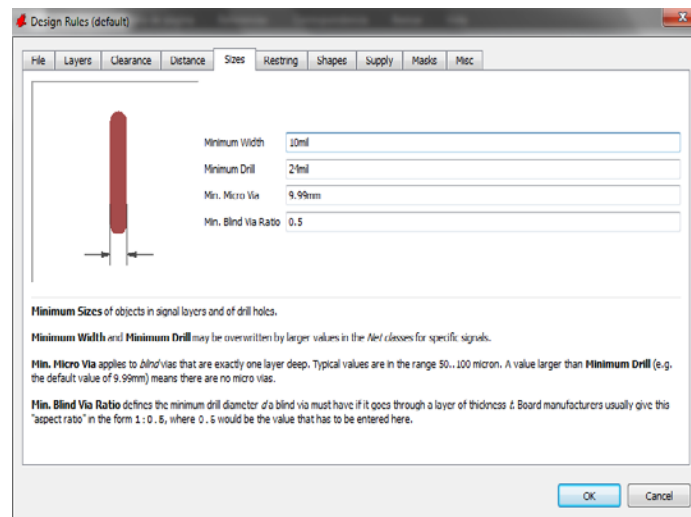


Figura 2.18 Ventana de Design Rules

User Language Programs, Scripts, CAM Jobs

En esta entrada se muestran los contenidos de los directorios *ulp*, *scr* y *cam*

Contiene varios programas de usuario en lenguaje (*. ULP), archivos de script (*. Scr).

Proyectos (Projects)

Los distintos proyectos se gestionan desde el Panel de control. Con un doble-click en Projects muestra el registro de proyectos en diferentes carpetas.

Un proyecto consiste generalmente en una carpeta (directorio) que representa el proyecto por su nombre.

La carpeta por lo general contiene todos los archivos que pertenecen al proyecto, por ejemplo, se encuentra un esquemático, los archivos de placa .brd, archivos de script entre otros.

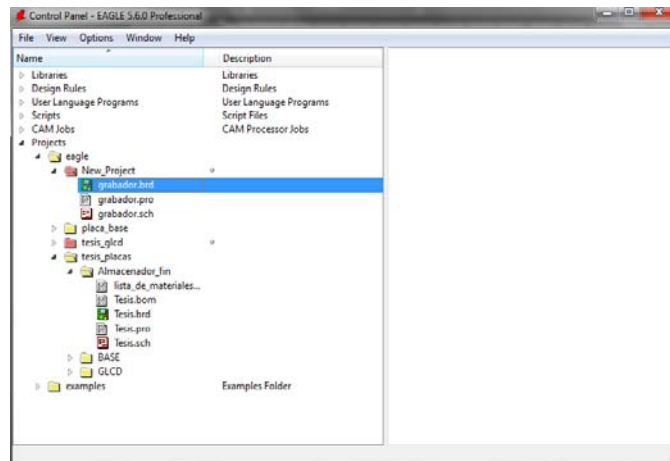


Figura 2.19 ventana de proyectos

Creación de un nuevo proyecto

Para la creación de un nuevo proyecto en el Panel de Control se selecciona:

Projects / Clic Derecho / New Folder

Se nombra a la nueva carpeta como **almacenador_fin**, la cual contendrá al proyecto almacenador_fin usado en este manual, el cual contendrá los archivos de esquema (.sch) y los archivos de placa (.brd). El proyecto actualmente activo se visualiza en color verde en el Panel de Control, como se muestra en la figura 2.20

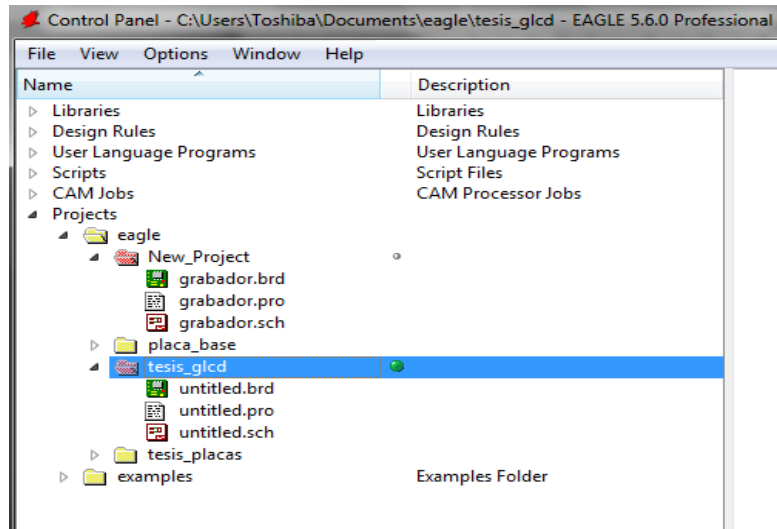


Figura 2.20

2.3.2 Esquemas

Creación de un esquema

Para la creación del esquema del proyecto se selecciona:

Clic Derecho en el proyecto / New / Schematic

A continuación aparece una ventana en blanco donde se puede empezar a crear el esquema.

Los comandos más utilizados para la creación de un esquema son:

Rejilla



La rejilla estándar para un esquema es de 2.54 mm (0.1 pulgadas). Los símbolos deberían ser colocados sobre esta rejilla o un múltiplo de ella, para facilitar la localización de todos los dispositivos (Guadilla y Gálvez 2000). En la figura 2.21 se muestra el cuadro con parámetros para este comando, los cuales se encuentran configurados en sus valores por defecto.

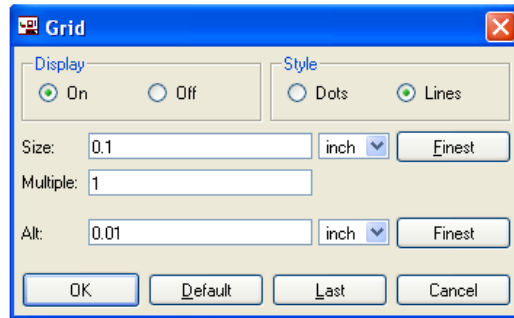


Figura 1.21 Cuadro de parámetros del comando Grid.

ADD

El comando ADD permite seleccionar símbolos de una librería. Al dar clic sobre él aparece una nueva ventana donde es posible introducir uno o más caracteres de búsqueda en el campo *Search*. Se puede usar el nombre de un dispositivo o cualquier palabra de la descripción del mismo. Por ejemplo, si se desea colocar el dispositivo tl084, se introduce en el campo de búsqueda *TL084* (Guadilla y Gálvez 2000).

El comando ADD permite agregar los componentes, así como también los símbolos para VCC, V +, y GND desde la librería *supply.lbr*.

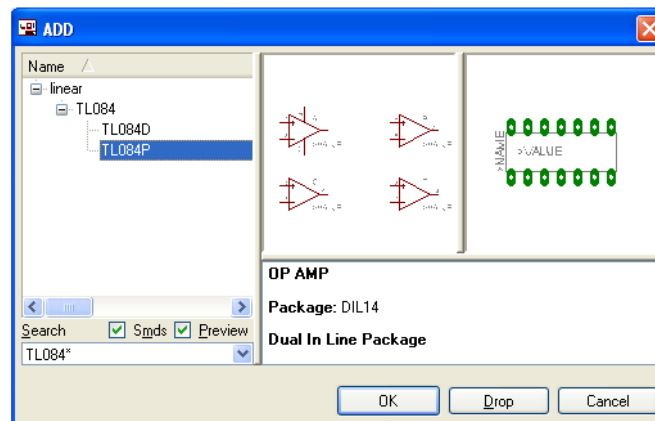


Figura 2.22 Ventana de búsqueda de dispositivos usando librerías.

MOVE

Una vez que se han colocado todos los elementos se puede empezar a trasladarlos a una posición apropiada mediante la selección de este comando en

la barra de herramientas, luego se sitúa el cursor en el elemento que se quiere mover y se lo ubica en un lugar apropiado.

Mientras esta activo este comando, se tiene la opción de girar el elemento.

(Línea de conexión)

Este comando permite conectar entre sí los diferentes elementos de un esquema, como se observa en la figura 2.23.

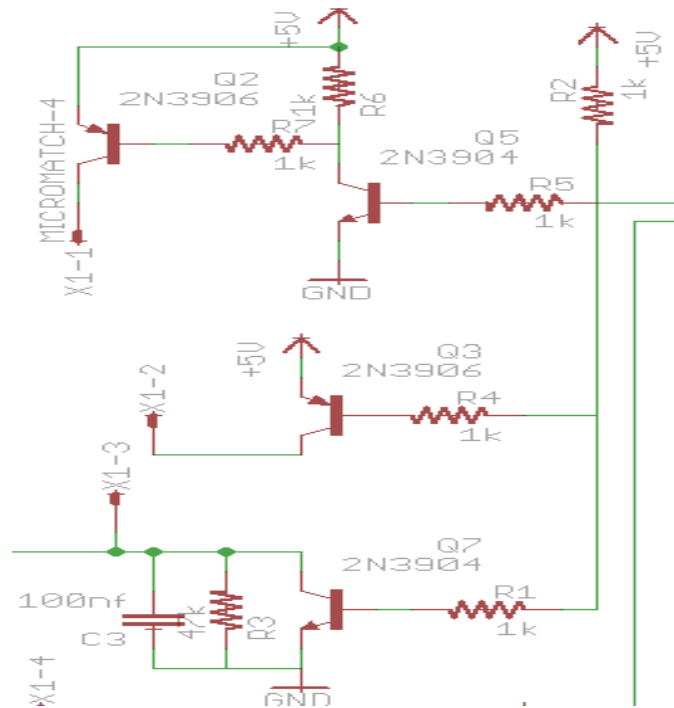


Figura 2.23. Comando Net para conexiónado entre elementos.

JUNCTION

El arrastre de una línea de conexión sobre otra línea genera una conexión entre ellas. La conexión se representa por un nudo (*junction*), que se coloca automáticamente (Guadilla y Gálvez 2000).

SHOW

Resalta el objeto seleccionado. En la figura 2.24 se resalta en verde fosforescente los cables conectados al nodo así como cada uno de los terminales conectados.

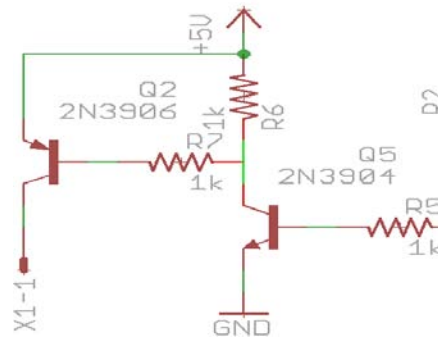


Figura 2.24 Comando Show para visualización de conexiones.

NAME

Con este comando se le puede dar un nombre a los componentes, redes, o los buses. En la figura 2.25 se observa el uso de este comando para dar nombre a una resistencia.

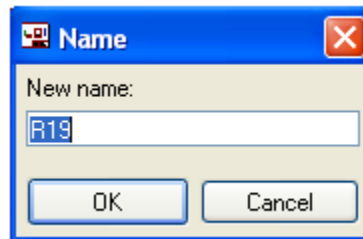


Figura 2.25 Comando Name.

VALUE

EAGLE permite dar los valores a las resistencias, condensadores, e inductancias; en el caso de los circuitos integrados normalmente recibe el tipo (por ejemplo, 74LS00N) como su valor. En la figura 2.26 se observa el uso de este comando para colocar un valor a una resistencia.

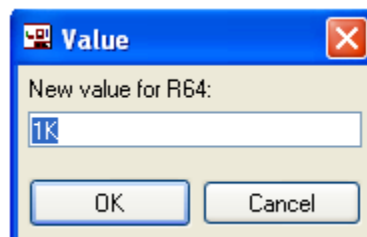


Figura 2.26 Comando Value.

INVOKE

INVOKE permite visualizar la alimentación de circuitos integrados que se encuentran en un esquema.

2.3.3 PCB

Una vez que se tiene dibujado el esquema se desea diseñar una placa, se da doble-click sobre el icono Board en la barra de herramientas.



Figura 2.27. Icono Board

Se genera un archivo .brd con el mismo nombre que el esquema correspondiente y con todos los dispositivos son colocados al lado izquierdo y superior de la placa, como se ve en la figura 2.28

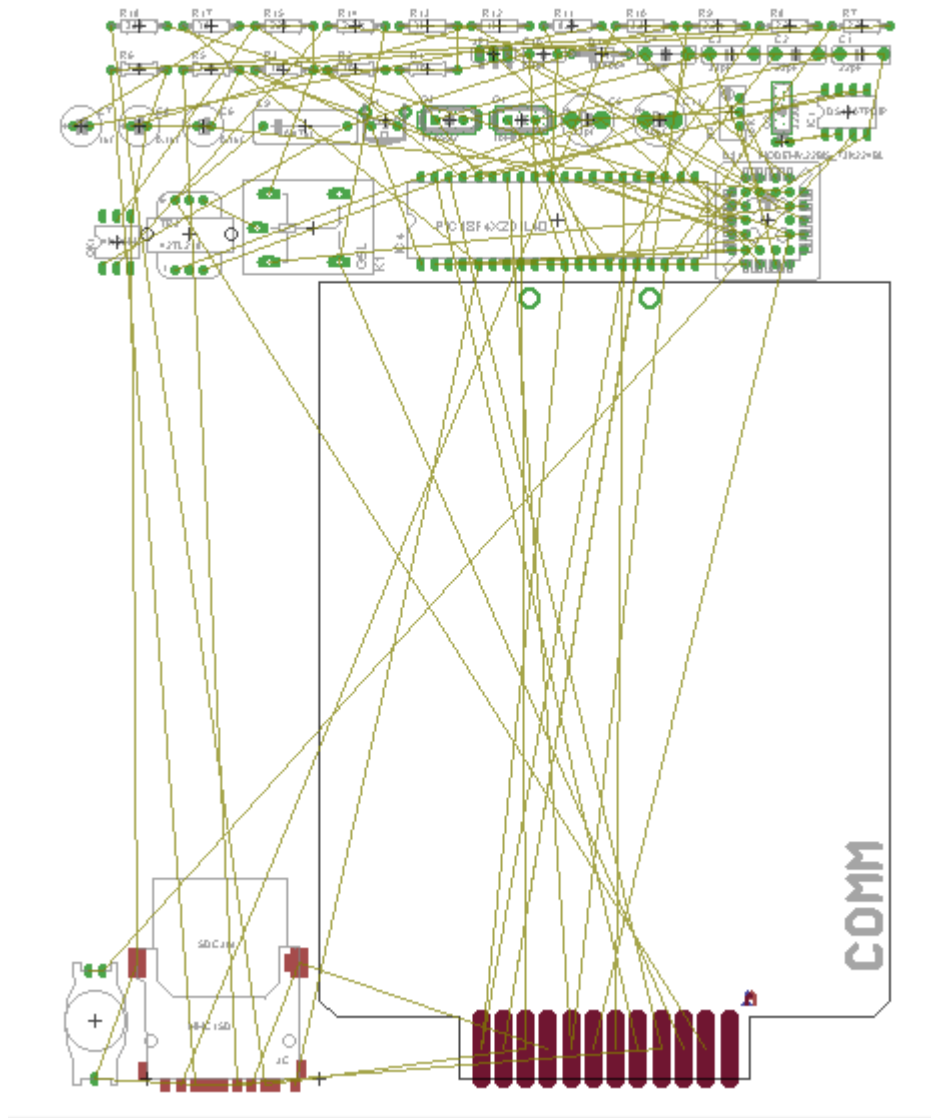


Figura 2.28 Creación de un archivo board

Se deben mover todos los componentes dentro del área asignada con el comando Move, como se muestra en la figura 2.29, para después realizar el ruteado.



CAPITULO 3: DISEÑO DE TARJETAS

3.1 Sistema de almacenamiento de datos y temporización

3.1.1 SD/MMC con SPI, uso de librerías de Mikroc Pro 3.2

3.1.1.1 Protocolo de comunicación SPI

El protocolo SPI (*Serial Peripheral Interface*) es un protocolo de comunicaciones, empleado para la transferencia de información entre circuitos integrados en equipos electrónicos.

SPI es un bus de 4 líneas para interconectar dispositivos periféricos de baja y media velocidad. Cada una de estas líneas lleva información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede trabajar como transmisor y receptor, este tipo de comunicación serie es full dúplex.

Los dispositivos conectados al bus son definidos como maestros y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control.

Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada Chip Select, por lo tanto un esclavo es activado solo cuando esta línea es seleccionada con un cero lógico. Generalmente una línea de selección es dedicada para cada esclavo, en la figura 3.1 se observa varios dispositivos comunicados a través del bus SPI donde se puede apreciar que existe una línea de chip select dedicada para la selección de cada esclavo.

Dependiendo del número de esclavos presentes en el bus, se requerirán más o menos líneas de selección (chip select), cada una de ellas será nombrada como SS1, SS2, SS3, etc. Este medio de selección permite conectar varios esclavos de una forma simple.

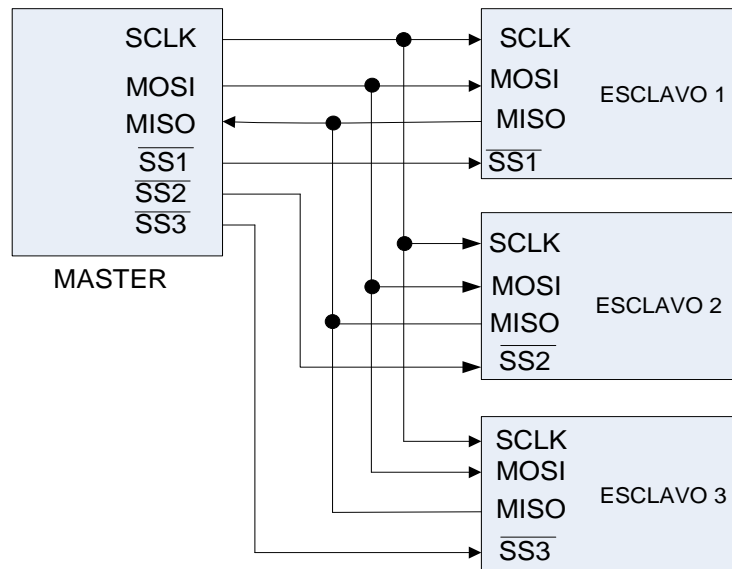


Figura 3.1 Bus SPI: un maestro y tres esclavos.

Los nombres de las señales son definidos dentro del protocolo como:

- **SCLK** reloj del bus,
- **MOSI** (Master Output Slave Input) salida de datos del maestro y entrada de datos a los esclavos,
- **MISO** (Master Input Slave Output) salida de datos de los esclavos y entrada de datos al maestro y
- **SS** (Slave Select) habilitación del esclavo por parte del maestro.

3.1.1.2 Comunicación entre periféricos por SPI

El dispositivo que hace las funciones de maestro, genera una señal de reloj (SCKL) y la envía a los dispositivos que están como esclavos. La línea de selección, chip select (SS), se utiliza para indicar con que esclavo se está tratando comunicar el dispositivo maestro.

En la figura 3.1 todas las señales excepto la de selección de esclavo (SS) son comunes a todos los esclavos, por eso el maestro debe indicar cuál de los esclavos será activado durante la comunicación.

En SPI se tiene que configurar dos bit para que se pueda realizar la transmisión de información, estos bits son CPOL (Clock polaity = polaridad del reloj) y CPHA

(Clock phase = reloj de fase). El bit CPOL determina si el estado IDLE de la línea de reloj se encuentra en alto (CPOL = 1), o si se halla en bajo (CPOL = 0).

CHPA, determina si la información es detectada en los flancos de subida o bajada del reloj que genera el dispositivo maestro, si CPHA=0 los datos sobre la línea MOSI son detectados en cada flanco de bajada del reloj.

Cada bit sea este CPOL y CHPA tiene dos posibles combinaciones (0,1) lo que permite que el reloj se pueda configurar de cuatro maneras diferentes, las cuales son incompatibles la una de la otra. Por lo tanto si dos a más dispositivos se quieren comunicar por SPI estos deben tener la misma polaridad de reloj, y la misma fase.

Existen cuatro modos de configurar el reloj para la comunicación SPI:

- Modo 1 (CPOL = 0, CPHA = 0)
- Modo 2 (CPOL = 0, CPHA = 1)
- Modo 3 (CPOL = 1, CPHA = 0)
- Modo 4 (CPOL = 1, CPHA = 1)

Modo 1

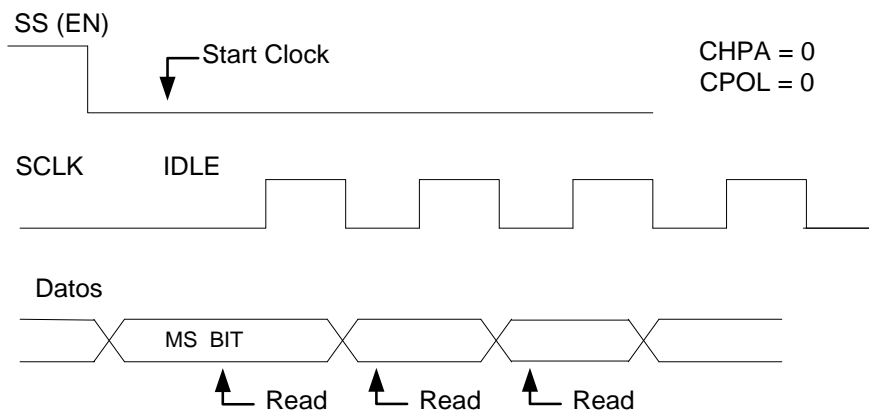


Figura 3.2 Modo 1 (CPOL = 0, CPHA = 0)

Modo 2

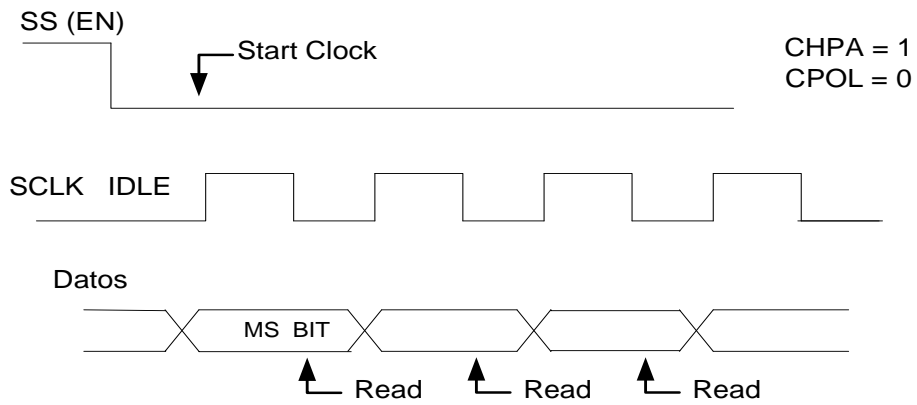


Figura 3.3 Modo 2 (CPOL = 0, CPHA = 1)

Modo 3

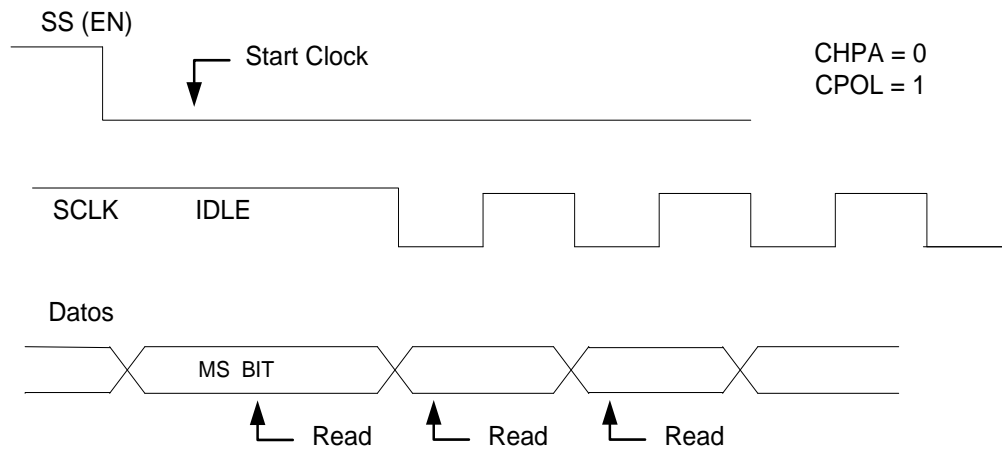


Figura 3.4 Modo 3 (CPOL = 1, CPHA = 0)

Modo 4

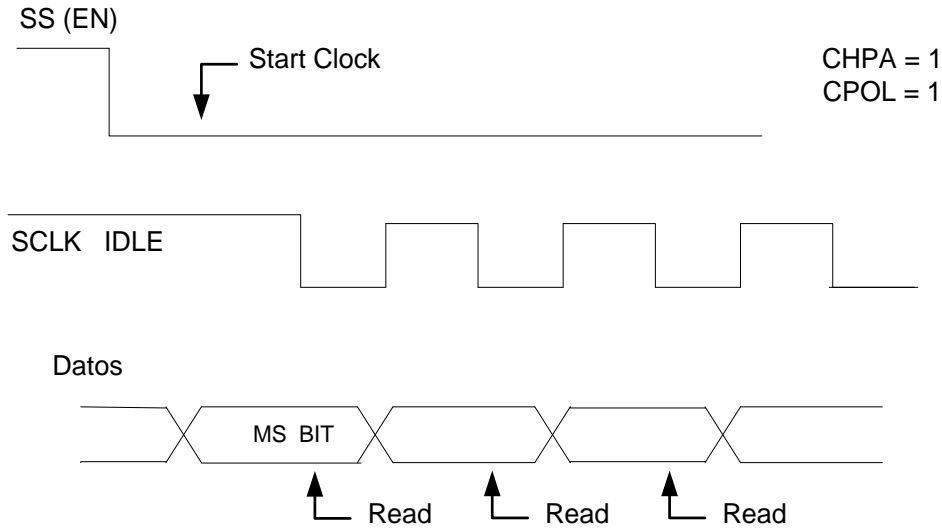


Figura 3.5 Modo 4 (CPOL = 1, CPHA =1)

3.1.1.3 Tarjeta SD

Las tarjetas SD son dispositivos de memoria flash diseñada para proporcionar gran capacidad de almacenamiento de datos, además de tener la propiedad de ser no volátil. La capacidad de almacenamiento de las tarjetas SD no cesa de aumentar con el transcurrir del tiempo, en la actualidad están disponibles en capacidades de 256MB hasta los 8GB. Las tarjetas SD vienen en varios tamaños MiniSD, MicroSD, SDcard en la tabla 3.1 se enumeran los principales especificaciones de las tarjetas SDcard y MiniSD.

| | SDcard estándar | Mini SD |
|-----------------------------|--------------------|--------------------|
| Dimensiones | 32 x 24 x 2.1mm | 21.5 x 20 x 1.4 mm |
| Peso de la tarjeta | 2 gramos | 1 gramo |
| Voltaje de operación | 2.7 – 3.6 voltios | 2.7 – 3.6 voltios |
| Protección contra escritura | si | no |
| Pines | 9 | 11 |
| Interfaz de comunicación | SD o SPI | SD o SPI |
| Consumo de corriente | <75 mA (escritura) | <40 mA (escritura) |

Tabla 3.1 Especificaciones de las memorias SDcard y MiniSD

Las memorias MiniSD, MicroSD son tarjetas eléctricamente compatibles con las tarjetas SDcard estándar, se pueden insertar en adaptadores especiales para utilizarlas como tarjetas SDcard estándar en las ranuras para tarjetas estándar.

Para la realización del sismógrafo digital de tres componentes se utiliza una tarjeta SDcard y MicroSD. Las especificaciones para las tarjetas SD más pequeñas (mini y micro SD) son las mismas y no se describen con más detalle.



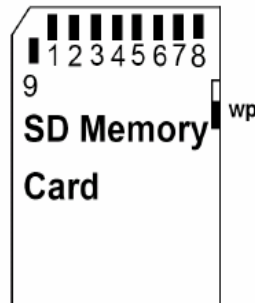
SDcard estándar

Mini SD MicroSD con adaptador

Figura 3.6 Tipos de Memorias SD

Las tarjetas SDcard, MiniSD y MicroSD se puede conectar con Pics usando dos protocolos de comunicación diferente: protocolo SDcard y el SPI (Serial Peripheral Interface). El protocolo SPI, siendo el más utilizado, para la realización del sismógrafo se utiliza el protocolo SPI.

La tarjeta SD estándar tiene 9 pines con la disposición, como se muestra en la Figura 3.7. Los pines tienen funciones diferentes dependiendo de la interfaz del protocolo. La Tabla 3.2 indica función de cada pin, para el modo de comunicación en SPI.

**Figura 3.7** Pines de la tarjeta SDcard

| pin | Nombre | Descripción |
|-----|--------|---|
| 1 | /CS | Chip Select (activo en 0) |
| 2 | DI | Entrada de datos y comandos hacia la memoria |
| 3 | VSS | GND |
| 4 | VCC | Alimentación de la tarjeta 2.7 – 3.6 V |
| 5 | SCLK | Reloj |
| 6 | VSS | GND |
| 7 | DO | Salida de datos y estado de la tarjeta hacia el Pic |
| 8 | RSV | Reservado (pin no usado en SPI) |
| 9 | RSV | Reservado (pin no usado en SPI) |

Tabla 3.2 Líneas de la interfaz SPI en una SDcard

/CS permite al Pic18f452 seleccionar la SDcard, así cuando !CS vale 0 la tarjeta se encuentra seleccionada y lista para operar. Este pin puede ser controlado por cualquier pin de salida del controlador.

DataIn (DI) es la entrada de datos serie a la memoria y debe estar conectada a la salida MOSI de la interfaz SPI del Pic18f452.

DataOut (DO) es la salida de datos serie de la memoria y debe estar conectada al pin MISO de la interfaz SPI del Pic18f452.

CLCK (SCLK) es el pin para la entrada de la señal de reloj generada por el Pic18f452, es la que marca el ritmo de transferencia de la información serie entre ambos.

VDD es el pin de alimentación, la tensión debe encontrarse en el rango de los 2,6 y 3,7V, si se sobrepasa de este valor la memoria se puede quemar.

VSS1 y **VSS2** son los pines de más a tierra (GND) de la tarjeta.



3.1.1.4 Funcionamiento de la memoria en modo spi

Registros del PIC18F452 en modo SPI.

Existen 4 registros básicos del módulo SPI. Dichos registros sirven para el establecimiento y control de la comunicación.

SSPSTAT Registro de estado.

| | | | | | | | |
|-------|-------|--------------|-----|-----|------------------|-----|-------|
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| SMP | CKE | D/ \bar{A} | P | S | $\overline{R/W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

bit 7 **SMP**: Bit de muestreo

SPI modo maestro:

1 = los datos de entrada son muestreados al final de tiempo de los datos de Salida.

0 = los datos de entrada son muestreados en la mitad de tiempo de los datos de salida.

SPI modo esclavo:

SMP debe ser borrado cuando se trabaja en el modo esclavo.

bit 6 **CKE**: selector del flanco de reloj en SPI

CKP = 0:

1 = Dato transmitido en el flanco ascendente del SCK

0 = Dato transmitido en el flanco descendente del SCK

CKP = 1:

1 = Dato transmitido en el flanco descendente del SCK

0 = Dato transmitido en el flanco ascendente del SCK

bit 5 **D/A**: Bit de dirección/datos.

Usado solo cuando se trabaja en el modo I2C.

bit 4 **P**: bit de parada.

Usado solo cuando se trabaja en el modo I2C.

bit 3 **S**: bit de inicio.

Usado solo cuando se trabaja en el modo I2C.

bit 2 **R/W**: bit de información de lectura/escritura.

Usado solo cuando se trabaja en el modo I2C.

bit 1 **UA**: bit de actualización de dirección.

Usado solo cuando se trabaja en el modo I2C.



bit 0 **BF**: bit de estado del buffer (solo en el modo de recepción)

1 = Recepción completa, SSPBUF lleno.

0 = Recepción incompleta, SSPBUF vacío.

SSPCON1. Registro de control.

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **WCOL**: bit detector de colisión(solo en el modo transmisor)

1 = el registro SSPBUF es escrito mientras está en curso otra transmisión (debe borrarse por software)

0 = No hay colisión

bit 6 **SSPOV**: bit indicador de sobreflujo (solo en el modo recepción)

SPI Modo esclavo:

1 = un nuevo byte es recibido mientras el registro SSPBUF está guardando el dato anterior.

0 = No hay sobreflujo.

bit 5 **SSPEN**: bit habilitador del Puerto serial síncrono

1 = habilitación el Puerto serial y configuración de SCK, SDO, SDI, y SS Comopins del puerto serie.

0 = deshabilitación del Puerto serial y configuración de sus respectivos pines como puertos I/O.

bit 4 **CKP**: bit selector de polaridad del reloj

1 = el estado IDLE para el reloj es un nivel alto

0 = el estado IDLE para el reloj es un nivel bajo

bit 3-0 **SSPM3:SSPM0**: bits de selección de modo del Puerto serial

0101 = SPI modo esclavo, clock = SCK, pin SS deshabilitado.

0100 = SPI modo esclavo, clock = SCK, pin SS habilitado.

0011 = SPI modo maestro, clock = TMR2 output/2

0010 = SPI modo maestro, clock = FOSC/64

0001 = SPI modo maestro, clock = FOSC/16

0000 = SPI modo maestro, clock = FOSC/4

SSPBUF. Buffer de transmisión/recepción serie.



Es un registro de almacenamiento temporal, en el que se carga la información tanto para la recepción como para la transmisión.

SSPSR. Registro de desplazamiento.

Es un registro auxiliar utilizado por el pic para el desplazamiento bit a bit en la transmisión/recepción de datos.

Maestro(PIC18F452)

Primero configuramos al PIC como maestro. Esto se realiza seteando los respectivos bits **SSPM3:SSPM0** del registro **SSPCON**. Lo cual ya no será necesario, porque las librerías provistas por el compilador mikroC Pro automáticamente cargan estos registros, cuando se hace el llamado a la función de inicialización del modo SPI. Solo en caso de utilizar lenguaje ensamblador deberemos recurrir a la configuración bit por bit.

Luego se elige la frecuencia de la señal de reloj(SCLK). Esta debe ser menor o igual a la máxima soportada por el dispositivo esclavo. Siendo 25MHz para las memorias estándar.

Por ultimo indicar a la tarjeta SD que se quiere establecer la comunicación. Para esto colocamos un nivel lógico en la línea del chip select (CS=0). Garantizando así la respuesta de un solo esclavo conectado al bus SPI.

Esclavo (memoria SD)

Cabe destacar que para las tarjetas SD no es necesario ninguna configuración adicional, ya que por defecto vienen para trabajar como esclavos. Teniendo en cuenta eso sí, la máxima tasa de transferencia que estas pueden soportar, e indicándole que se va a utilizar el protocolo SPI.

Transmisión

La transmisión inicia en el momento que se carga el registro SSPBUF, luego el PIC pasa este dato al SSPSR, el cual envía un bit por cada ciclo de reloj a través de la línea MOSI. Paralelamente el esclavo también utiliza el registro de desplazamiento para colocar los datos en la línea MISO por cada ciclo de reloj que

el PIC le envía. Esta secuencia sigue hasta que se haya completado el tamaño de la palabra que en este caso es de 8 bits.

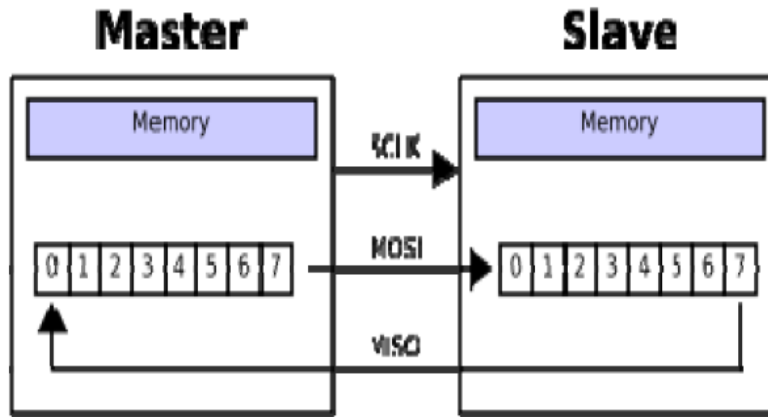


Figura 3.8 Bits de transmisión por SPI

3.1.1.4.1 Comunicación SPI con la tarjeta de memoria

Para establecer la comunicación con la memoria SD mediante SPI, existe un grupo de comandos hexadecimales normalizados por el protocolo que el dispositivo maestro debe enviar a la tarjeta dependiendo de los requerimientos del usuario.

Por cada comando o instrucción que el maestro envíe por la línea MOSI, la SD coloca un byte de respuesta en la línea MISO. Luego de cada comando enviado el PIC debe verificar que la respuesta sea la indicada antes de seguir con el flujo del programa.

| Byte 1 | | | Byte 2-5 | Byte 6 | |
|------------|--------|---------|-----------------|--------|---|
| Num bit : | 7 6 | 5-0 | 31-0 | 7 | 0 |
| valor bit: | 0 1 | command | Commandargument | CRC | 1 |

Tabla 3.3. Estructura de un comando

Por ejemplo el comando cero(CMD0) seria 01000000.Y la estructura la cadena total 0x40,0x00,0x00,0x00,0x00,0x00. Normalmente las cadenas de comandos en modo SPI no llevan argumento por lo que valen cero, tampoco byte de verificación a menos que sea estrictamente necesario.

A cada uno de estos comandos le corresponde un byte de respuesta que serán R1 o R2. Estos bytes de respuesta son colocados en la línea MISO cada vez que la SD detecte un comando valido en la MOSI.

| Bit | |
|-----|-----------------|
| 0 | Idle state |
| 1 | Erase Reset |
| 2 | Illegal Command |
| 3 | Com CRC Error |
| 4 | Erase_Seq_Error |
| 5 | Address Error |
| 6 | Parameter Error |
| 7 | 0 |

Tabla 3.4. Estructura del Byte de la respuesta R1.
byte es el mismo que R1).

| Bit | |
|-----|-----------------|
| 0 | 0 |
| 1 | WP Erase Ship |
| 2 | Error |
| 3 | CC Error |
| 4 | Card ECC Failed |
| 5 | WpViolation |
| 6 | Erase Parameter |
| 7 | Out of Range |

Tabla 3.5 estructura del primer byte de la respuesta R2. (El segundo

A continuación se explica el propósito de cada uno de estos comandos así como sus respectivas respuestas.

| Comando: | Argumentos: | Respuesta: | Descripción: |
|----------|------------------------------|------------|---|
| CMD0 | No | R1 | Resetea la tarjeta |
| CMD1 | No | R1 | Inicializa la tarjeta |
| CMD9 | No | R1 | Pide a la tarjeta su información CSD |
| CMD10 | No | R1 | Pide a la tarjeta su identificación CID |
| CMD13 | No | R2 | Consulta el estado de la tarjeta |
| CMD16 | [31..0] Longitud del bloque. | R1 | Establece la longitud (en bytes) del bloque para los datos en las operaciones de lectura y escritura. |
| CMD17 | [31..0] Dirección de datos. | R1 | Lee un bloque del tamaño indicado por el comando 16. |
| CMD24 | [31..0] Dirección de datos | R1 R1 R1 | Escribe un bloque del tamaño indicado por el comando 16. |
| | | | |

Tabla 3.6. Lista de comandos.

En lenguaje de bajo nivel estos comandos son esenciales para el manejo de las memorias SD. Pero en lenguajes más avanzados como el C, el mismo que utiliza **mikroCpro**, todos estos comandos vienen ya implementados dentro de librerías para el manejo de este tipo de dispositivos. Disminuyendo considerablemente el tiempo de programación, y basta con solo llamar a determinada función para realizar una operación de lectura o escritura. Mientras que en bajo nivel todo esto se lo realiza comando por comando y verificación tras verificación dando como resultado periodos muy largos y tediosos de programación. Además de la adquisición de software auxiliar (por ejemplo el WinHex) para poder visualizar los datos grabados en las tarjetas desde cualesquier PC.

Pero como en todo existe sus pros y sus contras, cuando se utiliza lenguaje asm se tiene control total del flujo del programa además de los tiempos de ejecución de tal o cual subrutina. Por otro lado con el uso de las librerías no se puede alterar ninguna instrucción dentro de las funciones integradas a ellas. Esto conlleva a un retardo considerable en la compilación y en la depuración del programa, porque siempre se cargan un conjunto de funciones muchas veces innecesarias.

El compilador MikroC Pro proporciona dos tipos de librerías para el manejo de tarjetas de memoria; una sirve para lectura y escritura por sectores y la otra para el manejo de ficheros en FAT16 que es el que se utilizara en este proyecto, para el almacenamiento de datos provenientes del ADC(PIC18F2553). Esto se explicara a detalle más adelante.

Por el momento basta con saber cómo se realiza la comunicación SPI y que existen los comandos para el acceso a las memorias SD, recalcando que estos no se implementaran en ninguna etapa del proyecto, debido al tiempo que esto conlleva. Optando por utilizar las librerías que proporciona MikroC Pro.

3.1.1.4.2 Uso de las librerías del módulo SPI del PIC18F452

MikroC Pro proporciona dos librerías para trabajar con la comunicación SPI. La una es implementada por software, mientras que la otra viene solo para los PICs que tienen el módulo SPI integrado físicamente (Hardware) es el caso del PIC18F452 y PIC18F2553. A continuación se explican las funciones básicas de la librería.

Tabla 3.7. *SPI1_Init_Advanced*:

| Función | SPI1_Init_Advanced |
|-------------|---|
| Estructura | SPI1_Init_Advanced(modo de trabajo SPI, intervalo de muestreo de dato, estado inactivo del CLK, eje de transmisión); |
| Descripción | Configura e inicializa el módulo SPI. Y debe ser llamada primero antes de utilizar otra función de la librería. |

Los parámetros pueden tomar estos valores:

| Descripción | Constante de la librería predefinida |
|--------------------------------------|--------------------------------------|
| Modo de trabajo SPI | |
| Reloj del Master= Fosc/4 | _SPI_MASTER_OSC_DIV4 |
| Reloj del Master= Fosc/16 | _SPI_MASTER_OSC_DIV16 |
| Reloj del Master= Fosc/64 | _SPI_MASTER_OSC_DIV64 |
| Fuente de Reloj del Master TMR2 | _SPI_MASTER_TMR2 |
| Selección de esclavo habilitado | _SPI_SLAVE_SS_ENABLE |
| Selección de esclavo deshabilitado | SPI_SLAVE_SS_DIS |
| Intervalo de Muestreo de Dato | |

| | |
|--|--------------------------------------|
| El dato de entrada es muestreado en la mitad del intervalo | <code>_SPI_DATA_SAMPLE_MIDDLE</code> |
| El dato de entrada es muestreado al final del intervalo | <code>_SPI_DATA_SAMPLE_END</code> |
| Estado Inactivo del Reloj SPI | |
| Estado Inactivo del Reloj: Alto(HIGH) | <code>_SPI_CLK_IDLE_HIGH</code> |
| Estado Inactivo del Reloj: Bajo(LOW) | <code>_SPI_CLK_IDLE_LOW</code> |
| Eje de Transmisión | |
| Dato transmitido en el eje de transición bajo a alto | <code>_SPI_LOW_2_HIGH</code> |
| Dato transmitido en el eje de transición alto a bajo | <code>_SPI_LOW_2_LOW</code> |

Tabla 3.8. *SPI1_Read*:

| Función | <code>SPI1_Read</code> |
|--------------------|--|
| Estructura | <code>SPI1_Read(unsigned short buffer);</code> |
| Descripción | Lee un byte del bus SPI, y al mismo tiempo coloca la variable buffer sobre la línea de salida del bus. |

Tabla 3.9. *SPI1_Write*:

| Función | <code>SPI1_Write</code> |
|--------------------|---|
| Estructura | <code>SPI1_Write(unsigned short data);</code> |
| Descripción | Escribe el byte data, en el bus SPI |

3.1.1.5 USO DE LIBRERÍAS DE MIKROC PRO 3.2 PARA LA SDCARD

3.1.1.5.1 Tabla de asignación de archivos(FAT16)

Sector: bloque compuesto por bytes de datos consecutivos. Su tamaño típicamente es de 512bytes.

Cluster: contiene varios sectores de 512 bytes cada uno. Su tamaño es de 32 Kbytes para memorias de hasta 4GB.



El sistema de archivos *FAT* se compone de cuatro secciones:

1. Sector de arranque. Es el primer sector de la partición y contiene información básica: como las direcciones de punteros de las distintas secciones y la del sector de arranque del sistema operativo.
2. Región FAT. Contiene dos copias de la tabla de asignación de archivos (por motivos de seguridad). Estos son mapas de la partición, indicando qué clusters están ocupados por los archivos.
3. Directorio raíz. Es el índice principal de carpetas y archivos.
4. Región de datos. Es el lugar donde se almacena el contenido de archivos y carpetas. Por tanto, ocupa casi toda la partición. El tamaño de cualquier archivo o carpeta puede ser ampliado siempre que queden suficientes clusters libres. Cada cluster está enlazado con el siguiente mediante un puntero. Si un determinado cluster no se ocupa por completo, su espacio remanente se desperdicia.

Cada archivo consta de varios clusters, cuyo número depende de su tamaño. Formándose una cadena enlazada de clusters que no necesariamente es continua.

La tabla de asignación de archivos tiene una lista de entradas con información acerca de los distintos clusters dentro de un archivo determinado. El tamaño depende del sistema en uso en este caso FAT16 (16 bits). Y los datos que contienen es:

- La dirección del siguiente clúster en la cadena.
- Si es pertinente, la indicación de "fin de archivo" (que es también el fin de la cadena).
- Un carácter especial para indicar que el clúster es defectuoso.
- Un carácter especial para indicar que el clúster está reservado (es decir, ocupado por un archivo).
- El número cero para indicar que el clúster está libre (puede ser usado por un archivo).

A continuación se muestra como se distribuye el espacio de la memoria para el manejo de archivos en FAT16.

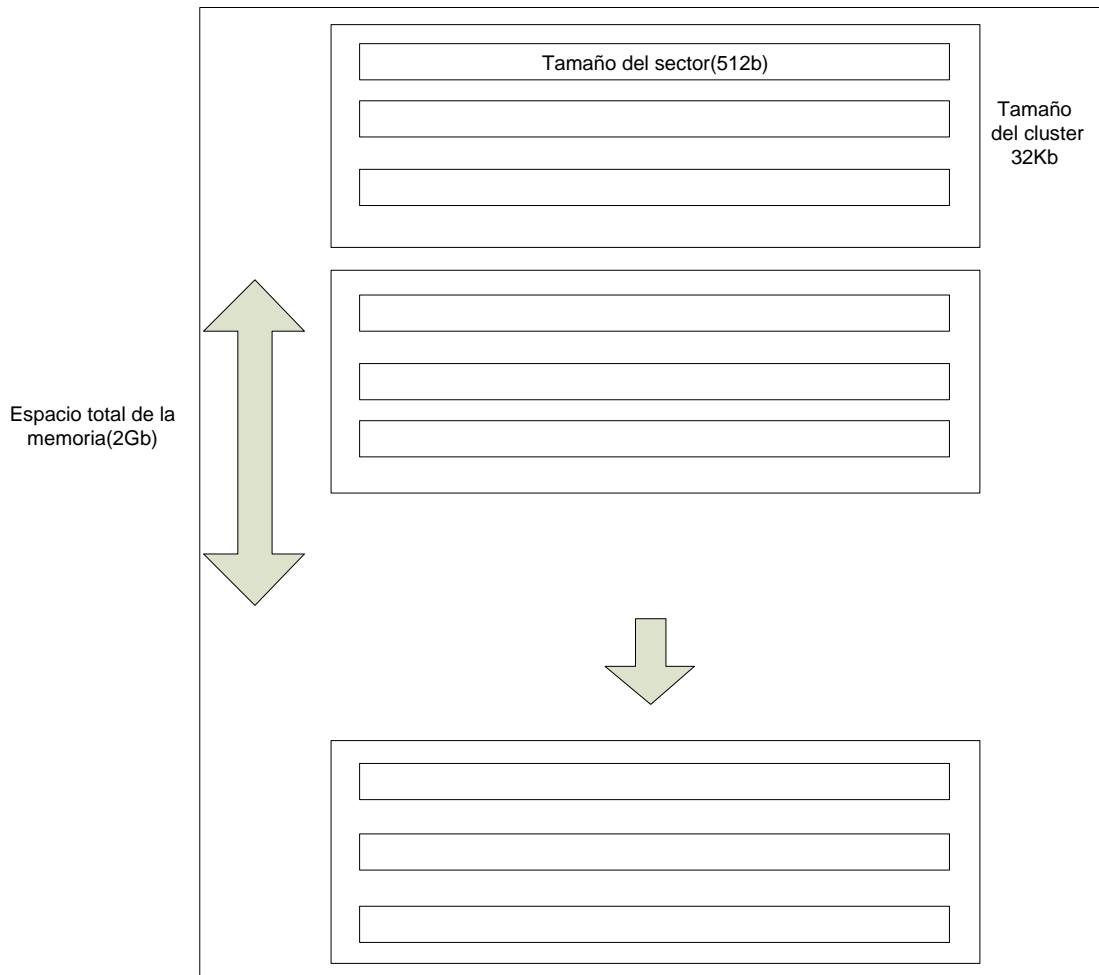


Fig.3.9 Distribución de clusters en FAT16.

3.1.1.5.2 Funciones para trabajar con FAT16

El uso de las librerías tiene sus ventajas en lo que se refiere al tiempo que lleva la implementación de un programa en C para manejo de memorias SD.

Dentro del paquete de librerías que tiene integrado MikroC pro v3.2 existe la **Multi Media Card Library** que contiene todas las funciones necesarias para el manejo de archivos en formato FAT16 (File Allocation Table). Este formato es muy utilizado para el almacenamiento de datos por muchos dispositivos, es el caso concreto de las tarjetas SD. Lo que lo hace que la información contenida sea legible desde cualesquier PC o laptop.

La librería **Multi Media Card Library** trabaja solo con los PICs de la familia 18Fxx.

Tabla 3.10. *Mmc_Fat_Init*:

| Función | Mmc_Fat_Init |
|-----------------------|--|
| Estructura | Mmc_Fat_Init(); |
| Devuelve | 0 - Si la tarjeta SD/MMC fue detectada e inicializada correctamente. 1 - Si el sector de arranque para FAT16 no fue encontrado. 255 - Si la tarjeta SD/MMC no fue detectada. |
| Descripción | Inicializa la tarjeta SD/MMC, lee el sector de arranque para FAT16, y extrae la información necesaria para la librería. NOTA: la tarjeta SD/MMC debe estar formateada en FAT16. |
| Requerimientos | Mmc_Chip_Select: línea de selección de la memoria. Mmc_Chip_Select_Direction: dirección de la línea de selección. El módulo de SPI debe ser inicializado previamente. |

Tabla 3.11. *Mmc_Fat_Assign*:

| Función | Mmc_Fat_Assign |
|--------------------|---|
| Estructura | Mmc_Fat_Assign(char nombre del archivo, char atributos del archivo); |
| Devuelve | 1 – Si el archivo ya existe, o no existe pero un nuevo archivo fue creado. 0 – El archivo no existe y no fue creado ningún nuevo archivo. |
| Descripción | Asigna un archivo para operaciones de lectura, escritura, y borrado. Todas las siguientes operaciones serán aplicadas al archivo asignado. Nombre del archivo: es el nombre del |

| | <p>archivo que se asigna para futuras operaciones. Y debe tener formato DOS 8.3, es decir un nombre y una extensión.</p> <p>Atributos del archivo: es un byte donde cada bit corresponde a una bandera de atributo que se le asigna al archivo en cuestión.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|--|---|------|-------------|---|---|--------------|---|---|--------|---|---|---------|---|---|---------------------|---|---|---------------|---|---|---------|---|---|---|---|---|---|
| | <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Descripcion</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Solo lectura</td> </tr> <tr> <td>1</td> <td>1</td> <td>Oculto</td> </tr> <tr> <td>2</td> <td>1</td> <td>Sistema</td> </tr> <tr> <td>3</td> <td>1</td> <td>Etiqueta de volumen</td> </tr> <tr> <td>4</td> <td>1</td> <td>Subdirectorio</td> </tr> <tr> <td>5</td> <td>1</td> <td>Archivo</td> </tr> <tr> <td>6</td> <td>1</td> <td>Elemento(solo de uso interno, nunca se encuentra en el disco)</td> </tr> <tr> <td>7</td> <td>1</td> <td>Bandera de creación de archivo(si no existe el archivo y la bandera esta seteada en 1, entonces un nuevo archivo será creado)</td> </tr> </tbody> </table> | Bit | Mask | Descripcion | 0 | 1 | Solo lectura | 1 | 1 | Oculto | 2 | 1 | Sistema | 3 | 1 | Etiqueta de volumen | 4 | 1 | Subdirectorio | 5 | 1 | Archivo | 6 | 1 | Elemento(solo de uso interno, nunca se encuentra en el disco) | 7 | 1 | Bandera de creación de archivo(si no existe el archivo y la bandera esta seteada en 1, entonces un nuevo archivo será creado) |
| Bit | Mask | Descripcion | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Solo lectura | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Oculto | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 1 | Sistema | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 1 | Etiqueta de volumen | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 1 | Subdirectorio | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 1 | Archivo | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 1 | Elemento(solo de uso interno, nunca se encuentra en el disco) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 1 | Bandera de creación de archivo(si no existe el archivo y la bandera esta seteada en 1, entonces un nuevo archivo será creado) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Requerimientos | Inicializar la SD/MMC y la librería para el manejo de archivos. | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tabla 3.12. Mmc_Fat_Reset:

| Función | Mmc_Fat_Reset |
|--------------------|--|
| Estructura | Mmc_Fat_Reset(unsigned longsize); |
| Devuelve | Nada. |
| Descripción | <p>Procede a resetear el puntero del archivo asignado (mueve este al inicio del archivo) para operaciones de lectura.</p> <p>Size: variable donde se guarda el tamaño del archivo asignado, luego de haberlo abierto en el modo de lectura.</p> |

| | |
|-----------------------|--|
| Requerimientos | Inicializar la tarjeta SD/MMC y asignar el archivo a leer. |
|-----------------------|--|

Tabla 3.13. *Mmc_Fat_Read*:

| Función | Mmc_Fat_Read |
|-----------------------|---|
| Estructura | Mmc_Fat_Read(unsigned shortdata); |
| Devuelve | Nada. |
| Descripción | Lee un byte del archivo actualmente asignado en el modo lectura. Una vez terminada la operación el puntero se coloca en el siguiente carácter. Data: buffer para guardar el dato leído. |
| Requerimientos | Inicializar la tarjeta SD/MMC, asignar el archivo a leer, e iniciarlo en el modo de lectura. |

Tabla 3.14. *Mmc_Fat_Rewrite*:

| Función | Mmc_Fat_Rewrite |
|-----------------------|--|
| Estructura | Mmc_Fat_Rewrite(); |
| Devuelve | Nada. |
| Descripción | Abre el archivo asignado actualmente en el modo de escritura. Si el archivo no está vacío su contenido será borrado. |
| Requerimientos | Inicializar la tarjeta SD/MMC y asignar el archivo a escribir. |

Tabla 3.15. *Mmc_Fat_Append*:

| Función | Mmc_Fat_Append |
|--------------------|--|
| Estructura | Mmc_Fat_Append(); |
| Devuelve | Nada. |
| Descripción | Abre el archivo asignado actualmente para agregarle un nuevo dato. Con de la ejecución de esta función el |

| | |
|-----------------------|--|
| | puntero de archivo se posiciona luego del ultimo byte. De modo que la siguiente operación de escritura se iniciara desde allí. |
| Requerimientos | Inicializar la tarjeta SD/MMC y asignar el archivo al cual se le va a añadir un byte. |

Tabla 3.16. *Mmc_Fat_Delete*:

| Función | Mmc_Fat_Delete |
|-----------------------|--|
| Estructura | Mmc_Fat_Delete(); |
| Devuelve | Nada. |
| Descripción | Elimina el archivo actualmente asignado de la tarjeta SD/MMC. |
| Requerimientos | Inicializar la tarjeta SD/MMC y asignar el archivo a eliminar. |

Tabla 3.17. *Mmc_Fat_Write*:

| Función | Mmc_Fat_Write |
|-----------------------|---|
| Estructura | Mmc_Fat_Write(data, data_len); |
| Devuelve | Nada. |
| Descripción | <p>Escribe un número determinado de bytes al archivo actualmente asignado y abierto en el modo de escritura.</p> <p>Data: dato a escribirse.</p> <p>Data_len: número de bytes a escribirse.</p> |
| Requerimientos | Inicializar la tarjeta SD/MMC, asignar el archivo, e iniciarlo en el modo reescribir o adjuntar datos. |

Tabla 3.18.Mmc_Set_File_Date:

| Función | Mmc_Set_File_Date |
|-----------------------|---|
| Estructura | Mmc_Fat_Set_File_Date(unsigned intaño, unsigned short més, unsigned short día, unsigned short hora, unsigned short minutos, unsigned short segundos); |
| Devuelve | Nada. |
| Descripción | Coloca la fecha y el tiempo al archivo actualmente asignado. Año: Valores validos: 1980-2107 Mes: Valores validos: 1-12 Día: Valores validos: 1-31 Hora: Valores validos: 0-23 Minutos: Valores validos: 0-59 Segundos: Valores validos: 0-59 |
| Requerimientos | Inicializar la tarjeta SD/MMC, asignar el archivo, e iniciarlo en el modo reescribir o adjuntar datos. |

Tabla 3.19. Mmc_Fat_Get_File_Date:

| Función | Mmc_Fat_Get_File_Date |
|-----------------------|--|
| Estructura | Mmc_Fat_Get_File_Date(unsigned intaño, unsigned short més, unsigned short día, unsigned short hora, unsigned short minutos, unsigned short segundos); |
| Devuelve | Nada. |
| Descripción | Devuelve la fecha/tiempo de creación del archivo actualmente asignado. |
| Requerimientos | Inicializar la tarjeta SD/MMC, asignar el archivo. |

Tabla3.20.Mmc_Fat_Get_File_Size:

| Función | Mmc_Fat_Get_File_Size |
|-----------------------|--|
| Estructura | Mmc_Fat_Get_File_Size(); |
| Devuelve | Retorna el tamaño (en bytes) del archivo actualmente asignado. |
| Descripción | Lee el tamaño del archivo actualmente asignado. |
| Requerimientos | Inicializar la tarjeta SD/MMC, asignar el archivo. |

Estas funciones se entenderán mejor ya en la programación, donde se explicara el diagrama de flujo para el microprocesador que maneja la tarjeta SD/MMC.

3.1.1. 6 Conexión de la memoria SD con el pic18f452

El circuito de la figura 3.10 muestra la memoria SDcard conectada al puerto C del Pic18f452 a través de las resistencias de 2.2K y 3.3K, usando los siguientes pines:

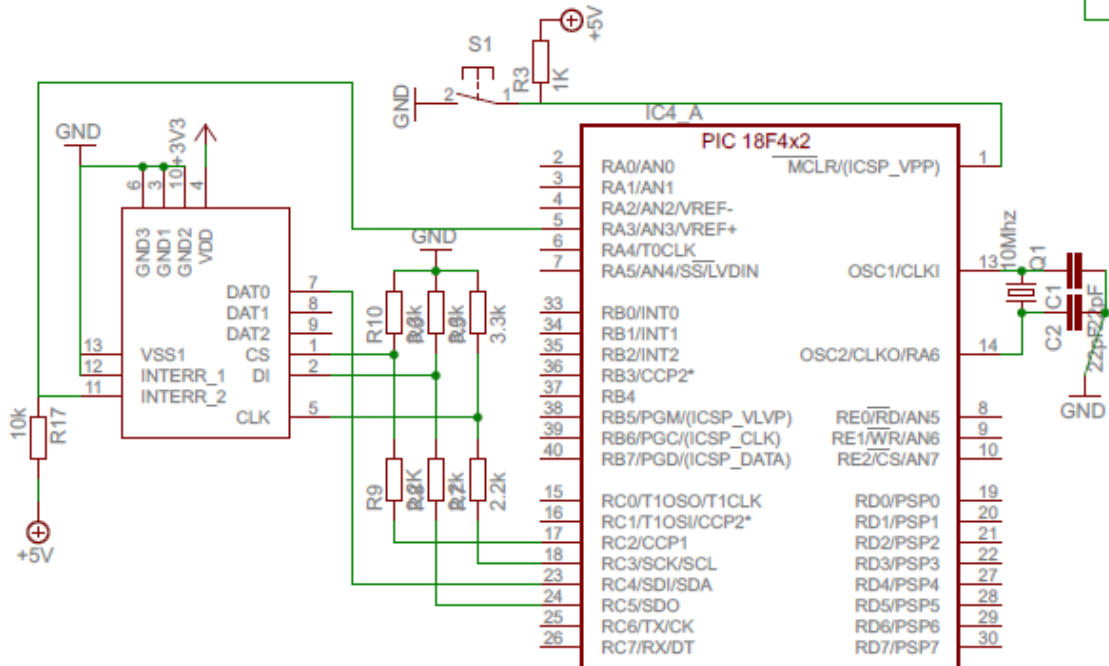


Figura 3.10 Conexión entre la SD y e pic18f452



SS (chip select) de la SD hasta el pin C2 del puerto C
SCLK de la SD hasta el pin C3 del puerto C
DO de la SD hasta el pin C4 del puerto C
DI de la SD hasta el pin C5 del puerto C

De acuerdo a las especificaciones de la SDcard, estas operan con un voltaje de alimentación de 2.6 hasta 3.7 voltios, pero cómo se tiene un regulador de tensión LM3.3 la memoria será alimentada con 3.3 voltios.

La salida producida por la memoria SD (2.475 V) es suficiente para que el driver del puerto C del Pic18f452 le detecte como un uno lógico (1), la salida que produce el Pic para un uno lógico es de 4.3 voltios y la máxima entrada admisible que permite la memoria SD es de 3.6 voltios. Entonces existe la necesidad de hacer un partidor de tensión para la entrada a los pines de la SDcard con una resistencia de 2.2K y 3.3K.

Entrada de tensión a la SDcard = $4.3V \times 3.3K / (2.2K + 3.3K) = 2.48V$

Entonces la tensión de entrada a los pines DI, CS (SS), SCLK de la memoria SD es de 2.48 voltios.

3.1.2 Reloj de tiempo real, comunicación con GPS con el protocolo NMEA

3.1.2 .1 RTC (Reloj en Tiempo Real)

Una vez que se produce un evento sísmico existe la necesidad de saber la fecha (día/mes/año) y hora (hora/minutos/segundos) a la que se sucede, entonces se da la necesidad de implementar un bloque basado en el integrado DS1307, el cual mediante protocolo de comunicación I2C es capaz de entregar fecha y hora como una cadena de caracteres, el chip tiene una pila de respaldo de 3.3V para evitar la pérdida de datos.

3.1.2.1.1 Circuito integrado DS1307

Un RTC (Reloj en tiempo real) brinda la lógica necesaria para el manejo del tiempo real (fecha y hora) en sistemas electrónicos, además dispone de una memoria NV SRAM de 56 bytes .

La transmisión de los datos se realiza de forma serial, tipo síncrono utilizando el bus de comunicaciones I2C. El DS1307 (RTC) entrega los datos de segundos, minutos, horas, días, mes y año, las fechas son ajustadas automáticamente para los meses que tienen 31, 30, 29 días, la hora puede ser ajustada a 12 o 24 horas, utilizando en el caso de 12 horas un indicador de am y pm, además dispone de un sensor interno que evalúa las fallas de la fuente de alimentación conmutando automáticamente con la batería de 3.3V.

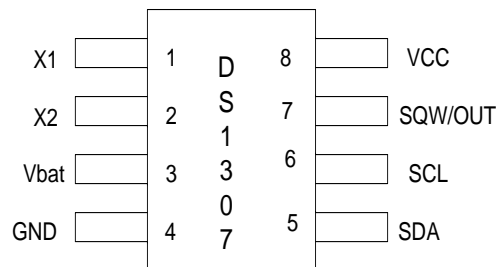


Figura 3.11 DS1307

Características:

- Consume menos de 500nA en operación con respaldo de batería.
- Temperatura de operación: -40°C hasta +85°C (más detalles en la hoja de Anexos).
- En la figura N° 3.10 consta la distribución de pines del integrado y en la tabla N°3.2 la descripción de los mismos.
- El diagrama de conexión del DS1307 sugerido por el fabricante se muestra en la figura N° 3.10.

| | | |
|---|------|---|
| 1 | X1 | pin para la conexión del osilador externo |
| 2 | X2 | pin para la conexión del osilador externo |
| 3 | Vbat | entrada para la pila externa, se debe utilizar pila de litio de 3.3V |
| 4 | Gnd | tierra |
| 5 | SDA | entrada /salida de datos seriales para la interfaz I2C, este pin es drenador abierto por lo que necesita de una resistencia de pull-up. |

| | | |
|---|---------|---|
| 6 | SCL | entrada de reloj usado para la sincronización de la comunicación en la interfaz I2C |
| 7 | SWQ/OUT | salida de señal cuadrada, puede generarse una salida de 4khz, 8Khz, 32Khz. |
| 8 | Vcc | Entrada para alimentación de voltaje, de 5V |

Tabla 3.21 Descripción de pines del DS1307

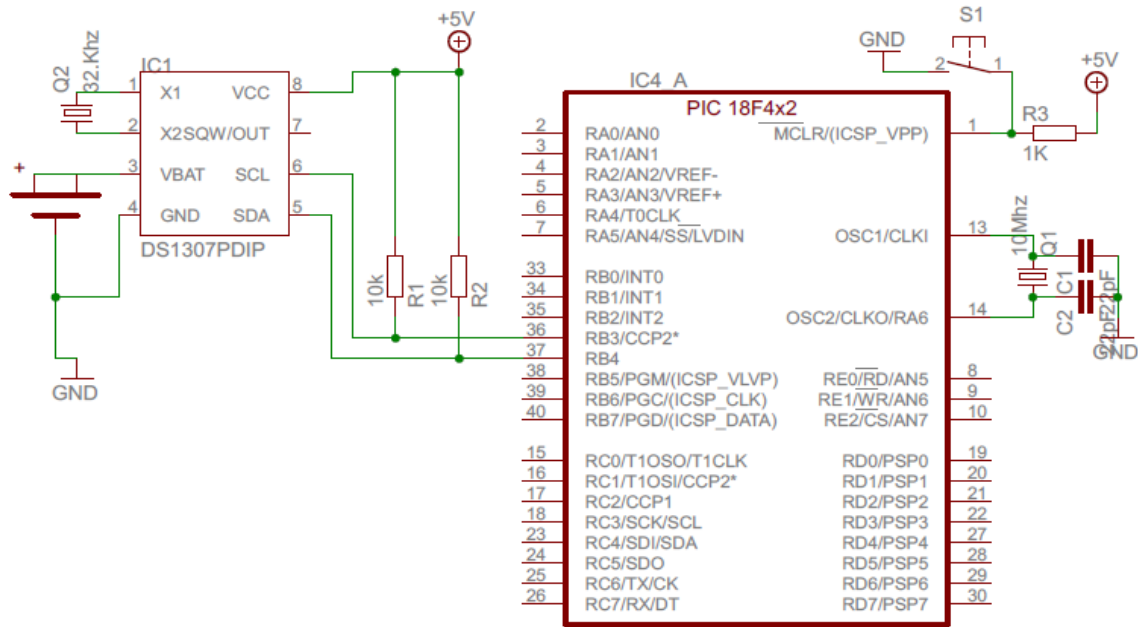


Figura 3.12 Conexión del DS1307

3.1.2.2 Receptor GPS

El sismógrafo digital lleva incorporado un GPS para determinar la posición exacta donde se encuentra ubicado, con una precisión de 2.5 metros.

3.1.2.2.1 Medidas de posicionamiento global

EL GPS tiene una precisión 2.5 metros y funciona mediante una red de 27 satélites (de los cuales 24 son los operativos) en órbita en el planeta tierra con trayectorias sincronizada para cubrir así toda la superficie. Cuando se desea determinar la posición, el receptor localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la posición y el reloj

de cada uno de ellos. Con base en estas señales, se sincroniza el reloj del GPS y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" calcula la posición en que éste se encuentra. La triangulación en el caso del GPS se basa en determinar la distancia de cada satélite respecto al punto de medición.

Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene las posiciones absolutas o coordenadas reales del punto de medición. También se consigue una exactitud en el reloj del GPS.

3.1.2.2.2 EI GPS MN5010HS

El receptor GPS utiliza un módulo que incorpora un chip MN5010HS que proporciona una alta sensibilidad para detectar satélites. Además el módulo incorpora una antena externa LNA y un LED de color verde que indica cuando está recibiendo las señales correctas de los satélites.

El chip tiene cuatro pines de control VDC, TX, RX y GND. Es un módulo que consume, 36mA, y por lo tanto deberá programarse para ser activado solamente en el momento de realizar la medición.

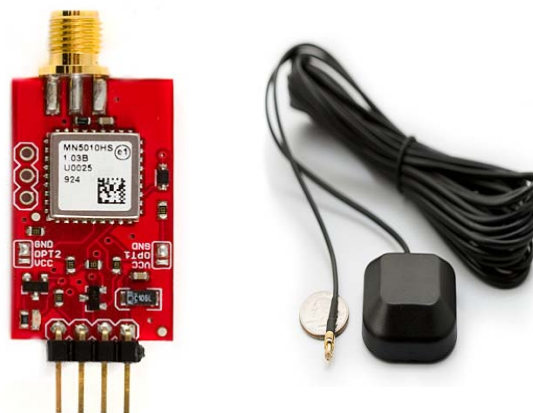


Figura 3.13 Módulo GPS y la antena

| | | |
|---|-----|---|
| 1 | Vcc | En pin es para la alimentación de 3.3 Vcd |
| 2 | TX | Transmite información hacia el Pic |
| 3 | Rx | Es pin es usado para la configuración del GPS (Recibir datos) |
| 4 | Gnd | Tierra (GND) |

Tabla 3.22 Pines de conexión del GPS

3.1.2.2.3 Recepción de datos

El módulo GPS tiene un puerto serial (Usart) para recibir datos RX y enviar datos TX. En el puerto para recibir datos se puede enviar instrucciones al GPS para configurar la velocidad de transmisión (1200,4800 bps), el tiempo, etc. El puerto de transmisión es el que envía los datos en formato NMEA que irán directamente al puerto de comunicaciones UART del Pic18f452. El puerto RX del GPS no será utilizado porque los valores que lleva configurados por defecto son suficientes para hacer funcionar el GPS. Por defecto envía información a una velocidad de 4800bps (Fig 3.14).

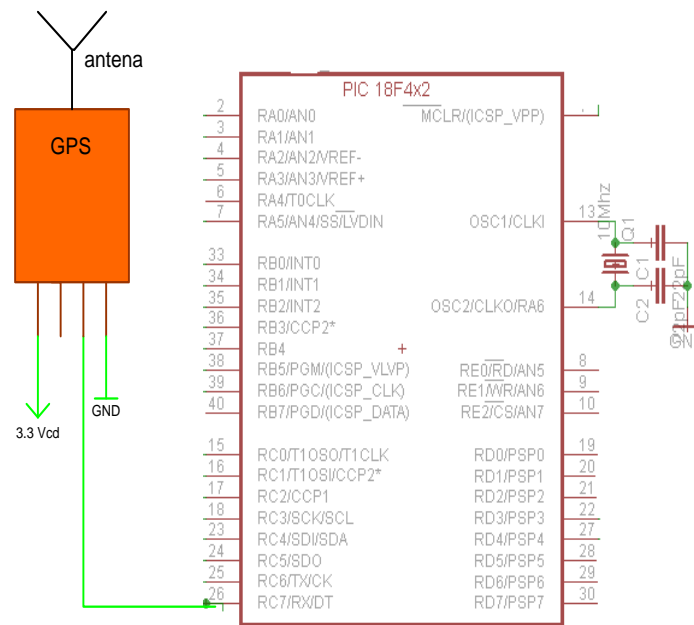


Figura 3.14 Conexión del GPS con el Pic mediante Usart a 4800 bps



Los datos que envía el GPS están en formato NMEA (National Marine Electronics Association) es una especificación combinada eléctrica y de datos. El protocolo utilizado exactamente es el NMEA 0183, que es el que utilizan la mayoría de receptores GPS.

Los comandos NMEA comienzan por el signo dólar (\$) seguido por una cabecera y una coma. Las frases terminan por “\r\n”. Estos son los valores que por defecto vamos a recibir (Tabla 3.2).

De la gran cantidad de información que se recibe del GPS solo se va a utilizar la hora y la fecha en formato GMT. Así que solo se utilizara los mensajes con cabecera \$GPRMC. Una trama con cabecera \$GPRMC tiene el siguiente aspecto:

```
$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,200910,020.3,E*68
```

| | |
|------------|---|
| 225446 | hora de lectura de los satélites 22:54:46 GMT |
| A | Un receptor de advertencia de navegación A=OK , V = advertencia |
| 4916.45,N | Latitud 49 deg. 16.45 min Norte |
| 12311.12,W | Longitud 123 deg. 11.12 min este |
| 000.5 | velocidad de desplazamiento del GPS sobre la tierra en nudos |
| 191194 | fecha de lectura de los satélites 20 septiembre2010 |
| 020.3,E | Variación MAGNETICA 20,3 grados Este |
| *68 | suma de comprobación obligatoria |

3.2 SISTEMA DE COMUNICACIONES.

3.2.1 Implementación de drivers para manejo de modem telefónico convencional

El sistema de comunicación está basado en un modem telefónico 73k224BL, en la figura 3.15 se observa cómo está integrado el modem en el sistema de comunicación.

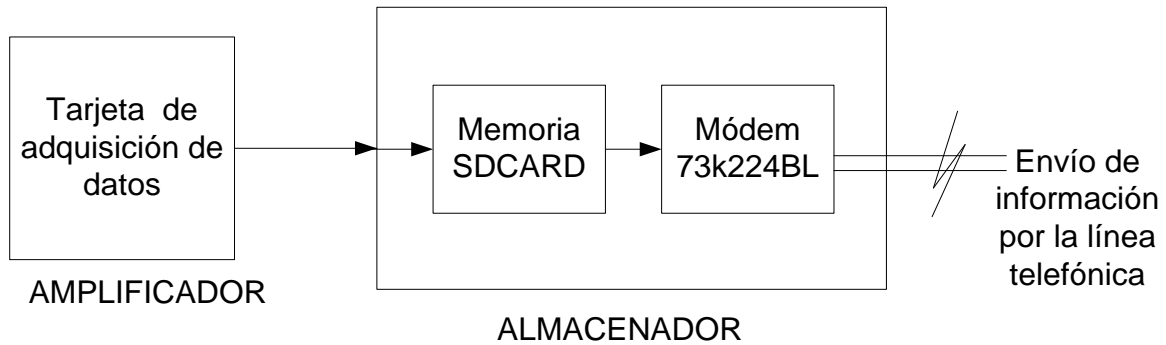


Figura 3.15 Esquema de comunicación del Sismógrafo

El envío de los datos a través de línea telefónica se la realiza con el modem 73K224BL. El Pic18f452 lee los datos almacenados en la memoria SD, los que son transferidos al modem en forma serial.

3.2.1.1 Modem 73k224BL

El 73K224BL es un módem que integra en un solo chip todas las funciones necesarias para la construcción de un módem V.22, V.22bis, capaces de transmitir información a 1200, 2400 bits/s respectivamente su funcionamiento en full-duplex sobre líneas telefónicas.

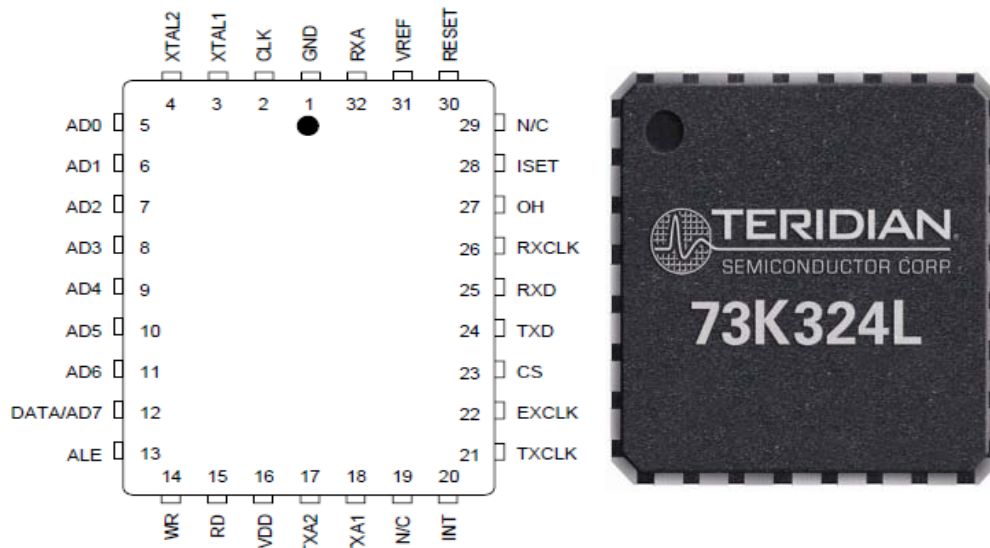


Figura 3.16 Modem 73k224BL

El 73K224BL funciona con una alimentación de 5 V teniendo un bajo consumo de energía, es de fácil integración con un Pic (Pic18f45), el control de las funciones del módem se las realiza a través de un bus de datos y direcciones multiplexado, por el cual se puede configurar los registros internos del modem, y leer los registros que indican el estado del modem. Un control ALE simplifica el direccionamiento del bus si es datos o direcciones.

El envío de información desde al pic al modem se lo realiza en forma serial a 600, 1200 y 2400 bits/s dependiendo del tipo de comunicación que se desea realizar.

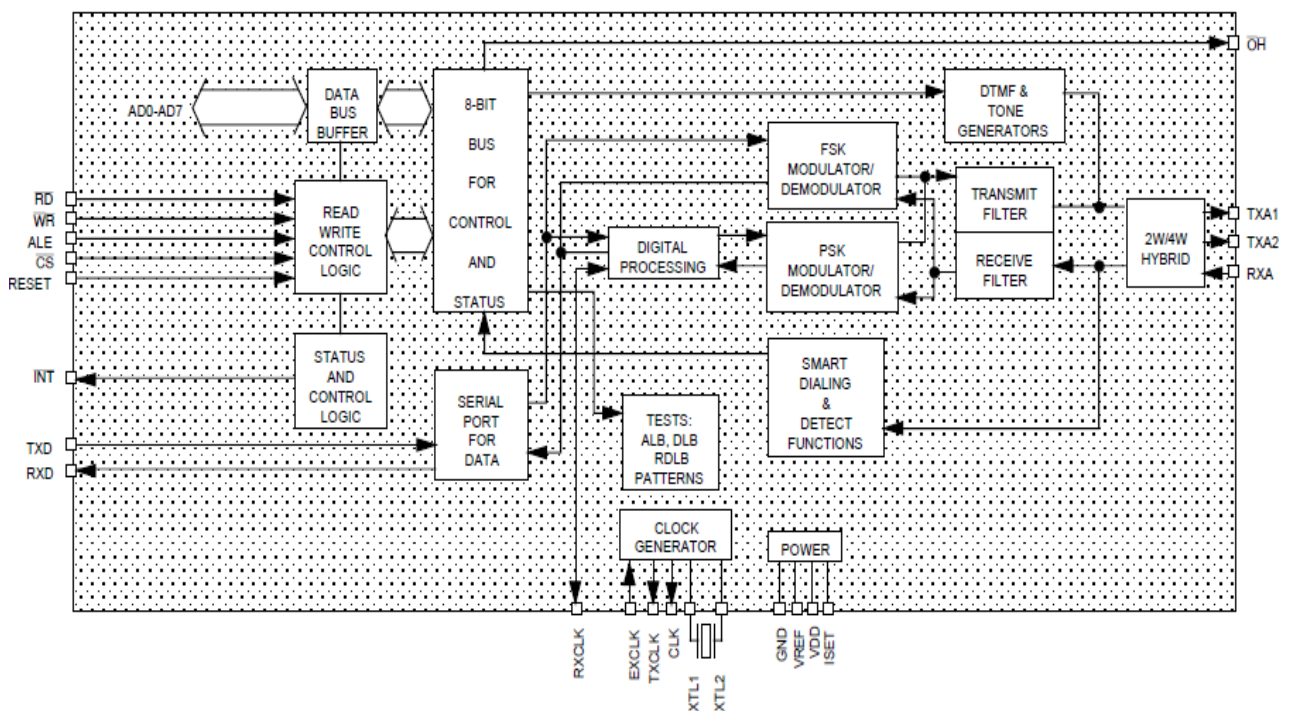


Figura 3.17 Diagrama interno del Modem 73k224BL

Características:

- Modem telefónico en un solo chip
- Tiene un controlador híbrido para dos a cuatro hilos
- Configurable para varios tipos de comunicaciones V.22bis, V.22, V.21 Bell212A/103, además es compatible con el envío de datos a FSK (300 bits/s), DPSK (600, 1200 bit/s), QAM (2400 bits/s).
- Interactúa directamente cualquier tipo de microcontrolador.
- Modo de funcionamiento Sincrónico y Asincrónico.
- Tiene un bus paralelo para el control interno de los registros.

- Genera tonos DTFM, tonos respuesta y genera tono de guardia.
- Programable para transmitir los datos a través de una línea telefónica, con una atenuación de 16db hasta el 1db. (programables en pasos de un 1db).
- Modalidades de prueba ALB, DL , RDL, generación y detecciones de patrones S1.

3.2.1.1.1 Descripción de los pines

Tabla 3.23. Descripción de pines de modem

| Pin | Nombre | tipo | descripción |
|-----|--------|------|--|
| 1 | GND | I | Tierra del Sistema |
| 16 | VCC | I | Entrada de alimentación de 5 V \pm 10% (73K224BL). Bypass con 0,1 y condensadores de 22 uF a Gnd |
| 31 | VREF | O | Un voltaje de referencia generado internamente. Bypass con Condensador de 0,1 uF a tierra |
| 28 | ISET | I | Corriente de referencia para el chip. Corriente de suministro para los amplificadores internos. La conexión se establece a través de una resistencia de 2 Mega ohmios conectada a VCC. |

| Pin | Nombre | tipo | descripción |
|------|---------|-----------------|---|
| 13 | ALE | O | Habilita retención de direcciones. Las direcciones AD0-AD2 son retenidas en el flanco de bajada del ALE y previamente se tiene que seleccionar el modem con !CS en bajo 0 lógico. |
| 5-12 | AD0-AD7 | I-O / triestado | Bus de datos/direcciones. Este bus es multiplexado es decir bidireccional lleva información desde y hacia el Pic. El bus de direcciones es A0-A2. |

| | | | |
|----|-------|---|--|
| 23 | !CS | I | Chip select. Una baja en este pin permite un ciclo de lectura o un ciclo de escritura. |
| 2 | CLK | O | Salida del reloj. Es pin se puede usar como reloj del Pic. |
| 20 | !INT | O | Interrupción. Este pin es de salida para informar cuando se ha producido una llamada telefónica, o se ha dado algún cambio en el funcionamiento interno del modem. |
| 15 | RD | I | Lectura. Un bajo en este pin indica que se tiene que leer los registros internos del 73k224BL. Para realizar una lectura RD y CS deben estar en bajo. |
| 30 | RESET | I | Una señal de alto en este pin pone al modem en estado de reposo. Todos los registros CR0,CR1, CR2, CR3 y TONO (TR) son puestos en cero. La salida CLK es puesta a la frecuencia del cristal externo. |
| 14 | WR | I | Escritura. Un bajo en este pin indica que se tiene que escribir los registros internos del 73k224BL. Para realizar una escritura WR y CS deben estar en bajo. |

| Pin | Nombre | tipo | descripción |
|-------|-----------|------|--|
| 32 | RXA | O | Interfaz para la línea telefónica. Pin para la entrada de la señal analógica modulada desde el teléfono. |
| 18/17 | TX1/TX2 | O | Transmite productos diferenciales. Estos pines proporciona la señal analógica que se transmite a la línea telefónica. Estos pines se deben acoplar a la línea a través de un transformador de aislamiento (para proporcionar una aislación galvánica) y una resistencia. Se puede construir también un híbrido externo solo utilizando el pin TX1. |
| 3/4 | XTL1/XTL2 | I | Estos pines son para el oscilador interno que requieren un cristal de cuarzo externo |

| | | | |
|----|----|---|---|
| | | | de 11,092 Mhz, condensadores de carga deben estar conectados a los pines XTL1 y XTL2 a tierra. |
| 27 | OH | O | IC DRIVER (descolgado). Es pin es capaz de manejar la bobina e un relé de hasta 30 ma. La salida es el complemento del bit de registro OH. Se utiliza para conectar o desconectar el modem a la línea telefónica. |

| Pin | Nombre | tipo | descripción |
|-----|--------|------|---|
| 22 | EXCLK | I | Reloj externo. Esta señal se utiliza cuando la transmisión va a realizar en forma síncrona y la opción externa de reloj ha sido seleccionada. |
| 26 | RXCLK | O | Reloj de recepción. En cada flanco de bajada del reloj es coincidente coincide con la transición de los datos recibidos en forma serial. Cada flanco de subida de este reloj se utiliza para la transmisión de datos. |
| 25 | RXD | O | Salida de datos recibidos. Los datos seriales recibidos están disponibles en este pin. Los datos son válidos en el flanco de subida de RXCLK cuando está en el modo síncrono. |
| 21 | TXCLK | O | Reloj para transmitir. Esta señal se utiliza para transmitir los datos en forma síncrona, a la entrada serie del pin TXD. El reloj para la transmisión puede venir de diferentes fuentes dependiendo de la configuración en los registros del modem. En el modo interno el reloj se genera internamente. En el modo externo |
| 24 | TXD | | Transmite datos de entrada. Para la transmisión de datos se aplica en este pin en forma serial. En modo síncrono los datos son leídos en cada flanco de subida del reloj TXCLK. En modo asíncrono se puede |

transmitir en varios modos (1200/600 bits/s o 300 baudios), no es necesario que el reloj este TXCLK este sincronizado con el PIC. Los datos en DPSK deben ser de 1200/600 bits/s.

3.2.1.1.2 Descripción de los registros internos del modem

Ocho registros internos de 8 bits son accesibles para control y supervisión del estado. Los registros se pueden acceder para operaciones de escritura o lectura. El registro CR0 controla el método por el cual los datos se transfieren a través de la línea telefónica.

CR1 controla como se va a realizar la comunicación entre el PIC y el 73K224BL.

DR es el registro de detección, proporciona la información sobre las condiciones del estado del modem.

TR es el control de registro, permite la generación de tonos DFTM, permite la generación de tonos de guarda.

CR2 sirve para configurar al modem de acuerdo al estado de la línea telefónica.

CR3 da la opción de establecer la amplitud de la señal que será transferida a la línea, y se selecciona la fuente para la transmisión de los datos.

ID este registro es solo de lectura e indica que tipo de modem se está utilizando.

Tabla 3.23 Registros del modem



| REGISTER | ADDRESS | DATA BIT NUMBER | | | | | | | | |
|---------------------------|---------|---|--------------------------------------|--|---|---|--|--|--|--|
| | | AD2 - AD0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| CONTROL REGISTER 0 | CR0 | 000 | MODULATION OPTION | MODULATION TYPE 1 | MODULATION TYPE 0 | TRANSMIT MODE 2 | TRANSMIT MODE 1 | TRANSMIT MODE 0 | TRANSMIT ENABLE | ORIGINATE/ ANSWER |
| | | QAM: 0 = 2400 BIT/S DPSK: 0 = 1200 BIT/S 1 = 600 BIT/S FSK: 0 = 103 MODE 1 = V.21 | | 10-QAM 00-DPSK 01-FSK | 0000 = PWR DOWN 0001 = INT SYNCH 0010 = EXT SYNCH 0011 = SLAVE SYNCH 0100 = ASYNCH 8 BITS/CHAR 0101 = ASYNCH 9 BITS/CHAR 0110 = ASYNCH 10 BITS/CHAR 0111 = ASYNCH 11 BITS/CHAR 1X00 = FSK | | | 0 = DISABLE TXA OUTPUT 1 = ORIGINATE TXA OUTPUT | 0 = ANSWER TXA OUTPUT 1 = ORIGINATE | |
| CONTROL REGISTER 1 | CR1 | 001 | TRANSMIT PATTERN 1 | TRANSMIT PATTERN 0 | ENABLE DETECT INTERRUPT | BYPASS SCRAMBLER | CLK CONTROL | RESET | TEST MODE 1 | TEST MODE 0 |
| | | 00 = TX DATA 01 = TX ALTERNATE (DOTTING) 10 = TX MARK 11 = TX SPACE | | 0 = DISABLE 1 = ENABLE | 0 = NORMAL 1 = BYPASS SCRAMBLER | 0 = XTAL 1 = 16 X DATA RATE OUTPUT AT CLK PIN IN QAM/DPSK MODE ONLY | 0 = NORMAL 1 = RESET | 00 = NORMAL 01 = ANALOG LOOPBACK 10 = REMOTE DIGITAL LOOPBACK 11 = LOCAL DIGITAL LOOPBACK | | |
| DETECT REGISTER READ ONLY | DR | 010 | RECEIVE LEVEL INDICATOR | S1 PATTERN DETECT | RECEIVE DATA | UNSCR. MARKS DETECT | CARRIER DETECT | ANSWER TONE DETECT | CP TONE DETECT | SIGNAL QUALITY INDICATOR |
| | | 0 = SIGNAL BELOW THRESHOLD 1 = ABOVE THRESHOLD | 0 = NOT PRESENT 1 = PATTERN FOUND | OUTPUTS RECEIVED DATA STREAM | 0 = CONDITION NOT DETECTED 1 = CONDITION DETECTED | 0 = GOOD 1 = BAD | | | | |
| tone CONTROL REGISTER | TR | 011 | RXD OUTPUT CONTROL | TRANSMIT GUARD/ TONE | TRANSMIT ANSWER TONE | TRANSMIT DTMF | DTMF3 | DTMF2/ 4W/4DX | DTMF1/ EXTENDED OVERSPEED | DTMF0/ GUARD/ ANSWER |
| | | RXD PIN 0 = NORMAL 1 = OPEN | 0 = OFF 1 = ON | 0 = OFF 1 = ON | 0 = DATA CARRIER 1 = TX DTMF | 4 BIT CODE FOR 1 OF 16 DUAL TONE COMBINATIONS | | 0 = 1800 Hz G.T. 2225 Hz ANNS TONE GENERATED 1 = 550 Hz ANS TONE 2100 Hz ANS TONE GENERATED & DETECTED (V.21, V.22) | | |
| CONTROL REGISTER 2 | CR2 | 100 | 0 | SPECIAL REGISTER ACCESS | CALL INITIALIZE | TRANSMIT S1 | 16 WAY | RESET DSP | TRAIN INHIBIT | EQUALIZER ENABLE |
| | | | | 0 = ACCESS CR3 1 = ACCESS SPECIAL REGISTER | 0 = DSP IN DEMOD MODE 1 = DSP IN CALL PROGRESS MODE | 0 = NORMAL DOTTING 1 = S1 | 0 = RX-TX 1 = RX-16WAY TX-4WAY IN DPSK | 0 = DSP INACTIVE 1 = DSP ACTIVE | 0 = ADAPT EQ ACTIVE 1 = ADAPT EQ FROZEN | 0 = ADAPT EQ IN RESET STATE 1 = ADAPT EQ ACTIVE |
| CONTROL REGISTER 3 | CR3 | 101 | TXDALT | TRISTATE TX/RXCLK | OH | RECEIVE GAIN BOOST | TRANSMIT ATTEN. 3 | TRANSMIT ATTEN. 2 | TRANSMIT ATTEN. 1 | TRANSMIT ATTEN. 0 |
| | | ALTERNATE TRANSMIT DATA SOURCE | 0 = NORMAL 1 = TRISTATE | 0 = OH RELAY DRIVER OPEN 1 = OH OPEN DRAIN DRIVER PULLING LOW | 0 = NO BOOST 1 = 18 dB BOOST | 0000-1111, 0.25dB TRANSMIT ATTENUATOR 16 dB RANGE DEFAULT=0100 -10 dB=0 | | | | |
| SPECIAL REGISTER | SR | 101 | 0 | TX BAUD CLOCK | RX UNSCR. DATA | 0 | TXD SOURCE | SQ SELECT 1 | SQ SELECT 0 | 0 |
| | | | | OUTPUTS TXBAUD CLOCK | OUTPUTS UNSCR. DATA | 0 = TXD PIN 1 = TXALT BIT | 00 10 ⁻⁵ BER 01 10 ⁻⁶ BER 10 10 ⁻⁴ BER 11 10 ⁻³ BER | | | |
| ID REGISTER READ ONLY | 10 | 110 | 1 | 1 | 0 | 0 | X | X | X | X |

00XX-73K212AL, 322L, 321L
 01XX-73K221AL, 302L
 10XX-73K222AL, 222BL
 1100-73K224L, 224BL
 1110-73K224L, 324BL

0 = Only write zeros to these locations
 X = Undefined, mask in software

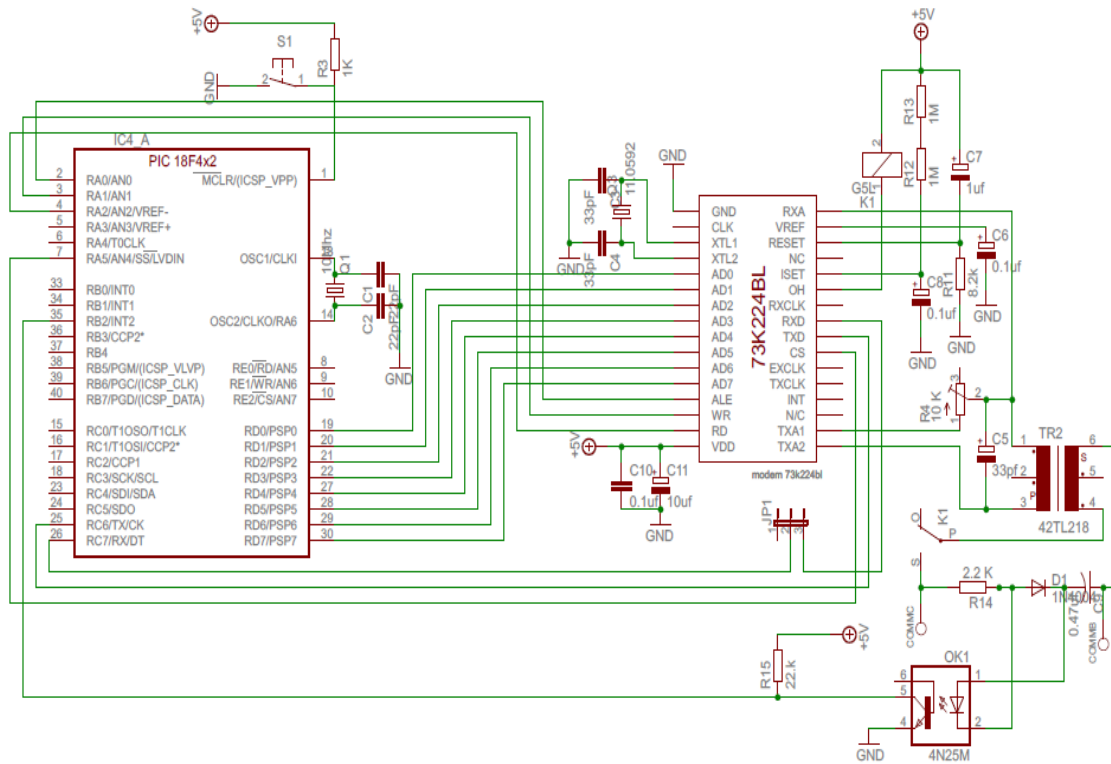


Figura 3.18 Conexión del modem 73K224BL con el Pic

3.2.1.1.3 Consideraciones generales para diseño

Como se observa en la figura 3.18 el esquema de conexión básico para el modem 73k224BL, en el circuito se utiliza un PIC18f452 para el control de todas las funciones internas como externas del modem, un puerto RS232 serie para los datos, y una interfaz con la línea telefónica.

El 73K224BL se puede configurar de dos modos, una interfaz paralela y otra para una interfaz en serie. La interfaz paralela está diseñada para usarse con el Pic18f452 u otros microcontroladores con el cual el control del modem se lo realiza por un bus paralelo de 8bits multiplexado entre direcciones y datos, es decir el bus puede ser de datos o direcciones dependiendo del ALE. La interfaz en serie se utiliza en aplicaciones en las que sólo se tiene un número limitado de hilos para la comunicación entre el PIC y el modem.

3.2.1.1.4 Transformador de aislamiento

Para poder enviar los datos a través de la línea telefónica es necesario adaptar la impedancia de la línea con el modem. Para esto se usa un transformador (figura 3.19) de acoplamiento para telecomunicaciones.

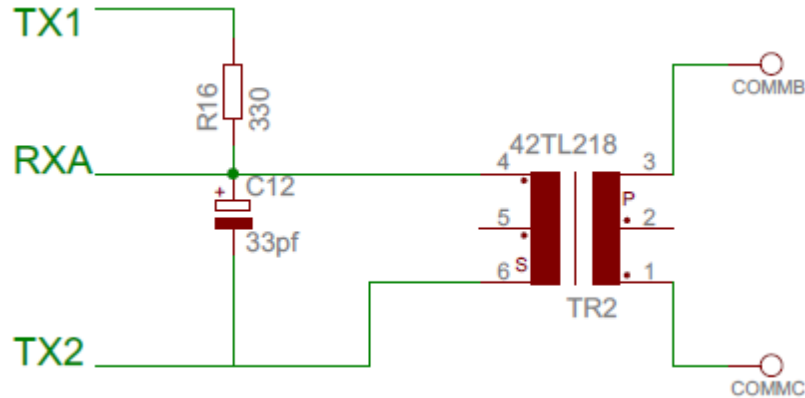


Figura 3.19 Interfaz analógica HÍBRIDO DE CARGA

3.2.1.1.5 Circuito detección de llamada

Cuando la línea telefónica se encuentra en estado de llamada, se presenta una tensión de alrededor 100voltios con una frecuencia de 25 ciclos de forma intermitente hasta que se descuelga el teléfono, en ese momento la corriente de llamada cesa y la tensión de la línea baja hasta una tensión aproximada de 12 a 15Vca. Entonces lo que se necesita es un circuito electrónico que solo detecte la llamada telefónica.

En primer lugar se debe tener en cuenta que el circuito sólo debe actuar cuando exista corriente de llamada, al ser ésta una corriente alterna, se puede discriminarla mediante un condensador de 470nf, el cual sólo dejará pasar la corriente alterna, bloqueando la continua, la resistencia de 2k2 es para limitar la corriente que ingresa al optoaislador.

Cada vez que se produce una llamada el optoaislador va a entregar un tren de pulsos en el pin 5. Por lo tanto el pulso de salida del optoaislador será un reflejo fiel de la señal de llamada de la línea telefónica.

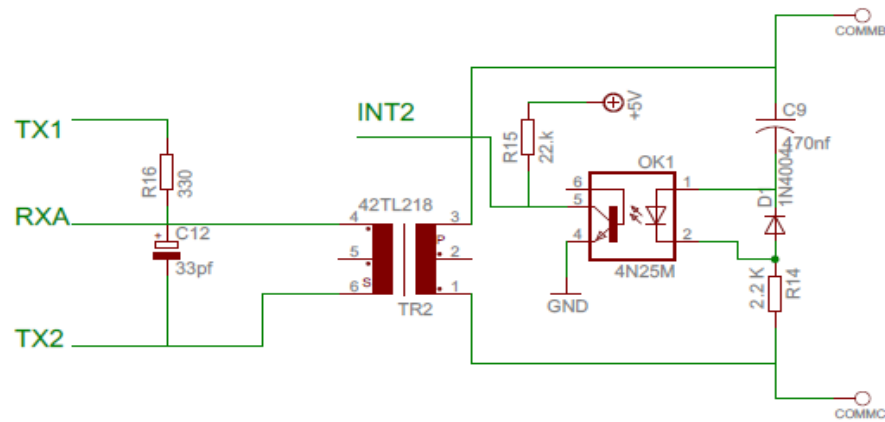


Figura 3.20 Circuito detector de Ring

El modem 73k224BL necesita de un cristal externo que opera a 11,0592 MHz, Es importante que esta frecuencia se mantenga con una tolerancia de $\pm 0,01\%$ de precisión para no afectar las operaciones internas del modem, debe tener un condensador conectado a la unión de cada uno de los pines del cristal, el valor de los condensadores es 33 pf.

Se recomienda colocar un condensador 22 uF electrolítico en paralelo con un 0,1 uF condensador cerámico entre VDC y GND en la entrada a VDC del modem para eliminar el ruido de alta y baja frecuencia que se puede producir al modem por la alimentación.

3.3 DESPLIEGUE EN INTERFAZ GRAFICO

3.3.1 Pantalla táctil

Una pantalla táctil es la unión de un panel táctil con un GLCD. Un panel táctil (figura 3.21) es un fino panel autoadhesivo que se coloca en la parte superior de una pantalla de un GLCD gráfico. Este muy sensible a la presión de forma que un suave toque provoca algunos cambios en la señal de salida.

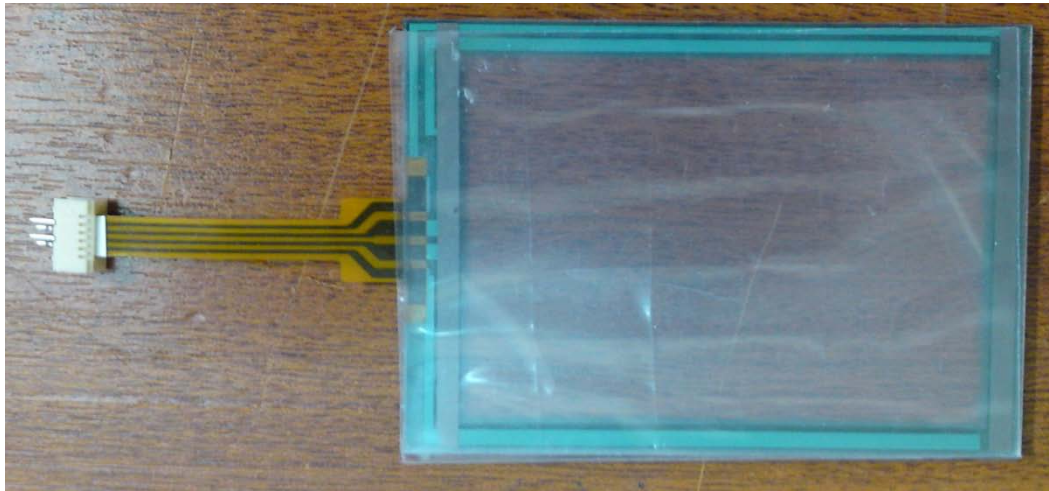


Figura 3.21 Panel Táctil

3.3.2 Principio de funcionamiento

Un panel táctil resistivo está compuesto por dos láminas rígidas transparentes, formando una estructura tipo sándwich, que tienen una capa resistiva en sus caras internas. La resistencia de estas capas no excede de 1Kohm.

Los lados opuestos de las láminas disponen de contactos para acceder a un cable plano los cuales irán conectados al Pic18f452.

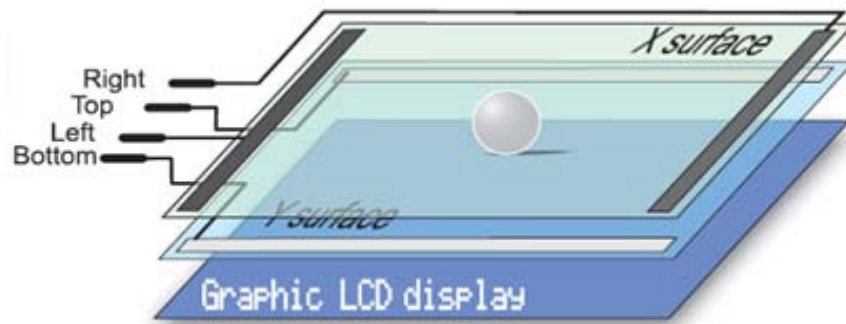


Figura 3.22 Estructura interna del panel táctil



Figura 3.23 Cable plano conectado al panel táctil

3.3.3 Determinación de coordenadas de la pantalla táctil

El proceso para la determinación de las coordenadas de la posición del panel, que ha sido presionada puede dividirse en dos pasos.

- Primero la determinación de las coordenadas del eje X
- Segundo la determinación de las coordenadas del eje Y

Para la determinación de la coordenada X, es necesario conectar el contacto izquierdo de la superficie (eje X) X a GND y el contacto derecho a la fuente de alimentación (5 VDC). Esto permite tener un divisor de tensión cuando se presiona el panel táctil. El valor de la tensión resultante se puede leer en el contacto inferior de la coordenada Y, el valor de la lectura puede variar entre 0 a 5 voltios. Si el punto presionado está cerca del lado izquierdo la el valor de la tensión leída será de 0 VDC pero si se presiona la pantalla cerca del lado derecho el valor leído será de 5 VDC y si la presión es ejercida en el punto medio la tensión de lectura será de 2.5 VDC.

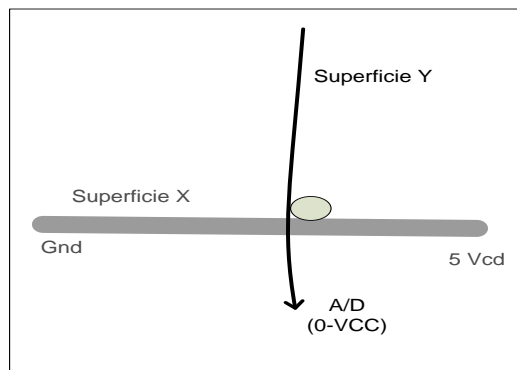


Figura 3.24 Determinación de la coordenada x

Para la determinación de la coordenada Y, es necesario conectar el contacto inferior de la superficie Y a GND, y el contacto superior se conectara a 5 VDC, en este caso la lectura de la tensión obtenida al presionar sobre el panel se realizara en el contacto izquierdo de la superficie X.

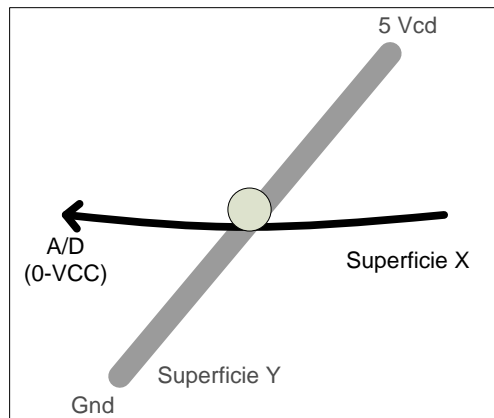


Figura 3.25 Determinación de la coordenada Y

3.3.4 Conexión al Pic18f452

Para conectar el panel táctil al Pic es necesario, implementar un circuito que, permita conectar los contactos del panel táctil a masa y a la tensión de alimentación en el momento adecuado, Para la determinación de las coordenadas del eje X del eje Y, es necesario implementar el circuito de la figura 3.26.

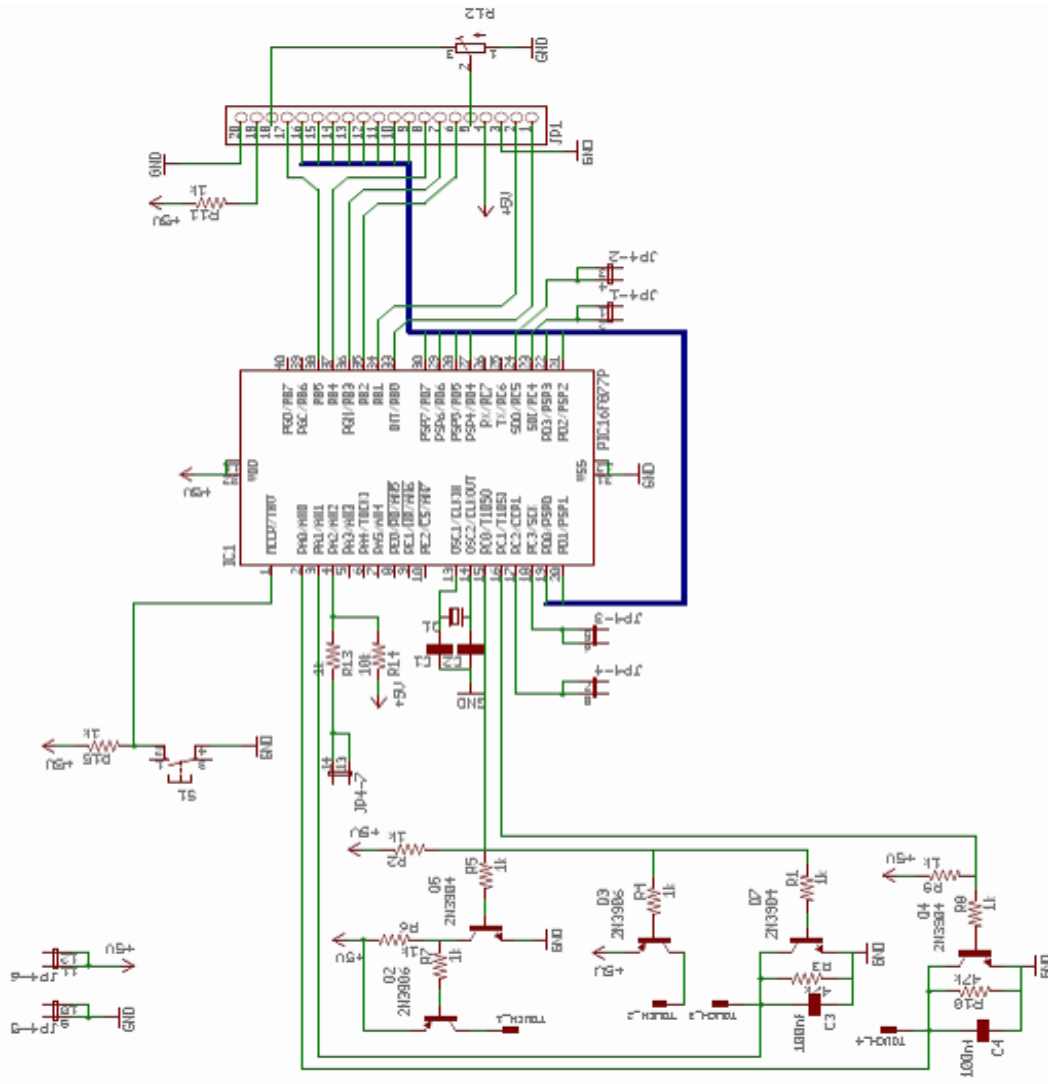


Figura 3.26 Conexión de la pantalla táctil con el Pic18f452

El contacto inferior de la superficie Y la superficie X están conectados a los Pines AN0, AN1 del Pic que son las entradas del conversor Analógico/Digital del Pic. Las coordenadas se determinan midiendo la tensión en los respectivos contactos.

3.3.5 Uso de librerías de MikroC Pro 3.2 para la pantalla táctil



| | |
|-------------------|------------------------------------|
| ADC_Read() | lee el valor analógico |
| Glcd_box() | dibuja un cuadrado relleno |
| Glcd_circle() | dibuja un circulo |
| Glcd_Dot() | dibuja en rectángulo |
| Glcd_Fill() | Limpia la pantalla |
| Glcd_H_Line() | dibuja una línea horizontal |
| Glcd_Image() | importa una imagen |
| Glcd_Init() LCD | inicia el GLCD |
| Glcd_Line() | dibuja una línea |
| Glcd_Read_Data() | lee datos de la memoria de la GLCD |
| Glcd_Rectangle() | dibuja un rectángulo |
| Glcd_Set_Font() | Seleccione la fuente |
| Glcd_Set_Page() | Selecciona la cara |
| Glcd_Set_Side() | selecciona el lado del GLCD |
| Glcd_Set_X() | Determina la coordenada X |
| Glcd_V_line() | Dibuja una línea vertical |
| Glcd_Write_Char() | escribe un carácter |
| Glcd_Write_Data() | escribe un dato |
| Glcd_Write_Text() | escribe un texto |

3.4 INTEGRACIÓN DE LAS PARTES

3.4.1 Objetivos.

Los objetivos más importantes que se consideran para el diseño del sismógrafo digital son:

- Muestreo de señales analógicas del geófono de tres componentes.
- Detección de eventos, en base a comparación de las señales digitalizadas con valores previamente determinados.
- Almacenamiento de eventos en la tarjeta SD/MMC.
- Graficar la forma de onda generada por un movimiento. Para este propósito se utiliza el interfaz gráfico "TOUCH SCREEN".
- Transmisión de la información almacenada en la SD/MMC, a través de la línea telefónica convencional.

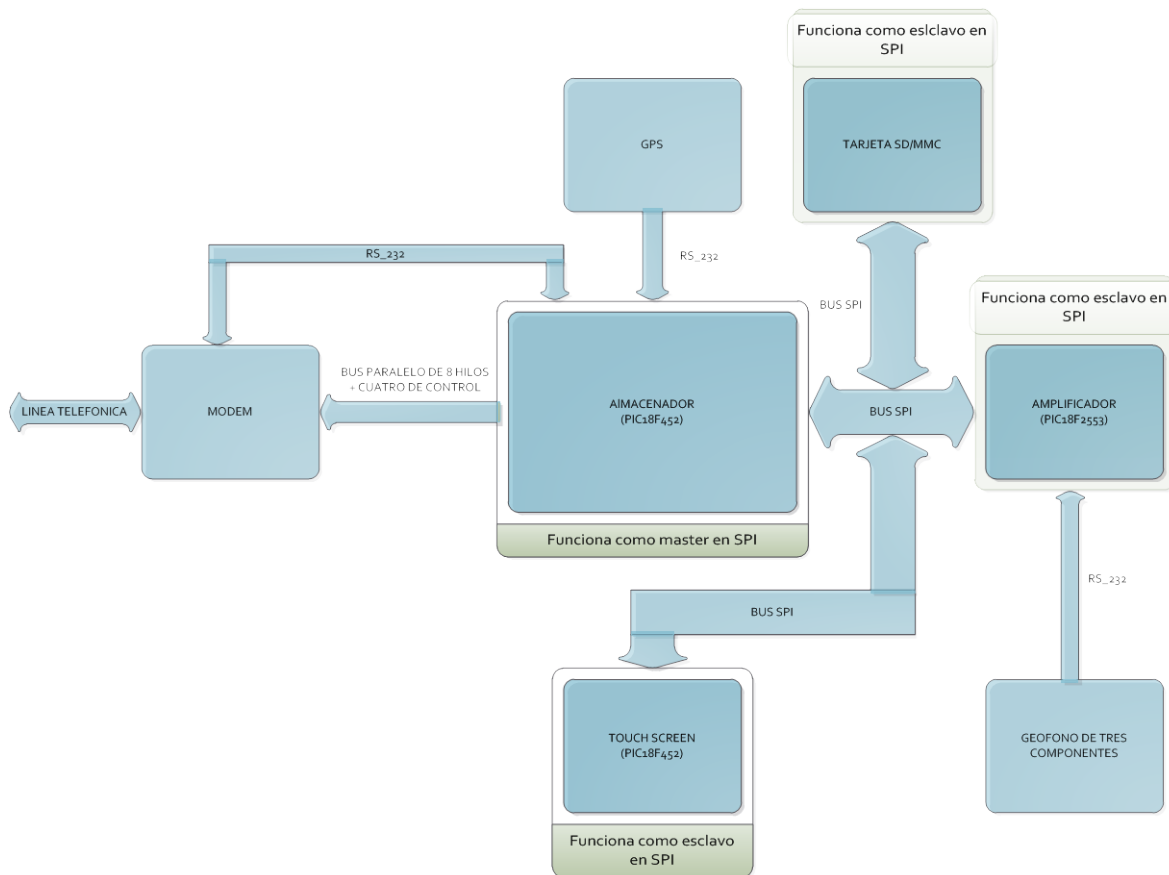


Figura 3.27 Diagrama de bloques del sismógrafo digital.

3.4.2 Funcionamiento general.

El sistema completo consta de tres microprocesadores, comunicados mediante el protocolo SPI. Donde se utiliza la siguiente configuración:

1. Sistema de muestreo y amplificación de señales (PIC18F2553, como Esclavo).
2. Sistema de almacenamiento y georeferenciación (PIC18F4552, como Master)
3. Sistema de visualización (PIC18F4552, como Esclavo)

El microprocesador denominado master, es el que controla todo lo referente a: comunicación SPI, almacenamiento de los eventos, y transmisión de datos a través de la línea telefónica.

El amplificador digitaliza las señales de los geófonos y las transmite al PIC master para almacenarlas cuando sea necesario. Por otra parte el bloque de interfaz gráfico, es solo para verificar que la parte de amplificación y comunicación estén en perfecto estado.

A continuación se detalla, que es lo que hace y cómo funciona cada uno de los bloques del sistema.

3.4.2.1 Sistema de muestreo y amplificación

Esta tarjeta se encarga de la amplificación y muestreo de las señales provenientes; del geófono vertical y los dos horizontales. La detección de eventos se realiza mediante la comparación de la señal digitalizada del canal vertical, una vez que esta sobrepasa determinado nivel de tensión, entonces se dispara el mecanismo de detección.

El microprocesador PIC18F2553 está constantemente verificando la existencia de un evento, de ser así activa la interrupción INT0 y espera la respuesta por parte del master. Una vez que el master detecta la interrupción, inmediatamente atiende la subrutina de almacenamiento de datos, comunicándose con el amplificador y grabando en la SD/MMC por 60 segundos. Para establecer el tiempo de almacenamiento se utiliza el TIMER0 del PIC master.

Debido a que mayoría de los sismos promedio duran entre 10-15 segundos pero existen registros de hasta 3 min, por lo que se optó tomar los 60 seg como valor predeterminado para grabar.

Además de la detección de eventos, el amplificador se encarga de filtrar elementos que generalmente no son sismos como por ejemplo, el paso de una persona cerca de la estación sensora, garantizando así que la mayoría de los registros sean sismos reales. Para esto simplemente se utiliza el TIMER0 del PIC18F2553.

3.4.2.2 Sistema de almacenamiento y georeferenciación

Este bloque tiene como elemento principal el Microprocesador PIC18F452. Es el encargado de establecer y controlar la comunicación SPI entre todos los dispositivos conectados al bus. Pero quizás la parte más importante es; guardar en la tarjeta SD/MMC, los datos que envía el PIC18F2553 cuando ocurre un evento sísmico.

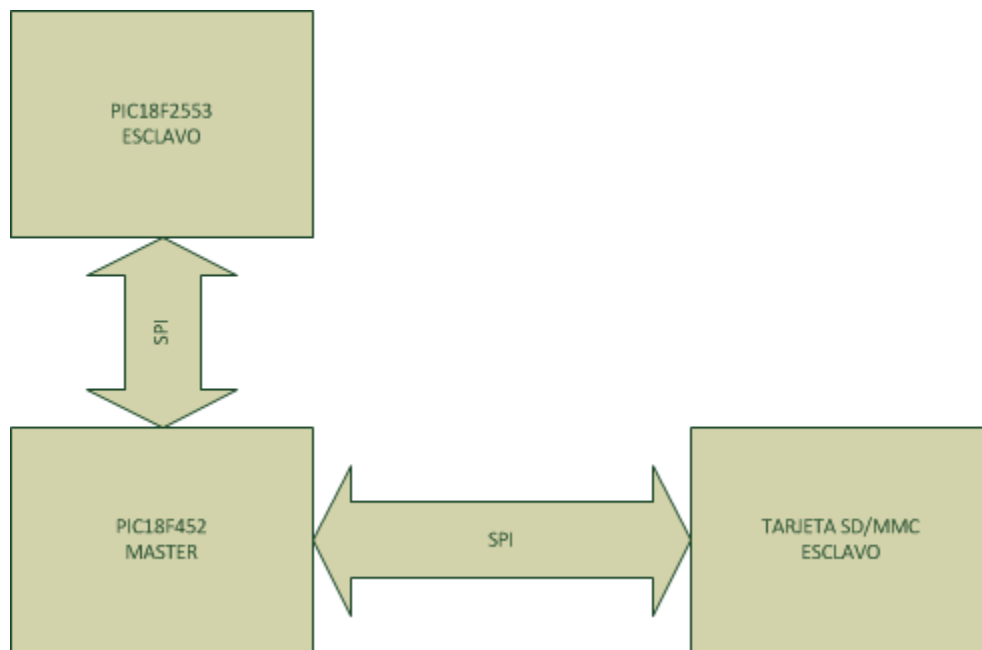


Figura 3.28 Comunicación SPI durante un evento.

También realiza la lectura del GPS cada cierto tiempo, para igualar el reloj de tiempo real (RTC). Con esto se evita cualesquier error en lo referente al momento en que se dio el sismo, grabando siempre la hora exacta en que sucedió.

Debido a las limitaciones que se tiene en lo referente a la memoria RAM del PIC master. La subrutina de envío de datos a través de la línea telefónica, se probará e implementará individualmente. Para integrarla cuando se tenga un PIC con mayores prestaciones de memoria RAM.

Para la respuesta del PIC a una llamada telefónica se ha empleado la interrupción INT2. Esta alterará el flujo del programa, como todas las demás interrupciones, y

enviara los datos grabados en la tarjeta de memoria. Utilizando para este propósito el modem anteriormente descrito.

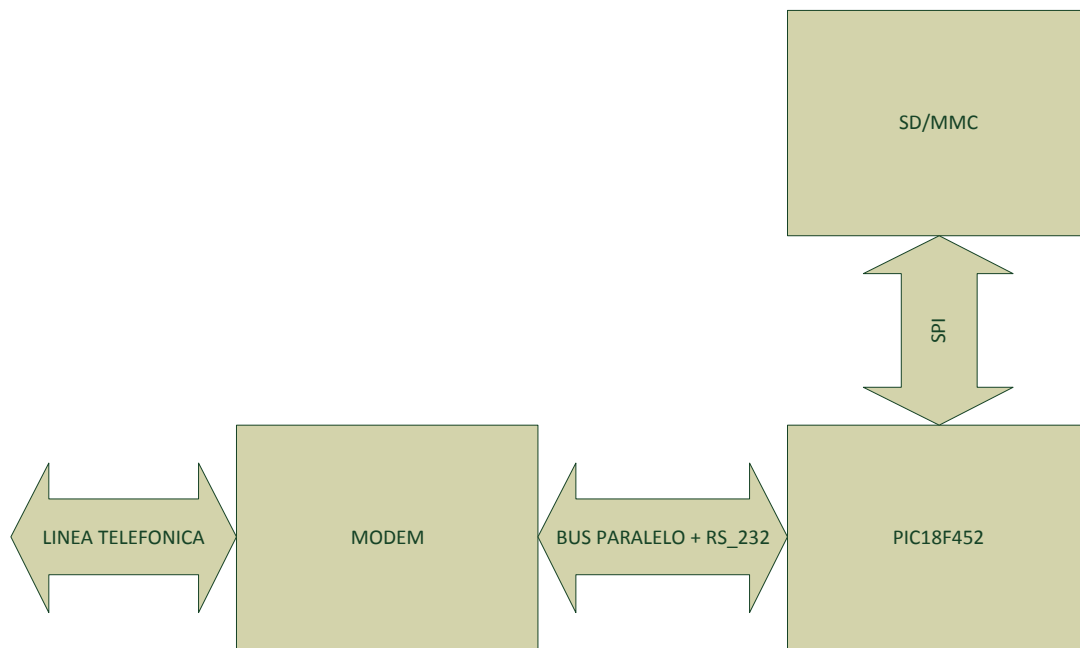


Figura 3.29 Transmisión de datos mediante la línea telefónica.

3.4.2.3 Sistema de visualización

Como se explicó anteriormente el PIC18F452 es quien controla el TOUCH ESCREEN, que se emplea para la visualización de los datos que envía el PIC master, cuando existe una petición de graficar por parte del usuario. Este PIC está configurado como esclavo durante la comunicación SPI.

Se tiene conectado un pin de este PIC a la entrada de la interrupción INT1 del PIC master, para que responda en el instante que se lo necesite. Una vez que se activa la interrupción, el PIC esclavo está esperando hasta que se dé un movimiento para iniciar la visualización del mismo. Y solo se finaliza la comunicación si se pulsa determinada posición en la pantalla GLCD.

A continuación la secuencia de la comunicación SPI.

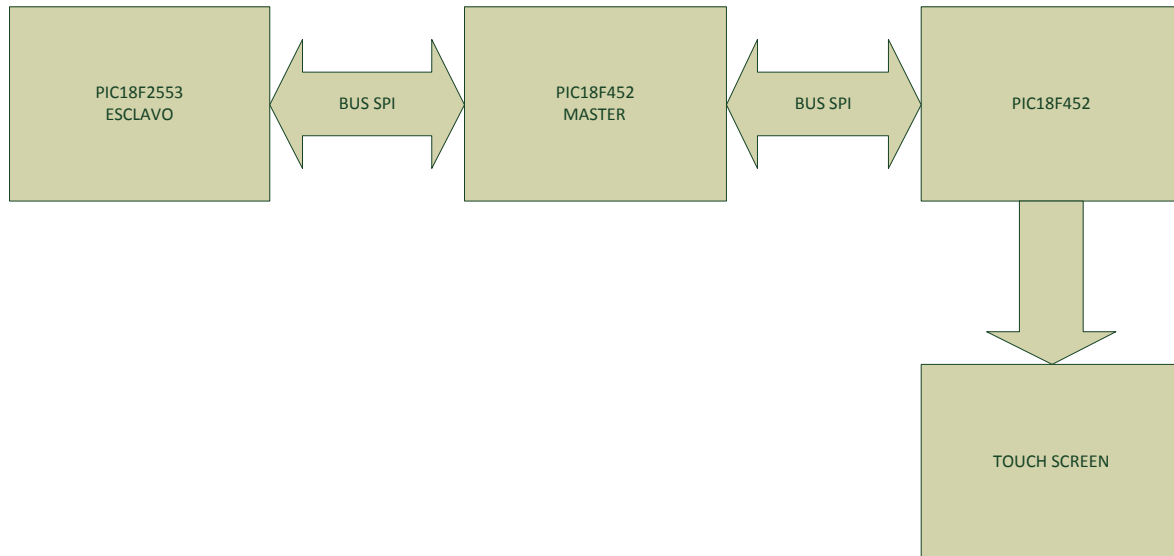


Figura 3.30 Comunicación SPI durante una petición de graficar.

3.4.3 Diagramas de flujo.

3.4.3.1 Sistema de muestreo y amplificación

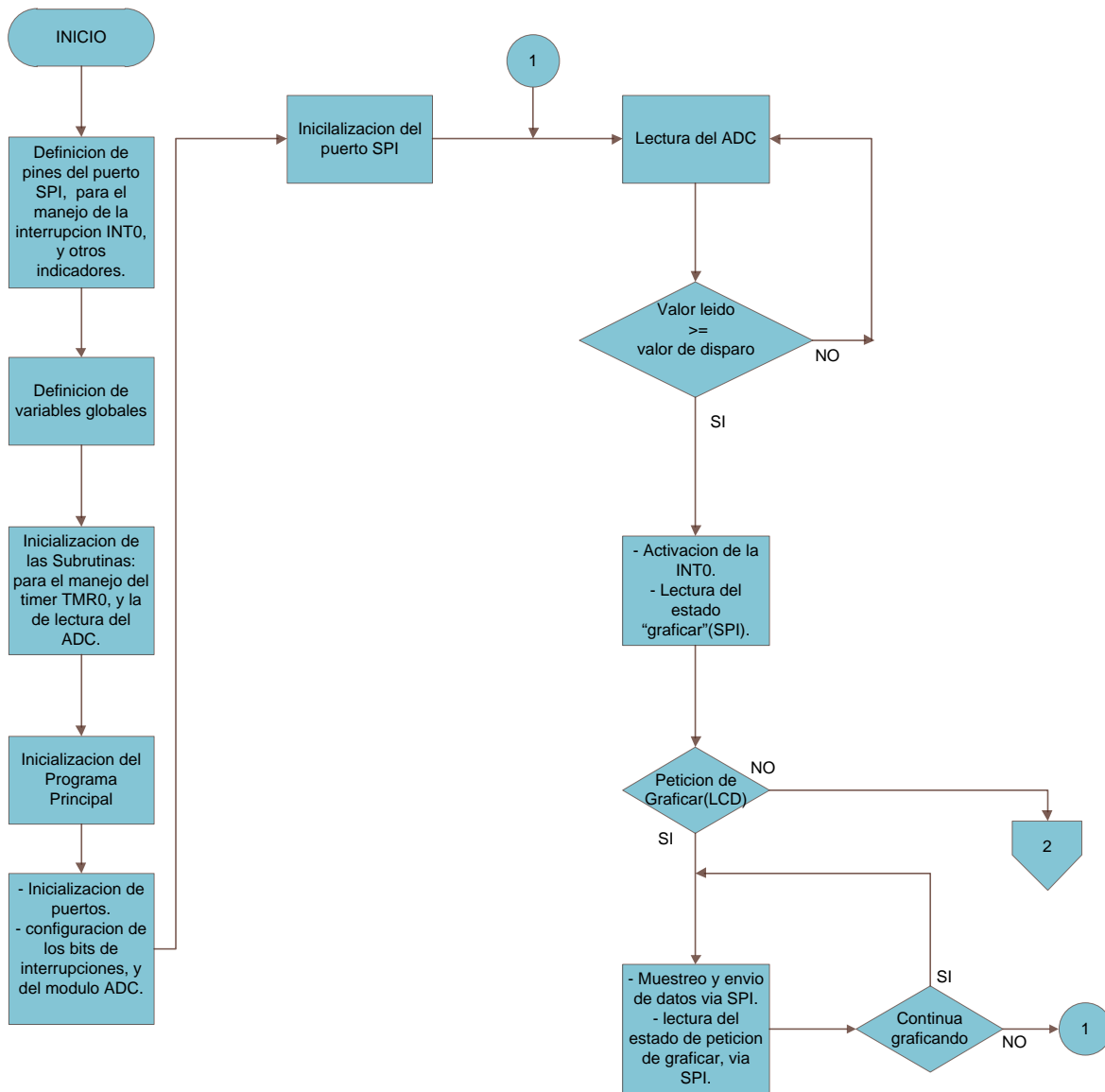


Figura 3.31 Diagrama de flujo del sistema de muestreo y amplificación.

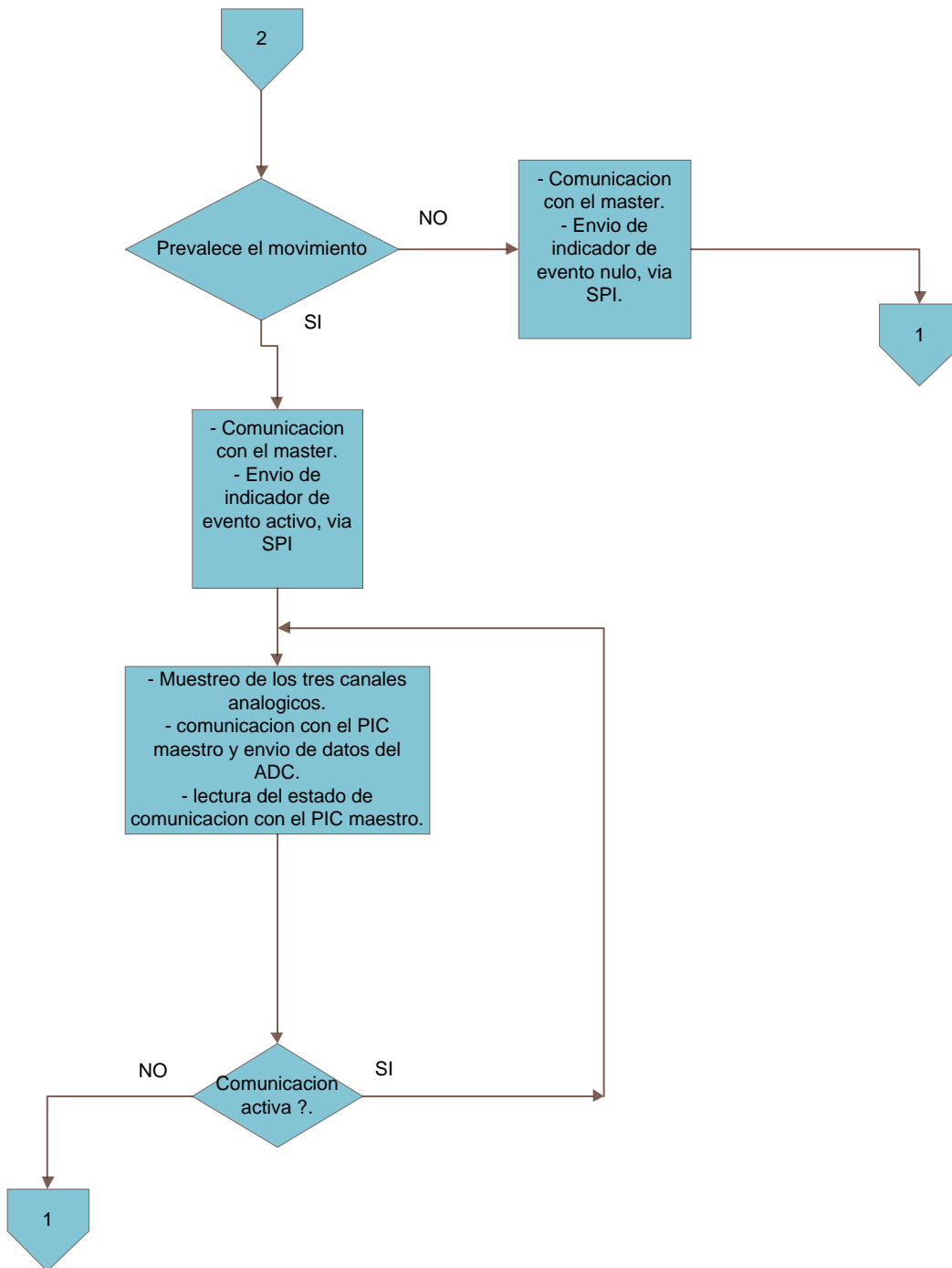


Figura 3.32 Diagrama de flujo del sistema de muestreo y amplificación (continuación).

3.4.3.2 Sistema de almacenamiento y georeferenciación

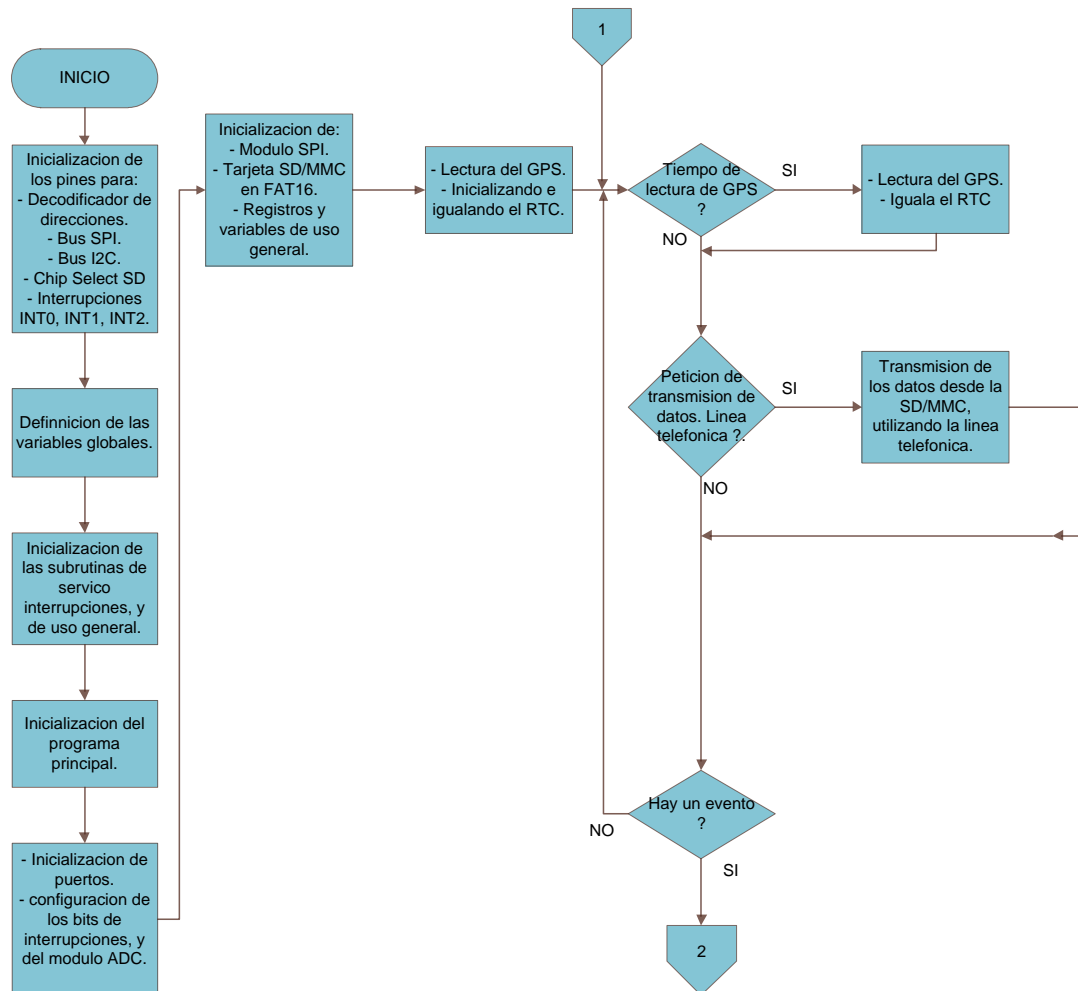


Figura 3.33 Diagrama de flujo del sistema de almacenamiento y georeferenciación.

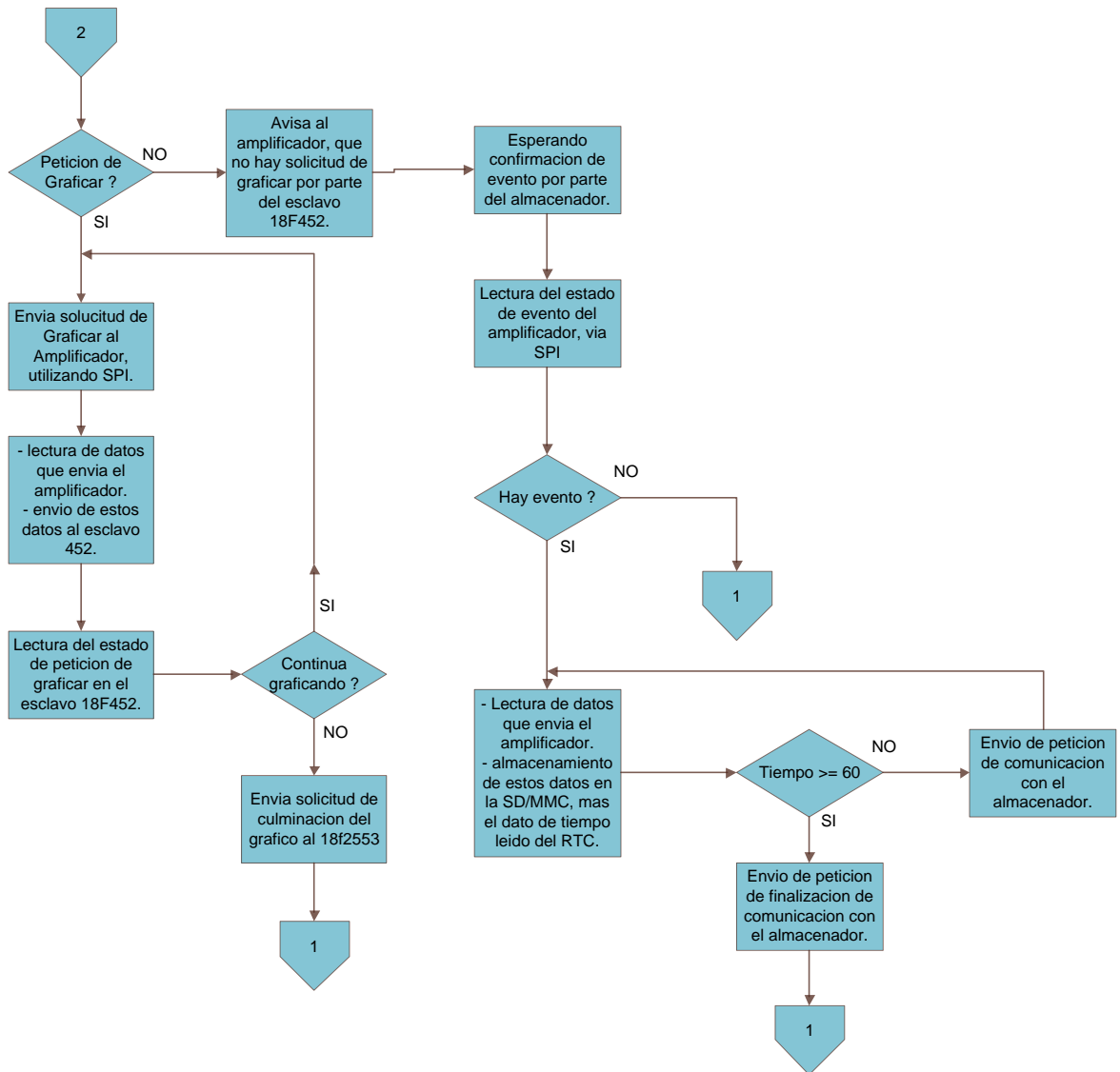


Figura 3.34 Diagrama de flujo del sistema de almacenamiento y georeferenciación (continuación).

3.4.3.3 Sistema de visualización

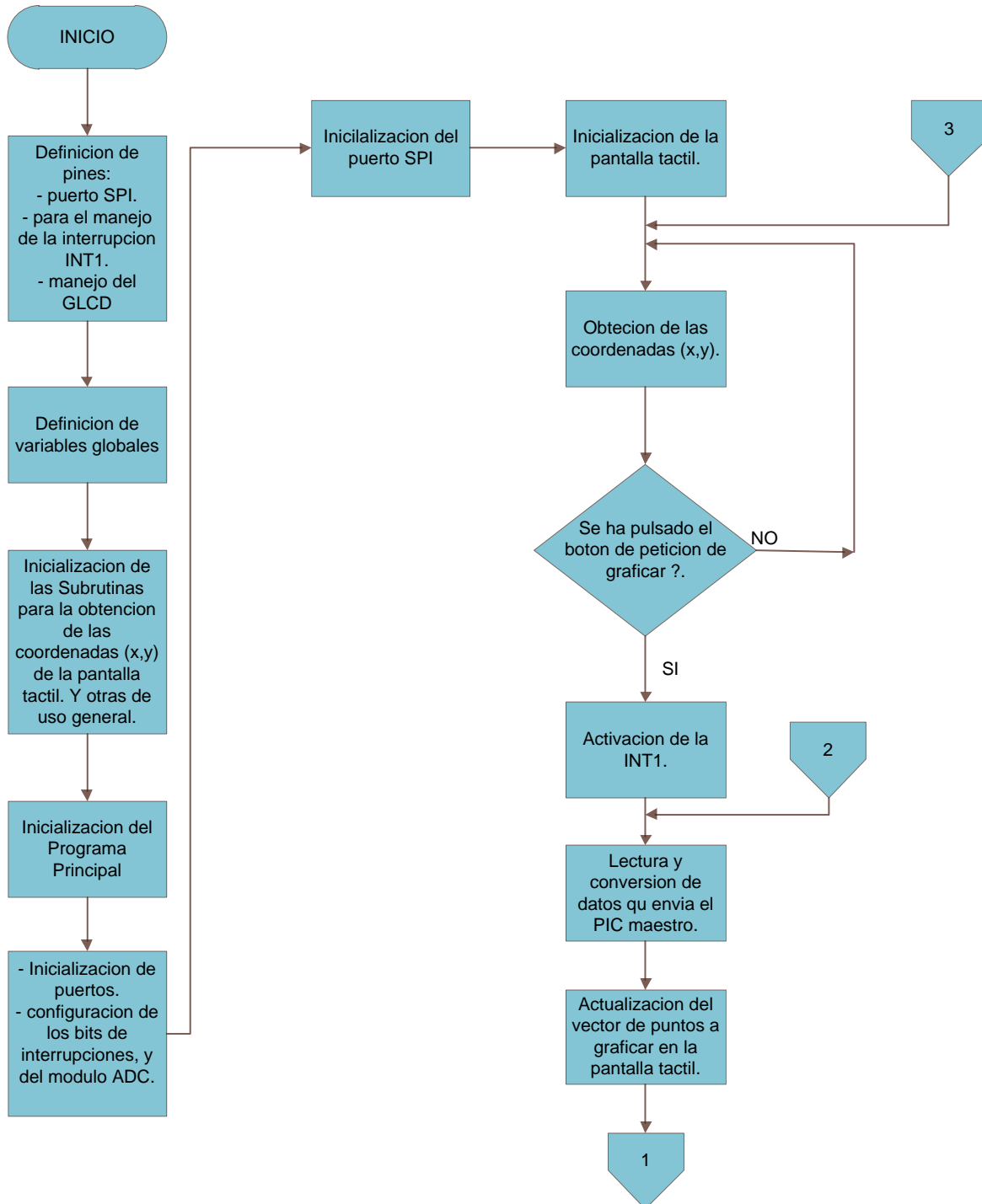


Figura 3.35 Diagrama de flujo del sistema de visualización.

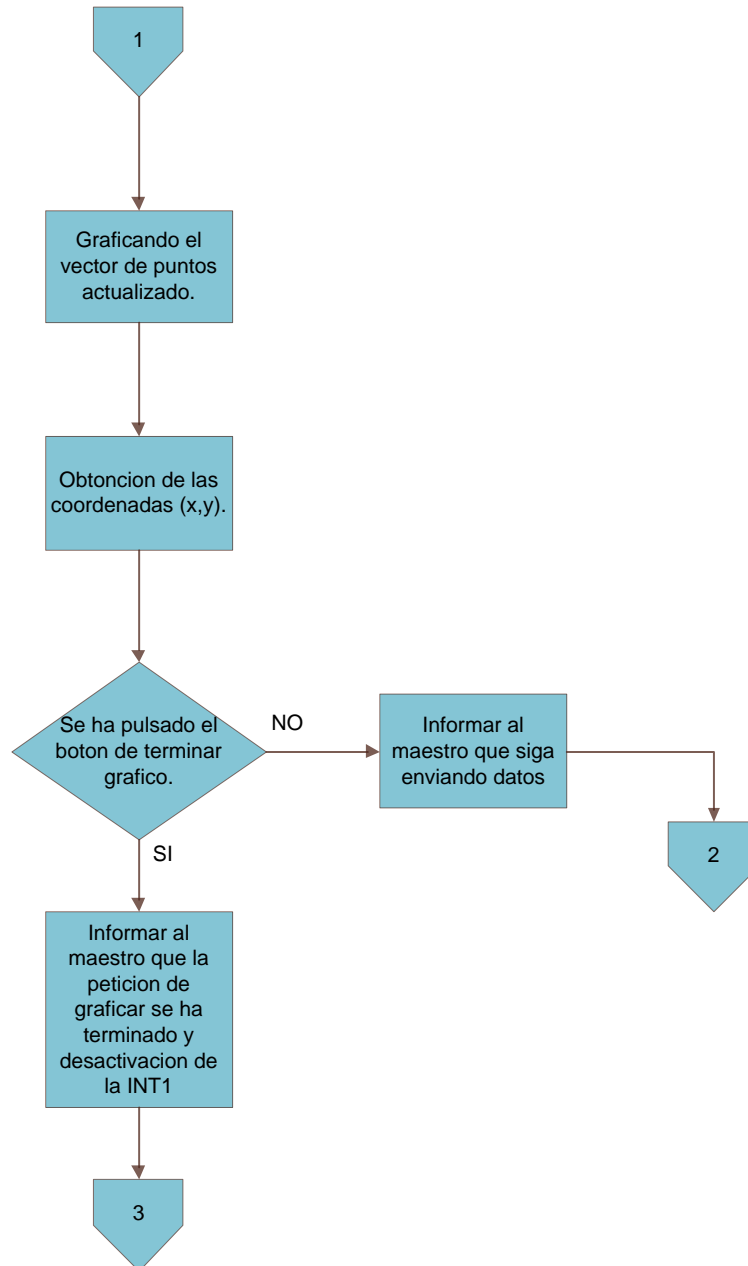


Figura 3.36 Diagrama de flujo del sistema de visualización (continuación).

CAPITULO 4: CARACTERISTICAS Y ALGORITMOS DE LOS MICROPROCESADORES

4.1 CARACTERISTICAS GENERALES DE LOS MICROPROCESADORES

4.1.1 CARACTERÍSTICAS PIC 18F452 y 18f2553

Entre las características más sobresalientes de estos PICs están:

- Cristales de 4 MHz a 10 MHz utilizando un multiplicador de frecuencia PLL.
- Instrucciones de 16 bits con bus de datos de 8 bits.
- Prioridad de interrupciones
- Tres pines para manejo de interrupciones externas.
- Timer 1 de 16 bits, Timer 2 de 8 bits.
- Una memoria FLASH de programa de 32 KBytes.
- Módulos de Captura/Comparación/PWM.
- Módulo de comunicación serial con soporte para RS-485 y RS-232
- Módulo de comunicación SPI
- Módulo de comunicación I2C
- Módulo ADC
- Número de instrucciones: 16384.

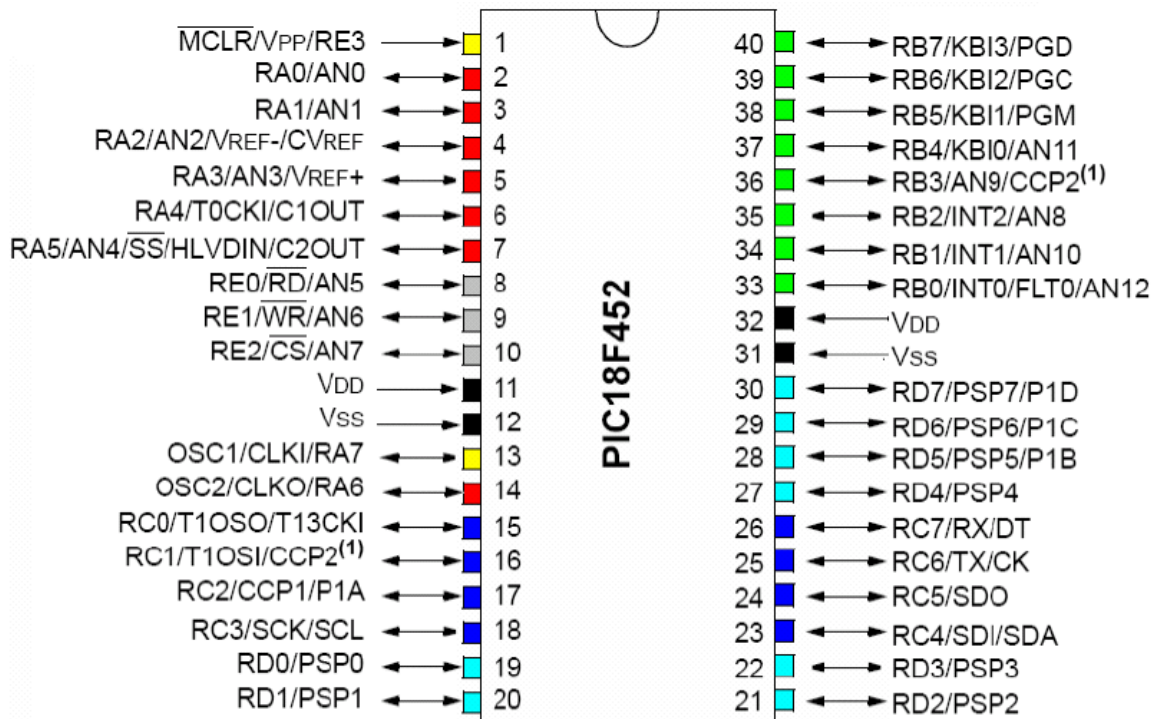


Figura 4.1.1 PIC18f452

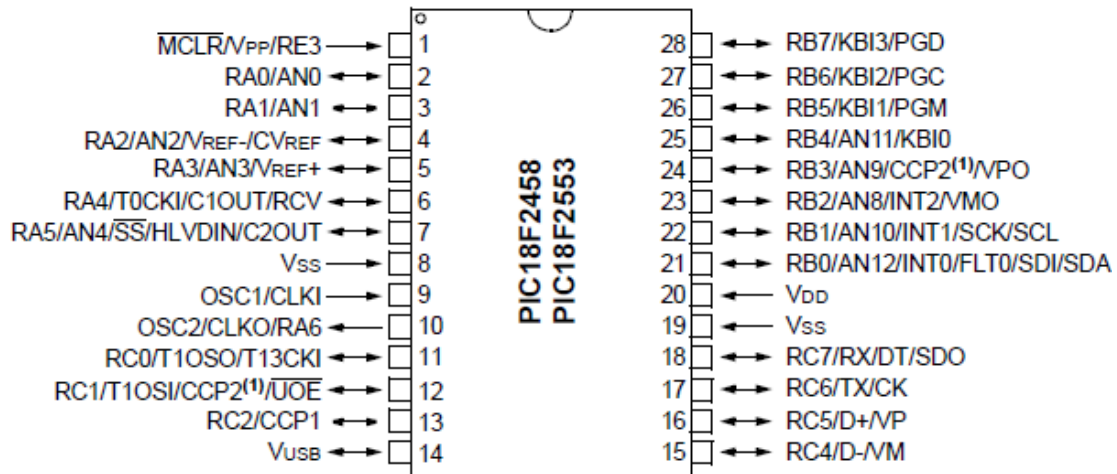


Figura 4.1.2 PIC18f2553

4.1.2 OSCILADOR MODO HS + PLL

En los PICS 18f2553 y 18f452 la configuración del oscilador es la siguiente:

- 18F452 master: HS + PLL, FOSC = 10 MHz.
- 18F452 Esclavo: HS + PLL, FOSC = 7,3728MHz.
- 18F2553 Esclavo: XT + PLL, FOSC = 4MHz.

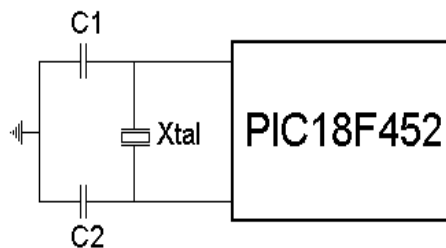


Figura 4.1.3 Conexión externa del oscilador.

Si la programación del cristal se ha realizado en otro modo que no sea a través del PLL, este no se habilita y la fuente de reloj proviene directamente de OSC1 y OSC2.

Esquema del PLL del PIC

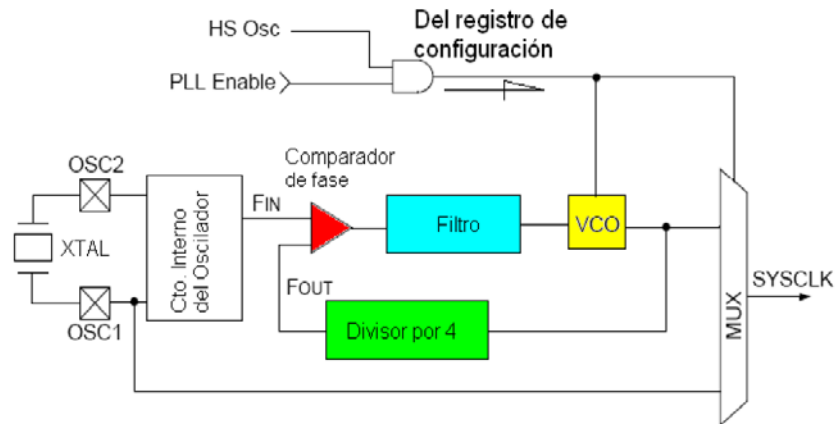
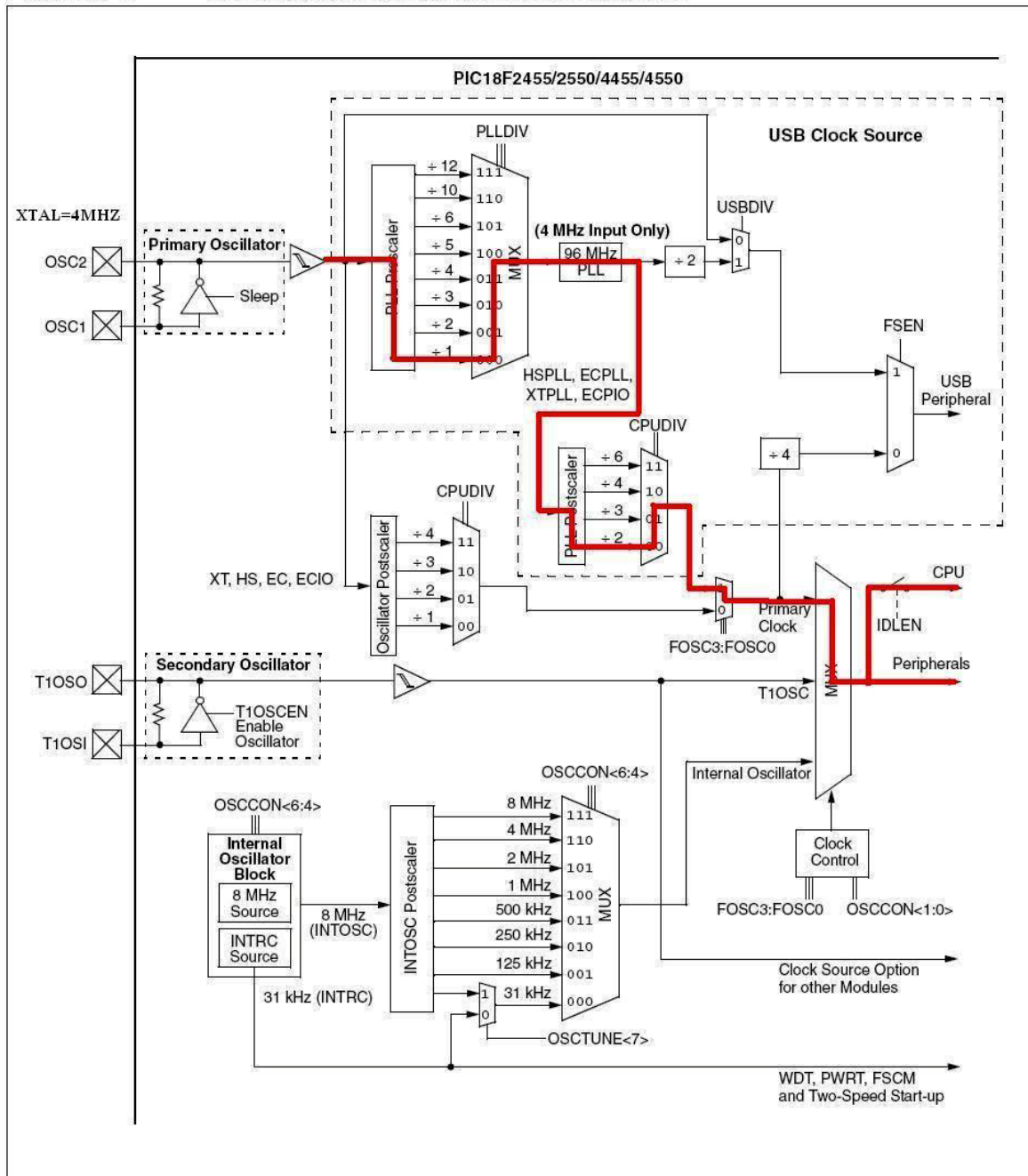


Figura 4.1.4 Configuración del reloj a través del PLL.

Cuando se requiere el multiplicador de frecuencias se establece una secuencia de configuración (ver Figura 4.1.5), con las distintas opciones a lo largo de la misma. Donde la ruta más adecuada queda a elección del programador.

FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM

Figura 4.1.5 Diagrama de configuración del PLL para los PICs: 18F452 y 18F2553.

4.1.3 ORGANIZACIÓN DE LA MEMORIA

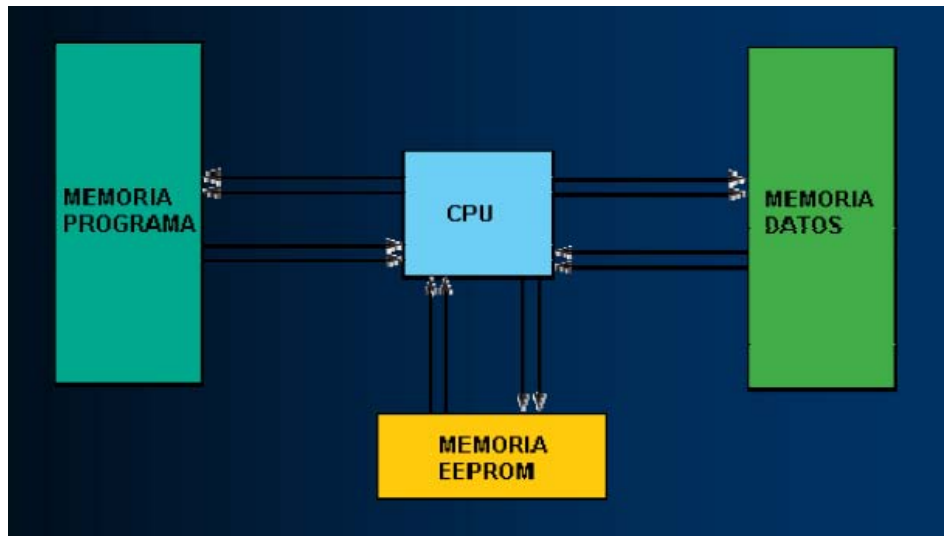


Figura 4.1.6 Organización de la memoria, para los PICs: 18F452 y 18F2553.

Memoria de programa

El PIC18F452 posee 32 KBytes de memoria FLASH de programa, agrupados de 2 bytes para contener instrucciones complejas. Este pic puede almacenar 16 mil instrucciones simples.

PIC18F2553 posee 32 KBytes de memoria FLASH de programa, al igual que el pic18f452 están agrupados de 2 bytes para contener instrucciones complejas. Este pic puede almacenar 16 mil instrucciones.

Memoria de Datos

La memoria de datos del PIC18f452 está dividida en 16 bancos, cada una de 256 bytes. Lo que equivale a tener 1536 bytes de memoria RAM.

La memoria de datos del PIC18f2553 está dividida en 8 bancos, cada una de 256 bytes. Lo que equivale a tener 2048 bytes de memoria RAM.

Memoria EEPROM

En los dos PICs la memoria EEPROM es de 256 bytes.



| | memoria de programa | | memoria de datos | | I/O | 12 bits ADC | MSSP | |
|-------------------|---------------------|-----------------|------------------|--------|-----|-------------|------|------|
| | flash | # instrucciones | SRAM (byte) | EEPROM | | | SP I | I2 C |
| pic18f2553 | 32 K | 16384 | 2048 | 256 | 24 | 10 | si | si |
| pic18f452 | 32 k | 16348 | 4096 | 256 | 40 | 8 | si | si |

Tabla 4.1.1 Característica de los PICS 18f2553 y 18F452

4.1.4 DIAGRAMA DE BLOQUES PIC18F452

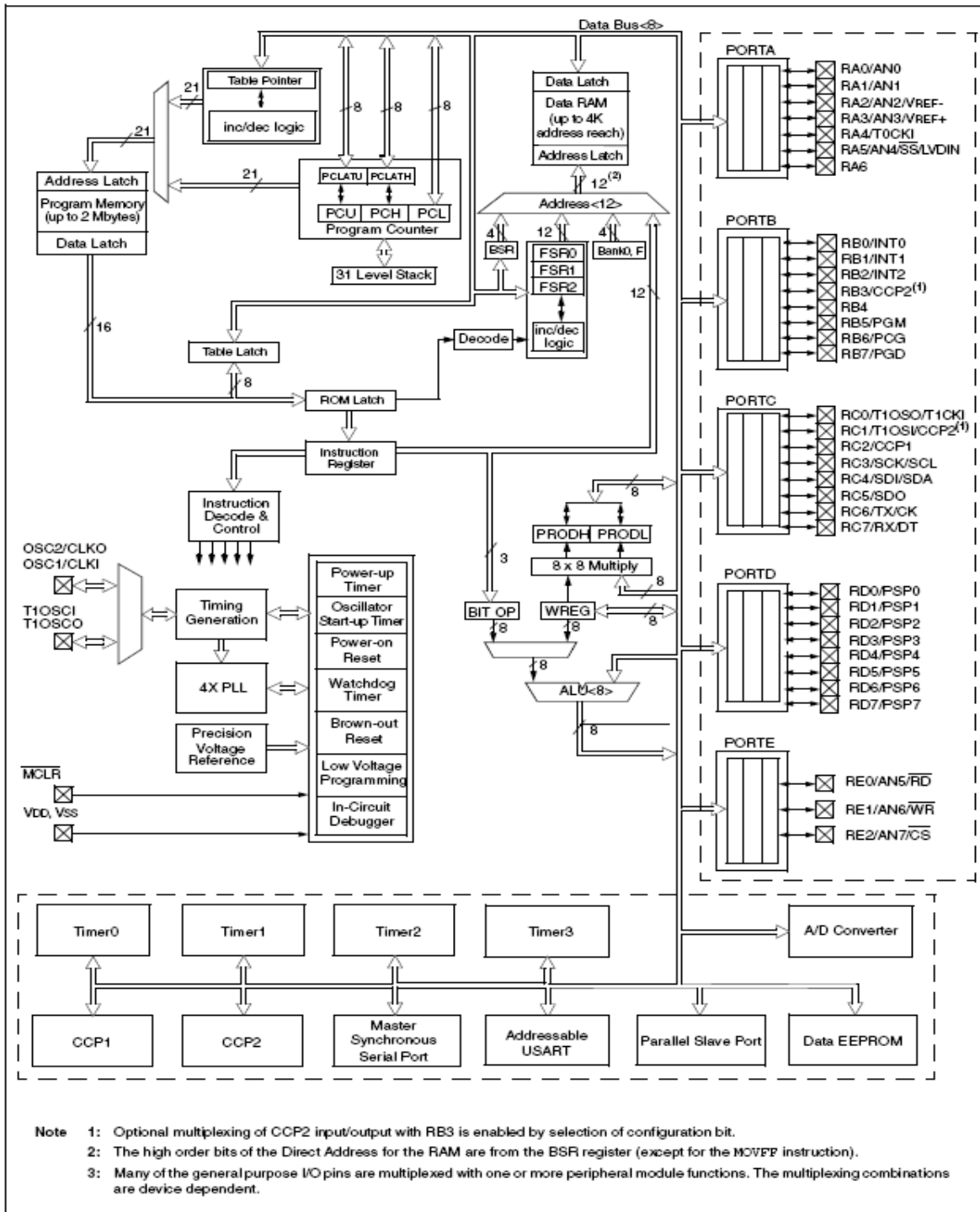


Figura 4.1.7 Diagrama de bloques del PIC18F452.

4.1.5 DIAGRAMA DE BLOQUES PIC18F2553

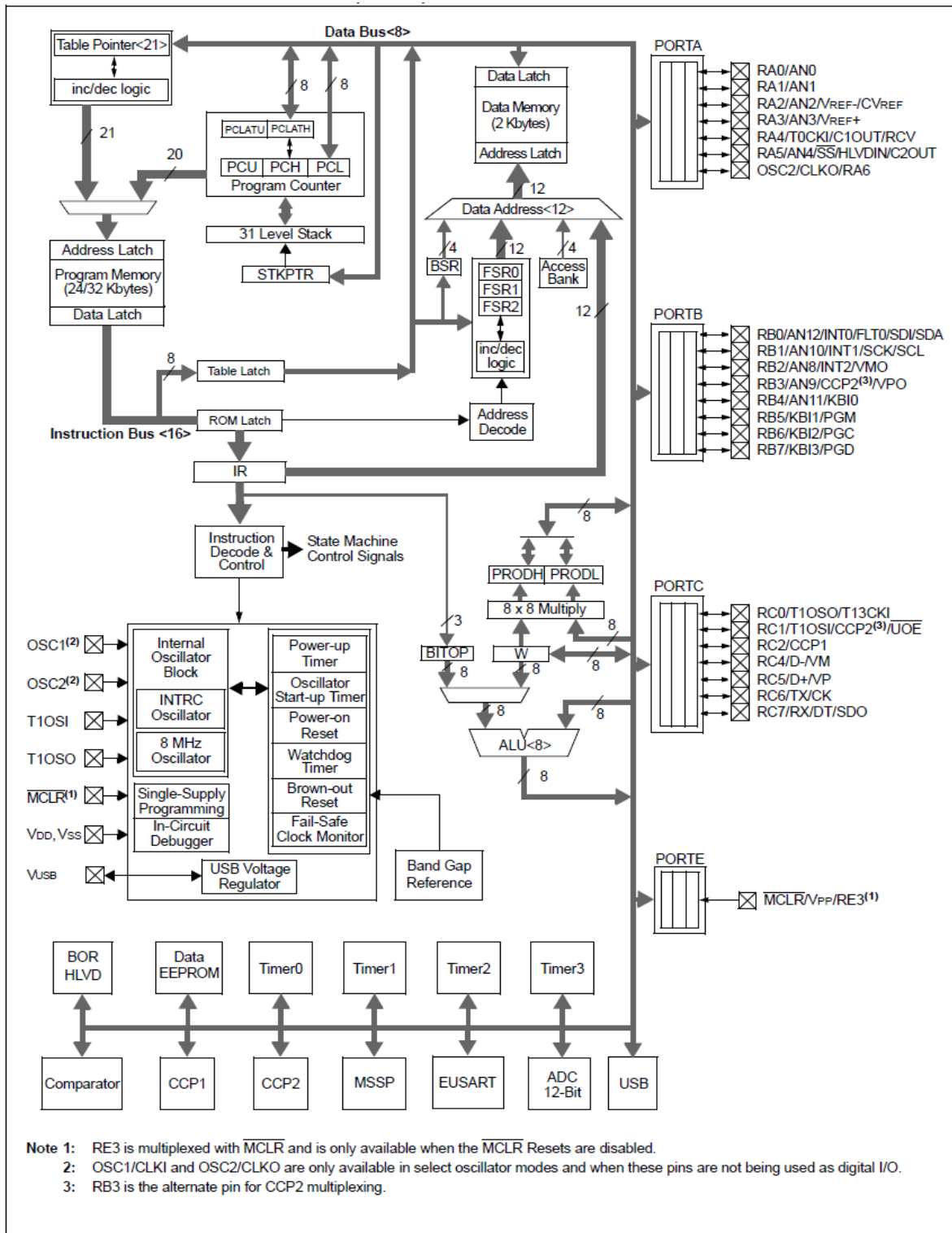


Figura 4.1.8 Diagrama de bloques del PIC18F2553.

4.2 ALGORITMOS EN LOS MICROPROCESADORES

4.2.1 Algoritmo del sistema de almacenamiento y georeferenciación.

4.2.1.1 Configuración de frecuencia de trabajo del microprocesador

Una vez creado el nuevo proyecto en MikroC Pro, vamos a la barra de herramientas del editor, damos clic en **Project**, luego en **Edit Project**, y aparece un cuadro con diferentes tipos de configuraciones para el PIC con el que se está trabajando (PIC18F452).

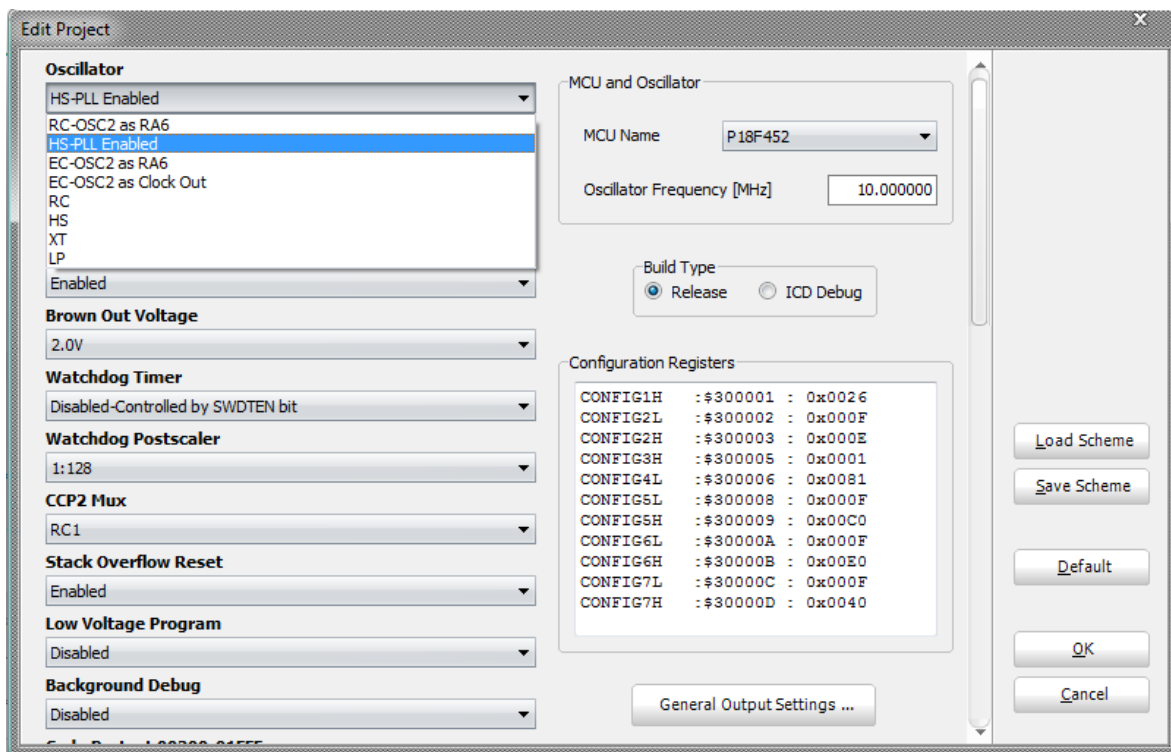


Figura 4.2.1.1 Configuración de la frecuencia de trabajo para el PIC18F452

Según la hoja de características del PIC18F452, la máxima frecuencia a la que este puede trabajar es de 40 MHz, utilizando el PLL con un oscilador de 10 MHz en las entradas **OSC1:OSC2**. Se elige esta configuración, debido a que este PIC controla todos los procesos tanto internos como externos (comunicaciones con otros dispositivos periféricos, almacenamiento, visualización, etc.). Además de que, las exigencias de frecuencia por parte de las señales sísmicas son un

aspecto que se tiene en consideración. Porque es el PIC master quien controla el muestro de los canales del geófono de tres componentes por parte de la tarjeta amplificadora (PIC18F2553 como esclavo en SPI).

4.2.1.2 Frecuencia de muestreo de las señales del Geófono

La frecuencia mínima requerida por el teorema de Nyquist para el muestreo de las señales del geófono. Y tomando en cuenta sus características de respuesta:

$$F_s = 2 * F_{max_{geofono}}$$

La mayoría de los geófonos y acelerógrafos actuales tienen un amplio rango de frecuencias las mismas que oscilan entre 0 - 100 Hz, y más de 100 Hz(típicamente los acelerógrafos) para algunos cuyas sollicitaciones en frecuencia son muy elevadas.

El filtro antialiasing que posee la tarjeta amplificadora, está diseñado para una frecuencia de corte de 500 Hz. Esto se debe a que; al sistema que actualmente funciona con un geófono de tres componentes, posteriormente se le implementara un acelerógrafo, donde se requiere un espectro de tales características para suplir las necesidades en frecuencia.

Pero tomando en cuenta: el rango de frecuencias de respuesta de los geófonos, los requerimientos de la **RSA** en cuanto al procesamiento de señales sísmicas, y las limitaciones de velocidad de ejecución de las instrucciones por parte de los PICs. Se eligen los 20 Hz como la máxima frecuencia que se tendrá en las entras del ADC de la tarjeta amplificadora. Entonces para la digitalización de señales se tiene:

$$F_s = 2 * 20 = 40 \text{ Hz}$$

Para tener una mejor caracterización de las señales, se opta por elegir 200 Hz por canal del geófono:

$$F_{SCH} = 10 * 20 = 200\text{Hz}$$

4.2.1.3 Configuración de la velocidad del bus SPI

Este PIC controla toda la comunicación SPI, por lo tanto se inicializa:

- PIC configurado como master.
- SCK es común para todos los dispositivos esclavos.

Según las opciones de configuración de reloj que presentan las librerías de **MikroC Pro**, se tiene en baja velocidad:

$$SCK = \frac{Fosc}{64} = \frac{40MHz}{64} = 625KHz$$

Y alta velocidad:

$$SCK = \frac{Fosc}{16} = \frac{40MHz}{16} = 2.5MHz$$

Estas son las dos velocidades que se utiliza para la comunicación SPI durante la programación. Tomando en consideración eso sí, las tasas de transferencia que los dispositivos esclavos pueden soportar.

Cuando se presenta un sismo de determinada magnitud el amplificador se comunica con el almacenador, obedeciendo el diagrama de tiempos de la figura **Figura 4.2.1.2**.

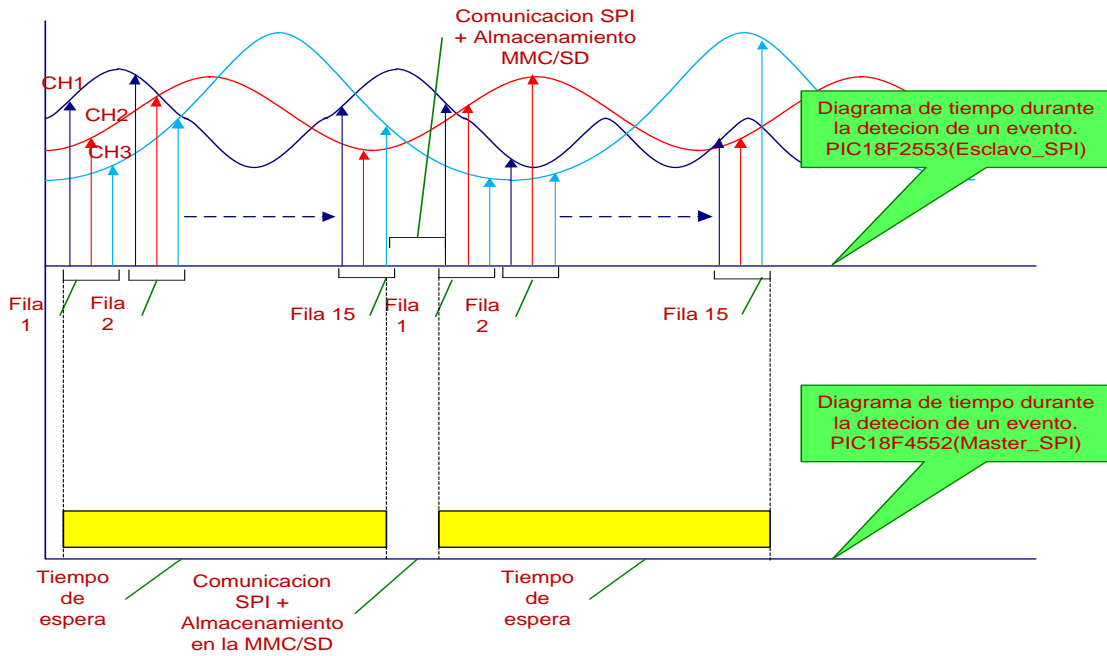


Figura 4.2.1.2 Diagrama de tiempo de la comunicación SPI Amplificador_Almacenador.



4.2.1.4 Algoritmo

//////////////////////LINEAS DE ENTRADA PARA EL DECODIFICADOR////////////////////

| CS2 | CS1 | Activación |
|-----|-----|--------------|
| 0 | 0 | Amplificador |
| 0 | 1 | Visualizador |
| 1 | 0 | Ninguno |
| 1 | 1 | Ninguno |

//////////////////////LINEAS DE ENTRADA PARA EL DECODIFICADOR////////////////////

```

sbit CS1 at RC0_bit;
sbit CS1_Direction at TRISC0_bit;
sbit CS2 at RC1_bit;
sbit CS2_Direction at TRISC1_bit;

//////////////////////PINES ASIGNADOS PARA BUS SPI////////////////////

sbit SCK at RC3_bit;
sbit SCK_Direction at TRISC3_bit;
sbit SDI at RC4_bit;
sbit SDI_Direction at TRISC4_bit;
sbit SDO at RC5_bit;
sbit SDO_Direction at TRISC5_bit;
sbit LED at RB7_bit;
sbit LED_Direction at TRISB7_bit;
sbit Mmc_Chip_Select at RC2_bit;
sbit Mmc_Chip_Select_Direction at TRISC2_bit;

```

//////////////////////PINES ASIGNADOS PARA EL BUS I2C////////////////////

```

sbit Soft_I2C_Scl      at Rb3_bit;
sbit Soft_I2C_Sda     at Rb4_bit;

```



```
sbit Soft_I2C_Scl_Direction at TRISb3_bit;
sbit Soft_I2C_Sda_Direction at TRISb4_bit;
/////////////////////////////////LINEAS DE INTERRUPCIONES/////////////////////////////////
sbit INT0 at RB0_bit;
sbit INT0_Direction at TRISB0_bit;
sbit INT1 at RB1_bit;
sbit INT1_Direction at TRISB1_bit;
sbit INT2 at RB2_bit;
sbit INT2_Direction at TRISB2_bit;
/////////////////////////////////INICIALIZACION DE VARIABLES GENERALES/////////////////////////////////
long i,k,j,u;
int datx,evento,estado,graficar,exit,tiempo,modem,timeGPS;
static char buf[690];
long yr;
char sismo[11]={'e','v','e','n',0x30,0x30,0x30,',' ,'x','l','s'};
/////////////////////////////////INICIALIZACION DE LAS VARIABLES DE LECTURA DEL GPS/////////////////////////////////
int hora,seg,minu,diaa,mess,anoo;          // variables de lectura del GPS
char day,date,month,year,minutes,hours,seconds;// Valores de escrituta/lectura
//                                     del reloj de tiempo Real(RTC)
/////////////////////////////////
void leergps(void){// lectura del GPS via RS_232 hasta encontrar la cadena GPRMT.
k = 0;
do{ // lectura hasta encontrar al comando GPRMT. Una vez encontrado,
continuar: // convertimos y almacenamos los respectivos parámetros de tiempo.
while(UART1_Data_Ready()!=1);
buf[0]=UART1_Read();
if(buf[0]=='$'){ while(UART1_Data_Ready()!=1);
                buf[1]=UART1_Read();
                if(buf[1] =='G'){ while(UART1_Data_Ready()!=1);
                                buf[2]=UART1_Read();
                                if(buf[2] =='P'){ while(UART1_Data_Ready()!=1);
```



```
    buf[3]=UART1_Read();
if(buf[3] =='R'){ while(UART1_Data_Ready()!=1);
    buf[4]=UART1_Read();
    if(buf[4]== 'M'){ while(UART1_Data_Ready()!=1);
        buf[5]=UART1_Read();
        if(buf[5]== 'C'){
            k = 1;
            for(i=6;i<=63;i++){
                while(UART1_Data_Ready()!=1);
                buf[i]=UART1_Read();
            }
            buf[73]=buf[7];
            buf[74]=buf[8];
            buf[75]=0x00;
            buf[76]=buf[9];
            buf[77]=buf[10];
            buf[78]=0x00;
            buf[79]=buf[11];
            buf[80]=buf[12];
            buf[81]=0x00;
            hora=atoi(&buf[73]);

            minu=atoi(&buf[76]);

            seg=atoi(&buf[79]);
            j=0;
            i=0;
            for(i=6;i<=63;i++){
                if(buf[i]==','){
                    j=j+1;
                    if(j==9){
                        buf[i+9]=0x00;
```




```

                                                                    buf[i+8]=buf[i+6];
                                                                    buf[i+7]=buf[i+5];
                                                                    buf[i+6]=0x00;
                                                                    buf[i+5]=buf[i+4];
                                                                    buf[i+4]=buf[i+3];
                                                                    buf[i+3]=0x00;
                                                                    diaa=atoi(&buf[i+1]);

mess=atoi(&buf[i+4]);

anoo=atoi(&buf[i+7]);

                                                                    yr=2000+anoo;
                                                                    break;}
                                                                    }
                                                                    }
                                                                    }
                                                                    else {goto continuar;}}
                                                                    else {goto continuar;}}
                                                                    else {goto continuar;}}
                                                                    else {goto continuar;}}
                                                                    else {goto continuar;}
}while(k = 0);
}

void iniciar_MMC(void){//Creación de un nuevo archivo con el nombre contenido
// en el vector llamado sismo.
Mmc_Fat_Assign(sismo,0xA0);
Mmc_Fat_Rewrite();
Mmc_Fat_Write(" CHANEL1\t CHANEL2\t CHANEL3\t \r\n",40);
Mmc_Fat_Append();
Mmc_Fat_Set_File_Date(yr,month,date,hours,minutes,0);
Mmc_Fat_Append();
```



```
Mmc_Fat_Write("\r\n",2);
}
void igualar_RTC(){// Grabación de los parámetros de tiempo en el RTC
// estos son los leídos en la rutina del GPS.
seconds = Dec2Bcd(seg);
minutes = Dec2Bcd(minu);
hours = Dec2Bcd(hora);
date = Dec2Bcd(diaa);
month = Dec2Bcd(mess);
year = Dec2Bcd(anoo);
Soft_I2C_Init(); // Initialize Soft I2C communication
Soft_I2C_Start(); // Issue start signal
Soft_I2C_Write(0xD0); // Address PCF8583, see PCF8583 datasheet
Soft_I2C_Write(0); // Start from address 2
Soft_I2C_write(seconds); // Issue repeated start signal
Soft_I2C_Write(minutes);
Soft_I2C_write(hours); // Issue repeated start signal
Soft_I2C_Write(0x00);
Soft_I2C_write(date); // Issue repeated start signal
Soft_I2C_Write(month);
Soft_I2C_write(year); // Issue repeated start signal
Soft_I2C_write(0x00);
Soft_I2C_Stop();
}
void leer_RTC(){ // Lectura de los parámetros de tiempo desde el RTC.
Soft_I2C_Start();
Soft_I2C_Write(0xD0);
Soft_I2C_Write(0x00);
Soft_I2C_Stop();
Soft_I2C_Start();
Soft_I2C_Write(0xD1); // Address PCF8583 for reading R/W=1
```



```
seconds = Soft_I2C_Read(1); // Read seconds byte
minutes = Soft_I2C_Read(1); // Read minutes byte
hours = Soft_I2C_Read(1); // Read hours byte
day = Soft_I2C_Read(1); // Read seconds byte
date = Soft_I2C_Read(1); // Read minutes byte
month = Soft_I2C_Read(1); // Read hours byte
year = Soft_I2C_Read(0);
Soft_I2C_Stop();
seconds= Bcd2Dec16(seconds);
minutes= Bcd2Dec16(minutes);
hours= Bcd2Dec16(hours);
date= Bcd2Dec16(date);
month= Bcd2Dec16(month);
year= Bcd2Dec16(year);
yr=2000+year;
}
void grabar_datos_MMC(void){// lectura de datos vía SPI. y escritura en la SD/MMC
for(i=0;i<=689;i++){
CS1=0;
CS2=0;
SPI1_Write(0x00);
CS1=1;
CS2=1;
buf[i]=SSPBUF;
}
CS1=1;
CS2=1;
Mmc_Fat_Append();
Mmc_Fat_Write(buf,690);
}
void interrupt(){// Rutina de servicio de las interrupciones de prioridad alta
```



```
//          INT0 y la del TMR0(timer cero)
INTCON.INT0IE=0;
if(INTCON.INT0IF==1){
    INTCON.INT0IF=0;
    T0CON.TMR0ON=0;
    LED=1;
    TMR0L=0x6A;
    TMR0H=0x67;
    evento=1;
    LED=0;
}
if(INTCON.TMR0IF==1){
    INTCON.TMR0IF=0;
    if(evento==0){
        timeGPS++;
    }
    tiempo++;
    TMR0L=0x6A;
    TMR0H=0x67;
}
INTCON.TMR0IE=1;
INTCON.INT0IE=1;
}

void interrupt_low() { // Rutina de servicio de las interrupciones de prioridad baja
//          INT1 e INT2.
INTCON3.INT1IE=0; // HABILITATA LA INT1.
//INTCON3.INT2IE=0; // HABILITATA LA INT2.
if (INTCON3.INT1IF==1) {
    INTCON3.INT1IF=0;
    LED=1;
    graficar=1;
}
```



```
}
LED=0;
INTCON3.INT1IE=1;// HABILITITA LA INT1.
//INTCON3.INT2IE=1;//HABILITITA LA INT2.
}
void contar_evento(void){// Rutina para el conteo de eventos almacenados en la SD/MMC.
sismo[6]=sismo[6]+1;
if(sismo[6]>'9'){
sismo[6]=0x30;
sismo[5]=sismo[5]+1;
if(sismo[5]>'9'){
sismo[5]=0x30;
sismo[4]=sismo[4]+1;
}
}
}
////////////////////////////////PROGRAMA PRINCIPAL////////////////////////////////
void main(void){// INICIALIZACION DE LOS PUERTOS Y CONFIGURACION DE
LOS REGISTROS
// GENERALES.
ADCON1=0b10000111;
TRISC.F6=0;////////Línea para la escritura del GPS vía RS_232////////
TRISC.F7=1;////////Línea para la lectura del GPS vía RS_232////////
INT0_Direction=1;
INT1_Direction=1;
INT2_Direction=1;
CS1_Direction=0;
CS2_Direction=0;
SCK_Direction=0;
SDI_Direction=1;
SDO_Direction=0;
```




```
yr=0;
RCSTA.SPEN=1;
RCSTA.CREN=1;
UART1_Init(4800); // inicializando el módulo RS_232 a 4800 bps.
TXSTA.BRGH=0;

//wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

//////////Inicializacion del bus SPI//////////

SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64,_SPI_DATA_SAMPLE_MIDDLE,_S
PI_CLK_IDLE_LOW,_SPI_LOW_2_HIGH);

while(Mmc_Fat_Init()!=0)LED=0; // inicializacion de la SD en FAT 16

SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV16,
_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_LOW,_SPI_LOW_2_HIGH);

//////////

LED=0;

timeGPS=0;

TMR0L=0x6A; // carga los registros del TIMER0 para establecer un conteo cada segundo.
TMR0H=0x67;
sismo[6]=0x30;
sismo[5]=0x30;
sismo[4]=0x30;

leergps(void);
igualar_RTC(void);
INTCON.TMR0IE=1;
TOCON.TMR0ON=1;

now:

if(timeGPS>=86400){//Es tiempo de lectura del GPS ?. el GPS se lee cada 24 horas.
LED=~LED;
leergps(void);
igualar_RTC(void);
timeGPS=0;
```



TMROL=0x6A; // carga los registros del TIMERO para establecer un conteo cada segundo.

TMR0H=0x67;

}

tiempo=0;

if(evento==1){ // si está colocada la bandera de detección de un evento,

// es establece la comunicación SPI Amplificador - Almacenador

// y se guardan los datos en la tarjeta SD/MMC.

if(graficar==1){// si está colocada la bandera de petición de graficar,

//se establece la comunicación SPI: Amplificador - Almacenador - Visualizador.

////////////////////////////////////

INTCON.INT0IE=0; // interrupcion INT0.

INTCON.TMR0IE=0;

T0CON.TMR0ON=0;

////////////////////////////////////

evento=0;

exit=0x00;

while(exit==0x00){// permanece dentro de esta rutina mientras

// no cambie el indicador exit.

delay_ms(35);

for(i=0;i<=689;i++){//comunicación con el esclavo2553. recibe 15 datos.

CS1=0;

CS2=0;

SPI1_Write(0x00);

CS1=1;

CS2=1;

buf[i]=SSPBUF;

}

CS1=1;

CS2=1;

CS1=0;



```
    CS2=0;
    SPI1_Write(0x01); //envío estado de seguir con la comunicación SPI al 2553.
    CS1=1;
    CS2=1;
    for(i=0;i<=689;i++){ //envío 15 datos al esclavo452.
CS1=1;
        CS2=0;
        SPI1_Write(buf[i]);
        CS1=1;
CS2=1;
        datx=SSPBUF;
    }
    CS1=1;
    CS2=1;
    delay_ms(1650);
    CS1=1;
    CS2=0;
    SPI1_Write(0x00); //comunicación con el esclavo452.
    CS1=1;
    CS2=1;
    exit=SSPBUF; // lee estado del test. Enviado por el esclavo452.
    }
    delay_ms(35);
for(i=0;i<=689;i++){
    CS1=0;
        CS2=0;
    SPI1_Write(0x00); //comunicación con el esclavo2553. recibe 15 datos.
    CS1=1;
        CS2=1;
        buf[i]=SSPBUF;
    }
}
```



```
CS1=1;
CS2=1;
CS1=0;
CS2=0;
SPI1_Write(0x00);//envío el estado de terminar la comunicación SPI al 2553.
CS1=1;
CS2=1;
////////////////////////////////////
INTCON.TMR0IE=1;
T0CON.TMR0ON=1;
INTCON.INT0IE=1;
timeGPS=0;
////////////////////////////////////
evento=0;
graficar=0;
goto now;
}
contar_evento(void);
leer_RTC(void);
iniciar_MMC(void);
INTCON.INT0IE=0; // interrupcion INT0.
INTCON.TMR0IE=1;
T0CON.TMR0ON=0;

T0CON.TMR0ON=1;
while(tiempo<60){// permanece grabando durante 60 segundos.
// (Comunicación SPI Amplificador - Almacenador)
delay_ms(35);
grabar_datos_MMC(void);//se comunica con el PIC18F2553 y guarda los datos en la
SD
CS1=0;
```



```
CS2=0;
SPI1_Write(0x01);// envío de estado de seguir muestreando al Esclavo 2553.
CS1=1;
CS2=1;
}
delay_ms(35);
grabar_datos_MMC(void);//se comunica con el PIC18F2553 y guarda los datos en la SD
CS1=0;
CS2=0;
SPI1_Write(0x00);//envío de estado de terminar el muestreo al Esclavo 2553.
CS1=1;
CS2=1;
T0CON.TMR0ON=0;
INTCON.TMR0IE=0;
tiempo=0;
evento=0;
timeGPS=0;
INTCON.TMR0IE=1;
T0CON.TMR0ON=1;
INTCON.INT0IE=1;
}
goto now;
}
```

```
*****
*****
*****
```

Debido a las limitaciones de memoria RAM del PIC18F452, no se puede incluir la subrutina de transmisión de datos a través de la línea telefónica. Por esta razón se la implemento y prubo de forma individual. A continuación se presenta el algoritmo para el manejo del modem telefónico.

```
*****
*****
```



//////////////////líneas de interrupcion////////////////////////

```
sbit INT0 at RB0_bit;
sbit INT0_Direction at TRISB0_bit;
sbit INT1 at RB1_bit;
sbit INT1_Direction at TRISB1_bit;
sbit INT2 at RB2_bit;
sbit INT2_Direction at TRISB2_bit;
```

////////////////////////líneas para control del modem////////////////////////

```
sbit ale at Ra0_bit;
sbit wri at Ra1_bit;
sbit read at Ra2_bit;
sbit rele at Ra3_bit;
sbit cs at Ra5_bit;
sbit led at Rb7_bit;
```

////////////////////////direcciones de las líneas de control////////////////////////

```
sbit ale_Direction at TRISa0_bit;
sbit wri_Direction at TRISa1_bit;
sbit read_Direction at TRISa2_bit;
sbit rele_Direction at TRISa3_bit;
sbit cs_Direction at TRISa5_bit;
sbit led_Direction at TRISb7_bit;
```

////////////////////////LINEAS PARA EL decodificador////////////////////////

```
sbit CS1 at RC0_bit;
sbit CS1_Direction at TRISC0_bit;
sbit CS2 at RC1_bit;
sbit CS2_Direction at TRISC1_bit;
```

////////////////////////BUS SPI////////////////////////

```
sbit SCK at RC3_bit;
sbit SCK_Direction at TRISC3_bit;
sbit SDI at RC4_bit;
sbit SDI_Direction at TRISC4_bit;
```



```
sbit SDO at RC5_bit;
sbit SDO_Direction at TRISC5_bit;
sbit Mmc_Chip_Select at RC2_bit;
sbit Mmc_Chip_Select_Direction at TRISC2_bit;
//////////////////////////////////GENERALES//////////////////////////////////
unsigned long i,size,j,tiempo,exist,tamano;
static char buf2[690];
static char buf1[42];
static char buf3[4];// indicador de eventos, # de evento.
char sismo[11]={'e','v','e','n',0x30,0x30,0x30,',' ,'\x','l','s'};
char text[11];
int modem;
//////////////////////////////////
void interrupt(){
INTCON.TMR0IE=0;
if(INTCON.TMR0IF==1){
    INTCON.TMR0IF=0;
    tiempo++;
    TMR0L=0xF6;
    TMR0H=0xFF;
}
INTCON.TMR0IE=1;
}
void contar_evento(void){
sismo[6]=sismo[6]+1;
if(sismo[6]>'9'){
sismo[6]=0x30;
sismo[5]=sismo[5]+1;
if(sismo[5]>'9'){
sismo[5]=0x30;
sismo[4]=sismo[4]+1;
```



```
}  
}  
}  
//////////////////////////////////interrupciones de baja prioridad  
void interrupt_low() {  
INTCON3.INT1IE=0; //DESAHABILITITA LA INT1.  
INTCON3.INT2IE=0; //DESAHABILITITA LA INT2.  
if (INTCON3.INT1IF==1) {  
INTCON3.INT1IF=0;  
}  
if (INTCON3.INT2IF==1) {  
INTCON3.INT2IF=0;  
modem=1;  
}  
INTCON3.INT1IE=0;//HABILITITA LA INT1.  
INTCON3.INT2IE=1;//HABILITITA LA INT2.  
}  
//////////////////////////////////  
//subrutina para la dirección registro  /////  
void prog(void) {  
ale = 1 ;  
cs = 0 ;  
delay_us(1);  
ale = 0 ;  
cs = 1 ;  
}  
//////////////////////////////////  
//subrutina para escribir en el registro  /////  
void write(void) {  
delay_us(1);  
wri = 0 ;
```



```
delay_us(1);
wri = 1 ;
}
////////////////////SUBROUTINA DE ATENCION DEL MODEM////////////////////
void moden(void){
for(i=0 ;i<=4 ; i++){
led = 1;
delay_ms(100);
led = 0;
delay_ms(100);
}
led=0;
UART1_Init(1200);// modulo rs-232
portd = 0x00;    //CR0 00 configura el tipo de comunicación DPSK, QAM
prog();        //habilita y deshabilita los pines diferenciales TX1 y TX2
portd = 0b01010000; //asincrónico 8bits / char
write();      //modo respuesta , TXA deshabilitado
portd = 0x01;    //CR1 01 control entre el micro y el modem
prog();        //transmite datos presentes en el pin TXD
portd = 0b00001000; //salida de reloj 11.052 MHZ, pasa a través del mezclador
write();
portd = 0x03;    //TR control de tonos, deshabilita RXD
prog();        //deshabilita RXD, deshabilita generador de tono y banda de guarda,
portd = 0b01000000; //transmite dato, deshabilita respuesta de tono
write();
portd = 0b00000100; //CR2 100 habilitar acceso a registro especial
                // (configuración de la señal de entrada)
prog();        //
portd = 0b01100100; //activo registro especial
write();
portd = 0b00000101; //CR3 configura registro especial
```



```
prog();          //calidad de la señal recibida 00 por defecto
portd = 0x00;    //D3 fuente de datos TX pin
write();
portd = 0b00000100; //desactivo acceso a registro especial
prog();
portd = 0b00001100;
write();
portd = 0b00000101; //CR3 registro OH
prog();          // modulación DPSK a 2400 bit/s
portd = 0b00100000; // modo asincrónico 10 bits, 1 bit de parada,
write();
led = 1 ;
portd = 0b00000000;
prog();
portd = 0b01010010;
write();
led = 0x00;
////////////////////lectura de datos de la SD y comunicación rs-232
//hasta que todos los eventos hayan sido enviados////////////////////
sismo[6]=0x30;
sismo[5]=0x30;
sismo[4]=0x30;
exist=0;
contar_evento(void);
exist=Mmc_Fat_Assign(sismo,0x01);
while(exist==1){//permanece mientras no se hayan acabado de leer
//      todos los eventos.
buf3[0]=0x31;// ha existido eventos.
buf3[1]=sismo[6];
buf3[2]=sismo[5];
buf3[3]=sismo[4];
```




```
size=Mmc_Fat_Get_File_Size();
LongWordToStr(size,text);
tamano=(size-42)/690;
Mmc_Fat_Reset(size);
for(i=0;i<=41;i++){
Mmc_Fat_Read(&buf1[i]); // cabecera de datos.
}
////////////////////// escribimos la cabecera de datos vía rs-232.
for(i=0;i<=3;i++){
UART1_Write(buf3[i]);//hay evento,#evento
}
for(i=0;i<=10;i++){
UART1_Write(text[i]);// tamaño del evento.
}
for(i=0;i<=41;i++){
UART1_Write(buf1[i]); // cabecera de evento
}
for(j=1;j<=tamano;j++){
for(i=0;i<=689;i++){
Mmc_Fat_Read(&buf2[i]); // bloque de 15 datos.
}
// escribimos via rs-232 los datos del primer bloque.
for(i=0;i<=689;i++){
UART1_Write(buf2[i]);
}
}
contar_evento(void);
exist=Mmc_Fat_Assign(sismo,0x01);
}
buf3[0]=0x30; // no hay más eventos y se termina la comunicación.
buf3[1]=0x30;
```



```
buf3[2]=0x30;
buf3[3]=0x30;
////////////////////////////////////
modem = 0;
portd = 0b00000000;
prog();
portd = 0b01010000;
write();
led = 0x00;
portd = 0b00000101; //CR3 registro OH
prog(); // modulacion DPSK a 2400 bit/s
portd = 0b00000000; // modo asincrónico 10 bits, 1 bit de parada,
write();
delay_ms(500);
delay_ms(500);
modem = 0;
}
```

////////////////////////////////PROGRAMA PRINCIPAL////////////////////////////////

```
void main(void){
portb = 0X00;
trisb = 0x07;
porta = 0X00; //puerto de control
trisa = 0x00;
portc = 0X00; // envío de datos seriales al modem
trisc = 0x00;
portd = 0X00; //puerto para la configuración del modem
trisd = 0x00;
// configura estado inicial de las líneas de control del modem
ale = 0x00; //ale
cs = 0x01; //chip select
```



```
wri = 0x01; //write
read = 0x01; //read
rele = 0x00; //relé
modem = 0; // bandera de interrupción
////////////////////////////////////
////////////////////////////////////
INT0_Direction=1;
INT1_Direction=1;
INT2_Direction=1;
TRISC.F6=0;/////////UART/////////GPS/////////
TRISC.F7=1;/////////UART/////////GPS/
CS1_Direction=0;
CS2_Direction=0;
SCK_Direction=0;
SDI_Direction=1;
SDO_Direction=0;
Mmc_Chip_Select_Direction=0;
Mmc_Chip_Select=1;
CS1=1;
CS2=1;
led=1;

/////////////////////////////////CONFIGURACION DE LOS BITS DE LAS INTERRUPCIONES/////////////////////////////////
RCON.IPEN=1;//NIVELES DE PRIORIDAD DE LAS INTERRUPCIONES.
INTCON3.INT1IP=0;
INTCON3.INT2IP=0;
INTCON2.TMR0IP=1;
TOCON=0x08;// APAGADO
INTCON2.INTEDG0=0; // SELECCION DEL FLANCO DE RESPUESTA.
INTCON2.INTEDG1=0;
INTCON2.INTEDG2=0;
```



INTCON.GIEH=1;//HABILITA TODAS LAS INTERRUPCIONES DE PRIORIDAD ALTA.

INTCON.GIEL=1;//HABILITA TODAS LAS INTERRUPCIONES PERIFERICAS DE PRIORIDAD BAJA.

INTCON.INT0IE=0;// HABLITITA LA INT0.

INTCON3.INT1IE=0;//HABLITITA LA INT1.

INTCON3.INT2IE=1;//HABLITITA LA INT2.

////////////////////////////////////

tiempo=0;

INTCON.TMR0IE=1;

TOCON.TMR0ON=0;

////////////////////////////////////

SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64,_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_LOW,_SPI_LOW_2_HIGH);

while(Mmc_Fat_Init()!=0)LED=0;

SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV16,_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_LOW,_SPI_LOW_2_HIGH);

////////////////////////////////////

for(i=0;i<=689;i++){

buf2[i]=0x30;

}

////////////////////////////////////

TMR0L=0xF6;

TMR0H=0xFF;

portd = 0b00000101; //CR3 registro OH

prog(); // modulacion DPSK a 2400 bit/s

portd = 0b00000000; // modo asincrónico 10 bits, 1 bit de parada,

write();

continuar:

CS1=1;

CS2=1;

if (modem == 1){

```
moden();  
}  
led =~led;  
delay_ms(500);  
goto continuar;  
}
```


4.2.2 Algoritmo del sistema de visualización.

4.2.2.1 Configuración de frecuencia de trabajo del microprocesador

El PIC18F452 se utiliza para el control del TOUCH SCREEN, y se encarga de visualizar la señal que viene del componente vertical del geófono.

Para el funcionamiento correcto de la pantalla táctil y considerando la frecuencia de trabajo del PIC master. Se utiliza el multiplicador de frecuencia PLL con la configuración siguiente:

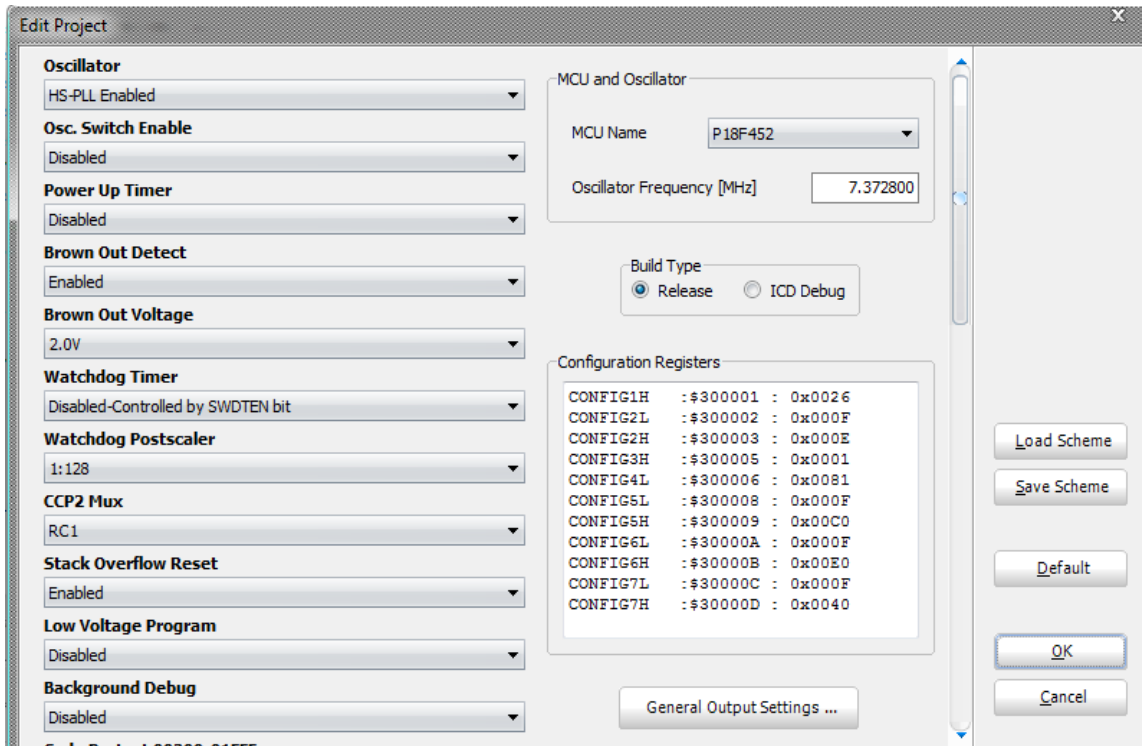


Figura 4.2.2.1 Configuración de la frecuencia de trabajo para el PIC18F452

4.2.2.2 Configuración de la velocidad del bus SPI

En el momento de la inicialización del programa se establecen todos los parámetros adecuados para esta comunicación, que solo permanece activa durante una solicitud de graficar por parte del usuario.

Siguiendo las especificaciones del bus SPI:

- PIC 18F452 como Esclavo.
- SCK proviene del PIC master

El diagrama de tiempos en la comunicación SPI es:

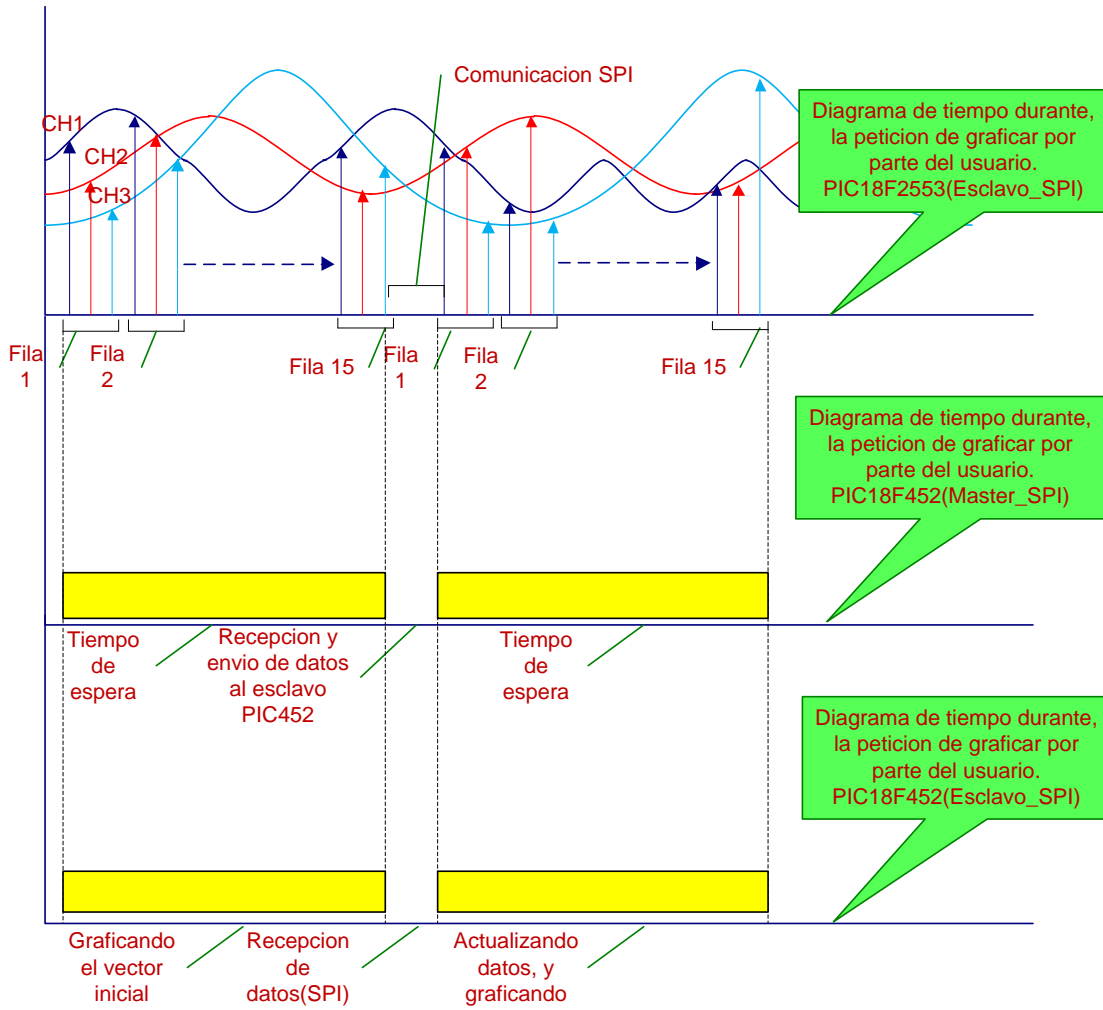


Figura 4.2.2.2 Diagrama de tiempo durante una petición de graficar.

4.2.2.3 Algoritmo

```

*****
*****
*****

```



```
sbit CS2 at RA5_bit;// CHIP SELEC PARA LA COMUNICACION SPI
sbit CS2_Direction at TRISA5_bit;
sbit interrupcion at RC2_bit;// LINEA DE INTERRUPCION EN CASO DE PETICION
DE TEST.
sbit interrupcion_Direction at TRISC2_bit;
/////////////////////////////////ASIGNACION DE PINES PARA EL PUERTO SPI/////////////////////////////////
sbit SCK at RC3_bit;
sbit SCK_Direction at TRISC3_bit;
sbit SDI at RC4_bit;
sbit SDI_Direction at TRISC4_bit;
sbit SDO at RC5_bit;
sbit SDO_Direction at TRISC5_bit;
sbit LED at RB7_bit;
sbit LED_Direction at TRISB7_bit;
/////////////////////////////////PINES PARA EL MANEJO DEL TOUCH_SCREEN/////////////////////////////////
char GLCD_DataPort at PORTD;
sbit GLCD_CS1 at RB0_bit;
sbit GLCD_CS2 at RB1_bit;
sbit GLCD_RS at RB2_bit;
sbit GLCD_RW at RB3_bit;
sbit GLCD_EN at RB4_bit;
sbit GLCD_RST at RB5_bit;
sbit GLCD_CS1_Direction at TRISB0_bit;
sbit GLCD_CS2_Direction at TRISB1_bit;
sbit GLCD_RS_Direction at TRISB2_bit;
sbit GLCD_RW_Direction at TRISB3_bit;
sbit GLCD_EN_Direction at TRISB4_bit;
sbit GLCD_RST_Direction at TRISB5_bit;
/////////////////////////////////DEFINICION_VARIABLES DE USO GENERAL/////////////////////////////////
```




```
float x_coord,y_coord,x_coord128,y_coord64;
static char buf[690]; // buffer para almacenamiento temporal de los datos en código ASCII.
float result[15];
int punt[128],exit;
long i,k,datx;
////////////////////////////////////
void getx(){// Obtención de la coordenada X del touch screen.
portc.RC0 =1;
portc.RC1 =0;
delay_ms(5);
x_coord = ADC_Read(0);
}
void gety(){// // Obtención de la coordenada Y del touch screen.
portc.RC0 = 0;
portc.RC1 = 1;
delay_ms(5);
y_coord = ADC_Read(1);
}
//_____PROGRAMA
PRINCIPAL_____
void main(void){
// configuración de puertos y registros de uso general.
ADCON1=0x0D;// I/O DIGITALES+RA0+RA1 COMO ANALOGICAS
AN3=VREF+,AN2=VREF-;
TRISA.F0=1;
TRISA.F1=1;
////////////////////////////////////
TRISC.F0=0;
TRISC.F1=0;
```



```
CS2_Direction=1;
SDI_Direction=1;
SCK_Direction=1;
SDO_Direction=0;
LED_Direction=0;
interrupcion_Direction=0;
interrupcion=1;
LED=1;
// inicialización del módulo SPI.
SPI1_Init_Advanced(_SPI_SLAVE_SS_ENABLE,
_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);
////////////////////////////////////
LED=0;
// inicializacion del GLCD(Touch Screen).
Glcd_Init();
Glcd_Fill(0X00);
Glcd_Write_Text("Ing:Cobos_Chin",30 ,0,1);
Glcd_Write_Text("RSA",55,7,1);
Glcd_Rectangle(8,16,60,48,1);
Glcd_Box(10,18,58,46,1);
Glcd_Write_Text("TEST",14,3,0);
for(i=0;i<=127;i++){// inicialización del vector a graficar.
    punt[i]=0;
    punt[i]=50-((punt[i]/5)*51);// establecimiento de coordenadas para graficar
    //          el vector inicial.
}
while(1){
    getx();
    gety();
```



```
x_coord128=(x_coord*128)/1024;// determinación de la coordenada x, considerando
// que el conversor analógico digital tiene una resolución de 10 bits.
// y una cantidad de pixeles de 128x64.
y_coord64=64-((y_coord*64)/1024);
/////////////////////////////////////////////////////////////////
if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) && (y_coord64
<= 46)){
// siempre y cuando se haya pulsado el cuadro de TEST en el GLCD.
LED=1;
  Glcd_Fill(0x00);
  exit=0;
  Glcd_Write_Text("waiting movement..",14 ,3,0);
while(exit==0){
  interrupcion=0;// activación de la interrupción INT1.
  for(i=0;i<=689;i++){// recepción del bloque de datos para graficar.
    // estos datos los envía primero el amplificador al almacenador,
    // y luego este al visualizador.
  buf[i]=SPI1_Read(0x00);
  }
  for(i=0;i<=112;i++){
    punt[i]=punt[i+15]; // actualización de puntos a graficar.
  }
  k=0;
  for(i=0;i<=14;i++){ // conversión de los caracteres ASCII
    // a valores en punto flotante.
  result[i]=atof(&buf[k]);
    k=k+46;
    punt[i+113]=50-((result[i]/5)*51);
  }
}
```



```
// inicialización de los ejes para graficar.
Glcd_Fill(0x00);
    Glcd_H_Line(0,128,50,1);
    Glcd_V_Line(0,50,0, 1);
    Glcd_H_Line(0,1,40,1);
    Glcd_H_Line(0,1,30,1);
    Glcd_H_Line(0,1,20,1);
    Glcd_H_Line(0,1,10,1);
    Glcd_H_Line(0,1,0,1);
Glcd_Write_Text("exit",104,7,0);
    for(i=0;i<=126;i++){// graficando los 127 puntos actualizados
//          dentro del touch screen.
        Glcd_Line(i,punt[i],i+1,punt[i+1], 1);
    }
    // obtención de las coordenadas X e Y.
getx();
    gety();
    x_coord128=(x_coord*128)/1024;
y_coord64=64-((y_coord*64)/1024);
    // si ha sido pulsado el boton de exit ?.
if ((x_coord128 >= 102) && (x_coord128 <=126) && (y_coord64 >=53) && (y_coord64
<=62)){

exit=0x01; // cambio en el valor de estado exit.
Glcd_Fill(0x00);
    Glcd_Write_Text("Ing:Cobos_Chin",30 ,0,1);
Glcd_Write_Text("RSA",55,7,1);
    Glcd_Rectangle(8,16,60,48,1);
    Glcd_Box(10,18,58,46,1);
```



```

    Glcd_Write_Text("TEST",14,3,0);
}

    datx=SPI1_Read(exit); // envío del valor del estado exit al PIC 18f452_master.
}

    interrupcion=1; // desactivación de la INT1.
}

////////////////////////////////////

}

*****
*****
*****

```

4.2.3 Algoritmo del sistema de muestreo y detección de eventos.

4.2.3.1 Configuración de frecuencia de trabajo del microprocesador

El PIC18F2553 se encarga de verificar la existencia o no de un evento sísmico, en base a la ejecución del software diseñado para tal cometido.

Cuando se ha detectado la presencia de un sismo el PIC debe muestrear las señales de los geófonos a la mayor velocidad posible, para luego enviarlas al PIC almacenador. Es por eso que también se utiliza el PLL para la configuración de la frecuencia de trabajo del PIC (ver Figura 4.2.3.1).

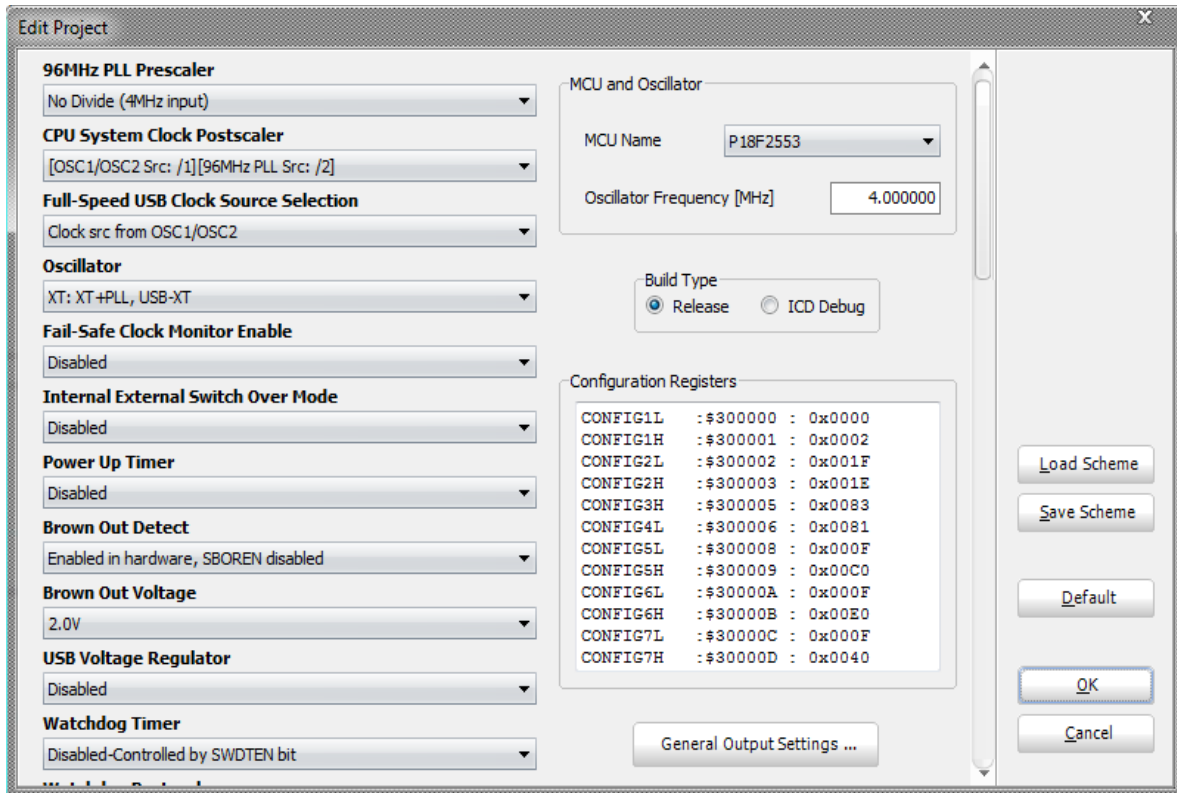


Figura 4.2.3.1 Configuración de la frecuencia de trabajo para el PIC18F2553

4.2.3.2 Configuración de la velocidad del bus SPI

Los parametros para la inicializacion del bus es:

- PIC18F2553 Configurado como esclavo.
- SCK proveniente del Master(Almacenador, PIC18F452).

El diagrama de tiempos durante la detección y almacenamiento de un evento, es el de la Figura 4.2.1.2.

4.2.3.3 Algoritmo

```

*****
*****
*****

```

```

////////////////////////////////////pines del puerto SPI////////////////////////////////////
sbit CS1 at RA5_bit;
sbit CS1_Direction at TRISA5_bit;

```



```
sbit SDI at RB0_bit;
sbit SDI_Direction at TRISB0_bit;
sbit SCK at RB1_bit;
sbit SCK_Direction at TRISB1_bit;
sbit SDO at RC7_bit;
sbit SDO_Direction at TRISC7_bit;
//////////pines para el manejo de INTERRUPCION INT0//////////
sbit LED at RC2_bit;
sbit LED_Direction at TRISC2_bit;
sbit INTERRUPCION at RB2_bit;
sbit INTERRUPCION_Direction at TRISB2_bit;
//////////variables globales//////////
short datx;
static char buf[690]; // buffer para el almacenamiento temporal de los datos en ASCII.
long i,k,j,u,tiempo1,estado,tiempo2;
float val[3];

//////////INTERRUPCION_manejo del TIMER0//////////
void interrupt(){
  if(INTCON.TMR0IF==1){
    INTCON.TMR0IF=0;

    if(tiempo1>=5){
      tiempo2++;
    }

    tiempo1++;
  }
  TMR0L=0xE5;
  TMR0H=0x48;
  INTCON.TMR0IE=1;
}

//////////lectura del ADC y conversión ASCII//////////
void bufferadc(void){

  for(i=0;i<=689;i++){
    buf[i]=0x30;
  }
  k=0;

  for(u=0;u<=14;u++){
    for(i=0;i<=2;i++){
```



```
    val[i] =Adc_Read(i);
    val[i] = (double)val[i]*0.001220703;
}
for(j=0;j<=1;j++){
    FloatToStr(val[j],&buf[k]);
    buf[k+14]=0x09;
    k=k+15;
}
FloatToStr(val[2],&buf[k]);
k=k+15;
buf[k-1]=0x0d;
buf[k]=0x0a;
k=k+1;
}

}

//_____PROGRAMA
PRINCIPAL_____

void main(){
// configuracion de puertos, direcciones de pines, y registros de uso general.
ADCON1=0x0C;// I/O DIGITALES y AN0,AN1,AN2 como analógicas.
TRISA.F0=1;
TRISA.F1=1;
TRISA.F2=1;
INTERRUPCION_Direction=0;
INTERRUPCION=1;
////////////////////inicialización de los pines que controlan el multiplexor////////////////
TRISA.F3=0; //A
TRISA.F4=0; //B
TRISC.F0=0; //C
PORTA.F3=1;
PORTA.F4=0;
PORTC.F0=0;

TRISC.F4=0; //A
TRISC.F5=0; //B
TRISC.F6=0; //C
PORTC.F4=1;
PORTC.F5=0;
PORTC.F6=0;

TRISB.F3=0; //A
```




```
TRISB.F4=0; //B
TRISB.F5=0; //C
PORTB.F3=1;
PORTB.F4=0;
PORTB.F5=0;

////////////////////////////////////
CS1_Direction=1;
SDI_Direction=1;
SCK_Direction=1;
SDO_Direction=0;
LED_Direction=0;
////////////////////////////////////
RCON.IPEN=1; // asignación de prioridad de las interrupciones.
INTCON2.TMR0IP=1;
T0CON=0x07;// Timer0 APAGADO
TMR0L=0xE5;// valores de los registros de timer para realizar el conteo cada segundo.
TMR0H=0x48;
tiempo1=0;
tiempo2=0;
// activación de las interrupciones.
INTCON.GIEH=1;
INTCON.GIEL=1;
INTCON.TMR0IE=1;
T0CON.TMR0ON=0; //APAGADO
////////////////////////////////////
LED=1;
// inicialización del módulo SPI.
SPI1_Init_Advanced(_SPI_SLAVE_SS_ENABLE,
_SPI_DATA_SAMPLE_MIDDLE,_SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);

now:
tiempo1=0;
tiempo2=0;
////////////////////////////////////establecimiento de un valor da ganancia
//para cada canal del geófono.
PORTA.F3=1;
PORTA.F4=0;
PORTC.F0=0;

PORTC.F4=1;
PORTC.F5=0;
PORTC.F6=0;
```



```
PORTB.F3=1;
PORTB.F4=0;
PORTB.F5=0;
////////////////////////////////////

INTERRUPCION=1;
for(i=0;i<=2;i++){// conversion analogica digital.
  val[i] = Adc_Read(i);
  val[i] = (double)val[i]*0.001220703;
}

estado=0;
if(val[0]>=4){// si el valor del canal horizontal del geófono es mayor a 4V ?.
  T0CON.TMR0ON=1; //ENCENDIDO
  while(tiempo1<=5);// tiempo de espera.

  while(tiempo2<=2){// realiza la lectura del ADC y verifica la persistencia del sismo.
    // de ser asi activa la INTERRUPCION INT0 y manda a almacenar los datos
    // en la SD por 60 seg. caso contrario sale del if sin ejecutar
    // la funcion de almacenamiento.
    for(i=0;i<=2;i++){
      val[i] = Adc_Read(i);
      val[i] = (double)val[i]*0.001220703;
    }

    if(val[0]>=4){// persiste el movimiento ?.
      estado=0x01;
      while(estado==0x01){// mientras no se sobrepasen los 60 seg controlados
        // por al Almacenador.
        bufferadc(void);

        LED=0;
        INTERRUPCION=0;
        for(i=0;i<=689;i++){ // envío de los datos vía SPI al almacenador.
          datx=SPI1_Read(buf[i]);
        }

        estado=SPI1_Read(0x00);
      }
    }
  }
  T0CON.TMR0ON=0; //ENCENDIDO
}
```



```
goto now;  
}
```

```
*****  
*****  
*****
```

4.2.4 Descripción del algoritmo del controlador de carga

A continuación se indica el diagrama de flujo a seguir:

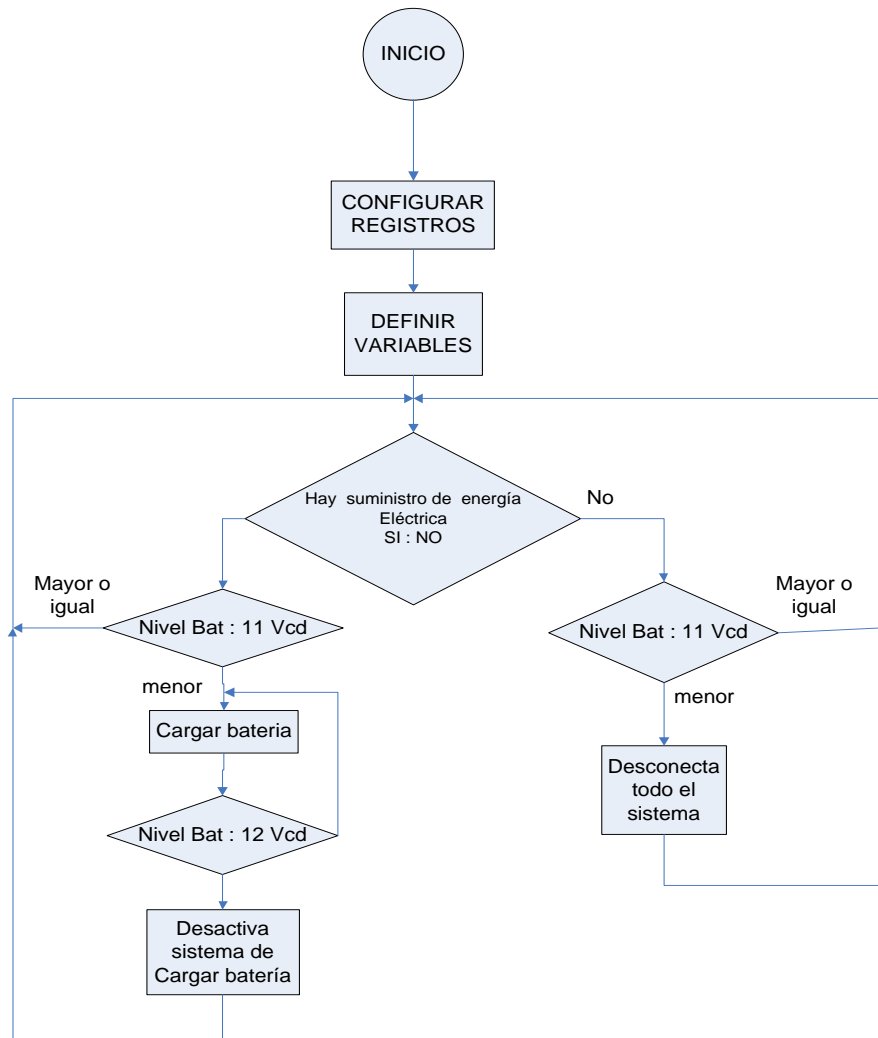


Figura 4.2.4.1 Diagrama de flujo del controlador de carga.

En la inicialización del programa se definen las variables, se configura el PIC16f628A para que trabaje con el oscilador interno a una frecuencia de 4 MHz.

El pic16f628 tiene dos comparadores de señales analógicas. Las mismas que se comparan de forma alternada. La tensión del cargador y de la batería, están conectadas a los canales CH0 y CH1 respectivamente.

La configuración de los comparadores se realiza mediante el registro CMCON, el voltaje de referencia para la comparación con el registro VRCON. En el registro CMCON se configura en encendido o apagado de los comparadores, se selecciona que canales que se van a comparar con el voltaje de referencia, se lee el resultado de la comparación en los bits C1OUT, C2OUT. En el registro VRCON se establece el voltaje de referencia para realizar la comparación, además se habilita o deshabilita en modulo interno del Vref.

Para configurar los puertos se utiliza los registros TRISA, TRISB donde el puerto A se configura como entrada, el puerto B como salida. Los pines RB0, RB1 se utilizan para activar o desactivar el relé de enclavamiento que suministra las tensiones a las distintas placas, con el pin RB6 se activa el relé para cargar la batería.

Para configurar el Voltaje de referencia en el registro VRCON se procede a calcular los valores reales de tensión de la siguiente manera:

$$V_{REF} = \frac{Vr < 3:0 > \times VDD}{32} + \left(VDD * \frac{1}{4} \right)$$

Donde VDD = tensión de alimentación del dispositivo (fondo escala)

Vr<3:0> = puede tomar valores entre 0 y 15

El valor de tensión para cargar la batería es de 11 VDC cuando este valor pasa por el divisor de tensión es de 2.45 VDC conectado al pin RA1 entonces:

$$V_{REF} = \frac{8 \times 5}{32} + \left(5 * \frac{1}{4} \right) = 2.45 \text{ Vcd}$$

La tensión que indica que la batería ya se encuentra cargada es de 12 VDC, cuando pasa por el divisor de tensión es de 2.68

$$V_{REF} = \frac{10 \times 5}{32} + \left(5 * \frac{1}{4} \right) = 2.79 \text{ VDC}$$

Entonces el voltaje de referencia Vref para proceder a cargar la batería cuando hay suministro de energía eléctrica es 2.45 VDC, para desconectar la carga de la batería es de 2.79.



Por último, se tiene un relé principal (de enclavamiento) que activa o desactiva el suministro de energía a las distintas placas, un relé de carga de batería y cuya activación o desactivación es como sigue: si la tensión del cargador es 12V , y la tensión de batería es menor a 11V (nivel inferior), entonces se activa el relé de carga; caso contrario, si la tensión de batería es mayor a 11 V (nivel superior), entonces el relé de carga seguirá desactivado. Ahora, si la tensión del cargador, la batería es menor a 11V y, el relé de enclavamiento desconecta todo el sistema hasta cuando se superen estos umbrales.

CAPITULO 5: IMPLEMENTACIÓN.

5.1 PRUEBAS FINALES DEL EQUIPO.

Las pruebas realizadas se resumen en los siguientes pasos:

5.1.1 Amplificación de la señal proveniente del geófono.

5.1.1.1 Señal del Geófono

La señal proveniente del Geófono tiene valores de milivoltios como se puede observar en la parte superior de la figura 5.1, pero una señal de este tipo no es útil para los objetivos planteados en este trabajo. Entonces es necesario amplificarla como se muestra en la parte inferior de la figura 5.1. El canal 1 del osciloscopio se encuentra conectado directamente a la salida del geófono y se puede observar que está seteado en 200 milivoltios, mientras que el canal 2 se encuentra ubicado a la salida del amplificador y seteado a 5 voltios.

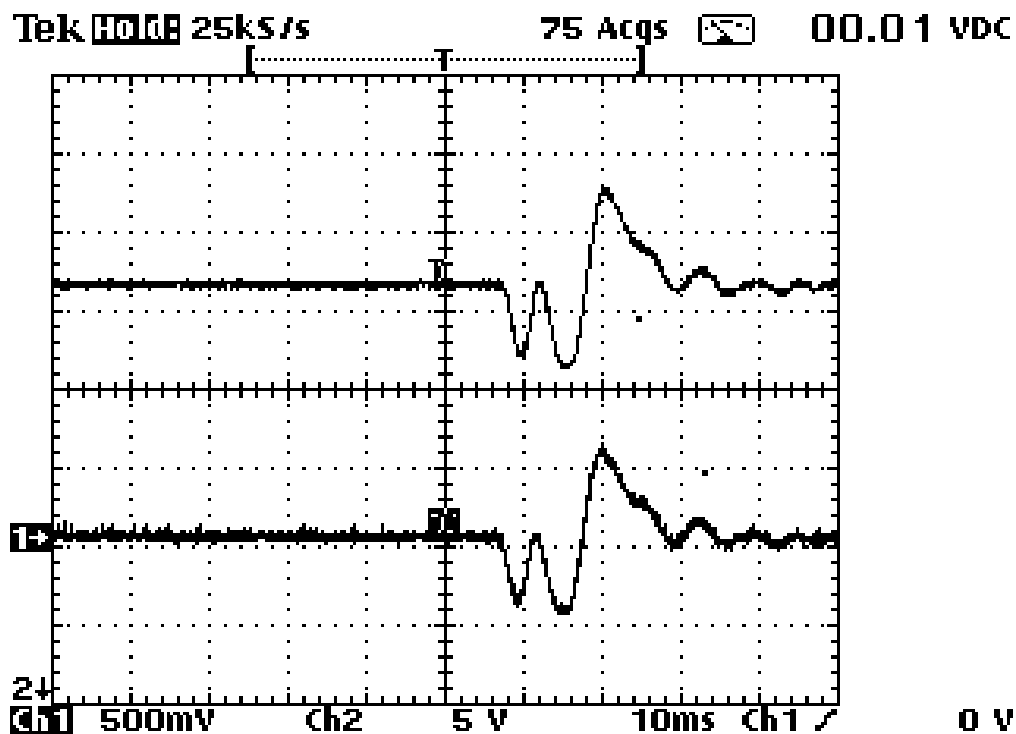


Figura 5.1 Amplificación de la señal proveniente del geófono.

5.1.1.2 Señal de entrada al PIC2553.

El microcontrolador utilizado PIC2553 recepta únicamente señales positivas, es por esta razón que la señal que proviene de la etapa de amplificación debe atravesar por un circuito sumador para ingresar a la captura de datos.

5.1.2 Lectura del GPS y funcionamiento RTC

El GPS envía datos a 4800 baudios en forma serial (figura 5.2), la cual es leída y procesada por el microcontrolador del almacenador para igualar el RTC, la información tomada del GPS está en hora GMT (hora, minuto, segundos, día, mes, año).

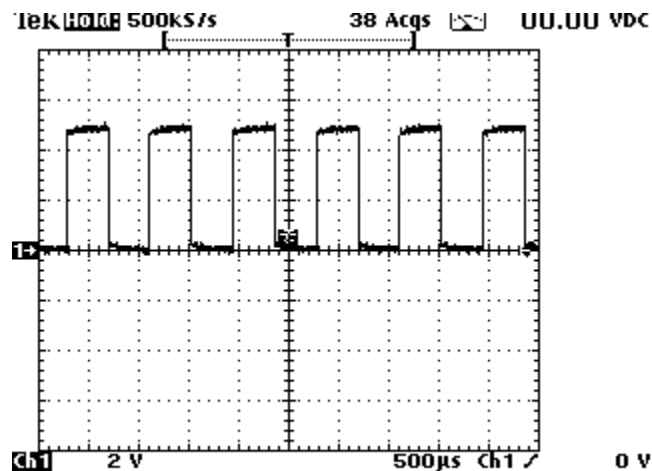


Figura 5.2. Trama enviada por el GPS hacia el almacenador

Una vez que se ha igualado el RTC con la hora GMT proveniente del GPS, los datos que son capturados por el amplificador se guardan en la memoria SD con el nombre EVEN001, EVEN002, EVEN003,..., así sucesivamente cada vez que se produzca un evento sísmico, la información de fecha y hora que se leen del RTC, se coloca en cada archivo como fecha y hora de creación del mismo.

En la figura 5.3 se puede observar varios archivos con el nombre de EVEN00X, donde cada uno tiene su fecha y hora de creación.

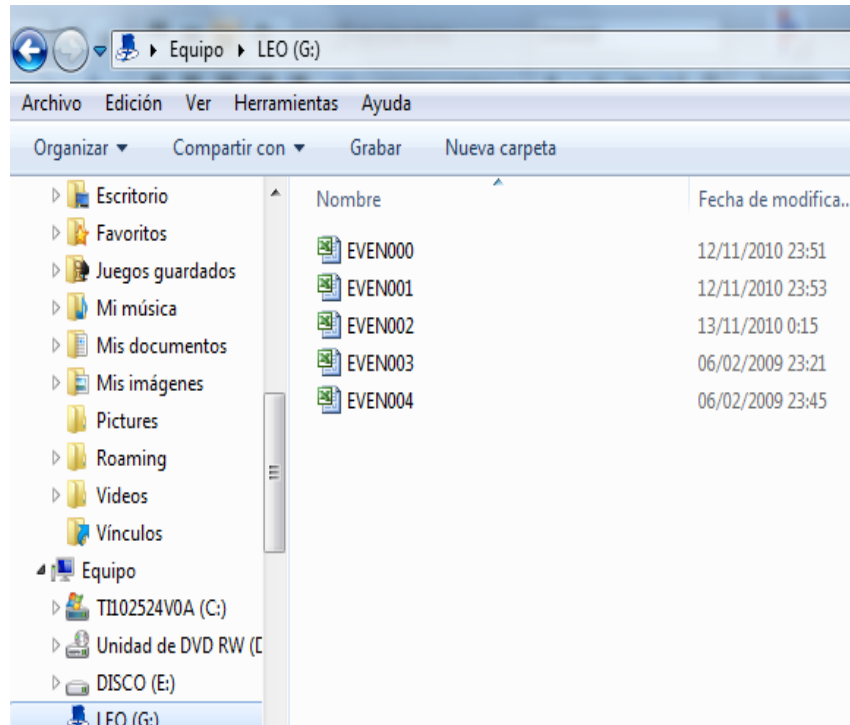


Figura 5.3 Archivos guardados en la memoria SD, con hora GMT.

5.1.3 Detección de un evento, transmisión de datos hacia el almacenador

Cuando se detecta un evento sísmico el amplificador, captura datos de los tres canales del geófono, acondiciona y los envía a guardar en la tarjeta SD.

En la tabla 5.1 se puede apreciar las *11611 muestras* que se han capturado durante el minuto de muestreo, cuando se ha simulado un evento sísmico.

Los datos de la tabla 5.1 se grafican en la figura 5.4

| # muestra | CHANEL1 | CHANEL2 | CHANEL3 |
|-----------|----------|----------|----------|
| 1 | 2.559813 | 2.894286 | 3.284911 |
| 2 | 1.978759 | 2.893065 | 2.860106 |
| 3 | 1.557616 | 2.890624 | 2.385253 |
| 4 | 1.362304 | 2.896728 | 1.977538 |
| 5 | 1.414794 | 2.896728 | 1.726073 |
| 6 | 1.678466 | 2.90161 | 1.696777 |
| 7 | 2.094726 | 2.907714 | 1.879882 |
| 8 | 2.572021 | 2.916259 | 2.213134 |
| 9 | 3.031005 | 2.923583 | 2.602538 |
| 10 | 3.386229 | 2.912597 | 2.965087 |
| 11 | 3.582763 | 2.908935 | 3.232421 |
| 12 | 3.607177 | 2.904052 | 3.359374 |
| 13 | 3.468017 | 2.905272 | 3.323974 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 11599 | 2.84912 | 2.905272 | 2.728271 |
| 11600 | 2.858886 | 2.904052 | 2.74414 |
| 11601 | 2.86621 | 2.906493 | 2.760009 |
| 11602 | 2.875976 | 2.908935 | 2.76245 |
| 11603 | 2.880858 | 2.905272 | 2.767333 |
| 11604 | 2.879638 | 2.904052 | 2.768554 |
| 11605 | 2.877196 | 2.902831 | 2.769774 |
| 11606 | 2.871093 | 2.908935 | 2.768554 |
| 11607 | 2.868651 | 2.911376 | 2.752685 |
| 11608 | 2.861327 | 2.902831 | 2.740478 |
| 11609 | 2.860106 | 2.904052 | 2.731933 |
| 11610 | 2.858886 | 2.904052 | 2.72705 |
| 11611 | 2.862548 | 2.910155 | 2.716063 |

Tabla 5.1 Datos guardados en la memoria SD

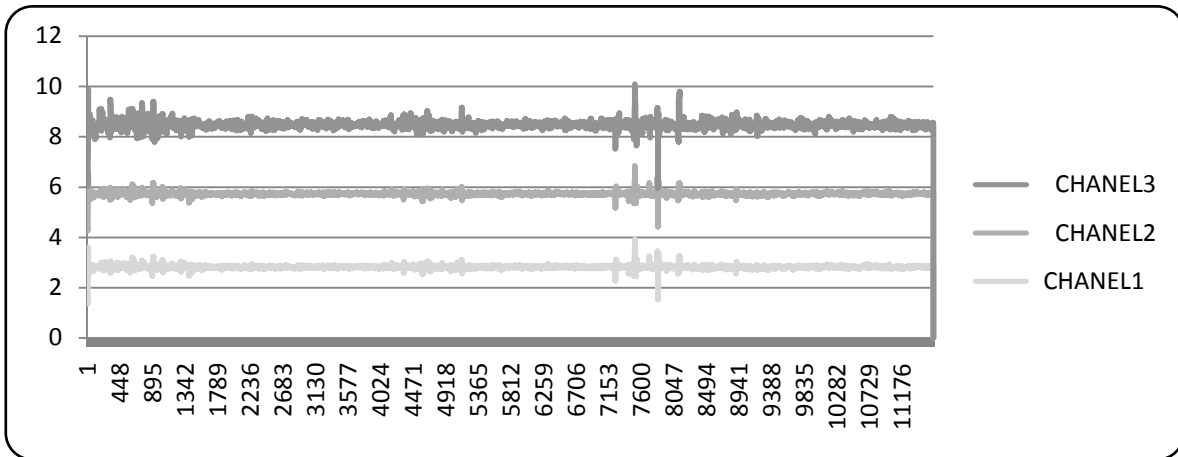


Figura 5.4 Gráfica de las señales guardadas en la tarjeta SD, de la tabla 5.1

5.1.4 Visualización en la pantalla GLCD

Para desplegar los valores provenientes del amplificador (cuando se ha producido un evento sísmico) se ha desarrollado un módulo de visualización basado en el PIC452, el cual se comunica con la tarjeta del almacenador mediante SPI. En la figura 5.5 se puede observar el gráfico de un evento simulado.

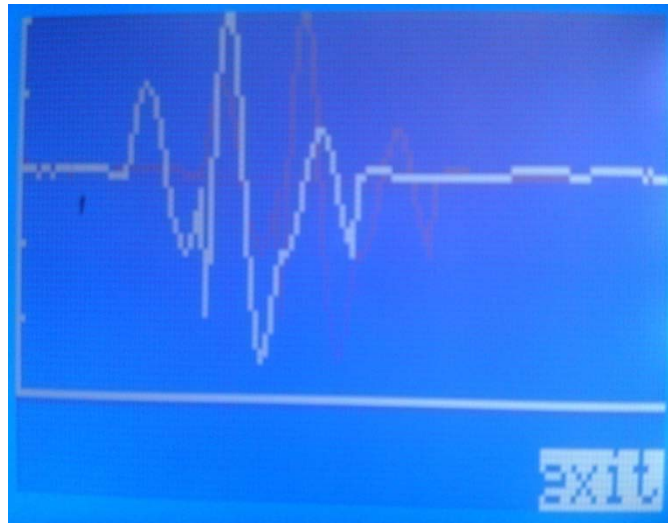


Figura 5.5. Presentación gráfica de los valores provenientes de la tarjeta del amplificador

5.1.5 Transmisión por el modem telefónico.

El proceso de transmisión de datos se realiza mediante el PIC452 conectado al modem 73K224BL, el mismo que consiste en:

- El PIC Lee los valores almacenados en la tarjeta SD, y los envía (protocolo RS-232) hacia el modem telefónico, para que este transfiera la información hacia la línea telefónica.
- Una vez que se ha enviado todos los datos de la tarjeta SD, el microcontrolador mediante el modem cierra la comunicación.

En la siguiente gráfica se muestran los resultados del proceso de enviar los datos por la línea telefónica.

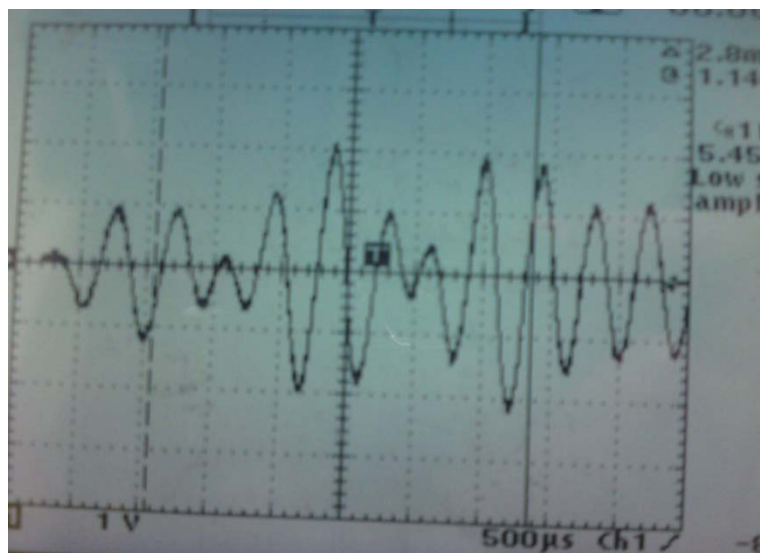


Figura 5.6 Envío de información por la línea telefónica

5.2 MANUAL DE FUNCIONAMIENTO DEL EQUIPO CONSTRUIDO.

Implementación del equipo construido

5.2.1 Controlador de Carga

5.2.1.1 Descripción general

La tarjeta del controlador de carga está basada en el PIC16f628A. Las señales que provienen del cargador y de la batería ingresan a los canales analógicos del PIC por medio de unos divisores de tensión, la información que ingresa en el canal A0 proviene del cargador y la del canal A1 de la batería. Además posee dos relés uno para cargarla, y otro para la conexión o desconexión de la alimentación a las distintas tarjetas. La tarjeta tiene una fuente de alimentación de +5, +3.3 VDC, y +/-12.

5.2.1.2 Funcionamiento

Las señales analógicas de tensión de la batería y el cargador ingresan a los canales analógicos AN0, AN1 del PIC16F628A a través de los divisores de tensión R1,R2 y R3,R4; el PIC una vez que obtiene las señales se comparan los valores con un voltaje de referencia seteado en el registro VRCON. Para activar o desactivar los relés, K1 (relé del cargador de la batería) y K2 (relé para suministrar energía a las demás placas).

Si la tensión del cargador es menor a 12 voltios y el nivel de voltaje de la batería está por debajo a los 11 voltios, el PIC a través del relé K2 desconecta el suministro de energía a todas las demás tarjetas. Si el nivel de tensión de cargador es mayor a los 12 VDC y el de la batería está por los 11 voltios entonces se activa el relé K1 para cargar la batería, el relé K1 estará activado hasta que el voltaje de la batería este en los 12 voltios.

A partir de una tensión de +12 de la batería y con el regulador de tensión LM7805 se obtiene una tensión de 5V para la alimentación de las tarjetas del Almacenador, Amplificador; con el regulador LM3.3AZIIL 3.3 VDC para la memoria SD y el GPS, con el circuito integrado 7662 se obtiene una tensión de -12 VDC para los amplificadores operacionales de la tarjeta del Amplificador.

5.2.1.3 Especificaciones del controlador de carga

- Posee entradas para la tensión del cargador de la batería y la batería.
- Tensión de entrada +12
- Tensiones de salida 5, 3.3, +12, -12 VDC
- Resistencias de precisión para los divisores de tensión.
- Tensión de 5 VDC para alimentar de pic.

5.2.1.4 Descripción de los pines de la tarjeta del controlador de carga

1. CON1, CON2, CONA, CONB entrada de tensión proveniente del cargador para la batería.
2. CON7, CON8, CONH, CONG entrada de tierra del cargador de la batería.
3. CON4, CON5, COND, CONBE, entrada de tensión para la batería.
4. CON10, CON11, CONM, CONL, entrada de tierra de la batería.
5. CONC salida de -12 VDC.
6. CON3 Salida de 12 VDC.
7. CON6 salida de 3.3 VDC.
8. CON12 salida de 5 VDC.
9. PIN4, PIN3 pines para la conexión de una inductancia entra la tierra del cargador y la batería.
10. PIN1, PIN2 pines para alimentar directamente a la tarjeta sin la necesidad de la batería

5.2.1.5 Descripción de los elementos del controlador de carga

11. K1 relé de carga de la batería.
12. K1 relé enclavamiento (principal).
13. ICL7662: Circuito integrado convertidor de voltaje.
14. D1, D2, D3 Diodos para conectar el terminal positivo del cargador con el contacto central del relé K1.
15. PIC16f628A
16. 78LZ05 regulador de voltaje de 5 VDC.
17. LM7805, Lm7803 reguladores de voltaje.

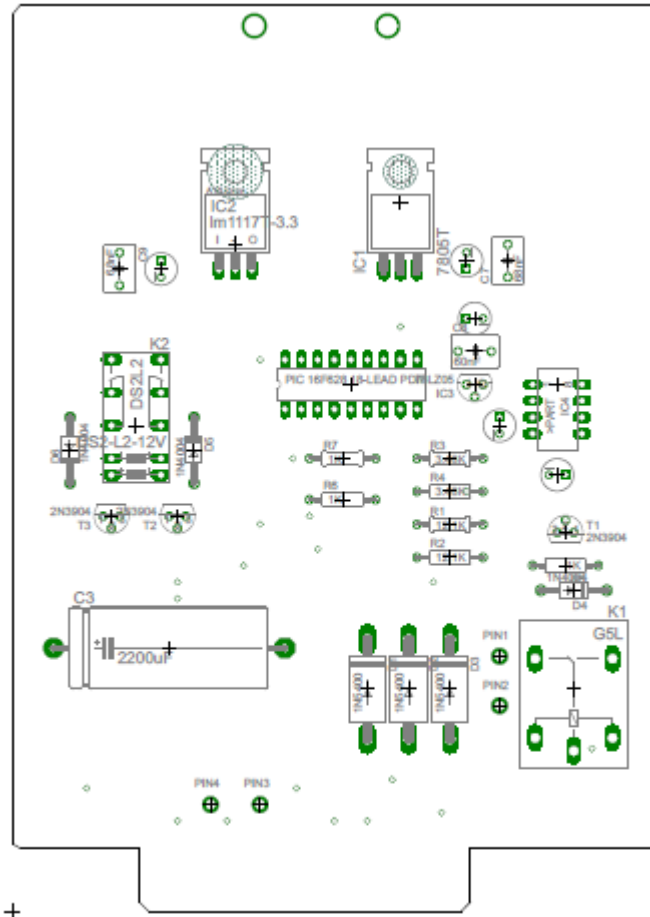


Figura.5.7. Tarjeta del controlador de carga

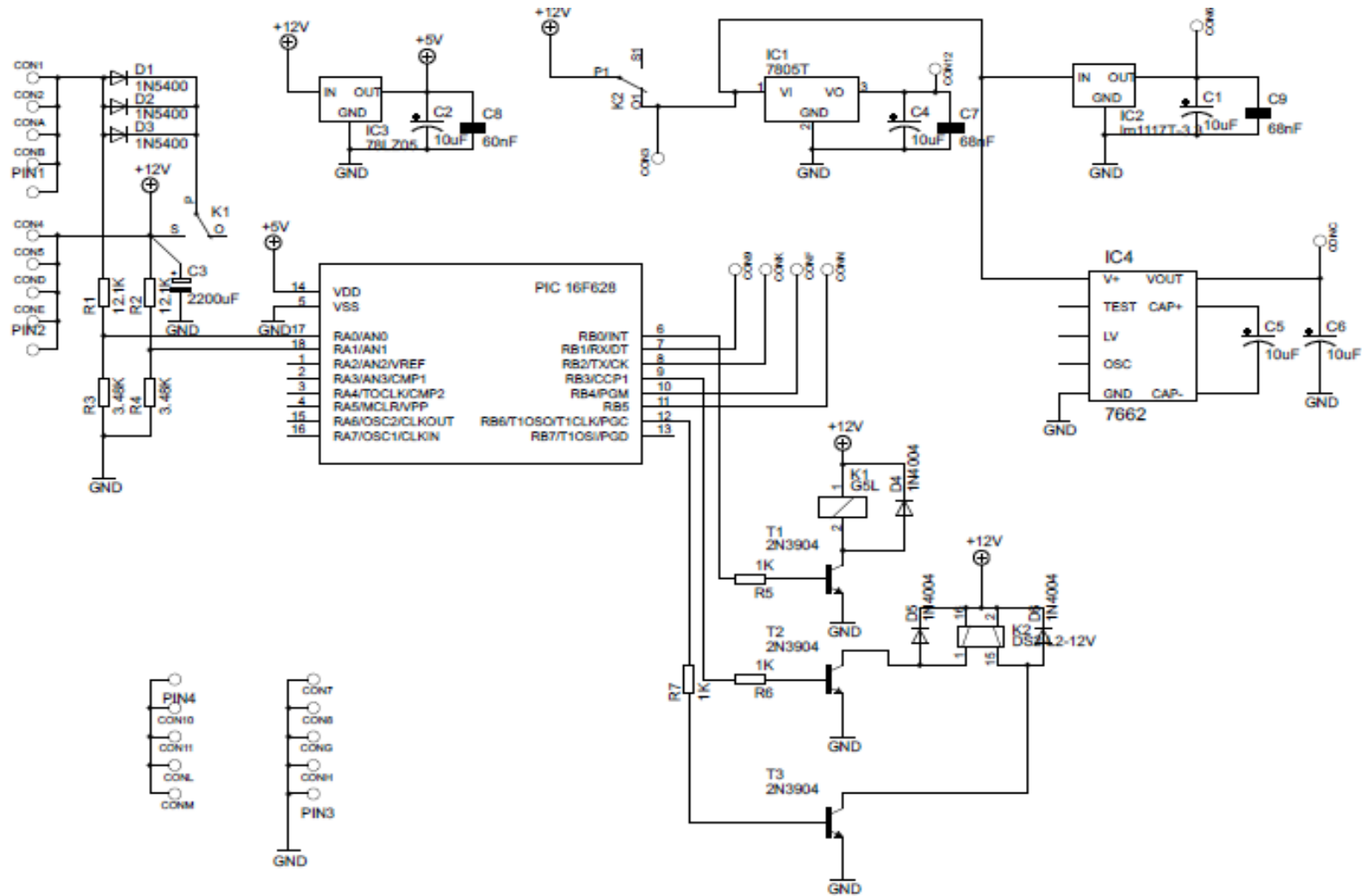


Figura 5.8 esquema del controlador de carga

5.2.2 Amplificador

5.2.2.1 Descripción general del amplificador

El amplificador es el encargado de tomar las señales analógicas provenientes del geófono, controlar el nivel su nivel amplificación para luego enviarlas por SPI a la placa del almacenador siempre y cuando se detecte un evento. La amplificación de las señales está basada en el circuito integrado INA 114.

5.2.2.2 Funcionamiento

El amplificador está basado en el PIC18F2553, el cual es el encargado de digitalizar las tres señales analógicas provenientes del geófono, las mismas que ingresan al PIC cuya asignación de canales es la siguiente: AN0_canal0, AN1_canal1, AN2_canal2. Las señales antes de ingresar a los canales analógicos del microcontrolador deben ser amplificadas, filtradas.

Las tres señales provenientes del geófono ingresan cada una a un amplificador de instrumentación INA114, con la asignación siguiente: canal0_amplificadorU2, canal1_amplificadorU1, canal3_amplificadorU3, el nivel de amplificación es controlado por el PIC18f253 mediante tres multiplexores 4051 IC1 ,IC2 ,IC3, los cuales controlan la ganancia de cada amplificador (amplificadorU2_multiplexorIC1, U1_IC2, U3_IC3), los multiplexores tiene incorporada a su salidas de escalera de 8 resistencias que sirven para setear mediante software un nivel de ganancia específico. El control de los multiplexores se realiza con la asignación de pines del PIC18f2553 RA3_RA4_RC0_multiplexorIC1,RC4_RC5_RC6_multiplexorIC2,RB3_RB3_RB5_multiplexorIC3.

Una vez que la señal sale de la etapa de amplificación pasa a través del filtro antialiasing de Sallen y Key para señales de 500 HZ. Luego se acondiciona la señal a través de los potenciómetros R39, R47, R55 que sirven para sumar una componente de continua de 2.5 VDC. para ajustar su ingreso al PIC.

5.2.2.3 Especificaciones del amplificador

- Tensión de alimentación para la tarjeta +12,-12, 5 y GND
- Resistencias de precisión para calibrar las ganancias del amplificador INA114.
- Tres canales de entrada para la lectura del geófono.
- Tres amplificadores de instrumentación INA114AP.
- Canales de salida para la comunicación SPI con las otras tarjetas.
- Un cristal de cuarzo de 4 MHZ para el PIC16F628A.

5.2.2.4 Descripción de los pines del amplificador

1. AMP_1, AMP_A entrada para el canal 1 del geófono.
2. AMP_2 entrada de 12 VDC para la alimentación de los amplificadores.
3. AMP_3 entrada de -12 VDC.
4. AMP_5, AMP_6 entradas para el canal 2 del geófono.
5. AMP_7, AMP_8 entradas para el canal 3 del geófono
6. AMP_9, AMP_K entrada de GND.
7. AMP_10 activación para la interrupción INT0 del almacenador.
8. AMP_11 salida de datos seriales cuando se da la comunicación SPI.
9. AMP_12 Entrada de 5 VDC.
10. AMP_H Entrada del reloj cuando se establece la comunicación SPI.
11. AMP_L Entrada de datos seriales cuando se estable la comunicación SPI.
12. AMPL_M /CS para activar el amplificador durante la comunicación SPI.
13. AMP_F Entrada de reset proveniente del amplificador.

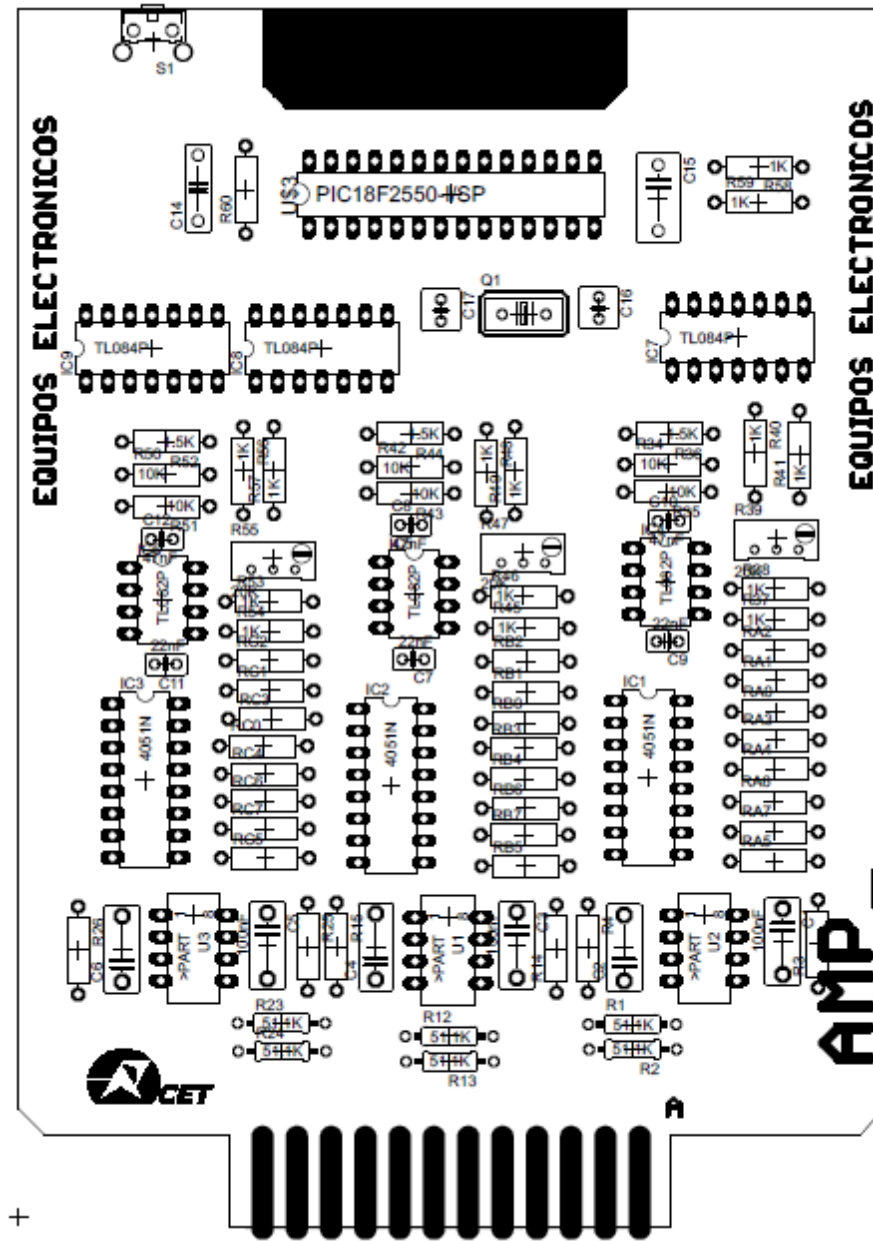


Figura.5.9 Tarjeta del amplificador.

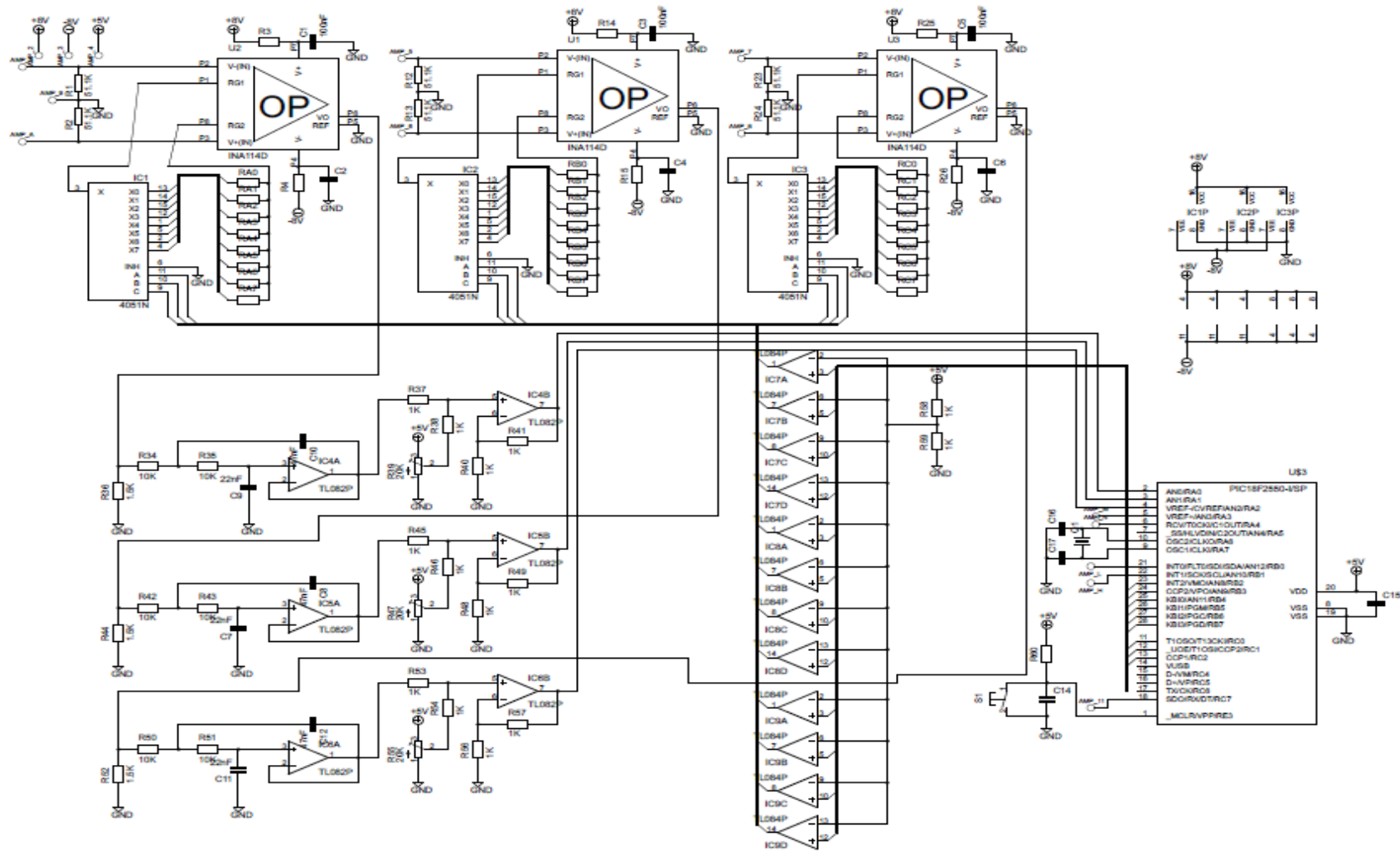


Figura 5.10 Esquema del amplificador



5.2.2 Almacenador

5.2.2.1 Descripción general

Esta tarjeta almacena los datos provenientes del amplificador cuando se ha producido un evento, envía la información necesaria para la visualización en la pantalla GLCD y transmite los datos por la línea telefónica.

5.2.2.2 Funcionamiento

La tarjeta es basada en el PIC18f452, el mismo que está conectado a la tarjeta SDCARD (o MIROSD) por medio de divisores de tensión con la siguiente asignación de pines RC2(PIC)_/CS(SDCARD), RC3_SCK, RC5_SDI y el pin RC4 queda unido de forma directa en el PIN SDO de la SD.

El módem telefónico utiliza el puerto D y A del microcontrolador. El bus multiplexado de datos y direcciones del módem está conectado al puerto D del microcontrolador, el bus de control (ALE, WRITE, READ,CS) con el puerto A con la asignación siguiente ALE_RA0, WRITE_RA1, READ_RA2, /CS_RA5. Para la conexión con la línea telefónica se utiliza un transformador (TR2) de instrumentación con relación 1:1 que proporciona una aislación galvánica.

GPS y el RTC (reloj de tiempo real). El

5.2.2.3 Especificaciones del almacenador

- Tensión de alimentación para la tarjeta +5, 3.3, GND
- Canales de para la comunicación SPI con las otras tarjetas.
- Un relé de 5V para activar o desactivar la línea telefónica.
- Un transformador de aislamiento con relación 1:1.
- Un cristal de cuarzo de 10 MHz y 32 KHz.
- Una memoria SD de 2GB.
- RTC (DS1307)

5.2.2.4 Descripción de los pines de la tarjeta del almacenador

1. COMM2 Salida de reset.
2. COMM8 entrada de interrupción INT1 proveniente del GLCD.
3. COMM9 entrada de GND.
4. COMM10 entrada de interrupción INT0 proveniente del amplificador.
5. COMM11 SDI (entrada de datos seriales cuando se realiza la comunicación por SPI)
6. COMMB TIP
7. COMMC RIN
8. COMMF RX entrada de datos seriales
9. COMMG TX salida de datos seriales
10. COMMH SCK reloj para la comunicación SPI
11. COMMK entrada de GND
12. COMML SDO (salida de satos cuando se realiza la comunicación SPI)
13. COMMM, COMMN salida del chip select.

5.2.3 Tarjeta base

5.2.3.1 Funcionamiento

Posee 3 rack de 12x2 para la colocación de las tarjetas del controlador de carga, amplificador, almacenador, y un jumper de 7x2 donde se coloca el cable de la pantalla GLCD. Se dispone de un decodificador de direcciones formado por el IC 78LS138 para seleccionar la tarjeta con la que se quiere comunicar el almacenador, un jumper 2 que habilitar/deshabilitar el cargador de la batería, tres jumper de 3 pines que selecciona las canales de lectura del geófonos, dos pines (Bat+, Bat-) para conectar la batería de 12 VDC.

5.2.3.2 Especificaciones

- Tres Rack de 12x2
- Un rack de 7x2
- Varios jumpers.
- IC 74LS138
- 1 conector DB9 hembra para placa de circuito impreso
- 1 conector DB9 macho para placa de circuito impreso
- Una batería de 12 VDC.

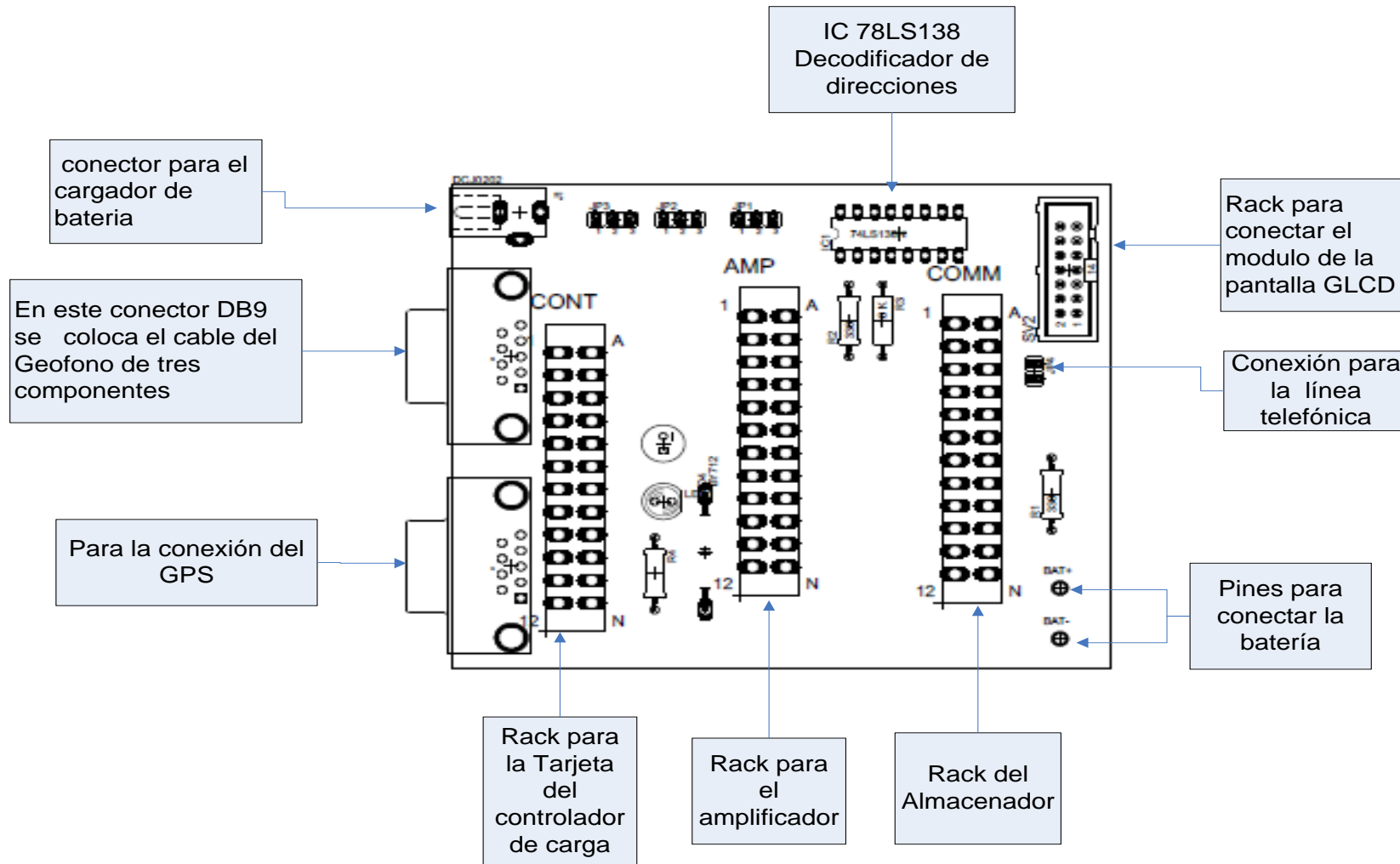


Figura 5.13 Tarjeta base



CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- El instante en el que se da un evento sísmico es un aspecto muy importante, para la obtención de las coordenadas de origen del mismo. Por tal motivo la implementación del GPS en el proyecto, brinda información valiosísima para los analizadores de señales sísmicas. Pero desgraciadamente el tiempo de lectura del GPS(recepción de datos a 4800 bps) es un limitante para el proyecto, debido a que se pierde una considerable parte del evento. Por lo tanto se optó por implementar un reloj de tiempo real (RTC), con lo cual se solucionó en gran parte este inconveniente.
- Las librerías que proporciona MikroC Pro fueron de gran ayuda durante la implementación de cada una de las partes del proyecto, ahorrando tiempo y abaratando su costo final. Pero a pesar de sus innumerables prestaciones también se tuvo inconvenientes, porque no se tiene el control total de las mismas, si bien es cierto realizan una tarea específica pero no se sabe cómo, ni mucho menos se conocen los tiempos de ejecución de las rutinas. Esto trajo muchos problemas a la hora de calcular los tiempos que se utilizan para determinada tarea.
- El almacenamiento de datos provenientes del geófono en la tarjeta SD/MMC es la parte principal del proyecto. El limitante aquí fue el tiempo de adquisición y grabación de los datos, porque estos debían; primero convertirse a punto flotante, luego a código ASCII, para finalmente ser enviados mediante SPI para su almacenamiento. Todo esto tiene que ver con la frecuencia de trabajo de los PICs que intervienen en esta tarea. Este hecho hace que los microprocesadores tengan que trabajar a la máxima velocidad posible, en pos de satisfacer los requerimientos de “frecuencia de muestreo” para los tres canales del geófono.
- El software con el que se conto fue de gran ayuda para las pruebas y simulaciones iniciales de algunas partes del equipo. Conforme se avanzaba en el proyecto esto nos daba ideas de cómo se debía ejecutar e



implementar tal o cual programa. Es el caso concreto de: Proteus, MikroC Pro for PIC, Mplab, CCS.Y para la programación de los PICs: PICkit 2, Mplab.

- La velocidad y memoria de datos de los PICs utilizados es muy limitada, cuando se trata de implementar programas que contienen una gran cantidad de funciones y librerías. Por tal motivo se eliminó uno de los objetivos del proyecto planteado inicialmente, es el caso concreto de la transmisión a través de la línea telefónica. La misma que está probada, en lo que se refiere a sus requerimientos de software y hardware, pero no se grabó en el programa final. Eso queda para microprocesadores que superen dicho inconveniente.
- Las elevadas frecuencias de trabajo de los microprocesadores pueden causar fallas en la ejecución del programa, además del mayor consumo de potencia que se tiene como resultado del incremento de corriente requerida por el PIC.
- Y finalmente todos los conocimientos y experiencias adquiridos a lo largo de la carrera universitaria, fueron las bases principales para la culminación exitosa de esta tesis. Ya que nos brindaron ideas y soluciones ante cualesquier problema que se nos presentó.



RECOMENDACIONES

- Si bien es cierto el internet es una herramienta de consulta muy importante, pero existen algunas páginas que no prestan información bien fundamentada, por lo que se recomienda la verificación del origen de dichas páginas, para no cometer errores en el desarrollo del proyecto.
- Utilizar software actualizado y totalmente funcional, de no ser así se tendrán errores durante las etapas de pruebas e implementación del proyecto. Esto se debe principalmente a: incompatibilidades con el sistema operativo que se está utilizando, aplicaciones que ya no se utilizan, cambio en el hardware necesario para su funcionamiento.
- Una revisión detallada de los manuales y datasheets de los distintos dispositivos a utilizar en el desarrollo del hardware y software. Es responsabilidad del diseñador entender cómo funciona tal o cual elemento para luego incluirlo en el sistema completo.
- Antes de probar determinada parte del equipo se recomienda revisar de forma exhaustiva, que todas las conexiones eléctricas estén correctas: sistema de alimentación, de transmisión, almacenamiento, etc. Teniendo en cuenta que un mal funcionamiento de algunos dispositivos importantes como los microprocesadores nos puede ocasionar fallas en el funcionamiento lógico del programa e incluso a quemar el PIC, dando como resultado mayor tiempo de culminación del proyecto.
- Las elevadas frecuencias de trabajo seleccionadas para los microprocesadores, pueden causar fallas en la ejecución de algunas funciones dentro del programa, sobre todo aquellas que tiene integradas una gran cantidad de instrucciones y por ende solicitudes significativas de memoria RAM y ROM. Por lo que se recomienda utilizar frecuencias intermedias, garantizando el correcto funcionamiento del PIC.
- La fragmentación de las tarjetas de memoria SD/MMC es un aspecto a tener en consideración sobre todo cuando se trata de eventos de corta duración, porque existe la posibilidad de que se pierda ciertos tramos del evento sísmico, que por muy pequeños que sean, siempre serán motivo de análisis. Una vez extraídos y salvaguardados los sismos de la tarjeta SD, se recomienda formatearla, para superar estos inconvenientes.



BIBLIOGRAFIA

- [1] ROBERT L. BOYLESTAD, LOUIS NASHESKY, *Electrónica: Teoría de circuitos y dispositivos electrónicos, octava edición*, PEARSON EDUCACION- PRENTICE HALL, INC, México, 2003
- [2] FERREL G. STREMLER, *Introducción a los Sistemas de Comunicación*, Segunda Edición, Editorial Alfa omega, México, 1989.
- [3] MICROCHIP, *PIC18Fxx228/40pins Data Sheet, High Performance, Enhanced Flash, Enhanced Flash, 10-Bit A/D USB Microcontrollers with nanoWatt Technology*, Microchip Technology Incorporated, DS39564C, U.S.A., 2006
- [4] MICROCHIP, *PIC18F255328/40/44-Pin High Performance, Enhanced Flash, USB Microcontrollers with 12-Bit A/D and nano Watt Technology*, Microchip Technology Incorporated, DS39887B, USA, 2007
- [5] DALLAS, *DS1307 64 X 8 Serial Real Time Clock Data Sheet*, Dallas Technologies Co., LTD, 081800, 2006
- [6] TDK SEMICONDUCTOR CORP, *73K224BL V.22bis/V.22/V.21/Bell 212A/103 Single-Chip Modem w/ Integrated Hybrid*, TDK Semiconductor Corporation, April 2000.
- [7] LITEON, *General Purpose Type Photocoupler 4N25 Series/4N26 Series/4N27 Series/4N28 Series*,
- [8] ST, *TL084A , TL084B GENERAL PURPOSE J-FET QUAD OPERATIONAL AMPLIFIERS*, STMicroelectronics GROUP OF COMPANIES, January 1999
- [9] BURR-BROWN, *INA114 Precision INSTRUMENTATION AMPLIFIER*, 1992 Burr-Brown Corporation, PDS-1142D, 1998.
- [10] FAIRCHILD SEMICONDUCTOR, *DM74LS138 DM74LS139 Decoder/Demultiplexer*, Fairchild Semiconductor Corporation, DS006391, 2000.
- [11] SG12864H (128 DOTS X 64 DOTS)
- [12] ST, *2N3904 SMALL SIGNAL NPN TRANSISTOR*, Fairchild Semiconductor Corporation, February 2003.



- [13] SanDisk Secure Digital Card, Manual de tarjetas SD, en www.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf
- [14] *NMEA Reference Manual*, en <http://www.sparkfun.com/products/9159>
- [15] ETH Zürich Department of Geophysics, Geophysical Data Processing 2.
- [16] Manual de Eagle, Tutorial Cadsoft Computer, INC, Traducido por M. Guadilla, en <http://www.scribd.com/doc/24928754/Tutorial-Spa-EAGLE>

Páginas Web,

www.alldatasheet.com
www.sparkfun.com
www.mikroe.com
www.microchip.com
www.4shared.com
es.wikipedia.org



ANEXOS

ANEXO 1

Datasheet del amplificador de instrumentación INA114P.



INA114

Precision INSTRUMENTATION AMPLIFIER

FEATURES

- LOW OFFSET VOLTAGE: 50μV max
- LOW DRIFT: 0.25μV/°C max
- LOW INPUT BIAS CURRENT: 2nA max
- HIGH COMMON-MODE REJECTION: 115dB min
- INPUT OVER-VOLTAGE PROTECTION: ±40V
- WIDE SUPPLY RANGE: ±2.25 to ±18V
- LOW QUIESCENT CURRENT: 3mA max
- 8-PIN PLASTIC AND SOL-16

APPLICATIONS

- BRIDGE AMPLIFIER
- THERMOCOUPLE AMPLIFIER
- RTD SENSOR AMPLIFIER
- MEDICAL INSTRUMENTATION
- DATA ACQUISITION

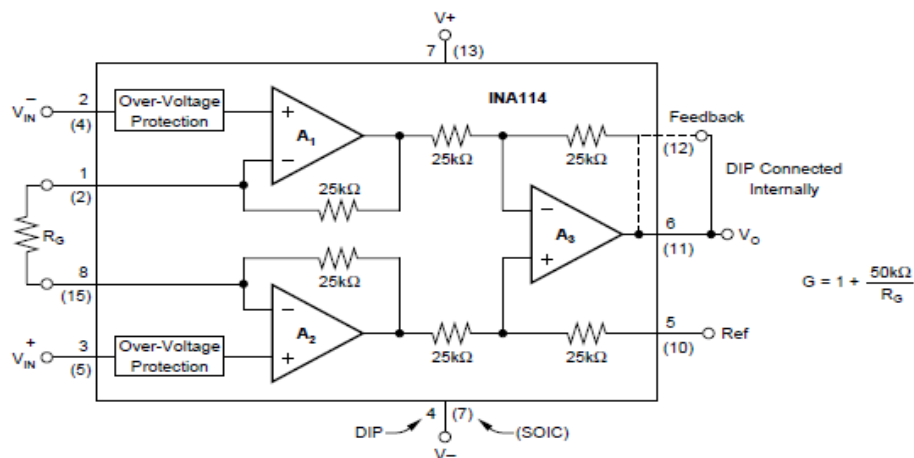
DESCRIPTION

The INA114 is a low cost, general purpose instrumentation amplifier offering excellent accuracy. Its versatile 3-op amp design and small size make it ideal for a wide range of applications.

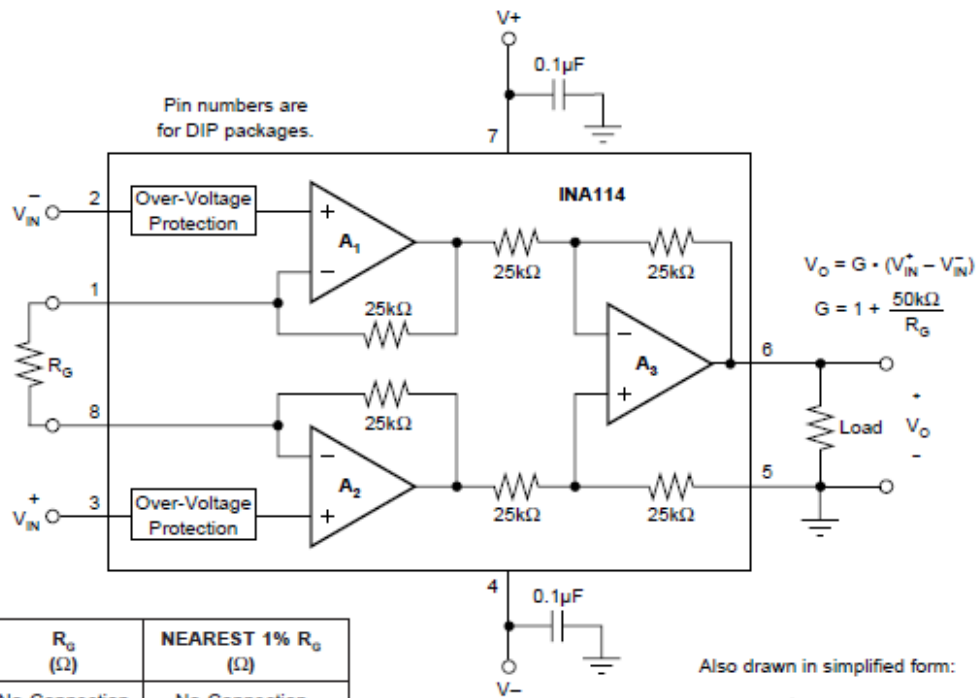
A single external resistor sets any gain from 1 to 10,000. Internal input protection can withstand up to ±40V without damage.

The INA114 is laser trimmed for very low offset voltage (50μV), drift (0.25μV/°C) and high common-mode rejection (115dB at G = 1000). It operates with power supplies as low as ±2.25V, allowing use in battery operated and single 5V supply systems. Quiescent current is 3mA maximum.

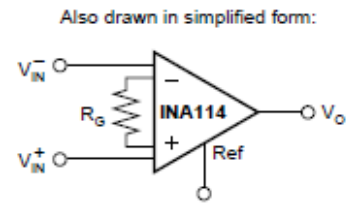
The INA114 is available in 8-pin plastic and SOL-16 surface-mount packages. Both are specified for the -40°C to +85°C temperature range.



Datasheet del amplificador de instrumentación INA114P.



| DESIRED GAIN | R _G (Ω) | NEAREST 1% R _G (Ω) |
|--------------|--------------------|-------------------------------|
| 1 | No Connection | No Connection |
| 2 | 50.00k | 49.9k |
| 5 | 12.50k | 12.4k |
| 10 | 5.556k | 5.62k |
| 20 | 2.632k | 2.61k |
| 50 | 1.02k | 1.02k |
| 100 | 505.1 | 511 |
| 200 | 251.3 | 249 |
| 500 | 100.2 | 100 |
| 1000 | 50.05 | 49.9 |
| 2000 | 25.01 | 24.9 |
| 5000 | 10.00 | 10 |
| 10000 | 5.001 | 4.99 |

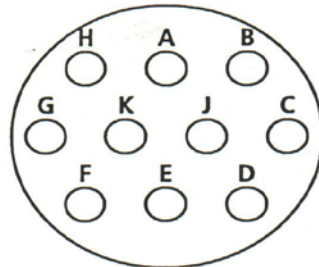


Datasheet del amplificador de instrumentación INA114P (continuación).

ANEXO 2

Módulo de conexión del geófono KPT01E12-10S.

KPT01E12-10S Soldering side / Lötseite




Wire colours / Farben der Adern

| Pin Kontakt | Name | Description Beschreibung | Colour Farbe |
|----------------|-------|--|--|
| A | +OUT0 | positive output channel 0 <i>positiver Ausgang Kanal 0</i> | white <i>weiss</i> |
| B | -OUT0 | negative output channel 0 <i>negativer Ausgang Kanal 0</i> | brown <i>braun</i> |
| C | +OUT1 | positive output channel 1 <i>positiver Ausgang Kanal 1</i> | yellow <i>gelb</i> |
| D | -OUT1 | negative output channel 1 <i>negativer Ausgang Kanal 1</i> | green <i>grün</i> |
| E | +OUT2 | positive output channel 2 <i>positiver Ausgang Kanal 2</i> | pink <i>rosa</i> |
| F | -OUT2 | negative output channel 2 <i>negativer Ausgang Kanal 2</i> | gray <i>grau</i> |
| G | +GSUP | 12 V supply input (average consumption 10 mA for 3D, 3 mA for 1D; see sensor's Inspection Sheet for exact consumption) <i>12 V Versorgungseingang (durchschnittlicher Verbrauch 10 mA bei 3D, 3 mA bei 1D; der genaue Wert kann dem Inspection Sheet des jeweiligen Sensors entnommen werden)</i> | violet <i>violett</i> |
| H | -CAL | "Calibration" input <i>„Kalibrier“-Eingang</i> | blue <i>blau</i> |
| J | SHLD | Signal shield <i>Schirmung für die Signalleitungen</i> | metallic braid <i>metallisches Schirmgeflecht</i> |
| K | GND | Supply ground <i>Versorgungs-Masse</i> | black <i>schwarz</i> |

End of document
Ende des Dokuments



ANEXO 3 Especificaciones técnicas del geófono.



Telefon 07071-9355-0
 Fax 07071-9355-30
 International ++49-7071-9355-30

Lennartz electronic GmbH • Bismarckstrasse 136 • D-72072 Tübingen

Ing. Jose M. Jalil Haas
 Enlazando Ciencia y Tecnologia
 Jorge Cevallos
 Av. Republica de EL Salvador # 525

EC QUITO
 ECUADOR

| | |
|---|--|
| Delivery note: 110587 dt. 02.04.01 Customer : 2238 Your Order : 01601LEN.FUN O/C no. : 125446 dt. 15.03.01 | DELIVERY NOTE Page: 1 Wa / Walker |
|---|--|

| pos. item no. | description | quantity | remarks |
|--|--|----------|---------|
| 10 223-0034 | LE-3Dlite | 1 p | _____ |
| | Ultra-compact three component active geophone, 1 Hz eigenfrequency. Built-in active electronic circuitry provides internal damping and preamplification (400 V/m/s output). 1.5 m cable with KPTC connector (male) included. Requires 12 V DC power supply. S/N: I-447 | | |
| 20 390-0056 | geophone cable, open end | 1 p | _____ |
| | KPTC-cable connector (female), 5 m. | | |
| 1 parcel, marked 125446, net: 5kg, gross: 6kg | | | |
| 40*40*23ccm | | | |
| Geophysical Instruments, Tariff number 9015 80 11. | | | |

ANEXO 4

Datasheet de la pantalla GLCD LM12864B-NSW-BBS.

■ FEATURES

- ◆ BUILT-IN CONTROLLER (KS0107 OR EQUIVALENT)
- ◆ +5 V POWER SUPPLY
- ◆ 1/64 DUTY CYCLE
- ◆ 8-BIT PARALLEL INTERFACE
- ◆ 4.2 V LED FORWARD VOLTAGE

■ MECHANICAL DATA

| ITEM | DIMENSIONS | UNIT |
|-------------------------|-----------------------------|------|
| Module Size (W x H x T) | 75.0 x 52.7 x 7.9 (9.2 LED) | mm |
| Viewing Area (W x H) | 60.0 x 32.5 | mm |
| Active Area (W x H) | 55.01 x 27.49 | mm |
| Dot Size (W x H) | 0.40 x 0.40 | mm |
| Dot Pitch (W x H) | 0.43 x 0.43 | mm |

■ INTERFACE PIN CONNECTIONS

| NO. | SYMBOL | LEVEL | FUNCTION |
|------|------------------|-------|------------------------------|
| 1 | V _{DD} | 5V | Power Supply Voltage |
| 2 | V _{SS} | 0V | Power Supply Ground |
| 3 | V ₀ | - | Contrast Adjustment Voltage |
| 4-11 | DB0-DB7 | H/L | Data Bus Line |
| 12 | CS1 | H | Chip Select Signal For IC1 |
| 13 | CS2 | H | Chip Select Signal For IC2 |
| 14 | /RES | L | Reset Signal |
| 15 | R/W | H/L | H: Read / L: Write |
| 16 | D/I | H/L | H: Data, L: Instruction Code |
| 17 | E | H→L | Enable Signal |
| 18 | V _{out} | - | Power Supply Voltage For LCD |
| 19 | A | 4.2V | LED Power (+) |
| 20 | K | 0V | LED Power (-) |

■ ABSOLUTE MAXIMUM RATINGS

| ITEM | SYMBOL | MIN. | TYP. | MAX. | UNIT |
|------------------------------|----------------------------------|-----------------|------|-----------------|------|
| Supply Voltage For Logic | V _{DD} -V _{SS} | 0 | - | 7 | V |
| Supply Voltage For LCD Drive | V _{DD} -V ₀ | 0 | - | 18 | V |
| Input Voltage | V _i | V _{SS} | - | V _{DD} | V |

■ ELECTRICAL CHARACTERISTICS

| ITEM | SYMBOL | CONDITION | MIN. | TYP. | MAX. | UNIT | |
|--------------------------|----------------------------------|-----------------------------|------|------|------|-----------------|---|
| Supply Voltage For Logic | V _{DD} -V _{SS} | - | 4.5 | 5 | 5.5 | V | |
| Supply Voltage For LCD | V _{DD} -V ₀ | V _{DD} =5V Ta=25°C | 7.8 | 8.6 | 9.2 | V | |
| Supply Current | I _{DD} | V _{DD} =5V | - | 6.2 | 10 | mA | |
| Input Voltage | "HIGH" Level | V _{ih} | - | 2.2 | - | V _{DD} | V |
| | "LOW" Level | V _{il} | - | - | - | 0.6 | V |
| Output Voltage | "HIGH" Level | V _{oh} | - | 2.4 | - | - | V |
| | "LOW" Level | V _{ol} | - | - | - | 0.4 | V |

■ EXTERNAL DIMENSIONS

