

## **TRABAJO ESPECIAL DE GRADO**

### **DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ PARA LA RECOLECCIÓN, ALMACENAMIENTO Y TRANSMISIÓN DE DATOS A UN COMPUTADOR PERSONAL.**

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br. César U. Gutiérrez T.  
para optar al Título de  
Ingeniero Electricista

Caracas, 2016

# **TRABAJO ESPECIAL DE GRADO**

## **DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ PARA LA RECOLECCIÓN, ALMACENAMIENTO Y TRANSMISIÓN DE DATOS A UN COMPUTADOR PERSONAL.**

**Tutor Académico:** Ing. José Romero

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br César U. Gutiérrez T.  
para optar al Título de  
Ingeniero Electricista

Caracas, 2016

# CONSTANCIA DE APROBACIÓN

Caracas, 18 de mayo de 2016

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller CÉSAR GUTIERREZ, titulado:

## **"DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ PARA LA RECOLECCIÓN, ALMACENAMIENTO Y TRANSMISIÓN DE DATOS A UN COMPUTADOR PERSONAL"**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

PROF. SERVANDO ALVAREZ  
JURADO

PROF. SIMÓN MORALES  
JURADO

PROF. JOSÉ ROMERO  
TUTOR ACADÉMICO

Gutiérrez T., César U.

## **DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ PARA LA RECOLECCIÓN, ALMACENAMIENTO Y TRANSMISIÓN DE DATOS A UN COMPUTADOR PERSONAL.**

**Tutor: Ing. José Romero. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica. 2016. 71h + anexos.**

**Palabras claves:** Interfaz; Microcontrolador; Acelerómetro; RS-485; SPI; PIC24.

**RESUMEN.** En los ensayos realizados en estructuras civiles se miden múltiples variables físicas con diversidad de sensores, particularmente en los ensayos dinámicos la medición de la aceleración es muy importante. El Instituto de Materiales y Modelos Estructurales adquirió recientemente un grupo de acelerómetros digitales SUMMIT INSTRUMENTS 35203A que desea incorporar en este tipo de ensayos. El uso simultáneo de más de un acelerómetro es común y la solución dada por el fabricante no lo permite, por lo que se plantea en este trabajo, el diseño de una interfaz capaz de cumplir con ese requerimiento. Se plantea un prototipo electrónico programable, basado en Microcontroladores PIC24 en una topología maestro-esclavo. El micro esclavo establece comunicación con dos acelerómetros bajo Standart de comunicación RS-485, los cuales transmiten los datos adquiridos y este los envía a un maestro vía SPI, el cual los organiza y transmite al computador personal donde son procesados y mostrados en un instrumento virtual de LabVIEW. Con el prototipo diseñado se realizaron pruebas de verificación de funcionamiento de adquisición de datos de los distintos módulos y de la interfaz LabVIEW, de lo cual se concluye que esta interfaz permite gestionar datos provenientes de un par de acelerómetros digitales de manera simultánea, demostrándose así que esta solución es viable para la realización de los ensayos dinámicos en el Instituto.

# ÍNDICE GENERAL

Resumen .....	iii
Índice General .....	iv
Lista de Figuras .....	viii
Lista de Tablas .....	x
Acrónimos .....	xi
Introducción .....	1
Capítulo I.....	3
1    Problema .....	3
1.1    Planteamiento.....	3
1.2    Justificación .....	4
1.3    Objetivos.....	4
1.3.1    Objetivo General.....	4
1.3.2    Objetivo Específicos .....	4
Capítulo II .....	6
2    Marco Referencial.....	6
2.1    El Acelerómetro.....	6
2.1.1    Acelerómetros Electromecánicos .....	7
2.1.2    Acelerómetros Piezoeléctricos.....	7
2.1.3    Acelerómetros Piezoresistivos.....	8
2.1.4    Acelerómetros de Capacitancia diferencial .....	8
2.1.5    Acelerómetros tipo MEM's (Micro Electromecánicos).....	8
2.2    Características.....	10
2.2.1    Rango de Frecuencia.....	10
2.2.2    Sensibilidad.....	10
2.2.3    Rango Dinámico .....	11
2.2.4    Ruido.....	11

2.2.5	Masa.....	11
2.3	Comunicación en Dispositivos Digitales.....	11
2.3.1	Comunicación en Paralelo.....	12
2.3.2	Comunicación Serial.....	12
2.3.3	CRC-16.....	14
2.3.4	UART (Receptor / Transmisor Asíncrono Universal).....	14
2.3.5	EIA-485.....	15
2.3.6	MAX-485.....	15
2.3.7	Comunicación SPI.....	16
2.4	FAT.....	18
2.4.1	Multimedia Memory Card (MMC/SDC).....	19
2.5	Acelerómetro SUMMIT 35203A.....	19
2.6	Configuración de Acelerómetro SUMMIT 35203A.....	21
2.6.1	Ventana Principal de Configuración.....	22
2.6.2	Bloque “Program Function”.....	23
2.7	Protocolo SIP.....	27
2.8	Principales Comandos Utilizados en Protocolo SIP.....	28
2.9	Formas de Comunicación con Sensor por Medios Externos.....	28
2.9.1	Adquisición de medidas del Sensor Digital (Acquiring Digital Sensor Measurements).....	29
2.9.2	Adquisición de Medidas Usando Scripts.....	30
2.9.3	Adquisición de Datos en Modo High Speed.....	31
Capítulo III	.....	34
3	Definición del sistema.....	34
3.1	Definición del Hardware.....	35
3.1.1	Módulo Esclavo.....	36
3.1.2	Modulo Maestro.....	39
3.1.3	Modulo FAT.....	40
3.1.4	Modulo Fuente de Alimentación.....	40

3.2	Definición del Software .....	40
3.2.1	Módulo Esclavo .....	40
3.2.2	Módulo Maestro.....	41
3.2.3	Interfaz PC .....	41
3.3	Descripción del Hardware .....	42
3.3.1	Módulo Esclavo .....	42
3.3.2	Módulo Maestro.....	44
3.3.3	Módulo de Alimentación de Tensión.....	45
3.4	Descripción de Software.....	46
3.4.1	Módulo Esclavo .....	46
3.4.2	Módulo Maestro.....	52
3.4.3	Sincronización de Interfaz .....	55
3.4.4	Interfaz PC .....	58
Capítulo IV	.....	61
4	Pruebas y Resultados .....	61
4.1	Pruebas de comunicación con acelerómetros SUMMIT 35203A .....	61
4.2	Administración de inicialización con acelerómetros SUMMIT 35203A .....	62
4.3	Transmisión SPI Maestro-Esclavo.....	63
4.4	Ajuste de tamaño de trama requerida por SPI .....	63
4.5	Recepción de datos en computador personal.....	65
4.6	Almacenamiento de Datos .....	66
Conclusiones	.....	70
Recomendaciones.....		72
<b>PROPUESTA DE SOLUCIÓN A PROBLEMÁTICA EN RESULTADOS</b>		
OBTENIDOS.....		73
•	Agregar byte de muestreo a trama de muestra.....	73
•	Bajar velocidad de transmisión serial a 57600 baudios .....	73
•	Reconfiguración a modo Maestro SPI (Maestro) .....	73

• Configuración de interrupción de recepción serial en modo 3 para interrupción al tener FIFO lleno cada recepción de 4 bytes. ....	73
• Implementación de buffer cíclico (Estudiar tamaño según tiempos de funcionamiento) sólo de recepción de bytes en rutina de interrupción Rx.....	73
• Implementación de función de verificación de muestras validas de datos en buffer recibido Rx y paso de los mismos a Buffer Val_Rx. Se requiere que la función verifique y valide las muestras recogidas en los datos colectados y descarte muestras erróneas. ....	73
• Implementación de función para ensamblado de trama, la función debe armar supertrama con cabecera para identificación, tomar las muestras de los buffers Val_Rx1 y Val_Rx2 de ambos canales e intercalar muestras de canal 1 y canal 2 en construcción de supertrama. ....	73
• Implementación de función de intercambio SPI maestro-esclavo, para enviar al nuevo módulo esclavo datos una vez armada una supertrama. ....	74
Bibliografía .....	76
5 Anexos .....	78
5.1 [Notas Rápidas sobre uso de ICU].....	79
5.2 [MANUAL DE OPERACIÓN] .....	85
5.3 [Pinout de Microcontrolador Maestro, Esquemático y Layout] .....	92
5.4 [Pinout de Microcontrolador Esclavo].....	94



## LISTA DE FIGURAS

Figura 1. Algunos tipos de Acelerómetros.....	9
Figura 2. Acelerómetro tipo MEM's.....	10
Figura 3. Comunicación Serial y Paralelo.....	12
Figura 4. Forma de Dato Serial.....	15
Figura 5. Integrado Max 485.....	16
Figura 6. Topología de comunicación SPI.....	17
Figura 7. Modos de Comunicación SPI.....	18
Figura 8. Acelerómetro SUMMIT 35203A.....	20
Figura 9. Ventana principal de configuración About/Setup ICU.....	23
Figura 10. Ventana ICU On-Line Configuration.....	25
Figura 11. Opciones de Telemetría (On-Line Configuration).....	26
Figura 12. Parámetros de Canales (On-Line Configuration).....	26
Figura 13. Sistema solución.....	34
Figura 14. Esquema de módulos de sistema solución.....	35
Figura 15. Trama muestra enviada por sensor acelerómetro.....	37
Figura 16. Diagrama de módulo esclavo.....	43
Figura 17. Diagrama de modulo Maestro.....	45
Figura 18. Diagrama de módulo de alimentación.....	45
Figura 19. Diagrama de flujo de Rutina Principal Módulo Esclavo.....	47
Figura 20. Trama Muestra con cambio de cabecera de trama para señalización de canal de adquisición.....	48
Figura 21. Trama de datos en arreglo SPI.....	49
Figura 22. Subrutina de ensamblado de supertrama SPI.....	50
Figura 23. Rutina de recepción serial.....	51
Figura 24. Rutina de interrupción SPI microcontrolador esclavo.....	52
Figura 25. Diagrama de flujo de programa principal de módulo maestro.....	53

Figura 26. Interrupción de timer encargada de realizar el envío de datos a PC y módulo FAT.....	55
Figura 27. Tiempo $t_a$ de vaciado de arreglos de datos seriales. ....	56
Figura 28. Tiempo de vaciado $\Delta t_1$ de arreglos de datos de llegada y de requerimientos desde Maestro.....	57
Figura 29. Diagrama de tiempo completo de la dinámica de transferencia de datos en funcionamiento de interfaz.....	58
Figura 30. Proceso de conversión y presentación de datos en PC .....	60
Figura 31. Supertrama final fijada luego de ajuste por observación de comportamiento.....	65
Figura 32. Captura de datos recibidos en computador personal .....	68
Figura 33. Datos presentados en Interfaz PC usando paquete de Software LabVIEW. ....	68
Figura 34. Captura de datos para eje 1 aceleración negativa. ....	69
Figura 35. Captura de datos para aceleración eje 3 negativa. ....	69

## **LISTA DE TABLAS**

Tabla 1. Modos de Configuración SPI.....	18
Tabla 2. Especificaciones de acelerómetro SUMMIT 35203A. ....	20

## ACRÓNIMOS

**CRC:** Cyclic Redundancy Check.

**FAT:** File Allocation Table.

**ICU:** Instrument Configuration Utility.

**LED:** Light-emitting Diode.

**MIPS:** Millions of Instructions per Second.

**PC:** Personal computer.

**SPI:** Serial Peripheral Interface.

**UART:** Universal Asynchronous Receiver Transmitter.

**VISA:** Virtual Instrument Software Architecture.

## INTRODUCCIÓN

El Instituto de Materiales y Modelos Estructurales (IMME) adscrito a la Universidad Central de Venezuela, es una Institución que se dedica al estudio de características y desempeño de estructuras civiles en cualquier parte del territorio nacional; para algunos estudios realizados por este instituto, se contempla la recolección de datos de diferentes sensores con el objetivo de conocer el comportamiento de variables físicas tales como desplazamiento, velocidad,, aceleración, fuerza y presión.

Actualmente el IMME posee 10(diez) acelerómetros marca SUMMIT INSTRUMENTS modelo 35203A; los cuales cuentan con características significativas para las investigaciones del Instituto. En algunos de los ensayos llevados a cabo por el IMME se requiere manejar de forma simultánea la recolección de datos de los acelerómetros de más de 2 sensores hacia un computador personal; lo cual, no es posible con la interfaz que ofrece el fabricante (SUMMIT INSTRUMENTS modelo 35203 A), por lo tanto, se propuso el diseño de una interfaz que cuenta con esta característica además de otras deseables como son: el almacenamiento de datos, a fin de proporcionar un mayor desempeño en los estudios llevados a cabo por el IMME.

En la propuesta de desarrollo de este proyecto se contempla el diseño de un sistema digital que gestione, transmita, comunique, recolecte y almacene datos obtenidos por los acelerómetros SUMMIT 35203A hacia un computador personal para su posterior análisis.

En el capítulo 1, se presentan tanto el planteamiento del problema, la justificación, el alcance y los objetivos del presente trabajo.

En el capítulo 2, se cubren todos los aspectos teóricos y referenciales necesarios para la consecución del presente proyecto.

En el capítulo 3, se establecen las definiciones de software y hardware necesarios para alcanzar los objetivos planteados así como la descripción de todos los elementos que integran la solución propuesta.

En el capítulo 4, se definen las pruebas realizadas para la verificación del equipo, se presentan los resultados obtenidos y el análisis de los mismos.

Como parte final del trabajo se presentan las conclusiones así como las recomendaciones y los anexos.

# CAPÍTULO I

## 1.1 Planteamiento

El Instituto de Materiales y Modelos Estructurales (IMME), realiza estudios referentes a características dinámicas a diferentes tipos de estructuras y materiales, donde el uso de herramientas como los acelerómetros es de vital importancia para los análisis y estudios que ejecuta.

El IMME ha adquirido acelerómetros marca SUMMIT INSTRUMENTS modelo 35203A, dichos acelerómetros poseen características convenientes para los estudios realizados por el instituto tales como: medición de aceleración en 3(tres) ejes coordenados, salida digital bajo estándar RS-485, alimentación de tensión continua entre +8[Volt] ~ +30 [Volt] y rango de escala configurable entre +1g y +5g como características más relevantes; no obstante, también presentan la desventaja en la limitación de uso de un (1) solo acelerómetro con la interfaz que ofrece el fabricante (SUMMIT INSTRUMENTS modelo 35250A), mientras que el propósito deseado es recolectar datos de más de un sensor. Por lo cual se plantea lo siguiente ¿es posible realizar una interfaz que pueda manejar más de un acelerómetro? Además, ¿es posible establecer comunicación entre la interfaz y el computador mediante alguno de los puertos del computador?, ¿es posible que la interfaz además de establecer comunicación entre el computador y el acelerómetro también pueda almacenar datos? y, ¿puede realizarse además una interfaz de usuario para la administración de funcionamiento de los sensores por medio de la interfaz diseñada desde el computador personal?

## **1.2 Justificación**

Actualmente el IMME cuenta con 10 (diez) acelerómetros (marca SUMMIT INSTRUMENTS modelo 35203A), es necesario un sistema de gestión para la recolección, almacenaje y presentación de datos de los diferentes sensores que permita el flujo de información entre los sensores el computador además de un programa de monitoreo de los datos en el computador, flexible y adaptable a diferentes condiciones de prueba.

Los sensores que posee el IMME no son utilizados de manera conjunta debido a la limitación de la interfaz suministrada por el fabricante, por lo que la propuesta de diseño y construcción de una nueva interfaz permitirá aumentar el número de sensores simultáneos utilizados, además de agregar funcionalidades aprovechando la experiencia del personal del Instituto en este tipo de sistemas, presentando de esta manera un beneficio para el IMME en la cantidad de información recolectada con los acelerómetros destinados para tal fin.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Diseñar y construir una interfaz para la recolección y transmisión de datos de los acelerómetros SUMMIT 35203A hacia un computador personal.

### **1.3.2 Objetivo Específicos**

- Determinar las metodologías de comunicación, recolección y almacenamiento de datos en dispositivos digitales para



administración de funcionamiento de acelerómetro SUMMIT 35203A.

- Seleccionar la metodología para el establecimiento de comunicación, recolección y almacenamiento de datos entre componentes Acelerómetro-Interfaz-Computador.
- Construir dispositivo electrónico a partir de la metodología seleccionada.
- Desarrollar Firmware de los dispositivos digitales.
- Diseñar e implementar software para presentación de datos adquiridos a través del dispositivo diseñado en el computador.
- Elaborar manual de operación del dispositivo

## CAPÍTULO II

El diseño del equipo presentado en este trabajo implica el conocimiento de conceptos necesarios que fundamentan la concepción del sistema desarrollado como solución al problema planteado, por lo cual a continuación se presentarán aspectos teóricos relevantes.

En sistemas en los que intervienen cantidades de aceleración, velocidad y distancia recorrida como: sistemas de vibración, mediciones de choque o impacto, estudios sísmicos, análisis nodales, entre otros las magnitudes medidas son aceleraciones y su medición es llevada a cabo por acelerómetros.

### 2.1 El Acelerómetro

Es un instrumento construido con la finalidad de medir aceleraciones debidas a movimientos y vibraciones; el basamento físico de todos los acelerómetros es la acción de aceleración que se genera sobre una masa producida por una fuerza, este hecho está contenido en la segunda ley de Newton que enuncia lo siguiente "El cambio de movimiento es proporcional a la fuerza motriz impresa y ocurre según la línea recta a lo largo de la cual aquella fuerza se imprime"

Este enunciado se resume en la siguiente ecuación

$$F = m \cdot a$$

dónde:

$F$ : Fuerza medida en Newtons.

$m$ : Masa de prueba en Kg.

$a$ : Aceleración en  $m/seg^2$ .

Esta fórmula establece entonces que la fuerza es directamente proporcional a la aceleración generada en función de la masa.

Los acelerómetros como tal cuentan con una masa referencial dentro del instrumento sobre la cual se realiza la medición mediante diferentes técnicas según la tecnología de construcción, para así obtener el valor de aceleración asociado a las fuerzas de interés. Según la tecnología de construcción se pueden señalar los siguientes tipos de acelerómetros:

### **2.1.1 Acelerómetros Electromecánicos**

Este tipo de acelerómetro se caracteriza por medir desplazamientos de una masa (sistema mecánico) de manera indirecta mediante un dispositivo transductor, sensando las variaciones de magnitudes del circuito eléctrico asociado al sistema mecánico, en el cual se generan corrientes o fuerzas magnéticas que pueden ser determinadas, y mediante las cuales se puede conocer el valor de aceleración asociado.

### **2.1.2 Acelerómetros Piezoeléctricos**

Este tipo de acelerómetros utiliza una masa que se encuentra en contacto directo con un componente piezoeléctrico, de modo que cuando existe movimiento asociado al acelerómetro el componente piezoeléctrico experimenta una variación de movimiento proporcional a la fuerza de excitación, esta a su vez es proporcional a la variación de carga eléctrica en el componente piezoeléctrico, por lo que, a través de la medición de variación de carga eléctrica se determina el valor de aceleración.

### **2.1.3 Acelerómetros Piezoresistivos**

Los acelerómetros piezoresistivos obtienen el valor de aceleración por medio de la medición de variación de resistividad que existe por la compresión del material piezoresistivo durante el movimiento de excitación, por lo tanto estableciendo la relación existente entre la fuerza y la variación de resistividad se obtiene el valor de aceleración.

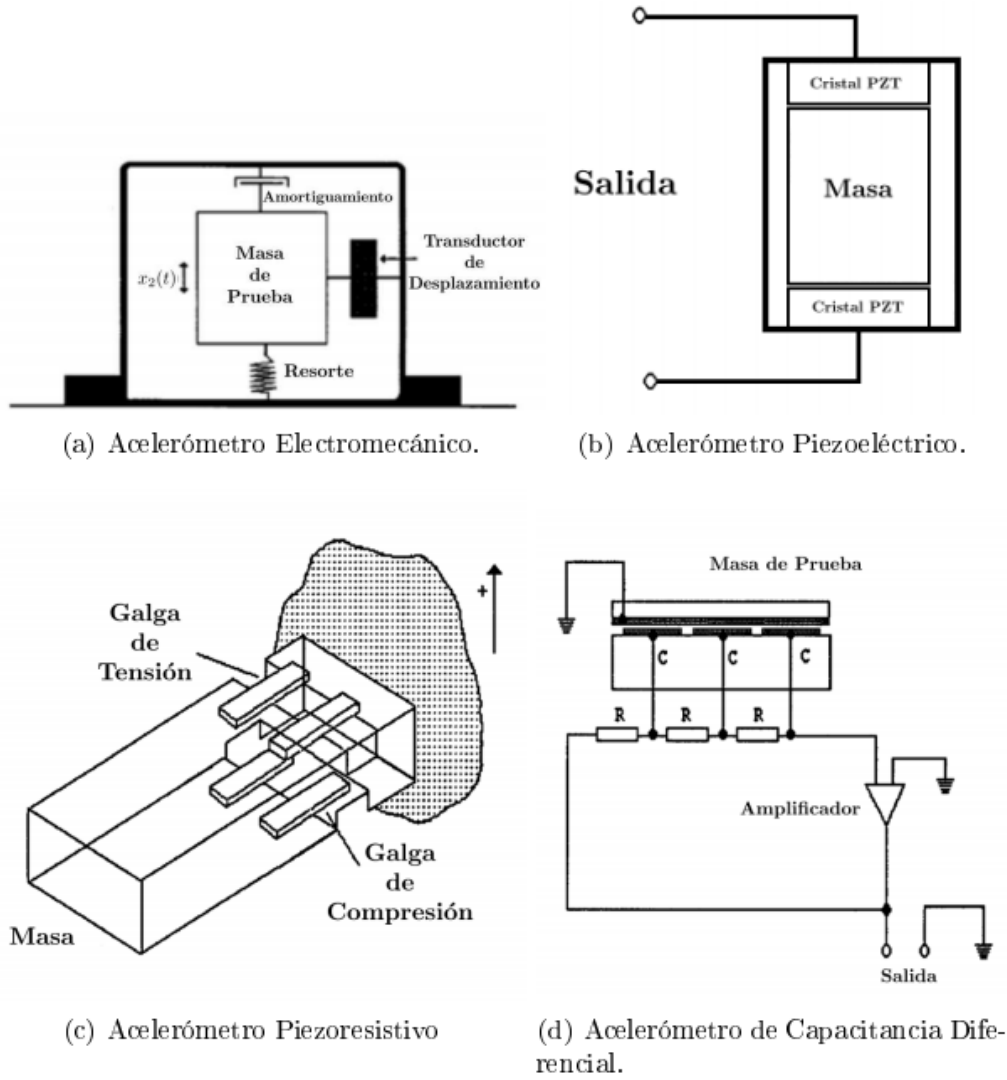
### **2.1.4 Acelerómetros de Capacitancia diferencial**

El principio de funcionamiento de los acelerómetros de capacitancia diferencial es la variación de capacitancia en proporción a la aceleración aplicada que se da en una configuración movable en un arreglo en el cual uno de los polos está fijo y el otro experimenta movimiento debido a la fuerza de excitación.

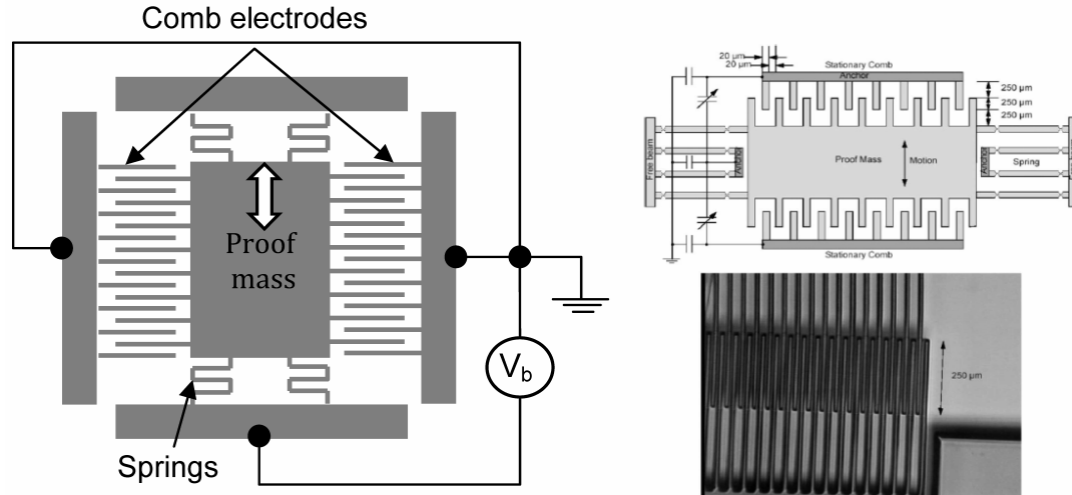
### **2.1.5 Acelerómetros tipo MEM's (Micro Electromecánicos)**

Los acelerómetros microelectromecánicos "Micro Electro Mechanical" como se les conoce en Estados Unidos y el resto de América o "Microsystems Technology" en Europa son un tipo de acelerómetros fabricado bajo una tecnología de miniaturización conocida como "Micromachinig", básicamente es posible hacer estructuras electromecánicas a niveles de miniaturización en la misma forma en la que se construyen los circuitos integrados, sin embargo, el principio de medición es similar al utilizado en otros tipos de acelerómetros, una masa de prueba que en presencia de una fuerza experimenta una aceleración, en este caso el sistema electromecánico sobre el que se mide la fuerza de prueba se construye en base a varias capas de sustrato y diferentes tipos de material, que bajo diferentes procesos químicos resulta en una estructura electromecánica a nivel microscópico que se rige bajo las mismas reglas físicas ya mencionadas, el nivel de miniaturización de este

tipo de estructuras permite la integración con otros tipos de circuitos integrados. En Figura 1 se muestran algunos tipos.



**Figura 1. Algunos tipos de Acelerómetros.**



**Figura 2. Acelerómetro tipo MEM's**

## 2.2 Características

Entre algunas de las características de los acelerómetros que se deben tener en cuenta se tienen las siguientes.

### 2.2.1 Rango de Frecuencia

Las mediciones de aceleración por lo general se limitan al uso de una porción lineal de la curva útil de la respuesta del instrumento, esta respuesta está limitada en bajas frecuencias y altas frecuencias por frecuencias de resonancias naturales y este rango de trabajo es conocido como rango de frecuencia.

### 2.2.2 Sensibilidad

Sensibilidad es la mínima magnitud en la señal de entrada requerida para producir una unidad de magnitud en la señal de salida, en los acelerómetros también puede ser conocida como “factor de escala”.

### **2.2.3 Rango Dinámico**

Se entiende como rango dinámico al rango de aceleración entre los límites superior e inferior dentro del cual el acelerómetro responde linealmente antes que la señal sea distorsionada o recortada.

### **2.2.4 Ruido**

El ruido es una señal aleatoria y de pequeña amplitud que es generada por la misma composición e interacción de parámetros internos y externos propios del instrumento y que en casos en los que las señales medidas sean muy pequeñas pueden ser enmascaradas por la señal de ruido propia del instrumento.

### **2.2.5 Masa**

La masa del instrumento puede llegar a cobrar importancia cuando la masa del objeto es pequeña, debido a que la masa del instrumento puede afectar el comportamiento del sistema, como regla general se recomienda que la masa del acelerómetro sea 10 veces menor a la masa de la estructura u objeto a la que se quiere realizar la medición.

## **2.3 Comunicación en Dispositivos Digitales**

Las comunicaciones en dispositivos electrónicos se realizan utilizando código binario enviando información en forma de bits, tomando valores de “cero” o “uno” que electrónicamente son interpretados entre dispositivos como una tensión de referencia tomada como “baja” o una tensión de referencia tomada como “alta”, en función de esto se pueden señalar dos maneras de comunicación.

### 2.3.1 Comunicación en Paralelo

En la comunicación en paralelo cada bit del código binario es transmitido al mismo tiempo por diferentes hilos de comunicación entre dos dispositivos como se muestra en la Figura 3.

### 2.3.2 Comunicación Serial

En la comunicación serial cada uno de los bits es enviado por un solo hilo de comunicación entre dispositivos. Ver Figura 3.

Entre la comunicación paralelo y la serial existen ventajas y desventajas que deben ser tomadas en cuenta en su uso, la comunicación paralela resulta mucho más rápida que la comunicación serial debido a que en un instante  $t_0$  la información deseada puede ser transmitida completamente, mientras que en la comunicación serial se necesitan  $nx t_0$  instantes para transmitir la misma información deseada, la diferencia radica en que en la comunicación en paralelo el número de hilos de comunicación puede resultar inconveniente frente a un solo hilo de comunicación de la comunicación serial.

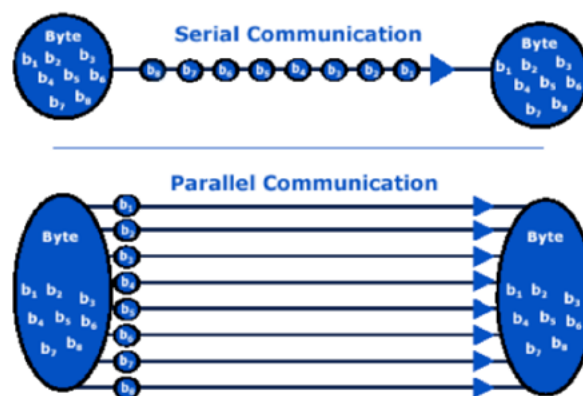


Figura 3. Comunicación Serial y Paralelo



## **Modos de transmisión**

Para los circuitos de comunicación digital se tienen cuatro tipos de transmisión:

### **Simplex**

La información se realiza en una sola dirección, sólo para transmitir o sólo para recibir.

### **Half Duplex (HDX)**

En el modo Half Duplex la transmisión es posible en ambos sentidos, pero no al mismo tiempo.

### **Full Duplex (FDX)**

En el modo full dúplex las transmisiones son posibles en ambos sentidos simultáneamente (ambos sentidos de comunicación con una misma estación).

### **Full/full Duplex (F/FDX)**

En el modo Full/full dúplex las transmisiones son posibles en ambas direcciones, pero puede haber involucrada una tercera estación (mientras se recibe dato de una estación, simultáneamente se puede estar enviando dato a otra estación diferente)

### 2.3.3 CRC-16

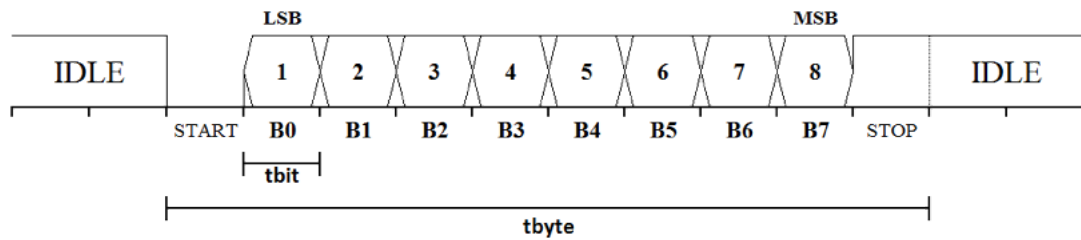
Es un esquema de detección de errores conocido como “Chequeo de Redundancia Cíclica” con un porcentaje de detección de errores de aproximadamente 99.95%. Este proceso utiliza 16 bits para establecer una secuencia de chequeo de bloque (BCS) donde se asocia el mensaje a un polinomio  $G(x)$  que es dividido por un polinomio generador  $P(x)$ , el cociente es descartado y el residuo se trunca a 16 bits y se agrega al mensaje como secuencia de chequeo de bloque (BCS), de manera que al enviar este bloque en el mensaje, el receptor puede confirmar la integridad del mensaje por medio del cálculo y comparación del CRC del mensaje recibido.

### 2.3.4 UART (Receptor / Transmisor Asíncrono Universal)

Es un circuito que se encarga de la transmisión de datos seriales asíncronos entre dos dispositivos, está conformado por un transmisor y un receptor; realiza las siguientes funciones:

- a) Realizar conversión de datos de serial a paralelo y paralelo a serial.
- b) Realizar detección de errores en bits de paridad.
- c) Insertar y detectar bits de arranque y parada.

La estructura del mensaje en forma serial se ha estandarizado, en principio el mensaje está conformado por un bit de inicio correspondiente a un cero lógico seguido de 8 bits de dato que conforman un byte, empezando por el bit menos significativo (LSB) hasta el bit más significativo (MSB) un bit de paridad que puede o no ser utilizado y luego un bit de parada que corresponde a un uno lógico.



**Figura 4. Forma de Dato Serial**

### 2.3.5 EIA-485

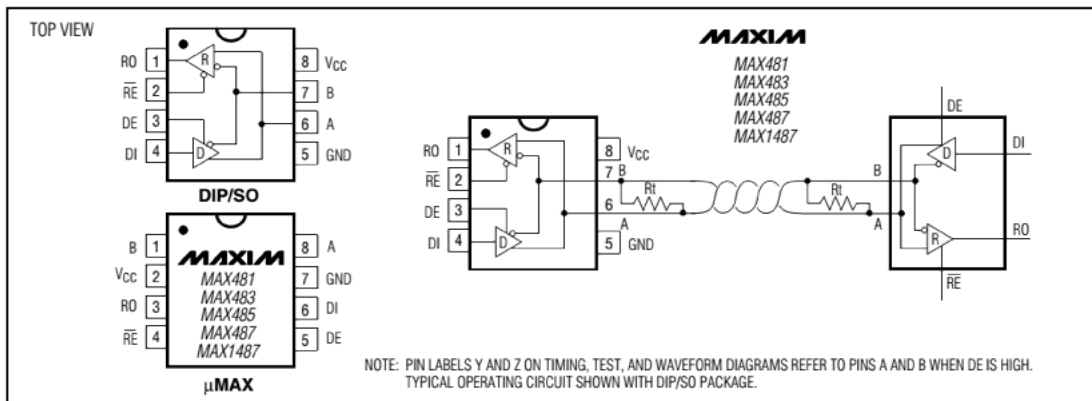
El estándar de transmisión balanceada 485 fue desarrollado en 1983 para implementar comunicación de un host a sus periféricos, éste especifica sólo la capa física. El protocolo, tipo de transmisión, conector y demás aspectos de la comunicación son dejados para ser definidos por el diseñador.

Originalmente el estándar 485 fue definido como una actualización del estándar 422, donde solo se permitía comunicación en modo simplex, el 485 permite comunicación multipunto, facilitando también comunicación half-duplex, entre las características más resaltante de este estándar está que permite un cableado de hasta 1.2 Km y velocidades de comunicación hasta 10 Mbits/seg

### 2.3.6 MAX-485

El chip MAX-485 es un integrado diseñado para establecer comunicación serial bajo parámetros de la norma RS-485. La norma RS-485 establece comunicación asíncrona por medio un par trenzado (A y B) donde se interpretan diferencias de tensión como valores lógicos. Al transmitir en modo diferencial, cuando la tensión en A es 0,2 volts superior a B se estará interpretando un “1 lógico”,

en caso contrario, cuando B es 0.2 volts superior a la tensión en A se interpreta un “0 lógico”. Es necesario controlar los terminales DE(habilita envío) y RE(habilita recepción) que puede realizarse al unir los pines debido a que presentan una lógica inversa, por lo que puede controlarse de este modo la transmisión y recepción sobre un solo pin habilitador, en la transmisión se coloca el pin habilitador en “1 lógico” y se envía la trama de dato deseada por el pin 4, en el caso de la recepción el pin habilitador se coloca en un “0 lógico” y se recibe la trama de dato por el pin 1, para proveer de una adaptación de impedancia se coloca una resistencia generalmente de 120  $\Omega$ .



**Figura 5. Integrado Max 485**

### 2.3.7 Comunicación SPI

La comunicación SPI viene de sus siglas en ingles “Serial Peripheral interface” que significa Interface Serial Periférica; es un protocolo maestro-esclavo diseñado para controlar varios periféricos dentro de un mismo bus de datos, su forma

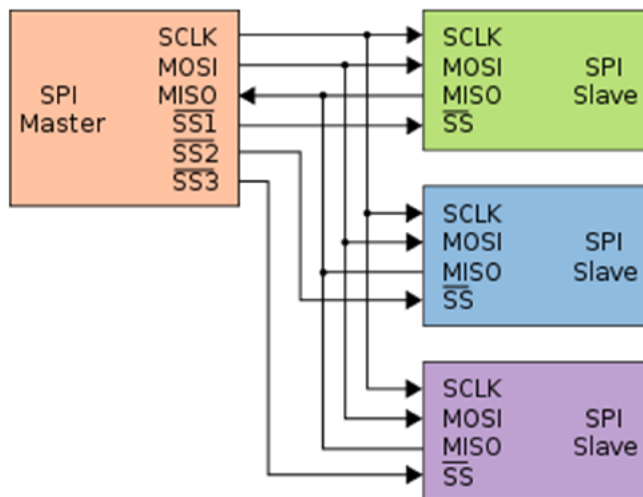
de comunicación implica cuatro hilos de comunicación pero también pueden ser utilizado tres en caso de tener un solo dispositivo esclavo, los cuatro hilos de comunicación tienen las siguientes características

**CLK (Clock):** control de operaciones

**MISO (Master in, Slave Out):** línea de recepción de datos.

**MOSI (Master Out, Slave In):** línea de transmisión de datos.

**SS (Slave Select):** línea de habilitación de esclavo



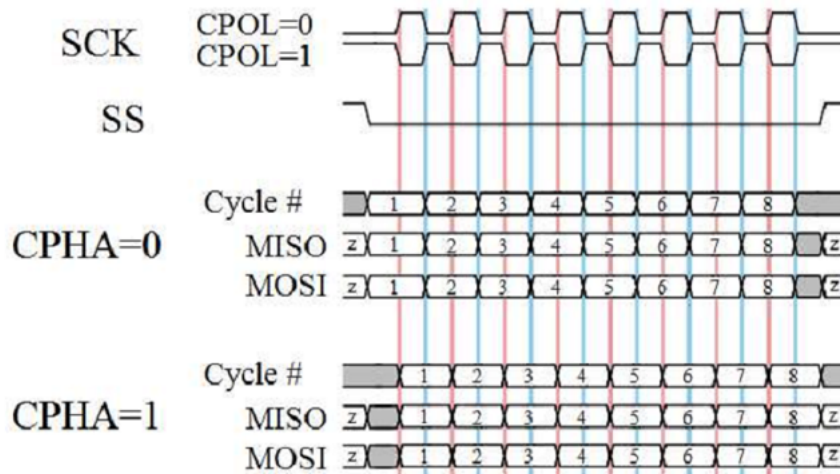
**Figura 6. Topología de comunicación SPI**

Al colocar un "0" lógico en el terminal SS se selecciona el esclavo con el cual se iniciará comunicación, el maestro provee una señal de reloj que gobernará el proceso de transmisión mediante el terminal CLK, mediante registros llamados CPOL(polaridad) y CPHA(fase) se puede configurar cuando puede cambiar el dato y cuando es válido leer el dato, de allí se desprenden cuatro modos de funcionamiento

SPI (Tabla 1. Modos de Configuración SPI Tabla 1), en los modos 0 y 3 el dato es enviado en los flancos de subida de CLK(SCK), mientras que en los modos 1 y 2 el dato es enviado en los flancos de bajada de la señal CLK(SCK), el protocolo SPI es full dúplex por lo que al transmitir un dato también se recibe un dato desde el esclavo.

**Tabla 1. Modos de Configuración SPI**

Modo SPI	<i>CPOL</i> (polaridad)	<i>CPHA</i> (fase)
0	0	0
1	0	1
2	1	0
3	1	1



**Figura 7. Modos de Comunicación SPI**

## 2.4 FAT

Es un sistema para la organización de la información dentro de un medio de almacenamiento, sus siglas en ingles corresponden a “File Allocation Table” (FAT). Fue desarrollado en 1977, por Marc McDonald y Bill Gates; su primera versión fue

el FAT 12 que era utilizado para el almacenamiento en medios flexibles, luego le siguieron el FAT16 y FAT32 que han sido implementados para discos duros, memorias de estado sólido como SD, MMC , Flash Drive. Actualmente es un formato ampliamente usado debido a su compatibilidad y simplicidad de uso en dispositivos electrónicos con necesidades de almacenamiento

#### **2.4.1 Multimedia Memory Card (MMC/SDC)**

Son un estándar de memorias digitales no volátiles utilizadas en equipos móviles que pueden ser borradas y reprogramadas de forma eléctrica. Son conocidas como MMC de sus siglas en inglés “Multi Media Card”, luego se desarrollaron las memoria SD o Secure Digital Card como una modernización de las MMC por lo cual son compatibles con las MMC; están disponibles en tres tamaños RS-MMC, MiniSD, microSD. Las MMC/SDC contienen un microcontrolador dentro que controla los procesos de control en la memoria flash (escritura, lectura, borrado, control de errores). Los datos son transferidos entre el Host y la memoria en bloques de 512 bytes, el sistema de archivos sobre el cual operan es FAT12/16 y FAT32 en el caso de Memorias mayores a tamaños de 4 Gigabytes de memoria.

#### **2.5 Acelerómetro SUMMIT 35203A**

El instrumento SUMMIT 35203A es un acelerómetro triaxial construido por la casa SUMMIT INSTRUMENT, pertenece a la clasificación de tipo MEM's, posee cuatro canales (medición de aceleración en tres ejes “X”, ”Y”, ”Z” y un canal que reporta la temperatura), es configurable a “ ±1g” o “±5 g”, cuenta además con un procesador digital de señales (Digital Signal Procesor) y un microcontrolador interno que permite la conversión de la señal analógica a digital reportando la salida tanto de manera analógica como de manera digital; la salida en formato digital es de tipo serial bajo estándar RS-485, la configuración de este instrumento se realiza mediante un

software que proporciona el fabricante Instrument Configuration Utility (ICU) que establece comunicación con el sensor bajo un protocolo propietario llamado Serial Instrument Protocol (SIP).

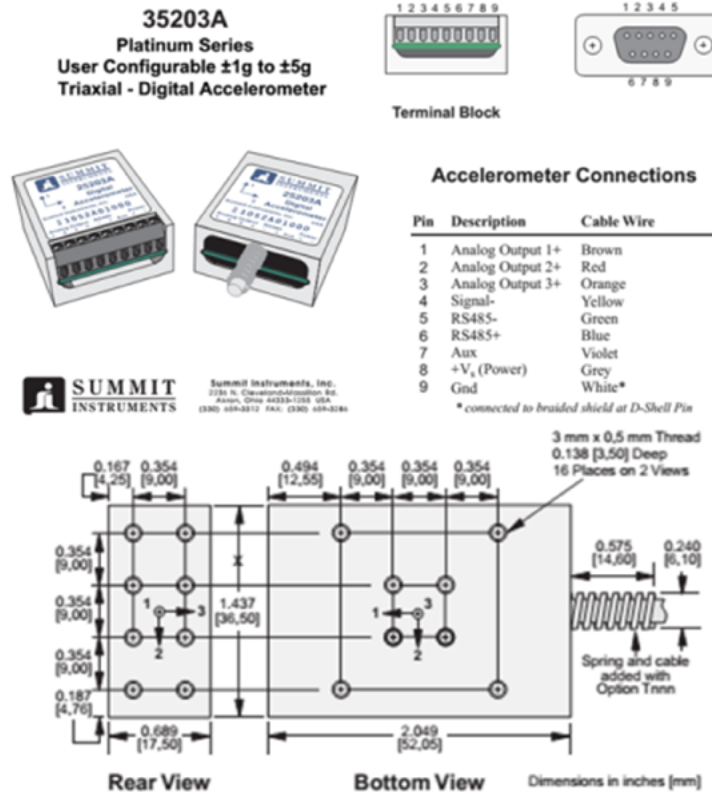


Figura 8. Acelerómetro SUMMIT 35203A.

Tabla 2. Especificaciones de acelerómetro SUMMIT 35203A.



	Mín	Típ	Máx	Unid
<b>Especificaciones de Acelerómetro</b>				
Máximo rango de escala completa	±5	±1		g
Deriva de Sensibilidad 25°C a Tmin ó Tmax		±5		%
Deriva de Cero G 25°C a Tmin o Tmax		±0,2		g
Alineamiento		±0,2		grados
Sensibilidad Transversa		0,25		%
No Linealidad		0,2		%FSR
Frecuencia de Corte Superior	1	1000	10000	Hz
Densidad de Ruido		0,255	0,325	$mg/\sqrt{Hz}$
<b>Especificaciones de Temperatura</b>				
Rango	-55		+125	°C
Resolución		0,25		°C
Precisión		±2	±3	°C
<b>Especificaciones del DSP</b>				
Tamaño de Palabra Interna			32	bits
Rata de Escaneo			42500	Hz
<b>Especificaciones de Salida Analógica</b>				
Oscilación de voltaje	0,25	-	4,75	Volts
Impedancia a Tierra	100	130	220	Ω
No Linealidad			0,15	%FSR
<b>Tamaño de palabra digital de salida</b>			16	bits
<b>Rata de Salida Bajo Estándar IRIG-106</b>			3	Mbit/seg
<b>Especificaciones de Fuente de Tensión</b>				
Límites de voltaje de entrada	-20	-	+60	Volts
Límites de operación	+8		+30	Volts
Entrada de corriente		50		mA
Factor de rechazo a Fuente de alimentación	80	120		dB
<b>Rango de Temperatura de Trabajo</b>			-40	+85
<b>Especificaciones de Choque</b>				
Alimentado	-500		+500	g
Sin alimentación	-1000		+1000	g
<b>Masa</b>			85	gramos

## 2.6 Configuración de Acelerómetro SUMMIT 35203A

El acelerómetro Summit 35203A se configura por medio del software ICU; este software está diseñado para configurar, activar, calibrar y monitorear algunos instrumentos digitales fabricados por la empresa SUMMIT INSTRUMENTS; al ejecutar este programa se muestra una ventana donde es posible seleccionar los parámetros de transmisión para establecer comunicación con el sensor (Figura 9) y así realizar la sucesivas configuraciones para su funcionamiento en modo de trabajo.

### 2.6.1 Ventana Principal de Configuración

En la ventana de entrada se tienen los siguientes parámetros de selección:

**COM Port:** Puerto serial de entrada.

**COM Baud:** Velocidad de comunicación que establecerá el programa con el sensor para realizar la configuración.

**Data directory:** directorio de programa.

**Default Instrument:** Selección entre el modelo de instrumento 35203A y 65210AS.

**Logo File:** Selección de logo personalizado.

**Readings per Measurements:** Lecturas por medida.

**Calibration Scrip File:** Scrip de calibración.

**Oversampling Warning Limit:** Límite de oversampling.



**Figura 9. Ventana principal de configuración About/Setup ICU.**

### 2.6.2 Bloque “Program Function”

En este bloque se pueden seleccionar uno de los modos de funcionamiento del programa que da acceso a otra ventana, la cual permite acceder al modo seleccionado. Los modos en este bloque son:

- On-line Configuration
- Create Application File(s)
- Activate and / or Load Application File(s)
- Calibration
- Receive Telemetry

## On-line Configuration

La ventana de On-line Configuration (Figura 10) presenta los parámetros configurables del sensor como lo son:

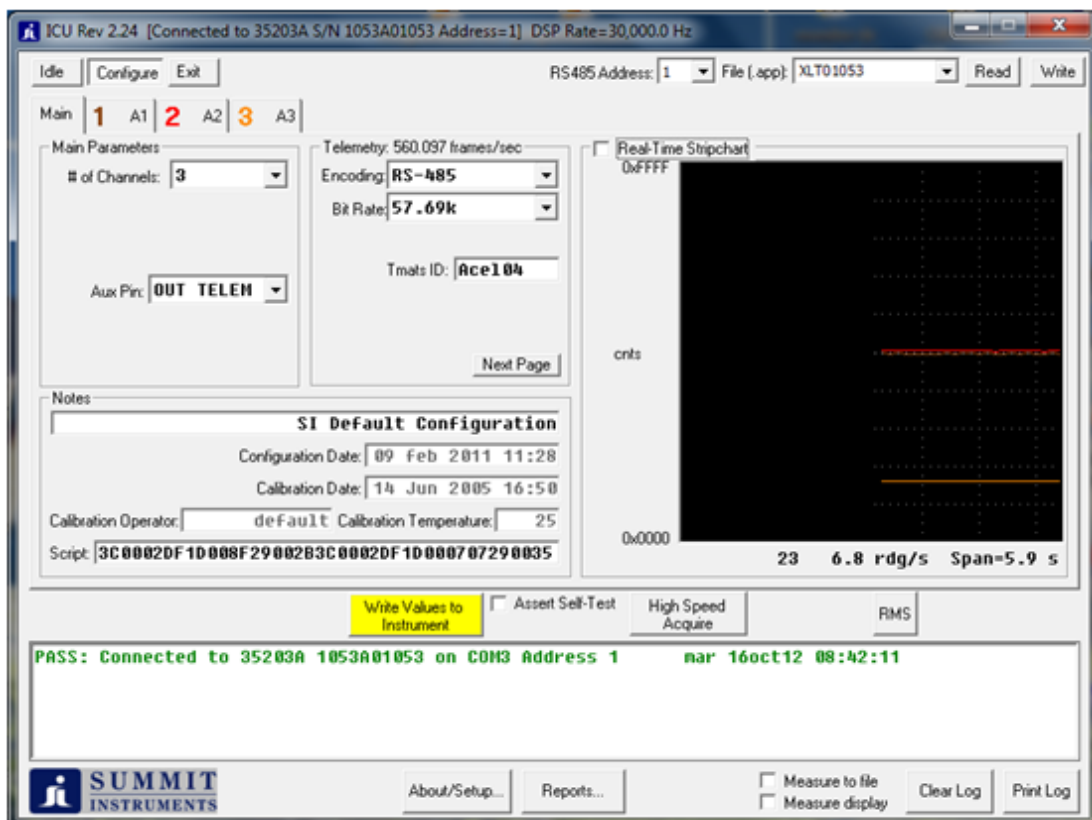
- Parámetros principales:
  - Dirección o Nombre del Dispositivo
  - Número de Canales a Utilizar
  - Función de Pin Auxiliar
- Parámetros de configuración de trama de datos y telemetría:
  - Estándar de comunicación (En el caso del modelo 35203A solo está disponible la opción RS-485)
  - Velocidad de transmisión serial del formato rs-485
  - Tamaño de byte de sincronismo
  - Tamaño de byte de señalización de trama
  - Opción agregar a la trama serial del sensor
  - Opción de agregar un bit de relleno (Pad bit) que señala el valor de la medida realizada en comparación a la primera medida.
  - Tipo de conmutación
  - opción para configurar una trama personalizada

“Bloque de Notas” donde se pueden registrar aspectos considerados importantes, también presenta casillas de información para las fechas de configuración, calibración, nombre del operador, temperatura y script; el código de script es un código único que corresponde con el total de configuraciones realizadas, pudiendo tenerse entonces un conjunto de configuraciones específicas asociadas a un script y así realizar ese conjunto de configuraciones con solo ese código específico.

Además se tienen pestañas correspondientes a cada canal (como se muestra en la figura tal). En cada pestaña se configura el elemento del acelerómetro que se quiere asociar al canal correspondiente y se pueden establecer los parámetros de:

- Frecuencia de Corte
- Rango de magnitud de aceleración

También se obtienen parámetros de ajuste para la conversión de los datos recibidos como lo son la ganancia y el offset.



**Figura 10. Ventana ICU On-Line Configuration.**

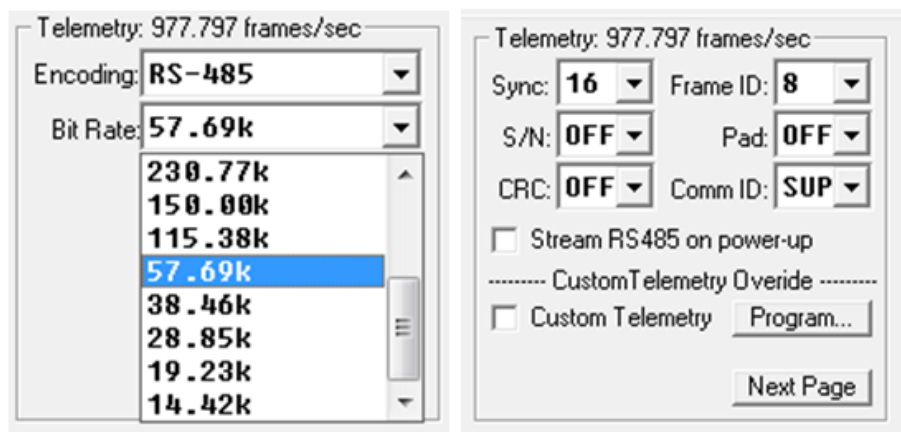


Figura 11. Opciones de Telemetría (On-Line Configuration).

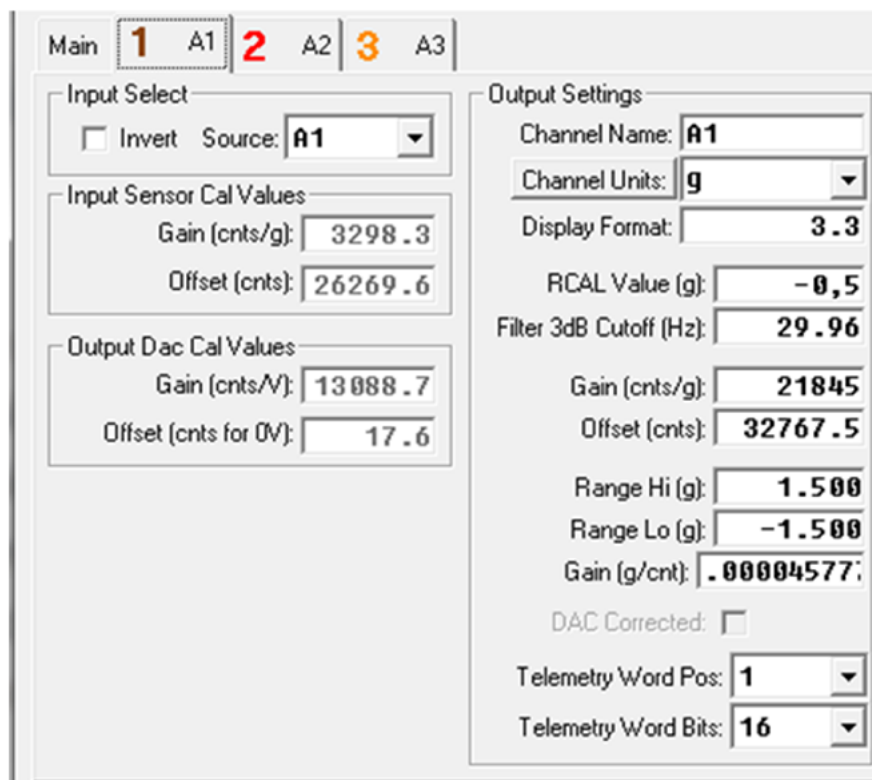


Figura 12. Parámetros de Canales (On-Line Configuration).

## **Bloque “High Speed Acquire Options”**

En este bloque se seleccionan parámetros para el modo de alta velocidad referentes al formato de archivos deseado para guardar los datos.

## **Bloque “Stripchart Graphics Method”**

En este bloque se seleccionan opciones de presentación para los datos obtenidos dentro del software ICU.

### **2.7 Protocolo SIP**

El protocolo SIP (Serial Instrument Protocol) es un protocolo de propósito general diseñado en torno a los requerimientos de los instrumentos y comunicación de sensores construidos por SUMMIT INSTRUMENT. El protocolo es del tipo comando-respuesta en forma serial, está implementado sobre el modelo 35203A bajo estándar RS-485. Cada comando se compone de tres partes:

#### **I. Byte de Información.**

- Bits del 0 al 2 Comando o Respuesta al Comando
- Bit 3 Señala si los datos son en bytes(1) o si los datos son en bloques de 16 bytes(0)
- Bit 4-7 Largo de Datos en (0 a 15 bytes) o bloques de 16 bytes

#### **II. 0 a 16 Bytes de Información.**

#### **III. Bytes de CRC.**

## 2.8 Principales Comandos Utilizados en Protocolo SIP

De esta manera la trama de comandos varía en tamaño para cada comando específico. Los principales comandos utilizados son los siguientes:

**LISTEN (0x18):** Comando utilizado para identificar un sensor dentro de un bus de comunicación obtiene como respuesta el comando **Acknowledge** en respuesta del reconocimiento de la llamada de comunicación.

**WRITE REGISTER (0x4C):** comando utilizado para indicar la operación de escritura sobre un registro dentro del micro o DSP del sensor.

**WRITE DATA@ADDRESS (0x1D):** comando utilizado para escribir dato en la dirección apuntada con el comando anterior.

**SYNCRONIZE (0xF8):** comando utilizado para preparar el sensor para sincronización de algunos parámetros necesarios dentro de la comunicación establecida o por establecerse.

**RESTART (0x0E):** comando utilizado para inicializar el sensor.

**ACKNOWLEDGE (0x0D):** comando de respuesta de parte del sensor ante petición de comandos de entrada.

## 2.9 Formas de Comunicación con Sensor por Medios Externos

Es posible adquirir las medidas realizadas directamente del sensor sin utilizar el software del fabricante por medio de software personalizado o propio, las únicas restricciones son que la configuración del sensor sea realizada previamente usando el programa ICU.



El fabricante por medio del material de referencia provee tres formas para establecer comunicación y adquirir datos del sensor sin la utilización del software del fabricante, (estas metodologías están descritas en las notas técnicas del fabricante #411, #412, #418).

### **2.9.1 Adquisición de medidas del Sensor Digital (Acquiring Digital Sensor Measurements)**

En esta metodología se describen los pasos a seguir para obtener las medidas de los canales deseados en un instante cualquiera. La metodología requiere enviar un bloque de comandos que indica al sensor que será inicializado (SETUP), luego un bloque de comandos que permite obtener los datos requeridos del sensor (SCAN). Pasos requeridos para adquirir datos:

#### **SETUP**

1. Enviar el comando `SYNCHRONIZE` para sincronizar con el sensor deseado dentro del bus de comunicación
2. Esperar por lo menos 128 milisegundos para asegurar que el sensor haya procesado el comando
3. Enviar el comando `LISTEN` para habilitar el sensor al proceso de comandos
4. Esperar la respuesta del sensor.

#### **SCAN**

1. Enviar comando `WRITE REGISTER` para escribir dato de dirección en el registro de dirección.

2. Esperar a que el sensor reconozca el proceso del comando `WRITE REGISTER` mediante respuesta `ACKNOWLEDGE`.
3. Utilizar el comando `WRITE DATA @ ADDRESS` para escribir cero y disparar el ciclo de mediciones del sensor.
4. Esperar a que el sensor reconozca el proceso del comando `WRITE DATA @ ADDRESS` mediante respuesta `ACKNOWLEDGE`.
5. Utilizar el comando `WRITE REGISTER` para escribir el dato de dirección en el registro de dato de dirección y así reiniciar la dirección.
6. Esperar a que el sensor reconozca el proceso del comando `WRITE REGISTER` mediante respuesta `ACKNOWLEDGE`.
7. Enviar comando `WRITE DATA @ ADDRESS` para leer los datos del sensor.
8. Esperar la respuesta del sensor con las medidas requeridas.
9. Convertir las mediciones a las unidades deseadas por la aplicación de los valores de correcciones de ganancia y offset dadas en ICU para tal fin.

### **2.9.2 Adquisición de Medidas Usando Scripts**

Las medidas realizadas por el instrumento pueden ser adquiridas del dispositivo usando un *Script* de configuración guardado en memoria. Para utilizar esta función se siguen los siguientes pasos:

1. Enviar comando `LISTEN`.
2. Esperar a que el sensor procese el comando `LISTEN`.
3. Enviar comando `JUMP` con la dirección del script en memoria para ejecutar el *Script*.

4. Esperar la respuesta del sensor con los datos por medio del comando `WRITE DATA @ ADDRESS`.
5. Convertir las medidas adquiridas a las unidades deseadas mediante los parámetros de corrección de *ganancia* y *offset* arrojados por el programa ICU.
6. Repetir los pasos 3, 4 y 5 cada vez para obtener la lectura requerida.

### **2.9.3 Adquisición de Datos en Modo High Speed**

Este modo permite la adquisición de datos al ejecutar una metodología en la inicialización del sensor, que una vez realizada activa el envío de una trama de mediciones de manera periódica que puede ser detenida al enviar un comando de fin de envío de datos `BREAK`, la inicialización de este modo se realiza mediante los siguientes pasos:

#### **SETUP (Requerido solo al inicio)**

1. Enviar el comando `SYNCHRONIZE` para sincronizar con el sensor deseado dentro del bus de comunicación
2. Esperar por lo menos 128 milisegundos para asegurar que el sensor haya procesado el comando
3. Enviar el comando `LISTEN` para habilitar el sensor al proceso de comandos
4. Esperar la respuesta del sensor.

### **RESTART** (Rutina de Reinicio de Sensor)

1. Enviar comando `RESTART`.
2. Esperar 1.2 segundos para que el proceso de reinicio ocurra en el instrumento.
3. Enviar el comando `SYNCHRONIZE`.
4. Esperar por lo menos 128 milisegundos para asegurar que el sensor haya procesado el comando.
5. Enviar el comando `LISTEN` para habilitar el sensor al proceso de comandos.
6. Esperar a que el sensor reconozca el proceso del comando `LISTEN` mediante respuesta `ACKNOWLEDGE`.

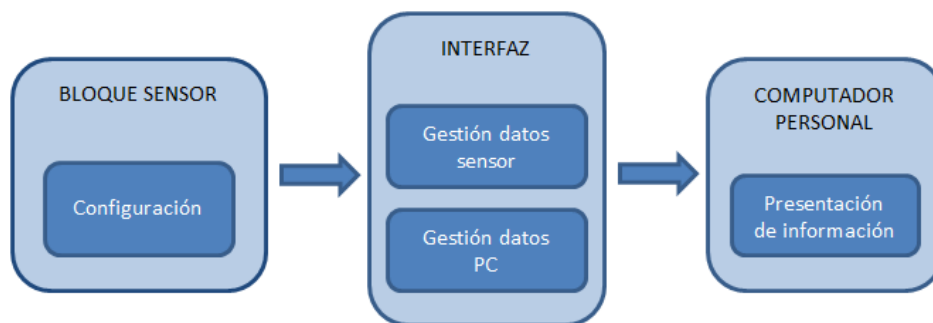
### **HIGH SPEED** (Inicialización en Modo de Alta Velocidad)

1. Enviar comando `WRITE REGISTER` a la dirección de habilitación del modo *High Speed* (**0x0700**)
2. Esperar por el proceso de reconocimiento del sensor (Comando `ACKNOWLEDGE`) del comando `WRITE REGISTER`.
3. Utilizar el comando `WRITE DATA @ ADDRESS` para escribir cero (0) en la dirección de habilitación del modo *High Speed*. Seguidamente el sensor comenzara a transmitir la data en modo *High Speed*
4. Cambiar la velocidad de comunicación serial (Baud Rate) entre el dispositivo receptor y el sensor a la velocidad de trabajo previamente establecida en la configuración
5. Leer la información enviada por el sensor

6. Para detener el envío de data desde el sensor se debe enviar un comando Break al sensor
7. Desechar los datos recibidos después de enviar comando BREAK.
8. Convertir las medidas adquiridas desde el sensor por medio de los parámetros de corrección *ganancia* y *offset*.

## CAPÍTULO III

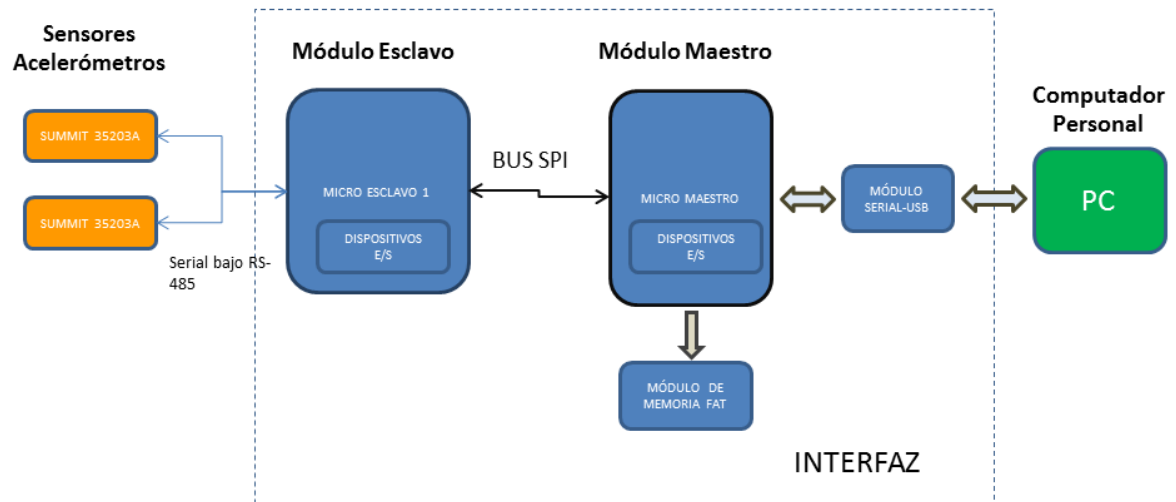
Para el desarrollo de la solución se realizó un proceso de investigación para conocer las tecnologías asociadas a los diferentes dispositivos del sistema para lograr funcionamiento, se seleccionaron los métodos más adecuados según las funciones deseadas y se propuso un sistema solución con funciones específicas. En la Figura 13 se presenta el esquema general del sistema solución propuesto. Se desarrolló un prototipo interfaz y a continuación se presentan la definición de hardware y software además de la descripción de hardware y software bajo la cual se desarrolló el prototipo.



**Figura 13. Sistema solución.**

El sistema realiza la adquisición de datos de un conjunto de Sensores acelerómetros SUMMIT 35203A bajo protocolo RS-485, organiza los datos recibidos, transmite los datos a un computador personal donde serán mostrados por el

sistema y guarda los datos. La Figura 14 muestra un esquema de los módulos que componen el sistema.



**Figura 14. Esquema de módulos de sistema solución.**

### 3.1 Definición del Hardware

Para la ejecución de funciones a realizar el sistema deberá contar con un módulo esclavo que reciba los datos de los sensores acelerómetros SUMMIT 35203A y organice los datos recibidos, un módulo maestro que se encargue de tomar los datos organizados del módulo esclavo, enviar los datos al computador personal y al módulo FAT para que sean guardados, un programa interfaz en el computador personal para mostrar los datos recibidos y un módulo fuente de alimentación para proveer de alimentación de energía. Tanto el módulo esclavo como el maestro contarán con un microcontrolador que se encargara de realizar las funciones de hardware requeridas.

### 3.1.1 Módulo Esclavo

El módulo esclavo deberá cumplir los siguientes requerimientos para ejecutar las funciones deseadas:

- Debe tener un periférico UART con la interfaz necesaria para recibir datos Provenientes de Sensores acelerómetros SUMMIT 35203A bajo protocolo RS-485.
- Debe tener un periférico SPI para transmitir los datos recibidos desde el conjunto de acelerómetros al módulo maestro.

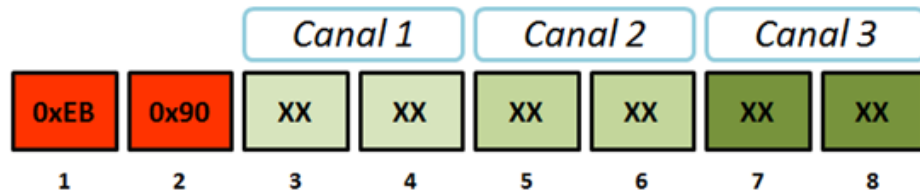
Las pruebas de funcionamiento realizadas con el sensor permitieron escoger el método para la comunicación con éste, el cual fue la transmisión de trama continua de datos al encender la alimentación de energía del sensor. Esto debido a que se omiten las configuraciones iniciales de envío y recepción de comandos para la inicialización, descartando así complicaciones en la dinámica de inicialización. Esta forma es una manera simplificada del modo “High Speed”, mientras que los otros métodos (SCAN y SCRIPT) fueron descartados por consumir mucho tiempo en el envío y recepción de comandos para una sola muestra.

La velocidad de transmisión necesaria se definió en base a que el rango de frecuencia de las señales a estudiar está entre 60Hz y 100Hz.

En vista de esto y considerando el teorema de Nyquist que establece que una frecuencia de muestreo debe ser mayor o igual al doble de la frecuencia de la señal bajo estudio, además de la experiencia práctica de que para la reconstrucción de la señal debe tenerse una frecuencia de muestreo que debe estar alrededor de 8 o más veces la frecuencia de la señal bajo estudio, así pues se estableció que en este caso fuera de 10 veces.



Por tanto, el tiempo de una muestra viene dado por la el tamaño de una trama de datos, en este caso la trama de datos está compuesta por 2 bytes de señalización (0xEB, 0x90) y 2 bytes por cada canal de aceleración asociado a los ejes X-Y-Z, así una muestra viene dada por 8 bytes.



**Figura 15. Trama muestra enviada por sensor acelerómetro**

De manera tal que el tiempo de una muestra es:

$$tmuestra = tbyte * \text{número de bytes en muestra}$$

Donde

$$tbyte = \frac{1}{\text{Baudrate}} * \text{número de bits}$$

Así

$$tmuestra = tbyte * \text{número de bytes en muestra}$$

Sabiendo que

$$fmuestra = \frac{1}{tmuestra}$$

$$f_{muestra} = \frac{Baudrate}{\text{número de bits} * \text{número de bytes en muestra}}$$

número de bits = 10(1 bit de inicio, 8bits de dato, 1 bit de parada)

número de bytes en muestra = 8(2 bytes de señalización de trama, 6bytes de datos)

$$f_{muestra} = \frac{Baudrate}{10 * 8}$$

$$f_{muestra} \geq 10 * f_{señal}$$

$$\frac{Baudrate}{10 * 8} \geq 10 * f_{señal}$$

$$f_{señal} \leq \frac{Baudrate}{10 * 8 * 10}$$

$$f_{señal} \leq \frac{Baudrate}{800}$$

$$Baudrate \geq 800 * f_{señal}$$

Sabiendo que,  $60[\text{Hz}] \leq f_{señal} \leq 100[\text{Hz}]$

$$Baudrate \geq 800[\text{Baudios/Hz}] * 60[\text{Hz}] \rightarrow Baudrate \geq 48000[\text{Baudios}]$$

$$Baudrate \geq 800[\text{Baudios/Hz}] * 100[\text{Hz}] \rightarrow Baudrate \geq 80000[\text{Baudios}]$$

Ahora bien, las velocidades que permite configurar el acelerómetro son 57690 Baudios, 115380 Baudios, y 150000 Baudios. Por lo que la velocidad escogida para transmisión bajo protocolo RS-485 debido a que cumple ambas condiciones es la de 115380 Baudios.

### 3.1.2 Modulo Maestro

El modulo maestro deberá:

- Contener periférico SPI para recibir los datos adquiridos por el modulo esclavo.
- Contener un periférico UART para transmitir los datos al módulo FAT a fin de ser guardados.
- Contener un periférico UART para transmitir los datos al computador personal.

Los datos adquiridos por el módulo esclavo llegan a una velocidad de 115380 *Baudios* como se presentó anteriormente, de donde se tiene lo siguiente.

$$t_{Byte} = \frac{1}{\text{velocidad en Baud Rate}} * n \text{ de bits} = \frac{1}{115380} * 10\text{bits} = \frac{10}{115380} \text{seg}$$

$$t_{Byte} = \frac{n \text{ de bits}}{\text{velocidad en Baud Rate}} = \frac{1}{115380} * 10\text{bits} = \frac{10}{115380} \text{seg/byte}$$

$$N_{Bytes/seg} = \frac{1}{t_{Byte}} = \frac{\text{vel. Baud Rate}}{n \text{ de bits}} = \frac{115380}{10} \text{byte/seg} = 11538 \text{ Bytes/seg}$$

Así que es necesario tomar en cuenta esta velocidad de transferencia por cada sensor involucrado al momento de establecer las velocidades de transmisión.

### **3.1.3 Modulo FAT**

Deberá contener un microcontrolador que gestione el almacenamiento en una memoria SD y debe tener conector SD.

### **3.1.4 Modulo Fuente de Alimentación**

El modulo fuente de alimentación deberá alimentar con corriente directa el bloque de sensores acelerómetros SUMMIT 35203A, así como los módulos del sistema. La tensión de entrada a este módulo de corriente directa debe ser igual o mayor a 9 volts.

## **3.2 Definición del Software**

El software del sistema solución se compone del firmware de los microcontroladores que lo componen, así como la programación de la interfaz encargada de mostrar los datos obtenidos.

### **3.2.1 Módulo Esclavo**

El software del módulo esclavo será el firmware del microcontrolador el cual deberá realizar las siguientes funciones:

- Configurar los registros y periféricos del microcontrolador
- Gestionar la adquisición de datos seriales de los acelerómetros SUMMIT 35203A a través de periférico UART y guardarlos en un arreglo de datos
- Indicar mediante función de señalización canales en funcionamiento
- Ensamblar trama con datos recibidos de los sensores implicados para ser requeridos por módulo maestro y guardar en un arreglo de memoria.

- Gestionar requerimiento de datos en trama previamente elaborada por petición de modulo maestro a través de periférico SPI del microcontrolador.
- Dar señalización al módulo maestro para sincronización de tiempos de funcionamiento.

### **3.2.2 Módulo Maestro**

El software del módulo maestro será el firmware del microcontrolador el cual deberá realizar las siguientes funciones:

- Configurar los registros y periféricos del microcontrolador
- Requerir datos al módulo esclavo por medio de periférico SPI
- Transmitir datos obtenidos del módulo esclavo hacia el computador personal y al módulo FAT usando periférico UART

### **3.2.3 Interfaz PC**

La interfaz de software realizada en el PC deberá realizar:

- Recepción de datos provenientes del módulo maestro.
- Organizar la información en un arreglo de datos.
- Desensamblar la trama de datos previamente elaborada en el módulo esclavo.
- Discriminar según canales de entrada los datos recibidos.
- Presentar los datos al usuario en un entorno gráfico.

### **3.3 Descripción del Hardware**

El sistema solución compuesto por los módulos esclavo y maestro, son sistemas electrónicos desarrollados en base a un microcontrolador encargado de realizar las tareas necesarias de comunicación debido a sus características, el microcontrolador elegido para este desarrollo es un microcontrolador de la casa microchip, familia PIC 24F, específicamente el modelo pic24fj64ga002, este microcontrolador cuenta con dos (2) periféricos de hardware UART, también cuenta con periférico SPI, su capacidad de cómputo es de 16 MIPS, lo que lo hace uno de los más rápidos existentes en el mercado en relación a su costo al momento de su compra, tiene amplia variedad de vectores de interrupción adecuadas para las funciones implementadas, también cuenta con amplio soporte de manejo en la escuela de ingeniería eléctrica, por lo que se decidió seleccionar este modelo.

El módulo esclavo cuenta también con un circuito integrado MAX-485 que se encarga de convertir las señales del protocolo RS-485 del conjunto de sensores directamente a señales procesadas por el microcontrolador.

El módulo de alimentación está constituido por reguladores de tensión 78M33 y 7805 para obtener las tensiones adecuadas para cada dispositivo, en este caso son necesarias tensiones de 3,3volts para los microcontroladores, 5volts para los MAX-485 y tensión mayor a 8 volts para alimentar el conjunto de sensores, el módulo FAT se implementó con un módulo comercial denominado *OpenLog* de la empresa *SparkFun* Electronics. Fue necesario el uso de un convertidor serial-USB para la comunicación con el computador personal y el módulo maestro.

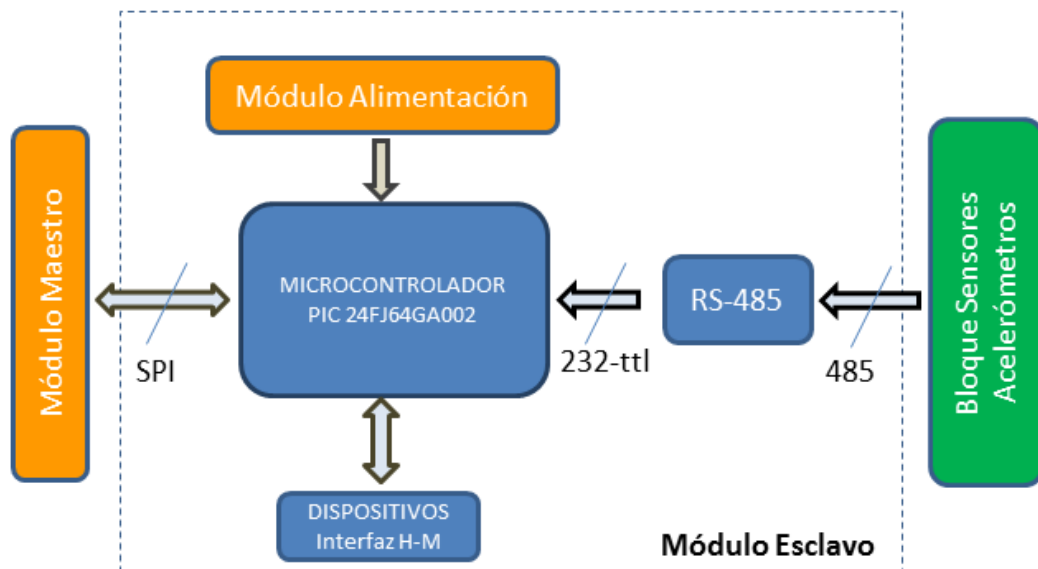
#### **3.3.1 Módulo Esclavo**

El módulo esclavo fue diseñado teniendo en cuenta que la función principal es una transmisión serial bajo protocolo RS-485 por lo que el sistema se compone de

dos circuitos integrados MAX-485 conectados a dos canales seriales del periférico UART permitidos por el microcontrolador pic24fj64ga002, los datos provenientes del conjunto de sensores son procesados por el circuito integrado MAX-485.

La velocidad de transmisión de cada acelerómetro SUMMIT 35203A fue fijada a 115.380 baudios lo que representa un flujo de 11.538 Bytes por segundo, teniendo en cuenta que el microcontrolador encargado maneja dos sensores acelerómetros simultáneamente, este flujo se duplica, por lo que la velocidad de datos de entrada es de 23.076 Bytes por segundo, así, la velocidad de transmisión hacia el módulo maestro debe tener un flujo de Bytes superior a esta velocidad.

La transmisión hacia el módulo maestro se realiza por el periférico SPI, se configuró una velocidad de transmisión SPI de 1Mhz tomando en cuenta que esta velocidad representa un flujo de datos de 125.000 Bytes por segundo y es 5,42 veces mayor que el flujo de datos de trabajo, además de que físicamente no presenta problemas de perdidas ni distorsión de datos transmitidos.



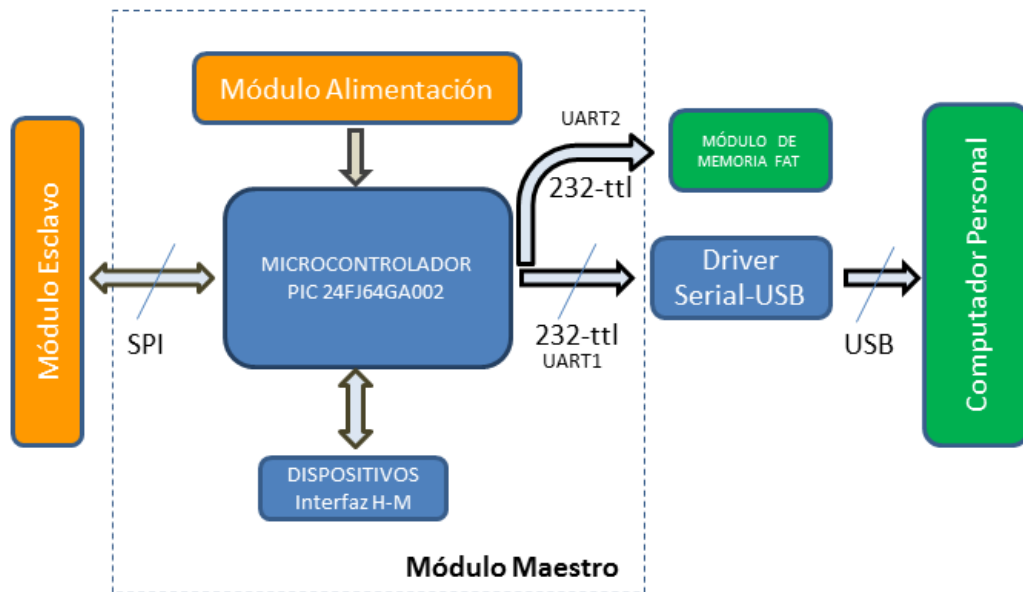
**Figura 16. Diagrama de módulo esclavo.**

### 3.3.2 Módulo Maestro

El diseño del módulo maestro se realizó teniendo en cuenta transmisión desde el módulo esclavo, y hacia los módulos FAT y computador personal, para esto se debe tener en cuenta que la velocidad de flujo de datos de entrada del esclavo es de 23076 bytes por segundo, por lo que la velocidad de flujo de hacia el maestro debe ser superior.

En Al igual que con el módulo esclavo se utilizó un microcontrolador pic24fj64ga002 de la casa Microchip, se configuró el periférico SPI para una velocidad de 1 MHz equivalente a una velocidad de flujo de 125.000 bytes por segundo para la transmisión de datos entre los módulos maestro-esclavo tal que se garantiza una transmisión sin pérdida de datos. Las velocidades configuradas en los dos periféricos UART son de 250.000baudios(25.000 bytes por segundo) para la transmisión al computador personal y 115.200 Baudios (11.520 bytes por segundo) al módulo FAT debido a que es la velocidad máxima soportada por el modulo FAT. Se utiliza un módulo convertidor serial a USB, que es reconocido por el PC como un puerto virtual de comunicación serial RS-232.

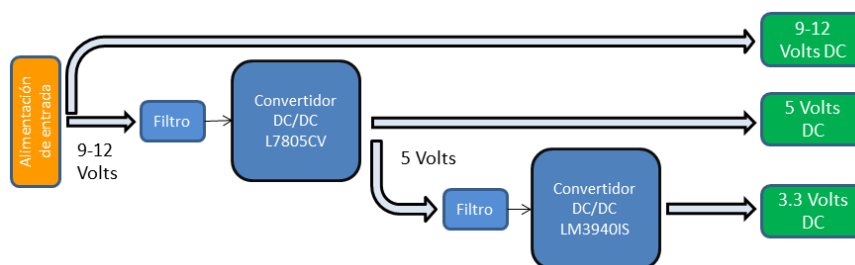




**Figura 17. Diagrama de módulo Maestro.**

### 3.3.3 Módulo de Alimentación de Tensión

El módulo de alimentación se diseñó con dos reguladores de tensión, un L7805 para regular tensión de entrada a 5 volts y un LM3940 para regular tensión a 3.3 volts la tensión de entrada de alimentación de este módulo es de 9 volts a 12 volts de manera que el modulo provee de energía a las tensiones indicadas para los microcontroladores, circuitos integrados MAX-485 y los sensores acelerómetros.



**Figura 18. Diagrama de módulo de alimentación.**

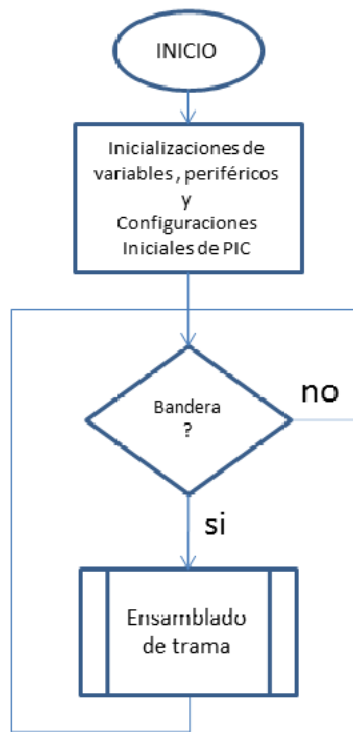
### **3.4 Descripción de Software**

El software se constituye del firmware mediante el cual fueron programados los Microcontroladores, así las rutinas de funcionamiento lo constituyen

#### **3.4.1 Módulo Esclavo**

Para el funcionamiento del módulo esclavo fueron diseñadas rutinas de programación en lenguaje C utilizando el ambiente de desarrollo de Microchip MPLABX IDE v1.90, estas rutinas configuran, inicializan y controlan los diferentes procesos, registros y periféricos en el microcontrolador. En el módulo esclavo se tienen las siguientes rutinas que controlan el funcionamiento del módulo.

La rutina principal lleva a cabo la configuración de registros de: inicialización de periféricos, asignación de pines, inicialización de variables además de controlar el ensamblado periódico de una supertrama constituida por los datos tomados de los arreglos de recepción de los periféricos UART del microcontrolador. La Figura 19 muestra el diagrama de flujo de esta rutina principal.



**Figura 19. Diagrama de flujo de Rutina Principal Módulo Esclavo**

El ensamblado de la supertrama se realiza tomando datos de los arreglos Buff\_Rx1 y Buff\_Rx2 para ser guardados en el arreglo Buff\_SPI. se estableció un tiempo en el cual se cambia el valor de una variable constantemente verificada en el hilo principal del programa que permite realizar la subrutina de ensamblado de los datos previamente guardados en Buff\_Rx1 y Buff\_Rx2, cada rutina de interrupción de recepción serial se encarga de guardar los datos provenientes de los sensores acelerómetros, al llegar el primer dato activa una variable dentro del microcontrolador del módulo esclavo y levanta una señal de salida hacia el módulo maestro dando así una sincronización y una señal de inicio en las rutinas internas del esclavo.

Los arreglos Buff\_Rx1 y Buff\_Rx2 dentro del módulo esclavo fueron fijados con un tamaño de 512 Bytes, calculando el tiempo de cada byte a una velocidad de

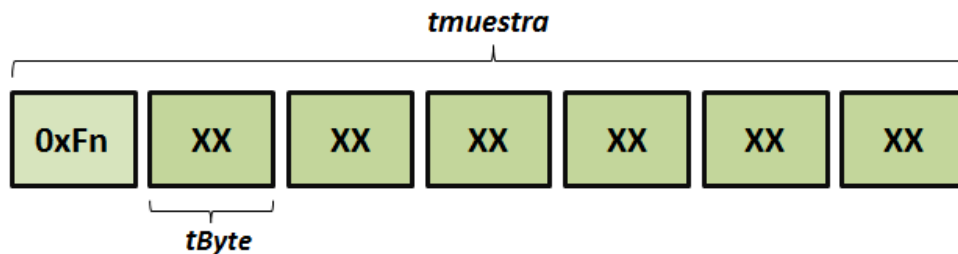
transmisión de 115200 baudios que es lo que permite configurar el microcontrolador se tiene que el tiempo de llenado total de los arreglos es de aproximadamente 44.4mseg y una trama de muestra un tiempo de 69.44mseg como se demuestra a continuación.

$$t_{bit} = \frac{1}{115200[\text{baudios}]} \cong 8.68\mu\text{seg}$$

$$t_{Byte} = 10\text{bits} * t_{bit} \cong 86.8\mu\text{seg}$$

$$t_{Buff\_RxN} = 512 \text{ Bytes} * t_{Byte} \cong 44.4\text{mseg}$$

$$t_{muestra} = 8\text{bytes} * t_{byte} \cong 69.44\text{mseg}$$



**Figura 20. Trama Muestra con cambio de cabecera de trama para señalización de canal de adquisición.**

Los datos guardados en estos dos arreglos necesitan ser traspasados de manera ordenada al arreglo Buff\_SPI desde donde son requeridos por el microcontrolador maestro a través de las rutinas de interrupción del periférico SPI.

En las rutinas de interrupción de recepción serial se discriminan los dos bytes de señalización de cada muestra (0xEB, 0x90) propios del protocolo propietario de los sensores acelerómetros SUMMIT 35203A y se intercambian por un solo Byte

que permite conocer de cual periférico UART proviene la muestra, así para el UART 1 se intercambia por 0xF1 y para el UART 2 por 0xF2.

Se fijó que para un tiempo igual al 60% del tiempo total de llenado de los arreglos Buff\_Rx1 y Buff\_Rx2 tomado desde la activación de la bandera dada por el primer dato recibido en cualquiera de los dos UARTS de recepción serial se active una señal dada por el cambio de valor en una variable que permite correr la subrutina que se encarga de ensamblar la supertrama y que será guardada en el Buff\_SPI, este tiempo corresponde a

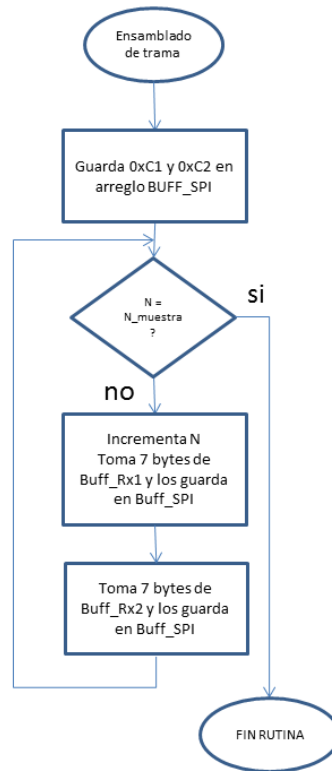
$$t_{Buff_{Rx}@60\%} = 44.4mseg * 60\% = 26.64mseg$$

Así cada 26.64 mseg son tomados los datos previamente guardados en cada arreglo Buff\_Rxn de 7 bytes en 7 bytes que es el nuevo tamaño de la muestra luego de la discriminación realizada por la rutina de interrupción serial encargada de guardar los valores recibidos. La cantidad de muestras requeridas es de 44 muestras.

Para identificar esta nueva trama de datos a enviar al maestro en forma SPI se colocan 2 bytes de señalización que son 0xC1 y 0xC2, así la trama de datos en el arreglo Buff\_SPI se ilustra en la Figura 21. El diagrama de flujo de la subrutina encargada de ensamblar la supertrama se muestra en la Figura 22.



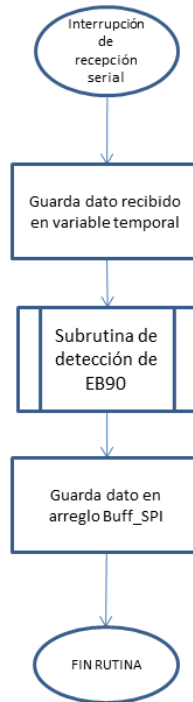
**Figura 21. Trama de datos en arreglo SPI.**



**Figura 22. Subrutina de ensamblado de supertrama SPI.**

La rutina encargada de la recepción de datos por medio del periférico UART es la interrupción por recepción serial, esta rutina en ambos UART es activada al recibir un byte, de manera que al activarse la interrupción la ejecución se encarga de guardar el dato recibido en un arreglo de datos que fue denominado Buff\_Rx1 para el UART1 y BuffRx2 para el UART 2, al recibir el byte se realiza una verificación para seleccionar los bytes que constituyen la cabecera de una trama los cuales son 0xEB y 0x90, estos bytes son sustituidos por un solo byte 0xF1 o 0xF2 dependiendo el UART asociado y así es guardado el byte sustituido o recibido en el arreglo pertinente, la Figura 23 muestra el diagrama de funcionamiento de la rutina. Se incluye en las rutinas de recepción serial una variable que será activada y una señal en un pin

externo que permite conocer el inicio de recepción de datos y así sincronizar el funcionamiento en la interfaz.

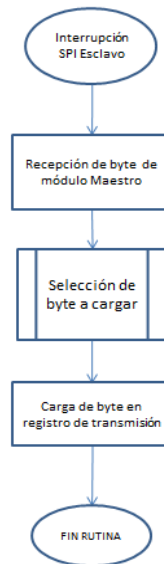


**Figura 23. Rutina de recepción serial.**

Se estableció una rutina de interrupción timer que inicia cada 26.64mseg correspondiente al 60% del tiempo de llenado total de los arreglos de datos, esta rutina de interrupción tiene la finalidad de actualizar el valor de una variable llamada Bandera que está siendo constantemente verificada en el programa principal y que permite iniciar la subrutina de ensamblado de la supertrama guardada en el arreglo Buff\_SPI. Al recibir por primera vez un dato serial en alguno de los UART es iniciado el contador del timer, de esta manera se logra la sincronización de las rutinas en proceso dentro del microcontrolador del módulo esclavo.

La forma de transmisión entre módulos esclavo y maestro es mediante el periférico SPI, de manera que el software asociado a esta función es la rutina de

interrupción SPI. Esta rutina se encarga de recibir un byte desde el módulo maestro y colocar en el registro de transmisión un byte tomado del arreglo Buff\_SPI. Fue programada una subrutina que permite seleccionar un byte del arreglo Buff\_SPI o un byte fijado como nulo en caso de no existir datos en el arreglo y así cargar al registro de transmisión del periférico SPI el dato que será enviado.



**Figura 24. Rutina de interrupción SPI microcontrolador esclavo.**

### 3.4.2 Módulo Maestro

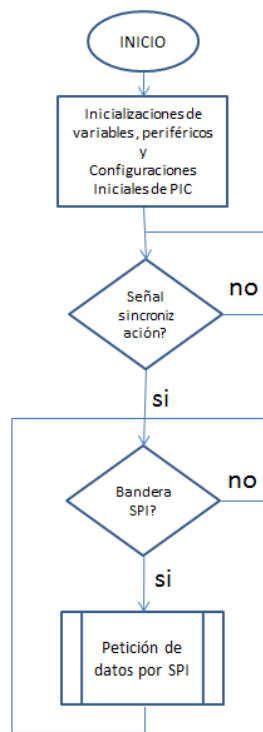
Al igual que en el módulo esclavo para la programación del microcontrolador del módulo maestro fueron diseñadas rutinas de programación en lenguaje C utilizando el ambiente de desarrollo de Microchip MPLABX IDE v1.90 para configurar, inicializar y controlar los diferentes registros y periféricos en el microcontrolador.

La rutina principal en el microcontrolador del módulo maestro se encarga de configurar los registros del microcontrolador y así inicializar los periféricos, asignar



las funciones a los pines, inicializar las variables y esperar la señalización desde el módulo esclavo para iniciar la subrutina de requerimiento de datos por medio de periférico SPI, esta subrutina es habilitada por actualización de una variable a cargo de una interrupción de timer configurada para activar cada 26.64mseg, la variable llamada Bandera\_SPI es constantemente verificada en el programa principal a partir de la señalización del módulo esclavo.

Se espera un retardo de 13.32mseg equivalente a la mitad del periodo para requerir el total de una supertrama ya ensamblada en el microcontrolador esclavo, los bytes recibidos son guardados en un arreglo de memoria llamado Buff\_SPI en la memoria del microcontrolador maestro que tiene un tamaño fijado de 4096 bytes.



**Figura 25. Diagrama de flujo de programa principal de módulo maestro.**

Para transmitir los datos almacenados en el arreglo de memoria Buff\_SPI del microcontrolador esclavo se configuró una interrupción de timer donde son enviados los datos del arreglo por medio de los UART dispuestos para la transmisión hacia el computador personal y el modulo FAT, el periodo de esta interrupción fue fijado en base a que el flujo de datos de entrada viene limitado por el flujo de datos recibidos desde los acelerómetros en el módulo esclavo, con una velocidad de 23.076 bytes por segundo. Una transmisión de datos a 250.000 baudios equivale a 25.000 bytes por segundo y el tiempo de transmisión de un byte a 250.000 baudios es de

$$t_{bit@250k} = \frac{1}{250000[baudios]} \cong 4 \mu seg$$

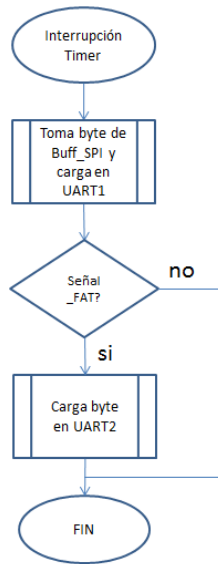
$$t_{Byte@250k} = 10bits * t_{bit} \cong 40 \mu seg$$

El modulo FAT limita a una transmisión a velocidad máxima de 115.200 baudios

$$t_{bit@115.2k} = \frac{1}{115200[baudios]} \cong 8.68 \mu seg$$

$$t_{Byte@115.2k} = 10bits * t_{bit} \cong 86.8 \mu seg$$

De esta manera se fijó un tiempo de interrupción de 45  $\mu seg$  tal que se permite el envío al computador personal de bytes a 250.000 baudios y se elaboró una lógica de programación para que cada dos interrupciones correspondientes a 90  $\mu seg$  se active una Señal\_FAT que permite enviar al módulo FAT los datos a una velocidad de 115.200 baudios. El diagrama de la Figura 26 ilustra la programación de esta interrupción.

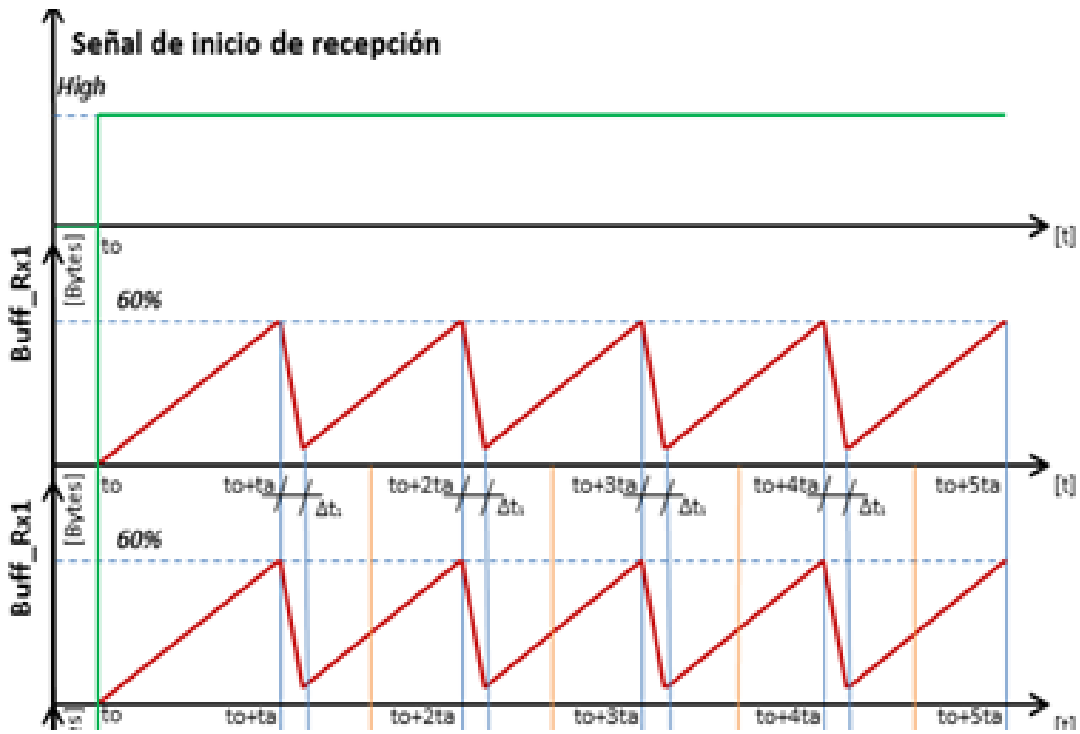


**Figura 26. Interrupción de timer encargada de realizar el envío de datos a PC y módulo FAT.**

### 3.4.3 Sincronización de Interfaz

Para que el funcionamiento de la interfaz sea adecuado fue necesario sincronizar los procesos de cada una de las partes.

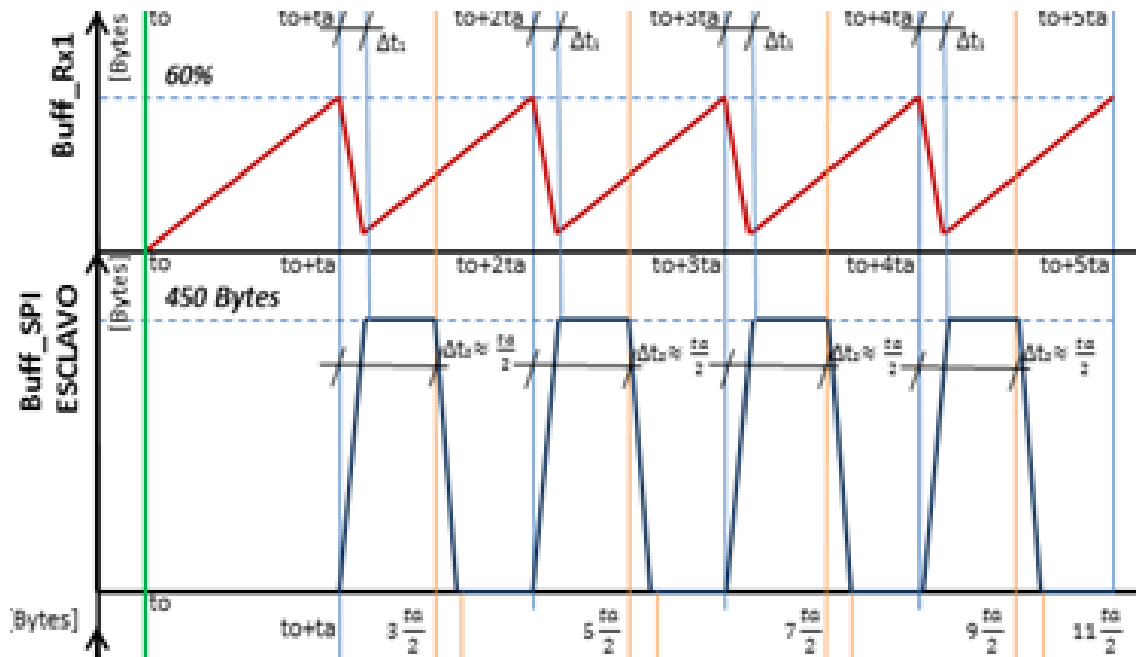
En principio se estableció un tiempo periódico “ $t_a$ ” (fijado a 60% de llenado del total de datos de los arreglos de recepción serial, así  $t_a = 26,65$  mseg, sabiendo de antemano el tiempo total de llenado de los arreglos Buff\_Rx1 y Buff\_Rx2)



**Figura 27. Tiempo  $t_a$  de vaciado de arreglos de datos seriales.**

El inicio del tiempo  $t_a$  se da con la llegada del primer dato a cualquiera de los dos arreglos Buff\_Rx1 o Buff\_Rx2, a lo que se llamó señal de sincronización, la cual se verifica en las interrupciones de UART de ambos canales del microcontrolador.

Luego de activarse la señal de sincronización al recibir por primera vez un byte, en un tiempo  $t_a$  los datos son tomados de los arreglos Buff\_Rx1 y Buff\_Rx2 y guardarlos en un tercer arreglo Buff\_SPI (en un tiempo  $\Delta t_1$ ).

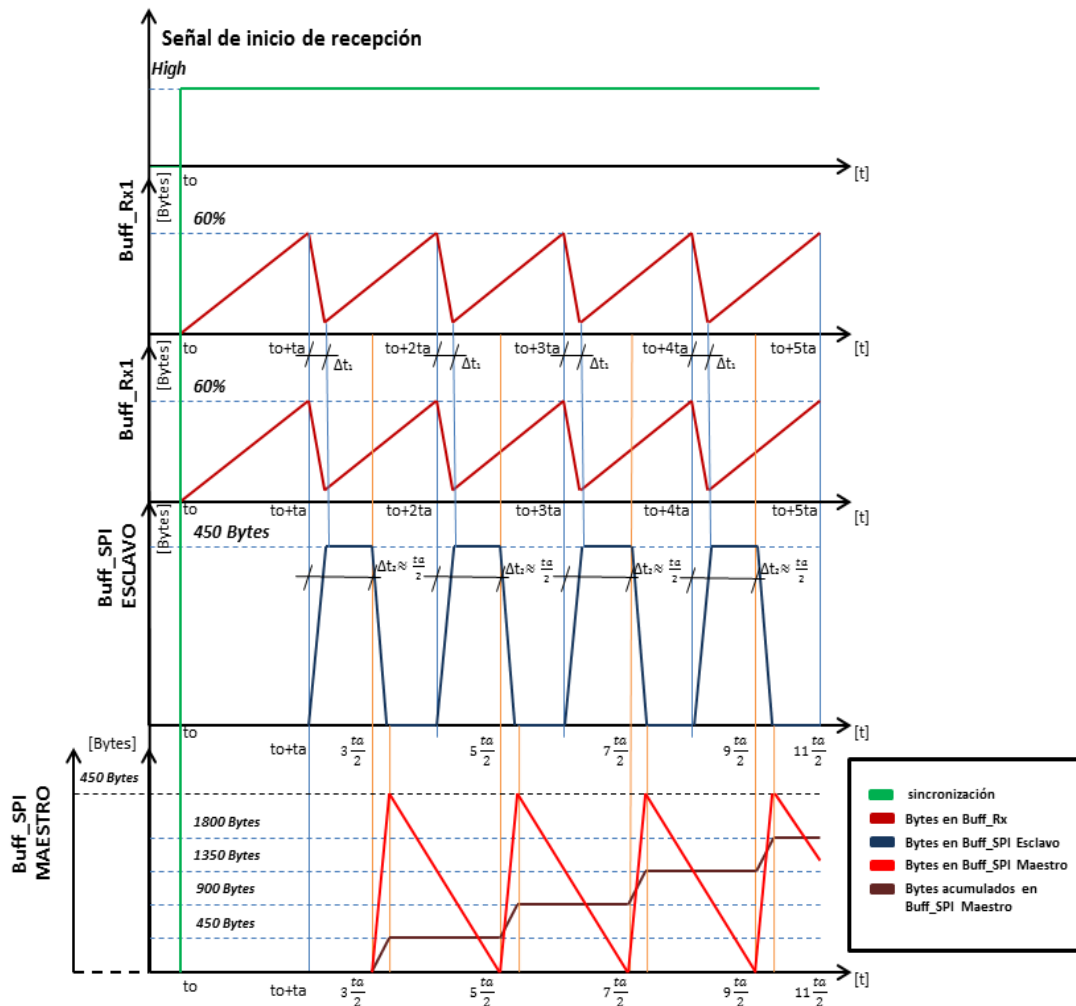


**Figura 28. Tiempo de vaciado  $\Delta t_1$  de arreglos de datos de llegada y de requerimientos desde Maestro.**

El guardado de los datos sustraídos de ambos arreglos se hace de manera ordenada, colocando primero los bytes 0xC1 y 0xC2 seguido de 16 muestras de 7 bytes cada una, intercalando muestras de canal 1 y canal 2 cada vez, quedando una trama de datos estructurada como la de la Figura 31.

Debido a que el tiempo de transferencia de datos de los arreglos Buff\_Rx1 y Buff\_Rx2 a Buff\_SPI es mucho menor a la mitad del tiempo “ $t_a$ ” ( $\Delta t_1 < t_a$ ) es seguro que la cantidad prevista de datos existentes en los arreglos Buff\_Rx1 y Buff\_Rx2 han sido transferidos al arreglo Buff\_SPI a mitad del siguiente periodo  $t_a$  donde nuevamente serán tomados los datos de los arreglos, se estableció que a mitad del periodo de vaciado de Buff\_Rx1 y Buff\_Rx2 se realice la petición por parte del módulo maestro de la cantidad de datos correspondientes a una trama por medio del periférico SPI, el cual se configuró a una transmisión de 1Mhz, estos datos son guardados también en un arreglo de datos Buff\_SPI de un tamaño de 4096 Bytes cada

$t_a$  segundos empezando en el momento  $1.5 t_a$ . En la Figura 28 se ilustra el diagrama de tiempos de la dinámica de transferencia de datos descrita.



**Figura 29. Diagrama de tiempo completo de la dinámica de transferencia de datos en funcionamiento de interfaz.**

### 3.4.4 Interfaz PC

La interfaz PC se realizó mediante la programación de un instrumento virtual elaborado con el paquete de programación LABIEW. La recepción serial se realizó

utilizando los módulos VISA propios para trabajar con señales seriales bajo protocolo RS-232.

Una vez recibidos los datos provenientes del prototipo en el instrumento virtual, se reciben dichos datos en un arreglo, seguidamente se procede a identificar las cabeceras (0xC1, 0xC2) de las supertramas y seccionar la cantidad de bytes correspondientes a cada canal (7 bytes).

Se envían estos datos a dos arreglos, uno para cada canal y posteriormente se realiza la identificación en los nuevos arreglos de las cabeceras de muestras 0xF1 y 0xF2.

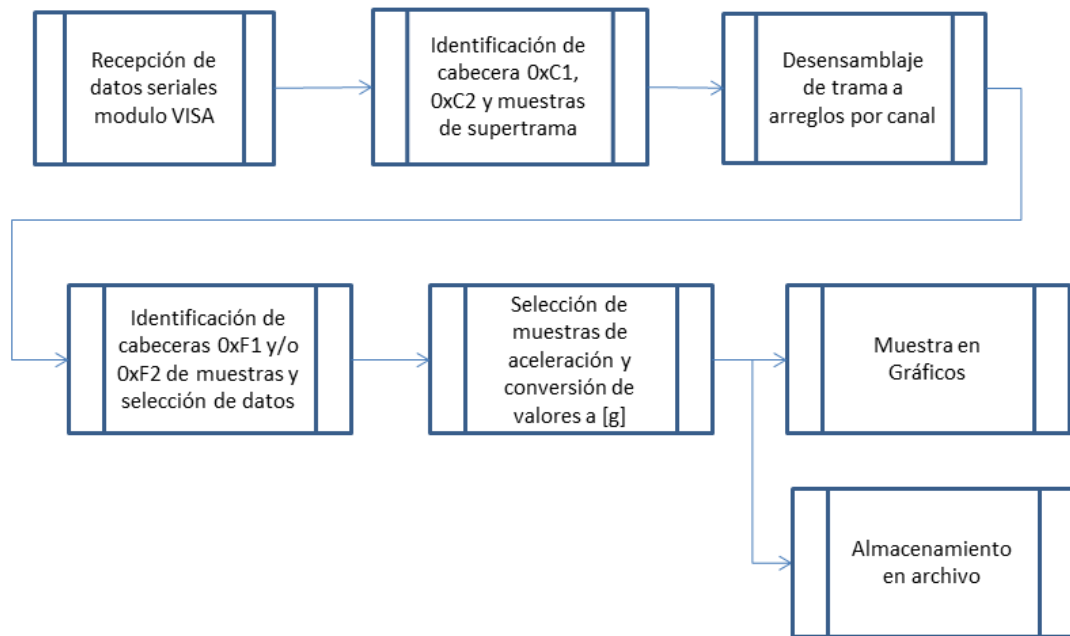
Se seleccionan y calculan los valores de aceleración asociados a las muestras recibidas para luego ser presentados en un instrumento virtual que grafica estos datos.

Los cálculos se realizan aplicando la fórmula presentada en la nota técnica #421 “Converting Measurement Values to Engineering Units “

$$(counts - offset) * gain = EU$$

Se toman los valores de offset y gain del reporte de configuración y haciendo la conversión de base hexagecimal a decimal de los dos bytes recibidos en la trama de datos para cada canal se realiza la conversión a valores de Gravedad.

En la Figura 30 se muestra el proceso que sigue la programación realizada en el instrumento virtual diseñado en LABVIEW.



**Figura 30. Proceso de conversión y presentación de datos en PC**



## CAPÍTULO IV

Durante el desarrollo de la interfaz fueron realizadas pruebas para verificar funcionamientos importantes a llevar a cabo por los módulos que componen el sistema interfaz.

### 4.1 Pruebas de comunicación con acelerómetros SUMMIT 35203A

En principio se utilizó el módulo *SparkFun* RS485-USB con el que fue posible configurar los sensores acelerómetros debido a que la comunicación serial con los acelerómetros se realiza bajo protocolo RS-485.

Esta configuración fue realizada utilizando el software propietario ICU. Se establecieron diferentes parámetros de funcionamiento para estudiar el comportamiento de los sensores acelerómetros. Se comprobaron los diferentes métodos de inicialización de para requerir muestras de aceleración descritos en el capítulo 2.

Las verificaciones de inicialización se realizaron mediante una interfaz hipertextual mediante la cual fue posible emular los comandos del protocolo propietario de los sensores. De estas pruebas se obtuvo que el método High Speed Measurement descrito en la nota técnica #412 y el método de recepción de datos al iniciar el acelerómetro resulta el más adecuado. Se realizaron pruebas a diferentes velocidades todas satisfactorias.

Se programó en simulación mediante software Proteus ISIS un microcontrolador virtual para emular la dinámica de inicialización descrita en la nota

técnica 412 utilizando el módulo *SparkFun* RS485-USB para recibir la trama de datos en software proteus con bytes identificables 0xEB y 0x90 como cabecera de trama muestra, tomando este reconocimiento como verificación de transmisión satisfactoria.

Utilizando un circuito integrado MAX-485 con la topología recomendada en norma TIA-485 y un microcontrolador pic24fj64ga002 se replicó el experimento de inicialización del acelerómetro solamente enviando los comandos de dinámica de inicialización High Speed Measurement al acelerómetro, logrando iniciar la transmisión de datos e identificando también los bytes 0xEB y 0x90 como cabecera de trama.

#### **4.2 Administración de inicialización con acelerómetros SUMMIT 35203A**

Se desarrolló una placa de circuito prototipo (Anexo 4) con la topología anteriormente probada definida como modulo esclavo y se implementó la dinámica de inicialización, esta vez se procedió a verificar los comandos recibidos lo cual requirió la administración del habilitador de transmisión-recepción del circuito integrado MAX-485, el funcionamiento no fue satisfactorio. Mediante análisis de la dinámica de inicialización se descubrió funcionamiento diferente al descrito en hoja de datos para interrupción por transmisión serial en la lógica esperada del microcontrolador debido a que el cambio en la señal de habilitación transmisión - recepción era realizado en un intervalo de tiempo diferente al descrito en manual de funcionamiento, ocasionando que el comando final de inicialización fuera enviado erróneamente.

Se realizó prueba de funcionamiento configurando método de inicio de transmisión de trama de datos al energizar el acelerómetro, este método permitió simplificar la dinámica de inicialización previamente implementada y se logró obtener resultado satisfactorio de inicialización al recibir trama de datos.

En la trama de datos recibidos el patrón cabecera de trama 0xEB, 0x90 no estaba presente, en su lugar se observó patrón 0x6F, 0x1F. La obtención de este patrón errado fue analizada encontrándose que la resistencia típica de 220  $\Omega$  indicada en la topología clásica del MAX-485 tenía un efecto de carga que incidía en el cambio de valores recibidos. Al retirar la resistencia el funcionamiento de recepción de datos fue satisfactorio teniéndose en la trama de datos el patrón 0xEB, 0x90

#### **4.3 Transmisión SPI Maestro-Esclavo**

Para verificar la transmisión satisfactoria mediante periférico SPI entre maestro-esclavo fue programada una rutina de requerimiento de datos desde el maestro a una velocidad de 1Mhz, y una rutina de envío por UART también en módulo maestro a una velocidad de 115.200 baudios de los datos de llegada donde fue verificado el patrón 0xEB, 0x90 en los bytes transmitidos de manera serial por el módulo maestro. Mediante un hiperterminal asociado al UART de transmisión fue posible esta verificación.

#### **4.4 Ajuste de tamaño de trama requerida por SPI**

Fue necesario determinar los valores de tiempo y tamaño para un funcionamiento aceptable de manera de no perder datos recibidos, para lograr esto fueron tomados los siguientes criterios.

En principio se estableció el tamaño de los arreglos Buff\_Rx1 y Buff\_Rx2 con un valor de 512 bytes y de 4096Bytes para el arreglo Buff\_SPI, ya se determinó que el tiempo en el cual estos arreglos Buff\_RxN se llenan completamente es de 44.4mseg así que se tomó como segundo criterio importante vaciar los datos de este arreglo al llegar al 60% de su capacidad.

$$t_{Buff_{Rx}@60\%} = 44.4mseg * 60\% = 26.64mseg$$

Una vez establecido este tiempo como periodo de vaciado de ambos arreglos se calculó la cantidad de bytes a vaciar sabiendo que el tiempo de un byte a 115.200 baudios es de 86,8useg

$$N_{Bytes@6\ 0\%} = \frac{t_{Buff_{Rx}@6\ 0\%}}{t_{Byte@115,2k}} = \frac{26.64mseg}{86.8\mu seg} \cong 306\ Bytes$$

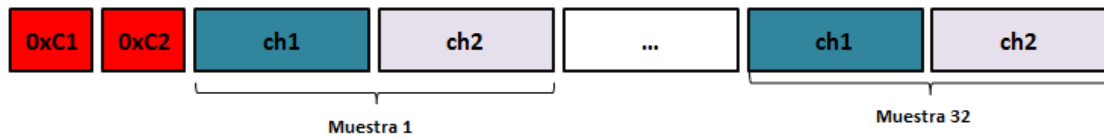
Como se tienen dos canales entonces es necesario vaciar 612 Bytes cada 26.64mseg.

$$N_{total\_Bytes@6\ 0\%} = 2 * N_{Bytes@6\ 0\%} = 612\ Bytes$$

De manera que se implementó una rutina con una interrupción de Timer en el microcontrolador esclavo que cada 26.65mseg tomara datos de ambos arreglos Buff\_Rx1 y Buff\_Rx2. Cada muestra guardada en estos arreglos tiene un tamaño de 7 bytes por lo que 612Bytes guardados en ambos arreglos representan un total de 44 muestras correspondientes a 22 muestras de cada canal.

$$N_{total\_muestras@6\ 0\%} = \frac{N_{total\_Bytes@6\ 0\%}}{Bytes\_muestra} = \frac{612\ Bytes}{7Bytes} \cong 44\ muestras$$

Experimentalmente se determinó que la cantidad de 44 muestras requeridas de los arreglos Buff\_Rx1 y Buff\_Rx2 supera la cantidad de datos existente en los arreglos, por lo que de manera también experimental se observó que al determinar una petición de 32 muestras (16 por cada canal) el ordenamiento de datos en el arreglo Buff\_SPI es más adecuado, de esta manera se optó por fijar que la trama de datos SPI definida para el ensamblado de la supertrama fuera de 32 muestras correspondientes a 450 Bytes. Figura 31



**Figura 31. Supertrama final fijada luego de ajuste por observación de comportamiento.**

En la rutina establecida en el módulo maestro para petición de datos vía SPI se solicitan en cada periodo de activación un total de 450 bytes.

#### **4.5 Recepción de datos en computador personal**

El funcionamiento del sistema en total fue probado enviando un tren de datos al computador personal y registrado en un archivo de texto por medio de un programa terminal de donde se pudo observar presencia de la supertrama enviada al computador personal. La Figura 32 muestra el archivo de texto donde han sido guardados los datos enviados, se señalan los bytes cabecera de trama, 5 muestras en recuadro y todos los bytes 0xF1 y 0xF2 que señalan el inicio de una muestra por canal. En el paquete de Software LabVIEW se realizó la recepción y procesamiento de datos recibidos por la interfaz, en la Figura 33 se muestra una ventana de tiempo de los datos de aceleración graficados por el programa interfaz.

En las **figuras 33-38** se muestran las gráficas de aceleración como resultados obtenidos en el instrumento virtual desarrollado en labVIEW del sistema interfaz encargado de procesar los datos enviados desde el prototipo, en cada una de las figuras se presenta los valores positivos y negativos de gravedad ( $\pm 1$  [g]) de cada uno de los ejes coordenados para ambos acelerómetros funcionando simultáneamente. Los diferentes valores de aceleración son presentados en colores: blanco, rojo, verde para los ejes X, Y, Z.

De estas Figuras la ventana 1(superior) corresponde a los datos del canal 1 y los de la ventana 2(inferior) al canal 2. La ventana 1 y ventana 2 difieren en forma y es debido a que en el canal 1 se implementó un filtro digital de primer orden de tipo:

$$\begin{aligned}y(k) &= y(k - 1) \\y(k) &= a_1y(k - 1) + b_1y(k - 1) \\u(k - 1) &= 1; \quad u(k) = 1 \\ \text{con } y(k) &= 1; \quad a_1 + b_1 = 1\end{aligned}$$

En la ventana 2 se tienen las señales de aceleración tal cual llegan desde el prototipo interfaz. Se observa que las señales de aceleración que llegan sin tratamiento poseen variaciones importantes, fluctuaciones que interfieren con la interpretación correcta de los valores reales de aceleración, se supone que estas variaciones están asociadas a errores de parámetros en los tamaños, tiempos y sincronización en la dinámica de traspaso de datos entre los acelerómetros, módulos maestro-esclavo-PC. De esta manera la implementación de filtro digital de primer orden se realizó con la intención de, por medio de una técnica de tratamiento de señales eliminar las fluctuaciones no deseadas en las señales recolectadas. Sin embargo se observa que el filtro implementado altera de manera significativa los valores obtenidos además de la eliminación de fluctuaciones no deseadas en las señales recolectadas.

#### **4.6 Almacenamiento de Datos**

Para este dispositivo interfaz se realizó el diseño de manera de obtener una salida serial directamente del dispositivo a un módulo FAT para almacenar los datos en una memoria SD, las pruebas realizadas consistieron en la adquisición de datos mediante el uso del UART2 del microcontrolador a un módulo FAT comercial

modelo *OpenLog* del fabricante *SparkFun*, aunque el modulo FAT realiza la adquisición, el experimento no fue completamente satisfactorio debido a que el modulo recibe datos hasta una velocidad de 115.200 baudios y por lo expuesto en el apartado 3.3.1 la velocidad mínima de transmisión debe ser de 230.760 baudios de manera que solo se envía la mitad de los datos de la trama en el tiempo que a 250k baudios se realiza la transmisión completa. En el computador personal se realizaron dos pruebas de almacenamiento.

La primera consistió de almacenar las tramas de datos enviadas desde el módulo maestro sin realizar el procesamiento, esto a una velocidad de transmisión de 250k baudios y mediante el software terminal *PuTTYPortable* que permite almacenar en un archivo “\*.tex” los datos recibidos.

La segunda prueba consistió en el almacenamiento de los datos procesados en un archivo de salida mediante el software LabView que permite mediante un instrumento virtual propio del conjunto de herramientas por defecto del programa guardar datos recibidos.

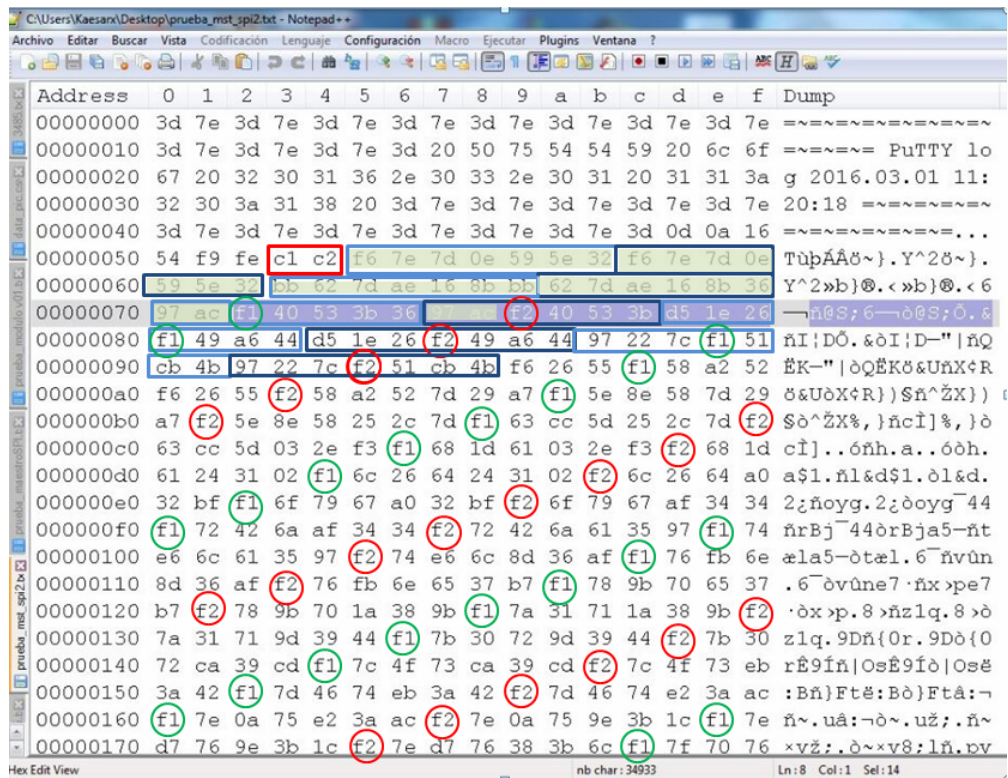


Figura 32. Captura de datos recibidos en computador personal

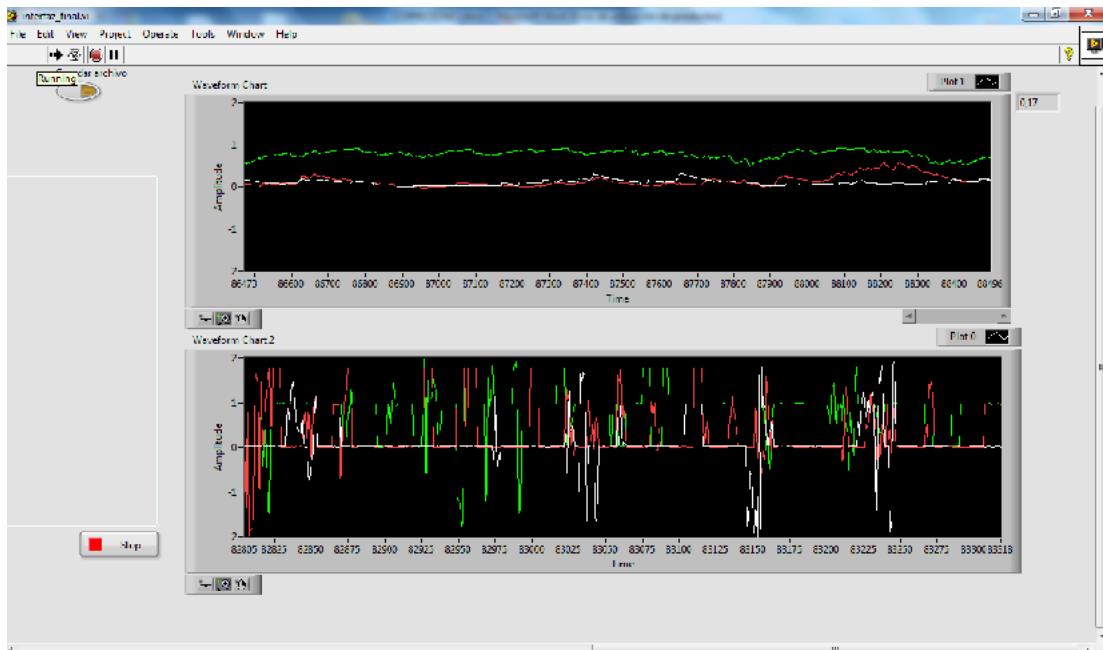
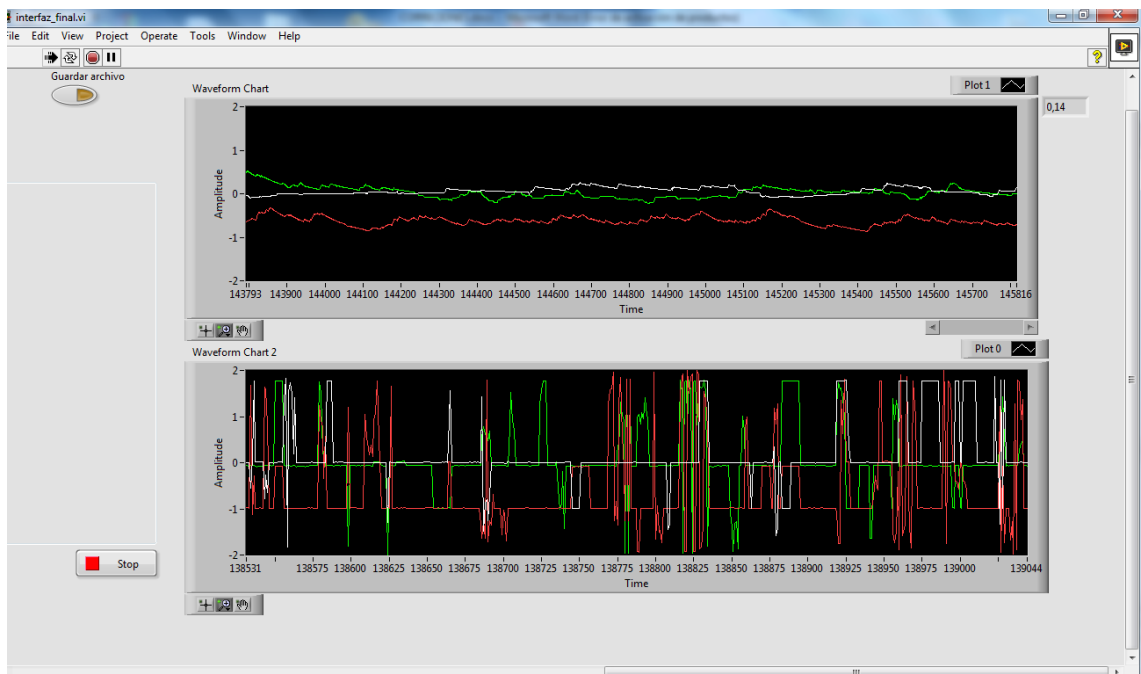


Figura 33. Datos presentados en Interfaz PC usando paquete de Software LabVIEW.





**Figura 34. Captura de datos para eje 1 aceleración negativa.**



**Figura 35. Captura de datos para aceleración eje 3 negativa.**

## CONCLUSIONES

A partir del desarrollo del trabajo se puede concluir lo siguiente:

Se implementó una interfaz capaz de manejar de forma simultanea dos sensores de aceleración triaxiales digitales, de la casa SUMMIT Instruments modelo 35203A. Para visualizar los datos de aceleración en tiempo real se desarrolló además, un programa en la plataforma LabVIEW en su versión 2010, capaz de recibir mediante un puerto serial (USB) la información de interés para luego escalarla, presentarla en pantalla mediante gráficos en el tiempo y almacenarla en disco si es requerido.

Para obtener este resultado se llevaron a cabo los siguientes aspectos:

- ✓ Mediante el estudio de los métodos descritos en las notas técnicas dadas por el fabricante de los acelerómetros SUMMIT INSTRUMENTS 35203A y los resultados de las pruebas de comunicación, fue posible escoger una metodología para realizar la comunicación, denominado *Hight Speed Measurements*.
- ✓ Se planteó un sistema electrónico programable para el *hardware* basado en microcontroladores PIC24fj64ga002 integrados en una topología maestro-esclavo el cual permite establecer la comunicación, recepción, almacenamiento de datos y transmisión de estos hacia un computador vía serial, por lo cual esta interfaz se encuentra entre los acelerómetros y el computador.
- ✓ Se construyó un prototipo de la interfaz, para lo cual se desarrolló el firmware de los microcontroladores seleccionados y además se realizó un programa basado

en la plataforma LabVIEW capaz de establecer comunicación con la interfaz electrónica, para de esta forma procesar los datos y presentar la información de aceleración en el tiempo y almacenarla.

- ✓ Mediante el encendido simultáneo de los acelerómetros digitales se persigue que el muestreo de las señales de aceleración se haga de manera simultánea lo cual se logra si el tiempo de encendido y configuración es idéntico para los acelerómetros que integran el sistema.
- ✓ Se realizó un manual de operación a fin de describir el uso de la interfaz diseñada.

Los siguientes aspectos fueron determinantes en la realización de este proyecto:

- ✓ La velocidad de transmisión es condicionante importante para el diseño de dispositivos digitales de transmisión de datos.
- ✓ En los ensayos civiles el ancho de banda de las señales de aceleración es del orden de 30Hz a 60Hz, la selección de la velocidad de transmisión de los acelerómetros digitales con los que trabaja esta interfaz permite el estudio de señales con un ancho de banda hasta 100Hz.
- ✓ Por condicionante de las características de los microcontroladores utilizados en esta implementación además de las funciones requeridas, el número máximo de acelerómetros digitales que puede manejar la interfaz es de dos acelerómetros.

## RECOMENDACIONES

- ✓ Hacer portátil el dispositivo de manera que pueda funcionar con alimentación por banco de baterías.
- ✓ Hacer la recolección de datos realizada por el módulo esclavo de los acelerómetros mediante un módulo inalámbrico para dar mayor versatilidad al equipo en el aspecto de accesibilidad.
- ✓ Si se desea ampliar el número de sensores acelerómetros a utilizar por la interfaz se recomienda utilizar un dispositivo electrónico más robusto como modulo maestro bien pudiera ser un FPGA o un SBC tipo *Galileo*, *PCDuino*, *Raspberry PI* u otro a fin de contar con elementos más potentes en hardware y software para un mejor procesamiento y gestión de las funciones deseadas.

## **PROPUESTA DE SOLUCIÓN A PROBLEMÁTICA EN RESULTADOS OBTENIDOS.**

### **Reprogramación de Firmware en Interfaz.**

Intercambio de modo maestro-esclavo SPI entre Placa A y Placa B llegando a ser ahora la placa receptora de los datos de acelerómetros la placa Maestro. Esta recomendación se hace con la intención de que la recolección se haga en función de la disponibilidad de datos en la tarjeta conectada a los acelerómetros y no en función del requerimiento de la tarjeta maestro a fin de evitar pérdida de datos.

#### **En nueva tarjeta maestro:**

- Agregar byte de muestreo a trama de muestra
- Bajar velocidad de transmisión serial a 57600 baudios
- Reconfiguración a modo Maestro SPI (Maestro)
- Configuración de interrupción de recepción serial en modo 3 para interrupción al tener FIFO lleno cada recepción de 4 bytes.
- Implementación de buffer cíclico (Estudiar tamaño según tiempos de funcionamiento) sólo de recepción de bytes en rutina de interrupción Rx.
- Implementación de función de verificación de muestras validas de datos en buffer recibido Rx y paso de los mismos a Buffer Val\_Rx. Se requiere que la función verifique y valide las muestras recogidas en los datos colectados y descarte muestras erróneas.
- Implementación de función para ensamblado de trama, la función debe armar supertrama con cabecera para identificación, tomar las muestras de los buffers Val\_Rx1 y Val\_Rx2 de ambos canales e intercalar muestras de canal 1 y canal 2 en construcción de supertrama.

- Implementación de función de intercambio SPI maestro-esclavo, para enviar al nuevo módulo esclavo datos una vez armada una supertrama.

#### **En nueva tarjeta Esclavo:**

- Reconfiguración de pines y periféricos
- Reconfiguración de modo SPI (Esclavo)
- Implementación de interrupción SPI (Esclavo) para recepción de datos de placa A.
- Implementación de buffer cíclico para recepción de datos SPI; se debe estudiar el tamaño del buffer en función de los parámetros de tiempo y cantidad de datos recibidos para evitar pérdida de los mismos.
- Implementación de buffer cíclico para datos a ser enviados por UART de manera que exista separación entre función y datos de llegada vía SPI y aquellos a ser enviados de manera serial por UART.
- Implementación de función para colocar en nuevo buffer cíclico Buff\_Tx al que sólo tendrá acceso la función o interrupción encargada de realizar el envío serial.
- Implementación de Interrupción timer o Interrupción serial de transmisión. La rutina de interrupción deberá cargar datos previamente almacenados en buffer cíclico Buff\_Tx.

#### **Ajuste de Software de Recepción en PC.**

- Reconfiguración de valores de byte de reconocimiento en lógica de instrumento virtual encargado de realizar desensamblaje de supertrama transmitida.
- Reconfiguración de número de bytes según número establecido en configuración previa de trama muestra en acelerómetro.

- Reconfiguración de valores de byte de señalización de muestras por canales individuales previamente establecidos en interfaz prototipo.

## BIBLIOGRAFÍA

SUMMIT INSTRUMENTS.(s.f.). A beginner's Guide to using the Digital Accelerometer. Obtenido de [http://precisionsensors.meas-spec.com/pdfs/Beginners\\_Guide\\_Digital.pdf](http://precisionsensors.meas-spec.com/pdfs/Beginners_Guide_Digital.pdf)

SUMMIT INSTRUMENTS. (s.f.). Instrument Configuration Utility Reference Manual. Obtenido de <http://precisionsensors.meas-spec.com/pdfs/ICURef.pdf>

SUMMIT INSTRUMENTS. (s.f.). Instrument Configuration Utility User's Manual. Obtenido de <http://precisionsensors.meas-spec.com/pdfs/ICUUser.pdf>

SUMMIT INSTRUMENTS. (s.f.). Technical Note 409 - IRIG Encoding External Data. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/IRIG%20Encoding%20External%20DataTN409.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/IRIG%20Encoding%20External%20DataTN409.pdf)

SUMMIT INSTRUMENTS (s.f.). Technical Note 410 - CRC16 Algorithm. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/CRC16%20AlgorithmTN410.v1.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/CRC16%20AlgorithmTN410.v1.pdf)

SUMMIT INSTRUMENTS (s.f.). Technical Note 411 - Acquiring Digital Sensor Measurements. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/Acquiring%20Digital%20Sensor%20MeasurementsTN411.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/Acquiring%20Digital%20Sensor%20MeasurementsTN411.pdf)

SUMMIT INSTRUMENTS (s.f.). Technical Note 412 - High-Speed Digital Sensor Measurements. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/High%20Speed%20Digital%20Sensor%20MeasurementsTN412.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/High%20Speed%20Digital%20Sensor%20MeasurementsTN412.pdf)



SUMMIT INSTRUMENTS (s.f.). Technical Note 415 - Reading Application Information from a Platinum Digital Sensor. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/Reading%20Application%20Information%20from%20a%20Digital%20SensorTN415.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/Reading%20Application%20Information%20from%20a%20Digital%20SensorTN415.pdf)

SUMMIT INSTRUMENTS (s.f.). Technical Note 421 - Converting Measurement Values to Engineering Units. Obtenido de [http://precisionsensors.meas-spec.com/knowledge\\_center/pdf/Converting%20Measurement%20Values%20to%20Engineering%20UnitsTN421.pdf](http://precisionsensors.meas-spec.com/knowledge_center/pdf/Converting%20Measurement%20Values%20to%20Engineering%20UnitsTN421.pdf)

ALVARES, S. (2009). Trabajo Especial de Grado, EQUIPO ELECTRÓNICO PARA LA ESTIMACIÓN DEL ÍNDICE DE RUGOSIDAD INTERNACIONAL (IRI) CON BASE EN EL PERFILÓMETRO MERLIN. Caracas.

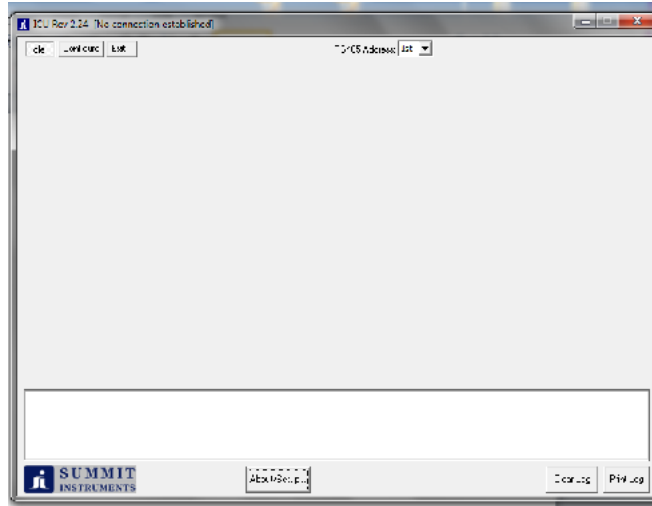
MÁRQUEZ, I. (2010). Trabajo Especial de Grado, DISEÑO DE UNA UNIDAD DE ADQUISICIÓN DE DATOS PARA ESTACIONES HIDROMETEREOLÓGICAS DE SUPERFICIE, ADAPTADA A LAS NECESIDADES DEL PAÍS. Caracas.

GONZÁLEZ, E. (2011). Trabajo Especial de Grado, DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO DE ADQUISICIÓN DE DATOS PARA ESTACIONES SISMOLÓGICAS. Caracas.

## **ANEXOS**

## 5.1 [Notas Rápidas sobre uso de ICU]

### 1.- Abriendo ICU



### 2.- Configuración de transmisión:

Puerto COM a utilizar (COM Port)

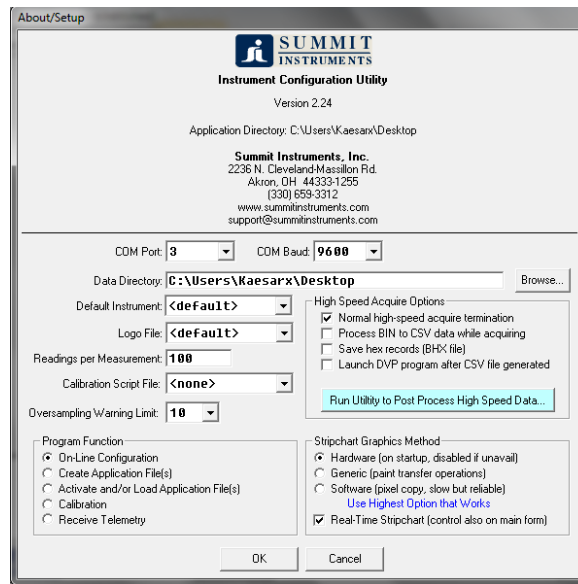
Velocidad de conexión de puerto serial (COM Baud)

Instrumento a utilizar (En esta versión de ICU se deja por defecto, en otras versiones se selecciona 35203A )

Lecturas por medida

Límite de Oversamplig

La función del programa se selecciona en configuración on-line

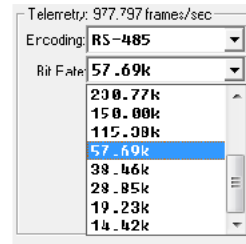
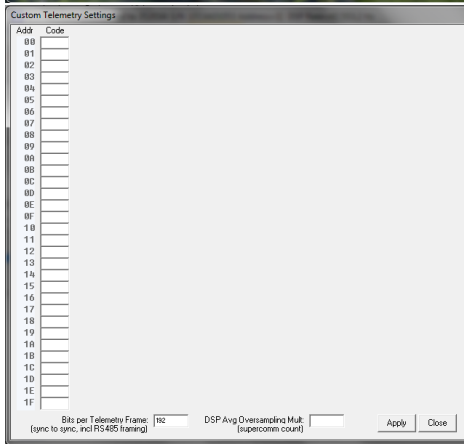
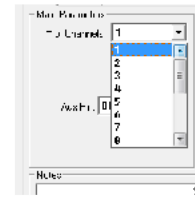
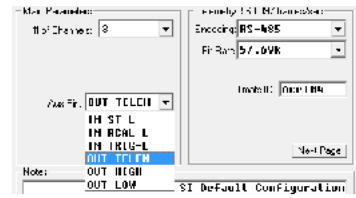
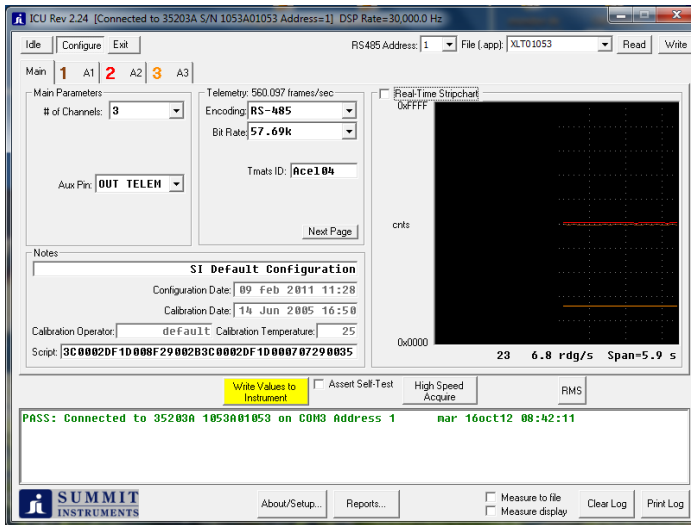


En la pantalla de configuración se tienen varios parámetros a configurar:

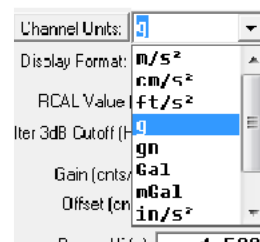
-#N° de canales (de 1 a 16 canales)

-Aux Pin: con 6 distintas variables IN ST-L, IN RCAL-L, INTRIG-L, OUT TELEM, OUT HIGH, OUT LOW

-Telemetría: configura los parámetros de la telemetría, encoding = RS-485, Bit rate = 14.43k -3.0M, Tmats ID = Ace104. Es posible hacer una configuración más detallada de la telemetría : sync, frame Id, S/N, Pad, CRC, Comm ID, se puede activar el modo Stream RS485 on Power-up, y ajustar una telemetría personalizada



Para cada canal hay una pestaña donde se obtienen algunos parametros importantes y otros de entrada a configurar.



Main **1** A1 **2** A2 **3** A3

Input Select  
 Invert Source: **A1**

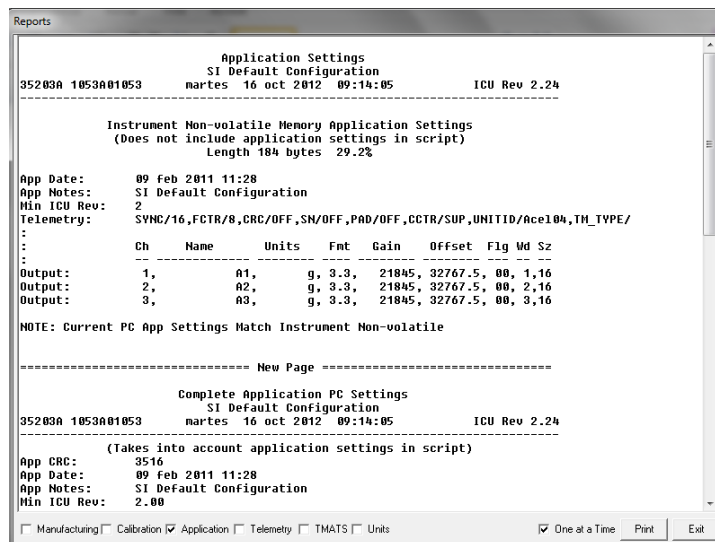
Input Sensor Cal Values  
Gain (cnts/g): **3298.3**  
Offset (cnts): **26269.6**

Output Dac Cal Values  
Gain (cnts/V): **13088.7**  
Offset (cnts for 0V): **17.6**

Output Settings  
Channel Name: **A1**  
Channel Units: **g**  
Display Format: **3.3**  
RCAL Value (g): **-0.5**  
Filter 3dB Cutoff (Hz): **29.96**  
Gain (cnts/g): **21845**  
Offset (cnts): **32767.5**  
Range Hi (g): **1.500**  
Range Lo (g): **-1.500**  
Gain (g/cnt): **.00004577**  
DAC Corrected:

Telemetry Word Pos: **1**  
Telemetry Word Bits: **16**

Tambien se puede obtener un reporte con los valores que estan configurados en el sensor



Contenido del reporte:

Application Settings  
SI Default Configuration  
35203A 1053A01053 martes 16 oct 2012 09:16:19 ICU Rev 2.24

Instrument Non-volatile Memory Application Settings  
 (Does not include application settings in script)  
 Length 184 bytes 29.2%

```
App Date:      09 feb 2011 11:28
App Notes:     SI Default Configuration
Min ICU Rev:   2
Telemetry:     SYNC/16,FCCTR/8,CRC/OFF,SN/OFF,PAD/OFF,CCTR/SUP,UNITID/Ace104,TM_TYPE/
:
:              Ch   Name      Units   Fmt   Gain   Offset  Flg Wd Sz
:              ---
Output:        1,    A1,      g, 3.3, 21845, 32767.5, 00, 1,16
Output:        2,    A2,      g, 3.3, 21845, 32767.5, 00, 2,16
Output:        3,    A3,      g, 3.3, 21845, 32767.5, 00, 3,16
```

NOTE: Current PC App Settings Match Instrument Non-volatile

===== New Page =====

Complete Application PC Settings  
 SI Default Configuration

35203A 1053A01053      martes 16 oct 2012 09:16:19      ICU Rev 2.24

(Takes into account application settings in script)

```
App CRC:       3516
App Date:      09 feb 2011 11:28
App Notes:     SI Default Configuration
Min ICU Rev:   2.00
PCM b/s:       57.69k
PCM Enabled:   True
PCM CRC Bits:  OFF
PCM FCCTR Bits: 8
PCM CCTR Bits: SUP
TMATS Unit ID: Ace104
PCM S/N Bits:  OFF
PCM Pad Bits:  OFF
PCM SYNC Bits: 16
PCM Encoding:  RS-485
S/W Self-Test: False
Aux Mode:      OUT TELEM
DAC Enable:    True
Max Chan Rate: 560.097
DSP Rate:      30kHz (30000)
DSP Enable:    True
Custom TM:     False 103 1
PCM Def:       SYNC:0/16,1:1/16,2:2/16,3:3/16,FCCTR:4/8
2bScript:     3C0002DF1D008F29002B3C0002DF1D000707290035
```

```
:              Ch#  Src      Name      Units   Fmt   Filter   Gain   Offset  RCAL Flg Wd Sz
:              ---
Output:        1, +A1,      A1,      g, 3.3, 29.965, 21845, 32767.5, 602C, 00, 1,16
Output:        2, +A2,      A2,      g, 3.3, 29.965, 21845, 32767.5, 71F6, 00, 2,16
Output:        3, +A3,      A3,      g, 3.3, 29.965, 21845, 32767.5, 70C0, 00, 3,16
```

```
: ----- Range Limits -----
:  Ch#   Name      Units   Lower      Upper      Cause of Limit
: -----
:  1,    A1,      g,    -1.500,    1.500,  Lo trim by DSP
:  2,    A2,      g,    -1.500,    1.500,
:  3,    A3,      g,    -1.500,    1.500,
```

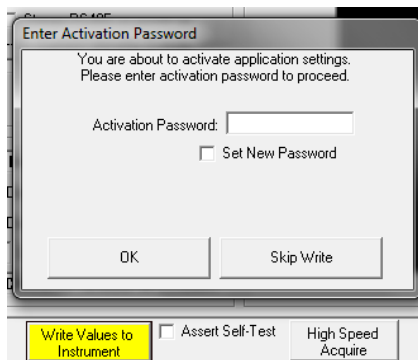
```
: ----- RCAL Outputs -----
:  Ch#   Name      Units   Programmed  Actual  Hex
: -----
RCAL:   1,    A1,      g,    -.500137,  -0.500, 5552
RCAL:   2,    A2,      g,    .499908,   0.500, AAA8
```

```

RCAL: 3,          A3,          g,  -.499908,  -0.500, 5557
:
: NOTE: RCAL hex outputs will be within one count.
:
: DAC Output Engineering Units Conversions
:      Gain      Offset      Eng
: Ch  (Volts/EU) (Volts)  Units
: ---
: 1,      1.669,  2.50215,    g
: 2,      1.66749, 2.50524,    g
: 3,      1.66796, 2.49942,    g

```

Luego de la configuración del dispositivo es necesario guardar la configuración, para esto es necesario colocar un password que por defecto es “SI” como acrónimo por defecto de *Summit Instruments*, esto se realiza haciendo click en el botón amarillo “Write Values to Instruments”





5.2 [MANUAL DE OPERACIÓN] <sup>[Anexo N° 2]</sup>

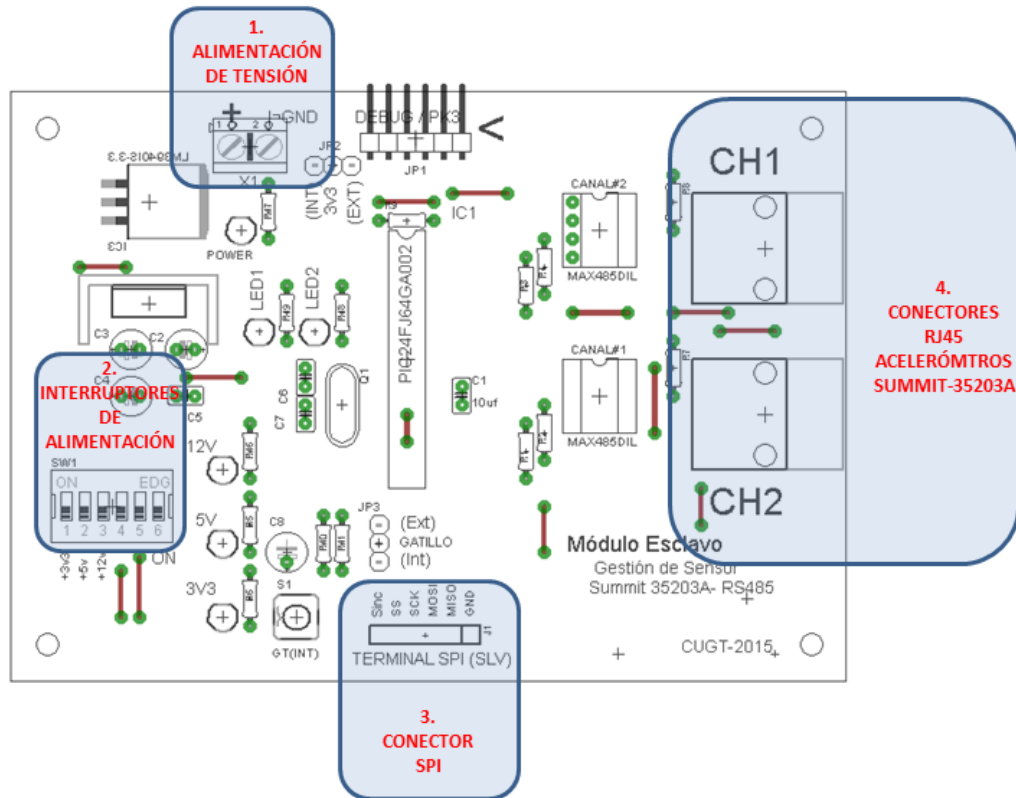
**INTERFAZ DE RECEPCIÓN DE  
DATOS DE ACELERÓMETROS  
SUMMIT INSTRUMENTS  
35203A**



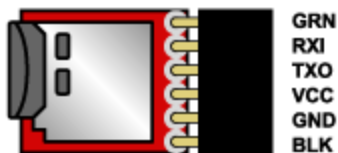
## Módulo Esclavo:

Encargado de gestionar recepción de datos de acelerómetros, sus elementos importantes son

1. Alimentación de tensión.
2. Interruptores de encendido.
3. Conector SPI
4. Conectores RJ45 de puertos seriales.



**Módulo FAT:** Gestiona almacenamiento de datos enviados desde el módulo esclavo

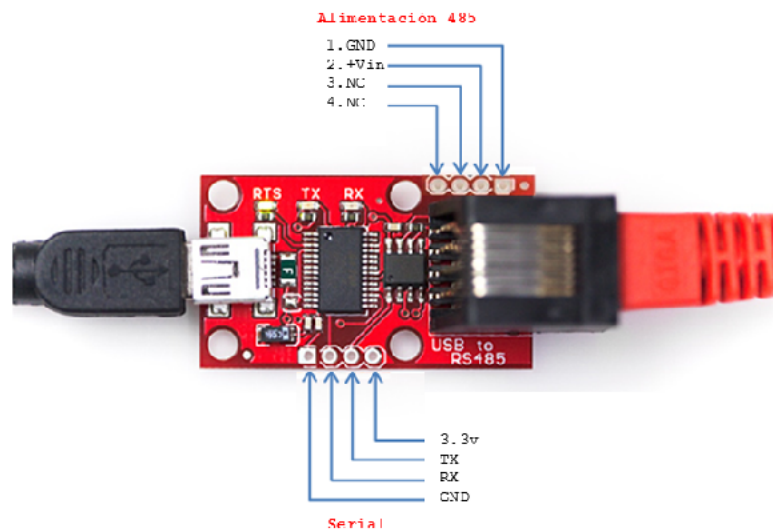


## Modulo Serial:

El módulo USB485 permite conexión USB-485 o USB-serial 232.

En caso de conexión USB-485 con el sensor para configuración la alimentación del sensor se realiza por el pin (2)+Vin/(1)GND.

Para transmisión de datos se utiliza el puerto serial del módulo.

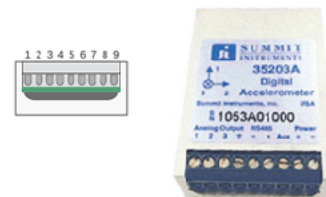


## Pinout de conexión de conector RJ45 de sensor acelerómetro.

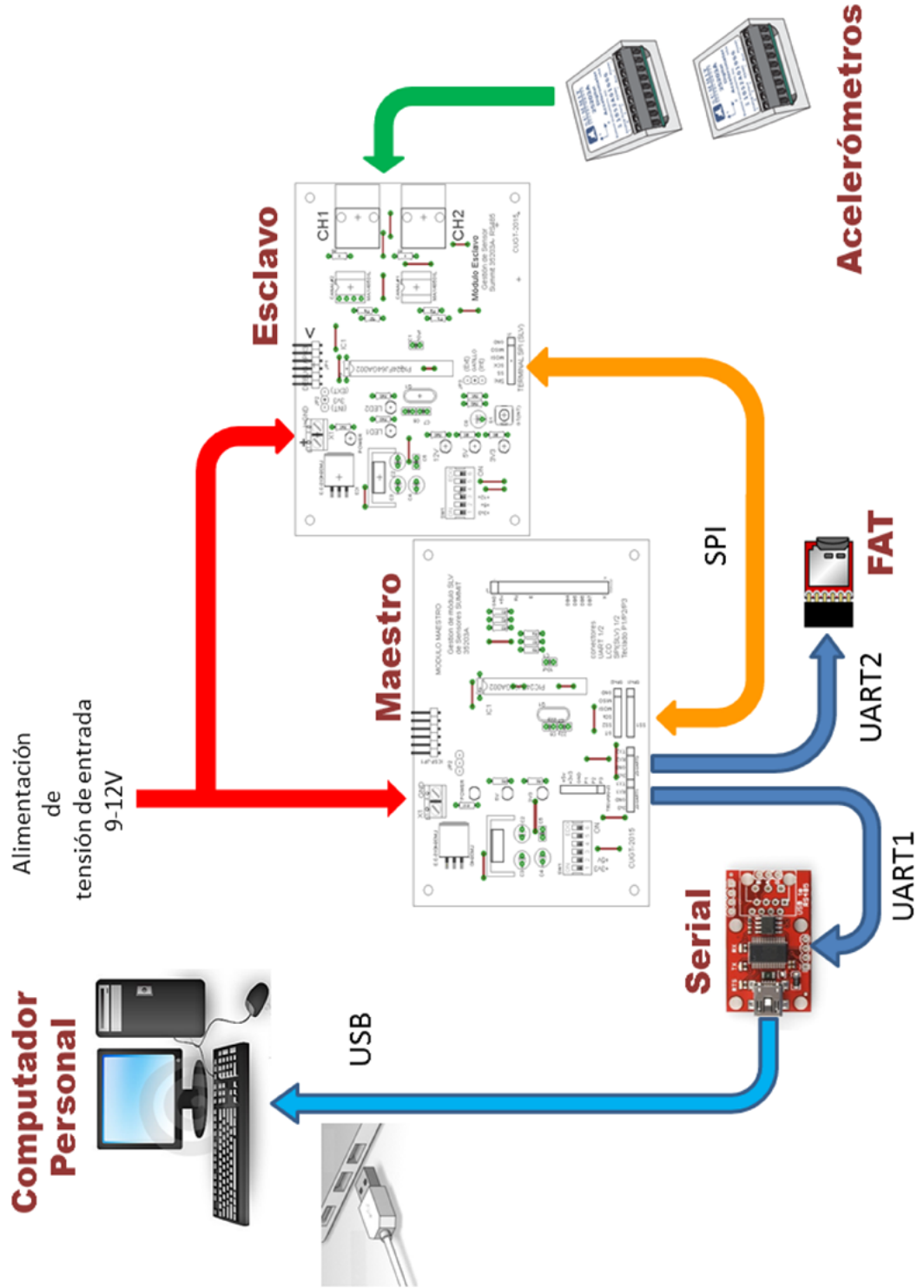


PINOUT RJ45		PINOUT SI35203A	
1	←	V-	9
2	←	V+	8
3	←	NC	-
4	←	V-	9
5	←	GND	4
6	←	NC	-
7	←	RS485-	5
8	←	RS485+	6

## Acelerómetro 35203A



# Diagrama de conexiones de Interfaz



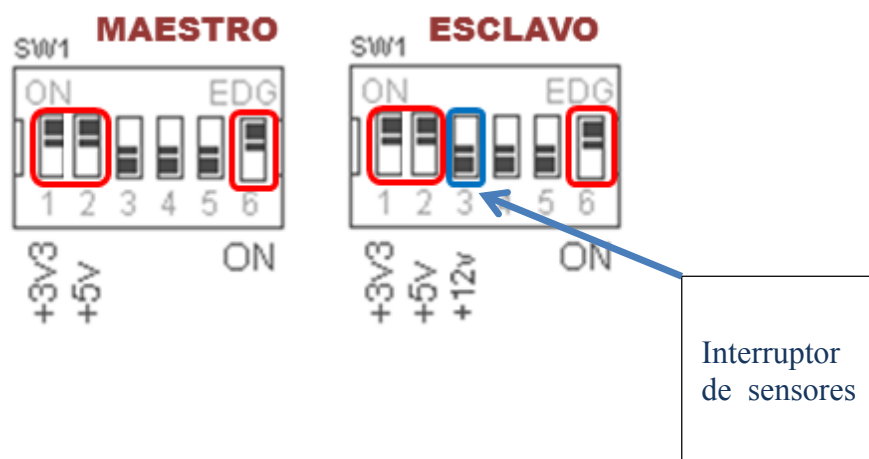
## PROCEDIMIENTO DE OPERACIÓN

Una vez realizadas las conexiones de la interfaz se realizan los siguientes pasos para coleccionar los datos.

1. Configurar puerto COM de entrada USB a 250K baudios.
2. Encender la alimentación de la interfaz
3. Iniciar el Instrumento virtual de recepción Labview.



4. Encender alimentación de tensión en módulos Maestro - Esclavo a excepción de alimentación de sensores.



5. Encender alimentación de sensores al momento de querer iniciar la transmisión de datos.

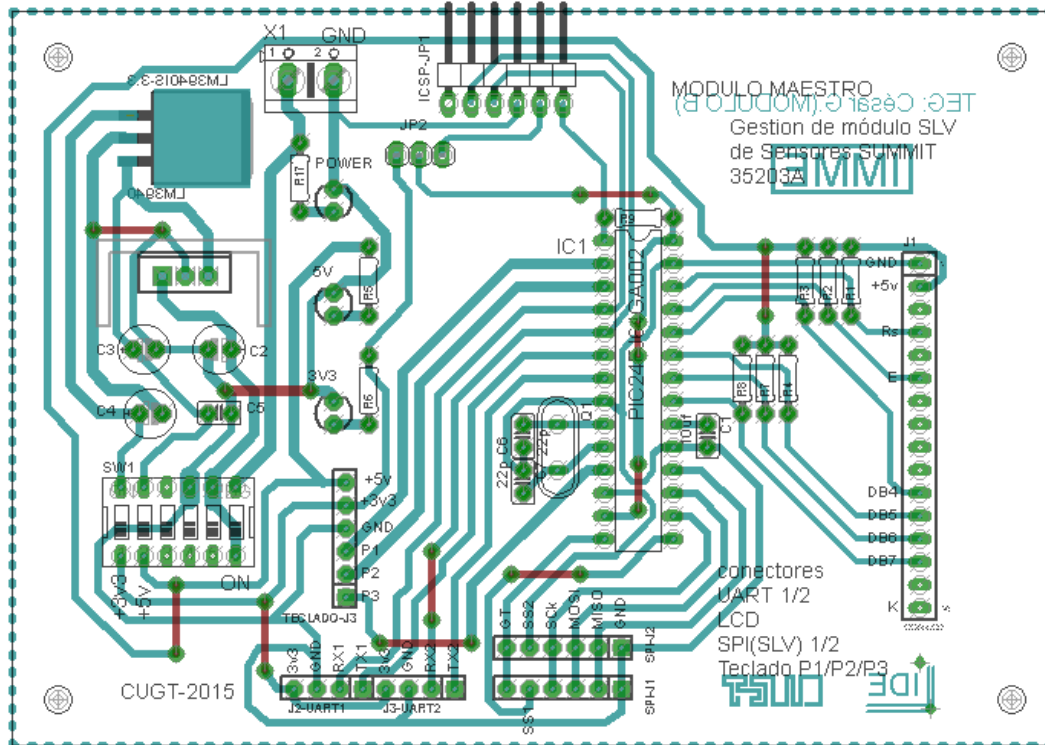


6. Al finalizar el experimento simplemente se desactiva la alimentación de los sensores y alimentación de los módulos

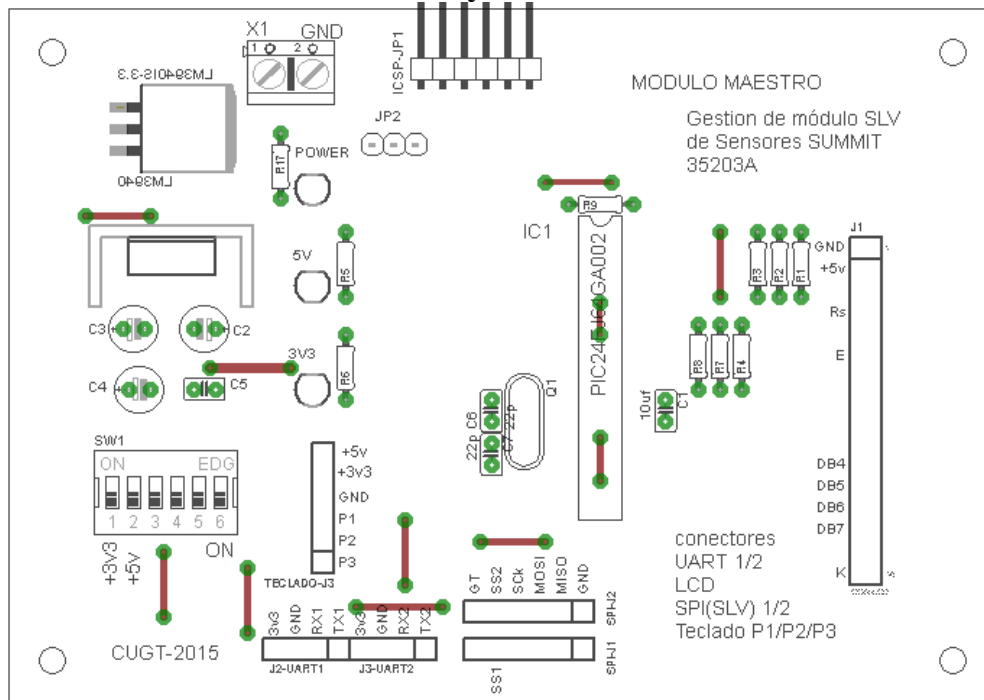




## Diagrama Esquemático Módulo Maestro



## Placa Layout Maestro



## Diagrama de Partes placa Maestro

### 5.4 [Pinout de Microcontrolador Esclavo]

MODULO ESCLAVO *																	
#PIN	Anx	RAN/F	CNx									RPN	RBN	#PIN			
1	-	MCLR		PK3	MCLR	PK3	0	Vdd	+	VDD	RP0	RB0	4	PK3		SPI1	
2	AN0	RA0	CN2	LIBRE	-	LIBRE	1	Vss	-	GND	RP1	RB1	5	PK3		UART1	
3	AN1	RA1	CN3	LIBRE	-	LIBRE	2	TX1	x	RB15	OD	RP2	RB2	6	LED	x	UART2
8	-	Vss	-	Vss	PGD	PK3/PGD	3	EN1	x	RB14	OD	RP3	RB3	7	LED	x	
9		RA2	CN30	OSC	PGC	PK3/PGC	4	TX2	x	RB13	OD	RP4	RB4	11	LIBRE	-	
10		RA3	CN29	OSC	RB2	X	LED	5	EN2	x	RB12	OD	RP5	RB5	14	GATILLO	o
12		RA4	CN0	LIBRE	RB3	X	LED	6	RX2	o	RB11		RP6	RB6	15	SS	o
13	-	Vdd	-	-	GND		Vss	7	RX1	o	RB10		RP7	RB7	16	SCK	o
19	-	DISVREG	-	GND			OSC	8	Vcap	-			RP8	RB8	17	MOSI	o
20	-	Vcap/Vddcore	-	-			OSC	9	DISVREG	GND			RP9	RB9	18	MISO	x
27	-	Vss	-	GND			LIBRE	10					RP10	RB10	21	RX1	o
28	-	Vdd	-	VDD			LIBRE	11	MISO	x	RB9	SDO	RP11	RB11	22	RX2	o
					VDD	Vdd		12	SCK	o	RB8	SDI	RP12	RB12	23	EN2	x OD
					RB5	X	sincronia	13	SCK	o	RB7	SCK	RP13	RB13	24	TX2	x OD
								14	SS	o	RB6	SS	RP14	RB14	25	EN1	x OD
													RP15	RB15	26	TX1	x OD

Pinout de Microcontrolador Esclavo

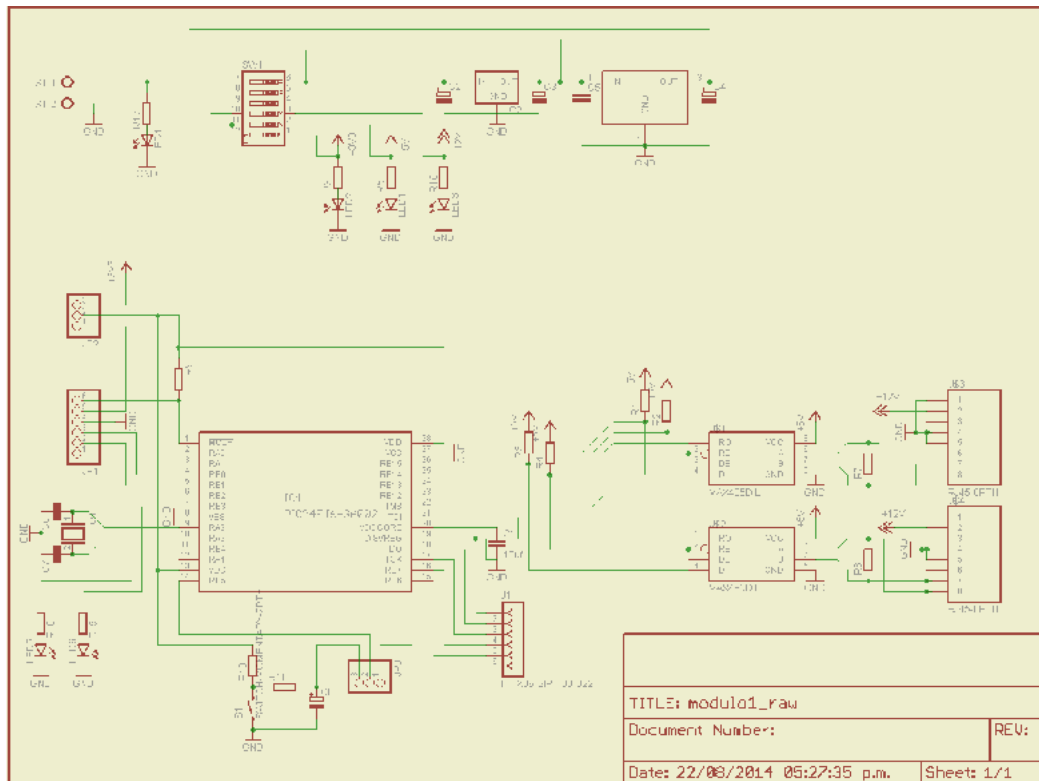
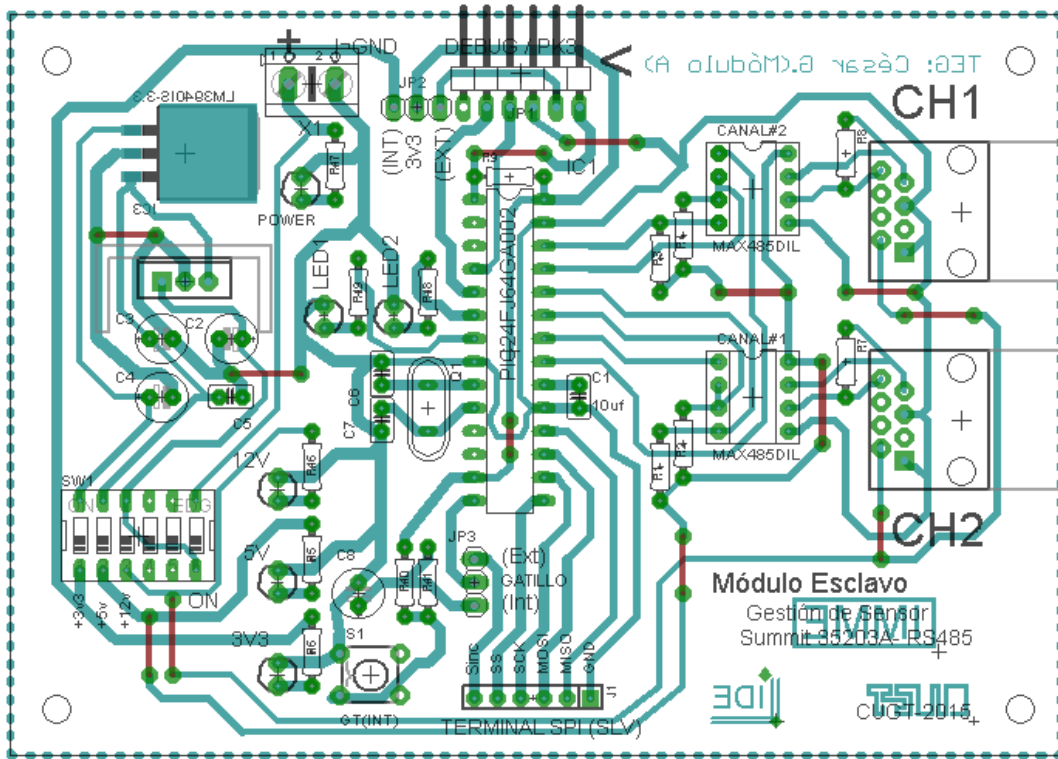
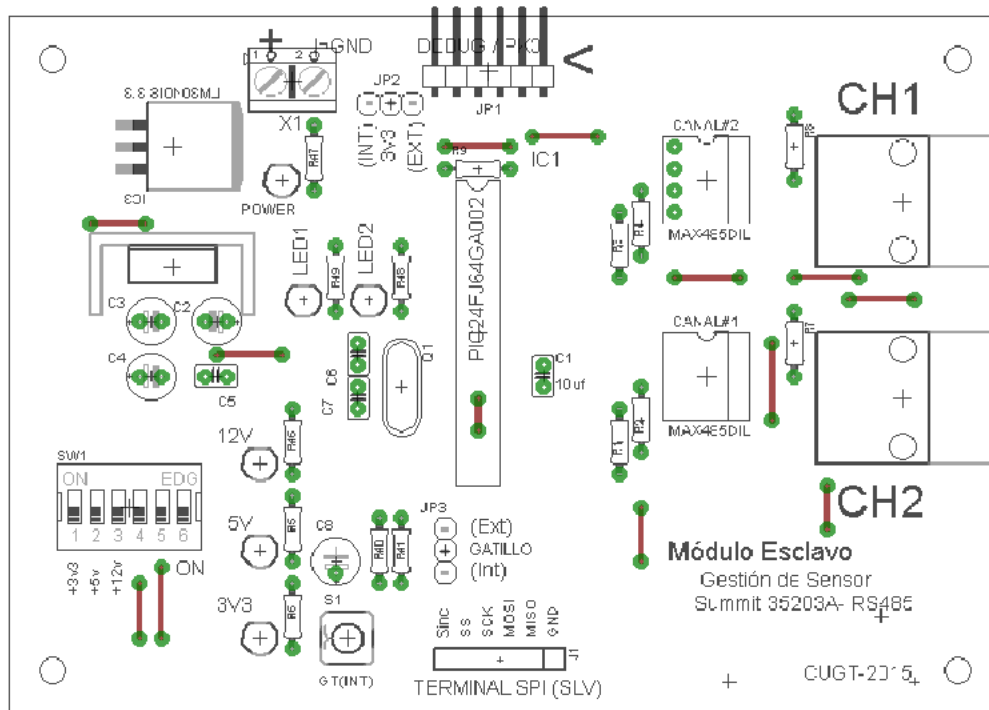


Diagrama Esquemático Módulo Esclavo



**Placa Layout Esclavo**



**Diagrama de Partes placa Maestro**