



UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERÍA

**DESARROLLO DE UN SISTEMA DE MONITOREO PARA MEDIR Y
OPTIMIZAR EL RENDIMIENTO DE LOS ATLETAS DE KAYAK
DURANTE EL ENTRENAMIENTO**

**T É S I S P R O F E S I O N A L
Q I E P A R A O B T E N E R E L T Í T U L O D E
I N G E N I E R O E L É C T R I C O E L E C T R Ó N I C O
P R E S E N T A:**

C A B A N I L L A S T E R Á N J O R G E

No. Cta. 8926698-0

DIRECTOR DE TESIS: ING. ROMÁN V. OSORIO COMPARÁN

México D.F., a (día) de junio del 2004

INDICE TEMATICO

1. Introducción	1
1.1 Justificación	
1.2 Propuesta	
1.3 Resultados esperados	
1.4 Metodología	
2. Conceptos y Características de Competencia del kayak	5
2.1 Tipos de kayak	
2.2 Explicación detallada del kayak de competencia olímpica	
2.2.1 Partes constitutivas del kayak	
2.2.2 La pala	
2.2.3 Distancias de competencia	
3. Sensores	13
3.1 Magnitudes y sensores	
3.2 Tipos de sensor y aplicación típica	
3.3 Elección de un sensor	
3.4 Sistemas de medición basados en acelerómetros	
4. Protocolos de Comunicación	21
4.1 Medios formas y tipos de transmisión	
4.2 Clasificación de los protocolos de comunicación	
4.3 Arquitectura de niveles	
4.4 Modelo OSI	
5. Diseño del Sistema	27
5.1 Planeación y conceptualización del sistema de monitoreo	
5.2 Dispositivo transmisor	
5.2.1 Cálculo de la fuerza	
5.2.2 Modulo de comunicaciones	
5.3 Dispositivo receptor	
5.4 Diseño a bloques de los dispositivos transmisor y receptor	
6. Desarrollo del Sistema	33
6.1 Microcontroladores	
6.1.1 Descripción del microcontrolador mc68hc11 de Motorola	
6.1.2 Descripción del microcontroladorpic16f84 de microchip	
6.2 Control del LCD (Liquid Crystal Display)	
6.2.1 Inicialización para 4 bits	
6.2.2 Inicialización para 8 bits	
6.2.3 Rutina de inicialización del LCD para el receptor	

6.2.4	Rutina de inicialización del LCD para el transmisor	
6.3	Acelerómetro	
6.3.1	Principio de funcionamiento	
6.3.2	Consideraciones de diseño	
6.3.3	Integración del acelerador al microcontrolador	
6.3.4	Caracterización del acelerómetro	
6.4	Módulos de transmisión y recepción	
6.4.1	Descripción y características generales	
6.4.2	Transmisión de la información	
6.4.3	Transmisor	
6.4.3.1	Principio de operación de transmisión de la interfase SCI (Serial Communication Interface)	
6.4.3.2	Registros del sistema SCI	
6.4.3.3	Programación de la SCI para transmisión	
6.4.3.4	Código fuente de la subrutina de transmisión y registros asociados a la SCI	
6.4.4	receptor	
6.4.4.1	Código fuente del algoritmo de recepción	
6.5	Integración del sistema	
6.5.1	Integración del transmisor	
6.5.1.1	Calculo de la fuerza	
6.5.1.2	Calculo de las BPM (bogas por minuto)	
6.5.1.3	Transmisión de la información	
6.5.1.4	Características y funcionamiento del transmisor	
6.5.1.5	Programación del transmisor	
6.5.1.6	Diagrama eléctrico del transmisor	
6.5.2	Integración de receptor	
6.5.2.1	Características del funcionamiento del receptor	
6.5.2.2	Programación del receptor	
6.5.2.3	Diagrama eléctrico del receptor	
7.	Pruebas y Resultados	75
7.1	Medición de las bogas por minuto y fuerza aplicada	
7.1.1	Análisis de resultados de las pruebas 1,2 y 3	
7.2	Solución a los problemas detectados	
7.3	Pruebas de medición de las BPM y fuerza aplicada después de la optimización del sistema	
7.3.1	Análisis de resultados de las pruebas 1,2 y 3 optimizadas	
7.4	Distancia máxima de recepción y pérdida de datos	
7.5	Integración del sistema a un plan de entrenamiento	
7.6	Posibles mejoras al sistema	
7.7	Otras aplicaciones	
8.	Conclusiones	93

9. Apéndices	95
9.1 Guía de usuario	
9.2 Diagramas de flujo del transmisor	
9.3 Diagramas de flujo del receptor	
9.4 Código fuente del transmisor	
9.5 Código fuente de los programas: tbajo.asm, t1.asm, t2.asm	
9.6 Circuito impreso del transmisor	
9.7 Circuito impreso del receptor	
9.8 Hojas de especificaciones de los dispositivos empleados	
10. Glosario	181
11. Referencias Bibliográficas	183
12. Bibliografía	185

1. INTRODUCCIÓN

El Canotaje es un deporte acuático que se práctica a bordo de Kayaks o Canoas, en lagos, lagunas, pistas artificiales o en cualquier espejo de agua que sea propicio para su práctica.

El Canotaje es un deporte olímpico que se divide en dos especialidades: El Kayak y la Canoa Canadiense. El Kayak es una embarcación en la que el atleta se posiciona sentado, viendo hacia el frente, e impulsa la embarcación por medio de una pala de doble cuchara, la cual tiene un ángulo de inclinación entre cucharas, que depende de, tanto la diestra del remero como de sus cualidades físicas (figura 1.1). La Canoa Canadiense es una embarcación en la que el atleta se encuentra hincado sobre una de sus rodillas y la otra pierna apoyada sobre el pie al frente de la rodilla que esta apoyada, de tal forma que la distancia entre ambas piernas permita al atleta remar con una pala de una sola cuchara, provista de un apoyo en el otro extremo, es así que existen remeros zurdos y derechos (figura 1.2). En ambas especialidades existen embarcaciones para 1, 2 y 4 tripulantes, en las categorías femenil y varonil, a las distancias de 200, 500 y 1000 metros y también se efectúan carreras de maratón 42.2 km. y 10 km.



Figura 1.1 Kayak



Figura 1.2 Canoa Canadiense

Habitualmente un entrenamiento de Canotaje, incluye la práctica de otros deportes, como lo son entre otros: la natación, el atletismo y levantamiento de pesas, además de las actividades propias de la especialidad como lo es la realización de recorridos a diferentes distancias, dependiendo de la etapa de preparación a la que se encuentre la temporada de competencias. Las distancias que generalmente incluye un plan de entrenamiento en sus diferentes etapas van desde los 50 metros hasta los 16 Km. Siendo las más importantes las distancias oficiales de competencia: 200, 500 y 1000 mts.

El entrenamiento al que se ven sometidos los atletas de Canotaje exige en todo momento ser monitoreado, para vigilar, controlar y mejorar, el desarrollo técnico, el desarrollo físico-atlético, el incremento en las capacidades físicas y el estado médico de los atletas. Todos estos factores tienen que conjugarse de manera adecuada para mejorar el desempeño competitivo de los deportistas.

1.1 JUSTIFICACIÓN

Los parámetros más frecuentes que se utilizan para realizar el monitoreo antes mencionado, además de los concernientes al aspecto médico en una sesión de entrenamiento son: El tiempo y las bogas por minuto (cantidad de paladas o remadas que se ejecutan por minuto), realizadas en las distancias correspondientes a la etapa atlética en la que se encuentra preparándose el atleta, estos parámetros son perfectamente determinados con la ayuda de un cronómetro, que entre una de sus variadas funciones permite contabilizar las bogas por minuto.

Un problema frecuente en el monitoreo del entrenamiento de estos atletas es que habitualmente se tienen estándares en tiempos y bogas por minuto que se deben de realizar en una determinada

distancia, sin embargo, es difícil para el atleta mantener un ritmo de bogas en todo momento y simultáneamente cumplir con el tiempo establecido, esto se debe a que cuando se presenta el cansancio en los atletas, es muy común el incremento del ritmo de bogas por minuto, suponiendo que se incrementa la fuerza aplicada en cada boga para tratar de mantener la velocidad del kayak y así realizar el tiempo requerido, cuando en realidad se incrementa el ritmo de las bogas porque la fuerza aplicada en cada una de ellas es cada vez menor, de manera que es difícil controlar por parte del atleta que parámetro incrementar, sin descuidar los otros factores que le permitan mejorar los tiempos.

El objetivo fundamental de un entrenamiento es reducir el tiempo empleado en recorrer una distancia, ya sea en Kayak o en Canoa, esta mejoría se obtiene aumentando la velocidad de la embarcación.

Las formas perfectamente definidas para aumentar la velocidad de un kayak son las siguientes:

- 1) Incrementando la fuerza aplicada en cada palada y manteniendo el ritmo de bogas por minuto.
- 2) Incrementando el ritmo de bogas por minuto y manteniendo la fuerza aplicada en cada palada.
- 3) Incrementando la fuerza aplicada en cada palada e incrementando el ritmo de bogas por minuto.

Cabe mencionar que la velocidad puede incrementarse combinando de otras maneras estos dos factores, de las que más adelante se hace referencia, sin embargo estas son las maneras perfectamente definidas.

En una competencia y/o entrenamiento, estas 3 maneras de incrementar la velocidad se presentan en momentos clave y estratégicos del recorrido, sin embargo, también se presentan momentos en los que la disminución de la velocidad es ocasionada por el cansancio físico del atleta, que provoca que la combinación de los factores: Bogas por minuto y fuerza, tengan una repercusión negativa en la velocidad del bote.

Las causas de la disminución de la velocidad del bote no necesariamente se presentan cuando hay una disminución en la fuerza aplicada o cuando disminuye el ritmo de bogas por minuto, ya que esta disminución en uno de estos factores puede ser compensada por el incremento del factor complementario, de tal manera que la velocidad del bote puede mantenerse e incluso aumentar, cuando la combinación en el incremento y/o disminución de estos dos factores, así lo consiguen.

Por lo tanto la única manera definida, que ocasiona una disminución en la velocidad del bote es:

- 1) Disminución de la fuerza aplicada en cada palada y disminución de las bogas por minuto.

Las maneras que no están perfectamente definidas en el aumento de la velocidad son las siguientes:

- 1) Disminución de la fuerza aplicada en cada palada y complementariamente, el aumento en las bogas por minuto, sobrecompensa la disminución de la fuerza.
- 2) Disminución de las bogas por minuto y complementariamente, el aumento en la fuerza aplicada en cada palada, sobrecompensa la disminución de las bogas.

Las maneras que no están perfectamente definidas en la disminución de la velocidad son:

- 1) Disminución de la fuerza aplicada en cada palada y complementariamente, el aumento en las bogas por minuto, no compensa la disminución de la fuerza.
- 2) Disminución de las bogas por minuto y complementariamente, el aumento en la fuerza aplicada en cada palada no compensa la disminución de bogas.

Finalmente el mantener una velocidad constante se consigue de la siguiente manera:

- 1) Aumento/disminución de la fuerza en cada palada es compensada con la disminución/aumento de las bogas por minuto.

El problema que claramente se aprecia, es que no se conoce que tanto se debe de compensar o sobrecompensar un factor con el otro para lograr las mejoras necesarias en el desenvolvimiento de una competencia y en la manera en como se tienen que ajustar los planes de entrenamiento para lograr una combinación adecuada de estos factores. Las únicas herramientas que se tienen actualmente para ubicarnos alrededor de estos factores son la capacidad de medir el tiempo, las bogas por minuto y la combinación de estos dos parámetros con valores empíricos, que deben de efectuarse en determinadas distancias en una competencia y/o entrenamiento.

1.2 PROPUESTA

De este análisis surge la necesidad de medir la fuerza aplicada en cada palada para poder controlar su magnitud, como un parámetro decisivo en el entrenamiento de la práctica del Canotaje.

Se propone la implementación de un prototipo portátil para instalarse en el kayak sin afectar la estructura del mismo, basado en microcontroladores y un circuito de tecnología de punta que permita medir la aceleración para posteriormente realizar cálculo de la fuerza. Dicho dispositivo contará con una pantalla LCD que permita al atleta visualizar en todo momento el valor de la fuerza aplicada, además de que la información obtenida será transmitida inalámbricamente hacia un dispositivo receptor para ser visualizada por el entrenador, ya que el atleta al realizar grandes esfuerzos debe concentrar toda su atención en la ejecución de sus movimientos y es el entrenador el encargado de alentar al atleta a mantener el ritmo de trabajo.

El desarrollo de este sistema engloba la teoría de microcontroladores, protocolos de comunicación y tratamiento digital de señales. Los dispositivos propuestos para este fin son los microcontroladores MC68HC11E2 de Motorola y el PIC16F84 de Microchip, el acelerómetro ADXL202JQC1 de Analog Devices y el módulo de comunicaciones RWS440 y TWS440 de Reynolds Electronics.

1.3 RESULTADOS ESPERADOS

Con el desarrollo de este sistema, se espera que los resultados de las mediciones obtenidas, sean utilizados como un parámetro más, para el análisis de los planes de entrenamiento y de esta forma se optimicen los criterios empleados en la metodología del desarrollo de dichos planes.

El entrenador y el atleta deben recibir la información simultáneamente para poder estar en perfecta coordinación en la ejecución del entrenamiento e impulsar y alentar al atleta a mejorar y mantener de una manera mas controlada el ritmo de trabajo en el entrenamiento.

Los resultados obtenidos no necesariamente deben estar dentro de la escala real de fuerza, si no simplemente dentro de una escala numérica que permita el análisis comparativo entre las diferentes mediciones obtenidas.

1.4 METODOLOGÍA

Se empleará una metodología experimental, derivada de las características de los circuitos electrónicos empleados ya que así lo requieren, para poder adaptarlos e integrarlos en los dispositivos transmisor y receptor de la mejor manera, con el fin de que al realizar las pruebas sobre la embarcación se ajusten tan solo algunos parámetros de la programación empleada en cada uno de los microcontroladores de cada dispositivo.

2. CONCEPTOS Y CARACTERÍSTICAS DE COMPETENCIA DEL KAYAK

El kayak es una embarcación que tiene sus orígenes en los pueblos esquimales y originalmente fue utilizada para transportarse y transportar víveres y demás pertrechos útiles para la vida en la región, así como para la pesca. Dicha embarcación estaba construida con una estructura de madera recubierta con piel de foca, completamente hermética inclusive en la parte por donde el tripulante se sitúa para remar previniendo de esta forma que el agua entrara en el bote (figura 2.1).



Figura 2.1 Kayak esquimal

El kayak fue evolucionando en forma y tamaño con el empleo de nuevos materiales y técnicas de fabricación así como por el nuevo uso que se hizo de ellos. Actualmente podemos encontrar una gran variedad de kayaks dependiendo de la actividad para la cual fueron fabricados, la característica fundamental en todos los kayaks es que el remero se impulsa con una pala de doble cuchara que en sus inicios fue de madera y de cucharas planas, en la actualidad se construyen con fibra de carbón y de cucharas cóncavas.

2.1 TIPOS DE KAYAK

Existen muchos tipos de kayak que son utilizados para diferentes actividades deportivas y recreativas, por lo que tanto su forma y material de construcción son diferentes en muchos de los casos. Las variantes más importantes de kayak se describen a continuación, de acuerdo a su uso.

Kayak de turismo.- Son kayaks destinados a la actividad recreativa en aguas tranquilas, no tienen timón y muchas veces se utilizan para enseñar a los principiantes a remar y corregir la técnica, generalmente son para una persona y comúnmente están contruidos de fibra de vidrio (figura 2.2).



Figura 2.2 Kayak de turismo

Kayak para descenso de río.- Este tipo de embarcación se usa para las llamadas aguas bravas (ríos con grandes corrientes y caídas pronunciadas), no tienen timón y el tripulante rema con una falda generalmente hecha de neopreno que se usa para sellar herméticamente el bote para que el agua no se introduzca en el, ya que esta embarcación se ve sometida a constantes inmersiones y volcaduras. Actualmente pueden estar contruidos de fibra de vidrio, fibra de carbón, plásticos, kevlar o la combinación de estos, además de los herrajes correspondientes a los apoyos de los pies que generalmente son de aluminio (figura 2.3).



Figura 2.3 Kayak de descenso de río

Kayak de mar.- Este bote como lo indica su nombre se usa para navegar en el mar, pero principalmente para deslizarse sobre las olas de mar como las tablas de surf, este bote tampoco cuenta con timón y el tripulante va sentado encima del bote no dentro de el, estos botes pueden ser de un tripulante o dos y habitualmente son de plástico (figura 2.4).



Figura 2.4 Kayak de mar

Kayak de competencia olímpica.- Este es el tipo de Kayak, motivo de este trabajo de tesis, por lo cual se dará una explicación mas detallada (figura 2.5).



Figura 2.5 Kayak Olímpico

2.2 EXPLICACIÓN DETALLADA DEL KAYAK DE COMPETENCIA OLÍMPICA

Los primeros kayaks de velocidad fueron diseñados empíricamente, simplemente con la idea de cortar el agua para avanzar mas rápidamente y que fueran ligeros, el material por excelencia y manejo fue la madera, pero paulatinamente, se fueron perfeccionando los botes y se empezó a usar la fibra de vidrio para su construcción, paralelamente se fabricaban botes de madera, siendo este material de la preferencia de los remeros por su rigidez y durabilidad, sin embargo el avance tecnológico en el campo de los materiales sintéticos y epóxicos, permitieron realizar diseños en los cuales el análisis hidrodinámico del bote demandaba, esfuerzos que solo materiales de este tipo como la fibra de carbón, kevlar, resinas polyester y epóxicas soportaban, además de no sufrir significativas deformaciones y degradaciones progresivas por el efecto del calentamiento por la luz solar y/o contacto con agua de mar o aguas duras, estos materiales ofrecen una rigidez excelente, son ligeros y permiten hacer diseños que con otros materiales sería prácticamente imposible. Actualmente la fabricación y diseño de kayaks es una obra de alta ingeniería, ya que diferentes disciplinas de la ingeniería intervienen en la construcción de estos; el análisis estructural, diseño hidrodinámico, mecánica de materiales y la estética, por mencionar algunas, se conjugan para obtener kayaks que sean capaces de romper las barreras físicas del cuerpo humano para ayudar a los atletas a mejorar sus marcas.

2.2.1 PARTES CONSTITUTIVAS DEL KAYAK

Por reglamento un kayak debe de medir 5.20 m. de largo por 0.52 m. de ancho, la medida de la altura es libre y el peso mínimo que deben de tener estas embarcaciones para una competencia oficial debe de ser de 12 Kg.

La figura 2.6, muestra las partes que constituyen un kayak.

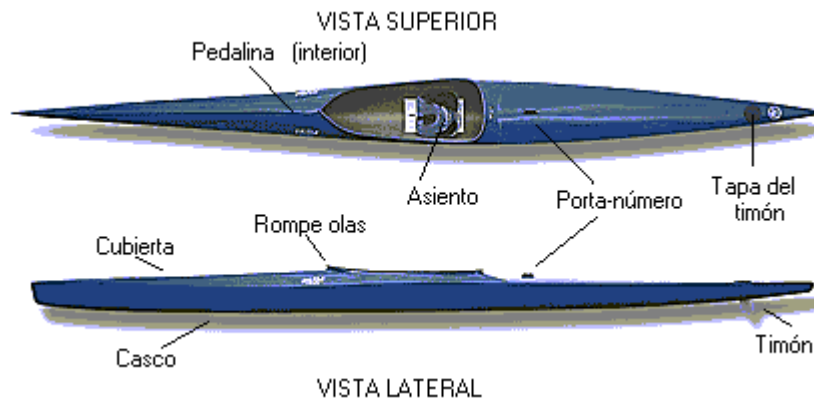


Figura 2.6 Partes constitutivas de un kayak

Casco.- Es la parte que está en contacto directo con el agua, su diseño es el más importante ya que de este depende el adecuado y óptimo desplazamiento del bote sobre el agua, el análisis hidrodinámico juega un papel primordial en el diseño y construcción del casco, incluso muchos fabricantes diseñan cascos para distancias específicas y distintos diseños para los diferentes rangos de peso de los tripulantes.

Cubierta.- Como su nombre lo indica, es la parte que cubre al casco en su totalidad excepto por el rompeolas que es por donde entra el remero, su función es la de proporcionar rigidez al casco además de proteger contra las embestidas del agua evitando que se meta agua en el bote, su diseño incluye un análisis tanto hidrodinámico como aerodinámico, puesto que estos dos elementos coaccionan con esta parte del kayak.

Rompeolas.- Muchas veces llamada bañera, esta es una pieza que tiene una doble función, desplaza las olas que pudieran llegar a esta parte del bote evitando que se llene de agua y además provee al remero de un sostén para que este coloque una falda (cubre-bañera) en toda la superficie descubierta del rompeolas para evitar que se meta el agua y que la fuerza del viento afecte el desplazamiento del bote, esta falda protectora está hecha con materiales ligeros e impermeables y generalmente se usa en competencias donde el viento es muy fuerte, en botes de equipo, días fríos y en aguas con mucho oleaje (figura 2.7).



Figura 2.7 Rompeolas

Timón.- Sirve para dar dirección al bote y es controlado con sistema de cables que van desde la pedalina hasta el timón. Está construido generalmente con plásticos rígidos con un alma metálica que le da soporte al timón y a los cables para su comando.

Tapa del timón.- Es una tapa que está en la parte trasera del bote y sirve para tener un acceso rápido y fácil al timón, para posibles reparaciones.

Pedalina.- Puede estar hecha con diferentes materiales como la madera, fibra de vidrio, kevlar o fibra de carbón, con soportes de aluminio. La pedalina es de vital importancia para el desenvolvimiento y correcta aplicación de la fuerza en cada palada ya que en esta parte del kayak el atleta coloca los pies (generalmente descalzo), para con estos maniobrar el timón además de que en conjunción con el asiento formar un robusto y firme punto de apoyo para los movimientos del cuerpo al momento de remar. La pedalina puede moverse hacia atrás o adelante según convenga al remero dependiendo del largo de sus piernas (figura 2.8)



Figura 2.8 Pedalina

Asiento.- Además de proveer al tripulante de la comodidad necesaria para una correcta ejecución de sus movimientos, como ya se menciono, el asiento forma parte del apoyo necesario para que la fuerza aplicada en el agua, tenga un punto de apoyo, ya que el atleta en cada palada tiene que patear en dirección horizontal para equilibrar el golpe con el agua que se produce en cada palada.



Figura 2.9 Asiento

El asiento puede moverse hacia arriba, abajo, adelante y atrás dependiendo de las características físicas del tripulante (figura 2.9).

Porta-número.- Es una especie de ceja metálica, que sirve para colocar el número del carril asignado al participante (figura 2.10).



Figura 2.10 Porta-número

2.3 LA PALA

La pala es el aditamento para impulsar el bote y al igual que el kayak, a sufrido varias modificaciones al pasar de los años y al tener mejores herramientas de análisis para su diseño. En un principio la pala simplemente era una pértiga con una cuchara en cada extremo de madera, totalmente simétrica, es decir no existía un ángulo entre las cucharas.

Pero se descubrió que se podía aumentar la velocidad de paladas y la fuerza aplicada en ella, si el ángulo entre cucharas se incrementaba, así es que en la actualidad el ángulo promedio entre cucharas es de 70°, varía dependiendo las cualidades del remero.

También la forma de las cucharas ha evolucionado pasando de ser de una forma plana y cuadrada a una forma curva por el extremo del ataque en el agua y cóncava en la parte opuesta a la del ataque, el primer modelo de esta pala que se estandarizo fue el modelo "Wing" que como su nombre lo indica asemeja un ala, posteriormente surgió un nuevo modelo llamada "Rasmussen" que además de tener la forma de ala esta torcida a lo largo de su eje longitudinal. Estos avances en las palas permiten que el remero tenga un punto de apoyo en el agua más firme, ya que la idea primordial de estos modelos es que la pala al entrar al agua quede fija en ese punto y sea el bote el que consiga el mayor desplazamiento (figura 2.11).



Figura 2.11 Pala y cucharas actuales hechas de fibra de carbón

2.4 DISTANCIAS DE COMPETENCIA

En el Canotaje existen tres distancias oficiales para campeonatos mundiales que son: 1000,500 y 200 mts. en las dos especialidades , Kayak y Canoa Canadiense, en las ramas varonil y femenil, en las categorías Senior (mayor de 18 años) y Juvenil (hasta 18 años de edad), para todas las embarcaciones que son: individual, doble y cuádruple, con alineación de hasta de 9 competidores, por carrera.

En los Juegos olímpicos la distancia de 200 mts., no se corre en ninguna de sus variantes, el kayak cuádruple varonil y la canoa cuádruple varonil solo corren a 1000 mts y el femenil solo a 500 mts., no existe limite de edad.

Todos estos eventos se llevan a cabo en lagos o pistas artificiales, donde preferentemente no predominen fuertes vientos ni oleajes intensos.

2. Conceptos y características de competencia del kayak

Existen también competencias de 10 Km. varonil y 5Km femenil, que desde el año de 1994 dejó de ser distancia oficial para los campeonatos mundiales, pero aún se sigue efectuando en algunas competencias amistosas.

Por último, cabe mencionar que también existe el mundial de maratón (42.2 Km.) en kayak individual para categoría abierta, en las ramas varonil y femenil, el cual es muy espectacular, esta competencia se hace en ríos de aguas tranquilas o en grandes lagos, en donde se hace linealmente o en un circuito de vueltas.

Existen otras competencias a diferentes distancias que se celebran como promoción y fomento del deporte y en algunos países, como en España, que se celebra una carrera de 20 Km., en el río Sella que es toda una fiesta nacional y una competencia de gran tradición internacional.

3. SENSORES

Actualmente las aplicaciones de la electrónica, están presentes en innumerables aspectos de nuestra vida cotidiana, muchas de las cuales no serían posibles sin los sensores. La utilización de sensores es indispensable en los procesos de automatización utilizados en las industrias de manufactura, robótica, ingeniería experimental, ahorro energético, control ambiental (aire, ruido, calidad del agua), automóviles, electrodomésticos, medicina, etc.

En un sistema, que es la combinación de dos o más elementos, subconjuntos y partes necesarias para realizar una o varias funciones, es necesaria la vigilancia y seguimiento de procesos como puede ser la medición y/o control de distintas magnitudes tales como: la temperatura ambiente, los niveles de gas y de agua, las fuerzas ejercidas sobre un conductor simulado a bordo de un vehículo cuando este choca contra un objeto, por mencionar algunos.

En un sentido amplio la realización de una medida implica, pues, además de la adquisición de la información realizada por un elemento sensor, también el procesamiento de dicha información y la presentación de resultados que puedan ser percibidos por nuestros sentidos.

Un sensor es un dispositivo que a partir de la energía del medio donde se mide, proporciona una señal de salida traducible que es función de la variable medida, en general a todo dispositivo que convierte una señal de una forma física en una señal correspondiente pero de otra forma física distinta se le denomina, sensor.

El proceso de transducción siempre extrae cierta energía del sistema donde se mide, por lo que es importante garantizar que éste no lo perturbe, ya que de ser así probablemente el sistema quedaría imposibilitado para realizar su función y dependiendo de la cantidad de energía extraída se podrían obtener mediciones erróneas.

Dado que existen diferentes tipos de magnitudes: mecánicas, térmicas, magnéticas, eléctricas, ópticas, moleculares (químicas), se han diseñado diferentes tipos de sensores para transducir cada una de dichas magnitudes, en otras que sean útiles para el posterior tratamiento de la información que el sensor proporciona, para así realizar mediciones en las diferentes aplicaciones que así lo requieran.

Desde el punto de vista de la Ingeniería Electrónica, es más atractiva la clasificación de los sensores de acuerdo con el parámetro variable: resistencia, capacitancia, inductancia, etc..., para que la señal obtenida pueda ser manipulada por medios electrónicos.

Los sistemas de medida electrónicos ofrecen entre otras, las siguientes ventajas:

1. Debido a la estructura electrónica de la materia, cualquier variación de un parámetro no eléctrico de un material, viene acompañada por la variación de un parámetro eléctrico. Eligiendo el material adecuado, esto permite realizar sensores con salida eléctrica para cualquier magnitud física de naturaleza no eléctrica.
2. Dado que en el proceso de medición no conviene extraer energía del sistema que se mide, lo mejor es amplificar la señal de salida del sensor. Con amplificadores electrónicos se pueden obtener fácilmente altas ganancias de potencia en una sola etapa, a baja frecuencia.
3. Además de la amplificación, hay una gran variedad de recursos en forma de circuitos integrados, para acondicionar o modificar las señales eléctricas, incluso hay sensores que incorporan físicamente en un encapsulado parte de estos recursos.

3. Sensores

4. Existen también numerosos recursos para presentar o registrar información si se hace electrónicamente, permitiendo manejar no solo datos numéricos, sino también textos gráficos y diagramas.
5. La transmisión de señales eléctricas es más versátil que la de señales mecánicas, hidráulicas o neumáticas, y si bien no hay que olvidar que éstas pueden ser mas convenientes en ciertas circunstancias, como pueden ser la presencia de radiaciones ionizantes o atmósferas explosivas, en muchos casos estos sistemas han sido sustituidos por otros eléctricos. De hecho, mientras en industrias de proceso (química, petróleo, gas, alimentación, textil, etc.), en donde la automatización de procesos se expandió rápidamente, se encuentran actualmente sistemas neumáticos junto a sistemas electrónicos más recientes, en cambio en las industrias de manufacturados donde hay una serie de procesos discontinuos y que son de automatización mas reciente, apenas existen sistemas neumáticos.

3.1 MAGNITUDES Y SENSORES.

La tabla 3.1 muestra los diferentes tipos de sensores y métodos de detección ordinarios para las magnitudes mas frecuentes. ^[3.1]

Tabla 3.1 Tipos de sensores y métodos de detección

SENSORES	MAGNITUDES								
	Posición, Distancia, Desplazamiento	Velocidad	Aceleración Vibración	Temperatura	Presión	Caudal Flujo	Nivel	Fuerza	Humedad
Resistivos	Potenciómetros, Galgas, Magnetoresistencias		Galgas + masa-resorte	RTD Termistores	Potenciómetros + tubo Bordon	Anemómetros de hilo caliente Galgas + voladizo Termistores	Potenciómetro + flotador Termistores LDR	Galgas	Humistor
Capacitivos	Condensador diferencial				Condensador variable+ diafragma		Condensador Variable	Galgas capacitivas	Dieléctrico variable
Inductivos y electro-magnéticos	LVDT Corrientes Foucault Efecto Hall	Ley Faraday LVT Efecto Hall Corrientes Foucault	LVDT + masa – resorte		LVDT + diafragma Reluctancia variable + diafragma	LVDT + rotámetro LeyFaraday	LVDT + flotador Corrientes de Foucault	Magneto-elástico LVDT + célula de carga	
Generadores			Piezoeléctricos + masa – resorte	Termopares piroeléctricos	Piezo-eléctrico			Piezo-eléctrico	
Digitales	Codificadores incrementales y absolutos	Codificadores incrementales		Osciladores de cuarzo	Codificador + Tubo Bourdon	Vórtices			SAW (Surface Acoustic Wave)
Uniones P-N	Fotoeléctricos			Diodo, Transistor			Fotoeléctricos		
Ultrasónicos	Reflexión	Efecto Doppler				Efecto Doppler Tiempo-tránsito Vórtices	Reflexión Absorción		

3.2 TIPO DE SENSOR Y APLICACIÓN TÍPICA

Las siguientes tablas (tablas 3.2,3.3,3.4,3.5), engloban los diferentes tipos de sensores de acuerdo a su aplicación típica y principio de operación. ^[3.2]

Tabla 3.2 Sensores resistivos

Resistencia		
Parámetro Eléctrico y Clase de Transductor	Principio de Operación y Naturaleza del Dispositivo	Aplicación Típica
Potenciómetro	El posicionamiento de un cursor por medio de una fuerza externa varía la resistencia en el dispositivo o en un circuito puente	Presión, desplazamiento.
Gálga extensiométrica resistiva	La resistencia de un material metálico o semiconductor cambia según la elongación o compresión debida a esfuerzos aplicados externamente	Fuerza, par, desplazamiento.
Medidor de alambre caliente o medidor Pirani	La resistencia de un elemento caliente varía enfriándolo con flujo de gas.	Flujo de gas, presión de gas.
Termómetro de resistencia	La resistencia de un alambre de metal puro, con un coeficiente de temperatura de resistencia positivo grande, varía con la temperatura.	Temperatura, calor radiante.
Termistor	La resistencia de ciertos óxidos de metal con coeficiente de temperatura de resistencia negativo cambia con la temperatura	Temperatura, calor radiante.
Higrómetro de resistencia	La resistencia de una cinta conductiva se altera con el contenido de humedad	Humedad relativa
Celda fotoconductiva	La resistencia de una celda como un elemento del circuito se modifica con la luz incidente	Relevador fotosensible

Tabla 3.3 Sensores Inductivos

Inductancia		
Parámetro Eléctrico y Clase de Transductor	Principio de Operación y Naturaleza del Dispositivo	Aplicación Típica
Transductor de circuito magnético	Los cambios del circuito magnético modifican la autoinductancia o inductancia mutua de una bobina excitada por C.A.	Presión, desplazamiento
Detector de reluctancia	La reluctancia de un circuito magnético varía al cambiar la posición del núcleo de hierro de una bobina	Presión, desplazamiento, vibración, posición
Transformador diferencial	El voltaje diferencial de dos devanados secundarios de un transformador varía al mover el núcleo magnético por medio de una fuerza exterior	Presión, fuerza, desplazamiento, posición
Medidor de corriente parásita	La inductancia de una bobina se altera por la proximidad de una placa con corrientes parásitas inducidas	Desplazamiento, espesor
Medidor de magnetorestricción	Las propiedades magnéticas cambian por presión y esfuerzos	Fuerza, presión, sonido

Tabla 3.4 Sensores Capacitivos

Capacitancia		
Parámetro Eléctrico y Clase de Transductor	Principio de Operación y Naturaleza del Dispositivo	Aplicación Típica
Medidor de presión de capacitancia variable	Una fuerza aplicada externamente varía la distancia entre dos placas paralelas	Desplazamiento, presión.
Micrófono capacitor	La presión del sonido altera la capacitancia entre una placa fija y un diafragma móvil	Voz, música, ruido.
Medidor dieléctrico	La capacitancia varía por cambios en el dieléctrico	Nivel de líquidos, espesor

Tabla 3.5 Sensores de Voltaje y Corriente

Voltaje y corriente		
Parámetro Eléctrico y Clase de Transductor	Principio de Operación y Naturaleza del Dispositivo	Aplicación Típica
Detector por efecto de Hall	Se genera una diferencia de potencial a través de una placa semiconductora (de Germanio) cuando un flujo magnético interactúa con una corriente aplicada	Flujo magnético, corriente
Cámara de Ionización	Se induce flujo de electrones mediante la ionización de un gas debido a radiación radiactiva	Conteo de partículas, radiación
Celda fotoemisiva	Hay una emisión de electrones debido a la radiación incidente en una superficie fotoemisiva	Luz y radiación
Tubo fotomultiplicador	La emisión de electrones secundarios es debido a la radiación incidente sobre un cátodo fotosensible	Luz y radiación, relevadores fotosensibles
Termopar y termopila	Se genera una Fem por la unión de dos metales disímiles o semiconductores cuando la unión se calienta	Temperatura, flujo de calor, radiación
Generador de bobina móvil	El movimiento de una bobina en un campo magnético genera un voltaje	Velocidad, vibración
Detector piezoeléctrico	Se genera una fem cuando una fuerza externa se aplica a ciertos materiales cristalinos como el cuarzo	Sonido, vibración, aceleración, cambios de presión
Celda fotovoltaica	Se genera voltaje en un dispositivo de unión semiconductora cuando la energía radiante estimula la celda	Medidor de luz, celda solar

3.3 ELECCION DE UN SENSOR

Para la elección de un sensor ^[3.3], además de tener en cuenta la aplicación para la cual se requiere, es necesaria la consideración de los siguientes factores:

Magnitud a medir

- Margen de medida
- Resolución
- Exactitud deseada
- Estabilidad
- Ancho de banda
- Tiempo de respuesta
- Límites absolutos posibles de la magnitud a medir
- Diferentes Magnitudes

Características de salida

- Sensibilidad
- Tipo: tensión, corriente, frecuencia.
- Forma de la señal: unipolar, flotante, diferencial.
- Impedancia
- Destino: presentación analógica, conversión digital, teledada.

Características de alimentación

- Tensión
- Corriente
- Potencia disponible
- Frecuencia
- Estabilidad

Características ambientales

- Margen de temperaturas
- Humedad
- Vibraciones
- Agentes químicos
- Entorno electromagnético

Otros factores

- Peso, dimensiones, vida media, coste de adquisición, disponibilidad, tiempo de instalación, situación en caso de fallo, coste de verificación, coste de mantenimiento, coste de sustitución.

3.4 SISTEMAS DE MEDICIÓN BASADOS EN ACELERÓMETROS

Para la medición de cantidades como lo son, la velocidad, el desplazamiento y la aceleración, en algunos sistemas se emplean acelerómetros. El principio de funcionamiento de este tipo de dispositivos se basa en el efecto que produce la acción de la aceleración sobre una masa, dando como resultado una fuerza, correspondiente a la ecuación $F=ma$ (segunda ley de Newton)^[3.4], y con el adecuado tratamiento de la señal obtenida se pueden obtener las mediciones antes mencionadas.

La masa que es usada como parte de un dispositivo medidor de aceleración, está soportada en un plano ya sea vertical, horizontal o en ambos ejes. Para los desplazamientos en el plano vertical la masa además de la aceleración experimentada por el movimiento en ese eje debe de ser tomada en cuenta el efecto de la gravedad para el diseño de tales dispositivos.

Un acelerómetro sencillo^[3.5] sería colocar una masa sobre ruedas que se desplace sobre un eje horizontal, sujeta a un resorte y medir las variaciones de desplazamiento, tomando en consideración las características del resorte, masa y fricción.

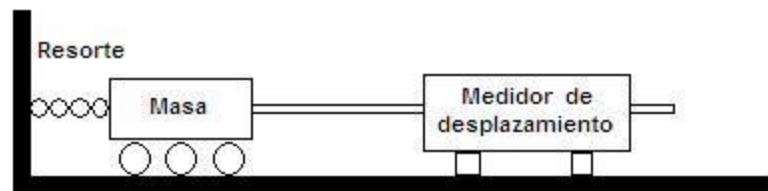


Figura 3.1 Acelerómetro sencillo

Actualmente los circuitos integrados, han sustituido a los antiguos sistemas de medición de aceleración que comúnmente eran de naturaleza mecánica, gran cantidad de dichos circuitos basan su funcionamiento en el principio capacitivo^[3.6] (figura 3.2), o en el efecto piezoeléctrico^[3.7] (figura 3.3).

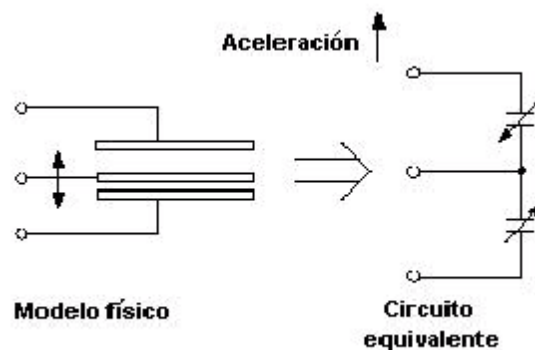


Figura 3.2 Acelerómetro capacitivo

Los acelerómetros capacitivos, basan su funcionamiento en el efecto capacitivo que se presenta entre tres placas formando dos capacitores variables. Dos de las placas, se encuentran fijas y la restante en medio de las dos sostenida en alguno de sus extremos, de tal manera que al desplazar el dispositivo en la dirección del eje diseñado, la placa intermedia se mueve en sentido contrario al del desplazamiento, con la consecuente variación de la capacitancia entre placas.

Usualmente el cambio de capacitancia se mide con un circuito oscilador el cual refleja las variaciones de capacitancia en variaciones proporcionales de frecuencia, con este principio se elabora la electrónica necesaria dentro del mismo integrado, para proporcionar una salida que sea electrónicamente útil para el posterior tratamiento e integración del dispositivo a un sistema de medición. Los acelerómetros capacitivos tienen una excelente respuesta en frecuencia y pueden medir fenómenos estáticos como dinámicos. Sus desventajas son sensibilidad a variaciones de temperatura y la posibilidad de que se presenten señales erráticas o distorsionadas debido a terminales de gran longitud, a menudo se requiere de otro oscilador de frecuencia fija para propósitos heterodínos.

Los acelerómetros piezoeléctricos (figura 3.3), contruidos con materiales cristalinos asimétricos, como el cuarzo, la sal de Rochelle y el titanio de Bario, producen una fem cuando están expuestos a un esfuerzo.

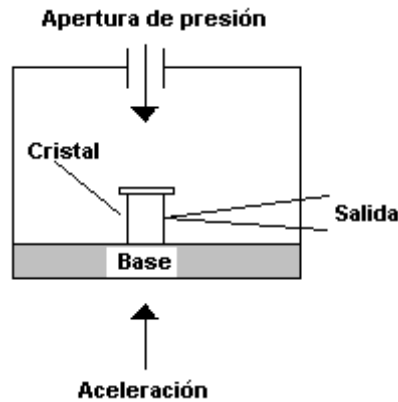


Figura 3.3 Acelerómetro piezoeléctrico

El funcionamiento básico es colocar el cristal en una base sólida. Una fuerza aplicada desde el exterior entra en el transductor a través de su apertura de presión aplicando presión a la parte superior del cristal. Esta produce una fem a través del cristal proporcional a la magnitud de la presión aplicada, el voltaje generado es proporcional a la aceleración aplicada. Este tipo de transductores tienen una muy buena respuesta en alta frecuencia, típicamente los voltajes de salida son de 1 a 30 mV por gravedad. El dispositivo no necesita fuente de potencia externa por consiguiente es del tipo de autogeneración. La desventaja principal de éste es que no puede medir condiciones estáticas, las variaciones de temperatura del cristal también afectan el voltaje de salida.

El acelerómetro empleado en el diseño de este trabajo de tesis es el ADXL202JQC1 de "Analog Devices" el cual es descrito ampliamente en la fase correspondiente a la integración del sistema.

4. PROTOCOLOS DE COMUNICACION

Un protocolo de comunicación es un conjunto de reglas que definen la manera en que se conduce la comunicación entre dos o más partes. Define cómo se inician, efectúan y terminan las comunicaciones, así como el tipo de datos que se transfieren.

Independientemente del medio de transmisión que se utilice para establecer el enlace, los sistemas de comunicación de datos necesariamente deben tener un protocolo de comunicación.

Para la comunicación de datos se requieren cuatro elementos básicos que son:

- Emisor: Dispositivo que transmite los datos
- Mensaje: Lo conforman los datos a ser transmitidos
- Medio : Es la parte física por donde viaja la información (atmósfera, cable).
- Receptor: Dispositivo de destino de los datos

Las funciones más importantes de un protocolo de comunicación, además del intercambio de información son:

- Control de Errores: Protege integridad de los datos del usuario y de los mensajes de control.
- Control de Flujo y Congestión: Permite a la red compartir sus recursos entre un gran número de usuarios, entregando a cada uno un servicio satisfactorio sin que sus operaciones corran peligro.
- Estrategias de Encaminamiento: Permite optimizar la utilización de los recursos de la red, aumentando la disponibilidad de los servicios de la red al proveer caminos alternativos entre nodos terminales.

La terminología empleada en la comunicación de datos incluye los siguientes conceptos:

- BIT: Es la unidad más pequeña de información y la unidad base en comunicaciones.
- BYTE: Conjunto de bits continuos mínimos que hacen posible, un direccionamiento de información en un sistema computarizado. Está formado por 8 bits.
- TRAMA : Tira de bits con un formato predefinido usado en protocolos orientados a bit.
- PAQUETE : Fracciones de un mensaje de tamaño predefinido, donde cada fracción o paquete contiene información de procedencia y de destino, así como información requerida para el reensamblado del mensaje.
- INTERFACES: Conexión que permite la comunicación entre dos o más dispositivos.
- CÓDIGOS: Acuerdo previo sobre un conjunto de significados que definen una serie de símbolos y caracteres. Toda combinación de bits representa un carácter dentro de la tabla de códigos, las tablas de códigos más reconocidas son las del código ASCII y la del código EBCDIC.
- PARIDAD: Técnica que consiste en la adición de un bit a un carácter o a un bloque de caracteres para forzar al conjunto de unos (1) a ser par o impar. Se utiliza para el chequeo de errores en la validación de los datos. El bit de paridad será cero (0=SPACE) o uno (1=MARK).
- MODULACIÓN: Proceso de manipular de manera controlada las propiedades de una señal portadora para que contenga la información que se va a transmitir
- Usualmente el rol que toma un equipo de comunicaciones puede ser:

- DTE (Data Terminal Equipment): Equipos que son la fuente y destino de los datos, comprenden equipos de computación (Host, Microcomputadores y Terminales).
- DCE (Data Communications Equipment): Equipos de conversión entre el DTE y el canal de transmisión, es decir, los equipos a través de los cuales conectamos los DTE a las líneas de comunicación.

4.1 MEDIOS, FORMAS Y TIPOS DE TRANSMISION

Medios

- Aéreos: Basados en ondas electromagnéticas, utilizan la atmósfera como medio de transmisión.
- Sólidos: Principalmente el cobre, en par trenzado o cable coaxial y la fibra óptica.

Formas

- Transmisión en Serie: Los bits se transmiten de uno a uno sobre una línea única. Se utiliza para transmitir a larga distancia.
- Transmisión en Paralelo: Los bits se transmiten en grupo sobre varias líneas al mismo tiempo. La transmisión en paralelo es más rápida que la transmisión en serie pero en la medida que la distancia entre equipos se incrementa, además del costo adicional de los cables aumenta la complejidad de los transmisores y los receptores de la línea a causa de la dificultad de transmitir y recibir señales de pulsos a través de cables largos.

Tipos

- Transmisión Simplex: La transmisión de datos se produce en un solo sentido, siempre existe un nodo emisor y un nodo receptor los cuales no cambian sus funciones.
- Transmisión Half-Duplex: La transmisión de los datos se produce en ambos sentidos pero alternativamente, en un solo sentido a la vez. Si se está recibiendo datos no se puede transmitir.
- Transmisión Full-Duplex: La transmisión de los datos se produce en ambos sentidos al mismo tiempo, un extremo que esta recibiendo datos puede, al mismo tiempo, estar transmitiendo otros datos.
- Transmisión Asíncrona: Cada byte de datos incluye señales de inicio y parada al principio y al final. La misión de estas señales consiste en avisar al receptor de que está llegando un dato y darle suficiente tiempo al receptor de realizar funciones de sincronismo antes de que llegue el siguiente byte.
- Transmisión Síncrona: se utilizan canales separados de reloj que administran la recepción y transmisión de los datos. Al inicio de cada transmisión se emplean unas señales preliminares llamadas Bytes de sincronización en los protocolos orientados a byte y Flags en los protocolos orientados a bit su misión principal es alertar al receptor de la llegada de los datos.
- Nota: Las señales de reloj determinan la velocidad a la cual se transmite o recibe.

4.2 CLASIFICACIÓN DE LOS PROTOCOLOS DE COMUNICACIÓN

De acuerdo a su estructura

- Protocolos Orientados a Bit: son aquellos protocolos en los cuales los bits por si solos pueden proveer información, son protocolos muy eficientes y trabajan en tramas de longitud variable.
- Protocolos Orientados a Byte: son aquellos en los que la información viene provista por la conjunción de bytes de información y bytes de control.

De acuerdo a su disciplina de comportamiento

- Protocolos de Sondeo Seleccíon: son aquellos que utilizan un DTE como nodo principal de canal. Este nodo primario controla todas las demás estaciones y determina si los dispositivos pueden comunicarse y, en caso afirmativo, cuando deben hacerlo.
- Protocolos Peer to Peer: son aquellos en los cuales ningún nodo es el principal, y por lo general todos los nodos poseen la misma autoridad sobre el canal de transmisión.

4.3 ARQUITECTURA DE NIVELES

El propósito de la arquitectura de niveles es reducir la complejidad de la comunicación de datos agrupando lógicamente ciertas funciones en áreas de responsabilidad (niveles), las principales características de esta arquitectura son:

- Cada nivel provee servicios al nivel superior y recibe servicios del nivel inferior.
- Un mensaje proveniente de un nivel superior contiene una cabecera con información a ser usada en el nodo receptor.
- El conjunto de servicios que provee un nivel es llamado Entidad y cada entidad consiste en un manejador (manager) y un elemento (worker).

Se han desarrollado diferentes modelos estándar para la transmisión de datos, uno de los más importantes es el modelo OSI (Open Standar Interface) desarrollado por la ISO (International Standards Organization).

4.4 MODELO OSI

En el modelo OSI ^[4.1](Open System Interconnection), el propósito de cada nivel es proveer servicios al nivel superior, liberándolo de los detalles de implementación de cada servicio. La información que se envía de un computador a otro debe pasar del nivel superior al nivel inferior atravesando todos los demás niveles de forma descendente, dentro del computador que origina los datos. A su paso por cada nivel a los datos se les adiciona información que será removida al llegar a su destino. La información adicionada se clasifica en:

- Información de Control: Dirigida a su nivel correspondiente en el computador de destino. Cada nivel se comporta como si estuviera comunicándose con su contraparte en el otro computador.
- Información de Interface: Dirigida al nivel adyacente con el cual se está interactuando. El objeto de esta información es definir los servicios provistos por el nivel inferior, y como deben ser accesados estos servicios. La información tras ser empleada por el nivel adyacente es removida.

El modelo OSI se estructura en 7 niveles:

- 1. Nivel Físico:** Este nivel dirige la transmisión de flujos de bits, sin estructura aparente, sobre un medio de conexión. Se encuentra relacionado con condiciones eléctricas u ópticas, mecánicas y funcionales de la interfaz al medio de transmisión. A su vez está encargado de aportar la señal empleada para la transmisión de los datos generados por los niveles superiores. En este nivel se define la forma de conectar el cable a las tarjetas de red, cuántos pines debe tener cada conector y el uso funcional de cada uno de ellos. Define también la técnica de transmisión a emplear para el envío de los datos sobre el medio empleado. Se encarga de activar, mantener y desactivar un circuito físico. Este nivel trata la codificación y sincronización de los bits y es el responsable de hacer llegar los bits desde un computador a otro.
- 2. Nivel de Enlace de Datos:** Este nivel se encarga, en el computador de origen, de alojar en una estructura lógica de agrupación de bits, llamada Trama (Frame), los datos provenientes de los niveles superiores. En el computador de destino, se encarga de agrupar los bits provenientes del nivel físico en tramas de datos (Frames) que serán entregadas al nivel de red. Este nivel es el responsable de garantizar la transferencia de tramas libres de errores de un computador a otro a través del nivel físico.
- 3. Nivel de Red:** Es responsable del direccionamiento de mensajes y de la conversión de las direcciones lógicas y nombres, en direcciones físicas. Está encargado también de determinar la ruta adecuada para el trayecto de los datos, basándose en condiciones de la red, prioridad del servicio, etc. El nivel de red agrupa pequeños fragmentos de mensajes para ser enviados juntos a través de la red.
- 4. Nivel de Transporte:** Se encarga de la recuperación y detección de errores. Garantiza también, la entrega de los mensajes del computador originados en el nivel de aplicación. Es el nivel encargado de informar a los niveles superiores del estatus de la red.
- 5. Nivel de Sesión:** Permite que dos aplicaciones residentes en computadoras diferentes establezcan, usen y terminen una conexión llamada sesión. Este nivel realiza reconocimientos de nombres y las funciones necesarias para que dos aplicaciones se comuniquen a través de la red, como en el caso de funciones de seguridad.
- 6. Nivel de Presentación:** Determina el formato a usar para el intercambio de datos en la red. Puede ser llamado el traductor de la red. Este nivel también maneja la seguridad de emisión pues, provee a la red servicios como el de encriptación de datos.
- 7. Nivel de Aplicación:** Sirve como ventana para los procesos que requieren acceder a los servicios de red.

Existen diferentes protocolos de comunicación los cuales trabajan en distintos niveles del modelo OSI ya que cada nivel es usado para una tarea específica.

La información que se quiere transmitir, se genera en la capa superior y en cada capa subsecuente se le va añadiendo la información de control e interfase correspondiente a dicha etapa, de tal forma que la siguiente etapa, recibe esta información adicional como parte del mensaje completo y la trata solo como información a transmitir. En el receptor la información de control e interfase se procesa y se elimina al pasar a la etapa superior, de esta forma el mensaje llega a su destino.

La manera en como funciona un protocolo de comunicación es variada, sin embargo la mayoría tienen una estructura de mensaje como se muestra la figura 4.1, en donde el Header y el Trailer contienen la información de control, necesaria para la correcta transmisión de la información.

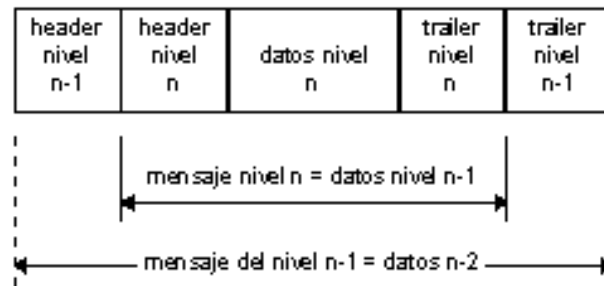


Figura 4.1 Estructura de un mensaje

La siguiente secuencia ejemplifica un posible algoritmo de un protocolo de comunicación entre dos dispositivos en donde uno de ellos envía un mensaje y espera recibir una respuesta de confirmación de la recepción.

1. El dispositivo origen, tiene la información a ser enviada y conoce la dirección del dispositivo destino y la incluye en el mensaje.
2. El dispositivo destino ejecuta la función especificada en el mensaje, construye la respuesta (con resultados y dirección del origen) y envía el mensaje respuesta por una puerta de salida, (quedando libre para aceptar otro mensaje).
3. La respuesta llega al originador, quien realiza un chequeo para asegurarse que viene del lugar correcto antes de aceptarla, luego, pasa al estado "no espera respuesta" en esa puerta de entrada.

Este es un ejemplo de protocolo muy sencillo, necesita de la sintaxis para definición de formatos de los mensajes y una semántica muy simple. Sin embargo muestra la necesidad de incluir información adicional de control para la transferencia de información.

De esta forma los protocolos de comunicación juegan un papel fundamental en la transmisión de datos, presentes en cualquier sistema de comunicación digital, entre procesos y entre dispositivos. Algunos de los protocolos mas usuales para la transferencia de información entre computadoras son: HTTP, FTP, POP3, NFS, IPX/SPX, que realizan diferentes tareas y que se pueden clasificar dentro de las diferentes capas de un modelo como el OSI.

5. DISEÑO DEL SISTEMA

Esta etapa consistió en la planeación y desarrollo del sistema de monitoreo para lograr una óptima funcionalidad y utilidad, del mismo, tomando en cuenta las características del deporte del canotaje así como de los circuitos que integran el sistema.

5.1 PLANEACIÓN Y CONCEPTUALIZACIÓN DEL SISTEMA DE MONITOREO

Debido a las condiciones ambientales de los sitios en donde se practica el Canotaje y a las características de las embarcaciones de kayak en donde se debe implementar el sistema de monitoreo, se requirió que dicho sistema además de ser resistente al agua, fuese un prototipo portátil que pudiera instalarse en el kayak de manera sencilla sin afectar la estructura del mismo, de fácil manejo y con la característica de poder intercambiarse entre una embarcación y otra sin mayor implicación que la referente a la fijación sobre la embarcación.

La fuerza que aplica un remero en cada palada, debe ser visualizada por él atleta a lo largo del trayecto en las distancias de entrenamiento para controlar el ritmo de bogas y magnitud de fuerza establecidas en el plan de entrenamiento diario, sin embargo, dicho control no puede realizarlo el atleta continuamente ya que en momentos de gran esfuerzo debe concentrar toda su atención en la ejecución y desempeño de sus movimientos, ya que además de los parámetros técnicos y teóricos del entrenamiento en sí, debe entrenarse para ampliar su margen de resistencia al cansancio, circunstancia que demanda toda la energía y concentración del atleta en momentos clave durante un entrenamiento. Esta circunstancia hace prácticamente imposible que un atleta con un entrenamiento de alto rendimiento, preste atención a la información que arroja el sistema de monitoreo, por lo que no le es de completa utilidad si no puede analizar posteriormente su ejecución en esos instantes precisos que son los más importantes para efectos del diseño de los planes de entrenamiento individualizados, ya que cada organismo reacciona de manera diferente ante situaciones de extremo cansancio, lo cual permitiría una mejoría en la metodológica de los planes de entrenamiento para incrementar las capacidades tácticas y físicas del atleta en una competencia. Por esta razón se opta por la inclusión de un dispositivo que sea capaz de recibir la información procesada desde la embarcación de manera inalámbrica para ser visualizada por el entrenador quien además de ser el responsable del diseño de los planes de entrenamiento es el encargado de alentar e impulsar al atleta a mantener el ritmo de trabajo.

Por tanto, los dispositivos del sistema basan su funcionamiento en microcontroladores, que por sus características de tamaño, capacidad de procesamiento y facilidad de integración con otros circuitos electrónicos resultan ser los más adecuados para el diseño de este sistema de monitoreo.

5.2 DISPOSITIVO TRANSMISOR

El desplazamiento del kayak hacia el frente, es posible, debido a que el atleta al introducir la cuchara de la pala dentro del agua, ésta, pretende ser un punto fijo dentro del agua desde el cual el remero puede apoyarse para impulsar la embarcación hacia el frente, de tal forma que la pala quede fija en el agua y lo que se desplace sea el kayak, por lo que la técnica particular de cada remero puede ser en mayor o menor grado más eficiente para lograr tal propósito, el cual depende del ángulo de incidencia de la pala en el agua, dirección y recorrido relativo del kayak con respecto de la pala, por lo que no es posible realizar un sistema de monitoreo estándar que mida la fuerza aplicada en cada dirección por el atleta, sin embargo la aceleración total experimentada por el kayak, es debida a la sumatoria de fuerzas realizadas y aplicadas sobre el sistema pala-kayak por el atleta, lo que brinda la posibilidad de realizar una medición que conjunte el sistema de fuerzas total ejercido al medir la aceleración del kayak. Por esta razón el sistema debe montarse en el kayak y no en la pala ni en alguna parte del cuerpo del atleta.

En la práctica, toda embarcación además del movimiento de desplazamiento hacia el frente, experimenta movimientos laterales y verticales los cuales generan olas, que son traducidas como energía sobre el agua, la cual podría ser tomada en consideración para el diseño del sistema, sin embargo para esta aplicación y por las características del kayak se puede decir que éste, tiene un movimiento rectilíneo que prácticamente no es afectado por los movimientos laterales y verticales existentes, ya que además de ser de muy poca magnitud, el atleta al ir remando equilibra dichos movimientos con el cuerpo y con las piernas, con el fin de atenuar el efecto de dichos movimientos y para equilibrarse en la embarcación, ya que el kayak mide tan solo 51 cm. de ancho y cualquier movimiento aplicado que no sea equilibrado por el remero, inevitablemente lo hará volcarse. Por esta razón no hay necesidad de considerar en el sistema 3 ejes de adquisición, y con solo considerar el eje correspondiente a la línea imaginaria proa-popa del kayak, se puede lograr el objetivo de obtener una medición adecuada para calcular la fuerza promedio aplicada en cada remada en una escala dada.

Para el diseño del prototipo ubicado abordo del kayak, se necesitó un dispositivo electrónico sensor, capaz de entregar una señal o parámetro electrónico proporcional a la fuerza, aceleración o velocidad a la que se viera sometido tal circuito en la dirección proa-popa anteriormente especificada. El dispositivo que por sus características fue el más adecuado para este diseño fue el acelerómetro ADXL202JQC de Analog Devices, el cual entrega dos tipos de señales: Una analógica, que entrega un voltaje proporcional a la aceleración y una digital que consiste en una señal cuadrada de frecuencia configurable, con ciclo de trabajo proporcional a la aceleración, en dos diferentes direcciones, ortogonales entre ellas, denominadas X e Y, para ambos tipos de señal. Se decidió usar el eje X como eje de adquisición para el sistema de monitoreo.

La salida digital proveniente del acelerómetro, es la señal idónea para ser procesada por un microcontrolador, ya que al no necesitar de un convertidor analógico-digital se optimiza el proceso de adquisición eliminando tanto el tiempo empleado en dicha conversión como el error derivado de la misma, permitiendo aprovechar los recursos del microcontrolador de una forma mas efectiva, ahorrando memoria y tiempo de procesamiento.

El microcontrolador seleccionado fue el MC68HC811E2 debido a que cuenta con una fuente de interrupción por flancos de subida y bajada, sumamente exacta y sensible a los cambios de nivel lógico, idóneas para el procesamiento de la señal proveniente del acelerómetro, además de contar con una unidad interna de comunicación de tipo serial que facilita el envío de información debido a que la programación de la misma es muy fácil de implementar y cuenta con una cantidad suficiente de puertos para conectar y controlar una pantalla de cristal líquido para la visualización de la información procesada.

5.2.1 CÁLCULO DE LA FUERZA

Independientemente del procesamiento empleado de la señal proveniente del acelerómetro, el parámetro resultante es la aceleración, por lo que una vez obtenido y partiendo de que la velocidad de una embarcación en el agua a bajas velocidades, se puede emplear la “**fórmula general de resistencia o arrastre de un fluido**”^[5.1], que es igual a la resultante de las fuerzas en dirección del movimiento de una embarcación, la cual se deduce mediante el estudio de mecánica de fluidos con un método llamado “**Análisis dimensional**”^[5.2].

Se sabe que la fuerza de resistencia o arrastre **Fres** es función de diferentes variables como se muestra en la ec. 5-1:

$$\mathbf{Fres} = f(\mathbf{v}, \mathbf{L}, \mathbf{A}, \rho, \eta) \quad (5-1)$$

Donde:

v.- Velocidad de la corriente imperturbada teóricamente en el infinito

L.- Longitud característica, que da el tamaño del cuerpo, por ejemplo, la cuerda que va de proa a popa de una embarcación.

A.- Área característica del cuerpo

ρ.- Densidad

η.- Viscosidad dinámica

Al aplicar los conceptos de mecánica de fluidos, se obtiene la “**Formula general de resistencia**”: ec. 5-2.

$$\mathbf{Fres} = (\mathbf{Cw} \mathbf{A} \rho \mathbf{v}^2) / 2 \quad (5-2)$$

Donde:

Cw .- Coeficiente adimensional de arrastre, que es función del Número de Reynolds dependiente de **L**, **ρ**, **v** y **η**, el cual se obtiene de manera experimental.

De esta fórmula se puede apreciar que los valores **Cw** y **ρ**, son valores constantes del fluido y **A** es un valor constante de la embarcación, y como lo que se quiere obtener es una medida de la fuerza aplicada en cada palada dentro de una escala de referencia propia del dispositivo, se pueden agrupar estos términos (ec. 5-3), como una sola constante, lo cual da como resultado la expresión representada en la ec. 5-4:

$$(\mathbf{Cw} \mathbf{A} \rho) / 2 = \mathbf{Z} = \text{cte.} \quad (5-3)$$

$$\mathbf{Fres} = \mathbf{Zv}^2 \quad (5-4)$$

De esta expresión se puede apreciar que la velocidad de la embarcación depende de la fuerza de arrastre que ésta experimente, por lo que para esta aplicación, se puede aplicar la igualdad 5-5:

$$\mathbf{Fres} = \mathbf{ma} = \mathbf{Zv}^2 \quad (5-5)$$

De este análisis, se deduce que la fuerza **Fres** es un parámetro independiente y la velocidad un parámetro dependiente de **Fres**, ya que la fuerza aplicada es la que provoca el movimiento el cual tiene una velocidad caracterizada por la ecuación 5-5.

Así pues, al multiplicar la aceleración por la masa de la embarcación y del atleta, se puede obtener un valor aproximado de la fuerza aplicada, con la particularidad de que para diferentes características del agua en donde se utilice el sistema de monitoreo, se obtendrán diferentes valores de la fuerza para una misma velocidad, sin embargo esto no es de vital importancia ya que lo que se quiere es obtener el valor de la fuerza, en una escala propia del sistema para monitorear los valores de este parámetro durante el entrenamiento y basar su utilidad dentro de la escala del dispositivo.

5.2.2 MÓDULO DE COMUNICACIONES

Para la transmisión y recepción de la información procesada, se optó, por sus prestaciones y bajo costo, el módulo de comunicaciones de transmisión serial RWS434-TWS434 de Reynolds Electronics, que permite la transmisión de datos a una distancia de 400 mts. inalámbricamente utilizando una frecuencia de portadora de 433Mhz, frecuencia que no interfiere con otros sistemas de comunicación que pudieran afectar al sistema ^[5.3], como teléfonos celulares los cuales trabajan a una frecuencia de 800-1900 MHz , ni por walkie-talkies, talkabout, juguetes o sistemas de control remoto que aunque algunos de esos sistemas trabajan a frecuencias que van desde VHF (144-174 MHz) y de UHF (450-470MHz) la posibilidad de ser afectado es muy remota, ya que generalmente los entrenamientos se realizan en espejos de agua de gran extensión siendo muy remota la posibilidad de interferencia. La velocidad de transmisión del módulo de comunicaciones es de 2400 bits/s, velocidad de transmisión suficiente para esta aplicación. El protocolo de comunicaciones utilizado, basó su funcionamiento en el protocolo UART, al cual se le realizaron ligeras modificaciones ya que las características del módulo de comunicaciones así lo requirió.

5.3 DISPOSITIVO RECEPTOR

El dispositivo receptor debía ser capaz de recibir y desplegar la información proveniente del emisor, para lo cual se pensó en primera instancia en usar el mismo modelo de microcontrolador que en el emisor, sin embargo se eligió el PIC16F84, por ser un microcontrolador de menor costo, capaz de realizar dicha tarea con un tipo de programación muy sencilla y principalmente por estar disponible en un encapsulado más pequeño ya que el dispositivo debía ser compacto porque generalmente los entrenadores siguen al atleta en bicicleta o en lancha y deben tener las manos libres para maniobrar su vehículo.

Este dispositivo esta compuesto por una pantalla de LCD, el microcontrolador y el receptor del módulo de comunicaciones.

5.4 DISEÑO A BLOQUES DE LOS DISPOSITIVOS TRANSMISOR Y RECEPTOR

Transmisor

El transmisor esta compuesto principalmente por los siguientes elementos electrónicos:

- Acelerómetro ADXL202JQC (Analog Devices)
- Microcontrolador MC68HC811E2 (Motorola)
- Transmisor TWS-434 (Reynols Electronics)
- LCD 16x2 (Tianma)

La figura 5.1, muestra la disposición y los elementos constitutivos del transmisor.

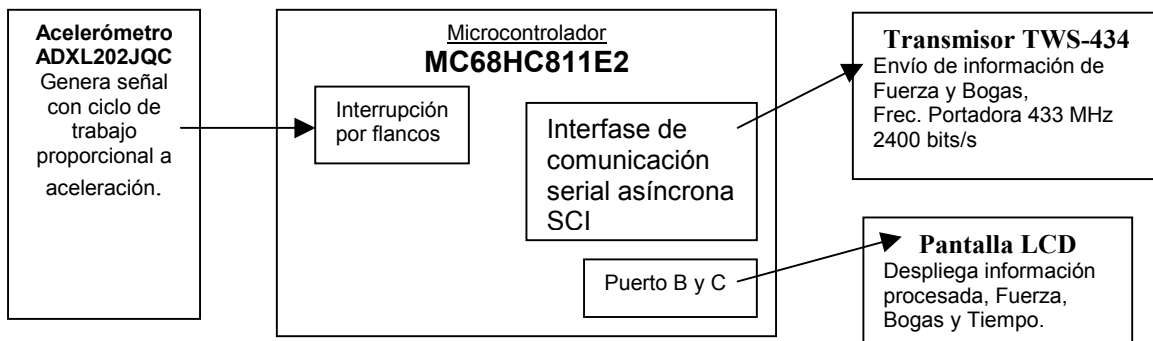


Figura 5.1 Diseño a bloques del transmisor

Receptor

El receptor esta compuesto principalmente por los siguientes elementos electrónicos:

- Microcontrolador PIC16f84A (Microchip)
- Receptor RWS-434 (Reynols Electronics)
- LCD 16x2 (Tianma)

La figura 5.2, muestra la disposición y los elementos constitutivos del receptor.

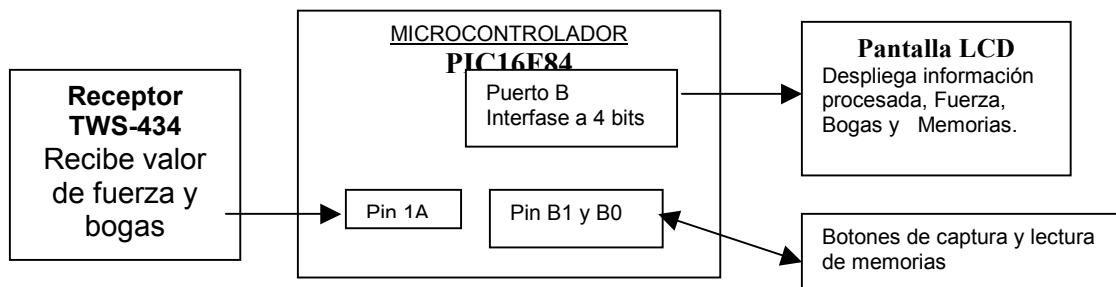


Figura 5.2 Diseño a bloques del receptor

6. DESARROLLO DEL SISTEMA

Esta etapa consistió en la caracterización de cada elemento constitutivo del sistema para finalmente poder integrarlos de manera adecuada.

6.1 MICROCONTROLADORES

Los microcontroladores son dispositivos electrónicos que actualmente se encuentran en la mayoría de los aparatos electrónicos ya que además de ser baratos y confiables, sus recursos y programación permiten obtener grandes beneficios al poder controlar, obtener y procesar información tanto de otros circuitos como de elementos pasivos. Son circuitos integrados que incorporan en su interior, una unidad central de proceso (CPU) y diferentes recursos y tipos de memoria como pueden ser RAM, EPROM, EEPROM, FLASH, por mencionar algunas, en las cuales se almacena y ejecuta la programación de la aplicación para la que son empleados, además de contar con puertos de entrada/salida programables, convertidores analógico-digital, temporizadores, comparadores, capturadores, transmisiones tanto seriales como paralelas. En resumen, un microcontrolador es un circuito integrado independiente, que no necesita memoria ni puertos externos pues los lleva en su interior, que facilitan la tarea de diseño y reduce el espacio, traduciéndose todo a una aplicación final más económica y fiable.

6.1.1 DESCRIPCIÓN DEL MICROCONTROLADOR MC68HC811E2 DE MOTOROLA

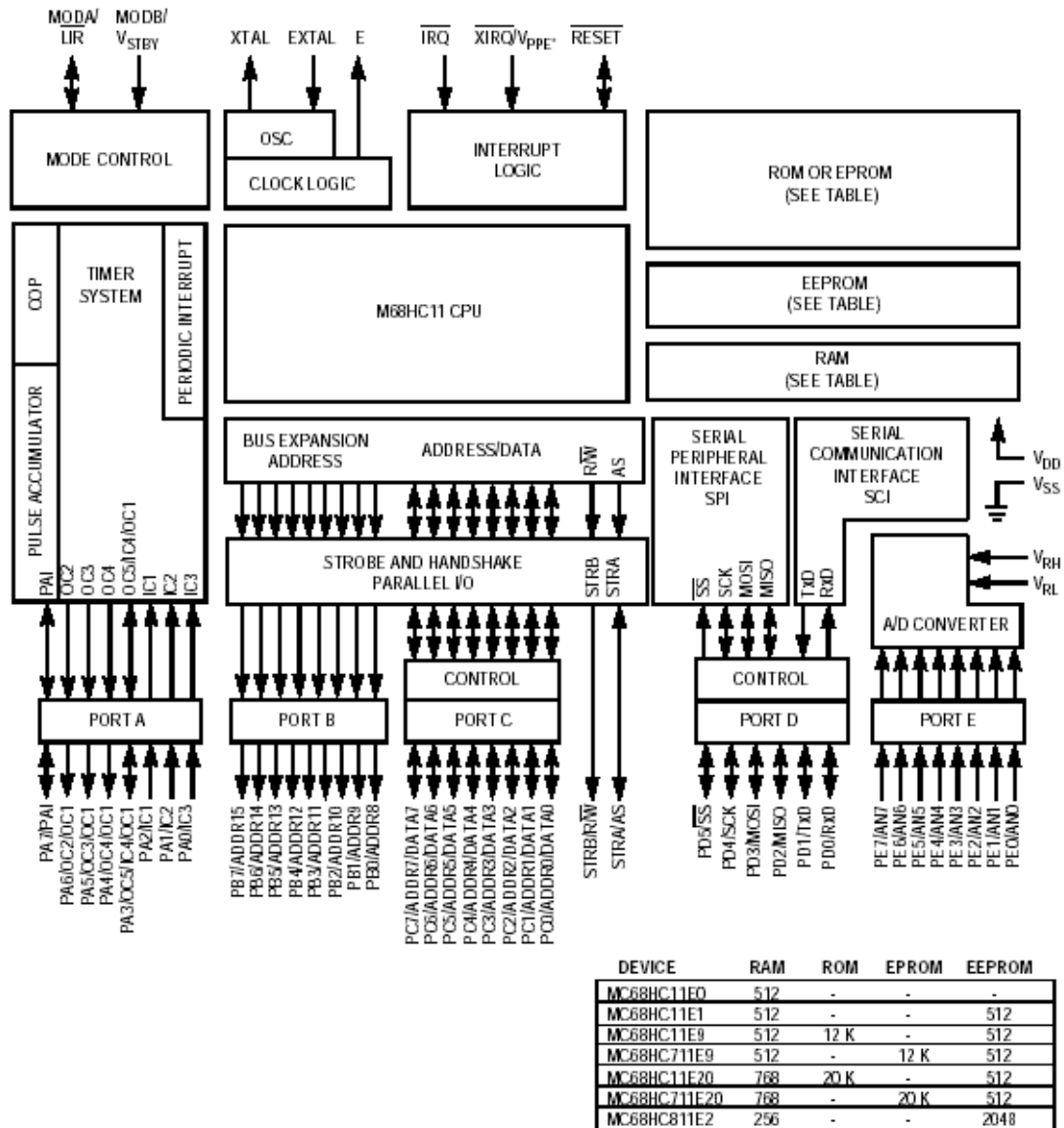
El microcontrolador MC68HC811 es un versión mejorada de la familia MC68HC11 el cual se caracteriza por su alta velocidad y bajo consumo de energía, su diseño permite trabajar a frecuencias desde DC hasta 4 MHz.

Características

- Cuenta con CPU de la familia MC68HC11.
- Modos de ahorro de energía STOP y WAIT.
- Dispositivo de bajo consumo de energía (3.5 – 5.5 Vdc).
- 2048 bytes de memoria EEPROM.
- 256 bytes de memoria RAM.
- Bus de datos y de direcciones no multiplexado.
- Timer de 16 bits mejorado.
- 3 canales de captura (IC).
- 4 canales de comparación (OC).
- Canal adicional seleccionable como de captura (IC) o de comparación (OC).
- Dos acumuladores de 8 bits.
- Dos registros de 16 bits para direccionamiento indexado.
- Circuito de interrupción en tiempo real.
- Perro guardián para la adecuada operación del sistema (COP).
- Comunicaciones seriales asíncronas con código de línea NRZ.
- Comunicación serial síncrona.
- Convertidor analógico digital 8 canales de 8 bits cada uno .
- 4 modos seleccionables de trabajo.
- 38 pines de propósito general.
 - 16 pines bidireccionables
 - 11 pines de entrada
 - 11 pines de salida

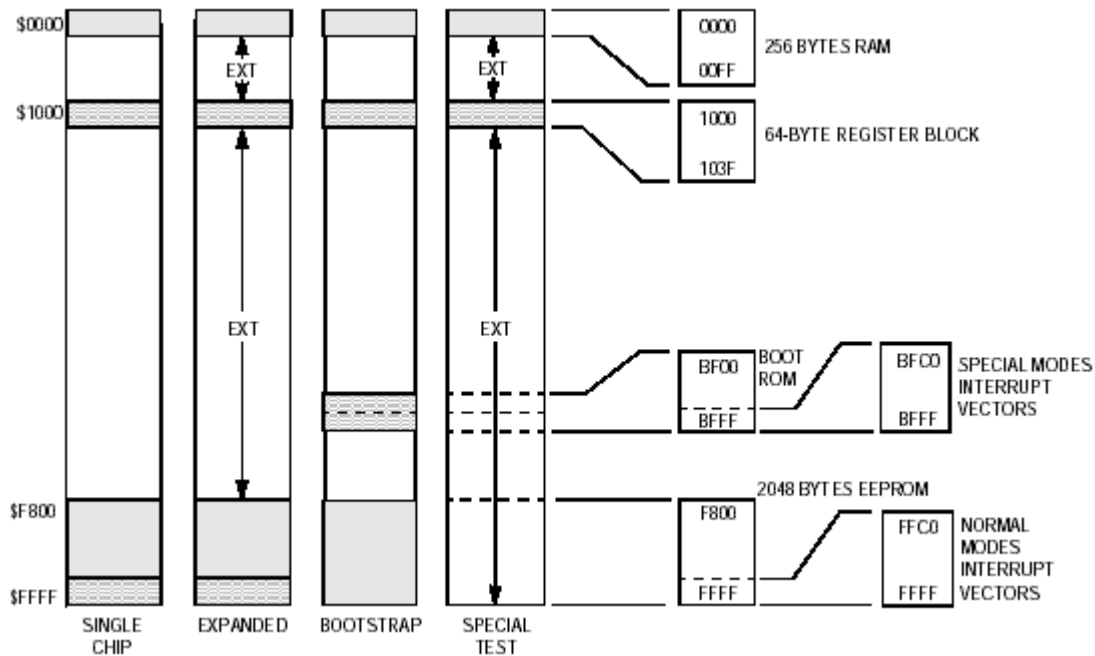
La figura 6.1 muestra el diagrama a bloques de la familia de microcontroladores a la que pertenece el microcontrolador empleado. La figura 6.2 muestra el mapa de memoria de dicho microcontrolador.

Figura 6.1 DIAGRAMA A BLOQUES DE LA FAMILIA DE MICROCONTROLADORES MC68HC811



* V_{PEP} applies only to devices with EPROM/OTPROM.

Figura 6.2 MAPA DE MEMORIA DEL MICROCONTROLADOR MC68HC811E2



Los modos de trabajo son seleccionables por medio de los pines MODA y MODB, cada uno de estos modos de trabajo son utilizados para usarse de acuerdo a los requerimientos de la aplicación, los modos más usuales son **Bootstrap** y **Single chip**, porque el microcontrolador puede ser usado de manera independiente, utilizando solo los recursos internos de cada modelo, sin embargo para comunicaciones o acceso con otros dispositivos como memorias se emplea el **Modo expandido**. El modo **Special Test** es empleado por el fabricante para la comprobación del chip.

El programa utilizado para generar el código fuente del microcontrolador fue IASM11 y el software para cargar el programa fue el PCBUG11, ambos diseñados por el fabricante.

La tarjeta de programación fue diseñada de acuerdo a las recomendaciones del fabricante, en una tableta universal, para programarse con el puerto serial de una PC.

6.1.2 DESCRIPCIÓN DEL MICROCONTROLADOR PIC16F84 DE MICROCHIP

El PIC16F84 es un microcontrolador de 8 bits con las características comunes de los microcontroladores modernos: CPU, ALU, memorias RAM, ROM, Flash, cuenta con 2 puertos bidireccionales de 8 y 5 pines.

Los microcontroladores PIC son de arquitectura RISC, lo que los hace más eficaces para la mayoría de aplicaciones, ya que este tipo de arquitectura permite efectuar la mayoría de las instrucciones en un ciclo de instrucción, lo que permite realizar programas estructuralmente más veloces y eficientes. El direccionamiento de instrucciones corren simultáneamente por buses diferentes, las instrucciones son almacenadas en una memoria flash de 1K x 14, en donde cada instrucción ocupa 14 bits de los cuales los 8 bits menos significativos son de información y los siguientes 5 son utilizados para el direccionamiento de datos o de instrucciones, esta propiedad los hace ser unos dispositivos muy competentes, además de ser de un costo reducido.

Las principales características del PIC16F84 son:

- CPU de arquitectura RISC de alto desempeño
- Juego de solo 35 instrucciones para programación
- Todas las instrucciones se ejecutan en un ciclo excepto aquellas de salto de programa que se ejecutan en dos ciclos
- Velocidad de operación de DC a 20 MHz DC - 200 ns ciclo de instrucción
- 1024 palabras de memoria de programa
- 68 bytes de memoria RAM para datos
- 64 bytes de memoria EEPROM para datos
- Ancho de palabra de instrucción de 14-bits, 8 bits para datos
- 15 registros de configuración
- Stack de 8 niveles
- Direccionamiento directo, indirecto y relativo.
- Cuatro Fuentes de interrupción
- Pin RB0/INT para interrupciones de flancos de subida o bajada.
- Interrupción por desbordamiento del timer TMR0
- Cambio de estado del Puerto B PORTB<7:4>
- Finalización de la escritura de la memoria EEPROM

Características de los periféricos

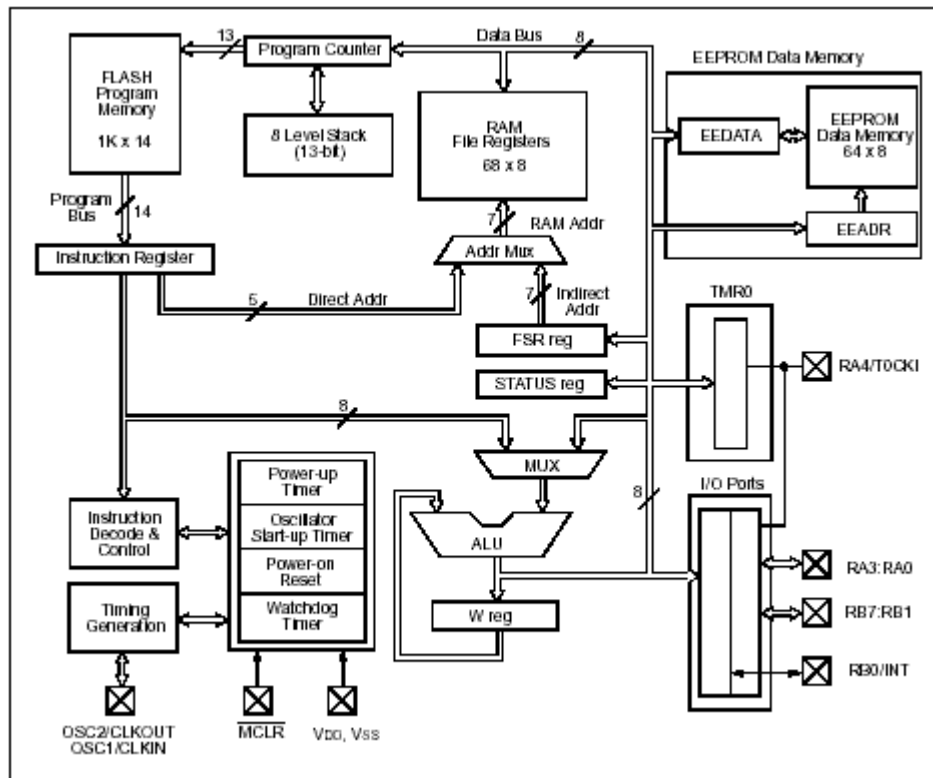
- 13 pines configurables de entrada /salida
- 25 mA de corriente de entrada por pin
- 25 mA de corriente de salida por pin
- Contador (TIMER) de 8 bits de cuenta programable

Características especiales

- 10,000 ciclos de lectura/escritura de la memoria FLASH
- 10,000,000 ciclos de lectura/escritura de la memoria EEPROM
- Retención de datos en la EEPROM > 40 años
- Programación serial via 2 pines
- Perro guardian WDT, Power-on Reset (POR), Power-up Timer (PWRT),
- Oscillator Start-up Timer (OST)
- Protección de código
- Protección de código
- Modo de ahorro de energía SLEEP
- Diferentes opciones del circuito oscilador

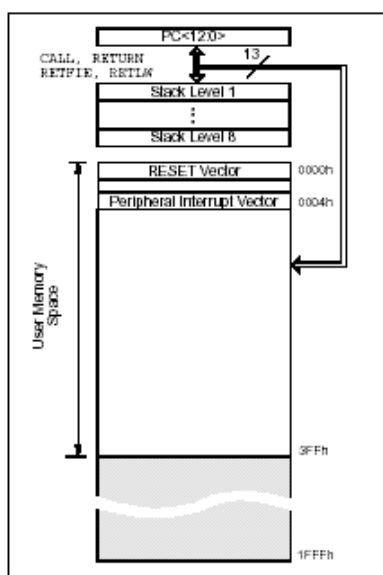
La figura 6.3 muestra de manera esquemática los diferentes recursos del PIC16F84, así como su funcionalidad.

Figura 6.3 DIAGRAMA FUNCIONAL A BLOQUES DEL PIC16F84



La figura 6.4 muestra el mapa de memoria con que cuenta el PIC16F84.

Figura 6.4 MAPA DE MEMORIA DE PROGRAMA DEL PIC16F84



Existen diferentes tipos de lenguajes para la programación de estos dispositivos, el lenguaje utilizado en este trabajo fue: JAL (Just Another Language), desarrollado por el alemán Wouter Van Ooijen, el cual es un lenguaje de distribución libre, el cual integra la programación, compilación, subrutinas y recursos característicos de lenguajes de alto nivel como las instrucciones **for**, **if**, **do while** y demás herramientas que ofrecen al usuario la posibilidad de desarrollar programas más rápidamente y de manera menos compleja, características que aceleran el proceso de experimentación e implementación del proyecto.

Para la carga del programa al microcontrolador se usó el software y tarjeta **NOPPP** (No Parts PIC Programmer) por sus siglas en ingles, que como lo indica se construye con muy pocos elementos electrónicos y su manejo es bastante simple.

Nota : Se anexan el circuito del NOPPP y links relacionados, en el apéndice correspondiente.

6.2 CONTROL DEL LCD (LIQUID CRISTAL DISPLAY).

Esta fase consistió, en la realización de los algoritmos y códigos correspondientes para implementar el control del LCD con los microcontroladores e integrar este elemento electrónico, a los dispositivos que componen este trabajo.

Este elemento esta constituido por un circuito impreso en donde esta integrado el controlador del display y los pines del mismo además del LCD en sí, en el cual se pueden visualizar dos líneas de 16 caracteres cada una, aunque cada línea realmente esta constituida por 40 caracteres, los cuales se almacenan en una memoria DDRAM (Display Data RAM) como se muestra en la figura .

Las direcciones de acceso para esta memoria en formato hexadecimal son:

Línea 1 = \$80 hasta la \$8F para los primeros 16 caracteres que son los que se pueden visualizar y de la \$90 hasta la \$A8 para los restantes 24 caracteres

Línea 2 = \$C0 hasta la \$CF para los primeros 16 caracteres que son los que se pueden visualizar y de la \$D0 hasta la \$E8 para los restantes 24 caracteres

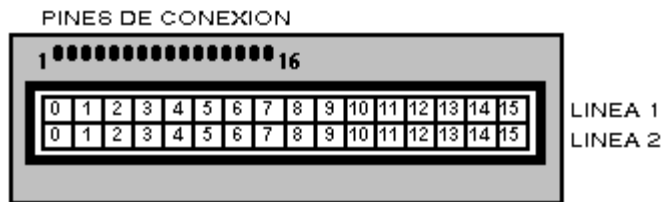


Figura 6.5 Display de 16 caracteres 2 líneas

El LCD dispone de una matriz de 5x8 ó 5X10 puntos para representar cada carácter.

En total se pueden representar 256 caracteres diferentes, 240 caracteres están grabados dentro del LCD y representan las letras mayúsculas, minúsculas, signos de puntuación, números, etc... Existen 8 caracteres que pueden ser definidos por el usuario los cuales se almacenan en la memoria CGRAM (Character Generator RAM).

En la figura 6.6 se muestra gráficamente cómo es la matriz de representación de los caracteres.

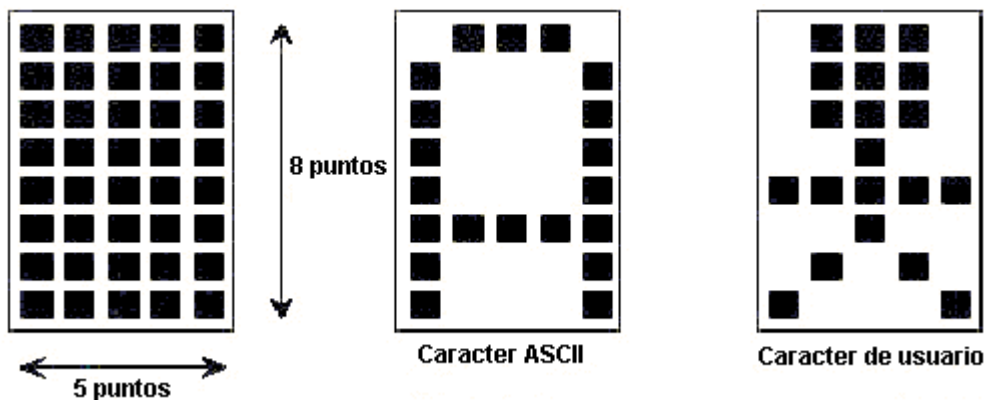


Figura 6.6 Representación de la matriz que compone un carácter

El display tiene 16 pines numerados, los cuales se describen en la tabla 6.

Tabla 6.1 Pines del LCD

No. De PIN	Simbolo	Descripción
1	Vss	Tierra
2	Vdd	Alimentación
3	Vc	Voltaje de ajuste de contraste
4	RS	Selección de registro
5	R/W	Lectura/Escritura
6	E	Enable (habilitación)
7 al 14	Pin 7 = D0 Pin 14 = D7	Bits de datos: D0 Bit menos significativo (LSB) D7 Bit mas significativo (MSB)
15	Led +	Contraste +
16	Led -	Contraste -

Después de encender el LCD, aparecerá la línea superior un poco más oscura que la inferior, esto quiere decir que el display no ha sido inicializado todavía. Para desplegar caracteres en el LCD, primero se debe realizar una inicialización del mismo, ya que el controlador de este dispositivo así lo requiere. Dependiendo de la marca del LCD el fabricante recomienda un proceso de inicialización, en el cual además de encender y habilitar el display para lectura y/o escritura de caracteres, se configura el tamaño de los caracteres que serán empleados, la forma en que aparecerá el cursor, se configura también si el cursor ira escribiendo los caracteres hacia la derecha o hacia la izquierda y también la forma en como se enviaran los datos para que los caracteres aparezcan en el display, ya que el formato de envío puede ser de 8 o 4 bits. Se puede observar cómo la línea superior deja de estar más oscura que la inferior cuando el LCD ha sido inicializado.

La inicialización para el caso de 8 bits es sencilla de implementar, ya que los datos tanto de comandos como de caracteres se envían de forma paralela a los pines de bits datos, sin embargo para el caso de 4 bits es un poco más complejo. En el caso de 4 bits sólo se conectan los 4 bits mas significativos del LCD, dejando los otros 4 al 'aire' y las transferencias hay que realizarlas en dos partes: primero se envían los 4 bits mas significativos y después los 4 bits menos significativos. Por ejemplo para enviar el código \$28 hay que enviar primero los bits (0 0 1 0) y después los bits (1 0 0 0).

Los comandos son las instrucciones que se envían al LCD para manipular las propiedades del cursor como lo es su posición, encendido (parpadeo), desplazamiento (izquierda o derecha) por mencionar los mas importantes, y también para las opciones de pantalla como lo es el borrado de la misma. Por ejemplo para imprimir caracteres se debe especificar el renglón y la columna donde se quiere imprimir, para lo cual primero se envía el comandos correspondiente de posición y posteriormente el valor del caracter.

En resumen los códigos de comando o los valores del caracter se deben enviar a los pines de datos D0-D7 siendo D7 el bit mas significativo para el caso de 8 bits y para el caso de 4 bits primero se envían los 4 bits mas significativos y posteriormente los 4 bits menos significativos hacia los pines D4-D7 siendo D7 el correspondiente al bit mas significativo.

Los pines "RS,RW y ENABLE" son los pines de control para habilitar la lectura o escritura de los caracteres así como la habilitación de envío de comandos o caracteres.

- **RS (Register Select)**.- Este pin tiene la función de habilitar el proceso de envío de caracteres o comandos **RS = 0** Registro de control (Comandos), **RS = 1** Registro de datos (Caracteres)
- **R/W (Read / Write)**.- Permite seleccionar si la operación que se va a realizar en el display es de lectura o de escritura, **R/W=1** lectura, **R/W=0** escritura, en cualquiera de las memorias CG RAM o DD RAM, así como leer el estado interno del display (ocupado o disponible). Cuando el display se encuentra procesando la información enviada, este pin R/W cambia a estado lógico "1", indicando que no se puede realizar ninguna transferencia de información en ese momento, por lo cual se debe esperar hasta que cambie de estado a "0", es entonces cuando el display esta listo para realizar cualquier otra operación, sin embargo si solo se quiere escribir en el display no es completamente necesario leer el estado de este pin, ya que generalmente cada proceso interno dura un tiempo especificado por el fabricante de aproximadamente 40 μ s, por lo que se puede programar un retardo que sea mayor o igual a este tiempo de procesamiento en el dispositivo que controla el LCD y posteriormente realizar la siguiente instrucción, en esta modalidad el pin debe conectarse a tierra.
- **E (Enable)**.- Pin de habilitación, para que el código de carácter o comando sea efectuado por el LCD, este pin debe de cambiar de estado lógico "0" a "1" y dependiendo del estado del pin RS funcionaran como datos de carácter o como datos de comando, el estado lógico de este pin en "1" debe permanecer un tiempo mínimo (especificado por el fabricante), para que la información enviada tenga efecto, posteriormente se debe regresar al estado lógico "0". Cada vez que se quiera introducir información al LCD para enviar comandos o caracteres, se debe de realizar el procedimiento de cambio de estado antes mencionado en el pin ENABLE y la información en los pines de datos debe permanecer activa durante ese tiempo de captura.

Los siguientes procedimientos muestran los procedimientos de inicialización típicas de un display para 4 y 8 bits.

6.2.1 INICIALIZACIÓN PARA 4 BITS

- Se envían los códigos en partes, primero los 4 bits mas significativos y después los 4 menos significativos. Su utilizan los pines de datos D4-D7.
 - Cambio de estado en el bit ENABLE en cada envío de datos.
 - Pines RS=0 y R/W =0
1. Encendido del display.
 2. Esperar a que se alcance el 90% del valor de voltaje nominal, aprox. 15ms para un display de 5 volts de alimentación.
 3. Código = 0011_0011.- La interfase de datos es a 4 bits
 4. Esperar mas de 4.1 ms
 5. Código = 0011_0011.- La interfase de datos es a 4 bits
 6. Esperar 100 μ s
 7. Código = 0011_0011.- La interfase de datos es a 4 bits
 8. Código = 0010_0010.- La interfase de datos es a 4 bits
 9. Código = 0010_1000.- 2 líneas y matriz de 5x10 puntos
 10. Código = 0000_1100.- Enciendo display
 11. Código = 0000_0001.- Borro display
 12. Código = 0000_0110.- Cursor apagado, desplazamiento hacia la derecha
 13. Fin de inicialización

6.2.2 INICIALIZACIÓN PARA 8 BITS

- Se envían los códigos completos al los pines de datos. Se utilizan los pines de datos D0-D7 siendo D7 el bit mas significativo
 - Cambio de estado en el bit ENABLE en cada envío de datos.
 - Pines RS=0 y R/W =0
1. Encendido del display.
 2. Esperar a que se alcance el 90% del valor de voltaje nominal, aprox. 15ms para un display de 5 volts de alimentación.
 3. Código = 0011_1000.- La interface de datos es a 8 bits
 4. Esperar mas de 4.1 ms
 5. Código = 0011_1000.- La interface de datos es a 8 bits
 6. Esperar 100 μ s
 7. Código = 0011_1000.- La interface de datos es a 8 bits
 8. Código = 0011_1000.- La interface de datos es a 8 bits
 9. Código = 0010_1000.- 2 líneas y matriz de 5x10 puntos
 10. Código = 0000_1100.- Enciendo display
 11. Código = 0000_0001.- Borro display
 12. Código = 0000_0110.- Cursor apagado, desplazamiento hacia la derecha
 13. Fin de inicialización

Los anteriores procedimientos de inicialización pueden optimizarse, variando los tiempos de espera entre cada comando así como los tiempos empleados en las rutinas de los pines de control y adecuando un retardo entre cada instrucción que se envíe al display equivalente o mayor al tiempo que le toma al display en realizar los procedimientos internos de cada instrucción.

Después de realizar la inicialización, el LCD esta listo para imprimir caracteres, el siguiente ejemplo muestra la manera en que se imprimen los caracteres.

Ejemplo de control del LCD a 8 bits

“Borrar pantalla e impresión del carácter “A” en la línea 2 y columna 4, (formato Hex).

- R/W =0, RS=0, E=0
 - Ingresar el código “\$01” en los pines de datos, “Borrar Pantalla”
 - R/W =0, RS=0, E=1 ;pulso en pin Enable
 - Esperar un tiempo definido por el fabricante
 - R/W=0, RS=0, E=0
 - Esperar aprox 40 μ s
 - R/W =0, RS=0, E=0
 - Ingresar el código “\$C4” en los pines de datos, “Posicion cursor Linea 2 columna 4”
 - R/W =0, RS=1, E=1 ;pulso en pin Enable
 - Esperar un tiempo definido por el fabricante
 - R/W=0, RS=0, E=0
 - Esperar aprox 40 μ s
-
- R/W =0, RS=0, E=0
 - Ingresar el código “\$41” en los pines de datos, Letra “A” ASCII
 - R/W =0, RS=0, E=1 ;pulso en pin Enable
 - Esperar un tiempo definido por el fabricante
 - R/W=0, RS=0, E=0

- Esperar aprox 40µs
- Carácter A impreso en el display

El LCD empleado en este proyecto fue el TM1562AD marca TIANMA y los procedimientos de inicialización y rutinas de envío de comandos y caracteres fueron adecuados en el lenguaje JAL el cual incluye subrutinas de control de LCD para el PIC 16F84 en el dispositivo receptor y programados en ensamblador usando IASM11 para el transmisor con el MC68HC811E2.

*Se anexan las hojas de especificaciones de LCD en el apartado correspondiente.

6.2.3 RUTINA DE INICIALIZACIÓN DEL LCD PARA EL RECEPTOR

Interfase: 4 bits pines B4 – B7

Pines de control: R/W = tierra, RS= pin B3, E = pin B2

Lenguaje: JAL (fragmento)

Dispositivo: PIC16F84

*****Inicialización LCD usando PIC16F84*****

```
-- initialisation
_hd44780_4_init      -- IO pin initialisation
delay_10mS( 10 )    -- gives the HD44780 time to wake up
hd44780_4_DI = low  -- select instructions
HD44780_IO( 0b_0011 ) -- sync one
delay_10uS( 50 )    -- extra delay
HD44780_IO( 0b_0011 ) -- and one more
delay_10uS( 50 )    -- extra delay
HD44780_IO( 0b_0011 ) -- end yet another
delay_10uS( 50 )    -- extra delay
HD44780_instruction( 0b_0011_0010 ) -- init
HD44780_instruction( 0b_0010_1000 ) -- two lines, 5x7 (high nibble)
```

*****fin inicialización*****

6.2.4 RUTINAS DEL LCD PARA EL TRANSMISOR

Interfase: 8 bits puerto B

Pines de control: R/W = tierra, RS = pin, C0, E= pin C1

Lenguaje: Ensamblador IASM11

Dispositivo: MC68HC811E2

*****INICIALIZACION DISPLAY*****

```
ldx #$ffff ;Retardo para que encienda el display
l   dex      ;en modo bootstrap
bne l
ldaa #$38  ;function set,8 bits, 2 lineas,5x10 puntos
jsr coman
ldaa #$0c  ;display on,cursor off
jsr coman
ldaa #$01  ;borra display
jsr coman
```



```
ldaa #$06 ;entry mode set,desplaza a la derecha
jsr coman
```

*****FIN INICIALIZACION DISPLAY*****

*****SUBROUTINAS DEL DISPLAY 811E2 *****

```
dato      psha ;subrutina de escribir dato en display
          pshx
          ldx #$1000
          staa $04,x ;datos en portb
          bset $03,x,03 ;RS=1 E=1 ;port C pines 0 y 1
          ;RS=PC0 , E=PC1
          bclr $03,x,02 ;RS=1 E=0
          bclr $03,x,03 ;RS=0 E=0
          bsr delay
          pulx
          pula
          rts

coman     psha ;sub para mandar comandos al display
          pshx
          ldx #$1000
          staa $04,x ;comandos en port B
          bclr $03,x,03 ;RS=0 E=0
          bset $03,x,02 ;RS=0 E=1 ;port C pines 0 y 1
          ;RS=PC0 , E=PC1
          bclr $03,x,02 ;RS=0 E=1
          bsr delay
          pulx
          pula
          rts

delay     psha ;delay de 1.414 mseg.
          pshx
          ldaa #$ff
retar     deca
          nop
          nop
          nop
          bne retar
          pulx
          pula
          rts
```

*****FIN SUBROUTINAS DEL DISPLAY*****

6.3 ACELEROMETRO

El acelerómetro empleado, fue el ADXL202JQC de la familia de acelerómetros de Analog Devices, usado en aplicaciones tales como, dispositivos de realidad virtual, navegación, monitoreo sísmico, sistemas de seguridad en vehículos, sensores de inclinación, pruebas de impacto, y demás aplicaciones en donde esté implícita la aceleración dinámica (vibraciones, desplazamientos) o la aceleración estática (fuerza de gravedad), en un rango de $\pm 2g$, capaz de medir aceleración positiva y negativa, en dos ejes, X e Y perpendiculares entre sí, para cada uno de los cuales el dispositivo genera una señal digital con ciclo de trabajo proporcional a la aceleración y de periodo configurable por el usuario en el rango de 0.5 ms -10 ms. (2 KHz – 100 Hz). También tiene la posibilidad de generar voltajes proporcionales a la aceleración en los pines YFILT y XFILT.

Existen modelos de esta familia con rangos de medición desde $\pm 10g$. hasta a $\pm 50g$. con 3 ejes de adquisición.

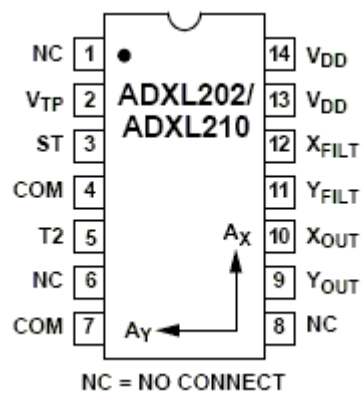


Figura 6.7 Pines de conexión y ejes de adquisición del acelerómetro ADXL202JQc

El ancho de banda puede ser fijado desde 0.01 Hz hasta 5 kHz. El ruido característico del circuito es de $500\mu g/Hz$ lo cual permite realizar adquisiciones con variaciones de gravedad de $5mg$, para anchos de banda por debajo de 60 Hz.

Las señales de salida del acelerómetro pueden ser fácilmente acopladas a los puertos de entrada de un microcontrolador o un DSP, los cuales permiten procesar de manera sencilla dicha señal, a un costo reducido, lo cual hace de este circuito un dispositivo versátil, de fácil manejo y apropiado para un sin fin de aplicaciones.

6.3.1 PRINCIPIO DE FUNCIONAMIENTO

El acelerómetro ADXL202^[6.1], contiene en su interior un sensor de micromecanizado de superficie fabricado con polisilicon y circuitería de acondicionamiento de señal, para la implementación de una arquitectura de lazo abierto para la medición de la aceleración. A la salida de cada eje de adquisición, se encuentra un circuito modulador que convierte la señal analógica, en una señal digital modulada en ciclo de trabajo (figura 6.8).

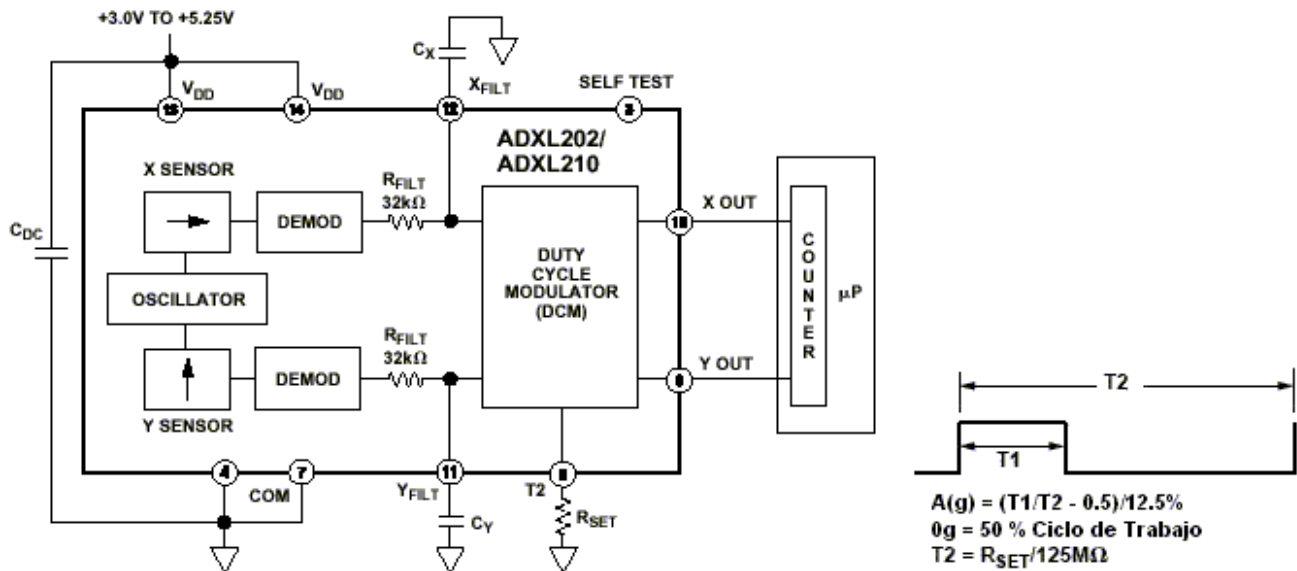


Figura 6.8 Diagrama funcional y señal de salida del acelerómetro

El sensor tiene una estructura de micromecanizado de superficie, construida en la parte superior de la oblea de polisilicon del circuito integrado. Unas alas de polisilicon suspenden la estructura sobre la superficie de la oblea las cuales proveen la resistencia necesaria en contra de la aceleración a la que se somete la estructura para permitir que se mueva lo suficiente sobre el eje para el cual esta diseñada. El desplazamiento de la estructura al experimentar una aceleración, es medida usando capacitores diferenciales formados por dos placas pegadas en la superficie de la estructura y dos placas fijas en la oblea, cada uno de estos capacitores diferenciales se encuentran a la salida de un oscilador que genera dos señales cuadradas que inicialmente se encuentran desfasadas 180° entre sí, cuando se presenta una variación en la capacitancia, las señales cuadradas presentan una amplitud y desfase proporcionales a la aceleración. Posteriormente se utilizan técnicas de desmodulación sensibles al desfase entre las dos señales, usadas para rectificar la señal y determinar la dirección de la aceleración. La señal resultante es usada por el modulador de ciclo de trabajo a través de una resistencia de 32 kΩ, en este punto se tiene la posibilidad de fijar el ancho de banda requerido con un capacitor, esta medida mejora la resolución y ayuda a prevenir el fenómeno de "aliasing", ya que se forma un filtro pasa-bajas. Después de este filtro la señal analógica es introducida al DCM (Duty Cycle Modulator) y es convertida en una señal digital modulada en ciclo de trabajo, es decir, el ciclo de trabajo es proporcional a la aceleración. Una resistencia es utilizada en uno de los pines para fijar el periodo de la señal T2 entre 0.1 ms y 10 ms. correspondientes a las frecuencias 2 KHz – 100 Hz respectivamente.

Cuando el dispositivo no se ve sometido a aceleración alguna, la señal de salida presenta un ciclo de trabajo del 50%, en caso contrario, el ciclo de trabajo cambia dependiendo de la dirección en que se presenta la aceleración, el cambio en porcentaje del ciclo de trabajo nominalmente es de 12.5%, por gravedad (9.81 m/s^2). La figura 6.9 muestra los valores nominales de la señal de salida cuando el dispositivo se ve sometido a la fuerza de gravedad, en distintas posiciones.

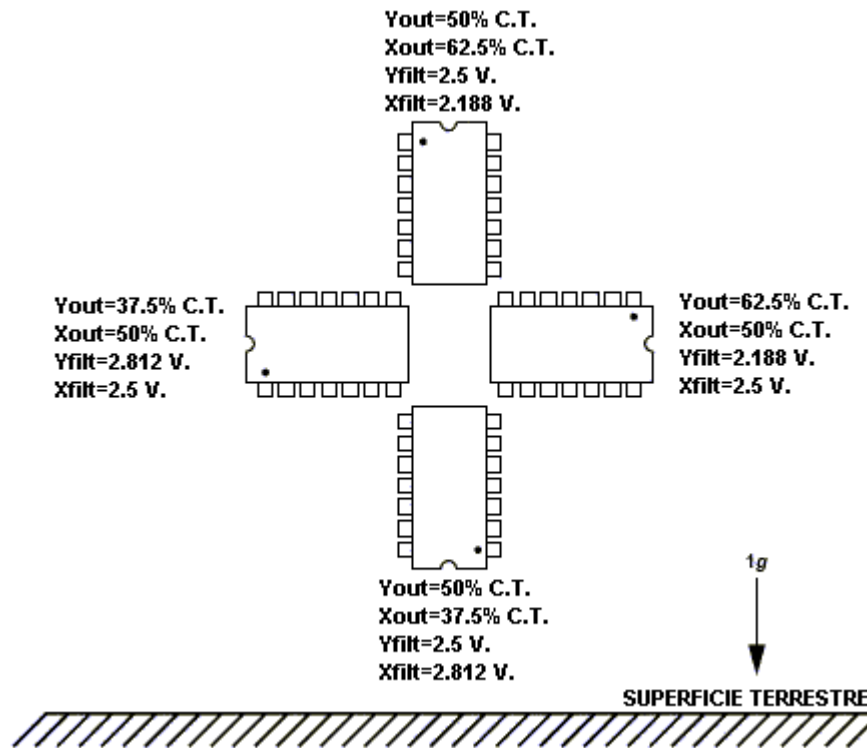


Figura 6.9 Valores nominales de respuesta a la gravedad

Se deben de realizar las pruebas experimentales pertinentes para determinar los valores exactos de cada circuito en especial y con esto determinar la variación de la señal de salida con respecto de la gravedad, para definir de manera adecuada este parámetro de comparación.

6.3.2 CONSIDERACIONES DE DISEÑO

El periodo y ancho de banda son características de la señal de salida que deben ser determinadas por el usuario, para integrar de la manera más conveniente el acelerómetro al sistema de aplicación final. Además de los pines de alimentación, el acelerómetro tiene para la configuración de la señal de salida, la posibilidad de controlar el ancho de banda y ruido con elementos pasivos de forma sencilla.

La figura 6.10 describe los pines de conexión usados para configurar la señal de salida.

Pines de conexión del acelerómetro ADXL202JQC1		
No. Pin	Nombre	Descripción
1	NC	Sin conexión
2	Vtp	Punto de prueba
3	ST	Auto prueba
4	COM	Tierra
5	T2	Ajuste periodo de señal
6	NC	Sin conexión
7	COM	Tierra
8	NC	Sin conexión
9	Yout	Salida digital eje Y
10	Xout	Salida Digital eje X
11	Yfilt	Filtro eje Y
12	Xfilt	Filtro eje X
13	VDD	Alimentación 3 – 5.25 V
14	VDD	Alimentación 3 – 5.25 V

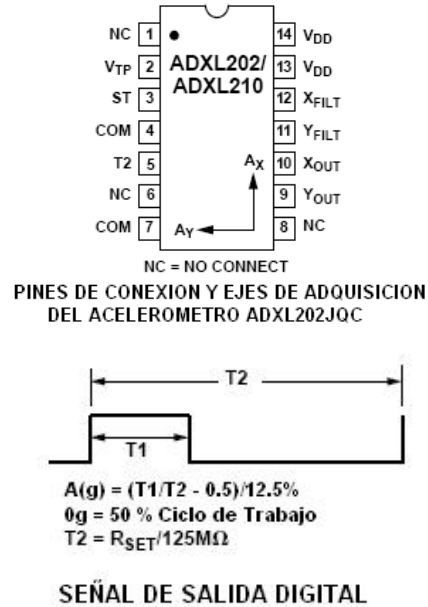


Figura 6.10 Disposición física, descripción de los pines de conexión y señal de salida del acelerómetro.

VTP

Este pin debe dejarse sin conexión, no realizar conexión de ningún tipo.

ST

Controla la opción de Auto Prueba. Cuando el pin se conecta a la alimentación se produce una variación del 10% en el ciclo de trabajo de la señal de salida, con esto se verifica que el circuito se encuentra funcional. En funcionamiento normal no se debe conectar o conectarlo a tierra.

T2

En este pin, se conecta una resistencia (RSET) para configurar el periodo deseado (0.5 -10 msec.) $T2=RSET/125M\Omega$. Cuando el dispositivo no está sometido a ningún tipo de aceleración, la señal de salida tiene un ciclo de trabajo del 50% y la variación nominal por gravedad en el ciclo de trabajo es de 12.5%, cuando se usan las salidas analógicas, se recomienda una resistencia de entre 500k Ω y 2M Ω , la cual debe ubicarse físicamente lo más cerca posible de este pin para minimizar la capacitancia parásita.

XFILT, YFILT

Estos pines sirven para fijar el ancho de banda en el cual trabajara la señal, es decir, la resolución de la señal de salida (el mínimo valor de aceleración detectable), formando un filtro pasa-bajas, para minimizar el aliasing y el ruido de la señal de salida. La resistencia interna con la cual se forma el filtro tiene un valor de 32 kΩ con una tolerancia de ±25%, la cual debe considerarse en el diseño.

La frecuencia de corte de un filtro RC esta dada por la ec. 6-1:

$$F_{-3dB} = 1 / (2\pi RC) \quad (6-1)$$

Para este caso R= 32kΩ, cuyo resultado se expresa en la ec. 6-2:

$$F_{-3dB} = 5\mu F / C \quad (6-2)$$

El valor mínimo de capacitancia recomendado por el fabricante es de 0.001μF

6.3.3 INTEGRACION DEL ACELEROMETRO AL MICROCONTROLADOR

La pieza medular del sistema de monitoreo es el acelerómetro, ya que este dispositivo es el que proporciona la señal para la medición de la aceleración, por lo que debía integrarse de manera conveniente al microcontrolador usado el MC68HC811E2, usando las interrupciones de flancos de bajada y de subida (capturadores del microcontrolador).

Para la determinación del periodo de la señal fue necesario el análisis del número máximo de bogas por minuto que un ser humano puede llegar a efectuar y se observó que el número máximo de bogas que un remero puede realizar por minuto no sobrepasa las 170 bpm, lo que da un máximo de 2.83 bogas por segundo, o sea que el mínimo intervalo entre una boga y otra es de 353ms. Por lo que con una señal con periodo de 10ms sería suficiente para medir la aceleración obteniendo como mínimo 35 mediciones por segundo, suficientes para poder determinar la aceleración máxima efectuada en cada palada y daría la posibilidad de realizar otras tareas con microcontrolador debido a que el tiempo entre la ocurrencia de interrupciones por flanco sería más amplio. Sin embargo se empleo un periodo de 8ms. debido a que comercialmente es mas fácil conseguir una resistencia de 1000 kΩ que una de 1250 kΩ.

Se busco la más alta resolución en la medición, por lo que el ancho de banda requerido fue el máximo 500Hz con un error inherente al sistema de 5.48%, el cual no es un error significativo. Los capacitores empleados fueron de 0.01μF y los valores de configuración de señal son los mostrados en las ecs. 6-3 y 6-4:

$$\text{Periodo (T2)} = 8\text{ms}, R_{\text{set}} = 1\text{M}\Omega \quad (6-3)$$

$$\text{BW} = 500 \text{ Hz}, C_x = C_y = 0.01\mu\text{F} \quad (6-4)$$

6.3.4 CARACTERIZACIÓN DEL ACELERÓMETRO

Se procedió a la caracterización experimental del circuito para la determinación de los valores de variación de la gravedad con el eje X usando la gravedad terrestre.

Tabla 6.2 Resultados obtenidos con el osciloscopio (mseg.)

Medición	0g.	1g.	-1g
Tiempo en bajo	3.5	4.4	2.5
Tiempo en alto (T1)	4.0	3.0	4.9
Periodo	7.4	7.4	7.4
Ciclo de Trabajo (CT)	54.05%	40.54%	66.21%
Variación CT/g	0%	13.51%	12.16%
Variación segundos	0	0.9	1.0

Las resultados de las mediciones realizadas con el osciloscopio (tabla 6.2), mostraron que existe un ruido característico de la señal al pasar de un nivel alto a un nivel bajo, que consiste en la presencia de múltiples cambios de nivel en la transición de un nivel alto a un nivel bajo. El rango de esta variación es de aproximadamente 48-50 μ s como se muestra en la figura 6.11.

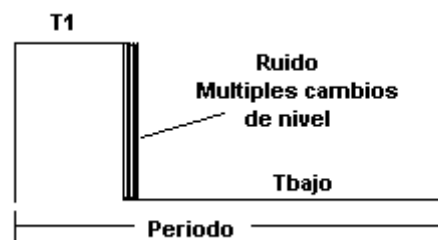


Figura 6.11 Ruido característico en la señal de salida

Se debe considerar que las lecturas realizadas con el osciloscopio tienen además los errores inherentes al ángulo de visión, resolución utilizada y error de la señal.

Debido a la presencia del ruido en la señal, se realizaron 3 programas para el microcontrolador, que midieran: el periodo, el tiempo en bajo y el tiempo en alto de la señal generada para el eje X (tabla 6.3), tomando como referencia el temporizador principal del microcontrolador con un tiempo de incremento de 8 μ s. usando las interrupciones por flanco de subida y bajada. Para tener un valor exacto dentro de la escala del temporizador la cual va de 0 a 65535. Se programaron 170 muestras en cada medición para obtener un valor promedio, ya que el ruido ocasionaba lecturas erróneas. El código fuente de cada programa se anexa en el apéndice correspondiente, y se utilizó la notación especificada en la figura 6.12, para la identificación de los niveles lógicos y periodo de la señal de salida.

Tabla 6.3 Programas realizados para medir los parámetros de la señal de salida

Medición	Nombre del programa
Tiempo en bajo (Tbajo)	t1.asm
Tiempo en alto (T1)	tbajo.asm
Periodo (Periodo)	t2.asm

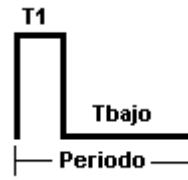


Figura 6.12 Notación usada en la señal de salida

La tabla 6.4, muestra los resultados al realizar las mediciones tomando como referencia el timer interno del microcontrolador con un incremento de $8\mu\text{seg}$. por unidad, con los programas mencionados en la tabla 6.3.

Tabla 6.4 Resultados obtenidos con el microcontrolador (escala del timer)

Medición	0g.	1g.	-1g
Tiempo en bajo	439	555	323
Tiempo en alto	491	375	607
Periodo	930	930	930
Ciclo de Trabajo (CT)	52.79%	40.32%	65.26%
Variación CT/g	0%	12.47%	12.47%
Variación Timer	0	116	116

Se puede observar claramente que la medición hecha con el microcontrolador es más confiable ya que la variación de la aceleración positiva y negativa es la misma, a diferencia de la realizada con el osciloscopio.

Con el análisis de estos resultados se concluyó que:

$$\mathbf{116 \text{ unidades timer} = 1g} \quad \mathbf{(6-5)}$$

Con este valor representado en (ec. 6-5) se obtuvo la referencia, para realizar los cálculos de fuerza del sistema de monitoreo.

6.4 MODULOS DE TRANSMISIÓN Y RECEPCIÓN

El módulo de comunicaciones TWS-RWS esta diseñado para operar de manera sencilla con tan solo introducir los datos de forma serial en uno de los pines del transmisor y recibirlos de igual forma en el receptor, lo que facilita la tarea de transferencia de información de manera inalámbrica entre dispositivos.

6.4.1 DESCRIPCIÓN Y CARACTERÍSTICAS GENERALES

El módulo de comunicaciones empleado, esta compuesto por el transmisor TWS-434 y el receptor RWS-434 de Reynolds Electronics, diseñados para trabajar a una frecuencia de 434.92 MHz, con una velocidad máxima de transmisión de 2400 bits/s. El transmisor tiene una potencia de salida de 8mW lo que permite un alcance aproximado de 120 mts. en campo abierto. La alimentación con la que operan ambos circuitos es de 1.5 V. a 12 V. El tipo de modulación empleada es AM. Las antenas son del tipo monopolo, de $\frac{1}{2}$ de longitud de onda (35 cm).

La figura 6.13 muestra los pines de conexión del modulo de comunicaciones, así como sus dimensiones.

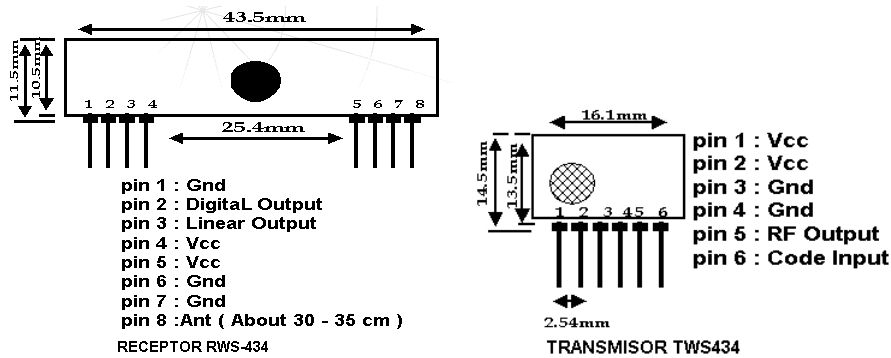


Figura 6.13 Modulo de comunicaciones

6.4.2 TRANSMISIÓN DE INFORMACIÓN.

La información a transmitir, puede ser introducida de manera digital o analógica a través del pin 6 "Code Input" del transmisor. El receptor cuenta con dos salidas, una digital y una analógica, correspondientes al tipo de señal emitida. La figura 6.14 muestra las conexiones necesarias para establecer un enlace unidireccional.

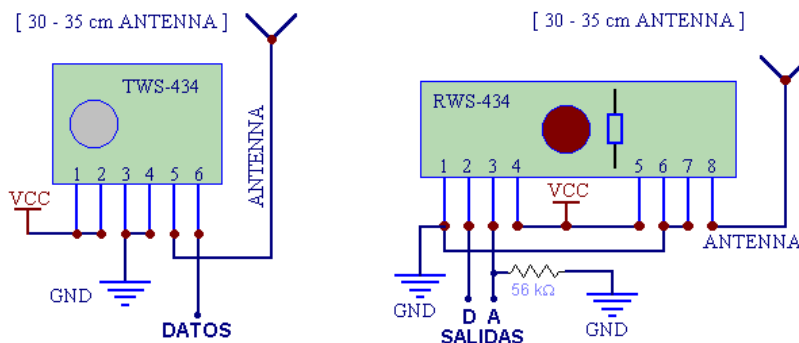


Figura 6.14 Conexionado necesario para la transmisión de la información

En esta fase se realizaron las pruebas correspondientes para determinar el alcance real del módulo, la forma de transmisión y la velocidad máxima del mismo.

En las pruebas preliminares, consistentes en armar en tarjetas de experimentación los circuitos transmisor y receptor del módulo y simplemente switchear manualmente el pin 6 del transmisor entre 5V y 0V. y visualizar los cambios de nivel en el receptor con leds indicadores en ambas salidas, se encontró que el módulo no mantiene un nivel alto en la recepción, aún cuando en la parte transmisora se mantenga dicho nivel, el receptor conserva dicho estado solo unas fracciones de segundo, característica que debió tomarse en cuenta para el posterior diseño del protocolo de transmisión. La distancia máxima que se logro en esta prueba fue de 130 mts.

Posteriormente se armaron y se hicieron pruebas de enlace de los circuitos transmisor y receptor. En el dispositivo transmisor se utilizo la interfase SCI (Serial Communication Interface) integrada en el microcontrolador MC68HC811E2. En el dispositivo receptor se desarrolló un programa especial para el PIC16F84 que pudiera detectar y procesar información procedente del transmisor.

6.4.3 TRANSMISOR

La interfase SCI basa su funcionamiento en el estándar **UART^[6,2] (Universal Asynchronous Receiver Transmitter)**, que permite de manera sencilla el envío y recepción de información de manera serial, cargando y recibiendo cada byte de información en un registro diseñado para tal fin. El formato de envío consiste en un bit de inicio, 8 o 9 bits de información y un bit de parada. La velocidad de transferencia es configurable por el usuario.

El estándar de comunicaciones UART comprende la parte transmisora y receptora, de un sistema capaz de aceptar caracteres provenientes de una fuente digital de información como un microcontrolador o un CPU en formato paralelo y transmitirlos en forma serial, así como de recibir información en forma serial y convertir dicha información a un formato paralelo, estas dos tareas fundamentales pueden ser realizadas simultáneamente debido a que prácticamente las secciones transmisora y receptora de un sistema UART trabajan independientemente. En el microcontrolador MC68HC811E2 estas dos tareas fundamentales pueden realizarse simultáneamente, ya que aunque la interfase SCI tiene un solo registro de corrimiento utilizado por ambas secciones de transmisión y recepción integradas dentro del microcontrolador, el sistema cuenta con una serie de bits de estado (banderas), que permiten saber si se está transmitiendo o recibiendo información, característica que permite saber si el registro de corrimiento esta siendo utilizado, ya sea por la parte receptora o la transmisora y desarrollar la programación adecuada para evitar la pérdida o traslape de información.

Ya que el dispositivo transmisor solo ocuparía la parte transmisora del sistema SCI, únicamente fue necesaria la manipulación de esta parte de la interfase SCI.

6.4.3.1 PRINCIPIO DE OPERACIÓN DEL TRANSMISOR DE LA INTERFASE SCI (SERIAL COMMUNICATION INTERFACE)

El principio de funcionamiento de la **SCI** [6.3], incluye un registro de transmisión de datos en paralelo llamado SCDR y un registro de corrimiento serial. El contenido del registro de corrimiento serial solo puede ser escrito a través del SCDR, la salida de este registro de corrimiento serial es aplicada al pin TxD mientras el bit TE del registro de comunicación serial SCCR2 este habilitado.

La figura 6.15 muestra como se constituye la parte de transmisión del SCI.

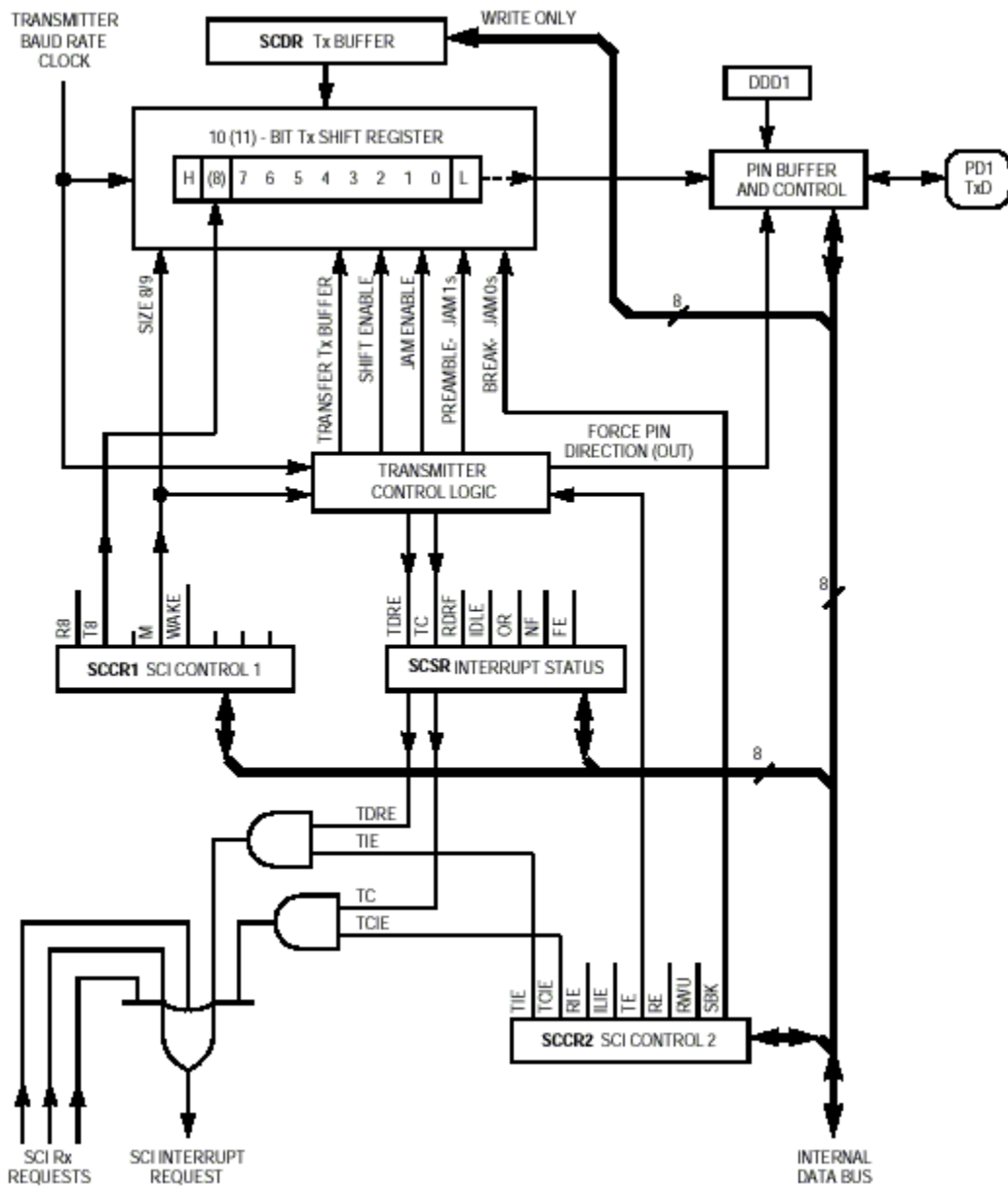


Figura 6.15 Diagrama a bloques del sistema SCI

6.4.3.2 REGISTROS DEL SISTEMA SCI

La interfase SCI trabaja para su configuración y monitoreo con registros especialmente dedicados para tal fin.

Existen cinco registros asociados con el SCI.

- Cuatro registros de control y estado.
 - **SCCR1** Registro de control de comunicación serial 1.
 - **SCCR2** Registro de control de comunicación serial 2.
 - **BAUD** Registro de velocidad de transmisión.
 - **SCSR** Registro de estado de comunicación serial.
- Un registro de datos.
 - **SCDR** Registro de datos de comunicación serial.

SCCR1 SCI Control Register 1. (Registro de control de comunicación serial 1)

En este registro se encuentran los bits de control para determinar el tamaño de palabra a transmitir y seleccionar el método usado para la característica “wakeup”, es el modo en el cual el receptor se queda con las interrupciones inhibidas esperando un evento hardware externo, (asociado a la línea de recepción), que le devuelva al estado activo con interrupciones. El evento externo puede ser de dos tipos y se selecciona con el bit 3 (WAKE). Si WAKE=1 entonces se espera hasta detectar una marca de dirección, si por el contrario WAKE=0 se espera hasta detectar que la línea de recepción esta vacía. El modo WAKE-UP se selecciona mediante software activando el bit 1 del “Registro de control de comunicación serial 2 (RWU)”. Los bits 0,1,2 y 5 no se usan y permanecen siempre a cero.

SCCR1 Dirección: **\$102C**

	Bit 7	6	5	4	3	2	1	Bit 0
	R8	T8	-	M	WAKE	-	-	-
RESET:	1	1	0	0	0	0	0	0

R8.- Bit de recepción (Bit 8).

Si M esta habilitado (1), R8 guarda el noveno bit de un caracter recibido

T8.- Bit de Transmisión (Bit 8).

Si M esta habilitado (1), T8 guarda el noveno bit de un caracter transmitido.

M.- Modo (Selecciona el Formato del Caracter).

0 = Bit de inicio, 8 bits de datos, Bit de parada.

1 = Bit de inicio, 9 bits de datos, Bit de parada.

WAKE.- Modo especial de activación (Detección Marca / Línea Libre.)

0 = Activación cuando se detecta que la línea esta libre.

1 = Activación cuando se detecta una Marca (1), en la línea.

SCCR2 SCI Control Register 2. (Registro de control de comunicación serial 2)**SCCR2** Dirección: \$102D

	Bit 7	6	5	4	3	2	1	Bit 0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:	0	0	0	0	0	0	0	0

TIE.- Interrupción de transmisión.

0 = Interrupción TDRE inhabilitada.

1 = Petición de interrupción del SCI, si bandera de TDRE habilitada.

TCIE.- Interrupción de transmisión completa.

0 = Interrupción de TC inhabilitada.

1 = Petición de interrupción del SCI si bandera TC habilitada.

RIE.- Interrupción de recepción.

0 = Interrupciones RDRF y OR inhabilitadas.

1 = Petición de interrupción del SCI si bandera de RDRF o OR habilitadas.

ILIE.- Interrupción de línea libre.

0 = Interrupción línea libre inhabilitada.

1 = Petición de interrupción del SCI si bandera IDLE habilitada.

TE.- Habilitación de transmisión.

0 = Transmisión inhabilitada

1 = Transmisión habilitada.

RE.- Habilitación habilitada.

0 = Recepción inhabilitada.

1 = Recepción habilitada.

RWU.- Control "Wakeup" de recepción.

0 = Recepción normal.

1 = Wakeup habilitado e interrupciones de recepción inhibidas.

SBK.- Envío código de Break.

Cuando se activa el bit 1, (SBK=1), se mandan señales de BREAK indefinidamente hasta que el desactive. Las señales de BREAK se caracterizan porque se envía todo ceros por la línea serie, no sólo son cero los bits de datos, sino que también se hace cero el bit de stop que siempre vale 1.

0 = Generador de código inhabilitado.

1 = Generador de código habilitado.

BAUD Baud Rate Register (Registro de velocidad de transmisión).

Este registro se utiliza para seleccionar las diferentes velocidades de transmisión del SCI.

BAUD Dirección: \$102B

	Bit 7	6	5	4	3	2	1	Bit 0
	TCLR	-	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET:	0	0	0	0	0	U	U	U

U.- Valor indefinido después del reset.

TCLR.- Modo de prueba.

SCP[1:0].- Divisor del reloj interno.

Los bits (SCP0 y SCP1) determinan la máxima velocidad en baudios. Esta velocidad depende del cristal que se haya conectado al microcontrolador. Lo habitual es colocar un cristal de 8Mhz. Para este cristal se tienen los valores mostrados en la tabla 6.5:

Tabla 6.5 Factor de división del reloj interno

SCP1	SCP0	Factor de división reloj interno	Bauds cristal 8 MHz
0	0	1	125000
0	1	3	41667
1	0	4	31250
1	1	13	9600

RCKB.- Modo de prueba.

SCR[2:0].- Selección de velocidad.

Una vez determinada la velocidad máxima en baudios, con los bits 0,1 y 2 (SCR0, SCR1 y SCR2) divide la velocidad máxima por un valor como se muestra en la tabla 6.6:

Tabla 6.6 Factor de división de velocidad de comunicación

SCR[2:0]	Factor de división.
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

SCSR Serial Communication Status Register (Registro de estado del SCI).

SCSR Dirección: \$102E

	Bit 7	6	5	4	3	2	1	Bit 0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	-
RESET:	1	1	0	0	0	0	0	0

TDRE.- Bandera de registro de transmisión.

0 = Registro SCDR ocupado.

1 = Registro SCDR vacío.

TC.- Bandera de transmisión completada.

6. Desarrollo del Sistema

0 = Transmitiendo
1 = Libre para transmitir.

RDRF.- Bandera del registro de recepción.

0 = Registro SCDR ocupado.
1 = Registro SCDR vacío.

IDLE.- Bandera de detección línea libre.

0 = Línea recepción activa.
1 = Línea recepción libre.

OR.- Bandera de error de traslape.

0 = No errores.
1 = Error detectado.

NF.- Bandera de error por ruido.

0 = Sin ruido.
1 = Ruido detectado.

SCDR Serial Communication Data Register (Registro de datos de comunicación serial)

Este es un registro en paralelo que desarrolla dos funciones. Es el registro de recepción cuando es leído y el registro de transmisión cuando es escrito. La lectura accesa al bufer de recepción de datos y cuando se escribe accesa al bufer de transmisión de datos.

SCDR Dirección: \$102F

	Bit 7	6	5	4	3	2	1	Bit 0
	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	1	1	1	1	1	1	1	1

6.4.3.3 PROGRAMACIÓN DE LA SCI PARA TRANSMISIÓN

Las rutinas de comunicaciones serie pueden seguir dos caminos distintos: Espera Activa o Interrupciones. Al utilizar Espera Activa todas las interrupciones de la SCI están inhibidas. Es el propio software el que debe determinar cuándo se puede enviar un dato y cuándo hay un dato recibido para ser leído. Esto se realiza "explorando" el registro de estado del SCI. Cuando se elige trabajar con las interrupciones el sistema detecta por ejemplo cuando se ha terminado de transmitir y la línea esta libre, o cuando en la línea de recepción se ha detectado que se va a iniciar una transferencia de información, de esta manera el sistema trabaja de forma más eficiente debido a que no pierde tiempo esperando ciertos sucesos que absorben recursos y tiempo del sistema.

Una vez que se definen los parámetros de velocidad, tamaño de palabra y rutina de comunicaciones (activa, interrupción), se deben configurar los registros asociados al la SCI para que la comunicación se lleve a cabo de la manera deseada.

Cabe mencionar que al encender el microcontrolador o después de un reset, en el registro BAUD, los valores de los bits destinados a configurar la velocidad de transferencia, están indefinidos, por lo que al configurar este registro, preferentemente se debe de cargar el valor deseado utilizando los acumuladores A o B para asegurar que el valor de cada bit del registro es el adecuado, en lugar de utilizar la opción BSET para la configuración de dicho registro.

Después de configurar correctamente los registros de la SCI el sistema esta listo para transmitir caracteres simplemente cargando el valor del carácter deseado en el registro **SCDR** dirección: **\$102F** y el sistema inmediatamente comenzará el proceso de envío de información, si se desea enviar la información en un formato de nueve bits, el 9no bit que será el de mayor peso, debe ser cargado en el bit T8 del registro **SCCR1** dirección: **\$102C**, y posteriormente cargar el byte de información en el registro **SCDR**. Es muy importante verificar el registro de estado **SCSR** Dirección: **\$102E** en donde se puede monitorear el bit TC para verificar si la transmisión ha finalizado y el sistema está listo para enviar otro carácter, para evitar que la información se traslape, ya que si se carga el registro **SCDR** con un nuevo dato antes de que haya finalizado la transmisión del byte anterior, la información se traslapará con la consecuente perdida de información: Esta operación debe ser programada por software.

La figura 6.16 muestra procedimiento para evitar traslapos en la información enviada.

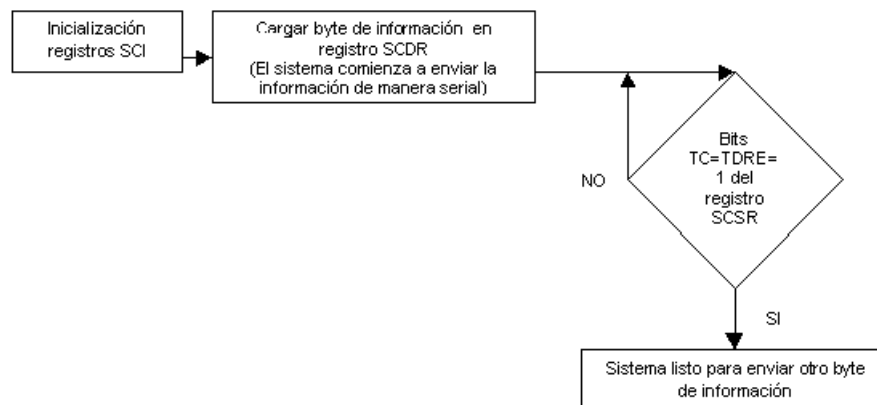


Figura 6.16 Diagrama de flujo de transmisión para evitar traslapos en la información enviada

El formato de transmisión de la SCI esta configurado para enviar de manera serial la siguiente cadena de bits de información (ec. 6-6):

Bit Inicio = 0; 8 ó 9 Bits de información (LSB....MSB); Bit Parada = 1 (6-6)

Debido a que el bit de inicio es un "0", este formato de transmisión contempla que la línea de transmisión mantiene un estado lógico "1" cuando está libre, por lo que el pin TxD debe conectarse con una resistencia de pull-up de 4.7KΩ para lograr dicho estado.

Una característica importante del módulo de comunicaciones es que en el receptor no se mantiene un nivel lógico "1" por mucho tiempo, aunque el transmisor este enviando dicho estado. Debido a esta característica, se tuvo que adecuar el protocolo de comunicaciones empleado por el microcontrolador, ya que el sistema receptor al ser incapaz de mantener un estado lógico de "1", sería imposible detectar el bit de inicio enviado por el transmisor, con la consecuente pérdida de información, por lo que se necesitaba un formato de envío que tuviera como bit de inicio un "1" para que el sistema de recepción fuera capaz de detectarlo y posteriormente realizar el procesamiento necesario para la obtención de los bits subsecuentes, por lo que se debía diseñar un formato de transmisión que cumpliera con la condición antes mencionada

Para no desaprovechar las prestaciones de la SCI del microcontrolador, se diseño un formato de envío que empleara los recursos y facilidad de uso de la misma, de esta forma el formato de envío que se obtuvo con excelentes resultados fue el siguiente (ec. 6-7):

Bit Inicio = 1; 8 Bits de información (LSB....MSB); Bit Parada = 1 (6-7)

Con este formato el receptor al recibir siempre como bit de inicio un "1", el programa en el microcontrolador PIC16F84 seria capaz de detectar el bit de inicio y posteriormente realizar las operaciones necesarias para la obtención de la información.

La manera en como se obtuvo el formato anterior fue configurando la SCI con un formato de envío de 9 bits.

1. Se realiza un corrimiento hacia la izquierda de 1 bit en el byte de información, el bit de mayor peso queda almacenado en el carry del sistema y el valor del bit menos significativo queda con un valor de "0".
2. Posteriormente el valor del carry es cargado al bit T8 del registro **SCCR1** dirección: **\$102C**
3. Después se le suma un "1" al byte de información inicial, el cual tiene por default en la posición del bit menos significativo un "0" después del corrimiento. Al efectuar esta suma el bit menos significativo se carga con el valor "1".
4. Se procede a cargar el registro **SCDR** con el valor del byte inicial. El sistema envía entonces la siguiente cadena (ec. 6-8):

Bit Inicio = 0; 9 Bits de información (LSB....MSB); Bit Parada = 1 (6-8)

el formato equivalente es (ec. 6-9):

Bit Inicio = 0; 1+8 Bits de información (LSB....MSB); Bit Parada = 1 (6-9)

5. De esta manera el formato de envío es detectado en el receptor como (6-10):

Bit Inicio = 1; 8 Bits de información (LSB....MSB); Bit Parada = 1 (6-10)

Porque el bit LSB de la cadena de 9 bits siempre es "1"

La figura 6.17 muestra el diagrama de flujo para la subrutina de transmisión empleada, en donde se incluye la secuencia antes descrita.

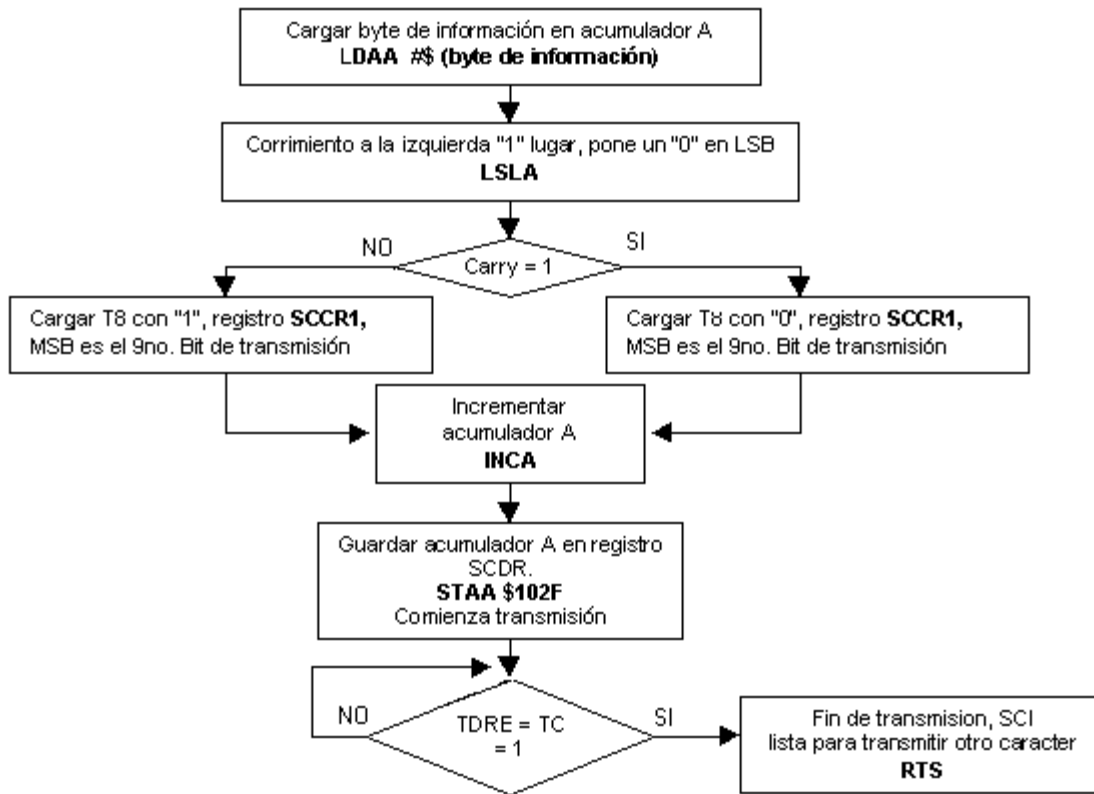


Figura 6.17 Subrutina de transmisión

El dispositivo a bordo del kayak, solo tenía que transmitir la información procesada por el microcontrolador, así la subrutina de transmisión desarrollada fue probada a diferentes velocidades, y se observó que la velocidad máxima de transmisión soportada por el módulo de comunicaciones (2400 bits/s) operó adecuadamente lográndose un enlace satisfactorio a una distancia de 120 mts. sin errores en la recepción.

El módulo de comunicaciones se conectó directamente al pin TxD del microcontrolador.

CODIGO FUENTE DE LA SUBROUTINA DE TRANSMISION Y REGISTROS ASOCIADOS CON LA SCI

*****Registros*****

Idx # $\$1000$

Idaa # $\$32$

staa \$102b ;baud 2400 (registro indefinido al reset)

bset \$2d,x,\$08 ;tx habilitad

bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg

bset \$09,x,\$02 ;port d tx salida

bset \$2c,x,\$10 ;0_9bits_1

```
*****Sub-transmisión*****
tx      ldaa #$12 ;byte de información
        psha
        pshx
        pshb
        ldx #$1000
        lsla ;corrimiento a la izquierda 1 lugar pone un cero en lsb
        bcs uno; si el carry es uno
        bra cero

uno     bset $2c,x,$40
        inca ;sumo 1 a "A" bit de inicio
        bra serie

cero    bclr $2c,x,$40 ;tengo el mayor msb en el bit9
        inca

serie   staa $102f
wait    brclr $2e,x,$c0,wait ;espero a que termine de transmitir
        pulb
        pulx
        pula
        rts
*****fin Sub-transmision*****
```

6.4.4 RECEPTOR

El dispositivo receptor esta compuesto por la parte receptora del modulo de comunicaciones tomando la salida analógica conectada al pin A1 de PIC16F84, se utilizo esta salida, porque los resultados de las diferentes pruebas realizadas fueron mejores en dicha salida.

El formato de datos enviado por el transmisor, es finalmente visto por el receptor de la siguiente forma (ec 6-10):

Bit Inicio = 1; 8 Bits de información (LSB....MSB); Bit Parada = 1

La velocidad de transmisión empleada fue de 2400 bit/s lo que significa que cada bit tiene una duración de 416.66 μ s. Para la detección y procesamiento de la cadena de datos recibida se utilizo el siguiente algoritmo:

1. Esperar hasta detectar un "1" en el pin de recepción (A1),
2. Esperar la mitad de la duración de 1 bit (416.66 / 2 μ s)
3. Verificar nuevamente el valor de ese bit detectado
4. Si es "1", se puede suponer que es un bit de inicio y continuar, en caso contrario regresar al paso 1.
5. Esperar la duración de 1 bit (416.66 μ s)
6. Leer el valor presente en el pin de recepción y tomarlo como valido
7. Realizar 8 veces el paso anterior e ir guardando los valores de los bits leídos tomando en cuenta que el primer bit recibido es el LSB de manera que se debe desarrollar el algoritmo correspondiente para acomodar los bits recibidos en una localidad de memoria de manera adecuada.
8. Esperar la duración de 1 bit (416.66 μ s)
9. Leer el valor presente en el pin de recepción, si el valor leído es un "1" entonces es un bit de parada, lo que significa que la cadena de bits se recibió completa, de otra forma significa que el supuesto bit de inicio detectado al principio del procedimiento, no lo era, si no mas bien era

un bit de datos, por lo que toda la información recibida debe ser desechada y volver al inicio del algoritmo hasta detectar un bit de inicio y parada validos.

La figura 6.18, muestra la forma de adquisición de la información, de acuerdo al algoritmo empleado.

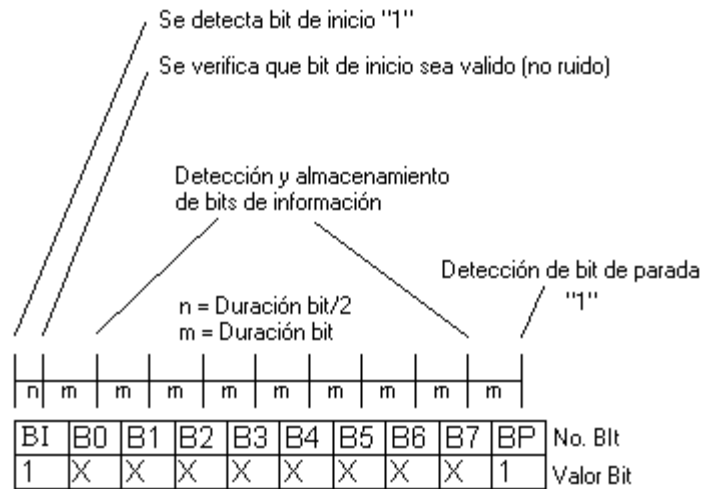


Figura 6.18 Detección de datos seriales

Existen diferentes algoritmos para la obtención de datos seriales, en donde el muestreo de bits se puede hacerse mas de una vez para detectar un bit y validarlo, sin embargo para velocidades de transmisión bajas es suficiente con realizar el muestreo una sola vez. En este caso el algoritmo desarrollado resultó bastante eficiente.

6.4.4.1 CODIGO FUENTE DEL ALGORITMO DE RECEPCIÓN

El algoritmo de recepción fue programado en JAL, cabe mencionar que este lenguaje de programación incluye subrutinas y procesos, que facilitan de manera sustancial el desarrollo de programas de acuerdo a las necesidades de los usuarios, pero debe de tomarse en consideración que algunas de estas subrutinas invocan a otras, lo que se refleja en un incremento en el tiempo de ejecución de las mismas. Una de las subrutinas mas empleadas en esta aplicación fue la llamada "Delay" que permite generar retrasos en intervalos que van desde $1\mu s$ hasta 1 s, usando multiplos de hasta 8 bits, pero al utilizar diferentes multiplos e intervalos, el sistema invoca la subrutina tantas veces como sea el valor del multiplo empleado, por lo que se generan tiempos de retraso debidos a la concatenación de subrutinas, que producen inexactitudes en los retrasos programados, es por eso que los valores de retraso empleados en la programación de esta aplicación, no coinciden con los valores teóricos ($416.66\ \mu s$ y $416.66\ \mu s / 2$). Los valores programados fueron determinados experimentalmente.

La figura 6.19, muestra el diagrama de flujo del algoritmo de recepción, que se empleo para la adquisición de la información.

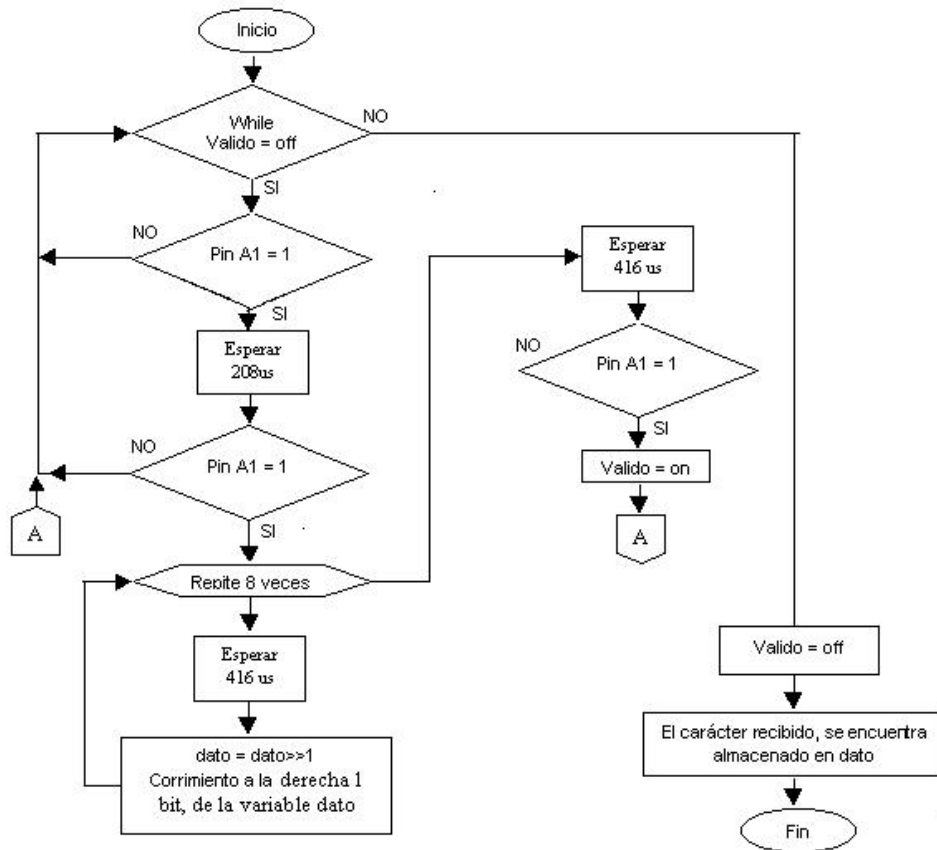


Figura 6.19 Diagrama de flujo del algoritmo de recepción

A continuación se lista el código correspondiente al algoritmo de recepción.

CODIGO FUENTE DEL ALGORITMO DE RECEPCIÓN

```

var bit bit_llegada at dato : 7 -- defino variable bit de llegada
    -- en el bit 7 de variable dato
-- *****inicio a recibir caracter*****
while valido == off loop
    pin_a0 = off

    if pin_a1 == on then -- si llega un 1 checa si es bit de inicio
        delay_1us ( 197 )
        if pin_a1 == on then -- es el bit de inicio valido

            for 8 loop -- comienzo a leer y almacenar bits informacion
                delay_1us ( 255 ) -- delay aprox de 416µs
                delay_1us ( 139 )
                dato = dato >> 1 -- corrimiento hacia la derecha de variable dato
                bit_llegada = pin_a1 -- bit en pin A1 se guarda
                -- en bit7 de variable dato
            end loop -- fin lectura bits información

            delay_1us ( 255 ) -- delay aprox 416 µs
            delay_1us ( 139 )

```

```
if pin_a1 == on then -- Checo bit de parada valido
    valido = on -- la cadena recibida es correcta
    pin_a0 = on -- enciende led de cadena correcta
```

```
end if
```

```
end if -- fin del bit de inicio valido
```

```
end if -- fin del bit de inicio
```

```
end loop
```

```
valido = off
```

```
■ ***** fin de recepcion de caracter *****
```

En resumen, en esta etapa se desarrollaron los algoritmos de transmisión y recepción, las pruebas de alcance máximo del módulo y la adecuación del protocolo de comunicaciones del sistema UART, para la implementación de la transferencia de información entre los dispositivos transmisor y receptor del sistema de monitoreo desarrollado.

6.5 INTEGRACIÓN DEL SISTEMA DE MONITOREO

En esta etapa se integraron todos los algoritmos y elementos del sistema de monitoreo, para su desarrollo final de acuerdo al diseño y características previamente definidas, en el Capítulo 5. La disposición física de los dispositivos se realizó conforme a las recomendaciones de ensamble de los fabricantes conectando las terminales de los dispositivos tal y como se muestran en los diagramas eléctricos.

Para el “Dispositivo Sensor-Transmisor”, fue necesario el análisis del movimiento del Kayak, con la consecuente necesidad de desarrollar un algoritmo capaz de procesar la información proveniente del acelerómetro, analizando la variación de la aceleración cuando el kayak se encuentra en movimiento y finalmente integrar este análisis en la programación.

Para el “Dispositivo Receptor”, la tarea más importante fue la de la correcta recepción de la información, además del despliegue y almacenamiento de la información.

Para ambos “Dispositivos Transmisor y Receptor”, se realizó una exhaustiva experimentación del protocolo de comunicaciones desarrollado para lograr una correcta y satisfactoria transferencia y recepción de la información.

Para la integración del dispositivo transmisor se siguieron los siguientes pasos:

- 1.- Análisis del movimiento del kayak y el comportamiento de la señal de salida al experimentar dicho movimiento, para definir la manera en que se procesaría dicha señal (6.5.1).
- 2.- Cálculo de la fuerza partiendo de la ec. 5-5 obtenida en 5.2.1, adecuando dicho cálculo para el microcontrolador (6.5.1.1)
- 3.- Análisis y cálculo de las B.P.M. y la implementación para el microcontrolador (6.5.1.2).
- 4.- El análisis de la transmisión de la información (6.5.1.3).
- 5.- Características de funcionalidad del transmisor (6.5.1.4)
- 6.- Finalmente al tener definido perfectamente los parámetros y características del transmisor, se procedió a la programación del mismo, haciendo uso de los algoritmos y subrutinas previamente desarrollados (6.5.1.5)
- 7.- La integración física de los dispositivos, se muestra en el diagrama eléctrico (6.5.1.6), de acuerdo a las pruebas desarrolladas previamente para cada dispositivo.

Para la integración del dispositivo receptor se siguieron los siguientes pasos:

- 1.- La principal función del receptor es la de adquirir la información proveniente del transmisor para desplegarla en el display, las subrutinas del display y de recepción, se desarrollaron en los apartados 5.5 y 5.2.1 respectivamente, por lo que únicamente se unieron en un solo programa, para programarse en el PIC16F84.
- 2.- Se incluyeron las subrutinas de captura y despliegue de memorias en el programa para realizar dicha función (6.5.2.2).
- 3.- La integración física de los dispositivos, se muestra en el diagrama eléctrico (6.5.2.3), de acuerdo a las pruebas desarrolladas previamente para cada dispositivo

Cabe mencionar, que el funcionamiento y experimentación de cada elemento electrónico fueron analizadas y probadas previamente, como se vio en los apartados de los capítulos 5 y 6, y en este apartado solo se integraron físicamente como se muestra en los diagramas eléctricos de las figuras (6.5.1.6 y 6.5.2.3) y a nivel lógico en un solo programa (apéndices correspondientes).

6.5.1 INTEGRACIÓN DEL DISPOSITIVO TRANSMISOR

El movimiento del kayak al desplazarse hacia adelante es caracterizado por experimentar una aceleración positiva con cada palada que el remero realiza, y posteriormente una aceleración negativa (desaceleración), en cada cambio de palada cuando la pala no tiene contacto alguno con el agua.

La figura 6.20, muestra las etapas del movimiento descrito.



ATLETA REMANDO, EJEMPLIFICACION DE LA PALADA

APLICACION DE LA PALADA, ACELERACION POSITIVA (IZQUIERDA)
LA PALA NO TOCA EL AGUA, DESACELERACION (ARRIBA)

Figura 6.20 Ejemplificación de la palada

El acelerómetro genera una señal con ciclo de trabajo proporcional a la aceleración; cuando el acelerómetro no se encuentra sometido a aceleración alguna, la señal generada tiene un ciclo de trabajo del 50%. Al desplazar el acelerómetro en un sentido positivo (eje +X), la aceleración producida ocasiona que el ciclo de trabajo decremente y cuando el acelerómetro experimenta un movimiento en el sentido negativo (eje -X), el ciclo de trabajo aumenta, como se muestra en la figura 6.21.



Figura 6.21 Variaciones del ciclo de trabajo debidas a las variaciones de la aceleración

Este comportamiento de la aceleración, ocasiona en la señal de salida un aumento y disminución en el ciclo de trabajo. Por tanto, el algoritmo desarrollado para el tratamiento de la variación del ciclo de trabajo, tenía que ser capaz no de detectar el incremento de la aceleración con respecto del punto de partida correspondiente al valor inicial de aceleración cero (CT=50%), que sería lo conveniente para medir una aceleración máxima experimentada; si no que debía medir los incrementos relativos

con respecto de los máximos y mínimos relativos del ciclo de trabajo de la señal, ya que el dispositivo, al experimentar una aceleración positiva correspondiente a una disminución del ciclo de trabajo y posteriormente experimentar una desaceleración (aceleración negativa) correspondiente a un incremento en el ciclo de trabajo, se debía de tomar como punto de referencia (equivalente a la aceleración cero), el cambio entre una aceleración positiva y una aceleración negativa y viceversa, para poder realizar los cálculos correspondientes.

El ciclo de trabajo de una señal cuadrada esta definido como se muestra en la ec 6-11:

$$C.T. = (\text{Tiempo en alto} / \text{Periodo}) \times 100 \quad (6-11)$$

Por lo que, por facilidad y congruencia entre datos, se eligió utilizar para la medición, el tiempo en bajo (**nivel lógico "0"**) de la señal, ya que un incremento en la aceleración, corresponde a un decremento en el ciclo de trabajo, con el implícito incremento en el tiempo en bajo de la señal como se muestra en la figura 6.21. De esta forma existe una proporcionalidad directa entre tiempo en bajo de la señal (**nivel lógico "0"**) y aceleración.

El nivel lógico "0" (**Tbajo**), se midió en unidades del Timer del microcontrolador, por medio de interrupciones de flanco con un incremento programado entre cada unidad de $8\mu\text{seg.}$, comenzando la medición al detectar un flanco de bajada y posteriormente finalizarla al detectar un flanco de subida en la señal del acelerómetro y simplemente realizar una resta entre estos valores, para obtener el valor del nivel lógico "0" en unidades del timer, como se muestra en la ec. 6-12.

$$Tbajo = (\text{Valor Timer en flanco de bajada}) - (\text{Valor Timer en flanco de subida}) \quad (6-12)$$

De esta forma con cada periodo de la señal se obtiene un valor "**Tbajo**" cuyo valor depende de la aceleración a la que se ve sometido el acelerómetro.

El movimiento del kayak ocasiona variaciones en "**Tbajo**", que producen que este parámetro oscile entre valores máximos y mínimos relativos dependiendo de las variaciones en la aceleración en sentido positivo y negativo del eje X del acelerómetro que es el eje utilizado en esta aplicación.

La figura 6.22, muestra el comportamiento del **nivel lógico "0"**, al experimentar aceleraciones positivas y negativas. El valor **Vacel**, muestra un incremento positivo en la aceleración equivalente al desplazamiento del kayak hacia delante cada vez que el atleta aplica un remada, y al cambiar de lado de remada, se produce una desaceleración, lo que va generando en la señal máximos y mínimos relativos, los cuales tuvieron que ser tomados en cuenta para realizar los cálculos correspondientes de fuerza y BPM. En momentos en los que la embarcación se ve sometida a vibraciones, la señal de salida refleja una serie de incrementos y decrementos muy variados y rápidos, por lo que también se tuvo que implementar una rutina que detectara y solucionara el problema de las vibraciones, para que no afectara en los resultados de los cálculos de fuerza y BPM. La figura 6.21(a) muestra las variaciones del ciclo de trabajo debidas a aceleración experimentada.



Figura 6.21(a) Variaciones del ciclo de trabajo debidas a las variaciones de la aceleración

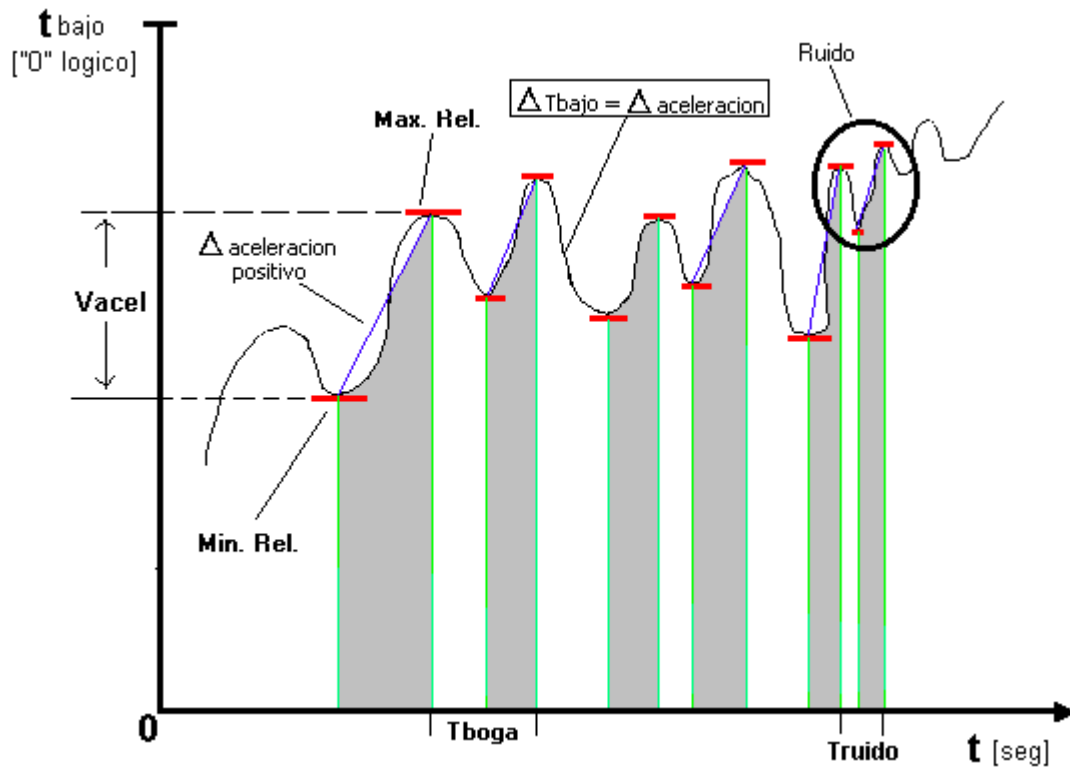


Figura 6.22 Comportamiento del nivel lógico “0” de la señal de salida, cuando el kayak se desplaza hacia adelante

- Max. Rel.-** Máximo relativo del valor de la aceleración, equivalente a un máximo relativo del nivel lógico “0” (T_{bajo}) de la señal.
- Min. Rel.-** Mínimo relativo del valor de la aceleración, equivalente a un mínimo relativo del nivel lógico “0” (T_{bajo}) de la señal.
- Vacel.-** Es la diferencia de duración del nivel lógico “0” de la señal, entre un máximo y mínimo relativos, correspondiente a un incremento positivo de la aceleración.
- Tboga.-** Tiempo entre dos máximos relativos, equivalentes al tiempo en realizarse una boga (una remada).
- Truido.-** Tiempo entre dos máximos relativos, debido a vibraciones detectadas por el acelerómetro, considerado ruido, por ser de muy poca duración.

El nivel lógico “0”, es proporcional a las variaciones de aceleración, por lo que se puede apreciar en la gráfica que se presentan máximos y mínimos relativos, debidos a la variación de la aceleración, por lo que no es válido tomar como referencia lo mostrado por la ec. 6-13:

$$\text{C.T.} = 50\% \rightarrow a = 0g \quad (6-13)$$

Porque para este caso cada mínimo relativo debe ser tomado como punto de referencia para el cálculo de la aceleración total experimentada y calcular el valor de **Vacel**, ya que el desplazamiento del kayak es representado por este parámetro que va “De un mínimo relativo al máximo relativo siguiente”, y el tiempo transcurrido entre dichos máximos relativos, es utilizado para calcular las BPM, como se muestra en la figura 6.22 el parámetro **Tboga**.

Por tanto los parámetros a medir en el movimiento del kayak son:

- El incremento positivo de la aceleración (“Vacel”), para posteriormente calcular la fuerza.
- El tiempo transcurrido entre máximos relativos mínimo y máximo (“Tboga”), para el cálculo de las BPM.

Los incrementos excesivamente rápidos deben ser interpretados como ruido ya que corresponden a vibraciones, que para esta aplicación deben ser despreciadas debido a que el valor máximo de bogas por minuto que un remero puede realizar, oscila entre 140 y 160 BPM.

Utilizando los valores obtenidos experimentalmente, el valor del incremento de la aceleración definido por la ec. 6-14, se obtiene como se muestra en la ec. 6-15 para obtener “Vacel”:

$$\Delta a = \Delta T_{\text{bajo}} = \Delta \text{nivel lógico "0"} \quad (6-14)$$

$$V_{\text{acel}} = \frac{T_{\text{bmax}} - T_{\text{bmin}}}{116} [g] \quad (6-15)$$

donde:

Δa : Incremento de aceleración.

ΔT_{bajo} : Incremento del nivel lógico “0”.

T_{bmax} : Valor del nivel lógico “0” (T_{bajo}) en unidades timer, en el nivel máximo relativo (Max. Rel.).

T_{bmin} : Valor del nivel lógico “0” (T_{bajo}) en unidades timer, en el nivel mínimo relativo (“Min. Rel.”)

V_{acel} : Valor del incremento de aceleración, en unidades “g”; $1g = (9.81 \text{ m/s}^2)$.

Nota: $116 [\text{unidades timer}] = 1g = (9.81 [\text{m/s}^2])$, valor obtenido experimentalmente (ec. 6-5)

6.5.1.1 CALCULO DE LA FUERZA

Para esta aplicación la fuerza que aplica un remero en cada palada debe ser medida en unidades de “Kilogramos Fuerza, kg_F ” ^[6.4], ya que esta unidad corresponde al peso de los cuerpos, unidad utilizada en los entrenamientos así como en la vida diaria.

Una vez obtenido el valor del incremento de la aceleración “Vacel”, para el cálculo de la fuerza se utilizó la expresión (ec 6-16), de la definición de Peso ^[6.5]:

$$P = 1kg \times 9.81(\text{m/s}^2) = 9.81 \text{ Newtons} = 1kg_F \quad (6-16)$$

Por lo que un cuerpo con masa de 80 kg tiene un peso de $80kg_F$, entonces se tiene que la fuerza aplicada es como se muestra en ec. 6-17:

$$F = mV_{\text{acel}}[kg_F] = m \frac{T_{\text{bmax}} - T_{\text{bmin}}}{116} [kg_F] = \frac{m(T_{\text{bmax}} - T_{\text{bmin}})}{116} [kg_F] \quad (6-17)$$

con la expresión anterior (6-17), se realizó el calculo de la fuerza con una masa de 80 kg, que es un valor promedio de la suma de las masa de un remero y su kayak.

6.5.1.2 CALCULO DE LAS BPM

Para el cálculo de las “Bogas por minuto” se desarrollo un algoritmo que incrementa una cuenta llamada BPM cada que se detecta un flanco de bajada, de esta forma la cuenta BPM se incrementa en una unidad en cada periodo de la señal y cada que se detecta un máximo relativo en el nivel lógico “0”, se guarda dicho valor para realizar los cálculos de bogas por minuto y posteriormente se reinicia la cuenta (BPM=0).

El periodo de la señal proveniente del acelerómetro en términos de unidades del timer el cual es equivalente a $8\mu\text{s}$ por unidad es (ec. 6-18):

$$T_{\text{señal}} = 930 \text{ unidades timer} \quad (6-18)$$

Por tanto la frecuencia de la señal que el microcontrolador detecta es (ec. 6-19):

$$F = 1 / (930 \times 8\mu\text{s}) = 134.4 \text{ Hz} \quad (6-19)$$

Por lo que se deduce que en un minuto la cuenta BPM como máximo es de (ec. 6-20):

$$\text{BPM}_{\text{max}} = 134.4 \times 60 = 8064 \quad (6-20)$$

Este valor es prácticamente imposible de realizar por un atleta, sin embargo esto indica que el dispositivo puede detectar vibraciones con frecuencias iguales o menores a 134.4 Hz. Finalmente la expresión, para el cálculo de las la Bogas por minuto validas “ BPM_{val} ” realizadas por el atleta, la cual es calculada en cada palada esta representada en la ec. 6-21 :

$$\text{BPM}_{\text{val}} = \text{BPM}_{\text{max}} / \text{BPM} \quad (6-21)$$

Como ya se menciona anteriormente el dispositivo puede calcular valores en un rango de 0 a 8040 bogas por minuto, por tanto valores de BPM_{val} mayores a 160, deben ser considerados como ruido (vibraciones) en esta aplicación, ya que es humanamente imposible remar a valores superiores a 160 BPM_{val} , por lo tanto los valores límite de estos dos parámetros son los expresados en ec. 6-22 y ec 6-23:

$$\text{BPM}_{\text{val}} \leq 160 \quad (6-22)$$

$$\text{BPM} \geq 50.4 \quad (6-23)$$

La forma en como se determina el valor de las BPM_{val} , teóricamente es el más exacto, sin embargo para esta aplicación tal exactitud no es requerida, porque una variación en la cuenta BPM de 10 unidades equivalentes a 74.4 mseg., produce una variación en BPM_{val} de 30 unidades, por lo que para esta aplicación dicha variacion en BPM_{val} no es útil para el análisis del desempeño de un atleta. Este tipo de variaciones se presenta en cada una de las paladas, debido a que el cuerpo humano no es capaz de realizar movimientos repetitivos con la misma periodicidad. La solución a este problema fue promediar los valores obtenidos 4 veces.

6.5.1.3 TRANSMISIÓN DE LA INFORMACIÓN

La transmisión de la información se realizó de acuerdo al protocolo previamente desarrollado, con la particularidad de enviar 3 veces el carácter de inicio “B”, seguido de los 6 caracteres de información y un caracter de parada ”F”. La cadena se envía 3 veces para lograr una mejor adquisición del receptor, lo cual se perfecciono experimentalmente, hasta llegar a lo anteriormente descrito.

6.5.1.4 CARACTERÍSTICAS DE FUNCIONAMIENTO DEL DISPOSITIVO TRANSMISOR

El diseño final del transmisor integra además del hardware contemplado para el correcto procesamiento, despliegue y transmisión de la información, los algoritmos y conceptos anteriormente desarrollados, con las siguientes características:

- Cálculo y visualización de la fuerza aplicada cada 4 paladas.
- Cálculo y visualización de las bogas por minuto cada 4 paladas.
- Despliegue de un cronómetro de referencia
- Transmisión de los valores de fuerza y bogas por minuto.

6.5.1.5 PROGRAMACIÓN DEL TRANSMISOR

El programa desarrollado para el transmisor en donde se implementan todos los conceptos anteriormente descritos, se realizó utilizando subrutinas e interrupciones en el microcontrolador, por lo cual, existe un programa principal que invoca subrutinas y procesa la información proveniente de las interrupciones. En el apéndice A1 se muestran los diagramas de flujo etiquetados con el nombre de la subrutina e interrupción correspondientes, así como el diagrama de flujo del programa principal llamado simplemente, "Programa".

El código fuente del transmisor (apéndice correspondiente), detalla cada procedimiento realizado.

6.5.1.6 DIAGRAMA ELÉCTRICO DEL TRANSMISOR

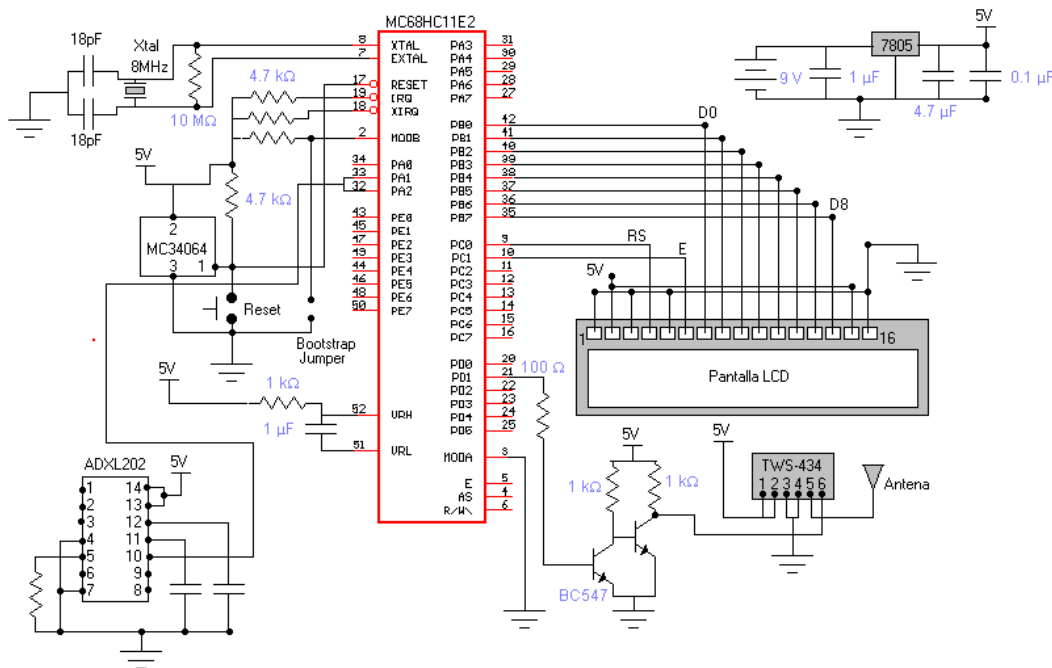


Figura 6.23 Diagrama eléctrico del Transmisor

6.5.2 INTEGRACIÓN DEL RECEPTOR

La función principal del dispositivo receptor, es el procesamiento de la señal recibida proveniente del Dispositivo Transmisor para su visualización y almacenamiento de dicha información en memorias que posteriormente pueden ser visualizadas.

El protocolo de comunicaciones usado, el cual esta basado en el protocolo UART, desarrollado en la etapa correspondiente a las comunicaciones, es la pieza fundamental para el funcionamiento del receptor. Las pruebas correspondientes, dieron como resultado que la cadena de caracteres a enviar se recibía mejor y a mayor distancia, si el carácter de inicio se enviaba 3 veces, y la cadena de datos 3 veces, ya que se logra una mejor sincronía y por lo tanto una correcta detección de la información, para su procesamiento. Una vez que la información es detectada el dispositivo tiene la posibilidad de almacenar hasta 5 memorias, las cuales pueden ser borradas y vueltas a cargar durante el proceso de adquisición.

6.5.2.1 PROGRAMACIÓN DEL RECEPTOR

El programa desarrollado para el receptor, invoca subrutinas ya incluidas en el lenguaje utilizado (JAL), con ligeras modificaciones realizadas para conectar el Display.

En el apéndice A3 se encuentra el diagrama de flujo y en el apéndice A4 el código fuente, donde se detalla cada procedimiento.

6.5.2.2 CARACTERÍSTICAS DE FUNCIONAMIENTO DEL RECEPTOR

- Visualización de la fuerza aplicada en cada palada.
- Visualización de las bogas por minuto cada 4 paladas.
- Capacidad de almacenamiento en memoria de hasta 5 valores de bogas por minuto y fuerza
- Visualización de las memorias en pantalla simultáneamente con los datos adquiridos.

6.5.2.3 DIAGRAMA ELECTRICO DEL RECEPTOR

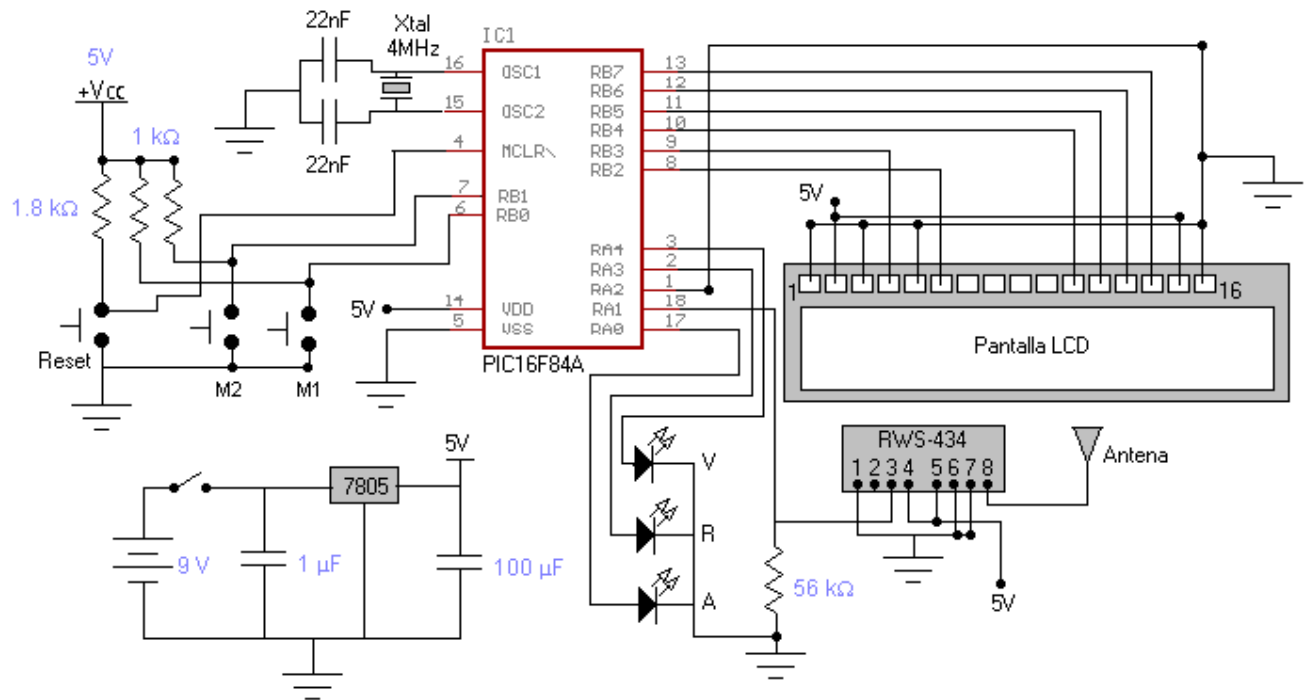


Figura 6.24 Diagrama eléctrico del receptor

7. PRUEBAS Y RESULTADOS

Una vez integrado y armado el sistema de monitoreo, se procedió a probarlo a bordo de un kayak, con el objeto de medir su desempeño y perfeccionarlo.

Las pruebas realizadas, consistentes en la medición de los parámetros del sistema, fueron las siguientes:

- Medición de las bogas por minuto y fuerza aplicada
- Distancia máxima de recepción y pérdida de datos

Las pruebas fueron desarrolladas en diferentes intervalos de tiempo, así como con diferentes condiciones ambientales.

Una vez realizadas las pruebas, se procedió al análisis de la información obtenida para mejorar las deficiencias del sistema.

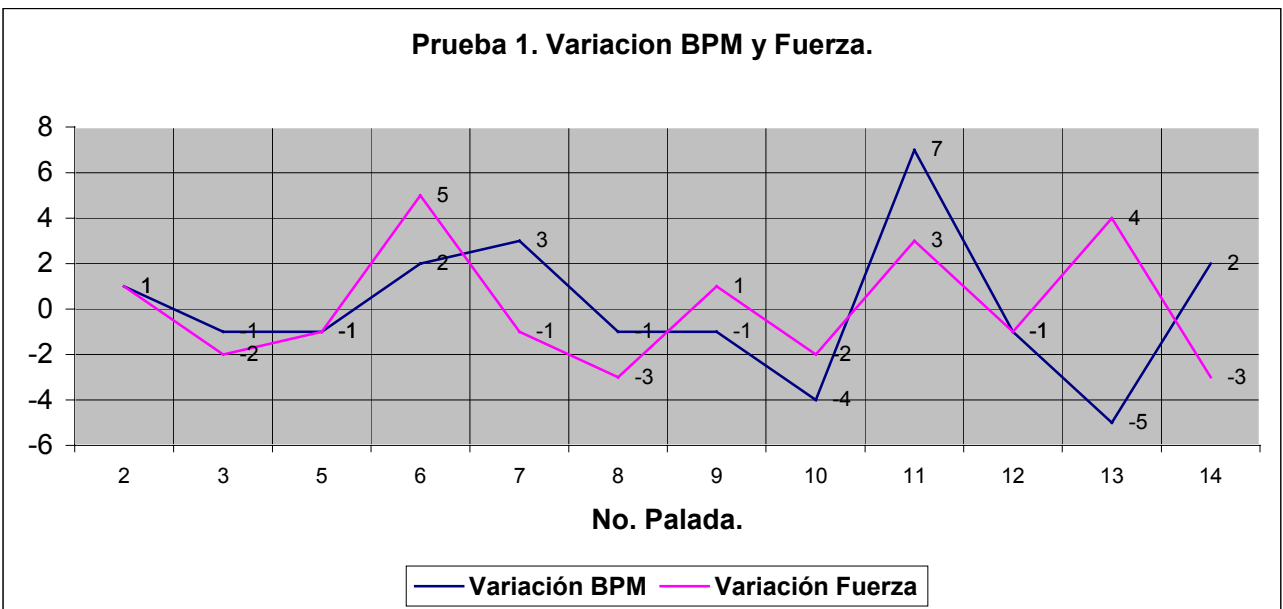
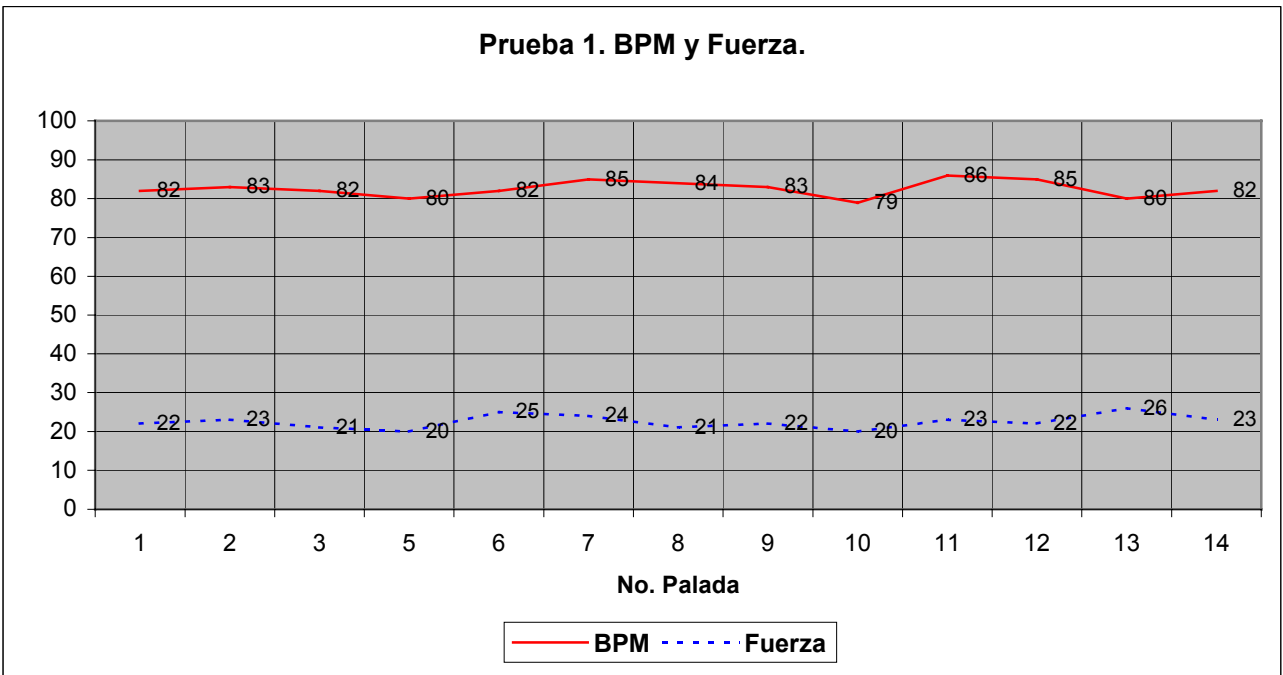
7.1 MEDICIÓN DE LAS BOGAS POR MINUTO Y FUERZA APLICADA.

Para la medición de estos parámetros, se realizaron diferentes pruebas consistentes en la medición del número de paladas y el valor de fuerza en cada palada, en diferentes intervalos de tiempo, con un atleta remando a un ritmo e intensidad constantes (el atleta llevaba como referencia un marcapasos comercial). Se planearon las pruebas de esa forma (ritmo e intensidad constantes), para evaluar la funcionalidad práctica del sistema, ya que de esta forma la información obtenida nos permitiría evaluar de una manera cualitativa y cuantitativa los valores calculados por el sistema, esperando obtener valores de los parámetros dentro de un rango de variación acotado, permitiendo evaluar las deficiencias del sistema para perfeccionarlo, por lo que no fue necesario realizar pruebas con cambios de ritmo e intensidad ya que pruebas de este tipo arrojarían resultados con variaciones considerables no útiles para la evaluación del sistema. Los diferentes intervalos de duración de las pruebas son con el objeto de compensar las variaciones inherentes en los parámetros a evaluar, ocasionadas por el cansancio del atleta, ya que entre mas duración tenga una prueba, son mayores las variaciones en ritmo e intensidad.

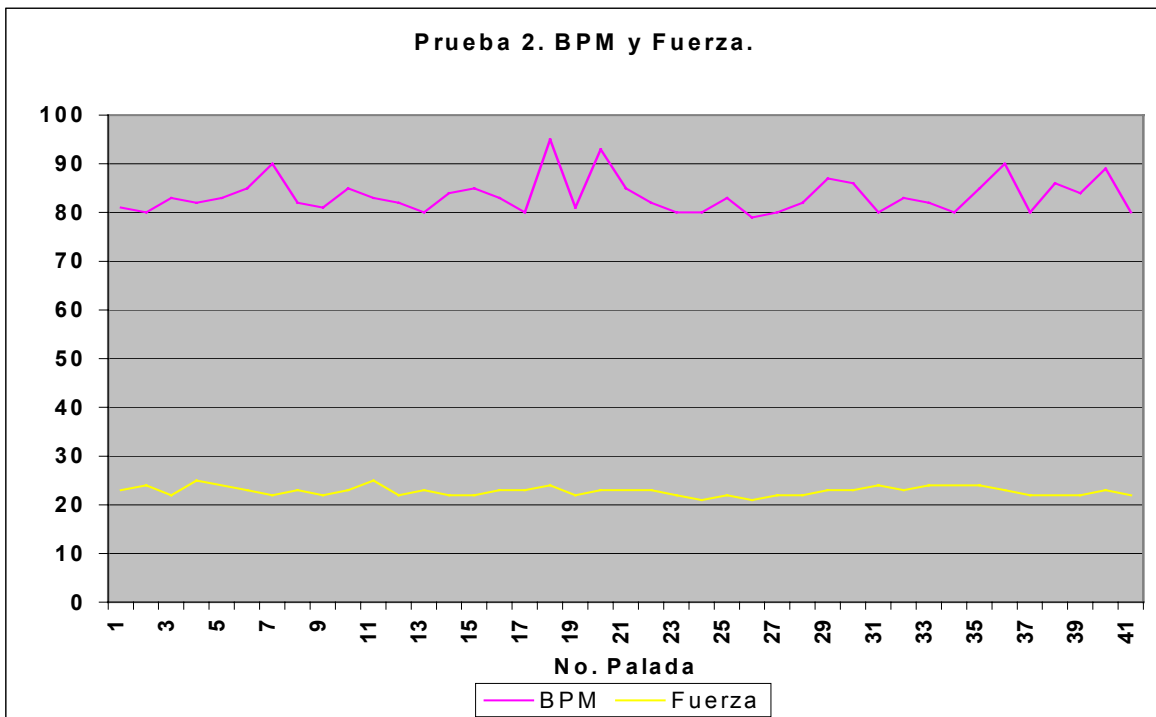
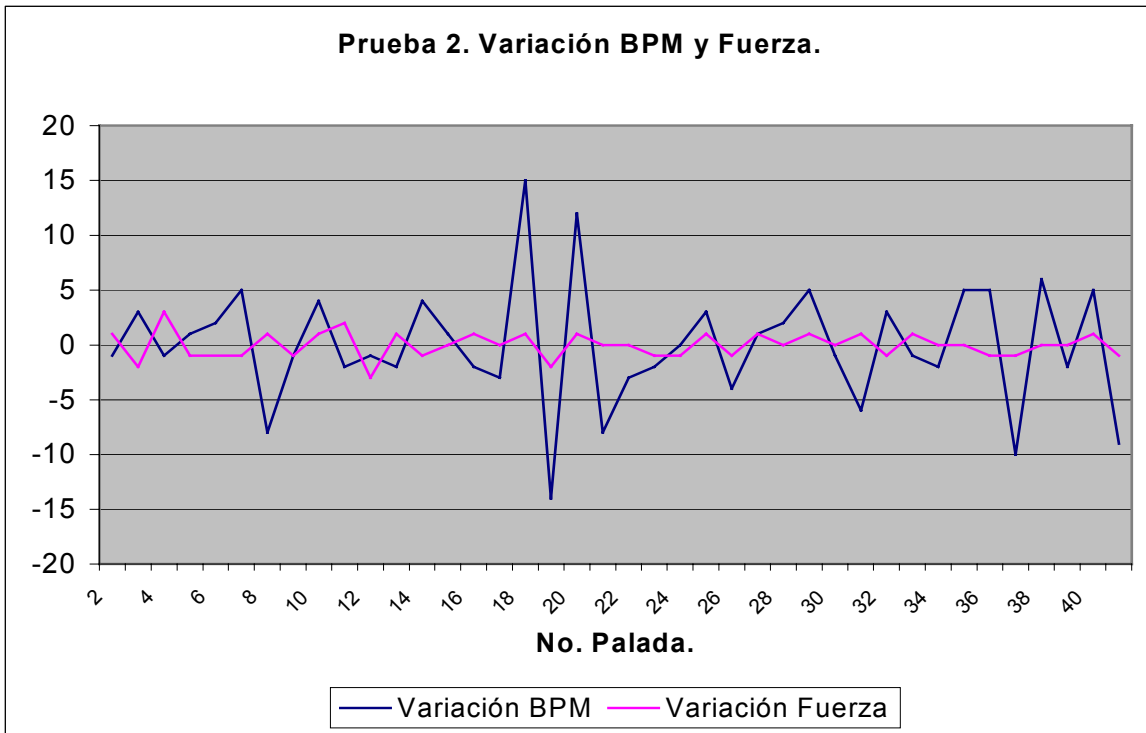
En las tablas y gráficas siguientes se muestran los resultados de dichas pruebas, cada una de las cuales tienen los siguientes campos:

- PALADA.- Número de palada muestreada.
- BPM.- Resultado obtenido de BPM.
- FUERZA.- Resultado obtenido de Fuerza.
- VARIACIÓN ENTRE PALADAS.- Valor calculado de la variación de tiempo entre una palada y otra.
- VARIACION BPM.- Valor calculado de la variación de las BPM entre una palada y otra.
- VARIACION FUERZA.- Valor calculado de la variación de la fuerza entre una palada y otra.
- PROMEDIOS BPM Y FUERZA.- Valores calculados del promedio de BPM y Fuerza
- CALCULO BPM.- Valor calculado de las BPM, que realizaría el atleta tomando como referencia el número de paladas adquiridas en el intervalo de tiempo de la prueba.

PRUEBA 1					
Intensidad: 80%					
BPM:80					
Duración:10 seg.					
Palada	BPM	Fuerza	Variación entre paladas [mseg.]	Variación BPM	Variación Fuerza
1	82	22			
2	83	23	-8.82	1	1
3	82	21	8.82	-1	-2
5	80	20	9.26	-1	-1
6	82	25	-18.29	2	5
7	85	24	-25.82	3	-1
8	84	21	8.40	-1	-3
9	83	22	8.61	-1	1
10	79	20	36.60	-4	-2
11	86	23	-61.82	7	3
12	85	22	8.21	-1	-1
13	80	26	44.12	-5	4
14	82	23	-18.29	2	-3
Promedio BPM: 82.43					
Promedio Fza: 22.36					
Cálculo BPM: 84.00					

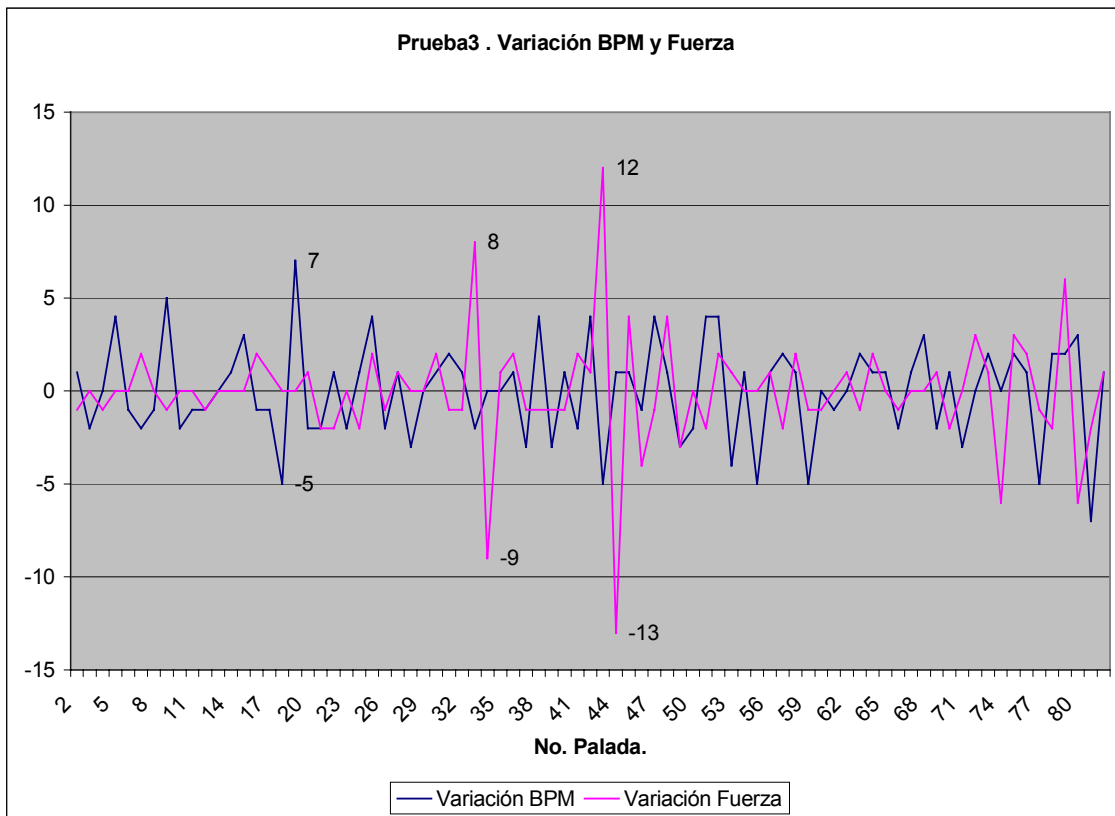
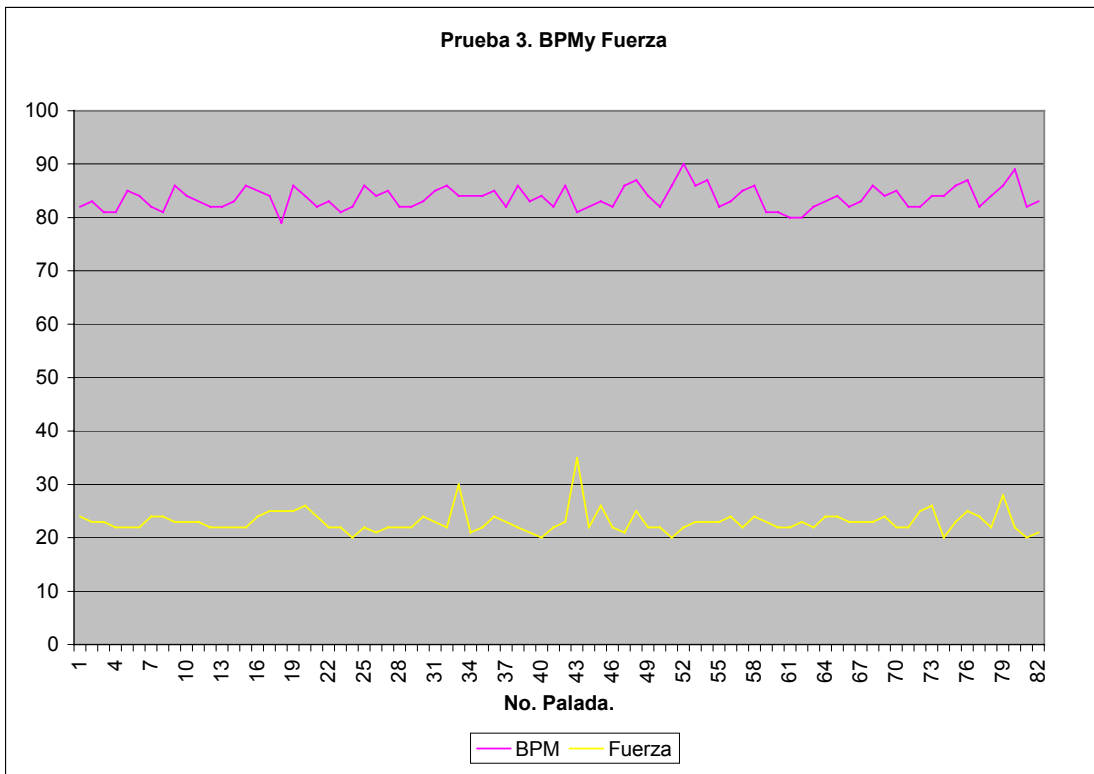


PRUEBA 2					
Intensidad:					
BPM					
Duración:30					
Palada	BPM	Fuerza	Variación entre paladas [mseg.]	Variación BPM	Variación Fuerza
1	81	23			
2	80	24	9.26	-1	1
3	83	22	-27.11	3	-2
4	82	25	8.82	-1	3
5	83	24	-8.82	1	-1
6	85	23	-17.01	2	-1
7	90	22	-39.22	5	-1
8	82	23	65.04	-8	1
9	81	22	9.03	-1	-1
10	85	23	-34.86	4	1
11	83	25	17.01	-2	2
12	82	22	8.82	-1	-3
13	80	23	18.29	-2	1
14	84	22	-35.71	4	-1
15	85	22	-8.40	1	0
16	83	23	17.01	-2	1
17	80	23	27.11	-3	0
18	95	24	-118.42	15	1
19	81	22	109.16	-14	-2
20	93	23	-95.58	12	1
21	85	23	60.72	-8	0
22	82	23	25.82	-3	0
23	80	22	18.29	-2	-1
24	80	21	0.00	0	-1
25	83	22	-27.11	3	1
26	79	21	36.60	-4	-1
27	80	22	-9.49	1	1
28	82	22	-18.29	2	0
29	87	23	-42.05	5	1
30	86	23	8.02	-1	0
31	80	24	52.33	-6	1
32	83	23	-27.11	3	-1
33	82	24	8.82	-1	1
34	80	24	18.29	-2	0
35	85	24	-44.12	5	0
36	90	23	-39.22	5	-1
37	80	22	83.33	-10	-1
38	86	22	-52.33	6	0
39	84	22	16.61	-2	0
40	89	23	-40.13	5	1
41	80	22	75.84	-9	-1
Promedio	83.44				
Promedio	22.80				
Cálculo	82.00				



PRUEBA 3					
Intensidad: 80%					
BPM:80					
Duración:60 seg.					
Palada	BPM	Fuerza	Variación entre paladas [mseg.]	Variación BPM	Variación Fuerza
1	82	24			
2	83	23	-8.82	1	-1
3	81	23	17.85	-2	0
4	81	22	0.00	0	-1
5	85	22	-34.86	4	0
6	84	22	8.40	-1	0
7	82	24	17.42	-2	2
8	81	24	9.03	-1	0
9	86	23	-43.07	5	-1
10	84	23	16.61	-2	0
11	83	23	8.61	-1	0
12	82	22	8.82	-1	-1
13	82	22	0.00	0	0
14	83	22	-8.82	1	0
15	86	22	-25.22	3	0
16	85	24	8.21	-1	2
17	84	25	8.40	-1	1
18	79	25	45.21	-5	0
19	86	25	-61.82	7	0
20	84	26	16.61	-2	1
21	82	24	17.42	-2	-2
22	83	22	-8.82	1	-2
23	81	22	17.85	-2	0
24	82	20	-9.03	1	-2
25	86	22	-34.03	4	2
26	84	21	16.61	-2	-1
27	85	22	-8.40	1	1
28	82	22	25.82	-3	0
29	82	22	0.00	0	0
30	83	24	-8.82	1	2
31	85	23	-17.01	2	-1
32	86	22	-8.21	1	-1
33	84	30	16.61	-2	8
34	84	21	0.00	0	-9
35	84	22	0.00	0	1
36	85	24	-8.40	1	2
37	82	23	25.82	-3	-1
38	86	22	-34.03	4	-1
39	83	21	25.22	-3	-1
40	84	20	-8.61	1	-1
41	82	22	17.42	-2	2
42	86	23	-34.03	4	1
43	81	35	43.07	-5	12
44	82	22	-9.03	1	-13
45	83	26	-8.82	1	4
46	82	22	8.82	-1	-4
47	86	21	-34.03	4	-1
48	87	25	-8.02	1	4
49	84	22	24.63	-3	-3
50	82	22	17.42	-2	0

51	86	20	-34.03	4	-2
52	90	22	-31.01	4	2
53	86	23	31.01	-4	1
54	87	23	-8.02	1	0
55	82	23	42.05	-5	0
56	83	24	-8.82	1	1
57	85	22	-17.01	2	-2
58	86	24	-8.21	1	2
59	81	23	43.07	-5	-1
60	81	22	0.00	0	-1
61	80	22	9.26	-1	0
62	80	23	0.00	0	1
63	82	22	-18.29	2	-1
64	83	24	-8.82	1	2
65	84	24	-8.61	1	0
66	82	23	17.42	-2	-1
67	83	23	-8.82	1	0
68	86	23	-25.22	3	0
69	84	24	16.61	-2	1
70	85	22	-8.40	1	-2
71	82	22	25.82	-3	0
72	82	25	0.00	0	3
73	84	26	-17.42	2	1
74	84	20	0.00	0	-6
75	86	23	-16.61	2	3
76	87	25	-8.02	1	2
77	82	24	42.05	-5	-1
78	84	22	-17.42	2	-2
79	86	28	-16.61	2	6
80	89	22	-23.52	3	-6
81	82	20	57.55	-7	-2
82	83	21	-8.82	1	1
Promedio	83.63				
Promedio	23.04				
Cálculo	82.00				



7.1.1 ANALISIS DE RESULTADOS DE LAS PRUEBAS 1, 2 Y 3.

Los resultados obtenidos demuestran que existen diferencias considerables entre los valores adquiridos, lo cual se debe a dos causas:

1.-Vibraciones.

El sistema es capaz de detectar vibraciones ocasionadas por movimientos bruscos y rápidos en el kayak, debidos al golpeteo de la pala con el agua al iniciar la palada, por los movimientos del atleta para controlar el equilibrio y por el oleaje que golpea contra el kayak.

2.- Repetibilidad inconstante.

Las variaciones de BPM y de fuerza, no se deben a un mal funcionamiento en el sistema, sino a que inherentemente un humano no es capaz de repetir una acción en intervalos constantes. En los resultados obtenidos se puede apreciar como una variación de tiempo, imperceptible para un atleta en la ejecución de cada palada, provoca grandes variaciones en los resultados de BPM, esto se debe a que dicho parámetro es calculado para cada palada sin tomar en cuenta los resultados anteriores, y que para cada adquisición se realiza el cálculo suponiendo que la duración de dicha adquisición y de las subsecuentes serán constantes en el intervalo de 1 minuto, por esta razón una pequeña variación de tiempo se traduce como una gran variación en el resultado de las BPM.

Por otro lado, los resultados del parámetro fuerza son considerablemente más constantes que las BPM, esto se debe a que como anteriormente se ha explicado, la fuerza, es un parámetro independiente de las BPM y depende de la aceleración experimentada, no del tiempo, como es el caso de las BPM, sin embargo también se presentaron picos en los valores de este parámetro, ocasionados por vibraciones detectadas.

7.2 SOLUCIÓN A LOS PROBLEMAS DETECTADOS EN LAS PRUEBAS TÉCNICAS.

Se detectaron 2 problemas en el sistema:

- Alta sensibilidad a las vibraciones
- Repetibilidad inconstante

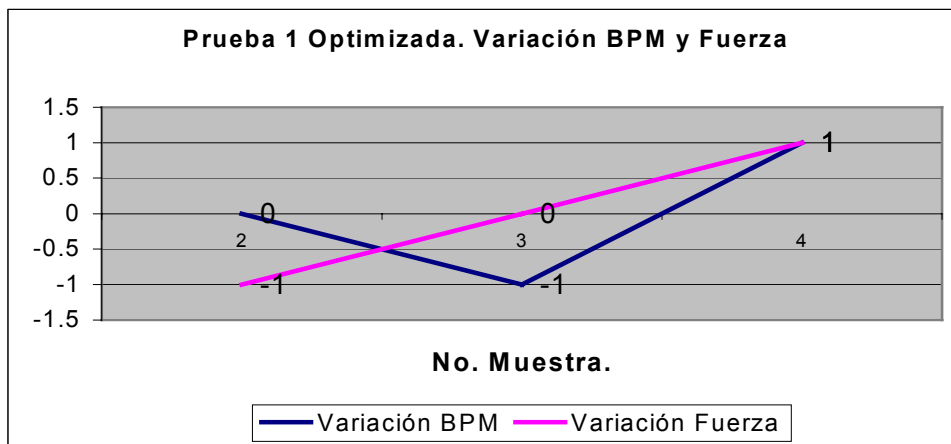
Ambos problemas son ocasionados principalmente por la generación de máximos y mínimos relativos en la señal, los cuales son interpretados como paladas realizadas. La solución consistió en limitar el valor de BPM calculado en la interrupción de flanco de subida del microcontrolador, a un valor no mayor de 150, este límite implica que el valor de fuerza y BPM de alguna adquisición que sobrepase ese valor límite, son eliminadas y no son procesadas en cálculos posteriores. Además de la limitación antes mencionada se realizaron diferentes pruebas concluyendo que es más confiable y útil el despliegue y envío de la información de los valores de BPM y fuerza, resultantes del promedio de cada 4 paladas.

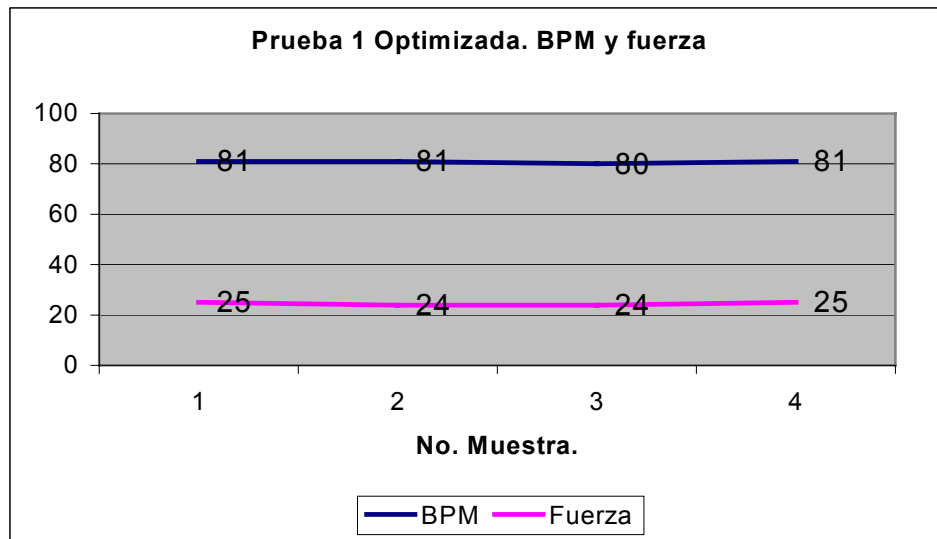
De esta forma el sistema de monitoreo resultó ser más eficaz para esta aplicación, porque además de no presentar variaciones tan grandes principalmente en el valor de BPM, es de mayor utilidad práctica, ya que tanto el atleta como el entrenador pueden analizar el desenvolvimiento del entrenamiento de una forma más conveniente, porque tienen más tiempo para la corrección y mejora del entrenamiento y verificar su desempeño cada 4 paladas, con una variación en los valores visualizados más adecuada para esta aplicación.

7.3 PRUEBAS DE MEDICION DE LAS BPM Y FUERZA, DESPUES DE LA OPTIMIZACION DEL SISTEMA.

Una vez realizadas las modificaciones al sistema de monitoreo, se procedió a repetir las pruebas antes realizadas y evaluar las mejoras del sistema, en estas pruebas no se incluyo el campo "Cálculo BPM" porque las muestras son cada 4 paladas. Los resultados fueron los siguientes:

PRUEBA 1 Optimizada					
Intensidad: 80%					
BPM:80					
Duración:10 seg.					
No.	BPM	Fuerza	Variación entre paladas 1mseg 1	Variación BPM	Variación Fuerza
1	81	25			
2	81	24	0.00	0	-1
3	80	24	9.26	-1	0
4	81	25	-9.26	1	1
Promedio BPM:			80.75		
Promedio Fza:			24.50		

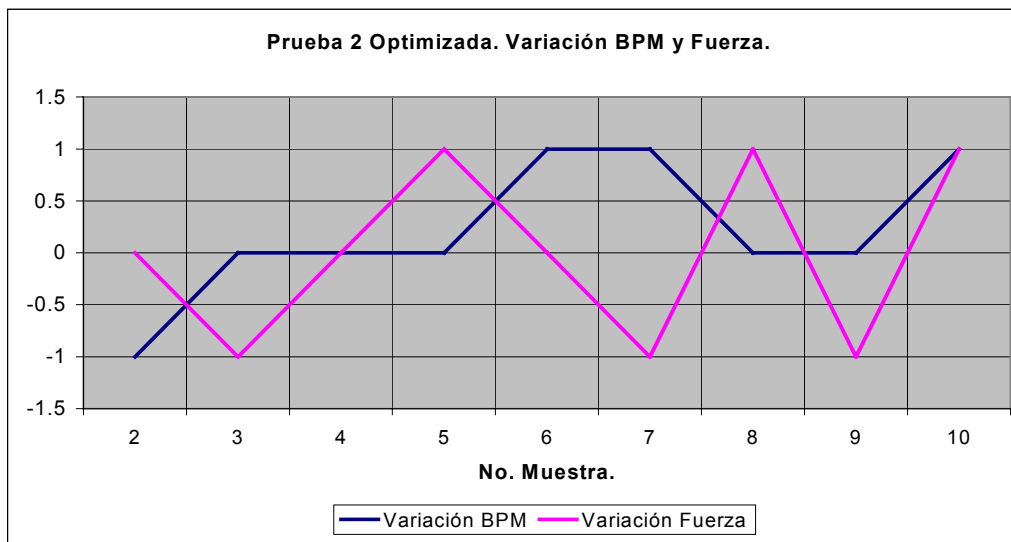
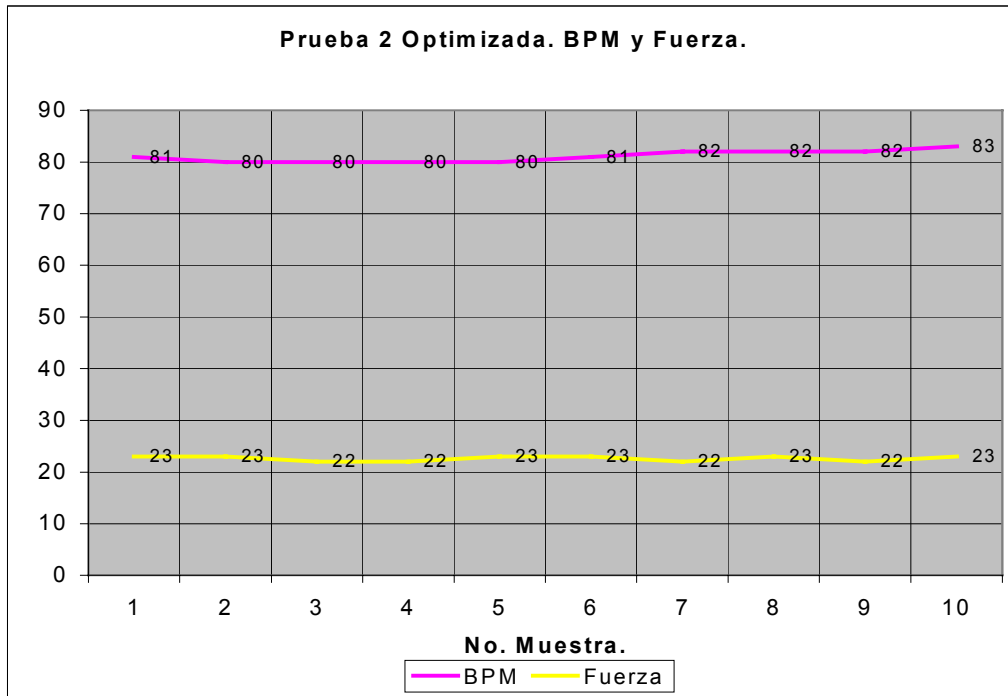




PRUEBA 2 Optimizada

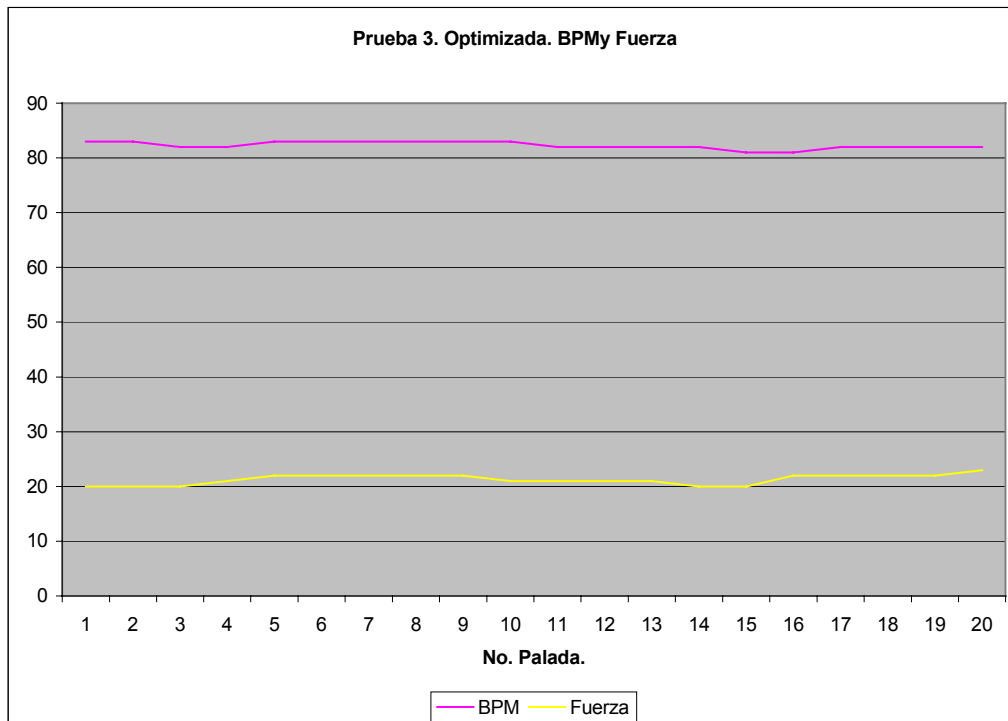
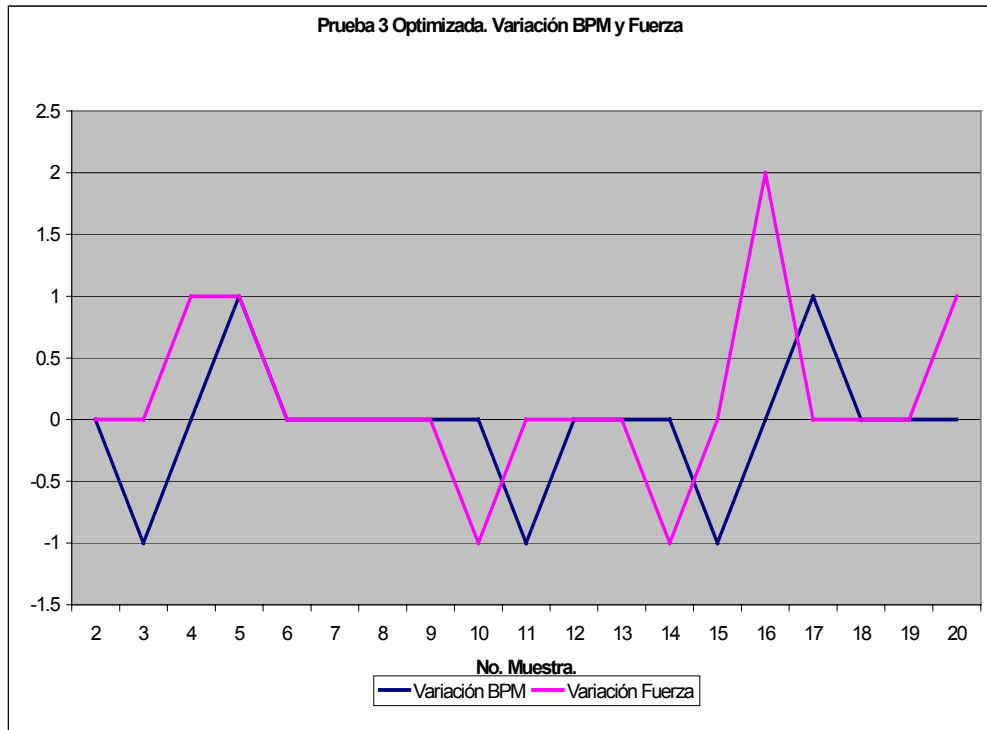
Intensidad: 80%
BPM:80
Duración:30 seg.

No. Muestra	BPM	Fuerza	Variacion entre paladas [mseg.]	Variación BPM	Variación Fuerza
1	81	23			
2	80	23	9.26	-1	0
3	80	22	0.00	0	-1
4	80	22	0.00	0	0
5	80	23	0.00	0	1
6	81	23	-9.26	1	0
7	82	22	-9.03	1	-1
8	82	23	0.00	0	1
9	82	22	0.00	0	-1
10	83	23	-8.82	1	1
Promedio BPM:		81.10			
Promedio Fza:		22.60			



PRUEBA 3 Optimizada.					
Intensidad: 80%					
BPM:80					
Duración:60 seg.					
No. Muestra	BPM	Fuerza	Variación entre paladas [mseg.]	Variación BPM	Variación Fuerza
1	83	20			
2	83	20	0.00	0	0
3	82	20	8.82	-1	0
4	82	21	0.00	0	1
5	83	22	-8.82	1	1
6	83	22	0.00	0	0
7	83	22	0.00	0	0
8	83	22	0.00	0	0
9	83	22	0.00	0	0
10	83	21	0.00	0	-1
11	82	21	8.82	-1	0
12	82	21	0.00	0	0
13	82	21	0.00	0	0
14	82	20	0.00	0	-1
15	81	20	9.03	-1	0
16	81	22	0.00	0	2
17	82	22	-9.03	1	0
18	82	22	0.00	0	0
19	82	22	0.00	0	0
20	82	23	0.00	0	1
Promedio BPM			82.30		
Promedio Fza:			21.30		

7. Pruebas y Resultados



7.3.1 ANALISIS DE LAS PRUEBAS 1, 2 Y 3, OPTIMIZADAS.

Claramente las adecuaciones hechas al sistema muestran una mejoría considerable, los valores obtenidos muestran variaciones acordes al desempeño del atleta, ya que dichos valores se encuentran dentro de los rangos prácticos, que resultan ser de mayor utilidad y de esta forma ser evaluados posteriormente, para una mejor planificación de los programas de entrenamiento.

7.4 DISTANCIA MÁXIMA DE RECEPCIÓN Y PÉRDIDA DE DATOS

Esta prueba se realizó a campo abierto, midiendo la distancia máxima a la que el dispositivo receptor pudo captar la información proveniente del transmisor el cual estaba siendo manipulado manualmente por una persona y el receptor siendo leído por otra persona, las cuales verificaban la información enviada y recibida.

La distancia máxima de enlace que se alcanzo fue de 50 mts. sin errores y de aproximadamente de 60 mts. con recepción intermitente de datos.

7.5 INTEGRACIÓN DEL SISTEMA A UN PLAN DE ENTRENAMIENTO

El aumento de velocidad (disminución de tiempo), en las distancias oficiales de competencia, es el objetivo final de un plan de entrenamiento. El diseño de un plan de entrenamiento comprende entre otras cosas el manejo de intensidades (%de capacidad física) y ritmo(BPM) a realizarse en las distancias correspondientes al plan. El sistema de monitoreo puede ser usado para diseñar el plan de entrenamiento tomando en cuenta la fuerza en sustitución de la intensidad, que es una medida que solo es controlada por el atleta de manera empírica y obviamente las BPM.

Los resultados obtenidos muestran claramente que no existe una proporcionalidad directa entre los parámetros BPM y Fuerza, acorde a lo que se planteo inicialmente. Sin embargo, el sistema de monitoreo indica que la fuerza aplicada es proporcional a la velocidad real del kayak, de acuerdo con la *“Formula general de resistencia”*.

La tabla 7.1, muestra el conocimiento empírico de las posibles combinaciones de variación de Fuerza aplicada por un atleta en cada palada y variación de BPM y el efecto producido sobre la velocidad de la embarcación.

Tabla 7.1

Variación de velocidad (conocimiento empírico)		
Variación de fuerza en cada palada	Variación de BPM	Resultado de la variación de velocidad
Incremento de fuerza	Incremento BPM	Incremento de velocidad
	Mantenimiento BPM	Incremento de velocidad
	Disminución BPM	Desconocido
Mantenimiento de fuerza	Incremento BPM	Incremento de velocidad
	Mantenimiento BPM	Mantenimiento de velocidad
	Disminución BPM	Disminución de velocidad
Disminución de fuerza	Incremento BPM	Desconocido
	Mantenimiento BPM	Disminución de velocidad
	Disminución BPM	Disminución de velocidad

La tabla 7.2, muestra las posibles combinaciones de variación de Fuerza adquiridas por el sistema de monitoreo y variación de BPM y el efecto producido sobre la velocidad de la embarcación.

Tabla 7.2
Variación de Velocidad
(realizada con los resultados del Sistema de Monitoreo)

Variación de fuerza en el sistema de monitoreo	Variación de BPM	Resultado de la variación de velocidad
Incremento de fuerza	Incremento BPM	Incremento de velocidad
	Mantenimiento BPM	Incremento de velocidad
	Disminución BPM	Incremento de velocidad
Mantenimiento de fuerza	Incremento BPM	Mantenimiento de velocidad
	Mantenimiento BPM	Mantenimiento de velocidad
	Disminución BPM	Mantenimiento de velocidad
Disminución de fuerza	Incremento BPM	Disminución de velocidad
	Mantenimiento BPM	Disminución de velocidad
	Disminución BPM	Disminución de velocidad

Analizando las tablas anteriores, se puede observar que de manera empírica, no es posible determinar un resultado entre ciertas combinaciones de los parámetros evaluados, por lo cual al incluir estos parámetros dentro de un plan de entrenamiento, la evaluación tendría que realizarse de manera cualitativa basada en la experiencia no permitiendo realizar las correcciones y perfeccionamientos adecuados al atleta para mejorar la ejecución de sus entrenamientos.

Por otro lado utilizando el sistema de monitoreo, tanto el diseño como la evaluación de un plan de entrenamiento permite atacar y detectar las fallas en momentos clave del desempeño de un atleta en sus entrenamientos, para mejorar sus marcas personales.

La utilidad del sistema de monitoreo radica en que permite detectar cuando un atleta no aplica de manera correcta la técnica de palada para impulsar la embarcación y cuando se presentan cuadros de cansancio excesivo que obligan al atleta a realizar cambios de ritmo que afectan o benefician en su rendimiento.

Por ejemplo, en las pruebas realizadas se presentan variaciones de fuerza, proporcionales a la velocidad de la embarcación, las cuales no corresponden a las variaciones de BPM y que por consiguiente tomando como referencia la tabla 7.2 "**Variación de Velocidad (realizada con los resultados del Sistema de Monitoreo)**", se puede evaluar el desempeño de un atleta de la muestra en la tabla 7.3

Tabla 7.3

Evaluación de resultados			
Variación de fuerza en el sistema de monitoreo	Variación de BPM	Resultado de la variación de velocidad	Evaluación
Incremento de fuerza	Incremento BPM	Incremento de velocidad	Apropiado para la salida y cierre de la distancia a recorrer, incremento de cansancio.
	Mantenimiento BPM	Incremento de velocidad	Apropiado para realizar jalones en distancias cortas, el cansancio es controlable.
	Disminución BPM	Incremento de velocidad	Apropiado para distancias largas, técnica muy eficiente, poco cansancio.
Mantenimiento de fuerza	Incremento BPM	Mantenimiento de velocidad	Inapropiado, deformación de técnica.
	Mantenimiento BPM	Mantenimiento de velocidad	Adecuado para cualquier distancia de competencia. Cansancio normal
	Disminución BPM	Mantenimiento de velocidad	Efectividad de ejecución al 100%, el cansancio disminuye
Disminución de fuerza	Incremento BPM	Disminución de velocidad	Inapropiado, deformación de técnica
	Mantenimiento BPM	Disminución de velocidad	Cansancio normal
	Disminución BPM	Disminución de velocidad	Cansancio excesivo

Esta forma de evaluación permitiría a cualquier atleta detectar sus puntos débiles y perfeccionarlos, la experiencia y el uso continuo del sistema de monitoreo permitiría perfeccionar la forma de evaluación.

7.6 POSIBLES MEJORAS AL SISTEMA.

El sistema desarrollado puede ser mejorado en diferentes aspectos técnicos, brindando la posibilidad de obtener un mejor provecho de la información procesada e incluirlo como herramienta fundamental en el monitoreo de la práctica del Canotaje.

Algunas características que aumentarían el potencial del sistema de monitoreo, son las siguientes:

- Inclusión de memoria y captura de valores en el transmisor a intervalos programables.
- Mayor capacidad de memoria en el receptor.
- Capacidad de programación del transmisor y receptor, para trabajar dentro de límites preestablecidos de BPM y fuerza, con alarma sonora cuando se rebasen dichos límites.
- Cronómetro con cuenta regresiva en ambos dispositivos
- Mayor distancia de enlace
- Incorporación en el receptor de la capacidad de señal de varios emisores simultáneamente.
- Comunicación bidireccional en ambos dispositivos para control y programación remota.
- Interface para PC en ambos dispositivos.

- Software de análisis de resultados con un módulo de optimización de los planes de entrenamiento.
- Reducción del tamaño de los dispositivos

Para el desarrollo de dichas mejoras técnicas, es necesario además del desarrollo de ingeniería y diseño, las propuestas y recomendaciones de los usuarios, para adaptarlo aún más, a las necesidades reales de entrenamiento y con esto lograr un perfeccionamiento adecuado.

7.7 OTRAS APLICACIONES

El sistema desarrollado basa su funcionamiento en un circuito que puede ser utilizado en toda aplicación en donde este presente el parámetro aceleración, o sea prácticamente en cualquier aplicación con movimiento implícito.

Realizando las modificaciones y adaptaciones necesarias, el sistema puede ser utilizado en aplicaciones como las siguientes:

- Alarmas de movimiento en automóviles
- Sistemas de navegación aérea
- Sensores de movimiento
- Control de equilibrio de sistemas autónomos (robots).

Estas aplicaciones son solo un ejemplo del potencial que tiene el sistema además de que si se incluye la comunicación bidireccional en el mismo, cualquier aplicación posible puede ser controlada de manera remota.

8. CONCLUSIONES

El sistema de monitoreo es una herramienta de mucha utilidad para el diseño, monitoreo y corrección de los planes de entrenamiento, así como de la detección de fallas en el desempeño de los atletas en cada entrenamiento.

El prototipo desarrollado satisface los requerimientos básicos de información necesarios para el desarrollo e implementación de planes de entrenamiento con un enfoque más especializado.

El resultado de mayor importancia en el sistema de monitoreo es la capacidad de medir la fuerza resultante del esfuerzo del atleta y la posibilidad de comparación con las BPM, parámetros que en conjunto siempre han sido evaluados de manera empírica y que con esta herramienta se pueden evaluar y desarrollar patrones de desempeño, que permitan detectar y corregir las carencias físicas y tácticas de un atleta.

El interés que despertó el prototipo en la comunidad deportiva, en los Entrenadores, Atletas y Directivos de la especialidad, genera muchas expectativas a futuro para mejorar y adecuar el sistema no solo a las embarcaciones de Canotaje sino también a las embarcaciones de Remo, donde en ambas disciplinas se visualiza una gran utilidad en el dispositivo, ya que con ligeras adaptaciones y modificaciones se puede mejorar aún más su funcionalidad y utilidad, con mejoras como: mayor capacidad de memoria tanto en el receptor como en el transmisor, capacidad para programar el dispositivo para realizar las capturas de memoria a intervalos configurables por el usuario, mayor distancia de enlace, cronómetro y timer en ambos dispositivos, programación de rangos de trabajo con alarma sonora que indique al atleta cuando esta fuera del rango programado, inclusive implementar la posibilidad de recibir la información simultanea de varios dispositivos transmisores para poder medir y controlar el entrenamiento de varios atletas al mismo tiempo, interfase serial o USB para vaciar la información adquirida en una PC, y analizarla con un software especializado en donde no solo se pueda analizar la información de manera gráfica sino también que permita realizar y llevar el control y evolución de planes de entrenamiento así como del seguimiento del desempeño de un atleta, con metas a corto mediano y largo plazo. Además de las mejoras a nivel lógico del sistema también es importante pensar en las mejoras a nivel físico como permeabilidad al 100%, carcasa resistente a los golpes y caídas, flotabilidad, además de un diseño moderno y agradable.

Aunque son muchas las mejoras que necesita el sistema para lograr ser una herramienta de uso masivo y común en la disciplina del canotaje, el sistema desarrollado demuestra muchas ventajas por sobre otros sistemas ya que existen algunos aparatos que permiten medir solo las BPM, o la velocidad pero no ambos, a diferencia de este sistema que permite medir de manera simultanea dos parámetros de suma importancia, que son la fuerza y las BPM, que analizándolos de manera conjunta generan información de mayor utilidad para determinar las causas de las variaciones de velocidad, que si simplemente se obtuviera el valor de la velocidad o BPM, que nos darían la información del rendimiento de un atleta pero no sus causas. Además el sistema ofrece la gran utilidad de poder recibir la información generada de manera inalámbrica, característica que permite en todo momento monitorear el desempeño de un atleta en instantes críticos del entrenamiento. Vale la pena mencionar que uno de los sistemas más vendidos en el mundo que consiste de dos sensores y el aparato en donde se procesa la información, se coloca uno de ellos en la pala, el cual basa su funcionamiento en un "tilt switch" (switch de inclinación) para medir las BPM y otro sensor colocado en el asiento que por medio de la presión ejercida en él, calcula la fuerza del atleta en cada palada. Ese sistema es muy delicado tanto en su uso como en su durabilidad ya que para alimentar y recibir información del tilt switch se coloca un cable que va del aparato en el kayak hacia la pala, el cual invariablemente tiene que ser reemplazado constantemente porque el continuo movimiento al cual se ve sometido ocasiona fracturas internas con la consecuente pérdida de la información, además de la incomodidad que ocasiona el no tener la pala libre para realizar los movimientos de forma cómoda.

La forma en como se mide la fuerza en ese sistema no es la adecuada porque si bien es cierto que cada palada es apoyada con el movimiento que realiza la pierna entre la pedalina y el asiento, dicho movimiento no representa la resultante de la fuerza total ejercida sobre el kayak, además de que cada remero según su estilo de remar realiza dicho movimiento con diferentes magnitudes al apoyarse con la pedalina y el asiento porque no solo se realiza ese movimiento para apoyar la palada sino también para conservar el equilibrio por lo que las variaciones de presión en el asiento no son representativas solo de la fuerza ejercida.

Es por esto que el sistema desarrollado no se queda en este trabajo de tesis sino que ya se tiene planeada una producción inicial, de 5 aparatos con mayor capacidad de memoria, posibilidad de programar rangos de esfuerzo y alarma sonora que suene cuando se este fuera del rango programado, los cuales serán utilizados de manera experimental por el Equipo Nacional de Canotaje, rumbo a las olimpiadas de Atenas 2004.

Con la implementación de este sistema se demuestra el infinito potencial que tienen los sistemas electrónicos para solucionar problemas, satisfacer necesidades y facilitar las tareas de las diferentes actividades humanas.

Con el desarrollo de este sistema se ampliaron los conocimientos del funcionamiento y programación de los microcontroladores empleados así como de la adecuada integración con otros dispositivos electrónicos. El aprendizaje que se obtuvo con la parte del proyecto dedicada a la transmisión y recepción de la información fue de suma importancia, ya que el conocimiento adquirido acerca de las comunicaciones inalámbricas, dejo ver la importancia que tiene el desarrollo de un adecuado protocolo de comunicaciones, para optimizar los recursos de un sistema dado y de esta forma lograr el mejor enlace posible.

9. APENDICES

9.1 GUÍA DE USUARIO

DISPOSITIVO TRANSMISOR

1. Colocar el dispositivo transmisor en el kayak enfrente del remero delante de la bañera.
2. Encender el dispositivo, la transmisión y el despliegue de datos comienza inmediatamente (figura A1).

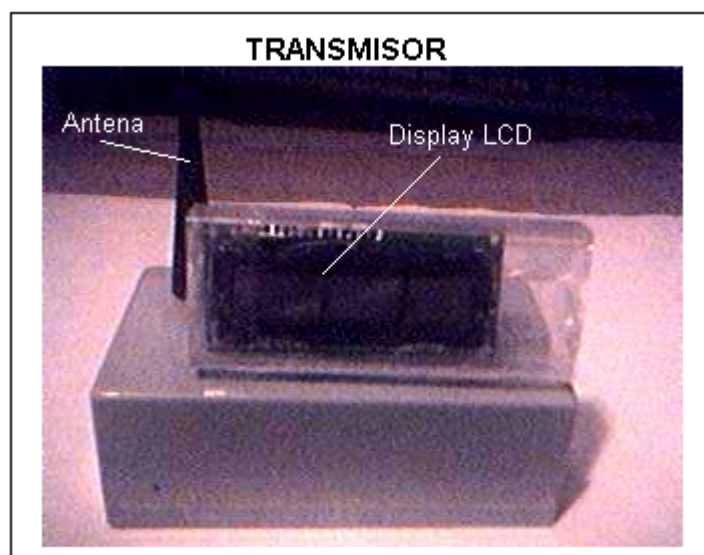


Figura A1 Dispositivo transmisor

DISPOSITIVO RECEPTOR

1. Encender el dispositivo, la recepción de datos y despliegue de datos comienza inmediatamente.
 - **Captura de memorias**
 1. Presionar el botón "C" para capturar la información de BPM y Fuerza recibida, cada vez que se presione dicho botón aparecerá la cuenta del número de memorias capturadas en la pantalla de LCD, hasta 5 memorias(figura A2).
 - **Visualización de memorias**
 1. Presionar el botón "V", para visualizar las memorias previamente capturadas, la información en memoria aparecerá a la derecha de la información recibida así como el número correspondiente de memoria (figura A3).
 - **Borrado de memoria**
 1. Si se desean borrar las memorias, presionar el botón "V" repetidamente hasta visualizar en la pantalla de LCD el símbolo "&", posteriormente presionar el botón "C". Todas las memorias serán borradas y aparecerá en el indicador de memorias capturadas el número "0"

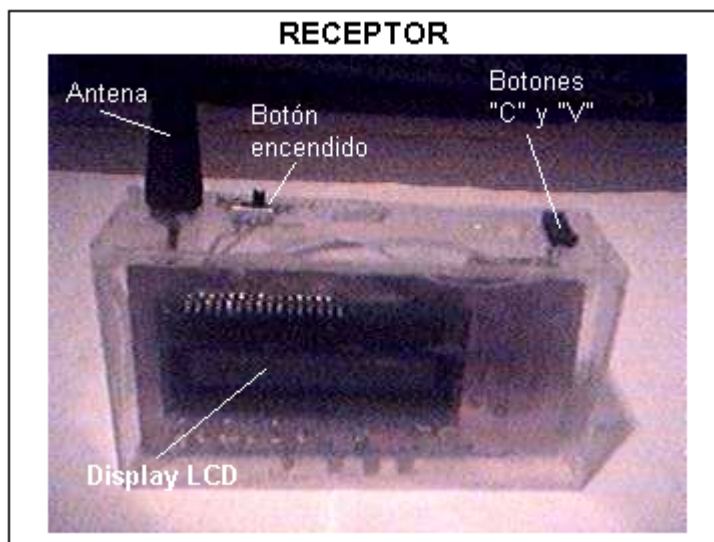


Figura A2 Dispositivo receptor

Indicadores en pantalla LCD del Receptor

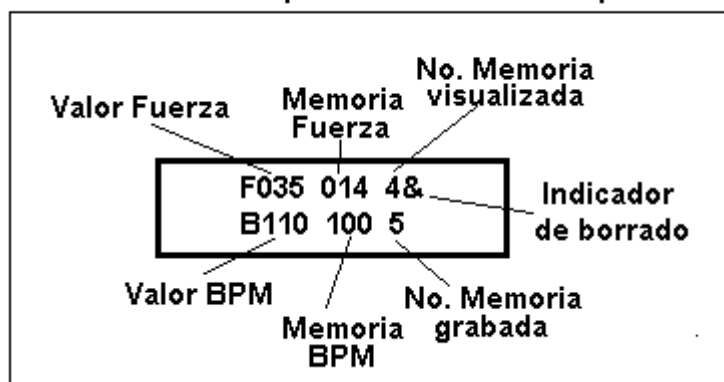
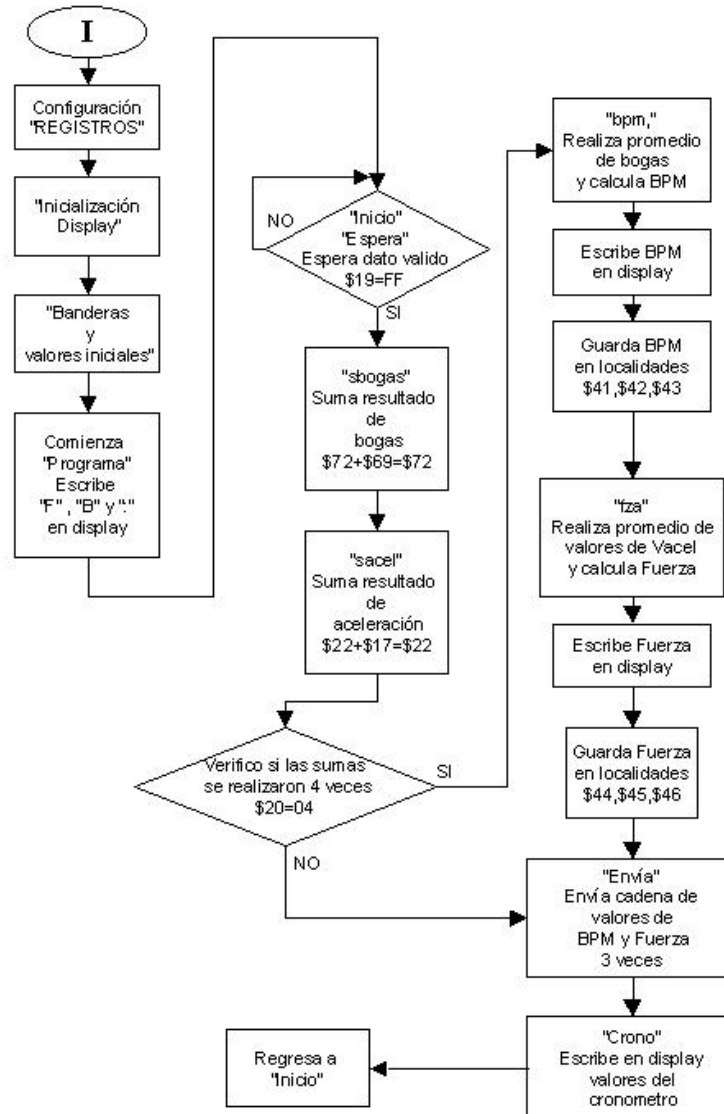


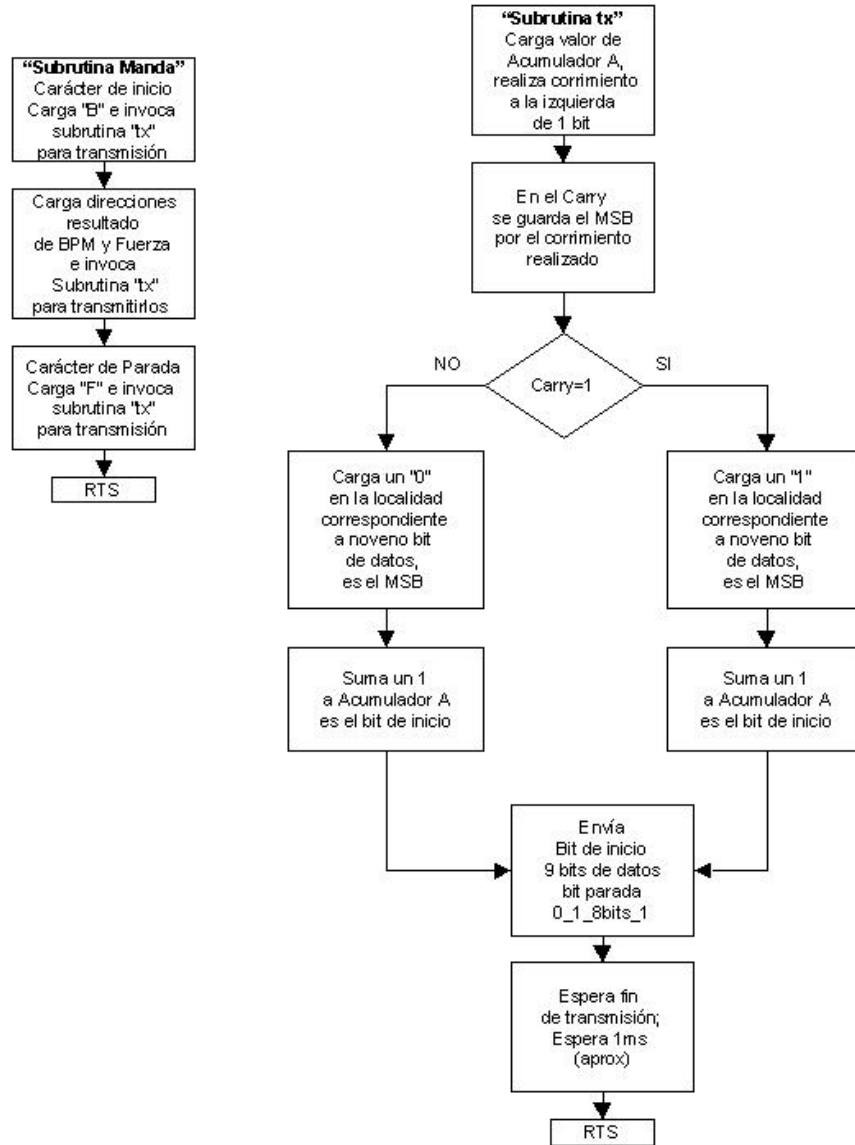
Figura A3 Indicadores de display

9.2 DIAGRAMAS DE FLUJO DEL TRANSMISOR

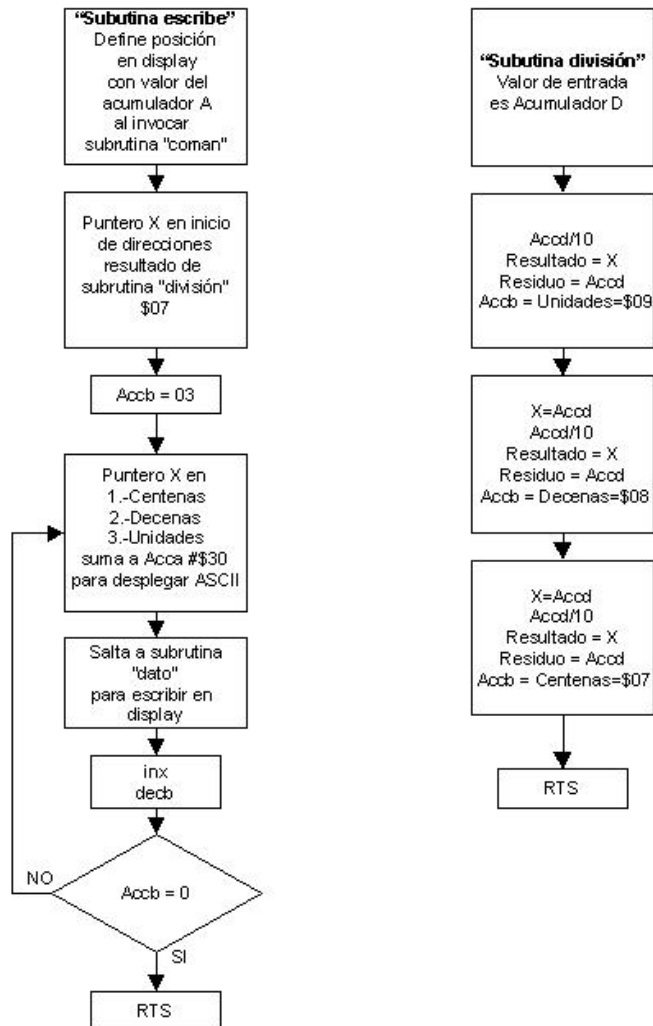
"Programa Transmisor"



Subrutinas "Manda" y "Tx"



Subrutinas "escribe" y "división"



Subrutinas del display "dato", "coman" y "delay"

"Subrutina dato"
 (Escribe datos en display)
 Acca es el valor de entrada
 R/W=0

Carga valor de entrada en bus de datos de display

RS=1 E=1

RS=1 E=0

RS=0 E=0

Espera aprox. 1ms.
 Subrutina "delay"

RTS

"Subrutina coman"
 (Envía comandos al display)
 Acca es el valor de entrada
 R/W=0

Carga valor de entrada en bus de datos de display

RS=0 E=0

RS=0 E=1

RS=0 E=0

Espera aprox. 1ms.
 Subrutina "delay"

RTS

"Subrutina delay"
 Carga acumulador A con #\$ff

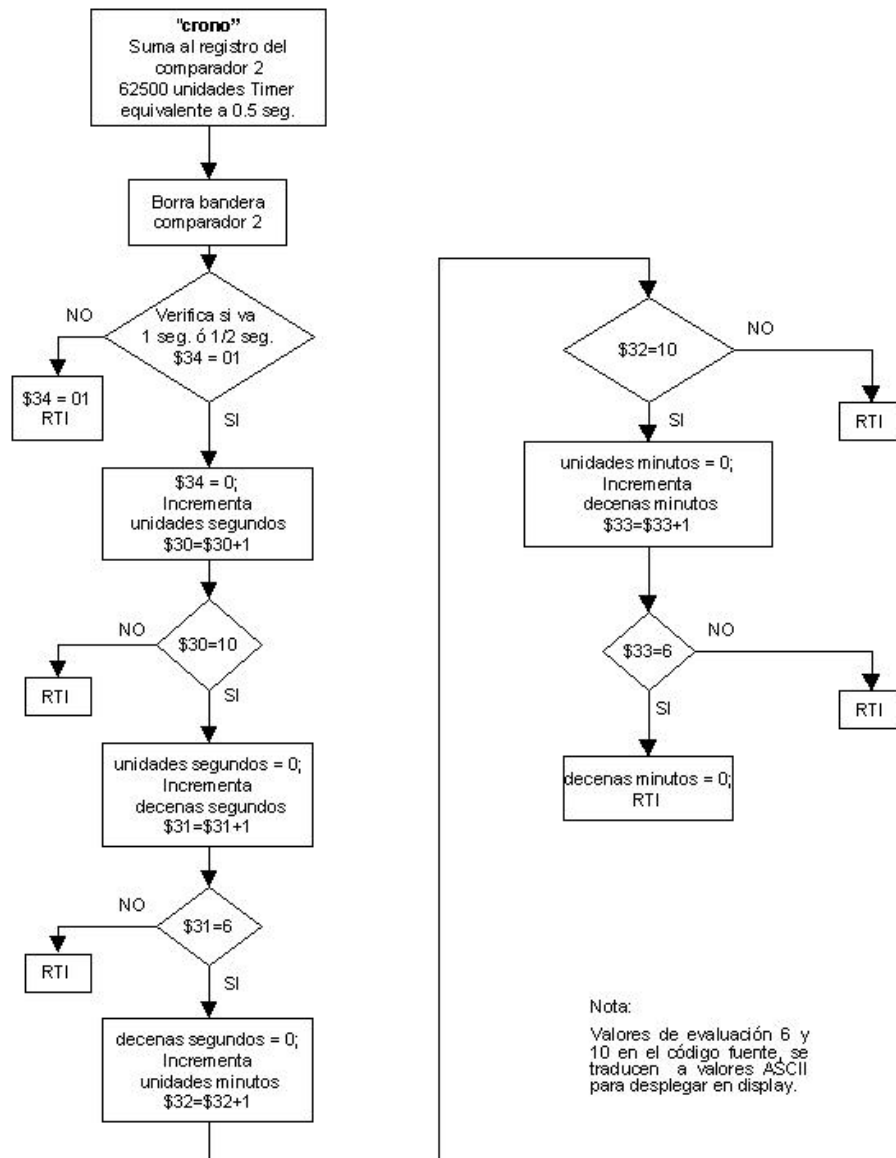
deca

NO
 Acca = 0

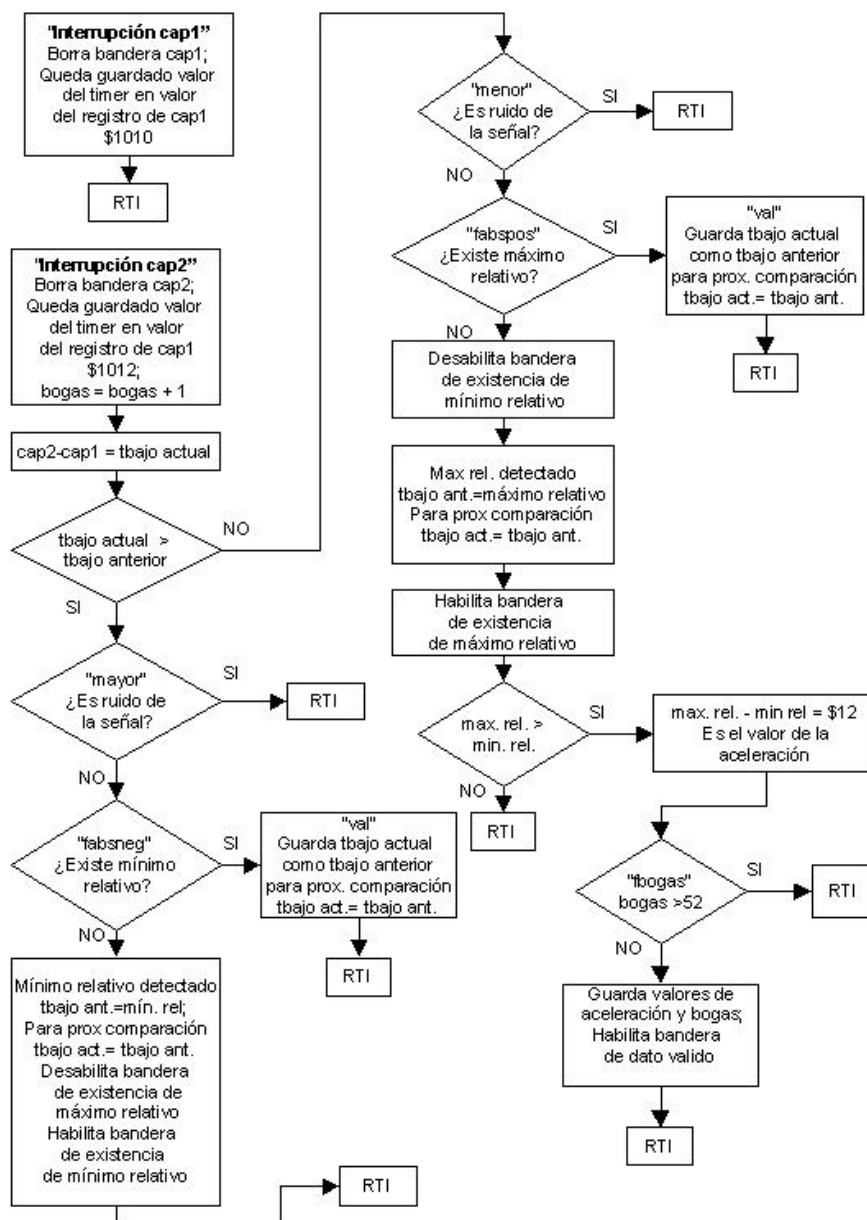
SI

RTS

Interrupción del Comparador 2 "crono"

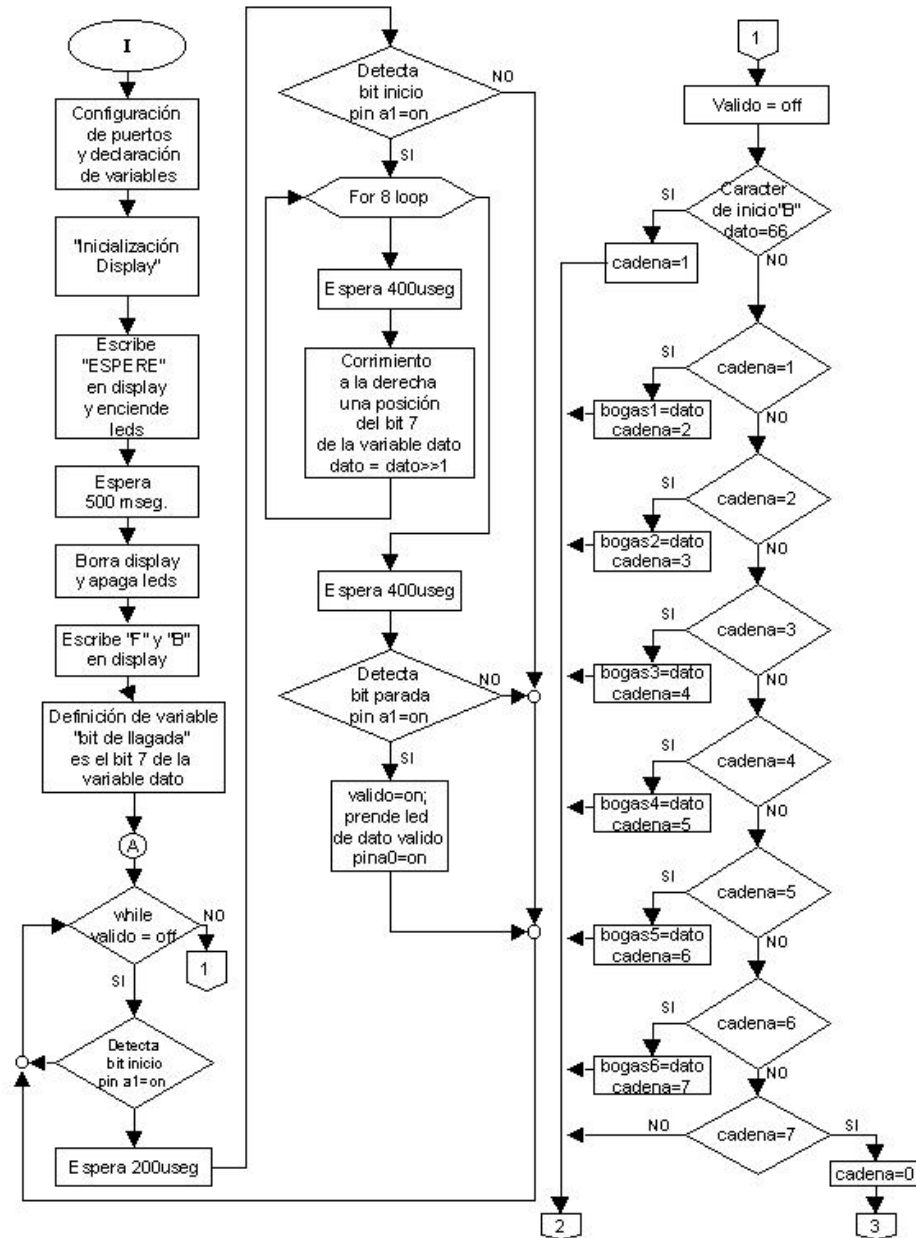


Interrupciones de Capturadores 1 y 2 "cap1" y "cap2"

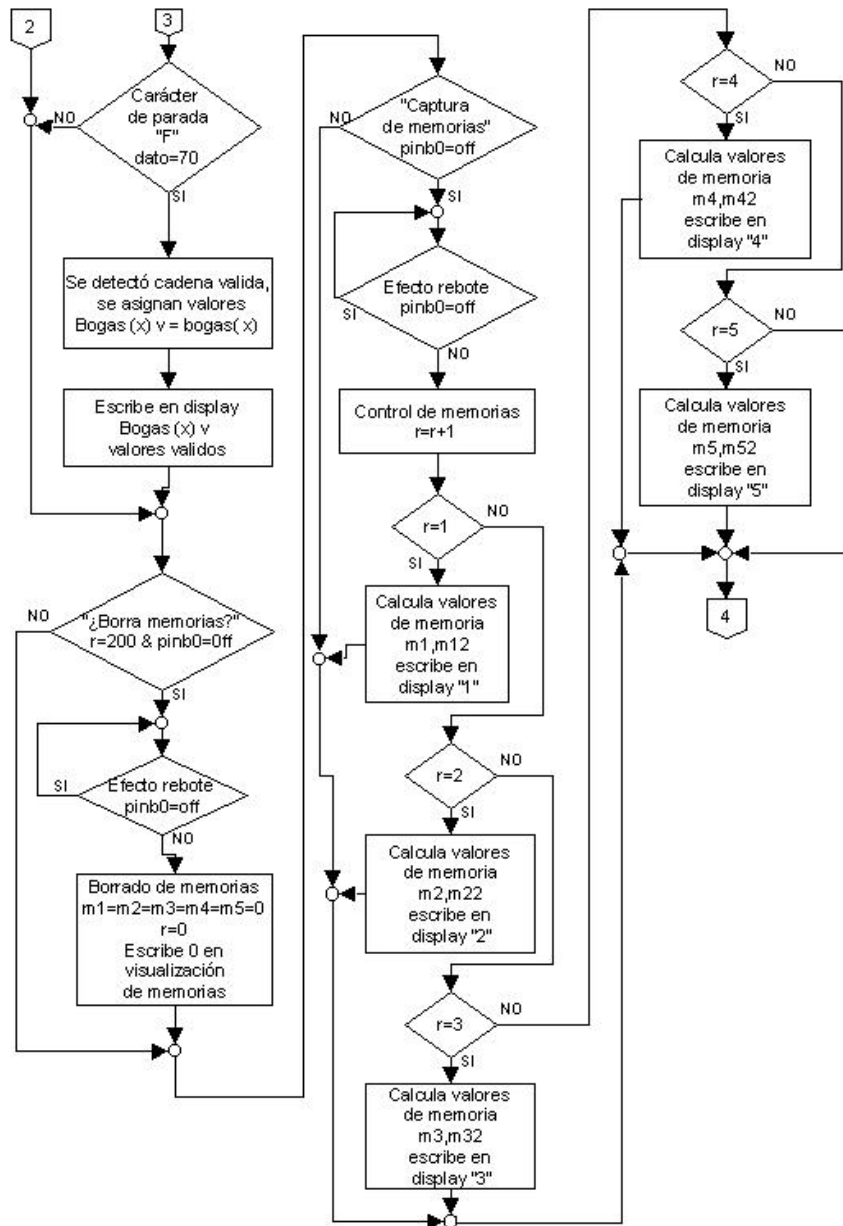


9.3 DIAGRAMAS DE FLUJO DEL RECEPTOR

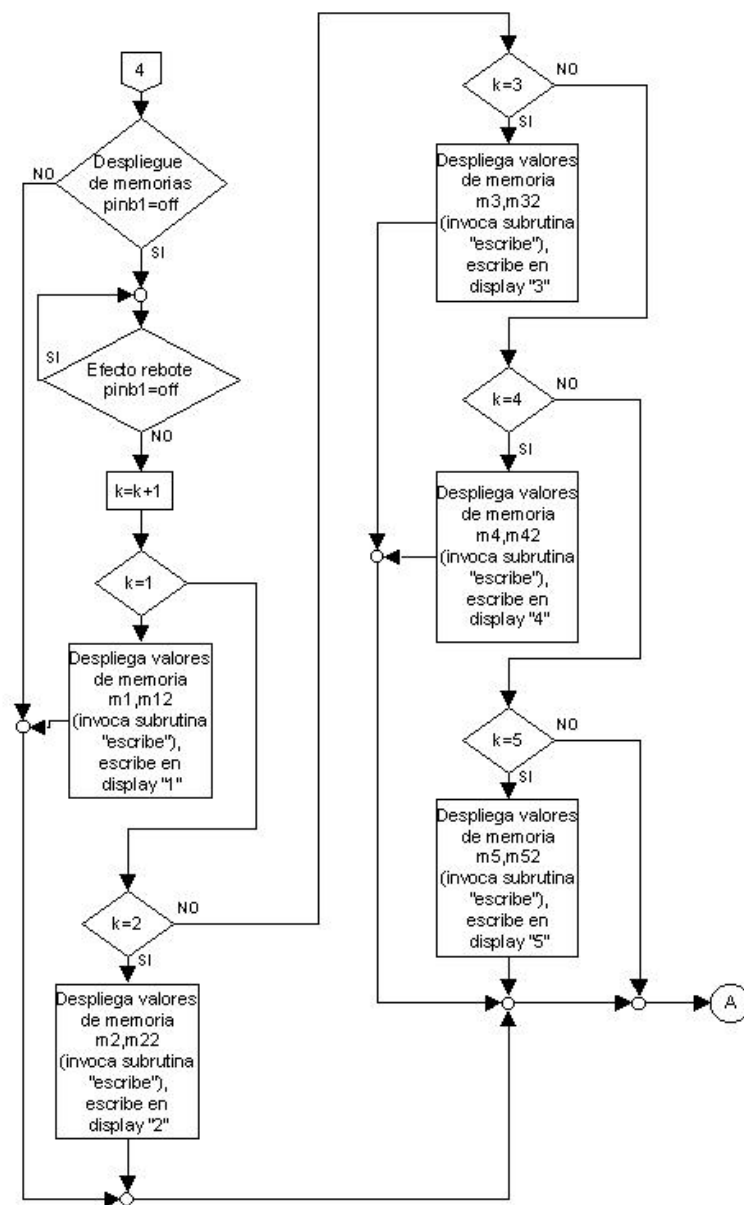
"Receptor"



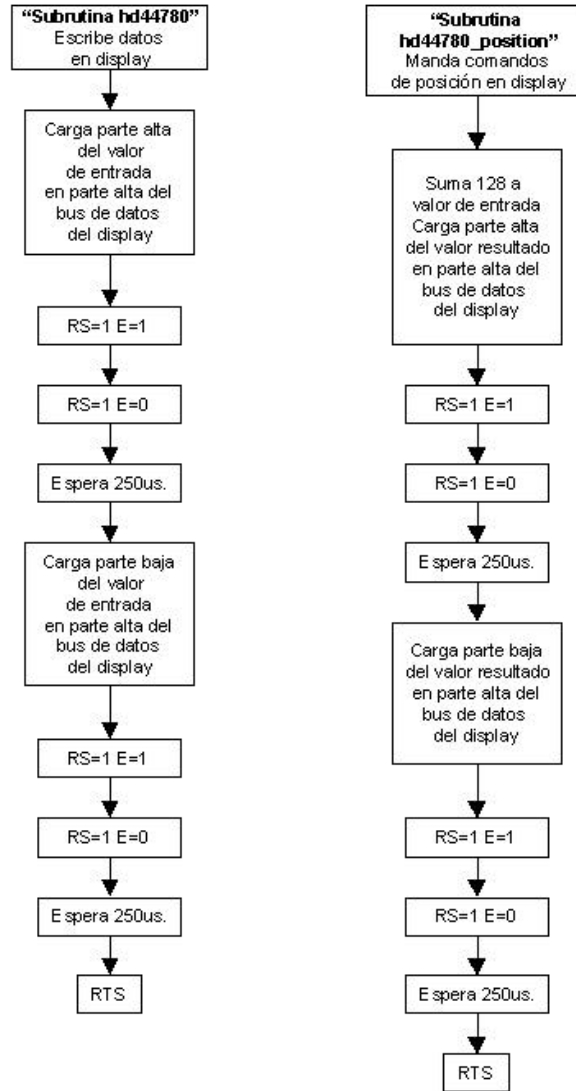
“Receptor (cont.)”



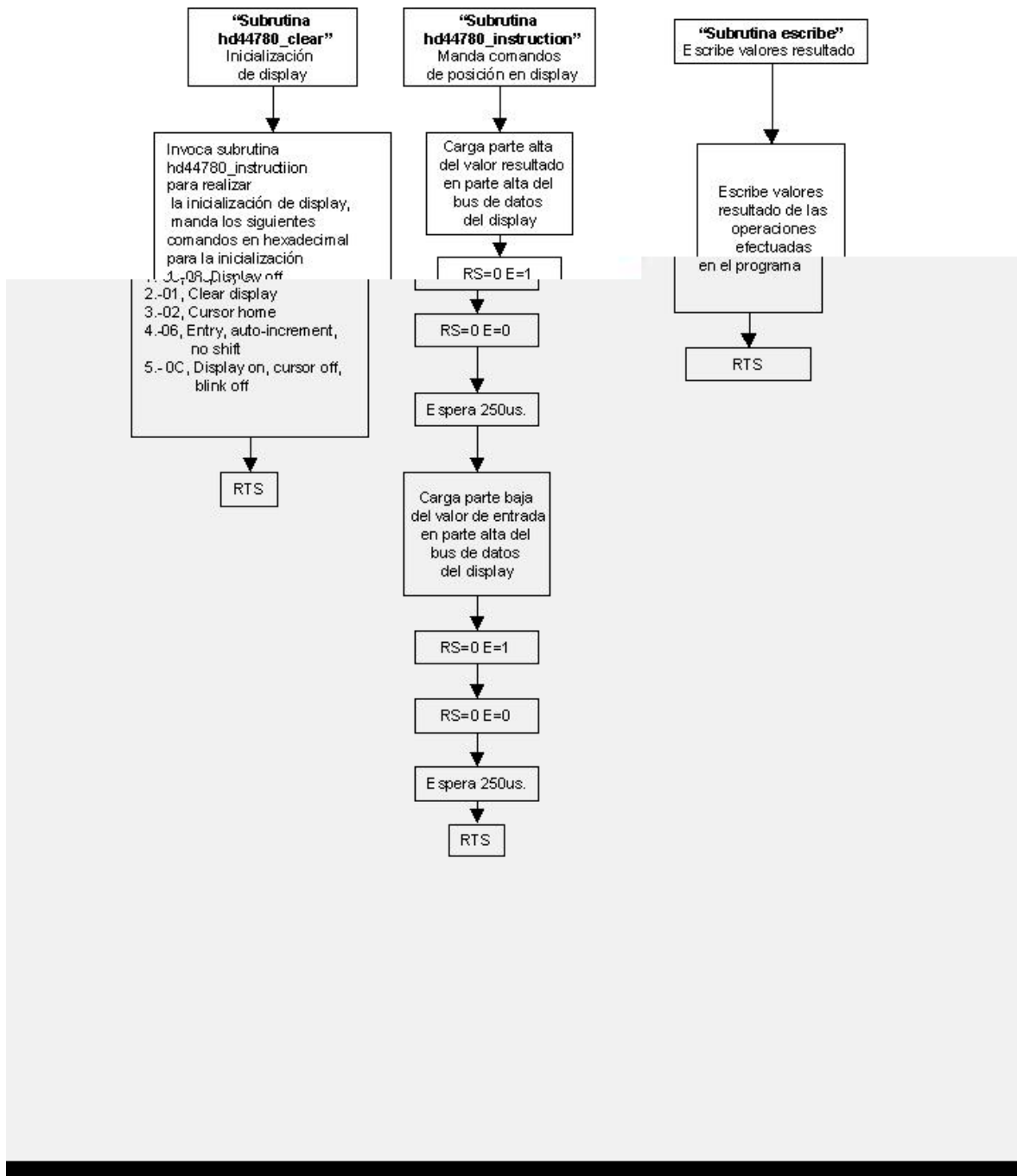
"Receptor (cont.1)"



Subrutinas "hd44780" y "hd44780_position"



Subrutinas "hd44780_clear", "hd44780_instruction" y "escribe"



9.4 CÓDIGO FUENTE DEL TRANSMISOR

```

**archivo final de la tesis
**Registros cap1=$1010-$1011
**   cap2=$1012-$1013

    org $f800
main   equ *
    lds #$ff ;stack al final de la RAM

*****REGISTROS*****
    idx #$1000

    ldaa #$32
    staa $102b ;baud 2400 (registro indefinido al reset)

    bset $2d,x,$08 ;tx habilitado
    bset $24,x,$03 ;incremento de cuenta del tiempo cada 8useg
    bset $22,x,$46 ;habilito capturadores 1 y 2, comparador 2
    bset $21,x,$24 ;flancos de bajada cap1 y de subida cap2
    bset $20,x,$00 ;no hay salida hardware del comparador
    bset $07,x,$ff ;port C como salidas
    bset $3c,x,$0c ;maxima prioridad de interrupcion la de comparador 2
    bset $09,x,$02 ;port d tx salida
    bset $2c,x,$10 ;0_9bits_1
    cli ;habilito interrupciones

*****FIN REGISTROS*****

*****INICIALIZACION DISPLAY*****
    ldx #$ffff ;Retardo para que encienda el display
l     dex      ;en modo bootstrap
    bne l
    ldaa #$38 ;function set,8 bits, 2 lineas,5x10 puntos
    jsr coman
    ldaa #$0c ;display on,cursor off
    jsr coman
    ldaa #$01 ;borra display
    jsr coman
    ldaa #$06 ;entry mode set,desplaza a la derecha
    jsr coman

*****FIN INICIALIZACION DISPLAY*****

*****banderas y valores iniciales*****

    ldx #$f424
    stx $1018 ;62500 ciclos de 8us = 500ms
              ;la interrupcion se producira cada 0.5 segs
              ;para el reloj

```

```

Idx #0000 ;
stx $50 ;valor inicial"tiempo anterior"
stx $60 ; absneg=00
stx $67 ;valor de la cuenta de bogas
Idx #1aa
stx $62 ;abspos
Idaa #17
staa $65 ;fabsneg bandera absoluto negativo
staa $66 ;fabspos bandera absoluto positivo
Idx #1f80 ;ciclos por minuto frec. 134.4
stx $69 ;134.4 x 60 = 8064 = 1f80
Idx #50 ;80 decimal peso
stx $90
Idaa #00
staa $17
staa $22 ;valor inicial aceleracion
staa $71 ;valor inicial bogas
Idaa #04
staa $20 ;control promedios

```

```

Idaa #30 ;cero ascci
staa $30 ;dir de unidades segundos 0-9
staa $31 ;dir decenas segundos 0-6
staa $32 ;dir unidades minutos 0-9
staa $33 ;dir decenas minutos 0-6
staa $34 ;dir de dos veces int. comp2 1=1/2 seg, 2pases=1seg aprox.
Idaa #ff ;bandera dato valido
staa $19

```

*****PROGRAMA*****

```

Idaa #85
jsr coman
Idaa #46 ;escribe F
jsr dato
Idaa #c5
jsr coman
Idaa #42 ;B
jsr dato

```

```

Idaa #cd
jsr coman
Idaa #3a ; ":" dos puntos para el crono
jsr dato

```

inicio Idx # \$00

**espera brclr \$19,x,\$ff,envia ;espero a que haya un dato valido
 bclr \$19,x,\$ff ;pongo a cero la bandera de dato valido**

**sbogas Idd \$71
 add \$69
 std \$71 ;suma resultado bogas**

**sacel Idd \$22
 add \$17
 std \$22 ;suma resultado aceleracion**

**Idab \$20 ;checo cuantas veces va \$20=04
decb
stab \$20
bne envia ;salta si no es cero a envia**

**Idaa # \$04
staa \$20 ;reinicio control promedios**

**bpm Idd \$71 ;realizo promedio bogas adquiridas
 Idx # \$04
 idiv
 stx \$71 ;resultado en X residuo en D
 Idd # \$1f80 ; 134*60=1f68=8040 134.4*60=1f80=8064
 Idx \$71 ;valor promedio bogas
 idiv ;divido valor max bpm/valor promedio
 ;resultado en X residuo en d
 stx \$71 ;guardo resultado bpm**

**Idd \$71 ;division trabaja con acumulador D
jsr division ;saco valores decimales
Idaa # \$c7 ; escribe bpm
jsr escribe**

**Idaa \$07 ;guardo valores de bogas por minuto
staa \$41 ;centenas**

**Idaa \$08 ;decenas
staa \$42**

**Idaa \$09 ;unidades
staa \$43**

**fza Idd \$22 ;carga con valor de aceleracion
 Idx # \$04
 idiv**

```

    stx $80
    stx $82
    ldd $80
    ldx $90 ;80 decimal=masa
f    addd $82
    std $80 ;aceleracion + aceleracion ,$190 veces
    dex
    bne f
    std $80 ;guardo valor de la multiplicaciøn
    ldd $80
    ldx #$74 ;116 decimal valor prom de 1G
    idiv
    stx $80 ;valor de fuerza

vfza    ldd $80 ;fza
        jsr division
        ldaa #$87
        jsr escribe ; *****escribe fza y incluido tx de fza*****

        ldaa $07 ;guardo valores de fuerza
        staa $44 ;centenas

        ldaa $08 ;decenas
        staa $45

        ldaa $09 ;unidades
        staa $46

        ldx #$00
        stx $71
        stx $22

envia    jsr manda ; envio datos 3 veces
        jsr manda
        jsr manda

crono    ldaa #$cb ; escribo valores del reloj
        jsr coman
        ldaa $33
        jsr dato

        ldaa $32
        jsr dato

        ldaa #$ce
        jsr coman

        ldaa $31
        jsr dato

        ldaa $30

```

jsr dato

jmp inicio

*****FIN PROGRAMA*****

*****SUBROUTINAS*****

*****subrutina manda tx datos*****

```

manda   psha
          pshx
          pshb

          Idaa #42 ;"B" manda 3 veces
          jsr tx
          jsr tx
          jsr tx
          Idx #41 ;carga datos de las direcciones resultado
          Idab #06 ;6 datos a transmitir
yatx   Idaa $0,x ;puntero de resultados
          adda #30 ;se le suman 30 para que despliegue ASCII
          jsr tx
          inx
          decb
          bne yatx

          Idaa #46 ;"F" fin de cadena
          jsr tx

          pulb
          pulx
          pula
          rts
*****fin manda tx *****

```

*****transmision*****

```

tx      psha
          pshx
          pshb
          Idx #1000

```

```

Isla ;corrimiento a la izquierda 1 lugar pone un cero en lsb
bcs uno; si el carri es uno
bra cero

```

```

uno      bset $2c,x,$40
         inca ;sumo 1 a "A" bit de inicio
         bra serie
cero     bclr $2c,x,$40 ;tengo el mayor msb en el bit9
         inca
serie    staa $102f
wait    brclr $2e,x,$c0,wait ;espero a que termine de transmitir
        jsr delay
        pulb
        pulx
        pula
        rts

```

*****fin transmision*****

*****division*****

```

division psha
        pshx
        pshb
        ldx #$a ;divido entre 10 decimalresiduo en D
        idiv
        stx $02 ;resultado en x residuo en D
        stab $09 ;UNIDADES

```

```

        ldd $02
        ldx #$a
        idiv
        stx $02 ;resultado en x residuo en d
        stab $08 ;DECENAS

```

```

        ldd $02
        ldx #$a
        idiv
        stx $02
        stab $07 ;CENTENAS
        pulb
        pulx
        pula
        rts

```

*****fin division*****

*****escribe 3 lugares*****

```

escribe psha ;posicion del cursor del display
        pshx
        pshb

```

```

    jsr coman
    ldx #$07 ;carga datos de las direcciones resultado
    ldab #$03 ;y despliega en el display
ya   ldaa $0,x ;puntero de resultados
    adda #$30 ;se le suman 30 para que despliegue ASCII
    jsr dato
    inx
    decb
    bne ya
    pulb
    pulx
    pula
    rts
*****fin escribe*****

```

```

*****comparador 2 (reloj)*****
* las direcciones del reloj son:
* $30 ;dir de unidades segundos 0-9
* $31 ;dir decenas segundos 0-6
* $32 ;dir unidades minutos 0-9
* $33 ;dir decenas minutos 0-6
* $34 ;dir de dos veces int. comp2 1=1/2 seg, 2=1seg aprox.

```

```

comp2  psha
       pshx

       ldd $100e
       addd #$f424 ; le sumo 62500 ciclos de reloj
       std $1018

       ldaa #$ff
       staa $1023 ; borro bandera del comparador 2

       ldaa $34 ; cargo con bandera de 1/2 seg
       cmpa #$01
       beq reloj
       ldaa #$01
       staa $34
       bra sreloj

reloj  ldaa #$00
       staa $34 ; pongo bandera de 1/2 seg a cero

uniseg ldaa $30
       inca

```

```
staa $30
cmpa #$3a ; si= 9 decimal incrementa decenas
beq decseg
bra sreloj
```

```
decseg Idaa #$30
staa $30 ;ceros uniseq
Idaa $31
inca
staa $31
cmpa #$36
beq unimin
bra sreloj
```

```
unimin Idaa #$30
staa $31 ;ceros decseg
Idaa $32
inca
staa $32
cmpa #$3a
beq decmin
bra sreloj
```

```
decmin Idaa #$30
staa $32 ;ceros unimin
Idaa $33
inca
staa $33
cmpa #$36
beq ceros
bra sreloj
```

```
ceros Idaa #$00
staa $33 ;ceros decmin
```

```
sreloj pulx
pula
rti
```

```
*****fin comparador 2 (reloj)*****
```

```
*****RUTINAS DE INTERRUPT. CAP1 Y CAP2*****
```

```
cap1 Idaa #$ff
staa $1023 ;borro bandera de cap1
rti
```


cap2 ldaa #\$ff

staa \$1023 ;borro bandera de capturador2

bogas ldx \$67 ;empiezo a contar el tiempoentre una remada y otra

inx

stx \$67

ldd \$1012 ;cargo con valor cap2

subd \$1010 ;resto valor cap1

std \$52 ;obtengo tiempo en bajo(valor actual)

cpd \$50 ;comparo con valor en bajo anterior

bhs mayor ;si actual es mayor salta a mayor

bra menor ;si valor actual es menor salta a menor

;necesito verificar que tan menor es

;alrededor de 4 de diferencia(ruido)

menor ;implemento la rutina de ruido

ruidon ldd \$50 ;valor anterior

subd \$52 ;valor actual

cpd #\$08

bhs fabspos ;no hay ruido salta a checar bandera

bra salcap2 ;si hay ruido, sale de interrupcion

fabspos ldaa \$66

cmpa #\$ff; ff=habilitada 00=no habilitada

beq val ; en val tbact=tbant

ldaa #\$00

staa \$65 ;desabilito flag abs.negativo

ldd \$50 ;cargo valor anterior

std \$62 ;guardo en absoluto positivo

ldd \$52

std \$50 ;t.bajo actual guardo en tbajoanterior para prox.comparacion

ldaa #\$ff

staa \$66 ;habilito flag de abs . positivo

difabs ldd \$62 ;abspos-absneg

cpd \$60

bhs resta ;doble chequeo

bra salcap2

resta ldd \$62

subd \$60 ;aqui ya es un dato valido

guarda std \$12 ; valor aceleracion valido en \$12 y \$13

```

fbogas ldd $67 ;cargo valor de cuenta bogas
      cpx #$32 ; y no calculo vibraciones 1f80/2d
      blo bbogas ; si es menor salta, limitadoa a 201 bpm

      std $69 ;guardo bogas en esta dir.
      ldx #$00
      stx $67 ;borro cuenta bogas
      ldd $12
      std $17 ;guardo valor aceleracion

si     ldaa #$ff
      staa $19 ;bandera de dato valido
      bra salcap2

bbogas ;ldx #$00 ; borro cuenta de bogas
      ;stx $67
      bra salcap2

mayor ;implemento la rutina de ruido
ruidop ldd $52 ;valor actual
      subd $50 ;valor anterior
      cpd #$08
      bhs fabsneg ;no hay ruido salta a checar bandera
      bra salcap2 ;si hay ruido sale de interrupcion

fabsneg ldaa $65 ;existe minimo relativo? ff=si 00=no
      cmpa #$ff; ff=habilitada 00=no habilitada
      beq val ; guarda tiempo actual como anterior
      ldaa #$00
      staa $66 ;desabilito bandera abs.positivo
      ldd $52
      std $50 ;tbact=tbant tiempo bajo actual=t.bajo anterior
      std $60 ;tbact=absneg t.bajo actual = abs.negativo
      ldaa #$ff
      staa $65 ;habilito flag de abs . negativo
      bra salcap2 ;salto a verificar si voy de subida

val     ldd $52
      std $50 ;guardo valor actual como anterior

salcap2 rti

```

*****FIN INTS. CAP1 Y CAP2*****

*****SUBROUTINAS DEL DISPLAY 811E2 *****

```

dato     psha ;subrutina de escribir dato en display
          pshx

```

```

Idx #$1000
staa $04,x ;datos en portb
bset $03,x,03 ;RS=1 E=1 ;port C pines 0 y 1
;RS=PC0 , E=PC1
bclr $03,x,02 ;RS=1 E=0
bclr $03,x,03 ;RS=0 E=0
bsr delay
pulx
pula
rts

```

```

coman      psha ;sub para mandar comandos al display
pshx
Idx #$1000
staa $04,x ;comandos en port B
bclr $03,x,03 ;RS=0 E=0
bset $03,x,02 ;RS=0 E=1 ;port C pines 0 y 1
;RS=PC0 , E=PC1
bclr $03,x,02 ;RS=0 E=1
bsr delay
pulx
pula
rts

```

```

delay      psha ;delay 1.414 mseg
pshx
ldaa #$ff
retar      deca
nop
nop
nop
bne retar
pulx
pula
rts

```

*****FIN SUBROUTINAS DEL DISPLAY*****

```

org $ffee ;$ffee $e8 interrupcion capturador1
fdb cap1

```

```

org $ffec ;$ffec $e5 interrupcion capturador2
fdb cap2

```

```
org $ffe6 ;interrupcion del comparador 2  
fdb comp2
```

```
org $ffe  
fdb main ;vector de reset
```

9.5 CÓDIGO FUENTE DEL RECEPTOR

```
-- programa final de la tesis este es el receptor
-- modulo de comunicaciones
-- este es el archivo de recepcion despliega los valores en pantalla del receptor
-- se enciende el led amarillo(o verde) cada que se recibe un caracter valido
-- se enciende el led verde cada que se recibe una cadena de caracteres valida
-- se enciende el led rojo cada que se pierde una cadena de caracteres
-- cacracter de inicio es "B" y de parada es "F"
-- la cadena es B_bbb_fff_F
-- este archivo tiene la programacuion para la memoria
```

```
include 16f84_10
```

```
-- standard library
include jlib
include hd447804
pin_a0_direction = output -- led verde o amarillo caracter valido
pin_a1_direction = input  -- receptor de la antena
pin_a2_direction = output -- led rojo indica cadena erronea
pin_a3_direction = output -- led verde cadena valida
```

```
pin_b0_direction = input -- boton de inicio de captura
pin_b1_direction = input -- boton de fin de captura
pin_b2_direction = output
pin_b3_direction = output
port_b_high_direction = all_output
```

```
var bit valido = off
var byte dato = 0
```

```
var byte bogas1 = 0 -- variables de dato
var byte bogas2 = 0
var byte bogas3 = 0
var byte bogas4 = 0
var byte bogas5 = 0
var byte bogas6 = 0
```

```
var byte bogas1v = 0 -- variables de dato
var byte bogas2v = 0
var byte bogas3v = 0
var byte bogas4v = 0
var byte bogas5v = 0
var byte bogas6v = 0
```

```
var byte cadena = 0 -- banderas de dato
```

```
var byte m1 = 0 -- variables memoria
var byte m12 = 0
```

```
var byte m2 = 0
var byte m22 = 0
```

```
var byte m3 = 0
var byte m32 = 0
```

```
var byte m4 = 0
var byte m42 = 0
```

```
var byte m5 = 0
var byte m52 = 0
```

```
var byte r = 0 -- cuenta de memorias
var byte k = 0 -- revision de memorias
```

```
var byte u
var byte v
```

```
procedure escribe ( byte in out u, byte in out v ) is
  hd44780_position ( 72 ) -- escribo valor de bogas
  hd44780 = (u / 100) + 48 -- centenas
  hd44780 = (( u - ( u / 100 ) * 100 ) / 10 ) + 48 -- decenas
  hd44780 = (u - ((u / 10 ) * 10)) + 48 -- unidades
  hd44780_position ( 8 ) -- escribo valor de fuerza
  hd44780 = (v / 100) + 48 -- centenas
  hd44780 = (( v - ( v / 100 ) * 100 ) / 10 ) + 48 -- decenas
  hd44780 = (v - ((v / 10 ) * 10)) + 48 -- unidades
end procedure
```

```
hd44780_clear -- inicializacion
```

```
hd44780_position ( 6 )
hd44780 = "E" -- verifico que entremos al programa
hd44780 = "S"
hd44780 = "P"
hd44780 = "E"
hd44780 = "R"
hd44780 = "E"
pin_a0 = on
pin_a2 = on
pin_a3 = on -- chequeo de leds y lcd
```

```
delay_500ms ( 2 )
```

```
pin_a0 = off
pin_a2 = off
pin_a3 = off
```

hd44780_clear -- inicializacion**-- inicio****hd44780_position (67) -- c1****hd44780 = 66 -- "B"****hd44780_position (3) -- 81****hd44780 = 70 -- "F"****var bit bit_llegada at dato : 7 -- defino el bit de llegada en el bit 7 de dato****forever loop -- espero llegada de datos protocolo = (fin)1_9bits_0(inicio)****-- *****inicio a recibir caracter*********while valido == off loop****pin_a0 = off****if pin_a1 == on then -- si llega un 0 checa si es bit de inicio****delay_1us (200)****if pin_a1 == on then -- es el bit de inicio valido****for 8 loop****delay_1us (200)****delay_1us (200)****dato = dato >> 1****bit_llegada = pin_a1****end loop****delay_1us (200)****delay_1us (200)****if pin_a1 == on then -- bit de parada valido****valido = on****pin_a0 = on****end if****end if -- fin del bit de inicio valido****end if -- fin del bit de inicio****end loop****valido = off****-- ***** fin de recepcion de caracter *********if dato == 66 then**

```
if          cadena = 1 -- empieza recepcion de la cadena          end
    elsif cadena == 1 then
        bogas1 = dato
        cadena = 2

    elsif cadena == 2 then
        bogas2 = dato
        cadena = 3

    elsif cadena == 3 then
        bogas3 = dato
        cadena = 4

    elsif cadena == 4 then
        bogas4 = dato
        cadena = 5

    elsif cadena == 5 then
        bogas5 = dato
        cadena = 6

    elsif cadena == 6 then
        bogas6 = dato
        cadena = 7
    elsif cadena == 7 then
        cadena = 0

        if dato == 70 then

            bogas1v = bogas1 -- variables de dato
            bogas2v = bogas2
            bogas3v = bogas3
            bogas4v = bogas4
            bogas5v = bogas5
            bogas6v = bogas6

            hd44780_position ( 68 ) -- escribo valor de bogas
            hd44780 = bogas1v
            hd44780 = bogas2v
            hd44780 = bogas3v
            hd44780_position ( 4 ) -- escribo valor de fuerza
            hd44780 = bogas4v
            hd44780 = bogas5v
            hd44780 = bogas6v

        end if
    end if
```



```
        end if

if (r == 200) & (pin_b0 == off) then -- borro memorias
    while pin_b0 == off loop asm nop end loop -- efecto rebote

    m1 = 0 -- variables memoria
    m12 = 0
    m2 = 0
    m22 = 0
    m3 = 0
    m32 = 0
    m4 = 0
    m42 = 0
    m5 = 0
    m52 = 0
        r = 0
    hd44780_position ( 76 )
    hd44780 = 48 -- & pregunta de borrar

end if -- fin borro memorias

if pin_b0 == off then -- inicio captura de memorias
    while pin_b0 == off loop asm nop end loop -- efecto rebote

    r = r + 1

    if r == 1 then -- seleccion memoria
        m1 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
        m12 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)
        hd44780_position ( 76 )
            hd44780 = 49 -- 1
        elsif r == 2 then

            hd44780_position ( 76 )
                hd44780 = 50 -- 2

        m2 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
        m22 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

            elsif r == 3 then
                hd44780_position ( 76 )
                    hd44780 = 51 -- 3

            m3 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
            m32 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

            elsif r == 4 then

                hd44780_position ( 76 )
                    hd44780 = 52 -- 4
```

```

m4 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m42 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

```

```

elseif r == 5 then
hd44780_position ( 76 )
hd44780 = 53 -- 5

```

```

m5 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m52 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

```

```

end if -- fin seleccion memoria
end if -- fin captura de memorias

```

```

if pin_b1 == off then -- inicio revision de memorias

```

```

while pin_b1 == off loop asm nop end loop -- efecto rebote

```

```

k = k + 1

```

```

if k == 1 then -- seleccion memoria

```

```

hd44780_position ( 12 )

```

```

hd44780 = 49 -- 1

```

```

hd44780 = 32 -- 1

```

```

escribe ( m1, m12 )

```

```

elseif k == 2 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 50 -- 2

```

```

escribe ( m2, m22 )

```

```

elseif k == 3 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 51 -- 3

```

```

escribe ( m3, m32 )

```

```

elseif k == 4 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 52 -- 4

```

```

escribe ( m4, m42 )

```

```

elseif k == 5 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 53 -- 5

```

```

escribe ( m5, m52 )

```

```

elseif k == 6 then

```

```

hd44780_position ( 13 )

```

```

hd44780 = 38 -- & pregunta de borrar

```

```

r = 200

```

```

k = 0

```

```

end if -- fin seleccion memoria

```

```

end if -- fin revision memorias

```

```

end loop

```

9.6 CÓDIGO FUENTE DE LOS PROGRAMAS t1.asm, tbajo.asm, t2.asm

Estos programas se realizaron con las subrutinas descritas, por lo cual se omiten y solo se deja la parte funcional del programa.

****T1.ASM**

****REGISTRA EL TIEMPO EN ALTO DE LA SEÑAL PROVENIENTE DEL ACELEROMETRO**

****REGISTROS CAP1=\$1010-\$1011**

**** CAP2=\$1012-\$1013**

*******REGISTROS*******

Idx #\$1000

bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg

bset \$22,x,\$06 ;habilito capturadores 1 y 2

bset \$21,x,\$24 ;flancos de bajada cap1 y de subida cap2

bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******banderas y valoresiniciales*******

Idaa #\$aa ;numero de muestras= 170 decimal

staa \$20 ;guardo ese valor en esta direcciòn

Idx #\$0000

stx \$17 ;suma de muestras en esta dir.

*******PROGRAMA*******

inicio

division Idd \$00 ;valor de la resta

Idx #\$a ;divido entre 10 decimalresiduo en D

idiv

stx \$02 ;resultado en x residuo en D

stab \$09 ;UNIDADES

Idd \$02

Idx #\$a

idiv

stx \$02 ;resultado en x residuo en d

stab \$08 ;DECENAS

Idd \$02

Idx #\$a

idiv

stx \$02

stab \$07 ;CENTENAS

Idd \$02

Idx #\$a

```

idiv
stx $02
stab $06 ;UNIDADES MILES

```

```

ldd $02
ldx #$a
idiv
stx $02
stab $05 ;DECENAS MILES

```

```

ldaa #$87 ;posicion del cursor del display
jsr coman
escribe ldx #$05 ;carga datos de las direcciones resultado
ldab #$05 ;y despliega en el display
ya ldaa $0,x ;puntero de resultados
adda #$30 ;se le suman 30 para que despliegue ASCII
jsr dato
inx
dec b
bne ya

jsr mas

bra inicio

```

*****ROUTINAS DE INTERRUP. CAP1 Y CAP2*****

```

cap1 ldaa #$04
staa $1023 ;borro bandera de cap1

```

```

ldd $1010 ;valor flanco de bajada
cpd $1012
bhs resta

```

```

ldd $1012
subd $1010
std $15
bra sigue

```

```

resta ldd $1010
subd $1012 ;valor flanco de subida
std $15 ;guardo

```

```

sigue ldd $15
addd $17 ;sumo el valor a esta dir.
std $17
ldaa $20 ;sumo 150 veces
deca
staa $20
bne scap1 ;salta a scap1(sale de interrup) si no es cero

```

```
Idd $17
Idx #$aa ;divido la suma entre 170 muestras (promedio)
idiv
stx $00 ;resultado en x
std $13 ;residuo en D
Idaa #$aa ;pongo la cuenta de 150 muestras otra vez
staa $20
Idx #$00 ;pongo a ceros la suma de muestras
stx $17
```

```
scap1    rti
```

```
cap2    Idaa #$02
        staa $1023 ;borro bandera de capturador2
```

```
        rti
```

```
*****FIN INTS. CAP1 Y CAP2*****
```

```
mas     pshx
        psha
        Idx #$ff
si
        jsr delay
        jsr delay

        dex
        bne si
        pula
        pulx
        rts
```

****TBAJO.ASM**
****REGISTRA EL TIEMPOO EN BAJO DE LA SEÑAL**
****PROVENIENTE DEL ACELERÓMETRO**

*******REGISTROS*******

Idx # \$1000
bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg
bset \$22,x,\$06 ;habilito capturadores 1 y 2
bset \$21,x,\$24 ;flancos de bajada cap1 y de subida cap2
bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******banderas y valoresiniciales*******

Idaa # \$72 ;numero de muestras= 115 decimal
staa \$20 ;guardo ese valor en esta direccion
Idx # \$0000
stx \$17 ;suma de muestras en esta dir.

*******PROGRAMA*******

inicio

division Idd \$00 ;valor de la resta
Idx # \$a ;divido entre 10 decimalresiduo en D
idiv
stx \$02 ;resultado en x residuo en D
stab \$09 ;UNIDADES

Idd \$02
Idx # \$a
idiv
stx \$02 ;resultado en x residuo en d
stab \$08 ;DECENAS

Idd \$02
Idx # \$a
idiv
stx \$02
stab \$07 ;CENTENAS

Idd \$02
Idx # \$a
idiv
stx \$02

stab \$06 ;UNIDADES MILES

Idd \$02

Idx #\$a

idiv

stx \$02

stab \$05 ;DECENAS MILES

Idaa #\$87 ;posicion del cursor del display

jsr coman

escribe Idx #\$05 ;carga datos de las direcciones resultado

ldab #\$05 ;y despliega en el display

ya Idaa \$0,x ;puntero de resultados

adda #\$30 ;se le suman 30 para que despliegue ASCII

jsr dato

inx

decb

bne ya

jsr mas

bra inicio

*****RUTINAS DE INTERRUP. CAP1 Y CAP2*****

cap1 Idaa #\$04

staa \$1023 ;borro bandera de cap1

rti

cap2 Idaa #\$02

staa \$1023 ;borro bandera de capturador2

Idd \$1012 ;valor flanco de subida

subd \$1010 ;valor flanco de bajada

std \$15 ;guardo

Idd \$15

addd \$17 ;sumo el valor a esta dir.

std \$17

Idaa \$20 ;sumo 150 veces

deca

staa \$20

bne scap2 ;salta a scap1(sale de interrup) si no es cero

Idd \$17

Idx #\$72 ;divido la suma entre 170 muestras (promedio)

```
    idiv
    stx $00 ;resultado en x
    std $13 ;residuo en D
    ldaa #$72 ;pongo la cuenta de 150 muestras otra vez
    staa $20
    ldx #$00 ;pongo a ceros la suma de muestras
    stx $17
scap2    rti
```

```
*****FIN INTS. CAP1 Y CAP2*****
```

```
mas     pshx
        psha
        ldx #$ff
si
        jsr delay
        jsr delay

        dex
        bne si
        pula
        pulx
        rts
```


****T2.ASM**

****REGISTRA EL PERIODO DE UNA SEÑAL PROVENIENTE DEL ACELEROMETRO**

****REGISTROS CAP1=\$1010-\$1011**

**** CAP2=\$1012-\$1013**

*******REGISTROS*******

Idx # \$1000

bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg

bset \$22,x,\$02 ;habilito capturador 2

bset \$21,x,\$04 ;flancos de subida cap2

bset \$20,x,\$00 ;no hay salida hardware del comparador

bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******valores*******

Idaa # \$00

staa \$10 ; bandera de periodo 2

*******banderas y valores iniciales*******

Idaa # \$00

staa \$10 ;bandera periodo

staa \$35 ;bandera promedio

staa \$20 ;valor 1ra captura

staa \$22 ;valor periodo

staa \$25 ;valor suma

staa \$30 ;bandera cuenta

*******PROGRAMA*******

Idaa # \$80

jsr coman

Idaa # \$35

jsr dato

inicio Idaa \$35

cmpa # \$ff

beq calcula

bra inicio

calcula Idaa # \$00

staa \$35

ldd \$25

ldx # \$30

idiv ;resultado en x residuo en d

stx \$40

Idd \$40

jsr division
 ldaa #\$85
 jsr escribe

ldd #\$ffff
 jsr division
 ldaa #\$c5
 jsr escribe
 ldx #\$0000
 stx \$25 ;borro suma

bra inicio

*****FIN PROGRAMA*****

*****SUBROUTINAS*****

*****division*****

division psha
 pshx
 pshb
 ldx #\$a ;divido entre 10 decimalresiduo en D
 idiv
 stx \$02 ;resultado en x residuo en D
 stab \$09 ;UNIDADES

ldd \$02
 ldx #\$a
 idiv
 stx \$02 ;resultado en x residuo en d
 stab \$08 ;DECENAS

ldd \$02
 ldx #\$a
 idiv
 stx \$02
 stab \$07 ;CENTENAS

ldd \$02
 ldx #\$a
 idiv
 stx \$02
 stab \$06 ;unidades millares

ldd \$02
 ldx #\$a
 idiv

```

stx $02
stab $05 ;decenas millares
pulb
pulx
pula
rts

```

*****fin division*****

*****escribe 5 lugares*****

```

escribe  psha  ;posicion del cursor del display
        pshx
        pshb
        jsr coman
        ldx #$05 ;carga datos de las direcciones resultado
        ldab #$05 ;y despliega en el display
ya      ldaa $0,x ;puntero de resultados
        adda #$30 ;se le suman 30 para que despliegue ASCII
        jsr dato
        inx
        decb
        bne ya
        pulb
        pulx
        pula
        rts

```

*****fin escribe*****

*****RUTINAS DE INTERRUP. CAP1 Y CAP2*****

```

cap1    ldaa #$ff
        staa $1023 ;borro bandera de cap1
        rti

```

```

cap2    ldaa #$ff
        staa $1023 ;borro bandera de capturador2

```

```

        ldaa $10
        cmpa #$00
        beq periodo ;checo si es el inicio del periodo

```

```

        ldaa #$00
        staa $10

```

```

        ldd $1012
        subd $20

```

std \$22 ;guardo valor de periodo

ldd \$25

add \$22

std \$25 ; guardo valor suma

ldaa \$30 ;cuenta

cmpa #30 ;48 veces

beq sal

inca

staa \$30

bra salcap2

sal ;borro cuenta de suma

ldaa #ff

staa \$35 ;bandera de cuenta

ldaa #00

staa \$30

bra salcap2

periodo ldd \$1012 ;cargo con valor cap2

std \$20 ;guardo valor cap2

ldaa #ff

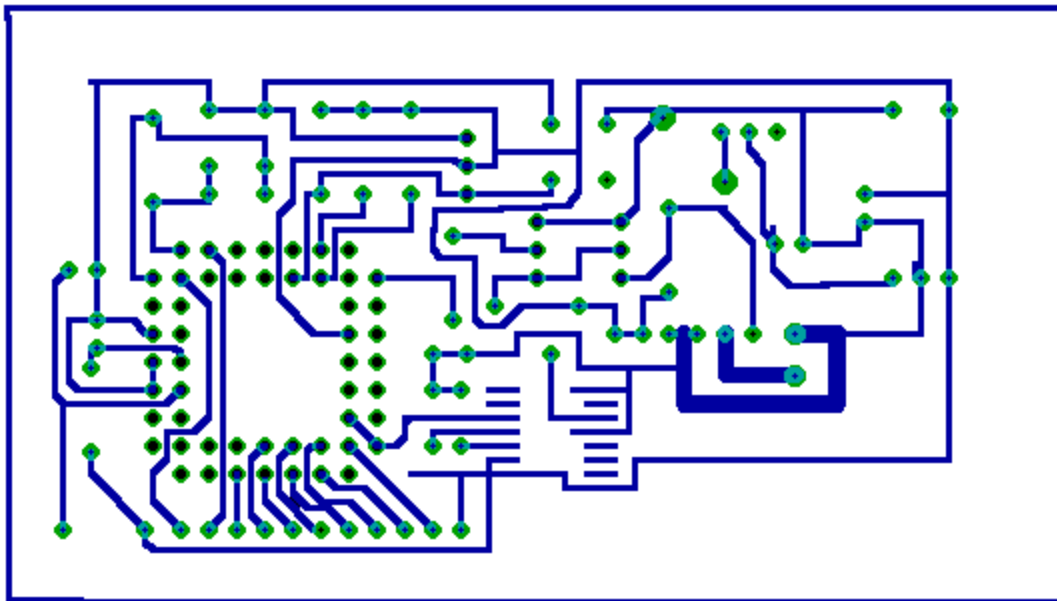
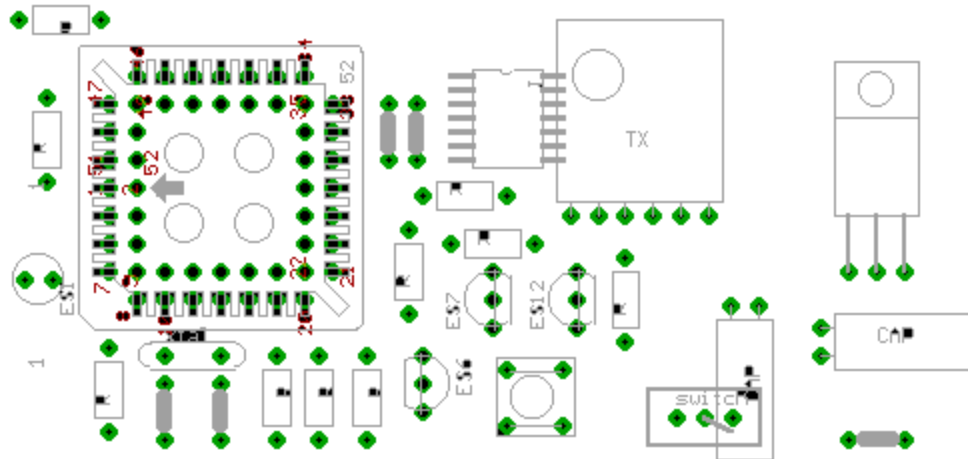
staa \$10 ;bandera periodo

bra salcap2

salcap2 rti

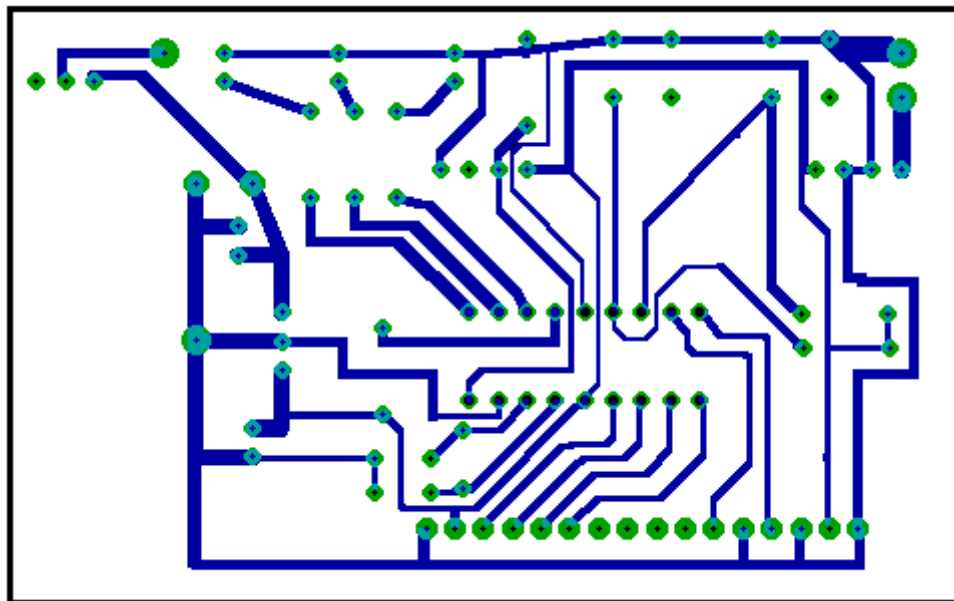
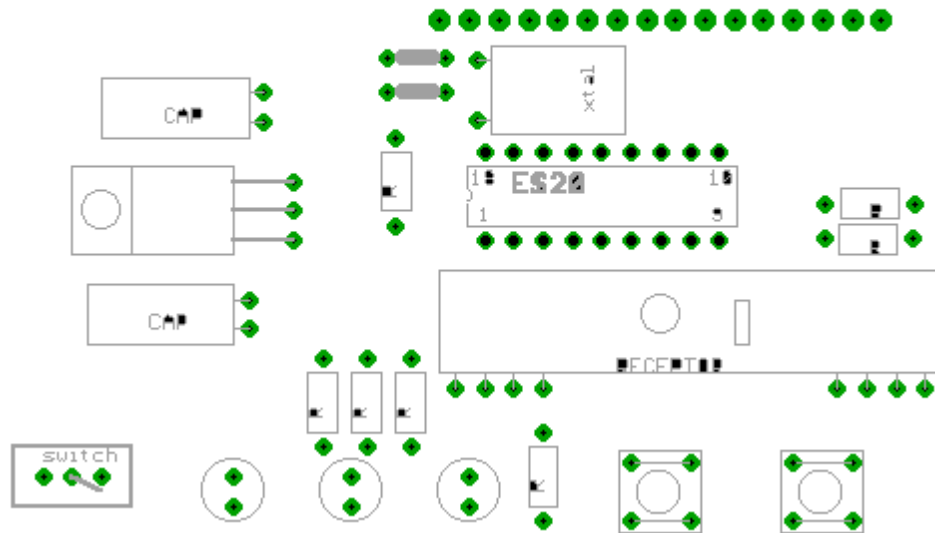
*****FIN INTS. CAP1 Y CAP2*****

9.7 CIRCUITO IMPRESO DEL TRANSMISOR

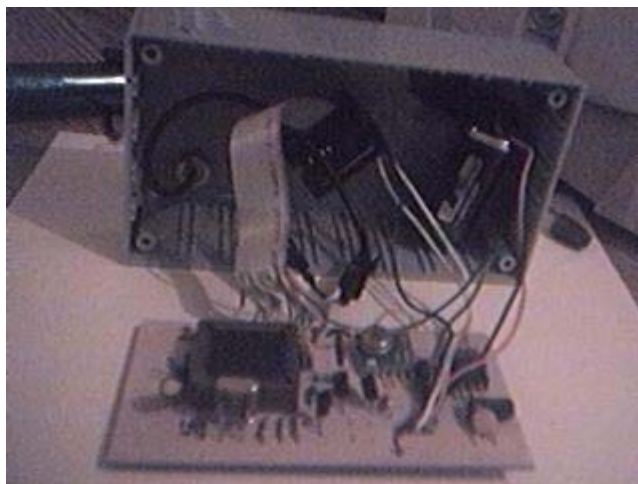
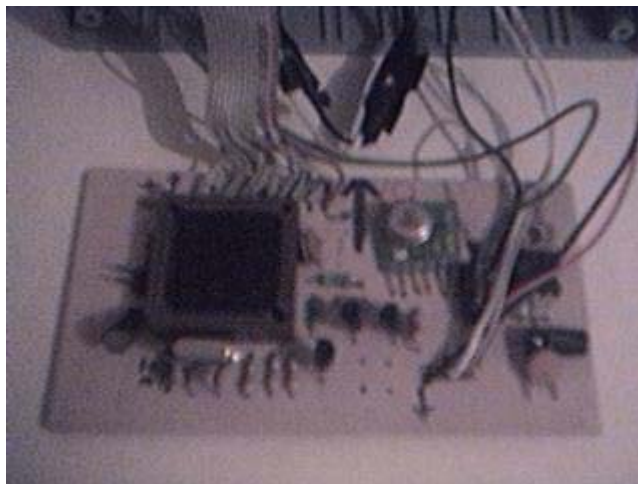
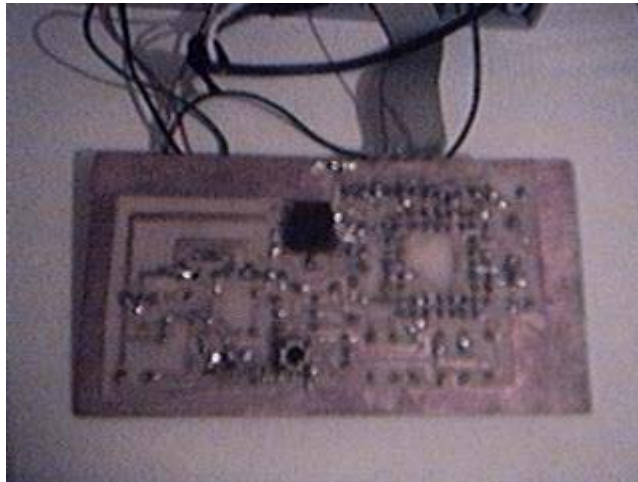


9.8 CIRCUITO IMPRESO DEL RECEPTOR

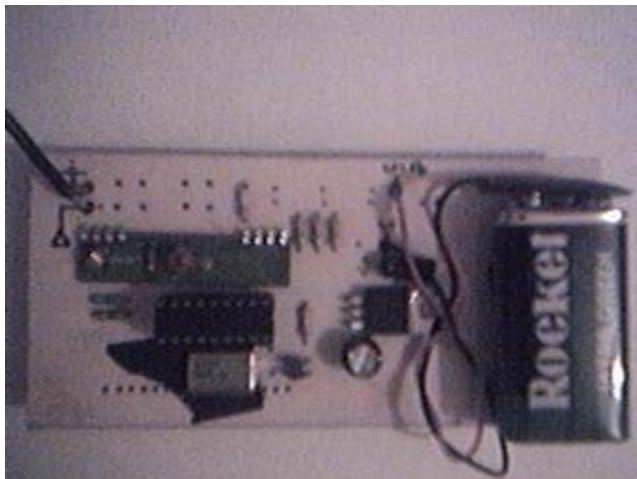
Circuio Impreso "Receptor" ubicacion de componentes



9.9 IMÁGENES DEL CIRCUITO IMPRESO ENSAMBLADO DEL TRANSMISOR



9.10 IMÁGENES DEL CIRCUITO IMPRESO ENSAMBLADO DEL RECEPTOR



9.11 HOJAS DE ESPECIFICACIONES DE LOS DISPOSITIVOS EMPLEADOS



Low Cost $\pm 2 g/\pm 10 g$ Dual Axis iMEMS[®] Accelerometers with Digital Output

ADXL202/ADXL210

FEATURES

2-Axis Acceleration Sensor on a Single IC Chip
Measures Static Acceleration as Well as Dynamic Acceleration
Duty Cycle Output with User Adjustable Period
Low Power <0.6 mA
Faster Response than Electrolytic, Mercury or Thermal Tilt Sensors
Bandwidth Adjustment with a Single Capacitor Per Axis
5 mg Resolution at 60 Hz Bandwidth
+3 V to +5.25 V Single Supply Operation
1000 g Shock Survival

APPLICATIONS

2-Axis Tilt Sensing
Computer Peripherals
Inertial Navigation
Seismic Monitoring
Vehicle Security Systems
Battery Powered Motion Sensing

GENERAL DESCRIPTION

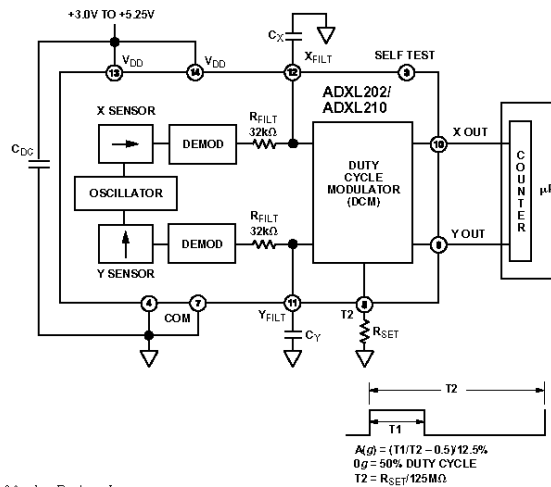
The ADXL202/ADXL210 are low cost, low power, complete 2-axis accelerometers with a measurement range of either $\pm 2 g/\pm 10 g$. The ADXL202/ADXL210 can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are digital signals whose duty cycles (ratio of pulse-width to period) are proportional to the acceleration in each of the 2 sensitive axes. These outputs may be measured directly with a microprocessor counter, requiring no A/D converter or glue logic. The output period is adjustable from 0.5 ms to 10 ms via a single resistor (R_{SET}). If a voltage output is desired, a voltage output proportional to acceleration is available from the X_{FILT} and Y_{FILT} pins, or may be reconstructed by filtering the duty cycle outputs.

The bandwidth of the ADXL202/ADXL210 may be set from 0.01 Hz to 5 kHz via capacitors C_X and C_Y . The typical noise floor is $500 \mu g/\sqrt{Hz}$ allowing signals below 5 mg to be resolved for bandwidths below 60 Hz.

The ADXL202/ADXL210 is available in a hermetic 14-lead Surface Mount CERPAK, specified over the $0^\circ C$ to $+70^\circ C$ commercial or $-40^\circ C$ to $+85^\circ C$ industrial temperature range.

FUNCTIONAL BLOCK DIAGRAM



iMEMS is a registered trademark of Analog Devices, Inc.

REV. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 1999

ADXL202/ADXL210—SPECIFICATIONS ($T_A = T_{MIN}$ to T_{MAX} , $T_A = +25^\circ\text{C}$ for J Grade only, $V_{DD} = +5\text{V}$, $R_{SET} = 125\text{ k}\Omega$, Acceleration = 0 g, unless otherwise noted)

Parameter	Conditions	ADXL202/JQC/AQC			ADXL210/JQC/AQC			Units
		Min	Typ	Max	Min	Typ	Max	
SENSOR INPUT	Each Axis							
Measurement Range ¹		± 1.5	± 2		± 8	± 10		g
Nonlinearity	Best Fit Straight Line		0.2			0.2		% of FS
Alignment Error ²			± 1			± 1		Degrees
Alignment Error	X Sensor to Y Sensor		± 0.01			± 0.01		Degrees
Transverse Sensitivity ³			± 2			± 2		%
SENSITIVITY	Each Axis							
Duty Cycle per g	T1/T2 @ +25°C	10	12.5	15	3.2	4.0	4.8	%/g
Sensitivity, Analog Output	At Pins X _{FILT} , Y _{FILT}		312			100		mV/g
Temperature Drift ⁴	Δ from +25°C		± 0.5			± 0.5		% Rdg
ZERO g BIAS LEVEL	Each Axis							
0 g Duty Cycle	T1/T2	25	50	75	42	50	58	%
Initial Offset			± 2			± 2		g
0 g Duty Cycle vs. Supply			1.0	4.0		1.0	4.0	%/V
0 g Offset vs. Temperature ⁴	Δ from +25°C		2.0			2.0		mg/°C
NOISE PERFORMANCE								
Noise Density ⁵	@ +25°C		500	1000		500	1000	$\mu\text{g}/\sqrt{\text{Hz}}$
FREQUENCY RESPONSE								
3 dB Bandwidth	Duty Cycle Output		500			500		Hz
3 dB Bandwidth	At Pins X _{FILT} , Y _{FILT}		5			5		kHz
Sensor Resonant Frequency			10			14		kHz
FILTER								
R _{FILT} Tolerance	32 k Ω Nominal		± 15			± 15		%
Minimum Capacitance	At X _{FILT} , Y _{FILT}	1000			1000			pF
SELF TEST								
Duty Cycle Change	Self-Test "0" to "1"		10			10		%
DUTY CYCLE OUTPUT STAGE								
F _{SET}		125 M Ω /R _{SET}			125 M Ω /R _{SET}			
R _{SET} Tolerance	R _{SET} = 125 k Ω	0.7		1.3	0.7		1.3	kHz
Output High Voltage	I = 25 μA	V _S - 200 mV			V _S - 200 mV			mV
Output Low Voltage	I = 25 μA			200			200	mV
T2 Drift vs. Temperature			35			35		ppm/°C
Rise/Fall Time			200			200		ns
POWER SUPPLY								
Operating Voltage Range		3.0		5.25	2.7		5.25	V
Specified Performance		4.75		5.25	4.75		5.25	V
Quiescent Supply Current			0.6	1.0		0.6	1.0	mA
Turn-On Time ⁶	To 99%	160 C _{FILT} + 0.3			160 C _{FILT} + 0.3			ms
TEMPERATURE RANGE								
Operating Range	JQC	0		+70	0		+70	°C
Specified Performance	AQC	-40		+85	-40		+85	°C

NOTES

¹For all combinations of offset and sensitivity variation.

²Alignment error is specified as the angle between the true and indicated axis of sensitivity.

³Transverse sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.

⁴Specification refers to the maximum change in parameter from its initial at +25°C to its worst case value at T_{MIN} to T_{MAX}.

⁵Noise density ($\mu\text{g}/\sqrt{\text{Hz}}$) is the average noise at any frequency in the bandwidth of the part.

⁶C_{FILT} in μF . Addition of filter capacitor will increase turn on time. Please see the Application section on power cycling.

All min and max specifications are guaranteed. Typical specifications are not tested or guaranteed.

Specifications subject to change without notice.

ADXL202/ADXL210

ABSOLUTE MAXIMUM RATINGS*

Acceleration (Any Axis, Unpowered for 0.5 ms)	1000 g
Acceleration (Any Axis, Powered for 0.5 ms)	500 g
+V _S	-0.3 V to +7.0 V
Output Short Circuit Duration (Any Pin to Common)	Indefinite
Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; the functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Drops onto hard surfaces can cause shocks of greater than 1000 g and exceed the absolute maximum rating of the device. Care should be exercised in handling to avoid damage.

PIN FUNCTION DESCRIPTIONS

Pin	Name	Description
1	NC	No Connect
2	V _{TP}	Test Point, Do Not Connect
3	ST	Self Test
4	COM	Common
5	T2	Connect R _{SET} to Set T2 Period
6	NC	No Connect
7	COM	Common
8	NC	No Connect
9	Y _{OUT}	Y Axis Duty Cycle Output
10	X _{OUT}	X Axis Duty Cycle Output
11	Y _{FILT}	Connect Capacitor for Y Filter
12	X _{FILT}	Connect Capacitor for X Filter
13	V _{DD}	+3 V to +5.25 V, Connect to 14
14	V _{DD}	+3 V to +5.25 V, Connect to 13

PACKAGE CHARACTERISTICS

Package	θ _{JA}	θ _{JC}	Device Weight
14-Lead CERPAK	110°C/W	30°C/W	5 Grams

PIN CONFIGURATION

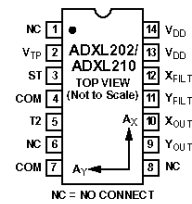


Figure 1 shows the response of the ADXL202 to the Earth's gravitational field. The output values shown are nominal. They are presented to show the user what type of response to expect from each of the output pins due to changes in orientation with respect to the Earth. The ADXL210 reacts similarly with output changes appropriate to its scale.

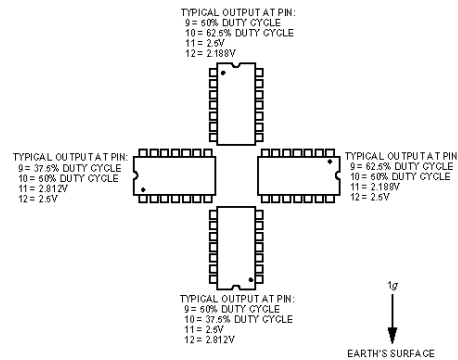


Figure 1. ADXL202/ADXL210 Nominal Response Due to Gravity

ORDERING GUIDE

Model	g Range	Temperature Range	Package Description	Package Option
ADXL202JQC	±2	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL202AQC	±2	-40°C to +85°C	14-Lead CERPAK	QC-14
ADXL210JQC	±10	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL210AQC	±10	-40°C to +85°C	14-Lead CERPAK	QC-14

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADXL202/ADXL210 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



ADXL202/ADXL210

TYPICAL CHARACTERISTICS (@ +25°C $R_{SET} = 125\text{ k}\Omega$, $V_{DD} = +5\text{ V}$, unless otherwise noted)

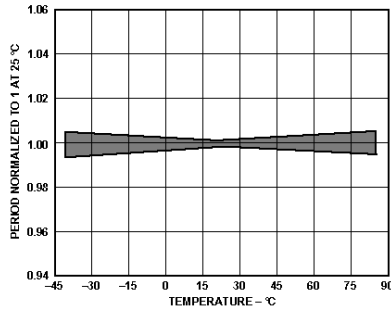


Figure 2. Normalized DCM Period (T_2) vs. Temperature

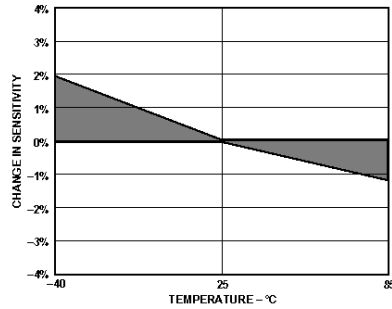


Figure 5. Typical X Axis Sensitivity Drift Due to Temperature

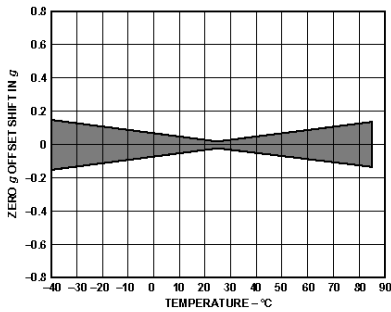


Figure 3. Typical Zero g Offset vs. Temperature

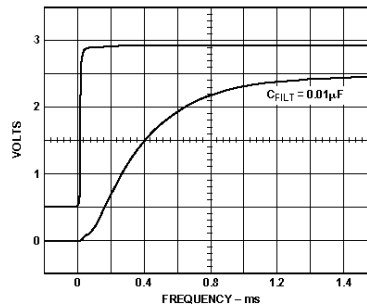


Figure 6. Typical Turn-On Time

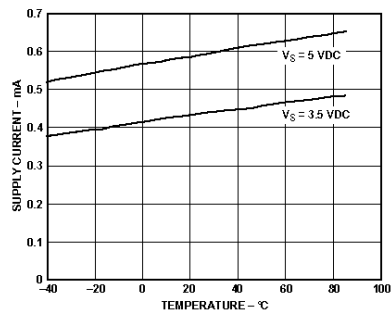


Figure 4. Typical Supply Current vs. Temperature

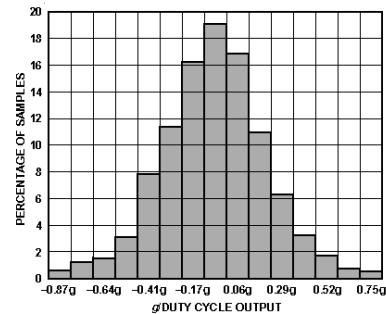


Figure 7. Typical Zero g Distribution at +25°C

ADXL202/ADXL210

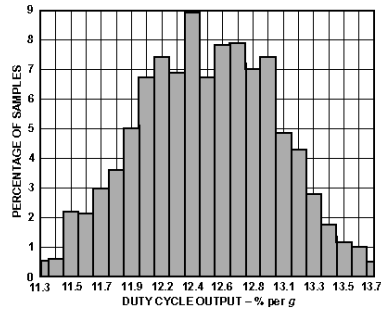


Figure 8. Typical Sensitivity per g at +25°C

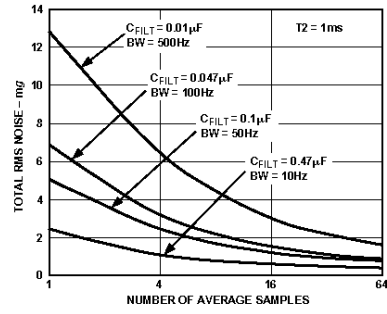


Figure 10. Typical Noise at Digital Outputs



Figure 9. Typical Noise at X_{FILT} Output

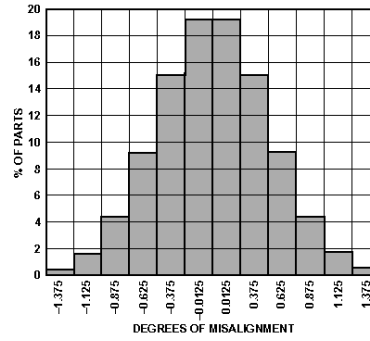


Figure 11. Rotational Die Alignment

ADXL202/ADXL210

DEFINITIONS

T1	Length of the “on” portion of the cycle.
T2	Length of the total cycle.
Duty Cycle	Ratio of the “on” time (T1) of the cycle to the total cycle (T2). Defined as T1/T2 for the ADXL202/ADXL210.
Pulsewidth	Time period of the “on” pulse. Defined as T1 for the ADXL202/ADXL210.

THEORY OF OPERATION

The ADXL202/ADXL210 are complete dual axis acceleration measurement systems on a single monolithic IC. They contain a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open loop acceleration measurement architecture. For each axis, an output circuit converts the analog signal to a duty cycle modulated (DCM) digital signal that can be decoded with a counter/timer port on a microprocessor. The ADXL202/ADXL210 are capable of measuring both positive and negative accelerations to a maximum level of $\pm 2 g$ or $\pm 10 g$. The accelerometer measures static acceleration forces such as gravity, allowing it to be used as a tilt sensor.

The sensor is a surface micromachined polysilicon structure built on top of the silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and central plates attached to the moving mass. The fixed plates are driven by 180° out of phase square waves. An acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration.

The output of the demodulator drives a duty cycle modulator (DCM) stage through a $32 k\Omega$ resistor. At this point a pin is available on each channel to allow the user to set the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

After being low-pass filtered, the analog signal is converted to a duty cycle modulated signal by the DCM stage. A single resistor sets the period for a complete cycle (T2), which can be set between 0.5 ms and 10 ms (see Figure 12). A $0 g$ acceleration produces a nominally 50% duty cycle. The acceleration signal can be determined by measuring the length of the T1 and T2 pulses with a counter/timer or with a polling loop using a low cost microcontroller.

An analog output voltage can be obtained either by buffering the signal from the X_{FILT} and Y_{FILT} pin, or by passing the duty cycle signal through an RC filter to reconstruct the dc value.

The ADXL202/ADXL210 will operate with supply voltages as low as 3.0 V or as high as 5.25 V.

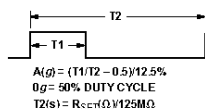


Figure 12. Typical Output Duty Cycle

APPLICATIONS

POWER SUPPLY DECOUPLING

For most applications a single $0.1 \mu\text{F}$ capacitor, C_{DC}, will adequately decouple the accelerometer from signal and noise on the power supply. However, in some cases, especially where digital devices such as microcontrollers share the same power supply, digital noise on the supply may cause interference on the ADXL202/ADXL210 output. This is often observed as a slowly undulating fluctuation of voltage at X_{FILT} and Y_{FILT}. If additional decoupling is needed, a 100Ω (or smaller) resistor or ferrite beads, may be inserted in the ADXL202/ADXL210's supply line.

DESIGN PROCEDURE FOR THE ADXL202/ADXL210

The design procedure for using the ADXL202/ADXL210 with a duty cycle output involves selecting a duty cycle period and a filter capacitor. A proper design will take into account the application requirements for bandwidth, signal resolution and acquisition time, as discussed in the following sections.

V_{DD}

The ADXL202/ADXL210 have two power supply (V_{DD}) Pins: 13 and 14. These two pins should be connected directly together.

COM

The ADXL202/ADXL210 have two commons, Pins 4 and 7. These two pins should be connected directly together and Pin 7 grounded.

V_{TP}

This pin is to be left open; make no connections of any kind to this pin.

Decoupling Capacitor C_{DC}

A $0.1 \mu\text{F}$ capacitor is recommended from V_{DD} to COM for power supply decoupling.

ST

The ST pin controls the self-test feature. When this pin is set to V_{DD}, an electrostatic force is exerted on the beam of the accelerometer. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output will be 10% at the duty cycle outputs (corresponding to 800 mg). This pin may be left open circuit or connected to common in normal use.

Duty Cycle Decoding

The ADXL202/ADXL210's digital output is a duty cycle modulator. Acceleration is proportional to the ratio T1/T2. The nominal output of the ADXL202 is:

$$0 g = 50\% \text{ Duty Cycle}$$

Scale factor is 12.5% Duty Cycle Change per g

The nominal output of the ADXL210 is:

$$0 g = 50\% \text{ Duty Cycle}$$

Scale factor is 4% Duty Cycle Change per g

These nominal values are affected by the initial tolerance of the device including zero g offset error and sensitivity error.

T2 does not have to be measured for every measurement cycle. It need only be updated to account for changes due to temperature, (a relatively slow process). Since the T2 time period is shared by both X and Y channels, it is necessary only to measure it on one channel of the ADXL202/ADXL210. Decoding algorithms for various microcontrollers have been developed. Consult the appropriate Application Note.

ADXL202/ADXL210

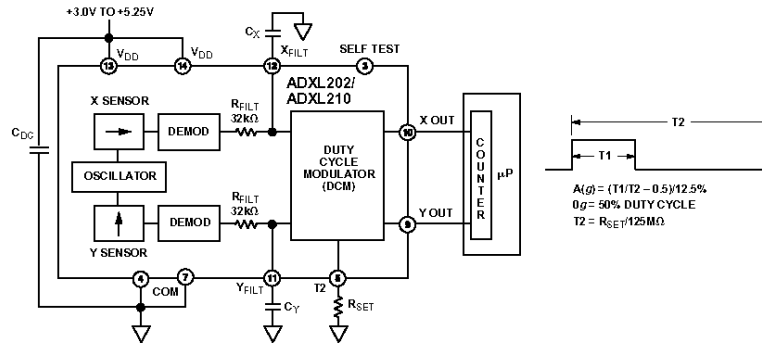


Figure 13. Block Diagram

Setting the Bandwidth Using C_X and C_Y

The ADXL202/ADXL210 have provisions for bandlimiting the X_{FILT} and Y_{FILT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is:

$$F_{-3dB} = \frac{1}{(2\pi (32\text{ k}\Omega) \times C(x,y))}$$

or, more simply, $F_{-3dB} = \frac{5\ \mu F}{C_{(x,y)}}$

The tolerance of the internal resistor (R_{FILT}), can vary as much as ±25% of its nominal value of 32 kΩ; so the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for C_X, C_Y is required in all cases.

Table I. Filter Capacitor Selection, C_X and C_Y

Bandwidth	Capacitor Value
10 Hz	0.47 μF
50 Hz	0.10 μF
100 Hz	0.05 μF
200 Hz	0.027 μF
500 Hz	0.01 μF
5 kHz	0.001 μF

Setting the DCM Period with R_{SET}

The period of the DCM output is set for both channels by a single resistor from R_{SET} to ground. The equation for the period is:

$$T2 = \frac{R_{SET} (\Omega)}{125\text{ M}\Omega}$$

A 125 kΩ resistor will set the duty cycle repetition rate to approximately 1 kHz, or 1 ms. The device is designed to operate at duty cycle periods between 0.5 ms and 10 ms.

Table II. Resistor Values to Set T2

T2	R _{SET}
1 ms	125 kΩ
2 ms	250 kΩ
5 ms	625 kΩ
10 ms	1.25 MΩ

Note that the R_{SET} should always be included, even if only an analog output is desired. Use an R_{SET} value between 500 kΩ and 2 MΩ when taking the output from X_{FILT} or Y_{FILT}. The R_{SET} resistor should be placed close to the T2 Pin to minimize parasitic capacitance at this node.

Selecting the Right Accelerometer

For most tilt sensing applications the ADXL202 is the most appropriate accelerometer. Its higher sensitivity (12.5%/g allows the user to use a lower speed counter for PWM decoding while maintaining high resolution. The ADXL210 should be used in applications where accelerations of greater than ±2 g are expected.

MICROCOMPUTER INTERFACES

The ADXL202/ADXL210 were specifically designed to work with low cost microcontrollers. Specific code sets, reference designs, and application notes are available from the factory. This section will outline a general design procedure and discuss the various trade-offs that need to be considered.

The designer should have some idea of the required performance of the system in terms of:

- Resolution:* the smallest signal change that needs to be detected.
- Bandwidth:* the highest frequency that needs to be detected.
- Acquisition Time:* the time that will be available to acquire the signal on each axis.

These requirements will help to determine the accelerometer bandwidth, the speed of the microcontroller clock and the length of the T2 period.

When selecting a microcontroller it is helpful to have a counter timer port available. The microcontroller should have provisions for software calibration. While the ADXL202/ADXL210 are highly accurate accelerometers, they have a wide tolerance for

ADXL202/ADXL210

initial offset. The easiest way to null this offset is with a calibration factor saved on the microcontroller or by a user calibration for zero *g*. In the case where the offset is calibrated during manufacture, there are several options, including external EEPROM and microcontrollers with “one-time programmable” features.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The accelerometer bandwidth selected will determine the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor and improve the resolution of the accelerometer. Resolution is dependent on both the analog filter bandwidth at X_{FILT} and Y_{FILT} and on the speed of the microcontroller counter.

The analog output of the ADXL202/ADXL210 has a typical bandwidth of 5 kHz, much higher than the duty cycle stage is capable of converting. The user must filter the signal at this point to limit aliasing errors. To minimize DCM errors the analog bandwidth should be less than 1/10 the DCM frequency. Analog bandwidth may be increased to up to 1/2 the DCM frequency in many applications. This will result in greater dynamic error generated at the DCM.

The analog bandwidth may be further decreased to reduce noise and improve resolution. The ADXL202/ADXL210 noise has the characteristics of white Gaussian noise that contributes equally at all frequencies and is described in terms of μg per root Hz; i.e., the noise is proportional to the square root of the bandwidth of the accelerometer. It is recommended that the user limit bandwidth to the lowest frequency needed by the application, to maximize the resolution and dynamic range of the accelerometer.

With the single pole roll-off characteristic, the typical noise of the ADXL202/ADXL210 is determined by the following equation:

$$\text{Noise (rms)} = (500 \mu\text{g}/\sqrt{\text{Hz}}) \times (\sqrt{\text{BW} \times 1.5})$$

At 100 Hz the noise will be:

$$\text{Noise (rms)} = (500 \mu\text{g}/\sqrt{\text{Hz}}) \times (\sqrt{100 \times (1.5)}) = 6.12 \text{ mg}$$

Often the peak value of the noise is desired. Peak-to-peak noise can only be estimated by statistical methods. Table III is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table III. Estimation of Peak-to-Peak Noise

Nominal Peak-to-Peak Value	% of Time that Noise Will Exceed Nominal Peak-to-Peak Value
2.0 × rms	32%
4.0 × rms	4.6%
6.0 × rms	0.27%
8.0 × rms	0.006%

The peak-to-peak noise value will give the best estimate of the uncertainty in a single measurement.

Table IV gives typical noise output of the ADXL202/ADXL210 for various C_X and C_Y values.

Table IV. Filter Capacitor Selection, C_X and C_Y

Bandwidth	C_X, C_Y	rms Noise	Peak-to-Peak Noise Estimate 95% Probability (rms × 4)
10 Hz	0.47 μF	1.9 mg	7.6 mg
50 Hz	0.10 μF	4.3 mg	17.2 mg
100 Hz	0.05 μF	6.1 mg	24.4 mg
200 Hz	0.027 μF	8.7 mg	35.8 mg
500 Hz	0.01 μF	13.7 mg	54.8 mg

CHOOSING T2 AND COUNTER FREQUENCY: DESIGN TRADE-OFFS

The noise level is one determinant of accelerometer resolution. The second relates to the measurement resolution of the counter when decoding the duty cycle output.

The ADXL202/ADXL210's duty cycle converter has a resolution of approximately 14 bits; better resolution than the accelerometer itself. The actual resolution of the acceleration signal is, however, limited by the time resolution of the counting devices used to decode the duty cycle. The faster the counter clock, the higher the resolution of the duty cycle and the shorter the T2 period can be for a given resolution. The following table shows some of the trade-offs. It is important to note that this is the resolution due to the microprocessors's counter. It is probable that the accelerometer's noise floor may set the lower limit on the resolution, as discussed in the previous section.

Table V. Trade-Offs Between Microcontroller Counter Rate, T2 Period and Resolution of Duty Cycle Modulator

T2 (ms)	R _{SET} (k Ω)	ADXL202/ADXL210 Sample Rate	Counter-Clock Rate (MHz)	Counts per T2 Cycle	Counts per <i>g</i>	Resolution (mg)
1.0	124	1000	2.0	2000	250	4.0
1.0	124	1000	1.0	1000	125	8.0
1.0	124	1000	0.5	500	62.5	16.0
5.0	625	200	2.0	10000	1250	0.8
5.0	625	200	1.0	5000	625	1.6
5.0	625	200	0.5	2500	312.5	3.2
10.0	1250	100	2.0	20000	2500	0.4
10.0	1250	100	1.0	10000	1250	0.8
10.0	1250	100	0.5	5000	625	1.6

ADXL202/ADXL210

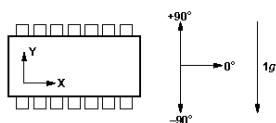
STRATEGIES FOR USING THE DUTY CYCLE OUTPUT WITH MICROCONTROLLERS

Application notes outlining various strategies for using the duty cycle output with low cost microcontrollers are available from the factory.

USING THE ADXL202/ADXL210 AS A DUAL AXIS TILT SENSOR

One of the most popular applications of the ADXL202/ADXL210 is tilt measurement. An accelerometer uses the force of gravity as an input vector to determine orientation of an object in space.

An accelerometer is most sensitive to tilt when its sensitive axis is perpendicular to the force of gravity, i.e., parallel to the earth's surface. At this orientation its sensitivity to changes in tilt is highest. When the accelerometer is oriented on axis to gravity, i.e., near its +1 g or -1 g reading, the change in output acceleration per degree of tilt is negligible. When the accelerometer is perpendicular to gravity, its output will change nearly 17.5 mg per degree of tilt, but at 45° degrees it is changing only at 12.2 mg per degree and resolution declines. The following table illustrates the changes in the X and Y axes as the device is tilted $\pm 90^\circ$ through gravity.



X AXIS ORIENTATION TO HORIZON (°)	X OUTPUT		Y OUTPUT (g)	
	X OUTPUT (g)	Δ PER DEGREE OF TILT (mg)	Y OUTPUT (g)	Δ PER DEGREE OF TILT (mg)
-90	-1.000	-0.2	0.000	17.5
-75	-0.966	4.4	0.259	16.9
-60	-0.866	8.6	0.500	15.2
-45	-0.707	12.2	0.707	12.4
-30	-0.500	15.0	0.866	8.9
-15	-0.259	16.8	0.966	4.7
0	0.000	17.5	1.000	0.2
15	0.259	16.9	0.966	-4.4
30	0.500	15.2	0.866	-8.6
45	0.707	12.4	0.707	-12.2
60	0.866	8.9	0.500	-15.0
75	0.966	4.7	0.259	-16.8
90	1.000	0.2	0.000	-17.5

Figure 14. How the X and Y Axes Respond to Changes in Tilt

A DUAL AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT

When the accelerometer is oriented so both its X and Y axes are parallel to the earth's surface it can be used as a two axis tilt sensor with a roll and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between -1 g and +1 g, the output tilt in degrees is calculated as follows:

$$\text{Pitch} = \text{ASIN}(Ax/1 \text{ g})$$

$$\text{Roll} = \text{ASIN}(Ay/1 \text{ g})$$

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than $\pm 1 \text{ g}$ due to vibration, shock or other accelerations.

MEASURING 360° OF TILT

It is possible to measure a full 360° of orientation through gravity by using two accelerometers oriented perpendicular to one another (see Figure 15). When one sensor is reading a maximum change in output per degree, the other is at its minimum.

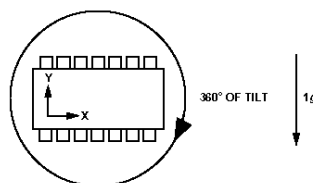


Figure 15. Using a Two-Axis Accelerometer to Measure 360° of Tilt

ADXL202/ADXL210

USING THE ANALOG OUTPUT

The ADXL202/ADXL210 was specifically designed for use with its digital outputs, but has provisions to provide analog outputs as well.

Duty Cycle Filtering

An analog output can be reconstructed by filtering the duty cycle output. This technique requires only passive components. The duty cycle period (T_2) should be set to 1 ms. An RC filter with a 3 dB point at least a factor of 10 less than the duty cycle frequency is connected to the duty cycle output. The filter resistor should be no less than 100 k Ω to prevent loading of the output stage. The analog output signal will be ratiometric to the supply voltage. The advantage of this method is an output scale factor of approximately double the analog output. Its disadvantage is that the frequency response will be lower than when using the X_{FILT} , Y_{FILT} output.

X_{FILT} , Y_{FILT} Output

The second method is to use the analog output present at the X_{FILT} and Y_{FILT} pin. Unfortunately, these pins have a 32 k Ω output impedance and are not designed to drive a load directly. An op amp follower may be required to buffer this pin. The advantage of this method is that the full 5 kHz bandwidth of the accelerometer is available to the user. A capacitor still must be added at this point for filtering. The duty cycle converter should be kept running by using $R_{SET} < 10 M\Omega$. Note that the accelerometer offset and sensitivity are ratiometric to the supply voltage. The offset and sensitivity are nominally:

$$\begin{aligned} 0g \text{ Offset} &= V_{DD}/2 && 2.5 \text{ V at } +5 \text{ V} \\ \text{ADXL202 Sensitivity} &= (60 \text{ mV} \times V_S)/g && 300 \text{ mV/g at } +5 \text{ V, } V_{DD} \\ \text{ADXL210 Sensitivity} &= (20 \text{ mV} \times V_S)/g && 100 \text{ mV/g at } +5 \text{ V, } V_{DD} \end{aligned}$$

USING THE ADXL202/ADXL210 IN VERY LOW POWER APPLICATIONS

An application note outlining low power strategies for the ADXL202/ADXL210 is available. Some key points are presented here. It is possible to reduce the ADXL202/ADXL210's average current from 0.6 mA to less than 20 μA by using the following techniques:

1. Power Cycle the accelerometer.
2. Run the accelerometer at a Lower Voltage, (Down to 3 V).

Power Cycling with an External A/D

Depending on the value of the X_{FILT} capacitor, the ADXL202/ADXL210 is capable of turning on and giving a good reading in 1.6 ms. Most microcontroller based A/Ds can acquire a reading in another 25 μs . Thus it is possible to turn on the ADXL202/ADXL210 and take a reading in <2 ms. If we assume that a 20 Hz sample rate is sufficient, the total current required to take 20 samples is $2 \text{ ms} \times 20 \text{ samples/s} \times 0.6 \text{ mA} = 24 \mu\text{A}$ average current. Running the part at 3 V will reduce the supply current from 0.6 mA to 0.4 mA, bringing the average current down to 16 μA .

The A/D should read the analog output of the ADXL202/ADXL210 at the X_{FILT} and Y_{FILT} pins. A buffer amplifier is recommended, and may be required in any case to amplify the analog output to give enough resolution with an 8-bit to 10-bit converter.

Power Cycling When Using the Digital Output

An alternative is to run the microcontroller at a higher clock rate and put it into shutdown between readings, allowing the use of the digital output. In this approach the ADXL202/ADXL210 should be set at its fastest sample rate ($T_2 = 0.5 \text{ ms}$), with a 500 Hz filter at X_{FILT} and Y_{FILT} . The concept is to acquire a reading as quickly as possible and then shut down the ADXL202/ADXL210 and the microcontroller until the next sample is needed.

In either of the above approaches, the ADXL202/ADXL210 can be turned on and off directly using a digital port pin on the microcontroller to power the accelerometer without additional components. The port should be used to switch the common pin of the accelerometer so the port pin is "pulling down."

CALIBRATING THE ADXL202/ADXL210

The initial value of the offset and scale factor for the ADXL202/ADXL210 will require calibration for applications such as tilt measurement. The ADXL202/ADXL210 architecture has been designed so that these calibrations take place in the software of the microcontroller used to decode the duty cycle signal. Calibration factors can be stored in EEPROM or determined at turn-on and saved in dynamic memory.

For low g applications, the force of gravity is the most stable, accurate and convenient acceleration reference available. A reading of the 0 g point can be determined by orientating the device parallel to the earth's surface and then reading the output.

A more accurate calibration method is to make measurements at +1 g and -1 g . The sensitivity can be determined by the two measurements.

To calibrate, the accelerometer's measurement axis is pointed directly at the earth. The 1 g reading is saved and the sensor is turned 180° to measure -1 g . Using the two readings, the sensitivity is:

$$\begin{aligned} \text{Let } A &= \text{Accelerometer output with axis oriented to } +1 g \\ \text{Let } B &= \text{Accelerometer output with axis oriented to } -1 g \text{ then:} \\ \text{Sensitivity} &= [A - B]/2 g \end{aligned}$$

For example, if the +1 g reading (A) is 55% duty cycle and the -1 g reading (B) is 32% duty cycle, then:

$$\text{Sensitivity} = [55\% - 32\%]/2 g = 11.5\%/g$$

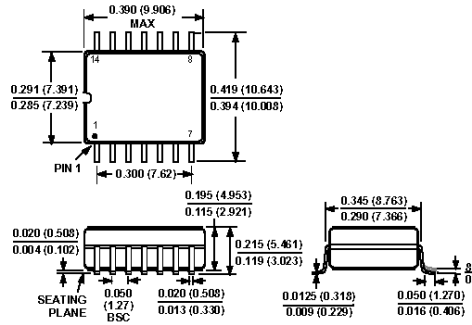
These equations apply whether the output is analog, or duty cycle.

Application notes outlining algorithms for calculating acceleration from duty cycle and automated calibration routines are available from the factory.

ADXL202/ADXL210

OUTLINE DIMENSIONS
Dimensions shown in inches and (mm).

**14-Lead CERPAK
(QC-14)**

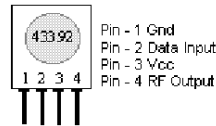


C30037 b-2-4/99

PRINTED IN U.S.A.

TWS-434 / RWS-434
<http://www.rentron.com>

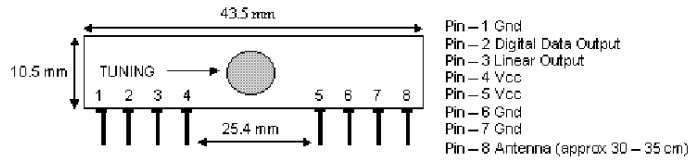
TWS-434A RF Transmitter



Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 2 - 12 VDC

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		2.0	-	12.0	V
Ip	Peak Current	2V / 12V	-	1.64 / 19.4	-	mA
Vh	Input High Voltage	Idata = 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
VI	Input Low Voltage	Idata = 0 uA (Low)	-	-	0.3	V
Fo	Operating Frequency		433.90	433.92	433.94	MHz
Tr / Tf	Modulation Rise / Fall Time	External Coding	-	-	100 / 100	uS
Po	RF Output Power – Into 50Ω	Vcc = 9 to 12 V Vcc = 5 to 6V	-	16 14	-	dBm
Dr	Data Rate	External Coding	-	2.4K	3K	Bps

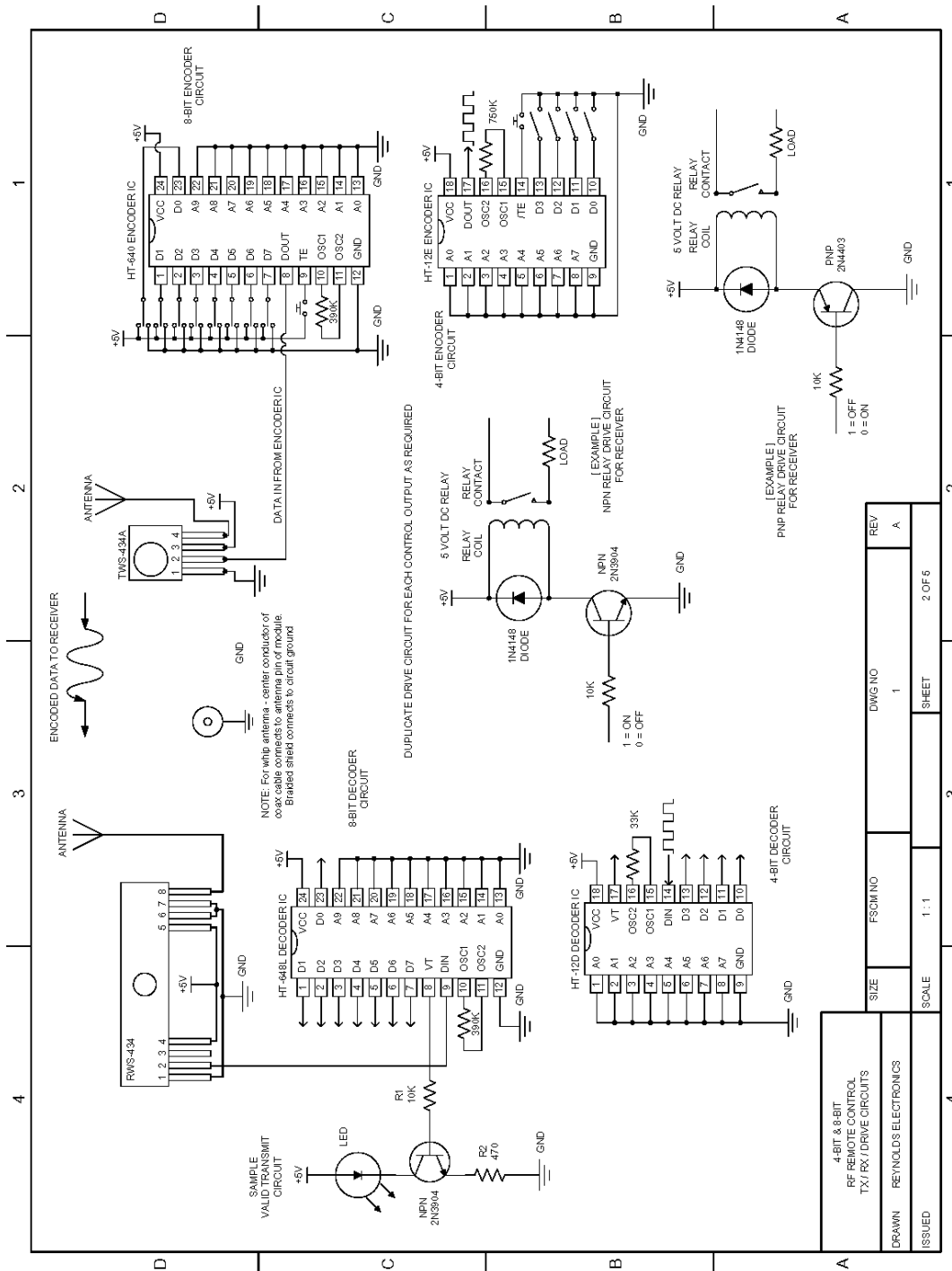
RWS-434 RF Receiver



Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 4.5 - 5.5 VDC
Output: Digital & Linear

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		4.5	5	5.5	V
It	Operating Current		-	3.5	4.5	mA
	Channel Width	+ / - 500				kHz
Rd	Data Rate				3k	Bps
Vdat	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Reynolds Electronics
 3101 Eastridge Lane
 Canon City, CO. 81212
 Tel: (719) 269-3469 Fax: (719) 276-2853
 Web: <http://www.rentron.com>



SPECIFICATION FOR LCD MODULE

Model No. TM162AAA6-1

Prepared by:	Date:
Checked by :	Date:
Verified by :	Date:
Approved by:	Date:

TIANMA MICROELECTRONICS CO., LTD

REVISION RECORD

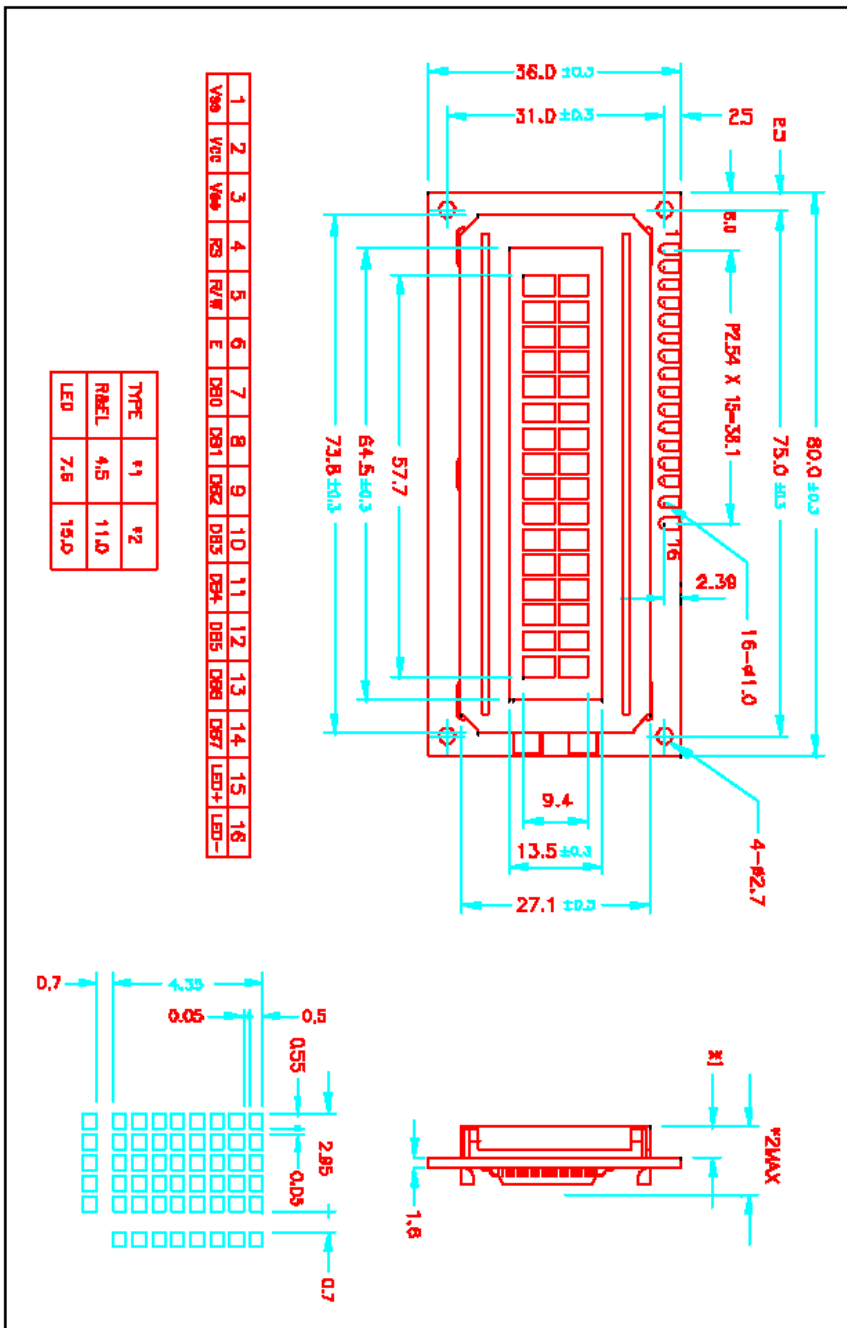
Date	Ref. Page	Revision No.	Revision Items	Check & Approval

1. General Specifications:

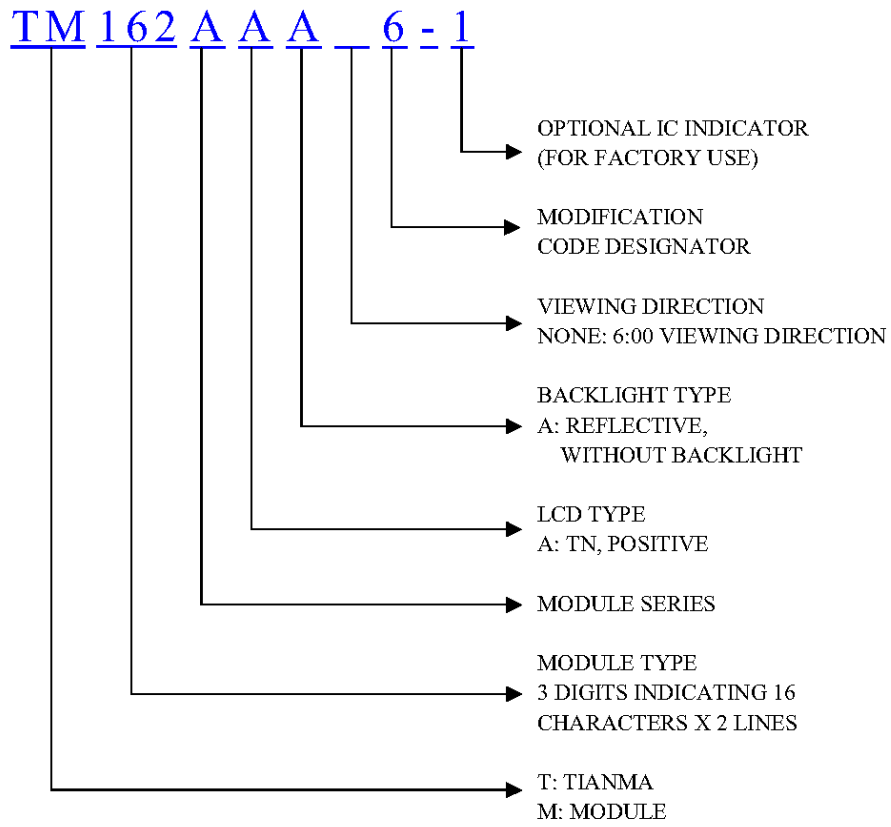
- 1.1 Display type: TN
- 1.2 Display color*:
 - Display color: Blue-Black
 - Background: White
- 1.3 Polarizer mode: Reflective/Positive
- 1.4 Viewing Angle: 6:00
- 1.5 Driving Method: 1/16 Duty 1/5 Bias
- 1.6 Backlight: None
- 1.7 Display Fonts: 5 x 7 dots + Cursor (1 Character)
- 1.8 Controller: HCD66701A00
- 1.9 Data Transfer: 8 Bit Parallel
- 1.10 Operating Temperature: 0----+50°C
 - Storage Temperature: -20----+60°C
- 1.11 Outline Dimensions: Refer to outline drawing on next page
- 1.12 Dot Matrix: 16 Characters X 2 Lines
- 1.13 Dot Size: 0.55X0.50(mm)
- 1.14 Dot Pitch: 0.60X0.55 (mm)
- 1.15 Weight: 20g
- 1.16 PCB edition: TM162AD P1-2

* Color tone is slightly changed by temperature and driving voltage.

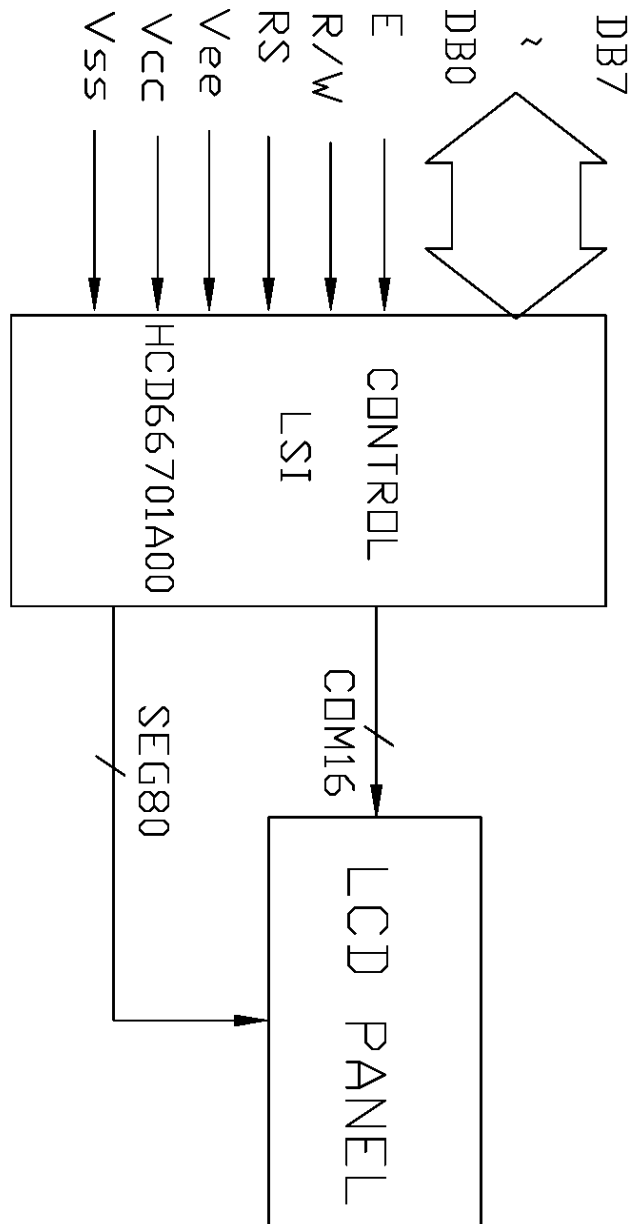
2. Outline Drawing



3. LCD Module Part Numbering System



4. Circuit Block Diagram



5. Absolute Maximum Ratings

Item	Symbol	Min.	Max.	Unit	Remark
Power Supply Voltage	$V_{DD}-V_{SS}$	-0.3	7.0	V	
LCD Driving Voltage	V_{LCD}	-0.3	13.0		
Operating Temperature Range	T_{OP}	0	+50	°C	No Condensation
Storage Temperature Range	T_{ST}	-20	+60		

6. Electrical Specifications and Instruction Code

6.1 Electrical characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	
Supply Voltage (Logic)	$V_{DD}-V_{SS}$	4.5	5.0	5.5	V	
Supply Voltage (LCD Drive)	V_{LCD}	-	4.7	-	V	
Input Signal Voltage	High	V_{IH} ($V_{DD}=5.0$)	$0.7V_{DD}$	-	$V_{DD}+0.3$	V
	Low	V_{IL} ($V_{DD}=5.0$)	-0.3	-	$0.2 V_{DD}$	V
Supply current (Logic)	I_{DD} ($V_{DD}-V_{SS}=5.0$)	-	1.5	2.0	mA	
Supply current (LCD Drive)	I_{EE}	-	0.40	0.6	mA	

6.2 Interface Signals

Pin No.	Symbol	Level	Description
1	Vss	0V	Ground
2	Vcc	5.0V	Power supply voltage for logic and LCD(+)
3	Vee	0.3V	Power supply voltage for LCD(-)
4	RS	H/L	Selects registers
5	R/W	H/L	Selects read or write
6	E	H/L	Starts data read/write
7	DB0	H/L	Data bit0
8	DB1	H/L	Data bit1
9	DB2	H/L	Data bit2
10	DB3	H/L	Data bit3
11	DB4	H/L	Data bit4
12	DB5	H/L	Data bit5
13	DB6	H/L	Data bit6
14	DB7	H/L	Data bit7
15	NC	-	No signal
16	NC	-	No signal

6.3 Interface Timing Chart

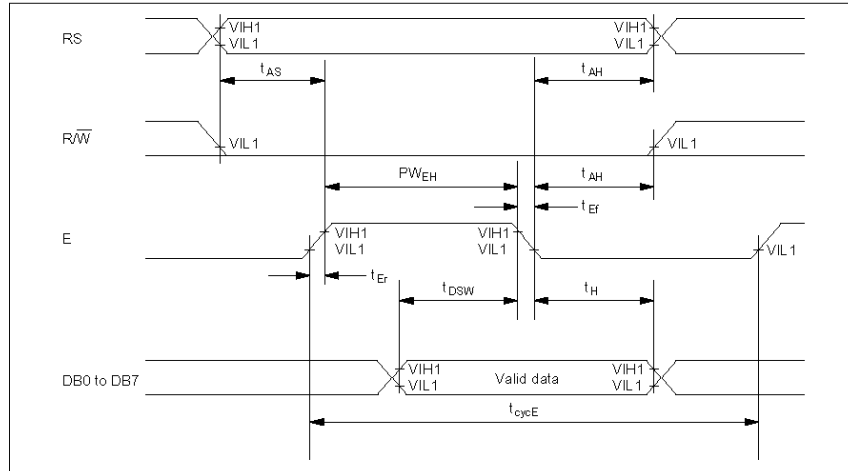
AC Characteristics ($V_{CC} = 2.7$ to $5.5V$, $T_a = -30$ to $+75^\circ C^{*3}$)**Write Operation**

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, $R\bar{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

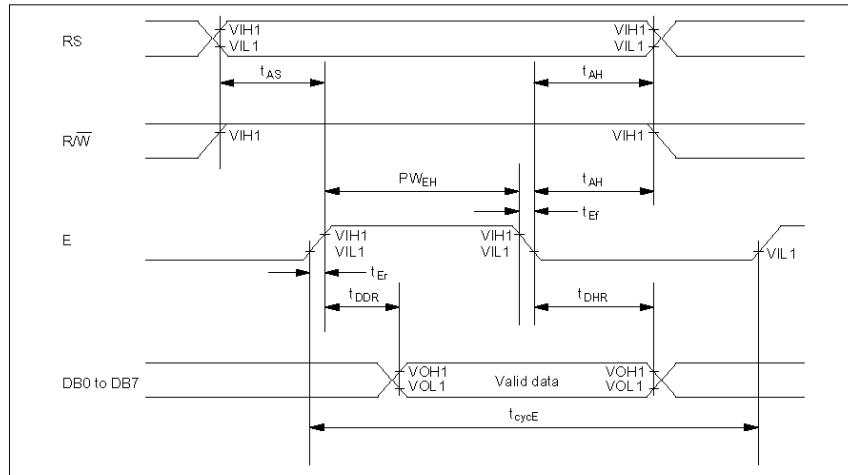
Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, $R\bar{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	350		
Data hold time	t_{DHR}	10	—	—		

Timing Characteristics



Write Operation



Read Operation

6.4 Instruction Code

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 320 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	1.28 ms
Return home	0	0	0	0	0	0	0	0	0	1	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.28 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	31 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	31 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	31 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (L), and character font (F).	31 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	31 μ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	31 μ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 320 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	31 μ s $t_{ADD} = 4.7 \mu$ s*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	31 μ s $t_{ADD} = 4.7 \mu$ s*
			VD = 1: Increment VD = 0: Decrement								DDRAM: Display data RAM	Execution time changes when frequency changes Example: When f_p or f_{osc} is 270 kHz, 31μ s $\times \frac{320}{270} = 37 \mu$ s
			S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move								CGRAM: Character generator RAM ACG: CGRAM address	
			R/L = 1: Shift to the right R/L = 0: Shift to the left								ADD: DDRAM address (corresponds to cursor address)	
			DL = 1: 8 bits, DL = 0: 4 bits								AC: Address counter used for both DD and CGRAM addresses	
			N = 1: 2 lines, N = 0: 1 line									
			F = 1: 5 \times 10 dots, F = 0: 5 \times 7 dots									
			BF = 1: Internally operating BF = 0: Instructions acceptable									

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

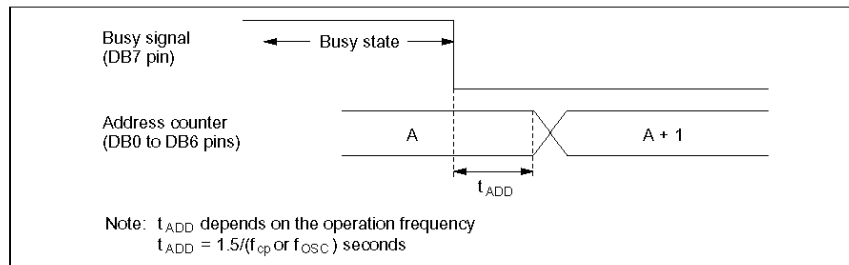


Figure 10 Address Counter Update

6.5 Character generator ROM(HCD66701A00)

Lower 4 Bits \ Upper 4 Bits	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	C.G RAM (1)		0	1	2	3	4	5	6	7	8	9	A
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	△	ä	g
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	ß	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ハ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	1	σ	Ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	ア	キ	ヌ	ラ	q	π
xxxx1000	(1)	(8	H	X	h	x	イ	ク	ネ	リ	∫	×
xxxx1001	(2))	9	I	Y	i	y	ウ	ツ	ル		∫	∫
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ン	ク	j	≠
xxxx1011	(4)	+	;	K	[k	[オ	サ	ヒ	ロ	*	≠
xxxx1100	(5)	,	<	L	¥	l	l	ャ	シ	フ	ワ	Φ	≠
xxxx1101	(6)	-	=	M]	m]	ユ	ズ	ハ	ン	≠	÷
xxxx1110	(7)	.	>	N	^	n	^	ヲ	セ	ホ	ッ	≠	
xxxx1111	(8)	/	?	O	_	o	_	キ	ツ	マ	マ	ö	■

Note: The user can specify any pattern for character-generator RAM.



PIC16F84A
Data Sheet

18-pin Enhanced FLASH/EEPROM
8-bit Microcontroller

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks


The Microchip name and logo, the Microchip logo, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, KEELoQ, SEEVAL, MPLAB and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, ICSP, In-Circuit Serial Programming, Filter-Lab, MXDEV, microID, FlexROM, fuzzyLAB, MPASM, MPLINK, MPLIB, PICC, PICDEM, PICDEM.net, ICEPIC, Migratable Memory, FanSense, ECONOMONITOR, Select Mode and microPort are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2001, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



PIC16F84A

18-pin Enhanced FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
 - External RB0/INT pin
 - TMR0 timer overflow
 - PORTB<7:4> interrupt-on-change
 - Data EEPROM write complete

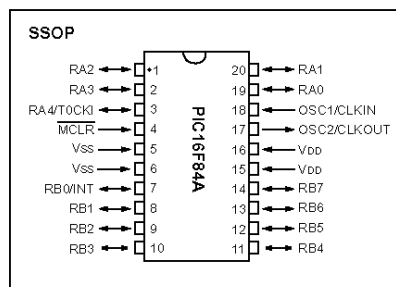
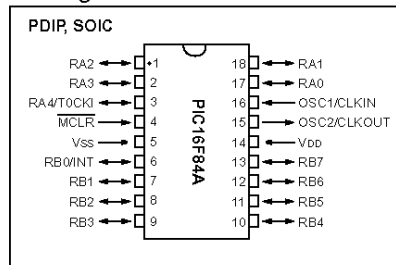
Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
 - 25 mA sink max. per pin
 - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

Pin Diagrams



CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
 - Commercial: 2.0V to 5.5V
 - Industrial: 2.0V to 5.5V
- Low power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 15 µA typical @ 2V, 32 kHz
 - < 0.5 µA typical standby current @ 2V

PIC16F84A

Table of Contents

1.0 Device Overview	3
2.0 Memory Organization	5
3.0 Data EEPROM Memory	13
4.0 I/O Ports	15
5.0 Timer0 Module	19
6.0 Special Features of the CPU	21
7.0 Instruction Set Summary	35
8.0 Development Support	43
9.0 Electrical Characteristics	49
10.0 DC/AC Characteristic Graphs	61
11.0 Packaging Information	71
Appendix A: Revision History	75
Appendix B: Conversion Considerations	78
Appendix C: Migration from Baseline to Mid-Range Devices	78
Index	79
On-Line Support	83
Reader Response	84
PIC16F84A Product Identification System	85

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@mail.microchip.com or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on all of our products.

PIC16F84A

1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F84A device. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F84A belongs to the mid-range family of the PICmicro® microcontroller devices. A block diagram of the device is shown in Figure 1-1.

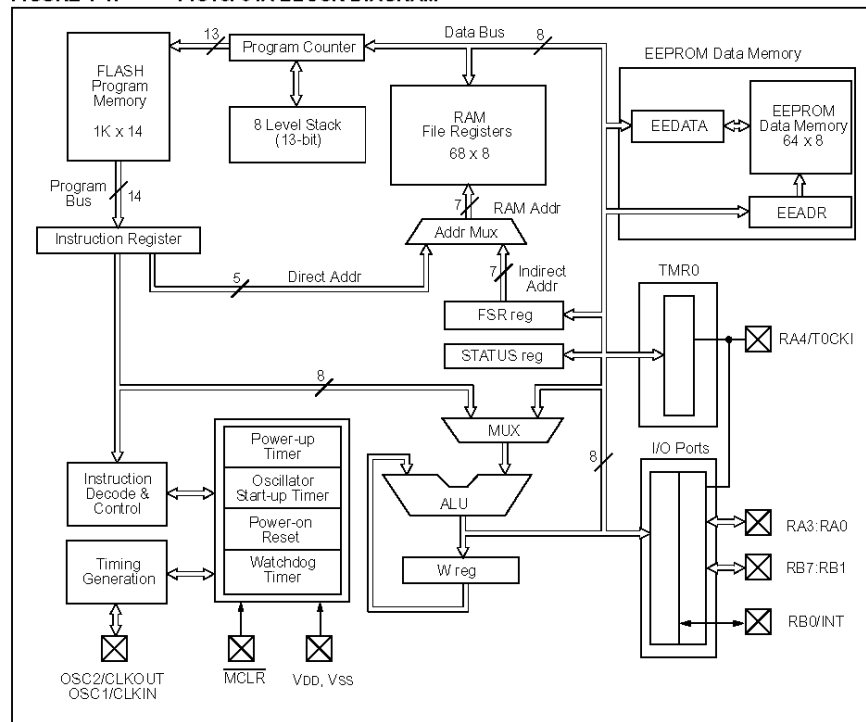
The program memory contains 1K words, which translates to 1024 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 68 bytes. Data EEPROM is 64 bytes.

There are also 13 I/O pins that are user-configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input

Table 1-1 details the pinout of the device with descriptions and details for each pin.

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



PIC16F84A

TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	17	19	I/O	TTL	PORTA is a bi-directional I/O port. Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	
RB0/INT	6	6	7	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0/INT can also be selected as an external interrupt pin. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin. Serial programming clock. Interrupt-on-change pin. Serial programming data.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	
RB5	11	11	12	I/O	TTL	
RB6	12	12	13	I/O	TTL/ST ⁽²⁾	
RB7	13	13	14	I/O	TTL/ST ⁽²⁾	
V _{SS}	5	5	5,6	P	—	Ground reference for logic and I/O pins.
V _{DD}	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = Output I/O = Input/Output P = Power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F84A

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F84A. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh. More details on the EEPROM memory can be found in Section 3.0.

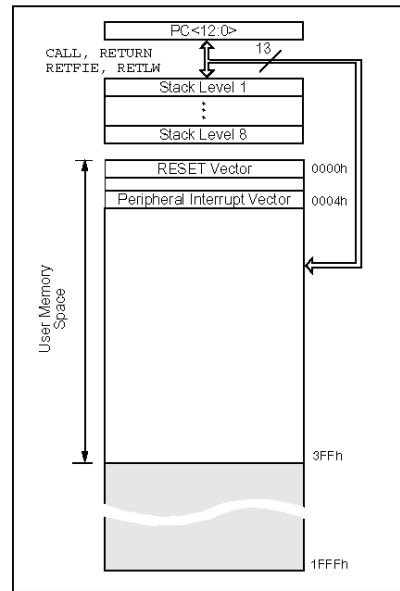
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84A, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 2-1). Accessing a location above the physically implemented address will cause a wraparound. For example, for locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h, the instruction will be the same.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK - PIC16F84A



PIC16F84A

2.2 Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device.

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 2-2 shows the data memory map organization.

Instructions `MOVWF` and `MOVF` can move values from the *W* register to any location in the register file (*F*), and vice-versa.

The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) (Section 2.5). Indirect addressing uses the present value of the *RP0* bit for access into the banked areas of data memory.

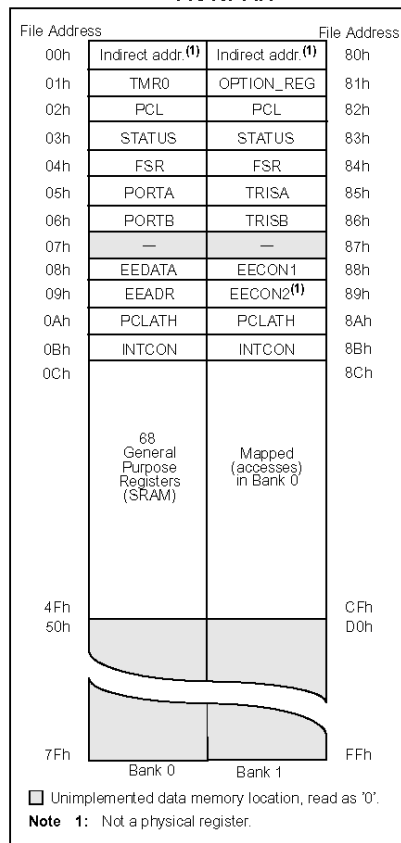
Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the *RP0* bit (*STATUS<5>*). Setting the *RP0* bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers. The remainder are General Purpose Registers, implemented as static RAM.

2.2.1 GENERAL PURPOSE REGISTER FILE

Each General Purpose Register (GPR) is 8-bits wide and is accessed either directly or indirectly through the FSR (Section 2.5).

The GPR addresses in Bank 1 are mapped to addresses in Bank 0. As an example, addressing location 0Ch or 8Ch will access the same GPR.

FIGURE 2-2: REGISTER FILE MAP - PIC16F84A



M68HC11 E SERIES HCMOS MICROCONTROLLER UNIT

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. MOTOROLA and the Motorola logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© MOTOROLA, INC., 1993, 1996

SECTION 1 INTRODUCTION

This document contains a detailed description of the M68HC11 E series of 8-bit microcontroller units (MCUs). These MCUs all combine the M68HC11 CPU with high-performance, on-chip peripherals. The E series is comprised of many devices with various configurations of RAM, ROM or EPROM, and EEPROM. Several low-voltage devices are also available. With the exception of a few minor differences, the operation of all E-series MCUs is identical. A fully static design and high-density complementary metal-oxide semiconductor (HCMOS) fabrication process allow E-series devices to operate at frequencies from 3 MHz to dc, with very low power consumption.

1.1 Features

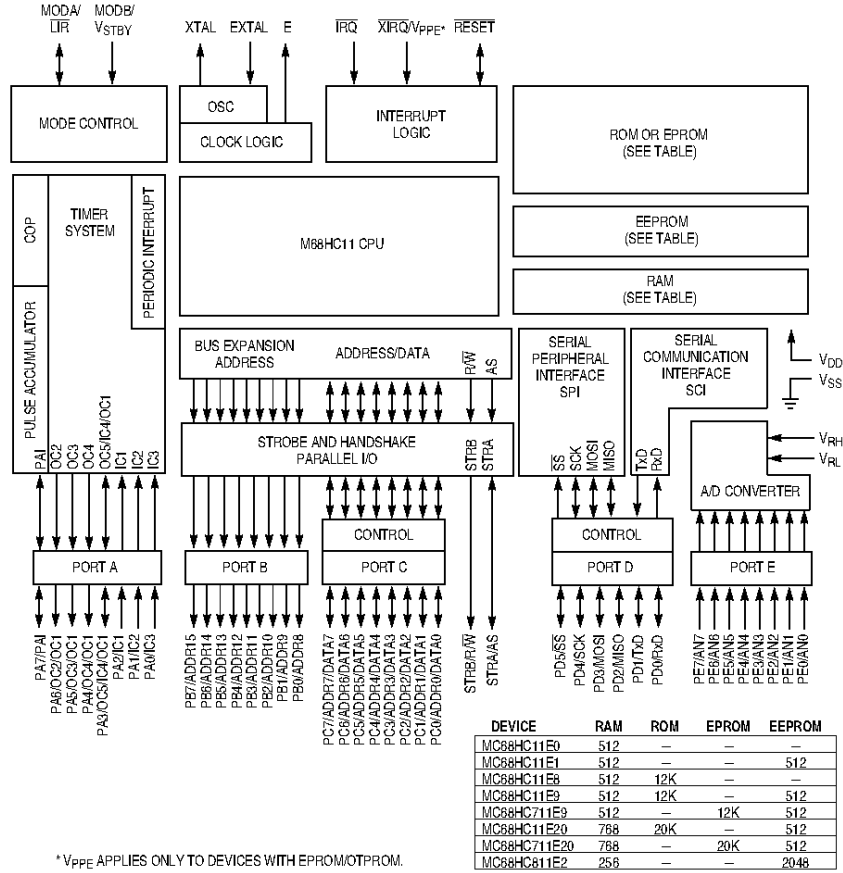
- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- Low-Voltage Devices Available (3.0 – 5.5 Vdc or 2.7 – 5.5 Vdc)
- 0, 256, 512, or 768 Bytes of On-Chip RAM, Data Retained During Standby
- 0, 12, or 20 Kbytes of On-Chip ROM or EPROM
- 0, 512, or 2048 Bytes of On-Chip EEPROM with Block Protect for Security
 - 2048 bytes of EEPROM with selectable base address in MC68HC811E2
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
 - Additional Baud Rates Available on MC68HC(7)11E20
- Synchronous Serial Peripheral Interface (SPI)
- 8-Channel 8-Bit Analog-to-Digital (A/D) Converter
- 16-Bit Timer System
 - Three Input Capture (IC) Channels
 - Four Output Compare (OC) Channels
 - One Additional Channel, Selectable as Fourth IC or Fifth OC
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System
- 38 General-Purpose Input/Output (I/O) Pins
 - 16 Bidirectional I/O Pins
 - 11 Input-Only Pins
 - 11 Output-Only Pins
- Several Packaging Options
 - 52-Pin Plastic Leaded Chip Carrier (PLCC)
 - 52-Pin Windowed Ceramic Leaded Chip Carrier (CLCC)
 - 52-Pin Plastic Thin Quad Flat Pack, 10 mm X 10 mm (TQFP)
 - 64-Pin Plastic Quad Flat Pack (QFP)
 - 48-Pin Plastic Dual In-Line Package (DIP), MC68HC811E2 only
 - 56-Pin Plastic Dual In-Line Package, .070" Lead Spacing (SDIP)

1

1.2 Structure

Below is the functional diagram of the E-series MCUs. Differences among devices are noted in the table.

1

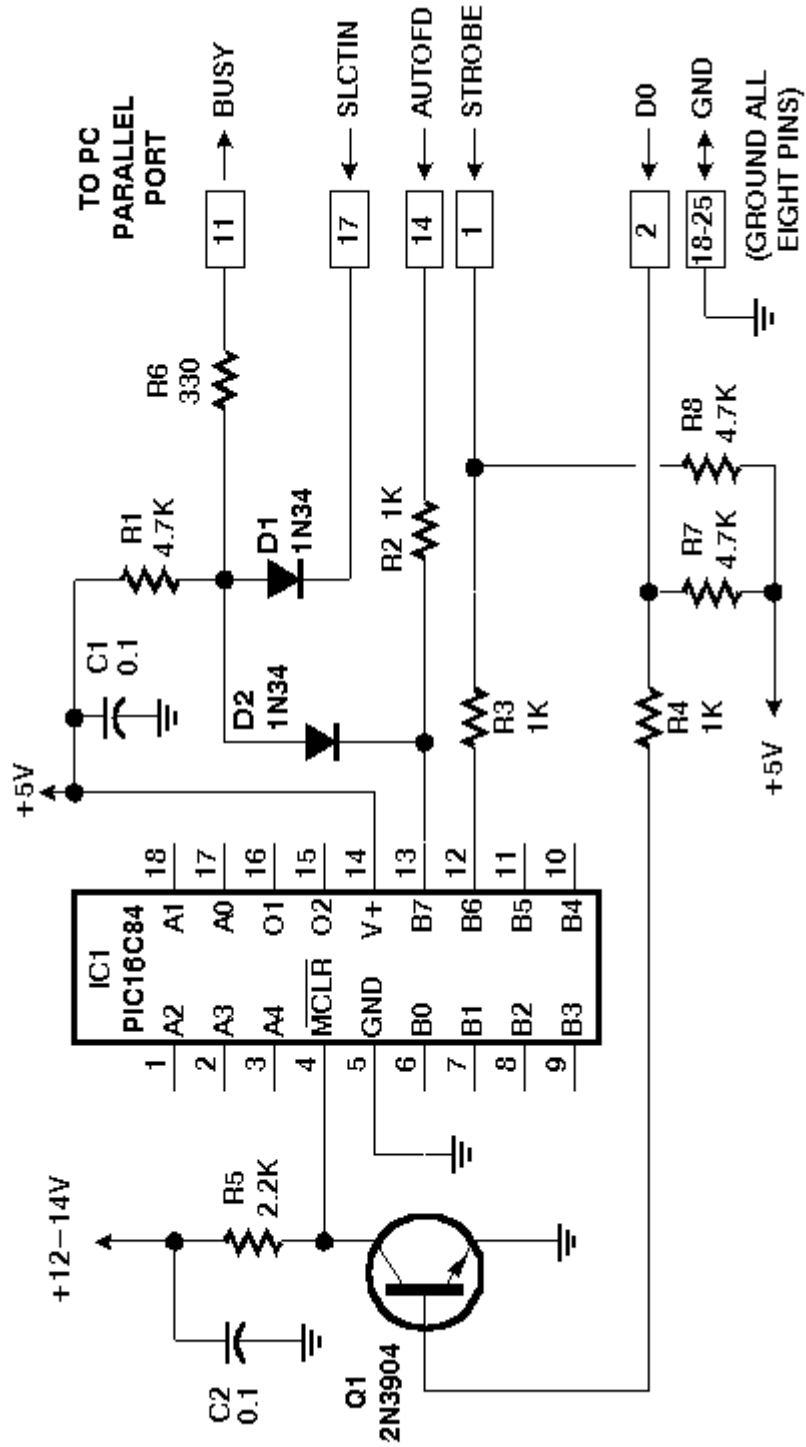


E-SERIES BLOCK

Figure 1-1 M68HC11 E-Series Block Diagram

**REVISED NOPPP – MICHAEL A. COVINGTON, OCT. 31, 1998
FOR GREATER COMPATIBILITY WITH SOME PC PARALLEL PORTS.**

START SOFTWARE FIRST; APPLY POWER AND INSERT PIC WHEN TOLD TO DO SO.



9. APENDICES

9.1 GUÍA DE USUARIO

DISPOSITIVO TRANSMISOR

1. Colocar el dispositivo transmisor en el kayak enfrente del remero delante de la bañera.
2. Encender el dispositivo, la transmisión y el despliegue de datos comienza inmediatamente (figura A1).

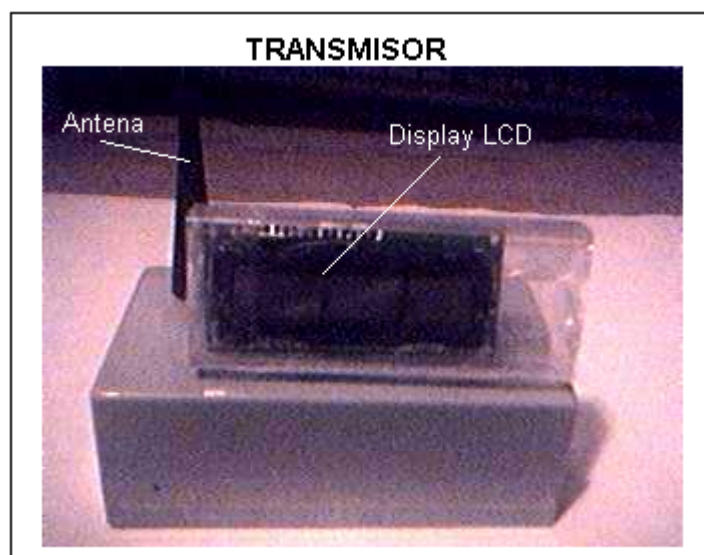


Figura A1 Dispositivo transmisor

DISPOSITIVO RECEPTOR

1. Encender el dispositivo, la recepción de datos y despliegue de datos comienza inmediatamente.
 - **Captura de memorias**
 1. Presionar el botón "C" para capturar la información de BPM y Fuerza recibida, cada vez que se presione dicho botón aparecerá la cuenta del número de memorias capturadas en la pantalla de LCD, hasta 5 memorias(figura A2).
 - **Visualización de memorias**
 1. Presionar el botón "V", para visualizar las memorias previamente capturadas, la información en memoria aparecerá a la derecha de la información recibida así como el número correspondiente de memoria (figura A3).
 - **Borrado de memoria**
 1. Si se desean borrar las memorias, presionar el botón "V" repetidamente hasta visualizar en la pantalla de LCD el símbolo "&", posteriormente presionar el botón "C". Todas las memorias serán borradas y aparecerá en el indicador de memorias capturadas el número "0"

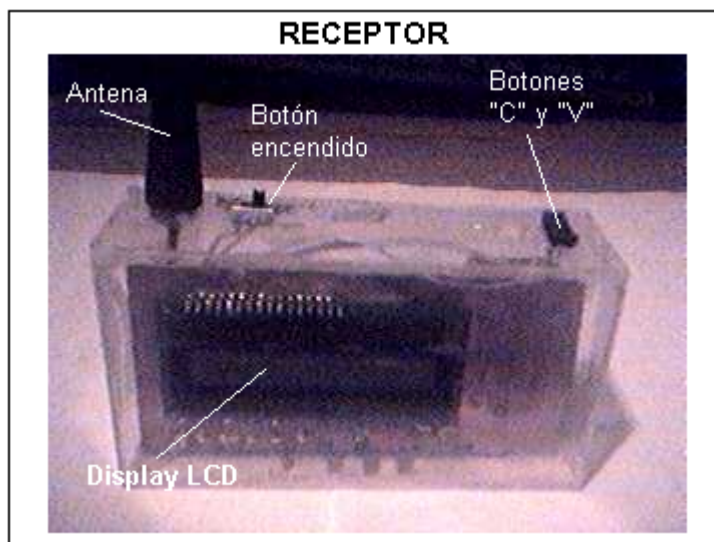


Figura A2 Dispositivo receptor

Indicadores en pantalla LCD del Receptor

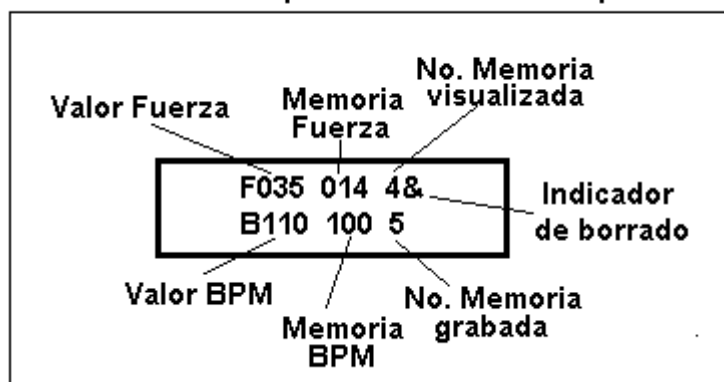
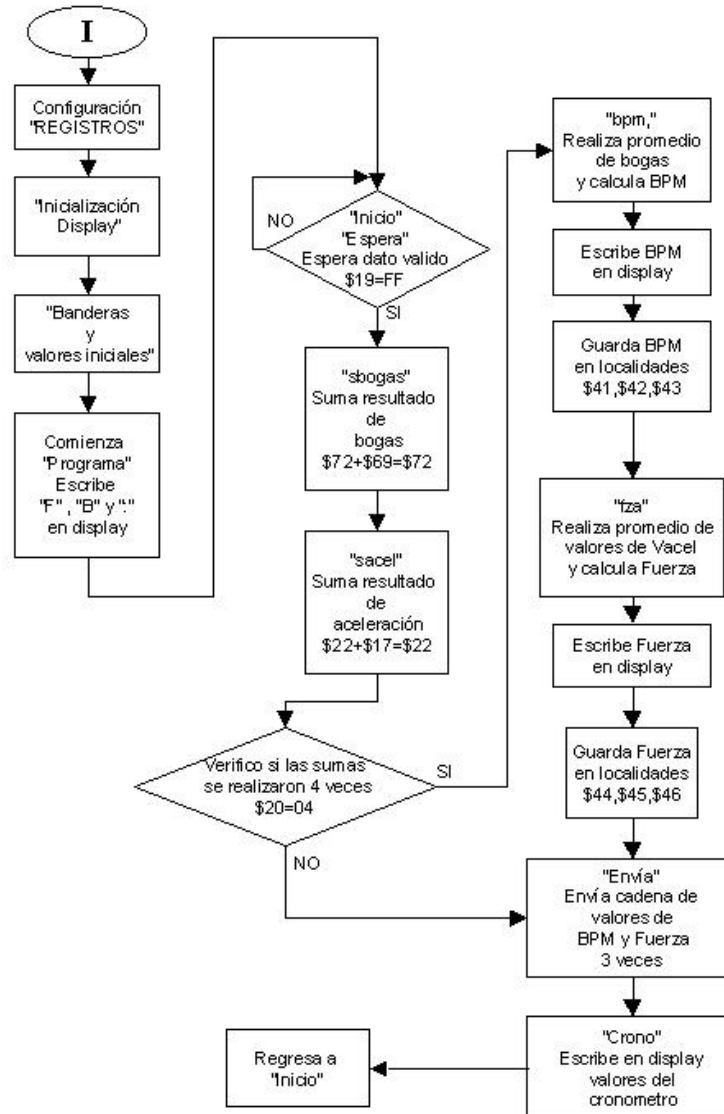


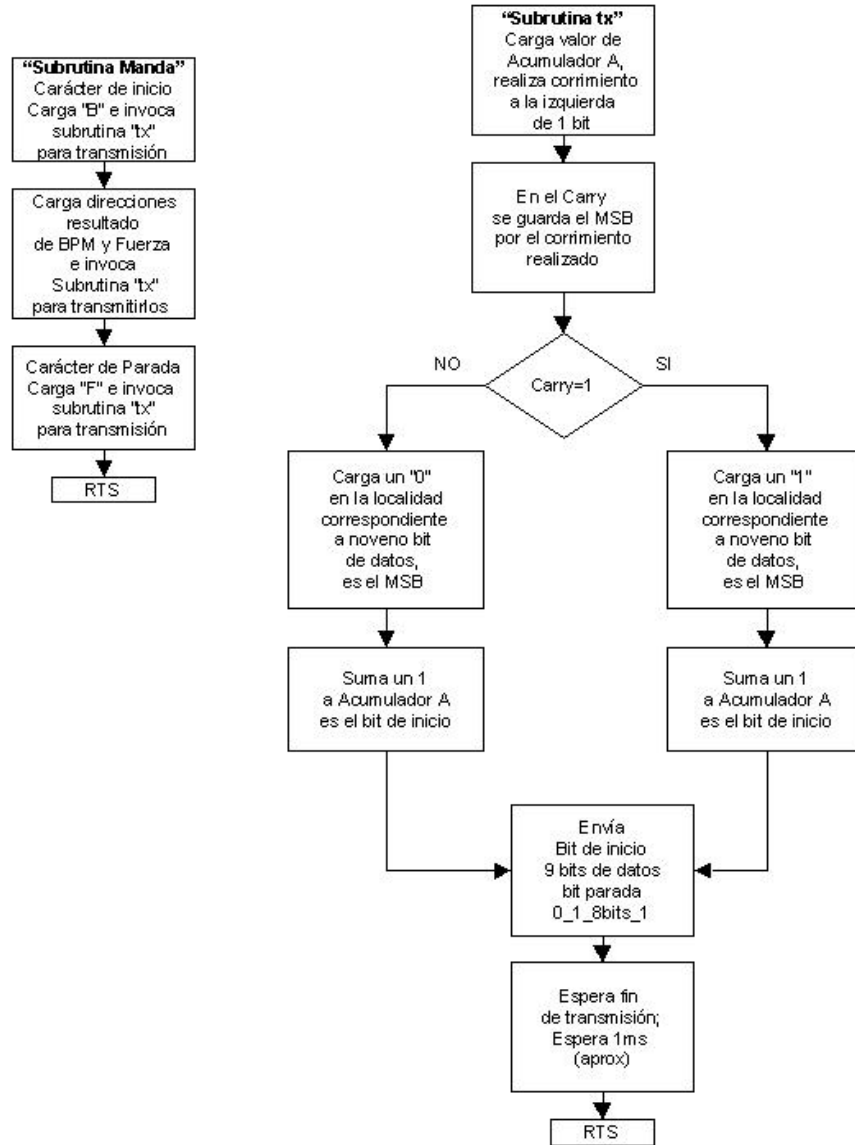
Figura A3 Indicadores de display

9.2 DIAGRAMAS DE FLUJO DEL TRANSMISOR

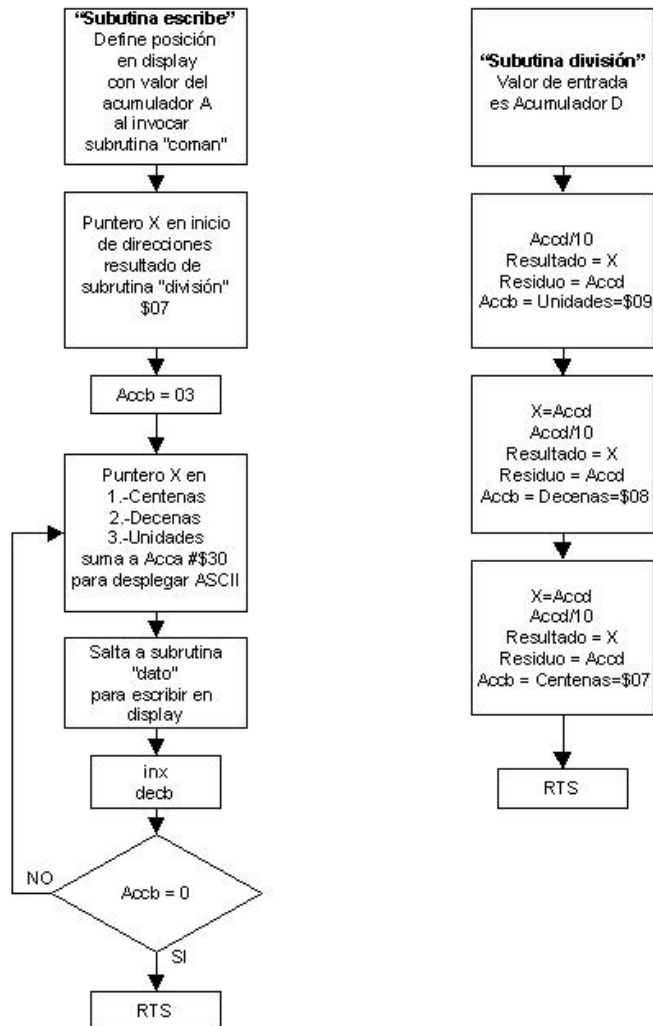
"Programa Transmisor"



Subrutinas "Manda" y "Tx"



Subrutinas "escribe" y "división"



Subrutinas del display "dato", "coman" y "delay"

"Subrutina dato"
 (Escribe datos en display)
 Acca es el valor de entrada
 R/W=0

Carga valor de entrada en bus de datos de display

RS=1 E=1

RS=1 E=0

RS=0 E=0

Espera aprox. 1ms.
 Subrutina "delay"

RTS

"Subrutina coman"
 (Envía comandos al display)
 Acca es el valor de entrada
 R/W=0

Carga valor de entrada en bus de datos de display

RS=0 E=0

RS=0 E=1

RS=0 E=0

Espera aprox. 1ms.
 Subrutina "delay"

RTS

"Subrutina delay"
 Carga acumulador A con #\$ff

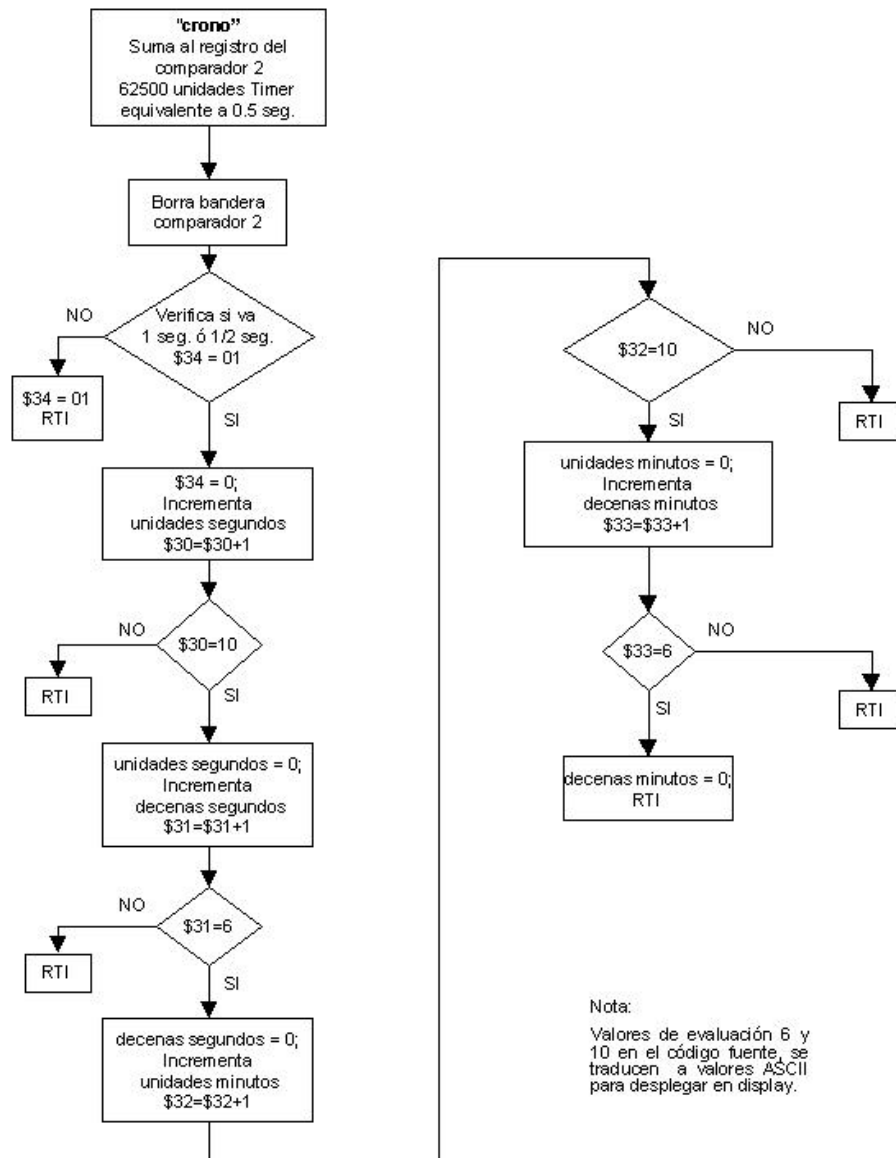
deca

NO
 Acca = 0

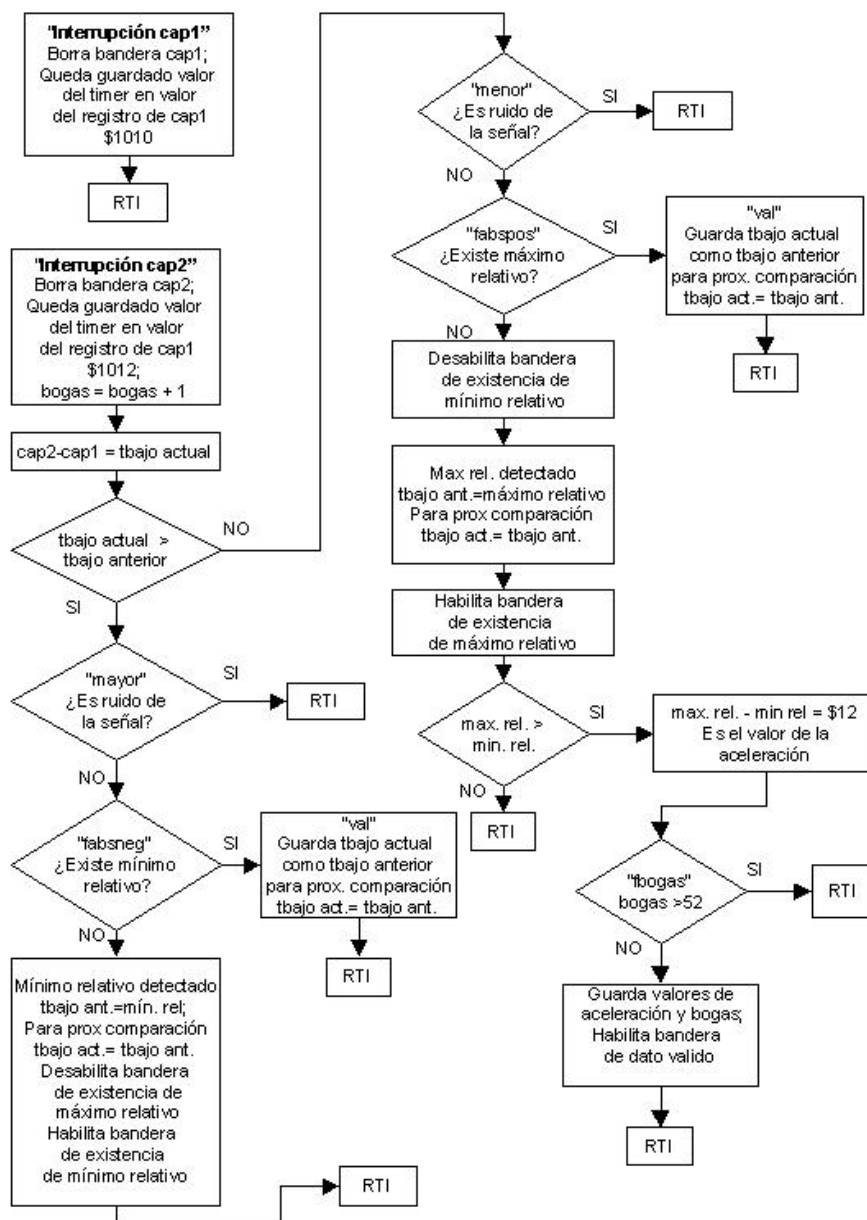
SI

RTS

Interrupción del Comparador 2 “crono”

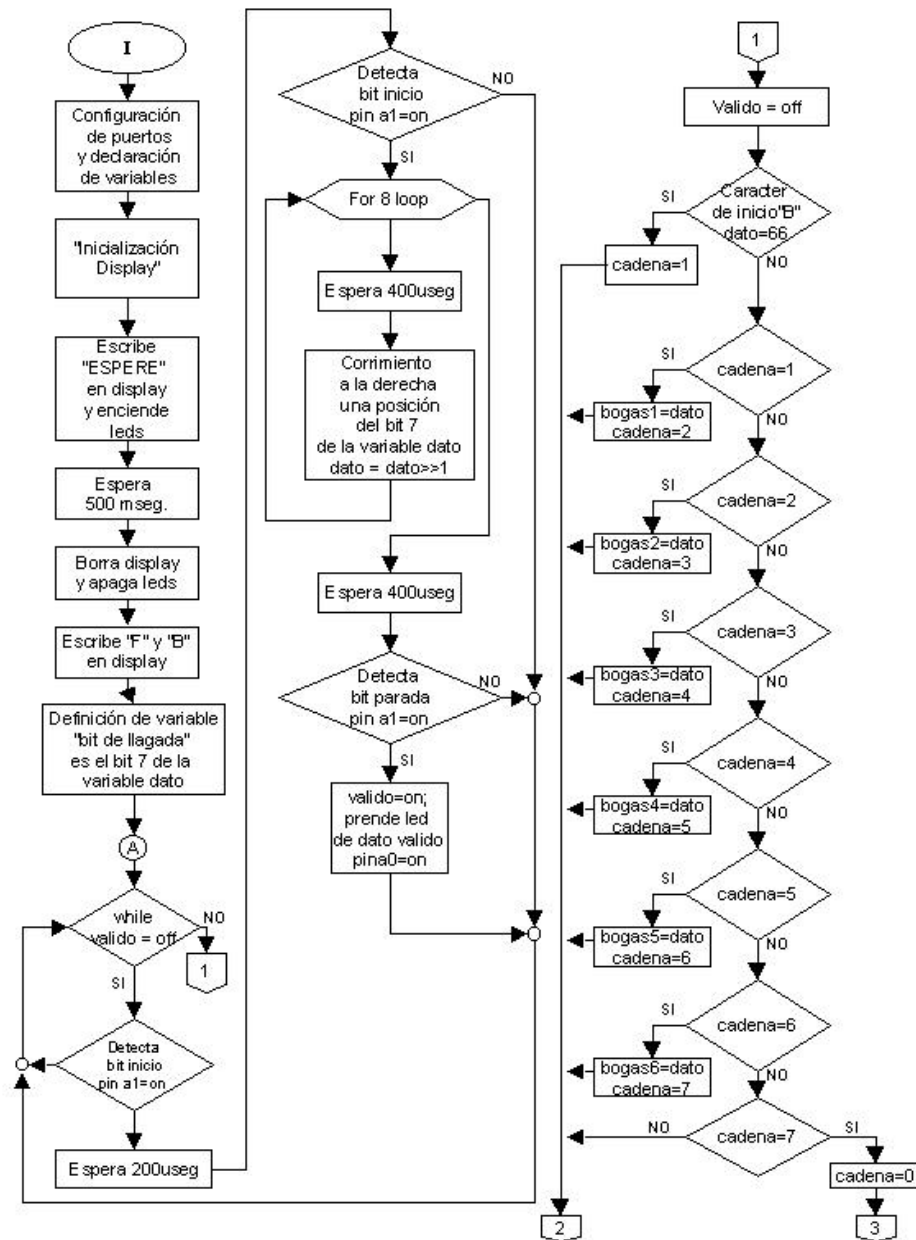


Interrupciones de Capturadores 1 y 2 "cap1" y "cap2"

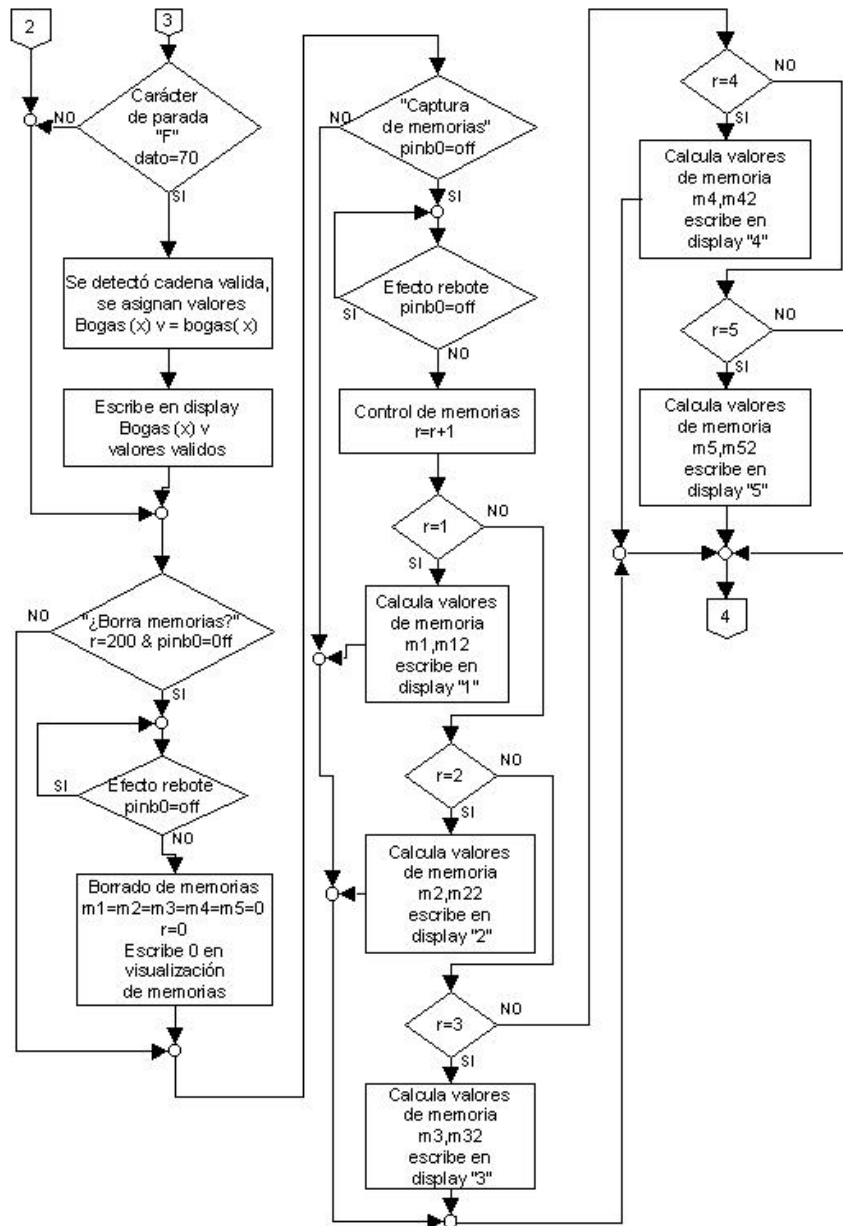


9.3 DIAGRAMAS DE FLUJO DEL RECEPTOR

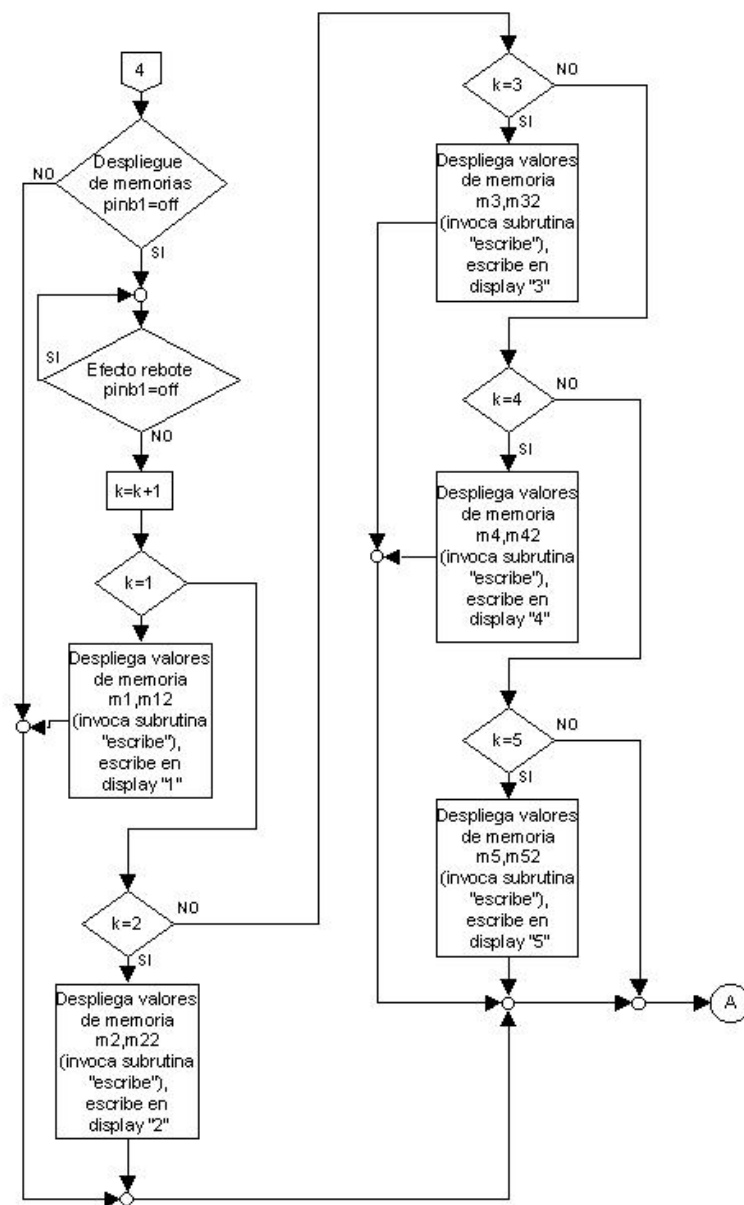
"Receptor"



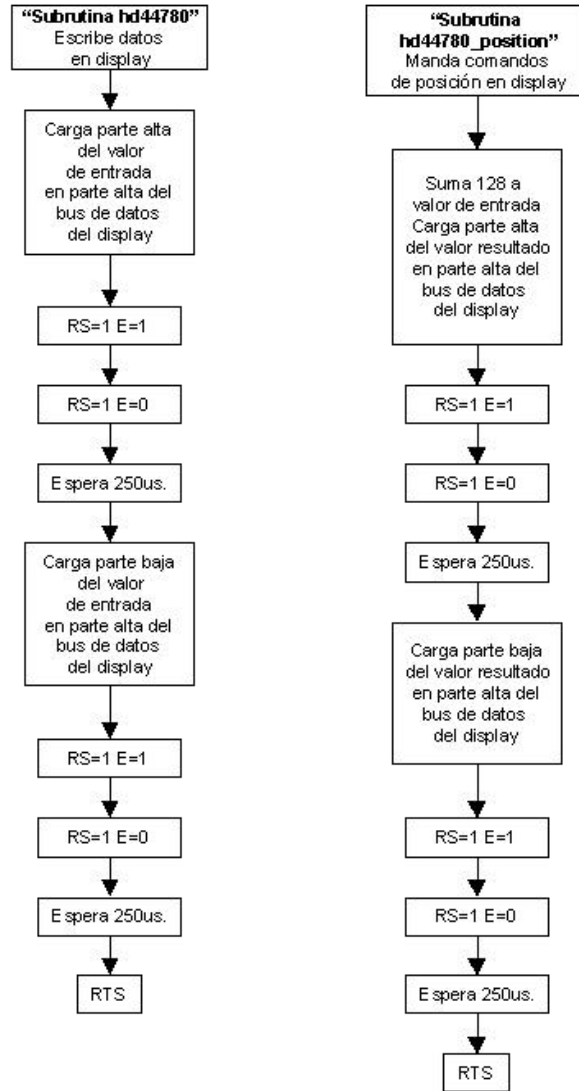
“Receptor (cont.)”



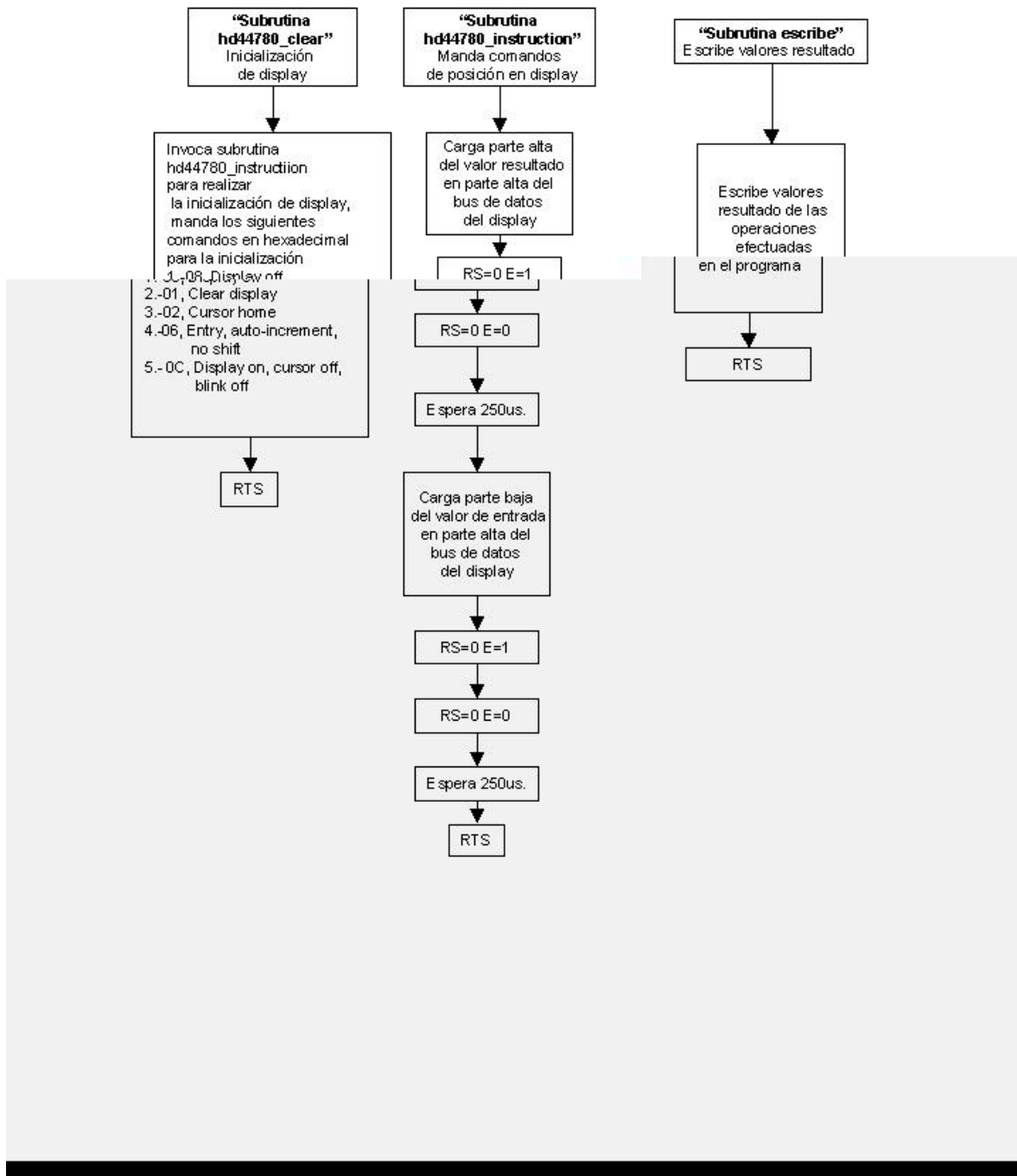
"Receptor (cont.1)"



Subrutinas "hd44780" y "hd44780_position"



Subrutinas "hd44780_clear", "hd44780_instruction" y "escribe"



9.4 CÓDIGO FUENTE DEL TRANSMISOR

```

**archivo final de la tesis
**Registros cap1=$1010-$1011
**   cap2=$1012-$1013

    org $f800
main   equ *
    lds #$ff ;stack al final de la RAM

*****REGISTROS*****
    ldx #$1000

    ldaa #$32
    staa $102b ;baud 2400 (registro indefinido al reset)

    bset $2d,x,$08 ;tx habilitado
    bset $24,x,$03 ;incremento de cuenta del tiempo cada 8useg
    bset $22,x,$46 ;habilito capturadores 1 y 2, comparador 2
    bset $21,x,$24 ;flancos de bajada cap1 y de subida cap2
    bset $20,x,$00 ;no hay salida hardware del comparador
    bset $07,x,$ff ;port C como salidas
    bset $3c,x,$0c ;maxima prioridad de interrupcion la de comparador 2
    bset $09,x,$02 ;port d tx salida
    bset $2c,x,$10 ;0_9bits_1
    cli ;habilito interrupciones

*****FIN REGISTROS*****

*****INICIALIZACION DISPLAY*****
    ldx #$ffff ;Retardo para que encienda el display
l     dex      ;en modo bootstrap
    bne l
    ldaa #$38 ;function set,8 bits, 2 lineas,5x10 puntos
    jsr coman
    ldaa #$0c ;display on,cursor off
    jsr coman
    ldaa #$01 ;borra display
    jsr coman
    ldaa #$06 ;entry mode set,desplaza a la derecha
    jsr coman

*****FIN INICIALIZACION DISPLAY*****

*****banderas y valoresiniciales*****

    ldx #$f424
    stx $1018 ;62500 ciclos de 8us = 500ms
            ;la interrupcion se producira cada 0.5 segs
            ;para el reloj

```

```

Idx #0000 ;
stx $50 ;valor inicial"tiempo anterior"
stx $60 ; absneg=00
stx $67 ;valor de la cuenta de bogas
Idx #1aa
stx $62 ;abspos
Idaa #17
staa $65 ;fabsneg bandera absoluto negativo
staa $66 ;fabspos bandera absoluto positivo
Idx #1f80 ;ciclos por minuto frec. 134.4
stx $69 ;134.4 x 60 = 8064 = 1f80
Idx #50 ;80 decimal peso
stx $90
Idaa #00
staa $17
staa $22 ;valor inicial aceleracion
staa $71 ;valor inicial bogas
Idaa #04
staa $20 ;control promedios

```

```

Idaa #30 ;cero ascci
staa $30 ;dir de unidades segundos 0-9
staa $31 ;dir decenas segundos 0-6
staa $32 ;dir unidades minutos 0-9
staa $33 ;dir decenas minutos 0-6
staa $34 ;dir de dos veces int. comp2 1=1/2 seg, 2pases=1seg aprox.
Idaa #ff ;bandera dato valido
staa $19

```

*****PROGRAMA*****

```

Idaa #85
jsr coman
Idaa #46 ;escribe F
jsr dato
Idaa #c5
jsr coman
Idaa #42 ;B
jsr dato

```

```

Idaa #cd
jsr coman
Idaa #3a ; ":" dos puntos para el crono
jsr dato

```


inicio Idx # \$00

**espera brclr \$19,x,\$ff,envia ;espero a que haya un dato valido
 bclr \$19,x,\$ff ;pongo a cero la bandera de dato valido**

**sbogas Idd \$71
 add \$69
 std \$71 ;suma resultado bogas**

**sacel Idd \$22
 add \$17
 std \$22 ;suma resultado aceleracion**

**Idab \$20 ;checo cuantas veces va \$20=04
decb
stab \$20
bne envia ;salta si no es cero a envia**

**Idaa # \$04
staa \$20 ;reinicio control promedios**

**bpm Idd \$71 ;realizo promedio bogas adquiridas
 Idx # \$04
 idiv
 stx \$71 ;resultado en X residuo en D
 Idd # \$1f80 ; 134*60=1f68=8040 134.4*60=1f80=8064
 Idx \$71 ;valor promedio bogas
 idiv ;divido valor max bpm/valor promedio
 ;resultado en X residuo en d
 stx \$71 ;guardo resultado bpm**

**Idd \$71 ;division trabaja con acumulador D
jsr division ;saco valores decimales
Idaa # \$c7 ; escribe bpm
jsr escribe**

**Idaa \$07 ;guardo valores de bogas por minuto
staa \$41 ;centenas**

**Idaa \$08 ;decenas
staa \$42**

**Idaa \$09 ;unidades
staa \$43**

**fza Idd \$22 ;carga con valor de aceleracion
 Idx # \$04
 idiv**

```

    stx $80
    stx $82
    ldd $80
    ldx $90 ;80 decimal=masa
f    addd $82
    std $80 ;aceleracion + aceleracion ,$190 veces
    dex
    bne f
    std $80 ;guardo valor de la multiplicaciøn
    ldd $80
    ldx #$74 ;116 decimal valor prom de 1G
    idiv
    stx $80 ;valor de fuerza

vfza    ldd $80 ;fza
        jsr division
        ldaa #$87
        jsr escribe ; *****escribe fza y incluido tx de fza*****

        ldaa $07 ;guardo valores de fuerza
        staa $44 ;centenas

        ldaa $08 ;decenas
        staa $45

        ldaa $09 ;unidades
        staa $46

        ldx #$00
        stx $71
        stx $22

envia    jsr manda ; envio datos 3 veces
        jsr manda
        jsr manda

crono    ldaa #$cb ; escribo valores del reloj
        jsr coman
        ldaa $33
        jsr dato

        ldaa $32
        jsr dato

        ldaa #$ce
        jsr coman

        ldaa $31
        jsr dato

        ldaa $30

```

jsr dato

jmp inicio

*******FIN PROGRAMA*******

*******SUBROUTINAS*******

*******subrutina manda tx datos*******

manda psha

pshx

pshb

Idaa #\$42 ;"B" manda 3 veces

jsr tx

jsr tx

jsr tx

Idx #\$41 ;carga datos de las direcciones resultado

ldab #\$06 ;6 datos a transmitir

yatx Idaa \$0,x ;puntero de resultados

adda #\$30 ;se le suman 30 para que despliegue ASCII

jsr tx

inx

decb

bne yatx

Idaa #\$46 ;"F" fin de cadena

jsr tx

pulb

pulx

pula

rts

*******fin manda tx *******

*******transmision*******

tx psha
pshx
pshb
Idx #\$1000

```

Isla ;corrimiento a la izquierda 1 lugar pone un cero en lsb
bcs uno; si el carri es uno
bra cero

```

```

uno      bset $2c,x,$40
         inca ;sumo 1 a "A" bit de inicio
         bra serie
cero     bclr $2c,x,$40 ;tengo el mayor msb en el bit9
         inca
serie    staa $102f
wait     brclr $2e,x,$c0,wait ;espero a que termine de transmitir
         jsr delay
         pulb
         pulx
         pula
         rts

```

*****fin transmision*****

*****division*****

```

division psha
         pshx
         pshb
         ldx #$a ;divido entre 10 decimalresiduo en D
         idiv
         stx $02 ;resultado en x residuo en D
         stab $09 ;UNIDADES

```

```

         ldd $02
         ldx #$a
         idiv
         stx $02 ;resultado en x residuo en d
         stab $08 ;DECENAS

```

```

         ldd $02
         ldx #$a
         idiv
         stx $02
         stab $07 ;CENTENAS
         pulb
         pulx
         pula
         rts

```

*****fin division*****

*****escribe 3 lugares*****

```

escribe psha ;posicion del cursor del display
         pshx
         pshb

```

```

    jsr coman
    ldx #$07 ;carga datos de las direcciones resultado
    ldab #$03 ;y despliega en el display
ya   ldaa $0,x ;puntero de resultados
    adda #$30 ;se le suman 30 para que despliegue ASCII
    jsr dato
    inx
    decb
    bne ya
    pulb
    pulx
    pula
    rts
*****fin escribe*****

```

```

*****comparador 2 (reloj)*****
* las direcciones del reloj son:
* $30 ;dir de unidades segundos 0-9
* $31 ;dir decenas segundos 0-6
* $32 ;dir unidades minutos 0-9
* $33 ;dir decenas minutos 0-6
* $34 ;dir de dos veces int. comp2 1=1/2 seg, 2=1seg aprox.

```

```

comp2  psha
       pshx

       ldd $100e
       addd #$f424 ; le sumo 62500 ciclos de reloj
       std $1018

       ldaa #$ff
       staa $1023 ; borro bandera del comparador 2

       ldaa $34 ; cargo con bandera de 1/2 seg
       cmpa #$01
       beq reloj
       ldaa #$01
       staa $34
       bra sreloj

reloj  ldaa #$00
       staa $34 ; pongo bandera de 1/2 seg a cero

uniseg ldaa $30
       inca

```

```
staa $30
cmpa #$3a ; si= 9 decimal incrementa decenas
beq decseg
bra sreloj
```

```
decseg Idaa #$30
staa $30 ;ceros uniseg
Idaa $31
inca
staa $31
cmpa #$36
beq unimin
bra sreloj
```

```
unimin Idaa #$30
staa $31 ;ceros decseg
Idaa $32
inca
staa $32
cmpa #$3a
beq decmin
bra sreloj
```

```
decmin Idaa #$30
staa $32 ;ceros unimin
Idaa $33
inca
staa $33
cmpa #$36
beq ceros
bra sreloj
```

```
ceros Idaa #$00
staa $33 ;ceros decmin
```

```
sreloj pulx
pula
rti
```

```
*****fin comparador 2 (reloj)*****
```

```
*****RUTINAS DE INTERRUP. CAP1 Y CAP2*****
```

```
cap1 Idaa #$ff
staa $1023 ;borro bandera de cap1
rti
```

cap2 ldaa #\$ff

staa \$1023 ;borro bandera de capturador2

bogas ldx \$67 ;empiezo a contar el tiempoentre una remada y otra

inx

stx \$67

ldd \$1012 ;cargo con valor cap2

subd \$1010 ;resto valor cap1

std \$52 ;obtengo tiempo en bajo(valor actual)

cpd \$50 ;comparo con valor en bajo anterior

bhs mayor ;si actual es mayor salta a mayor

bra menor ;si valor actual es menor salta a menor

;necesito verificar que tan menor es

;alrededor de 4 de diferencia(ruido)

menor ;implemento la rutina de ruido

ruidon ldd \$50 ;valor anterior

subd \$52 ;valor actual

cpd #\$08

bhs fabspos ;no hay ruido salta a checar bandera

bra salcap2 ;si hay ruido, sale de interrupcion

fabspos ldaa \$66

cmpa #\$ff; ff=habilitada 00=no habilitada

beq val ; en val tbact=tbant

ldaa #\$00

staa \$65 ;desabilito flag abs.negativo

ldd \$50 ;cargo valor anterior

std \$62 ;guardo en absoluto positivo

ldd \$52

std \$50 ;t.bajo actual guardo en tbajoanterior para prox.comparacion

ldaa #\$ff

staa \$66 ;habilito flag de abs . positivo

difabs ldd \$62 ;abspos-absneg

cpd \$60

bhs resta ;doble chequeo

bra salcap2

resta ldd \$62

subd \$60 ;aqui ya es un dato valido

guarda std \$12 ; valor aceleracion valido en \$12 y \$13

```

fbogas ldd $67 ;cargo valor de cuenta bogas
      cpx #$32 ; y no calculo vibraciones 1f80/2d
      blo bbogas ; si es menor salta, limitadoa a 201 bpm

      std $69 ;guardo bogas en esta dir.
      ldx #$00
      stx $67 ;borro cuenta bogas
      ldd $12
      std $17 ;guardo valor aceleracion

si    ldaa #$ff
      staa $19 ;bandera de dato valido
      bra salcap2

bbogas ;ldx #$00 ; borro cuenta de bogas
      ;stx $67
      bra salcap2

mayor ;implemento la rutina de ruido
ruidop ldd $52 ;valor actual
      subd $50 ;valor anterior
      cpd #$08
      bhs fabsneg ;no hay ruido salta a checar bandera
      bra salcap2 ;si hay ruido sale de interrupcion

fabsneg ldaa $65 ;existe minimo relativo? ff=si 00=no
      cmpa #$ff; ff=habilitada 00=no habilitada
      beq val ; guarda tiempo actual como anterior
      ldaa #$00
      staa $66 ;desabilito bandera abs.positivo
      ldd $52
      std $50 ;tbact=tbant tiempo bajo actual=t.bajo anterior
      std $60 ;tbact=absneg t.bajo actual = abs.negativo
      ldaa #$ff
      staa $65 ;habilito flag de abs . negativo
      bra salcap2 ;salto a verificar si voy de subida

val    ldd $52
      std $50 ;guardo valor actual como anterior

salcap2 rti

```

*****FIN INTS. CAP1 Y CAP2*****

*****SUBROUTINAS DEL DISPLAY 811E2 *****

```

dato          psha ;subrutina de escribir dato en display
              pshx

```



```

Idx #$1000
staa $04,x ;datos en portb
bset $03,x,03 ;RS=1 E=1 ;port C pines 0 y 1
;RS=PC0 , E=PC1
bclr $03,x,02 ;RS=1 E=0
bclr $03,x,03 ;RS=0 E=0
bsr delay
pulx
pula
rts

```

```

coman      psha ;sub para mandar comandos al display
pshx
Idx #$1000
staa $04,x ;comandos en port B
bclr $03,x,03 ;RS=0 E=0
bset $03,x,02 ;RS=0 E=1 ;port C pines 0 y 1
;RS=PC0 , E=PC1
bclr $03,x,02 ;RS=0 E=1
bsr delay
pulx
pula
rts

```

```

delay      psha ;delay 1.414 mseg
pshx
ldaa #$ff
retar      deca
nop
nop
nop
bne retar
pulx
pula
rts

```

*****FIN SUBROUTINAS DEL DISPLAY*****

```

org $ffee ;$ffee $e8 interrupcion capturador1
fdb cap1

```

```

org $ffec ;$ffec $e5 interrupcion capturador2
fdb cap2

```

```
org $ffe6 ;interrupcion del comparador 2  
fdb comp2
```

```
org $ffe  
fdb main ;vector de reset
```

9.5 CÓDIGO FUENTE DEL RECEPTOR

```
-- programa final de la tesis este es el receptor
-- modulo de comunicaciones
-- este es el archivo de recepcion despliega los valores en pantalla del receptor
-- se enciende el led amarillo(o verde) cada que se recibe un caracter valido
-- se enciende el led verde cada que se recibe una cadena de caracteres valida
-- se enciende el led rojo cada que se pierde una cadena de caracteres
-- cacracter de inicio es "B" y de parada es "F"
-- la cadena es B_bbb_fff_F
-- este archivo tiene la programacuion para la memoria
```

```
include 16f84_10
```

```
-- standard library
include jlib
include hd447804
pin_a0_direction = output -- led verde o amarillo caracter valido
pin_a1_direction = input -- receptor de la antena
pin_a2_direction = output -- led rojo indica cadena erronea
pin_a3_direction = output -- led verde cadena valida
```

```
pin_b0_direction = input -- boton de inicio de captura
pin_b1_direction = input -- boton de fin de captura
pin_b2_direction = output
pin_b3_direction = output
port_b_high_direction = all_output
```

```
var bit valido = off
var byte dato = 0
```

```
var byte bogas1 = 0 -- variables de dato
var byte bogas2 = 0
var byte bogas3 = 0
var byte bogas4 = 0
var byte bogas5 = 0
var byte bogas6 = 0
```

```
var byte bogas1v = 0 -- variables de dato
var byte bogas2v = 0
var byte bogas3v = 0
var byte bogas4v = 0
var byte bogas5v = 0
var byte bogas6v = 0
```

```
var byte cadena = 0 -- banderas de dato
```

```
var byte m1 = 0 -- variables memoria
var byte m12 = 0
```

```
var byte m2 = 0
var byte m22 = 0
```

```
var byte m3 = 0
var byte m32 = 0
```

```
var byte m4 = 0
var byte m42 = 0
```

```
var byte m5 = 0
var byte m52 = 0
```

```
var byte r = 0 -- cuenta de memorias
var byte k = 0 -- revision de memorias
```

```
var byte u
var byte v
```

```
procedure escribe ( byte in out u, byte in out v ) is
  hd44780_position ( 72 ) -- escribo valor de bogas
  hd44780 = (u / 100) + 48 -- centenas
  hd44780 = (( u - ( u / 100 ) * 100 ) / 10 ) + 48 -- decenas
  hd44780 = (u - ((u / 10 ) * 10)) + 48 -- unidades
  hd44780_position ( 8 ) -- escribo valor de fuerza
  hd44780 = (v / 100) + 48 -- centenas
  hd44780 = (( v - ( v / 100 ) * 100 ) / 10 ) + 48 -- decenas
  hd44780 = (v - ((v / 10 ) * 10)) + 48 -- unidades
end procedure
```

```
hd44780_clear -- inicializacion
```

```
hd44780_position ( 6 )
hd44780 = "E" -- verifico que entremos al programa
hd44780 = "S"
hd44780 = "P"
hd44780 = "E"
hd44780 = "R"
hd44780 = "E"
pin_a0 = on
pin_a2 = on
pin_a3 = on -- chequeo de leds y lcd
```

```
delay_500ms ( 2 )
```

```
pin_a0 = off
pin_a2 = off
pin_a3 = off
```

hd44780_clear -- inicializacion**-- inicio****hd44780_position (67) -- c1****hd44780 = 66 -- "B"****hd44780_position (3) -- 81****hd44780 = 70 -- "F"****var bit bit_llegada at dato : 7 -- defino el bit de llegada en el bit 7 de dato****forever loop -- espero llegada de datos protocolo = (fin)1_9bits_0(inicio)****-- *****inicio a recibir caracter*********while valido == off loop****pin_a0 = off****if pin_a1 == on then -- si llega un 0 checa si es bit de inicio****delay_1us (200)****if pin_a1 == on then -- es el bit de inicio valido****for 8 loop****delay_1us (200)****delay_1us (200)****dato = dato >> 1****bit_llegada = pin_a1****end loop****delay_1us (200)****delay_1us (200)****if pin_a1 == on then -- bit de parada valido****valido = on****pin_a0 = on****end if****end if -- fin del bit de inicio valido****end if -- fin del bit de inicio****end loop****valido = off****-- ***** fin de recepcion de caracter *********if dato == 66 then**

```
if          cadena = 1 -- empieza recepcion de la cadena          end
    elsif cadena == 1 then
        bogas1 = dato
        cadena = 2

    elsif cadena == 2 then
        bogas2 = dato
        cadena = 3

    elsif cadena == 3 then
        bogas3 = dato
        cadena = 4

    elsif cadena == 4 then
        bogas4 = dato
        cadena = 5

    elsif cadena == 5 then
        bogas5 = dato
        cadena = 6

    elsif cadena == 6 then
        bogas6 = dato
        cadena = 7
    elsif cadena == 7 then
        cadena = 0

        if dato == 70 then

            bogas1v = bogas1 -- variables de dato
            bogas2v = bogas2
            bogas3v = bogas3
            bogas4v = bogas4
            bogas5v = bogas5
            bogas6v = bogas6

            hd44780_position ( 68 ) -- escribo valor de bogas
            hd44780 = bogas1v
            hd44780 = bogas2v
            hd44780 = bogas3v
            hd44780_position ( 4 ) -- escribo valor de fuerza
            hd44780 = bogas4v
            hd44780 = bogas5v
            hd44780 = bogas6v

        end if
    end if
```

```
        end if

if (r == 200) & (pin_b0 == off) then -- borro memorias
    while pin_b0 == off loop asm nop end loop -- efecto rebote

    m1 = 0 -- variables memoria
    m12 = 0
    m2 = 0
    m22 = 0
    m3 = 0
    m32 = 0
    m4 = 0
    m42 = 0
    m5 = 0
    m52 = 0
        r = 0
    hd44780_position ( 76 )
    hd44780 = 48 -- & pregunta de borrar

end if -- fin borro memorias

if pin_b0 == off then -- inicio captura de memorias
    while pin_b0 == off loop asm nop end loop -- efecto rebote

    r = r + 1

        if r == 1 then -- seleccion memoria
m1 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m12 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)
    hd44780_position ( 76 )
        hd44780 = 49 -- 1
        elsif r == 2 then

            hd44780_position ( 76 )
            hd44780 = 50 -- 2

m2 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m22 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

            elsif r == 3 then
                hd44780_position ( 76 )
                hd44780 = 51 -- 3

m3 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m32 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

            elsif r == 4 then

                hd44780_position ( 76 )
                hd44780 = 52 -- 4
```

```

m4 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m42 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

```

```

elseif r == 5 then
hd44780_position ( 76 )
hd44780 = 53 -- 5

```

```

m5 = ( (bogas1v - 48) * 100 ) + ((bogas2v - 48) * 10) + (bogas3v - 48)
m52 = ( (bogas4v - 48) * 100 ) + ((bogas5v - 48) * 10) + (bogas6v - 48)

```

```

end if -- fin seleccion memoria
end if -- fin captura de memorias

```

```

if pin_b1 == off then -- inicio revision de memorias

```

```

while pin_b1 == off loop asm nop end loop -- efecto rebote

```

```

k = k + 1

```

```

if k == 1 then -- seleccion memoria

```

```

hd44780_position ( 12 )

```

```

hd44780 = 49 -- 1

```

```

hd44780 = 32 -- 1

```

```

escribe ( m1, m12 )

```

```

elseif k == 2 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 50 -- 2

```

```

escribe ( m2, m22 )

```

```

elseif k == 3 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 51 -- 3

```

```

escribe ( m3, m32 )

```

```

elseif k == 4 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 52 -- 4

```

```

escribe ( m4, m42 )

```

```

elseif k == 5 then

```

```

hd44780_position ( 12 )

```

```

hd44780 = 53 -- 5

```

```

escribe ( m5, m52 )

```

```

elseif k == 6 then

```

```

hd44780_position ( 13 )

```

```

hd44780 = 38 -- & pregunta de borrar

```

```

r = 200

```

```

k = 0

```

```

end if -- fin seleccion memoria

```

```

end if -- fin revision memorias

```

```

end loop

```


9.6 CÓDIGO FUENTE DE LOS PROGRAMAS t1.asm, tbajo.asm, t2.asm

Estos programas se realizaron con las subrutinas descritas, por lo cual se omiten y solo se deja la parte funcional del programa.

****T1.ASM**

****REGISTRA EL TIEMPO EN ALTO DE LA SEÑAL PROVENIENTE DEL ACELEROMETRO**

****REGISTROS CAP1=\$1010-\$1011**

**** CAP2=\$1012-\$1013**

*******REGISTROS*******

Idx #\$1000

bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg

bset \$22,x,\$06 ;habilito capturadores 1 y 2

bset \$21,x,\$24 ;flancos de bajada cap1 y de subida cap2

bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******banderas y valoresiniciales*******

Idaa #\$aa ;numero de muestras= 170 decimal

staa \$20 ;guardo ese valor en esta direcciøn

Idx #\$0000

stx \$17 ;suma de muestras en esta dir.

*******PROGRAMA*******

inicio

division Idd \$00 ;valor de la resta

Idx #\$a ;divido entre 10 decimalresiduo en D

idiv

stx \$02 ;resultado en x residuo en D

stab \$09 ;UNIDADES

Idd \$02

Idx #\$a

idiv

stx \$02 ;resultado en x residuo en d

stab \$08 ;DECENAS

Idd \$02

Idx #\$a

idiv

stx \$02

stab \$07 ;CENTENAS

Idd \$02

Idx #\$a

```

idiv
stx $02
stab $06 ;UNIDADES MILES

```

```

ldd $02
ldx #$a
idiv
stx $02
stab $05 ;DECENAS MILES

```

```

ldaa #$87 ;posicion del cursor del display
jsr coman
escribe ldx #$05 ;carga datos de las direcciones resultado
ldab #$05 ;y despliega en el display
ya ldaa $0,x ;puntero de resultados
adda #$30 ;se le suman 30 para que despliegue ASCII
jsr dato
inx
decb
bne ya

jsr mas

bra inicio

```

*****ROUTINAS DE INTERRUP. CAP1 Y CAP2*****

```

cap1 ldaa #$04
staa $1023 ;borro bandera de cap1

```

```

ldd $1010 ;valor flanco de bajada
cpd $1012
bhs resta

```

```

ldd $1012
subd $1010
std $15
bra sigue

```

```

resta ldd $1010
subd $1012 ;valor flanco de subida
std $15 ;guardo

```

```

sigue ldd $15
addd $17 ;sumo el valor a esta dir.
std $17
ldaa $20 ;sumo 150 veces
deca
staa $20
bne scap1 ;salta a scap1(sale de interrup) si no es cero

```

```
Idd $17
Idx #$aa ;divido la suma entre 170 muestras (promedio)
idiv
stx $00 ;resultado en x
std $13 ;residuo en D
Idaa #$aa ;pongo la cuenta de 150 muestras otra vez
staa $20
Idx #$00 ;pongo a ceros la suma de muestras
stx $17
```

```
scap1    rti
```

```
cap2    Idaa #$02
        staa $1023 ;borro bandera de capturador2
```

```
        rti
```

```
*****FIN INTS. CAP1 Y CAP2*****
```

```
mas     pshx
        psha
        Idx #$ff
si
        jsr delay
        jsr delay

        dex
        bne si
        pula
        pulx
        rts
```

****TBAJO.ASM**
****REGISTRA EL TIEMPOO EN BAJO DE LA SEÑAL**
****PROVENIENTE DEL ACELERÓMETRO**

*******REGISTROS*******

Idx # \$1000
bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg
bset \$22,x,\$06 ;habilito capturadores 1 y 2
bset \$21,x,\$24 ;flancos de bajada cap1 y de subida cap2
bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******banderas y valoresiniciales*******

Idaa # \$72 ;numero de muestras= 115 decimal
staa \$20 ;guardo ese valor en esta direccion
Idx # \$0000
stx \$17 ;suma de muestras en esta dir.

*******PROGRAMA*******

inicio

division Idd \$00 ;valor de la resta
Idx # \$a ;divido entre 10 decimalresiduo en D
idiv
stx \$02 ;resultado en x residuo en D
stab \$09 ;UNIDADES

Idd \$02
Idx # \$a
idiv
stx \$02 ;resultado en x residuo en d
stab \$08 ;DECENAS

Idd \$02
Idx # \$a
idiv
stx \$02
stab \$07 ;CENTENAS

Idd \$02
Idx # \$a
idiv
stx \$02

stab \$06 ;UNIDADES MILES

Idd \$02

Idx #\$a

idiv

stx \$02

stab \$05 ;DECENAS MILES

Idaa #\$87 ;posicion del cursor del display

jsr coman

escribe Idx #\$05 ;carga datos de las direcciones resultado

ldab #\$05 ;y despliega en el display

ya Idaa \$0,x ;puntero de resultados

adda #\$30 ;se le suman 30 para que despliegue ASCII

jsr dato

inx

decb

bne ya

jsr mas

bra inicio

*****RUTINAS DE INTERRUP. CAP1 Y CAP2*****

cap1 Idaa #\$04

staa \$1023 ;borro bandera de cap1

rti

cap2 Idaa #\$02

staa \$1023 ;borro bandera de capturador2

Idd \$1012 ;valor flanco de subida

subd \$1010 ;valor flanco de bajada

std \$15 ;guardo

Idd \$15

addd \$17 ;sumo el valor a esta dir.

std \$17

Idaa \$20 ;sumo 150 veces

deca

staa \$20

bne scap2 ;salta a scap1(sale de interrup) si no es cero

Idd \$17

Idx #\$72 ;divido la suma entre 170 muestras (promedio)

```
    idiv
    stx $00 ;resultado en x
    std $13 ;residuo en D
    ldaa #$72 ;pongo la cuenta de 150 muestras otra vez
    staa $20
    ldx #$00 ;pongo a ceros la suma de muestras
    stx $17
scap2    rti
```

```
*****FIN INTS. CAP1 Y CAP2*****
```

```
mas     pshx
        psha
        ldx #$ff
si
        jsr delay
        jsr delay

        dex
        bne si
        pula
        pulx
        rts
```

****T2.ASM**

****REGISTRA EL PERIODO DE UNA SEÑAL PROVENIENTE DEL ACELEROMETRO**

****REGISTROS CAP1=\$1010-\$1011**

**** CAP2=\$1012-\$1013**

*******REGISTROS*******

Idx # \$1000

bset \$24,x,\$03 ;incremento de cuenta del tiempo cada 8useg

bset \$22,x,\$02 ;habilito capturador 2

bset \$21,x,\$04 ;flancos de subida cap2

bset \$20,x,\$00 ;no hay salida hardware del comparador

bset \$07,x,\$ff ;port C como salidas

cli ;habilito interrupciones

*******FIN REGISTROS*******

*******valores*******

Idaa # \$00

staa \$10 ; bandera de periodo 2

*******banderas y valores iniciales*******

Idaa # \$00

staa \$10 ;bandera periodo

staa \$35 ;bandera promedio

staa \$20 ;valor 1ra captura

staa \$22 ;valor periodo

staa \$25 ;valor suma

staa \$30 ;bandera cuenta

*******PROGRAMA*******

Idaa # \$80

jsr coman

Idaa # \$35

jsr dato

inicio Idaa \$35

cmpa # \$ff

beq calcula

bra inicio

calcula Idaa # \$00

staa \$35

ldd \$25

ldx # \$30

idiv ;resultado en x residuo en d

stx \$40

Idd \$40

jsr division
 ldaa #\$85
 jsr escribe

ldd #\$ffff
 jsr division
 ldaa #\$c5
 jsr escribe
 ldx #\$0000
 stx \$25 ;borro suma

bra inicio

*****FIN PROGRAMA*****

*****SUBROUTINAS*****

*****division*****

division psha
 pshx
 pshb
 ldx #\$a ;divido entre 10 decimalresiduo en D
 idiv
 stx \$02 ;resultado en x residuo en D
 stab \$09 ;UNIDADES

ldd \$02
 ldx #\$a
 idiv
 stx \$02 ;resultado en x residuo en d
 stab \$08 ;DECENAS

ldd \$02
 ldx #\$a
 idiv
 stx \$02
 stab \$07 ;CENTENAS

ldd \$02
 ldx #\$a
 idiv
 stx \$02
 stab \$06 ;unidades millares

ldd \$02
 ldx #\$a
 idiv


```

stx $02
stab $05 ;decenas millares
pulb
pulx
pula
rts

```

*****fin division*****

*****escribe 5 lugares*****

```

escribe  psha  ;posicion del cursor del display
        pshx
        pshb
        jsr coman
        ldx #$05 ;carga datos de las direcciones resultado
        ldab #$05 ;y despliega en el display
ya      ldaa $0,x ;puntero de resultados
        adda #$30 ;se le suman 30 para que despliegue ASCII
        jsr dato
        inx
        decb
        bne ya
        pulb
        pulx
        pula
        rts

```

*****fin escribe*****

*****RUTINAS DE INTERRUP. CAP1 Y CAP2*****

```

cap1    ldaa #$ff
        staa $1023 ;borro bandera de cap1
        rti

```

```

cap2    ldaa #$ff
        staa $1023 ;borro bandera de capturador2

```

```

        ldaa $10
        cmpa #$00
        beq periodo ;checo si es el inicio del periodo

```

```

        ldaa #$00
        staa $10

```

```

        ldd $1012
        subd $20

```

std \$22 ;guardo valor de periodo

ldd \$25

add \$22

std \$25 ; guardo valor suma

ldaa \$30 ;cuenta

cmpa #30 ;48 veces

beq sal

inca

staa \$30

bra salcap2

sal ;borro cuenta de suma

ldaa #ff

staa \$35 ;bandera de cuenta

ldaa #00

staa \$30

bra salcap2

periodo ldd \$1012 ;cargo con valor cap2

std \$20 ;guardo valor cap2

ldaa #ff

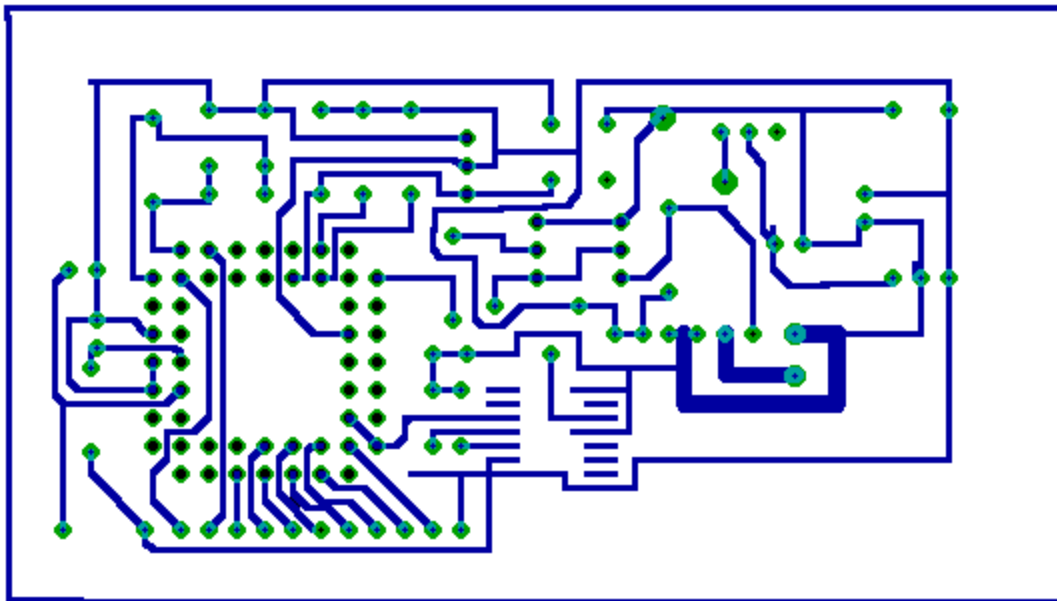
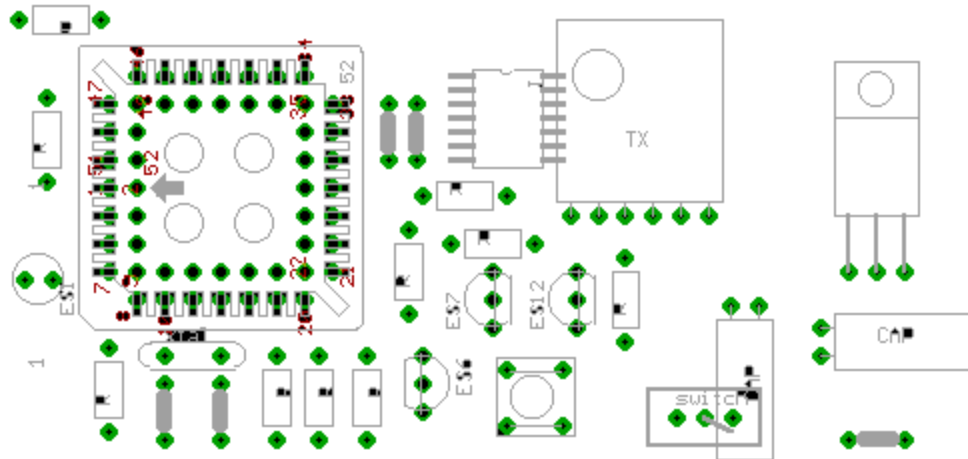
staa \$10 ;bandera periodo

bra salcap2

salcap2 rti

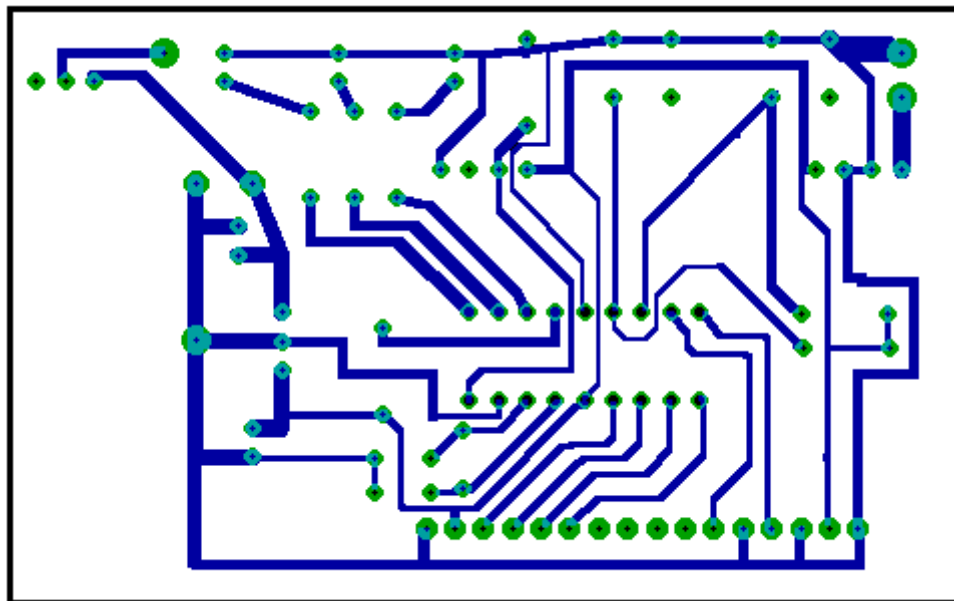
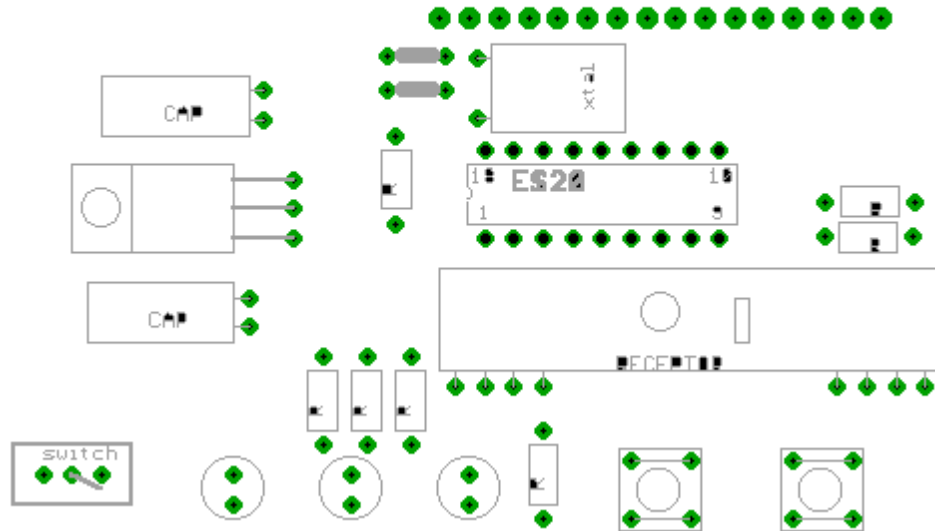
*****FIN INTS. CAP1 Y CAP2*****

9.7 CIRCUITO IMPRESO DEL TRANSMISOR

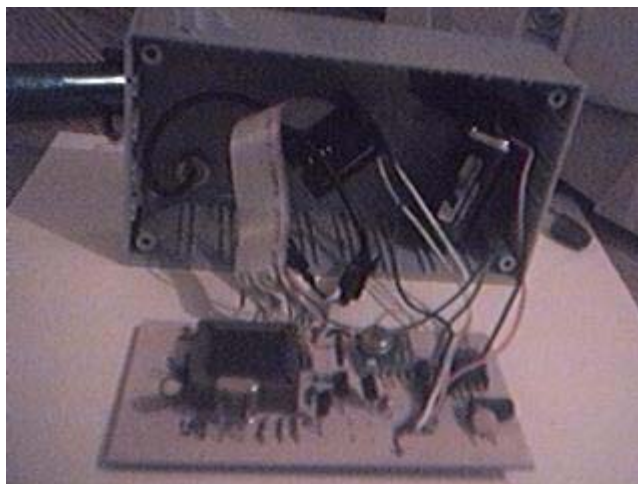
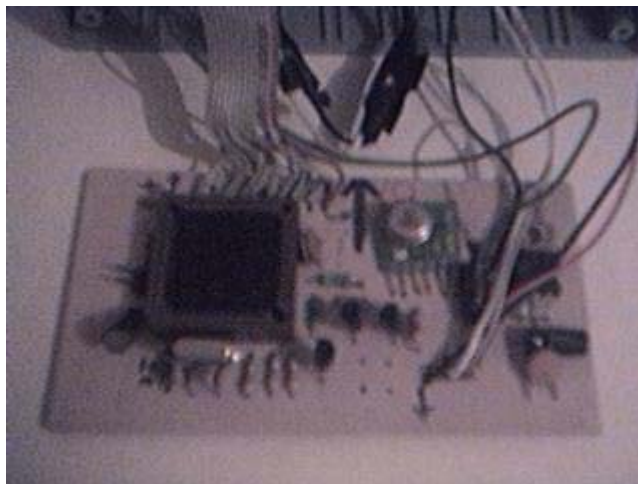
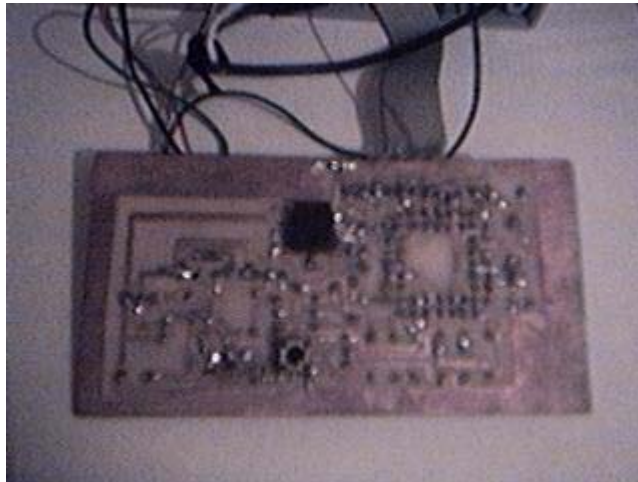


9.8 CIRCUITO IMPRESO DEL RECEPTOR

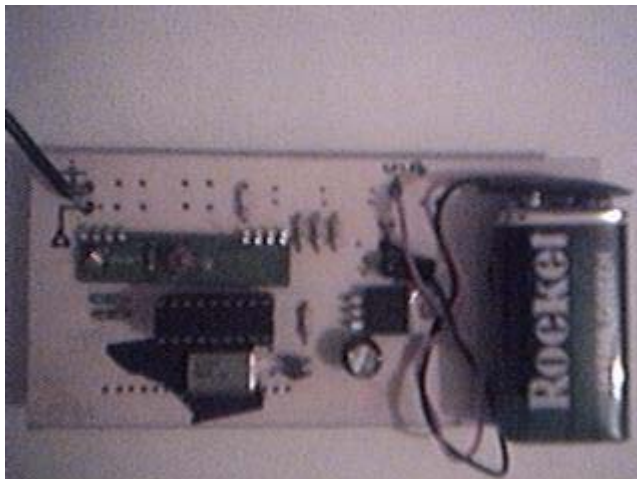
Circuio Impreso "Receptor" ubicacion de componentes



9.9 IMÁGENES DEL CIRCUITO IMPRESO ENSAMBLADO DEL TRANSMISOR



9.10 IMÁGENES DEL CIRCUITO IMPRESO ENSAMBLADO DEL RECEPTOR



9.11 HOJAS DE ESPECIFICACIONES DE LOS DISPOSITIVOS EMPLEADOS



Low Cost $\pm 2 g/\pm 10 g$ Dual Axis iMEMS[®] Accelerometers with Digital Output

ADXL202/ADXL210

FEATURES

2-Axis Acceleration Sensor on a Single IC Chip
Measures Static Acceleration as Well as Dynamic Acceleration
Duty Cycle Output with User Adjustable Period
Low Power <0.6 mA
Faster Response than Electrolytic, Mercury or Thermal Tilt Sensors
Bandwidth Adjustment with a Single Capacitor Per Axis
5 mg Resolution at 60 Hz Bandwidth
+3 V to +5.25 V Single Supply Operation
1000 g Shock Survival

APPLICATIONS

2-Axis Tilt Sensing
Computer Peripherals
Inertial Navigation
Seismic Monitoring
Vehicle Security Systems
Battery Powered Motion Sensing

GENERAL DESCRIPTION

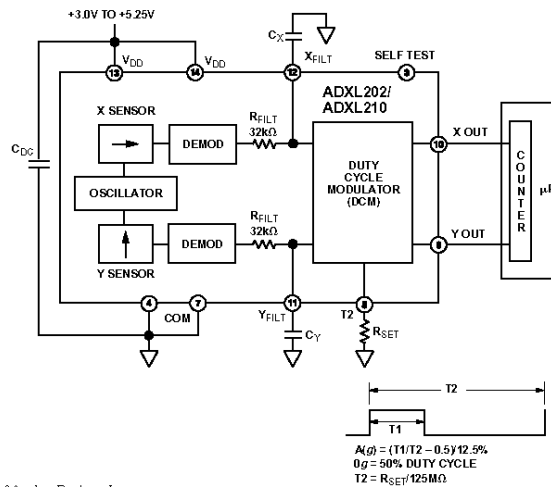
The ADXL202/ADXL210 are low cost, low power, complete 2-axis accelerometers with a measurement range of either $\pm 2 g/\pm 10 g$. The ADXL202/ADXL210 can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are digital signals whose duty cycles (ratio of pulse-width to period) are proportional to the acceleration in each of the 2 sensitive axes. These outputs may be measured directly with a microprocessor counter, requiring no A/D converter or glue logic. The output period is adjustable from 0.5 ms to 10 ms via a single resistor (R_{SET}). If a voltage output is desired, a voltage output proportional to acceleration is available from the X_{FILT} and Y_{FILT} pins, or may be reconstructed by filtering the duty cycle outputs.

The bandwidth of the ADXL202/ADXL210 may be set from 0.01 Hz to 5 kHz via capacitors C_X and C_Y . The typical noise floor is $500 \mu g/\sqrt{Hz}$ allowing signals below 5 mg to be resolved for bandwidths below 60 Hz.

The ADXL202/ADXL210 is available in a hermetic 14-lead Surface Mount CERPAK, specified over the $0^\circ C$ to $+70^\circ C$ commercial or $-40^\circ C$ to $+85^\circ C$ industrial temperature range.

FUNCTIONAL BLOCK DIAGRAM



iMEMS is a registered trademark of Analog Devices, Inc.

REV. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 1999

ADXL202/ADXL210—SPECIFICATIONS ($T_A = T_{MIN}$ to T_{MAX} , $T_A = +25^\circ\text{C}$ for J Grade only, $V_{DD} = +5\text{V}$, $R_{SET} = 125\text{ k}\Omega$, Acceleration = 0 g, unless otherwise noted)

Parameter	Conditions	ADXL202/JQC/AQC			ADXL210/JQC/AQC			Units
		Min	Typ	Max	Min	Typ	Max	
SENSOR INPUT	Each Axis							
Measurement Range ¹		± 1.5	± 2		± 8	± 10		g
Nonlinearity	Best Fit Straight Line		0.2			0.2		% of FS
Alignment Error ²			± 1			± 1		Degrees
Alignment Error	X Sensor to Y Sensor		± 0.01			± 0.01		Degrees
Transverse Sensitivity ³			± 2			± 2		%
SENSITIVITY	Each Axis							
Duty Cycle per g	T1/T2 @ +25°C	10	12.5	15	3.2	4.0	4.8	%/g
Sensitivity, Analog Output	At Pins X _{FILT} , Y _{FILT}		312			100		mV/g
Temperature Drift ⁴	Δ from +25°C		± 0.5			± 0.5		% Rdg
ZERO g BIAS LEVEL	Each Axis							
0 g Duty Cycle	T1/T2	25	50	75	42	50	58	%
Initial Offset			± 2			± 2		g
0 g Duty Cycle vs. Supply			1.0	4.0		1.0	4.0	%/V
0 g Offset vs. Temperature ⁴	Δ from +25°C		2.0			2.0		mg/°C
NOISE PERFORMANCE								
Noise Density ⁵	@ +25°C		500	1000		500	1000	$\mu\text{g}/\sqrt{\text{Hz}}$
FREQUENCY RESPONSE								
3 dB Bandwidth	Duty Cycle Output		500			500		Hz
3 dB Bandwidth	At Pins X _{FILT} , Y _{FILT}		5			5		kHz
Sensor Resonant Frequency			10			14		kHz
FILTER								
R _{FILT} Tolerance	32 k Ω Nominal		± 15			± 15		%
Minimum Capacitance	At X _{FILT} , Y _{FILT}	1000			1000			pF
SELF TEST								
Duty Cycle Change	Self-Test "0" to "1"		10			10		%
DUTY CYCLE OUTPUT STAGE								
F _{SET}		125 M Ω /R _{SET}			125 M Ω /R _{SET}			
R _{SET} Tolerance	R _{SET} = 125 k Ω	0.7		1.3	0.7		1.3	kHz
Output High Voltage	I = 25 μ A	V _S - 200 mV			V _S - 200 mV			mV
Output Low Voltage	I = 25 μ A			200			200	mV
T2 Drift vs. Temperature			35			35		ppm/°C
Rise/Fall Time			200			200		ns
POWER SUPPLY								
Operating Voltage Range		3.0		5.25	2.7		5.25	V
Specified Performance		4.75		5.25	4.75		5.25	V
Quiescent Supply Current			0.6	1.0		0.6	1.0	mA
Turn-On Time ⁶	To 99%	160 C _{FILT} + 0.3			160 C _{FILT} + 0.3			ms
TEMPERATURE RANGE								
Operating Range	JQC	0		+70	0		+70	°C
Specified Performance	AQC	-40		+85	-40		+85	°C

NOTES

¹For all combinations of offset and sensitivity variation.

²Alignment error is specified as the angle between the true and indicated axis of sensitivity.

³Transverse sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.

⁴Specification refers to the maximum change in parameter from its initial at +25°C to its worst case value at T_{MIN} to T_{MAX}.

⁵Noise density ($\mu\text{g}/\sqrt{\text{Hz}}$) is the average noise at any frequency in the bandwidth of the part.

⁶C_{FILT} in μF . Addition of filter capacitor will increase turn on time. Please see the Application section on power cycling.

All min and max specifications are guaranteed. Typical specifications are not tested or guaranteed.

Specifications subject to change without notice.

ADXL202/ADXL210

ABSOLUTE MAXIMUM RATINGS*

Acceleration (Any Axis, Unpowered for 0.5 ms)	1000 g
Acceleration (Any Axis, Powered for 0.5 ms)	500 g
+V _S	-0.3 V to +7.0 V
Output Short Circuit Duration (Any Pin to Common)	Indefinite
Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; the functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Drops onto hard surfaces can cause shocks of greater than 1000 g and exceed the absolute maximum rating of the device. Care should be exercised in handling to avoid damage.

PIN FUNCTION DESCRIPTIONS

Pin	Name	Description
1	NC	No Connect
2	V _{TP}	Test Point, Do Not Connect
3	ST	Self Test
4	COM	Common
5	T2	Connect R _{SET} to Set T2 Period
6	NC	No Connect
7	COM	Common
8	NC	No Connect
9	Y _{OUT}	Y Axis Duty Cycle Output
10	X _{OUT}	X Axis Duty Cycle Output
11	Y _{FILT}	Connect Capacitor for Y Filter
12	X _{FILT}	Connect Capacitor for X Filter
13	V _{DD}	+3 V to +5.25 V, Connect to 14
14	V _{DD}	+3 V to +5.25 V, Connect to 13

PACKAGE CHARACTERISTICS

Package	θ _{JA}	θ _{JC}	Device Weight
14-Lead CERPAK	110°C/W	30°C/W	5 Grams

ORDERING GUIDE

Model	g Range	Temperature Range	Package Description	Package Option
ADXL202JQC	±2	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL202AQC	±2	-40°C to +85°C	14-Lead CERPAK	QC-14
ADXL210JQC	±10	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL210AQC	±10	-40°C to +85°C	14-Lead CERPAK	QC-14

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADXL202/ADXL210 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN CONFIGURATION

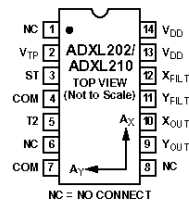


Figure 1 shows the response of the ADXL202 to the Earth's gravitational field. The output values shown are nominal. They are presented to show the user what type of response to expect from each of the output pins due to changes in orientation with respect to the Earth. The ADXL210 reacts similarly with output changes appropriate to its scale.

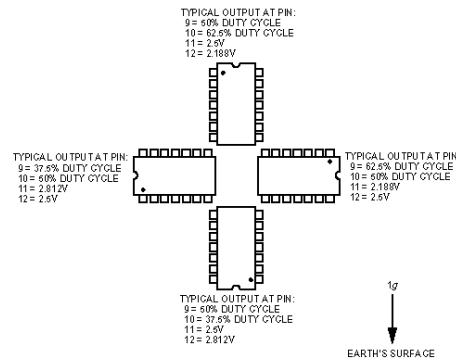


Figure 1. ADXL202/ADXL210 Nominal Response Due to Gravity



ADXL202/ADXL210

TYPICAL CHARACTERISTICS (@ +25°C $R_{SET} = 125\text{ k}\Omega$, $V_{DD} = +5\text{ V}$, unless otherwise noted)

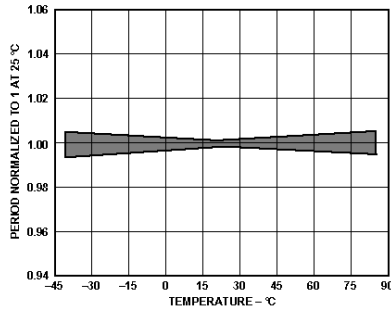


Figure 2. Normalized DCM Period (T_2) vs. Temperature

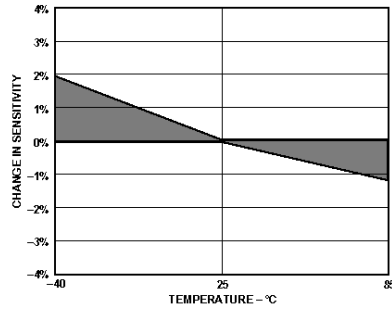


Figure 5. Typical X Axis Sensitivity Drift Due to Temperature

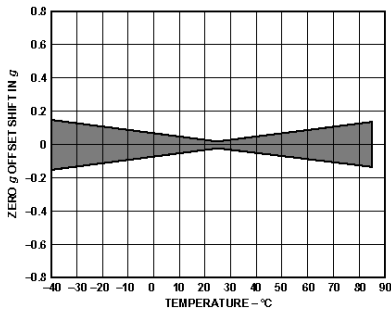


Figure 3. Typical Zero g Offset vs. Temperature

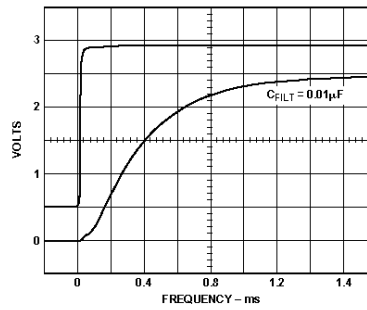


Figure 6. Typical Turn-On Time

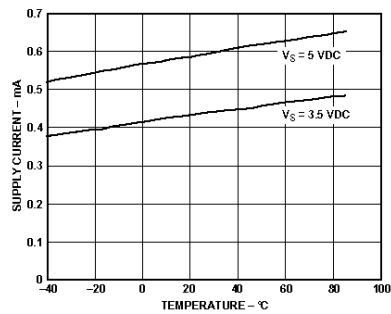


Figure 4. Typical Supply Current vs. Temperature

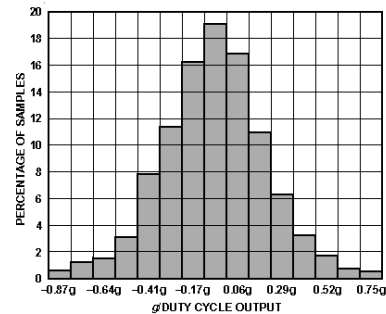


Figure 7. Typical Zero g Distribution at +25°C

ADXL202/ADXL210

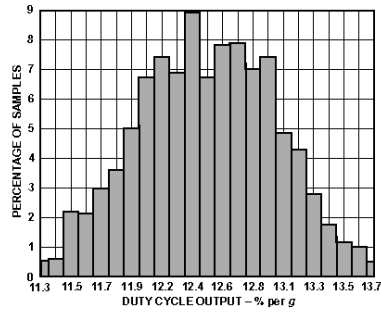


Figure 8. Typical Sensitivity per g at +25°C

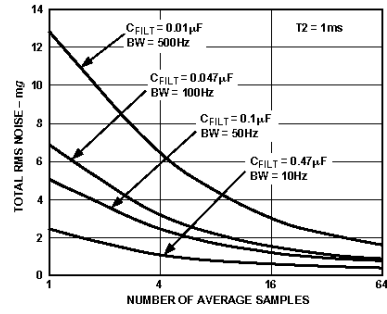


Figure 10. Typical Noise at Digital Outputs

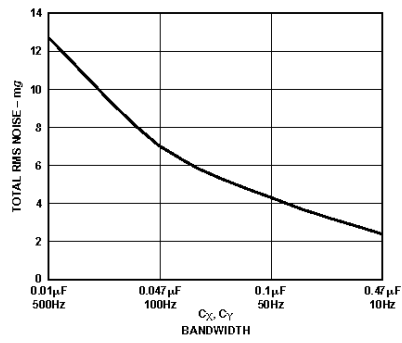


Figure 9. Typical Noise at X_{FILT} Output

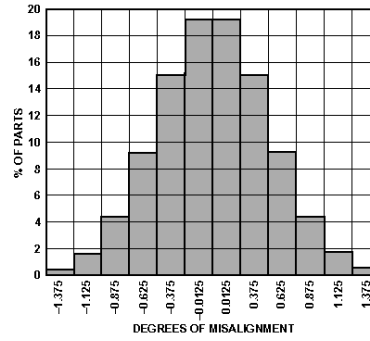


Figure 11. Rotational Die Alignment

ADXL202/ADXL210

DEFINITIONS

T1	Length of the “on” portion of the cycle.
T2	Length of the total cycle.
Duty Cycle	Ratio of the “on” time (T1) of the cycle to the total cycle (T2). Defined as T1/T2 for the ADXL202/ADXL210.
Pulsewidth	Time period of the “on” pulse. Defined as T1 for the ADXL202/ADXL210.

THEORY OF OPERATION

The ADXL202/ADXL210 are complete dual axis acceleration measurement systems on a single monolithic IC. They contain a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open loop acceleration measurement architecture. For each axis, an output circuit converts the analog signal to a duty cycle modulated (DCM) digital signal that can be decoded with a counter/timer port on a microprocessor. The ADXL202/ADXL210 are capable of measuring both positive and negative accelerations to a maximum level of $\pm 2 g$ or $\pm 10 g$. The accelerometer measures static acceleration forces such as gravity, allowing it to be used as a tilt sensor.

The sensor is a surface micromachined polysilicon structure built on top of the silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and central plates attached to the moving mass. The fixed plates are driven by 180° out of phase square waves. An acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration.

The output of the demodulator drives a duty cycle modulator (DCM) stage through a $32 k\Omega$ resistor. At this point a pin is available on each channel to allow the user to set the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

After being low-pass filtered, the analog signal is converted to a duty cycle modulated signal by the DCM stage. A single resistor sets the period for a complete cycle (T2), which can be set between 0.5 ms and 10 ms (see Figure 12). A 0 g acceleration produces a nominally 50% duty cycle. The acceleration signal can be determined by measuring the length of the T1 and T2 pulses with a counter/timer or with a polling loop using a low cost microcontroller.

An analog output voltage can be obtained either by buffering the signal from the X_{FILT} and Y_{FILT} pin, or by passing the duty cycle signal through an RC filter to reconstruct the dc value.

The ADXL202/ADXL210 will operate with supply voltages as low as 3.0 V or as high as 5.25 V.

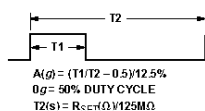


Figure 12. Typical Output Duty Cycle

APPLICATIONS

POWER SUPPLY DECOUPLING

For most applications a single 0.1 μF capacitor, C_{DC}, will adequately decouple the accelerometer from signal and noise on the power supply. However, in some cases, especially where digital devices such as microcontrollers share the same power supply, digital noise on the supply may cause interference on the ADXL202/ADXL210 output. This is often observed as a slowly undulating fluctuation of voltage at X_{FILT} and Y_{FILT}. If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite beads, may be inserted in the ADXL202/ADXL210's supply line.

DESIGN PROCEDURE FOR THE ADXL202/ADXL210

The design procedure for using the ADXL202/ADXL210 with a duty cycle output involves selecting a duty cycle period and a filter capacitor. A proper design will take into account the application requirements for bandwidth, signal resolution and acquisition time, as discussed in the following sections.

V_{DD}

The ADXL202/ADXL210 have two power supply (V_{DD}) Pins: 13 and 14. These two pins should be connected directly together.

COM

The ADXL202/ADXL210 have two commons, Pins 4 and 7. These two pins should be connected directly together and Pin 7 grounded.

V_{TP}

This pin is to be left open; make no connections of any kind to this pin.

Decoupling Capacitor C_{DC}

A 0.1 μF capacitor is recommended from V_{DD} to COM for power supply decoupling.

ST

The ST pin controls the self-test feature. When this pin is set to V_{DD}, an electrostatic force is exerted on the beam of the accelerometer. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output will be 10% at the duty cycle outputs (corresponding to 800 mg). This pin may be left open circuit or connected to common in normal use.

Duty Cycle Decoding

The ADXL202/ADXL210's digital output is a duty cycle modulator. Acceleration is proportional to the ratio T1/T2. The nominal output of the ADXL202 is:

$$0 g = 50\% \text{ Duty Cycle}$$

Scale factor is 12.5% Duty Cycle Change per g

The nominal output of the ADXL210 is:

$$0 g = 50\% \text{ Duty Cycle}$$

Scale factor is 4% Duty Cycle Change per g

These nominal values are affected by the initial tolerance of the device including zero g offset error and sensitivity error.

T2 does not have to be measured for every measurement cycle. It need only be updated to account for changes due to temperature, (a relatively slow process). Since the T2 time period is shared by both X and Y channels, it is necessary only to measure it on one channel of the ADXL202/ADXL210. Decoding algorithms for various microcontrollers have been developed. Consult the appropriate Application Note.

ADXL202/ADXL210

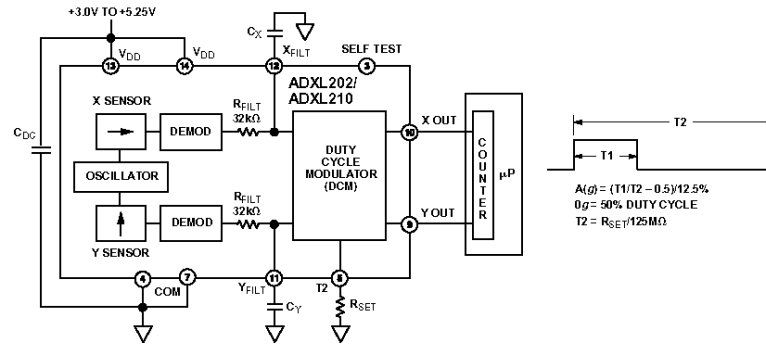


Figure 13. Block Diagram

Setting the Bandwidth Using C_X and C_Y

The ADXL202/ADXL210 have provisions for bandlimiting the X_{FILT} and Y_{FILT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is:

$$F_{-3dB} = \frac{1}{(2\pi (32\text{ k}\Omega) \times C(x, y))}$$

or, more simply, $F_{-3dB} = \frac{5\ \mu F}{C_{(x, y)}}$

The tolerance of the internal resistor (R_{FILT}), can vary as much as $\pm 25\%$ of its nominal value of 32 k Ω ; so the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for $C_{(x, y)}$ is required in all cases.

Table I. Filter Capacitor Selection, C_X and C_Y

Bandwidth	Capacitor Value
10 Hz	0.47 μF
50 Hz	0.10 μF
100 Hz	0.05 μF
200 Hz	0.027 μF
500 Hz	0.01 μF
5 kHz	0.001 μF

Setting the DCM Period with R_{SET}

The period of the DCM output is set for both channels by a single resistor from R_{SET} to ground. The equation for the period is:

$$T2 = \frac{R_{SET} (\Omega)}{125\text{ M}\Omega}$$

A 125 k Ω resistor will set the duty cycle repetition rate to approximately 1 kHz, or 1 ms. The device is designed to operate at duty cycle periods between 0.5 ms and 10 ms.

Table II. Resistor Values to Set T2

T2	R_{SET}
1 ms	125 k Ω
2 ms	250 k Ω
5 ms	625 k Ω
10 ms	1.25 M Ω

Note that the R_{SET} should always be included, even if only an analog output is desired. Use an R_{SET} value between 500 k Ω and 2 M Ω when taking the output from X_{FILT} or Y_{FILT} . The R_{SET} resistor should be placed close to the T2 Pin to minimize parasitic capacitance at this node.

Selecting the Right Accelerometer

For most tilt sensing applications the ADXL202 is the most appropriate accelerometer. Its higher sensitivity (12.5%/g allows the user to use a lower speed counter for PWM decoding while maintaining high resolution. The ADXL210 should be used in applications where accelerations of greater than $\pm 2\text{ g}$ are expected.

MICROCOMPUTER INTERFACES

The ADXL202/ADXL210 were specifically designed to work with low cost microcontrollers. Specific code sets, reference designs, and application notes are available from the factory. This section will outline a general design procedure and discuss the various trade-offs that need to be considered.

The designer should have some idea of the required performance of the system in terms of:

Resolution: the smallest signal change that needs to be detected.

Bandwidth: the highest frequency that needs to be detected.

Acquisition Time: the time that will be available to acquire the signal on each axis.

These requirements will help to determine the accelerometer bandwidth, the speed of the microcontroller clock and the length of the T2 period.

When selecting a microcontroller it is helpful to have a counter timer port available. The microcontroller should have provisions for software calibration. While the ADXL202/ADXL210 are highly accurate accelerometers, they have a wide tolerance for

ADXL202/ADXL210

initial offset. The easiest way to null this offset is with a calibration factor saved on the microcontroller or by a user calibration for zero *g*. In the case where the offset is calibrated during manufacture, there are several options, including external EEPROM and microcontrollers with “one-time programmable” features.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The accelerometer bandwidth selected will determine the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor and improve the resolution of the accelerometer. Resolution is dependent on both the analog filter bandwidth at X_{FILT} and Y_{FILT} and on the speed of the microcontroller counter.

The analog output of the ADXL202/ADXL210 has a typical bandwidth of 5 kHz, much higher than the duty cycle stage is capable of converting. The user must filter the signal at this point to limit aliasing errors. To minimize DCM errors the analog bandwidth should be less than 1/10 the DCM frequency. Analog bandwidth may be increased to up to 1/2 the DCM frequency in many applications. This will result in greater dynamic error generated at the DCM.

The analog bandwidth may be further decreased to reduce noise and improve resolution. The ADXL202/ADXL210 noise has the characteristics of white Gaussian noise that contributes equally at all frequencies and is described in terms of μg per root Hz; i.e., the noise is proportional to the square root of the bandwidth of the accelerometer. It is recommended that the user limit bandwidth to the lowest frequency needed by the application, to maximize the resolution and dynamic range of the accelerometer.

With the single pole roll-off characteristic, the typical noise of the ADXL202/ADXL210 is determined by the following equation:

$$\text{Noise (rms)} = \left(500 \mu\text{g}/\sqrt{\text{Hz}} \right) \times \left(\sqrt{\text{BW} \times 1.5} \right)$$

At 100 Hz the noise will be:

$$\text{Noise (rms)} = \left(500 \mu\text{g}/\sqrt{\text{Hz}} \right) \times \left(\sqrt{100 \times (1.5)} \right) = 6.12 \text{ mg}$$

Often the peak value of the noise is desired. Peak-to-peak noise can only be estimated by statistical methods. Table III is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table III. Estimation of Peak-to-Peak Noise

Nominal Peak-to-Peak Value	% of Time that Noise Will Exceed Nominal Peak-to-Peak Value
2.0 × rms	32%
4.0 × rms	4.6%
6.0 × rms	0.27%
8.0 × rms	0.006%

The peak-to-peak noise value will give the best estimate of the uncertainty in a single measurement.

Table IV gives typical noise output of the ADXL202/ADXL210 for various C_X and C_Y values.

Table IV. Filter Capacitor Selection, C_X and C_Y

Bandwidth	C_X, C_Y	rms Noise	Peak-to-Peak Noise Estimate 95% Probability (rms × 4)
10 Hz	0.47 μF	1.9 mg	7.6 mg
50 Hz	0.10 μF	4.3 mg	17.2 mg
100 Hz	0.05 μF	6.1 mg	24.4 mg
200 Hz	0.027 μF	8.7 mg	35.8 mg
500 Hz	0.01 μF	13.7 mg	54.8 mg

CHOOSING T2 AND COUNTER FREQUENCY: DESIGN TRADE-OFFS

The noise level is one determinant of accelerometer resolution. The second relates to the measurement resolution of the counter when decoding the duty cycle output.

The ADXL202/ADXL210's duty cycle converter has a resolution of approximately 14 bits; better resolution than the accelerometer itself. The actual resolution of the acceleration signal is, however, limited by the time resolution of the counting devices used to decode the duty cycle. The faster the counter clock, the higher the resolution of the duty cycle and the shorter the T2 period can be for a given resolution. The following table shows some of the trade-offs. It is important to note that this is the resolution due to the microprocessors's counter. It is probable that the accelerometer's noise floor may set the lower limit on the resolution, as discussed in the previous section.

Table V. Trade-Offs Between Microcontroller Counter Rate, T2 Period and Resolution of Duty Cycle Modulator

T2 (ms)	R _{SET} (k Ω)	ADXL202/ADXL210 Sample Rate	Counter-Clock Rate (MHz)	Counts per T2 Cycle	Counts per <i>g</i>	Resolution (mg)
1.0	124	1000	2.0	2000	250	4.0
1.0	124	1000	1.0	1000	125	8.0
1.0	124	1000	0.5	500	62.5	16.0
5.0	625	200	2.0	10000	1250	0.8
5.0	625	200	1.0	5000	625	1.6
5.0	625	200	0.5	2500	312.5	3.2
10.0	1250	100	2.0	20000	2500	0.4
10.0	1250	100	1.0	10000	1250	0.8
10.0	1250	100	0.5	5000	625	1.6

ADXL202/ADXL210

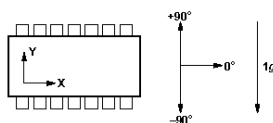
STRATEGIES FOR USING THE DUTY CYCLE OUTPUT WITH MICROCONTROLLERS

Application notes outlining various strategies for using the duty cycle output with low cost microcontrollers are available from the factory.

USING THE ADXL202/ADXL210 AS A DUAL AXIS TILT SENSOR

One of the most popular applications of the ADXL202/ADXL210 is tilt measurement. An accelerometer uses the force of gravity as an input vector to determine orientation of an object in space.

An accelerometer is most sensitive to tilt when its sensitive axis is perpendicular to the force of gravity, i.e., parallel to the earth's surface. At this orientation its sensitivity to changes in tilt is highest. When the accelerometer is oriented on axis to gravity, i.e., near its +1 g or -1 g reading, the change in output acceleration per degree of tilt is negligible. When the accelerometer is perpendicular to gravity, its output will change nearly 17.5 mg per degree of tilt, but at 45° degrees it is changing only at 12.2 mg per degree and resolution declines. The following table illustrates the changes in the X and Y axes as the device is tilted $\pm 90^\circ$ through gravity.



X AXIS ORIENTATION TO HORIZON (°)	X OUTPUT		Y OUTPUT (g)	
	X OUTPUT (g)	Δ PER DEGREE OF TILT (mg)	Y OUTPUT (g)	Δ PER DEGREE OF TILT (mg)
-90	-1.000	-0.2	0.000	17.5
-75	-0.966	4.4	0.259	16.9
-60	-0.866	8.6	0.500	15.2
-45	-0.707	12.2	0.707	12.4
-30	-0.500	15.0	0.866	8.9
-15	-0.259	16.8	0.966	4.7
0	0.000	17.5	1.000	0.2
15	0.259	16.9	0.966	-4.4
30	0.500	15.2	0.866	-8.6
45	0.707	12.4	0.707	-12.2
60	0.866	8.9	0.500	-15.0
75	0.966	4.7	0.259	-16.8
90	1.000	0.2	0.000	-17.5

Figure 14. How the X and Y Axes Respond to Changes in Tilt

A DUAL AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT

When the accelerometer is oriented so both its X and Y axes are parallel to the earth's surface it can be used as a two axis tilt sensor with a roll and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between -1 g and +1 g, the output tilt in degrees is calculated as follows:

$$\text{Pitch} = \text{ASIN}(Ax/1 \text{ g})$$

$$\text{Roll} = \text{ASIN}(Ay/1 \text{ g})$$

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than $\pm 1 \text{ g}$ due to vibration, shock or other accelerations.

MEASURING 360° OF TILT

It is possible to measure a full 360° of orientation through gravity by using two accelerometers oriented perpendicular to one another (see Figure 15). When one sensor is reading a maximum change in output per degree, the other is at its minimum.

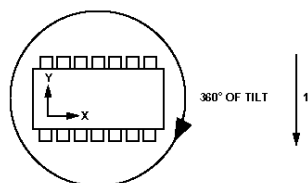


Figure 15. Using a Two-Axis Accelerometer to Measure 360° of Tilt

ADXL202/ADXL210

USING THE ANALOG OUTPUT

The ADXL202/ADXL210 was specifically designed for use with its digital outputs, but has provisions to provide analog outputs as well.

Duty Cycle Filtering

An analog output can be reconstructed by filtering the duty cycle output. This technique requires only passive components. The duty cycle period (T_2) should be set to 1 ms. An RC filter with a 3 dB point at least a factor of 10 less than the duty cycle frequency is connected to the duty cycle output. The filter resistor should be no less than 100 k Ω to prevent loading of the output stage. The analog output signal will be ratiometric to the supply voltage. The advantage of this method is an output scale factor of approximately double the analog output. Its disadvantage is that the frequency response will be lower than when using the X_{FILT} , Y_{FILT} output.

X_{FILT} , Y_{FILT} Output

The second method is to use the analog output present at the X_{FILT} and Y_{FILT} pin. Unfortunately, these pins have a 32 k Ω output impedance and are not designed to drive a load directly. An op amp follower may be required to buffer this pin. The advantage of this method is that the full 5 kHz bandwidth of the accelerometer is available to the user. A capacitor still must be added at this point for filtering. The duty cycle converter should be kept running by using $R_{SET} < 10 M\Omega$. Note that the accelerometer offset and sensitivity are ratiometric to the supply voltage. The offset and sensitivity are nominally:

$$\begin{aligned} 0g \text{ Offset} &= V_{DD}/2 && 2.5 \text{ V at } +5 \text{ V} \\ \text{ADXL202 Sensitivity} &= (60 \text{ mV} \times V_S)/g && 300 \text{ mV/g at } +5 \text{ V, } V_{DD} \\ \text{ADXL210 Sensitivity} &= (20 \text{ mV} \times V_S)/g && 100 \text{ mV/g at } +5 \text{ V, } V_{DD} \end{aligned}$$

USING THE ADXL202/ADXL210 IN VERY LOW POWER APPLICATIONS

An application note outlining low power strategies for the ADXL202/ADXL210 is available. Some key points are presented here. It is possible to reduce the ADXL202/ADXL210's average current from 0.6 mA to less than 20 μA by using the following techniques:

1. Power Cycle the accelerometer.
2. Run the accelerometer at a Lower Voltage, (Down to 3 V).

Power Cycling with an External A/D

Depending on the value of the X_{FILT} capacitor, the ADXL202/ADXL210 is capable of turning on and giving a good reading in 1.6 ms. Most microcontroller based A/Ds can acquire a reading in another 25 μs . Thus it is possible to turn on the ADXL202/ADXL210 and take a reading in <2 ms. If we assume that a 20 Hz sample rate is sufficient, the total current required to take 20 samples is $2 \text{ ms} \times 20 \text{ samples/s} \times 0.6 \text{ mA} = 24 \mu\text{A}$ average current. Running the part at 3 V will reduce the supply current from 0.6 mA to 0.4 mA, bringing the average current down to 16 μA .

The A/D should read the analog output of the ADXL202/ADXL210 at the X_{FILT} and Y_{FILT} pins. A buffer amplifier is recommended, and may be required in any case to amplify the analog output to give enough resolution with an 8-bit to 10-bit converter.

Power Cycling When Using the Digital Output

An alternative is to run the microcontroller at a higher clock rate and put it into shutdown between readings, allowing the use of the digital output. In this approach the ADXL202/ADXL210 should be set at its fastest sample rate ($T_2 = 0.5 \text{ ms}$), with a 500 Hz filter at X_{FILT} and Y_{FILT} . The concept is to acquire a reading as quickly as possible and then shut down the ADXL202/ADXL210 and the microcontroller until the next sample is needed.

In either of the above approaches, the ADXL202/ADXL210 can be turned on and off directly using a digital port pin on the microcontroller to power the accelerometer without additional components. The port should be used to switch the common pin of the accelerometer so the port pin is "pulling down."

CALIBRATING THE ADXL202/ADXL210

The initial value of the offset and scale factor for the ADXL202/ADXL210 will require calibration for applications such as tilt measurement. The ADXL202/ADXL210 architecture has been designed so that these calibrations take place in the software of the microcontroller used to decode the duty cycle signal. Calibration factors can be stored in EEPROM or determined at turn-on and saved in dynamic memory.

For low g applications, the force of gravity is the most stable, accurate and convenient acceleration reference available. A reading of the 0 g point can be determined by orientating the device parallel to the earth's surface and then reading the output.

A more accurate calibration method is to make measurements at +1 g and -1 g . The sensitivity can be determined by the two measurements.

To calibrate, the accelerometer's measurement axis is pointed directly at the earth. The 1 g reading is saved and the sensor is turned 180° to measure -1 g . Using the two readings, the sensitivity is:

$$\begin{aligned} \text{Let } A &= \text{Accelerometer output with axis oriented to } +1 g \\ \text{Let } B &= \text{Accelerometer output with axis oriented to } -1 g \text{ then:} \\ \text{Sensitivity} &= [A - B]/2 g \end{aligned}$$

For example, if the +1 g reading (A) is 55% duty cycle and the -1 g reading (B) is 32% duty cycle, then:

$$\text{Sensitivity} = [55\% - 32\%]/2 g = 11.5\%/g$$

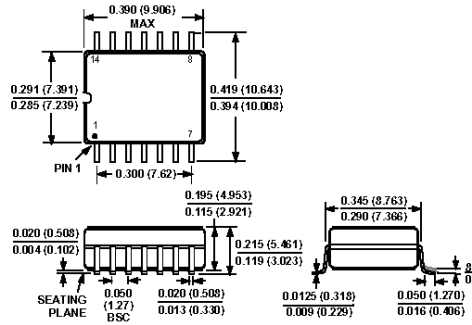
These equations apply whether the output is analog, or duty cycle.

Application notes outlining algorithms for calculating acceleration from duty cycle and automated calibration routines are available from the factory.

ADXL202/ADXL210

OUTLINE DIMENSIONS
Dimensions shown in inches and (mm).

**14-Lead CERPAK
(QC-14)**

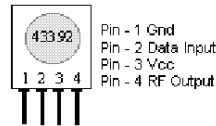


C30037 b-2-4/99

PRINTED IN U.S.A.

TWS-434 / RWS-434
<http://www.rentron.com>

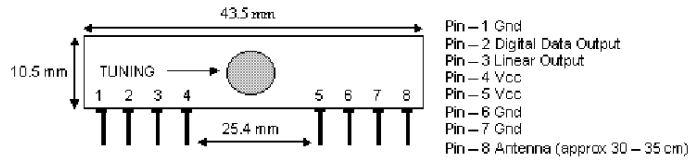
TWS-434A RF Transmitter



Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 2 - 12 VDC

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		2.0	-	12.0	V
Ip	Peak Current	2V / 12V	-	1.64 / 19.4	-	mA
Vh	Input High Voltage	Idata = 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
VI	Input Low Voltage	Idata = 0 uA (Low)	-	-	0.3	V
Fo	Operating Frequency		433.90	433.92	433.94	MHz
Tr / Tf	Modulation Rise / Fall Time	External Coding	-	-	100 / 100	uS
Po	RF Output Power – Into 50Ω	Vcc = 9 to 12 V Vcc = 5 to 6V	-	16 14	-	dBm
Dr	Data Rate	External Coding	-	2.4K	3K	Bps

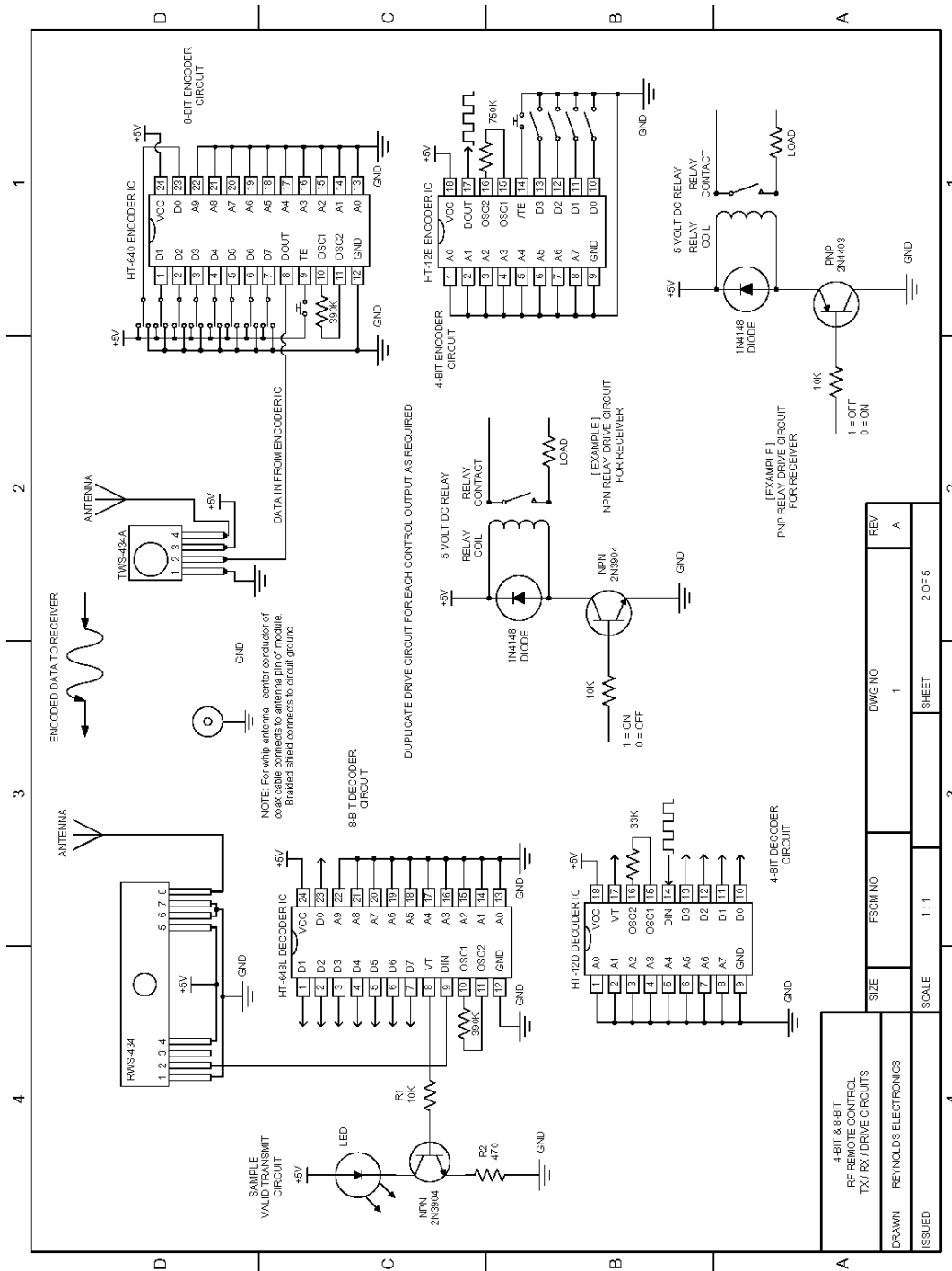
RWS-434 RF Receiver



Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 4.5 - 5.5 VDC
Output: Digital & Linear

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		4.5	5	5.5	V
It	Operating Current		-	3.5	4.5	mA
	Channel Width	+ / - 500				kHz
Rd	Data Rate				3k	Bps
Vdat	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Reynolds Electronics
 3101 Eastridge Lane
 Canon City, CO. 81212
 Tel: (719) 269-3469 Fax: (719) 276-2853
 Web: <http://www.rentron.com>



SPECIFICATION FOR LCD MODULE

Model No. TM162AAA6-1

Prepared by:	Date:
Checked by :	Date:
Verified by :	Date:
Approved by:	Date:

TIANMA MICROELECTRONICS CO., LTD

REVISION RECORD

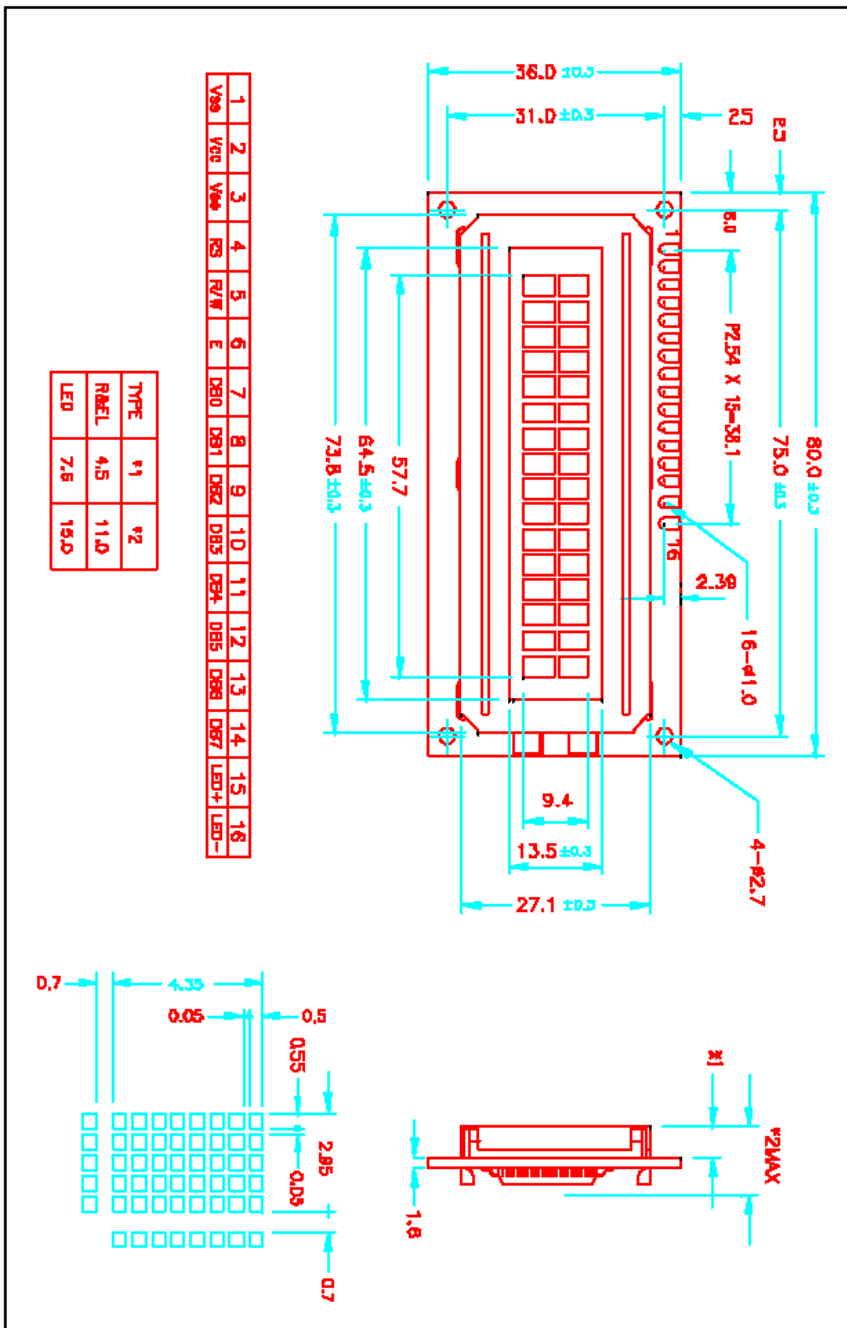
Date	Ref. Page	Revision No.	Revision Items	Check & Approval

1. General Specifications:

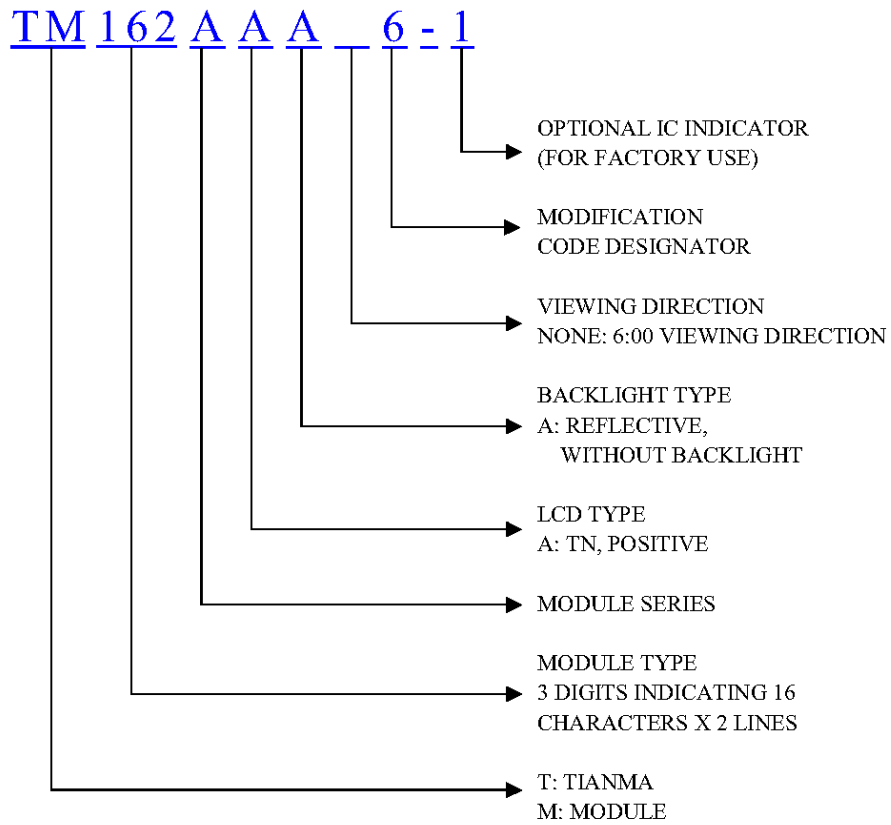
- 1.1 Display type: TN
- 1.2 Display color*:
 - Display color: Blue-Black
 - Background: White
- 1.3 Polarizer mode: Reflective/Positive
- 1.4 Viewing Angle: 6:00
- 1.5 Driving Method: 1/16 Duty 1/5 Bias
- 1.6 Backlight: None
- 1.7 Display Fonts: 5 x 7 dots + Cursor (1 Character)
- 1.8 Controller: HCD66701A00
- 1.9 Data Transfer: 8 Bit Parallel
- 1.10 Operating Temperature: 0----+50°C
 - Storage Temperature: -20----+60°C
- 1.11 Outline Dimensions: Refer to outline drawing on next page
- 1.12 Dot Matrix: 16 Characters X 2 Lines
- 1.13 Dot Size: 0.55X0.50(mm)
- 1.14 Dot Pitch: 0.60X0.55 (mm)
- 1.15 Weight: 20g
- 1.16 PCB edition: TM162AD P1-2

* Color tone is slightly changed by temperature and driving voltage.

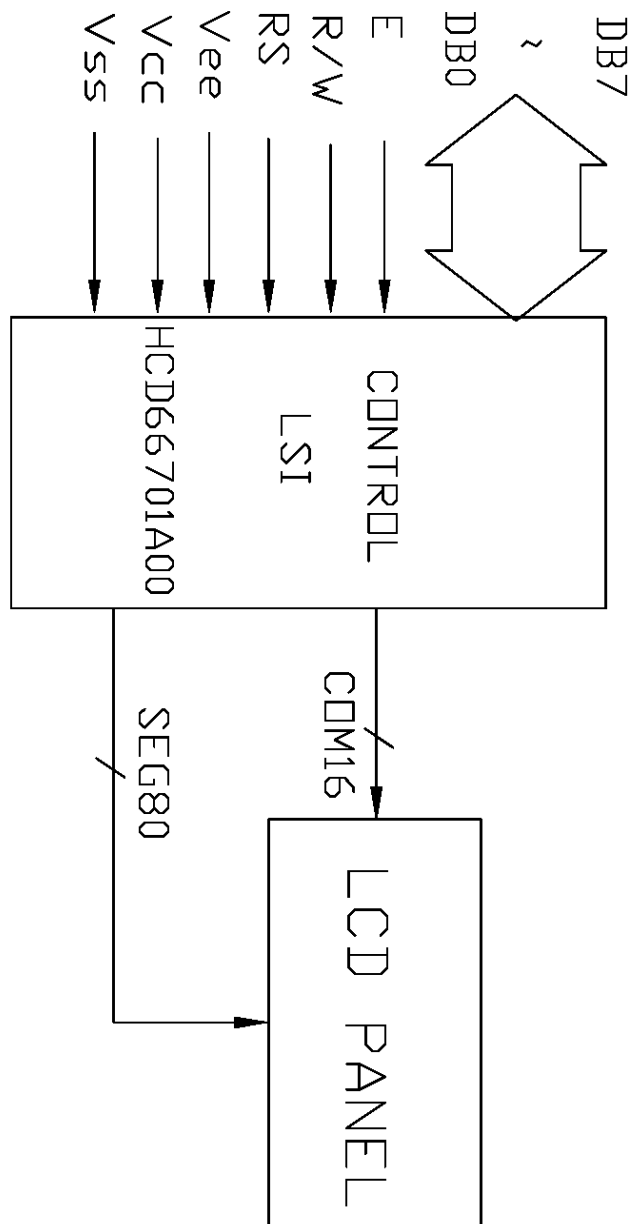
2. Outline Drawing



3. LCD Module Part Numbering System



4. Circuit Block Diagram



5. Absolute Maximum Ratings

Item	Symbol	Min.	Max.	Unit	Remark
Power Supply Voltage	$V_{DD}-V_{SS}$	-0.3	7.0	V	
LCD Driving Voltage	V_{LCD}	-0.3	13.0		
Operating Temperature Range	T_{OP}	0	+50	°C	No Condensation
Storage Temperature Range	T_{ST}	-20	+60		

6. Electrical Specifications and Instruction Code

6.1 Electrical characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	
Supply Voltage (Logic)	$V_{DD}-V_{SS}$	4.5	5.0	5.5	V	
Supply Voltage (LCD Drive)	V_{LCD}	-	4.7	-	V	
Input Signal Voltage	High	V_{IH} ($V_{DD}=5.0$)	$0.7V_{DD}$	-	$V_{DD}+0.3$	V
	Low	V_{IL} ($V_{DD}=5.0$)	-0.3	-	$0.2 V_{DD}$	V
Supply current (Logic)	I_{DD} ($V_{DD}-V_{SS}=5.0$)	-	1.5	2.0	mA	
Supply current (LCD Drive)	I_{EE}	-	0.40	0.6	mA	

6.2 Interface Signals

Pin No.	Symbol	Level	Description
1	Vss	0V	Ground
2	Vcc	5.0V	Power supply voltage for logic and LCD(+)
3	Vee	0.3V	Power supply voltage for LCD(-)
4	RS	H/L	Selects registers
5	R/W	H/L	Selects read or write
6	E	H/L	Starts data read/write
7	DB0	H/L	Data bit0
8	DB1	H/L	Data bit1
9	DB2	H/L	Data bit2
10	DB3	H/L	Data bit3
11	DB4	H/L	Data bit4
12	DB5	H/L	Data bit5
13	DB6	H/L	Data bit6
14	DB7	H/L	Data bit7
15	NC	-	No signal
16	NC	-	No signal

6.3 Interface Timing Chart

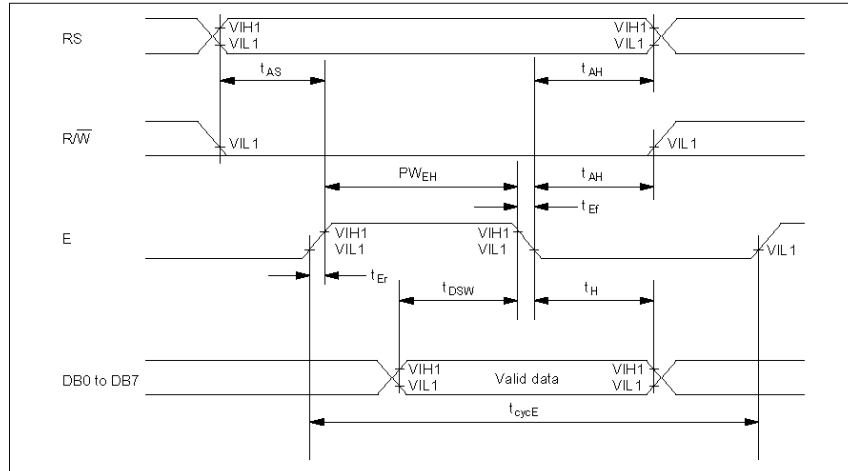
AC Characteristics ($V_{CC} = 2.7$ to $5.5V$, $T_a = -30$ to $+75^\circ C^{*3}$)**Write Operation**

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, $R\bar{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

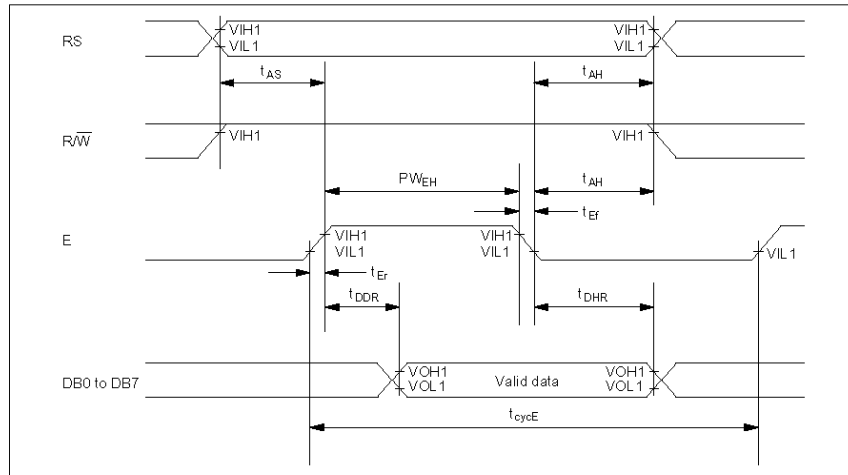
Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, $R\bar{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	350		
Data hold time	t_{DHR}	10	—	—		

Timing Characteristics



Write Operation



Read Operation

6.4 Instruction Code

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 320 kHz)
	RS	R/ \bar{W}	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	1.28 ms
Return home	0	0	0	0	0	0	0	0	0	1	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.28 ms
Entry mode set	0	0	0	0	0	0	0	1	\bar{V}	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	31 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	31 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	31 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (L), and character font (F).	31 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	31 μ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	31 μ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 320 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	31 μ s $t_{ADD} = 4.7 \mu$ s*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	31 μ s $t_{ADD} = 4.7 \mu$ s*
VD = 1: Increment VD = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 x 10 dots, F = 0: 5 x 7 dots BF = 1: Internally operating BF = 0: Instructions acceptable											DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_p or f_{osc} is 270 kHz, 31μ s $\times \frac{320}{270} = 37 \mu$ s

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

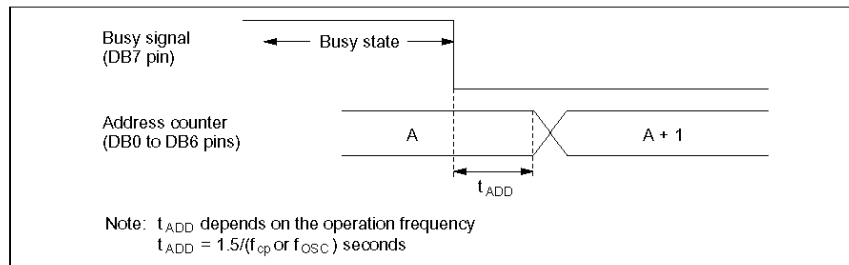


Figure 10 Address Counter Update

6.5 Character generator ROM(HCD66701A00)

Lower 4 Bits \ Upper 4 Bits	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	C.G RAM (1)		0	1	2	3	4	5	6	7	8	9	A
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	△	ä	g
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	ß	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ハ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	1	σ	Ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	ア	キ	ヌ	ラ	q	π
xxxx1000	(1)	(8	H	X	h	x	イ	ク	ネ	リ	∫	×
xxxx1001	(2))	9	I	Y	i	y	ウ	ツ	ル		∫	∫
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ン	ク	j	≠
xxxx1011	(4)	+	;	K	[k	[オ	サ	ヒ	ロ	*	≠
xxxx1100	(5)	,	<	L	¥	l	l	ャ	シ	フ	ワ	Φ	≠
xxxx1101	(6)	-	=	M]m]m]m	ユ	ズ	ハ	ン	≠	÷
xxxx1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ッ	≠	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	マ	ö	■

Note: The user can specify any pattern for character-generator RAM.



PIC16F84A
Data Sheet

18-pin Enhanced FLASH/EEPROM
8-bit Microcontroller

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks


The Microchip name and logo, the Microchip logo, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, KEELoQ, SEEVAL, MPLAB and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, ICSP, In-Circuit Serial Programming, Filter-Lab, MXDEV, microID, FlexROM, fuzzyLAB, MPASM, MPLINK, MPLIB, PICC, PICDEM, PICDEM.net, ICEPIC, Migratable Memory, FanSense, ECONOMONITOR, Select Mode and microPort are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2001, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



PIC16F84A

18-pin Enhanced FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
 - External RB0/INT pin
 - TMR0 timer overflow
 - PORTB<7:4> interrupt-on-change
 - Data EEPROM write complete

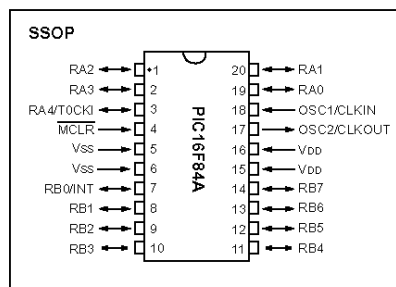
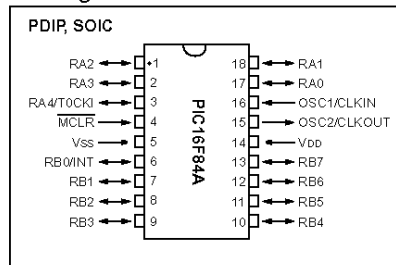
Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
 - 25 mA sink max. per pin
 - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

Pin Diagrams



CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
 - Commercial: 2.0V to 5.5V
 - Industrial: 2.0V to 5.5V
- Low power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 15 µA typical @ 2V, 32 kHz
 - < 0.5 µA typical standby current @ 2V

PIC16F84A

Table of Contents

1.0 Device Overview	3
2.0 Memory Organization	5
3.0 Data EEPROM Memory	13
4.0 I/O Ports	15
5.0 Timer0 Module	19
6.0 Special Features of the CPU	21
7.0 Instruction Set Summary	35
8.0 Development Support	43
9.0 Electrical Characteristics	49
10.0 DC/AC Characteristic Graphs	61
11.0 Packaging Information	71
Appendix A: Revision History	75
Appendix B: Conversion Considerations	78
Appendix C: Migration from Baseline to Mid-Range Devices	78
Index	79
On-Line Support	83
Reader Response	84
PIC16F84A Product Identification System	85

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@mail.microchip.com or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on all of our products.

PIC16F84A

1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F84A device. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F84A belongs to the mid-range family of the PICmicro® microcontroller devices. A block diagram of the device is shown in Figure 1-1.

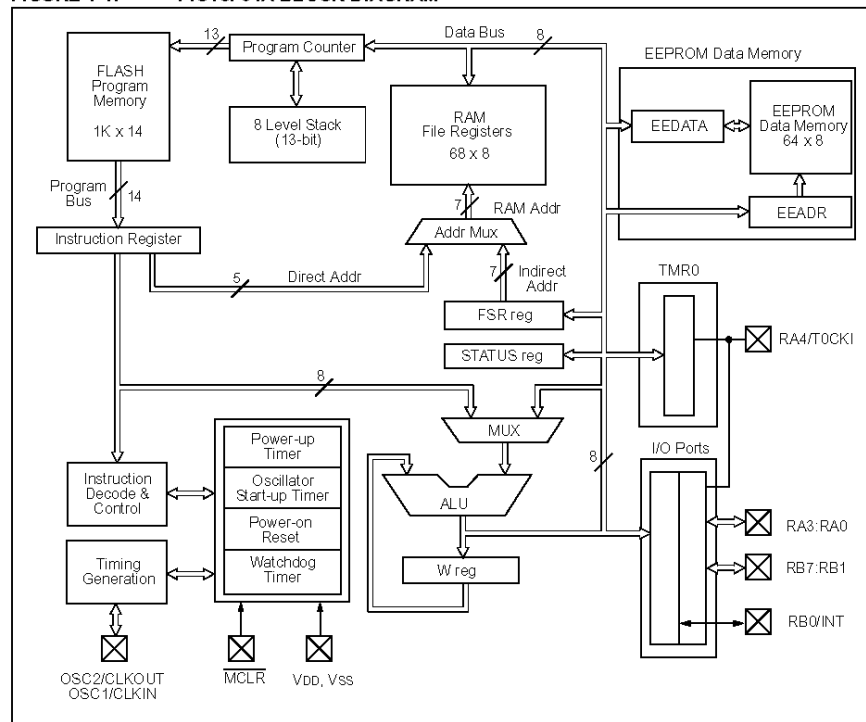
The program memory contains 1K words, which translates to 1024 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 68 bytes. Data EEPROM is 64 bytes.

There are also 13 I/O pins that are user-configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input

Table 1-1 details the pinout of the device with descriptions and details for each pin.

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



PIC16F84A

TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	17	19	I/O	TTL	PORTA is a bi-directional I/O port. Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	
RB0/INT	6	6	7	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0/INT can also be selected as an external interrupt pin. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin. Serial programming clock. Interrupt-on-change pin. Serial programming data.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	
RB5	11	11	12	I/O	TTL	
RB6	12	12	13	I/O	TTL/ST ⁽²⁾	
RB7	13	13	14	I/O	TTL/ST ⁽²⁾	
V _{SS}	5	5	5,6	P	—	Ground reference for logic and I/O pins.
V _{DD}	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = Output I/O = Input/Output P = Power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F84A

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F84A. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh. More details on the EEPROM memory can be found in Section 3.0.

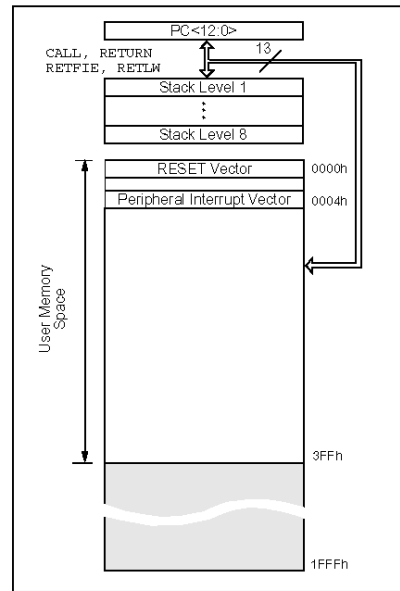
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84A, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 2-1). Accessing a location above the physically implemented address will cause a wraparound. For example, for locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h, the instruction will be the same.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK - PIC16F84A



M68HC11 E SERIES HCMOS MICROCONTROLLER UNIT

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. MOTOROLA and the Motorola logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© MOTOROLA, INC., 1993, 1996

SECTION 1 INTRODUCTION

This document contains a detailed description of the M68HC11 E series of 8-bit microcontroller units (MCUs). These MCUs all combine the M68HC11 CPU with high-performance, on-chip peripherals. The E series is comprised of many devices with various configurations of RAM, ROM or EPROM, and EEPROM. Several low-voltage devices are also available. With the exception of a few minor differences, the operation of all E-series MCUs is identical. A fully static design and high-density complementary metal-oxide semiconductor (HCMOS) fabrication process allow E-series devices to operate at frequencies from 3 MHz to dc, with very low power consumption.

1.1 Features

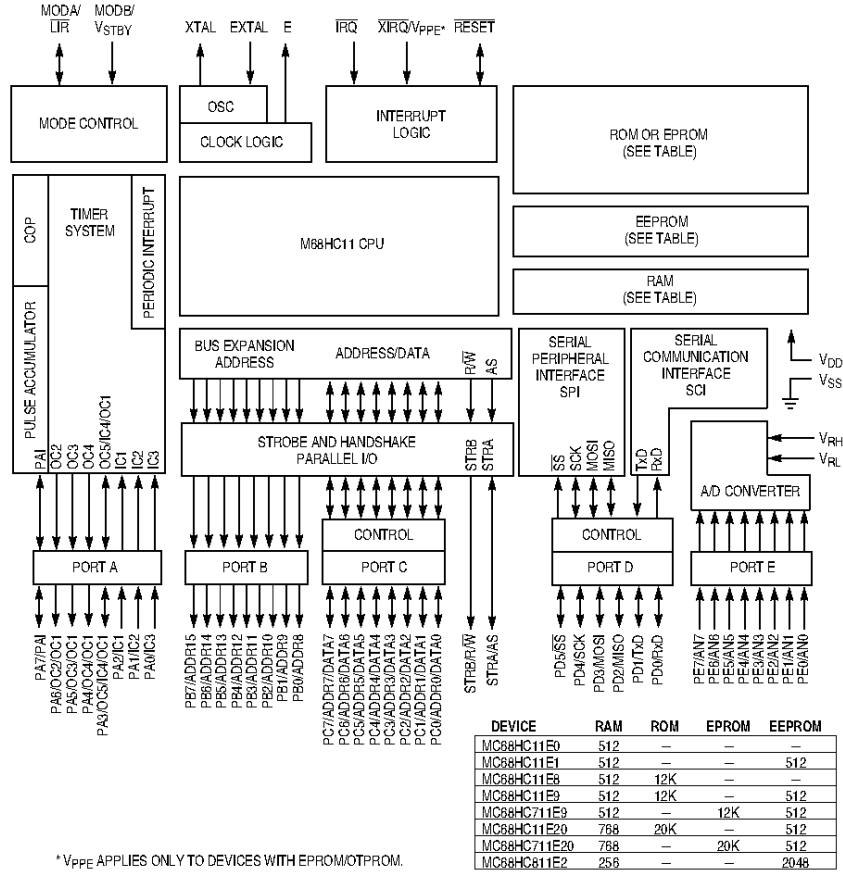
- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- Low-Voltage Devices Available (3.0 – 5.5 Vdc or 2.7 – 5.5 Vdc)
- 0, 256, 512, or 768 Bytes of On-Chip RAM, Data Retained During Standby
- 0, 12, or 20 Kbytes of On-Chip ROM or EPROM
- 0, 512, or 2048 Bytes of On-Chip EEPROM with Block Protect for Security
 - 2048 bytes of EEPROM with selectable base address in MC68HC811E2
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
 - Additional Baud Rates Available on MC68HC(7)11E20
- Synchronous Serial Peripheral Interface (SPI)
- 8-Channel 8-Bit Analog-to-Digital (A/D) Converter
- 16-Bit Timer System
 - Three Input Capture (IC) Channels
 - Four Output Compare (OC) Channels
 - One Additional Channel, Selectable as Fourth IC or Fifth OC
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System
- 38 General-Purpose Input/Output (I/O) Pins
 - 16 Bidirectional I/O Pins
 - 11 Input-Only Pins
 - 11 Output-Only Pins
- Several Packaging Options
 - 52-Pin Plastic Leaded Chip Carrier (PLCC)
 - 52-Pin Windowed Ceramic Leaded Chip Carrier (CLCC)
 - 52-Pin Plastic Thin Quad Flat Pack, 10 mm X 10 mm (TQFP)
 - 64-Pin Plastic Quad Flat Pack (QFP)
 - 48-Pin Plastic Dual In-Line Package (DIP), MC68HC811E2 only
 - 56-Pin Plastic Dual In-Line Package, .070" Lead Spacing (SDIP)

1

1.2 Structure

Below is the functional diagram of the E-series MCUs. Differences among devices are noted in the table.

1

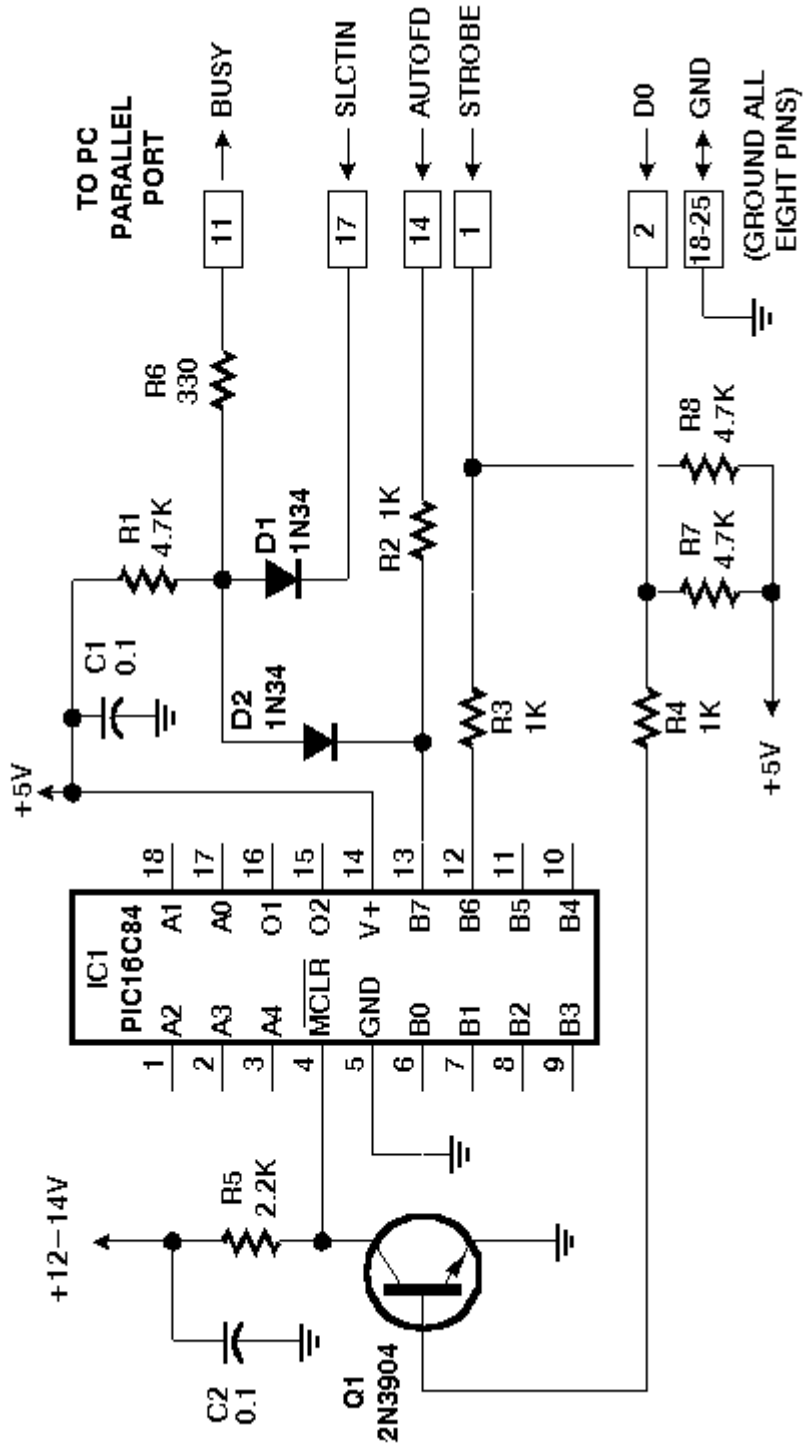


E-SERIES BLOCK

Figure 1-1 M68HC11 E-Series Block Diagram

**REVISED NOPPP – MICHAEL A. COVINGTON, OCT. 31, 1998
FOR GREATER COMPATIBILITY WITH SOME PC PARALLEL PORTS.**

START SOFTWARE FIRST; APPLY POWER AND INSERT PIC WHEN TOLD TO DO SO.



REFERENCIAS BIBLIOGRAFICAS

- [3.1] Pagina 8, Sensores y acondicionadores de señal, Ramón Pallás Areny
- [3.2] Pagina 340, Instrumentación Electrónica Moderna y Técnicas de Medición.
William D Cooper, Albert D. Helfrick
- [3.3] Capítulos 1.3 a 1.6 , Sensores y acondicionadores de señal, Ramón Pallás Areny
Capítulos 1 a 2, Instrumentación Electrónica Moderna y Técnicas de Medición.
William D Cooper, Albert D. Helfrick
- [3.4] Pagina 32, Principios de física, Beltrán V., Braun E.
- [3.5] Paginas 350-351, Instrumentación Electrónica Moderna y Técnicas de Medición.
William D Cooper, Albert D. Helfrick
- [3.6] Pagina 351-352, Instrumentación Electrónica Moderna y Técnicas de Medición
William D Cooper, Albert D. Helfrick
- [4.1] Modelo OSI (Open System Interconnection)
- <http://www.iso.org>
- http://www.pchardware.org/redes/redes_osi.php
- http://elsitiodetelecomunicaciones.iespana.es/elsitiodetelecomunicaciones/modelo_osi.htm
- http://rho.frm.utn.edu.ar/comunicaciones/modelo_osi.html
- [5.1] Formula General de Resistencia, Capitulo 13.
Mecánica de Fluidos y Maquinas Hidráulicas
Mataix, Claudio
- [5.2] Capitulo 7 a 9, Mecánica de Fluidos y Maquinas Hidráulicas
Mataix, Claudio
- [5.3] Capitulo VII Interconexión de Redes de Telecomunicaciones,
Reglamento de Telecomunicaciones de México.
http://www.cofetel.gob.mx/html/9_publica/5_reglamento/reglamtelecom/reglatel_07.html#Radio
Cuadro Nacional de Atribución de Frecuencias
<http://www.radionet.udg.mx/~ararm/frecs.htm>
- [6.1] Ver Hojas de especificaciones del Acelerómetro ADXL202
- [6.2] Definición UART
- http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/
- <http://www.wireless-open.com.ar/block.html>

11. Referencias Bibliográficas

[6.3] Sección 7. HC11 Technical Data Manual

[6.4] Paginas 59 a 61, Principios de física, Beltrán V., Braun E

[6.5] Pagina 60, Principios de física, Beltrán V., Braun E

BIBLIOGRAFÍA

- MC68HC11 Reference Manual
Digital DNA, Motorola.
- PICmicro™ Mid-Range MCU Family Reference Manual
Microchip.
- Diseño Digital
M. Morris Mano
Prentice Hall
- Principios de física
Beltrán V., Braun E.
Trillas
- Receptores para Sistemas de Radiocomunicación
Golovin Oleg, Jardón Hildeberto
Alfaomega
- Sensors and Transducers
Sinclair Ian R.
BH NEWNES
- Teoría del Buque
Antonio Bonilla de la corte
Grafias Bouzas, S.L.(vigo) España
- Naval Architecture
Cecil H. Peabody
John Wiley & Sons. Inc
- Sensors and transducers aguide for technicians
Ian R. Sinclair
B.H. NEWNES
- Sensores y acondicionadores de señal
Ramon Pallás Areny
Marcombo Boixareu Editores
- Instrumentacion Electronica Moderna y Tecnicas de Medicion
William D Cooper, Albert D. Helfrick
Prentice Hall
- Mecanica de Fluidos y Maquinas Hidraulicas
Mataix, Claudio
Edit Harla

Links relacionados con el programador del PIC

- <http://www.covingtoninnovations.com/noppp/>

Links relacionados con el lenguaje de programación para el PIC (JAL)

- <http://www.voti.nl/jal>
- <http://www.piclist.org/techref/piclist/jal>
- <http://www.voti.nl/jal/index.html>
- <http://www.portlandrobotics.org/links/software.html>

12. Bibliografía

Sitio WEB del modulo de comunicaciones

- <http://www.rentron.com>

Sitio WEB de Analog Devices fabricante del Acelerometro

- <http://www.analogdevices.com>
- http://www.analog.com/Analog_Root/productPage/productHome/0%2C2121%2CADXL202%2C00.html

Sitio WEB de Motorola

- <http://www.motorola.com>
- <http://e-www.motorola.com> (semiconductores)