AUTOMATIZACIÓN DE UNA ESTACION DE SERVICIO DE COMBUSTIBLE

ARBEY HERNANDEZ GRANADOS
LEONARDO GIL AREIZA

Proyecto presentado como requisito de cumplimiento de la **práctica empresarial**

Director
ELIECER OLIVEROS ARIAS
Ingeniero Mecánico

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
DIVISIÒN DE CIENCIAS NATURALES E INGENIERIA
INGENIERIA MECATRÓNICA
BUCARAMANGA
2003

NOTA DE ACEPTACION

El proyecto titulado "Automatización de una estación de servicio de combustible". Presentado por Arbey Hernandez Granados y Leonardo Gil Areiza, como requisito para el cumplimiento de la práctica empresarial.

_____

Presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Bucaramanga, 28 de Mayo del 2003

CONTENIDO

INTRODUCCIÓN

El mundo ha cambiado mucho en los últimos cien años y es probable que cambié aún más en este siglo. Algunos preferirían detener tales cambios y retornar a la que consideran una edad más pura y simple. Pero, como muestra la historia, el pasado no fue tan maravilloso. Para la mayoría de la población era desagradable, brutal y breve.

En cualquier caso, no se pueden olvidar los conocimientos y las técnicas adquiridas, ni impedir los ulteriores progresos. Aunque se suspendiera toda la financiación oficial de las investigaciones, la fuerza de la competición determinaría todavía progresos tecnológicos.

Durante la última década, el crecimiento de las aplicaciones basadas en la adquisición de datos ha sido espectacular, tanto como el desarrollo tecnológico de las mismas.

Para el desarrollo de esta investigación se decidió implementar una comunicación via MODEM basada en un microcontrolador y con el fin de reducir costos y apoyar la industria Colombiana. Se consideró importante diseñar y construir un sistema de desarrollo para microcontroladores Motorota porque permiten que los usuarios tengan rápidamente un laboratorio donde puedan desarrollar sus aplicaciones en un menor tiempo y con una mayor confiabilidad.

Además con este proyecto de practica se desea impulsar y fortalecer el desarrollo regional y facilitar la creación de nuevos productos y servicios, fortaleciendo los vínculos entre la universidad autónoma de Santander y la empresa, en este caso INCATEL.

Este libro trata en su primer capitulo se describe profundamente el microcontrolador usado para la implementación en este caso (automatización de un surtidor de gasolina).

En el segundo capitulo se describen las diferentes partes que componen el hardware de la aplicación hecho con Motorola, así como el teclado, el LCD el reloj en tiempo real (DS1307)

En el tercer capitulo se presenta la descripción del componente que es el encargado de la comunicación del MODEM y se da un descripción de este.

En el anexo se encuentra la información que explica como implemento el programa en MOTOROLA y el programa de manejo de la aplicación; también se muestra otras labores hechas en la empresa.

MICROCONTROLADOR MOTOROLA MC68HC908GP32

Este microcontrolador pertenece a la familia *HC08* de Motorola, cuyas principales características son:

- Modelo de programación HC05 mejorado, con registro de 16 bits.
- Control de bucles optimizado.
- Set de instrucciones muy amplias y varios modos de direccionamiento.
- 16 modos de direccionamiento.
- Registro de 16 bits y stack pointer manipulable por el usuario.
- Instrucciones de transferencia de datos de memoria a memoria.
- Instrucciones de multiplicación rápida de 8x8.
- Instrucciones de división rápida de 16/8.
- Instrucciones BCD (Binary Coded Decimal).
- Optimización para aplicaciones de control.
- Fácil soporte de lenguajes de alto nivel como el C.

Las características principales del microcontrolador *MC68HC908GP32* son:

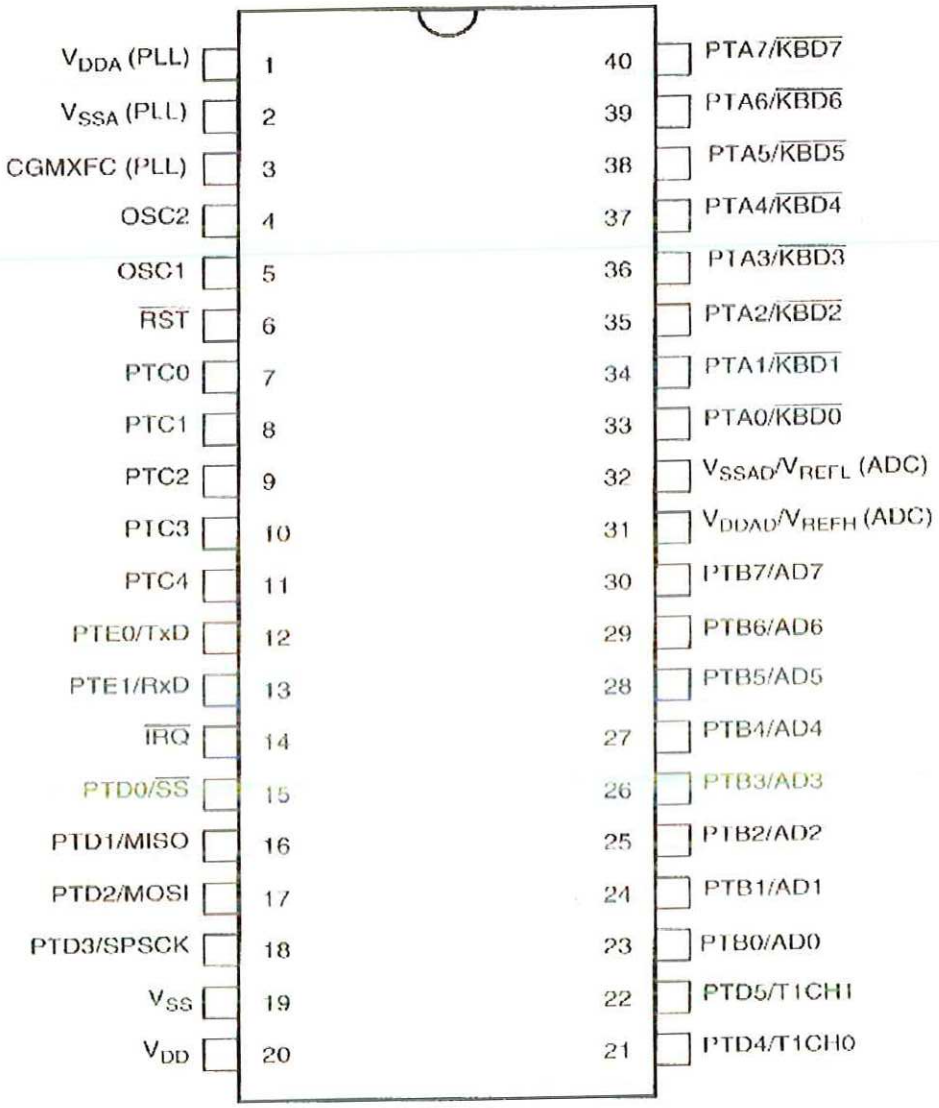
- CPU 08 de 8 bits.
- Operación interna a 8 MHz.
- Rango de operación desde 3V.
- LVI: protección de bajo voltaje.
- Fuentes de interrupciones totalmente vectorizadas.
- Modos de bajo consumo (STOP y WAIT).
- Software 100% compatible con la familia 05.
- Arquitectura de alto rendimiento M68HC08 optimizada para compiladores C.
- Código de seguridad para la lectura y programación de la memoria FLASH.
- Firmware On-chip para la programación desde PC.

- Programable en el circuito.

Sistemas de protección:

- "Watch Dog" opcional (Computer Operating Properly (COP) reset).
- Detección de baja tensión con reset opcional.
- Detección de código ilegal con reset.
- Detección de direccionamiento ilegal con reset.
- Diseño de bajo consumo, completamente estático y varios modos de operación:

    - 32 Kbytes de memoria FLASH programable en circuito.
    - 512 bytes de memoria RAM.
    - Módulo de interfaz serie asíncrono (SPI).
    - Módulo de interfaz serie síncrono (SCI).
    - Dos temporizadores de 2 canales de 16 bits (TIM1 y TIM2) con captura de entrada seleccionable, comparadores y capacidad de PWM en cada canal.
    - 8 canales para conversión AD por aproximaciones sucesivas de 8 bits.
    - Módulo generador de reloj con PLL "On-chip".
    - Hasta 33 pines de entradas / salidas de propósito general.
    - Pull - ups seleccionables en los puertos A, C, y D. La selección puede ser de forma individual, por bit.



Pins not available on 40-pin package	Internal connection
PTC5	Connected to ground
PTC6	Connected to ground
PTD6/T2CH0	Unconnected
PTD7/T2CH1	Unconnected

Figura 1. Configuración del MC68HC908GP32

DISPOSITIVOS ADICIONALES

2.1 MODULO LCD (LIQUID CRISTAL DYSPLAY)

En la actualidad los módulos LCD existen una gran variedad de versiones clasificadas en dos grupos. El primer grupo esta referido a los módulos LCD de caracteres (solamente se podrán presentar caracteres y símbolos especiales en las líneas predefinidas en el modulo LCD) y el segundo grupo esta referido a los módulos LCD matriciales (Se podrán presentar caracteres, símbolos especiales y gráficos). Los módulos LCD varían su tamaño físico dependiendo de la marca; por lo tanto en la actualidad no existe un tamaño estándar para los módulos LCD.

Para el caso de la empresa VARITRONIX especializada en la fabricación de LCD, existen configuraciones mínimas desde una línea con un mínimo de ocho caracteres y por el contrario, existen configuraciones desde 4 líneas hasta 40 caracteres por cada línea. La siguiente imagen muestra las dimensiones de una configuración típica de un modulo LCD de dos líneas por 16 caracteres por cada línea incluyendo los detalles de la matriz de como esta conformado un carácter.

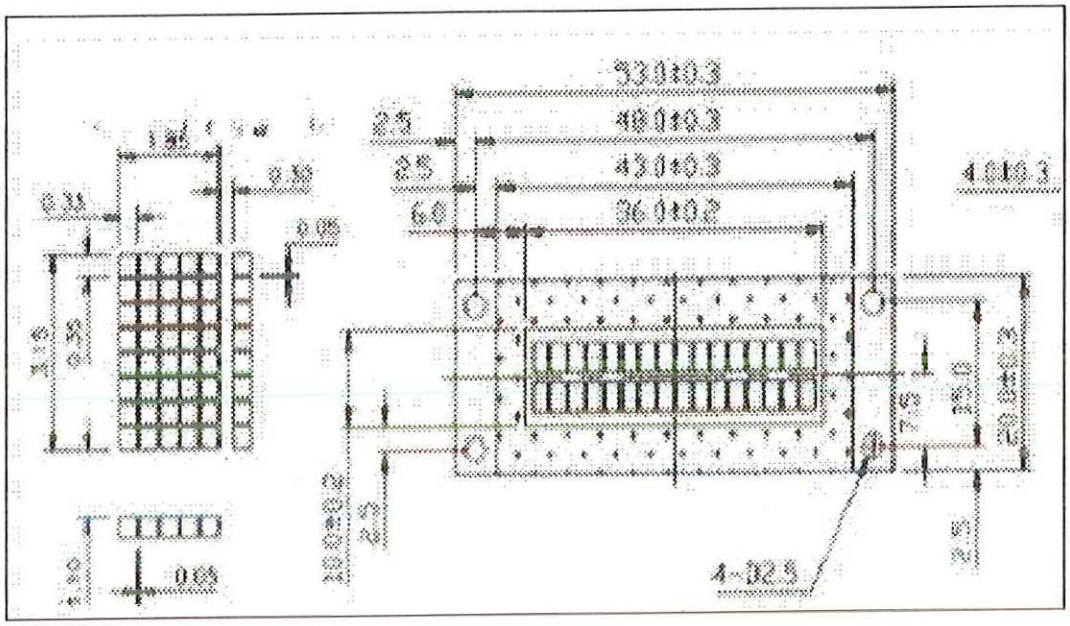


Figura 2. Estructura interna del LCD.

Otro patrón importante es el tamaño de los caracteres donde las dimensiones de la matriz que forma los caracteres tienen longitudes diferentes. La siguiente tabla muestra la matriz utilizada para poder representar un símbolo o un carácter alfa numérico en un modulo LCD. Esta matriz define algunos aspectos importantes del carácter o el símbolo que están mostrando. Los aspectos que define esta matriz son:

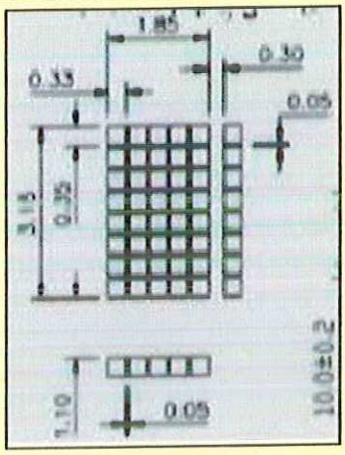
Matriz de punto para un solo carácter en un modulo LCD	Aspectos importantes que define la matriz de puntos para un solo caracter en un modulo LCD
	1-. Altura del carácter definida por dos variables: Alto de cada punto que conforma la matriz y longitud de separación entre cada punto que conforma la matriz.
	2-. Ancho del Carácter definido por dos variables: Ancho de cada punto que conforma la matriz y longitud de separación entre cada punto que conforma la matriz.
	3-. Calidad gráfica del carácter (A mayor cantidad de puntos dentro de la matriz, mayor será la calidad visual del carácter presentado por el modulo LCD.

Tabla 1. Aspectos importantes de la matriz de puntos en el LCD.

La siguiente figura representa un modulo LCD del tipo matricial y que tienen encendida la luz posterior (Back Light).

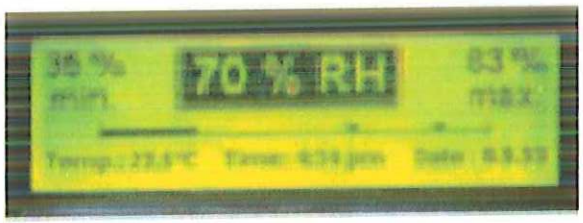


Figura 3. LCD con backlight.

Ahora la tecnología esta disponible en color para los módulos LCD desde 4 colores hasta los 256 y las combinaciones de ellos.

2.1.1. Características

Los pines de conexión de un modulo LCD han sido estandarizados por el cual en la mayoría de ellos son exactamente iguales siempre y cuando la línea de caracteres no sobrepase los ochenta caracteres por línea. En el caso de que esto suceda, localice la hoja de características del fabricante. Por otro lado es de suma importancia localizar exactamente cual es el Pin Numero 1 ya que en algunos módulos se encuentra hacia la izquierda y en otros módulos se encuentra a la derecha. En caso de no estar seguro de la asignación de los pines, localice la hoja de características del fabricante.

Pin N-.	Sismología	Nivel	I/O	Función
1	VSS	-	-	0 Vlts. Tierra (GND).
2	VCC	-	-	+ 5 Vlts. DC.
3	Vee = Vc	-	-	Ajuste del Contraste.
4	RS	0/1	I	0= Escribir en el modulo LCD. 1= Leer del modulo LCD
5	R/W	0/1	I	0= Entrada de una Instrucción. 1= Entrada de un dato.
6	E	1	I	Habilitación del modulo LCD
7	DB0	0/1	I/O	BUS DE DATO LINEA 1 (LSB).
8	DB1	0/1	I/O	BUS DE DATO LINEA 2
9	DB2	0/1	I/O	BUS DE DATO LINEA 3
10	DB3	0/1	I/O	BUS DE DATO LINEA 4
11	DB4	0/1	I/O	BUS DE DATO LINEA 5
12	DB5	0/1	I/O	BUS DE DATO LINEA 6
13	DB6	0/1	I/O	BUS DE DATO LINEA 7
14	DB7	0/1	I/O	BUS DE DATO LINEA 8 (MSB).
15	A	-	-	LED (+) Back Light
16	K	-	-	LED (-) Back Light.

Tabla 2. Descripción de los pines del LCD

2.1.2 Tiempos mínimos requeridos para que una instrucción o un dato puedan ser ejecutados.

Los Pines de control (E, RS y R/W) están estrechamente relacionados ya que por medio de ellos podemos especificar si queremos ejecutar una instrucción o leer / escribir un dato en la pantalla o la memoria RAM; sin embargo existe una condición importante que deberá tomarse en cuenta referida directamente al tiempo necesario que se necesita para cambiar de un estado a otro en los pines de control. (E, RS y R/W). En el caso de que este tiempo sea mas pequeño que el tiempo mínimo requerido, entonces el modulo LCD no tendrá el tiempo suficiente para responder a las instrucciones solicitadas por el usuario y por consecuencia se perderán los datos o instrucciones según sea el caso.

Diagrama de tiempo para una Instrucción:

Para enviarle una instrucción al modulo, primero hay que colocar la instrucción en el bus de datos (Pines del 7 al 14). Una vez que este presente la instrucción en el bus de datos se procede a ejecutar el diagrama de tiempo requerido para una instrucción en los pines de control.

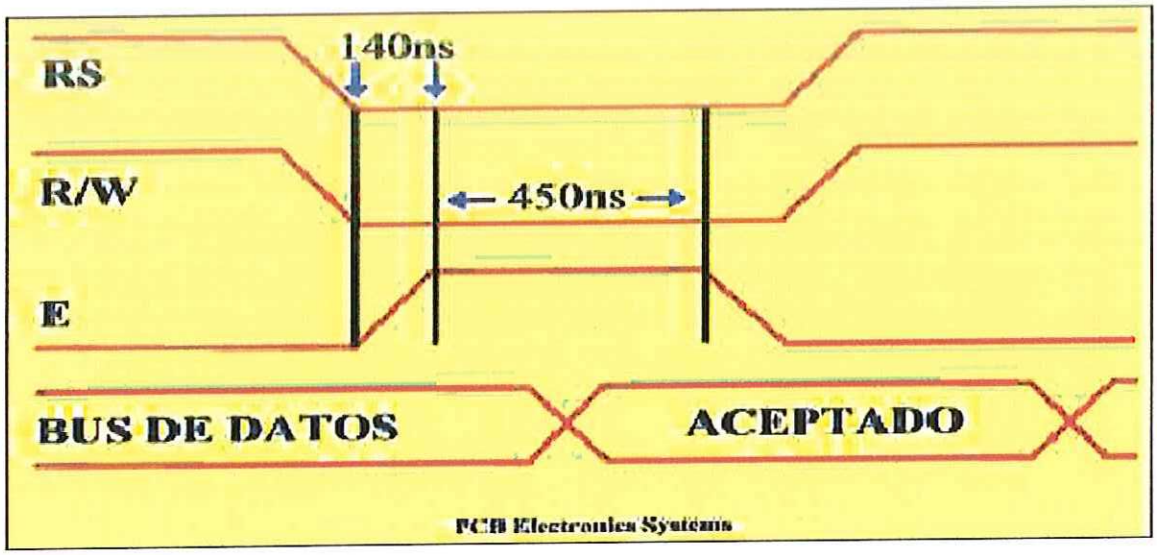


Figura 4. Diagrama de tiempo para una instrucción en el LCD

2.1.3. Diagrama de tiempo para escribir un Dato:

Para escribir un dato en el modulo LCD, primero hay que colocar el dato en el bus (Pines del 7 al 14). Una vez que este presente el dato en el bus se procede a ejecutar el diagrama de tiempo requerido para escribir un dato en los pines de control.

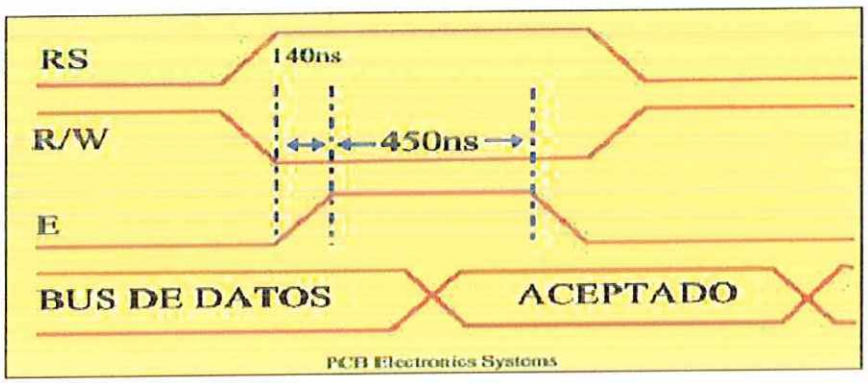


Figura 5. Diagrama de tiempo para escribir un dato en el LCD

Diagrama de tiempo para leer un Dato:

Para leer un dato de la pantalla o la memoria RAM en el modulo LCD, los pines de control deberán estar colocados como sigue: RS = 1, R/W = 1 y E = 0; Una vez colocados los pines con las tensiones mencionadas, proceda a cambiar el estado de E = 1.

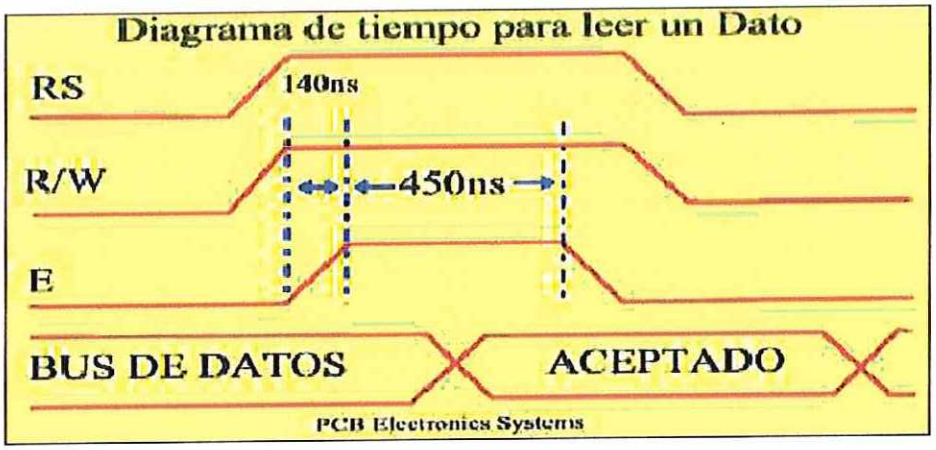


Figura 6. Diagrama de tiempo para leer un dato en el LCD

2.1.4. Bus de Datos de 4 y 8 Bits de Longitud:

El Bus de datos de un modulo LCD puede ser configurado para trabajar con 4 Bits y con 8 Bits. Para los diseños electrónicos que están limitados por la cantidad de líneas utilizadas en el Bus de datos, podrán utilizar un bus de datos con una longitud de 4 Bits; sin embargo si este no fuera su caso, podrá utilizar el bus de datos completo de 8 Bits. Las señales de control (RS - R/W - E) y los diagramas de tiempo explicados anteriormente, trabajan igual sea para un bus de datos de 4 Bits o de 8 Bits.

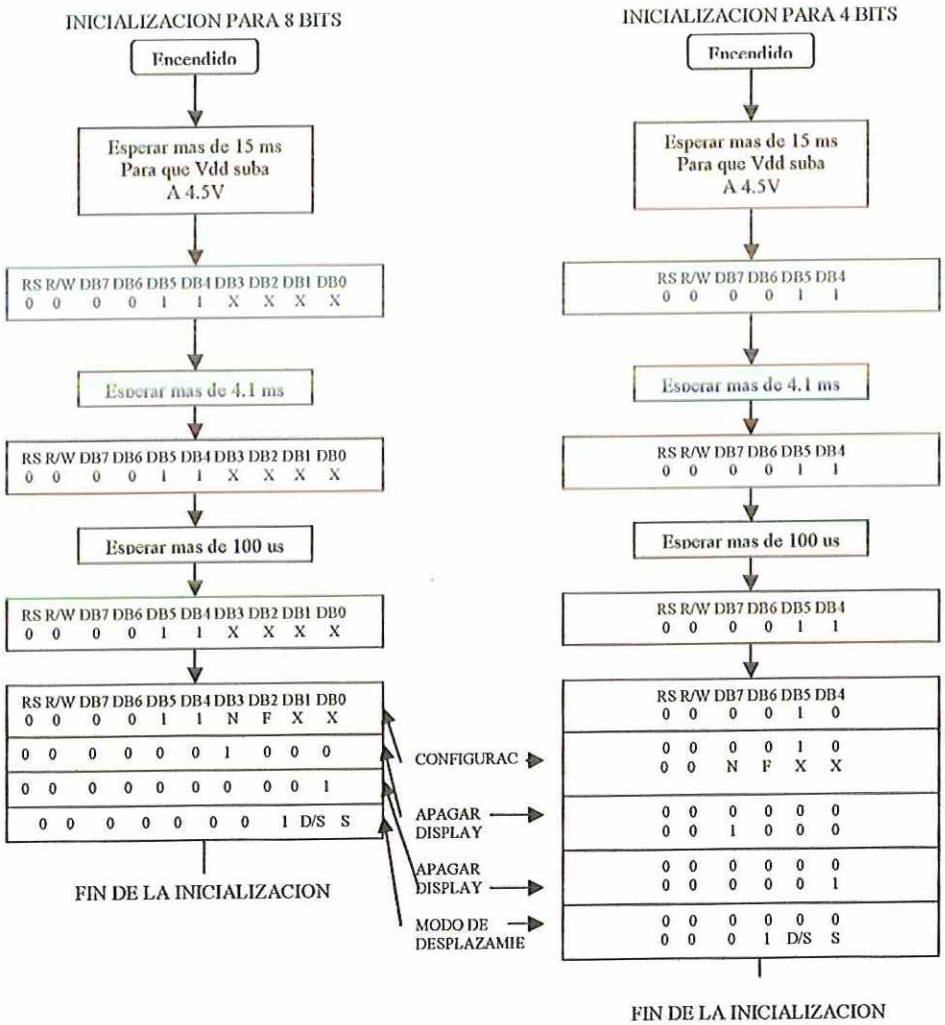


Figura 7. Diagrama de flujo para inicialización del modulo LCD.

2.1.5. Inicialización. Del modulo LCD:

Todo modulo LCD deberá inicializarse, esta inicialización indicara como deberá operar la pantalla. La inicialización representan las instrucciones que deberán ser ejecutadas por el modulo LCD antes de su funcionamiento normal. Las instrucciones que están dentro de la inicialización solamente se ejecuta después que se enciende el modulo LCD y no podrán ser cambiadas posteriormente. Por ejemplo tenemos algunos parámetros que pueden ser ejecutados en la inicialización antes de comenzar a funcionar nuestro modulo LCD:

- Selección de la longitud del bus de datos (4 Bits / 8 Bits).
- Activar el numero de líneas que se visualizaran el modulo LCD.
- Encender el Modulo LCD.
- Las siguientes instrucciones también podrán ser colocadas en la inicialización, con la diferencia que podrán ser cambiadas en cualquier parte del programa.
- Mantener el mensaje fijo y desplazar el cursor.
- Desplazar el mensaje y mantener el cursor fijo.
- Hacer que el carácter señalado parpadee o no.

2.1.6. Conjunto de Instrucciones básicas de un modulo LCD:

Instrucción.	CODIGO										Descripción	Tiempo de ejecución
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Borrar Pantalla	0	0	0	0	0	0	0	0	0	1	Borra la pantalla y retorna el cursor a la dirección 0 (Home)	1.64 mS.
Cursor Home	0	0	0	0	0	0	0	0	1	*	Retorna el cursor al inicio (Dirección o)	1.64 mS.
Modo de entrada de caracteres	0	0	0	0	0	0	0	1	I/D	S	Donde I/D=0 Decremente la posición del cursor, I/D=1 incrementa la posición del cursor,. S=0 El texto de la pantalla no se desplaza, S=1 El texto de la pantalla se desplaza en el momento que se escribe un carácter	40 uS.
Apagado y encendido de la pantalla.	0	0	0	0	0	0	1	D	C	B	Donde D=0 Pantalla apagada, D=1 Pantalla encendida, C=0 Cursor apagado, C=1 Cursor encendido, B=0 Intermitencia del cursor apagado, B=1 Intermitencia del cursor encendido.	40 uS..
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	*	*		40 uS.
Funtion Set	0	0	0	0	1	DL	N	F	*	*		40 uS.
Set CG RAM address	0	0	0	1	ACG							40 uS.
Set DD RAM address	0	0	1	ADD								40 uS.
Ready busy flag & address	0	1	BF	AC								1 uS.
Write data to CG or DD RAM	1	0	Escribir el Dato									120 uS.
Read data to CG or DD RAM	1	1	Leer el Dato									40 uS.

Tabla 3. Instrucciones básicas del LCD

La tabla numero cuatro, esta referida a las nomenclaturas utilizadas en la tabla numero tres

Nomenclatura	Variable = 1	Variable = 0
I/D	I/D=1 Incrementa el Cursor en una posición	I/D=0 Decrementa el Cursor en una posición.
D	D=1 Pantalla Encendida	D=0 Pantalla Apagada.
C	C=1 Cursor Encendido.	C=0 Cursor Apagado.
B	B=1 Intermitencia del cursor encendida.	B=0 Intermitencia del cursor apagado
S/C	S/C=1 Mover todo el texto.	S/C=0 Mover el cursor.
R/L	R/L=1 Mover todo el texto a la izquierda.	R/L=1 Mover todo el texto a la derecha.
DL	DL=1 Bus de datos de 8 Bits.	DL=0 Bus de datos de 4 Bits.
S	S=1 Desplazamiento del texto.	S=0 No desplazamiento del texto
BF	BF=1 Operación Interna en progreso.	BF=0 No puede aceptar instrucción
F	F=1 Matriz para el carácter de 5 X 10 dots	F=0 Matriz del carácter de 5 x 7 Dost
N	N=1 Activación de dos lineas.	N=0 Activación de 1 linea

Tabla 4. Nomenclaturas del LCD

La tabla numero cinco, esta referida a las abreviaturas utilizadas en la tabla numero tres

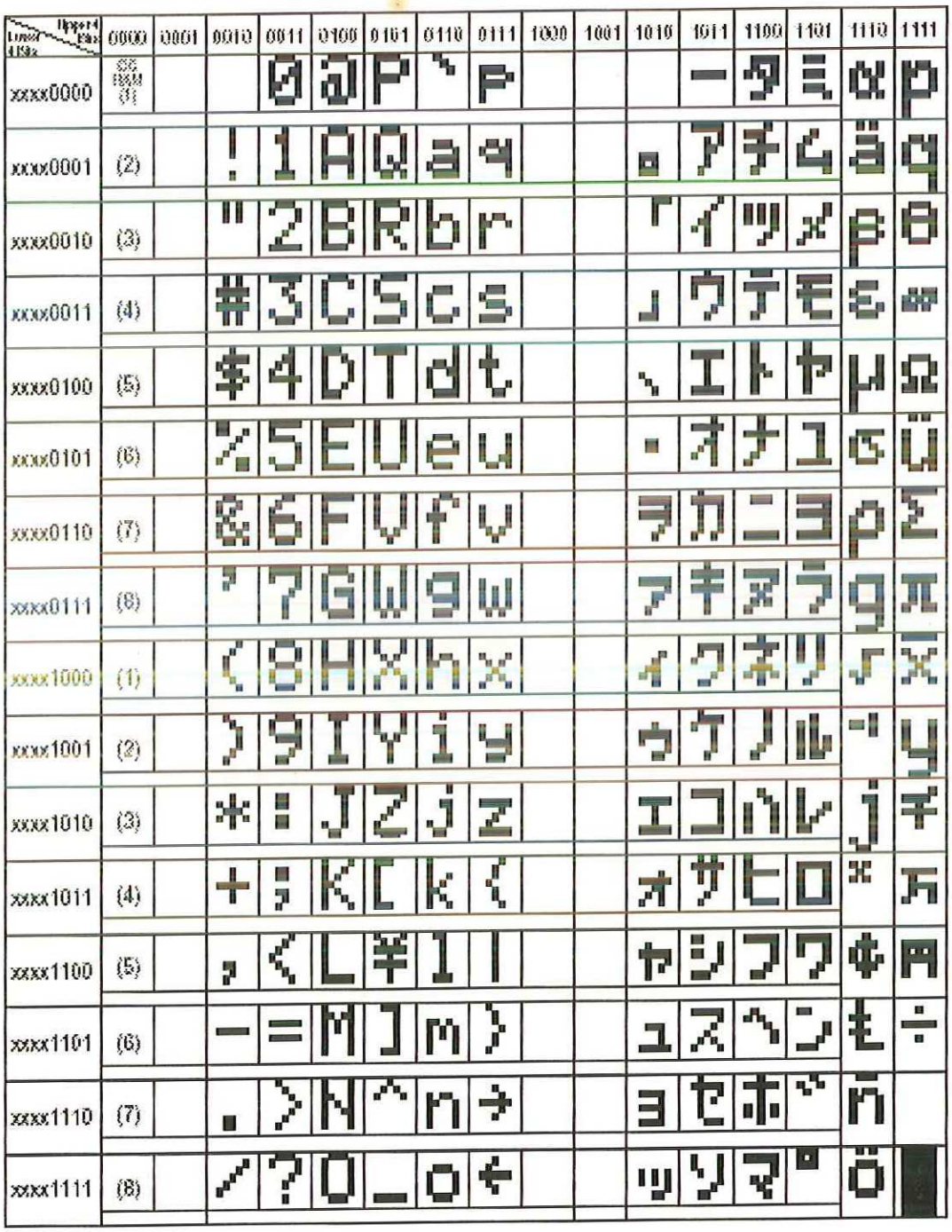
Abreviatura	
DD RAM	Display Data RAM
CG RAM	Generador de Caracteres RAM

Tabla 5. Abreviaturas del LCD

2.1.7 Caracteres que podrán ser presentados en la pantalla del modulo LCD:

La siguiente tabla representan los caracteres que podrán ser mostrados en un modulo LCD. Cada uno de los caracteres tiene su representación binaria de ocho bits. Por ejemplo si usted necesita el carácter "A" deberá representarlo con el

siguiente código 01000001 por otro lado si quisiera utilizar el carácter "T" deberá representarlo por el código 01010100. Este código deberá ser colocado en el Bus de Datos del Modulo LCD (Líneas del 7 al 14).



Note: The user can specify any pattern for character-generator RAM.

Tabla 7. Caracteres del LCD

2.2. TECLADO DE COMANDOS

Desde el punto de vista eléctrico, cada tecla de un teclado es un mecanismo idéntico a un pulsador. La aportación del teclado consiste en la configuración de las teclas para que necesite pocas líneas de entrada en la detección de la que se ha presionado.

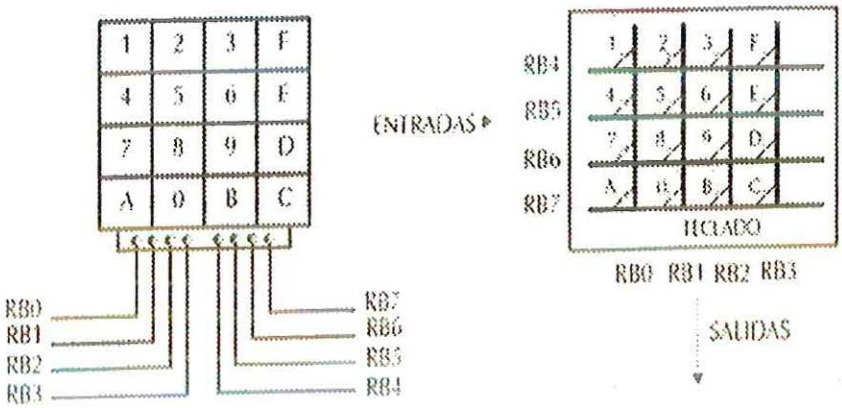


Figura 8. Teclado

Para disminuir las líneas necesarias para detectar la tecla pulsada, estas se agrupan de forma matricial de 16 teclas solo precisa 8 líneas del micro controlador para su gestión. Si cada tecla actuase como un pulsador individual se precisarían 16 líneas de E/S del microcontrolador para gestionarlas.

En la figura 9 las 4 líneas de menos peso de la puerta B (RB0-RB3) se configuran como salidas que aplican un patrón de estados lógicos a las 4 columnas del teclado. Las 4 líneas de más peso de la puerta B (RB4-RB7) están configuradas como entradas y reciben los niveles lógicos que tienen las filas del teclado.

El programa que gestiona el teclado envía, secuencialmente, un nivel bajo por una de las 4 líneas de salida que se aplica a las columnas, al mismo tiempo que lee el nivel lógico introducido por las filas en las líneas RB4-RB7. Si ninguna de las

teclas de la columna por la que se introduce el nivel bajo esta pulsada, se leerá un nivel alto en las cuatro filas, pasándose a activar la siguiente columna.

Si se aplica en una columna un nivel bajo y al leer las filas una de ellas se encuentra un nivel bajo, se deduce que la tecla asociada a dicha fila y dicha columna se haya presionada. Así se averigua la tecla que se presiona cada momento.

También es el software el encargadote realizar el tratamiento oportuno cuando se pulsan varias teclas a la vez, eliminar los rebotes, etc.

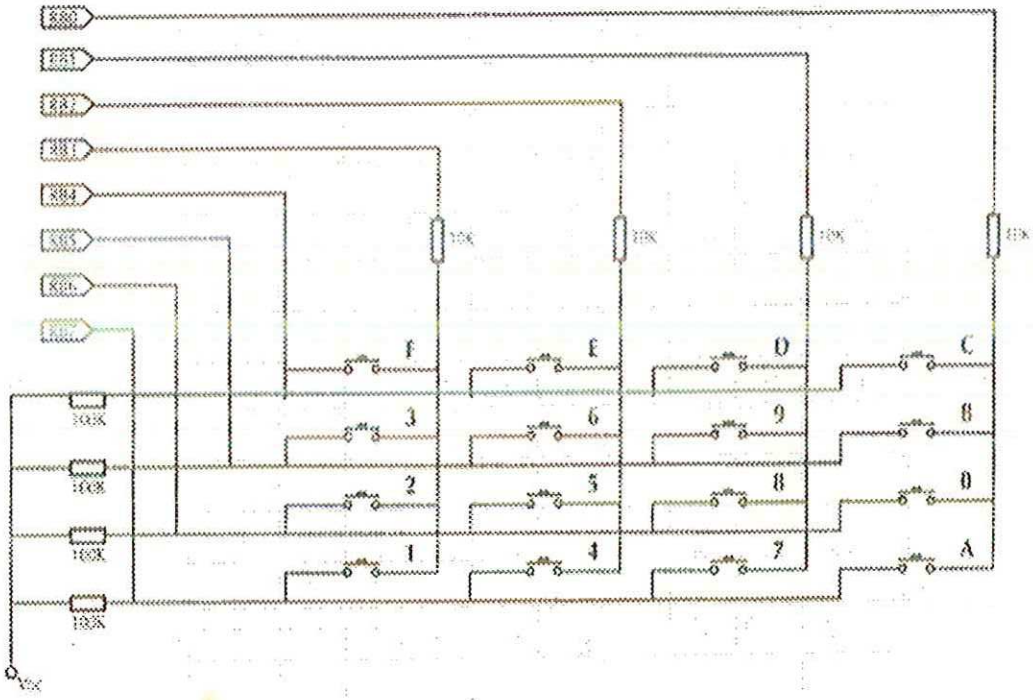


Figura 9. Configuración de un teclado

2.3. RELOJ DE TIEMPO REAL

En numerosas ocasiones, el diseñador requiere registrar un evento lo mismo que la hora y la fecha en que se presenta y son numerosas las ocasiones en las cuales esa necesidad se suple utilizando un microcontrolador para contar los segundos,

los minutos, las horas, los días, etc. Esta por lo regular es una solución aceptable, pero no la mejor.

No es la mejor opción, porque existen circuitos especializados para llevar este tipo de cuentas de una mejor manera, y con un menor consumo de corriente. Actualmente se puede conseguir una buena variedad de circuitos integrados de este tipo, y presentamos uno, que a nuestro juicio, se convertirá en el favorito de los experimentadores y diseñadores electrónicos tanto su bajo costo, su reducido tamaño, la facilidad para la lectura y escritura, su bajo consumo de corriente y el pequeño numero de componentes externos que necesita para un optimo funcionamiento.

Se trata del reloj de tiempo real DS1307, que cumple perfectamente con muchas de las necesidades normales en la adquisición y registro del tiempo. Este circuito integrado es un poderoso reloj calendario en BCD, cuyas características mas destacadas son:

- Reloj de tiempo real que cuenta los segundos, los minutos, las horas, la fecha, el mes, el día de la semana, y el año, con compensación del año bisiesto, valida hasta el año 2100.
- Protocolo I^2C
- 56 bytes de RAM no volátil, para almacenamiento de datos.
- Señal de salida de onda cuadrada, programable.
- Circuiteria interna de respaldo para la alimentación.
- Bajo consumo de potencia: menor a 500 nA en modo de respaldo, a 25ºC.
- Solo 8 pines.

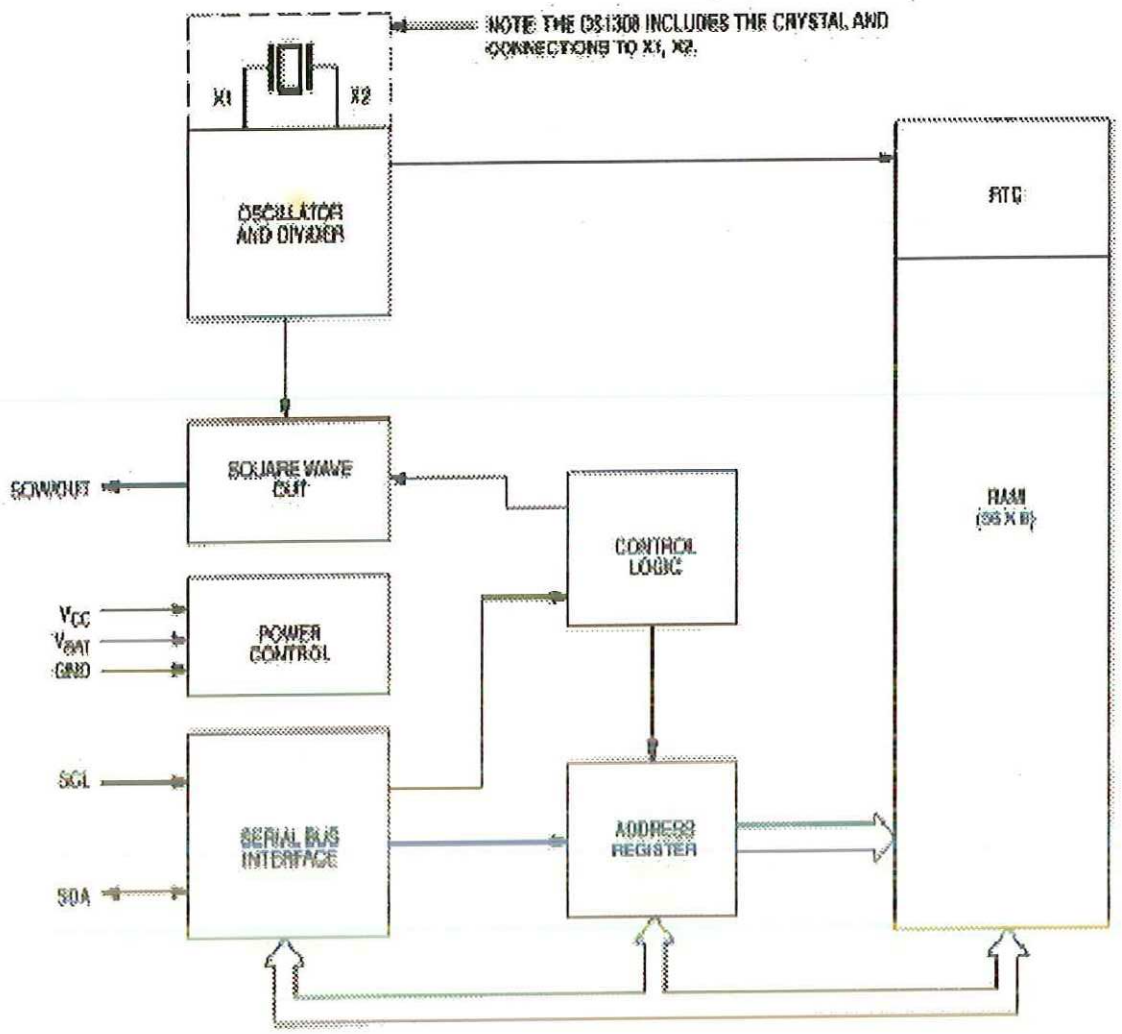


Figura 10. Diagrama a bloques de la estructura interna de un RTC

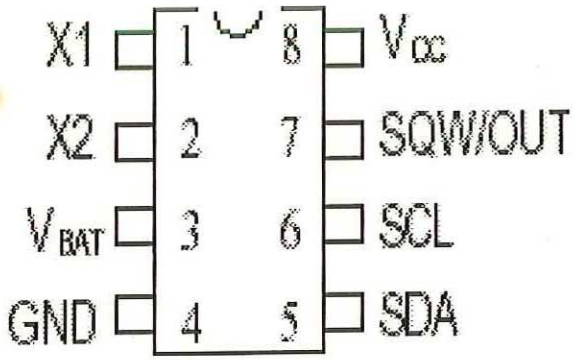


Figura 11. Disposición eléctrica del RTC

2.3.1. Descripción de los pines

VCC, GND: La alimentación DC es proporcionada al circuito a través de estos pines. VCC es la entrada de +5 voltios, mientras que GND es la referencia.

VBAT: Entrada de alimentación de una pila estándar de litio de 3 voltios. El voltaje debe estar entre 2.5 y 3.5 voltios para una operación apropiada. El fabricante hace cuentas que una pila de litio con una corriente de 35 mA o mayor, podrá respaldar la operación del DS1307 por más de 10 años en ausencia de energía.

SCL: Entrada para sincronizar el movimiento de datos en la interfase serial.

SDA: Entrada / salida de datos para la interfase I^2C. Este Pin es de drenaje abierto, por lo cual requiere una resistencia pull-up externa.

SQW/OUT: salida para generar una de las cuatro posibles frecuencias de salida: 1 Hz, 4KHz, 8KHz o 32KHz. Este Pin también es de drenaje abierto, por lo cual requiere la resistencia externa de pull-up.

X1, X2: conexiones para el cristal de cuarzo estándar de 32768Hz. La circuiteria interna del oscilador esta diseñada para obtener un cristal de una capacitancia de 12.5pf.

2.3.2 Mapa de direcciones:

En la siguiente figura se puede observar el mapa de direcciones para los registros del reloj de tiempo real (RTC) y la RAM del DS1307. Los registros del primero se localizan en las direcciones 00H hasta la 07H, mientras que la RAM desde la 08H hasta la 3FH.

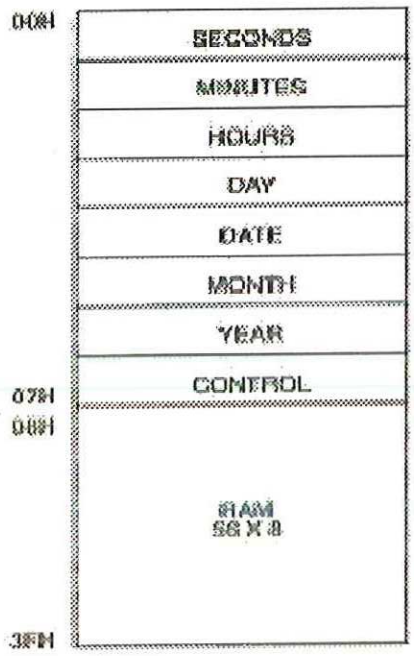


Figura 12. Mapa de direcciones del RTC

Reloj y calendario:

La información del tiempo y el calendario esta en formato BCD (Binario Codificado Decimal), y es obtenida leyendo los registros apropiados; estos se muestran en la figura 14. Como siempre, los bits que tienen "X" en esta figura son intranscendentes para el funcionamiento del circuito integrado.

Como se había mencionado, el RTC puede correr en el formato de 12 o de 24 horas; el bit 6 del registro 2 es el encargado de determinar esta función. Cuando se ajusta a 1, funciona en el modo de 12 horas, y el bit 5 es el indicador AM/PM, siendo 1 cuando es PM. En el formato de 24 horas, el bit 5 de este registro es la parte más significativa de las decenas de horas.

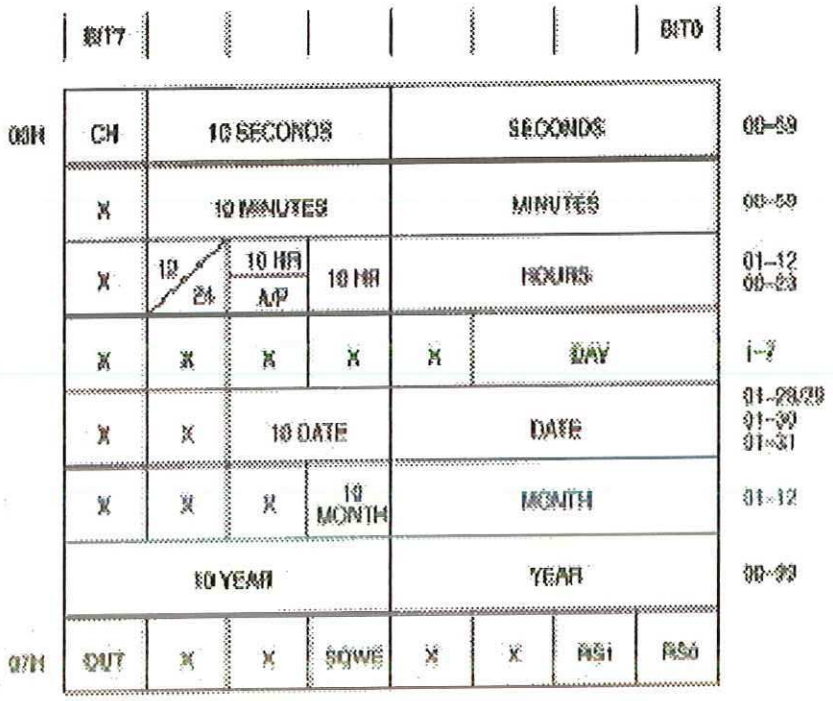


Figura 13. Registros del RTC

2.3.3. Registro de control:

Este registro es utilizado para determinar el funcionamiento del Pin de salida SQW/OUT.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	X	X	SQWE	X	X	RS1	RS0

Tabla 6. Registro de control SQW/OUT

OUT: Es el control de nivel de salida cuando la salida de la onda cuadrada esta deshabilitada. Si SQWE=0, el nivel en el Pin 7 del Ds1307 será alto si el bit OUT=1, y será bajo sí el bit OUT=0.

SQWE: Este bit cuando se ajusta a uno habilita la salida del oscilador. La frecuencia de onda cuadrada dependerá de los bits RS1 y RS0.

RS1, RS0: Estos bits controlan la frecuencia de onda cuadrada de salida, cuando esta se habilita. La siguiente tabla muestra las frecuencias que pueden seleccionarse con los bits RS1, RS0.

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

Tabla 7. Frecuencias determinadas por los bits RS1, RS0

2.3.4. Transferencia de datos:

Como se menciono, el DS1307 soporta un bus bidireccional de dos líneas y el protocolo I^2C patentado por PHILIPS SEMICONDUCTORS. En las dos líneas puede existir un buen número de transmisores y receptores de datos. El dispositivo que coloca datos sobre el bus es denominado transmisor, mientras que el que recibe datos se denomina receptor. Allí, el circuito integrado que controla el mensaje es llamado maestro (mas bien debería denominarse amo, ya que proviene de la palabra inglesa MASTER), mientras que aquellos que son dominados por el maestro son llamados esclavos (SLAVES).

El bus debe ser controlado por un dispositivo maestro, el cual genera la señal de reloj, controla el bus de acceso y genera las condiciones de arranque (Start) y paro (Stop).

El DS1307 opera como un esclavo en el bus I^2C, el cual puede tener una configuración típica. La forma de transferir los datos debe ajustarse al protocolo I^2C:

- La transferencia de datos puede ser iniciada solamente cuando el bus no esta ocupado.
- Durante la transferencia de datos, la línea SDA debe permanecer estable siempre que la línea de reloj SCL este en ALTO. Cambios en la línea SDA mientras la línea SCL este en alto serán interpretados como señales de control.

Concordando con lo anterior, las siguientes condiciones del bus han sido definidas.

Bus no ocupado: Cuando las líneas de datos y de reloj permanecen en un nivel alto.

Condición de arranque: El cambio en la línea de datos de alto a bajo, mientras la línea de reloj es alto.

Condición de paro: El cambio en la línea de datos de bajo a alto, mientras la línea de reloj es alto.

Dato valido: El estado de la línea de datos representa un dato valido cuando, después de una condición de arranque, la línea de datos es estable durante el estado alto de la señal de reloj. Se presenta un pulso de reloj por cada bit de dato.

Cada transferencia de datos es iniciada con una condición de arranque, y terminada con una condición de paro. El numero de bytes transferidos entre las condiciones de arranque y paro no esta limitada, y es determinada por el

dispositivo maestro. La información es transferida por bytes, y cada receptor genera un reconocimiento (acknowledge) durante un noveno bit.

Reconocimiento: Cada dispositivo, cuando es diseccionado, es obligado generar un reconocimiento de la recepción de cada byte. El maestro debe generar un pulso extra de reloj, el cual esta asociado con el bit de reconocimiento. El receptor debe llevar a un nivel bajo la línea SDA con este noveno bit, y mantenerlo estable durante este periodo.

2.3.5. Escritura de datos en el DS1307

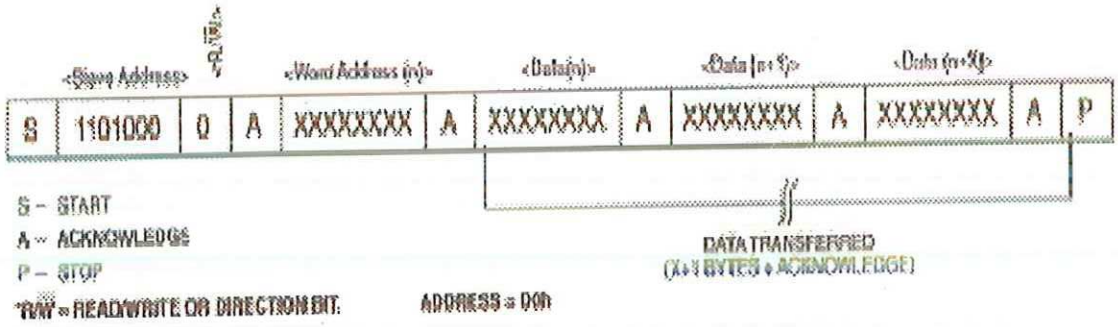


Figura 14. Escritura de datos en el RTC

Después de la generación de la condición de arranque, el primer byte transmitido por el maestro es la dirección esclava (que para el caso del DS1307 es el numero binario 1101000) + el bit que define la operación a realizar, que en este caso, ya que se trata de una escritura, es cero. El maestro genera el noveno bit para que el DS1307 responda colocando la línea SDA en cero.

Seguidamente se transmiten los ocho bits que conforman la dirección desde la cual se pretenden escribir los datos (que para el caso del DS1307 están desde la 00H hasta la 3FH), generando el noveno bit para el reconocimiento por parte del esclavo. Con este byte recibido, el DS1307 carga un puntero de direcciones. Que

señala la posición que se accederá dentro de el. Este puntero se incrementa automáticamente con cada lectura o escritura de datos.

Posteriormente, se transmiten uno a uno los bytes que conforman los datos a escribir en el reloj de tiempo real, y el DS1307 responde con el reconocimiento. Aquí, conviene recordar que solo se tienen 64 registros en total, y que la escritura en el registro 65 estaría sobre escribiendo el primero.

Para finalizar la escritura se genera la condición de paro, con la cual se libera el bus para que pueda llevar a cabo operaciones sobre otros dispositivos conectados a este.

2.3.6. Lectura de datos en el DS1307

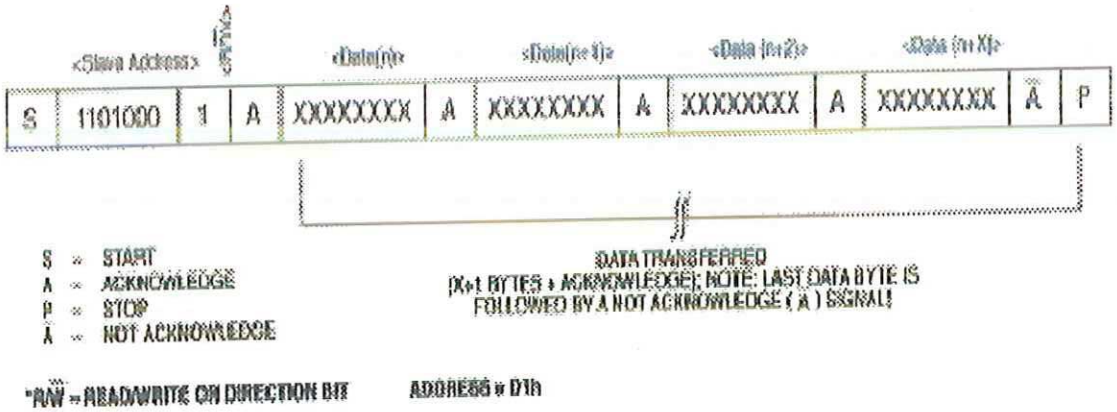


Figura 15. Lectura de datos en el RTC

El primer byte transmitido por el maestro esta conformado por la dirección del RTC (1101000) y el bit que indica la operación a realizar que en este caso es 1, ya que se trata de una lectura; el maestro genera el noveno bit, para el reconocimiento por parte del DS1307. A continuación, el DS1307 empezara a trasmitir los datos, empezando con el registro señalado por el puntero de direcciones.

El maestro se encarga de generar los pulsos de reloj, leer la línea de datos para conformar los bytes que están siendo enviados por el DS1307 y de generar los bits de reconocimiento, a excepción del ultimo dato, en el cual el maestro no genera el bit de reconocimiento, indicándole así al DS1307 que no se requiere el envió de mas datos por el momento.

Para finalizar, el maestro genera la condición de paro, liberando el bus para otras operaciones.

2.4. MEMORIA 24LC65/02

Las técnicas para almacenar información en medios electrónicos se perfeccionan más cada día. A diario vemos ejemplos de su utilización en nuestros hogares y oficinas, por ejemplo, en sintonizadores de televisión, reproductores de compact disk, sistemas de control remoto, impresoras, fotocopiadoras, teléfonos celulares, etc. Una de estas tecnologías corresponde a las llamadas memorias EEPROM seriales, las cuales tienen grandes ventajas si se comparan con otras posibilidades. Entre sus principales características se cuentan:

- Se puede conectar fácilmente con microprocesadores o microcontroladores, inclusive algunos de ellos tienen pines dedicados para esta labor.
- Transferencia de datos de manera serial, lo que permite ahorrar pines del micro para dedicarlos a otras funciones.
- Ocupan la décima parte del espacio de las memorias que trabajan en paralelo, esto permite ahorrar dinero debido al menor tamaño del circuito impreso.
- El consumo de corriente es mucho menor que en las memorias que trabajan en paralelo, esto las hace ideales para sistemas portátiles que funcionan con baterías.

Una vez que un diseñador se ha decidido a incorporar una memoria EEPROM serial en su circuito, el siguiente paso consiste en escoger el dispositivo mas adecuado. Por ejemplo, un dispositivo de 2 hilos se emplea cuando se requiere utilizar el bus I^2C, cuando se necesita inmunidad al ruido o cuando se tiene un número limitado de pines en el microcontrolador. Un dispositivo de 3 hilos se emplea cuando se requiere altas velocidades de transferencia de información o hay que manejar datos de 16 bits de longitud. En la siguiente tabla se muestra las principales características de cada uno.

2.4.1. MEMORIAS 24xx

Estas memorias utilizan el bus de 2 hilos para comunicarse con otros dispositivos. Dado que cumplen con el protocolo I2C, tienen un Pin llamado SCL que recibe los pulsos generados por el dispositivo maestro y otro llamado SDA que maneja el flujo de datos de forma bidireccional (entrada / salida). En la figura 16 se muestra el diagrama de pines correspondiente a estas memorias. Este dispositivo no requiere de un Pin habilitador o chip select ya que en este esquema la transferencia de información solo se puede iniciar cuando el bus este libre. En este caso, como cada dispositivo tiene su dirección determinada mediante los pines A0, A1y A2, solamente responderá la memoria cuya dirección coincida con la dirección que va encabezando la trama de información.

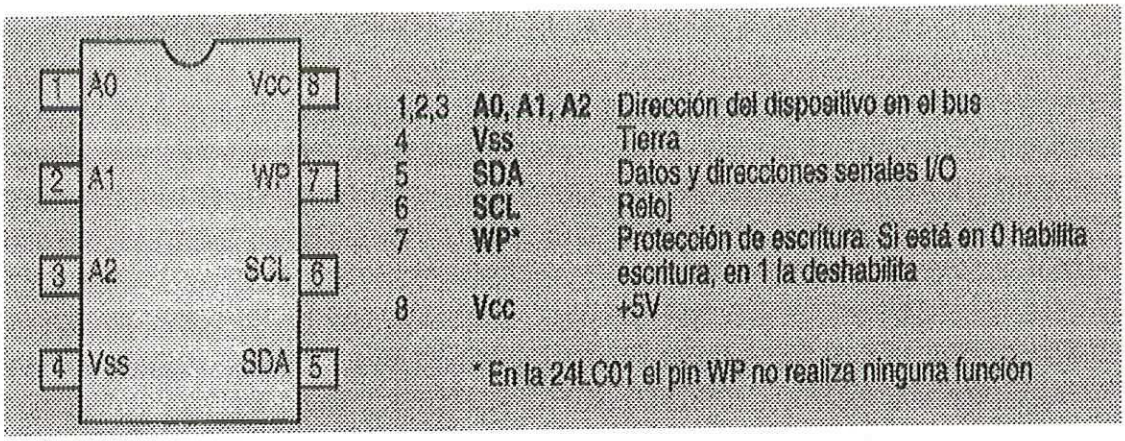


Figura 16. Descripción de los pines de la memoria

Transferencia de información

Cuando el microcontrolador desea entablar comunicación con la memoria, debe enviarle una serie de bits que llevan la siguiente información:

- Se envía el bit de arranque o Start bit.
- El código 1010
- La dirección del dispositivo (A2, A1, A0)
- Un bit que indica que se desea escribir en la memoria ("0")

Luego de esto la memoria debe enviar un reconocimiento para informarle al microcontrolador que recibió la información. Dicho asentimiento, llamado ACK, consiste en poner en el bus en un nivel bajo (lo hace la memoria). Después el microcontrolador debe enviar los bits que corresponden a la posición de memoria que se quiere leer o escribir; Nuevamente la memoria envía un reconocimiento. El paso siguiente depende de la operación que se vaya a ejecutar. Si se trata de un proceso de escritura, el microcontrolador debe enviar el dato a ser almacenado y esperar el asentimiento por parte de la memoria. Si se trata de una lectura, nuevamente se debe repetir los primeros cuatro pasos, solo que en lugar de un cero que indica escritura, se debe enviar un "1" que indica lectura. Después se espera el asentamiento y luego se puede leer el byte con el dato que estaba en la posición de memoria que se indico anteriormente. Cuando se termina la operación, el microcontrolador debe enviar una señal de parada o Stop bit.

2.5 CONECTOR RS232-MAX232

El **MAX232** dispone internamente de 4 conversores de niveles TTL al bus standard RS232 y viceversa, para comunicación serie como los usados en los ordenadores y que ahora están en desuso, el Com1 y Com2.

Funcionamiento:

El circuito integrado lleva internamente 2 conversores de nivel de TTL a RS232 y otros 2 de RS232 a TTL con lo que en total podremos manejar 4 señales del puerto serie del PC, por lo general las mas usadas son; TX, RX, RTS, CTS, estas dos ultimas son las usadas para el protocolo **handshaking** pero no es imprescindible su uso. Para que el MAX232 funcione correctamente debemos de poner unos condensadores externos, todo esto lo podemos ver en la siguiente figura en la que solo se han cableado las líneas TX y RX que son las más usualmente usadas para casi cualquier aplicación.

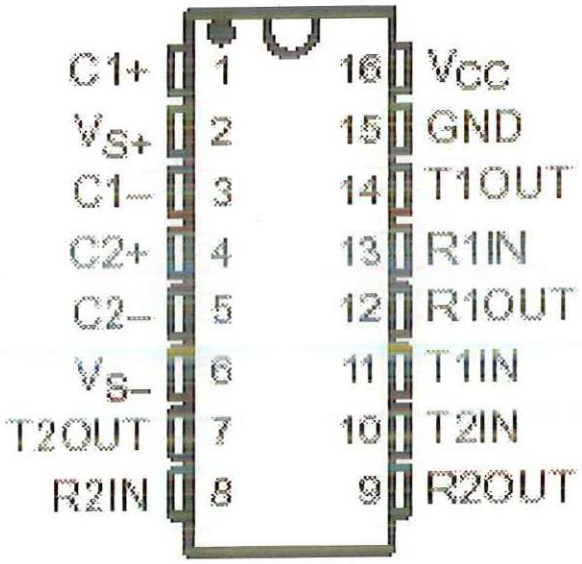


Figura 17. Esquema de conexión de MAX232.

MODEM CMX868

El CMX868 es un MODEM multi-estándar para uso en sistemas basados en información telefónica y Telemetría. El control del dispositivo es mediante bus serial simple de alta velocidad, compatible con casi todos los tipos de interfaces seriales de microcontroladores. Los datos transmitidos y recibidos por el MODEM son también transferidos sobre el mismo bus serial. Los USART's programables que vienen incorporados en el chip cumplen con los requerimientos del V.14 los cuales son suministrados para usar con datos asincrónicos y permitir que datos sincrónicos sin formatear sean recibidos y transmitidos como palabras de 8 bits.
El MODEM puede transmitir y detectar tonos DTMF estándar y señales de llamada y respuesta del MODEM o señales de tonos sencillos o dobles programadas por el usuario. También viene incluido un detector del progreso de llamada con propósito general.

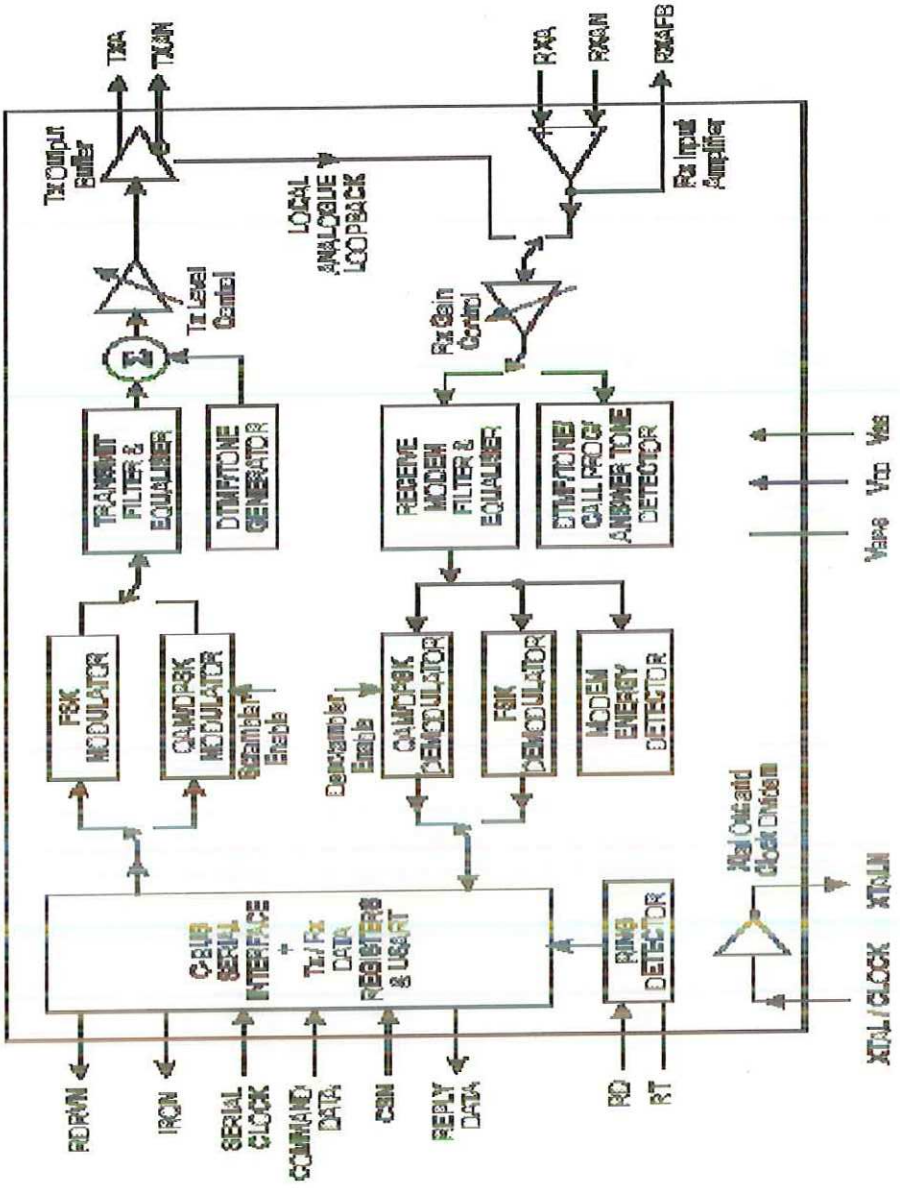


Figura 18. Estructura interna del MODEM CMX868.

CMX868 D2/E2/P4	Señal		Descripción
Pin No.	Nombre	Tipo	
1	XTALN	O/P	La salida del inversor interno del oscilador Xtal.
2	XTAL/CLOCK	I/P	La entrada del inversor del oscilador desde el circuito del Xtal o la fuente externa de reloj.
3	RDRVN	O/P	Salida de manejo del relé, baja resistencia de pull down a Vss en activo y mediana resistencia de pull up a Vdd en inactivo.
4, 8, 12, 17, 21	Vss	Power	La Línea negativa de alimentación (tierra).
5	RD	I/P	Entrada con Schmitt trigger hacia el detector de timbre. Conectar a Vss si no se usa el detector.
6	RT	BI	Salida de colector abierto o entrada con Schmitt trigger, formando parte del detector de timbre. Conectar a Vdd si no se usa el detector.
7, 16, 24	Vdd	Power	La línea de alimentación positiva. Niveles y umbrales dentro del dispositivo son proporcionales a este voltaje.
9	RXAFB	O/P	La salida del amplificador de entrada de Rx.
10	RXAN	I/P	La entrada invertida del amplificador de entrada de Rx.
11	RXA	I/P	La entrada no invertida del amplificador de entrada de Rx.
13	Vbías	O/P	Voltaje bias generado internamente de aproximadamente Vdd/2, excepto cuando el dispositivo esta en el modo de ahorro de potencia (Vbías conectado a Vss). Se desacopla con un capacitor colocado cerca del pin.
14	TXAN	O/P	La salida invertida del Buffer de salida Tx.
15	TXA	O/P	La salida no invertida del Buffer de salida Tx.
18	CSN	I/P	La entrada de selección del Chip del C-BUS desde el microcontrolador.
19	COMMAND DATA	I/P	La entrada serial de datos del C-BUS desde el microcontrolador.
20	SERIAL CLOCK	I/P	La entrada serial del reloj del C-BUS desde el microcontrolador.
22	REPPLY DATA	T/S	La salida serial de datos de 3 estados del C-BUS hacia el microcontrolador. Se coloca en alta impedancia cuando no se envían datos.
23	IRQN	O/P	La salida para conexión a la entrada de interrupción externa del microcontrolador. Cuando esta activa se coloca a Vss y cuando esta inactiva se coloca en alta impedancia. Requiere un resistor externo de pull up.

Tabla 8. Descripción de los pines del MODEM CMX868.

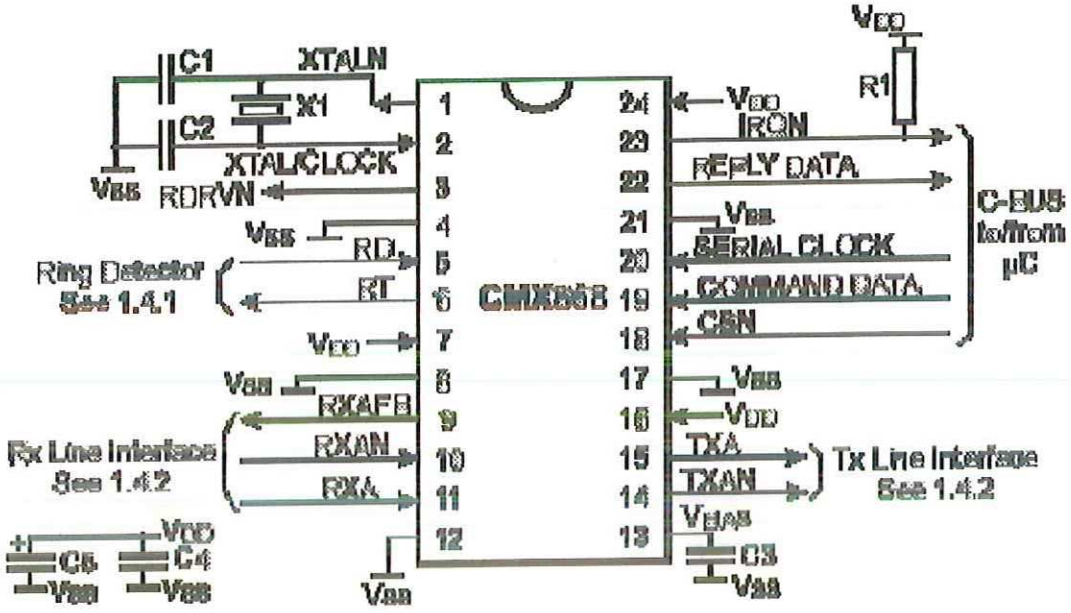


Figura 19. Componentes externos recomendados para una aplicación típica.

Este dispositivo es capaz de detectar y decodificar señales de pequeña amplitud. Para permitir esto Vdd y Vbías se deben desacoplar y la línea de recepción debe ir protegida de señales extrañas (ruido de línea). Es recomendado que el circuito impreso este diseñado con un plano de tierra bajo el área del CMX868 para proporcionar una conexión de baja impedancia entre los pines de Vss y Vdd, y los capacitares de desacople de Vbías. Las conexiones de Vss hacia los capacitores del oscilador de cristal C1 y C2 deben también tener baja impedancia y preferiblemente debe ser parte o estar sobre el plano de tierra para asegurar un arranque confiable del oscilador.

Interfase de detección de timbre

La mínima amplitud de la señal de timbre que puede ser detectada es:

$$\left[0.7 + Vthi * \frac{(R20 + R22 + R23)}{R23}\right] * 0.707Vrms$$

Donde Vthi voltaje umbral de subida del Schmitt trigger 'A'.

Con R20 - R22 todas en 470k. Colocando R23 a 68k. Se garantizara la detección de señales de timbre de 40 Vrms y superiores para un rango de Vdd desde 3 hasta 5 voltios.

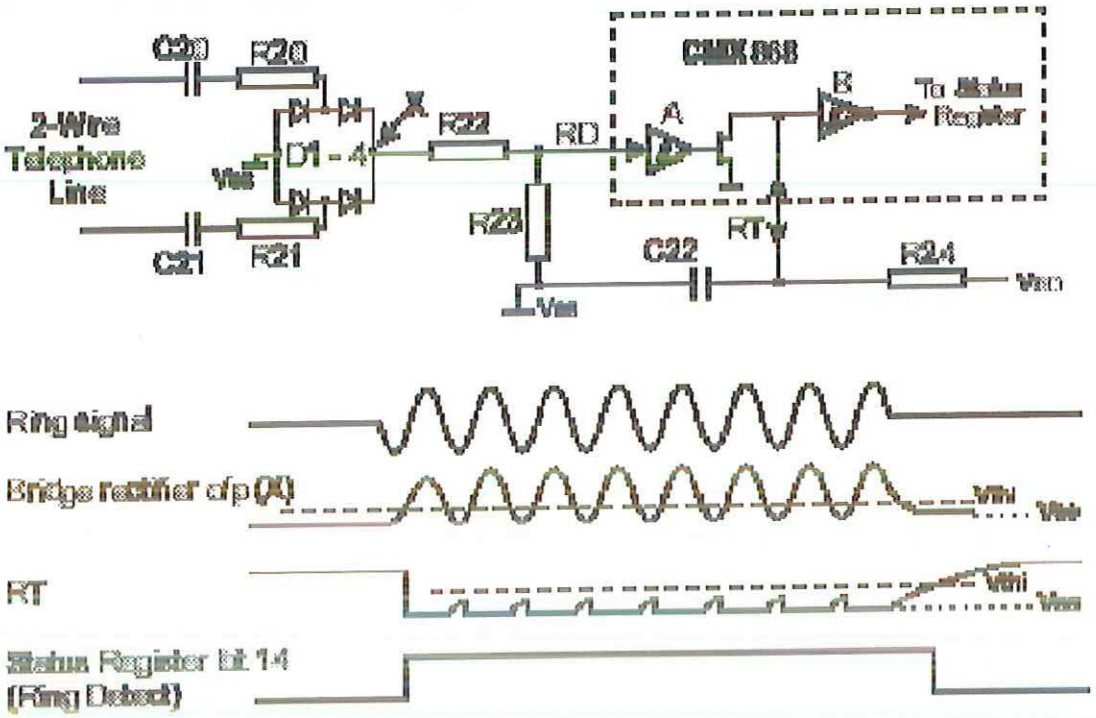


Figura 20. Circuito de interfase de detección de la señal de timbre.

Si la constante de tiempo de R24 y C22 es suficientemente grande entonces el voltaje en RT permanecerá por debajo del umbral del Schmitt trigger 'B' durante el transcurso de un ciclo de timbre. El tiempo para que el voltaje en RT se cargue desde Vss hasta Vdd puede ser derivado de esta formula:

$$Vrt = Vdd * \left[1 - e^{-t/(R24*C22)} \right]$$

Como el voltaje umbral de entrada del Schmitt trigger (Vthi) tiene un valor mínimo de 0.56xVdd, entonces la salida del Schmitt trigger 'B' permanecerá en alto por un tiempo t al menos 0.821xR24xC22 seguido de un pulso en RD.

Los valores de R24 y C22 dados (470k. y 0.33µF) dan un tiempo de carga mínimo en RT de 100mseg, el cual es adecuado para frecuencias de timbre de 10 Hz o superiores. Note que este circuito también responderá a una inversión de voltaje de línea telefónica. Es necesario que el microcontrolador pueda distinguir entre la señal de timbre y la inversión del voltaje de línea midiéndole tiempo que el bit 14 del registro de estado (detección de timbre) permanece en alto. Si la función de detección de timbre no se utiliza entonces el pin RD debe ser conectado a Vss y RT a Vdd.

3.1. Descripción General

Los modos operativos de recepción y transmisión del CMX868 son programables independientemente. El modo de transmisión puede ser colocado en cualquier configuración de las siguientes:

- V.22 bis MODEM. 2400bps QAM (modulación de amplitud por cuadratura).
- V.22 y Bell 212A MODEM. 1200 ó 600bps DPSK (Desplazamiento diferencial de fase).
- V.21 MODEM. 300bps FSK (Desplazamiento por códigos de frecuencia). Bell 103 MODEM. 300bps FSK.
- Bell 202 MODEM. 1200 o 150 bps FSK.
- Transmisión DTMF.
- Transmisión de tono sencillo (desde llamada de MODEM, respuesta y otros tonos).
- Transmisión por par de tonos o por tonos programados por el usuario (frecuencias y tonos programables).

El modo de recepción puede ser colocado en cualquier configuración de las siguientes:

- V.22 bis MODEM. 2400bps QAM (modulación de amplitud por cuadratura).

- V.22 y Bell 212A MODEM. 1200 o 600bps DPSK (Desplazamiento diferencial de fase).
- V.21 MODEM. 300bps FSK (Desplazamiento por códigos de frecuencia).
- Detección DTMF.
- Detección de tono de respuesta 2100Hz y 2225Hz.
- Detección de señales de progreso de llamada.
- Detección de par de tonos o tonos programados por el usuario.

El CMX868 puede ser colocado en modo de bajo consumo de potencia el cual deshabilita toda la circuitería excepto por la interfase del C-BUS y el detector de timbre.

3.2. TX USART

Un flexible TX USART es proporcionado para todos los modos del MODEM, cumpliendo con los requerimientos de V.14 para QAM y DPSK módems. Este puede ser programado para transmitir patrones continuos, caracteres de Stara-Stop, y datos síncronos. En ambos modos datos síncronos y modos de Start-stop, los datos a ser transmitidos son escritos por el microcontrolador en el registro de datos TX de 8 bits del C-BUS desde el cual es transferido al buffer de datos de TX.

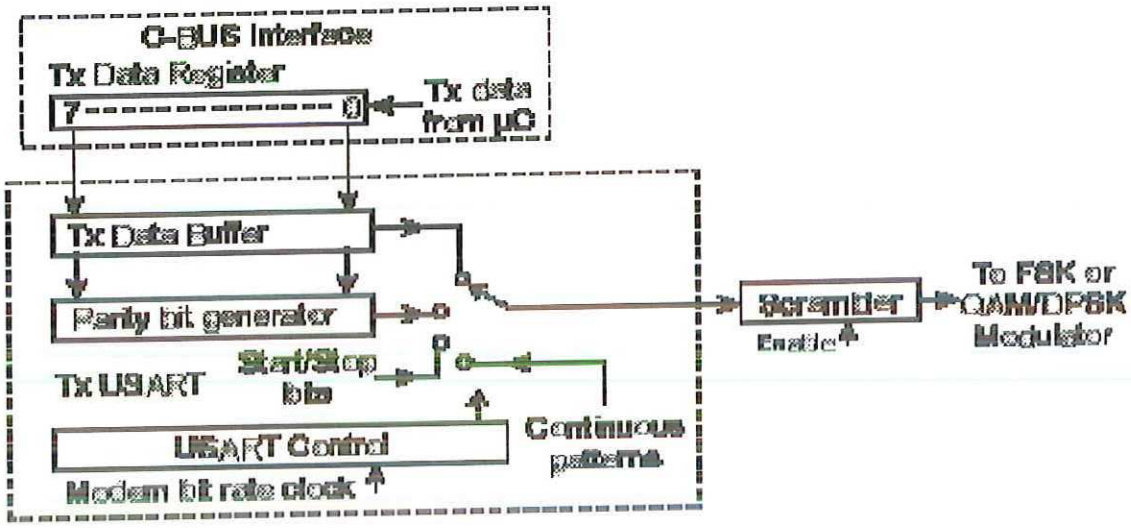


Figura 21. TX USART

Cada vez que los contenidos del registro de transmisión de datos del C-BUS son transferidos al buffer de transmisión de datos la bandera de datos transmitidos del registro de estado es colocada en uno para indicar que un nuevo valor puede ser cargado en el registro de transmisión de datos de C-BUS. Esta bandera es colocada en cero cuando un nuevo valor es cargado en el registro de transmisión de datos.

Si un nuevo valor no es cargado en el registro de transmisión de datos a tiempo para la siguiente transferencia del registro de transmisión de datos al buffer de transmisión de datos, entonces el bit de sobré flujo de datos transmitidos del registro de estado es colocado en uno. En este caso los contenidos del buffer de transmisión de datos son retransmitidos si el modo de datos síncronos ha sido seleccionado, o de los contrario si el modo seleccionado es el de Start- Stop entonces una señal continua de parada será transmitida hasta que un nuevo valor sea cargado en el registro de transmisión de datos.

En todos los modos los bits transmitidos y las velocidades de transmisión son nominales para el tipo de MODEM seleccionado, por una precisión determinada por la exactitud de la frecuencia del cristal, sin embargo para los modos QAM y

DPSK V.14 requiere que los caracteres de Start- Stop puedan ser transmitidos con una sobré velocidad del 1% (para el rango básico) o del 2.3% (para el rango extendido) mediante el borrado de un bit de Stop de cada 8 (rango básico) o 4 (rango extendido) caracteres transmitidos consecutivamente.

Para acomodar los requerimientos del V.14 el registro de transmisión de datos ha sido dotado con dos direcciones del C-BUS, $E3 y $E4. Los datos deben ser normalmente escritos en $E3.

En el modo Start-Stop para QAM o DPSK si los datos se escriben en $E4, entonces el numeró programado de bits de parada puede ser reducido por uno para ese carácter. De esta manera el microcontrolador puede borrar bits de parada transmitidos de acuerdo a lo que necesite. En el modo de Start-Stop para FSK, los datos escritos en $E4 pueden ser transmitidos con una reducción del 12.5% en la longitud del bit de parada hacia el final de cada carácter.

3.3. USART y Registro de datos Rx

Un flexible USART Rx es proporcionado para todos los modos de MODEM, cumpliendo con los requerimientos del V.14 para módems QAM y DPSK. Este puede ser programado para procesar las cadenas de bits de datos recibidas como datos síncronos o como caracteres de Start-Stop.

En el modo síncrono los bits de datos recibidos son conducidos al buffer de datos de Rx el cual es copiado en el registro de datos de recepción del C-BUS después de cada 8 bits.

En el modo de Start-Stop La lógica de control del USART busca por el inicio de cada carácter, luego almacena el número requerido de bits de datos (sin paridad) en el buffer de datos de Rx.

El bit de paridad (si es usado) y la presencia de un bit de parada son verificadas y los bits de datos del buffer de datos de Rx copiados al registro de datos de recepción del C-BUS.

Cada vez que un nuevo carácter es copiado en el registro de datos de recepción del C-BUS, la bandera de datos listos de Rx en el registro de estados es colocada en uno para indicarle al microcontrolador que puede leer el nuevo dato, y en el modo de Start-Stop, la bandera de paridad par del registro de estados es actualizada.

En el modo de Start-Stop, Si el bit de parada no es detectado (Recibe un cero en vez de un uno) el carácter recibido permanecerá en el registro de datos de recepción y la bandera de datos listos de Rx será colocada, pero, a menos que esto sea permitido por la opción de sobre velocidad V.14 descrita a continuación, el bit de error de cuadro del registro de estados será colocado en uno y el USART será resincronizado sobre la siguiente transición de '1'-'0' (Start-Stop). El bit de error de cuadro permanecerá activo hasta que el próximo carácter sea recibido.

Figura 22. Funcionamiento del Rx USART (modo de Start-Stop, 8 bits de datos + paridad)

Si el microcontrolador no ha leído el dato anterior del registro de datos de recepción pasado un tiempo los datos nuevos son copiados a este del buffer de recepción de RX y la bandera de desbordamiento de datos del registro de estados será colocada en uno.

La bandera de datos listos de Rx y el bit de desbordamiento de datos de Rx son llevados a cero cuando el registro de datos de recepción es leído por el microcontrolador.

Para los modos de Start-Stop QAM y DPSK, el V.14 que el receptor USART sea capaz de determinar los bits de parada perdidos; por encima de un bit de parada perdido en cada 8 caracteres recibidos consecutivamente es permitido para el +1% de sobré velocidad (tasa de señal básica) en el modo V.14 y de 1 a 4 para el modo de sobré velocidad del +2.3% (tasa de señal extendida).

Para acomodar los requerimientos del V.14, el registro de modos de recepción del CMX868 puede ser colocado en cero, para la operación en los modos Start-Stop QAM o DPSK operando a sobre velocidades del +1% o +2.3%. Los bits de parada perdidos mas allá de lo permitido por la opción de sobre velocidad seleccionada colocaran la bandera de errores de cuadros de Rx del registro de estados.

Para buscar que las señales de interrupción recibidas puedan ser manejadas correctamente en el modo de sobré velocidad de Rx V.14, un carácter recibido que tenga todos los bits en cero, incluyendo el de parada y cualquier bit de paridad, puede causar siempre al activación del bit de errores de cuadros y la resincronización del USART sobre la siguiente transición de '1' a '0'.

Adicionalmente el detector de ceros continuos recibidos responderá cuando mas de 2M + 3 ceros consecutivos sean recibidos, donde M es el numero total seleccionado de bits por carácter incluyendo parada y cualquier bit de paridad.

3.4. Interfase C-BUS

Este bloque es el encargado de la transferencia de datos y control o información del estado entre los registros internos del CMX868 y el microcontrolador sobre el bus serial C-BUS. Cada transferencia consiste de un byte de dirección de registro enviado desde el microcontrolador el cual puede ir precedido de uno o mas bytes de datos enviados desde el microcontrolador los cuales van dirigidos hacia uno de los registros de solo escritura del CMX868, o uno o mas bytes de datos leídos desde uno de los registros de solo lectura del CMX868.

Los datos enviados desde el microcontrolador sobre la línea de comandos registrados en el CMX868 en cada flanco de subida de la línea del reloj serial. Los datos de respuesta enviados desde el CMX868 hacia el microcontrolador son validos cuando el reloj serial esta en alto. La línea CSN debe mantenerse en bajo durante la transferencia de datos y en alto entre cada transferencia de comandos. La interfase C-BUS es compatible con la mayoría de interfases seriales de microcontroladores y puede ser fácilmente implementada con pines de I/O de propósito general del microcontrolador controlados por una simple rutina de software.

Las siguientes direcciones y registros del C-BUS son usadas por el CMX868:

Comando de reset general (solo dirección)	Dirección $01
Registro de control general, 16 bits solo escritura	Dirección $E0
Registro de modo de transmisión, 16 bits solo escritura	Dirección $E1
Registro de modo de recepción, 16 bits solo escritura	Dirección $E2
Registro de datos transmitidos, 8 bits solo escritura	Dirección $E3 & $E4
Registro de datos recibidos, 8 bits solo lectura	Dirección $E5
Registro de estados, 16 bits solo lectura	Dirección $E6
Registro de programación, 16 bits solo escritura	Dirección $E8

3.4.1. Comando General de Reset (Sin datos) Dirección del C-BUS $01

Este comando resetea el dispositivo y inicializa todos los bits de los registros de control General, de modo de transmisión, de modo de recepción y los bits 15 y 13-0 del registro de estados.
Todas las veces que la alimentación sea aplicada al CMX868 un comando General de Reset debe ser enviado al dispositivo, después del cual el registro de control general debe ser configurado como se requiera.

3.4.2. Registro de Control General: 16-bit Solo escritura. Dirección del C-BUS $E0

Este registro controla características generales del CMX868 como los modos de bajo consumo de potencia y realimentación, los bits de enmascaramiento del IRQ y la salida de manejo del relé. También permite que los ecualizadores de compromiso fijo en TX y los caminos de señal en Rx sean deshabilitados si se desea, y configura los divisores de reloj internos para usar cristales de ambas frecuencias, tanto 11.0592 o 12.288 MHz.

Todos los bits de este registro son llevados a cero mediante el comando General de Reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Xtal freq	LB	Equ	Rly drv	Pwr	Rst	Irqn en	IRQ Mask bits					

Tabla 9. Registro general de control

Bits 15-13 Registro General de Control: Reservados, colocados en 000

Bit 12 Registro General de Control: frecuencia del Cristal

Este bit debe ser configurado de acuerdo a la frecuencia del cristal externo.

b12 = 1	11.0592MHz
b12 = 0	12.2880MHz

Bit 11 Registro General de Control: Modo de prueba de realimentación análoga

Este bit configura el modo de prueba de realimentación análoga. Note que en este modo ambos registros de modo de transmisión y recepción deben ser colocados en el mismo tipo de MODEM y banda o tasa de bits.

b11 = 1	Modo de realimentación local análoga habilitado
b11 = 0	Sin realimentación (operación normal de modem)

BIT 10 Registro General de control: Ecualizadores en TX y Rx

Este bit permite que los ecualizadores de compromiso fijo en el bloque de filtrado del receptor y el transmisor sean deshabilitados.

b10 = 1	Deshabilita los ecualizadores
b10 = 0	Habilita los ecualizadores (modos de 600, 1200 o 2400 bps)

Bit 9 Registro General de Control: Manejo del relé

Este bit controla directamente el pin de salida RDRVN.

b9 = 1	RDRVN pin de salida colocado a Vss
b9 = 0	RDRVN pin de salida colocado a Vdd

BIT 8 Registro General de Control: Powerup

Este bit controla la alimentación interna de la mayoría de los circuitos del chip, incluyendo el oscilador del cristal y el voltaje Vbías. Note que el comando general de reset clarea este bit, colocando el dispositivo en modo de bajo consumo de potencia.

b8 = 1	Dispositivo alimentado normalmente
b8 = 0	Modo de bajo consumo de potencia (todos los circuitos excepto el de detección de timbre, RDRVN y C-BUS esta deshabilitados)

Cuando la alimentación es aplicada por primera vez al dispositivo, el siguiente procedimiento de powerup debe ser llevado a cabo para asegurar su correcta operación.

i. (Alimentación es aplicada al circuito)
ii. Emite un comando general de reset
iii. Escribir el registro general de control (Dirección $E0) colocando tanto el bit de powerup (b8) como el bit de reset (b7) en uno –dejar en este estado por un mínimo de 20ms- esto es necesario para que el cristal corra inicialmente durante este tiempo permitiendo que la lógica interna alcance un estado definido. El dispositivo esta ahora energizado con el cristal y el voltaje Vbías operando, pero sin ninguna función de recepción o transmisión.
iv. El dispositivo esta ahora listo para ser programado cuando y como se desee.

Bit 7 Registro General de Control: Reset

Colocando este bit en uno se resetea la circuiteria interna del CMX868, clareando todos los bits de los registros de modo de transmisión y recepción y los bits 13-0 del registro de estados.

b7 = 1	Circuiteria interna en condición de reset
b7 = 0	Operación normal

Bit 6 Registro General de Control: IRQNEN (IRQ O/P Enable)

Colocando este bit en uno habilita el pin de salida IRQN.

b6 = 1	El pin IRQN es llevado a Vss si el bit IRQ es uno
b6 = 0	El pin IRQN es deshabilitado (alta impedancia)

Bits 5-0 Registro General de Control: Bits de enmascaramiento de IRQ

Esto bits afectan la operación del bit de IRQ en el registro de estados.

3.4.3. Registro de modo de transmisión: 16-bit solo escritura. Dirección C-BUS $E1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modo de Tx=modem				Nivel de Tx			Tono de guarda		scrambler		Datos Sinc/ Start-stop		# de bits de datos/ fuente de datos sin		
Modo de TX=DTMF/Tonos				Nivel de Tx			Sin uso colocar en 0000				Selección de tonos o DTMF				
Modo de Tx=deshabilitado				Colocar en 0000 0000 0000											

Tabla 10. Registro de modo de transmisión.

Bits 15-12 Registro de modo de TX: Modo de operación

Estos cuatro bits seleccionan el modo operativo de transmisión.

b15	b14	b13	b12		
1	1	1	1	V.22 bis 2400 bps QAM	Banda alta (respuesta modem)
1	1	1	0	"	Banda baja (llamada modem)
1	1	0	1	V.22/Bell 212A 1200 bps	Banda alta (respuesta modem)
1	1	0	0	"	Banda baja (llamada modem)
1	0	1	1	V.22 600 bps DPSK	Banda alta (respuesta modem)
1	0	1	0	"	Banda baja (llamada modem)
1	0	0	1	V.21 300 bps FSK	Banda alta (respuesta modem)
1	0	0	0	"	Banda baja (llamada modem)
0	1	1	1	Bell 103 300 bps FSK	Banda alta (respuesta modem)
0	1	1	0	"	Banda baja (llamada modem)
0	1	0	1	V.23 FSK	1200 bps
0	1	0	0	"	75 bps
0	0	1	1	Bell 202 FSK	1200 bps
0	0	1	0	"	150 bps
0	0	0	1	DTMF/Tonos	
0	0	0	0	Transmisor deshabilitado	

Bits 11-9 Registro de modo de TX: Nivel de TX

Estos tres bits ajustan la ganancia del bloque de control del nivel de TX

b11	b10	b9	
1	1	1	0dB
1	1	0	-1.5dB
1	0	1	-3.0dB
1	0	0	-4.5dB
0	1	1	-6.0dB
0	1	0	-7.5dB
0	0	1	-9.0dB
0	0	0	-10.5dB

Bits 8-7 Registro de modo de TX: Tono de guarda (modos QAM y DPSK)

Estos dos bits seleccionan el tono de guarda a ser transmitido junto con las señales de QAM o DPSK. Colocar ambos bits a cero en el modo de FSK.

b8	b7	
1	1	Tono de guarda de 550Hz en Tx
1	0	Tono de guarda de 1800Hz en Tx
0	x	Sin tono de guarda

Bits 6-5 Registro de modo de TX: Enmascarador (modos QAM y DPSK)

Estos dos bits controlan la operación del enmascarador de TX usado en los modos QAM y DPSK. Colocar ambos bits a cero en el modo de FSK.

b6	b5	
1	1	Enmascarador habilitado, circuito de deteccion de 64 unos hab.
1	0	Enmascarador habilitado, circuito de deteccion de 64 unos deshab.
0	x	Enmascarador deshabilitado

Bits 4-3 Registro de modo de TX: Formato de datos (modos QAM, DPSK y FSK)

Estos dos bits seleccionan entre el modo Start-Stop y el modo síncrono, y la adición de un bit de paridad para transmitir caracteres en el modo de Start-Stop.

b4	b3	
1	1	Modo síncrono
1	0	Modo Start-Stop, sin paridad
0	1	Se adiciona un bit de paridad par a los datos seleccionados
0	0	Se adiciona un bit de paridad impar a los datos seleccionados

Bits 2-0 Registro de modo de TX: Bits de datos y parada (modos Start-Stop)

En los modos de Start-Stop estos tres bits seleccionan el número de datos a transmitir y la cantidad de bits de parada.

b2	b1	b0	
1	1	1	8 bits de datos, 2 bits de parada
1	1	0	8 bits de datos, 1 bit de parada
1	0	1	7 bits de datos, 2 bits de parada
1	0	0	7 bits de datos, 1 bit de parada
0	1	1	6 bits de datos, 2 bits de parada
0	1	0	6 bits de datos, 1 bit de parada
0	0	1	5 bits de datos, 2 bits de parada
0	0	0	5 bits de datos, 1 bit de parada

Bits 2-0 Registro de modo de TX: fuente de los datos (modos síncrono)

En los modos síncronos (bits 4-3 = 11) estos tres bits seleccionan la fuente de los datos que se envían al modulador FSK o QAM/DPSK.

b2	b1	b0	
1	x	x	Bits de datos desde el buffer de datos de Tx
0	1	1	Unos continuos
0	1	0	Ceros continuos
0	0	x	Patrones dibits S1 continuos '00,11' en modo de cuelgue V.22 bis en modo QAM y DPSK, unos y ceros alternos en los otros modos.

Bits 8-0 Registro de modo de TX: modo de Tonos/DTMF

Si el modo de transmisión de tonos/DTMF ha sido seleccionado (b15-12 = 0001) entonces b8-5 deben ser colocados en 0000 y b4-0 seleccionaran una señal DTMF o un tono fijo o uno de cuatro tonos programados o par de tonos para transmisión.

b4 = 0: Tono fijo o par de tonos programados

b3	b2	b1	b0	Frecuencia del tono (Hz)	
0	0	0	0	Sin tono	
0	0	0	1	697	
0	0	1	0	770	
0	0	1	1	852	
0	1	0	0	941	
0	1	0	1	1209	
0	1	1	0	1336	
0	1	1	1	1477	
1	0	0	0	1633	
1	0	0	1	1300	(tono de llamada)
1	0	1	0	2100	(tono de respuesta)
1	0	1	1	2225	(tono de respuesta)
1	1	0	0	Par de tonos TA	Par o tono programado (1.5.10.8)
1	1	0	1	Par de tonos TB	"
1	1	1	0	Par de tonos TC	"
1	1	1	1	Par de tonos TD	"

b4 = 1: DTMF

b3	b2	b1	b0	Frecuencia baja	Frecuencia alta	Símbolo de teclado
0	0	0	0	941	1633	D
0	0	0	1	697	1209	1
0	0	1	0	697	1336	2
0	0	1	1	697	1477	3
0	1	0	0	770	1209	4
0	1	0	1	770	1336	5
0	1	1	0	770	1477	6
0	1	1	1	852	1209	7
1	0	0	0	852	1336	8
1	0	0	1	852	1477	9
1	0	1	0	941	1336	0
1	0	1	1	941	1209	*
1	1	0	0	941	1477	#
1	1	0	1	697	1633	A
1	1	1	0	770	1633	B
1	1	1	1	852	1633	C

3.4.4. Registro de modo de recepción: 16-bit solo escritura. Dirección C-BUS $E2

Este registro controla el tipo y nivel de la señal recibida por el CMX868. Todos los bits de este registro son llevados a cero mediante el comando general de reset, o cuando b7 (reset) del registro de control general es uno.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modo de Rx=modem				Nivel de Rx			eq	descrambler		Sincro/ Start-stop			# de bits de datos y paridad		
Modo de RX=Detec. Tonos				Nivel de Rx			Selección de progreso de llamada/tonos/DTMF								
Modo de Rx=deshabilitado				Colocar en 0000 0000 0000											

Tabla 11. Registro de modo de recepción.

Bits 15-12 Registro de modo de Rx: Modo de operación

Estos cuatro bits seleccionan el modo operativo de recepción.

b15	b14	b13	b12		
1	1	1	1	V.22 bis 2400 bps QAM	Banda alta (llamada modem)
1	1	1	0	"	Banda baja (respuesta modem)
1	1	0	1	V.22/Bell 212A 1200 bps	Banda alta (llamada modem)
1	1	0	0	"	Banda baja (respuesta modem)
1	0	1	1	V.22 600 bps DPSK	Banda alta (llamada modem)
1	0	1	0	"	Banda baja (respuesta modem)
1	0	0	1	V.21 300 bps FSK	Banda alta (llamada modem)
1	0	0	0	"	Banda baja (respuesta modem)
0	1	1	1	Bell 103 300 bps FSK	Banda alta (llamada modem)
0	1	1	0	"	Banda baja (respuesta modem)
0	1	0	1	V.23 FSK	1200 bps
0	1	0	0	"	75 bps
0	0	1	1	Bell 202 FSK	1200 bps
0	0	1	0	"	150 bps
0	0	0	1	Detección de progreso de llamada, respuesta, par de tonos, DTMF	
0	0	0	0	Receptor deshabilitado	

Bits 11-9 Registro de modo de Rx: Nivel de Rx

Estos tres bits ajustan la ganancia del bloque de control del nivel de Rx

b11	b10	b9	
1	1	1	0dB
1	1	0	-1.5dB
1	0	1	-3.0dB
1	0	0	-4.5dB
0	1	1	-6.0dB
0	1	0	-7.5dB
0	0	1	-9.0dB
0	0	0	-10.5dB

Bits 8 Registro de modo de Rx: Auto-ecualizador (modos QAM y DPSK)

Este bit controla la operación del auto-ecualizador del receptor QAM/DPSK. Colocar en cero en el modo de FSK. Colocar en uno en el modo QAM a 2400bps.

b8 = 1	Habilitar auto-ecualizador
b8 = 0	Modo DPSK: Auto-ecualizador deshabilitado Modo QAM: configuración del Auto-ecualizador congelada

Bits 7-6 Registro de modo de Rx: Enmascarador (modos QAM y DPSK)

Estos dos bits controlan la operación del desenmascarador de Rx usado en los modos QAM y DPSK. Colocar ambos bits a cero en el modo de FSK.

b7	b6	
1	1	Desenmascarador habilitado, circuito de deteccion de 64 unos hab.
1	0	Desenmascarador habilitado, circuito de deteccion de 64 unos des.
0	x	Desenmascarador deshabilitado

Bits 5-3 Registro de modo de Rx: Configuración USART (QAM, DPSK y FSK)

b5	b4	b3	
1	1	1	Modo síncrono
1	1	0	Modo Start-Stop, sin sobrevelocidad
1	0	1	Start-Stop, +1% sobrevelocidad (perdida de 1 en 8 bits parada)
1	0	0	Start-Stop, +2.3% sobrevelocidad (perdida de 1 en 4 bits parada)
0	x	x	Funcion USART deshabilitada

Bits 2-0 Registro de modo de Rx: Bits de datos y paridad (modos Start-Stop)

En los modos de Start-Stop estos tres bits seleccionan el número de datos (más cualquier bit de paridad) en cada carácter recibido. Estos bits se ignoran en el modo síncrono.

b2	b1	b0	
1	1	1	8 bits de datos + paridad
1	1	0	8 bits de datos
1	0	1	7 bits de datos + paridad
1	0	0	7 bits de datos
0	1	1	6 bits de datos + paridad
0	1	0	6 bits de datos
0	0	1	5 bits de datos + paridad
0	0	0	5 bits de datos

Bits 2-0 Registro de modo de Rx: modo de detección de tonos

En el modo de detección de tonos (b15-12 = 0001) b8-3 deben ser colocados en 000000, bits 2-0 seleccionan el tipo de detector.

b2	b1	b0	
1	0	0	Deteccion de par de tonos programados
0	1	1	Detección de progreso de llamada
0	1	0	Detección de tono de respuesta de 2100, 2225Hz
0	0	1	Detección DTMF
0	0	0	Deshabilitado

3.4.5. Registro de datos TX: 8-bit solo escritura. Dirección C-BUS $E3 y $E4

7	6	5	4	3	2	1	0
Bits de datos a ser transmitidos							

Tabla 12. Registros de datos de transmisión.

En el modo de datos de transferencia síncrona este registro contiene los siguientes 8 bits de datos a ser transmitidos. El bit 0 se transmite primero. En el modo Start-Stop el número especificado de bits de datos será transmitido desde este registro (b0 primero). Un bit de inicio, bit de paridad (si se requiere) y bit(s) de parada serán adicionados automáticamente. Este registro solo debe ser escrito cuando el bit de datos listo del registro de estados sea uno. La dirección C-BUS $E3 es normalmente usada, $E4 es para implementar el requerimiento V.14 de sobré velocidad de transmisión en el modo Start-Stop.

3.4.6. Registro de datos Rx: 8-bit solo lectura. Dirección C-BUS $E5

7	6	5	4	3	2	1	0
Bits de datos a recibidos							

Tabla 13. Registros de datos de recepción.

En el modo de datos sin formatear este registro contiene los 8 bits de datos recibidos, b0 contiene el primer bit recibido, b7 el último. En el modo de datos Start-Stop este registro contiene el número especificado de bits de datos de un carácter recibido, b0 contiene el primer bit recibido.

3.4.7. Registro de estados: 16-bit solo lectura. Dirección C-BUS $E6

Bits 13-0 de este registro son llevados a cero mediante un comando general de reset, o cuando b7 (reset) del registro general de control sea uno.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ	RD	PF	Ver a continuación el uso de estos bits												

Tabla 14. Registros de estados.

El significado de los bits 12-0 del registro de estados depende de si el circuito receptor esta en modo de MODEM o detección de tono.

	Modos de MODEM Rx	Modos de detección de tonos Rx	** IRQ Mask bits
b15	IRQ		
b14	Se coloca en 1 en detección de timbre		b5
b13	Bit bandera de programación		b4
b12	Se coloca en 1 cuando hay datos listos en Tx Se borra cuando se escribe el registro de datos de Tx		b3
b11	Se coloca en 1 cuando hay datos faltantes en Tx Se borra cuando se escribe el registro de datos de Tx		b3
b10	1 cuando se detecta energía en banda de señal Rx del MODEM	1 cuando se detecta energía en la banda de progreso de llamada o cuando se detectan ambos tonos programables	b2
b9	1 cuando el patrón S1 es detectado en modos DPSK o QAM, o cuando el patrón '1010..' es detectado en FSK.	0	b1
b8	Ver la siguiente tabla.	0	b1
b7	Ver la siguiente tabla.	1 cuando tono de respuesta de 2100hz o segundo tono programado es detectado.	b1
b6	Pasa a 1 cuando hay datos listo en Rx. Se borra leyendo el registro de datos de Rx.	1 cuando tono de respuesta de 2225hz o primer tono programado es detectado.	b0
b5	Pasa a 1 cuando hay desborda-miento de datos en Rx. Se borra leyendo el registro de datos de Rx.	1 cuando un código dtmf es detectado.	b0
b4	Pasa a 1 cuando hay error de cuadros en Rx.	0	-
b3	Pasa a 1 con paridad par en Rx.	Bit 3 código DTMF Rx, ver tabla	-
b2	Bit 2 calidad de señal Rx.	Bit 2 código DTMF Rx.	-
b1	Bit 1 calidad de señal Rx.	Bit 1 código DTMF Rx.	-
b0	Bit 0 calidad de señal Rx QAM/DPSK. Salida de frecuencia del modulador FSK.	Bit 0 código DTMF Rx.	

Tabla 15. Bits de registro de estados.

Nota: Esta columna muestra la correspondencia de los bits de enmascaramiento de IRQ en el registro general de control. Una transición de 0 a 1 en cualquiera de los bits 14-5 del registro de estados puede causar que el bit 15 se coloque en uno, si el correspondiente bit de enmascaramiento de IRQ es uno. El bit de IRQ es

borrado mediante la lectura del registro de estados, mediante un comando general de reset o colocando los bits 7 y 8 del registro general de control en 1. Una falsa detección de ceros continuos puede ser generada dentro de los 4mseg de cambio del registro de modos de recepción a cualquiera de los modos QAM o DPSK.

B8	B7	**Descrambler deshabilitado**	**Descrambler habilitado (solo modos QAM/DPSK)**
1	1	"	1s continuos enmascarados (ver nota)
1	0	0s continuos desenmascarados	0s continuos enmascarados
0	1	1s continuos desenmascarados	1s continuos desenmascarados
0	0	"	"

Cuando el desenmascarador es deshabilitado entonces la detección de 1s continuos desenmascarados puede inhibir el detector de 1s continuos enmascarados.

El pin de salida IRQN será llevado a bajo (a Vss) cuando el bit de IRQ del registro de estados y el bit IRQNEN (b6) del registro general de control son ambos 1. Cambios en los bits del registro de estado causados por el cambio del modo de operación de TX o Rx pueden demorarse por encima de 140µs para hacer efecto. En el modo de bajo consumo de potencia o cuando el bit de reset (b7) del registro general de control es 1 el bit de detector de timbre (b14) continua operando.

Los detectores de 0s o 1s continuos la señal Rx después del desenmascarador QAM/DPSK, y por lo tanto detectaran 1s o 0s continuos si el desenmascarador es deshabilitado, o 1s o 0s continuos enmascarados si el desenmascarador esta habilitado.

En los modos Rx de MODEM QAM/DPSK los bits 2-0 del registro de estados contienen un valor que indica el BER de la señal recibida, ver figura 11. En los modos Rx de MODEM FSK bits 2 y 1 serán cero y b0 demostrara la salida del demodulador de frecuencia, actualizado a 8 veces la tasa nominal de datos.

b3	b2	b1	b0	Frecuencia baja	Frecuencia alta	Símbolo de teclado
0	0	0	0	941	1633	D
0	0	0	1	697	1209	1
0	0	1	0	697	1336	2
0	0	1	1	697	1477	3
0	1	0	0	770	1209	4
0	1	0	1	770	1336	5
0	1	1	0	770	1477	6
0	1	1	1	852	1209	7
1	0	0	0	852	1336	8
1	0	0	1	852	1477	9
1	0	1	0	941	1336	0
1	0	1	1	941	1209	*
1	1	0	0	941	1477	#
1	1	0	1	697	1633	A
1	1	1	0	770	1633	B
1	1	1	1	852	1633	C

Tabla 16. Código DTMF recibido: b3-0 del registro de estados

3.4.8. Registro de programación: 16-bit solo escritura. Dirección C-BUS $E8

Este registro es usado para programar los pares de tonos programados para recepción y transmisión mediante la escritura de valores apropiados a direcciones de RAM dentro del CMX868. Note que estas direcciones RAM son borradas mediante el modo de bajo consumo de potencia o el reset. El registro de programación solo debe ser escrito cuando el bit bandera de programación del registro de estados sea 1. La acción de escribir el registro de programación borra este bit bandera. Cuando la acción de programación ha sido completada (normalmente dentro de 150µs) el CMX868 devolverá este bit a 1. Cuando se programan los pares de tonos de recepción o transmisión, no se deben cambiar los registros de modo de recepción o transmisión hasta que se complete la programación y el bit bandera de programación haya retornado a 1.

3.5. ALIMENTACION

Para la alimentación del sistema de control de rondas de seguridad se ha optado por utilizar una fuente fija de 5VDC.

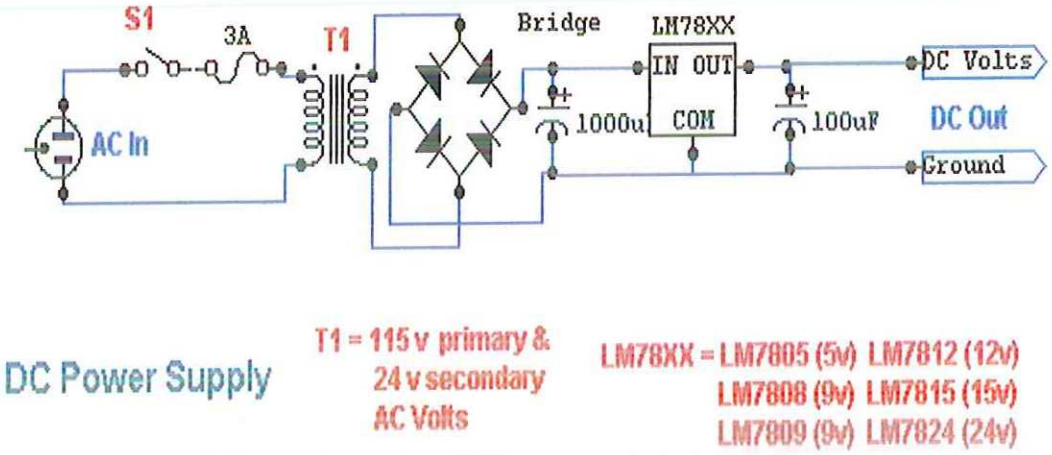


Figura 24. Interfaz de la alimentación del circuito

ANEXOS

DISEÑO DE SOTWARE

4.1 SUBRUTINA PARA EL MANEJO DEL PROTOCOLO I2C

ESTE PROTOCOLO ES DE MUCHA IMPORTANCIA PORQUE CON ESTE SE HACE POSIBLE LA COMUNICACIÓN CON OTROS HARDWARE (RTC, MODEM, MEMORIAS, ECT); ESTE PROTOCOLO CUENTA DE DOS SALIDAS QUE SON UNA PARA RECIBIR Y TRANSMITIR LOS DATOS Y OTRA QUE ES UN PULSO DE RELOJ. EN LA APLICACIÓN SE VE CONFIGURANDO EL RTC (RELOJ EN TIEMPO REAL).

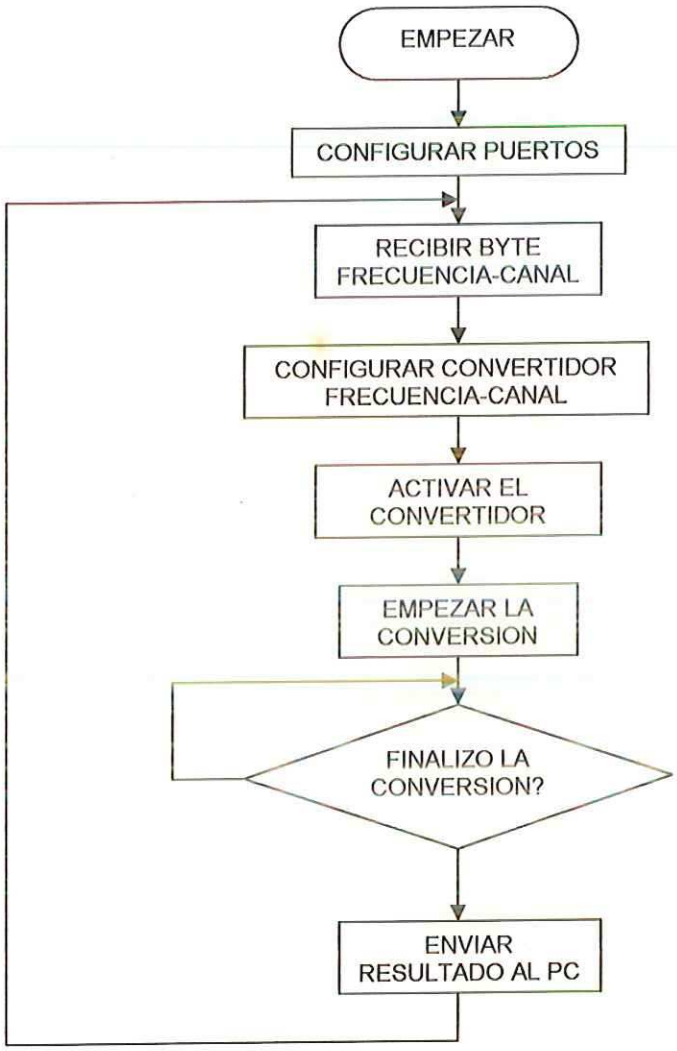
```
**********RUTINAS DE MANEJO DEL DS1307**********
Bitstart  Bset    SDA, Ddre
          Bclr    SCL, Porte
          Bset    SDA, Porte
          Jsr     Small
          Bset    SCL, Porte
          Jsr     Small
          Bclr    SDA, Porte
          Jsr     Small
          Bclr    SCL, Porte
          Jsr     Small
          Rts

Bitstop   Bset    SDA, Ddre
          Bclr    SCL, Porte
          Bclr    SDA, Porte
          Jsr     Small
          Bset    SCL, Porte
          Jsr     Small
          Bset    SDA, Porte
          Jsr     Small
          Bclr    SCL, Porte
          Jsr     Small
          Rts

Escribir  Jsr     Bitstart
          Jsr     Small
          Bclr    0, ESCLAVO
          Lda     ESCLAVO
          Sta     TXBUF
          Jsr     Tx
          Lda     DIRECCION
```

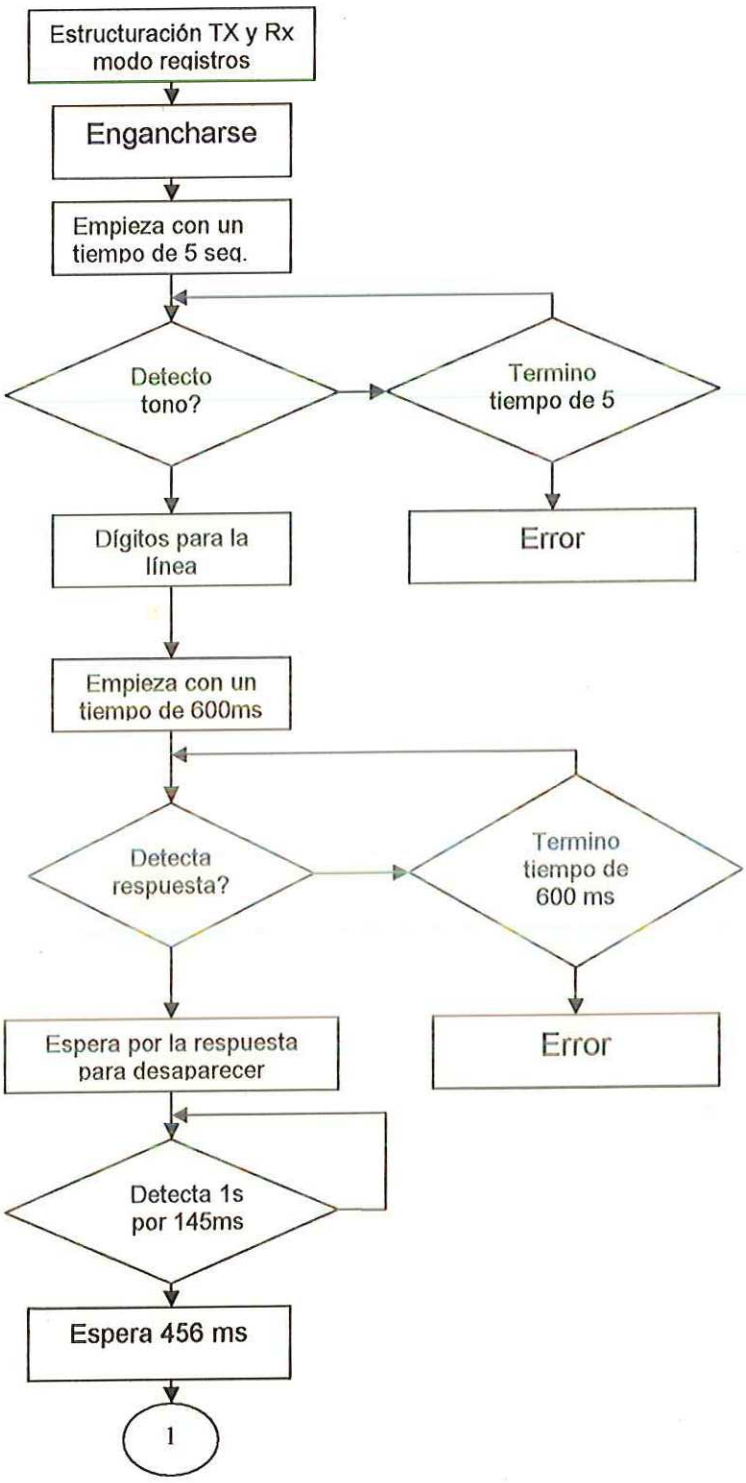
```
        Sta     TXBUF
        Jsr     Tx
        Lda     INFO
        Sta     TXBUF
        Jsr     Tx
        Jsr     Bitstop
        Jsr     Big
        Rts

Leer    Jsr     Bitstart
        Jsr     Small
        Bclr    0, ESCLAVO
        Lda     ESCLAVO
        Sta     TXBUF
        Jsr     Tx
        Lda     DIRECCION
        Sta     TXBUF
        Jsr     Tx
        Jsr     Small
        Jsr     Bitstart
        Jsr     Small
        Bset    0, ESCLAVO
        Lda     ESCLAVO
        Sta     TXBUF
        Jsr     Tx
        Jsr     Small
        Jsr     Rx
        Jsr     Bitstop
        Rts

Tx      Mov     #08t, CONTA
Tx1     Rol     TXBUF
        Bcc     Bajo
Alto    Bset    SDA, Ddre
        Bset    SDA, Porte
        Jsr     Output
        Dbnz    CONTA, Tx1
        Jmp     Ack
Bajo    Bset    SDA, Ddre
        Bclr    SDA, Porte
        Jsr     Output
        Dbnz    CONTA, Tx1
Ack     Jsr     Input
        Rts
Rx      Mov     #08t, CONTA
        Clr     RXBUF
Rx1     Jsr     Input
        Rol     RXBUF
        Dbnz    CONTA, Rx1
        Bset    SDA, Ddre
        Bclr    SDA, Porte
        Jsr     Output
        Rts

Output  Jsr     Small
        Bset    SCL, Porte
```

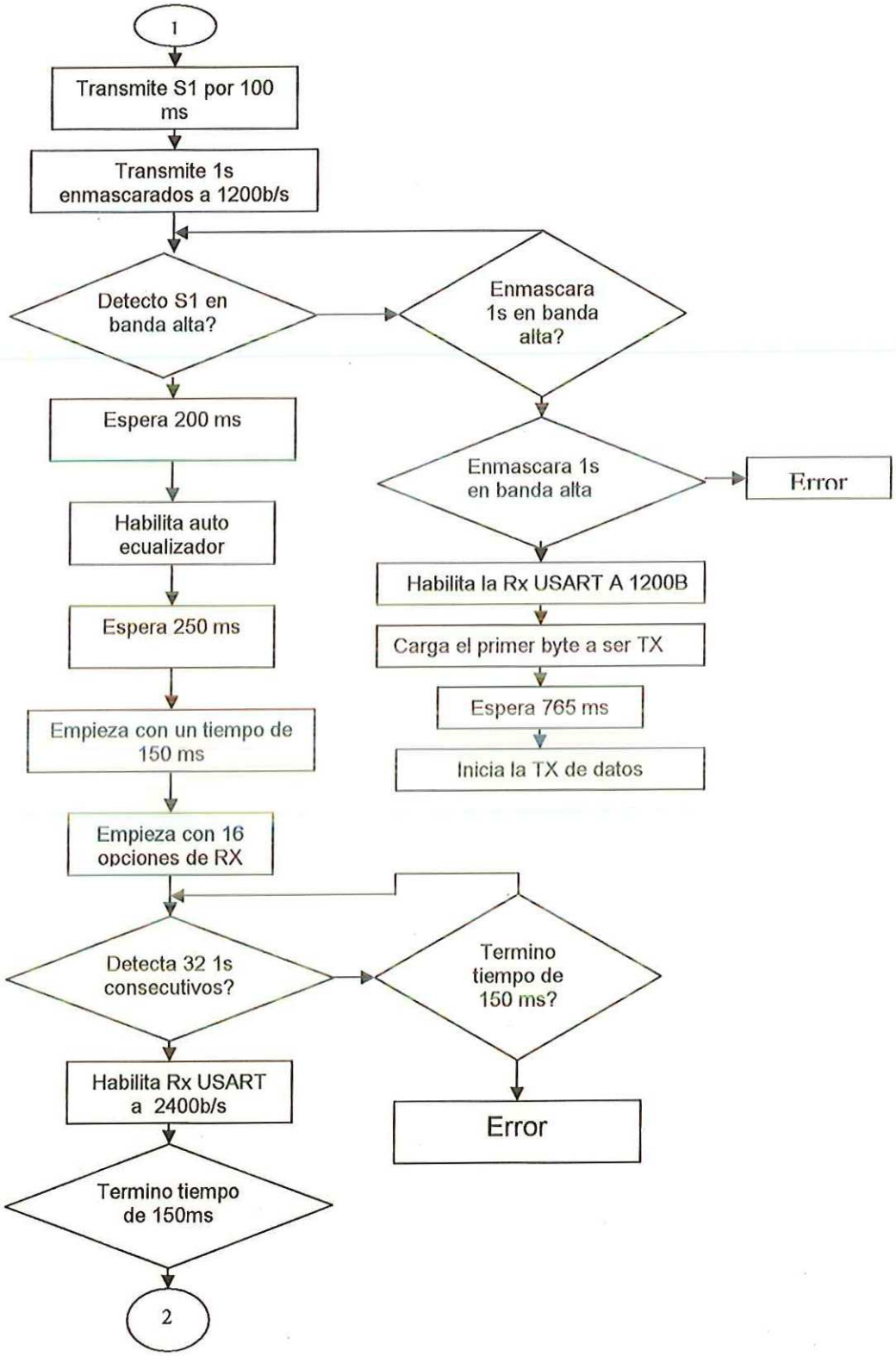
```
        Jsr     Small
        Bclr    SCL, Porte
        Jsr     Small
        Rts

Input   Bclr    SDA, Ddre
        Jsr     Small
        Jsr     Small
        Bset    SCL, Porte
        Jsr     Small
        Jsr     Small
        Jsr     Small
        Brclr   SDA, Porte, Abajo
        Sec
        Bclr    SCL, Porte
        Jsr     Small
        Rts
Abajo   Clc
        Bclr    SCL, Porte
        Jsr     Small
        Rts

Small   Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Nop
        Rts

Big     Mov     #255t, VAR
Big1    Jsr     Small
        Dbnz    VAR, Big1
        Rts
```

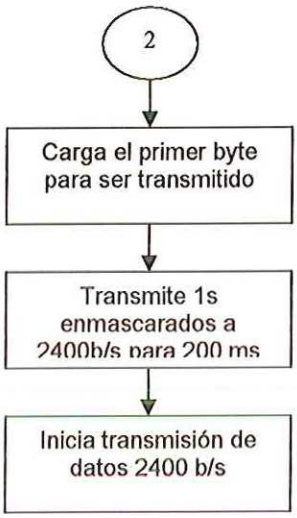
4.2 DIGRAMA DE FLUJO Y RUTINAS DE MANEJO MODEM(CMX868)

4.2.1 Diagrama de flujo para la inicialización del modem cmx868 para la tx




Estructuración TX y Rx modo registros
Engancharse
Empieza con un tiempo de 5 seg.
Detecto tono?
Termino tiempo de 5
Dígitos para la línea
Error
Empieza con un tiempo de 600ms
Detecta respuesta?
Termino tiempo de 600 ms
Espera por la respuesta para desaparecer
Error
Detecta 1s por 145ms
Espera 456 ms
1


1
Transmite S1 por 100 ms
Transmite 1s enmascarados a 1200b/s
Detecto S1 en banda alta?
Enmascara 1s en banda alta?
Espera 200 ms
Enmascara 1s en banda alta
Error
Habilita auto ecualizador
Habilita la Rx USART A 1200B
Espera 250 ms
Carga el primer byte a ser TX
Espera 765 ms
Empieza con un tiempo de 150 ms
Inicia la TX de datos
Empieza con 16 opciones de RX
Detecta 32 1s consecutivos?
Termino tiempo de 150 ms?
Habilita Rx USART a 2400b/s
Error
Termino tiempo de 150ms
2



➢ PROGRAMA DE TX PARA EL MODEM CMX 868

EN ESTE PROGRAMA SE DESARROLLA EL DIAGRAMA DE FLUJO MOSTRADO EN LA SECCION ANTERIOR APARTE DE ESTO EN ESTE PROGRAMA SE MUESTRA EN DETALLE LA INTERACCION CON OTROS DISPOSITIVOS.

```
modemwork    jsr     resetmodem          ;1. secuencia de encendido
             jsr     ret20ms
iniciomodem  bset    back,portd
             clr     flags1
             jsr     startmodem
             jsr     setupreg            ;2. configuracion de registros tx y rx
             jsr     takeoffhook         ;3. take off-hook
             mov     #$07,tabla          ;clear line 2
             jsr     printline2
             jsr     dialtonedet         ;4. dial tone detect(before 5s timer expires)
             brclr   tonedet,flags1,iniciomodem
             mov     #$36,t1sc
             bset    back,portd
             mov     #$31,caracter       ;si encontro tono en la linea
             jsr     printchar2
             jsr     digtoline           ;5. dial digits to line (dtmf)
             mov     #$32,caracter       ;marcado
             jsr     printchar2
             jsr     detectans           ;6. detect answertone
             brclr   ansdet,flags1,finmodem
             mov     #$33,caracter       ;si detecto tonos de reputa
             jsr     printchar2
             jsr     det1suns            ;8. detect unscrambled 1s for 145 ms, 456ms
```

```
        brclr   onesundet,flags1,nounosdes
        mov     #$34,caracter           ;si detecto unos desenmascarados
        jsr     printchar2
        jsr     ret100ms                ;9. transmit s1 for 100 ms at 1200 b/s
        jsr     transscr1st             ;10. transmit scrambled 1s at 1200 b/s
        mov     #$35,caracter           ;envio unos enmascarados
        jsr     printchar2
        jsr     detoneshb               ;11. detect scrambled 1s in high band
        brclr   scrambled1s,flags1,nounosscr
        mov     #$37,caracter           ;recepcion de unos enmascarados
        jsr     printchar2
        jsr     check1s270              ;12. wait 270 ms
        jsr     enrxusartt              ;13. enable rx usart
        mov     #$38,caracter           ;espero 270 ms y cargo el rx usart
        jsr     printchar2
        jsr     txdatareadyi            ;14. load first data byte to be transmited
        jsr     loadfirstt
        mov     #$39,caracter           ;cargo el primer data byte
        jsr     printchar2
        jsr     ret765ms                ;15. wait for 765 ms
        jsr     entxusartt              ;start data transmission at 1200 b/s
        jsr     ret3300ms
        jsr     txdatareadyi            ;14. load first data byte to be transmited
        jsr     loadfirstt
        mov     #$41,caracter           ;cargo el primer data byte
        jsr     printchar2
        jsr     ret3300ms
        jsr     sendtrama
        jsr     ret2150ms
        mov     #$42,caracter           ;cargo el primer data byte
        jsr     printchar2
        jsr     rxdataready
        brclr   confirm1,flags1,finmodem
        jsr     recibedato
        lda     bytehigh
        sta     caracter
        jsr     printchar2
        bra     finmodem

finmodem    jsr     resetmoreg
        mov     #$36,t1sc
        mov     #$46,t1sc
        brclr   tof,t1sc,*              ;wait until the time is complete
        brclr   confirm1,flags1,endperfect
        jsr     iniciomodem

nounosdes   mov     #$0a,tabla
        jsr     printline2
        bra     finmodem

nounosscr   mov     #$0b,tabla
        jsr     printline2
        bra     finmodem

endperfect  bra     endperfect
```

SUBRUTINAS VARIAS PARA EL MANEJO DEL MODEM TX

```
txmobyte    mov     #$08,cont
            lda     bytemodem
rotbyte     rola
            bclr    serialclk,portc
            bcc     cero
uno         bset    comdata,portc
            bra     follow
cero        bclr    comdata,portc
follow      jsr     delaymo
            bset    serialclk,portc
            jsr     delaymo
            dec     cont
            beq     endtxmobyte
            bra     rotbyte
endtxmobyte rts

rxmobyte    clr     bytemodem
            mov     #$08,cont
            clc
rotar       rol     bytemodem
            bclr    serialclk,portc
            brset   repdata,portc,recuno
reccero     bclr    0,bytemodem
            bra     next
recuno      bset    0,bytemodem
next        jsr     delaymo
            bset    serialclk,portc
            jsr     delaymo
            dec     cont
            beq     endrxmobyte
            bra     rotar
endrxmobyte rts

modaddress  bclr    csn,portc
            jsr     delaymo
            lda     admobyte
            sta     bytemodem
            jsr     txmobyte
            jsr     delaymo
            bset    csn,portc
endmodaddress rts

modabtx     bclr    csn,portc
            jsr     delaymo
            lda     admobyte
            sta     bytemodem
            jsr     txmobyte
            lda     bytehigh
            sta     bytemodem
            jsr     txmobyte
            jsr     delaymo
            bset    csn,portc
endmodabtx  rts
```

```
moda2btx        bclr    csn,portc
                jsr     delaymo
                lda     admobyte
                sta     bytemodem
                jsr     txmobyte
                lda     bytehigh
                sta     bytemodem
                jsr     txmobyte
                lda     bytelow
                sta     bytemodem
                jsr     txmobyte
                jsr     delaymo
                bset    csn,portc
endmoda2btx     rts

modabrx         bclr    csn,portc
                jsr     delaymo
                lda     admobyte
                sta     bytemodem
                jsr     txmobyte
                jsr     rxmobyte
                lda     bytemodem
                sta     bytehigh
                jsr     delaymo
                bset    csn,portc
endmodabrx      rts

moda2brx        bclr    csn,portc
                jsr     delaymo
                lda     admobyte
                sta     bytemodem
                jsr     txmobyte
                jsr     rxmobyte
                lda     bytemodem
                sta     bytehigh
                jsr     rxmobyte
                lda     bytemodem
                sta     bytelow
                jsr     delaymo
                bset    csn,portc
endmoda2brx     rts

resetmodem      mov     #$01,admobyte
                jsr     modaddress
endresetmodem   rts

resetmoreg      mov     #$e0,admobyte
                mov     #$03,bytehigh
                mov     #$80,bytelow
                jsr     moda2btx
endresetmoreg   rts

startmodem      mov     #$e0,admobyte           ;this subroutine is equal to colgar
                mov     #$03,bytehigh
                mov     #$00,bytelow
                jsr     moda2btx
```

```
endstartmodem     rts

setupreg      mov     #$e1,admobyte
              mov     #$1e,bytehigh          ;select dtmf / tones mode
              mov     #$00,bytelow           ;user defined tx level -  no tones
              jsr     moda2btx
              mov     #$e2,admobyte
              mov     #$1e,bytehigh          ;select dtmf / tones mode
              mov     #$03,bytelow           ;deteccion de progreso de llamada
              jsr     moda2btx
endsetupreg   rts

takeoffhook   mov     #$e0,admobyte
              mov     #$01,bytehigh          ;relay drive pin is pulled to vdd
              mov     #$00,bytelow
              jsr     moda2btx
endtakeoffhook rts

dialtonedet   mov     #$36,t1sc
              clr     t2sc
              bset    tstop,t2sc
              mov     #$06,contt             ;temporizado de 5s
              mov     #$fa,t2modh
              mov     #$00,t2modl
              bset    ps2,t2sc
              bclr    tstop,t2sc
ocur          mov     #$e6,admobyte
              jsr     moda2brx
              brset   2,bytehigh,sihaytono
              brclr   tof,t2sc,ocur
              bclr    tof,t2sc
              dbnz    contt,ocur
              bset    tstop,t2sc
              mov     #$08,tabla
              jsr     printline2
              mov     #$46,t1sc
              brclr   tof,t1sc,*     ;wait until the time is complete
              bclr    tonedet,flags1
              bra     enddialtonedet
sihaytono     bclr    tof,t2sc
              bset    tstop,t2sc
              bset    tonedet,flags1
enddialtonedet rts

digtoline     jsr     ret100ms
              mov     #$11,digitdial
              jsr     marcar
              mov     #$18,digitdial
              jsr     marcar
enddigtoline  rts

marcar        mov     #$e1,admobyte
              mov     #$1e,bytehigh
              lda     digitdial
              sta     bytelow
              jsr     moda2btx
```

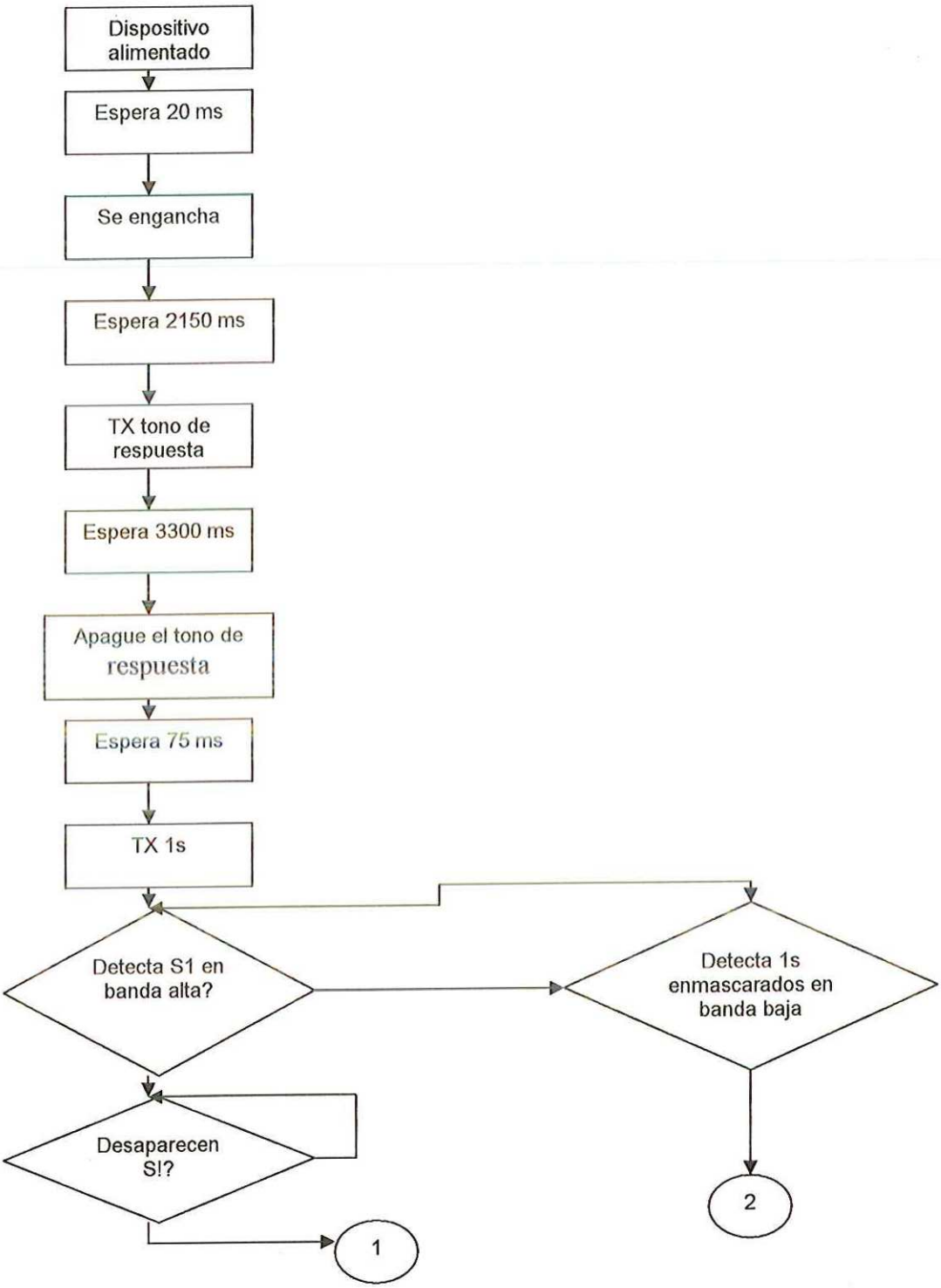
```
        jsr     ret765ms
        mov     #$e1,admobyte
        mov     #$1e,bytehigh
        mov     #$00,bytelow
        jsr     moda2btx
        jsr     ret765ms
endmarcar       rts

detectans   mov     #$36,t1sc
        mov     #$e2,admobyte
        mov     #$1e,bytehigh
        mov     #$02,bytelow
        jsr     moda2btx
        clr     t2sc
        bset    tstop,t2sc          ;temporizado de  s
        mov     #$0c,contt
        mov     #$fa,t2modh
        mov     #$00,t2modl
        bset    ps2,t2sc
        bclr    tstop,t2sc
noansdet    mov     #$e6,admobyte
        jsr     moda2brx
        brset   7,bytelow,anstonedetect
        brclr   tof,t2sc,noansdet
        bclr    tof,t2sc
        dbnz    contt,noansdet
        bset    tstop,t2sc
        mov     #$09,tabla
        jsr     printline2
        mov     #$46,t1sc
        brclr   tof,t1sc,*      ;wait until the time is complete
        bclr    ansdet,flags1
        jmp     enddetectans
anstonedetect   jsr     i2cdelay        ;9. wait for answertone to disappear
        mov     #$e6,admobyte
        jsr     moda2brx
        brset   7,bytelow,anstonedetect
        bset    ansdet,flags1
enddetectans    rts

det1suns    mov     #$e2,admobyte
        mov     #$df,bytehigh
        mov     #$38,bytelow
        jsr     moda2btx
ans2225     mov     #$e6,admobyte
        jsr     moda2brx
        brclr   6,bytelow,ans2225       ;verifica si hay tono de respuesta del modem
(2100hz)
        mov     #$34,t2sc           ;coloca en preescalador del timer en 16
        mov     #$bd,t2modh
        mov     #$b8,t2modl         ;en este caso 155 mseg (194272 ciclos de
maquina)
        bclr    tstop,t2sc          ;coloca a correr el contador
        brclr   tof,t2sc,*          ;espera a que se complete el tiempo
        bset    tstop,t2sc          ;detiene el contador
        bclr    tof,t2sc            ;clarea el tof
```

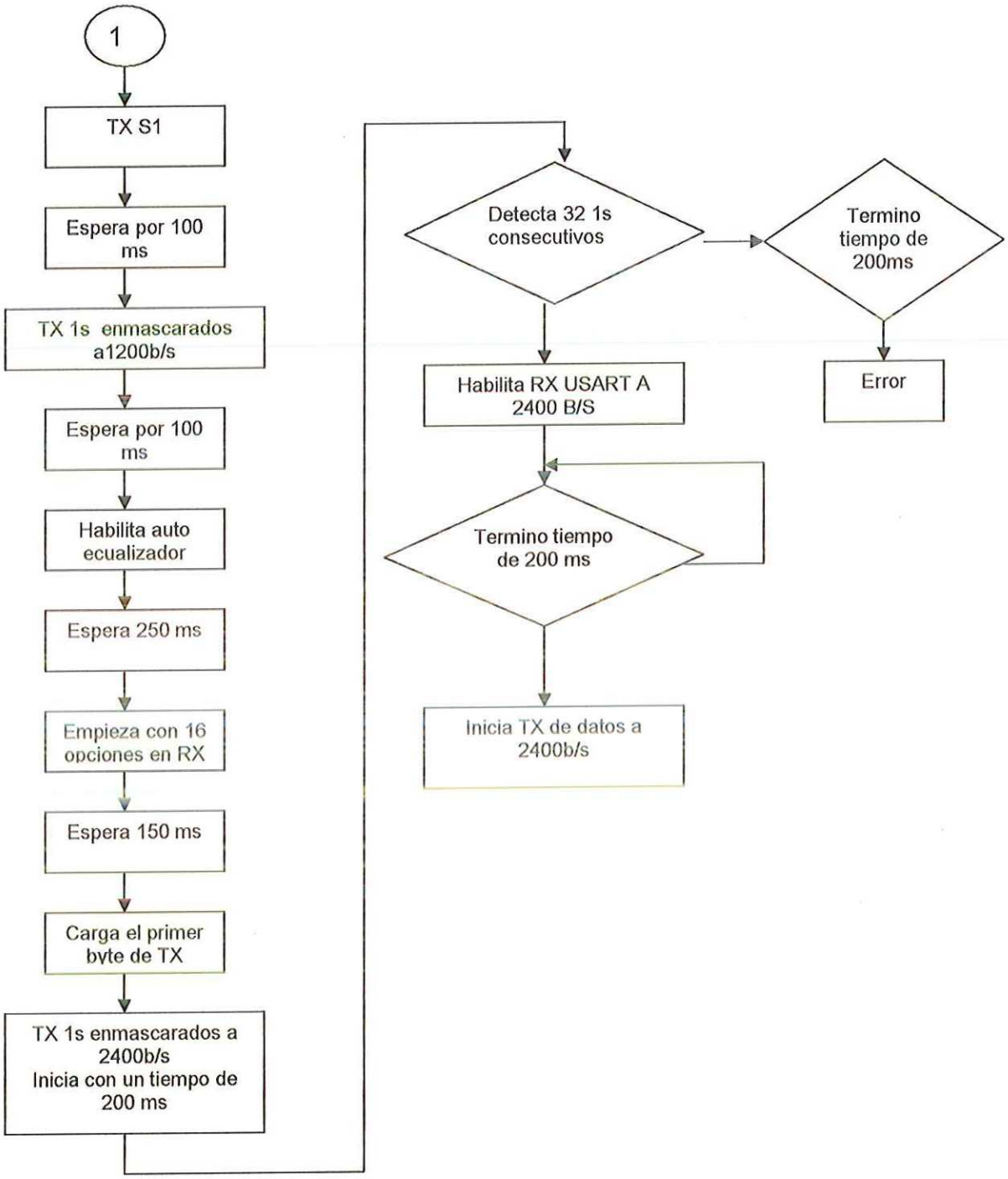
```
ans2225m    mov     #$e6,admobyte
            jsr     moda2brx
            brclr   6,bytelow,ans2225m      ;verifica si hay tono de respuesta del modem
(2100hz)
            mov     #$32,t2sc
            mov     #$8b,t2modh             ;configura el valor hasta el que se va a contar
            mov     #$89,t2modl             ;en este caso 456 mseg (571536 ciclos de
maquina)
            bclr    tstop,t2sc              ;coloca a correr el contador
ans2225e    mov     #$e6,admobyte
            jsr     moda2brx
            brclr   6,bytelow,ans2225e      ;verifica si hay tono de respuesta del modem
(2100hz)
            bset    onesundet,flags1
enddet1suns rts

transscr1st mov     #$e1,admobyte
            mov     #$cf,bytehigh
            mov     #$fb,bytelow
            jsr     moda2btx
endtransscr1st rts

detoneshb   bclr    scrambled1s,flags1
            mov     #$e2,admobyte
            mov     #$df,bytehigh
            mov     #$f8,bytelow
            jsr     moda2btx
            mov     #$06,contt
            mov     #$36,t2sc
            mov     #$ff,t2modh
            mov     #$ff,t2modl
            bclr    tstop,t2sc
letsdetones mov     #$e6,admobyte
            jsr     moda2brx
            brset   tof,t2sc,decremcontt
            brclr   0,bytehigh,letsdetones      ;wait for detect
            brclr   7,bytelow,letsdetones
            bset    scrambled1s,flags1
            bra     enddetoneshb
decremcontt bclr    tof,t2sc                ;clarea el tof
            dbnz    contt,letsdetones
enddetoneshb bset   tstop,t2sc              ;detiene el contador
            bclr    tof,t2sc
              rts

check1s270  mov     #$e6,admobyte
            jsr     moda2brx
            brclr   7,bytelow,check1s270
            brclr   0,bytehigh,check1s270       ;wait for detect
            mov     #$34,t2sc               ;timer 270 ms
            mov     #$52,t2modh
            mov     #$9e,t2modl
            bclr    tstop,t2sc
*           brclr   tof,t2sc,*              ;espera a que se complete el tiempo
det_s111    mov     #$e6,admobyte
            jsr     moda2brx
```

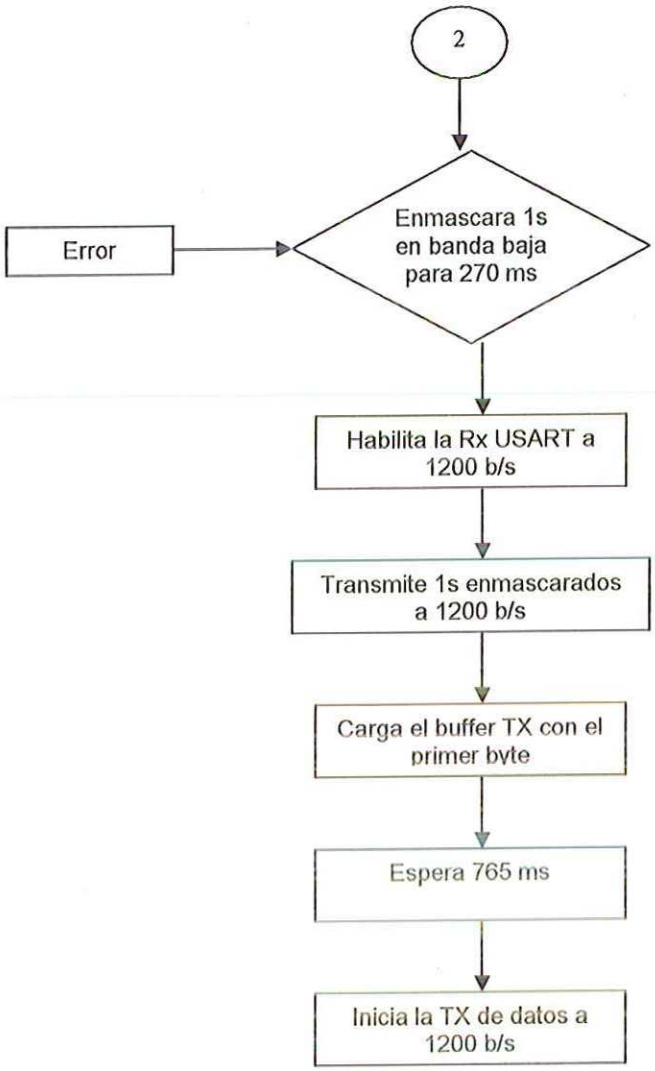
```
            brclr       7,bytelow,det_s111
            brclr       0,bytehigh,det_s111
            bset        tstop,t2sc
            bclr        tof,t2sc
endcheck1s270       rts

enrxusartt      mov         #$e2,admobyte
            mov         #$df,bytehigh
            mov         #$f6,bytelow
            jsr         moda2btx
endenrxusartt       rts

entxusartt      mov         #$e1,admobyte
            mov         #$cf,bytehigh
            mov         #$f7,bytelow
            jsr         moda2btx
endentxusartt   rts

txdatareadyi    mov         #$e6,admobyte
            jsr         moda2brx
            brclr       4,bytehigh,txdatareadyi
endtxdatareadyi rts

loadfirstt      mov         #$e3,admobyte
            mov         #$40,bytehigh
            jsr         modabtx
endloadfirstt   rts

rxdataready     bclr        confirm1,flags1
            mov         #$04,contt
            mov         #$36,t2sc
            mov         #$ff,t2modh
            mov         #$ff,t2modl
            bclr        tstop,t2sc
askagain        mov         #$e6,admobyte
            jsr         moda2brx
            brset       tof,t2sc,cumpliotiempo
            brclr       6,bytelow,askagain
            bset        confirm1,flags1
            bra         endrxdataready
cumpliotiempo   bclr        tof,t2sc                ;clarea el tof
            dbnz        contt,askagain
endrxdataready  bset        tstop,t2sc              ;detiene el contador
            bclr        tof,t2sc
            rts

rxdatareadyi    mov         #$e6,admobyte
            jsr         moda2brx
            brclr       6,bytelow,rxdatareadyi
endrxdatareadyi rts

recibedato      mov         #$e5,admobyte
            jsr         modabrx
endrecibedato   rts

sendtrama       jsr         txdatareadyi        ;enviaremos la palabra byte
```

```
                mov     #$e3,admobyte
                mov     #$42,bytehigh
                jsr     modabtx
                jsr     ret100ms
                jsr     txdatareadyi
                mov     #$e3,admobyte
                mov     #$59,bytehigh
                jsr     modabtx
                jsr     ret100ms
                jsr     txdatareadyi
                mov     #$e3,admobyte
                mov     #$54,bytehigh
                jsr     modabtx
                jsr     ret100ms
                jsr     txdatareadyi
                mov     #$e3,admobyte
                mov     #$45,bytehigh
                jsr     modabtx
                jsr     ret100ms
endsendtrama    rts
```

➢ **Diagrama de flujo del modem cmx868 para la rx**




1
TX S1
Espera por 100 ms
TX 1s enmascarados a1200b/s
Espera por 100 ms
Habilita auto ecualizador
Espera 250 ms
Empieza con 16 opciones en RX
Espera 150 ms
Carga el primer byte de TX
TX 1s enmascarados a 2400b/s
Inicia con un tiempo de 200 ms
Detecta 32 1s consecutivos
Termino tiempo de 200ms
Error
Habilita RX USART A 2400 B/S
Termino tiempo de 200 ms
Inicia TX de datos a 2400b/s



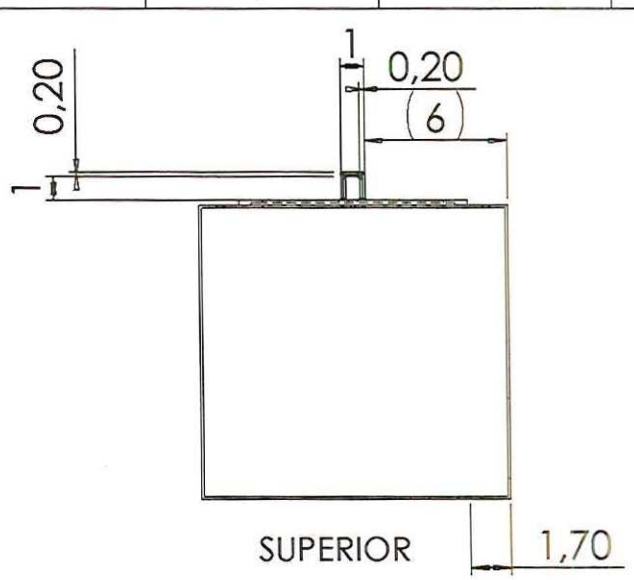
- **SUBRUTINAS PAR EL MANEJO DEL MODEM DE MODO RX (SE VAN A QUITAR ALGUNAS SUBRUTINAS YA USADAS EN EL TX)**

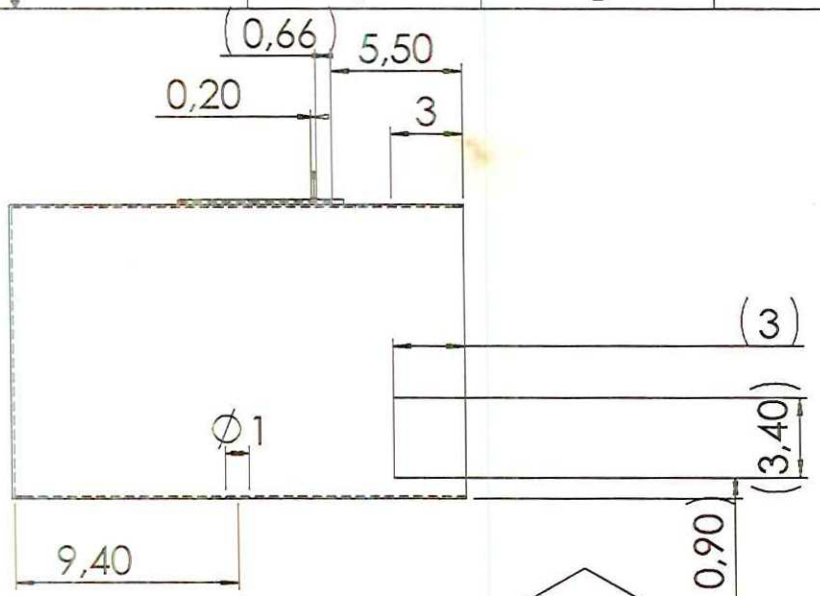
```
startmodem      mov     #$e0,admobyte
                mov     #$03,bytehigh
                mov     #$00,bytelow
                jsr     moda2btx
endstartmodem   rts

lowconsump      mov     #$e0,admobyte
                mov     #$02,bytehigh
                mov     #$00,bytelow
```

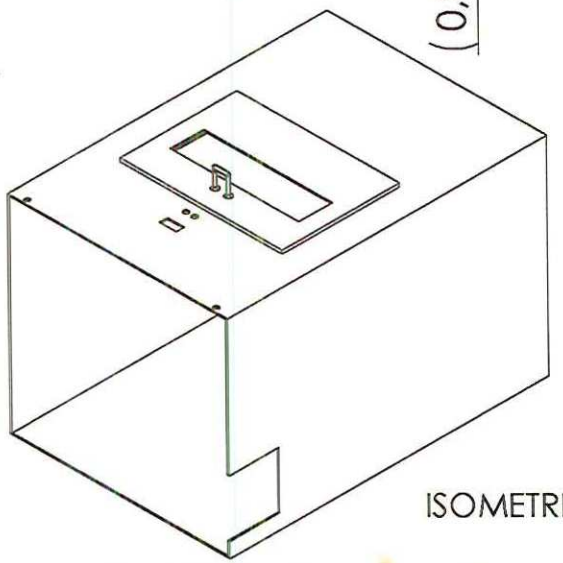
```
            jsr         moda2btx
endlowconsump   rts

ringflag    jsr         i2cdelay
            mov         #$e6,admobyte
            jsr         moda2brx
            brclr       6,bytehigh,ringflag
endringflag     rts

setupregrec mov     #$e1,admobyte
            mov     #$1e,bytehigh       ;select dtmf / tones mode
            mov     #$00,bytelow        ;user defined tx level -  no tones
            jsr     moda2btx
            mov     #$e2,admobyte
            mov     #$cf,bytehigh       ;select dtmf / tones mode
            mov     #$c0,bytelow        ;deteccion de progreso de llamada
            jsr     moda2btx
endsetupregrec  rts

takeoffhook mov     #$e0,admobyte
            mov     #$01,bytehigh       ;relay drive pin is pulled to vdd
            mov     #$00,bytelow
            jsr     moda2btx
endtakeoffhook  rts

transans    mov     #$e1,admobyte
            mov     #$1e,bytehigh
            mov     #$0a,bytelow
            jsr     moda2btx
endtransans     rts

turnoffans  mov     #$e1,admobyte
            mov     #$1e,bytehigh
            mov     #$00,bytelow
            jsr     moda2btx
endturnoffans   rts

transunscr1s mov    #$e1,admobyte
            mov     #$df,bytehigh
            mov     #$9b,bytelow
            jsr     moda2btx
endtransunscr1s rts

detoneslb   bclr    scrambled1s,flags1
            mov     #$e2,admobyte
            mov     #$cf,bytehigh
            mov     #$f8,bytelow
            jsr     moda2btx
            mov     #$06,contt
            mov     #$36,t2sc
            mov     #$ff,t2modh
            mov     #$ff,t2modl
            bclr    tstop,t2sc
letsdetoneslb mov   #$e6,admobyte
            jsr     moda2brx
            brset   tof,t2sc,decremconttlb
```

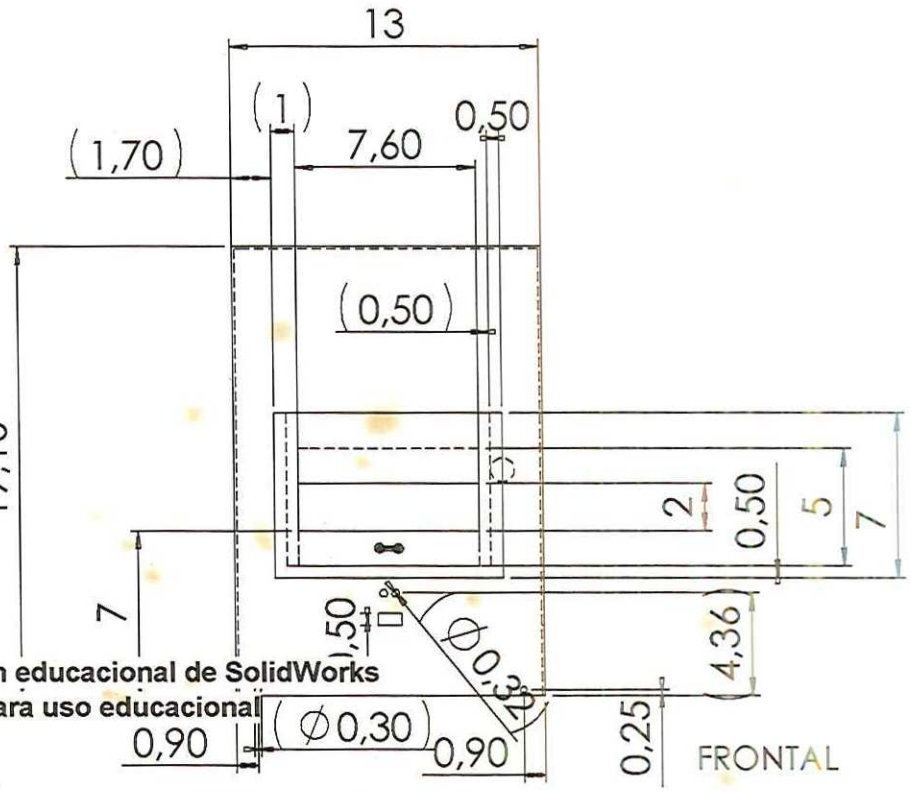
```
            brclr   0,bytehigh,letsdetoneslb     ;wait for detect
            brclr   7,bytelow,letsdetoneslb
            bset    scrambled1s,flags1
            bra     enddetoneslb
decremconttlb  bclr   tof,t2sc                  ;clarea el tof
            dbnz    contt,letsdetoneslb
enddetoneslb   bset   tstop,t2sc                ;detiene el contador
            bclr    tof,t2sc
            rts

check1s270    mov     #$e6,admobyte
            jsr     moda2brx
            brclr   7,bytelow,check1s270
            brclr   0,bytehigh,check1s270       ;wait for detect
            mov     #$34,t2sc                   ;timer 270 ms
            mov     #$52,t2modh
            mov     #$9e,t2modl
            bclr    tstop,t2sc
*            brclr   tof,t2sc,*                 ;espera a que se complete el tiempo
det_s111      mov     #$e6,admobyte
            jsr     moda2brx
            brclr   7,bytelow,det_s111
            brclr   0,bytehigh,det_s111
            bset    tstop,t2sc
            bclr    tof,t2sc
endcheck1s270   rts

enrxusartr    mov     #$e2,admobyte
            mov     #$cf,bytehigh
            mov     #$f6,bytelow
            jsr     moda2btx
endenrxusartr   rts

entxusartr    mov     #$e1,admobyte
            mov     #$df,bytehigh
            mov     #$f7,bytelow
            jsr     moda2btx
endentxusartr   rts

transscr1sr   mov     #$e1,admobyte
            mov     #$df,bytehigh
            mov     #$fb,bytelow
            jsr     moda2btx
endtransscr1sr  rts

loadfirstr    mov     #$e3,admobyte
            mov     #$26,bytehigh
            jsr     modabtx
endloadfirstr   rts

rxdataready   bclr    confirm1,flags1
            mov     #$04,contt
            mov     #$36,t2sc
            mov     #$ff,t2modh
            mov     #$ff,t2modl
            bclr    tstop,t2sc
```

```
askagain      mov     #$e6,admobyte
              jsr     moda2brx
              brset   tof,t2sc,cumpliotiempo
              brclr   6,bytelow,askagain
              bset    confirm1,flags1
              bra     endrxdataready
cumpliotiempo bclr    tof,t2sc              ;clarea el tof
              dbnz    contt,askagain
endrxdataready bset   tstop,t2sc            ;detiene el contador
              bclr    tof,t2sc
              rts

recibedato    mov     #$e5,admobyte
              jsr     modabrx
endrecibedato rts

rxdatareadyi  mov     #$e6,admobyte
              jsr     moda2brx
              brclr   6,bytelow,rxdataready
endrxdatareadyi rts

txdatareadyi  mov     #$e6,admobyte
              jsr     moda2brx
              brclr   4,bytehigh,txdatareadyi
endtxdatareadyi rts

receivetrama  jsr     rxdatareadyi
              jsr     recibedato
              lda     bytehigh
              sta     caracter
              jsr     printchar2
              jsr     rxdatareadyi
              jsr     recibedato
              lda     bytehigh
              sta     caracter
              jsr     printchar2
              jsr     rxdatareadyi
              jsr     recibedato
              lda     bytehigh
              sta     caracter
              jsr     printchar2
              jsr     rxdatareadyi
              jsr     recibedato
              lda     bytehigh
              sta     caracter
              jsr     printchar2
endreceivetrama rts
```

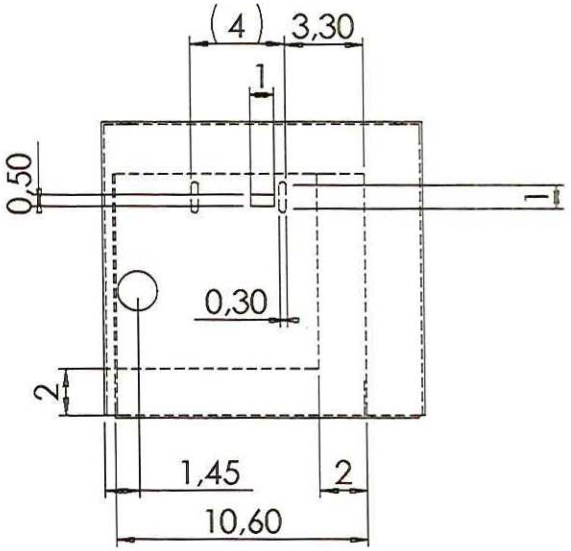




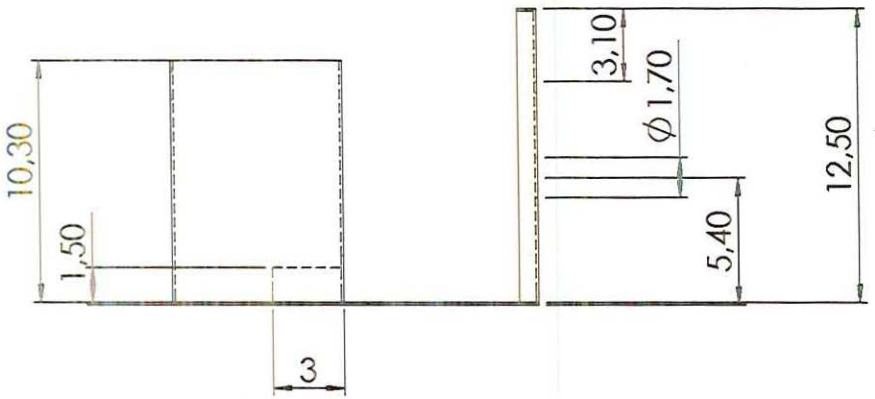




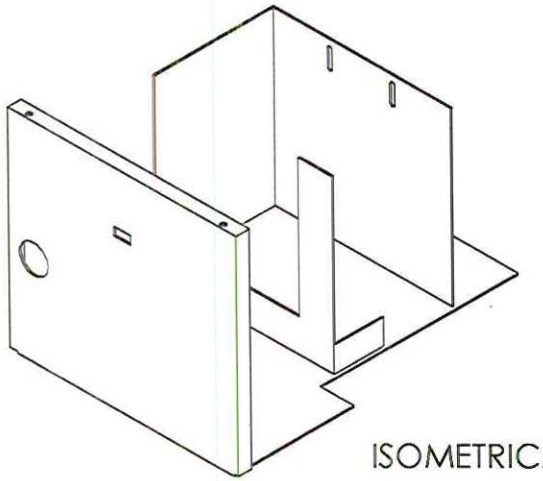
MEDIDAS: CMS	INCATEL
NUMERO: PLANO 2	CAJA RECEPTOR
TERMINADO 26/05/2003	MATERIAL COLROL 18
ESCALA: 1:3	



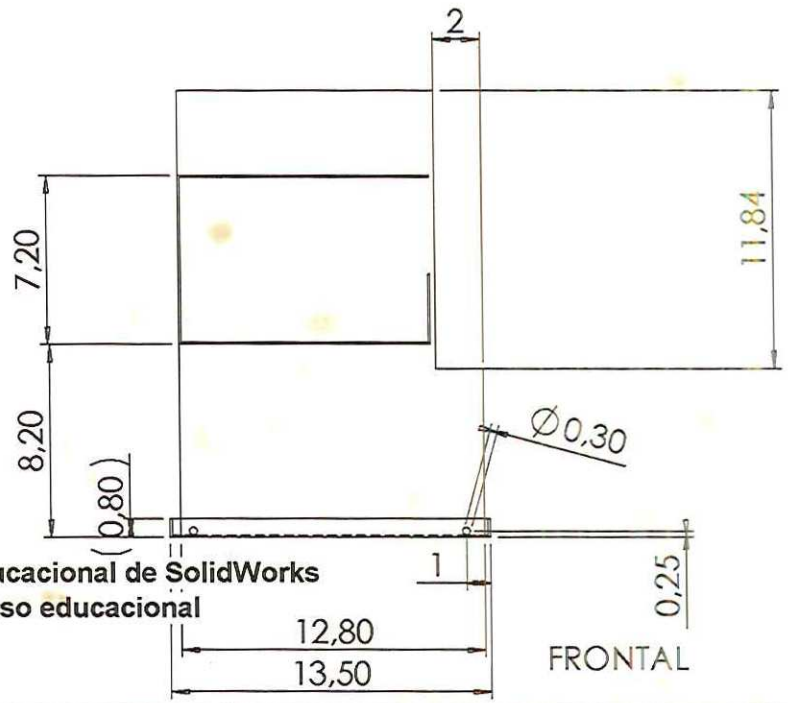
SUPERIOR



LATERAL

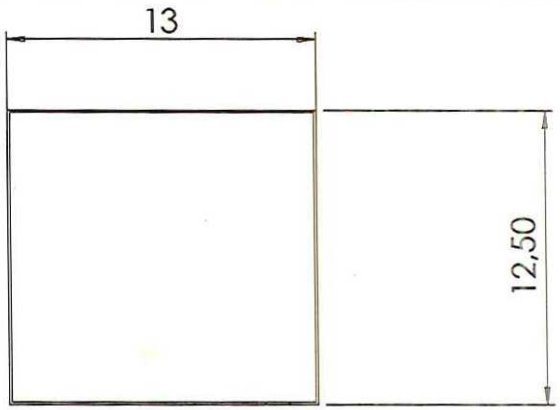


ISOMETRICA

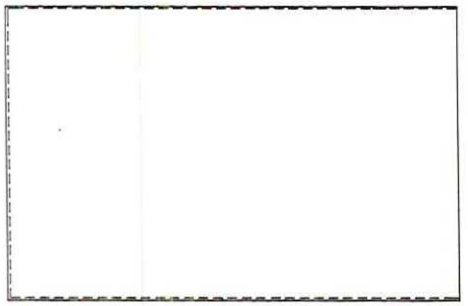


FRONTAL

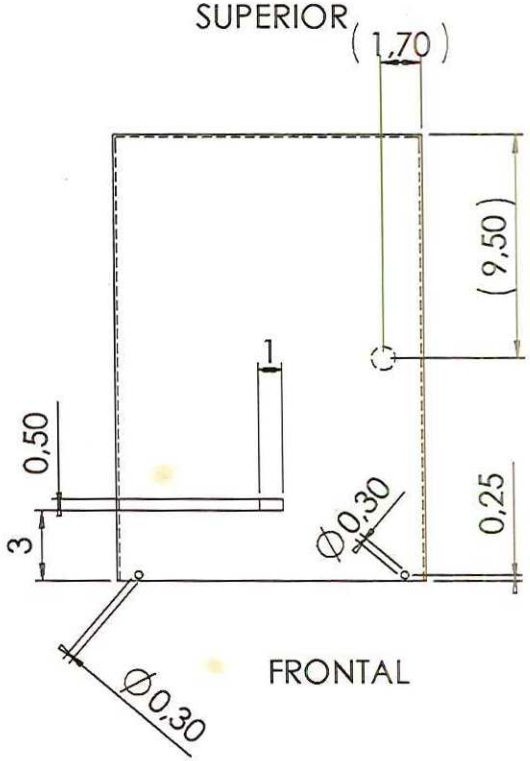
MEDIDAS: CMS	INCATEL
NUMERO: PLANO 4	TAPA RECEPTOR
TERMINADO 26/05/2003	MATERIAL COLROL 18
ESCALA: 1:3	



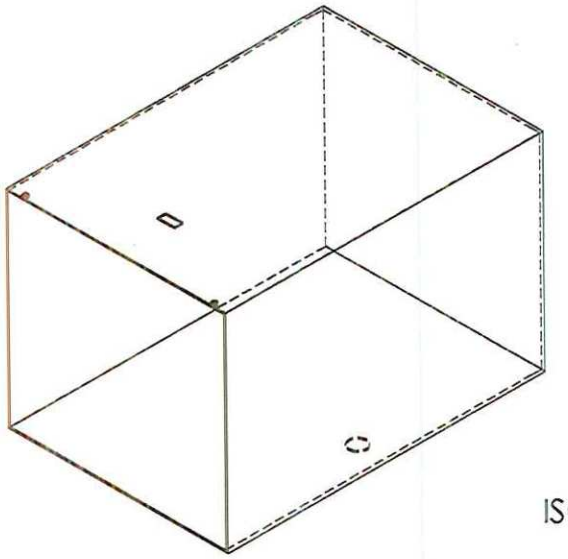
SUPERIOR



LATERAL

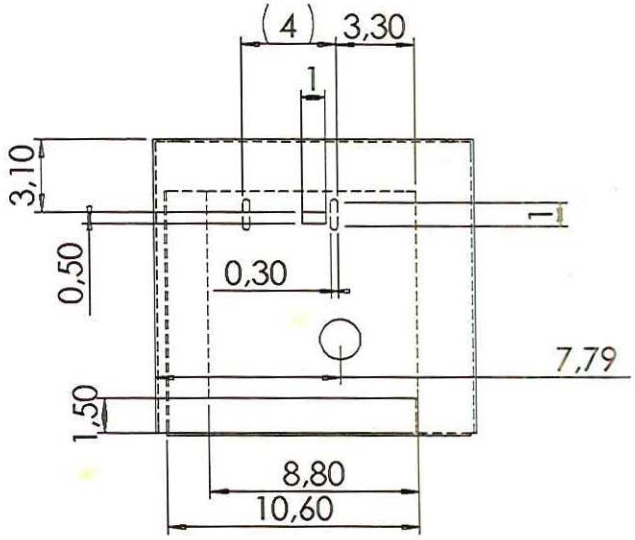


FRONTAL

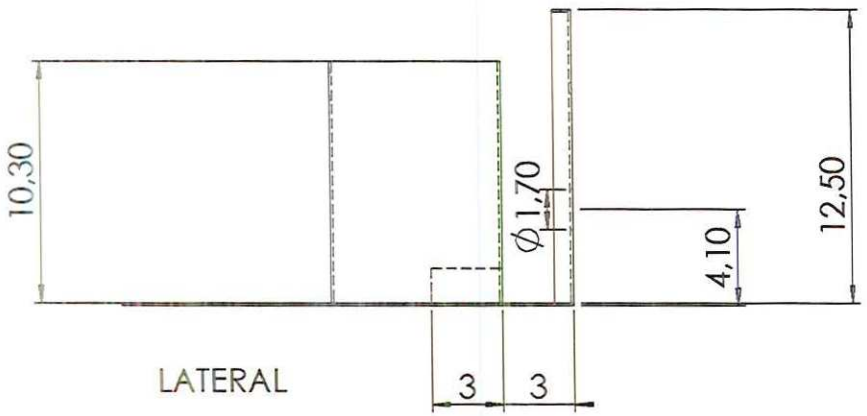


ISOMETRICA

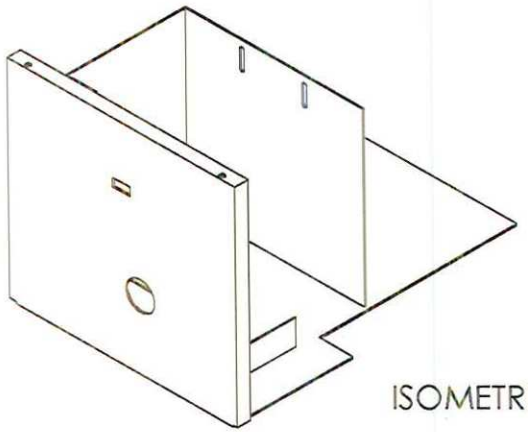
MEDIDAS: CMS	INCATEL
NUMERO: PLANO 1	CAJA TRANSMISOR
TERMINADO 23/05/2003	MATERIAL COLROL 18
ESCALA: 1:3	



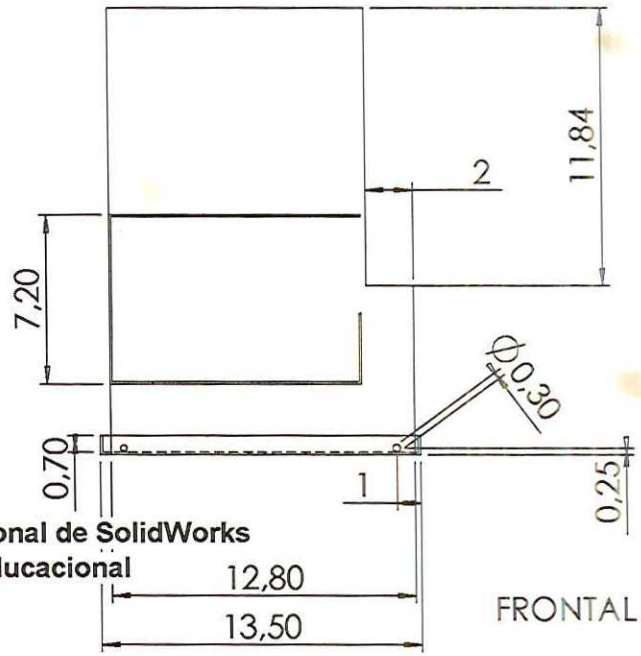
SUPERIOR



LATERAL



ISOMETRICA



FRONTAL

MEDIDAS: CMS	INCATEL
NUMERO: PLANO 3	TAPA TRANSMISOR
TERMINADO 26/05/2003	MATERIAL COLROL 18
ESCALA: 1:3	