



**CENTRO DE CIENCIAS APLICADAS Y
DESARROLLO TECNOLÓGICO**

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



REPORTE TECNICO

Sistema SMEEP Prototipo I

PATROCINADO POR EL CCADET

**Antonio Garcés Madrigal
José Luis Pérez Silva**

Febrero de 2009

INDICE

	Introducción	4
1.	El sistema	5
1.1	Módulo de control	5
1.2	Módulo local de presentación e I/O	6
1.3	Módulo de adquisición y almacenamiento	7
1.4	Módulo de comunicaciones e interfaz PC	8
1.5	Modos de operación	9
1.6	Módulo de alimentación eléctrica	10
2.	Transducción de variables físicas	10
2.1	Transductores inteligentes	11
3.	Ingeniería del producto	12
3.1.	Listado de componentes	15
3.2	Circuitos Impresos	15
4.	Pruebas de funcionamiento.	17
5.	Conclusiones	17
6.	Apéndices	19
6.1	Listado de programas	19
6.2	Hojas de componentes	48

RESUMEN

El objetivo de este reporte técnico es dar toda la información del diseño y construcción de un equipo de medición para la realización de experimentos en un ambiente escolar interactivo, orientado a proporcionar los elementos necesarios para la realización de prácticas escolares en las que los profesores y los alumnos puedan medir variables físicas (voltaje, corriente, temperatura, tiempo, PH, frecuencia, intensidades acústica y luminosa, velocidad de giro y desplazamiento, entre otras) de tal forma que puedan realizar experiencias que les permitan validar experimentalmente los conocimientos adquiridos en el aula. Este sistema es continuación y mejora del desarrollo original del CCADET con el nombre de LABSIS, el cual en su momento fue transferido a la compañía Harry Mazal S.A. Entre las mejoras que se incorporaron en el diseño están la portabilidad, la incorporación de atributos de identificación automática de sus sensores, auto escala y auto despliegue de las variables relacionadas, la posibilidad de calibrar, configurar y programar la operación de SMEEP desde un teclado auxiliar removible, la capacidad de almacenar en memoria no volátil hasta 512 MB de datos o transmitirlos a una PC. Este reporte incluye todos los elementos del diseño, como son los circuitos esquemáticos, los diseños de las tabletas impresas, las especificaciones de los elementos de circuito, los programas del sistema de medición, las hojas técnicas de los componentes empleados, así como los elementos de la ingeniería del producto.

INTRODUCCIÓN

SMEEP es un sistema orientado a proporcionar los elementos necesarios para la realización de prácticas escolares en las que los profesores y los alumnos puedan medir variables físicas (voltaje, corriente, temperatura, tiempo, PH, frecuencia, intensidades acústica y luminosa, velocidad de giro y desplazamiento, entre otras) de tal forma que puedan realizar experiencias que les permitan validar experimentalmente los conocimientos adquiridos en el aula. Algunas de las ventajas principales son la portabilidad, lo que permite realizar prácticas en el laboratorio, en campo o en el domicilio de los estudiantes; la incorporación de atributos de identificación automática de sus transductores, auto escala y auto despliegue de las variables relacionadas, la posibilidad de calibrar, configurar y programar la operación de SMEEP desde un teclado auxiliar removible, la capacidad de almacenar en memoria no volátil hasta 512 MB (configuración básica 32 MB) de datos de las mediciones realizadas o transmitirlos (de forma simultánea o posterior a la experiencia) a una PC. Adicionalmente el sistema cuenta con un programa que ayuda a realizar la presentación de datos en la computadora de tal forma que se pueden seleccionar diversos métodos de visualización, ya sea tipo osciloscopio, o de presentación de datos como son graficación de diversas variables registradas o a través de métodos de regresión, abriendo la posibilidad que el usuario pueda incluso utilizar sus propios métodos de análisis. El sistema esta integrado en un pequeño maletín y puede operar por medio de alimentación de energía eléctrica AC o por medio de baterías, dependiendo del tipo de operación seleccionada por el usuario. A continuación se muestra el diagrama general del sistema:

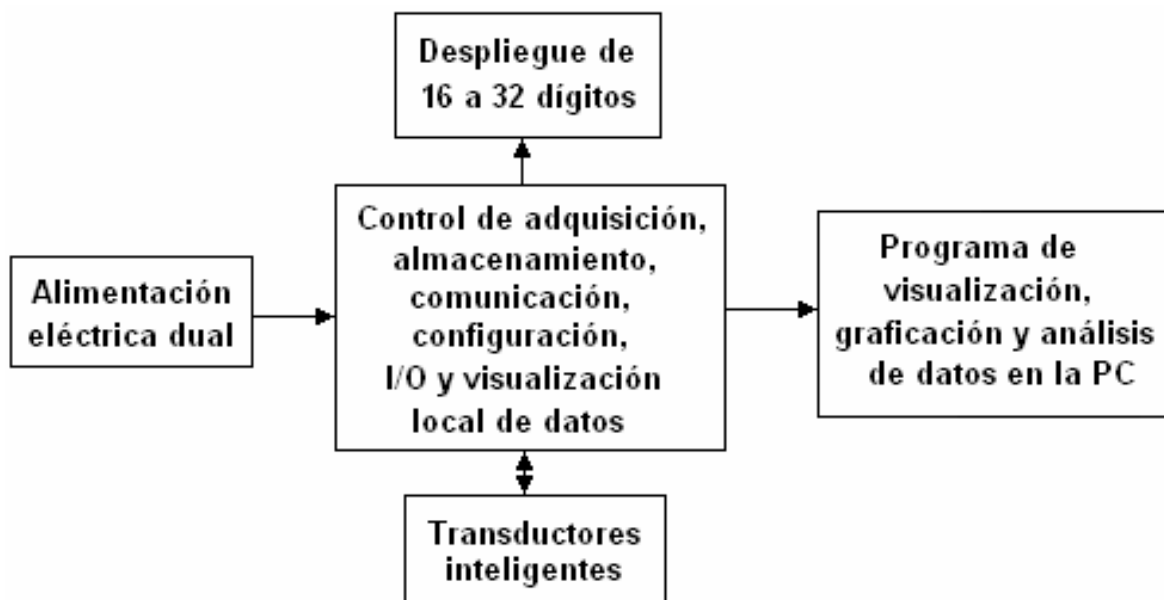


Figura 1. Diagrama de bloques del smeep

1. EL SISTEMA

El sistema esta constituido por dos bloques: un primer bloque está constituido por el sistema central controlado por un microprocesador, que permite la interacción con el usuario a través de una interfaz que cuenta con un teclado y una pantalla de despliegue, además incluye un puerto de comunicación a PC y una ranura de conexión. Un segundo bloque lo constituyen las tarjetas electrónicas de medición de variables físicas, las cuales dan versatilidad al equipo al trabajar una interfaz abierta para realizar mediciones con el sistema.

El equipo está diseñado para operar en forma autónoma, o con una PC. En el modo de operación autónomo el equipo puede realizar cualquiera de las mediciones asociadas a sus módulos de medición y los datos pueden ser desplegados en la pantalla y almacenados en memoria para su posterior transmisión a un archivo en PC. Adicionalmente en su modo de operación interactiva con PC, el sistema puede transmitir la información a la computadora, permitiendo interactuar al usuario con una aplicación diseñada para el manejo de datos. El equipo puede ser configurado a través de su interfaz de usuario, en la velocidad de adquisición, transmisión de datos y espacio de memoria.

En el modo de operación con PC cuenta con una aplicación que permite incorporar la metodología de trabajo particular de cada profesor o alumno a su modo de operación. Adicionalmente el sistema se configura de forma automática cada vez que el usuario cambia un módulo o tarjeta de medición, de forma tal que las unidades de medición y despliegue se ajustan de acuerdo a la medición a realizar.

El equipo esta diseñado para ser instalado en un portafolio de tipo rígido en el cual se concentran tanto el sistema central, sus puertos de conexión y expansión, su interfaz de usuario y varios de sus módulos de medición, lo cual lo convierte en un equipo portátil.

Para lo anterior se ha diseñado una plataforma propietaria que integra a todos los componentes que conforman al sistema y que son los que se indican a continuación: módulo de interfaz de usuario, local y remoto, módulo de comunicaciones, módulo de configuración del sistema y almacenamiento de datos, módulo de adquisición de datos y control, módulo de manejo de archivos, módulo de conversión de datos y adecuación de la información, módulo de despliegue de datos, módulo de alimentación fija y portátil, programa de interfaz remota (cuando opera con PC) despliegue de información y herramienta que permite realimentar la interpretación de los resultados al proceso de aprendizaje y un grupo de sensores auto adaptables que permiten al alumno medir variables físicas como: voltaje corriente, resistencia, capacitancia, inductancia, temperatura, pH, intensidad luminosa y acústica, entre otros.

A continuación describimos cada uno de los módulos que componen el sistema smeeep.

1.1 Módulo de control

La etapa de control está dividida en tres submódulos:

1. El módulo de entrada/salida local.
2. El módulo de interfaz y adquisición.
3. El módulo de comunicación con la PC.

La separación en submódulos se justifica en el sentido de dar mayor versatilidad y autonomía al equipo, así como extender las capacidades de expansión o velocidad de operación, proporcionando a su vez una relativa redundancia que permitiría operar al equipo aún en condiciones de falla de alguno de sus componentes y así como proporcionar al usuario un mayor

nivel de interactividad con el equipo. A continuación se presentan los esquemas generales de los módulos componentes de esta etapa.

1.2 Módulo local de presentación de I/O

El módulo local de presentación de entrada/salida es el responsable de presentar los datos en el sistema, el cual recibe los datos y comandos desde el teclado auxiliar y da formato a los datos provenientes de los otros módulos y que se visualizarán en el despliegue local, y está constituido de las siguientes partes:

1. Un sistema de teclado auxiliar que se puede poner o quitar para poder vía éste introducir información al sistema
2. El sistema que da formato a los datos de las etapas de adquisición de entrada/salida y teclado, así como la transmisión de estos datos a la etapa de visualización local, es decir en el despliegue de cristal líquido instalado en el equipo portátil.
3. Un sistema de visualización de datos vía una pantalla de cristal líquido.
4. El bus de datos a otros módulos de control.

El diagrama de bloques de esta etapa se muestra en la figura 2, en la figura 3, se muestra el diagrama esquemático de la etapa de presentación local de datos, en la figura 4 se muestra el diagrama esquemático de entrada/salida del teclado.

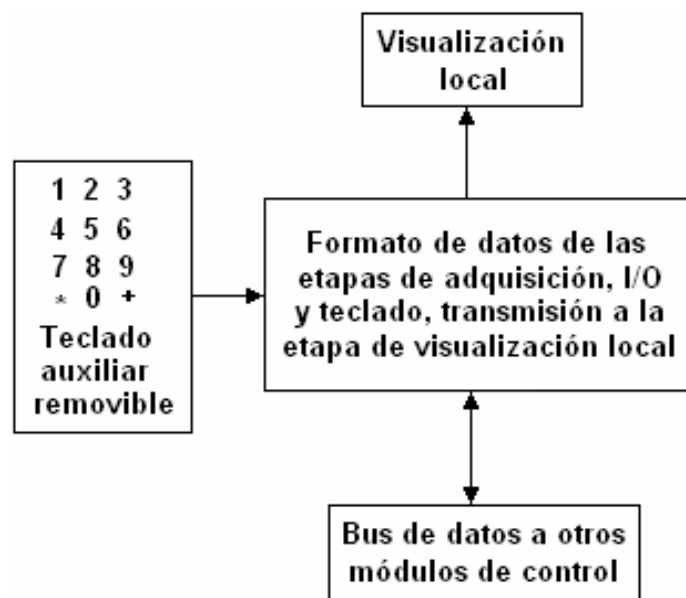


Figura 2. Diagrama de bloques de módulo I/O local

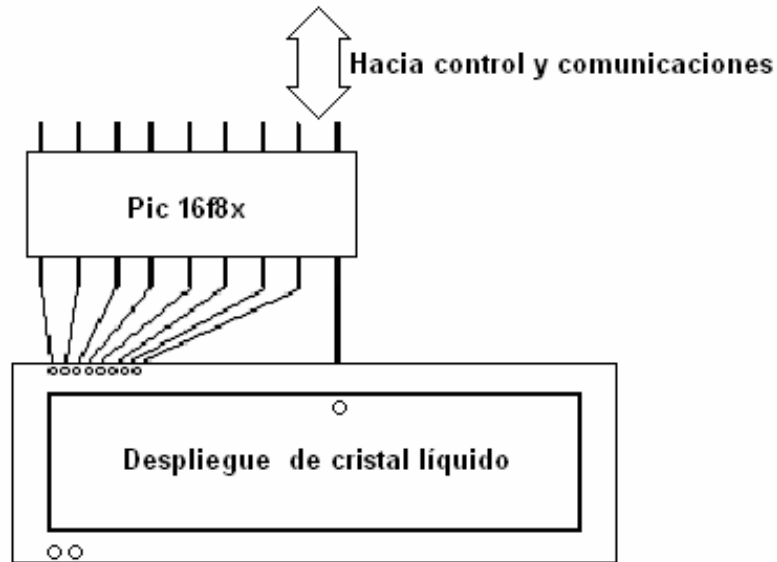


Figura 3. Diagrama esquemático de etapa presentación local de datos

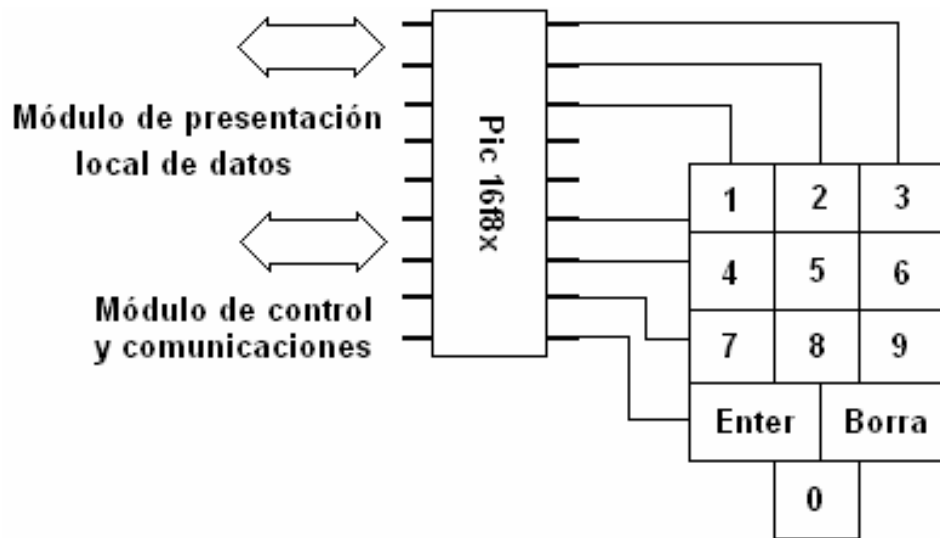


Figura 4. Diagrama esquemático de I/O teclado

El elemento central de esta etapa es el Pic16f8x, cuyas características y especificaciones se muestran en sus hojas de datos en el anexo de este reporte.

1.3 Módulo de adquisición y almacenamiento

El módulo de adquisición y almacenamiento se encarga de comunicarse con los transductores a través de protocolos propios de comunicación así como del almacenamiento de los datos en la memoria no volátil. En la figura 5 se muestra el diagrama de bloques del módulo de adquisición y almacenamiento.

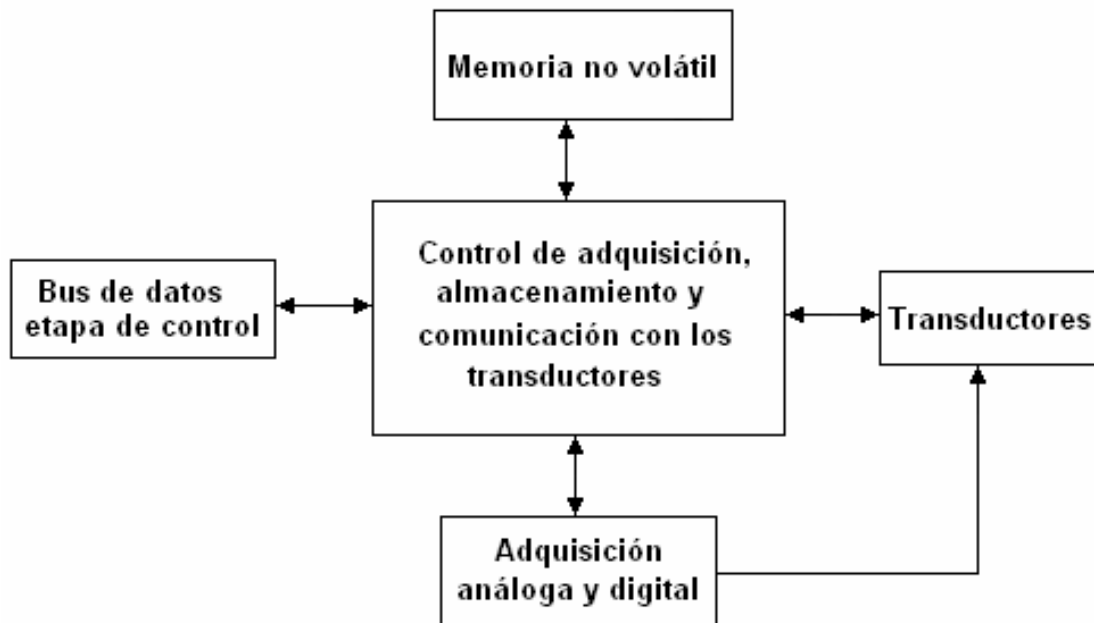


Figura 5. Diagrama de bloques del módulo de adquisición y almacenamiento

En la figura 6 se muestra el diagrama esquemático del módulo de adquisición y almacenamiento

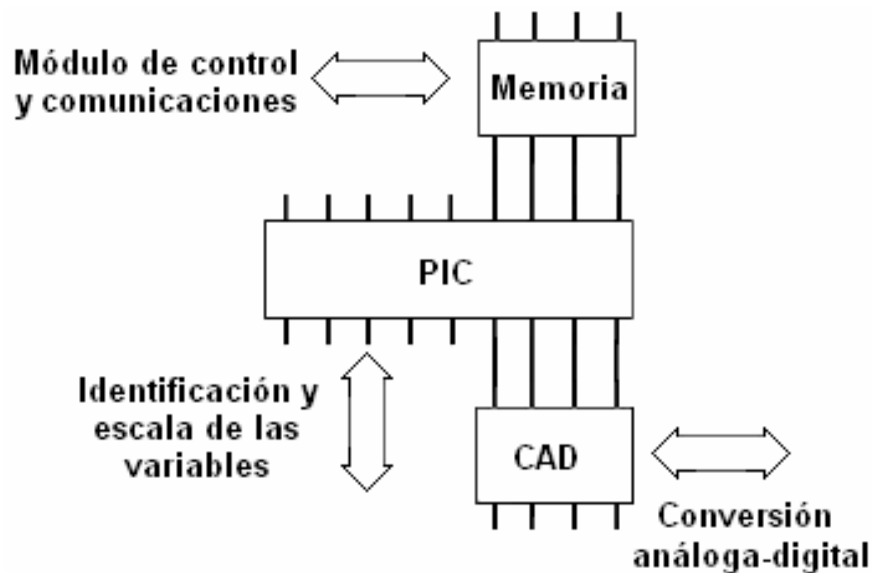


Figura 6. Diagrama esquemático del módulo de adquisición y almacenamiento

1.4. Módulo de comunicaciones e interfaz PC

El módulo de comunicaciones e interfaz con la PC transmite toda la información a los programas residentes en ésta. En la figura 7 se muestra el diagrama de bloques del módulo de comunicaciones e interfaz PC, y en la figura 8. se muestra el diagrama esquemático del módulo de control y comunicaciones



Figura 7. Diagrama de bloques del módulo de comunicaciones e interfaz PC

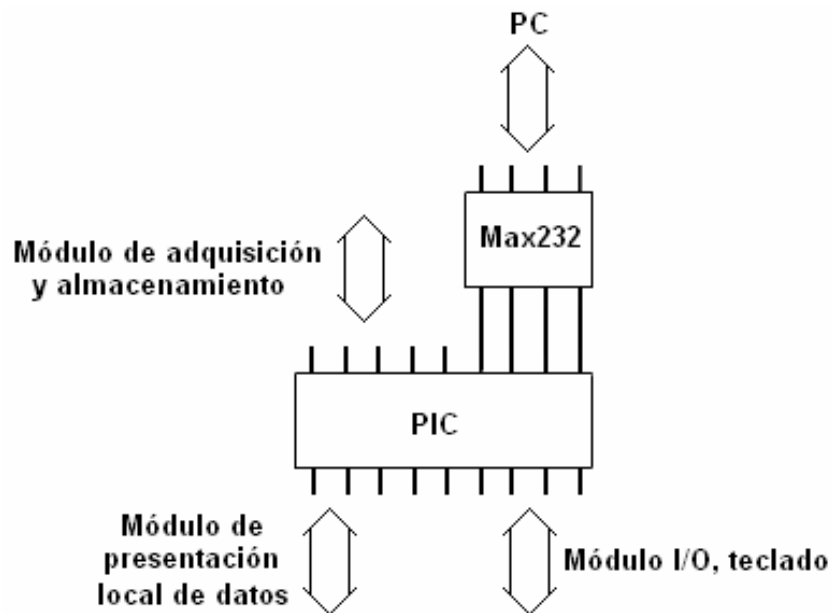


Figura 8. Diagrama esquemático del módulo de control y comunicaciones

1.5. Modos de operación.

La forma en que los diferentes microcontroladores se comunican entre si es coordinada por el módulo de comunicación y control, de forma tal que escudriña los diferentes módulos activos, pudiendo prescindir de algunos de ellos dependiendo de las diferentes modalidades de funcionamiento disponibles:

1. Modo portátil (por medio de baterías),
2. Modo estacionario (con suministro eléctrico de una toma de AC), y
3. Modo de mantenimiento y configuración:

1.6 Módulo de alimentación eléctrica

El smEEP cuenta con dos fuentes de alimentación, una requiere alimentación estándar de la línea de alimentación eléctrica de 120 volts y uno basado en baterías recargables en esquemas de 6 volts dependiendo de la portabilidad y autonomía requeridas, además el módulo de baterías es recargado por el suministro eléctrico de A/C con el fin de hacer más económica la operación del equipo. En la figura 9 se muestra el diagrama de bloques de la fuente de alimentación, y en la figura 10 el diagrama esquemático de la fuente de alimentación.



Figura 9. Diagrama de bloques de la fuente de alimentación

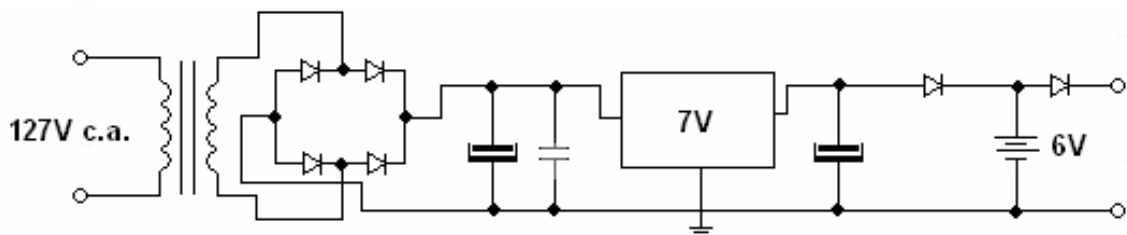


Figura 10. Diagrama esquemático de la fuente de alimentación

2. Transducción de variables físicas

Los transductores están diseñados para que el alumno pueda trabajar con la parte real de las experiencias prácticas propuestas por el profesor o por él mismo, con un mínimo de riesgo a fallas y accidentes pues los transductores establecen la escala de operación y se identifican en el sistema de manera automática, además cuentan con etapas de aislamiento que protegen al usuario y al sistema smEEP, de acuerdo a los niveles de protección convencionales para los equipos que se conectan a la red de alimentación eléctrica de 120 V. En la figura 11 se muestra el diagrama esquemático del sistema completo.

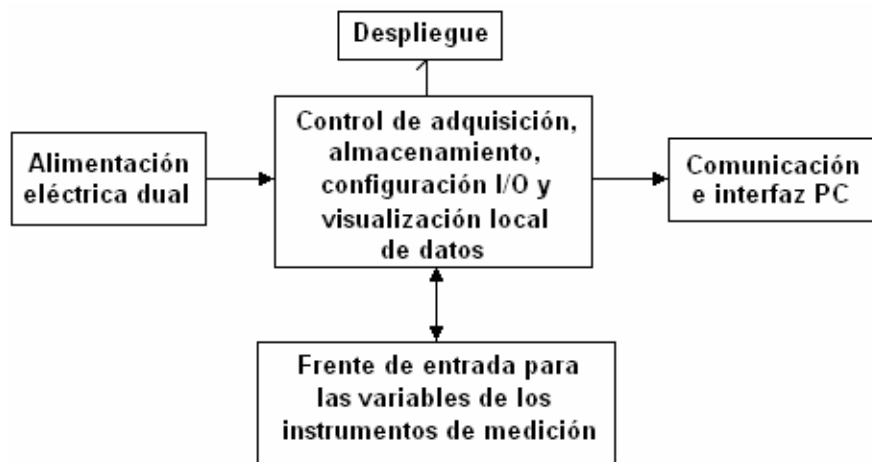


Figura 11. Diagrama del sistema smEEP

En la fotografía de la figura 12 se muestra el sistema completo armado en protoboard y separado por módulos.

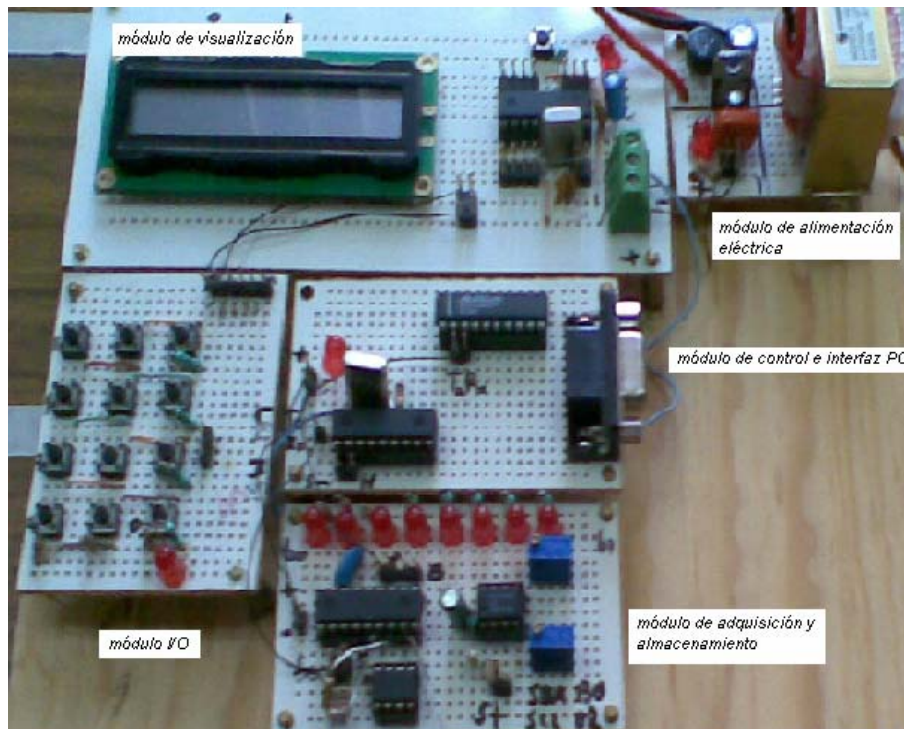


Figura 12. Foto de prototipo operativo del sistema smEEP

2.1. Transductores inteligentes

Los transductores inteligentes son los instrumentos de medición de las variables físicas a medir y realizan sus procesos en un esquema de operación automática, generando por tal proceso una señal eléctrica analógica y la palabra digital que lleva información del rango de escala y variable a medir, de forma tal que el equipo presenta al usuario, almacena en su memoria y transmite a la PC la información en formato y dimensiones correspondientes, esto es voltaje, corriente, resistencia, etcétera, el factor de escala y valor numérico de la variable medida. En la figura 13 se muestra el diagrama de bloques de los transductores 'inteligentes'

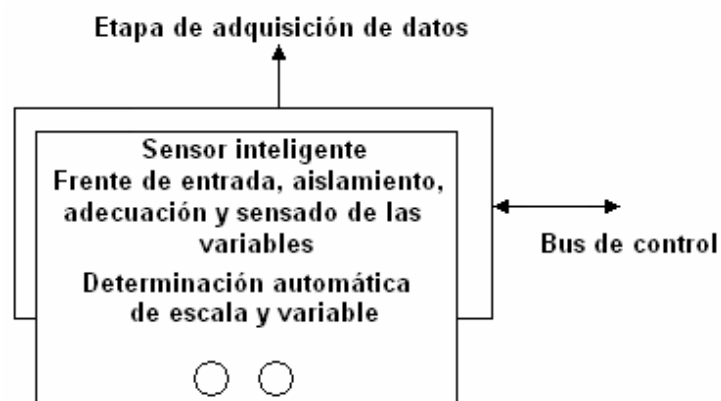


Figura 13. Diagrama general de los sensores 'inteligentes'

3. Ingeniería del producto

Se cuenta con una colección actual de 13 instrumentos y se están diseñando más. Deseamos que el sistema permita convertir la interfaz en cualquier instrumento que van desde voltímetro de CA y CD hasta fotocolorímetro. La idea central del producto en su ingeniería fue desarrollar todos los instrumentos y optimizar su tamaño de forma tal que pudiésemos encontrar el máximo volumen necesario para colocarlos. De la misma forma se diseñó la interfaz y se diseñó el circuito impreso para lograr la misma optimización. En la interfaz deberían de poder caber la electrónica completa, el procesador, la memoria, el despliegue y el puerto de comunicación.

Logrado el concepto pasamos a pensar en la ingeniería de la envolvente. Las características deseadas eran que fuese mayor que un multímetro portátil comercial, que pudiese tomar las diferentes configuraciones de posición y que pudiese acoplarse cualquiera de los instrumentos en forma sencilla a la interfaz.

El primer problema que se nos presentó fue definir cual sería la mejor forma de conexión, ya que deberíamos poder tener una forma de codificar el instrumento conectado para que la interfaz se pudiera acoplar a él y tomara el programa de calibración y presentación de datos que le correspondiera. Se pensó en una extensión tipo bus con cuatro líneas de conexión que corresponde a la tierra y el voltaje de alimentación, la salida de voltaje a la interfaz y el código de identificación del instrumento. El resultado es el que se presenta en la siguiente figura:



Figura 14. Modulo de instrumentación

La interfaz debería de contar con todos los elementos propuestos por lo que se llegó al concepto mostrado en la figura que se muestra a continuación.



Figura 15. Concepto del Sistema.

El concepto se pensó fundamentalmente en función de las posibilidades de producción con las que contamos en el CCADET, por lo que las envolventes se pueden termoformar de láminas plásticas desarrollando el molde de aluminio y después aplicado pintura o en dos partes una la caja rectangular de plástico blanco y las orillas de plástico suave de color.

La forma como se pensó la pieza para colocar el sistema en diferentes posiciones es desplazando la estructura superior del perfil de la interfaz y girando sobre pernos. Este sistema nos permite también construir termo-formada la tapa del conector de comunicaciones y la cerradura de las baterías se pueden construir del mismo plástico por medio de un troquel térmico.

El concepto terminado se muestra en la figura que se muestra a continuación.



Figura 16. Concepto de ingeniería terminado.

En el sistema de lectura directa por la computadora el menú de transferencia tiene la encomienda de enviar por el puerto de comunicaciones la información concerniente del instrumento, esta información transmitida es evaluada por el programa y como respuesta se obtiene el valor leído de cada instrumento. Este valor se muestra en la pantalla del instrumento. En la figura inferior se muestra la pantalla de esta interfaz en la PC.

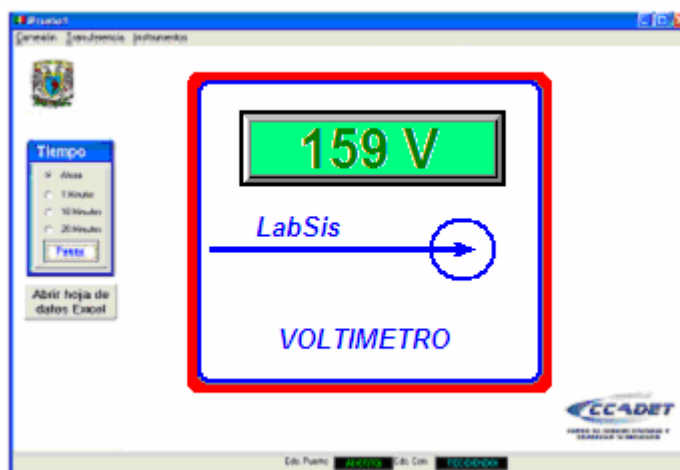


Figura 17. Interfaz de usuario

Para la elaboración del prototipo I con el que se han venido desarrollando las pruebas de funcionamiento se han requerido de los componentes eléctricos y electrónicos que se indican a continuación:

3.1. Listado de componentes

Para la realización del prototipo se utilizaron los siguientes componentes eléctricos y electrónicos.

Cantidad	Componente	Tipo	Hoja Técnica
1	ADC0831	8 bit serial I/O A/D Converter	Apéndice B
1	AT24022	Two-wire Serial EEPROM	Apéndice B
1	LM020L	Display LCD	Apéndice B
1	MX232	Fully decoder	Apéndice B
1	MC7808	Positive Voltage Regulator	Apéndice B
4	Pic16f84	8 bits microconteller	Apéndice B
1	10k	Potenciómetro de pot1, pot2	
1	470 @ 1/4	Resistor de carbón	
3	100 @ 1/4	Resistor de carbón	
4	10k @ 1/4	Resistor de carbón	
4	1000µf 15 V	Capacitores polarizados	
5	0.1 µf 250 V	Capacitares no polarizados	
1	127-9 a 300mA @60 Hz	Transformador	
1	W005M	Puente rectificador	Apéndice B
3	IN4001	Diodos	
1	3mm rojo	Led	Apéndice B
1	cal 14 de 1.5 mts	Clavija polarizada con cable	
12	2 terminales	Micro switch	Apéndice B

En el apéndice B de este reporte se encuentran las hojas técnicas de estos componentes.

Así mismo se han desarrollado los PCBs de las placas de circuito impreso que se muestran en las siguientes figuras:

3.2. Circuitos impresos

En las siguientes figuras se muestran las placas de circuito impreso de los módulos que forman el sistema smEEP.

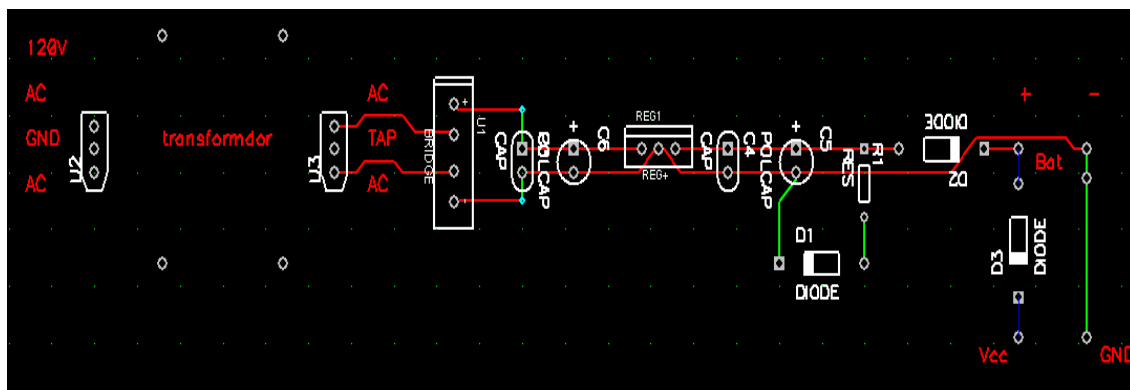


Figura 18. Módulo de la fuente de alimentación

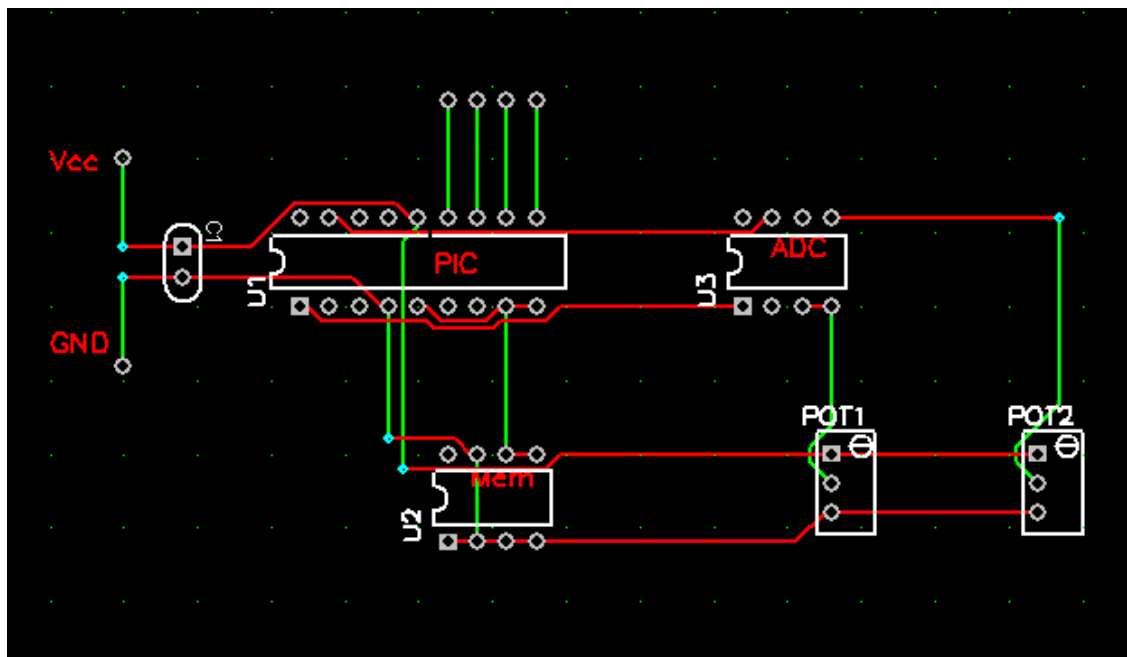


Figura 18. Módulo de adquisición y almacenamiento

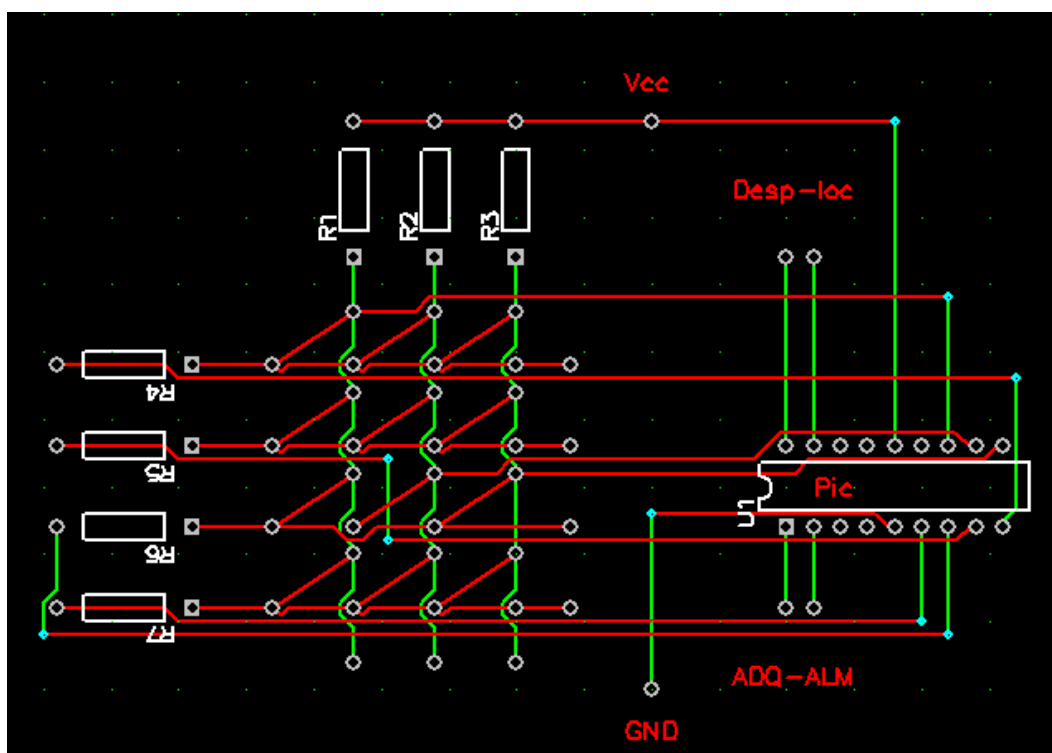


Figura 19. Módulo I/O teclado local

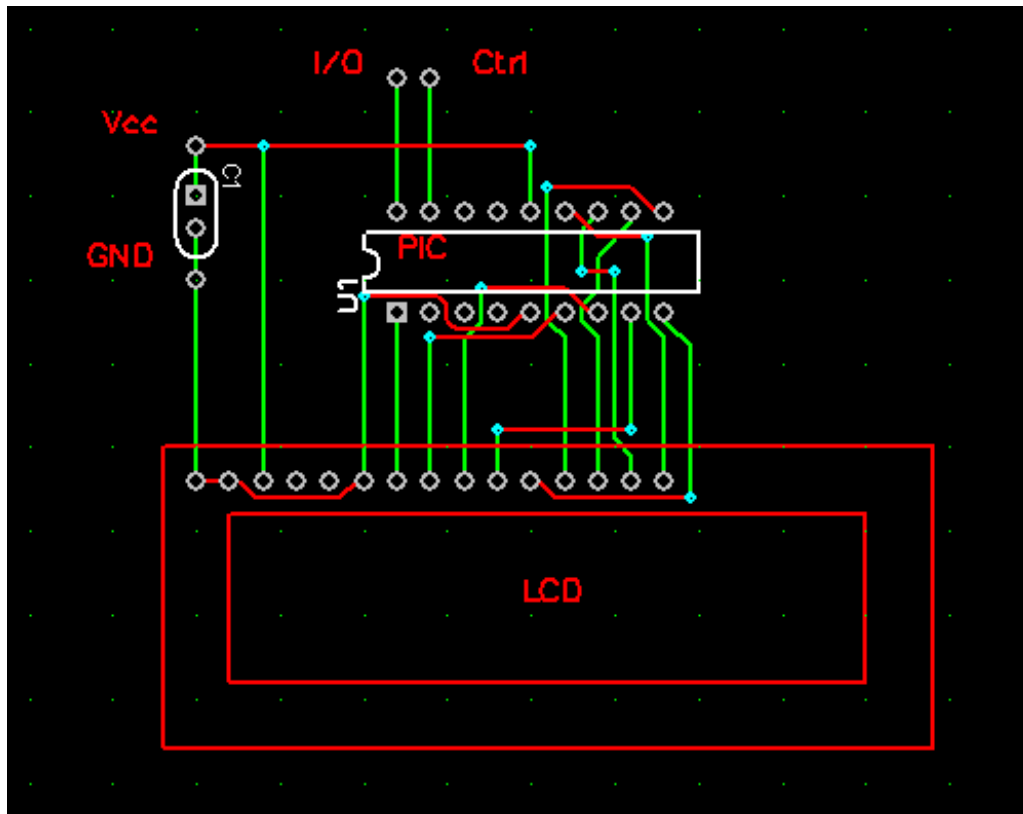


Figura 20. Módulo de despliegue

4. Pruebas de funcionamiento.

El prototipo del sistema smeeep fue puesto a prueba realizando diversas operaciones básicas de funcionamiento, conectándole un primer instrumento prototipo construido para tal fin y será reportado en el manual de funcionamiento de voltímetro para el sistema smeeep que esta en elaboración, desarrollándose pruebas de comunicación con PC para el módulo de comunicaciones, introduciendo datos al despliegue desde el módulo I/O local, se almacenó información en su memoria no volátil y posteriormente se transmitió a la PC, así mismo se operó el sistema en operación simultánea con PC y con teclado. Se utilizaron las primeras interfaces de depuración auxiliar con el fin de validar los datos capturados, almacenados, transmitidos a PC y a despliegue local, así como los provenientes de la interfaz I/O local.

Conclusiones

La realización de este primer prototipo del sistema smeeep nos ha proporcionado una serie de resultados que listamos a continuación:

La experiencia redundó en la realización de un equipo robusto en su funcionamiento y que soportó un uso intensivo durante más de una semana de pruebas sin reportar fallas durante la aplicación de los protocolos de verificación propuestos, obteniendo al final de todo este trabajo un producto que bien puede ser usado como una herramienta auxiliar para la elaboración de experiencias prácticas en laboratorios escolares.

El prototipo construido no solo resultó en una experiencia útil para la adquisición de un acervo práctico importante, pues nos permitió introducirnos a ambientes de trabajo multiprocesador, sino que también nos permitió empezar a desarrollar protocolos de comunicación para estos ambientes, en los cuales no contabamos con experiencia previa.

La realización de pruebas con un primer instrumento de medición (voltímetro), nos permitió observar de mejor forma la problemática para la realización de los otros instrumentos de medición mencionados al principio del trabajo, y que aunque fue bien planteada, fue ampliamente enriquecida durante la elaboración y programación del sistema principal en conjunto con el voltímetro, abriendonos un abánico de alternativas para la elaboración de otros prototipos mas avanzados.

6. Apéndices

6.1. Listado de programas

Modulo de control y comunicaciones

;-----modulo de control y comunicaciones-----

```
tmr0    equ    0x01
status  equ    0x03
porta   equ    0x05
portb   equ    0x06
intcon  equ    0x0b
sendreg equ    0x0c
count   equ    0x0d
instr   equ    0x0e
char    equ    0x0f
addr    equ    0x10
count1  equ    0x11
digctr  equ    0x12
rowctr  equ    0x13
colctr  equ    0x14
rowbits equ    0x15
colbits equ    0x16
temp    equ    0x17
ncount  equ    0x18
mcount  equ    0x19
rcvreg  equ    0x1a
ctrlk   equ    0x1b
vez     equ    0x1c
vsa     equ    0x1d
cent    equ    0x1e
dec     equ    0x1f
uno     equ    0x20
temp2   equ    0x21
temp3   equ    0x22
optreg  equ    0x81
trisa   equ    0x85
trisb   equ    0x86
```

;-----

; mapa de bits

```
c      equ    0
z      equ    2
rp0    equ    5
```

;-----

;

```
org 0x00
goto start
```

```
org 0x004
goto iserv
```

```
start  bsf     status,rp0      ;cambia a banco1
        movlw  0x09           ;1001 0 antes 1000 0 0x01 definición de lineas de entrada y salida en
puertoa
        movwf  trisa
        movlw  0x70           ;definición de entrada salida portb
```

```

movwf trisb
bcf status,rp0 ;cambia a banco0
bsf porta,2
bsf porta,1 ;nivel de "marca" de salida, bit1 LCD
bsf portb,0 ;renglones en alto (1)
bsf portb,1 ;renglones en alto (1)
bsf portb,2 ;renglones en alto (1)
bsf portb,3 ;renglones en alto (1)
bsf portb,7 ;línea no utilizada en alto
bcf intcon,7 ;deshab. ints. global
clrf vez
ckpad call scan10
sem movlw 0x04 ;indica LCD Hex Byte sigue

; movwf sendreg
; call ser_out
; movf digctr,w;obten el dígito
; movwf sendreg
; call ser_out

; btfsc digctr,3
; goto txs
; btfsc digctr,2
; goto txn
; btfss digctr,2
; goto icd
; movlw 0x00
; movwf sendreg
; call ser_outk
; btfss porta,3 ;sem
; call ser_in
; movlw 0x04 ;indica LCD Hex Byte sigue
; movwf instr
; movf rcvreg,w ;obten el dígito
; movwf char
; call sndstf
; movf digctr,w;obten el dígito
; movwf sendreg
sync btfsc porta,3
call ser_outk
btfss porta,3 ;sem
call ser_in

; movf rcvreg,w ;obten el dígito
; call esc_100
; movf dec,w

; movwf char
; movlw 0x04 ;indica LCD Hex Byte sigue
; movwf instr
; call sndstf

; movf uno,w
; movwf char
; movlw 0x04
; movwf instr

```

```

        call    sndstf

        call    debounce      ;retraso temporal, estabiliza los interruptores del teclado
        goto    ckpad         ;espera otro dato
;txs    bsf     porta,2
;
;    movwf    sendreg
;    call     ser_outk
;    goto     sem
;txn    bcf     porta,2
;    movwf    sendreg
;    call     ser_outk
;    goto     sem
;-----
scan10   bcf intcon,2 ;limpia bandera desbordamiento TMRO
        bsf intcon,7 ;hab ints global
        bsf intcon,5 ;hab int tmr
        bsf     portb,0 ;renglones en alto (1)
        bsf     portb,1 ;renglones en alto (1)
        bsf     portb,2 ;renglones en alto (1)
        bsf     portb,3 ;renglones en alto (1)
        clrf    digctr ;contador de dígito en cero
        bcf     portb,3 ;renglon 4en bajo (0)
        btfss   portb,5 ;verifica columna 2
        return  ;se presionó tecla cero (0)
        bsf     portb,3 ;deselecciona renglon 4
        movlw   0x01
        movwf   digctr ;contador de dígito 1
        movwf   rowctr ;contador de renglon en 1
        movwf   rowbits ;=0000 0001
rowout   movf    rowbits,w ;obten var. rowbits
        xorlw   0x0f ;complementa var. rowbits
        movwf   portb ;salida a var. rowbits
        movlw   0x01
        movwf   colctr ;contador de columna a cero
        movlw   0x10 ;=0001 0000
        movwf   colbits ;columna=1
tstcol   movf    portb,w ;lee puertob
        andlw   0x70 ;complementa los bits de columna
        movwf   temp ;temporal tiene var columnas
        movf    colbits,w ;ten bits de columna
        xorlw   0x70 ;complementa los bits de columna
        subwf   temp,w ;compara con el contenido de temp
        btfsc   status,z
        return  ;dígito leído disponible
lastc    movf    colctr,w ;obten la cuenta de columna
        sublw   0x03
        btfsc   status,z ;=3?
        goto    lastr
        rlf     colbits,f ;desplaza los bits de columna
        bcf     colbits,0 ;arregla basura en el acarreo
        incf    colctr,f
        incf    digctr,f
        goto    tstcol
lastr    movf    rowctr,w ;obten cuenta del renglon
        sublw   0x03
        btfsc   status,z ;=3?

```

```

        goto    scan10 ;verifica nuevamente tecla orimida
        rlf     rowbits,f      ;desplaza los bits de renglon
        bcf     rowbits,0      ;arregla basura en el acarreo
        incf    rowctr,f
        incf    digctr,f
        goto    rowout
;-----
debounce    movlw  0xff      ;M
            movwf  mcount ;carga contador M
loadn       movlw  0xff      ;N
            movwf  ncount ;carga contador N
decn        decfsz ncount,f   ;decrementa contador N
            goto   decn      ;repite
            decfsz mcount,f   ;decrementa M
            goto   loadn     ;repite
            return ;fin de rutina de rebote
;-----
sndstf      movf   instr,w    ;obten instrucción
            movwf  sendreg     ;a ser enviada
            call   ser_out ;a la subrutina de serialización de salida
            movf   char,w     ;obtén caaracter o byte en HEX
            movwf  sendreg     ;a enviarse
            call   ser_out ;a la subrutina de serialización de salida
            movf   addr,w     ;obtén dirección
            movwf  sendreg     ;a ser enviada
            call   ser_out ;a la rutina de serialización de salida
            return
;-----
ser_out     bcf     intcon,5;deshabilita las interrupciones del timer0
            bcf     intcon,7;deshabilitación global de interrupciones
            clrf    tmr0      ;limpia el contador temporizador
            clrwdt      ;limpia el watchdog prepara asignación preescalador
            bsf     status,rp0 ;cambia a banco1
            movlw   0xd8     ;configura el contador-temporizador
            movwf   optreg
            bcf     status,rp0 ;cambia a banco0
            movlw   0x08     ;inicia contador de corrimiento
            movwf   count
            bcf     porta,1 ;bit de inicio
            clrf    tmr0     ;limpia-arranca contador-temporizador
            bcf     intcon,2;limpia la bandera de desbordamiento del temporizador-cont.
time1       btfss   intcon,2;desbordamiento de la bandera del contador-tempor?
            goto    time1    ;aún no
            bcf     intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit      rlf     sendreg,f ;rota el BMS en el acarreo
            bcf     porta,1 ;limpia el bit1 del puerto a
            btfsc   status,c;verifica estado del la bandera de acarreo
            bsf     porta,1 ;el bit esta en alto
time2       btfss   intcon,2;desbordamiento del C-T?
            goto    time2    ;no
            bcf     intcon,2;limpia bandera desbord. C-T
            decfsz  count,f ;ya se recorrieron los 8 bits?
            goto    nxtbit   ;no
            bsf     porta,1 ;si, puerto a bit 1 en "marca"
time3       btfss   intcon,2;desbordamiento del C-T?
            goto    time3    ;no

```

```

        return ;finalizado
;-----
ser_in bcf intcon,5;deshabilita interrupciones del TMR0
      bcf intcon,7;deshabilita interrupciones globales
      clrf tmr0 ;limpia y arranca el C-T
      clrwdt ;limpia watchdog y prepara asig. del preescalador
      bsf status,rp0 ;al banco1
      movlw 0xd8 ;configura el C-T
      movwf optreg
      bcf status,rp0 ;regresa al banco 0
      movlw 0x08 ;inicia el contador de corrimiento
      movwf count
sbit btfsc porta,3
      call dest
; bcf intcon,2 ;limpia bandera desbordamiento TMR0
; bsf intcon,7 ;hab ints global
; bsf intcon,5

;sbit
      btfsc porta,0 ;aqui estaba sbit antes inc call dest espera bit de inicio
      goto sbit ;esta en "marca" (1)

; bcf intcon,5;deshabilita interrupciones del TMR0
; bcf intcon,7

      movlw 0x80 ;bit de inicio recibido, espera medio tiempo
      movwf tmr0 ;carga e inicia el TMR0
      bcf intcon,2;limpia la bandera de desbordamiento del TMR0
time11 btfss intcon,2;desbordamiento de la bandera del TMR0?
      goto time11
      btfsc porta,0 ;el bit de inicio aún en bajo?
      goto sbit ;arranque en falso reiniciar
      clrf tmr0 ;si, bit de medio tiempo, inicia el TMR0
      bcf intcon,2;limpia bandera desbordamiento TMR0
time21 btfss intcon,2;desbordamiento del TMR0?
      goto time21
      bcf intcon,2;si limpia la bandera de desbordamiento del tMR0
      movf porta,w ;lee puerto a
      movwf temp ;almacena el dato
      rrf temp,f ;trota el bit cero en el bit de acarreo
      rlf rcvreg,f ;rota el bit de acarreo en el registro de recup.
      decfsz count,f ;roto 8 posc?
      goto time21 ;no
time31 btfss intcon,2;desbordamiento del TMR0?
      goto time31 ;no
      return ;si, byte recibido
;-----
iserv bcf intcon,2 ;limp bdr tmr
      bcf intcon,7 ;deshab. ints. global
      goto sem ;inc temp medidor dist
; retfie
;-----
;-----
ser_outk bcf intcon,5;deshabilita las interrupciones del timer0
        bcf intcon,7;deshabilitación global de interrupciones

```

```

    clrf    tmr0    ;limpia el contador temporizador
    clrwdt    ;limpia el watchdog prepara asignación preescalador
    bsf     status,rp0    ;cambia a banco1
    movlw   0xd8    ;configura el contador-temporizador
    movwf   optreg
    bcf     status,rp0    ;cambia a banco0
    movlw   0x08    ;inicia contador de corrimiento
    movwf   count
    bcf     porta,2    ;bit de inicio
    clrf    tmr0    ;limpia-arranca contador-temporizador
    bcf     intcon,2;limpia la bancera de desbordamiento del temporizador-cont.
time12    btfss   intcon,2;desbordamiento de la bandera del contador-tempor?
    goto   time12    ;aún no
    bcf     intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit2   rlf     sendreg,f    ;rota el BMS en el acarreo
    bcf     porta,2    ;limpia el bit1 del puerto a
    btfsc   status,c;verifica estado de la bandera de acarreo
    bsf     porta,2    ;el bit esta en alto
time22    btfss   intcon,2;desbordamiento del C-T?
    goto   time22    ;no
    bcf     intcon,2;limpia bandera desbord. C-T
    decfsz  count,f    ;ya se recorrieron los 8 bits?
    goto   nxtbit2    ;no
    bsf     porta,2    ;si, puerto a bit 1 en "marca"
time32    btfss   intcon,2;desbordamiento del C-T?
    goto   time32    ;no
    return    ;finalizado

;-----
dest      movf    digctr,w;obten el dígito
    movwf   sendreg
    call    ser_outk
    return

;-----
conv_bcd  movf    vsa,w
    movwf   temp3
x100      movlw   0x64    ;carga valor de centenas
    subwf   temp3,f
    movf    temp3,w    ;guarda residuo en auxiliar
    movwf   temp    ;guardado
    btfsc   status,c;resultado neg, menor que 100?
    goto   inc_cent
    movf    temp,w    ;prepara carga de decenas
    movwf   temp3    ;operando en temp3
x10       movlw   0x0a    ;carga diez x comparar
    subwf   temp3,f;guarda
    movf    temp3,w    ;residuo
    movwf   temp    ;en temp
    btfsc   status,c;menor que 10?
    goto   inc_dec;no
    movf    temp,w    ;si
    movwf   uno
    return

inc_cent  incf     cent
    goto   x100
inc_dec  incf     dec
    goto   x10

```

```

;-----
esc_100      clrf    cent
             clrf    dec
             clrf    uno
             movf    rcvreg,w
             movwf   vsa
             movwf   temp3
cont100      movlw   0x0a
             subwf   temp3,f
             movf    temp3,w
             movwf   temp    ;residuo
             btfsc   status,c
             goto    dec100
             rlf     dec     ;x2
             movlw   0x04
             subwf   temp,f ;residuo
             btfsc   status,c;menor que 5?
             incf    uno    ;si
             movf    dec,w
             subwf   vsa,f  ;vsa-2dec
             movf    uno,w
             subwf   vsa,f  ;vsa-dec-uno
             movlw   0x03
             subwf   vsa,f  ;vsa-2dec-uno-3
             call    conv_bcd
             return

dec100 incf dec
      gotocont100

;-----
      end
;-----

```

Modulo I/O entrada local

```

;=====modulo de entrada local=====
;
;-----
;
;          mapa de memoria
tmr0      equ    0x01
status    equ    0x03
porta     equ    0x05
portb     equ    0x06
intcon    equ    0x0b
sendreg   equ    0x0c
count     equ    0x0d
instr     equ    0x0e
char      equ    0x0f
addr      equ    0x10
count1    equ    0x11
digctr    equ    0x12
rowctr    equ    0x13
colctr    equ    0x14

```

```

rowbits equ    0x15
colbits  equ    0x16
temp     equ    0x17
ncount   equ    0x18
mcount   equ    0x19
optreg   equ    0x81
trisa     equ    0x85
trisb     equ    0x86
;-----
;
;                               mapa de bits
c         equ    0
z         equ    2
rp0       equ    5
;-----
;
;                               org    0x00
;
start     bsf     status,rp0      ;cambia a banco1
          movlw   0x00      ;definición de líneas de entrada y salida en puertoA
          movwf   trisa
          movlw   0x70      ;definición de entrada salida portB
          movwf   trisb
          bcf     status,rp0      ;cambia a banco0
          bsf     porta,1 ;nivel de "marca" de salida, bit1 LCD
          bsf     portb,0 ;renglones en alto (1)
          bsf     portb,1 ;renglones en alto (1)
          bsf     portb,2 ;renglones en alto (1)
          bsf     portb,3 ;renglones en alto (1)
          bsf     portb,7 ;línea no utilizada en alto
ckpad     call    scan10
          movlw   0x04      ;indica LCD Hex Byte sigue
          movwf   instr
          movf    digctr,w ;obten el dígito
          movwf   char
          call    sndstf ;llama rutina envía datos (send stuff)
          call    debounce ;retraso temporal, estabiliza los interruptores del teclado
          goto    ckpad ;espera otro dato
;-----
scan10    bsf     portb,0 ;renglones en alto (1)
          bsf     portb,1 ;renglones en alto (1)
          bsf     portb,2 ;renglones en alto (1)
          bsf     portb,3 ;renglones en alto (1)
          clrf    digctr ;contador de dígito en cero
          bcf     portb,3 ;renglon 4en bajo (0)
          btfss   portb,5 ;verifica columna 2
          return  ;se presionó tecla cero (0)
          bsf     portb,3 ;deselecciona renglon 4
          movlw   0x01
          movwf   digctr ;contador de dígito 1
          movwf   rowctr ;contador de renglon en 1
          movwf   rowbits ;=0000 0001
rowout     movf    rowbits,w ;obten var. rowbits
          xorlw   0x0f ;complementa var. rowbits
          movwf   portb ;salida a var. rowbits
          movlw   0x01
          movwf   colctr ;contador de columna a cero
          movlw   0x10 ;=0001 0000

```

```

movwf colbits ;columna=1
tstcol movf portb,w ;lee puertob
andlw 0x70 ;complementa los bits de columna
movwf temp ;temporal tiene var columnas
movf colbits,w ;ten bits de columna
xorlw 0x70 ;complementa los bits de columna
subwf temp,w ;compara con el contenido de temp
btfsc status,z
return ;digito leído disponible
lastc movf colctr,w ;obten la cuenta de columnaA
sublw 0x03
btfsc status,z ;=3?
goto lastr
rlf colbits,f ;desplaza los bits de columna
bcf colbits,0 ;arregla basura en el acarreo
incf colctr,f
incf digctr,f
goto tstcol
lastr movf rowctr,w ;obten cuenta del renglon
sublw 0x03
btfsc status,z ;=3?
goto scan10 ;verifica nuevamente tecla orimida
rlf rowbits,f ;desplaza los bits de renglon
bcf rowbits,0 ;arregla basura en el acarreo
incf rowctr,f
incf digctr,f
goto rowout
;-----
debounce movlw 0xff ;M
movwf mcount ;carga contador M
loadn movlw 0xff ;N
movwf ncount ;carga contador N
decn decfsz ncount,f ;decrementa contador N
goto decn ;repite
decfsz mcount,f ;decrementa M
goto loadn ;repite
return ;fin de rutina de rebote
;-----
sndstf movf instr,w ;obten instrucción
movwf sendreg ;a ser enviada
call ser_out ;a la subrutina de serialización de salida
movf char,w ;obten caaracter o byte en HEX
movwf sendreg ;a enviarse
call ser_out ;a la subrutina de serialización de salida
movf addr,w ;obten dirección
movwf sendreg ;a ser enviada
call ser_out ;a la rutina de serialización de salida
return
;-----
ser_out bcf intcon,5 ;deshabilita las interrupciones del timer0
bcf intcon,7 ;deshabilitación global de interrupciones
clrf tmr0 ;limpia el contador temporizador
clrwtdt ;limpia el watchdog prepara asignación preescalador
bsf status,rp0 ;cambia a banco1
movlw 0xd8 ;configura el contador-temporizador
movwf optreg

```

```

        bcf     status,rp0      ;cambia a banco0
        movlw  0x08      ;inicia contador de corrimiento
        movwf  count
        bcf     porta,1      ;bit de inicio
        clrf   tmr0        ;limpia-arranca contador-temporizador
        bcf     intcon,2;limpia la bandera de desbordamiento del temporizador-cont.
time1    btfss  intcon,2;desbordamiento de la bandera del contador-tempor?
        goto   time1      ;aún no
        bcf     intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit   rlf     sendreg,f      ;rota el BMS en el acarreo
        bcf     porta,1      ;limpia el bit1 del puerto a
        btfsc  status,c;verifica estado de la bandera de acarreo
        bsf     porta,1      ;el bit esta en alto
time2    btfss  intcon,2;desbordamiento del C-T?
        goto   time2      ;no
        bcf     intcon,2;limpia bandera desbord. C-T
        decfsz count,f      ;ya se recorrieron los 8 bits?
        goto   nxtbit      ;no
        bsf     porta,1      ;si, puerto a bit 1 en "marca"
time3    btfss  intcon,2;desbordamiento del C-T?
        goto   time3      ;no
        return ;finalizado
;-----
;
;               end
;-----

```

Modulo de despliegue

```

;=====modulo de despliegue local=====
;
;
;-----
;               Mapa de memoria
indf     equ    0x00
tmr0     equ    0x01
pc       equ    0x02
status   equ    0x03
fsr      equ    0x04
porta    equ    0x05
portb    equ    0x06
intcon   equ    0x0b
hexbyte  equ    0x0c
ms_dig   equ    0x0d
ls_dig   equ    0x0e
hold     equ    0x0f
sa       equ    0x10
sb       equ    0x11
sc       equ    0x12
sd       equ    0x13
count1   equ    0x14
count2   equ    0x15
rcvreg   equ    0x16
count    equ    0x17
temp     equ    0x18

```

```

instr    equ    0x19
char     equ    0x1a
addr     equ    0x1b
holdbits equ    0x1c
optreg   equ    0x81
trisa    equ    0x85
trisb    equ    0x86
;-----
;      mapa de bits
c        equ    0
z        equ    2
rp0      equ    5
;-----
;      org      0x00
;
start    bsf     status,rp0      ;cambia a banco 1
        movlw   0x01      ;configuracion de entrada y salidas en puertoa
        tris    porta
        movlw   0x00      ;puertob salidas
        movwf   trisb
        tris    portb
        bcf     status,rp0      ;cambia a banco 0
        movlw   0x00      ;prepara las salidas de puertos a cero
        movwf   portb
        bcf     porta,1 ;salidas de bits puertoa a cero
        bcf     porta,2
        bcf     porta,3
        bcf     porta,4
        call    blanks ;llena la memoria RAM de "blancos"
        call    hello  ;crea el mensaje "HELLO" en memoria RAM
        call    del_5  ;da temporización de inicio del LCD
        call    initlcd ;inicializa despliegue
        call    disp16 ;encía 16 char. a LCD
enterm   clrf     rcvreg ;si
        call    ser_in ;llama subrutina de entrada serializada
        movf    rcvreg,w ;obten el byte recibido
        movwf   instr ;almacena la instrucción
        clrf    rcvreg
        call    ser_in ;llama la rutina de serialización de entrada
        movf    rcvreg,w ;obten el byte recibido
        movwf   char ;almacena el char. o byte
        clrf    rcvreg
        call    ser_in ;llama subrutina de serialización de entrada
        movf    rcvreg,w ;obten el byte recibido
        movwf   addr ;almacena la dirección
        movf    instr,w ;obten una copia de la instrucción deseada
        sublw   0x00 ;compara con 0x00
        btfsc   status,z;Z esta en 1 si son iguales
        goto    blnkram ;los byte son iguales
        movf    instr,w ;obten copia de la instrucción
        sublw   0x01 ;compara con 0x01
        btfsc   status,z;la bandera de Z esta en alto y son iguales
        goto    send16 ;los bytes son iguales
        movf    instr,w ;obten copia de la instrucción
        sublw   0x02 ;compara con 0x02
        btfsc   status,z;la bandera de Z esta en alto los bytes son iguales

```

```

    goto    sndtst ;los bytes son iguales
    movf    instr,w ;obten una copia de la instrucción
    sublw   0x03 ;compara con 0x03
    btfsc   status,z;la bandera de Z esta en alto los bytes osn iguales
    goto    chr_ram ;los bytes son iguales
    movf    instr,w ;obten una copia de la instrucción
    sublw   0x04 ;compara con 0x04
    btfsc   status,z;la bandera de Z esta en alto los bytes osn iguales
    goto    convhex ;los bytes son iguales
    goto    enterm ;espera por la próxima transmisión
;-----
blnkram    call    blanks ;llena la memoria RAM de "blancos"
    goto    enterm ;regresa al programa principal
;-----
send16 call    disp16 ;envia el contenido de la RAM al LCD
    goto    enterm ;regresa al programa principal
;-----
sndtst call    test ;carga la RAM con "TEST"
    call    disp16;envía el contenido e la RAM al LCD
    goto    enterm ;regresal al programa principal
;-----
chr_ram    movf    addr,w ;obten una copia de la dirección RAM
    movwf    fsr ;almacenala en el FSR
    movf    char,w ;obten una copia del char a ser desplegada
    movwf    indf ;a la dirección RAM apuntada por el FSR
    goto    enterm ;regresa al programa principal
;-----
convhex    movf    char,w ;obten una copia del carater hexadecimal a ser enviado
    call    disp16;convierte el byte a ser desplegado
    call    disp16 ;envía el contenido e la RAM al LCD
    goto    enterm ;regresa al programa principal
;-----
disphexmovwf hexbyte;almacena una copuia del byte HEX
    call    blanks ;llena el despliegue de "blancos"
    call    sephex ;separa el byte HEX en dos dígitos ASCII
    movf    ms_dig,w ;obten el digito mas significativo
    movwf    0x20 ;a la memoria RAM
    movf    ls_dig,w ;obten el dígito menos sihnificativo
    movwf    0x21 ; a la memoria RAM
    swapf    hexbyte,w ;obten una copua del byte HEX e intercambia MS-LS
    call    hexbits ;llama rutina de HEX a bits
    movf    sa,w ;obten el primer bit
    movwf    0x23 ;a la RAM
    movf    sb,w ;obten el segundo bit
    movwf    0x24 ;a la RAM
    movf    sc,w ;obten el tercer bit
    movwf    0x25 ;a la RAM
    movf    sd,w ;obten el cuarto bit
    movwf    0x26 ;a la RAM
    movf    hexbyte,w ;obten una copia del byte HEX
    call    hexbits ;llama rutina HEX a bits
    movf    sa,w ;obten el primer bit
    movwf    0x28 ;a la RAM
    movf    sb,w ;obten el segundo bit
    movwf    0x29 ;a la RAM
    movf    sc,w ;obten el tercer bit

```

```

movwf 0x2a ;a la RAM
movf sd,w ;obten el cuarto bit
movwf 0x2b ;a la RAM
return
;-----
sephex movf hexbyte,w ;obtén copia del HEX byte
andlw 0x0f ;enmaskara el nibble mas alto
call hex2asc ;llama conversión HEX to ASCII }
movwf ls_dig ;almacena el LSN
swapf hexbyte,w ;obten una copia del HEX byte intercambia MS-LS
andlw 0x0f ;enmascara el nibble mas alto
call hex2asc ;llama rutina de conversión HEX to ASCII
movwf ms_dig ;almacenalo
return
;-----
hex2asc movwf hold ;almacena una copia del dígito HEX
sublw 0x09 ;sustraer el acumulador de uno menos que 0x0a
btfss status,c ;el acarreo esta en alto si W es menor de 0x0a
goto add37
goto add30
add37 movf hold,w ;obten el dígito HEX
addlw 0x37
return
add30 movf hold,w ;obten el dígito HEX
addlw 0x30
return ;regresa con char ASCII en W
;-----
hexbits movwf hold ;salva una copia del dígito HEX
movlw sd ;inicia con la ultima dirección de las notas
movwf fsr ;almacenalo en el FSR
movlw 0xf0 ;pon en uno los cuatro dígitos +significativos
iorwf hold,w ;se usan unos como marcadores
movwf holdbits ;y almacenalo
btf status,c ;limpia el acarreo para que tenga cero cuando se gire
bitslp movlw 0x30 ;asumir que el cero en asciies necesario
btfsc holdbits,0 ; si el LSB es cero entonces el ascii cero es correcto
;brinca la siguiente instrucción
movlw 0x31 ;se necesita el ascii 1 en cambio
movwf indf ;0 o 1 para indicar la nota en l borrador al FSR
decf fsr,f ;apunta al siguiente nota del borrador que almacenara
;el char. ascii para el próximo bit más significativo
rrf holdbits,f ;rota el lsb fuera y rota el bit marcadores dentro

btfsc holdbits,4 ;es el bit 4 un cero? si es asi viene del acarreo
; y hemos finalizado
goto bitslp ;si no es así continúa hasta terminar
return
;-----
blanks movlw 0x10 ;cuenta a 16
movwf count1
movlw 0x20 ;primera dirección RAM de despliegue
movwf fsr ;direccionamiento indexado
movlw 0x20 ;ascii "blanco"
store movwf indf ;almacena el la localidad RAM indicada por el FSR
decfsz count1,f ;16?
goto incfsr ;no

```

```

        return    ;si finalizado
incfsr  incf      fsr,f    ;incrementa el fsr
        goto     store

;-----
hello   movlw    "H"
        movwf    0x20
        movlw    "E"
        movwf    0x21
        movlw    "L"
        movwf    0x22
        movwf    0x23
        movlw    "O"
        movwf    0x24
        return

;-----
test    movlw    "T"
        movwf    0x20
        movwf    0x23
        movlw    "E"
        movwf    0x21
        movlw    "S"
        movwf    0x22
        movlw    " "
        movwf    0x24
        return

;-----
initlcd bcf      porta,1 ;manda la linea de enable a 0
        bcf      porta,2 ;linea RS a 0, para configurar entrada a instrucciones
        call     del_125 ;retraso de 125us (vs 39us del nuestro)
        movlw    0x38 ;selecciona el modoe de comunic. 8 bits char. 5x7
        movwf    portb ;0011 1000
        call     pulse ;rutina para enviar correctamente datos al lcd y
                        ;esperar su estabilizacion
        movlw    0x0c ;display encendido, cursor parpadeando
        movwf    portb ;salida 0000 1100
        call     pulse
        movlw    0x06 ;instruccion para borar el LCD
        movwf    portb ;0000 0110
        call     pulse
        call     del_5
        return

;-----
disp16  bcf      porta,1 ;linea E en bajo
        bcf      porta,2 ;línea RS en bajo para control
        call     del_125
        movlw    0x80 ;palabra de control primera mitad
        movwf    portb
        call     pulse
        bsf      porta,2 ;RS en 1 para datos
        call     del_125
        movlw    0x20 ;inicia FSR
        movwf    fsr
getchar movf     0x00,w ;obten el caracter almacenado en la localidad 16
        movwf    portb
        call     pulse
        movlw    0x27 ;se envio ya el 8vo char?

```

```

        subwf    fsr,w    ;sustraer w del fsr
        btfsc    status,z;verifica la bandera de Z
        goto     half
        movlw    0x2f    ;número de prueba
        subwf    fsr,w    ;sustraer w del fsr
        btfsc    status,z;verifica la bandera de Z
        return    ;16 char. al LCD
        incf     fsr,f    ;desplazate a la siguiente localidad
        goto     getchar
half    bcf       porta,2 ;RS=0 para control
        call     del_125
        movlw    0xc0    ;palabra de control LC segunda mitad
        movwf    portb
        call     pulse
        bsf       porta,2 ;RS=1 para datos
        incf     fsr,f    ;incrementa el FSR al próximo registro
        call     del_125
        goto     getchar
;-----
del_125    movlw    0x2a    ;retrasa aprox. 42x3 ciclos (decimal)
        movwf    count1 ;carga el contador
repeat    decfsz    count1,f ;decrementa el contador a cero
        goto     repeat ;no cero aun
        return
;-----
del_5     movlw    0x28    ;40 en decimal
        movwf    count2 ;carga el valor al contador
delay    call     del_125    ;rutina de retraso de 125us
        decfsz    count2,f    ;rutina de retraso (40x125us) aprox 5 ms
        goto     delay
        return    ;contador 2 a cero
;-----
pulse    bsf       porta,1 ;da el pulso de salida a linea enable
        nop
        bcf       porta,1
        call     del_125    ;espera la estabilizacion del lcd tiempo de espera
        return
;-----
ser_in    bcf       intcon,5;deshabilita interrupciones del TMR0
        bcf       intcon,7;deshabilita interrupciones globales
        clrf      tmr0    ;limpia y arranca el C-T
        clrwdt    ;limpia watchdog y prepara asig. del preescalador
        bsf       status,rp0 ;al banco1
        movlw    0xd8    ;configura el C-T
        movwf    optreg
        bcf       status,rp0 ;regresa al banco 0
        movlw    0x08    ;inicia el contador de corrimiento
        movwf    count
sbit     btfsc    porta,0 ;espera bit de inicio
        goto     sbit    ;esta en "marca" (1)
        movlw    0x80    ;bit de inicio recibido, espera medio tiempo
        movwf    tmr0    ;carga e inicia el TMR0
        bcf       intcon,2;limpia la bandera de desbordamiento del TMR0
time1    btfss    intcon,2;desbordamiento de la bandera del TMR0?
        goto     time1
        btfsc    porta,0 ;el bit de inicio aún en bajo?

```

```

        goto    sbit    ;arranque en falso reiniciar
        clrf    tmr0    ;si, bit de medio tiempo, inicia el TMR0
        bcf     intcon,2;limpia bandera desbordamiento TMR0
time2    btfss   intcon,2;desbordamiento del TMR0?
        goto    time2
        bcf     intcon,2;si limpia la bandera de desbordamiento del tMR0
        movf    porta,w ;lee puerto a
        movwf   temp    ;almacena el dato
        rrf     temp,f   ;rota el bit cero en el bit de acarreo
        rlf     rcvreg,f ;rota el bit de acarreo en el registro de recup.
        decfsz  count,f  ;roto 8 posc?
        goto    time2    ;no
time3    btfss   intcon,2;desbordamiento del TMR0?
        goto    time3    ;no
        return   ;si, byte recibido
;-----
        end
;-----

```

Modulo de adquisición y almacenamiento

```

;-----modulo de adquisicion y almacenamiento-----
;cpu equates
tmr0    equ     0x01
porta   equ     0x05
portb   equ     0x06
status  equ     0x03
intcon   equ    0x0b
rcvreg   equ    0x0c
count    equ    0x0d
temp     equ    0x0e
ncount   equ    0x10
mcount   equ    0x11
sendreg   equ    0x12
nyble    equ    0x13
acci     equ    0x14
ah       equ    0x15
al       equ    0x16
w        equ    0x17
r        equ    0x18
optreg   equ    0x81
trisa    equ    0x85
trisb    equ    0x86
;-----
;bit equates
c        equ     0
z        equ     2
rp0      equ     5
;-----
        org     0x00
        goto    start
;
        org 0x004
        goto iserv
;

```

```

start    bsf     status,rp0      ;switch bank1
        movlw   0x01      ;bit0 =entrada
        movwf   trisa
        movlw   0x5A      ;0101 1010;antes 4A 0100 1010; antes-antes 0x40
        movwf   trisb
        bcf     status,rp0      ;cambio al banco 1
        bcf     portb,5
        bsf     portb,7
        bsf     portb,2 ;reloj en alto
        bsf     portb,0 ;linea datos en alto
        clrf    ah          ;apunt mem alto flash 00
        clrf    al          ;apunt low mem flash 00
;        bsf     portb,6
        movlw   0x04      ;0000 0100
        movwf   porta      ;inicializa
        nop      ;retraso
;        clrf    acci
;        btfsc   acci,2
;        goto    paro
;        btfsc   portb,4 ;tx archivo pc?
        call    read
convert  bcf     porta,2 ;T setup, CS low
        nop
        bsf     porta,1 ;clock high
        nop
        bcf     porta,1 ;clock low start conversion
        nop
        bsf     porta,1 ;clock high
        call    ser_in      ;a la subrutina de entrada de datos
        bsf     porta,1 ;ultimo pulso de reloj
        nop
        bcf     porta,1 ;reloj bajo
        nop
        bsf     porta,2 ;CS high
        movf    rcvreg,w      ;obten datos
        addlw   0x80
;        movwf   portb      ;despliegalos
        movf    rcvreg,w
        movwf   sendreg
        movwf   w
        call    write
        call    ser_out
;        btfsc   portb,6
;        goto    notx

;        btfsc   acci,0
;        goto    notx
        movlw   0x02
        movwf   nyble
determina    movf    rcvreg,w
        movwf   temp
        btfss   nyble,1
        swapf   temp,f
        movlw   0xf0
        andwf   temp,f
        btfsc   temp,7

```

```

        goto    af
decimal movf   rcvreg,w
        movwf  temp
        btfss  nyble,1
        swapf  temp,f
        movlw  0xf0
        andwf  temp,f
        swapf  temp,w
        addlw  0x30
envia   movwf  sendreg
        call   ser_out2
        decfsz nyble
        goto   determina
;
;paro   call   ser_in2
        movf   rcvreg,w
        addlw  0x84 ;antes 0x80
        movwf  portb
;
;       movwf  acci
notx    call   debounce ;espera 200 milisegundos
        goto   convert ;busca al ADC otra vez
af      btfsc  temp,5
        goto   siaf
        btfsc  temp,6
        goto   siaf
        goto   decimal
siaf    movf   rcvreg,w
        movwf  temp
        btfss  nyble,1
        swapf  temp,f
        movlw  0xf0
        andwf  temp,f
;
;       btfss  nyble,1
;
;       swapf  temp,f
        swapf  temp,w
        addlw  0x37
        goto   envia
;-----
ser_in  clrf    rcvreg ;limpia registro
        movlw  0x08 ;inicia contador 8 bits
        movwf  count
getbit  bcf     porta,1 ;reloj bajo
        nop
        movf   porta,w ;lee puerto a
        movwf  temp ;almacena copia
        rrf    temp,f ;rota el bit en el acarreo
        rlf    rcvreg,f ;pasa el acarreo al registro de recepcion
        decfsz count,f ;decrementa el contador
        goto   shift
        return ;hecho
shift  bsf     porta,1 ;reloj en alto
        goto   getbit
;-----
debounce movlw  0xfd ;M antes ff 500 us menos por write-read
        movwf  mcount ;al contador M

```

```

loadn    movlw    0xff    ;N
          movwf    ncount ;al contador N
decn     decfsz   ncount,f ;decrementa N
          goto     decn    ;hasta el 0
          decfsz   mcount,f ;decrementa M
          goto     loadn   ;otra vez
          return    ;hecho
;-----
ser_out   bcf     intcon,5;deshabilita las interrupciones del timer0
          bcf     intcon,7;deshabilitación global de interrupciones
          clrf    tmr0    ;limpia el contador temporizador
          clrwdt   ;limpia el watchdog prepara asignación preescalador
          bsf     status,rp0 ;cambia a banco1
          movlw   0xd8    ;configura el contador-temporizador
          movwf   optreg
          bcf     status,rp0 ;cambia a banco0
          movlw   0x08    ;inicia contador de corrimiento
          movwf   count
          bcf     porta,3 ;bit de inicio
          clrf    tmr0    ;limpia-arranca contador-temporizador
          bcf     intcon,2;limpia la bandera de desbordamiento del temporizador-cont.
time1     btfss   intcon,2;desbordamiento de la bandera del contador-tempor?
          goto     time1  ;aún no
          bcf     intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit    rlf     sendreg,f ;rota el BMS en el acarreo
          bcf     porta,3 ;limpia el bit1 del puertoA
          btfsc   status,c;verifica estado de la bandera de acarreo
          bsf     porta,3 ;el bit esta en alto
time2     btfss   intcon,2;desbordamiento del C-T?
          goto     time2  ;no
          bcf     intcon,2;limpia bandera desbord. C-T
          decfsz  count,f ;ya se recorrieron los 8 bits?
          goto     nxtbit ;no
          bsf     porta,3 ;si, puertoA bit 1 en "marca"
time3     btfss   intcon,2;desbordamiento del C-T?
          goto     time3  ;no
          return    ;finalizado
;-----
;-----
ser_out2   bcf     intcon,5;deshabilita las interrupciones del timer0
          bcf     intcon,7;deshabilitación global de interrupciones
          clrf    tmr0    ;limpia el contador temporizador
          clrwdt   ;limpia el watchdog prepara asignación preescalador
          bsf     status,rp0 ;cambia a banco1
          movlw   0xd8    ;configura el contador-temporizador
          movwf   optreg
          bcf     status,rp0 ;cambia a banco0
          movlw   0x08    ;inicia contador de corrimiento
          movwf   count
          bcf     portb,7 ;bit de inicio
          movlw   0x30    ;defina el contador para 4800 bauds
          movwf   tmr0    ;inici el C-T
          bcf     intcon,2;limpia la bandera de desbordamiento del temporizador-cont.
time11     btfss   intcon,2;desbordamiento de la bandera del contador-tempor?
          goto     time11 ;aún no
          movlw   0x30    ;defina el contador para 4800 bauds

```

```

        movwf tmr0    ;inici el C-T
        bcf      intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit1 rrf      sendreg,f    ;rota el LMS en el acarreo
        bcf      portb,7 ;limpia el bit1 del puerto a
        btfsc    status,c;verifica estado del la bandera de acarreo
        bsf      portb,7 ;el bit esta en alto
time21  btfss    intcon,2;desbordamiento del C-T?
        goto     time21 ;no
        movlw    0x30    ;defina el contador para 4800 bauds
        movwf    tmr0    ;inici el C-T
        bcf      intcon,2;limpia bandera desbord. C-T
        decfsz   count,f ;ya se recorrieron los 8 bits?
        goto     nxtbit1 ;no
        bsf      portb,7 ;si, puerto a bit 1 en "marca"
time31  btfss    intcon,2;desbordamiento del C-T?
        goto     time31 ;no
        return    ;finalizado
;-----
;-----
ser_in2 bsf      portb,5
        bcf      intcon,5;deshabilita interrupciones del TMR0
        bcf      intcon,7;deshabilita interrupciones globales
        clrf     tmr0    ;limpia y arranca el C-T
        clrwdt   ;limpia watchdog y prepara asig. del preescalador
        bsf      status,rp0 ;al banco1
        movlw    0xd8    ;configura el C-T
        movwf    optreg
        bcf      status,rp0 ;regresa al banco 0
        movlw    0x08    ;inicia el contador de corrimiento
        movwf    count

; bcf intcon,2 ;limpia bandera desbordamiento TMR0
; bsf intcon,7 ;hab ints global
; bsf intcon,5 ;hab int tmr

sbit3   btfsc    portb,6 ;espera bit de inicio
        goto     sbit3 ;esta en "marca" (1)

;       bcf      intcon,5;deshabilita interrupciones del TMR0
;       bcf      intcon,7

        movlw    0x80    ;bit de inicio recibido, espera medio tiempo
        movwf    tmr0    ;carga e inicia el TMR0
        bcf      intcon,2;limpia la bandera de desbordamiento del TMR0
time113 btfss    intcon,2;desbordamiento de la bandera del TMR0?
        goto     time113
        btfsc    portb,6 ;el bit de inicio aún en bajo?
        goto     sbit3 ;arranque en falso reiniciar
        clrf     tmr0    ;si, bit de medio tiempo, inicia el TMR0
        bcf      intcon,2;limpia bandera desbordamiento TMR0
time213 btfss    intcon,2;desbordamiento del TMR0?
        goto     time213
        bcf      intcon,2;si limpia la bandera de desbordamiento del tMR0
        movf     portb,w
        movwf    temp
        rlf      temp,f

```

```

        rlf      temp,f
        rlf      rcvreg,f ;rota el bit cero en el bit de acarreo
        decfsz   count,f ;roto 8 posc?
        goto     time213      ;no
time313      btfss   intcon,2;desbordamiento del TMR0?
        goto     time313      ;no
        bcf      portb,5
        return    ;si, byte recibido

;-----
;
;      btfss     portb,6 ;lee puerto a
;      goto      uno
;      btfsc     portb,6 ;lee puerto a
;      goto      cero

;uno      bsf      temp,0 ;almacena el dato
;      goto      dato
;cero      bcf      temp,0 ;almacena el dato
;      goto      dato
;-----
;-----
iserv      bcf intcon,2 ;limp bdr tmr
           bcf intcon,7 ;deshab. ints. global
           goto notx ;inc temp medidor dist
;      retfie
;-----
;escritura de memoria
write      bcf      portb,0 ;start
           bcf      portb,2 ;clk abajo
           bsf      portb,0 ;primer bit del comando de escritura 1
           bsf      portb,2 ;detecta primer bit de comando
           bcf      portb,2 ;clk abajo
           bcf      portb,0 ;segundo bit del comando de escritura 0
           bsf      portb,2 ;detecta segundo bit comm wrt
           bcf      portb,2 ;clk abajo
           bsf      portb,0 ;tercer bit comm wrte 1
           bsf      portb,2 ;detecta 3er bit comm wrte
           bcf      portb,2 ;clk low
           bcf      portb,0 ;4to bit comm wrte 0
           bsf      portb,2 ;det 4to bit comm wrte
           bcf      portb,2 ;clk abajo
           bcf      portb,0 ;1er bit adrs 0
           bsf      portb,2 ;det 1er bit adrs
           bcf      portb,2 ;clk low
           bcf      portb,0 ;2do bit adrs 0
           bsf      portb,2 ;det 2do bit adrs
           bcf      portb,2 ;clk low
           bcf      portb,0 ;3er bit adrs
           bsf      portb,2 ;det 3er bit adrs
           bcf      portb,2 ;clk low
           bcf      portb,0 ;bit wrte 0
           bsf      portb,2 ;det bit wrte
           bcf      portb,2 ;clk low
           bsf      portb,0 ;libera bus x ack
           bsf      portb,2 ;det bit ack

```

```

ackcmbt      btfsc  portb,1 ;ack?
              goto  ackcmbt
              bcf    portb,2 ;clk low si ack
              btfsc  ah,7    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,7    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 15
              bcf    portb,2 ;clk low
              btfsc  ah,6    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,6    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 14
              bcf    portb,2 ;clk low
              btfsc  ah,5    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,5    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 13
              bcf    portb,2 ;clk low
              btfsc  ah,4    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,4    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 12
              bcf    portb,2 ;clk low
              btfsc  ah,3    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,3    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 11
              bcf    portb,2 ;clk low
              btfsc  ah,2    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,2    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 10
              bcf    portb,2 ;clk low
              btfsc  ah,1    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,1    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 09
              bcf    portb,2 ;clk low
              btfsc  ah,0    ;direccion memoria high bit7 1
              bsf    portb,0 ;el esta en 1
              btfss  ah,0    ;dir mem alta 0
              bcf    portb,0 ;el bit esta en 0
              bsf    portb,2 ;det bit 08
              bcf    portb,2 ;clk low
              bsf    portb,0 ;libera bus x ack
              bsf    portb,2 ;det ack
ackadh btfsc  portb,1 ;ack
              goto  ackadh ;no
              bcf    portb,2 ;clk low si ack

```

```

btfsc    al,7    ;dir mem low bit 7 1
bsf      portb,0 ;el bit esta en 1
btfss    al,7    ;dir mem low bit 7 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 7
bcf      portb,2 ;clk low
btfsc    al,6    ;dir mem low bit 6 1
bsf      portb,0 ;el bit esta en 1
btfss    al,6    ;dir mem low bit 6 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 6
bcf      portb,2 ;clk low
btfsc    al,5    ;dir mem low bit 5 1
bsf      portb,0 ;el bit esta en 1
btfss    al,5    ;dir mem low bit 5 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 5
bcf      portb,2 ;clk low
btfsc    al,4    ;dir mem low bit 4 1
bsf      portb,0 ;el bit esta en 1
btfss    al,4    ;dir mem low bit 4 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 4
bcf      portb,2 ;clk low
btfsc    al,3    ;dir mem low bit 3 1
bsf      portb,0 ;el bit esta en 1
btfss    al,3    ;dir mem low bit 3 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 3
bcf      portb,2 ;clk low
btfsc    al,2    ;dir mem low bit 2 1
bsf      portb,0 ;el bit esta en 1
btfss    al,2    ;dir mem low bit 2 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 2
bcf      portb,2 ;clk low
btfsc    al,1    ;dir mem low bit 1 1
bsf      portb,0 ;el bit esta en 1
btfss    al,1    ;dir mem low bit 1 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 1
bcf      portb,2 ;clk low
btfsc    al,0    ;dir mem low bit 0 1
bsf      portb,0 ;el bit esta en 1
btfss    al,0    ;dir mem low bit 0 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 0
bcf      portb,2 ;clk low
bsf      portb,0 ;libera bus ack
bsf      portb,2 ;det ack
ackadl   btfsc    portb,1 ;ack?
        goto     ackadl ;no
        bcf      portb,2 ;clk low si ack
        btfsc    w,7    ;bit 7 dato 1
        bsf      portb,0 ;el bit esta en 1
        btfss    w,7    ;bit 7 dato 0

```

```

bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 7
bcf    portb,2 ;clk low
btfsc  w,6     ;bit 6 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,6     ;bit 6 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 6
bcf    portb,2 ;clk low
btfsc  w,5     ;bit 5 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,5     ;bit 5 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 5
bcf    portb,2 ;clk low
btfsc  w,4     ;bit 4 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,4     ;bit 4 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 4
bcf    portb,2 ;clk low
btfsc  w,3     ;bit 3 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,3     ;bit 3 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 3
bcf    portb,2 ;clk low
btfsc  w,2     ;bit 2 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,2     ;bit 2 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 2
bcf    portb,2 ;clk low
btfsc  w,1     ;bit 1 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,1     ;bit 1 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 1
bcf    portb,2 ;clk low
btfsc  w,0     ;bit 0 dato 1
bsf    portb,0 ;el bit esta en 1
btfss  w,0     ;bit 0 dato 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det dato bit 0
bcf    portb,2 ;clk low
bsf    portb,0 ;libera bus ack
bsf    portb,2 ;det ack
ackdat btfsc  portb,1 ;ack
goto   ackdat ;no
bcf    portb,2 ;clk low si ack
bcf    portb,0 ;prepara stop
bsf    portb,2 ;det stop
bsf    portb,0 ;stop
;      bsf    portb,2 ;clk high stop
incfsz al      ;incrementa al
return ;fin comando write

```

```

    incf    ah
    btfss   ah,7    ;mas de 32kbits?
    return
    clrf    ah      ;si, inic cont alto mem
    return
;-----
read      bcf     portb,0 ;start
          bcf     portb,2 ;clk abajo
          bsf     portb,0 ;primer bit del comando de escritura 1
          bsf     portb,2 ;detecta primer bit de comando
          bcf     portb,2 ;clk abajo
          bcf     portb,0 ;segundo bit del comando de escritura 0
          bsf     portb,2 ;detecta segundo bit comm wrt
          bcf     portb,2 ;clk abajo
          bsf     portb,0 ;tercer bit comm wrte 1
          bsf     portb,2 ;detecta 3er bit comm wrte
          bcf     portb,2 ;clk low
          bcf     portb,0 ;4to bit comm wrte 0
          bsf     portb,2 ;det 4to bit comm wrte
          bcf     portb,2 ;clk abajo
          bcf     portb,0 ;1er bit adrs 0
          bsf     portb,2 ;det 1er bit adrs
          bcf     portb,2 ;clk low
          bcf     portb,0 ;2do bit adrs 0
          bsf     portb,2 ;det 2do bit adrs
          bcf     portb,2 ;clk low
          bcf     portb,0 ;3er bit adrs
          bsf     portb,2 ;det 3er bit adrs
          bcf     portb,2 ;clk low
          bcf     portb,0 ;bit wrte 0
          bsf     portb,2 ;det bit wrte
          bcf     portb,2 ;clk low
          bsf     portb,0 ;libera bus x ack
          bsf     portb,2 ;det bit ack
ackcmbtr  btfsc   portb,1 ;ack?
          goto    ackcmbtr
          bcf     portb,2 ;clk low si ack
          btfsc   ah,7    ;direccion memoria high bit7 1
          bsf     portb,0 ;el esta en 1
          btfss   ah,7    ;dir mem alta 0
          bcf     portb,0 ;el bit esta en 0
          bsf     portb,2 ;det bit 15
          bcf     portb,2 ;clk low
          btfsc   ah,6    ;direccion memoria high bit7 1
          bsf     portb,0 ;el esta en 1
          btfss   ah,6    ;dir mem alta 0
          bcf     portb,0 ;el bit esta en 0
          bsf     portb,2 ;det bit 14
          bcf     portb,2 ;clk low
          btfsc   ah,5    ;direccion memoria high bit7 1
          bsf     portb,0 ;el esta en 1
          btfss   ah,5    ;dir mem alta 0
          bcf     portb,0 ;el bit esta en 0
          bsf     portb,2 ;det bit 13
          bcf     portb,2 ;clk low
          btfsc   ah,4    ;direccion memoria high bit7 1

```

```

bsf      portb,0 ;el esta en 1
btfss    ah,4    ;dir mem alta 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 12
bcf      portb,2 ;clk low
btfsc    ah,3    ;direccion memoria high bit7 1
bsf      portb,0 ;el esta en 1
btfss    ah,3    ;dir mem alta 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 11
bcf      portb,2 ;clk low
btfsc    ah,2    ;direccion memoria high bit7 1
bsf      portb,0 ;el esta en 1
btfss    ah,2    ;dir mem alta 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 10
bcf      portb,2 ;clk low
btfsc    ah,1    ;direccion memoria high bit7 1
bsf      portb,0 ;el esta en 1
btfss    ah,1    ;dir mem alta 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 09
bcf      portb,2 ;clk low
btfsc    ah,0    ;direccion memoria high bit7 1
bsf      portb,0 ;el esta en 1
btfss    ah,0    ;dir mem alta 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 08
bcf      portb,2 ;clk low
bsf      portb,0 ;libera bus x ack
bsf      portb,2 ;det ack
ackadhr  btfsc    portb,1 ;ack
goto     ackadhr ;no
bcf      portb,2 ;clk low si ack
btfsc    al,7    ;dir mem low bit 7 1
bsf      portb,0 ;el bit esta en 1
btfss    al,7    ;dir mem low bit 7 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 7
bcf      portb,2 ;clk low
btfsc    al,6    ;dir mem low bit 6 1
bsf      portb,0 ;el bit esta en 1
btfss    al,6    ;dir mem low bit 6 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 6
bcf      portb,2 ;clk low
btfsc    al,5    ;dir mem low bit 5 1
bsf      portb,0 ;el bit esta en 1
btfss    al,5    ;dir mem low bit 5 0
bcf      portb,0 ;el bit esta en 0
bsf      portb,2 ;det bit 5
bcf      portb,2 ;clk low
btfsc    al,4    ;dir mem low bit 4 1
bsf      portb,0 ;el bit esta en 1
btfss    al,4    ;dir mem low bit 4 0
bcf      portb,0 ;el bit esta en 0

```

```

bsf    portb,2 ;det bit 4
bcf    portb,2 ;clk low
btfsc  al,3    ;dir mem low bit 3 1
bsf    portb,0 ;el bit esta en 1
btfss  al,3    ;dir mem low bit 3 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det bit 3
bcf    portb,2 ;clk low
btfsc  al,2    ;dir mem low bit 2 1
bsf    portb,0 ;el bit esta en 1
btfss  al,2    ;dir mem low bit 2 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det bit 2
bcf    portb,2 ;clk low
btfsc  al,1    ;dir mem low bit 1 1
bsf    portb,0 ;el bit esta en 1
btfss  al,1    ;dir mem low bit 1 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det bit 1
bcf    portb,2 ;clk low
btfsc  al,0    ;dir mem low bit 0 1
bsf    portb,0 ;el bit esta en 1
btfss  al,0    ;dir mem low bit 0 0
bcf    portb,0 ;el bit esta en 0
bsf    portb,2 ;det bit 0
bcf    portb,2 ;clk low
bsf    portb,0 ;libera bus ack
bsf    portb,2 ;det ack
ackadlr btfsc  portb,1 ;ack?
goto    ackadlr ;no

bcf    portb,2 ;clk low si ack( inicia lectura)
bsf    portb,0 ;prepara start
bsf    portb,2 ;det start
bcf    portb,0 ;start

bcf    portb,2 ;clk abajo inicia el read desde loc random
bsf    portb,0 ;primer bit del comando de lectura 1
bsf    portb,2 ;detecta primer bit de comando
bcf    portb,2 ;clk abajo
bcf    portb,0 ;segundo bit del comando de lectura 0
bsf    portb,2 ;detecta segundo bit comm read
bcf    portb,2 ;clk abajo
bsf    portb,0 ;tercer bit comm read 1
bsf    portb,2 ;detecta 3er bit comm read
bcf    portb,2 ;clk low
bcf    portb,0 ;4to bit comm read 0
bsf    portb,2 ;det 4to bit comm read
bcf    portb,2 ;clk abajo
bcf    portb,0 ;1er bit adrs 0
bsf    portb,2 ;det 1er bit adrs
bcf    portb,2 ;clk low
bcf    portb,0 ;2do bit adrs 0
bsf    portb,2 ;det 2do bit adrs
bcf    portb,2 ;clk low
bcf    portb,0 ;3er bit adrs

```

```

    bsf     portb,2 ;det 3er bit adrs
    bcf     portb,2 ;clk low
    bsf     portb,0 ;bit read 1
    bsf     portb,2 ;det bit read
    bcf     portb,2 ;clk low
    bsf     portb,0 ;libera bus x ack
    bsf     portb,2 ;det bit ack
ackcmbtr  btfsc   portb,1 ;ack?
          goto   ackcmbtr

    bcf     portb,2 ;clk low si ack read
    bsf     portb,2 ;det dato bit 7
    btfsc   portb,1 ;det edo bit 7 mem 1
    bsf     r,7     ;bit 7 dato en 1
    btfss   portb,1 ;det edo bit 7 mem 0
    bcf     r,7     ;bit 7 dato en 0

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 6
    btfsc   portb,1 ;det edo bit 6 mem 1
    bsf     r,6     ;bit 6 dato en 1
    btfss   portb,1 ;det edo bit 6 mem 0
    bcf     r,6     ;bit 6 dato en 0

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 5
    btfsc   portb,1 ;det edo bit 5 mem 1
    bsf     r,5     ;bit 5 dato en 1
    btfss   portb,1 ;det edo bit 5 mem 0
    bcf     r,5     ;bit 5 dato en

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 4
    btfsc   portb,1 ;det edo bit 4 mem 1
    bsf     r,4     ;bit 4 dato en 1
    btfss   portb,1 ;det edo bit 4 mem 0
    bcf     r,4     ;bit 4 dato en

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 3
    btfsc   portb,1 ;det edo bit 3 mem 1
    bsf     r,3     ;bit 3 dato en 1
    btfss   portb,1 ;det edo bit 3 mem 0
    bcf     r,3     ;bit 3 dato en

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 2
    btfsc   portb,1 ;det edo bit 2 mem 1
    bsf     r,2     ;bit 2 dato en 1
    btfss   portb,1 ;det edo bit 6 mem 0
    bcf     r,2     ;bit 2 dato en

    bcf     portb,2 ;clk low
    bsf     portb,2 ;det dato bit 1
    btfsc   portb,1 ;det edo bit 1 mem 1
    bsf     r,1     ;bit 1 dato en 1

```

```

btfss    portb,1 ;det edo bit 1 mem 0
bcf      r,1     ;bit 1 dato en

bcf      portb,2 ;clk low
bsf      portb,2 ;det dato bit 0
btfsc    portb,1 ;det edo bit 0 mem 1
bsf      r,0     ;bit 0 dato en 1
btfss    portb,1 ;det edo bit 0 mem 0
bcf      r,0     ;bit 0 dato en

bcf      portb,2 ;clk low
bcf      portb,0 ;prep bus stop
bsf      portb,2 ;det stop
bsf      portb,0 ;stop read
movf     r,w     ;prep envio byte leido
movwf    sendreg ;prep envio byte pc
call     ser_out2

incfsz   al      ;incrementa al
goto     read    ;fin comando write
incf     ah
btfss    ah,7    ;mas de 32kbits?
goto     read
clrf     ah      ;si, inic cont alto mem
return

```

```

;-----
end
;-----

```

6.2. Componentes electrónicos



PIC16F8X

18-pin Flash/EEPROM 8-Bit Microcontrollers

Devices Included in this Data Sheet:

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84
- Extended voltage range devices available (PIC16LF8X, PIC16LCR8X)

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single cycle except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
DC - 400 ns instruction cycle

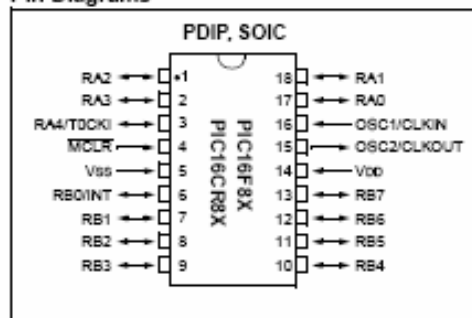
Device	Program Memory (words)	Data RAM (bytes)	Data EEPROM (bytes)	Max. Freq (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

- 14-bit wide instructions
- 8-bit wide data path
- 15 special function hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
 - External RB0/INT pin
 - TMR0 timer overflow
 - PORTB<7:4> interrupt on change
 - Data EEPROM write complete
- 1000 erase/write cycles Flash program memory
- 10,000,000 erase/write cycles EEPROM data memory
- EEPROM Data Retention > 40 years

Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
 - 25 mA sink max. per pin
 - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagrams



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) - via two pins (ROM devices support only Data EEPROM programming)
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Flash/EEPROM Technology:

- Low-power, high-speed technology
- Fully static design
- Wide operating voltage range:
 - Commercial: 2.0V to 6.0V
 - Industrial: 2.0V to 6.0V
- Low power consumption:
 - < 2 mA typical @ 5V, 4 MHz
 - 15 µA typical @ 2V, 32 kHz
 - < 1 µA typical standby current @ 2V

LM193/LM293/LM393/LM2903

Low Power Low Offset Voltage Dual Comparators

General Description

The LM193 series consists of two independent precision voltage comparators with an offset voltage specification as low as 2.0 mV max for two comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM193 series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM193 series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

The LM393 and LM2903 parts are available in National's innovative thin micro SMD package with 8 (12 mil) large bumps.

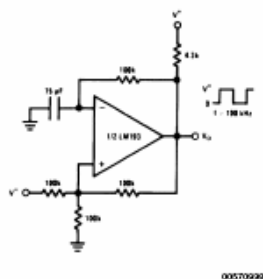
Advantages

- High precision comparators
- Reduced V_{OS} drift over temperature
- Eliminates need for dual supplies
- Allows sensing near ground
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

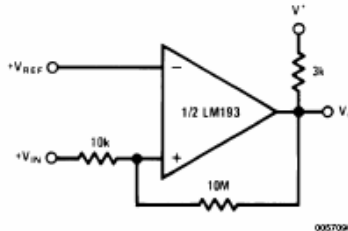
- Wide supply
 - Voltage range: 2.0V to 36V
 - Single or dual supplies: $\pm 1.0V$ to $\pm 18V$
- Very low supply current drain (0.4 mA) — independent of supply voltage
- Low input biasing current: 25 nA
- Low input offset current: ± 5 nA
- Maximum offset voltage: ± 3 mV
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Low output saturation voltage: 250 mV at 4 mA
- Output voltage compatible with TTL, DTL, ECL, MOS and CMOS logic systems
- Available in the 8-Bump (12 mil) micro SMD package
- See AN-1112 for micro SMD considerations

Squarewave Oscillator



00570999

Non-Inverting Comparator with Hysteresis



00570999

ADC0831/ADC0832/ADC0834/ADC0838

8-Bit Serial I/O A/D Converters with Multiplexer Options

General Description

The ADC0831 series are 8-bit successive approximation A/D converters with a serial I/O and configurable input multiplexers with up to 8 channels. The serial I/O is configured to comply with the NSC MICROWIRE™ serial data exchange standard for easy interface to the COPS™ family of processors, and can interface with standard shift registers or μ Ps. The 2-, 4- or 8-channel multiplexers are software configured for single-ended or differential inputs as well as channel assignment.

The differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Features

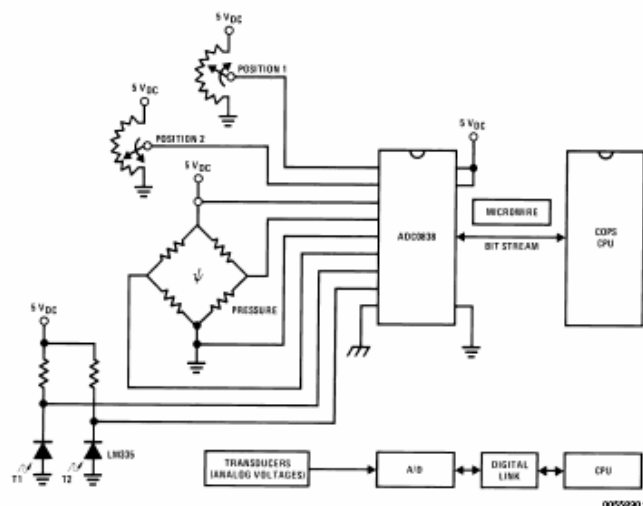
- NSC MICROWIRE compatible—direct interface to COPS family processors
- Easy interface to all microprocessors, or operates "stand-alone"

- Operates ratiometrically or with 5 V_{DC} voltage reference
- No zero or full-scale adjust required
- 2-, 4- or 8-channel multiplexer options with address logic
- Shunt regulator allows operation with high voltage supplies
- 0V to 5V input range with single 5V power supply
- Remote operation with serial digital data link
- TTL/MOS input/output compatible
- 0.3" standard width, 8-, 14- or 20-pin DIP package
- 20 Pin Molded Chip Carrier Package (ADC0838 only)
- Surface-Mount Package

Key Specifications

- | | |
|--------------------------|-------------------------------|
| ■ Resolution | 8 Bits |
| ■ Total Unadjusted Error | $\pm 1/2$ LSB and ± 1 LSB |
| ■ Single Supply | 5 V _{DC} |
| ■ Low Power | 15 mW |
| ■ Conversion Time | 32 μ s |

Typical Application



TRI-STATE® is a registered trademark of National Semiconductor Corporation.
COPS™ and MICROWIRE™ are trademarks of National Semiconductor Corporation.

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
 - Primary (Run): RC oscillator, 76 μ A, 1 MHz, 2V
 - RC_RUN: 7 μ A, 31.25 kHz, 2V
 - SEC_RUN: 9 μ A, 32 kHz, 2V
 - Sleep: 0.1 μ A, 2V
- Timer1 Oscillator: 1.8 μ A, 32 kHz, 2V
- Watchdog Timer: 2.2 μ A, 2V
- Two-Speed Oscillator Start-up

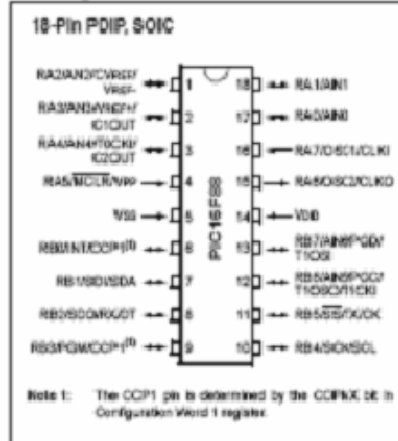
Oscillators:

- Three Crystal modes:
 - LP, XT, HS: up to 20 MHz
- Two External IRC modes
- One External Clock mode:
 - ECIO: up to 20 MHz
- Internal oscillator block:
 - 6 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
 - Capture is 16-bit, max. resolution is 12.5 ns.
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I2C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection:
 - RS-232 operation using Internal oscillator (no external crystal required)
- Dual Analog Comparator module:
 - Programmable on-chip voltage reference
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Pin Diagram



Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
 - Programmable period from 1 ms to 255s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	ICSP (P/M)	USART	Comparators	SSP	Timer/Counter 8/16-bit
	Flash (bytes)	# 8-Bit Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC15F87	7168	4096	358	255	15	N/A	1	Y	2	Y	2/1
PIC15F88	7168	4096	358	255	15	1	1	Y	2	Y	2/1

LM392

Low Power Operational Amplifier/Voltage Comparator

General Description

The LM392 series consists of 2 independent building block circuits. One is a high gain, internally frequency compensated operational amplifier, and the other is a precision voltage comparator. Both the operational amplifier and the voltage comparator have been specifically designed to operate from a single power supply over a wide range of voltages. Both circuits have input stages which will common-mode input down to ground when operating from a single power supply. Operation from split power supplies is also possible and the low power supply current is independent of the magnitude of the supply voltage.

Application areas include transducer amplifier with pulse shaper, DC gain block with level detector, VCO, as well as all conventional operational amplifier or voltage comparator circuits. Both circuits can be operated directly from the standard 5 V_{DC} power supply voltage used in digital systems, and the output of the comparator will interface directly with either TTL or CMOS logic. In addition, the low power drain makes the LM392 extremely useful in the design of portable equipment.

Advantages

- Eliminates need for dual power supplies
- An internally compensated op amp and a precision comparator in the same package
- Allows sensing at or near ground

- Power drain suitable for battery operation
- Pin-out is the same as both the LM358 dual op amp and the LM393 dual comparator

Features

- Wide power supply voltage range
Single supply: 3V to 32V
Dual supply: $\pm 1.5V$ to $\pm 16V$
- Low supply current drain—essentially independent of supply voltage: 600 μA
- Low input biasing current: 50 nA
- Low input offset voltage: 2 mV
- Low input offset current: 5 nA
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage

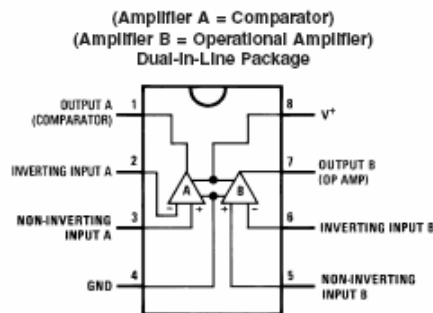
ADDITIONAL OP AMP FEATURES

- Internally frequency compensated for unity gain
- Large DC voltage gain: 100 dB
- Wide bandwidth (unity gain): 1 MHz
- Large output voltage swing: 0V to $V^+ - 1.5V$

ADDITIONAL COMPARATOR FEATURES

- Low output saturation voltage: 250 mV at 4 mA
- Output voltage compatible with all types of logic systems

Connection Diagram



00779001

(Top View)
Order Number LM392M or LM392MX
See NS Package Number M08A
Order Number LM392N
See NS Package Number N08E

8-bit parallel-in/serial-out shift register

74HC/HCT165

FEATURES

- Asynchronous 8-bit parallel load
- Synchronous serial input
- Output capability: standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT165 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT165 are 8-bit parallel-load or serial-in shift registers with complementary serial outputs (Q_7 and $\overline{Q}_7$) available from the last stage. When the parallel load ($\overline{PL}$) input is LOW, parallel data from the D_0 to D_7 inputs are loaded into the register asynchronously.

When $\overline{PL}$ is HIGH, data enters the register serially at the D_8 input and shifts one place to the right ($Q_0 \rightarrow Q_1 \rightarrow Q_2$, etc.) with each positive-going clock transition. This feature allows parallel-to-serial converter expansion by tying the Q_7 output to the D_8 input of the succeeding stage.

The clock input is a gated-OR structure which allows one input to be used as an active LOW clock enable ($\overline{CE}$) input. The pin assignment for the CP and $\overline{CE}$ inputs is arbitrary and can be reversed for layout convenience. The LOW-to-HIGH transition of input $\overline{CE}$ should only take place while CP HIGH for predictable operation. Either the CP or the $\overline{CE}$ should be HIGH before the LOW-to-HIGH transition of $\overline{PL}$ to prevent shifting the data when $\overline{PL}$ is activated.

APPLICATIONS

- Parallel-to-serial data conversion

QUICK REFERENCE DATA

GND = 0 V; $T_{amb} = 25^\circ\text{C}$; $t_r = t_f = 6\text{ ns}$

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{pHL}/t_{pLH}	propagation delay CP to Q_7 , $\overline{Q}_7$ PL to Q_7 , $\overline{Q}_7$ D_7 to Q_7 , $\overline{Q}_7$	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	16 15 11	14 17 11	ns ns ns
f_{max}	maximum clock frequency		56	48	MHz
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per package	notes 1 and 2	35	35	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

$\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is $V_i = \text{GND to } V_{CC}$
For HCT the condition is $V_i = \text{GND to } V_{CC} - 1.5\text{ V}$

ORDERING INFORMATION

See "74HC/HCT/HCU/HCMOS Logic Package Information".

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Features

Superior to Bipolar

- ♦ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ♦ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ♦ Meet All EIA/TIA-232E and V.28 Specifications
- ♦ Multiple Drivers and Receivers
- ♦ 3-State Driver and Receiver Outputs
- ♦ Open-Line Detection (MAX243)

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Three-State	Rx Active In SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

MAXIM

Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

Features

- Low-voltage and Standard-voltage Operation
 - 1.8 ($V_{CC} = 1.8V$ to 5.5V)
- Internally Organized 256 x 8 (2K)
- Two-wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- 1 MHz (5V), 400 kHz (1.8V, 2.5V, 2.7V) Compatibility
- Write Protect Pin for Hardware Data Protection
- 8-byte Page (2K) Write Modes
- Partial Page Writes Allowed
- Self-timed Write Cycle (5 ms max)
- High-reliability
 - Endurance: 1 Million Write Cycles
 - Data Retention: 100 Years
- 8-lead PDIP, 8-lead JEDEC SOIC, 8-lead Mini-MAP (MLP 2x3), 5-lead SOT23, 8-lead TSSOP and 8-ball dBGAA2 Packages
- Lead-free/Halogen-free
- Die Sales: Wafer Form, Waffle Pack and Bumped Wafers

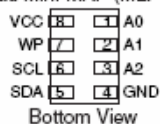
Description

The AT24C02B provides 2048 bits of serial electrically erasable and programmable read-only memory (EEPROM) organized as 256 words of 8 bits each. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. The AT24C02B is available in space-saving 8-lead PDIP, 8-lead JEDEC SOIC, 8-lead Mini-MAP (MLP 2x3), 5-lead SOT23, 8-lead TSSOP, and 8-ball dBGAA2 packages and is accessed via a Two-wire serial interface. In addition, the AT24C02B is available in 1.8V (1.8V to 5.5V) version.

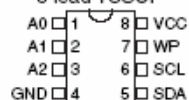
Table 1. Pin Configuration

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
NC	No Connect
GND	Ground
VCC	Power Supply

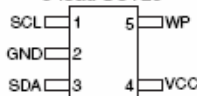
8-lead Mini-MAP (MLP 2x3)



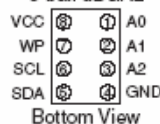
8-lead TSSOP



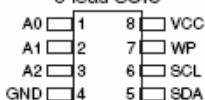
5-lead SOT23



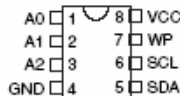
8-ball dBGAA2



8-lead SOIC



8-lead PDIP



Two-wire Serial EEPROM

2K (256 x 8)

AT24C02B

5126B-SEEPR-10/05

LM020L-LM020XMBL

- 16 character x 1 line
- Controller LSI HD44780 is built-in (See page 79).
- +5V single power supply
- Display color: LM020L : Gray
LM020XMBL : New-gray

Ta = 25°C	4.4 V typ.
Ta = 50°C	4.2 V typ.

OPTICAL DATA See page 7

INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function
1	V _{SS}	—	0V
2	V _{DD}	—	+5V
3	V _O	—	—
4	RS	H/L	L: Instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module→MPU) L: Data write (LCD module→MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	

Notes:

In the HD44780, the data can be sent in either 4-bit 2-operation or 8-bit 1-operation so that it can interface to both 4 and 8 bit MPU's.

- (1) When interface data is 4 bits long, data is transferred using only 4 buses of $DB_4 \sim DB_7$, and $DB_0 \sim DB_3$ are not used. Data transfer between the HD44780 and the MPU completes when 4-bit data is transferred twice. Data of the higher order 4 bits (contents of $DB_4 \sim DB_7$ when interface data is 8 bits long) is transferred first and then lower order 4 bits (contents of $DB_0 \sim DB_3$ when interface data is 8 bits long).
- (2) When interface data is 8 bits long, data is transferred using 8 data buses of $DB_0 \sim DB_7$.

MECHANICAL DATA (Nominal dimensions)

Module size	80W x 36H x 12T (max.) mm
Effective display area	64.5W x 13.8H mm
Character size (5 x 7 dots)	3.07W x 5.73H mm
Character pitch	3.77 mm
Dot size	0.55W x 0.75H mm
Weight	about 25 g

ABSOLUTE MAXIMUM RATINGS

ABSOLUTE MAXIMUM RATINGS	min.	max.
Power supply for logic ($V_{DD}-V_{SS}$)	0	7.0 V
Power supply for LCD drive ($V_{DD}-V_O$)	0	13.5 V
Input voltage (V_i)	V_{SS}	V_{DD}
Operating temperature (T_a)	0	50 40°C
Storage temperature (T_{stg})	-20	70 60°C

* Shows the value of type LM20XMBL.

ELECTRICAL CHARACTERISTICS

 $T_a = 25^\circ\text{C}$, $V_{DD} = 5.0\text{ V} \pm 0.25\text{ V}$

Input "high" voltage (V_{IH}) 2.2 V min.

Input "high" voltage (V_{IH})	2.1 V max.
Input "low" voltage (V_{IL})	0.6 V max.

Output high voltage (V_{OH}) ($-I_{OH} = 0.2$ mA) . 2.4 V min.

Output low voltage (V_{OL}) ($I_{OL} = 1.2 \text{ mA}$) . . . 0.4 V max.

Power supply current (I_{PP}) ($V_{DD} = 5.0\text{ V}$) . . . 1.0 mA typ.

2.0 mA max.

POWER SUPPLY FOR LCD DRIVE (Recommended) (VDD-V0)

POWER SUPPLY FOR LCD DRIVE (Recommended) (VDD-V0)

Duty = 1/16

Range of $V_{DD}-V_O$ 1.5~5.25 V

$T_a = 0^\circ\text{C}$ 4.6 V typ.

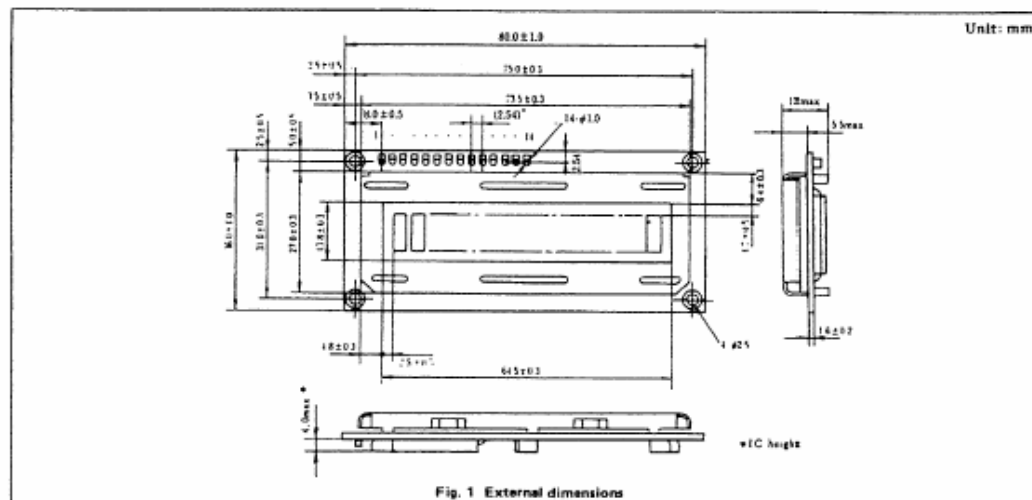


Fig. 1 External dimensions

74VHC4051

8-Channel Analog Multiplexer

74VHC4052

Dual 4-Channel Analog Multiplexer

74VHC4053

Triple 2-Channel Analog Multiplexer

General Description

These multiplexers are digitally controlled analog switches implemented in advanced silicon-gate CMOS technology. These switches have low "on" resistance and low "off" leakage. They are bidirectional switches, thus any analog input may be used as an output and vice-versa. Also these switches contain linearization circuitry which lowers the on resistance and increases switch linearity. These devices allow control of up to $\pm 6V$ (peak) analog signals with digital control signals of 0 to 6V. Three supply pins are provided for V_{CC} , ground, and V_{EE} . This enables the connection of 0–5V logic signals when $V_{CC}=5V$ and an analog input range of $\pm 5V$ when $V_{EE}=5V$. All three devices also have an inhibit control which when high will disable all switches to their off state. All analog inputs and outputs and digital inputs are protected from electrostatic damage by diodes to V_{CC} and ground.

74VHC4051: This device connects together the outputs of 8 switches, thus achieving an 8 channel Multiplexer. The binary code placed on the A, B, and C select lines determines which one of the eight switches is "on", and connects one of the eight inputs to the common output.

74VHC4052: This device connects together the outputs of 4 switches in two sets, thus achieving a pair of 4-channel

multiplexers. The binary code placed on the A, and B select lines determine which switch in each 4 channel section is "on", connecting one of the four inputs in each section to its common output. This enables the implementation of a 4-channel differential multiplexer.

74VHC4053: This device contains 6 switches whose outputs are connected together in pairs, thus implementing a triple 2 channel multiplexer, or the equivalent of 3 single-pole-double throw configurations. Each of the A, B, or C select lines independently controls one pair of switches, selecting one of the two switches to be "on".

Features

- Wide analog input voltage range: $\pm 6V$
- Low "on" resistance: 50 typ. ($V_{CC}-V_{EE}=4.5V$)
30 typ. ($V_{CC}-V_{EE}=9V$)
- Logic level translation to enable 5V logic with $\pm 5V$ analog signals
- Low quiescent current: 80 μA maximum
- Matched Switch characteristic
- Pin and function compatible with the 74HC4051/4052/4053

Commercial	Package Number	Package Description
74VHC4051M	M16A	16-Lead Molded JEDEC SOIC (0.150" Wide)
74VHC4051WM	M16B	16-Lead Molded JEDEC SOIC (0.300" Wide)
74VHC4051N	N16E	16-Lead Molded DIP
74VHC4052M	M16A	16-Lead Molded JEDEC SOIC (0.150" Wide)
74VHC4052WM	M16B	16-Lead Molded JEDEC SOIC (0.300" Wide)
74VHC4052N	N16E	16-Lead Molded DIP
74VHC4053M	M16A	16-Lead Molded JEDEC SOIC (0.150" Wide)
74VHC4053WM	M16B	16-Lead Molded JEDEC SOIC (0.300" Wide)
74VHC4053N	N16E	16-Lead Molded DIP

Note: Surface mount packages are also available on Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

1N4001 THRU 1N4007

PLASTIC SILICON RECTIFIER

VOLTAGE - 50 to 1000 Volts CURRENT - 1.0 Ampere

FEATURES

- Low forward voltage drop
- High current capability
- High reliability
- High surge current capability
- Exceeds environmental standards of MIL-S-19500/228

MECHANICAL DATA

Case: Molded plastic, DO-41

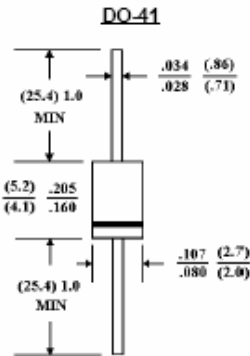
Epoxy: UL 94V-O rate flame retardant

Lead: Axial leads, solderable per MIL-STD-202, method 208 guaranteed

Polarity: Color band denotes cathode end

Mounting Position: Any

Weight: 0.012 ounce, 0.3 gram



Dimensions in inches and (millimeters)

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Ratings at 25 ° ambient temperature unless otherwise specified.

Single phase, half wave, 60 Hz, resistive or inductive load.

For capacitive load, derate current by 20%.

	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNITS
Maximum Recurrent Peak Reverse Voltage	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	35	75	140	280	420	560	700	V
Maximum DC Blocking Voltage	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Current .375"(9.5mm) Lead Length at T _A =75 °	1.0							A
Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load (JEDEC method)	30							A
Maximum Forward Voltage at 1.0A DC and 25 °	1.1							V
Maximum Full Load Reverse Current Full Cycle Average at 75 ° Ambient	30							° A
Maximum Reverse Current at T _A =25 °	5.0							° A
At Rated DC Blocking Voltage T _A =100 °	500							° A
Typical Junction capacitance (Note 1)	15							pF
Typical Thermal Resistance (Note 2) R _{JA}	50							°/W
Typical Thermal resistance (NOTE 2) R _{JA}	25							°/W
JL								
Operating and Storage Temperature Range T _J , T _{STG}	-55 to +150							°

NOTES:

1. Measured at 1 MHz and applied reverse voltage of 4.0 VDC.
2. Thermal Resistance Junction to Ambient and from junction to lead at 0.375"(9.5mm) lead length P.C.B mounted.

Diodes

Switching diode

1N4148 / 1N4150 / 1N4448 / 1N914B

*This product is available only outside of Japan.

●Applications

High-speed switching

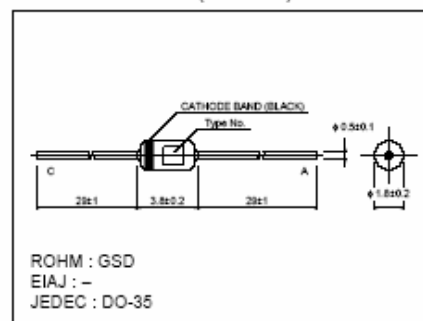
●Features

- 1) Glass sealed envelope. (GSD)
- 2) High speed.
- 3) High reliability.

●Construction

Silicon epitaxial planar

●External dimensions (Units : mm)



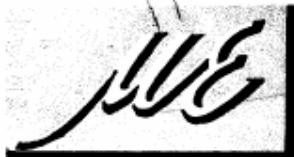
●Absolute maximum ratings (Ta = 25°C)

Type	V _{RM} (V)	V _R (V)	I _{RM} (mA)	I _O (mA)	I _F (mA)	I _{FSM} 1μs (A)	P (mW)	T _J (°C)	T _{opr} (°C)	T _{stg} (°C)
1N4148	100	75	450	150	200	2	500	200	-65~+200	-65~+200
1N4150	50	50	600	200	250	4	500	200	-65~+200	-65~+200
1N4448 (1N914B)	100	75	450	150	200	2	500	200	-65~+200	-65~+200

●Electrical characteristics (Ta = 25°C)

Type	V _F (V)												BV (V) Min.		I _S (μA) Max.				C _J (pF)	t _r (ns)	
	0.1mA	0.25mA	1mA	2mA	5mA	10mA	20mA	30mA	50mA	100mA	200mA	250mA	@25°C		@150°C						
													V _R (V)	I _S (μA)	V _R (V)	I _S (μA)					
1N4148						1.0				0.75	0.82	0.87		75	100	0.025 5.0	20 75	50.0	20	4	4
1N4150			0.54			0.66				0.75	0.82	0.87		-	50	0.1	50	100.0	50	2.5	4
			0.62			0.74				0.86	0.92	1.0									
1N4448 (1N914B)					0.82									-	100	0.025 5.0	20 75	50.0	20	4	4
					0.72						1.0										

The upper figure is the minimum V_F and the lower figure is the maximum V_F value.



BC 546 through BC 550

NPN SILICON AF SMALL SIGNAL TRANSISTORS

THE BC546 THROUGH BC550 ARE NPN SILICON PLANAR EPITAXIAL TRANSISTORS FOR USE IN AF SMALL SIGNAL AMPLIFIER STAGES AND DIRECT COUPLED CIRCUITS. THEY ARE COMPLEMENTARY TO BC556 THROUGH BC560.

THE BC549, BC550 ARE CHARACTERIZED BY LOW NOISE FIGURE.

CASE TO-92F



CEB

ABSOLUTE MAXIMUM RATINGS

		BC546	BC547	BC548	BC549	BC550
Collector-Base Voltage	V _{CB0}	80V	50V	30V	30V	50V
Collector-Emitter Voltage (V _{BE} =0)	V _{CES}	80V	50V	30V	30V	50V
Collector-Emitter Voltage (I _B =0)	V _{CEO}	65V	45V	30V	30V	45V
Emitter-Base Voltage	V _{EB0}	6V	6V	5V	5V	5V
Collector Current	I _C	100mA				
Collector Peak Current	I _{CM}	200mA				
Total Power Dissipation (T _A ≤ 25°C)	P _{tot}	500mW				
		derate 4mW/°C above 25°C				
Operating Junction & Storage Temperature T _j , T _{stg}		-55 to 150°C				

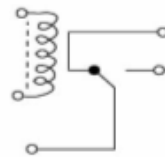
ELECTRICAL CHARACTERISTICS (T_A=25°C unless otherwise noted)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	TEST CONDITIONS
Collector-Base Breakdown Voltage	BV _{CB0}					I _C =10μA I _B =0
BC546		80			V	
BC547		50			V	
BC548		30			V	
BC549		30			V	
BC550		50			V	
Collector-Emitter Breakdown Voltage	BV _{CES}					I _C =10μA V _{BE} =0
BC546		80			V	
BC547		50			V	
BC548		30			V	
BC549		30			V	
BC550		50			V	
Collector-Emitter Breakdown Voltage	LV _{CEO}					I _C =2mA(Pulsed) I _B =0
BC546		65			V	
BC547		45			V	
BC548		30			V	
BC549		30			V	
BC550		45			V	

MICRO ELECTRONICS LTD.

38 HUNG TO ROAD, KWUN TONG, HONG KONG. TELEX 43510
KWUN TONG P. O. BOX 69477 CABLE ADDRESS "MICROTRON"
TELEPHONE: 3-430181-6 3-693309, 3-692423
FAX: 3-410321

THD---RELEVADOR-MINIATURA-1P/2T-BOBINA-DE-5-VCD-y-12-VCD¶



.....CARACTERÍSTICAS.....

ORDER CODE

MD-5-W-M-D

D: DIODE

□: Plastic Cover
M: Metallic Cover

□: PIN PITCH 5.08mm
W: PIN PITCH 10.16mm

COIL VOLTAGE — 5: DC 5V
12: DC 12V

TYPE

- Relay SIP-REED.¶
- Disponible en plástico y metal.¶
- Totalmente lavable.¶
- Aumento de fuerza a 1500-VDC (entre bobina y contacto)¶
- Provee alta velocidad, miniatura y costo efectivo de solución de switcheo.¶

OKI semiconductor**MSC2328A-XX YS2/KS2****T-46-23-17****262,144 BY 8 BIT DYNAMIC RAM MODULE**

OKI SEMICONDUCTOR GROUP

GENERAL DESCRIPTION

The OkI MSC2328A-XX YS2/KS2 is a fully decoded, 262,144 words $\times$ 8 bit CMOS dynamic random access memory composed of two 1Mb DRAMs in SOJ (MSM514256AJS). The mounting of two SOJs together with two 0.2 μ F decoupling capacitors on a 30 pin glass epoxy Single-In-Line Package provides any application where high density and large capacity of storage memory are required. The electrical characteristics of the MSC2328A-XX YS2/KS2 are quite same as the original MSM514256A JS; each timing requirements are noncritical, and power supply tolerance is very wide.

4**FEATURES**

- 262,144 word $\times$ 8 bit Organization
- Single $\pm$ 5V Supply (10% Tolerance)
- 30-Pin Socket Insertable Module
- Refresh Period ... 8ms (512 cycles)
- All Inputs, Outputs, Clocks Fully TTL compatible
- 3-States Outputs
- Common $\overline{\text{CAS}}$ Control for two Common Data-In and Data-Out Lines
- Row Access Time:
 - 80ns max. (MSC2328A-8A/80 YS2/KS2)
 - 100ns max. (MSC2328A-1A/10 YS2/KS2)
- Low Power Dissipation:
 - 825mW max. (MSC2328A-8A/80 YS2/KS2)
 - 715mW max. (MSC2328A-1A/10 YS2/KS2)
- Operating Temperature ... 0°C to 70°C

FUNCTIONAL BLOCK DIAGRAM