



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

REPORTE TÉCNICO DEL DISEÑO DE UNA MESA INTERACTIVA DE TRABAJO CON APLICACIÓN A LA EDUCACIÓN

**Antonio Garcés Madrigal
José Luis Pérez Silva**

25 de mayo de 2013

Informe Técnico

RESUMEN

En este informe técnico se presenta el diseño de una mesa interactiva de trabajo que está fundamentalmente pensada para la educación en ciencias y técnica. Las herramientas utilizadas para este fin son la informática, los multimedia y la computación, entre otros.

La mesa es una herramienta que permite al profesor y al alumno practicar los conocimientos obtenidos en clases, sin la necesidad de conocer de manera específica un entorno multimedia basado en la computación, facilitando y motivando con ello, la participación e intercambio de ideas de todos los actores del proceso educativo.

El modelo como lo hemos mencionado hace que el actor central del proceso educativo sea el estudiante, colocando a la computadora personal en una función transparente al alumno.

Para el profesor, el modelo le facilita una serie de herramientas que le permiten observar el comportamiento del alumno en el aula, el grado de interacción con otros alumnos e interactuar en tiempo real y de manera ubicua con las experiencias que un alumno o grupo de alumnos están desarrollando en el aula.

La mesa presenta al alumno una serie de objetos que se adaptan a cualquier escenario, esto es, pueden ser células, componentes mecánicos, o electrónicos, entre otros. Una vez seleccionados y configurados son simulados por la PC, la cual muestra la evolución o el resultado de la experiencia definida por el alumno por medio del cañón de vídeo.

El profesor puede proponer o realizar una experiencia, cambiar los parámetros de evolución o hipótesis, e intercambiar sus puntos de vista con otros compañeros de clase.

El informe técnico contiene la información correspondiente al diseño electrónico de la mesa y sus elementos para su posible reproducción. Para lograr esto dividiremos el informe en las siguientes partes: Diseño de la mesa, la mesa alámbrica, la superficie interactiva, los objetos de la mesa, la tabletas de circuito impreso, el prototipo del diseño, la interfaz con el usuario, y las conclusiones del diseño.

Índice

	Introducción.	4
1.	Diseño de una mesa interactiva de trabajo.	4
2.	Mesa Alámbrica.	6
2.1.	Superficie Interactiva.	6
2.2.	Objetos de la mesa.	8
2.3.	Placas de circuito impreso.	11
2.4.	Prototipo.	12
2.5.	Interfaz de usuario.	14
	Conclusiones.	14
3.	Apéndice A Código de los programas.	16
4	Apéndice B hojas de especificaciones.	28

Introducción

En la actualidad en el campo de la educación se buscan nuevos modelos que ayuden a mejorar los métodos de enseñanza utilizados en las aulas. Las herramientas utilizadas para este fin son la informática, los multimedia y la computación, entre otros. La nueva metodología que se desea obtener debe brindar una mejora en el grado y velocidad de asimilación de los conocimientos y habilidades impartidas; la sociedad debe asumir de mejor manera la necesidad de educar a más individuos en más etapas y por periodos más prolongados, desde la educación básica hasta la actualización y la capacitación continua. Los métodos de producción y la demanda de mejoras en la productividad en todos los niveles de interacción y producción de la sociedad, hacen que la demanda de los servicios de educación sea, en ocasiones, mayores a la oferta de los mismos. Esta deficiencia afecta desde la falta de profesores capacitados, hasta la infraestructura requerida para asistir a los métodos y planes educativos.

1. Diseño de una mesa interactiva de trabajo.

Dentro de este escenario, decidimos presentar ante un grupo de trabajo interdisciplinario en el campo de la educación, una herramienta que no este restringida a un solo campo del conocimiento, que permita interactuar a profesores y alumnos de una forma más familiar al modo de pensar y de actuar del humanos, esto es, una herramienta que permita al profesor y al alumno practicar los conocimientos obtenidos en clases, sin la necesidad de conocer de manera específica un entorno multimedia basado en la computación, facilitando y motivando con ello la participación e intercambio de ideas de todos los actores del proceso educativo.

El modelo como lo hemos mencionado hace que el actor central del proceso educativo sea el estudiante, colocando a la computadora personal en una función transparente al alumno; para el profesor, dependiendo de su habilidad en el uso de las herramientas de computo, se le puede transparentar el uso de la PC, sin embargo el modelo le facilita una serie de herramientas que le permiten observar el comportamiento del alumno en el aula, el grado de interacción con otros alumnos e interactuar en tiempo real y de manera ubicua con las experiencias que un alumno o grupo de alumnos están desarrollando en el aula.

El modelo general en el que se basa la herramienta desarrollada se muestra en la figura 1.

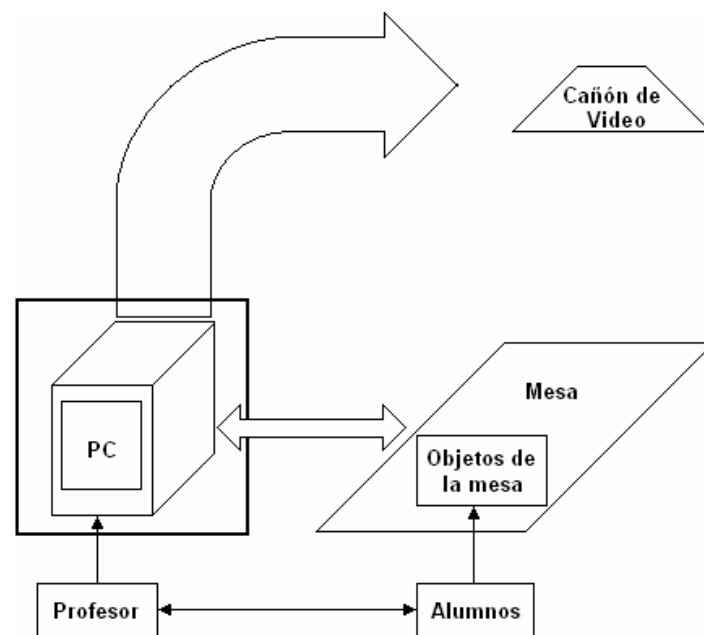


Figura 1. Modelo General de la Mesa.

El método de operación es el siguiente: el alumno y el profesor participan dentro del aula en una sesión de clases, una vez presentados los objetivos de la clase, el profesor propone a los alumnos una serie de experiencias a desarrollar, con el fin de que el alumno se involucre de manera personal en el aprendizaje o práctica de los conocimientos. La mesa, de acuerdo al campo del conocimiento, presenta al alumno una serie de objetos que se adaptan a cualquier escenario, esto es, pueden ser células, componentes mecánicos, electrónicos, entre otros. Una vez seleccionados y configurados son simulados por la PC, la cual muestra la evolución o el resultado de la experiencia definida por el alumno por medio del cañón de vídeo. El profesor previamente escoge desde la PC el escenario y el alumno sin tener que utilizar la PC, empieza a intervenir en el escenario propuesto de acuerdo los conocimientos presentados en la clase, de tal forma que puede proponer o realizar una experiencia, cambiar los parámetros de evolución o hipótesis, e intercambiar sus puntos de vista con otros compañeros de clase. En este proceso el profesor puede ver como los alumnos obtienen las respuestas del sistema, que parámetros o ajustes aplican, se puede comunicar con los alumnos de cualquier mesa, puede saber si el alumno interactúa con uno o varios grupos de alumnos y el sistema le puede presentar una serie de reportes de la evolución de la experiencia así como el comportamiento del alumno en el período. En la figura 2 se muestran los módulos generales que componen el sistema descrito.

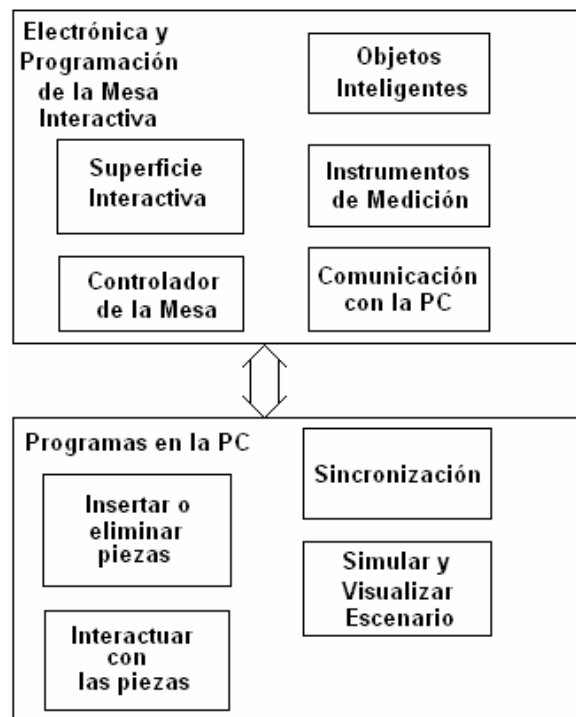


Figura 2. Módulos generales que componen el sistema

Todo esto se logra a través de los objetos y la mesa que permiten al alumno hacer ajustes sobre de ellos, estos objetos también se comunican entre sí y la mesa, ésta con la PC y la computadora presenta al profesor todos los valores, ajustes y acciones que se dan sobre las mesas y los objetos que están el aula. Los alumnos al presentarse a la clase toman identificadores electrónicos que ubican sus posición durante toda la experiencia; con ello el profesor puede inferir algunos hábitos de los alumnos, ubicar más fácilmente los grupos con un desempeño o comportamiento determinado y en su momento puede a través de estos elementos, interactuar en tiempo real con los alumnos y sus experiencias en el aula.

Algunas de las formas por medio de las cuales se pueden comunicar o detectar a los objetos en la mesa o a los alumnos dentro del aula, son de manera general los que se muestran en la figura 3:

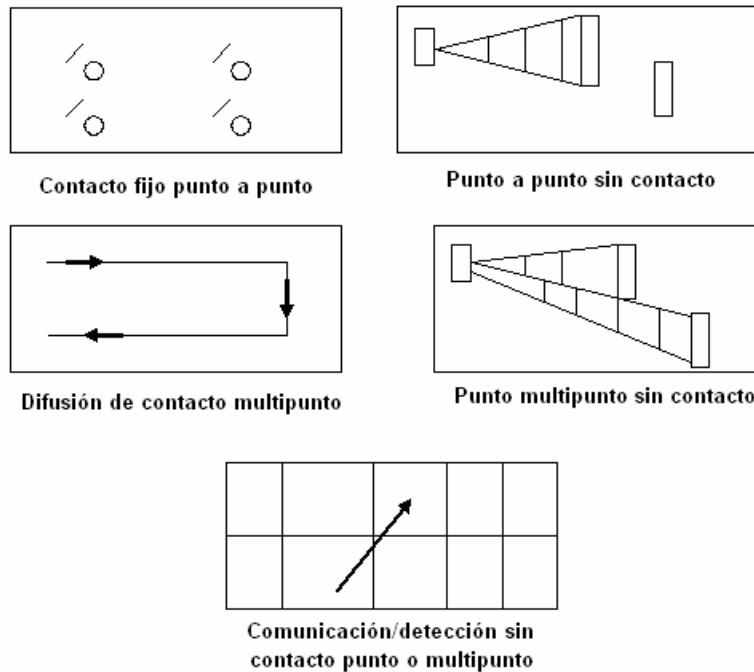


Figura 3. Métodos de detección de objetos sobre la superficie de trabajo

Dentro de estas posibles alternativas hemos desarrollado la siguiente implementación.

2. Mesa alámbrica (contacto fijo punto a punto)

Esta implementación permite que en una mesa se coloquen de forma fija un número N de objetos. Alrededor de la mesa se colocan de cuatro a ocho alumnos, que de manera independiente pueden tomar objetos y colocarlos en la mesa, ajustar sus parámetros, establecer su esquema o arquitectura de interacción, ponerse de acuerdo con otros compañeros de clase y con el profesor y una vez establecido su escenario dejar que la computadora les presente la evolución de la experiencia propuesta, pudiendo en cualquier momento reiniciar, repetir o cambiar cualquier parte del proceso.

2.1. Superficie interactiva (Mesa)

Este elemento es conceptualmente asociado a la división física de la mesa en donde se colocan una serie de conectores que permiten ubicar e interactuar eléctricamente con cada uno de los objetos de la mesa y por medio de los cuales se puede saber su ubicación exacta, mientras están conectados operando en la mesa. Los conectores se han diseñado de forma tal que es posible ubicar los objetos tanto en posición horizontal o vertical protegiendo la integridad mecánica y eléctrica del módulo insertado. Para registrar la interacción de los usuarios con los diferentes objetos de la mesa, un microprocesador localiza la ubicación de cada uno de los objetos de la mesa, así como la configuración asignada a cada uno de ellos, las interconexiones entre los objetos y su función en la mesa, definidas cada una por el usuario para cada uno de los diferentes objetos conectados a la mesa. La transferencia de toda esta información se da entre los objetos, la mesa y la computadora por medio de protocolos de comunicación establecidos a propósito entre la mesa y los elementos de la mesa y la PC.

La estructura operativa de la mesa se muestra en la figura 4.

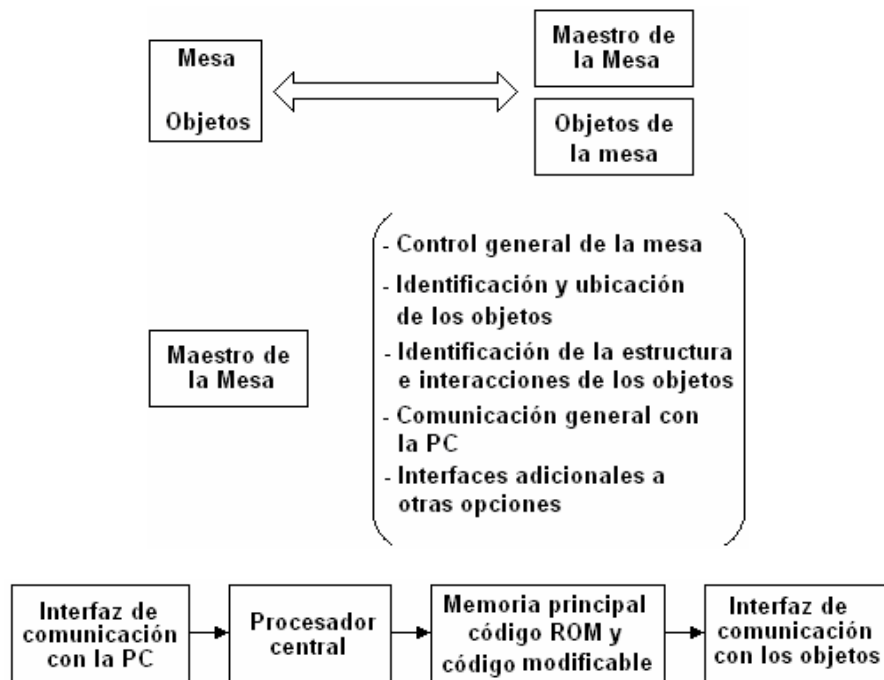


Figura 4. Estructura operativa de la mesa

En la figura 5 se muestra el diagrama de flujo de la tableta de control.

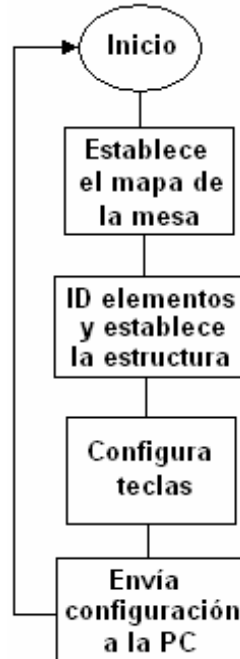


Figura 5. Diagrama de flujo de la tableta de control.

La estructura de la tableta electrónica de control maestra esta dada por el diagrama de la figura 6.

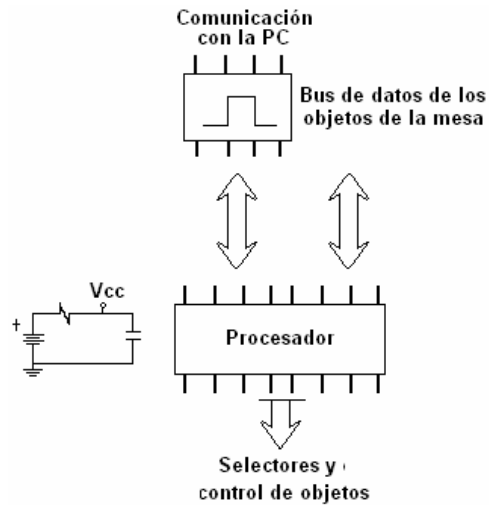


Figura 6. Estructura de la tableta electrónica de control

2.2. Objetos de la mesa

Cada uno de estas tabletas electrónicas permite a los usuarios la configuración de variables y parámetros de identificación. Por medio de estos elementos el alumno puede determinar de qué elemento se trata, que valores o configuración toma el entorno, y los va transmitiendo al maestro de la mesa de tal forma que la computadora va generando el modelo de simulación del escenario deseado. El esquema operativo de los objetos se muestra en la figura 7.



Figura 7. Esquema operativo de los objetos

La configuración y el diagrama de bloques correspondiente a ésta etapa se muestra en la figura 8.



Figura 8. Diagrama de bloques del esquema operativo de los objetos

En la figura 9 se muestra el diagrama de flujo para la tableta de un objeto genérico para la mesa interactiva.

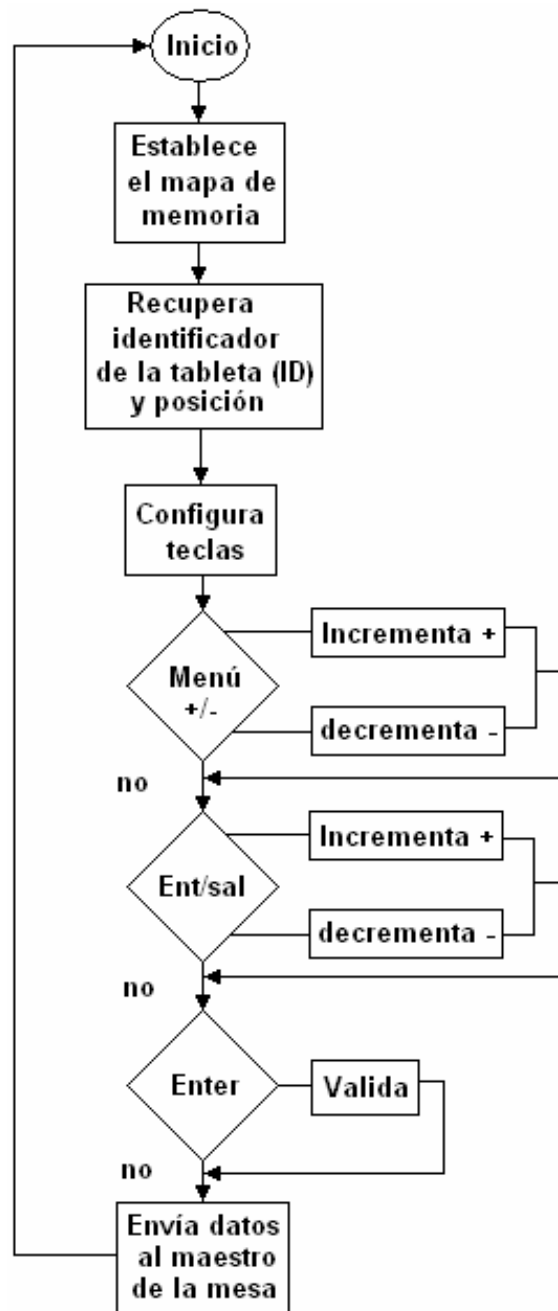


Figura 9. Tableta objeto genérico para la mesa interactiva.

La estructura de las tabletas electrónicas de los objetos, está dada por el diagrama esquemático de la tableta electrónica genérica de la figura 10.

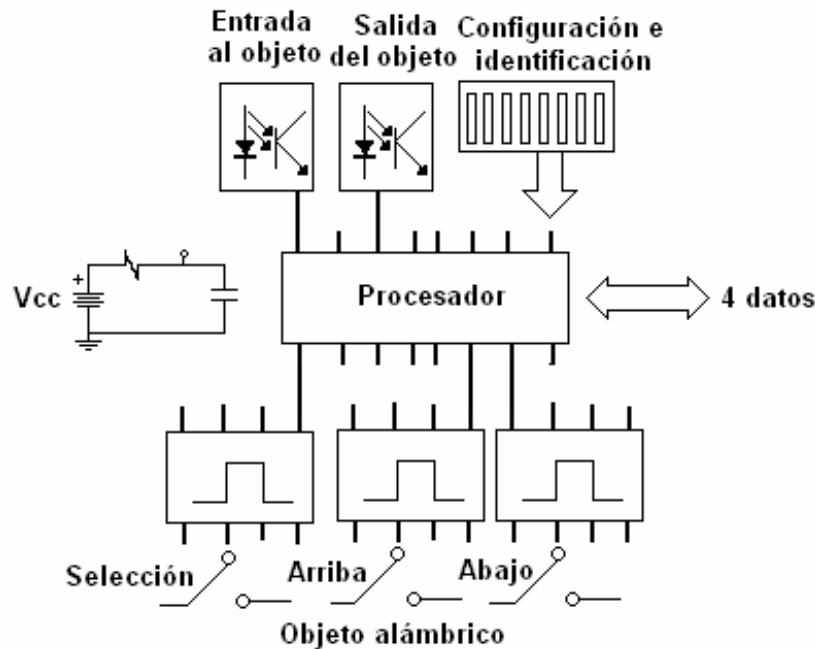


Figura 10. Diagrama esquemático de la tableta electrónica genérica.

Cada objeto se ubica sobre la superficie interactiva de tal forma que el maestro de la mesa va obteniendo los parámetros que se le han suministrado para la simulación del escenario propuesto. Conforme el maestro de la mesa escudriña cada objeto actualiza el mapa de la mesa y transmite la información para que la PC pueda presentar el avance de la implementación o la simulación de la experiencia propuesta. La tabla 1 muestra la ubicación de hasta 16 objetos sobre la mesa de prueba construida, sin embargo ya se tienen diseñadas las opciones para 64 o más objetos sobre una sola mesa. En la actualidad consideramos que este número de objetos, 16, es suficiente para las características físicas de la mesa propuesta.

A3	A2	A1	A0	B11	B12	B13	B14	B15	B16	B17	B18	B21	B22	B23	B24	B25	B26	B27	B28
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Tabla 1. Secuencias para la ubicación de objetos sobre la mesa interactiva

En la figura 11 se muestra el diagrama esquemático del secuenciador de los objetos sobre la mesa.

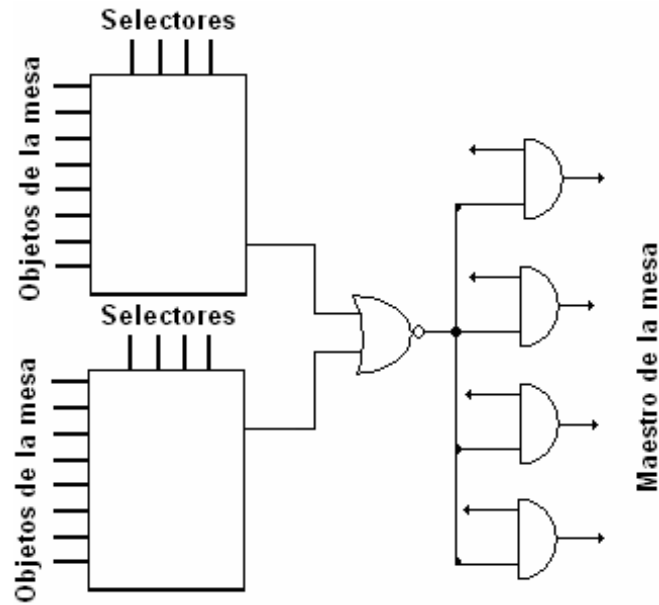


Figura 11. Secuenciador de objetos en la mesa

A la vez que se selecciona el elemento del multiplexor de objetos, el objeto se prepara para insertar sus datos en un bus de datos de difusión, activando para ello la interfaz correspondiente, evitando de esta forma la presencia de colisiones en el canal de comunicación de la mesa. Este circuito está descrito por el diagrama esquemático de la figura 12.

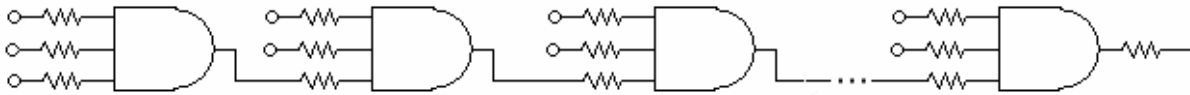


Figura 12. Circuito para la inserción de datos de n dispositivos.

2.3. Placas de circuito impreso

Se muestran las placas de circuito impreso construidas para la construcción de prototipos de pruebas de la mesa interactiva de trabajo alámbrica. En la figura 13 se muestra la tableta impresa para la etapa de control y comunicaciones.

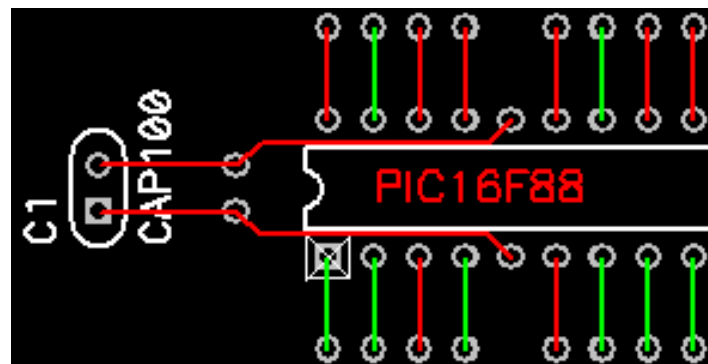


Figura 13. Placa de circuito impreso para la etapa de control y comunicaciones

En la figura 14 se muestra la tarjeta impresa para el circuito del objeto alámbrico.

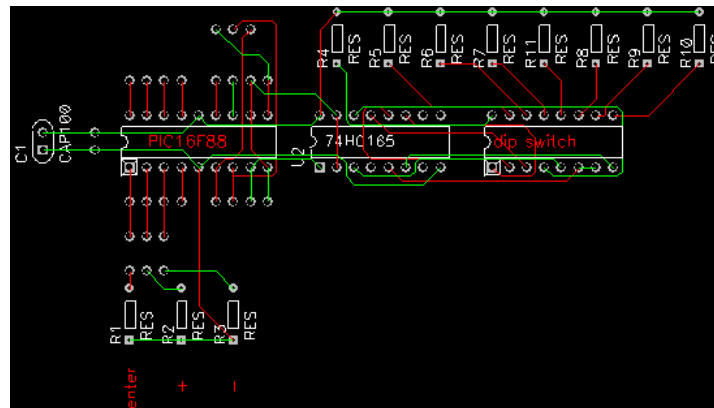


Figura 14. Placa de circuito impreso objeto alámbrico

En la figura 15 se muestra el circuito impreso para la etapa de multiplexado de 16 objetos

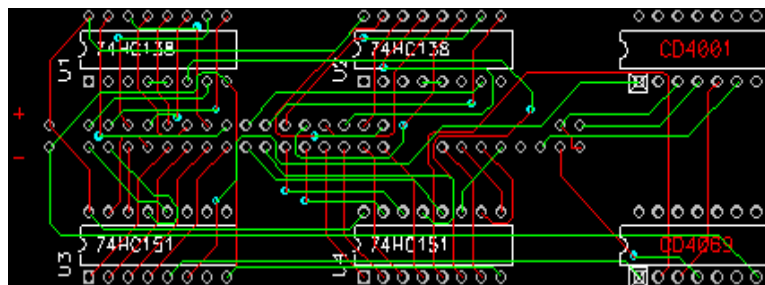


Figura 15. Placa de circuito impreso para etapa de multiplexado de 16 objetos.

2.4. Prototipo

El objeto prototipo construido para la realización de pruebas, se muestra ya operando en la mesa interactiva de trabajo en la fotografía de la figura 16.

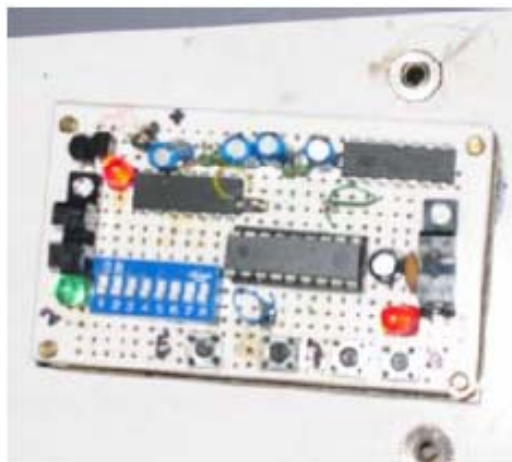


Figura 16. Objeto genérico

Uno de los prototipos de mesa construidos para la realización de pruebas con los usuarios, ya en operación se muestra en la fotografía de la figura 17, en la cual se pueden observar un par de grupos de conectores en los cuales el objeto puede ser insertado en forma vertical u horizontal, dependiendo de las necesidades de construcción del usuario. Los objetos claramente muestran la estructura mostrada en el diagrama del objeto alámbrico.

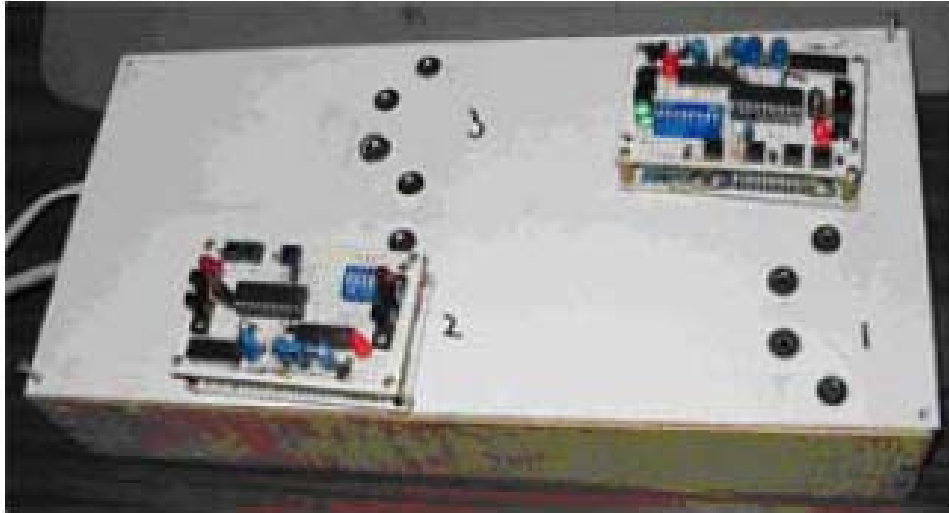


Figura 17. Mesa interactiva de trabajo con dos objetos operando

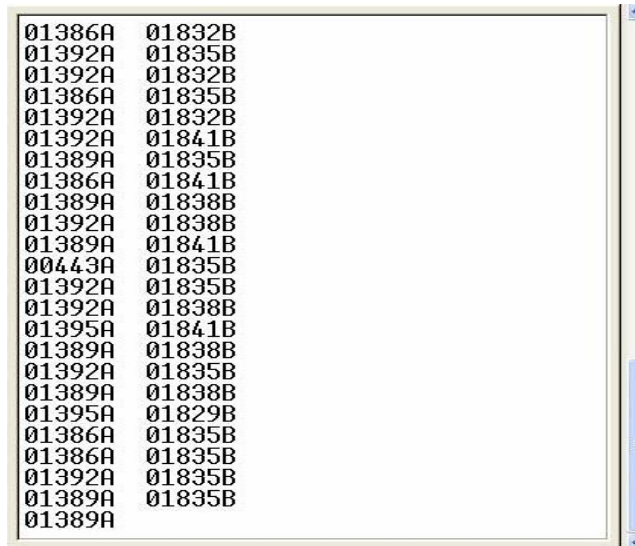
La fotografía de la figura 18, nos muestra la mesa prototipo conteniendo al maestro de la mesa, la fuente de alimentación, la electrónica de comunicación y selección, así como los objetos sobre la superficie de la mesa interactiva..



Figura 18. Mesa interactiva de trabajo en operación

2.5. Interfaz del usuario

La información proporcionada por los objetos al maestro de la mesa es transmitida a la PC con el fin de que los datos sean alimentados a los programas de simulación y visualización correspondientes, una pantalla de monitoreo y depuración de los datos de comunicación se presenta en la figura 19 tomada directamente de la pantalla. En ella se observa la transmisión de información de dos objetos que están conectados a la mesa interactiva.



01386A	01832B
01392A	01835B
01392A	01832B
01386A	01835B
01392A	01832B
01392A	01841B
01389A	01835B
01386A	01841B
01389A	01838B
01392A	01838B
01389A	01841B
00443A	01835B
01392A	01835B
01392A	01838B
01395A	01841B
01389A	01838B
01392A	01835B
01389A	01838B
01395A	01829B
01386A	01835B
01386A	01835B
01392A	01835B
01389A	01835B
01389A	

Figura 19. Pantalla de monitoreo de datos de la interfaz de comunicación del maestro de la mesa interactiva.

Conclusiones del diseño y construcción de la mesa alámbrica.

De las pruebas realizadas podemos concluir lo siguiente:

El equipo cumple con las características propuestas en el objetivo, proporcionando además los elementos para continuar trabajando con alternativas adicionales a la de los métodos de contacto.

La construcción del prototipo proporciona herramientas a un costo muy reducido, las cuales se pueden adaptar completamente a las necesidades planteadas por el grupo de trabajo interdisciplinario, teniendo el control total de la tecnología necesaria, presentando un potencial para la realización de prototipos de mayor capacidad, considerando adicionalmente que no existe un producto comercial que se adapte a la solución del problema.

Una desventaja vista para el actual modelo, es con respecto a la rigidez de la ubicación de los objetos, sin embargo la construcción de esta versión permite ir evaluando escenarios donde sí se requiere contar con una mesa de trabajo sin contacto y a su vez potenciar métodos de solución alternativa al problema.

Apéndices

A1. Código de máquina del maestro de la mesa

Los programas asociados a cada etapa se listan a continuación

```
;------mesasclv.asm-----  
;  
;-----  
;mapa de memoria  
indf    equ    0x00    ;system  
tmr0    equ    0x01    ;system  
status  equ    0x03    ;idem  
fsr     equ    0x04    ;idem  
porta   equ    0x05    ;idem  
portb   equ    0x06    ;idem  
intcon  equ    0x0b    ;idem  
optreg  equ    0x81    ;idem  
;progrm regs  
sendreg  equ    0x0c    ;ser out  
count    equ    0x0d    ; ser out, ser in  
cambios  equ    0x0e  
bytes    equ    0x0f    ;apuntador de bytes parcial aux serializain  
                        ;(serializain serializao idrcvr=byte1)  
byte1    equ    0x10    ;dispos scent (00en ctdr) este masnx3 idem  
                        ;en=enter, ctdr=scroll  
byte2    equ    0x11    ;idem inout (00ou 00in) este masnx3 idem  
byte3    equ    0x12    ;idem local n (Bidf ffff) este masnx3 idem  
                        ;numero d byte B solo en alto ultimo byte,func  
                        ;es dip switch(000f ffff),id es de serializain  
  
byte4    equ    0x13    ;dispos n-1  
byte5    equ    0x14    ;idem  
byte6    equ    0x15    ;idem  
byte7    equ    0x16    ;dispos n-2  
byte8    equ    0x17    ;idem  
byte9    equ    0x18    ;idem  
byte10   equ    0x19    ;dispos n-3  
byte11   equ    0x1a    ;idem  
byte12   equ    0x1b    ;idem  
byte13   equ    0x1c    ;dispos n-1  
byte14   equ    0x1d    ;idem  
byte15   equ    0x1e    ;idem  
byte16   equ    0x1f    ;dispos n-2  
byte17   equ    0x20    ;idem  
byte18   equ    0x21    ;idem  
byte19   equ    0x22    ;dispos n-3  
byte20   equ    0x23    ;idem  
byte21   equ    0x24    ;idem  
byte22   equ    0x25    ;dispos n-1  
byte23   equ    0x26    ;idem  
byte24   equ    0x27    ;idem  
byte25   equ    0x28    ;dispos n-2  
byte26   equ    0x29    ;idem  
byte27   equ    0x2a    ;idem  
byte28   equ    0x2b    ;dispos n-3  
byte29   equ    0x2c    ;idem  
byte30   equ    0x2d    ;idem
```

```

byte31 equ    0x2e    ;dispos n-1
byte32 equ    0x2f    ;idem
byte33 equ    0x30    ;idem
byte34 equ    0x31    ;dispos n-2
byte35 equ    0x32    ;idem
byte36 equ    0x33    ;idem
byte37 equ    0x34    ;dispos n-3
byte38 equ    0x35    ;idem
byte39 equ    0x36    ;idem
byte40 equ    0x37    ;dispos n-1
byte41 equ    0x38    ;idem
;byte42 equ    0x39    ;idem
;byte43 equ    0x3a    ;dispos n-2
;byte44 equ    0x3b    ;idem
;byte45 equ    0x3c    ;dispos n-1
;byte46 equ    0x3d    ;idem
;byte47 equ    0x3e    ;idem
;byte48 equ    0x3f    ;dispos n-2
temp_1 equ    0x39    ;
temp_2 equ    0x3a    ; offset apuntador de bytes totales recibidos
temp   equ    0x3b
rcvreg equ    0x3c
temp_c equ    0x3d    ;
temp_d equ    0x3e    ;
temp_e equ    0x3f    ;
;-----
;mapa de bits
c      equ    0
rp0    equ    5
z      equ    2
;-----
      org     0x00
inicio bcf     intcon,7;desahb. ints. global
      movlw   0xf0    ;config puertob, eeee ssss
      tris    portb
      movlw   0x0f    ;no transmitir a nign dispositivo esclavo
      movwf   portb   ;todos los bits de control en alto
      movlw   0x1d    ;config. puertoa, xxxe eese
      tris    porta
      movlw   0x3d    ;empieza proceso limpieza regs
      movwf   temp_e  ;ultimo reg
      movlw   0x0c
      movwf   temp_d  ;primer reg
limpia movf    temp_e,w ;apunta reg
      movwf   fsr     ;indirecto
      clrf    indf    ;limpialo
      movf    temp_d,w ;define limite inicial
      subwf   temp_e,w ;vs limite superior actual
      decf    temp_e,f ;el que sigue
      btfss   status,z;todos?
      goto    limpia  ;no
repite clrf    temp_d ;byte de bit renglon (0000 vvvv)
;
;                                0 no importa el valor en temp_1,
;                                1 importa valor del idx asociado en temp_1
;                                para multx3 y saber dispositivos x reng
;                                para multx3 y saber dispositivos x reng
      clrf    temp_e ;contador de rengloner 00-11 (0-3)

```

```

        clrf    temp_1;contiene los idxs en va en conj con temp_d (idid idid)
        clrf    temp_2;contador de byte totales, proporc el offset de almcento.
        clrf    cambios      ;estado de espera del transmisor 0 no tx, 1 si tx
        bcf     portb,temp_e  ;prepara recepcion de renglon por señalizacion b6i (b0)
        call    espera ;da tiempo muerto de margen
continua  clrf     temp_c ;limpia apuntador comodin
        clrf    temp      ;limpia apuntado aux serin
        movlw   0x02      ;valor requerido para que serializain opere los lazos
        movwf   bytes     ;por medio de la var aux de rutina bytes
        movlw   0x04      ;apuntador de bi, numero de renglones-
        subwf   temp_e,w   ;en proceso-
        btfsc   status,z;terminaron?
        goto    finserin ;si
siguiente goto    serializain ;no,busca obtener datos de vecinos
verifica goto    primero;primer byte del renglon? X1
finserin movlw   0x0f      ;prepara no transmitir-
        movwf   portb     ;todos los renglones esclavos
        btfsc   cambios,1  ;datos x enviar?
        call    serializao ;envía datos
        goto    repite     ;repite recoleccion de bytes
;-----
primero movlw   0x00      ; X1,prueba condicionales,byte-
        subwf   rcvreg,w   ;recuperado
        btfsc   status,z;nulo?
        goto    cero      ;renglon en cero
        btfsc   rcvreg,1   ;primer byte del renglon?
        goto    uno
        goto    siguiente ;no
;-----
cero      bcf     temp_d,temp_e;cerro en el bit correspondiente del renglon
;
;                                     es decir sin cambios, info no presente (no se tx)
;                                     su idx no importa #bytes en cero
        bsf     portb,temp_e ;no recibir + trans de ese renglon
        incf    temp_e,f    ;siguiente renglon
        bcf     portb,temp_e ;permite trans siguiente renglon
        decf    temp_2,f    ;no se almacena el bye nulo se regresa el contador
        goto    continua
;-----
uno       bsf     temp_d,temp_e;uno en el bit correspondiente del renglon
;
;                                     es decir con cambios, info presente (si tx)
;                                     su idx va de 00 a 11 implica #bytes de 3 a 12
idrcvr   movlw   0x60      ; extrae # ident. dispos.
        andwf   rcvreg,w   ;
        movwf   temp_c ; reg. aux. comodin
        swapf   temp_c,f   ;acomoda identif. en nyble bajo
        rrf     temp_c,f   ;recupera adecuadamente el nyble de id
        movlw   0x83      ;limpia
        andwf   temp_c,w   ;el carry. El valor recup. puede ser 00-11
        addwf   temp_1,f   ;primer idx al byte contenedor
        rlf     temp_1,f   ;prepara insercion del siguiente idx
        rlf     temp_1,f   ;son dos bits
        movlw   0xfc      ;limpia los bits de los acarreo (1111 1100)
        andwf   temp_1,f   ;en el byte contenedor
        clrf    bytes     ;por el momento contador auxiliar de esta parte
        incf    temp_c,f   ;ajusta el procesode multiplicacion por el cero de 00-11
multip   movlw   0x03      ;numero de bytes del dispositivo (base nultiplicacion)

```

```

    addwf    bytes,f ;calcula los bytes del renglon-multiplicando
    decfsz   temp_c,f ;por multiplicacion, esta es var multiplicador
    goto     multip ;siguel el proceso de multip.
    decf     bytes,f ;el contador de bytes ayudara a introducir n(var bytes) -1
                                ; bytes adicionales pues el primero ya entro

    bsf      cambios,1 ;datos por enviar
    goto     siguiente
;-----
serializain    bcf      intcon,5;desh. ints tmr0
               bcf      intcon,7;deshb ints. global
               clrf     tmr0 ;limpia tmr0
               clrwdt   ;prepara asig. preescaler
               bsf      status,rp0 ;cambio a banco 1
               movlw    0xd4 ;config tmr-ctr
               movwf    optreg
               bcf      status,rp0 ;regresa bco zero
               movlw    0x08 ;shift bit ctr
               movwf    count ;carga valor contador especific
sbit           btfscc   porta,0 ;busca bit de inicio
               goto     sbit ;marca
               movlw    0x80 ;bit de inicio, medio periodo
               movwf    tmr0 ;carga valor tmr-ctr
               bcf      intcon,2;limpia bandera tmr0
time1          btfscc   intcon,2;espera periodo, bandera desb.
               goto     time1 ;aun no
               btfscc   porta,0 ;bit aun en bajo?
               goto     falso ;falso inicio, sal de rutina
               clrf     tmr0 ;si bit inicio, limpia contador0
               bcf      intcon,2;limpia desbordamiento en int tmr0
time2          btfscc   intcon,2;desbord. dabdera nueva cta?
               goto     time2 ;no
               bcf      intcon,2;si, limpia bandera tmr-ctr
               movf     porta,w ;lee puertoa
               movwf    temp ;prepara rotacion de bit a carry
               rrf      temp,f ;rota bit a carry
               rlf      rcvreg,f;primer bit a reg de recup de dato via carry
               decfsz   count,f ;corrieron los 8 bits?
               goto     time2 ;no
               bsf      cambios,1 ;datos por transmitir
time3          btfscc   intcon,2;fin periodo de espera?
               goto     time3 ;no
;             bsf      intcon,7;hab ints globales
               clrf     temp ;ahora registro auxiliar en multiplos
               movlw    0x00 ;el byte recup-
               subwf    rcvreg,w ;viene en ceros?
               btfscc   status,z;?
               goto     falso ;si
               movf     temp_2,w ;prepara formacion de dir. de almacenamiento
               addlw    0x10 ;con el offset del apuntador base, var bytes a idrcvr
               movwf    fsr ;con indirect addressing
               movf     rcvreg,w ;el byte recuperado
               movwf    indf ;en secuencia
               incf     temp_2,f ;incrementa apuntador de offset (total)
               decfsz   bytes,f ;contador de bytes del renglon (parcial) en zero?
               goto     verifica ;no, extrae sig byte.
;             movlw    0x2a ;offset maximo- debe ser 30 (se cambio el limite por limite fis de regs pic)

```

```

;      subwf  temp_2,w      ;alcanzado-
;      btfss  status,z;?
;      bsf    portb,temp_e  ;si
;      incf   temp_e,f      ;apunta sig. renglon
;      bcf    portb,temp_e  ;activa bit de tx siguiente renglon (b6i)
;      goto   continua
;-----
falso  clrf   rcvreg
;      goto   verifica
;-----
serializao
;      movf   temp_d,w      ;prepara envio byte-bit renglon totales
;      movwf  sendreg       ;transfierelo a registro de tx
;      call   ser_out;envialo
;      movf   temp_1,w      ;prepara envio de contador de bytesxrenglon totales
;      movwf  sendreg       ;transfierelo a registro de tx
;      call   ser_out;envialo
todos  movlw  0x10          ;apuntador base de la tabla de datos de los dispos.
;      addwf  temp_2,w      ;prepara tx de los bytes de los dispos.
;      movwf  fsr           ;carga apuntador
;      movf   indf,w        ;extrae su contenido
;      movwf  sendreg       ;envialo para transmision
;      call   ser_out;sacalos
;      decfsz temp_2,f      ;ultimo byte?
;      goto   todos        ;no
;      clrf   cambios      ;si, nada por transmitir
;      return  ;fin de transmision
;-----
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
;      movlw  0x30         ;defina el contador
;      movwf  tmr0         ;inici el C-T
;      bcf    intcon,2;limpia la bancera 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
;      movwf  tmr0         ;inici el C-T
;      bcf    intcon,2;si, limpia la bandera de desbordamiento del C-T
nxtbit rrf     sendreg,f    ;rota el LMS 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
time21 btfss  intcon,2;desbordamiento del C-T?
;      goto   time21      ;no
;      movlw  0x30         ;defina el contador
;      movwf  tmr0         ;inici el C-T
;      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"
time31  btfss   intcon,2;desbordamiento del C-T?
        goto   time31  ;no
        bsf    intcon,5;hab int tmr0
        bsf    intcon,7;hab ints. global
        return  ;si
;-----
espera  movlw   0x20    ;prepara tiempo muerto en comodin
        movwf  temp
        decfsz temp,f  ;transcurrido
        goto   espera
        return  ;finde espera
;-----
;
;
        End

```

A.2. Código de máquina para la operación de los objetos esclavos de la mesa interactiva de trabajo.

```

indf    equ     0x00    ;system
tmr0    equ     0x01    ;system
status  equ     0x03    ;idem
fsr     equ     0x04
porta   equ     0x05    ;idem
portb   equ     0x06    ;idem
intcon  equ     0x0b    ;idem
optreg  equ     0x81    ;idem
;progrmregs
sendreg  equ     0x0c    ;ser out
count    equ     0x0d    ; ser out, ser in
bytes    equ     0x0e    ;serializain serializao idrcvr=byte1
ctr      equ     0x0f    ;idem
byte1    equ     0x10    ;dispos scent (00en ctr) este masnx3 idem
                        ;en=enter, ctr=scroll
byte2    equ     0x11    ;idem  inout (00ou 00in) este masnx3 idem
byte3    equ     0x12    ;idem  local n (Bidf ffff) este masnx3 idem
                        ;numero d byte B solo en alto ultimo byte,func
                        ;es dip switch(000f ffff),id es de serializain

byte4    equ     0x13    ;dispos n-1
byte5    equ     0x14    ;idem
byte6    equ     0x15    ;idem
byte7    equ     0x16    ;dispos n-2
byte8    equ     0x17    ;idem
byte9    equ     0x18    ;idem
byte10   equ     0x19    ;dispos n-3
byte11   equ     0x1a    ;idem
byte12   equ     0x1b    ;idem
scroll   equ     0x1c
enter    equ     0x1d
inout    equ     0x1e
func     equ     0x1f
idctr    equ     0x20    ;formacion byte3
temp_wequ  equ     0x21    ;acumulador en ints

```

```

temp_s equ    0x22    ;status en ints
temp_1 equ    0x23    ;en in
temp_2 equ    0x24    ;en out
temp_3 equ    0x25    ;estado en in
temp_4 equ    0x26    ;estado en out
scent equ     0x27    ;var de serializa0
cambios equ    0x28    ;var de estado gral
in equ        0x29    ;idem
out equ       0x30    ;idem
temp equ      0x31
ctrra equ     0x32
ctrrc equ     0x33
nyble equ     0x34
nyblea equ    0x35
rcvreg equ    0x36
masterid equ   0x37
temp_c equ    0x38    ;registro comodin
temp_d equ    0x39    ;registro comodin limpieza limte inf
temp_e equ    0x3a    ;registro comodin limpza limite sup
org          0x00
goto inicio
goto int_ser

inicio bcf     intcon,7;desahab. ints. global
        movlw 0xbf    ;config puertob, esee eeee
        tris  portb
        movlw 0x1d    ;config. puertoa, xxxe eese
        tris  porta
        movlw 0x38    ;empieza proceso limpieza regs
        movwf temp_e ;ultimo reg
        movlw 0x0c    ;primer reg
        movwf temp_d
limpia  movf   temp_e,w ;apunta reg
        movwf fsr      ;indirecto
        clrf  indf     ;limpialo
        decf  temp_e,f  ;el que sigue
        movf  temp_d,w  ;define limite inicial
        subwf temp_e,w  ;vs limite superior actual
        btfss status,z;todos?
        goto  limpia   ;no
        movf  portb,w  ;lee puertob
        andlw 0x1f     ;separa dip switch
;        movwf temp_c ;reg. COMODIN
;        rrf   temp_c ;prepara identificacion dip switch
;        movf  temp_c,w
        movwf func     ;guarda identificador de dip switch
;generacion de la variable local byte3
        addlw 0x80     ;genera el B=1 de primer byte
        movwf byte3    ;igual pero para transmision incluye creacion de B=1
                        ;y limpieza de idctr (0x1f)x(Bidf ffff)+0x80= (100f ffff)
;fin de la generacion de byte3
        bsf   portb,6  ;no recibir datos
        bsf   intcon,7;habilita ints. global
barrido movf   porta,w ;lee in-outs
        call  in1      ;entradas?
        call  out1     ;salidas?

```

```

        bcf     portb,6 ;prepara recibir datos vecinos
        movlw  0x40    ;TIEMPO DE ESPERA
        call   espera  ;ventana minima por rutinas en ejec. peor caso
;
;
continua    call serializain ;busca obtener datos de vecinos
        btfsc  rcvreg,7      ;primer byte?
        call   idrcvr ;si
        movlw  0x7f
        btfsc  masterid,4    ;1er byte absoluto del arreglo?
        andwf  rcvreg,f,no, limpia bit 7 de bytes subsecuentes
        movlw  0x03          ;offset local
        addwf  bytes,w;prepara limite inferior de mapa de memoria
        movwf  temp_c;a reg comodin
        movlw  0x03          ;offset local
        addwf  ctrra,f; carga offset total limite superior map. mem.
        movf   ctrra,w ;prepara direeccionamiento de bytes-dispostivos
;
        movwf  ctrrc ;auxiliar en serializao, aqui no
;
        addlw  0x03          ;offset local
        addwf  bytes,w;crea direccion de almac. actual
        movwf  fsr           ;ubicala, indirect addrsng OJO
        movf   rcvreg,w      ;prepara variable OJO
        movwf  indf          ;guarda byte en arreglo OJO
        decf   ctrra,f ; siguiente posicion de memoria ultimo byte?
        movf   temp_c,w      ;hasta el limite establecido
        subwf  ctrra,w
        btfss  status,z;si?
        goto   continua      ;no
        clrf   masterid      ;limpia identificador primer byte
                                ;absoluto recibido
;
        bsf    cmbios,1      ;datos x transmitir
        bsf    portb,6 ;no recibir datos
        btfsc  cmbios,1      ;datos x enviar?
        call   serializao    ;envía datos
        goto   barrido ;repite in-out
;-----
espera movwf  temp_c ;prepara tiempo muerto en comodin
        decfsz temp_c,f      ;transcurrido
        goto   espera
        return ;finde espera
;-----
in1    btfsc  porta,2 ;entrada alta
        call   con          ;uno
        btfss  porta,2 ;entrada baja
        call   sin          ;cero
        return
;-----
con    bsf    temp_1,1      ;uno al temporal_1
        movf   temp_3,w      ;recupera estado anterior
        subwf  temp_1,w      ;comparalos
        btfsc  status,z
        return ;iguales
        bsf    cmbios,1      ;diferentes
        movf   temp_1,w      ;respalda estado actual
        movwf  temp_3
        movlw  0x01 ; pon cable almacena dato de cambio a PC

```

```

        movwf in      ;guarda en reg. correspondiente
        call  forin
        return
;-----
sin      bcf      temp_1,1      ;cero al temporal_1
        movf     temp_3,w      ;recupera estado anterior
        subwf    temp_1,w      ;comparalos
        btfsc    status,z
        return   ;iguales
        bsf      cambios,1     ;diferente
        movf     temp_1,w      ;respalda estado actual
        movwf    temp_3
        movlw    0x02          ;quita cable almacena dato de cambio a PC
        movwf    in           ;guarda registro correspondiente
        call     forin
        return
;-----
forin    movf     inout,w      ;recupera variable inout
        andlw    0xf0          ;prepara nyble bajo inout (L)
        movwf    inout        ;guardala con nyble bajo limpio
        movf     in,w         ;estado actual
        addwf    inout,w      ;guardalo en nyble bajo
        movwf    byte2        ;generacion de var local byte2
        movwf    inout
        clrf     in           ;limpieza preventiva
        return   ;fin
;-----
out1     btfsc    porta,3      ;entrada alta
        call     cono         ;uno
        btfss    porta,3      ;entrada baja
        call     sino         ;cero
        return
;-----
cono     bsf      temp_2,1     ;uno al temporal_2
        movf     temp_4,w      ;recupera estado anterior
        subwf    temp_2,w      ;comparalos
        btfsc    status,z;
        return   ;iguales
        bsf      cambios,1     ;diferentes
        movf     temp_2,w      ;respalda estado actual
        movwf    temp_4;en registro temporal_4
        movlw    0x10          ;pon
        movwf    out          ;guarda en reg. correspondiente
        call     forout
        return
;-----
sino     bcf      temp_2,1     ;cero al temporal_2
        movf     temp_4,w      ;recupera estado anterior
        subwf    temp_2,w      ;comparalos
        btfsc    status,z;
        return   ;iguales
        bsf      cambios,1     ;diferentes
        movf     temp_2,w      ;respalda estado actual
        movwf    temp_4;en registro temporal_4
        movlw    0x20          ;quita
        movwf    out          ;guarda en reg. correspondiente

```

```

        call    forout
        return
;-----
forout  movf    inout,w ;recupera variable inout
        andlw   0x0f    ;prepara nyble alto (M)
        movwf   inout   ;guardala con nyble alto limpio (M)
        movf    out,w    ;estado actual
        addwf   inout,w  ;guardalo en nyble alto
        movwf   inout
        movwf   byte2    ;generacion de var local byte2
        clrf    out      ;limpieza preventiva
        return ;fin
;-----
int_ser bcf     intcon,7;deshab. ints. global
        movwf   temp_w;salva acumulador
        swapf   status,w ;prepara respaldo status
        movwf   temp_s;respalda status intercambiado (swapd)
        btfsc   intcon,1;intx scroll (b0)?
        call    scrl    ;si
        btfsc   intcon,0;int x enter (b7)?
        call    enter1
        swapf   temp_s,w ;prepara recup status
        movwf   status ;recup status
        swapf   temp_w,f ;prepara recup. acumulador
        swapf   temp_w,w ;recup acumulador
        bsf     intcon,7;rehabilita ints. global
        retfie
;-----
scrl    incf     scroll,f ;incrementa opcion del menu
        movlw   0x0f    ;hasta 15 opciones
        subwf   scroll,w;
        btfsc   status,z;iguales?
        clrf    scroll ;iguales, limpia scroll a zero
        bsf     cambios,1 ;diferentes dato x enviar
        movf    enter,w ;prepara byte de estado
        addwf   scroll,w;scroll+enter
        movwf   scent   ;a byte maestro
        movwf   byte1    ;generacion de var local byte1
        bcf     intcon,1;limpia bandera scroll (b0)
        return
;-----
enter1  bsf     cambios,1 ;enter, datos x enviar
        movlw   0x10    ;prepara codigo de accion de tecla
        movwf   enter   ;cierre de menu (00-10)
        addwf   scroll,w;scroll+enter
        movwf   scent   ;guarda el valor en byte maestro
        movwf   byte1    ;generacion de var local byte1
        clrf    scroll ;limpia scroll
        clrf    enter   ;limpia enter
        bcf     intcon,0;limpia bandera int. x enter (b7)
        return
;-----
idrcvr  movlw   0x60    ;extrae ident. dispos.
        andwf   rcvreg,w ;
        movwf   temp_c ; reg. aux. comodin
        swapf   temp_c,f ;acomoda identif. en nyble bajo

```

```

        rrf      temp_c,f      ;prepara adecuadamente nyble
        movf     temp_c,w      ;en acumulador
        movwf    nyblea        ;en var. aux.
        movwf    nyble         ;y en var. maestra
        movlw    0x03         ;bytes del zer0
nxtdsp   addlw    0x03         ;bytes sig. dispos.
        decfsz   nyblea,f;cuantos?
        goto     nxtdsp        ;incrementalos en 03
        movwf    ctrra         ;a var. aux
        movwf    ctrr          ;y a var. maestra
        incf     nyble,f       ;da nuevo num de dispostvs al arreglo
        rlf      nyble,f       ;prepara inserción de nyble en byte identificador
        movlw    0x1f          ;prepara nyble alto para el ctr e ident.de arreg
        andwf    rcvreg,f;listo
        swapf    nyble,w        ;prep.valor del nyble alto (00000idX)(0idX0000)
        andlw    0x60          ;listo (0110 0000)(0id0 0000)
        addlw    0x80          ;identif. 1er. byte (1000 0000), (1idf ffff)
        addwf    rcvreg,f;guarda valor de id de dispos y func en el
                                   ;registro 1er byte xtransmted
        bsf      masterid,4     ; identfic global de primer byte
        return
;-----
serializain   bcf      intcon,5;desh. ints tmr0
              bcf      intcon,7;deshb ints. global
              clrf     tmr0      ;limpia tmr0
              clrwdt    ;prepara asig. preescaler
              bsf      status,rp0 ;cambio a banco 1
              movlw    0xd4      ;config tmr-ctr
              movwf    optreg
              bcf      status,rp0 ;regresa bco zero
              movlw    0x08      ;shift bit ctr
              movwf    count      ;carga valor contador especific
sbit          btfscc   porta,0 ;busca bit de inicio
              goto     sbit       ;marca
              movlw    0x80      ;bit de inicio, medio periodo
              movwf    tmr0      ;carga valor tmr-ctr
time1         bcf      intcon,2;limpia bandera tmr0
              btfscc   intcon,2;espera periodo, bandera desb.
              goto     time1      ;aun no
              btfscc   porta,0 ;bit aun en bajo?
              return    ;falso inicio, sal de rutina
              clrf     tmr0      ;si bit inicio, limpia contador0
              bcf      intcon,2;limpia desbordamiento en int tmr0
time2         btfscc   intcon,2;desbord. dabdera nueva cta?
              goto     time2      ;no
              bcf      intcon,2;si, limpia bandera tmr-ctr
              movf     porta,w ;lee puerto a
              movwf    temp      ;prepara rotacion de bit a carry
              rrf      temp,f ;rota bit a carry
              rlf      rcvreg,f;primer bit a reg de recup de dato via carry
              decfsz   count,f ;corrieron los 8 bits?
              goto     time2      ;no
              bsf      cambios,1 ;datos por transmitir
time3         btfscc   intcon,2;fin periodo de espera?
              goto     time3      ;no
              bsf      intcon,7;hab ints globales

```

```

        return ;byte recibido
;-----
serializao    movlw 0x03
              addwf ctrr,w ;numero de byte a transmtr
              addwf bytes,w ;forma direccion de extraccion
              movwf ctrrc ;introducela a contador de transmision
              btfss porta,4 ;puedo transmitir?
              call cuentaB ;si, empieza a transmitir
              return ;no se puede enviar
cuentaB      movf ctrrc,w ;prepara apuntador de bytes
              movwf fsr ;carga apuntador
              movf indf,w ;extrae su contenido
              movwf sendreg ;envialo para transmision
              call ser_out ;sacalo
              decfsz ctrrc,f ;estima cuantos bytes van
              movf bytes,w ;carga limite inferior
              subwf ctrrc,w ;en acumulador
              btfss status,z ;ultimo byte?
              goto cuentaB ;no
              clrf cambios ;si, nada por transmitir
              clrf masterid ;limpia identificador primer byte
                          ;absoluto recibido, limp preventiva
              return ;fin de transmision
;-----
ser_out      bcf intcon,5;deshb tmr0 ints
              bcf intcon,7;deshb global ints.
              clrf tmr0 ;limpia contador
              clrwdt ;prep asig preescaler
              bsf status,rp0 ;a banco 1
              movlw 0xd4 ;config. timer-ctr
              movwf optreg ;carga config
              bcf status,rp0 ;cambia a banco0
              movlw 0x08 ;inicia contador de bits
              movwf count ;carga contador de bits
              bcf porta,1 ;bit de inicio
              clrf tmr0 ;limpia timer0
              bcf intcon,2;limpia bandera de desbord.
time11      btfss intcon,2;termino periodo?
              goto time11 ;no
              bcf intcon,2;si, limpia bandera de desb. tmr0
nxtbit      rlf sendreg,f ;rota msb en carry
              bcf porta,1 ;limpia bit1
              btfsc status,c;prueba la bandera de carry
              bsf porta,1 ;bit en alto
time21      btfss intcon,2;paso periodo?(desbord)
              goto time21 ;no
              bcf intcon,2;si, limpia bandera
              decfsz count,f ;ya corrieron los 8 bits?
              goto nxtbit ;no
              bsf porta,1 ;si, marca de fin de xmision
time31      btfss intcon,2;paso periodo?
              goto time31 ;no
              bsf intcon,7;hab ints. global
              return ;si
;-----
end

```

B. Hojas de especificaciones

Hojas de especificaciones de los circuitos integrados utilizados para la construcción física de la mesa interactiva de trabajo

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:

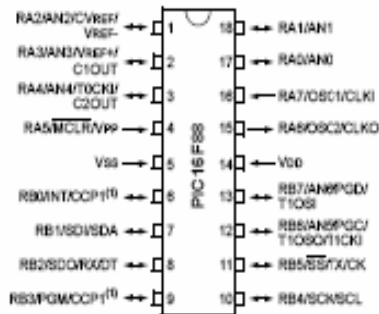
- Three Crystal modes:
 - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
 - ECIO: up to 20 MHz
- Internal oscillator block:
 - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

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 (AUSART/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

18-Pin PDIP, SOIC



Note 1: The CCP1 pin is determined by the CCPX SE in Configuration Word 1 register.

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 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	1	1	Y	2	Y	2/1

CD4069UBC Inverter Circuits

General Description

The CD4069UB consists of six inverter circuits and is manufactured using complementary MOS (CMOS) to achieve wide power supply operating range, low power consumption, high noise immunity, and symmetric controlled rise and fall times.

This device is intended for all general purpose inverter applications where the special characteristics of the MM74C901, MM74C907, and CD4049A Hex Inverter/Buffer are not required. In those applications requiring larger noise immunity the MM74C14 or MM74C914 Hex Schmitt Trigger is suggested.

All inputs are protected from damage due to static discharge by diode clamps to V_{DD} and V_{SS} .

Features

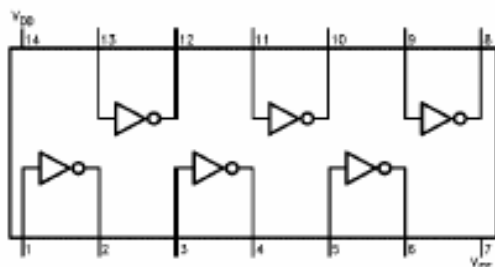
- Wide supply voltage range: 3.0V to 15V
- High noise immunity: $0.45 V_{DD}$ typ.
- Low power TTL compatibility: Fan out of 2 driving 74L or 1 driving 74LS
- Equivalent to MM74C04

Ordering Code:

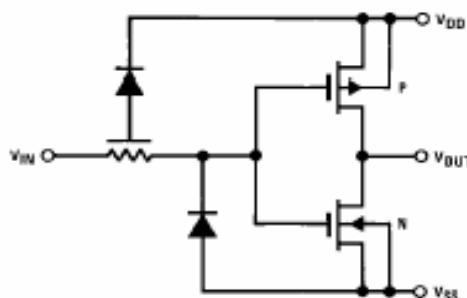
Order Number	Package Number	Package Description
CD4069UBCM	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150" Narrow
CD4069UBCSJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
CD4069UBCN	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Device also available in Tape and Reel. Specify by appending suffix "X" to the ordering code.

Connection Diagram



Schematic Diagram



8-input multiplexer

74HC/HCT151

FEATURES

- True and complement outputs
- Multifunction capability
- Permits multiplexing from n lines to 1 line
- Non-inverting data path
- See the "251" for the 3-state version
- Output capability: standard
- I_{OC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT151 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.

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 I_n to $Y, \bar{Y}$	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	17	19	ns
	S_n to $Y, \bar{Y}$		19	20	ns
	$\bar{E}$ to Y		12	13	ns
	$\bar{E}$ to $\bar{Y}$		14	18	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per package	notes 1 and 2	40	40	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".

8-input multiplexer

74HC/HCT151

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
4, 3, 2, 1, 15, 14, 13, 12	I_0 to I_7	multiplexer inputs
5	Y	multiplexer output
6	$\bar{Y}$	complementary multiplexer output
7	$\bar{E}$	enable input (active LOW)
8	GND	ground (0 V)
11, 10, 9	S_0, S_1, S_2	select inputs
16	V_{CC}	positive supply voltage

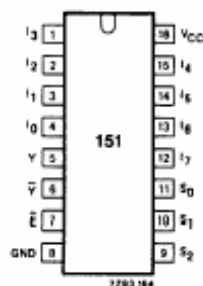


Fig.1 Pin configuration.

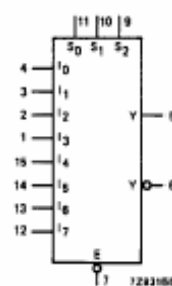


Fig.2 Logic symbol.

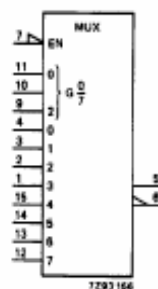


Fig.3 IEC logic symbol.

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

FEATURES

- Demultiplexing capability
- Multiple input enable for easy expansion
- Ideal for memory chip select decoding
- Active LOW mutually exclusive outputs
- Output capability: standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT138 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LS TTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT138 decoders accept three binary weighted address inputs (A_0, A_1, A_2) and when enabled, provide 8 mutually exclusive active LOW outputs ($\overline{Y}_0$ to $\overline{Y}_7$).

The "138" features three enable inputs: two active LOW ($\overline{E}_1$ and $\overline{E}_2$) and one active HIGH (E_3). Every output will be HIGH unless $\overline{E}_1$ and $\overline{E}_2$ are LOW and E_3 is HIGH.

This multiple enable function allows easy parallel expansion of the "138" to a 1-of-32 (5 lines to 32 lines) decoder with just four "138" ICs and one inverter.

The "138" can be used as an eight output demultiplexer by using one of the active LOW enable inputs as the data input and the remaining enable inputs as strobes. Unused enable inputs must be permanently tied to their appropriate active HIGH or LOW state.

The "138" is identical to the "238" but has inverting outputs.

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	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	12	17	ns
t_{PHL}/t_{PLH}	A_n to $\overline{Y}_n$				
t_{PHL}/t_{PLH}	E_3 to $\overline{Y}_n$				
	E_n to $\overline{Y}_n$		14	19	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per package	notes 1 and 2	67	67	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".

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 2, 3	A_0 to A_2	address inputs
4, 5	$\overline{E}_1, \overline{E}_2$	enable inputs (active LOW)
6	E_3	enable input (active HIGH)
8	GND	ground (0 V)
15, 14, 13, 12, 11, 10, 9, 7	$\overline{Y}_0$ to $\overline{Y}_7$	outputs (active LOW)
16	V_{CC}	positive supply voltage

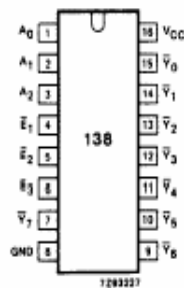


Fig.1 Pin configuration.

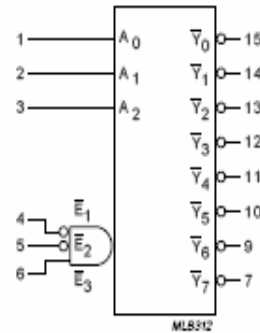
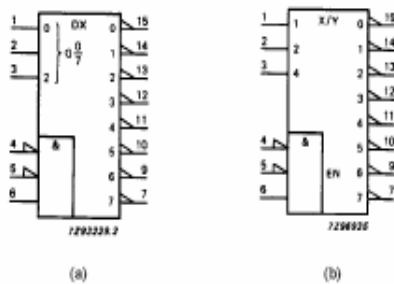


Fig.2 Logic symbol.



(a) (b)

Fig.3 IEC logic symbol.

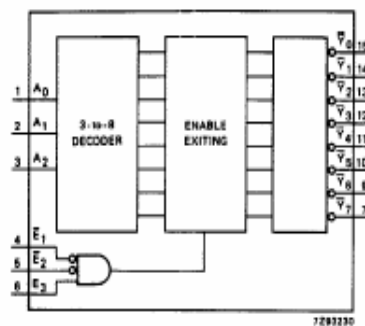


Fig.4 Functional diagram.

Quad 2-input AND gate

74HC08; 74HCT08

FEATURES

- Complies with JEDEC standard no. 8-1A
- ESD protection:
HBM EIA/JESD22-A114-A exceeds 2000 V
MM EIA/JESD22-A115-A exceeds 200 V.
- Specified from -40 to $+85$ °C and -40 to $+125$ °C.

DESCRIPTION

The 74HC/HCT08 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/HCT08 provide the 2-input AND function.

QUICK REFERENCE DATA

GND = 0 V; $T_{amb} = 25$ °C; $t_r = t_f = 6$ ns.

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			74HC08	74HCT08	
t_{PLH}/t_{PLH}	propagation delay nA, nB to nY	$C_L = 15$ pF; $V_{CC} = 5$ V	7	11	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	10	20	pF

Notes

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

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

f_i = input frequency in MHz;

f_o = output frequency in MHz;

C_L = output load capacitance in pF;

V_{CC} = supply voltage in Volts;

N = total load switching outputs;

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

2. For 74HC08: the condition is $V_i = \text{GND to } V_{CC}$.

For 74HCT08: the condition is $V_i = \text{GND to } V_{CC} - 1.5$ V.

FUNCTION TABLE

INPUT		OUTPUT
nA	nB	nY
L	L	L
L	H	L
H	L	L
H	H	H

Note

1. H = HIGH voltage level;
L = LOW voltage level.

CD4001B, CD4002B, CD4025B Types

CMOS NOR Gates

High-Voltage Types (20-Volt Rating)

Quad 2 Input - CD4001B
Dual 4 Input - CD4002B
Triple 3 Input - CD4025B

■ CD4001B, CD4002B, and CD4025B NOR gates provide the system designer with direct implementation of the NOR function and supplement the existing family of CMOS gates. All inputs and outputs are buffered.

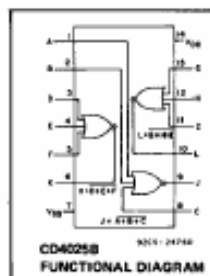
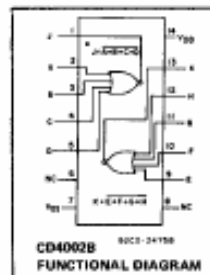
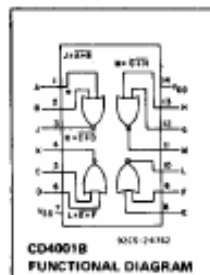
The CD4001B, CD4002B, and CD4025B types are supplied in 14-lead hermetic dual-in-line ceramic packages (D and F suffixes), 14-lead dual-in-line plastic packages (E suffix), 14-lead small-outline package (NSR suffix), and in chip form (H suffix).

Features:

- Propagation delay time = 60 ns (typ.) at $C_L = 60$ pF, $V_{DD} = 10$ V
- Buffered inputs and outputs
- Standardized symmetrical output characteristics
- 100% tested for maximum quiescent current at 20 V
- 5-V, 10-V, and 15-V parametric ratings
- Maximum input current of 1 μ A at 18 V over full package-temperature range; 100 nA at 18 V and 25°C
- Noise margin (over full package temperature range):
 - 1 V at $V_{DD} = 5$ V
 - 2 V at $V_{DD} = 10$ V
 - 2.5 V at $V_{DD} = 15$ V
- Meets all requirements of JEDEC Tentative Standard No. 13B, "Standard Specifications for Description of "B" Series CMOS Devices"

STATIC ELECTRICAL CHARACTERISTICS

CHARACTER- ISTIC	CONDITIONS			LIMITS AT INDICATED TEMPERATURES (°C)								UNITS
	V _O (V)	V _{IN} (V)	V _{DD} (V)					+25				
				-55	-40	+85	+125	Min.	Typ.	Max.		
Quiescent Device Current, I _{DD} Max.	—	0.5	5	0.25	0.25	7.5	7.5	—	0.01	0.25	μA	
	—	0.10	10	0.5	0.5	15	15	—	0.01	0.5		
	—	0.15	15	1	1	30	30	—	0.01	1		
	—	0.20	20	5	5	150	150	—	0.02	5		
Output Low (Sink) Current I _{OL} Min.	0.4	0.5	5	0.64	0.61	0.42	0.36	0.51	1	—	mA	
	0.5	0.10	10	1.6	1.5	1.1	0.9	1.3	2.6	—		
	1.5	0.15	15	4.2	4	2.8	2.4	3.4	6.8	—		
Output High (Source) Current, I _{OH} Min.	4.6	0.5	5	-0.64	-0.61	-0.42	-0.36	-0.51	-1	—	mA	
	2.5	0.5	5	-2	-1.8	-1.3	-1.15	-1.6	-3.2	—		
	9.5	0.10	10	-1.6	-1.5	-1.1	-0.9	-1.3	-2.6	—		
	13.5	0.15	15	-4.2	-4	-2.8	-2.4	-3.4	-6.8	—		
Output Voltage: Low-Level, V _{OL} Max.	—	0.5	5	0.05				—	0	0.05	V	
	—	0.10	10	0.05				—	0	0.05		
	—	0.15	15	0.05				—	0	0.05		
Output Voltage: High-Level, V _{OH} Min.	—	0.5	5	4.95				4.95	5	—	V	
	—	0.10	10	9.95				9.95	10	—		
	—	0.15	15	14.95				14.95	15	—		
Input Low Voltage, V _{IL} Max.	0.5, 4.5	—	5	1.5				—	—	1.5	V	
	1.9	—	10	3				—	—	3		
	1.5, 13.5	—	15	4				—	—	4		
Input High Voltage, V _{IH} Min.	0.5	—	5	3.5				3.5	—	—	V	
	1	—	10	7				7	—	—		
	1.5	—	15	11				11	—	—		
Input Current I _{IN} Max.		0.18	18	±0.1	±0.1	±1	±1	—	±10 ⁻⁵	±0.1	μA	



CD4001B, CD4002B, CD4025B Types

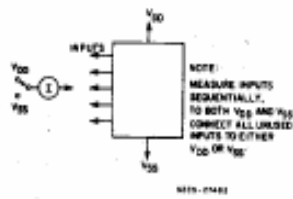


Fig. 13 - Input leakage current test circuit.

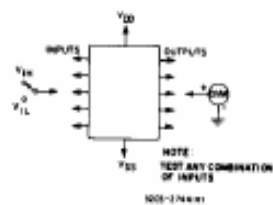


Fig. 14 - Input-voltage test circuit.

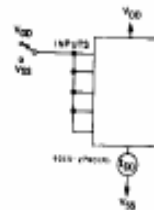
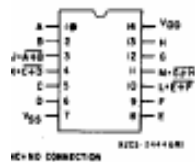
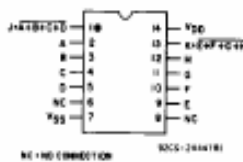


Fig. 15 - Quiescent-device current test circuit.

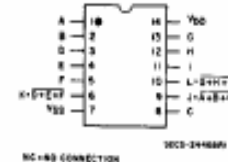
TERMINAL ASSIGNMENTS (TOP VIEW)



CD4001B

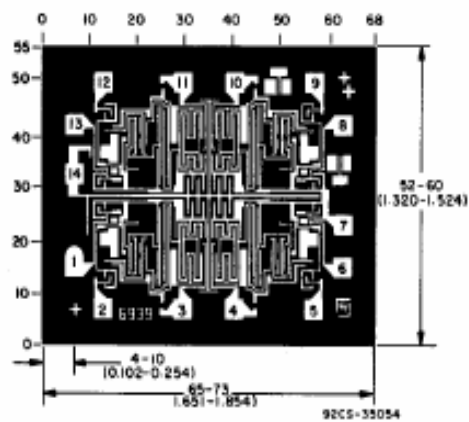


CD4002B

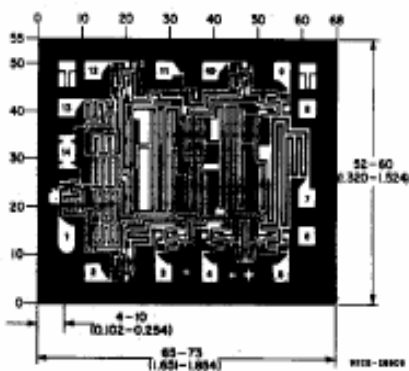


CD4025B

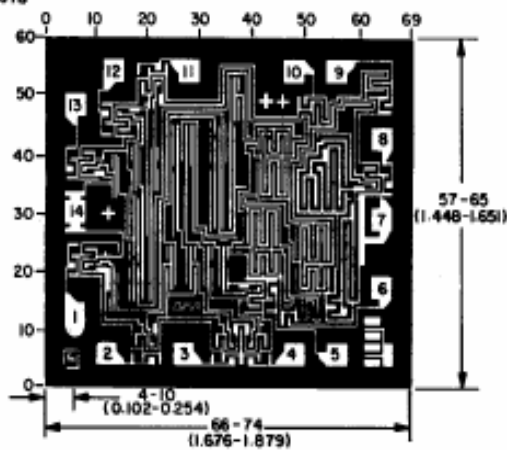
Chip Dimensions and Pad Layouts



CD4001B



CD4002B



CD4025B

MC14001B Series

B-Suffix Series CMOS Gates

MC14001B, MC14011B, MC14023B,
MC14025B, MC14071B, MC14073B,
MC14081B, MC14082B

The B Series logic gates are constructed with P and N channel enhancement mode devices in a single monolithic structure (Complementary MOS). Their primary use is where low power dissipation and/or high noise immunity is desired.

Features

- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- All Outputs Buffered
- Capable of Driving Two Low-power TTL Loads or One Low-power Schottky TTL Load Over the Rated Temperature Range.
- Double Diode Protection on All Inputs Except Triple Diode Protection on MC14011B and MC14081B
- Pin-for-Pin Replacements for Corresponding CD4000 Series B Suffix Devices
- Pb-Free Packages are Available*

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage Range	-0.5 to +18.0	V
V_{in}, V_{out}	Input or Output Voltage Range (DC or Transient)	-0.5 to $V_{DD} + 0.5$	V
I_{in}, I_{out}	Input or Output Current (DC or Transient) per Pin	± 10	mA
P_D	Power Dissipation, per Package (Note 1)	500	mW
T_A	Ambient Temperature Range	-55 to +125	°C
T_{stg}	Storage Temperature Range	-65 to +150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

Maximum ratings are those values beyond which device damage can occur. Maximum ratings applied to the device are individual stress limit values (not normal operating conditions) and are not valid simultaneously. If these limits are exceeded, device functional operation is not implied, damage may occur and reliability may be affected.

1. Temperature Derating:

Plastic "P" and "D" Packages: - 7.0 mW/°C From 65°C To 125°C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Unused outputs must be left open.

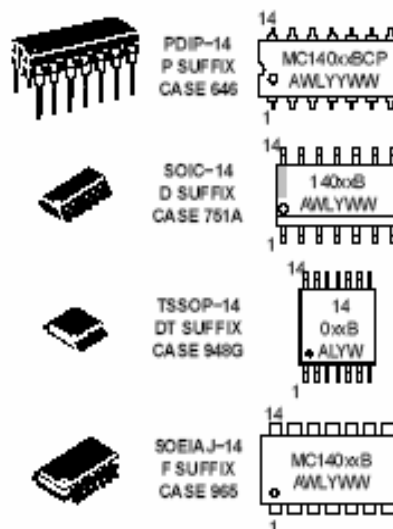
*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRMWD.



ON Semiconductor®

<http://onsemi.com>

MARKING DIAGRAMS



xx = Specific Device Code
A = Assembly Location
WL, L = Wafer Lot
YY, Y = Year
WW, W = Work Week

DEVICE INFORMATION

Device	Description
MC14001B	Quad 2-Input NOR Gate
MC14011B	Quad 2-Input NAND Gate
MC14023B	Triple 3-Input NAND Gate
MC14025B	Triple 3-Input NOR Gate
MC14071B	Quad 2-Input OR Gate
MC14073B	Triple 3-Input AND Gate
MC14081B	Quad 2-Input AND Gate
MC14082B	Dual 4-Input AND Gate

ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 8 of this data sheet.