



**Universidad Tecnológica Nacional  
Facultad Regional Córdoba**

**TRABAJO FINAL  
TECNICAS DIGITALES II**

**“SISTEMA  
DE  
ADQUISICIÓN  
DE  
MUESTRAS  
DE  
TEMPERATURA”**

**AUTORES:**

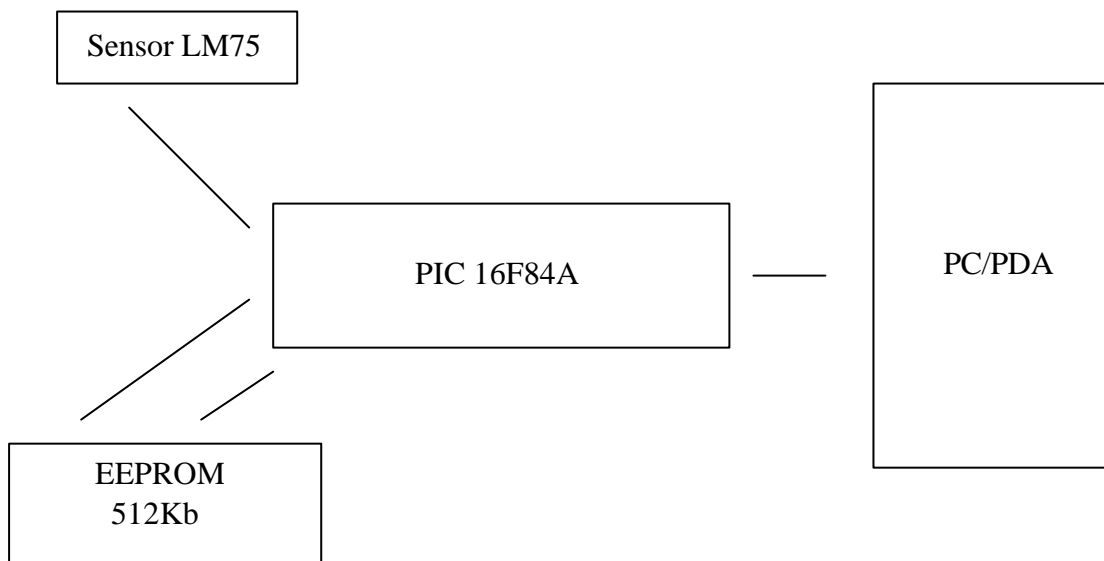
**TOUZ, FEDERICO ANDRES LEG: 43310  
MONDINO, MAURO ANDRES LEG: 43311**

**AÑO 2003**

## Objetivo del Práctico

Este práctico consiste en tomar muestras de temperaturas y almacenarlas en memoria, para luego recopilar los datos con cualquier dispositivo con conexión serial y capacidad de procesar datos en ASCII.

Como muestra la figura siguiente se a escogido el sensor de temperatura LM75 de National por su practicidad , tambien como memoria de almacenamiento utilizamos una EEPROM 24AA512 de Microchip, todo esto esta controlado por un microcontrolador 16F84A del mismo fabricante.



## Implementación

La implementación de este práctico podría estar vinculada con cualquier chequeo de temperatura de cualquier dispositivo, fijándole el tiempo de de toma de la muestra. Con esto, se quiere decir que no solamente podría usarse para la toma de temperatura de dispositivos, sino también para la toma de temperatura de por supuesto, cualquier proceso que se quiera controlar.

Debido a su versatilidad con respecto a su tiempo de muestreo, y también debido al tipo de conexionado que se utiliza para con el sensor (conexión i2c), podríamos decir que es bastante interesante el uso de sus aplicaciones.

con respecto a la elección de los elementos utilizados en el práctico, se eligió el LM75, debido a su pequeño tamaño, a su tipo de conexionado serial, que nos permite sensor hasta una distancia máxima de 50 metros, lo que implica que solo se necesitarían mandar solo 2 cables para intercomunicarse con su base de toma de datos, por llamarlo así (lógicamente, también se debe tener en cuenta la alimentación del integrado y masa). La elección de la memoria, también tiene su explicación. Esto fué debido a que como utiliza el mismo protocolo de comunicación que el sensor , se utiliza no sólo una programación similar para ambos, sino que también se pueden utilizar los mismos 2 cables que le transportan y reciben información al sensor, para, valga la redundancia, transportar y recibir información desde y hacia la memoria.

Se utilizó el pic 16f84A, el cual no posee UART, con lo cual, todo lo referido a

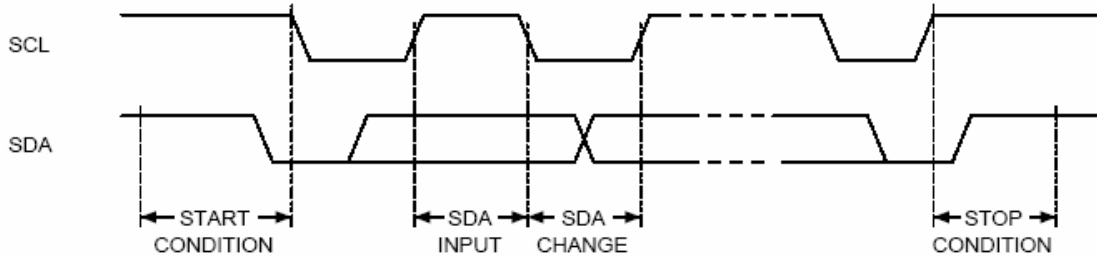
## comunicación Conexión I2C



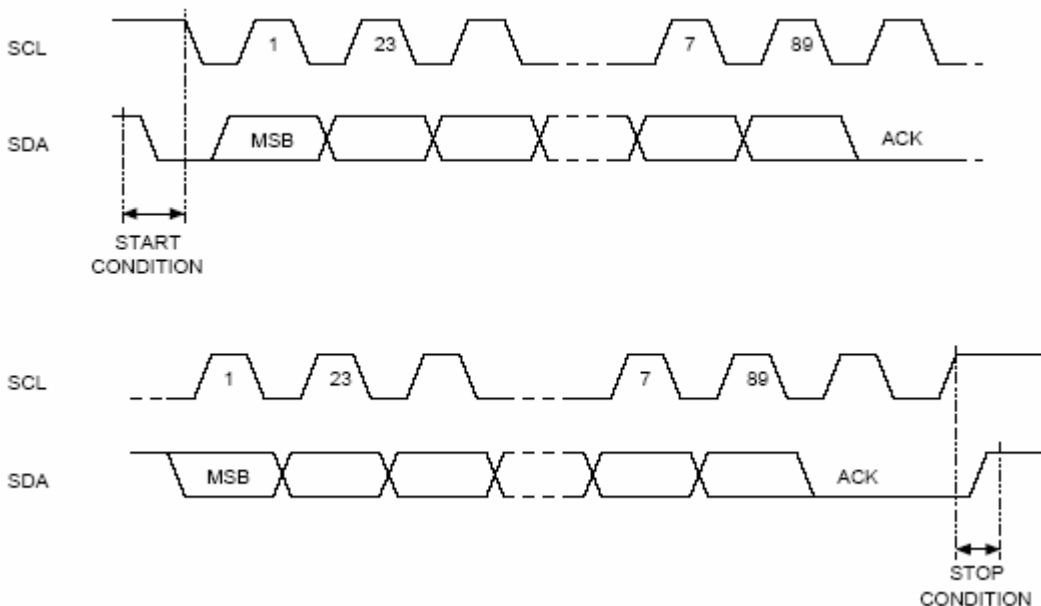
serial e I2C, debió ser implementado por software.

El bus I<sup>2</sup>C es un bus diseñado para que sobre éste puedan colocarse varios dispositivos dentro de la misma tarjeta electrónica (comunicación multipunto), cada dispositivo tendrá una dirección lógica asignada físicamente mediante los pines A0, A1 y A2 de acuerdo al nivel lógico al que estos sean conectados. Para poder interactuar con un dispositivo que trabaje bajo i2c, es necesario básicamente, enviarle la dirección del dispositivo al cual nos referimos, indicándole si se lo va a escribir o a leer, y enviarle o recibirle lo datos que queremos convenientes.

El principio básico de funcionamiento de este tipo de conexionado, es el siguiente:

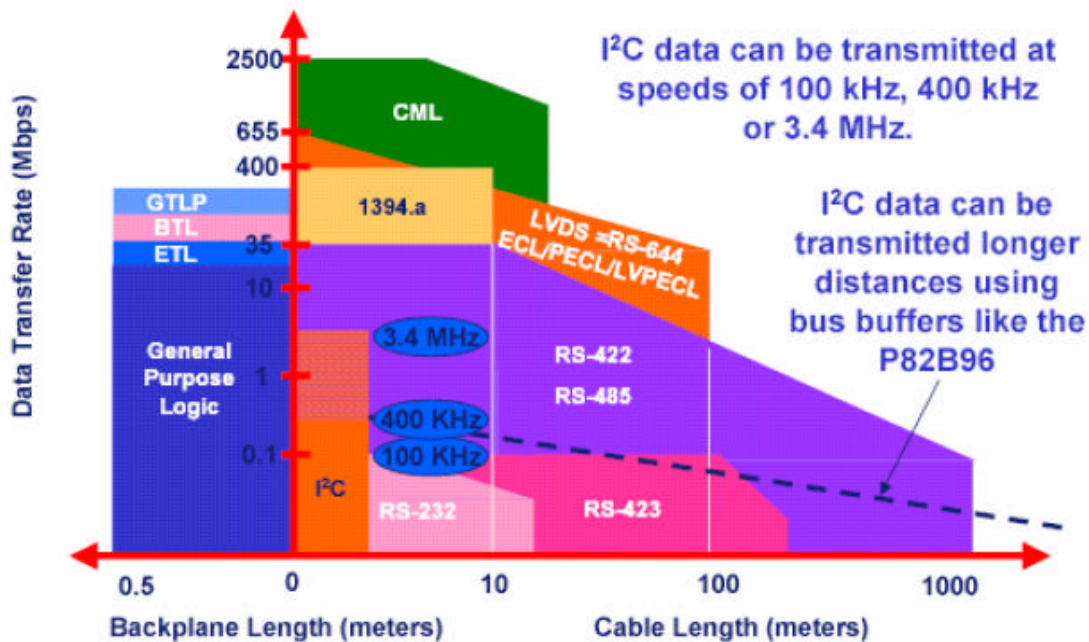


Existen dos señales para establecer la comunicación, una es la señal de reloj SCL y la otra es la de datos bidireccional SDA. Para iniciar la comunicación sobre un dispositivo I<sup>2</sup>C debe realizarse la secuencia denominada bit de START que consiste en pasar la línea de datos SDA de nivel alto a bajo mientras que la línea SCL permanece en alto. Para la culminar la comunicación con el dispositivo I<sup>2</sup>C debe ejecutarse la secuencia denominada bit de STOP la cual consiste en pasar la línea de datos SDA de nivel bajo a alto mientras que la línea de reloj SCL permanece en alto. Un bit de datos es aceptado por el dispositivo mientras que sobre la línea de datos SDA permanece el nivel adecuado al bit en cuestión, y sobre la línea de reloj SCL se lleva a cabo un pulso, es decir, el paso de nivel de bajo a alto y luego de alto a bajo. Los tiempos implicados en esta secuencia dependen básicamente del fabricante del dispositivo.

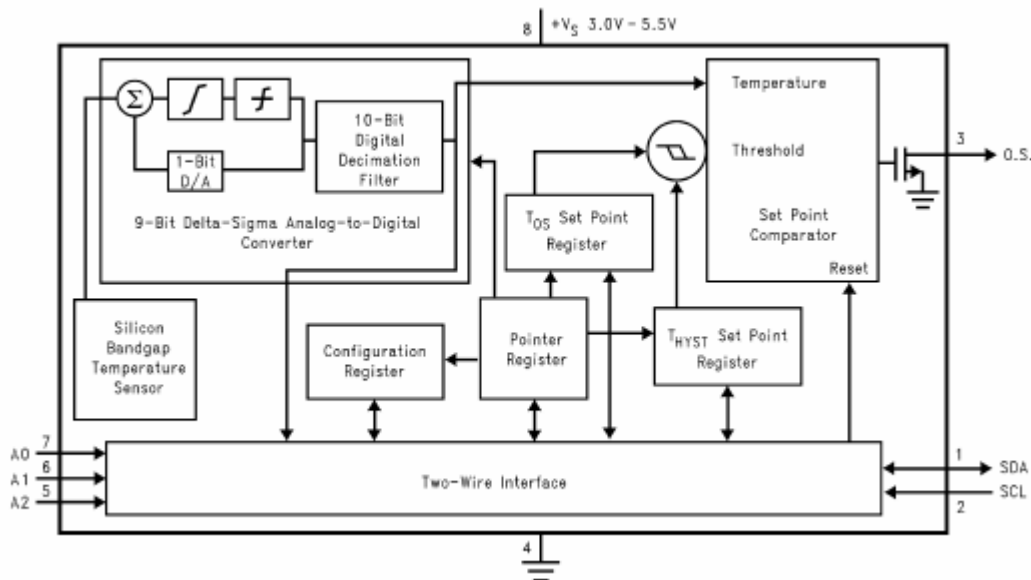


Por ultimo una de las implicancias más importantes de utilizar i2c, es que permite un conexionado capaz de soportar grandes distancias

## Transmission Standards



## Sensor de temperatura LM75



Este sensor utiliza el protocolo de comunicación i2c, y consta de un conversor analógico-digital, por lo que de acuerdo a la temperatura que se esté midiendo, este sensor nos enviará un determinado dato en hexadecimal, que luego con la respectiva tabla de conversión, podremos interpretar lo que nos está enviando.

De acuerdo con el diagrama de tiempos del LM75 este sensor responde a un byte denominado address byte que contiene la dirección 0x9X, con lo cual podemos decir que los primeros 4 bits, ya está, prefijados en el dispositivo, y los 3 bits de la parte baja pueden ser seteados por nosotros mismos, para poder completarle su dirección.

1	0	0	1	A2	A1	A0
MSB				LSB		

Vale aclarar que el último bit de la dirección, corresponde a la operación que se desea realizar sobre el sensor, ya se escritura o lectura.

En este caso LM75 será únicamente utilizado para leerlo, lo que implica que su bit menos significativo está en 1, ya que el último bit en 0 implica escritura.

Luego seguido de un señal de acknowledge (reconocer), se puede leer el word proveniente del registro de temperatura, como se observa en la figura, el byte menos significativo contiene la información de + o - .

#### 1.12 TEMPERATURE REGISTER

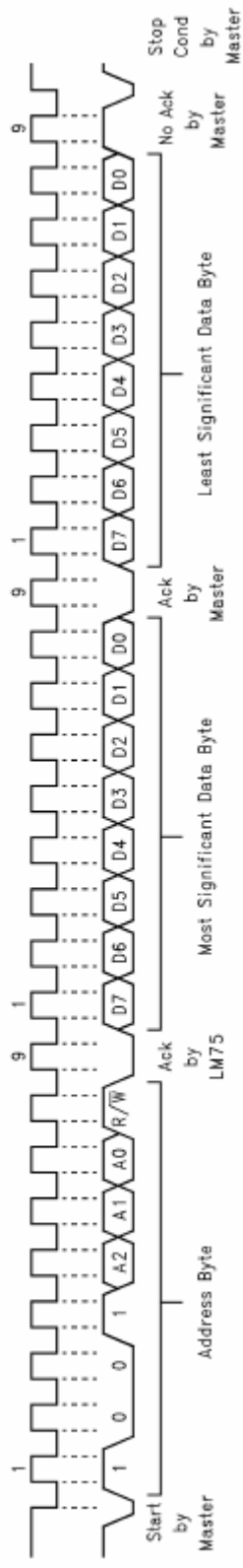
(Read Only):

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MSB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB	X	X	X	X	X	X	X

D0–D6: Undefined

D7–D15: Temperature Data. One LSB = 0.5°C. Two's complement format.

## I<sup>2</sup>C Timing Diagrams



Con respecto al rango de variación de temperatura en el cual hacemos trabajar al sensor, es de 0 a 125 grados centígrados.

De acuerdo a la tabla de conversión que se nos presenta en la hoja de datos, se vé que cuando el dispositivo está midiendo 125 grados centígrados, nos entrega el número 0xFA, y por ejemplo cuando nos está entregando 25 grados centígrados, nos entrega el número 0x32, etc. Con esto podemos fijar la manera en que convertiremos los datos de temperatura en hexadecimal, a datos de temperatura, legible y entendible para nosotros, en la escala centígrada.

Temperature	Digital Output	
	Binary	Hex
+125°C	0 1111 1010	0FAh
+25°C	0 0011 0010	032h
+0.5°C	0 0000 0001	001h
0°C	0 0000 0000	000h
-0.5°C	1 1111 1111	1FFh
-25°C	1 1100 1110	1CEh
-55°C	1 1001 0010	192h

## Memoria 24AA512



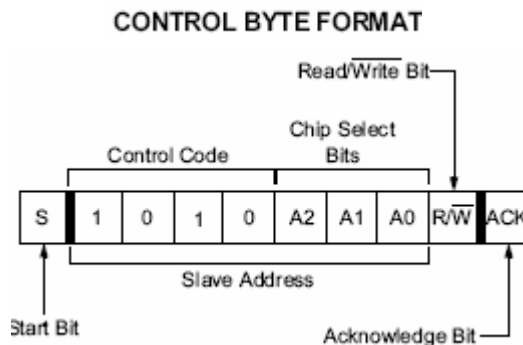
Estamos utilizando la memoria antes mencionada, con el objetivo de almacenar los datos de temperaturas tomados por el sensor LM75.

Esta memoria, tiene una capacidad de 512K, y trabaja también con i2c. también consta de 3 direcciones de entrada, que permite referirnos a ella, como lo hemos estado hablando con el sensor, en el momento de la comunicación.

La memoria cuenta con un almacenamiento de un byte, por cada dirección de la misma, lo que se quiere decir es que en la posición 0000 se almacenará como máximo, un byte. Para la paginación de la misma, podemos decir que los datos se almacenan a partir de la dirección 0000, hasta la dirección FFFF, y una vez que se llegó al máximo (FFFF), se vuelve a sobrescribir desde el principio (0000).

Otra de las cosas que cabe rescatar, es que ésta memoria cuenta con un pin (el cual nosotros no estamos utilizando, pero que lo podríamos haber hecho), el cual permite que una vez que se han almacenado los datos que se quisieron guardar, poniendo éste pin del que hablamos, en un nivel alto, se protege la memoria para que no pueda ser borrada. Obviamente que si se quiere volver a escribir, no hay más que poner este pin de protección a cero, y realizar nuevamente otra escritura, si así se lo desea.

para poder comunicarnos con este dispositivo, es necesario tener varias cosas en cuenta:



a) protocolo de

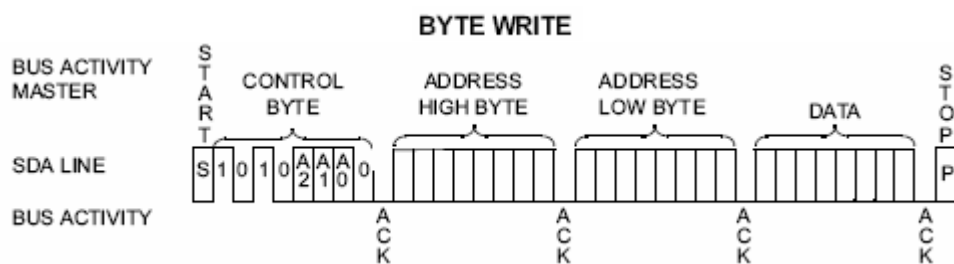
comunicación para

escritura: se debe en primer lugar enviar por el bus de transmisión, la dirección de la memoria, la cual ya tiene preestablecida la parte superior de su dirección como 0xA.

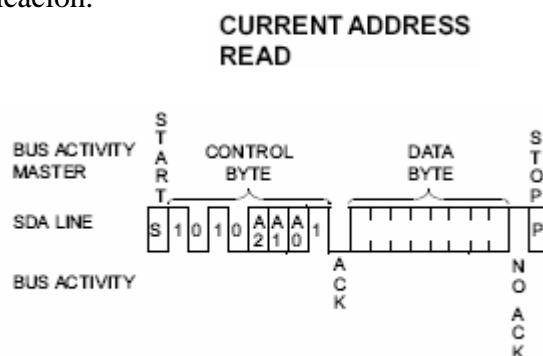
Lo que hacemos nosotros para referirnos a ella, es enviarle la dirección 0xA0, lo cual la información que nos permite manejar esto es que por supuesto la parte superior de la dirección se refiere a la parte preestablecida por el fabricante, y los tres bits más significativos de la parte baja, significa que hemos colocado los tres pines de direccionamiento que tiene la memoria, a masa.

Con respecto al bit menos significativo de la parte baja de la dirección, en este caso también es igual a lo que estábamos hablando en el LM75, que sería que si este bit es un 1, es porque queremos leer la memoria, y que si este bit es un cero, es que se la quiere escribir.

Luego se le debe esperar una respuesta por parte de la memoria, refiriéndose a que se pudo acceder exitosamente a ella, y ahí nomás se le manda la dirección de la parte alta en donde se quiere escribir, se vuelve a esperar una respuesta para comprobar que recibió la dirección, y se manda la dirección de la parte baja en donde se quieren almacenar los datos, se vuelve a esperar el control, y si todo está bien, se manda el byte a escribir, y se termina la comunicación.



b) protocolo de comunicación para lectura: En el caso de la lectura es un poco más complejo que el de la escritura, porque lo primero que se debe hacer es también mandar por el bus la dirección del dispositivo pero en modo de escritura (bit menos significativo en cero), se espera la respuesta de que todo está bien, se le envía la dirección de la parte alta a donde se quiere acceder a leer, se espera respuesta, se le manda la dirección de la parte baja, se espera respuesta, y se manda otra vez la dirección de la memoria a la cual responde ella (la cual es 0xA0), pero ahora en el modo de lectura (con el bit menos significativo en 1), se espera respuesta, se toma el byte que nos manda, se envía respuesta por parte de nosotros hacia la memoria, de que hemos recibido bien los datos, y se termina la comunicación.





## Dispositivo Palm M100

## Palm OS®

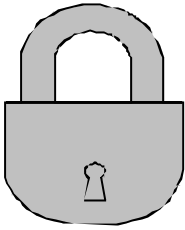
Hemos elegido este dispositivo gracias a su portabilidad y a la capacidad de operar como cualquiera PC , esta cuenta con una puerto serie en el cual será usado para dicho proyecto.

Con este tomaremos las muestras, almacenándolas e imprimiéndolas en la pantalla del LCD.



Utilizaremos el programa Online 2.0 que cumple la función de HyperTerminal ya que Palm OS carece de ella. Este será explicado mas adelante.

## ESQUEMATICO

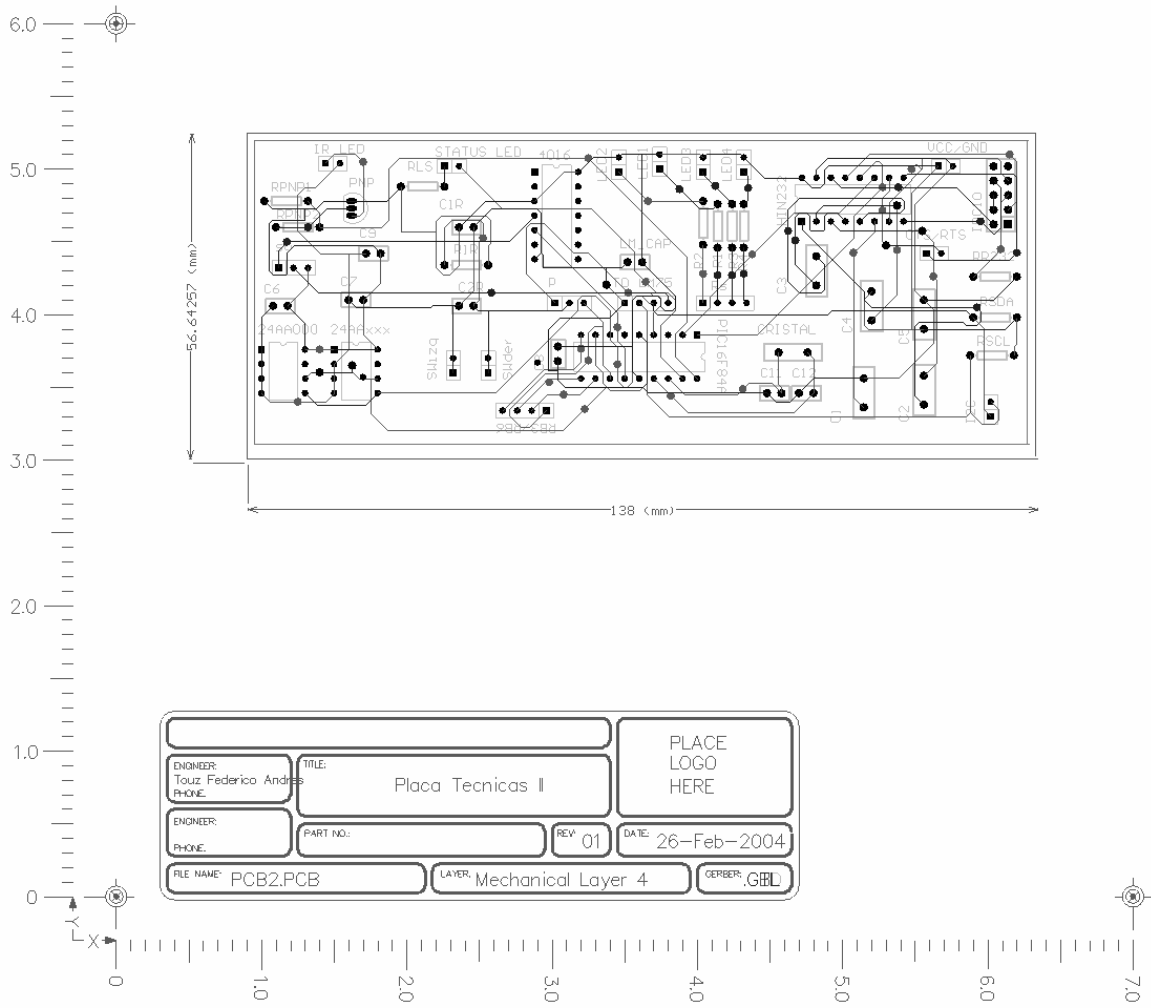


El esquemático no está incluido en este informe por razones de copyright©.  
Por cualquier consulta puede contactarse con los autores.

[fedetouz@yahoo.com](mailto:fedetouz@yahoo.com)

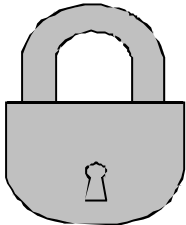
[mandresmondino@hotmail.com](mailto:mandresmondino@hotmail.com)

# PCB



## SOFTWARE

El software se creo en PICC de HI-TECH el cual nos permite programar en ANSI C  
Las librerías como DELAY.C y I2C.C son completamente creadas por HI-TECH , en  
cambio iserial fue creada y configurada en parte por los autores del proyecto.



El software no esta incluido en este informe por razones de copyright©.  
Por cualquier consulta puede contactarse con los autores.

[fedetouz@yahoo.com](mailto:fedetouz@yahoo.com)

[mandresmondino@hotmail.com](mailto:mandresmondino@hotmail.com)

.....  
CONFIGURACION ON LINE 2.0  
.....

Una vez instalado el soft siga los siguientes pasos:

- 1) Seleccionar el botón demo
- 2) Seleccionar comunicación
- 3) Seleccioné Serial
- 4) Seleccioné 1200 baudios, 8 data bits, Parity N, Stop bits 2,32K buffer
- 5) Seleccioné Terminal
- 6) Click en VT100,LARGE.
- 7) En la pantalla principal seleccione ON y listo



Online VT-100™ is Copyright 1999-2003 Mark/Space. Mark/Space and Online are trademarks of Mark/Space. All other product names are trademarks or registered trademarks of their respective holders.

## Funcionamiento



- \_\_\_\_\_ Boton de liberación de EEPROM (atención trate de estar conectado con On line software)
  
- \_\_\_\_\_ Enciende al tener el botón apretado por unos seg.
- \_\_\_\_\_ Led infrarrojo no utilizado
- \_\_\_\_\_ Led de escritura en EEPROM
  
- \_\_\_\_\_ Conexión Serial
- \_\_\_\_\_ Led error de BUS I2C

## **Bibliografía**

[www.national.com](http://www.national.com)

[www.microchip.com](http://www.microchip.com)

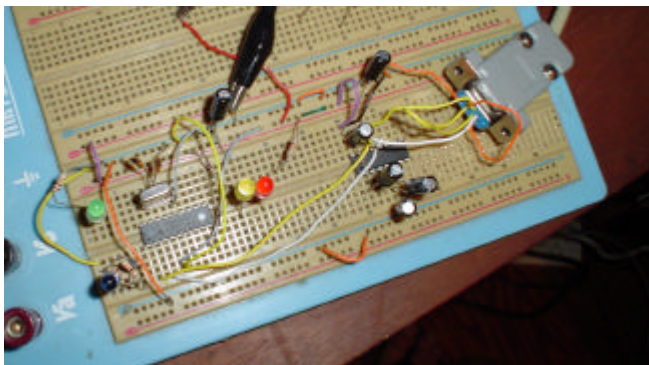
[www.philips.com](http://www.philips.com) ( I2C BUS)

[www.intersil.com](http://www.intersil.com) (HIN232)

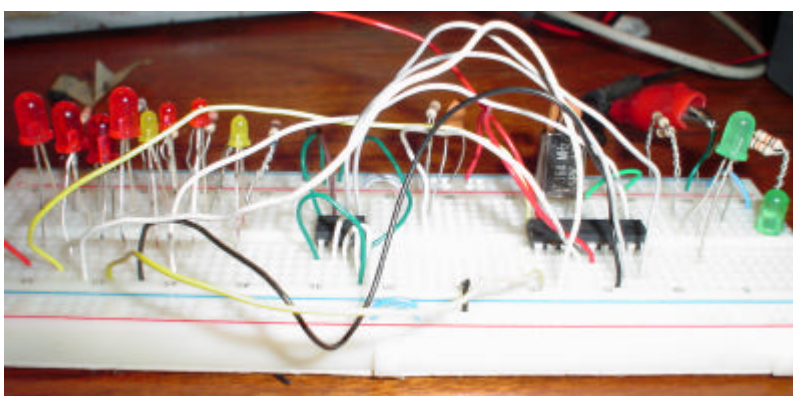
[www.markspace.com](http://www.markspace.com)

## HISTORIAL

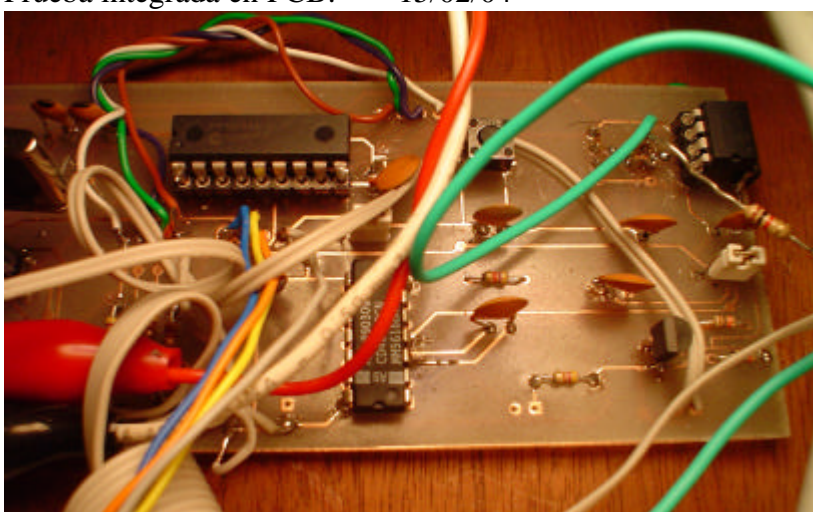
Primera prueba de la comunicación serial 02/02/04



Primera prueba de la EEPROM 02/02/04



Prueba integrada en PCB: 15/02/04



**Autores:**

**Touz Federico Andrés , Mondino Mauro Andrés**



23 1 2004

FECHA DE FINALIZACION 26 de febrero de 2004



