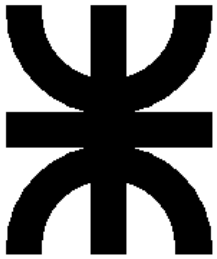


UNIVERSIDAD TECNOLOGICA NACIONAL
FACULTAD REGIONAL SAN NICOLAS

INGENIERIA EN ELECTRONICA

PROBLEMA DE INGENIERÍA

TECNICAS DIGITALES III

SIMULADOR DE PRESENCIA GSM

Integrantes**:**

-Berthet, Gustavo
-Furch, Juan
-Lucci,José

Docentes**:**

- Profesor: Poblete Felipe
- Auxiliar: Gonzalez Mariano

AÑO 2012

INDICE

OBJETIVOS DEL TRABAJO

El objetivo del trabajo es poder simular la presencia humana en una casa, para dicho caso diseñaremos un activador remoto GSM que controlará la iluminación y artefactos electrónicos para generar la sensación que alguien esta interactuando con la casa.

MATERIAS INTEGRADAS

• Técnicas Digitales I
• Electrónica Aplicada II

POSIBLES APLICACIONES

Como el proyecto es controlar relevadores por GSM, es posible activar y desactivar remotamente, focos, motores, sistemas de agua para riego, actuadores, válvulas y otros dispositivos. En caso necesario, si la potencia de las cargas fuese muy alta, pueden conectarse relevadores auxiliares ó contactores de mayor potencia en cascada con los relevadores.

BIBLIOGRAFÍA

• Apuntes de cátedra.
• Foros de comunicaciones y electrónica avanzada
• Datasheets
• Paginas:

http://roso-control.com/_Download/Cursos/MDK-B28/Cap_No_08.pdf
www.perso.wanadoo.es/pictob/micropic.htm
http://solocodigo.com/31050/microcontrolador-nokia-3220/100/
http://www.cpkb.org/wiki/File:Nokia_3200_pinout.jpg
http://micropic.wordpress.com/2007/06/08/manejo-de-interrupciones/
http://quidel.inele.ufro.cl/~jhuircan/PICRS232.html
http://www.cpkb.org/wiki/Nokia_3220_pinout

• Libros:
Compilador C CCS y simulador proteus para microcontroladores pic – Autor: Eduardo Garcia Breijo – Editorial alfaomega

DESARROLLO

INTRODUCCIÓN

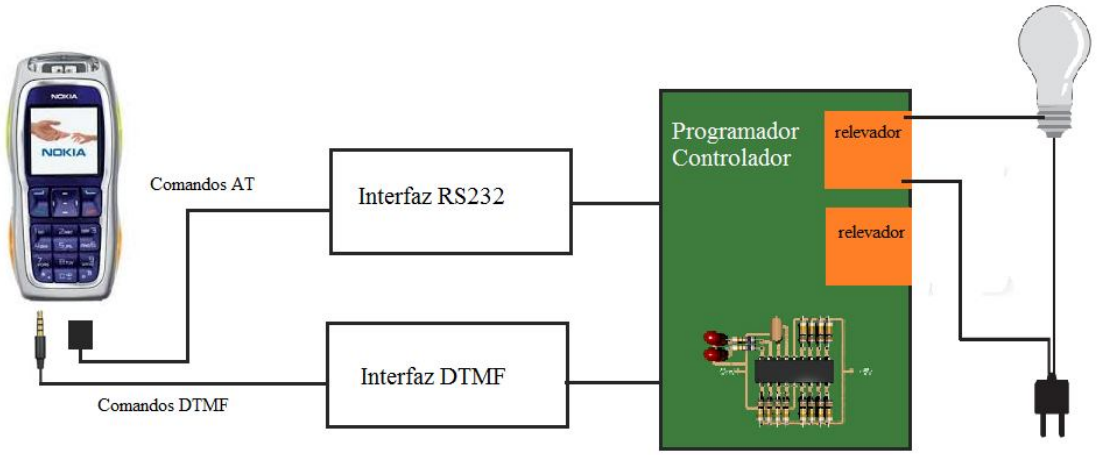


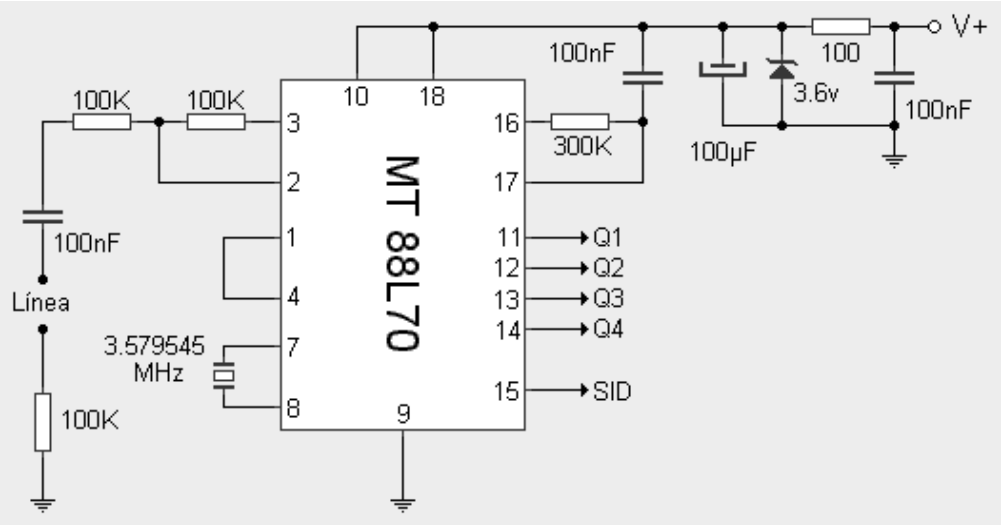
Fig 1

TEORÍA DE FUNCIONAMIENTO

Se desarrollara el prototipo de un controlador GSM. Se trata de un sistema con base en un módulo controlador-programador (CP) con capacidad de activación de 2 dispositivos y con control remoto a través de la red GSM. El control puede realizarse desde cualquier teléfono celular ó local.
El usuario utiliza su teléfono remoto y realiza primeramente una llamada al número del teléfono NOKIA integrado al controlador GSM. El módulo CP detecta el ingreso de la llamada y contesta el teléfono NOKIA en forma automática. Una vez establecido el enlace de audio, el usuario puede activar ó desactivar remotamente cualquiera de los 2 relevadores del controlador, pulsando comandos por medio del teclado de su teléfono, a través de tonos DTMF. Como respuesta a la acción realizada, el sistema generará una serie de tonos para asegurarnos que la tarea se realizó correctamente. Por ejemplo el tono del pulso 6 se emitirá una vez si la salida 1 fue activada correctamente, y 2 veces para confirmar que dicha salida se desactivo.

DECODIFICADOR DE TONOS DTMF

El circuito que presentamos posee excelentes características en cuando a su relación costo/prestaciones. Con sólo un circuito integrado (cuyo precio no supera los 2 dólares) y un puñado de componentes externos discretos se obtiene un dispositivo capaz de entregar el código binario de la tecla pulsada en un teléfono por tonos multifrecuentes. Este circuito, además de decodificar las clásicas teclas del cero al nueve, asterisco y numeral, puede identificar las teclas A, B, C y D que usualmente no están presentes en la mayoría de los teléfonos comerciales, pero que la especificación DTMF las incluye.



El circuito está preparado para ser alimentado con 5v, presentes en cualquier circuito TTL o microcontrolado. La resistencia de 100 ohms limita la corriente y el diodo zener hace las veces de limitador de tensión, bajándola a 3.6v que es lo que el chip requiere para funcionar correctamente. Los capacitores aledaños a esos componentes cumplen con la función de filtrar la tensión de alimentación. La señal proveniente de la línea telefónica es aislada por medio de dos resistencias de 100K y un capacitor de 100nf. Este último impide el paso de corriente, pero deja circular señal de audio. Para su funcionamiento el circuito integrado requiere una base de tiempos, generada en este caso por el cristal de cuarzo de 3.579545MHz. Nótese que este cristal es muy común en el mercado dado que es el empleado para los sistemas de color de los equipos de TV. Una vez que un tono es recibido, decodificado y validado como correcto su valor binario es colocado en los terminales Q1, Q2 Q3 y Q4. A su vez, el terminal SID sube indicando la presencia del dato en la salida. Este terminal permanece alto durante el tiempo que el tono DTMF siga presente en el sistema, o sea que refleja el tiempo que el teléfono remoto permanece pulsado.
El circuito integrado incluye filtros contra ruido, RF y armónicos. Además, incluye controles automáticos de ganancia y nivel de señal para adecuar cualquier tipo de condición de trabajo. Es por ello que la cantidad de componentes externos es ínfima.

Datos presentes en la salida

Tecla	Q1	Q2	Q3	Q4
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1

B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

COMANDOS AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal modem.
En un principio, el juego de comandos AT fue desarrollado en 1977 por Dennis Hayes como un interfaz de comunicación con un modem para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías Microcomm y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.
Los comandos AT se denominan así por la abreviatura de *attention* .

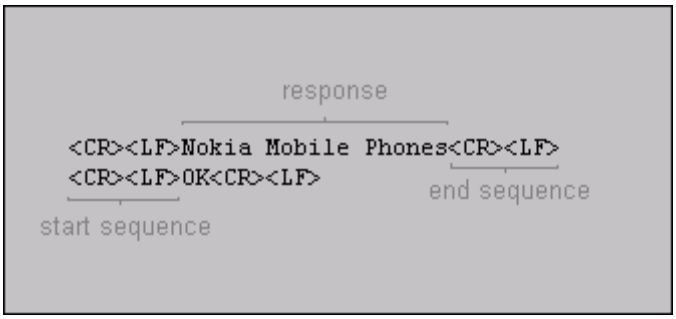
Aunque la finalidad principal de los comandos AT es la comunicación con modems, la telefonía móvil GSM también ha adoptado como estandar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales. Este juego de instrucciones puede encontrarse en la documentación técnica de los terminales GSM y permite acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal. Queda claro que la implementación de los comandos AT corre a cuenta del dispositivo GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable de serie, canal Infrarrojos, Bluetooth, etc. De esta forma, es posible distinguir distintos teléfonos móviles del mercado que permiten la ejecución total del juego de comandos AT o sólo parcialmente. Por ejemplo, Nokia 6820 no permite la ejecución de comandos AT relativos al manejo de la memoria de agenda de contactos y llamadas pero sí que permite acceder al servicio SMS; Nokia 6600 (basado en Symbian) no permite la ejecución de comandos AT relativos a la gestión de la agenda ni de SMSs.

El envío de comandos AT requiere la siguiente estructura:

· Petición:

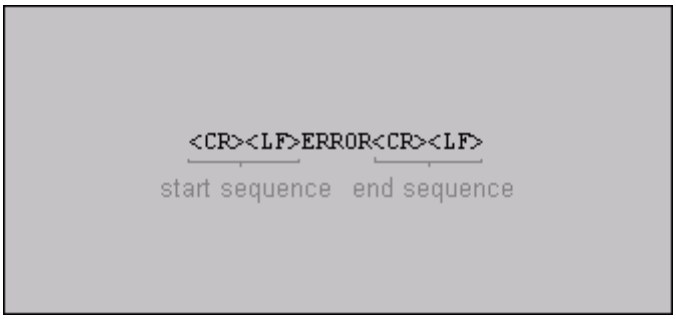
<CR> ... Carriage return

· **Respuesta correcta:**



<CR> ... Carriage return
<LF> ... Line feed

· **Respuesta incorrecta:**



<CR> ... Carriage return
<LF> ... Line feed

El juego de comandos AT específico para telefonía móvil GSM
Basado en la referencia simplificada del juego de comandos AT
en http://www.activexperts.com/activsms/atcommands/nokia/
Todos los comandos AT han sido testeados con éxito utilizando un Nokia 8220.

Notación empleada:
Comando AT: [Definición técnica]
- Funcionalidad del comando
- Sintaxis: Petición | Respuesta
- Respuesta obtenida al testear el comando con Nokia 8310

Comandos de Llamada

ATD: [Dial Command]
- Inicia una llamada telefónica
- Sintaxis: ATD64612345 para una llamada de Datos.
ATD64612345; para una llamada de Voz. (Importante la notación ';')
ATD>"Gospel"; para llamar al contacto almacenado en la agenda con el texto asociado Gospel.

Comandos de Operador

AT+CCFC: [Call Forwarding Number]
- Gestiona el Desvío de Llamadas. Permite redireccionar llamadas entrantes a otro número de teléfono.
- Sintaxis:
AT+CCFC=<razón>,<modo>,<número>,<tipo>,<clase>,[<subaddr>,<satype>,[<time>]]
<razón> Razón por la cual entra en acción el desvío de llamada.
0 - incondicional
1 - si teléfono ocupado
2 - si no obtiene respuesta
3 - si inalcanzable
4 - todos los desvíos de llamadas
5 - todos los desvíos de llamadas condicionales
<modo> Estado del desvío de llamada.
0 - desahabilitado
1 - habilitado
2 - query status
3 - registro
4 - erasure (borrado)
<número> Cadena de texto con el número de teléfono destino del desvío de llamada. Se especifica en el formato indicado en el campo <type>
<tipo> Tipo de código de dirección de teléfono:
145 - para código internacional +
129 - en otro caso
<clase> Código que representa la clase de información que contiene la llamada a desviar.
1 - voz
2 - datos
4 - fax
7 - cualquier clase (por defecto)
<time> Tiempo en segundos a esperar antes de desviar la llamada.
1..30 (por defecto, 20)
<status> Estado de la opción desvío de llamadas. (Sólo en respuesta AT)
0 - no activo
1 - activo
- Ejemplo: Implementación del comando en Blooover: "AT+CCFC=0,3,\"+4913377001\",145,7\r"
Vemos que utiliza los siguientes parámetros:
<razón> = 0, incondicional
<modo> = 3, registro
<número> = +4913377001
<tipo> = 145, formato de código internacional
<clase> = 7, cualquier clase de información a desviar

Comandos de Control del Teléfono

AT+CPAS: [Phone Activity Status]

1) AT+CPAS=?
- Muestra la implementación del comando.
- Sintaxis: AT+CPAS=? | +CPAS: (lista de estados soportados)
0 - Ready (Encendido pero inactivo)

1 - Unavailable (No disponible)
2 - Unknown (Desconocido)
3 - Ringing (Llamada entrante en proceso)
4 - Call in progress (Llamada saliente en proceso)
5 - Asleep (Dormido)

- Respuesta: +CMGD: (0,2,3,4)

2) AT+CPAS
- Informa del estado de actividad del teléfono.
- Sintaxis: AT+CPAS | +CPAS: <estado>
- Respuesta: +CPAS: 0, en estado normal de inactividad.
+CPAS: 3, si el teléfono atacado está sonando a causa de una llamada entrante.

AT+CBC: [Battery Charge]
- Devuelve el estado de carga de la batería.
- Sintaxis: AT+CBC | +CBC: <bcs>, <bcl>
<bcs> = 0 indica que el teléfono está conectado a una batería.
<bcl> = 0 indica que el teléfono tiene la batería agotada.
= 1..100 indica el porcentaje de carga que aún queda por agotar.
- Respuesta: +CBC:0,56

AT+CGMI: [Request Manufacturer Identification]
- Petición de identificación del Fabricante (Marca del teléfono).
- Sintaxis: AT+CGMM | <fabricante>
- Respuesta: Nokia Mobile Phones

AT+CGMM: [Request Model Identification]
- Petición de identificación del modelo de teléfono.
- Sintaxis: AT+CGMM | <modelo>
- Respuesta: Nokia 8310

AT+CGSN: [Request Product Serial Number Identification]
- Petición de identicación del número de serie del producto.
- Sintaxis: AT+CGSN | <IMEI>
- Respuesta: 1234567890etc

AT+CSQ: [Signal Quality]
- Devuelve el estado de calidad de la señal de cobertura.
- Sintaxis: AT+CSQ | +CSQ: <rssi>,<ber>
<rssi> = 0 indica -113 dBm o menos
= 1 indica -111 dBm
= 2..30 indica -109..-53 dBm
= 31 indica -51dBm o más
= 99 indica desconocido
<ber> = 99 indica porcentaje desconocido
- Respuesta: +CSQ: 13,99

AT+CPBS: [Select Phone Book Memory Storage]

1) AT+CPBS?

- Informa de los dispositivos de memoria que soporta el teléfono para almacenar las distintas listas de contactos.
- Sintaxis: AT+CPBS? | +CPBS: "XX", donde "XX" se sustituye por el dispositivo de almacenamiento:
  "SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
  "TA" - TA phonebook list [Lista de contactos del terminal]
  "LD" - SIM last dialing list [Lista de números marcados]
  "DC" - Dialled call list [Lista de llamadas realizadas]
  "RC" - ME received calls list [Lista de llamadas recibidas]
  "MC" - ME missed call list [Lista de llamadas perdidas]
  "EN" - Emergency number list [Lista de números de emergencia]
  "FD" - SIM fix dialing list
  "MT" - ME + SIM conbined list
  "ON" - SIM o ME own number list
- Respuesta: +CPBS: "SM"

2) AT+CPBS="XX"
- Selecciona por defecto uno de los dispositivos de memoria que soporta el teléfono para almacenar las distintas listas de contactos.
- Sintaxis: AT+CPBS="XX", donde "XX" se sustituye por el dispositivo de almacenamiento:
  "SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
  "TA" - TA phonebook list [Lista de contactos del terminal]
  "LD" - SIM last dialing list [Lista de números marcados]
  "DC" - Dialled call list [Lista de llamadas realizadas]
  "RC" - ME received calls list [Lista de llamadas recibidas]
  "MC" - ME missed call list [Lista de llamadas perdidas]
  "EN" - Emergency number list [Lista de números de emergencia]
  "FD" - SIM fix dialing list
  "MT" - ME + SIM conbined list
  "ON" - SIM o ME own number list

AT+CPBR: [Read Phone Book Entry]

1) AT+CPBR=?
- Informa del tamaño de la agenda de contactos.
- Sintaxis: AT+CPBR=? | +CPBR: <(1-n)>,<nlen>,<tlen>
  <(1-n)> indica el rango de índices que la agenda puede contener.
  <nlen> indica la longitud máxima permitida para un número de teléfono.
  <tlen> indica la longitud máxima permitida para el texto asociado a ese número (nombre del contacto).
- Respuesta: +CPBR: (1-150),48,14

2) AT+CPBR=<indice>
- Leer una entrada de la agenda de contactos.
- Sintaxis: AT+CPBR=<indice inicial> [,<indice final>] | +CPBR: <índice>, <número>, <tipo>, <texto>
  <índice> indica el índice de la agenda de contactos.
  <número> indica el número de teléfono almacenado en el índice.
  <tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.

<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
- Respuesta a AT+CPBR=8: +CPBR: 8,"646123456",129,"Gospel"

Nota: Para leer todas las entradas de la agenda, basta con preguntar por el tamaño de la agenda, almacenarlo en una variable int PhoneBookSize y lanzar un bucle FOR preguntando por cada índice:

```
for (int n = 1; n <= PhoneBookSize; n++)
{
   Enviar("AT+CPBR=%d", n);
}
```

Combinación de AT+CPBS;+CPBR [Leer una entrada de una lista de contactos seleccionada]

- Primero elegimos la lista de contactos a la que queremos acceder, y luego leemos una entrada por su índice.

Sintaxis: AT+CPBS="XX";+CPBR=<índice>, donde "XX" se sustituye por el dispositivo de almacenamiento:

"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]
"TA" - TA phonebook list [Lista de contactos del terminal]
"LD" - SIM last dialing list [Lista de números marcados]
"DC" - Dialled call list [Lista de llamadas realizadas]
"RC" - ME received calls list [Lista de llamadas recibidas]
"MC" - ME missed call list [Lista de llamadas perdidas]
"EN" - Emergency number list [Lista de números de emergencia]
"FD" - SIM fix dialing list
"MT" - ME + SIM conbined list
"ON" - SIM o ME own number list

- Ejemplo de Respuesta a AT+CPBS="DC";+CPBR=2: +CPBR: 1,"646123456",129,"Pepito" (Visualizamos el último contacto al que hemos llamado).
AT+CPBS="MC";+CPBR=1: +CPBR: 1,"646987654",129,"Jaimito" (Visualizamos la última llamada perdida).

AT+CPBF: [Find Phone Book Entries]

- Devuelve la entrada de la agenda de contactos cuyo texto asociado a un número contiene la cadena alfanumérica proporcionada.
- Sintaxis: AT+CPBF="textoaencontrar" | +CPBR: <índice>, <número>, <tipo>, <texto>
"textoaencontrar" es case-sensitive, así que cuidado con el uso de mays.
<índice> indica el índice de la agenda de contactos.
<número> indica el número de teléfono almacenado en el índice.
<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.
<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
- Respuesta a AT+CPBF="Pepito": +CPBF: 19, "646987654",129,"Pepito"

AT+CPBW: [Write Phone Book Entry]

- Escribe una entrada en la agenda de contactos.
- Sintaxis: AT+CPBW = <índice>, <número>, <tipo>, <texto>
<índice> indica el índice de la agenda de contactos donde se creará la entrada de contacto. Si no se proporciona índice, se añade la entrada en el primer hueco libre.
<número> indica el número de teléfono almacenado en el índice.

<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.

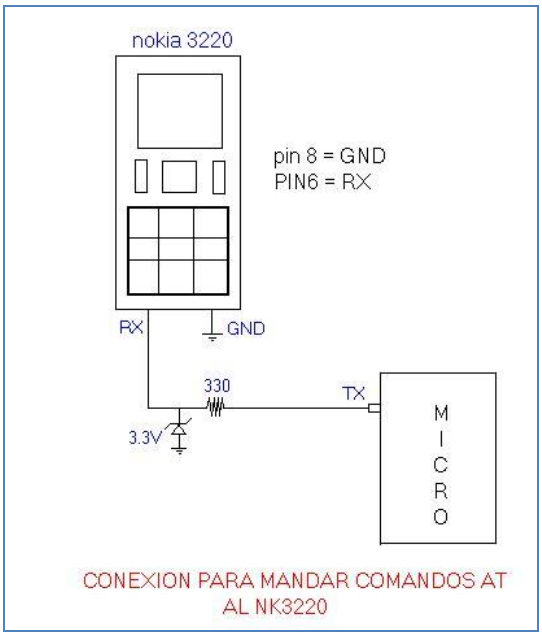
<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.
Nota: Si únicamente se proporciona el campo del índice (omitiendo el resto de campos), la entrada de la agenda asociada a ese índice se borrará.
- Ejemplo para crear un nuevo contacto: AT+CPBW=,"696224466",129,"Jaimito"

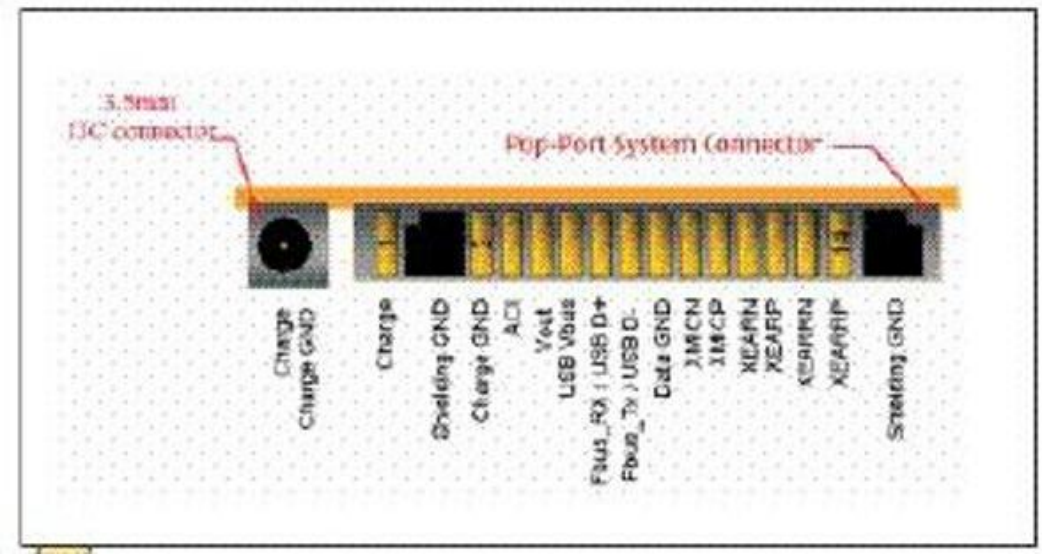
RESUMEN COMANDOS AT UTILIZADOS

```
// Cuelga   --> "AT+CHUP"
// Call ID  --> "AT+CLIP=1"
// Contesta --> "ATA"
// Respuesta corta o numérica --> "ATV[0]"  0 = OK, 2 = RING y 4 = ERROR.
// Respuesta larga o de texto (Default) --> "ATV1"
// Autocontestar --> "ATS0=X" donde X es el número de timbrazos para contestar
la llamada.
// Enviar DTMF --> "AT+VTS=X" donde X es el número o símbolo del cuál queremos
enviar el DTMF.
```

PUERTO DE COMUNICACIÓN DEL CELULAR NOKIA 3220

Las siguientes gráficas representa la conexión entre el modulo celular y el puerto Usart del Microcontrolador PIC16F88, y la codificación del los pines del puerto serie del dispositivo celular, respectivamente.





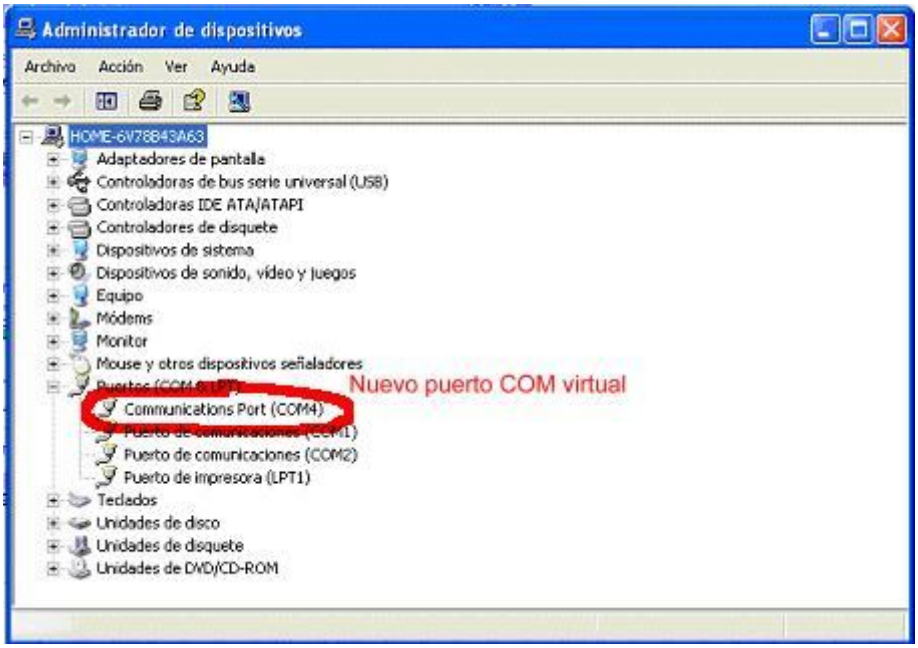
INTERFAZ RS232

El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa. Lo interesante es que sólo necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. Otros integrados que manejan las líneas RS232 requieren dos voltajes, +12V y -12V.
El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento en base a señales de nivel TTL/CMOS.
El circuito integrado posee dos conversores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL.
Estos conversores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS.
TX es la señal de transmisión de datos, RX es la de recepción, y RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.
El protocolo de transferencia usado entre la interfaz RS232 y el microcontrolador o microprocesador es el siguiente:
Ocho Bits de datos.
Un bit de parada.
Sin paridad.
Sin control de flujo.
El único hardware adicional que requiere la interfaz USB232 es el siguiente:

- Un cristal de 4Mhz
- Dos capacitores de 20pf
- UN capacitor 104.

Opcionalmente se pueden conectar dos Leds indicadores de actividad de la transmisión en los pines RX LED y TX LED.
La conexión con el PC se hace a través de una interfaz USB 2.0 a Full Speed 12Mb/s sin ninguna circuitería adicional, basta con conectar los pines D+ y D- a los correspondientes del Hub de PC.
Tan pronto la interfaz USB232 se conecta la PC por primera vez, Windows Xp detecta automáticamente el nuevo hardware e instala los drivers necesarios desde el CD que se incluye con la interfaz.

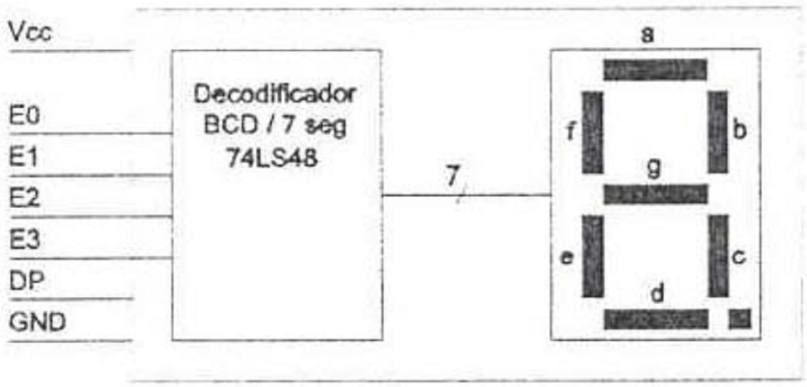
Tras la instalación de los driver de la interfaz en Windows XP, se crea un nuevo puerto COM Virtual, el cual se abre y opera para efectos de programación como cualquier otro puerto COM ordinario.



Este puerto COM Virtual tiene capacidad para transferir datos a 9600, 19200, 38400, 57600 y 115200 bps entre el PC y el puerto serie RS232 del dispositivo que se está acoplando por medio de la interfaz USB232.

DECODIFICADOR BCD/7 SEGMENTOS

Para visualizar un valor codificado con 4 bits se utiliza un display de 7 segmentos formados por diodos LEDs y un circuito decodificador BCD (4 bits) a 7 segmentos (circuito integrado 74LS48).

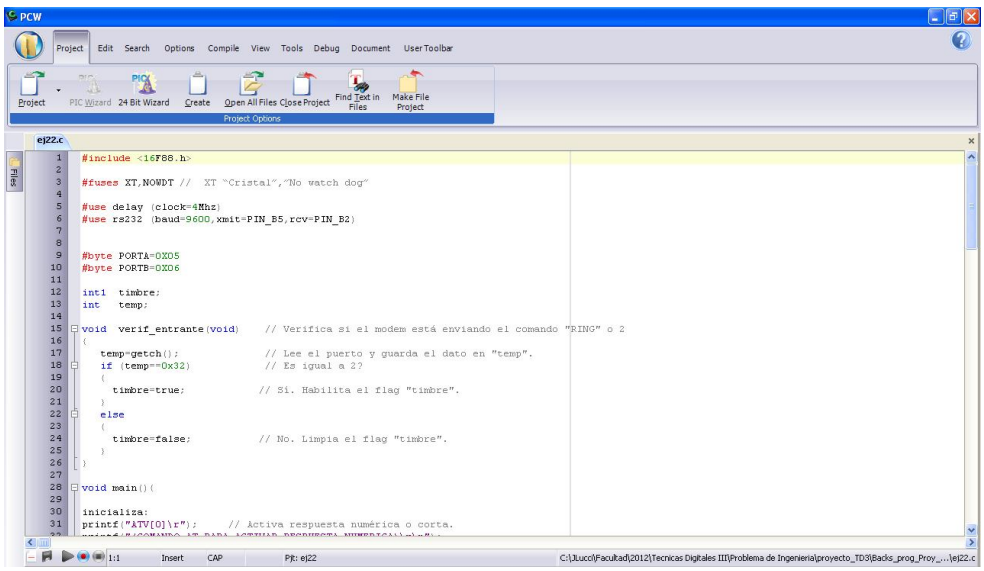


La tecnología empleada es TTL, por lo que se deberá alimentar a +5V en el terminal marcado como Vcc y a masa en el terminal marcado con GND. La lógica es positiva (+5V = 1 lógico; 0V = 0 lógico). Las 7 salidas del decodificador se conectan al segmento del display correspondiente. El encendido o no de cada uno de los segmentos del display depende de la tabla de verdad del 74LS48 que se puede consultar a continuación. La entrada DP permite el control del encendido o no del punto decimal del display.

Decimal	ENTRADAS				SALIDAS						
D	E3	E2	E1	E0	a	b	c	d	e	f	g
0	L	L	L	L	H	H	H	H	H	H	L
1	L	L	L	H	L	H	H	L	L	L	L
2	L	L	H	L	H	H	L	H	H	L	H
3	L	L	H	H	H	H	H	H	L	L	H
4	L	H	L	L	L	H	H	L	L	H	H
5	L	H	L	H	H	L	H	H	L	H	H
6	L	H	H	L	L	L	H	H	H	H	H
7	L	H	H	H	H	H	H	L	L	L	L
8	H	L	L	L	H	H	H	H	H	H	H
9	H	L	L	H	H	H	H	L	L	H	H
10	H	L	H	L	L	L	L	H	H	L	H
11	H	L	H	H	L	L	H	H	L	L	H
12	H	H	L	L	L	H	L	L	L	H	H
13	H	H	L	H	H	L	L	H	L	H	H
14	H	H	H	L	L	L	L	H	H	H	H
15	H	H	H	H	L	L	L	L	L	L	L

PROGRAMACION PIC

CCS COMPILER es el programa que utilizamos para crear nuestro proyecto.
El ambiente de desarrollo integrado de PCWHD le da a los programadores la capacidad de producir rápidamente códigos muy eficientes usando un lenguaje fácil y manejable, pero de alto nivel. Nos basamos en la programación en C para desarrollar nuestro proyecto. Básicamente programamos en este lenguaje y mediante este programa pudimos compilar, simular y convertir el código a lenguaje de bajo nivel comprensible para el pic (assembly)

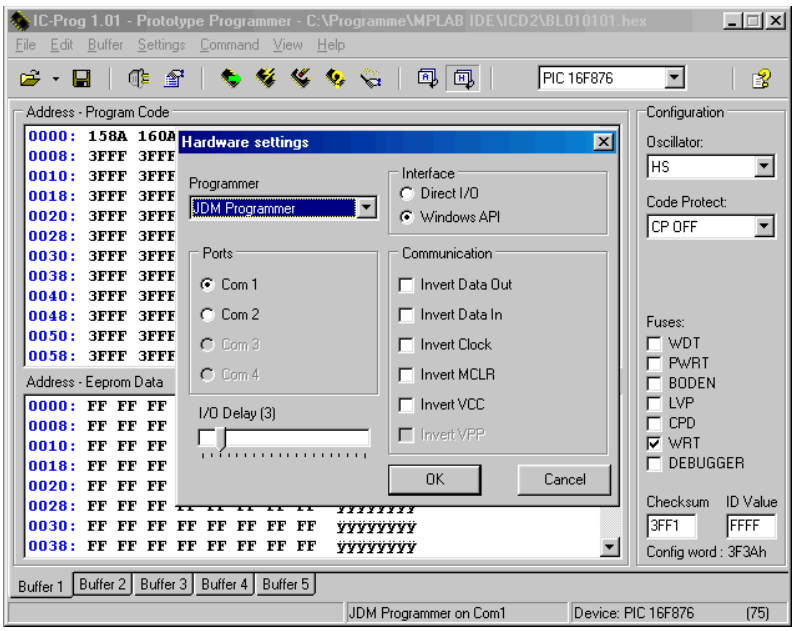


DTMF DIAL

Es un programa diseñado para emitir los tonos DTMF, de la tarjeta de sonido del PC. Lo hemos utilizado para emitir tonos y utilizarlos en la lógica de nuestro programa simulando la conexión con el equipo telefónico

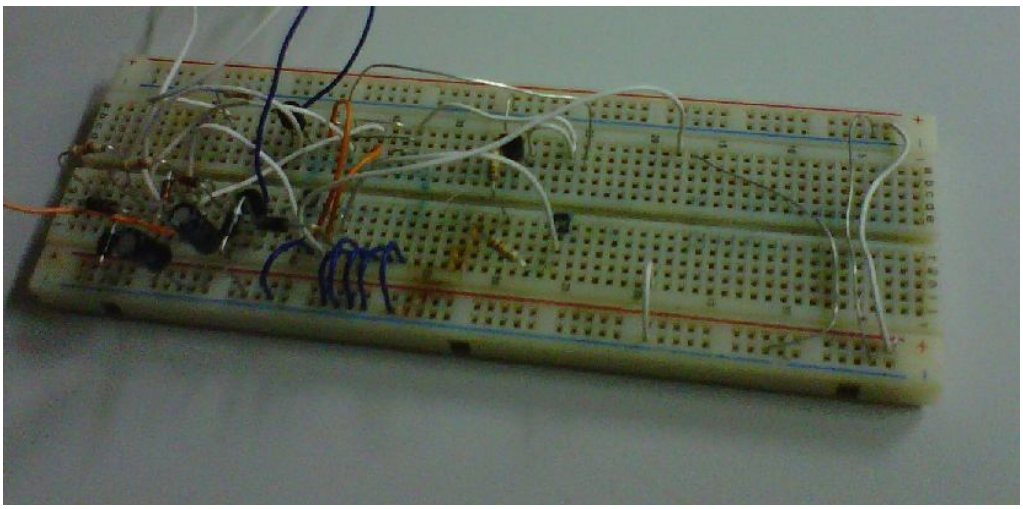
COMPILADOR ICPROG

Fue utilizado para cargar los archives en Assembly al PIC. Se puede descargar gratis en diferentes webs y posee gran amplitud de posibilidades de trabajar con diferentes pics, así como las diferentes memorias con las que también trabaja.

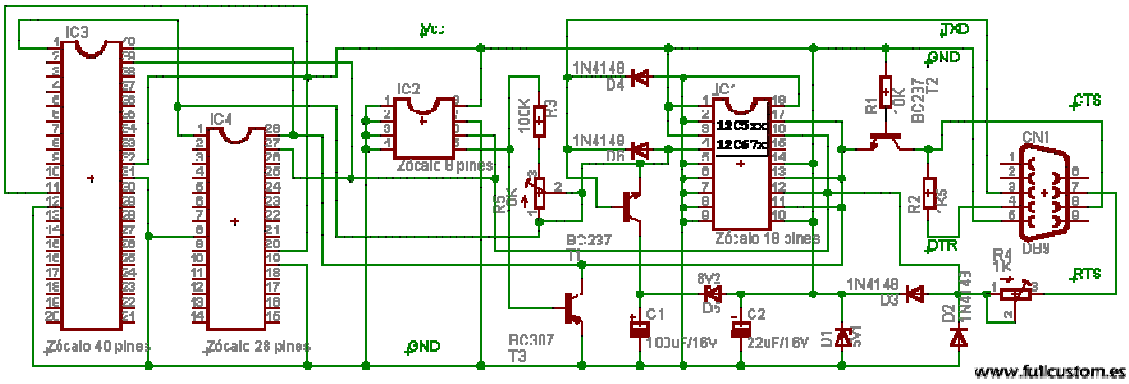


PROGRAMADOR JDM TE20SE

Se programa por el puerto serie. Se pueden programar dispositivos pics, memorias, etc.



ESQUEMATICO



PUERTO SERIE

DEFINICION

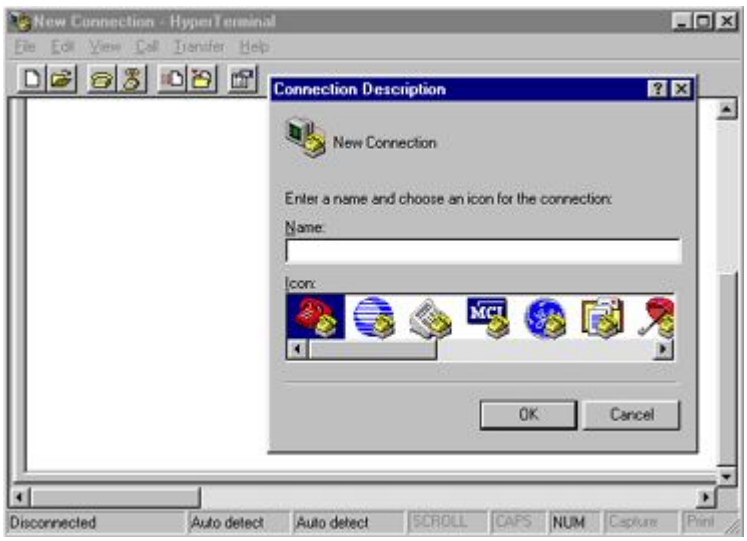
Un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente. La comparación entre la transmisión en serie y en paralelo se puede explicar usando una analogía con las carreteras. Una carretera tradicional de un sólo carril por sentido sería como la transmisión en serie y una autovía con varios carriles por sentido sería la transmisión en paralelo, siendo los vehículos los bits que circulan por el cable.

DB9

El siguiente diagrama corresponde al conector que utilizamos para interconectar el modulo GSM con la pc para realizar monitoreo en tiempo real.

HYPERTERMINAL

Es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, servicios en línea y equipos host, mediante un módem, un cable de módem nulo o Ethernet. Nosotros lo hemos utilizado para realizar pruebas previas al prototipo final, nuestra comunicación se baso en la utilización del Puerto serie de la pc. Se realizaron pruebas de comunicación entre la pc y el microcontrolador (pic 16f88, también hemos probado comunicación entre la pc y el teléfono celular (nokia 3220)



NOKIA PCSUITE

Programa utilizado para poder desbloquear el puerto serie del dispositivo celular.
Para poder habilitar el "pop-port" debemos conectar previamente el dispositivo celular a la PC. Esto es necesario debido a que el Programa PCSuite, propio de NOKIA envía una trama de hexadecimales al puerto para habilitarlo. Una vez que existe conexión entre el celular y la PC mediante el cable DKU-5 el puerto del celular se encuentra habilitado para ser utilizado hasta que se des-alimente o se agote la batería. Por eso una buena mejora al sistema seria construir una fuente de alimentación que cargue la batería del celular cuando la carga de esta descienda de cierto nivel (cargador automático de batería).

INICIO DE LAS APLICACIONES DE NOKIA PC SUITE

Las aplicaciones de Nokia PC Suite se pueden abrir de dos formas: en el menú Inicio: haga clic en Inicio, señale a Programas y a Nokia PC Suite 5 y finalmente seleccione la aplicación que desee en Nokia Phone Browser: seleccione Nokia Phone Browser en la vista del árbol del Explorador de Windows y haga doble clic en el icono de la aplicación que desee abrir en la vista de carpetas.
¿Cómo logro que un teléfono Nokia funcione en Nokia Connection Manager con el cable DKU-5 (serie)?
Condiciones previas: dispone de un teléfono compatible Nokia, un cable DKU-5 (serie), un sistema operativo de Windows compatible, los controladores correctos para el cable DKU-5 (serie) y un Nokia PC Suite compatible sin instalar.
1) Instale los controladores de DKU-5 en el equipo.
2) Instale Nokia PC Suite siguiendo las instrucciones.
3) Conecte el cable. Windows debe reconocer el cable y buscar los controladores.
Seleccione Sí para contestar a la pregunta sobre la firma digital si la acepta y se instalarán los controladores.
4) Inicie Nokia Connection Manager (si no está abierto).
5) Seleccione la conexión de puerto serie.
6) Seleccione el puerto COM correcto de la lista que se muestra al pulsar el botón Avanzadas en Nokia Connection Manager.
7) Conecte el cable al teléfono.
El teléfono se mostrará en la lista de teléfonos de Nokia Connection Manager.

Para la interconexión del dispositivo celular y la PC se utilizó el Cable DKU-5.

CIRCUITO UTILIZADO

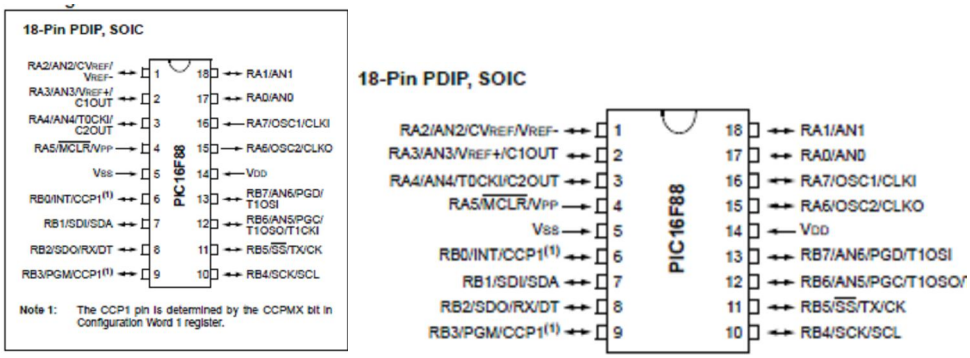
PIC16F88

CARACTERISTICAS PRINCIPALES

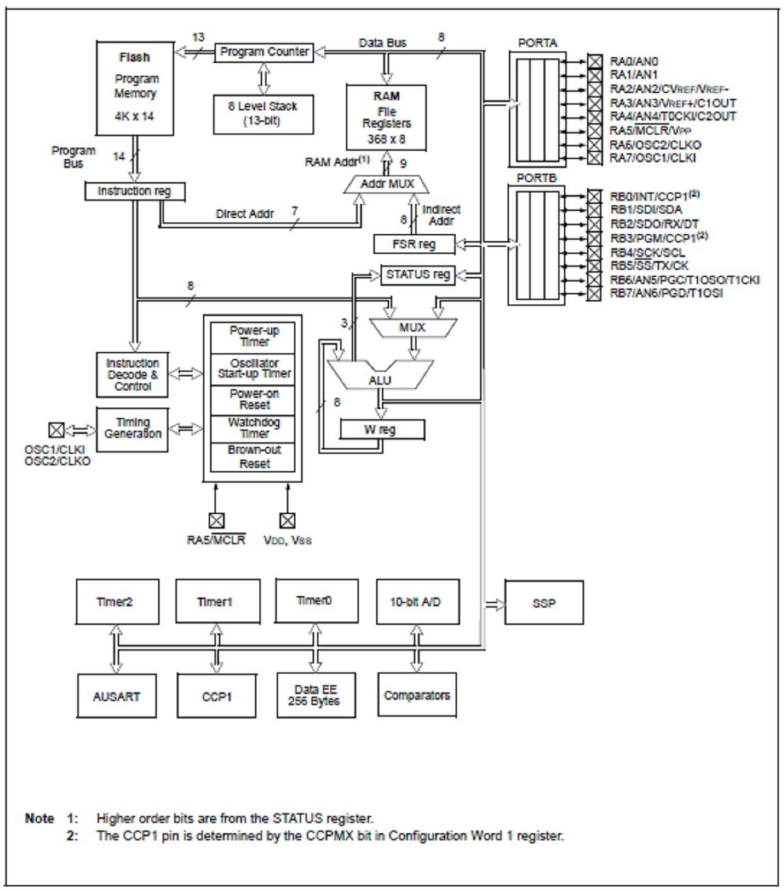
• Capture, Compare, PWM (CCP) module:
- Capture is 16-bit, max. resolution is 12.5 ns
- Compare is 16-bit, max. resolution is 200 ns
- PWM max. resolution is 10-bit
• 10-bit, 7-channel Analog-to-Digital Converter
• Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I2C™ (Slave)

• 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

DISTRIBUCIÓN DE PINES



ARQUITECTURA INTERNA

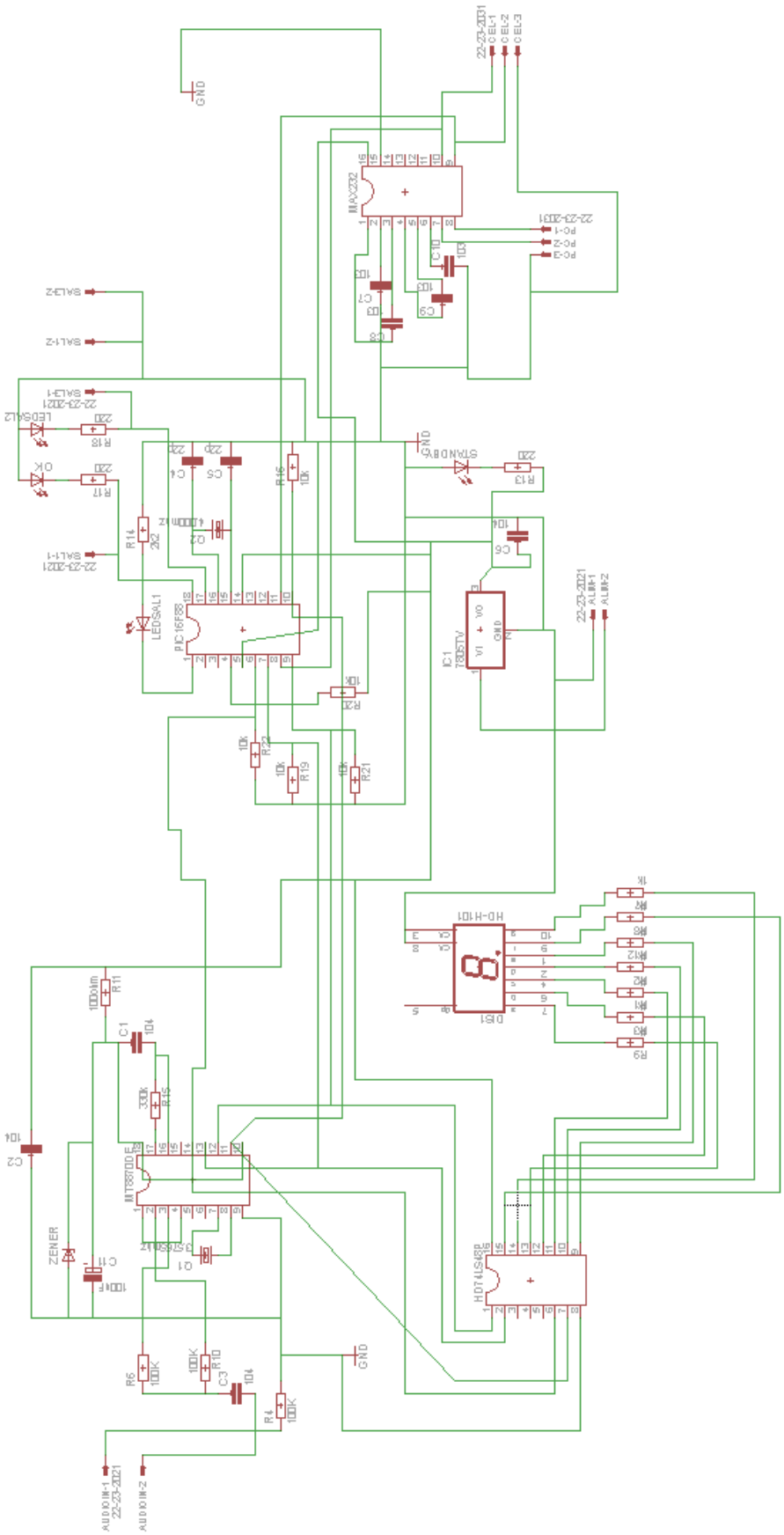


ASIGNACION DE PINES A NUESTRA APLICACIÓN

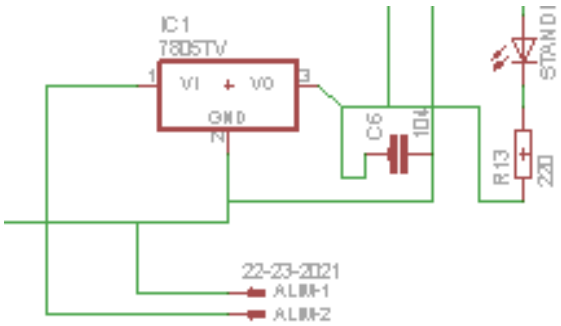
A continuación se describe la asignación de pines del PIC 16F88 para nuestra aplicación. Los BITS 1-4 son los pines que llegan del Decodificador DTMF para reconocer que numero es pulsado. La codificación se detalla con profundidad en la parte del decodificador DTMF.

PIC16F88	
PIN	**DESCRIPCION**
1	Salida 1
2	MCLR
3	GND
4	-
5	-
6	BIT1
7	BIT2
8	TX
9	BIT3
10	BIT4
11	RX
12	-
13	-
14	+5V
15	CRYSTAL1
16	CRISTAL2
17	Salida 2
18	LED OK

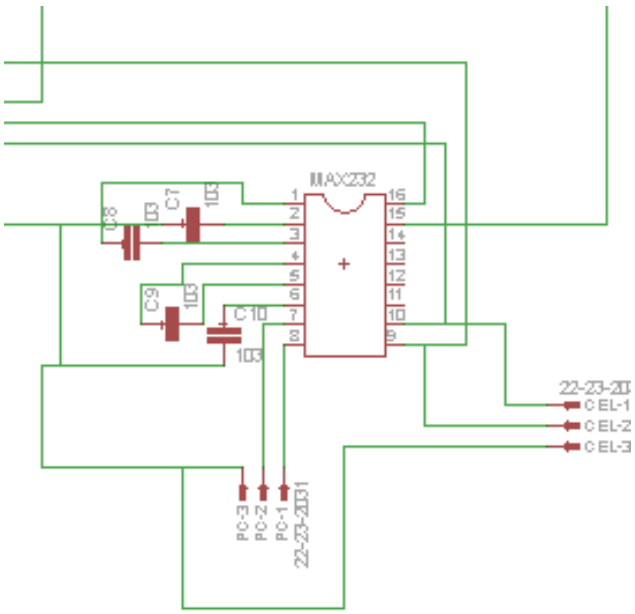
DIAGRAMAS



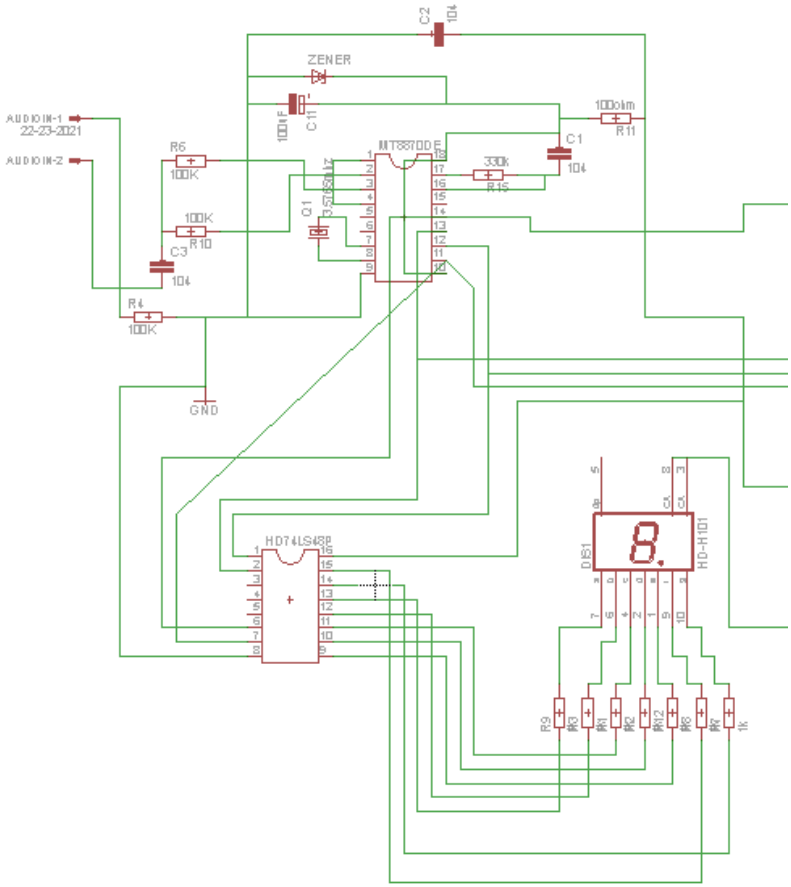
MODULO ALIMENTACION



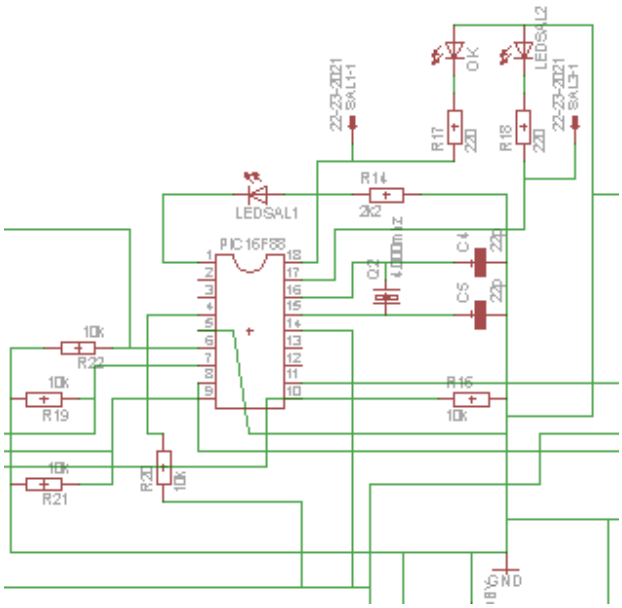
MODULO COMUNICACIÓN RS232



MODULO DECODIFICACION TONOS DTMF

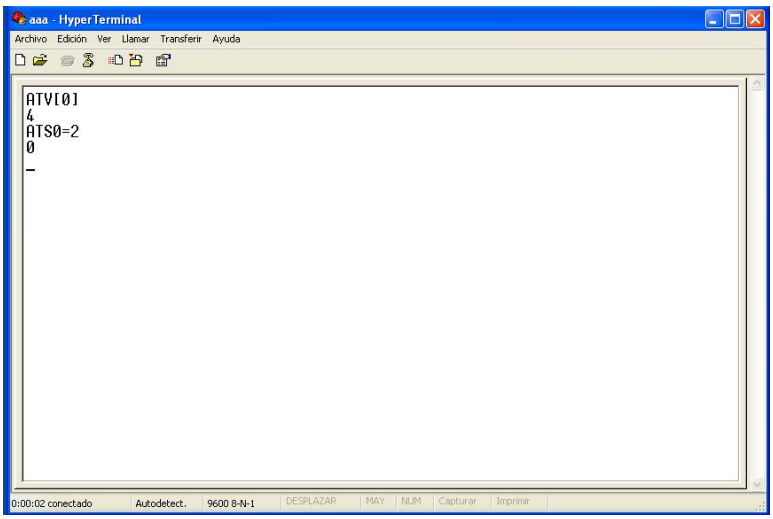


MODULO MICROCONTROLADOR PIC 16F88

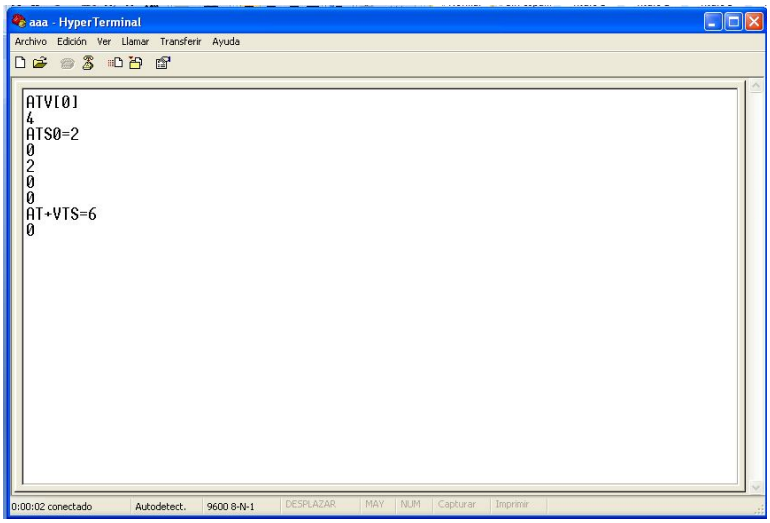


MONITOREO EN PC MEDIANTE HYPERTERMINAL

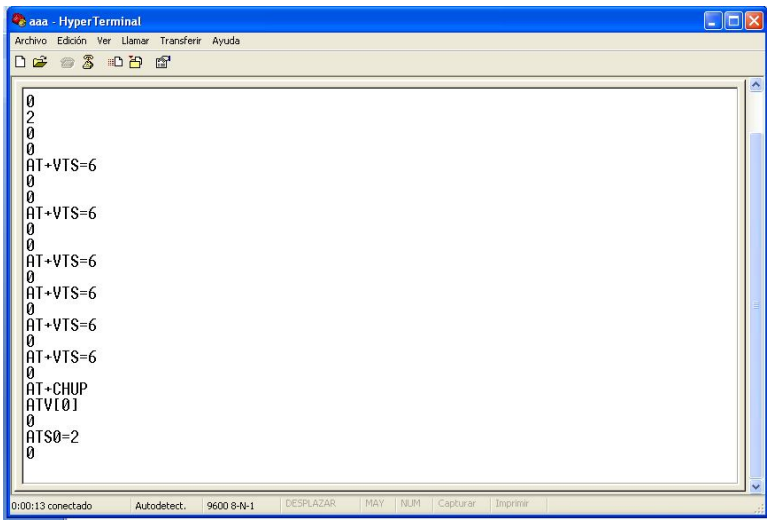
Para corroborar el funcionamiento del sistema, conectamos mediante el adaptador serie-usb nuestro sistema con la pc. El integrado MAX232 convierte las señales TTL que emite el PIC a señales RS232 que recibe la PC.
En la siguiente captura observamos el arranque del sistema. El comando ATV[0] configura la respuesta a los comandos AT emitidos por el celular de forma numérica (Se opta por esta modalidad porque es más fácil trabajar con números que con palabras en la toma de decisiones dentro del programa), seguidamente el sistema responde con un 4, que significa que la comunicación es correcta. Después aparece el comando ATS0=2. Este último comando indica que el sistema debe descolgar después de 2 timbrazos. A partir de aquí el sistema se encuentra en espera de la activación de alguna de las salidas.



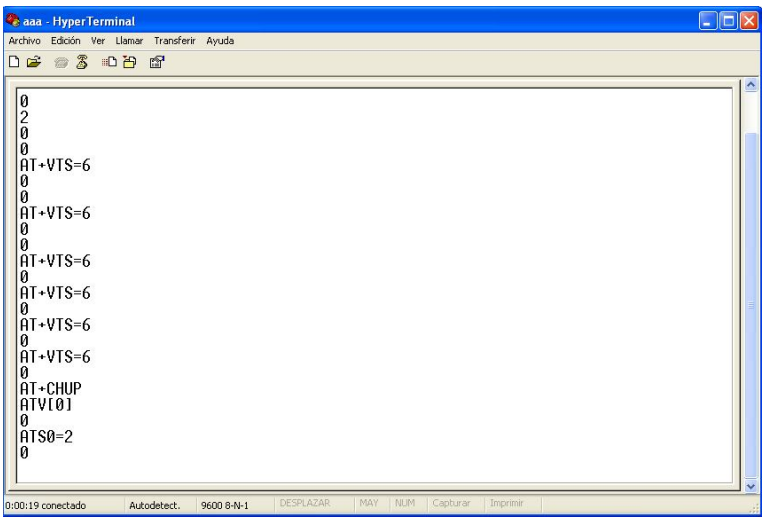
En la siguiente captura se observa que el sistema emite un sonido (AT+VTS=6), confirmando que alguna de las salidas fue activada. Esto es necesario para poder tener un feedback con el sistema, es decir, asegurarnos que el sistema responde correctamente a lo que le pedimos.



En la siguiente pantalla se observa que el tono es emitido 3 veces, señal que nos indica que el sistema está por colgar. Después aparece el comando AT+CHUP. Dicho comando es el encargado de colgar la llamada.

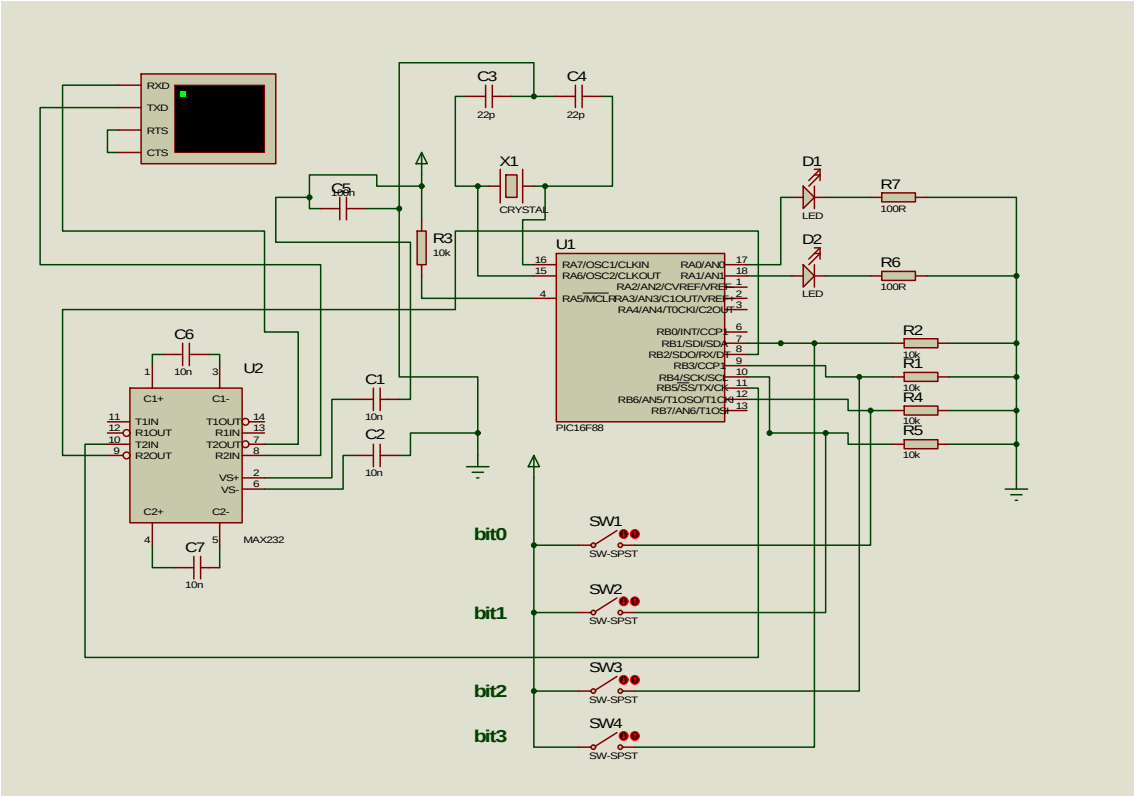


Y finalmente, se observa como el sistema se reinicia, quedando en estado de espera para recibir alguna orden.



SIMULACION EN PROTEUS

Para probar el funcionamiento del prototipo a diseñar realizamos una simulación en el PROTEUS ISIS 7 PROFESSIONAL. A continuación mostramos la captura de pantalla del circuito de simulación.



A continuación se muestra la pantalla de la simulación en tiempo real.

VERSION 1.30 SIMULACION DE PRESENCIA - TDII - 2013
FURCH.LUCCI.BERTHET
ATV[0]
(COMANDO AT PARA ACTIVAR RESPUESTA NUMERICA)
esperando llamada ENTRANTE.....

FUNCIONAMIENTO

La secuencia arranca con la llamada a la central, esta al recibir una llamada manda la señal ring al pic y este le indica que atienda (comando ata). Una vez establecida la comunicación, la central queda a la escucha de los botones que apretemos, cuando apretamos un botón en el teléfono este es recibido en la central como un tono, dicho tono suena en el auricular de la central y es enviado a un integrado que convierte el tono en el numero que apretamos (DTMF), de esta manera el pic interpreta que queremos que el haga en función al número que le enviamos.
Para terminar la comunicación se presiona el numero 40.
Mejoras que hemos hecho: para saber que el sistema está respondiendo hemos implementado un sistema de tonos que nos indica que la central acata la orden enviada. Cuando le damos la orden prender ella responde con un bip, cuando le damos la orden apagar ella responde con dos bip y al terminar la comunicación nos responde con tres bip.

NUMERO	ACCION
1	Prende led 1
2	Apaga led 1
3	Prende led 2

5	Apaga led 2
40	La central cuelga

CONCLUSIONES

Después de haber trabajado un buen tiempo en equipo, podemos asegurar que el trabajo realizado de esa forma es mucho más efectivo ya que las ideas de una persona se complementan con la de los demás integrantes. En lo que respecta al prototipo fabricado podemos decir que es un sistema muy versátil ya que el mismo se puede implementar como control remoto para tareas del hogar o también tareas industriales, por ejemplo, se conecta el sistema a una maquina en una fábrica, y a distancia, es decir lejos del lugar podemos controlar la misma. Además es un sistema que se puede mejorar muchísimo agregando salidas. El desarrollo del proyecto nos hizo aprender lo que respecta a comunicación serie, comandos AT (norma exclusiva de los dispositivos GSM para intercomunicarse), programación en lenguaje C y demás.

ANEXOS:

LISTADOS DE PROGRAMAS

El siguiente código es el programa principal en lenguje C de nuestro proyecto.

```
//CON 1 ENCIENDE LED A1
//CON 2 SE APAGA SALIDA A1
//CON 3 ENCIENDE LED A2
//CON 5 SE APAGA SALIDA A2
//CON 40 SE CORTA LA LLAMADA
//PARA CONFIRMAR CARGA ENCENDIDA MANDA TONO 6
//PARA CONFIRMAR CARGA APAGADA MANDA TONO 6 DOS VECES
//PARA CONFIRMAR FIN LLAMADA MANDA TONO 6 TRES VECES
//COMANDO CONTESTA: ATA
//COMANDO CUELGA AT+CHUP

//EJ41.C

#include <16F88.h>

#fuses XT,NOWDT //  XT “Cristal”,”No watch dog”

#use delay (clock=4Mhz)
#use rs232 (baud=9600,xmit=PIN_B5,rcv=PIN_B2)


#byte PORTA=0X05
#byte PORTB=0X06

void  un_tono(void)        // Función que envía un tono.
{
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(500);
}

void  dos_tonos(void)      // Función que envía dos tonos.
{
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(500);
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(500);
}

void  tres_tonos(void)     // Función que envía tres tonos.
{
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(500);
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(500);
   printf("AT+VTS=6\r");   // Comando AT para generar el DTMF de "6"
   delay_ms(2000);
}



void main(){
```

```
//printf("VERSION 1.30 SIMULACION DE PRESENCIA - TDII - 2013\r\n");
//printf("FURCH.LUCCI\r\n\r\n");
inicializa:
printf("ATV[0]\r");     // Activa respuesta numérica o corta.
output_high(pin_a0);  // encender intermitente RA1 para saber que el sistema
esta standby
delay_ms(2000);
printf("ATS0=2\r");
delay_ms(5000);
output_low(pin_a0);

prender:
if
(bit_test(PORTB,0)==0&&bit_test(PORTB,1)==0&&bit_test(PORTB,3)==0&&bit_test(PORT
B,4)==1) //se presiono el 1
  {
   output_high(pin_a1); // encender RA1
   delay_ms(100);
   un_tono();
//   printf("SE HA INGRESADO EL NUMERO 1 Y SE ACTIVO LA SALIDA\r\n");
//   printf("PRESIONE EL NUMERO 2 PARA DESACTIVARLA\r\n\r\n");
  }

else if
(bit_test(PORTB,0)==0&&bit_test(PORTB,1)==0&&bit_test(PORTB,3)==1&&bit_test(PORT
B,4)==0) //se presiono el 2
   {
    output_low(pin_a1); // apagar RA1
    delay_ms(100);
    dos_tonos();
//       printf("SE HA INGRESADO EL NUMERO 2 Y SE DESACTIVO LA SALIDA\r\n");
//       printf("PULSE 1 PARA ACTIVAR, 3 PARA LA INTERMIT o 45 PARA
COLGAR\r\n\r\n");
   }


else if
(bit_test(PORTB,0)==0&&bit_test(PORTB,1)==0&&bit_test(PORTB,3)==1&&bit_test(PORT
B,4)==1) //se presiono el 3
  {
   output_high(pin_a2); // encender RA2
   delay_ms(100);
   un_tono();
//   printf("SE HA INGRESADO EL NUMERO 3 Y SE ACTIVO LA SALIDA\r\n");
//   printf("PRESIONE EL NUMERO 2 PARA DESACTIVARLA\r\n\r\n");
  }

else if
(bit_test(PORTB,0)==0&&bit_test(PORTB,1)==1&&bit_test(PORTB,3)==0&&bit_test(PORT
B,4)==1) //se presiono el 5
   {
    output_low(pin_a2); // apagar RA2
    delay_ms(100);
    dos_tonos();
//       printf("SE HA INGRESADO EL NUMERO 5 Y SE DESACTIVO LA SALIDA 2\r\n");
   }

else if
(bit_test(PORTB,0)==0&&bit_test(PORTB,1)==1&&bit_test(PORTB,3)==0&&bit_test(PORT
B,4)==0) //condicion de PRESIONAR el 40 para finalizar llamada
```

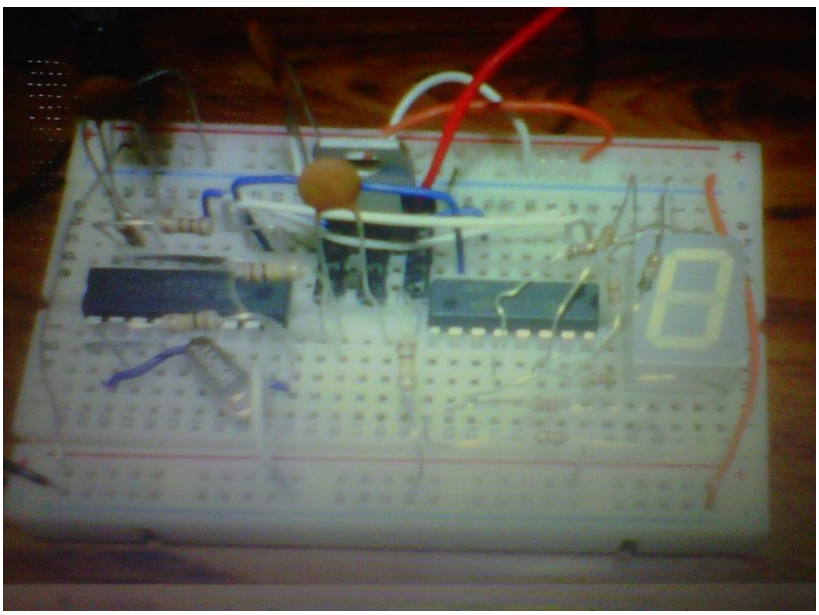
```
   {
    delay_ms(2000);
    tres_tonos();
    printf ("AT+CHUP\r");   // comando AT para Colgar
//    printf("(COMANDO AT PARA colgar llamada)\r\n\r\n");
//    printf("\r\n");
//    printf("\r\n");
//    printf("\r\n");
    goto inicializa; //reinicia el sistema
    }

goto prender;

}
```

Como observamos en el programa anterior, la versión de corrección es la nro. 41. Es decir tuvimos 40 revisiones del programa antes de llegar al definitivo.

FOTOS DEL PROTOTIPO

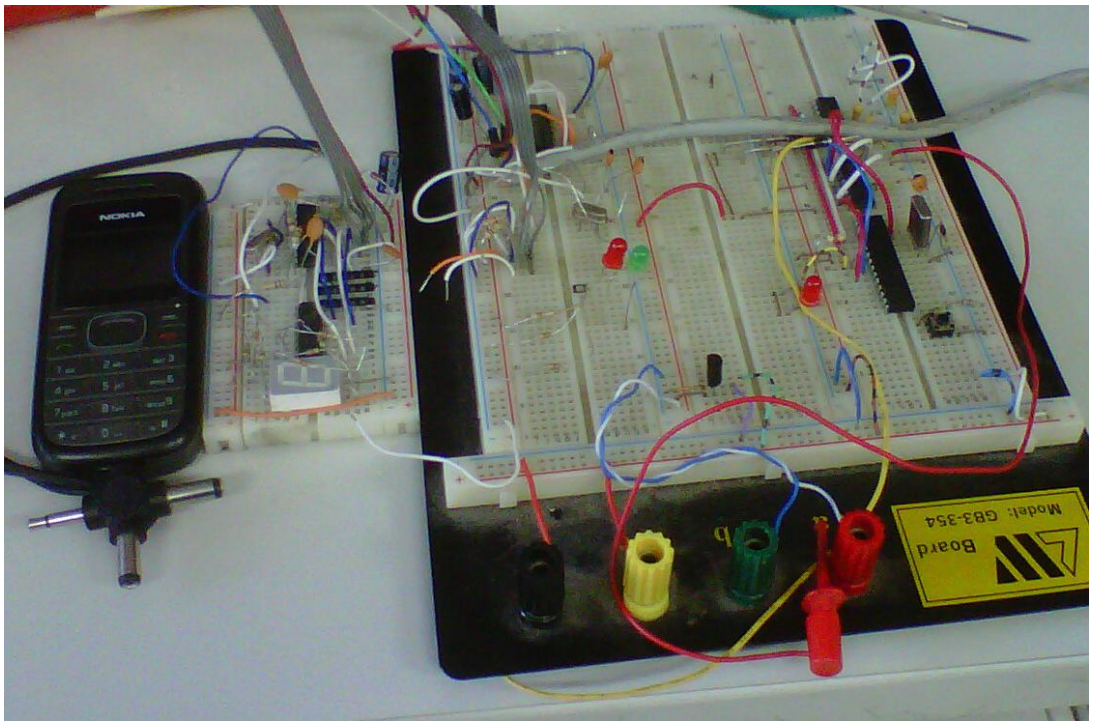
DETECTOR DE DTMF

El agregado del display de 7 segmentos es para una rápida y sencilla visualización del numero que el detector toma.



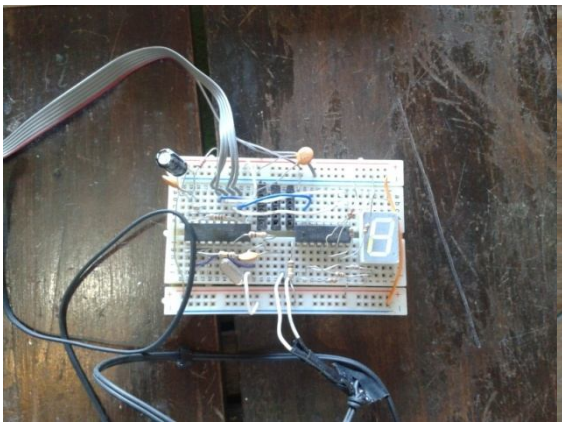
PIN OUT NOKIA 3220

PRIMERA PLAQUETA GENERAL versión 1.0 (sin utilización de puerto serie del Celular)



PLAQUETA DEFINITIVA

Módulo detector DTMF

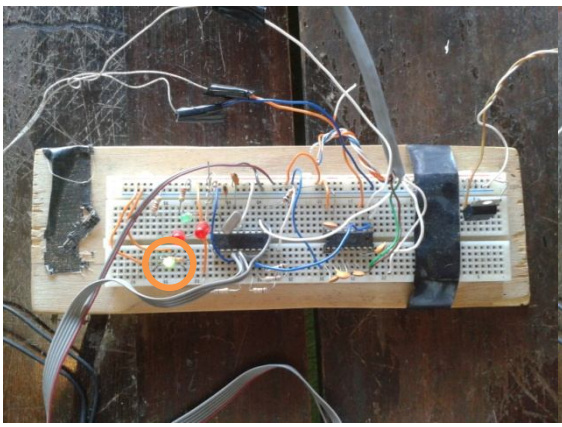


Modulo Celular (Nokia 3220)

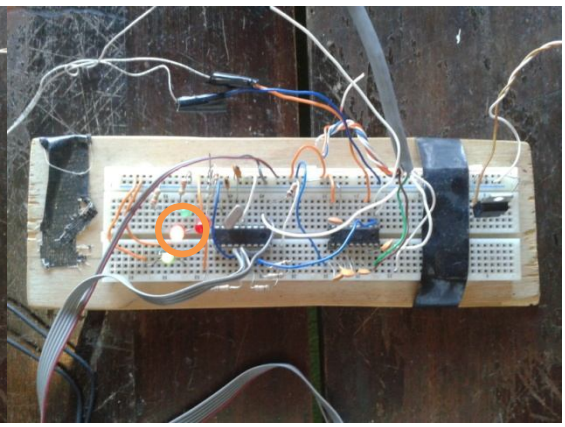
Conversor USB-Serie

Fuente de Alimentación

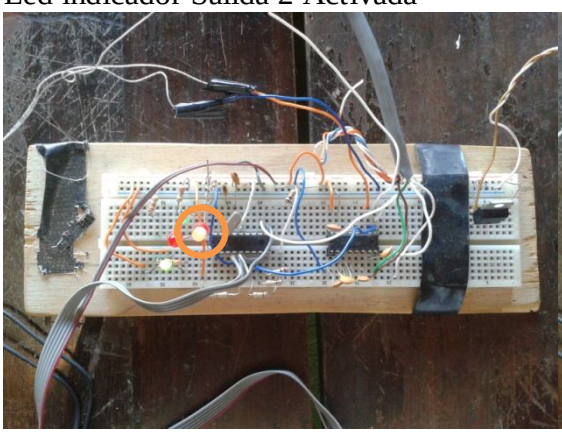
Led Indicador de Sistema Encendido



Led indicador Salida 1 Activada



Led indicador Salida 2 Activada



Led indicador Reinicio del Sistema