UCT

UNIDAD CONTROLADA TELEFÓNICAMENTE

Boris Estudiez
(C) slicetex electronixs

0.1: Versiones del Proyecto:

Nombre de proyecto:	**UCT** **U**nidad **C**ontrolada **T**elefónicamente.
Fecha de inicio	**12/09/2003**
Ultima Actualización	**06/03/2004**
UCT DOCUMENT **V**ERSION	**DV-1.0**
UCT HARDWARE **V**ERSION	**HV-0.6**
UCT SOFTWARE **V**ERSION	**SV-0.5.1**
UCT MESSAGES **V**ERSION	**MV-1.0**

0.2: Autor:

Autor: Boris Estudiez

E-mails:

- stk@freeshell.org
- slicetex@hotpop.com
- 43824@electronica.frc.utn.edu.ar

Web:

- http://stk.freeshell.org (pagina oficial del proyecto)
- http://geocities.com/slicesoft (mirror)

0.3: Historia:

El proyecto UCT fue presentado como practico final para regularizar la materia Técnicas Digitales II del nivel IV de la carrera Ingeniería Electrónica de la Universidad Tecnológica Nacional, Facultad Regional Córdoba.

0.4: Notas Sobre el Documento:

El presente informe fue escrito suponiendo conocimientos avanzados de electrónica por parte del lector, por lo tanto por simplificación se omiten muchas demostraciones y explicaciones que deben ser buscadas en libros u hojas técnicas de los componentes electrónicos.
La parte de software no se explica en este documento ya que es demasiada extensa y esta abierta a modificaciones continuas. Usted debe bajar de la pagina del proyecto el software necesario para que la parte de hardware del UCT funcione.
Es probable que por cuestiones de tiempo el documento no sea actualizado frente a cambios de los circuitos electrónicos (hardware), software y mensajes de audio del proyecto.
Al momento de escribir el documento las versiones de hardware, software y mensaje de audio, eran las listadas en la sección **0.1** del documento.
El proyecto fue probado exitosamente en las líneas telefónicas de la compañía telefónica **Telecom de Córdoba-Argentina**, si usted vive en otras partes del mundo quizás deba hacer pequeñas modificaciones en los circuitos del proyecto debido a que las **normas telefónicas pueden ser distintas**. Sin embargo el proyecto fue pensado para funcionar en casi cualquier condición.

0.5: Índice:

El documento se divide en las siguientes secciones:

1.0: Introducción:

El proyecto **UCT** (**U**nidad **C**ontrolada **T**elefónicamente) fue desarrollado con el objetivo de poder **comandar** equipos, maquinas, computadoras, electrodomésticos, luces y cualquier otro tipo de dispositivo de forma remota, por medio de un teléfono conectado a la línea telefónica.

El UCT no solo se limita a recibir ordenes remotas de un usuario a través de la línea telefónica, sino que también puede responder e informar con mensajes de **audio (voz)**, lo que permite que el usuario pueda seguir fácilmente las acciones que se están llevando a cabo en el UCT e interactuar sencillamente.

Finalmente y quizás unas de las características mas importantes del UCT, es su **puerto de comunicaciones** denominado **BUS DCE**, el cual permite una comunicación bidireccional entre el UCT y otros dispositivos externos a través de señales digitales. Esta ultima utilidad permite un sin fin de aplicaciones (limitada solo por la imaginación del usuario), y posibilita que el UCT sea expandido a través de módulos (circuitos electrónicos externos al UCT), añadiendo funcionalidades extras al aparato.

La posibilidad de expandir mediante módulos al UCT puede lograr aplicaciones complejas, como por ejemplo operar computadoras remotamente a través del teléfono o informar de la activación de una alarma a un numero telefónico predeterminado por el usuario.

2.0: Desarrollo del proyecto:

2.1: Vista superior de los componentes electrónicos del circuito

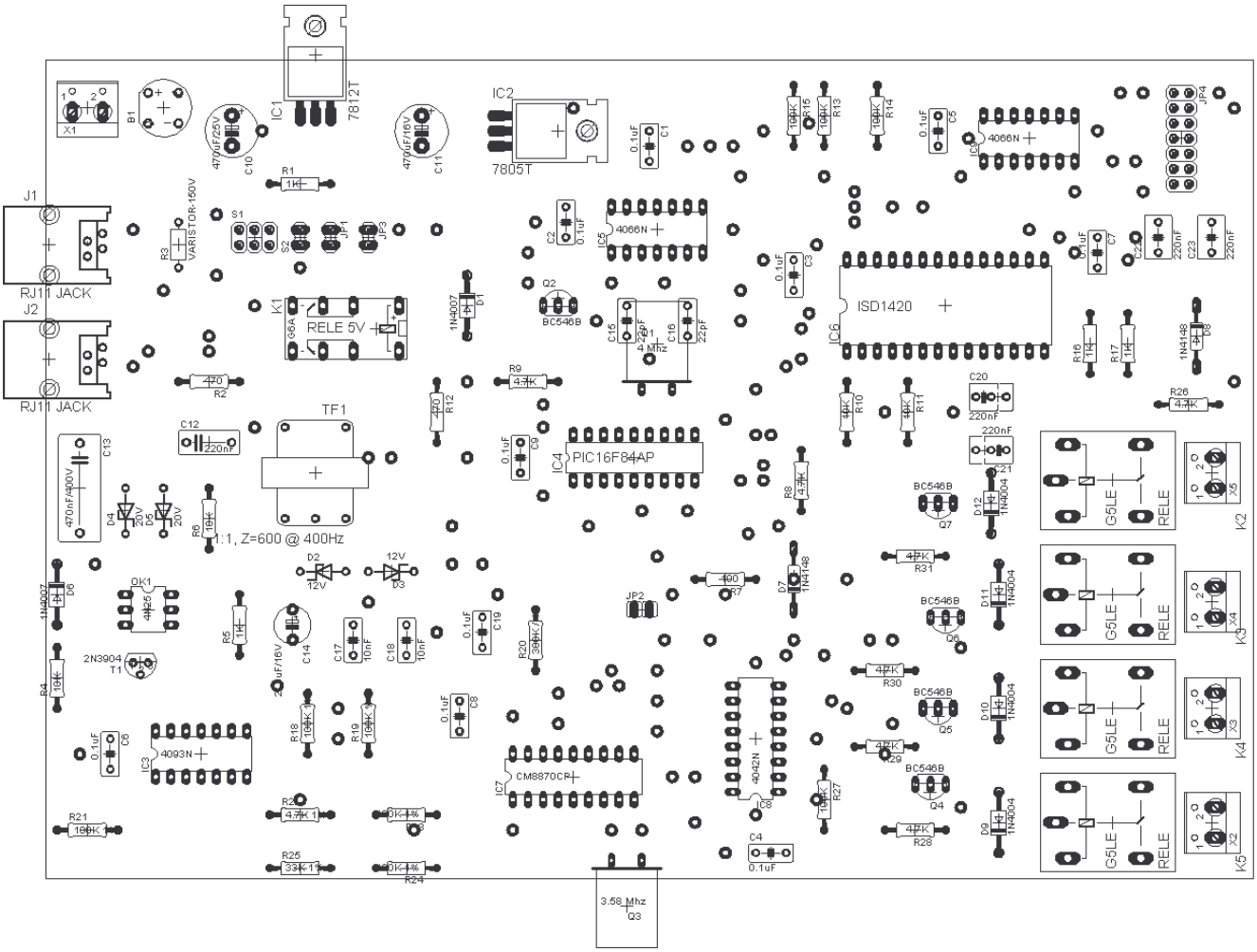


fig. 2.1-a

La fig. 2.1-a muestra la disposición de componentes de la placa electrónica del UCT, en la esquina superior izquierda están los conectores RJ11 que se conectan a la línea de teléfono. En el centro se ubica el microcontrolador PIC16F84A que es el “cerebro” del sistema. El integrado mas Grande es la ISD1420 que almacena los mensajes de audio. En el lado derecho de la placa se manejan los dispositivos externos de potencia y el puerto de comunicaciones BUS DCE.
Mas adelante se explicara detalladamente cada parte del circuito.

2.2: Guía de Uso del UCT

Antes de comenzar con las descripciones técnicas del proyecto, explicaremos su funcionamiento general y modo de uso, así será mas fácil de entender y explicar luego, el funcionamiento interno y desarrollo circuital del UCT.

2.2.0: Características generales del UCT

OBJETIVO GENERAL: Comandar dispositivos remotamente a través de la línea telefónica utilizando tonos DTMF generados por un aparato telefónico.

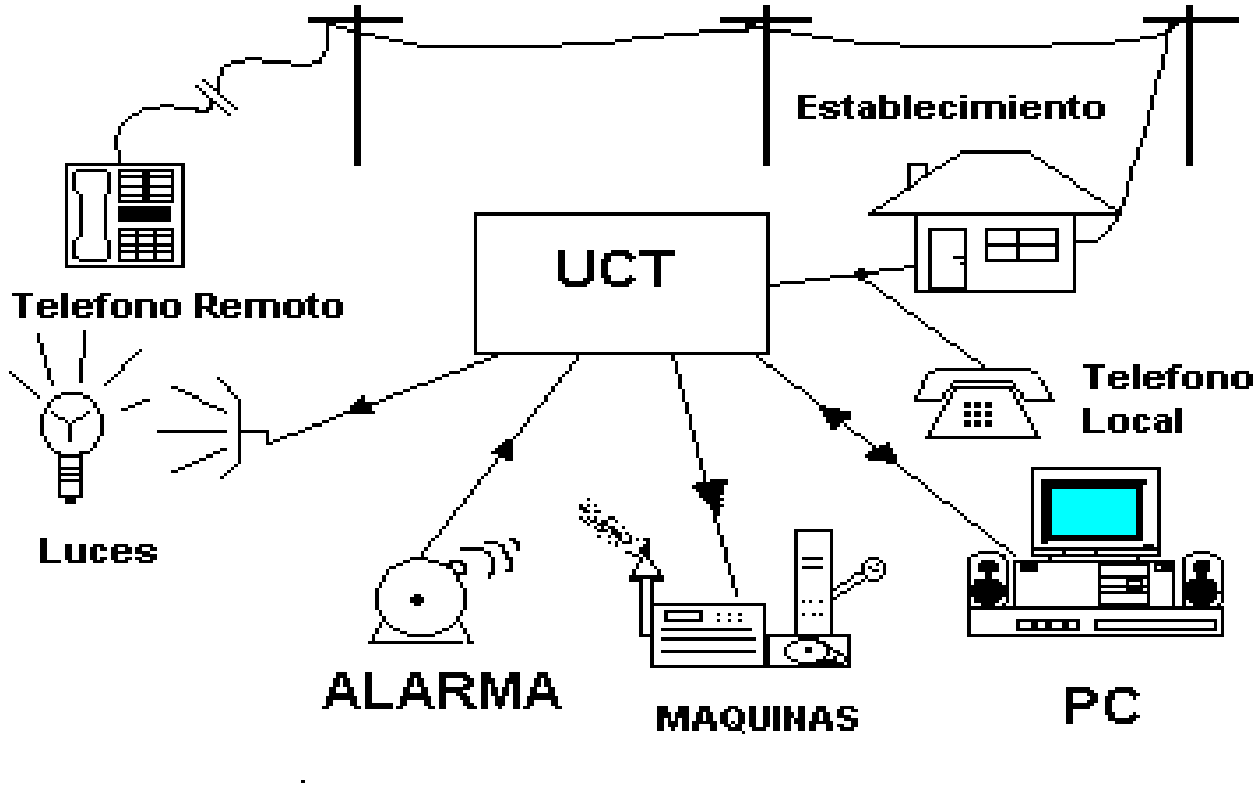


fig. 2.2.0-a

EL UCT PUEDE:

El UCT puede:

- Activar / Desactivar 4 dispositivos **ON/OFF** de una potencia máxima de **1800 W** cada uno.
- Enviar y recibir datos de dispositivos digitales conectados al Bus DCE (puerto de comunicaciones).
- Enviar y recibir audio de dispositivos conectados al Bus DCE.

- Incrementar sus capacidades electrónicas de forma dinámica mediante módulos electrónicos externos conectados al Bus DCE.
- Interactuar e informar, a través de menús y mensajes, de audio (voz) con el usuario remoto.
- Restringir el acceso remoto al aparato a través de una clave.
- Esperar un numero X de Rings antes de contestar un llamado.
- Permanecer en modo stand-by (bajo consumo) cuando no es utilizado.
- Permite la simulación completa de una llamada real, utilizando simplemente el teléfono local y sin ocupar la línea telefónica (control local). Útil para configurar al UCT y simular efectos de forma local.

2.2.1: Requerimientos y configuración del UCT:

2.2.1.2: El UCT requiere para su funcionamiento optimo:

- Una línea telefónica fija.
- Un teléfono con discado por tonos (DTMF).
- Alimentación eléctrica de 12 Vca.
- La línea telefónica y el teléfono deben ser conectados al UCT utilizando fichas RJ11.
- Los dispositivos de potencia son conectados utilizando borneras, las cuales tienen un tornillo para poder amarrar un cable y así usted puede conectar los dispositivos de potencia al UCT, que luego pueden ser activados o desactivados. El funcionamiento, es exacto al de cualquier interruptor.
- Los requerimientos para el BUS DCE serán explicados mas adelante.

2.2.1.2: Para configurar el UCT los pasos son los siguientes:

1) Conectarlo al suministro eléctrico.
2) Conectar la línea telefónica en el conector J1.
3) Conectar el teléfono en el conector J2.
4) Encender el UCT.
5) Poner el interruptor S2 en **modo control local**.
6) Descolgar el teléfono.
7) Presionar varias veces el interruptor S1 hasta que usted escuche en el auricular del teléfono: "**Ingresar Clave**"
8) Digite la secuencia: "**1234**" en su teléfono para poder entrar al UCT. Luego podrá cambiar esta clave. Solo tiene 3 oportunidades para ingresar

correctamente la clave. De lo contrario el UCT se pone en stand-by, y debe repetir el paso 7).

9) El UCT dará la bienvenida y reproducirá con audio el menú principal. Luego de que reproduzca el menú, puede elegir cualquier sub-menú u opción y ver los efectos de igual forma a como sucederían si usted estaría llamando remotamente al UCT. Por ahora simplemente presione "#" (numeral) en su teléfono para salir del UCT, y lea la sección **2.2.2** para conocer mas sobre los sub-menús del UCT.

Nota: Mientras el UCT este en **modo control local**, no podrá recibir llamadas telefónicas sobre el teléfono conectado al UCT, ni en el UCT.

2.2.2: Menús del UCT

El UCT informa los menús disponibles al usuario a través de mensajes de audio almacenados en su memoria. Los menús por lo general informan de las opciones disponibles a ejecutar en el UCT.

2.2.2.0: Reglas de los menús:

A menos que se especifique lo contrario:

- La tecla * (ASTERISCO) sirve para repetir los mensajes del menú actual.
- La tecla # (NUMERAL) sirve para salir del menú actual y volver al menú anterior. En caso de estar en el menú **principal**, desconecta al UCT de la llamada actual.
- Si no se presiona ninguna tecla del teléfono luego de que un mensaje fue reproducido o cuando el UCT espera un dato, durante el lapso de **1 minuto** el UCT se desconecta de la llamada actual.

2.2.2.1: Menú - Principal

Este menú es el primero en reproducirse al entrar con una clave correcta al UCT, el mismo informa sobre todas las funcionalidades disponibles en el UCT y permite acceder a otros sub-menús mas específicos.

Los siguientes mensajes serán reproducidos en el Menú principal:

"UNO, Control de Dispositivo"
"DOS, Ver estado de Dispositivo"
"TRES, Bus DCE"
"CUATRO, Configurar RINGS"
"CINCO, Cambiar Clave"
"NUMERAL, Salir"
"ASTERISCO, Repetir"

Donde las primero 5 opciones, son sub-menues.
La opción NUMERAL , desconecta al UCT y lo pone en stand-by a la espera de una nueva llamada.
La opción ASTERISCO, repite el menú principal.
Las demás opciones se explican por si solas, por Ej. si usted presiona la tecla DOS de su teléfono, podrá ver el estado de los dispositivos.

Hay una opción que no se reproduce en este menú, pero que es útil y puede ser usada, ella es la opción **0** (CERO)**.** Al digitar este numero habilitara o deshabilitara la reproducción de audio en el UCT, según sea su estado actual. Ello es útil para usuarios experimentados que no deseen perder el tiempo en escuchar mensajes que ya conocen.

Los siguientes párrafos explican cada submenú.

2.2.2.2: Menú – Control de Dispositivos:

El siguiente mensaje será reproducido al ingresar en el Menú - Control de Dispositivos:

"INGRESAR DISPOSITIVO"

Ahora usted debe elegir un dispositivo, para ello ingrese un numero entre el **1 y el 4**, el cual representa el numero de dispositivo a activar o desactivar.

A continuación ingrese la acción a realizar:

0 -> **para desactivar dispositivo seleccionado.**
1 -> **para activar dispositivo seleccionado.**

Luego, si todo salió bien el UCT responderá:

"DISPOSITIVO ACTIVADO" (Si usted eligió activar el dispositivo)

O

"DISPOSITIVO DESACTIVADO" (Si usted eligió desactivar el dispositivo)

Posteriormente el UCT volverá a repetir este menú para que usted pueda desactivar / activar otro dispositivo nuevamente, o simplemente use la tecla # (NUMERAL) para volver al menú principal.

Este tipo de dispositivos que solo pueden ser activado o desactivados, se llaman DISPOSITIVOS **ON/OFF** en terminología del UCT.

2.2.2.3: Menú – Ver Estado de Dispositivos:

El Menú – Ver Estado de Dispositivos, solo informa el estado actual de los dispositivos ON/OFF conectados al UCT. Una vez que ha informado el estado de los dispositivos, vuelve al menú principal sin necesidad de presionar ninguna tecla telefónica.
Por ejemplo si Usted tiene los dispositivos 1 y 3 activados, el UCT reproducirá los siguientes mensajes:

"DISPOSITIVO:"
"UNO ACTIVADO"
"DOS DESACTIVADO"
"TRES ACTIVADO"
"CUATRO DESACTIVADO"

Luego volverá al menú – principal.

2.2.2.4: Menu – Bus DCE:

EL Menú – Bus DCE es un puerto de comunicación bidireccional entre el UCT y dispositivos externos que requieren comandos mas complejos que un simple ON/OFF.
Este tipo de dispositivo se denomina **DCE** (Dispositivo de Control Externo) en la terminología del UCT.
Gracias a este puerto el UCT puede incrementar su funcionalidades o potencial, dependiendo del tipo de DCE que este conectado al UCT.

En este menú se reproducirán los siguientes mensajes:

"UNO, CONECTAR"
"DOS , DISPOSITIVO EXTERNO"

La primera opción, permite transferir datos a un DCE o que un DCE transfiera datos al UCT, través de un protocolo (**modo conectado**).
Actualmente no esta implementado en la versión actual del software del UCT esta opción, pero si lo esta por hardware . Lo cual en un futuro cercano, sin modificar el circuito, estará disponible esta funcionalidad.

La segunda opción permite solo transferir datos unidireccionalmente de forma cruda (**modo desconectado**) a un DCE.
Ello significa que enviaremos numero binarios puros sin el uso de protocolos desde el UCT al DCE.
Esto es útil para aplicaciones simples en las cuales no es necesario o practico usar un microcontrolador para interpretar los datos provenientes del UCT.
Esta opción esta implementada actualmente y al elegirla se reproduce el sig. Mensaje:

"INGRESAR NUMERO"

Usted ahora debe ingresar un numero entre el **0 y el 9**, una vez ingresado y si todo salió bien, el numero ingresado estará disponible en forma binaria en el DCE conectado al BUS DCE y el UCT reproducirá:

"LISTO"

Lea las especificaciones técnicas del BUS DCE para mas información.

Luego puede volver a ingresar otro numero cuando el UCT reproduzca nuevamente:

"INGRESAR NUMERO"

Nota: En este sub-menú el UCT espera un máximo de 4 minutos el ingreso de algún dato.

Para volver al menú anterior presione la tecla # (NUMERAL).

2.2.2.5: Menu – Configurar RINGS:

El Menú – Configurar RINGS, permite establecer el numero de RINGS que debe esperar el UCT antes de contestar un llamado.

Al seleccionar este Menú el UCT reproducirá:

"INGRESAR RINGS"

Ahora debe ingresar un numero comprendido entre el **1 y el 9**, lo cual especificara la cantidad de RINGS que el UCT debe esperar antes de contestar un llamado.
Si usted quiere establecer un numero mayor a 9 RINGS en el UCT, puede ingresar el numero **0**, que establece 18 RINGS a esperar. Esto es útil por si se quiere tener activo el UCT, pero que conteste solo cuando haya demasiados RINGS.
Note que si usted contesta primero el teléfono, el UCT permanecerá inactivo y no interferirá con la llamada actual.

Una vez ingresado el numero de RINGS el UCT responderá:

"LISTO"

Y retornara al menú principal.

Nota: Por fabrica el UCT esta configurado para responder a los 4 RINGS.

2.2.2.6: Menú – Cambiar Clave:

El Menú – Cambiar Clave, permite establecer una clave de **4** dígitos en el UCT, que será pedida cada vez que el UCT atienda el teléfono.

Si es ingresada incorrectamente 3 veces, el UCT se desconectara del actual llamado y deberá ser llamado de vuelta para poder ingresar nuevamente la clave.

Cuando se selecciona este menú, el sig. Mensaje es reproducido:

"UNO, CAMBIAR CLAVE"
"NUMERAL, SALIR"
"ASTERISCO, REPETIR"

Usted debe presionar la tecla **1** para confirmar que desea cambiar la clave o **#** (NUMERAL) para volver al menú principal.

Una vez presionado **1** usted debe ingresar la nueva clave, para ello digite una secuencia de 4 dígitos, la cual será su nueva clave.
Los dígitos permitidos para la nueva clave son:

0,1,2,3,4,5,6,7,8,9,#,*

Notar que en este punto no puede presionar # (NUMERAL) para salir, ya que será interpretado como parte de la nueva clave.

Luego de ingresar la nueva clave, el UCT reproducirá:

"LISTO"

Y retornara al menú principal.

Nota: La clave configurada por fabrica es "**1234"**, usted debe cambiarla por alguna otra de inmediato!.

Nota 2: Un intruso deberá realizar 6912 llamados telefónicos para poder realizar todas las combinaciones posibles de la clave y entrar al UCT.

2.2.3: Configuración Inicial del UCT:

Ahora estableceremos una clave nueva y el numero de RINGS en el UCT. Primero lea la sección **2.2.1.2**, y siga los pasos descriptos hasta llegar al numero **9)**, para mayor seguridad lea también la sección **2.2.2**.

Luego de que el menú principal sea reproducido digite:

5 (espere que se reproduzcan los mensajes) y luego digite **1.**

Ahora ingrese su nueva clave de **4** dígitos.

Espere a que se reproduzca el menú principal nuevamente, y digite:

4 (espere a que se reproduzcan los mensajes) y luego ingrese el numero de RINGS que el UCT deberá esperar antes de contestar el llamado.

Espere a que se reproduzca el menú principal nuevamente, y digite:

(numeral) para salir del UCT.

Ahora cambie de posición el interruptor **S2** a modo **control remoto,** para que el UCT pueda responder a un llamado telefónico.
Cuando un llamado telefónico se reciba el UCT esperara los RING establecidos por usted y luego le pedirá su clave personal para poder ingresar a comandar dispositivos.

2.2.4: Operación remota del UCT:

Para habilitar el UCT para que pueda ser operado a través de un llamado telefónico, debe poner el interruptor **S2** en modo **control remoto.**
Luego llame al UCT, ingrese su clave y luego navegue por los menú, para activar dispositivos, cambiar claves, etc... lea la sección **2.2.2** para mas información sobre menues.

2.2.5: Errores:

Cuando ingrese números o datos incorrectos el UCT le informara por medio de mensajes de audio del error, y luego volverá a pedirle que ingrese nuevamente el dato.

2.2.6: Preguntas Frecuentes (FAQ):

P: Puedo interrumpir al UCT mientras esta reproduciendo mensajes de audio?
R: NO, los tonos DMTF recibidos serán ignorados mientras el UCT reproduce un mensaje.

P: Como es la interfaz del BUS DCE?
R: Lea la sección técnica. (3.0 en adelante).

P: Puedo simular una llamada localmente y controlar al UCT?
R: Si lea sección **2.2.1.2.**

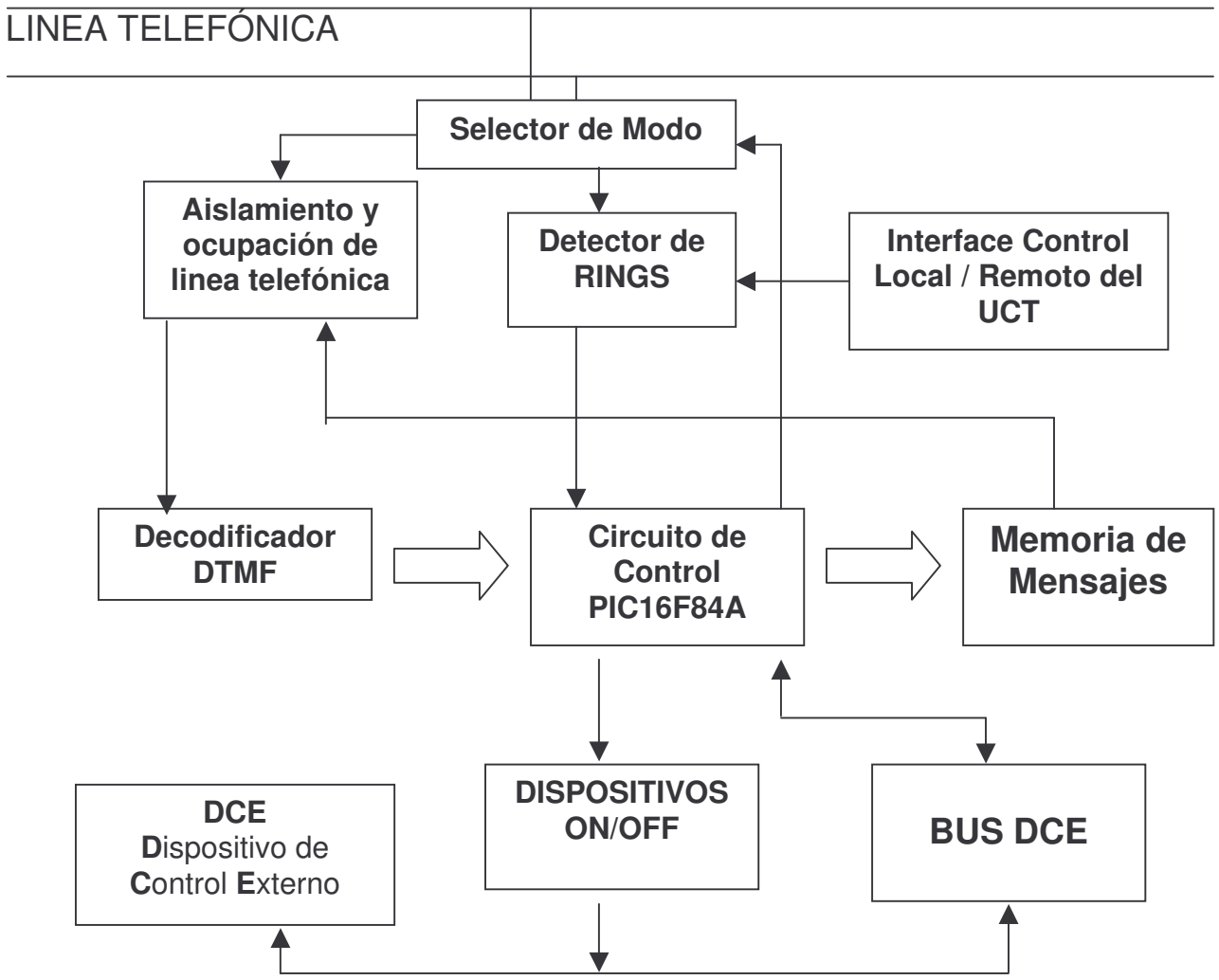
P: Conozco perfectamente los menús del UCT y sus mensajes, hay alguna forma de evitar que se reproduzcan sus mensajes de audio?
R: Si, hay dos formas:

1) Cuando ingresa la clave correctamente, tiene **2 segundos** para digitar el numero **0 (cero)**, el cual evitara la reproducción de cualquier menú.
2) Cuando se esta en el Menú principal, digite **0 (cero)**, ello desactivara o activara los mensajes de audio, según sea su estado actual.

Los mensajes de confirmación, continuaran reproduciéndose. Por ejemplo: **"DISPOSITIVO ACTIVADO"**, se reproducirá al activar un dispositivo, pero el menú principal y el menú de control de dispositivos, no serán reproducidos.

3.0: Desarrollo Electrónico del UCT

3.1: Funcionamiento por Bloques:



Básicamente el circuito espera en stand-by un llamado telefónico a través de la línea telefónica cuando es puesto en control modo remoto a través de la **interface de control Local / Remoto del UCT**, cuando el llamado ocurre, el **detector de RINGS** le informa al **circuito de control (pic16f84a)** la presencia de un RING. Luego **circuito de control (pic16f84a)**, decide si la cantidad de RINGS es suficiente y contesta el llamado ocupando la línea telefónica a través del **selector de modo** que activa el bloque **aislamiento y ocupación de línea telefónica**.
Posteriormente el **circuito de control** reproduce los mensajes correspondientes de la **memoria de mensajes**, y el usuario ingresa, claves, datos, comandos etc a través de tonos DTMF que son decodificados por el

decodificador DTMF y enviados al **circuito de control** para ser procesados. Luego el usuario activa o desactiva **dispositivos ON/OFF** o se comunica con **dispositivos de control externos** conectados al **bus dce.**

Lo mencionado anteriormente puede haber sido simulado de forma local poniendo al UCT en modo control local a través del bloque **interface de control Local / Remoto del UCT.**

3.1: Descripción Breve del funcionamiento de cada bloque:

Selector de Modo:

Cuando el UCT ha recibido una llamada, el **selector de modo** conecta la línea telefónica al **circuito de aislamiento eléctrico** (modo respuesta)**,** por donde fluirá la información al **decodificador DTMF.**
El **Selector de Modo** es controlado por el **circuito de control.**

Detector de RINGS

Detecta un llamado telefónico al UCT y activa el **circuito de control.** Permite también que el **circuito de control** cuente la cantidad de RINGs para contestar la llamada en un numero predeterminado de Rings.

Aislamiento y ocupación de línea telefónica

Circuito encargado de aislar la línea telefónica eléctricamente del UCT, también cumple la función de ocupar la línea telefónica en caso de que el UCT decida contestar una llamada telefónica. Por este circuito fluirá la información de los códigos DTMF provenientes del exterior y los mensajes de respuesta (en forma de audio) de la **memoria de mensajes**.

Decodificador DTMF

Permite decodificar los códigos DTMF a números binarios, que serán pasados en forma digital al **circuito de control** para ser procesados y consecuentemente llevar a cabo una acción.

Memoria de Mensajes

Contiene mensajes en forma de Voz (señales de audio) que permitirán indicar al usuario que llama remotamente las acciones que se están llevando a cabo en el UCT.
Los mensajes son del tipo: “Dispositivo Activado”, “Ingresar Clave “, “Dispositivo desactivado”, “Configurar RINGS”, “Bienvenido” , etc

Interface de control local / remoto del UCT

Permite elegir el modo de operación del UCT, los modos disponibles son:
Modo control Local y Modo control Remoto.
El primero permite el control local sin ocupar la línea telefónica, y el segundo permite dejar al UCT para recibir llamadas y ser controlado remotamente.

El **Modo control Local** permite configurar localmente al UCT, es decir establecer clave de acceso, numero de RINGs, activar dispositivos localmente, etc... sin la necesidad de ocupar la línea telefónica o llamar desde un teléfono remoto.
Este circuito simulara una llamada externa, de esta forma el usuario podrá cambiar la configuración usando solo el teléfono local y escuchando los cambios por el auricular, lo cual evita la necesidad de implementar controles externos en el UCT y abaratar costos.

Circuito De Control (PIC16F84A)

Este es básicamente el cerebro del UCT, permite procesar los códigos provenientes del exterior y llevar al cavo las acciones, tales como reproducir mensajes, activar dispositivos, pedir contraseña, ver el estado de los dispositivos, controlar al resto de los circuitos del UCT, etc.
Como esta construido con un microcontrolador, necesita un software para funcionar, el cual comanda el resto de los circuitos y procesa los datos proveniente de ellos.

DISPOSITIVOS ON/OFF

Etapa de manejo de potencia, permite al **circuito de control** activar / desactivar dispositivos de gran consumo eléctrico o simplemente de pequeño consumo. Brindando un aislamiento adecuado con el resto del UCT.

BUS DCE

Básicamente permite una expansión bidireccional inteligente del UCT, aquí se podrá conectar un dispositivo complejo que necesite una serie de comandos para ser activado, aquellos dispositivos que necesiten mas información que un simple ON/OFF. Por ejemplo podremos conectar una Alarma que requiera una clave para ser activada, o algún dispositivo que permita el uso de comandos para llevar acciones que el UCT no puede realizar, o quizás un sensor que reporte algún evento, etc.

DCE (Dispositivo de Control Externo)

Este bloque no forma parte del UCT, es solo un modulo electrónico que puede ser conectado al BUS DCE opcionalmente para expandir las funcionalidades del UCT o simplemente para recibir ordenes del UCT.

3.2: Términos:

- Modo de espera: El UCT se encuentra en STAND-BY consumiendo poca energía. Por lo general cuando no hay una llamada en progreso.
- Modo respuesta: El UCT recibió una llamada y se encuentra interactuando con el usuario remoto o fue activado por el bus DCE.
- Modo Control Local: El UCT funciona normalmente, pero no por acción de un usuario remoto, sino que fue activado por un usuario local que desea simular una llamada con su teléfono local. En este modo el UCT no puede recibir llamadas externas.

3.3: Esquema circuital de los bloques

Debido a que muchos bloques están estrechamente relacionados, pero no por ello dejan de ser independientes, dividiremos al UCT en 3 Circuitos principales.

1: Circuito de interfaz de línea y Alimentación.

El cual contiene los módulos:

- **Selector de Modo**
- **Detector de RINGS**
- **Aislamiento y Ocupación de línea telefónica**
- **Interface de configuración local.**

2: Circuito Principal

El cual contiene los módulos:

- **Circuito de Control (PIC16F84A)**
- **Decodificador DTMF**
- **Memoria de mensajes**

3: Circuito de Potencia y BUS DCE

El cual contiene los módulos:

- **Dispositivos ON/OFF**
- **BUS DCE**

3.4: Circuito de interfaz de línea y Alimentación

La sección de alimentación de este circuito contiene una entrada de **12 Vca**. La cual es rectificada, para luego proveer +5V y +12V de CC a través de los reguladores de tensión LM7805 y LM7812, que serán necesarios para alimentar el resto de los circuitos.
El circuito completo del UCT consume aproximadamente como corriente mínima 20 mA (en modo stand-by) y una máxima de 250 mA (cuando se controla localmente y todos los reles de los dispositivos ON/OFF están activados).
El circuito de alimentación fue diseñado para poder suministrar 1 A como corriente máxima, y así poder alimentar con suficiente corriente a los circuitos externos conectados al **BUS DCE**.

La línea telefónica se conecta al conector J1, donde se ubica un varistor de 150 V, (R3) que preteje al resto de los circuitos de tensiones superiores a 150 V que pudieran presentarse en la línea telefónica.

Luego la llave S1 permite **seleccionar el modo** de operación del UCT, los modos pueden ser: **Control remoto** o **Control local.**
Cuando se elige **modo Control Remoto** la línea telefónica es derivada al **circuito detector de RINGS** a través del rele K5 y al conector J2 donde se coloca el aparato telefónico (el cual podrá ser operado normalmente en este modo, se podrán hacer llamadas, recibir llamadas, etc.).
El **circuito detector de RINGS** esta compuesto por:
C13: Capacitor donde se elimina CC. También ofrece relativa baja impedancia a la señal de RING de 25 Hz y soporta tensiones elevadas (470nF/400V).
D4,D5: Diodos Zener en back-to-back que filtran picos de ruidos menores a 20 V presentes en la línea TE.
D6: Protege diodo del opto acoplador de tensiones inversas superiores a 2 V.
R4: Requerido para presentar una impedancia de 10Kohms a señales de audio y limitar la corriente de RING de 10mA A 20 mA.
OK1: Opto acoplador 4N25 que detecta ciclo positivo de la señal de RING. Cuando un RING es detectado se presenta un 1 lógico en el Emisor de T1 (el cual esta en configuración Darligton con el transistor del opto acoplador, necesario ya que la corriente generada por un RING es muy pequeña para saturar completamente el transistor del Opto acoplador) .
C14: mantiene el 1 lógico del RING de forma continua, ya que la señal de RING tiene una frecuencia de 25 Hz y eso produciría un estado lógico “1” pulsante de 25 Hz a la salida del emisor de T1.

IC3A: Inversor con histéresis (trigger Schmith) que produce la señal lógica /RING (0-> RING PRESENTE, 1-> RING AUSENTE).
Por ultimo el detector de RING fue calculado para funcionar con una señal de RING de 135 Vpico-pico de 25 Hz montada sobre una tensión de 35 a 50 Vcc.

Al elegir el **modo Control local** con la llave S1, la entrada de línea es desconectada del circuito y se alimenta al conector J2 con 12 Vcc, esto permite alimentar el aparato telefónico sin necesidad de usar la línea telefónica. Permitiendo de esta manera operar al teléfono normalmente, pero sin la posibilidad de ocupar la línea telefónica.

El **circuito detector de ring** en su entrada no se ve afectado ya que los 12Vcc no generara circulación de CC gracias a C13.
Para permitir un circuito consistente y transparente desde el punto de vista de su funcionamiento, se emplea el interruptor S2 que permite polarizar la base del transistor del opto acoplador OK1 de forma manual, de esta manera generamos o simulamos RINGS de forma local a través de S1.
Debido a que S1 no presenta un circuito antirebote, se empleo un capacitor C14 de 220 uF, que junto a R5 de 1Kohms, establecen una constante de tiempo suficientemente grande para eliminar rebotes de la llave S1 y así mismo generar un 1 lógico estable al producirse un RING en el modo **control remoto.**
Cabe destacar que C14 no es un perfecto circuito antirebote, pero es barato y sastiface las necesidades del aparato.

Cuando el UCT detecta los RINGS necesarios para ocupar la línea telefónica activa al RELE K1 a través del PIC16F84, de modo que se conecta la línea telefónica con la entrada de audio telefónico del UCT.
De otra forma, T_LINEA-1 y T_LINEA-2 son conectados al transformador TF1.

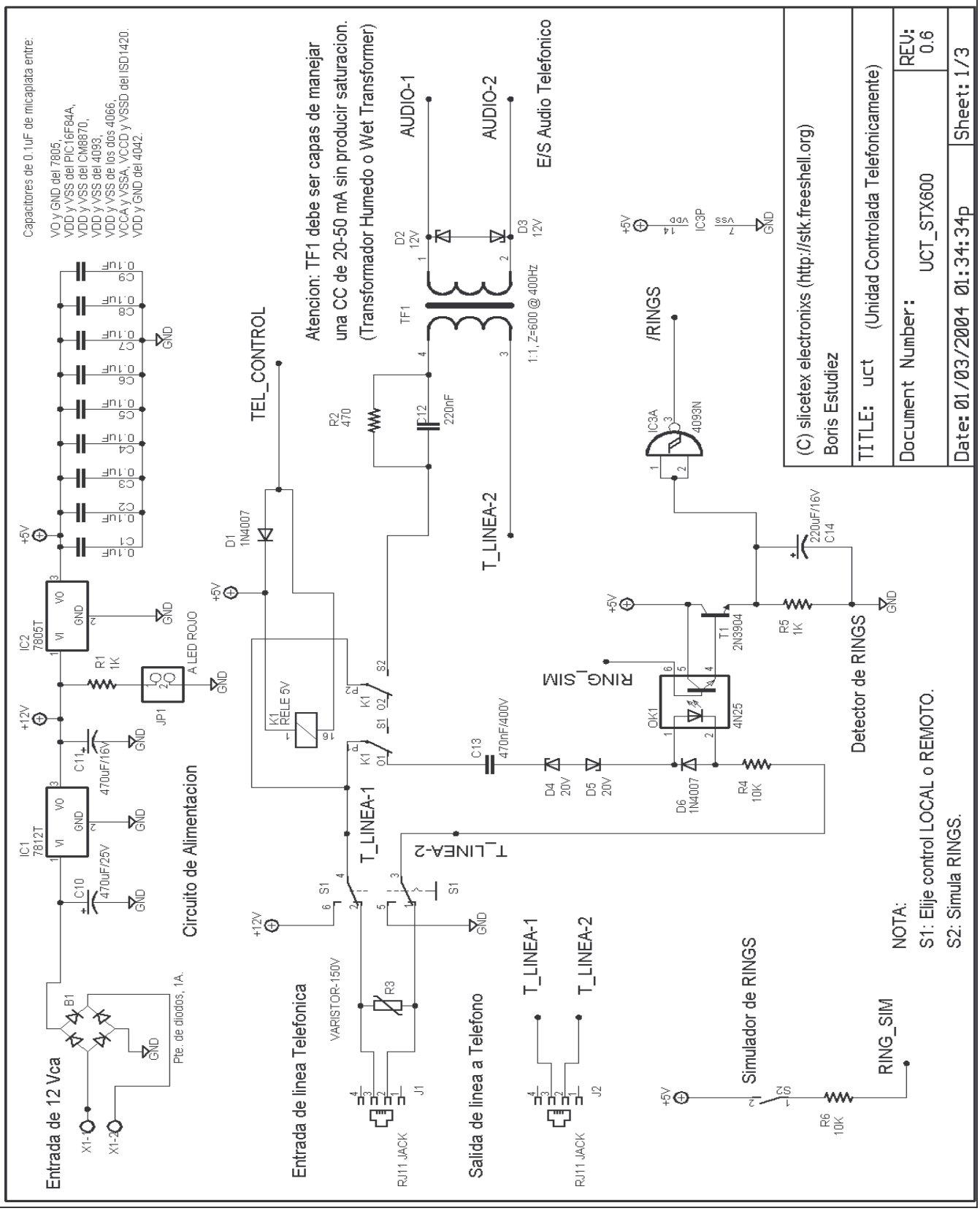
En este punto se ocupa la línea telefónica de la siguiente manera:
R2: Toma suficiente CC de la línea telefónica como para ocupar la línea telefónica (toma de unos 15 a 25 mA dependiendo de la tensión disponible en la línea telefónica). Este valor fue elegido tomando como promedio la impedancia de los aparatos telefónicos, que es del orden de los 300 a 500 Ohms.
C12: Evita la perdida innecesaria de potencia para las señales de audio superiores a 300 Hz.
TF1: es un transformador que presenta una impedancia de 600 Ohms a 400 Hz y relación 1:1, ello sastiface ciertas normas telefónicas que especifican aislamiento eléctrico del aparato conectado a la línea telefónica y adaptación de impedancias para señal de 600 Ohms a 400 Hz.

TF1 debe ser un transformador Húmedo, ello significa que permita el paso de CC a través de su bobinado, y no produzca saturación o distorsión cuando la señal alterna también pase por su bobinado. Este tipo de transformadores son ligeramente mas caros, pero simplifican el resto del circuito asociado que ocupa la línea telefónica. Finalmente a la salida de TF1 se encuentran dos diodos Zener (D2 y D3) de 12V en configuración back-to-back, que permite que el audio que sale de TF1 este limpio de picos de tensiones superiores a 12V (circuito limitador), necesario para proteger al resto de circuitos conectados a las líneas AUDIO-1 y AUDIO-2, que permiten la Entrada / salida de audio telefónico.

Algunas Notas:

- Cuando se elige **modo control local** se escucha un zumbido en el parlante del aparato telefónico. Ello se debe a que circula demasiada CC en TF1 y se produce saturación. La solución simple consiste en poner una resistencia de 220 Ohms en serie con el aparato telefónico. Ello puede ser logrado ubicándola en serie con fuente de 12Vcc y la pata 6 del interruptor S2. Si 220 ohms no es la solución busque otro rango de resistencias entre los 100 y 570 Ohms. Recuerde que a mayor resistencia, menor será la tensión disponible para alimentar su teléfono en **modo control local.**
- La forma de ocupar la línea telefónica con R2 en serie con TF1 no es una solución elegante, pero si económica y simple. En futuras versiones el UCT mejorara esta parte del circuito.

El esquema del **Circuito de interfaz de línea y Alimentación** se muestra a continuación:


Capacitores de 0.1uF de micaplata entre:
VO y GND del 7805,
VDD y VSS del PIC16F84A,
VDD y VSS del CM8870,
VDD y VSS del 4093,
VDD y VSS de los dos 4066,
VCCA y VSSA, VCCD y VSSD del ISD1420.
VDD y GND del 4042.
Entrada de 12 Vca
Pte. de diodos, 1A.
Circuito de Alimentacion
IC1 7812T
IC2 7805T
470uF/25V
470uF/16V
A LED ROJO
JP1
R1 1K
TEL_CONTROL
D1 1N4007
K1 RELE 5V
Atencion: TF1 debe ser capas de manejar una CC de 20-50 mA sin producir saturacion.
(Transformador Humedo o Wet Transformer)
R2 470
TF1
220nF
1:1, Z=600 @ 400Hz
AUDIO-1
AUDIO-2
D2 12V
D3 12V
E/S Audio Telefonico
T_LINEA-1
T_LINEA-2
S1
S2
C13 470nF/400V
D4 20V
D5 20V
D6 1N4007
R4 10K
OK1 4N25
RING_SIM
/RINGS
IC3A 4093N
IC3P
T1 2N3904
R5 1K
220uF/16V C14
Detector de RINGS
Entrada de linea Telefonica
VARISTOR-150V
R3
RJ11 JACK
J1
Salida de linea a Telefono
J2
Simulador de RINGS
S3
R6 10K
RING_SIM
NOTA:
S1: Elije control LOCAL o REMOTO.
S2: Simula RINGS.
(C) slicetex electronixs (http://stk.freeshell.org)
Boris Estudiez
TITLE: uct (Unidad Controlada Telefonicamente)
Document Number: UCT_STX600
REV: 0.6
Date: 01/03/2004 01:34:34p
Sheet: 1/3

3.5: Circuito Principal

Este circuito es el principal del UCT y su funcionamiento depende altamente de la parte de software que controla al microcontrolador PIC16F84A (IC4). Igualmente trataremos de describir el funcionamiento por hardware:

El **IC7 es un CM8870**, el cual es un decodificador DMTF, que traduce las señales DTMF a códigos binarios.
Su funcionamiento se explica mejor en su hoja de datos, aquí solo diremos que:
AUDIO-1 y AUDIO-2 entran en forma diferencial al IC7, ello mejora el comportamiento frente a ruidos. Los capacitores y resistencias asociadas son del 1 % como se especifica en la hoja de datos.
Sus salidas digitales son:
STD: En nivel alto, informa que hay un DTMF presente en la entrada de audio y que ya esta decodificado en binario en Q1, Q2, Q3 y Q4. Esta salida esta conectado a RA4 del IC4.
Q1, Q2, Q3 y Q4: Salidas que representan el código binario del DTMF decodificado. Están conectadas al bus de datos RB[4..7] del IC4.
Su entrada digital es:
TOE: en nivel bajo, mantiene en alta impedancia a Q1, Q2, Q3 y Q4. Se conecta a RA0 del IC4.

El integrado **IC5 es un 4066**, el cual funciona como 4 llaves analógicas, permiten separar con alta impedancia el bus de datos RB[4..7] de la memoria **ISD1420 (IC5)**.
Para habilitar las llaves de este integrado se debe poner en nivel alto la línea RA1 del IC4.

La memoria **ISD1420 (IC6)** es una memoria de voz, que almacena todos los mensajes de audio del UCT.
Dichos mensajes son seleccionados por el IC4 y reproducidos por la memoria ISD1420 (IC6) a través de sus salidas analógicas SP+ y SP- conectadas a AUDIO-1 y AUDIO-2. Los capacitores C20, C21 y resistores R16 y R17 son necesarios para reducir la amplitud de la señal de salida de audio de la memoria y para filtrar frecuencias bajas.
Notar que la salida de la memoria es en forma diferencial, esto mejora los ruidos producidos por la misma y permite mejor aprovechamiento de la potencia de salida (4 veces superior que en modo común).
El modo de operación de la memoria puede ser leído de su hoja de datos o interpretado de la función **play_isd()** del software interno del PIC16F84A.

PIC16F84A:

Finalmente explicaremos el conexionado y funcionamiento general del **IC4, PIC16F84A** en el **UCT.**
El PIC16F84A es un microcontrolador de propósito general y de reducido costo. Posee 1 K Words de memoria de código de programa, 64 Bytes de memoria EEPROM y 68 bytes de memoria RAM y un CPU RISC.
Se utilizo un oscilador de cristal de cuarzo a 4 MHz, (Q1), para generar una velocidad de clock de 1 MHz, velocidad suficiente para el proyecto.

Debido a su escasa memoria de código de programa, el UCT debió ser programado de forma tal que se aprovechara al máximo la memoria disponible del PIC16F84A, ello ocasionó, que muchas características del originales del UCT, todavía no estén disponibles debido que deben pensarse para que puedan ocupar la menor cantidad de memoria.

DISTRIBUCIÓN DE PUERTOS:

PORTB:

- **RB[4...7]**: Entrada / salida, **Bus de datos** entre dispositivos y el PIC16F84A.
- **RB3**: Salida, envía señal de ocupación de línea telefónica.
- **RB2**: Salida, envía interrupción a un DCE conectado al BUS DCE. (/DCEINT)
- **RB1**: Entrada, Identifica interrupción generada por un DCE conectado al BUS DCE. (/UCTINT_PIN).
- **RB0:** Entrada, Recibe interrupciones externas. El programa interno del PIC, se encarga de analizar el origen de tal interrupción.

PORTA:

- **RA0**: Salida, Elige al 8870 (IC7) y conecta el bus de datos a sus salidas.
- **RA1**: Salida, Elige a al 4066 (IC5) que conecta a la ISD1420 (IC6) al bus de datos para controlarla.
- **RA2**: Salida, Elige el 4042 (IC8) y conecta bus de datos a su entrada.
- **RA3**: Salida, Elige al 4066 (IC9) y conecta al bus de datos al bus dce.
- **RA4**: Entrada, recibe pulso que indica que el IC7 recibió un DTMF.

Adicionalmente se dejo un jumper JP2 que permite resetear al IC4, para fines de depuración.

RB3 al estar en nivel alto satura el transistor Q2 y el rele K5 se activa ocupando la línea telefónica como se menciono en el apartado **3.4.** La misma señal activa un LED verde conectado a JP3 que indica que hay una conexión o que se esta ocupando la línea telefónica.

INTERRUPCIONES:

La compuerta IC3C cuando esta en CERO lógico interrumpe al IC4 a través de **RB0.** Para determinar la fuente de la interrupción, nos valemos de la siguiente lógica:
Si un dispositivo conectado al bus DCE genero una interrupción, el /**UCTINT_PIN** estará en un nivel bajo, por lo tanto **RB1** estará junto a **RB0** en nivel bajo.
Por lo cual deducimos que la interrupción provino del BUS DCE.
Si solo hay un RING presente el la línea telefónica, **RB1** estará a nivel alto y **RB0** en nivel bajo**.** Por lo cual deducimos que la interrupción es por un intento de llamada o RING.
Si ambos, el bus DCE y un RING están presente, la prioridad de interrupción la tiene el BUS DCE.
Ello se explica mejor en la siguiente tabla:

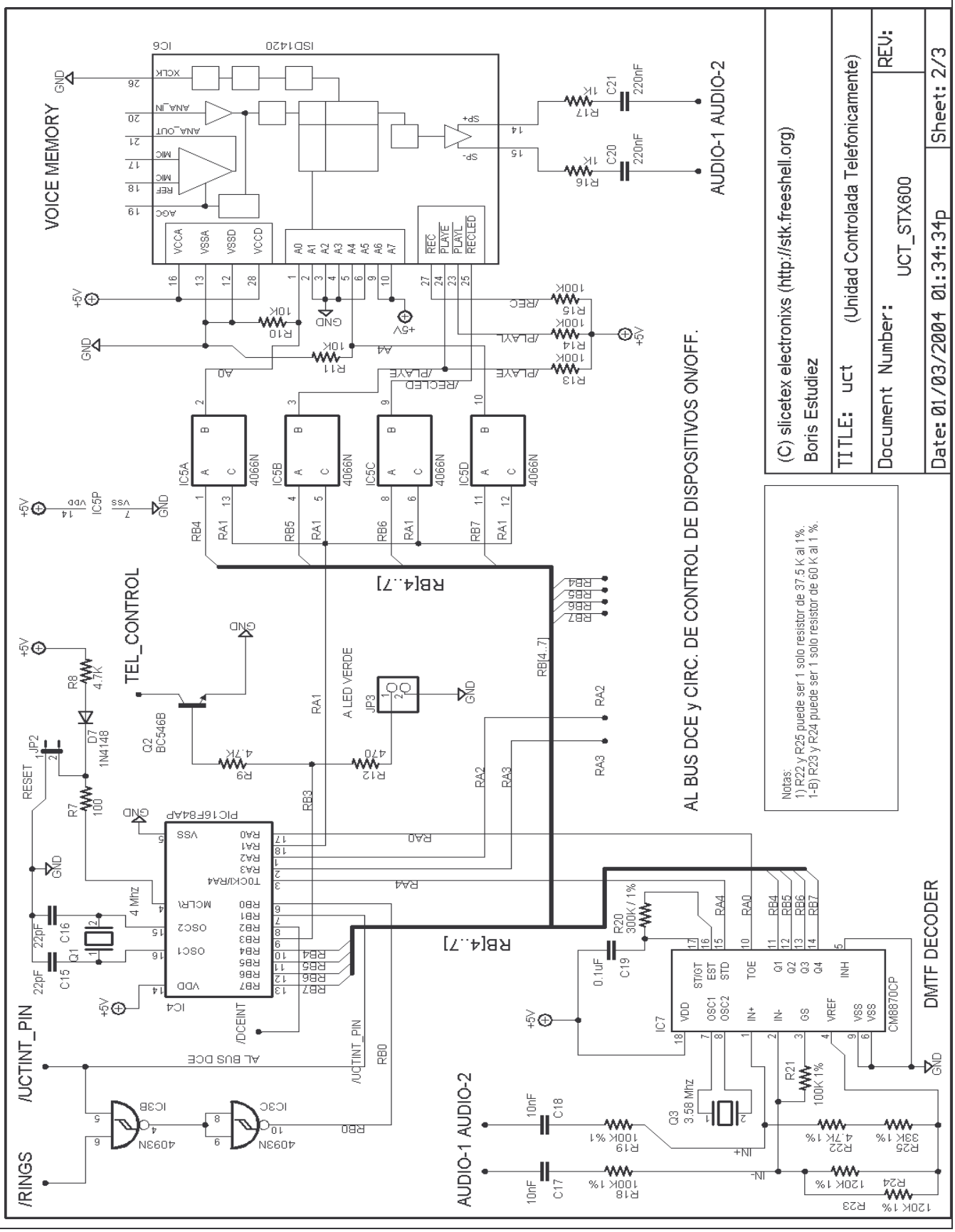
RB0	RB1	Fuente de la Interrupción / Efecto
1	1	No hay interrupción presente
0	1	Interrupción por **detector de RING**.
0	0	Interrupción por **BUS DCE**.
1	0	Nunca puede suceder!.

Finalmente, el comportamiento interno del PIC16F84A, es producido por el software que lo controla, para ello ver los archivos:

uct.asm y **uct.inc**

Que están disponible en la pagina del proyecto.

El esquema del **Circuito Principal** se muestra a continuación:


VOICE MEMORY
ISD1420
IC6
AUDIO-1 AUDIO-2
AL BUS DCE y CIRC. DE CONTROL DE DISPOSITIVOS ON/OFF.
(C) slicetex electronixs (http://stk.freeshell.org)
Boris Estudiez
TITLE: uct (Unidad Controlada Telefonicamente)
Document Number: UCT_STX600
REV:
Date: 01/03/2004 01:34:34p
Sheet: 2/3
Notas:
1) R22 y R25 puede ser 1 solo resistor de 37.5 K al 1%.
1-B) R23 y R24 puede ser 1 solo resistor de 60 K al 1 %.
TEL_CONTROL
A LED VERDE
RB[4..7]
PIC16F84AP
IC4
DMTF DECODER
CM8870CP
IC7
/UCTINT_PIN
/RINGS
AUDIO-1 AUDIO-2
4093N
4066N
RESET
4 Mhz
3.58 Mhz

3.6: Circuito de Potencia y BUS DCE

El **circuito de potencia** consta de 4 Reles que son manejados por el **latch 4042 (IC8)**.
Cuando el Latch es habilitado a través de un pulso en E0, la salidas Q0...Q3 del latch reflejan los valores del bus de datos conectado a la entradas D0...D3. Cuando el latch se inhabilita, con E0=0, el ultimo valor ingresado al latch es mantenido por sus salidas Q0...Q3.
De esta manera se polariza los transistores conectados a las salidas Q0...Q3, activando los reles K2...K5, según sea el ultimo valor ingresado al latch. De esta forma se mantiene el ultimo estado de los **dispositivos ON/OFF .**
Como el latch 4042 memoriza el estado de los dispositivos ON/OFF hasta que el suministro eléctrico deje de estar presente, el software de control del UCT debe actualizar el ultimo estado de los dispositivos ON/OFF al energizarse el UCT. Es por ello que se debe mantener una copia en la EEPROM del PIC16F84A del ultimo estado de los dispositivos ON/OFF.
Los reles se comportan como interruptores y pueden manejar corrientes de hasta 8A a 120Vac o 15ª a 120Vac según que tipo de rele de compre.

3.6.1 BUS DCE (PUERTO DE COMUNICACIONES DEL UCT):

El Bus DCE fue pensado para expandir dinámicamente a través de módulos electrónicos el hardware del UCT.
Es por ello que se lo diseño de tal forma que se pudiera establecer una comunicación bidireccional entre el UCT (Unidad Controlada Telefónicamente) y un DCE (Dispositivo de Control Externo).
Esta ingeniosa funcionalidad del UCT, permite la transferencia de datos digitales en ambos sentidos UCT/DCE, que deberá ser controlada por el software del PIC16F84A.

HARDWARE DEL BUS DCE:

El bus DCE provee que las siguientes líneas sean accesible por un DCE:

- **D0...D3:** Líneas de entrada / salida de datos, que acceden al bus de datos RB[4...7] del PIC16F84A.
- /**DCEINT:** Permite interrumpir a un DCE conectado al BUS DCE.
- /**UCTINT:** Permite interrumpir al UCT desde un DCE conectado al BUS DCE.
- **AUDIO-1 y AUDIO-2:** Entrada / salida de audio telefónico en forma diferencial disponible para ser accedida por el DCE.
- **+5V:** Tensión de 5Vcc disponible para alimentar un DCE.
- **+12V:** Tensión de 12Vcc disponible para alimentar un DCE.
- **GND:** Tierra o masa de la fuente de alimentación del UCT.

Internamente, el integrado **4066 (IC9)** separa el bus de datos RB[4...7] con las líneas de datos del bus DCE, D0...D3. El PIC16F84A, decide cuando habilitar el 4066 con RA3 y hacer accesible el bus de datos RB[4...7] en el BUS DCE.

SOFTWARE DEL UCT PARA EL BUS DCE:

El bus DCE fue pensado para funcionar en:

Modo desconectado:

Este modo permite la transferencia de datos en forma cruda desde el UCT al DCE en forma unidireccional.
Esto significa, que los datos se transfieren sin ningún tipo de protocolo y en paralelo a través de D0...D3. Útil para manejar circuitos simples conectados al BUS DCE.
El funcionamiento es muy simple:

El usuario debe acceder desde el teléfono al Bus DCE y elegir la opción "**DISPOSITIVO EXTERNO"** (ver **sección 2.2.2.4**).

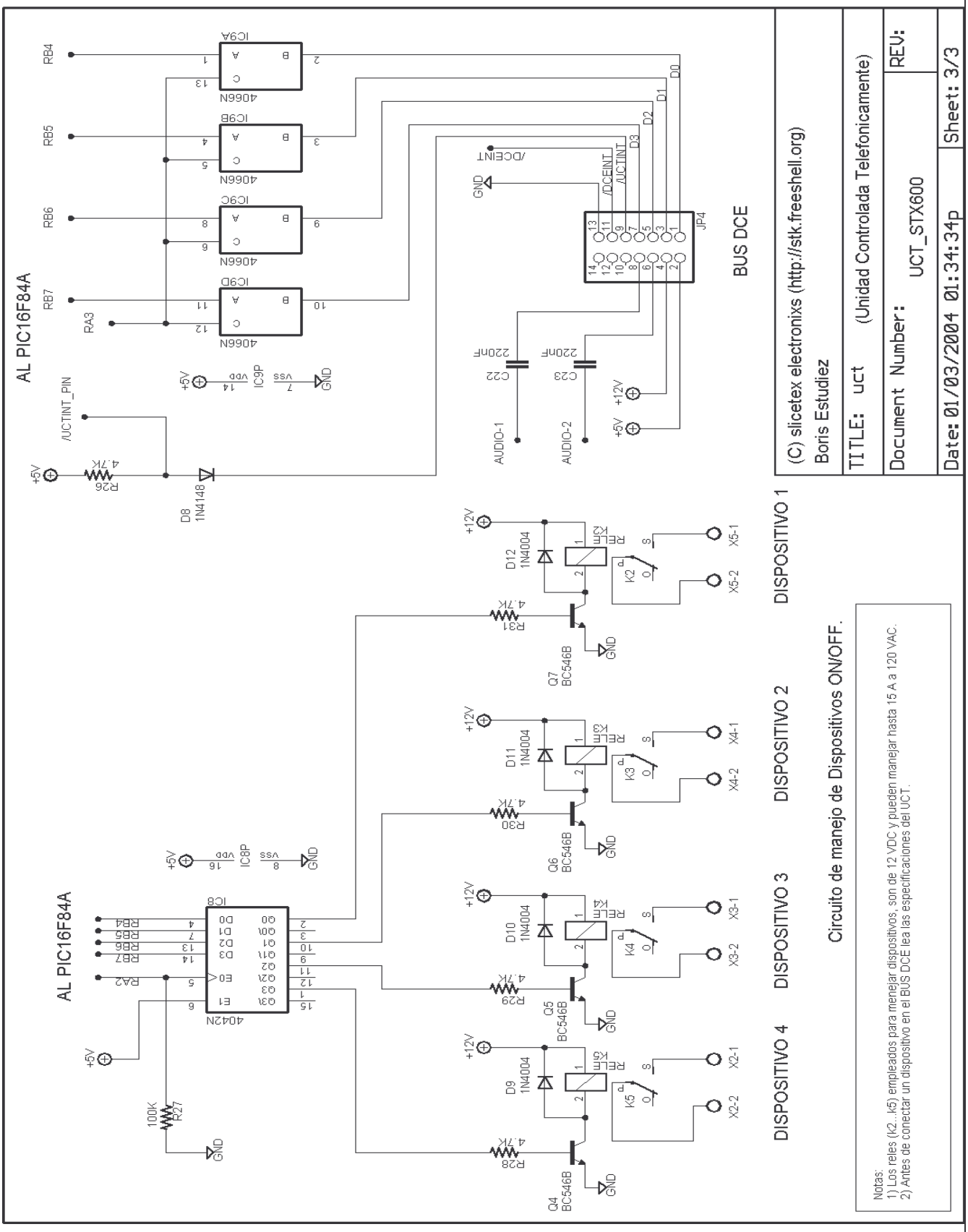
Una vez elegida esta opción se debe ingresar un numero. Una vez ingresado el numero a través del teléfono, se lo envía al BUS DCE en formato binario y se lo mantiene durante 5 mili-segundos. Al mismo tiempo, se pone en nivel bajo la línea /DCEINT durante el mismo periodo de tiempo.
Esto permite al usuario enviar los números 0...9 en forma binaria al bus DCE.
La siguiente tabla muestra lo dicho anteriormente:

NUMERO INGRESADO TELEFONICAMENTE	BUS DCE				
	D3	**D2**	**D1**	**D0**	**/DCEINT**
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
NINGUNO	**Z**	**Z**	**Z**	**Z**	**1**

Z=Alta impedancia.

Como el bus fue pensado para trabajar con un protocolo, si **/UCTINT** esta a nivel bajo al momento de enviar un numero al BUS DCE, el UCT responderá con un mensaje de error y el numero no se enviara.
Ello es debido a que el BUS DCE podría estar ocupado por un DCE que se esta tratando de comunicar con el UCT.

Modo Conectado:
Aquí la transferencia de datos es bidireccional y controlada por un protocolo previamente convenido en ambas partes (UCT y DCE).
En este modo se podrá acceder a muchas funcionalidades del UCT, como por ejemplo a su memoria de mensajes o ocupar la línea telefónica a través de una orden del DCE.
Actualmente esta funcionalidad por cuestiones de tiempo del proyecto, no fue implementada. Pero en un futuro estará disponible.


AL PIC16F84A
BUS DCE
DISPOSITIVO 1
DISPOSITIVO 2
DISPOSITIVO 3
DISPOSITIVO 4
Circuito de manejo de Dispositivos ON/OFF.
(C) slicetex electronixs (http://stk.freeshell.org)
Boris Estudiez
TITLE: uct (Unidad Controlada Telefonicamente)
Document Number: UCT_STX600
REV:
Date: 01/03/2004 01:34:34p
Sheet: 3/3
Notas:
1) Los reles (k2...k5) empleados para menejar dispositivos, son de 12 VDC y pueden manejar hasta 15 A a 120 VAC.
2) Antes de conectar un dispositivo en el BUS DCE lea las especificaciones del UCT.

3.7: Errores:

- El latch 4042 IC8 usado para el control de dispositivos ON/OFF, no tiene establecidas las condiciones iniciales de sus salidas cuando el UCT se energiza por primera vez, por lo tanto pueden activarse por un tiempo de 2 a 75 ms los reles K2...K5 de los dispositivos ON/OFF. Ello sucede solamente cuando el UCT se energiza, pero no cuando se resetea.

4.0: Notas de Construcción:

Esta sección pretende ser una guía general para construir al UCT. Lógicamente existen muchos caminos para llegar al mismo resultado. Usted debe tener experiencia en este tipo de circuitos ya que explicare los pasos generales:

1) Obtener los componentes electrónicos.

Luego de obtener los componentes electrónicos del proyecto, aconsejo comprar zócalos para la ISD1420, PIC16F84A, CD4066 (IC9) y el opto acoplador 4N25, que puede ser útil cambiarlos una vez que estén en la plaqueta.
También puede comprar un transformador de 220 Vca a 12Vca de 500 mA para poder alimentar al UCT.
Gabinetes, borneras, cables, etc a Gusto.
Con respecto al Bus DCE yo utilice cable IDC de 14 hilos con su respectivos conectores hembras en sus puntas y en la plaqueta utilice 7 pin-headers o jumpers, que en total hacen 14 conectores machos. Luego en el gabinete puse un DB15 para dejarlo mas presentable.... pero eso es relativo.
Los interruptores S1, S2 y leds son a gusto, siempre y cuando sean los correctos para el circuito.

2) Grabar Mensajes de Audio y Software .

En la pagina del proyecto (ver sección **0.2**) usted debe bajar las ultimas versiones disponible del:

Software:
Este contiene las instrucciones en assembler del PIC16F84A, que usted debe ensamblar (puede usar el MPLAB de Microchip) y grabárselas al PIC16F84A mediante un programador (puede usar el programador JDM).

Mensajes de Audio:
Los mensajes de audio están en formato **.wav** y deben ser grabados en el orden correcto (según se especifica en uct.inc) a la memoria ISD1420.
Luego estos mensajes de audio serán reproducidos a través de la línea telefónica por el UCT.
Debido a que no encontré ningún grabador gratuito para este tipo de memorias decidí construirme uno.
El mismo se llama **RAP-ISD** y funciona para GNU/LINUX. Puede bajarse en la pagina del proyecto (ver sección **0.2**).

3) PCB

El PCB es el circuito impreso del UCT y sirve para hacer la plaqueta.
En la pagina del proyecto esta disponible el PCB a escala y en formato PDF.

La plaqueta electrónica del UCT es en doble Faz (ambas caras de la plaqueta tienen pistas electrónicas) y su dimensiones son de: 135.89 mm x 201.93 mm. Recomiendo realizarla en una plaqueta de 15 cm x 21 cm para tener suficiente espacio y de material fibra de vidrio, para mantener buenas prestaciones del circuito con el paso del tiempo.

Con respecto a la construcción usted puede mandarme las dudas especificas por e-mail, pero no aceptare preguntas del tipo: Como Grabo el PIC? Como grabo la ISD1420? Como realizo el PCB?, etc... igualmente este tipo de preguntas generales se pueden aplicar al resto del documento (RTFM!).

5.0: Apéndice

5.1: Aplicación simple del BUS DCE

Puede utilizar un CD4042 (Latch D) y conectar sus entradas al bus DCE de la sig. Forma:

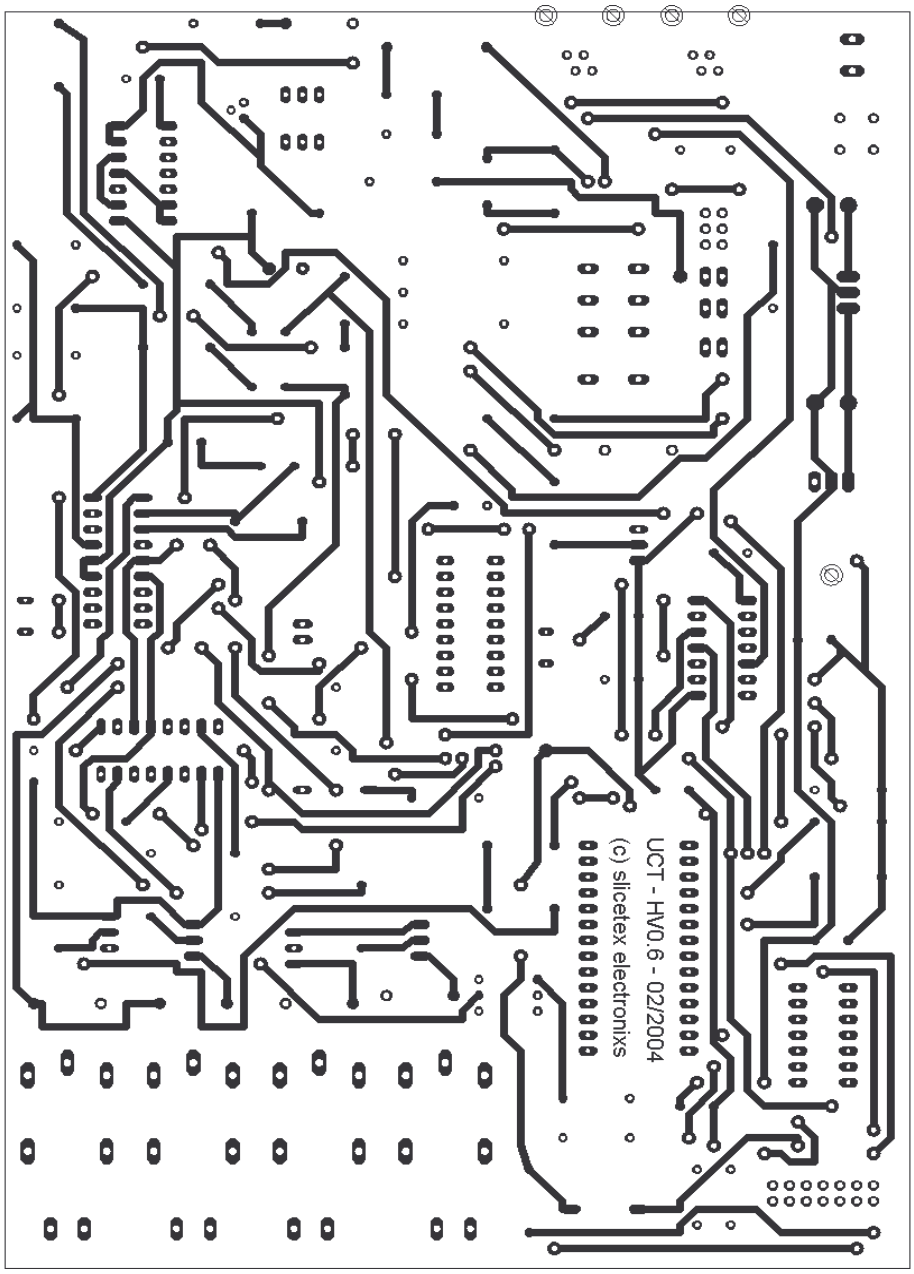
Entrada 4042	A salida del Bus DCE
D0	D0
D1	D1
D2	D2
D3	D3
E0	/DCEINT
E1	+5V
VDD	+5V
VSS	GND

Luego las salidas del CD4042, Q0, Q1, Q2 y Q3 se conectan individualmente a leds (con una resistencia de 470 ohms en serie).
De esta forma podrá visualizar en los leds el valor de los números binarios de los DTMF enviamos en modo desconectado al Bus DCE.
El latch es necesario para mantener los números, ya que como se menciono en el documento, solo están presente por 5 ms.

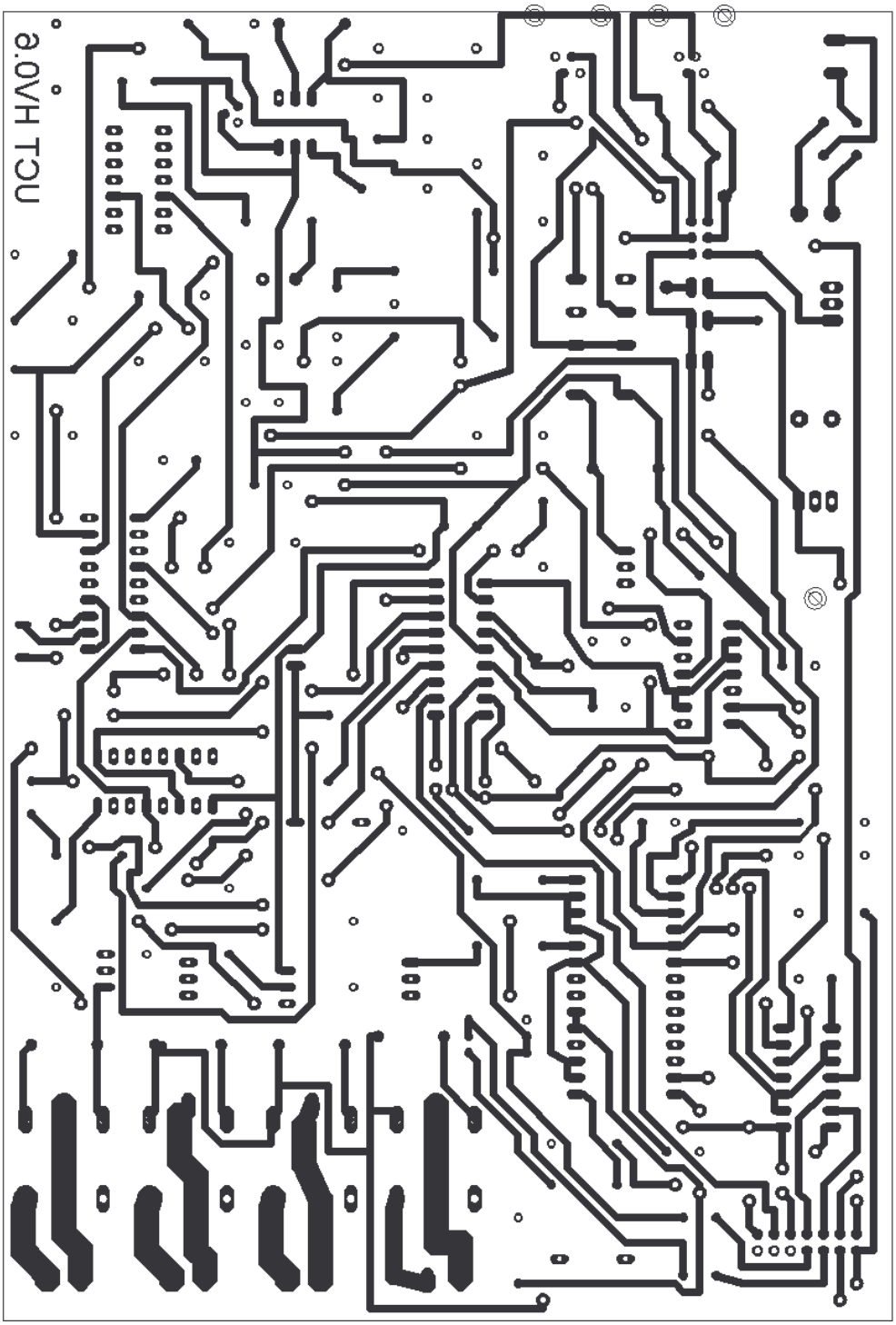
Usted se dará cuenta en este paso como puede expandir con lógica combinacional al UCT a través del Bus DCE. Supóngase que quiere activar 10 reles mas, solamente debe agregar un decodificador de 4 a 16 líneas en esta etapa.
Recuerde que también tiene las salidas / entradas Analógicas AUDIO-1 y AUDIO-2.

5.2: PCB (PRINTED BOARD CIRCUIT) del UCT

5.2.1 TOP VIEW



5.2.2 BOTTOM VIEW



NOTA: El PCB no esta a escala. Bajarlo desde la pagina del proyecto. En formato PDF.

5.3: Software del PIC16F84A usado en el UCT

Debe bajarlo de la pagina del proyecto (ver sección 0.2). Aunque igualmente los adjuntamos en el documento.

ATENCIÓN: Quizás no es la ultima versión disponible y puede contener menos funcionalidades y mas errores que la disponible en la pagina web.
Ello es debido a que no actualizare el documento con la misma frecuencia del software. Por otro lado para imprimirlos, se visualiza mejor en los archivos de la pagina web, ya que al transferirlo a este tipo de documento los espacios en blanco entre las líneas del programa se desordenan. Pero solo es una cuestión visual, el programa debería funcionar de forma correcta si se lo copia.

Archivo: UCT.ASM (codigo principal)
Archivo: UCT.INC (constantes, variables, macros, etc de UCT.ASM).

[ARCHIVO: UCT.ASM]

```
; asmsyntax=pic16f84
;
; File        : uct.asm
; MCU         : PIC-16F84A @ 4 MHz
; Date        : 18/09/2003
; Update      : 30/01/2004
; Version     : 0.5.1
; Author      : Boris Estudiez.
;
; DESCRIPCION:
; Software de control del UCT (Unidad Controlada por Telefono).
;
; HARDWARE VERSION: UCT HV-0.5 o superior.
;
; (c) Copyright slicetex electronixs (2001-2004)
;     http://stk.freeshell.org
;

   LIST P=PIC16F84A
   #include <p16f84a.inc>
   #include "uct.inc"

   __CONFIG _PWRTE_ON & _WDT_OFF & _XT_OSC & _CP_OFF

;*****************************************************************
;*             INICIO DEL CODIGO DE PROGRAMA                     *
;*****************************************************************
RESET: ORG 0x00
   goto SETUP

;*****************************************************************
;*        RUTINAS DE ATENCION A INTERRUPCIONES                   *
;*****************************************************************
RSI: ORG 0x04

   bcf INTCON, GIE       ;Inhibir toda interrupcion
   call push_STATUS      ;Salvamos registro STATUS y W.
   call push_w
```

```
   btfsc INTCON, INTF    ;Int. externa?
     goto int_ext       ;Si, determinar tipo de int. externa.
   btfsc INTCON, T0IF    ;Int. TMR0 Overflow?
     goto RSITMR0       ;Si, atender int. por TMR0.
   goto end_RSI          ;No, devolver control a programa.

int_ext:
   btfss UCTINT_PIN      ;Interrupcion por BUS DCE ?
     goto RSI_DCE       ;Si, int. por BUS DCE. -> UCTINT_PIN=0.
   goto RSILL            ;No, int. por Detector de llamada. Atender.

end_RSI:
   call pop_w            ;Reponer W y STATUS.
   call pop_STATUS
   retfie                ;Devolver control.

;RSILL: Rutina de Servicio a Interrupcion por LLamada
;Modifica: RINGS_LEFT, TMR0 y UCT_STATUS.
RSILL:
   bcf INTCON, INTF          ;INTF=0, reponer flag.

   bsf UCT_STATUS, RINGING    ;Intento de llamada. RINGING=1
   movlw S10
   call set_timeout          ;Seteo timeout en 10 segundos.
   decf RINGS_LEFT, F        ;RINGS_LEFT--

end_RSILL:
   goto end_RSI              ;Devolver control a programa.

;RSITMR0: Rutina de Servicio a Interrupcion por TMR0 Overflow.
;Dependiendo de los valores de los bits del UCT_OPTION toma
;una accion. Este servicio solo puede ser usado con
;la funcion set_timeout().
;SI:
;QTIMEOUT-> 1: vuelve a stand-by ; 0:vuelve al programa.
;
;Setea con 1 el flag TIMEOUT de UCT_STATUS al producirce un timeout.
;Nota: esta rutina deja seteado el Banco 0.

RSITMR0:
   bcf INTCON, T0IF          ;T0IF=0, Reponer flag.
```

```
   BANK_0
   movlw 0x06              ;Recordar que TMR0 esta en el banco 0.
   movwf TMR0              ;Proxima interrupcion en 250 cuentas.

   decfsz TMR0_AUX, F      ;TMR0_AUX == 0?, paso 1 segundo?
     goto end_RSITMR0      ;No, seguir esperando.
   movlw d'125'            ;Si cargar TMR0_AUX con 125 y decrementar
   movwf TMR0_AUX          ;SECONDS.
   decfsz SECONDS, F       ;SECONDS==0? TIMEOUT?
     goto end_RSITMR0      ;No, seguir esperando.
   bsf UCT_STATUS, TIMEOUT ;Si, setear TIMEOUT.
   btfsc UCT_OPTION, QTIMEOUT ;QTIMEOUT==1?
     call close_call       ;Si, inicializar UCT -> volver a stand-by.

end_RSITMR0:
   goto end_RSI            ;devolver control.

;RSI_DCE: Rutina de Servicio a Interrupcion por bus DCE.
;NOTA: Esta funcionalidad no esta habilitada aun en el UCT.
;No poner el pin UCTINT a cero en el bus DCE hasta que
;sean completadas las rutinas del BUS DCE.
RSI_DCE:
   bcf INTCON, INTF        ;INTF=0, reponer flag.
   bsf UCT_STATUS, DCEINT  ;Int. por bus DCE. DCEINT=1.

end_RSI_DCE:
   goto end_RSI            ;Devolver control a programa.

;*********************************************************************
;*                  RUTINA PRINCIPAL DEL UCT                         *
;*********************************************************************

;Notas de programacion:
; * Banco 0 se supone siempre por defecto, si se cambia a
;   banco 1 en una rutina, luego al salir, volver a poner a 1.
; * QTIMEOUT se supone siempre 1, si se pone a 0 en una rutina
;   luego al salir de rutina volver a poner a 1.

;Seteamos Condiciones Iniciales de Funcionamiento.
SETUP:
```

```
   clrf INTCON       ;Impedimos toda interrupcion inicialmente.

   BANK_1            ;accedemos al banco 1 de la mem. ram
   ;Config. Puerto A
   movlw b'00010000' ;A7-5=X,A4=E,A3=S,A2=S,A1=S,A0=S
   movwf TRISA
   ;Config. Puerto B
   movlw b'00000011' ;B7-4=S,B3=S,B2=S,B1=E,B0=E
   movwf TRISB

   ;Conf. OPTION_REG Reg.
   movlw b'1000111'
   movwf OPTION_REG   ;RBPU=OFF PSA=ON->TMR0 DIV=111, INTEDG=0
FALL EDGE.

   BANK_0            ;accedemos al banco 0 de la mem. ram
   ;Valor inicial Puerto A
   ;A0 | A1 | A2 | A3 | Poner Dispositivo en el bus RB4-7:
   ; 1 |  0 |  0 |  0 |8870 (DTMF DECODER)
   ; 0 |  1 |  0 |  0 |ISD1420/4016 (Memoria De Voz)
   ; 0 |  0 |  1 |  0 |4042 (LATCH/DISP. ON OFF)
   ; 0 |  0 |  0 |  1 |BUS DCE/4016
   ; 0 |  0 |  0 |  0 |Ningun Dispositivo.
   clrf PORTA        ;Limpiamos Latch's de puerto A, Salidas=0.

   ;Valor inicial Puerto B
   ;Telefono/Linea Desocupado.
   clrf PORTB        ;Limpiamos Latch's de puerto B, Salidas=0
   bsf DCEINT_PIN    ;DCEINT_PIN=1 -> RB2=1.

   ;Actualizamos estado de Dispositivos ON/OFF.
   movlw 0x05        ;W > 4.
   call ctrl_disp    ;Actualizar Latch 4042. Ver ctrl_disp().

   ;Borramos registro de estado y opciones del UCT
   clrf UCT_STATUS
   clrf UCT_OPTION           ;SKIPMENU=0, QMSG=0.
   bsf  UCT_OPTION, QTIMEOUT ;QTIMEOUT=1 -> salir si hay timeout.
                             ;Por lo general siempre esta activado.
   ;Seteamos variables
```

```
   movlw EE_RINGS
   call eeprom_read     ;W=*EE_RINGS
   movwf RINGS_LEFT          ;Numero de RINGS que faltan para contestar
llamado.

   ;Activar Interrupcion solo externa por RB0/INT
   movlw b'10010000'
   movwf INTCON      ;GIE=1, INTE=1

;standby_code: Mantiene al UCT en standby a la espera de algun evento
;externo.
standby_code:
   sleep                ;Permanecer en Stand-by
   ;clrwdt              ;Limpiamos el Watchdog Timer
   nop                  ;Ejecutada al salir de sleep.

   btfsc UCT_STATUS, RINGING ;Intento de llamada? RINGING==1
     goto  wait_rings       ;Si, esperar RINGS.
   btfsc UCT_STATUS, DCEINT  ;Interrupcion por DCE? DCEINT==1
     goto dce_connection    ;Si, establecer coneccion con DCE
   goto  standby_code       ;No, permanecer en stand-by.

wait_rings:
      ;Esperamos a que RINGS_LEFT sea cero, antes de atender llamado.
      movf RINGS_LEFT, F
      btfss STATUS, Z      ;RINGS_LEFT == 0?
        goto wait_rings    ;No, segir esperando.
      call unset_timeout   ;Si, sacamos TIMEOUT introducido por RSILL
      goto reply_call      ;y Respondemos Llamada.


;*********************************************************************
;*     RUTINAS DE ATENCION DE LLAMADO DEL UCT       *
;*********************************************************************

;reply_call: establece la comunicacion entre el UCT y el usuario
;que llama al UCT.
reply_call:

   BANK_0
   bcf INTCON, INTE          ;Inhibir int. externas, bus DCE inhibido.
```

```
  bcf UCT_STATUS, RINGING     ;RINGING=0
  bsf UCT_STATUS, ANSWERCALL  ;ANSWERCALL=1 - UCT respondiendo llamada.
  bsf UCT_OPTION, QMSG        ;QMSG=1, Desconectar con mensage de aviso.

  bsf PORTB, 3      ;RB3=1, Ocupar linea, al circ. selector de modo.
  movlw S2
  call spause       ;Esperar aprox. 2 segundos antes de contestar

  ;Pedimos clave de acceso de 4 digitos. y damos 3 oportunidades.
  movlw 3
  movwf cx          ;cx = 3

for_passwd:
    movlw ISD_INGRESAR
    call play_isd       ;reproducir "INGRESAR"
    movlw ISD_CLAVE
    call play_isd       ;reproducir "CLAVE"

    movlw 4
    call get_dtmfstr    ;Tomar 4 digitos DTMF y almacenar en AX:BX.
    movlw EE_PASSW_0    ;Comprobamos password.
    call eeprom_read
    xorwf ax, W
    btfss STATUS, Z     ;ax==EE_PASSW_0? (digito 3 y 2)
      goto wrong_pass   ;No, passwd incorrecto.
    movlw EE_PASSW_1
    call eeprom_read
    xorwf bx, W
    btfss STATUS, Z     ;bx==EE_PASSW_1?  (digito 1 y 0)
      goto wrong_pass   ;No, passwd incorrecto.
    goto end_for_passwd  ;Passwd Correcto.

wrong_pass:
      movlw ISD_CLAVE
      call play_isd       ;"CLAVE"
      movlw ISD_INCORRECTO
      call play_isd       ;"INCORRECTO"

      decfsz cx, F        ;Pedir de nuevo clave?
```

```
        goto for_passwd   ;Si. Uhh :)
end_for_passwd:

  movf cx, F
  btfsc STATUS, Z     ;Desconectar? cx==0?
    call close_call  ;Si, Clave ingresada mal 3 veces, desconectar :(

  bcf UCT_OPTION, QTIMEOUT      ;QTIMEOUT=0
  movlw S2
  call getdtmf                  ;Esperar 2 segundos para DTMF_0.
  bsf UCT_OPTION, QTIMEOUT      ;QTIMEOUT=1

  xorlw 0x00
  btfsc STATUS, Z               ;DTMF_0 recibido?
    bsf UCT_OPTION, SKIPMENU   ;Si, UCT_OPTION/SKIPMENU=1
  movlw ISD_BIENVENIDO
  call play_isd                 ;"BIENVENIDO"

;Rutina de Menu principal.
uct_main_menu:

  btfsc UCT_OPTION, SKIPMENU    ;SKIPMENU==1?
    goto get_menu_option       ;Si, omitir mensages del main_menu.
  movlw ISD_UNO
  call play_isd          ;"UNO"
  movlw ISD_CONTROL_DE
  call play_isd          ;"CONTROL DE"
  movlw ISD_DISPOSITIVO
  call play_isd          ;"DISPOSITIVO"
  movlw ISD_DOS
  call play_isd          ;"DOS"
  movlw ISD_VER_ESTADO_DE
  call play_isd          ;"VER ESTADO DE"
  movlw ISD_DISPOSITIVO
  call play_isd          ;"DISPOSITIVO"
  movlw ISD_TRES
  call play_isd          ;"TRES"
  movlw ISD_BUS_DCE
  call play_isd          ;"BUS_DCE"
  movlw ISD_CUATRO
  call play_isd          ;"CUATRO"
```

```
   movlw ISD_CONFIGURAR
   call play_isd
   movlw ISD_RINGS
   call play_isd          ;"CONFIGURAR RINGS"
   movlw ISD_CINCO
   call play_isd          ;"CINCO"
   movlw ISD_CAMBIAR
   call play_isd          ;"CAMBIAR"
   movlw ISD_CLAVE
   call play_isd          ;"CLAVE"
   call play_sal_rep      ;"NUMERAL SALIR, ASTERISCO REPETIR".

   ;Tomamos DTMF y ejecutamos lo que corresponda.
get_menu_option:
   movlw S60              ;Tiempo de espera maximo para getdtmf.
   call getdtmf
   movwf ax               ;ax=DTMF recibido.

   movf ax, F
   btfsc STATUS, Z        ;ax==DTMF_0 ?
     goto skip_menu_msg   ;Si, des/activar UCT_OPTION/SKIPMENU
   movlw 0x01
   xorwf ax, W
   btfsc STATUS, Z        ;ax==DTMF_1 ?
     goto ctrl_disp_menu  ;Si, ir a control de dispositivos.
   movlw 0x02
   xorwf ax, W
   btfsc STATUS, Z        ;ax==DTMF_2 ?
     goto disp_stat_menu  ;Si, ver estado de dispositivos.
   movlw 0x03
   xorwf ax, W
   btfsc STATUS, Z        ;ax==DTMF_3 ?
     goto bus_dce_menu    ;Si, ir a BUS DCE.
   movlw 0x04
   xorwf ax, W
   btfsc STATUS, Z        ;ax==DTMF_4 ?
     goto conf_ring_menu  ;Si, ir menu de configurar RINGS.
   movlw 0x05
   xorwf ax, W
   btfsc STATUS, Z        ;ax==DTMF_5 ?
     goto chpassw_menu    ;Si, ir a menu de cambio de clave.
```

```
   movlw DTMF_AST
   xorwf ax, W
   btfsc STATUS, Z          ;ax==DTMF_* ?
     goto uct_main_menu     ;Si, repetir main menu.
   movlw DTMF_NUM
   xorwf ax, W
   btfsc STATUS, Z          ;ax==DTMF_# ?
     call close_call        ;Si, desconectar UCT.

   call play_incorrecto     ;"NUMERO INCORRECTO"
   goto uct_main_menu       ;Volver a reproducir el menu principal.

;skip_menu_msg: des/activar UCT_OPTION/SKIPMENU
skip_menu_msg:
     btfsc UCT_OPTION, SKIPMENU    ;SKIPMENU==0? Desactivado?
       goto skipmenu_off           ;No, desactivar.
     bsf UCT_OPTION, SKIPMENU      ;Si, activar.
     goto uct_main_menu
skipmenu_off:
     bcf UCT_OPTION, SKIPMENU      ;Desactivar.
     goto uct_main_menu

;ctrl_disp_menu: menu de control ON/OFF de dispositivos
ctrl_disp_menu:
     btfsc UCT_OPTION, SKIPMENU    ;SKIPMENU==1?
       goto get_disp               ;Si, omitir descripcion del menu.
     movlw ISD_INGRESAR
     call play_isd                 ;"INGRESAR"
     movlw ISD_DISPOSITIVO
     call play_isd                 ;"DISPOSITIVO"

get_disp:                  ;Esperamos numero de dispositivo.
     movlw S60
     call getdtmf
     movwf ax              ;ax=Numero de Dispositivo.

     movf ax, F
     btfsc STATUS, Z       ;ax==0 ?
       goto wrong_disp     ;Si, dispositivo incorrecto.
     movlw 5
     subwf ax, W           ;w=ax-5
```

```
      btfss STATUS, C         ;ax < 5 ?
        goto on_off_disp      ;Si, des/activar dispositivo.
      movlw DTMF_AST
      xorwf ax, W
      btfsc STATUS, Z         ;ax==DTMF_* ?
        goto ctrl_disp_menu   ;Si, repetir menu.
      movlw DTMF_NUM
      xorwf ax, W
      btfsc STATUS, Z         ;ax==DTMF_# ?
        goto uct_main_menu    ;Si, volver a menu principal.

wrong_disp:
      call play_incorrecto    ;"NUMERO INCORRECTO"
      goto ctrl_disp_menu     ;Repetir menu.

on_off_disp:
      movlw S60
      call getdtmf            ;Esperamos accion a tomar,
      movwf bx                ;activar o desactivar dispositivo?

      movlw 0x01
      xorwf bx, W
      btfsc STATUS, Z         ;bx==DTMF_1 ?
        goto on_disp          ;Si, activar dispositivo.
      movf bx, F
      btfsc STATUS, Z         ;bx==DTMF_0 ?
        goto off_disp         ;Si, desactivar dispositivo.
      movlw DTMF_NUM
      xorwf bx, W
      btfsc STATUS, Z         ;bx==DTMF_# ?
        goto ctrl_disp_menu   ;Si, volver a pedir dispositivo.
      movlw DTMF_AST
      xorwf bx, W
      btfsc STATUS, Z         ;bx==DTMF_* ? (No tiene sentido repetir)
        goto ctrl_disp_menu   ;Si, volver a pedir dispositivo.

      call play_incorrecto    ;"NUMERO INCORRECTO"
      goto on_off_disp        ;Pedir de vuelta accion.

on_disp:
        bsf ax, 7
```

```
        movf ax, W              ;w=1000xxxx
        call ctrl_disp          ;activar dispositivo.
        movlw ISD_DISPOSITIVO
        call play_isd           ;"DISPOSITIVO"
        movlw ISD_ACTIVADO
        call play_isd           ;"ACTIVADO"
        goto ctrl_disp_menu     ;Repetir Menu.

off_disp:
        movf ax, W              ;w=0000xxxx
        call ctrl_disp          ;desactivar dispositivo.
        movlw ISD_DISPOSITIVO
        call play_isd           ;"DISPOSITIVO"
        movlw ISD_DESACTIVADO
        call play_isd           ;"DESACTIVADO"
        goto ctrl_disp_menu     ;Repetir Menu.


;disp_stat_menu: muestra estado de los dispositivos conectados.
disp_stat_menu:
     movlw EE_DISPSTAT
     call eeprom_read
     movwf ax                   ;ax=estado de dispositivos.

     movlw ISD_DISPOSITIVO
     call play_isd              ;"DISPOSITIVO"

     movlw ISD_UNO
     call play_isd              ;"UNO"
     btfss ax, 0                ;Disp. 1 activado ?
     goto d1_off                ;No.
       movlw ISD_ACTIVADO       ;Si.
       call play_isd            ;"ACTIVADO"
       goto d1_end
d1_off:
       movlw ISD_DESACTIVADO
       call play_isd            ;"DESACTIVADO"
d1_end:

     movlw ISD_DOS
     call play_isd              ;"DOS"
```

```
    btfss ax, 1             ;Disp. 2 activado?
    goto d2_off             ;No.
      movlw ISD_ACTIVADO    ;Si.
      call play_isd         ;"ACTIVADO"
      goto d2_end
d2_off:
      movlw ISD_DESACTIVADO
      call play_isd         ;"DESACTIVADO"
d2_end:

    movlw ISD_TRES
    call play_isd           ;"TRES"
    btfss ax, 2             ;Disp. 3 activado?
    goto d3_off             ;No.
      movlw ISD_ACTIVADO    ;Si.
      call play_isd         ;"ACTIVADO"
      goto d3_end
d3_off:
      movlw ISD_DESACTIVADO
      call play_isd         ;"DESACTIVADO"
d3_end:

    movlw ISD_CUATRO
    call play_isd           ;"CUATRO"
    btfss ax, 3             ;Disp. 4 activado?
    goto d4_off             ;No.
      movlw ISD_ACTIVADO    ;Si.
      call play_isd         ;"ACTIVADO"
      goto d4_end
d4_off:
      movlw ISD_DESACTIVADO
      call play_isd         ;"DESACTIVADO"
d4_end:
    goto uct_main_menu      ;Volver a menu Princial.

;bus_dce_menu: Menu para interactuar con un dispositivo conectado
;al bus dce.
;OPCION UNO: Desactivada. Protocolo no terminado aun.
bus_dce_menu:
    btfsc UCT_OPTION, SKIPMENU  ;SKIPMENU==1?
      goto get_bus_dce_opt      ;Si, omitir descripcion del menu.
```

```
      movlw ISD_UNO
      call play_isd              ;"UNO"
      movlw ISD_CONECTAR
      call play_isd              ;"CONECTAR"
      movlw ISD_DOS
      call play_isd              ;"DOS"
      movlw ISD_DISPOSITIVO
      call play_isd              ;"DISPOSITIVO"
      movlw ISD_EXTERNO
      call play_isd              ;"EXTERNO"

get_bus_dce_opt:
      movlw S60
      call getdtmf
      movwf ax              ;ax=opcion.

      movlw 0x01
      xorwf ax, W
      btfsc STATUS, Z       ;ax == 1?
        goto dce_connect    ;Si, Conectar con DCE.
      movlw 0x02
      xorwf ax, W
      btfsc STATUS, Z       ;ax == 2?
        goto send_dtmf2dce  ;Si, enviar codigos DTMF del 0-9 al DCE.
      movlw DTMF_AST
      xorwf ax, W
      btfsc STATUS, Z       ;ax = DTMF_*?
        goto  bus_dce_menu  ;Si, Repetir Menu.
      movlw DTMF_NUM
      xorwf ax, W
      btfsc STATUS, Z       ;ax = DTMF_#?
        goto uct_main_menu  ;Si, Volver a menu Princial.

;dce_connect: Conecta con un dispositivo usando el bus DCE.
;La conexion se efectua mediante un protocolo.
;Esta funcionalidad NO esta implementada aun en el UCT.
dce_connect:
      movlw ISD_ERROR     ;Reproducir error y volver a menu anterior.
      call play_isd
      movlw ISD_CONECTAR
      call play_isd
```

```
      movlw ISD_BUS_DCE
      call play_isd
      goto bus_dce_menu

;send_dtmf2dce:
;Solo envia los codigos binarios DTMF del 0 al 9 al bus DCE, que
;el abonado remoto solicite.
;En este modo solo hay conexion unidireccional desde el UCT
;al DCE.
send_dtmf2dce:
      btfsc UCT_OPTION, SKIPMENU    ;SKIPMENU==1?
        goto get_dtmf_code          ;Si, omitir descripcion del menu.
      movlw ISD_INGRESAR
      call play_isd                 ;"INGRESAR"
      movlw ISD_NUMERO
      call play_isd                 ;"NUMERO"

get_dtmf_code:
      movlw M04                 ;Esperar codigo DTMF por 4 minutos.
      call getdtmf
      movwf ax                  ;ax=codigo dtmf.

      movlw DTMF_AST
      xorwf ax, W
      btfsc STATUS, Z           ;ax = DTMF_*?
        goto  send_dtmf2dce     ;Si, Repetir Menu.
      movlw DTMF_NUM
      xorwf ax, W
      btfsc STATUS, Z           ;ax = DTMF_#?
        goto bus_dce_menu       ;Si, Volver a menu anterior.

      ;Enviamos codigo DTMF al DCE.

      movf ax, W                ;W=ax
      call bdce_write           ;Escribimos en bus DCE y nos fijamos
      xorlw ERR                 ;si la operacion fue exitosa.
      btfsc STATUS, Z             ;W==ERR?
        goto send_dtmf2dce_err  ;Si, error al escribir dato.
      movlw ISD_LISTO           ;No, dato escrito con exito.
      call play_isd             ;"LISTO"
      goto send_dtmf2dce        ;Volver a pedir dtfm.
```

```
send_dtmf2dce_err:
     movlw ISD_BUS_DCE
     call play_isd            ;"BUS DCE"
     movlw ISD_ERROR
     call play_isd            ;"ERROR"
     goto send_dtmf2dce       ;Volver a pedir dtfm.

;conf_ring_menu: Menu para configurar el numero de rings que
;deben pasar antes del que UCT conteste un llamado.
;Si el usuario ingresa DTMF_0, EE_RINGS se establece a _LOT_RINGS.

conf_ring_menu:
     btfsc UCT_OPTION, SKIPMENU    ;SKIPMENU==1?
       goto get_rings              ;Si, omitir descripcion del menu.
     movlw ISD_INGRESAR
     call play_isd                 ;"INGRESAR"
     movlw ISD_RINGS
     call play_isd                 ;"RINGS"

get_rings:                   ;Esperamos cantidad de RINGS.
     movlw S60
     call getdtmf
     movwf ax                ;ax=Numero de RINGS.

     movf ax, F
     btfsc STATUS, Z         ;ax == 0?
       goto set_LOT_RINGS    ;Si, setar EE_RINGS = _LOT_RINGS
     movlw DTMF_AST
     xorwf ax, W
     btfsc STATUS, Z         ;ax = DTMF_*?
       goto  conf_ring_menu ;Si, Repetir Menu.
     movlw DTMF_NUM
     xorwf ax, W
     btfsc STATUS, Z         ;ax = DTMF_#?
       goto uct_main_menu    ;Si, Volver a menu Princial.

     goto set_EE_RINGS       ;No, Setear EE_RINGS = ax.

set_LOT_RINGS:
     movlw _LOT_RINGS
```

```
    movwf ax            ;ax = _LOT_RINGS

set_EE_RINGS:
    movlw EE_RINGS
    movwf ex
    movf ax, W
    call eeprom_write   ;EE_RINGS = ax

    movlw ISD_LISTO
    call play_isd       ;"LISTO"

    goto uct_main_menu  ;Volver a menu Princial.

;chpassw_menu: Menu para cambiar clave de acceso en el UCT.
;Por seguridad, primero debe ingresar DTMF UNO, y luego la
;nueva clave.

chpassw_menu:
    btfsc UCT_OPTION, SKIPMENU  ;SKIPMENU==1?
      goto get_chpassw_menu_option ;Si, omitir descripcion.
    movlw ISD_UNO
    call play_isd           ;"UNO"
    movlw ISD_CAMBIAR
    call play_isd           ;"CAMBIAR"
    movlw ISD_CLAVE
    call play_isd           ;"CLAVE"
    call play_sal_rep       ;"# SALIR, * REPETIR"


get_chpassw_menu_option:    ;Esperamos opcion.
    movlw S60
    call getdtmf
    movwf ax            ;ax=opcion.

    movlw 0x01
    xorwf ax, W
    btfsc STATUS, Z     ;ax == 1?
      goto set_passwd   ;Si, setear nuevo password.
    movlw DTMF_AST
    xorwf ax, W
    btfsc STATUS, Z     ;ax = DTMF_*?
```

```
        goto chpassw_menu    ;Si, Repetir Menu.
      movlw DTMF_NUM
      xorwf ax, W
      btfsc STATUS, Z        ;ax = DTMF_#?
        goto uct_main_menu   ;Si, Volver a menu Princial.

      call play_incorrecto
      goto chpassw_menu      ;Volver a repetir Menu.


set_passwd:
      movlw ISD_INGRESAR
      call play_isd          ;"INGRESAR"
      movlw ISD_CLAVE
      call play_isd          ;"CLAVE"

      movlw 0x04             ;Tomar 4 digitos DTMF y almacenar en
      call get_dtmfstr       ;AX:BX.

      movlw EE_PASSW_0       ;EE_PASSWD_0 = Digito 3 y 4
      movwf ex
      movf ax, W
      call eeprom_write      ;EE_PASSWD_0 = ax

      movlw EE_PASSW_1       ;EE_PASSWD_1 = Digito 1 y 2
      movwf ex
      movf bx, W
      call eeprom_write      ;EE_PASSWD_1 = bx

      movlw ISD_LISTO
      call play_isd          ;"LISTO"

      goto uct_main_menu     ;Volver a menu Princial.

;*********************************************************************
;*                RUTINAS DEL BUS DCE DEL UCT                        *
;*********************************************************************
;NOTA: Solo es posible enviar datos binarios del 0 al 9 por el
;      bus DCE, pero no recibir datos, ni usar el protocolo CTP.
;      El bus DCE aun no esta totalmente implementado.
```

```
;----> Rutinas del PROTOCOLO CTP (Command Transmision Protocol) <----
;NO DEFINIDO!

;----> RUTINAS DEL UCT PARA COMUNICARSE POR EL BUS DCE <----

;bdce_write(): escribe el nibble menos significativo
;de W en el BUS DCE.
;I:
; W  = valor a escribir en el bus DCE.
;O:
; W = OK  -> El dato fue escrito en el bus DCE.
; W = ERR -> El BUS DCE esta ocupado, dato no se escribe.
;Destruye: ex, W.
;NOTA:
;* El dato se mantiene 5 ms en el BUS DCE y DCEINT_PIN se mantiene
;  a 0 por el mismo periodo. Luego el bus queda en alta Z y
;  DCEINT_PIN=1.
bdce_write:

   movwf ex            ;ex=W

   BANK_1
   movlw 0x0F
   andwf TRISB, F      ;TRISB=0000xxxx -> RB4-7=S
   BANK_0

   ;Preparamos datos para escribir en el bus DCE.
   movlw 0x0F
   andwf PORTB, F      ;PORTB=0000xxxx
   swapf ex, F         ;ex=swap(ex)
   movlw 0xF0
   andwf ex, W         ;W=xxxx0000=AND(0xF0,ex)
   iorwf PORTB, F      ;Escribimos dato en RB4-7 del PORTB.

   ;Comprobamos si se puede escribir dato en BUS DCE.
   btfss UCTINT_PIN    ;Bus DCE ocupado? UCTINT_PIN == 0?
     retlw ERR         ;Si, retornar con W=ERR.

   ;Escribir dato.
   bsf PORTA, 3        ;Si, Habilitar 4016 (sacar de alta Z).
   bcf DCEINT_PIN      ;Ocupamos BUS DCE.
```

```
   movlw d'5'
   call mpause          ;Mantener dato por 5 ms.

   bsf DCEINT_PIN       ;Desocupamos BUS DCE. -> DCEINT_PIN=1.
   bcf PORTA, 3         ;Deshabilitar 4016 (poner en alta Z).

   movlw 0x0F
   andwf PORTB, F       ;PORTB=0000XXXX, Bus de datos a CERO.

   retlw OK             ;Ecritura finalizada, retornar con W=OK.

;dce_connection: establece conexion con DCE.
;Funcion no habilitada.
dce_connection:
   goto RESET           ;Resetear en caso de interrupcion por DCE.
;NO DEFINIDO!


;*********************************************************************
;*              RUTINAS GENERALES DEL UCT                            *
;*********************************************************************

;close_call(): Desconecta al UCT de la llamada actual.
close_call:

   btfss UCT_OPTION, QMSG  ;Reproducir mensages antes de salir?
      goto end_close_call  ;No, salir sigilosamente...
   movlw ISD_SALIR         ;Si, "SALIR".
   call play_isd           ;antes de desconectar.
   movlw S2
   call spause             ;Esperar 2 seg. entes de RESETEAR.

end_close_call:
   goto RESET           ;Good Nigth!

;get_dtmfstr: Toma una cadena de 4 codigos DTMF como maximo, usando
;la funcion getdtmf.
;I: W = Numero de codigos DTMF a tomar (max. 4)
;O: AX y BX almacenan los 4 codigo DTMF recibidos. Donde el nibble
;menos significativo de BX contiene el ultimo codigo DTMF recibido.
```

```
;Si W < 4 los codigos DTMF faltantes en AX y BX se completan con
;cero. TIMEOUT en 30 segundos si no se recive ningun cod. DTMF.
;Destruye: W
;Registros stack-eados: ax, cx, dx, ex

get_dtmfstr:

   ;ax y ex se stakea en getdtmf.
   call push_cx
   call push_dx

   clrf ax
   clrf bx              ;BX y AX Almacenan DMTF's recibido.
   movwf cx             ;Numero de digitos DTMF a tomar.
for_get_dtmfstr:
   movlw S60            ;timeout en 60 segundos para getdtmf
   call getdtmf         ;W = codigo DTMF recibido. W=0000XXXX
   addwf bx, F          ;bx=DTMF
   decfsz cx, F         ;Tomar otro codigo DTMF?
   goto get_next_dtmfstr     ;Si!
   goto end_for_get_dtmfstr  ;No! cx==0.

get_next_dtmfstr:       ;Preparo [ax:bx] para recibir proximo DTMF.
   movlw 0x04
   movwf dx             ;dx=4 numero de rotaciones.
rotate_4left:           ;Desplazo [ax:bx] 4 lugares a la izquierda.
      bcf STATUS, C     ;Limpio CARRY, en proxima rotacion meto C=0.
      rlf bx, F         ;Roto izq. bx y 7 bit se pone en CARRY.
      rlf ax, F         ;Roto izq. ax e introdusco CARRY anterior.
      decfsz dx, F
      goto rotate_4left
   goto for_get_dtmfstr  ;Busco proximo DMTF!

end_for_get_dtmfstr:    ;No hay mas DTMF que obtener.

   call pop_cx
   call pop_dx

   return               ;back to caller!

;play_incorrecto: reproduce "NUMERO INCORRECTO".
```

```
;Destruye: W
play_incorrecto:
   movlw ISD_NUMERO
   call play_isd            ;"NUMERO"
   movlw ISD_INCORRECTO
   call play_isd            ;"INCORRECTO"
   return

;play_rep-sal: reproduce "NUMERAL SALIR, ASTERISCO REPETIR".
;Destruye: W
play_sal_rep:
   movlw ISD_NUMERAL
   call play_isd            ;"NUMERAL"
   movlw ISD_SALIR
   call play_isd            ;"SALIR"
   movlw ISD_ASTERISCO
   call play_isd            ;"ASTERISCO"
   movlw ISD_REPETIR
   call play_isd            ;"REPETIR"
   return

;spause: genera una pausa de W segundos en el programa.
;I: W = Pausa de N segundos.

spause:

   bcf UCT_OPTION, QTIMEOUT ;No resetear al producirse un TIMEOUT
   call set_timeout         ;Producir un TIMEOUT en W segundos.

wait_TIMEOUT:
   btfsc UCT_STATUS, TIMEOUT  ;Ocurrio un TIMEOUT==1?
      goto end_spause         ;Si, volver.
   goto  wait_TIMEOUT         ;No, seguir esperando.

end_spause:
   call unset_timeout         ;Desactivar interrupcion por TMR0.
   bsf UCT_OPTION, QTIMEOUT   ;QTIMEOUT siempre debe estar activada.
   return                     ;back to caller.

;set_timeout: setea el temporizador para que use clock interno
;del pic16f84a y produce un TIMEOUT en W segundos.
```

```
;
;I: W = tiempo en segundos antes de producir el TIMEOUT.
;Modifica: SECONDS, TMR0_AUX, TMR0 y OPTION_REG.
;
;Notas:
;* Se debe usar unset_timeout() para desactivar esta funcion.
;* GIE debe estar habilitado y para desabilitar esta funcion
;se debe inhibir el flag de interrupcion por TMR0. Para ello usar:
;call unset_timeout -> desactiva un TIMEOUT seteado por set_timeout().
;
;Funcionamiento:
;La frecuencia del clock interno sera fi=4Mhz/4 = 1 Mhz .
;Luego el prescaler divide por 32, fi/32= 31250 Hz. E incrementara
;al TMR0 cada 32 pulsos de clock.
;El registro TMR0 se carga con TMR0=6, por lo tanto
;se producira una interrupcion cada 250 incrementos de TMR0.
;Esto es: (fi/32)/250 = 125 Hz.
;Posteriormente se carga la variable TMR0_AUX con 125, y se la
;decrementa cada 125 interrupciones del timer.
;Quedando: ((fi/32)/250)/125 = 1 HZ. Lo que equivale a 1 segundo.
;Finalmente se decrementa la variable SECONDS y si resulta cero
;activa el flag TIMEOUT de la variable UCT_STATUS.
;Luego si flag QTIMEOUT esta activada en UCT_OPTION el UCT se rein-
;niciara.
;El encargado de manegar cada interrupcion es la funcion RSITMR0 y
;decide que hacer con dicha interrupcion.

set_timeout:

   BANK_1
   bcf OPTION_REG, T0CS ;Usar fuente de clock interna, 1 MHz.
   bcf OPTION_REG, PSA  ;Prescaler asignado al TMR0.
   bsf OPTION_REG, PS2  ;Prescaler divide frec. de clock por 32.
   bcf OPTION_REG, PS1  ;Es decir cada 32 pulsos de clock se incremen-
   bcf OPTION_REG, PS0  ;tara en 1 el registro TMR0.
   BANK_0

   movwf SECONDS        ;Numero de segundos para que ocurra el TIMEOUT.
   movlw d'125'         ;Decrementar SECONDS cada 125 interrupciones
   movwf TMR0_AUX       ;del timer 0.
   movlw 0x06           ;Producir una interrupcion cada 250 cuentas
```

```
   movwf TMR0          ;o incrementos del TMR0
   bsf INTCON, T0IE    ;habilitar interrupcion por TMR0
   return              ;back to caller!

;unset_timeout: desactiva un TIMEOUT y asigna prescaler al watchdog.
;* Repone el flag TIMEOUT a cero.
unset_timeout:
   bcf UCT_STATUS, TIMEOUT ;Nos aseguramos que TIMEOUT=0.
   bcf INTCON, T0IE        ;desactivar interrupcion por TMR0.
   BANK_1
   bsf OPTION_REG, PSA
   BANK_0
   return

;*********************************************************************
;      RUTINAS DE MANEJO DE CIRCUITOS DEL UCT       *
;*********************************************************************

;ctrl_disp: activa o desactiva dispositivo externo.
;I: W=S000xxxx
;Donde si S: 1 -> activa dispositivo 0 -> desactiva dispositivo.
;y xxxx representa el numero de dispositivo a activar.
;Destruye: cx, dx, ex,W
;NOTA:
;Si el numero de dispositivo es distinto de [1,2,3,4], solo actualiza
;el latch 4042, con el estado de los disp. almacenado en EEPROM.
;Util para inicializar dispositivos al comienzo del UCT.

ctrl_disp:

   movwf ex       ;Dispositivo a activar.

   BANK_1
   ;Configuramos BUS de datos RB4-7 entre dispositivos.
   ;RB4-7=S -> conectados a latch 4042.
   movlw 0x0F
   andwf TRISB, F ;TRISB=0000XXXX

   BANK_0
   clrf cx        ;cx bit-0 = Tipo de operacion. 1=Activar/0=Desac.
   btfsc ex, 7    ;Activar disp. -> S=1?
```

```
    bsf cx, 0   ;Si activar dispositivo.

  bcf ex, 7     ;Borramos bit-7 de ex.

  movlw EE_DISPSTAT    ;EE_DISPSTAT=0000XXXX
  call eeprom_read
  movwf dx             ;Estado de dispositivos ON/OFF actual.

  ;Determinamos que dispositivo activar o desactivar.
  ;El nuevo estado de los disp. ON/OFF queda en dx.

;if_d4
  movlw 4
  xorwf ex, W
  btfss STATUS, Z           ; ex==4, Dispositivo 4 ?
  goto else_if_d3           ; No.
    btfss cx, 0             ; Activar?
    goto d4_turn_off        ; No.
      bsf dx, 3             ; D4 Activado.
      goto end_if_switch
d4_turn_off:
      bcf dx, 3             ; D4 Desactivado.
      goto end_if_switch

else_if_d3:
  movlw 3
  xorwf ex, W
  btfss STATUS, Z           ; ex==3, Dispositivo 3 ?
  goto else_if_d2           ; No.
    btfss cx, 0             ; Activar?
    goto d3_turn_off        ; No.
      bsf dx, 2             ; D3 Activado.
      goto end_if_switch
d3_turn_off:
      bcf dx, 2             ; D3 Desactivado.
      goto end_if_switch

else_if_d2:
  movlw 2
  xorwf ex, W
  btfss STATUS, Z           ; ex==2, Dispositivo 2 ?
```

```
   goto else_if_d1          ; No.
     btfss cx, 0            ; Activar?
     goto d2_turn_off       ; No.
       bsf dx, 1            ; D2 Activado.
       goto end_if_switch
d2_turn_off:
       bcf dx, 1            ; D2 Desactivado.
       goto end_if_switch

else_if_d1:
   movlw 1
   xorwf ex, W
   btfss STATUS, Z          ; ex==1, Dispositivo 1 ?
   goto end_if_switch       ; No, solo actualizar latch 4042.
     btfss cx, 0            ; Activar?
     goto d1_turn_off       ; No.
       bsf dx, 0            ; D1 Activado.
       goto end_if_switch
d1_turn_off:
       bcf dx, 0            ; D1 Desactivado.
       goto end_if_switch

end_if_switch:

   ;Grabamos estado de los dispositivos en el memoria EEPROM.
   ;Si el nuevo estado de disp. es igual al anterior, no
   ;grabamos en la mem. eeprom, pero si actualizamos latch 4042.
   movlw EE_DISPSTAT
   call eeprom_read
   xorwf dx, W
   btfsc STATUS, Z          ;dx == EE_DISPSTAT ?
   goto latch_update        ;Si. Solo actualizar latch.
     movlw EE_DISPSTAT      ;No. Grabar en eeprom y actualizar latch.
     movwf ex               ;Direccion eeprom y
     movf dx, W             ;Dato a grabar.
     call eeprom_write

latch_update:
   ;Ponemos estado de disp. en el BUS de datos RB4-7
   movlw 0x0F
   andwf PORTB, F           ; PORTB=0000xxxx
```

```
   swapf dx, F             ; dx=xxxx0000
   movf dx, W              ; W=xxxx0000
   iorwf PORTB, F          ; PORTB[4-7]=dx

   ;Seleccionamos Latch 4042 para usar con BUS de datos RB4-7 y
   ;el estado de los dispositivos se transfieren al LATCH.
   bsf PORTA, 2      ;RA2=1 -> activa entradas de 4042.

   nop               ;Esperamos tiempo de propagacion del 4042
   nop               ;del orden de nano-segundos.

   ;Desactivamos Latch 4042 del Bus de datos RB4-7
   bcf PORTA, 2      ;RA2=0 -> desactiva entradas de 4042.

   movlw 0x0F
   andwf PORTB, F    ;PORTB=0000XXXX, Bus de datos a CERO.

end_ctrl_disp:
   return            ;back to caller!

;getdtmf: Espera un codigo DTMF por un tiempo determinado
;almacenado en W. Si el codigo dtmf no se recibe en el
;tiempo especificado en W, produce una intertupcion
;que es manejada por RSITMR0. De lo contrario devuelve
;en W el codigo DTMF recibido.
;Si QTIMEOUT=0, y el tiempo de espera expira la funcion regresa
;W=0xFF, como indicacion que ningun dtmf fue recibido.
;I: W = time-out en segundos.
;O: W = codigo DTMF recibido. (DMTF 0 se codifica como W=0x00)
;Registros stack-eados: ax, ex
;

getdtmf:

   call push_ax      ; salvamos ax.
   call push_ex      ; salvamos ex.
   call set_timeout  ; seteamos un timeout en W segundos.

wait_dtmf:
     btfsc PORTA, 4        ;Hay un codigo DTMF presente en el 8870 ?
       goto dtmf_recived       ;Si, tomar codigo.
```

```
      btfss UCT_STATUS, TIMEOUT  ;Se produjo un TIMEOUT==1?
        goto wait_dtmf          ;No, seguir esperando.
      call unset_timeout         ;Si, sacar timeout y retornar
      movlw 0xFF                ;con W=0xFF.
      goto end_getdtmf           ;Salir de funcion.

dtmf_recived:
   ;Codigo DTMF recibido.
   call unset_timeout   ; No permitir timeout en W segundos.

   BANK_1
   ;Configuramos BUS de datos RB4-7 entre dispositivos como entradas.
   movlw 0xF0
   iorwf TRISB, F      ;RB4-7=Entradas -> conectado a Q1-4 de 8870.

   BANK_0
   ;Seleccionamos al decodificador 8870 para usar con BUS de datos.
   bsf PORTA, 0      ;RA0=1 -> saca de alta Z, Q1-Q4 de 8870.
   nop               ;2 uS para que Q1-Q4 salgan de alta impedancia.
   nop               ;(del orden de nano-segundos)

   ;Leemos codigo DTMF
   movf PORTB, W     ;En 8870 Leo Codigo DTMF de Q1-Q4
   andlw 0xF0        ;W=xxxx0000
   movwf ax          ;ax=W=xxxx0000
   swapf ax,F        ;ax=0000xxxx

   ;Quitamos 8870 del Bus de datos RB4-7
   bcf PORTA, 0      ;RA0=0 -> pone en alta impedancia Q1-Q4 de 8870.

   ;Dejamos Bus de datos RB4-7 conf. como salidas.
   BANK_1
   movlw 0x0F
   andwf TRISB, F    ;RB4-7=Salidas
   BANK_0            ;Dejamos banco cero (por convencion)
   movlw 0x0F
   andwf PORTB, F    ;PORTB=0000XXXX, Bus de datos a CERO.

   ;El 8870 codifica el DTMF 0 como 0x0A, si se recibe DTMF 0
   ;la funcion devolvera W=0. Para preservar naturalidad en numeros.
   movlw 0x0A
```

```
   xorwf ax, W
   btfsc STATUS, Z   ;ax == 0x0A ?
     clrf  ax       ;Si, hacer ax=0.

   movf ax, W        ;W = DTMF recibido.

   ;Esperamos que el usuario deje de enviar el mismo DTMF.
wait_dtmf_end:
   btfsc PORTA, 4      ;El codigo DTMF todavia esta presente?
   goto wait_dtmf_end  ;Si, esperar a que finalize.

end_getdtmf:
   call pop_ax       ;reponemos viejo valor de ax.
   call pop_ex       ;reponemos viejo valor de ex.
   return            ;back to caller!

;play_isd: Reproduce el mensage numero W almacenado en la memoria
;ISD1420, retorna el control al programa cuando termina de
;reproducirse el mensage.
;I: W = Numero de mensage
;Destruye: W.
;Registros stack-eados: ax y ex.

play_isd:

   call push_ax
   call push_ex
   movwf ax          ;ax=W -> Numero de mensage.

   BANK_1
   ;Configuramos BUS de datos RB4-7 entre dispositivos.
   bcf TRISB, 4      ;RB4=S Conectado a A0 de ISD1420
   bcf TRISB, 5      ;RB5=S Conectado a /PLAYE de ISD1420
   bsf TRISB, 6      ;RB6=E Conectado a /RECLED de ISD1420
   bcf TRISB, 7      ;RB7=S Conectado a A4 de ISD1420

   BANK_0
   ;Valor inicial del BUS de datos RB4-7
   ;ISD1420: en modo MESSAGE CUEING + CONSECUTIVE ADDRESS
   movlw b'10110000'       ;RB7=RB5=RB4=1 => (A4=1,A0=1, /PLAYE=1)
   iorwf PORTB, F
```

```
   ;Seleccionamos ISD1420 para usar con BUS de datos.
   bsf PORTA, 1      ;RA1=1 -> Saca de alta Z a 4016.
   nop               ;Esperamos Tiempos de propagacion.
   nop

   ;Elegimos mensage a reproducir en ISD1420.
   decf ax, F        ;AX=W-1 (Ver principio de funcion)
loop_msg_select:
      bcf PORTB, 5   ;RB5=0 -> /PLAYE=0 pone puntero interno de ISD en
                     ;siguiente mensage.
      movlw d'20'
      call mpause    ;Esperar 20 mS
      bsf PORTB, 5   ;RB5=1 -> /PLAYE=1
      movlw d'20'
      call mpause    ;Esperar 20 mS. Esperar Avance de Ptro Interno.
      decfsz ax, F
      goto loop_msg_select

   ;Quitamos modo MESSAGE CUEING, pero dejamos CONSECUTIVE
ADDRESS
   ;en ISD1420 para que no se resetee puntero interno.
   bcf PORTB, 4      ;RB4=0 -> A0=0 en ISD1420
   movlw d'5'        ;Esperar 5 mS
   call mpause

   ;Reproducimos mensage
   bcf PORTB, 5      ;RB5=0 -> /PLAYE=0 reproduce mensage seleccionado.
   movlw d'5'        ;Esperar 5 mS
   call mpause
   bsf PORTB, 5      ;RB5=1 -> /PLAYE=1

   ;Esperamos a que el mensage se termine de reproducir.
wait_play_msg:
      btfsc PORTB, 6       ;(RB6=/EOM) == 0?
      goto wait_play_msg   ;No, /EOM == 1. Seguir esperando.

   bcf PORTB, 7       ;RB7=0 -> A4=0 en ISD1420.
                      ;Reseteamos puntero interno de memoria de la ISD1420.

   ;Quitamos ISD1420 del Bus de datos RB4-7
```

```
   bcf PORTA, 1      ;RA1 -> pone en alta Z a 4016.

   ;Dejamos Bus de datos RB4-7 conf. como salidas.
   BANK_1
   bcf TRISB, 6            ;RB6=S
   movlw 0x0F
   andwf PORTB, F          ;PORTB=0000XXXX, Bus de datos a CERO.
   BANK_0                  ;Dejamos banco cero (por convencion)

   movlw d'20'
   call mpause      ;Esperar 20 mS. Por si se usa play_isd() en loop.

   call pop_ax
   call pop_ex
   return          ;back to caller!

;*****************************************************************
;  RUTINAS DE APOYO INDEPENDIENTES DEL UCT       *
;*****************************************************************

;mpause: establece una pausa de W mili-segundos en el programa.
;La pausa es de: [((1Mhz/8)/125)]^(-1) * W = 1 ms * W .
;I: W = Multiplicador o factor .
;Destruye: W, ex.
;Nota:
; *Desactiva interrupcion Por TMR0 y modifica OPTION_REG.
; *No produce interrupciones.
mpause:

   movwf ex              ;ex=W -> Multiplicador.

   BANK_1
   bcf OPTION_REG, T0CS    ;T0CS=0 -> Usar clock interno.
   bcf OPTION_REG, PSA     ;PSA=0 -> Prescaler asignado a TMR0.
   bcf OPTION_REG, PS2     ;PRESCALER RATE: 1:8
   bsf OPTION_REG, PS1
   bcf OPTION_REG, PS0

   BANK_0
   bcf INTCON, T0IE        ;T0IE=0 -> no interrupcion por TMR0.
   bcf INTCON, T0IF        ;TOIF=0 -> Ponemos Flag a cero.
```

```
wait_mpause:
   movlw 0x83              ;W=neg(0x7D)=neg(d'125')=0x83
   movwf TMR0              ;Cargamos TMR0=W

tmr0_overflow:
   btfss INTCON, T0IF      ;Desbordo el TMR0?
      goto tmr0_overflow   ;No, Seguir esperando.
   bcf INTCON, T0IF        ;Si, Reponemos flag.

   decfsz ex, F            ;ex == 0?
      goto wait_mpause     ;No, seguir esperando.
   return                  ;Si, volver.

;eeprom_read: W=*(W)
;Regresa en W el contenido de una direccion de la memoria EEPROM
;almacenado en W.
;I: W = direcccion de memoria EEPROM
;O: W = valor de la variable en la direccion de memoria EEPROM.

eeprom_read:
   BANK_0
   movwf EEADR       ;Direccion a leer
   BANK_1
   bsf EECON1, RD    ;Leer EEPROM
   BANK_0
   movf EEDATA, W    ;W = EEDATA

   return            ;Back to caller!

;eeprom_write: Escribe el valor de W en la memoria EEPROM, en la
;direccion contenida en ex.
;I: W  = valor a escribir en la mem EEPROM.
;   ex = direccion de la mem EEPROM a escribir.
;Destruye: W
;Nota: bit GIE de INTCON es activado.

eeprom_write:

   BANK_0
   movwf EEDATA
```

```
   movf ex, W
   movwf EEADR

   BANK_1
   bcf INTCON, GIE      ;impedir interrupciones.
   bsf EECON1, WREN     ;activar permiso de escritura de eeprom.
   movlw 0x55           ;start: secuencia de inicializacion.
   movwf EECON2
   movlw 0xAA
   movwf EECON2         ;end: secuencia de inicializacion.

   bsf EECON1, WR       ;empezar a escribir en eeprom.

wait_eeprom_write_time:
      btfss EECON1, EEIF           ;escritura finalizada?
      goto wait_eeprom_write_time   ;No, seguir esperando.

   bcf EECON1, EEIF     ;reponer flag.
   bcf EECON1, WREN     ;desactivar permiso de escritura de eeprom.

   bsf INTCON, GIE      ;activar interrupciones.
   BANK_0
   return               ;back to caller!


;MINI-STACK:
;Funciones para salvar y reponer registros de forma individual.
;Funciones pop_* reponen un registro salvado previamente con
;push_*.
;Donde *=[ax|bx|cx|dx|ex|w|STATUS]
;ATENCION: Usar dos veces concecutivas push_* sobreescribira el valor
;almacenado previamente con el primer push_*.
;Por ello en cada funcion que use el stack debemos, especificar
;que registro pone en el stack para no sobrescribir y perder datos.
;NOTA:
;* Ninguna funcion modifica el registro W, excepto pop_w.
;* Excepto push_w, push_STATUS, pop_w y pop_STATUS, todas
;  las funciones pueden modificar el bit Z del registro STATUS.

;pop_* -> Reponen un registro que fue almacenado con push_*.
```

```
pop_ax:
   movwf TEMPX       ;TEMPX=W
   movf TEMP_ax, W   ;W=TEMP_ax
   movwf ax          ;ax=TEMP_ax=W
   movf TEMPX, W     ;W=TEMPX
   return
pop_bx:
   movwf TEMPX
   movf TEMP_bx, W
   movwf bx
   movf TEMPX, W
   return
pop_cx:
   movwf TEMPX
   movf TEMP_cx, W
   movwf cx
   movf TEMPX, W
   return
pop_dx:
   movwf TEMPX
   movf TEMP_dx, W
   movwf dx
   movf TEMPX, W
   return
pop_ex:
   movwf TEMPX
   movf TEMP_ex, W
   movwf ex
   movf TEMPX, W
   return
pop_w:                ;pop_w no altera al registro STATUS.
   swapf TEMP_W, F       ;TEMP_W = swap(TEMP_W)
   swapf TEMP_W, W       ;W=swap(TEMP_W)
   return
pop_STATUS:           ;pop_STATUS, Restaura STATUS y no afecta el
   movwf TEMPX          ;registro STATUS ni a W.
   swapf TEMP_STATUS, W  ;Ver push_STATUS para entender esta linea.
   movwf STATUS
   swapf TEMPX, F
   swapf TEMPX, W
   return
```

```
;push_* -> Ponen un registro en el stack.

push_ax:
   movwf TEMPX       ;TEMPX=W
   movf ax, W        ;W=ax
   movwf TEMP_ax     ;TEMP_ax=W=ax
   movf TEMPX, W     ;W=TEMPX
   return
push_bx:
   movwf TEMPX
   movf bx, W
   movwf TEMP_bx
   movf TEMPX, W
   return
push_cx:
   movwf TEMPX
   movf cx, W
   movwf TEMP_cx
   movf TEMPX, W
   return
push_dx:
   movwf TEMPX
   movf dx, W
   movwf TEMP_dx
   movf TEMPX, W
   return
push_ex:
   movwf TEMPX
   movf ex, W
   movwf TEMP_ex
   movf TEMPX, W
   return
push_w:              ;push_w y push_STATUS no modifican el registro
   movwf TEMP_W         ;STATUS.
   return
push_STATUS:
   movwf TEMPX          ;TEMPX=W
   swapf STATUS, W      ;W=swap(STATUS)
   movwf TEMP_STATUS    ;TEMP_STATUS=W
   swapf TEMPX, F       ;TEMPX=swap(TEMPX)
```

```
swapf TEMPX, W      ;W=swap(swap(TEMPX))=TEMPX
return

END   ;Fin de codigo/Archivo. (EOF?)
```

[ARCHIVO: UCT.INC]

```
; asmsyntax=pic16f84
;
; File: uct.h
; MCU: PIC-16F84A @ 4 MHz
; Date: 18/09/2003
; Update: 26/01/2004
; Author: Boris Estudiez
;
; DESCRIPCION:
; Definicion de Constantes/Macros/Variables para uct.asm.

;Definicion de Macros
#define BANK_0 bcf STATUS, RP0
#define BANK_1 bsf STATUS, RP0

;*****************************************************************
;* Definicion de "variables estaticas" en EEPROM .          *
;*****************************************************************

EEPROM_ADDR:   ORG 0x2100
EE_RINGS       equ   0x00 ;Cantidad de RINGS para atender llamado.
               de    0x04 ;Valor de fabrica: 4 RINGS.
EE_PASSW_0     equ   0x01 ;Digito 3 y 2 de la clave de acceso (MSB)
               de    0x12 ;Valor de fabrica: 1 y 2
EE_PASSW_1     equ   0x02 ;Digito 1 y 0 de la clave de acceso (LSB)
               de    0x34 ;Valor de fabrica: 3 y 4
EE_DISPSTAT    equ   0x03 ;Ultimo Estado de Dispositivos ON/OFF 0000xxxx
               de    0x00 ;Valor de fabrica: Todos disp. desactivados.

;*****************************************************************
;* Definicion de constantes del programa de control del UCT *
;*****************************************************************

;Definicion de Nombres de Variables en RAM
ax             equ 0x20 ;Variable de proposito general
bx             equ 0x21 ;Variable de proposito general
cx             equ 0x22 ;Variable de proposito general
dx             equ 0x23 ;Variable de proposito general
ex             equ 0x24 ;Variable de proposito general
```

```
UCT_STATUS     equ 0x30 ;Registro de estado del UCT.
                        ;Bit0=RINGING. Intento de llamada.
                        ;Bit1=ANSWERCALL. Respondiendo llamada.
                        ;Bit2=DCEINT. Interrupcion por DCE.
                        ;Bit3=TIMEOUT. Ocurrio un time-out.
                        ;(se debe borrar antes luego de ser usada).
UCT_OPTION     equ 0x31 ;Opciones de funcionamiento del UCT .
                        ;Bit4=SKIPMENU. Evita repro. mensages de menu.
                        ;Bit6=QMSG. Reiniciar reproduciendo mensage/aviso.
                        ;Bit7=QTIMEOUT. Reiniciar al producirse un TIMEOUT.
RINGS_LEFT        equ 0x32 ;Numero de RINGS que faltan para contestar llamado.

TMR0_AUX       equ 0x46 ;Variable auxiliar para ser usada como contador
                        ;del timer.
SECONDS        equ 0x47 ;Almacena numero de segundos que se deben
                        ;temporizar. Usada por set_timeout.
TEMPX          equ 0x48 ;Variable para almacenar valor temporal.
                        ;No usar. P/uso interno de funciones del stack.
TEMP_ax        equ 0x49 ;Ultimas 7 direcciones reservadas para usar
TEMP_bx        equ 0x4A ;con funciones pop_* y push_*
TEMP_cx        equ 0x4B ;Ambas manejan el stack de profundidad: 1
TEMP_dx        equ 0x4C
TEMP_ex        equ 0x4D
TEMP_W         equ 0x4E
TEMP_STATUS    equ 0x4F

;Definicion de constantes numericas del programa.
DTMF_AST       equ 0x0B  ;Codigo del DTMF Asterisco.
DTMF_NUM       equ 0x0C  ;Codigo del DTMF Numeral.

_LOT_RINGS     equ d'18' ;Permite poner 18 RINGS. Ver conf_ring_menu().

ERR            equ 0xFF  ;Operacion no exitosa.
OK             equ 0x00  ;Operacion exitosa.

;Ctes. usadas por UCT_OPTION - UCT_STATUS (Refieren a posicion de bit)
QTIMEOUT       equ 7
QMSG           equ 6
SKIPMENU       equ 4
TIMEOUT        equ 3
```

```
RINGING       equ 0
ANSWERCALL    equ 1
DCEINT        equ 2

;Ctes. de tiempo, pueden usarse con set_timeout().
M04           equ d'240'  ;240 segundos -> 4 Minutos.
S60           equ d'60'   ; 60 segundos -> 1 Minuto.
S30           equ d'30'   ; 30 segundos
S10           equ d'10'   ; 10 segundos
S2            equ d'2'    ;  2 segundos
S1            equ d'1'    ;  1 segundos

;Ctes. usadas por el BUS DCE (DCE = Dispositivo de Control Externo)

;Pines del BUS DCE.
;-->Interrupciones (utilizan logica negada):
#define       DCEINT_PIN    PORTB, 2  ;Interrupme a un DCE.
#define       UCTINT_PIN    PORTB, 1  ;Comprueba Interrupcion por DCE.


;*****************************************************************************
;* Numeros de posicion Asociados a los Mensages Almacenados en la *
;* memoria ISD1420.                                                          *
;*****************************************************************************

;Silencio (primeros dos mensages), Duracion 25 ms cada uno.
ISD_SILENCIO_1          equ d'1'
ISD_SILENCIO_2          equ d'2'

;Generales/Genericos
ISD_NUMERO              equ d'3'
ISD_INCORRECTO          equ d'4'
ISD_INGRESAR            equ d'5'
ISD_DISPOSITIVO         equ d'6'
ISD_LISTO               equ d'7'
ISD_REPETIR             equ d'8'
ISD_DESACTIVADO         equ d'9'
ISD_ACTIVADO            equ d'10'
ISD_CLAVE               equ d'11'
ISD_CAMBIAR             equ d'12'
ISD_SALIR               equ d'13'
```

```
ISD_CONFIGURAR          equ d'14'
ISD_ERROR               equ d'15'

;Menus
ISD_CONTROL_DE          equ d'16'
ISD_VER_ESTADO_DE       equ d'17'
ISD_BUS_DCE             equ d'18'

;Teclas
ISD_UNO                 equ d'19'
ISD_DOS                 equ d'20'
ISD_TRES                equ d'21'
ISD_CUATRO              equ d'22'
ISD_CINCO               equ d'23'
ISD_ASTERISCO           equ d'24'
ISD_NUMERAL             equ d'25'

;Especiales
ISD_EXTERNO             equ d'26'
ISD_CONECTAR            equ d'27'
ISD_RINGS               equ d'28'
ISD_TELEFONICO          equ d'29'
ISD_ALARMA_2            equ d'30'

;Otros
ISD_BIENVENIDO          equ d'31'
```

5.4: Lista de componentes electrónicos del UCT

Esta lista puede bajarse mas actuaslizada y ordenada de la pagina del proyecto.

Partlist

Exported from uct.sch at 25/02/2004 01:37:45a

EAGLE Version 4.11 Copyright (c) 1988-2003 CadSoft

Part	Value	Device	Package	Library	Sheet
B1		RB1A	RB1A	rectifier	1
C1	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C2	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C3	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C4	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C5	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C6	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C7	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C8	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C9	0.1uF	C-EU050-030X075	C050-030X075	rcl	1
C10	470uF/25V	CPOL-USE5-8.5	E5-8,5	rcl	1
C11	470uF/16V	CPOL-USE5-8.5	E5-8,5	rcl	1
C12	220nF	C-EU075-042X103	C075-042X103	rcl	1
C13	470nF/400V	C-EU150-072X183	C150-072X183	rcl	1
C14	220uF/16V	CPOL-USE5-6	E5-6	rcl	1
C15	22pF	C-EU050-030X075	C050-030X075	rcl	2
C16	22pF	C-EU050-030X075	C050-030X075	rcl	2
C17	10nF	C-EU050-030X075	C050-030X075	rcl	2
C18	10nF	C-EU050-030X075	C050-030X075	rcl	2
C19	0.1uF	C-EU050-030X075	C050-030X075	rcl	2
C20	220nF	C-EU025_050-045X075	C025_050-045X075	rcl	2
C21	220nF	C-EU025_050-045X075	C025_050-045X075	rcl	2
C22	220nF	C-EU050-045X075	C050-045X075	rcl	3
C23	220nF	C-EU050-045X075	C050-045X075	rcl	3
D1	1N4007	1N4004	DO41-10	diode	1
D2	12V	ZENER-DIODEZD-7.5	ZDIO-7.5	diode	1
D3	12V	ZENER-DIODEZD-7.5	ZDIO-7.5	diode	1
D4	20V	ZENER-DIODEZD-7.5	ZDIO-7.5	diode	1
D5	20V	ZENER-DIODEZD-7.5	ZDIO-7.5	diode	1
D6	1N4007	1N4004	DO41-10	diode	1
D7	1N4148	1N4148	DO35-10	diode	2
D8	1N4148	1N4148	DO35-10	diode	3
D9	1N4004	1N4004	DO41-10	diode	3
D10	1N4004	1N4004	DO41-10	diode	3

D11	1N4004	1N4004	DO41-10	diode	3
D12	1N4004	1N4004	DO41-10	diode	3
IC1	7812T	7812T	TO220H	linear	1
IC2	7805T	7805T	TO220H	linear	1
IC3	4093N	4093N	DIL14	40xx	1
IC4	PIC16F84AP	PIC16F84AP	DIL18	microchip	2
IC5	4066N	4066N	DIL14	40xx	2
IC6	ISD1420	ISD1400D	DIL28-6	isd	2
IC7	CM8870CP	CM8870CP	DIL18	california-micro-devices	2
IC8	4042N	4042N	DIL16	stdlib	3
IC9	4066N	4066N	DIL14	40xx	3
J1	RJ11 JACK	RJ11	RJ11	stdlib	1
J2	RJ11 JACK	RJ11	RJ11	stdlib	1
JP1		PINHD-1X2	1X02	pinhead	1
JP2		JP1E	JP1	jumper	2
JP3		PINHD-1X2	1X02	pinhead	2
JP4		PINHD-2X7	2X07	pinhead	3
K1	RELE 5V	G6A-234P	G6A-234P	relay	1
K2	RELE	G5L	G5LE	relay	3
K3	RELE	G5L	G5LE	relay	3
K4	RELE	G5L	G5LE	relay	3
K5	RELE	G5L	G5LE	relay	3
OK1	4N25	4N35	DIL06	optocoupler	1
Q1	4 Mhz	XTAL	Q	special	2
Q2	BC546B	BC546B	TO92-EBC	transistor-npn	2
Q3	3.58 Mhz	XTAL	Q	special	2
Q4	BC546B	BC546B	TO92-EBC	transistor-npn	3
Q5	BC546B	BC546B	TO92-EBC	transistor-npn	3
Q6	BC546B	BC546B	TO92-EBC	transistor-npn	3
Q7	BC546B	BC546B	TO92-EBC	transistor-npn	3
R1	1K	R-US_0207/10	0207/10	rcl	1
R2	470	R-US_0207/10	0207/10	rcl	1
R3	VARISTOR-150V	VARISTOR-7,5	R-7,5	varistor	1
R4	10K	R-US_0207/10	0207/10	rcl	1
R5	1K	R-US_0207/10	0207/10	rcl	1
R6	10K	R-US_0207/10	0207/10	rcl	1
R7	100	R-US_0207/10	0207/10	rcl	2
R8	4.7K	R-US_0207/10	0207/10	rcl	2
R9	4.7K	R-US_0207/10	0207/10	rcl	2
R10	10K	R-US_0207/10	0207/10	rcl	2
R11	10K	R-US_0207/10	0207/10	rcl	2
R12	470	R-US_0207/10	0207/10	rcl	2
R13	100K	R-US_0207/10	0207/10	rcl	2
R14	100K	R-US_0207/10	0207/10	rcl	2
R15	100K	R-US_0207/10	0207/10	rcl	2
R16	1K	R-US_0207/10	0207/10	rcl	2
R17	1K	R-US_0207/10	0207/10	rcl	2
R18	100K 1%	R-US_0207/10	0207/10	rcl	2

R19	100K %1	R-US_0207/10	0207/10	rcl	2	
R20	300K / 1%	R-US_0207/10	0207/10	rcl	2	
R21	100K 1%	R-US_0207/10	0207/10	rcl	2	
R22	4.7K 1%	R-US_0207/10	0207/10	rcl	2	
R23	120K 1%	R-US_0207/10	0207/10	rcl	2	
R24	120K 1%	R-US_0207/10	0207/10	rcl	2	
R25	33K 1%	R-US_0207/10	0207/10	rcl	2	
R26	4.7K	R-US_0207/10	0207/10	rcl	3	
R27	100K	R-US_0207/10	0207/10	rcl	3	
R28	4.7K	R-US_0207/10	0207/10	rcl	3	
R29	4.7K	R-US_0207/10	0207/10	rcl	3	
R30	4.7K	R-US_0207/10	0207/10	rcl	3	
R31	4.7K	R-US_0207/10	0207/10	rcl	3	
S1		SLIDING-INVERSOR-SW 2X03		stdlib	1	
S2		NON_INV_SWITCH	1X02	stdlib	1	
T1	2N3904	2N3904	TO92	transistor	1	
TF1	1:1, Z=600 @ 400Hz	TEL_TRAFO_1_1	TEL_TRAFO_1_1	stdlib		1
X1		W237-02P	W237-132	con-wago-508	1	
X2		W237-02P	W237-132	con-wago-508	3	
X3		W237-02P	W237-132	con-wago-508	3	
X4		W237-02P	W237-132	con-wago-508	3	
X5		W237-02P	W237-132	con-wago-508	3	

[FIN DE PROYECTO UCT]