

ANEXOS

CONTENIDO

CONTENIDO	II
ANEXO 1	1
1. GUIA DE USUARIO	1
SISTEMA DE VIGILANCIA Y SEGURIDAD PARA VIVIENDAS “SIVISEV”	2
1.1 INTRODUCCIÓN	3
1.2 PARTES DEL SISTEMA	4
3.1.1 MÓDULO PRINCIPAL.....	4
3.1.2 DISPOSITIVO SERVIDOR.....	5
3.1.3 DISPOSITIVO DE ENVÍO DE TONOS.....	7
3.2 INSTALACIÓN DEL SISTEMA	8
4.1.1 INSTALACIÓN DE LOS SENSORES	9
4.1.2 INSTALACIÓN DE LOS FOCOS Y LA SIRENA	10
4.1.3 INSTALACIÓN DEL DISPOSITIVO SERVIDOR.....	11
4.2 CONFIGURACIÓN EL SISTEMA	12
5.1.1 CREACIÓN DEL TONO A REPRODUCIR.....	12
5.1.2 CREACIÓN DEL GRUPO Y CONFIGURACIÓN DEL TONO PARA LLAMADA Y LLAMADA DE VIDEO.....	13
6.1 FUNCIONAMIENTO DEL SISTEMA	14
6.1.1 MODO 1.....	15
6.1.2 MODO 2.....	17
7.1 USO DEL SISTEMA	18
7.1.1 COMO REALIZAR UNA LLAMADA DE VIDEO	18
7.1.2 COMO REALIZAR UNA VIDEO-LLAMADA Y MONITOREAR.....	18
SIGNIFICADO DE SIGLAS	20
ANEXO 2	21
2. TABLAS ESPECIALES	21
2.1 TABLAS DE MEDICIONES DE LAS PRUEBAS DE HARDWARE	22
2.2 TABLAS DE VALORES ECONÓMICOS	27

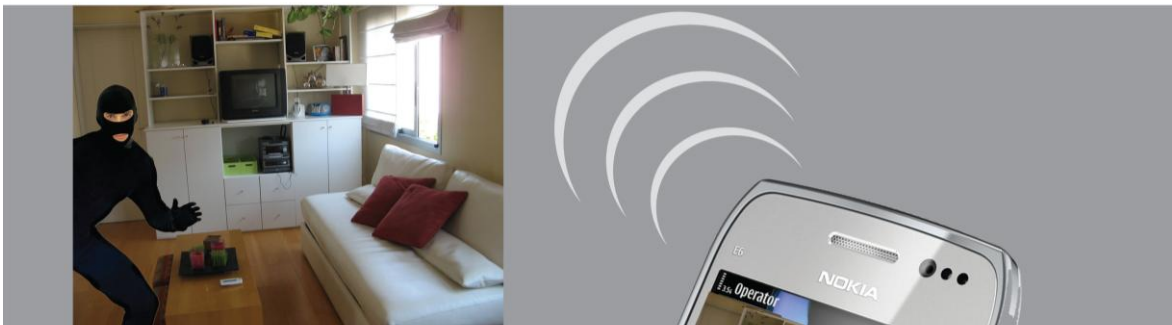
ANEXO 3.....	34
3. CÓDIGO FUENTE	34
ANEXO 4.....	55
4. FOTOGRAFÍAS VARIAS DEL PROYECTO	55
ANEXO 5.....	64
5. PRECIOS DE SISTEMAS.....	64

ANEXO 1

1. GUIA DE USUARIO

SISTEMA DE VIGILANCIA Y SEGURIDAD PARA VIVIENDAS “SIVISEV”

SIVISEV



Sistema
de vigilancia
y seguridad
para viviendas



1.1 INTRODUCCIÓN

SIVISEV es un sistema destinado a brindar seguridad de bienes o personas, y monitoreo de un lugar o ambiente especialmente viviendas donde se tenga artículos o artefactos que necesiten estar en constante protección, con la fácil utilización de su propio teléfono móvil.

Ha sido diseñado pensando en la necesidad de proteger los bienes de personas con la instalación de varios sensores que proporcionan protección del perímetro y del interior del lugar, vigilar las actividades que se realizan en un lugar específico por medio de una video-llamada con movimiento de imágenes, y manejar el encendido/apagado de focos y una sirena de forma remota con tan solo presionar una tecla.

Para obtener el máximo rendimiento del sistema, se cuenta con una batería de respaldo, para el caso de corte de energía, también es importante mencionar la fácil utilización del mismo con la intención de que el usuario se sienta cómodo.

El sistema es de fácil instalación y utiliza una tecnología basada en los microprocesadores para supervisar todas las zonas de protección, revisar el estado del sistema, manejar los dispositivos eléctricos utilizando un teléfono móvil con tecnología 3G.

1.2 PARTES DEL SISTEMA

El sistema se divide en: Módulo principal, Dispositivo servidor y Dispositivo de envío de tonos.

3.1.1 MÓDULO PRINCIPAL

El módulo principal es una caja metálica que contiene a un circuito llamado **unidad de control** (Véase **Figura 1**), es donde se encuentra un micro-controlador que es el cerebro del sistema y el encargado de la administración de todas las acciones que el usuario manifieste, ya sea desde su celular como desde la misma vivienda, además es el encargado de informar mediante una llamada telefónica al usuario, de que existe una emergencia, o sea que si el sistema estaba en vigilancia, se activó un sensor.



Figura 1.- Unidad de control

A este módulo se conectan físicamente los siguientes elementos exteriores: elementos eléctricos (focos y sirena), dispositivo servidor, servomotor (para el movimiento de la cámara durante una video-llamada) y sensores (magnéticos y de movimiento).

En la **Figura 2** se muestra un esquema del módulo principal donde internamente se encuentra la Unidad de control, la Fuente de voltaje de 9 Vcc. y la batería de respaldo (6 Vcc.) con sus diferentes entradas y salidas.

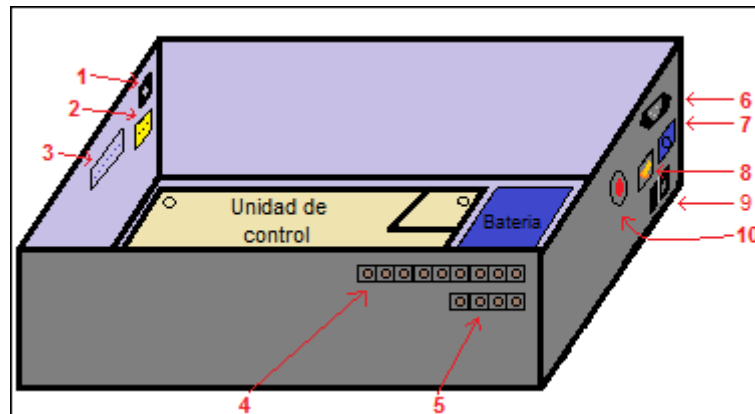


Figura 2.- Esquema del módulo principal

- | | |
|--|---|
| 1) Entrada fuente de 5 Vcc. para el servomotor | 2) Conector del servomotor. |
| 3) Salida al teclado del dispositivo servidor. | 4) Salida de elementos eléctricos. |
| 5) Entrada de sensores. | 6) Entrada de 110/120 Vac. |
| 7) Entrada de audio del dispositivo servidor. | 8) Switch de activación /desactivación del sistema. |
| 9) Switch de encendido/apagado. | 10) Botón de reset |

3.1.2 DISPOSITIVO SERVIDOR

Fue diseñado para recibir una llamada o llamadas de video y realizar las siguientes acciones:

1. Reproducir el tono especial, solo para llamadas entrantes de los números previamente configurados.
2. Contestar automáticamente.
3. Aceptar el envío de imágenes en una llamada de video, y
4. Enviar y recibir tonos DTMF.

Al momento que el usuario realiza una llamada de video, mediante el envío y recepción de tonos desde el conector de sus manos libres (**Véase Figura 3**) puede realizar las siguientes acciones:

1. Recibir la clave de ingreso
2. Controlar y verificar el estado de los elementos eléctricos.

3. Mover la cámara en un rango de 180°.
4. Activar, desactivar y verificar el estado del sistema.



Figura 3.- Conector para la conexión *Dispositivo servidor - Módulo principal*

Al momento que por lo menos uno de los sensores ubicados estratégicamente en la vivienda se acciona, el dispositivo servidor realiza una marcación rápida, a un número especial previamente configurado. Para esto existe un cable que une 3 teclas del dispositivo servidor con el módulo principal (**Véase Figura 4**).

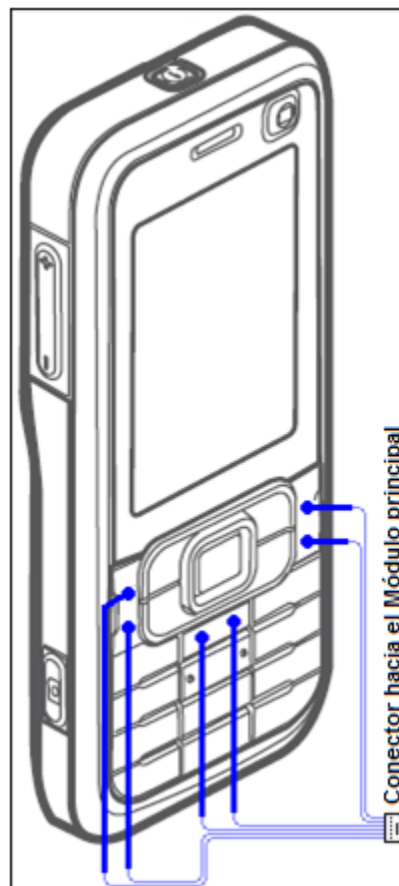


Figura 4.- Esquema del dispositivo servidor

3.1.3 DISPOSITIVO DE ENVÍO DE TONOS

En vista que al realizar una llamada del video, la conexión que brinda la operadora es por un canal diferente al tradicional en el que no se puede enviar tonos DTMF, los cuales son parte fundamental del sistema, se ha diseñado un pequeño circuito que simule el envío de estos por medio de un teclado de 3x4 (Véase *Figura 5*).

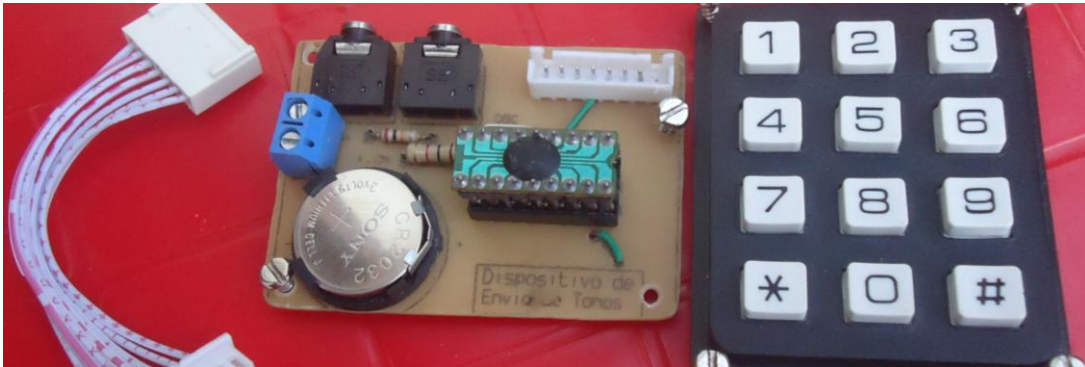


Figura 5.- Dispositivo de envío de tonos.

Como se ve en la *Figura 6*, a este circuito se conecta un par de audífonos para recibir los tonos informativos desde la unidad de control y el teléfono móvil del usuario (haciéndolo detectar como si fuese un dispositivo manos libres).



Figura 6.- Dispositivo de envío de tonos y móvil del usuario.

Nota: El dispositivo de envío de tonos fue diseñado para teléfonos NOKIA de tercera generación, en vista que son la mayoría en los que la red permite llamadas de video, no funciona en teléfonos de 2G.

3.2 INSTALACIÓN DEL SISTEMA

Para una instalación lo más importante es la parte del cableado, el tipo de cable y saber en qué parte del módulo principal conectar dichos cables. En la **Figura 7** se muestra un ejemplo de conexión.

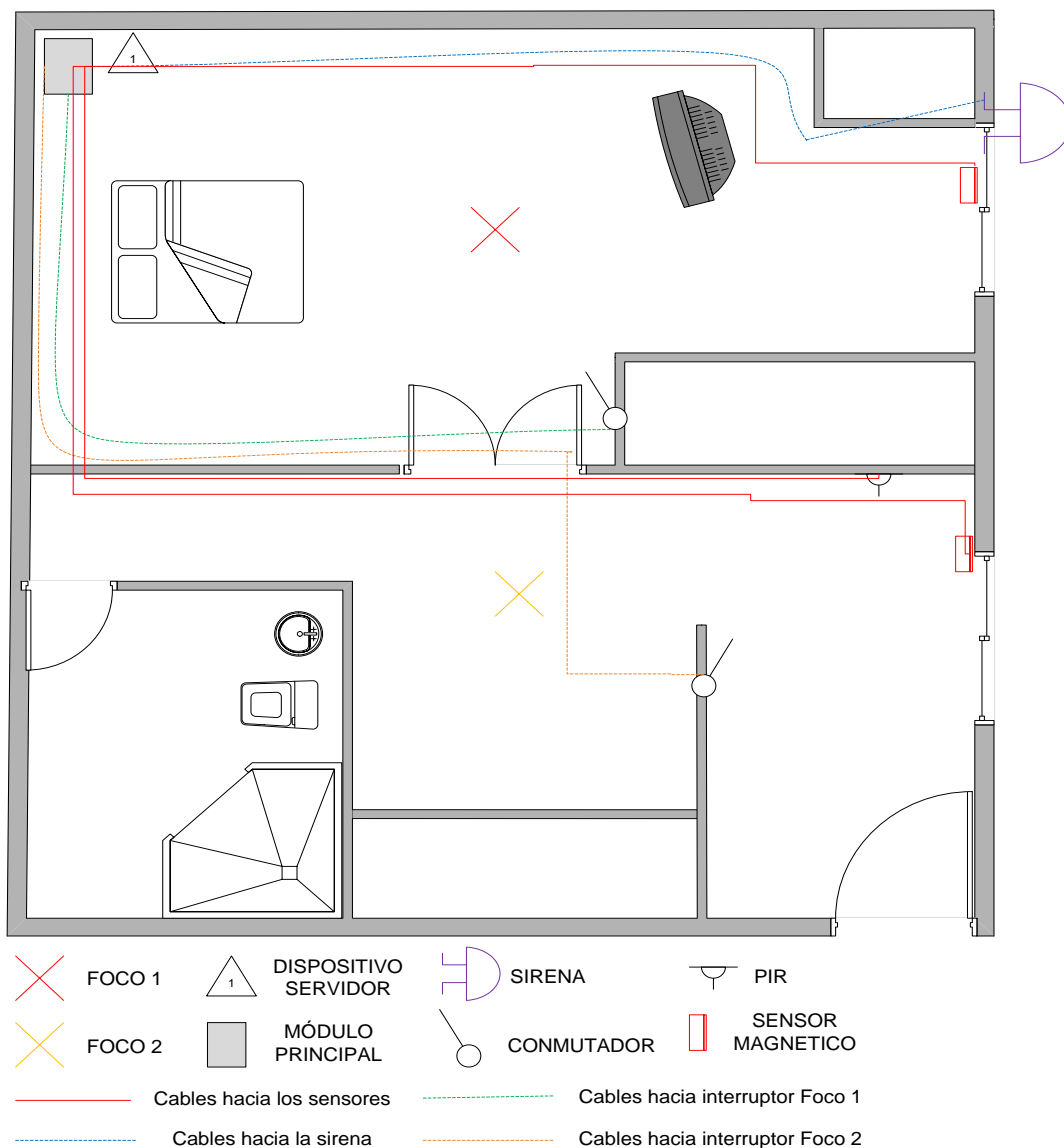


Figura 7.- Cableado y conexión de dispositivos eléctricos y sensores

En el ejemplo de la **Figura 7** se conecta 2 sensores magnéticos en las ventanas y un sensor de movimiento PIR frente a la entrada principal con la intención de cubrir todas las aéreas críticas.

Mientras que para la parte de los elementos eléctricos se conectan 2 focos y una sirena. La salida desde el módulo principal estos elementos se la hace en modo de conmutador eléctrico, por lo tanto se debe reemplazar los interruptores típicos por conmutadores de pared.

A continuación se tiene los elementos a conectar y el tipo de cable a utilizar.

ELEMENTO	TIPO DE CABLE RECOMENDADO
Sensores	Cable plano de teléfono de 2 hilos por sensor
Focos	Cable gemelo 20 AWG
Sirena	Cable gemelo 24 AWG
Teclado del dispositivo servidor	Cable FTP de 8 hilos
Conector del manos libres y Servomotor	Cable UTP de 8 hilos

Tabla 3.1.- Elementos que se conectan con el módulo principal y tipo de cable

4.1.1 INSTALACIÓN DE LOS SENSORES

Los sensores magnéticos y el sensor de movimiento tienen el mismo accionamiento (abrir el circuito de su par de hilos), cada uno con un par de hilos, deben ser conectados de tal forma que 1 hilo de cada sensor se unan en 1 pin común y los 3 hilos restantes a 3 pines distintos del módulo principal, como se muestra en la **Figura 8**.

Los 3 sensores deben ser colocados en sitios estratégicos, para así tener una mayor eficacia.

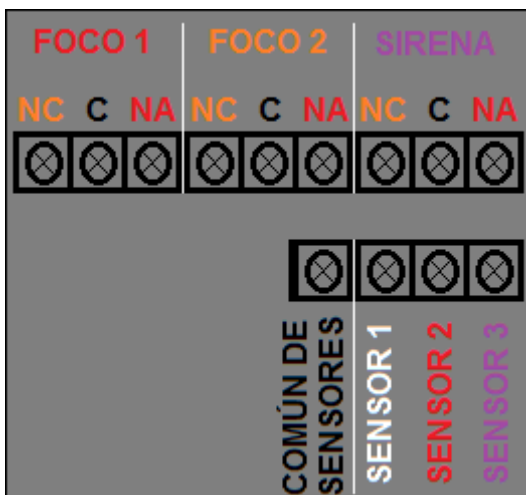


Figura 8.- Conexión de dispositivos eléctricos y sensores en el módulo principal

4.1.2 INSTALACIÓN DE LOS FOCOS Y LA SIRENA

En vista que los 3 pines de salida del módulo principal están configurados como conmutadores, los antiguos interruptores simples de los 2 focos deben ser reemplazados por 2 conmutadores eléctricos cada uno y conectarse como se muestra en la **Figura 8** y **Figura 9**.

Explicación breve: de los tres cables que salen del módulo principal NA (normalmente abierto), C (común) y NC (normalmente cerrado); el NA y NC del módulo irán al NA y NC del conmutador respectivamente, mientras que el común ira empatado con uno de los 2 hilos del anterior interruptor y el otro hilo de este interruptor ira al común del conmutador.

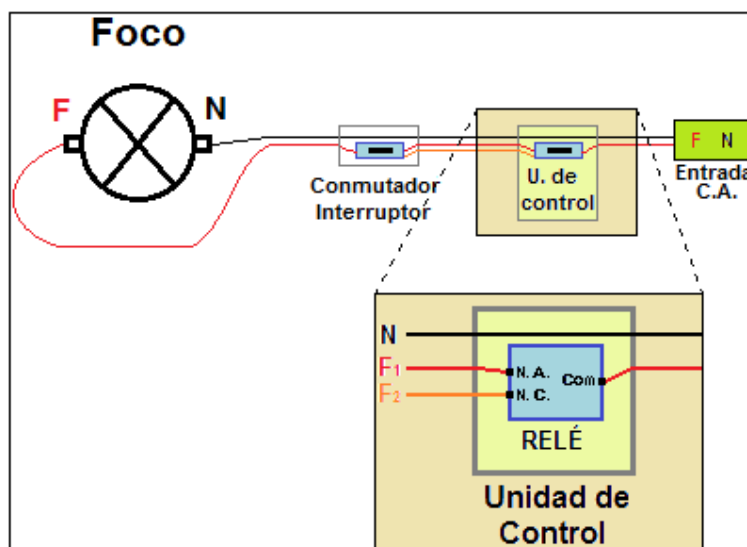


Figura 9.- Conexión de un foco al módulo principal

4.1.3 INSTALACIÓN DEL DISPOSITIVO SERVIDOR

Un elemento que viene incluido en el dispositivo servidor es una pequeña base de madera, diseñada con 2 ejes variables (**Véase Figura 10**) para fácil colocación frente al sitio a monitorear; también tiene a un extremo un servomotor donde se sienta el dispositivo servidor y al otro extremo un metal empotrado a la pared.

Con respecto a los conectores hasta aquí deben llegar:

- Un conector de 6 hilos con el cable FTP (para manejar el teclado), y
- Un cable UTP de 8 hilos, donde utilizan:
 - 3 hilos con un plug que va colocado al conector del manos libres del dispositivo servidor
 - 2 hilos con un conector para el cargador de la batería del dispositivo servidor, y
 - 3 hilos (mando, 5Vcc y tierra) para el manejo del servomotor.

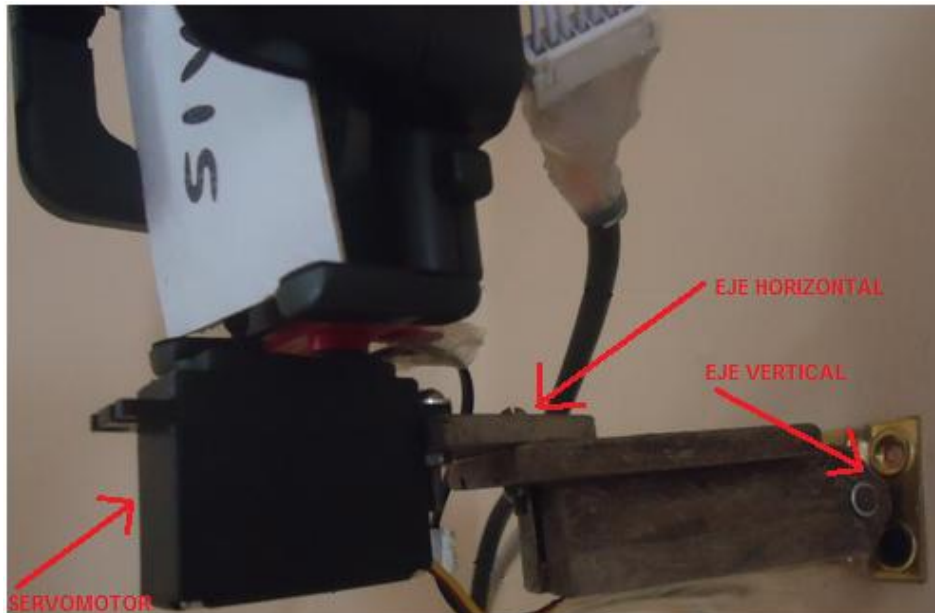


Figura 10.- Base y servomotor del dispositivo servidor

4.2 CONFIGURACIÓN DEL SISTEMA

5.1.1 CREACIÓN DEL TONO A REPRODUCIR

Antes de comenzar se debe crear el tono que reproducirá el dispositivo servidor, el mismo puede ser creado en cualquier editor de sonidos.

Este tono debe tener un Sample rate de 44100, en canal mono, resolución de 16bits y la siguiente secuencia:

- Reproducir el tono DTMF de la tecla 1 durante 750ms
- Reproducir un silencio durante 750ms
- Reproducir el tono DTMF de la tecla 2 durante 750ms
- Reproducir un silencio durante 750ms
- Reproducir el tono DTMF de la tecla 3 durante 750ms
- Reproducir un silencio durante 750ms

Como se muestra en la **Figura 11**.

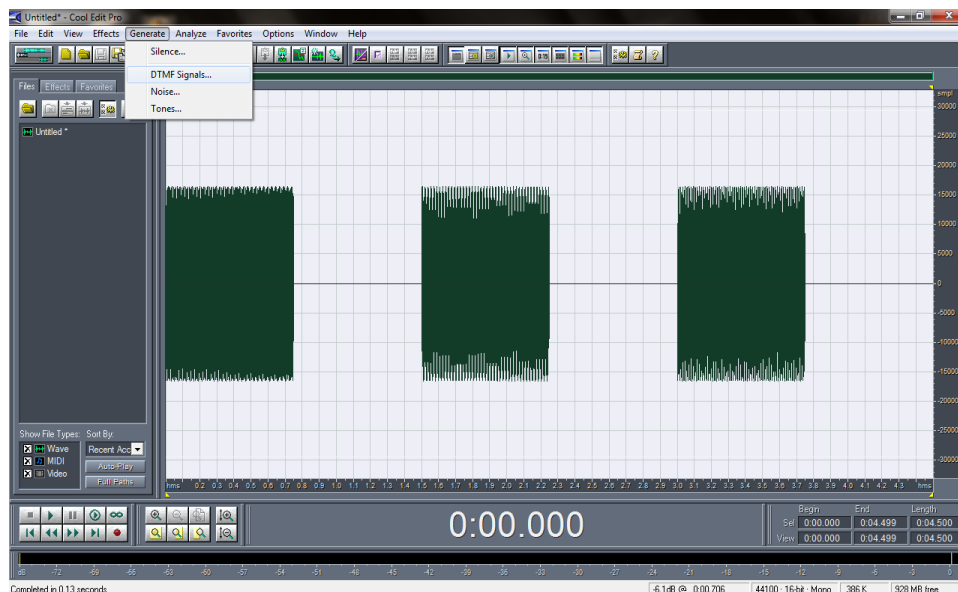


Figura 11.- Creación del tono DTMF en el programa “Cool Edit Pro”

5.1.2 CREACIÓN DEL GRUPO Y CONFIGURACIÓN DEL TONO PARA LLAMADA Y LLAMADA DE VIDEO

Para esta configuración primero se debe crear el contacto telefónico, luego lo guardamos con el respectivo número telefónico así: Seleccionamos **Menú > Comunicación > Contactos > Opciones > Nuevo contacto** e introducimos la información del contacto (nombre y número primordialmente).

Una vez creado el contacto que recibirá las llamadas en caso de emergencia, creamos el grupo así:

Nos colocamos en vista de contactos, nos colocamos sobre el nombre del contacto deseado y Seleccionamos **Opciones > Grupo > Añadir a grupo > Crear nuevo grupo > Nombre de grupo** y escriba el nombre del grupo (**Véase Figura 12**).



Figura 12.- Configuración del grupo de contactos en el dispositivo servidor

Ahora procedemos a configurar el tono para el grupo así:

Nos colocamos en vista de contactos y sobre el nombre del grupo y Seleccionamos **Opciones > Grupo > Tono de timbre** y escogemos el tono que creamos anteriormente (**Véase Figura 13**).

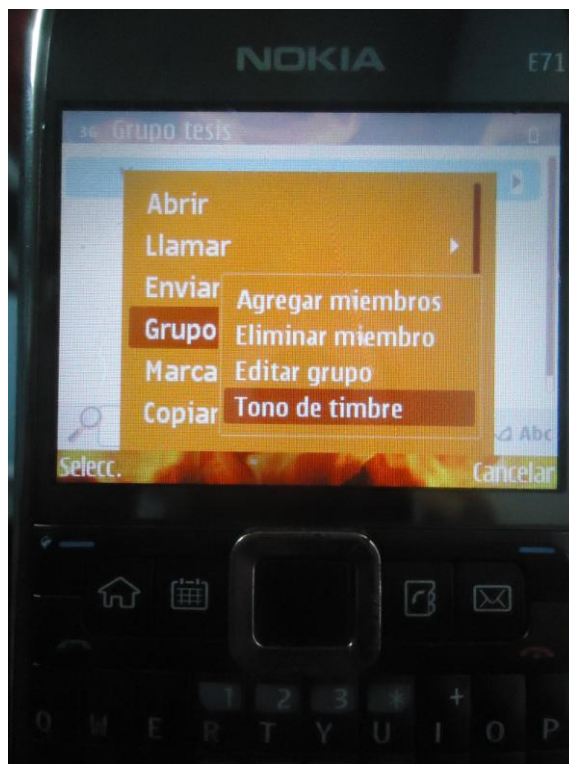


Figura 13.- Configuración del tono de timbre del grupo de contactos en el dispositivo servidor

Nota: En el perfil que se encuentre se debe poner el volumen de timbre al máximo, para que el codificador de tonos de la unidad de control detecte el tono deseado, y el tono creado no se debe ampliar, porque generaría ruido que el detector de tonos no sabrá discriminar.

6.1 FUNCIONAMIENTO DEL SISTEMA

Al momento de iniciarse el sistema existe un tiempo de espera (30 segundos) considerado para la estabilización del sensor de movimiento y para que el usuario pueda salir de la vivienda al activar el sistema (con un switch de forma manual), la activación también se la puede hacer de forma remota mediante cualquier comunicación establecida entre el usuario y la unidad de control.

La finalidad de este sistema es la de monitorear una vivienda basándose en una conexión entre dos dispositivos móviles y usando la tecnología **3G** o superior. En dicha conexión intervienen un dispositivo servidor y el móvil del usuario.

Además del monitoreo, este sistema permite al usuario tomar acciones como encender/apagar los dispositivos eléctricos basado en el envío y recepción de tonos del sistema **DTMF**.

La conexión entre los dos dispositivos se la realiza en dos modalidades: **Modo 1)** Desde el usuario hacia la unidad de control, y **Modo 2)** Desde la unidad de control hacia el usuario.

6.1.1 MODO 1

Tras conectar el **Dispositivo de envío de tonos (Véase Figura 6)**, el usuario se conecta con el otro equipo móvil (y por ende a la unidad de control) desde cualquier parte del mundo con la facilidad del roaming, y tener acceso visual y sonoro (tonos de aviso) hacia el lugar donde se encuentra la vivienda.

Utilizando un servomotor se da movimiento al dispositivo servidor, brindando mayor visualización del lugar (**monitoreo**), de esta manera se constata que las cosas marchan bien, caso contrario tomar acciones como: encender los focos o la sirena en caso de emergencia.

Al momento que el usuario haga una llamada de video se reproducirá el tono antes configurado, que solo a ciertos números reproducirá (**primer aspecto de seguridad**); luego se procederá a la autenticación del usuario (**segundo aspecto de seguridad**) por medio de la toma de la clave de ingreso.

Tras realizar una llamada de video, la unidad de control enviara 3 tonos seguidos, para que el usuario sepa que la unidad de control está esperando la digitación de la clave y luego simulará la digitación de la tecla **“Aceptación de envío de imágenes”** para que el usuario tenga visión amplia de la vivienda.

Por cada dígito de la clave de ingreso bien ingresado, el módulo envía un tono, caso contrario se termina la comunicación.

La clave por defecto es 4-5-6.

Luego del ingreso de la clave el usuario puede hacer uso de cualquier tecla con la siguiente respuesta como se muestra en la Tabla 6.1.

DIGITO DEL TECLADO	ACCION	RESPUESTA DE TONOS
1	Enciende o apaga el foco 1	1 Tono normal
2	Enciende o apaga el foco 2	1 Tono normal
3	Enciende o apaga la sirena	1 Tono normal
4	Mueve la cámara 10° a la izquierda	1 Tono normal
5	Mueve la cámara 10° al centro	1 Tono normal
6	Mueve la cámara 10° a la derecha	1 Tono normal
7	Activa el sistema	1 Tono largo
8	Verifica el estado del sistema	1 Tono largo = activado, 4 Tono cortos = desactivado
9	Desactiva el sistema	4 Tono cortos
0	Edita clave	1 Tono normal por el 1er y 2do dígito ingresado, y 3 tonos cortos por el 3er dígito
*	Verifica el estado de los elementos eléctricos	Se recibirá 3 tonos, donde: El 1er tono representa el estado del foco 1, el 2do el del foco 2 y el 3er tono el de la sirena. 1 tono corto = apagado, y 1 tono largo = encendido
#	Termina comunicación	3 tonos normales

Tabla 6.1.- Teclas digitadas, acciones y respuestas en tonos

6.1.2 MODO 2

En este segundo modo, se tiene instalado varios captadores de eventos (sensores: movimiento y magnético) colocados en puntos estratégicos. En caso de activarse un captador, el circuito de control tendrá 2 funciones:

- a) Realizar una llamada telefónica al usuario, valiéndose del servicio de marcación rápida y de la electrónica, pudiendo simular la digitación de dos teclas del dispositivo servidor.

Primero simulando la marcación de la tecla terminación de llamada, para tener el móvil en modo espera y luego se simula la marcación del dígito con el número “2”, donde previamente estará configurada la marcación rápida con el número celular del usuario, y segundo encender el foco 1 (si estuvo encendido lo apaga por 5 segundos y lo enciende).

En el momento que el usuario reciba la llamada, este la reconocerá y sabrá que se trata de una emergencia, entonces el usuario deberá conectarse con el circuito de control (realizar una llamada de video), pudiendo: **1)** Monitorear, **2)** Encender/Apagar cualquier foco, **3)** Activar la sirena logrando ahuyentar a cualquier intruso, **4)** Llamar a un vecino que acuda por ayuda, o **5)** Llamar a la policía.

Transcurrido 15 segundos de la llamada al usuario la unidad de control simulará la digitación de la tecla “**terminación de llamada**”, con la finalidad de no gastar saldo por si ingresa al buzón de mensajes.

- b) Cuando el circuito de control hace la llamada al usuario y transcurrido un minuto sin respuesta (tiempo iniciado tras la terminación de la llamada explicada en el párrafo anterior), éste procederá a la activación de la sirena que estará encendida hasta que el usuario se percate que recibió una llamada desde el circuito de control y la pueda desactivar al conectarse con el circuito.

Nota: mientras el usuario no conteste a la llamada de emergencia e ingrese correctamente la clave de ingreso, la unidad de control estará encendiendo y apagando el foco 1 y llamando al usuario en un plazo de 2 minutos.

7.1 USO DEL SISTEMA

7.1.1 COMO REALIZAR UNA LLAMADA DE VIDEO

Para realizar una llamada de video, introduzca el número de teléfono o seleccione el destinatario de la llamada en Contactos, y seleccione **Opciones > Llamar > Llamada de video**. Al comenzar la llamada de video, se activará la cámara del dispositivo. Si la cámara ya está en uso, el envío de vídeo está desactivado. Si el destinatario de la llamada no desea enviarle un vídeo, se mostrará en su lugar una imagen fija. Puede definir la imagen fija en **Menú > Herramientas > Configuración > Teléfono > Llamada > Imagen en llamada de video**.

Para desactivar el envío de audio, vídeo o de vídeo y audio, seleccione **Opciones > Desactivar > Envío de audio, Envío de vídeo o Envío audio y vídeo**.

7.1.2 COMO REALIZAR UNA VIDEO-LLAMADA Y MONITOREAR

Antes de establecer una conexión con el dispositivo servidor se debe verificar que su móvil se encuentra en cobertura 3G o superior.

Típicamente puede verificar este estado en un icono pequeño en la parte izquierda superior como se puede ver en la **Figura 14**, dependiendo de la marca y modelo del móvil esta opción puede estar disponible o no.



Figura 14.- Teléfono móvil en cobertura 3G

Luego de revisar esta opción, el usuario haciendo uso de su teléfono móvil y después de conectar el dispositivo de envío de tonos debe realizar una llamada de video al número del dispositivo servidor como se muestra en el **numeral 7.1.1**, y realizar cualquier acción como se muestra en el numeral **6.1**.

FUNCIONAMIENTO DEL SISTEMA.

SIGNIFICADO DE SIGLAS

Vcc.: Voltaje de corriente alterna

Vac.: Voltaje de corriente alterna

DTMF: Dual Tone Multifrequency (*Multi-Frecuencia de Doble Tono*)

PIR: Passive Infra Red (Pasivo infra rojo)

AWG: American Wire Gauge (Calibre de alambre estadounidense)

UTP: Unshielded Twisted Pair (Par Trenzado sin Blindaje)

FTP: Foil Twisted Pair (Par trenzado laminado)

2G: Segunda Generación.

3G: Tercera Generación.

ANEXO 2

2. TABLAS ESPECIALES

2.1 TABLAS DE MEDICIONES DE LAS PRUEBAS DE HARDWARE

ELEMENTO	PIN DEL ELEMENTO	VOLTAJE IDEAL (V)	MEDICIÓN (V)	
			Ambas Fuentes	Solo Batería
PROTOTIPO				
Fuente 9V	Salida	9.00	9.02	8.88
Batería 6V	Salida	6.00	5.58	6.04
LM7805	Salida	4.8 - 5.2	5.09	5.09
Zener de 3.6V	Cátodo	3.42 - 3.78	3.45	3.45
Zener de 6V	Cátodo	5.89 - 6.51	6.40	6.50
DISPOSITIVO DE ENVÍO DE TONOS				
Batería de 3V	Salida	3.00	NA	3.32

Tabla I.- Voltaje en la salida de las fuentes de alimentación
(Fuente: propia)

ELEMENTO	PIN DEL ELEMENTO	VOLTAJE IDEAL (V)	MEDICIÓN (V)	
			Ambas Fuentes	Solo Batería
PROTOTIPO				
LM7805	Entrada	35 máx.	6.42	6.00
PIC16F887	11, 32	2.0 - 5.5	5.04	5.04
	12, 31	0	0.01	0
MT8870	18	2.7 - 3.6 (7 máx.)	3.3	3.3
	9	0	0	0
HM9102	10	2.0 - 5.5	3.3	3.3
	11	0	0.01	0
Servomotor	Del cable Rojo	4.8 - 6	5.25	5.25
	Del cable Negro	0	0	0
DISPOSITIVO DE ENVÍO DE TONOS				
HM9102 (encendido)	10	2.0 - 5.5	NA	3.32
HM9102 (encendido y en uso del teclado)	10	2.0 - 5.5	NA	3.30

Tabla II.- Voltajes en los elementos principales
(Fuente: propia)

ELEMENTO	CORRIENTE IDEAL (A)	MEDICIÓN (mA)	
		Ambas Fuentes	Solo Batería
PROTOTIPO			
Fuente 9V	500m (máx.)	180	N.A.
Batería 6V	1 (máx.)	178	178
LM7805	5.0m - 8.0m	178	177
Zener de 3.6V	10m (máx.)	89	89
Zener de 6V	21m (máx.)	70	70
DISPOSITIVO DE ENVÍO DE TONOS			
Batería de 3V (encendido)	10	NA	0.033
Batería de 3V (encendido y en uso del teclado)	10	NA	0.531

Tabla III.- Corriente de salida de las fuentes de alimentación
(Fuente: propia)

ELEMENTO	PIN DEL ELEMENTO	CORRIENTE IDEAL (A)	MEDICIÓN (mA)	
			Ambas Fuentes	Solo Batería
PROTOTIPO				
LM7805	Entrada	2.20A (máx.)	180	177
PIC16F887	11, 32	25.00m	2.80	2.80
MT8870	18	10.00m	1.40	1.40
HM9102	10	0.60m - 2.00m	1.30	1.30
Servomotor	Rojo	4.80m - 10.00m	7.30	N.A.
DISPOSITIVO DE ENVÍO DE TONOS				
HM9102 (encendido)	10	10	NA	0.033
HM9102 (encendido y en uso del teclado)	10	10	NA	0.531

Tabla IV.- Corriente en los elementos principales
(Fuente: propia)

	DURACIÓN PROMEDIO DE LA BATERÍA (PROTOTIPO)
DESCARGA EN ESTADO DE REPOSO (INACTIVIDAD)	7 horas
DESCARGA EN ESTADO DE EMERGENCIA	6 horas con 40 minutos
DESCARGA EN ESTADO DE COMUNICACIÓN ENTRE DISPOSITIVOS	6 horas con 35 minutos
TIEMPO DE RECARGA	3 horas

Tabla V.- Duración de la batería de respaldo del prototipo
(Fuente: propia)

	DURACIÓN PROMEDIO DE LA BATERÍA (DISPOSITIVO DE ENVÍO DE TONOS)
DESCARGA EN ESTADO DE REPOSO (encendido)	72 horas
DESCARGA EN ESTADO DE COMUNICACIÓN ENTRE DISPOSITIVOS (en uso del teclado)	70 horas

Tabla VI.- Duración de la batería de respaldo del dispositivo de envío de tonos
(Fuente: propia)

2.2 TABLAS DE VALORES ECONÓMICOS

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Base para móvil	1	3.00	3.00
Baquelita	1	3.50	3.50
Broca	1	0.80	0.80
Ácido	2	0.50	1.00
Impresiones de acetatos	2	0.50	1.00
Pulsador (Reset)	1	0.50	0.50
Switch	2	0.50	1.00
Plug estéreo	1	0.30	0.30
Jack para plug estéreo	1	0.40	0.40
Plug Voltaje DC	1	0.30	0.30
Jack Voltaje DC	1	0.50	0.50
Espadines macho/hembra	2	0.85	1.70
Batería	1	19.00	19.00
Fuente 9V DC	1	4.80	4.80
Cable para 110/120 VAC	1	3.00	3.00
Servomotor 3Kg HITEC	1	19.60	19.60
Dispositivo servidor	1	140.00	140.00
Relé 5V	3	0.10	0.30
Diodo 1N4004	2	0.10	0.20
Diodo zener 3,9V y 6.2V	2	0.15	0.30
Transistor 2N3904	3	0.10	0.30
Bornera de 2 contactos	9	0.35	3.15
Bornera de 3 contactos	3	0.42	1.26
Regulador LM7805	1	0.90	0.90

Decodificador MT8870	1	8.00	8.00
Codificador HM9102	1	5.00	5.00
PIC16F887	1	10.00	10.00
Opto-acoplador 4N25	5	0.50	2.50
Oscilador 3.57	2	0.40	0.80
Condensador 0,1uF	2	0.10	0.20
Condensador 100uF/16V	1	0.12	0.12
Zócalo de 8 pines	5	0.20	1.00
Zócalo de 18 pines	2	0.25	0.50
Zócalo de 40 pines	1	0.46	0.46
Diodos LED	8	0.10	0.80
Resistencias	33	0.04	1.32
		TOTAL	237.21

Tabla VII.- Costo del material para la creación del prototipo
(Fuente: propia)

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Caja	1	2.00	2.00
Impresiones de acetatos	2	0.50	1.00
Pulsador (Reset)	1	0.50	0.50
Plug estéreo	1	0.30	0.30
Jack para plug estéreo	2	0.40	0.80
Batería 3V.	1	0.80	0.80
Bornera de 2 contactos	1	0.35	0.35
Codificador HM9102	1	5.00	5.00
Oscilador 3.57	1	0.40	0.40
Teclado de 3x4	1	5.00	5.00
Resistencias	2	0.04	0.08
		TOTAL	16.23

Tabla VIII.- Costo del material para la creación del dispositivo de envío de tonos
(Fuente: propia)

DESCRIPCIÓN	VALOR (USD)
Materiales para el prototipo	237.21
Materiales para el dispositivo de envío de tonos	16.23
Diseño de los 2 circuitos	30.00
Construcción de los 2 circuitos	30.00
TOTAL	313.44

Tabla IX.- Precio de venta al público del prototipo y dispositivo de envío de tonos
(Fuente: propia)

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Sensor PIR Paradox	1	14.50	14.50
Sensor Magnético Paradox	1	2.50	2.50
Sirena 6 tonos	1	6.00	6.00
Fuente 12V DC	3	5.00	15.00
Cable UTP	9 metros	0.50	4.50
Cable Flexible 20 AWG	42 metros	0.48	20.16
Conmutador para cajetín	2	1.50	3.00
TOTAL			65.66

Tabla X.- Costo de elementos varios del sistema “SIVISEV”
(Fuente: propia)

DESCRIPCIÓN	VALOR U (USD)
Diseño y construcción de los circuitos (P.V.P.)	<u>313.44</u>
Dispositivo cliente	<u>140.00</u>
Elementos varios	<u>65.66</u>
Instalación del sistema	<u>50.00</u>
TOTAL	519.10

Tabla XI.- Costos del sistema de seguridad “SIVISEV”
(Fuente: propia)

Dispositivo	Corriente (mA)	Voltaje(V)	Potencia Consumida (W)	Ener. C. (Kwh)
Foco 1	181.00	110	19.91	0.01991
Foco 2	181.00	110	19.91	0.01991
Sirena	10.00	12	0.12	0.00012
Prototipo	47.90	9	0.432	0.000432
Servomotor	70.00	5	0.35	0.00035
Sensor PIR	27.70	12	0.3324	0.0003324
			TOTAL	0.0410544

Tabla XII.- Energía consumida en una hora de los dispositivos eléctricos
(Fuente: propia)

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Paquete mensual de Video-llamada (40min)	1	3.00	3.00
Consumo de energía eléctrica	1	2.21	2.21
		TOTAL	5.21

Tabla XIII.- Costos de operación del sistema “SIVISEV”
(Fuente: propia)

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Dispositivo	1	451.88	451.88
Móvil del cliente	1	140.00	140.00
		TOTAL	591.88

Tabla XIV.- Precio del sistema MÓVILCam¹
(Fuente: propia)

¹ Imagen 17.- Muestra de una proforma del sistema MÓVILCam

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Computador	1	450.00	450.00
Software	1	500.00	500.00
Central GSM	1	200.00	200.00
Sim card	1	5.00	5.00
Instalación	1	50.00	50.00
		TOTAL	1205.00

Tabla XV.- Precio total del sistema GSM
(Fuente: propia)

DESCRIPCIÓN	SUBTOTAL (USD)
Saldo para central GSM	6.00
Saldo mensual disponible	6.00
Consumo de energía eléctrica	16.26
TOTAL	28.26

Tabla XVI.- Costos de operación del sistema GSM
(Fuente: propia)

DESCRIPCIÓN	CANTIDAD	VALOR U (USD)	SUBTOTAL (USD)
Computador	1	450.00	450.00
Cámara infrarroja	1	180.00	180.00
Central	1	200.00	200.00
Sensor de movimiento	1	25.00	25.00
Sensor magnético	1	15.00	15.00
Instalación	1	60.00	60.00
		TOTAL	930.00

Tabla XVII.- Costos de un sistema de monitoreo y seguridad básico
(Fuente: propia)

DESCRIPCIÓN	SUBTOTAL (USD)
Gasto llamadas a celulares mensual	6.00
Consumo de energía eléctrica	16.54
TOTAL	22.54

Tabla XVIII.- Costos de operación de un sistema de seguridad básico
(Fuente: propia)

SISTEMA O DISPOSITIVO	PRECIO (USD)	COSTO DE OPERACIÓN (USD)
SIVISEV	519.10	5.21
MÓVILCam	591.88	3.00
Sistema GSM	241.00	28.26
Sistema de seguridad básico	930.00	22.54

Tabla XIX.- Comparación de precios y costos de operación entre sistemas
(Fuente: propia)

ANEXO 3

3. CÓDIGO FUENTE

```

#include <16F887.h> // PIC a utilizar
#fuses INTRC_IO,NOWDT,MCLR,NOLVP,NOCPD // Configuración de fusibles
#use standard_io(a) // Configuración i/o puertos
#use standard_io(b)
#use standard_io(c)
#use standard_io(d)
#use standard_io(e)
#use delay(clock=4M) // Librería de retardos
#byte porta=0x05 // Asignación de memoria RAM
#byte portb=0x06
#byte portc=0x07
#byte portd=0x08
#byte porte=0x09
int pines=0,Activa=0; // Variables globales
int contRTCC=0,contTMR1=0,llamada=0; // " "
int lecturaA=0,lecturaB=0,lecturaC=0; // " "
int Emer=0,tono=0; // " "
int novaclave1=0; // " "
int novaclave2=0; // " "
char a=0,b=0,c=0,d=0,i=0,f=0; // " "
int x=0,y=0,z=0; // Registros auxiliares para las cargas

//////////Subprogramas para el encendido del LED de aviso//////////
void PinMas(){ // Subprograma para encender LED de aviso
    lecturaC = portc;
    portc = lecturaC + 4;
}
void PinMenos(){ // Subprograma para apagar LED de aviso
    lecturaC = portc;
    portc=lecturaC-4;
}
void UnoCero300(){ // Subprograma para encender y apagar LED de...
    PinMas(); // ... aviso mas unos retardos de 300ms

```

```

delay_ms(300);
PinMenos();
delay_ms(300);
}
void UnoCero100(){           // Subprograma para encender y apagar LED de...
  PinMas();                 // ... aviso mas unos retardos de 100ms
  delay_ms(100);
  PinMenos();
  delay_ms(100);
}
}//////////////////FIN de Subprogramas para el encendido del LED de aviso//////////////////

////////////////// Subprogramas para el Envío de Tonos ////////////////////
void TonoAviso500(){        // Subprograma para enviar tono con retardo de 500ms
  porte = 4;                // COMENTARIO: Si envía un tono el puerto no tiene...
  delay_ms(500);           //... que hacer nada mas, por eso va directo al...
  porte = 0;                //... pin, y no ve estado anterior
  delay_ms(500);
}
void TonoAviso200(){        // Subprograma para enviar tono con retardo de 200ms
  porte = 4;
  delay_ms(200);
  porte = 0;
  delay_ms(200);
}
void TonoAviso100(){        // Subprograma para enviar tono con retardo de 100ms
  porte = 4;
  delay_ms(100);
  porte = 0;
  delay_ms(100);
}
void TonoCorto(){// Subprograma para enviar tono de 200ms y espera 700ms
  TonoAviso200();

```



```

    delay_ms(500);          // Retardo
}
void TonoSistemaDesactivado(){// Subprograma para enviar tonos de sistema...
                                //... desac.
    TonoAviso100();          // Envía 4 tonos de 100ms cada uno
    TonoAviso100();
    TonoAviso100();
    TonoAviso100();
}
//////////////////// FIN de Subprogramas de Envío de Tonos //////////////////////
//////////////////// Subprogramas utilizados en el momento de una Video-llamada //////////////////////
void ServoAlto(){ // Subprograma para activar el pin conectado al servo
    lecturaA=lecturaA+8;
    porta=lecturaA;        // Acción del Subprograma
}
void ServoBajo(){ // Subprograma para desactivar el pin conectado al servo
    lecturaA=lecturaA-8;
    porta=lecturaA;
}
void MoverIzquierda(){ // Mueve el Servomotor hacia la izquierda una vez
    ServoAlto();          // Envía pulso
    delay_us(500);        // 500ms = 90 Grados = mueve a la izquierda
    ServoBajo();          // Termina pulso
    delay_ms(50);         // Retardo para diferenciar pulsos en el servo
}
void MoverDerecha(){ // Mueve el Servomotor hacia la derecha una vez
    ServoAlto();          // Envía pulso
    delay_us(2500);       // 2500ms = -90 Grados = mueve a la derecha
    ServoBajo();          // Termina pulso
    delay_ms(50);         // Retardo para diferenciar pulsos en el servo
}
void MoverCentro(){ // Mueve el Servomotor hacia el centro más un retardo
    ServoAlto();          // Envía pulso

```

```
delay_us(1500);          // 1500ms = 0 Grados = mueve al centro
ServoBajo();            // Termina pulso
delay_ms(50);           // Retardo para diferenciar pulsos en el servo
}

void MoverIzquierda10(){ // Mueve el Servomotor hacia la izquierda 10 veces
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
  MoverIzquierda();
}

void MoverDerecha10(){ // Mueve el Servomotor hacia la derecha 10 veces
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
  MoverDerecha();
}

// Subprograma para enviar imagen durante una videollamada
void EnviarImagenes(){
  portc=32;             // Activa Optoacoplador
  delay_ms(500);        // Retardo de 500ms
  portc=0;              // Desactiva Optoacoplador
```

```

}
void TeclaMarcacionRapida(){
    porte = 2; // Relé para presionar la tecla 2 del móvil y llamar al usuario
    delay_ms(1000);
    porte = 0;          // Luego ya no presiona la tecla
}///// FIN de Subprogramas utilizados en el momento de una Video-llamada /////

////////////////////////////////////Sub Programas Varios////////////////////////////////////
void GrabaReles(){//Subpro. de grabación en la EEPROM, el estado de las cargas
    porta = lectura;          // Carga del estado de las cargas al puerto
    write_eeprom(3,lecturaA); // Grabación en la posición Nro. 4
}
void AceptarRechazarLlamada(){// Subprograma para Contestar/Terminar llamada
    porte=1;          // Activa Opto-acoplador
    delay_ms(500);    // Retardo de 500ms
    porte=0;          // Desactiva Opto-acoplador
}
void TerminaComunicacion(){ // Subpro. para terminar llamada y reiniciar variables
a=b=c=d=tono=lecturaA=0;      // Todo vuelve a estado cero
lecturaB=lecturaC=contRTCC=0; // " " " " "
disable_interrupts(INT_RTCC); // Deshabilita Temporizador Timer0
    UnoCero100();          // Enciende LED de aviso por 200ms
    TonoAviso100();        // Envía tono con 100ms de duración
    AceptarRechazarLlamada(); // Activa Opto para terminar comunicación
}
void TeclaFinalizar(){
    portc = 16;          // Opto para presionar la tecla "Terminar...
    delay_ms(500);      //... llamada (no auriculares)
    portc = 0;          // Deja de presionar la tecla
    delay_ms(500);      // Retardo
}
void PrendeFoco(){//Subprograma para encender Foco1 y así ahuyentar al intruso
//"if" son para ver el estado del Foco1, si x = 0 ==> apagado, x = 1 encendido

```

```

lecturaA = porta & 0b00000111; // Se lee estado de los relés por si ...
//... estuvo encendido solo el Foco 1 o 2
if(x == 0){ // si estaba apagado el Foco1
    lecturaA = lecturaA + 1; //Carga nuevo valor del estado del puerto
    x = 1; // Carga nuevo estado en la EEPROM
}else{ // Apaga y enciende el Foco 1
    lecturaA = lecturaA - 1;
    GrabaReles(); // Apaga el Foco 1 más estado anterior
    x = 0; // Carga nuevo estado en la EEPROM
    delay_ms(5000); // Espera 5 segundos encendido
    lecturaA = porta & 0b00000111; // Nuevamente lee estado de los relés
    lecturaA = lecturaA + 1;
    x = 1; // Carga nuevo estado en la EEPROM
}
GrabaReles(); // Graba en la posición 3 de la EEPROM
}//////////////////// FIN de Subprogramas Varios////////////////////

//////////////////// TIMERS o Temporizadores////////////////////
///TIMER0 Para cortar comunicación a los 15seg. al ingresar o editar clave/////
#INT_RTCC // Llamada a función interrupción del Timer0
void RTCC_isr(){
    contRTCC++; // Incremento del contador
    if(contRTCC < 150){ // Comparación para alcanzar 15 segundos
        set_RTCC(217); // Recarga del timer0
    }
    else{ // Se completó los 15 segundos
        disable_interrupts(INT_RTCC); // Deshabilita Temporizador
        TerminaComunicacion(); //Termina comunicación y reinicia variables
    }
}
}///TIMER1 Para accionar sirena si el usuario no respondió a llamada emergencia
// Debe contar: 15s (Circuito Timbra al Usuario) + 13s (Usuario Timbra) + 10s...
//... (IngresaClave) + 15s (Espera) = 53 segundos, para encender sirena

```

```

#INT_TIMER1 // Llamada a función interrupción del Timer0
void temps1_isr(){
    contTMR1++; // Incremento del contador
    if(contTMR1 == 50){ // Para temporizar 25seg (0.5seg * 50)
        TeclaFinalizar(); // Para colgar por si el usuario no contesta, 8 seg
        // comienza 30 seg. mínimo promedio.
    }
    if(contTMR1 == 120){ // 60 seg (0.5seg * 120) para activar sirena
        if(z == 0){ // Si la sirena está apagada
            lecturaA = porta & 0b00000111; // Se lee estado de los relés
            lecturaA = lecturaA + 4; // Carga valor para encender la sirena
            z = 1; // Cambia el valor de la bandera
            GrabaReles(); // Grabación en la posición Nro. 4...
        } //... memoria EEPROM
    }
    if(contTMR1 <= 240){ // Para temporizar 2 min. (0.5seg * 240)
        set_timer1(3036); // Recarga del timer1 cada 0.5ms
        if(contTMR1 >= 160){ // Pasaron 2 min
            contTMR1 = 0; // Reinicia contador para que vuelva a contar 2 min.
            TeclaFinalizar(); // Primera despierta de Sleep al móvil
            TeclaMarcacionRapida(); // Simulación de presionar tecla "2"
            PrendeFoco(); // Va a prender el Foco1, nuevamente.
        }
    }
}

//////////Subprograma para el control de las cargas y cambio de clave//////////
void Reles(){
    switch(lecturaB){//Se lee la información enviada por el decodificador de tonos
        case 1:if(x==0){ // 1er "Foco"
            lecturaA=lecturaA+1; // Carga el estado para encender el Foco1
            x = 1; // Actualiza bandera del estado de la carga
        }
        else{ // Apaga Foco1

```

```
    lecturaA=lecturaA-1; // Carga el estado para apagar el Foco1
    x = 0;                // Actualiza bandera del estado de la carga
}
GrabaReles();           // Graba estado en EEPROM y acciona cargas
case 2:if(y==0){        //2do "Foco", parecido a la 1ra carga//
    lecturaA=lecturaA+2;
    y = 1;
}
else{
    lecturaA=lecturaA-2;
    y = 0;
}
GrabaReles();
case 3:if(z==0){        // 3ra carga, (Des)acciona la "SIRENA"...
    lecturaA=lecturaA+4; // ... parecido a 1ra y 2da carga
    z = 1;
}
else{
    lecturaA=lecturaA-4;
    z = 0;
}
GrabaReles();
case 4:{                // Mueve el servo hacia la izquierda una vez
    MoverIzquierda();
    UnoCero100();       // Enciende LED durante 200ms
    TonoAviso100();     // Envía tono de acción realizada
}
case 5:{                // Mueve el servo con dirección al centro una vez
    MoverCentro();
    UnoCero100();       // Enciende LED durante 100ms
    TonoAviso100();     // Envía tono de acción realizada
}
case 6:{                // Mueve el servo hacia la derecha una vez
```

```

    MoverDerecha();
    UnoCero100();          // Enciende LED durante 100ms
    TonoAviso100();        // Envía tono de acción realizada
}
case 7:{                  // Activa el sistema
    TonoAviso500();        // Envía tono largo de sistema activado
    pines = Activa = 1;    // Modifica variables
}
case 8:                   // Envía tonos del estado del sistema
    if(Activa==1){
        TonoAviso500();    // Envía tono largo de sistema activado
    }else{
        TonoSistemaDesactivado();//Envía 4 tonos cortos=sistema desac.
    }
case 9:{                  // Desactiva el sistema
    TonoSistemaDesactivado(); //Envía 4 tonos cortos de sistema desac.
    pines=1;              // Modifica variables
    Activa=0;             // "    "
}
case 10: // Subprograma para redirigir hacia el Subprograma "editar clave"
    UnoCero100();          // Encendido del LED de aviso, 4 veces...
    UnoCero100();
    UnoCero100();
    UnoCero100();
    TonoAviso100();        // Envía 4 tonos de 100ms de duración
    TonoAviso100();
    TonoAviso100();
    TonoAviso100();
    c=9;                   // Bandera para q ingrese a cambiaclave()
    contRTCC=0;            // Reinicia el contador del Timer0
    enable_interrupts(INT_RTCC);// Habilita Temporizador para tiempo...
//... de ingreso de clave nueva; si no envía 3 dígitos termina la comunicación
case 11:                  // Envía tonos de estado de los relés

```

```

if(x==0){          // Tonos cortos apagado, Tonos largos activo
    if(y==0){
        if(z==0){          // 000
            TonoCorto();    // 1er relé, 200ms up + 700ms down
            TonoCorto();    // 2do relé, 200ms up + 700ms down
            TonoCorto();    // 3er relé, 200ms up + 700ms down
        }
    }
    if(y==0){
        if(z==1){          // 001
            TonoCorto();    // 1er relé, 200ms up + 700ms down
            TonoCorto();    // 2do relé, 200ms up + 700ms down
            TonoAviso500(); // 3er relé, 500ms up + 500ms down
        }
    }
    if(y==1){
        if(z==0){          // 010
            TonoCorto();    // 1er relé, 200ms up + 700ms down
            TonoAviso500(); // 2do relé, 500ms up + 500ms down
            TonoCorto();    // 3er relé, 200ms up + 700ms down
        }
    }
    if(y==1){
        if(z==1){          // 011
            TonoCorto();    // 1er relé, 200ms up + 700ms down
            TonoAviso500(); // 2do relé, 500ms up + 500ms down
            TonoAviso500(); // 3er relé, 500ms up + 500ms down
        }
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

if(x==1){
    if(y==0){

```



```
        if(z==0){                // 100
            TonoAviso500();      // 3er relé, 500ms up + 500ms down
            TonoCorto();         // 2do relé
            TonoCorto();         // 3er relé
        }
    }
    if(y==0){
        if(z==1){                // 101
            TonoAviso500();      // 1er relé
            TonoCorto();         // 2do relé
            TonoAviso500();      // 3er relé
        }
    }
    if(y==1){
        if(z==0){                // 110
            TonoAviso500();      // 1er relé
            TonoAviso500();      // 2do relé
            TonoCorto();         // 3er relé
        }
    }
    if(y==1){
        if(z==1){                // 111
            TonoAviso500();      // 1er relé
            TonoAviso500();      // 2do relé
            TonoAviso500();      // 3er relé
        }
    }
}
}
}
case 12:// Presionó tecla numeral, Termina la comunicación
    TerminaComunicacion();
}
}
```

```

////////Subprograma de Modificación de clave de tres dígitos en la EEPROM////////
void CambiaClave(){
switch(d){          // En la variable d se carga el Nro. del digito
  case 0:           // 1er digito
    novaclave1=lecturaB;    // Guarda clave en variable novaclave1
    TonoAviso100();        // Envía tono de aviso (100ms) al usuario
    d=1;                  // Incrementa variable Nro. de digito recibido
  case 1:           // 2do digito, igual que el 1er digito recibido
    novaclave2=lecturaB;
    TonoAviso100();// Pero en este envía 2 tonos al usuario
    TonoAviso100();
    d=2;
  case 2:           // 3er digito
    write_eeprom(0,novaclave1);    // Graba en la EEPROM digito 1 recibido
    write_eeprom(1,novaclave2);    // Graba en la EEPROM digito 2 recibido
    write_eeprom(2,lecturaB);      // Graba en la EEPROM digito 3 recibido
    disable_interrupts(INT_RTCC);  // Deshabilita Temporizador
    TonoAviso100();    // Aviso para q el usuario sepa que se edito la...
    TonoAviso100();    // ... clave correctamente, con 4 tonos cortos
    TonoAviso100();
    TonoAviso100();
    d = c = 0;        // Reinicia variables de Cambio de clave
}
}
}***** Subprograma para tomar clave y comparar con datos en la EEPROM *****/
void IngresaClave(){
switch(a){          // En la variable "a" se carga el Nro del digito
  case 0:           // 1er digito
    if(lecturaB==read_eeprom(0)){ // Comparación de dato recibido con digito 1
      UnoCero300();          //Enciende LED de aviso de 300ms de duración
      TonoAviso200();        // Envía tono de aviso al usuario
      a=1;                  // Incrementa variable Nro. de digito recibido
    }else{                // Si erro en el 1er digito
      TerminaComunicacion();// 1er digito mal Ingresado, termina comunicación

```

```

    }
case 1:                // 2do digito
    if(lecturaB==read_eeprom(1)){ // Comparación de dato recibido con digito 2
        UnoCero300();
        TonoAviso200();
        a=2;                //Incrementa variable Nro. de digito recibido
    }else{                // Si erro en el 2do digito
        TerminaComunicacion();// 2do digito mal Ingresado, termina comunicación
    }
case 2:                // 3er digito
    if(lecturaB==read_eeprom(2)){ // Comparación de dato recibido con digito 3
        UnoCero100();        // Envía 3 tonos cortos al usuario
        UnoCero100();
        UnoCero100();
        TonoAviso200();        // Después de descolgar da tres...
        TonoAviso200();        // ... tonos para que ingrese clave
        TonoAviso200();
        disable_interrupts(INT_RTCC); // Deshabilita Timer0
        a=9;                // Ingreso Clave bien, cambia bandera
    }else{
        TerminaComunicacion();// 3er digito mal Ingresado, termina comunicación
    }
}
}

///Subprograma que precede al Manejo de las cargas e Ingresar y editar clave///
void Subrutina(){
    if(a!=9){                // a!=9, esperando ingreso de clave
        IngresaClave();        // Subprograma para ingresar clave
        contRTCC=0;            // Reinicia el contador del Timer0 por cada/Digito
    }
    else{                    // a=9, la clave fue bien ingresada
        if(c==0){            // No espera Cambio de clave

```

```

    Reles();           // Ingresa a Modificar estado de las cargas,...
}                     //... escuchar su estado, o(des)activar sistema
else{
    if(c==9){         // Se presionó "0" en subprograma Reles()
        CambiaClave(); // Ingresa a modificar la clave
    }                 //... que presiona una tecla
}
}
}

#INT_EXT             // Llamada a interrupción externa en el pin RB0
void intrRB0(){     //... en este subprog. se recibe los tonos por la interrup.
lecturaA = porta;   // Lee solo los 4 1ros pines.
lecturaC = portc;   // Lee estado de las cargas y decodificador
lecturaB = portb;   // Para asegurar la lectura de los 4 últimos pines
lecturaB = lecturaB / 16; // Como si leyera los 1ros. 4 pines del puerto B
if(tono!=3){        // Recibió 3er sonido del tono de llamada?
//***** Tratamiento de sonidos de tono de llamada *****//
    if ((lecturaB==1)&(tono==0)){// 1er digito de llamada
        tono=1;           // Para esperar por el 2do sonido
        UnoCero100();     // Enciende LED de aviso, 1 vez
        enable_interrupts(INT_RTCC); // Inicia temporización
    }
    else{
        if((lecturaB==2)&(tono==1)){// 2do digito de llamada
            tono=2;           // Para esperar por el 3er sonido
            UnoCero100();    // Enciende LED de aviso, 2 veces
        }
        else{
            if((lecturaB==3)&(tono==2)){// 3er digito de llamada
                tono=3;           // Listo para ingresar clave
                AceptarRechazarLlamada(); // Contesta llamada, por medio del opto
                TonoAviso200();   // Envía tonos, de espera de Clave

```

```

    TonoAviso200();
    TonoAviso200();
    UnoCero100();           // Enciende LED de aviso, 3veces
    UnoCero100();
    UnoCero100();
    EnviarImagenes();      // ... pasaron 2750ms. Envía de imagen
  }
}
}
else{ // Ya recibió hasta el 3er sonido del tono de llamada
  Subrutina();// Subprog. para recibir/editar clave y para manejo de las cargas
}
}
// Después tiene que recibir por lo menos un tono más: sea por timbre o por...
//... que el usuario presiono una tecla

// Subprog. Utilizados mientras hay una emergencia, o sea se accionó un sensor//
void LedDeAlerta(){
//Subp utilizado mientras se hace la autollamada de emergencia
do{
do{ // Mientras este accionado un sensor, un LED rojo estará intermitente
  Sensor();
  portc = 8;           // Enciende LED de aviso de emergencia
  delay_ms(250);      // Más un retardo
  Sensor();
  portc = 8;           // Enciende LED de aviso de emergencia
  delay_ms(250);      // Más un retardo
  portc = 0;           // Apaga LED de aviso de emergencia
  delay_ms(250);      // Y también se mantiene un peque retardo
  Sensor();
  portc = 0;           // Apaga LED de aviso de emergencia
  delay_ms(250);      // Y también se mantiene un peque retardo

```

```

    Sensor();
    Emer++;           // Incrementa variable para solo encender LED...
                    //... 6 veces (si se desactivo el sensor enseguida)
}while(llamada == 1); //Sensores activos?
}while(Emer<6);     // Se encendió LED 6 veces?
Emer=0;            // Reinicia variable
portc=0;          // Reinicia salida del puerto c
}

void Sensores(){   // Subprograma para censar
    if((!input(pin_D0))(!input(pin_D1))(input(pin_D2))){
        TeclaFinalizar();           // Por si estuvo presionado...
                                    //...algún número en la pantalla del móvil
        TeclaMarcacionRapida();     // Simulación de presionar tecla "2"
        llamada = 1;                // Si llamada es 1, usuario no ha devuelto...
                                    //... la llamada después de una emergencia
        PrendeFoco();               // Va a prender el Foco1
        enable_interrupts(INT_TIMER1); // Inicia tiempo para encender la sirena
        LedDeAlerta();              // Va a encender LED intermitente
        portc=0;                    // Todo regresa a cero
    }
}

void Sensor(){ // Subprograma para censar con 100ms de retardo, dos veces
    Sensores();           // Verifica el estado de los sensores
    delay_ms(100);        // Retardo
    Sensores();
    delay_ms(100);
}

////////// FIN de Subprogramas utilizados mientras hay una emergencia //////////

void SistemaActivado1(){// Activación del sistema por medio del switch
if(f==0){
    f = 1;                // Para solo ingresar 1 vez

```

```

for(i = 0 ; i < 200 ; i++){          // Tiempo de espera por alarma activada...
                                     //... 150ms * 200 = 30 segundos
    if(pines==1){                    //Interrumpe, se está activando por software
        if((Activa==0)|(!input(pin_C0))){// Desactivando remotamente
            f=0;                      // Encera variable para LED intermitente
        }
    }
    portc=4; // Enciende LED intermitente de sistema recientemente activado
    delay_ms(80);
    portc=0;
    delay_ms(70);
}
}
portc = 2;                            // Enciende LED intermitente de sistema activado
Sensor();
portc = 0;
Sensor();
}

void SistemaActivado2(){// Activación del sistema por software (remotamente)
if(f==0){
    f = 1;                            // Para solo ingresar 1 vez
    for(i = 0 ; i < 200 ; i++){        // Tiempo de espera por alarma activada...
                                         //... 150ms * 200 = 30 segundos

        if(!input(pin_C0)){           // Activo moviendo el Switch
            f=0;                        // Encera variable
        }
        portc=4; // Enciende LED intermitente de sistema recientemente activado
        delay_ms(80);
        portc=0;
        delay_ms(70);
    }
}
}

```

```

portc = 2;                //Enciende LED intermitente de sistema activado
Sensor();
portc = 0;
Sensor();                // Apaga LED
}

```

```

//***** Programa Principal*****//

```

```

void main(void){
    porta=0x00;           // Enceramiento de puertos
    portb=0x00;
    portc=0x00;
    portd=0x00;
    porte=0x00;
    set_tris_a(0x00);     // Configuración de E/S de puertos
    set_tris_b(0xFF);
    set_tris_c(0x01);
    set_tris_d(0x07);
    set_tris_e(0x00);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256);// Configuración de Timer0
    set_RTCC(217);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);// Configuración de Timer1 (0.5ms)
    set_timer1(3036);
    ext_int_edge(H_to_L); // Configuración de interrup externa
    enable_interrupts(GLOBAL); // Habilitación interrupción global
    disable_interrupts(INT_RTCC); // Des-habilitación de interrupción Timer0
    disable_interrupts(INT_TIMER1); // Des-habilitación de interrupción Timer1
    disable_interrupts(int_ext); // Des-habilitación de interrupción externa
    lecturaA = read_eeprom(3); // Lectura del estado anterior de las cargas
    porta = lecturaA; // Carga estado anterior de las cargas
    switch (lecturaA) { // Cargar las BANDERAS del estado anterior de los relés
        case 0:  z=0;y=0;x=0; // 000
            break;
        case 1:  z=0;y=0;x=1; // 001

```



```

        break;
    case 2:  z=0;y=1;x=0;    // 010
        break;
    case 3:  z=0;y=1;x=1;    // 011
        break;
    case 4:  z=1;y=0;x=0;    // 100
        break;
    case 5:  z=1;y=0;x=1;    // 101
        break;
    case 6:  z=1;y=1;x=0;    // 110
        break;
    case 7:  z=1;y=1;x=1;    // 111
        break;
}
MoverCentro ();           // Para mover servomotor a "0 grados"

enable_interrupts(int_ext); //Habilitación de interrupción externa
do{                        // Inicia lecturas
    if(pines==0){         // Revisa Switch
        if(!input(pin_C0)){ // Switch en posición de sistema activado?
            Activa = 1;     // Bandera de sistema activado = 1
            write_eeprom(4,1); //Graba estado del sistema en la posición Nro. 5
            SistemaActivado1(); // Subrutina para Switch
        }
    }
    else{                 // Switch en posición de sistema desactivado
        Activa=f=0;       // Enceramiento de variables
        write_eeprom(4,0);
    }
}
else{                    // No revisa switch, está en (des)activación remota
    if(Activa==1){       // Sistema activado?
        SistemaActivado2(); // Subrutina para sistema activado
        if(!input(pin_C0)){ // Se activó por switch?

```

```
        pines=0;          // Movi6 el switch una vez, pasa control al Switch
    }
}
else{                    // Sistema desactivado
    f=0;
    if(input(pin_C0)){   // Si desactiva por switch?
        pines=0;        // Movi6 el switch una vez, pasa control al Switch
    }
}
}
}
}while(true);
}
```

ANEXO 4

4. FOTOGRAFÍAS VARIAS DEL PROYECTO

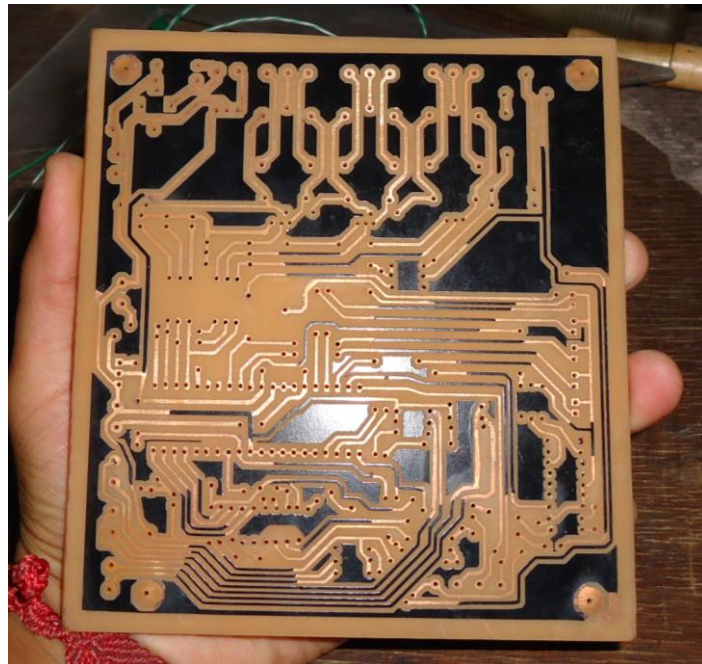


Imagen 1.- Pistas de la placa del prototipo
(Fuente: propia)

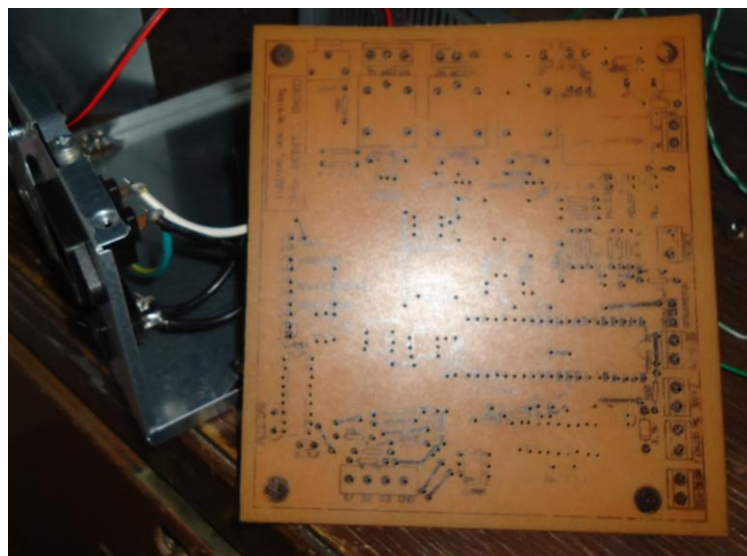


Imagen 2.- Caja metálica y placa base del prototipo
(Fuente: propia)

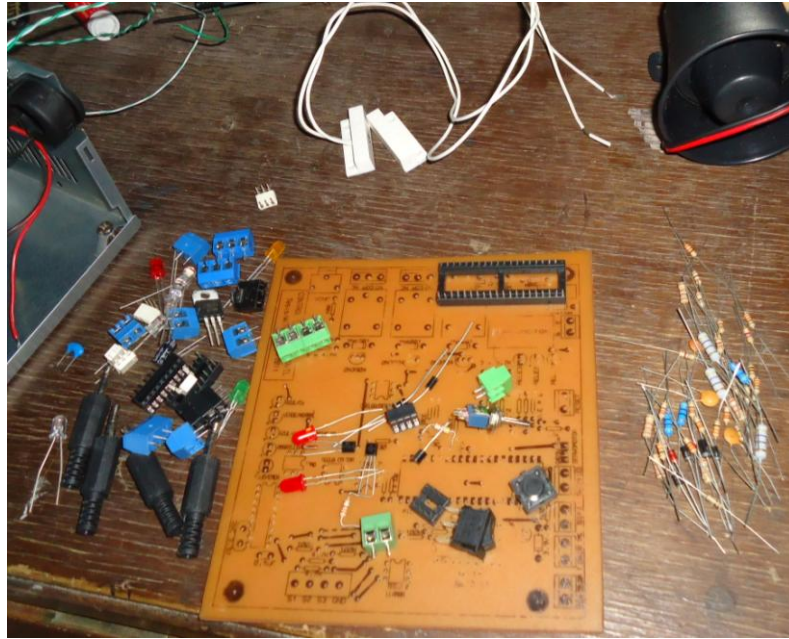


Imagen 3.- Materiales listos para el ensamblaje en la placa base, sensor magnético y sirena
(Fuente: propia)



Imagen 4.- Elementos en la placa del Prototipo
(Fuente: propia)

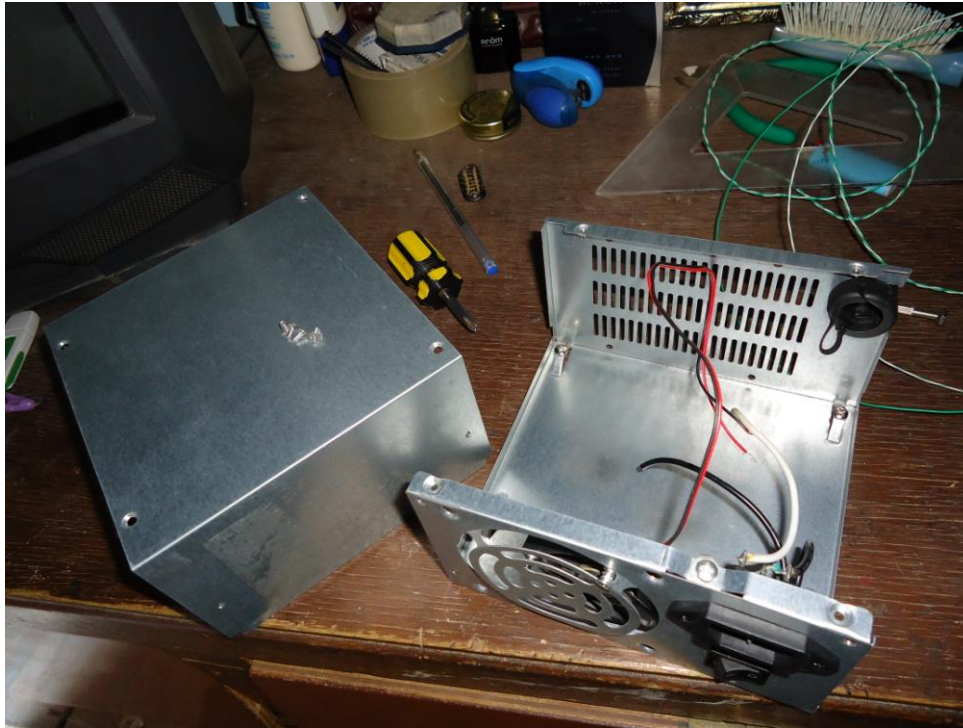


Imagen 5.- Caja metálica para el prototipo
(Fuente: propia)

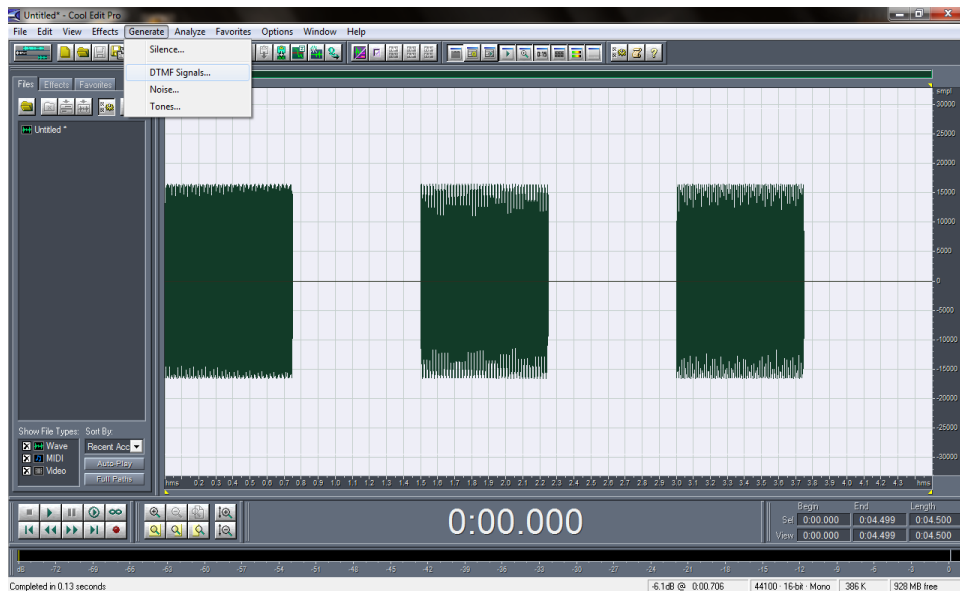


Imagen 6.- Creación del tono que reproducirá el dispositivo servidor
(Fuente: propia)



Imagen 7.- Configuración del grupo de contactos en el dispositivo servidor
(Fuente: propia)



Imagen 8.- Configuración del tono de timbre del grupo de contactos en el dispositivo servidor
(Fuente: propia)



Imagen 9.- Parlante del Dispositivo servidor
(Fuente: propia)



Imagen 10.- Soldas en el teclado del dispositivo servidor
(Fuente: propia)



Imagen 11.- Vista frontal del dispositivo servidor
(Fuente: propia)

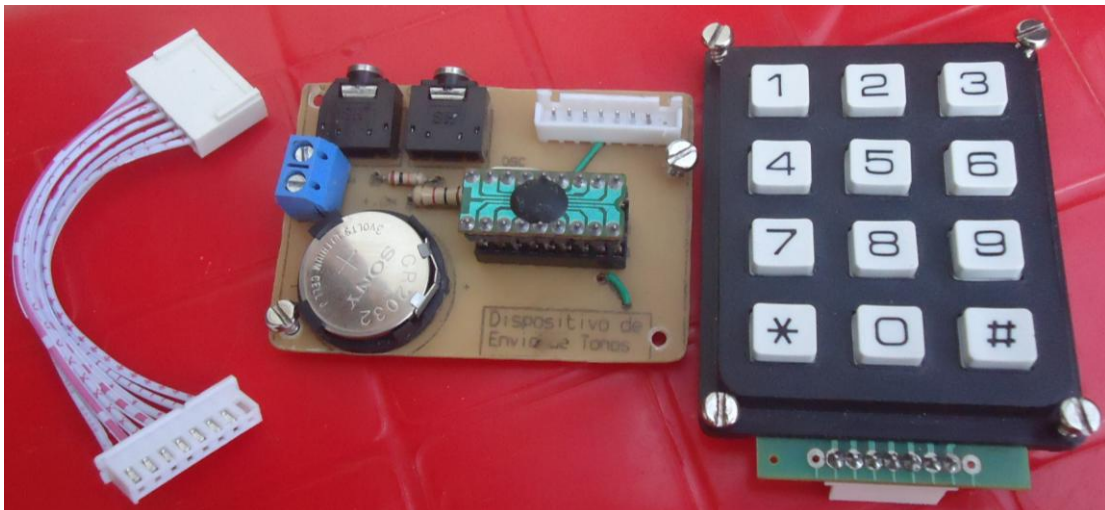


Imagen 12.- Partes del Dispositivo de envío de tonos
(Fuente: propia)



Imagen 13.- Instalando el dispositivo servidor en la pared de la vivienda
(Fuente: propia)



Imagen 14.- Colocando los cables que conectan al dispositivo servidor con el prototipo
(Fuente: propia)



Imagen 15.- Instalación de un sensor de movimiento
(Fuente: propia)



Imagen 16.- Instalación de un sensor magnético y la sirena
(Fuente: propia)

ANEXO 5

5. PRECIOS DE SISTEMAS

Siéntelos siempre cerca con MóvilCam de PORTA

Más que una cámara, la mejor forma
de estar cerca de todo.



Ahora podrás saber al instante lo que ocurre en tu casa,
oficina o lugar de trabajo, sólo tienes que llamar a tu MóvilCam
desde cualquier lugar y podrás saber todo lo que pasa.

Realiza Video Llamadas a tu
MóvilCam a sólo **\$ 0.04 + imp****
el minuto registrándola o activándola
como Más Familia + Amigos

\$199⁰⁰

12 cuotas mensuales
Total pago inicial

\$17.99
\$235.88*

Incluye costo del servicio y cuota del equipo*.



Además, desde tu celular puedes dirigir la lente
de la cámara donde más te interesa ver.

T 09/2016 P08118.10

Imagen 17.- Muestra de una proforma del sistema **MÓVILCam**
(Fuente: propia)