UNIVERSIDAD RICARDO PALMA

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

"DISEÑO DE UN SISTEMA CRIPTOGRÁFICO DE VOZ PARA TELÉFONO FIJO"

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE

INGENIERO ELECTRÓNICO

PRESENTADO POR

LUIS ANGEL VELASCO MELLADO

LIMA - PERÚ

2014

RESUMEN

Este proyecto de tesis se encuentra enfocado en el diseño de un sistema criptográfico para la transmisión de voz por la línea telefónica; es decir, transmitir la señal de voz de manera cifrada debido a la inseguridad que se viene teniendo hoy en día en nuestro país, utilizando un algoritmo de encriptación y un hardware diseñado para la comunicación telefónica.

Este sistema cuenta con el fácil manejo del cambio entre modo seguro e inseguro, y el ingreso de una clave que se requiere para efectuar debidamente el sistema, así como una fácil conexión del equipo telefónico a este equipo y el mismo a la línea telefónica mediante conectores típicos de telefonía.

La señal de voz se hace pasar por un conversor de 2 a 4 hilos para separar la señal de transmisión de la recepción y así procesar cada una de manera independiente. Además se hace pasar por un filtro y luego se amplifica la señal debido a la baja amplitud que tiene. Luego se aplica una codificación para convertir la señal análoga en digital y permitir que un microcontrolador pueda procesar la señal y aplicar el algoritmo de manera digital, todo ello a nivel de bytes para luego ser transmitida por la señal telefónica por medio de un modem que permite la transmisión de datos digitales por la red telefónica.

ABSTRACT

This tesis project is focus in the design of a crypting system for voice transmition trough the telephone line; that means to transmit voice signal in a safety way due to the unsecurity

that there is nowadays in our country, using a crypting algorithm and a hardware designed for telephone comunication.

This system count on an easy control for changing between safety mode and unsafety mode, and the entry of a key that it's used for a good execution of the system and an easy conection betwen this system with the telephone and the thelephone line.

The voice signal pass through a converter from 2 to 4 lines for divide the signal into transmition and reception signals for processing each one in an independet way. This signal pass through a filter and then is amplified due to the low amplitud that the voice has. Later a codification is applied to the signal for converting the analog signal into digital and permit that a microcontroller can process the voice and apply the algorithm in a digital way, all that in bytes levels for finally transmit the signal into the telephone line through a modem that permits tha transmition of digital data trough the telephone network.

ÍNDICE:

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Los sistemas de telefonía fija han pasado a ser enlaces no seguros ya que hoy en día se va efectuando mucho lo que se denomina “chuponeo” (intercepción de la comunicación telefónica), el cual trata de interceptar la comunicación que tienen dos personas y así poder utilizar esta información obtenida para cualquier fin que desee el interceptor. Este acto va en contra de la ley, cuando se encuentra fuera de los regímenes militares para temas de terrorismo y narcotráfico, y por más que realicemos una denuncia hacia quien comete este delito, en el caso que realmente sepamos quien lo hizo, nuestra privacidad y la de los demás se encuentra en riesgo y en ciertos casos esta información obtenida puede ser perjudicial para nosotros si se llega a publicar.

En nuestro país se han tenido muchos casos sobre intercepciones telefónicas y más en el ámbito político y es por estos hechos actuales que ya no podemos estar seguros de nuestra privacidad, por ello se debe de dar una solución a este problema que pueda estar al alcance de varias personas.

Por eso se plantea hacer un sistema que modifique (cifre) nuestra voz para que únicamente la pueda reconocer la persona con la cual nos estamos comunicando directamente. Se espera que este encriptado de la voz sea inentendible para cualquiera que no cuente con la tecnología necesaria y que tenga un dispositivo parecido.

CAPÍTULO 1: MARCO TEÓRICO

1.1 Planteamiento del Problema

Los sistemas de telefonía fija han pasado a ser enlaces no seguros ya que hoy en día se va efectuando mucho lo que se denomina "chuponeo" (intercepción de la comunicación telefónica fija), el cual trata de interceptar la comunicación que tienen dos personas y así poder utilizar esta información obtenida para fines perjudiciales.

Este acto va en contra de la ley, cuando se encuentra fuera de los regímenes militares para temas de terrorismo y narcotráfico, y por más que realicemos una denuncia hacia quien comete este delito, en el caso que realmente sepamos quien lo hizo, nuestra privacidad y la de los demás se encuentra en riesgo y en ciertos casos esta información obtenida puede ser perjudicial para nosotros si se llega a publicar.

En nuestro país se ha tenido últimamente mucha presencia, tal como fueron los casos de:

- Los petroaudios involucrando a Rómulo León y Alberto Quimper. (Octubre de 2008)
- El comentario de molestia de la alcaldía de Lourdes Flores Nano a puertas de las elecciones municipales. (Setiembre de 2010)
- El caso de Roberto Martínez de adquirir equipos de chuponeo para el alcalde del Callao. (Marzo de 2012)

Y es por estos hechos actuales que vivimos, ya no podemos estar seguros de nuestra privacidad, por ello se debe de dar una solución a este problema que pueda estar al alcance de varias personas, donde ya depende de cada persona si usarlo de una manera correcta o incorrecta.

Por eso se plantea hacer un sistema que modifique (encripte) nuestra voz para que únicamente la pueda reconocer la persona con la cual nos estamos comunicando directamente. Se espera que el encriptado de la voz sea inentendible para cualquiera que no cuente con la tecnología necesaria y que tenga un dispositivo parecido.

1.2 Antecedentes

La empresa Tactical Security proporciona equipos de encriptación para teléfono fijo como el "HS-9330 Encriptacall" que cuenta con una robusta caja de forma que sea 100% portable. Se ajusta a varios tipos de dispositivos de comunicación y transmisión de voz con excelente compatibilidad; teléfonos convencionales, contestador telefónico, teléfonos móviles, auriculares Bluetooth, equipo manos libres, etc. y que tiene un costo de alrededor de 1945 dólares. [1]

La empresa estadounidense Technical Communications Corporation ha desarrollado un equipo cifrador llamado "CSD 3324 SP" que provee un sistema totalmente seguro para comunicaciones telefónicas tanto para corporaciones gubernamentales y otras aplicaciones. Utiliza un algoritmo de encriptación aes 256-bit genera llaves aleatorias y las comparte mediante la línea de comunicación mediante otro algoritmo de encriptación llamado Diffie-Hellman que no requiere una clave adicional. [2]

La empresa alemana CryptoPhone dedicada al rubro de seguridad informática avanzada y especializada en alta tecnología de seguridad, ofrece un producto llamado "GSMK Cryptophone PSTN 1i" que es un teléfono fijo seguro con cifrado de voz de extremo a extremo que opera en cualquier red pública de telefonía. [3]

A continuación se muestra una tabla indicando el precio (puesto en Lima) de los equipos antes mencionados. Ver tabla 1.

Nombre	Empresa	Costo(S/.)
HS-9330 Encriptacall	Tactical Security	10580.8
CSD 3324 SP	Technical Communications Corporation	12345.2
GSMK Cryptophone PSTN1i	CryptoPhone	15552

Tabla 1: Listado de Equipos Comerciales en el Mercado

Los equipos anteriormente mencionados tienen el problema de un costo alto, que tomando en cuenta lo adicional que pagaríamos por el traslado hacia nuestro país y pagos en aduanas de flete y otros que se deben realizar, el costo del equipo crece y no se encuentra al alcance de cualquier persona que desee adquirirlo y limita la adquisición de estos equipos a ciertas personas o entidades.

1.3 Objetivos

1.3.1 General

- Diseñar un sistema criptográfico que mantenga de manera muy segura las comunicaciones, que se tienen mediante los teléfonos fijos, a base de un algoritmo de encriptación.

1.3.2 Específicos

- Realizar la encriptación de la voz de manera muy rápida para que el flujo de la comunicación no se vea afectada por retardos.
- Prevenir el fácil desencriptamiento de los datos, mediante otros medios que no sean los de este equipo.
- Construir un equipo portátil y de fácil conexión basado en firmware y hardware, y que permita un fácil manejo y sea de bajo costo.

1.4 Metodología de Investigación

- Método de Análisis: inicia por la identificación de cada una de las partes que caracterizan una realidad, para este caso el análisis de la situación actual de intercepción telefónica. De esa manera, se establece la relación causa - efecto, donde la causa es la inseguridad de la comunicación y el efecto son las intercepciones entre los elementos que compone el objeto de investigación.
- Método Experimental: con el fin de desarrollar el proyecto no solo desde el punto de vista de investigación, sino también como aplicación implementado en hardware, para así demostrar físicamente su funcionamiento.

1.5 Terminal Telefónico

Hay una gran variedad de terminales telefónicos que funcionan en las redes de comunicaciones de diversas tecnologías, modelos y funcionamiento tales como los teléfonos fijos, que funcionan de manera análoga o digital, y los teléfonos móviles. En telefonía fija se suele utilizar mucho la comunicación análoga, esto entre abonado y central; mientras que entre central y central se utiliza todo en digital. Primero se debe entender cómo funciona la realización de una llamada y todas las señalizaciones que se suscitan en todo el proceso de comunicación.

1.5.1 Funcionamiento de un Teléfono Fijo

El teléfono permite la comunicación bidireccional entre dos personas distantes mediante el uso de un auricular y un micrófono, los cuales permiten al usuario realizar una conversación cuya señal de voz será transmitida mediante una línea de transmisión de 2 hilos hacia el otro usuario con el cual se está comunicando. La señal de voz es transmitida en banda base sin ninguna modulación o codificación previamente realizada.

La comunicación se inicia descolgando el teléfono lo cual genera una impedancia en la línea telefónica y se genera una corriente como se aprecia en la figura 1, y la central telefónica genera una señal de tono de espera a que el usuario realice la marcación del número telefónico al cual se desea comunicar.

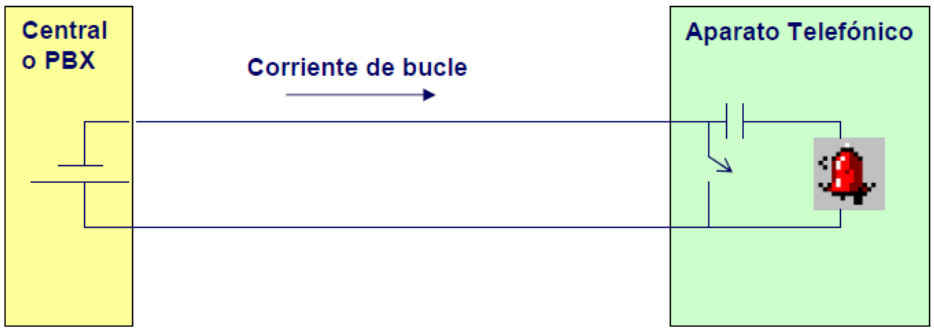


Fig. 1: Generación de Corriente de la Central para Inicio de Transmisión [4]

El teléfono genera los tonos de marcación la cual es recibida por la central telefónica y realiza la conexión con el usuario con el cual se quiere comunicar, a la vez se genera un

enlace directo punto a punto con el otro usuario y a partir de este momento se puede realizar la transmisión de las señales de voz de un extremo a otro.

1.5.2 Señalización entre Centrales y Teléfonos Analógicos

La señalización es algo fundamental en toda comunicación entre teléfonos, la cual ha permanecido desde los inicios del teléfono, y esto debido a que permite identificar las diversas acciones que tiene una comunicación, como por ejemplo realización de una llamada, marcación, timbrado, llamada en espera, etc; para los cuales se requiere el uso de ciertos protocolos para poder identificar o realizar cada suceso de acuerdo a lo que acontece en la línea de comunicación y en el terminal.

Mediante la señalización podemos recibir la siguiente información: [4]

- Solicitud de inicio de una conversación.
- Indicar con que abonado se desea comunicar.
- Indicación del progreso de la llamada (timbrado, ocupado, descolgado, etc).
- Indicar recepción de llamada.

1.6 Codificación de la Señal Analógica

Para poder procesar la señal de voz de manera digital se debe primero digitalizar la voz para así poder trabajar la señal a nivel de bytes, para esto realizaremos un proceso de codificación que permitirá transformar la señal análoga en digital.

Para temas de señales en telefonía la UIT (Unión Internacional de Telecomunicaciones) recomienda el estándar G.711 el cual es usado para compansión de señales de audio en especial para telefonía. Es un método de codificación también llamado PCM (Pulse Code Modulation) el cual es muy utilizado para termas de codificación de señales de audio. [5]

El estándar G.711 es un codificador de banda angosta el cual entrega una señal con un ancho de banda de 64kbps. Trabaja con señales de audio en el rango de 300 – 3400 KHz (ancho de banda de las señales de voz) y realiza el proceso indicado en la figura 2.

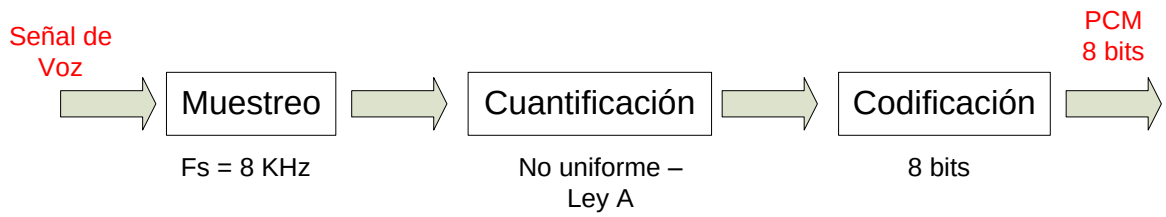


Fig. 2: Codificación Digital – Ley A

1.6.1 Muestreo de la Señal

Todo comienza con el muestreo de la señal en muestras de 8 bits a una frecuencia no menor al doble de la máxima frecuencia de la señal (para señales de voz se utiliza una frecuencia de muestreo de 8KHz = 8000 muestras por segundo). Ver figura 3.

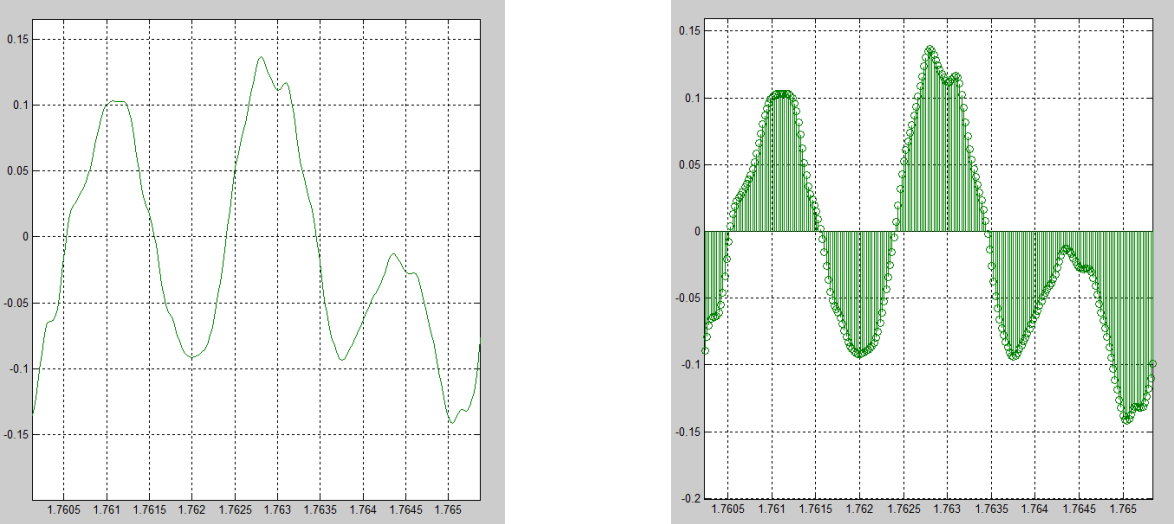


Fig. 3: Muestreo de la Señal

Este proceso da como resultado la señal en muestras de intervalos de 0.125 ms, las cuales son trabajadas una a una.

1.6.2 Cuantificación

En este proceso se le atribuye a la señal muestreada unos niveles con valores de diferencia de amplitud constantes o variables los cuales son llamados niveles de cuantificación. Lo que hace el cuantificador es aproximar el valor de cada muestra al valor del nivel más cercano a este.

Existen 2 modos de cuantificación:

- Uniforme: Utiliza niveles de igual espaciamiento entre ellos.
- No uniforme: Utiliza niveles de diferente espaciamiento entre ellos

En una cuantificación uniforme al tener la misma amplitud entre niveles tanto para señales de mayor como para las de menor valor, esto ocasiona lo que se denomina error o ruido de cuantificación.

El error de cuantificación es generado al aproximar el valor de una muestra dada a un nivel de cuantificación más cercano, lo cual produce un delta de variación entre la señal de entrada y la señal cuantificada.

Debido al error de cuantificación se ve afectado la calidad de la señal (S/N = Signal/Noise) de las muestras que ingresan al cuantificador, ya que se genera un mayor error de cuantificación en las muestras de menor amplitud a diferencia de las muestras de mayor amplitud ocasionando que la calidad de la señal de las muestras de mayor amplitud sea mejor que las de menor amplitud, es decir S/N = variable.

Es debido al error de cuantificación que se recomienda trabajar mediante 2 formas:

- Aumentando los niveles de cuantificación: al subir el número de niveles disminuiremos el ruido de cuantificación pero vamos a requerir un mayor número de bits lo cual generaría un mayor ancho de banda.
- Usando cuantificación No Uniforme: se asigna un número determinado de niveles los cuales tienen una mayor apertura para las señales de mayor valor mientras que para las señales de menor valor se utiliza un mayor número de niveles.

En este caso, para esta tesis, se utilizó una cuantificación no uniforme debido a sus ventajas frente a la otra opción.

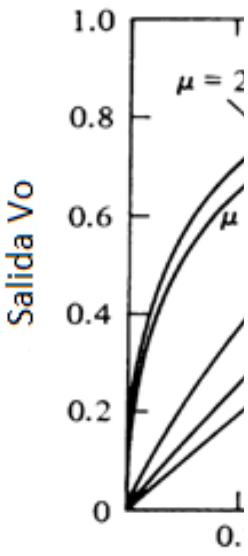
La cuantificación no uniforme consiste en hacer pasar la señal por un compresor y luego aplicar a la señal comprimida un cuantificador uniforme, en donde al final se tiene S/N = constante.

La norma G.711 asigna 2 leyes o algoritmos para temas de compresión:

- Ley μ: Usado en América del Norte y Japón. [6]

…….. (1)

En la ecuación (1) observamos que Vo y Vi vienen a ser los niveles de voltaje de salida y entrada respectivamente y μ un número entero positivo el cual mientras más grande sea, mejor será la cuantificación de la señal sin aumentar el error de cuantificación. Ver figura 4.



$$|Vo| = \frac{\ln(1 + \mu * |Vi|)}{\ln(1 + \mu)}$$

Fig. 4: Característica Gráfica de la Ley μ [6]

- Ley A: usado en Europa y el resto del mundo. [6]

…… (2)

En la ecuación (2) observamos que Vo y Vi vienen a ser los niveles de voltaje de salida y entrada respectivamente y A un número entero positivo el cual toma comúnmente el valor de A=87,5. Ver figura 5.

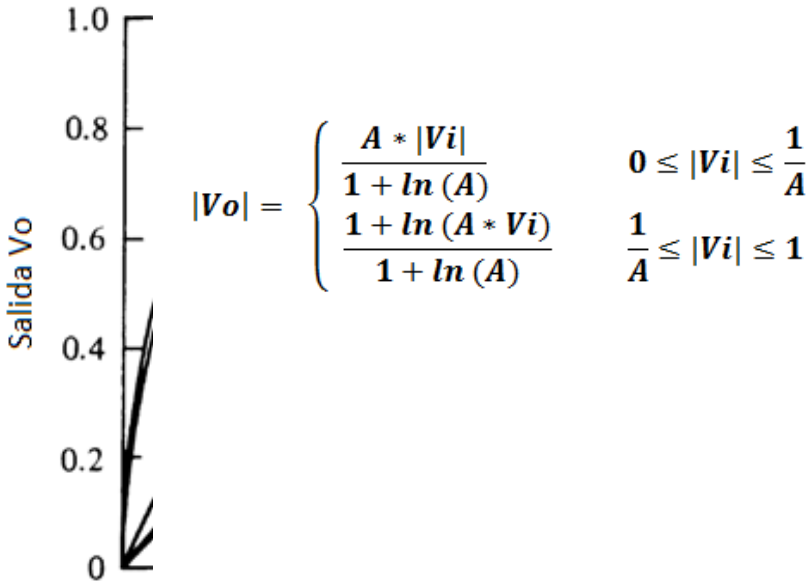


Fig. 5: Característica Gráfica de la Ley A [6]

Luego de pasar la señal muestreada por el compresor se pasa a aplicar a la señal una cuantificación uniforme la cual consiste en asignar cierta cantidad de niveles de cuantificación.

1.6.2 Codificación

Este proceso se encarga de asignar a cada nivel un valor binario el cual dependerá de la cantidad de niveles que se tenga al haber aplicado la cuantificación uniforme.

La cantidad de niveles dependerá de la cantidad de bits con la que deseamos desarrollar el proceso de encriptación:

$$\#\,\boldsymbol{Niveles} = \mathbf{2}^{\boldsymbol{N}}$$

Donde N viene a ser el número de bits que vamos a utilizar y se debe tomar en cuenta que el bit más significativo debe representar el signo de la muestra (positivo o negativo) y los bits restantes el del valor de la señal cuantificada.

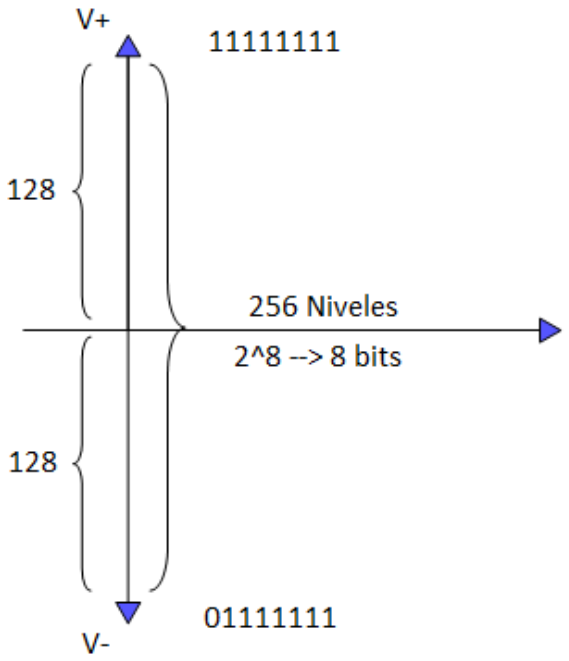


Fig. 6: Gráfica de Niveles de Cuantificación

En la figura 6 observamos que en el caso que utilicemos trabajar con 8 bits la cantidad de niveles de cuantificación con la que se estaría trabajando sería:

$$\mathbf{N = 8 \ \rightarrow \ 2^8 = 256}$$

Por lo que estaríamos trabajando con 256 niveles de cuantificación, donde 128 niveles corresponderán para la parte positiva de la señal y los otros 128 niveles corresponderán a la parte negativa de la señal y donde a cada nivel se le asigna un número binario donde el MSB para la parte positiva será "1" y el MSB para la parte negativa será "0", y los demás bits iniciaran la cuenta en descendente a partir del origen hacia sus límites correspondiente de la señal.

1.7 Transmisión de Datos por la Línea Telefónica

Los datos que se envían por la línea telefónica deben ser señales analógicas pero la señal de voz encriptada es digital por lo cual debemos transformar esta en una señal analógica para así enviar la señal hacia el otro extremo de la línea, en este caso la central telefónica.

Para los casos de transmisión de datos mediante la línea telefónica se recomienda el uso de módems, los cuales modulan la señal digital a una señal analógica modulada permitida en telefonía.

1.7.1 Modem y Recomendación v.92

El modem es un dispositivo que permite la conexión entre 2 terminales remotos mediante el uso de la línea telefónica y permite el intercambio de información entre sí. El modem es utilizado principalmente para la conexión de las computadoras a internet o teléfonos digitales a la central telefónica, en los cuales se requiere convertir sus señales digitales a señales analógicas para poder enviar su información por la línea telefónica

El modem convierte la señal digital en señal analógica para poderla transmitir por medio de la línea telefónica. Esta conversión se realiza mediante la modulación, la cual es una modulación digital donde puede variar su frecuencia, amplitud o fase de acuerdo al modem y la recomendación que utiliza.

Existen diferentes métodos de modulación las cuales han ido variando de acuerdo a los avances en la transmisión de datos digitales así como también las mejoras en la velocidad de transmisión. Los diferentes métodos de modulación y velocidades de transmisión

dependerá tanto de la aplicación y la correspondiente norma que indique la UIT para su respectivo funcionamiento.

Una de las recomendaciones últimamente usada es la v.92 la cual trabaja con una modulación QAM y a velocidades altas tanto en subida como en bajada.

1.7.2 Modulación QAM

La modulación QAM es una modulación lineal que se basa en modular en doble banda lateral 2 portadoras de una misma frecuencia que se encuentran desfasadas 90°. Cada portadora es modulada por una de las dos señales a transmitir y al final ambas señales son sumadas y esta es la señal que se transmite del modulador.

QAM es una modulación analógica por cuadratura que combina la modulación PSK y ASK, es decir la salida estará modulada tanto en fase como en amplitud la cual trabaja con entradas digitales (binarias) y salidas analógicas, las cuales tendrán N cantidad de estados de modulación con variaciones en fase y amplitud y se define a cada uno con N-QAM.

La cantidad N de estados dependerá de la cantidad de bits de entrada al modulador:

$$\boldsymbol{N = 2^b}$$

b = número de bits de entrada

N = cantidad de estados de modulación

Con el uso de 4 bits de entrada origino 16 diferentes estados de modulación, los cuales dependerán de un diagrama que esquematiza el funcionamiento de la modulación QAM llamado “Diagrama de Constelación”.

El diagrama de constelación representa en 4 cuadrantes N cantidad de puntos de constelación los cuales están separados equidistantemente uno de otro y donde cada uno representa un estado de modulación de acuerdo a su ubicación en el cuadrante. Ver figura 7.

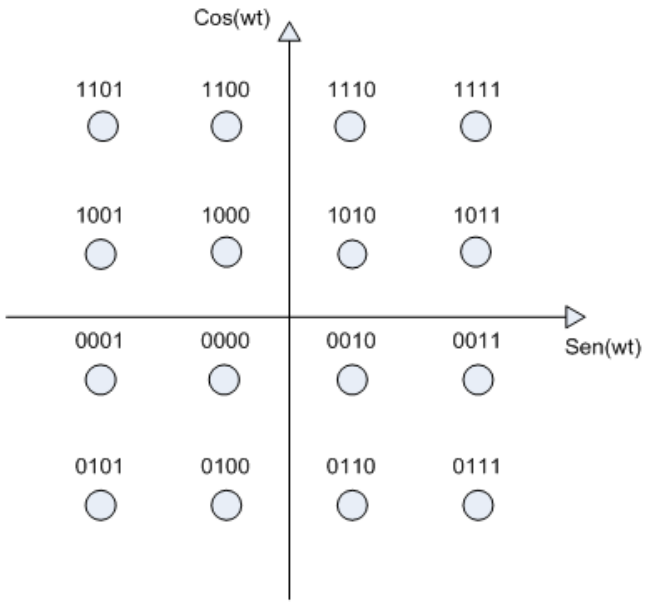


Fig. 7: Diagrama de Constelación de 16-QAM

La figura 8 muestra el funcionamiento de detallado del proceso de modulación QAM para una señal de entrada de 4 bits.

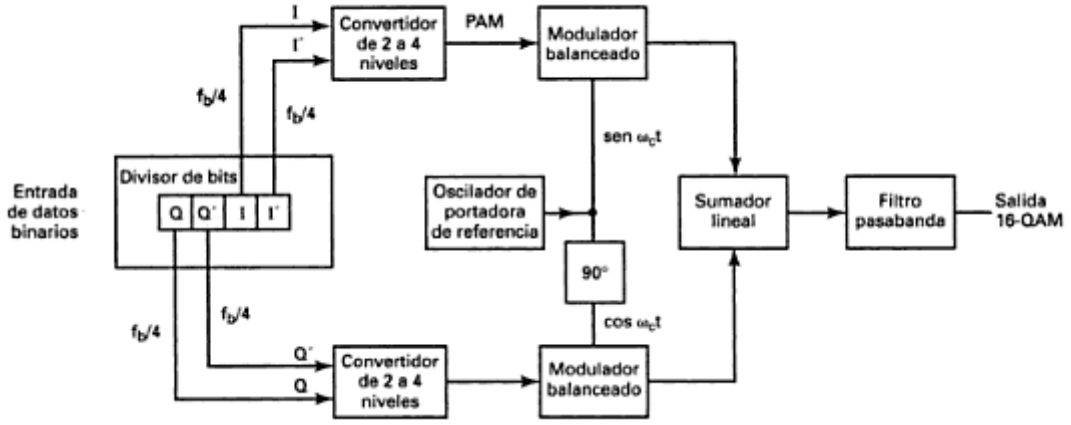


Fig. 8: Diagrama de Bloques de una Modulación 16-QAM [7]

En el Anexo 1 se puede observar la tabla con los valores en amplitud y fase que obtendrá la señal a la salida del modem de acuerdo a los niveles de entrada en bits.

1.8 Algoritmo de Encriptación

La encriptación es una manera de codificación que se le realiza a una señal digital para así convertir esta señal en una nueva, de manera que no pueda ser reconocible al momento de escucharla. Aunque es una forma de codificación, a diferencia de los métodos de codificaciones actuales que se les realiza a las señales en banda base que son usados para transmisiones que codifican la señal, dando una nueva señal codificada la cual siempre va a ser la misma para esa señal original. En la encriptación la señal codificada puede ir variando muchas veces aunque sea la misma señal original y esto debido a que la encriptación no solo se basa en el algoritmo a usar, sino también de lo que se le llama "llave" o "clave", la cual permite que el proceso de cifrado y descifrado dé bloques sea la correcta para obtener las señales exactas en ambos procesos.

La base de toda encriptación es el algoritmo que se utiliza el cual tiene como función la de codificar la información para que sea indescifrable a simple vista, de manera que una palabra cualquiera como “hola” pueda equivaler a: "F4R6vb" o bien a "dr45GT". El trabajo del algoritmo es precisamente determinar cómo será transformada la información de su forma original a otro que sea difícil de descifrar.

Una vez que la información llega a su destino final, se aplica el algoritmo de manera inversa al contenido codificado " F4R6vb" o bien a " dr45GT" y desencriptarlo en la palabra "hola" o según sea el caso. Hoy en día los algoritmos de encriptación son ampliamente conocidos (sea el caso de algoritmos estándar), es por esto que para prevenir a un usuario no autorizado el descifrar información encriptada, el algoritmo utiliza lo que es denominado “llave” para controlar la encriptación y desencriptación de la información que se desea transmitir.

Existen dos tipos de llaves que se utilizan para encriptar la información: llave pública y llave privada. La llave pública es dada a conocer a cualquier persona que lo desee y es utilizada únicamente para encriptar la información, mientras que la llave privada es solamente conocida por las personas involucradas en la transmisión y es utilizada tanto para la encriptación como para desencriptación.

El uso de estas llaves dependerá del tipo de encriptación que se desea utilizar, las cuales pueden ser:

- Encriptación simétrica.
- Encriptación asimétrica.

Para esta aplicación utilizaremos la encriptación simétrica la cual es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican deben de estar de acuerdo de antemano sobre la clave a usar, la

cual debe ser intercambiada mediante un canal seguro. Una vez que ambas tienen acceso a esta clave, el emisor cifra un mensaje usando esta llave, lo envía al destinatario, y éste lo descifra con la misma llave. La figura 9 muestra el proceso de cifrado utilizando la encriptación simétrica:

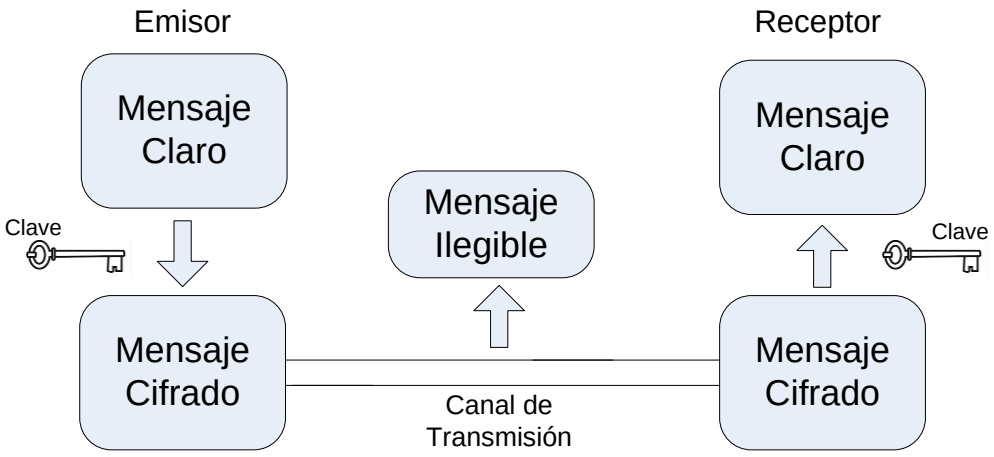


Fig. 9: Proceso de Encriptación Simétrica

Existen 2 tipos de cifrado simétricos:

a) Cifrado en Flujo:

Los cifradores de flujo son algoritmos de encriptación que pueden realizar el cifrado incrementalmente, convirtiendo el texto en claro en texto cifrado bit a bit. Esto se logra construyendo un generador de flujo de clave (secuencia pseudoaleatoria). Una secuencia pseudoaleatoria es una secuencia de bits de tamaño arbitrario que puede emplearse para oscurecer los contenidos de un flujo de datos combinando el flujo de clave con el flujo de datos mediante la función XOR. Si el flujo de clave es seguro, el flujo de datos cifrados también lo será.

b) Cifrado en Bloque:

Opera en grupos de bits de longitud fija (8, 16, 32, 64,…., etc bytes), llamados bloques, aplicándoles una transformación invariante. Cuando se realiza el cifrado, una unidad de cifrado por bloques toma un bloque de texto plano o claro como entrada y produce un bloque de igual tamaño de texto cifrado. La transformación exacta es controlada utilizando una segunda entrada, la clave secreta (llave privada). El descifrado es similar, se ingresan bloques de texto cifrado y se producen bloques de texto plano.

Las unidades de cifrado por bloques se diferencian de las unidades de flujo de cifrado en que un flujo de cifrado trabaja sobre dígitos individuales, uno después del otro, y la transformación varía durante el proceso de cifrado. La diferencia entre los dos tipos de unidades es algo difusa, dado que una unidad de cifrado por bloques puede ser operada en un modo que permite utilizarla como una unidad de flujo de cifrado, donde en lugar de dígitos se opera con bloques.

El algoritmo de encriptación que se ha de utilizar en este proyecto es el AES.

El algoritmo AES (Advanced Encryption Standar), llamado también algoritmo Rjindael, es un sistema simétrico de cifrado por lo que utiliza la misma clave para el proceso de cifrado como para el proceso de descifrado. Utiliza el método de cifrado por bloques y trabaja a nivel de bytes. Su diseño permite el uso de claves de sistema con longitud variable siempre que sea múltiplo de 4 bytes. La longitud de las claves utilizadas por defecto es de 128, 192 y 256 bits. Así también el algoritmo permite el uso de bloques de información con un tamaño variable siempre que sea también múltiplo de 4 bytes, siendo el tamaño mínimo recomendado de 128 bits (a mayor tamaño en bits de la llave y bloque de información, será más difícil el intento de descifrar la información de manera forzosa). [8]

El algoritmo se basa en aplicar una serie de rondas donde se va modificando la información original y donde en cada ronda se va generando un bloque llamado “matriz de estado”.

El algoritmo representa a la matriz de estado como una matriz rectangular que está formado por 4 filas y Nb columnas, donde Nb depende del tamaño del bloque. Ver ecuación 3. [8]

Nb = Tamaño de la Matriz de Estado / 32 …….…….. (3)

Por ejemplo una matriz de estado de 256 bits tendrá la siguiente matriz de estado. Ver tabla 2.

m0,0	m0,1	m0,2	m0,3	m0,4	m0,5	m0,6	m0,7
m1,0	m1,1	m1,2	m1,3	m1,4	m1,5	m1,6	m1,7
m2,0	m2,1	m2,2	m2,3	m2,4	m2,5	m2,6	m2,7
m3,0	m3,1	m3,2	m3,3	m3,4	m3,5	m3,6	m3,7

Tabla 2: Matriz de Estado [8]

Donde “m” vienen a ser los bytes correspondientes a las muestras que se toman de la señal de voz. Esta matriz de estado tiene un Nb = 8.

Al igual que la matriz de estado, la clave es una matriz rectangular de 4 filas y Nk columnas, donde Nk está dada por la ecuación 4.

Nk = Tamaño de la Clave / 32 ……(4)

Por ejemplo una clave de 128 bits se representa de la siguiente manera. Ver tabla 3.

k0,0	k0,1	k0,2	k0,3
k1,0	k1,1	k1,2	k1,3
k2,0	k2,1	k2,2	k2,3
k3,0	k3,1	k3,2	k3,3

Tabla 3: Matriz de Clave [8]

Donde "k" viene a ser los bytes de la llave se que ha de utilizar para el proceso de encriptación. La matriz de clave tiene un Nk = 4.

El análisis durante todo el proceso se realiza columna por columna y siguiendo la secuencia $m_{0,0}$, $m_{1,0}$, $m_{2,0}$, $m_{3,0}$, $m_{0,1}$……, $m_{3,7}$ para la matriz de estado y de igual manera $k_{0,0}$, $k_{1,0}$, $k_{2,0}$,……, $k_{3,3}$ para la clave.

La señal original pasa por una serie de rondas que se va modificando de acuerdo a 4 transformaciones donde se utilizan las subclaves generadas a partir de la clave en cada ronda. El número de rondas que realiza el algoritmo depende mucho del tamaño del bloque y de la clave, los cuales según los autores del algoritmo, recomiendan para tamaños entre 128 a 256 bits (con incremento de 32 bits) se puede definir al número de rondas Nr por la fórmula 5.

$Nr = max(Nk,Nb)+6$ ………………………………………. (5)

Donde se elije el número máximo entre Nk y Nb y le sumamos 6, para así obtener el número de rondas que efectuará el algoritmo durante todo el proceso de cifrado.

Durante el proceso de encriptación se utilizan 4 transformaciones que afectan directamente a la matriz de estado ocasionando que esta varíe sus bytes de acuerdo a la ronda correspondiente que indica el algoritmo.

Estas transformaciones se basan en aplicar a la matriz de estado operaciones lógicas, substituciones y métodos matemáticos realizados previamente por los autores del algoritmo. Las 4 transformaciones que se utilizan durante el proceso son:

1.8.1 Función ByteSub

La función ByteSub se encarga de reemplazar el valor de cualquier posición de la matriz por un valor dado de una tabla de sustitución denominada S-box ya predeterminada por los autores del algoritmo. Ver tabla 4.

		Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	10	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	B0	80	54	BB	16

Tabla 4: Tabla de Sustitución - Proceso de Cifrado [8]

Para saber por cual valor de la tabla debemos reemplazar, se toma el byte de la matriz de estado al cual se le va a aplicar la función y a este valor seleccionado de 8 bits se le separa en 2 partes de 4 bits cada uno donde los 4 MSB indican la fila denominados por “X” y los 4 LSB indican la columna denominados por “Y”.

Para el caso del proceso de desencriptación se utiliza la tabla 5.

		Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Tabla 5: Tabla de Sustitución – Proceso de Decifrado [8]

1.8.2 Función ShiftRow

Esta función se encarga de desplazar cierto número de posiciones cada fila de la matriz de estado hacia la izquierda dependiendo del tamaño de la matriz de estado con la cual se está trabajando. Ver tabla 6. [8]

Tamaño de Matriz de Estado	Fila 0	Fila 1	Fila 2	Fila 3
128 bits (Nb = 4)	0	1	2	3
192 bits (Nb = 6)	0	1	2	3

256 bits (Nb = 8)	0	1	3	4

Tabla 6: Desplazamiento de Acuerdo al Tamaño de la Matriz de Estado [8]

En el proceso de desencriptación, la función inversa simplemente realiza el desplazamiento hacia la derecha que es el sentido contrario al de encriptación.

1.8.3 Función MixColumns

Esta función consiste en multiplicar cada columna de la matriz de estado por una matriz dada por los diseñadores del algoritmo de acuerdo a ciertos criterios matemáticos. La matriz que realiza la multiplicación esta dado por la figura 10 para la encriptación y la figura 11 para la desencriptación. [8]

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Fig. 10: Matriz de Multiplicación del Proceso de Encriptación [8]

$$\begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix}$$

Fig. 11: Matriz de Multiplicación del Proceso de Desencriptación [8]

La multiplicación que se realiza entre cada byte no es una típica multiplicación matemática, esta tiene una forma especial de realizarse. En el caso de multiplicar un byte cualquiera por 2 simplemente es desplazado una posición a la izquierda y en caso sobrepase el valor de 255 se realiza una operación XOR con 0x11B. En caso se trate de multiplicar por 3, primero se realiza una multiplicación por 2 como se indicó anteriormente y luego una operación XOR con el valor original. Y así sucesivamente se pueden ir generando varias secuencias de acuerdo al valor por el cual se desea multiplicar. [8]

1.8.4 Función AddRoundKey

La función addroundkey se encarga simplemente de realizar una función XOR entre los valores de la matriz de estado y los de la tabla de subclave que corresponde a dicha ronda, la cual ha sido generada previamente en base a la llave de encriptación. [8]

Todas estas funciones serán aplicadas a la matriz de estado en un orden establecido y dependiendo de la ronda que se encuentre el proceso. En la figura 12 se puede ver el proceso general del algoritmo:

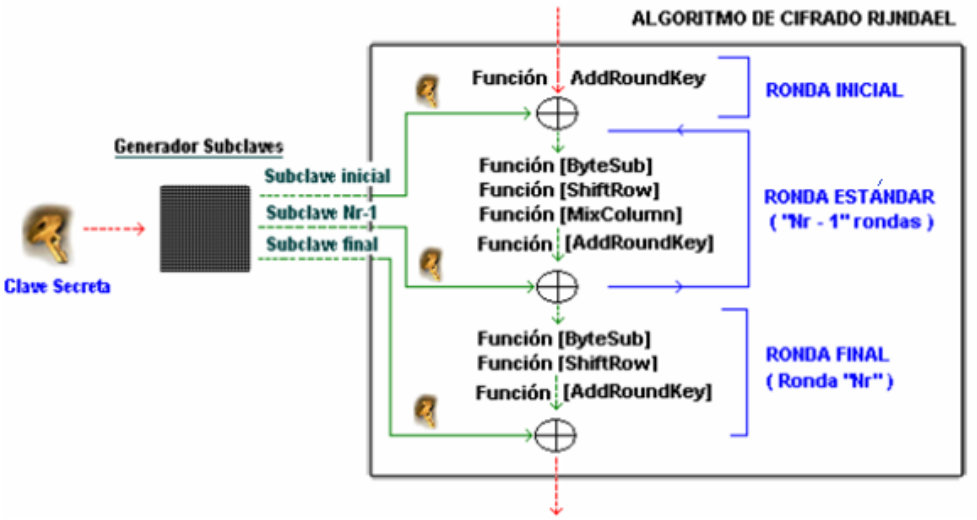


Fig. 12: Descripción del Proceso del Cifrado [8]

2.1 Descripción del Sistema

En la figura 13 se muestra el diagrama de bloques del sistema de encriptación planteado, que a su vez describe las etapas principales del sistema.

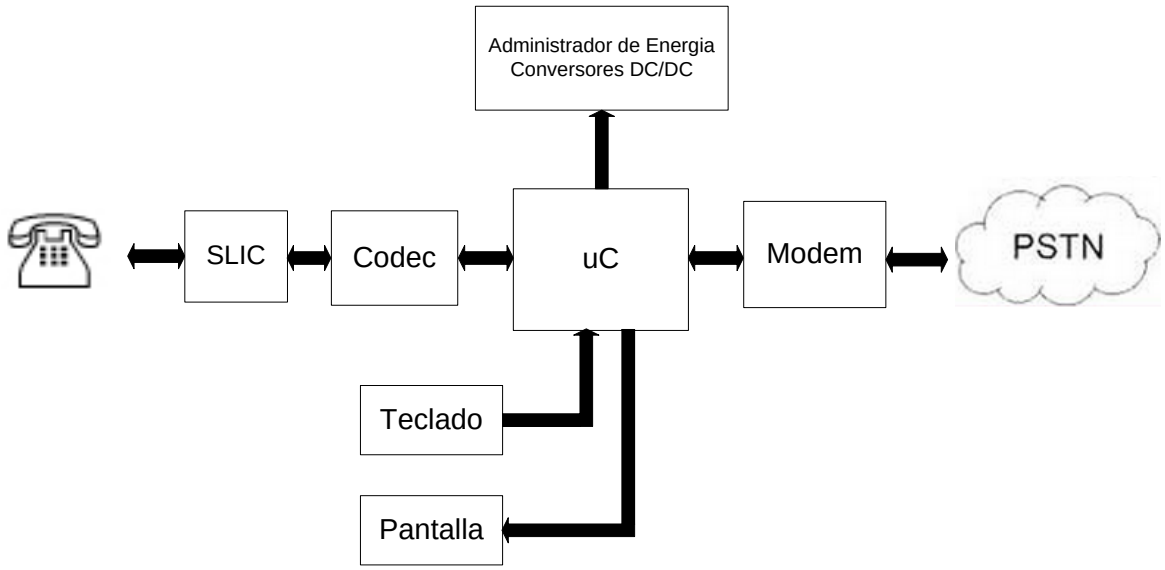


Fig. 13: Diagrama de Bloques del Sistema

La señal a la salida del teléfono fijo se hará pasar por un conversor de 2 a 4 hilos (SLIC), el cual separa la señal de transmisión de la señal de recepción. El SLIC tiene un circuito conversor DC/DC que se encarga de generar el voltaje necesario que requiere tener la línea de transmisión para las señales que se encuentran en estas.

Luego de separadas las señales, se hace pasar por un filtro pasa bajas y un amplificador, debido a que la señal de voz es menor a 1 V y se requiere amplificar, para luego pasar por un codificador con cuantificación no uniforme, utilizando ley A o ley u, el cual genera la señal digital en PCM.

La señal PCM es direccionada hacia el microcontrolador para que se encargue del proceso de encriptación de la señal de voz digitalizada en base a una "llave" que se hace ingresar de manera manual mediante el teclado hacia el microcontrolador. Y finalmente la señal de voz encriptada pasa por el modem para que realice el proceso de modulación de la señal para que se pueda transmitir por la línea telefónica de acuerdo a las normas de la UIT.

El sistema cuenta con una pantalla LCD para permitir la visualización del menú que permite seleccionar si se desea pasar a realizar una llamada segura o mantener la línea de comunicación normal, ingresar la llave de encriptación, etc.

Como se indicó anteriormente el sistema de encriptación se acopla a la salida del teléfono, lo cual se realiza en ambos extremos de la comunicación emisor receptor, donde dicha comunicación pasa por la central telefónica el cual realiza el bypass de la señal, lo cual para esto se debe de haber tenido abierta la señal de un extremo a otro, lo cual requiere que antes de iniciar el proceso de encriptación se realice una llamada telefónica de manera "no segura" es decir de manera normal para así poder tener la línea abierta de un extremo a otro.

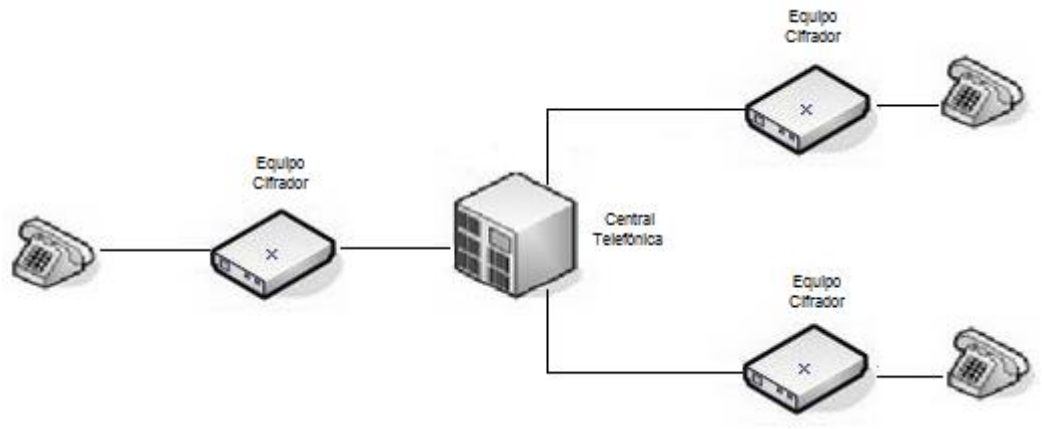


Fig. 14: Enlace de un Terminal Telefónico a Otro(s) con Conexión al Equipo Cifrador

La figura 14 muestra como es la conexión de todo el sistema de un extremo a otro de la comunicación, apreciando que un teléfono se puede comunicar con varios siempre y cuando tengan un equipo cifrador y tengan la misma llave de cifrado.

Cuando el enlace sea de manera “no segura”, es decir no se encripta la señal de voz, el equipo cifrador solamente deja pasar la señal de voz sin realizar ninguna modificación lo cual dura hasta que el usuario indique que desea inicializar una comunicación “segura” es decir encriptada e ingrese la llave que se utiliza en ambos extremos del enlace.

A continuación se describen las partes del sistema:

2.1.1 SLIC

El SLIC (Subscriber Line Interface Circuit) es un circuito integrado que permite la separación de la señal de transmisión y de la señal de recepción de la línea telefónica de manera independiente una de otra, lo cual permite un mejor manejo y acoplamiento de la

cualquier señal que se desea transmitir o que se desea recibir; es decir, convierte la señal de 2 hilos a 4 hilos donde cada par es para transmisión y recepción respectivamente. [9]

También permiten adaptar los niveles de voltaje necesario para la línea telefónica, tanto como la tensión DC con la que trabajan las líneas telefónicas como también contralar los niveles de amplitud de tensión y corriente que se generen al envío de datos por la línea telefónica. [9]

2.1.2 Codec/Decodec

Circuito integrado que convierte la señal analógica que entrega el SLIC, la cual es la señal recibida de la línea telefónica, en una señal en PCM; mientras que el decodificador se encarga de convertir la señal digital en PCM que entregue el microcontrolador a una señal analógica que es direccionado al SLIC. [9]

2.1.3 Microcontrolador

Circuito integrado que trabaja como el corazón del sistema que cuenta con las características necesarias: CPU, memoria, unidades de Entrada/Salida.

Debido a que el encriptador trabaja con otros componentes e interfaces, el microcontrolador se encarga de controlar las demás partes del sistema mediante el uso de periféricos y protocolos de comunicación con pantallas, teclados y otros chips que forman el sistema.

2.1.4 Modem

Socket que lleva toda la circuitería necesaria para realizar el proceso de modulación el cual modula la señal digital a la salida del microcontrolador y es llevada a la línea telefónica para la transmisión de datos de acuerdo a una previa configuración de enlace mediante el uso de comandos AT las cuales servirán para controlar el modem.

2.1.5 Pantalla LCD

Permite la visualización de los menús del sistema para elegir las opciones adecuadas como ingreso de clave, conexión o desconexión de la comunicación de cifrado.

2.1.6 Teclado Matricial

Permite ingresar la selección de los menús e ingreso de la clave que se requerirá para procesar la encriptación de la señal.

2.1.7 Batería

Este componente es utilizado para proporcionar la energía que requiere el sistema y para que le equipo funcione de manera portátil.

2.2 Selección de Componentes

2.2.1 SLIC

El SLIC es el dispositivo que permite convertir los 2 hilos de la señal telefónica en 4 hilos que son 2 de transmisión y 2 de recepción. A continuación en la tabla 7 se muestran 3 opciones de dispositivo:

Modelo	Si3210	HC55185DMIZ	MTIFM.R3-SP
Niveles de voltaje programable	Si	No	Si
Limitador de corriente	Si	Si	Si
Filtro	Si	Si	Si
Amplificador	Si	Si	Si
Codificador	Si	No	Si
Empaque	TSSOP-38	PLCC-28	Socket
Precio (US $)	9.2	8.28	69.69

Tabla 7: Comparación de SLICs

En este proyecto se decidió utilizar el ProSLIC Si3210 de la familia de Silicon Labs, el cual es un chip que provee una completa interface para teléfonos análogos. Está provisto de un SLIC, codificador de voz y generador de nivel de voltaje.

Contiene las siguientes características: [9]

- Parámetros de línea programables
- Frecuencia, amplitud y forma de onda programable a la salida de la línea telefónica
- Salida dinámica de 0 a 95V DC a la salida de la línea telefónica.
- Generador de señal de ring, ocupado y DTMF.
- Codificador y decodificador PCM (ley u y ley A - 16 bits)
- Consumo de corriente = 33mA

La selección del ProSLIC3210 se debe a que integra no solo el bloque SLIC, como es lo habitual en estos dispositivos, sino también la parte de codificación PCM que se encarga de realizar la conversión de la señal de análoga a digital. Proporciona una señal serial codificada/decodificada el cual se sincroniza mediante una señal de reloj.

2.2.2 Microcontrolador

El procesador es el núcleo del sistema, el cual realiza todo el procesamiento de encriptación de la señal codificada a la salida del Si3210 para encriptación y el proceso de desencriptación a la salida de la señal del modem.

Debido a que el procesador realiza tanto el proceso de encriptación y desencriptación de la señal, que en situaciones se tendrá que realizar en paralelo ya que la transmisión de señal telefónica es full-duplex, se realiza el procesamiento a una gran velocidad para que el tiempo que se demore en realizar el proceso de encriptación no se vea reflejada como un retardo al momento de la transmisión de la señal.

En la tabla 8 se muestran 3 microcontroladores con sus características principales y donde se muestra las semejanzas y diferencias las cuales ayudarán a una mejor elección del procesador a utilizar.

Modelo	DsPIC33FJ64GP	ATXMEGA64D3	TMS320F28032
FLASH (Bytes)	64	64	64
SRAM (Kbytes)	8	4	10
Ciclo de instrucción	25 ns	31.3 ns	16.6 ns
Numero de E/S	35	50	33
SPI	si	si	si
USART	si	si	si
Precio (US $)	5.44	3.63	8.46

Tabla 8: Comparación de Microcontroladores

Para poder elegir correctamente el microcontrolador a utilizar se deben tener en cuenta 2 criterios: costo y beneficio, donde para este caso analizamos lo primordial en beneficio la velocidad del procesador.

En la tabla apreciamos que las principales diferencias entre los 3 microcontroladores es la velocidad del ciclo de instrucción y el costo. El TMS320F28032 es el que genera una instrucción en un tiempo más corto a diferencia de los demás microcontroladores y a pesar que su costo es alto a comparación de los demás, se aprecia que no supera el doble de los demás precios y se compensa con su velocidad.

2.2.3 Modem

La selección del modem se basa principalmente en la versión con la cual deseamos trabajar de acuerdo a la velocidad que se enviarán y recibirán los datos por la línea telefónica.

Existen varias normas para el tema de módems, donde cada una varía de acuerdo a su velocidad de subida y bajada, las cuales han sido establecidas por la UIT para comunicación de datos mediante la red telefónica. Ver tabla 9.

Norma	Modulación	Velocidad
V.22bis	QAM	2.4 Kbps
V.26bis	PSK	2.4 Kbps
V.27ter	PSK	4.8 Kbps
V.32	QAM	9.6 Kbps
V.32bis	QAM	14.4 Kbps
V.34	QAM	28.8 Kbps
V.34+	QAM	33.6 Kbps
V.90	QAM	56 / 33.6 Kbps *
V.92	QAM	56 / 48 Kbps *

Tabla 9: Recomendación de la UIT - Modem [10]
(*) Velocidad de Subida / Velocidad de Bajada

Cada recomendación ha ido variando de acuerdo a su evolución en temas de velocidad de transmisión de datos las cuales son aplicables de acuerdo a la conexión que se haga entre algún ordenador y la central telefónica, ya que algunos requieren una velocidad mucho mayor de transmisión.

El diseño de un circuito que realice la modulación de la señal requiere de una gran cantidad de componentes para su funcionamiento lo cual lleva a un incremento en el costo y tiempo de diseño, por lo que se optó por elegir un socket que tenga toda la circuitería necesaria para la modulación.

El socket modem que se eligió fue el MT5692SMI-L-92.R1, que tiene las siguientes características: [11]

- Norma V.92, 56 Kbps de subida y 48 Kbps de bajada.
- Compatible con velocidades bajas.
- Interface serial asíncrona.
- Comunicación full dúplex
- Tecnología DAA
- Precio: US $ 35.57

2.2.4 Administrador de Fuentes de Alimentación

El sistema es alimentado en base a una batería Li-Po de doble celda con una capacidad total de 8.4 V y 2200 mAh, que tiene un tamaño pequeño y de fácil conexión.

Debido a que este sistema basa su alimentación en una batería, se debe de distribuir de manera eficiente los niveles de tensión y corriente necesario para cada bloque del sistema. Ver tabla 10.

	Voltaje	Corriente	Funcionamiento
SLIC	3.3 V	42.8 mA	Solo durante encriptación
Conversor DC-DC (SLIC)	5 V	600 mA	Solo durante encriptación
Microcontrolador	3.3 V	80 mA	Modo activo
Modem	3.3 V	115 mA	Solo durante encriptación
Pantalla LCD con Backlight	3.3 V	65 mA	Modo activo

Tabla 10: Niveles de Alimentación de los Bloques del Sistema (valores máximos)

La selección de reguladores depende de los niveles de voltaje y corriente necesario. Es necesario que el microcontrolador cuente con su fuente de alimentación propia debido a que siempre está energizado y será este quien active los reguladores de los periféricos para lo que se requiere que los reguladores de los periféricos tengan habilitadores.

El microcontrolador tiene su propio regulador, así como la pantalla LCD y el conversor DC-DC, mientras que el SLIC y el modem comparten el mismo regulador. A continuación se muestran opciones de reguladores en la tabla 11.

Modelo	LP2985-33DBVR	SPX5205M5-L-3-3	MCP1801T-3302I
Caída de voltaje	0.28 V	0.21 V	0.2 V
Corriente de salida	150 mA	150 mA	150 mA
Corriente de fuga	850 uA	70 uA	25 uA
Rango de temperatura	-40 °C ~ 125 °C	-40 °C ~ 80 °C	0 °C ~ 125 °C
Empaque	SOT-23-5	SOT-23-5	SOT-23-5
Costo	0.59	0.62	0.5

Tabla 11: Comparación de Reguladores (Microcontrolador y Pantalla LCD)

Según lo que apreciamos en la tabla anterior podemos observar que la mejor opción es el regulador MCP1801T-3302I, ya que tiene la más baja caída de voltaje entre la entrada y salida y una muy baja corriente de fuga lo que indica que no se perderá mucha corriente hacia tierra.

Esta opción de regulador seleccionada se utiliza tanto para el microcontrolador como para la pantalla. Para el SLIC y modem que comparten el mismo regulador se requerirá un regulador que proporcione como mínimo 200 mA. En la tabla 12 se muestran opciones de reguladores:

Modelo	LP2992AIM5	ADP3330ARTZ3.3	MAX8881EUT33

Caída de voltaje	0.45 V	0.14 V	0.1 V
Corriente de salida	250 mA	200 mA	200 mA
Corriente de fuga	1.5 mA	1.6 mA	1 mA
Rango de temperatura	-40 °C ~ 125 °C	-40 °C ~ 85 °C	-40 °C ~ 85 °C
Empaque	SOT-23-5	SOT-23-6	SOT-23-6
Costo	1.23	2.35	4.27

Tabla 12: Comparación de Reguladores (SLIC + Modem)

En este caso seleccionamos el LP2992AIM5, que a pesar de su alta caída de voltaje en comparación con los demás, proporciona más corriente y tiene un menor costo. Este regulador es utilizado para alimentar a la vez al SLIC y al modem.

Para el caso del conversor DC-DC se requiere de un regulador que proporcione min 600 mA a una salida de 5 V pero sin llegar a calentar debido a la potencia que entrega. Para evitar esto se requirió utilizar una fuente switching que puede llegar a proporcionar amperios a una carga sin que este aumente su temperatura a niveles críticos. Ver tabla 13.

Modelo	MCP16321T	TL2575-05N	LM2575D2T
Corriente de salida	1 A	1 A	1 A
Corriente de fuga	5.2 mA	5 mA	5 mA
Eficiencia	83 %	75 %	77 %

Empaque	16-QFN	16-PDIP	D2PAK-5
Precio (US $)	1.86	1.86	2.36

Tabla 13: Comparación de Reguladores Switching (Conversor DC-DC)

En este caso se eligió el regulador switching LM2575D2T ya que a pesar de tener menor eficiencia frente a la primera opción, su empaque ayuda a disipar mejor el calor cuando la switching entregue gran cantidad de corriente.

2.2.5 Diagrama Esquemático

La figura 15 nos muestra las fuentes de alimentación necesarias para cada dispositivo del sistema.

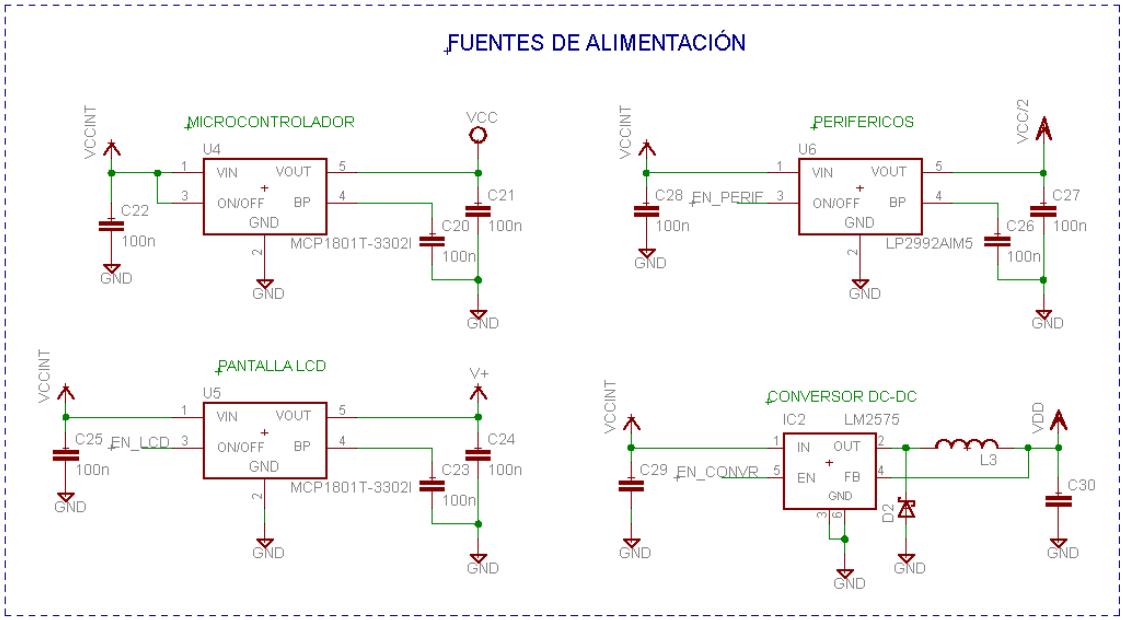


Fig. 15: Reguladores de Voltaje

La figura 16 nos muestra el circuito necesario para cargar la batería de 2 celdas.

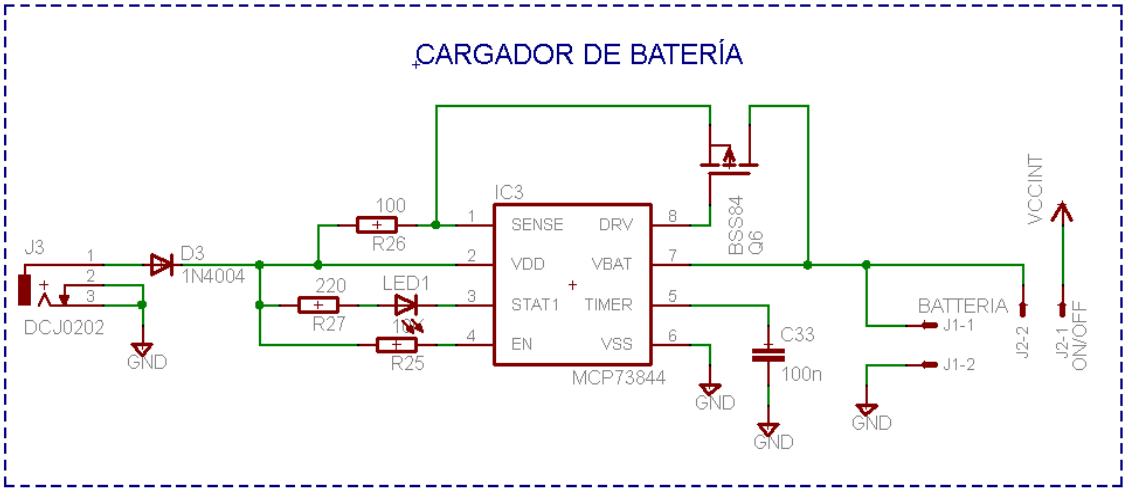
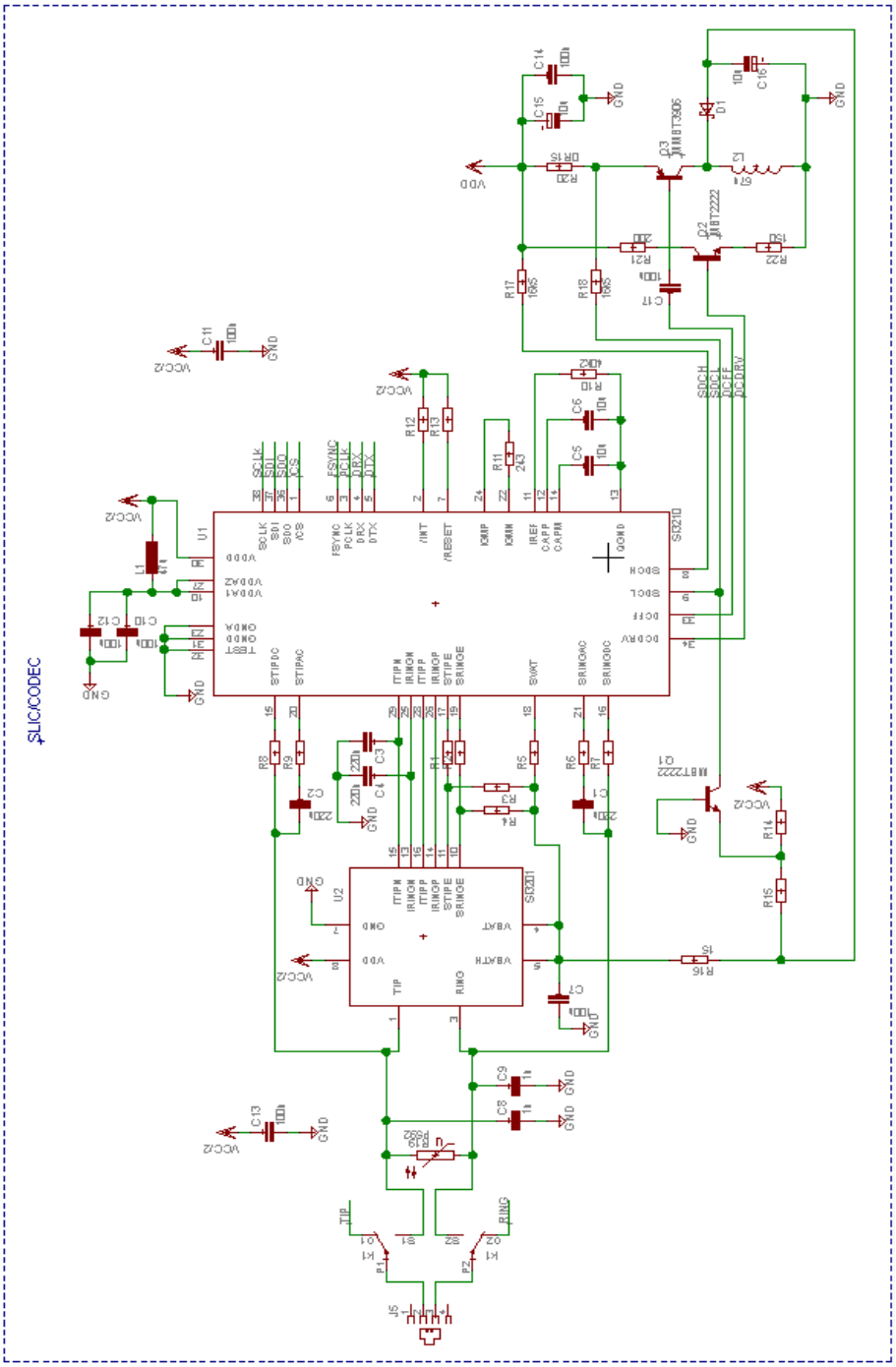


Fig. 16: Cargador de Batería

En la figura 17 se muestra el circuito del ProSLIC con sus respectivas señales digitales de salida y también el circuito DC-DC para el ajuste de los niveles de voltaje en la línea del teléfono.

Fig. 17: SLIC/Codec y Conversor DC-DC

En la figura 18 se muestra el microcontrolador a utilizarse con sus conexiones al teclado y la



pantalla LCD.

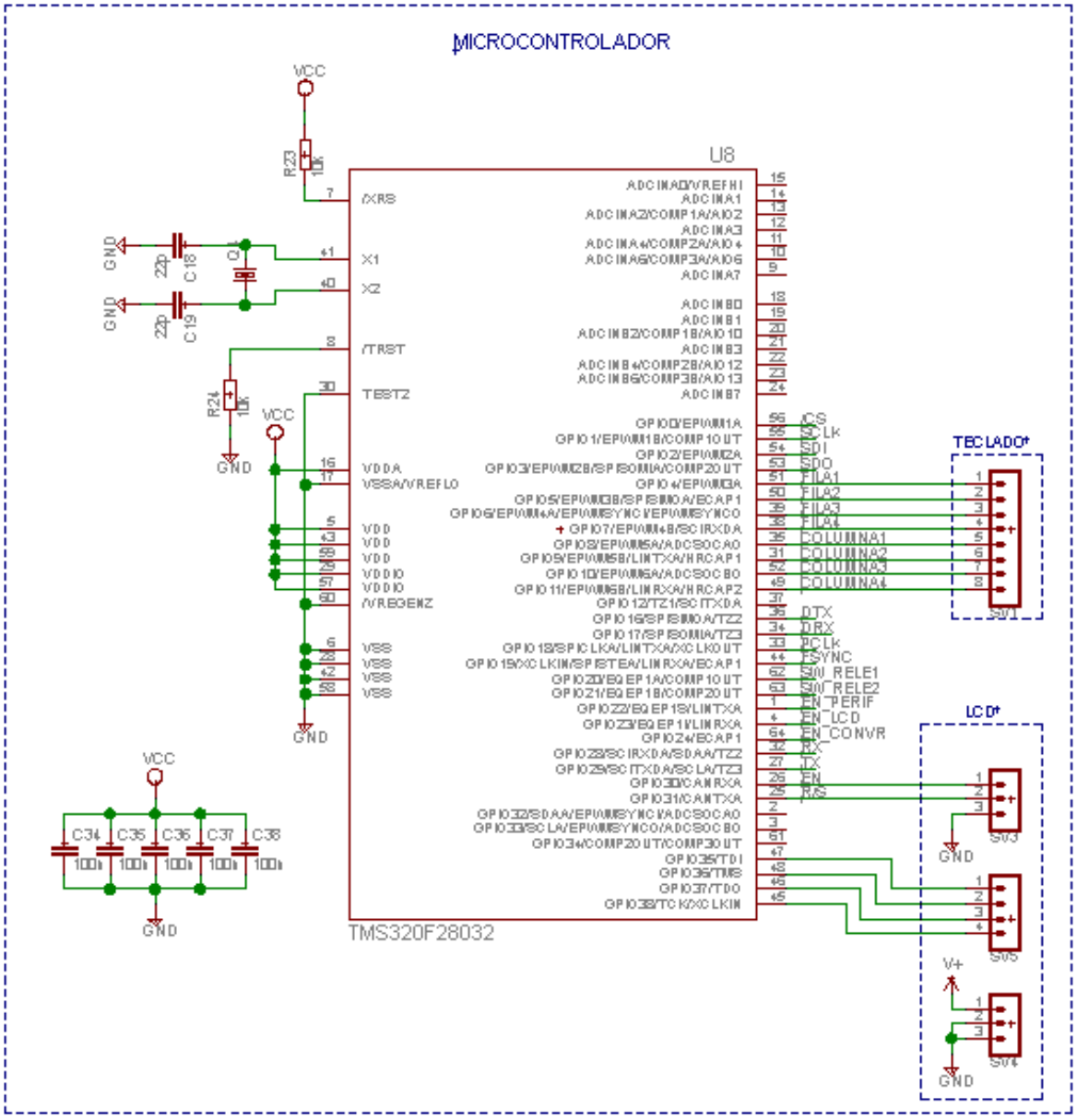


Fig. 18: Microcontrolador, Teclado y LCD

La figura 19 nos muestra el circuito de conexión del modem con el conector de la salida de la señal a la línea telefónica.

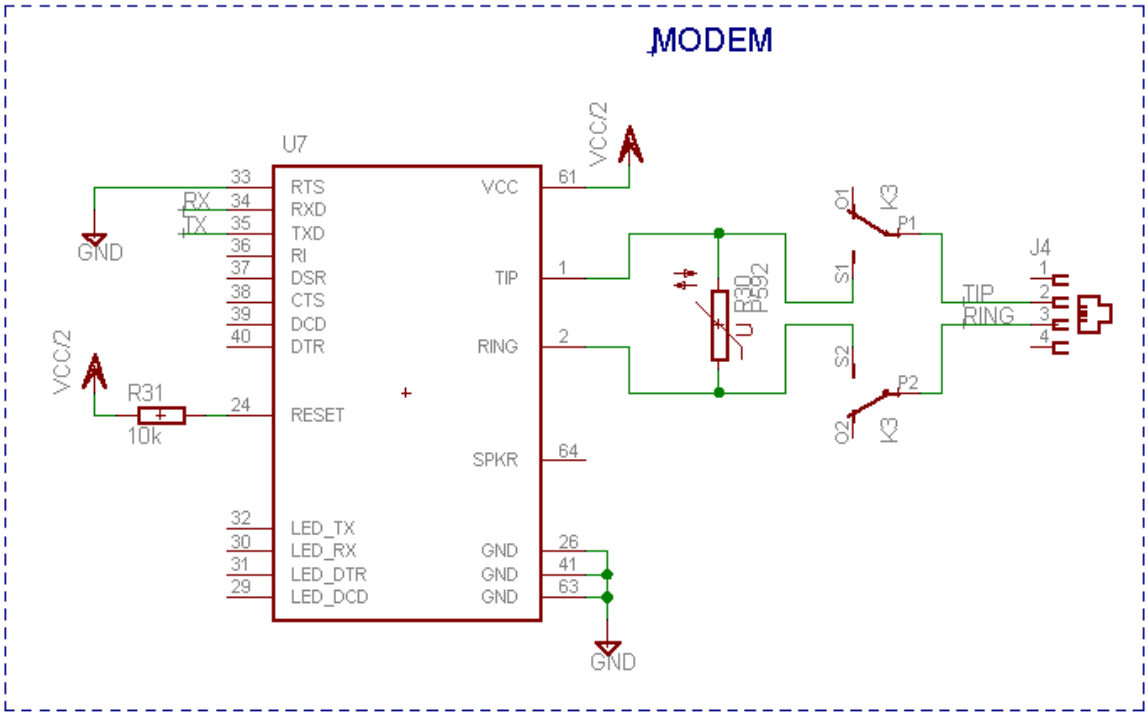


Fig. 19: Modem y Conexión a la Red Pública

La figura 20 muestra los conmutadores de la línea telefónica entre el modo seguro e inseguro

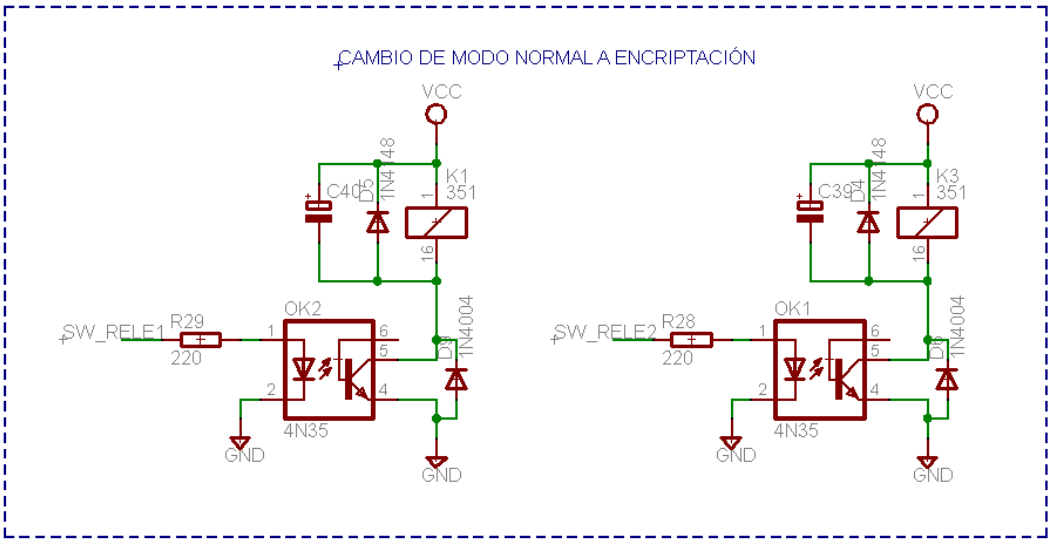


Fig. 20: Cambio de Modo Normal a Modo Cifrado

3.1 Conversión de 2 a 4 hilos y Codificación

Para el diseño y la configuración del ProSLIC Si3210 se deben tener cuenta ciertos criterios como los niveles de voltaje que se utiliza en la línea telefónica, la frecuencia de la señal de voz para realizar un correcto filtraje y la amplificación de la señal para generar una perfecta señal en PCM sin mucha distorsión en la señal debido al proceso de codificación.

Internamente el ProSLIC tiene un filtro tanto en la transmisión y la recepción de la señal, cuyas respuestas en frecuencia son las que se muestran en las figuras 21 y 22 respectivamente.

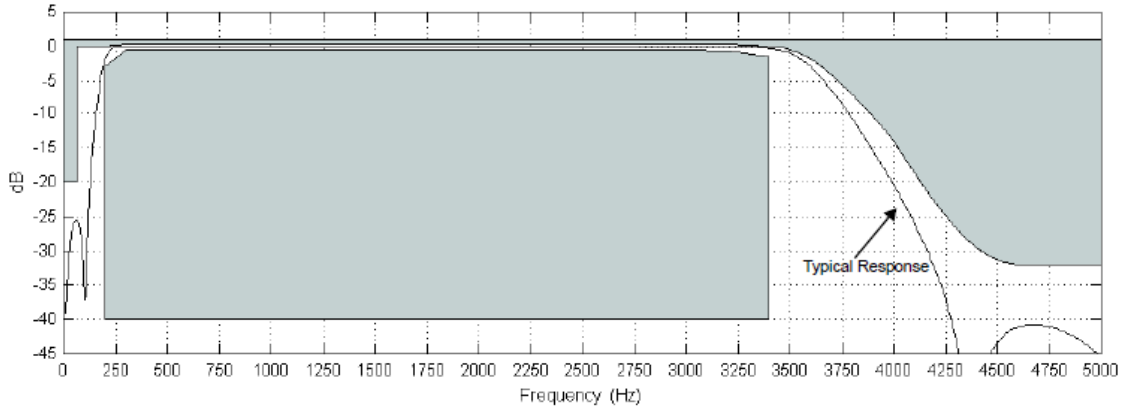


Fig. 21: Respuesta en Frecuencia en la Transmisión [9]

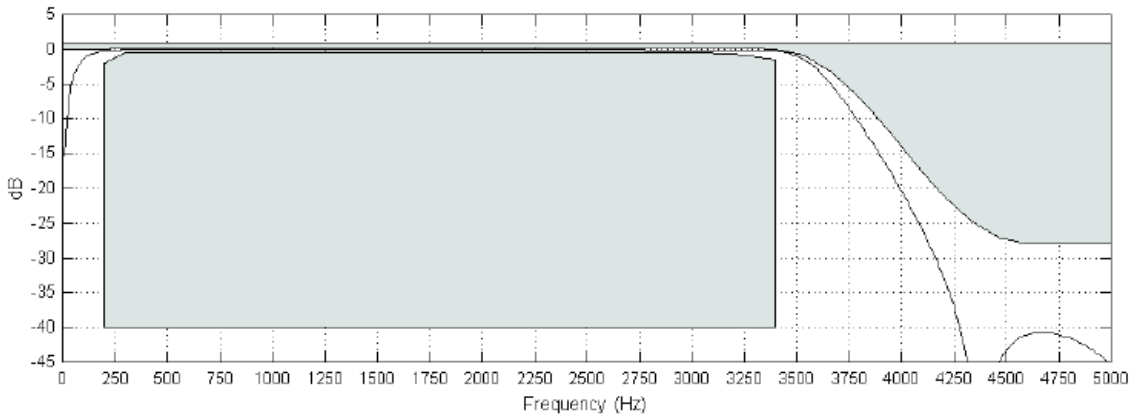


Fig. 22: Respuesta en Frecuencia en la Recepción [9]

3.1.1 Configuración de Conexión con el Si3201

El Si3210 trabaja junto a una interface de alimentación de línea el cual puede ser componentes discretos o un circuito integrado provisto de las características necesarias. En este caso se considera el uso del chip Si3201 el cual se usa para controlar los altos niveles de voltaje necesarios para el funcionamiento de la interface de línea en la línea telefónica.

El Si3201 es controlado mediante el mismo Si3210 en base a unos comandos de control los cuales indican el valor de los niveles de corriente y voltaje, los cuales son constantes durante el funcionamiento. Se define la zona de voltaje constante entre TIP y RING programable de 0V a 94.5 V y también la zona de corriente programable desde 20 mA a 41 mA. En la siguiente tabla se observa el rango de los valores programable para los niveles de voltaje y corriente y sus respectivos registros de configuración.

Parameter	Programmable Range	Default Value	Register Bits	Location*
I_{LIM}	20 to 41 mA	20 mA	ILIM[2:0]	Direct Register 71
V_{OC}	0 to 94.5 V	48 V	VOC[5:0]	Direct Register 72
V_{CM}	0 to 94.5 V	3 V	VCM[5:0]	Direct Register 73

Tabla 14: Características de Rangos Programables [9]

Como se aprecia en la tabla 14 el Si3210 tiene por defectos unos valores tanto para los niveles de voltaje como para el nivel de corriente. Estos valores se pueden modificar de acuerdo a la línea telefónica la cual se utiliza, ya que algunas pueden llegar a un nivel de voltaje de hasta los 55 VDC y el nivel de corriente depende del teléfono que se utilice ya que algunos se alimentan de baterías y la mayoría de la misma línea telefónica, lo cual ocasiona que le pida más corriente a la línea telefónica.

En este caso se configuró el Si3210 para que el Si3201 trabaje a un nivel de tensión de 45V (medidos en la línea telefónica de una casa) y que entregue y reciba un nivel de corriente de 25 mA.

LF[2:0]*	Linefeed State	Description
000	Open	TIP and RING tri-stated
001	Forward Active	$V_{TIP} > V_{RING}$
010	Forward On-Hook Transmission	$V_{TIP} > V_{RING}$; audio signal paths powered on
011	TIP Open	TIP tri-stated, RING active; used for ground start
100	Ringing	Ringing waveform applied to TIP and RING
101	Reverse Active	$V_{RING} > V_{TIP}$
110	Reverse On-Hook Transmission	$V_{RING} > V_{TIP}$; audio signal paths powered on
111	Ring Open	RING tri-stated, TIP active
Note: The linefeed register (LF) is located in direct Register 64.		

Tabla 15: Configuración de Operación [9]

En la tabla 15 se observa la configuración del estado del Si3201 con respecto a la conexión con la línea telefónica, que en este caso se configuró en estado "Forward On-Hook Transmission" el cual es el estado normal de funcionamiento para la transmisión de señales por la línea telefónica.

3.1.2 Configuración del Codec/Decodec

El Si3210 soporta la Ley u-255 y la ley A para el compansor que en este caso se utiliza la ley A debido a que es la utilizada comúnmente para Sudamérica. El método de cuantificación es seleccionable mediante el registro 1 (PCM Mode Select) configurando los bits PCMF [1:0]. Ver figura 23.

4:3	PCMF[1:0]	**PCM Format.** 00 = A-Law 01 = µ-Law 10 = Reserved 11 = Linear

Fig. 23: Configuración del Cuantificador [9]

3.1.3 Diseño y Configuración del Conversor DC-DC

El Si3210, a diferencia de los comunes SLIC, no requiere el diseño de una fuente de alto voltaje externo ya que él mismo genera todos los niveles de voltaje necesario en base a su propio conversor DC-DC, el cual solo requiere de algunos componentes discretos y configuración de registros internos para el voltaje de salida.

En este caso se utiliza componentes discretos para el diseño del conversor DC – DC por un tema de bajo costo y también porque el diseño no es muy complejo. La hoja de datos recomienda un circuito basado en BJT resistencias, condensadores e inductores, con valores determinados de acuerdo a las características del nivel de voltaje que se desea generar a la salida del conversor. Ver figura 24.

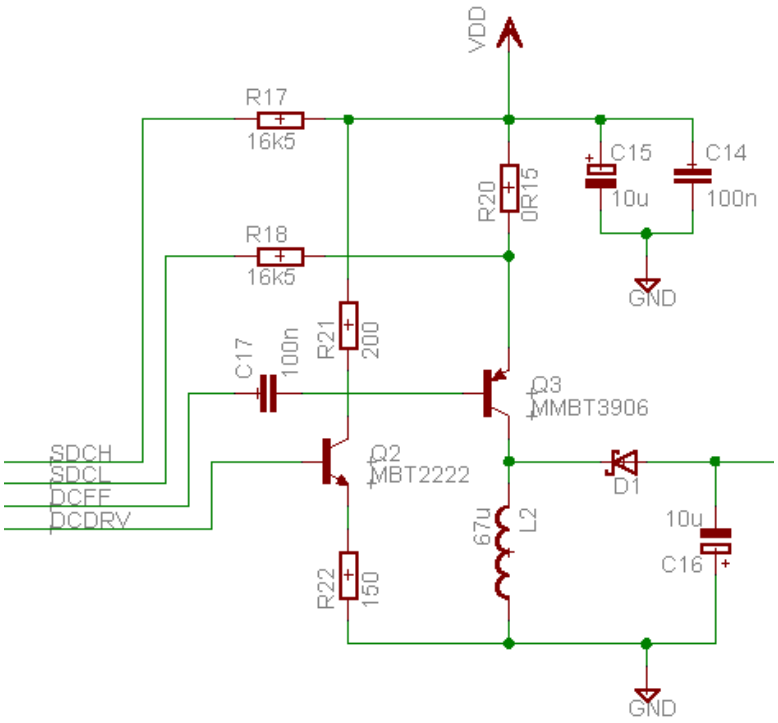


Fig. 24: Circuito del Conversor DC-DC, VDD = 5V

El conversor DC – DC trabaja básicamente como una fuente Step-Up Inverter, que en base a la selección del correcto valor de inductor y un transistor que conmute a cierta velocidad, eleva el nivel de voltaje e invierte la polaridad a un valor deseado de acuerdo a la configuración que realice el Si3210 al circuito mediante los pines SDCH, SDCL, DCFF y DCDRV. La salida del conversor DC-DC será llevada a los pines VBATH y VBAT del Si3201.

La fuente VDD = 5V hace pasar una corriente por la resistencia R20 la cual genera un nivel de voltaje en este para así poder sensar el voltaje entre los pines SDCH y SDCL ´para saber el nivel de corriente que está circulando. Luego el pin DCFF conmuta el transistor Q2 para dejar pasar la corriente hacia el inductor L1 y almacena cierta cantidad de energía en la

inductancia hasta que el pin DCFF abre el transistor y la energía almacenada en el inductor es entregada como una cantidad fija de potencia hacia la carga.

La ecuación 6 ayuda a conocer a que frecuencia se debe configurar el Si3210 para poder generar la potencia deseada.

$P = (L*I^2*fo)/2$ ……. (6)

Esta potencia es direccionada al VBATH el cual requiere, según hoja de datos, como máximo 45 V a 1 A y sabiendo que el valor de la inductancia del conversor es de 67 uH (observar figura 25), se hizo el cálculo respectivo despejando en la fórmula 6 "fo" por lo que se obtuvo una frecuencia de 1.373 MHz que corresponde a un periodo de 728.26 ns.

La frecuencia con la cual se configura la conmutación del transistor se realiza mediante la configuración del registro 92 (con la resolución correspondiente de acuerdo a la frecuencia deseada) y para activar el conversor DC-DC se coloca el bit 4 del registro 14 en "0".

La configuración para el funcionamiento de los pines del Si3210 es a base de la tabla 16.

Parameter	Range	Resolution	Register Bit	Location
DC-DC Converter Power-off Control	N/A	N/A	DCOF	Direct Register 14
DC-DC Converter Calibration Enable/Status	N/A	N/A	DCCAL	Direct Register 93
DC-DC Converter PWM Period	0 to 15.564 µs	61.035 ns	DCN[7:0]	Direct Register 92
DC-DC Converter Min. Off Time	(0 to 1.892 µs) + 4 ns	61.035 ns	DCTOF[4:0]	Direct Register 93
High Battery Voltage—V_{BATH}	0 to –94.5 V	1.5 V	VBATH[5:0]	Direct Register 74
Low Battery Voltage—V_{BATL}	0 to –94.5 V	1.5 V	VBATL[5:0]	Direct Register 75
V_{OV}	0 to –9 V or 0 to –13.5 V	1.5 V	VMIND[3:0] VOV	Indirect Register 41 Direct Register 66

Tabla 16: Registros de Configuración del Circuito Conversor DC-DC [9]

3.1.4 Protocolo SPI para Configuración del Si3210

La comunicación que se realiza entre el microcontrolador y el Si3210 para la configuración de este, se realiza mediante el protocolo de comunicación SPI (Serial Peripheral Interfase), el cual es una comunicación serial de manera síncrona y que puede llegar a alcanzar velocidades de comunicación de hasta el orden los Mega Hz y el cual ayuda a reducir el uso de pines del microcontrolador ya que solo usa 4 pines para la comunicación los cuales son SCLK, SDI, SDO y /CS del Si3210.

La configuración del protocolo SPI debe cumplir con las siguientes características para que la comunicación con el Si3210 sea óptima: [9]

- Frecuencia de reloj máximo 160 MHz.
- El primer bit que se envía del dato debe ser el MSB.
- Polaridad en alta de la señal de reloj en estado de reposo.
- La lectura de cada bit será en flanco de subida de la señal de reloj.

El formato de la comunicación entre el Si3210 y el TMS320F28032 es de la siguiente manera:

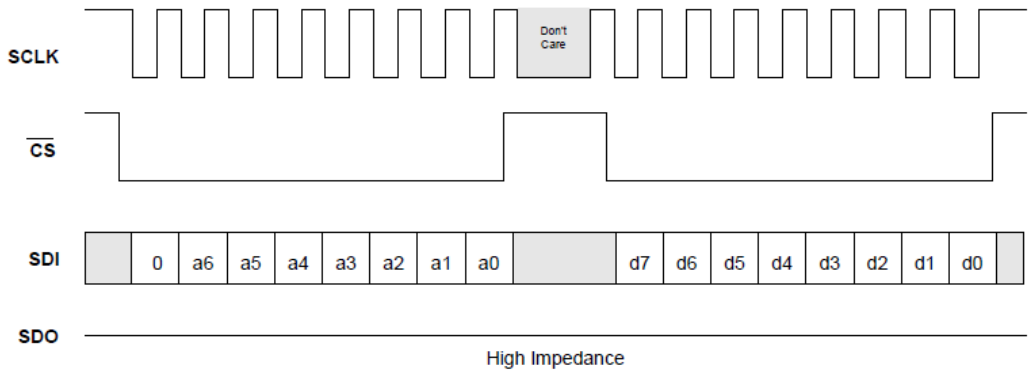


Fig. 25: Modo Escritura de 8 bits [9]

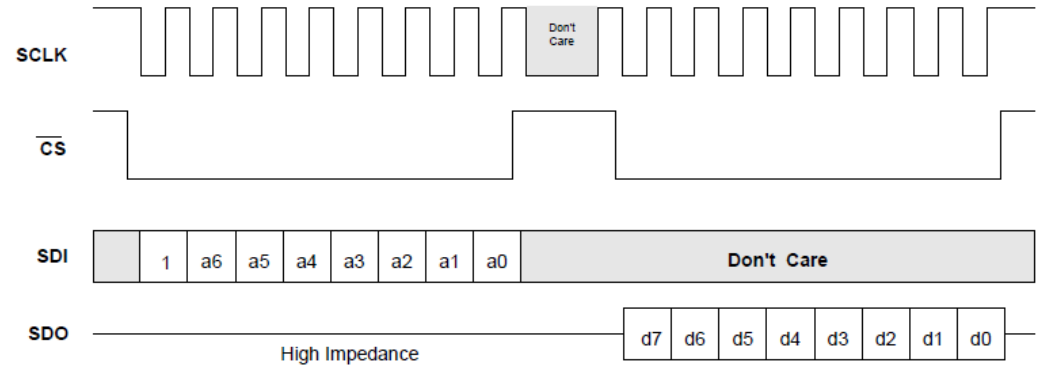


Fig. 26: Modo Lectura de 8 bits [9]

Como apreciamos en las figuras 25 y 26 el primer byte que se manda está formado por el bit a(7) que indica el modo (0 = escritura y 1 = lectura) y los 7 bits restantes indican la dirección del registro al cual se quiere acceder. Luego se envía o se recibe el dato correspondiente al registro direccionado.

3.1.5 Configuración de la Interface PCM

El Si3210 ofrece una interface programable de transmisión y recepción PCM el cual es controlado vía PCLK (PCM Bus Clock) y FSYNC (Frame Synchronization). Se puede configurar para que trabaje en cierto rango de frecuencias de 256 KHz y 8192 KHz. [9]

Esta señal PCM es la que contiene la señal de transmisión y recepción de la voz digitalizada, y tiene un funcionamiento similar al protocolo SPI ya que también es una señal serial síncrona, pero a diferencia del SPI tradicional, la interface PCM usa la señal FSYNC la cual genera un pulso alto el cual indica el inicio de la transmisión y/o recepción de un dato. Este pulso puede tener un largo o corto ancho de pulso de acuerdo a su configuración. Ver figura 27.

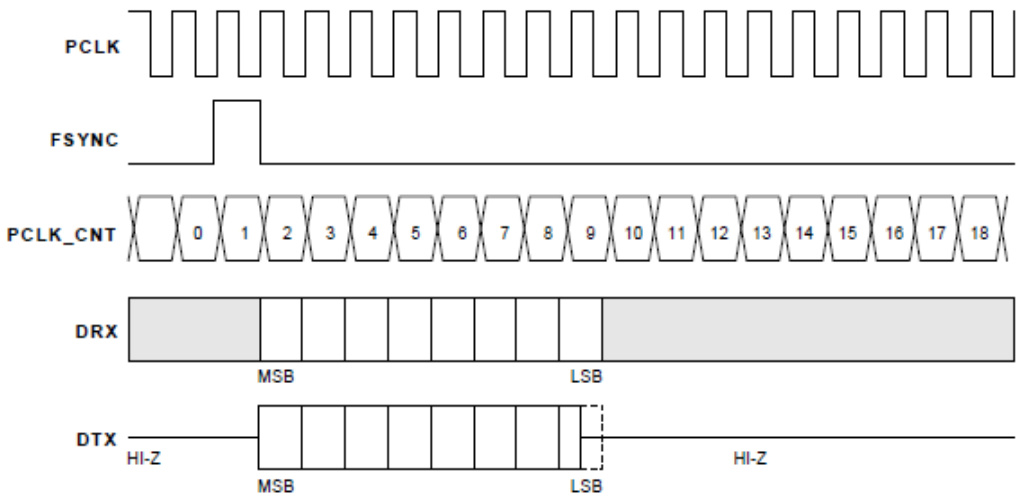


Fig.27: Ejemplo de Transmisión PCM [9]

3.2 Encriptación

Como se vio anteriormente para la comunicación entre el TMS32F28032 y el Si3210 existe la señal SPI para configuración y la señal PCM para la transmisión y/o recepción de datos, lo cual indica que necesitaríamos 2 protocolos de señal serial síncrona en el

microcontrolador, pero el TMS320F28032 solo tiene una salida SPI, lo cual lleva a que debemos realizar una señal serial síncrona “simulada” la cual se hace mediante firmware.

La señal simulada se realiza a nivel de bit a bit cada una de las señales de reloj y data, la cual debe cumplir con las velocidades correspondientes. Como se tiene 2 buses de señal serial síncrona, el que elegiremos para que sea serial simulado deberá ser el que no requiera tanta velocidad ya que esto lleva a que no nos preocupe el uso de memoria flash para realizar la señal simulada.

La señal simulada será la señal de configuración SPI, ya que esta solo se realiza cuando el equipo se active la cual se realiza solo una vez y este no requiere tanta velocidad ya que se trata de solo configuración del ProSLIC, mientras que la señal PCM si necesita una respuesta más rápida ya que esta es la señal que se procesará y se recibirán datos continuamente.

3.2.1 Armado y Envío de la Matriz de Estado

En este proceso de encriptación utilizaremos una matriz de estado de tamaño 4x8 es decir 256 bits la cual estará formada por la señal PCM de salida del SLIC/Codec. La señal PCM de entrada, mediante el protocolo SPI, irá almacenándose en SRAM hasta llegar a los 32 datos que requiere para formar la matriz de estado. Esta primera matriz de estado formada se hará pasar por el algoritmo de encriptación compuesto por las funciones previamente mencionada utilizando el proceso de la figura 12, el cual tiene un determinado tiempo de ejecución.

Para el caso en que llegaran muestras de la señal en pleno proceso de encriptación, estos datos se van almacenando en un buffer de memoria para que de esta forma estos datos no se pierdan y cuando se termina de procesar un bloque se pasa al siguiente sin pérdida de datos.

Luego de que la matriz de estado pasa el proceso de encriptación mediante el algoritmo, la matriz de estado será transmitida junto a 4 bytes de cabecera los cuales indicaran el inicio de la matriz de estado de N bytes según el tamaño que se ha decidido. Esta cabecera se ha añadido para ayudar, en caso de pérdida de algún byte al momento de la transmisión o recepción, a detectar el inicio de cada matriz de estado y cuando llega la siguiente matriz, y así revisar que esté completa, que en caso haya pérdida de algún byte, esta matriz se descarta por completo.

La figura 28 muestra el diagrama de flujo que realiza el microcontrolador para el proceso de cifrado.

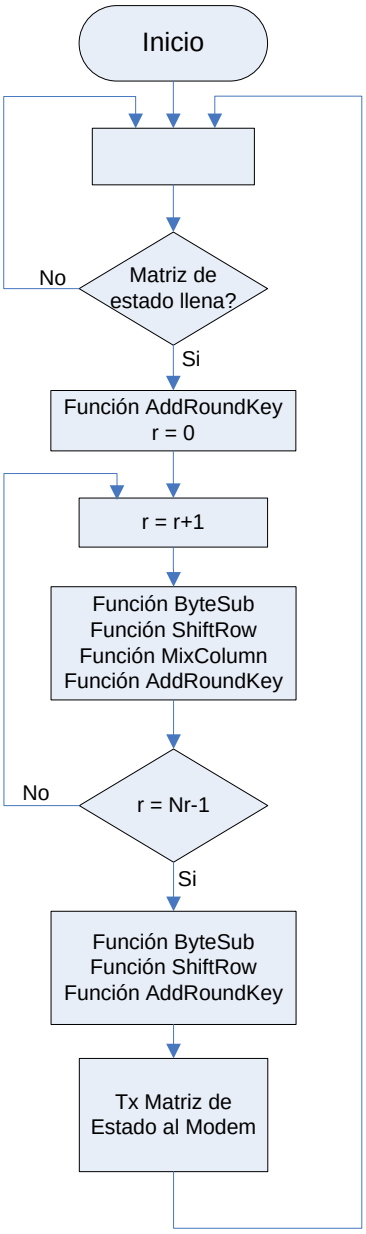


Fig. 28: Diagrama de Flujo del Proceso de cifrado

3.2.2 Generación de Subclaves

Durante todo el proceso de encriptación, algunas funciones en el algoritmo utilizan las subclaves, las cuales son valores generados a partir de la matriz que representa a la llave de encriptación.

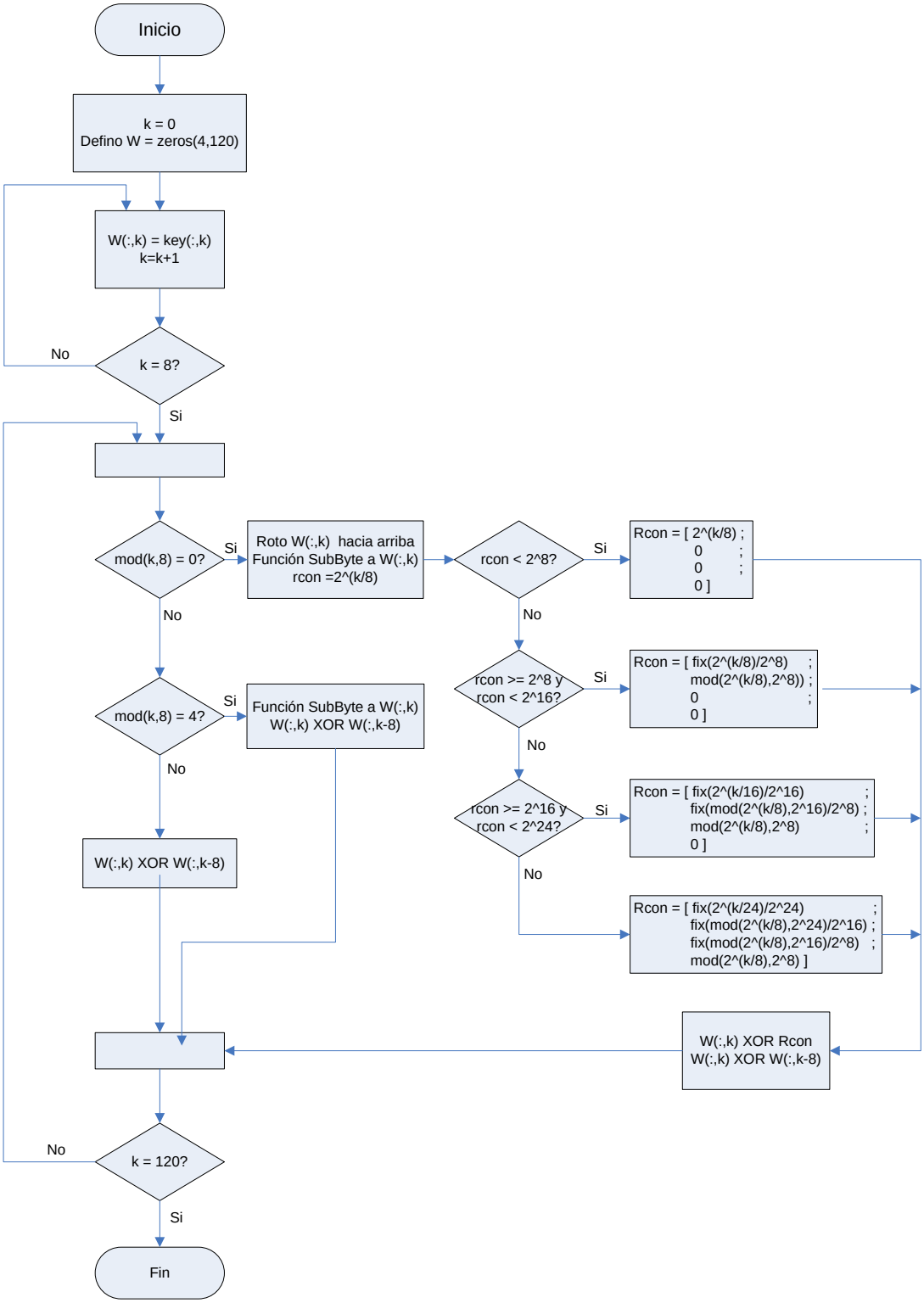


Fig. 29: Diagrama de Flujo del Algoritmo de Generación de Subclaves

La figura 29 muestra el diagrama de flujo del algoritmo que se utiliza para generar las subclaves que necesita el algoritmo de encriptación para realizar todo el proceso de cifrado, donde la matriz “key” representa la matriz de la llave de encriptación a utilizar y la matriz “W” indica todas las subclaves que se han generado la cual tiene un tamaño de 4 x 120 que sirve para el proceso del algoritmo principal.

3.3 Ingreso de la Llave de Encriptación

El ingreso de la llave de encriptación se realiza mediante un teclado seleccionando en el menú de la pantalla la activación del modo seguro para activar el proceso de cifrado, el cual automáticamente indica el ingreso de la llave de encriptación que será ingresado mediante el teclado.

Esta llave de encriptación está compuesta de 32 caracteres entre letras y números lo cual nos da una gran variedad de combinaciones, donde para este caso se requiere de un teclado especial que pueda proporcionar el ingreso de este tipo de caracteres por lo que usaremos un modelo de teclado como el de la figura 30.

Fig. 30: Teclado para el Control del Equipo de Cifrado

La llave ingresada se valida siempre y cuando contenga 32 caracteres como lo indicado, y esta se irá almacenando en la SRAM del microcontrolador para el inicio del cifrado y el enlace con el otro usuario.

3.4 Configuración del Modem

Para controlar y configurar el modem, se utiliza los comandos AT, los cuales son utilizados comúnmente para transmisión de datos telefónicos, GPRS, GSM, etc; que serán ingresados al modem mediante una comunicación serial asíncrona.

Esta comunicación serial asíncrona se define en el microcontrolador como USART (Universal Synchronous and Asynchronous Reciver/Transmitter) el cual debe contar con las siguientes características para enlazarse con el modem:

- Velocidad de transmisión 115000 Kbps
- 8 bits de datos
- 1 bits de parada
- Ningún bit de paridad

Los pines que definen la comunicación USART del microcontrolador son: SCIRXDA y SCITXDA y se conectan a los pines RXD y TXD respectivamente del modem para la transmisión de datos.

El comportamiento del modem se rige en base a los registros S que son los que le indican al modem como debe ser siempre su funcionamiento durante el proceso de comunicación.

Cada registro S tiene una característica especial que define al modem y las cuales tienen valores predeterminados y pueden ser modificados manualmente dependiendo del uso que se desee dar o la respuesta que deseamos que tenga el sistema.

La configuración de cada registro se realiza enviando el comando ATSn=<valor>, donde n es el número del registro y <valor> es el valor que se desea almacenar en el registro de acuerdo a la configuración que deseamos.

El modem tiene una configuración determinada de los registros S que salen de fábrica, los cuales están configurados con valores necesarios para que el modem funcione para una transmisión de datos normales.

Las características típicas que tendrá el funcionamiento del modem serán las siguientes:

- 2 segundos que espera el modem después de conectarse antes de marcar el primer digito del número telefónico.
- 50 segundos que espera el modem hasta esperar una señal portadora de respuesta.
- Velocidad máxima de línea de marcación.

Primero se comprueba de que el microcontrolador se puede comunicar con el modem, para esto el microcontrolador envía el comando AT, el cual es para verificación de comunicación, y el modem responde con un OK. Luego se verifica la velocidad de comunicación que tiene el modem para conectarse con el microcontrolador utilizando el comando AT+IPR?, el cual indica el Baud Rate del modem. En caso de que la velocidad no corresponda con la velocidad a la cual deseamos trabajar, esta se puede cambiar enviando el comando AT+IPR=115200 y para validar el cambio se vuelve a enviar el comando AT+IPR?.

```
AT<0D>
<0D>
<0A>OK<0D>
<0A>AT+IPR?<0D>
<0D>
<0A>+IPR: 0<0D>
<0A>OK<0D>
<0A>AT+IPR=115200<0D>
<0D>
<0A>OK<0D>
<0A>AT+IPR?<0D>
<0D>
<0A>+IPR: 115200<0D>
<0A>OK<0D>
<0A>
```

Fig. 31: Comandos de Verificación de Conexión entre el Microcontrolador y el Modem

En la figura 31 se observa que por defecto el Baud Rate del modem es 0 Kbps, es por eso que se debe de configurar a la velocidad de 115200 Kbps. Luego de configurado el Baud Rate, se pasa a configurar los registros S, solo de ser necesarios, pero para este caso mantendremos los valores por defecto.

Antes de que se pueda establecer una comunicación por la línea telefónica, se debe de realizar un enlace de verificación y handshaking entre los modem de ambos extremos del equipo de encriptación, las cuales deben ser:

- Enviar el comando ATD, el cual le dice al modem que haga una marcación por tonos, siempre y cuando este detecte el tono de llamada, y luego envía una señal portadora al otro modem.
- Si no recibe una respuesta durante un tiempo definido por el registro S, muestra el mensaje de la figura 32 y cancela el llamado.

```
ATD<0D>
<0D>
<0A>NO CARRIER<0D>
<0A>
```

Fig. 32: No Detección de Portadora del Modem Principal

- Mientras el modem de un extremo realiza dicha conexión, el modem del otro extremo debe de estar en modo escucha enviándole el comando ATA lo cual lo mantiene esperando la señal portadora que envía el otro modem y si no detecta nada durante un tiempo indicado por S7, muestra el mensaje de la figura 33 y cuelga la llamada.

```
ATA<0D>
<0D>
<0A>NO CARRIER<0D>
<0A>
```

Fig. 33: No Detección de Portadora del Modem en Estado Escucha

- Cuando el otro modem detecte la señal portadora, enviará igualmente una señal portadora en la línea como respuesta y el modem que empezó la conexión recibirá esta respuesta y sitúa la portadora de comienzo de línea. Ver figura 34.

```
ATD<0D>          ATA<0D>
<0D>             <0D>
<0A>OK<0D>       <0A>OK<0D>
<0A>             <0A>
```

Fig. 34: Respuesta de Detección de Portadora de Ambos Modem

- Luego de esto, ambos módems se ponen de acuerdo en la velocidad y la modulación con la cual van a trabajar. Y de ahí determinan la técnica de compresión y detección de errores a utilizar.

- Luego de haber realizado el handshaking y establecido el modo de operación el modem responderá con un “CONNECT” e indicando la velocidad que los modem establecieron para la transmisión.
- A partir del momento que el modem indica que la conexión está establecida, el modem pasa al modo de datos, lo cual pone al modem en un estado donde todo lo que reciba del host serán datos directos a la línea telefónica.

El sistema inicia el enlace cuando se seleccione mediante el teclado la opción de modo seguro, lo cual luego de verificar el tamaño de la llave de ingreso, se enlaza con el otro usuario esperando una respuesta satisfactorio que en el caso contrario simplemente cancelará el enlace, lo cual indica que el otro usuario aun no ha iniciado el enlace y hay que volver a intentar.

CAPÍTULO 4: PRUEBAS Y RESULTADOS

La principal prueba que se debe realizar es el funcionamiento del algoritmo de encriptación ya que este es el corazón del sistema, para lo cual se debe de verificar que el algoritmo este bien hecho ya que la variación o error de modificación de un solo byte de los datos de la matriz de estado puede alterar la señal resultante de todo el sistema y al momento que se desea regenerar la señal a su forma original se pueden tener datos errados, por lo que se debe realizar una previa simulación del algoritmo.

La simulación del algoritmo de encriptación se realizó en Matlab, ya que este software nos permite visualizar la señal de entrada tanto en tiempo como en frecuencia, y de la misma manera la señal cifrada y la señal regenerada.

Para la generación de subclaves, utilizando el proceso de la figura 29, definiremos una matriz para la "llave" con valores aleatorios que se muestra en la figura 35.

```
key = [     27    23   113   133    62   178   171   173
           156    68   135   242   173    17   216     2
           199    39   117   163    74    65    88   154
           108    72   224   245   172    57   200    99 ];
```

Fig. 35: Matriz de la Llave de Encriptación

Luego de generar las subclaves se toma una señal de voz grabada previamente llamada "hola" en formato .wav para que el Matlab pueda reconocerlo, y debido a que esta prueba

es simplemente para corroborar el algoritmo no es necesario pensar como ingresaremos luego una señal de voz al sistema ya que esta tendrá otro método de entrada.

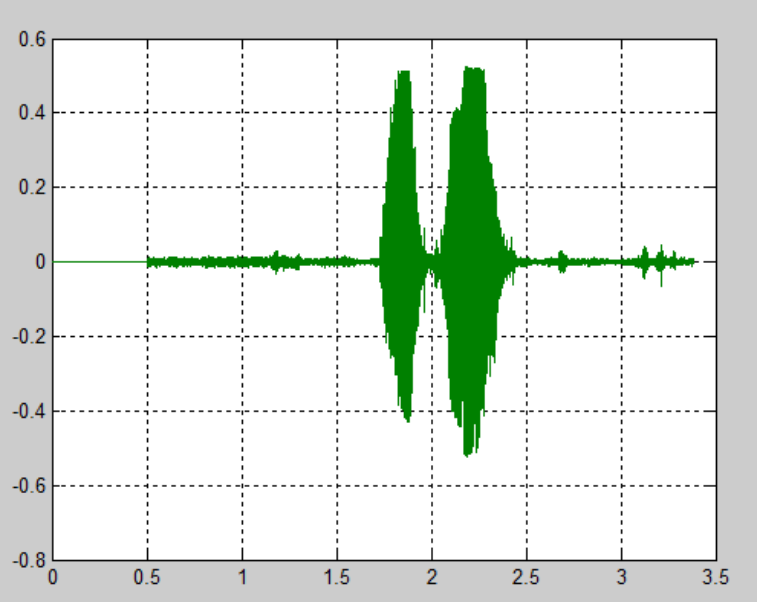


Fig. 36: Señal de Voz Original en el Tiempo

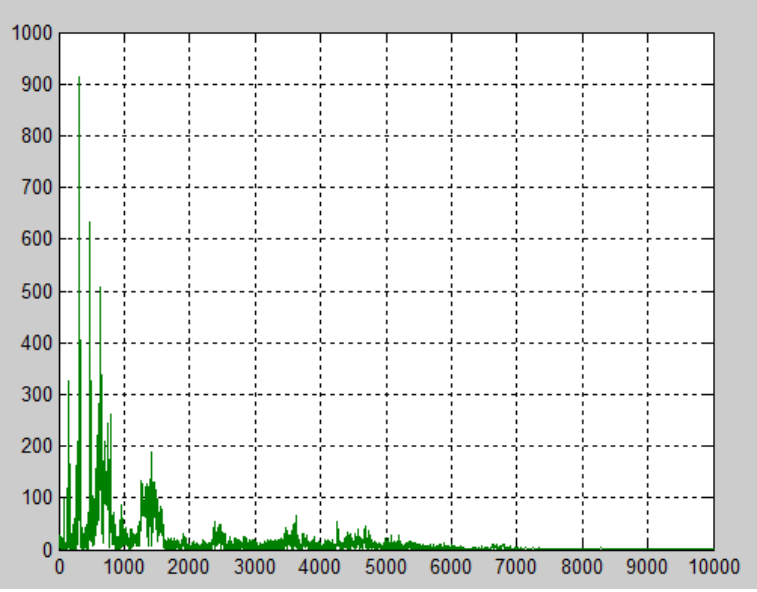


Fig. 37: Espectro de Frecuencia de la Señal de Voz Original

Como apreciamos en la figura 36, se tiene una señal de voz cualquiera de un tiempo de duración determinado, donde también en la figura 37 apreciamos el espectro de frecuencia de la señal que nos muestra que la señal se encuentra dentro del rango de frecuencia de un señal de voz.

Esta señal de voz se hizo pasar por el algoritmo de encriptación señalado en el Anexo 2. Los bytes de la señal de voz se almacenaron en la matriz C, la matriz de estado está representada por la matriz "a" y la señal cifrada se almacenó en la matriz "aa".

Al aplicar la señal de voz al algoritmo de encriptación, se obtuvo la siguiente señal cifrada:

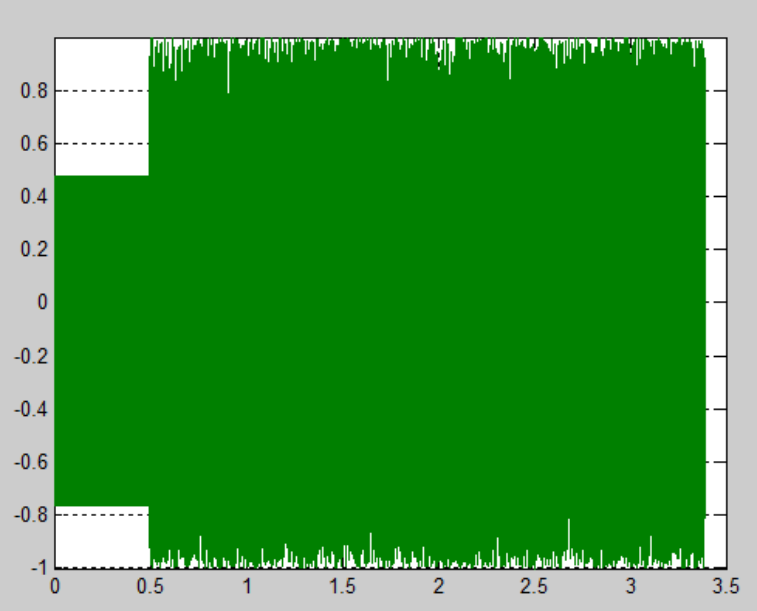


Fig. 38: Señal de Voz Encriptada en el Tiempo

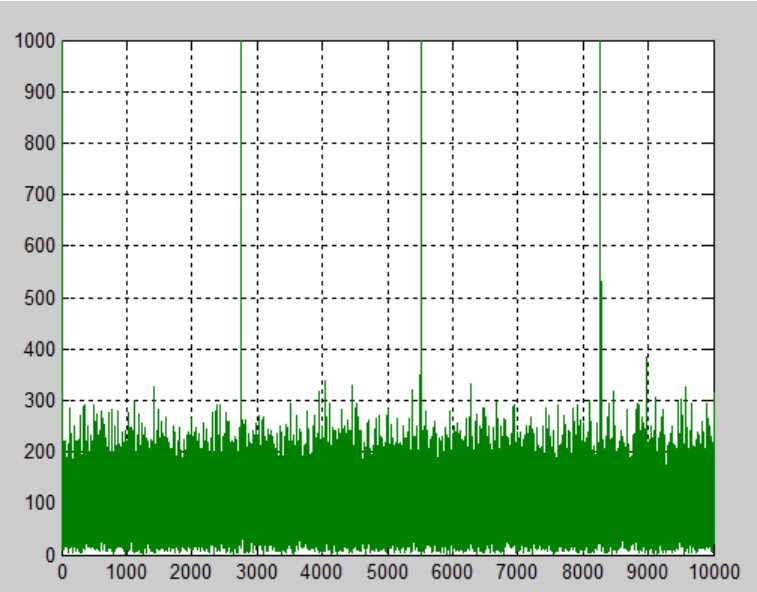


Fig. 39: Espectro de Frecuencia de la Señal de Voz Encriptada

Como se observa en la figura 38 gracias al algoritmo AES se ha transformó la señal de voz en una señal de varias frecuencias y amplitudes, y en la figura 39 su espectro de frecuencia, lo cual hace a la señal original que sea inentendible al oído de cualquier persona.

A pesar de que ya se generó una señal totalmente diferente a la señal original, se debe validar que realmente se efectuó el algoritmo como debe ser. Para esto se regeneró la señal cifrada a la señal original realizando el proceso inverso del algoritmo, la cual se realizó mediante el algoritmo indicado en el Anexo 3. Ver figura 40 que contiene la señal descifrada en el tiempo y la figura 41 que contiene su respectivo espectro de frecuencia.

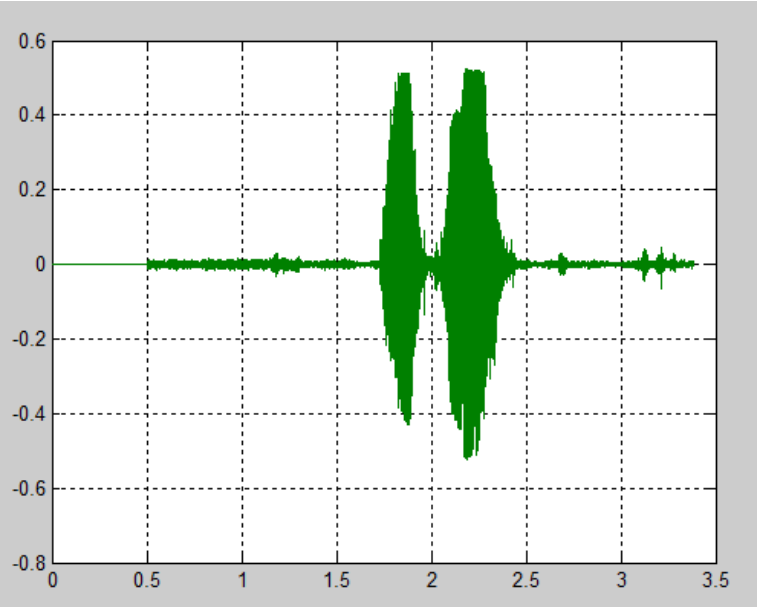


Fig. 40: Señal de Voz Desencriptada en el Tiempo

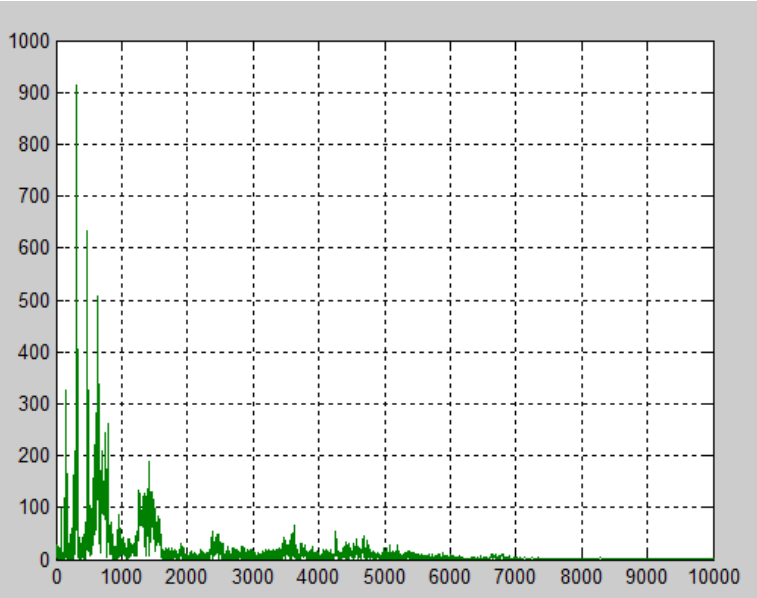


Fig. 41: Espectro de Frecuencia de la Señal de Voz Desencriptada

Para conocer el error que se genera a partir de la modificación de la señal y la pérdida o distorsión que tiene la señal de voz durante todo el proceso de encriptación se utilizó la

fórmula del error cuadrático medio, el cual sirve para conocer el error que existe entre dos señales una procedente de la otra.

$$\mathbf{EMC} = \sqrt{\frac{\sum_{\mathbf{i=0}}^{\mathbf{N-1}}((\mathbf{Señal\_Original(i)} - \mathbf{Señal\_Decifrada(i)})^{\mathbf{2}})}{\mathbf{N}}} \quad \text{..... (7)}$$

El resultado del valor de la EMC luego de aplicar la fórmula 7 es 2.6212e-006, lo cual indica que el error entre la señal original y la señal decifrada se encuentra en el orden de 10^{-6} que viene a ser micro voltios de diferencia entre ambas señales la cual no será irreconocible la diferencia entre ambas señales al oído humano.

Este error se generó debido al proceso de amplificación y codificación de la señal ya que la señal de voz se encuentra en el orden de los mili voltios y al momento de la amplificación aun se mantienen números con valores decimales, los cuales al momento de la codificación no son tomados en cuenta y es ahí donde se genera el error entre ambas señales de entrada y salida.

En la figura 42 se muestra las conexiones que se realizó para las pruebas del modem telefónico, donde cada sector de las conexiones son enumeradas de acuerdo al funcionamiento de cada parte:

- 1: Software “Bootloader” para la transmisión de los comandos AT y recepción de la respuesta del modem.
- 2: Modem MT5692SMI-L-92.R1.
- 3: Conversor RS232 a USB para la conexión con la PC.
- 4: MAX232, interfaz TTL a RS232.

- 5: Batería de 12V.
- 6: Regulador de Voltaje LM7805.

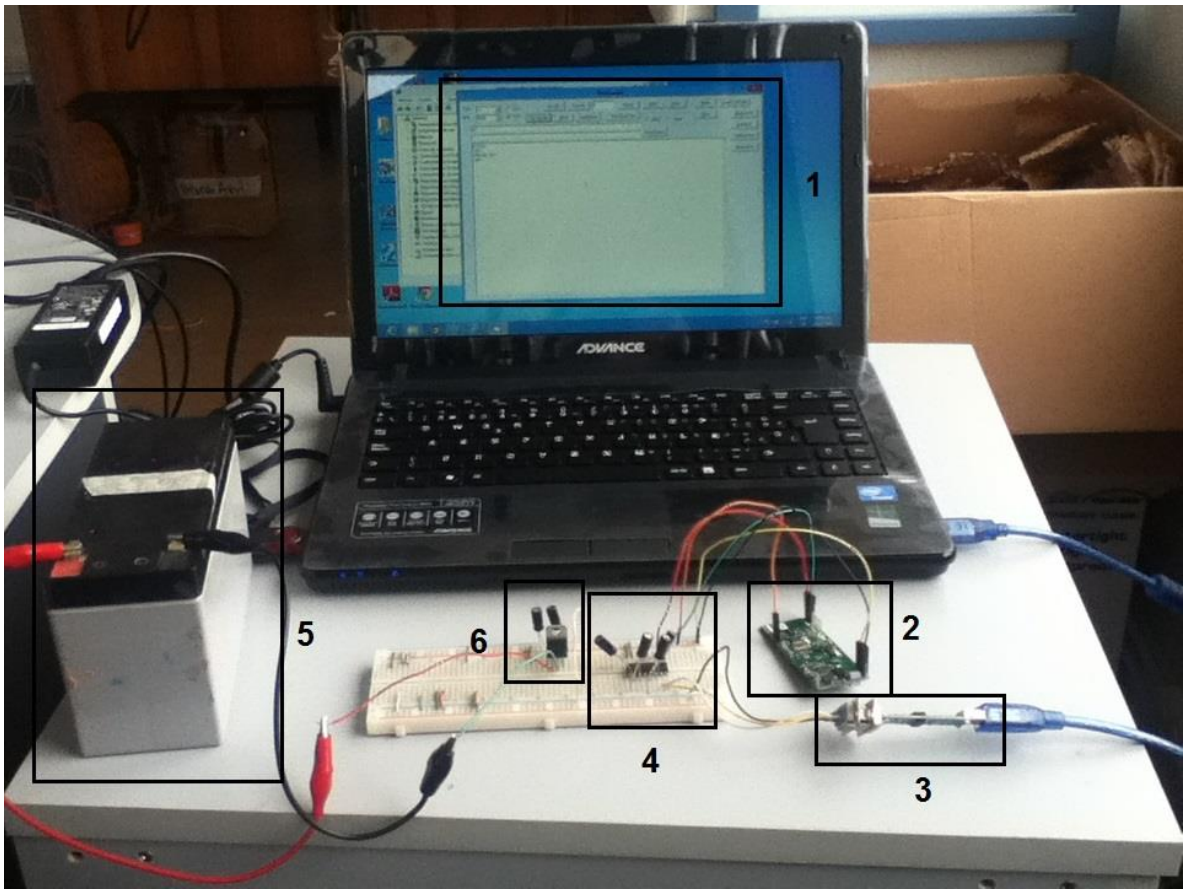


Fig. 42: Conexión de Pruebas del Modem

En la figura 43 se muestra las conexiones que se realizaron para las pruebas de corroboración del algoritmo de encriptación con una señal de voz proporcionada por un auricular telefónico. En la imagen se muestran las siguientes partes:

- 1: Auricular telefónico.
- 2: Etapa de amplificación de la señal.
- 3: Módulo de Evaluación: Piccolo F28027 con el algoritmo de cifrado programado.

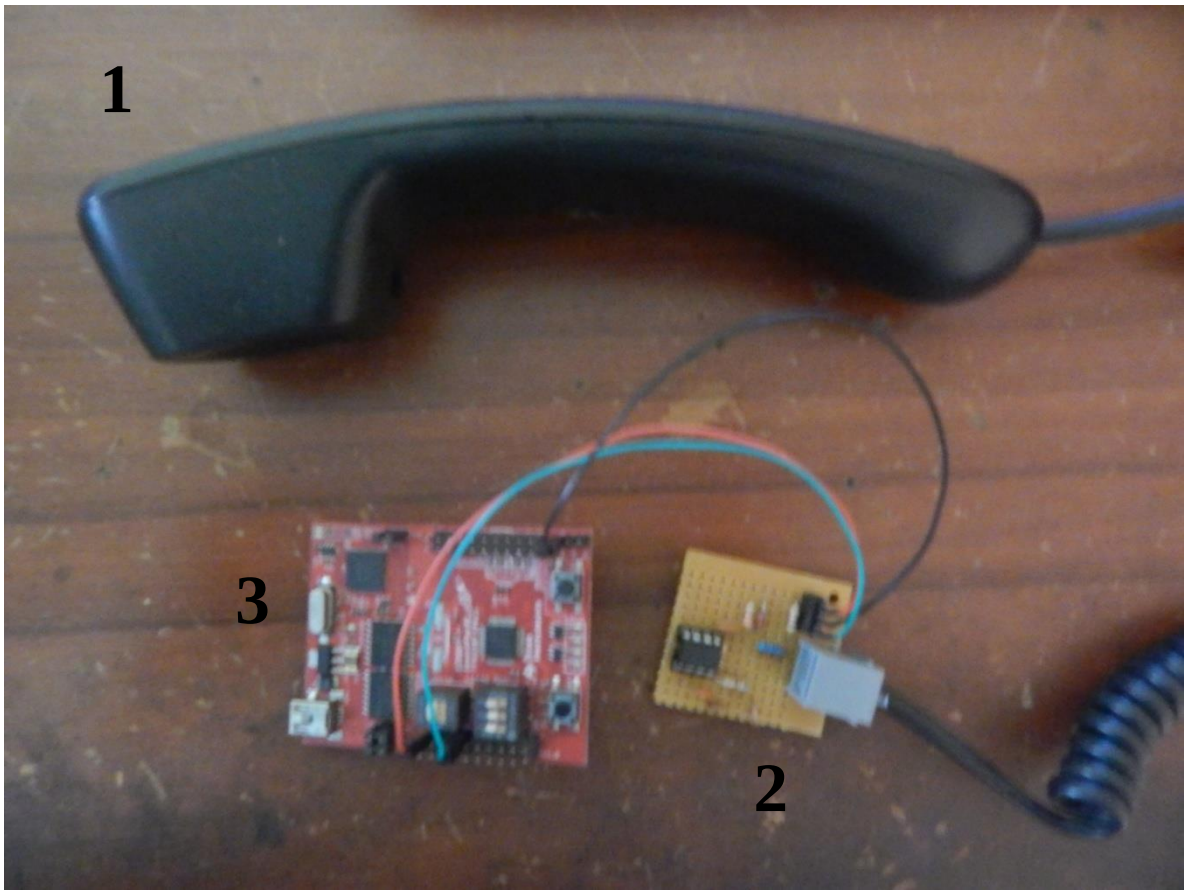


Fig. 43: Prueba del Algoritmo de Encriptación con Señal de Voz

A continuación se muestra un presupuesto detallado para la fabricación de un prototipo con el costo de los componentes puestos localmente. Ver tabla 17.

ITEM	CANTIDAD	MATERIALES/EQUIPOS	CODIGO	COSTO (S/.)
1	1	SLIC	SI3210	51.15
2	1	MICROCONTROLADOR	TMS320F28032	47.04
3	1	MODEM	MT5692SMI-L-92.R1	197.77
4	2	REGULADOR	MCP1801T-3302I	5.56
5	1	REGULADOR	LP2992AIM5	6.84
6	1	REGULADOR	LM2575D2T	13.12
7	1	CARGADOR DE BATERIA	MCP73844	8.67
8	1	MOSFET	BSS84	1.22
9	1	CRYSTAL		1.00
10	3	DIODO RECTIFICADOR	1N4004	0.60
11	1	DIODO SCHOTKY	1N5819	0.50
12	2	DIODO LIMITADOR	1N4148	0.40
13	2	CONNCTOR JACK		2.00
14	2	MOLEX		2.00
15	2	TRANSISTOR NPN	MBT2222	1.56
16	1	TRANSISTOR PNP	MMBT3906	0.67
17	2	CONECTOR	RJ-11	1.00
18	2	RELE		2.00
19	2	OPTOCOPLADOR	4N33	1.40
20	1	PANTALLA LCD		20.00
21	1	TECLADO	4X4	15.00
22	2	VARISTOR		2.00
23	1	LINEFEED	SI3201	11.18
24	1	BATERIA (2 CELDAS)	FLIGHTMAX	45.20
25	1	PCB		40.00
26		RESITENCIAS		1.00
27		CONDENSADORES		1.00
28		BOBINA		1.00
29	1	CASE		70.00
			TOTAL	550.88

Tabla 17: Presupuesto de un Prototipo

CONCLUSIONES

1. Se efectuó un procesamiento de encriptación de la voz de manera fluida y sin retardos debido a la velocidad de codificación mediante el chip SLIC/Codec y una programación eficiente en el tiempo en que el microcontrolador encripta la señal y la velocidad con la que se transmite la señal a la línea telefónica.
2. Se utilizó un algoritmo de encriptación muy eficiente y que hasta el momento no ha podido ser roto y en el equipo es difícil afectar los datos que se envían debido a que este no tiene ningún otro medio de acceder a la señal que no sea la misma salida que ya tiene la señal encriptada.
3. Se diseñó un sistema totalmente portátil que no requiere conexión a una fuente externa, ya que consta de una batería y tiene un sistema en hardware altamente diseñado con componentes en bajo costo y de buena respuesta para un buen desempeño y fácil manejo para el usuario.

RECOMENDACIONES

Las mejoras que se recomiendan tener en el diseño del sistema criptográfico de voz son los siguientes:

1. Cambio en el lenguaje de programación del microcontrolador de lenguaje C a lenguaje assembler para obtener una mayor velocidad de procesamiento.
2. Ingreso de la llave de encriptación mediante conexión USB en base a un dispositivo que genere una cierta cantidad de llaves que son previamente grabados y van siendo seleccionadas de manera aleatoria en ambos extremos de la transmisión.

3. Incluir en el sistema la conexión de un auricular telefónico y el reconocimiento de tonos de señalización y de marcación, para que todo el sistema funcione también como un teléfono fijo normal que incluye la opción de transmisión criptográfica y sea un equipo completo y único.

BIBLIOGRAFÍA Y REFERENCIAS

http://encriptadores.com/productos/encriptacall, Mayo de 2012

http://www.tccsecure.com/products/voice-fax-data-encryption/CSD3324sp-detail.aspx,Estados Unidos, Abril de 2013

http://www.cryptophone.de/upload/files/6/original/CPPSTN1.pdf, Abril de 2013

Joskowicz, J. (2013). Conceptos Básicos de Telefonía, Uruguay: Instituto de Ingeniería Eléctrica

http://www.itu.int/rec/T-REC-G.711-198811-I/en, Marzo de 2013

Miró, V. (2007). Cuantificación, Comunicaciones Eléctricas

Wayne, T. (2003). Sistemas de comunicaciones electrónicas (4ta ed), Mexico: Pearson Educación

http://www.kriptopolis.org/docs/rijndael.pdf, Junio de 2012

Silicon Labs Si3210/Si3201, Febrero de 2013

http://www.itu.int/net/itu_search/index.aspx?cx=001276825495132238663%3Anqzm45z846q&cof=FORID%3A9&ie=UTF-8&q=modem, Junio de 2013

http://www.multitech.com/en_us/documents/collateral/data_sheets/86002113.pdf, Diciembre de 2012

http://www.encriptadores.com/fullaccess/archivo_2.pdf, Mayo de 2012

http://computacion.cs.cinvestav.mx/~jjangel/aes/AES_v2005_jjaa.pdf, Marzo de 2012

MATLAB/Help/Getting Started /Introduction/Product Overview, Junio de 2012

http://www.ti.com/lit/ds/symlink/tms320c25.pdf, Enero de 2012

GLOSARIO

Asíncrona: Señal de datos que no requiere de la sincronización de una señal de reloj.

Baud Rate: Taza de baudios que tiene toda señal el cual indica la velocidad a la cual se envían los bits de un dato.

Bypass: Sistema o dispositivo que tiene como función el de actuar como un puente entre dos sistemas que se comunican entre sí.

Compansor: Bloque que realiza el proceso de expansión y compansión digital, que sirve para mejorar la calidad de la señal durante el proceso de codificación.

Conversor DC-DC: Realiza la conversión de un voltaje continuo a otra voltaje continuo de diferente valor ya sea mayor o menor.

Desencriptamiento: Proceso de regeneración de una señal cifrada.

Encriptación: Proceso de codificación basado en un algoritmo y una clave, quienes se encargan de modificar una señal.

Espectro de Frecuencia: Descomposición espectral de las frecuencias de una señal analógica donde se visualiza las diferentes frecuencias que esta contiene y sus respectivas amplitudes.

Firmware: Programa principal que realiza todo el proceso de control el cual se almacena en un microcontrolador.

Handshaking: Proceso donde 2 sistemas se ponen de acuerdo sobre los parámetros de comunicación sobre el canal que utilizarán para enlazarse.

Matriz de Estado: Bloque principal el cual es modificado por el algoritmo de cifrado donde vienen a estar los bytes que se desean cifrar.

Microcontrolador: Dispositivo electrónico que viene a ser el corazón de todo el sistema y el que se encarga de manejar el funcionamiento de todos los protocolos y periféricos del sistema.

PCM: Pulse Code Modulation (Modulación por Codificación de Pulsos)

Pseudoaleatorio: Proceso que realiza una supuesta generación aleatoria ya sea de números o bits, pero que en realidad se basa en un procedimiento determinado en base a un método específico.

Subclave: Claves generadas en base a la clave principal que son utilizadas durante todo el proceso de encriptación, donde la cantidad de subclaves es igual a la cantidad de rondas del proceso de cifrado.

Tip & Ring: **Nombres de las 2 tomas del cable que se utiliza en telefonía para la transmisión de la señal de voz, donde TIP tiene un nivel de tensión de 0V y RING tiene un nivel de tensión de -48V.**

ANEXO 1

Salida del Modulador 16-QAM

Entrada				Salida 16-QAM	
Q	Q'	I	I'	Amplitud	Fase
0	0	0	0	0.311 V	-135°
0	0	0	1	0.850 V	-165°
0	0	1	0	0.311 V	-45°
0	0	1	1	0.850 V	-15°
0	1	0	0	0.850 V	-105°
0	1	0	1	1.161 V	-135°

0	1	1	0	0.850 V	-75°
0	1	1	1	1.161 V	-45°
1	0	0	0	0.311 V	135°
1	0	0	1	0.850 V	165°
1	0	1	0	0.311 V	45°
1	0	1	1	0.850 V	15°
1	1	0	0	0.850 V	105°
1	1	0	1	1.161 V	135°
1	1	1	0	0.850 V	75°
1	1	1	1	1.161 V	45°

ANEXO 2

Algoritmo de Encriptación

```
% ENCRIPTACIÓN
    aa = zeros(N,1);
    f = 1;
    for rr = 1:N/32
        a = zeros(4,Nb);
        a1 = zeros(4,Nb);
        for j = 1:Nb
            for i = 1:4
                a(i,j) = C(f,1);
                f = f+1;
            end
        end
        %%%%%%%%% RONDA INICIAL %%%%%%%%%%%%%%%
        % Funcion AddRoundKey
        a(:,1) = bitxor(a(:,1),W(:,1));
        a(:,2) = bitxor(a(:,2),W(:,2));
        a(:,3) = bitxor(a(:,3),W(:,3));
        a(:,4) = bitxor(a(:,4),W(:,4));
        a(:,5) = bitxor(a(:,5),W(:,5));
        a(:,6) = bitxor(a(:,6),W(:,6));
        a(:,7) = bitxor(a(:,7),W(:,7));
        a(:,8) = bitxor(a(:,8),W(:,8));

        %%%%%%%% RONDA INTERMEDIA %%%%%%%%%%%
        for r = 1:Nr-1
            % Funcion SubByte
            for j = 1:Nb
                for i = 1:4
                    sboxf = fix(a(i,j)/16);
                    sboxc = mod(a(i,j),16);
                    a(i,j) = S_box(sboxf+1,sboxc+1);
                end
            end

            % Funcion ShiftRow
            a(2,:) = circshift(a(2,:),[0,-1]);
            a(3,:) = circshift(a(3,:),[0,-3]);
            a(4,:) = circshift(a(4,:),[0,-4]);

            % Funcion MixColumns
            for k = 1:8
                a1(1,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),2),prodaes(a(2,k),3)),a(3,k)),a(4,k));
                a1(2,k) = bitxor(bitxor(bitxor(a(1,k),prodaes(a(2,k),2)),prodaes(a(3,k),3)),a(4,k));
                a1(3,k) = bitxor(bitxor(bitxor(a(1,k),a(2,k)),prodaes(a(3,k),2)),prodaes(a(4,k),3));
                a1(4,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),3),a(2,k)),a(3,k)),prodaes(a(4,k),2));
            end
            a = a1;

            % Funcion AddRoundKey
            a(:,1) = bitxor(a(:,1),W(:,r*8+1));
            a(:,2) = bitxor(a(:,2),W(:,r*8+2));
            a(:,3) = bitxor(a(:,3),W(:,r*8+3));
            a(:,4) = bitxor(a(:,4),W(:,r*8+4));
            a(:,5) = bitxor(a(:,5),W(:,r*8+5));
            a(:,6) = bitxor(a(:,6),W(:,r*8+6));
            a(:,7) = bitxor(a(:,7),W(:,r*8+7));
            a(:,8) = bitxor(a(:,8),W(:,r*8+8));
        end

        %%%%%%%%%%% RONDA FINAL %%%%%%%%%%%%%%%%%%
        % Funcion SubByte
        for j = 1:Nb
            for i = 1:4
```

```
            sboxf = fix(a(i,j)/16);
            sboxc = mod(a(i,j),16);
            a1(i,j) = S_box(sboxf+1,sboxc+1);
        end
    end
    a=a1;

    % Funcion ShiftRow
    a(2,:) = circshift(a(2,:),[0,-1]);
    a(3,:) = circshift(a(3,:),[0,-3]);
    a(4,:) = circshift(a(4,:),[0,-4]);

    % Funcion AddRoundKey
    a(:,1) = bitxor(a(:,1),W(:,Nr*8+1));
    a(:,2) = bitxor(a(:,2),W(:,Nr*8+2));
    a(:,3) = bitxor(a(:,3),W(:,Nr*8+3));
    a(:,4) = bitxor(a(:,4),W(:,Nr*8+4));
    a(:,5) = bitxor(a(:,5),W(:,Nr*8+5));
    a(:,6) = bitxor(a(:,6),W(:,Nr*8+6));
    a(:,7) = bitxor(a(:,7),W(:,Nr*8+7));
    a(:,8) = bitxor(a(:,8),W(:,Nr*8+8));

    k = 1;
    for j = 1:Nb
        for i = 1:4
            aa((rr-1)*32+k,1) = a(i,j);
            k = k + 1;
        end
    end
end
```

ANEXO 3

Algoritmo de Desencriptación

```
% DESENCRIPTACIÓN
    load Señalc

    c = zeros(N,1);
    f = 1;
    for rr = 1:N/32
        a = zeros(4,Nb);
        a1 = zeros(4,Nb);
        k = 1;
        for j = 1:Nb
            for i = 1:4
                a(i,j) = aa((rr-1)*32+k,1);
                k = k + 1;
            end
        end
        %%%%%%%%%%%% VUELTA FINAL %%%%%%%%%%%%%
        % Función InvAddRoundKey
        a(:,1) = bitxor(a(:,1),W(:,Nr*8+1));
        a(:,2) = bitxor(a(:,2),W(:,Nr*8+2));
        a(:,3) = bitxor(a(:,3),W(:,Nr*8+3));
        a(:,4) = bitxor(a(:,4),W(:,Nr*8+4));
        a(:,5) = bitxor(a(:,5),W(:,Nr*8+5));
        a(:,6) = bitxor(a(:,6),W(:,Nr*8+6));

        a(:,7) = bitxor(a(:,7),W(:,Nr*8+7));
        a(:,8) = bitxor(a(:,8),W(:,Nr*8+8));

        % Función InvShiftRow
        a(2,:) = circshift(a(2,:),[0,1]);
        a(3,:) = circshift(a(3,:),[0,3]);
        a(4,:) = circshift(a(4,:),[0,4]);

        % Función InvByteSub
        for j = 1:Nb
            for i = 1:4
                sboxf = fix(a(i,j)/16);
                sboxc = mod(a(i,j),16);
                a(i,j) = S_boxinv(sboxf+1,sboxc+1);
            end
        end
        %a = a1;

        %%%%%%%%%% VUELTA INTERMEDIA %%%%%%%%%
        for r = Nr-1:-1:1
            % Funcion AddRoundKey
            a(:,1) = bitxor(a(:,1),W(:,r*8+1));
            a(:,2) = bitxor(a(:,2),W(:,r*8+2));
            a(:,3) = bitxor(a(:,3),W(:,r*8+3));
            a(:,4) = bitxor(a(:,4),W(:,r*8+4));
            a(:,5) = bitxor(a(:,5),W(:,r*8+5));
            a(:,6) = bitxor(a(:,6),W(:,r*8+6));
            a(:,7) = bitxor(a(:,7),W(:,r*8+7));
            a(:,8) = bitxor(a(:,8),W(:,r*8+8));

            % Funcion MixColumns
            for k = 1:8
                a1(1,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),14),prodaes(a(2,k),11)),prodaes(a(3,k),13)),prodaes(a(4,k),9));
                a1(2,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),9),prodaes(a(2,k),14)),prodaes(a(3,k),11)),prodaes(a(4,k),13));
                a1(3,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),13),prodaes(a(2,k),9)),prodaes(a(3,k),14)),prodaes(a(4,k),11));
                a1(4,k) = bitxor(bitxor(bitxor(prodaes(a(1,k),11),prodaes(a(2,k),13)),prodaes(a(3,k),9)),prodaes(a(4,k),14));
            end
            a = a1;

            % Funcion ShiftRow
```

```
        a(2,:) = circshift(a(2,:),[0,1]);
        a(3,:) = circshift(a(3,:),[0,3]);
        a(4,:) = circshift(a(4,:),[0,4]);

         % Funcion SubByte
        for j = 1:Nb
            for i = 1:4
                sboxf = fix(a(i,j)/16);
                sboxc = mod(a(i,j),16);
                a(i,j) = S_boxinv(sboxf+1,sboxc+1);
            end
        end
        %a=a1;
    end
    %%%%%%%%%%%%% SALIDA %%%%%%%%%%%%%%%%%
    % Funcion AddRoundKey
    a(:,1) = bitxor(a(:,1),W(:,1));
    a(:,2) = bitxor(a(:,2),W(:,2));
    a(:,3) = bitxor(a(:,3),W(:,3));
    a(:,4) = bitxor(a(:,4),W(:,4));
    a(:,5) = bitxor(a(:,5),W(:,5));
    a(:,6) = bitxor(a(:,6),W(:,6));
    a(:,7) = bitxor(a(:,7),W(:,7));
    a(:,8) = bitxor(a(:,8),W(:,8));
    %bloque_descifrado = a;

    for j = 1:Nb
        for i = 1:4
            c(f,1) = a(i,j);
            f = f+1;
        end
    end
end
```