INSTITUTO TECNOLOGICO DE TUXTLA GUTIÉRREZ

RESIDENCIA PROFESIONAL

IMPLEMENTACIÓN DE ALGORITMOS DE CRIPTOGRAFÍA BASADO EN MICROCONTROLADOR.

ALUMNA: ANAYELI GALLEGOS GUILLÉN.

No. CONTROL: 04270267

ASESOR: DR. HÉCTOR RICARDO HERNANDEZ DE LEÓN.

REVISOR: ING. RAÚL MORENO RINCON.

TUXTLA GUTIÉRREZ, CHIAPAS. DICIEMBRE DE 2008.

ÍNDICE

1. INTRODUCCIÓN

En este proyecto se presenta un sistema digital que demuestra la eficiencia del esquema de encripción y decripción en un sistema criptográfico, cuyo objetivo principal es proporcionar seguridad en el acceso y transferencia electrónica de datos, la cual, se logra mediante la proporción de una llave de seguridad que permite el acceso exclusivo a la información a quien posea tal llave.

La investigación se presenta a partir de la realización de un análisis completo de los fundamentos matemáticos y su comprobación a través de la evaluación propuesta en un sistema real.

Cabe mencionar que este trabajo se desarrolló en el Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez, bajo la supervisión del Dr. Héctor Ricardo Hernández De León.

1.1. PLANTEAMIENTO DEL PROBLEMA.

Hoy por hoy el tráfico e intercambio de la información es fundamental, así como lo es el crear sistemas de seguridad para darle más certeza al usuario en la confidencialidad de su información, existen diversos sistemas que procuran satisfacer dicha necesidad de seguridad tales como los que se encargan de encriptar, sin embargo; los existentes tienen códigos de fácil decoficación, es decir; son sistemas falibles. Otros consisten en la división de las claves para acceder a la información requerida la cual se logra mediante la combinación de

tres de las cuatro claves de tal manera que de forma individual no se podrá violar el código de acceso.

Ahora bien, para tener el sistemas se tuvieron que superar varias etapas, primero se tuvo que diseñar y construir el hardware para la encripción y decripción; utilizando un microcontrolador PIC18F4550. Enseguida, si hizo la programación del software para el PIC utilizando el programa mikroC y se diseño las interfaces graficas con ayuda de VISUAL BASIC, para posteriormente probar el funcionamiento del sistema.

1.2 OBJETIVOS

Objetivo general:

Diseñar e implementar un circuito digital basado en microcontrolador que permita la encripción y decripción de archivos.

Objetivos específicos:

1. Conocer los diferentes algoritmos criptográficos actuales y sus implementaciones en Hardware.

2. Seleccionar el algoritmo de criptografía que será implementado en el circuito digital, basado en microcontroladores.

3. Probar el algoritmo criptográfico del circuito digital implementado en una computadora personal.

1.3 JUSTIFICACIÓN

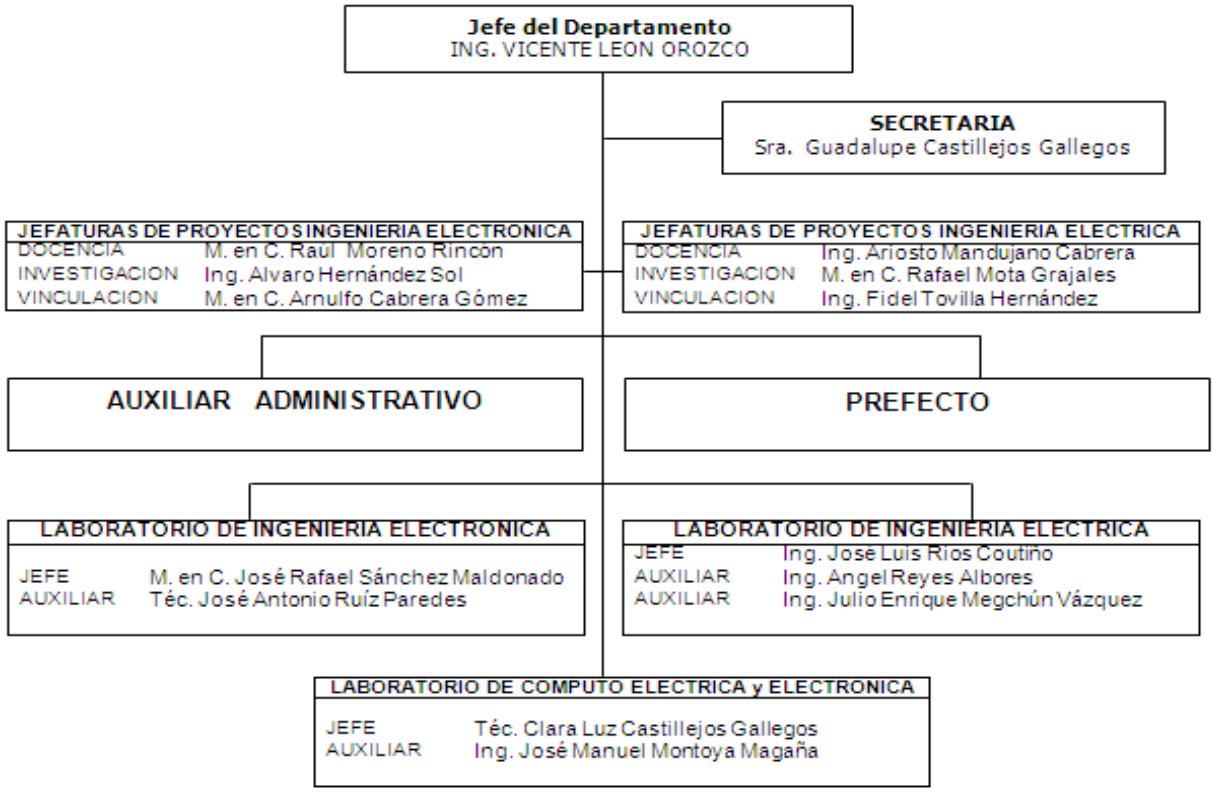
Ante el creciente problema de crear nuevas tecnologías que se enfoquen en la seguridad rigurosa de la información, se hace cada vez más preponderante crear sistemas que respondan a dicha necesidad, es por ello, que se diseña este proyecto el cual busca maximizar los niveles de seguridad en el intercambio de información.

1.4 ALCANCES Y DELIMITACIONES

La base de desarrollo de la implementación de la residencia profesional inicial consistía en la utilización de circuitos reconfigurables, pero en el diseño que se presenta se realizaron las pruebas con circuitos microcontroladores que proporcionan un buen desempeño, por lo que el sistema de encripción se realización usando este tipo de circuitos.

2. DATOS DE LA EMPRESA

ORGANIGRAMA DEPARTAMENTO DE INGENIERIA ELÉCTRICA Y ELECTRONICA



3. FUNDAMENTO TEÓRICO

3.1 COMUNICACIÓN SERIAL

El puerto serial de las computadoras es conocido como puerto RS-232, este permite las comunicaciones entre otros dispositivos tales como otra computadora, el mouse, impresora y para nuestro caso con los microcontroladores.

Existen dos formas de intercambiar información binaria: la paralela y la serial. La comunicación paralela transmite todos los bits de un dato de manera simultánea, por lo tanto la velocidad de transferencia es rápida, sin embargo tiene la desventaja de utilizar una gran cantidad de líneas, por lo tanto se vuelve mas costoso y tiene las desventaja de atenuarse a grandes distancias, por la capacitancia entre conductores así como sus parámetros distribuidos.

3.1.1 Tipos de Comunicaciones Seriales:

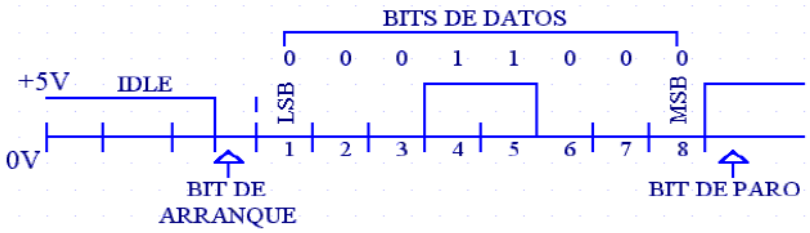
Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona. En la comunicación serial sincronía además de una línea sobre la cual se transmitirán los datos se necesita de una línea la cual contendrá los pulsos de reloj que indicaran cuando un dato es valido.

Ejemplos de este tipo de comunicación son:

- **I2C**
- **ONE WIRE**
- **SPI**

En la comunicación serial asíncrona, no son necesarios los pulsos de reloj. La duración de cada bit esta determinada por la velocidad con la cual se realiza la transferencia de datos.

La siguiente figura muestra la estructura de un carácter que se trasmite en forma serial asíncrona.



"Transmisión de un Carácter"

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en esto de (idle) este quiere decir en estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante determinado tiempo, lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir con un intervalo de tiempo los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT mas significativo.

Si el receptor no esta sincronizado con el transmisor, este desconoce cuando se van a recibir los datos. Por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de BIT de parada.

En los circuitos digitales, cuyas distancias son relativamente cortas, se pueden manejar transmisiones en niveles lógicos TTL (0-5V), pero cuando las distancias aumentan, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y su resistencia eléctrica. El efecto se incrementa a medida que se incrementa la velocidad de la transmisión. Todo esto origina que los datos recibidos nos sean igual a los datos transmitidos, por lo que no se puede permitir la transferencia de datos.

Una de las soluciones mas lógica es aumentar los márgenes de voltaje con que se transmiten los datos, de tal manera que las perturbaciones a causa de la línea se pueda corregir.

3.1.2 La Norma RS-232

Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre si. La **EIA (Electronics Industry Association)** elaboro la norma RS-232, la cual define la interfase mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:

- Un “1” lógico es un voltaje comprendido entre –5v y –15v en el transmisor y entre -3v y –25v en el receptor.

- Un “0” lógico es un voltaje comprendido entre +5v y +15 v en el trasmisor y entre +3v y +25 v en el receptor.

El envío de niveles lógicos (bits) a través de cables o líneas de transmisión necesita la conversión a voltajes apropiados. En los microcontroladores para representar un 0 lógico se trabaja con voltajes inferiores a 0.8v, y para un 1 lógico con voltajes mayores a 2.0V. En general cuando se trabaja con familias TTL y CMOS se asume que un “0” lógico es igual a cero Volts y un “1” lógico es igual a cinco Volts.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial del ordenador, ya que son diferentes a los que utilizan los microcontroladores y los demás circuitos integrados. Por lo tanto se necesita de una interfase que haga posible la conversión de los niveles de voltaje a los estándares manejados por los CI TTL.

3.1.3 Conector de la comunicación serial según la Norma RS-232:

La norma especifica dos tipos de conectores: DB-25 de 25 pines, que es similar al del puerto paralelo, y el DB-9 de 9 pines, que es más barato y más utilizado. En cualquier caso, no se suele emplear más de 9 pines en el conector DB-25. Ambos conectores con machos en la parte trasera de la computadora, así que para conectar algún dispositivo, se necesitará un conector DB-25 o DB-9 hembra. A continuación se muestra una figura con la distribución de los pines en el DB-9. El DB-25 es similar al del puerto paralelo.

"Conectores Seriales Macho y Hembra"

La asignación de cada pin, tanto en el DB-25 como en el DB-9 se muestra en la siguiente tabla:

"Descripción de Pines del Conector Serial DB-25 y DB9"

NÚMERO DE PIN		SEÑAL	DESCRIPCIÓN	E/S
DB-25	DB-9			
1	1	-	Chasis (Protección eléctrica)	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E

El término de Entrada y Salida es con respecto a al DTE (la computadora).

La función de cada pin se muestra en la siguiente tabla:

SEÑAL	DESCRIPCIÓN	FUNCIÓN
TxD	Transmit Data	Salida de datos seriales
RxD	Receive Data	Recepción de datos seriales
CTS	Clear To Send	Indica que el MODEM esta listo para intercambiar datos
DCD	Data Carrier Detect	Detección de portadora
DSR	Data Set Ready	Indica al DTE que el MODEM esta listo para comunicarse
DTR	Data Terminal Ready	Indica al MODEM que el DTE esta listo para comunicarse
RTS	Request To Send	Indica al MODEM que el DTE esta listo para intercambiar datos
RI	Ring Indicador	Detección del sonido de llamada

"Función de Pines del Conector Serial DB-25 y DB9"

Figura del conector serial-USB

3.1.4 Transmisión y Recepción de Datos

La transmisión y recepción de Datos a través del Puerto Serial utiliza los pines TxD (para transmitir) y RxD (Para recibir). La comunicación del RS-232C es asíncrona, esto quiere decir que los datos no son enviados de acuerdo a una señal de reloj, sino que, cada byte de dato, es enviado utilizando un bit de inicio. La transmisión de un carácter o byte se realiza de la siguiente forma:

La línea que transmite los datos en serie está inicialmente en estado alto. Al comenzar la transferencia, se envía un bit a 0 ó **bit de inicio**. Tras él irán los **bits de datos** a transmitir (puede configurarse para que sean 8, 7, 6 o 5 bits de datos); estos bits están espaciados con un intervalo temporal fijo y preciso, ligado a la **velocidad de transmisión** que se esté empleando. Tras ellos podría venir o no un **bit de paridad** generado automáticamente. El bit de paridad indica si el número de bits transmitidos es par o impar se utiliza para detectar fallos en la transmisión. Al final, aparecerá un bit a 1, que es el bit de parada o **bit de stop** (también puede configurarse para que sea un bit y medio o 2 bits de stop). Lo de medio bit significa que la señal correspondiente en el tiempo a un bit dura la mitad; realmente, en

comunicaciones se utiliza el término **baudio** para hacer referencia a las velocidades, y normalmente un baudio equivale a un bit por segundo. La presencia de bits de inicio y parada permite sincronizar el dispositivo emisor con el receptor, haciendo que los relojes de ambos vayan a la par. Es por esto que también se dice que el RS-232 es asíncrono por carácter y síncrono por bit.

Tanto el dispositivo a conectar como la computadora (o el programa germinal) tienen que usar el mismo protocolo serie para comunicarse entre si. Puesto que el estándar RS-232 no permite indicar en que modo se esta trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Como ya se ha visto, los parámetros que hay que configurar son:

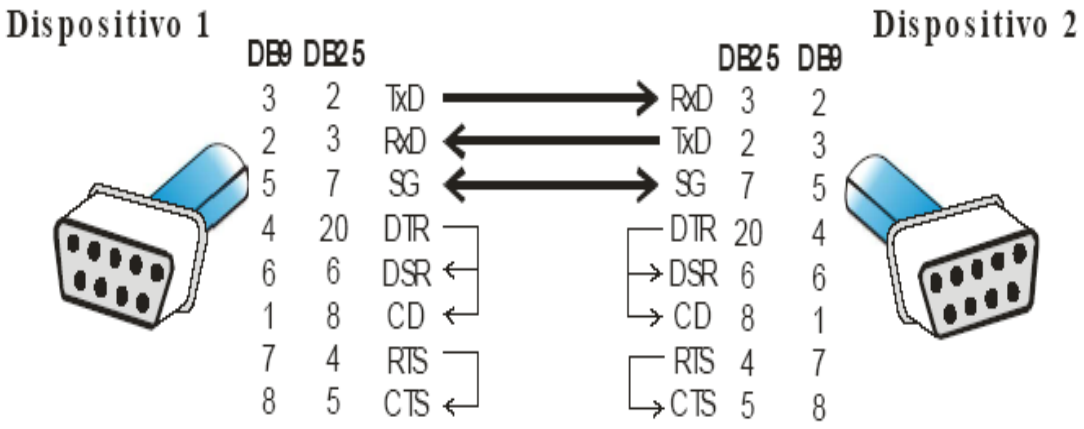
- Número de bits de datos a transmitir
- Bit de paridad
- Numero de bits de stop
- Velocidad de transmisión

Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que la computadora está encendida, DSR que el aparato conectado a dicho puerto esta encendido, RTS que la computadora puede recibir datos (porque no esta ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, presencia de datos.

3.1.5 Puerto serial en NULL

El modo NULL del puerto serial es utilizado para conectar dos DTEs juntos. Esta es una forma muy común, fácil y barata de transferir datos entre dos computadoras, entre una computadora y un microcontrolador con interfaz de comunicación serial.

En el siguiente diagrama se muestra el cableado de pines en modo NULL:

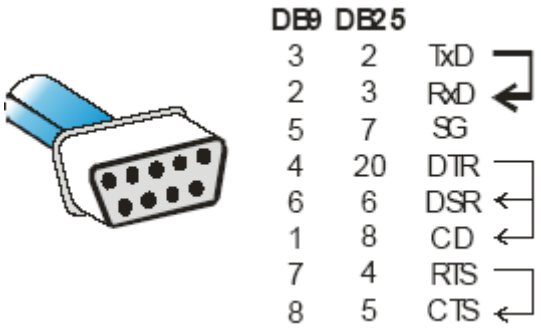


"Cableado de Pines en Modo Null"

Se puede observar que solo se necesitan 3 líneas para interconectarse: TxD, RxD y SG. El resto de los pines se encuentran interconectados entre sí como se

observa en el diagrama. La teoría de funcionamiento de esta conexión es muy fácil de entender: los datos transmitidos por el dispositivo 1, deben ser recibidos por el dispositivo 2, entonces conectamos TxD del dispositivo 1 con RxD del dispositivo 2. Los datos transmitidos por el dispositivos 2 deben ser recibidos por el dispositivo 1, entonces conectamos TxD del dispositivo 2 con RxD del dispositivo 1. La 18 terminal de tierra debe ser común para los dos.

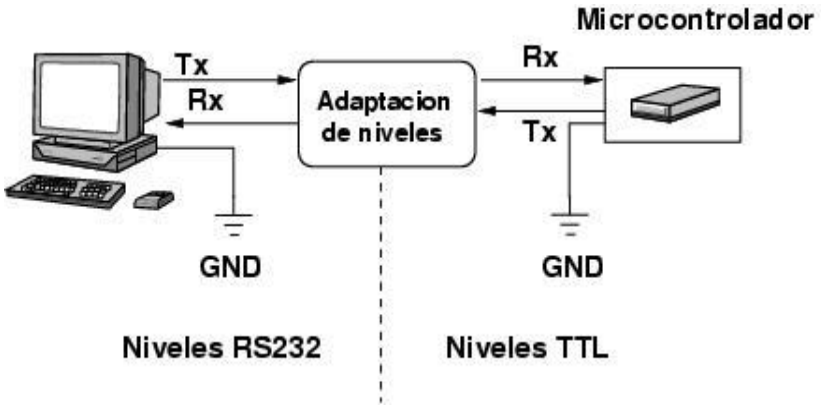
Utilizando el modo NULL se puede realizar un arreglo llamado “conexión de retroalimentación”, el cual consiste en conectar el pin transmisor del puerto con el pin receptor del mismo puerto, tal como se observa a continuación:



“Conexión de Retroalimentación”

3.1.6 Conexión de un Microcontrolador y PC

Para conectar el PC a un microcontrolador por el puerto serie en modo Null, se utilizan las señales Tx, Rx y GND. El PC utiliza la norma RS232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios. Los microcontroladores normalmente trabajan con niveles TTL (0-5v). Es necesario por tanto intercalar un circuito que adapte los niveles:



"Acoplamiento de Voltajes entre Microcontrolador y PC"

3.1.7 El Circuito MAX-232

Este circuito soluciona los problemas de niveles de voltaje cuando se requiere enviar unas señales digitales sobre una línea RS-232.

Este chip se utiliza en aquellas aplicaciones donde no se dispone de fuentes dobles de +12 y –12 Volts. El MAX 232 necesita solamente una fuente de +5V para su operación, internamente tiene un elevador de voltaje que convierte el voltaje de +5V al de doble polaridad de +12V y –12V. Cabe mencionar que existen una gran variedad de CI que cumplen con la norma RS-232 como lo son: MAX220, DS14C232, MAX233, LT1180A.

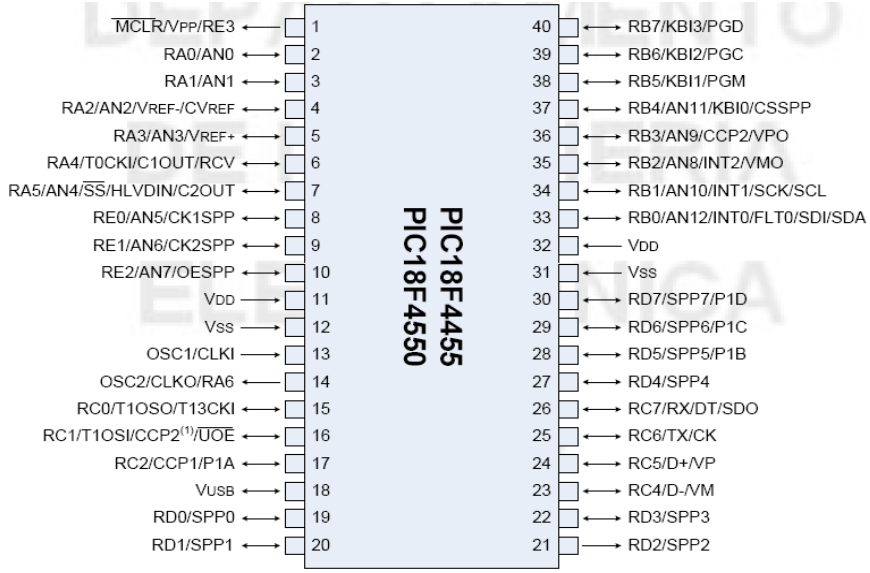
3.2 MICROCONTROLADORES

3.2.1 Familia de Microcontroladores 18F

Características fundamentales:

- Arquitectura RISC avanzada Harvard: 16- bit con 8- bit de datos.
- 77 instrucciones
- Desde 18 a 80 pines
- Hasta 64K bytes de programa (hasta 2 Mbytes en ROMless)
- Multiplicador Hardware 8x8
- Hasta 3968 bytes de RAM y 1KBytes de EEPROM
- Frecuencia máxima de reloj 40Mhz. Hasta 10 MIPS.
- Pila de 32 niveles.
- Múltiples fuentes de interrupción
- Periféricos de comunicación avanzados (CAN y USB)

3.2.2 Estructura del patillaje del 18F4550



"Pines del Microcontrolador 18F4550"

3.2.3 Organización de memoria

El uC PIC18F4550 dispone de las siguientes memorias:

- Memoria de programa: memoria flash interna de 32.768 bytes

  – Almacena instrucciones y constantes/datos

  – Puede ser escrita/leída mediante un programador externo o durante la ejecución programa mediante unos punteros.

- Memoria RAM de datos: memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial.

  –Almacena datos de forma temporal durante la ejecución del programa

  – Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones

- Memoria EEPROM de datos: memoria no volátil de 256 bytes.

  – Almacena datos que se deben conservar aun en ausencia de tensión de alimentación

  – Puede ser escrita/leída en tiempo de ejecución a través de registros

- Pila: bloque de 31 palabras de 21 bits

  – Almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina

- Memoria de configuración: memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).

3.2.4 ARQUITECTURA HARDVARD:

- El uC PIC18F4550 dispone buses diferentes para el acceso a memoria de programa y memoria de datos (arquitectura Harvard):
- Bus de la memoria de programa:

  – 21 líneas de dirección

  – 16/8 líneas de datos (16 líneas para instrucciones/8 líneas para datos)

- Bus de la memoria de datos:

  – 12 líneas de dirección

  – 8 líneas de datos

3.2.5 MEMORIA RAM DE DATOS:

- El uC PIC18F4550 dispone una memoria RAM de datos 2.048 bytes (8 bancos de 256 bytes). Además dispone de 160 bytes dedicados a los registros de función especial (SFR’s) situados en la parte alta del banco 15.

- Para acceder a un byte de la memoria RAM de datos primero debe seleccionarse el banco al que pertenece el byte mediante el registro de selección de banco (BSR) y a continuación direccionar el byte dentro del banco. Además existe una modalidad de acceso rápido a las 96 posiciones de la parte baja del banco 0 y a los 160 bytes de SFR's (banco de acceso rápido).

3.3 ENCRIPTACIÓN

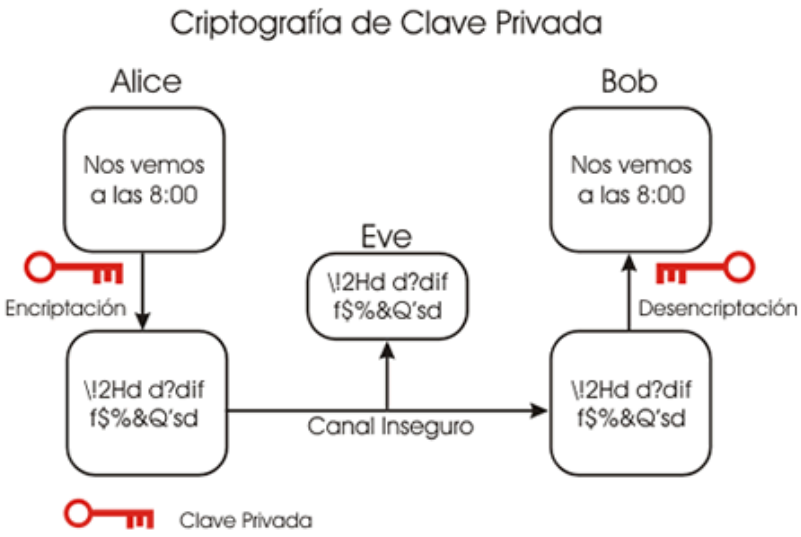
3.3.1 Algoritmos de clave simétrica

Los algoritmos de clave simétrica, también llamados de clave secreta o privada, son los algoritmos clásicos de encriptación en los cuales un mensaje es encriptado utilizando para ello una cierta clave sin la cual no puede recuperarse el mensaje original.

El esquema básico de los algoritmos de clave simétrica es:

MENSAJE + CLAVE = CÓDIGO (encriptación)

CÓDIGO + CLAVE = MENSAJE (desencriptación)



"Criptografía de Clave Privada"

Esto se lleva a cabo sustituyendo porciones del mensaje original por porciones de mensaje encriptado usando la clave. La sustitución puede ser de varias formas:

Monoalfabética:

Cuando se encripta, cada carácter encriptado corresponde a un carácter del mensaje original y viceversa.

Homofónica:

Cuando un carácter de texto original se encripta en varios caracteres del texto encriptado.

Poligráfica:

Cuando n caracteres del mensaje original generan n caracteres del mensaje encriptado.

Polialfabética:

Cuando n caracteres del texto original se encriptan en m caracteres del texto encriptado (m≠n). Cabe destacar que la sustitución poligráfica y la sustitución homofónica son casos particulares de la sustitución polialfabética.

Sistemas monoalfabéticos y polialfabéticos:

Un algoritmo de encriptación por clave privada es monoalfabético si cada ocurrencia de un mismo carácter en el mensaje original es reemplazada siempre por un mismo carácter en el código cifrado.

Un algoritmo de encriptación por clave privada es polialfabético si cada ocurrencia de un mismo carácter en el mensaje original es reemplazada por distintos caracteres en el código cifrado.

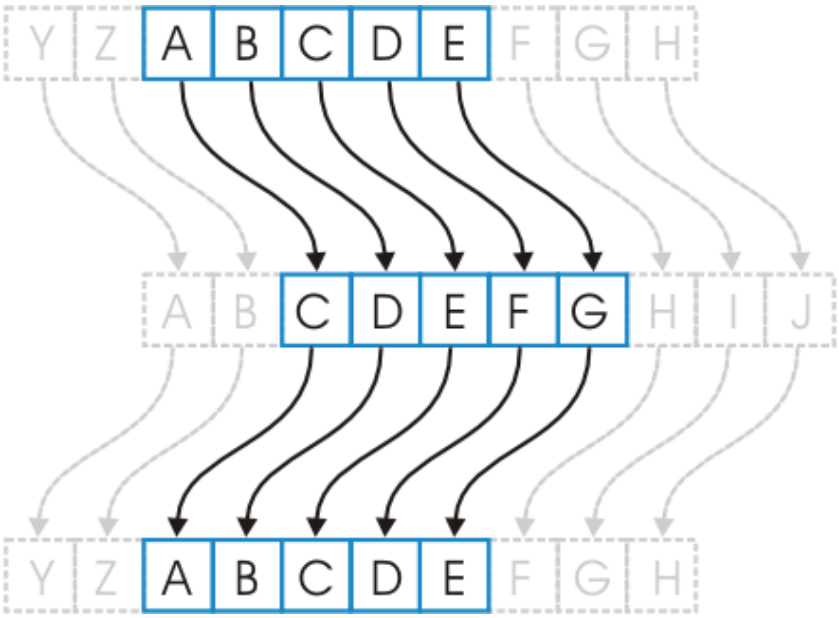
3.3.1.1 Criptosistema Caesar

El sistema Caesar o desplazamientos Caesar es una de las técnicas de criptografía más simples y mayormente difundidas. Fue el primero que se utilizó del cual se tienen registros. El sistema es monoalfabético y es realmente muy malo, su único valor es el valor histórico de haber sido el primero.

En un sistema Caesar la encriptación se hace por sustitución, cada carácter del mensaje original será reemplazado por un carácter en el mensaje cifrado, el carácter cifrado se obtiene avanzando 'k' pasos en el alfabeto a partir del carácter original. Obviamente 'k' es la clave.

Ejemplo con **k=2**:

Si el texto original es "ABCDE" se codifica como "CDEFG"



“Desplazamiento del Abecedario”

Este es todo el secreto del sistema ‘CAESAR’ veamos ahora cuan malo es:

Criptoanálisis

Para el sistema Caesar la tarea de un criptoanalista es realmente sencilla, pues la cantidad de posibles claves de este sistema es muy limitada. Trabajando con un alfabeto de 25 caracteres hay solamente 25 posibles claves (1…25) la clave 26, es idéntica a la clave 1, la clave 27 es idéntica a la 2 y así sucesivamente. De esta forma el criptoanalista puede chequear una por una las 25 posibles claves y observando el resultado obtenido se llega fácilmente y en muy poco tiempo al mensaje original.

Este es un criptosistema cuyo punto débil es el espacio de claves, como hay muy pocas claves posibles la técnica mas recomendable para el criptoanalista es simplemente probar todas las posibles claves. A este método se lo denomina ‘ataque por fuerza bruta’ y cuando el tiempo estimado para el ataque es razonable es un método infalible.

3.3.1.2 Criptosistema Hill

Este sistema esta basado en el álgebra lineal y ha sido importante en la historia de la criptografía. Fue Inventado por Lester S. Hill en 1929, y fue el primer sistema criptográfico polialfabético que era práctico para trabajar con más de tres símbolos simultáneamente.

Este sistema es polialfabético pues puede darse que un mismo carácter en un mensaje a enviar se encripte en dos caracteres distintos en el mensaje encriptado.

Suponiendo que trabajamos con un alfabeto de 26 caracteres.

Las letras se numeran en orden alfabético de forma tal que A=0, B=1, … ,Z=25

“ Letras numeradas en orden alfabético”

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Se elije un entero d que determina bloques de d elementos que son tratados como un vector de d dimensiones.

Se elije de forma aleatoria una matriz de d × d elementos los cuales seran la clave a utilizar. Los elementos de la matriz de d × d serán enteros entre 0 y 25, además la matriz M debe ser inversible en $\mathbb{Z}_{26}^{n}$.

Para la encriptación, el texto es dividido en bloques de d elementos los cuales se multiplican por la matriz d × d.

Todas las operaciones aritméticas se realizan en la forma modulo 26, es decir que 26=0, 27=1, 28=2 etc. Dado un mensaje a encriptar debemos tomar bloques del mensaje de "d" caracteres y aplicar:

$M \times P_i = C$, donde C es el código cifrado para el mensaje P_i

Criptoanálisis

El sistema de Hill plantea a los criptoanalistas problemas muchos mayores a los que planteaba 'CAESAR'. Para empezar el espacio de claves es mucho mayor, en este caso es de 4C25, es decir las permutaciones de 4 elementos tomados de entre 25 posibles. Y usando una matriz más grande la cantidad de posibles claves se puede hacer tan grande como sea necesario para hacer que sea imposible un ataque por fuerza bruta.

Lo mejor que puede hacer un criptoanalista es tratar de conseguir un código para el cual se conozca una parte del mensaje. Y ver si con ambos datos es capaz de encontrar cual fue la matriz utilizada para encriptar el mensaje.

3.3.1.3 Criptosistema Playfair

Este sistema criptográfico fue inventado en 1854 por Charles Wheatstone, pero debe su nombre al Baron Playfair de St Andrews quien promovió el uso de este criptosistema.

El algoritmo utiliza una tabla o matriz de 5x5.

La tabla se llena con una palabra o frase secreta descartando las letras repetidas. Se rellenan los espacios de la tabla con las letras del alfabeto en orden. Usualmente se omite la "W" y se utiliza la "V" en su lugar o se reemplazan las "J" por "I". Esto se hace debido a que la tabla tiene 25 espacios y el alfabeto tiene 26 símbolos. La frase secreta usualmente se ingresa a la tabla de izquierda a derecha y arriba hacia abajo o en forma de espiral, pero puede utilizarse algún otro patrón. La frase secreta junto con las convenciones para llenar la tabla de 5x5 constituye la clave de encriptación.

Por ejemplo:

Si la frase secreta es "CRIPTOSISTEMA PLAYFAIR"

Llenaremos de izquierda a derecha y arriba hacia abajo y omitiremos la W

"Clave de Encriptación"

C	**R**	**I**	**P**	**T**
O	**S**	**E**	**M**	**A**
L	**Y**	**F**	**B**	**D**
G	**H**	**J**	**K**	**N**
Q	**U**	**V**	**X**	**Z**

La encriptación se realiza de la siguiente forma:

El mensaje original que se desea encriptar es dividido en bloques de dos caracteres cada uno y se le aplican las siguientes cuatro reglas en orden

1. Si en el bloque las dos letras son la misma, se reemplaza la segunda generalmente por una X (o alguna letra poco frecuente) y se encripta el nuevo par.

2. Si las dos letras del bloque aparecen en la misma fila de la tabla, cada una se reemplaza por la letra adyacente que se encuentra a su derecha (si es la letra que se encuentra en la última posición a la derecha de la fila se la reemplaza con la primera de la izquierda de esa fila). Ej. SM se reemplazará por EA y AE por OM.

3. Si las dos letras del bloque aparecen en la misma columna de la tabla, cada una se reemplaza por la letra adyacente que se encuentra por debajo (si es la letra que se encuentra en la última posición inferior de la columna se la reemplaza con la primera de arriba de esa columna). Ej. LC se reemplazará por GO y GQ por QC.

4. Si las letras no se encuentran en la misma fila ni columna se las reemplaza se determina el rectángulo formado por los dos caracteres y se encripta tomando los caracteres que están en las esquinas del rectángulo y en la misma fila que el carácter a encriptar. Ej. SB se reemplazará por MY y KR por HP.

“Trasposición de elementos”

C	R	I	P	T
O	S	E	M	A
L	Y	F	B	D
G	H	J	K	N
Q	U	V	X	Z

Para desencriptar se aplican estas cuatro reglas en forma inversa, descartando las "X" que no tengan sentido en el mensaje final.

Criptoanálisis:

El sistema Playfair es un sistema de encriptación bastante bueno, la cantidad de posibles claves es enorme ya que son las permutaciones de 25 elementos tomados de entre 26 lo cual da un número muy grande como para derrotar al algoritmo por fuerza bruta. Además es un sistema polialfabético por lo que un análisis de la frecuencia de aparición de cada carácter en el código cifrado no nos aporta nada.

La técnica que se debe utilizar con el esquema Playfair consiste en analizar la frecuencia de aparición de los pares de letras (diagramas) y compararlas con los diagramas mas frecuentes del idioma en el cual se supone que se escribió el mensaje original, en castellano los diagramas mas probables son:

Ordenados por frecuencia:

ES, EN, EL, DE, LA, OS, AR, UE, RA, RE, ER, AS, ON, ST, AD, AL, OR, TA, CO

El criptoanalista deberá analizar cual es el diagrama mas ocurrente en el código cifrado y ver que ocurre si se lo reemplaza por 'ES', de esta forma se van probando distintas combinaciones entre los diagramas mas frecuentes en el mensaje cifrado y los diagramas mas frecuentes del idioma hasta que se consigue descifrar el texto. Esta es una técnica muy habitual del criptoanálisis y suele funcionar muy bien.

3.3.1.4 Cifrado Bífido

El método Bífido es un cifrado fraccionario. Es decir que cada letra viene representada por una o más letras o símbolos, y donde se trabaja con estos símbolos más que con las letras mismas.

El método comienza con la utilización de un alfabeto desordenado en una matriz 5x5. Observa un ejemplo donde la clave para el alfabeto desordenado es DIPLOMA:

"Alfabeto Clave"

*	1	2	3	4	5
1	D	IJ	P	L	O
2	M	A	B	C	E
3	F	G	H	K	N
4	Q	R	S	T	U
5	V	W	X	Y	Z

Al ser un cuadro sólo de 5x5 nos vemos obligados a cifrar de la misma forma la I y la J. El contexto nos permitirá distinguir cual de las dos letras se pretendía cifrar.

Para cifrar el texto en claro se escriben los equivalentes numéricos de cada letra, utilizando sus "coordenadas". Si por ejemplo el texto en claro es:

VEN A LAS TRES

El equivalente numérico es:

51 25 35 22 14 22 43 44 42 25 43

Que dividido en dos partes queda:

5 1 2 5 3 5 2 2 1 4 2

2 4 3 4 4 4 2 2 5 4 3

Si ahora leemos los números como columnas en lugar de por filas resulta: **52 14 23 54 34 54 22 22 15 44 23**

Que volviendo a consultar la tabla resulta en el mensaje cifrado:

WLBYKYAAOTB

Este método altera la frecuencia de los caracteres a diferencia de lo que ocurre por ejemplo con los cifrados monoalfabéticos. Admite algunas variaciones como por ejemplo dividir la lista en 3, 4,...n partes

3.3.1.5 El cifrado de Gronsfeld

El cifrado de Gronsfeld es del tipo conocido como polialfabéticos. Esto significa que se usa más de un alfabeto cifrado para poner en clave el mensaje y que se cambia de uno a otro según se pasa de una letra del texto en claro a otra.

Es decir que deben tenerse un conjunto de alfabetos cifrados y una forma de hacer corresponder cada letra del texto original con uno de ellos. Para dejar esto más claro veamos una de las tablas que se usaban antes de la era de los ordenadores para hacer un cifrado de este tipo.

“Alfabetos para el Cifrado Gronsfeld”

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0:	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
1:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
2:	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
3:	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
4:	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
5:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
6:	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
7:	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
8:	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
9:	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Ahora cogemos una serie de dígitos como clave. Supongamos que usamos 1203456987. Vamos a utilizar la tabla para el cifrado del mensaje:

EMBARCAMOS AL ANOCHECER

Empezamos por escribir la clave debajo del texto original las veces que sea necesario:

EMBARCAMOS AL ANOCHECER

1203456987 12 034569871

Y sustituimos cada letra por la correspondiente del alfabeto que indica el número debajo de él.

HRDHCPROLL DQ CUZPYGZXU

El cifrado de Gronsfeld altera la frecuencia de las letras del texto, pues por ejemplo la letra más corriente en castellano, la E, se cifra de forma diferente según su posición en el texto original.

3.3.1.6 El cifrado ADFGVX

Este cifrado apareció por primera vez al final de la primera guerra mundial, antes de lo que iba a ser la ofensiva definitiva del ejército alemán. La fortaleza del método estribaba en que era uno de los primeros en unir transposición y sustitución, dos procesos que producen, en la terminología actual, difusión y confusión.

Según explica Simon Singh en "Los códigos secretos", fue descifrado por primera vez por un francés, Georges Painvin, que trabajo día y noche perdiendo quince kilos en el proceso.

Se empieza disponiendo las 26 letras del alfabeto anglosajón y los diez dígitos en una matriz 6x6. Las líneas y las columnas van encabezados por las letras A D F G V X. El modo de ordenar letras y números en la cuadrícula forma parte de la clave y necesita ser comunicada al receptor del mensaje.

Pongamos por ejemplo:

"Cuadricula Clave"

	A	D	F	G	V	X
A	0	q	9	z	7	c
D	m	u	1	h	f	2
F	4	8	w	n	r	g
G	l	6	v	t	p	a
V	y	3	d	5	e	k
X	j	s	i	o	b	x

En primer lugar tomaremos cada letra del mensaje en claro sustituyéndola por las letras correspondientes a su fila y columna. Por ejemplo el número 5 sería sustituido por las letras VG y la j por el par de letras XA.

Pongamos como ejemplo el mensaje: Envíen municiones

“Sustitución de Caracteres”

E n v i e n m u n i c i o n e s

VV FG GF XF VV FG DA DD FG XF AX XF XG FG VV XD

Hasta aquí solo un cifrado ordinario por sustitución que se descifra con un análisis de frecuencia si se dispone de suficiente texto. Sigue otra frase con una transposición dependiente de una palabra clave. Supongamos que la clave es WHISKY. Las letras de la clave se escriben en la cabecera de una cuadrícula.

El texto que hemos cifrado antes se escribe por filas en dicha cuadrícula así:

“Palabra Clave”

W	H	I	S	K	Y
V	V	F	G	G	F
X	F	V	V	F	G
D	A	D	D	F	G
X	F	A	X	X	F
X	G	F	G	V	V
X	D	A	A	A	A

Donde hemos añadido dos caracteres de relleno (00 ~ AA AA) para que el cuadro quede completo. Ahora las columnas de la cuadrícula se cambian de posición de modo que las letras de la clave queden en orden alfabético:

“Ordenamiento Alfabético de Columnas”

H	I	K	S	W	Y
V	F	G	G	V	F
F	V	F	V	X	G
A	D	F	D	D	G
F	A	X	X	X	F
G	F	V	G	X	V
D	A	A	A	X	A

Para acabar leemos por columnas la cuadrícula y el resultado es el texto cifrado, en este caso:

VFAFGDFVDAFAGFFXVAGVDXGAVXDXXXFGGFVA

El inconveniente de este tipo de encriptación, es que entrega el doble de caracteres a encriptar, es decir, si tenemos un texto con 10 caracteres, al aplicar este algoritmo nos devuelve un texto encriptado el cual posee 20 caracteres.

3.4 ESTADO DEL ARTE

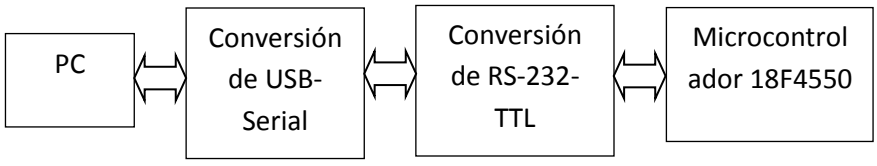
Existen diversos trabajos que sirven de referencia para el presente proyecto los cuales son: la “Teoría matemática de la comunicación” por Claude Shannon en 1948 y el trabajo donde se introduce el concepto de criptografia de clave pública, publicado por Whitfield Diffie en 1975, además, tenemos otro antecedente, proporcionada por la tesis del Dr. Héctor Ricardo Hernández de León; la cual se titula “Esquema de Encripción Autentificado empleando verificación compartida” en 2001.

4. DISEÑO E IMPLEMENTACIÓN

4.1 -DISEÑO DEL SISTEMA DE ENCRIPTACION

4.1.1 Encripción de archivos de texto plano

El sistema cuenta con programa desarrollado en Visual Basic, el cual es la interfaz que nos permite encriptar por hardware un archivo de texto plano, a continuación se muestra un diagrama a bloques del sistema:



“Diagrama a bloques del sistema”

El sistema implementado en la PC, presenta una interfaz muy amigable y fácil de instalar. La encriptación de lo archivos de texto plano esta limitado a 232 caracteres, pero la encriptación ADFGVX únicamente tiene la capacidad de encriptar 232 letras.

Para el Diseño del sistema de encriptación, se selecciono algunos algoritmos más adecuados que permitieran su implementación en un microcontrolador.

A continuación se presenta una explicación detallada de la forma en que se adaptaron los algoritmos seleccionados para la Encriptación de texto plano.

4.1.1.1 Encriptación Diseñada por el residente

4.1.1.1.1 Encriptación

Para realizar este tipo de encriptación primero se divide el valor de la clave pública entre 4, a este resultado se pasa a formato binario y los 4 bits menos significativos se rotan de posición y se almacena en la clave interna. El carácter a encriptar se pasa a binario y se le suma la clave interna, este valor obtenido se le invierten los 4 bits menos significativos y se le suma el valor de la posición cero en un arreglo de 8 bytes. El arreglo de 8 bytes posee 8 elementos de la serie Fibonacci[8]={ 1,1,2,3,5,8,13,21 }.

El resultado obtenido se le invierte otra vez los 4 bits menos significativos y estos procesos finalmente nos entregan nuestro carácter encriptado.

Para encriptar un segundo carácter ser realiza el mismo procedimiento, salvo que en el momento de sumar el valor del arreglo de 8 bytes, ahora se le suma el valor que se encuentre posicionado en la posición uno, para los caracteres siguientes re realiza la misma operación y cuando ya son 8 caracteres encriptados el apuntador se regresa a la posición cero, nuevamente.

Para realizar este tipo de encriptación se logran implementar dos conceptos muy importantes, como lo son la confusión y la difusión. Aunado a este también se efectúa una encriptación por bloques de 8 caracteres, lo que nos da más seguridad al texto encriptado, debido que no se encuentran patrones en los textos encriptados, esto debido, a que se trata de una encriptación polialfabética.

A continuación se muestra la encriptación de un carácter para mostrar el funcionamiento de este algoritmo.

Texto a Encriptar: ANAYELI

Calculo para Clave Interna:

Clave: 200

1.-Clave interna: 200/4=50= 00110010

2.-Invirtiendo los 4 bits menos significativos nos queda:

3.-Clave interna=0011 0100

Clave interna=52

El primer carácter a encriptar es A, cuyo valor ASCII es 65:

A=65

1.- salida=65+clave interna

2.- salida=65+52=117

3.- x=salida+figonacci(fibo)

X=117+1=118

X=118 X=v

Dependiendo la letra es la suma, de figonacci=(1,1,2,3,5,8,13,21),es decir; para **A** es 1, **N** es 1, **A** es 2, **Y** es 3; hasta el 21 que seria la octava letra, después la siguiente letra se volverá a iniciar con el 1,1,2…

Este procedimiento se realiza para cada carácter y el texto encriptado para la palabra ANAYELI es:

'vfw~^ŠD

4.1.1.1.2 Desencriptación

Para realizar la desencriptación de un archivo texto, es necesario contar con el conocimiento de la clave con la cual fue encriptado el texto. Seguido de esto se realizan las mismas operaciones al texto, pero esta vez comenzado a realizarlas en diferente orden, es decir, las ultimas operaciones se realizan primero y las primeras operaciones ser realizan al final. De esta manera obtenemos el texto desencriptado.

A continuación se Muestra un ejemplo con el texto Encriptado

“vfw~^ŠD”

Calculo para Clave Interna:

Clave: 200

1.-Clave interna: 200/4=50= 0011 0010

2.-Invirtiendo los 4 bits menos significativos nos queda:

3.-Clave interna=0011 0100

Clave interna=23

El primer carácter a encriptar es **v** , cuyo valor ASCII es 118:

1.- Se invierten los 4 bits menos significativos de este valor

118= 0111 0110 invirtiendo nos queda 118

X=118

2.- X = inversión de 4 bits del Valor 118- elemento cero en arreglo

X=118-1

X=117

3.-invirtiendo los 4 bits menos significativos de 117 nos queda:

X=117

4.- A este valor se le resta la clave interna:

X=117-52

X=65

EL Valor 65 corresponde en ASCII a la letra: A .

Este proceso se realiza con cada uno de los caracteres del texto encriptado hasta tener la palabra ANAYELI.

4.2. IMPLEMENTACION DEL SISTEMA DE ENCRIPTACION

A continuación se enumeran las actividades realizar para implementar el sistema:

1. el Desarrollar un Programa en una PC, que sea capaz de comunicarse con microcontrolador 18F4550, mediante la comunicación serial, utilizando Visual Basic.

2. Programar el microcontrolador para la comunicación serial.

3. Implementar la circuitería necesaria para realizar la comunicación serial entre la PC y el Microcontrolador.
4. Realizar una pequeña prueba entre el programa realizado en la PC, y el microcontrolador, para esto basta con probar enviando caracteres de la PC al Microcontrolador y luego un tiempo después k el microcontrolador los envié a la PC.

5. Implementar los algoritmos de encriptación mencionados anteriormente en el Microcontrolador 18F4550.

6. Simular el programa diseñado para el microcontrolador y revisar para posibles ajustes.

7. Grabar el archivo .Hex al Microcontrolador.

8. Realizar el Programa completo en Visual Basic, para que responda a los diferentes algoritmos a implementados.

4.3. PRUEBAS DE FUNCIONAMIENTO

Realizar pruebas con el microcontrolador utilizado, con el programa diseñado en la PC y el hardware necesario para el sistema. Pruebas del programa en el laboratorio, simulando la actividad que realizará de encriptar y desencriptar información, para verificar el funcionamiento por partes y luego en su conjunto.

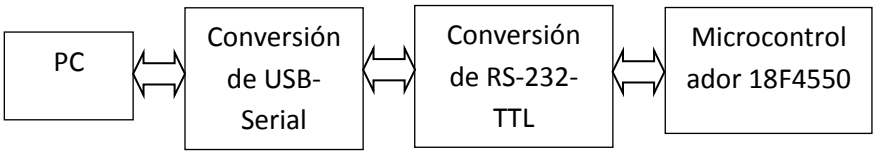
4.4. PRUEBAS FINALES DEL PROTOTIPO

Verificar el sistema utilizando una interfase serie USB y una computadora personal, efectuar las pruebas finales al sistema con el objetivo de validar el prototipo y verificar su funcionamiento. También, realizar la integración del manual y observaciones correspondientes.

5. ANÁLISIS DE RESULTADOS

5.1 DESCRIPCIÓN DEL SISTEMA

El sistema cuenta con programa desarrollado en Visual Basic, el cual es la interfaz que nos permite comprimir y encriptar por hardware un archivo de texto plano, a continuación se muestra un diagrama a bloques del sistema:



"Diagrama a bloques del sistema"

El sistema implementado en la PC, presenta una interfaz muy amigable y fácil de instalar. La encriptación de los archivos de texto plano esta limitado a 232 caracteres, la compresión que utiliza la codificación a 4 bits esta habilitada para 232 caracteres.

5.2 IMPLEMENTACIÓN DE LOS DIFERENTES ALGORITMOS EN EL MICROCONTROLADOR.

5.2.1 ENCRIPTACIÓN

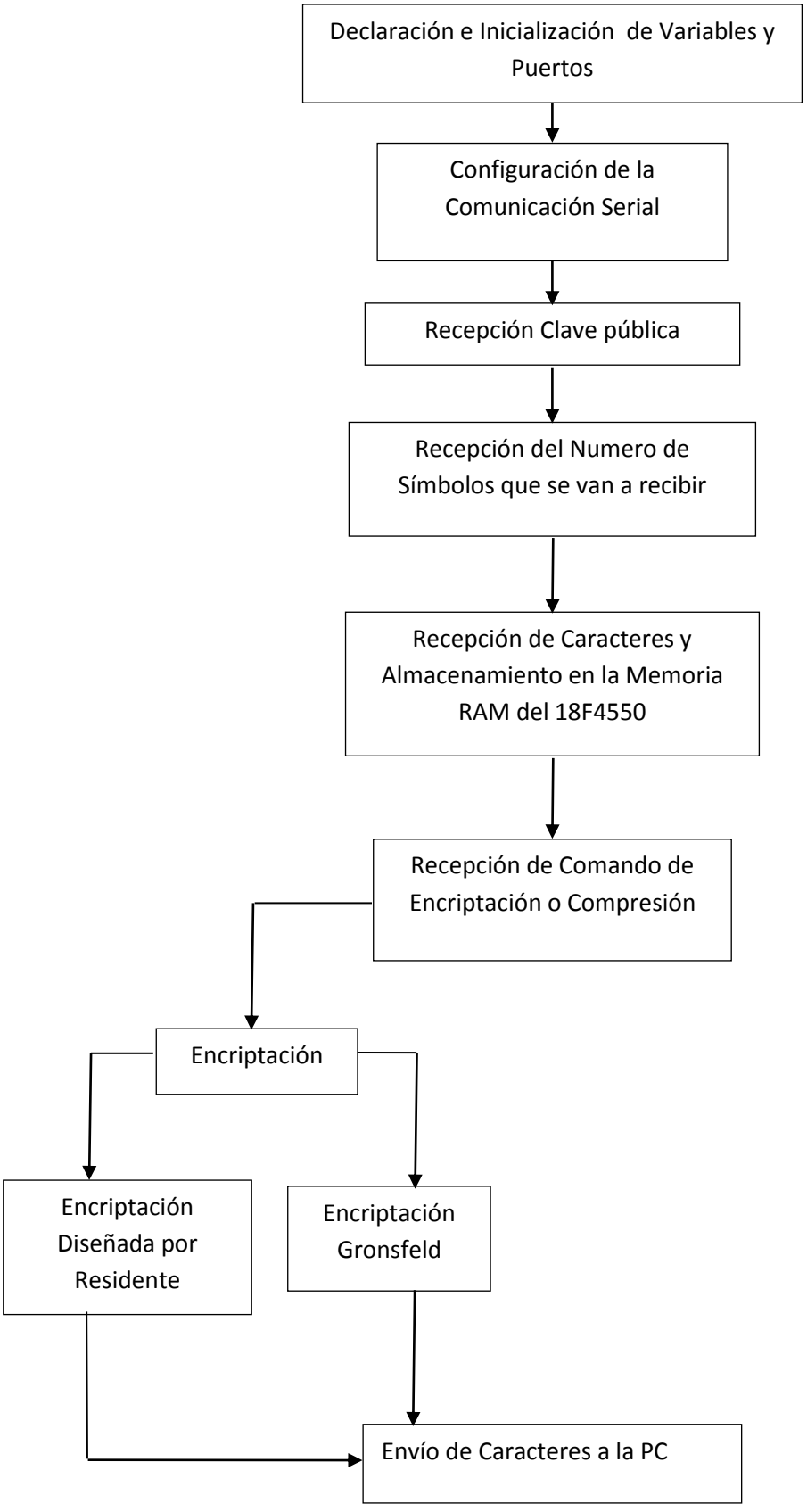
Algoritmo Diseñado Por Residente

Este algoritmo tiene la capacidad de Encriptar un texto que contenga 232 caracteres. Los textos deben estar basados en un alfabeto que incluye los siguientes caracteres:

SPC	!	"	#	$	%	_	'	(	)	*	+	,	-	.	/
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

Figura No. 1"Alfabeto Para Encriptación"

Diagrama de flujo del programa instalado en el Microcontrolador:



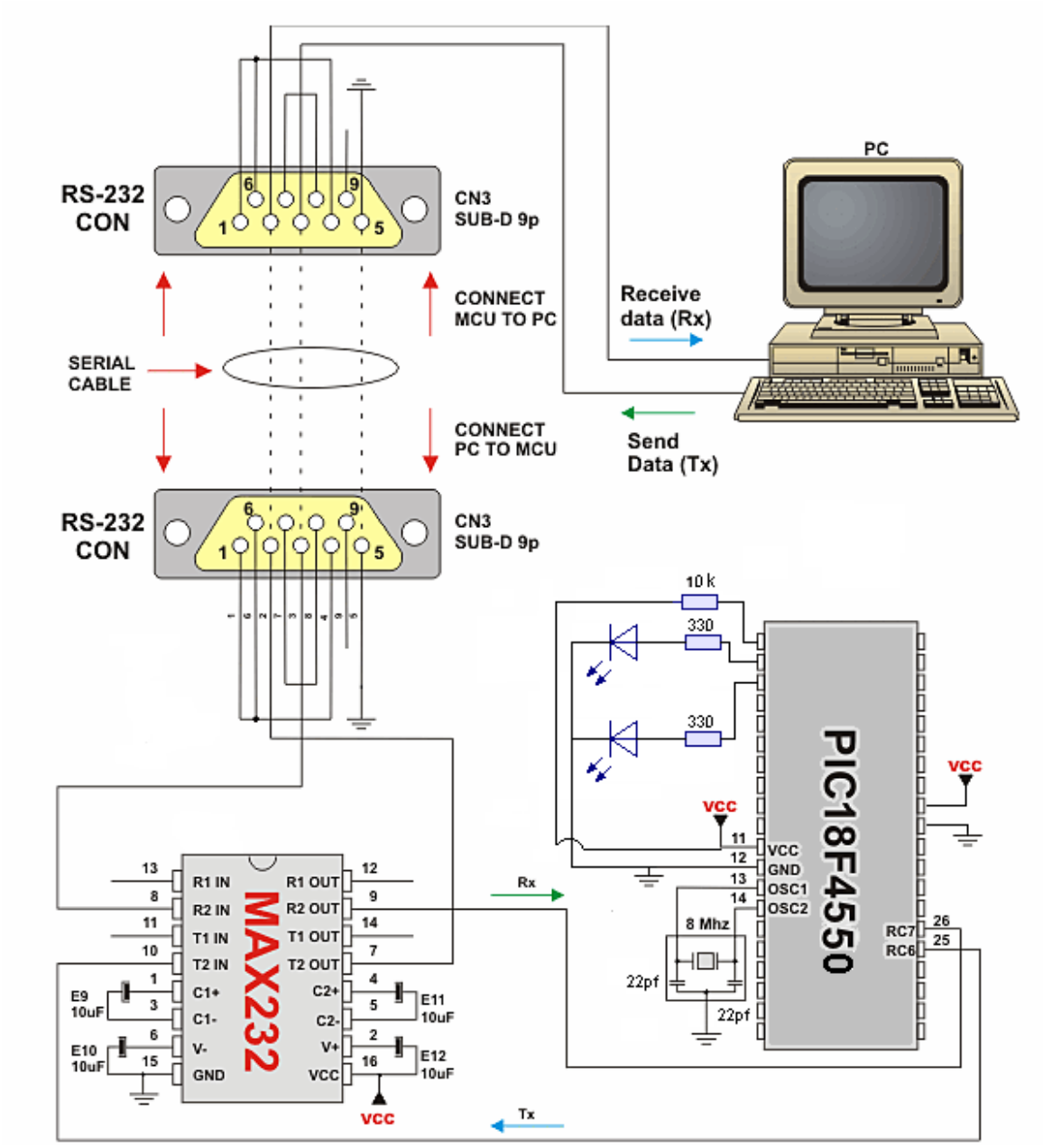
5.3 DESCRIPCIÓN DEL CIRCUITO

Para desarrollar el circuito se necesitaron los siguientes componentes:

- 1 MAX232
- 1 PIC 18F4550
- 1 Cristal de 8 Mhz
- 2 Capacitores de 22 pf
- 4 Capacitores electrolíticos de 1 uF a 16 V
- 1 Capacitor electrolítico de 1000 uF a 16 V
- 2 resistencias de 330 Ω
- 1 resistencia de 10KΩ
- 2 LEDS color rojo
- 1 base de 40 pines
- Conector DB9 hembra
- 1 placa fenolica de 10x10

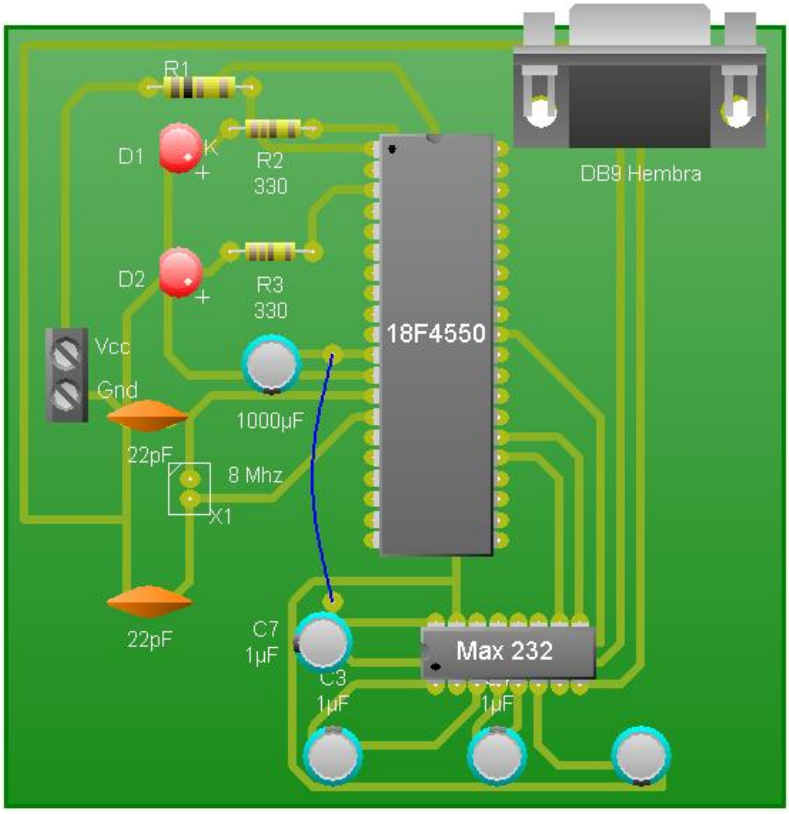
El circuito del sistema tiene un conector serial DB9 Hembra, requiere una fuente de alimentación de 5 Volts de corriente directa. Cuenta con dos LEDS de color rojo, el primero de ellos indica que el sistema se encuentra en funcionamiento, el segundo significa que se el microcontrolador ha terminado de ejecutar el algoritmo seleccionado.

La comunicación entre PC y Microcontrolador se realiza mediante el cable serial DB9, pero es posible realizar la conexión por medio de un adaptador de USB-Serial, y además realizar la instalación del software del Driver del Adaptador en la PC. Para el sistema se esta utilizando un adaptador de la empresa Prolific.



"El diagrama del Circuito del Sistema"

EL Diagrama del Circuito Impreso es el Siguiente:

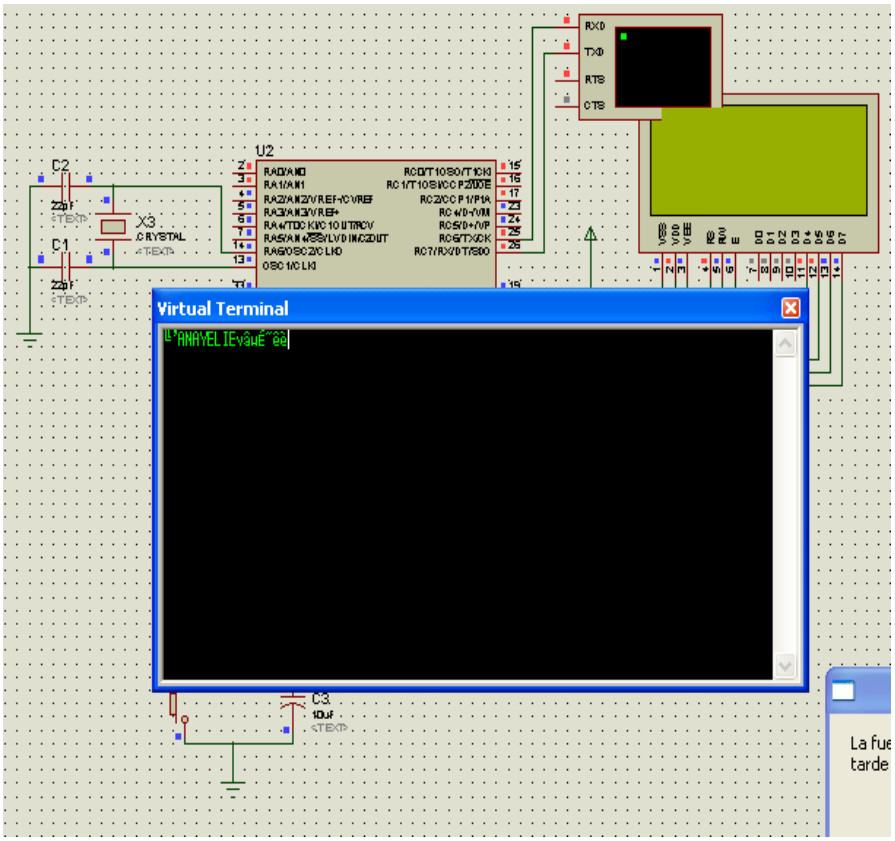


5.4 SIMULACIÓN

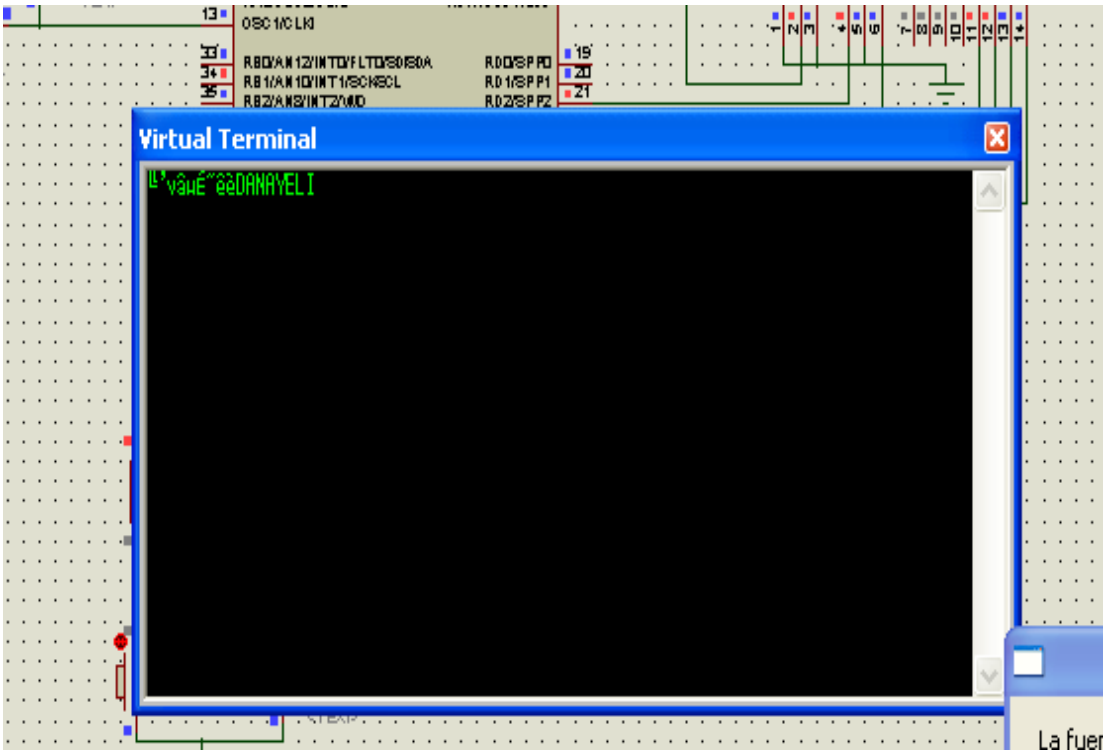
La simulación se realizo en ISIS Proteus 7.2 Professional, se utilizo un componente de Proteus para simular la comunicación serial llamado “Virtual Terminal”.

Para la Encriptación se tienen que enviar al microcontrolador: la clave, el número de caracteres a enviar, el texto, y el comando para encriptar. Para la encriptación

diseñada por el usuario se muestra la siguiente figura, donde se envía el símbolo ASCII de la clave 200=**È**, y del numero de caracteres 7=', ANAYELI y el Comando de Encriptación =E. Se observa como el sistema entrega el texto encriptado.



Para desencriptar, se envía la clave **È**=200, ‘=7 caracteres, vƒw~ˆš Comando de Desencriptación =D. Se observa como el sistema entrega el texto desencriptado.



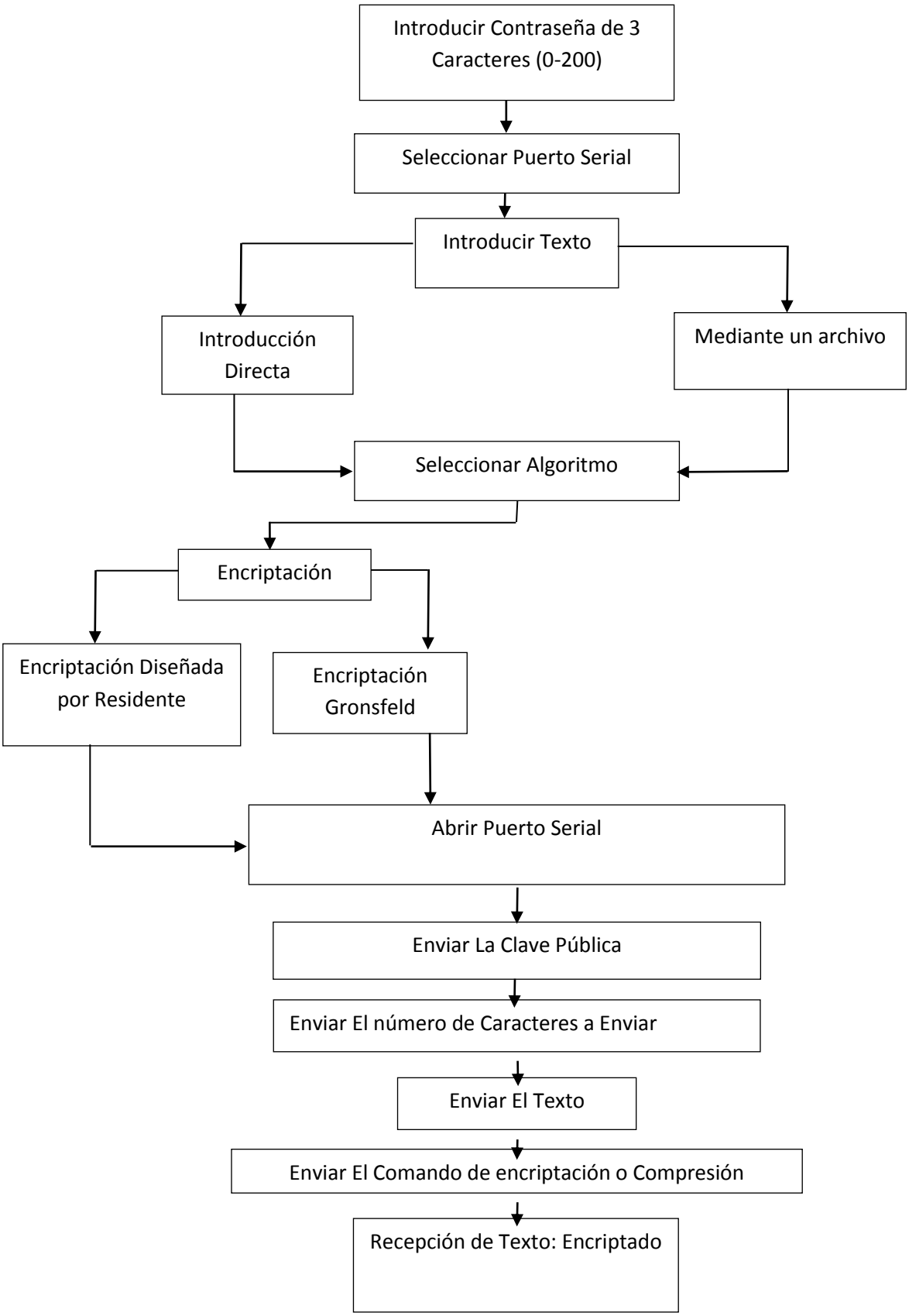
5.

5.5 DESCRIPCIÓN DEL PROGRAMA INSTALADO EN LA PC

VENTANA DE ENCRIPCIÓN Y DECRIPCIÓN

En esta ventana se deberá escribir la clave del número del 0 al 200, el número del puerto serial que la PC indique, se escribirá el texto a encriptar, o de lo contrario se copiara el texto a desencriptar. La ventana contiene también un menú para realizar los la encripción o decripción.

Diagrama de flujo del programa instalado en la PC



5. CONCLUSIÓN

El logro del objetivo principal del proyecto se alcanza debido a la realización de un análisis completo de los fundamentos teóricos y su comprobación correspondiente mediante la evaluación de los algoritmos propuestos en el sistema real.

Por los resultados obtenidos en la presente investigación, las conclusiones pueden ser presentadas de acuerdo a los siguientes aspectos:

Software generados; el programa implementado en la pc que operan con el sistema criptográfico es fácil de usar. También es flexible, pues tienen las funciones claramente separadas. El sistema se declara seguro por el tipo de algoritmo utilizado.

BIBLIOGRAFÍA

HERNÁNDEZ de León Héctor R.,“Esquema de Encripción Autentificado Empleando Verificación Compartida”, TESIS, CENIDET Cuernavaca, Morelos, México, Junio del 2001.

TORRANO Enrique, “El Criptosistema RSA, elaboración de herramientas y Criptoacelerador usando un procesador digital de señales”, TESIS, CENIDET Cuernavaca, Morelos, México, agosto de 1994.

Bruce Schneier, Applied Cryptography, Edition Second,1996.

http://forum.microchip.com/tm.asp

http://www.alldatasheet.com/

ANEXOS

CÓDIGO DEL MICROCONTROLADOR

```
unsigned short contador=0,texto[1000],m=0;

unsigned short caracter=0,enviar=0;

unsigned short aux1=0,nana=0,na=0,clave=0,claveinterna=0;

char salida,numero;

int salidax=0;

int dato=0,en=0;

/////

int invertir(int inver);

void encriptar(void);
```

```
    void desencriptar(void);
    int dato1=0,invertido=0;
    unsigned short comando=0,var1=0,fibonacci[8]={1,1,2,3,5,8,13,21},fibo=0;
    void main()
    {
    TRISB=0;
    TRISD=0;
    //TRISC.F6=0;
    //TRISC.F7=1;
    TRISC=128;
    PORTB=0;
    PORTD=0;
    //TXSTA=0b00100100; // Asincrono, alta velocidad, 8 bits, transmisión hab.
    //RCSTA=0b10010000; // Habilita el puerto y la recep de 8 bits,
    ADCON1 = 15;
    TRISA=0;
    PORTA=0;
// Initialize USART module (8 bit, 2400 baud rate, no parity bit..)
    Usart_Init(19200);
    PORTA=1;
     while(1)
    {
     while(na<1)    //esto es para la clave publica
      {
     if (Usart_Data_Ready())
       { PORTA=1;
         clave=Usart_Read();
    na++;
       }
     }
```

```
       while(nana<1)    //esto es para chekar el numero de simbolos k se van a enviar
        {
       if (Usart_Data_Ready())
         {
           numero=Usart_Read();
           numero=numero-32;
           nana++;
         }
}     //este for solo funciona una vez
 for(aux1=0;caracter<numero;m++)//para la recepcion de caracteres
        {
          if (Usart_Data_Ready())
          {
           texto[contador]=Usart_Read();
   //Usart_Write(salida);
contador++;
  caracter++;

          }
        }
          //esto es para recibir el comando de encripcion o Desencripcion
       while(var1<1)
        {
          if (Usart_Data_Ready())
          {
           comando=Usart_Read();
           var1++;
          }
        }
          if(comando==69)  //69 corresponde a la letra E de Encriptar
```

```
  {
  encriptar();
  enviar=0;
  while(enviar<numero)
  {
   salida=texto[enviar];
   Usart_Write(salida);
   enviar++;
  }
    comando=0;
  }
  if(comando==68)  //68 corresponde a la letra D de Desencriptar
  {desencriptar();
    while(enviar<numero)
  {
   salida=texto[enviar];
   Usart_Write(salida);
   enviar++;
  }
   comando=0;
  }
   PORTA=3;
 na=0;
nana=0;
aux1=0;
caracter=0;
contador=0;
enviar=0;
var1=0;
fibo=0;
```

```
} //ciclo infinito
      }//void main
      int invertir(int inver)
      {
      PORTB=0;
      PORTD=0;
      //dato1=4;
      //PORTB=dato1;
      PORTB=inver;
      //comienza inversion de bits
      if(PORTB.F0==0)
      {PORTD.F3=0;}
      if(PORTB.F0==1)
      {PORTD.F3=1;}
      if(PORTB.F1==0)
      {PORTD.F2=0;}
      if(PORTB.F1==1)
      {PORTD.F2=1;}
      if(PORTB.F2==0)
      {PORTD.F1=0;}
      if(PORTB.F2==1)
      {PORTD.F1=1;}
      if(PORTB.F3==0)
      {PORTD.F0=0;}
      if(PORTB.F3==1)
      {PORTD.F0=1;}
      //los siguientes datos no se invierten
      if(PORTB.F4==0)
      {PORTD.F4=0;}
      if(PORTB.F4==1)
```

```
{PORTD.F4=1;}
if(PORTB.F5==0)
{PORTD.F5=0;}
if(PORTB.F5==1)
{PORTD.F5=1;}
if(PORTB.F6==0)
{PORTD.F6=0;}
if(PORTB.F6==1)
{PORTD.F6=1;}
 (PORTB.F7==0)
{PORTD.F7=0;}
if(PORTB.F7==1)
{PORTD.F7=1;}
invertido=PORTD;
return(invertido) ;
}
void encriptar()
{
   enviar=0;
   fibo=0;
  clave=clave/4;
 claveinterna=invertir(clave);
 (en<200)
{
   while(enviar<numero)
           {
            dato=texto[enviar];
            salida=dato+claveinterna;
           salida=salida+fibonacci[fibo];
   texto[enviar]=salida;
```

```
            //Usart_Write(salida);
            if(fibo>7)
            {fibo=0;}
               enviar++;
           fibo++;
           }
      enviar=0;
        fibo=0;
        en++;
      }
en=0;
} //fin de funcion
void desencriptar()
      {  enviar=0;
         fibo=0;
       clave=clave/4;
       claveinterna=invertir(clave);
      while(en<200)
      {
        while(enviar<numero)
          {
             salida=texto[enviar];
           salida=salida-fibonacci[fibo];
            salida=salida-claveinterna;
            texto[enviar]=salida;
            //Usart_Write(salida);
            if(fibo>7)
            {fibo=0;}
           enviar++;
           fibo++;
```

```
    }
    enviar=0;
    fibo=0;
    en++;
}
    en=0;
}  //fin de función
```

CÓDIGO DEL PROGRAMA EN LA PC (VISUAL BASIC)

```
Private Sub Abrirencriptado_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB2.LoadFile CD1.FileName, 1
End Sub
Private Sub Abrirencriptar_Click(Index As Integer)
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB1.LoadFile CD1.FileName, 1
End Sub
Private Sub AbrirRTB1_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB1.LoadFile CD1.FileName, 1
End Sub
Private Sub AbrirRTB2_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
```

```
CD1.ShowOpen
RTB2.LoadFile CD1.FileName, 1
End Sub
Private Sub Acerca_Click(Index As Integer)
MsgBox " Encriptor Version 1.0", 64, "Encriptor de Archivos de Texto"
End Sub
Private Sub EncriptorDesencriptar_Click()
MSComm1.CommPort = Text3.Text
MSComm1.PortOpen = True
MSComm1.Output = Chr(Val(Text1.Text) + 32)
MSComm1.Output = Chr(Len(RTB2.Text) + 32)
MSComm1.Output = RTB2.Text
MSComm1.Output = Chr(68)
MsgBox "Envio Completo", 64, "Encriptor De Texto"
RTB1.Text = MSComm1.Input
MSComm1.PortOpen = False
End Sub

Private Sub EncriptorEncriptar_Click()
MSComm1.CommPort = Text3.Text
MSComm1.PortOpen = True
MSComm1.Output = Chr(Val(Text1.Text) + 32)
MSComm1.Output = Chr(Len(RTB1.Text) + 32)
MSComm1.Output = RTB1.Text
MSComm1.Output = Chr(69)
MsgBox "Envio Completo", 64, "Encriptor De Texto"
RTB2.Text = MSComm1.Input
MSComm1.PortOpen = False
```

```
End Sub
Private Sub Form_Load()
MSComm1.Settings = "19200,n,8,1"
Contextual.Visible = False
End Sub
Private Sub Label6_Click()
End Sub
Private Sub GuardarDesencriptado_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB1.SaveFile CD2.FileName, 1
End Sub
Private Sub Guardarencriptar_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB2.SaveFile CD2.FileName, 1
End Sub
Private Sub GuardarRTB1_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB1.SaveFile CD2.FileName, 1
End Sub
Private Sub GuardarRTB2_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB2.SaveFile CD2.FileName, 1
End Sub
Private Sub RTB1_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
```

```
If Button = 2 Then PopupMenu menuRTB1
End Sub
Private Sub RTB2_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
If Button = 2 Then PopupMenu menuRTB2
End Sub
Private Sub Salir_Click(Index As Integer)
End
End Sub
Private Sub StatusBar1_PanelClick(ByVal Panel As MSComctlLib.Panel)
End SubPrivate Sub Abrirencriptado_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB2.LoadFile CD1.FileName, 1
End Sub
Private Sub Abrirencriptar_Click(Index As Integer)
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB1.LoadFile CD1.FileName, 1
End Sub
Private Sub AbrirRTB1_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB1.LoadFile CD1.FileName, 1
End Sub
Private Sub AbrirRTB2_Click()
CD1.Filter = "Ficheros de Texto |*.txt"
CD1.ShowOpen
RTB2.LoadFile CD1.FileName, 1
```

```
End Sub
Private Sub Acerca_Click(Index As Integer)
MsgBox " Encriptor Version 1.0", 64, "Encriptor de Archivos de Texto"
End Sub
Private Sub EncriptorDesencriptar_Click()
MSComm1.CommPort = Text3.Text
MSComm1.PortOpen = True
MSComm1.Output = Chr(Val(Text1.Text) + 32)
MSComm1.Output = Chr(Len(RTB2.Text) + 32)
MSComm1.Output = RTB2.Text
MSComm1.Output = Chr(68)
MsgBox "Envio Completo", 64, "Encriptor De Texto"
RTB1.Text = MSComm1.Input
MSComm1.PortOpen = False
End Sub
Private Sub EncriptorEncriptar_Click()
MSComm1.CommPort = Text3.Text
MSComm1.PortOpen = True
MSComm1.Output = Chr(Val(Text1.Text) + 32)
MSComm1.Output = Chr(Len(RTB1.Text) + 32)
MSComm1.Output = RTB1.Text
MSComm1.Output = Chr(69)
MsgBox "Envio Completo", 64, "Encriptor De Texto"
RTB2.Text = MSComm1.Input
MSComm1.PortOpen = False
End Sub
Private Sub Form_Load()
MSComm1.Settings = "19200,n,8,1"
```

```
Contextual.Visible = False
End Sub
Private Sub Label6_Click()
End Sub
Private Sub GuardarDesencriptado_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB1.SaveFile CD2.FileName, 1
End Sub
Private Sub Guardarencriptar_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB2.SaveFile CD2.FileName, 1
End Sub
Private Sub GuardarRTB1_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB1.SaveFile CD2.FileName, 1
End Sub
Private Sub GuardarRTB2_Click()
CD2.Filter = "Ficheros de Texto |*.txt"
CD2.ShowSave
RTB2.SaveFile CD2.FileName, 1
End Sub
Private Sub RTB1_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
If Button = 2 Then PopupMenu menuRTB1
End Sub
Private Sub RTB2_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
```

```
If Button = 2 Then PopupMenu menuRTB2

End Sub

Private Sub Salir_Click(Index As Integer)

End

End Sub

Private Sub StatusBar1_PanelClick(ByVal Panel As MSComctlLib.Panel)

End Subvvv
```