**Maquina enigma**

**Motivo:**

Tras leer el libro: THE CODE BOOK (ISBN 978-0-385-73062-4) que habla sobre la criptografía, explica desde cómo se encriptaban las cartas hasta los métodos que se usaban para desencriptarlas, fue la maquina enigma la que revolucionó la criptografía puesto que podía encriptar la misma letra sin repetir otras, el funcionamiento de la maquina me pareció sencillo además de hacerlo un proyecto interesante

**Índice**:

**Conceptos** ………………………………………………….............………… Pág. (1-2)

**Materiales** ……………………………...…………………..………………… Pág. (2-3)

**Funcionamiento** **electrónico** ……………………………….………………... Pág. (3-4)

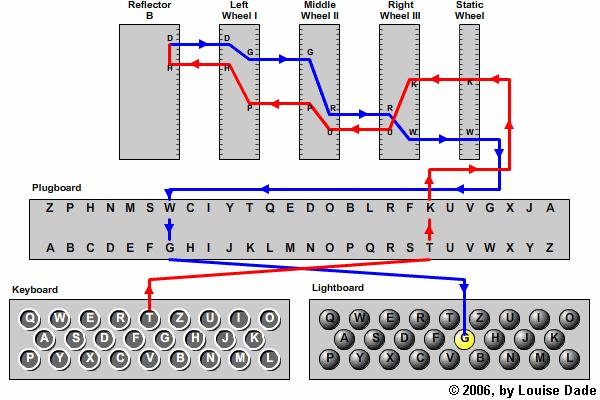
**Proceso de Ejecución** ……………………………………..…...…………...... Pág. (5-6)

**Conclusión** ………………………………………………….……………..…… Pág. (6)

**Archivos Externos** …………………………………………………………… Pág. (6-7)

**Conceptos:**

La máquina enigma es una maquina mecánica que gira los rotores al ser pulsada una tecla, dejando pasar la corriente eléctrica para que ilumine una bombilla, la versión más antigua que consiste de tres rotores que cambian sus conexiones al ser girados, luego pasa por un reflector, después pasa por los rotores en sentido opuesto y final por un panel de enchufes.

[](http://enigma.louisedade.co.uk/howitworks.html)

En el proyecto decidí usar tres rotores puesto que la pantalla de 8 segmentos solo tiene cuatro cifras además decidí eliminar el panel de enchufes puesto que no sabia como podría implementarlo y también investigué la forma de colocar un teclado en la máquina, pero lleva un gran coste del trabajo y complicación que eran innecesarias, al final terminé decantándome por colocar un modulo bluetooth (HC-06) y hacer una aplicación móvil a través de Android Studio. Puesto que ya había hecho una aplicación móvil similar para controlar la posición de un motor paso a paso de forma remota.

En el proyecto, el usuario introduce la letra a través de la aplicación bluetooth, el microcontrolador encripta la letra y renderiza en un matriz de leds de 8x8 e incrementa la posición de los rotores que se ven por la pantalla de 8 segmentos y para que el usuario tenga que desencriptar la letra tiene que colocar los rotores en la posición en la que estaba la letra y pulsar la letra que estaba encriptada, cuando la encripta devuelve la letra inicial desencriptada, es como realizar el recorrido de la anterior imagen pero en el otro sentido, así permite que dos personas puedan hablar con los rotores ya prestablecidos sin que una tercera persona pueda desencriptar el mensaje

**Materiales:**

* Arduino Uno
* Tabla perforada
* Led Matrix 8x8
* Pantalla de cuatro cifras de 8 segmentos
* Modulo bluetooth (Hc-06)
* Cables rígidos y flexibles
* Tabla de madera de 30x40cm
* Resistencias (12ud 330ohm, 10k ohm, 15kohm)
* 4ud Tornillos
* Goma elástica
* 2 Bisagras con sus tornillos y tuercas

Los materiales que he tenido que comprar para realizar el proyecto:

* Condensadores (2ud 22pF, 10uF, 1uF)
* Pila 9v
* DIP o Zócalos (28, 2ud 14, 4ud 16)
* Clip de pila de 9V
* Cristal oscilador 160000MHz
* Transformador de 5v L7805CV
* *Shift Registers* Ic74hc595
* Puerta lógica NOT Ic74hc04

Las herramientas necesarias

* 2 Sargentos
* Segueta
* Lija
* Pistola de silicona
* Alicates
* Pinzas de electricista
* Pelador de cables
* Soldador
* 3ª Mano
* Destornillador
* Ordenador del que programar

**Funcionamiento electrónico:**

Puesto que el Arduino Uno tiene 13 pines y los pines que necesito para hacer funcionar las cuatro cifras de 8 segmentos son 12(8+4) pines, mas los 16(8+8) pines del matriz de 8x8 y el módulo bluetooth que necesita 2(Tx, Rx) pines para hacer una conexión serial entre el Arduino y el módulo. Utilizo registros de desplazamiento (Shift Registers Ic74hc595) para controlar los leds, mientras conecto el módulo directamente al Arduino. Los registros de desplazamiento son útiles para poder aumentar la cantidad de pines del micro controlador, se pueden conectar en serie para aumentar mas su capacidad, son útiles para los elementos que no requieren de una comunicación compleja, utilizo dos para el matriz de 8x8 en el que uno esta en tierra mientras el otro suministra voltaje, pasando por resistores de 330 ohm sin que dañen los leds, para que el matriz se ilumine, como es un matriz es decir se iluminan 64 leds con 16 conexiones, no se pueden iluminar todos a la vez pero se pueden iluminar los leds deseados de una columna y pasar a la siguiente deprisa sin que el ojo humano perciba la diferencia, es este el fundamento con el que consigo bajar la cantidad de pines necesarios de 64 a 16 y luego a 3 para controlar los dos registros de desplazamiento, siendo ese el lado positivo de esa idea tiene el lado negativo que es mas complejo y si el procesador se atasca realizando una operación el usuario solo podrá ver una columna, es decir, que no podrá identificar la letra, los ordenadores usan *threads* para resolver este tipo de problemas donde reparte las tareas que hacen, durante un tiempo realizan un programa, por ejemplo calculaciones, lego paran y se cambian a realizar otras operaciones como por ejemplo refrescar la pantalla, lo hacen de forma instantánea que no se nota la diferencia pero las calculaciones no afectan a la pantalla. En el Arduino no es el caso, no he encontrado la posibilidad de hacer esto entonces si el usuario empieza a pulsar varias teclas deprisa, se puede notar que la intensidad de la pantalla baja puesto que tarda más en refrescar la pantalla, lo mismo también pasa con las cuatro cifras de 8 segmentos, que en el circuito necesito conectar 8 pines a tierra y para ello necesito una puerta NOT(Ic74hc04) que se conecta a los segmentos, decidiendo así que segmentos iluminar, los otros 4 pines pasan por resistencias de 330 ohm, iluminando las cifras, para poder obtener cifras diferentes, es decir controlar 32 leds con solo 12 pines, solo se le da voltaje a uno de los cuatro para poder decidir como iluminar la cifra, luego se pasa a la siguiente deprisa como en el caso anterior ese es el lado positivo pero también sin mas tenues cuando el microcontrolador se ralentiza, se puede apreciar que la cifra que esta a la derecha es la mas brillante puesto que se queda encendida mientras el procesador realiza otras tareas, el módulo bluetooth hc-06 necesita voltaje y tierra y dos pines para establecer una comunicación serial con el Arduino, el pin Rx del módulo tiene que estar conectado por una resistencia de 15k ohm al Arduino y otra resistencia de 10k ohm en paralelo a tierra para hacer un divisor de voltaje y bajar la tensión de 5V a 3.3V que es como funciona el módulo, existen varios fabricantes con sus variaciones haciendo que no sea totalmente necesaria esta parte.

En el circuito además uso el L7805CV un transformador que baja los 9V de la pila, que esta conecta al circuito a través de un clip, a los 5V con los que se puede trabajar con el Arduino y el resto de componentes, además esta conectado a un condensador de 10uF antes del transformador para asegurar una corriente estable y luego después del trasformador tengo otro condensador de 1uF lo que uso para bajar el voltaje, no es el circuito mas eficiente porque este baja el voltaje transformando el exceso en energía calorífica, calentándose, mientras hay otros que usan pulsos para limitar la corriente junto a condensadores, son mas complejos, pero no generan ni calor ni desperdician tanta energía, para la finalidad de este proyecto este circuito es suficientemente simple y seguro.

Para poder integrar el Arduino fuera de la placa en mi circuito, además del suministro de 5v necesito darle la señal del reloj, el cristal oscilador 160 000MHz es como un diapasón pero como es de cuarzo tiene unas propiedades que le permite seguir vibrando a la misma frecuencia sin ser alterada si se le aplica corriente eléctrica, ambos pines del cristal van conectados al Arduino y en paralelo un condensador de 22pF por cada pin para asegurar que el Arduino no tenga problemas a la hora de comunicarse con el módulo bluetooth y final mente como en mis circuitos en lugar de soldar los chips directamente en la tabla agujereada, sueldo un DIP y luego coloco por encima el chip así puedo quitar el chip sin desoldar nada y cuando se suelda, no hay que preocuparse de sobrecalentar el circuito integrado (IC). Los cables rígidos que utilicé eran simplemente un cable de ethernet que en su interior tiene 8 cables entrelazados entre si, es un proceso tedioso el desenredarlos.

**Proceso de Ejecución:**

Primero se diseña el circuito electrónico, y se comprueba en una *protoboard* y si funciona bien se sueldan los componentes, a la hora de soldarlos se introducen los componentes en la tabla perforada y se doblan los pines en la dirección deseada, luego se corta el exceso con unos alicates, después de aprietan los pines para juntar los componentes a la tabla y luego se termina saldando los componentes con la ayuda de pasta de soldar y una tercera mano que sujeta la tabla mientras las dos manos están ocupadas manteniendo el soldador y el estaño, una vez colocados todos los componentes se pelan, se miden, se cortan y se preparan los cables en la tabla perforada, apretándolos conrea ella, cortando los excesos y terminando soldándolos después de cada soldadura se limpia la punta del soldador para quitar restos de suciedad quemados, la mayor parte del tiempo se dedica a cortar y preparar la tabla antes de soldarla que el tiempo en soldadura.

Para cortar la tabla de madera se intenta buscar la máxima eficiencia sin desperdiciar material, hacer cortes no muy largos y que compartan lados para reducir el perímetro de corte, para realizar cortes en el interior de la madera de hace un agujero con el tornillo, luego se agranda con la punta del destornillador puesto que la madera es blanda y se hace pasar el filo de la segueta por dentro, se ajusta la segueta, se termina cortando por el interior todo eso con la ayuda de sargentos para estabilizar la pieza y al final se lija la pieza quitando las astillas.

Una vez que esta hecho el circuito eléctrico se pegan las dos pantallas con silicona y se colocan 4 tornillos alrededor de la tabla perforada en el que se coloca por encima una goma que mantenga el circuito en su sitio sin moverse, para colocar las bisagras, se realizan agujeros con un tornillo de carpintería, una vez echo el orificio se pasa el tornillo que sujetara las tuercas, se pasa la bisagra y luego la tuerca que aprieta la bisagra contra la madera, para apretar la tuerca se usan unas pinzas para aplicar la fuerza pero con el cuidado de no enterrar el tornillo dentro del a madera por el otro lado una vez listo los componentes eléctricos y la carcasa de madera, se procede a juntarlo todo como una pieza con la ayuda de la pistola de silicona y celo que mantiene la pieza en su lugar.

A la hora de programar la aplicación en Android use como punto de partida una aplicación similar que controlaba el motor paso a paso por bluetooth tan solo tenía que cambiar la interfaz de usuario ajustar el protocolo para que se comunicara con el Arduino, el proceso de programación se hizo en el IDE de Android Studio, como se programa una aplicación de verdad.

En el código que se ejecuta en el Arduino tuve que averiguar como hacer funcionar los componentes sin la ayuda de una librería externa, a excepción de la librería Serial que establece una conexión estándar con el Arduino, en el código están almacenadas las configuraciones de como el procesador renderiza las letras, como interpretar los comandos Bluetooth (BT), la configuración de los rotores y el reflector. A la hora de encriptar la letra, el procesador transforma la letra en el numero empezando por a = 0 y siguiendo en orden alfabético hasta acabar en z = 25, la configuración de los rotores es la mayor complejidad a la hora de programarlo, se puede visualizar que un rotor es como una función por ejemplo f(x) x=a ;f(a) = n la propiedad de los rotores que por cada letra(numero) tiene que dar otro que sea diferente y único es decir que cuando se encripte una letra el resultado final no será repetido por otra letra, y también se puede devolver la misma que fue introducida, los que se hace es aplicar el resultado del primero como entrada del segundo como si fuera así: h(g(f(x))) siendo f(x) el primer rotor, g(x) el segundo y h(x) el tercero luego la señal pasa por un reflector este lo que hace es unir fijamente los resultados de una letra con otra, así se consigue que cuando se pulse una letra salga el resultado de otra, si se pulsa esta otra sin mover los rotores da el resultado de la primera letra que hemos pulsado, quedaría así: r(h(g(f(x)))) sien do r(x) la transformación que hace el reflector, luego como vuelve en sentido inverso pasa por el tercer rotor luego el segundo y el primero, además como pasa por el otro sentido, hay que sacar la función inversa es decir la función que transforma el resultado de una función en la variable de su parámetro de entrada es decir si f(a) = n entonces f’(n) = a entonces el funcionamiento del la maquina enigma se puede representar así: f’(g’(h’(r(h(g(f(x))))))) sin llegar a tener en cuenta que los rotores están cambiando de posición. Entonces para programar los rotores uso *arrays* son como una variable que contiene una lista de números en este caso las 26 letras, no 27 con la ñ porque es impar y la maquina no podría tener un reflector para esta letra, estas listas de números tienen un *index* la localización del numero dentro del array entonces el índex es la entrada por donde la letra llega, además se le añade otra variable que incrementa y se le aplica modulo 26 que representa la rotación del rotor, para el resultado de la suma se le aplica el módulo de 26, también en lugar de aplicar módulos se podrían aplicar restas y sumas pero sería muy complejo, para el resultado de se aplica en el siguiente array más la posición del rotor y el módulo, de la misma forma funciona el reflector pero no tiene posición y es especial porque siempre cumple esta propiedad: f(a) = n y f(n) = a esto hace que la maquina enigma nunca de como resultado la letra que se desea encriptar

**Conclusión:**

Este no es un proyecto muy complejo, pero tampoco sencillo, lo bueno es que puedo aplicar los conocimientos que ya había cogido previamente como programar en Android, Arduino, el motor gráfico 3D que enseña la pieza, que programe años atrás, también habilidad de soldar circuitos y el diseño de las piezas mediante el uso de la trigonometría.

También trajo problemas interesantes como una conexión entre los cables que no estaba del todo buen echa que daba un efecto interesante en el matriz de 8x8, problemas en la soldadura, como conexiones que se unían con las de al lado, problemas a la hora de programar en Arduino e incluso problemas moviendo la aplicación apk al móvil debido a que los drivers eran antiguos, problemas en las matemáticas a la hora de intentar encontrar la medida, etc.

**Archivos Externos:**

La carpeta Img contiene imágenes sobre mis apuntes, el diseño de la pieza, sus medidas y como se colocaban en la tabla de cortar, también está adjuntado la EnigmaApp.apk (el instalador de la aplicación, solo funciona cuando se conecta por bluetooth), ModelView.jar Es el motor 3D que usa OpenGL para renderizar la pieza donde se usan las teclas Q,W,E,A,S y D para mover el objeto y la tecla R cambia su rotación, la F es más técnico, donde representa la primera vez que hice la pieza, no se renderizaban las piezas porque los normales de las caras de la pieza no apuntaban en la dirección correcta. En la carpeta Bender contiene la pieza 3D en la extensión de la aplicación que use para modelar la pieza, en esa aplicación se puede ver de forma más detallada toda la información sobre la pieza, que en papel no necesite averiguar. Dentro de la carpeta de Development están imágenes que use como referencia para hacer las letras y EnigmaCodeButtons.ino está el código donde experimente con los botones que estaban conectados a un Shift Register que iba alternando por cada botón para averiguar cual estaba presionado. Por último, la ficha de Excel con los costes de los materiales que compre recientemente, el resto me los regalaron o ya los tenía desde hace años