

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD TÉCNICA  
CARRERA ELECTRONICA Y TELECOMUNICACIONES**



**EXAMEN DE GRADO  
TRABAJO DE APLICACIÓN  
NIVEL LICENCIATURA**

**“DISEÑO E IMPLEMENTACION DE UN CIRCUITO GENERADOR DE  
CARACTERES ROTATIVO DE LEDS DE PERSISTENCIA VISUAL”**

**POSTULANTE: ROBERTO YASIR PATTY GALLEGOS**

**AGOSTO DE 2012**

**LA PAZ - BOLIVIA**

## ÍNDICE

	Pág.
RESUMEN DEL TRABAJO .....	1
1. GENERALIDADES .....	2
1.1. Planteamiento del problema .....	2
1.2. Justificación del trabajo .....	2
1.2.1 Justificación Técnica.....	2
1.2.2 Justificación Económica .....	3
1.3. Objetivo General .....	3
1.4. Objetivos Específicos .....	3
CAPITULO I	
1. Fundamento Teórico .....	4
1.1. Sistema Visual Humano .....	4
1.2. Persistencia Retiniana .....	6
1.3. Microcontroladores .....	12
1.3.1 Introducción a los Microcontroladores .....	12
1.3.2 Diferencia entre Microcontrolador y Microprocesador.....	12
1.3.3 Microcontrolador PIC 18F4550.....	13
1.3.4 Controlador USB .....	16
1.4. Motor DC Cebek modelo C6044 .....	18
1.5. Led de Alta Luminosidad .....	20
CAPITULO II	
2. Ingeniería del Proyecto .....	21
2.1. Desarrollo del Trabajo .....	21
2.2. Circuito de Control de Leds .....	21
2.3. Sistema de Selección de Caracteres .....	41
2.4. Rotación del Circuito .....	51
CAPITULO III	
3. Costos .....	54
4. Conclusiones .....	55
5. Recomendaciones .....	55
6. Bibliografía .....	56
7. Anexos .....	57

## ÍNDICE DE FIGURAS

	Pág.
Figura 1.1. Corte transversal del ojo humano .....	4
Figura 1.2. Distribución de los conos y bastones .....	5
Figura 1.3. Respuesta al espectro del ojo humano.....	6
Figura 1.4. Fenakistocopio .....	7
Figura 1.5. Zoótrofo.....	7
Figura 1.6. Traumátropo.....	8
Figura 1.7. Praxinoscopio.....	8
Figura 1.8. Cine NIC .....	9
Figura 1.9. Lámina de dibujos del cine NIC.....	9
Figura 1.10. Esquema de funcionamiento del disco de Nipkow.....	10
Figura 1.11. Disco de Nipkow .....	10
Figura 1.12. Dispositivo de LEDs .....	11
Figura 1.13. Cruz de LEDs de una farmacia.....	11
Figura 1.14. Panel informativo del tren.....	11
Figura 1.15. PIC18F4550 .....	13
Figura 1.16. Pines del PIC18F4550 .....	14
Figura 1.17. Diagrama de bloques PIC18F4550 .....	15
Figura 1.18. Conexión USB PIC18F4550.....	17
Figura 1.19. Motor de continua C6044 .....	18
Figura 1.20. Corte transversal de un motor DC .....	19
Figura 1.21. Led de alta Luminosidad .....	20
Figura 2.1. Diagrama de Bloques General del Sistema.....	21
Figura 2.2. Esquema de una matriz de leds .....	22
Figura 2.3. Lectura del osciloscopio de la frecuencia de giro .....	23
Figura 2.4. Letra E .....	24
Figura 2.5. Letra P .....	24
Figura 2.6. Compilador MIKRO C.....	25
Figura 2.7. USB conectado y esperando datos .....	38
Figura 2.8. USB conectado y esperando datos .....	39
Figura 2.9. Plantilla para la comunicación por USB.....	40
Figura 2.10. USB desconectado.....	48
Figura 2.11. USB conectado .....	49
Figura 2.12. Prototipo de la placa de anillos.....	50
Figura 2.13. Placa de anillos .....	51
Figura 2.14. Placa de anillos .....	51
Figura 2.15. Maderas sin pintar.....	52
Figura 2.16. Base con motor DC parte frontal .....	52

## ÍNDICE DE TABLAS

	Pág.
Tabla 3.1. Presupuesto Proyecto .....	53

## DEDICATORIA

El presente trabajo de aplicación va dedicado a mi madre Teresa Gallegos y a mi padre Fidel Patty, a quienes amo profundamente y siempre me brindaron todo su cariño, apoyo y paciencia para seguir adelante.

A mi hermana Lizeth y mis sobrinos Marcos y Keolany por su comprensión y paciencia.

A mi tía (mamá) Sandra y mi abuela Yolanda Lenz Vda de Gallegos por su constante apoyo.

## **AGRADECIMIENTOS**

Quiero agradecer al Licenciado Nixon Vargas por toda su colaboración y sus consejos brindados en todo el proceso de titulación.

A el Lic. Richard Márquez, Ing, Ramiro Velarde, Ing. Franklin Rada, Lic. Javier, Yujra, por haberme transmitido sus conocimientos a lo largo de mi vida universitaria y todo el tiempo dedicado en la supervisión de este documento.

A mis compañeros y amigos incondicionales de la Facultad Técnica Lic. Carlos Nina, Lic. Wilfredo Mayta, Lic. Ivan Quisbert, Rafael Chuquimia, Lucas Collanque, por su valiosa colaboración para la realización de este proyecto.

# **“DISEÑO E IMPLEMENTACION DE UN CIRCUITO GENERADOR DE CARACTERES ROTATIVO DE PERSISTENCIA VISUAL”**

## **RESUMEN DEL TRABAJO**

En el presente trabajo de aplicación, se desarrolla el prototipo de un circuito generador de caracteres rotativo que aprovecha un fenómeno conocido como la “persistencia visual o retiniana” para provocar el efecto de que dichos caracteres son escritos en el aire.

En el capítulo I, se explica el sistema visual humano y los efectos de la persistencia retiniana el cual es el fundamento de este trabajo de aplicación. También se identifica a los principales elementos que se utilizan en el proyecto, como son el microcontrolador PIC 18F4550, el motor DC y los diodos led de alta luminosidad.

En el capítulo II, se explica el desarrollo del trabajo, que se divide en tres partes concretas: circuito de control de leds, sistema de selección de caracteres y rotación del circuito. En las cuales se presenta una detallada descripción del código de programa del microcontrolador para obtener una secuencia determinada de encendido y apagado de los leds para representar un carácter específico, con la opción de seleccionar un mensaje diferente desde un ordenador por medio de Visual Basic 6.0 a través de el puerto USB, en el mismo instante en que el circuito posea un movimiento giratorio provocado por un motor DC. Además se presentará una detalla aclaración del montaje de la estructura de soporte y de la fijación de la placa.

En el capítulo III, se hace referencia a los costos monetarios del proyecto, a las conclusiones y recomendaciones realizadas en el diseño e implementación del trabajo de aplicación.

## **1. GENERALIDADES**

### **1.1. PLANTEAMIENTO DEL PROBLEMA**

En la actualidad los diseños de tableros generadores de caracteres construidos en base a leds no cambiaron mucho en los últimos años y se presentan con similares características que requieren de un óptimo mantenimiento para su funcionamiento, además ocupan mucho espacio.

La situación descrita define un problema que necesita ser atendido desde un punto de vista técnico con la finalidad de proporcionar una solución alternativa a la problemática detectada, para cuyo efecto se formula la siguiente interrogante que se constituye en el problema de desarrollo: ¿Cómo diseñar un circuito generador de caracteres que aproveche el fenómeno de la persistencia visual con un movimiento circular provocado por un motor DC paso a paso controlado por el microcontrolador PIC18F4550?

### **1.2 JUSTIFICACION DEL TRABAJO**

#### **1.2.1 JUSTIFICACION TECNICA**

El diseño de un circuito generador de caracteres con movimiento rotativo permite manipular con más facilidad cadenas de símbolos o letras visualizadas a través de leds.

El operador podrá desplegar cualquier cantidad de mensajes limitados tan solo por la capacidad de memoria del microcontrolador y de su imaginación, utilizando solamente una fila de leds.

## **1.2.2 JUSTIFICACION ECONOMICA**

Los tableros de leds actuales son muy costosos y necesitan de un mantenimiento continuo. En el trabajo de aplicación que se presenta a continuación se reducen los costos puesto que, se utilizara una menor cantidad de componentes electrónicos y con un resultado similar al de los tableros de leds estandar.

Se pretende realizar el trabajo de aplicación usando elementos sencillos y de bajo coste, para demostrar así que se pueden realizar aplicaciones muy interesantes con leds y que no representan grandes inversiones.

## **1.3 OBJETIVO GENERAL**

Diseñar un circuito generador de caracteres rotativo de leds, que aproveche el fenómeno de la persistencia visual.

## **1.4 OBJETIVOS ESPECIFICOS**

- Realizar un programa en MikroBasic Pro y Visual Basic para establecer la comunicación serial con el ordenador por el puerto USB y desarrollar un modelo y una simulación virtual del circuito utilizando el software PROTEUS.
- Idear un sistema de alimentación para la placa tomando en cuenta que dicha placa funcionará en movimiento circular.
- Optimizar el espacio de circuitería en la placa ya que ésta debe simular a una hélice y distribuir su peso para evitar vibraciones por desequilibrio.
- Sincronizar la velocidad de giro con el programa del microcontrolador para visualizar correctamente la imagen que proporcionaran los diodos led.
- Evaluar los costos y el desarrollo del trabajo demostrando su viabilidad.



# CAPITULO I

## 1. FUNDAMENTO TEORICO

### 1.1 SISTEMA VISUAL HUMANO

Los órganos de la visión en los seres humanos y en los animales son los ojos. Éstos varían entre las diferentes especies, desde las estructuras más sencillas, capaces de diferenciar sólo entre la luz y la oscuridad, hasta los órganos complejos que tienen los seres humanos y otros mamíferos, que pueden diferenciar variaciones muy pequeñas de forma, color, luminosidad y distancia.

De hecho, el órgano que realiza el proceso de la visión es el cerebro; la función del ojo es traducir las vibraciones electromagnéticas de la luz en un determinado tipo de impulsos nerviosos que se transmiten al cerebro. La parte trasera del globo ocular está recubierta casi totalmente por una membrana de células sensibles a la luz. Esta capa fotosensible se denomina retina, donde se detectan los estímulos visuales. La luz se filtra en el ojo a través de la córnea y es enfocada por el cristalino (una lente adaptable) en la retina. El iris de la pupila puede adaptar su tamaño para dejar pasar más o menos luz, según su intensidad.

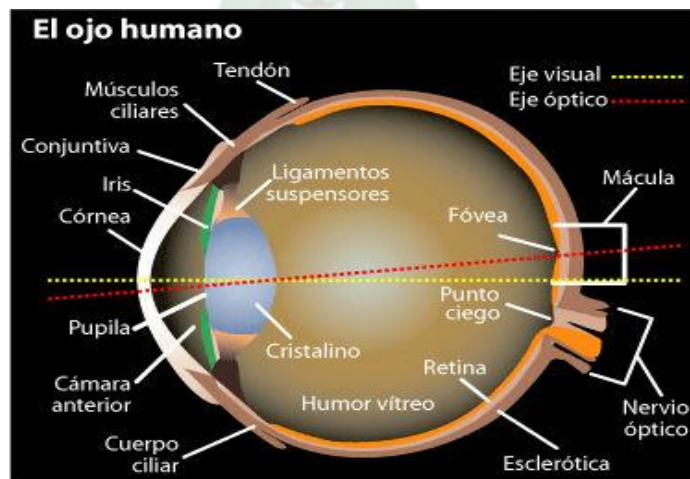


Figura 1.1 Corte transversal del ojo humano [1]

Volviendo de nuevo a la retina explicaremos que es una superficie que contiene diferentes clases de células. Entre ellas destacaremos las células fotosensibles llamadas conos y bastones, localizadas en la capa más profunda de la retina. Los conos que son las células receptoras de luz y los bastones las que emiten señales a otras células. Los conos se concentran cerca del centro de la retina, alrededor de 4 millones, son muy sensibles a los colores en cambio poco sensibles al brillo. Por otro lado los bastoncillos están muy repartidos dentro de la retina aproximadamente 100 millones. Son muchos más sensibles al brillo que los conos y son los responsables de la visión nocturna.

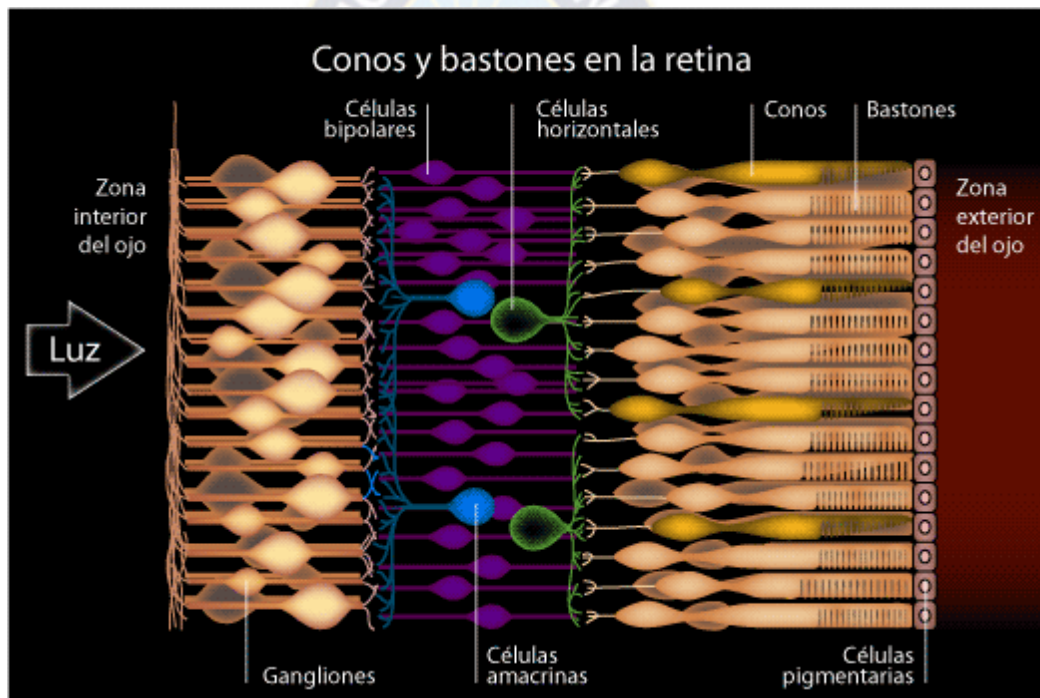


Figura 1.2 Distribución de los conos y bastones [2]

En la siguiente imagen se muestra la respuesta del espectro a la sensibilidad del ojo humano, si nos fijamos el color que mejor se detecta es el verde, por este hecho en el montaje se dejarán fijados LEDs smd de dicho color. Aunque se realizaran pruebas con el color azul para ver la diferencia de sensibilidad en el ojo.

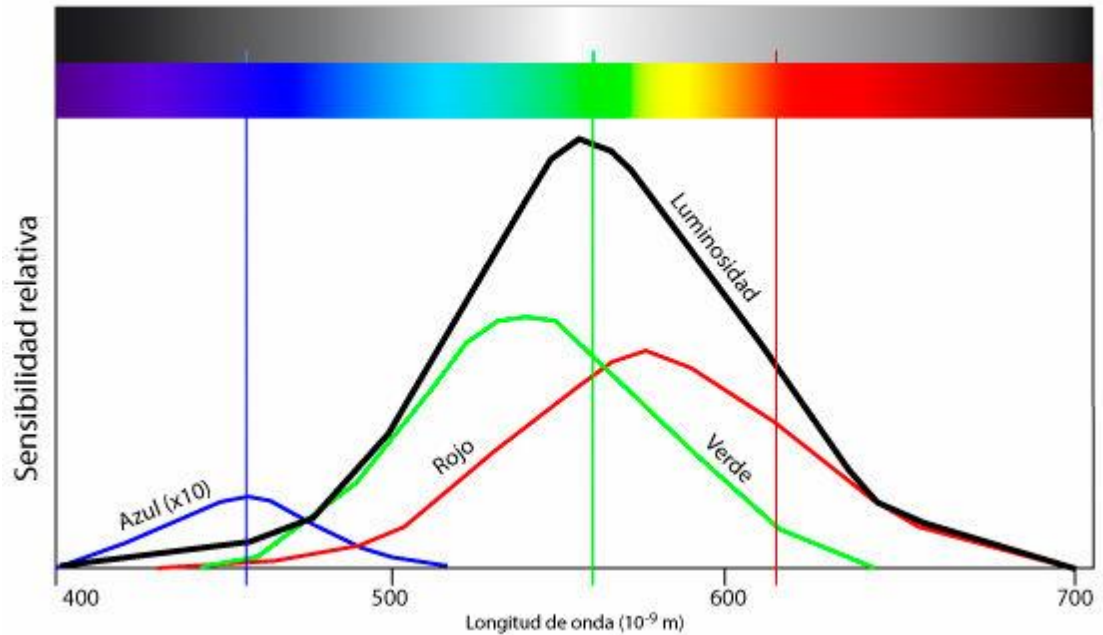


Figura 1.3 Respuesta al espectro del ojo humano [3]

## 1.2 PERSISTENCIA RETINIANA

Este fenómeno visual descubierto por el científico belga Joseph-Antoine Ferdinand Plateau es conocido como POV (Persistence Of Vision) o persistencia retiniana. Este hecho demuestra cómo una imagen permanece en la retina humana una décima de segundo antes de desaparecer completamente, lo que permite que veamos la realidad como una secuencia de imágenes que el cerebro "enlaza" como una sola imagen visual móvil y continua. En el cine en blanco y negro, se pasaban 18 imágenes por segundo porque se decía que era el mínimo para poder crear el efecto de movimiento, las películas actuales se proyectan a imágenes por segundo, e incluso a 25 para adaptarse mejor a la frecuencia utilizada en televisión. La persistencia retiniana es una característica de nuestro ojo que provoca que las imágenes que se observan no se borren instantáneamente. Este hecho, hace que las imágenes que se visualizan, queden guardadas por un instante en el cerebro. Por ejemplo, al hacer girar una cerilla podemos lograr que ésta parezca formar un círculo de fuego en el aire, al igual que si se hace con un LED encendido. Si a dicho LED o a un conjunto de ellos los

activamos de manera apropiada en tiempo y forma, obtendremos, gracias a la persistencia de nuestra visión, imágenes que parecen estar volando en el aire.

A continuación se detallará el funcionamiento de algunos juguetes ópticos comentados anteriormente:

El Fenakistocopio de Plateau es un disco giratorio que produce, al mirar por una rendija las sucesivas imágenes creando la sensación de que estas forman una secuencia en movimiento.

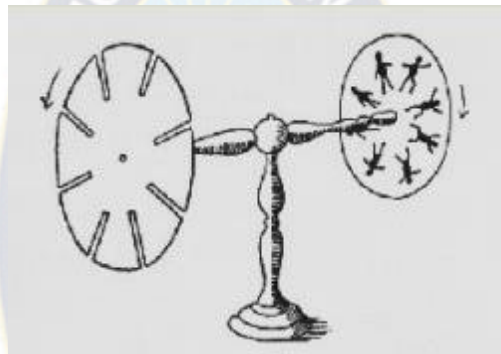


Figura 1.4 Fenaquistiscopio [4]

En la figura 5 se puede ver un ejemplar del zootropo, fue la versión mejorada del Fenaquistiscopio, con varias de las cartulinas usadas para ver las imágenes en movimiento, como se observa simplemente son el mismo dibujo con algún pequeño cambio para crear la sensación de un movimiento continuo.



Figura 1.5 Zoótrofo [4]

El taumátropo como se ha comentado, es uno de los juegos más antiguos basado en un disco de cartón que se hace girar sobre su eje mediante cuerdas atadas a sus extremos. En cada cara hay un dibujo distinto y, al girar el disco, las dos imágenes parecen fundirse.

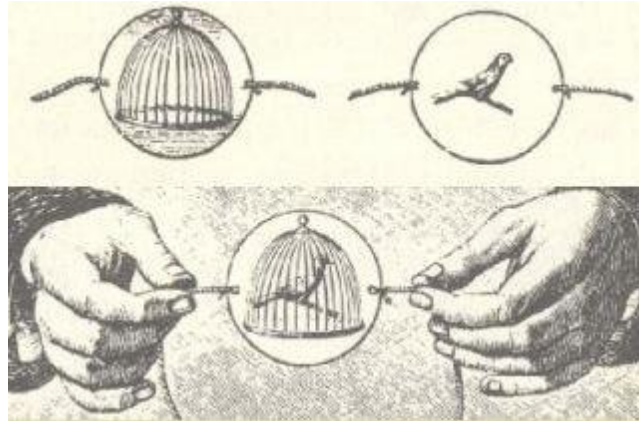


Figura 1.6 Traumátropo [5]

El Praxinoscopio de Reynaud: en el círculo exterior se sitúan las distintas "fases" de una secuencia en movimiento y al girar dicho círculo alrededor del eje, se refleja en cada cara de espejo del prisma central lo que parece una secuencia animada.



Figura 1.7 Praxinoscopio [6]

Un juguete más moderno: el cine NIC. Este pequeño proyector fue patentado en 1931 por los hermanos Nicolau Griñó y se calcula que entre esa fecha y 1974 se produjeron casi diez millones de reproductores. El cine NIC se basaba en dos bandas horizontales (con pequeñas variaciones una respecto de la otra) de manera que, al girar la manivela, la película se movía en vertical alternando las dos bandas y dando sensación de imagen animada.



Figura 1.8 Cine NIC [7]



Figura 1.9 Lámina de dibujos del cine NIC [7]

En 1882 el ingeniero Paul Nipkow patentó un sistema de barrido de imagen consistente en un disco al que se le habían practicado unos agujeros estratégicamente colocados para recorrer toda la imagen. Estos pequeños agujeros se disponían en forma de espiral. Si el disco se hacía girar entre una fuente de luz y un objeto, todo ese objeto era explorado por la luz a través de los agujeros después de una revolución del disco. Las porciones más oscuras y las más claras del objeto reflejaban diferentes cantidades de luz y, si esta luz reflejada se dirigía sobre una sucesión de células de selenio, las células enviaban una serie de impulsos eléctricos a través de un alambre.

En el extremo receptor, los impulsos eléctricos se usaban para producir señales que correspondían a la brillantez de las partes individuales del objeto iluminado en el extremo transmisor tal como se puede observar en la figura 10. Si las

señales, como por ejemplo el encendido de elementos individuales en un mosaico de luces, se veían a través de un segundo disco de Nipkow, el ojo recibía momentáneamente una serie de imágenes transitorias que correspondían a las partes del objeto que se exploraba. Si la rueda se hacía girar lo suficientemente rápido, el ojo transmitía estas partes a la mente como si fueran una sola imagen.

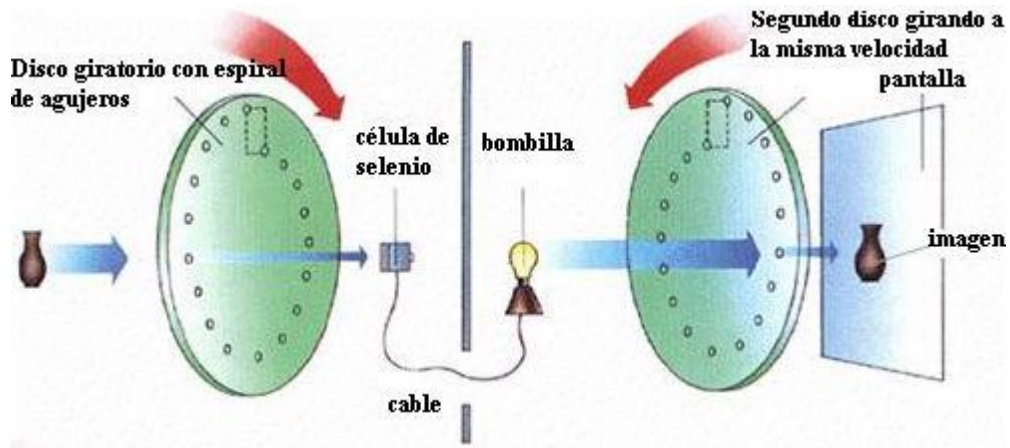


Figura 1.10 Esquema de funcionamiento del disco de Nipkow [5]

Este sistema tenía un fallo, los motores del disco emisor y del receptor tenían que estar sincronizados, ya que si no habían problemas de alineado de imagen.

Aún y con estos errores, este sistema se puso en práctica en 1923 (ver figura X) y estuvo en funcionamiento con emisiones regulares de la BBC (UK) hasta que surgieron los televisores de rayos catódicos.

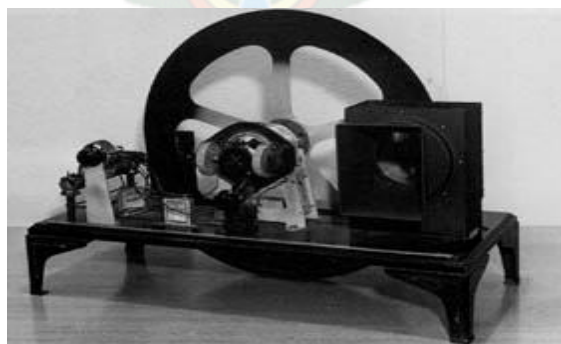


Figura 1.11 Disco de Nipkow [5]

Actualmente, en el mercado de los juguetes y/o gadgets es muy común encontrar relojes, paneles de saludos o bolas de publicidad que sean mensajes transmisibles mediante la palabra o las imágenes. Este tipo de productos basan su principio en los scroll-texts o pasa-mensajes que podemos ver en el supermercado, en las entradas del metro, en el autobús, etc. En el caso de los paneles informativos, se genera una línea (o columna) de leds, que mediante una sincronización de encender y apagar dichos leds y su traslación rápida y continua, van formando el mensaje que vemos y leemos.

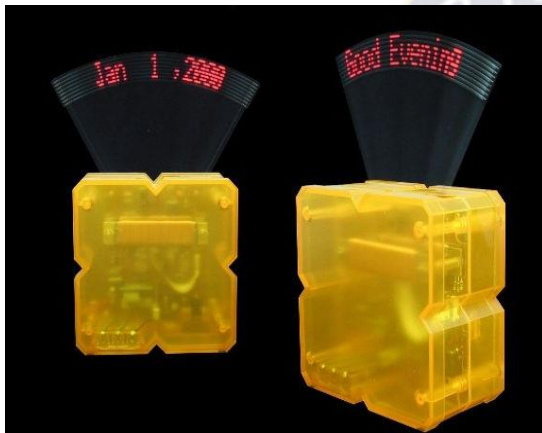


Figura 1.12 Dispositivo de LEDs [10]



Figura 1.13 Cruz de LEDs de una farmacia [10]



Figura 1.14 Panel informativo del tren [10]



## **1.3 MICROCONTROLADORES**

### **1.3.1 INTRODUCCIÓN A LOS MICROCONTROLADORES**

Se llama microcontrolador a un circuito integrado que incluye en su interior las tres unidades principales de un ordenador: Memoria, CPU y Periféricos de E/S. Un microcontrolador es capaz de realizar procesos lógicos. Dichos procesos o acciones son programables en el lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador. Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o CPU.
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de Entrada/Salida para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

### **1.3.2 DIFERENCIA ENTRE MICROCONTROLADOR Y MICROPROCESADOR**

Un error común es confundir los términos de microcontrolador y microprocesador. Un microcontrolador es un sistema completo, con unas prestaciones limitadas que no pueden modificarse y que puede llevar a cabo las tareas para las que ha sido programado de forma autónoma. Un microprocesador, en cambio, es simplemente un componente que constituye el microcontrolador, que lleva a cabo ciertas tareas y que, en conjunto con otros componentes, forman un microcontrolador.

En definitiva el primero es un sistema autónomo e independiente, mientras que el segundo es una parte, cabe decir esencial, que forma parte de un sistema mayor.

### 1.3.3 MICROCONTROLADOR PIC 18F4550

El PIC 18f4550, pertenece a los microcontroladores PIC18 de gama alta. Posee una arquitectura RISC (Reduced Instruction Set Computer) de 16 bits longitud de instrucciones y 8 bits de datos.



Figura 1.15 PIC18F4550 [11]

El microcontrolador PIC18F4550 tiene muchas características básicas que lo hacen apropiado para ser utilizado en nuestro diseño.

#### Características fundamentales:

- 77 tipos instrucciones.
- Compatibilidad con el código fuente de un PIC16F.
- Memoria de Programa para direccionamiento hasta 2 MB.
- Memoria de Datos para direccionamiento hasta 4 kB.
- Operable de DC a 40MHz.
- 8X8 Multiplicador de Hardware.
- Niveles de interrupción con prioridad
- Instrucciones de *16-bit-de ancho* y ruta de data de *8-bit-de ancho*
- Dos Temporizadores de 8 bits y dos *contadores* de 8 bits.
- Tres *Temporizadores* de 16 bits y Tres contadores de 16 bits.

- Cuatro Interruptores Externos.
- Alta corriente (25mA) para capacidad de la fuente.
- Convertidor Rápido de Análogo a Digital de 10 bits.
- Módulo de Detección Programable de bajo voltaje (LVD).
- Inicialización al ser encendido "Power on Reset" (POR).
- Temporizador de encendido "Power-up timer" (PWRT).
- Oscilador con Temporizador para iniciar (OST).
- Programación de Instrucciones en Lenguaje C.

### Diagrama de conexiones del PIC18F4550

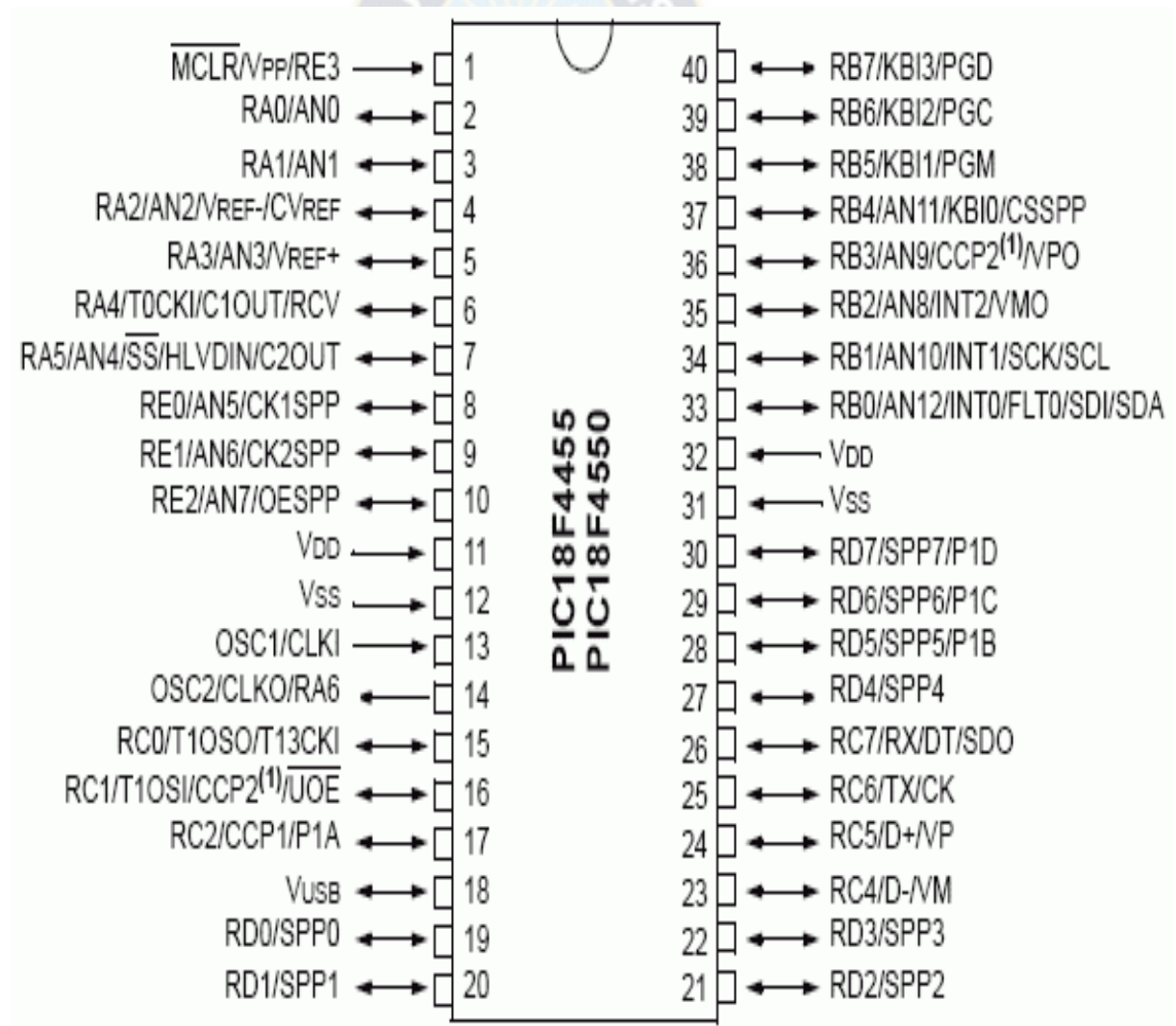


Figura 1.16 Pines del PIC18F4550 [11]

## Diagrama de bloques:

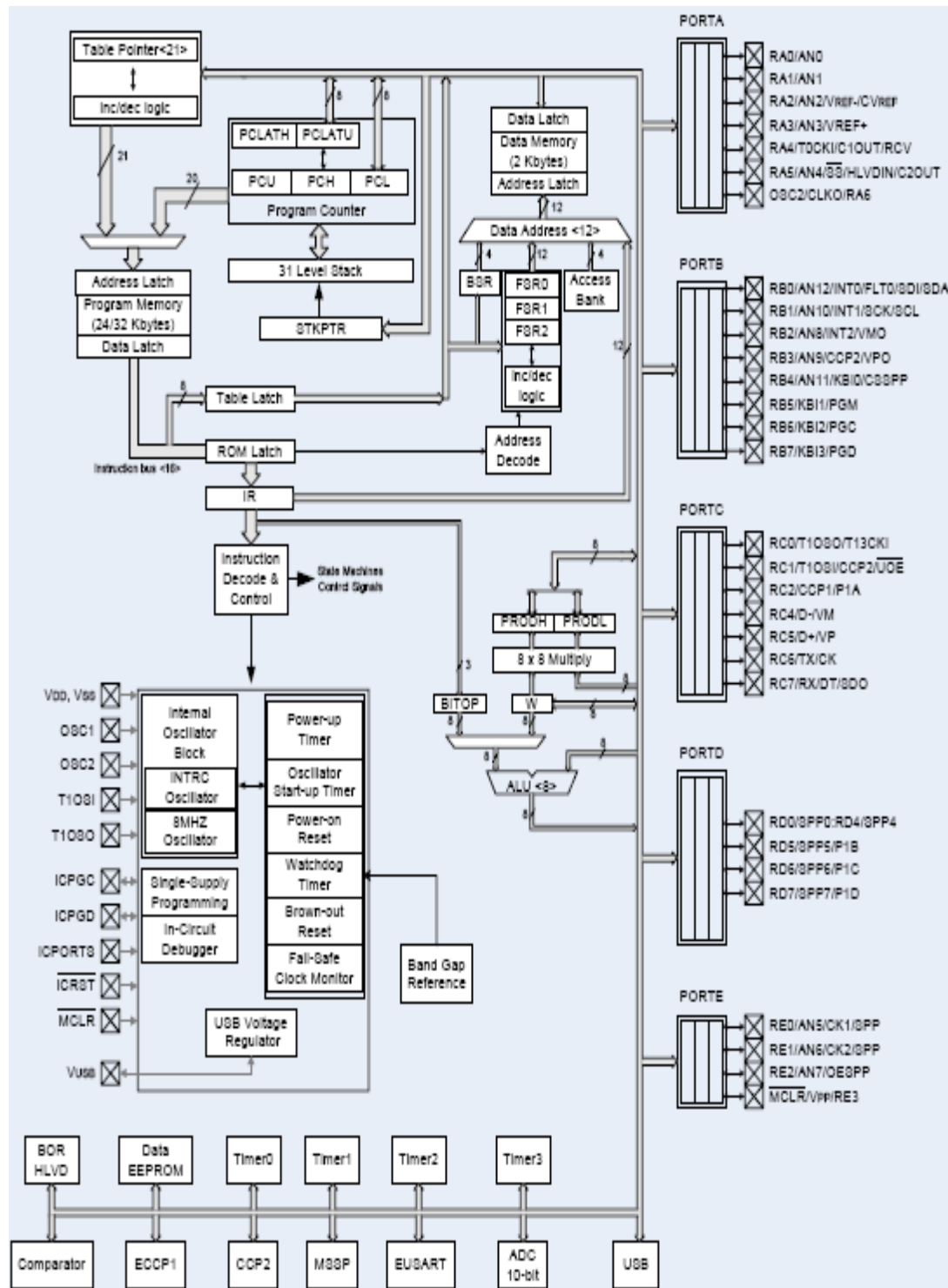


Figura 1.17 Diagrama de bloques PIC18F4550 [11]

## Memoria de configuración

Se trata de un bloque de memoria situado a partir de la posición 30000H de memoria de programa (mas alla de la zona de memoria de propagación de usuario).

En esta memoria de configuración se incluyen:

Bits de configuración: contenidos en 12 bytes de memoria flash permiten la configuración de algunas opciones del uC como:

- Opciones del Oscilador
- Opciones de reset
- Opciones del watchdog
- Opciones de la circuitería de depuración y programación
- Opciones de protección contra escritura de memoria de programa y memoria EEPROM de datos.

Estos bits se configuran generalmente durante la programación de uC, aunque también pueden ser leídos y modificados durante la ejecución del programa.

Registros de identificación: se trata de dos registros situados en las direcciones 3FFFEH y 3FFFFH que contienen información del modelo y revisión del dispositivo. Son registros de solo lectura y no pueden ser modificados por el usuario.

### 1.3.4 CONTROLADOR USB

El PIC18F4550 contiene un controlador USB que permite la comunicación entre un host USB y el PIC. Usa un transceiver5 USB interno y un regulador de 3.3 V

(para polarización independiente mediante el pin VUSB), debido a que la especificación USB requiere este voltaje; permitiendo así que el resto de aplicaciones del microcontrolador se puedan desarrollar con los 5 V necesarios.

Un requisito para transmitir a full-speed por el puerto USB6 es trabajar a 48 MHz, esta frecuencia se obtiene del pll7 interno del microcontrolador y el cristal de 20MHz externo utilizado.

La conexión del microcontrolador con la interfaz USB para su funcionamiento se indica en la Figura 1.18.

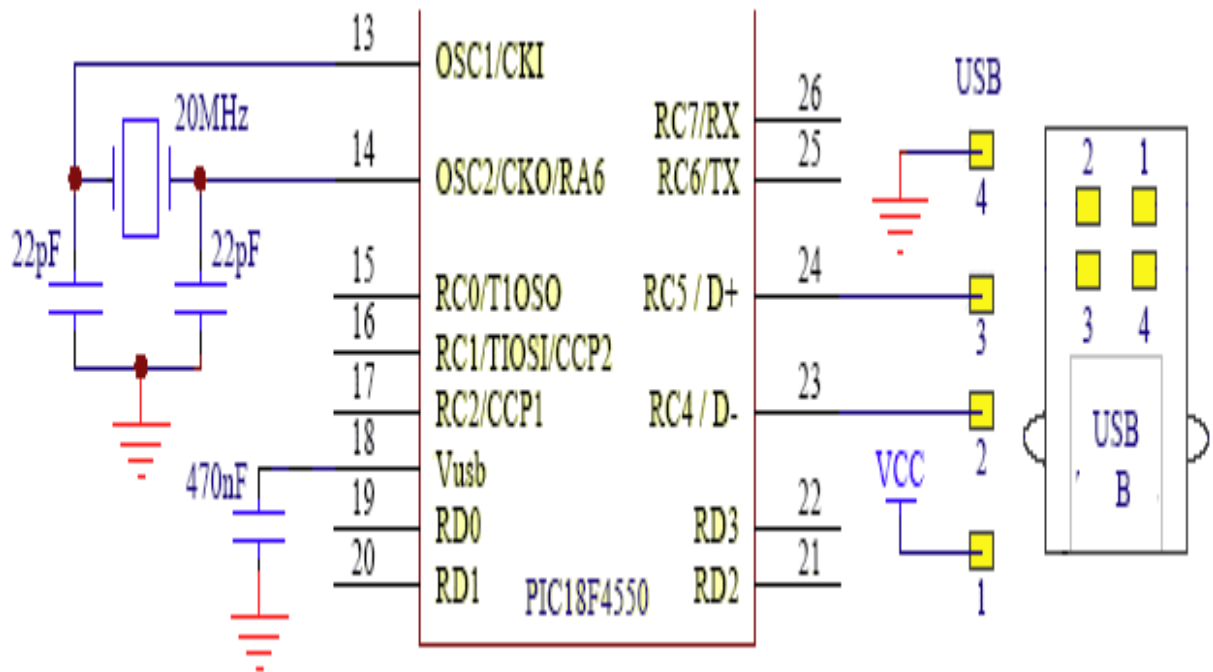


Figura 1.18 Conexión USB PIC18F4550 [12]

#### 1.4 MOTOR DC CEBEK modelo C6044

Se trata de un motor de continua, este tipo de motores son muy habituales en aplicaciones de baja potencia, concretamente uno de 20W de potencia y con un rango de trabajo de 6 a 12V de la casa CEBEK modelo C6044.



Figura 1.19 Motor de continua C6044 [13]

Poner en marcha un motor DC es muy sencillo, solo es necesario aplicar la tensión de alimentación entre sus bornes. Para invertir el sentido de giro basta con invertir la alimentación y el motor comenzará a girar en sentido contrario.

A diferencia de los motores paso a paso y los servomecanismos, los motores DC no pueden ser posicionados y/o enclavados en una posición específica. Estos simplemente giran a la máxima velocidad y en el sentido que la alimentación aplicada se los permite.

El motor de corriente continua está compuesto por las diferentes partes (obsérvese figura 1.20):

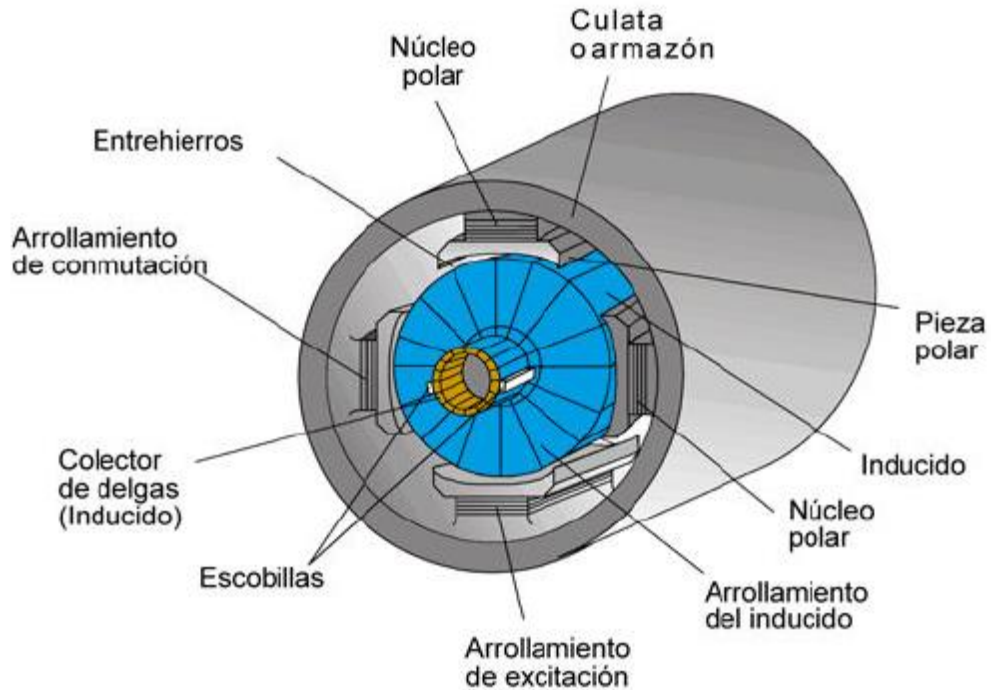


Figura 1.20 Corte transversal de un motor DC [13]

- **Inductor o estator (Arrollamiento de excitación):** Es un electroimán formado por un número par de polos. Las bobinas que los arrollan son las responsables de producir el campo inductor al circular por ellas la corriente de excitación.
- **Inducido o rotor (Arrollamiento de inducido):** Es una pieza giratoria formada por un núcleo magnético alrededor del cual va el devanado de inducido, sobre el que actúa el campo magnético.
- **Colector de delgas:** Es un anillo de láminas de cobre llamadas delgas, dispuesto sobre el eje del rotor que sirve para conectar las bobinas del inducido con el circuito exterior a través de las escobillas.
- **Escobillas:** Son unas piezas de grafito que se colocan sobre el colector de delgas, permitiendo la unión eléctrica de las delgas con los bornes de conexión del inducido. Al girar el rotor, las escobillas van rozando con las delgas, conectando la bobina de inducido correspondiente a cada par de delgas con el circuito exterior.



## 1.5 LED DE ALTA LUMINOSIDAD

El LED SMD cuenta con ventajas en su construcción, ya que el encapsulado permite una mayor superficie del semiconductor, ganando en la cantidad y calidad de luz que puede emitir el dispositivo por varias veces en comparación con su antecesor. Otra ventaja es el encapsulado automatizado que provee un bajo perfil de apenas entre 2 y 4 milímetros ya ensamblado como se puede observar en la imagen 30.

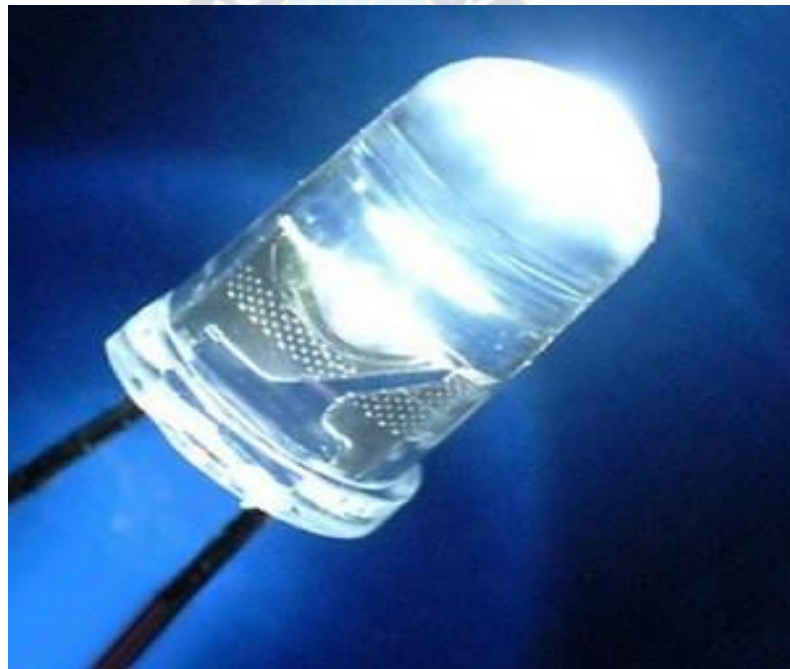


Figura 1.21 Led de alta Luminosidad [14]

## CAPITULO II

### 2. INGENIERIA DEL PROYECTO

#### 2.1 DESARROLLO DEL TRABAJO

En el presente trabajo de aplicación se desea, diseñar, implementar y controlar un circuito generador de caracteres rotativo de leds, que se basa en un mecanismo giratorio que consta básicamente de un motor al cual va fijada una placa electrónica que hace la función de hélice como si de un ventilador se tratase. En el extremo de esta placa va una hilera vertical con un número variable de leds. A través de la velocidad y de una adecuada programación del microcontrolador PIC 18F4550 que domina a los leds, se aprovecha de la persistencia de imagen en la retina para generar la ilusión óptica que queda representada en una imagen, que podrá ser un dibujo o bien palabras.

Para cumplir con este objetivo, el trabajo de aplicación se divide en 3 partes, las cuales se detallan a continuación:

- Circuito de control de leds
- Sistema de selección de caracteres
- Rotación del circuito

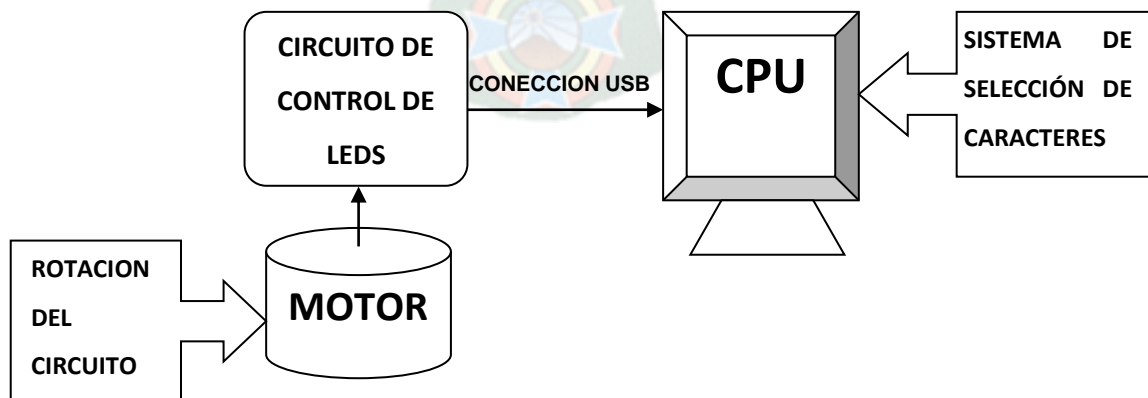


Figura 2.1 Diagrama de Bloques general del Sistema

## 2.2 CIRCUITO DE CONTROL DE LEDS

El circuito de control de leds define al dispositivo principal del trabajo de aplicación basado en el microcontrolador, el cual maneja el encendido y el apagado de los diodos led en secuencias suficientemente exactas para provocar el efecto de persistencia visual retiniana de los ojos.

Para lograr este propósito, se debe realizar un código de programa, que en nuestro caso será compilado en el software MikroC. Pero antes de ponerse a implementar el código, se ha de estar seguro de lo que se quiere hacer y cómo se va a hacer. Para ello, se va a explicar primero como se van a formar las imágenes para así entender mejor su programación.

Lo que se va a desarrollar es una matriz de píxeles, aprovechando nuestros leds (8 píxeles verticales) y el giro del eje. Lo que se hace es dividir el texto en una columna de un pixel de ancho y 8 filas de altura. Con los 8 píxeles de altura ya podemos enviarlos directamente a la columna de leds.



1	
2	
3	
4	
5	
6	
7	
8	

Figura 2.2 Esquema de una matriz de leds

Para hacerse una idea de su metodología imaginemos que se encienden los leds durante una fracción de tiempo y luego se apagan. Seguidamente nos movemos una columna a la derecha y volvemos a crear nuestro conjunto de 8 píxeles verticales.

Volvemos a encender los leds durante otra corta fracción de tiempo y volvemos a apagarlos. En el intervalo entre las dos columnas, el eje ha seguido girando y, por tanto, ha avanzado unos milímetros. Con esto, al encender la 2ª columna, nos encontramos con que está desplazada con respecto a la anterior. La persistencia de imagen en la retina provoca que veamos las dos a la vez como se explicó anteriormente.

Sabemos que para que una imagen no tenga parpadeos ha de tener una frecuencia de refresco superior a 20 Hz (que corresponde al tiempo de persistencia de imagen en el ser humano), con lo que necesitamos que el brazo de nuestro circuito gire como mínimo a esa velocidad. Se ha detectado en el osciloscopio una frecuencia de refresco de 32 Hz, tenemos que la placa tiene que girar a  $32 * 60 = 1929$  RPM.

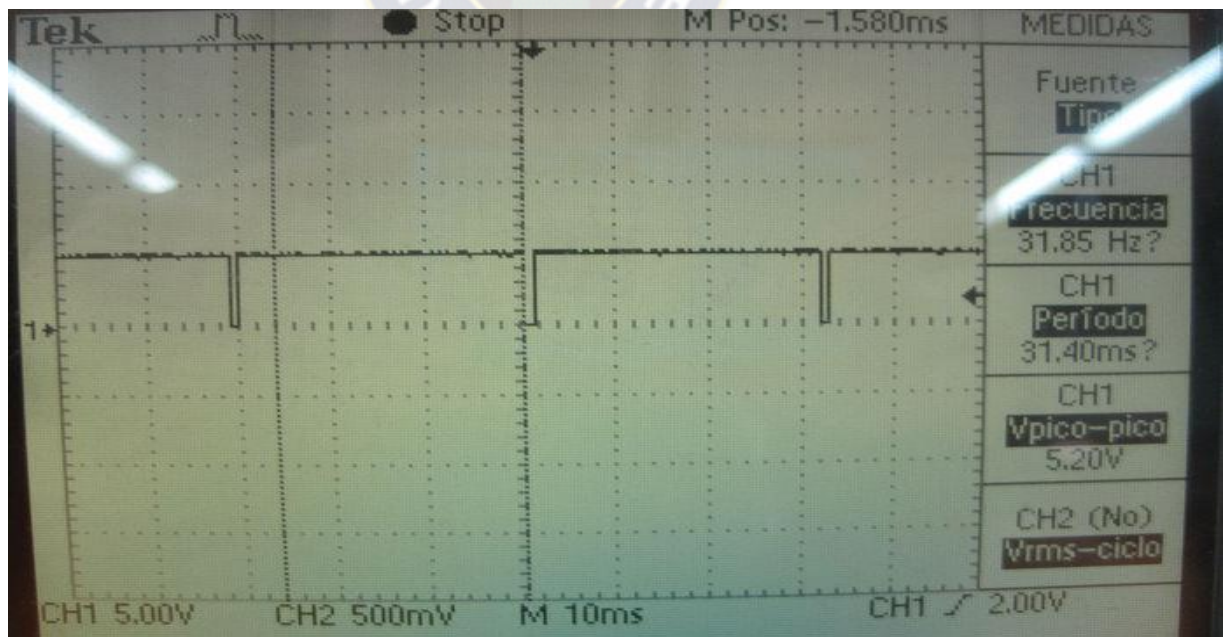


Figura 2.3 Lectura del osciloscopio de la frecuencia de giro

El área de visión que se ha establecido para el circuito que se ha montando es de 360°. La placa principal donde van soldados los leds tiene 32 mm de largo, y aplicando  $2\pi R$  tenemos una longitud de circunferencia de 201 mm. Nuestro campo de visión es una cuarta parte, así que tenemos 50 mm (5 cm).

Quedamos en una frecuencia de 32 Hz, que corresponden a 31 mseg cada giro completo, 7'7 mseg para nuestra área de visualización. Ese es el tiempo que tenemos para mostrar la imagen.

Ejemplo: Generar las letras "E" y "P"

Necesitamos 2 letras. Cada letra ocupa 4 píxeles (llevamos 8). Después de cada carácter E y P ponemos una columna en blanco, excepto al final (1). Total 9 píxeles.

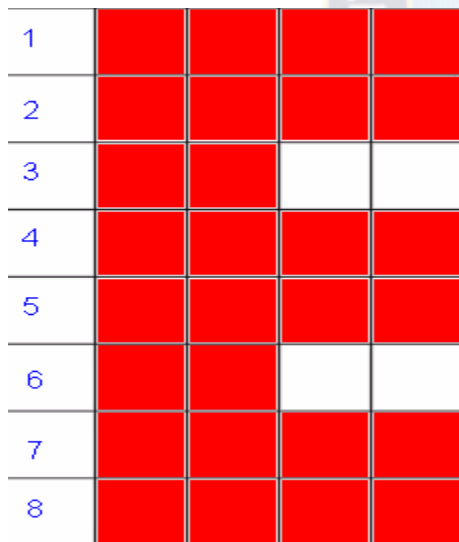


Figura 2.4 Letra E

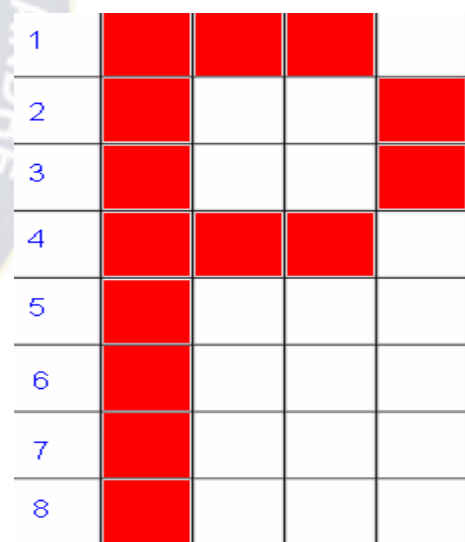


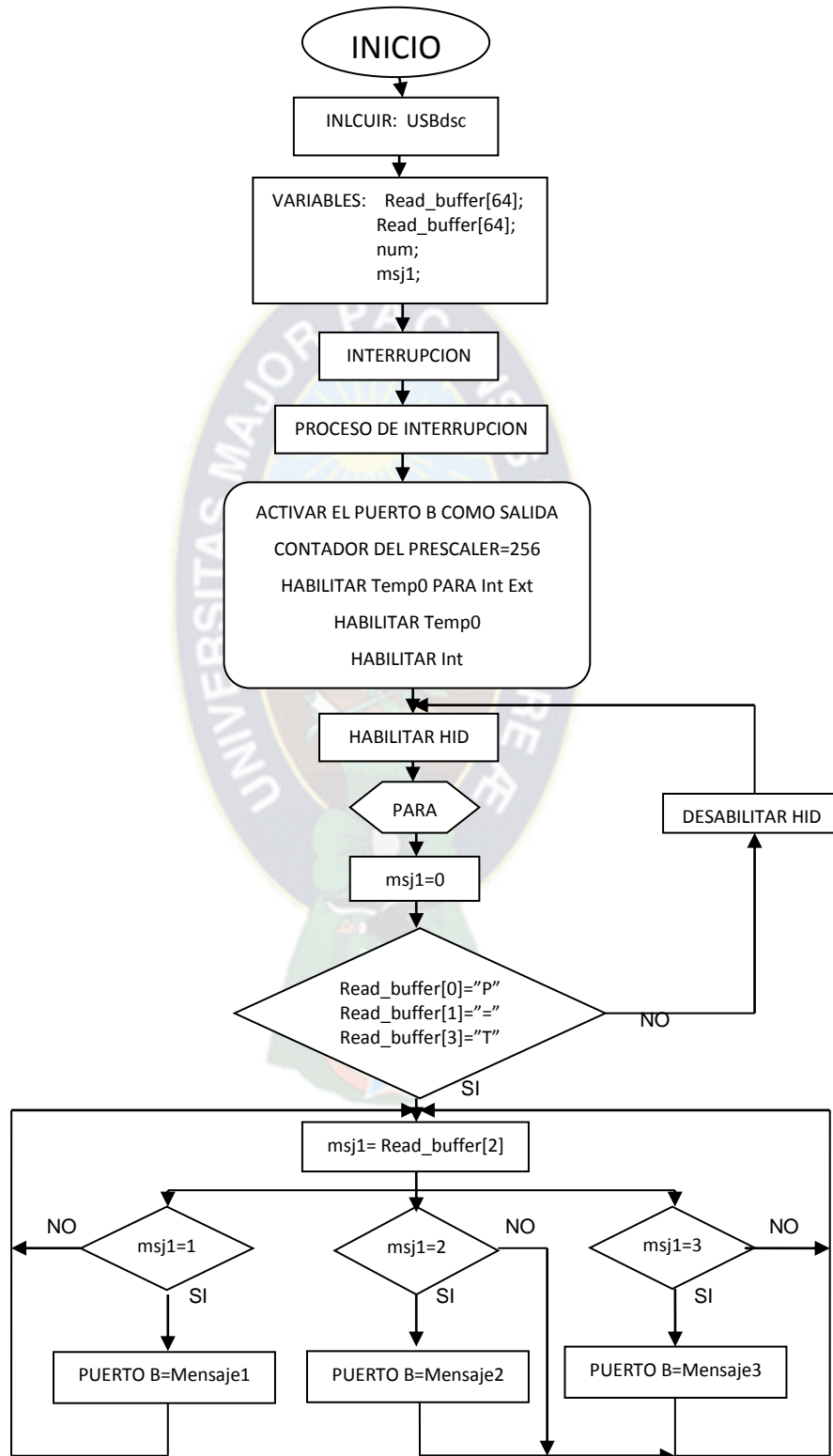
Figura 2.5 Letra P

Si tenemos 9 píxeles y 7'7 milisegundos para dibujarlos todos, sacamos que tenemos que pintar uno cada 855'5 uSeg.

Una vez comprendido lo anterior podemos proceder a realizar el código de programa.

# PROGRAMACION DEL MICROCONTROLADOR PIC18F4550

## DIAGRAMA DE FLUJO



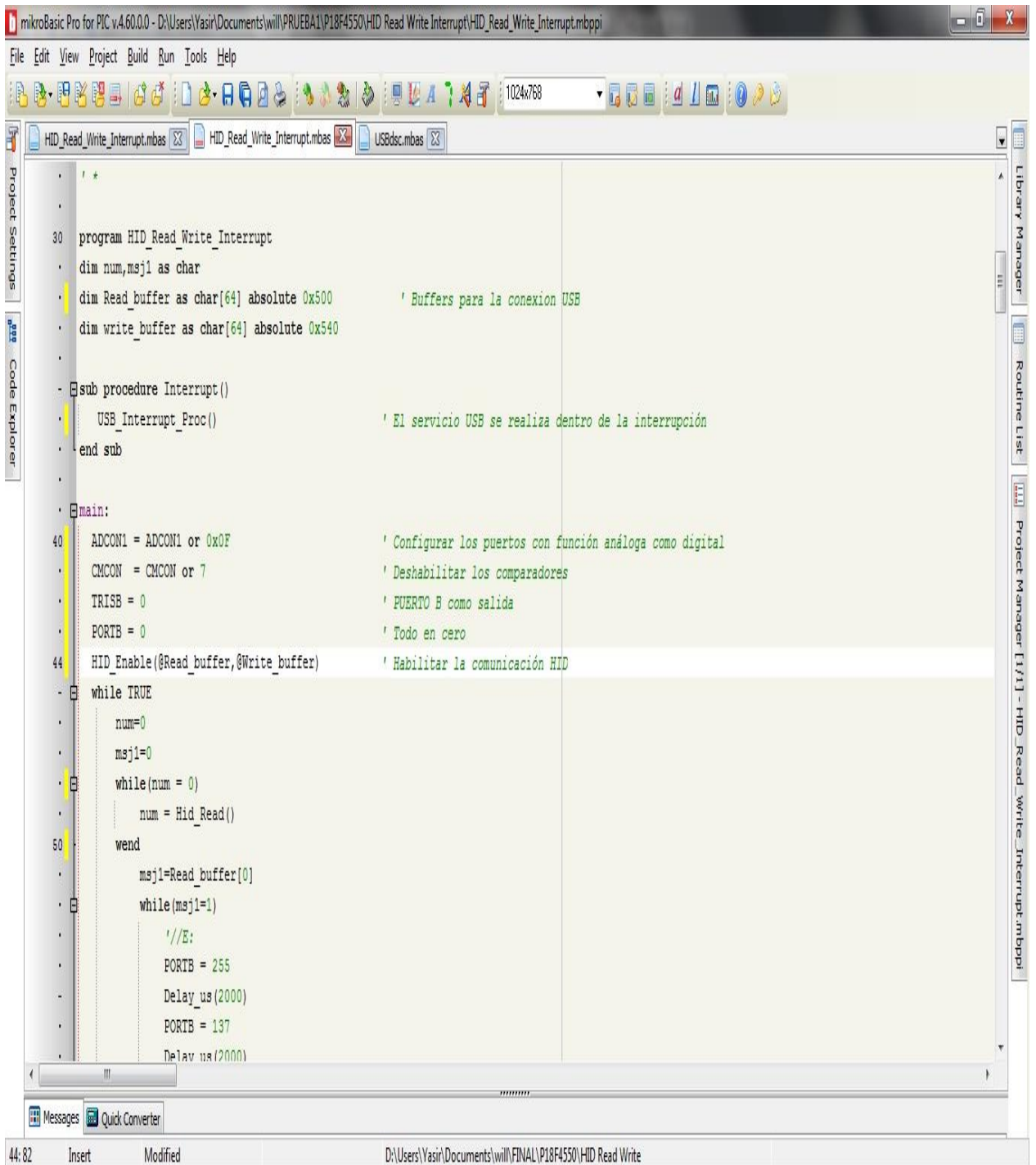


Figura 2.6 Compilador MikroBasic Pro

## CODIGO DE PROGRAMA EN MikroBasic Pro

```
program HID_Read_Write_Interrupt
dim num,msg1 as char
dim Read_buffer as char[64] absolute 0x500           ' Buffers para la conexión USB
dim write_buffer as char[64] absolute 0x540

sub procedure Interrupt()
  USB_Interrupt_Proc()           ' El servicio USB se realiza dentro de la interrupción
end sub

main:
ADCON1 = ADCON1 or 0x0F         ' Configurar los puertos con función analógica como digital
CMCON = CMCON or 7             ' Deshabilitar los comparadores
TRISB = 0                       ' PUERTO B como salida
PORTB = 0                       ' Todo en cero
HID_Enable(@Read_buffer,@Write_buffer)           ' Habilitar la comunicación HID
while TRUE
  num=0
  msg1=0
  while(num = 0)
    num = Hid_Read()
  wend
  'if((Read_buffer[0] = "P") and (Read_buffer[1] = "=") and (Read_buffer[3] = "T"))then
    msg1=Read_buffer[0]
    while(msg1=1)
      '//E:
      PORTB = 255
      Delay_us(2000)
      PORTB = 137
```



Delay\_us(2000)

Delay\_us(2000)

PORTB = 0

Delay\_us(2000)

//L:

PORTB = 255

Delay\_us(2000)

PORTB = 128

Delay\_us(2000)

Delay\_us(2000)

PORTB = 0

Delay\_us(2000)

//T:

PORTB = 1

Delay\_us(2000)

PORTB = 255

Delay\_us(2000)

PORTB = 1

Delay\_us(2000)

PORTB = 0

Delay\_us(2000)

//:-

PORTB = 8

Delay\_us(2000)

Delay\_us(2000)

PORTB = 0

Delay\_us(2000)

//T:

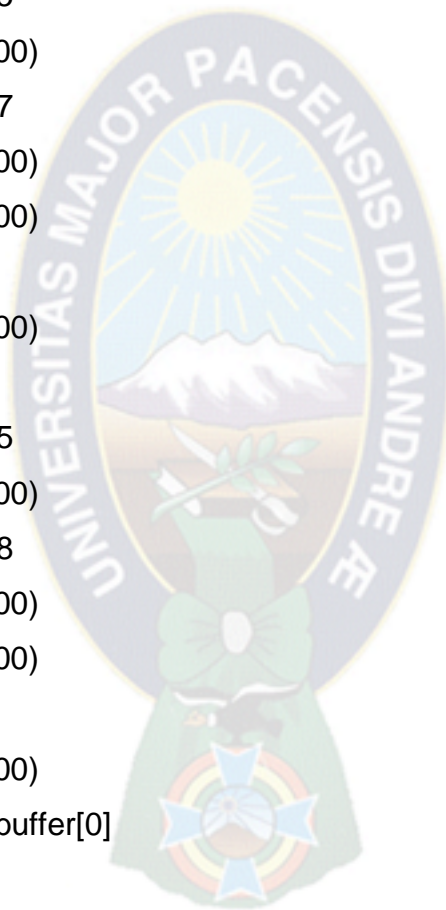
PORTB = 1

Delay\_us(2000)



```
PORTB = 255
Delay_us(2000)
PORTB = 1
Delay_us(2000)
PORTB = 0
Delay_us(2000)
//E:
PORTB = 255
Delay_us(2000)
PORTB = 137
Delay_us(2000)
Delay_us(2000)
PORTB = 0
Delay_us(2000)
//L:
PORTB = 255
Delay_us(2000)
PORTB = 128
Delay_us(2000)
Delay_us(2000)
PORTB = 0
Delay_us(2000)
msj1=Read_buffer[0]
wend
```

```
while(msj1=2)
//F:
PORTB = 255
Delay_ms(2)
PORTB = 9
```



Delay\_ms(2)  
Delay\_ms(2)  
PORTB = 1  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//A:  
PORTB = 192  
Delay\_ms(2)  
PORTB = 48  
Delay\_ms(2)  
PORTB = 46  
Delay\_ms(2)  
PORTB = 33  
Delay\_ms(2)  
PORTB = 46  
Delay\_ms(2)  
PORTB = 48  
Delay\_ms(2)  
PORTB = 192  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//C:  
PORTB = 60  
Delay\_ms(2)  
PORTB = 66  
Delay\_ms(2)  
PORTB = 129  
Delay\_ms(2)



PORTB = 129

Delay\_ms(2)

PORTB = 129

Delay\_ms(2)

PORTB = 66

Delay\_ms(2)

PORTB = 0

Delay\_ms(2)

//:-

PORTB = 8

Delay\_ms(2)

Delay\_ms(2)

PORTB = 0

Delay\_ms(500)

//T:

PORTB = 1

Delay\_ms(2)

PORTB = 255

Delay\_ms(2)

PORTB = 1

Delay\_ms(2)

PORTB = 0

Delay\_ms(2)

//E:

PORTB = 255

Delay\_ms(2)

PORTB = 137

Delay\_ms(2)

Delay\_ms(2)

PORTB = 0



```
Delay_ms(2)
//C:
PORTB = 60
Delay_ms(2)
PORTB = 66
Delay_ms(2)
PORTB = 129
Delay_ms(2)
PORTB = 129
Delay_ms(2)
PORTB = 129
Delay_ms(2)
PORTB = 66
Delay_ms(2)
PORTB = 0
Delay_ms(2)
msj1=Read_buffer[0]
wend
while(msj1=3)
//E:
PORTB = 255
Delay_ms(2)
PORTB = 137
Delay_ms(2)
Delay_ms(2)
PORTB = 0
Delay_ms(2)
//X:
PORTB = 129
Delay_ms(2)
```



PORTB = 102

Delay\_ms(2)

PORTB = 24

Delay\_ms(2)

PORTB = 102

Delay\_ms(2)

PORTB = 129

Delay\_ms(2)

PORTB = 0

Delay\_ms(2)

//A:

PORTB = 192

Delay\_ms(2)

PORTB = 48

Delay\_ms(2)

PORTB = 46

Delay\_ms(2)

PORTB = 33

Delay\_ms(2)

PORTB = 46

Delay\_ms(2)

PORTB = 48

Delay\_ms(2)

PORTB = 192

Delay\_ms(2)

PORTB = 0

Delay\_ms(300)

//M:

PORTB = 255

Delay\_ms(2)



PORTB = 6  
Delay\_ms(2)  
PORTB = 56  
Delay\_ms(2)  
PORTB = 192  
Delay\_ms(2)  
PORTB = 56  
Delay\_ms(2)  
PORTB = 6  
Delay\_ms(2)  
PORTB = 255  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//E:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 137  
Delay\_ms(2)  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//N:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 2  
Delay\_ms(2)  
PORTB = 12  
Delay\_ms(2)  
PORTB = 48



Delay\_ms(2)  
PORTB = 64  
Delay\_ms(2)  
PORTB = 255  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//D:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 129  
Delay\_ms(2)  
Delay\_ms(2)  
PORTB = 66  
Delay\_ms(2)  
PORTB = 60  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//E:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 137  
Delay\_ms(2)  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//G:  
PORTB = 60  
Delay\_ms(2)





PORTB = 66  
Delay\_ms(2)  
PORTB = 129  
Delay\_ms(2)  
PORTB = 145  
Delay\_ms(2)  
PORTB = 82  
Delay\_ms(2)  
PORTB = 52  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//R:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 9  
Delay\_ms(2)  
PORTB = 25  
Delay\_ms(2)  
PORTB = 105  
Delay\_ms(2)  
PORTB = 134  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//A:  
PORTB = 192  
Delay\_ms(2)  
PORTB = 48  
Delay\_ms(2)



PORTB = 46  
Delay\_ms(2)  
PORTB = 33  
Delay\_ms(2)  
PORTB = 46  
Delay\_ms(2)  
PORTB = 48  
Delay\_ms(2)  
PORTB = 192  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
Delay\_ms(2)  
//D:  
PORTB = 255  
Delay\_ms(2)  
PORTB = 129  
Delay\_ms(2)  
Delay\_ms(2)  
PORTB = 66  
Delay\_ms(2)  
PORTB = 60  
Delay\_ms(2)  
PORTB = 0  
Delay\_ms(2)  
//O:  
PORTB = 60  
Delay\_ms(2)  
PORTB = 66  
Delay\_ms(2)



```
PORTB = 129
Delay_ms(2)
Delay_ms(2)
PORTB = 66
Delay_ms(2)
PORTB = 6
Delay_ms(2)
PORTB = 60
Delay_ms(2)
PORTB = 0
Delay_ms(2)
msj1=Read_buffer[0]
wend
'end if
wend
Hid_Disable()
end.
```



## SIMULACION DEL CIRCUITO EN PROTEUS 7.8

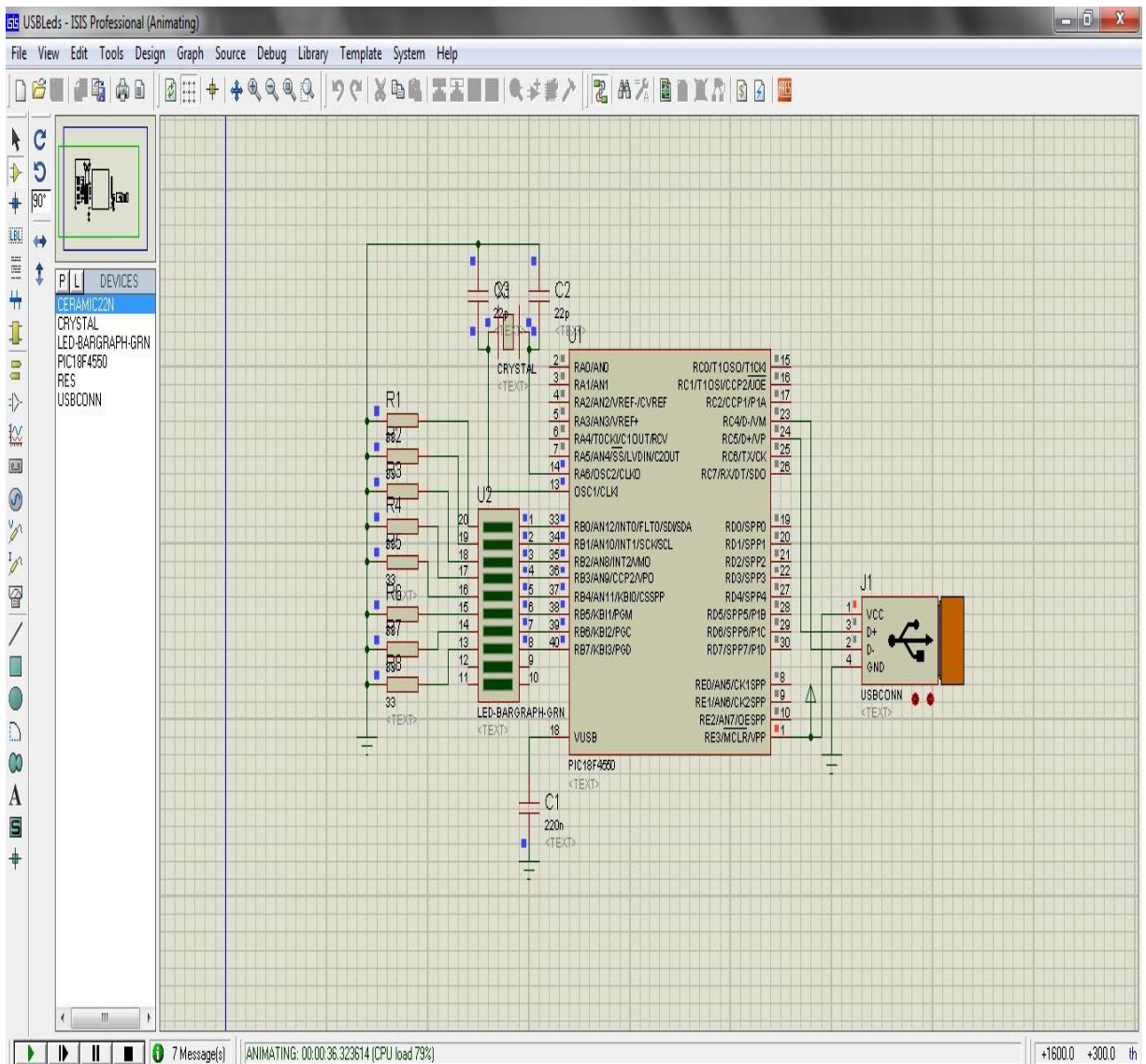


Figura 2.7 USB conectado y esperando datos

En la figura 2.6 se observa el diagrama del circuito y la simulación del estado de conexión del circuito con el ordenador.

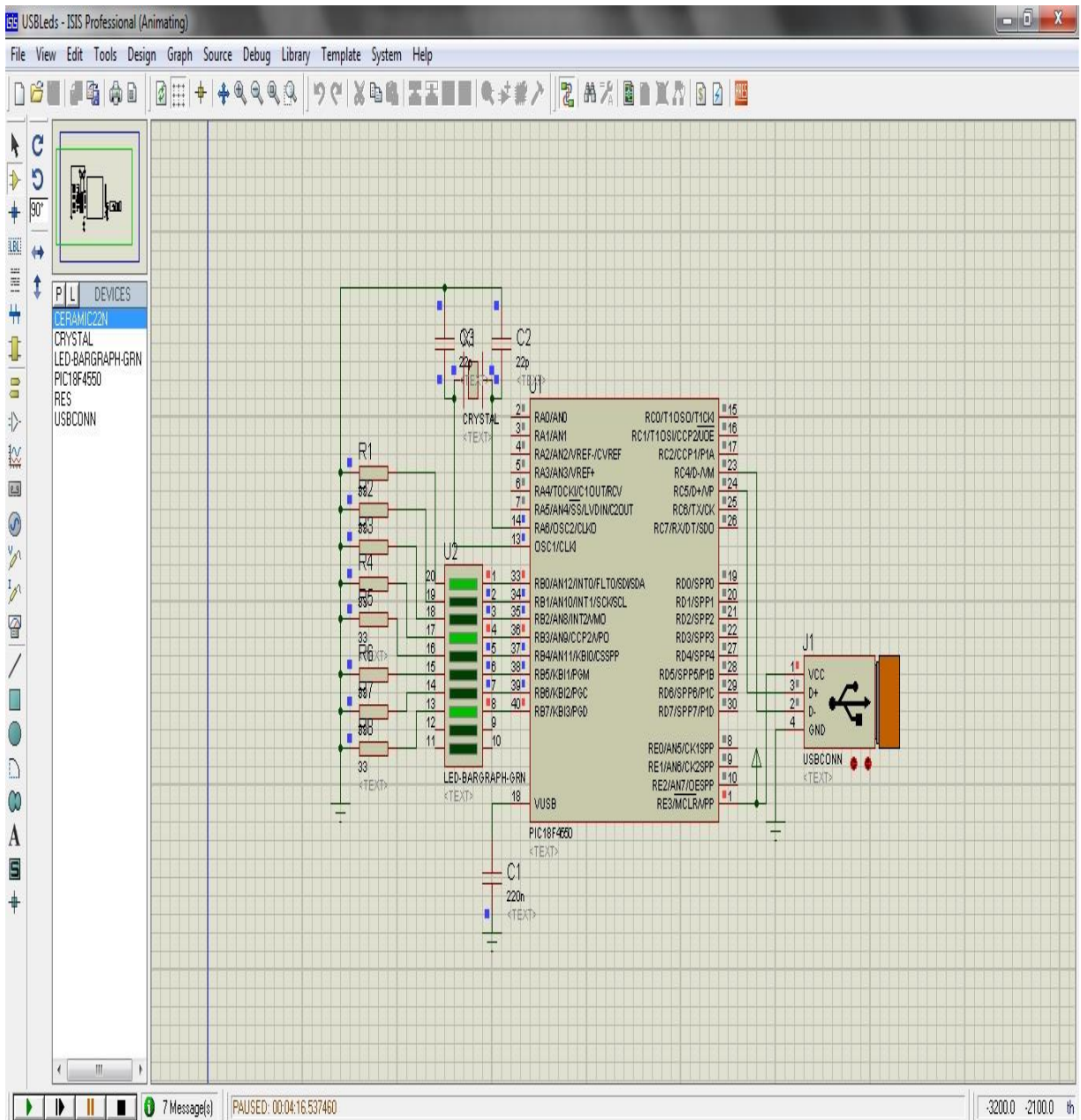


Figura 2.8 USB conectado y esperando datos

En la figura 2.7 se observa la simulación del estado de conexión del circuito con el ordenador, por la cual se transmite la trama de datos desde el programa realizado en Visual Basic 6.0.

## 2.3 SISTEMA DE SELECCIÓN DE CARACTERES

En esta parte se realiza la interacción del ordenador con el microcontrolador por medio del programa elaborado en Visual Basic 6.0, a través de la interface USB de la clase HID (dispositivo de interfaz humana)

### PROGRAMA DE INTERACCION EN VISUAL BASIC 6.0

Para la aplicación del protocolo HID se debe tomar en cuenta dos procesos. Primero, se elaborara la plantilla de interacción y después se crea el modulo de comunicación para la conexión por medio del USB

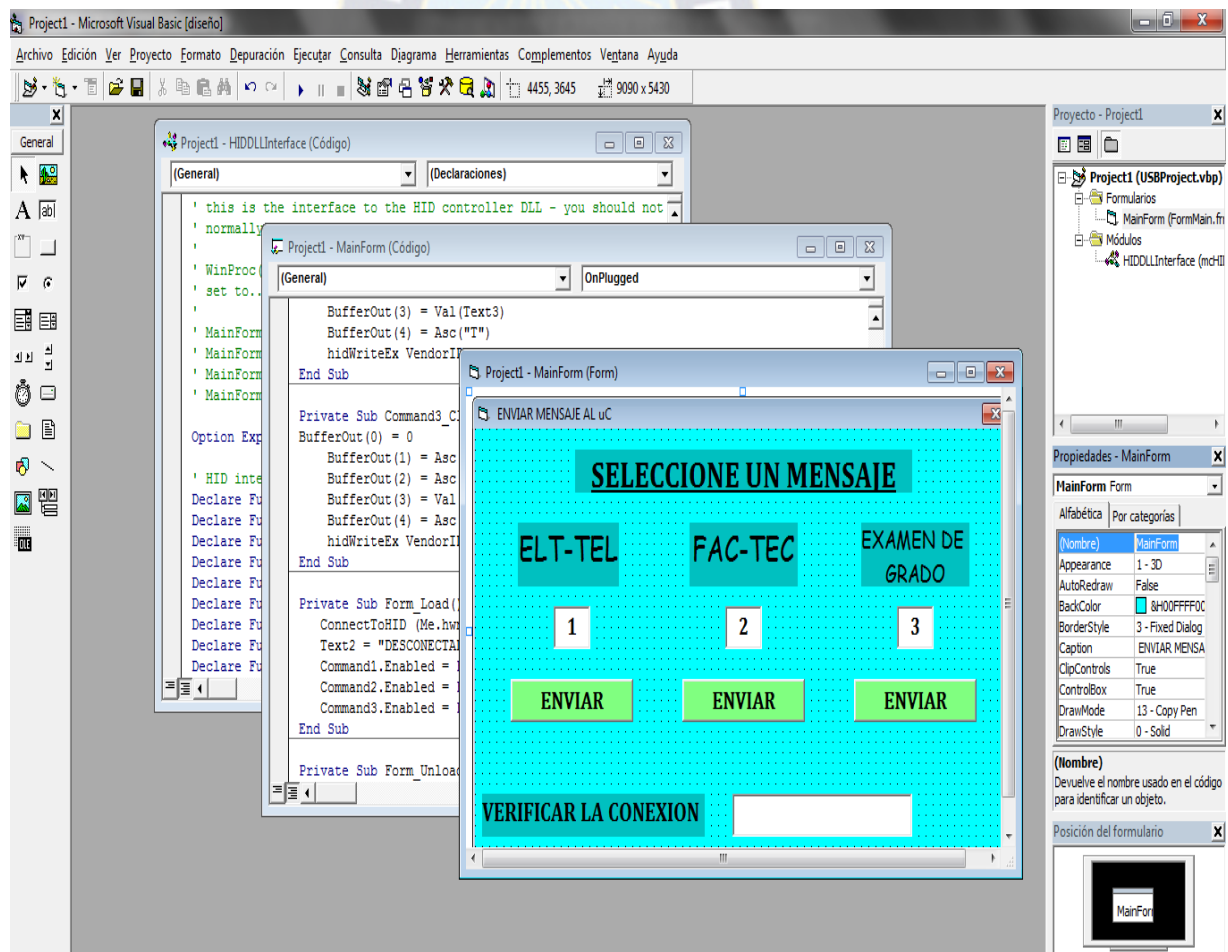


Figura 2.9 Plantilla para la comunicación por USB

En la figura 2.8 se muestra el diseño de la planilla en Visual Basic 6.0 se tiene tres botones de comando para el envío de datos de los mensajes y cuatro cajas de texto, de las cuales tres indican el número de mensaje que se enviara y una indica el estado de conexión del puerto USB.

## **PROGRAMA DEL TRABAJO DE APLICACION EN VISUAL BASIC 6.0**

'ID vendedor y producto

Private Const VendorID = 4660

Private Const ProductID = 1

Dim BufferIn(0 To 8) As Byte

Dim BufferOut(0 To 8) As Byte

'escribir

Private Sub Command1\_Click()

    BufferOut(0) = 0

    BufferOut(1) = Asc("P")

    BufferOut(2) = Asc("=")

    BufferOut(3) = Val(Text1)

    BufferOut(4) = Asc("T")

    hidWriteEx VendorID, ProductID, BufferOut(0)

End Sub

Private Sub Command2\_Click()

    BufferOut(0) = 0

    BufferOut(1) = Asc("P")

    BufferOut(2) = Asc("=")

    BufferOut(3) = Val(Text3)

    BufferOut(4) = Asc("T")

    hidWriteEx VendorID, ProductID, BufferOut(0)

End Sub

Private Sub Command3\_Click()

BufferOut(0) = 0

BufferOut(1) = Asc("P")

BufferOut(2) = Asc("=")

BufferOut(3) = Val(Text4)

BufferOut(4) = Asc("T")

hidWriteEx VendorID, ProductID, BufferOut(0)

End Sub

Private Sub Form\_Load()

ConnectToHID (Me.hwnd)

Text2 = "DESCONECTADO"

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

End Sub

Private Sub Form\_Unload(Cancel As Integer)

DisconnectFromHID

End Sub

'evento de conexion del dispositivo HID

Public Sub OnPlugged(ByVal pHandle As Long)

If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =  
ProductID Then

Text2 = "USB conectado"

Command1.Enabled = True

Command2.Enabled = True



```
Command3.Enabled = True
End If
End Sub
```

```
'evento de desconexion del dispositivo HID
```

```
Public Sub OnUnplugged(ByVal pHandle As Long)
```

```
    If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =  
ProductID Then
```

```
    Text2 = "USB desconectado"
```

```
    Command1.Enabled = False
```

```
End If
```

```
End Sub
```

```
'Compara los ID con los ID del controlador HID
```

```
Public Sub OnChanged()
```

```
    Dim DeviceHandle As Long
```

```
    DeviceHandle = hidGetHandle(VendorID, ProductID)
```

```
    hidSetReadNotify DeviceHandle, True
```

```
End Sub
```

```
'Evento de llegada de datos HID por USB, sirve para leer informacion
```

```
Public Sub OnRead(ByVal pHandle As Long)
```

```
    If hidRead(pHandle, BufferIn(0)) Then
```

```
        'Codigo para leer
```

```
    End If
```

```
End Sub
```

## DESARROLLO DEL MODULO PARA LA CONEXIÓN POR USB

El modulo de la conexión cuenta con las siguientes instrucciones de programa para la comunicación con el puerto USB.

```
' this is the interface to the HID controller DLL - you should not
' normally need to change anything in this file.
'
' WinProc() calls your main form 'event' procedures - these are currently
' set to..
'
' MainForm.OnPlugged(ByVal pHandle as long)
' MainForm.OnUnplugged(ByVal pHandle as long)
' MainForm.OnChanged()
' MainForm.OnRead(ByVal pHandle as long)
```

Option Explicit

```
' HID interface API declarations...
```

```
Declare Function hidConnect Lib "mchID.dll" Alias "Connect" (ByVal pHostWin As Long) As Boolean
```

```
Declare Function hidDisconnect Lib "mchID.dll" Alias "Disconnect" () As Boolean
```

```
Declare Function hidGetItem Lib "mchID.dll" Alias "GetItem" (ByVal pIndex As Long) As Long
```

```
Declare Function hidGetItemCount Lib "mchID.dll" Alias "GetItemCount" () As Long
```

```
Declare Function hidRead Lib "mchID.dll" Alias "Read" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
```

```
Declare Function hidWrite Lib "mchID.dll" Alias "Write" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean
```

```
Declare Function hidReadEx Lib "mcHID.dll" Alias "ReadEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidWriteEx Lib "mcHID.dll" Alias "WriteEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidGetHandle Lib "mcHID.dll" Alias "GetHandle" (ByVal pVendorID As Long, ByVal pProductID As Long) As Long
Declare Function hidGetVendorID Lib "mcHID.dll" Alias "GetVendorID" (ByVal pHandle As Long) As Long
Declare Function hidGetProductID Lib "mcHID.dll" Alias "GetProductID" (ByVal pHandle As Long) As Long
Declare Function hidGetVersion Lib "mcHID.dll" Alias "GetVersion" (ByVal pHandle As Long) As Long
Declare Function hidGetVendorName Lib "mcHID.dll" Alias "GetVendorName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetProductName Lib "mcHID.dll" Alias "GetProductName" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetSerialNumber Lib "mcHID.dll" Alias "GetSerialNumber" (ByVal pHandle As Long, ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetInputReportLength Lib "mcHID.dll" Alias "GetInputReportLength" (ByVal pHandle As Long) As Long
Declare Function hidGetOutputReportLength Lib "mcHID.dll" Alias "GetOutputReportLength" (ByVal pHandle As Long) As Long
Declare Sub hidSetReadNotify Lib "mcHID.dll" Alias "SetReadNotify" (ByVal pHandle As Long, ByVal pValue As Boolean)
Declare Function hidIsReadNotifyEnabled Lib "mcHID.dll" Alias "IsReadNotifyEnabled" (ByVal pHandle As Long) As Boolean
Declare Function hidIsAvailable Lib "mcHID.dll" Alias "IsAvailable" (ByVal pVendorID As Long, ByVal pProductID As Long) As Boolean
```

' windows API declarations - used to set up messaging...

```
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA"  
(ByVal lpPrevWndFunc As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal  
wParam As Long, ByVal lParam As Long) As Long
```

```
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA"  
(ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As  
Long
```

```
' windows API Constants
```

```
Private Const WM_APP = 32768
```

```
Private Const GWL_WNDPROC = -4
```

```
' HID message constants
```

```
Private Const WM_HID_EVENT = WM_APP + 200
```

```
Private Const NOTIFY_PLUGGED = 1
```

```
Private Const NOTIFY_UNPLUGGED = 2
```

```
Private Const NOTIFY_CHANGED = 3
```

```
Private Const NOTIFY_READ = 4
```

```
' local variables
```

```
Private FPrevWinProc As Long ' Handle to previous window procedure
```

```
Private FWinHandle As Long ' Handle to message window
```

```
' Set up a windows hook to receive notification
```

```
' messages from the HID controller DLL - then connect
```

```
' to the controller
```

```
Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean
```

```
    FWinHandle = pHostWin
```

```
    ConnectToHID = hidConnect(FWinHandle)
```

```
    FPrevWinProc = SetWindowLong(FWinHandle, GWL_WNDPROC, AddressOf  
WinProc)
```

End Function

' Unhook from the HID controller and disconnect...

Public Function DisconnectFromHID() As Boolean

    DisconnectFromHID = hidDisconnect

    SetWindowLong FWinHandle, GWL\_WNDPROC, FPrevWinProc

End Function

' This is the procedure that intercepts the HID controller messages...

Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal

wParam As Long, ByVal lParam As Long) As Long

    If pMsg = WM\_HID\_EVENT Then

        Select Case wParam

            ' HID device has been plugged message...

            Case Is = NOTIFY\_PLUGGED

                MainForm.OnPlugged (lParam)

            ' HID device has been unplugged

            Case Is = NOTIFY\_UNPLUGGED

                MainForm.OnUnplugged (lParam)

            ' controller has changed...

            Case Is = NOTIFY\_CHANGED

                MainForm.OnChanged

            ' read event...

            Case Is = NOTIFY\_READ

                MainForm.OnRead (lParam)

        End Select

End If

' next...

WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, lParam)

End Function

## SIMULACION DE LA CONEXIÓN POR EL PUERTO USB



Figura 2.10 USB desconectado

En la figura 2.10 se puede observar que es posible verificar la conexión USB en la plantilla del programa, cuando no existe conexión USB no se activa los botones de comando que envían los datos para la generación de los mensajes



Figura 2.11 USB conectado

En la figura 2.11 se puede observar que la conexión USB ha sido establecida y los botones de comando fueron habilitados, por lo tanto es posible enviar los datos que representaran los mensajes en el microcontrolador para el manejo de la secuencia de prendido y apagado de los leds.

## 2.4 ROTACION DEL CIRCUITO

La rotación del circuito es una etapa muy importante del trabajo de aplicación, por lo tanto fue desarrollada con mucho cuidado y fueron tomados en cuenta muchos aspectos que serán explicados a continuación.

### Placa de anillos

En un principio se iba a utilizar como base o plataforma del proyecto un ventilador de mano pero debido a la insuficiente potencia del motor se optó por construir una base más sólida y usar un motor de mayor potencia. Por este motivo se tuvo que hacer una división de alimentaciones, una para el motor y otra para la placa.

A la placa principal se le tiene que superponer una placa de anillos o placa de contacto, la cual se encargará de alimentar con 5V la placa principal. Esta placa se distribuye de la siguiente manera:

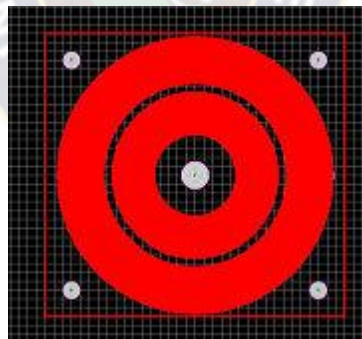
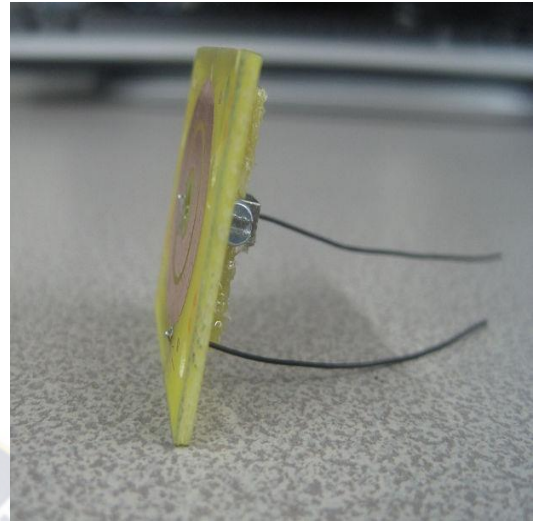
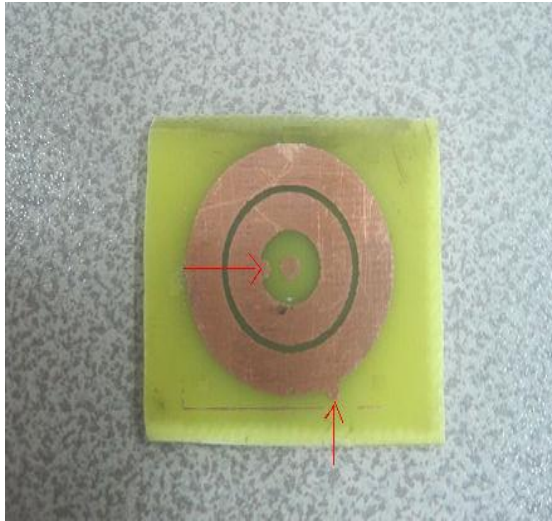


Figura 2.12 Prototipo de la placa de anillos

En el círculo interior se conectará la tensión de entrada. El círculo exterior, se encargará de conectar la masa desde la fuente.





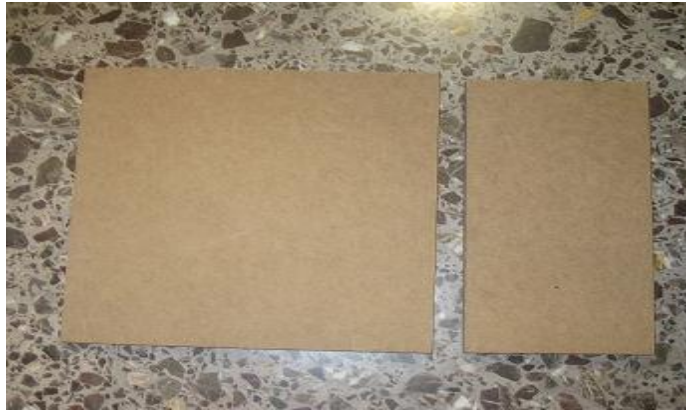
**Figuras 2.13 y 2.14 Placa de anillos**

Si se observa con detenimiento las imágenes 2.13 y 2.14 se pueden observar en los dos anillos una pequeña semicircunferencia en la que se hará un agujero para pasar un cable que vaya a la tensión de entrada ( $V_{in}$ ) y a masa.

Una vez se hayan pasado los cables de  $V_{in}$  y masa se atornillará la placa de anillos a la principal. Es importante que el peso esté bien repartido a lo largo del eje y esté bien centrado en el rotor para evitar que al girar, produzca vibraciones.

### **Montaje de la estructura de soporte**

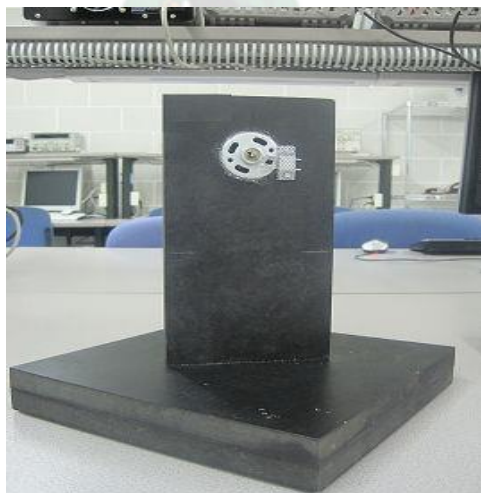
Para montar una buena base se ha tenido en cuenta escoger un material que sea rígido ya que ha de sostener el motor que pesa 147g y ejercerá una fuerza al girar. Por ese motivo hemos cogido una madera tanto para la base como para hacer de mástil.



**Figura 2.15 Maderas sin pintar**

La madera que hace de base tiene unas medidas de 40x40cm y el mástil 20x40cm. Estas medidas, son las ideales para evitar vibraciones durante el giro del motor y para soportar las posibles fuerzas provocadas por el movimiento de la placa.

En la madera que quedarás en posición vertical se le ha hecho un agujero con una broca tipo pala del diámetro del motor para así pasarlo por medio. Las maderas se unen en posición tangencial mediante tornillos y finalmente se han pintado de negro para que se vean mejor las imágenes generadas por los leds y al mismo tiempo por estética.



**Figura 2.16 Base con motor DC parte frontal**

## CAPITULO III

### 3. COSTOS

En la siguiente tabla se detalla el presupuesto de todo el montaje del proyecto desde los dispositivos de la placa hasta los materiales de su estructura de soporte.

<b>MATERIAL</b>	<b>PRECIO POR UNIDAD</b>	<b>CANTIDAD</b>	<b>TOTALBs.</b>
MICROCONTROLADOR PIC 18F4550	65,00.-	1	65,00.-
PLACA BASE	15,00.-	1	15,00.-
PLACA SECUNDARIA	10,00.-	1	10,00.-
RESISTENCIAS	0,30.-	8	2,40.-
DIODOS LED	1,50.-	8	12,00.-
CAPACITORES CERAMICOS	1,00.-	4	4,00.-
CAPACITOR ELECTROLOTICO	2,50.-	1	2,50.-
CRISTAL DE CUARZO	8,00.-	1	8,00.-
MOTOR DC	15,00.-	1	15,00.-
MADERAS	25,00.-	2	50,00.-
ZOCALO PARA MICROCONTROLADOR	29,00.-	1	29,00.-
<b>TOTAL</b>			<b>212,90</b>

Tabla 3.1 Presupuesto Proyecto

#### **4. CONCLUSIONES**

Como culminación del presente trabajo de aplicación se pueden establecer las siguientes conclusiones:

- La construcción que se ha aplicado al circuito está basada en una placa experimental para poder jugar con distintas variables de programación y disposiciones mecánicas, como pueden ser las distancias entre leds para obtener caracteres más grandes o más pequeños y también para que posean una versatilidad suficiente como para instalarlos en un álabe de algún ventilador o para acoplarlo fácilmente a cualquier sistema mecánico de elevada velocidad de rotación.
- La variedad de usos, aplicaciones, textos o imágenes que se pueden mostrar a partir de este sencillo principio son increíbles y están sólo limitadas por nuestra imaginación.

#### **5. RECOMENDACIONES**

Se recomienda que para la implementación del trabajo, revisar con mucho cuidado la etapa de rotación del circuito, puesto que, es una parte fundamental del proyecto.

Se recomienda que para una mejor definición visual de los caracteres que generan los leds, se incremente la cantidad de los mismos. Además no hay que olvidar que los leds deben ser de alta luminosidad preferentemente.

## 6. BIBLIOGRAFIA

### DIRECCIONES URL

- [1] <http://neuro-fabiola.blogspot.com/>
- [2] [http://azu-zavaleta-cbs.blogspot.com/2011\\_03\\_27\\_archive.html](http://azu-zavaleta-cbs.blogspot.com/2011_03_27_archive.html)
- [3] <http://www.ojodigital.com/foro/tutoriales/263269-blanco-y-negro-decente-en-unos-sencillos-pasos.html>
- [4] <http://criticatucine.blogspot.com/2012/01/el-cine-antes-de-nacer.html>
- [5] <http://timerime.com/es/evento/480812/El+traumtropo/>
- [6] <http://flickeflu.com/tags/zootropo>
- [7] <http://www.ilusionario.es/APLICACIONES/animac.htm>
- [8] <http://www.monografias.com/trabajos73/miranda-panoptica-ythos-logocentric>
- [9] <http://leonra.wordpress.com/2009/12/07/el-disco-de-nipkow/>
- [10] <http://www.neoteo.com/pov-escribe-con-luz-en-el-aire>
- [11] <http://www.todopic.com.ar/foros/index.php?topic=35993.15;wap2>
- [12] <http://ladelec.com/practicas/circuitos-por-arossini/593-como-aprender-a-usar-el-puerto-usb-con-pic-18f4550-y-visual-basic-60>
- [13] <http://www.electronicatoribio.es/micro-motores/motor-de-corriente-6044.html>
- [14] <http://luzsumergible.com/tienda/es/normal/12-100x-led-5mm-alta.html>

## 7. ANEXOS

### Imágenes de la implementación

