

# **UNIVERSIDAD TÉCNICA DE BABAHOYO**



**FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA**

**Escuela de Electrónica**

## **Tesis de Grado**

**CONSTRUCCION DE LETRERO LED PROGRAMABLE UTILIZANDO  
MICROCONTROLADORES PICS PARA MOSTRAR MENSAJES INFORMATIVOS EN  
LA UNIVERSIDAD TECNICA DE BABAHOYO**

**PREVIA A LA OBTENCIÓN DEL TÍTULO DE:**

**Tecnólogo en Electrónica y Telecomunicaciones**

**AUTORES**

**Fabricio Muñoz Gómez**

**Milton Rodríguez Martínez**

**BABAHOYO**

**2011**

## **TEMA.**

CONSTRUCCION DE LETRERO LED PROGRAMABLE UTILIZANDO  
MICROCONTROLADORES PICS PARA MOSTRAR MENSAJES  
INFORMATIVOS EN LA UNIVERSIDAD TECNICA DE BABAHOYO

**EL JURADO CALIFICADOR**

**OTORGA AL PRESENTE TRABAJO LA**

**CALIFICACION: -----**

**EQUIVALENTE A: -----**

Babahoyo, Septiembre del 2011

## **DEDICATORIA**

Dedico este proyecto a Dios y a nuestras familias porque son los pilares fundamentales, por ser los principales autores de nuestra existencia y darnos su apoyo incondicional, al Ing. José Sandoya Villafuerte por su incansable labor y apoyo incondicional para la realización de este proyecto y a las personas que de una u otra manera nos brindaron sus conocimientos, para la culminación de este trabajo.

Autor.

# ÍNDICE

## **CAPÍTULO I**

1. EL PROBLEMA
  - 1.1 PLANTEAMIENTO DEL PROBLEMA
  - 1.2. FORMULACION DEL PROBLEMA
  - 1.3 DELIMITACION

## **CAPÍTULO II**

2. OBJETIVOS
  - 2.1 OBJETIVOS GENERAL
  - 2.2 OBJETIVOS ESPECIFICOS
  - 2.3 JUSTIFICACION

## **CAPÍTULO III**

- 3 MARCO TEÓRICO
  - 3.1 Matriz de Leds.
    - COMPOSICION DE LOS LEDCompuestos empleados en la construcción de LED
    - FUNCIONAMIENTO FISICO DEL LED
    - LED DE COLORES
    - APLICACIONES DE LOS LED
    - VENTAJAS DEL LED
    - DESVENTAJAS DEL LED
    - CONEXIÓN DE LOS LED
    - PRINCIPIO FISICO
    - TEORIA DE BANDAS
    - CARACTERISTICAS DEL LEDDimensiones y color del diodo
  - 3.2 Registro De DesplazamientoTipos de registros
  - Entrada serie/salida paralelo
  - Entrada paralelo/salida serie

Cómo funcionan los registros de Desplazamientos.

3.3 Resistencias

3.4 Microcontrolador PIC 16F628A.

- Características
- Diagrama de conexión.

3.5 Transistores

- Características de los Transistores:
- Funcionamiento.

3.6 Optoacopladores.

- Estructura interna de un optoacoplador

¿Qué tipo de Optoacopladores hay?

Características.

Funcionamiento.

3.7 Circuito Programador PICS

3.8 Software y aplicaciones

3.9 Mensajes Informativos a Presentar en el Letrero Leds

4 ¿Cómo funciona la matriz?

4.1 El circuito controlador

4.2 El display

4.3 Procedimientos

- Materiales

4.4 Diagrama de Circuito de control

4.5 Recursos Técnicos

- HARDWARE
- Software
- Recursos Materiales

**CONCLUSIONES**

**BIBLIOGRAFÍA**

# CAPÍTULO I

## 1. EL PROBLEMA

### 1.1 PLANTEAMIENTO DEL PROBLEMA

Como es de conocimiento general en la universidad técnica de Babahoyo no existen letreros de este tipo por este motivo los estudiantes tenemos que andar preguntando en los diversos departamentos para obtener información acerca de cualquier evento de índole general tales como matrículas, seminarios etc. ya que estamos comúnmente acostumbrados a observar hojas volantes, periódicos murales o simplemente lo escuchamos. La construcción de este letrero led programable con pics se lo plantea ante la necesidad de contar con un medio que permita llamar la atención a los estudiantes y de esta forma optimizar la comunicación en la universidad técnica de Babahoyo para así ahorrar tiempo al alumno y al personal docente cabe indicar también que ahorraríamos energía eléctrica ya que su consumo es muy bajo por ende puede permanecer encendido si se desea las 24 horas del día ya que no emana calor y la durabilidad de los led va dentro de **50.000 a 60.000 horas**. Comparadas con **1000-2000 horas** de una lámpara tradicional y si vamos a potencia luminosa la relación es 3w led igual a una bombilla de 50w. Su costo es un poco alto pero se compensa con el ahorro de energía eléctrica y poco mantenimiento que este necesita además tiene la ventaja de

poder cambiar el mensaje cuando y cuantas veces queramos y de esta forma podemos anunciar lo que deseamos .

## **1.2 FORMULACION DEL PROBLEMA**

Como construir un letrero led programable con pics para mostrar mensajes informativos en la universidad técnica de Babahoyo aprovechando al máximo sus ventajas de bajo consumo, durabilidad, poco mantenimiento.

## **1.3 DELIMITACION**

El presente proyecto se lo desarrollara en el laboratorio de Electricidad y Electrónica de la Escuela de Sistemas de la Facultad de Administracion Finanzas e Informatica de la Universidad Técnica de Babahoyo en un tiempo no mayor a 90 días a partir de la aprobacion del presente anteproyecto.

**Objeto de investigación.-** Sistema en Basic 6.0 para programar un letrero led utilizando microcontroladores pics

**Campo de acción.-** Tecnólogo en Electrónica y Telecomunicaciones

**Lugar.-** Área Académica (Secretaria F.A.F.I)

**Tiempo.-** 2010-2011



## **CAPÍTULO II**

### **2. OBJETIVOS**

#### **2.1 OBJETIVOS GENERAL**

Construcción de letrero led programable utilizando microcontroladores pic.

#### **2.2 OBJETIVOS ESPECIFICOS**

- Investigar bibliografía requerida para la construcción de este letrero.
- Realizar una investigación de cada uno de los componentes electrónicos a utilizar en la fabricación del letrero.
- Programar diversos tipos de mensajes en el pic 16f628a y probarlos en el letrero.

### **2.3 JUSTIFICACION**

El interés de este trabajo radica en proponer una alternativa novedosa, eficiente, ecológica y menos costosa mediante la utilización de Tecnología leds y microcontroladores. Por otra parte, la difusión de los resultados obtenidos servirá de motivación a los estudiantes de electrónica para indagar en nuevos campos de aplicación.

En sentido nuestra preocupación se orienta no solamente al aspecto económico si no a la formación profesional de los estudiantes de la escuela de Sistemas especialización Electrónica, de la Universidad Técnica de Babahoyo quienes requieren estar en contacto directo con problemas surgidos de la práctica, lo cual se posibilitará en gran medida con la construcción de proyectos similares al de la presente propuesta. Ya que los microcontroladores tienen una gama muy extensa de utilización.

## CAPÍTULO II

### 3. MARCO TEÓRICO

#### MATRIZ DE LEDS.



El LED (*Light-Emitting Diode*: Diodo Emisor de Luz), es un dispositivo semiconductor que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN en la cual circula por él una corriente eléctrica. Este fenómeno es una forma de electroluminiscencia, el LED es un tipo especial de diodo que trabaja como un diodo común, pero que al ser atravesado por la corriente eléctrica, emite luz. Este dispositivo semiconductor está comúnmente encapsulado en una cubierta de plástico de mayor resistencia que las de vidrio que usualmente se emplean en las lámparas incandescentes. Aunque el plástico puede estar coloreado, es sólo por razones estéticas, ya que ello no influye en el color de la luz emitida. Usualmente un LED es una fuente de luz compuesta con diferentes partes, razón por la cual el patrón de intensidad de la luz emitida puede ser bastante complejo.

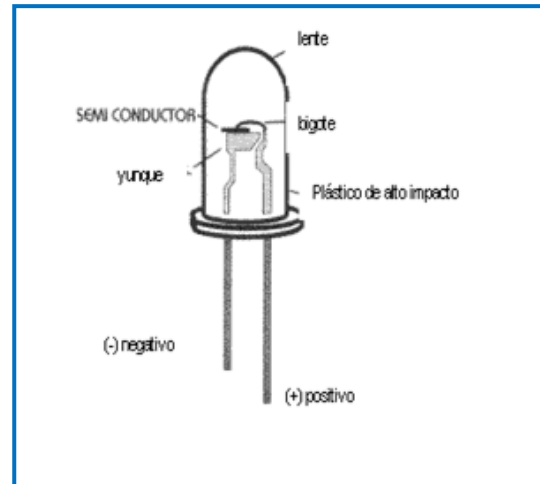
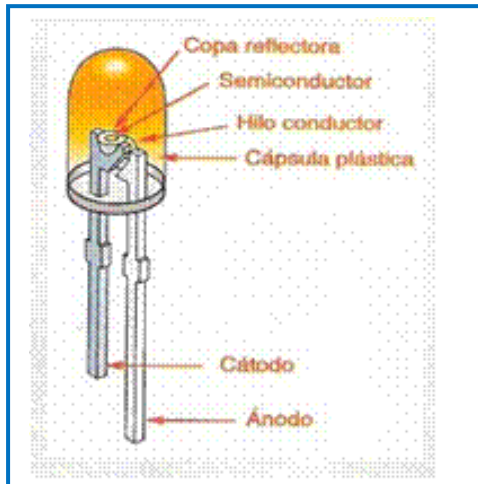
Para obtener una buena intensidad luminosa debe escogerse bien la corriente que atraviesa el LED y evitar que este se pueda dañar; para ello, hay que tener en cuenta que el voltaje de operación va desde 1,8 hasta 3,8 voltios aproximadamente (lo que está relacionado con el material de fabricación y el color de la luz que emite) y la gama de intensidades que debe circular por él varía según su aplicación. Los Valores típicos de corriente directa de polarización de un LED están comprendidos entre los 10 y 20 miliamperios (mA) en los diodos de color rojo y de entre los 20 y 40 miliamperios (mA) para los otros LED. Los diodos LED tienen enormes ventajas sobre las lámparas indicadoras comunes, como su bajo consumo de energía, su mantenimiento casi nulo y con una vida aproximada de 100,000 horas. Para la protección del LED en caso haya picos inesperados que puedan dañarlo. Se coloca en paralelo y en sentido opuesto un diodo de silicio común.

En general, los LED suelen tener mejor eficiencia cuanto menor es la corriente que circula por ellos, con lo cual, en su operación de forma optimizada, se suele buscar un compromiso entre la intensidad luminosa que producen (mayor cuanto más grande es la intensidad que circula por ellos) y la eficiencia (mayor cuanto menor es la intensidad que circula por ellos).



Símbolo del LED

- **ESTRUCTURA DEL LED**



- **COMPOSICION DE LOS LED**

Existen diodos LED de varios colores que dependen del material con el cual fueron contruidos. Hay de color rojo, verde, amarillo, ámbar, infrarrojo, entre otros.

LED rojo: Formado por GaP consiste en una unión p-n obtenida por el método de crecimiento epitaxial del cristal en su fase líquida, en un substrato.

La fuente luminosa está formada por una capa de cristal p junto con un complejo de ZnO, cuya máxima concentración está limitada, por lo que su luminosidad se satura a altas densidades de corriente. Este tipo de LED funciona con baja densidades de corriente ofreciendo una buena luminosidad, utilizándose como dispositivo de visualización en equipos portátiles. El constituido por GaAsP consiste en una capa p

obtenida por difusión de Zn durante el crecimiento de un cristal n de GaAsP, formado en un substrato de GaAs, por el método de crecimiento epitaxial en fase gaseosa.

Actualmente se emplea los LED de GaAs debido a su mayor luminosidad. El máximo de radiación se halla en la longitud de onda 660 nm.

LED anaranjado y amarillo: Están compuestos por GaAsP al igual que sus hermanos los rojos pero en este caso para conseguir luz anaranjada y amarilla así como luz de longitud de onda más pequeña, lo que hacemos es ampliar el ancho de la "banda prohibida" mediante el aumento de fósforo en el semiconductor. Su fabricación es la misma que se utiliza para los diodos rojos, por crecimiento epitaxial del cristal en fase gaseosa, la formación de la unión p-n se realiza por difusión de Zn.

Como novedad importante en estos LED se mezcla el área emisora con una trampa isoelectrónica de nitrógeno con el fin de mejorar el rendimiento.

**LED verde:** El LED verde está compuesto por GaP. Se utiliza el método de crecimiento epitaxial del cristal en fase líquida para formar la unión p-n.

Al igual que los LED amarillos, también se utiliza una trampa isoelectrónica de nitrógeno para mejorar el rendimiento. Debido a que

este tipo de LED posee una baja probabilidad de transición fotónica, es importante mejorar la cristalinidad de la capa n. La disminución de impurezas a larga la vida de los portadores, mejorando la cristalinidad. Su máxima emisión se consigue en la longitud de onda 555 nm

Material	Longitud de onda	Color
GaAs : Zn	9000 Å	Infrarrojo
GaAsP <sub>0.4</sub>	6600 Å	Rojo
GaAsP <sub>0.5</sub>	6100 Å	Ambra
GaAsP <sub>0.55</sub> : N	5900 Å	Amarillo
GaP : N	5600 Å	Verde

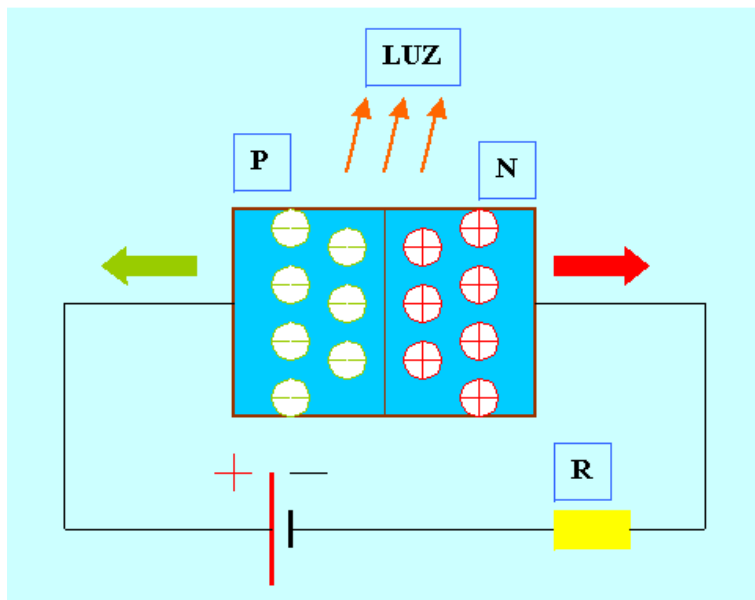
### Compuestos empleados en la construcción de LED

- **FUNCIONAMIENTO FISICO DEL LED**

El funcionamiento físico consiste en que, en los materiales semiconductores, un electrón al pasar de la banda de conducción a la de valencia, pierde energía; esta energía perdida se puede manifestar en forma de un fotón desprendido, con una amplitud, una dirección y una fase aleatoria. El que esa energía se manifieste en (calor por ejemplo) va a depender principalmente del tipo de material semiconductor. Cuando Al polarizar directamente un diodo LED conseguimos que por la unión PN sean inyectados huecos en el material tipo N y electrones en el material tipo P; O sea los huecos de

la zona p se mueven hacia la zona n y los electrones de la zona n hacia la zona p, produciéndose por consiguiente, una inyección de portadores minoritarios.

Ambos desplazamientos de cargas constituyen la corriente que circula por el diodo. Si los electrones y huecos están en la misma región, pueden recombinarse, es decir, los electrones pueden pasar a "ocupar" los huecos, "cayendo" desde un nivel energético superior a otro inferior más estable



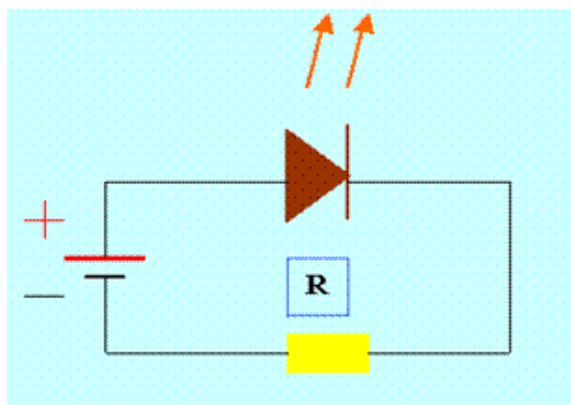
### **Diodo emisor de luz con la unión polarizada en sentido directo**

Cuando estos portadores se recombinan, se produce la liberación de una cantidad de energía proporcional al salto de banda de energía del material semiconductor. Una parte de esta energía se libera en forma de luz, mientras que la parte restante lo hace en forma de calor,



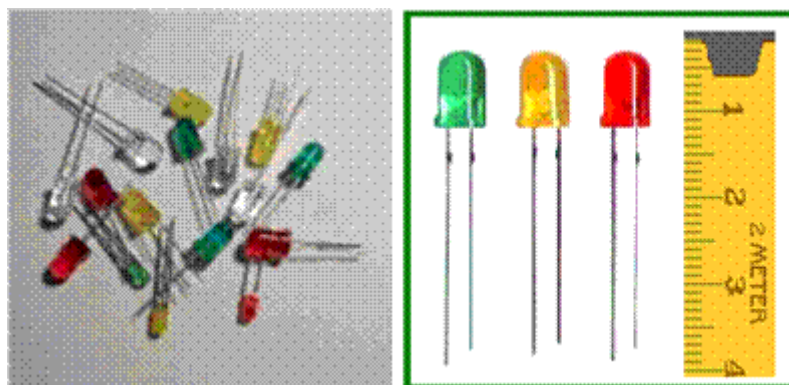
estando determinadas las proporciones por la mezcla de los procesos de recombinación que se producen.

La energía contenida en un fotón de luz es proporcional a su frecuencia, es decir, su color. Cuanto mayor sea el salto de banda de energía del material semiconductor que forma el LED, más elevada será la frecuencia de la luz emitida.



### **Diodo emisor de luz con la unión polarizada en sentido directa**

- **LED DE COLORES**



## • **APLICACIONES DE LOS LED**

Los diodos infrarrojos (IRED) se emplean desde mediados del siglo XX en mandos a distancia de televisores, habiéndose generalizado su uso en otros electrodomésticos como equipos de aire acondicionado, equipos de música, etc. y en general para aplicaciones de control remoto, así como en dispositivos detectores. Los LED se emplean con profusión en todo tipo de indicadores de estado (encendido/apagado) en dispositivos de señalización (de tránsito, de emergencia, etc.) y en paneles informativos. También se emplean en el alumbrado de pantallas de cristal líquido de teléfonos móviles, calculadoras, agendas electrónicas, etc., así como en bicicletas y usos similares. Existen además impresoras LED.

También se usan los LED en el ámbito de la iluminación (incluyendo la señalización de tráfico) es moderado y es previsible que se incremente en el futuro, ya que sus prestaciones son superiores a las de la lámpara incandescente y la lámpara fluorescente, desde diversos puntos de vista.

Se utiliza ampliamente en aplicaciones visuales, como indicadoras de cierta situación específica de funcionamiento y desplegar contadores

- Para indicar la polaridad de una fuente de alimentación de corriente continúa.

- Para indicar la actividad de una fuente de alimentación de corriente alterna.

- En dispositivos de alarma.

- **VENTAJAS DEL LED**

Fiabilidad, mayor eficiencia energética, mayor resistencia a las vibraciones, mejor visión ante diversas circunstancias de iluminación, menor disipación de energía, menor riesgo para el medio ambiente, capacidad para operar de forma intermitente de modo continuo, respuesta rápida, etc. Asimismo, con LED se pueden producir luces de diferentes colores con un rendimiento luminoso elevado, a diferencia de muchas de las lámparas utilizadas hasta ahora, que tienen filtros para lograr un efecto similar (lo que supone una reducción de su eficiencia energética). Todo ello pone de manifiesto las numerosas ventajas que los LED ofrecen.

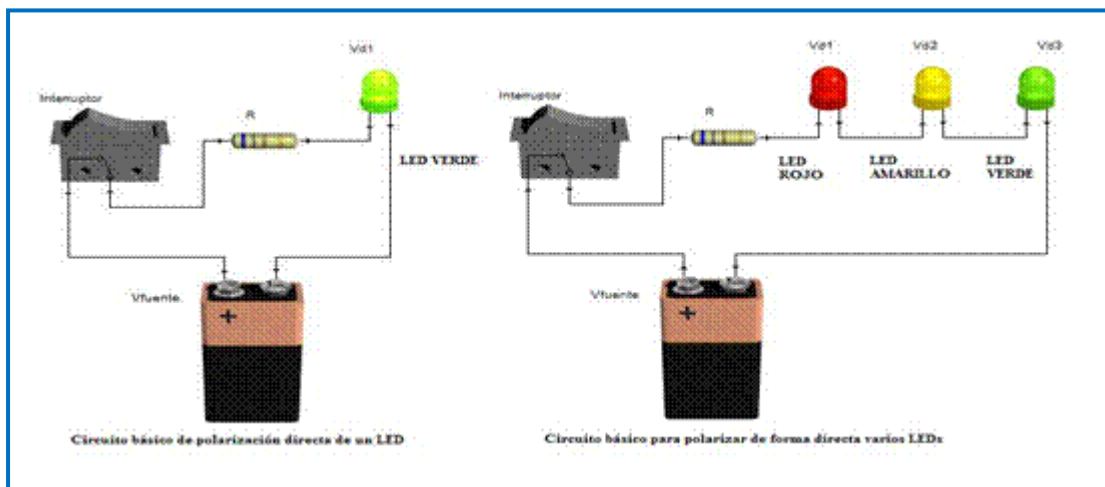
También se utilizan en la emisión de señales de luz que se transmiten a través de fibra óptica.

- **DESVENTAJAS DEL LED**

Las desventajas del **diodo LED** son que su potencia de iluminación es tan baja, que su luz es invisible bajo una fuente de luz brillante y que su ángulo de visibilidad está entre los 30° y 60°. Este último problema se corrige con cubiertas difusores de luz.

- **CONEXIÓN DE LOS LED**

Para conectar LED de modo que iluminen de forma continua, deben estar polarizados directamente, es decir, con el polo positivo de la fuente de alimentación conectada al ánodo y el polo negativo conectado al cátodo. Además, la fuente de alimentación debe suministrarle una tensión o diferencia de potencial superior a su tensión umbral. Por otro lado, se debe garantizar que la corriente que circula por ellos no excede los límites admisibles (Esto se puede hacer de forma sencilla con una resistencia R en serie con los LED). Unos circuitos sencillos que muestran cómo polarizar directamente LED son los siguientes:



- **PRINCIPIO FISICO**

El fenómeno de emisión de luz está basado en la teoría de bandas, por la cual, una tensión externa aplicada a una unión p-n polarizada directamente, excita los electrones, de manera que son capaces de atravesar la banda de energía que separa las dos regiones. Si la

energía es suficiente los electrones escapan del material en forma de fotones.

Cada material semiconductor tiene unas determinadas características que y por tanto una longitud de onda de la luz emitida.

A diferencia de la lámpara de incandescencia cuyo funcionamiento es por una determinada tensión, los Led funcionan por la corriente que los atraviesa. Su conexión a una fuente de tensión constante debe estar protegida por una resistencia limitadora.

- **TEORIA DE BANDAS**

En un átomo aislado los electrones pueden ocupar determinados niveles energéticos pero cuando los átomos se unen para formar un cristal, las interacciones entre ellos modifican su energía, de tal manera que cada nivel inicial se desdobra en numerosos niveles, que constituyen una banda, existiendo entre ellas huecos, llamados bandas energéticas prohibidas, que sólo pueden salvar los electrones en caso de que se les comunique la energía suficiente. En los aislantes la banda inferior menos energética (banda de valencia) está completa con los e- más internos de los átomos, pero la superior (banda de conducción) está vacía y separada por una banda prohibida muy ancha ( $\sim 10$  eV), imposible de atravesar por un e-. En el caso de los conductores las bandas de conducción y de valencia se encuentran superpuestas, por lo que cualquier aporte de energía es suficiente para producir un desplazamiento de los electrones.

Entre ambos casos se encuentran los semiconductores, cuya estructura de bandas es muy semejante a los aislantes, pero con la diferencia de que la anchura de la banda prohibida es bastante pequeña. Los semiconductores son, por lo tanto, aislantes en condiciones normales, pero una elevación de temperatura proporciona la suficiente energía a los electrones para que, saltando la banda prohibida, pasen a la de conducción, dejando en la banda de valencia el hueco correspondiente. En el caso de los diodos LED los electrones consiguen saltar fuera de la estructura en forma de radiación que percibimos como luz (fotones).

- **CARACTERISTICAS DEL LED**

### **Dimensiones y color del diodo**

Actualmente los LED tienen diferentes tamaños, formas y colores. Tenemos LED redondos, cuadrados, rectangulares, triangulares y con diversas formas.

Los colores básicos son rojo, verde y azul, aunque podemos encontrarlos naranjas, amarillos incluso hay un Led de luz blanca.

Las dimensiones en los LED redondos son 3mm, 5mm, 10mm y uno gigante de 20mm. Los de formas poliédricas suelen tener unas dimensiones aproximadas de 5x5mm.

Color	Luminosidad	Consumo	Longitud onda	Diámetro
Rojo	1,25 mcd	10 mA	660 nm	3 y 5 mm
Verde, amarillo y naranja	8 mcd	10 mA		3 y 5 mm
Rojo (alta luminosidad)	80 mcd	10 mA	625 nm	5 mm
Verde (alta luminosidad)	50 mcd	10 mA	565 nm	5 mm
Hiper Rojo	3500 mcd	20 mA	660 nm	5 mm
Hiper Rojo	1600 mcd	20 mA	660 nm	5 mm
Hiper Verde	300 mcd	20 mA	565 nm	5 mm
Azul difuso	1 mcd 60°		470	5 mm
Rojo y verde	40 mcd	20 mA		10 mm

### 3.2 Registro De Desplazamiento

Los registros de desplazamiento son circuitos secuenciales formados por biestables o flip-flops generalmente de tipo D conectados en serie y una circuitería adicional que controlará la manera de cargar y acceder a los datos que se almacenan.

En los de desplazamiento se transfiere información de un flip-flops hacia el adyacente, dentro del mismo registro o a la entrada o salida del mismo.

La capacidad de almacenamiento de un registro es el número total de bits que puede contener.

El funcionamiento se realiza de manera síncrona con la señal de reloj.

Gran parte de los registros de desplazamiento reales incluyen una señal RESET o CLEAR asíncrona, que permite poner simultáneamente todas las salidas en "0" o estado bajo, sin necesidad de introducir ceros seguidos. Esto permite limpiar rápidamente el registro de desplazamiento lo cual es muy importante a nivel práctico.

Sus funciones dentro del sistema digital son:

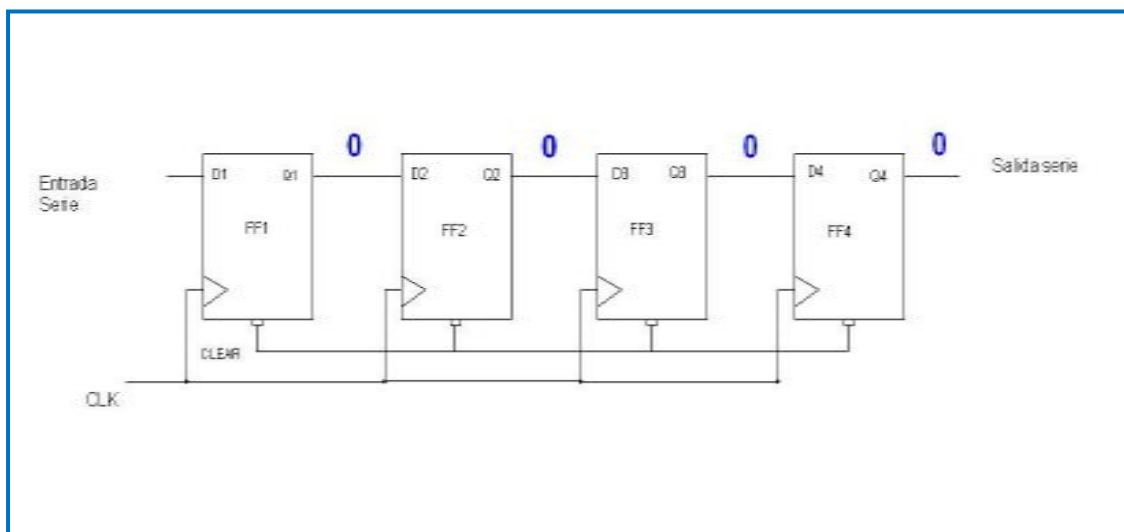
- Servir de almacenamiento temporal de un conjunto de bits sobre los que se está realizando una labor de procesamiento.
- Desplazamiento de datos a lo largo de los flip-flops.

### **Tipos de registros**

- Entrada serie/salida serie:

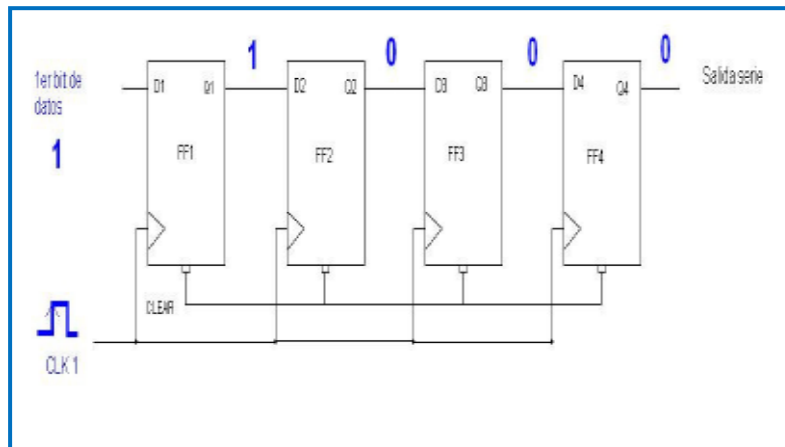
Los datos deben introducirse en serie, es decir, bit a bit por una única línea.

La Salida se obtendrá de la misma manera

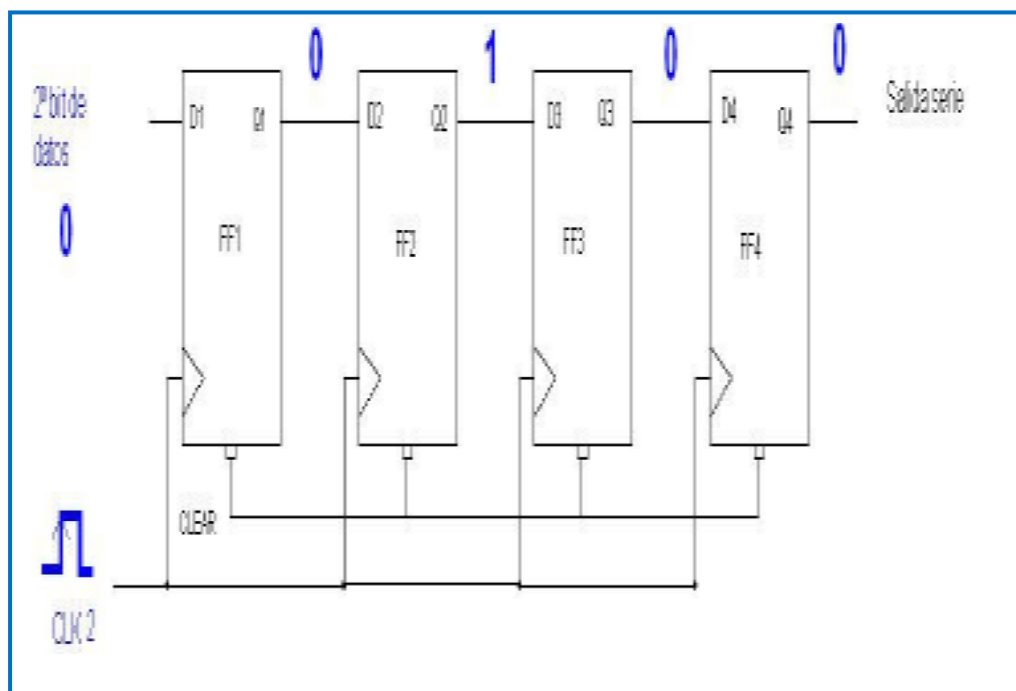




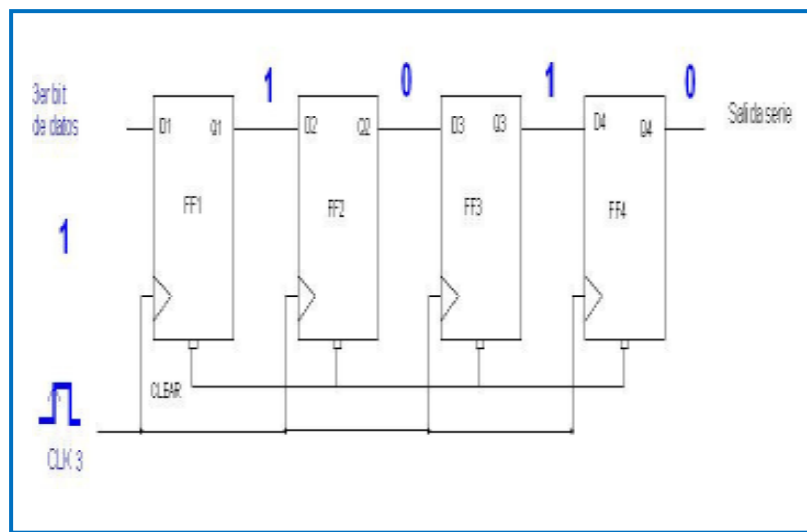
El bit de la derecha es un 1 por lo que le aplicaremos en la entrada lo que hace que  $D=1$  en el FF1. En el momento que halla un flanco de reloj el FF1 pasara al estado SE almacenando el 1.



Seguidamente introduciremos el segundo bit un 0. Esto significa que en el FF1  $D=0$  y en el FF2  $D=1$ . El 1er "1" se ha desplazado al producirse el 2º flanco de reloj y a su vez se ha introducido un nuevo bit.

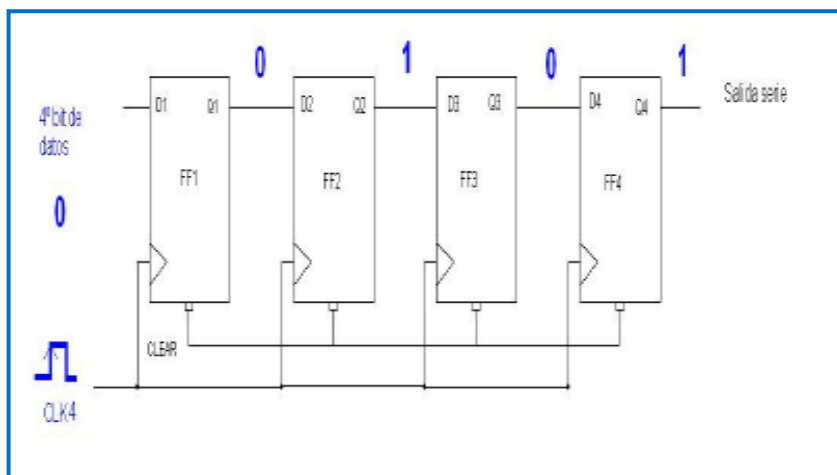


Introducimos el tercer bit de datos entonces tendremos FF3 D=1; FF2 D=0; FF1 D=1



- Introducimos el último bit con lo que quedara la secuencia guardada al producirse el 4° pulso de reloj.

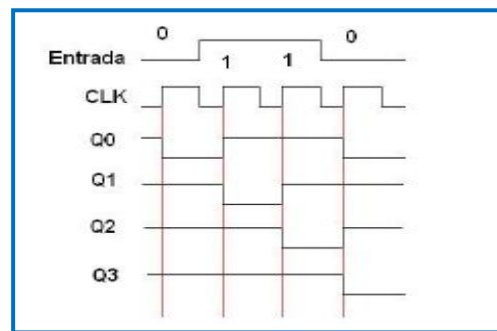
FF4 D=1; FF3 D=0; FF2 D=1; FF1 D=0.



Para obtener la secuencia a la salida debemos seguir el mismo procedimiento que para introducirlos los datos para lo cual tendremos que aplicar tantos pulsos de reloj como sea necesario, en este caso serán otros 4.

- **Entrada serie/salida paralelo:**

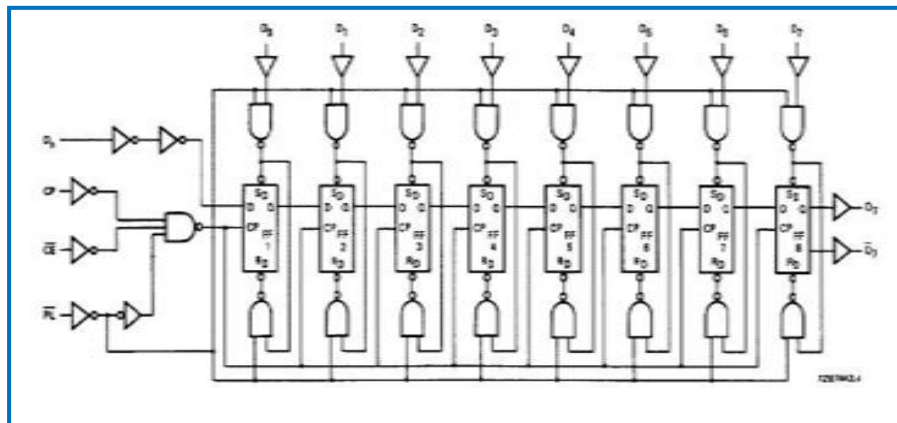
En este tipo de registros con salida en paralelo se dispone de la salida de cada flip-flops por lo que una vez almacenados los datos cada bits se representa en su respectiva salida. De esta manera todos los bits de salida estarán disponibles al mismo tiempo. En este caso mostraremos gráficamente los estados del registro para unos datos de entrada determinados. Mostraremos los cambios fijándonos en la señal de reloj.



- **Entrada paralelo/salida serie:**

En este tipo de registros los bits de datos se introducen simultáneamente a través de líneas paralelo en lugar bit a bit. La salida serie se hace de igual modo que en el primer caso explicado una vez que los datos hayan sido almacenados.

El funcionamiento de un registro de desplazamiento de 8bits con carga en paralelo, utilizaremos como ejemplo.

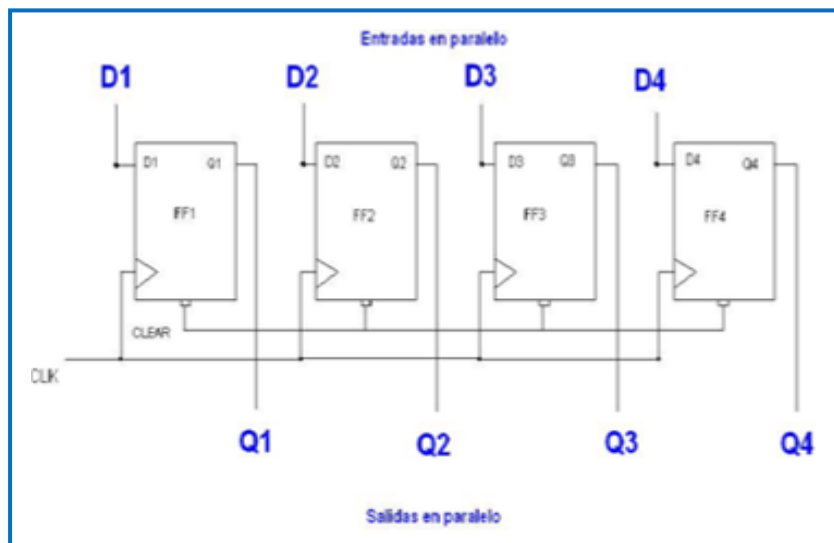


Este dispositivo posee una entrada (pl) que estando a nivel bajo activa todas las puertas NAND permitiendo la carga de los datos de entrada en paralelo.

Cuando a la entrada nos encontramos un “1” el flip-flop correspondiente pasa al estado de SET de manera asíncrona debido al nivel bajo de la puerta superior.

Por el contrario cuando la entrada es “0” el flip-flop pasará a estado de RESET de forma asíncrona por el mismo motivo. Este dispositivo tiene la opción de introducir también los datos en serie a través de la entrada SER. Las salidas de datos serie del registro son Q7 y Q7’.

- **Entrada paralelo/salida paralelo:**



- **Registros de desplazamiento bidireccionales:**

Son aquellos en que los datos se pueden desplazar a la izquierda o la derecha. Se puede implementar utilizando puertas lógicas que permitan la transferencia de un bits de datos de una etapa a la

siguiente de la izquierda o de la derecha dependiendo del nivel de una línea de control.

Posee una entrada que estando a nivel alto hace que los bits almacenados en el registro se desplacen hacia la derecha y a nivel bajo se desplazarán hacia la izquierda.

Cuando este a alto las puertas impares estarán activas y el estado de la salida de cada flip-flop pasará a la entrada D del siguiente. Cuando haya un flanco de reloj los bits se desplazarán una posición a la derecha.

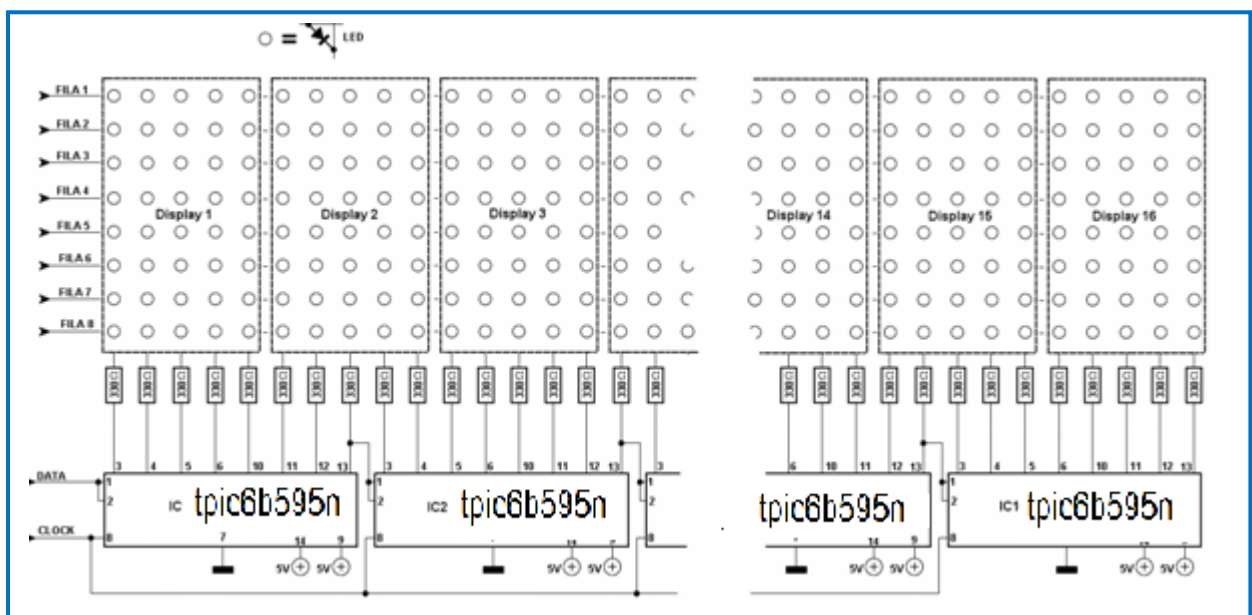
Cuando este a bajo las puertas pares estarán activas y el estado de la salida de cada flip-flop pasará a la entrada D del anterior. Cuando haya un flanco de reloj los bits se desplazarán una posición a la izquierda.

### **Cómo funcionan los registros de Desplazamientos.**

Un registro de desplazamiento funciona de la misma manera en que funciona una cola de gente que espera para entrar en un cine. Por un extremo de la cola van ingresando las personas que llegan, y por el otro van saliendo de la fila. En un registro de desplazamiento, en lugar de personas tenemos “0” y “1”. Lo bueno de esto es que para “meter” datos (“0”s y “1”s) en el registro de desplazamiento solo hacen falta tres pines del microcontrolador, independientemente de lo largo que sea. Estos pines se encargan de tres tareas: Uno de ellos, al que denominaremos **“DATA”** es el encargado de decirle al registro de desplazamiento que lo que introduciremos es un “0” o un “1”. El

segundo se encarga de avisar al registro que el dato ya está listo para ser ingresado, y lo llamaremos “CLOCK”. Y el ultimo, que no es indispensable, es el “RESET”, que se encarga de “vaciar” la fila escribiendo “0”s en todas las salidas del registro.

### Utilizacion de Registro desplazamiento con matriz de Leds



### 3.3 Resistencias



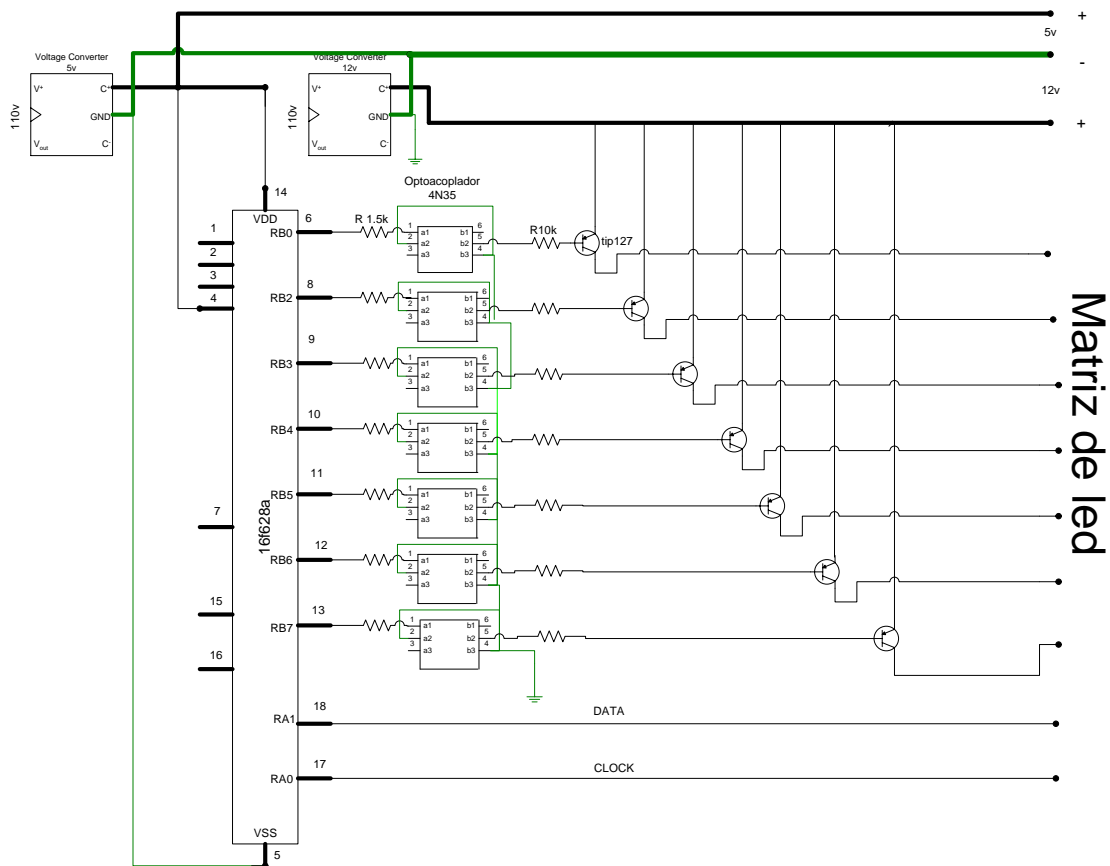
La resistencia es uno de los componentes imprescindibles en la construcción de cualquier equipo electrónico, ya que permite distribuir adecuadamente la corriente y voltaje a todos los puntos necesarios.

El valor de la resistencia se expresa en ohmio, al cual representamos con el símbolo  $\Omega$

Si sometemos los extremos de una resistencia al paso de una corriente continua se producirá en la misma una caída de tensión proporcional a su valor. La intensidad que la atraviese será también proporcional a la tensión aplicada y al valor en ohmios de la resistencia. Para calcular dicha relación no hay mas que aplicar la Ley de Ohm:

$$R = \frac{V}{I}$$

### Circuito de Control



### 3.4 Microcontrolador PIC 16F628A.

Un microcontrolador es como un ordenador en pequeño: dispone de una memoria donde se guardan los programas, una memoria para almacenar datos, dispone de puertos de entrada y salida, etc. A menudo se incluyen puertos seriales (RS-232), conversores analógico/digital, generadores de pulsos PWM para el control de motores, bus I2C, y muchas cosas más. Por supuesto, no tienen ni teclado ni monitor, aunque podemos ver el estado de teclas individuales o utilizar pantallas LCD o LED para mostrar información.

El pic16f628a es un microcontrolador de 8 bit, posee una arquitectura RISC avanzada así como un juego reducido de 35 instrucciones.



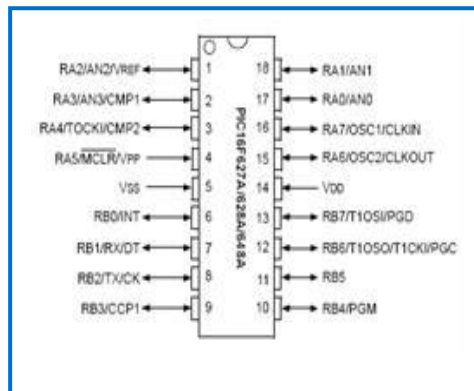
#### **Características:**

- \* Procesador: 8 bits.
- \* Velocidad: 20MHz.
- \* Oscilador interno: 4MHz
- \* Tipo de Memoria: FLASH.
- \* Memoria de Programa: 2k.
- \* EEPROM: 128 bytes.
- \* RAM: 224 bytes.



- \* Entradas/Salidas: 16.
- \* PWM: 1.
- \* Comparadores: 2.
- \* Timers: 2x8bits, 1x16bits.

### Diagrama de conexión.



### 3.5 Transistores



El transistor es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador. El término "transistor" es la contracción en inglés de transfer resistor ("resistencia de transferencia"). Actualmente se los encuentra prácticamente en todos los aparatos domésticos de uso diario: radios, televisores, grabadoras, reproductores de audio y video, hornos de

microondas, lavadoras, automóviles, equipos de refrigeración, alarmas, relojes de cuarzo, ordenadores, calculadoras, impresoras, lámparas fluorescentes, equipos de rayos X, tomógrafos, ecógrafos, reproductores mp3, teléfonos móviles, etc.

### **Características.**

#### ***Características de los Transistores:***

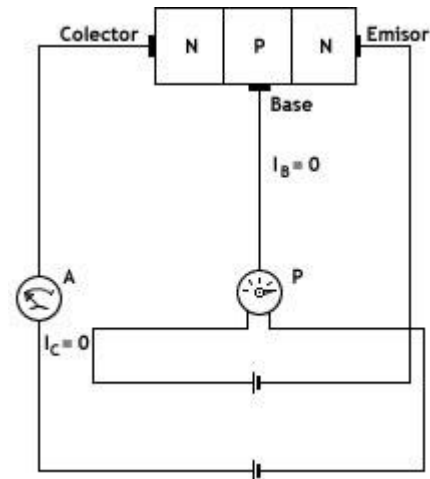
- ❖ El consumo de energía es relativamente bajo.
- ❖ El tamaño de los transistores es relativamente más pequeño que los tubos de vacío.
- ❖ El peso.
- ❖ Una vida larga útil (muchas horas de servicio).
- ❖ Puede permanecer mucho tiempo en depósito (almacenamiento).
- ❖ No necesita tiempo de calentamiento.
- ❖ Resistencia mecánica elevada.
- ❖ Los transistores pueden reproducir el fenómeno de la fotosensibilidad (fenómenos sensibles a la luz).

### **Funcionamiento.**

Para interpretar los esquemas es muy importante saber con detalle el funcionamiento del transistor. Para ello es conveniente ver como se

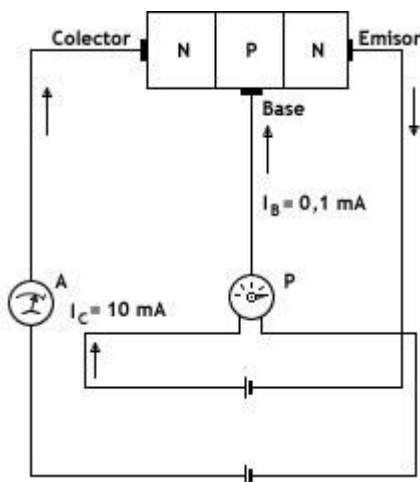
comporta de acuerdo con la corriente de base, que es la principal particularidad de este dispositivo electrónico. Lo analizaremos mejor por medio de imágenes.

En la imagen seguimos con un transistor de tipo NPN, pero sería lo mismo hacer la prueba con el otro tipo de transistor, el PNP, pero habría que hacerlo con las conexiones invertidas para ese caso. En esa imagen va sernos de gran utilidad el potenciómetro (P) que se aprecia en la



parte baja y también el amperímetro (A) que nos indicará el valor de la corriente que circulará por el colector. Aseguramos de que hemos hecho bien las conexiones, es decir, el negativo de la batería al cristal N

emisor, el positivo al colector; y en lo que



respecta a la base con su conexión positiva por ser cristal P. En esa imagen que

vimos tenemos el potenciómetro a cero, de modo que su alta resistencia impide el paso de la corriente a la base y el transistor no conduce corriente. Cuando accionamos el cursor del potenciómetro y

disminuimos la resistencia del circuito, como se ve en la siguiente imagen; dando paso a una intensidad de corriente ( $I_B$ ) de, por ejemplo 0,1 mA, la corriente pasa a alimentar la base y observamos que el

miliamperímetro conectado en serie con el colector mueve su aguja y causa un paso de corriente de 10 mA. Si accionamos el potenciómetro de modo que pase la máxima corriente posible, la aguja del miliamperímetro también delata el aumento del paso de corriente de colector. Entonces deducimos que la corriente de base, cuanto más intensa es, más intensa permite que sea la corriente del colector. De ahí sacamos una importante característica del transistor, y es que se puede regular la corriente de paso por el mismo, por el hecho de establecer una determinada corriente de base. En el ejemplo anterior vimos que con una corriente de 0,1 mA puede controlarse otra corriente de 10 mA, es decir, una corriente  $10/0,1 = 100$  veces superior.

Otra condición de la mayor importancia para conocer para conocer el funcionamiento del transistor son las siguientes reglas que hemos de considerar siempre cuando se trata de interpretar su funcionamiento.

En estos casos:

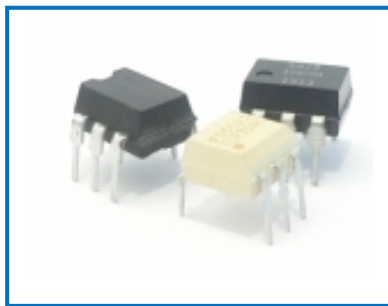
- Al emisor deberá aplicársele una polaridad del mismo signo que el cristal que los constituye. Si el cristal es del tipo P se le deberá aplicar polaridad positiva; y si es del tipo N se le deberá aplicar polaridad negativa.

- A la base se le aplicará igualmente una polaridad del mismo signo que el cristal que lo constituye. Si es un cristal N se le aplicará polaridad negativa; y si es un cristal P deberá ser positiva.

- Al colector se le aplicará una polaridad opuesta al cristal que lo constituye. Si es un cristal P se le deberá aplicar la polaridad negativa; y si es de cristal N deberá aplicársele la polaridad positiva.

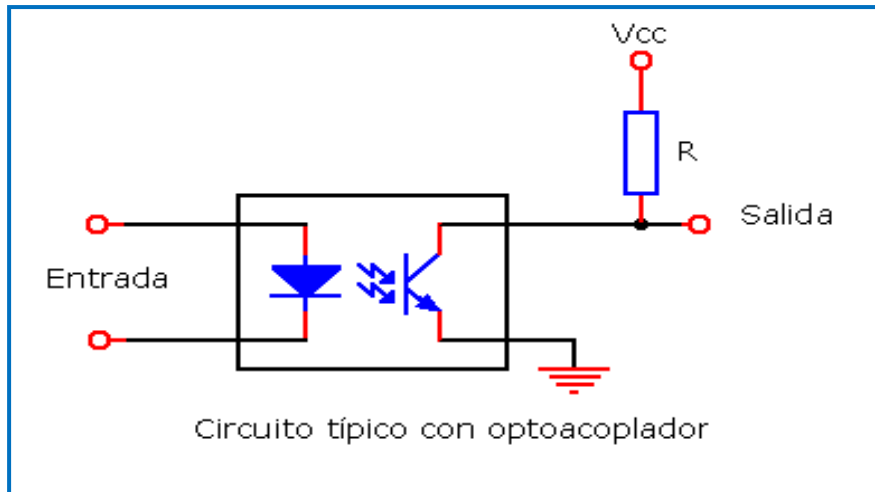
Estas condiciones hay que tenerlas muy en cuenta cada vez que tenga que conectar un transistor en un circuito.

### **3.6 Optoacopladores.**



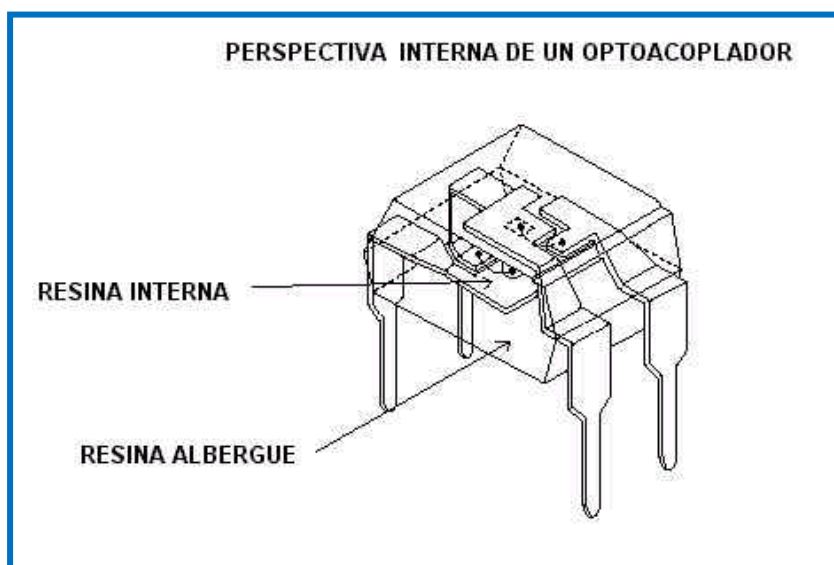
¿Qué son los optoacopladores y cómo funcionan?

Son conocidos como optoaisladores o dispositivos de acoplamiento óptico, basan su funcionamiento en el empleo de un haz de radiación luminosa para pasar señales de un circuito a otro sin conexión eléctrica. Estos son muy útiles cuando se utilizan por ejemplo, Microcontroladores PICs y/o PICAXE y tarjetas electrónicas ,siendo el 4N35 escogido para realizar nuestro proyecto si queremos proteger nuestro micro controlador este dispositivo es una buena opción. En general pueden sustituir los relés ya que tienen una velocidad de conmutación mayor, así como, la ausencia de rebotes.



La gran ventaja de un opto acoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida. Fundamentalmente este dispositivo está formado por una fuente emisora de luz, y un fotosensor de silicio, que se adapta a la sensibilidad espectral del emisor luminoso, todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP.

### **Estructura interna de un optoacoplador**



La figura muestra la perspectiva interna de un optoacoplador. Una resina aloja al elemento sensitivo a la luz (fototransistor o fototransistor de salida Darlington) que esta rodeado por otra resina que permite la transmisión de la luz.

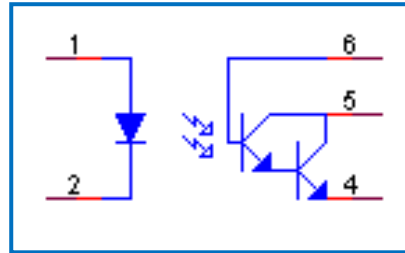
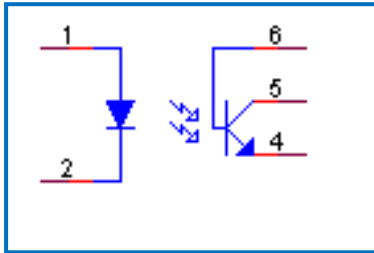
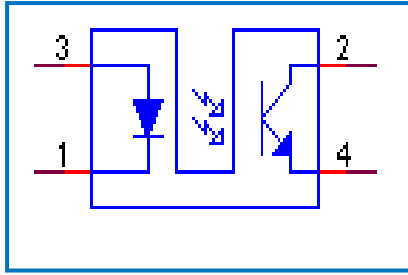
Una señal luminosa es transmitida por un diodo emisor de luz hacia el transistor fotosensitivo a través de la resina transmisora de luz interna. La resina albergue y la resina interior tienen el mismo coeficiente de la expansión.

El alto aislamiento voltaje se obtiene gracias al gran área existente entre la resina externa y la interna que no es modificada por los cambios de temperatura pues los coeficientes de expansión son iguales, además, si la temperatura aumenta las resinas se expanden obteniéndose como resultado una mayor área entre los elementos conductores.

### **¿Qué tipo de Optoacopladores hay?**

Existen varios tipos de optoacopladores cuya diferencia entre sí depende de los dispositivos de salida que se inserten en el componente. Según esto tenemos los siguientes tipos:

Fototransistor: se compone de un optoacoplador con una etapa de salida formada por un transistor



### **Características.**

Existen muchas situaciones en las cuales se necesita transmitir información entre circuitos conmutadores aislados eléctricamente uno del otro. Este aislamiento (aislamiento galvánico) ha sido comúnmente provisto por relés o transformadores de aislamiento. Existen sin embargo en el mercado otros dispositivos capaces de proporcionar el aislamiento requerido, los cuales son muy efectivos para solucionar este tipo de situaciones. Estos dispositivos se llaman optoacopladores, los optoacopladores son más necesarios en situaciones donde desea protección contra altos voltajes y aislamiento de ruidos, así como cuando el tamaño de dispositivo es un factor a considerar. Al realizar un acople entre dos sistemas mediante la transmisión de energía radiante (fotones), se elimina la necesidad de una tierra común, es decir que ambas partes acopladas pueden tener diferente voltajes de



referencia, lo cual constituye la principal ventaja de los optoacopladores. La señal de entrada es aplicada al fotoemisor y la salida es tomada del fotorreceptor. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica.

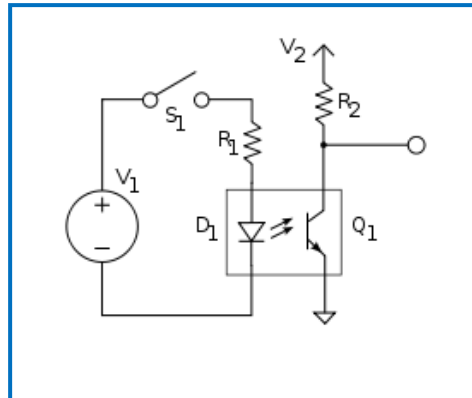
La gran ventaja de un optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

Los fotoemisores que se emplean en los optoacopladores de potencia son diodos que emiten rayos infrarrojos (IRED) y los fotorreceptores pueden ser tiristores o transistores.

Cuando aparece una tensión sobre los terminales del diodo IRED, este emite un haz de rayos infrarrojo que transmite a través de una pequeña guía-ondas de plástico o cristal hacia el fotorreceptor. La energía luminosa que incide sobre el fotorreceptor hace que este genere una tensión eléctrica a su salida. Este responde a las señales de entrada, que podrían ser pulsos de tensión.

Para utilizar completamente las características ofrecidas por un optoacoplador es necesario que el diseñador tenga conocimiento de las mismas. Las diferentes características entre las familias son atribuidas principalmente a la diferencia en la construcción.

## Funcionamiento.

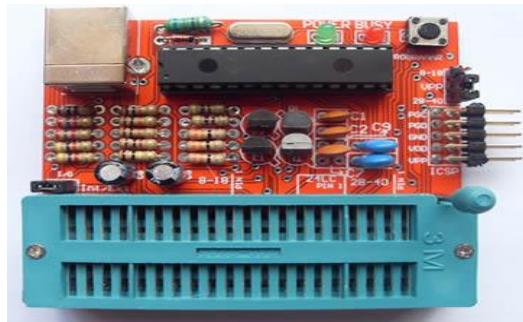


La figura muestra un optoacoplador 4N35 formado por un LED y un fototransistor. La tensión de la fuente de la izquierda y la resistencia en serie establecen una corriente en el LED emisor cuando se cierra el interruptor  $S_1$ . Si dicha corriente proporciona un nivel de luz adecuado, al incidir sobre el fototransistor lo saturará, generando una corriente en  $R_2$ . De este modo la tensión de salida será igual a cero con  $S_1$  cerrado y a  $V_2$  con  $S_1$  abierto.

Si la tensión de entrada varía, la cantidad de luz también lo hará, lo que significa que la tensión de salida cambia de acuerdo con la tensión de entrada. De este modo el dispositivo puede acoplar una señal de entrada con el circuito de salida, aunque hay que tener en cuenta que las curvas tensión/luz del LED no son lineales, por lo que la señal puede distorsionarse. Se venden optoacopladores especiales para este propósito, diseñados de forma que tengan un rango en el que la señal de salida sea casi idéntica a la de entrada.

La ventaja fundamental de un optoacoplador es el aislamiento eléctrico entre los circuitos de entrada y salida. Mediante el optoacoplador, el único contacto entre ambos circuitos es un haz de luz. Esto se traduce en una resistencia de aislamiento entre los dos circuitos del orden de miles de  $M\Omega$ . Estos aislamientos son útiles en aplicaciones de alta tensión en las que los potenciales de los dos circuitos pueden diferir en varios miles de voltios.

### **3.7 Circuito Programador PICS**



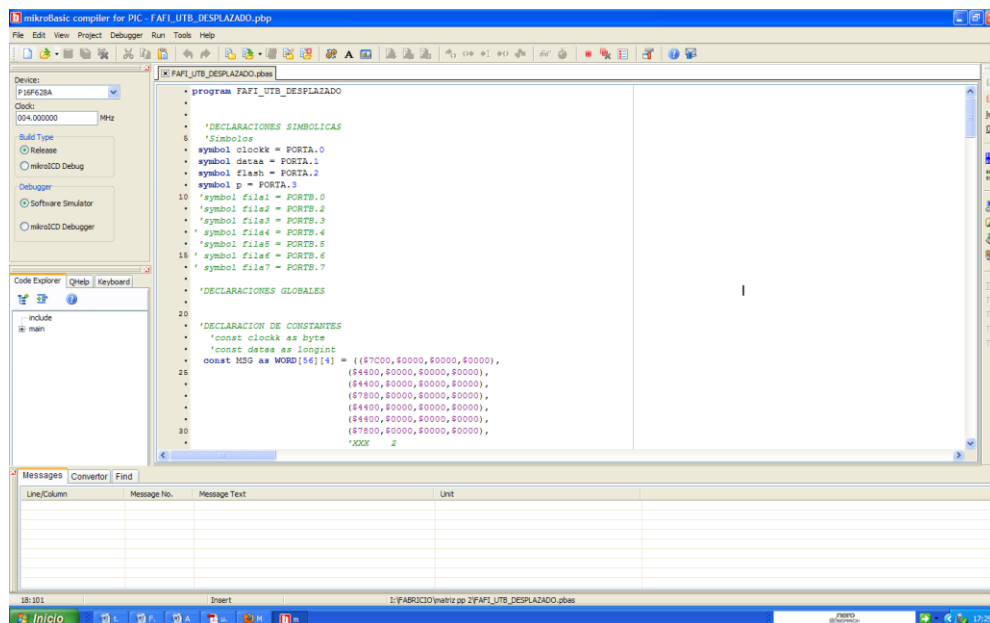
### **3.8 Software y aplicaciones**

Para transferir el código de un ordenador al PIC normalmente se usa un dispositivo llamado programador. La empresa mikroElectrónica distribuye una serie de compiladores para microcontroladores, entre los que se destacan el mikroC y mikroBasic. Las características más destacadas de estos compiladores, y en particular del que nos ocupará en esta serie de artículos es la inclusión de un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment) que hace muy cómoda la programación, ya que resalta la sintaxis del lenguaje, proporciona acceso muy rápido a la excelente ayuda

incluida, estadísticas sobre el uso de recursos del microcontrolador, y muchas ventajas mas.

Además, mikroElectrónica nos permite descargar una versión gratuita del compilador, que a pesar de estar limitado en la generación de código a 2Kb., es más que suficiente para muchos proyectos, y sobre todo, sirve perfectamente para que podamos aprender el lenguaje. Probablemente mikroBasic sea el entorno que soporta mas modelos de micros y además dispone de un enorme grupo de librerías, divididas en comunicaciones RS-232, RS-485 e I2C; teclados PS/2, conexiones USB, interfaz para LCD, y un largísimo etc.

Nos proponemos a lo largo de estos artículos a profundizar en el set de instrucciones de mikroBasic.



Respecto de la organización interna del programa, debemos saber que es necesario que las partes que componen el programa (funciones, rutinas, etc.) sigan ciertas reglas en su escritura. No es necesario que

todas estén presentes. Las secciones que son obligatorias son “program” y el bloque “main-end”. Las demás, opcionales, solo las usaremos si las necesitamos.

```
program include

'*****

Declaraciones (globales, disponibles en todo el programa
'*****

' Constantes
  const ...

' variables
  dim ...

' Simbolos

  symbol ...

' Procedimientos

sub procedure nombre_del_procedimiento(...)
  ...
end sub

' Funciones

sub function Nombre_de_la_funcion(...)

  < declaraciones locales > ...
end sub

'*****

'* Programa principal:
'*****

main: ' Aquí escribimos nuestro código
      end.
```

Ahora, empecemos a ver sus instrucciones más elementales:  
Comentarios: se puede (y debe!) comentar el código usando el apóstrofe “'”. Los comentarios pueden ir solos en una línea, o a continuación de una instrucción cualquiera. No se permiten comentarios de más de una línea:

'Esto es un comentario A = 10 ' y esto también....

Estos comentarios son los que nos permitirán realizar modificaciones o mejoras en nuestro código sin necesidad de perder horas intentando comprender lo que escribimos tiempo atrás.

Identificadores: Los llamados “identificadores” son en realidad los nombres de nuestras variables, procedimientos, funciones, constantes, etc. En mikroBasic los nombres de identificadores pueden contener letras desde la “a” hasta “z” y desde la “A” hasta “Z”, el guión bajo (“\_”) y los dígitos del “0” al “9”. El primer carácter no puede ser un dígito. Además, los nombres de los identificadores no son “case-sensitive”, es decir que “Total”, “total” y “TOTAL” son nombres de la misma variable, y podemos usar cualquiera de ellos por igual.

Correctos:

Temperatura

aux\_12

rele\_activado

Incorrectos:

3boton 'Comienza con un digito

Aux.12 'Contiene un punto\_salto 'Comienza con un gui3n bajo

Humedad% 'Contiene un car3cter invalido (%)

Variables: Al igual que en otros dialectos Basic, las variables son objetos cuyo valor puede cambiar con la ejecuci3n del programa. Cada variable tiene un nombre 3nico, que se ajuste a las reglas definidas mas arriba para los identificadores. Deben declararse con dim antes de ser usadas, y su 3mbito de validez var3a de acuerdo a la secci3n en que se declaran (ver la ayuda para m3s datos sobre esto). La sintaxis de dim es:

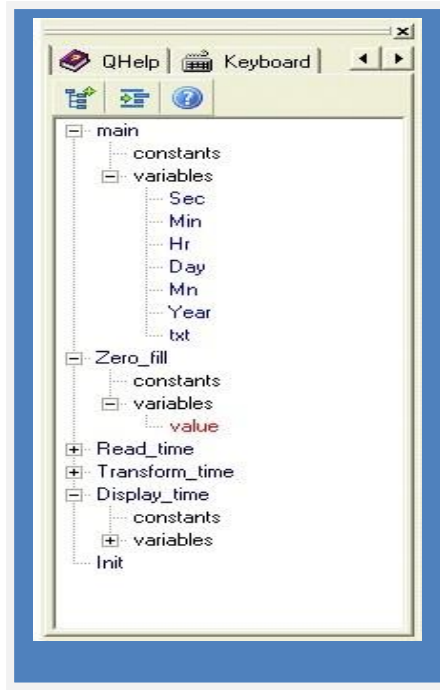
dim lista de variables as tipo.

La lista de tipos disponibles incluyen Byte (8 bits, con valores de 0 a 255), Word (2 bytes, o 16 bits, con valores de 0 a 65535) y Longint (con valores desde -2147483648 a 2147483647). No son los 3nicos, la lista completa de tipos se puede consultar en la documentaci3n del compilador. Algunos ejemplos validos de dim son:

dim i, j, k as byte

dim contar, temp as word

dim cadena as longint[100]



*La opción "code explorer" permite analizar nuestro programa.*

Bucles: Tenemos 3 formas de lograr que un grupo de instrucciones se repitan:

1) FOR – NEXT : Se repite mientras variable sea menor que valor\_final.  
En cada ciclo variable se incrementa en incremento o en 1 si step no esta presente.

```
for variable = valor_inicial to valor_final [step incremento]
```

Instrucciones a repetir

```
next variable
```

Ejemplo:



```
s = 0

for i = 0 to n

s = s + a[i] * b[i]

next i
```

2) WHILE – END: Mientras que la expresión sea verdadera, las instrucciones a repetir se ejecutaran. Si dentro de esas instrucciones no existe alguna que modifique el resultado de expresión, el ciclo se repetirá eternamente. Como la expresión se evalúa al comenzar el ciclo, puede ocurrir que si al comenzar el programa la expresión sea falsa las instrucciones a repetir no se ejecuten nunca.

while expresión Instrucciones a repetir wend.

Ejemplo 1:

```
s = 0
i = 0
while i < n
s = s + a[i] * b[i]
i = i + 1
wend
Ejemplo 2: Un ciclo sin fin.
while TRUE
Instrucciones a repetir
wend
```

3) DO - LOOP: En este caso, las instrucciones a repetir se repiten mientras la expresión sea verdadera. Al evaluarse al final del ciclo, el grupo de instrucciones a repetir se ejecutaran al menos una vez.

do Instrucciones a repetirse loop until expresión

Ejemplo: Producto escalar entre dos vectores.

```
s = 0  
  
i = 0  
  
...  
  
do  
  
s = s + a[i] * b[i]  
  
i = i + 1  
  
loop until i = n
```

Cualquiera es estas tres formas de crear un bucle puede ser interrumpida si dentro de las instrucciones a repetir se coloca la instrucción break. Al ejecutarla, el control del programa salta a la sentencia escrita inmediatamente a continuación de next, wend o loop until, según el caso.

Toma de decisiones: Existen dos instrucciones para la toma de decisiones dentro de un programa escrito en Mikrobasic:

```
else si es falsa.  
  
if expresión then  
  
instrucciones  
  
[else  
  
instrucciones]  
  
end if
```

1) IF – THEN – ELSE – ENDIF: Las instrucciones a continuación del then se ejecutan si la expresión es verdadera, y las que después del

2) SELECT – CASE: Se utiliza para transferir el control a una rama del programa, basándose en una determinada condición. Consiste en una expresión y una lista de sus posibles valores. Las instrucciones a continuación del case else, si existen, se ejecutan en caso de que selector tome un valor para el que no exista una rama case disponible.

```
select case selector  
  
case valor_1  
  
instrucciones_1  
  
...  
case valor_n  
  
instrucciones_n  
[case else  
  
Instrucciones por defecto]  
  
end select
```

Salto: También tenemos dos instrucciones que permiten saltos incondicionales:

1) GOTO: Generalmente desaconsejada, esta instrucción nos permite transferir el control a cualquier punto dentro del programa. A pesar de estar demostrado que es posible construir cualquier programa sin utilizar nunca esta instrucción, se utiliza con cierta frecuencia. El control del programa se transfiere a la línea que tenga la etiqueta utilizada en el goto:

```
for i = 0 to n
  for j = 0 to m
    ...
    if desastre
      goto Error
    end if
    ...
  next j
next i
.
```

Error: ' código para manejar el error. . .

2) GOSUB – RETURN: Funciona de manera similar a GOTO, pero con la ventaja de que el control del programa regresa a la instrucción posterior a GOSUB luego de ser ejecutado el return. Esto la hace ideal para crear subrutinas que pueden ser llamadas varias veces y/o desde distintos puntos del programa

```
gosub mi_rutina  
  
instrucciones  
  
...  
mi_rutina: código de la subrutina  
  
...  
Return
```

### **CODIFICACION FUENTE EJEMPLO**

```
program FAFI_UTB_DESPLAZADO
```

```
'DECLARACIONES SIMBOLICAS
```

```
'Simbolos
```

```
Symbol clock = PORTA.0
```

```
Symbol dataa = PORTA.1
```

```
Symbol flash = PORTA.2
```

```
Symbol p = PORTA.3
```



(\$1044,\$0000,\$0000,\$0000),

(\$1044,\$0000,\$0000,\$0000),

(\$1078,\$0000,\$0000,\$0000),

(\$1044,\$0000,\$0000,\$0000),

(\$1044,\$0000,\$0000,\$0000),

(\$1078,\$0000,\$0000,\$0000),

'XXX 3

(\$447C,\$7C00,\$0000,\$0000),

(\$4410,\$4400,\$0000,\$0000),

(\$4410,\$4400,\$0000,\$0000),

(\$4410,\$7800,\$0000,\$0000),

(\$4410,\$4400,\$0000,\$0000),

(\$4410,\$4400,\$0000,\$0000),

(\$4410,\$7800,\$0000,\$0000),

'XXX 4

(\$0044,\$7C7C,\$0000,\$0000),

(\$0044,\$1044,\$0000,\$0000),

(\$0044,\$1044,\$0000,\$0000),

(\$0044,\$1078,\$0000,\$0000),

(\$0044,\$1044,\$0000,\$0000),

(\$0044,\$1044,\$0000,\$0000),

(\$0044,\$1078,\$0000,\$0000),

'XXX 5

(\$4000,\$447C,\$7C00,\$0000),

(\$4000,\$4410,\$4400,\$0000),

(\$4000,\$4410,\$4400,\$0000),

(\$4000,\$4410,\$7800,\$0000),

(\$4000,\$4410,\$4400,\$0000),

(\$4000,\$4410,\$4400,\$0000),

(\$4000,\$4410,\$7800,\$0000),

'XXX 6

(\$7C40,\$0044,\$7C7C,\$0000),

(\$4040,\$0044,\$1044,\$0000),

(\$4040,\$0044,\$1044,\$0000),

(\$7840,\$0044,\$1078,\$0000),

(\$4040,\$0044,\$1044,\$0000),

(\$4040,\$0044,\$1044,\$0000),

(\$4040,\$0044,\$1078,\$0000),

'XXX 7

(\$7C7C,\$4000,\$447C,\$7C00),

(\$4440,\$4000,\$4410,\$4400),

(\$4440,\$4000,\$4410,\$4400),

(\$7C78,\$4000,\$4410,\$7800),

(\$4440,\$4000,\$4410,\$4400),



(\$4440,\$4000,\$4410,\$4400),  
(\$4440,\$4000,\$4410,\$7800),  
'XXX 8  
(\$7C7C,\$7C40,\$0044,\$7C7C),  
(\$4044,\$4040,\$0044,\$1044),  
(\$4044,\$4040,\$0044,\$1044),  
(\$787C,\$7840,\$0044,\$1078),  
(\$4044,\$4040,\$0044,\$1044),  
(\$4044,\$4040,\$0044,\$1044),  
(\$4044,\$4040,\$007C,\$1078))

## 'DECLARACION DE VARIABLES

'Declaración de variables

Dim zz As word

Dim xx As BYTE

Dim pp As word

Dim col As Byte

Dim aux As word

dim dato15 as word

Dim FF1 As word

Dim CC1 As word

'DECLARACION DE PROCEDIMIENTOS

'sub procedure xx

'end sub

'DECLARACION DE FUNCIONES

'CUERPO DEL PROGRAMA

main:

'Configuro el portA:

TRISA.0 = 0 'Salida CLOCK PIN 17

TRISA.1 = 0 'Salida DATA PIN 18

TRISA.2 = 0 'Salida de reloj flash PIN 1

TRISA.3 = 0 'DATOS ENVIADOS PIN 2

TRISA.4 = 1 'Entrada Dip 2 PIN 3

TRISA.5 = 1 'Entrada Dip 3 PIN 4

'Configuro el portB:

TRISB=\$02

'----- CONFIGURO PUERTOS-----

PORTA = 0

CMCON = 7 'Configuro PORTA como Digital I/O

PORTB = 255 'Apago todas las filas antes de comenzar

'-----BUCLE PRINCIPAL-----'

Loop:

FOR pp=1 to 10

FOR FF1=0 TO 6

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

xx=FF1

Gosub fila

NEXT FF1

next pp

'OTRA 2

FOR pp=1 to 10

FOR FF1=7 TO 13

FOR CC1=0 TO 3

aux=not MSG[FF1][CC1]

Gosub escribo

NEXT CC1

xx=FF1-7

Gosub fila

NEXT FF1

next pp

'OTRA 3

FOR pp=1 to 10

FOR FF1=14 TO 20

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

Gosub fila

NEXT FF1

xx=FF1-14

next pp

'OTRA 4

FOR pp=1 to 10

FOR FF1=21 TO 27

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

Gosub fila

NEXT FF1

xx=FF1-21

next pp

'OTRA 5

FOR pp=1 to 10

FOR FF1=28 TO 34

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

Gosub fila

xx=FF1-28

NEXT FF1

next pp

'OTRA 6 xxxxxxxxxxxxxxxxxxxxxxx

FOR pp=1 to 10

FOR FF1=35 TO 41

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

Gosub fila

NEXT FF1

xx=FF1-35

next pp

'OTRA 7

FOR pp=1 to 10

FOR FF1=42 TO 48

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

Gosub fila

NEXT FF1

xx=FF1-42

next pp

'OTRA 8

FOR pp=1 to 10

```
FOR FF1=49 TO 55
```

```
FOR CC1=0 TO 3
```

```
aux=MSG[FF1][CC1]
```

```
Gosub escribo
```

```
NEXT CC1
```

```
Gosub fila
```

```
xx=FF1-49
```

```
NEXT FF1
```

```
next pp
```

```
goto loop
```

```
'Subrutina que llena el registro de desplazamiento
```

```
escribo:
```

```
For col = 1 To 16
```

```
dato15=aux and %10000000000000000
```

```
If dato15 = %10000000000000000 Then
```

```
dataa = 1
```

```
delay_us (2)
```

```
clockk = 1
```

```
delay_us (2)
```

clockk = 0

Else

dataa = 0

delay\_us (2)

clockk = 1

delay\_us (2)

clockk = 0

End if

flash =1

delay\_us (2)

flash =0

aux = aux << 1 'DESPLASA LOS BITS DE LA VARIABLE aux A LA IZQUIERDA

Next col

Return

'subrrutina que prende la fila

fila:

IF XX =0 then portb=\$1 end if

IF XX =1 then portb=\$4 end if



```
IF XX =2 then portb=$8 end if

IF XX =3 then portb=$10 end if

IF XX =4 then portb=$20 end if

IF XX =5 then portb=$40 end if

IF XX =6 then portb=$80 end if

delay_Ms (1)

PORTB=$00

Return

End.
```

### **3.9 Mensajes Informativos a Presentar en el Letrero Leds**

Como es de suponer, el desarrollo, construcción y programación de un cartel de este tipo es una tarea bastante compleja para un principiante, por lo cual tomamos el reto de hacerlo, ya que te guiaremos paso a paso a lo largo de todo el proceso.

Y seguramente aprenderás un montón de trucos al hacerlo. Por motivos de simplificar el circuito y de no gastar demasiado dinero, nuestro cartel será monocromático, utilizando LEDs de color rojo únicamente. Las dimensiones de la matriz utilizada serán de 7 filas por 64 columnas, aunque físicamente el letrero consta de 2 diodos por cada fila y columna, esto representa (14 filas por 128 columnas) lo que

permite escribir hasta 7 caracteres por pantalla. A pesar de no ser demasiado grande, ya habrás sacado la cuenta de que se necesitan 1792 LEDs para armar el cartel.

A lo largo de este tutorial hemos visto que encender un LED desde un microcontrolador. Y de hecho es algo muy simple: conectamos el ánodo del LED al PIC, el cátodo a una resistencia y el extremo de la resistencia a +V. Cuando el pin del microcontrolador está en “1”, el LED enciende. Este esquema, lamentablemente, no sirve para la construcción de un cartel matricial como este, ya que al disponer de 1792 LEDs necesitaríamos tener un microcontrolador que tenga como mínimo ese número de pines de salida y por supuesto, no existe.

El secreto está en el multiplexado. Es decir, utilizar unos pocos pines de E/S del microcontrolador para manejar una serie de circuitos integrados que se encarguen de excitar los LEDs. Hay varias maneras, y muchos modelos diferentes de circuitos para hacer esto.

Pueden usarse un tipo de integrado digital llamado “LATCH”, que básicamente es una memoria en la que escribimos un valor, y lo mantiene en sus salidas hasta que nosotros lo indiquemos. De esta manera, usando varios latches podríamos encender los LEDs por turnos, rápidamente para que no se note el parpadeo, y de esa manera formar una palabra en el cartel.

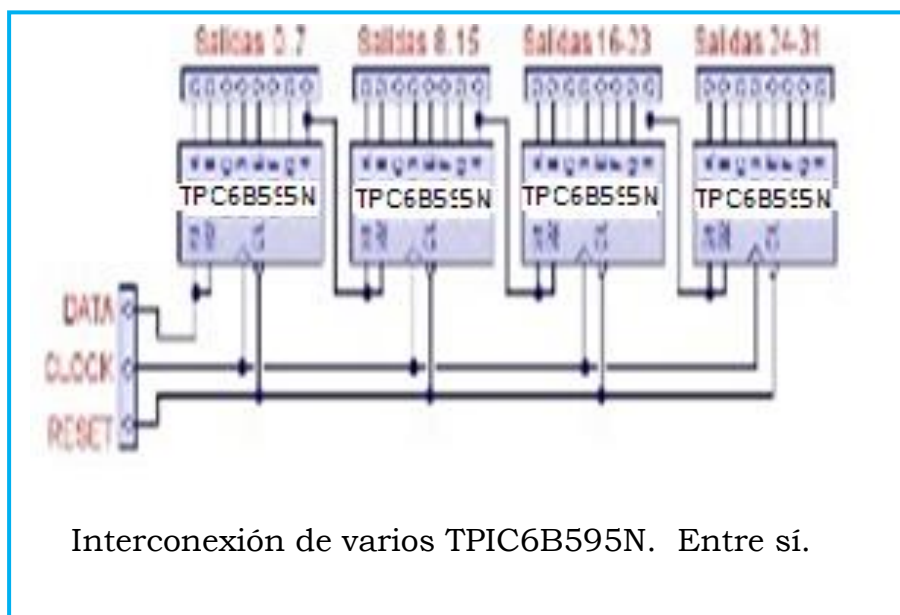
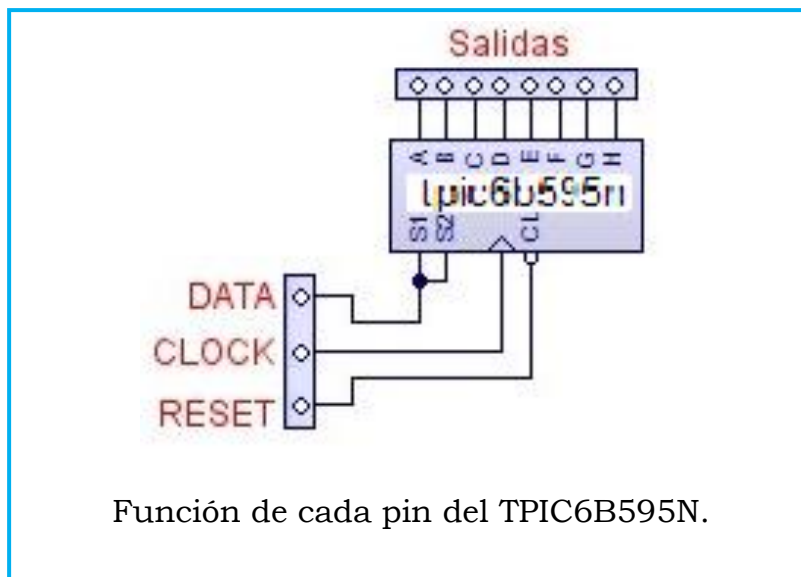
Otra forma es utilizar un registro de desplazamiento. Y de hecho, es de esta forma cómo vamos a diseñar nuestro cartel. Un registro de

desplazamiento funciona de la misma manera en que funciona una cola de gente que espera para entrar en un cine. Por un extremo de la cola van ingresando las personas que llegan, y por el otro van saliendo de la fila. En un registro de desplazamiento, en lugar de personas tenemos “0” y “1”. Lo bueno de esto es que para “meter” datos (“0”s y “1”s) en el registro de desplazamiento solo hacen falta tres pines del microcontrolador, independientemente de lo largo que sea.

Estos pines se encargan de tres tareas: Uno de ellos, al que denominaremos “DATA” es el encargado de decirle al registro de desplazamiento que lo que introduciremos es un “0” o un “1”. El segundo se encarga de avisar al registro que el dato ya está listo para ser ingresado, y lo llamaremos “CLOCK”. Y el último, que no es indispensable, es el “RESET”, que se encarga de “vaciar” la fila escribiendo “0”s en todas las salidas del registro.

En este proyecto utilizaremos un modelo de circuito integrado conocido como TPIC6B595N, que es un registro de desplazamiento de 8 bits. Es decir, con el se puede armar una “fila” de 7 “personas”. Como nuestro cartel tiene 64 columnas, necesitaremos utilizar 8 de estos integrados, uno a continuación del otro.

En la figura siguiente puedes ver la función de cada pin de este integrado, y la manera de conectar uno a continuación del otro para obtener un registro de desplazamiento de cualquier longitud.



Bien, con el esquema explicado podemos encender los LEDs que queramos de una fila de 80 bits de largo. Si en el registro de desplazamiento introducimos “11111...111”, los 80 LEDs estarán encendidos. Si queremos encender uno por medio, escribiremos “10101...01”. Por supuesto, cuando llegemos a la parte de la

programación veremos cómo se ingresan uno a uno los “0” y “1” en el registro.

Ahora bien: nuestro cartel tiene 7 filas, y lo explicado solo sirve para manejar una de ellas.

¿Debemos utilizar un registro de desplazamiento para cada una de las filas restantes? Afortunadamente, la respuesta es no.

Si bien podríamos utilizar 7 registros de este tipo, la cantidad de circuitos integrados necesarios (56 de ellos), la complejidad del circuito impreso y el costo implicado lo hacen poco aconsejable.

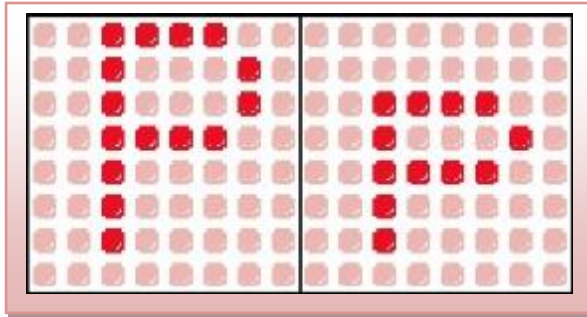
Nosotros aprovecharemos un “defecto” del ojo humano, que mantiene la imagen vista durante unos 20 o 30 milisegundos, para “dibujar” una fila a la vez, pero muy rápidamente, de forma que todo el cartel parezca estar encendido a la vez. Si, como en la tele o el cine.

Cuando comenzamos el artículo mencionábamos que para manejar cada LED serían necesarios 560 pines de entrada/salida.

Con el esquema propuesto solo necesitamos 7 para seleccionar la fila a escribir, y tres para manejar el registro de desplazamiento. Es decir, un PIC de 3 u\$ y 18 pines serviría perfectamente para realizar el proyecto. Y es lo que usaremos, un 16F628A.



El resultado de este proyecto es muy vistoso.



Se pueden tener diferentes tipos de letras.

#### 4. ¿Cómo funciona la matriz?

Como dijimos antes, la pantalla está formada por una serie de filas y columnas. La intersección entre ambas contiene un LED. Para que este encienda, tiene que recibir simultáneamente un “0” en la fila, y un “1” en la columna. Cuando se dan estas condiciones, la electrónica de la placa se encarga del encendido. La forma de generar un mensaje sobre el display es relativamente sencilla, si nos atenemos al siguiente algoritmo:

- 1) Apagar todas las filas.
- 2) Escribir los valores correspondientes a la primer fila en el registro de desplazamiento, teniendo en cuenta que el primer dígito binario colocado corresponde al último LED de la fila, y el último en poner al de la primer columna.
- 3) Encenderla primer fila, esperar un tiempo, y volver a apagarla.
- 4) Repetir los pasos para las filas 2 a 7.

Los tiempos de demora que utilizamos en el programa de ejemplo permiten una visualización correcta, sin molestos parpadeos y con los LEDs brillantes. Hay que tener en cuenta que si utilizamos tiempos mayores para el encendido de cada fila, el brillo de los LEDs será

mayor, pero también aumentara el parpadeo. No utilizamos vectores ni otras alternativas que hubieran servido para crear un código más compacto, buscando la claridad del programa, para que pueda servir como base a otros más completos/complejos.

Un punto a tener en cuenta es el brillo de los LEDs. Un LED, utilizado en aplicaciones “normales”, se alimenta con unos 3V y requiere unos 15mA (varia ligeramente de un modelo a otro<sup>9</sup> para brillar con una buena intensidad. En nuestro caso, a pesar de que veremos las 7 filas encendidas al mismo tiempo, cada LED solo estará encendido la séptima parte del tiempo, por lo que su brillo será siete veces inferior al normal, y nuestro cartel apenas será visible.

Afortunadamente eso tiene solución: dado que los tiempos que permanecerá encendido cada LED no superara unos pocos milisegundos, no se dañaran si hacemos circular una corriente mayor a la nominal, con brillará mucho más intensamente, dando como resultado un cartel perfectamente visible.

Respecto de los LEDs, podremos utilizar LEDs discretos (y soldar 1120 terminales) o comprar “paneles” de 7x5 LEDs que tienen unos 14 o 16 terminales (según el modelo), estando ya interconectados en forma de matriz.

Hemos dividido este proyecto en varias partes, ya que su complejidad impide explicarlo en un solo artículo, así que en la próxima entrega diseñaremos la totalidad del hardware necesario, y en la siguiente nos dedicaremos a programar nuestro cartel. ¡Hasta la próxima!



Nuestro cartel tiene fines meramente educativos, y la intención es gastar poco dinero para construirlo, así que intentaremos realizarlo en base a un microcontrolador pequeño. Si el lector necesita un cartel de mayor tamaño o con capacidad para almacenar textos o imágenes más extensos, deberá utilizar algún micro con mayor capacidad y velocidad. El tamaño de la memoria **EEPROM** externa es bastante grande, pero también puede ser ampliado con mucha facilidad.



Dividiremos el esquema electrónico del cartel en dos partes: en primer lugar veremos toda la **lógica de control**, y en segundo, la “pantalla” con el **registro de desplazamiento**. A la hora de llevarlo a la práctica puede incluso hacer dos circuitos impresos por separado. Esto le permitirá al lector experimentar con otros controladores sin necesidad de volver a montar la placa de los displays.

#### **4.1 El circuito controlador**

Este es el cerebro de nuestro cartel. Será el encargado de gestionar el encendido de cada **LED** mediante órdenes enviadas a las columnas (mediante el registro de desplazamiento que mencionamos en la nota anterior) y a las filas.

Como una fila tendrá 128 LEDs, que eventualmente pueden estar todos encendidos, no podemos conectarlas directamente a pines de E/S del **PIC**, por que la corriente que demandarían haría que el puerto del PIC se destruya. Para evitar esto, utilizaremos en medio un **transistor** capaz de manejar esa corriente. Analicemos el circuito. El centro de todo es el microcontrolador **16F628A**.

Para este proyecto utilizaremos el reloj interno del 16f628a. Todo el puerto B del PIC está dedicado a controlar las filas del cartel. Como ya habrán notado, tenemos 8 salidas para filas, y nuestro cartel tiene solo 7 filas. Efectivamente, la fila 8 no se utilizara si nuestra “pantalla” está construida con una matriz, de 7x64 .

Por último, los pines 17 y 18, correspondientes a los bits A0 y A1 del micro se encargan de la gestión del registro de desplazamiento. El programa deberá generar los pulsos de reloj necesarios por el pin 18, y “meter” los datos en el registro por el pin 17.

Hemos incluido una **fuentes de alimentación**. Cualquier fuente comercial que sea capaz de entregar 5V/1A y 12 V /5A será suficiente. Esos 5V deben estar bien regulados, y por supuesto, el software deberá estar escrito correctamente, es decir, no encender varias filas al mismo tiempo, para de esta forma apreciar bien el mensaje.

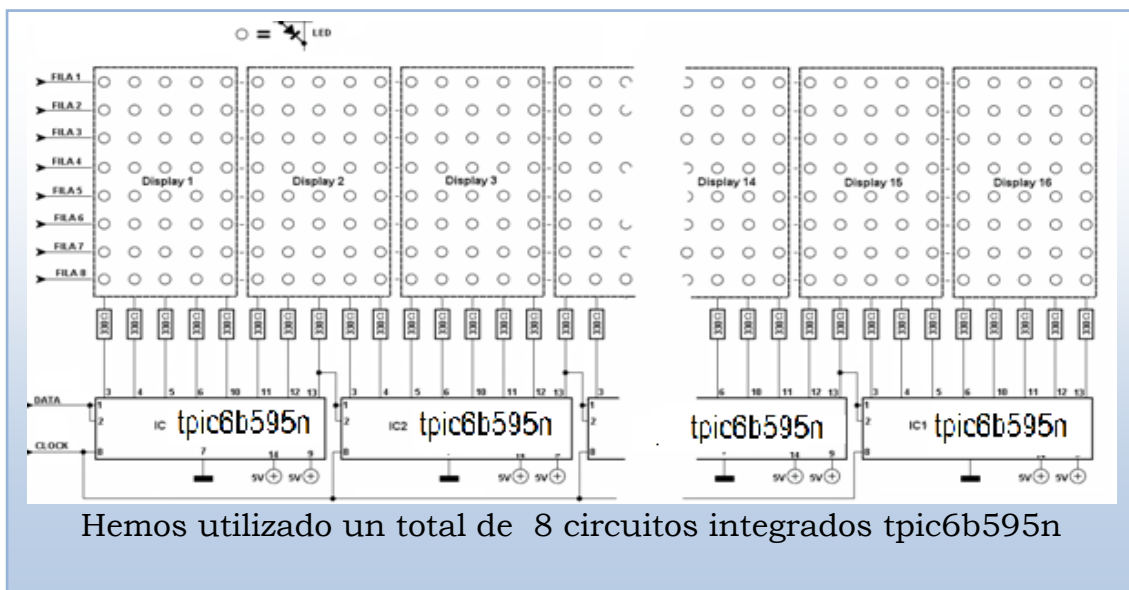
## **4.2 El display**

Esta es la parte del proyecto que todo el mundo va a mirar, así que debemos ser prolijos al montarlo. Como puede verse en el esquema eléctrico, hemos utilizado un total de 8 circuitos integrados TPIC6B595N para construir el registro de desplazamiento de 64 bits de largo, uno para cada columna. Si alguien quiere hacer un cartel más largo o más corto, deberá poner más o menos integrados. Cada uno maneja 8 columnas.

En el dibujo del circuito que hemos tomado como referencia podemos observar las conexiones entre el registro y una matriz de 8 por 5 dejando en claro que la conexión con nuestra matriz de 14 por 128 las conexiones son similares por cuestión de espacio hemos utilizado este grafico para que tengan una idea de la conexión entre los registros y los led.

No utilizaremos el pin de **RESET** de los TPIC6B595N. En lugar de ser controlados desde el microcontrolador, cada RESET está puesto a +5V, de forma que el integrado funcione continuamente. Si por algún motivo se desea borrar la pantalla, basta con enviar 80 "0"s al registro de desplazamiento y listo. El tiempo empleado para esa tarea es despreciable, ya que el microcontrolador estará ejecutando 1 millón de instrucciones por segundo. El utilizar una línea de control menos nos permitirá tener un PCB ligeramente más sencillo.

Cada salida de los TPIC6B595N, como dijimos, se conecta a una columna de la serie de displays. Esta conexión se efectúa mediante un **resistor** de 1/8 de watt, que en el esquema se ha dibujado con un valor de 330 ohm. Ese fue el valor adecuado para el tipo de módulos que conseguimos para hacer el prototipo, pero su valor variara de un modulo a otro. Se puede montar solo un display con resistores de 330 ohms, y ver como es el brillo de los LEDs. Si es escaso, se puede bajar el valor a 220 o 100 ohms. Con eso debería ser suficiente.





Los carteles de este tipo se utilizan cada día más.

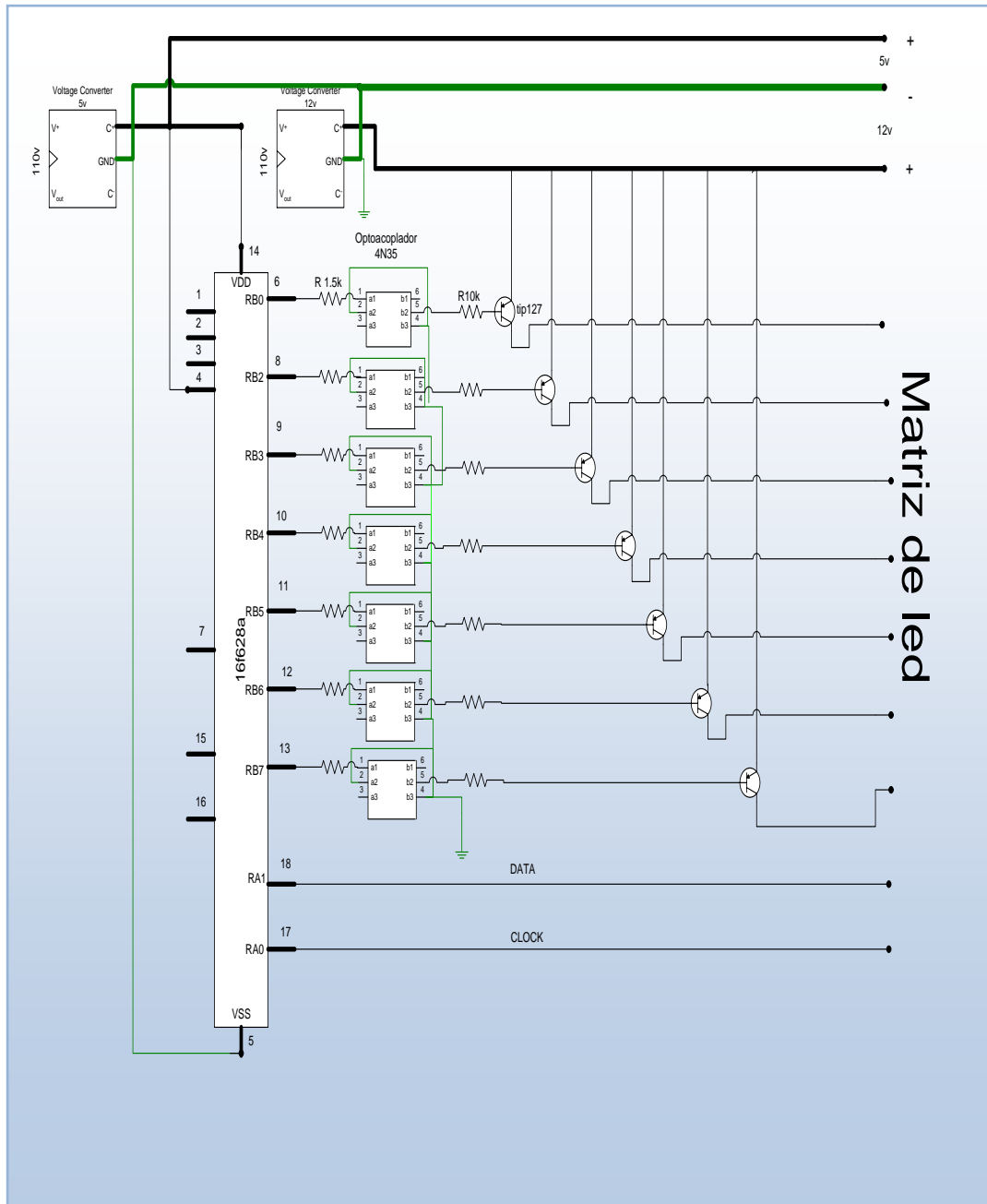


### 4.3 Procedimientos

#### Materiales

<b>1 microcontrolador PIC16F628A, Con su zócalo.</b>	<b>\$3,50</b>
<b>6 circuito integrado TPIC6B595N, Con su zócalo.</b>	<b>\$30</b>
<b>1 fuente regulada de voltaje 5V.</b>	<b>\$25</b>
<b>1 fuente regulada de voltaje 12V.</b>	<b>\$25</b>
<b>7 opto acopladores 4N35.</b>	<b>\$14</b>
<b>7 transistores TIP 127.</b>	<b>\$3,50</b>
<b>7 resistencias de 1k.</b>	<b>\$1,40</b>
<b>7 resistencias de 1,5k.</b>	<b>\$1,40</b>
<b>7 resistencias de 10k.</b>	<b>\$1,40</b>
<b>3 borneras conectoras.</b>	<b>\$3,00</b>
<b>1 Matriz de diodos leds de 14 x 128.</b>	<b>\$400</b>
<b>1 Placa de Baquelita</b>	<b>\$40</b>
<b>1 rollito de Estaño</b>	<b>\$4</b>
<b>1 pasta para Soldar</b>	<b>\$2</b>
<b>Total</b>	<b>\$554,20</b>

#### 4.4 Diagrama de Circuito de control



## 4.5 Recursos Técnicos

### HARDWARE

<b>Cantidad</b>	<b>Característica</b>
1 computador	procesador Pentium IV  intel 3.2 ghz  Disco duro 120gb  Memoria ram DDR 512 mb,  tarjeta de video 32 mb.  Puerto paralelo, puerto serial, puerto usb, monitor Samsung "17",  CD write 52x32x52  Fax modem 56 kpps  Tarjeta red10/100 tarjeta de serial  Mouse PS-2,teclado

### Software

- mikroBasic Compiler for PIC

### Recursos Materiales

Memory Flash de 4GB

## **CONCLUSIONES**

1. El Letrero Led Programable está concebido para ser utilizado como medio de comunicación visual.
2. Los diodos Led usados en el letrero tienen un tiempo de vida útil prolongado y por su bajo consumo de energía permitirá ahorrar en gastos de electricidad.
3. Este letrero es de fácil transportación y conexión ya que su tamaño y peso están dentro de un rango que cualquier persona lo puede manipular.
4. La realización de esta tesis ofrecerá investigar bibliografía para la investigación de futuras tesis de este tipo.

## **RECOMENDACIONES**

1. Se debe de utilizar una fuente de alimentación de corriente de 110 V para el letrero.
2. Los mensajes a programarse deben ser programados primeramente en el pic.
3. Para el mejoramiento de este letrero se podrá usar una memoria EEPROM que permitirá su ampliación en el manejo de información.



## **BIBLIOGRAFÍA**

### **DIRECCIONES DE INTERNET**

- <http://es.wikipedia.org/wiki/Wikipedia:Portada>
- <http://html.rincondelvago.com/concepto-de-base-de-datos.html>
- <http://ola.icmyl.unam.mx/biblio/Tesis-Bus.php>
- <http://www.programacion.com/foros/5/msg/25774/>
- [http://www.fvet.uba.ar/biblioteca/como\\_hacer.htm](http://www.fvet.uba.ar/biblioteca/como_hacer.htm)
- <http://www.programacion.com/php/>
- <http://www.br.uipr.edu/caiweb/default2.asp?tree=625>
- <http://www.asesoriatesis.com/cgi-bin/default.php>

## EJEMPLO DE PROGRAMAS A MOSTRARSE EN EL LETRERO

### PROGRAM DIBUJOS Y FIGURAS

'DECLARACIONES SIMBOLICAS

'Simbolos

symbol clockk = PORTA.0

symbol dataa = PORTA.1

symbol flash = PORTA.2

symbol p = PORTA.3

'symbol fila1 = PORTB.0

'symbol fila2 = PORTB.2

'symbol fila3 = PORTB.3

' symbol fila4 = PORTB.4

'symbol fila5 = PORTB.5

' symbol fila6 = PORTB.6

' symbol fila7 = PORTB.7

'DECLARACIONES GLOBALES

'DECLARACION DE CONSTANTES

'const clockk as byte

'const dataa as longint

const MSG as WORD[42][4] = ((\$3800,\$7C00,\$0000,\$0000),

(\$4400,\$4010,\$0000,\$0004),

(\$4400,\$4010,\$3870,\$3804),

(\$FE00,\$4010,\$4440,\$443C),

(\$FE00,\$4010,\$4430,\$7C44),

(\$FE00,\$4010,\$4408,\$4044),

(\$FE00,\$7C10,\$3870,\$3C3C),

'XXX 2

(\$0003,\$8038,\$787C,\$4200), '7

(\$0004,\$4044,\$4440,\$4200),

(\$0004,\$4044,\$4440,\$6200),

(\$00FE,\$0044,\$7878,\$5200),

(\$00FE,\$0044,\$4040,\$4A00),

(\$00FE,\$0044,\$4040,\$4600),  
(\$00FE,\$0038,\$407C,\$4200),  
'XXX 3  
(\$007E,\$3FE3,\$FE08,\$0000), '14  
(\$0666,\$3FE3,\$FE04,\$1C1C),  
(\$07E6,\$38E3,\$8E02,\$2A36),  
(\$07FF,\$F8FF,\$8FFF,\$2422),  
(\$07FE,\$3FE3,\$FE02,\$2236),  
(\$06F6,\$3563,\$5604,\$1C2A),  
(\$039C,\$1DC1,\$DC08,\$0000),  
'XXX 4  
(\$1C1C,\$4410,\$7C22,\$7C44), '21  
(\$2222,\$6C10,\$4024,\$4044),  
(\$23E2,\$5410,\$4028,\$4028),  
(\$1C1C,\$4410,\$4030,\$7810),  
(\$0410,\$4410,\$4028,\$4010),  
(\$0410,\$4410,\$4024,\$4010),  
(\$03E0,\$4410,\$7C22,\$7C10),  
'XXX 5  
(\$0020,\$3C00,\$0000,\$0000), '28  
(\$0010,\$4038,\$1010,\$0438),  
(\$0008,\$4004,\$1000,\$0404),  
(\$3FFC,\$383C,\$1010,\$3C3C),  
(\$0008,\$0444,\$1010,\$4444),  
(\$0010,\$0444,\$1010,\$4444),  
(\$0020,\$783C,\$1010,\$3C3C),  
'XXX 6  
(\$0000,\$007F,\$FE00,\$0000), '35  
(\$0000,\$0081,\$0100,\$0000),  
(\$0000,\$3FFF,\$FFE0,\$0000),  
(\$0000,\$2000,\$0020,\$0000),  
(\$0000,\$2200,\$2020,\$0000),  
(\$0000,\$3DFF,\$DFE0,\$0000),  
(\$0000,\$0200,\$2000,\$0000))

'DECLARACION DE VARIABLES

'Declaracion de variables

Dim zz As word

Dim xx As BYTE

Dim pp As word

Dim col As Byte

Dim aux As word

dim dato15 as word

Dim FF1 As word

Dim CC1 As word

'DECLARACION DE PROCEDIMIENTOS

'sub procedure xx

'end sub

'DECLARACION DE FUNCIONES

'CUERPO DEL PROGRAMA

main:

'Configuro el portA:

TRISA.0 = 0 'Salida CLOCK PIN 17

TRISA.1 = 0 'Salida DATA PIN 18

TRISA.2 = 0 'Salida de reloj flash PIN 1

TRISA.3 = 0 'DATOS ENVIADOS PIN 2

TRISA.4 = 1 'Entrada Dip 2 PIN 3

TRISA.5 = 1 'Entrada Dip 3 PIN 4

'Configuro el portB:

TRISB=\$02

'----- CONFIGURO PUERTOS-----

PORTA = 0

CMCON = 7 'Configuro PORTA como Digital I/O

PORTB = 255 'Apago todas las filas antes de comenzar

'-----BUCLE PRINCIPAL-----

```
Loop:
FOR pp=1 to 90
  FOR FF1=0 TO 6
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1
    Gosub fila
  NEXT FF1
```

```
next pp
```

```
'OTRA 2
```

```
  FOR pp=1 to 90
  FOR FF1=7 TO 13
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-7
    Gosub fila
  NEXT FF1
```

```
next pp
```

```
'OTRA 3
```

```
FOR pp=1 to 90
  FOR FF1=14 TO 20
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-14
    Gosub fila
  NEXT FF1
```

```
next pp
```

```
'OTRA 4
```

```
FOR pp=1 to 90
  FOR FF1=21 TO 27
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-21
    Gosub fila
  NEXT FF1
```

next pp

'OTRA 5

```
FOR pp=1 to 90
  FOR FF1=28 TO 34
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-28
    Gosub fila
```

```
NEXT FF1
```

next pp

'OTRA 6

```
FOR pp=1 to 90
  FOR FF1=35 TO 41
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-35
    Gosub fila
  NEXT FF1
```

next pp

```
goto loop
```

```
'Subrutina que llena el registro de desplazamiento  
escribo:
```

```
For col = 1 To 16
```

```
dato15=aux and %10000000000000000
```

```
    If dato15 = %10000000000000000 Then
```

```
        dataa = 1
```

```
        delay_us (2)
```

```
        clockk = 1
```

```
        delay_us (2)
```

```
        clockk = 0
```

```
    Else
```

```
        dataa = 0
```

```
        delay_us (2)
```

```
        clockk = 1
```

```
        delay_us (2)
```

```
        clockk = 0
```

```
    End if
```

```
    flash =1
```

```
    delay_us (2)
```

```
    flash =0
```

```
    aux = aux << 1 'DESPLASA LOS BITS DE LA VARIABLE aux A LA IZ-  
QUIERDA
```

```
Next col
```

```
Return
```

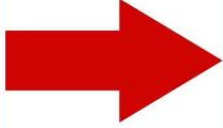
```
'subrutina que prende la fila
```

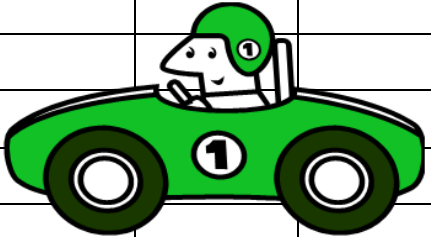
```
fila:
```


```
IF XX =0 then portb=$1 end if
IF XX =1 then portb=$4 end if
IF XX =2 then portb=$8 end if
IF XX =3 then portb=$10 end if
IF XX =4 then portb=$20 end if
IF XX =5 then portb=$40 end if
IF XX =6 then portb=$80 end if
delay_Ms (1)
PORTB=$00
Return
```


End.

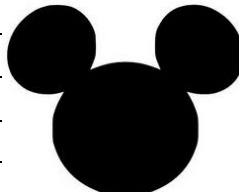


<b>CODIGO BINARIO PARA VARIAS IMÁGENES</b>							
							
		<b>S</b>	<b>a</b>	<b>l</b>	<b>i</b>	<b>d</b>	<b>a</b>
0020		3C	00	00	00	00	00
0010		40	38	10	10	04	38
0008		40	24	10	00	04	24
3FFC		38	3C	10	10	3C	3C
0008		04	44	10	10	44	44
0010		04	44	10	10	44	44
0020		78	3C	10	10	3C	3C

			0000	007F	FE00	0000
			0000	0081	0100	0000
			0000	3FFF	FFE0	0000
			0000	2000	0020	0000
			0000	2200	2020	0000
			0000	3DFF	DFE0	0000
			0000	0200	2000	0000

	<b>C</b>	<b>L</b>	<b>O</b>	<b>S</b>	<b>E</b>	<b>D</b>
38	7C	00	00	00	00	00
44	40	00	00	00	00	04
44	40	10	38	70	38	04
FE	40	10	44	40	44	3C
FE	40	10	44	30	7C	44
FE	40	10	44	08	40	44
FE	7C	10	38	70	3C	3C

					
		<b>O</b>	<b>P</b>	<b>E</b>	<b>N</b>
0380	38	78	7C	42	
0440	44	44	40	42	
0440	44	44	40	62	
FE	44	78	78	52	
FE	44	40	40	4A	
FE	44	40	40	46	
FE	38	40	7C	42	

								
			<b>M</b>	<b>I</b>	<b>C</b>	<b>K</b>	<b>E</b>	<b>Y</b>
1C1C	44	10	7C	22	7C	44		
2222	6C	10	44	24	40	44		
23E2	54	10	44	28	40	28		
1C1C	44	10	44	30	78	10		
0410	44	10	44	28	40	10		
0410	44	10	44	24	40	10		
03E0	44	10	7C	22	7C	10		

## PROGRAM ECUADOR\_MIP AIS

'DECLARACIONES SIMBOLICAS

'Simbolos

symbol clockk = PORTA.0

symbol dataa = PORTA.1

symbol flash = PORTA.2

symbol p = PORTA.3

'symbol fila1 = PORTB.0

'symbol fila2 = PORTB.2

'symbol fila3 = PORTB.3

' symbol fila4 = PORTB.4

'symbol fila5 = PORTB.5

' symbol fila6 = PORTB.6

' symbol fila7 = PORTB.7

'DECLARACIONES GLOBALES

'DECLARACION DE CONSTANTES

'const clockk as byte

'const dataa as longint

const MSG as WORD[42][4] = ((\$07C0,\$0000,\$0000,\$0780),

(\$0400,\$0000,\$0000,\$0440),

(\$0400,\$0000,\$0000,\$0440),

(\$0780,\$0000,\$0000,\$0780),

(\$0400,\$0000,\$0000,\$0500),

(\$0400,\$0000,\$0000,\$0480),

(\$07C0,\$0000,\$0000,\$0440),

'XXX 2

(\$07C7,\$C000,\$0003,\$8780),

(\$0404,\$0000,\$0004,\$4440),

(\$0404,\$0000,\$0004,\$4440),

(\$0784,\$0000,\$0004,\$4780),

(\$0404,\$0000,\$0004,\$4500),

(\$0404,\$0000,\$0004,\$4480),

(\$07C7,\$C000,\$0003,\$8440),

'XXX 3

(\$07C7,\$C440,\$0783,\$8780),

(\$0404,\$0440,\$0444,\$4440),

(\$0404,\$0440,\$0444,\$4440),

(\$0784,\$0440,\$0444,\$4780),

(\$0404,\$0440,\$0444,\$4500),

(\$0404,\$0440,\$0444,\$4480),

(\$07C7,\$C380,\$0783,\$8440),

'XXX 4

(\$07C7,\$C447,\$C783,\$8780),

(\$0404,\$0444,\$4444,\$4440),

(\$0404,\$0444,\$4444,\$4440),

(\$0784,\$0447,\$C444,\$4780),

(\$0404,\$0444,\$4444,\$4500),

(\$0404,\$0444,\$4444,\$4480),

(\$07C7,\$C384,\$4783,\$8440),

'XXX 5

(\$0000,\$0044,\$1000,\$0000),

(\$0000,\$006C,\$1000,\$0000),

(\$0000,\$0054,\$1000,\$0000),

(\$0000,\$0044,\$1000,\$0000),

(\$0000,\$0044,\$1000,\$0000),

(\$0000,\$0044,\$1000,\$0000),

(\$0000,\$0044,\$1000,\$0000),

'XXX 6

(\$0000,\$787C,\$103C,\$0000),

(\$0000,\$4444,\$1040,\$0000),

(\$0000,\$4444,\$1040,\$0000),

(\$0000,\$787C,\$1038,\$0000),

```
($0000,$4044,$1004,$0000),  
($0000,$4044,$1004,$0000),  
($0000,$4044,$1078,$0000))
```

#### 'DECLARACION DE VARIABLES

'Declaracion de variables

Dim zz As word

Dim xx As BYTE

Dim pp As word

Dim col As Byte

Dim aux As word

dim dato15 as word

Dim FF1 As word

Dim CC1 As word

#### 'DECLARACION DE PROCEDIMIENTOS

'sub procedure xx

'end sub

#### 'DECLARACION DE FUNCIONES

#### 'CUERPO DEL PROGRAMA

main:

'Configuro el portA:

TRISA.0 = 0 'Salida CLOCK PIN 17

TRISA.1 = 0 'Salida DATA PIN 18

TRISA.2 = 0 'Salida de reloj flash PIN 1

TRISA.3 = 0 'DATOS ENVIADOS PIN 2

TRISA.4 = 1 'Entrada Dip 2 PIN 3

TRISA.5 = 1 'Entrada Dip 3 PIN 4

'Configuro el portB:

TRISB=\$02

'----- CONFIGURO PUERTOS-----

PORTA = 0

CMCON = 7 'Configuro PORTA como Digital I/O

PORTB = 255 'Apago todas las filas antes de comenzar

'-----BUCLE PRINCIPAL-----

Loop:

FOR pp=1 to 5

FOR FF1=0 TO 6

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

xx=FF1

Gosub fila

NEXT FF1

next pp

'OTRA 2

FOR pp=1 to 5

FOR FF1=7 TO 13

FOR CC1=0 TO 3

aux=MSG[FF1][CC1]

Gosub escribo

NEXT CC1

xx=FF1-7

Gosub fila

NEXT FF1

next pp

'OTRA 3

FOR pp=1 to 5

FOR FF1=14 TO 20

```
FOR CC1=0 TO 3
  aux=MSG[FF1][CC1]
  Gosub escribo
NEXT CC1
xx=FF1-14
Gosub fila
NEXT FF1
```

next pp

'OTRA 4

```
FOR pp=1 to 90
  FOR FF1=21 TO 27
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-21
    Gosub fila
  NEXT FF1
```

next pp

'OTRA 5

```
FOR pp=1 to 90
  FOR FF1=28 TO 34
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-28
    Gosub fila
```

```
NEXT FF1
```

next pp

```
'OTRA 6
FOR pp=1 to 90
  FOR FF1=35 TO 41
    FOR CC1=0 TO 3
      aux=MSG[FF1][CC1]
      Gosub escribo
    NEXT CC1
    xx=FF1-35
    Gosub fila
  NEXT FF1
```

```
next pp
```

```
goto loop
```

'Subrutina que llena el registro de desplazamiento  
escribo:

```
For col = 1 To 16
  dato15=aux and %10000000000000000
```

```
  If dato15 = %10000000000000000 Then
```

```
    dataa = 1
    delay_us (2)
    clockk = 1
    delay_us (2)
    clockk = 0
```

```
  Else
```

```
    dataa = 0
    delay_us (2)
    clockk = 1
    delay_us (2)
```



clockk = 0

End if

flash =1

delay\_us (2)

flash =0

aux = aux << 1 'DESPLASA LOS BITS DE LA VARIABLE aux A  
LA IZQUIERDA

Next col

Return

'subrrutina que prende la fila

fila:

IF XX =0 then portb=\$1 end if

IF XX =1 then portb=\$4 end if

IF XX =2 then portb=\$8 end if

IF XX =3 then portb=\$10 end if

IF XX =4 then portb=\$20 end if

IF XX =5 then portb=\$40 end if

IF XX =6 then portb=\$80 end if

delay\_Ms (1)

PORTB=\$00

Return

End.

## CODIGO BINARIO DE FRASES

<b>CODIGO BINARIO DE FRASES</b>						
<b>E</b>	<b>C</b>	<b>U</b>	<b>A</b>	<b>D</b>	<b>O</b>	<b>R</b>
7C	7C	44	7C	78	38	7800
40	40	44	44	44	44	4400
40	40	44	44	44	44	4400
78	40	44	7C	44	44	7800
40	40	44	44	44	44	5000
40	40	44	44	44	44	4800
7C	7C	38	44	78	38	4400
		<b>M</b>	<b>I</b>			
00	00	44	10	00	00	0000
00	00	6C	10	00	00	0000
00	00	54	10	00	00	0000
00	00	44	10	00	00	0000
00	00	44	10	00	00	0000
00	00	44	10	00	00	0000
00	00	44	10	00	00	0000
		<b>P</b>	<b>A</b>	<b>I</b>	<b>S</b>	
00	00	78	7C	10	3C	0000
00	00	44	44	10	40	0000
00	00	44	44	10	40	0000
00	00	78	7C	10	38	0000
00	00	40	44	10	04	0000
00	00	40	44	10	04	0000
00	00	40	44	10	78	0000

## CODIGO BINARIO ALFABETO MINUSCULA

CODIGO BINARIO ALFABETO MINUSCULA						
a	b	c	d	e	f	g
00	00	00	00	00	00	00
38	40	00	04	00	78	38
04	40	3C	04	38	40	44
3C	78	40	3C	44	70	44
44	44	40	44	7C	40	3C
44	44	40	44	40	40	04
3C	78	3C	3C	3C	40	78
h	i	j	k	l	m	n
00	00	00	00	00	00	00
40	10	08	00	00	00	00
40	00	00	40	10	80	80
40	10	08	50	10	EE	E0
7C	10	08	60	10	92	90
44	10	08	60	10	92	90
44	10	70	50	10	92	90
o	p	q	r	s	t	u
00	00	00	00	00	00	00
00	00	1C	00	00	00	00
38	70	24	50	70	20	48
44	48	24	60	40	70	48
44	70	1C	40	30	20	48
44	40	04	40	08	20	48
38	40	04	40	70	18	34
v	w	x	y	z		
00	00	00	00	00		
44	00	00	88	00		
44	44	88	88	F8		
44	44	50	50	10		
44	54	20	20	20		
28	6C	50	20	40		
10	44	88	20	F8		

## CODIGO BINARIO ALFABETO MAYUSCULA

A	B	C	D	E	F	G
10	78	7C	78	7C	7C	7C
28	44	40	44	40	40	40
44	44	40	44	40	40	40
44	78	40	44	78	78	7C
7C	44	40	44	40	40	44
44	44	40	44	40	40	44
44	78	7C	78	7C	40	7C
H	I	J	K	L	M	N
44	10	3E	22	40	44	42
44	10	08	24	40	6C	42
44	10	08	28	40	54	62
7C	10	08	30	40	44	52
44	10	08	28	40	44	4A
44	10	48	24	40	44	46
44	10	30	22	7C	44	42
O	P	Q	R	S	T	U
38	78	38	78	3C	7C	44
44	44	44	44	40	10	44
44	44	44	44	40	10	44
44	78	44	78	38	10	44
44	40	54	50	04	10	44
44	40	4C	48	04	10	44
38	40	3C	44	78	10	38
V	W	X	Y	Z		
44	44	44	44	7E		
44	44	44	44	04		
44	44	28	28	08		
44	44	10	10	10		
44	54	28	10	20		
28	6C	44	10	40		
10	44	44	10	FE		

## CODIGO BINARIOS DE NUMEROS

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
38	18	78	78	04	7C	3C	78	38	7C
44	28	04	04	0C	40	40	08	44	44
44	48	04	04	14	40	40	08	44	44
44	08	38	38	3E	7C	78	1C	38	7C
44	08	40	04	04	04	44	08	44	04
44	08	40	04	04	04	44	08	44	04
38	1C	3C	78	04	7C	78	08	38	04