

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD TÉCNICA
CARRERA ELECTRÓNICA Y TELECOMUNICACIONES



EXAMEN DE GRADO
TRABAJO DE APLICACIÓN
NIVEL LICENCIATURA

“LETRERO ELECTRÓNICO LED PARA TRANSPORTE PÚBLICO”

Postulante: FLORES VILLCA RUBÉN

DICIEMBRE 2012
LA PAZ - BOLIVIA

Dedicatoria

El presente trabajo de aplicación va dedicado a la memoria de mi madre, a mi esposa e hijos quienes me brindaron su apoyo y cariño. A mi padre, hermanos y sobrinos. A todos ellos se los dedico.

Agradecimientos

*A todo el plantel docente de la carrera de Electrónica y
Telecomunicaciones por transmitirme sabiamente sus conocimientos
a lo largo de mi vida universitaria.*

ÍNDICE

I. RESUMEN DEL TRABAJO	pg. 1
II. GENERALIDADES.....	pg. 2
II.1. PLANTEAMIENTO DEL PROBLEMA	pg. 2
II.2. JUSTIFICACIÓN DEL TRABAJO.....	pg. 2
II.2.1. JUSTIFICACIÓN TÉCNICA	pg. 2
II.2.2. JUSTIFICACIÓN ECONÓMICA	pg. 2
III. OBJETIVO GENERAL	pg. 3
III.1. OBJETIVOS ESPECÍFICOS.....	pg. 3

CAPITULO I

1. FUNDAMENTO TEÓRICO.....	pg. 4
1.1. SISTEMA VISUAL HUMANO.....	pg. 4
1.1.1. PERSISTENCIA RETINIANA.....	pg. 4
1.2. FUNCIONAMIENTO Y CARACTERÍSTICAS DE LOS DIODOS EMISORES DE LUZ.....	pg. 5
1.3. MICROCONTROLADORES	pg. 7
1.3.1. INTRODUCCIÓN A LOS MICROCONTROLADORES.....	pg. 7
1.3.2. CARACTERÍSTICAS PRINCIPALES.....	pg. 8
1.3.3. CARACTERÍSTICAS SEGÚN MODELO	pg. 9
1.3.4. DIFERENCIA ENTRE MICROCONTROLADOR Y MICROPROCESADOR	pg. 10
1.4. MICROCONTROLADORES PIC 16F877A	pg. 10
1.5. LENGUAJE DE PROGRAMACIÓN MIKROBASIC PRO	pg. 11
1.6. MATRIZ DE LED'S	pg. 12

CAPITULO II

2. INGENIERÍA DEL PROYECTO.....	pg. 19
2.1. DESARROLLO DEL PROYECTO.....	pg. 19
2.2. DISEÑO DEL HARDWARE DE CONTROL	pg. 22
2.3. DISEÑO DEL SOFTWARE DE CONTROL.....	pg. 25
2.4. DISEÑO DE PLACAS DE CIRCUITO IMPRESO	pg. 29

CAPITULO III

3. COSTOS.....	pg. 32
CONCLUSIONES	pg. 32
RECOMENDACIÓN.....	pg. 33
BIBLIOGRAFÍA.....	pg. 33
ANEXOS.....	pg. 34

“LETRERO ELECTRÓNICO DE LED’S PARA TRANSPORTE PUBLICO”

I. RESUMEN DEL TRABAJO

En la actualidad el transporte público, en especial los minibuses de la ciudad de La Paz, utilizan varios letreros para anunciar su recorrido, a nuestro entender esta situación constituye una desventaja visual para el conductor, proponemos que sería más efectiva la utilización de un solo letrero conformado por una matriz de diodos emisores de luz (Led’s) que tenga la capacidad de mostrar el recorrido con un simple manejo por parte del conductor. Por ello el presente trabajo de aplicación explica el diseño, construcción e implementación de un prototipo para un letrero luminoso con Led’s.

En una aproximación del contenido general del trabajo, ahora nos permitimos dar algunas explicaciones preliminares:

En el primer capítulo se exponen las bases fundamentales fisiológicas de la visión humana que permite ver el funcionamiento de una matriz de Led’s, como cartel luminoso, también se describen los principales dispositivos electrónicos utilizados para la realización de este proyecto, como ser la matriz de Led’s y el micro controlador de la familia PIC 16F877A, también los circuitos de control.

En el segundo capítulo se explica el diseño del proyecto, el diseño del hardware que incluye el circuito de control y el diseño del software que describe el programa que controla todo el circuito previamente descrito, así también el diseño de las placas de circuito impreso para la implantación del circuito.

En el tercer capítulo se describe los costos económicos para el desarrollo del proyecto. Finalmente establecemos nuestras conclusiones y recomendación.

II. GENERALIDADES

II.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad el desarrollo de la tecnología nos permite brindar a la sociedad soluciones que antes podrían ser costosas en cuanto a la importación de carteles luminosos en base a Led's, el diseño y construcción de estos carteles no es de coste elevado en comparación a los importados y pueden dar un servicio más dirigido a un sector que así lo requiera, que para este caso es el sector del transporte público.

De esta situación se extrae que el problema es el brindar a la sociedad la producción nacional de estos carteles luminosos dirigidos específicamente a movilidades de transporte público, a un costo por debajo de la producción de importación y diseño específicamente para el requerimiento del cliente.

III. JUSTIFICACIÓN DEL TRABAJO

III.1. JUSTIFICACIÓN TÉCNICA

El diseño del circuito está basado en una matriz de Led's en la cual se desplegaran la cantidad de mensajes necesarios para cada uno de los distintos requerimientos del servicio de transporte público, la matriz para este efecto tendrá un tamaño de 8 por 64, es decir 8 filas por 64 columnas, contando con 512 Led's y una dimensión aproximada de 1 metro de largo por 12 cm de alto, este diseño por lo tanto es de la dimensión suficiente para ser optimo al requerimiento del grupo al cual está dirigido.

III.2. JUSTIFICACIÓN ECONÓMICA

El costo del proyecto concluido es menor al de un letrero luminoso importado, así también el mantenimiento y soporte técnico, por lo cual justifica su desarrollo

brindando a la sociedad de una producción a medida y de menor coste, además de ser de producción nacional.

IV. OBJETIVO GENERAL

Diseñar, construir e implementar un letrero electrónico Led's para transporte público.

IV.I. OBJETIVOS ESPECÍFICOS

- Realizar el diseño del hardware para el control de matrices de Led's.
- Efectuar el diseño del Software para el control del circuito de control
- Diseñar el circuito impreso de las distintas partes del hardware.
- Plasmar el acabado e implementación del proyecto.

CAPITULO I

1. FUNDAMENTO TEÓRICO

1.1. SISTEMA VISUAL HUMANO

La visión como uno de nuestros cinco sentidos es capaz de brindarnos información de lo que nos rodea, siendo tan útil para nuestro desarrollo diario posee ciertas limitaciones, limitaciones que aprovecharemos para propósitos de este proyecto. La visión humana no puede registrar movimientos muy rápidos, a cierta velocidad la información en la vista digamos que no se actualiza quedando grabada la imagen en la memoria del cerebro, nos valdremos de esta peculiaridad para el funcionamiento del proyecto, he aquí una explicación más detallada del fenómeno.

1.1.1. PERSISTENCIA RETINIANA

La persistencia de la visión es un fenómeno visual descubierto por Joseph Plateau¹ que demuestra como una imagen permanece en la retina humana una décima de segundo antes de desaparecer por completo. Esto permite que veamos la realidad como una secuencia de imágenes ininterrumpidas y que podamos calcular fácilmente la velocidad y dirección de un objeto que se desplaza, si no existiese, veríamos pasar la realidad como sucesión de imágenes independientes y estáticas. Plateau descubrió que nuestros ojos ven con una cadencia de 10 imágenes por segundo. En virtud de dicho fenómeno, las imágenes se superponen en la retina y el cerebro las "enlaza" como una sola imagen visual, móvil y continua. El cine aprovecha este efecto y provoca ese "enlace" proyectando a más de 10 imágenes por segundo (generalmente 24), lo que genera en nuestro cerebro la ilusión de movimiento (en televisión se dan 25 fotogramas por segundo).

¹ Joseph-Antoine Ferdinand Plateau (Bruselas, Bélgica, 14 de octubre de 1801 – Gante, Bélgica, 15 de septiembre de 1883) fue un físico belga que definió en 1829 el principio de la persistencia de la visión. En 1832 inventó el fenaquistiscopio, uno de los precursores del cinematógrafo.

Durante la época victoriana², algunos juguetes ópticos fueron utilizados para mostrar diversas ilusiones y, en algunos casos, se les esgrimía como factores probatorios de diversas teorías de la visión, entre ellas la persistencia retiniana.

Fueron populares en la época victoriana, un ejemplo conocido es el traumatropo con dibujo de pájaro y jaula: se dibujaba un pájaro en una cara de una hoja y una jaula en el otro lado. Se pegaba una varilla atravesando la hoja. Al girar rápidamente la hoja frotando una varilla se veía la imagen de un pájaro enjaulado. Otro ejemplo es el zoótrofo³.

1.2. FUNCIONAMIENTO Y CARACTERÍSTICAS DE LOS DIODOS EMISORES DE LUZ

Cuando un Led⁴ se encuentra en polarización directa, los electrones pueden recombinarse con los huecos en el dispositivo, liberando energía en forma de fotones. Este efecto es llamado electroluminiscencia y el color de la luz (correspondiente a la energía del fotón) se determina a partir de la banda de energía del semiconductor.

Por lo general, el área de un led es muy pequeña (menor a 1 mm²), y se pueden usar componentes ópticos integrados para formar su patrón de radiación. Los Led's

² La época victoriana de Gran Bretaña marcó la cúspide de la Revolución industrial británica y el ápice del Imperio Británico. Se refiere al periodo del reinado de la reina Victoria del 20 de junio de 1837 al 22 de enero de 1901.

³ Zoótrofo. Denominado en inglés *zoetrope* o *daedalum*, máquina estroboscópica creada en 1834 por William George Horner, compuesta por un tambor circular con unos cortes, a través de los cuales mira el espectador para que los dibujos dispuestos en tiras sobre el tambor, al girar, den la ilusión de movimiento. Fue un juguete muy popular en la época y uno de los avances hacia la aparición del cine que se crearon en la primera mitad del siglo XIX.

⁴ El primer led fue desarrollado en 1927 por Oleg Vladimírovich Lósev (1903-1942); sin embargo no se usó en la industria hasta la década de 1960. Solo se podían construir de color rojo, verde y amarillo, con poca intensidad de luz y se limitaba su utilización a mandos a distancia (controles remotos) y electrodomésticos, como indicadores para señalar el encendido y apagado. A finales del siglo XX se inventaron los Led's ultravioletas y azules, lo que dio paso al desarrollo del led blanco, que es un led de luz azul con recubrimiento de fósforo que produce una luz amarilla, la mezcla del azul y el amarillo (colores complementarios en el espectro RGB) produce una luz blanquecina denominada «luz de luna» consiguiendo alta luminosidad (7 lúmenes unidad) con lo cual se ha ampliado su utilización en sistemas de iluminación.

presentan muchas ventajas sobre las fuentes de luz incandescente y fluorescente, principalmente con un consumo de energía mucho menor, mayor tiempo de vida, tamaño más pequeño, gran durabilidad, resistencia a las vibraciones, no es frágil.

Reducen considerablemente la emisión de calor que produce el efecto invernadero en nuestro planeta, no contienen mercurio (el cual al exponerse en el medio ambiente es altamente venenoso) a comparación de la tecnología fluorescente, ni crean campos magnéticos altos como la tecnología de inducción magnética con los cuales se crea mayor radiación residual hacia el ser humano, cuentan con mejor Índice de reproducción cromática que otros tipos de luminarias, reducen ruidos en las líneas eléctricas.

Son especiales para utilizarse con sistemas foto voltaicos (paneles solares) a comparación de cualquier otra tecnología actual, no les afecta el encendido intermitente (es decir pueden funcionar como luces estroboscópicas) y esto no reduce su vida promedio, son especiales para sistemas anti-exploración ya que no es fácil quebrar un diodo emisor de luz y, en la mayoría de los colores (a excepción de los Led's azules) cuentan con una muy alta fiabilidad y duración.

Los Led's con la potencia suficiente para la iluminación de interiores son relativamente caros y requieren una corriente eléctrica más precisa, por su sistema electrónico para funcionar con voltaje alterno y requieren de disipadores de calor cada vez más eficientes a comparación de las bombillas fluorescentes de potencia equiparable.

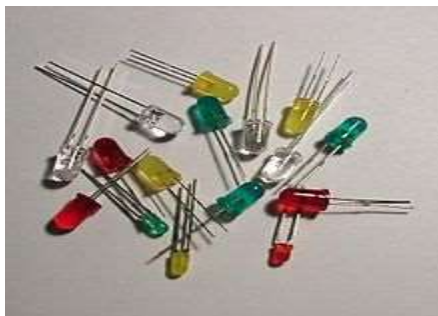


Figura 1: Diodos emisores de luz (Led's)

1.3. MICROCONTROLADORES

1.3.1. INTRODUCCIÓN A LOS MICROCONTROLADORES

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.



Figura 2: Microcontrolador PIC 16F877A

Al ser fabricados, la memoria ROM del microcontrolador no posee datos. Para que pueda controlar algún proceso es necesario generarlo o crearlo y luego grabarlo en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

Existe una amplia variedad de microcontroladores, para el presente proyecto se utilizará el microcontrolador de la familia PIC 16f87x más precisamente el 16F877A, que se describe a continuación.

1.3.2. CARACTERÍSTICAS PRINCIPALES

Las características principales de estos dispositivos son:

1. CPU de arquitectura RISC (*Reduced Instruction Set Computer*).
2. Set de 35 instrucciones.
3. Frecuencia de reloj de hasta 20MHz (ciclo de instrucción de 200ns).
4. Todas las instrucciones se ejecutan en un único ciclo de instrucción, excepto las de salto.
5. Hasta 8K x 14 palabras de Memoria de Programa FLASH (ver tabla a continuación).
6. Hasta 368 x 8 bytes de Memoria de Datos tipo RAM (ver tabla a continuación).
7. Hasta 256 x 8 bytes de Memoria de Datos tipo EEPROM (ver tabla a continuación).
8. Hasta 15 fuentes de Interrupción posibles.
9. 8 niveles de profundidad en la Pila hardware.
10. Modo de bajo consumo (Sleep).
11. Tipo de oscilador seleccionable (RC, HS, XT, LP y externo).
12. Rango de voltaje de operación desde 2,0V a 5,5V.
13. Conversor Analógico/Digital de 10 bits multicanal.
14. 3 Temporizadores
15. Watchdog Timer o Perro Guardián.
16. 2 módulos de captura/comparación/PWM.
17. Comunicaciones por interfaz USART (Universal SynchronousAsynchronous Receiver Transmitter).
18. Puerto Paralelo Esclavo de 8 bits (PSP).
19. Puerto Serie Síncrono (SSP) con SPI e I²C.

1.3.3. CARACTERÍSTICAS SEGÚN MODELO

La familia de los PIC16F87X cuenta con un set reducido de 35 instrucciones, lo que facilita enormemente su memorización. La mayoría de estas instrucciones se ejecutan en un solo ciclo de instrucción, mientras que aquellas instrucciones que realizan saltos de ejecución lo hacen en dos ciclos de instrucción. Cada una de estas instrucciones se codifica en una palabra de 14 bits única e inequívoca, que conforman en conjunto, lo que se conoce como código máquina del dispositivo.

Aunque podría emplearse directamente este código máquina para programar el dispositivo (ya fuera en formato binario o en hexadecimal), resultaría demasiado lento y complicado. Por esta razón, cada una de las instrucciones se representa de manera equivalente con un nemónico, que es más fácil de recordar y de interpretar por el programador. A este tipo de lenguaje de programación se le llama lenguaje ensamblador.

Actualmente, para la programación de microcontroladores puede emplearse tanto el lenguaje ensamblador como lenguajes de más alto nivel, tipo C, Basic o Pascal, empleando el compilador adecuado.

Los microcontroladores tienen un espacio de memoria que varía según el modelo y este espacio de memoria es limitado. Hay que tener en cuenta que a cada instrucción escrita en lenguaje de alto nivel le corresponderán por lo general varias instrucciones en lenguaje ensamblador, lo que significa que un programa escrito en lenguaje de alto nivel ocupa más espacio de memoria que uno escrito en lenguaje ensamblador. Por este motivo, la elección de uno u otro tipo de lenguaje a la hora de programar deberá hacerse teniendo en cuenta el espacio de memoria disponible.

En caso de utilizar un lenguaje de alto nivel, siempre se requerirá compilar el programa terminado para obtener el archivo hexadecimal (lenguaje máquina) con el cual se programará el microcontrolador.

La programación física del dispositivo se hará normalmente mediante un programador, capaz de grabar el programa de usuario en la memoria del microcontrolador. Para el diseño de este proyecto se utilizara el compilador de Mikro Basic Pro.

1.3.4. DIFERENCIA ENTRE MICROCONTROLADOR Y MICROPROCESADOR

Un Microprocesador no es un ordenador completo. No contiene grandes cantidades de memoria ni es capaz de comunicarse con dispositivos de entrada -como un teclado, un *joystick* o un ratón- o dispositivos de salida como un monitor o una impresora.

Un Microcontrolador es un tipo diferente de circuito integrado llamado microcontrolador es de hecho una computadora completa situada en un único chip, que contiene todos los elementos del microprocesador básico además de otras funciones especializadas. Los microcontroladores se emplean en videojuegos, reproductores de vídeo, automóviles y otras máquinas.

1.4. MICROCONTROLADORES PIC 16F877A

El microcontrolador PIC16F877 forma parte de una subfamilia de microcontroladores PIC (*Peripheral Interface Controller*) de gama media de 8 bits, fabricados por Microchip Technology Inc.

Cuentan con memoria de programa de tipo EEPROM Flash mejorada, lo que permite programarlos fácilmente usando un dispositivo programador de PIC. Esta

característica facilita sustancialmente el diseño de proyectos, minimizando el tiempo empleado en programar los microcontroladores.

La "A" final del PIC16F877A indica que estos modelos cuentan con módulos de comparación analógicos. El hecho de que se clasifiquen como microcontroladores (MCU) de 8 bits hace referencia a la longitud de los datos que manejan las instrucciones, y que se corresponde con el tamaño del bus de datos y el de los registros de la C.P.U.

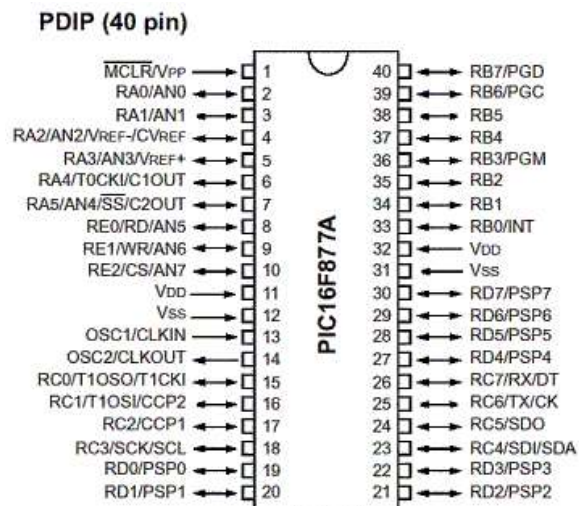


Figura 3: Descripción de pines del PIC16F877A

1.5. LENGUAJE DE PROGRAMACIÓN MIKROBASIC PRO

MikroElektronika es una empresa con base en Belgrado, Serbia, su producción es la de tarjetas entrenadoras de microcontroladores de distintas gamas y distintos fabricantes, accesorios para sus tarjetas entrenadoras, compiladores y libros sobre el tema.



Figura 4: Logotipo mikroElectrónica

De entre estos compiladores existe el MikroBasic un lenguaje de alto nivel con el cual se facilita mucho la tarea de programación ya que al ser un compilador de alto nivel

que utiliza el lenguaje Basic libera al programador de las complejidades del hardware, esto debido a que posee librerías que están diseñadas para controlar dispositivos como pantallas LCD, teclados matriciales, pantallas táctiles, memorias SD, y los distintos componentes del microcontrolador, que en este caso es el seleccionado PIC.



Figura 5: Libro de Lenguaje de Programación

1.6. MATRIZ DE LED'S

Se conoce también como arreglo matricial de Led's que posee filas y columnas para el acceso a cada uno de sus componentes, es a través de estas filas y columnas que se selecciona el Led o los Led's que se van a encender, existen por lo tanto arreglos matriciales de ánodo o de cátodo común.



Figura 6: Apariencia de una matriz para Led's

La cantidad de filas y columnas determina el tamaño de la matriz o la pantalla de Led's, que para este diseño será de 8 filas por 64 columnas, aunque para la explicación del funcionamiento bastara con explicarlo todo con una matriz de 8x8 la que se irá repitiendo hasta alcanzar el tamaño deseado.

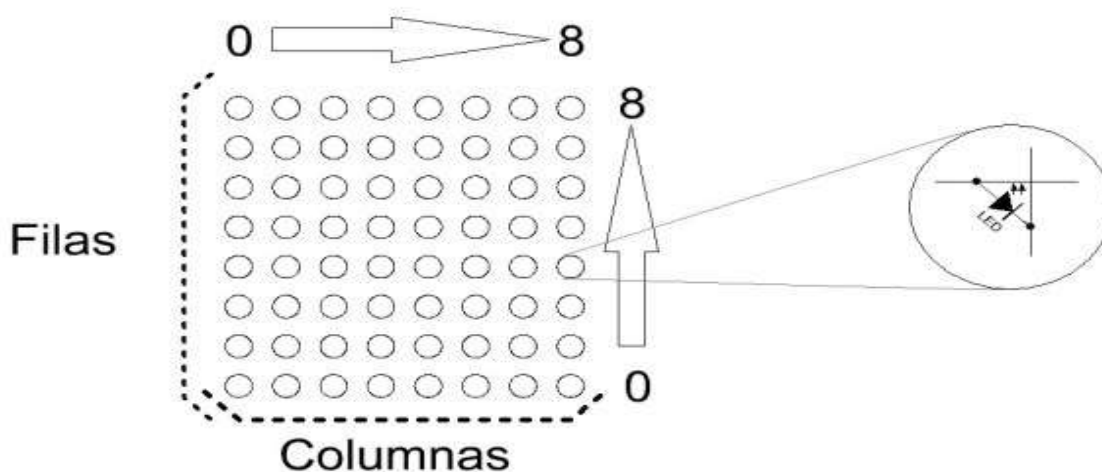
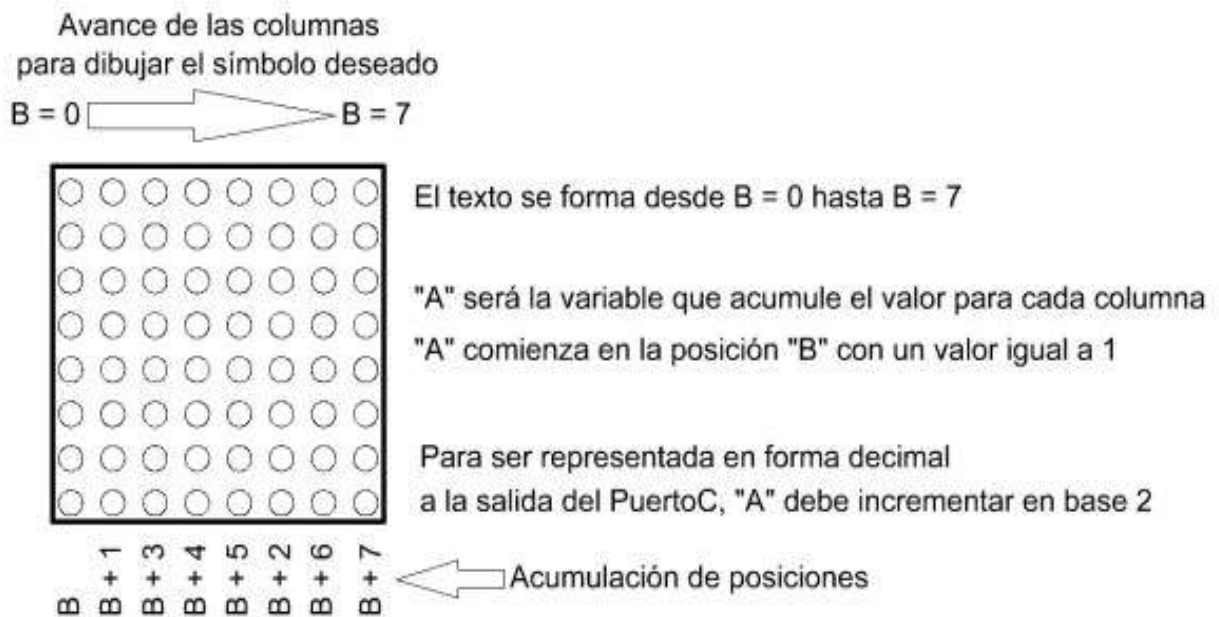


Figura 7: Organización de la Matriz de Led's

Para visualizar un carácter, mostrar un pequeño dibujo u ofrecer cualquier símbolo en la pantalla, debemos desmenuzar el análisis de funcionamiento y, de esa manera, comprenderemos el concepto esencial que rige la mecánica de estos displays gráficos. Tomando como ejemplo una letra, la letra "R", debemos armar inicialmente un "mapa" dentro de la matriz de cómo queremos que se vea o grafique lo que

estamos queriendo mostrar. Es decir, debe existir un trabajo previo, un trabajo duro y a veces tedioso (pero necesario) para lograr llevar a la pantalla la imagen deseada.

Todas las letras del alfabeto, todos los números y símbolos a mostrar deben planificarse (“mapearse”) previamente para que al llegar el momento de la programación del microcontrolador, sólo sean necesarios un par de simples movimientos y formar así el gráfico que deseamos presentar



Contador de posiciones a la salida del puerto C para generar una imagen fija

Figura 8: Explicación lógica de la matriz de Led's

Tal como se puede apreciar en el gráfico siguiente, la letra “R” será la representación de la cadena de valores 0, 0, 127, 76, 72, 74, 49 y 0. Estos valores serán enviados a la matriz mediante las “filas” (8 bits), colocando cada valor en la “columna” correspondiente. Repitiendo el proceso de colocar cada fila en su respectiva columna muchas veces por segundo, tendremos la impresión de estar observando una imagen estática y fija, siendo que en realidad los datos ingresan por las filas, se visualizan en la columna correcta y se apagan. Luego ingresan otros datos en un nuevo ciclo del programa, se visualizan en la columna correcta y se apagan. El ciclo

se repite una y otra vez hasta completar el símbolo deseado, como dijimos antes, muchas veces en un segundo, lo que nos dará la sensación visual de que permanece siempre fijo allí.

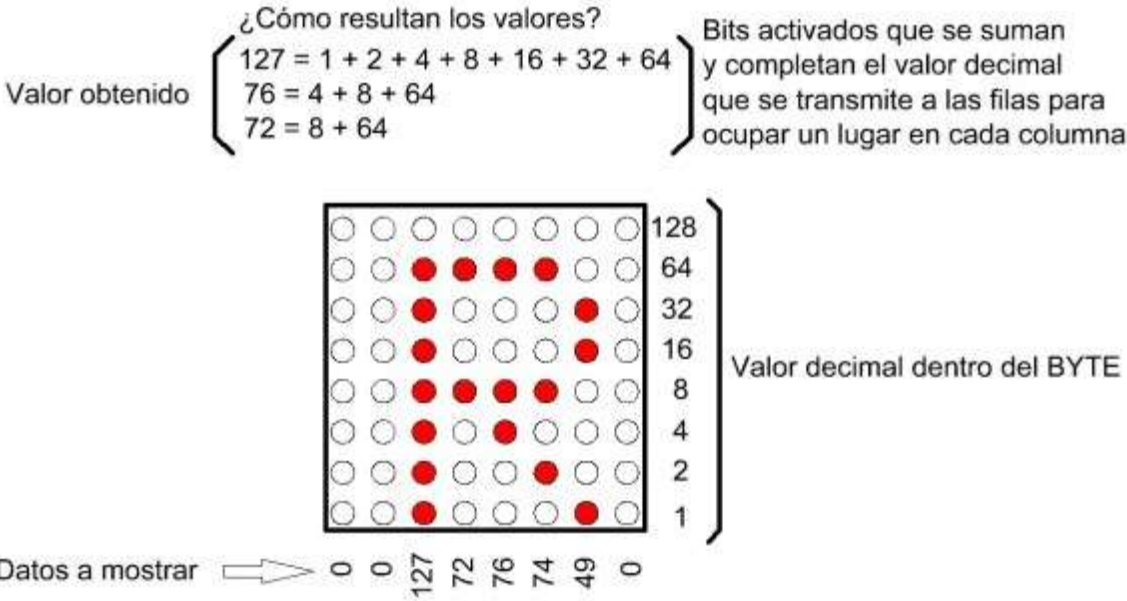


Figura 9: Obtención de valores de la Matriz de Led's.

La imagen se forma por el “barrido” de las columnas a las que se les hacen llegar en el momento apropiado los datos que queremos visualizar. Estos datos llegan de a uno por vez y se colocan en el lugar que le corresponde a cada uno dentro del mapa de bits creado en forma previa. Al repetir el ciclo muchas veces en un segundo, la imagen dará la impresión de estar fija. Para que este efecto se logre con mejor resultado, interviene la inercia de encendido y apagado del Led, sumado a la velocidad de nuestra visión en detectar los mencionados cambios de estado.

Se iniciara el trabajo partiendo de una matriz que sea capaz de representar visualmente los símbolos que deseamos mostrar. En nuestro caso, hemos construido una matriz de 8 LEDs por lado. Pero si lo deseas, puedes reducir el tamaño a 7 Led's de alto por 5 de ancho y seguirás visualizando sin problemas caracteres del alfabeto, números, símbolos de puntuación y cualquier gráfico elemental que sea capaz de verse en esa mínima resolución.

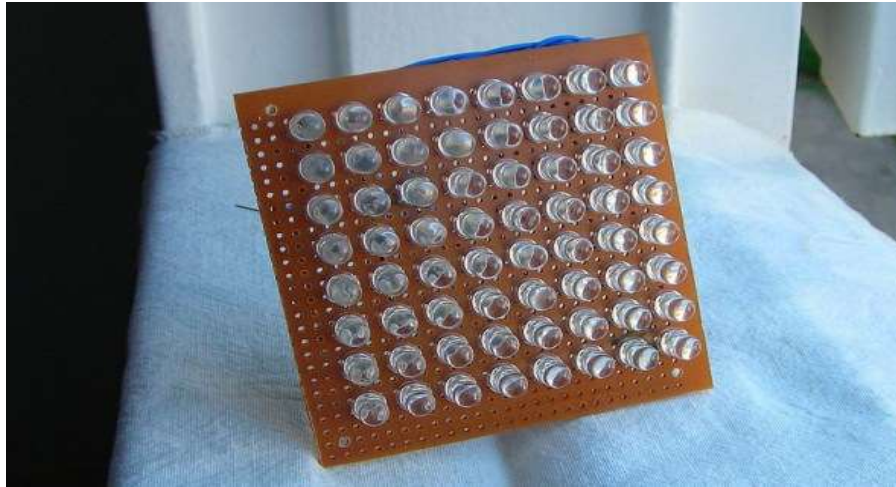


Figura 10: Matriz de Led's (objeto del Trabajo de Aplicación)

La conexión de los 64 Led's que componen la matriz es una de las partes más tediosas del montaje. Se debe tener mucha paciencia y calma, pues al trabajar bajo presión o con los tiempos muy acotados este trabajo puede llevar a un error involuntario que signifique desarmar todo y volver a empezar. Por lo tanto, nuestra recomendación es admitir la necesidad de quemar toda una jornada de trabajo en el armado de la matriz de Led's, para luego no estar buscando posibles fallas que serán muy difíciles de encontrar.

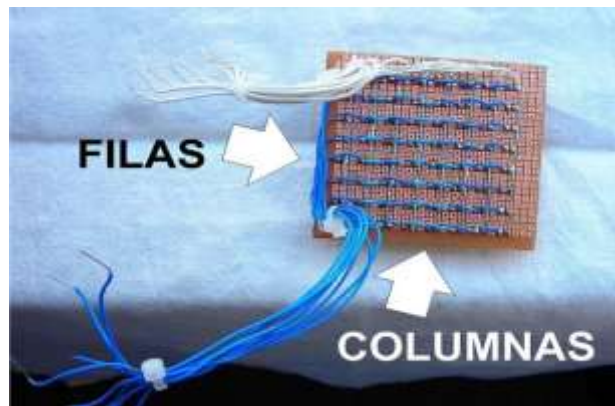


Figura 11: Montaje de la matriz de Led's

Las conexiones son muchas y las posibilidades de error se multiplican, por lo que será necesario poner lo mejor de nuestra paciencia en esta parte del desarrollo. La

situación puede simplificarse si construimos un circuito impreso de doble faz y diseñado en forma específica para esta aplicación. Pero los costos de una buena placa doble faz con agujeros metalizados (conducción de un lado a otro de la placa), el tiempo que se pueda demorar en el diseño de la mencionada PCB, sumado al tiempo que podemos demorar utilizando una placa experimental, pueden acumular escollos que por lo general se resuelven de manera práctica utilizando las placas impresas experimentales, como vemos en la fotografía superior. Visto el diagrama esquemático de conexiones de los LEDs, quedaría de la forma que te mostramos en el siguiente gráfico:

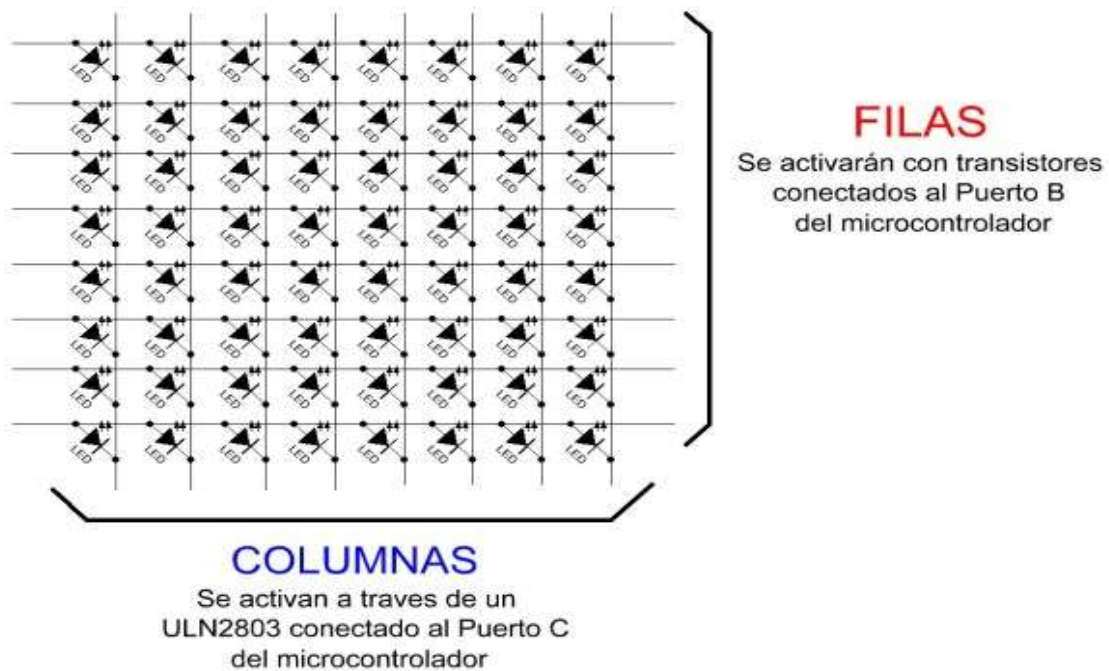


Figura 12: Simbología de la matriz de Led's

El correcto armado de la matriz es el punto clave en la construcción de un cartel de Led's. si existe error en la conexión o posición de alguno de los Led's, se puede pasar horas hasta descubrir el origen de la falla. Por esto, además de la atención en el armado, una vez concluido el montaje se debe comprobar con un multímetro de aguja el correcto funcionamiento de todos y cada uno de los 64 Led's que componen la matriz. Obviar este paso puede significar más adelante un dolor de cabeza al no

saber con certeza si la falla estará en la matriz de conexiones o en los circuitos de activación de los Led's. Por lo tanto, la recomendación es esta: ir por pasos y controlar muy bien todo lo que se está realizando. Una corrección a tiempo puede significar el ahorro de horas de estancamiento en el trabajo.

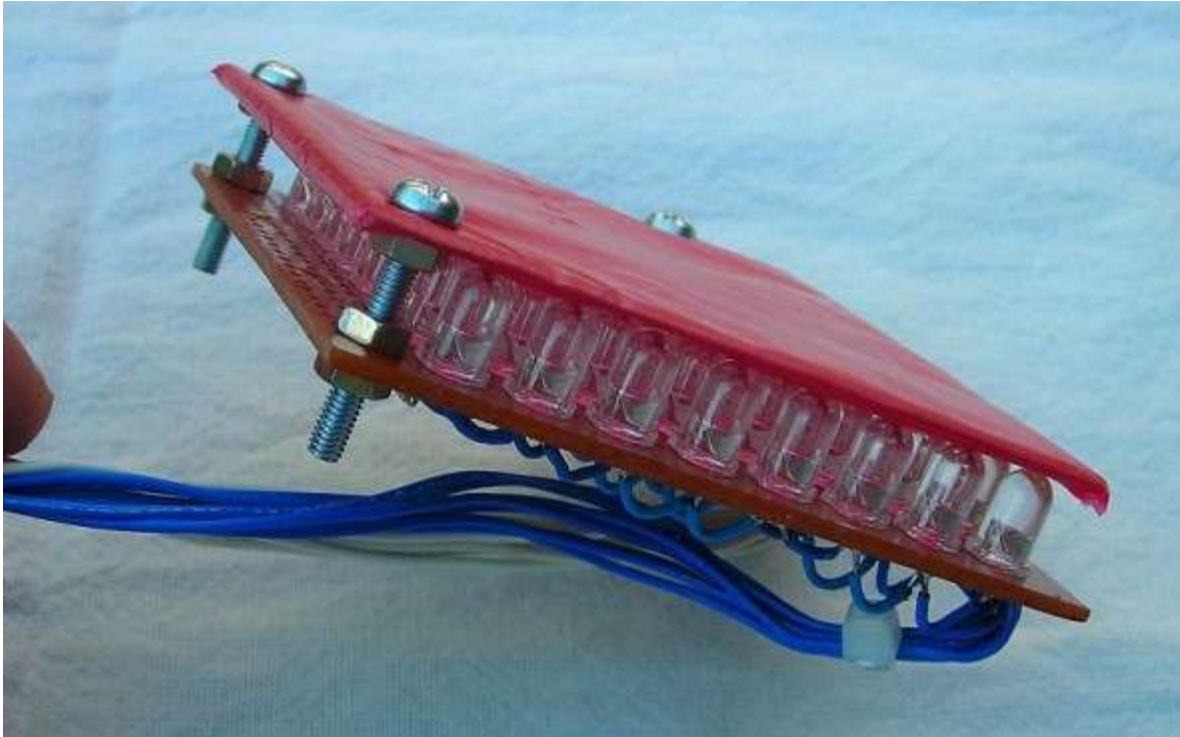


Figura 13: Matriz de Led's con cubierta protectora

Si se utiliza una matriz comprada y armada, solamente se reducirá el trabajo a determinar el pin “out” de la misma (donde está cada conexión de cada Led) y colocarle o no al frente del montaje una cubierta protectora para que no se vean los espacios entre Led's y para que aparezca como un frente ciego que sólo se iluminará al momento de funcionar. En nuestro caso, por utilizar dispositivos del tipo “waterclear”, necesitamos colocarle un plástico en su parte frontal para aplacar un poco el reflejo de la luz ambiente en el cuerpo plástico de cada Led. Además, este método de colocar un material semi-transparente delante de los Led's nos permitirá trabajar con tasas de refresco (los tiempos de encendido/apagado de los Led's) más lentas y eso traerá consigo un mayor brillo final obtenido con una mejor apreciación del mensaje.

CAPITULO II

2. INGENIERÍA DEL PROYECTO

2.1. DESARROLLO DEL PROYECTO

Para la realización del proyecto se utilizara un arreglo matricial más conocido como matriz de Led's, estos serán controlados por el cerebro del circuito un microcontrolador PIC 16F877A, la intensidad lumínica del arreglo matricial estará a cargo de transistores en corte y saturación para seleccionar los Led's que se van a encender, la alimentación del circuito viene dada por la batería del vehículo o por otra fuente de 12 voltios,, la selección de los mensajes estará a cargo de un pulsador.

A continuación el diagrama en bloques que describe cada uno de estos sistemas que juntos forman el proyecto en sí.

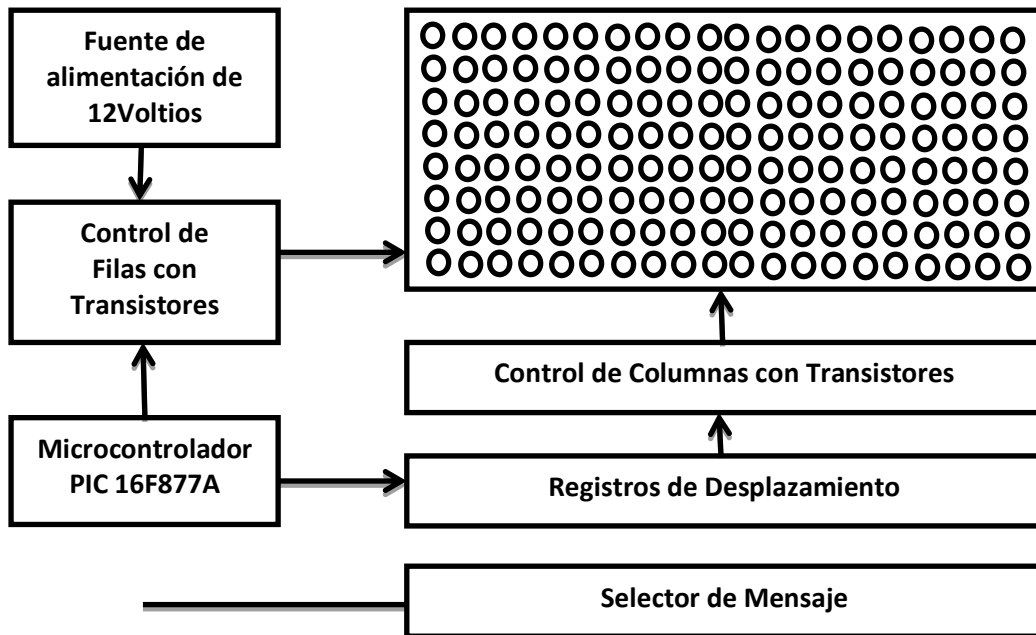


Figura 14: Ingeniería de funcionamiento



Como el cerebro del circuito se utilizara un PIC 16F877A del que se dispondrá de el puerto B para activar las filas y el puerto C para activar las columnas de la matriz, Para las columnas se utilizando un clásico registro de desplazamiento que cargarlas 64 columnas que para garantizar una buena luminosidad requieren de transistores en corte y saturación, para lo cual se utilizara un ULN2803 transistores Darlington en un solo encapsulado, garantizando así su buen funcionamiento. Las 8 filas serán controladas directamente por el puerto Ba través de unos transistores que en base a corte y saturación alimentaran con corriente suficiente a los Led's para generar la luminosidad suficiente para su visualización.

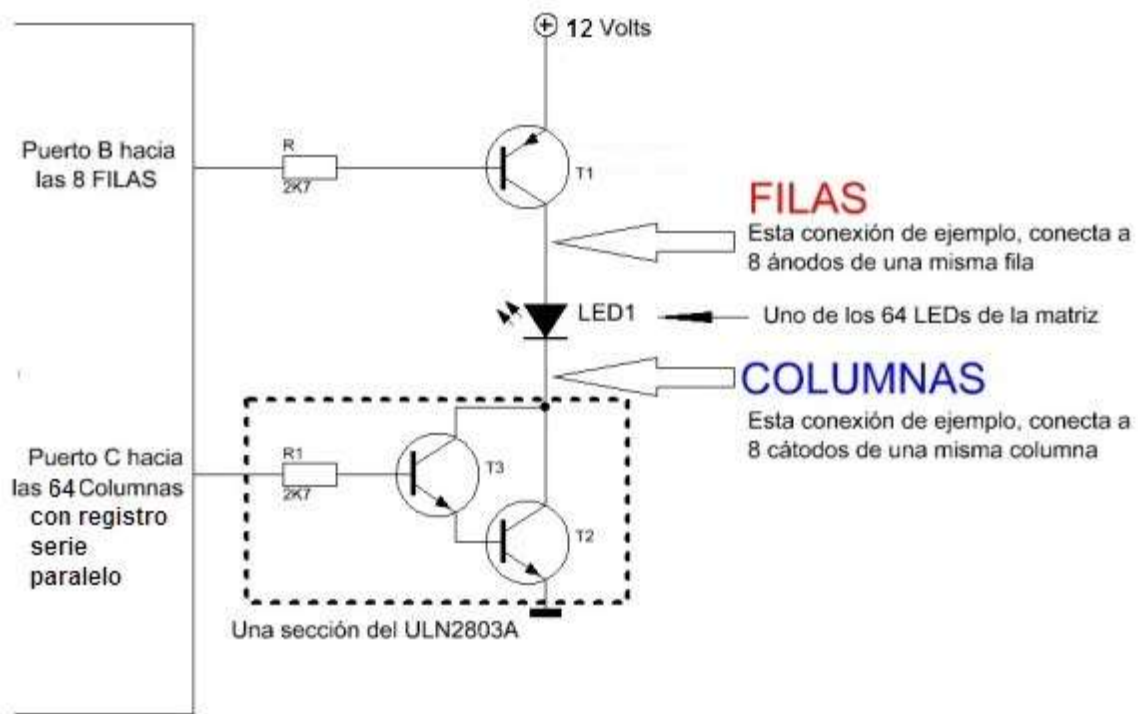


Figura 15: Conexión de la matriz de Led's

El gráfico superior muestra de manera simplificada la conexión de uno de los 64 Led's que componen la matriz. Es un ejemplo genérico de selección de una de las 8 filas y una de las 8 columnas. Cualquier combinación de esta naturaleza energizará un único Led. Esta imagen demuestra que debemos colocar un pin en estado bajo en el Puerto B y otro en estado alto con el Puerto C a través de los registros de desplazamiento serie paralelo para que los transistores conectados a ellos se activen

y logren encender el LED seleccionado. El resultado final será un circuito compuesto por el PIC 16F877A, trabajando con un cristal de 12Mhz, drivers ULN 2803A, algunas resistencias y un circuito de alimentación que permita trabajar con batería o con entrada de tensión externa.

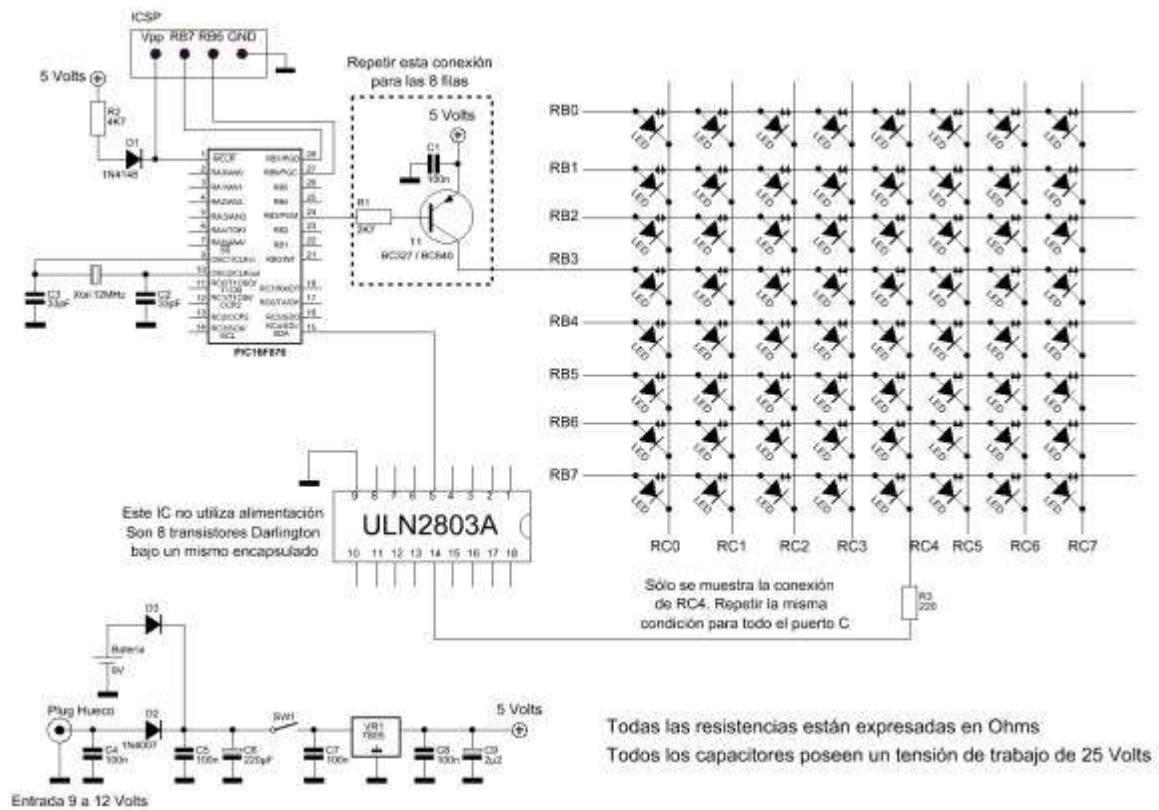


Figura 16: Esquema del Trabajo de aplicación

La inclusión y posición de los diodos 1N4007 en el circuito sirve para proteger la unidad ante accidentales inversiones de polaridad que puedan dañar el PIC. Por otro lado, la variación en el valor de las resistencias de 220 Ohm significará una alteración en el brillo obtenido en el cartel. Es decir, se puede colocar hasta resistencias de 56 Ohm en esos lugares sin provocar problemas funcionales. Pero se debe estar atento a que el consumo de corriente subirá de manera considerable y esto es un problema cuando trabajamos con desarrollos pequeños alimentados a batería. En este caso, que pensamos utilizar el proyecto para exteriores, será

necesaria tanta intensidad luminosa. Pero en caso de utilizarlo para interiores se puede utilizar resistencias de mayor valor.

2.2. DISEÑO DEL HARDWARE DE CONTROL

Como se ha mencionado anteriormente el cerebro del circuito será el microcontrolador PIC16f877A con un cristal de 12Mhz, un arreglo de capacitor y resistencia que asegure el RESET del circuito después de ser alimentado, esto gracias a la carga del capacitor, el puerto B va conectado al Driver de las Filas. Para el control de las columnas el Puerto C brinda 2 pines para el envío de la información a los registros de desplazamiento serie paralelo, que componen junto a los ULN2803 del grupo Driver de columnas. Además el microcontrolador utiliza un pulsador externo para el control de los mensajes, que serán seleccionados por el usuario y creados a su requerimiento.

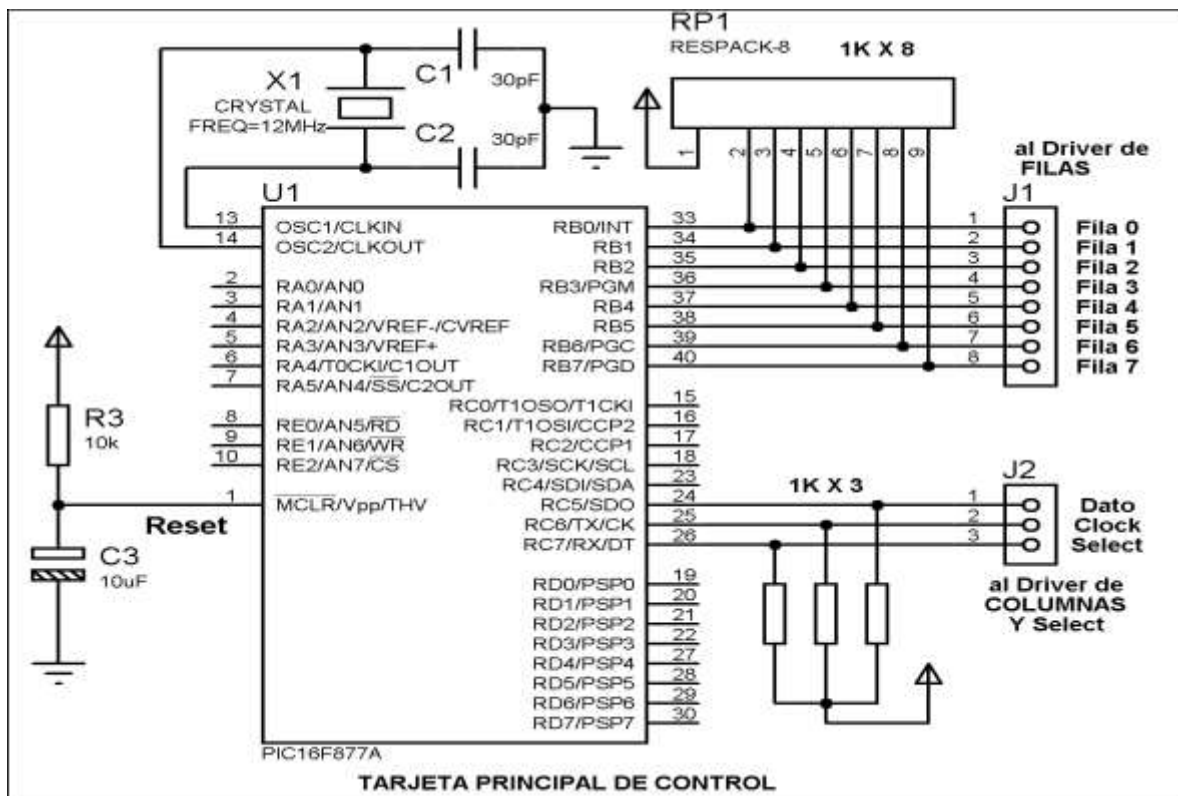


Figura 17: Entradas y salidas lógicas del microcontrolador

Los drivers que controlan las filas están compuestos por transistores de potencia media que funcionan en corte y saturación y conectan a las filas a un voltaje de 12 voltios, suficiente para iluminar a los 64 Led's que puede seleccionar las columnas. Para proteger a el micro controlador se lo ha aislado del circuito de control de Filas a través de un Buffer 74LS245, un simple amplificador de corriente que separa las dos partes y protege a la tarjeta de control.

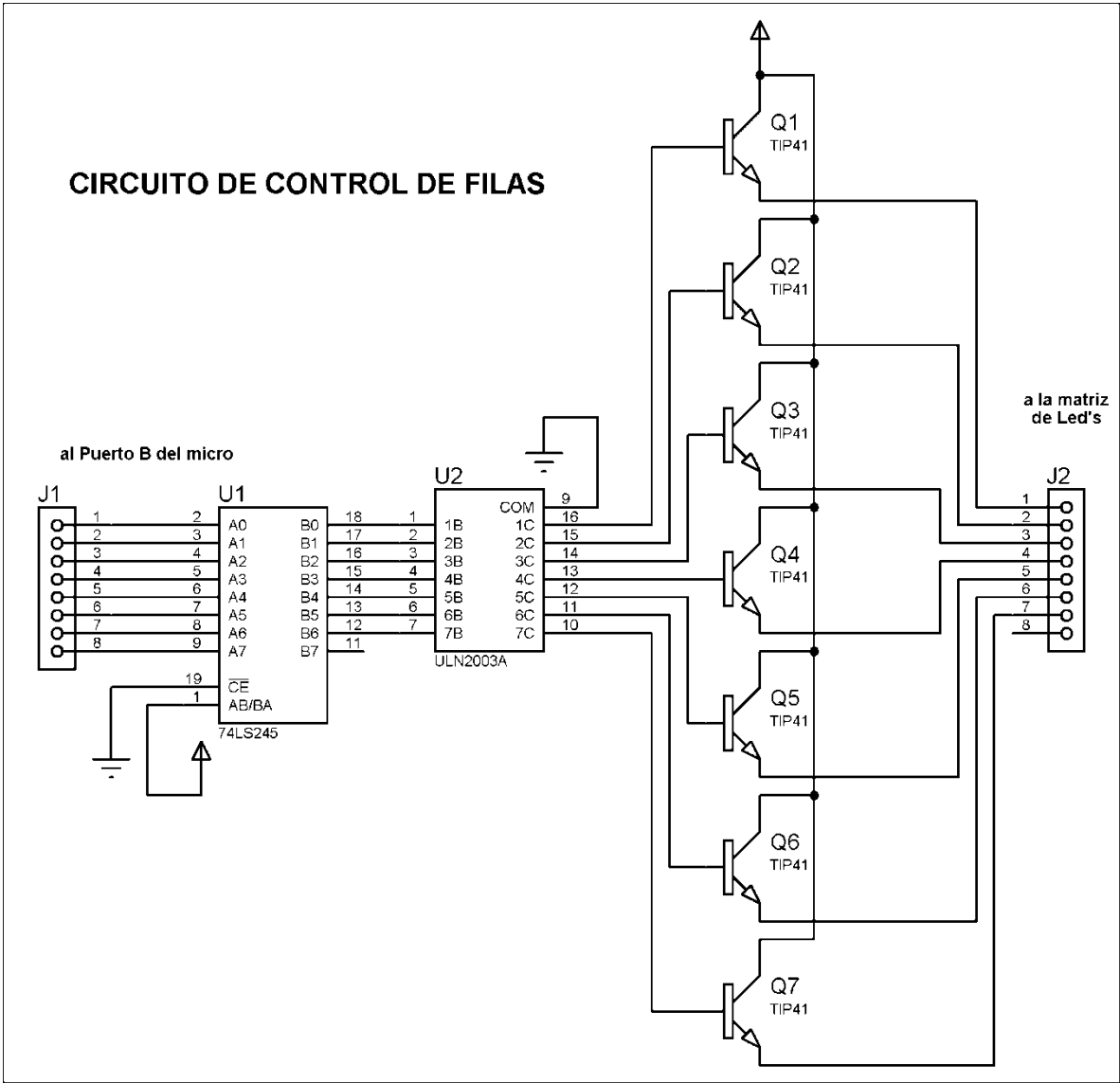


Figura 18: Control de filas

El circuito controlador de Columnas está compuesto por varios CD4094 convertidores Serie Paralelo, esto para poder cargar las 64 columnas con información para poder ser seleccionadas luego por la fila correspondiente donde se mostrara esta información, los CD4094 se colocan en cascada, el microcontrolador provee la información a través de las líneas Clock y Dato, el Clock es común para cada uno de los CD4094 en cascada y la señal Dato+1 se pasa al siguiente CD4094 y así sucesivamente a la cantidad de conversor Serie Paralelo que sea necesario, que para nuestro caso será de 8 por ser 64 Led's en las columnas, a cada CD4094 va conectado un ULN2803 un arreglo Darlington en un solo encapsulado, lo que garantiza que el corte y saturación sea seguro para poder iluminar correctamente el arreglo matricial, el ULN2803 puede soportar corrientes de hasta 500mA, mucho más de lo que en realidad se requiere para poder iluminar cada uno de los Led's.

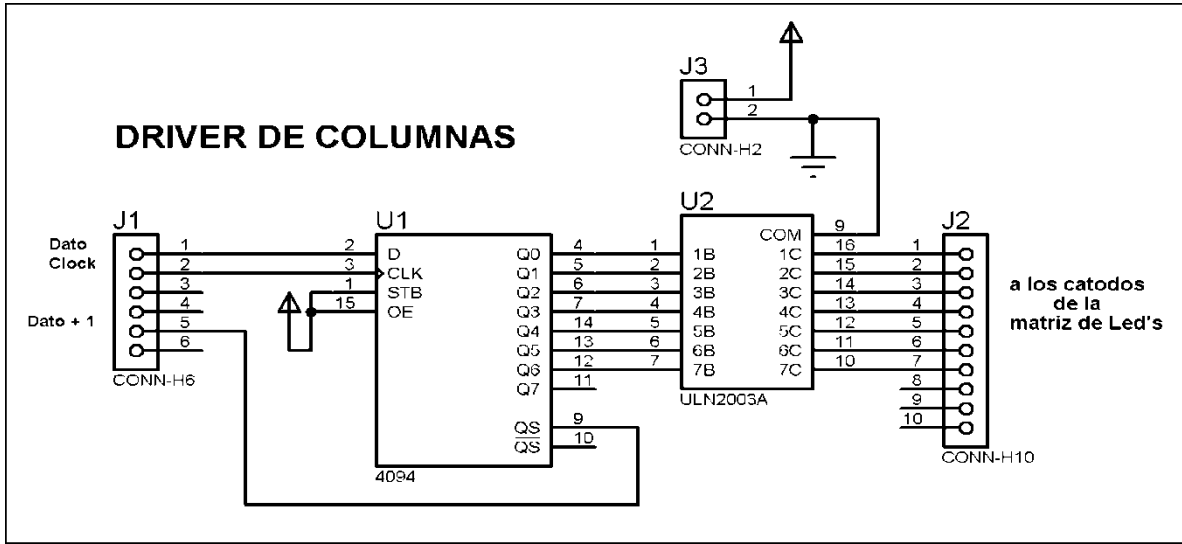


Figura 19: Control de columnas

Un simple pulsador conectado a la terminal Select de la tarjeta de control principal selecciona de forma externa cada uno de los mensajes que se mostraran y que serán predefinidos por el usuario. Visto que el usuario final es el transportista ya sea que cumpla una ruta u otra requiere de solo un solo selector para cambiar la ruta dicha ruta se mostrara en el letrero luminoso y será prediseñada para cada usuario, brindándole una gran variedad, que podrá ser modificable si el así lo solicitare, ya

que esto depende de solamente el programa que se diseñara y que se explicará a continuación.

2.3. DISEÑO DEL SOFTWARE DE CONTROL

El software como se ha explicado está diseñado en lenguaje de alto nivel, en lenguaje Basic, y consta también de bloques que se describirán a través de un diagrama de flujo.

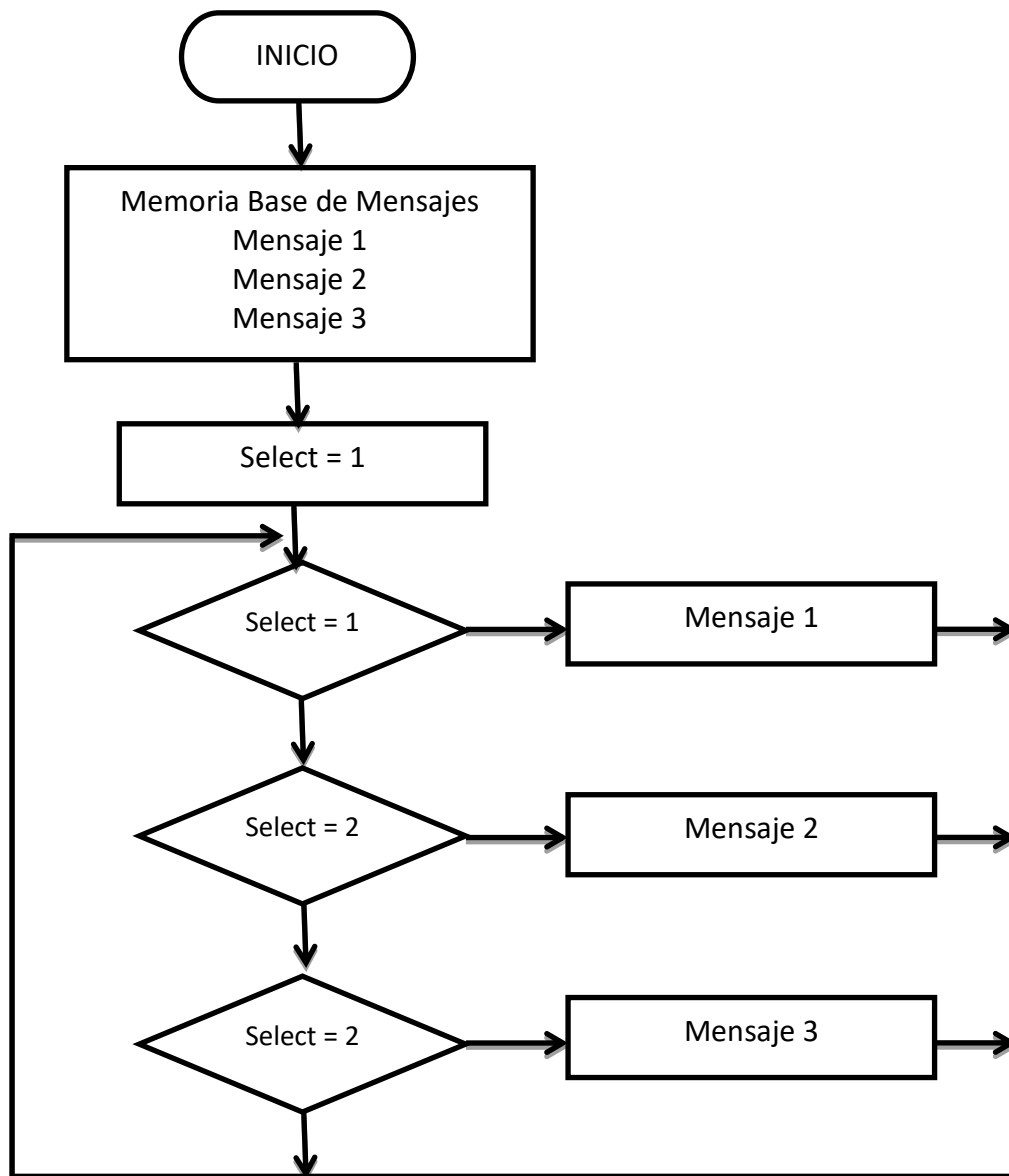


Figura 20: Algoritmo lógico del Trabajo de Aplicación

Se requiere de una memoria de mensajes donde se almacenen los mensajes predefinidos de cada una de la opciones posibles seleccionables, para el ejemplo serán solo tres, pueden ser muchas más y a requerimiento del usuario final, En esta memoria se diseñan los mensajes de cada una de las opciones, cada una de ellas tiene un nombre identificador y en el programa es tratado como un vector donde se almacenan cada una de las palabras en formato binario, cada una de las letras de los mensajes que se van a desplegar.

```

' Declarations section
const mensaje1 as byte [64] = (
000,000,
062,065,065,065,034,000,      'C
062,065,065,065,062,000,      'O
062,065,065,065,034,000,      'C
127,008,008,008,127,000,      'H
126,009,009,009,126,000,      'A
127,073,073,073,054,000,      'B
126,009,009,009,126,000,      'A
127,002,004,002,127,000,      'M
127,073,073,073,054,000,      'B
126,009,009,009,126,000,      'A
000,000 )

dim mensaje2 as byte[64] = (
000,000,
000,000,000,000,000,000,      '
000,000,000,000,000,000,      '
000,000,000,000,000,000,      '
127,064,064,064,064,000,      'L
126,009,009,009,126,000,      'A
000,000,000,000,000,000,      '
127,009,009,009,006,000,      'P
126,009,009,009,126,000,      'A
097,089,073,077,067,000,      'Z
000,000,000,000,000,000,      '
000,000,000,000,000,000,      '
000,000)

dim mensaje3 as byte[64] = (
000,000,000,000,000,
000,000,000,000,000,000,      '
000,000,000,000,000,000,      '
062,065,065,065,062,000,      'O
127,009,025,041,070,000,      'R
063,064,064,064,063,000,      'U
127,009,025,041,070,000,      'R
062,065,065,065,062,000,      'O
000,000,000,000,000,000,      '
000,000,000,000,000,000,      '
000,000,000,000,000)

```

Para desplegar otros posible mensajes solo basta con tener una base de letras prediseñadas e ir formando las palabras que se requieran para cada uno de los posibles mensajes requeridos por el usuario

```
126,009,009,009,126,000, 'A
127,073,073,073,054,000, 'B
062,065,065,065,034,000, 'C
127,065,065,065,062,000, 'D
127,073,073,065,065,000, 'E
127,009,009,001,001,000, 'F
062,065,073,073,058,000, 'G
127,008,008,008,127,000, 'H
065,127,065,000,000,000, 'I
048,064,065,063,001,000, 'J
127,008,020,034,065,000, 'K
127,064,064,064,064,000, 'L
127,002,004,002,127,000, 'M
127,004,008,016,127,000, 'N
062,065,065,065,062,000, 'O
127,009,009,009,006,000, 'P
062,065,081,033,094,000, 'Q
127,009,025,041,070,000, 'R
038,073,073,073,050,000, 'S
001,001,127,001,001,000, 'T
063,064,064,064,063,000, 'U
031,032,064,032,031,000, 'V
063,064,056,064,063,000, 'W
099,020,008,020,099,000, 'X
001,002,124,002,001,000, 'Y
097,089,073,077,067,000, 'Z
```

Y el código principal del programa.

```
dim seleccion, as byte
' Main program
main:
seleccion = 1
while (TRUE)
select case seleccion
case 1
most_men1
case 2
most_men2
case 3
most_men3
endselect
wend
end.
```


Para el control de los registros de corrimientos CD4094 tendremos el siguiente código, que controlara el desplazamiento de la información hacia las columnas.

```

sub procedure cd4094
  dim k as integer
  ClearBit(PORTC, 7)
  if (var = 1) then
    SetBit(PORTC, 6)
  else
    ClearBit(PORTC, 6)
  end if
  SetBit(PORTC, 7)
  Delay_us(1)
  ClearBit(PORTC, 7)
end sub

```

Para mostrar la información en las pantallas cada cierto tiempo y asegurar que la información tenga un barrido uniforme se utiliza el timer 0 del microcontrolador programado para que cada 2ms refresque la información de todo el letrero luminosos y así generar el efecto de persistencia visual que se ha explicado con anterioridad.

```

sub procedure interrupt      ' Rutina de interrupción
  PORTB = 0xff              ' apaga pantalla
  ' carga las columnas
  for i = 15 to 0 step -1
    var = Testbit(pantalla[i],j)
    cd4094
  next i
  ' siguiente columna
  inc(j)
  if (j = 8) then
    j = 0
  end if
  ' mostrar fila
  PORTB = not(filas[j])
  ' cargar nuevamente el timer y la interrupcion
  TMR0 = 6                  ' Temporizador Timer0 devuelve su valor inicial
  INTCON = 0x20            ' Bit TOIE está a 1, bit TOIF está a 0
end sub

```

Para seleccionar la información que se va a enviar a la pantalla de Led's se utiliza una interrupción, esta es activada por el pulsador del usuario final e incrementa la selección del mensaje a mostrar.

```
sub procedure interrupt
  Delay_ms(10)
  seleccion = Seleccion + 1
  if (Seleccion = 4) then
    Seleccion = 1
  end if
end sub
```

2.4. DISEÑO DE PLACAS DE CIRCUITO IMPRESO

Una vez diseñado el software y el hardware del proyecto, la realización de las placas para el armado final se muestra a continuación.

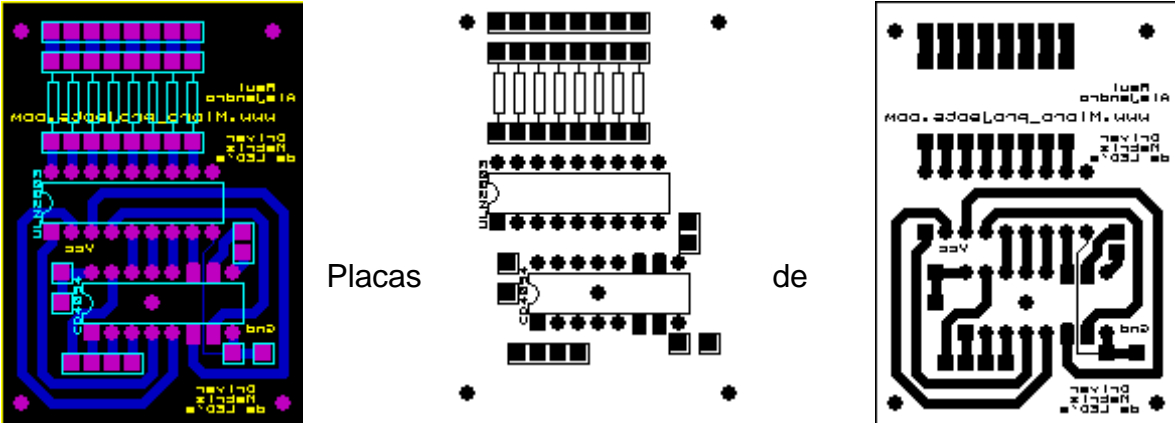


Figura 21: Forma de la placa base del Trabajo de Aplicación

circuito control de columnas, de izquierda a derecha se muestran, el diagrama de conexión, el diagrama de posición de los componentes, el diagrama de las pistas del lado de la soldadura.

Para el diseño de las placas de control de Filas, se muestran la placa vista de arriba.

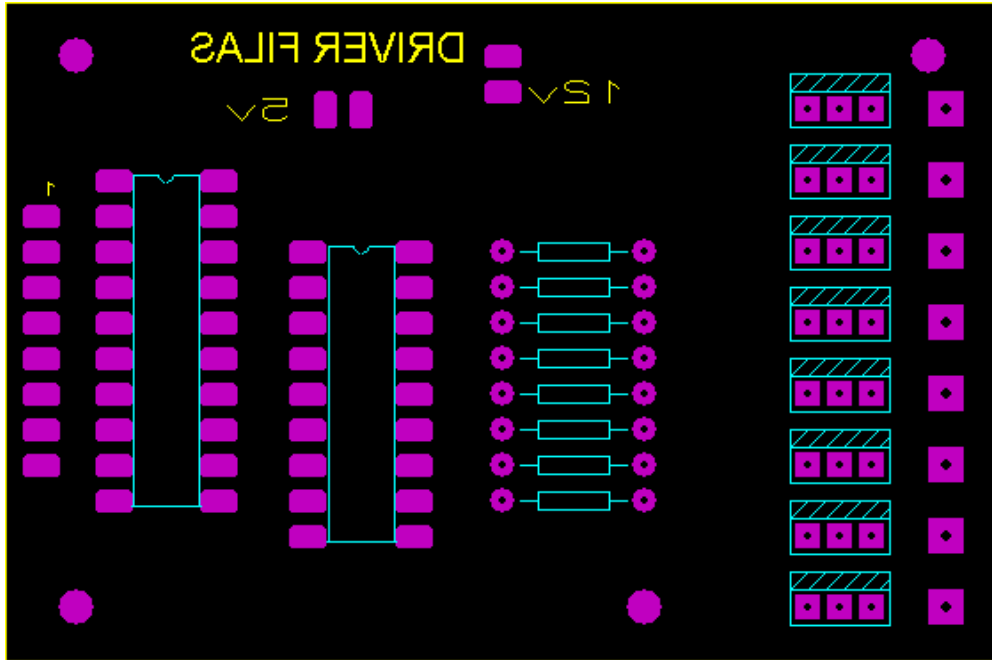


Figura 22: Placa vista de abajo

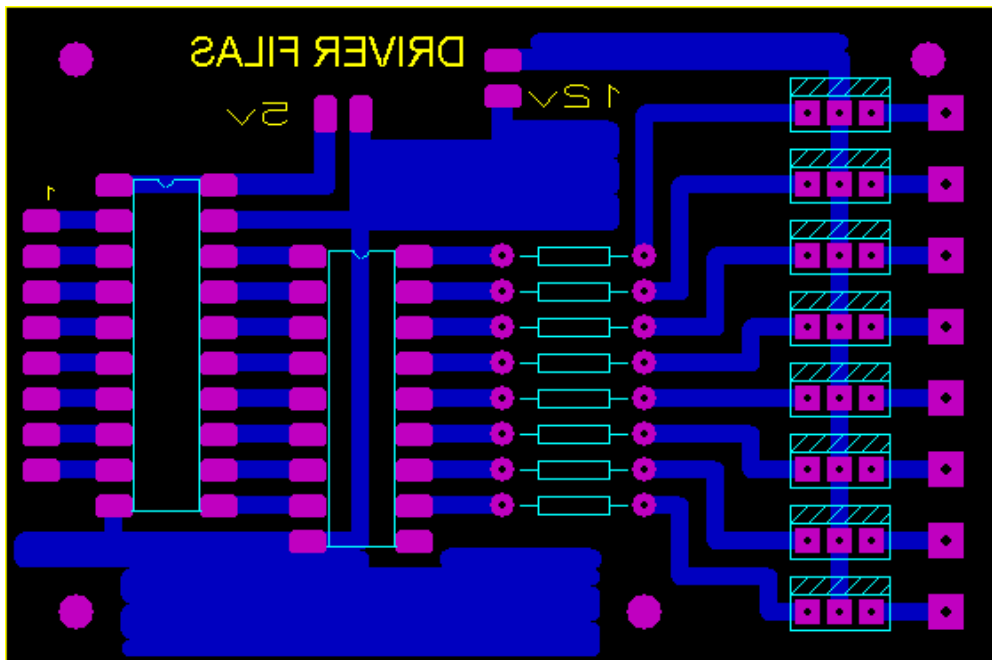


Figura 23: Placa vista de arriba

La placa del circuito principal está compuesta por la cara superior de la placa.

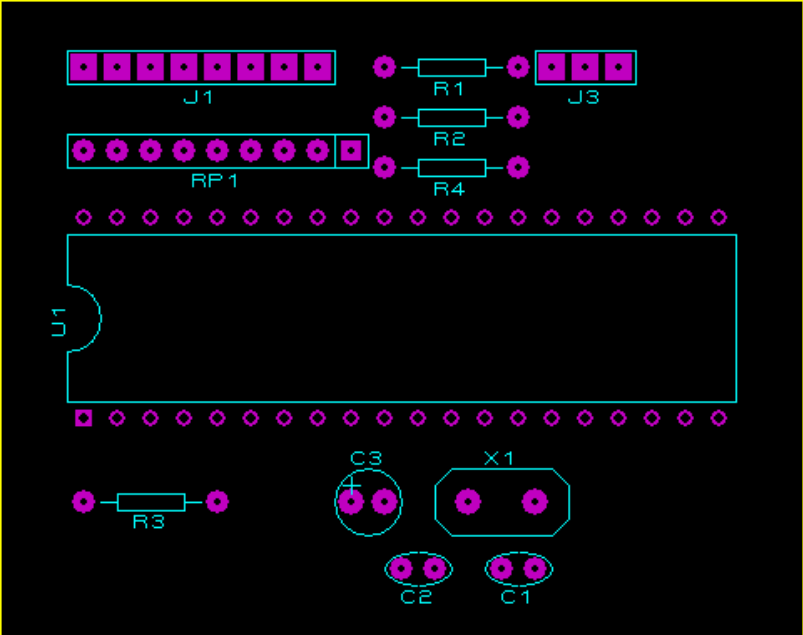


Figura 24: La cara inferior para realizar soldaduras

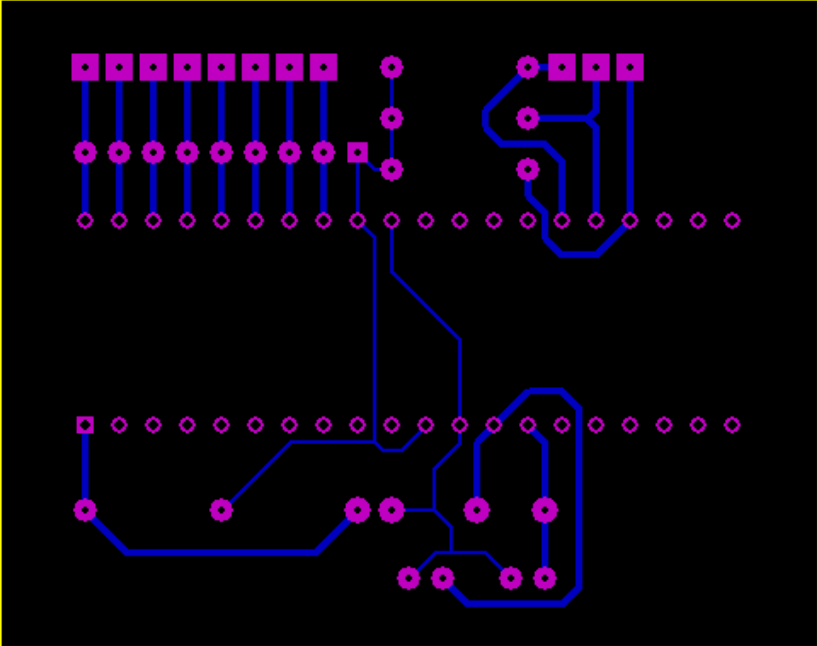


Figura 25: Conexiones impresas

CAPITULO III

3. COSTOS

#	Descripción	Cantidad	Precio Unitario Bs.	Precio Total Bs.
1	Led's	512	0.5	256
2	Transistores	8	5	40
3	Microcontrolador PIC 16F877A	1	65	65
4	ULN2803	8	8	64
5	CD4093	8	4	32
6	Resistencias	64	0.5	32
7	74LS245	1	5	5
8	cristal	1	8	8
9	pulsador	1	1	1
10	capacitores	1	2	2
11	Placa	1	80	80
Costo Total Bs.				585

CONCLUSIONES

Es factible de controlar una matriz de Led's para uso de la sociedad en el ámbito de la señalización de tráfico actualmente moderado y es previsible que se incremente en el futuro, ya que su función es superior a la de la lámpara incandescente y la lámpara fluorescente. La iluminación con Led's presenta indudables ventajas: 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, y tiene respuesta rápida.

RECOMENDACIÓN

Se recomienda tener sumo cuidado con la soldadura de los Led's ya que un mal soldado puede no solo afectar a que ese Led funcione mal sino que afectara toda la fila y toda la columna, generando un fallo catastrófico en el letrero electrónico.,

BIBLIOGRAFÍA

Libros consultados:

- Curso de Electronica Digital Cedit - Volumen 1
- Curso de Electronica Digital Cedit - Volumen 2
- Microprocesadores y microcontroladores -Marcelo Estévez.
- More Electronic Gadgets for the Evil Genius 40 NEW Projects 2da edition
- Pic 2da Parte Pic 16f87x - Angulo , Romero , Angulo

Páginas web consultadas:

- Microcontroladores PIC 16F62X 16F81X 16F87X
- www.matrixdotdisplay.com.eu
- www.microchip.com
- www.wikipedia.com

ANEXOS

ANEXO 1: ALGORITMO DEL PROGRAMA

dim selección, var,i,j as byte

Definición de contantes letras

const letras as byte [180] = (

126,009,009,009,126,000,	'A
127,073,073,073,054,000,	'B
062,065,065,065,034,000,	'C
127,065,065,065,062,000,	'D
127,073,073,065,065,000,	'E
127,009,009,001,001,000,	'F
062,065,073,073,058,000,	'G
127,008,008,008,127,000,	'H
065,127,065,000,000,000,	'I
048,064,065,063,001,000,	'J
127,008,020,034,065,000,	'K
127,064,064,064,064,000,	'L
127,002,004,002,127,000,	'M
127,004,008,016,127,000,	'N
062,065,065,065,062,000,	'O
127,009,009,009,006,000,	'P
062,065,081,033,094,000,	'Q
127,009,025,041,070,000,	'R
038,073,073,073,050,000,	'S
001,001,127,001,001,000,	'T
063,064,064,064,063,000,	'U
031,032,064,032,031,000,	'V
063,064,056,064,063,000,	'W
099,020,008,020,099,000,	'X
001,002,124,002,001,000,	'Y

```

097,089,073,077,067,000) 'Z
Declaración de mensaje 1
const mensaje1 as byte [64] =
(000,000,
062,065,065,065,034,000, 'C
062,065,065,065,062,000, 'O
062,065,065,065,034,000, 'C
127,008,008,008,127,000, 'H
126,009,009,009,126,000, 'A
127,073,073,073,054,000, 'B
126,009,009,009,126,000, 'A
127,002,004,002,127,000, 'M
127,073,073,073,054,000, 'B
126,009,009,009,126,000, 'A
000,000)'

```

```

' Declaración de mensaje 2
const mensaje2 as byte[64] = (
000,000,
000,000,000,000,000,000, '
000,000,000,000,000,000, '
127,064,064,064,064,000, 'L
126,009,009,009,126,000, 'A
000,000,000,000,000,000, '
127,009,009,009,006,000, 'P
126,009,009,009,126,000, 'A
097,089,073,077,067,000, 'Z
000,000,000,000,000,000, '
000,000,000,000,000,000, '
000,000)

```

```

' Declarations de mensaje3

```



```

const mensaje3 as byte[64] = (
000,000,000,000,000,      '
000,000,000,000,000,000,  '
000,000,000,000,000,000,  '
062,065,065,065,062,000,  'O
127,009,025,041,070,000,  'R
063,064,064,064,063,000,  'U
127,009,025,041,070,000,  'R
062,065,065,065,062,000,  'O
000,000,000,000,000,000,  '
000,000,000,000,000,000,  '
000,000,000,000,000,000)  '

'carga de datos
'de los registros de desplazamiento
'serie paralelo
Sub procedure cd4094
dim k as integer
  clearBit (PORTC,7)
  if (var = 1) then
    SetBit(PORTC,7)
  else
    clearBit (PORTC,7)
    Delay_us(1)
    clearBit (PORTC,7)
  end if
end sub

'procedimiento de interrupción
' para cambio de mensaje al siguiente
sub procedure interrupt
  Delay_ms(10)
  seleccion = Seleccion + 1

```

```

if (Seleccion = 4) then
    Seleccion = 1
end if
end sub
' mostrar mensaje 1
sub procedure most_men1
    PORTB = 0xff
    ' cargar las columnas
    for i = 15 to 0 step -1
        var = TestBit(mensaje1[i],j)
        cd4094
    next i
    ' siguiente columnas
    inc(j)
    if (j = 0) then
        j = 0
    end if
    ' mostrar fila
    PORTB = not(mensaje1[j])
    ' cargar nuevamente el timer y la interrupción
    ' TMR0 = 6
    INTCON = 0x20
end sub
' mostrar mensaje 2
sub procedure most_men2
    PORTB = 0xff
    ' cargar las columnas
    for i = 15 to 0 step -1
        var = TestBit(mensaje1[i],j)
        cd4094
    next i

```

```

' siguiente columnas
inc(j)
if (j = 0) then
  j = 0
end if
' mostrar fila
PORTB = not(mensaje1[j])
' cargar nuevamente el timer y la interrupción
' TMR0 = 6
INTCON = 0x20
end sub
' mostrar mensaje 3
sub procedure most_men3
PORTB = 0xff
' cargar las columnas
for i = 15 to 0 step -1
  var = TestBit(mensaje1[i],j)
  cd4094
next i
' siguiente columnas
inc(j)
if (j = 0) then
  j = 0
end if
' mostrar fila
PORTB = not(mensaje1[j])
' cargar nuevamente el timer y la interrupcion
' TMR0 = 6
INTCON = 0x20
end sub
' Programa Principal

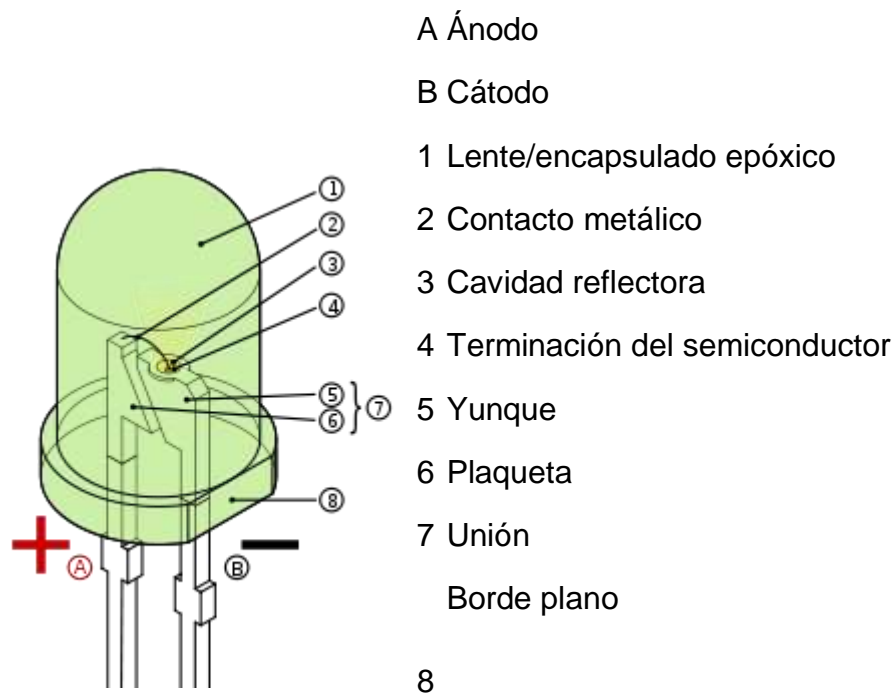
```

```

main:
seleccion = 1
while (TRUE)
select case seleccion
case 1
most_men1
case 2
most_men2
case 3
most_men3
end select
wend
end.

```

ANEXO 2: ESTRUCTURA DE UN LED



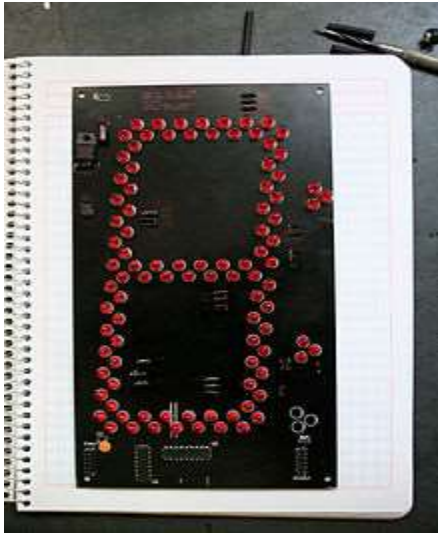
ANEXO 3: COMPUESTOS EMPLEADOS EN LA CONSTRUCCIÓN DE LED'S

Compuesto	Color	Long. de onda
arseniuro de galio (GaAs)	Infrarrojo	940 nm
arseniuro de galio y aluminio (AlGaAs)	rojo e infrarrojo	890 nm
arseniuro fosfuro de galio (GaAsP)	rojo, anaranjado y amarillo	630 nm
fosfuro de galio (GaP)	verde	555 nm
nitruro de galio (GaN)	verde	525 nm
seleniuro de cinc (ZnSe)	azul	
nitruro de galio e indio (InGaN)	azul	450 nm
carburo de silicio (SiC)	azul	480 nm
diamante (C)	ultravioleta	
silicio (Si)	en desarrollo	

ANEXO 4: USO EN EL TRANSPORTE PÚBLICO



ANEXO 5: FORMA DE LETREROS ILUMINADOS CON LED'S



ANEXO 6: DISPOSITIVOS UTILIZADOS

