

FECHA	2010-01-26
-------	------------

NÚMERO RA	
PROGRAMA	PROGRAMA DE INGENIERÍA ELECTRÓNICA

AUTOR (ES)	RAMÍREZ ENRIQUE, Julián
TÍTULO	Desarrolló de un sistema electrónico funcional que proyecta imágenes holográficas empleando leds rgb

PALABRAS CLAVES	Diodo emisor de luz, microcontrolador, modulación ask y rgb.
-----------------	--

DESCRIPCIÓN	Desarrolló e implementación de un mecanismo electrónico que proyecta la imagen deseada, mediante la rotación de una columna de leds RGB a una velocidad determinada, se da una apreciación subjetiva de la imagen, esta sensación se produce en respuesta a la estimulación del ojo y a sus mecanismos nerviosos por la energía luminosa de ciertas longitudes de onda, generando una ilusión de imagen que combinada con pequeñas variaciones de la imagen producen el video o imágenes en movimiento.
-------------	---

FUENTES BIBLIOGRÁFICAS	<p><u>"MICROCONTROLADOR PIC16F84. Desarrollo de proyectos"</u> de la Editorial Ra-Ma</p> <p>José M^a Angulo Usategui, Ignacio Angulo Martínez - Microcontroladores PIC</p> <p>libro práctico de los generadores, transformadores y motores eléctricos Escrito por Gilberto Harper Enriquez</p> <p>Electronica_ucontrol_revista_0001</p>
------------------------	---

NÚMERO RA	
PROGRAMA	PROGRAMA DE INGENIERÍA ELECTRÓNICA

CONTENIDOS	<p>los pasos de la investigación:</p> <p>1.1 SELECCIÓN DE GIRO</p> <p>1.2 PRUEBAS DE GIRO</p> <p>Se inició por elegir cuál sería la mejor forma de giro del dispositivo, la cual se podría realizar en dos formas, es decir si marca un plano cartesiano, un giro sería sobre el eje x y el otro sobre el eje y</p> <p>1.3 SELECCIÓN DEL MOTOR</p> <p>1.4 SELECCIÓN DE LEDS</p> <p>1.5 SELECCIÓN DE MICRO</p> <p>Se observo los de gama media; la cual hay varios microcontroladores. En esta gama se han presentado varias evoluciones, desde cuando salió al mercado el PIC16C84 que posteriormente fue remplazado por el PIC16F84 (la F significa que tiene memoria flash), y este por el PIC16F84A, hasta llegar al PIC16F877a que es un microcontrolador muy poderoso.</p> <p>1.6 SELECCIÓN DE BATERIAS</p> <p>1.7 AVANCE PROTOTIPO CON MODULO RF</p> <p>Los módulos de radiofrecuencia de 434Mhz (TLP y RLP 434) para comunicar dos circuitos, en este caso un transmisor TX y un receptor RX.</p> <p>1.8 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO FINAL</p> <p>Objetivo General Desarrollar un sistema electrónico funcional que proyecte imágenes holográficas de la mejor calidad posible empleando leds RGB.</p>
------------	---

NÚMERO RA	
PROGRAMA	PROGRAMA DE INGENIERÍA ELECTRÓNICA

METODOLOGÍA	<p>2.ENFOQUE DE LA INVESTIGACIÓN</p> <p>El enfoque a emplear en la investigación y de acuerdo a las políticas para la investigación establecidas por la Universidad, es:</p> <p>Empírico- analítico: Cuyo interés es el técnico, orientado a la interpretación y transformación del mundo material; proporciona una estructura particular a la metodología de investigación en tanto que orienta el trabajo a la contrastación permanente de las aseveraciones teóricas con la verificación experimental, de manera que los cálculos generados a través de modelos matemáticos y simulaciones computacionales se deben retroalimentar con la experiencia, en la búsqueda de información cada vez más confiable y practica para la solución del problema.</p> <p>2.1 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA</p> <p>2.1.1 TECNOLOGÍAS ACTUALES Y SOCIEDAD: La sociedad requiere de conocimientos técnicos y científicos de vanguardia que ayuden a la solución de problemas o faciliten los procesos de mejoramiento de la calidad de vida de las personas que pertenecen a un grupo social determinado.</p> <p>2.1.2 INSTRUMENTACIÓN Y CONTROL DE PROCESOS: Procesos asociados con la interacción de los diversos elementos que constituyen mecanismos generadores de potencia como lo son los motores eléctricos.</p> <p>2.1.3 MICROELECTRÓNICA: Con el desarrollo de este campo el hombre ha avanzado en los últimos 40 años. Es por ello que se hace necesario el estudio de éste para seguir en el avance y el mejoramiento de la calidad de vida de la humanidad</p> <p>· La microelectrónica es fundamental permite el desarrollo de</p>
-------------	--

	<p>aplicaciones como:</p> <ul style="list-style-type: none"> • Juegos. • Amplificadores especiales. • Emuladores de otros dispositivos. • Instrumentación. • Sensórica.
--	--

<p>CONCLUSIONES</p>	<p>El tipo de giro más adecuado es en el eje x, y contó con la observación de características como el tipo de visualización, debido a que se puede ver la imagen en su rotación de 360 grados, y no de una forma unipersonal, otro punto a resaltar es la imagen debido a que no se recorta en ningún punto, formando un pantalla perfecta sin recortes , el único inconveniente es por su diseño debido a que en este eje x, el tipo de construcción de la columna de leds tiene que ser en su respectivo centro de gravedad o proporcionaría deficiencia en la construcción de la imagen.</p>
	<p>El tipo de leds más adecuado, depende de la finalidad de la imagen a requerir, en el dispositivo se usó los leds RGB, aunque se optó en un principio por diseñar el Led RGB con los tres leds unidos (rojo, verde y azul) pero generaba en la visualización de la imagen espacios e imágenes borrosas, además el Led RGB nos permite no solo ver colores principales sino combinación de ellos como el amarillo, el magenta y el cian.</p>

DESARROLLO DE UN SISTEMA ELECTRÓNICO FUNCIONAL QUE
PROYECTA IMÁGENES HOLOGRÁFICAS EMPLEANDO LEDS RGB

JULIÁN ENRIQUE RAMÍREZ TORRES

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D. C.
2010

DESARROLLO DE UN SISTEMA ELECTRÓNICO FUNCIONAL QUE
PROYECTA IMÁGENES HOLOGRÁFICAS EMPLEANDO LEDS RGB

JULIÁN ENRIQUE RAMÍREZ TORRES

PROYECTO DE GRADO COMO REQUISITO PARA OPTAR POR EL TITULO
DE ING.ELECTRONICO

Asesor:
NESTOR FERNANDO PENAGOS QUINTERO
INGENIERO ELECTRÓNICO

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D. C.
2010

Tabla de Contenido

INTRODUCCIÓN	9
1. PLANTEAMIENTO DEL PROBLEMA.....	10
1.1 ANTECEDENTES (ESTADO DEL ARTE)	10
1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA.....	11
1.3 JUSTIFICACIÓN	11
1.4 OBJETIVOS DE LA INVESTIGACIÓN	12
1.4.1 Objetivo General	12
1.4.2 Objetivos Específicos.....	13
1.5 ALCANCES Y LIMITACIONES DEL PROYECTO	13
2. MARCO DE REFERENCIA.....	13
2.1 MARCO TEÓRICO - CONCEPTUAL.....	13
2.1.1 La Luz y el Ojo Humano.	14
2.1.2 La Luz.....	14
2.1.3 El Ojo Humano y la Visión.	17
2.1.4 Curva de Sensibilidad Relativa.	20
2.1.5 Integración de la visión.	21
2.1.6 La luminancia.	21
2.1.8 Saturación.	22
2.1.9 Agudeza visual.....	22
2.1.10 Memoria visual.	23
3. METODOLOGÍA	24
3.1 ENFOQUE DE LA INVESTIGACIÓN	24
3.2 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA.....	24
3.2.1 TECNOLOGÍAS ACTUALES Y SOCIEDAD	24
3.2.2 INSTRUMENTACIÓN Y CONTROL DE PROCESOS.....	24
3.2.3 MICROELECTRÓNICA.....	24
3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN	25
4. DESARROLLO INGENIERIL.....	26
4.1 SELECCIÓN DE GIRO	26
4.2 PRUEBAS DE GIRO	27

4.2.1 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO 1	28
4.3 SELECCIÓN DEL MOTOR	30
4.3.1 CÁLCULO DE LA POTENCIA DEL MOTOR	34
4.3.2 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO 3	44
4.4 SELECCIÓN DE LEDS.....	49
4.4.1 DESVENTAJAS Y VENTAJAS DE SELECCIÓN DE LEDS.....	49
4.5 SELECCIÓN DE MICRO	52
4.6 SELECCIÓN DE BATERIAS.....	54
4.7 AVANCE PROTOTIPO CON MODULO RF.....	55
5. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS.....	66
CONCLUSIONES.....	70
RECOMENDACIONES.....	71
BIBLIOGRAFÍA.....	72
GLOSARIO.....	73
ANEXOS A.....	75
ANEXO B.....	76

INTRODUCCIÓN

Actualmente se desarrollan aplicaciones muy ingeniosas para ser implementadas en grandes empresas, y ha llegado a nivel de las Pymes, en todas las ciudades, con el convencimiento de que esto reduce cada vez más la brecha digital en gran medida.

En cualquier lugar del mundo los proyectos constan de múltiples etapas; de recopilación innumerable de información e inclusive de pautas de diseño para que el resultado final sea excelente, sin duda el desarrollo de un sistema electrónico funcional que proyecta imágenes holográficas empleando leds RGB, es una de las respuestas a muchos de las necesidades, como publicis, tableros de información y enseñanza visual para niños o adultos.

Un Holograma, más que un dispositivo constituido por una superficie rotatoria, facilita y permita ingresar datos a un micro controlador el cual proporciona un elemento tanto de aprendizaje, como de diversión e integración, más aun cuando es sencillo de manipular.

El objetivo del proyecto es dar a conocer como los hologramas permiten mostrar imágenes mediante el giro a una velocidad constante de una columna de leds que da una apreciación subjetiva de la imagen al ojo humano.

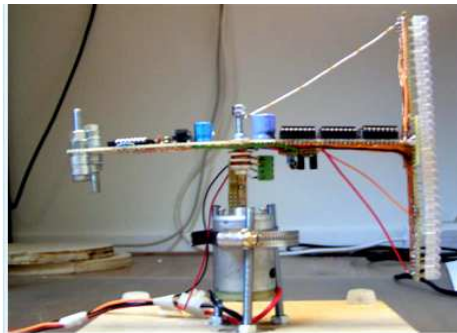
Con el proyecto se busca demostrar que con el giro constante de la columna de leds se puede reemplazar una matriz de punto, y enfocar el proyecto como una ayuda interactiva para el aprendizaje de niños en identificación de figuras geométricas.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES (ESTADO DEL ARTE)

En el mundo se han trabajado distintos prototipos pero no hay una empresa que produzca esto en serie con alguna aplicación concreta. Los prototipos que se encuentran funcionan en dos sentidos, como si estuviéramos en un plano cartesiano y uno de los primeros giros fuese de x a $-x$ y el otro giro de y a $-y$ o inversamente en ambos giros. Los prototipos de giro de x a $-x$ son los siguientes:

Figura 1. 32 RGB LED Magic Ball



Fuente: http://es.youtube.com/watch?v=93HcgS4W_8g (18\08\08)

Este proyecto consiste en una columna de 32 leds RGB multiplexados y manejados por un micro controlador, dando alusión del movimiento por medio del motor.

Figura 2. RGB LED POV display



Fuente: <http://es.youtube.com/watch?v=gkIBWwGyreM> (18\08\08)

Este proyecto no tiene base fija se mueve en el mismo sentido que el anterior pero consta de una columna de 8 leds RGB y manipulado con un pic16f628a y por el poco número de leds no necesita multiplexación. Pero el software implementado es mínimo siendo repetitivo el mensaje.

Los prototipos de giro de y a –y son los siguientes:

Figura 3. Led display



Fuente: <http://www.youtube.com/watch?v=yF1iAYHVyC8&NR=1> (18\08\08)

Este proyecto tiene una columna en forma de leds RGB en forma de triángulo isósceles, lo que permite ocupar más espacio de la circunferencia, para completar un giro la columna de leds se mueve 12 veces y evidencia que al adquirir mayor velocidad mejor imagen.

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

En la actualidad el desarrollo de imágenes holográficas no fue evolucionando del mismo modo que el desarrollo de tipos de leds lo cual hizo que el mercado electrónico se enfocara en producir matrices de punto, para suplir la necesidad de imágenes lo cual proporciono el uso innecesario de leds, y mayor cantidad de consumo de corriente. Además el uso de las matrices ocupa mucho espacio en la implementación de cualquier sistema. A raíz de esto surge la pregunta:

¿Qué velocidad de giro e intensidad deben tener los leds RGB para generar una imagen holográfica tanto estática como en movimiento supliendo una matriz de punto?

1.3 JUSTIFICACIÓN

Se realiza este proyecto con el fin de dar a conocer una de las diferentes aplicaciones que tiene los nuevos procesos tecnológicos en el mundo y para mostrar cómo se lleva a cabo el desarrollo y la evolución de aplicaciones novedosas, que permitirán ofrecer nuevos servicios en los crecientes requerimientos de los usuarios día a día.

Las nuevas propuestas muestran el afán de los desarrolladores de sistemas de entretenimiento en hacerlos cada vez más reales, y estos sistemas se están enfocando en las imágenes holográficas que entregan a los usuarios una percepción casi igual a la realidad. Con el desarrollo de la tecnología se encuentran diferentes medios para generar una imagen holográfica, la aplicabilidad y la calidad de las imágenes es la que hace la diferencia entre estos sistemas y es precisamente lo que tenemos que tener en cuenta en el desarrollo de este proyecto para lograr encontrar un campo en el mercado, en el cual nos podamos desarrollar.

Se presenta entonces el desarrollo e implementación de un mecanismo electrónico que proyecta la imagen deseada, mediante la rotación de una columna de leds RGB a una velocidad determinada, se da una apreciación subjetiva de la imagen, esta sensación se produce en respuesta a la estimulación del ojo y a sus mecanismos nerviosos por la energía luminosa de ciertas longitudes de onda, generando una ilusión de imagen que combinada con pequeñas variaciones de la imagen producen el video o imágenes en movimiento.

Los constantes descubrimientos de nuevas tecnologías hacen evolucionar inevitablemente la mentalidad de las personas. Es por esto, que basándose en los nuevos procesos electrónicos que se imponen en el mercado, se ha optado por desarrollar esta aplicación que explota las tecnologías de vanguardia de los hologramas y su adaptabilidad a la vida cotidiana.

Uno de los motivos por los cuales se pensó en implementar este sistema de entretenimiento es por la necesidad de ciertos usuarios de obtener tecnología de punta y que por su poco poder de adquisición, no pueden adquirirla, la idea es producir sistemas que puedan estar al alcance y entregar la posibilidad de obtener esta nueva innovación a la que estamos llegando y permitir que estos usuarios dejen a un lado el ser espectador y formen parte del cambio.

1.4 OBJETIVOS DE LA INVESTIGACIÓN

1.4.1 Objetivo General Desarrollar un sistema electrónico funcional que proyecte imágenes holográficas de la mejor calidad posible empleando leds RGB.

1.4.2 Objetivos Específicos

- Determinar el tipo y forma de leds RGB más adecuado para que el sistema entregue la mejor calidad de imagen posible.
- Determinar el hardware más adecuado para el control de los leds
- Diseñar el programa adecuado para el sistema de control de encendido de los leds RGB dependiendo de la velocidad de giro del motor.
- Discriminar la técnica de giro más adecuada para generar la mejor calidad de imagen

1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

El sistema estará en la capacidad de generar imágenes holográficas e imágenes en movimiento, en donde la calidad será limitada por los leds RGB que se encuentran en el mercado. Contando con una parte de software que solo soporta un tipo de imagen, y en cuanto a la parte de hardware la distancia que tendrá entre el eje de giro y la columna de leds no será mayor a 25 cm y el largo de la columna de leds no será mayor a 15 cm.

2. MARCO DE REFERENCIA

2.1 MARCO TEÓRICO - CONCEPTUAL

Para ofrecer el servicio de mando o control por medio del holograma se deben tener en cuenta los diferentes requerimientos, tanto en el lugar de trabajo, como los procesos en cada una de las aplicaciones que se desee implementar.

Es importante tener la más completa información acerca de las condiciones del recinto donde el usuario va a operar el sistema, ya que dependiendo de tales condiciones, el dispositivo podrá o no operar en forma óptima.

De otra parte, es indispensable tener una adecuada sincronización y configuración de cada uno de los procesos que se ejecuten en los circuitos implementados y de los mandos o controles generados desde el micro controlador, con el fin de tener un sistema de control confiable; dado que se trata de manipular la frecuencia de transmisión, la sincronización entre los diferentes dispositivos y circuitos es esencial, para lograr un adecuado funcionamiento de los diferentes módulos que conforman el sistema.

2.1.1 La Luz y el Ojo Humano. Muchos de los equipos electrónicos están pensados y desarrollados para suministrar cierta clase de información susceptible de ser percibida por alguno de los sentidos de la persona humana, en especial el oído y la vista.

2.1.2 La Luz. El experimento de Newton, muestra que al hacer pasar un rayo de luz blanca a través de un prisma, ésta se descompone en colores que van desde el rojo al violeta, pasando por el naranja, amarillo, verde, azul y añil. Puede afirmarse que la luz blanca no existe como tal en la Naturaleza, sino que es la suma de todo el conjunto de radiaciones visibles que, al integrarse en nuestro ojo, nos producen la sensación de blanco.

Si aislamos uno solo, de entre los colores que atraviesan un prisma, y que, a su vez, se le hace pasar por otro prisma más para ver en qué se descompone, se comprobará que el color resultante es el mismo que se había introducido. Se habrá aislado, una radiación monocromática pura.

Estas radiaciones monocromáticas son ondas de tipo electromagnético, que se caracterizan, al igual que las ondas electromagnéticas, por su longitud de onda y una frecuencia determinada; entre ambas se establece la siguiente relación:

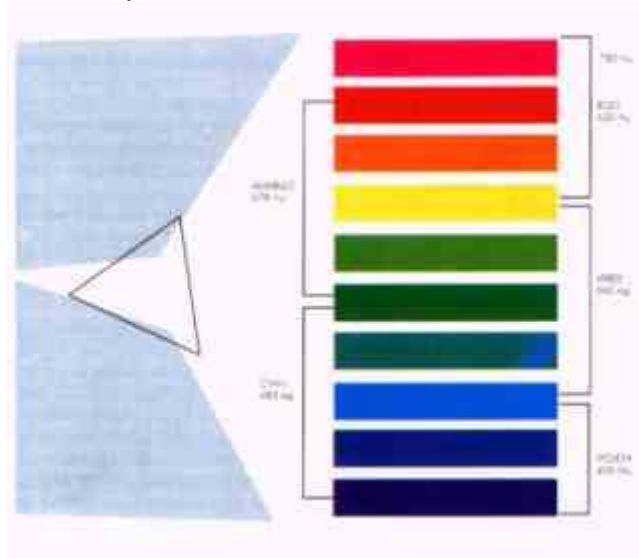
$$\lambda = c / f_1$$

Donde "c" representa la velocidad de la luz que es, aproximadamente, de 300.000 Km/s en el vacío absoluto.

Con un instrumento que mida la energía radiada por un objeto emisor de luz blanca, se obtiene una curva que indica la composición espectral de dicha luz blanca. Al estudiar este espectro, se observará que más allá de los dos extremos en los que el ojo humano es capaz de percibir sensaciones, el instrumento sigue midiendo una cierta energía procedente de radiaciones cuya longitud de onda no es perceptible por el mismo.

¹ http://es.wikipedia.org/wiki/Velocidad_de_la_luz

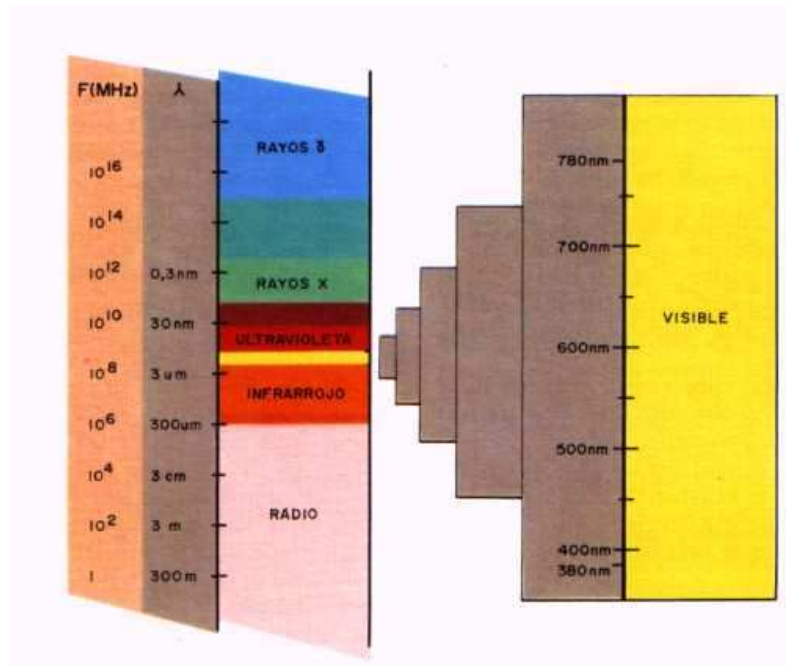
Figura 4. Composición espectral de la luz blanca



Fuente: <http://www.siste.com.ar/serv021.htm>

El experimento de Newton demuestra que la luz blanca no existe, sino que es una percepción subjetiva de nuestro cerebro debido a nuestro sistema de visión.

Figura 5. Longitudes de onda de luz



Fuente: <http://www.siste.com.ar/serv021.htm>

Las radiaciones luminosas que nuestro ojo es capaz de percibir constituyen una pequeña franja del conjunto de las ondas electromagnéticas.

Las radiaciones electromagnéticas que percibe el ojo humano, conforman una pequeña banda que va desde los $3,85 \times 10^8$ MHz, a los $7,9 \times 10^8$ MHz

Estas radiaciones electromagnéticas se transmiten en forma de "cuantos de energía" (son los denominados fotones) que, dependiendo del foco que los emite, tienen una frecuencia determinada. La energía que portan los fotones es directamente proporcional a la frecuencia de la señal emisora y a una constante llamada constante de Planck. De tal manera que:

$$e = h \times f$$

Fuente: <http://www.siste.com.ar/serv021.htm>

Donde "h" es la constante de Planck y "f" la frecuencia del fotón.

Toda radiación electromagnética, visible o no, transporta una cierta cantidad de energía que depende de la propia energía del fotón y del número de fotones recibidos.

Estas cantidades de energía, tanto emitidas como recibidas, han sido ponderadas y delimitadas creándose unidades para que puedan ser medidas.

Si se dispone de una esfera con un cierto radio "R", que se toma como unidad, y en cuyo centro existe un punto emisor radiando fotones en todas direcciones y con la misma intensidad y que forma un cono sólido desde el centro de la esfera a su superficie. La base de este cono no será plana, sino que tendrá una cierta curvatura, y su superficie se mide en estereorradianes. El estereorradián es el ángulo que proporciona una unidad de superficie de la base del cono formado. Si el radio de la esfera tuviese 1 m. de longitud, un estereorradián será aquel ángulo que formaría un cono con una superficie en la base de $1m^2$. La intensidad del foco emisor se mide en "candelas" y proporciona la cantidad de luz emitida que pasa por el ángulo sólido de un estereorradián. A la cantidad de energía que atraviesa la superficie del cono sólido en la unidad de tiempo se le denomina flujo luminoso, y se mide en "lumen".

La candela, el lumen y el estereorradián guardan relación entre sí, de tal manera que:

$$1 \text{ candela} = 1 \text{ lumen} / 1 \text{ estereorradian}^2$$

Si se sitúa en el lugar del cuerpo receptor de luz, la cantidad de flujo luminoso recibido por una unidad de superficie se le denomina iluminación y viene expresado en "lux", estableciéndose la equivalencia siguiente:

$$1 \text{ lux} = 1 \text{ lumen} / 1 \text{ m}^2$$

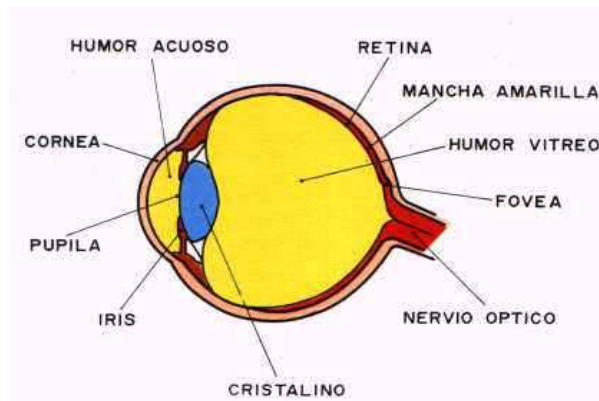
2.1.3 El Ojo Humano y la Visión. La estructura de una cámara fotográfica es comparable con la constitución del ojo humano. La cámara negra del ojo está constituida por una esfera llena de un líquido denominado humor vítreo, el objetivo lo constituyen la córnea y el cristalino; el diafragma, el iris con la pupila; la superficie fotosensible, la retina; y el obturador, sería el párpado.

Si se comienza por la recepción de una imagen, se puede afirmar que lo que llega al ojo sólo es la luz reflejada por los objetos, excepción hecha de los cuerpos emisores.

Figura 6. Partes de ojo humano

² //www.siste.com.ar/serv021.htm

³ //www.siste.com.ar/serv021.htm



Fuente: <http://www.siste.com.ar/serv021.htm>

Así, dependiendo de la intensidad luminosa que llegue al ojo humano, la pupila se ajusta de forma automática e involuntaria, abriendo o cerrando más o menos la apertura del iris para que la imagen recibida al pasar por el cristalino se proyecte sobre la retina.

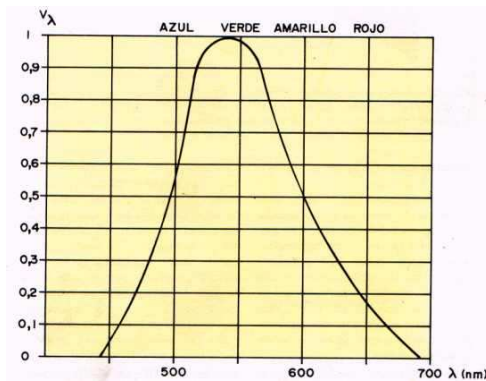
El cristalino es una lente biconvexa de foco variable que proyecta la imagen sobre la retina llevándola al punto de la misma en que la imagen es más nítida y con un correcto enfoque.

La córnea es una membrana que envuelve al ojo y que encierra el humor acuoso, un líquido viscoso aposentado delante de la pupila que, junto al humor vítreo contenido en la cámara posterior, sirve para mantener turgente al ojo.

De todos los componentes del ojo, el más importante es la retina. Es aquí donde se realiza la percepción de la imagen. Es la décima capa de una película formada por diez. Las nueve primeras están compuestas de neuronas, en su mayor parte, y son atravesadas sin ningún tipo de dificultad por los fotones provenientes del exterior.

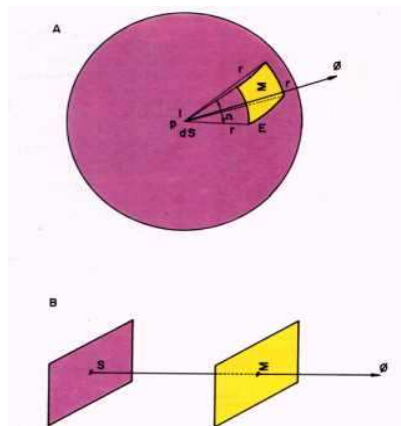
La retina está constituida por una enorme cantidad de captadores fotosensibles, cada uno de los cuales da en cada momento información correspondiente de la intensidad y el color de la radiación incidente. Estos captadores fotosensibles son de dos tipos: conos y cilindros o bastones; se los conoce con estos nombres debido a su forma. Los conos y los cilindros son las terminaciones de una gran cantidad de fibras nerviosas que forman el nervio óptico, el cual se encarga de transmitir todas las informaciones a las superficies estriadas del cerebro situadas en el lóbulo occipital. Los conos poseen un enlace nervioso individual con el cerebro, mientras que los bastones van unidos a un mismo conducto nervioso por grupos que pueden llegar a ser miles de ellos. De aquí que los conos den mensajes sensoriales más precisos y que los bastones tengan una mayor sensibilidad, incluso con niveles de energía luminosa muy bajos.

Figura 7. Curva de sensibilidad relativa del ojo en función de las longitudes de onda de las radiaciones percibidas.



Fuente: <http://www.siste.com.ar/serv021.htm>

Figura 8. Esfera con un foco puntual de radiación uniforme y omnidireccional



Fuente: <http://www.siste.com.ar/serv021.htm>

Esfera con un foco puntual de radiación uniforme y omnidireccional. Angulo sólido, cuya unidad es el estereorradián, desde el que se ve una superficie (S) desde el foco puntual: es la superficie que determina en la esfera de radio unidad a la figura geométrica que tiene por vértice el punto y , cuyas aristas, pasan por el contorno de la superficie.

Todo lo que se sabe acerca de la recepción de una imagen por el ojo, y la interpretación que de ella realiza el cerebro, se ha obtenido de una manera experimental estudiando y comparando el comportamiento del ojo ante los estímulos a que era sometido.

Examinando al microscopio la superficie de la retina se comprueba que la densidad de los conos es mucho mayor que la de los bastones en una zona

situada en el extremo opuesto de la pupila, denominada mancha amarilla. La parte central de esta zona, denominada fovea, está desprovista totalmente de bastones, mientras que la densidad de conos es máxima.

En la fovea, los conos son de tres tipos diferentes dependiendo del tipo de radiación a la que sean sensibles.

Estos conos permiten realizar la percepción cromática de las imágenes. Los bastones son los encargados de la luminancia, o visión monocroma (de blanco y negro). Este hecho, junto con la mayor sensibilidad de los bastones, explica el porqué, ante bajos niveles de iluminación, se dejan de percibir los colores, y la imagen que nos queda es en blanco y negro (a la noche todos los gatos son pardos).

Los conos y los cilindros están inmersos en un medio líquido que toma instantáneamente color blanco ante la presencia de fotones, volviendo más lentamente a su estado inicial cuando se produce el cese de fotones en la fuente emisora. Este retardo en la vuelta al estado inicial, se admite que es debido al proceso inverso de la reacción fotoquímica que se produce ante la presencia de fotones y tiene una duración de, aproximadamente, 50×10^{-3} s.

2.1.4 Curva de Sensibilidad Relativa. La sensibilidad del ojo con respecto a las radiaciones colorimétricas no es uniforme en toda la banda del espectro visible, es decir, para tener la misma sensación subjetiva de intensidad luminosa de dos colores diferentes, una de dichas intensidades será más elevada que la otra, dependiendo de la posición de su frecuencia dentro del espectro.

Figura 9. Experimento del trébol de Young.



Fuente: <http://www.siste.com.ar/serv021.htm>

El experimento del trébol de Young muestra cómo se realizó la mezcla aditiva de tres colores primarios, rojo, verde y azul, y que pueden producir la sensación de blanco.

Tabla 1. Diferencia de sensibilidad.

l (nm)	VI	l	VI	L	VI
475	0.1126	515	0.6082	645	0.1382
480	0.1399	520	0.7100	650	0.1070
485	0.1693	550	1.0000	655	0.0816
490	0.2080	600	0.4900	660	0.0610
495	0.2586	625	0.3210	665	0.0446
500	0.3230	630	0.2650	700	0.0010
505	0.4073	635	0.2170	750	0.0001
510	0.5030	640	0.1750	770	0.0000

Fuente: <http://www.siste.com.ar/serv021.htm>

2.1.5 Integración de la visión. Para entender mejor esta propiedad se puede partir de un pequeño experimento:

Se dirigen tres focos de luz, uno rojo, otro verde y otro azul, hacia la superficie de un mismo plano, de tal manera que haya una zona donde incidan los tres focos a la vez. También habrá áreas en las que sólo incida un foco y sectores donde lo hagan dos. En las zonas donde hay varios colores incidentes, el ojo (o más bien el cerebro, ya que es aquí donde se produce esta característica) es incapaz de distinguir los colores que inciden sobre la zona, puesto que lo que se percibe es la integración de dichos colores. Así, por ejemplo, en la zona donde incidan los tres focos se tendrá la sensación de blanco, mientras que en las zonas de dos colores se percibe, según el caso, la mezcla de ambos. Cuando la mezcla de colores está formada por radiaciones situadas en los dos extremos del espectro visible, el rojo y el violeta, la impresión coloreada que se percibe no corresponde con ninguna de las radiaciones monocromáticas que se obtiene cuando se hace que la luz solar atraviese un prisma. Se obtiene, así, una gama de colores subjetivos denominada gama de los púrpuras.

Al igual que esta sensación subjetiva, hay tres parámetros psicofísicos que hace diferenciar una percepción coloreada de otra.

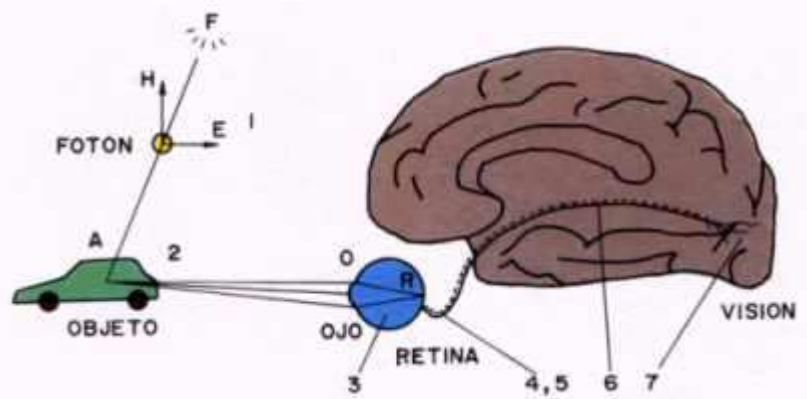
2.1.6 La luminancia. En una imagen en blanco y negro (monocromática) se atribuye más luminancia, brillo, a sus partes blancas que a las grises, y más a

las grises que a las negras. Si ahora se observa una imagen en color, se establecen las mismas diferencias entre las partes muy iluminadas y las que quedan en la sombra, independientemente del color que tengan. El ojo no responde de igual manera ante igual intensidad de iluminación de dos colores diferentes; y así, ante dos colores con diferente iluminación, se produce la sensación de que tienen la misma. A la energía luminosa percibida se le denomina luminancia y es diferente de la realmente emitida.

Los experimentos realizados demuestran que el ojo puede distinguir cerca de mil niveles de luminancia diferentes, desde el umbral de percepción hasta el de deslumbramiento.

2.1.7 El Tono o Matiz. El Tono se puede definir como el atributo que permite diferenciar y separar la longitud de onda predominante de la señal recibida, es decir, la visión distinta de los colores cromáticos. Son, aproximadamente, 250 tonos diferentes los que el ojo humano es capaz de distinguir, a los que hay que añadir los correspondientes a la gama de los púrpuras.

Figura 10. Fases en que se puede dividir el fenómeno de la visión.



Fuente: <http://www.siste.com.ar/serv021.htm>

2.1.8 Saturación. La saturación da la noción de pureza de color o, lo que es lo mismo, la mayor o menor mezcla de blanco con el color cromático, proporcionando la posibilidad de distinguir un color vivo de un color pálido. La cantidad de colores distinguibles, en función de su tono y saturación, es del orden de 20.000. Por lo tanto, la cantidad de estímulos visuales distinguibles, en función de estos tres parámetros, supera el millón, y quedan perfectamente definidos en función de los mismos.

2.1.9 Agudeza visual. Debido a que los elementos perceptores, conos y bastones, de la superficie de la retina no están unidos unos a otros, sino que hay una cierta distancia entre ellos (alrededor de 2×10^{-6} m. en la zona más densa), dos fuentes concretas, próximas entre sí, proyectarán su imagen sobre el mismo cono, con lo que el cerebro sólo recibirá una información sensorial. Al ser la distancia focal del cristalino de 2 cm., el mínimo ángulo que pueden

formar los dos rayos luminosos para tener una percepción separada es de medio minuto. Aunque, habitualmente, se toma como media el ángulo de 1 minuto.

Este límite de agudeza visual es muy importante en el campo de la televisión, ya que no tiene mucho sentido reproducir imágenes con una definición por encima de este ángulo separador.

Por otro lado, aprovechando esta cualidad del ojo, podemos conseguir que con tres puntos muy próximos entre sí con información de color, tan sólo se pueda percibir la integración de los tres puntos luminosos. Esta es la base de los sistemas de televisión en color actualmente explotados.

2.1.10 Memoria visual. El efecto que se produce en la retina al incidir los fotones de luz sobre el líquido en el que están inmersos los conos y bastones, y el tiempo que tarda, aproximadamente, una imagen en desaparecer, como tal efecto percibido, una vez que ha cesado el estímulo es de 50×10^{-3} s.

Como media. Si antes de que transcurra este tiempo, el ojo recibe un nuevo estímulo, la impresión visual que se tendrá será la suma de las dos. Si el tiempo de separación es mayor de 50×10^{-3} s. La visión es distinta, produciéndose el efecto de parpadeo.

Esta propiedad es utilizada en la reproducción de imágenes animadas, tanto en televisión, donde se proyectan 25 imágenes por segundo, como en el cine, donde la velocidad es de 24 imágenes.

La memoria visual es válida también para estímulos de cromaticidad diferentes, donde la suma de estímulos sucesivos producirá la misma sensación que la ocasionada por una radiación continua compuesta por diferentes radiaciones que coinciden con el grupo de estímulos sucesivos. Si se divide un círculo en siete sectores, cada uno de ellos con uno de los colores del espectro visible y se lo hace girar por el eje que pasa por su centro a una velocidad suficiente, se percibirá un blanco grisáceo, indicando que se ha realizado una integración de los colores.

El aprovechamiento de estas características del ojo para el desarrollo de los sistemas de televisión ha sido determinante a la hora de conseguir equipos de un diseño no demasiado complicado y con costos no muy elevados.⁴

⁴ http://www.siste.com.ar/vision_del_ojo.htm

3. METODOLOGÍA

3.1 ENFOQUE DE LA INVESTIGACIÓN

El enfoque a emplear en la investigación y de acuerdo a las políticas para la investigación establecidas por la Universidad, es:

Empírico- analítico: Cuyo interés es el técnico, orientado a la interpretación y transformación del mundo material; proporciona una estructura particular a la metodología de investigación en tanto que orienta el trabajo a la contrastación permanente de las aseveraciones teóricas con la verificación experimental, de manera que los cálculos generados a través de modelos matemáticos y simulaciones computacionales se deben retroalimentar con la experiencia, en la búsqueda de información cada vez más confiable y practica para la solución del problema.

3.2 LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA

3.2.1 TECNOLOGÍAS ACTUALES Y SOCIEDAD: La sociedad requiere de conocimientos técnicos y científicos de vanguardia que ayuden a la solución de problemas o faciliten los procesos de mejoramiento de la calidad de vida de las personas que pertenecen a un grupo social determinado.

3.2.2 INSTRUMENTACIÓN Y CONTROL DE PROCESOS: Procesos asociados con la interacción de los diversos elementos que constituyen mecanismos generadores de potencia como lo son los motores eléctricos.

3.2.3 MICROELECTRÓNICA: Con el desarrollo de este campo el hombre ha avanzado en los últimos 40 años. Es por ello que se hace necesario el estudio de éste para seguir en el avance y el mejoramiento de la calidad de vida de la humanidad

La microelectrónica es fundamental permite el desarrollo de aplicaciones como:

- Juegos.
- Amplificadores especiales.
- Emuladores de otros dispositivos.

- Instrumentación.
- Sensórica.

3.3 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

Se utilizo el software MPLAB IDE v7.20 para la programación de los pics y comprobación de las frecuencias de funcionamiento del dispositivo, además para mayor interpretación de los comandos utilizados en lenguaje assembler, se simulo en proteus que recrea la manipulación del software en el micro controlador con todos los componentes es decir los leds y el motor de giro. Y para mayor veracidad de la utilización de los dispositivos con el prototipo se evidencia los resultados.

4. DESARROLLO INGENIERIL

4.1 SELECCIÓN DE GIRO

Se inició por elegir cuál sería la mejor forma de giro del dispositivo, la cual se podría realizar en dos formas, es decir si marca un plano cartesiano, un giro sería sobre el eje x y el otro sobre el eje y, para ello se realizó la tabla 2 donde se puede observar un comparativo teniendo en cuenta los tres ítems de mayor relevancia del sistema.

Tabla 2. Comparación de ejes

Características	Eje x	Eje y
Visualización	<ul style="list-style-type: none">• Garantiza una visualización del dispositivo de 360 grados.	<ul style="list-style-type: none">• Una visualización casi unipersonal, debido al poco espacio visual que genera.
Imagen	<ul style="list-style-type: none">• Como el dispositivo de control de imagen está detrás del eje de los leds la visualización no se recorta.	<ul style="list-style-type: none">• Imagen recortada en su eje de rotación, debido a la implementación del dispositivo en dicha posición
Diseño	<ul style="list-style-type: none">• El diseño del prototipo tiene más complejidad debido a la posición de la columna de los leds	<ul style="list-style-type: none">• El diseño del prototipo es más eficaz debido a que en un aspa de un ventilador se realiza la posición de leds.

Debido a lo referenciado en la tabla 2, se optó por el giro en el eje x, porque ofrece mejor visualización e imagen que en el eje y.

Una vez hecho esto se inicia las pruebas para generar imágenes, para ello se usa una columna de leds planos azules así como se puede observar en la figura 11.

4.2 PRUEBAS DE GIRO

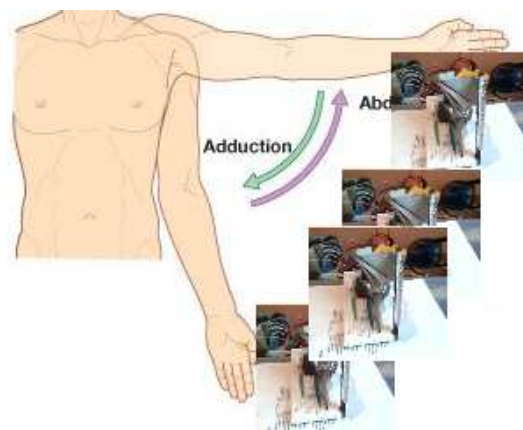
Figura 11. Primer prototipo.



En la figura 11 se puede apreciar el funcionamiento on /off de los leds con un microcontrolador al libre albedrío en posicionamiento de puertos.

Usando una simple multiplexación de cátodos de los leds para solo usar los ánodos en cada puerto del micro, en esta fase se usaron tres puertos del micro para 24 leds y se logro desarrollar la fase inicial del programa, identificando letras.

Figura 12. Forma de giro del primer prototipo



Debido a que en esta fase no hay la posibilidad de rotación, se hizo manualmente el movimiento de izquierda a derecha así como se muestra en la figura 12, con una velocidad constante con la fuerza del brazo, para evidenciar la ilusión óptica de ver las letras, las cuales fueron las vocales, de esa forma se mostro que se puede reemplazar la multiplexación usada en matrices de

punto. Y con la manipulación del programa con tiempos de encendido y apagado del Led se hacían más gruesos y delgados la imagen a ilustrar.

4.2.1 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO 1

Para desarrollar el programa en esta fase primero se hizo un diseño de cada letra no superior a 3x8 cuadros, así como se puede observar en la figura 12.1, donde se ve claramente la letra A, que es sacar uno en el led que se quiere ver prendido y cero el led que se quiere ver apagado. El lector puede obtener los códigos de las demás letras solo es saber que led prende y cual apaga.

Figura 12.1. Diseño de imagen a montar en el software primer prototipo

	1	2	3	4	5
B0		0	1	0	
B1		1	0	1	
B2		1	0	1	
B3		1	0	1	
B4		1	0	1	
B5		1	1	1	
B6		1	0	1	
B7		1	0	1	
Dato hexadecimal		FE	21	FE	

El alto de cada letra no puede ser mayor a 8 pues solo se tienen 8 leds y el ancho se opta por 3 para no ocupar mayor espacio.

En el diagrama de flujo 1 se muestra diseño del programa para generar la letra A y el programa completo se puede ver en el anexo E

Diagrama 1. Flujograma del primer prototipo

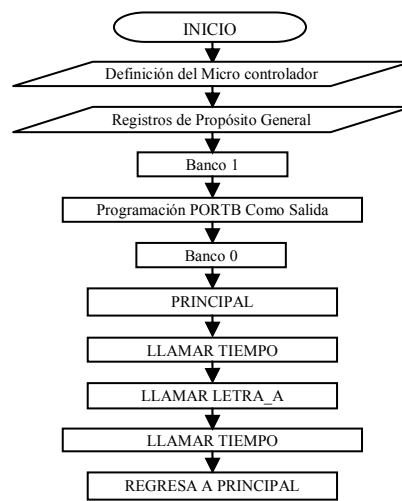


Diagrama 1.1 Flujograma subrutina LETRA_A

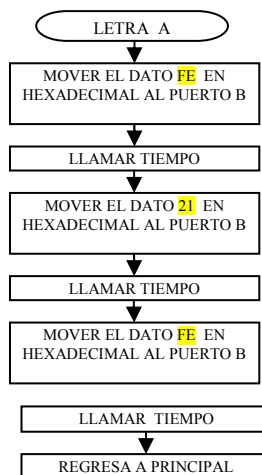
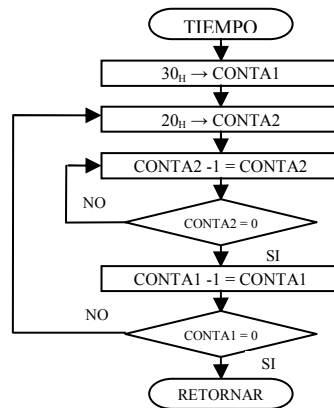


Diagrama 1.2 Flujograma subrutina TIEMPO



En la subrutina del programa llamada LETRA_A , el movimiento de los datos al puerto B se hizo en hexadecimal (ver figura 12.1 , última fila) y se copia en el programa empezando de la columna 4, luego la 3 y por último la 2, no se empieza en la columna dos porque la imagen es un espejo y al transcribir la letras se vería al revés, desde luego en este caso la letra A , no hay inconveniente pero en las demás letras pueden quedar mal y se observarían al revés.(ver video 1)

4.3 SELECCIÓN DEL MOTOR

Después de verificar el perfecto funcionamiento se procedió a determinar cuál es el motor más adecuado para dar movimiento a la columna de leds para comenzar a proporcionar imágenes sin ayuda del movimiento del brazo.

Figura 13. Motor de 12 voltios DC



Figura 14. Motor de 24 voltios DC



Se hicieron pruebas con varios motores DC y AC de diferentes voltajes y corrientes hasta encontrar un motor ideal para el prototipo, las pruebas de dichos motores variaron debido a las cualidades de diseño de cada prototipo y peso del circuito.

Se diseñaron dos prototipos; el primer prototipo con un ventilador de 12 voltios al cual le instalamos una columna de leds de no más de 8 leds planos azules con un micro controlador de 18 pines, todo diseñado en baquela y alimentado con dos pilas de reloj de 1.5 voltios.

Solo manipulamos un puerto del microcontrolador para hacer el diseño no tan pesado y evidenciar resultados. Ver figura 15

Figura 15. Prototipo funcional con el motor de 12 voltios.



La finalidad de este primero, era demostrar que el primer diseño de programa era funcional y ver la rotación del mensaje en forma de circunferencia como se logra evidenciar en esta figura 16.

Figura 16. Visualización de letras en el prototipo funcional.



Figura 17. Visualización de mensajes en el prototipo funcional.

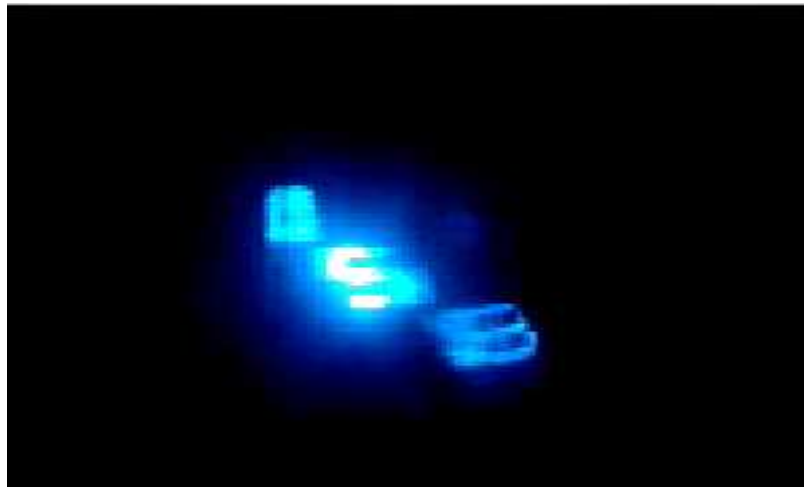


- Hay que tener en cuenta que hay dos baterías independientes para la alimentación del diseño una para el micro y otra para el movimiento del motor.
- Las bases que sostienen la columna son hechas en metal y balsa.

El tercer prototipo se realizó con un motor de 24 voltios y manipulamos una columna de 29 leds a chorro planos azules, en este prototipo ya con la excelente formación de imágenes pero muy básicas que eran solo letras, y no mayor a palabras compuestas por cuatro letras como “USB” y “hola” entre otras, dimos el siguiente paso, al añadir más leds para proporcionar mayor diversidad de imágenes.

Cambiamos de micro otra vez por el de 40 pines debido a que manipulamos más leds, y diseñamos figuras como “USB” pero representada cada letra en un puerto de 8 bits, un puerto por cada letra.

Figura 18. Visualización de mensaje el prototipo 3 funcional de 24 voltios.



4.3.1 CÁLCULO DE LA POTENCIA DEL MOTOR

El problema que se plantea es el de elevar una carga de Q [Kg.], desplazándose con una velocidad de v [m/seg.]; la potencia que debe desarrollar la máquina en esta situación es $P = Q \times v$ [Kg m/seg.]. Si llamamos η al rendimiento de la máquina, la potencia del motor se puede escribir como:

$$P = \frac{Q \cdot v}{\eta} \text{ [Kg} \cdot \text{m / seg]}$$

Las ecuaciones de potencia⁵.

Como 1 [Kg m/seg.] equivale a 9,81 [w.], resulta:

$$P = \frac{Q \cdot v}{102 \cdot \eta} \text{ [Kw.]}$$

El par motor será igual a:

⁵ HARPER, Gilberto Enriquez, Generadores, transformadores y motores eléctricos, Primera Edición. Madrid: Alfa y Omega.

$$P = M \cdot n \qquad P = M \cdot \frac{2 \cdot \pi}{60} \cdot n$$

$$M = \frac{P}{n \cdot \frac{\pi}{30}} \text{ [Nw} \cdot \text{m.]} = \frac{P}{n \cdot \left(\frac{\pi}{30} \cdot 9,81 \right)} \text{ [m} \cdot \text{Kg.]}$$

$$M = 975 \cdot \frac{P}{n} \text{ [m} \cdot \text{Kg.]} = 9,55 \cdot \frac{Q \cdot v}{n \cdot \eta} \text{ [m} \cdot \text{Kg.]}$$

Donde **P** es la potencia del motor en [Kw] y **n** es la velocidad de giro en [r.p.m.]

Los prototipos que se realizaron con excepción del prototipo 2 tienen una carga de aproximada de 1 kg, lo cual nos permite tomar este dato como general para poder hacer las medidas de potencia del motor (se descarta el prototipo 2 no solo por la carga, sino por la poca cantidad de leds que tiene), la velocidad $v=1$ [m/seg.]. El rendimiento del mecanismo es $\eta = 0,85$ y la velocidad de giro es de $n=850$ r.p.m

Potencia:

$$P = \frac{Q \cdot v}{102 \cdot \eta} \text{ [Kw.]}$$

$$\left| \frac{1 \cdot 1}{102 \cdot 0,85} = \right|$$

$$= 0,015 \text{ KW}$$

Par motor:

$$M = 9,55 \cdot \frac{Q \cdot v}{n \cdot \eta} = 9,55 \cdot \frac{1 \cdot 1}{850 \cdot 0,85}$$

$$M = 0,0132 \text{ m x Kg}$$

También se determino el tiempo relativo de conexión, el par motor medio y la potencia media del motor. Considerar que en el servicio tanto la elevación como la bajada se realizan con plena carga de $Q=1 \text{ Kg}$ o 1000 g ; considerar que la velocidad a mecanismo lanzado o velocidad de régimen es de $v_b=1 \text{ m/seg.}$. El mecanismo rinde el 80% cuando a plena carga gira a $n=1000 \text{ r.p.m.}$ se conoce el módulo total de inercia que vale $\Sigma GD^2=5 \text{ Kg m}^2$.

Datos:

$$\begin{array}{ll} Q=1 \text{ Kg.} & \eta=0,8 \\ v_b=1 \text{ m/seg.} & n=1000 \text{ r.p.m.} \\ s=5 \text{ m.} & \Sigma GD^2=5 \text{ Kg m}^2. \end{array}$$

Vamos a denominar con:

Subíndice (a): periodo de aceleración
Subíndice (b): mecanismo lanzado
Subíndice (c): periodo de deceleración

4.3.1.1 ELEVACIÓN DE LA CARGA

Marcha a velocidad de régimen ó a mecanismo lanzado

Par

$$\begin{aligned} M_{1b} &= 9,55 \cdot \frac{Q \cdot v}{n \cdot \eta} = 9,55 \cdot \frac{1000 \cdot 1}{0,8 \cdot 1000} \\ &= 12 \text{ [m} \cdot \text{ g.]} \end{aligned}$$

Potencia

$$P_{1b} = \frac{Q \cdot v}{102 \cdot \eta} = \frac{1000 \cdot 1}{102 \cdot 0,8} = 12,25 \text{ [Kw.]; ó } P_{1b} = \frac{M \cdot n}{975} = \frac{12 \cdot 1000}{975}$$

$$= 12,25 \text{ [w.]}$$

Suponemos un par acelerador de 1,5 veces.

$$M_a = 1,5 \cdot M_{1b} = 1,5 \cdot 12 = 18 \text{ [m \cdot g.]}$$

Según la curva de revoluciones, en este caso el número de revoluciones desciende hasta el 90% □ 900 r.p.m.; puesto que el par motor en el arranque es 150% del par nominal (1,5 veces el par nominal).

$$t_{1a} = 2,67 \cdot \frac{n}{1000} \cdot \frac{\sum GD^2}{M_{1a} - M_{1b}} = 2,67 \cdot \frac{900}{1000} \cdot \frac{5}{18 - 12} = 2,1 \text{ [seg.]}$$

El recorrido de aceleración:

$$s_{1a} = \frac{v_{1b}}{2} \cdot t_{1a} = \frac{1}{2} \cdot 2,1 = 1,05 \text{ [m.]}$$

La potencia de punta durante la aceleración es igual a:

$$P_{1a} = 2,74 \cdot \left(\frac{n}{1000} \right)^2 \cdot \frac{\sum GD^2}{t_{1a}} + P_{1b} = 2,74 \cdot \left(\frac{900}{1000} \right)^2 \cdot \frac{5}{2,07} + 12,25 = 17,6 \text{ [Kw.]}$$

$$P_{1a} = 2,74 \cdot \left(\frac{n}{1000}\right)^2 \cdot \frac{\sum GD^2}{t_{1a}} + P_{1b} = 2,74 \cdot \left(\frac{900}{1000}\right)^2 \cdot \frac{5}{2,07} + 12,25 = 17,6 \text{ [Kw.]}$$

Suponiendo un tiempo de deceleración, $t_{1c}=2$ seg., nos calculamos el espacio de frenado:

$$s_{1c} = \frac{v_{1b}}{2} \cdot t_{1c} = \frac{1}{2} \cdot 2 = 1 \text{ [m.]}$$

El espacio recorrido a mecanismo lanzado será:

$$s_{1b} = s - s_{1a} - s_{1c} = 5 - 1,05 - 1 = 2,95 \text{ [m.]}$$

y el tiempo tardado en recorrer este espacio a mecanismo lanzado:

$$t_{1b} = \frac{s_{1b}}{v_{1b}} = \frac{2,9}{1} = 2,9 \text{ [seg.]}$$

El tiempo total para la elevación de la carga será:

$$t_1 = t_{1a} + t_{1b} + t_{1c} = 2,1 + 2,9 + 2 = 7 \text{ [seg.]}$$

4.3.1.1 BAJA DE LA CARGA

Marcha a velocidad de régimen ó a mecanismo lanzado.

$$\text{Par} \quad M_{2b} = -9,55 \cdot \frac{Q \cdot v}{n} \cdot \eta = -9,55 \cdot \frac{1000 \cdot 1}{1000} \cdot 0,8 = -7,65 \text{ [m} \cdot \text{Kg.]}$$

$$\text{Potencia} \quad P_{2b} = -\frac{M_{2b} \cdot n}{975} = -\frac{7,65 \cdot 1000}{975} = 7,85 \text{ [Kw.]}$$

Fijamos en 1 seg. el tiempo de aceleración t_{2a} , el par de giro correspondiente, será:

$$M_{2a} = 2,67 \cdot \frac{n}{1000} \cdot \frac{\sum GD^2}{t_{2a}} - M_{2b} = 2,67 \cdot \frac{1000}{1000} \cdot \frac{5}{1} - 7,65 = 5,7 \text{ [m} \cdot \text{kg.]}$$

Potencia de aceleración:

$$P_{2a} = \frac{M_{2a} \cdot n}{975} = \frac{5,71 \cdot 1000}{975} = 5,86 \text{ [Kw.]}$$

el recorrido de aceleración:

$$s_{2a} = \frac{v_{2b}}{2} \cdot t_{2a} = \frac{1}{2} \cdot 1 = 0,5 \text{ [m.]}$$

Con un par máximo de deceleración de:

$$M_{2c} = 1,5 \cdot \eta \cdot M_{1b} = 1,5 \cdot 0,8 \cdot 12 = 14,4 \text{ [m} \cdot \text{kg.]}$$

Se obtiene un periodo de deceleración de:

$$t_{2c} = 2,67 \cdot \frac{n}{1000} \cdot \frac{\sum GD^2}{M_{2b} - M_{2c}} = 2,67 \cdot \frac{1000}{1000} \cdot \frac{5}{14,4 - 7,64} - 7,65 = 1,3 \text{ [seg.]}$$

Calculamos el recorrido de frenado:

$$s_{2c} = \frac{v_{2b}}{2} \cdot t_{2c} = \frac{1}{2} \cdot 1,3 = 0,65 \text{ [m.]}$$

El espacio recorrido a mecanismo lanzado o marcha en régimen será:

$$s_{2b} = s - s_{2a} - s_{2c} = 5 - 0,5 - 0,65 = 3,85 \text{ [m.]}$$

y el tiempo tardado en recorrer este espacio a mecanismo lanzado:

$$t_{2b} = \frac{s_{2b}}{v_{2b}} = \frac{3,85}{1} = 3,85 \text{ [seg.]}$$

El tiempo total necesario para la bajada de la carga será:

$$t_2 = t_{2a} + t_{2b} + t_{2c} = 1 + 3,85 + 1,3 = 6,15 \text{ [seg.]}$$

4.3.1.2 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO 2

Para desarrollar el programa en esta segunda fase, primero se hizo un diagrama no superior a 15x8 cuadros, dibujando la imagen a diseñar que en este caso es el mensaje hola, representado en la figura 17.1. En dos puertos del microcontrolador un puerto del micro, el puerto B para los ánodos de la columna de 8 leds y el puerto A para los cátodos de los leds multiplexados En una sola línea de datos

Figura 17.1 Diseño de imagen a montar en el software 2 prototipo

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B0	1	0	1		1	1	1		1	0	0		0	1	0
B1	1	0	1		1	0	1		1	0	0		1	0	1
B2	1	0	1		1	0	1		1	0	0		1	0	1
B3	1	0	1		1	0	1		1	0	0		1	0	1
B4	1	1	1		1	0	1		1	0	0		1	1	1
B5	1	0	1		1	0	1		1	0	0		1	0	1
B6	1	0	1		1	0	1		1	0	0		1	0	1
B7	1	0	1		1	1	1		1	1	1		1	0	1

FLUJOGRAMA

Diagrama 2. Flujo grama prototipo 2

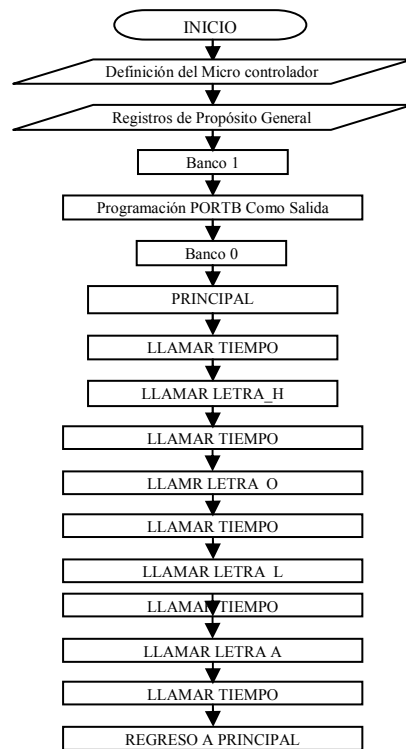


Diagrama 2.1 Flujograma subrutina LETRA_H_1

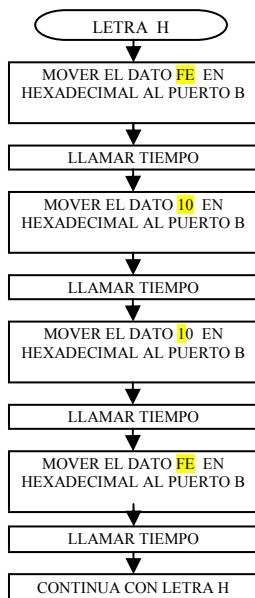


Diagrama 2.2 Flujograma subrutina LETRA_O_1

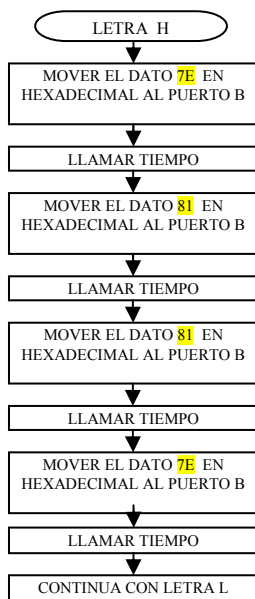


Diagrama 2.3 Flujograma subrutina LETRA_L_1

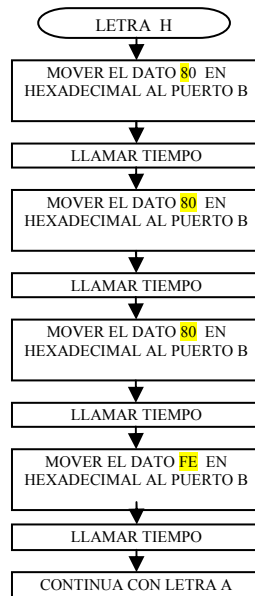


Diagrama 2.4 Flujograma subrutina LETRA_A__1

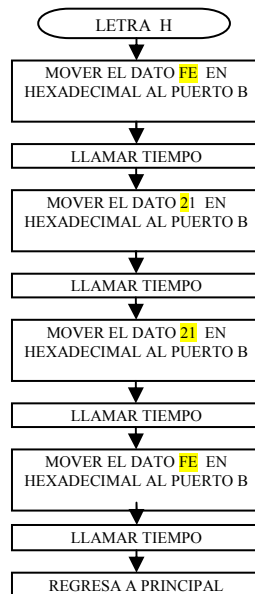
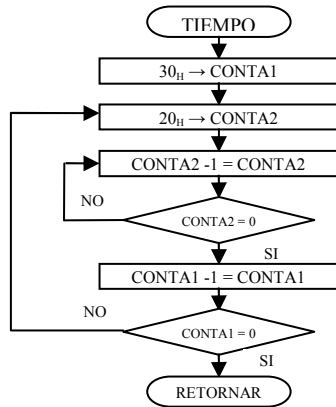


Diagrama 2.5 Flujograma subrutina TIEMPO



Se debe tener en cuenta como en el primer programa que el diseño de las letras se hace empezando por la última columna, en este caso se empieza por la columna 15 hasta llegar a la columna 1, pero se diseña letra por letra como en el primer programa teniendo en cuenta, que la subrutina de tiempo nos proporciona el ancho de la letra y espació entre ellas mismas, para no tener todas las letras pegadas, el programa completo se puede ver en el anexo F

En el programa se realizaron las letras con una columna más que en la figura 17.1, demostrando que las letras se puede forma con mínimo 3 columnas o como las diseñadas en el programa, que son mas estructuradas de 4 columnas, es importan manipular los tiempos de la subrutina igual al de los espacios de las letras , para visualizar una excelente imagen. (Ver video2)

4.3.2 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO 3

Para desarrollar el programa en esta tercera fase, primero se hacen 3 diagramas no superiores a 5x8 cuadros, dibujando la imagen a diseñar que en este caso es el mensaje USB, representado en la figura 18.1. La cual ahora se va a manipular utilizando tres puertos el B, C y D, cada uno de 8 bits, el cual manipula uno de los 24 leds de la columna.

Figura 18.1 Diseño de imagen a montar en el software 3 prototipo

	1	2	3	4	5
B0		1	0	1	
B1		1	0	1	
B2		1	0	1	
B3		1	0	1	
B4		1	0	1	
B5		1	0	1	
B6		1	0	1	
B7		1	1	1	

	1	2	3	4	5
B0		0	1	1	
B1		1	0	1	
B2		1	0	0	
B3		0	1	1	
B4		0	0	1	
B5		0	0	1	
B6		1	0	1	
B7		1	1	1	

	1	2	3	4	5
B0		0	1	0	
B1		1	0	1	
B2		1	0	1	
B3		1	1	1	
B4		1	1	1	
B5		1	0	1	

B6		1	0	1	
B7		1	1	1	

Diagrama 3. Flujo grama prototipo 3.

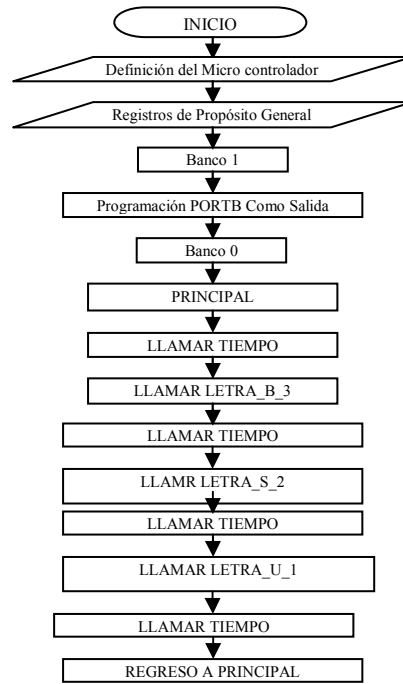


Diagrama 3.1 Flujograma subrutina LETRA_B_3

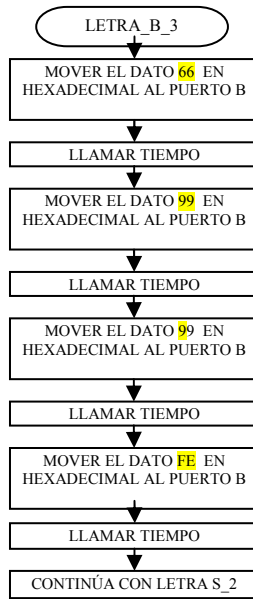


Diagrama 3.2 Flujograma subrutina LETRA_S_2

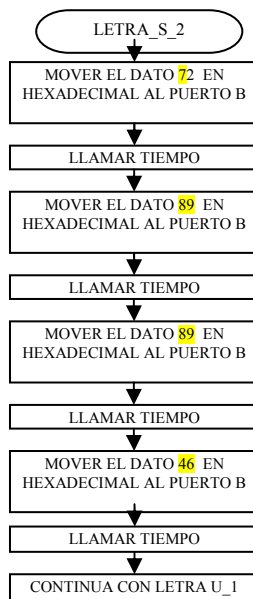


Diagrama 3.3 Flujograma subrutina LETRA_U_1

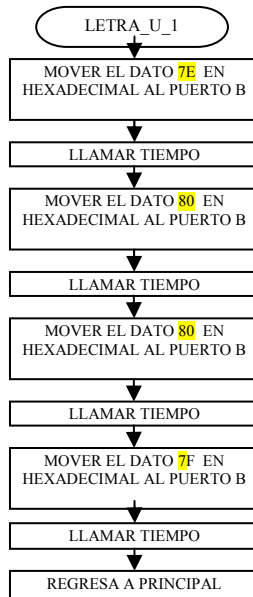
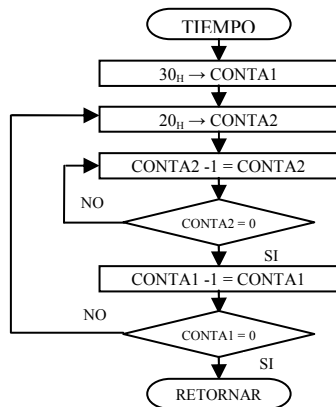


Diagrama 3.4 Flujograma subrutina TIEMPO



En este programa se manipula en subrutina cambio_1, cambio_2 y cambio_3 el puerto a mover la letra realizada, en el programa se realizo que la secuencia de movimiento de los puertos fueran B, C y D y las letras se observan de forma ascendente como en la figura 18.(Ver Video3)

Las observaciones realizadas con las subrutinas tiempos antes mencionadas se mantienen, como lo mencionado de la formación de letras porque el programa se hizo con 4 columnas y en las figuras 18.1 son elaboradas con tres columnas no más. El programa completo se puede ver en el anexo G

4.4 SELECCIÓN DE LEDS

Al ver la forma de desarrollo de letras con 29 leds se continuó con el desarrollo de otro prototipo pero utilizando tres columnas de los colores principales que son el rojo, verde y azul.

En este caso como tenemos tres columnas una de cada color, utilizamos la multiplexación de todos los cátodos por columna, y con los ánodos se unieron las filas (quiere decir que las tierras de cada columna se manipulan con un solo pin ósea que con solo tres pines manipulamos el encendido del rojo, verde y azul si el ánodo estuviera en vcc), la idea de llegar hasta ahora a la manipulación de este tipo de leds es debido a que en el momento de desarrollo del mismo en el mercado colombiano, la facilidad de adquirir el tipo de Led RGB no era viable y abordaba precios exorbitantes y el proceso de imagen con respecto a tonalidades no fue evolucionado tan rápido como el hardware pero de igual forma se continuó el desarrollo de imágenes en monocromático, hasta el punto de evidenciar que necesitamos demostrar otro tipo de figuras con combinación de dicho colores, además la llegada de nuevos tonos a la imagen le daban más vida y más atracción visual que antes.

Pero tuvo su proceso de evolución con respecto al presente, en las figuras que están a continuación, debido que al ser tres columnas, los colores de los extremos al ser combinados dejaban un espacio no mayor a 5 mm lo cual proporcionaban imágenes sobre puestas, de otra realidad, dándonos a ver el mismo prototipo en cambio a los leds RGB pero no armados en columnas sino ya diseñados, así los cuales están referenciados en prototipo final de la figura 23 y figura 24.

4.4.1 DESVENTAJAS Y VENTAJAS DE SELECCIÓN DE LEDS

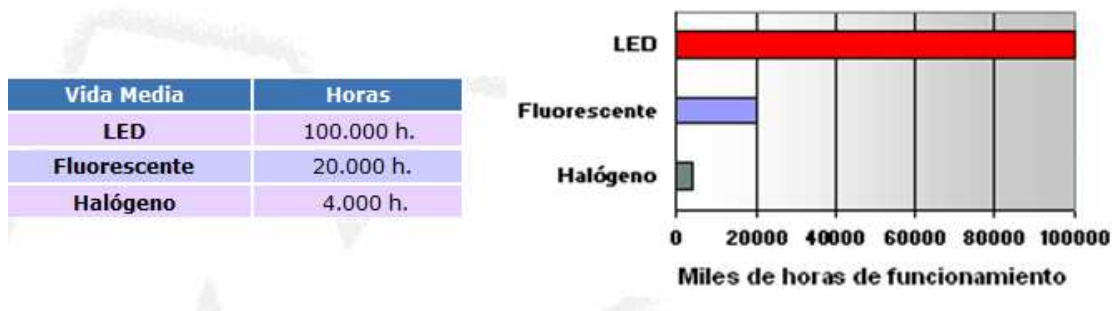
Ventajas del LED:

- **Bajo consumo:** Una lámpara LED se alimenta a baja tensión, consumiendo así poca potencia.

Ejemplo: Una lámpara halógena de 50W de potencia ilumina 25 lumens/W consiguiendo un total de 1250 lumens. Para conseguir la misma iluminación con una lámpara de LEDs vamos a necesitar 179 LEDs (utilizando LEDs de alta luminosidad que iluminan 7 lumens/unidad), de esta forma tendremos la misma iluminación con ambas lámparas pero sin embargo nuestro consumo con la lámpara de LEDs va ser 4 veces menor, ya que sólo consumirá 13W. (*Datos aproximados).

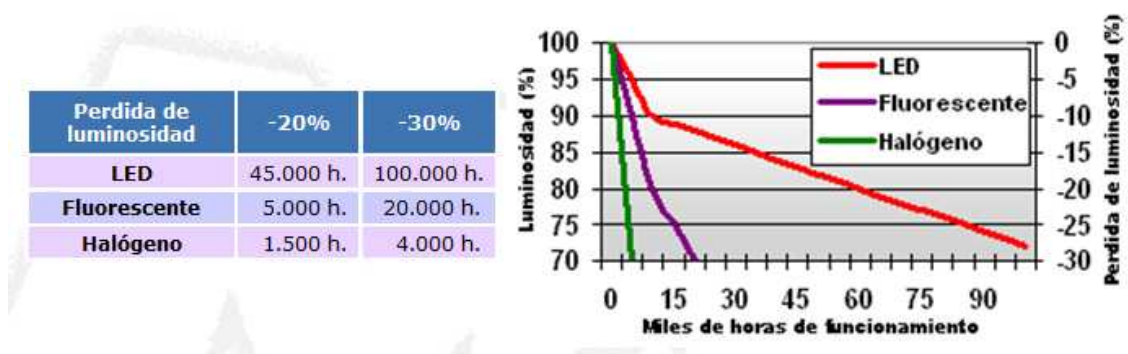
- **Baja tensión:** Todos nuestros productos se alimentan a 24V de corriente continua, adaptándose perfectamente a la mayoría de las fuentes de alimentación de los equipos, y reduciendo al mínimo los posibles riesgos de electrocución.
- **Baja temperatura:** El LED se alimenta a baja tensión, consumiendo así poca energía y por lo tanto emitiendo poco calor. Esto es debido a que el LED es un dispositivo que opera a baja temperatura en relación con la luminosidad que proporciona. Los demás sistemas de iluminación en igualdad de condiciones de luminosidad que el LED emiten mucho más calor.
- **Pequeña anchura espectral:** Los LEDs tiene una anchura espectral pequeña, convirtiéndolos de esta forma en el sistema perfecto de iluminación para visión artificial, ya que de esta manera la cámara capta con mucho más detalle el objeto, pudiendo apreciar mejor las posibles imperfecciones del mismo.
- **Amplia banda espectral:** El LED es un dispositivo de longitud de onda fija pero que puede trabajar en una amplia banda del espectro. Para cubrir todo este ancho de banda existen en el mercado una gran gama de LEDs que nos permitirán iluminar con una longitud de onda específica, o lo que es lo mismo en un determinado color (rojo, verde, ámbar, blanco e incluso ultra violeta).
- **Mayor rapidez de respuesta:** El LED tiene una respuesta de funcionamiento mucho más rápida que el halógeno y el fluorescente, del orden de algunos microsegundos, ello lo hace ideal para funcionar con un estrobo (sistemas estroboscópicos), aumentando así las prestaciones de este último.
- **Luz más brillante:** En las mismas condiciones de luminosidad que sus rivales, la luz que emite el LED es mucho más nítida y brillante.
- **Sin fallos de iluminación:** Absorbe las posibles vibraciones a las que pueda estar sometido el equipo sin producir fallos ni variaciones de iluminación. Esto es debido a que el LED carece de filamento luminiscente evitando de esta manera las variaciones de luminosidad del mismo y su posible rotura.
- **Mayor duración y fiabilidad:** La vida de un LED es muy larga en comparación con los demás sistemas de iluminación:

4.4.1.1 FIGURA COMPARATIVA DE LED_1



Y su fiabilidad es mucho mayor, ya que la degradación de la luz es mínima en relación a la de halógenos y fluorescentes:

4.4.1.2 FIGURA COMPARATIVA DE LED_2



*Datos obtenidos de las mejores marcas de iluminación del mercado.

Desventajas del LED:

- **Precio:** El mayor inconveniente que tiene el LED sin duda es su precio, pero si evaluamos sus múltiples e inmejorables condiciones de funcionamiento, y sobre todo su larga vida en comparación con los demás sistemas de iluminación, estamos en condiciones de afirmar que es la inversión más sensata, eficaz y rentable que podemos hacer.

4.5 SELECCIÓN DE MICRO

Se observo los de gama media; la cual hay varios microcontroladores. En esta gama se han presentado varias evoluciones, desde cuando salió al mercado el PIC16C84 que posteriormente fue remplazado por el PIC16F84 (la F significa que tiene memoria flash), y este por el PIC16F84A, hasta llegar al PIC16F877a⁶ que es un microcontrolador muy poderoso.

En esta gama también pertenecen los microcontroladores PIC16F873A, PIC16F874A PIC16F876A⁷, entre otros. En la tabla 10 se muestra algunas de las características de estos.

**Tabla 3. Comparación de algunos Microcontroladores Gama Media.
(Microchip, 2006).**

Características	PIC16F84A	PIC16F628A	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Memoria de Programa (ROM) FLASH	1K X 14	2K X 14	4K X 14	4K X 14	8K X 14	8K X 14
Memoria RAM	68 X 8	224 X 8	192 X 8	192 X 8	368 X 8	368 X 8
Numero de Instrucciones (RISC)	35	35	35	35	35	35
Memoria EEPROM Data	64 X 8	128 X 8	128 X 8	128 X 8	256 X 8	256 X 8
USART	NO	SI	SI	SI	SI	SI
Oscilador Interno	NO	SI	NO	NO	NO	NO
Pines de I/O	13	15	22	33	22	33
Tipo de Empaque	DIP	DIP	DIP	PDIP	DIP	PDIP
Numero de Pines	18	18	28	40	28	40

A simple vista se ven las ventajas del PIC16F877a, frente a los otros, pues es un microcontrolador con 33 pines de entrada y salida a comparación del **PIC16F874A** y el **PIC16F876A** que también cuenta con 40 pines, es por ello que para desarrollar las aplicaciones se toma este micro controlador, el lector podrá comparar y se dará cuenta que estas podrán ser usadas los otros, con

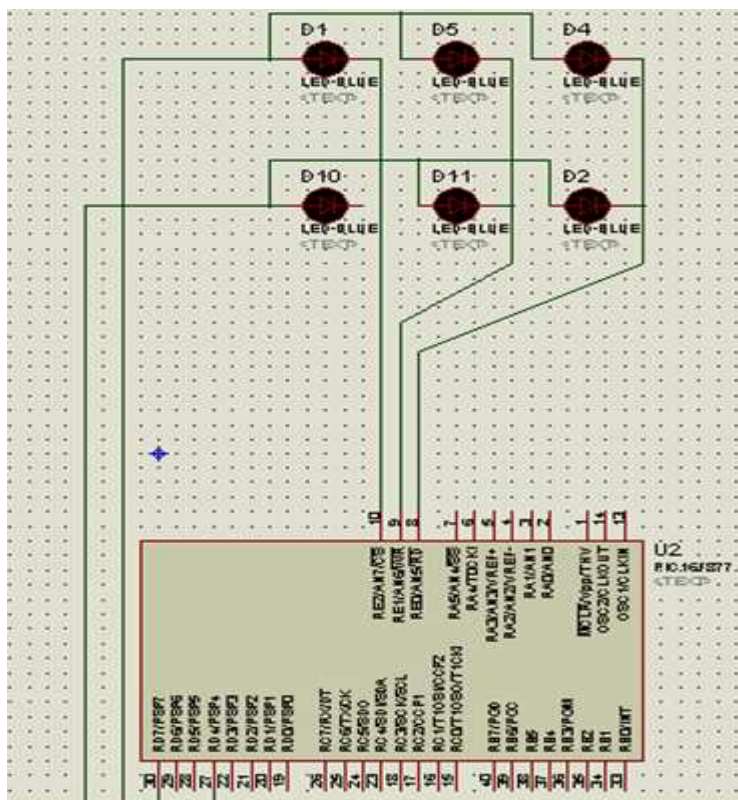
⁶ Microchip [Internet] [consultado 26 Agosto de 2009]. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>

⁷ Microchip [Internet] [consultado 26 Agosto de 2009]. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

pocas modificaciones y en algunos sin ninguna. Pero por la necesidad de leds se prefieren I/O.

Debido a la elaboración de los prototipos anteriores se evidencio que las imágenes se pueden diseñar con 24 leds no más y el microcontrolador que se adapta mejor a esto es el microcontrolador pic16f877a pero no de superficie, debido a que se administra a cada puerto 8 leds respectivamente y se configuran en este orden ascendente, puerto B, C y D, pero desde luego el microcontrolador posee más puertos los cuales cumplen la siguiente la función , el puerto E es el selector de tonalidad de nuestro dispositivo, debido a que maneja todas las tierras , además dicho puerto se constituye de 3 pines y por último el puerto A el cual se uso como anexo al proyecto debido a que se le instalo un dispositivo RF y en dicho puerto se realizan las interrupciones para manipular el sistema a distancia.

Figura 18.2. Diagrama de conexión de leds al microcontrolador.



(Figura 18.2 distribución de pines del micro en una conexión de 2 leds RGB en dos filas, debido a la cantidad de leds no se conectaron los 26 leds faltantes).

4.6 SELECCIÓN DE BATERIAS

Al alimentar el dispositivo de voltaje del prototipo final se tuvo en cuenta cual era la mejor opción para nuestro requerimiento de peso y recarga, al indagar sobre las pilas recargables AA son 10 veces mejores que las de litio en todos los aspectos, pero la única desventaja, es el tamaño de las AA.

La cantidad de energía que produce una batería se mide en miliamperios. Tenemos por un lado nuestro prototipo que usa una batería recargable de litio, o litio-ion, que genera como máximo 700-900 mA. Tenemos por el otro lado, el prototipo que usa 2 pilas AA a la cual le hemos puesto dos pilas AA de 2700 mA, tendríamos entonces un total de 5400 mA.

Matemáticamente hablando tenemos que nuestro prototipo sería capaz funcionar casi 8 veces más tiempo que las de litio. La principal ventaja de las AA se da que si te quedas sin energía puedes comprar pilas nuevas en casi cualquier lugar del mundo.

Pero optamos por emplear las de Litio-Ión (ver grafica 24) las cuales por su tamaño permiten hacer el prototipo final más pequeño y practico de manipular.

4.7 AVANCE PROTOTIPO CON MODULO RF

Los módulos de radiofrecuencia de 434Mhz (TLP y RLP 434) para comunicar dos circuitos, en este caso un transmisor TX y un receptor RX.

Los módulos empleados solo nos permiten realizar una comunicación simplex, existes en el mercado mejores módulos que cuentan con modulación FSK (más inmune al ruido) y full dúplex.

En el transmisor esta una botonera que al momento de ser presionada hace que el micro en el transmisor envíe un dato cuyo valor será validado en el receptor, esta comunicación se realiza gracias a los módulos de radio frecuencia mencionada. A continuación mostramos los datos de conexionado y polarización de los módulos empleados y el código utilizado para que el proyecto funcione.

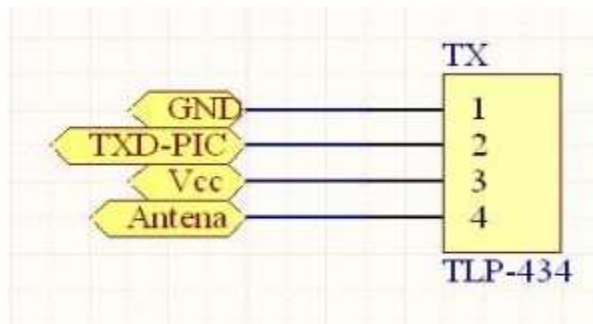
19. DIAGRAMA DE CONECCIÓN DE LOS MODULOS TLP-434 Y RLP-434

Estos módulos de radio frecuencia nos permiten hacer aplicaciones inalámbricas de radio control. El rango de Vcc en el TLP-434 está entre 2 – 12 V. El TXD-PIC va conectado al pin TX del pic, para usar la comunicación UART. La antena debe medir un cuarto de la longitud de onda (L/4)

$$L=c/f=300.000.000m/434.000.000Hz=0.69m$$

$$\text{Long Ant} = L/4 = 17\text{cm}$$

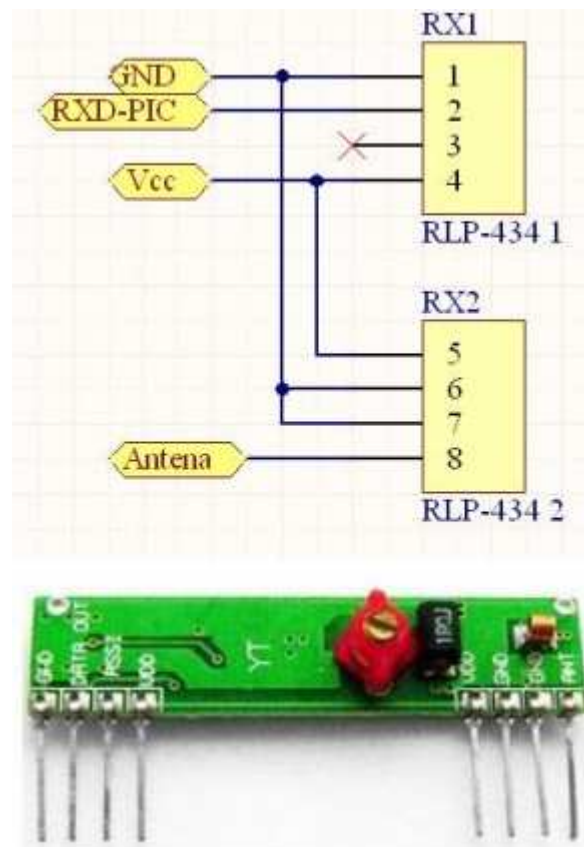
19.1 PINES DEL Tlp 434



El rango de Vcc en el RLP-434 está entre 4.5 – 5.5 V. El TXD-PIC va conectado al pin RX del pic, para usar la comunicación UART.

Long Ant = $L/4 = 17\text{cm}$

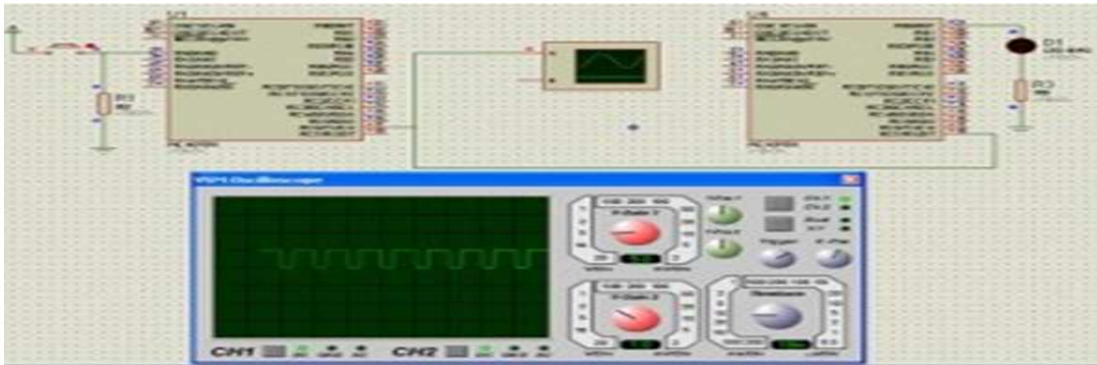
19.2 TXMODULO RF



19.3 SIMULACION DE LA COMUNICACIÓN ENTRE LOS MICROS

En la simulación basta con unir los micros (Pics) con un cable representando así la comunicación seria entre el PIC que transmite y el PIC que recibe

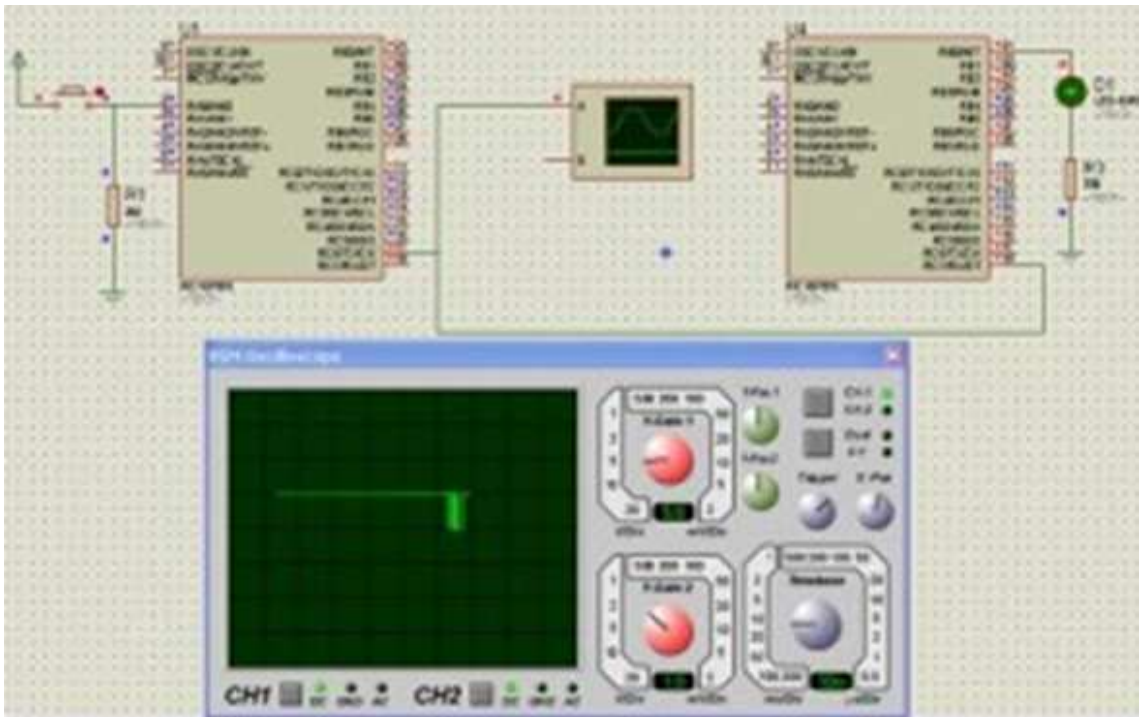
19.3 SIMULCIÓN



En esta figura vemos como se transmite el dato cero (0X00) constantemente mientras no se presiona la botonera en el transmisor.

El receptor al recibir el dato 0x00 (hexadecimal) no enciende el Led indicador

19.4 SIMULACION



En esta figura apreciamos que al momento de presionar la botonera, el transmisor envía el dato 23 (decimal), este dato lo recibe el receptor y lo valida encendiendo entonces el Led indicador como se ve en la figura.

19.5 CODIGO EN PICC DEL MICRO TX

```
#include <16F876A.h>

#FUSES NOWDT //No Watch Dog Timer

#FUSES HS //High speed Osc (> 4 MHz)

#FUSES NOPUT //No Power Up Timer

#FUSES PROTECT //Code not protected from reading

#FUSES NODEBUG //No Debug mode for ICD

#FUSES BROWNOUT //Reset when brownout detected

#FUSES NOLVP //Low Voltage Programming on B3 (PIC16) or B5 (PIC18)

#FUSES NOCPD //No EE protection

#FUSES NOWRT //Program memory not write protected

#use delay (clock=1000000)

#use rs232(baud=2400,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

void main() {

set_tris_a (0b00000001);

set_tris_b (0b00000000);

set_tris_c (0b00000000);

setup_adc_ports(NO_ANALOGS);

setup_spi(FALSE);

setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);

setup_timer_1(T1_DISABLED);

setup_timer_2 (T2_DIV_BY_1, 0, 1);

setup_comparator (NC_NC_NC_NC);

setup_vref (FALSE);

While (1) {

If (input (PIN_A0) ==1) {
```

```

While (input (PIN_A0) ==1) {
}

Putc (23);

}else{

putc(0);

}

}

}

```

19.6 CODIGO EN PICC DEL MICRO RX

```

#include <16F876A.h>

#FUSES NOWDT //No Watch Dog Timer

#FUSES HS //High speed Osc (> 4 MHz)

#FUSES NOPUT //No Power Up Timer

#FUSES PROTECT //Code not protected from reading

#FUSES NODEBUG //No Debug mode for ICD

#FUSES BROWNOUT //Reset when brownout detected

#FUSES NOLVP //Low Voltage Programming on B3 (PIC16) or B5 (PIC18)

#FUSES NOCPD //No EE protection

#FUSES NOWRT //Program memory not write protected

#use delay (clock=1000000)

#use rs232 (baud=2400, parity=N, xmit=PIN_C6, rcv=PIN_C7, bits=8)

int it;

Void main () {

set_tris_a (0x00);

set_tris_b (0b00000000);

```

```

set_tris_c (0b10000000);

setup_adc_ports (NO_ANALOGS);

setup_spi (FALSE);

setup_timer_0 (RTCC_INTERNAL|RTCC_DIV_1);

setup_timer_1 (T1_DISABLED);

setup_timer_2 (T2_DIV_BY_1, 255, 1);

setup_comparator (NC_NC_NC_NC);

setup_vref (FALSE);

while(1){

delay_ms(5);

if(getc()=='23'){

if(it==0){ it=1;

output_low(pin_B0);

}

else{ it=0;

output_high(pin_B0);

}

}

}

}

```

Además se integró módulo RF para manipular el cambio de imagen de forma inalámbrica, con el cual se diseñó un tablero con un teclado de 4x3 y una LCD para manipular todo desde ese tablero.

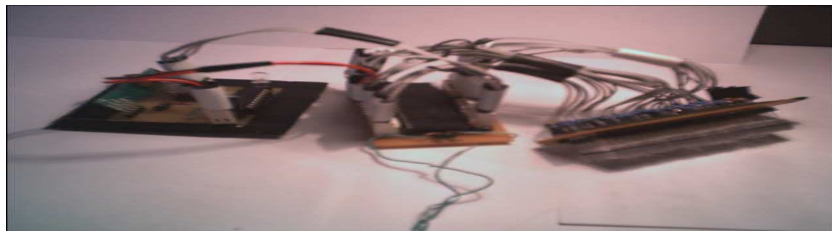
Imágenes de partes del prototipo:

Figura 19.1 Tablero con Tx de modulo RF y LCD.



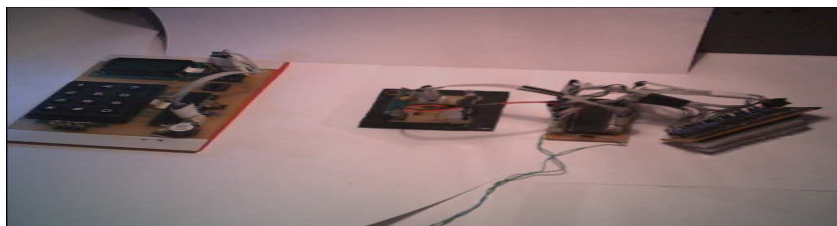
- El tablero tiene un microcontrolador pic16f877a que controla la LCD y datos de entrada y de envío del mensajes por medio del transmisor del módulo RF, al dispositivo de leds para cambiar la imagen deseada.

Figura 20. Circuito con Rx de modulo RF y columna de leds.



- En esta imagen tenemos el receptor del modulo, otro microcontrolador RF y los leds planos a chorro rojos, verdes y azules cada uno con su respectiva baquela.
- La alimentación de voltaje del sistema es independiente una para los leds con el micro y otra para el motor de rotación del sistema.

Figura 21. Tablero con Tx de modulo RF y LCD. "prototipo 3 "



- La alimentación del tablero se hizo a través de un adaptador a 12 voltios y se reguló la entrada, y el del motor de rotación del sistema de leds con un adaptador de 24 voltios en la figura 11 evidenciamos una imagen con el dispositivo.

Figura 22. Visualización prototipo funcional 3.



La imagen vista en el dispositivo fue la del mejor equipo de futbol del mundo (club deportivo los millonarios), se desarrollaron las anteriores imágenes vistas en los prototipos pasados y se manipularon varios colores y combinación de ellos.

4.8 DISEÑO DEL PROGRAMA EN ASSEMBLER DEL PROTOTIPO FINAL

En esta fase del desarrollo del software final, la cual tome una fracción del anexo D, debido a que es muy extenso pero a la vez repetitivo, explicare la funcionalidad de este subprograma llamado colombia_1.

Aunque se desarrollan de la misma forma anteriormente explicada como la recreación de letras, hay pequeñas diferencias que son ahora por cada columna, se realiza una selección de tiempo adicional con una subrutina de tiempo más, pero de igual valor en todas las columnas de imágenes a recrear, lo cual garantiza que la imagen sea estática, debido a que este tiempo extra proporciona permanecía en la ejecución del programa siendo más visible la ilustración de dicho color.

Como se muestra en el programa que esta a continuación.

```
#INCLUDE<p16f877A.inc>
```

```
ORG 0X0000;
```

```
CBLOCK 0X20  
MONO1  
MONO2  
MONO3  
MONO4
```

CAMBIO
CAMBIO_1
CONTA3
CONTA1
CONTA2
PERRO
GATO
LEON
BBB
ENDC

BSF STATUS, 5 ; Se escoge el Banco 1 para programar
Puertos.
BCF STATUS, 6

CLRF TRISB; SALIDA
CLRF TRISC; SALIDA C
CLRF TRISD; SALIDA D
CLRF TRISE; SALIDA E SON TIERRAS

BCF STATUS, 5 ; Se escoge el Banco 0 para hacer la señal
seno
BCF STATUS, 6

CLRF PORTB
CLRF PORTC
CLRF PORTD

PRINCIPAL

.....
.....PUNTO_1
.....

PUNTO_1

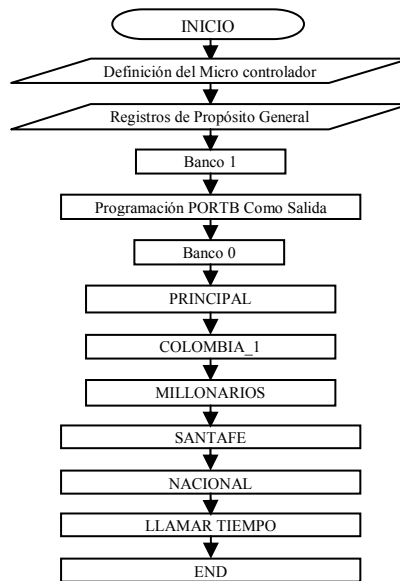
MOVLW 0X05; BANDERA COLOMBIA
MOVWF PERRO

DESARROLLO_1

CALL COLOMBIA_1
DECFSZ PERRO, 1
GOTO DESARROLLO_1

.....
.....
.....

Diagrama 4. Flujo grama prototipo 4.



Se recomienda al lector ver el video llamado video final en el cual se observa las imágenes del programa final, en anexos I se encuentra todo el software y las librerías en anexo J y k.

5. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

Con los cuatro prototipos se evidenció procesos que tenían que ser mejorados y los decidimos cambiar en el último prototipo

1. Se Diseñó una baquela la cual soportará los leds, el módulo RF y el microcontrolador, dicha baquela se diseñó a doble cara en vez de ser montada en Báquelas universales o protoboard. Figura 26
2. Se Cambió de nuevo la materia prima del proyecto por otro tipo de leds llamados RGB debido a que podemos manipular más matices de color y es menos pesado que tener tres columnas de leds formando las tres tonalidades.

Figura 23. Leds RGB no difusos.



Fuente:

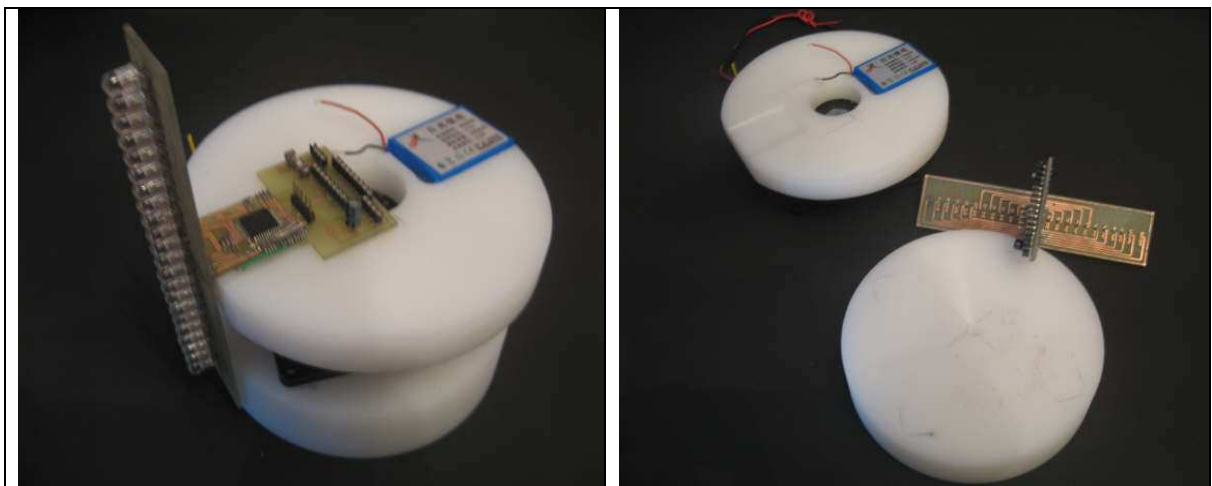
www.decelectronics.com/html/OPTOELECTRONICA/OPTOELECTRONICA.com

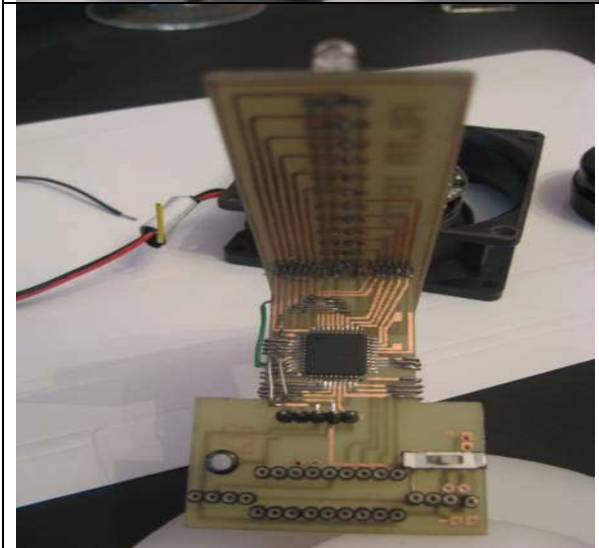
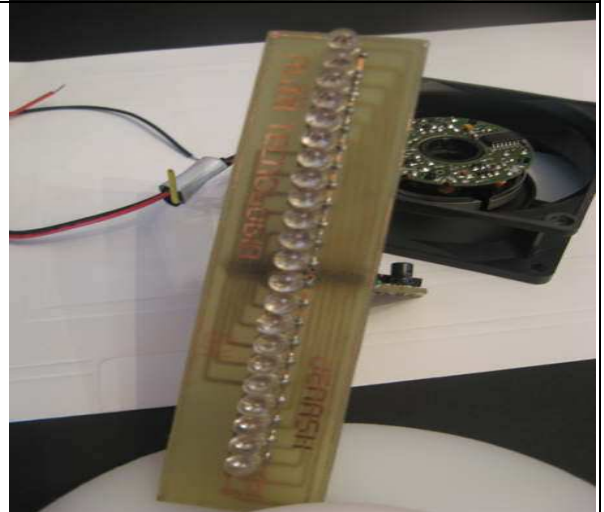
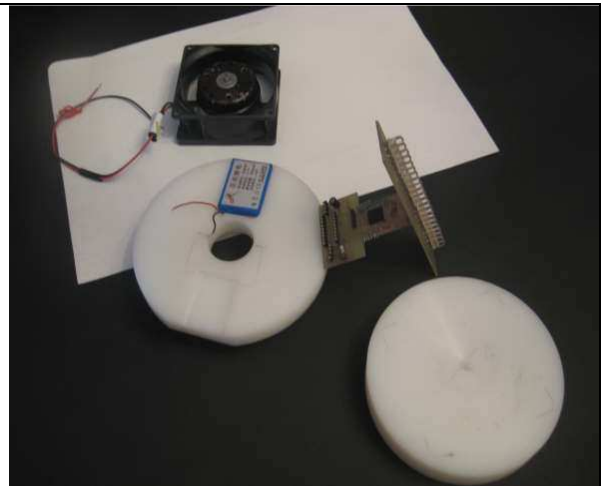
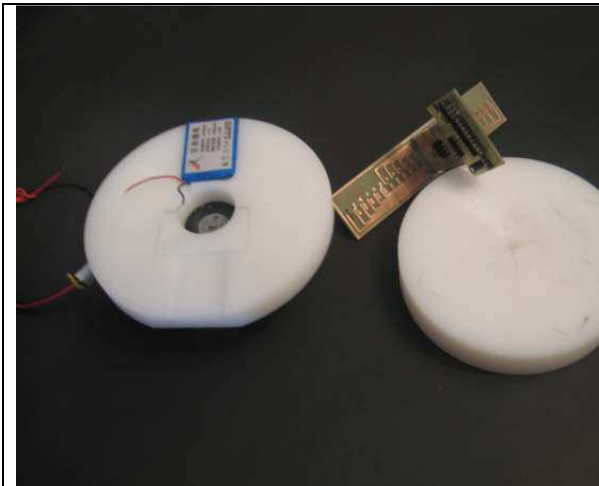
3. Se utilizó una batería de litios para alimentar de voltaje el micro , los leds y el receptor del modulo RF y se dejó a un lado por el peso las pilas doble A
4. Se diseñó la base de todos los dispositivos donde se sostienen las Báquelas con Empac, el cual es un material muy resistente y fácil de utilizar en la fresadora para adquirir cualquier forma deseada.
5. Se utilizó una alimentación de voltaje para el movimiento de todo el Empac con los dispositivos con un motor de 24 voltios.

6. Para equilibrar pesos en la base del dispositivo de rotación adherimos una base de mármol
7. En desarrollo del programa se hizo totalmente en lenguaje assembler y manipulando todos los pines del microcontrolador, la distribución fue la siguiente debido a que el microcontrolador consta de 5 puertos pero no todos con la misma cantidad de pines se realizó la siguiente distribución: el puerto A, para manipulación del módulo de recepción de datos, el puerto B que consta de 8 pines para manipulación de los primeros 8 leds de la columna de leds RGB, luego el puerto C para los siguientes 8 leds RGB y el puerto D de la misma forma, debido a que los puertos B, C y D contienen 8 pines cada uno y manipulando así los 28 leds de la columna, luego uno de los puertos más importante es el E debido a que con él se manipulo todas las tonalidades de cualquier figura que desee realizar.

Para visualizar mejor lo anteriormente explicado, el prototipo en las siguientes imagines de la figura 24

Figura 24. Prototipo final.

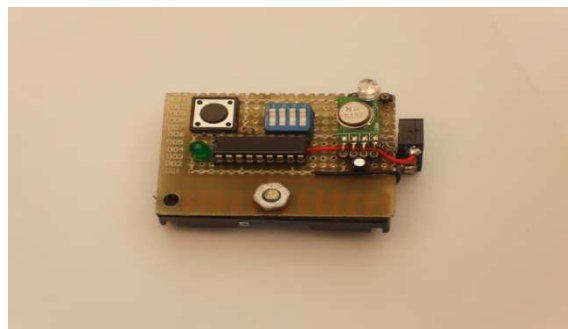






8. Por último se diseñó el transmisor del módulo para cambiar las imágenes en el dispositivo el cual se diseñó en una baquela universal y puede manipular mas 128 interrupciones en el programa principal y consta de una alimentación de dos pilas doble A y un dip- swich para el cambio de datos en código binario para elegir el cambio deseado. Ver figura 25

Figura 25. Modulo de Tx del prototipo final.



CONCLUSIONES

El tipo de giro más adecuado es en el eje x, y contó con la observación de características como el tipo de visualización, debido a que se puede ver la imagen en su rotación de 360 grados, y no de una forma unipersonal, otro punto a resaltar es la imagen debido a que no se recorta en ningún punto, formando un pantalla perfecta sin recortes, el único inconveniente es por su diseño debido a que en este eje x, el tipo de construcción de la columna de leds tiene que ser en su respectivo centro de gravedad o proporcionaría deficiencia en la construcción de la imagen.

El tipo de leds más adecuado, depende de la finalidad de la imagen a requerir, en el dispositivo se usó los leds RGB, aunque se optó en un principio por diseñar el Led RGB con los tres leds unidos (rojo, verde y azul) pero generaba en la visualización de la imagen espacios e imágenes borrosas, además el Led RGB nos permite no solo ver colores principales sino combinación de ellos como el amarillo, el magenta y el cian.

A mayor velocidad mejor eficiencia de imagen en el dispositivo, debido a que en el prototipo final se optó por acelerar el programa y se perdió realidad en la formación de imágenes. Además entre menor sea el tamaño del Led y difuso, mejora el detalles de la imagen y matices del mismo.

El hardware más adecuado se genera con la implementación de motores dc debido a la poca utilización de corriente y el diseño de la base del montaje del dispositivo se diseña mejor en acrílico que en Empac debido a la durabilidad del material.

El programa adecuado para el sistema de control de encendido de los leds RGB depende de la velocidad de giro del motor. En el software se evidencia las diferentes manipulaciones de los tiempos de cada figura no siendo un solo tiempo, general para todos, siendo así el desarrollo de cada imagen un trabajo tedioso.

La implementación de la caja de protección es esencial, debido a que la velocidad de giro podría causar lesiones en los miembros del cuerpo y además es de protección del mismo sistema debido a que cualquier golpe podría dañar los caminos de conexión del mismo y no permitir el perfecto funcionamiento, cortando la producción de la imagen.

RECOMENDACIONES

Hay que tener en cuenta la frecuencia en la cual funciona el micro controlador y el tiempo de ejecución del software para reproducir mensajes por el prototipo, debido a que las imágenes pueden verse de diferente tamaño.

La cantidad de leds a utilizar y el tipo de leds que se empleen depende el tipo de imagen a diseñar y por consiguiente cambio de frecuencias en la manipulación del programa.

Utilizar solo una fuente y no manipular varias para dar mayor confort en la manipulación del sistema, siendo un beneficio al consumo de corriente del sistema.

Utilizar escobillas en la manipulación del giro de los dispositivos conectados al motor de rotación, para dejar a un lado el peso de las baterías extras en el dispositivo.

Manipular micro controladores de superficie para mayor eficiencia de corriente y espacio del dispositivo.

Cambiar el sistema de modulo Rf por bluetooth, para estar un poco más a la vanguardia de la tecnología.

Buscar siempre eficiencia en mayor velocidad de giro, que estabilidad debido a que con mayor velocidad, se evidencia mejor las imágenes en el sistema.

Buscar siempre el micro controlador que este a la vanguardia no monopolizar al sistema a un solo micro.

Minimizar al máximo el programa del dispositivo debido que en assembler se desarrollan más líneas que otros lenguajes es mejor optar por lenguaje c.

BIBLIOGRAFÍA

- "MICROCONTROLADOR PIC16F84. Desarrollo de proyectos" de la Editorial Ra-Ma
- José M^a Angulo Usategui, Ignacio Angulo Martínez - Microcontroladores PIC
- libro práctico de los generadores, transformadores y motores eléctricos Escrito por Gilberto Harper Enriquez
- Electronica_ucontrol_revista_0001
- Programming the PIC Microcontroller with MBASIC
- <http://www.ladyada.net/make/spokepov/>
- http://drewish.com/blogger/archives/2005/03/17/2sided_pov_toy.html
- <http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2006/ja94/Amsel%20-%20Klitinek%20Final%20Project/index.htm>
- <http://hackedgadgets.com/2007/09/11/globe-led-pov/>
- http://www.pyrofersprojects.com/blog/?page_id=107
- <http://hacknmod.com/hack/dirt-simple-pov-led-display/>
- <http://www.instructables.com/id/SpokePOV:-LED-Bike-Wheel-Images/>
- <http://www.bricogeek.com/index/cat/7/nid/967/>
- <http://hackaday.com/2008/10/17/led-pov-hard-drive-clock/>
- Microchip [Internet] [consultado 26 Agosto de 2009]. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/40044F.pdf>
- Microchip [Internet] [consultado 26 Agosto de 2009]. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

GLOSARIO

BLUETOOTH: es un protocolo inalámbrico abierto para el intercambio de datos en distancias cortas (mediante ondas de radio de corta duración) de los dispositivos fijos y móviles, la creación de redes de área personal (PAN).

DIODO EMISOR DE LUZ: también conocido como LED (acrónimo del inglés de Light-Emitting Diode) es un dispositivo semiconductor (diodo) que emite luz coherente de espectro reducido cuando se polariza de forma directa la unión PN del mismo y circula por él una corriente eléctrica. Este fenómeno es una forma de electroluminiscencia. El color (longitud de onda), depende del material semiconductor empleado en la construcción del diodo y puede variar desde el ultravioleta, pasando por el visible, hasta el infrarrojo. Los diodos emisores de luz que emiten luz ultravioleta también reciben el nombre de UV LED (UltraViolet Light-Emitting Diode) y los que emiten luz infrarroja suelen recibir la denominación de IRED (Infra-Red Emitting Diode).

EL MOTOR DE CORRIENTE CONTINUA: es una máquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio. En la actualidad existen nuevas aplicaciones con motores eléctricos que no producen movimiento rotatorio, sino que con algunas modificaciones, ejercen tracción sobre un riel. Estos motores se conocen como motores lineales. Esta máquina de corriente continua es una de las más versátiles en la industria. Su fácil control de posición, par y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Pero con la llegada de la electrónica su uso ha disminuido en gran medida, pues los motores de corriente alterna, del tipo asíncrono, pueden ser controlados de igual forma a precios más accesibles para el consumidor medio de la industria. A pesar de esto los motores de corriente continua se siguen utilizando en muchas aplicaciones de potencia (trenes y tranvías) o de precisión (máquinas, micros motores, etc.).

FRECUENCIA: es una medida para indicar el número de repeticiones de cualquier fenómeno o suceso periódico en la unidad de tiempo. Para calcular la frecuencia de un evento, se contabilizan un número de ocurrencias de este teniendo en cuenta un intervalo temporal, luego estas repeticiones se dividen por el tiempo transcurrido.

LUZ INFRARROJO: La luz infrarroja es una forma de ondas electromagnéticas. La luz puede ser visible o infrarroja puede ser en forma de microondas. En su forma visible, el infrarrojo se puede ver el rojo o el violeta.

UN MICRO CONTROLADOR: es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

MODULACION ASK: es una forma de modulación que representa los datos digitales como variaciones en la amplitud de una onda portadora.

RADIO FRECUENCIA: es un subconjunto de la radiación electromagnética de longitud de onda de 100 km a 1 mm, que es una frecuencia de 3 kHz a 300 GHz, respectivamente.

RGB: es un modelo de color aditivo en la que verde rojo, y la luz azul se suman varias formas de reproducir una amplia gama de colores. El nombre del modelo proviene de las iniciales de los tres colores primarios aditivos, rojo, verde y azul.

ANEXOS A

MÓDULO RF

TLP434A - RF ASK Hybrid Modules for Radio Control



TLP 434A - Ultra Small Transmitter

Frequency : 433.92MHz

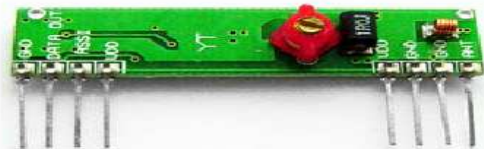
Spec:

- Frequency Range: 433.92 MHz
- Modulate Mode: ASK
- Circuit Shape: S/F
- Data Rate: 8Kbps
- Supply Voltage: 3~ 12 V
- Voltage:3V; Current:4.9mA
- Voltage:5V; Current:8.4mA
- Power Supply and All Input / Output Pins: -0.3 to 12.0 V
- Non-Operating Case Temperature: -10 to +85 centigrade
- Soldering Temperature: 230 centigrade (10 Seconds)
- Simple To Apply with Low External Count
- High Sensitivity Passive Design.

Applications:

- Wireless security systems
- Car alarms
- Remote gate controls
- Remote sensing
- Date communication
- Remote water/electric switch
- Pager system
- PDA data communicated

RLP434 - RF ASK Hybrid Modules for Radio Control



RLP 434 - SAW Based Receiver

Frequency : 433.92MHz

Spec:

- Frequency Range:433.92MHz
- Modulate Mode: ASK
- Circuit Shape: LC
- Date Rate: 4800bps
- Sensitivity: -106 dBm
- Channel Spacing: +/-500KHz.
- Supply Voltage: 5V
- High Sensitivity Passive Design
- Simple To Apply with Low External Count

Applications:

- Wireless security systems
- Car alarms
- Remote gate controls
- Remote sensing
- Date communication
- Remote water/electric switch
- Pager system
- PDA data communicated

DC Characteristics :

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
Vcc	Operating Supply Voltage		4.9	5	5.1	
I Tot	Operating Supply Voltage			4.5		
V Data	Data Out	1 Data = +200 uA (High)	Vcc -0.5	Vcc		V
		1 Data = -10 uA (Low)			0.3	V

Electrical Characteristics:

Parameter	Sym.	Min.	Typ.	Max.	Unit
Operating Radio Frequency	FC		433.92		MHz
Sensitivity	Pref.		-108		dBm
Channel Width		-500		+500	KHz
Noise Equivalent BW	NEB		5	4	KHz
Baseboard Data Rate			3	KB/S	
Receiver Turn On Time				3	ms

ANEXO B

MICROCONTROLADOR PIC 16F877

Se denomina microcontrolador a un dispositivo programable capaz de realizar diferentes actividades que requieran del procesamiento de datos digitales y del control y comunicación digital de diferentes dispositivos.

Los Microcontroladores poseen una memoria interna que almacena dos tipos de datos; las instrucciones, que corresponden al programa que se ejecuta, y los registros, es decir, los datos que el usuario maneja, así como registros especiales para el control de las diferentes funciones del microcontrolador.

Los Microcontroladores se programan en Assembler y cada microcontrolador varía su conjunto de instrucciones de acuerdo a su fabricante y modelo. De acuerdo al número de instrucciones que el microcontrolador maneja se le denomina de arquitectura RISC (reducido) o CISC (complejo).

Los Microcontroladores poseen principalmente una ALU (Unidad Lógica Aritmética), memoria del programa, memoria de registros, y pines I/O (entrada y/o salida). La ALU es la encargada de procesar los datos dependiendo de las instrucciones que se ejecuten (ADD, OR, AND), mientras que los pines son los que se encargan de comunicar al microcontrolador con el medio externo; la función de los pines puede ser de transmisión de datos, alimentación de corriente para el funcionamiento de este o pines de control específico.

En este proyecto se utilizó el PIC 16F877. Este microcontrolador es fabricado por MicroChip familia a la cual se le denomina PIC. El modelo 16F877 posee varias características que hacen a este microcontrolador un dispositivo muy versátil, eficiente y práctico para ser empleado en la aplicación que posteriormente será detallada.

Algunas de estas características se muestran a continuación:

- Soporta modo de comunicación serial, posee dos pines para ello.
- Amplia memoria para datos y programa.
- Memoria reprogramable: La memoria en este PIC es la que se denomina FLASH; este tipo de memoria se puede borrar electrónicamente (esto corresponde a la "F" en el modelo).
- Set de instrucciones reducido (tipo RISC), pero con las instrucciones necesarias para facilitar su manejo.

CARACTERÍSTICAS

En siguiente tabla de pueden observar las características más relevantes del dispositivo:

CARACTERÍSTICAS	16F877
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales Pwm	2
Pila Harware	-
Ejecución En 1 Ciclo Máquina	-

Descripción de los puertos:

Puerto A:

- Puerto de e/s de 6 pines
- RA0 è RA0 y AN0

- RA1 è RA1 y AN1
- RA2 è RA2, AN2 y Vref-
- RA3 è RA3, AN3 y Vref+
- RA4 è RA4 (Salida en colector abierto) y T0CKI(Entrada de reloj del modulo Timer0)
- RA5 è RA5, AN4 y SS (Selección esclavo para el puerto serie síncrono)

Puerto B:

- Puerto e/s 8 pines
- Resistencias pull-up programables
- RB0 è Interrupción externa
- RB4-7 Interrupción por cambio de flanco
- RB5-RB7 y RB3 è programación y debugger in circuit

Puerto C:

- Puerto e/s de 8 pines
- RC0 è RC0, T1OSO (Timer1 salida oscilador) y T1CKI (Entrada de reloj del modulo Timer1).
- RC1-RC2 è PWM/COMP/CAPT
- RC1 è T1OSI (entrada Osc timer1)
- RC3-4 è IIC
- RC3-5 è SPI
- RC6-7 è USART

Puerto D:

- Puerto e/s de 8 pines
- Bus de datos en PPS (Puerto paralelo esclavo)
- Puerto E:
- Puerto de e/s de 3 pines
- RE0 è RE0 y AN5 y Read de PPS
- RE1 è RE1 y AN6 y Write de PPS
- RE2 è RE2 y AN7 y CS de PPS

Dispositivos periféricos:

- Timer0: Temporizador-contador de 8 bits con preescaler de 8 bits
- Timer1: Temporizador-contador de 16 bits con preescaler que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con preescaler y postescaler.
- Dos módulos de Captura, Comparación, PWM (Modulación de Anchura de Impulsos).
- Conversor A/D de 10 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I2C (Master/Slave).
- USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter) con 9 bit.
- Puerta Paralela Esclava (PSP) solo en encapsulados con 40 pines

DIAGRAMA DE BLOQUES

Para ver el gráfico seleccione la opción "Descargar" del menú superior

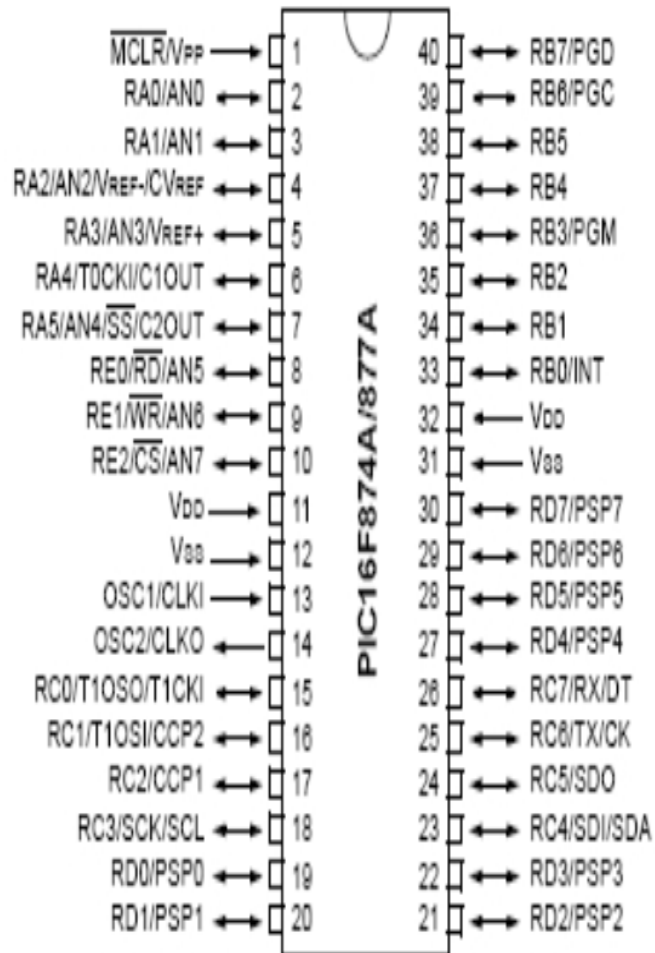
DESCRIPCIÓN DE PINES

Para ver el gráfico seleccione la opción "Descargar" del menú superior

NOMBRE DEL PIN	PIN	TIPO	TIPO DE BUFFER	DESCRIPCIÓN
OSC1/CLKIN	13	I	ST/MOS	Entrada del oscilador de cristal / Entrada de señal de reloj externa
OSC2/CLKOUT	14	O	-	Salida del oscilador de cristal
MCLR/Vpp/THV	1	I/P	ST	Entrada del Master clear (Reset) o entrada de voltaje de programación o modo de control high voltaje test
RA0/AN0				PORTA es un puerto I/O bidireccional
RA1/AN1	2	I/O	TTL	RA0: puede ser salida analógica 0
RA2/AN2/ Vref-	3	I/O	TTL	RA1: puede ser salida analógica 1
RA3/AN3/Vref+	4	I/O	TTL	RA2: puede ser salida analógica 2 o referencia negativa de voltaje
RA4/T0CKI	5	I/O	TTL	RA3: puede ser salida analógica 3 o referencia positiva de voltaje
RA5/SS/AN4	6	I/O	ST	RA4: puede ser entrada de reloj el timer0.
	7	I/O	TTL	RA5: puede ser salida analógica 4 o el esclavo seleccionado por el puerto serial síncrono.
RBO/INT				PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas
RB1				
RB2	33	I/O	TTL/ST	RB0 puede ser pin de interrupción externo.
RB3/PGM	34	I/O	TTL	RB3: puede ser la entrada de programación de bajo voltaje
RB4	35	I/O	TTL	

RB5	36	I/O	TTL	Pin de interrupción
RB6/PGC	37	I/O	TTL	Pin de interrupción
RB7/PGD	38	I/O	TTL	Pin de interrupción. Reloj de programación serial
	39	I/O	TTL/ST	
	40	I/O	TTL/ST	
RC0/T1OSO/T1CKI	15	I/O	ST	PORTC es un puerto I/O bidireccional
RC1/T1OS1/CCP2	16	I/O	ST	RCO puede ser la salida del oscilador timer1 o la entrada de reloj del timer1
RC2/CCP1	17	I/O	ST	
RC3/SCK/SCL				RC1 puede ser la entrada del oscilador timer1 o salida PWM 2
RC4/SD1/SDA	18	I/O	ST	
RC5/SD0				RC2 puede ser una entrada de captura y comparación o salida PWN
RC6/Tx/CK	23	I/O	ST	RC3 puede ser la entrada o salida serial de reloj síncrono para modos SPI e I2C
RC7/RX/DT	24	I/O	ST	
	25	I/O	ST	
	26	I/O	ST	RC4 puede ser la entrada de datos SPI y modo I2C
				RC5 puede ser la salida de datos SPI
				RC6 puede ser el transmisor asíncrono USART o el reloj síncrono.
				RC7 puede ser el receptor asíncrono USART o datos síncronos

RD0/PSP0				PORTD es un puerto bidireccional paralelo
RD1/PSP1	19	I/O	ST/TTL	
RD2/PSP2	20	I/O	ST/TTL	
RD3/PSP3	21	I/O	ST/TTL	
RD4/PSP4	22	I/O	ST/TTL	
RD5/PSP5	27	I/O	ST/TTL	
RD6/PSP6	28		ST/TTL	
RD7/PSP7	29		ST/TTL	
	30		ST/TTL	
RE0/RD/AN5	8	I/O	ST/TTL	PORTE es un puerto I/O bidireccional RE0: puede ser control de lectura para el puerto esclavo paralelo o entrada analógica 5 RE1: puede ser escritura de control para el puerto paralelo esclavo o entrada analógica 6 RE2: puede ser el selector de control para el puerto paralelo esclavo o la entrada analógica 7.
RE1/WR/AN				
RE2/CS/AN7	9	I/O	ST/TTL	
	10	I/O	ST/TTL	
Vss	12.31	P	-	Referencia de tierra para los pines lógicos y de I/O
Vdd	11.32	P	-	Fuente positiva para los pines lógicos y de I/O
NC	-	-	-	No está conectado internamente



ANEXO C

DISEÑO DE CIRCUITOS

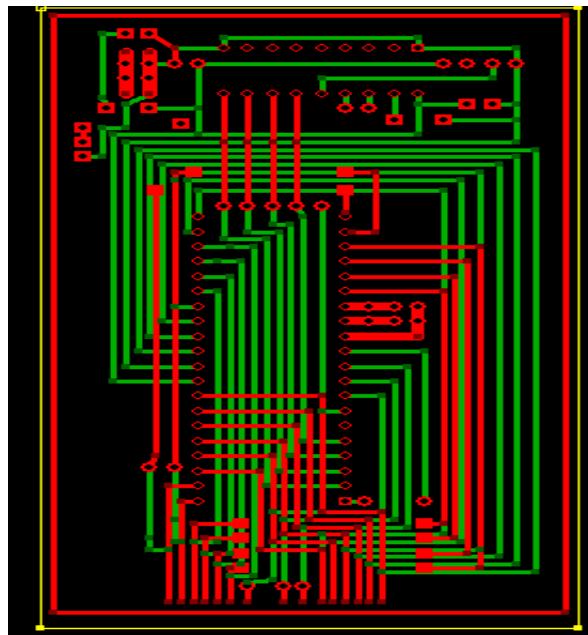
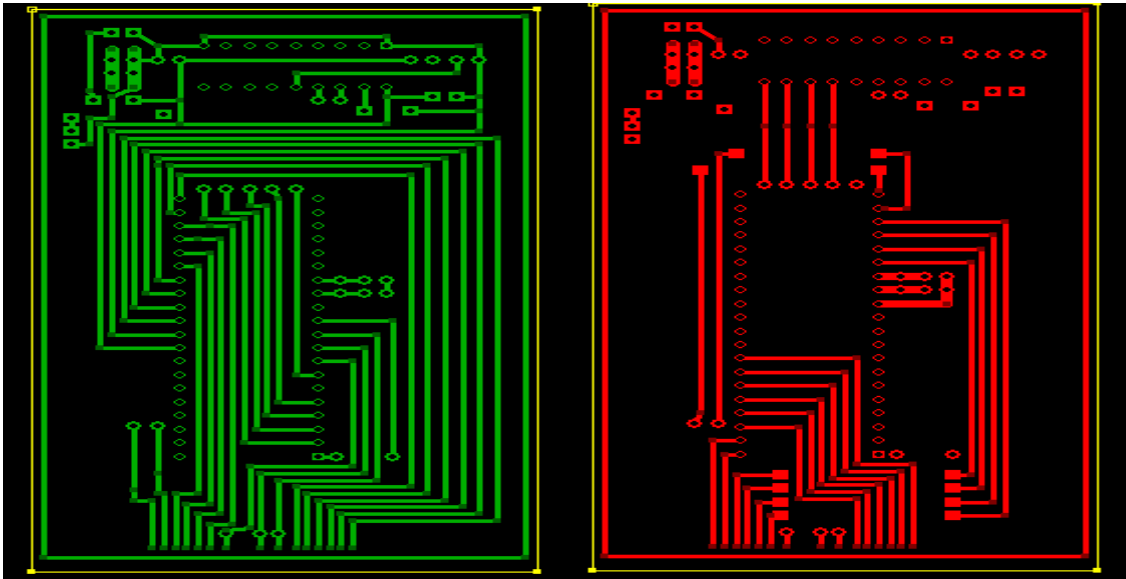


Figura 26

ANEXO D

Software

```
#INCLUDE<p16f877A.inc>
```

```
ORG 0X0000;
```

```
CBLOCK 0X20
```

```
MONO1
```

```
MONO2
```

```
MONO3
```

```
MONO4
```

```
CAMBIO
```

```
CAMBIO_1
```

```
CONTA3
```

```
CONTA1
```

```
CONTA2
```

```
PERRO
```

```
GATO
```

```
LEON
```

```
BBB
```

```
ENDC
```

```
BSF STATUS, 5 ; Se escoge el Banco 1 para programar
```

Puertos.

```
BCF STATUS, 6
```

```
CLRF TRISB;
```

```
SALIDA
```

```
CLRF TRISC;
```

```
SALIDA C
```

```
CLRF TRISD;
```

```
SALIDA D
```

```
CLRF TRISE;
```

```
SALIDA E SON TIERRAS
```

```
BCF STATUS, 5 ; Se escoge el Banco 0 para hacer la señal
```

seno

```
BCF STATUS, 6
```

```
CLRF PORTB
```

```
CLRF PORTC
```

```
CLRF PORTD
```

```
PRINCIPAL
```

```
;;  
;;  
;;PUNTO_1
```

```
PUNTO_1
```

```
MOVLW 0X05; BANDERA COLOMBIA
```

```
MOVWF PERRO
```

```
DESARROLLO_1
```

```
CALL COLOMBIA_1
```


PUNTO_8

**MOVLW 0X2F; USB
MOVWF LEON**

DESARROLLO_8

**BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_1
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ LEON, 1
GOTO DESARROLLO_8**

PUNTO_9

**MOVLW 0X2F; USB EN COLORES
MOVWF CONTA2**

DESARROLLO_9

**BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_2
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ CONTA2, 1
GOTO DESARROLLO_9**

PUNTO_10

**MOVLW 0X1F; HOLA_1
MOVWF BBB**

DESARROLLO_10

**BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_3
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ BBB, 1
GOTO DESARROLLO_10**

PUNTO_11

**MOVLW 0X1F; HOLA_2
MOVWF BBB**

DESARROLLO_11

**BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_4
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ BBB, 1
GOTO DESARROLLO_11**

PUNTO_12

**MOVLW 0X1F; HOLA_3
MOVWF BBB**

DESARROLLO_12

**BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_5**

CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ BBB, 1
GOTO DESARROLLO_12

PUNTO_KKK

MOVLW 0X1F; COLOMBIA
MOVWF LEON

DESARROLLO_KKK

BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_COLOMBIA
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ LEON, 1
GOTO DESARROLLO_KKK

PUNTO_PPP

MOVLW 0X1F; JENASH
MOVWF LEON

DESARROLLO_PPP

BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL MENSAJE_JENASH
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ LEON, 1
GOTO DESARROLLO_PPP

PUNTO_0123

MOVLW 0X1F; 0123
MOVWF LEON

DESARROLLO_0123

BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL NUMBERS_0123
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ LEON, 1
GOTO DESARROLLO_0123

PUNTO_456

MOVLW 0X1F; 456
MOVWF LEON

DESARROLLO_456

BSF PCLATH, 3; SEGUNDA PÁGINA
BCF PCLATH, 4
CALL NUMBERS_456
CLRF PCLATH; LIMPIA SEGUNDA PAGINA
DECFSZ LEON, 1
GOTO DESARROLLO_456

PUNTO_789

```
    MOVLW 0X1F; 789
    MOVWF LEON
DESARROLLO_789
    BSF PCLATH, 3; SEGUNDA PÁGINA
    BCF PCLATH, 4
    CALL NUMBERS_789
    CLRF PCLATH; LIMPIA SEGUNDA PAGINA
    DECFSZ LEON, 1
    GOTO DESARROLLO_789
```

```
PUNTO_X
    MOVLW 0X5F; LOGO1
    MOVWF CONTA3
DESARROLLO_X
    BCF PCLATH, 3; TERCERA PÁGINA
    BSF PCLATH, 4
    CALL LOGO_1
    CLRF PCLATH; LIMPIA TERCERA PAGINA
    DECFSZ CONTA3, 1
    GOTO DESARROLLO_X
```

```
PUNTO_COLORS_1
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_1
    CALL COLORS_1
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_1
```

```
PUNTO_COLORS_2
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_2
    CALL COLORS_2
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_2
```

```
PUNTO_COLORS_3
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_3
    CALL COLORS_3
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_3
```

```
PUNTO_COLORS_4
```

```
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_4
    CALL COLORS_4
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_4
```

```
PUNTO_COLORS_5
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_5
    CALL COLORS_5
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_5
```

```
PUNTO_COLORS_6
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_6
    CALL COLORS_6
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_6
```

```
PUNTO_COLORS_7
    MOVLW 0X7F; COLORES
    MOVWF PERRO
DESARROLLO_COLORS_7
    CALL COLORS_7
    DECFSZ PERRO, 1
    GOTO DESARROLLO_COLORS_7
```

```
GOTO PRINCIPAL
```

ANEXO E

```
#INCLUDE<p16f877A.inc> ; Definición del microcontrolador

    CBLOCK 0X20 ; Registro de propósito general

    CONTA1 ; CONTA1 Para subrutina de tiempo
    CONTA2 ; CONTA2 Para subrutina de tiempo

    ENDC

    BSF STATUS, 5 ; Se escoge el Banco 1 para programar Puertos.
    BCF STATUS, 6

    CLRF TRISB ; Salida para leds en puerto B
    CLRF TRISE ; Salida para tierra

    BCF STATUS, 5 ; Se escoge el Banco 0 para hacer programa
    BCF STATUS, 6 ; de la letra A

PRINCIPAL

    BSF PORTE, 0 ; Garantiza la tierra y funcionalidad del dispositivo

    CALL TIEMPO ; Subrutina de tiempo
    CALL LETRA_A ; subrutina de tiempo para letra A
    CALL TIEMPO

    GOTO PRINCIPAL ; Regresa a principal

LETRA_A

    MOVLW 0XFE; 4
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW 0X21; 3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW 0XFE; 2
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;

CAMBIO1
    MOVWF PORTB ; Mueve los datos al puerto B
    RETURN

TIEMPO
```

```

MOVLW 0X7C
MOVWF  CONTA1      ; Valor con que se carga el contador 1
                    ; Depende el ancho de la imagen, entre mayor
                    ; Tiempo se hace ancha la imagen.
NIVEL1
MOVLW 0X5E      ; Valor con que se carga el contador 2.
MOVWF  CONTA2
NIVEL2
DECFSZ CONTA2, 1; Se resta en 1 el contador 2 y se pregunta si es cero.
GOTO NIVEL2; Aun el contador 2 no es cero.
DECFSZ CONTA1, 1; Se resta en 1 el contador 1 y se pregunta si es cero.
GOTO NIVEL1; Aun el contador 1 no es cero.
RETURN; Retorna de la subrutina.

END; Final del programa

```

ANEXO F

```
#INCLUDE<p16f628A.inc> ; Definición del microcontrolador

    CBLOCK 0X20 ; Registro de propósito general

    CONTA1 ; CONTA1 Para subrutina de tiempo
    CONTA2 ; CONTA2 Para subrutina de tiempo

    ENDC

    BSF STATUS, 5 ; Se escoge el Banco 1 para programar Puertos.
    BCF STATUS, 6

    CLRF TRISB ; Salida para leds en puerto B
    CLRF TRISA ; Salida para tierra

    BCF STATUS, 5 ; Se escoge el Banco 0 para hacer programa
    BCF STATUS, 6 ; del mensaje hola

PRINCIPAL

    Bcf PORTA, 0 ; Garantiza la tierra de todos los leds

    CALL TIEMPO ; Subrutina de tiempo
    CALL TIEMPO ; Subrutina de tiempo para todas las letras

    CALL TIEMPO
    CALL LETRA_A_1 ; Subrutina de cada letra

    CALL TIEMPO
    CALL LETRA_L_1

    CALL TIEMPO
    CALL LETRA_O_1

    CALL TIEMPO
    CALL LETRA_H_1

    CALL TIEMPO
    CALL TIEMPO

    GOTO PRINCIPAL ; Regresa a principal
```

LETRA_H_1

```
    MOVLW    0XFE; 1; Dato en hexadecimal
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X10; 2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X10; 3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE; 4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;
```

LETRA_O_1

```
    MOVLW    0X7E; 1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81; 2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81; 3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X7E; 4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;
```

LETRA_L_1

```
    MOVLW    0X80; 1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X80; 2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X80; 3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE; 4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;
```

LETRA_A_1

```
    MOVLW    0XFE; 1
    CALL CAMBIO1
```

```

CALL TIEMPO
MOVLW    0X21; 2
CALL CAMBIO1
CALL TIEMPO
MOVLW    0X21; 3
CALL CAMBIO1
CALL TIEMPO
MOVLW    0XFE; 4
CALL CAMBIO1
CALL TIEMPO
RETURN;

```

```

CAMBIO1
    MOVWF    PORTB    ; Mueve los datos al puerto B
    RETURN

```

TIEMPO

```

MOVLW 0X7C    ; Con el dato cambia el ancho de la letra
            ; Valor con que se carga el contador 1
MOVWF  CONTA1 ; Depende el ancho de la imagen, entre mayor

```

; Tiempo se hace ancha la imagen.

NIVEL1

```

MOVLW 0X5E    ; Con el dato cambia el ancho de la letra

```

```

MOVWF  CONTA2    ; Valor con que se carga el contador 2.

```

NIVEL2

```

DECFSZ CONTA2, 1; Se resta en 1 el contador 2 y se pregunta si es cero.

```

```

GOTO NIVEL2; Aun el contador 2 no es cero.

```

```

DECFSZ CONTA1, 1; Se resta en 1 el contador 1 y se pregunta si es cero.

```

```

GOTO NIVEL1; Aun el contador 1 no es cero.

```

```

RETURN; Retorna de la subrutina.

```

```

END; Final del programa

```

ANEXO G

```
#INCLUDE<p16f877A.inc>      ; Definición del microcontrolador

    CBLOCK 0X20      ; Registro de propósito general

    CONTA1           ; CONTA1 Para subrutina de tiempo
    CONTA2           ; CONTA2 Para subrutina de tiempo

    ENDC

    BSF STATUS, 5    ; Se escoge el Banco 1 para programar Puertos.
    BCF STATUS, 6

    CLRF TRISB       ; Salida para leds en puerto B
    CLRF TRISC       ; Salida para leds
    CLRF TRIS D      ; SALIDA PARA LEDS

    BCF STATUS, 5    ; Se escoge el Banco 0 para hacer programa
    BCF STATUS, 6    ; del mensaje hola

PRINCIPAL

Bcf   PORTA, 0      ; Garantiza la tierra de todos los leds

CALL TIEMPO         ; Subrutina de tiempo
CALL TIEMPO         ; Subrutina de tiempo para todas las letras

CALL LETRA_B_3; Subrutina de cada letra

CALL TIEMPO
CALL LETRA_S_2

CALL TIEMPO
CALL LETRA_U_1

CALL TIEMPO
CALL TIEMPO

GOTO PRINCIPAL     ; Regresa a principal
```

LETRA_U_1

```
MOVLW    0X7F; 1
CALL CAMBIO1
CALL TIEMPO
MOVLW    0X80; 2
CALL CAMBIO1
CALL TIEMPO
MOVLW    0X80; 3
CALL CAMBIO1
CALL TIEMPO
MOVLW    0X7F; 4
CALL CAMBIO1
CALL TIEMPO
RETURN;
```

LETRA_S_2

```
MOVLW    0X72; 1
CALL CAMBIO2
CALL TIEMPO
MOVLW    0X89; 2
CALL CAMBIO2
CALL TIEMPO
MOVLW    0X89; 3
CALL CAMBIO2
CALL TIEMPO
MOVLW    0X46; 4
CALL CAMBIO2
CALL TIEMPO
RETURN;
```

LETRA_B_3

```
MOVLW    0X66; 1
CALL CAMBIO3
CALL TIEMPO
MOVLW    0X99; 2
CALL CAMBIO3
CALL TIEMPO
MOVLW    0X99; 3
CALL CAMBIO3
CALL TIEMPO
MOVLW    0XFE; 4
CALL CAMBIO3
CALL TIEMPO
RETURN;
```

```

CAMBIO1
    MOVWF    PORTB; Mueve los datos al puerto B
    RETURN
CAMBIO2
    MOVWF    PORTC; Mueve los datos al puerto C
    RETURN
CAMBIO3
    MOVWF    PORTD; Mueve los datos al puerto D
    RETURN
TIEMPO

MOVLW 0X7C          ; Con el dato cambia el ancho de la letra
                   ; Valor con que se carga el contador 1
MOVWF    CONTA1    ; Depende el ancho de la imagen, entre mayor

                   ; Tiempo se hace ancha la imagen.
NIVEL1

MOVLW 0X5E          ; Con el dato cambia el ancho de la letra

MOVWF    CONTA2    ; Valor con que se carga el contador 2.

NIVEL2

DECFSZ CONTA2, 1; Se resta en 1 el contador 2 y se pregunta si es cero.
GOTO NIVEL2; Aun el contador 2 no es cero.

DECFSZ CONTA1, 1; Se resta en 1 el contador 1 y se pregunta si es cero.
GOTO NIVEL1; Aun el contador 1 no es cero.

RETURN; Retorna de la subrutina.

END; Final del programa

```

ANEXO I

#INCLUDE<p16f877A.inc>

ORG 0X0000;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

**CBLOCK 0X20
MONO1
MONO2
MONO3
MONO4
CAMBIO
CAMBIO_1
CONTA3
CONTA1
CONTA2
PERRO
GATO
LEON
BBB
ENDC**

BSF STATUS,5 ; Se escoge el Banco 1 para programar
**Puertos.
BCF STATUS,6**

**CLRF TRISB; SALIDA
CLRF TRISC; SALIDA C
CLRF TRISD; SALIDA D
CLRF TRISE; SALIDA E SON TIERRAS**

BCF STATUS,5 ; Se escoge el Banco 0 para hacer la señal
**seno
BCF STATUS,6**

**CLRF PORTB
CLRF PORTC
CLRF PORTD**

PRINCIPAL

;;
;PUNTO_1

**PUNTO_1
MOVLW 0X05 ;BANDERA COLOMBIA
MOVWF PERRO
DESARROLLO_1
CALL COLOMBIA_1
DECFSZ PERRO,1
GOTO DESARROLLO_1**

```

.....
PUNTO_2
    MOVLW 0X4F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;MILLONARIOS
    MOVWF PERRO
DESARROLLO_2
    CALL MILLONARIOS
    DECFSZ    PERRO,1
    GOTO      DESARROLLO_2

```

```

.....
PUNTO_3
    MOVLW 0X04F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;SANTAFE
    MOVWF MONO1
DESARROLLO_3
    CALL SANTAFE
    DECFSZ    MONO1,1
    GOTO      DESARROLLO_3

```

```

.....
PUNTO_4
    MOVLW 0X05F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;NACIONAL
    MOVWF MONO1
DESARROLLO_4
    CALL NACIONAL
    DECFSZ    MONO1,1
    GOTO      DESARROLLO_4

```

```

PUNTO_5
    MOVLW 0X4F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CUBO AZUL
    MOVWF MONO1
DESARROLLO_5
    CALL CUBO_AZUL
    DECFSZ    MONO1,1
    GOTO      DESARROLLO_5

```

```

PUNTO_6
    MOVLW 0X4F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CUBO ROJO
    MOVWF MONO1
DESARROLLO_6
    CALL CUBO_ROJO
    DECFSZ    MONO1,1
    GOTO      DESARROLLO_6

```

```

PUNTO_7
    MOVLW 0X4F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;CUBO VERDE
    MOVWF MONO1
DESARROLLO_7
    CALL CUBO_VERDE
    DECFSZ    MONO1,1
    GOTO      DESARROLLO_7

```

```

PUNTO_8
    MOVLW 0X2F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;USB

```

```

MOVWF LEON
DESARROLLO_8
BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_1
CLRF PCLATH;;;;;;;;;;;;;;;;;;LIMPIA SEGUNDA PAGINA
DECFSZ LEON,1
GOTO DESARROLLO_8

PUNTO_9
MOVLW 0X2F ;;;;;;;;;;;;;;;;;;USBEN COLORES
MOVWF CONTA2
DESARROLLO_9
BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_2
CLRF PCLATH;;;;;;;;;;;;;;;;;;LIMPIA SEGUNDA PAGINA
DECFSZ CONTA2,1
GOTO DESARROLLO_9

PUNTO_10
MOVLW 0X1F ;;;;;;;;;;;;;;;;;;HOLA_1
MOVWF BBB
DESARROLLO_10
BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_3
CLRF PCLATH;;;;;;;;;;;;;;;;;;LIMPIA SEGUNDA PAGINA
DECFSZ BBB,1
GOTO DESARROLLO_10

PUNTO_11
MOVLW 0X1F ;;;;;;;;;;;;;;;;;;HOLA_2
MOVWF BBB
DESARROLLO_11
BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_4
CLRF PCLATH;;;;;;;;;;;;;;;;;;LIMPIA SEGUNDA PAGINA
DECFSZ BBB,1
GOTO DESARROLLO_11

PUNTO_12
MOVLW 0X1F ;;;;;;;;;;;;;;;;;;HOLA_3
MOVWF BBB
DESARROLLO_12
BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_5
CLRF PCLATH;;;;;;;;;;;;;;;;;;LIMPIA SEGUNDA PAGINA
DECFSZ BBB,1

```

GOTO DESARROLLO_12

PUNTO_KKK

MOVLW 0X1FCOLOMBIA
MOVWF LEON

DESARROLLO_KKK

BSF PCLATH,3SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_COLOMBIA
CLRF PCLATH;....., LIMPIA SEGUNDA PAGINA
DECFSZ LEON,1
GOTO DESARROLLO_KKK

PUNTO_PPP

MOVLW 0X1FJENASH
MOVWF LEON

DESARROLLO_PPP

BSF PCLATH,3SEGUNDA PAGINA
BCF PCLATH,4
CALL MENSAJE_JENASH
CLRF PCLATH;....., LIMPIA SEGUNDA PAGINA
DECFSZ LEON,1
GOTO DESARROLLO_PPP

PUNTO_0123

MOVLW 0X1F0123
MOVWF LEON

DESARROLLO_0123

BSF PCLATH,3SEGUNDA PAGINA
BCF PCLATH,4
CALL NUMBERS_0123
CLRF PCLATH;....., LIMPIA SEGUNDA PAGINA
DECFSZ LEON,1
GOTO DESARROLLO_0123

PUNTO_456

MOVLW 0X1F456
MOVWF LEON

DESARROLLO_456

BSF PCLATH,3SEGUNDA PAGINA
BCF PCLATH,4
CALL NUMBERS_456
CLRF PCLATH;....., LIMPIA SEGUNDA PAGINA
DECFSZ LEON,1
GOTO DESARROLLO_456

PUNTO_789

MOVLW 0X1F789
MOVWF LEON

```

DESARROLLO_789
  BSF PCLATH,3 ;;;;;;;;;;;;;;;;;;SEGUNDA PAGINA
  BCF PCLATH,4
  CALL NUMBERS_789
  CLRF PCLATH;,,,,,,,,,,,,,LIMPIA SEGUNDA PAGINA
  DECFSZ LEON,1
  GOTO DESARROLLO_789

```

```

PUNTO_X
  MOVLW 0X5F ;;;;;;;;;;;;;;;;;;LOGO1
  MOVWF CONTA3

```

```

DESARROLLO_X
  BCF PCLATH,3 ;;;;;;;;;;;;;;;;;;TERCERA PAGINA
  BSF PCLATH,4
  CALL LOGO_1
  CLRF PCLATH;,,,,,,,,,,,,,LIMPIA TERCERA PAGINA
  DECFSZ CONTA3,1
  GOTO DESARROLLO_X

```

```

PUNTO_COLORS_1
  MOVLW 0X7F ;;;;;;;;;;;;;;;;;;COLORES
  MOVWF PERRO

```

```

DESARROLLO_COLORS_1
  CALL COLORS_1
  DECFSZ PERRO,1
  GOTO DESARROLLO_COLORS_1

```

```

PUNTO_COLORS_2
  MOVLW 0X7F ;;;;;;;;;;;;;;;;;;COLORES
  MOVWF PERRO

```

```

DESARROLLO_COLORS_2
  CALL COLORS_2
  DECFSZ PERRO,1
  GOTO DESARROLLO_COLORS_2

```

```

PUNTO_COLORS_3
  MOVLW 0X7F ;;;;;;;;;;;;;;;;;;COLORES
  MOVWF PERRO

```

```

DESARROLLO_COLORS_3
  CALL COLORS_3
  DECFSZ PERRO,1
  GOTO DESARROLLO_COLORS_3

```

```

PUNTO_COLORS_4
  MOVLW 0X7F ;;;;;;;;;;;;;;;;;;COLORES
  MOVWF PERRO

```

```

DESARROLLO_COLORS_4
    CALL COLORS_4
    DECFSZ PERRO,1
    GOTO DESARROLLO_COLORS_4

```

```

PUNTO_COLORS_5
    MOVLW 0X7F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;COLORES
    MOVWF PERRO
DESARROLLO_COLORS_5
    CALL COLORS_5
    DECFSZ PERRO,1
    GOTO DESARROLLO_COLORS_5

```

```

PUNTO_COLORS_6
    MOVLW 0X7F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;COLORES
    MOVWF PERRO
DESARROLLO_COLORS_6
    CALL COLORS_6
    DECFSZ PERRO,1
    GOTO DESARROLLO_COLORS_6

```

```

PUNTO_COLORS_7
    MOVLW 0X7F ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;COLORES
    MOVWF PERRO
DESARROLLO_COLORS_7
    CALL COLORS_7
    DECFSZ PERRO,1
    GOTO DESARROLLO_COLORS_7

```

GOTO PRINCIPAL

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;PROGRAM BANDERA DE COLOMBIA
COLOMBIA_1

```

```

    CALL TIEMPO_PP_ESPACIO
    CALL TIEMPO_COLORES_1
    CALL TIEMPO_PP_ESPACIO
    RETURN

```

```

ESPACIO_123
    MOVLW    b'011';AZUL
    MOVWF    PORTE
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTD
    MOVLW    b'00000000';
    MOVWF    PORTC

```

```

    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'110';ROJO
    MOVWF    PORTE
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTB
    RETURN
COLORES_123
    MOVLW    b'010';AMARILLO
    MOVWF    PORTE
    MOVLW    b'11111111';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTB
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    MOVLW    b'101';AZUL
    MOVWF    PORTE
    MOVLW    b'11111111';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTC
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    MOVLW    b'110';ROJO
    MOVWF    PORTE
    MOVLW    b'11111111';
    MOVWF    PORTD
    MOVLW    b'00000000';
    MOVWF    PORTD
    RETURN

TIEMPO_PP_ESPACIO
    MOVLW 0XF ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL_PP_ESPACIO.1
    MOVLW 0XF; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL_PP_ESPACIO.1.1
    CALL ESPACIO_123
    DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
    GOTO NIVEL_PP_ESPACIO.1.1 ; Aun el contador 2 no es cero.
    DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
    GOTO NIVEL_PP_ESPACIO.1 ; Aun el contador 1 no es cero.
    RETURN

TIEMPO_COLORES_1

```



```

MOVWF    PORTB
MOVLW    b'00000100';
MOVWF    PORTC
MOVLW    b'00000100';
MOVWF    PORTD
CALL AZUL_M
;.....22
MOVWLW   b'10111110';
MOVWF    PORTB
MOVLW    b'00000011';
MOVWF    PORTC
MOVLW    b'00000100';
MOVWF    PORTD
CALL AZUL_M
;.....21
MOVLW    b'01111100';
MOVWF    PORTB
MOVLW    b'00000000';
MOVWF    PORTC
MOVLW    b'00001000';
MOVWF    PORTD
CALL AZUL_M
;.....20
MOVLW    b'11111000';
MOVWF    PORTB
MOVLW    b'11000000';
MOVWF    PORTC
MOVLW    b'00001011';
MOVWF    PORTD
CALL AZUL_M
;.....19
MOVLW    b'11111000';
MOVWF    PORTB
MOVLW    b'10000000';
MOVWF    PORTC
MOVLW    b'00010000';
MOVWF    PORTD
CALL AZUL_M
;.....18
MOVLW    b'11111000';
MOVWF    PORTB
MOVLW    b'00000001';
MOVWF    PORTC
MOVLW    b'00010001';
MOVWF    PORTD
CALL AZUL_M
;.....17
MOVLW    b'11111000';
MOVWF    PORTB
MOVLW    b'10000001';

```

```

MOVWF    PORTC
MOVLW    b'00010000';
MOVWF    PORTD
CALL AZUL_M
;.....16
MOVLW    b'11111000';
MOVWF    PORTB
MOVLW    b'11000011';
MOVWF    PORTC
MOVLW    b'00100011';
MOVWF    PORTD
CALL AZUL_M
;.....15
MOVLW    b'11111100';
MOVWF    PORTB
MOVLW    b'00000011';
MOVWF    PORTC
MOVLW    b'01000000';
MOVWF    PORTD
CALL AZUL_M
;.....14
MOVLW    b'11111110';
MOVWF    PORTB
MOVLW    b'00000111';
MOVWF    PORTC
MOVLW    b'11000000';
MOVWF    PORTD
CALL AZUL_M
;.....13
MOVLW    b'11111100';
MOVWF    PORTB
MOVLW    b'00000111';
MOVWF    PORTC
MOVLW    b'01000000';
MOVWF    PORTD
CALL AZUL_M
;.....12
MOVLW    b'00111000';
MOVWF    PORTB
MOVLW    b'00001111';
MOVWF    PORTC
MOVLW    b'00100000';
MOVWF    PORTD
CALL AZUL_M
;.....11
MOVLW    b'11011000';
MOVWF    PORTB
MOVLW    b'00001110';
MOVWF    PORTC
MOVLW    b'00010000';

```

```

MOVWF    PORTD
CALL AZUL_M
;.....10
MOV LW   b'00111000';
MOVWF   PORTB
MOV LW   b'00011111';
MOVWF   PORTC
MOV LW   b'00010000';
MOVWF   PORTD
CALL AZUL_M
;.....9
MOV LW   b'11011000';
MOVWF   PORTB
MOV LW   b'00011110';
MOVWF   PORTC
MOV LW   b'00010000';
MOVWF   PORTD
CALL AZUL_M
;.....8
MOV LW   b'00111000';
MOVWF   PORTB
MOV LW   b'00111111';
MOVWF   PORTC
MOV LW   b'00001000';
MOVWF   PORTD
CALL AZUL_M
;.....7
MOV LW   b'11111100';
MOVWF   PORTB
MOV LW   b'00111111';
MOVWF   PORTC
MOV LW   b'00001000';
MOVWF   PORTD
CALL AZUL_M
;.....6
MOV LW   b'11111110';
MOVWF   PORTB
MOV LW   b'01111111';
MOVWF   PORTC
MOV LW   b'00000100';
MOVWF   PORTD
CALL AZUL_M
;.....5
MOV LW   b'01111100';
MOVWF   PORTB
MOV LW   b'01111100';
MOVWF   PORTC
MOV LW   b'00000100';
MOVWF   PORTD

```

```

CALL AZUL_M
;.....4
    MOVLW    b'00111000';
    MOVWF    PORTB
    MOVLW    b'11111000';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_M
;.....3
    MOVLW    b'00010000';
    MOVWF    PORTB
    MOVLW    b'11110000';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_M
;.....2
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'11100000';
    MOVWF    PORTC
    MOVLW    b'00000001';
    MOVWF    PORTD
    CALL AZUL_M
;.....1
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_M
    CALL TIEMPO_M

    RETURN
;.....
AZUL_M
    MOVLW    b'101'
    MOVWF    PORTE
    CALL TIEMPO_M
    RETURN
TIEMPO_M
    MOVLW 0X1F ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL1_M
    MOVLW 0X1F ; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL2_M
    DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.

```

GOTO NIVEL2_M ; Aun el contador 2 no es cero.
 DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
 cero.

GOTO NIVEL1_M ; Aun el contador 1 no es cero.
 RETURN ; Retorna de la subrutina.

SANTAFE;.....SANTAFE

MOVLW b'00000000';
 MOVWF PORTB
 MOVWF PORTC
 MOVWF PORTD
 CALL TIEMPO_S
 CALL TIEMPO_S

;.....

MOVLW b'11100000';
 MOVWF PORTB
 MOVLW b'11111111';
 MOVWF PORTC
 MOVLW b'00011111';
 MOVWF PORTD
 CALL ROJO_S

;.....17

MOVLW b'01110000';
 MOVWF PORTB
 MOVLW b'00000000';
 MOVWF PORTC
 MOVLW b'00100000';
 MOVWF PORTD
 CALL ROJO_S

;.....16

MOVLW b'01111000';
 MOVWF PORTB
 MOVLW b'00000000';
 MOVWF PORTC
 MOVLW b'01000110';
 MOVWF PORTD
 CALL ROJO_S

;.....15

MOVLW b'01111000';
 MOVWF PORTB
 MOVLW b'00000000';
 MOVWF PORTC
 MOVLW b'10001111';
 MOVWF PORTD
 CALL ROJO_S

;.....14

MOVLW b'01111000';
 MOVWF PORTB
 MOVLW b'10000000';
 MOVWF PORTC

```

    MOVLW    b'10011111';
    MOVWF    PORTD
    CALL ROJO_S
;.....13
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'10000000';
    MOVWF    PORTC
    MOVLW    b'10011111';
    MOVWF    PORTD
    CALL ROJO_S
;.....12
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'10001111';
    MOVWF    PORTD
    CALL ROJO_S
;.....11
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'10001110';
    MOVWF    PORTD
    CALL ROJO_S
;.....10
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'10000000';
    MOVWF    PORTD
    CALL ROJO_S
;.....9
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'10000000';
    MOVWF    PORTD
    CALL ROJO_S
;.....8
    MOVLW    b'01111000';
    MOVWF    PORTB
    MOVLW    b'00000000';
    MOVWF    PORTC
    MOVLW    b'10000000';
    MOVWF    PORTD

```

```

CALL ROJO_S
;.....7
  MOVLW    b'01111000';
  MOVWF    PORTB
  MOVLW    b'00000000';
  MOVWF    PORTC
  MOVLW    b'10000000';
  MOVWF    PORTD
  CALL ROJO_S
;.....6
  MOVLW    b'01111000';
  MOVWF    PORTB
  MOVLW    b'00000000';
  MOVWF    PORTC
  MOVLW    b'10000000';
  MOVWF    PORTD
  CALL ROJO_S
;.....5
  MOVLW    b'01111000';
  MOVWF    PORTB
  MOVLW    b'00000000';
  MOVWF    PORTC
  MOVLW    b'01000000';
  MOVWF    PORTD
  CALL ROJO_S
;.....4
  MOVLW    b'01111000';
  MOVWF    PORTB
  MOVLW    b'00000000';
  MOVWF    PORTC
  MOVLW    b'00100000';
  MOVWF    PORTD
  CALL ROJO_S
;.....3
  MOVLW    b'01111000';
  MOVWF    PORTB
  MOVLW    b'00000000';
  MOVWF    PORTC
  MOVLW    b'00010000';
  MOVWF    PORTD
  CALL ROJO_S
;.....2
  MOVLW    b'11100000';
  MOVWF    PORTB
  MOVLW    b'11111111';
  MOVWF    PORTC
  MOVLW    b'00001111';
  MOVWF    PORTD
  CALL ROJO_S

```

```

;.....1
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_S
    CALL TIEMPO_S

    RETURN
;.....
ROJO_S
    MOVLW    b'110';ROJO
    MOVWF    PORTE
    CALL TIEMPO_S
    RETURN
TIEMPO_S
    MOVLW 0X25 ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL1_S
    MOVLW 0X25 ; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL2_S
    DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
    GOTO NIVEL2_S ; Aun el contador 2 no es cero.
    DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
    GOTO NIVEL1_S ; Aun el contador 1 no es cero.
    RETURN ; Retorna de la subrutina.

NACIONAL
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_N
    CALL TIEMPO_N
;.....nacional
    MOVLW    b'00011110';
    MOVWF    PORTB
    MOVLW    b'11111110';
    MOVWF    PORTC
    MOVLW    b'00000111';
    MOVWF    PORTD
    CALL VERDE_N
;.....19
    MOVLW    b'01010010';
    MOVWF    PORTB
    MOVLW    b'00000011';

```

```

MOVWF    PORTC
MOVLW    b'0000100';
MOVWF    PORTD
CALL VERDE_N
;.....18
MOVLW    b'10100110';
MOVWF    PORTB
MOVLW    b'00000010';
MOVWF    PORTC
MOVLW    b'00001000';
MOVWF    PORTD
CALL VERDE_N
;.....17
MOVLW    b'01100110';
MOVWF    PORTB
MOVLW    b'00000011';
MOVWF    PORTC
MOVLW    b'00001000';
MOVWF    PORTD
CALL VERDE_N
;.....16
MOVLW    b'10100100';
MOVWF    PORTB
MOVLW    b'10001010';
MOVWF    PORTC
MOVLW    b'00010111';
MOVWF    PORTD
CALL VERDE_N
;.....15
MOVLW    b'01100100';
MOVWF    PORTB
MOVLW    b'00001011';
MOVWF    PORTC
MOVLW    b'00010010';
MOVWF    PORTD
CALL VERDE_N
;.....14
MOVLW    b'10100100';
MOVWF    PORTB
MOVLW    b'00011011';
MOVWF    PORTC
MOVLW    b'00100001';
MOVWF    PORTD
CALL VERDE_N
;.....13
MOVLW    b'01100110';
MOVWF    PORTB
MOVLW    b'10011011';
MOVWF    PORTC

```

```

    MOVLW    b'00100111';
    MOVWF    PORTD
    CALL VERDE_N
;.....12
    MOVLW    b'10100010';
    MOVWF    PORTB
    MOVLW    b'00111010';
    MOVWF    PORTC
    MOVLW    b'01000000';
    MOVWF    PORTD
    CALL VERDE_N
;.....11
    MOVLW    b'01100010';
    MOVWF    PORTB
    MOVLW    b'01111011';
    MOVWF    PORTC
    MOVLW    b'10000000';
    MOVWF    PORTD
    CALL VERDE_N
;.....10
    MOVLW    b'10100110';
    MOVWF    PORTB
    MOVLW    b'11111010';
    MOVWF    PORTC
    MOVLW    b'01000000';
    MOVWF    PORTD
    CALL VERDE_N
;.....9
    MOVLW    b'01100100';
    MOVWF    PORTB
    MOVLW    b'00111011';
    MOVWF    PORTC
    MOVLW    b'00100001';
    MOVWF    PORTD
    CALL VERDE_N
;.....8
    MOVLW    b'10100100';
    MOVWF    PORTB
    MOVLW    b'01011010';
    MOVWF    PORTC
    MOVLW    b'00100001';
    MOVWF    PORTD
    CALL VERDE_N
;.....7
    MOVLW    b'01100100';
    MOVWF    PORTB
    MOVLW    b'01101011';
    MOVWF    PORTC
    MOVLW    b'00010001';
    MOVWF    PORTD

```

```

CALL VERDE_N
;.....6
    MOVLW    b'10100100';
    MOVWF    PORTB
    MOVLW    b'00011010';
    MOVWF    PORTC
    MOVLW    b'00010010';
    MOVWF    PORTD
    CALL VERDE_N
;.....5
    MOVLW    b'01100110';
    MOVWF    PORTB
    MOVLW    b'11111011';
    MOVWF    PORTC
    MOVLW    b'00010111';
    MOVWF    PORTD
    CALL VERDE_N
;.....4
    MOVLW    b'10100110';
    MOVWF    PORTB
    MOVLW    b'00000010';
    MOVWF    PORTC
    MOVLW    b'00001000';
    MOVWF    PORTD
    CALL VERDE_N
;.....3
    MOVLW    b'01010010';
    MOVWF    PORTB
    MOVLW    b'00000011';
    MOVWF    PORTC
    MOVLW    b'00000100';
    MOVWF    PORTD
    CALL VERDE_N
;.....2
    MOVLW    b'00011110';
    MOVWF    PORTB
    MOVLW    b'11111110';
    MOVWF    PORTC
    MOVLW    b'00000111';
    MOVWF    PORTD
    CALL VERDE_N
;.....1
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_N
    CALL TIEMPO_N

RETURN

```

```

;.....
VERDE_N
    MOVLW    B'011';VERDE
    MOVWF    PORTE
    CALL TIEMPO_M
    RETURN
TIEMPO_N
    MOVLW 0X2F ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL1_N
    MOVLW 0X25 ; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL2_N
    DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
    GOTO NIVEL2_N ; Aun el contador 2 no es cero.
    DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
    GOTO NIVEL1_N ; Aun el contador 1 no es cero.
    RETURN ; Retorna de la subrutina.

CUBO_AZUL
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_CUBO_AZUL
    CALL TIEMPO_CUBO_AZUL
;.....
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'00111111';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....12
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'01100000';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....11
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'10100001';
    MOVWF    PORTC

```

```

    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....10
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100010';
    MOVWF    PORTC
    MOVLW    b'00000001';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....9
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'11111100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....8
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....7
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....6
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'00111111';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....5
    MOVLW    b'10000000';
    MOVWF    PORTB
    MOVLW    b'01000100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD

```

```

CALL AZUL_CUBO_AZUL
;.....4
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'10000101';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....3
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'00000110';
    MOVWF    PORTC
    MOVLW    b'00000011';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....2
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'11111100';
    MOVWF    PORTC
    MOVLW    b'00000011';
    MOVWF    PORTD
    CALL AZUL_CUBO_AZUL
;.....1

    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_CUBO_AZUL
    CALL TIEMPO_CUBO_AZUL

    RETURN
;.....
AZUL_CUBO_AZUL
    MOVLW    b'101'
    MOVWF    PORTE
    CALL TIEMPO_CUBO_AZUL
    RETURN

TIEMPO_CUBO_AZUL
    MOVLW 0X1F ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL1_CUBO_AZUL
    MOVLW 0X1A ; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL2_CUBO_AZUL

```

DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.

GOTO NIVEL2_CUBO_AZUL ; Aun el contador 2 no es cero.

DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.

GOTO NIVEL1_CUBO_AZUL ; Aun el contador 1 no es cero.

RETURN ; Retorna de la subrutina.

CUBO_ROJO

```
MOVLW    b'00000000';  
MOVWF    PORTB  
MOVWF    PORTC  
MOVWF    PORTD  
CALL TIEMPO_CUBO_ROJO  
CALL TIEMPO_CUBO_ROJO
```

.....

```
MOVLW    b'11000000';  
MOVWF    PORTB  
MOVLW    b'00111111';  
MOVWF    PORTC  
MOVLW    b'00000000';  
MOVWF    PORTD  
CALL AZUL_CUBO_ROJO
```

.....12

```
MOVLW    b'11000000';  
MOVWF    PORTB  
MOVLW    b'01100000';  
MOVWF    PORTC  
MOVLW    b'00000000';  
MOVWF    PORTD  
CALL AZUL_CUBO_ROJO
```

.....11

```
MOVLW    b'01000000';  
MOVWF    PORTB  
MOVLW    b'10100001';  
MOVWF    PORTC  
MOVLW    b'00000000';  
MOVWF    PORTD  
CALL AZUL_CUBO_ROJO
```

.....10

```
MOVLW    b'01000000';  
MOVWF    PORTB  
MOVLW    b'00100010';  
MOVWF    PORTC  
MOVLW    b'00000001';  
MOVWF    PORTD  
CALL AZUL_CUBO_ROJO
```

.....9

```
MOVLW    b'01000000';  
MOVWF    PORTB
```

```

    MOVLW    b'11111100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....8
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....7
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....6
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'00111111';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....5
    MOVLW    b'10000000';
    MOVWF    PORTB
    MOVLW    b'01000100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....4
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'10000101';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....3
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'00000110';
    MOVWF    PORTC

```

```

    MOVLW    b'00000011';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....2
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVLW    b'11111100';
    MOVWF    PORTC
    MOVLW    b'00000011';
    MOVWF    PORTD
    CALL AZUL_CUBO_ROJO
;.....1

    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_CUBO_ROJO
    CALL TIEMPO_CUBO_ROJO

    RETURN
;.....
AZUL_CUBO_ROJO
    MOVLW    b'110';ROJO
    MOVWF    PORTE
    CALL TIEMPO_CUBO_ROJO
    RETURN
TIEMPO_CUBO_ROJO
    MOVLW 0X1F ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
    MOVWF CONTA1
NIVEL1_CUBO_ROJO
    MOVLW 0X1A ; Valor con que se carga el contador 2.
    MOVWF CONTA2
NIVEL2_CUBO_ROJO
    DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
    GOTO NIVEL2_CUBO_ROJO ; Aun el contador 2 no es cero.
    DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
    GOTO NIVEL1_CUBO_ROJO ; Aun el contador 1 no es cero.
    RETURN ; Retorna de la subrutina.

CUBO_VERDE
    MOVLW    b'00000000';
    MOVWF    PORTB
    MOVWF    PORTC
    MOVWF    PORTD
    CALL TIEMPO_CUBO_VERDE
    CALL TIEMPO_CUBO_VERDE

```

```

;.....
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'00111111';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....12
    MOVLW    b'11000000';
    MOVWF    PORTB
    MOVLW    b'01100000';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....11
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'10100001';
    MOVWF    PORTC
    MOVLW    b'00000000';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....10
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100010';
    MOVWF    PORTC
    MOVLW    b'00000001';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....9
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'11111100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....8
    MOVLW    b'01000000';
    MOVWF    PORTB
    MOVLW    b'00100100';
    MOVWF    PORTC
    MOVLW    b'00000010';
    MOVWF    PORTD
    CALL AZUL_CUBO_VERDE
;.....7
    MOVLW    b'01000000';

```

```

MOVWF    PORTB
MOVLW    b'00100100';
MOVWF    PORTC
MOVLW    b'00000010';
MOVWF    PORTD
CALL AZUL_CUBO_VERDE
;.....6
MOVWF    PORTB
MOVLW    b'11000000';
MOVWF    PORTC
MOVLW    b'00111111';
MOVWF    PORTD
MOVLW    b'00000010';
CALL AZUL_CUBO_VERDE
;.....5
MOVWF    PORTB
MOVLW    b'10000000';
MOVWF    PORTC
MOVLW    b'01000100';
MOVWF    PORTD
MOVLW    b'00000010';
CALL AZUL_CUBO_VERDE
;.....4
MOVWF    PORTB
MOVLW    b'00000000';
MOVWF    PORTC
MOVLW    b'10000101';
MOVWF    PORTD
MOVLW    b'00000010';
CALL AZUL_CUBO_VERDE
;.....3
MOVWF    PORTB
MOVLW    b'00000000';
MOVWF    PORTC
MOVLW    b'00000110';
MOVWF    PORTD
MOVLW    b'00000011';
CALL AZUL_CUBO_VERDE
;.....2
MOVWF    PORTB
MOVLW    b'00000000';
MOVWF    PORTC
MOVLW    b'11111100';
MOVWF    PORTD
MOVLW    b'00000011';
CALL AZUL_CUBO_VERDE
;.....1
MOVWF    PORTB
MOVLW    b'00000000';

```

```

MOVWF    PORTC
MOVWF    PORTD
CALL TIEMPO_CUBO_VERDE
CALL TIEMPO_CUBO_VERDE

```

```

RETURN

```

```

;.....
AZUL_CUBO_VERDE

```

```

    MOVLW    B'011';VERDE
    MOVWF    PORTE
    CALL TIEMPO_CUBO_VERDE
    RETURN

```

```

TIEMPO_CUBO_VERDE

```

```

    MOVLW    0X1F ; Valor con que se carga el contador 1.

```

```

CONTADORES DE GROSOR DE LINEA

```

```

    MOVWF    CONTA1

```

```

NIVEL1_CUBO_VERDE

```

```

    MOVLW    0X1A ; Valor con que se carga el contador 2.

```

```

    MOVWF    CONTA2

```

```

NIVEL2_CUBO_VERDE

```

```

    DECFSZ   CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.

```

```

    GOTO NIVEL2_CUBO_VERDE ; Aun el contador 2 no es cero.

```

```

    DECFSZ   CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es

```

```

cero.

```

```

    GOTO NIVEL1_CUBO_VERDE ; Aun el contador 1 no es cero.

```

```

    RETURN ; Retorna de la subrutina.

```

```

;.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

COLORS_1

```

```

    CALL ROJO_COLOR
    MOVLW    b'11111111';
    MOVWF    PORTB

```

```

;.....
;.....

```

```

    MOVLW    b'11111111';
    MOVWF    PORTC

```

```

;.....
;.....

```

```

    MOVLW    b'11111111';
    MOVWF    PORTD
    CALL TIEMPO_COLOR
    RETURN

```

```

COLORS_2

```

```

    CALL VERDE_COLOR

```

```

    MOVLW    b'11111111';
    MOVWF    PORTB
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTC
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTD
    CALL TIEMPO_COLOR
    RETURN
COLORS_3

```

```

    CALL AZUL_COLOR
    MOVLW    b'11111111';
    MOVWF    PORTB
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTC
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTD
    CALL TIEMPO_COLOR
    RETURN
COLORS_4

```

```

    CALL CIAN_COLOR
    MOVLW    b'11111111';
    MOVWF    PORTB
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTC
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTD
    CALL TIEMPO_COLOR
    RETURN
COLORS_5

```

```

    CALL AMARILLO_COLOR
    MOVLW    b'11111111';
    MOVWF    PORTB
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTC
;
;
;
    MOVLW    b'11111111';
    MOVWF    PORTD
    CALL TIEMPO_COLOR
    RETURN

```



```
MOVWF PORTE
RETURN
BLANCO_COLOR
MOVLW B'000';BLANCO
MOVWF PORTE
RETURN
```

TIEMPO_COLOR

MOVLW 0X5F ; Valor con que se carga el contador 1.

CONTADORES DE GROSOR DE LINEA

MOVWF CONTA1

NIVEL1_COLOR

MOVLW 0X1F ; Valor con que se carga el contador 2.

MOVWF CONTA2

NIVEL2_COLOR

DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es cero.

GOTO NIVEL2_COLOR

DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es cero.

GOTO NIVEL1_COLOR ; Aun el contador 1 no es cero.

RETURN ; Retorna de la subrutina.

INCLUDE<LOGO1F.INC>;LOGO FISCALIA

INCLUDE<ABC.INC>;LOGO FISCALIA

END ; Final del programa

ANEXO J

ORG 0X1000
BCF PCLATH,3 ;;;;;;;;;;;;;;;;;;;;;;;;;;TERCERA PAGINA
BSF PCLATH,4
LOGO_1
CALL TIEMPO_PP

CALL TIEMPO_26
CALL TIEMPO_25
CALL TIEMPO_24
CALL TIEMPO_23
CALL TIEMPO_22
CALL TIEMPO_21
CALL TIEMPO_20
CALL TIEMPO_19
CALL TIEMPO_18
CALL TIEMPO_17
CALL TIEMPO_16
CALL TIEMPO_15
CALL TIEMPO_14
CALL TIEMPO_13
CALL TIEMPO_12
CALL TIEMPO_11
CALL TIEMPO_10
CALL TIEMPO_9
CALL TIEMPO_8
CALL TIEMPO_7
CALL TIEMPO_6
CALL TIEMPO_5
CALL TIEMPO_4
CALL TIEMPO_3
CALL TIEMPO_2
CALL TIEMPO_1
CALL TIEMPO_0

CALL TIEMPO_PP
RETURN

;; FILAS
FILAPP

MOVLW b'111';AZUL
MOVWF PORTE

MOVWF PORTD

RETURN

FILA21

CALL AZUL

MOVLW b'11111111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

MOVLW b'11111111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10000000';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA20

CALL AZUL

MOVLW b'11111111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

MOVLW b'11111111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10100000';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA19

CALL AZUL

MOVLW b'11111111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

MOVLW b'11111111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10100000';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA18

CALL AZUL

MOVLW b'11000111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'00111000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

CALL AZUL

MOVLW b'11111000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00000111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10111111';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA17

CALL AZUL

MOVLW b'11100111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'00011000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

MOVLW b'11111000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00000111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10111111';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA15

CALL AZUL

MOVLW b'00000111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'11111000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

CALL AZUL

MOVLW b'11111000';
MOVWF PORTC

```
MOVLW    b'00000000';
MOVWF    PORTC
```

```
CALL AMARILLO
```

```
MOVLW    b'00000111';
MOVWF    PORTC
```

```
MOVLW    b'00000000';
MOVWF    PORTC
```

```
.....
```

```
CALL ROJO
```

```
MOVLW    b'10001011';
MOVWF    PORTD
```

```
MOVLW    b'00000000';
MOVWF    PORTD
```

```
RETURN
```

```
FILA14
```

```
CALL AZUL
```

```
MOVLW    b'00000101';
```

```
MOVWF    PORTB
```

```
MOVLW    b'00000000';
MOVWF    PORTB
```

```
CALL AMARILLO
```

```
MOVLW    b'11111010';
MOVWF    PORTB
```

```
MOVLW    b'00000000';
MOVWF    PORTB
```

```
.....
```

```
CALL AZUL
```

```
MOVLW    b'11101000';
MOVWF    PORTC
```

```
MOVLW    b'00000000';
MOVWF    PORTC
```


MOVLW b'00111111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

;;

CALL ROJO

MOVLW b'10000000';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA12

CALL AZUL

MOVLW b'00000000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'11111111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

;;

CALL AZUL

MOVLW b'11000000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00111111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10110011';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA11

CALL AZUL

MOVLW b'00000101';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'11111010';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

CALL AZUL

MOVLW b'11101000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00010111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

MOVLW b'10111111';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA9

CALL AZUL

MOVLW b'11000111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'00111000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

CALL AZUL

MOVLW b'11111000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00000111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10000000';
MOVWF PORTD

RETURN

FILA7

CALL AZUL

MOVLW b'11000111';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

CALL AMARILLO

MOVLW b'00111000';
MOVWF PORTB

MOVLW b'00000000';
MOVWF PORTB

.....

CALL AZUL

MOVLW b'11111000';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

CALL AMARILLO

MOVLW b'00000111';
MOVWF PORTC

MOVLW b'00000000';
MOVWF PORTC

.....

CALL ROJO

MOVLW b'10101001';
MOVWF PORTD

MOVLW b'00000000';
MOVWF PORTD

RETURN

FILA6

CALL AZUL

**MOVLW b'11111111';
MOVWF PORTB**

**MOVLW b'00000000';
MOVWF PORTB**

**MOVLW b'11111111';
MOVWF PORTC**

**MOVLW b'00000000';
MOVWF PORTC**

;;;

CALL ROJO

**MOVLW b'10100110';
MOVWF PORTD**

**MOVLW b'00000000';
MOVWF PORTD**

RETURN

FILA5

CALL AZUL

**MOVLW b'11111111';
MOVWF PORTB**

**MOVLW b'00000000';
MOVWF PORTB**

**MOVLW b'11111111';
MOVWF PORTC**

**MOVLW b'00000000';
MOVWF PORTC**

;;;

CALL ROJO

**MOVLW b'10000000';
MOVWF PORTD**

MOVLW b'00000000';

```
MOVWF PORTD
RETURN
```

FILA4

```
CALL AZUL
```

```
MOVLW b'11111111';
MOVWF PORTB
```

```
MOVLW b'00000000';
MOVWF PORTB
```

```
MOVLW b'11111111';
MOVWF PORTC
```

```
MOVLW b'00000000';
MOVWF PORTC
```

```
.....
;.....
```

```
CALL ROJO
```

```
MOVLW b'10111111';
MOVWF PORTD
```

```
MOVLW b'00000000';
MOVWF PORTD
```

```
RETURN
```

FILA3

```
CALL AZUL
```

```
MOVLW b'11111111';
MOVWF PORTB
```

```
MOVLW b'00000000';
MOVWF PORTB
```

```
MOVLW b'11111111';
MOVWF PORTC
```

```
MOVLW b'00000000';
MOVWF PORTC
```

```
.....
;.....
```

```
CALL ROJO
```


NIVEL_0.0.0

```
CALL FILA0
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_0.0.0 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_0.0 ; Aun el contador 1 no es cero.
RETURN
```

TIEMPO_1

```
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1
```

NIVEL_1.1

```
MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2
```

NIVEL_1.1.1

```
CALL FILA1
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_1.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_1.1 ; Aun el contador 1 no es cero.
RETURN
```

TIEMPO_2

```
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1
```

NIVEL_2.2

```
MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2
```

NIVEL_2.2.2

```
CALL FILA2
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_2.2.2 ; Aun el contador 2 no es cero.
```

DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_2.2 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_3
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_3.3
MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_3.3.3
CALL FILA3
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_3.3.3 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_3.3 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_4
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_4.4
MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_4.4.4
CALL FILA4
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_4.4.4 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_4.4 ; Aun el contador 1 no es cero.

RETURN

TIEMPO_5
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_5.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_5.1.1

**CALL FILA5
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.**

**GOTO NIVEL_5.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.**

**GOTO NIVEL_5.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_6

**MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_6.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_6.1.1

**CALL FILA6
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.**

**GOTO NIVEL_6.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.**

**GOTO NIVEL_6.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_7

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_7.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_7.1.1

CALL FILA7
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_7.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_7.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_8
MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_8.1

MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_8.1.1

CALL FILA8
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_8.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_8.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_9
MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_9.1

MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_9.1.1

CALL FILA9
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_9.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_9.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_10

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_10.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_10.1.1

**CALL FILA10
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_10.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_10.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_11

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_11.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_11.1.1

**CALL FILA11
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_11.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_11.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_12

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_12.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_12.1.1

**CALL FILA12
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_12.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_12.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_13

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_13.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_13.1.1

**CALL FILA13
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_13.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_13.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_14

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_14.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_14.1.1

CALL FILA14
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cer0.
GOTO NIVEL_14.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cer0.
GOTO NIVEL_14.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_15
MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_15.1

MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_15.1.1

CALL FILA15
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cer0.
GOTO NIVEL_15.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cer0.
GOTO NIVEL_15.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_16
MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_16.1

MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_16.1.1

CALL FILA16
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cer0.
GOTO NIVEL_16.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cer0.
GOTO NIVEL_16.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_17

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_17.1

**MOVLW 0X07; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_17.1.1

**CALL FILA17
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_17.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_17.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_18

**MOVLW 0X07; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_18.1

**MOVLW 0X07 ; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_18.1.1

**CALL FILA18
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_18.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_18.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_19

**MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_19.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_19.1.1

**CALL FILA19
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_19.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_19.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_20

**MOVLW 0X08 ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_20.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_20.1.1

**CALL FILA20
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_20.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_20.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_21

**MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_21.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_21.1.1

CALL FILA21
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_21.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_21.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_22
MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_22.1

MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_22.1.1

CALL FILA22
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_22.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.
GOTO NIVEL_22.1 ; Aun el contador 1 no es cero.
RETURN

TIEMPO_23
MOVLW 0X08 ; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1

NIVEL_23.1

MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2

NIVEL_23.1.1

CALL FILA23
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.
GOTO NIVEL_23.1.1 ; Aun el contador 2 no es cero.

**DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.**

**GOTO NIVEL_23.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_24

**MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_24.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_24.1.1

**CALL FILA24
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.**

**GOTO NIVEL_24.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.**

**GOTO NIVEL_24.1 ; Aun el contador 1 no es cero.
RETURN**

TIEMPO_25

**MOVLW 0X08; Valor con que se carga el contador 1.
CONTADORES DE GROSOR DE LINEA
MOVWF CONTA1**

NIVEL_25.1

**MOVLW 0X08; Valor con que se carga el contador 2.
MOVWF CONTA2**

NIVEL_25.1.1

**CALL FILA25
DECFSZ CONTA2,1 ; Se resta en 1 el contador 2 y se pregunta si es
cero.**

**GOTO NIVEL_25.1.1 ; Aun el contador 2 no es cero.
DECFSZ CONTA1,1 ; Se resta en 1 el contador 1 y se pregunta si es
cero.**

**GOTO NIVEL_25.1 ; Aun el contador 1 no es cero.
RETURN**

MOVLW b'101';AZUL
MOVWF PORTE
RETURN

ROJO

MOVLW b'110';ROJO
MOVWF PORTE
RETURN

VERDE

MOVLW B'011';VERDE
MOVWF PORTE
RETURN

AMARILLO

MOVLW B'010';;;;;;;;AMARILLO
MOVWF PORTE
RETURN

CIAN

MOVLW B'001';CIAN
MOVWF PORTE
RETURN

MAGENTA

MOVLW B'100';ROZADO
MOVWF PORTE
RETURN

BLANCO

MOVLW B'000';BLANCO
MOVWF PORTE
RETURN

.....
.....
.....

ANEXO K

ORG 0X800

BSF PCLATH,3;.....;SEGUNDA PAGINA
BCF PCLATH,4

.....
MENSAJE_1

CALL ESPERA
CALL ESPERA
CALL AZUL_MONO
CALL ESPACIOP_2
CALL LETRA_B_2
CALL ESPACIOP_2
CALL LETRA_S_2
CALL ESPACIOP_2
CALL LETRA_U_2
CALL ESPACIOP_2
CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_2

CALL ESPERA
CALL ESPERA
CALL AZUL_MONO
CALL ESPACIOP_3
CALL LETRA_B_3
CALL ESPACIOP_3

CALL ROJO_MONO
CALL ESPACIOP_2
CALL LETRA_S_2
CALL ESPACIOP_2

CALL VERDE_MONO
CALL ESPACIOP_1
CALL LETRA_U_1
CALL ESPACIOP_1

CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_3

CALL ESPERA
CALL ESPERA
CALL ROJO_MONO
CALL ESPACIOP_1
CALL LETRA_A_1
CALL ESPACIOP_1
CALL LETRA_L_1
CALL ESPACIOP_1
CALL LETRA_O_1
CALL ESPACIOP_1
CALL LETRA_H_1
CALL ESPACIOP_1

CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_4

CALL ESPERA
CALL ESPERA
CALL VERDE_MONO
CALL ESPACIOP_2
CALL LETRA_A_2
CALL ESPACIOP_2
CALL LETRA_L_2
CALL ESPACIOP_2
CALL LETRA_O_2
CALL ESPACIOP_2
CALL LETRA_H_2
CALL ESPACIOP_2

CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_5

CALL ESPERA
CALL ESPERA
CALL AZUL_MONO
CALL ESPACIOP_3
CALL LETRA_A_3
CALL ESPACIOP_3
CALL LETRA_L_3
CALL ESPACIOP_3
CALL LETRA_O_3
CALL ESPACIOP_3
CALL LETRA_H_3
CALL ESPACIOP_3

CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_COLOMBIA

CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL AZUL_MONO
CALL ESPACIOP_2
CALL LETRA_A_2
CALL ESPACIOP_2
CALL LETRA_I_2
CALL ESPACIOP_2
CALL LETRA_B_2
CALL ESPACIOP_2
CALL LETRA_M_2
CALL ESPACIOP_2
CALL LETRA_O_2
CALL ESPACIOP_2
CALL LETRA_L_2
CALL ESPACIOP_2
CALL LETRA_O_2
CALL ESPACIOP_2
CALL LETRA_C_2
CALL ESPACIOP_2
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
RETURN

MENSAJE_JENASH

CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA

CALL VERDE_MONO
CALL ESPACIOP_2
CALL LETRA_H_2
CALL ESPACIOP_2

CALL LETRA_S_2
CALL ESPACIOP_2
CALL LETRA_A_2
CALL ESPACIOP_2
CALL ESPACIOP_2
CALL LETRA_N_2
CALL ESPACIOP_2
CALL LETRA_E_2
CALL ESPACIOP_2
CALL LETRA_J_2
CALL ESPACIOP_2

CALL ESPERA
CALL ESPERA
CALL ESPERA
CALL ESPERA
RETURN

NUMBERS_0123

CALL ESPERA
CALL ESPERA

CALL ROJO_MONO

CALL ESPACIOP_2
CALL NUMBER_3
CALL ESPACIOP_2
CALL NUMBER_2
CALL ESPACIOP_2
CALL NUMBER_1
CALL ESPACIOP_2
CALL NUMBER_0
CALL ESPACIOP_2

CALL ESPERA
CALL ESPERA

RETURN

NUMBERS_456

CALL ESPERA
CALL ESPERA

CALL VERDE_MONO

CALL ESPACIOP_2
CALL NUMBER_6
CALL ESPACIOP_2
CALL NUMBER_5
CALL ESPACIOP_2
CALL NUMBER_4
CALL ESPACIOP_2

CALL ESPERA
CALL ESPERA

RETURN

NUMBERS_789

CALL ESPERA
CALL ESPERA

CALL AZUL_MONO

CALL ESPACIOP_2
CALL NUMBER_9
CALL ESPACIOP_2
CALL NUMBER_8
CALL ESPACIOP_2
CALL NUMBER_7
CALL ESPACIOP_2

CALL ESPERA
CALL ESPERA

RETURN

```

.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
ESPACIOP_1
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;,,,,,,,,;
ESPACIOP_2
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;,,,,,,,,;
ESPACIOP_3
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;,,,,,,,,;
ESPACIOG_1
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,;
ESPACIOG_2
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,;
ESPACIOG_3
    MOVLW  0X00;,,,,,,,,,,,,,,,,,,,,;1
    CALL CAMBIO3
    CALL TIEMPO

```

```

    MOVLW    0X00;.....2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X00;.....3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X00;.....4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....
PUNTO
    MOVLW    0XF0;.....1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XF0;.....2
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....
LETRA_A_1
    MOVLW    0XFE;.....1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X21;.....2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X21;.....3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;.....4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....
LETRA_A_2
    MOVLW    0XFE;.....1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X21;.....2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X21;.....3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....
LETRA_A_3
    MOVLW    0XFE;.....1
    CALL CAMBIO3
    CALL TIEMPO

```

```

    MOVLW    0X21;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X21;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_B_1
    MOVLW    0X66;.....;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X99;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X99;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_B_2
    MOVLW    0X66;.....;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X99;.....;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X99;.....;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_B_3
    MOVLW    0X66;.....;1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X99;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X99;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO3

```

```

CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_C_1
MOVLW  0X42;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X7E;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_C_2
MOVLW  0X42;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X7E;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_C_3
MOVLW  0X42;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X7E;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_D_1
MOVLW  0X32;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X42;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO1

```

```

CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_D_2
MOVLW  0X32;,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X42;,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_D_3
MOVLW  0X32;,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X42;,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X81;,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_E_1
MOVLW  0X91;,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X91;,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X91;,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,

```

```

LETRA_E_2
  MOVLW  0X91;.....;1
  CALL CAMBIO2
  CALL TIEMPO
  MOVLW  0X91;.....;2
  CALL CAMBIO2
  CALL TIEMPO
  MOVLW  0X91;.....;3
  CALL CAMBIO2
  CALL TIEMPO
  MOVLW  0XFE;.....;4
  CALL CAMBIO2
  CALL TIEMPO
  RETURN;.....;

```

```

LETRA_E_3
  MOVLW  0X91;.....;1
  CALL CAMBIO3
  CALL TIEMPO
  MOVLW  0X91;.....;2
  CALL CAMBIO3
  CALL TIEMPO
  MOVLW  0X91;.....;3
  CALL CAMBIO3
  CALL TIEMPO
  MOVLW  0XFE;.....;4
  CALL CAMBIO3
  CALL TIEMPO
  RETURN;.....;

```

```

LETRA_F_1
  MOVLW  0X11;.....;1
  CALL CAMBIO1
  CALL TIEMPO
  MOVLW  0X11;.....;2
  CALL CAMBIO1
  CALL TIEMPO
  MOVLW  0X11;.....;3
  CALL CAMBIO1
  CALL TIEMPO
  MOVLW  0XFF;.....;4
  CALL CAMBIO1
  CALL TIEMPO
  RETURN;.....;

```

```

LETRA_F_2
  MOVLW  0X11;.....;1
  CALL CAMBIO2
  CALL TIEMPO
  MOVLW  0X11;.....;2
  CALL CAMBIO2
  CALL TIEMPO
  MOVLW  0X11;.....;3

```

```

CALL CAMBIO2
CALL TIEMPO
MOVLW  0xFF;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_F_3
MOVLW  0x11;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0x11;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0x11;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0xFF;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_G_1
MOVLW  0x72;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0x81;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0x81;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0x7E;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_G_2
MOVLW  0x72;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0x81;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0x81;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0x7E;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_G_3
MOVLW  0x72;.....;1

```

```

CALL CAMBIO3
CALL TIEMPO
MOVLW  0X81;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X81;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X7E;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_H_1
MOVLW  0XFE;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X10;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X10;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_H_2
MOVLW  0XFE;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X10;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X10;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_H_3
MOVLW  0XFE;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X10;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X10;.....;3
CALL CAMBIO3
CALL TIEMPO

```

```

    MOVLW    0XFE;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_I_1
    MOVLW    0X81;.....;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_I_2
    MOVLW    0X81;.....;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X81;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_I_3
    MOVLW    0X81;.....;1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X81;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_J_1
    MOVLW    0X01;.....;1
    CALL CAMBIO1
    CALL TIEMPO

```

```

    MOVLW    0XFE;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X41;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_J_2
    MOVLW    0X01;.....;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X81;.....;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X41;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_J_3
    MOVLW    0X01;.....;1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X81;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X41;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_K_1
    MOVLW    0X83;.....;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X2C;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X10;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO1

```

```

CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_K_2
MOVLW 0X83;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW 0X2C;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW 0X10;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW 0XFE;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_K_3
MOVLW 0X83;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW 0X2C;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW 0X10;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW 0XFE;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_L_1
MOVLW 0X80;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW 0X80;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW 0X80;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW 0XFE;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_L_2
MOVLW 0X80;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW 0X80;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO2

```

```

CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_L_3
MOVLW  0X80;,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFE;,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_M_1
MOVLW  0XFF;,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X02;,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X04;,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X02;,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFF;,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_M_2
MOVLW  0XFF;,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X02;,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X04;,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X02;,,,,,,,,,,,,,4

```

```

CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFF;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_M_3
MOVLW  0XFF;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X02;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X04;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X02;.....;4
CALL CAMBIO3
MOVLW  0XFF;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_N_1
MOVLW  0XFE;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X18;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X06;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_N_2
MOVLW  0XFE;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X18;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X06;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;

```

```

LETRA_N_3
    MOVLW    0XFE;,,,,,,,,,,,,,1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X18;,,,,,,,,,,,,,2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X06;,,,,,,,,,,,,,3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;,,,,,,,,,,,,,4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_O_1
    MOVLW    0X7E;,,,,,,,,,,,,,1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X7E;,,,,,,,,,,,,,4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_O_2
    MOVLW    0X7E;,,,,,,,,,,,,,1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X7E;,,,,,,,,,,,,,4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_O_3
    MOVLW    0X7E;,,,,,,,,,,,,,1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X81;,,,,,,,,,,,,,3

```

```

CALL CAMBIO3
CALL TIEMPO
MOVLW  0X7E;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_P_1
MOVLW  0X6;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X9;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X9;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_P_2
MOVLW  0X6;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X9;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X9;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_P_3
MOVLW  0X6;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X9;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X9;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_Q_1
MOVLW  0XBE;.....;1

```

```

CALL CAMBIO1
CALL TIEMPO
MOVLW  0XA1;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XA1;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_Q_2
MOVLW  0XBE;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XA1;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XA1;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_Q_3
MOVLW  0XBE;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XA1;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XA1;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFE;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_R_1
MOVLW  0XC6;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X39;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X09;.....;3
CALL CAMBIO1
CALL TIEMPO

```

```

    MOVLW    0XFE;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_R_2
    MOVLW    0XC6;.....;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X39;.....;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X09;.....;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_R_3
    MOVLW    0XC6;.....;1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X39;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X09;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_S_1
    MOVLW    0X72;.....;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X89;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X89;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X46;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_S_2
    MOVLW    0X72;.....;1
    CALL CAMBIO2
    CALL TIEMPO

```

```

    MOVLW    0X89;.....2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X89;.....3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X46;.....4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....
LETRA_S_3
    MOVLW    0X72;.....1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X89;.....2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X89;.....3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X46;.....4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....
LETRA_T_1
    MOVLW    0X01;.....1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFF;.....2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFF;.....3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X01;.....4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....
LETRA_T_2
    MOVLW    0X01;.....1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFF;.....2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFF;.....3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X01;.....4
    CALL CAMBIO2

```

```

CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_T_3
MOVLW  0X01;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFF;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XFF;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X01;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_U_1
MOVLW  0X7F;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X7F;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_U_2
MOVLW  0X7F;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X7F;,,,,,,,,,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
LETRA_U_3
MOVLW  0X7F;,,,,,,,,,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,,,,,,,,,2
CALL CAMBIO3

```

```

CALL TIEMPO
MOVLW  0X80;,,,,,,,,,,,,,;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X7F;,,,,,,,,,,,,,;4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,;
LETRA_V_1
MOVLW  0X3F;,,,,,,,,,,,,,;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X3F;,,,,,,,,,,,,,;4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,;
LETRA_V_2
MOVLW  0X3F;,,,,,,,,,,,,,;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X3F;,,,,,,,,,,,,,;4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,;
LETRA_V_3
MOVLW  0X3F;,,,,,,,,,,,,,;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XC0;,,,,,,,,,,,,,;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X3F;,,,,,,,,,,,,,;4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,;

```

```

LETRA_W_1
    MOVLW    0XFE;,,,,,,,,,,,,,1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XFE;,,,,,,,,,,,,,4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_W_2
    MOVLW    0XFE;,,,,,,,,,,,,,1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XFE;,,,,,,,,,,,,,4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_W_3
    MOVLW    0XFE;,,,,,,,,,,,,,1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X40;,,,,,,,,,,,,,3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XFE;,,,,,,,,,,,,,4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;,,,,,,,,,,,,,

```

```

LETRA_X_1
    MOVLW    0XE7;,,,,,,,,,,,,,1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X18;,,,,,,,,,,,,,2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0X18;,,,,,,,,,,,,,3

```

```

CALL CAMBIO1
CALL TIEMPO
MOVLW  0XE7;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_X_2
MOVLW  0XE7;.....;1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X18;.....;2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X18;.....;3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XE7;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_X_3
MOVLW  0XE7;.....;1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X18;.....;2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X18;.....;3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XE7;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_Y_1
MOVLW  0X0F;.....;1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XF0;.....;2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XF0;.....;3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X0F;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_Y_2
MOVLW  0X0F;.....;1

```

```

CALL CAMBIO2
CALL TIEMPO
MOVLW  0XF0;,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XF0;,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X0F;,,,,,,,,,,,,,4
CALL CAMBIO2
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_Y_3
MOVLW  0X0F;,,,,,,,,,,,,,1
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XF0;,,,,,,,,,,,,,2
CALL CAMBIO3
CALL TIEMPO
MOVLW  0XF0;,,,,,,,,,,,,,3
CALL CAMBIO3
CALL TIEMPO
MOVLW  0X0F;,,,,,,,,,,,,,4
CALL CAMBIO3
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_Z_1
MOVLW  0X83;,,,,,,,,,,,,,1
CALL CAMBIO1
CALL TIEMPO
MOVLW  0X85;,,,,,,,,,,,,,2
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XB9;,,,,,,,,,,,,,3
CALL CAMBIO1
CALL TIEMPO
MOVLW  0XC1;,,,,,,,,,,,,,4
CALL CAMBIO1
CALL TIEMPO
RETURN;,,,,,,,,,,,,,
LETRA_Z_2
MOVLW  0X83;,,,,,,,,,,,,,1
CALL CAMBIO2
CALL TIEMPO
MOVLW  0X85;,,,,,,,,,,,,,2
CALL CAMBIO2
CALL TIEMPO
MOVLW  0XB9;,,,,,,,,,,,,,3
CALL CAMBIO2
CALL TIEMPO

```

```

    MOVLW    0XC1;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_Z_3
    MOVLW    0X83;.....;1
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0X85;.....;2
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XB9;.....;3
    CALL CAMBIO3
    CALL TIEMPO
    MOVLW    0XC1;.....;4
    CALL CAMBIO3
    CALL TIEMPO
    RETURN;.....;
LETRA_ADMUP_1
    MOVLW    0XC7;.....;1
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XC7;.....;2
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XC7;.....;3
    CALL CAMBIO1
    CALL TIEMPO
    MOVLW    0XC7;.....;4
    CALL CAMBIO1
    CALL TIEMPO
    RETURN;.....;
LETRA_ADMUP_2
    MOVLW    0XC7;.....;1
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XC7;.....;2
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XC7;.....;3
    CALL CAMBIO2
    CALL TIEMPO
    MOVLW    0XC7;.....;4
    CALL CAMBIO2
    CALL TIEMPO
    RETURN;.....;
LETRA_ADMUP_3
    MOVLW    0XC7;.....;1
    CALL CAMBIO3
    CALL TIEMPO

```

```

MOV LW    0XC7;.....;2
CALL CAMBIO3
CALL TIEMPO
MOV LW    0XC7;.....;3
CALL CAMBIO3
CALL TIEMPO
MOV LW    0XC7;.....;4
CALL CAMBIO3
CALL TIEMPO
RETURN;.....;
LETRA_ADMDOWN_1
MOV LW    0XF3;.....;1
CALL CAMBIO1
CALL TIEMPO
MOV LW    0XF3;.....;2
CALL CAMBIO1
CALL TIEMPO
MOV LW    0XF3;.....;3
CALL CAMBIO1
CALL TIEMPO
MOV LW    0XF3;.....;4
CALL CAMBIO1
CALL TIEMPO
RETURN;.....;
LETRA_ADMDOWN_2
MOV LW    0XF3;.....;1
CALL CAMBIO2
CALL TIEMPO
MOV LW    0XF3;.....;2
CALL CAMBIO2
CALL TIEMPO
MOV LW    0XF3;.....;3
CALL CAMBIO2
CALL TIEMPO
MOV LW    0XF3;.....;4
CALL CAMBIO2
CALL TIEMPO
RETURN;.....;
LETRA_ADMDOWN_3
MOV LW    0XF3;.....;1
CALL CAMBIO3
CALL TIEMPO
MOV LW    0XF3;.....;2
CALL CAMBIO3
CALL TIEMPO
MOV LW    0XF3;.....;3
CALL CAMBIO3
CALL TIEMPO
MOV LW    0XF3;.....;4
CALL CAMBIO3

```

```
CALL TIEMPO
RETURN;.....
.....
;.....;
```

NUMBER_0

```
MOVLW 0X7E;.....;1
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X81;.....;2
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X81;.....;3
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X7E;.....;4
MOVWF PORTC
CALL TIEMPO_NUMBER
RETURN;.....;
```

NUMBER_1

```
MOVLW 0X80;.....;4
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0XFF;.....;3
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X82;.....;2
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X04;.....;1
MOVWF PORTC
CALL TIEMPO_NUMBER
RETURN;.....;
```

NUMBER_2

```
MOVLW 0X8E;.....;4
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0X91;.....;3
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0XA2;.....;2
MOVWF PORTC
CALL TIEMPO_NUMBER
MOVLW 0XC2;.....;1
MOVWF PORTC
CALL TIEMPO_NUMBER
RETURN;.....;
```

```

NUMBER_3
    MOVLW    0X24;,,,,,,,,,,,,,,,,,,,,;4
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X99;,,,,,,,,,,,,,,,,,,,,;3
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X99;,,,,,,,,,,,,,,,,,,,,;2
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X66;,,,,,,,,,,,,,,,,,,,,;1
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    RETURN;,,,,,,,,,,,,,,,,,,,,;

```

```

NUMBER_4

    MOVLW    0XFF;,,,,,,,,,,,,,,,,,,,,;4
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X12;,,,,,,,,,,,,,,,,,,,,;3
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X14;,,,,,,,,,,,,,,,,,,,,;2
    MOVWF    PORTC
    CALL TIEMPO_NUMBER

    MOVLW    0X18;,,,,,,,,,,,,,,,,,,,,;1
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    RETURN;,,,,,,,,,,,,,,,,,,,,;

```

```

NUMBER_5
    MOVLW    0X71;,,,,,,,,,,,,,,,,,,,,;4
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X89;,,,,,,,,,,,,,,,,,,,,;3
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X89;,,,,,,,,,,,,,,,,,,,,;2
    MOVWF    PORTC
    CALL TIEMPO_NUMBER
    MOVLW    0X4E;,,,,,,,,,,,,,,,,,,,,;1
    CALL PORTC
    CALL TIEMPO_NUMBER
    RETURN;,,,,,,,,,,,,,,,,,,,,;

```

```

NUMBER_6

```

```

MOVLW    0X70;4
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X89;3
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X89;2
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X7E;1
CALL PORTC
CALL TIEMPO_NUMBER
RETURN;

```

NUMBER_7

```

MOVLW    0X0F;4
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X11;3
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0XE1;2
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X03;1
MOVWF    PORTC
CALL TIEMPO_NUMBER
RETURN;

```

NUMBER_8

```

MOVLW    0X6E;4
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X89;3
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X89;2
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X6E;1
MOVWF    PORTC
CALL TIEMPO_NUMBER
RETURN;

```

NUMBER_9

```

MOVLW    0X1E;4
MOVWF    PORTC
CALL TIEMPO_NUMBER

```

```

MOVLW    0X69;.....;3
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X89;.....;2
MOVWF    PORTC
CALL TIEMPO_NUMBER
MOVLW    0X06;.....;1
MOVWF    PORTC
CALL TIEMPO_NUMBER
RETURN;.....;

```

```

NUMBER_ESPACIOP_2
MOVLW    0X00;.....;1
MOVWF    PORTC
CALL TIEMPO_NUMBER
RETURN;.....;

```

```

TIEMPO_NUMBER
MOVLW 0X17
MOVWF MONO1
T1X
MOVLW 0X17
MOVWF MONO2
T2X
DECFSZ MONO2,1
GOTO T2X
DECFSZ MONO1,1
GOTO T1X
RETURN

```

```

TIEMPO_ESPERA
MOVLW 0X7F;...;7F
MOVWF MONO4
PP1
MOVLW 0X7F
MOVWF MONO3
PP2
DECFSZ MONO3,1
GOTO PP2
DECFSZ MONO4,1
GOTO PP1
RETURN

```

```

.....
;.....;
.....
AZUL_MONO
MOVLW    b'101';AZUL
MOVWF    PORTE

```

```

    RETURN
ROJO_MONO
    MOVLW    b'110';ROJO
    MOVWF    PORTE
    RETURN
VERDE_MONO
    MOVLW    B'011';VERDE
    MOVWF    PORTE
    RETURN
AMARILLO_MONO
    MOVLW    B'010';AMARILLO
    MOVWF    PORTE
    RETURN
CIAN_MONO
    MOVLW    B'001';CIAN
    MOVWF    PORTE
    RETURN
MAGENTA_MONO
    MOVLW    B'100';ROZADO
    MOVWF    PORTE
    RETURN
BLANCO_MONO
    MOVLW    B'000';BLANCO
    MOVWF    PORTE
    RETURN
LIMPIAR
    MOVLW    b'111';ROJO
    MOVWF    PORTE
    RETURN

```

```

TIEMPO
    MOVLW 0X17
    MOVWF MONO1
L1
    MOVLW 0X17
    MOVWF MONO2
L2
    DECFSZ MONO2,1
    GOTO L2
    DECFSZ MONO1,1
    GOTO L1
    RETURN

```

```

ESPERA
    MOVLW 0X9F;7F
    MOVWF MONO4
X1
    MOVLW 0X9F
    MOVWF MONO3

```

X2

```
DECFSZ MONO3,1  
GOTO X2  
DECFSZ MONO4,1  
GOTO X1  
RETURN
```

CAMBIO1

```
MOVWF PORTB  
RETURN
```

CAMBIO2

```
MOVWF PORTC  
RETURN
```

CAMBIO3

```
MOVWF PORTD  
RETURN
```