

**UNIVERSIDAD POLITECNICA DE VALENCIA**

**ESCUELA POLITECNICA SUPERIOR DE GANDIA**

**I.T. Telecomunicación (Sist. Electrónicos)**

---



**UNIVERSIDAD  
POLITECNICA  
DE VALENCIA**



**ESCUELA POLITECNICA  
SUPERIOR DE GANDIA**

# **“Diseño y desarrollo de un sistema POV (Persistence Of Vision) mediante PIC”**

***TRABAJO FINAL DE CARRERA***

Autor/es:

**José Vicente Plaza Herrero**

Director/es:

**D. Tomás Carlos Sogorb Devesa**

***GANDIA, 2013***



---

# ÍNDICE

---

<u>CONCEPTO</u>	<u>PÁGINA</u>
<b>1. INTRODUCCIÓN</b> .....	1
1.1. Estado del arte .....	1
1.2. Necesidades actuales .....	3
1.3. Motivación y objetivos del proyecto.....	4
<b>2. FUNDAMENTOS TEÓRICOS</b> .....	5
2.1. Arquitectura del ojo humano .....	5
2.2. Persistence of Vision (POV) o persistencia retiniana .....	7
2.3. Microcontroladores .....	16
2.3.1. Introducción a los microcontroladores .....	16
2.3.2. Repaso histórico.....	17
2.3.3. Características y aplicaciones.....	18
2.3.4. Arquitecturas .....	20
2.3.5. Diferencia entre microprocesador y microcontrolador .....	21
2.3.6. Microcontrolador PIC.....	22
2.3.6.1. Juego de instrucciones .....	23
2.3.6.2. Arquitectura del PIC .....	24
2.3.6.3. Programación del PIC .....	25
2.3.6.4. Características .....	27
2.3.6.5. Variaciones del PIC.....	27
2.3.6.6. PICs más usados .....	28
2.3.7. Familia PIC18 de Microchip.....	29
2.3.7.1. PIC 18F4680 .....	35
<b>3. DISEÑO DEL CIRCUITO</b> .....	41
3.1. Especificaciones .....	41
3.2. Diseño de la placa .....	41
3.2.1. Esquemático .....	41
3.2.1.1. Efecto Hall.....	43
3.2.1.2. Regulador de tensión.....	45
3.2.1.3. Diodos LED RGB .....	46
3.3. PCB .....	49
3.3.1. Placa principal.....	51
3.3.2. Placa LEDs .....	54
3.4. Realización del circuito impreso .....	55
<b>4. DESARROLLO Y MONTAJE</b> .....	57
4.1. Placa rotatoria.....	57
4.2. Montaje de la parte de potencia. Motor. ....	60
<b>5. PROGRAMACIÓN</b> .....	66
5.1. Conversión de imagen.....	66
5.2. Compilador y código fuente.....	67
5.3. Proceso de programación del PIC .....	71
<b>6. RESULTADO FINAL</b> .....	75

<b>7. PRESUPUESTO</b> .....	78
<b>8. CONCLUSIÓN</b> .....	79
<b>BIBLIOGRAFÍA</b> .....	80
<b>ÍNDICE DE IMÁGENES</b> .....	82
<b>ANEXO</b> .....	85

*Agradezco a mi tutor Tomás Carlos Sogorb Devesa y a los técnicos de la titulación Francisco José Ibáñez Sabater y Antoni Bou Escrivà la ayuda prestada para la consecución de este Proyecto.*



---

# 1. INTRODUCCIÓN

---

La publicidad es un elemento que, queramos o no, nos rodea. Su función no es otra que atraer las miradas de los potenciales clientes, ya sea con colores llamativos e incluso con pegadizas melodías. Ya son varias las empresas que utilizan el objeto de este Proyecto como, por ejemplo, la marca automovilística Peugeot que hace un par de años lanzó la campaña publicitaria “Persistence of Vision” para pretender explicarnos que sus vehículos se quedaban grabados en nuestros ojos.



*Figura 1.1. Anuncio Peugeot*

Dicho efecto óptico, también conocido como persistencia retiniana, será explicado más ampliamente según se vaya desarrollando esta memoria.

La base de esta idea se remonta a principios del siglo XIX en dónde ya se empezaba a investigar este fenómeno inherente a los ojos humanos. Con la aparición de la televisión y el cine se acentuaron los estudios y se jugó con esta característica para crear movimiento a partir de una sucesión de imágenes estáticas.

## 1.1– Estado del arte

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

Un ejemplo conocido es el **taumátropo**, basado en un disco de cartón que se hace girar sobre su eje mediante cuerdas atadas a sus extremos. En cada cara hay un dibujo distinto y, al girar el disco, las dos imágenes parecen fundirse. Uno de los taumátropos más conocidos fue el del pájaro enjaulado.

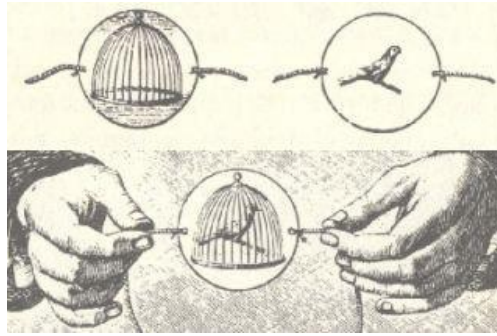


Figura 1.2. Taumátropo

Otro ejemplo lo podemos encontrar en el **zoótrofo** o deadelum, un juguete muy popular que fue uno de los avances más importantes en la primera mitad del siglo XIX hacia la aparición del cine.

Creado por William George Horner en 1834, consta de un cilindro o tambor de madera con cortes a través de los cuales el espectador ve una cinta con dibujos que debido al movimiento giratorio del cilindro y el fenómeno de persistencia retiniana dan la impresión de movimiento.



Figura 1.3. Zootropo

Posee un funcionamiento similar al fenaquistiscopio, del griego “espectador ilusorio” de Joseph-Antoine Ferdinand Plateau, con la ventaja de que no es necesario acudir a un espejo para visualizar la animación.



Figura 1.4. Fenaquistiscopio



Más adelante se estudiará el porqué de este fenómeno visual.

## 1.2– Necesidades actuales

Hoy en día la publicidad domina nuestro mercado empresarial y por este motivo las corporaciones se apoyan en reclamos publicitarios de toda clase para atraer a los clientes. Debido a esto, se utilizan anuncios en todos los medios de comunicación posibles, como pueden ser la televisión, periódicos, paneles luminosos, etc...

Un ejemplo claro de esto último se puede encontrar en las paradas de autobuses, como medio informativo de sus trayectos o en las farmacias, en las que se puede distinguir a distancia sus llamativas cruces, como también en los pubs o locales nocturnos en los que se pueden observar logotipos corporativos de alguna marca de licores.



Figura 1.5. Rótulo LED



Figura 1.6. Panel LED

Internacionalmente, podemos encontrar grandes paneles luminosos de publicidad en ciudades importantes como Nueva York o Londres, como también en los gigantes asiáticos (Japón, China, Singapur).

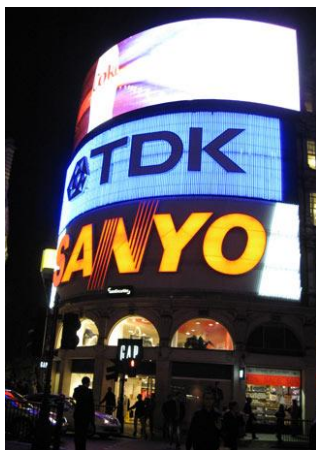


Figura 1.7. Londres



Figura 1.8. Nueva York

Otro concepto que se ha empezado a vislumbrar y que sería el siguiente paso de los paneles de publicidad con LEDs sería el holograma, que consiste en la creación de una imagen tridimensional a partir de haces de luz, como los utilizados en varios conciertos de artistas reconocidos o en diversas películas de ciencia-ficción ambientadas en el futuro.



Figura 1.9. Holograma Michael Jackson



Figura 1.10. Holograma película Avatar

### 1.3– Motivación y objetivos del proyecto

La motivación de este proyecto viene dada por la intención de crear un dispositivo de impacto visual en el que se pueda programar cualquier imagen que se desee, destinado al mundo de la publicidad anteriormente comentado para atraer de una manera distinta a los futuros clientes.

Este proyecto como se deduce por el título consistirá en el desarrollo y diseño de un juego de LEDs, controlados por un PIC o microcontrolador, que al realizar un movimiento circular dejarán como estela una imagen previamente programada. Para su realización, se pretenden usar componentes de bajo coste para demostrar que se pueden desarrollar aplicaciones muy interesantes con los LEDs sin invertir un gran presupuesto.

Para empezar el diseño y desarrollo del proyecto, se han de tener en cuenta unas premisas muy importantes:

- Programar un PIC para que cada cierto tiempo genere un código de colores.
- Conseguir almacenar estos códigos en memoria para la correcta visualización del dibujo persistente.
- Comprobar que el sistema genera una imagen estática para el ojo humano.
- Desarrollar la placa de circuito impreso con todos los componentes necesarios para el correcto funcionamiento del sistema.
- Idear un sistema de alimentación sin cables entre la placa rotatoria y el motor.

---

## 2. FUNDAMENTOS TEÓRICOS

---

En este capítulo se expondrán las bases teóricas del campo de la óptica que atañen a este proyecto. Se hablará de la arquitectura de nuestro ojo y de porqué se produce este efecto persistente en nuestras retinas, a la vez que se ilustrará con ejemplos prácticos para que sea más fácil comprender su funcionamiento.

Por otra parte, también se introducirán las características del “cerebro” de la placa electrónica de este proyecto, el microcontrolador, para poder entender porqué se utiliza este componente y conocer su funcionamiento interno.

### 2.1– Arquitectura del ojo humano

El ojo es un órgano que detecta la luz y es la base del sentido de la vista. Su función consiste básicamente en transformar la energía lumínica en señales eléctricas que son enviadas al cerebro a través del nervio óptico.

El **ojo humano** funciona de forma muy similar al de la mayoría de los vertebrados y algunos moluscos; posee una lente llamada cristalino que es ajustable según la distancia, un diafragma que se llama pupila cuyo diámetro está regulado por el iris y un tejido sensible a la luz que es la retina. La luz penetra a través de la pupila, atraviesa el cristalino y se proyecta sobre la retina, donde se transforma gracias a unas células llamadas fotorreceptoras en impulsos nerviosos que son trasladados a través del nervio óptico al cerebro.

Su forma es aproximadamente esférica, mide 2,5 cm de diámetro y está lleno de un gel transparente llamado humor vítreo que rellena el espacio comprendido entre la retina y el cristalino.

En la porción anterior del ojo se encuentran dos pequeños espacios: la cámara anterior que está situada entre la córnea y el iris, y la cámara posterior que se ubica entre el iris y el cristalino. Estas cámaras están llenas de un líquido que se llama humor acuoso, cuyo nivel de presión llamado presión intraocular es muy importante para el correcto funcionamiento del ojo.

Para que los rayos de luz que penetran en el ojo se puedan enfocar en la retina, se deben refractar. La cantidad de refracción requerida depende de la distancia del objeto al observador. Un objeto distante requerirá menos refracción que uno más cercano. La mayor parte de la refracción ocurre en la córnea, que tiene una curvatura fija. Otra parte de la refracción requerida se da en el cristalino. El cristalino puede cambiar de forma, aumentando o disminuyendo así su capacidad de refracción. Al envejecer, el ser humano va perdiendo esta capacidad de ajustar el enfoque, deficiencia conocida como presbicia o vista cansada.

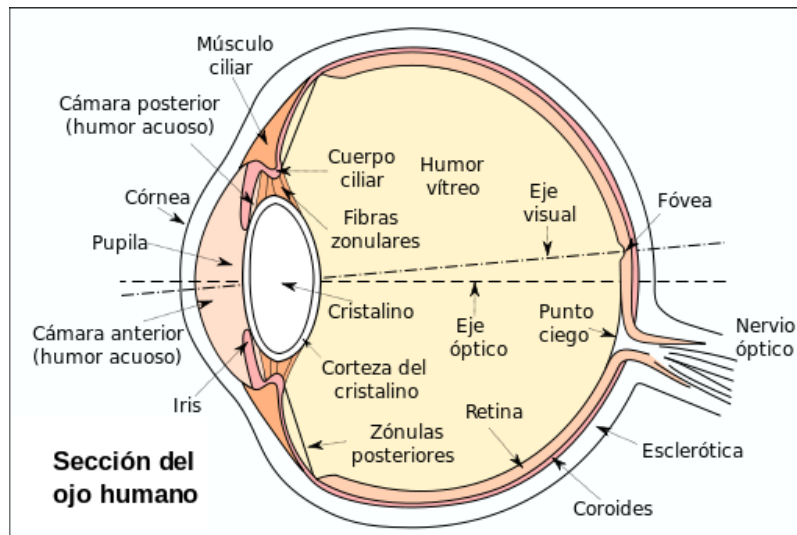


Figura 2.1. Corte transversal de un ojo humano

El componente ocular que nos concierne y el responsable de que se produzca el efecto objeto de este proyecto es la retina, de ahí que se llame también persistencia retiniana.

La **retina** de los vertebrados es un tejido sensible a la luz situado en la superficie interior del ojo. Es similar a una tela donde se proyectan las imágenes. La luz que incide en la retina desencadena una serie de fenómenos químicos y eléctricos que finalmente se traducen en impulsos nerviosos que son enviados hacia el cerebro por el nervio óptico.

La retina tiene una estructura compleja. Está formada básicamente por varias capas de neuronas interconectadas mediante sinapsis. Las únicas células sensibles directamente a la luz son los conos y los bastones. La retina humana contiene 6.5 millones de conos y 120 millones de bastones. Los bastones funcionan principalmente en condiciones de baja luminosidad y proporcionan la visión en blanco y negro, los conos sin embargo están adaptados a las situaciones de mucha luminosidad y proporcionan la visión en color.

En el ojo humano hay tres tipos de conos, sensibles a luz de color azul, rojo y verde respectivamente. Cada uno de ellos absorbe la radiación de una determinada porción del espectro gracias a que poseen unos pigmentos llamados opsinas. Las opsinas son unas moléculas que están formadas por una proteína y un derivado de la vitamina A. La eritropsina tiene mayor sensibilidad para las longitudes de onda largas de alrededor de 560 nm (luz roja), la cloropsina para longitudes de onda medias de unos 530 nm (luz verde) y por último la cianopsina con mayor sensibilidad para las longitudes de onda pequeñas de unos 430 nm (luz azul). Mediante las diferentes intensidades de las señales producidas por los 3 tipos de conos, podemos distinguir todos los colores que forman el espectro de luz visible.



Figura 2.2. Espectro de frecuencias ojo humano

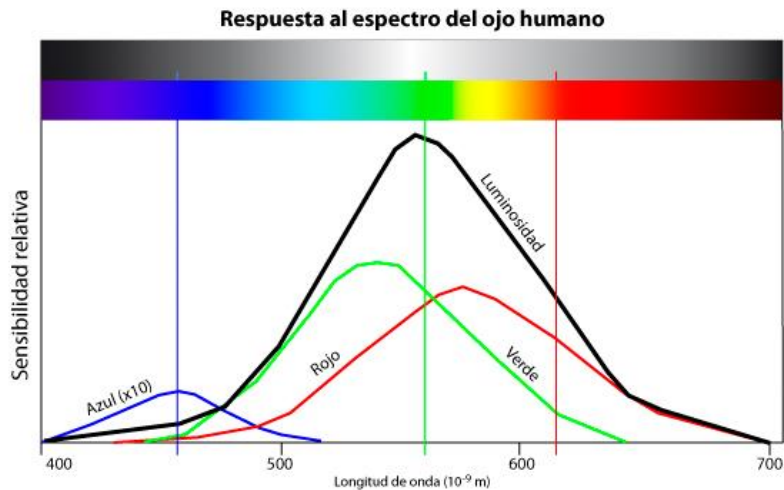


Figura 2.3. Respuesta del ojo al espectro

En la figura se puede observar como el verde es el color que más distinguimos, en detrimento del azul y el rojo, cuya combinación crea todos los colores posibles. En este proyecto se han utilizado LEDs RGB (Red Green Blue) que nos dan toda una gama de colores mezclando en diferentes tonalidades el rojo, azul y verde.

Los conos están concentrados en el centro de la retina, mientras que los bastones abundan más en la periferia de la misma. Cada cono está conectado individualmente con el centro visual del cerebro, lo que en la práctica permite distinguir a una distancia de 10 metros dos puntos luminosos separados por sólo un milímetro. Cada ojo humano dispone de 7 millones de conos y 125 millones de bastones.

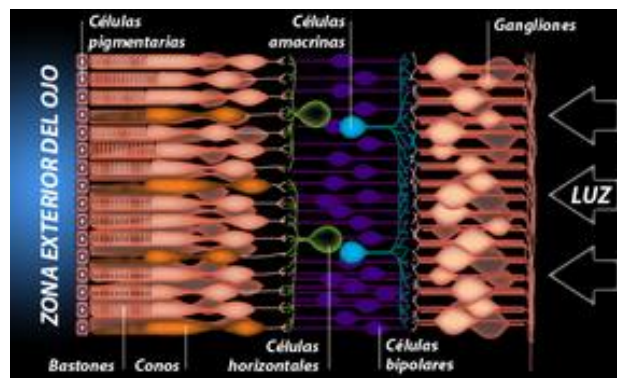


Figura 2.4. Conos y bastones

## 2.2– Persistence Of Vision (POV) o persistencia retiniana

El fenómeno visual que nos ocupa, el Persistence Of Vision (POV) o persistencia retiniana fue descubierto por el científico belga Joseph-Antoine Ferdinand Plateau, que demostraría como una imagen permanece en la retina humana una décima de segundo antes de desaparecer por completo.



Según sus estudios, esto permitiría que veamos la realidad como una secuencia de imágenes ininterrumpidas y que podamos calcular fácilmente la velocidad y dirección de un objeto que se desplaza, si no existiese, veríamos pasar la realidad como sucesión de imágenes independientes y estáticas. Plateau creyó descubrir que nuestro ojo ve con una cadencia de 10 imágenes por segundo. En virtud de dicho fenómeno, las imágenes se superponen en la retina y el cerebro las "enlaza" como una sola imagen visual, móvil y continua. En el cine en blanco y negro, se pasaban 18 imágenes por segundo porque se decía que era el mínimo para poder crear el efecto de movimiento, las películas actuales se proyectan a 24 imágenes por segundo, e incluso a 25 para adaptarse mejor a la frecuencia utilizada en televisión.

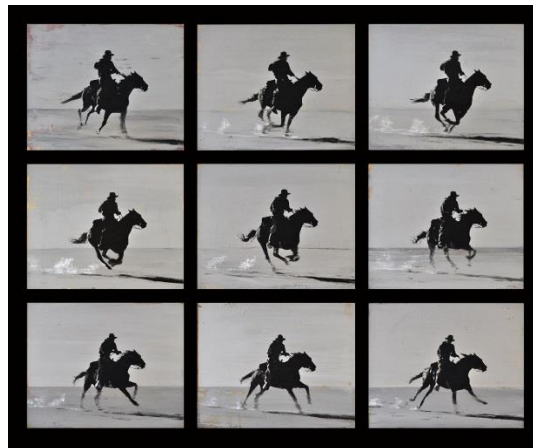


Figura 2.5. Transiciones de imágenes

Dicha característica de nuestro ojo provoca que las imágenes que se observan no se borren instantáneamente. Este hecho, hace que las imágenes que se visualizan, queden guardadas por un instante en el cerebro. Por ejemplo, al hacer girar una cerilla podemos lograr que ésta parezca formar un círculo de fuego en el aire, al igual que si se hace con un LED encendido. Si a dicho LED o a un conjunto de ellos los activamos de manera apropiada en tiempo y forma, obtendremos, gracias a la persistencia de nuestra visión, imágenes que parecen estar volando en el aire.

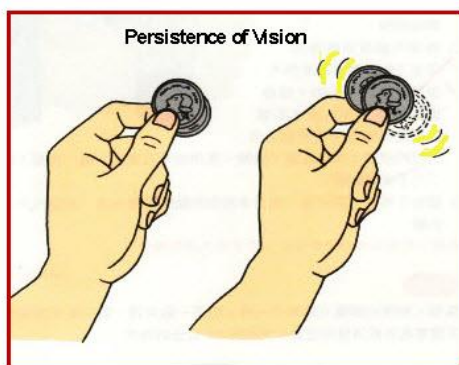


Figura 2.6. Persistencia retiniana



Figura 2.7. Estela LEDs

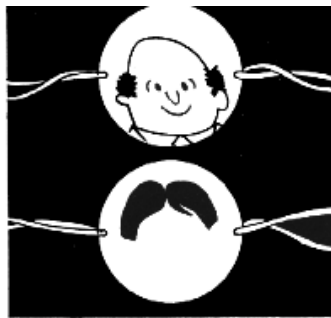
Este fenómeno ya se conocía en la antigüedad, pero entonces no se contaba con las posibilidades que ahora poseemos para realizar películas. A pesar de ello, muchos inventores, aun careciendo de medios técnicos adecuados, se esforzaron en representar imágenes en movimiento.

Fue forzoso empezar con imágenes dibujadas. Algunos de los experimentos realizados en tiempos remotos pueden sernos muy útiles para iniciar los ensayos con que conseguir el efecto del movimiento. En este caso prescindimos totalmente de la cámara y obtenemos resultados muy directos.

## El taumátropo o roboscopio

Hay muchas formas de demostrar la capacidad de retención de nuestros ojos. Uno de los sistemas más conocidos consiste en emplear un sencillo instrumento que hizo historia con el curioso nombre de **taumátropo**.

Con este utensilio, que se puede construir en menos de cinco minutos, se logra que dos imágenes se mezclen y confundan en una sola. Se corta un trozo de cartón en forma circular. En una cara se dibuja la cabeza de un hombre calvo, y en la otra (es el ejemplo clásico) una peluca. Se taladran dos orificios diametralmente opuestos y por cada uno se introduce un hilo retorcido. Al estirar de los hilos el disco gira rápidamente sobre sí mismo, y se ve la peluca cubriendo el cráneo del hombre calvo.



*Figura 2.8. Taumátropo*

## El estroboscopio

Se trata de un reflector provisto de un tubo de observación. De hecho es una variante del flash electrónico, con la diferencia de que este aparato emite una serie de destellos consecutivos. La frecuencia de estos destellos luminosos se ajusta desde uno por segundo hasta trescientos o más. Imaginemos que con ayuda de este tubo miramos un disco que gira y que regulamos la frecuencia de los destellos del flash de manera que se produzca uno por cada vuelta del disco. Para nuestros ojos, el objeto giratorio parece estar totalmente parado. Nos hallaremos, pues, en situación de observar el objeto giratorio como si permaneciese en reposo.

Podemos invertir el proceso. Imaginemos ahora que está dibujada en el disco una serie de fases de un movimiento determinado (es decir: varios dibujos en los que se halla representado el movimiento). Pegamos en el orden correcto esta serie de fases en el plato de un tocadiscos; lo ponemos en marcha y lo iluminamos con el estroboscopio, ajustando la frecuencia de los destellos de manera que se produzca uno cada vez que pase ante nosotros un dibujo y el plato quede sin iluminar durante el espacio que media entre un dibujo y el otro. El resultado está en que se aprecia el movimiento del dibujo. Un tal profesor Stampfer (Viena, 1832) creó el estroboscopio y logró que una serie de dibujos de una muñeca se convirtieran en una muñeca móvil. De hecho, fue el primer hombre que realizó una película de dibujos.

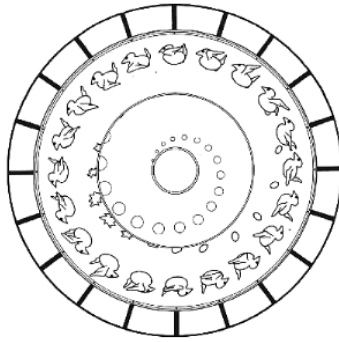


Figura 2.9. Estroboscopio

En la figura está representado el disco de un estroboscopio, el cual sirve para comprender mejor su funcionamiento. En primer lugar se necesita disponer de un dibujo igual o de reproducir éste. Se puede hacer una foto y ampliarla, o bien copiar el dibujo, con mucha exactitud, en el tamaño que desee. Una vez en posesión de la imagen se pega en un cartón y se recortan las ranuras señaladas con negro. En el centro se hace un pequeño orificio para aplicarle algún tipo de eje, por ejemplo un lápiz.

### El zoótropo

También se basa en la persistencia de las imágenes en la retina.

Denominado en inglés *zoetrope* o *daedalum*, máquina estroboscópica creada en **1834** por **William George Horner** que consiste en un tambor cilíndrico ranurado en cuyo interior se coloca una tira de papel la longitud de la cual es igual al perímetro de la circunferencia interna del tambor. En dicha tira de papel se dibuja una figura en movimiento, de la que se hacen tantas imágenes o fases de movimiento como ranuras tenga el tambor.

Al hacer girar el tambor y observar su interior a través de las ranuras, se ve como las diversas figuras adquieren movimiento.



Figura 2.10. Zoótropo



## El praxinoscopio

Un praxinoscopio es un aparato similar al zoótropo inventado por Émile Reynaud en 1877 y patentado el 21 de diciembre de ese año. El espectador mira por encima del tambor, dentro del cual hay una rueda interior con unos espejos formando ángulo, que reflejan unas imágenes dibujadas sobre tiras de papel situadas alrededor. Como resultado la persona observa una secuencia nítida, una animación estable donde las imágenes se fusionan y logran el efecto animado. Para fabricar los praxinoscopios que luego vendería como juguetes, Reynaud alquiló dos departamentos en París, uno de los cuales funcionaba como taller. El invento recibió de una mención honorífica en la Exposición Universal de París de 1878.



Figura 2.11. Praxinoscopio

## Disco de Nipkow

En **1882**, el ingeniero Paul **Nipkow** patentó un sistema de barrido de imagen consistente en un disco al que se le habían practicado unos agujeros estratégicamente colocados para recorrer toda la imagen. Estos pequeños agujeros se disponían en forma de espiral.

Si el disco se hacía girar entre una fuente de luz y un objeto, todo ese objeto era explorado por la luz a través de los agujeros después de una revolución del disco. Las porciones más oscuras y las más claras del objeto reflejaban diferentes cantidades de luz y, si esta luz reflejada se dirigía sobre una sucesión de células de selenio, las células enviaban una serie de impulsos eléctricos a través de un alambre.

En el extremo receptor, los impulsos eléctricos se usaban para producir señales que correspondían a la brillantez de las partes individuales del objeto iluminado en el extremo transmisor tal como se puede observar en la figura 2.12. Si las señales, como por ejemplo el encendido de elementos individuales en un mosaico de luces, se veían a través de un segundo disco de Nipkow, el ojo recibía momentáneamente una serie de imágenes transitorias que correspondían a las partes del objeto que se exploraba. Si la rueda se hacía girar lo suficientemente rápido, el ojo transmitía estas partes a la mente como si fueran una sola imagen.

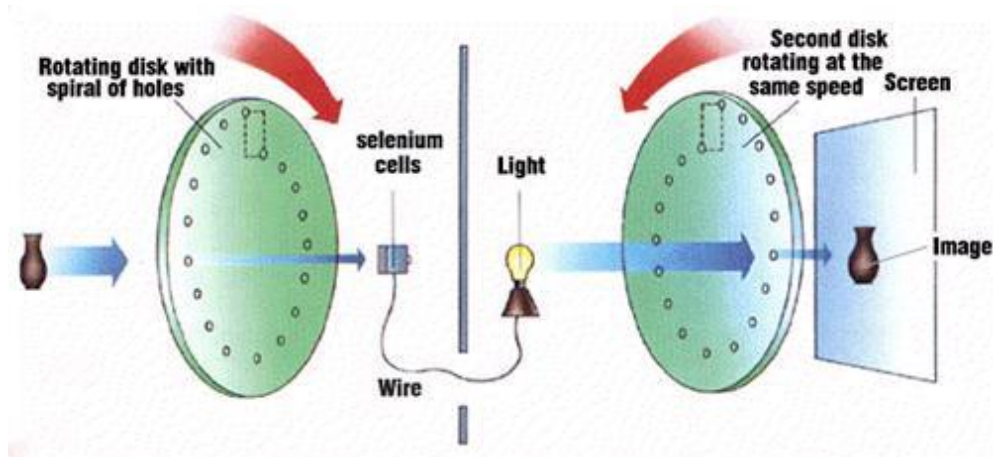


Figura 2.12. Funcionamiento disco de Nipkow

Este sistema tenía un fallo, los motores del disco emisor y del receptor tenían que estar sincronizados, ya que sinó habían problemas de alineado de imagen.

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



Figura 2.13. Televisión electromecánica

## Cine NIC

Ya en el siglo XX, concretamente en la década de los años 30, se creó en España un juguete llamado Cine NIC, en el que se veían imágenes que parecían estar en movimiento.

Se lanzó en el año 1931 y estuvo basado en los principios de la linterna mágica (un aparato óptico, precursor del cinematógrafo, que se basaba en el diseño de la cámara oscura, la cual recibía imágenes del exterior haciéndolas visibles en el interior de la misma, invirtiendo este proceso, y proyectando las imágenes hacia el exterior), pero que por su originalidad y especiales características alcanzó una difusión sin precedentes entre los niños de varias generaciones.



Figura 2.14. Linternas mágica

Este pequeño "cine" fue patentado por los hermanos Nicolau Griñó en Barcelona, el 25 de abril de 1931, y de su fabricación se encargó la empresa catalana Proyector NIC S.A. El funcionamiento era muy simple y a la vez ingenioso: una cinta de papel que tenía dos imágenes ligeramente distintas, arriba y abajo de la misma, era proyectada de modo que un obturador alternaba la visión de una y otra imagen (consiguiendo así una precaria ilusión de movimiento basado solo en dos imágenes que se veían alternativamente), y al mismo tiempo la película pasaba lentamente frente al objetivo, enrollándose a un lado y desenrollándose del otro. La óptica eran dos simples lentes que mostraban la parte inferior y superior de la película, debiendo buscarse el punto de enfoque mediante alejando o acercando el proyector de la pared o pantalla donde se veían las imágenes. La iluminación provenía de una bombilla doméstica.

Aunque pronto se incorporaron innovaciones técnicas, adaptando un gramófono para tener sonido, e incluso los modelos finales tuvieron un motor eléctrico que sustituía la manivela, el modelo básico no varió a lo largo de los años, y fue el más popular. La fabricación del Cine NIC no se interrumpió hasta **1974**, y en ese momento eran ya millones los aparatos vendidos en todo el mundo. Hay que señalar que uno de los atractivos de este juguete estaba en que cualquiera podía crear nuevas películas dibujándolas con cuidado sobre papel vegetal.

Las películas, por otra parte, eran pequeñas obras de arte de la síntesis y el ingenio; su temática era variada, pero sorprende encontrar personajes como **Popeye**, el **gato Félix** o incluso adaptaciones de las primeras películas de **Disney**, como **Pinocho** o **Los tres cerditos**.

Modernamente se ha fabricado y puesto a la venta un proyector similar en el diseño, pero con tecnología actual: metal, lámparas halógenas, etc., a modo de conmemoración, homenaje, o simplemente para nostálgicos.

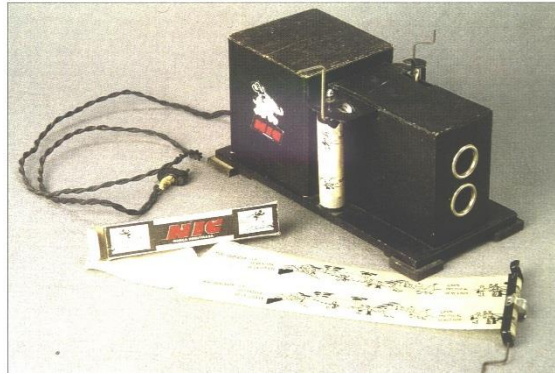


Figura 2.15. Cine NIC

Actualmente, en el mercado de los juguetes y/o gadgets es muy común encontrar relojes, paneles de saludos o bolas de publicidad que sean mensajes transmisibles mediante la palabra o las imágenes.

Este tipo de productos basan su principio en los scroll-texts o pasa-mensajes que podemos ver en el supermercado, en las entradas del metro, en el autobús, etc. En el caso de los paneles informativos, se genera una línea (o columna) de LEDs, que mediante una sincronización de encender y apagar dichos LEDs y su traslación rápida y continua, van formando el mensaje que vemos y leemos por ejemplo en los paneles de información que nos podemos encontrar dentro de los trenes modernos.



Figura 2.16. Paneles de publicidad con LEDs



Figura 2.17. Paneles de información scroll-text

Desde la perspectiva contemporánea, en el campo de la neurofisiología de la percepción, se afirma que la persistencia de la visión o retiniana o persistencia óptica es un mito.

La percepción del movimiento aparente, esto es, la que se obtiene a partir de la observación de secuencias de imágenes estáticas como las que se proyectan sucesivamente en una pantalla de cine o en un televisor o en un monitor de computador -ilustrada en este artículo con el caballo que parece moverse- se explica debido al procesamiento que hace el cerebro de las señales eléctricas provenientes de la retina las cuales a su vez son transmitidas mediante el nervio óptico al núcleo geniculado lateral y luego a otras zonas del cerebro para hacer procesamientos.

A pesar de la gran cantidad de evidencia empírica acopiada durante el último siglo de investigaciones en neurología y en neurofisiología de la percepción, algunas personas no familiarizadas con los hallazgos científicos todavía atribuyen, erróneamente, a la persistencia

retiniana la explicación de la percepción de movimiento aparente como resultado de la observación de secuencias de imágenes estáticas.

Quienes le asignan algún valor a la persistencia retiniana se basan en argumentos débiles como estos: "El ojo hace que cualquier imagen vista se grabe en la retina por un corto tiempo (durante aproximadamente 0,1 segundos). Por ejemplo, al colocar un objeto frente a la vista y retirarlo rápidamente, cualquier persona que estuviera mirándolo lo seguirá viendo, porque el cerebro retiene la impresión de la luz. Esta impresión es notoria al mirar por un momento una fuente de luz como el sol y luego apartar la vista." Una revisión de este argumento nos lleva a la conclusión de que si se superpusieran en la retina diversos estímulos visuales entonces el resultado no sería movimiento sino a lo sumo la mezcla o confusión de dichos estímulos dando lugar a imágenes borrosas, como la que se vería en la pintura de **Marcel Duchamp, *Desnudo descendiendo una escalera***.



Figura 2.18 *Desnudo bajando una escalera* de Duchamp

Este error se deriva de atribuirle una explicación al fenómeno visual solamente a partir de un órgano periférico como es el ojo sin la intervención del cerebro argumento que se ha desvirtuado sistemáticamente por hallazgos como el mencionado por el cognoscitivista Donald Hoffman en su libro *Inteligencia Visual* (pág. 197, Ed. Paidós, Barcelona, 2000) de sujetos que a pesar de tener la retina completamente sana no pueden captar el movimiento en el mundo real, debido a lesiones en el cerebro, a esta disfunción se la denomina acinetopsia, esto es, ceguera al movimiento.



## 2.3– Microcontroladores

### 2.3.1.- Introducción a los microcontroladores

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

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microvatios). Por lo general, tendrá la capacidad para mantener la funcionalidad a la espera de un evento como pulsar un botón o de otra interrupción, el consumo de energía durante el **sueño** (reloj de la CPU y los periféricos de la mayoría) puede ser sólo **nanovatios**, lo que hace que muchos de ellos sean muy **adecuados** para aplicaciones con **batería de larga duración.** Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

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

Un **microcontrolador** normalmente se compone de los siguientes elementos:

- Procesador o CPU.
- Memoria RAM para almacenar los datos.
- Memoria de programa tipo ROM/PROM/EPROM/EEPROM/Flash.
- Líneas de Entrada/Salida para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores o timers, puertos serie y paralelo, ADC: Conversores Analógico/Digital, DAC: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

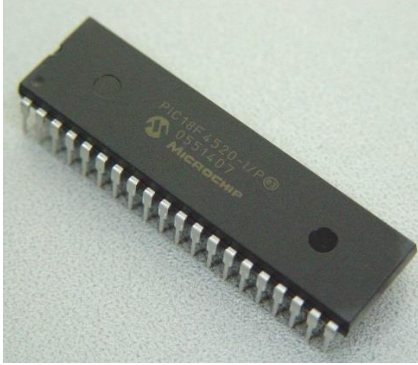


Figura 2.19. Microcontrolador PIC 18F4520



Figura 2.20. Microcontroladores

### 2.3.2.- Repaso histórico

El primer microprocesador fue el Intel 4004 de 4 bits, lanzado en 1971, seguido por el Intel 8008 y otros más capaces. Sin embargo, ambos procesadores requieren circuitos adicionales para implementar un sistema de trabajo, elevando el costo del sistema total.

El Instituto Smithsonian dice que los ingenieros de Texas Instruments Gary Boone y Michael Cochran lograron crear el primer microcontrolador, TMS 1000, en 1971; fue comercializado en 1974. Combina memoria ROM, memoria RAM, microprocesador y reloj en un chip y estaba destinada a los sistemas embebidos.

Debido en parte a la existencia del TMS 1000, Intel desarrolló un sistema de ordenador en un chip optimizado para aplicaciones de control, el Intel 8048, que comenzó a comercializarse en 1977. Combina memoria RAM y ROM en el mismo chip y puede encontrarse en más de mil millones de teclados de compatible IBM PC, y otras numerosas aplicaciones. El en ese momento presidente de Intel, Luke J. Valenter, declaró que el microcontrolador es uno de los productos más exitosos en la historia de la compañía, y amplió el presupuesto de la división en más del 25%.

La mayoría de los microcontroladores en este momento tienen dos variantes. Unos tenían una memoria EPROM reprogramable, significativamente más caros que la variante PROM que era sólo una vez programable. Para borrar la EPROM necesita exponer a la luz ultravioleta la tapa de cuarzo transparente. Los chips con todo opaco representaban un coste menor.

En 1993, el lanzamiento de la EEPROM en los microcontroladores (comenzando con el Microchip PIC16x84) permite borrarla eléctrica y rápidamente sin necesidad de un paquete costoso como se requiere en EPROM, lo que permite tanto la creación rápida de prototipos y la programación en el sistema. El mismo año, Atmel lanza el primer microcontrolador que utiliza memoria flash. Otras compañías rápidamente siguieron el ejemplo, con los dos tipos de memoria.

El costo se ha desplomado en el tiempo, con el más barato microcontrolador de 8 bits disponible por menos de 0,25 dólares para miles de unidades en 2009, y algunos microcontroladores de 32 bits a 1 dólar por cantidades similares. En la actualidad los microcontroladores son baratos y fácilmente disponibles para los aficionados, con grandes comunidades en línea para ciertos procesadores.

En el futuro, la MRAM podría ser utilizada en microcontroladores, ya que tiene resistencia infinita y el coste de su oblea semiconductor es relativamente bajo.



Figura 2.21. Microprocesador Intel 4004



Figura 2.22. Microcontrolador TMS 1000

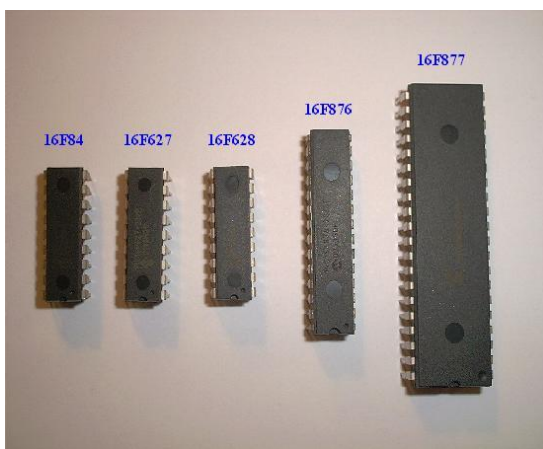


Figura 2.23. PICs de la familia de 16 bits



Figura 2.24. Atmel con memoria flash

### 2.3.3.- Características y aplicaciones

Los **microcontroladores** son **diseñados** para reducir el **coste económico** y el **consumo de energía** de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. El control de un **electrodoméstico sencillo** como una batidora utilizará un **procesador muy pequeño** (4 u 8 bits) porque sustituirá a un autómata finito (modelo computacional que realiza cálculos en forma automática sobre una entrada para producir una salida). En cambio, un **reproductor de música** y/o vídeo digital (MP3 o MP4) requerirá de un **procesador de 32 bits o de 64 bits** y de **uno o más códecs de señal digital** (audio y/o vídeo). El control de un sistema de frenos ABS (Antilock Brake System) se basa normalmente en un microcontrolador de 16 bits, al igual que el sistema de control electrónico del motor en un automóvil.

Los microcontroladores representan la inmensa mayoría de los chips de computadoras vendidos, sobre un 50% son controladores "simples" y el restante corresponde a DSP más especializados. Mientras se pueden tener uno o dos microprocesadores de propósito general en casa (Ud. está usando uno para esto), usted tiene distribuidos seguramente entre los electrodomésticos de su hogar una o dos docenas de microcontroladores. Pueden encontrarse en casi cualquier dispositivo electrónico como automóviles, lavadoras, hornos microondas, teléfonos, etc.



Un microcontrolador difiere de una unidad central de procesamiento normal, debido a que es más fácil convertirla en una computadora en funcionamiento, con un mínimo de circuitos integrados externos de apoyo. La idea es que el circuito integrado se coloque en el dispositivo, enganchado a la fuente de energía y de información que necesite, y eso es todo. Un microprocesador tradicional no le permitirá hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips. Hay que agregarle los módulos de entrada y salida (puertos) y la memoria para almacenamiento de información.

Uno típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria de acceso aleatorio PROM/EPROM/EEPROM/Flash, con lo que para hacerlo funcionar todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidor analógico digital, temporizadores o timers, UARTs (**U**niversal **A**synchronous **R**eceiver-**T**ransmitter) y buses de interfaz serie especializados, como I<sup>2</sup>C y CAN. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados. Los modernos microcontroladores frecuentemente incluyen un lenguaje de programación integrado, como el lenguaje de programación BASIC que se utiliza bastante con este propósito.

Los microcontroladores negocian la velocidad y la flexibilidad para facilitar su uso. Debido a que se utiliza bastante sitio en el chip para incluir funcionalidad, como los dispositivos de entrada/salida o la memoria que incluye el microcontrolador, se ha de prescindir de cualquier otra circuitería.

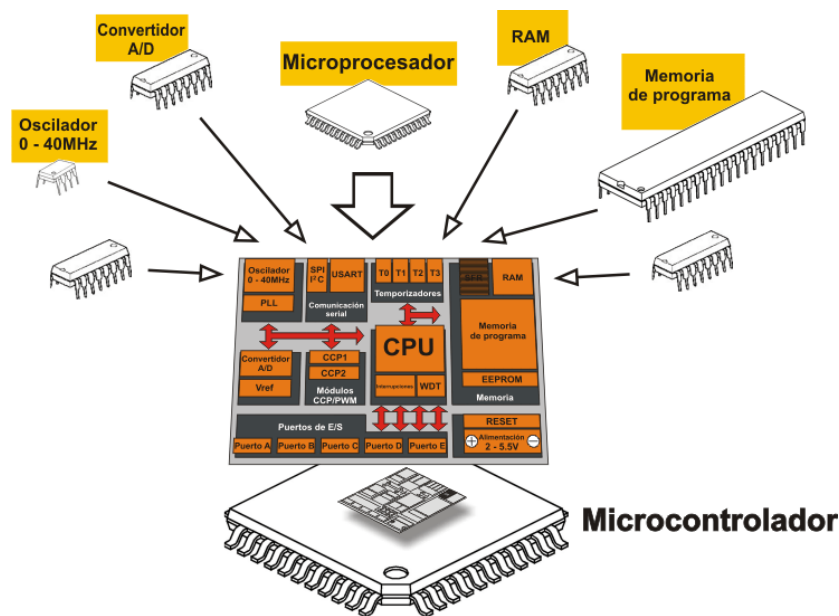


Figura 2.25. Estructura de un microcontrolador



Figura 2.26. Aplicaciones microcontrolador

### 2.3.4.- Arquitecturas

Básicamente existen **dos arquitecturas** de computadoras, y por supuesto, están presentes en el mundo de los microcontroladores: **Von Neumann y Harvard**. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

#### Arquitectura Von Neumann

La arquitectura **Von Neumann** utiliza el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos, siendo la que se utiliza en un ordenador personal porque permite ahorrar una buena cantidad de líneas de E/S, que son bastante costosas, sobre todo para aquellos sistemas donde el procesador se monta en algún tipo de zócalo alojado en una placa madre. También esta organización les ahorra a los diseñadores de placas madre una buena cantidad de problemas y reduce el costo de este tipo de sistemas.

En un ordenador personal, cuando se carga un programa en memoria, a éste se le asigna un espacio de direcciones de la memoria que se divide en segmentos, de los cuales típicamente tendremos los siguientes: código (programa), datos y pila. Es por ello que podemos hablar de la memoria como un todo, aunque existan distintos dispositivos físicos en el sistema (disco duro, memoria RAM, memoria flash, unidad de disco óptico...).

En el caso de los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente algún tipo de SRAM) y memoria de programas (ROM, PROM, EEPROM, flash u de otro tipo no volátil). En este caso la organización es distinta a las del ordenador personal, porque hay circuitos distintos para cada memoria y normalmente no se utilizan los registros de segmentos, sino que la memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

A pesar de que en los sistemas integrados con arquitectura Von Neumann la memoria esté segregada, y existan diferencias con respecto a la definición tradicional de esta arquitectura; los buses para acceder a ambos tipos de memoria son los mismos, del procesador solamente salen el bus de datos, el de direcciones, y el de control. Como conclusión, la arquitectura no ha sido alterada, porque la forma en que se conecta la memoria al procesador sigue el mismo principio definido en la arquitectura básica.

Algunas familias de microcontroladores como la Intel 8051 y la Z80 implementan este tipo de arquitectura, fundamentalmente porque era la utilizada cuando aparecieron los primeros microcontroladores.

## Arquitectura Harvard

La otra variante es la **arquitectura Harvard**, y por excelencia la utilizada en supercomputadoras, en los microcontroladores, y sistemas integrados en general. En este caso, además de la memoria, el procesador tiene los buses segregados, de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

La ventaja fundamental de esta arquitectura es que permite adecuar el tamaño de los buses a las características de cada tipo de memoria; además, el procesador puede acceder a cada una de ellas de forma simultánea, lo que se traduce en un aumento significativo de la velocidad de procesamiento. Típicamente los sistemas con esta arquitectura pueden ser dos veces más rápidos que sistemas similares con arquitectura Von Neumann.

La desventaja está en que consume muchas líneas de E/S del procesador; por lo que en sistemas donde el procesador está ubicado en su propio encapsulado, solo se utiliza en supercomputadoras. Sin embargo, en los microcontroladores y otros sistemas integrados, donde usualmente la memoria de datos y programas comparten el mismo encapsulado que el procesador, este inconveniente deja de ser un problema serio y es por ello que encontramos la arquitectura Harvard en la mayoría de los microcontroladores.

Por eso es importante recordar que un microcontrolador se puede configurar de diferentes maneras, siempre y cuando se respete el tamaño de memoria que este requiera para su correcto funcionamiento.

Este tipo de arquitectura es la que se desarrolla en el microcontrolador PIC que vamos a utilizar en este proyecto.

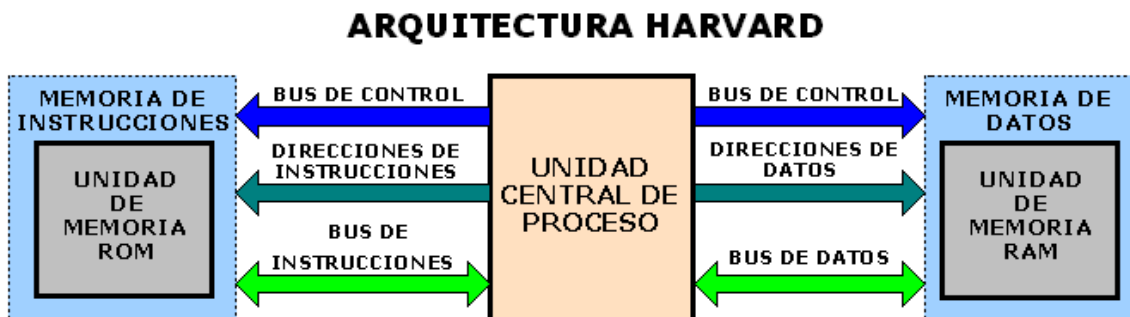


Figura 2.27. Arquitectura Harvard

### 2.3.5.- Diferencia entre microprocesador y microcontrolador

	<b>Microprocesadores</b>	<b>Microcontroladores</b>
<b>CPU</b>	El microcontrolador es igual de bruto que un micro, por lo cual solamente realiza sus funciones con lo que tiene (datos) y su algoritmo o programa establecida.	Es una de sus funciones principales, la cual se encarga de dirigir sus operaciones.

<b>Memorias RAM y ROM</b>	Son dispositivos externos que lo complementan para su óptimo funcionamiento.	Las incluye en un solo circuito integrado.
<b>Velocidad de operación</b>	Rápida	Lenta en comparación con la de un microprocesador.
<b>Tamaño</b>	La configuración mínima básica de un Microprocesador está constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria ROM de 28 Pines y un decodificador de direcciones de 18 Pines, lo cual lo convierte en un circuito bastante engorroso.	El microcontrolador incluye todo estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores, como por ejemplo, la disminución en el tamaño del circuito impreso por la reducción de los circuitos externos.
<b>Costes</b>	Para el microprocesador, el coste es muy alto en la actualidad.	El coste para un sistema basado en microcontrolador es mucho menor.
<b>Interferencias</b>	Son más susceptibles a la interferencia electromagnética debido a su tamaño y a su cableado externo que lo hace más propenso al ruido.	El alto nivel de integración reduce los niveles de interferencia electromagnética
<b>Tiempo de desarrollo</b>	El tiempo de desarrollo de un microprocesador es lento.	Por el contrario, el de un microcontrolador es rápido.

### 2.3.6.- Microcontrolador PIC

El **microcontrolador utilizado** en este proyecto es de la familia **PIC**, tipo **RISC** fabricados por **Microchip Technology Inc.** y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El nombre actual no es un acrónimo. En realidad, el nombre completo es **PICmicro**, aunque generalmente se utiliza como *Peripheral Interface Controller* (controlador de interfaz periférico).

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada y salida, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba microcódigo simple almacenado en ROM para realizar estas tareas; y aunque el

término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador.

En 1985 la división de microelectrónica de General Instrument se separa como compañía independiente que es incorporada como filial (el 14 de diciembre de 1987 cambia el nombre a Microchip Technology y en 1989 es adquirida por un grupo de inversores) y el nuevo propietario canceló casi todos los desarrollos, que para esas fechas la mayoría estaban obsoletos. El PIC, sin embargo, se mejoró con EPROM para conseguir un controlador de canal programable. Hoy en día multitud de PICs vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una *palabra* corresponde a una instrucción en lenguaje ensamblador, y puede ser de 12, 14, 16 ó 32 bits, dependiendo de la familia específica de PICmicro).

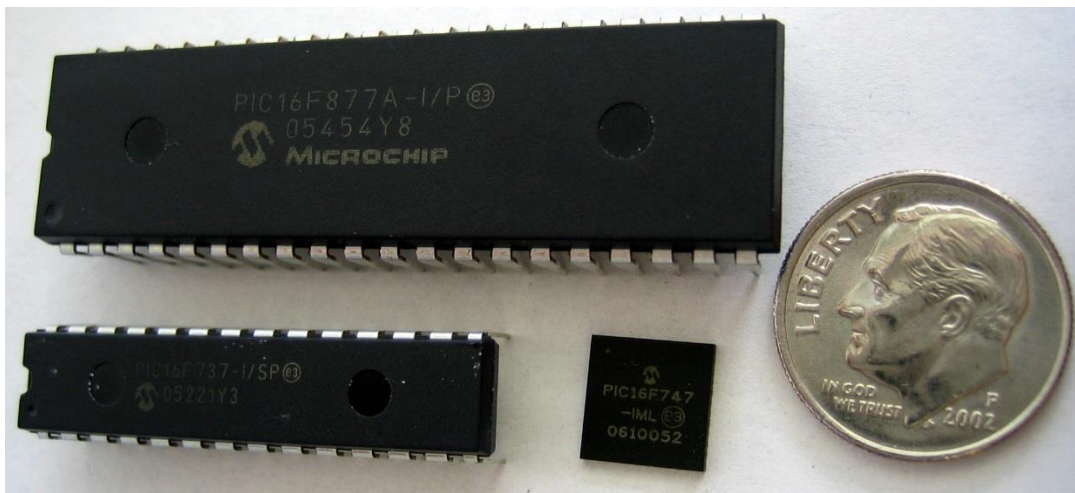


Figura 2.28. Comparativa tamaño varios PICs

### 2.3.6.1- Juego de instrucciones

El PIC usa un juego de instrucciones tipo RISC, cuyo número puede variar desde 35 para PICs de gama baja a 70 para los de gama alta. Las instrucciones se clasifican entre las que realizan operaciones entre el acumulador y una constante, entre el acumulador y una posición de memoria, instrucciones de condicionamiento y de salto/retorno, implementación de interrupciones y una para pasar a modo de bajo consumo llamada *sleep*.

Microchip proporciona un entorno de desarrollo freeware llamado MPLAB que incluye un simulador software y un ensamblador. Otras empresas desarrollan compiladores C y BASIC. Microchip también vende compiladores para los PICs de gama alta ("C18" para la serie F18 y "C30" para los dsPICs) y se puede descargar una edición para estudiantes del C18 que inhabilita algunas opciones después de un tiempo de evaluación.

Para el lenguaje de programación Pascal existe un compilador de código abierto, JAL, lo mismo que PicForth para el lenguaje Forth. GPUTILS es una colección de herramientas distribuidas bajo licencia GPL que incluye ensamblador y enlazador, y funciona en Linux, MacOS y Microsoft Windows. GPSIM es otra herramienta libre que permite simular diversos dispositivos hardware conectados al PIC.

Uno de los más modernos y completos compiladores para lenguaje C es [mikroC], que es un ambiente de desarrollo con editor de texto, bibliotecas con múltiples funciones para todos los módulos y herramientas incorporadas para facilitar enormemente el proceso de programación.

### 2.3.6.2- Arquitectura del PIC

La arquitectura del PIC es sumamente minimalista. Está caracterizada por las siguientes prestaciones:

- Área de código y de datos separadas (Arquitectura Harvard).
- Un reducido número de instrucciones de longitud fija.
- Implementa segmentación de tal modo que la mayoría de instrucciones duran 1 tiempo de instrucción (o 4 tiempos de reloj). Pueden haber instrucciones de dos tiempos de instrucción (saltos, llamadas y retornos de subrutinas y otras) o inclusive con más tiempo de instrucción en PICs de gama alta. Esto implica que el rendimiento real de instrucciones por segundo del procesador es de al menos 1/4 de la frecuencia del oscilador.
- Un solo acumulador (W), cuyo uso (como operador de origen) es implícito (no está especificado en la instrucción).
- Todas las posiciones de la RAM funcionan como registros de origen y/o de destino de operaciones matemáticas y otras funciones.
- Una pila de hardware para almacenar instrucciones de regreso de funciones.
- Una relativamente pequeña cantidad de espacio de datos direccionable (típicamente, 256 bytes), extensible a través de manipulación de bancos de memoria.
- El espacio de datos está relacionado con el CPU, puertos, y los registros de los periféricos.
- El contador de programa está también relacionado dentro del espacio de datos, y es posible escribir en él (permitiendo saltos indirectos).

A diferencia de la mayoría de otros CPU, no hay distinción entre los espacios de memoria y los espacios de registros, ya que la RAM cumple ambas funciones, y esta es normalmente referida como "archivo de registros" o simplemente, registros.

#### **Espacio de datos (RAM)**

Los **microcontroladores PIC** tienen una serie de registros que funcionan como una **RAM** de propósito general. Los registros de propósito específico para los recursos de hardware disponibles dentro del propio chip también están direccionados en la RAM. La direccionabilidad de la memoria varía dependiendo de la línea de dispositivos, y todos los dispositivos PIC tienen algún tipo de mecanismo de manipulación de bancos de memoria que pueden ser usados para acceder memoria externa o adicional. Las series más recientes de dispositivos disponen de funciones que pueden cubrir todo el espacio direccionable, independientemente del banco de memoria seleccionado. En los dispositivos anteriores, esto debía lograrse mediante el uso del acumulador.

Para implementar direccionamiento indirecto, se usa un registro de "selección de registro de archivo" (FSR) y uno de "registro indirecto" (INDF): Un número de registro es escrito en el FSR, haciendo que las lecturas o escrituras al INDF serán realmente hacia o desde el registro apuntado por el FSR. Los dispositivos más recientes extienden este concepto con post y



preincrementos/decrementos para mayor eficiencia al acceder secuencialmente a la información almacenada. Esto permite que se pueda tratar al FSR como un puntero de pila.

La memoria de datos externa no es directamente direccionable excepto en algunos microcontroladores PIC 18 de gran cantidad de pines.

### **Tamaño de palabra**

El tamaño de palabra de los microcontroladores PIC es fuente de muchas confusiones. Todos los PICs (excepto los dsPIC) manejan datos en trozos de 8 bits, con lo que se deberían llamar microcontroladores de 8 bits. Pero a diferencia de la mayoría de las CPU, el PIC usa arquitectura Harvard, por lo que el tamaño de las instrucciones puede ser distinto del de la palabra de datos.

De hecho, las diferentes familias de PICs usan tamaños de instrucción distintos, lo que hace difícil comparar el tamaño del código del PIC con el de otros microcontroladores. Por ejemplo, un microcontrolador tiene 6144 bytes de memoria de programa: para un PIC de 12 bits esto significa 4096 palabras y para uno de 16 bits, 3072 palabras.

#### 2.3.6.3- Programación del PIC

Para transferir el código de un ordenador al PIC normalmente se usa un dispositivo llamado programador. La mayoría de PICs que Microchip distribuye hoy en día incorporan ICSP (*In Circuit Serial Programming*, programación serie incorporada) o LVP (*Low Voltage Programming*, programación a bajo voltaje), lo que permite programar el PIC directamente en el circuito destino. Para la ICSP se usan los pines RB6 y RB7 (En algunos modelos pueden usarse otros pines como el GP0 y GP1 o el RA0 y RA1) como reloj y datos y el MCLR para activar el modo programación aplicando un voltaje de 13 voltios.

Existen muchos programadores de PICs, desde los más simples que dejan al software los detalles de comunicaciones, a los más complejos, que pueden verificar el dispositivo a diversas tensiones de alimentación e implementan en hardware casi todas las funcionalidades. Muchos de estos programadores complejos incluyen ellos mismos PICs preprogramados como interfaz para enviar las órdenes al PIC que se desea programar. Uno de los programadores más simples es el TE20, que utiliza la línea TX del puerto RS232 como alimentación y las líneas DTR y CTS para mandar o recibir datos cuando el microcontrolador está en modo programación. El software de programación puede ser el ICprog, muy común entre la gente que utiliza este tipo de microcontroladores, como también entornos de programación basados en intérpretes BASIC que nos facilitan la vida.

### **Programadores**

- PICStart Plus (puerto serie y USB)
- Promate II (puerto serie)
- MPLAB PM3 (puerto serie y USB)
- ICD2 (puerto serie y USB)
- ICD3 (USB)
- PICKit 1 (USB)

- IC-Prog 1.06B
  - PICAT 1.25 (puerto USB2.0 para PICs y Atmel)
  - WinPic 800 (puerto paralelo, serie y USB)
  - PICKit 2 (USB)
  - **PICKit 3 (USB) usado en este proyecto.**
  - Terusb1.0
  - Eclipse (PICs y AVR. USB.)
  - MasterProg (USB)
- Además es posible hacer un programador de manera casera en <http://microspics.blogspot.com> hay una lista con los más utilizados.



Figura 2.29. Programador PICKit 2

## Depuradores integrados

- ICD (Serie)
- ICD2 (Serie ó full speed USB - 2M bits/s)
- ICD3 (High speed USB - 480M bits/s)

## Emuladores

- Proteus - ISIS
- ICE2000 (puerto paralelo, convertidor a USB disponible)
- ICE4000 (USB)
- PIC EMU
- ISEC
- PIC CDlite
- PIC Simulator



#### 2.3.6.4- Características

Los PICs actuales vienen con una amplia gama de mejoras hardware incorporadas:

- Núcleos de CPU de 8/16 bits con Arquitectura Harvard modificada
- Memoria Flash y ROM disponible desde 256 bytes a 256 kilobytes
- Puertos de E/S (típicamente 0 a 5,5 voltios)
- Temporizadores de 8/16/32 bits
- Tecnología Nanowatt para modos de control de energía
- Periféricos serie síncronos y asíncronos: USART, AUSART, EUSART
- Conversores analógico/digital de 8-10-12 bits
- Comparadores de tensión
- Módulos de captura y comparación PWM
- Controladores LCD
- Periférico MSSP para comunicaciones I<sup>2</sup>C, SPI, y I<sup>2</sup>S
- Memoria EEPROM interna con duración de hasta un millón de ciclos de lectura/escritura
- Periféricos de control de motores
- Soporte de interfaz USB
- Soporte de controlador Ethernet
- Soporte de controlador CAN
- Soporte de controlador LIN
- Soporte de controlador Irda

#### 2.3.6.5- Variaciones del PIC

##### **PICs modernos**

Los viejos PICs con memoria PROM o EPROM se están renovando gradualmente por chips con memoria Flash. Así mismo, el juego de instrucciones original de 12 bits del PIC1650 y sus descendientes directos ha sido suplantado por juegos de instrucciones de 14 y 16 bits. Microchip todavía vende versiones PROM y EPROM de la mayoría de los PICs para soporte de aplicaciones antiguas o grandes pedidos.

Se pueden considerar tres grandes gamas de MCUs PIC en la actualidad: Los básicos (Linebase), los de medio rango (Mid Range) y los de alto desempeño (high performance). Los PIC18 son considerandos de alto desempeño y tienen entre sus miembros a PICs con módulos de comunicación y protocolos avanzados (USB, Ethernet, Zigbee por ejemplo).

##### **Clones del PIC**

Por todos lados surgen compañías que ofrecen versiones del PIC más baratas o mejoradas. La mayoría suelen desaparecer rápidamente. Una de ellas que va perdurando es Ubicom (antiguamente Scenix) que vende clones del PIC que funcionan mucho más rápido que el original. OpenCores tiene un núcleo del PIC16F84 escrito en Verilog.

## PICs wireless

El **microcontrolador rPIC** integra todas las prestaciones del PICmicro de Microchip con la capacidad de comunicación wireless UHF para aplicaciones RF de baja potencia. Estos dispositivos ofrecen un diseño muy comprimido para ajustarse a los cada vez más demandados requerimientos de miniaturización en aparatos electrónicos. Aun así, no parecen tener mucha salida en el mercado.

## PICs para procesamiento de señal (dsPICs)

Los **dsPICs** son el penúltimo lanzamiento de Microchip, comenzando a producirlos a gran escala a finales de 2004. Son los primeros PICs con bus de datos inherente de 16 bits. Incorporan todas las posibilidades de los anteriores PICs y añaden varias operaciones de **DSP** implementadas en hardware, como multiplicación con suma de acumulador (*multiply-accumulate*, o *MAC*), *barrel shifting*, *bit reversion* o multiplicación 16x16 bits.

## PICs de 32 bits (PIC32)

Microchip Technology lanzó en noviembre de 2007 los nuevos microcontroladores de 32 bits con una velocidad de procesamiento de hasta 1.6 DMIPS/MHz con capacidad HOST USB. Sus frecuencias de reloj pueden alcanzar los 80MHz a partir de cuarzos estándares de 4 a 5MHz gracias a un PLL interno. Funcionan a 3.3V en sus puertos de entrada y salida, aunque el fabricante indica que salvo en los pines con función analógica, en la mayoría se toleran tensiones de hasta 5V. Disponen de una arquitectura optimizada con alto grado de paralelismo y núcleo de tipo M4K y una elevada capacidad de memoria RAM y FLASH ROM. Todo ello hace que estos MCUs permiten un elevado procesamiento de información.

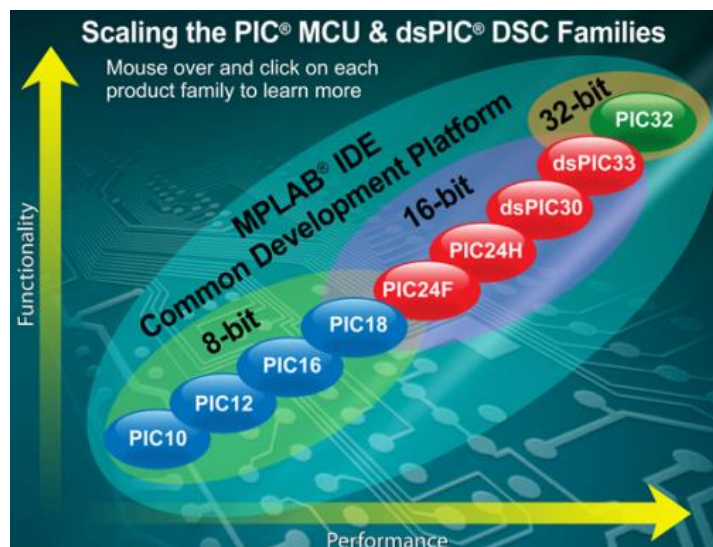


Figura 2.30. Clasificación según el tamaño de palabra

### 2.3.6.6- PICs más usados

- PIC12C508/509 (encapsulamiento reducido de 8 pines, oscilador interno, popular en pequeños diseños como el iPod remote).

- PIC12F629/675
- [PIC16F84](#) (Considerado obsoleto, pero imposible de descartar y muy popular)
- PIC16F84A (Buena actualización del anterior, algunas versiones funcionan a 20 MHz, compatible 1:1)
- PIC16F628A (Es la opción típica para iniciar una migración o actualización de diseños antiguos hechos con el PIC16F84A. Posee puerto serial, módulos de comparación análoga, PWM, módulo CCP, rango de operación de voltaje aumentado, entre otras )
- PIC16F88 (Nuevo sustituto del PIC16F84A con más memoria, oscilador interno, PWM, etc que podría convertirse en popular como su hermana).
- La subfamilia PIC16F87X y PIC16F87XA (los hermanos mayores del PIC16F84 y PIC16F84A, con cantidad de mejoras incluidas en hardware. Bastante común en proyectos de aficionados).
- PIC16F886/887 (Nuevo sustituto del 16F876A y 16F877A con la diferencia que el nuevo ya se incluye oscilador interno).
- PIC16F193x (Nueva gama media de PIC optimizado y con mucha RAM, ahora con 49 instrucciones por primera vez frente a las 35 de toda la vida).
- PIC18F2455 y similares con puerto USB 2.0
- PIC18F2550 manejo de puertos USB 2.0 y muy versátil.
- PIC18F452
- PIC18F4550
- dsPIC30F2010
- dsPIC30F3014
- dsPIC30F3011 (Ideales para control electrónico de motores eléctricos de inducción, control sobre audio, etc).
- PIC32 (Nueva gama de PIC de 32 bits, los más modernos ya compatible con USB 2.0).

### 2.3.7.- Familia PIC18 de Microchip

La familia de microcontroladores PIC18 de Microchip fueron introducidos en el año 2000 y provee dispositivos PICmicro en paquetes de 18 a 80 pines, cuyos sockets y software son compatibles con la familia PIC16. También incluye todos los periféricos populares, como el MSSP (Puerto Serie Maestro síncrono), ESCI (Enhanced Serial Communications Interface), CCP (Capture/Compare/PWM), temporizadores de 8 y 16 bit, PSP, convertidores A-D de 10 bit, WDT (Perro guardián), POR (Power-On Reset) y CAN 2.0B activo para la máxima flexibilidad. Muchos dispositivos de la familia PIC18 incluyen memoria de programa Flash en tamaños que van desde los 8 hasta los 128 Kbytes y RAM de datos de 4 hasta 256 Kbytes, operando desde los 2 hasta los 5,5 V. a velocidades de reloj de 40 MHz. Optimizados para lenguajes de alto nivel como el ANSI C, la familia PIC18 ofrece grandes soluciones a complejas aplicaciones.

A continuación, veremos una tabla dónde se detallan las características principales de todos los dispositivos PIC18 lanzados hasta la fecha, en donde queda remarcado el PIC usado en este proyecto, el 18F4680:

Producto	Arquitectura	Tipo de memoria	Memoria de programa KBytes	Memoria de datos EEPROM	RAM	Pi ns E/ S	Max. Vel. CPUM Hz
PIC18F13K22	8	Flash	8	256	256	18	64
PIC18F24J10	8	Flash	16	0	1024	21	40
PIC18F23K20	8	Flash	8	256	512	25	64
PIC18F14K22	8	Flash	16	256	512	18	64
PIC18F25J10	8	Flash	32	0	1024	21	40
PIC18F24K20	8	Flash	16	256	768	25	64
PIC18F13K50	8	Flash	8	256	512	15	48
PIC18F23K22	8	Flash	8	256	512	25	64
PIC18F25K20	8	Flash	32	256	1536	25	64
PIC18F44J10	8	Flash	16	0	1024	32	40
PIC18F43K20	8	Flash	8	256	512	36	64
PIC18F24K22	8	Flash	16	256	768	25	64
PIC18F45J10	8	Flash	32	0	1024	32	40
PIC18F14K50	8	Flash	16	256	768	15	48
PIC18F44K20	8	Flash	16	256	768	36	64
PIC18F45K20	8	Flash	32	256	1536	36	64
PIC18F25K22	8	Flash	32	256	1536	25	64
PIC18F24J11	8	Flash	16	0	3800	21	48
PIC18F24K50	8	Flash	16	256	2048	25	48
PIC18F26K20	8	Flash	64	1024	3936	25	64
PIC18F43K22	8	Flash	8	256	512	36	64
PIC18F44K22	8	Flash	16	256	768	36	64
PIC18F66J11	8	Flash	64	0	3904	50	48
PIC18F25K50	8	Flash	32	256	2048	25	48
PIC18F25J11	8	Flash	32	0	3800	21	48
PIC18F46K20	8	Flash	64	1024	3936	36	64
PIC18F65J10	8	Flash	32	0	2048	50	40
PIC18F63J11	8	Flash	8	0	1024	54	40
PIC18F24J50	8	Flash	16	0	3800	22	48
PIC18F66J16	8	Flash	96	0	3904	50	48
PIC18F45K22	8	Flash	32	256	1536	36	64
PIC18F25K80	8	Flash	32	1024	3648	24	64
PIC18F26K22	8	Flash	64	1024	3896	25	64
PIC18F64J11	8	Flash	16	0	1024	54	40
PIC18F2221	8	Flash	4	256	512	25	40
PIC18F44J11	8	Flash	16	0	3800	34	48
PIC18F67J11	8	Flash	128	0	3904	50	48
PIC18F1220	8	Flash	4	256	256	16	40
PIC18F65J15	8	Flash	48	0	2048	50	40
PIC18F45K50	8	Flash	32	256	2048	36	48
PIC18F25J50	8	Flash	32	0	3800	22	48
PIC18F63J90	8	Flash	8	0	1024	51	40

PIC18F65J11	8	Flash	32	0	2048	54	40
PIC18F1230	8	Flash	4	128	256	16	40
PIC18F2321	8	Flash	8	256	512	25	40
PIC18F64J90	8	Flash	16	0	1024	51	40
PIC18F85J10	8	Flash	32	0	2048	66	40
PIC18F26J11	8	Flash	64	0	3800	21	48
PIC18F45J11	8	Flash	32	0	3800	34	48
PIC18F66J10	8	Flash	64	0	2048	50	40
PIC18F83J11	8	Flash	8	0	1024	70	40
PIC18F44J50	8	Flash	16	0	3800	34	48
PIC18F46K22	8	Flash	64	1024	3896	36	64
PIC18F1320	8	Flash	8	256	256	16	40
PIC18F4221	8	Flash	4	256	512	36	40
PIC18F45K80	8	Flash	32	1024	3648	35	64
PIC18F65J90	8	Flash	32	0	2048	50	40
PIC18F26K80	8	Flash	64	1024	3648	24	64
PIC18F84J11	8	Flash	16	0	1024	70	40
PIC18F66J15	8	Flash	96	0	3936	50	40
PIC18F85J15	8	Flash	48	0	2048	66	40
PIC18F26J13	8	Flash	64	0	3800	22	48
PIC18F2450	8	Flash	16	0	768	23	48
PIC18F1330	8	Flash	8	128	256	16	40
PIC18F83J90	8	Flash	8	0	1024	66	40
PIC18F26J50	8	Flash	64	0	3800	22	48
PIC18F85J11	8	Flash	32	0	2048	70	40
PIC18F4321	8	Flash	8	256	512	36	40
PIC18F45J50	8	Flash	32	0	3800	34	48
PIC18F67J10	8	Flash	128	0	3936	50	40
PIC18F65J50	8	Flash	32	0	3904	49	48
PIC18F84J90	8	Flash	16	0	1024	66	40
PIC18F86J10	8	Flash	64	0	2048	66	40
PIC18F46J11	8	Flash	64	0	3800	34	48
PIC18F65J94	8	Flash	32	0	3862	53	64
PIC18F4450	8	Flash	16	0	768	34	48
PIC18F65K22	8	Flash	32	1024	2048	53	64
PIC18F26J53	8	Flash	64	0	3800	22	48
PIC18F85J90	8	Flash	32	0	2048	66	40
PIC18F86J15	8	Flash	96	0	3936	66	40
PIC18F27J13	8	Flash	128	0	3800	22	48
PIC18F46K80	8	Flash	64	1024	3648	35	64
PIC18F66J90	8	Flash	64	0	3900	51	48
PIC18F2410	8	Flash	16	0	768	25	40
PIC18F46J13	8	Flash	64	0	3800	34	48
PIC18F6310	8	Flash	8	0	768	54	40
PIC18F65K90	8	Flash	32	1024	2048	53	64
PIC18F87J10	8	Flash	128	0	3936	66	40

<b>PIC18F46J50</b>	8	Flash	64	0	3800	34	48
<b>PIC18F66J50</b>	8	Flash	64	0	3904	49	48
<b>PIC18F85J50</b>	8	Flash	32	0	3904	65	48
<b>PIC18F86J11</b>	8	Flash	64	0	3904	66	48
<b>PIC18F66J93</b>	8	Flash	64	0	3900	51	48
<b>PIC18F85J94</b>	8	Flash	32	0	3862	69	64
<b>PIC18F27J53</b>	8	Flash	128	0	3800	22	48
<b>PIC18F6390</b>	8	Flash	8	0	768	50	40
<b>PIC18F85K22</b>	8	Flash	32	1024	2048	69	64
<b>PIC18F66J94</b>	8	Flash	64	0	3862	53	64
<b>PIC18F65K80</b>	8	Flash	32	1024	3648	54	64
<b>PIC18F66K22</b>	8	Flash	64	1024	3862	53	64
<b>PIC18F86J16</b>	8	Flash	96	0	3904	66	48
<b>PIC18F66J55</b>	8	Flash	96	0	3904	49	48
<b>PIC18F67J90</b>	8	Flash	128	0	3900	51	48
<b>PIC18F2420</b>	8	Flash	16	256	768	25	40
<b>PIC18F4410</b>	8	Flash	16	0	768	36	40
<b>PIC18F46J53</b>	8	Flash	64	0	3800	34	48
<b>PIC18F86J90</b>	8	Flash	64	0	3900	67	48
<b>PIC18F47J13</b>	8	Flash	128	0	3800	34	48
<b>PIC18F85K90</b>	8	Flash	32	1024	2048	69	64
<b>PIC18F67J50</b>	8	Flash	128	0	3904	49	48
<b>PIC18F87J11</b>	8	Flash	128	0	3904	66	48
<b>PIC18F95J94</b>	8	Flash	32	0	3862	87	64
<b>PIC18F66K90</b>	8	Flash	64	1024	3828	53	64
<b>PIC18F67J93</b>	8	Flash	128	0	3900	51	48
<b>PIC18F86J50</b>	8	Flash	64	0	3904	65	48
<b>PIC18F6410</b>	8	Flash	16	0	768	54	40
<b>PIC18F86J93</b>	8	Flash	64	0	3900	67	48
<b>PIC18F67J94</b>	8	Flash	128	0	3862	53	64
<b>PIC18F4420</b>	8	Flash	16	256	768	36	40
<b>PIC18F6393</b>	8	Flash	8	0	768	50	40
<b>PIC18F67K22</b>	8	Flash	128	1024	3862	53	64
<b>PIC18F86J94</b>	8	Flash	64	0	3862	69	64
<b>PIC18F2510</b>	8	Flash	32	0	1536	25	40
<b>PIC18F47J53</b>	8	Flash	128	0	3800	34	48
<b>PIC18F86K22</b>	8	Flash	64	1024	3862	69	64
<b>PIC18F66K80</b>	8	Flash	64	1024	3648	54	64
<b>PIC18F86J55</b>	8	Flash	96	0	3904	66	48
<b>PIC18F87J90</b>	8	Flash	128	0	3900	67	48
<b>PIC18F6490</b>	8	Flash	16	0	768	50	40
<b>PIC18F8310</b>	8	Flash	8	0	768	70	40
<b>PIC18F2423</b>	8	Flash	16	256	768	25	40
<b>PIC18F66J60</b>	8	Flash	64	0	3808	39	42
<b>PIC18F2331</b>	8	Flash	8	256	768	24	40
<b>PIC18F67K90</b>	8	Flash	128	1024	3828	53	64

PIC18F87J50	8	Flash	128	0	3904	65	48
PIC18F86K90	8	Flash	64	1024	3828	69	64
PIC18F87J93	8	Flash	128	0	3900	67	48
PIC18F96J94	8	Flash	64	0	3862	87	64
PIC18F8390	8	Flash	8	0	768	66	40
PIC18F2520	8	Flash	32	256	1536	25	40
PIC18F4510	8	Flash	32	0	1536	36	40
PIC18F66J65	8	Flash	96	0	3808	39	42
PIC18F87J94	8	Flash	128	0	3862	69	64
PIC18F87K22	8	Flash	128	1024	3862	69	64
PIC18F2431	8	Flash	16	256	768	24	40
PIC18F4423	8	Flash	16	256	768	36	40
PIC18F6493	8	Flash	16	0	768	50	32
PIC18F2455	8	Flash	24	256	2048	24	48
PIC18F67J60	8	Flash	128	0	3808	39	42
PIC18F86J60	8	Flash	64	0	3808	55	42
PIC18F87K90	8	Flash	128	1024	4096	69	64
PIC18F8410	8	Flash	16	0	768	70	40
PIC18F97J94	8	Flash	128	0	3862	87	64
PIC18F4520	8	Flash	32	256	1536	36	40
PIC18F2515	8	Flash	48	0	3968	25	40
PIC18F8393	8	Flash	8	0	768	66	40
PIC18F2550	8	Flash	32	256	2048	24	48
PIC18F86J65	8	Flash	96	0	3808	55	42
PIC18F2523	8	Flash	32	256	1536	25	40
PIC18F4331	8	Flash	8	256	768	36	40
PIC18F8490	8	Flash	16	0	768	66	40
PIC18F4455	8	Flash	24	256	2048	35	48
PIC18F96J60	8	Flash	64	0	3808	70	42
PIC18F87J60	8	Flash	128	0	3808	55	42
PIC18F2458	8	Flash	24	256	2048	24	48
PIC18F2525	8	Flash	48	1024	3968	25	40
PIC18F4431	8	Flash	16	256	768	36	40
PIC18F4515	8	Flash	48	0	3968	36	40
PIC18F4550	8	Flash	32	256	2048	35	48
PIC18F96J65	8	Flash	96	0	3808	70	42
PIC18F4523	8	Flash	32	256	1536	36	40
PIC18F2480	8	Flash	16	256	768	25	40
PIC18F97J60	8	Flash	128	0	3808	70	42
PIC18F8493	8	Flash	16	0	768	66	32
PIC18F4458	8	Flash	24	256	2048	35	48
PIC18F4525	8	Flash	48	1024	3968	36	40
PIC18F2610	8	Flash	64	0	3968	25	40
PIC18F4480	8	Flash	16	256	768	36	40
PIC18F2220	8	Flash	4	256	512	25	40
PIC18F2620	8	Flash	64	1024	3968	25	40

PIC18F4610	8	Flash	64	0	3968	36	40
PIC18F2553	8	Flash	32	256	2048	24	48
PIC18F86J72	8	Flash	64	0	3923	51	48
PIC18F2580	8	Flash	32	256	1536	25	40
PIC18F4620	8	Flash	64	1024	3968	36	40
PIC18F4553	8	Flash	32	256	2048	35	48
PIC18F87J72	8	Flash	128	0	3923	51	48
PIC18F2320	8	Flash	8	256	512	25	40
PIC18F4580	8	Flash	32	256	1536	36	40
PIC18F4220	8	Flash	4	256	512	36	40
PIC18F6527	8	Flash	48	1024	3936	54	40
PIC18F2585	8	Flash	48	1024	3328	25	40
PIC18F4320	8	Flash	8	256	512	36	40
PIC18F4585	8	Flash	48	1024	3328	36	40
PIC18F8527	8	Flash	48	1024	3936	70	40
PIC18F2680	8	Flash	64	1024	3328	25	40
PIC18F6622	8	Flash	64	1024	3936	54	40
PIC18F4680	8	Flash	64	1024	3328	36	40
PIC18F2682	8	Flash	80	1024	3328	25	40
PIC18F8622	8	Flash	64	1024	3936	70	40
PIC18F4682	8	Flash	80	1024	3328	36	40
PIC18F2685	8	Flash	96	1024	3328	25	40
PIC18F6520	8	Flash	32	1024	2048	52	40
PIC18F6627	8	Flash	96	1024	3936	54	40
PIC18F4685	8	Flash	96	1024	3328	36	40
PIC18F8520	8	Flash	32	1024	2048	68	40
PIC18F8627	8	Flash	96	1024	3936	70	40
PIC18F6722	8	Flash	128	1024	3936	54	40
PIC18F6628	8	Flash	96	1024	3936	54	40
PIC18F8722	8	Flash	128	1024	3936	70	40
PIC18F8628	8	Flash	96	1024	3936	70	40
PIC18F6723	8	Flash	128	1024	3936	54	40
PIC18F8723	8	Flash	128	1024	3936	70	40

Como se puede apreciar, todos los dispositivos tienen una arquitectura de 8 bit con memoria flash de rápido acceso que va desde los 4 hasta los 128 Kbytes, con una velocidad de reloj media de 40 MHz.

Resumiendo, tenemos que las principales características de esta familia son las siguientes:

- Arquitectura RISC Harvard de 16 bit de ancho de código de memoria y 8 bit de datos.
- 77 instrucciones.
- Desde 18 a 80 pines.
- Hasta 128 KBytes de programa (hasta 2 MBytes en ROM).
- Multiplicador hardware 8x8.
- Hasta 3968 bytes de RAM y 1 Kbyte de EEPROM.



- Frecuencia máxima de reloj de 40 MHz, hasta los 10 millones de microoperaciones por segundo (MIPs).
- Pila de 32 niveles de profundidad y 21 bit de ancho, que puede ser escrita y leída.
- Múltiples fuentes de interrupción.
- Periféricos de comunicación avanzados, como el CAN.

### 2.3.7.1- PIC 18F4680

En este apartado nos centraremos en la arquitectura y características del PIC utilizado para la realización de este proyecto.

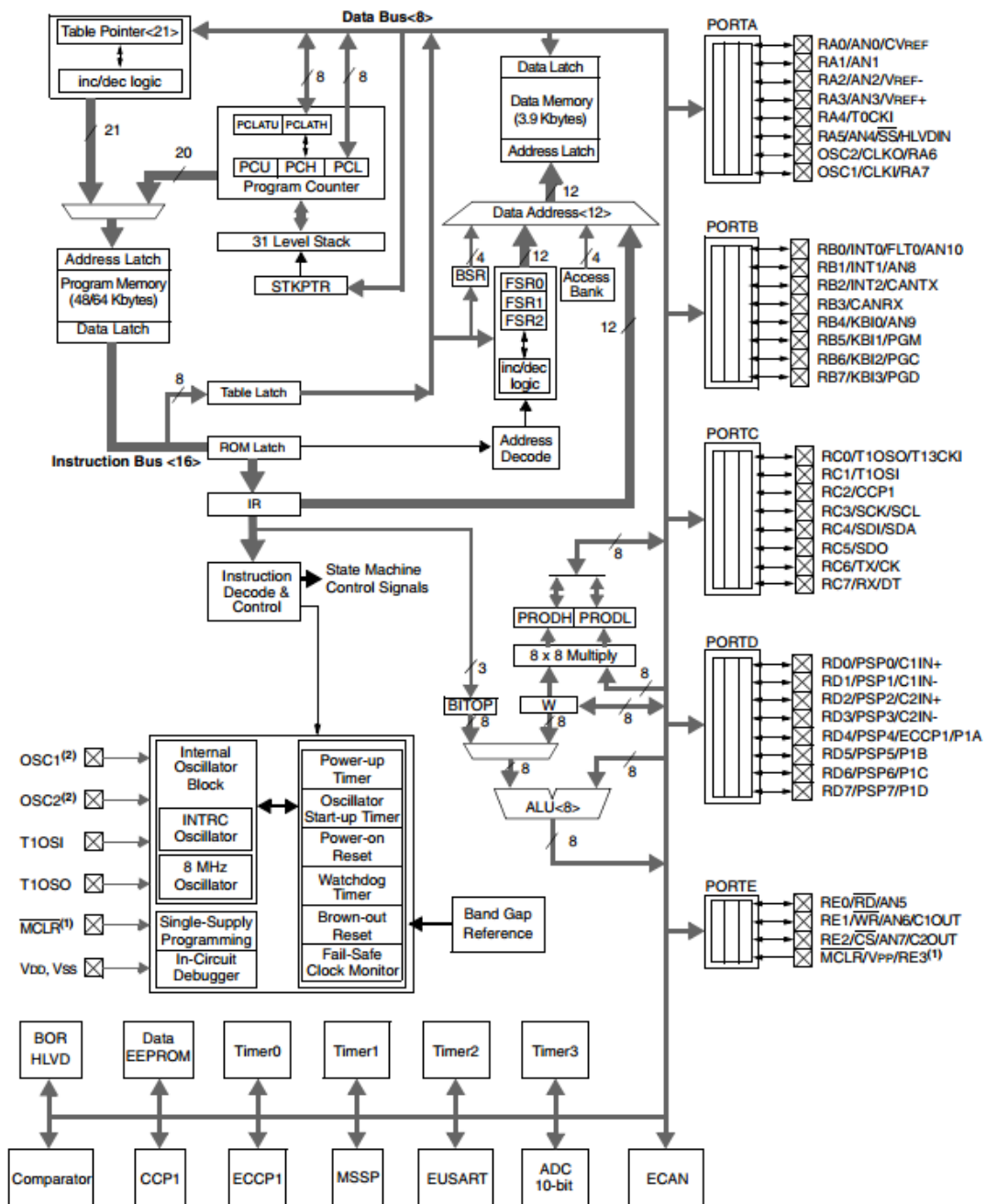


Figura 2.31. Arquitectura interna del PIC 18F4680

Como detalles a destacar, el dispositivo dispone de una memoria de programa Flash de acceso rápido de 64 KBytes, una memoria RAM de 3,9 Kbytes de tamaño y 12 bit de tamaño de palabra (4 bit de direccionamiento más 8 bit de offset en cada instrucción) con 3 registros de puntero de memoria o de selección de banco (FSR 0-2) de 12 bit cada uno y una memoria de datos EEPROM (data EEPROM) de 1 Kbyte.

Aparte, tenemos una ALU (Unidad Aritmético-Lógica) potente que puede realizar los cálculos que le vienen de un multiplicador 8x8 (ya incluido en el PIC17), 5 puertos configurables (A, B, C, D, E) de E/S, 4 temporizadores (Timer0-3) el primero de 8 bit y los restantes de 16, 1 módulo CCP que nos permite capturar, comparar y realizar una modulación por ancho de pulso (PWM), y otro ECCP (Enhanced CCP con 4 salidas síncronas) y un ADC (convertidor analógico-digital) de 10 bit.

También se puede ver como existe una pila de 31 niveles (31 level stack) que es una de las características innatas de esta gama de PICs.

Con el fin de ahorrar energía, el microcontrolador nos ofrece tres modos de funcionamiento:

- Modo **run**: CPU y periféricos on.
- Modo **idle**: CPU on y periféricos off.
- Modo **sleep**: Tanto la CPU como los periféricos están en off.

En el modo **idle**, la corriente baja hasta los **5.8  $\mu$ A** típicos y en el **sleep** hasta los **0.1  $\mu$ A**.

En esta familia de PICs, además existen 10 modos de oscilador diferentes que nos permiten un amplio rango de elección para cualquier aplicación hardware:

- 4 modos de cristal, usando cristales o resonadores cerámicos hasta los 40 MHz.
- 2 modos de reloj externo, dándonos la opción de usar dos pines o uno hasta los 40 MHz.
- 2 modos de osciladores RC, ofreciéndonos la misma opción que el apartado anterior hasta los 4 MHz.
- 1 oscilador interno de 8 MHz, con 6 modos de selección que van desde los 125 KHz hasta los 4 Mhz.
- 1 bloque de multiplicador de frecuencias PLL (Phase Lock Loop) disponible tanto como para los osciladores externos como los internos, que nos permite alcanzar velocidades de hasta 40 MHz. También permite la selección de frecuencias desde los 31 KHz hasta los 32 MHz.

Aparte de funcionar como reloj, el oscilador interno nos proporciona una referencia estable que da a esta familia características adicionales para operaciones fiables:

- **Fail-Safe Clock Monitor (FSCM)**: Si detecta un fallo en la fuente de reloj externa, automáticamente pasa al oscilador interno.
- **Arranque de dos velocidades**: Esta opción permite al oscilador interno servir como una fuente de reloj al reset o al modo sleep, hasta que la fuente primaria de reloj esté disponible.

Por último, se pueden observar módulos referidos a diferentes periféricos, como son el EUSART, MSSP y ECAN.

## Resumen de las especificaciones

Característica	Valor
Tipo de memoria de programa	Flash
Memoria de Programa (KB)	64
Velocidad CPU (MIPS)	10
RAM Bytes	3,328
EEPROM datos (bytes)	1024
Periféricos de comunicación digital	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Periféricos capturadores/comparadores/PWM	1 CCP (Capturador Comparador PWM), 1 ECCP (Enhanced CCP)
Temporizadores	1 x 8-bit, 3 x 16-bit
ADC (Conversores analógico-digital)	11 canales, 10-bit
Comparadores	2
CAN (Controller Area Network)	1 CAN
Rango de temperatura (°C)	-40 to 150
Rango de tensión (V)	2 to 5.5
Número de pines	40

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP1/ECCP1 (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I <sup>2</sup> C™			
PIC18F2585	48K	24576	3328	1024	28	8	1/0	Y	Y	1	0	1/3
PIC18F2680	64K	32768	3328	1024	28	8	1/0	Y	Y	1	0	1/3
PIC18F4585	48K	24576	3328	1024	44	11	1/1	Y	Y	1	2	1/3
PIC18F4680	64K	32768	3328	1024	40/44	11	1/1	Y	Y	1	2	1/3

Features	PIC18F2585	PIC18F2680	PIC18F4585	PIC18F4680
Operating Frequency	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	49152	65536	49152	65536
Program Memory (Instructions)	24576	32768	24576	32768
Data Memory (Bytes)	3328	3328	3328	3328
Data EEPROM Memory (Bytes)	1024	1024	1024	1024
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	1	1	1	1
Enhanced Capture/Compare/PWM Modules	0	0	1	1
ECAN Module	1	1	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Parallel Communications (PSP)	No	No	Yes	Yes
10-bit Analog-to-Digital Module	8 Input Channels	8 Input Channels	11 Input Channels	11 Input Channels
Comparators	0	0	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable High/Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

Figura 2.32. Comparativa entre PICs similares

Para el diseño de este proyecto se optó por usar la versión SMD (dispositivo de montaje superficial) de menor tamaño, con encapsulado TQFP.

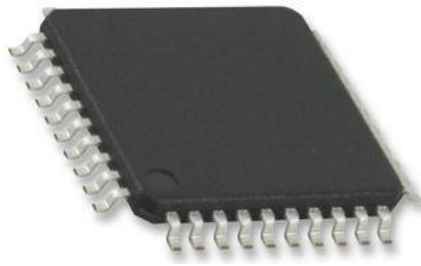


Figura 2.33. Encapsulado TQFP

#### 44-Pin TQFP

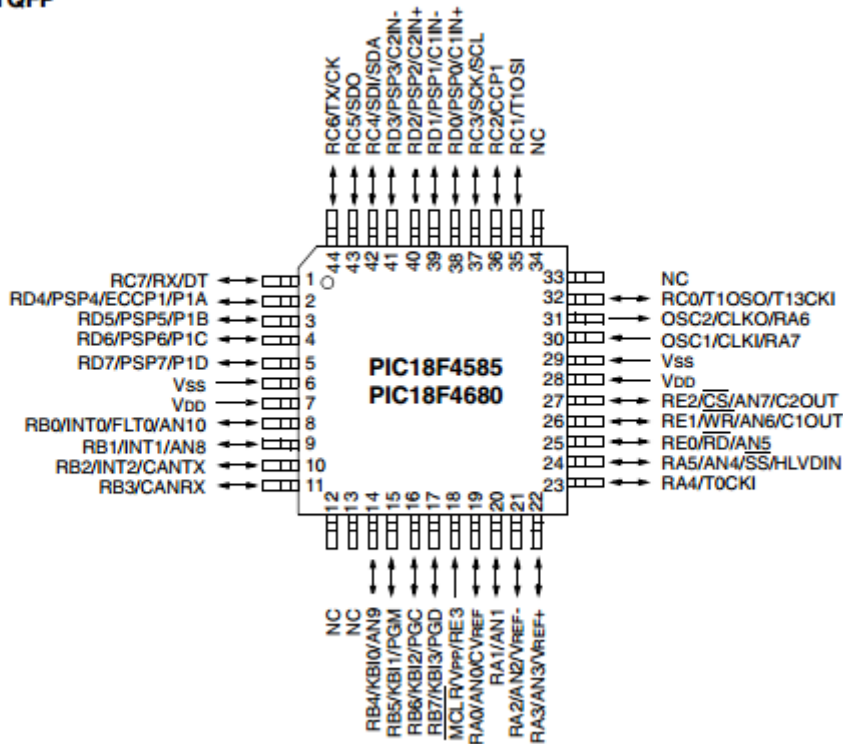


Figura 2.34. Distribución de pines PIC18F4680

Tenemos que los puertos no siguen un orden correlativo de pines y se dispersan a lo largo de los 44 totales de los que dispone el dispositivo.

**Pin 1** – Entrada y salida digital correspondiente al bit 7 del puerto C (RC7).

**Pines 2-5** – Entradas y salidas digitales, bits del 4 al 7 del puerto D (RD4-7). También están dedicados al ECCP, módulo capturador/comparador/ PWM con 4 salidas síncronas.

**Pines 6, 29** – Masa.

**Pines 7, 28** – Alimentación.

**Pines 9-11 – 14-17** – Puerto B de entrada y salida digital (RB0 hasta RB7).

**Pin 18** – Reset, tensión de programación y entrada digital correspondiente al bit 4 del puerto E (RE3).

**Pines 19-24** – Puerto A de entrada y salida digital (RA0 hasta RA5). Pines 20-22 y 24 entradas analógicas 1-4 (AN1-4). Pin 23 entrada de reloj del Timer 0. Los pines 21 y 22 se corresponden también con referencias de tensión (Vref– y Vref+).

**Pines 25-27** – Entradas y salidas digitales correspondientes a los bits 0-2 del puerto E, también se corresponden con las entradas analógicas del bit 6 al 8. (AN5-7).

**Pines 30-31** – Entradas y salidas digitales para los bits 7 y 8 del puerto A (RA6-7). También se utilizan para conectar un oscilador externo (OSC1-2).

**Pin 32** – Entrada y salida digital del bit 1 del puerto C (RC0).

**Pines 35-37** – Entradas y salidas digitales de los bits 2 al 4 del puerto C (RC1-3). Pin 35, entrada del oscilador del Timer 1 (T1OSI). Pin 36, entrada del módulo CCP (Capture/Compare/PWM).

**Pines 38-41** - Entradas y salidas digitales de los bits 1 al 4 del puerto D (RD0-3).

**Pines 42-44** - Entradas y salidas digitales de los bits 5 al 7 del puerto C (RC4-6).

---

## 3. DISEÑO DEL CIRCUITO

---

En este capítulo detallaremos las condiciones y los elementos que han intervenido en el desarrollo del diseño de este proyecto.

Dicho proyecto consistirá básicamente en una placa de circuito impreso, en cuyo extremo se encontrarán los LEDs controlados por un microcontrolador PIC, que rotará sobre si misma impulsada por un motor a modo de hélice y que dejará como estela una imagen previamente programada.

A tal dispositivo lo mencionaremos a lo largo de la memoria con el nombre de POV, cuyas siglas corresponden al efecto óptico que provoca (Persistence of Vision).

### 3.1– Especificaciones

Las principales especificaciones del dispositivo son:

- Microcontrolador: PIC18F4680 que contiene el código de la imagen.
- Alimentación del motor mediante una batería de 9V con tensión controlada.
- Alimentación de la placa principal regulada a 5V partiendo de una batería de 9V.
- Sensor que servirá como referencia del inicio de la imagen.
- Optimización del espacio de los componentes.
- Equilibrado de la placa mediante la repartición de pesos.

### 3.2– Diseño de la placa

Para el diseño se ha tenido en cuenta que para la correcta visualización de la imagen y así tener recorrido la placa principal tuviera una forma rectangular, como también que no fuera de grandes dimensiones para que así el motor tuviera la fuerza suficiente como para poder moverla.

#### 3.2.1.- Esquemático

Empezaremos el diseño de la placa haciendo el esquemático de la misma con el programa OrCAD Capture, que se puede encontrar en el paquete de OrCAD.

Para ello, crearemos un nuevo proyecto con la opción *Schematic*, ya que únicamente queremos hacer el “mapa” sin una posterior simulación.



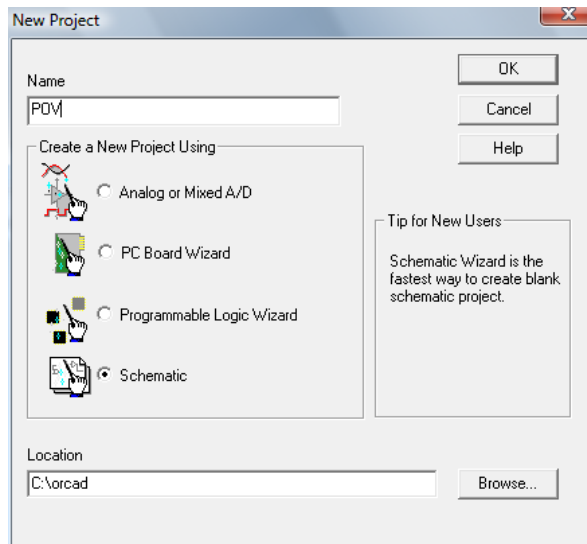


Figura 3.1. Configuración proyecto OrCAD Capture

Los componentes necesarios para el diseño de la placa principal los encontraremos en las respectivas librerías que nos proporciona el mismo programa.

- **1 PIC 18F4680**, parte editada a partir de un PIC encontrado en la librería *microcontroller.olb*.
- **1 regulador de tensión 7805** en la librería *opamp.olb/LM7805C*.
- **24 resistencias de 100 ohmios y 2 de 10k** de la librería *discrete.olb/R*.
- **4 condensadores de 100nF** de la librería *discrete.olb/C*.
- **1 sensor Hall** editado a partir de un bloque similar.
- **1 conector de 2** para la alimentación de la placa, encontrado en *connector.olb/CON2*.
- **1 conector de 6** para la programación del PIC, *connector.olb/CON6*.
- **2 conectores de 16** para la posterior colocación de una plaquita dedicada a los LEDs RGB, *connector.olb/CON16*.

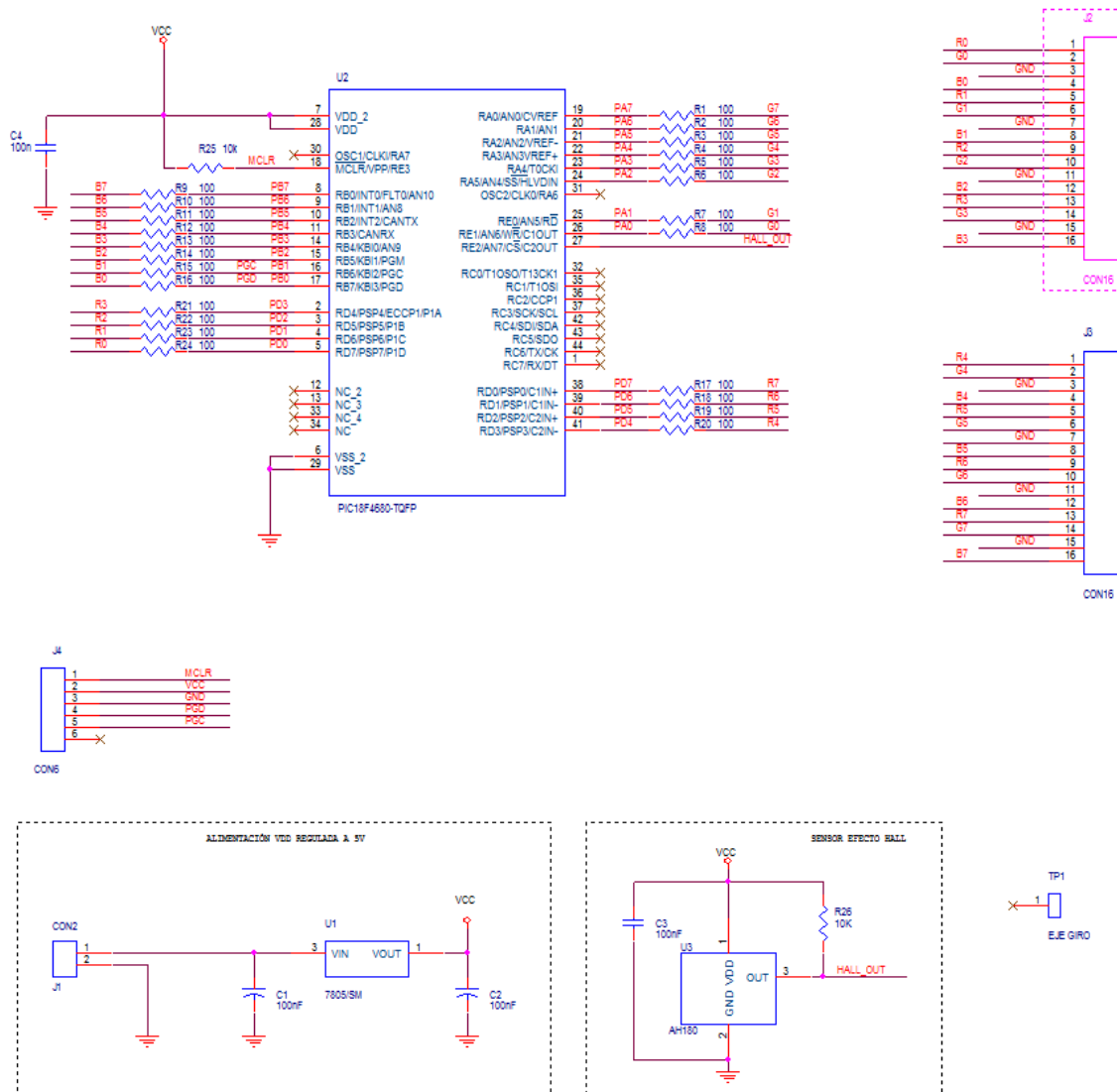


Figura 3.2. Esquemático placa principal

Los bloques bien diferenciados del circuito son:

- Regulación de la tensión de entrada a 5V.
- Microcontrolador PIC18F4680 para la parte de la programación.
- Sensor hall que indicará al microcontrolador cuando tiene que empezar a dibujar.
- 8 LED's RGB de montaje superficial que convertirá el código programado en una imagen visible.

### 3.2.1.1. – Efecto Hall

Para explicar el funcionamiento del sensor Hall que hemos incorporado en el circuito principal como referencia para el inicio de la imagen, primeramente vamos a exponer los fundamentos teóricos en los cuáles se basa este componente.

En un conductor por el que circula una corriente, en presencia de un campo magnético perpendicular al movimiento de las cargas, aparece una separación de cargas que da lugar a un campo eléctrico en el interior del conductor, perpendicular al movimiento de las cargas y al campo magnético aplicado. A este

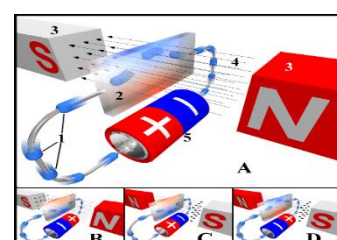


Figura 3.3. - Efecto Hall

campo eléctrico se le denomina **campo Hall**. Llamado efecto Hall en honor a su modelador **Edwin Herbert Hall**.

En épocas contemporáneas (1985) el físico alemán **Klaus von Klitzing** y colaboradores descubrieron el hoy conocido como **efecto Hall cuántico** que les valió el premio Nobel de Física en 1985. En 1998, se otorgó un nuevo premio Nobel de Física a los profesores Laughlin, Strömer y Tsui por el descubrimiento de un nuevo fluido cuántico con excitaciones de carga fraccionarias. Este nuevo efecto ha traído grandes problemas a los físicos teóricos y hoy en día, constituye uno de los campos de investigación de mayor interés y actualidad en toda la física del estado sólido.

El **sensor Hall** se sirve de este para la medición de **campos magnéticos** o **corrientes** o para la determinación de la posición.

Si fluye corriente por un sensor Hall y se aproxima a un campo magnético que fluye en dirección vertical al sensor, entonces el sensor crea un **voltaje saliente** proporcional al producto de la **fuerza del campo magnético** y de la corriente. Si se conoce el valor de la corriente, entonces se puede calcular la fuerza del campo magnético; si se crea el campo magnético por medio de corriente que circula por una bobina o un conductor, entonces se puede medir el valor de la corriente en el conductor o bobina.

Aparte de para la función que nos ocupa en este proyecto, el sensor Hall puede medir:

- La movilidad de una partícula cargada eléctricamente (electrones, lagunas, etc).
- Los campos magnéticos (teslámetros)
- La intensidad de corrientes eléctricas (sensores de corriente de Efecto Hall).
- También permiten la elaboración de sensores o detectores de posición sin contacto, utilizados particularmente en el automóvil, para detectar la posición de un árbol giratorio (árbol de levas, caja de cambios, paliers, etc.).
- Encontramos también sensores de efecto Hall bajo las teclas de los teclados de los instrumentos de música modernos (órganos, órganos digitales, sintetizadores) evitando así el desgaste que sufren los contactos eléctricos tradicionales.
- Encontramos sensores de efecto Hall en el codificador de un motor de CD.
- Los motores de Efecto Hall (HET) son aceleradores de plasma de gran eficacia.

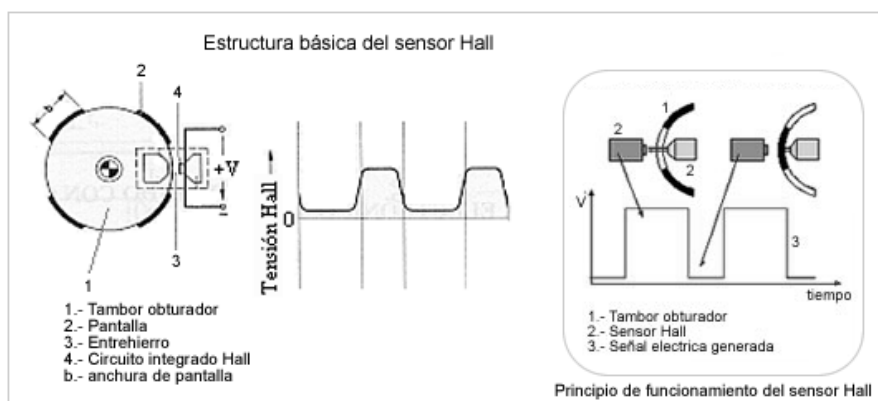


Figura 3.4. Estructura del sensor Hall

El sensor Hall que utilizaremos en este proyecto será uno de inserción ideal para diseños que necesiten una rápida respuesta. En el circuito que nos ocupa, ya que la velocidad mínima de rotación de la placa para que se pueda visualizar bien la imagen es del orden de 25-30 rpm (revoluciones por minuto), necesitaremos uno cuyo período sea igual o inferior a ese dato.

El modelo elegido que cumple con el requisito anterior es el SS411P de Honeywell, bipolar y con resistencia de pull-up incorporada, que se puede usar en múltiples aplicaciones que necesiten un dispositivo de rápida respuesta, como son los tacómetros, ventiladores o motores.

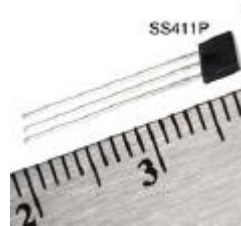


Figura 3.5. Tamaño sensor Hall

Además, teniendo en cuenta que alimentamos la placa a 5V, cuenta con un rango de tensión que va desde los 2.7 hasta los 7V, ideal para nuestro propósito.

Tal componente se ensamblará en la placa con un conector hembra de 3 pines para así tener una fácil extracción en el caso en que falle.

### 3.2.1.2. – Regulador de tensión

Para alimentar la placa principal necesitaremos una tensión de 5V. Al no existir baterías a esa tensión, la inmediatamente superior a ese valor es una de 9V, la cual bajaremos a la tensión requerida mediante un **regulador de tensión**.

Un regulador de tensión no es más que un componente electrónico diseñado para mantener un nivel de voltaje constante. Se encuentran en dispositivos como las **fuentes de alimentación** de los computadores, donde estabilizan los voltajes DC usados por el procesador y otros elementos. En los **alternadores** de los automóviles y en las plantas generadoras, los reguladores de voltaje controlan la salida de la planta. En un sistema de **distribución de energía eléctrica**, los reguladores de voltaje pueden instalarse en una subestación o junto con las líneas de distribución de forma que todos los consumidores reciban un voltaje constante independientemente de que tanta potencia exista en la línea.

El utilizado en este circuito es la versión SMD (montaje superficial) del modelo 7805.



Figura 3.6. Regulador de tensión 7805 SMD

### 3.2.1.3. – Diodos LED RGB

Un LED es un componente optoelectrónico pasivo, más concretamente, un diodo que emite luz. La palabra española “led” proviene del acrónimo inglés LED (Light-Emitting Diode: “diodo emisor de luz”).

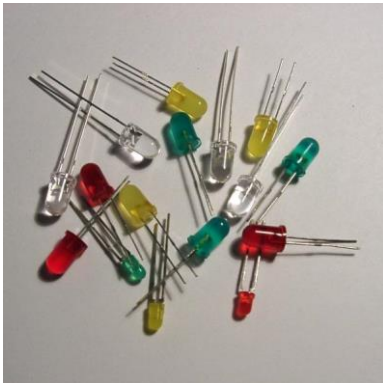


Figura 3.7. LEDs monocromo

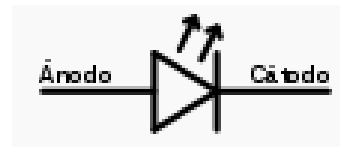


Figura 3.8. Símbolo electrónico LED

Los LEDs se usan como indicadores en muchos dispositivos y en iluminación. Los primeros LEDs emitían luz roja de baja intensidad, pero los dispositivos actuales emiten luz de alto brillo en el espectro infrarrojo, visible y ultravioleta.

Debido a sus altas frecuencias de operación son también útiles en tecnologías avanzadas de comunicaciones. Los LEDs infrarrojos también se usan en unidades de control remoto de muchos productos comerciales incluyendo televisores e infinidad de aplicaciones de hogar y consumo doméstico.

Los LEDs presentan muchas ventajas sobre las fuentes de luz incandescente y fluorescente, principalmente por el bajo consumo de energía, mayor tiempo de vida, tamaño reducido, durabilidad, resistencia a las vibraciones, reducen la emisión de calor, no contienen mercurio (el cual al exponerse en el medio ambiente es altamente venenoso), en comparación con la tecnología fluorescente, no crean campos magnéticos altos como la tecnología de inducción magnética, con los cuales se crea mayor radiación residual hacia el ser humano; cuentan con mejor índice de producción cromática que otros tipos de luminarias, reducen ruidos en las líneas eléctricas, son especiales para utilizarse con sistemas fotovoltaicos (paneles solares) en comparación con cualquier otra tecnología actual; no les afecta el encendido intermitente (es decir pueden funcionar como luces estroboscópicas) y esto no reduce su vida promedio, son especiales para sistemas antiexplosión ya que cuentan con un material resistente, y en la mayoría de los colores (a excepción de los LEDs azules), cuentan con un alto nivel de fiabilidad y duración.

También tienen la ventaja de poseer un tiempo de encendido muy corto (aproximadamente en un cuarto de segundo) en comparación con las luminarias de alta potencia como lo son las luminarias de alta intensidad de vapor de sodio, aditivos metálicos, halogenuro o halogenadas y demás sistemas con tecnología incandescente.

La excelente variedad de colores que producen los LEDs ha permitido el desarrollo de nuevas pantallas electrónicas de texto monocromáticas, bicolores, tricolores y RGB (pantallas a todo

color) con la habilidad de reproducción de vídeo para fines publicitarios, informativos o tipo indicadores.

Como nuestra pretensión es extraer una imagen de más de un color, en detrimento de los diodos LED monocromáticos, utilizaremos los LED RGB.

El nombre de LED RGB viene su nomenclatura del inglés Red, Green, Blue (rojo, verde, azul). Se trata de un sistema que se basa en la adición de colores para poder representar cualquier color. La mezcla se realiza a través de estos tres colores primarios comentados.



Figura 3.9. Tira de LEDs RGB

Está muy de moda en la decoración del hogar y en las zonas de ocio el utilizar tiras LED con este sistema para crear distintos ambientes y jugar con la iluminación.

Estos dispositivos LED tienen internamente tres chips. Uno es el chip LED de color rojo, otro es el de color verde y otro el de color azul. De esta manera, el dispositivo puede cambiar de color en función de la intensidad que reciba cada chip. O sea, se puede encender todos, ninguno, o la mezcla de ellos.

Por ello necesitan de un controlador para ajustar el color de una lámpara o un circuito con varios dispositivos como son las tiras LED RGB. Si aplicáramos tensión a un dispositivo RGB sin control funcionarían los tres circuitos y obtendríamos la luz blanca únicamente.

Existen de cátodo común (conectados a masa) y de ánodo común (a alimentación). Los 8 LEDs utilizados en este proyecto serán de **cátodo común**.

El modelo utilizado en este caso es uno de montaje superficial, concretamente el OVS-5309 de la empresa MULTICOMP con 5 mm. de diámetro y una alta luminosidad, aguantando una intensidad de 100 mA.

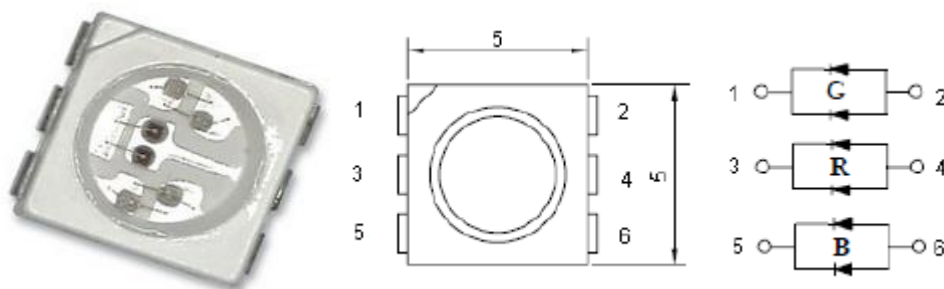


Figura 3.10. Estructura LED RGB SMD

Las tensiones para cada uno de los colores será de 3,2V (máx. 4V) para el azul y el verde, ambos utilizando como material el nitruro de galio-indio (InGaN) sobre un sustrato de zafiro, y

de 2,1V (máx. 2,6V) para el rojo, que utiliza el fosforo de aluminio-galio-indio (AlGaInP) sobre un sustrato de silicio (Si), como se puede observar en la siguiente tabla de especificaciones del componente.

LED Chip			Lens Colours	Dominant Wavelength (nm) at 20 mA	Luminous Intensity (mcd) at 20 mA		Forward Voltage (V) at 20 mA		Viewing Angle $2\theta^{1/2}$ (°)	Part Number
Material	Emitted Colours	Brightness			Minimum	Typical	Typical	Maximum		
InGaN / Sapphire	Blue	Mega	Water Clear	465	200	430	3.2	120	OVS-5309	
	True Green			520	700	1,460	3.2			
AlGaInP / Si	Red			625	500	1,010	2.1			2.6

Figura 3.11. Tabla de especificaciones del LED RGB

También se puede observar cómo, alimentando el componente a una intensidad de 20 mA, el color que más brillo nos da es el verde alcanzando un valor mínimo de 700 mcd (milicandelas) y uno típico de 1.460.

El diagrama de radiación nos indica cuál es la dispersión de la luminosidad del mismo, dejándonos ver una alta precisión y desviándose poco del punto de mira.

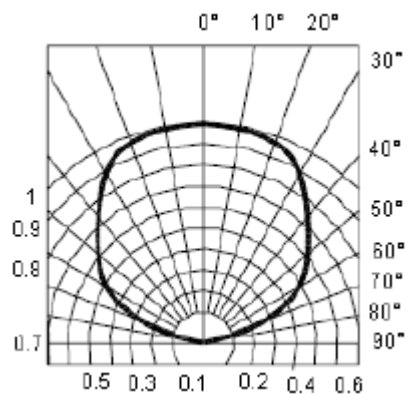


Figura 3.12. Diagrama de radiación del LED RGB

Para la implementación de tales diodos en la placa principal se ha procedido a la realización de una placa complementaria que dieran a los mismos una orientación vertical para así dejar como estela una pantalla al rotar.

A continuación, nos encontraremos con el diseño de la misma, dividido en dos partes, un adaptador que irá conectado a la placa y una placa de LEDs que ira soldada al adaptador en posición vertical.



## ADAPTADOR

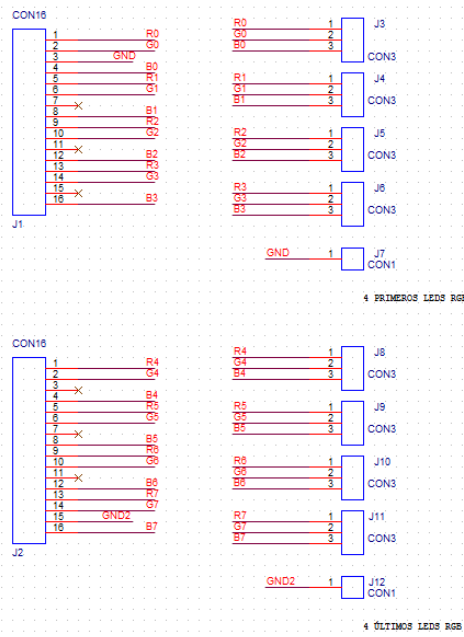


Figura 3.13. Esquemático del adaptador LED a placa

## PLACA DE LEDS

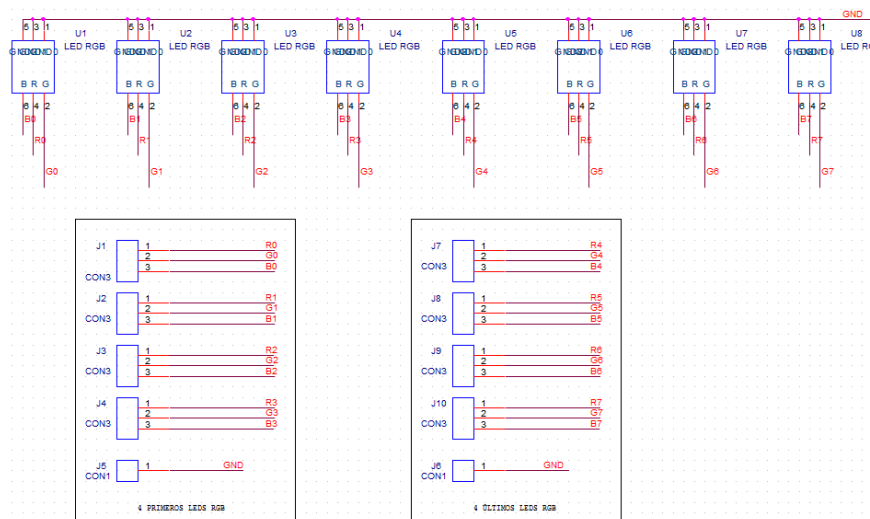


Figura 3.14. Esquemático de la placa de LEDs RGB

Podemos observar como se ha implementado un sistema de masa por cada 4 LEDs para así resultar más fácil la realización del circuito impreso en el Layout, en detrimento de una masa para cada uno de los 8 componentes.

### 3.3– PCB

Para la realización de la PCB utilizaremos el OrCAD Layout, incluido en la misma suite que el programa en el que hemos realizado el esquemático.

Partiremos del esquemático realizado en el Capture y lo pasaremos al programa mencionado mediante una opción existente dentro de *Tools* llamada *Create netlist*, que aparecerá al seleccionar la página del esquemático.

Dentro de *Create netlist*, le daremos a la pestaña *Layout* y marcaremos las opciones *Run ECO to Layout* (para que los cambios que realicemos en el Capture aparezcan también en el Layout) y también *User properties are in inches*, ya que las medidas de los componentes se encuentran en *inches* (pulgadas).

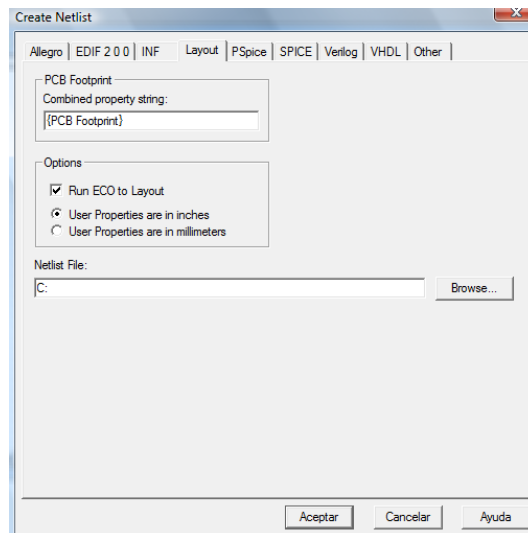


Figura 3.15. Configuración netlist OrCAD Capture

Ya en el Layout, le daremos a *File-New* y se nos abrirá una nueva ventana. En ella seleccionaremos el tipo de tecnología o conjunto de características con extensión *\*.tch* que queremos implementar en el diseño del circuito impreso. En este caso, seleccionaremos las opciones por defecto del programa *default.tch*. En el siguiente campo, seleccionaremos el archivo con extensión *\*.mnl* que se nos generó al hacer el Netlist en el OrCAD Capture y, por último, le indicaremos en qué ruta queremos que guarde el archivo *\*.max* que contendrá el proyecto que desarrollaremos en el OrCAD Layout.

Hecho esto, le daremos a *Apply ECO*.

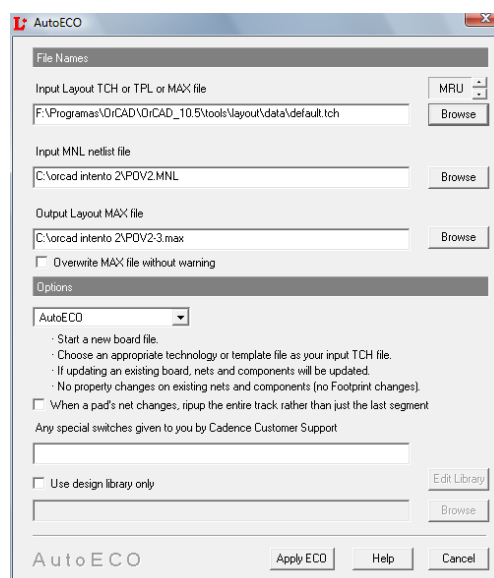


Figura 3.16. Configuración PCB OrCAD Layout

A continuación, sólo en el caso en el que no hayamos asignado encapsulado a cada componente en el Capture, nos dará la opción de elegirlos.

Las características del diseño de las siguientes dos placas que vamos a comentar a continuación (la placa principal y la placa de los LED RGB) son las siguientes:



- Sólo se rutarán las capas **TOP** y **BOTTOM** (opción Routing en  *View Spreadsheet-Layers*).
- Los valores de las anchuras de las pistas serán de (dentro de  *View Spreadsheet-Nets*):
  - **Anchura mínima** (Min width): 12 mils.
  - **Ancho de la conexión** (Conn width): 15 mils.
  - **Anchura máxima** (Max width): 25 mils.




Figura 3.17. Configuración anchura pistas OrCAD Layout

- En *Options-Global Spacing*, indicaremos las separaciones entre pistas o entre vías con un valor de 12 mils en todos sus campos.

Layer Name	Track to Track	Track to Via	Track to Pad	Via to Via	Via to Pad	Pad to Pad
TOP	12	12	12	12	12	12
BOTTOM	12	12	12	12	12	12
GND	12	12	12	12	12	12
POWER	12	12	12	12	12	12

Figura 3.18. Configuración separaciones OrCAD Layout

- Para designar los límites de la placa, usaremos la opción *Obstacle*, que se encuentra dentro de *Tool-Obstacle* de la barra de herramientas o directamente desde el acceso directo *Obstacle Tool* .

En este caso, seleccionaremos en *Obstacle type* la opción **Detail** con un **ancho (width) de 20 mils**.




Figura 3.19. Configuración borde PCB OrCAD Layout

### 3.3.1.- Placa principal

En este apartado expondremos los encapsulados de cada componente que interviene en la placa principal y las características de sus pads.

Medidas de los pads en los componentes de inserción:

- **25x resistencias de inserción (24 de 100 ohmios y 1 de 10k)**: Round de 80 mils.
- **Conector hembra de 3 pines (sensor Hall)**: Round de 62 mils.
- **Conector molex macho acodado de 6 pines para programación**: Oblong de 65x100 mils.

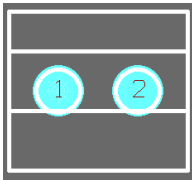
- **2x conector de 16 pines para placa LEDs RGB:** Round de 65 mils.
- **Conector 2 pines para alimentación placa:** Round de 120 mils.
- **Regulador tensión 7805 SMD:** Rectangle de 32x64 mils.
- **PIC 18F4680 SMD:** Oblong de 100x18 mils.
- **3x condensador 100nF SMD:** Square de 50 mils.
- **Resistencia 10k SMD (sensor Hall):** Square de 50 mils.
- **Taladro (Drill):** Round de 200 mils.

En cuanto a los encapsulados para cada componente:

- **1 PIC 18F4680 SMD**, al que asignaremos el encapsulado *QUAD.80M/44/WG14.15*
- **1 regulador de tensión 7805 SMD**, *SM/SOT89\_123*
- **24 resistencias de 100 ohmios y 1 de 10k de inserción**, *AX/.500X.100/.031*
- **1 resistencia de 10k SMD**, *SM/R\_0805*
- **4 condensadores de 100nF SMD**, *SM/C\_0805*
- **1 sensor Hall**, *BLKCON.100/VH/TM1SQS/W.100/3.*
- **1 conector de 2**, encapsulado personalizado *REGLETA\_2\_POLOS.*
- **1 conector molex macho acodado de 6 pines**, encapsulado personalizado *TIRA POSTE C.I. MACHO ACODADO\_6P.*
- **2 conectores de 16**, *BLKCON.100/VH/TM1SQ/W.100/16.*
- **Taladro central**, encapsulado personalizado *PIE\_PLASTICO.*

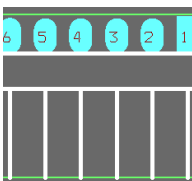
### Encapsulados personalizados

- **REGLETA\_2\_POLOS**



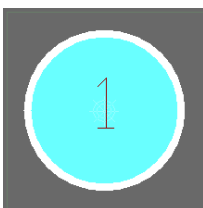
200 milésimas de pulgada (mils) entre pines con pads redondeados (Round) de 120 mils.

- **TIRA POSTE C.I. MACHO ACODADO\_6P**



100 mils entre pines con pads ovalados (Oblong) de 65x100 mils.

- **PIE\_PLASTICO**



Un único pad redondeado de 200 mils dedicado para el eje central.

Una característica de la placa principal es la creación de un plano de masa en las dos capas (TOP y BOTTOM) al que irán todas las masas de los componentes del circuito, facilitando así el rutado del mismo.

Para ello y utilizando la herramienta *Obstacle Tool*, elegiremos la opción **Copper pour** en *Obstacle type*, con un ancho de 10 mils, quedando cubiertas las dos capas de cobre.

Una vez posicionados los componentes de la mejor manera posible y delimitada la placa, nos quedará así:

### **CAPA TOP**

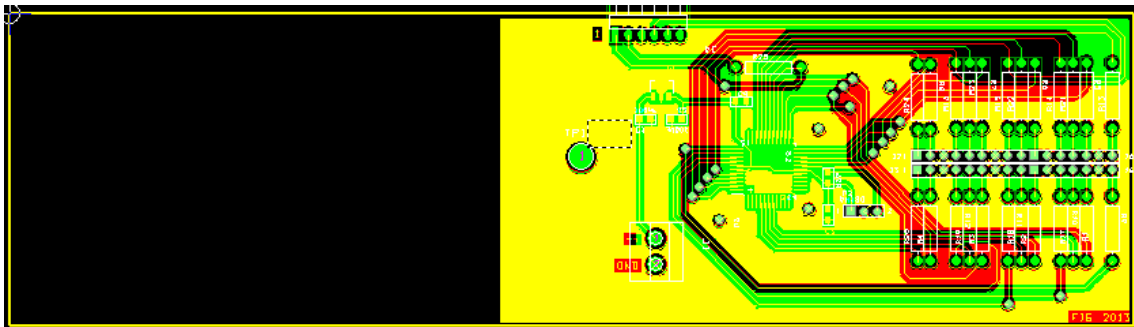


Figura 3.20. Vista layout capa top placa principal

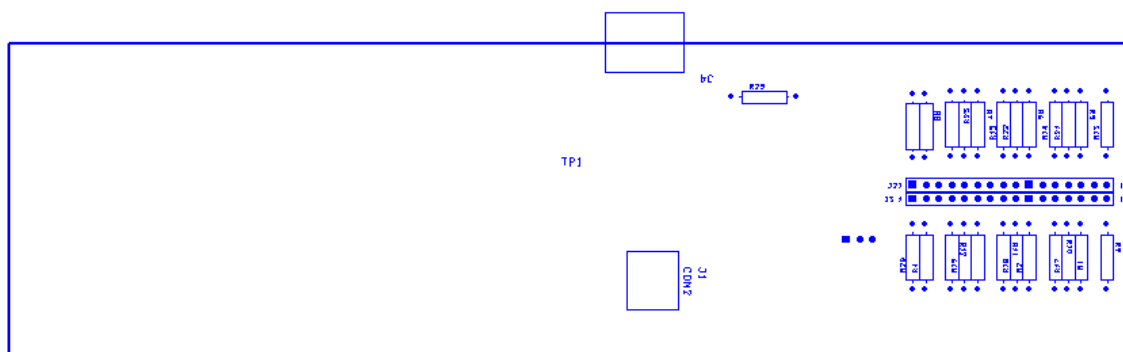


Figura 3.21. Vista ensamblaje capa top placa principal

Se encuentran los 2 conectores de 16 (placa LEDs), el de 6 pines (programación del PIC), el de 2 (alimentación), y 24 resistencias de 100 ohmios y una de 10k.

### **CAPA BOTTOM**

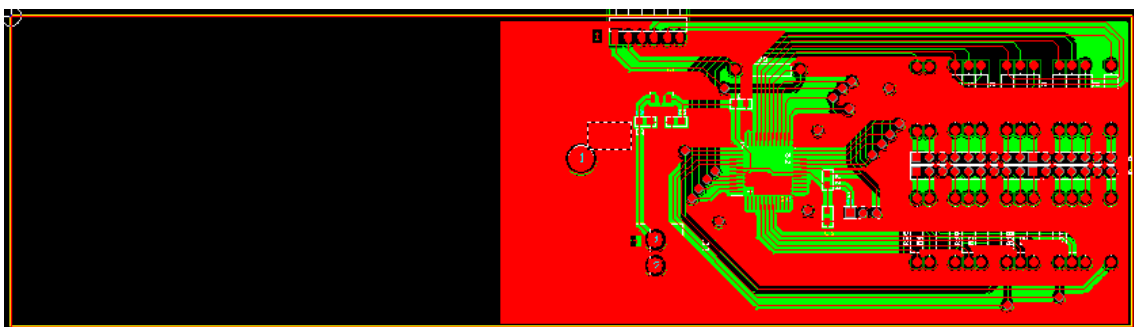


Figura 3.22. Vista layout capa bottom placa principal

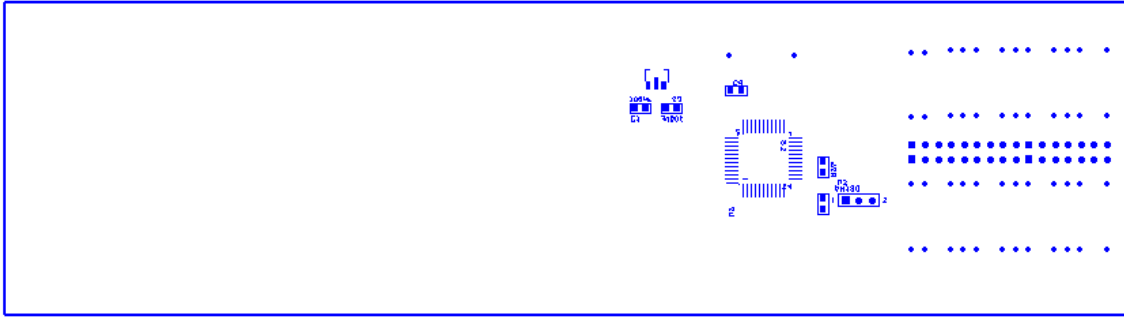


Figura 3.23. Vista ensamblaje capa bottom placa principal

Tenemos el regulador de tensión 7805 con sus condensadores de 100 nF, el PIC 18F4680 y el sensor Hall con sus resistencias de 10k.

Se puede observar como la placa es más larga que la superficie útil de componentes, tema que ya se pensó anteriormente al diseño de la misma para que al rotar la placa tuviera el diámetro suficiente como para mostrar la imagen correctamente.

Las dimensiones de la misma son de 8675 mils de largo (22,03 cm.) por 2425 de alto (6,16 cm.).

### 3.3.2.- Placa LEDs

La placa de los LEDs RGB se dividirá en 2 partes (adaptador y placa donde irán los LEDs para facilitar su rutado), que luego se encajarán como si de un puzzle se tratase. Una vez soldadas estas dos partes, la placa resultante irá conectada a la placa principal en los dos conectores de 16 de que dispone mediante un conector macho acodado con dos tiras de 16 pines.

Los encapsulados que intervienen y el tamaño de sus pads son:

#### Para el adaptador:

- **2 conectores de 16:** Encapsulado *BLKCON.100/VH/TM1SQ/W.100/16* con pads ovalados (oblong) de 62x85 mils.
- **26 pads rectangulares de 60x200 mils**, que irán en el extremo del adaptador y que servirán para soldar en ellos la placa dónde irán conectados los LEDs.

#### Para la placa de los LEDs:

- **26 pads rectangulares de 60x200 mils**, que irán soldados al adaptador anterior.
- **8 LEDs RGB SMD**, encapsulado personalizado RGB SMD con unas dimensiones de 5x5 mm., distancia entre pines de 1,5 mm. y pads rectangulares (rectangle) de 40x60 mils.

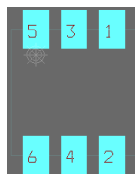


Figura 3.24. Encapsulado LED RGB

Una vez rutadas las dos placas, quedarían de la siguiente manera:

## ADAPTADOR

### CAPA TOP

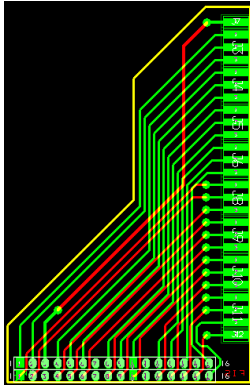


Figura 3.25. Vista layout capa top adaptador

### CAPA BOTTOM

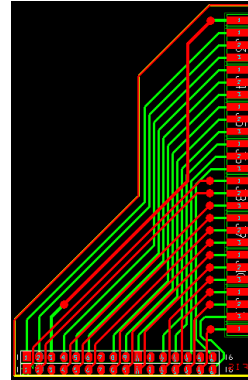


Figura 3.26. Vista layout capa bottom adaptador

## PLACA LEDs

### CAPA TOP

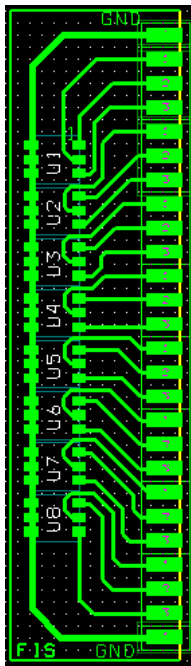


Figura 3.27. Vista layout capa top placa LEDs RGB

### CAPA BOTTOM

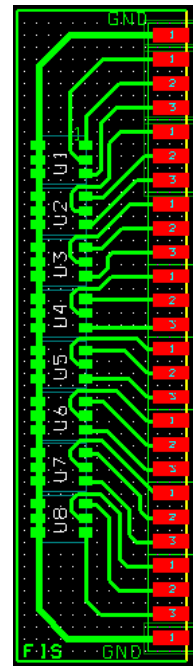


Figura 3.28. Vista layout capa bottom placa LEDs RGB

## 3.4– Realización del circuito impreso

Una vez ya tenemos todas las placas diseñadas, sólo queda imprimirlas y realizar el circuito impreso de cada una de ellas.

El proceso de imprimirlas siguiendo unas características dadas se realiza en la opción **Post Process Settings** que se encuentra dentro de *Options*.

Imprimiremos en papel de cebolla las capas **TOP** y **BOTTOM** (sólo la TOP en el caso de la placa de los LEDs).



Plot output File Name	Batch Enabled	Device	Shift	Plot Title
*.TOP	Yes	PRINT MANAGER	No shift Rot 90 Mir	Top Layer
*.BOT	Yes	PRINT MANAGER	No shift Rot 90	Bottom Layer

Figura 3.29. Tabla Post Process Settings OrCAD Layout

Vemos como la capa TOP irá impresa en modo espejo de 90º (Mir).

Dentro de cada capa, marcaremos las siguientes opciones:

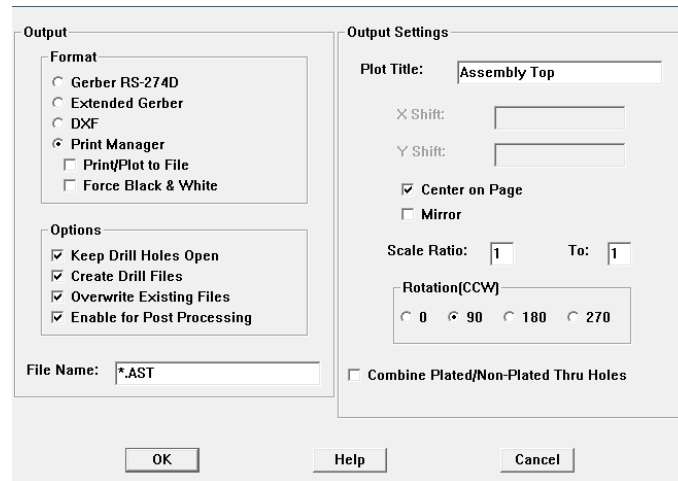


Figura 3.30. Configuración capa Post Process Settings OrCAD Layout

Como opciones importantes tenemos el *Print Manager*, para imprimir la capa seleccionada, y *Keep Drill Holes Open*, para mantener abiertos los taladros en los pads.

Marcaremos *Mirror* en el caso de la capa TOP.

---

## 4. DESARROLLO Y MONTAJE

---

En este apartado veremos el resultado final de las placas impresas y el montaje de las partes restantes, como la parte de potencia necesaria para impulsar la placa.

### 4.1.- Placa rotatoria

Una vez impresos en placa los circuitos (placa principal y placa de los LEDs) y soldados los componentes nos quedará algo así:

#### PLACA PRINCIPAL

##### CARA TOP

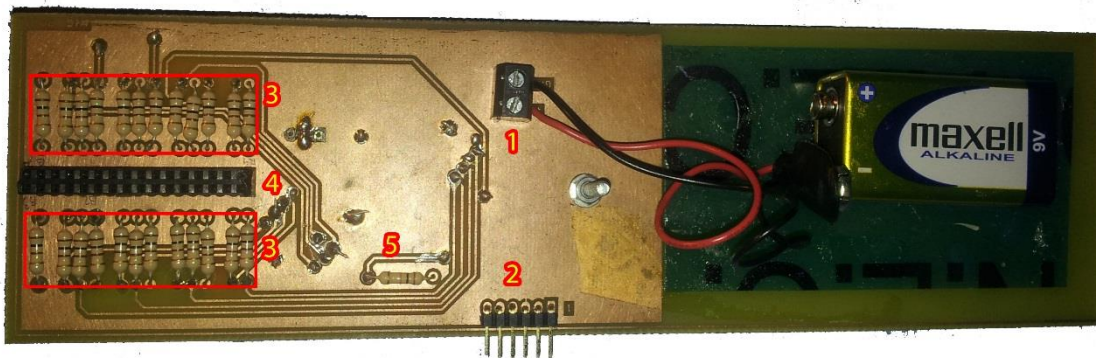


Figura 4.1. Vista superior placa principal o rotatoria

1. Conector de alimentación al que está conectado una batería de 9V.
2. Conector macho acodado de 6 pines para programar el PIC.
3. 2 filas de 12 resistencias de 100 ohmios cada una.
4. Conector hembra de 32 pines para conectar la plaquita de LEDs.
5. Resistencia de 10k.

##### CARA BOTTOM

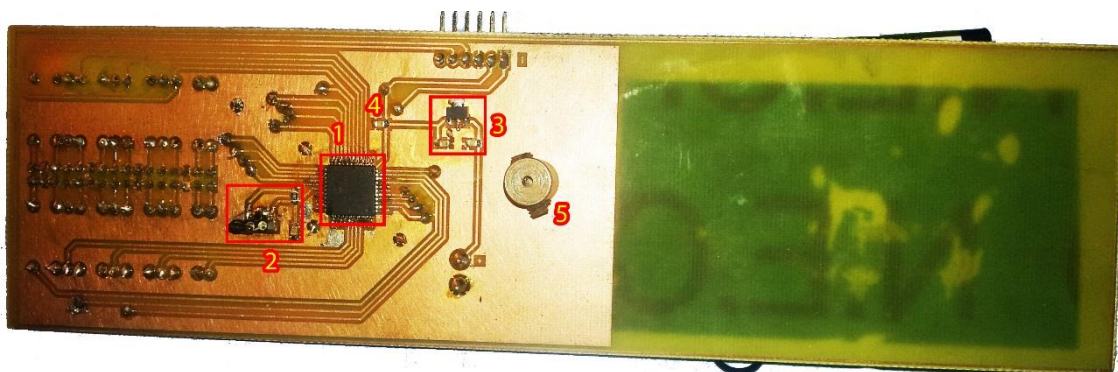


Figura 4.2. Vista inferior placa principal o rotatoria

1. PIC 18F4680.
2. Circuito del sensor Hall.
3. Circuito del regulador de tensión 7805, que convierte la tensión de entrada (9V) en 5V para el resto del circuito.
4. Condensador de 100nF que va a masa.
5. Soporte que conecta la placa principal con el motor que explicaremos más adelante.

Como se puede observar en las dos caras, la masa está representada por la cobertura de cobre que ya introdujimos en el Layout, *copper pour*.

### ADAPTADOR LEDS RGB

#### CAPA TOP

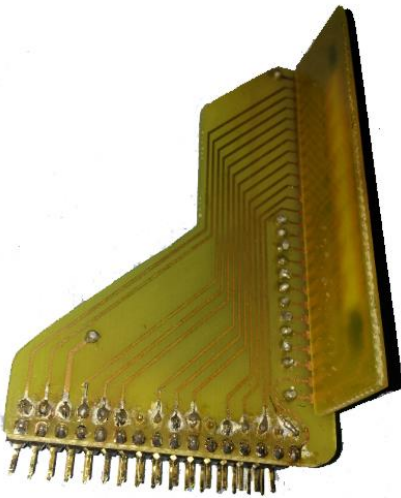


Figura 4.3. Vista superior adaptador LEDs RGB

#### CAPA BOTTOM

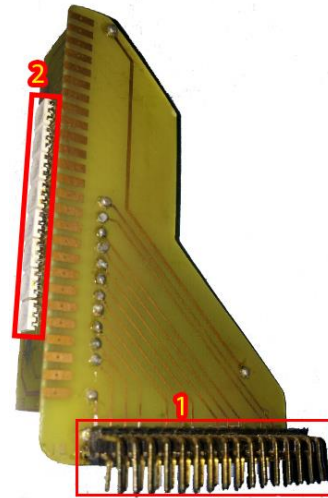


Figura 4.4. Vista inferior adaptador LED's RGB

1. Conector macho acodado con 2 filas de 16 pines cada una (SIL/DIL pin headers) que irá conectado al conector hembra de la placa principal.
2. 8 LEDs RGB SMD.

### PLACA LEDS

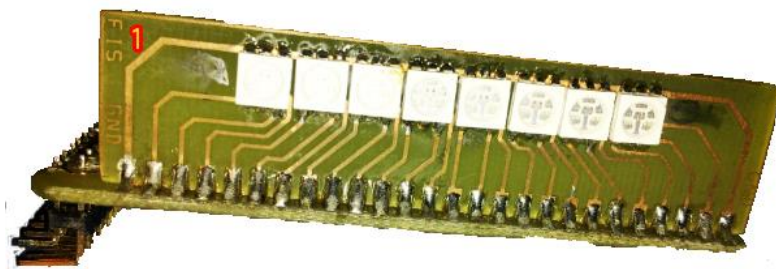


Figura 4.5. Placa LEDs RGB

1. Masa común a todos los LEDs a cuya pista irán soldados los 3 pines de cada LED correspondientes a los cátodos de cada diodo interno.

Ya teniendo disponibles todas las placas necesarias para el desarrollo de este proyecto, procederemos a montarlas para tener la parte útil del dispositivo.

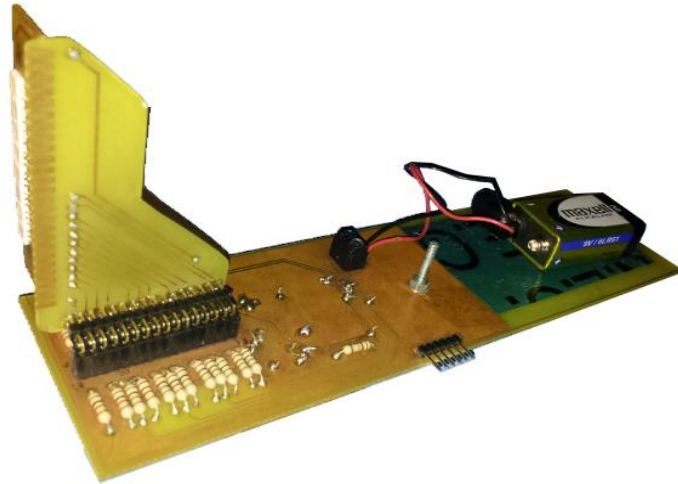


Figura 4.6. Placa rotatoria completa

En la anterior foto podemos ver como aparece conectado el adaptador de los LEDs RGB en la cara top de la placa principal.

En cuanto a la cara bottom, detallaremos como se montaron dos de los componentes más importantes:

### SENSOR HALL SMD

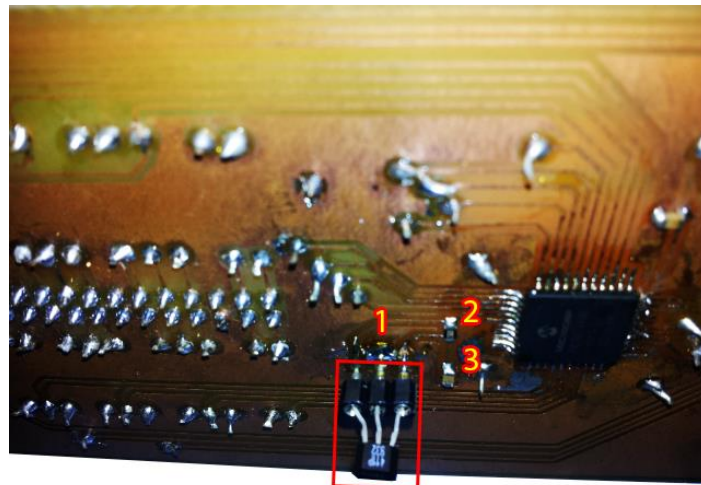


Figura 4.7. Circuito sensor Hall cara bottom

1. Conector hembra de 3 pines al que va conectado el sensor Hall de inserción.
2. Resistencia SMD de 10k.

3. Condensador SMD de 100nF.

El imán que excitará a dicho sensor será uno de neodimio que irá incorporado en el bloque del motor que explicaremos más adelante.

### REGULADOR DE TENSIÓN 7805 SMD

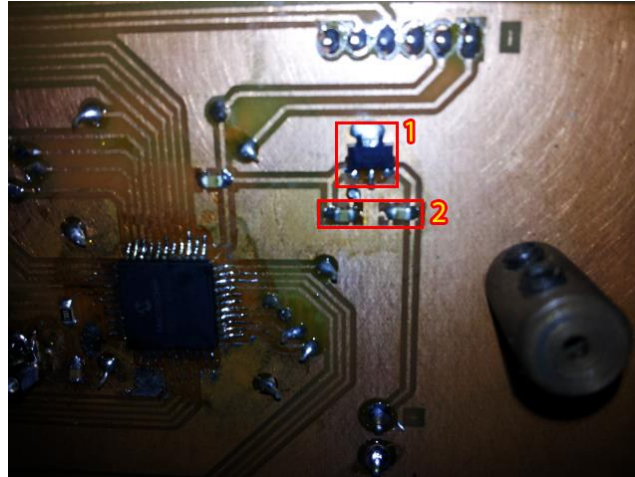


Figura 4.8. Circuito regulador tensión 7805 cara bottom

1. Regulador de tensión SMD modelo 7805.
2. 2 condensadores SMD de 100nF.

## 4.2.- Montaje de la parte de potencia. Motor.

Una vez terminada la placa que en un futuro tenemos intención de que rote, seguiremos el montaje del proyecto realizando la parte de potencia, que será la que impulse dicha placa.

Para fijar la placa principal al motor utilizaremos un soporte cilíndrico metálico personalizado en cuyas bases encontraremos taladros de 3mm (diámetro del tornillo que irá acollado a la placa principal por la cara top) y 2mm (diámetro del eje del rotor del motor), situado en la cara bottom.



Figura 4.9. Acoplador placa principal-motor





*Figura 4.10. Vista inferior del adaptador placa principal-motor*

1. Taladro de 2mm de diámetro para insertar el motor.

Como se puede apreciar, para poder fijar el motor y el tornillo superior de 3mm de diámetro, se introdujeron en el soporte tornillos hexagonales en cada extremo del mismo que se atornillarán con una llave Allen de 1,5 mm.



*Figura 4.11. Llave Allen*

Para fijar el tornillo a la cara top de la placa y así evitar que baile durante la rotación de la misma, se le extraerá su cabeza y se añadirá una arandela y una tuerca de 3mm. que harán fuerza contra la placa.



*Figura 4.12. Vista superior acople placa principal-motor*

Para la parte de potencia que hará que rote la placa principal, se utilizará un motor de continua ARISTON modelo MOV 112, actualmente descatalogado, que nos da un rango de tensión de los 1,5V. a los 9V., suficiente para la correcta visualización de la imagen (con 3V.

basta). Su sustituto podría ser un CEBEK modelo C6060 que se encuentra en la misma web de Ariston y que ofrece un rango de tensiones de 1,5V A 12V, arrancando con sólo 10mA.

Un motor de corriente continua, que convierte la energía eléctrica en energía mecánica, se compone principalmente de dos partes, un estátor que da soporte mecánico al aparato y tiene un hueco en el centro generalmente de forma cilíndrica. En el estátor además se encuentran los polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, al que llega la corriente mediante dos escobillas.

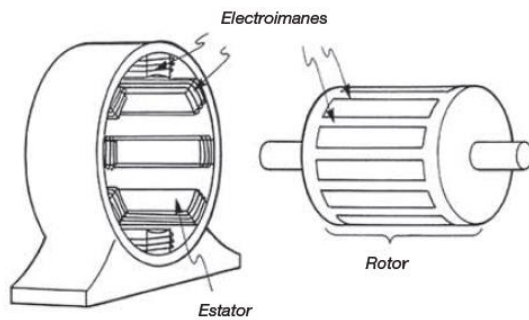


Figura 4.13. Estructura interna motor DC



Figura 4.14. Motor DC ARISTON MOV 112

En el recuadro rojo se puede observar el eje del rotor que irá encajado en el taladro de 2mm. del soporte.

Para sujetar el motor pensamos en adherirlo a un bloque de madera, cuyo peso deberá soportar la fuerza centrífuga que ejerza la placa giratoria sobre él, con unos soportes curvados que lo rodearan y que se atornillaran a dicho bloque.

El resultado será el siguiente:



Figura 4.15. Montaje Motor DC

Dicho motor lo alimentaremos de manera que gire hacia la derecha para así asegurarnos la correcta visualización de la imagen programada.

Ya elegido el motor que queremos incorporar para impulsar la placa principal, le añadiremos un potenciómetro rotativo de 100 ohmios cuya función será la de regular su velocidad, ya que no necesitamos toda la potencia del mismo ni todo el voltaje de la batería de 9V. (que nos



servirá como alimentador del motor) para poder ver bien la imagen. Se hicieron pruebas con una fuente de alimentación y nos dio como resultado un voltaje en torno a los 3 V. para dicho fin.

El montaje del potenciómetro en modo de resistencia variable será el siguiente:

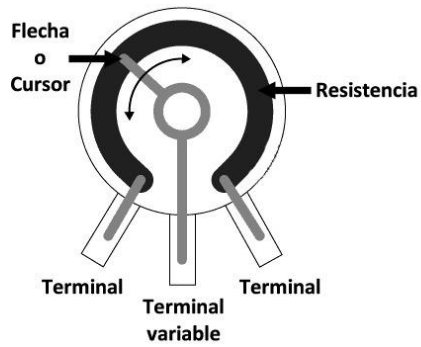


Figura 4.16. Estructura potenciómetro

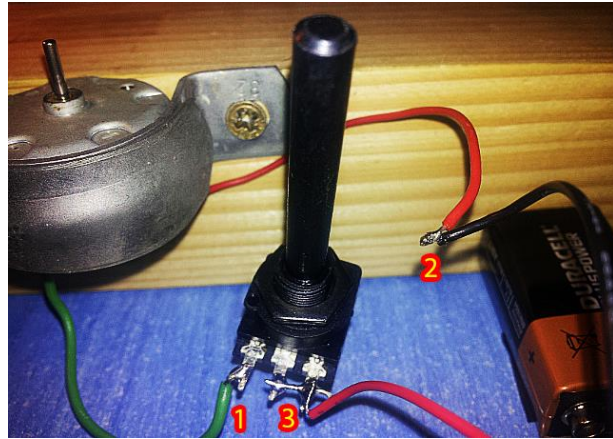


Figura 4.17. Montaje potenciómetro rotatorio 100 ohmios

1. Alimentación del motor conectada a un terminal del potenciómetro.
2. Las dos masas (motor y batería) unidas externamente.
3. Alimentación de la batería unida al otro terminal del potenciómetro, que a su vez está puenteado con el terminal variable.

Una vez terminado el montaje del motor y de su regulador en forma de potenciómetro, ya podemos dar por terminada la parte de potencia.

Para finalizar con el montaje, sólo nos quedará poner un imán en el bloque de madera que esté a la altura del sensor Hall, situado según donde queramos que el dispositivo comience a dibujar la imagen.

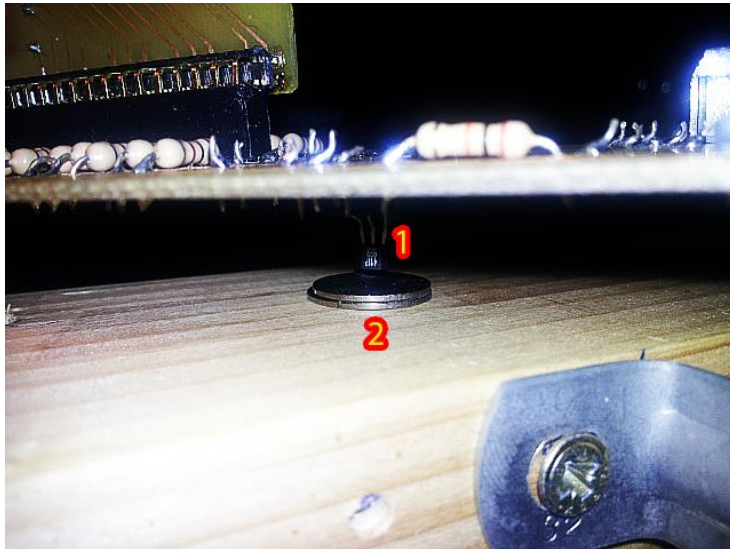


Figura 4.18. Montaje imán-sensor Hall

1. Sensor Hall
2. Imán de neodimio

Una vez detalladas todas las partes del proyecto, se procederá a su montaje final quedando de la siguiente manera:

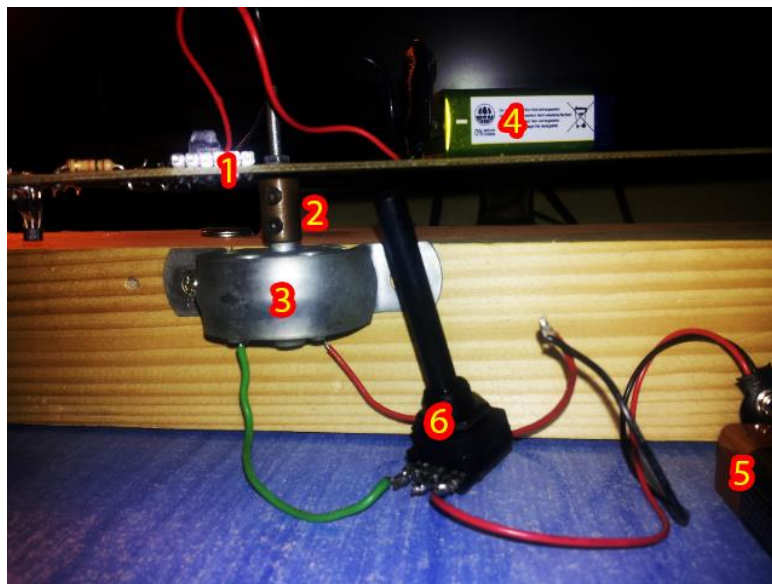
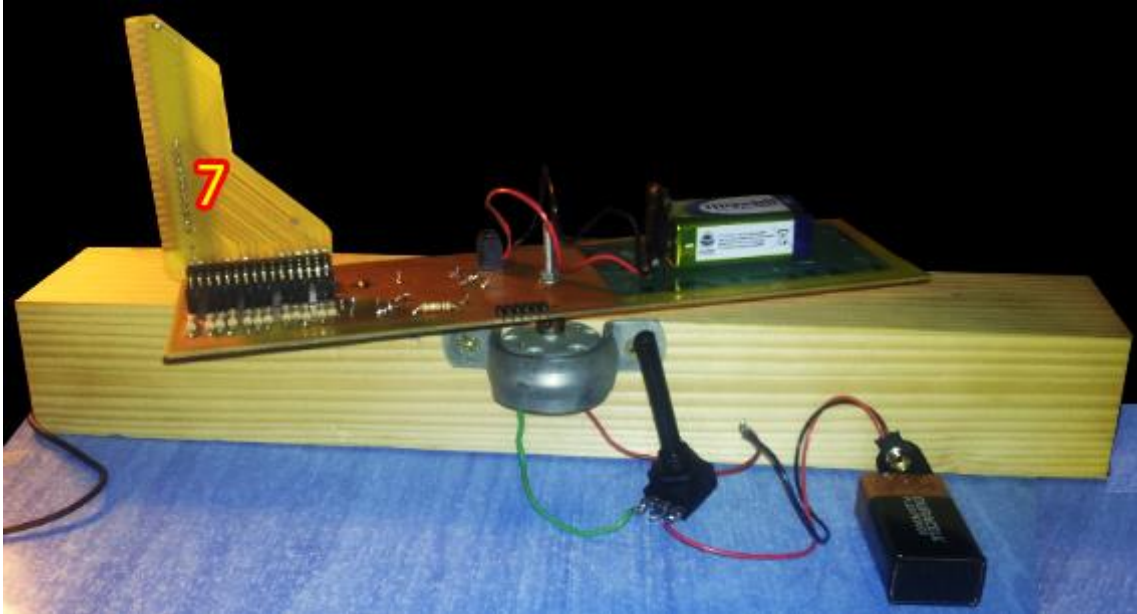


Figura 4.19. Montaje final 1

1. Placa principal
2. Soporte que fija la placa principal con el motor.
3. Motor.
4. Batería de 9V. que alimentará a la placa principal.
5. Batería de 9V. que alimentará al motor.
6. Potenciómetro rotatorio de 100 ohmios que regulará la velocidad del motor.



*Figura 4.20. Montaje final 2*

7. Placa de LEDs RGB y adaptador para la placa principal.

Una vez ya tenemos el proyecto final montado, sólo faltaría programar el PIC con la imagen que queremos que saque por los LEDs RGB.

---

## 5.PROGRAMACIÓN

---

En este apartado detallaremos los programas y dispositivos que intervienen en el proceso de programación del PIC, así como también un código de programa ejemplo para poder ver su funcionamiento interno.

### 5.1– Conversión de imagen

Para convertir una imagen cualquiera a la gama de colores que pueden ofrecernos los LEDs RGB, utilizaremos un programa ya desarrollado, el POV Graphic Converter, que la transforma en un string que se guardará en la memoria EEPROM del PIC.

Como imagen ejemplo, cogeremos el logo de McDonalds:



Figura 5.1. Logo McDonalds

Al tener la imagen original un tamaño más extenso que el permitido por la pantalla de LEDs, se le reducirá a 8 píxeles de alto, siendo ese el número de LEDs que tenemos, con algún programa de edición de imágenes, por ejemplo el Paint, quedando de la siguiente manera:



Figura 5.2. Logo McDonalds redimensionado 400x8 píxeles

Concretamente, hemos redimensionado la imagen a 400 píxeles de largo por 8 de alto. Como se puede ver, apenas se puede distinguir al tener escasos píxeles de altura.

Una vez ya tenemos la imagen con el tamaño deseado, abriremos el conversor a LEDs RGB:

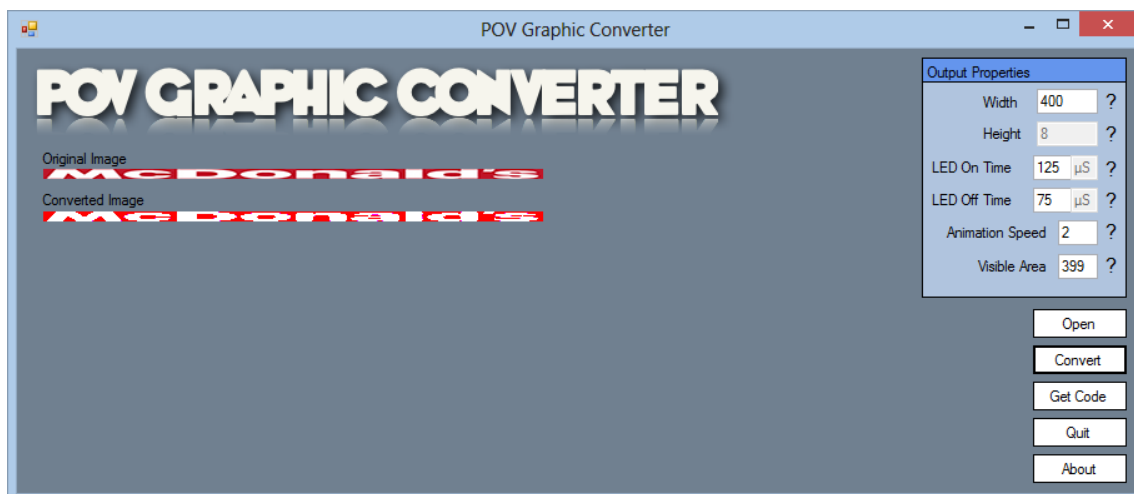


Figura 5.3. Interfaz programa conversor de imagen

Le daremos a *Open* para seleccionar la ruta de la imagen. Una vez seleccionada, aparecerá en el apartado *Original Image*.

Una vez tenemos la imagen en pantalla, asignaremos las propiedades que queremos que tenga la imagen convertida en el apartado *Output Properties*.

- **Width:** Ancho o largo de la imagen.
- **Height:** Altura de la imagen.
- **LED On Time:** Tiempo de encendido del LED, en microsegundos.
- **LED Off Time:** Tiempo de apagado del LED, en microsegundos.
- **Animation Speed:** Velocidad de scroll vertical de la animación de la imagen. Cuanto más bajo sea el número, más rápida irá la animación.
- **Visible Area:** Del largo total de la imagen original, decidiremos cuantos píxeles queremos ver. En el ejemplo aparece un valor de 399, uno menos que el largo original, ya que con 400 no tenía un buen acabado.

Una vez establecidas las anteriores características, se le dará a *Convert* y la imagen convertida aparecerá en el campo *Converted Image*.

Con la imagen convertida, le daremos a *Get code* para extraer el código de la misma, en lenguaje BASIC, el cuál pegaremos posteriormente en el *Swordfish*, un compilador de BASIC para microcontroladores PIC de la familia PIC18, útil para el PIC 18F4680 que hemos incorporado en este proyecto.

Tardará unos minutos y finalmente nos aparecerá un aviso en pantalla diciéndonos que el código ya está creado y listo para pegarlo en el programa citado anteriormente.

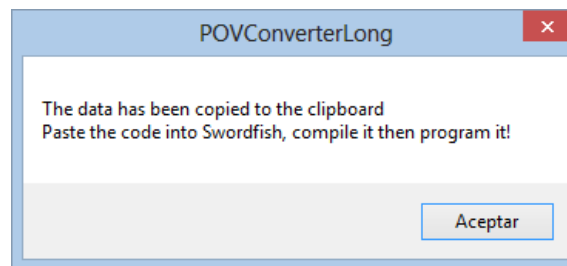


Figura 5.4. Aviso creación satisfactoria del código

Sólo nos quedará darle a *Quit* para cerrar el POV Graphic Converter.

## 5.2– Compilador y código fuente

El compilador utilizado será el *Swordfish*, mencionado en el apartado anterior. Lo abriremos y pegaremos el código que nos generó el *POV Graphic Converter* en la pantalla principal, sin modificar ninguna opción:

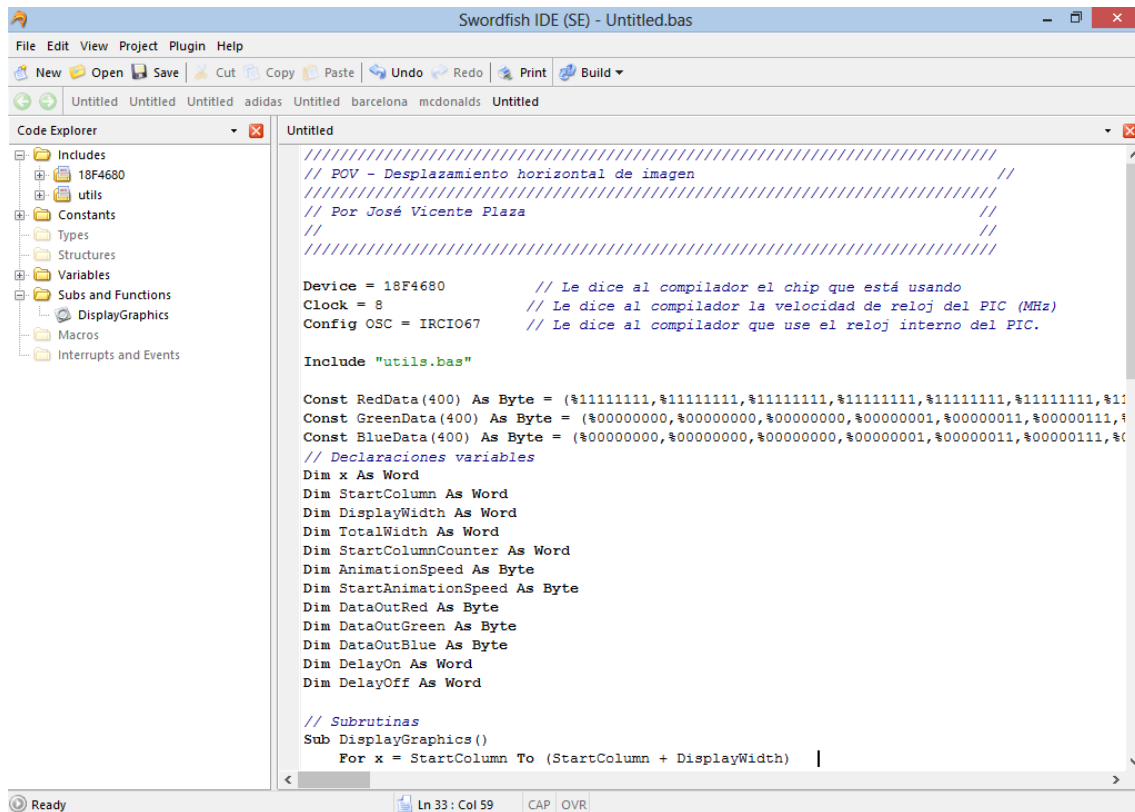


Figura 5.5. Interfaz compilador Swordfish

La totalidad del código de programa resultante de la imagen es el siguiente:

```

Device = 18F4680           // Le dice al compilador el chip que está usando.
Clock = 8                 // Le dice al compilador la velocidad de reloj del PIC (MHz)
Config OSC = IRCIO67     // Le dice al compilador que use el reloj interno del PIC.

Include "utils.bas"      //Se carga la librería del PIC.

Const RedData(400) As Byte = (%11111111,%11111111,%11111111,...)
//Constante String de 400 elementos en binario(400 píxeles de largo) que
constituye el color rojo de la imagen.

Const GreenData(400) As Byte = (%00000000,%00000000,%00000000,...)
//Constante String de 400 elementos en binario(400 píxeles de largo) que
constituye el color verde de la imagen.

Const BlueData(400) As Byte = (%00000000,%00000000,%00000000,...)
//Constante String de 400 elementos en binario(400 píxeles de largo) que
constituye el color azul de la imagen.

// Declaraciones de las variables (Word=16 bits [0 a 65535] y Byte=8
bits [0 a 255])
Dim x As Word
Dim StartColumn As Word
Dim DisplayWidth As Word
Dim TotalWidth As Word
Dim AnimationSpeed As Byte
Dim StartAnimationSpeed As Byte
Dim DataOutRed As Byte
Dim DataOutGreen As Byte
Dim DataOutBlue As Byte
Dim DelayOn As Word
Dim DelayOff As Word

```

```

// Subrutinas
Sub DisplayGraphics()
  For x = StartColumn To (StartColumn + DisplayWidth) // Bucle for
de 0 a 0+400=400 (Variará según el ancho de la imagen)
    PORTD=RedData(x) //Irá saliendo cada elemento del color rojo
por el PORTD.
    PORTB=GreenData(x) //Elemento del color verde por el PORTB.
    PORTA=BlueData(x) //Elemento del color azul por el PORTA.
    PORTE.bits(0)=GreenData(x).bits(6) //7ºbit del elemento verde
por el pin RE0.
    PORTE.bits(1)=GreenData(x).bits(7) //8ºbit del elemento verde
por el pin RE1.

    DelayUS(DelayOn)//Retardo de 125 us.
    PORTA = 0 //Todos los puertos a cero para que saquen
datos limpios.
    PORTB = 0
    PORTD = 0
    PORTE = 0
    DelayUS(DelayOff) //Retardo de 75 us.
  Next //Fin del bucle for.
  If AnimationSpeed <> 0 Then //Si la velocidad de la animación es
distinta de cero...entonces...(en este caso tiene como valor 2, declarado
más abajo)
    AnimationSpeed = AnimationSpeed - 1 //Se le restará 1 hasta que
sea cero y pase al else. Funciona como retardo.
  Else
    AnimationSpeed = StartAnimationSpeed //AnimationSpeed se
volverá a inicializar a 2.
    StartColumn = StartColumn + 1 //Se desplazará una columna.
    If StartColumn + DisplayWidth = TotalWidth Then
      StartColumn = 0 //Se seguirán desplazando las columnas
hasta que la suma del conteo de columnas más el área visible de la
imagen dé como resultado el ancho total de la imagen, lo cuál hará que
el contador de columnas se inicialice a cero y empiece por el comienzo
de la imagen.
    EndIf //Fin instrucción if.
  EndIf //Fin instrucción if...else.
End Sub //Fin de la subrutina.

// Comienzo del programa
OSCCON = %01111111 // Configura el oscilador interno para los 8 MHz.
SetAllDigital // Todas las entradas y salidas digitales.
TRISD = %00000000 // Configura el PORTD como salida (color rojo).
TRISB = %00000000 // Configura el PORTB como salida (color verde).
TRISC = %00000000 // Configura el PORTC como salida (no conectado).
TRISA = %00000000 // Configura el PORTA como salida (color azul).
TRISE = %00000100 // Configura el PORTE como salida menos el pin
RE2 que servirá como entrada para el estímulo del sensor Hall (RE0 y
RE1 color verde).
PORTA = %00000000 //Valores iniciales de todos los puertos a cero.
PORTB = %00000000
PORTC = %00000000
PORTD = %00000000
PORTE = %00000000

//Valores de las variables
StartColumn = 0 //Inicio de la imagen. Primera columna.

TotalWidth = 400 //Ancho total de la imagen.

```



```

DelayOn = 125 //Tiempo de encendido del LED.
DelayOff = 75 //Tiempo de apagado del LED.
StartAnimationSpeed = 2 //Velocidad del scroll horizontal de la
imagen.
DisplayWidth = 399 //Ancho visible de la imagen.

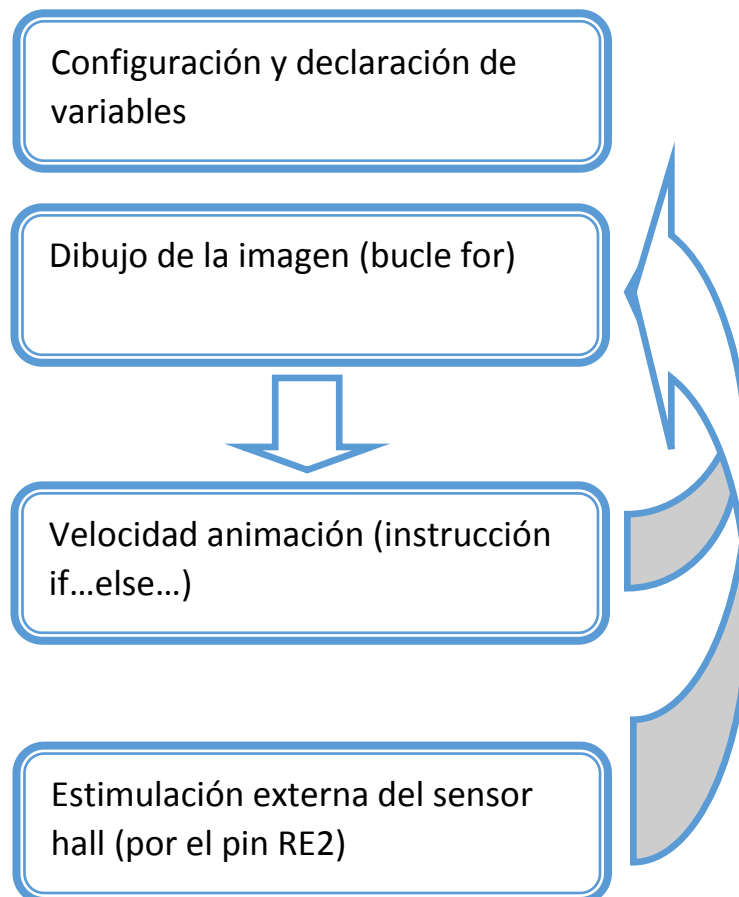
AnimationSpeed = StartAnimationSpeed //AnimationSpeed = 2.

// Bucle principal
While True() // Esto crea un bucle infinito.
  If PORTE.bits(2) = 0 Then
    DisplayGraphics //Cuando al pin RE2 de entrada le
entra el estímulo del sensor Hall (nivel bajo), el programa vuelve a
la subrutina DisplayGraphics que empezará a dibujar de nuevo la
imagen.

  EndIf //Fin instrucción if
Wend // Vuelve al bucle while siempre y
cuando no hayamos terminado.

```

Lo resumiremos todo en el siguiente diagrama:



Una vez realizado el código del programa, sólo nos faltará extraer el archivo \*.hex del mismo, que es el que necesitaremos para programarlo en el PIC. Para hacer esto, dentro del *Swordfish* nos iremos a **Project – Compile** y nos generará 6 archivos (\*.asm, \*.bas, \*.cof, \*.HEX, \*.idf y \*.sfp) en la carpeta en la que hayamos guardado previamente el código.

### 5.3– Proceso de programación del PIC

Ya teniendo el archivo HEX en nuestras manos, sólo nos faltará traspasarlo al PIC mediante el terminal de programación de 6 pines que se encuentra en la placa principal del Proyecto. Para ello, necesitaremos un programador para PICs (usaremos el PICKit 2) y su software correspondiente (PICKit 2 v2.61) que se puede encontrar en la página web de Microchip, concretamente en el siguiente [enlace](#), con el cuál cargaremos el archivo HEX y se lo traspasaremos al PIC.



Figura 5.6. Programador PICKit 2

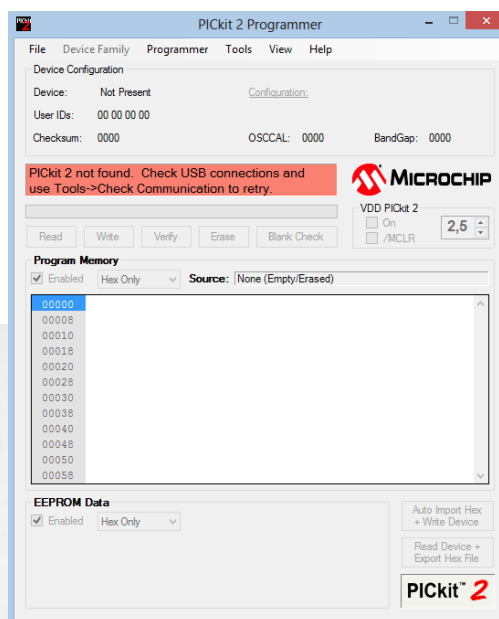


Figura 5.7. Software PICKit 2

El programador PICKit 2 se conectará al PC mediante un cable USB (PC)-microUSB (PICKit 2).



Figura 5.8. Cable USB-Micro USB



Figura 5.9. Entrada MicroUSB PICkit 2

Para conectar el PICkit 2 en la placa principal, sólo lo tendremos que insertar en el terminal de programación de 6 pines de la misma, fijándonos que la flecha del PICkit (1) coincida con el terminal “-“ de la placa (2).



Figura 5.10. Conector 6 pines hacia placa

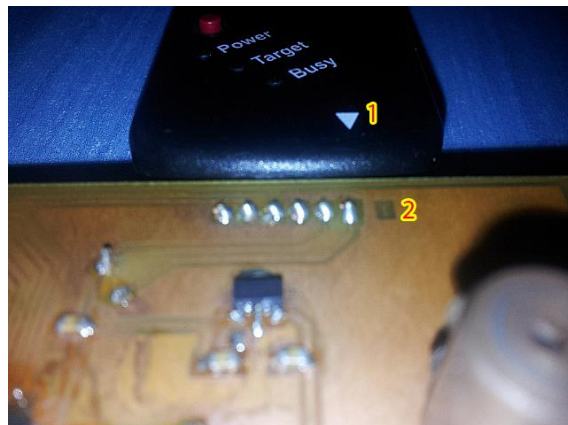


Figura 5.11. Conexión correcta con placa

Una vez conectado correctamente el programador PICkit 2 a la placa principal y al PC, nos aparecerá automáticamente el modelo del PIC en el software.

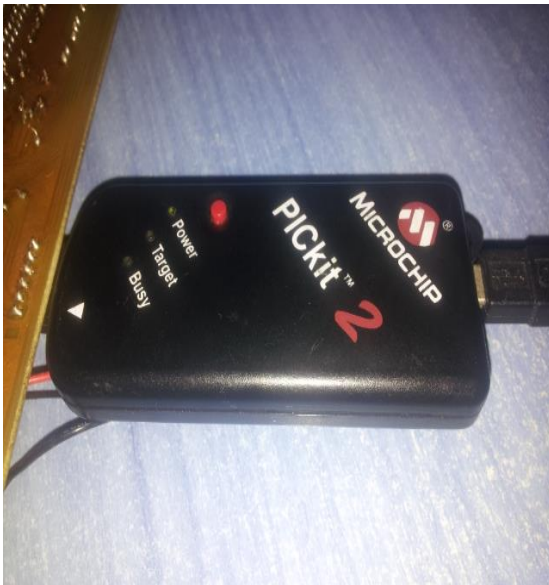


Figura 5.12. Conexión al ordenador

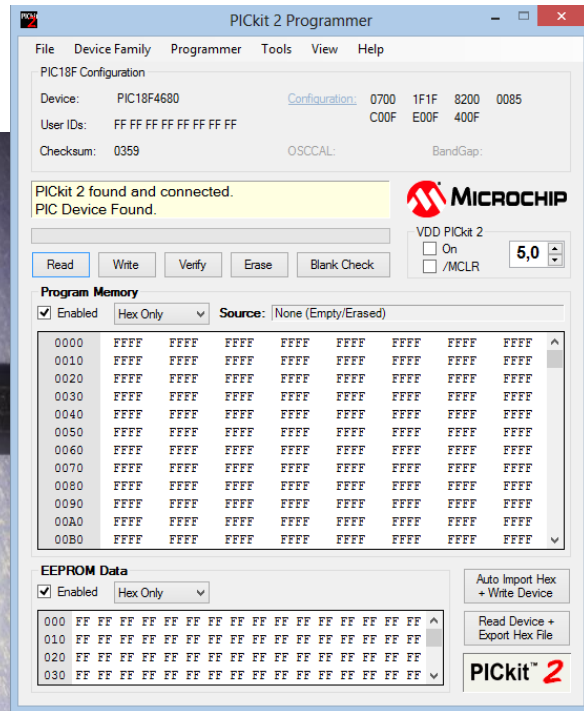


Figura 5.13. Verificación del PIC

Para cargar el archivo HEX, nos iremos a **File – Import Hex** y lo seleccionaremos de la carpeta en la que lo hayamos guardado en el *Swordfish*. Seguidamente, nos aparecerá el mensaje “Hex file succesfully imported” indicándonos que el HEX ha sido correctamente cargado. Además, los apartados **Program Memory** y **EEPROM Data** ahora ya no estarán llenos de “F” sino que estarán cargados con el código hexadecimal del programa.

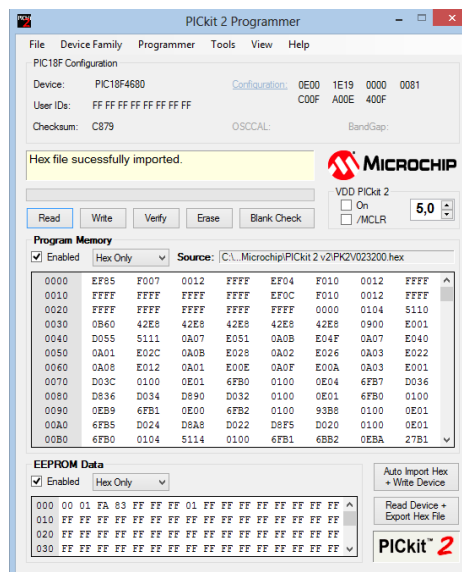


Figura 5.14. Programa correctamente cargado

Sólo nos quedará darle click al botón **Write** para traspasarlo al PIC y a **Verify** para ver si todo ha ido correctamente. Un atajo que podemos usar para cargar el archivo HEX y programárselo al PIC de una sola vez es el botón **Auto Import Hex +Write Device**.

Nota: No se deberá alimentar la placa principal mientras se esté usando el PICkit.

---

## 6.RESULTADO FINAL

---

Nos encontramos en la parte final del proyecto, en la que expondremos los pasos que se tienen que seguir para el buen funcionamiento del dispositivo POV.

Los pasos a seguir serán:

- 1 – Programar el PIC con la imagen deseada siguiendo los pasos del anterior apartado.
- 2 – Insertar el adaptador de LEDs RGB en el conector hembra de 32 pines que se encuentra en la placa principal.



*Figura 6.1. Conexión adaptador LEDs a placa*

- 3 – Insertar la placa principal en el soporte del motor.



*Figura 6.2. Inserción soporte placa en motor*

- 4 – Colocar el imán de neodimio de tal manera que quede debajo del sensor Hall al rotar la placa.





Figura 6.3. Colocación imán neodimio

5 – Alimentar la placa principal con la batería de 9V conectada al conector de 2 pines correspondiente.

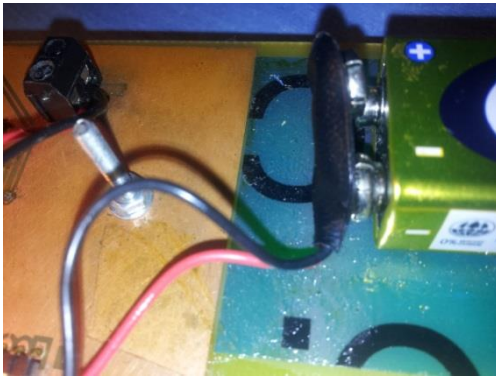


Figura 6.4. Alimentación placa 1



Figura 6.5. Alimentación placa 2

6 – Alimentar el motor con otra batería de la misma tensión.



Figura 6.6. Alimentación motor DC

7 – Regular la velocidad del motor con el potenciómetro hasta que la imagen tenga la longitud que se requiera.





Figura 6.7. Regulación velocidad motor

8 - ¡A funcionar!



Figura 6.8. En funcionamiento 1

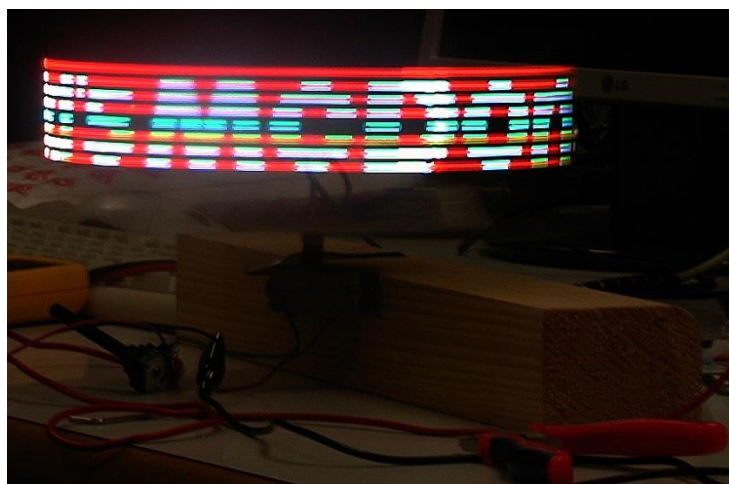


Figura 6.9. En funcionamiento 2

---

## 7.PRESUPUESTO

---

<b>Componente</b>	<b>Cantidad</b>	<b>Precio total (IVA incluido)</b>
Microcontrolador PIC18F4680 SMD	1	8,44€
Regulador de tensión 7805 SMD	1	0,44€
Resistencia 100 ohmios 0,25W	24	0,48€
Resistencia 10k ohmios 0,5W	1	0,04€
Resistencia 10k ohmios SMD	1	0,041€
Condensador 100nF SMD	4	0,828€
Sensor Hall Honeywell SS411P	1	0,50€
Potenciómetro rotatorio 100 ohmios 500V	1	4,26€
Motor DC CEBEX C6060 1,5V- 12V	1	7,74€
LED RGB SMD Multicomp OVS5309	8	15,68€
Conector hembra 2 pines	1	2,29€
Conector molex macho acodado 6 pines	1	1,15€
Conector tipo socket hembra de 32 pines	1	2,24€
Conector SIL/DIL macho acodado de 32 pines (adaptador a placa)	1	1,18€
Batería 9V	2	9€
		<b>54,309€</b>

---

## 8.CONCLUSIÓN

---

En este Proyecto Fin de Carrera se ha planteado el objetivo de controlar un mecanismo rotativo de LEDs a través de la programación de un microcontrolador. Ha sido necesario el estudio en profundidad del comportamiento de los fotogramas, a fin de comprender las necesidades que estos pueden desarrollar.

Hasta llegar a este diseño con un solo chip, se pasó por un prototipo inicial en el que se usaban biestables para sincronizar las salidas y una memoria EEPROM por separado para guardar el código de la imagen. Dadas las dificultades que se presentaron en el diseño, así como también el tamaño desmedido de la placa, se optó por implementar todas esas funciones en un único microcontrolador PIC que facilitaba en gran medida la construcción del dispositivo que se ha presentado.

También surgieron problemas a la hora de conectar la placa con el motor, ya que no se encontraba en el mercado una pieza que encajara a la perfección en tal propósito, así que finalmente la tuvo que personalizar un tornero. Una vez salvaguardado ese problema, para evitar que el tornillo superior se deslizara por la placa al realizar ésta el movimiento de rotación, se acolló a la misma con una arandela y una tuerca para que hiciera presión sobre ella.

En cuanto al proceso de diseño en el OrCAD, se implementaron los conocimientos ya vistos durante la especialidad de Sistemas Electrónicos, tanto en el Capture como en el Layout. En la programación del PIC, se usó un lenguaje de programación no cursado como es el Visual Basic, al igual que el compilador asignado a él (Swordfish), del que tuve que buscar información para conocer su funcionamiento.

Desde el punto de vista económico, no hizo falta un gran desembolso para desarrollar el dispositivo, siendo lo más costoso los LEDs RGB SMD.

---

# BIBLIOGRAFÍA

---

## Páginas web:

[1] Colaboradores de Wikipedia. *Led* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Led&oldid=70866650>>.

[2] Colaboradores de Wikipedia. *Motor de corriente continua* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Motor de corriente continua&oldid=69593704>>.

[3] Colaboradores de Wikipedia. *Regulador de tensión* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Regulador de tensi%C3%B3n&oldid=69462826>>.

[4] Colaboradores de Wikipedia. *Microcontrolador* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Microcontrolador&oldid=70094543>>.

[5] Colaboradores de Wikipedia. *Microcontrolador PIC* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Microcontrolador PIC&oldid=70094662>>.

[6] Colaboradores de Wikipedia. *Efecto Hall* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Efecto Hall&oldid=68145194>>.

[7] Colaboradores de Wikipedia. *Persistencia de la visión* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Persistencia de la visi%C3%B3n&oldid=69996903>>.

[8] Colaboradores de Wikipedia. *Ojo humano* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Ojo humano&oldid=70906544>>.

[9] Colaboradores de Wikipedia. *Taumatropo* [en línea]. Wikipedia, La enciclopedia libre, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://es.wikipedia.org/w/index.php?title=Taumatropo&oldid=69040438>>.

[10] Odilo Montero. *Leds* [Blog Internet]. Una Pausa para la Publicidad, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://unapausaparalapublicidad.com/tag/leds/>>

[11] Urriellu. *POV (Persistence of Vision display)*

[Blog Internet] Blog de desarrollo de Curuxa, 2010 [fecha de consulta: 20 de noviembre del 2013]. Disponible en < <http://blog.curuxa.org/electronica/pov-persistence-of-vision-display/>>

[12] Varios. *Iluminación LED*

[Blog Internet] Energética futura, 2013 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://energeticafutura.com/blog/category/categorias/iluminacion-led/>>

[13] Microsoft Corporation. *Instrucciones (Visual Basic)* [en línea] Microsoft.com, 2007 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <[http://msdn.microsoft.com/es-es/library/2f43da0y\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/2f43da0y(v=vs.90).aspx)>

[14] Rene Camacho. *Arquitectura von Neumann y arquitectura Harvard* [Blog Internet] Computo Integrado, 2012 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://rcmcomputointegrado.blogspot.com.es/2012/04/arquitectura-von-neumann.html>>

[15] *Microcontroladores PIC* [Blog Internet] Dispositivos lógicos programables, 2010 [fecha de consulta: 20 de noviembre del 2013]. Disponible en <<http://perso.wanadoo.es/pictob/micropic.htm>>

---

# ÍNDICE DE IMÁGENES

---

<b>CONCEPTO</b>	<b>PÁGINA</b>
1.1. Anuncio Peugeot.....	1
1.2. Taumátropo .....	2
1.3. Zoótrofo .....	2
1.4. Fenaquistiscopio .....	2
1.5. Rótulo LED.....	3
1.6. Panel LED .....	3
1.7. Londres .....	3
1.8. Nueva York.....	3
1.9. Holograma Michael Jackson .....	4
1.10. Holograma película Avatar.....	4
2.1. Corte transversal de un ojo humano .....	6
2.2. Espectro de frecuencias ojo humano.....	6
2.3. Respuesta del ojo al espectro .....	7
2.4. Conos y bastones .....	7
2.5. Transiciones de imágenes .....	8
2.6. Persistencia retiniana.....	8
2.7. Estela LEDs .....	8
2.8. Taumátropo .....	9
2.9. Estroboscopio .....	10
2.10. Zoótrofo .....	11
2.11. Praxinoscopio.....	11
2.12. Funcionamiento disco de Nipkow.....	12
2.13. Televisión electromecánica.....	12
2.14. Linterna mágica.....	13
2.15. Cine NIC.....	14
2.16. Paneles de publicidad de LEDs.....	14
2.17. Paneles de información scroll-text .....	14
2.18. Desnudo bajando una escalera de Duchamp.....	15
2.19. Microcontrolador PIC18F4520 .....	17
2.20. Microcontroladores .....	17
2.21. Microprocesador Intel 4004 .....	18
2.22. Microcontrolador TMS 1000 .....	18
2.23. PICs de la familia de 16 bits .....	18
2.24. Atmel con memoria flash.....	18
2.25. Estructura de un microcontrolador .....	19
2.26. Aplicaciones microcontrolador .....	20
2.27. Arquitectura Harvard .....	21
2.28. Comparativa tamaño varios PICs .....	23
2.29. Programador PICKit 2 .....	26
2.30. Clasificación según el tamaño de palabra.....	28
2.31. Arquitectura interna del PIC18F4680 .....	35
2.32. Comparativa entre PICs similares .....	38
2.33. Encapsulado TQFP.....	39
2.34. Distribución pines PIC18F4680 .....	39
3.1. Configuración proyecto OrCAD Capture .....	42
3.2. Esquemático placa principal .....	43

3.3. Efecto Hall .....	43
3.4. Estructura del sensor Hall .....	44
3.5. Tamaño sensor Hall.....	45
3.6. Regulador de tensión 7805 SMD .....	45
3.7. LEDs monocromo .....	46
3.8. Símbolo electrónico LED .....	46
3.9. Tira de LEDs RGB .....	47
3.10. Estructura LED RGB SMD .....	47
3.11. Tabla de especificaciones del LED RGB .....	48
3.12. Diagrama de radiación del LED RGB .....	48
3.13. Esquemático del adaptador LED a placa .....	49
3.14. Esquemático de la placa de LEDs RGB .....	49
3.15. Configuración netlist OrCAD Capture .....	50
3.16. Configuración PCB OrCAD Layout .....	50
3.17. Configuración anchura pistas OrCAD Layout .....	51
3.18. Configuración separaciones OrCAD Layout .....	51
3.19. Configuración borde PCB OrCAD Layout.....	51
3.20. Vista layout capa top placa principal .....	53
3.21. Vista ensamblaje capa top placa principal .....	53
3.22. Vista layout capa bottom placa principal.....	53
3.23. Vista ensamblaje capa bottom placa principal .....	54
3.24. Encapsulado LED RGB .....	54
3.25. Vista layout capa top adaptador .....	55
3.26. Vista layout capa bottom adaptador .....	55
3.27. Vista layout capa top placa LEDs RGB.....	55
3.28. Vista layout capa bottom placa LEDs RGB .....	55
3.29. Tabla Post Process Settings OrCAD Layout .....	56
3.30. Configuración capa Post Process Settings OrCAD Layout .....	56
4.1. Vista superior placa principal .....	57
4.2. Vista inferior placa principal.....	57
4.3. Vista superior adaptador LEDs RGB.....	58
4.4. Vista inferior adaptador LEDs RGB .....	58
4.5. Placa LEDs RGB .....	58
4.6. Placa rotatoria completa .....	59
4.7. Circuito sensor Hall cara bottom .....	59
4.8. Circuito regulador tensión 7805 cara bottom .....	60
4.9. Acoplador placa principal-motor .....	60
4.10. Vista inferior del adaptador placa principal-motor .....	61
4.11. Llave Allen.....	61
4.12. Vista superior acople placa principal-motor .....	61
4.13. Estructura interna motor DC .....	62
4.14. Motor DC ARISTON MOV 112.....	62
4.15. Montaje motor DC.....	62
4.16. Estructura potenciómetro .....	63
4.17. Montaje potenciómetro rotatorio 100 ohmios.....	63
4.18. Montaje imán-sensor Hall .....	64
4.19. Montaje final 1 .....	64
4.20. Montaje final 2 .....	65
5.1. Logo McDonalds .....	66
5.2. Logo McDonalds redimensionado 400x8 píxeles .....	66
5.3. Interfaz programa conversor de imagen .....	66
5.4. Aviso creación satisfactoria del código.....	67
5.5. Interfaz compilador Swordfish .....	68



5.6. Programador PICKit 2.....	71
5.7. Software PICKit 2 .....	71
5.8. Cable USB-Micro USB .....	72
5.9. Entrada MicroUSB PICKit 2 .....	72
5.10. Conector 6 pines hacia placa .....	72
5.11. Conexión correcta con placa .....	73
5.12. Conexión al ordenador .....	73
5.13. Verificación del PIC.....	73
5.14. Programa correctamente cargado .....	73
6.1. Conexión adaptador LEDs a placa .....	75
6.2. Inserción soporte placa en motor .....	75
6.3. Colocación imán neodimio .....	76
6.4. Alimentación placa 1 .....	76
6.5. Alimentación placa 2 .....	76
6.6. Alimentación motor DC.....	76
6.7. Regulación velocidad motor .....	77
6.8. En funcionamiento 1 .....	77
6.9. En funcionamiento 2 .....	77

---

# ANEXOS

---

## Anexo 1. Visual Basic

### INSTRUCCIONES BÁSICAS

Integer	Número entero
Double	Número con decimales
Msgbox	Caja de texto
Inputbox	Caja para meter el dato de una variable
Sub	Es la rutina se acompaña de la frase programal (), para el programa que se ejecuta.
End Sub	Final del programa
Dim	Define el nombre de una variable Dim num1 as integer Dice la variable num1 es un entero
As	Define como
or	Define tal cosa o tal otra
string	Define que la variable a introducir va a ser texto
*1, *2, *3...	Asterisco algo al lado de string, integer o doublé, define la longitud de los caracteres que pueden meterse en esa variable
&	Funciona como compilador de una acción por ejemplo de un resultado, algo así como and, o y, pasa tal cosa & tal otra.
If	Si ocurre tal cosa (condición) then
then	Entonces pasa tal cosa, instrucción 1/ instrucción 2
Else	De lo contrario o si no ocurre pasa tal cosa Instrucción3/Instrucción4
End if	Finaliza o cierra la variable If
Do while	Si la variable cumple con una condición pre fijada se realiza la siguiente instrucción, en lo que no cumpla más , deja de hacerse la instrucción
Loop	Ciclo o lazo, que debe repetirse.
For	Para tales valores o condiciones ocurre tal instrucción
to	Entre esta y esta condición por ejemplo de 1 to 10 se cumple con tal condición
next	Repite el ciclo anterior hasta que se haya cumplido con la condición señalada en for.
step	Es un paso, una suma, algo que se va agregando
Debug.Print	Instrucción que publica un mensaje en el panel de abajo, en el panel de inmediato
Byte	Numero entre el 0 y el 255
Exit for	Orden que se da, si se cumple con tal cosa, váyase hasta tal cosa o acción

## OPERADORES

Tipo	Operación	Operador en Visual Basic
Aritmético	Exponenciación	^
	Cambio de Signo	-
	Multiplicación, División	*, /
	División Entera	\
	Resto de una División Entera	Mod
	Suma y Resta	+, -
Concatenación	Enlazar o Concatenar	&, +
Relacional	Igual a	=
	Distinto	<>
	Menor que, Menor o igual que	<, <=
	Mayor que, Mayor o igual que	>, >=
Lógico	Negación	Not
	And	And
	Or inclusivo	Or
	Or Exclusivo	Xor

## Anexo 2. Unidades del Sistema Internacional

### UNIDADES FUNDAMENTALES

Magnitud física que se toma como fundamental	Unidad básica o fundamental	Símbolo
Longitud (l)	metro	m
Masa (m)	kilogramo	kg
Tiempo (t)	segundo	s
Intensidad de corriente eléctrica (I)	amperio	A
Temperatura (T)	kelvin	K
Cantidad de sustancia (n)	mol	mol
Intensidad luminosa (I)	candela	cd

## UNIDADES DERIVADAS

Magnitud física	Nombre de la unidad	Símbolo de la unidad	Expresada en unidades derivadas	Expresada en unidades básicas
Frecuencia	hertz	Hz		s <sup>-1</sup>
Fuerza	newton	N		m·kg·s <sup>-2</sup>
Presión	pascal	Pa	N·m <sup>-2</sup>	m <sup>-1</sup> ·kg·s <sup>-2</sup>
Energía, trabajo, calor	joule	J	N·m	m <sup>2</sup> ·kg·s <sup>-2</sup>
Potencia	watt	W	J·s <sup>-1</sup>	m <sup>2</sup> ·kg·s <sup>-3</sup>
Carga eléctrica	coulomb	C		A·s
Potencial eléctrico, fuerza electromotriz	volt	V	J·C <sup>-1</sup>	m <sup>2</sup> ·kg·s <sup>-3</sup> ·A <sup>-1</sup>
Resistencia eléctrica	ohm	Ω	V·A <sup>-1</sup>	m <sup>2</sup> ·kg·s <sup>-3</sup> ·A <sup>-2</sup>
Conductancia eléctrica	siemens	S	A·V <sup>-1</sup>	m <sup>-2</sup> ·kg <sup>-1</sup> ·s <sup>3</sup> ·A <sup>2</sup>
Capacitancia eléctrica	faraday	F	C·V <sup>-1</sup>	m <sup>-2</sup> ·kg <sup>-1</sup> ·s <sup>4</sup> ·A <sup>2</sup>
Densidad de flujo magnético, inductividad magnética	tesla	T	V·s·m <sup>-2</sup>	kg·s <sup>-2</sup> ·A <sup>-1</sup>
Flujo magnético	weber	Wb	V·s	m <sup>2</sup> ·kg·s <sup>-2</sup> ·A <sup>-1</sup>
Inductancia	henrio	H	V·A <sup>-1</sup> ·s	m <sup>2</sup> ·kg·s <sup>-2</sup> ·A <sup>-2</sup>
Ángulo plano	radián	rad		m·m <sup>-1</sup>
Ángulo sólido	estereorradián	sr		m <sup>2</sup> ·m <sup>-2</sup>
Flujo luminoso	lumen	lm	cd·sr	
Iluminancia	lux	lx	cd·sr·m <sup>-2</sup>	
Actividad radiactiva	becquerel	Bq		s <sup>-1</sup>
Dosis de radiación absorbida	gray	Gy	J·kg <sup>-1</sup>	m <sup>2</sup> ·s <sup>-2</sup>
Dosis equivalente	sievert	Sv	J·kg <sup>-1</sup>	m <sup>2</sup> ·s <sup>-2</sup>
Actividad catalítica	katal	kat		mol·s <sup>-1</sup>

## PREFIJOS DE LAS UNIDADES

yotta	Y	10 <sup>24</sup> (un cuatrillón)	deci	d	10 <sup>-1</sup> (un décimo)
zetta	Z	10 <sup>21</sup> (mil trillones)	centi	c	10 <sup>-2</sup> (un centésimo)
exa	E	10 <sup>18</sup> (un trillón)	mili	m	10 <sup>-3</sup> (un milésimo)
peta	P	10 <sup>15</sup> (mil billones)	micro	μ	10 <sup>-6</sup> (un millonésimo)
tera	T	10 <sup>12</sup> (un billón)	nano	n	10 <sup>-9</sup> (un milmillonésimo)
giga	G	10 <sup>9</sup> (mil millones)	pico	p	10 <sup>-12</sup> (un billonésimo)
mega	M	10 <sup>6</sup> (un millón)	femto	f	10 <sup>-15</sup> (un milbillonésimo)
miria	ma	10 <sup>4</sup> (diez mil)	atto	a	10 <sup>-18</sup> (un trillonésimo)
kilo	k	10 <sup>3</sup> (mil)	zepto	z	10 <sup>-21</sup> (un miltrillonésimo)
hecto	h	10 <sup>2</sup> (cien)	yocto	y	10 <sup>-24</sup> (un cuatrillonésimo)
deca	da	10 <sup>1</sup> (diez)			

## Anexo 3.Videos interesantes

### PANTALLA CIRCULAR PERSISTENCE OF VISION A TODO COLOR



Enlace: <http://www.youtube.com/watch?v=QAGnpKz7zvY>

### PANTALLA 3D PERSISTENCE OF VISION



Enlace: <http://www.youtube.com/watch?v=2Dx3eSXieIE>

### RELOJ PERSISTENCE OF VISION



Enlace: <http://www.youtube.com/watch?v=l4F8UbM-1t4>

### PANTALLA VOLUMÉTRICA PERSISTENCE OF VISION



Enlace: <http://www.youtube.com/watch?v=H6AQHsb5sa0>

