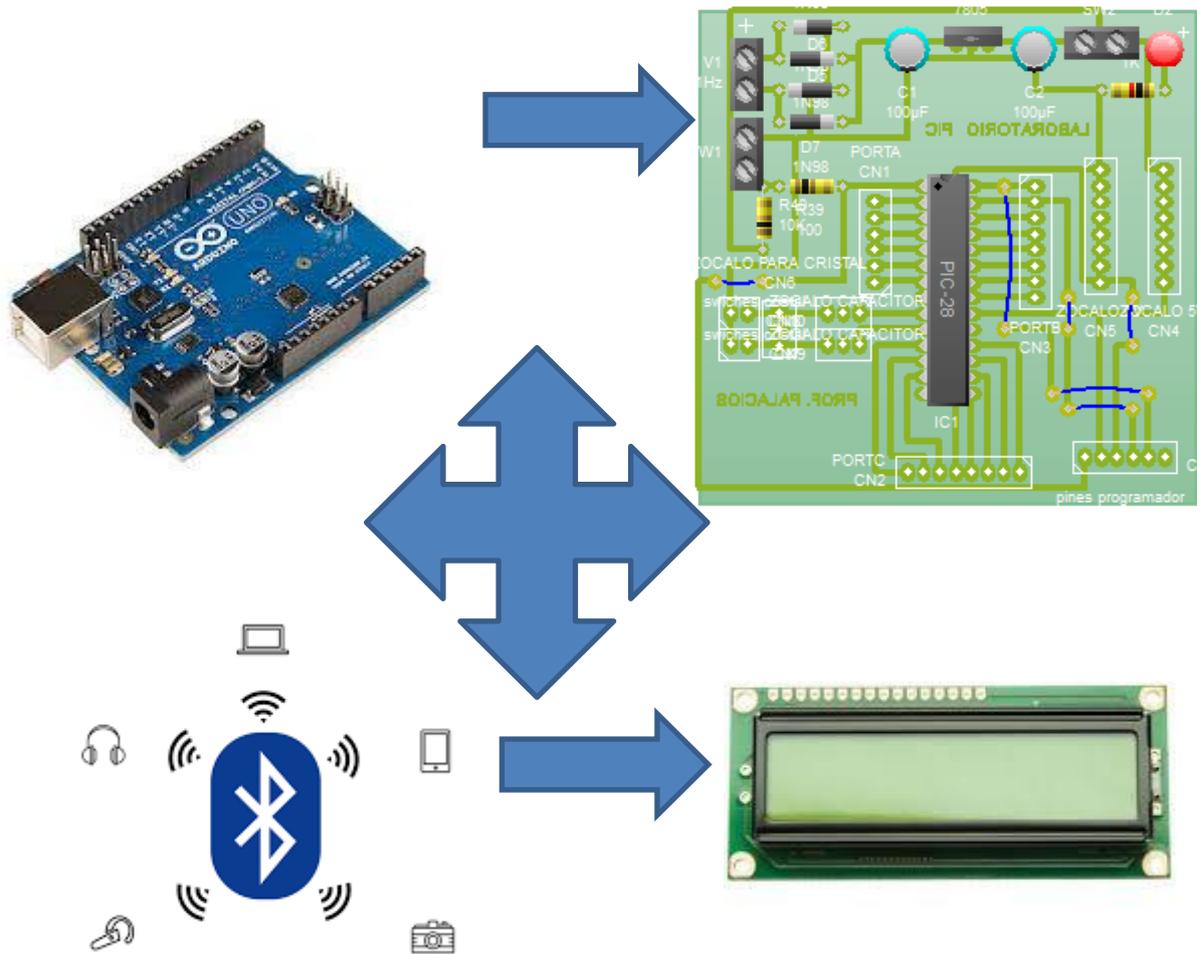


# Manual de introducción a la electrónica básica, digital y el uso de microcontroladores PIC, Arduino del taller de electrónica 5ºB



# WiFi

**Autor:** Palacios José Emanuel

**Títulos:** Técnico Electrónico

**Catedra:** Taller de electrónica de 5ºB

**Fecha:** 20/03/2018



## Índice

Prologo .....	7
Introducción a la electrónica digital .....	8
SEÑALES Y TIPOS .....	8
Señales analógicas:.....	8
Señales digitales: .....	8
VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS.....	8
DIGITALES .....	8
TIPO DE LÓGICA.....	9
SISTEMAS DE NUMERACIÓN .....	9
Sistema de numeración decimal: .....	9
Sistema de numeración binario. ....	9
Conversión entre números decimales y binarios.....	10
SISTEMAS DE NUMERACIÓN OCTAL Y HEXADECIMAL .....	10
Sistema de numeración octal .....	10
SISTEMA DE NUMERACIÓN HEXADECIMAL .....	11
Conversión de números binarios a octales y hexadecimales.....	11
ARITMÉTICA BINARIA .....	12
SUMA EN BINARIO.....	12
SUSTRACCIÓN EN BINARIO.....	12
MULTIPLICACIÓN BINARIA .....	14
DIVISIÓN BINARIA.....	14
COMPUERTAS LÓGICAS.....	15
COMPUERTA AND .....	15
COMPUERTA OR.....	16
COMPUERTA NOT.....	17
COMPUERTA NOR .....	17
COMPUERTA NAND.....	18
COMPUERTA XOR.....	18
COMPUERTA XNOR .....	18
COMPUERTAS BUFFER .....	19
EL TIMER 555.....	21
Descripción del Timer 555:.....	21
El microcircuito 555 es un circuito de tiempo que tiene las siguientes características:.....	22
Sus características más destacables son: .....	22
Configuración de pines.....	23



Multivibrador astable:.....	24
Multivibrador Monostable: .....	25
CD4017 .....	27
Velocidad de operación del CD4017 .....	28
Otros datos del CD4017 .....	28
CD 4017 en modo, secuenciador de punto luminoso .....	29
CD4017 en modo, Secuenciador de agujero. ....	29
CD4017 en cascada.....	30
Actividad integradora:.....	30
CONTADORES DIGITALES.....	31
CARACTERISTICAS IMPORTANTES .....	31
UTILIDAD .....	31
CONTADORES DE RIZADO.....	31
CONTADOR DE RIZADO MODULO- 16.....	32
CONTADORES PARALELOS.....	33
CONTADOR PARALELO DE 3 BITS MOD-8.....	33
OTROS CONTADORES. ....	34
CONTADOR DE RIZADO MOD-6.....	34
CONTADOR DECADA (CUENTA DECENAS).....	35
CONTADORES DESCENDENTES.....	36
CONTADOR DE RIZADO DESCENDENTE DE 3 BITS.....	36
CONTADORES CON CI TTL .....	36
CONTADOR BINARIO DE 4 BITS TTL 7493.....	37
CONTADORES CON CI CMOS. ....	37
CONTADOR CMOS 74HC393 .....	37
CONTADOR CMOS CI 74HC193 .....	38
DIVISION DE FRECUENCIA: EL RELOJ DIGITAL .....	39
DECODIFICADO BINARIO A DYSPLAY 7 SEGMENTO 4511 .....	40
Contador binario CD4518 o CD4520 .....	43
CONEXIÓN EN CASCADA .....	44
Actividad integradora:.....	44
El mundo de los PIC.....	47
Las familias de los PICmicro.....	48
Conocimientos previos básicos .....	48
Conexión de entradas salidas digitales de propósito general de un pic.....	50
Introducción: .....	50



Entradas/salidas digitales en el dsPIC16f873a .....	50
OSCILADOR POR HARDWARE PARA MICROCONTROLADOR.....	50
Programación de los puertos de entrada/salida paralelos .....	51
Funcionamiento como salida: .....	52
Funcionamiento como entrada:.....	52
Características eléctricas de los pines .....	52
Entradas tipo Schmitt-Trigger. ....	52
Interface de las entradas digitales. ....	53
Lectura de un pulsador o interruptor en inversora Pull-Up.....	54
Lectura de un pulsador o interruptor en directa Pull-Down.....	54
Lectura de una entrada de cualquier nivel de tensión con aislamiento inversora .....	54
Lectura de una entrada de cualquier nivel de tensión con aislamiento directa .....	55
Monitorización de corriente en un circuito (Si o no). ....	56
Lectura de una entrada con tensión alterna o sin polaridad aislada. ....	56
Lectura de un teclado.....	57
Interface de las salidas digitales.....	58
Activar un LED .....	58
Activar un LED u otro componente con corriente elevada:.....	58
Activar un relé con bobina de 5 Vcc.....	59
Activar un relé con bobina de más de 5 Vcc .....	60
Salidas en colector abierto:.....	61
Salida colector abierto NPN. ....	61
Salida colector abierto PNP.....	61
Salidas en colector abierto aisladas: .....	62
Salidas con triac:.....	63
Salidas con interruptor basado en MOSFET:.....	64
Manejo de motores DC directos y paso a paso .....	65
Objetivo y resumen del proyecto.....	65
Introducción .....	65
Descripción de los motores paso a paso .....	66
Holding torque .....	67
Detent torque.....	67
Clasificación de los motores paso a paso .....	67
Motores de reluctancia variable .....	67
Motores de imán permanente .....	69
Motores híbridos.....	69



Manual de introducción al uso de microcontroladores PIC ARDUINO	
Motores de imanes permanentes "Claw-Poles" .....	70
Modos de excitación .....	71
Modo paso entero .....	71
Fases excitadas alternativamente .....	71
Fases siempre excitadas .....	72
Modo medio paso .....	72
Manejo de motor DC Puente en H .....	73
El circuito integrado L293 y L293D .....	73
Puente h con transistor NPN .....	74
Puente h NPNsimplificado.....	75
Puente h con transistor NPN y PNP.....	76
Circuitos para programar los pic .....	78
Circuito programador PIP02.....	78
Utilización .....	80
Programador pickit 2 clone .....	80
Utilización.....	84
Ejemplo de conexión del programador para un pic .....	84
Introducción a los sistemas de control.....	85
Definiciones básicas .....	85
Ejemplos de sistema.....	85
Sistemas de control .....	86
Sistemas de control en lazo abierto y en lazo cerrado .....	86
Sistemas de control en lazo abierto .....	86
Sistemas de control en lazo cerrado .....	87
Tipos de control.....	91
Ejemplos de sistemas de control.....	91
Etapas en la realización de un sistema de control .....	93
PROGRAMACIÓN DE LOS PIC 16F87X.....	94
Empezando a programar en ANSEMBLER .....	94
Programa fuente: .....	94
Instrucciones .....	95
UNA MIRADA RAPIDA AL MPLAB .....	96
LAS HERRAMIENTAS DEL MPLAB.....	96
Software ensamblador: .....	96
Ejemplos de programas con ASEMBLER.....	98
Empezando a programar en FLOWCODE .....	101



Escuela de Educación Técnico Profesional N° 602  
Taller de electrónica 5º  
Sección Instrumentación y Control



Manual de introducción al uso de microcontroladores PIC ARDUINO ..... 101

El proceso de diseño..... 101

Diseño de sistemas..... 101

Flowcode en la educación ..... 102

Primera experiencia ..... 102

Ejemplo de programa armado ..... 107

Bibliografía: ..... 109



## Prologo

El objetivo de este documento es proporcionar al alumno información básica necesaria en el Taller de electrónica básica y digital, como por ejemplo, la forma de conectar y alimentar los circuitos integrados, introduciéndonos a los principios elementales del uso de microcontroladores, etc. Se ha sacrificado la rigurosidad por la sencillez, ya que no se pretende exponer conceptos teóricos excesivos, sino dar una guía rápida que sirva de apoyo al alumno en su primer acercamiento a la ELECTRONICA BASICA, pasando por la ELECTRONICA DIGITAL y llegando al manejo de los MICROCONTROLADORES. Las explicaciones, cálculos y razones de lo expuesto en este documento se impartirán en las clases de las materias específicas, tanto como las teórico prácticas como el taller de electrónica y están disponibles en la bibliografía de la asignatura.

Todo circuito electrónico o circuito activo necesita energía eléctrica de características precisas para alimentarse. La electrónica digital básica que se presenta sirve como cimiento al microcontrolador como herramienta fundamental, hoy en día, de los diseños de mediano nivel en el ámbito de la electrónica digital. La potencialidad de los microcontroladores está basada no solo en la tecnología actual que permite integrar a costos muy bajos alta capacidad de procesamiento en un solo chip, sino también en la habilidad del programador para incorporar en un circuito integrado funciones de alta potencialidad que permitan al microcontrolador transformarse en un sistema con altísimo valor agregado.

Los pasos a seguir para lograr diseños óptimos con sistemas digitales de bajo costo y alta performance basados en microcontroladores comienzan con el estudio de las características funcionales y estructurales que tiene un dispositivo tal o familia de ellos. Son muchos y muy variados los sistemas microcontroladores que podemos encontrar en el mercado por lo que se debió seleccionar algunos de ellos para realizar un estudio detallado y tratar, con esto, de introducir al estudiante en la materia.

No solo conocer a fondo algunos modelos de microcontroladores resuelve un proyecto. También debe tenerse la suficiente práctica en la confección de software para potenciar lo que sin él sería un circuito integrado inútil. Este texto está dedicado a la descripción detallada de algunos modelos de microcontroladores, con algunos ejemplos de programación desde pequeños retardos de tiempo hasta sistemas algo complejos como contadores u osciladores de precisión que ayudarán a generar una programación más elaborada para aquellos que comienzan a transitar este apasionante tema.

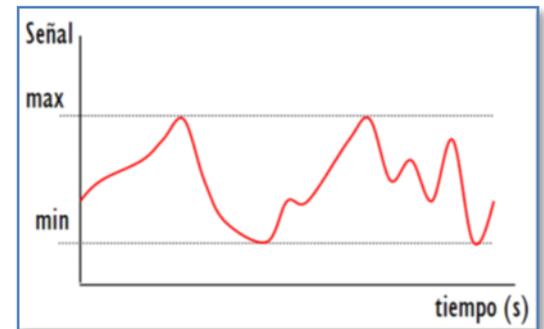


## Introducción a la electrónica digital

### SEÑALES Y TIPOS

Como vimos en el tema anterior, la electrónica es la rama de la ciencia que se ocupa del estudio de los circuitos y de sus componentes que permiten modificar la corriente eléctrica amplificándola, atenuándola, rectificándola y filtrándola y que aplica la electricidad al tratamiento de la información. Por otro lado el término digital deriva de la forma en que las computadoras realizan las operaciones; i.e. contando dígitos o números.

Una **señal** es la variación de una magnitud que permite transmitir información. Las señales pueden ser de dos tipos:



#### Señales analógicas:

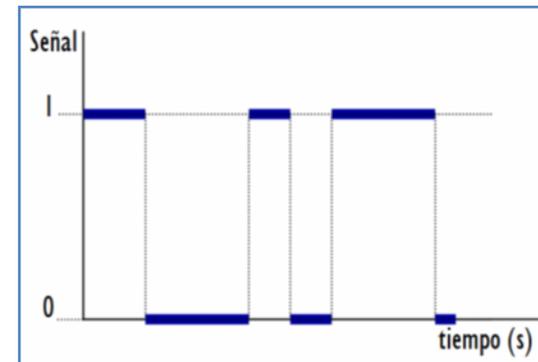
Aquellas donde la señal puede adquirir infinitos valores entre dos extremos cualesquiera. La variación de la señal forma una gráfica continua. La mayoría de las magnitudes en la naturaleza toman valores continuos, por ejemplo la temperatura. Para pasar de 20 a 25°C, la temperatura irá tomando los infinitos valores entre 20 y 25°C.

#### Señales digitales:

Las cuales pueden adquirir únicamente valores concretos; i.e. no varían de manera continua.

Para nosotros los sistemas digitales que tienen mayor interés, por ser los que se pueden implementar electrónicamente, son los sistemas binarios. Un **sistema binario** es aquel en el que las señales sólo pueden tomar dos valores, que representaremos de ahora en adelante con los símbolos 0 y 1. Por ejemplo, el estado de una bombilla sólo puede tener dos valores (0 apagada, 1 encendida).

A cada valor de una señal digital se le llama **bit** y es la unidad mínima de información.



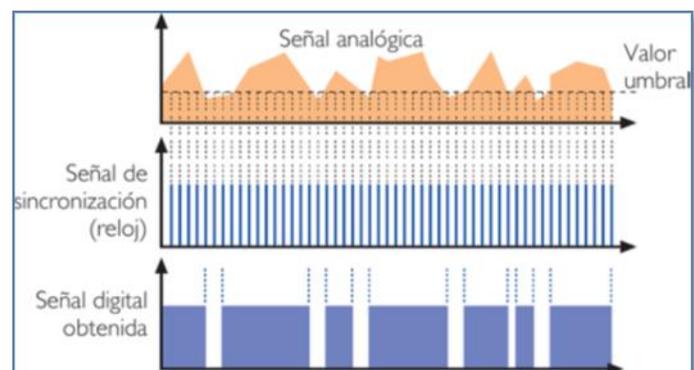
## VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS

### DIGITALES

El mejor argumento a favor de la mayor flexibilidad de los sistemas digitales se encuentra en los actuales ordenadores o computadoras digitales, basados íntegramente en diseños y circuitos digitales. Las principales ventajas de los sistemas digitales respecto a los analógicos son:

- Mayor facilidad de diseño, pues las técnicas están bien establecidas.
- El ruido (fluctuaciones de tensión no deseadas) afecta menos a los datos digitales que a los analógicos, ya que en sistemas digitales sólo hay que distinguir entre valor alto y valor bajo.
- Las operaciones digitales son mucho más precisas y la transmisión de señales es más fiable porque utilizan un conjunto discreto de valores, fácil de diferenciar entre sí, lo que reduce la probabilidad de cometer errores de interpretación.
- Almacenamiento de la información menos costoso

Los sistemas digitales presentan el inconveniente de que para transmitir una señal analógica debemos hacer un muestreo de la señal, codificarla y posteriormente transmitirla en formato digital y repetir el proceso inverso. Para conseguir obtener la señal analógica original todos estos pasos deben hacerse muy rápidamente (aunque los sistemas electrónicos digitales actuales trabajan a velocidades lo suficientemente altas como para realizarlo y obtener resultados satisfactorios).



Conversión de señal analógica a señal digital. Si el valor de la señal en ese instante está por debajo de un determinado umbral, la señal digital toma un valor mínimo (0). Cuando la señal analógica se encuentra por encima del valor umbral, la señal digital toma un valor máximo (1).



## TIPO DE LÓGICA



En los circuitos electrónicos digitales se emplean niveles de tensión distintos para representar los dos bits. Las tensiones que se utilizan para representar los unos y los ceros se les denominan niveles lógicos. Existen distintos tipos de lógica.

- **Lógica positiva:** al nivel alto se le da el valor de 1 y al nivel bajo un valor de 0 ( $V_H = 1$  y  $V_L = 0$ )
- **Lógica negativa:** al nivel alto se le da el valor 0 y al nivel bajo un valor de 1 ( $V_H = 1$  y  $V_L = 0$ ).
- **Lógica mixta:** se mezclan ambos criterios en el mismo sistema, eligiendo uno u otro según convenga.

Nosotros trabajaremos con la lógica positiva.

## SISTEMAS DE NUMERACIÓN

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos. La norma principal en un sistema de numeración posicional es que **un mismo símbolo tiene distinto valor según la posición que ocupe**.

### Sistema de numeración decimal:

El sistema de numeración que utilizamos habitualmente es el **decimal**, que se compone de diez símbolos o dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) a los que otorga un valor **dependiendo de la posición** que ocupen en la cifra: unidades, decenas, centenas, millares, etc.

El valor de cada dígito está asociado al de una potencia de base 10, número que coincide con la cantidad de símbolos o dígitos del sistema decimal, y un exponente igual a la posición que ocupa el dígito menos uno, contando desde la derecha.

En este sistema el número 528, por ejemplo, significa:

5 centenas + 2 decenas + 8 unidades, es decir:

500 + 20 + 8 o, lo que es lo mismo,

$$5 \cdot 10^2 + 2 \cdot 10^1 + 8 \cdot 10^0 = 528$$

En el caso de números con decimales, la situación es análoga aunque, en este caso, algunos exponentes de las potencias serán negativos, concretamente el de los dígitos colocados a la derecha del separador decimal. Por ejemplo, el número 8245,97 se calcularía como:

8 millares + 2 centenas + 4 decenas + 5 unidades + 9 décimos + 7 céntimos

8000 + 200 + 40 + 5 + 0,9 + 0,07 = 8245,97

$$8 \cdot 10^3 + 2 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 + 9 \cdot 10^{-1} + 7 \cdot 10^{-2} = 8245,97$$

### Sistema de numeración binario.

El sistema de numeración binario utiliza sólo dos dígitos, el cero (0) y el uno (1), que tienen distinto valor dependiendo de la posición que ocupen. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario 1011 tiene un valor que se calcula así:

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11$$



Manual de introducción al uso de microcontroladores PIC ARDUINO  
y lo escribimos así:

$$1011_2 = 11_{10}$$

### Conversión entre números decimales y binarios

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar divisiones sucesivas por 2 y colocar los restos obtenidos, en cada una de ellas. Para formar el número binario tomaremos los restos **en orden inverso** al que han sido obtenidos. Por ejemplo:

$$\begin{array}{l} 77 : 2 = 38 \text{ Resto: } 1 \\ 38 : 2 = 19 \text{ Resto: } 0 \\ 19 : 2 = 9 \text{ Resto: } 1 \\ 9 : 2 = 4 \text{ Resto: } 1 \\ 4 : 2 = 2 \text{ Resto: } 0 \\ 2 : 2 = 1 \text{ Resto: } 0 \\ 1 : 2 = 0 \text{ Resto: } 1 \end{array}$$

$$77_{10} = 1001101_2$$

La cantidad de dígitos necesarios, para representar un número en el sistema binario, dependerá del valor de dicho número en el sistema decimal. En el caso anterior, para representar el número 77 han hecho falta siete dígitos. Para representar números superiores harán falta más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de ocho dígitos, porque  $2^8=256$  y, por tanto, 255 es el número más grande que puede representarse con ocho dígitos.

Es importante distinguir entre los números que pueden representarse con  $n$  dígitos binarios, que es  $2^n$ , y el mayor de esos números, que es una unidad menos, es decir,  $2^n - 1$ .

$$1010011 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 83$$

$$1010011_2 = 83_{10}$$

El proceso para convertir un número del sistema binario al decimal es aún más sencillo; basta con desarrollar el número, teniendo en cuenta que el valor de cada dígito está asociado a una potencia de 2, cuyo

exponente es 0 en el bit situado más a la derecha, y se incrementa en una unidad según vamos avanzando posiciones hacia la izquierda, tal y como se muestra en el siguiente ejemplo:

### SISTEMAS DE NUMERACIÓN OCTAL Y HEXADECIMAL

El inconveniente de la codificación binaria es que la representación de algunos números resulta muy larga. Por este motivo se utilizan otros sistemas de numeración que resulten más cómodos de escribir: el sistema octal y el sistema hexadecimal. Afortunadamente, resulta muy fácil convertir un número binario a octal o a hexadecimal.

#### Sistema de numeración octal

En el sistema octal, los números se representan mediante **ocho** dígitos diferentes: 0, 1, 2, 3, 4, 5, 6 y 7. Cada dígito tiene, naturalmente, un valor distinto dependiendo del lugar que ocupen. El valor de cada una de las posiciones viene determinado por las potencias de base 8. La conversión de un número decimal a octal, y viceversa, se realiza del mismo modo que la de los números binarios, aunque, lógicamente, se emplea como base el número 8 en vez del 2.

La conversión de un número decimal a octal se hace del mismo modo: mediante divisiones sucesivas por 8 y colocando los restos obtenidos en orden inverso. Por ejemplo:

$$\begin{array}{l} 122 : 8 = 15 \text{ Resto: } 2 \\ 15 : 8 = 1 \text{ Resto: } 7 \\ 1 : 8 = 0 \text{ Resto: } 1 \end{array}$$

$$122_{10} = 172_8$$

La conversión de un número octal a decimal es igualmente sencilla.



Manual de introducción al uso de microcontroladores PIC ARDUINO  
Por ejemplo:

$$237_8 = 2 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0 = 128 + 24 + 7 = 159_{10}$$

$$237_8 = 159_{10}$$

## SISTEMA DE NUMERACIÓN HEXADECIMAL

En este sistema, los números se representan con dieciséis símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Se utilizan los caracteres A, B, C, D, E y F representando las cantidades decimales 10, 11, 12, 13, 14 y 15 respectivamente, porque no hay dígitos mayores que 9 en el sistema decimal. El valor de cada uno de estos símbolos depende, como es lógico, de su posición, que se calcula mediante potencias de base 16.

Ensayemos la  
conversión  
decimal a  
hexadecimal del

$$1735 : 16 = 108$$

Resto: 7

$$108 : 16 = 6$$

Resto: C (12<sub>10</sub>)

$$6 : 16 = 0$$

Resto: 6

$$1735_{10} = 6C7_{16}$$

$$1A3F_{16} = 1 \cdot 16^3 + A \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0 = 6719_{10}$$

$$1A3F_{16} = 6719_{10}$$

número 1735:

Ensayemos también la conversión inversa, de hexadecimal a decimal del número 1A3F:

## Conversión de números binarios a octales y hexadecimales

Cada dígito de un número octal equivale a tres dígitos en el sistema binario. Por tanto, el modo de convertir un número entre estos sistemas de numeración equivale a "expandir" cada dígito octal a tres dígitos binarios, o en "contraer" grupos de tres caracteres binarios a su correspondiente dígito octal. Por ejemplo:

$$101001011_2 = 513_8$$

$$750_8 = 111101000_2$$

$$101001110011_2 = A73_{16}$$

$$1F6_{16} = 00011110110_2$$

Análogamente, la conversión entre números hexadecimales y binarios se realiza "expandiendo" o "contrayendo" cada dígito hexadecimal a cuatro dígitos binarios. Por ejemplo:

En caso de que los dígitos binarios no formen grupos completos (de tres o cuatro dígitos, según corresponda), se deben añadir ceros a la izquierda hasta completar el último grupo. Por ejemplo:

$$101110_2 = 00101110_2 = 2E_{16}$$



Manual de introducción al uso de microcontroladores PIC ARDUINO

**ARITMÉTICA BINARIA**

La Unidad Aritmético Lógica, en la CPU del procesador, es capaz de realizar operaciones aritméticas, con datos numéricos expresados en el sistema binario. Naturalmente, esas operaciones incluyen la adición, la sustracción, el producto y la división. Las operaciones se hacen del mismo modo que en el sistema decimal, pero debido a la sencillez del sistema de numeración, pueden hacerse algunas simplificaciones que facilitan mucho la realización de las operaciones.

**SUMA EN BINARIO**

La tabla de sumar, en binario, es mucho más sencilla que en decimal. Sólo hay que recordar cuatro combinaciones posibles.

Recuerda que en el sistema decimal había que memorizar unas 100 combinaciones.

Las sumas 0+0, 0+1 y 1+0 son evidentes:

SUMA	0	1
0	0	1
1	1	0 + a

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1

Pero la suma de 1+1, que sabemos que es 2, debe escribirse en binario con dos cifras (10) y, por tanto 1+1 es 0 y se arrastra una unidad, que se suma a la posición siguiente a la izquierda.

Veamos algunos ejemplos:

010	$2_{10}$	001101	$13_{10}$
101	$5_{10}$	100101	$37_{10}$
111	$7_{10}$	110010	$50_{10}$
1011011	$91_{10}$	110111011	$443_{10}$
1011010	$90_{10}$	100111011	$315_{10}$
10110101	$181_{10}$	1011110110	$758_{10}$

**SUSTRACIÓN EN BINARIO**

Restar en binario es, nuevamente, igual que la misma operación en el sistema decimal. Pero conviene repasar la operación de restar en decimal

para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman minuendo, sustraendo y diferencia.

RESTA	0	1
0	0	1
1	1 + a	0

Las sumas 0-0, 1-0 y 1-1 son evidentes:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0

La resta 0 - 1 se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente:  $10 - 1$ , es decir,  $210 - 110 = 1$

111	$7_{10}$	10001	$17_{10}$
101	$5_{10}$	01010	$10_{10}$
010	$2_{10}$	00111	$7_{10}$
11011001	$217_{10}$	111101001	$489_{10}$
10101011	$171_{10}$	101101101	$365_{10}$
00101110	$46_{10}$	001111100	$124_{10}$

Esa unidad prestada debe devolverse, sumándola, a la posición siguiente. Veamos algunos ejemplos:

A pesar de lo sencillo que es el procedimiento de restar, es fácil confundirse. Tenemos interiorizado el sistema decimal y hemos aprendido a

restar mecánicamente, sin detenernos a pensar en el significado del arrastre. Para simplificar las restas y reducir la posibilidad de cometer errores hay varias soluciones:



Manual de introducción al uso de microcontroladores PIC ARDUINO

➤ Dividir los números largos en grupos. En el siguiente ejemplo, vemos cómo se divide una resta larga en tres restas cortas:

$$\begin{array}{r} 100110011101 \\ 010101110010 \\ \hline 010000101011 \end{array} = \begin{array}{r} 1001 \\ 0101 \\ \hline 0100 \end{array} \quad \begin{array}{r} 1001 \\ 0111 \\ \hline 0010 \end{array} \quad \begin{array}{r} 1101 \\ 0010 \\ \hline 1011 \end{array}$$

➤ Utilizando el **Complemento a dos**

**Complemento a dos**

El **complemento a dos** de un número  $N$ , con  $n$  cifras, se define como  $C_2^N = 2^n - N$ .

Veamos un ejemplo: tomemos el número  $N=1011012$  que tiene 6 cifras, y calculemos el complemento a dos de ese número:

$$N = 45_{10} \quad n = 6 \quad 2^6 = 64 \quad \text{y, por tanto: } C_2^N = 64 - 45 = 19 = 010011_2$$

**Complemento a uno**

El **complemento a uno** de un número  $N$ , con  $n$  cifras es, por definición, una unidad menor que el complemento a dos, es decir:

$$C_1^N = C_2^N - 1 \quad \text{y, por la misma razón, } C_2^N = C_1^N + 1$$

Calculemos el **complemento a uno** del mismo número del ejemplo anterior:

$$C_1^N = C_2^N - 1 \quad \begin{array}{r} 010011 \\ 000001 \\ \hline 010010 \end{array} \quad C_1^N = 010010$$

Da la sensación de que no va a ser más sencillo restar utilizando el complemento a dos, porque el procedimiento para calcular el complemento a dos es más difícil y laborioso que la propia resta. Pero es mucho más sencillo de lo que parece.

Si  $N=101101$   
su **complemento a uno** es:  $C_1^N = 010010$   
y su **complemento a dos** es:  $C_2^N = C_1^N + 1 = 010011$

En realidad, el **complemento a uno** de un número binario es el número resultante de invertir UNOS y CEROS.

Veamos otro ejemplo de cálculo de complementos:

Si  $N=011011010$

$$\begin{array}{l} \text{El complemento a uno es: } C_1^N = 1001001010 \quad C_1^N = 1001001010 \\ \text{y el complemento a dos es: } C_2^N = 1001001011 \end{array}$$

**Restar en binario usando el complemento a dos**

Y, por fin, vamos a ver cómo facilita la resta el complemento. La resta binaria de dos números puede obtenerse **sumando al minuendo el complemento a dos del sustraendo**.

Veamos algunos ejemplos:

$$\begin{array}{r} 1011011 \\ 0101110 \\ \hline 0101101 \end{array} \quad \begin{array}{r} 91_{10} \\ 46_{10} \\ \hline 45_{10} \end{array}$$

a) Hagamos la siguiente resta,  $91 - 46 = 45$ , en binario:

$$\begin{array}{r} 1011011 \\ 1010010 \\ \hline 10101101 \end{array}$$

Tiene alguna dificultad, cuando se acumulan los arrastres a la resta siguiente. Pero esta misma resta puede hacerse como una suma, utilizando el complemento a dos del sustraendo:

En el resultado nos sobra un bit, que se desborda por la izquierda. Como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

b) Hagamos esta otra resta,  $219 - 23 = 196$ , utilizando el complemento a dos:

$$\begin{array}{r} 219_{10} = 11011011_2 \\ 23_{10} = 00010111_2 \\ \hline 11001000 \end{array} \quad C_2^{23} = 11101001 \quad \begin{array}{r} 11011011 \\ 11101001 \\ \hline 111000100 \end{array}$$

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto:

$$11000100_2 = 196_{10}$$

¡Qué fácil!

### MULTIPLICACIÓN BINARIA

La multiplicación en binario es más fácil que en cualquier otro sistema de numeración.

Como los factores de la multiplicación sólo pueden ser CEROS o UNOS, el producto sólo puede ser CERO o UNO. En otras palabras, la tabla de multiplicar es muy fácil de aprender

En un ordenador, sin embargo, la operación de multiplicar se realiza mediante sumas repetidas. Eso crea algunos problemas en la programación porque cada suma de dos UNOS origina un arrastre, que se resuelven contando el número de UNOS y de arrastres en cada columna. Si el número de UNOS es par, la suma es un CERO y si es impar, un UNO. Luego, para determinar los arrastres a la posición superior, se cuentan las parejas de UNOS.

POR	0	1
0	0	0
1	0	1

### DIVISIÓN BINARIA

Igual que en el producto, la división es muy fácil de realizar, porque no son posibles en el cociente otras cifras que UNOS y CEROS.

Consideremos el siguiente ejemplo,  $42 : 6 = 7$ , en binario:

(Dividendo)	1 0 1 0 1 0	110 (Divisor)
	- 1 1 0	111 (Cociente)
	1 0 0 1	
	- 1 1 0	
	0 1 1 0	
	1 1 0	
	0 0 0	

Se intenta dividir el dividendo por el divisor, empezando por tomar en ambos el mismo número de cifras (100 entre 110, en el ejemplo). Si no puede dividirse, se intenta la división tomando un dígito más (1001 entre 100).

Si la división es posible, entonces, el divisor sólo podrá estar contenido **una vez** en el dividendo, es decir, la primera cifra del cociente es un UNO. En

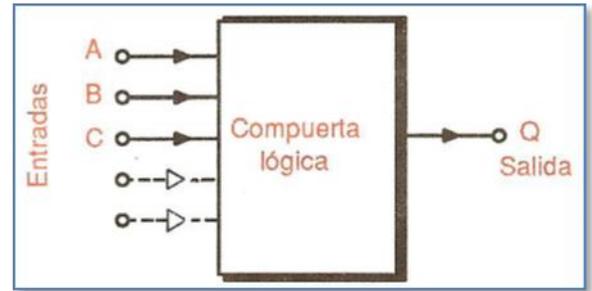
ese caso, el resultado de multiplicar el divisor por 1 es el propio divisor. Restamos las cifras del dividendo del divisor y bajamos la cifra siguiente.

El procedimiento de división continúa del mismo modo que en el sistema decimal.

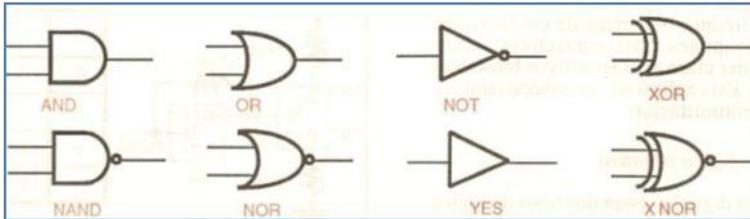
## COMPUERTAS LÓGICAS

Las compuertas o *gates*, son los bloques básicos de cualquier circuito digital. Todas las configuraciones circuitales digitales están formadas por compuertas.

Una compuerta digital (figura 1) es un circuito electrónico con dos o más líneas de entrada y una línea de salida, que tiene la característica de tomar decisiones. Esto consiste en situar su salida en 0 o en 1, dependiendo del estado de sus entradas y de la función lógica para la cual ha sido *FIGURA 1*



diseñada.



Todos los sistemas digitales se construyen utilizando tres compuertas lógicas básicas: AND, OR y NOT. Otras son NAND (o NO-AND), NOR (o NO-OR), OR EXCLUSIVA (XOR) y la NOR EXCLUSIVA (XNOR).

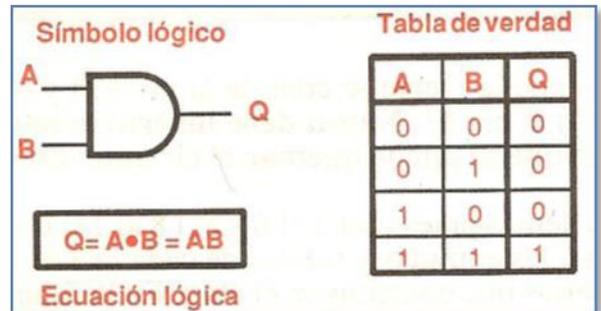
¿Cómo describir la operación de una compuerta?

La operación de una compuerta lógica se puede expresar mediante una tabla de verdad, una ecuación lógica o un diagrama de temporización.

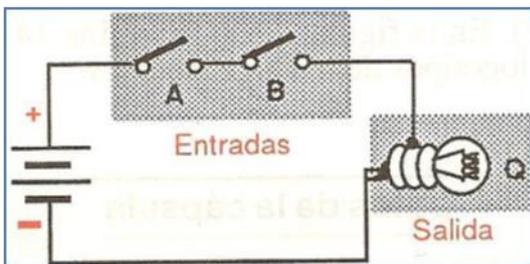
Una *tabla de verdad* representa ordenadamente todas las posibles combinaciones de estados lógicos que pueden existir en las entradas y el valor que toma la salida en cada caso.

La *ecuación lógica* relaciona matemáticamente la salida con las entradas.

Un *diagrama de temporización*, representa gráficamente el comportamiento de una compuerta con señales variables en el tiempo.



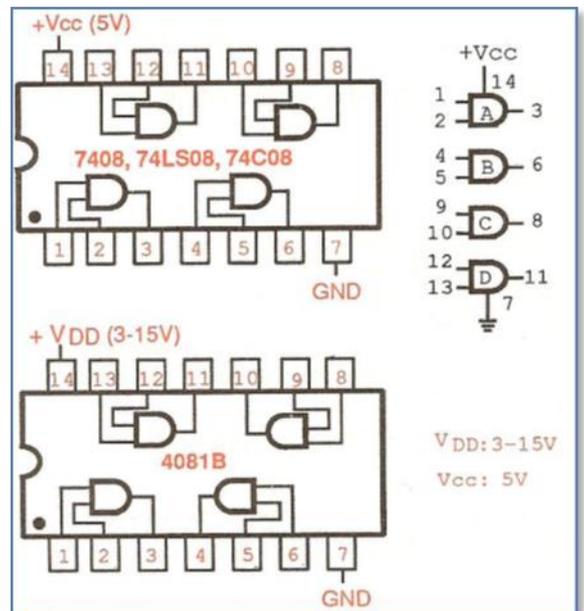
## COMPUERTA AND



EQUIVALENTE DE LA COMPUERTA AND

## COMPUERTA AND DE DOS ENTRADAS

### CIRCUITO

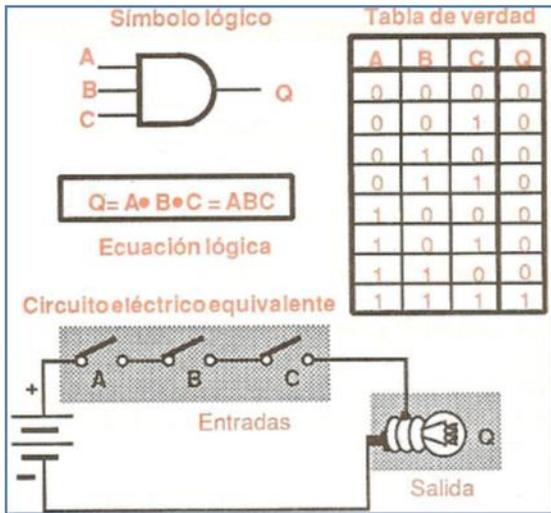


## COMPUERTAS AND DE DOS ENTRADAS INTEGRADAS (TTL/CMOS)

La operación realiza por la compuerta AND es el **PRODUCTO LOGICO**

Una compuerta AND de dos o más entradas entrega un nivel alto o 1 lógico en su salida cuando todas sus entradas

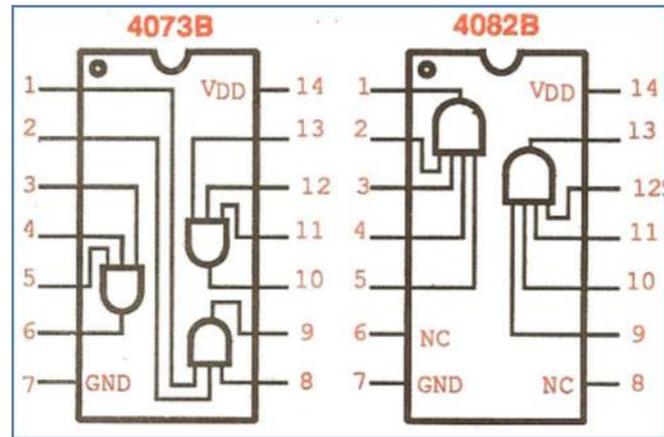
Manual de introducción al uso de microcontroladores PIC ARDUINO  
están en nivel alto y un nivel bajo o 0 lógico, cuando por lo menos una de ellas, o todas, están en nivel bajo.



COMPUERTA **AND** DE TRES ENTRADAS

Los siguientes son algunos de los circuitos TTL y CMOS que contienen compuertas AND de tres entradas:

-Dispositivos de las series 40 y 74C son de tecnología CMOS y de las series 74 y 74LS son de tecnología TTL -4073B, 7411, 74LS11

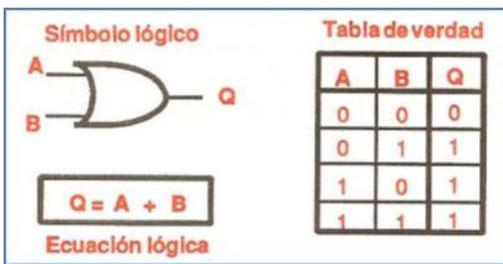


Para cuatro entradas: -4082B, 7421, 74LS21, dos compuertas AND de cuatro entradas.

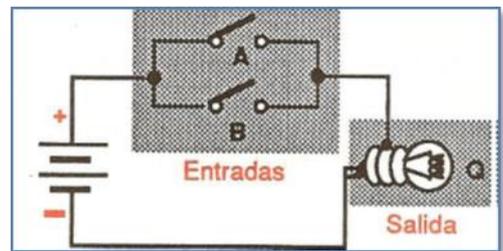
COMPUERTAS **AND** INTEGRADAS DE 3 Y 4 ENTRADAS

COMPUERTA **OR**

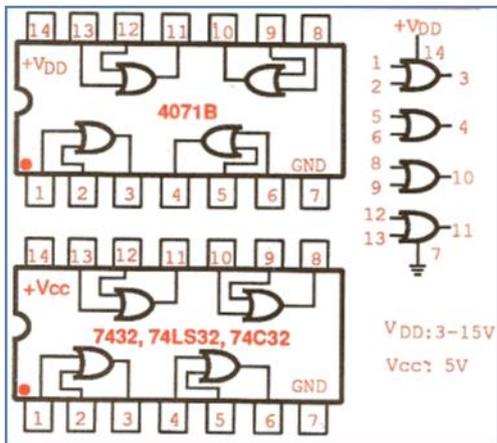
COMPUERTA **OR** DE DOS ENTRADAS



CIRCUITO EQUIVALENTE DE LA COMPUERTA **OR**



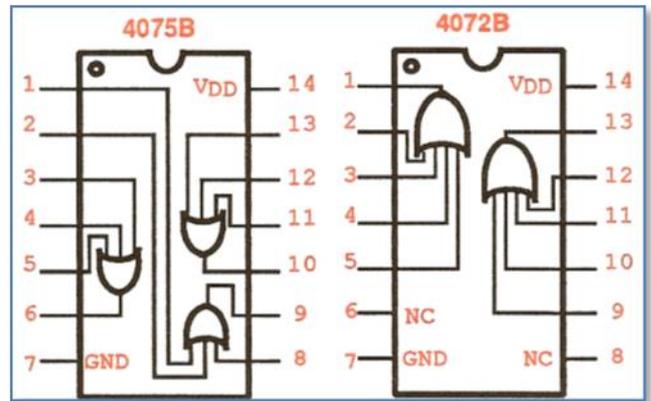
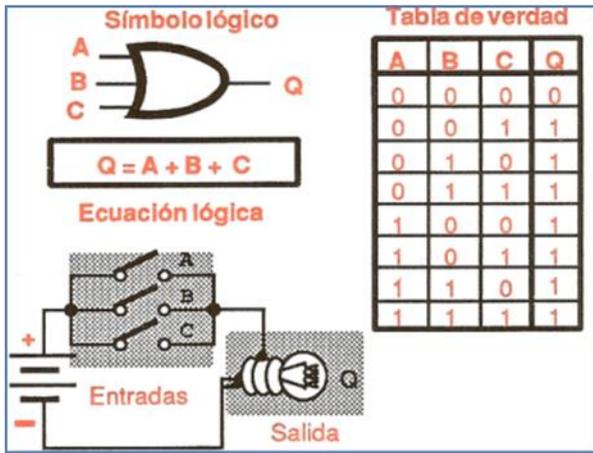
COMPUERTAS **OR** INTEGRADAS DE DOS ENTRADAS (TTL/CMOS)



La operación realizada por la compuerta **OR** es la **SUMA LOGICA**

Una compuerta OR de dos o más entradas entrega un nivel bajo en una salida cuando todas sus entradas están en nivel bajo y un nivel alto cuando por lo menos una de ellas, o todas, están en nivel alto.

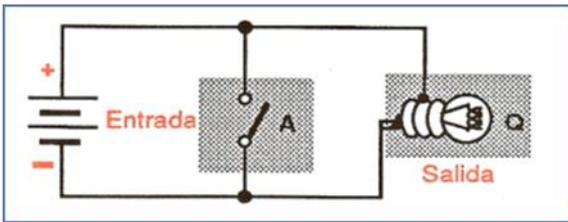
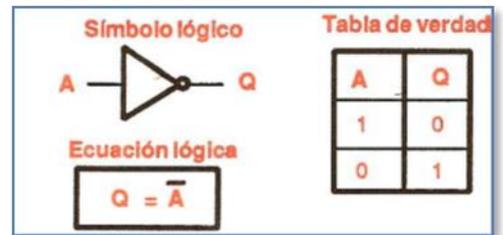
A continuación se mencionan algunos circuitos integrados que contienen compuertas OR de varias entradas. Por tratarse de circuitos con tecnología CMOS, operan con tensiones entre 3 a 15 volts.



COMPUERTAS OR INTEGRADAS DE 3 Y 4 ENTRADAS

COMPUERTA NOT

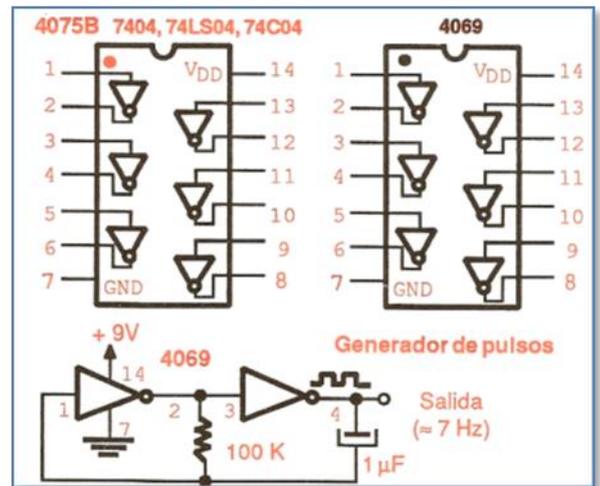
Un *inversor*, niega o complementa el nivel lógico de la señal de entrada.



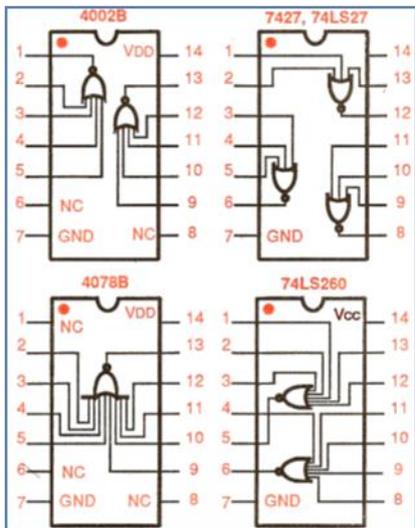
CIRCUITO

EQUIVALENTE DE UNA COMPUERTA NOT

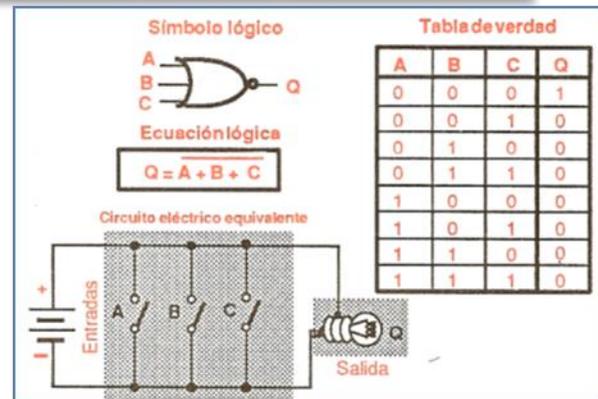
INVERSORES INTEGRADOS



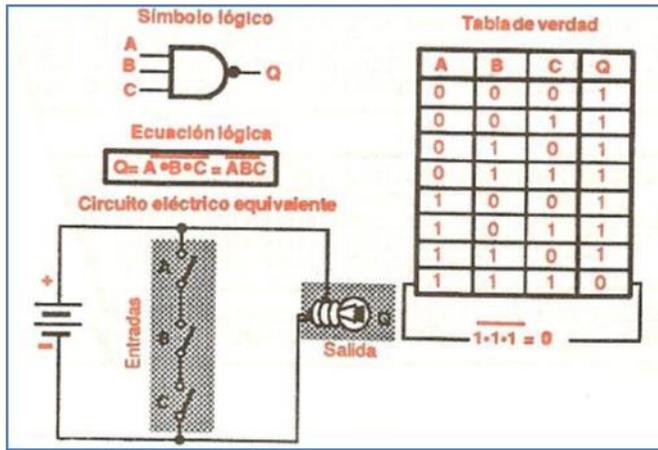
COMPUERTA NOR



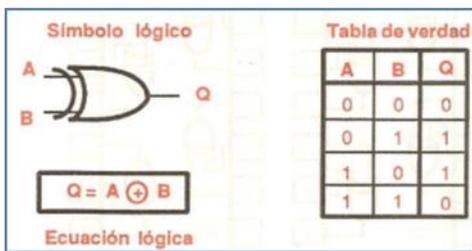
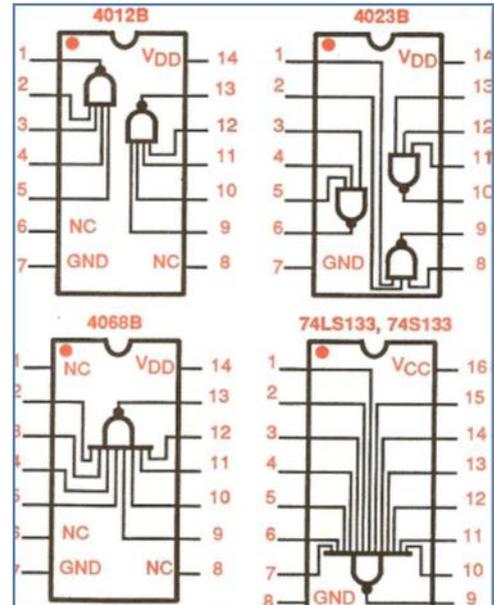
COMPUERTAS NOR DE VARIAS ENTRADAS



## COMPUERTA NAND



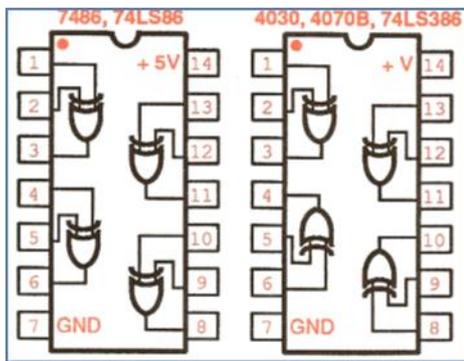
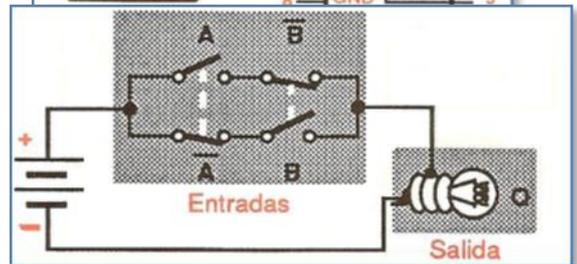
COMPUERTAS NAND DE VARIAS ENTRADAS



## COMPUERTA XOR

Una compuerta OR EXCLUSIVA es un dispositivo digital con dos líneas de entrada y una línea de salida que entrega una salida

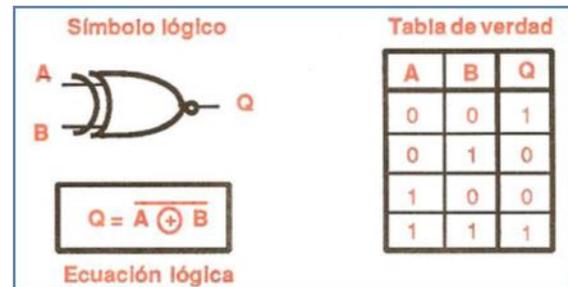
en nivel alto cuando una de sus entradas están en nivel bajo y la otra en nivel alto y una salida en nivel bajo, cuando sus entradas están en nivel alto (ambas) o en nivel bajo.



EQUIVALENTE ELECTRIC DE LA COMPUERTA XOR

COMPUERTAS XOR DE DOS ENTRADAS

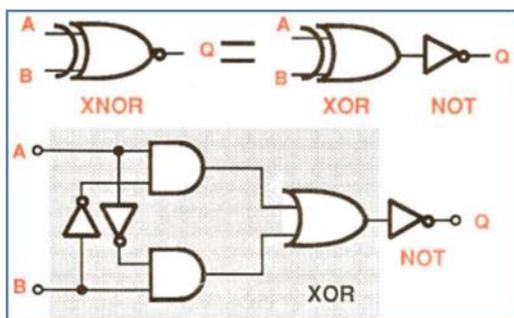
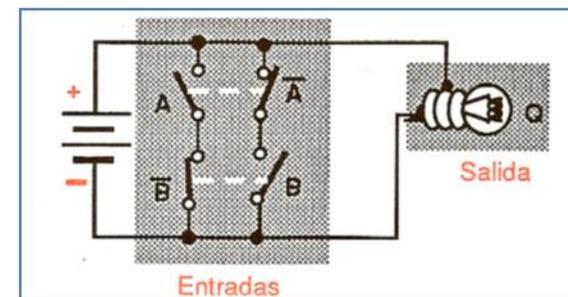
## COMPUERTA XNOR



EQUIVALENTE XNOR

CIRCUITO

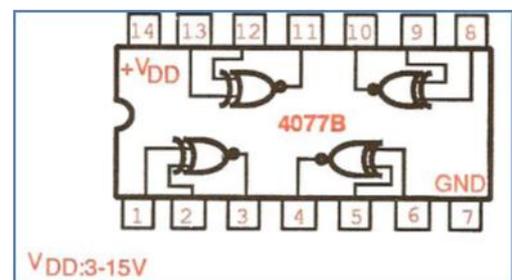
LOGICO



CIRCUITO EQUIVALENTE XNOR

INTEGRADA

COMPUERTA XNOR



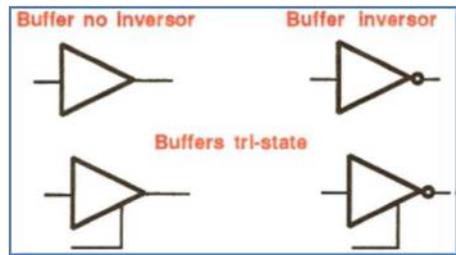
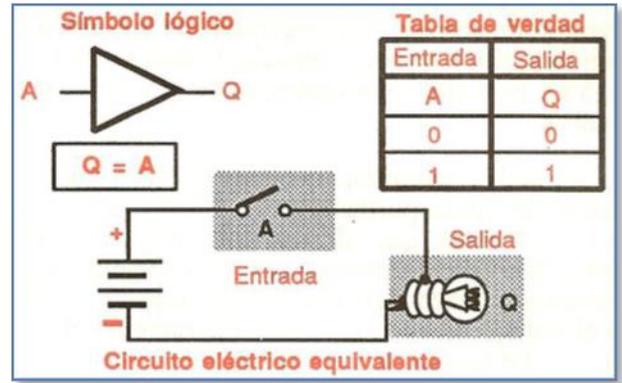
Manual de introducción al uso de microcontroladores PIC ARDUINO

**COMPUERTAS BUFFER**

Los *buffers* o separadores, son compuertas con una alta capacidad de corriente de salida. Esta característica le permite manejar directamente leds, relés y otras cargas que no pueden ser conectadas directamente a compuertas comunes.

Se utilizan principalmente como amplificadores de corriente. Un *buffer* a la salida de un circuito integrado digital aumenta su fan-out, es decir, la máxima corriente de salida que este puede suministrar.

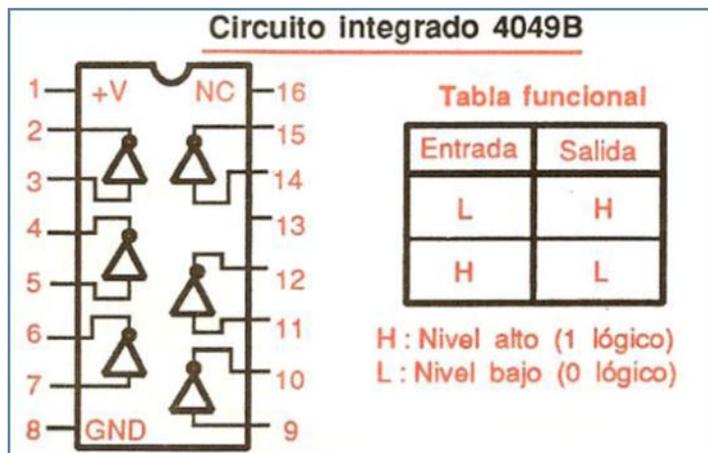
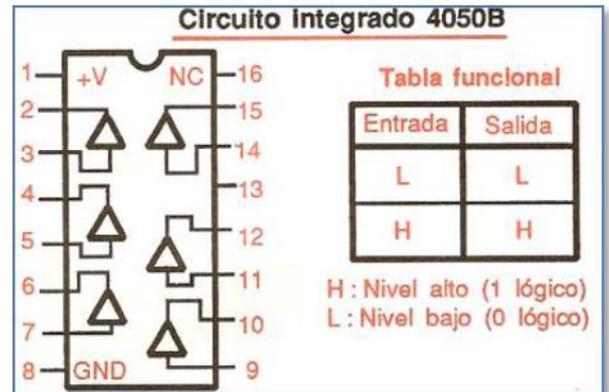
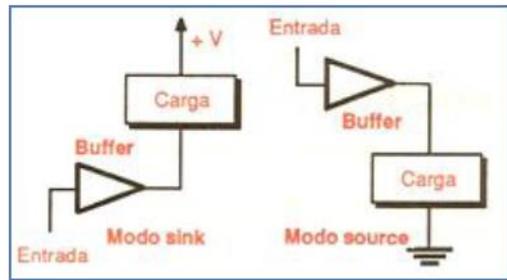
Existen dos clases de *buffers*, los inversores y los no inversores.



**BUFFERS INVERSORES Y NO INVERSORES**

Los *buffers* no inversores entregan el mismo nivel lógico que reciben.

Un *buffer* se puede conectar a una carga de dos formas: como disipador de corriente (modo sink), conectándose la carga entre la salida y el positivo de la fuente, o como fuente de corriente (modo source), conectándose la carga entre la salida y tierra.



# RESUMEN DE COMPUTERTAS DIGITALES

La siguiente tabla resume los símbolos lógicos, las ecuaciones lógicas, las tablas de verdad y los circuitos eléctricos equivalentes de las compuertas AND, OR, NOT, YES, NAND, NOR, XOR y XNOR. También se incluyen los símbolos de las compuertas especiales tipos Schmitt-trigger, colector abierto y tri-state así como la simbología ANSI/IEEE, de gran aceptación.

	AND	OR	NOT	YES	NAND	NOR	XOR	XNOR																																																																																								
<b>Símbolo lógico</b>																																																																																																
<b>Ecuación lógica</b>	$Q = A \cdot B = AB$	$Q = A + B$	$Q = \bar{A}$	$Q = A$	$Q = \overline{A \cdot B}$	$Q = \overline{A + B}$	$Q = A \oplus B$	$Q = A \oplus B$																																																																																								
<b>Tabla de verdad</b>	<table border="1"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Q	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>A</td><td>Q</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	Q	0	1	1	0	<table border="1"> <tr><td>A</td><td>Q</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	A	Q	0	0	0	0	1	1	<table border="1"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Q	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <tr><td>A</td><td>B</td><td>Q</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Q																																																																																														
0	0	0																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														
A	B	Q																																																																																														
0	0	0																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	1																																																																																														
A	Q																																																																																															
0	1																																																																																															
1	0																																																																																															
A	Q																																																																																															
0	0																																																																																															
0	0																																																																																															
1	1																																																																																															
A	B	Q																																																																																														
0	0	1																																																																																														
0	1	1																																																																																														
1	0	1																																																																																														
1	1	0																																																																																														
A	B	Q																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	0																																																																																														
A	B	Q																																																																																														
0	0	1																																																																																														
0	1	0																																																																																														
1	0	0																																																																																														
1	1	1																																																																																														
<b>Circuito eléctrico equivalente</b>																																																																																																

SIMBOLOS ESPECIALES				
<b>Schmitt trigger</b>				
<b>Colector abierto</b>				
<b>Tri-state</b>				
<b>ANSI/IEEE (**)</b>				

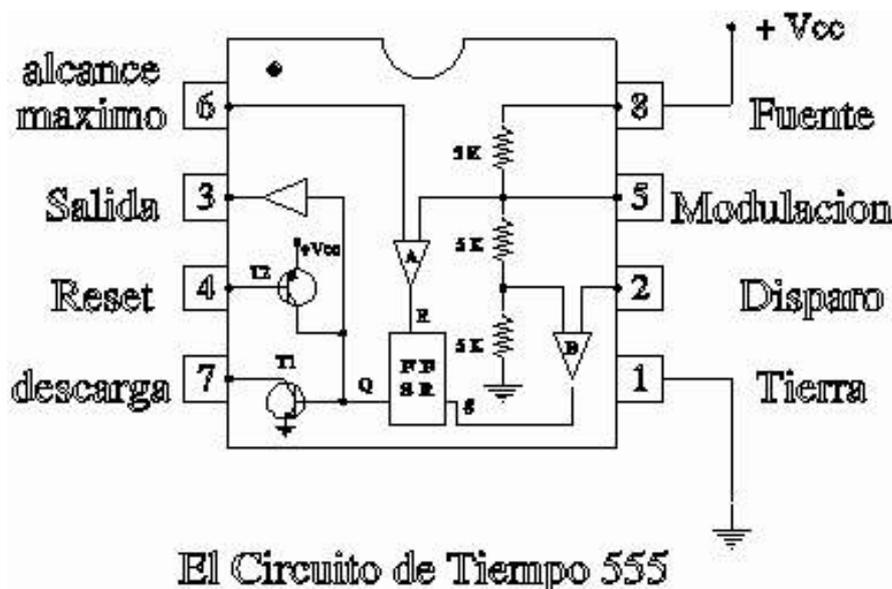
(\*\*): Simbología del ANSI o Instituto Nacional de Estándares Americanos (American National Standards Institute) y el IEEE o Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers).

## EL TIMER 555

Este excepcional Circuito Integrado muy difundido en nuestros días nació hace 30 años y continúa utilizándose actualmente, veamos una muy breve reseña histórica de este C.I. Jack Kilby ingeniero de Texas Instrument en el año de 1950 se las ingenió para darle vida al primer circuito integrado, una compuerta lógica, desde entonces y hasta nuestros tiempos han aparecido innumerables circuitos integrados, en Julio de 1972, apareció en la fábrica de circuitos integrados SIGNETICS CORP., un microcircuito de tiempo el NE555V, inventado por el grupo que dirigió el Jefe de Producción en ese tiempo, Gene Hanateck, este integrado se puede aplicar a diversas aplicaciones, tales como:

- **Control de sistemas secuenciales,**
- **Generación de tiempos de retraso,**
- **Divisor de frecuencias,**
- **Modulación por anchura de pulsos,**
- **Repetición de pulsos,**
- **Generación de pulsos controlados por tensión, etc.**

Además de ser tan versátil contiene una precisión aceptable para la mayoría de los circuitos que requieren controlar el tiempo, su funcionamiento depende únicamente de los componentes pasivos externos que se le interconectan al microcircuito 555.



### Descripción del Timer 555:

Se alimenta de una fuente externa conectada entre sus terminales (8) positiva y (1) tierra; el valor de la fuente de alimentación se extiende desde 4.5 Volts hasta 18.0 Volts de corriente continua, la misma fuente exterior se conecta a un circuito pasivo RC exterior, que proporciona por medio de la descarga de su Capacitor una señal de voltaje que está en función del tiempo, esta señal de tensión es de  $1/3$  de  $V_{cc}$  y se compara contra el voltaje aplicado externamente sobre la terminal (2) que es la entrada de un comparador como se puede apreciar en la gráfica anterior.

La terminal (6) se ofrece como la entrada de otro comparador, en la cual se compara a  $2/3$  de la  $V_{cc}$  contra la amplitud de señal externa que le sirve de disparo.

La terminal (5) se dispone para producir (PAM) modulación por anchura de pulsos, la descarga del condensador exterior se hace por medio de la terminal (7), se descarga cuando el transistor (NPN) T1, se



## Manual de introducción al uso de microcontroladores PIC ARDUINO

encuentra en saturación, se puede descargar prematuramente el Capacitor por medio de la polarización del transistor (PNP) T2.

Se dispone de la base de T2 en la terminal (4) del circuito integrado 555, si no se desea descargar antes de que se termine el periodo, esta terminal debe conectarse directamente a Vcc, con esto se logra mantener cortado al transistor T2 de otro modo se puede poner a cero la salida involuntariamente, aun cuando no se desee.

La salida está provista en la terminal (3) del microcircuito y es además la salida de un amplificador de corriente (buffer), este hecho le da más versatilidad al circuito de tiempo 555, ya que la corriente máxima que se puede obtener cuando la terminal (3) sea conecta directamente al nivel de tierra es de 200 mA.

La salida del comparador "A" y la salida del comparador "B" están conectadas al Reset y Set del FF tipo SR respectivamente, la salida del FF-SR actúa como señal de entrada para el amplificador de corriente (Buffer), mientras que en la terminal (6) el nivel de tensión sea más pequeño que el nivel de voltaje contra el que se compara la entrada reset del FF-SR no se activará, por otra parte mientras que el nivel de tensión presente en la terminal 2 sea más grande que el nivel de tensión contra el que se compara la entrada Set del FF-SR no se activará.

### El microcircuito 555 es un circuito de tiempo que tiene las siguientes características:

- **La corriente máxima de salida es de 200 mA cuando la terminal (3) de salida se encuentra conectada directamente a tierra.**
- **Los retardos de tiempo de ascenso y descenso son idénticos y tienen un valor de 100 nseg.**
- **La fuente de alimentación puede tener un rango que va desde 4.5 Volts hasta 16 Volts de CD.**
- **Los valores de las resistencias R1 y R2 conectadas exteriormente van desde 1 ohm hasta 100 kohms para obtener una corrimiento de temperatura de 0.5% a 1% de error en la precisión, el valor máximo a utilizarse en la suma de las dos resistencias es de 20 MΩ.**
- **El valor del Capacitor externo contiene únicamente las limitaciones proporcionadas por su fabricante.**
- **La temperatura máxima que soporta cuando se están soldando sus terminales es de 330 centígrados durante 19 segundos.**
- **La disipación de potencia o transferencia de energía que se pierde en la terminal de salida por medio de calor es de 600 mW.**

El dispositivo 555 es un circuito integrado muy estable cuya función primordial es la de producir pulsos de temporización con una gran precisión y que, además, puede funcionar como oscilador.

### Sus características más destacables son:

- Temporización desde microsegundos hasta horas.
- Modos de funcionamiento:

o Monoestable.

o Astable.

- Aplicaciones:

o Temporizador.

Manual de introducción al uso de microcontroladores PIC ARDUINO o Oscilador.

- o Divisor de frecuencia.
- o Modulador de frecuencia.
- o Generador de señales triangulares.

Las aplicaciones del 555 son tan numerosas que prácticamente no existe un sistema electrónico que no lo utilice de alguna forma. Su versatilidad, bajo costo y sencillez de uso lo hacen imprescindible en muchos casos. Además sobre el se ha escrito mucha literatura y existen cientos de libros, artículos y documentos sobre sus aplicaciones reales y potenciales.

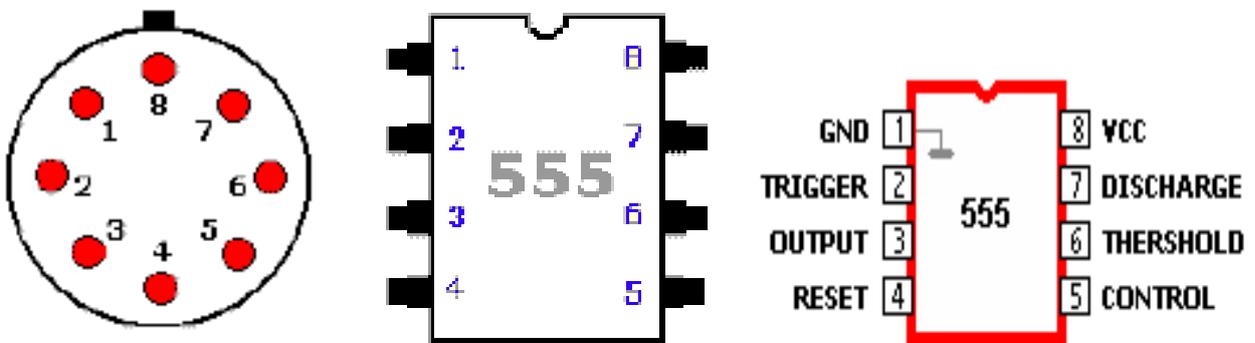
El circuito integrado 555 en su presentación usual de cápsula plástica dispone de 8 pines. Puede estar etiquetado bajo distintos nombres o referencias dependiendo del fabricante (NE555,  $\mu$ A555, LM555, SN72555, XR-555, CA555, HA1755, NC1455, TA7555P, ECG955, etc). También se consigue en otras presentaciones incluyendo cápsulas metálicas para aplicaciones de montaje superficial (SMT).

El 555 convencional consta internamente de 23 transistores, 2 diodos y 12 resistencias de las cuales 3 son de 5 Kohms por eso tiene el nombre de 555. Este circuito tiene una capacidad suficiente para impulsar directamente leds, zumbadores, bobinas de relé, parlantes, piezoeléctricos y otros componentes, además, es directamente compatible con circuitos integrados digitales estándares, que es otra de sus grandes ventajas.

### Configuración de pines

Se puede ver de la figura que independientemente del tipo de encapsulado la numeración de las pines es la misma.

El 556 es un C.I con 2 temporizadores tipo 555 en una sola unidad de 14 pines y el 558 es un C.I. con 4 temporizadores tipo 555 en una sola unidad de 14 pines



### Descripción de las pines del 555

**1 Tierra o masa: ( Ground )** Conexión a tierra del circuito en general.

**2 Disparo: ( Trigger )** Es en esta patilla, donde se establece el inicio del tiempo de retardo, si el 555 es configurado como monostable. Este proceso de disparo ocurre cuando este pin va por debajo del nivel de 1/3 del voltaje de alimentación. Este pulso debe ser de corta duración, pues si se mantiene bajo por mucho tiempo la salida se quedará en alto hasta que la entrada de disparo pase a alto otra vez.

**3 Salida: ( Output )** Aquí veremos el resultado de la operación del temporizador, ya sea que esté conectado como monostable, astable u otro. Cuando la salida es alta, el voltaje será el voltaje de aplicación (Vcc) menos 1.7 Voltios. Esta salida se puede obligar a estar en casi 0 voltios con la ayuda de la patilla # 4 (reset) **4 Reset:** Si se pone a un nivel por debajo de 0.7 Voltios, pone la patilla de salida # 3 a nivel bajo. Si por algún motivo esta patilla no se utiliza hay que conectarla a Vcc para evitar que el 555 se "reseteo"



### Manual de introducción al uso de microcontroladores PIC ARDUINO

**5 Control de voltaje: ( Control )** Cuando el temporizador se utiliza en el modo de controlador de voltaje, el voltaje en esta patilla puede variar casi desde  $V_{cc}$  (en la práctica como  $V_{cc} - 1$  voltio) hasta casi 0 V (aprox. 2 Voltios ). Así es posible modificar los tiempos en que la patilla # 3 está en alto o en bajo independiente del diseño (establecido por las resistencias y condensadores conectados externamente al 555). El voltaje aplicado a la patilla # 5 puede variar entre un 45 y un 90 % de  $V_{cc}$  en la configuración monostable. Cuando se utiliza la configuración astable, el voltaje puede variar desde 1.7 voltios hasta  $V_{cc}$ . Modificando el voltaje en esta patilla en la configuración astable causará la frecuencia original del astable sea modulada en frecuencia (FM). Si esta patilla no se utiliza, se recomienda ponerle un capacitor de 0.01uF para evitar las interferencias

**6 Umbral: ( Threshold)** Es una entrada a un comparador interno que tiene el 555 y se utiliza para poner la salida (Pin # 3) a nivel bajo .

**7 Descarga: ( Discharge )** Utilizado para descargar con efectividad el condensador externo utilizado por el temporizador para su funcionamiento.

**8 V+:** También llamado  $V_{cc}$ , es el pin donde se conecta el voltaje de alimentación que va de 4.5 voltios hasta 16 voltios (máximo). Hay versiones militares de este integrado que llegan hasta 18 Voltios

### Especificaciones generales del 555

$V_{cc}$	5-Voltios	10-Voltios	15-Voltios	Notas
Frecuencia máxima (Astable)	500-kHz a 2-MHz			Varia con el Mfg y el diseño
Nivel de tensión $V_c$ (medio)	3.3-V	6.6-V	10.0-V	Nominal
Error de frecuencia (Astable)	~ 5%	~ 5%	~ 5%	Temperatura 25° C
Error de temporización (Monoestable)	~ 1%	~ 1%	~ 1%	Temperatura 25° C
Máximo valor de $R_a + R_b$	3.4-Meg	6.2-Meg	10-Meg	
Valor mínimo de $R_a$	5-K	5-K	5-K	
Valor mínimo de $R_b$	3-K	3-K	3-K	
Reset VH/VL (pin-4)	0.4/<0.3	0.4/<0.3	0.4/<0.3	
Corriente de salida (pin-3)	~200ma	~200ma	~200ma	

El temporizador 555 se puede conectar para que funcione de diferentes maneras, entre los más importantes están: como multivibrador astable y como multivibrador monoestable

**Multivibrador astable:** Este tipo de funcionamiento se caracteriza por una salida con forma de onda cuadrada (o rectangular) continua de ancho predefinido por el diseñador del circuito. El esquema de conexión es el que se muestra. La señal, de salida tiene un nivel alto por un tiempo  $T_1$  y en un nivel bajo un tiempo  $T_2$ . Los tiempos de duración dependen de los valores de  $R_1$  y  $R_2$ .

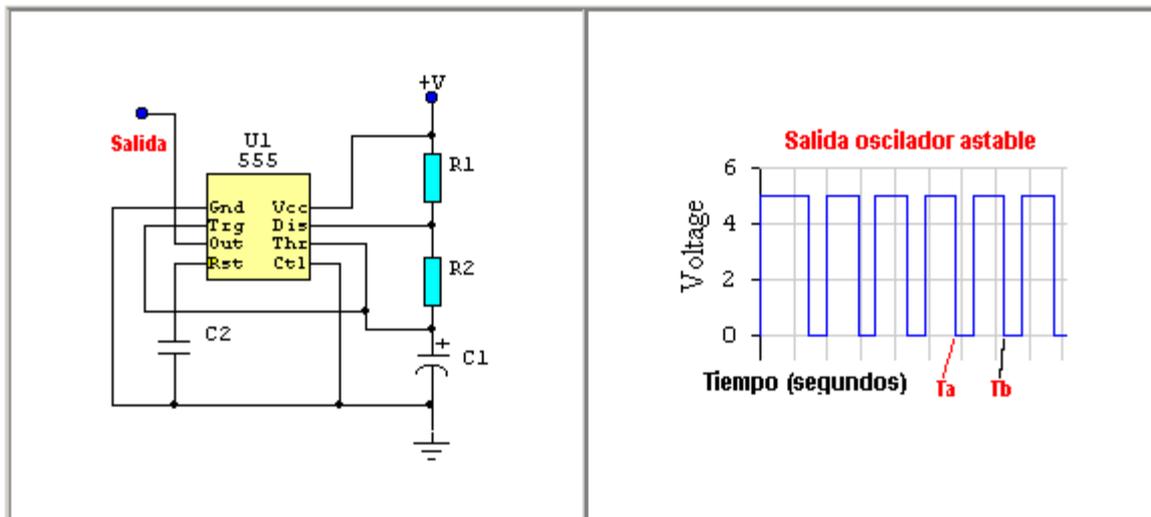
$$T1 = 0.693(R1+R2)C1 \quad (\text{seg}) \quad \text{y} \quad T2 = 0.693 \times R2 \times C1 \quad (\text{seg})$$

La frecuencia con que la señal de salida oscila está dada por la fórmula:

$$f = 1/(0.693 \times C1 \times (R1 + 2 \times R2)) \quad f = 1 / (T1 + T2)$$

y el período es simplemente  $T = 1 / f$

El ciclo de trabajo está dado por  $CT=100 \times R2 / (R1+ 2 \times R2)$



Cuando la señal de disparo está a nivel alto (ej. 5V con Vcc 5V) la salida se mantiene a nivel bajo (0V), que es el estado de reposo.

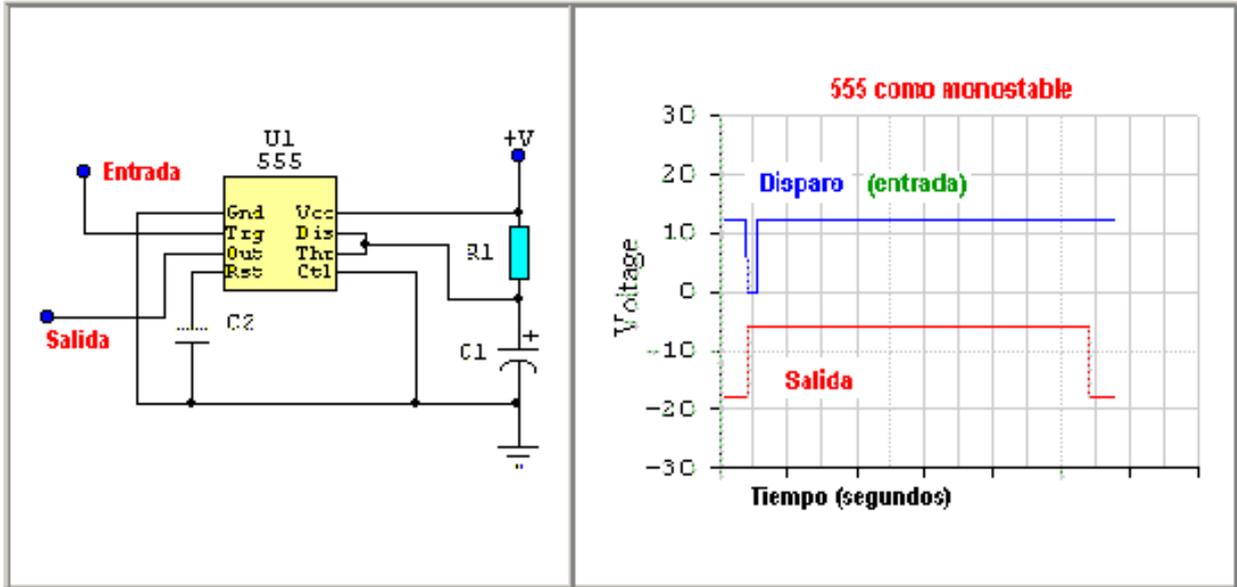
Una vez se produce el flanco descendente de la señal de disparo y se pasa por el valor de disparo, la salida se mantiene a nivel alto (Vcc) hasta transcurrido el tiempo determinado por la ecuación:

$$T = 1.1 \times R1 \times C \quad (\text{seg})$$

Es recomendable, para no tener problemas de sincronización que el flanco de bajada de la señal de disparo sea de una pendiente elevada, pasando lo más rápidamente posible a un nivel bajo (idealmente 0V).

**NOTA:** en el modo monoestable, el disparo debería ser puesto nuevamente a nivel alto antes que termine la temporización.

**Multivibrador Monostable:** En este caso el circuito entrega a su salida un sólo pulso de un ancho establecido por el diseñador (tiempo de duración). El esquema de conexión es el que se muestra. La Fórmula para calcular el tiempo de duración (tiempo que la salida está en nivel alto) es:  $T = 1.1 \times R1 \times C1$  (en segundos). Observa que es necesario que la señal de disparo, sea de nivel bajo y de muy corta duración en el PIN # 2 del C.I. para iniciar la señal de salida.



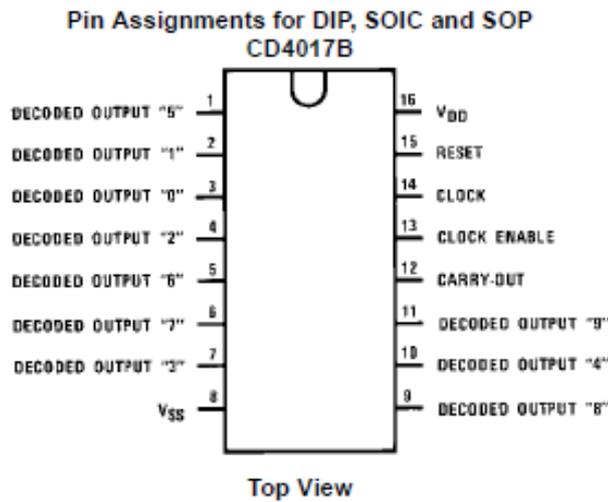
Manual de introducción al uso de microcontroladores PIC ARDUINO

# CD4017

Un El circuito integrado CD4017, es un contador y divisor hasta 10. Técnicamente se le conoce a este tipo de circuito, como contador jhonson de varias etapas (en este caso 5).

El CD4017, es utilizado frecuentemente como secuenciador de luces y divisor de bajas frecuencias. Es muy popular entre los aficionados y estudiantes que se inician en la electrónica.

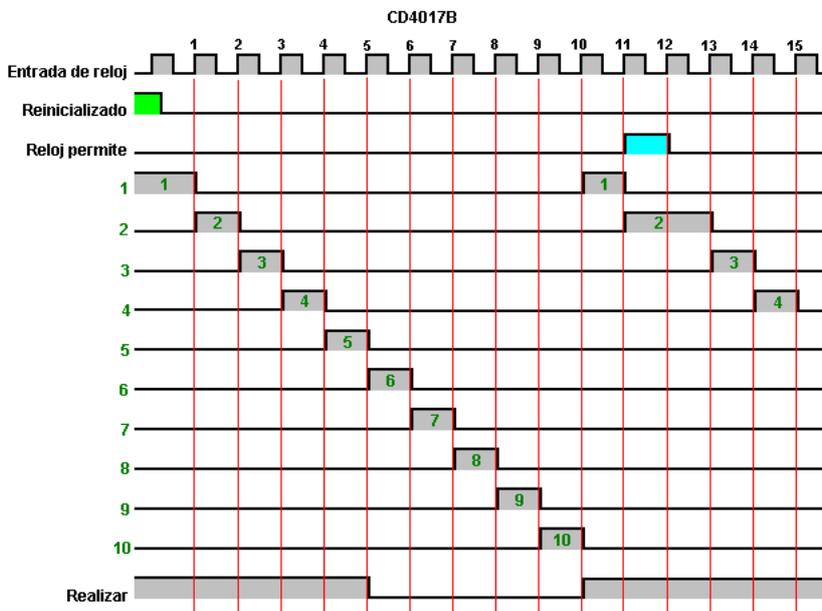
El CD4017, se puede adquirir por poco dinero, su bajo precio lo hace ideal en circuitos de prueba y experimentación. A continuación podemos ver la descripción de sus pines, los cuales son 16 en total.



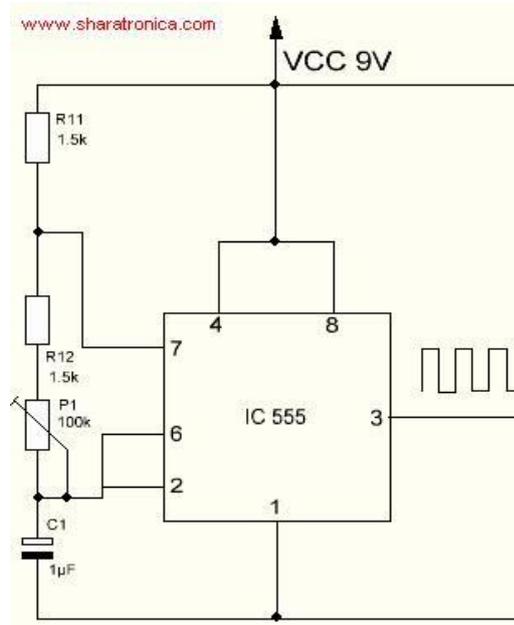
La alimentación del circuito integrado, se hace por medio del pin 16 y debido a su tecnología CMOS, el CD4017 puede ser alimentado desde 3 a unos 15 voltios, de corriente continua. Sus salidas (10 en total, de 0 a 9) comienzan desde el pin 3 (Q0) y no se encuentran de forma secuencial, terminando en el pin 11 (Q9), como se puede ver en la imagen.

Para que el CD4017 pueda realizar sus acciones, este debe recibir un tren de pulsos por el pin 14. Cada vez que reciba un flanco positivo, el CD4017 avanzara una posición en su contador y al llegar al final, activara el pin 14 (carry out) en donde podremos conectar otro circuito integrado CD4017 para ampliar el conteo hasta 20 pasos. A esta configuración se le suele llamar "conexión en cascada".

Grafica de entrada d reloj vs estado de los pines



Como fuente de pulsos digitales, se suele utilizar el popular circuito integrado 555, el cual también es de muy bajo costo. Para variar la velocidad de los pulsos digitales, se utiliza un potenciómetro de ajuste, tal cual se ve en la siguiente imagen.



## Velocidad de operación del CD4017

CD4017 a 5v = 2 Mhz

CD4017 a 15v = 6 Mhz

74HC4017 a 5v = 25 Mhz (Versión TTL de alta velocidad).

## Otros datos del CD4017

**Reset** (pin 15): En funcionamiento normal, este pin debe ser llevado a tierra (GND). Si queremos limitar el conteo a menos de 10 salidas, entonces se debe llevar el ultimo pin de la cuenta hacia el pin de Reset. Por ejemplo, queremos un conteo de 3 pasos, entonces se debe conectar, la salida 4 al pin de Reset, para iniciar nuevamente el conteo desde la primera posición.

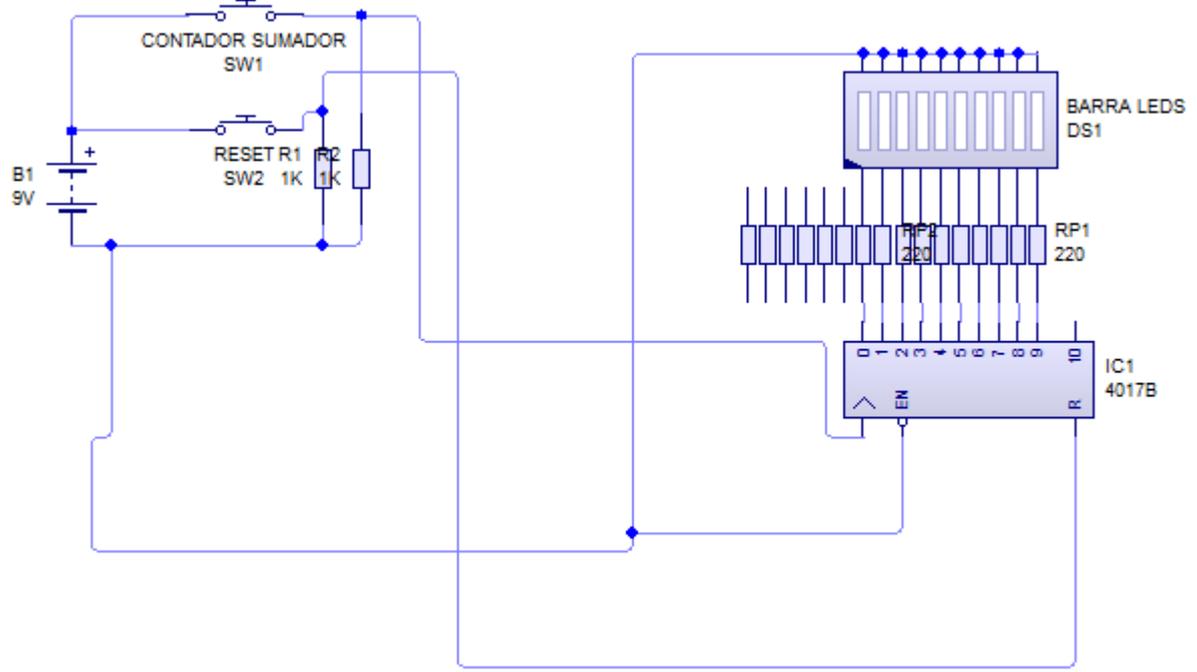
**Clock Inhibit** (pin 13): Normalmente debe estar conectado a tierra. Pero si queremos detener el conteo, este pin deberá recibir un estado alto, cuando pase nuevamente a nivel bajo, el conteo sigue desde donde se había detenido.

**Carry out** (pin 12): Cuando el conteo termina, este pin pasa a estado alto momentáneamente. Se puede utilizar para ampliar las salidas del contador, utilizando uno o más circuitos integrados.

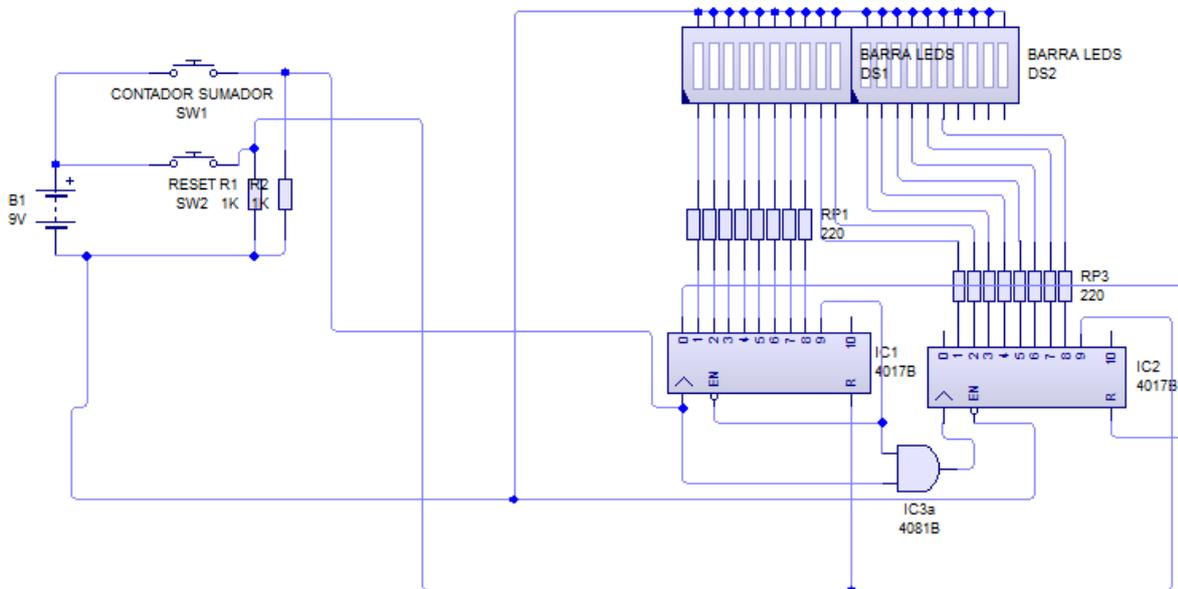
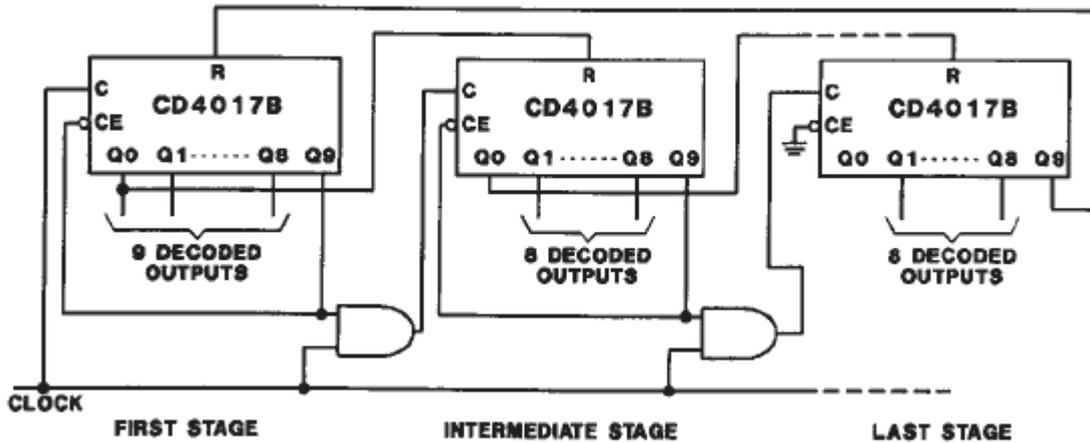
A las salidas del circuito integrado, se suele conectar diodos LED para monitorear, el estado de estas. Entre las aplicaciones posibles del CD4017, podemos encontrar al circuito integrado suministrando corriente (modo source) o drenando corriente (modo sink). En ambas aplicaciones, el circuito se comporta de manera eficiente, siendo el modo source, el más utilizado.

Finalmente mostramos los 2 circuitos más comunes para conectar el circuito integrado CD4017.

CD 4017 en modo, secuenciador de punto luminoso



CD4017 en modo, Secuenciador de agujero.  
Queda para diseño suyo.



### Actividad integradora:

Combina un contador binario con un decodificador de 7 segmentos

- 1) Basándose en lo aprendido anteriormente monta y explica los circuitos vistos.
- 2) ¿Qué función cumple la compuerta?
- 3) Basándose en lo aprendido anteriormente monta y explica tres cd4017 en cascada.



## CONTADORES DIGITALES

Un contador es un circuito en el que sus salidas siguen una secuencia fija que cuando acaba vuelve a empezar, o circuitos que reciben sus datos en forma serial ordenados en distintos intervalos de tiempo.

Los pulsos de entrada pueden ser pulsos de reloj u originarse en una fuente externa y pueden ocurrir a intervalos de tiempo fijos o aleatorios.

El número de salidas limita el máximo número que se puede contar.

Son circuitos digitales lógicos secuenciales de salida binaria o cuenta binaria, característica de temporización y de memoria, por lo cual están constituidos a base de [flip-flops](#).

## CARACTERISTICAS IMPORTANTES

1. Un número máximo de cuentas (módulo del contador)
2. Cuenta ascendente o descendente.
3. Operación síncrona o asíncrona.
4. Autónomos o de auto detención.

## UTILIDAD

Se utilizan para contar eventos.

Ejemplos:

1. número de pulsos de reloj.
2. medir frecuencias.
3. Se utilizan como divisores de frecuencia y para almacenar datos. Ejemplo: en un reloj digital.
4. Se utilizan para direccionamiento secuencial y algunos circuitos aritméticos.

## CONTADORES DE RIZADO.

Son dispositivos contadores que tienen conectados los [flip-flops](#) en forma asíncrona, es decir, que no tienen conectadas las entradas de reloj (CLK) en paralelo, sino que tiene que esperar que el primer [flip-flop](#), al activarse por el pulso conmute generando una salida, la cual active o coloque en modo de conmutación al siguiente flip-flop, el cual con el siguiente pulso conmuta activando al siguiente y así sucesivamente. El modo de conmutación en los [flip-flop](#) se consigue colocando las entradas J y K en ALTA (1 lógico).

El módulo de un contador está determinado por la cuenta máxima a la que es diseñado, es decir, si el contador es diseñado para que cuente de 0 a 15 su módulo es el 16 (contador módulo 16) y simplificado se denomina contador mod-16, si es diseñado para contar de 0 a 9 será un contador módulo 10 o mod-10, etc.

## CONTADOR DE RIZADO MODULO- 16.

Este contador se encuentra constituido por [flip-flop JK](#) en modo de conmutación al mantener presente en las entradas J y K un 1 lógico y conectados entre si de forma asíncrona, es decir, que la salida del flip-flop 1 (FF1) está conectada de forma directa a la entrada de reloj del siguiente flip-flop 2 (FF2). Los indicadores de salida dan una señal binaria, donde el indicador A (QA) es el LSB (Bit Menos Significativo), el indicador D (QD) es el MSB (Bit Más Significativo).

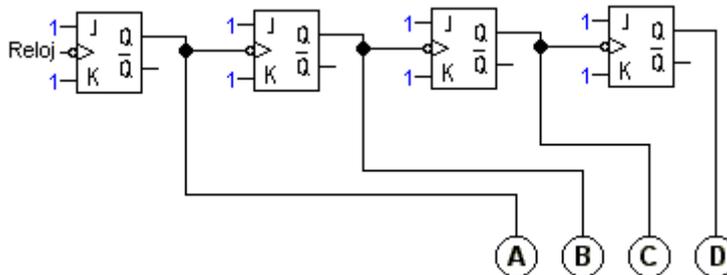


Figura 1: Contador de rizado mod-16

El circulito en la entrada de reloj (CLK) de los flip-flops, nos indica que trabajan o conmutan con lógica negativa, es decir, que se activan en la transición de ALTA a BAJA (flanco posterior) del pulso de reloj y la salida del FF1 (QA) va del nivel BAJO al ALTO dando como resultado la cuenta binaria 0001. En el pulso 2, en la transición del nivel ALTO a BAJO, FF1 se desactiva pasando su salida del nivel ALTO a BAJO, activando el FF2, conmutando la salida del nivel BAJO a ALTO generando la cuenta 0010, en el pulso 3 del reloj se activa FF1 generando la salida 0011, porque FF2 se encuentra en mantenimiento teniendo su salida (QB) activada, en el siguiente pulso se incrementa la cuenta a 0100, según se observa en el diagrama de tiempo de la figura 2.

Cuenta decimal	Cuenta binaria	Cuenta decimal	Cuenta binaria
0	0 0 0 0	8	1 0 0 0
1	0 0 0 1	9	1 0 0 1
2	0 0 1 0	10	1 0 1 0
3	0 0 1 1	11	1 0 1 1
4	0 1 0 0	12	1 1 0 0
5	0 1 0 1	13	1 1 0 1
6	0 1 1 0	14	1 1 1 0
7	0 1 1 1	15	1 1 1 1

Tabla 1: Secuencia de un contador mod-16

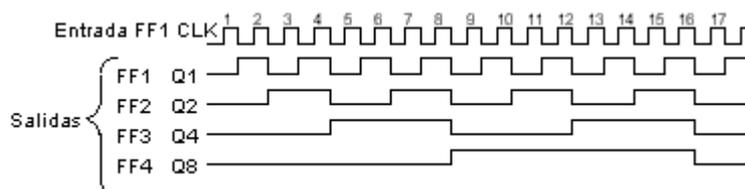


Figura 2: Diagrama de tiempos de un contador mod-16

## CONTADORES PARALELOS

Con este tipo de contadores se elimina o se atenúa el retardo que se presenta en los contadores asíncronos, donde se tiene que esperar que un [flip-flop](#) active al otro. Este efecto se consigue conectando el reloj directamente a las entradas de reloj (CLK) de los [flip-flops](#), es decir, conectando los pulsos de reloj en paralelo (síncronamente) y las salidas de los [flip-flops](#) a las entradas J y K de los mismos.

### CONTADOR PARALELO DE 3 BITS MOD-8.

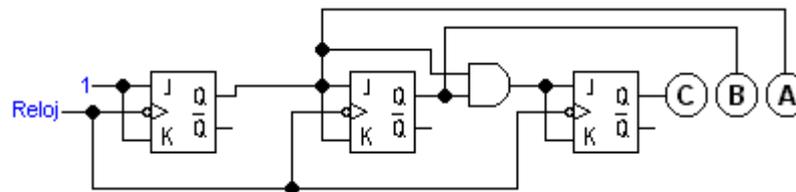


Figura 3: Contador paralelo mod-8

También está construido a base de [flip-flops JK](#), los cuales tienen conectadas sus entradas de reloj en paralelo y sus salidas QA, va conectada a las entradas J y K del siguiente flip-flop (FF2) y así sucesivamente por lo que estaría en modo de mantenimiento hasta que la salida del FF1 les de un 1 lógico lo que los colocaría en modo de conmutación a FF2, al estar las entradas del reloj en paralelo, la transición del primer pulso del nivel ALTO a BAJO, FF1 se activa mientras que FF2 se coloca en modo de conmutación y FF3 sigue en mantenimiento generando la cuenta 001. En el segundo pulso FF1 se desactiva y FF2 conmuta generando la salida 010, si en el tercer pulso estuviera la salida FF2 conectada directamente a las entradas J y K del FF3 se generaría la cuenta máxima 111, por que el FF2 se encuentra en estado de mantenimiento en este caso activado por el pulso anterior, teniendo en modo de conmutación a FF3 el cual, junto con FF1 se activaría en el pulso 3. Para evitar este inconveniente se conecta la salida del FF1 y del FF2 a las entradas de una puerta [AND](#) y las salidas de la puerta [AND](#) a las entradas J y K de FF3, colocandolo en modo de conmutación solamente cuando FF1 y FF2 estén activados, es decir, en el pulso 3. Generando en el pulso 4 de reloj que se desactiven FF1 y FF2 y se active FF3 generando la cuenta 100 y en los siguientes pulsos se generarán. El resto de cuenta como se muestra en el diagrama de tiempo de la figura 4.

Cuenta Binaria			Cuenta decimal
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Tabla 2: Secuencia de un contador mod-8

Manual de introducción al uso de microcontroladores PIC ARDUINO

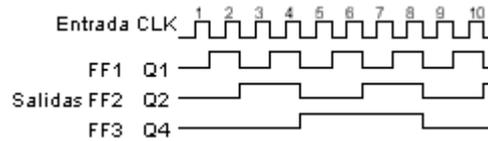


Figura 4: Diagrama de tiempos de un contador mod-8

**OTROS CONTADORES.**

Estos contadores no llegan a su cuenta máxima, porque se interrumpe su cuenta según el diseño o la necesidad que se tenga, por ejemplo, un contador MOD-6 o MOD-10.

**CONTADOR DE RIZADO MOD-6.**

Para conseguir este tipo de contador de bits, se utiliza una entrada de reset o borrado la cual se activa inmediatamente después de la cuenta más alta que se necesite, en este caso en la cuenta 110, colocando los [flip-flops](#) en 0 lógico. En la figura 5 se muestra el esquema de un contador mod-6.

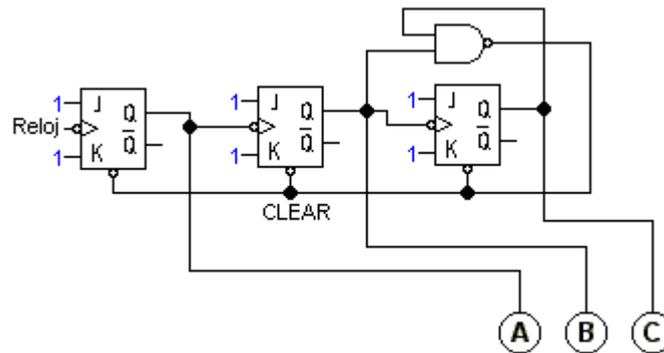


Figura 5: Esquema lógico de un contador de rizado mod-6

Este trabajo de activar las entradas de reset de cada [flip-flop](#) lo realiza una puerta [NAND](#) la cual da un 0 lógico a las entradas de reset. Al recibir en las entradas de la [NAND](#) los 1 lógicos de las salidas del FF2 y del FF3 colocándolos en 0 lógico todos los [flip-flops](#) y así el contador comienza de nuevo a contar desde 000 hasta 101 o inversamente si es de cuenta descendente.

C	B	A	Cuenta decimal
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	RESET
1	1	1	

Tabla 3: Secuencia de un contador mod-6

Manual de introducción al uso de microcontroladores PIC ARDUINO

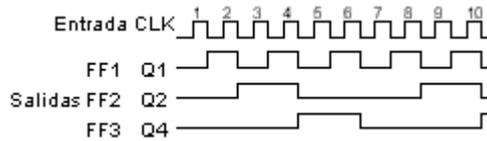


Figura 6: Diagrama de tiempos del contador mod-6

El retardo que es causado en el pulso 6 cuando va del nivel ALTO a BAJO hasta que FF2 y FF3 son puestos a 0 en el punto B del diagrama de tiempo, se le denomina tiempo de propagación y éste depende del retardo de propagación del [flip-flop](#) y de la [puerta](#) que se esté utilizando, este retardo de propagación en la familia [TTL](#) es del orden de unos 30ns (nanosegundos). En las otras familias son mayores.

### CONTADOR DECADA (CUENTA DECENAS)

Es uno de los más utilizados, esta construido a base de 4 [flip-flops JK](#) y una puerta [NAND](#) la cual pone en 0 los [flip-flops](#) al llegar la cuenta máxima (1010). Como se sabe un contador de 4 bits, llega a una cuenta máxima binaria de 1111 que equivale a 16 en decimal, por lo que la puerta [NAND](#) debe activarse inmediatamente después de la cuenta 1001 o 9 en decimal para que el contador sea mod-10.

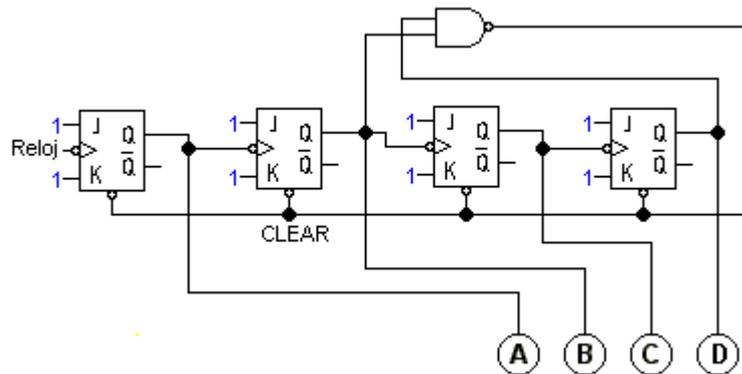


Figura 7: Diagrama lógico de un contador década rizado

Entonces, como la cuenta inmediatamente después de 1001 es 1010, entonces se conectan las entradas de la puerta [NAND](#) a las salidas de FF2 (QB) y FF4 (QD) que al mandar los unos a las entradas de la [NAND](#), la activan enviando un pulso a las entradas de reset (borrado o CLR) de los [flip-flops](#) colocándolos en cero y reiniciando la cuenta.

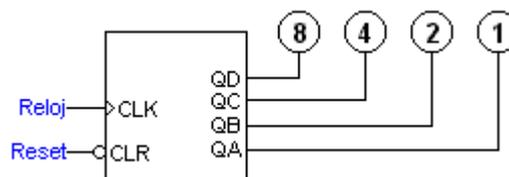


Figura 8: Símbolo lógico simplificado de un contador década

## CONTADORES DESCENDENTES

Son los contadores en los cuales su cuenta va en sentido inverso a la normal, es decir, de 16 a 0 o en binario de 1111 a 0000. (si es de mod-16)

### CONTADOR DE RIZADO DESCENDENTE DE 3 BITS

Esta diseñado similarmente a los demás contadores, con la diferencia que este trae en los [flip-flops](#) una salida negada (1), la cual da el pulso contrario a la salida normal (Q), es decir, cuando Q es positivo, 1 es negativo. Esta salida 1 es la que va a ir conectada a la entrada de reloj (CLR) de los otros flip-flops, de resto todas las conexiones son iguales como se muestra en la figura 9.

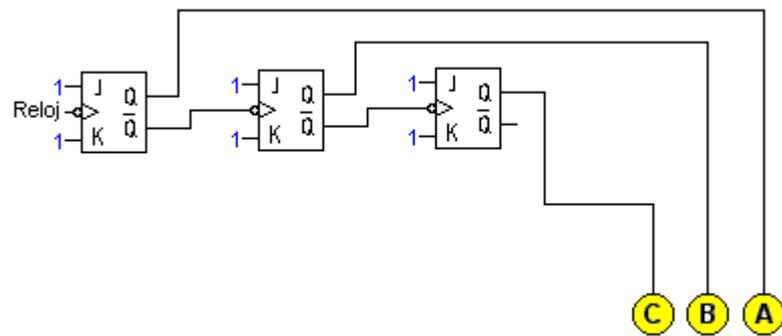


Figura 9: Contador de rizado descendente de 3 bits

El funcionamiento es el siguiente: al tener los 3 [flip-flops](#) sus entradas J y K en estado de conmutación (ambas entradas en ALTO) y sus salidas Q activadas o en estado de SET en los [flip-flops](#), al llegar el primer pulso en la transición de ALTO a BAJO, el FF1 conmuta, con lo cual Q va del nivel ALTO a BAJO y 1 va del nivel BAJO al ALTO y la cuenta pasa de 111 a 110 (de 7 a 6 en decimal), en el pulso 2 en la transición de ALTO a BAJO, FF1 conmuta con lo cual la salida Q va del nivel BAJO al ALTO y la salida 1 va del nivel BAJO al ALTO y se genera la cuenta 101 (5 en decimal) y así hasta llegar a la cuenta máxima, que en este caso es 0000 como se muestra en el diagrama de tiempo,

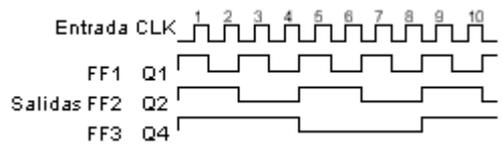


Figura 10: Diagrama de tiempos de un contador descendente de 3 bits

### CONTADORES CON CI TTL

Son circuitos integrados donde vienen incluidos los flip-flops conectados según el tipo de contador y las puertas. Estos contadores se pueden llamar de propósito general. El CI [74192](#) es un contador reversible BCD síncrono TTL, es decir, módulo-10. Tiene doble entrada de reloj, una para cuenta ascendente y una para cuenta descendente que conmutan en la transición del nivel BAJO al nivel ALTO del pulso. La entrada de borrado síncrono se activa en nivel ALTO colocándo las salidas en nivel BAJO (0000) y se inicializa en cualquier número que se cargue en las entradas de datos en forma binaria y se transfieren asincrónamente a la salida BCD (A=QA, B=QB, C=QC, D=QD). La salida de arrastre se utiliza para conectar en cascada serie varios contadores.

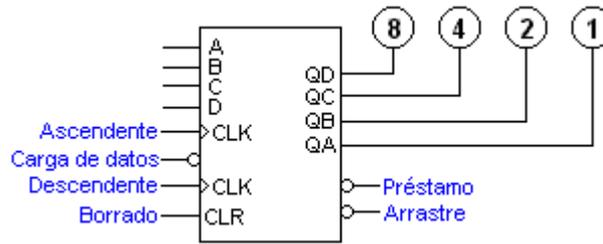


Figura 11: Símbolo del contador 74192

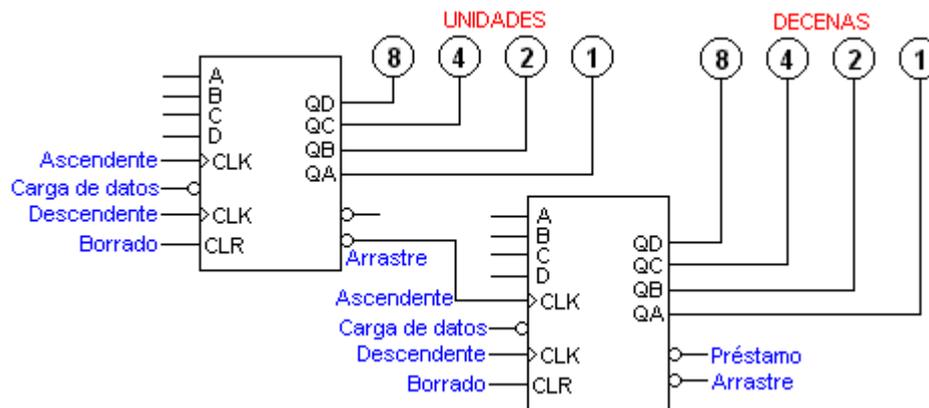


Figura 12: Conexión en cascada de dos contadores de 0 a 99

### CONTADOR BINARIO DE 4 BITS TTL 7493.

El contador [7493](#) utilizan 4 [flip-flops JK](#) en modo de conmutación, con entradas de reloj  $\text{CP0}$  y  $\text{CP1}$  en donde  $\text{CP1}$  es la entrada de reloj del segundo [flip-flop](#) por lo que para formar un contador de 4 bits mod-16 hay que conectar la salida del primer flip-flop de manera externa (puente) con la entrada  $\text{CP1}$ , quedando  $\text{CP0}$  como la entrada de reloj del contador. También tiene dos entradas de reset ( $\text{MR1}$  y  $\text{MR2}$ ) las cuales no se deben dejar desconectadas (flotando) porque, como estas se activan en ALTA, al estar flotando toman un nivel ALTO lo que mantendría en reset al contador.

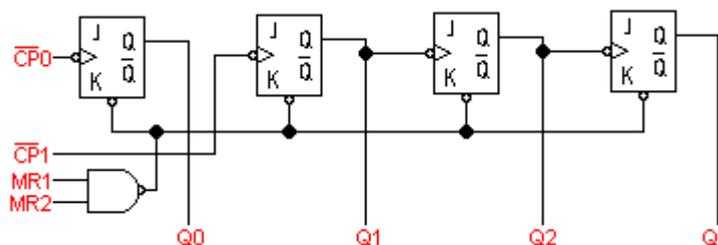


Figura 13: Contador 7493

### CONTADORES CON CI CMOS.

#### CONTADOR CMOS 74HC393

El CI [74HC393](#) es un doble contador binario de 4 bits. Esta construido a base del [flip-flop T](#). Las entradas de reloj ( $1\text{CP}$  y  $2\text{CP}$ ) son activadas por flanco posterior, o sea, en la transición de ALTO a BAJO del pulso de reloj. Las entradas de reset ( $1\text{MR}$  y  $2\text{MR}$ ) del maestro en el contador se activan en nivel ALTO, las salidas se etiquetan desde  $\text{Q0}$  a  $\text{Q3}$ , siendo  $\text{Q0}$  el LSB y,  $\text{Q3}$  el MSB del numero binario de 4 bits. Requiere una fuente de alimentación de 5V DC y viene en un CI DIP de 14 patillas.

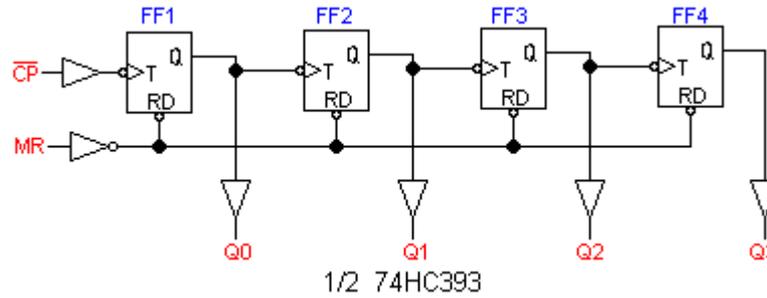


Figura 14: Diagrama lógico del contador CMOS 74HC393

### CONTADOR CMOS CI 74HC193

El CI [74HC193](#) es un contador reversible síncrono de 4 bits preinicializable como lo muestra la hoja de datos.

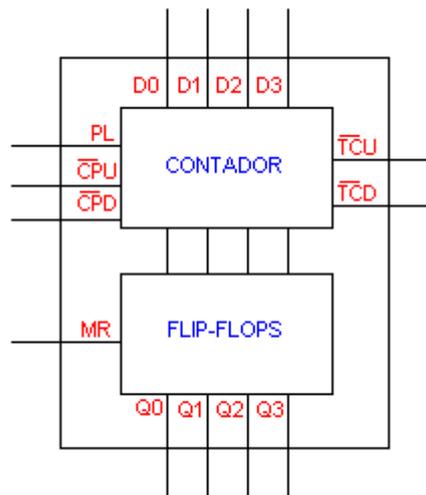


Figura 16: Contador CMOS 74HC193

Tiene 2 entradas de reloj (CPU y CPD), que se activan en la transición del nivel BAJO al ALTO del pulso de reloj, la entrada CPU es para la cuenta ascendente (UP) y la entrada CPD es para la cuenta descendente (D), por lo que dependiendo si el contador que se necesite se conecta al nivel alto o +5V. Los modos de operación del contador CMOS [74HC193](#) se muestran en la tabla de verdad 5. El modo de reset borra asincrónicamente las salidas (Q0 a Q3) al binario 0000 activándose en ALTO el cual puede ser un pulso de corta duración. Las entradas de carga de datos en paralelo (D0 a D3) se utilizan para programar un número en binario desde donde se quiere que empiece a contar de nuevo al activar la entrada de carga en paralelo (P) con un nivel BAJO y los datos son transferidos asincrónicamente a las salidas (Q0 a Q3). Las salidas de arrastre T<sub>C5</sub> y T<sub>C $\bar{D}$</sub>  generan un pulso negativo, para la conexión en cascada de contadores, ya sea en forma ascendente o en forma descendente la cuenta de estos. El contador [74HC193](#) viene en un DIP de 16 patillas y opera con una tensión de alimentación de +5V DC.

## DIVISION DE FRECUENCIA: EL RELOJ DIGITAL.

En un contador digital de salida binaria el retraso que se forma al activarse cada [flip-flop](#) a determinado pulso de reloj, en realidad es una división de frecuencia, por ejemplo, en un contador de 4 bits la salida QA divide la frecuencia en 2 porque necesita un pulso para activarse y otro para desactivarse, la salida QB divide en 4 la frecuencia del reloj de entrada porque tiene que esperar que pasen los 2 pulsos en la salida QA para poder activarse y otros 2 pulsos para desactivarse, la salida QC es una salida que divide por 8 y la salida QD divide por 16.

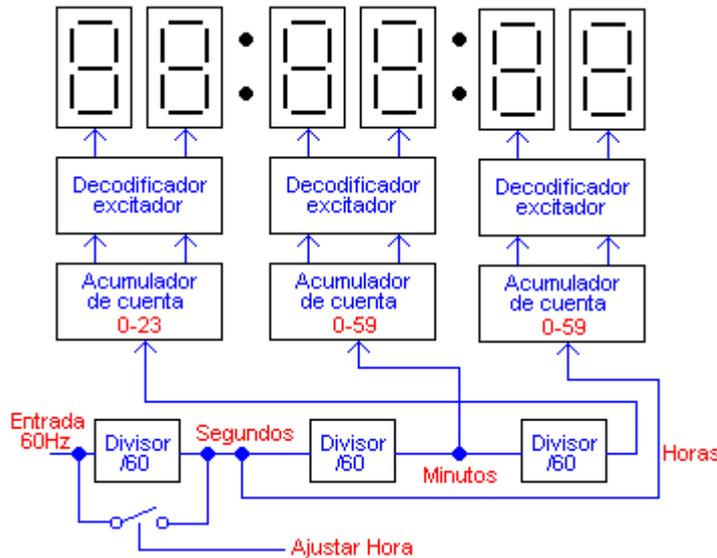


Figura 20: Diagrama de bloques de un reloj digital.

El reloj digital utiliza los contadores como divisores de frecuencia y acumuladores de cuenta. La función del contador como acumulador de cuentas es contar los pulsos de entrada y sirve como memoria temporalmente mientras muestra la hora actual que es decodificada y pasada a los visualizadores de hora. Los contadores como divisores de frecuencia tienen en su entrada una onda cuadrada de 60 Hz, el bloque divisor por 60, es construido por un contador divisor por 6, conectado a un contador divisor por 10.



Figura 21: Contador divisor por 60

El contador divisor por 6 transforma los 60 Hz en 10 Hz y el contador divisor por 10 transforma los 10Hz en 1Hz o 1 pulso por segundo. El contador divisor por 10 es construido con un CI [7493](#) por lo que la primera conexión que se debe hacer es un puente entre Q0 y ÇP1 para convertirlo en un contador de 4 bits, en segundo lugar el CI debe convertirse en un contador de década (mod-10) como se explicó anteriormente, conectando Q3 y Q1 a las 2 entradas de reset. El contador divisor por 6 es hecho con un CI [7493](#) conectando la entrada de reloj a ÇP1, es decir, que el primer [flip-flop](#) (entrada ÇP0) no se utiliza. Los acumuladores de cuenta de 0 a 59 son 2 contadores en donde uno es un contador mod-10 para acumular las unidades (0 al 9) de los segundos y el otro es un contador mod-6 que recibe el pulso de arrastre del mod-10 para contar las decenas de los segundos. Los decodificadores/excitadores sirven para decodificar la salida BCD al visualizador de 7 segmentos.



## DECODIFICADO BINARIO A DISPLAY 7 SEGMENTO 4511

En la imagen se puede ver el esquema de un 4511 o sus equivalentes en diferentes marcas CD4511, TC4511, MC14511 o HEF451. Este circuito integrado CMOS dispone en su interior un decodificador BCD a display de 7 segmentos con latch/decoder/driver.

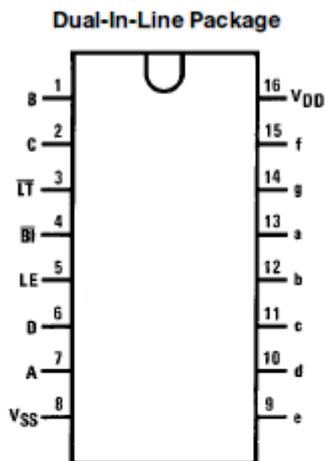
Si queremos utilizar tecnología TTL [7447](#).

El circuito proporciona 25 mA máximo de salida por segmento y los displays a utilizar tienen que ser de cátodo común, si se quiere utilizar los de ánodo común será necesario colocar un inversor en las salidas. Hay que recordar que hay que colocar una resistencia limitadora de corriente entre las salidas de los segmentos y el Display, esta resistencia será de unos 220 ohmios si trabajamos con una tensión VDD de 5V.

Las patillas de alimentación son el pin 16 VDD, que admite un voltaje máximo de 18V y el pin 8 VSS.

El funcionamiento normal pasa por ir activando la señal LE cuando se quiere cambiar el dígito representado (teniendo las líneas BL y LT desactivadas). Cuando LE está a "1" las salidas son las del último dígito que se ha visualizado y cuando esta señal pasa a "0" el integrado toma las entradas D, C, B y A, y genera la salida para activar los segmentos correspondientes. Tras la captura del dígito a representar se vuelve a desactivar (1) la señal LE.

### Connection Diagram

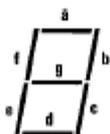


TL/F/5991-1

**Top View**

**Order Number CD4511B**

**Segment Identification**



TL/F/5991-3

En circuitos multiplexados que se disponen de varios circuitos 4511 y tienen las líneas de datos (BCD) conectadas en paralelo; activando la señal correspondiente de captura (LE) se determina para cuál de los circuitos es el dato.

En la tabla de la verdad se observa todas las combinaciones posibles para este circuito.

Estas son otras características con las que cuenta este integrado:

- Baja disipación de potencia del circuito lógico.
- Salidas que suministran alta corriente (hasta 25 mA).

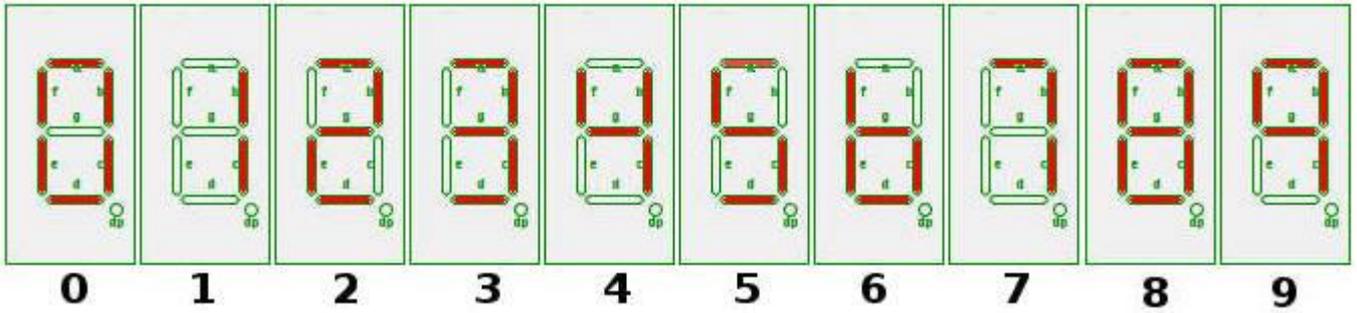


Escuela de Educación Técnico Profesional N° 602  
Taller de electrónica 5°  
Sección Instrumentación y Control



Manual de introducción al uso de microcontroladores PIC ARDUINO

- Almacenamiento de código en cerrojo.
- Entrada de borrado.
- Borrado del indicador en todas las condiciones ilegales de entrada.
- Facilidad de tiempo compartido (multiplexado).



Diferentes combinaciones que se pueden obtener en el display

Tabla de la verdad del 4511

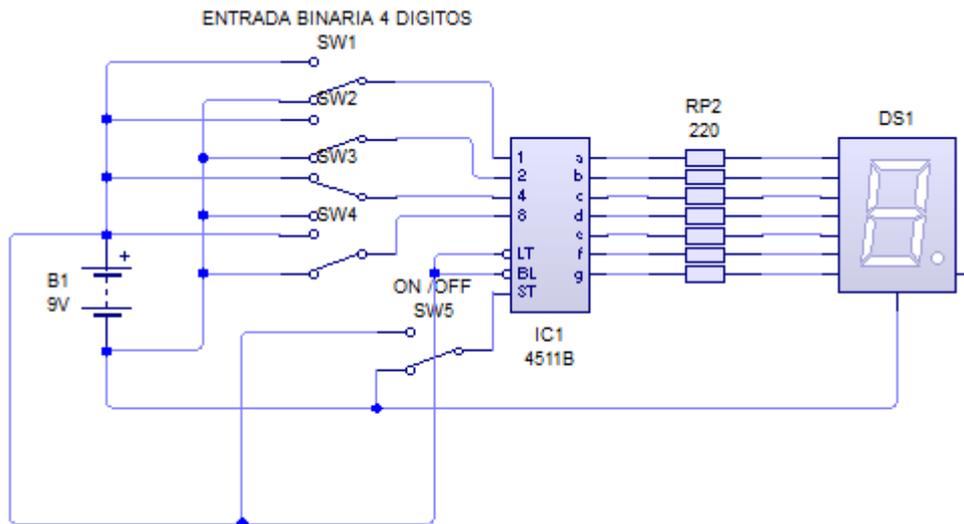
Entradas				Salidas	
LE	$\overline{BL}$	$\overline{LT}$	D C B A	a b c d e f g	Display
0	1	1	0 0 0 0	1 1 1 1 1 1 0	0
0	1	1	0 0 0 1	0 1 1 0 0 0 0	1
0	1	1	0 0 1 0	1 1 0 1 1 0 1	2
0	1	1	0 0 1 1	1 1 1 1 0 0 1	3
0	1	1	0 1 0 0	0 1 1 0 0 1 1	4
0	1	1	0 1 0 1	1 0 1 1 0 1 1	5
0	1	1	0 1 1 0	0 0 1 1 1 1 1	6
0	1	1	0 1 1 1	1 1 1 0 0 0 0	7
0	1	1	1 0 0 0	1 1 1 1 1 1 1	8
0	1	1	1 0 0 1	1 1 1 0 0 1 1	9
0	1	1	1 0 1 0	0 0 0 0 0 0 0	apagado
0	1	1	1 0 1 1	0 0 0 0 0 0 0	apagado
0	1	1	1 1 0 0	0 0 0 0 0 0 0	apagado
0	1	1	1 1 0 1	0 0 0 0 0 0 0	apagado
0	1	1	1 1 1 0	0 0 0 0 0 0 0	apagado
0	1	1	1 1 1 1	0 0 0 0 0 0 0	apagado
1	1	1	X X X X	*	*
X	X	0	X X X X	1 1 1 1 1 1 1	8
X	0	1	X X X X	0 0 0 0 0 0 0	apagado

### Manual de introducción al uso de microcontroladores PIC ARDUINO

"X" = Estado indiferente, 0 ó 1.

"\*" = Depende del código aplicado en las entradas A, B, C Y D durante la transición de la señal LE de 0 a 1.

Actividad: Realiza el siguiente circuito en circuit wizard y luego móntalo en protoboard



## Contador binario CD4518 o CD4520

LOS INTEGRADOS CD4020, CD4040 Y CD4060

Estos integrados, al igual que el CD4024, son todos contadores binarios y están constituidos por una cadena de flip-flops que componen sus etapas. Debido a la limitación que impone la cantidad de terminales de su encapsulado, no todos tienen acceso a las salidas de todos los biestables, por lo general cuantas mas etapas tienen más salidas quedarán aisladas, pero en contrapartida se logran factores de división mas elevados.

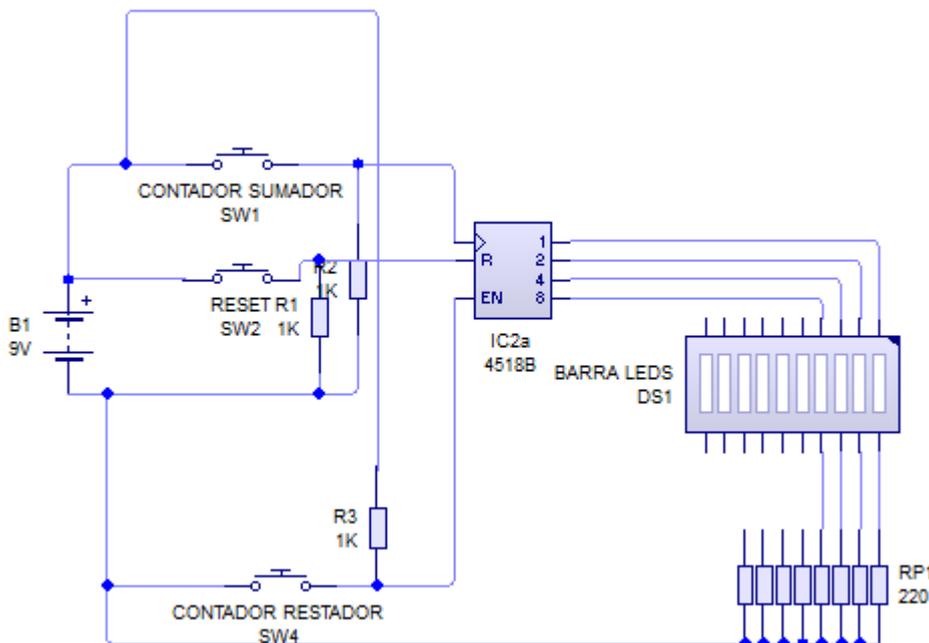
El CD4020 , CD4018 es un contador binario de 14 etapas, tiene acceso a 12 de ellas y falta en las etapas segunda y 3a; el máximo factor de división obtenible, como ya hemos dicho está dado por la cantidad de etapas, por lo tanto es de  $2^{14} = 16.384$ .

TRUTH TABLE

CLOCK	ENABLE	RESET	ACTION
	1	0	Increment Counter
0		0	Increment Counter
	X	0	No Change
X		0	No Change
	0	0	No Change
1		0	No Change
X	X	1	Q1 thru Q4 = 0

X = Don't Care    1 ≡ High State    0 ≡ Low State

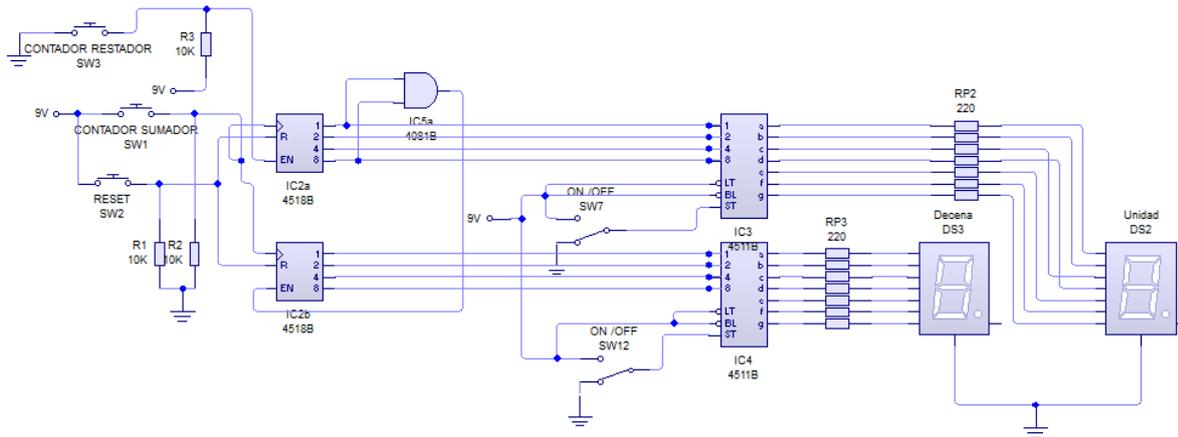
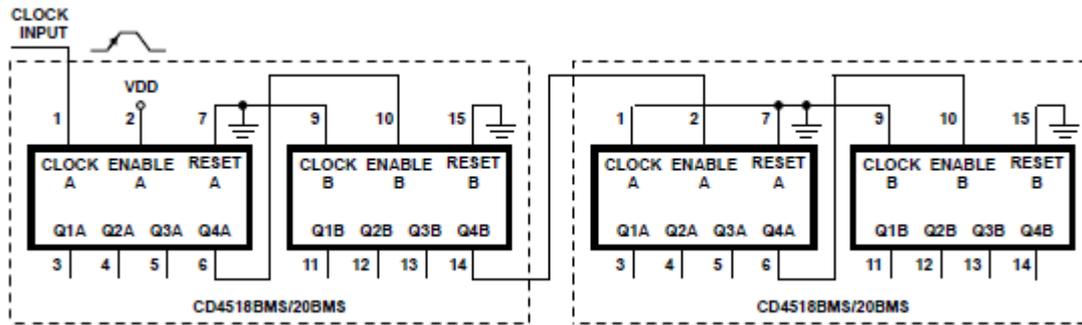
Actividad: Realiza el siguiente circuito en circuit wizard y luego móntalo en protoboard



Observar la configuración Pull-Up comprendida entre R3 y SW4 (Contador restador) y la Pull\_Down comprendida entre R2 y SW1 (Contador sumador).



Manual de introducción al uso de microcontroladores PIC ARDUINO  
**CONEXIÓN EN CASCADA**



¿Para qué está esa compuerta?

**Actividad integradora:**

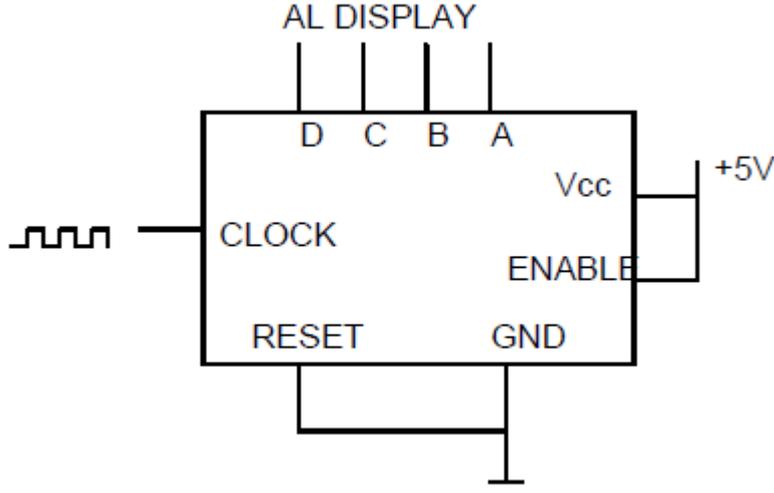
**Combina un contador binario con un decodificador de 7 segmentos**

- 4) *Basándose en el siguiente esquema agregar al circuito la opción de restar.*

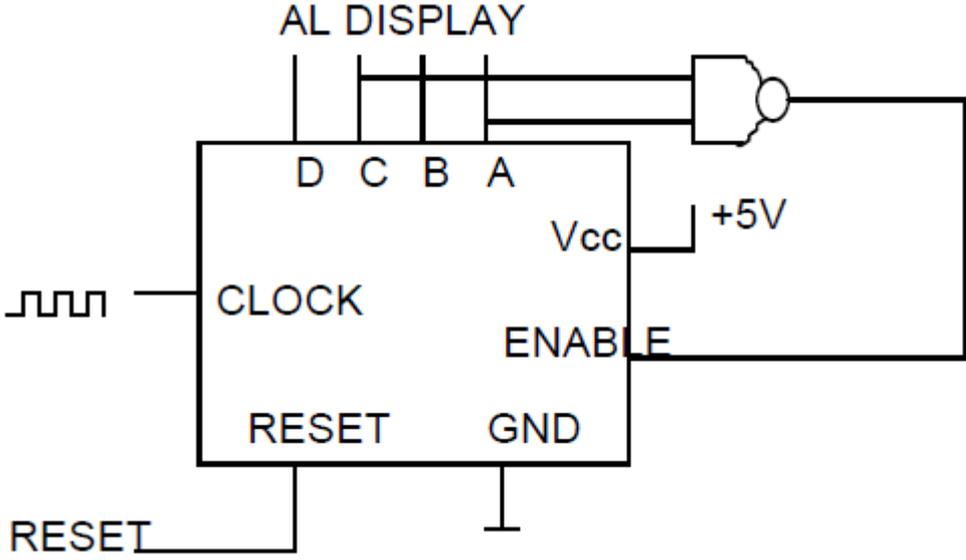
*Doble CONTADOR BCD 4518*

Dos contadores de decadas, independientes en un solo chip. Cuando enable está en 1 y reset en 0, el contador avanza una cuenta con cada pulso de clock.

Ver hoja de datos

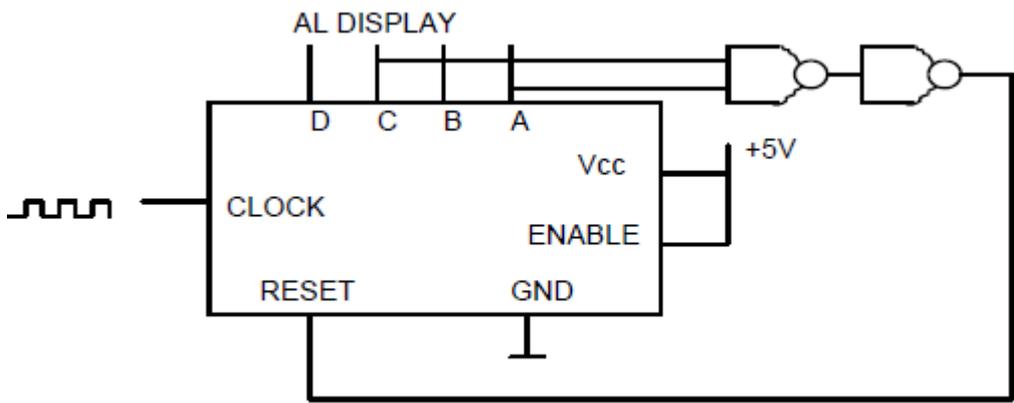


5) Basándose en el siguiente esquema agregar al circuito la opción de que cuente hasta 3 y pare. Cuenta hasta 5 y para



6) Basándose en el siguiente esquema agregar al circuito la opción de que cuente hasta 7 y se reinicie. La compuerta deshabilita al contador cuando este alcanza el estado 0101 (decimal 5)

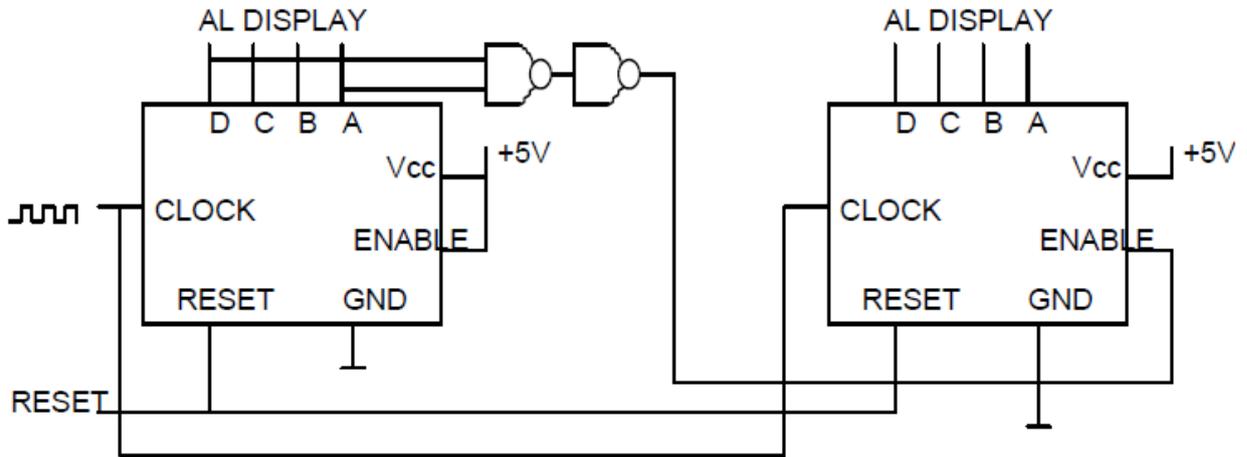
Cuenta hasta 4 y reinicia la cuenta.



7) Basándose en el siguiente esquema agregar al circuito la opción de que cuente hasta 60 y se reinicie.

Contador de 2 dígitos ( 0 - 99 )

Las dos compuertas habilitan al segundo contador mientras el primer contador está en 1001 ( decimal 9 ).



- 1) Basándose en lo aprendido diseñar un contador que me muestre en un display desde el 0 al 60 y que sea accionado por un tren de pulsos enviado por un 555.
- 2) Basándose en lo aprendido diseñar un contador que me muestre en un display desde el 0 al 999 y que sea accionado por pulsador para sumar y otro para restar.
- 3) Basándose en lo aprendido diseñar un contador que me muestre en un display los **segundos minutos y horas**; que sea accionado por un tren de pulsos enviado por un 555.
- 4) Calibrar el circuito anterior para que sea un reloj de tiempo real.



## El mundo de los PIC

Un PICmicro es un circuito integrado programable. Microchip, su fabricante dice: **Programmable Integrated Circuit.**

Programmable quiere decir que se puede planificar la manera cómo va a funcionar, que se puede adaptar a nuestras necesidades. En otras palabras que el integrado es capaz de modificar su comportamiento en función de una serie de instrucciones que es posible comunicarle.

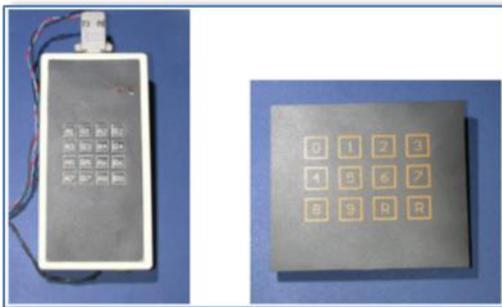
PIC Y PICMICRO para todos los fines prácticos describen el mismo microcontrolador ya que en 1997 Microchip registró el nombre PICmicro para su línea de microcontroladores.

Las aplicaciones de los PIC son realmente muy numerosas, veamos solamente algunas como para entusiasmar a nuestros amigos lectores:

### Control de pantallas alfanuméricas LCD

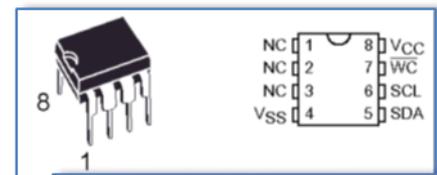
Los microcontroladores son especialmente útiles para controlar con muchísima facilidad los displays de cristal líquido LCD.

### Control de teclados



En muchas aplicaciones se requieren teclados especiales que se adapten exactamente a ciertas necesidades. Los PICmicro se pueden utilizar para realizar secuencias de rastreo y así saber que tecla se ha oprimido.

### Control de memorias EEPROM seriales



### Control de temperatura

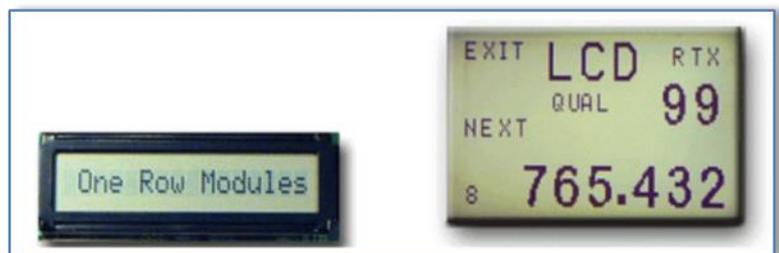
Gracias a que los PIC incluyen convertidores Analógico / Digitales el control de variables como Temperatura, Presión Flujo puede realizarse con circuitos sumamente simples. Con unas cuantas resistencias y un buen sensor de temperatura se puede hacer un Termostato o un controlador de Flujo.



### Robots

Casi se puede asegurar que no hay Robot que no incluya un PIC en alguna de sus múltiples funciones.

### Control de



Son igualmente importantes en control de servomecanismos, reconocimiento de voz, tareas secuenciales, etc.

### Control para motores

Los PIC también se pueden utilizar para controlar motores de diferentes tipos: de pasos, de corriente directa, servos, etc.



## OTRAS APLICACIONES



Escuela de Educación Técnico Profesional N° 602  
Taller de electrónica 5º  
Sección Instrumentación y Control



## Manual de introducción al uso de microcontroladores PIC ARDUINO

No puedo imaginar un límite para las aplicaciones de los PIC, lo mismo los podemos encontrar en Televisión, video, como interfaces de PC, máquinas herramientas, aparatos de audio, controles remoto, sistemas de alarmas, seguridad en general. Cada estudiante tiene sus propios proyectos y grandes ideas para desarrollar. La esencia de los PIC está precisamente en su facilidad de programación, de eso se trata de convertir el integrado en el circuito que soñamos.

### Las familias de los PICmicro

Para seleccionar de entre los 140 productos que ofrece la línea de microcontroladores de Microchip algunos de los criterios que deberíamos de tomar en cuenta son:

Los microcontroladores que ofrecen una mayor flexibilidad para la programación y una enorme facilidad para el ajuste final de los proyectos son los que cuentan con memoria eléctricamente borrable Flash. Para el principiante son también la mejor elección ya que permite grabar y regrabar el integrado muchas veces. A continuación se enlistan los PICmicro mejor conocidos por su memoria Flash.

PIC16F629	PIC16F83, PIC16F84, PIC16F84A
PIC16F675	PIC16F871, PIC16F872, PIC16F873, PIC16F873A
PIC16F627	PIC16F874, PIC16F874A, PIC16F876, PIC16F876A
PIC16F628	PIC16F877, PIC16F877A
PIC16F72, PIC16F73, PIC16F74	PIC18F242, PIC18F248

El PIC con memoria Flash más popular es el PIC16F84. Cuenta con una Memoria de Programación serial de 1024 instrucciones y 64 localidades de memoria RAM. La memoria de programación es eléctricamente borrable.

## Conocimientos previos básicos

Muchos estudiantes deseosos de aprender a probar este tipo de micros se preguntan a cerca de los conocimientos básicos o previos que se deben de tener para iniciarse en el mundo de los PIC. Veamos la lista siguiente.

### Sistema de numeración Hex / Decimal / Binaria

Es deseable que el estudiante este familiarizado sobre todo con la notación hexadecimal, con los registros de 8 bits, los byte, los bits, etc.

### Operación de circuitos de CD

Conocimientos generales sobre circuitos de CD: resistencias, capacitores, bobinas, potenciómetros y otros componentes electrónicos son muy importantes para el buen aprendizaje de los PICmicro.

### Operación de circuitos TTL / CMOS

Las compuertas lógicas, básculas, contadores y otros integrados son básicos en el mundo de los PIC.

### Nociones de programación

Algunos conocimientos de programación pueden ayudar mucho al estudiante, por ejemplo el conocimiento de Basic, C o FLOWCODE, PICC COMPILER, etc.

### Conocimientos de PC básicos (Software y Hardware)

No se requiere ser un experto en el manejo de Computadoras Personales, pero si es importante que el alumno se desenvuelva con soltura en estos temas.



Escuela de Educación Técnico Profesional N° 602  
Taller de electrónica 5º  
Sección Instrumentación y Control



Manual de introducción al uso de microcontroladores PIC ARDUINO

Existe una gran variedad de documentos relacionados con los PICmicro. Casi toda la información se puede bajar de la red en formato PDF.

En especial [www.Microchip.com](http://www.Microchip.com) es el lugar del fabricante. Desde ahí se pueden descargar una infinidad de documentos, en especial las **Notas de Aplicación**, que son documentos que describen aspectos específicos de aplicaciones generales. Son ejemplos de los cuales todos podemos aprender mucho. En estas notas de aplicación se tocan aspectos de software y también de hardware. Es una enorme colección de problemas resueltos, de tal manera que nunca tendremos que empezar desde cero al abordar algún proyecto. Cada nota de aplicación incluye una descripción de la aplicación. La presentación del diagrama esquemático y el software.

# Conexión de entradas salidas digitales de propósito general de un pic

## Introducción:

La tarea más básica que podemos hacer con un microprocesador es sustituir a un circuito combi nacional o secuencial. Podemos encender una luz para cierta combinación de entradas, conectar una bomba cuando el nivel de un depósito sea menor de cierto umbral y pararla cuando supere otro umbral, activar una alarma cuando se active algún sensor y la alarma esta armada, etc.

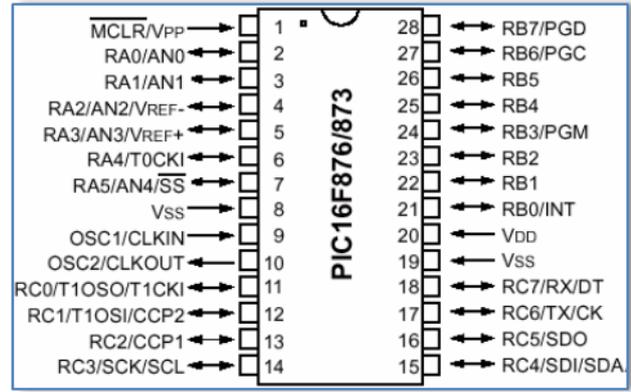
Para poder hacer estas cosas, el microprocesador necesitara tener entradas y salidas digitales, igual que las de cualquier puerta lógica. El microprocesador dsPIC16f873a tiene 28 patillas que pueden ser, a voluntad nuestra, entradas o salidas, de forma que podemos hacer, por ejemplo, un circuito secuencial con 10 funciones y 15 entradas. De todos modos, el tiempo de respuesta de esas funciones es mucho mayor que el de las funciones realizadas con puertas lógicas, aunque en la mayoría de los casos ese tiempo no es demasiado importante; que la alarma suene 1 milisegundo después de abrir la puerta no parece grave.

Ahora bien, estas entradas y salidas digitales tienen niveles pequeños, típicamente de 0 a 3,3 V o de 0 a 5 V, por lo que necesitaremos unos circuitos que harán de interface entre estos niveles (0 a 3,3 V) y los niveles con los que estemos trabajando, 220 VAC, 24 VDC o cualquier otra tensión.

## Entradas/salidas digitales en el dsPIC16f873a

A partir de la configuración de pines vamos a ver los puertos de que dispone este microcontrolador.

En resumen tenemos 28 pines de entrada y salida. Cada una de ellas puede ser entrada o salida independientemente de las otras, aunque algunas tienen alguna peculiaridad. La configuración de pines puede parecer absurda. En el diagrama de pines del micro se observa que la mayoría de pines tienen varios nombres, cada nombre se refiere a una de sus funciones.



## OSCILADOR POR HARDWARE PARA MICROCONTROLADOR

Es un circuito indispensable para el funcionamiento del microcontrolador y el cual además, define la velocidad a la cual va a trabajar. Para hacer funcionar nuestro diseño podemos elegir entre las siguientes cuatro opciones:

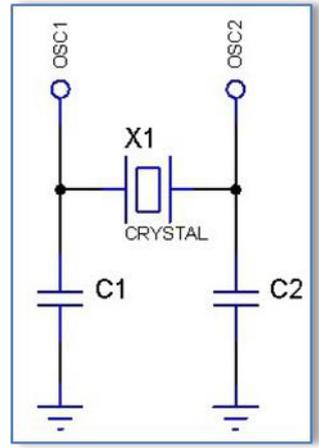
- Oscilador LP: Oscilador de bajo consumo (Low Power).
- Oscilador XT: Cristal / Resonador.
- Oscilador HS: Oscilador de alta velocidad (High Speed).
- Oscilador RC: Resistencia / Condensador.

**Los Modos LP, XT y HS** utilizan un oscilador externo como una fuente de reloj cuya frecuencia está determinada por un cristal de cuarzo o por resonadores cerámicos conectados a los pines OSC1 y OSC2.

**Modo LP** – (Baja potencia) se utiliza sólo para cristal de cuarzo de baja frecuencia. Este modo está destinado para trabajar con cristales de 32.768 KHz. Es fácil de reconocerlos por sus dimensiones pequeñas y una forma cilíndrica. Al utilizar este modo el consumo de corriente será menor que en los demás modos.

**Modo XT** se utiliza para cristales de cuarzo de frecuencias intermedias hasta 8 MHz. El consumo de corriente es media en comparación con los demás modos.

**Modo HS** – (Alta velocidad) se utiliza para cristales de reloj de frecuencia más alta



Manual de introducción al uso de microcontroladores PIC ARDUINO de 8 MHz. Al utilizar este modo el consumo de corriente será mayor que en los demás modos.

**En los modos de oscilador LP, XT y HS** el cristal debe ser conectado como se muestra en la siguiente figura.

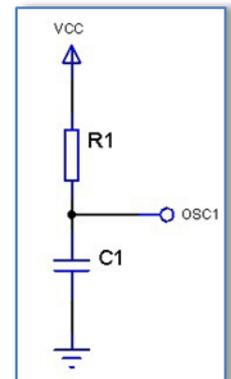
Los valores de los condensadores cerámicos vienen dados según la tabla que se muestra a continuación:

Modo	Frecuencia	OSC1	OSC2
LP	32 KHz	68 – 100 pF	68 – 100 pF
	200 KHz	15 – 33 pF	15 – 33 pF
XT	2 MHz	15 – 33 pF	15 – 33 pF
	4 MHz	15 – 33 pF	15 – 33 pF
HS	4 MHz	15 – 33 pF	15 – 33 pF
	10 MHz	15 – 33 pF	5 – 33 pF

**El Modo RC.** Cuando el oscilador externo se configura a funcionar en modo RC, el pin OSC1 debe estar conectado al circuito RC como se muestra en la figura siguiente. La señal de frecuencia del oscilador RC dividida por 4 está disponible en el pin OSC2. Esta señal se puede utilizar para la calibración, sincronización o para otros propósitos, pero también se puede ocupar el pin OSC2 como entrada o salida.

La frecuencia de este oscilador se calcula por medio de la siguiente fórmula:

$$f = \frac{1}{T} = \frac{1}{R_1 C_1}$$



Dónde:

f es la frecuencia en Hertz (Hz).

T es el periodo en Segundos (s).

R es la resistencia eléctrica en Ohms ( $\Omega$ ).

C es la capacitancia del condensador en faradios (F)

## Programación de los puertos de entrada/salida paralelos

En general, los puertos de entrada/salida tienen un funcionamiento similar. Para cada puerto existen tres registros, el TRISX, el PORTX y el LATX. La X indica cualquiera de los puertos que tiene el microprocesador, A, B, C, D, E o F.

**TRISX** Este registro sirve para indicar si un pin va a ser salida o no. Un 0 en un bit de este registro indica que el pin correspondiente es una salida. Un "1" indica que el pin es entrada. Siendo más preciso no es exactamente así, los pines siempre funcionan como entrada, y cuando ponemos el bit TRIS correspondiente a 1 indicamos que ponemos la salida en alta impedancia, y por lo tanto no funciona como salida, pero cuando ponemos TRIS a 0 la salida impone su nivel, pero al mismo tiempo siempre funciona como entrada, aunque salvo algún problema en el circuito (por ejemplo una salida cortocircuitada a masa), deberemos leer lo que hemos escrito en la salida. Además, podemos también leer la palabra TRISX, conociendo lo que previamente habíamos escrito en TRISX.

**PORTX** Con este registro nos comunicamos con el exterior. Al leer el registro leemos un 0 si la tensión en la entrada está a nivel bajo y un 1 si la tensión está a nivel alto, al escribir el registro hacemos que fuera aparezcan 0 V. o 5 V (3,3V.). Dependiendo de si hemos escrito un 0 o un 1. El nivel alto y el bajo no son valores fijos, sino que hay un margen de valores posibles. (\*)

**LATX** Este es un registro que no existía en otros micros, y podemos realizar programas sin utilizarlo. Este registro es el LATCH de salida. Una escritura en PORTX y en LATX hacen exactamente lo mismo, escriben en el LATCH y si está configurado como salida, el valor aparece en el pin de salida. En la lectura es donde se diferencian. En los pines que solo son entradas no coincidirá LATX y PORTX. En las que además son salidas, normalmente leeremos lo mismo de las dos formas, pero si fuera hay un cortocircuito y en PORTX o LATX hemos escrito un 1, al leer LATX obtendremos el 1, pero al leer PORTX leeremos un 0, ya que es la tensión que realmente hay en el pin. Al leer LATX leemos el último valor escrito por medio de PORTX o LATX y al leer PORTB leemos directamente el valor de tensión del pin (1 o 0). (\*)

\* Aunque podríamos utilizar indistintamente PORT y LAT para escribir datos, en el puerto B ocurren cosas extrañas. Para evitar ese problema debemos usar siempre LAT para escribir datos y PORT para leer. De este modo todos los puertos funcionan como se espera.

**Ejemplo de programación assembler:**

```

TRISB = 0b1111000011110101;
// Los pines RB15, RB14 y RB13 no existen en este micro. Los configuramos
// Como entradas, pero podemos también configurarlos como salidas
// Los pines RB11, 10, 9, 8, 3 y 1 son entradas y salidas
// Los pines RB12, 7, 6, 5, 4, 2 y 0 son entradas
LATB = 0b1010101000111100;
// A los pines que son entradas o que no existen no les pasa nada
// Los pines salidas sí que serán afectados:
// RB11=1 RB10=0 RB9=1 RB8=0 RB3=1 RB1=0
Entrada = PORTB; // Cada bit de la variable entrada tomara un valor dependiendo
// De la tensión que haya en el pin correspondiente. En los pines que no existen se lee 0
// Los bits que son salidas leerán el valor que se había escrito
// Salvo algún problema en el circuito externo, un cortocircuito,
// Varias salidas en paralelo, etc.
// Los bits que son exclusivamente entradas leerán un 0 o 1 dependiendo de la
// Tensión externa.

```

**Funcionamiento como salida:**

Al escribir un 0 en TRIS, este pasa de D a +Q como 0 y ese 0 habilita el buffer de salida que permite pasar lo que haya en la salida Q de "Data latch" al pin de salida. Si escribimos un 1 en TRIS las escrituras en PORT o en LAT no se verán reflejadas en la salida.

**Funcionamiento como entrada:**

Cuando hacemos una lectura de algún bit (o todos) de PORTX se activan las RD de los bits leídos, pasando el valor de la salida del buffer Schmitt-Trigger al bus. Al leer LAT pasa al bus el valor del Data Latch y al leer TRIS pasa al bus el valor del TRIS Latch.

**Características eléctricas de los pines**

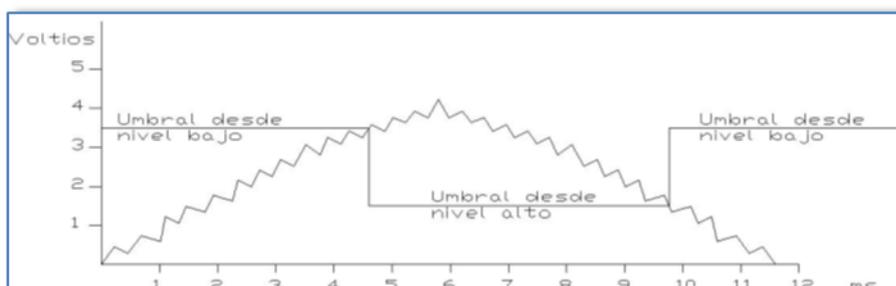
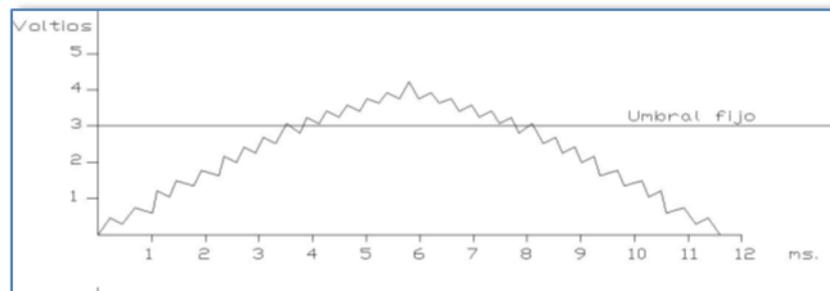
En primer lugar destacar que cada pin puede suministrar o absorber hasta 25 mA, pero entre todos los pines no pueden suministrar o absorber más de 200 mA.

La corriente máxima que puede entrar por los pines VDD es de 250 mA mientras que por el VSS pueden salir hasta 300 mA.

Todo depende de los datos que nos proporcione el fabricante.

**Entradas tipo Schmitt-Trigger.**

En una entrada convencional con un umbral nivel bajo nivel alto en los 3 voltios, al subir de 0 a 1 una señal ruidosa puede hacer el efecto que se ve en la figura. Sobre el ms. 3,5 la señal pasa a nivel alto, pasando



nuevamente a nivel bajo y volviendo a nivel alto. Si leemos esta entrada leeremos 00000101111 en lugar de 00000111111. Al pasar la señal de nivel alto a nivel bajo también hay un momento extraño sobre los 8 ms.

Con las entradas tipo Schmitt-Trigger el umbral de cambio no es constante. Cuando la entrada está a nivel bajo, el umbral está bastante alto, en este ejemplo a 3,5 V. En el momento que la señal supera dicho valor, el umbral baja a 1,5 V, con lo cual el



Manual de introducción al uso de microcontroladores PIC ARDUINO

circuito es mucho más inmune al ruido de la señal. Cuando la señal baja tenemos el efecto contrario obteniendo las mismas ventajas. En este caso leeríamos 0000001111111111000000000. No ha aparecido ningún valor extraño

## Interface de las entradas digitales.

El microprocesador trabaja con niveles de tensión que normalmente están comprendidos entre 0 y 5 VDC, aunque también puede funcionar, y cada vez más, con tensiones más pequeñas. Si queremos leer el estado de un contacto, lo habitual es que de tensiones de 0 o 24 VDC o de 0 o 230 VAC. Por lo tanto necesitaremos unos circuitos que hagan de interface entre los niveles de tensión del microprocesador y los niveles de tensión o corriente que haya en el exterior. Además, cada elemento externo puede tener una alimentación distinta, por lo que conectar varios elementos al microprocesador puede ser peligroso.

A continuación vamos a ver distintos interfaces:

**Lectura de un pulsador o interruptor inversora**

**Lectura de un pulsador o interruptor directa**

**Lectura de una entrada de cualquier nivel de tensión con aislamiento inversora.**

**Lectura de una entrada de cualquier nivel de tensión con aislamiento directa.**

**Monitorización de corriente en un circuito (si o no)**

**Lectura de una entrada con tensión alterna o sin polaridad.**

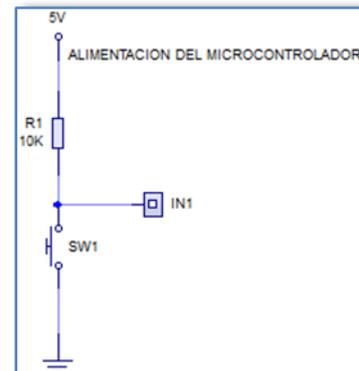
**Lectura de un teclado matricial.**

## Lectura de un pulsador o interruptor en inversora Pull-Up

El interface más sencillo será para leer pulsadores o interruptores situados en la misma placa, o muy cerca, y que alimentaremos a 3,3 o a 5 V con la misma tensión que el microprocesador.

Cuando el interruptor o pulsador está abierto, la patilla IN1 estará prácticamente a 3,3 o 5 voltios, salvo la pequeña caída de tensión que pueda haber en la resistencia provocada por la corriente de fugas. (max 1  $\mu$ A)

Cuando se cierre el interruptor, la patilla IN1 estará a 0. Aunque parezca que el valor de R1 no importa, hay que tener algunas precauciones. Por el pin IN1 del microprocesador puede que haya una pequeña corriente a nivel alto, supongamos que es de 10  $\mu$ A y que R1 valga 10 M. En ese caso en la resistencia caerían 10  $\mu$ A x 1 M $\Omega$  = 10 Voltios.



Esta corriente no puede ser tan grande, valdría menos y la tensión no serían 10 voltios, posiblemente 1 voltio, y por lo tanto el nivel alto no estaría tan claro. Por ello la resistencia debe ser bastante más baja, aunque sin pasarnos, pues al cerrar el interruptor la corriente sería grande. Un valor de 10 K $\Omega$  es el típico.

**A este tipo de entrada se le denomina inversora, porque cuando presionamos el pulsador leemos un cero y cuando no hay leemos un 1.**

## Lectura de un pulsador o interruptor en directa Pull-Down

El interface más sencillo será para leer pulsadores o interruptores situados en la misma placa, o muy cerca, y que alimentaremos a 3,3 o a 5 V con la misma tensión que el microprocesador.

Cuando se abra el interruptor, la patilla IN1 estará a 0. Aunque parezca que el valor de R1 no importa, hay que tener algunas precauciones. Por el pin IN1 del microprocesador puede que haya una pequeña corriente a nivel alto, supongamos que es de 10  $\mu$ A y que R1 valga 10 M. En ese caso en la resistencia caerían 10  $\mu$ A x 1 M $\Omega$  = 10 Voltios.



Cuando el interruptor o pulsador está cerrado, la patilla IN1 estará prácticamente a 3,3 o 5 voltios, salvo la pequeña caída de tensión que pueda haber en la resistencia provocada por la corriente de fugas. (max 1  $\mu$ A)

Esta corriente no puede ser tan grande, valdría menos y la tensión no serían 10 voltios, posiblemente 1 voltio, y por lo tanto el nivel alto no estaría tan claro. Por ello la resistencia debe ser bastante más baja, aunque sin pasarnos, pues al cerrar el interruptor la corriente sería grande. Un valor de

10 K $\Omega$  es el típico.

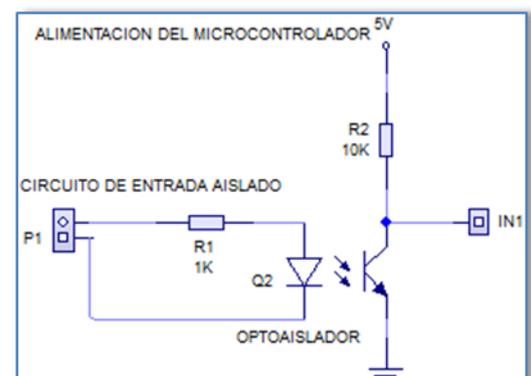
**A este tipo de entrada se le denomina directa, porque cuando presionamos el pulsador leemos un uno 1 y cuando no hay leemos un cero 0.**

## Lectura de una entrada de cualquier nivel de tensión con aislamiento inversora

El circuito anterior no se puede utilizar con tensiones mayores que la de alimentación y la distancia del interruptor al microcontrolador debe ser muy poca. Con el siguiente circuito podemos utilizar tensiones y distancias mayores.

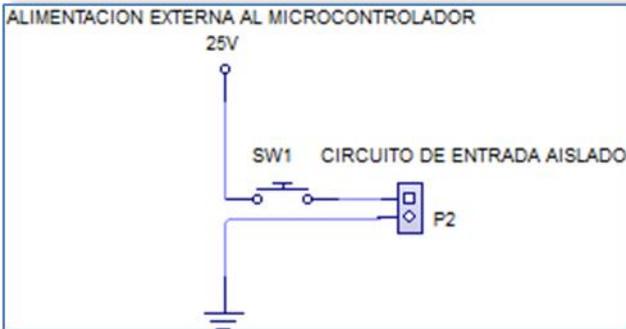
El funcionamiento del circuito es el siguiente:

Cuando no hay tensión entre las patillas 1 y 2 del conector P1 no hay corriente por el diodo del optoaislador.



Manual de introducción al uso de microcontroladores PIC ARDUINO  
Entonces el transistor estará cortado, por lo que en IN1 tendremos un nivel de aproximadamente 5 V (la de alimentación del microcontrolador).

Cuando apliquemos diferencia de tensión entre las patillas 1 y 2 del conector, circulara corriente por el diodo. Esta corriente deberá ser suficiente para que el transistor se sature, por lo que en IN1 habrá una tensión de 0.2 V aproximadamente.



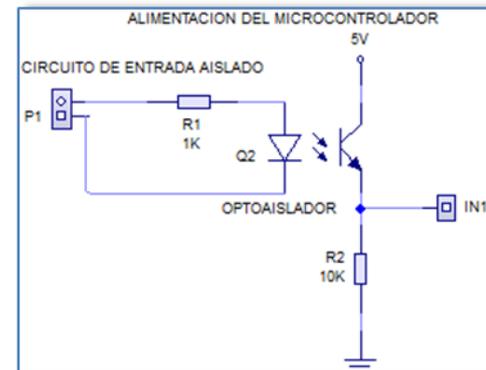
**A este tipo de entrada se le denomina inversora, porque cuando hay tensión leemos un cero 0 y cuando no hay leemos un uno 1.**

En el caso de poner un interruptor el circuito externo sería el siguiente:

Al cerrar el interruptor permitiremos que la diferencia de tensión entre los pines 1 y 2 del conector sea de 25 Vcc. La masa que vemos en el circuito es la de 24 V, y lo normal es que este aislada de la masa que tiene el micro.

Calculo de R1: El valor de la resistencia R1 depende del optoaislador y de la tensión que queramos aplicar a la entrada. Con el valor de corriente que debe circular por el transistor de salida y con ayuda del manual del optoaislador obtendremos la corriente que debe pasar por el LED y la tensión que produce en el LED esa corriente. Esta corriente debe ser suficiente para saturar al transistor y no ser excesivamente grande para no quemar el LED, ni tener un consumo absurdo. Si no tenemos las hojas características podemos suponer en el LED una corriente de 10 mA y una tensión de 1.5 V, que suelen ser adecuados para la mayoría de los optoaisladores.

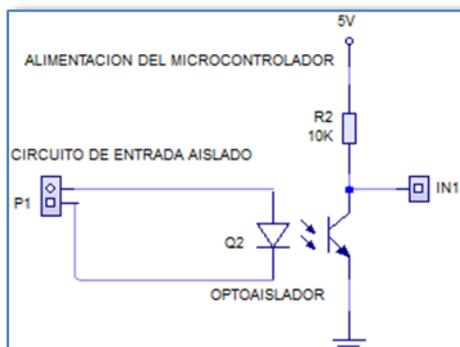
Queda por calcular la resistencia R2. Esta resistencia no debe ser muy baja, pues el transistor no entraría en saturación, ni demasiado alta, pues si a nivel alto la entrada del micro acepta algo de corriente, esta podría provocar una caída de tensión en la resistencia que bajaría la tensión para el nivel alto. Como la relación entrada/salida en la mayoría de opto acopladores está cercana a 1 supondremos ese valor, por lo que la corriente de salida debería ser la misma que la entrada (10 mA) para que estuviese en activa. Si la corriente de salida no pudiese llegar a ese valor el transistor estaría saturado. Para que este saturado, la tensión en la resistencia sería de  $5 - 0,2 = 4,8$  V. Para que pasasen 10 mA  $R = 4,8K/10mA = 0,48 K\Omega = 480\Omega$ , por lo que sí es mayor estará saturado. Un valor adecuado de esa resistencia suele ser 4K7 o 10K. Si la entrada del microcontrolador tiene la función CNX podemos usar la resistencia de pull-up interna.



## Lectura de una entrada de cualquier nivel de tensión con aislamiento directa

Ahora, cuando no hay tensión en la entrada leeremos un 0 y cuando hay tensión leeremos un 1. Los valores de las resistencias se calculan como antes, pero ahora ya no podemos usar las resistencias internas de pul-up.

**A este tipo de entrada se le denomina directa, porque cuando presionamos el pulsador leemos un uno 1 y cuando no hay leemos un cero 0.**



Para poder medir tensiones elevadas se podría simplificar el circuito eliminando el optoaislador y con un divisor de tensión y un zener se podría trabajar, pero el aislamiento soluciona problemas de tensiones no referidas a la misma masa y se evita que la masa y la alimentación del microprocesador se "paseen por el exterior", eliminando problemas de ruido, tanto generado por el micro como el que le puede llegar a al desde el exterior.

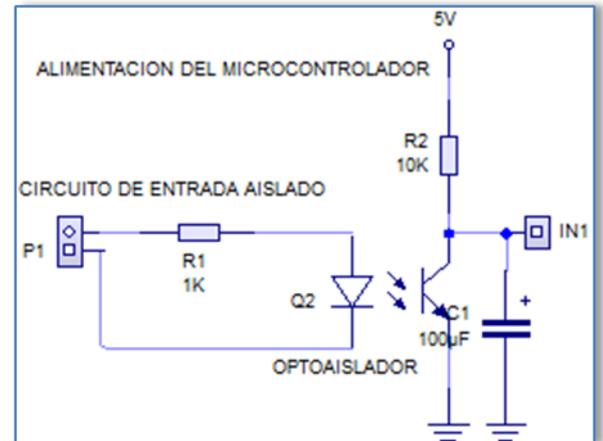
## Monitorización de corriente en un circuito (Si o no).

Si quisiéramos saber si por un cable pasa corriente, (corrientes que soporte el LED, normalmente entre 5 a 50 mA) podemos eliminar la resistencia de entrada, quedando un circuito por el que pasa la corriente a detectar. Tiene el problema de que elimina 1.5 V, pero puede ser útil en muchos casos. Toda la corriente pasa por el opto acoplador por lo que este debe soportar la corriente. Para poder detectar corrientes mayores habría que hacer un divisor de corriente.

## Lectura de una entrada con tensión alterna o sin polaridad aislada.

También se podría utilizar un optoaislador con dos diodos, con lo que podría detectar tensiones o corrientes independientemente del sentido de la tensión o corriente.

Colocando un condensador del valor adecuado podemos hacer que incluso el microprocesador detecte que hay una señal alterna. En los casos anteriores, si ponemos una señal alterna el microprocesador leería a veces señal, a veces no, con lo que el software tendría que complicarse para saber si eso es una señal alterna.



Con el condensador evitamos este problema.

Tal como está colocado el condensador, cuando no hay entrada el micro leerá un nivel alto. En cuanto haya corriente por el LED el transistor se saturará y aparecerá un nivel bajo en la entrada del micro. Cuando la señal de entrada baje, el transistor dejará de estar saturado, pero el condensador tardará cierto tiempo a adquirir la tensión de alimentación. Si antes de que suba 1 voltio volvemos a pasar corriente por alguno de los dos LEDs, volverá a bajar la tensión.

Los valores de R2 y C1 deberán calcularse de acuerdo con la frecuencia de la señal de entrada. En realidad se calculará teniendo en cuenta la tensión máxima permitida en un tiempo determinado de ausencia de tensión de entrada.

$$V = \frac{1}{C} \int i \cdot dt$$

Como la tensión en un condensador es:

y suponiendo que la corriente es constante, aunque solo será cierto si no dejamos subir casi la tensión. Por ejemplo de 0,2 a 0.5 voltios y dejaremos que esa tensión se alcance en un periodo de 50 Hz (20 ms.) La tensión de alimentación supondremos que es 5 y la aproximación que hacemos es que sube tan poco que puede mantenerse constante.

$$V_c \approx \frac{1}{C} I \cdot t \approx \frac{5-0,2}{C \cdot R} \cdot T \quad 0,3 = \frac{4,8}{C \cdot R} \cdot 20 \cdot 10^{-3}$$

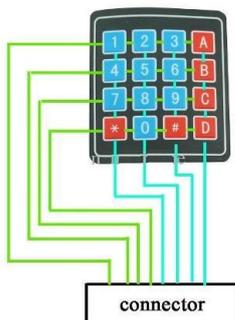
$$C \cdot R = \frac{4,8}{0,3} \cdot 20 \cdot 10^{-3} = 0,32$$

Podemos poner R = 10 K y C resultará 0,32/10000=32 uF. Ponemos un valor estándar mayor, 47 uF y aseguramos que funciona correctamente.

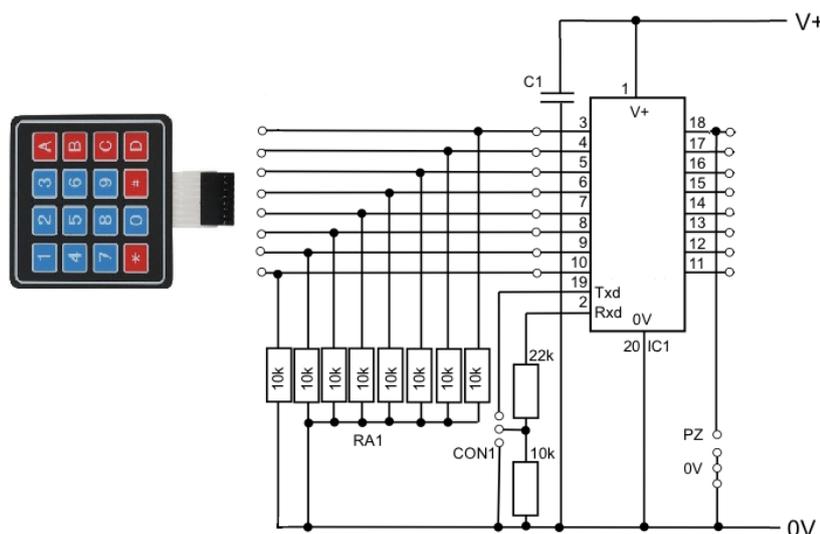
## Lectura de un teclado.

Teclado matricial es un arreglo de botones conectados en filas y columnas, de tal manera que se pueden leer varios botones con el mínimo número de entradas requeridas. Un teclado matricial 4x4 solamente ocupa 4 líneas de un puerto para leer las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 entradas (un solo puerto) de un microcontrolador.

El funcionamiento básico del teclado matricial



consiste en rastrear la unión entre una fila y una columna, mediante un algoritmo que descifra cual tecla fue presionada.



En la imagen se observa un diagrama de conexión básico para conectar un teclado matricial, sin importar a que puerto se conecte el teclado, se debe tener en cuenta la conexión de las resistencias de 10kΩ a masa a fin de generar un "0" de referencia cuando el teclado no está siendo presionado y de esta manera evitar lecturas indeseadas por ruido eléctrico.

Uno de los pensamientos lógicos para usar con un programa sería:

- Enviar un "1" lógico a uno de los pines de las columnas.
- Censar todas las filas, para de esta manera saber que botón se apretó. Por ejemplo si envió un "1" lógico a la columna 1 y detecto en las filas los siguientes datos:
  - A. fila 1 : 0
  - B. fila 2 : 1
  - C. fila 3 : 0
  - D. fila 4 : 0

Significaría que se apretó el botón que corresponde a las coordenadas: **columna 1; fila 2**. Que en este caso si nos guiamos por la imagen del teclado anterior, significaría que se oprimió la tecla correspondiente al número "2".

Este proceso se repetiría para las columnas restantes.

También se podría hacer el proceso inverso, trabajar con las filas y censar las columnas. Todo depende de la creatividad del programador.

## Interface de las salidas digitales.

Al igual que con las entradas, las tensiones que maneja el microprocesador y las que del exterior no son iguales. Con una tensión de 3,3 o 5 V. podremos activar un led y poco más. Si queremos conectar un motor necesitaremos algún relé o contactor. A continuación veremos varias formas de conectar las salidas:

### Activar un LED

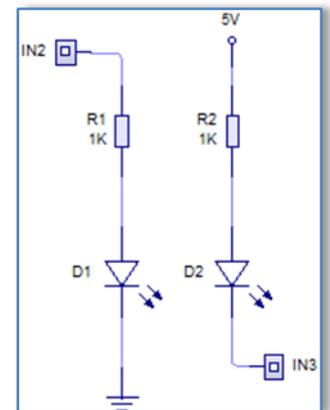
#### Activar un relé con bobina de 5Vcc

#### Activar un relé con bobina de más de 5 Vcc

#### Salidas en colector abierto

### Activar un LED

Activar un LED es lo más fácil que podemos hacer con una salida. Si queremos que el LED se encienda cuando a la salida ponemos un 1 montaremos la configuración de la izquierda. Si queremos que el LED se encienda cuando ponemos un 0 pondremos la configuración de la derecha. Aparentemente son iguales, salvo que se activan por 0 o por 1, pero hay alguna cosa más que hay que tener en cuenta. En la mayoría de microcontroladores, sobre todo antiguos, la tensión de salida a nivel bajo está más cerca del 0 que la salida a nivel alto de la alimentación, por lo que suele ser mejor activar las salidas a nivel bajo. Para calcular la resistencia tendremos que saber que corriente queremos que pase por el LED. Para cada LED habrá que ver su curva característica y ver para la iluminación que pretendemos que corriente necesita y que tensión cae en sus bornes, además saber que tensión proporciona el microprocesador para esa corriente. Una corriente típica puede ser de 10 a 20 mA y la tensión en el LED suele ser de 1.5V. Si no conocemos la tensión que proporciona el microprocesador para determinada corriente podemos suponer 0 o VDD, cometiendo un error sin demasiada importancia. Para el caso del circuito de la derecha el cálculo de la resistencia se hará como sigue. (El mismo cálculo sirve para el otro circuito).



Supongamos que queremos que pasen 20 mA. Y el microcontrolador esta alimentado a 5 VDC.

$$5 = I * R + V_{LED} \quad 5V = 20mA * R + 1.5V \quad R = \frac{5 - 1.5}{20} = 175 \Omega \approx 180 \Omega$$

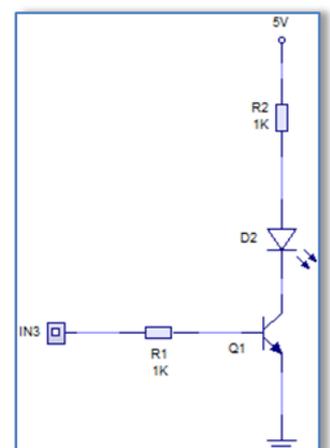
Calculo que se desprende de la utilización de las leyes ya conocidas.

Conviene mirar las características del LED, pues hay LEDs que con esa corriente tienen un brillo fortísimo y otros tienen poco brillo.

El consumo de este LED (20 mA) es mucho mayor que el consumo del propio microprocesador, por lo que es conveniente utilizar algún modo de disminuir su consumo, sobre todo si hay varios LEDs. Un método consiste en no iluminar el LED continuamente y debido a la persistencia del ojo parece que está siempre encendido. Por ejemplo podemos activarlo 1 ms cada 5 ms, con lo que el consumo disminuye a la quinta parte y el efecto es una iluminación constante aunque algo más débil. Otro método mucho mejor de bajar el consumo es hacer que los LEDs parpadeen y que el ojo vea un parpadeo. Puede iluminarse por ejemplo 5 ms cada segundo. Esos 5 ms son suficientes para que veamos un destello y el consumo se ha reducido en 200 veces.

### Activar un LED u otro componente con corriente elevada:

El microcontrolador no puede dar por una de sus salidas más de 25 mA (dependiendo de cada microcontrolador), por ello si tenemos que alimentar algún





Manual de introducción al uso de microcontroladores PIC ARDUINO  
dispositivo que necesite más de los 25 mA deberemos utilizar un circuito algo más complejo. Si bien no siempre En este caso vamos a activar un led que consuma 500 mA.

Los datos de los componentes utilizados son:

- Corriente del led 500 mA
- Tensión en el led a 500 mA = 2 V.
- $\beta$  del transistor > 100
- El transistor deberá estar saturado.

Como por el LED deben circular 500 mA, caen en el 2 V y el transistor en saturación tiene una tensión colector-emisor de 0,2 V tenemos:

$$5 = I_2 \cdot R_2 + V_{LED} + V_{CEsat} \quad 5V = 500mA \cdot R_2 + 2V + 0.2 \quad R_2 = \frac{5 - 2 - 0.2}{0.5} = 5,6 \Omega$$

$$\text{Ojo, potencia en la resistencia} = I^2 \cdot R_2 = 0,5^2 \cdot 5,6 = 1,4 W$$

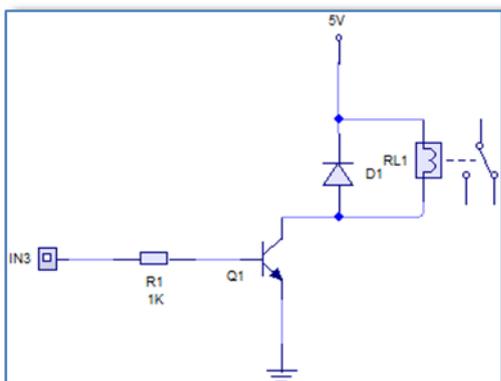
El LED necesita 500 mA. Como la  $\beta$  del transistor es por lo menos 100, necesitamos al menos  $500/100 = 5$  mA

$$5 = I_1 \cdot R_1 + V_{BEsat} \quad R_1 = \frac{5 - 0.8}{I_2 / \beta} = \frac{4,2}{500/100} = \frac{4,2}{5} = 0,84 K\Omega$$

Con esa resistencia estará justo para la saturación. Ponemos  $R_1 = 470 \Omega$  para asegurar la saturación.

Aunque hayamos hecho los cálculos con un LED, el proceso sería similar para otros componentes.

## Activar un relé con bobina de 5 Vcc



Uno de los casos más habituales de uso de una salida de un microprocesador es para activar un relé. Si tenemos un relé cuya bobina sea de 5 VDC, y el microcontrolador lo alimentamos a 5 VDC, podemos conectar directamente el relé a la salida, de forma que se active a nivel alto o a nivel bajo. Si leemos las características de un relé normal, veremos que la potencia que consume la bobina está en torno a los 500 mW y en un relé de bajo consumo en torno a los 100 mW. El problema es que la corriente necesaria para activar el relé sale o entra toda en el micro. Si utilizamos un relé de 5 VDC y consume 500 mW, la corriente que pasa por él, y por el microprocesador es de 100 mA, mucho mayor que los 25 mA que dice el fabricante que soporta la

entrada/salida. Con un relé de bajo consumo podría conectarse directamente. Pero existe una forma mucho mejor, que es utilizar un transistor entre el micro y el relé.

La ventaja de esta configuración es que la corriente que atraviesa el relé no pasa por el micro. El funcionamiento es el siguiente. Cuando la salida está a 0, no hay corriente de base, por lo que el transistor está cortado y no circula corriente por el relé. Cuando ponemos un 1 a la salida, existe una corriente de base que hace que el relé se ponga en saturación, cayendo 0.2 V en la unión CE y el resto en el relé, activándolo.

Para calcular  $R_1$  hay que tener en cuenta que el transistor deberá estar en saturación. Si la corriente que necesita el relé es de 100 mA, que será la corriente  $I_c$ , para que el transistor este en saturación se debe cumplir que  $I_C > \beta \cdot I_B$ . La  $\beta$  la buscamos en un las hojas de características del transistor y cogeremos la menor

Manual de introducción al uso de microcontroladores PIC ARDUINO

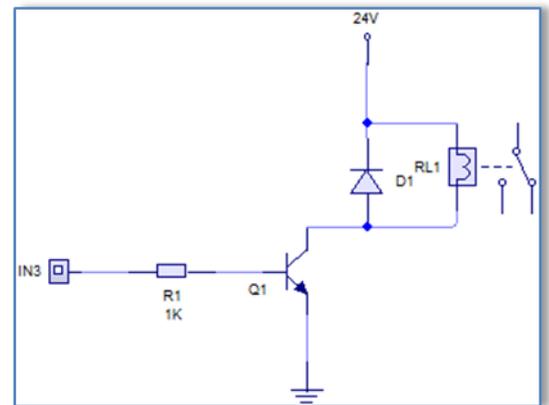
posible. Supongamos que  $\beta$  vale 100 y la corriente que necesita el relé es de 100 mA.  $100 \text{ mA} > 100 \text{ IB}$ . Por lo que la corriente mínima que debe salir del micro es de 1 mA. Para asegurarnos que está saturado ponemos algo más, por ejemplo 2 mA. Ahora calcularemos R1. La tensión en la salida del micro habría que verla en la gráfica, pero podemos suponer que es 5 V y la VBE de 0.8 V.

$$5 = I \cdot R + V_{BE} \quad 5 = I \cdot R + V_{BE} \quad 5 = 2 \text{mA} \cdot R + 0.8 \quad R = \frac{5 - 0.8}{2} = 2 \text{K}1 \approx 2 \text{K}2$$

El diodo D1 debe ponerse siempre en paralelo y en inversa con la bobina del relé (para bobinas alimentadas con corriente continua). Este diodo se utiliza porque al pasar el transistor de saturación a corte hay un cambio muy brusco de la corriente que pasa por el colector y por la bobina, el cual genera una tensión muy elevada en sus bornes. El diodo permite que al cortar el transistor, la corriente que pasa por la bobina siga circulando por el diodo, atenuándose de una forma no tan brusca.

### Activar un relé con bobina de más de 5 Vcc

Los relés con bobina de 5 V son muy cómodos de poner, pues la alimentación es la misma que la del microprocesador. Esa aparente ventaja tiene un problema, la bobina de los relés normales consume aproximadamente 500mW, y a 5 V la corriente es de 100 mA. Si hay 10 relés consumiríamos 1 A, que ya es una corriente respetable. Si los relés fuesen de 12 V, la corriente sería de 42 mA, menos de la mitad; y si los relés son de 24, la corriente que pasa por cada una es de 21 mA. En este caso para 10 relés el consumo con los 10 relés activados es de 210 mA. Los reguladores de tensión normales no suelen pasar de 1 A. Por ello, si hay varios relés puede ser más práctico tener una alimentación de 24 V para los relés y otra de 5 para el microprocesador. Esta tensión de 5 V puede obtenerse a partir de la de 24V, por medio de un regulador, pues la corriente que consume el microprocesador y los circuitos adyacentes suele ser muy pequeña.

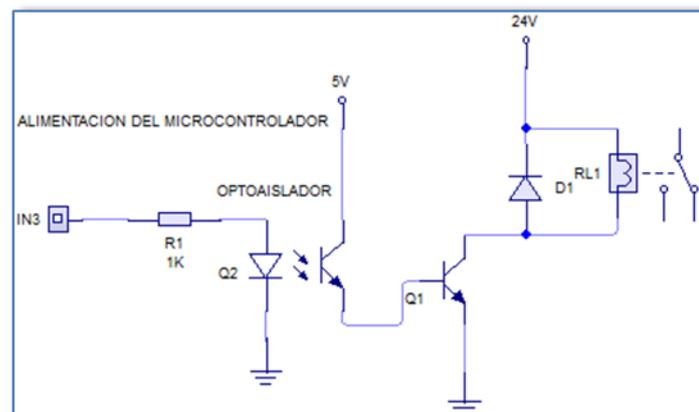


La colocación de los relés de mayor tensión no tiene ningún problema, simplemente las fuentes deben tener la misma masa. Con la diferencia que hay que rehacer los cálculos para la resistencia que hay en la base del transistor. Para 24 V la corriente necesaria es de 20,8 mA, si la  $\beta$  es de 100. La corriente de base deberá ser, al menos de 0.208 mA. Pongamos 0.6 mA para asegurar la saturación. Ahora calcularemos R3. La tensión en la salida del micro habría que verla en la gráfica, pero podemos suponer que es 5 V y la VBE en saturación de 0.8 V.

$$5 = I \cdot R + V_{BE} \quad 5 = I \cdot R + V_{BE} \quad 5 = 0.6 \text{mA} \cdot R + 0.8 \quad R = \frac{5 - 0.8}{0.6} = 7 \text{K} \approx 6 \text{K}8$$

Si el transistor se fundiese, podría llegar una tensión muy elevada al microprocesador, aunque hay una resistencia por medio. Para evitar este posible problema, y evitar problemas de ruido, puede ser adecuado aislar la salida por medio de un opto acoplador.

En este caso están totalmente aislados el microprocesador y el relé. Para que este circuito funcione adecuadamente es conveniente utilizar un optoaislador Darlington, pues en los optoaisladores normales el valor equivalente a la  $\beta$  del transistor es muy bajo, no suele llegar a la unidad; con un optoaislador Darlington tendremos un valor de  $\beta$  equivalente que puede llegar a 10. En lugar de  $\beta$  se le llama "relación de transferencia". El cálculo de R1 se





## Manual de introducción al uso de microcontroladores PIC ARDUINO

hace teniendo en cuenta la corriente que necesita el relé. Como el relé es de 24 V y 500 mW, la corriente que necesita es de:

$$500\text{mW}/24\text{V}=20,8 \text{ mA}$$

Si la relación de transferencia mínima del par Darlington es de 10, la corriente en la base deberá ser mayor que  $20,8 \text{ mA}/10= 2,8 \text{ mA}$

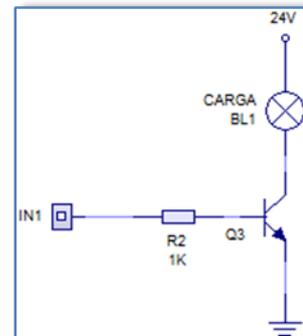
Para asegurar la saturación del transistor, en lugar de 2,8 mA pondremos 6 mA. Y según el fabricante, con esa corriente la tensión en el LED es de 1.5 V.

$$V_{\text{ent}} = I * R + V_{\text{LED}} \quad 5\text{V} = 6\text{mA} * R + 1.5\text{V} \quad R = \frac{5 - 1.5}{6} = 0.7\text{K} \approx 680 \Omega$$

Otro problema es la corriente máxima que puede soportar el optoaislador, dependiendo del optoaislador puede variar de 50 a 150 mA. En el caso de utilizar relés de 24 V no tendremos ningún problema.

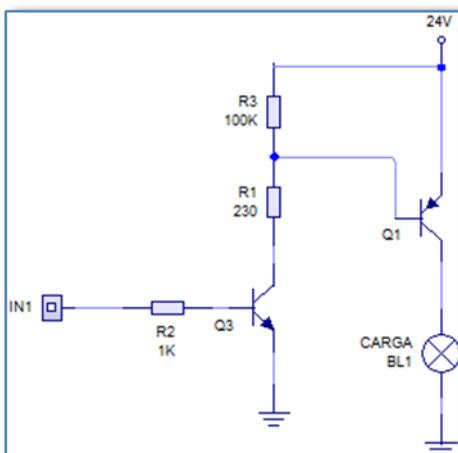
## Salidas en colector abierto:

Los relés tienen la gran ventaja de un total aislamiento entre la carga y nuestro circuito, aparte de que visualmente se puede comprobar su estado, pero tienen el problema de que no pueden funcionar a frecuencias muy elevadas. Para hacer salidas rápidas, e incluso para las normales, se suelen utilizar salidas en colector abierto. Estas salidas se basan en que el circuito que conmuta la salida no es un relé, sino un transistor. Por ello la velocidad de conmutación puede ser rapidísima, aunque las tensiones y corrientes que se pueden conmutar no suelen ser muy elevadas. Con una resistencia y un transistor podemos realizar una salida en colector abierto aumentando la capacidad de corriente de la salida. Para corrientes o tensiones elevadas el precio del transistor haría poco viable esta solución. La salida puede ser del tipo PNP o del tipo NPN.



## Salida colector abierto NPN.

El esquema adjunto muestra una salida del tipo NPN (la nomenclatura viene dada por el tipo de transistor de salida. En este caso, la carga se conecta entre alimentación, normalmente 24 VDC y el colector del transistor. La resistencia R1 debe ser tal que el transistor esté en saturación cuando conduzca, para que así la tensión VCE sea muy baja, aproximadamente 0.2 V. En este caso, y suponiendo que la corriente en la carga pueda llegar a ser de hasta 1 A, y la  $\beta$  del transistor sea de 100,  $I_c < \beta * I_B$ . Por lo tanto  $I_B > 10 \text{ mA}$ . Haremos que  $I_B = 20 \text{ mA}$ . Para ello  $R = (5 - 0.7)/20 = 0.215 \text{ K}\Omega$ . Pondremos  $200 \Omega$  y seguro que cumplimos. Con una resistencia menor aseguraríamos que para más rango de corriente en la salida el circuito funcionaría correctamente, pero la corriente de salida del micro sería más grande.



La potencia disipada en el transistor es  $P = I * V$ . Si está saturado  $V = 0,2$  si no está saturado la tensión puede ser mucho mayor y la potencia disipada en el transistor muy grande.

## Salida colector abierto PNP.

La salida en colector abierto tipo NPN es mucho más sencilla que la salida tipo PNP, pero activar una salida con un nivel bajo resulta en muchos casos engorrosa. Por ello, la salida tipo NPN es más utilizada en la industria. En esta salida cuando la tensión de salida es alta, la salida se activa. Otra ventaja que tiene es que al haber 2 transistores, la corriente de salida del micro se ve amplificada 2 veces, por lo que sale menos corriente del micro. El circuito es más complejo, aquí vemos un ejemplo diseñado para que pueda suministrar 10 A con un  $\beta$  de al menos 100 en cada transistor. Hay que tener en cuenta que el

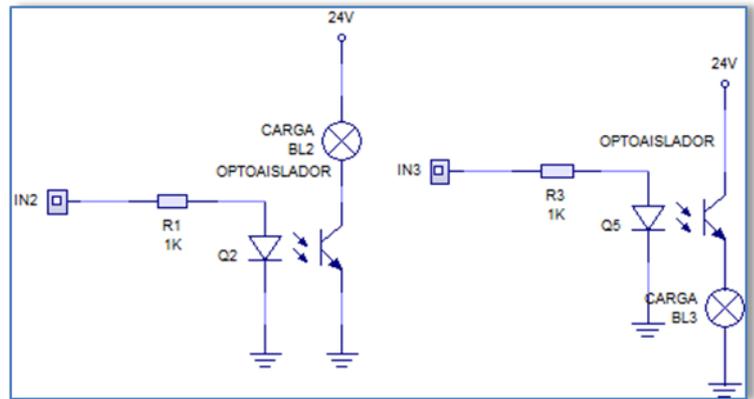
transistor de salida debe soportar la corriente exigida, 10 A en este caso, y la potencia disipada,  $0,2 \text{ V} \times 10 \text{ A} = 2 \text{ W}$ . Con corrientes elevadas la tensión colector-emisor de saturación suele ser algo mayor.



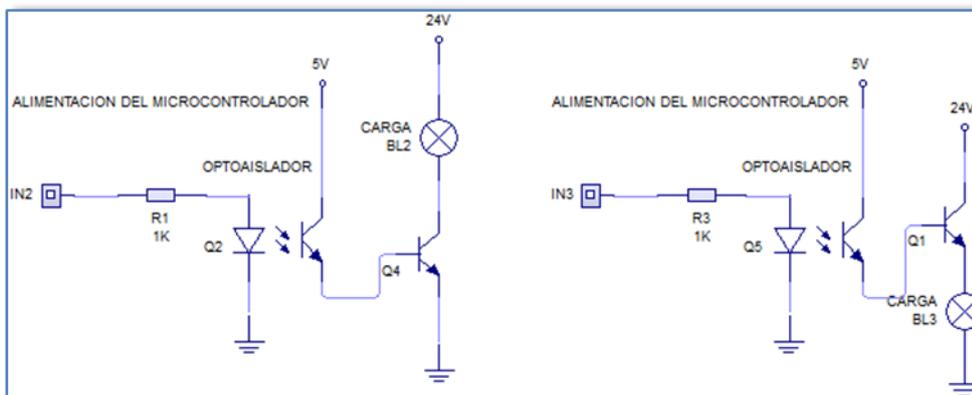
## Salidas en colector abierto aisladas:

La salida con relé tiene como principales virtudes que aísla la salida (puede haber tensiones muy elevadas en la salida mientras que la bobina es de 5 o 24 V) y permite grandes corrientes.

Las salidas en colector abierto pueden manejar grandes señales, depende del transistor, pero no están aisladas. Para aislarlas sustituiremos el transistor en colector abierto por un opto acoplador, y mejor si es Darlington.



Debido al uso del optoaislador, las tensiones a la salida no tienen por qué tener nada que ver con las de la entrada, y como el transistor no tiene base, podemos poner la resistencia a la salida arriba o abajo. En estos casos lo normal es que tengamos el colector y el emisor del transistor en la salida, conectando la carga y la alimentación según convenga.



Muchas veces la salida del microcontrolador va conectada a la entrada de algún equipo y ese equipo dice que en la entrada hay que poner un contacto libre de tensión. Ese contacto libre de tensión puede ser un relé o un opto acoplador.

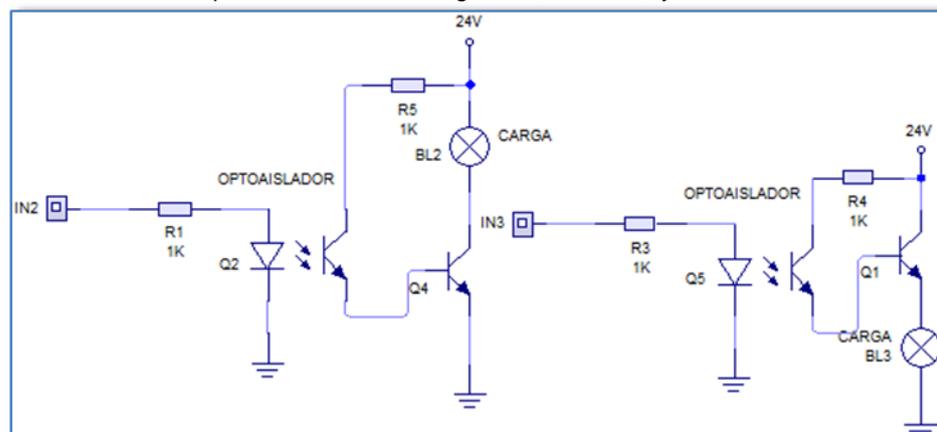
El problema más importante de estos circuitos es que el transistor del opto acoplador no aguanta grandes corrientes. Para solucionar ese problema podemos hacer un transistor Darlington con el transistor del opto acoplador y uno de potencia.

Estos circuitos pueden aguantar muchos amperios si los aguanta el transistor. La corriente que circula por el transistor del opto acoplador será del orden de  $\beta$  veces menor que la corriente que pase por el transistor de potencia.

En digital estamos acostumbrados a trabajar en corte y saturación, pero en los transistores Darlington el primer transistor se puede saturar, pero no el segundo. El segundo transistor no suele tener una tensión CE menor que 0.9 V. Por poner un ejemplo, supongamos que por un transistor pasan 20 A. Si la tensión CE fuera de 0.2 V, la potencia disipada en el transistor sería de 4 W, pero si la tensión CE es de 1 V, la potencia disipada es de 20W. Además si la corriente es grande la tensión CE puede ser bastante grande, 2 o 3 V, y en esos casos la potencia disipada puede ser muy grande.

Este problema puede solucionarse del siguiente modo:

El de la izquierda es con salida activa a nivel bajo. El de la derecha con salida a nivel alto. La resistencia del colector del opto acoplador deberá calcularse para que estén saturados los dos transistores, si es muy grande saturara el primero pero no habrá



Manual de introducción al uso de microcontroladores PIC ARDUINO  
suficiente corriente para saturar el segundo; si es muy pequeña no saturara el primero.

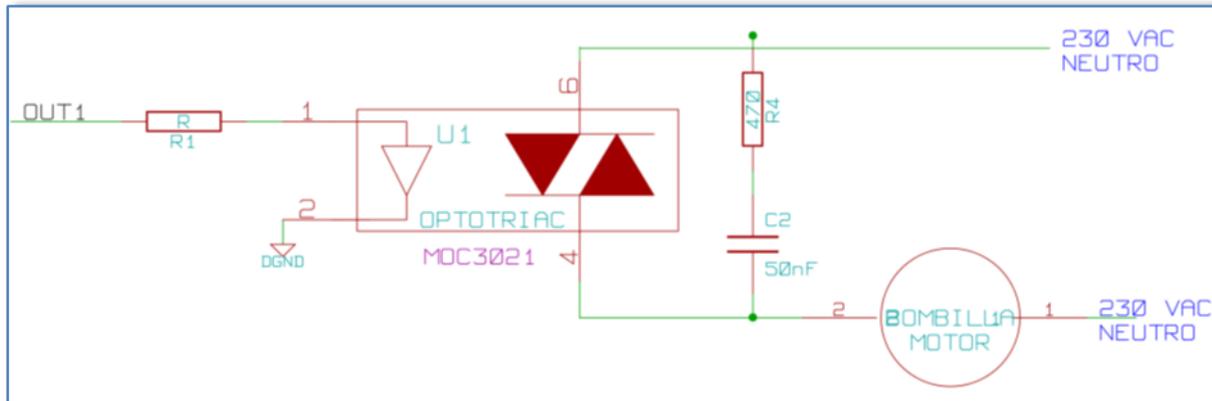
## Salidas con triac:

Para conectar a la salida elementos alimentados a tensiones elevadas y en alterna (230 Vac), podemos utilizar los relés, pero son grandes y no podemos hacer conmutaciones demasiado rápidas.

Para esto podemos utilizar los triacs y optotriacs.

Para corrientes pequeñas podemos utilizar directamente un optotriac.

El optotriac haría de interruptor, cortando o dejando pasar la corriente. La salida del optotriac está

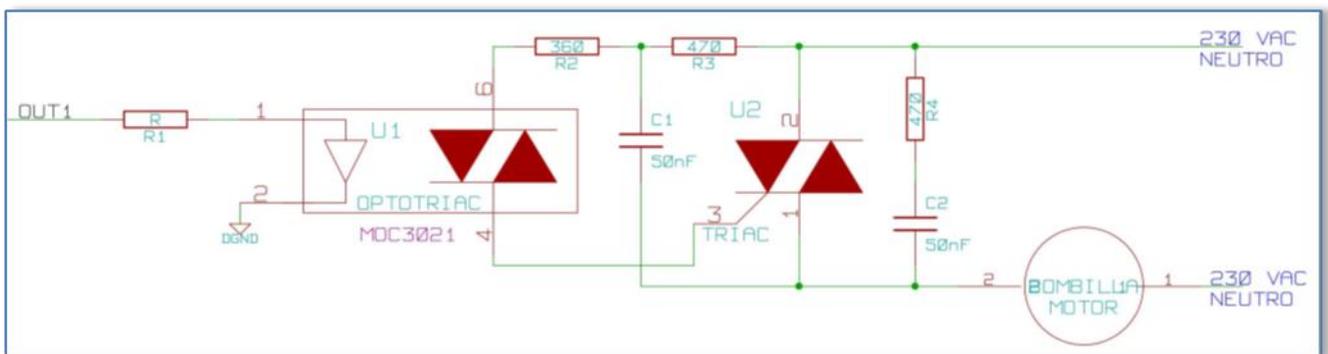


conectada a 230 Voltios, y el circuito puede alimentar una bombilla o un motor pequeño. Como el optotriac aguanta poca corriente debemos utilizar este circuito solo en casos en que la corriente sea muy pequeña.

Para corrientes mayores podemos utilizar el siguiente circuito:

Este circuito es una copia del manual del optotriac MOC3021 de Motorola. En él la corriente máxima que soporta el circuito no es la del optotriac, sino la del triac Q4, por lo que este puede ser de las dimensiones adecuadas.

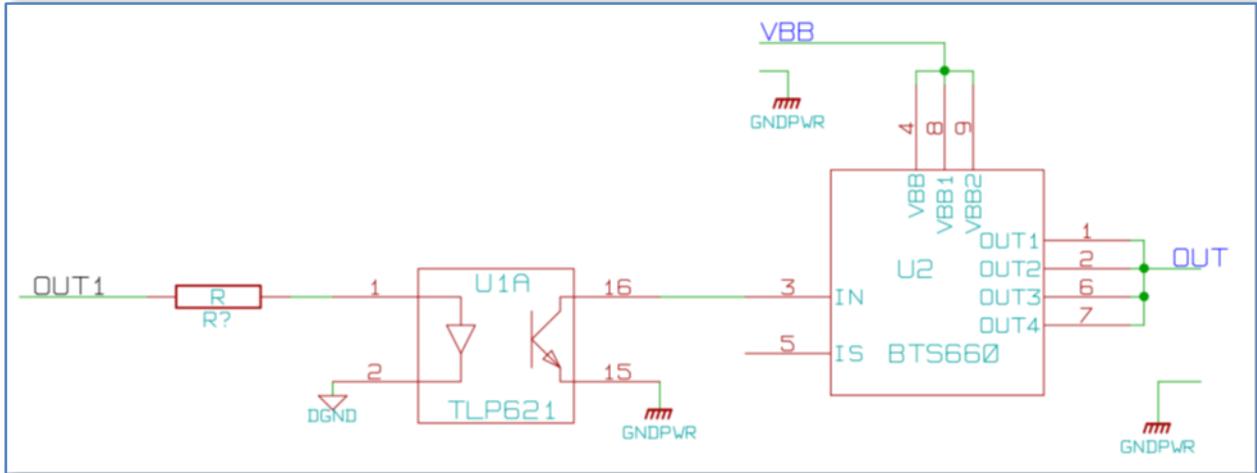
Al conectar la carga a 230 V conviene saber cuál es la fase y cual el neutro. En teoría da igual, pero es recomendable que en circuitos fijos (sin enchufe) el interruptor corte la fase, para que en ningún punto del motor haya tensión cuando esta desconectado.



## Salidas con interruptor basado en MOSFET:

Una tipo de salida muy recomendable son las salidas con integrados diseñados específicamente para trabajar como interruptores, estos circuitos suelen estar basados en transistores MOSFET con el circuito de control incorporado. Uno de ellos es el BTS660S, capaz de conmutar hasta 70A a tensiones de 60 VDC. Su resistencia

en



conducción es de solo 9 mΩ y está protegido contra temperatura, corriente y tensión. Además su precio es de unos 3 €, más barato que cualquier relé que soporte esas corrientes.



## Manejo de motores DC directos y paso a paso

### Objetivo y resumen del proyecto

El objetivo de este proyecto es crear un equipo con el que comprender y controlar, desde el ordenador, el funcionamiento de un motor paso a paso bifásico de imanes permanentes, a través de una comunicación USB, la ayuda de un microprocesador y el imprescindible driver del motor paso a paso.

Uno de los elementos utilizados en el control de un motor paso a paso es el microprocesador, encargado tanto de enviar las consignas al controlador (driver), como de captar las señales provenientes del encoder. Para cumplir esta función se ha decidido la utilización del microprocesador 18F2550 de MicroChip, principalmente por su capacidad de comunicación mediante el protocolo USB 2.0 con el ordenador. Así como los demás pics que no se conecten al USB.

### Introducción

El motor paso a paso es el convertidor electromecánico que permite la conversión de una información en forma de energía eléctrica, en una energía mecánica y una información de posición. Está constituido por un estator cuyos devanados se llaman fases y un rotor de un elevado número de polos. Su funcionamiento es síncrono y la alimentación cíclica de sus fases debe originar en cambio de configuración un giro elemental del rotor, constante, llamado paso.

Existe una gran diversidad de modelos de estos motores dependiendo del número de fases de su estator, de si la alimentación de estas es unipolar o bipolar, del número de paso por vuelta y de si su rotor es de reluctancia variable, imanes permanentes o híbridos.

En cuanto al control, existen tres modos de realizarlos, paso entero, medio paso y micropaso.

En el paso entero, cada vez que se modifica la alimentación de las fases del estator se avanza un paso disponiendo de par nominal del rotor.

En el medio paso se avanza sólo medio paso con lo que se dispone de mejor resolución, pero el par en las posiciones situadas entre pasos regulares se reduce a la mitad.

Estos dos tipos de funcionamiento disponen en el mercado de gran variedad de integrados para su control.

El funcionamiento en micropaso consiste en alimentar al mismo tiempo varias fases a la vez con corrientes medias distintas, de modo que la posición media del flujo en el entrehierro se puede fijar en cualquier posición. Con este funcionamiento se consigue una resolución inmejorable y existen en el mercado distintas tarjetas de control basadas en microprocesador.

Merece la pena comentar que el motor paso a paso es la primera de las máquinas eléctricas que sin el uso de la electrónica no tiene razón de ser.

El control de posición de motores paso a paso se puede efectuar en lazo abierto siempre que se tomen las precauciones necesarias para no perder ningún paso. Indicaremos que utilizando técnicas de PWM para el control de la corriente, asegurando de esta manera un aprovechamiento máximo de par y con la programación adecuada de aceleración y deceleración, se puede trabajar perfectamente en lazo abierto siempre que las variaciones que el par de carga sean conocidas de antemano. Es en estas aplicaciones de carga conocida donde el motor paso a paso tiene sus posibilidades industriales. En el caso de tener que accionar cargas desconocidas, su funcionamiento en lazo cerrado sería del todo perfecto, pero el coste del transductor de realimentación generalmente no justifica esta aplicación.

## Descripción de los motores paso a paso

La siguiente figura representa la sección de un típico motor paso a paso. Estudiaremos de forma sencilla el funcionamiento de esta máquina.

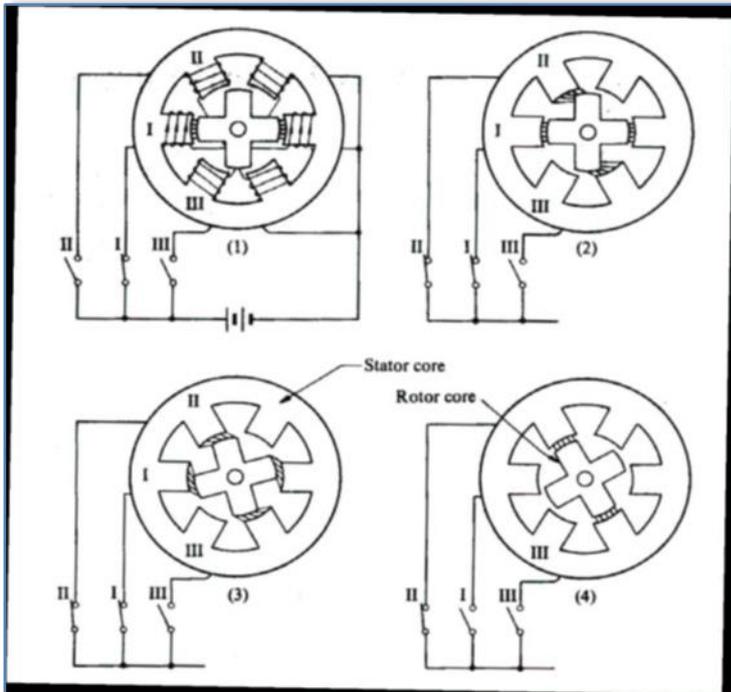


Figura 2.1-Esquema básico de funcionamiento de un motor paso a paso

El estator tiene seis polos salientes, mientras que el rotor solamente dispone de cuatro. Ambos se suelen construir generalmente de acero blando. Los polos del estator se han bobinado para formar tres fases, cada una de las cuales consta de dos bobinas conectadas en serie y situadas físicamente en polos opuestos. La corriente se aplica desde una fuente de potencia DC a través de los interruptores I, II y III.

Estando el motor en el estado (1) es la fase I que está excitada. El flujo magnético que cruza el entrehierro debido a esta excitación se indica con flechas. En este estado los dos polos del estator pertenecientes a la fase I están alineados con dos de los cuatro polos del rotor, permaneciendo este en posición de equilibrio.

Continuamos con el estado (2). Para ello cerramos el interruptor II. Primeramente se establece el flujo representado en (2), creándose un par en sentido anti-horario, debido a las tensiones Maxwell, que obliga al rotor a alcanzar la posición de equilibrio del estado (3), girando para ello 15°.

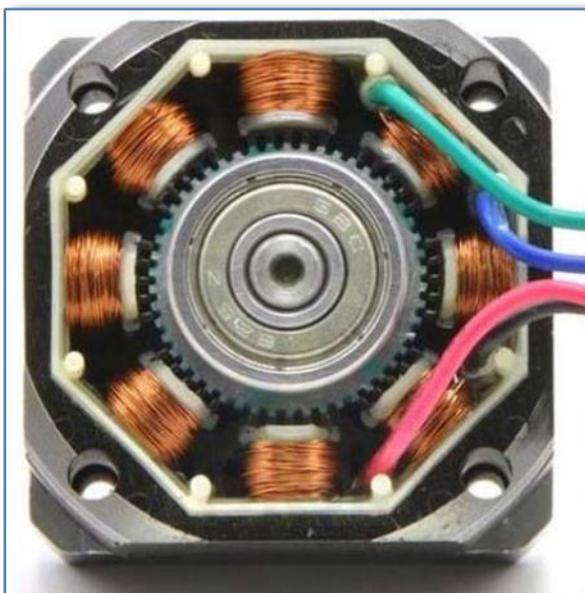


Figura 2.2

Cada vez que realizamos una apertura o cierre de un interruptor se produce un giro de 15° de rotor. Este ángulo fijo se denomina ángulo de paso y es una característica básica dentro de este tipo de motores. Existen motores paso a paso con una extensa variedad de ángulos de paso dependiendo de la mayor o menor resolución que necesite. Así existen motores con ángulos que van desde las décimas de grado hasta los 90°.

Si ahora abrimos el interruptor I, el rotor alcanza la posición de equilibrio representada en el estado (4).

Siguiendo una secuencia correcta de control de la apertura y cierre de los interruptores, podremos girar el motor en el sentido y a la velocidad que deseemos con la ventaja de no tener que utilizar ningún tipo de realimentación, Además el error de posición que puede tener este tipo de motores no es acumulativo y tiende a

cero en cuatro pasos, es decir cada 360° eléctricos. Cada cuatro pasos el rotor vuelve a la misma posición con respecto a la polaridad magnética y a la trayectoria del flujo. La precisión en el posicionado es un factor que mide la calidad de estos motores. Se diseñan de modo que tras recibir una señal eléctrica pasen de una situación de equilibrio a otra posición de equilibrio diferente separada de la anterior un determinado ángulo. Esta precisión depende en gran manera del mecanizado del rotor y estator con lo que su fabricación es delicada. Cuando una carga se aplica sobre el eje, se produce un par elevado que trata de posicionar el rotor en su posición natural de equilibrio. La responsabilidad de que este par sea mayor o menor recae en el

Manual de introducción al uso de microcontroladores PIC ARDUINO  
entrehierro. Cuanto más pequeño sea, y esto depende del al calidad de la fabricación, el par que presente el motor a la carga y su precisión serán mayores.

Existen dos conceptos que sirven para diferenciar el comportamiento del motor paso a paso en cuanto al par mencionado anteriormente.

### Holding torque

Definido como el máximo par estático que se le puede aplicar al eje de un motor excitado sin causarle rotación continua.

### Detent torque

Definido como el máximo par estático que se le puede aplicar al eje de un motor no excitado sin causarle rotación continua.

En general cuanto mayor sea el "holding torque" menor es el error de posición debido a la presencia de una carga externa sobre el eje. El "detent torque" aparece solamente en los motores paso a paso de imanes permanentes que se discutirán posteriormente.

## Clasificación de los motores paso a paso

Anteriormente se ha dedicado un breve apartado para explicar el principio general de funcionamiento de estos motores de una manera sencilla. Vamos a profundizar un poco más ya que dependiendo de su estructura física se puede hacer una clasificación en función del principio de funcionamiento.

### Motores de reluctancia variable

Pertenecen a esta categoría la mayoría de los motores paso a paso que se encuentran en el mercado. La figura mostrada a continuación nos servirá para indicar su funcionamiento:

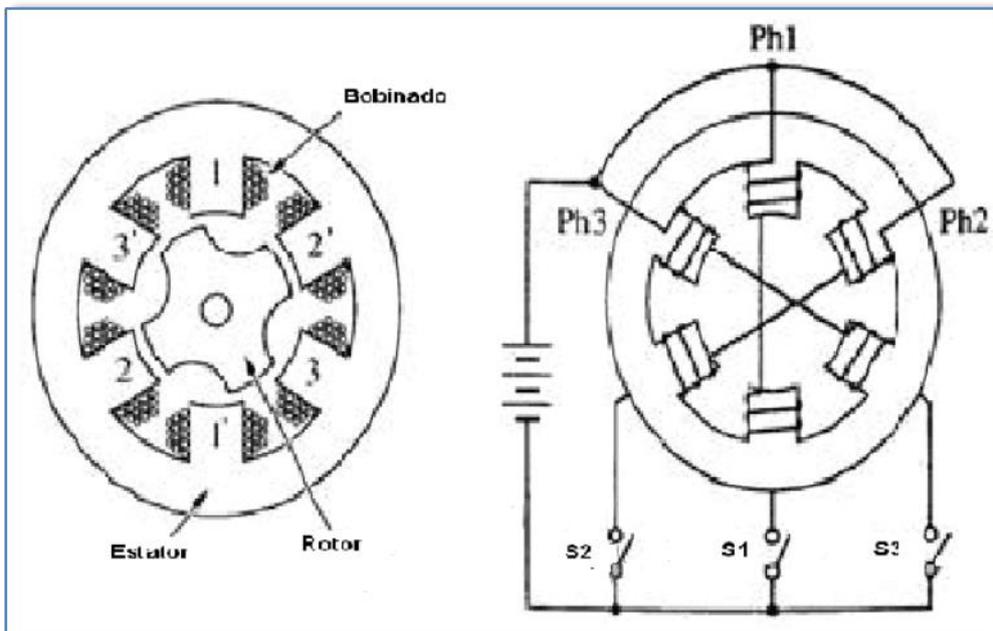


Figura 2.3-Sección de un motor paso a paso de reluctancia variable

En esta figura se representa un motor de tres fases con seis dientes salientes en el estator. Cada par de polos separados 180º entre sí constituyen una fase. Las bobinas de cada fase arrolladas sobre los correspondientes polos se conectan en serie. El rotor consta de cuatro polos. Tanto el rotor como el estator deben de estar

construidos con materiales de alta permeabilidad magnética y ser capaces de permitir el paso de un gran flujo magnético incluso cuando se aplique una pequeña fuerza magnetomotriz.

Aun cuando no siempre tiene por qué ser así, vamos a asumir que las polaridades de los polos pertenecientes a la misma fase son opuestas. Por tanto, en la figura 2.4 constituirán el polo norte y los polos I', II' y III' el polo sur cuando circule corriente por sus devanados.

Figura 2.4- disposición de líneas de flujo



Manual de introducción al uso de microcontroladores PIC ARDUINO

La corriente de cada fase se controla mediante la apertura y cierre de los diferentes interruptores. Si una corriente se aplica a las bobinas de la Fase 1º, dicho de otro modo, si excitamos las Fase 1, se establecerán unas líneas de flujo similares a las representadas en la figura 2.4

El rotor se posicionará de modo que queden alineados dos polos opuestos suyos con los polos I y I' del estator. Cuando los polos del rotor y del estator quedan alineados se minimiza la reluctancia magnética del circuito magnético y el motor se encuentra entonces en una posición de equilibrio. Si el rotor tiende a moverse de su posición de equilibrio debido al par generado por la presencia de una carga externa, internamente se genera un par en sentido contrario que intenta conducir al rotor a su posición de equilibrio original. La figura 2.5 ilustra esta situación.

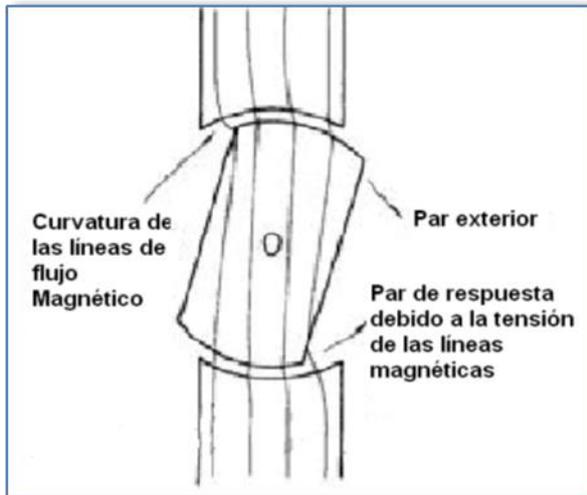


Figura 2.5- disposición de líneas de flujo al aplicarse un par externo

En esta figura el par externo se aplica en el sentido horario y el rotor se desplaza en esa misma dirección. Como resultado de este desplazamiento las líneas de flujo magnético que atraviesan el entrehierro y que inicialmente, antes de aplicar ningún par externo, eran rectilíneas se curvan en los bordes de los polos del estator y rotor. Estas líneas magnéticas tienden a ser cortas y rectilíneas como sea posible, creando una tensión, conocida como *tensión de Maxwell*, que provoca un par de sentido contrario al par inicial que había distorsionado estas líneas de flujo. Se puede ver en la

misma figura como cuando los polos del rotor y del estator están desalineados la reluctancia magnética es mayor, de modo que el motor de reluctancia variable trabaja siempre en condiciones de reluctancia mínima. Veamos ahora qué ocurre cuando la Fase 1 se desconecta y se conecta las Fase 2. La reluctancia magnética del motor vista desde la fuente de potencia DC se incrementará súbitamente justo después de la conmutación de los interruptores. El resultado se puede ver en la figura 2.6, el rotor girará 30º en sentido anti-horario con el fin de restablecer las condiciones de reluctancia mínima.

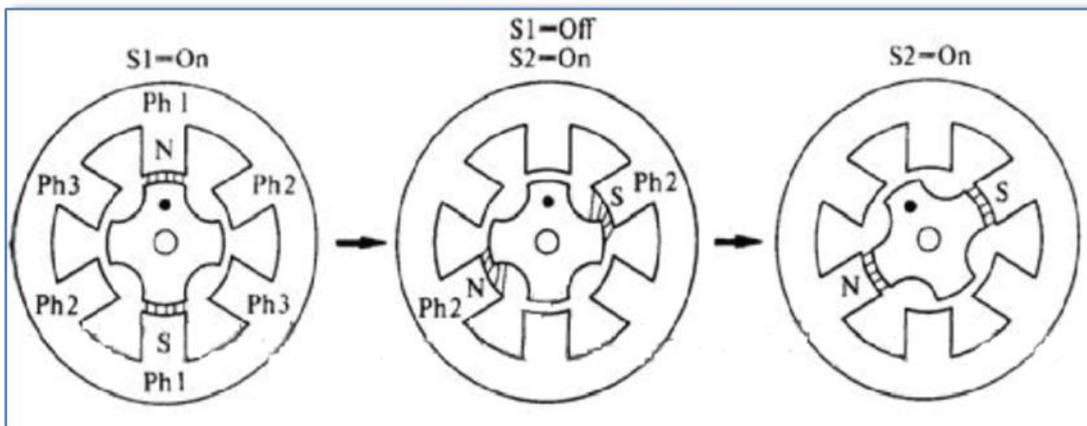
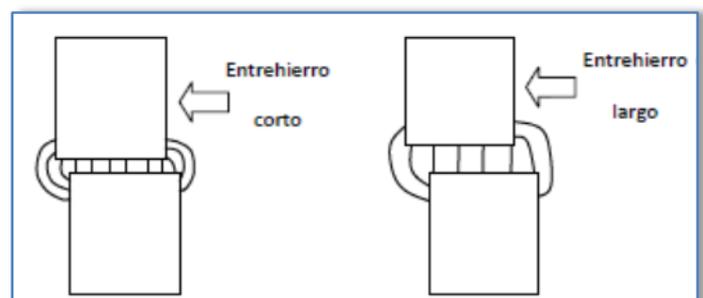


Figura 2.6- El rotor se desplaza un paso al cambiar la excitación de la Fase I a la Fase II

El entrehierro debe ser tan pequeño como sea posible para producir pares grandes a partir de

pequeños volúmenes de rotor y poder alcanzar gran precisión en el posicionado. La figura 2.7 muestra dos entre-hierros diferentes. Para el mismo valor de fuerza magnetomotriz un entrehierro pequeño proporcionará mayor flujo magnético, lo que se traduce en un par mayor.

Figura 2.7- Comparación de las líneas de flujo para dos entrehierros diferentes



Está claro que el desplazamiento a partir de la posición de equilibrio cuando se aplica un par externo es más pequeño cuanto menor sea el entrehierro. Bajo otro punto de vista y olvidándonos de la figura

## Manual de introducción al uso de microcontroladores PIC ARDUINO

2.7 es posible también afirmar la necesidad de que el entrehierro sea el menor posible. El citado entrehierro es también el lugar donde se almacena prácticamente toda la energía en un circuito magnético. Pero en los motores paso a paso no deseamos que la energía suministrada por la fuente de potencia se almacene en ningún entrehierro sino que lo que se pretende es convertir la mayor parte de ella en trabajo mecánico de movimiento del rotor. Por tanto, toda la energía que se almacene en el entrehierro es energía de pérdidas que hay que minimizar construyendo entrehierros con el menor espesor posible. Actualmente los entrehierros van desde los 30 hasta las 100µm.

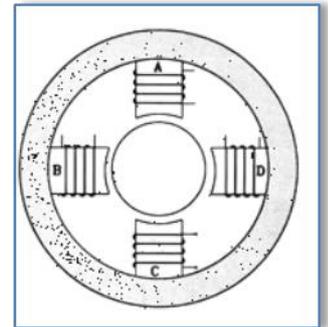
Para disminuir el ángulo de paso es necesario aumentar los polos del estator y el rotor. En el estator se suelen incluir una serie de dientes en cada polo, todos con la misma polaridad cuando se excita la fase correspondiente, para conseguir ángulos de paso menores.

### Motores de imán permanente

Se denominan así los motores paso a paso que poseen un imán como rotor. Para explicar el funcionamiento de este tipo de motores estudiaremos la figura 2.9 que representa un motor paso a paso de imán permanente de cuatro fases.

.Figura 2.9-Motor paso a paso de imán permanente de cuatro fases

El imán cilíndrico se utiliza como rotor. El estator, por el contrario, está formado por cuatro polos bobinados constituyendo cada uno de ellos una fase diferente. Cuatro interruptores conectando cada fase con una fuente de potencia DC completan el esquema de control del motor. Si las fases se excitan con la secuencia Fase 1-> 2-> 3-> 4 el motor girará en sentido horario girando en cada paso 90º. Para disminuir el ángulo de paso es necesario aumentar los polos del estator y los polos magnéticos del rotor.



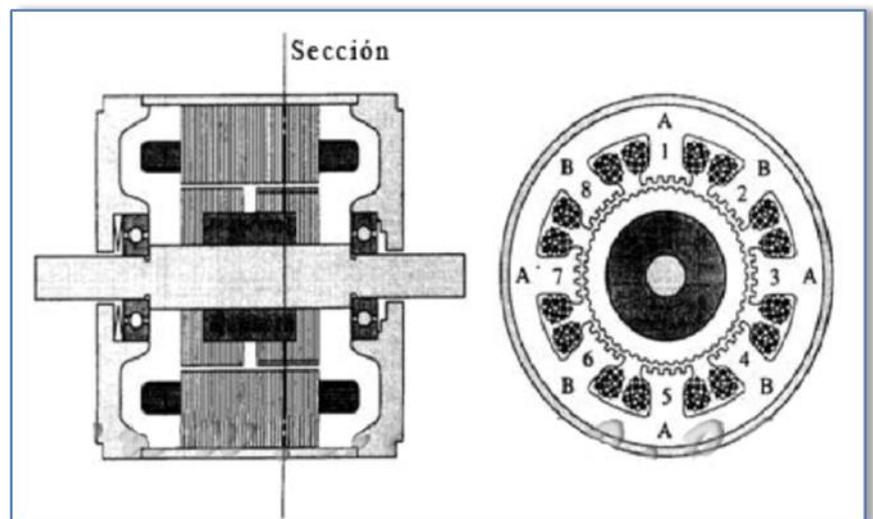
Una característica destacable de este tipo de motores es que el rotor permanece en posiciones fijas aunque se desconecte la fuente de potencia. Estas posiciones coinciden con las posiciones que va alcanzando el motor si es excitado con una secuencia tal que en todos los casos es una sola fase la que está excitada.

Una desventaja importante de este tipo de motores es que la máxima densidad de flujo viene limitada por el magnetismo remanente del rotor.

### Motores híbridos

Este tipo de motores también tiene por rotor un imán permanente. Se le denomina híbrido porque su funcionamiento se basa en los dos tipos de motores explicados anteriormente.

Figura 2.10-Sección de un motor paso a paso híbrido



La figura 2.10 ilustra un motor típico de estas características con cuatro fases. La estructura del estator coincide con la de un motor de reluctancia variable, no así los arrollamientos, ya que en este caso los dientes de los polos pueden corresponder a fases diferentes. En el caso de la figura, las bobinas de dos fases diferentes se arrollan en el mismo polo con lo cual según qué fase esté excitada en cada momento el polo pertenecerá a una fase o a otra.

Otra característica importante es la estructura del rotor

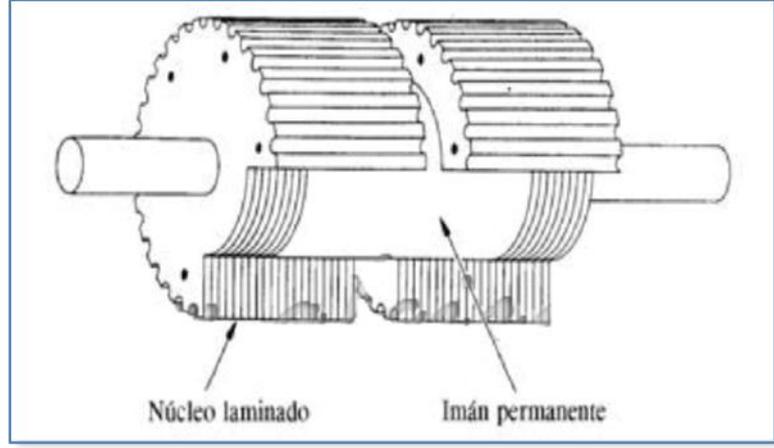


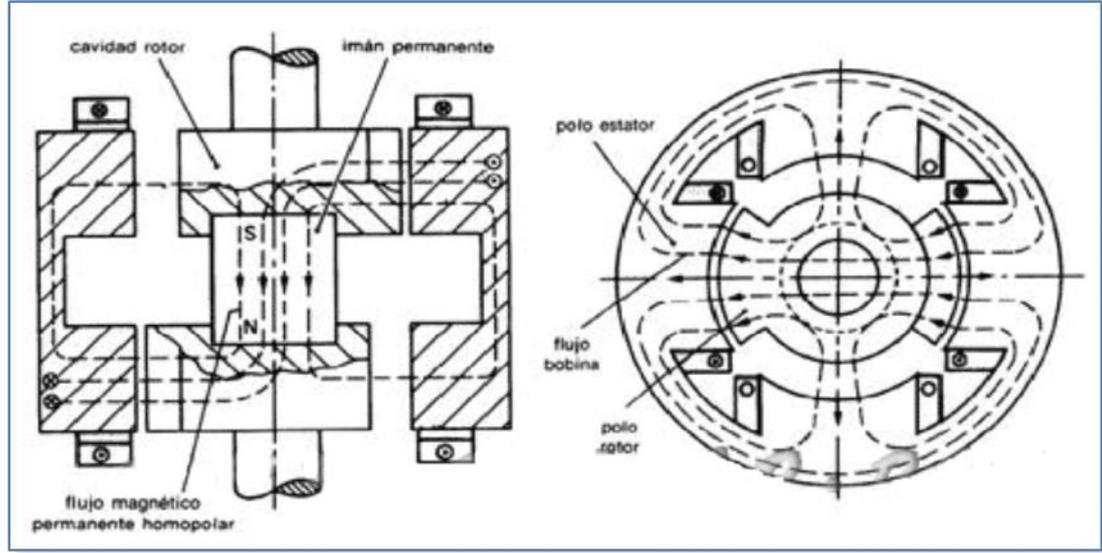
Figura 2.11-Estructura del rotor en un motor híbrido

La figura 2.11 ilustra como un imán permanente de forma cilíndrica se aloja en el núcleo del rotor. Está magnetizado longitudinalmente. Cada polo de este imán está recubierto de una estructura cilíndrica dentada construida generalmente de acero blando. Los dientes de las dos secciones están desalineados medio diente unos respecto otros.

Figura 2.12.a Líneas de flujo producidas por el imán permanente del rotor

El campo magnético generado por las bobinas del estator se representa en la figura 2.12.b.

El funcionamiento para una secuencia de una fase activa lo muestra la figura 2.13. En el estado 1 los polos de la fase



A están excitados, y los dientes del polo 1 atraen a los dientes del rotor del polo norte, mientras que los del polo 3 atraen de igual forma a los dientes del polo sur del rotor. Cuando la corriente ( $i$ ) es conmutada a la fase B (estado 2), el rotor se desplaza un cuarto de espacio de un diente, quedando alineados el polo norte del rotor con el polo 2 del estator y el polo sur del rotor con el polo 4 del estator. De nuevo la corriente ( $-i$ ) se conmuta a la fase A (estado 3) produciéndose un nuevo desplazamiento del rotor en un cuarto de espacio de diente, quedando alineado en sentido opuesto (polo 1 con polo sur y polo 3 con polo norte). Otra conmutación de la corriente ( $-i$ ) en la fase B (estado 4) produce un nuevo desplazamiento y una nueva alineación inversa de los polos de esta fase con el rotor. Retornando al estado 1 ( $i$ ), el rotor ha dado 4 pasos de un cuarto del espacio de un diente.

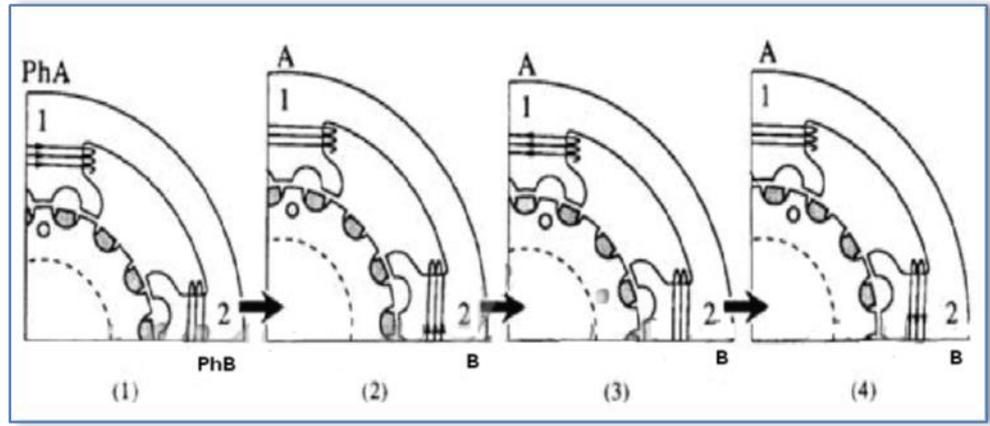


Figura 2.13- Operación de una fase activa de un motor híbrido de dos fases

Motores de imanes permanentes "Claw-Poles"

Con la explicación de los tres tipos de motores

anteriores se tiene ya un conocimiento básico del principio por el que operan la gran mayoría de los motores paso a paso. A la hora de su construcción, estos motores difieren ligeramente del modelo teórico al que

Escuela de Educación Técnico Profesional N° 602  
Taller de electrónica 5º  
Sección Instrumentación y Control  
Manual de introducción al uso de microcontroladores PIC ARDUINO

deberían pertenecer y otro tanto ocurre con su modo de operación. Este es el caso del motor utilizado en este proyecto, que se puede encuadrar dentro de la categoría de motores de imanes permanentes, pero que su especial construcción lleva a dedicarle un apartado exclusivo con el fin de comprender mejor su funcionamiento. La figura 2.14 muestra la sección de uno de estos motores.

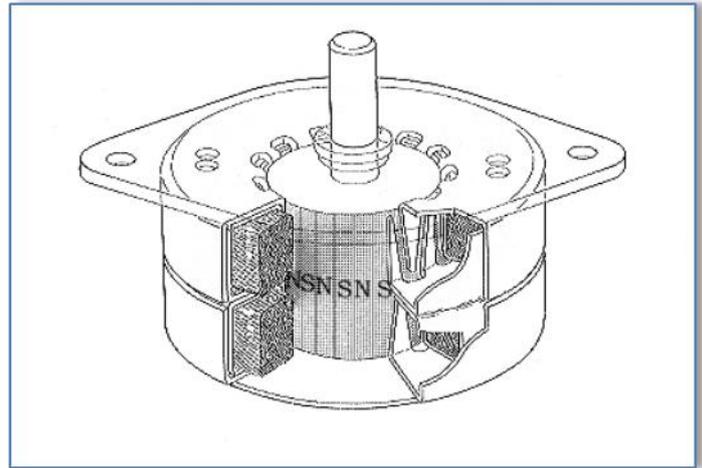


Figura 2.14- Sección de un motor de imán permanente "claw-pole"

Se observa como el estator está formado por dos partes. Cada una de estas partes está formada a su vez por dos estructuras provistas de dientes afilados que se entrelazan. Por el interior de estas estructuras dentadas se sitúan las bobinas necesarias para crear el campo magnético en el estator; en nuestro motor el número de bobinas es dos, una en cada parte del estator. Los dientes entre cada parte del estator están desalineados, una distancia correspondiente a medio diente. En este tipo especial de motores paso a paso, el rotor los constituye un imán permanente magnetizado con polaridad norte y sur tantas veces como pares de dientes entrelazados tiene el estator.

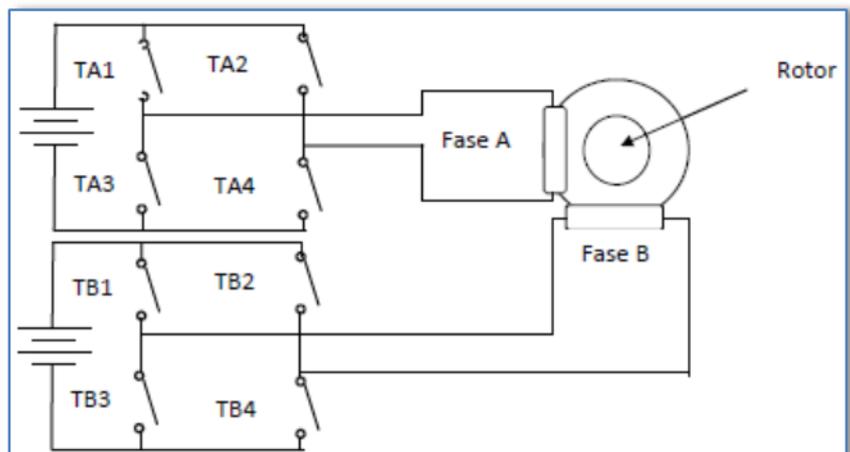
El movimiento se produce por la tensión de Maxwell originada en cada cambio de excitación de las fases debido a las polaridades magnéticas en rotor y estator. En cada paso el motor se desplaza medio diente hacia un sentido u otro, dependiendo del sentido de la corriente por las bobinas. Un motor bifásico con doce pares de dientes entrelazados en cada parte del estator dará cuarenta y ocho pasos por revolución lo que supone un ángulo de paso de 7.5°.

El movimiento se produce por la tensión de Maxwell originada en cada cambio de excitación de las fases debido a las polaridades magnéticas en rotor y estator. En cada paso el motor se desplaza medio diente hacia un sentido u otro, dependiendo del sentido de la corriente por las bobinas. Un motor bifásico con doce pares de dientes entrelazados en cada parte del estator dará cuarenta y ocho pasos por revolución lo que supone un ángulo de paso de 7.5°.

## Modos de excitación

Hasta ahora y con el único objetivo de simplificar las explicaciones, la excitación de los motores paso a paso siempre ha sido la misma. En cada paso del motor solamente una fase estaba excitada. Obviamente esto no tiene por qué ser así siempre. Según el número de fases que tenga el motor, la secuencia de éstas, necesaria para hacerlo girar, varía. Nos centraremos en un motor bifásico bipolar, ya que este tipo de motor es con el que se ha llevado a cabo la realización práctica de este proyecto. El término bipolar hace referencia al hecho de que la corriente por las bobinas de cada fase puede ser bidireccional dependiendo que pareja de interruptores estén abiertos o cerrados. La figura 2.19 servirá para comentar los diferentes modos de excitación de este tipo de motor.

Figura 2.19-esquema base para comentar los modos de excitación



## Modo paso entero

### Fases excitadas alternativamente

En este modo de excitación, en cada secuencia de comunicación solamente una fase está excitada. Para realizar una secuencia completa es necesario realizar cuatro conmutaciones, en cada una de las cuales el motor se desplazará un ángulo de paso. Este modo de excitación suele recibir por esto el nombre de "Secuencia de 4 pasos". La figura 2.20 muestra esta secuencia.

En este modo de excitación, en cada secuencia de comunicación solamente una fase está excitada. Para realizar una secuencia completa es necesario realizar cuatro conmutaciones, en cada una de las cuales el motor se desplazará un ángulo de paso. Este modo de excitación suele recibir por esto el nombre de "Secuencia de 4 pasos". La figura 2.20 muestra esta secuencia.

El término "+" indica que la corriente por la fase circula en un determinado sentido y el término "-" indica que lo hace en sentido contrario. El término "off" indica que no circula corriente alguna por la fase.



Manual de introducción al uso de microcontroladores PIC ARDUINO

**Figura 2.20-Secuencia modo paso entero. Fases excitadas alternativamente**

Paso	Fase A	Fase B
1	-	Off
2	Off	-
3	+	Off
4	Off	+

En este caso los dientes del estator y rotor están alineados para cada paso o posición.

**Fases siempre excitadas**

En este modo de excitación después de cada conmutación siempre resultan estar excitadas las dos fases. La figura 2.21 muestra este modo de excitación. Como en el caso anterior, la secuencia completa se compone de cuatro conmutaciones en cada una de las cuales el motor gira un ángulo de paso.

**Figura 2.21- Secuencia modo paso entero. Fases excitadas simultáneamente**

Paso	Fase A	Fase B
1	-	-
2	+	-
3	+	+
4	-	+

En este caso los dientes de estator y rotor están desalineados medio paso en cada posición de equilibrio con respecto a cada posición de equilibrio alcanzada con el modo de excitación anterior. Esta diferencia es la base fundamental para realizar el modo de excitación que se expondrá a continuación, el modo medio paso.

Existen otras dos diferencias importantes entre estos dos modos de excitación. Una se refiere al par denominado anteriormente holding torque que puede proporcionar el motor. En este caso al estar siempre las dos fases excitadas el par resultante es mayor que en el caso anterior.

La otra diferencia estriba en las oscilaciones que se producen antes de alcanzar cada posición de equilibrio. Sin entrar con mayor profundidad en este tema diremos solamente que las oscilaciones son mucho menores en este caso que en el caso de que excitemos las fases alternativamente como resultado de los diferentes circuitos magnéticos que se producen en cada modo de excitación.

**Modo medio paso**

Como su propio nombre indica, en este modo de excitación el motor se desplaza en cada conmutación la mitad del ángulo de paso. La secuencia de conmutación se basa en combinar las secuencias de los modos de excitación anteriores. La figura 2.22 muestra la secuencia para este modo de funcionamiento que necesita de ocho conmutaciones para completar una secuencia completa.

Paso	Fase A	Fase B
1	+	-
2	Off	-
3	-	-
4	-	Off
5	-	+
6	Off	+
7	+	+
8	+	Off

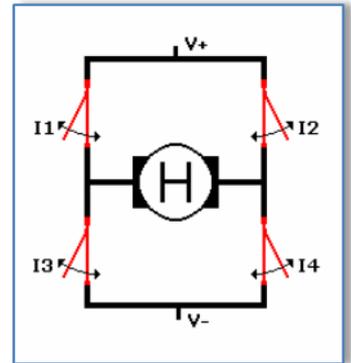
**Figura 2.22-Secuencia modo medio paso**

En las aplicaciones que utilicen este tipo de movimiento hay que tener en cuenta que el holding torque variará para cada paso ya que sólo se excitará una fase para una

posición de paso, pero en el próximo paso se excitan las dos fases. Esto da el efecto de un paso fuerte y otro débil.

## Manejo de motor DC Puente en H

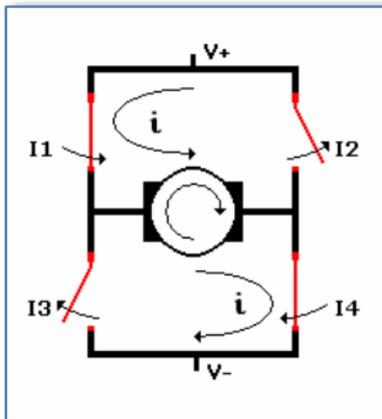
Un puente en H es un mecanismo que nos permitirá variar el sentido de giro a un motor de corriente continua. Estudiaremos un ejemplo basado en llaves para comprender su funcionamiento, aunque cabe aclarar que usualmente se implementan reemplazando las llaves mediante transistores o directamente usando un circuito integrado que suelen contener uno o más puentes H completamente instrumentado con mínimos o ningún componente externo.



### Ejemplos con llaves

El sistema de puente en H de ejemplo se basa en la utilización de interruptores, los cuales los permitirán pasar o cortar la corriente en un determinado sentido.

Para utilizar este sencillo método necesitaremos 4 interruptores como podemos



ver en la imagen de la izquierda. Cada uno de estos interruptores está numerado y puede estar en dos estados, abierto y cerrado. Cuando un interruptor está abierto no permite el paso de corriente a través de él, en cambio cuando está cerrado si lo permitirá.

Variando las posiciones de los interruptores podemos conseguir que el motor gire en un sentido u otro, o que se quede parado al fijar los dos terminales del motor a una misma tensión. También hay que tener en cuenta que no todas las posibilidades son correctas ya que como veremos en la sección de posibles problemas, algunas posiciones crean cortocircuitos.

Como podemos observar en la imagen nuestro puente en H tiene los interruptores I2 e I3 abiertos mientras que I1 e I4 permanecen cerrados.

Esto permite el paso de corriente entre sus dos terminales, pasando de izquierda a derecha. Gracias a este mecanismo hemos conseguido hacer girar el motor en un sentido.

En esta otra imagen se han invertido las posiciones de todos los interruptores, permitiendo la corriente en sentido contrario. De esta forma variando entre estas posiciones podemos conseguir que el motor gire a uno u otro lado.

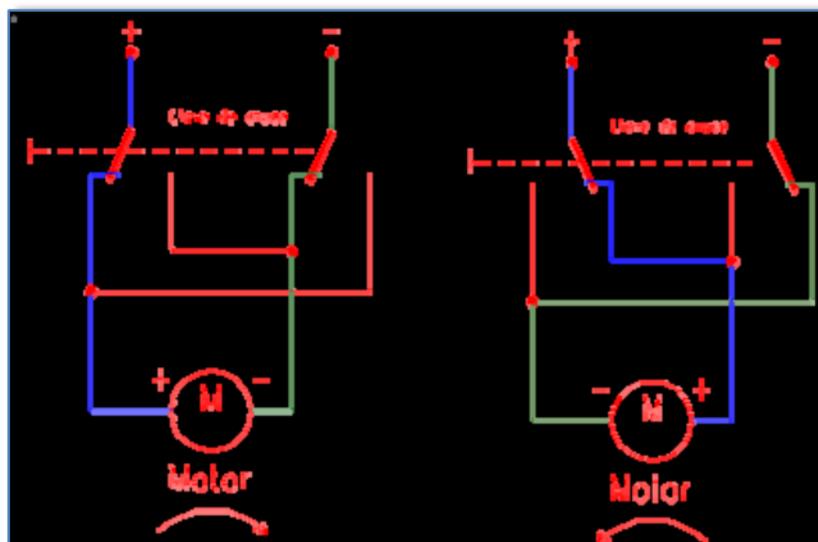
Por último, si pusiésemos los interruptores de uno de los lados (I1, I2) cerrados y los del otro abierto (I3, I4) tendríamos los dos terminales del motor a un mismo voltaje, con lo que el motor se pararía rápidamente, más incluso que si se cortase la alimentación. A este método se le llama "Fast stop".

En la práctica, la forma más simple de implementar la inversión de giro de un motor de CC es mediante una llave doble inversora, según el siguiente esquema.

Si la llave inversora está dotada de posición neutra central, sirve también como interruptor de alimentación del motor.

### El circuito integrado L293 y L293D

Se trata de un integrado que incorpora 4 medios puentes H, que pueden configurarse para conformar dos puentes H completos (inversión de giro en motores de CC) o para controlar hasta 4 motores (o relays)



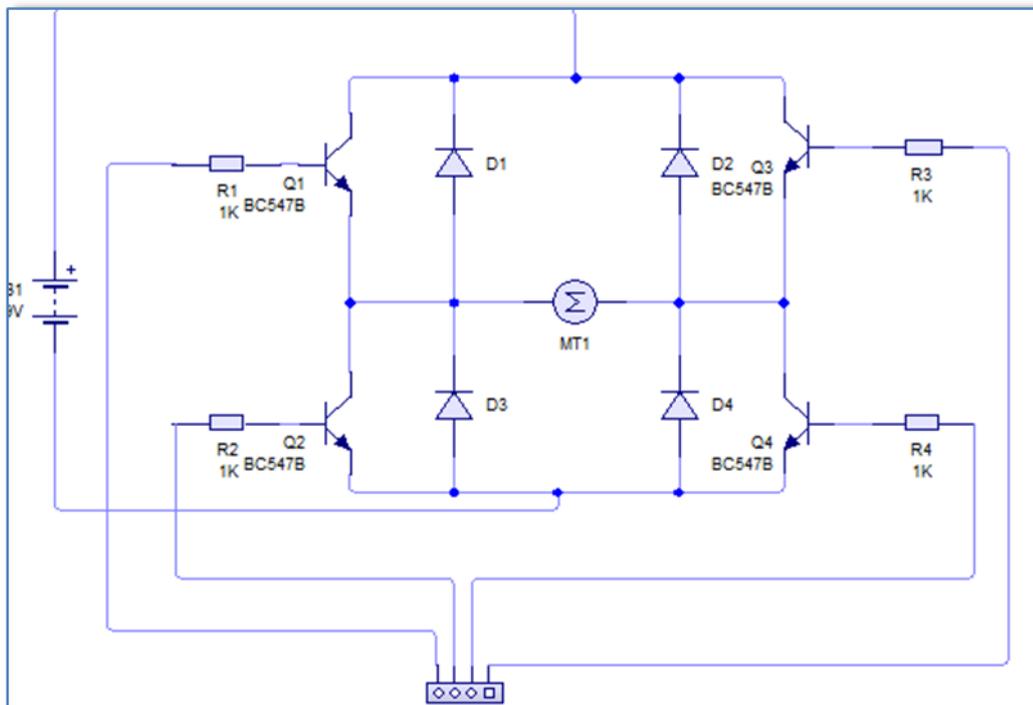
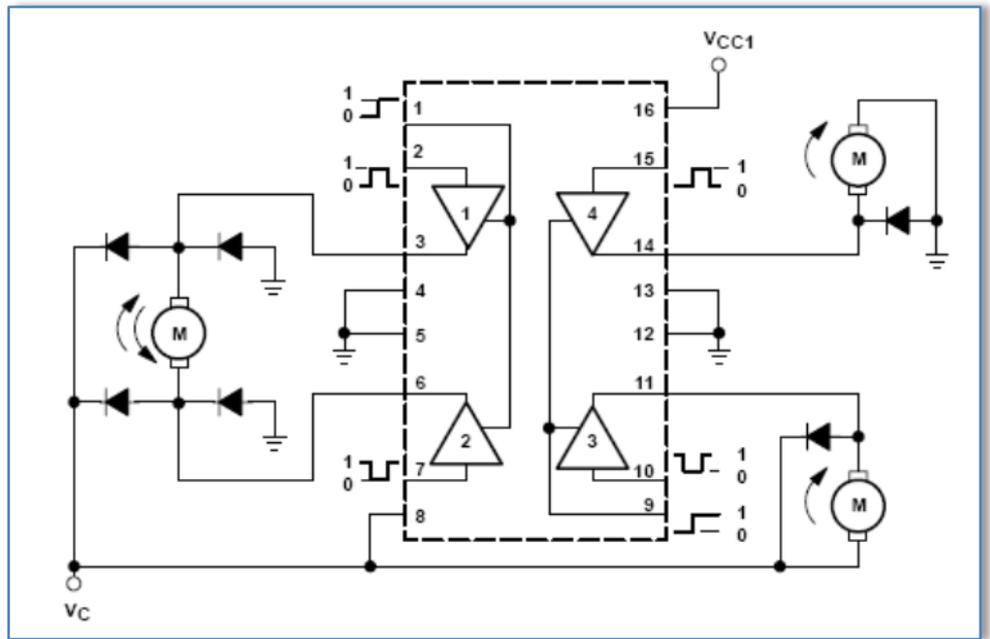
Manual de introducción al uso de microcontroladores PIC ARDUINO independientes en modo on-off.

En la imagen vemos de un lado el uso de 2 medios puentes para configurar un puente H completo (lado izquierdo) y el uso de los otros dos en forma independientes para encender apagar otros dos motores (lado derecho)

**Puente h con transistor NPN**

El Puente H formado de transistores NPN se utiliza para cambiar de giro un motor de Corriente Directa(CD), a diferencia de los relés los transistores

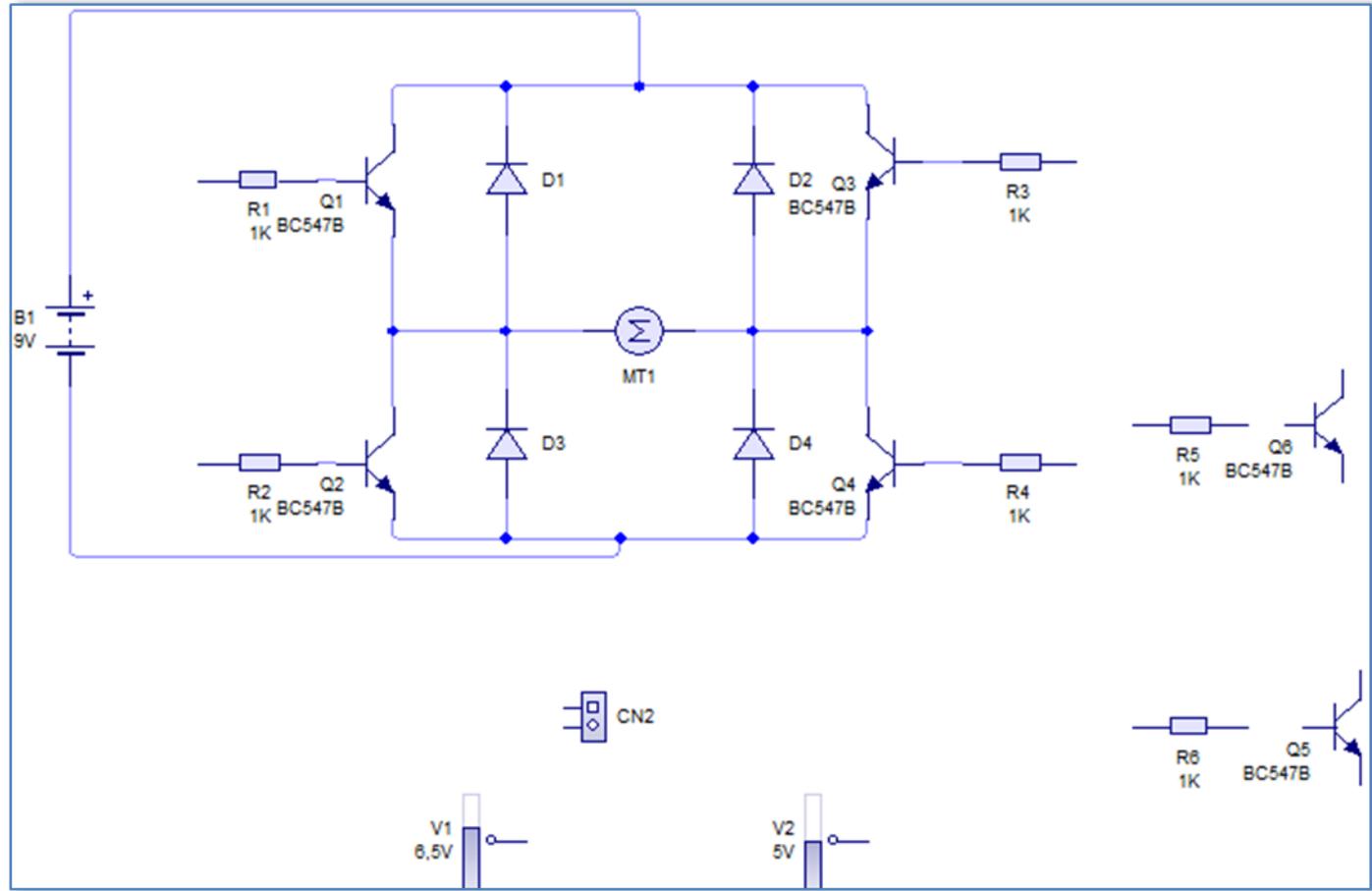
tienen un tiempo de reacción mucho mejor por no tener partes mecánicas que hagan contacto al aplicarles una corriente eléctrica, su funcionamiento es sencillo simplemente activamos la base de los transistores y estos dejan pasar la corriente, si aplicamos tensión en el Q1 y Q4 el motor dará el giro hacia la izquierda, por el contrario al aplicar tensión en Q2 y Q3 el motor dará el giro hacia la derecha.





## Puente h NPN simplificado

Queda en el lector simplificar el puente h para usar 2bits en lugar de cuatro bits, con los siguientes componentes:

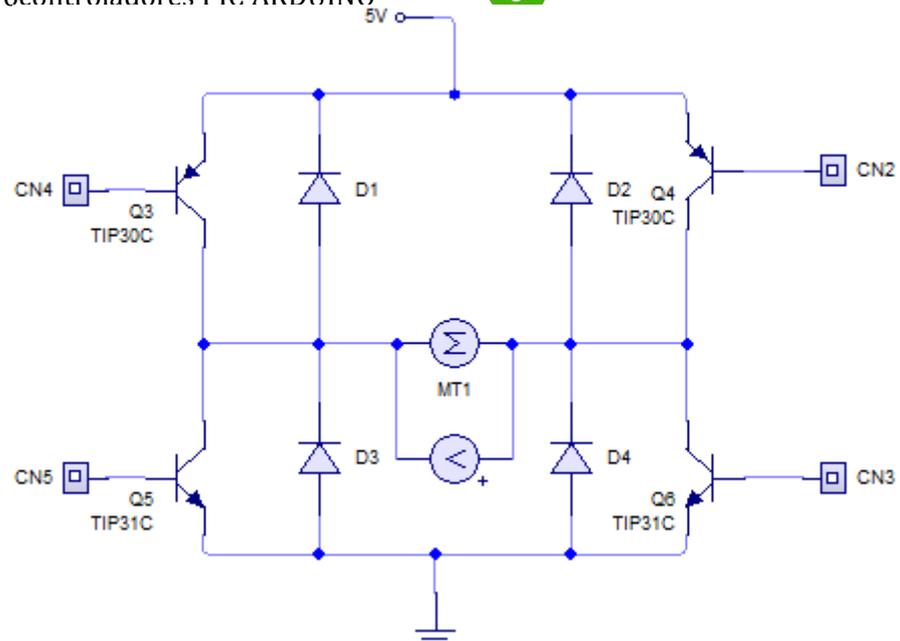


## Manual de introducción al uso de microcontroladores PIC ARDUINO Puente h con transistor NPN y PNP

En este caso el puente H consta de transistores NPN y PNP (par complementario).

Notar que si colocamos "1" en las bases de Q3 Q5 y "0" en Q4 Q6, se establece un sentido de circulación de corriente  $I_L$  como la indicada en la figura. Mientras que si colocamos "0" en las bases de Q3 Q5 y "1" en Q4 Q6, se establece un sentido de circulación de corriente  $I_L$  contrario.

Nuevamente podemos controlar el sentido de giro del motor M.

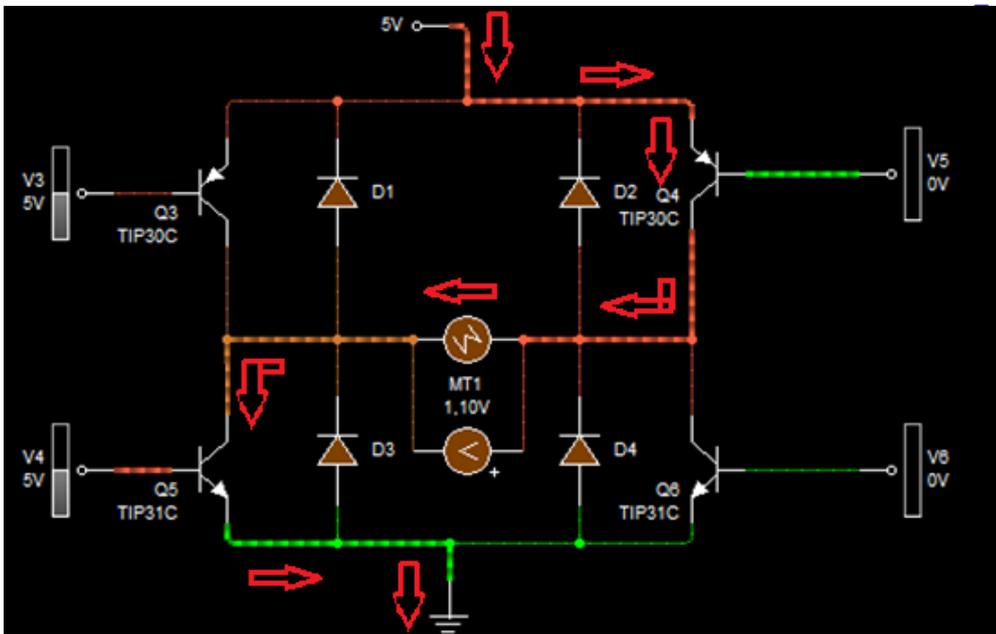


Típicamente  $Q3=Q4$  y  $Q5=Q6$ . Ej: TIP41 y TIP42 ; TIP30 Y TIP 31; BC337 y BC327 , etc.

$V_L$  (tensión de trabajo del motor) e  $I_L$  son datos.

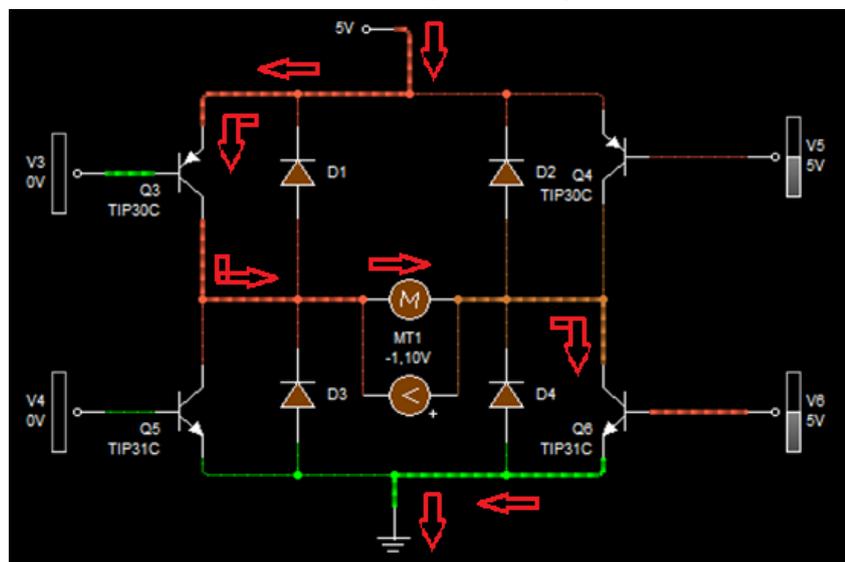
Para lograr nuestro objetivo elegiremos un motor cuya  $V_L$  sea inferior a  $V_{CC}$ (en este caso 5vdc), por lo tanto los TBJ podrán trabajar en la zona activa, y en ellos caerá la diferencia de tensiones entre  $V_{CC}$  y  $V_L$ .

Preferentemente convendrá que la  $V_{CE}$  y  $V_{EC}$  de los TBJ



sean lo más bajas posibles, asegurando de este modo la menor disipación de potencia. (Recuerde que  $P_T = I_C \cdot V_{CE}$ ). Sería ideal que trabajen en saturación.

Como los "1" y "0" los deberá asignar un circuito digital seguramente tendremos que agregar transistores adicionales al circuito para manejar las corrientes de bases de los Q3 Q4 Q5 Q6.







## Circuitos para programar los pic

### Circuito programador PIP02

Para la programación de los PICs es necesario el uso de un circuito programador. A través de este, se memoriza en el microcontrolador el programa de control que se desee.

Este modelo fue diseñado por Jose Manuel Garcia. La elección de este se debe a la realización de la tarea de programación de una manera sencilla y de bajo costo.

Dicho programador se conecta al PC de sobremesa a través del puerto serie (en los portátiles que no lo posean se puede usar un adaptador USB a puerto serie), y funciona con una cantidad nada despreciable de PICs distintos (entre ellos el PIC 16F873), todo ello utilizando software de programación estándar, como icprog1.4 o mplab, ponyprog, etc.

Se ha probado con éxito al programar los siguientes PICs:

16F627, 16F628, 16C84, 16F83, 16F84, 16F873, 16F874, 16F876, 16F877, 18F242, 18F252, 18F258, 18F442, 18F452 Y 18F458.

Y, al menos en teoría, debería funcionar correctamente con los siguientes:

16C62, 16C63, 16C64, 16C65, 16C66, 16C67, 16C71, 16C72, 16C73, 16C74, 16C75, 16C76, 16C77, 16C715, 16C620, 16C621, 16C622, 16C623, 16C624, 16C625, 16F870, 16F871, 16F872, 16F872, 16C923 Y 16C924.

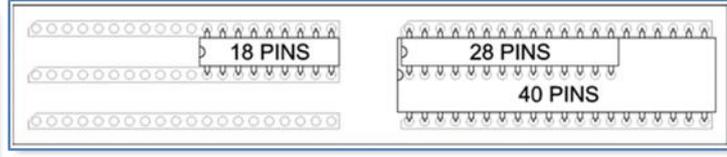
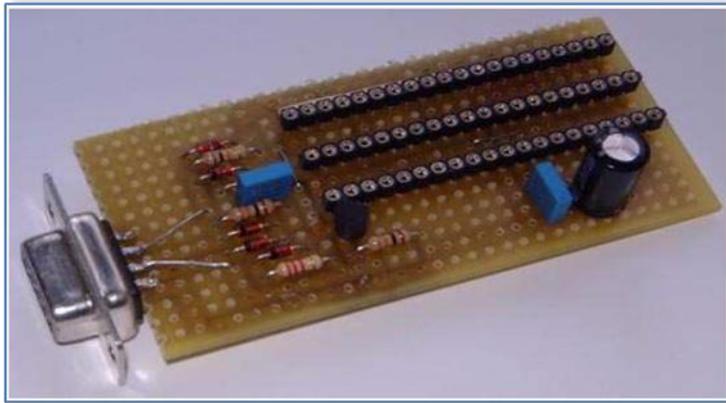
Teniendo en cuenta el costo de tiempo y dinero que se requiere la construcción de este programador, no se le puede pedir más.

Lista de componentes:

Cantidad	Referencia	Descripción
1	R2	2k2
3	R1,R3,R4	10K
1	C2	Capacitor electrolítico 1000uf 25V
2	C1,C2	Cap cerámico 100nF
4	D1,D2,D3,D4	Diodo 1N4148 o similar
1	D5	ZENER 12V
2	D6,D7	ZENER 5V
1	T1	BC547
1	CON1	Conector BD9 hembra
3	Z1,Z2,Z3	Tira zocalo 20 pines

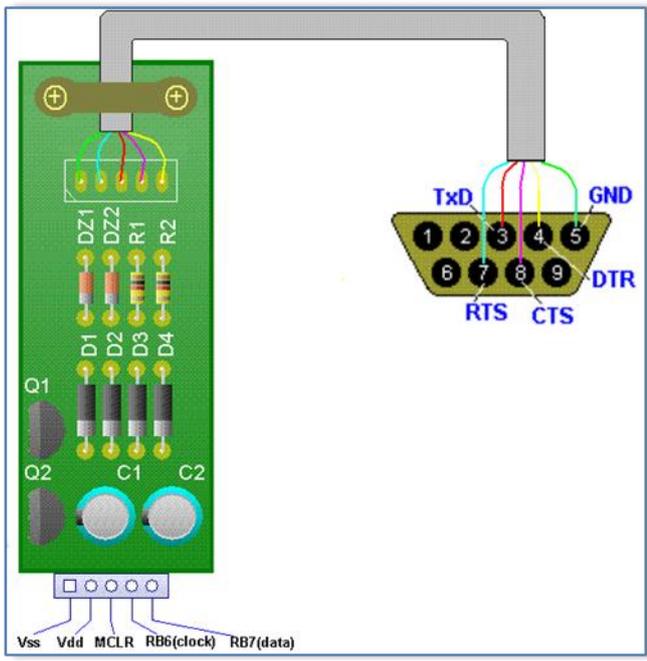
Sin entrar en detalle sobre el funcionamiento del circuito, decir solo que se basa en principios muy parecidos a los JDM2, pero con ciertos retoques en la temporización y la estabilidad de las señales. En este caso, se utiliza como alimentación de +5V el condensador C2, que se carga mediante D2, D3 Y D4 en los momentos en que las señales DTR, RTS Y TXD del puerto serie con negativas, y su tensión queda estabilizada mediante D7. D5 fija la tensión de programación en 12V y D6 limita la tensión de la señal CLOCK a +5V.





Ejemplo de inserción de pic para programar

Versión para no tener que conectar en pic al programador



Para conectar el programador al PC hará falta también un cable serie DB9 transparente (cableado pin a pin) que tenga cableados al menos los pines que se indican en los diagramas anteriores. Este cable se puede fabricar o comprarlo ya hecho (es el tipo de cable que se utiliza para conectar un modem al PC).

### Utilización

Lo primero que hay que tener en claro a la hora de utilizar este programador o cualquier otro es el orden en que se deben hacer las cosas para no estropear ni el programador, ni el PIC, ni el puerto serie del PC o el adaptador.

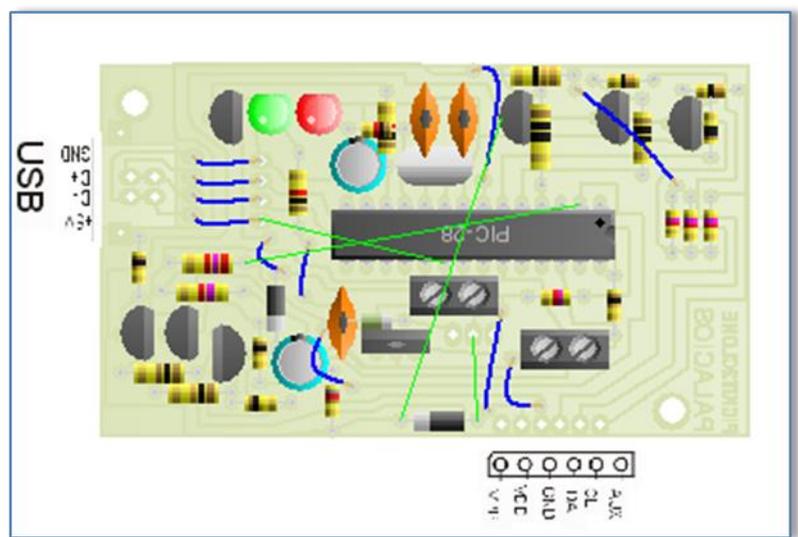
Siempre que debemos insertar o extraer el PIC del zócalo hay que desconectar el programador de puerto serie, ya que, al extraer la alimentación del puerto serie, mientras esté conectado estará alimentado. Por tanto, el proceso a seguir consta de los siguientes pasos:

- 1) Con el programador desconectado insertar el pic en el zócalo en la posición correcta.
- 2) Conectar el programador al cable que viene del puerto serie del pc.
- 3) Llevar a cabo las operaciones de grabación o lectura necesarias.
- 4) Desconectar el programador del cable que viene del puerto serie del pc.
- 5) Extraer el PIC del zócalo.

Existen una gran variedad de circuitos para programadores PIC y también programadores usb que son los más necesitados ya que el puerto serie pronto desaparecerá en las computadoras portátiles al menos.

### Programador pickit 2 clone

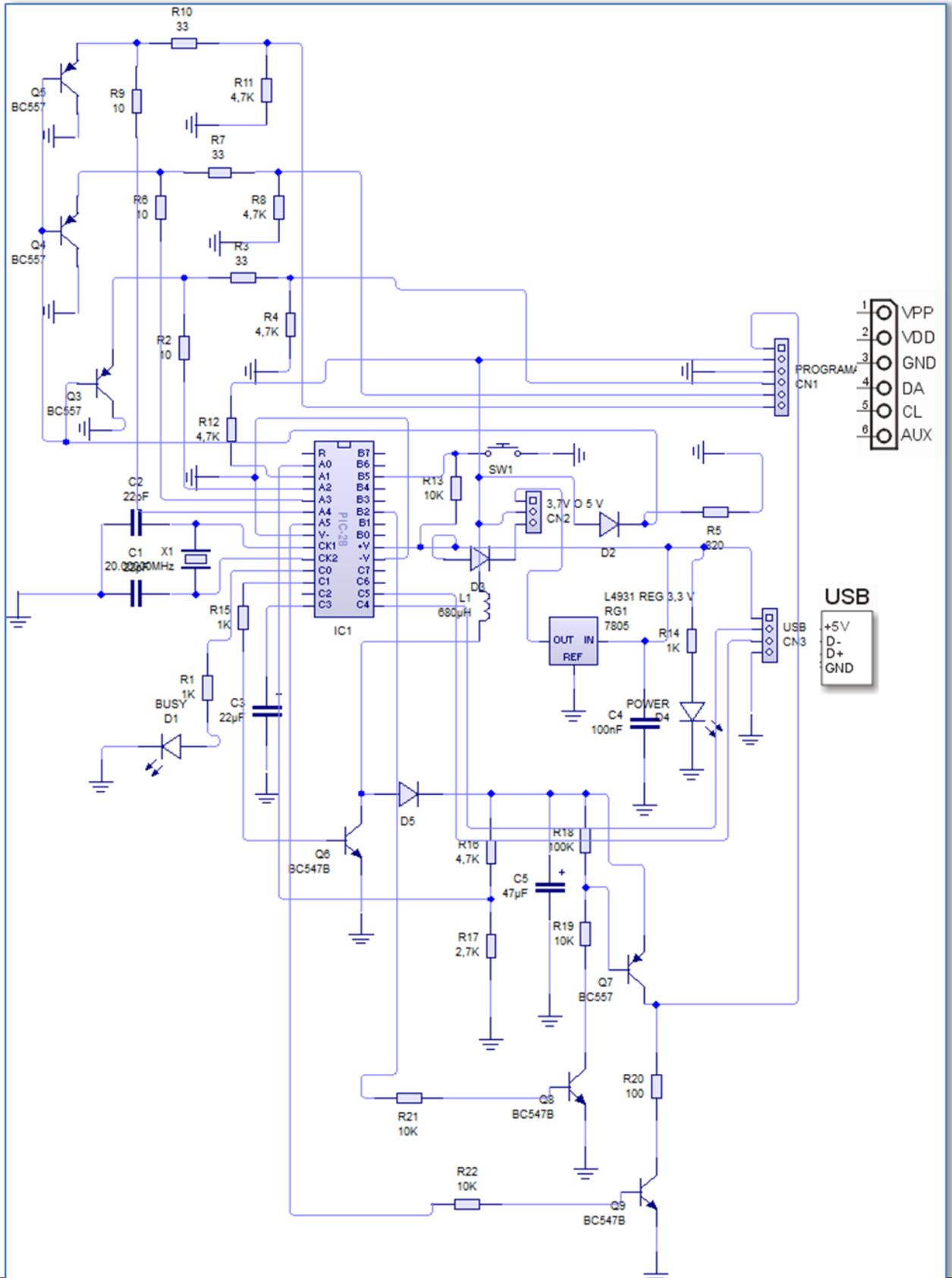
El siguiente programador de microcontroladores Pics que he construido siguiendo el modelo del PicKit2® de la Microchip® para poder trabajar también con micros de 3,3V. En mi caso he optado por usar un regulador específico en lugar del sistema con componentes discretos.



Manual de introducción al uso de microcontroladores PIC ARDUINO

Generalmente, los proyectos son completamente desarrollados por mí pero en este caso he hecho una excepción porque hay demasiadas zonas oscuras en torno al PicKit2® y me llevaría demasiado tiempo analizar la electrónica, el firmware y el software de los sistemas originales. Teniendo en cuenta que el circuito está libre de sufrir modificaciones.

El programador que les presento es formalmente el famoso PicKit2® en su versión Clone, compatible al 100% (eso espero) con el modelo de la Microchip® pero con un circuito bastante simplificado.

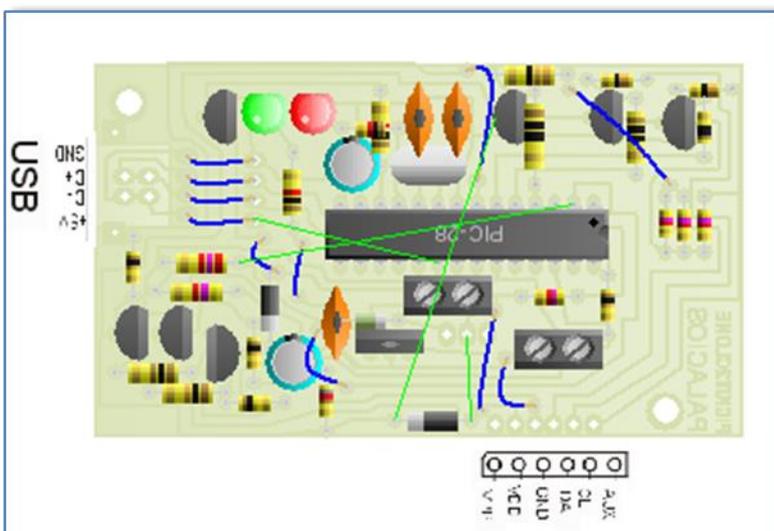
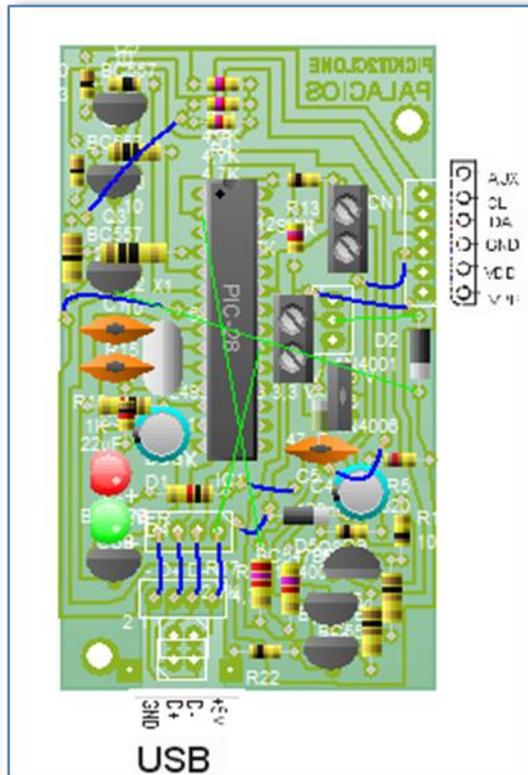
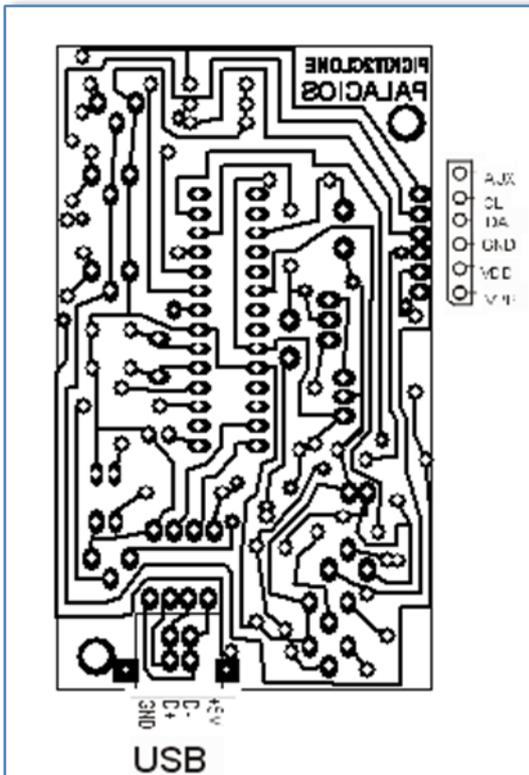


Manual de introducción al uso de microcontroladores PIC ARDUINO

Antes de seguir, quiero hacer una aclaración: este programador que publico hasta ahora ha funcionado perfectamente con distintos modelos de Pícs que he programado. De cualquier manera no puedo garantizar que funcione bien en todos los casos y tampoco dar un soporte técnico completo porque hay demasiadas variables en juego a nivel de software y de hardware y tendría que dedicar todo mi tiempo libre solo a esto.

Personalmente pienso que las herramientas de trabajo deben ser de buena calidad y considero un programador de pícs como una herramienta de trabajo. Por esto, personalmente prefiero usar programadores de marca y usar modelos clonados o modificados como este, solo en situaciones de emergencia o cuando no tenemos la posibilidad de comprar uno original.

Dimensiones de la plaqueta 50mm x 85mm





## Manual de introducción al uso de microcontroladores PIC ARDUINO

Respecto a otros programadores que se encuentran en la red, haber construido un programador compatible con los productos Microchip® me permite de usar como plataforma de desarrollo los software gratuitos de la Microchip como por ejemplo el PicKit2® software y fundamentalmente la suite MPLAB X IDE® (v 1.60) para Linux.

La cosa importante de esta experiencia es que funciona!!

Como decía antes, no he podido probar el programador con todos los modelos de pics, solamente algunos, pero parece ser que el sistema es bastante robusto y no he encontrado errores en fase de programación o de reconocimiento del programador por parte del software PicKit2 y MPLAB.

### Lista de materiales

#### Resistencias:

- 3 de 10 ohms 1/8 Watt
- 3 de 33 ohms 1/8 Watt
- 1 de 100 ohms 1/8 Watt
- 1 de 820 ohms 1/8 Watt
- 3 de 1 K ohms 1/8 Watt
- 1 de 2,7K ohms 1/8 Watt
- 4 de 4,7K ohms 1/8 Watt
- 4 de 10K ohms 1/8 Watt
- 1 de 100K ohms 1/8 Watt

#### Capacitores:

- 2 de 100nF
- 2 de 22pF (disco)
- 1 de 22uF 16V (electrolítico)
- 1 de 47uF 25V (electrolítico)

#### Diodos:

- 2 1N5817
- 1 1N4148
- 1 led verde 3mm
- 1 led rojo 3mm

#### Transistores:

- 4 BC557 o similar

- 3 BC547 o similar

#### Circuitos integrados

- 1 PIC18F2550 (programado con firmware Microchip PicKit2)
- 1 L4931CZ33 (u otro regulador 3,3V 100mA de tres patitas)

#### Otros componentes

- 1 bobina de 680uHy
- 1 pulsador para circuito impreso
- 1 zócalo de 28 pins
- 1 cristal 20MHz
- 1 conector USB tipo b hembra para impreso
- 3 conectores a 6 pin paso 2,54 mm para circuito impreso
- 3 conectores a 6 pin paso 2,54 mm para cablepreso
- 3 Circuitos impresos (1 principal y 2 con los zócalos ZIF)
- 1 cable USB tipo impresora ( tipo a y tipo b)
- tornillos, separadores y tuercas

NOTA: si se modifica el tipo de conector USB deberá modificarse la placa.

Construir un programador de PICS es todo un proyecto y no lo aconsejo a la gente que recién empieza. Más allá de la construcción electrónica propiamente dicha en la que se necesitan algunos componentes especiales, es necesario disponer de otro programador que permita de programar el PIC18F2550, corazón de todos los modelos USB como este.

La última versión de firmware para el PIC18F2550 y el programa PicKit2 lo pueden descargar directamente desde el sitio de la Microchip totalmente gratis.

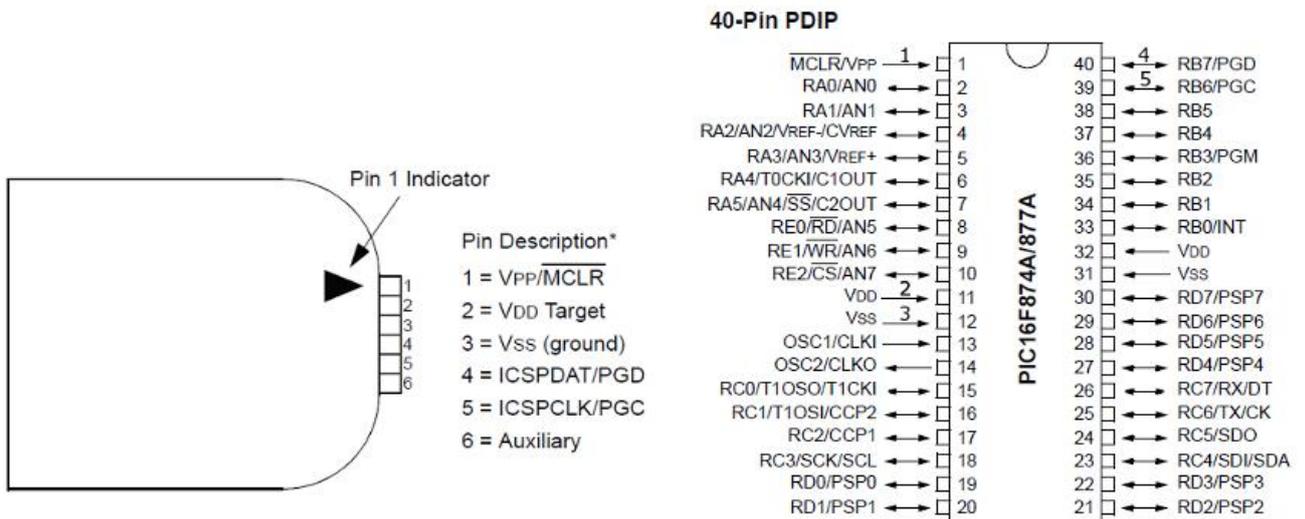
## Utilización

Lo primero que hay que tener en claro a la hora de utilizar este programador o cualquier otro es el orden en que se deben hacer las cosas para no estropear ni el programador, ni el PIC, ni el puerto USB del PC o el adaptador. Teniendo en cuenta que es un clon de pick kit 2, el manual de usuario de este deberá servir a la perfección.

Siempre que debemos insertar o extraer el PIC hay que desconectar el programador. Por tanto, el proceso a seguir consta de los siguientes pasos:

- 1) Con el programador desconectado conectar el pic teniendo en cuenta la conexión necesaria.
- 2) Conectar el programador al cable que viene del USB del pc.
- 3) Llevar a cabo las operaciones de grabación o lectura necesarias.
- 4) Desconectar el programador del cable que viene del puerto serie del pc.
- 5) Desconectar el PIC del programador.

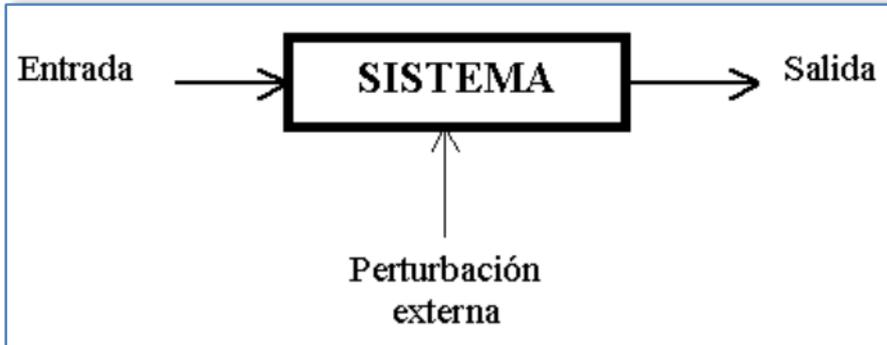
### Ejemplo de conexión del programador para un pic



## Introducción a los sistemas de control

### Definiciones básicas

**Sistema:** es la combinación de componentes que actúan conjuntamente y cumplen un determinado objetivo.



**Variable de entrada:** es una variable del sistema tal que una modificación de su magnitud o condición puede alterar el estado del sistema.

**Variable de salida:** es una variable del sistema cuya magnitud o condición se mide.

**Perturbación:** es una señal que tiende a afectar el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se la denomina interna, mientras que una perturbación externa se genera fuera del sistema y constituye una entrada.

### Ejemplos de sistema

**Variables de entrada Variables de salida Perturbaciones**

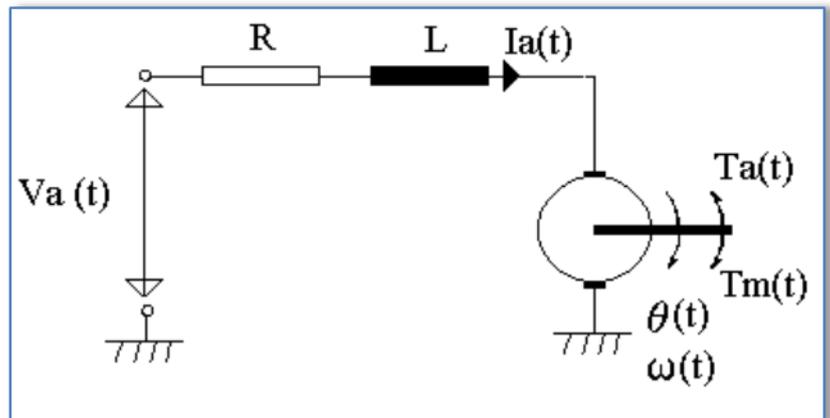
#### Motor DC

**Variables de entrada:** Tensión de armadura

**Variable de salida:** Velocidad de giro del eje

Posición del eje

¿Intensidad de armadura?



**Perturbaciones:**

Externa: par de carga

Interna: temperatura dentro del motor

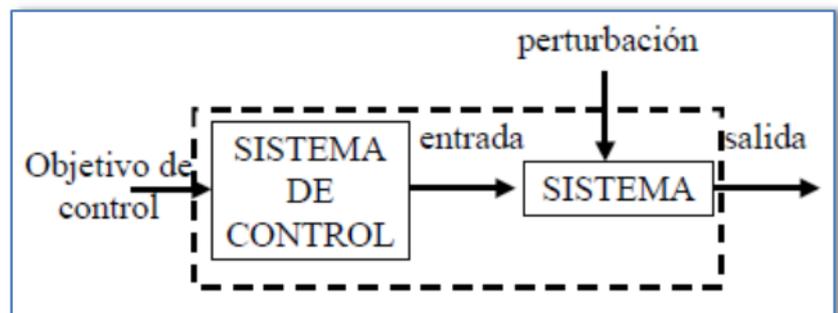
**Una habitación en la que se dispone de un calentador eléctrico que se puede encender o apagar, de un termómetro para medir la temperatura y de una puerta que puede estar abierta o cerrada**

**Variables de entrada:** Interruptor del calentador eléctrico

**Variables de salida:** Temperatura en la habitación

**Perturbaciones:**

**Externas:**



- Temperatura del exterior de la habitación



- Estado de la puerta



**Interna:** actividad de las personas dentro de la habitación

## Sistemas de control

- **Definición:** Sistema de control es el conjunto de dispositivos que actúan juntos para lograr un objetivo de control.

## Sistemas de control en lazo abierto y en lazo cerrado

- **Ejemplos de sistemas con control:**

Motor DC controlado por armadura

- Objetivo: velocidad o posición deseadas
- Variables de control: voltaje o intensidad de armadura
- Perturbaciones: par de carga

Cualquier planta de proceso continuo tiene muchos sistemas de control

- Control de nivel en depósitos
- Control de flujo en tuberías

Cuerpo humano:

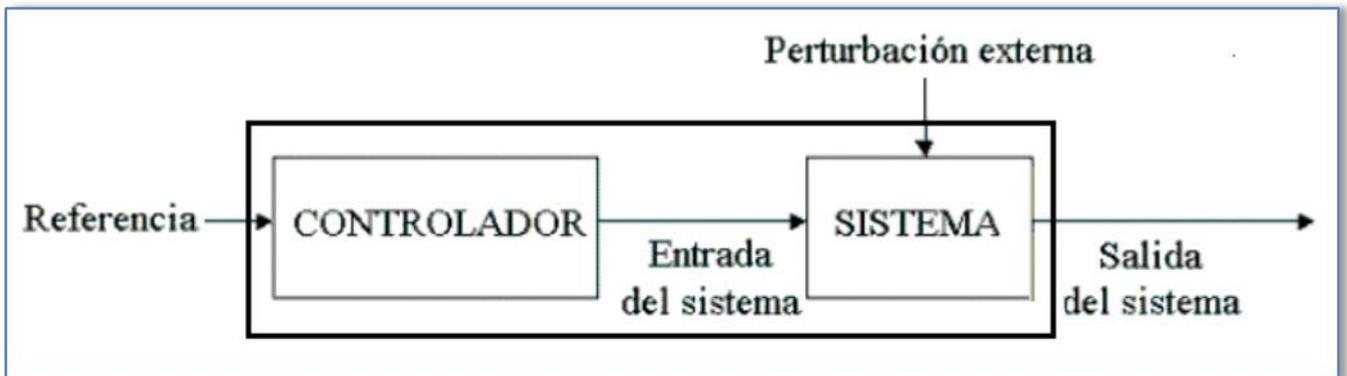
- A nivel microscópico: planta industrial y red de transporte con numerosos sistemas de control
- A nivel macroscópico: control de la temperatura corporal

Sistema económico mundial

- ¿Objetivo?, ¿Variables de control?, ¿Perturbaciones?

- **Clasificación: Lazo abierto o cerrado**

## Sistemas de control en lazo abierto



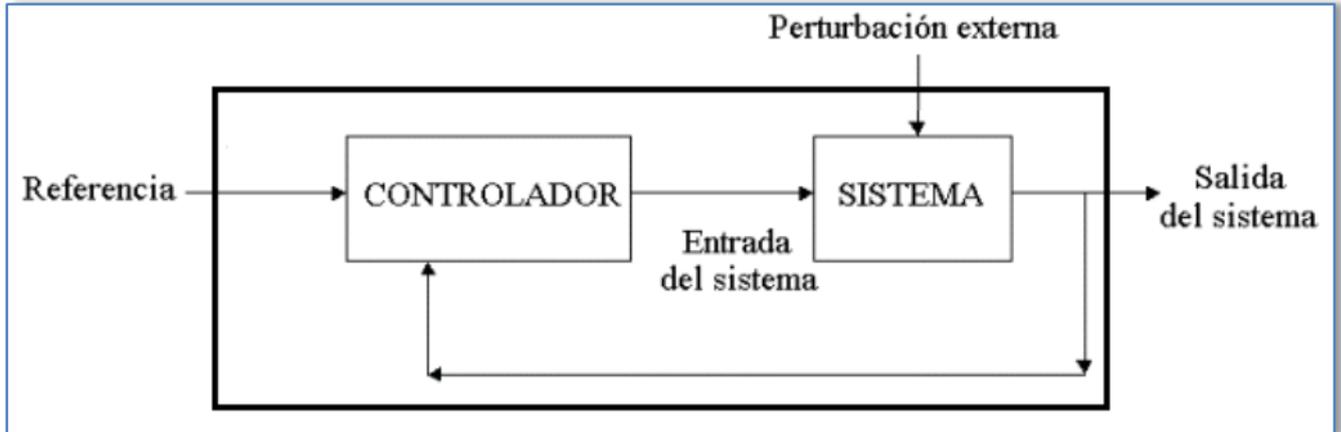
- Aquellos en los que la variable de salida (**variable controlada**) no tiene efecto sobre la acción de control (**variable de control**).
  - Características
    - No se compara la salida del sistema con el valor deseado de la salida del sistema (referencia).
    - Para cada entrada de referencia le corresponde una condición de operación fijada.
    - La exactitud de la salida del sistema depende de la calibración del controlador.
    - En presencia de perturbaciones estos sistemas de control no cumplen su función adecuadamente.
      - Ejemplo:
        - Control en lazo abierto por tensión de armadura de un motor DC de excitación independiente.
        - El control en lazo abierto suele aparecer en dispositivos con **control secuencial**, en el que no hay una regulación de variables sino que se realizan una serie de operaciones de una manera determinada. Esa secuencia de operaciones puede venir impuesta por eventos (event-driven) o por tiempo (timedrive).
- Se programa utilizando PLCs (controladores de lógica programable)
- Ejemplos:
    - Lavadora:
      - Funciona sobre una base de tiempos
      - Variable de salida “limpieza de la ropa” no afecta al funcionamiento de la lavadora.

Manual de introducción al uso de microcontroladores PIC ARDUINO

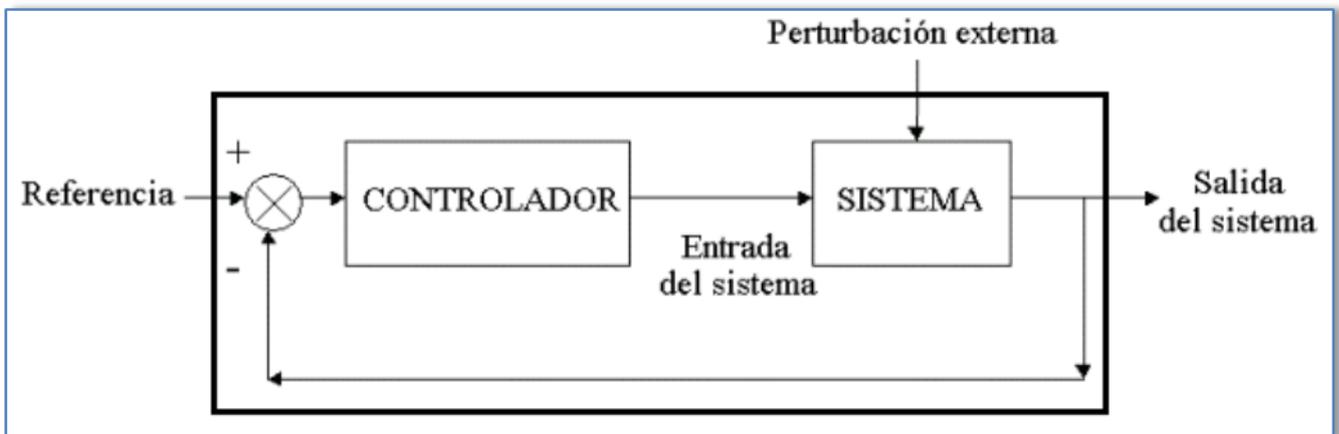
- Semáforos de una ciudad
- Funcionan sobre una base de tiempo
- Variable de salida “estado del tráfico” no afecta el funcionamiento del sistema.

## Sistemas de control en lazo cerrado

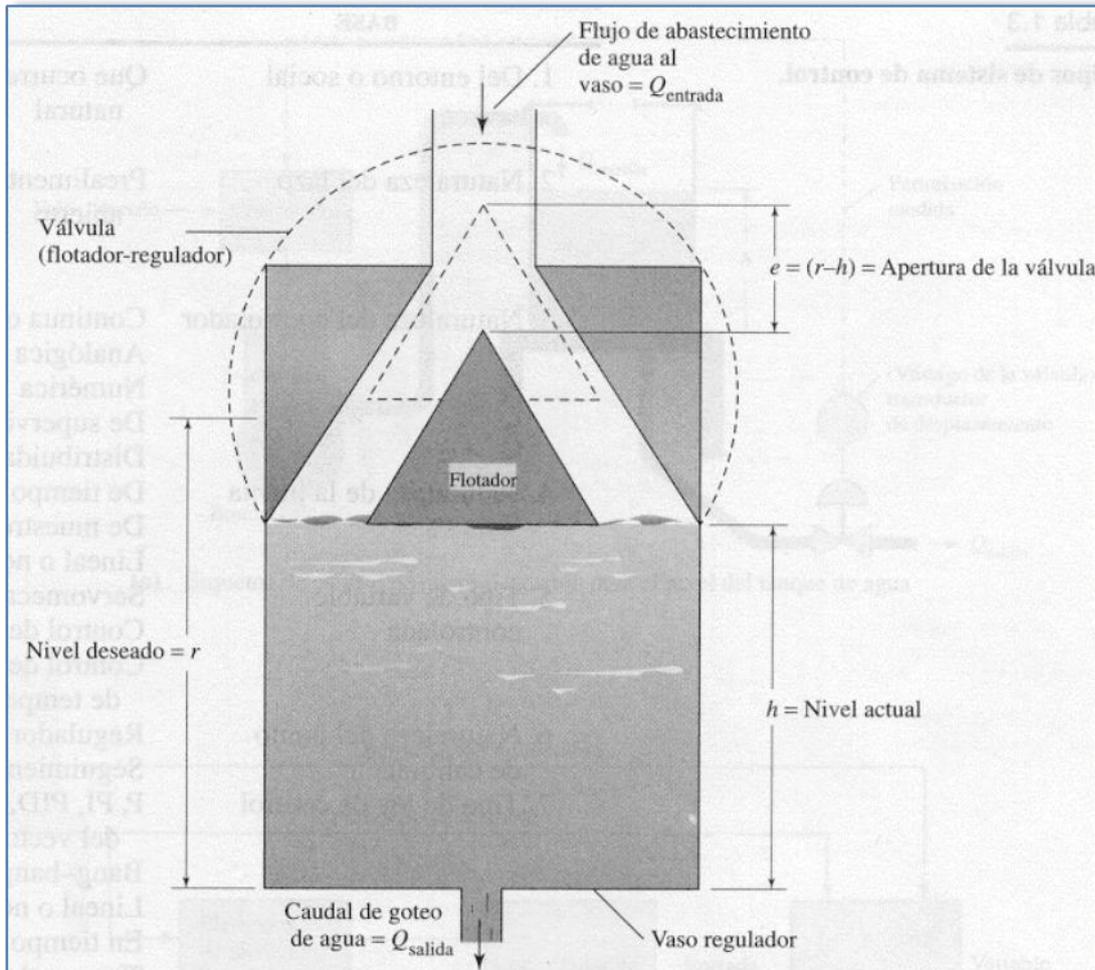
- **Definición: sistema de control en lazo cerrado**
- Aquellos en los que la señal de salida del sistema (**variable controlada**) tiene efecto directo sobre la acción de control (**variable de control**).



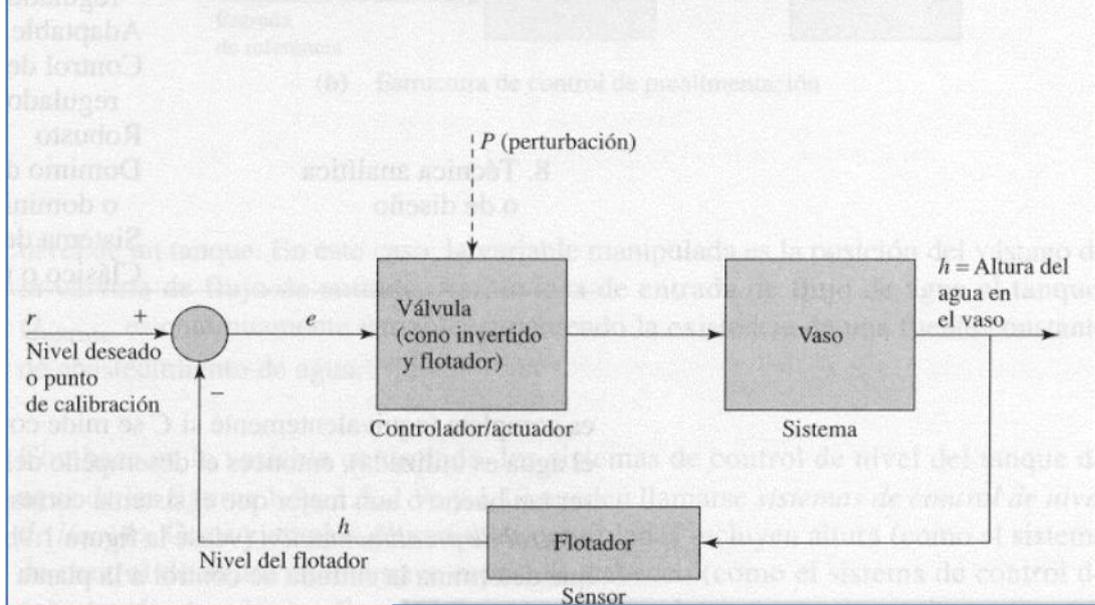
- **Definición: control retroalimentado**
- Operación que en presencia de perturbaciones tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia. Esta reducción se logra manipulando alguna variable de entrada del sistema, siendo la magnitud de dicha variable de entrada función de la diferencia entre la variable de referencia y la salida del sistema.



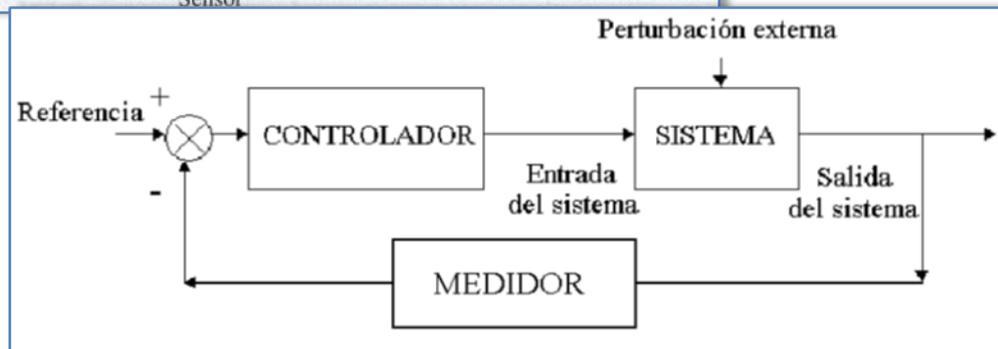
- **Ej: Reloj de agua**
- Probablemente el primer sistema de retroalimentación creado por el hombre.
- Inventado por Ctesibios de Alejandria (Egipto, aprox. 260 a.c.)



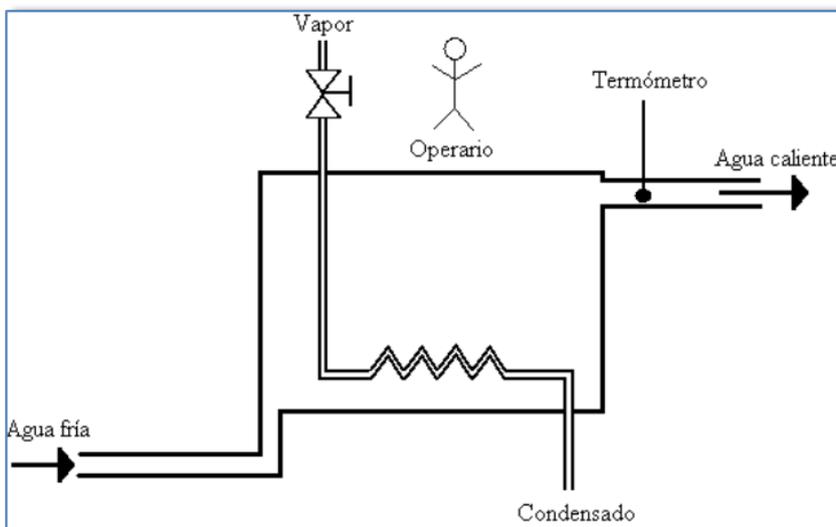
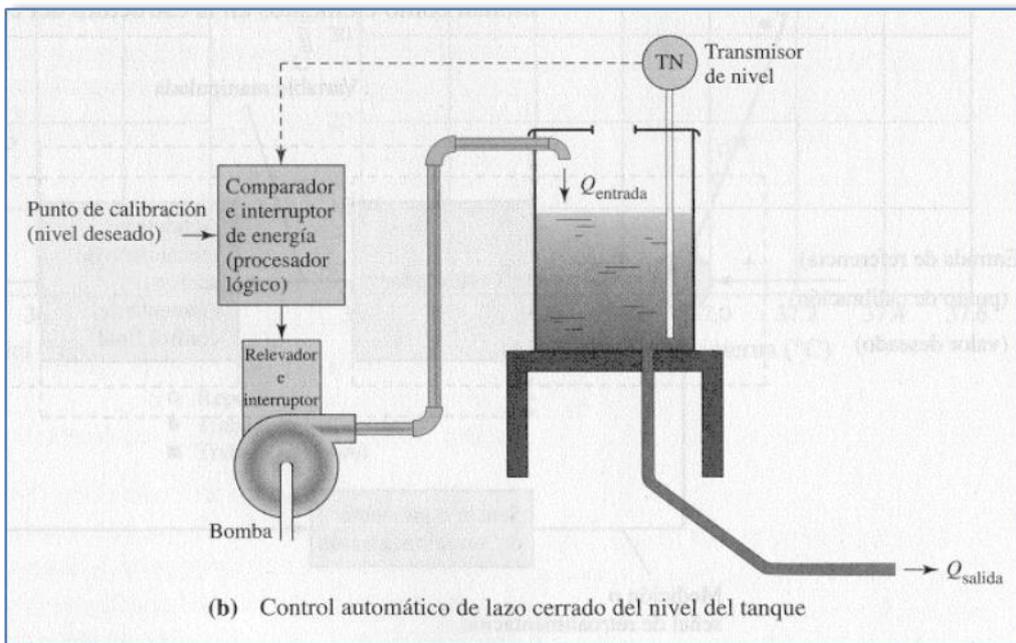
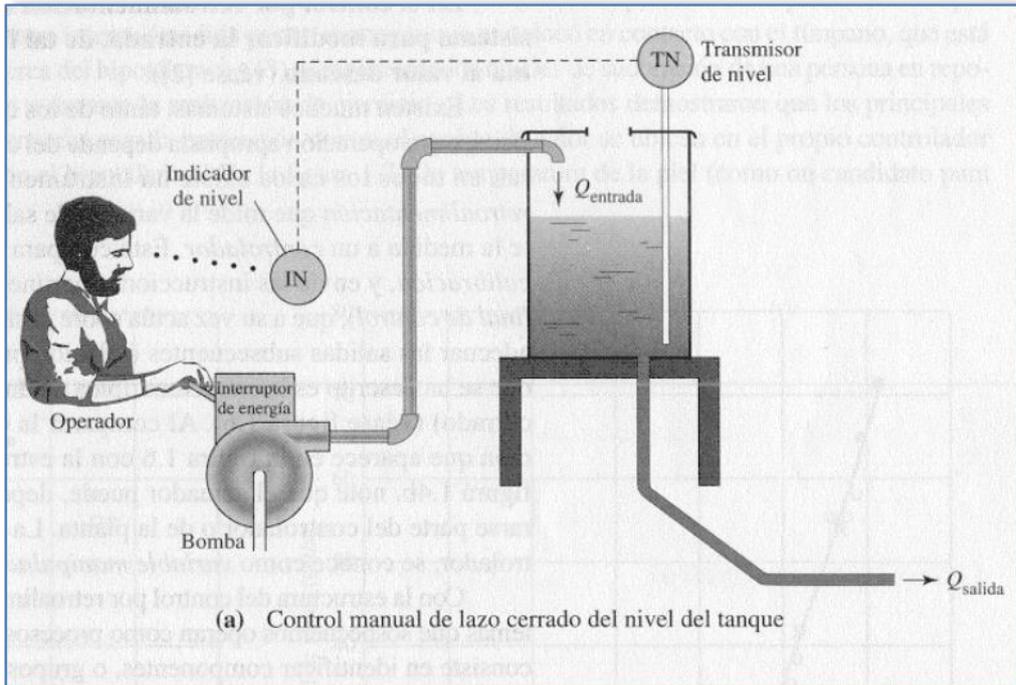
(a) Ilustración de los principales componentes del reloj de agua



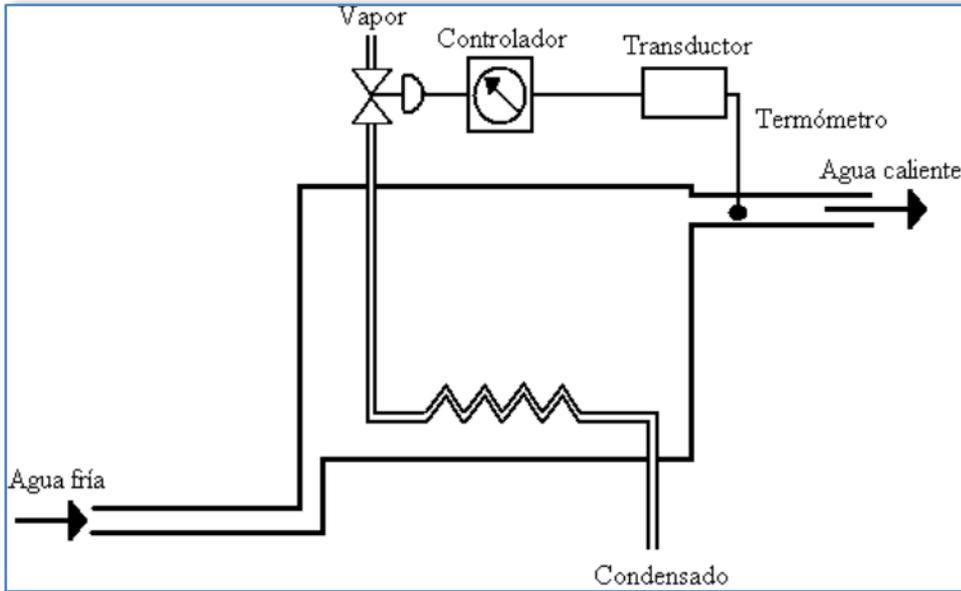
- Clasificación
- Manuales: controlador operador humano
- Automático: controlador dispositivo: Neumático, hidráulico, electrónico o digital (microprocesador)



Manual de introducción al uso de microcontroladores PIC ARDUINO



- Ejemplo
  - Control de temperatura de un intercambiador de calor usando vapor como medio calefactor.
- Conversión de manual a automático
  - Reemplazar el operario por un controlador automático en el que se pueda fijar la señal de referencia.
  - Acoplar un transductor (elemento que transforma un tipo de señal en otra) al elemento que mide la temperatura de forma que la señal de salida del transductor se introduzca al elemento controlador y sea del mismo tipo que la señal de referencia.



– Reemplazar la válvula de vapor manual por una automática y conectar la salida del controlador a la entrada de control de la válvula de regulación.

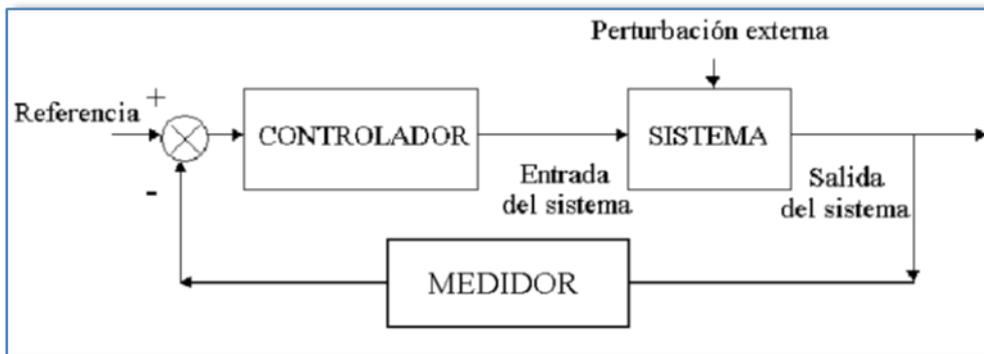
- Elementos de un lazo de control
- Sistema a controlar
- Controlador
- Actuador (puede incluirse en el sistema a controlar)
- Medidor: sensor + transductor

- Funciones de un lazo de control
- Medir el valor de la variable controlada (medida

y transmisión).

– Detectar el error y generar una acción de control (decisión).

– Usar la acción de control para manipular alguna variable en el proceso de modo que tienda a reducir el error (manipulación).



- Ventaja del control en lazo cerrado frente al control en lazo abierto:

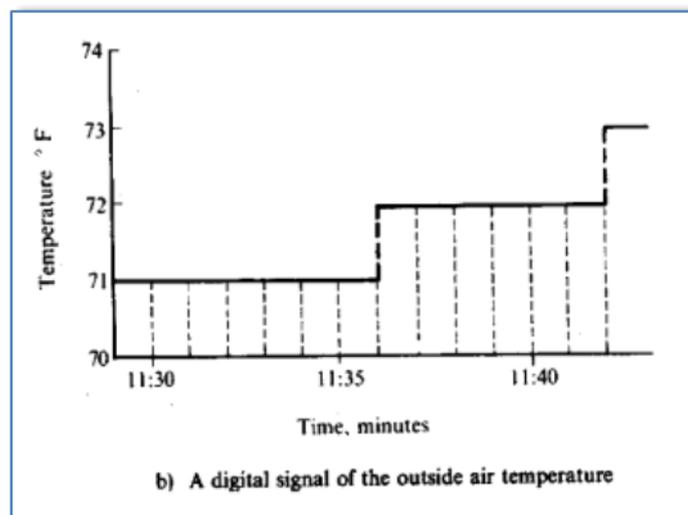
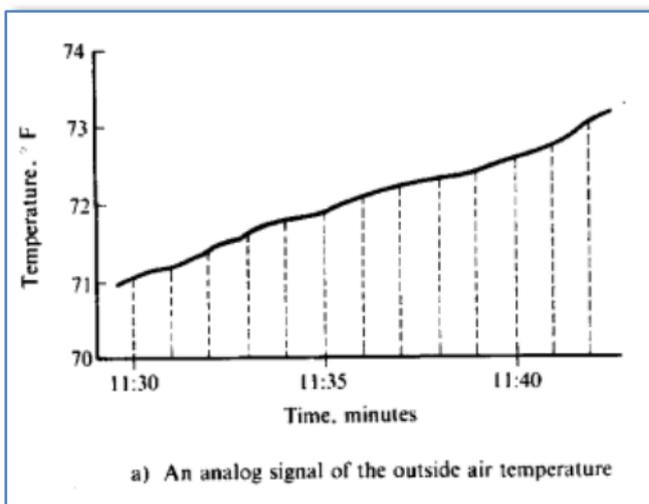
– Respuesta del sistema se hace relativamente insensible a perturbaciones externas y a variaciones internas de los parámetros del sistema.

- Desventaja del control en lazo cerrado:

– Aparece el problema de la estabilidad, ya que si el controlador no está bien ajustado puede tener tendencia a sobre corregir errores, que pueden llegar a producir en la salida del sistema oscilaciones de amplitud creciente llegando a inestabilizar el sistema.

## Tipos de control

- Realimentación de la salida:
  - Lazo abierto y lazo cerrado
    - Comportamiento de la señal de referencia.
- Sistemas seguidores
  - La entrada de referencia cambia de valor frecuentemente
- Ejemplo: servomecanismos (sistemas de control realimentado en el cual la salida es alguna posición, velocidad o aceleración mecánica).
- Ejemplo de servomecanismo: posicionamiento de los cañones de una batería de tiro antiaérea.
- Sistemas de regulación automática
  - La entrada de referencia es o bien constante o bien varía lentamente con el tiempo, y donde la tarea fundamental consiste en mantener la salida en el valor deseado a pesar de las perturbaciones presentes.
- Ejemplos: el sistema de calefacción de una casa, un regulador de voltaje, un regulador de presión de suministro de agua a una comunidad de vecinos
  - Tipo de señal
- Analógicos (continuos)
- Digitales (discretos)



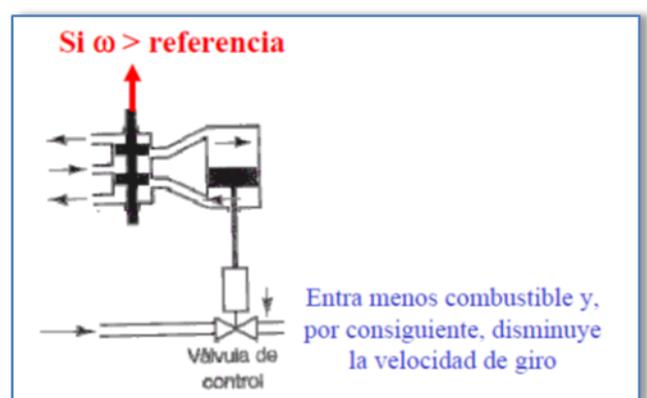
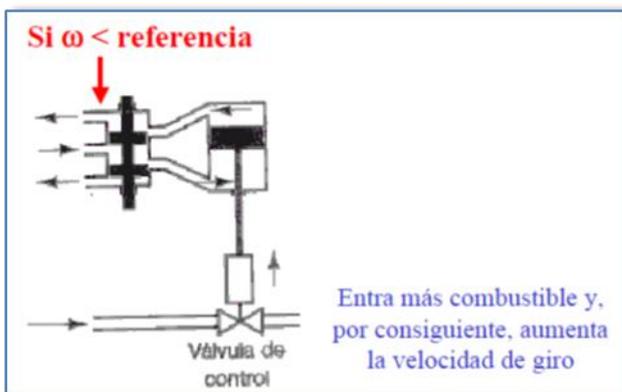
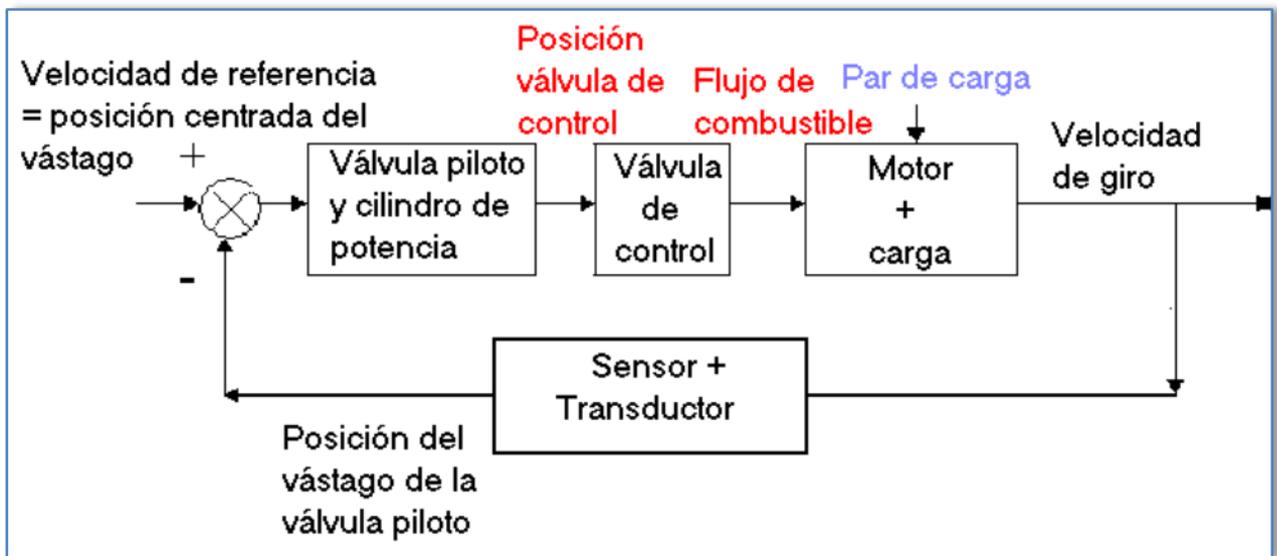
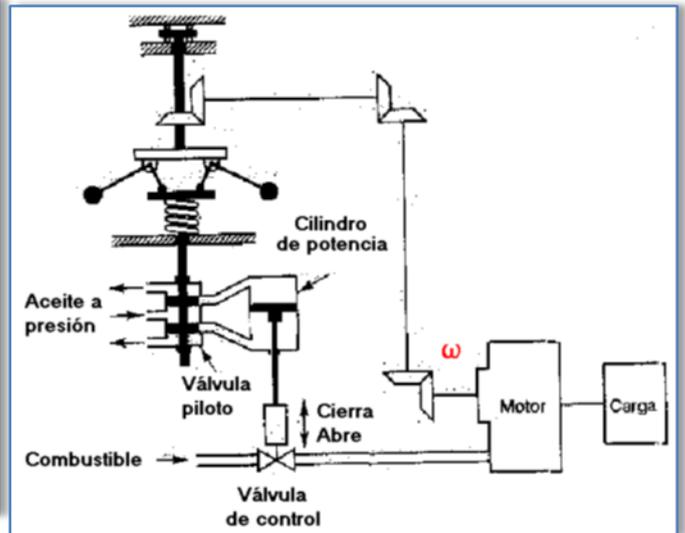
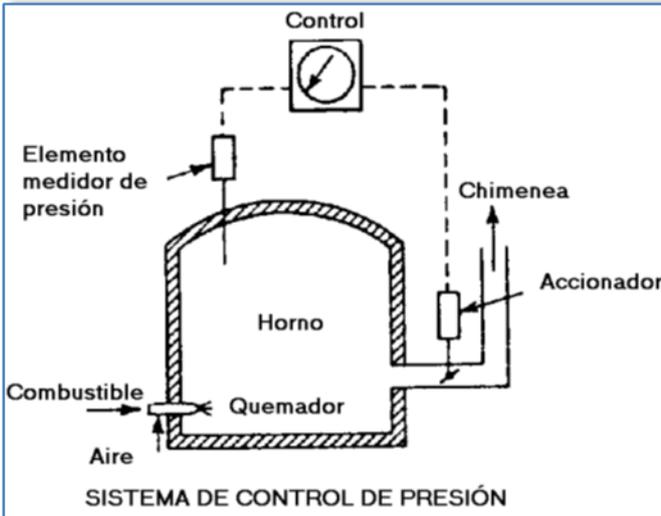
- En función de la industria
  - Control de procesos
    - Los sistemas de control de procesos son aquellos que requieren la regulación de variables de proceso (temperaturas, concentraciones, caudales, niveles ...). Estos sistemas de control requieren la manipulación de unidades de proceso continuas (no se interrumpe el flujo) y discontinuas, batch o por lotes (se interrumpe el flujo).
    - Ejemplos: refinería de petróleo, planta de producción de energía eléctrica, papelera ...
  - Control de máquinas manufactureras
    - Control Numérico
  - Usa un programa para controlar la secuencia de operaciones una máquina, dicho programa contiene instrucciones que especifican posiciones, direcciones, velocidades y velocidad de corte.
    - Control de robots
  - Un manipulador programable diseñado para mover materiales, herramientas en una secuencia determinada para realizar una tarea específica.

## Ejemplos de sistemas de control

### Control de presión

- ¿Cuáles son las variables de salida (controlada), de control, de entrada (referencia) y las perturbaciones externas?
- ¿Diagrama de bloques?

### Regulador centrífugo de Watt para el control de velocidad de una máquina de vapor que fue diseñado en el siglo XVIII

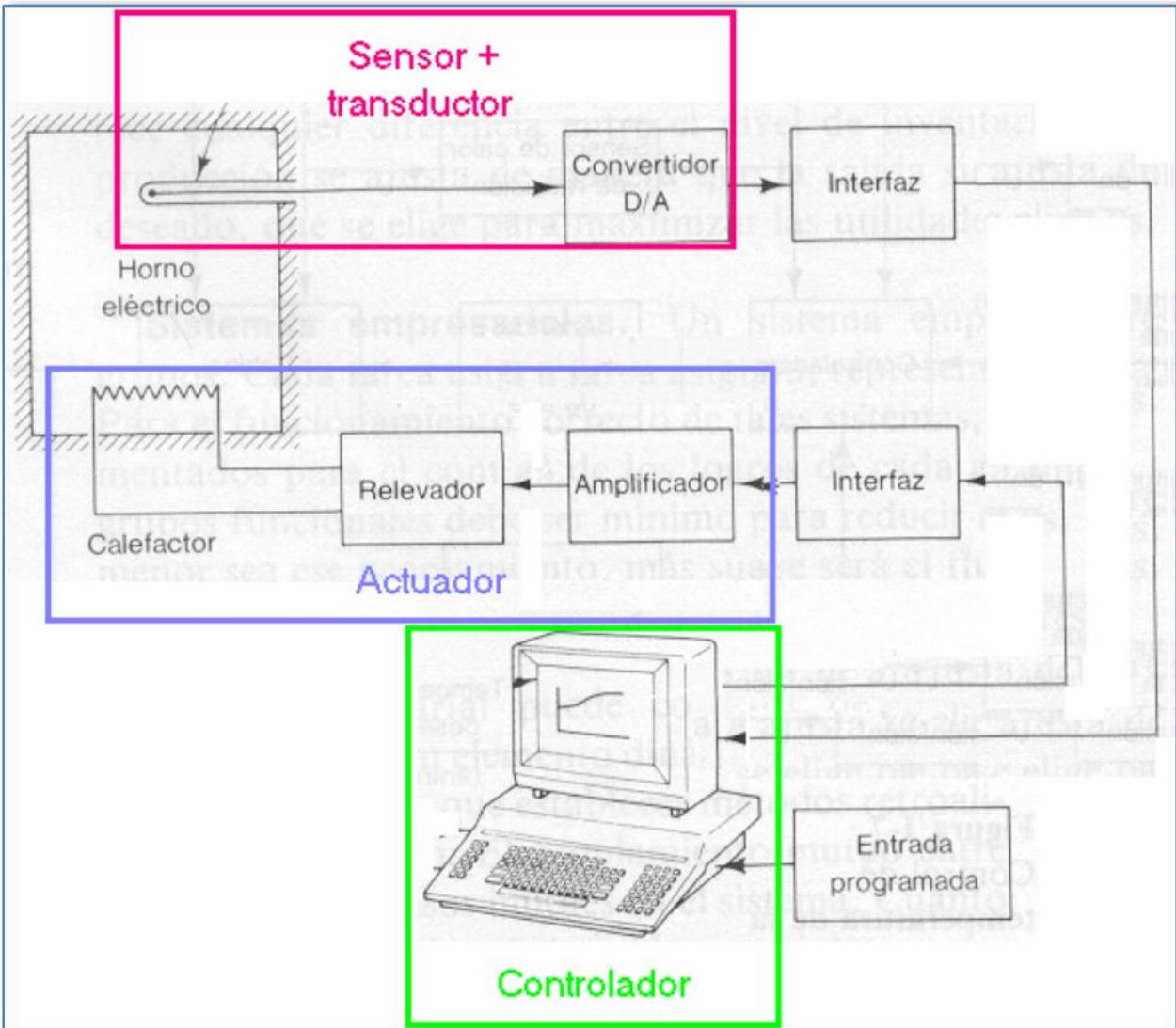


¿Es un sistema seguidor o de regulación automática?

¿Cuáles son las variables de salida (controlada), de control, de entrada (referencia) y las perturbaciones externas?

Sistema de control de temperatura por ordenador Control digital

¿Cuáles son las variables de salida (controlada), de control, de entrada (referencia) y las perturbaciones externas?



## Etapas en la realización de un sistema de control

- **Análisis** del sistema que se quiere controlar ⇒ **objetivos de control**
  - ¿Qué se quiere regular? Variables de referencia.
  - ¿Qué hay que medir? Variables de salida.
  - ¿Qué se puede manipular? Variables de control.
  - ¿Perturbaciones?
  - ¿Situaciones peligrosas?
  - ¿Procedimientos de arranque y parada?
    - Establecer la **estructura de regulación**.
- Especificar que variables se van a realimentar y que variables se van a manipular para lograr los objetivos de control.
  - **Seleccionar, diseñar y sintonizar** los reguladores seleccionados.
- La correcta ejecución de este paso es función de haber establecido antes los **criterios de control**.
- Rechazo de perturbaciones
- Errores estacionarios
- Respuesta dinámica ante cambios en la referencia
- Sensibilidad a cambios de parámetros del sistema
- Un correcto diseño de los reguladores puede requerir el disponer de un modelo lineal del proceso a controlar. Así una etapa previa es el desarrollo de un modelo matemático adecuado para los fines de control.
  - **Evaluar** el diseño del sistema de control, utilizando técnicas de simulación dinámica.
  - **Realización práctica** y puesta a punto del sistema de control diseñado.



## PROGRAMACIÓN DE LOS PIC 16F87X

La posibilidad de programar a esta subfamilia de PIC en serie, permite grabar en la memoria de código el programa de trabajo, estando colocado el Pic sobre el circuito o producto de aplicación final. Esta característica permite a los fabricantes construir y montar completamente la tarjeta de circuito impreso y dejar pendiente la grabación del programa.

La programación en serie típica, que se realiza con un Voltaje Alto de 12 a 14 V aplicado por la pata MCLR#/Vpp requiere el uso de 5 patas del PIC:

- a)  $VDD = 5V$
- b)  $GND$  o Tierra
- c)  $Vpp = 12$  a  $14 V$  que se introducen por la pata MCLR#/Vpp
- d) RB6 : Recibe los impulsos de reloj.
- e) RB7 : Línea de datos con los bits en serie.

Una gran aportación esta gama de PIC es la programación con Voltaje Bajo (LVP : Low Voltage Programming), que no requiere la tensión de 12 a 14V. Para grabar en este modo hay que poner el bit LVP = 1, que reside en la Palabra de Configuración y la pata RB3/PGM se debe conectar a nivel alto. Entonces por la pata MCLR#/Vpp se aplica la tensión VDD de 5V mientras dura la operación de grabado. Cuando no se opera en este modo de programación se puede usar la pata RB3 como una línea de E/S digital.

## Empezando a programar en ANSEMBLER

### Programa fuente:

El programa fuente está compuesto por una sucesión de líneas de programa. Cada línea de programa está compuesta por 4 campos separados por uno o más espacios o tabulaciones. Estos campos son:

**[Etiqueta] Comando [Operando(s)] [;Comentario]**

La etiqueta es opcional. El comando puede ser un mnemónico del conjunto de instrucciones. El operando está asociado al comando, si no hay comando no hay operando, e inclusive algunos comandos no llevan operando. El comentario es opcional para el compilador aunque es buena práctica considerarlo obligatorio para el programador.

La etiqueta, es el campo que empieza en la primera posición de la línea. No se pueden insertar espacios o tabulaciones antes de la etiqueta sino será considerado comando.

Identifica la línea de programa haciendo que el compilador le asigne un valor automáticamente. Si se trata de una línea cuyo comando es una instrucción de programa del microcontrolador, se le asigna el valor de la dirección de memoria correspondiente a dicha instrucción (location counter). En otros casos se le asigna un valor de una constante, o la dirección de una variable, o será el nombre de una macroinstrucción, etc.

El comando puede ser un código mnemónico de instrucción del microcontrolador, o una directiva o pseudoinstrucción para el compilador. En el primer caso será directamente traducido a código de máquina, en el segundo caso será interpretado por el compilador y realizara alguna acción en tiempo de compilación como ser asignar un valor a una etiqueta, etc.

El campo de parámetros puede contener uno o más parámetros separados por comas. Los parámetros dependen de la instrucción o directiva. Pueden ser números o literales que representen constantes o direcciones.

El campo de comentario debe comenzar con un carácter punto y coma. No necesita tener espacios o tabulaciones separándolo del campo anterior, e incluso puede empezar en la primera posición de la línea. El compilador ignora todo el texto que contenga la línea después de un carácter punto y coma. De esta manera



Manual de introducción al uso de microcontroladores PIC ARDUINO

pueden incluirse líneas que contengan solo comentarios, y es muy buena práctica hacer uso y abuso de esta posibilidad para que los programas resulten autodocumentados.

## Instrucciones

Los mismos formatos, iguales modos de direccionamiento y las mismas 35 instrucciones que tenía el PIC 16F84 sirven para todos los modelos PIC 16F87X. No obstante, en los nuevos PIC, al contener más recursos, existen nuevos registros específicos de control cuyos bits se deberán escribir o leer para su gobierno.

INSTRUCCIONES QUE MANEJAN REGISTROS	
ADDWF	Suma W y F
ANDWF	AND W con F
CLRF	Borra F
CLRW	Borra W
COMF	Complementa F
DECF	Decrementa F
DECFSZ	Decrementa F, si es 0 salta
INCF	Incrementa F
INCFSZ	Incrementa F, si es 0 salta
IORWF	OR entre W y f
MOVF	Mueve f
MOVWF	Mueve W a f
NOP	No opera
RLF	Rota f a la izquierda, a través del acarreo
RRF	Rota f a la derecha, a través del acarreo
SUBWF	Resta a f el registro W
SWAPF	Intercambia f
XORWF	XOR de W con f

INSTRUCCIONES QUE MANIPULAN BITS	
BCF	Borra bit de f
BSF	Pone a 1 el bit de f
BTFSC	Testea un bit de f y salta si vale 0
BTFSS	Testea un bit de f y salta si vale 1

INSTRUCCIONES DE CONTROL Y DE OPERANDOS INMEDIATOS	
ADDLW	Suma inmediata a W
ANDLW	AND inmediato con W
CALL	Llamada a subrutina
CLRWDT	Borra el Perro guardián
GOTO	Salto incondicional
IORLW	OR inmediato con W
MOVLW	Mueve a W un valor inmediato
RETFIE	Retorno desde interrupción
RETLW	Retorno y carga de W
RETURN	Retorno de subrutina
SLEEP	Pasa a estado de reposo
SUBLW	Resta W de un inmediato
XORLW	OR Exclusiva a W



## UNA MIRADA RAPIDA AL MPLAB



### Qué es el MPLAB ?

EL MPLAB es un “Entorno de Desarrollo Integrado “ (Integrated Development Environment, IDE) que corre en “Windows “, mediante el cual Usted puede desarrollar aplicaciones para los microcontroladores de las familias PIC 16/17.

EL MPLAB le permite a Usted escribir, depurar y optimizar los programas (firmware) de sus diseños con PIC 16/17. EL MPLAB incluye un editor de texto, un simulador y un organizador de proyectos. Además, el MPLAB soporta el emulador PICMASTER y a otras herramientas de desarrollo de Microchip como el PICSTART - Plus.

### ¿De qué forma le ayuda el MPLAB ?

#### Con el MPLAB Usted puede:

- Depurar sus programas fuente.
- Detectar errores automáticamente en sus programas fuente para editarlos.
- Depurar los programas utilizando puntos de corte (breakpoints) mediante valores de los registros internos.
- Observar el flujo del programa con el simulador MPLAB -SIM, ó seguirlo en tiempo real utilizando el emulador PICMASTER.
- Realizar medidas de tiempo utilizando un cronómetro.
- Mirar variables en las ventanas de observación.
- Encontrar respuestas rápidas a sus preguntas, utilizando la Ayuda en línea del MPLAB.

## LAS HERRAMIENTAS DEL MPLAB

El Organizador de Proyectos (Project Manager).

El organizador de proyectos (Project Manager) es parte fundamental de MPLAB. Sin crear un proyecto Usted no puede realizar depuración simbólica. Con el Organizador de

Proyectos (Project manager) puede utilizar las siguientes operaciones:

- Crear un proyecto.
- Agregar un archivo de programa fuente de proyecto.
- Ensamblar o compilar programas fuente.
- Editar programas fuente.
- Reconstruir todos los archivos fuente, o compilar un solo archivo.
- Depurar su programa fuente.

### Software ensamblador:

El software ensamblador que presenta Microchip viene en dos presentaciones, una, para entorno DOS llamado MPASM.EXE y la otra, para entorno Windows llamado MPASMWIN.EXE

Las dos presentaciones soportan a TODOS los microcontroladores de la familia PIC de Microchip.

El conjunto de instrucciones de los microcontroladores PIC es en esencia la base del lenguaje ensamblador soportado por este software.

**Directivas de uso frecuente:**

Son instrucciones para el compilador.

**#DEFINE**

ej. #define <nombre> [<valor a reemplazar>]

explicación: declara una cadena de texto como sustituto de otra

**END**

ej. end

explicación: indica fin de programa

**EQU**

ej. status equ 05

explicación: define una constante de ensamble

**INCLUDE**

ej. include <PIC16F84.h>

explicación: incluye en el programa un archivo con código fuente

**ORG**

ej. org 0x100

explicación: ensambla a partir de la dirección especificada Para información más completa referirse a la guía rápida del MPASM.

Una vez instalado adecuadamente el MPLAB, para realizar la simulación de un programa deben seguirse los siguientes pasos:

Edite en un archivo de texto el siguiente programa:

**;ejemplo:**

```
status equ 0x03 ;hace equivalencia entre el símbolo status indicándolo como 3 en hexadecimal
```

```
Cont equ 0x20
```

```
F equ 1
```

```
org 0 ;indica posición de memoria desde donde se ensambla
```

```
Inicio
```

```
movlw 0x0F ;carga de w con el valor constante 15 (literal)
```

```
movwf Cont ;el contenido de w se pasa al reg. CONT
```

```
Loop
```

```
decfsz Cont,F ;decremento de Cont y elude siguiente si=0
```

```
goto Loop ;salto incondicional a Loop
```

```
goto $ ;Salto incondicional aqui mismo
```

```
end ;Fin del código
```



1. Haga doble click en el ícono correspondiente a MPLAB.
2. Crear el archivo fuente correspondiente (menú File...New Source).
3. Salve el archivo (con extensión .ASM) una vez terminada su edición (menú FILE...Save).
4. Debe a continuación crearse un nuevo proyecto (menú Project...New Project).
5. Cuando aparezca la ventana de New Project editar las cajas de texto:  
Project path and Name y Development Mode, hacer click en <OK>.
6. En la siguiente ventana Edit Project, hacer click en la sección Non-project files sobre el nombre del archivo fuente realizado en los pasos 2 y 3.
7. Haga click en el botón <=>add y luego de que éste aparezca en la sección Project Files haga click sobre el botón <OK>.
8. Salvar el proyecto (en el menú Project...Save project).
9. Realizar la "construcción de todo el proyecto" (menú Project...Build All).
10. En esta etapa se realiza en forma automática el ensamble de nuestro programa fuente y el vaciado de éste en memoria de simulación. El proceso de ensamble generará un archivo de errores en caso de que estos existan, sí es así deben corregirse directamente sobre el archivo fuente, salvar las correcciones y reconstruir el proyecto (menú Project...Build All). <<<En esta etapa del proceso ya se tiene el entorno listo para la simulación>>>

## Ejemplos de programas con ASEMBLER

### PROGRAMA 1

Lee el estado de los 6 interruptores del entrenador (RA5-RA0) y reflejar el nivel lógico de los mismos sobre los leds RB5-RB0 conectados a la puerta B

```

List      p=16F876
include  "P16F876.INC"

org      0x05

Inicio   clrf      PORTB      ;Borra el Puerto B
         bsf      STATUS,RP0  ;Selecciona banco 1
         clrf      TRISB      ;Puerta B se configura como salida
         movlw    0x06
         movwf    ADCON1      ;Puerta A configurada como digital
         movlw    b'00111111'
         movwf    TRISA       ;Puerta A se configura como entrada
         bcf      STATUS,RP0  ;Selecciona banco 0

Aquí     movf      PORTA,W     ;Leer las entradas RA0-RA5
         movwf    PORTB      ;Reflejar en las salidas
         goto     Aquí        ;Bucle de lectura

end

```

**PROGRAMA 2**

Control de los leds RB0 y RB1 desde el interruptor RA0. RB0 refleja el estado de RA0, RB1 el complemento de RA0

```

List      p=16F876
include   "P16F876.INC"

org      0x05

Inicio    clrf      PORTB      ;Borra el Puerto B
          bsf      STATUS,RP0   ;Selecciona banco 1
          clrf      TRISB      ;Puerta B se configura como salida
          movlw    0x06
          movwf    ADCON1      ;Puerta A como entrada digital
          movlw    b'00011111'
          movwf    TRISA      ;Puerta A se configura como entrada
          bcf      STATUS,RP0   ;Selecciona banco 0

Aquí      btfsc    PORTA,0      ;RA0 = 1 ??
          goto     RA0_es_1     ;Si
          bcf      PORTB,0      ;No, desconecta RB0
          bsf      PORTB,1      ;Conecta RB1
          goto     Aquí         ;Bucle

RA0_es_1  bsf      PORTB,0      ;Activa RB0
          bcf      PORTB,1      ;Activa RB1
          goto     Aquí         ;Bucle

end

```

**PROGRAMA 3**

Se trata de realizar una rotación secuencial en el encendido de cada led conectados a la puerta B del entrenador. Si RA0 = 0, la rotación será de derecha a izquierda y viceversa.

Cada led permanece encendido 0.25 segundos (250 mS)



```

List           p=16F876      ;Tipo de procesador
include       "P16F876.INC" ;Definiciones de registros internos

Contador      equ        0x20      ;Variable para la temporización

org          0x05
goto         Inicio

```

-----  
;Delay es una rutina que realiza una temporización de 250 mS que es el tiempo en que han  
;de permanecer encendido cada uno de los leds. Se basa en repetir 25 veces la temporización  
;de 10mS que se empleó en el ejercicio anterior.

```

Delay         movlw      b'10'
              movwf     Contador      ;Carga el contador con 10
Delay_0       bcf       INTCON,T0IF   ;Desconecta el flag del TMR0

              movlw      b'195'
              movwf     TMR0          ;carga el TMR0 con 195
Delay_1       btfss     INTCON,T0IF   ;Rebosamiento del TMR0 ??
              goto      Delay_1      ;No. Todavía no han pasado los 1* mS
              decfsz    Contador,F    ;Decrementa contador.

              goto      Delay_0      ;Todavía no, temporiza otros 10 ms
              return    ;Ahora si

```

```

Inicio       clrf       PORTB        ;Borra el Puerto B
              bsf       STATUS,RP0   ;Selecciona banco 1
              clrf     TRISB        ;Puerta B se configura como salida
              movlw    0x06
              movwf    ADCON1       ;Puerta A digital
              movlw    b'00011111'
              movwf    TRISA        ;Puerta A se configura como entrada
              movlw    b'00000111'
              movwf    OPTION_REG   ;Preescaler de 256 para el TMR0
              bcf      STATUS,RP0   ;Selecciona banco 0

              bsf      STATUS,C      ;Activa el carry
Loop         call      Delay        ;Temporiza 250mS
              btfsc   PORTA,0      ;Está a 0 RA0 ??
              goto    A_Dcha       ;No, rotación a derecha
A_Izda      rlf       PORTB,F      ;Si, rotación a izquierda
              goto    Loop
A_Dcha      rrf       PORTB,F      ;Rotación a derecha
              goto    Loop

end

```



## Empezando a programar en FLOWCODE

EL PROGRAMA Flowcode es un potente lenguaje basado en gráficos que emplea macros para facilitar el control de dispositivos complejos como los displays de 7 segmentos, controladores de motor y pantallas LCD.

El uso de macros permite a los estudiantes controlar dispositivos electrónicos altamente complejos sin necesidad de ahogarse en el entendimiento de la programación implicada.

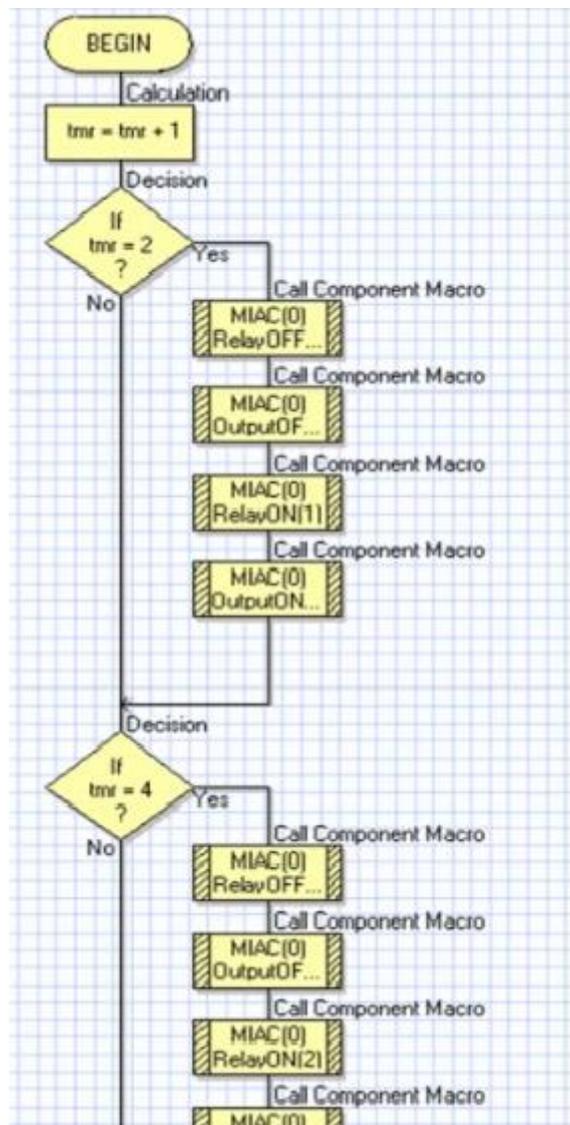
La gran ventaja de Flowcode es que permite a aquellos que tienen poca experiencia crear sistemas electrónicos y robóticos complejos.

## El proceso de diseño

Inserta iconos de diagrama de flujo para crear un programa.

Con solo un click en los iconos y componentes puedes cambiar las propiedades.

Cuenta con componentes que van desde un simple led hasta componentes de BLUETOOTH.

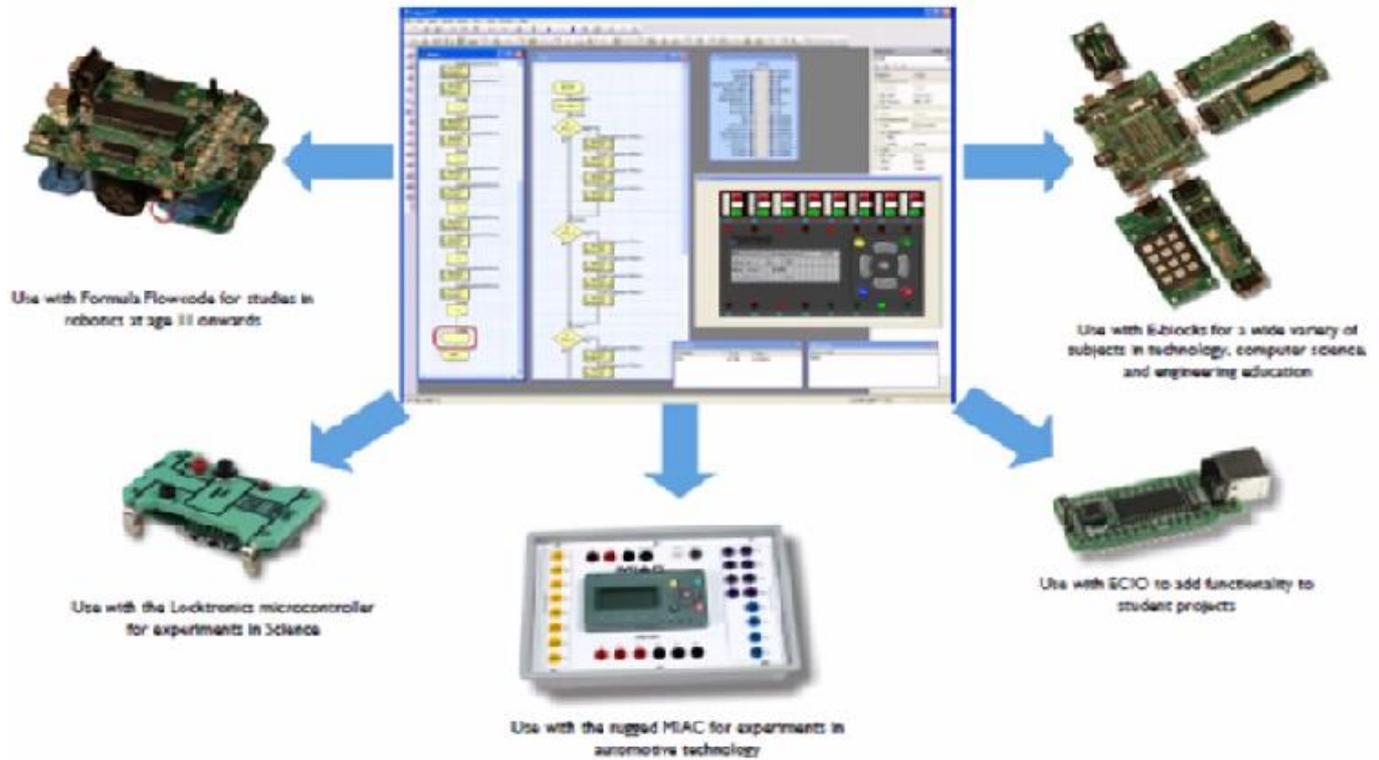


## Diseño de sistemas

Cuenta con un panel para colocar componentes y ajustarlos de manera gráfica.

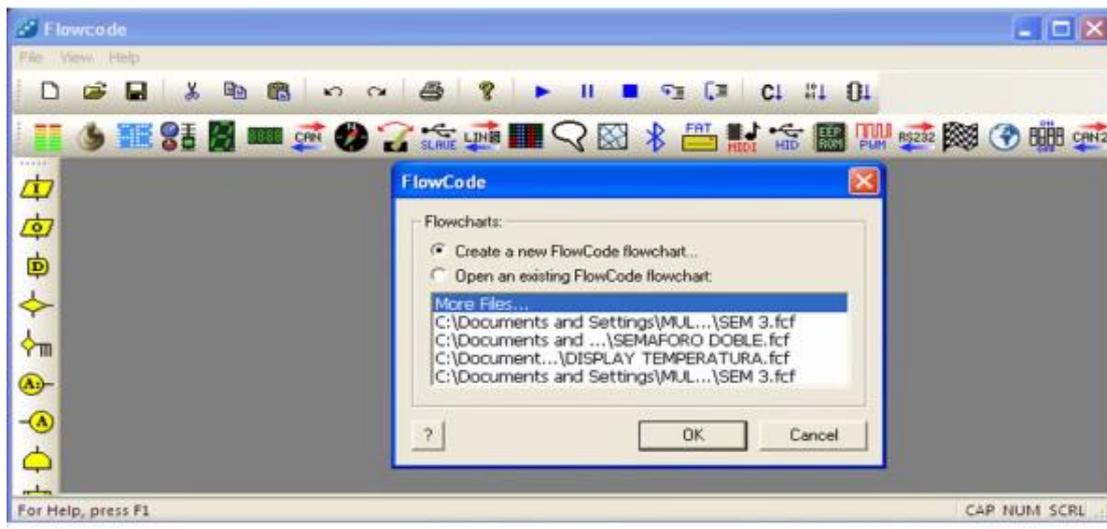
## Flowcode en la educación

Flowcode es reconocido internacionalmente como un líder en el desarrollo de microcontroladores para la educación. Es usado en más de 700 escuelas, colegios y universidades a nivel mundial.

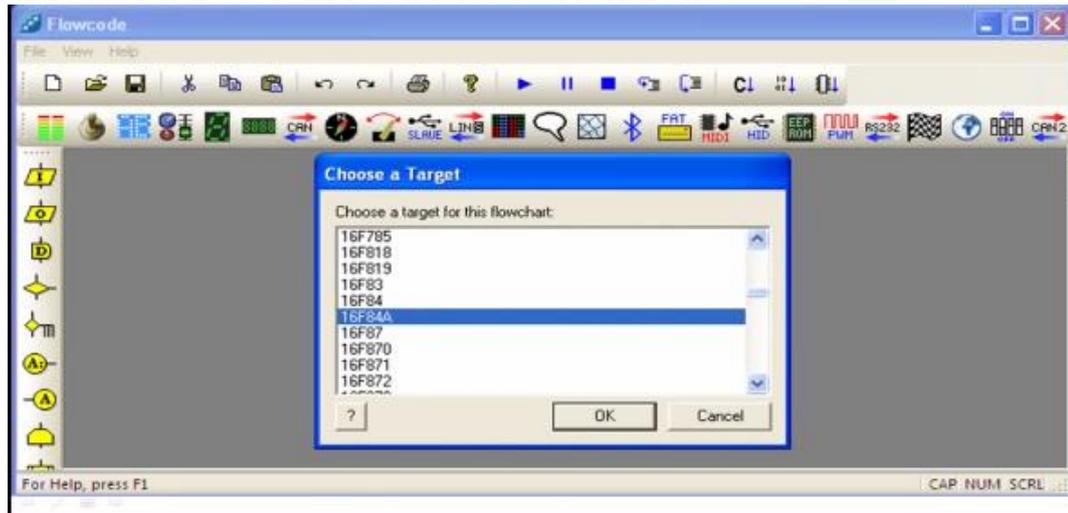


## Primera experiencia

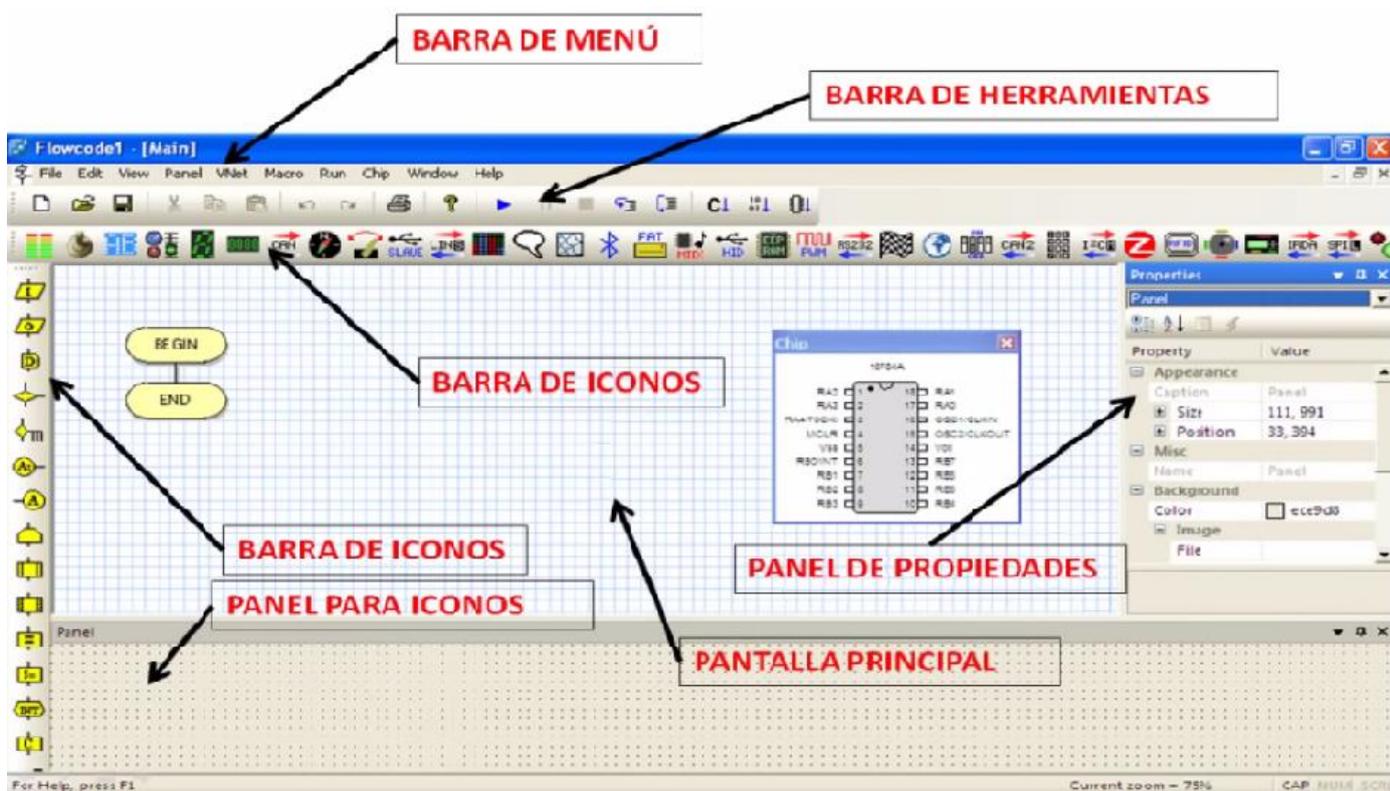
Paso 1: abrir FLOWCODE y seleccionar **CREAR UN DIAGRAMA DE FLUJO**



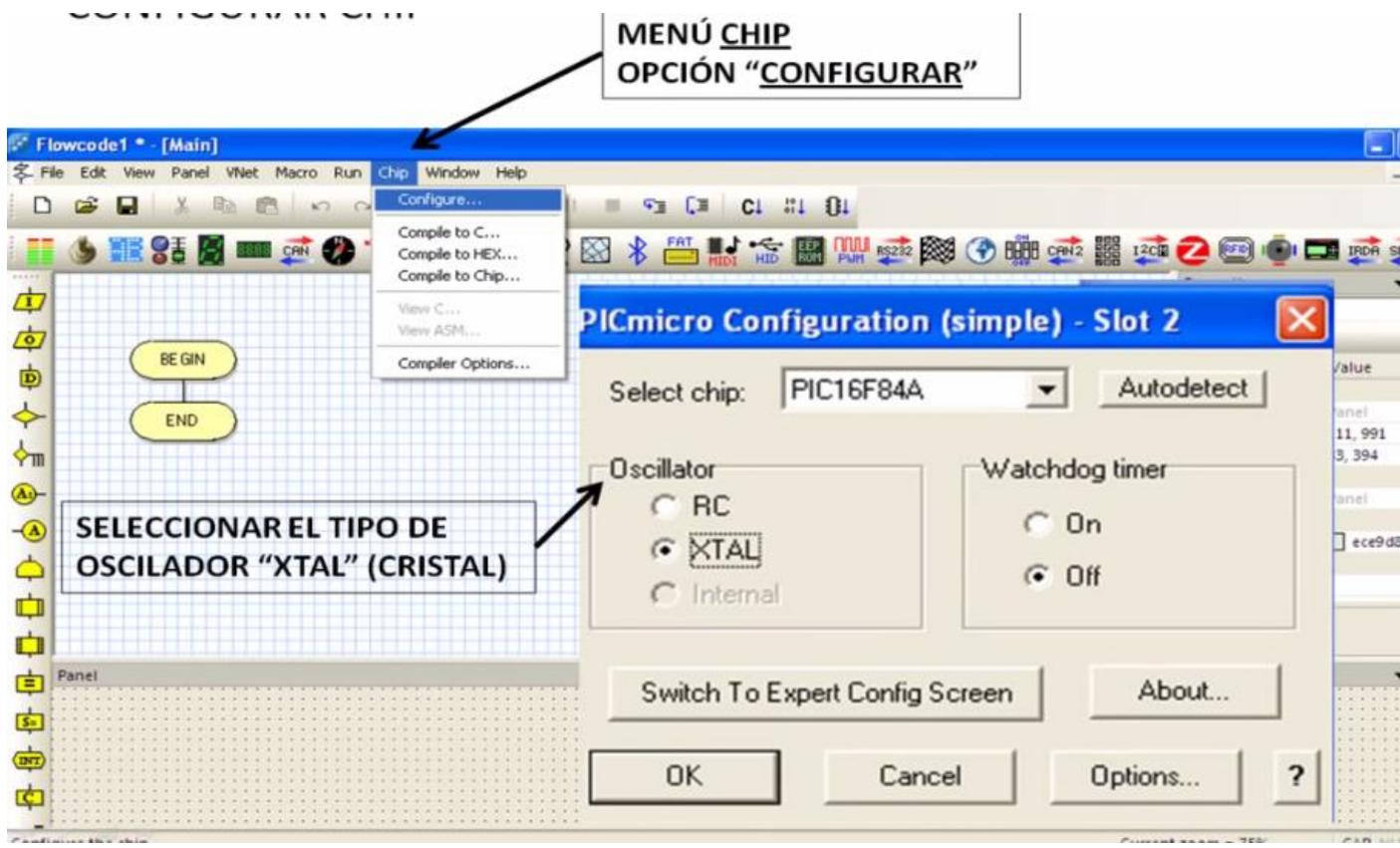
Paso 2: seleccionar el **TIPO DE PIC** de acuerdo a su familia.



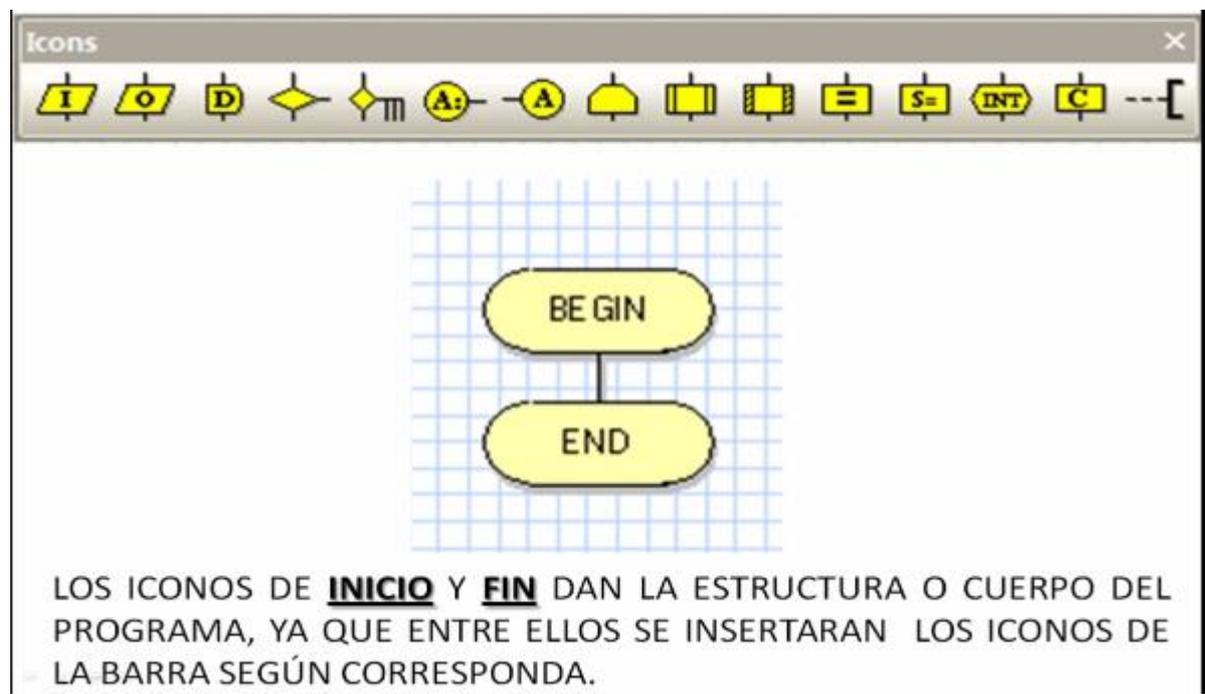
Pantalla principal del programa



Paso 3: configurar el **CHIP**



Barra de iconos





**ENTRADA (INPUT).**- INSERTA UNA ENTRADA DE DATOSA TRAVÉS DE UN PUERTO SELECCIONADO



**SALIDA (OUTPUT).**- INSERTA UNA SALIDA DE DATOS A TRAVÉS DE UN PUERTO SELECCIONADO



**RETARDO (DELAY).**- INSERTA UN TIEMPO DE RETARDO AL PROGRAMA



**DECISIÓN.**- INSERTA UNA DECISIÓN.



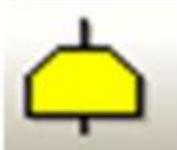
**SWITCH.**- INSERTA UNA SWITCH PARA TENER N CONDICIONES.



**PUNTO DE CONEXIÓN.**- DECLARA EL NOMBRE DE UN PUNTO DE CONEXIÓN.



**PUNTO DE CONEXIÓN.**- BRINCA A UN PUNTO DE CONEXIÓN.



**LAZO (LOOP).**- INSERTA UN LAZO CERRADO HASTA UNA DETERMINADA CONDICIÓN.



**MACRO.-** PERMITE INSERTAR UNA MACRO O SUB RUTINA QUE PUEDE SER USADA DENTRO DEL PROGRAMA.



**COMPONENTE DE MACRO.-** INSERTA UNA COMPONENTE DE MACRO QUE SOLO PUEDE SER USADA PARA UN COMPONENTE.



**CALCULO O PROCESO.-** INSERTA UNA OPERACIÓN O CALCULO.



**OPERACIÓN DE STRING.-** PERMITE REALIZAR OPERACIONES DE VARIABLES STRING.



**INTERRUPCIÓN.-** INSERTA UNA INTERRUPCIÓN AL PROGRAMA.



**CODIGO C.-** INSERTA CODIGO DE PROGRAMACIÓN EN LENGUAJE C.



**COMENTARIO.-** INSERTA UN COMENTARIO AL PROGRAMA.

## Ejemplo de programa armado

**1. PROGRAMA SE DESARROLLA COMO DIAGRAMA DE FLUJO**

**2. SE DEFINEN LAS COMPONENTES LEDS Y MICROSWITCHES DEL SISTEMA EB88**

**3. PROGRAMA PUEDE SIMULARSE EN TIEMPO REAL Ó PASO POR PASO**

**4. SE COMPILA EL PROGRAMA, SIN NECESIDAD DE ESCRIBIRLO EN ANSI C**

**5. EL ARCHIVO EJECUTABLE .HEX RESULTADO DE LA COMPILACIÓN, SE CARGA AL SISTEMA EB88, POR MEDIO DEL SOFTWARE "TINY"**

**Chip**

RADIAN2/REFV/REF-	1	18	RA1/AN1
RADIAN3/REF+/CIOUT	2	17	RADIAN0
RA1/AN4/IOCK/CIOUT	3	16	RA7/OSC1/CLKO
RASIMCLR/VPP	4	15	RA6/OSC2/CLKO
VDD	5	14	VDD
RB0/INT/CCP1	6	13	RB7/AN6/PB0T/IOB1
RB1/SDA/SDA	7	12	RB6/AN5/PB1T/IOB1/CKI
RB2/SCK/PXD1	8	11	RB5/SSTX/CK
RB3/PGM/CCP1	9	10	RB4/SCK/SCL

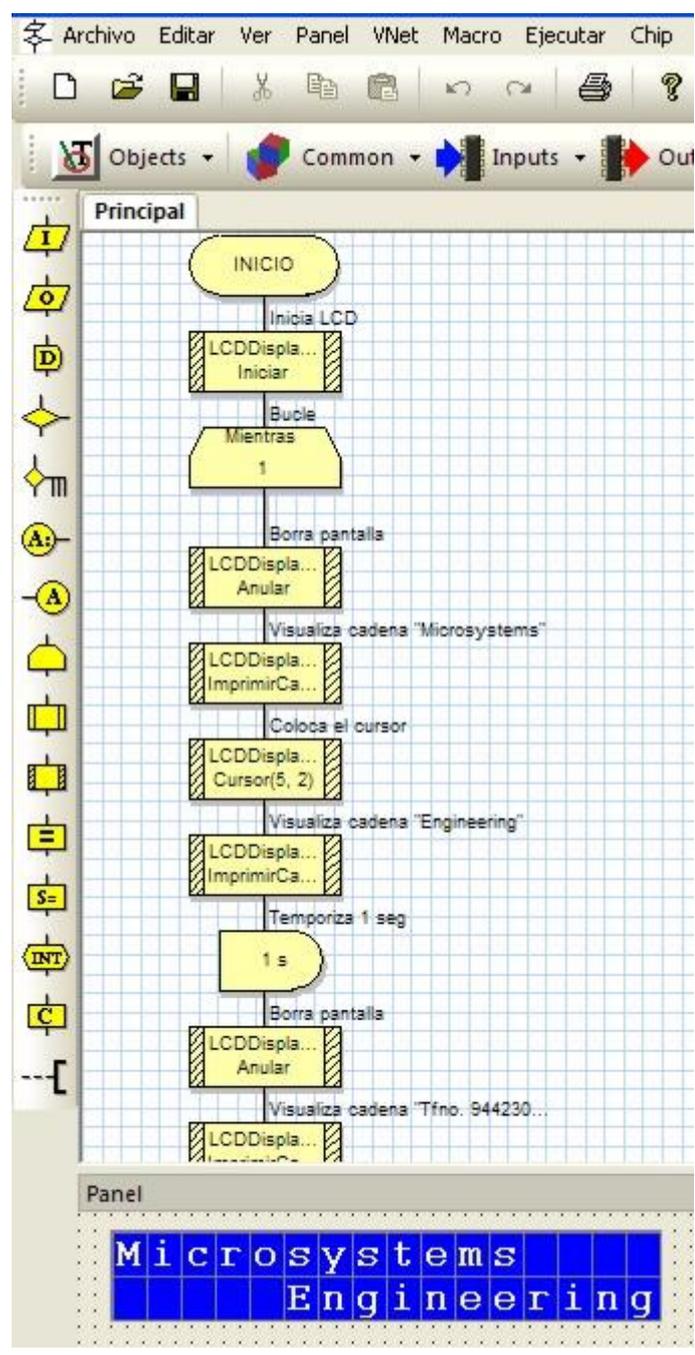
**LEDs(2)**

B7	B6	B5	B4	B3	B2	B1	B0
●	●	●	●	●	●	●	●

**Switch(1)**

A4	SW4
A3	SW3
A2	SW2
A1	SW1

Programando un lcd



Como pueden observar este método de programación es sumamente simple en comparación a los sistemas de programación tradicionales.

Las prácticas de este se dejan a cargo de los alumnos en el taller de electrónica.

Canal de YouTube donde se muestran ejemplo paso a paso:

<https://youtu.be/57kD9NJ7Kwo>



## Bibliografía:

- **Entradas y salidas digitales de propósito general en los microcontroladores dsPIC30 Ejemplos con el dsPIC30F4013** Autor: Juan Ramon Rufino Valor. Marzo 2010.
- **INTRODUCCIÓN AL MICROCONTROLADOR** Basado en el sitio web <http://usuarios.lycos.es/sfriswolker> Paul Aguayo S., [paguayo@olimex.cl](mailto:paguayo@olimex.cl) 10 de noviembre de 2004
- **Una Guía Práctica a los Dispositivos de Libre-energía.** autor: Patrick Kelly
- <http://www.free-energy-info.tuks.nl>
- <http://www.alldatasheet.com>
- **Curso avanzado de Microcontroladores PIC . CEKIT**
- [www.microchip.com](http://www.microchip.com)
- **Introducción a los microcontroladores,** José Adolfo González V., McGraw Hill
- **Microcontroladores PIC,** Tavernier, Editorial Paraninfo
- **ELECTRONICA ANALOGICA Y DIGITAL.** Autor ANTONIO GONZALEZ *Instituto Preuniversitario Escuela Industrial D. F. Sarmiento U.N.S.J.*

### SOBRE PROGRAMACIÓN

- <https://www.youtube.com/user/elprofegarcia/videos>
- [http://www.matrixtsl.com/lc\\_index.php?p=18](http://www.matrixtsl.com/lc_index.php?p=18)
- <https://youtu.be/57kD9NJ7Kwo>