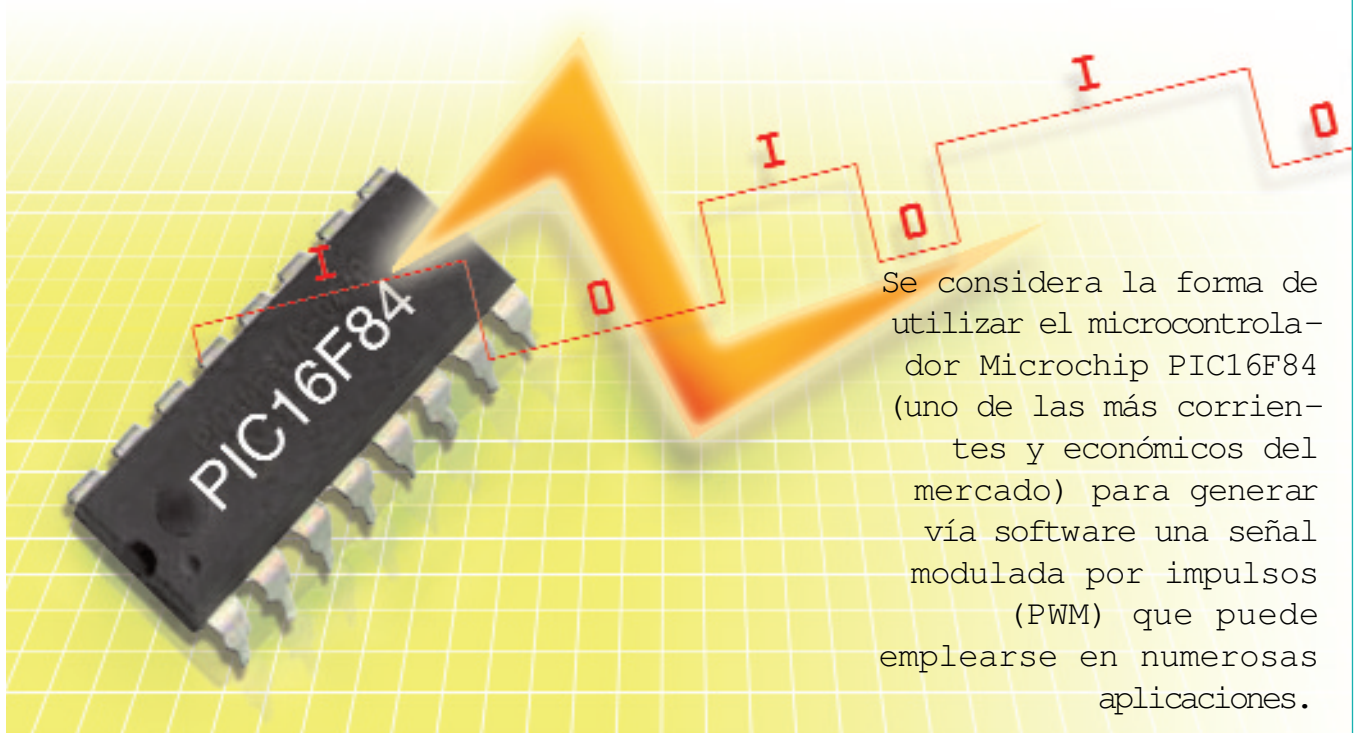


Generación de señales PWM con el microcontrolador PIC16F84

por Pietro Iogliesci



Se considera la forma de utilizar el microcontrolador Microchip PIC16F84 (uno de las más corrientes y económicos del mercado) para generar vía software una señal modulada por impulsos (PWM) que puede emplearse en numerosas aplicaciones.

PWM es el acrónimo de "Pulse Width Modulation", expresión que designa un modo concreto de modulación, la llamada "Modulación de impulsos en anchura". Inicialmente, esta técnica se utilizaba casi exclusivamente para el control de potencia y velocidad de motores de corriente continua, pero con el tiempo se ha ido ampliando el campo de aplicación, por ejemplo, en las fuentes de alimentación conmutadas, onduladores c.c.-c.a., etc.; con ella se ha conseguido realizar dispositivos mucho más eficientes, más compactos y más ligeros. El presente artículo se destina a

describir en qué forma puede utilizarse un microcontrolador PIC16F84 para generar una señal del tipo PWM con la que se regulará la emisión luminosa de un Led y se controlará la velocidad de un pequeño motor de corriente continua: con esta base, el lector estará capacitado para desarrollar sus propias aplicaciones.

¿Qué es la modulación PWM?

Antes de la puesta a punto de la técnica PWM, la velocidad de un motor de corriente continua se regulaba >

Características Técnicas

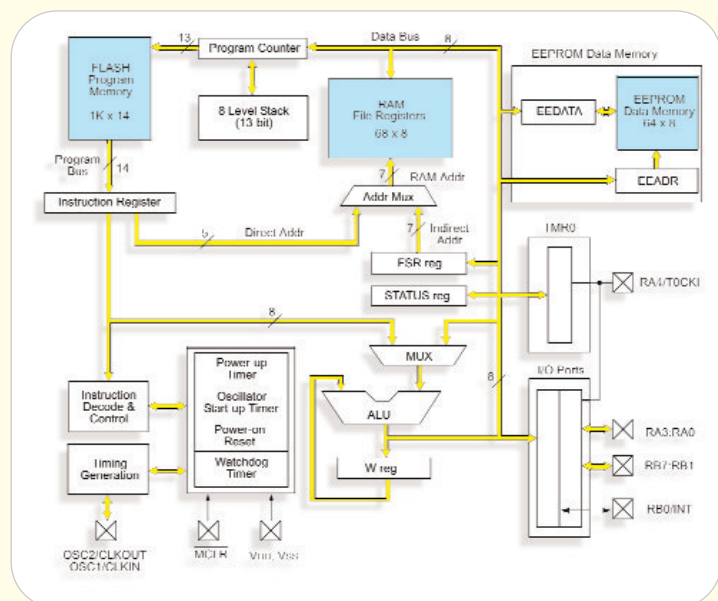
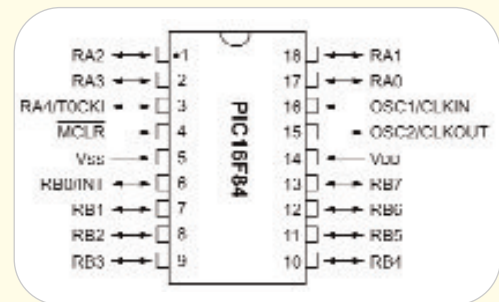
- Arquitectura RISC con 35 instrucciones.
- La mayor parte de las instrucciones se ejecuta en un ciclo máquina.
- Máxima frecuencia de reloj: 20 MHz, ciclo máquina 20 nanosegundos.
- 1024 localizaciones (14 bits) de memoria de programa.
- 68 bytes de memoria RAM.
- 64 bytes de memoria EEPROM.
- Localizaciones de memoria de programa de 14 bits.
- Localizaciones de memoria de datos de 8 bits.
- 15 Registros Especiales.
- Área de Stack de ocho niveles.
- Direccionamiento inmediato, directo, indirecto.
- Cuatro métodos posibles de interrupt:
 1. Interrupt externo mediante la línea RB0/INT;
 2. Interrupt interno con TIMER TMR0;
 3. Interrupt externo sobre las líneas RB4÷RB7;
 4. Interrupt para escritura hasta localización de datos EEPROM.

Características de los Periféricos

- 13 líneas bidireccionales de I/O;
- Importante corriente entregada/absorbida sobre cada línea:
 1. 25 mA máx consumidos (sink) en cada puerta;
 2. 25 mA máx entregados (source) por cada puerta;
- TMR0: contador temporizados de 8 bits con preescaler programable.

El microcontrolador PIC16F84

Este es uno de los más conocidos y ampliamente utilizados microcontroladores de 8 bits del mercado. Su clásica configuración de 18 patillas (Dual in Line 9 + 9) se reproduce en estas páginas, junto al esquema de bloques interno. La tendión de alimentación se aplica a las patillas VDD (positivo) y Vss (negativo, masa), que puede estar entre 2 y 6 V (valor típico 5 V). La frecuencia de reloj depende de las características del cuarzo o de la red RC conectada entre las patillas OSC1/CLKIN y OSC2/CLKOUT; habitualmente se utiliza la frecuencia es de 4 MHz, pero el integrado puede funcionar con una frecuencia de reloj de hasta 20 MHz (PIC16F84A). La CPU, del tipo RISC (Reduced Instruction Set Computer), puede elaborar hasta 35 instrucciones que se ejecutan en un solo ciclo máquina (excepto las instrucciones de salto, que precisan de dos ciclos). La elevada velocidad conseguida se debe a la técnica específica utilizada, denominada "pipeline" de dos estados, que consiste en ejecutar una instrucción mientras en el registro de instrucciones se carga la instrucción siguiente para su descodificación. El microcontrolador dispone de 15 registros especiales, los más importantes de los cuales son el Program Counter PC y el Register Accumulator W. El primero aumenta automáticamente durante la ejecución de un programa para que pueda contener la dirección de la siguiente instrucción a ejecutar; el segundo lo utiliza el microcontrolador para efectuar numerosas operaciones aritméticas y de memorización simultánea de datos. El RESET depende de la línea MCLR (patilla 4). El microcontrolador PIC16F84 dispone de 13 líneas de I/O bidireccionales instaladas en dos puertos, RA0÷RA4 y RB0÷RB7. Cada línea puede conducir 25 mA como máximo. Para gestionar las interrupciones se dispone de cuatro modalidades, que pueden depender de eventos internos o externos al microcontrolador. La línea principal de interrupt externo se dirige a RB0/INT, mientras que el TIMER TMR0 puede utilizarse como interrupt interno.



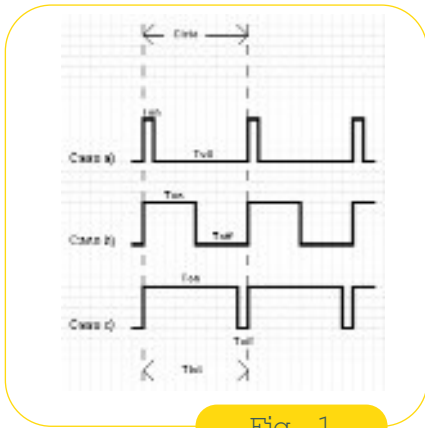


Fig. 1

mediante un potenciómetro o reostato en serie. Este "poco fino" método comportaba un gran gasto de energía, con mucha generación de calor. Cuanta más potencia, mayor desperdicio de energía. Un buen día alguien tuvo una idea feliz: aplicar toda la potencia disponible, pero no continuamente sino en forma de impulsos más o menos breves. Así fue como nacieron los circuitos "chopper", que en un primer momento se realizaron a base de semiconductores discretos. Puede hacerse un sencillo experimento con una pila y una bombillita de filamento, como las de las linternas de bolsillo. Soldar un polo de la pila a la bombillita con un tramo de hilo conductor y cerrar el circuito aproximando a mano el otro polo, a través de otro tramo de hilo. Al hacer un contacto intermitente, juntamente y separando rápidamente el terminal, se observará que, gracias a la inercia de la resistencia y a la persistencia del ojo humano, la bombillita lucirá de forma casi estable, aunque con una intensidad luminosa inferior. Se está aplicando íntegramente la tensión de la pila

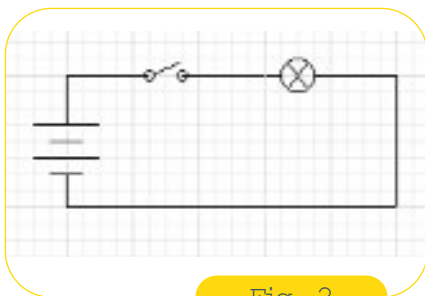


Fig. 2

pero sólo a intervalos. Dos parámetros tienen suma importancia: el tiempo que la bombillita permanece alimentada (Ton) y el tiempo entre dos impulsos de tensión (Toff). Dentro de un cierto periodo de tiempo, cuanto menor sea la suma de tiempos en que la bombillita queda sin alimentar, mayor es la intensidad luminosa emitida. De aquí sale la definición de "ciclo de trabajo" (duty cycle): es el porcentaje que indica el tiempo durante el cual la señal permanece a nivel alto, durante un periodo dado. El ejemplo típico de un ciclo de trabajo del 50 % es una señal perfectamente cuadrada, como las que entregan los generadores de señal. Una tensión de cero voltios corresponde a un ciclo de trabajo del 0 %. En la fig. 1, Ton representa el periodo durante el cual la bombillita recibe la tensión máxima y Toff es el tiempo en que no recibe tensión alguna. Modificando la duración relativa de estos tiempos se consigue modificar la tensión media que llega a los contactos de la lámpara y, por consiguiente, se regula su luminosidad. En la práctica, se actúa sobre Ton y se mantiene constante la duración del ciclo (Ton + Toff). O sea que sólo se modifica el porcentaje relativo de Ton y Toff respecto al ciclo total. Así, la señal PWM puede definirse como una señal de tensión rectangular en la cual se ha previsto una determinada distribución temporal entre impulso alto e impulso bajo. Admitiendo que Ttot no varía, al alargar la duración de Ton, la de Toff se acorta necesariamente. Aplicando una PWM a la alimentación de un motor, si Ton representa el 10 % del ciclo, la tensión aparece en bornes del motor a intervalos muy cortos y el ciclo de trabajo es bajo. Pero si Ton representa el 90 % del ciclo, el motor recibe tensión a intervalos proporcionalmente muy largos y el ciclo de trabajo es elevado.

¿Qué ventajas ofrece la modulación PWM?

Si para la generación de ondas PWM se recurre a un microcontrolador, basta con un solo bit de salida para comandar el paso de On a Off; resulta un circuito sumamente sencillo y muy económico. Por otra parte, para pilotar cargas con variación de potencia, la señal PWM permite elevados rendimientos reduciendo a un nivel mínimo el calor generado por el sistema de control y por la carga, mientras que ésta trabaja siempre en condiciones óptimas: todo abierto o todo cerrado. Pero la ventaja principal de la

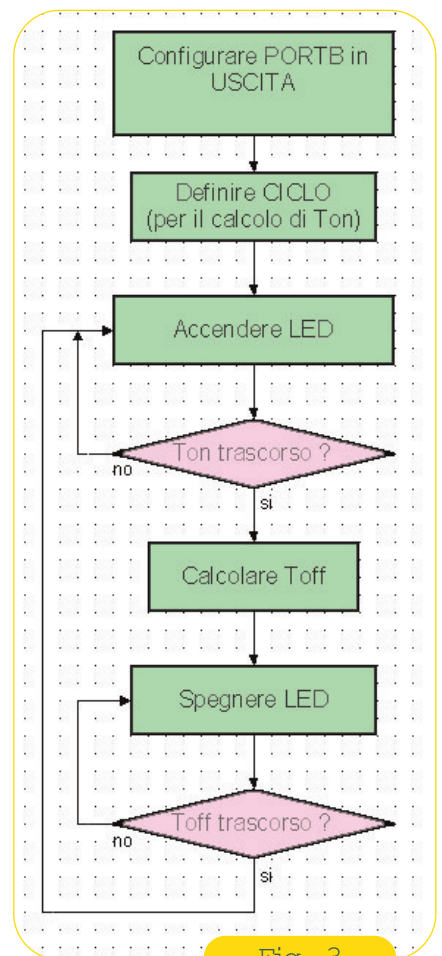


Fig. 3

modulación PWM es poder utilizar impulsos de ataque de frecuencia constante: este concepto prefigura el uso de sistemas sincronizados por un reloj (clock), es decir, por sistemas digitales. Actualmente, la generación de señales PWM se rea-

LISTADO EN ASSEMBLER

```

;Programa experimental para generar una seaal PWM

;-----
;          DIRECTIVAS
;-----
        PROCESSOR      16F84
        INCLUDE         "P16F84.INC"

;-----
;          DECLARACION DE VARIABLES
;-----
CICLO      EQU          0C          ;Para memorizar el ciclo de trabajo.
CONTATORE_1 EQU        0D          ;Para memorizar el multiplicador d Tcn.
CONTATORE_2 EQU        0E          ;Memoria necesaria para DELAY_15 s.

;-----
;          VECTOR DE RESET
;-----
        ORG            00          ;Inicio en Reset.

;-----
;          CONFIGURACION DE LAS PUERTAS
;-----
        BSF            STATUS,RP0  ;Paso por Pagina0.
        MOVLW          b'00000000' ;Se configura el
        MOVWF          TRISB      ;PORTB en USCITA.
        BCF            STATUS,RP0  ;Regreso a Pagina0.

;-----
;          PROGRAMA PRINCIPAL
;-----
        MOVLW          d'127'      ;Se define el ciclo de
        MOVWF          CICLO       ;trabajo (para un 50 %).
INIZIO   MOVF          CICLO,W      ;Transferencia del contenido de
        MOVWF          CONTATORE_1 ;CICLO en CONTATORE_1 (CONTATORE_1
        ;contiene 127).
T_ON     BSF          PORTB,0      ;Se pone a 1 la salida RB0 (encendido
        ;del diodo LED).
        CALL          DELAY_15 s   ;Se llama la temporizaci n y se
        DECFSZ        CONTATORE_1,f ;repite por el nmero de veces
        GOTO         T_ON         ;especificado en CONTATORE_1 (aqu :127).
        MOVLW          d'255'      ;Se carga CONTATORE_1 con 255 (valor
        MOVWF          CONTATORE_1 ;mximo cargable en un octeto).
        MOVF          CICLO,w      ;Transferencia de CICLO a W.
        SUBWF         CONTATORE_1,f ;Resta de 255 y el contenido de
        ;CICLO (aqu : 255-127 = 128).
T_OFF    BCF          PORTB,0      ;Se lleva a 0 la salida RB0 (se apaga
        ;el diodo LED).
        CALL          DELAY_15 s   ;Se llama la temporizaci n y se
        DECFSZ        CONTATORE_1,f ;repite por el nmero de veces
        GOTO         T_OFF        ;especificado en CONTATORE_1 (aqu: 128).

        GOTO         INIZIO

;-----
;          TEMPORIZACION
;-----
DELAY_15 s MOVLW          dZ
        MOVWF          CONTATORE_2
TEMPO     NOP
        DECFSZ        CONTATORE_2,f
        GOTO         TEMPO
        GOTO         FINE
FINE     RETURN
        END

```



liza con circuitos basados en puertas lógicas (dispositivos TTL y CMOS) y microprocesadores o microcontroladores programables (por ejemplo, el PIC16F84) que integran circuitos capaces de generar de forma autónoma señales moduladas con la técnica PWM, con la ayuda de muy pocos componentes periféricos específicos.

Frecuencia de modulación

Aunque el porcentaje relativo entre Ton y Toff pueda variar, y que, en cierto sentido, este valor pueda considerarse un dato analógico, la señal PWM es una señal digital dado que la tensión de salida sólo puede asumir uno de los dos valores digitales posibles: o máximo o nulo. Y eso, independientemente del instante considerado. Tensión o corriente se entregan a la carga en forma de impulsos repetitivos. Como el ancho de banda es considerable, la técnica PWM permite obtener a la salida cualquier valor de tensión comprendido entre cero voltios y la tensión de alimentación. En la fig. 1, el caso **a** muestra una salida PWM con un ciclo de trabajo del 10 %, es decir, que la señal es máxima durante un 10 % del periodo y baja durante el 90 % restante. Los casos **b** y **c** representan una salida PWM con un ciclo de traba-

jo del 50 y del 90 %, respectivamente. A estas tres señales PWM (digitales) corresponden 3 valores analógicos diferentes de la señal de salida: suponiendo una tensión de alimentación de 5 voltios, a un ciclo de trabajo del 10 % corresponde una tensión de salida de 0,5 voltios; al 50 % la salida sería de 2,5 V y al 90 % se obtendrían 4,5 voltios. Pueden aplicarse estas correspondencias al ejemplo de la bombillita (fig. 2); se alimenta de la pila de 9 voltios a través de un interruptor en serie. Cerrando este interruptor 50 milisegundos, abriéndolo 50 milisegundos y repitiendo esta operación 10 veces por segundo, la tensión media aplicada a la bombillita no es la que produce el máximo encendido de ésta: es como si se aplicasen 4,5 voltios. El ciclo de trabajo es del 50 % y la frecuencia de modulación es de 10 hertzios. Esta frecuencia basta para un sencillo ejemplito como éste, pero es claramente insuficiente para una aplicación práctica. En los alimentadores conmutados se utilizan frecuencias comprendidas entre 1 y 200 kHz. Hay varias razones para ello: en el ejemplo de la fig. 2, si se cierra el interruptor 10 segundos y se abre otros 10 segundos, aunque el ciclo de trabajo siga siendo del 50 %, la bombillita no luce como si se alimentase con 4,5 voltios, sino

que aparece 10 segundos apagada y 10 segundos encendida. Así, para obtener una iluminación homogénea equivalente a la obtenida con una pila de 4,5 voltios debe aumentarse la frecuencia hasta que la inercia resistiva de la bombillita y, sobre todo, la de la visión humana, hagan "desaparecer" los periodos de apagado.

Creando un primer programa para el PIC

Aquellos lectores que tienen algún conocimiento previo sobre modulación PWM, y los que ya conocen los rudimentos de la programación de microcontroladores PIC, seguramente ya saben utilizar esta técnica para encender y apagar un diodo, o para hacerlo parpadear. Ahora se trata de encenderlo hasta un cierto nivel de luminosidad. Más adelante se verá cómo hacerlo progresiva y lentamente, creando el llamado "efecto crepuscular". El sencillo programa descrito en el diagrama de flujo de la fig. 3 es capaz de encender un Led a una intensidad inferior a la máxima. La configuración del PortB no merece demasiados comentarios: es el que se ocupa de encender y apagar el Led. Los elementos que sí deben considerarse a fondo son:

- la definición del ciclo (para el cálculo de Ton);
- el cálculo de Toff, y
- la temporización.

Un microcontrolador PIC con su oscilador interno controlado por un cristal de cuarzo de 4 MHz entrega impulsos de 1 μ s, correspondientes a una frecuencia de 1 MHz. Esta frecuencia es demasiado elevada para generar señales PWM, si se acepta como normativa la gama entre 1 y 200 kHz. Por tanto, debe reducirse mediante un ciclo de temporización: se enciende el Led, se

Porcentaje porcentual de encendido (a)	Ciclo de trabajo (a : 255 = 100)	Tensión en el terminal de PORTB (volt) (5V x trabajo de trabajo = 100)
0	0	0
1	0,4 %	0,02
2	0,8 %	0,04
3	1,2 %	0,06
4	1,6 %	0,08
5	2,0 %	0,10
6	2,4 %	0,12
...
...
251	99,6 %	4,98
252	99,8 %	4,99
253	99,9 %	4,995
254	100,0 %	5,00
255	100,0 %	5,00

Fig. 4

temporiza, se apaga el Led, se temporiza y se reinicia el ciclo. En realidad, la duración de Ton y Toff es el resultado de un cálculo: Ton es el producto de una constante definida previamente (CICLO) y guardada en una memoria (CONTATORE_1) y la duración de una temporización (de algunos microsegundos) cuyo objetivo es reducir la frecuencia de modulación hasta que quede comprendida entre 1 y 100 kHz. Toff es la diferencia entre la duración total de CICLO (Ttot) y la duración de Ton: se dejará al propio PIC el trabajo de calcularlo. Como el microcontrolador trabaja con bytes de 8 bits cuyo valor máximo puede llegar a 255 (decimal), se le da al PIC la orden de calcular la diferencia 255-Ton. El resultado, multiplicado por el valor de la temporización, proporciona la duración del tiempo de apagado del Led (Toff).

La tabla de la fig. 4 ilustra sobre el funcionamiento descrito.

En estas páginas se reproduce el programa en Assembler que rige toda esta actividad. Como puede observarse, se ha implantado una temporización de 15 microsegundos (aunque podría ser más larga) en función de la carga y del efecto que se desea obtener. La oscilación de encendido del Led no es fácil de apreciar visualmente: sí podrá detectarse midiendo la tensión de salida del PIC con un polímetro (tester) situado en la posición de lectura de tensiones continuas (CC); también pueden instalarse dos Leds idénticos, uno junto al otro, conectando uno de ellos al puerto PORTB0 y el otro al PORTB1. El primero experimentará las variaciones de alimentación derivadas de la modulación PWM, mientras que el otro, que quedará alimentado de forma permanente, sirve de referencia. Pueden hacerse más pruebas modificando solamen-

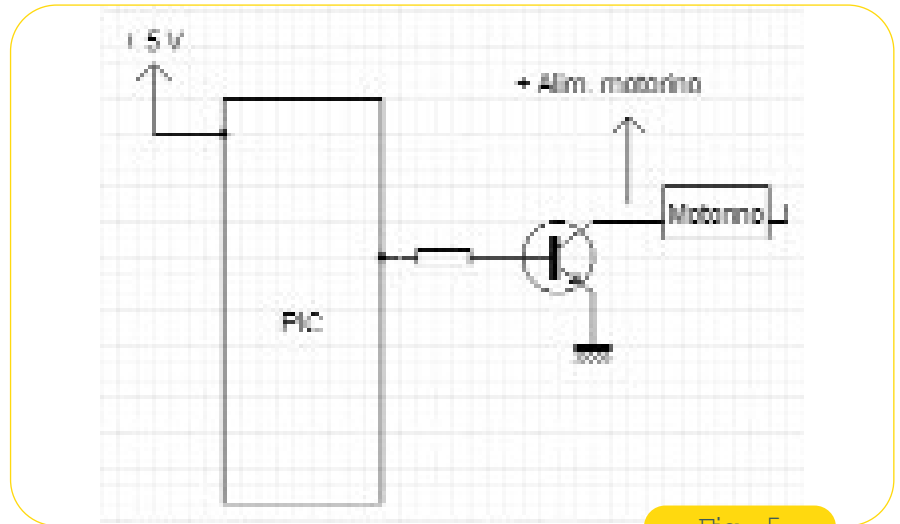


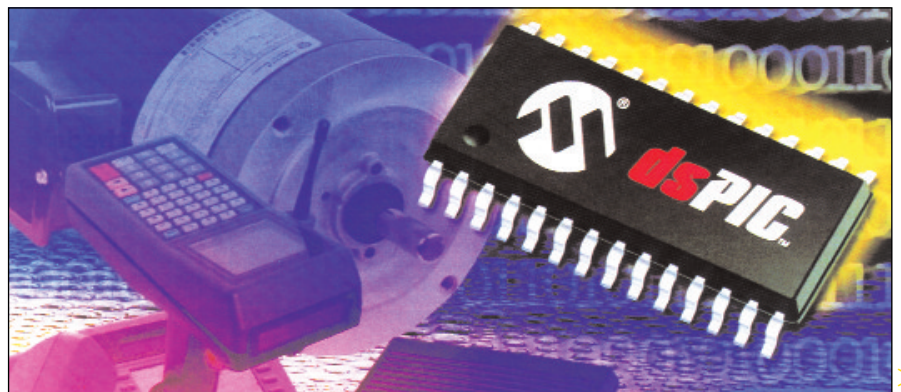
Fig. 5

te el valor memorizado en CICLO. Si, por ejemplo, en lugar de 127 se introduce 5, el microcontrolador calculará automáticamente un Ton de 75 μ s (5×15) y un Toff 3.750 μ s (250×15). Al comparar la iluminación que emiten el Led pilotado por la onda PWM y el permanentemente encendido, la diferencia se observa a simple vista, mejor cuanto menores son los niveles de iluminación, ya que el ojo humano distingue mejor las diferencias de luz a niveles bajos que a niveles altos. Sustituyendo el Led por un motor de corriente continua precedido de una etapa buffer puede obtenerse el mismo efecto, si bien en este caso el resultado se manifestará por diferentes velocidades del motor (fig. 5). El tipo de semiconductores y los valores de las resistencias de base dependen del tipo de motor, más exactamente, de la tensión de alimentación de éste y de la corriente

necesaria para su funcionamiento.

Efecto crepuscular

El truco consiste en confiar al PIC el cálculo de la Toff, evitando así errores en la distribución de las duraciones entre impulso alto e impulso bajo en cada ciclo concreto, lo que equivale a automatizar las operaciones de paso a un programa capaz de generar una señal PWM que se incremente por sí sola, creando un efecto crepuscular. Con este sistema es suficiente memorizar un cero (0) en la zona CICLO de la memoria y añadir al programa una sola instrucción (INCF CICLO) para que el Led se encienda de forma progresiva. Véase el programa modificado que se incluye en estas páginas: con él se produce el encendido del Led con efecto alba. Para obtener el efecto inverso (crepúsculo, al apagarse gradual-



E F F E T T O C R E P U S C U L A R

```

;Programa para generar una seaal PWM con efecto crepuscular (alba)
;-----
;
;          DIRECTIVAS
;-----
PROCESSOR      16F84
INCLUDE        "P16F84.INC"
;-----
;
;          DECLARACION DE VARIABLES
;-----
CICLO          EQU      0C      ;Para memorizar el ciclo de trabajo.
CONTATORE_1   EQU      0D      ;Para memorizar el multiplicador deTcn.
CONTATORE_2   EQU      0E      ;Memoria necesaria para DELAY_15 s.
;-----
;
;          VECTOR DE RESET
;-----
ORG           00              ;Inicio en Reset.
;-----
;
;          CONFIGURACION DE LAS PUERTAS
;-----
BSF           STATUS,RP0     ;Paso por Pagina1.
MOVLW        b'00000000'    ;Se configura el
MOVWF        TRISB          ;PORTB en SALIDA.
BCF           STATUS,RP0     ;Retorno a Pagina0.
;-----
;          PROGRAMA PRINCIPAL
;-----
INIZIO
BSF           PORTB,1        ;Encendido de un LED en modo
                           ;continuo
                           ;para verificar la
                           ;diferencia de luminosidad.
MOVLW        d'0            ;Se define el valor de inicio
MOVWF        CICLO          ;de la rampa.
INCF         CICLO,f        ;Se empieza a incrementar.
MOVWF        CONTATORE_1   ;Transferencia del contenido de
                           ;CICLO en CONTADOR_1.
T_ON
BSF           PORTB,0        ;Se lleva a 1 la salida RB0
                           ;(encendido del diodo LED).
CALL         DELAY_15 s     ;Se llama la temporizacion
                           ;de la si
DECFSZ       CONTATORE_1,f  ;Repite el numero de veces
GOTO         T_ON           ;especificado en CONTADORE_1.
MOVLW        d'255'        ;Se carga CONTATORE_1 con 255
                           ;(valor
                           ;maximo cargable en un
                           ;octeto).
MOVWF        CONTATORE_1
MOVWF        CICLO,w        ;Transferencia de CICLO a W.
SUBWF        CONTATORE_1,f  ;Resta de 255 y el
                           ;contenido de CICLO.
T_OFF
BCF           PORTB,0        ;Se lleva a 0 la salida RB0
                           ;(apagado
                           ;del diodo LED).
CALL         DELAY_15 s     ;Se llama la temporizacion
                           ;y se la repite
DECFSZ       CONTATORE_1,f  ;el numero de veces
GOTO         T_OFF         ;especificado en CONTATORE_1.
GOTO         INIZIO
;-----
;
;          TEMPORIZZAZIONE
;-----
DELAY_15 s    MOVLW        d'2
MOVWF        CONTATORE_2
TEMPO
NOP
DECFSZ       CONTATORE_2,f
GOTO         TEMPO
GOTO         FINE
FINE
RETURN
END

```

mente el Led) basta con intervenir en la primera instrucción que aparece inmediatamente tras las etiquetas T_On y T_Off: para que se arranque T_On con la instrucción BSF PORTB,0 habrá que escribir BCF PORTB,0, y en lugar de iniciar T_Off con la instrucción BCF PORTB,0, basta con escribir BSF PORT,0. Para actuar sobre la pendiente de las rampas o para modificar la duración del efecto crepuscular (es decir, para ralentizarlo o acelerarlo) deberá intervenir sobre la duración de la subrutina de temporización.