Solução por Software para Implementar PWM em qualquer Microcontrolador PIC

Prof. Francisco Fambrini

Universidade Anhanguera
Pólo Santa Bárbara do Oeste - SP

Abstract:
In this paper, the author worry in to describe a PWM Software Solution for the PIC12F675 and others PIC microcontrollers without hard-coded PWM.
This solution allow to build a digital pwm system embeeded at all PIC Microcontrollers chips.

Introdução:
Muitos dispostivos PIC possuem módulos PWM internos na própria pastilha, que possibilitam gerar sinais de pwm sem esforço computacional e sem firmware extra. Entretanto, todos os PICs mais antigos e alguns novos modelos (exemplo : PIC12F675) não possuem tal modulo PWM interno em sua pastilha.
Neste trabalho o autor pretende descrever um método para se implementar um Controlador PWM em qualquer microcontrolador PIC (Microchip), até mesmo em modelos que não possuem o modulo de PWM interno construído por hardware.

Conceitos:
PWM é uma sigla para Pulse Width Modulation, ou seja, Modulação por largura de pulsos.
É um método consagrado para controlar a energia (e consequentemente a potência) entregue à carga em dispositivos que trabalham em sistemas de Corrente Continua (DC).
Quanto maior a duração do Tempo Ligado (tempo em que o PWM permanece em On) daqui em diante denominado **Ton** neste trabalho, maior a Energia entregue à carga.

PWMs são por definição sistemas de frequência (F) e período (T) constantes e de Largura de Pulso (ciclo ativo) ajustável.

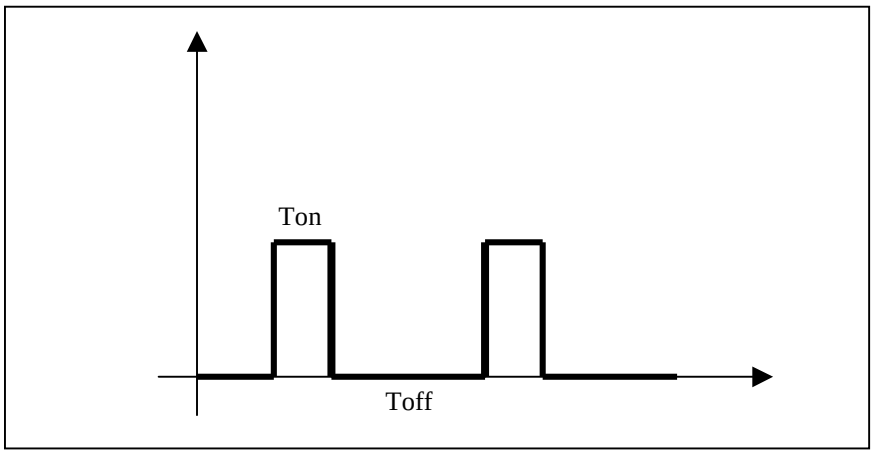
Denominando de **Toff** o tempo em que a carga é mantida desativada, o periodo total de nosso PWM será:

$$T = Ton + Toff$$

Define-se **Ciclo Ativo (Duty Cycle D.C.)** por:

$$D.C. = \frac{Ton}{Ton + Toff}$$

A figura abaixo ilustra a forma de onda típica de um sistema PWM digital:



A solução proposta foi testada e desenvolvida num microcontrolador PIC12F675 (que não possue módulo PWM interno) e baseia-se em firmware.

Nosso firmware foi escrito totalmente em Linguagem Assembly pois nossa intenção inicial de utilizar Linguagem C revelou-se incapaz de proporcionar a velocidade necessária de processamento para gerar o PWM, ler os dois botões (Up e Down) sem causar efeitos de "flicker" na onda retangular gerada.

Sobre velocidade de processamento, é importante frisar que optamos por utilizar o oscilador RC interno dos PIC12F675, cuja frequencia de clock é fixa e igual a 4 MHz.

Como os Microcontroladores PIC dividem internamente o clock por 4, temos um clock efetivo interno de valor 1 MHz o que nos dá um Ciclo de Máquina de duração 1 us (um microsegundo).

Com este ciclo de máquina, optamos por uma solução de software otimizada para a arquitetura do PIC, segundo nossa referência bibliográfica (1), o que nos proporcionou excelente desempenho no PWM.

O sistema utiliza-se da interrupção do TMR0 (Timer-Zero) cujo prescaler foi ajustado para 1:2, ou seja, a cada 2 us temos uma interrupção de timer 0 e o fluxo do programa será desviado para o endereço 0x04 (endereço do vetor de interrupção do PIC12F675).
Na rotina de interrupção temos uma estrutura de software da seguinte forma:

```
MOVF  pwmdesired , W
ADDWF PCL,F
bsf  LED
bsf  LED
bsf  LED
bsf  LED
bsf  LED
bsf  LED
.....
.....
.....
```

O registrador **pwmdesired** armazena o valor desejado de **Ton.** A seguir este valor é movido para o W e somado ao PCL que produz um salto relativo para a instrução que mantém o PWM pelo tempo correto na saída do microcontrolador.
O uso de interrupção do Timer0 nos possibilita independência para leitura dos botoes UP e DOWN sem prejudicar a geração da onda do PWM.
O botão UP incrementa o ciclo ativo e o botão DOWN decrementa.
O PWM inicializa-se sempre em zero, ou seja, ao restabelecer a energia, começa sempre desativado.

Outro detalhe é o uso dos resistores de pull-up internos ao Microcontrolador PIC em questão, o que nos dispensou de usá-los externamente. Também não foi utilizado nenhum oscilador de clock externo ao chip, nem cristal de quartzo, tendo-se optado pelo uso do oscilador RC interno ao chip como forma de minimizar a quantidade de componentes externos.
É importante tecer algumas considerações sobre o estágio de Potência. A escolha recaiu para um transistor MOS de Efeito de Campo (MOS-FET) da International Rectifier, o IRLZ 44 N. A letra "L" no prefixo IRLZ indica que seu gate é adequado para controles lógicos e que o mesmo pode ser saturado com tensões de 5 Volts, o que não ocorre com os tipos IRF ou IRFZ.
Nestes, um valor tão baixo de tensão pode não levá-lo à saturação, fazendo o Mos-Fet operar na região linear, o que iria provocar um considerável aquecimento do componente (ref. (6)).

Operando na condição de Corte e Saturação, o aquecimento é minimo (não ocorre perda de energia por dissipação de potência na junção Dreno – Source) .
Assim, o transistor poderá manipular correntes altissimas (até **47 A** segundo o datasheet do fabricante), exibindo (quando saturado) uma resistência DRENO-SOURCE (**Rds)** de apenas Rds = 0,022 ohms (6).
A tensão máxima admissível entre D e S é de 55 Volts, ainda de acordo com (6).
A escolha do resistor de Gate do mos-fet também é importante: a capacitancia interna entre G e S é elevada (da ordem de 1700 pF) e um resistor de valor elevado formaria com esta um RC com constante de tempo considerável, prejudicando o disparo rápido para corte e saturação deste transistor.
Por outro lado, o menor valor de resistor admissivel pelo PIC é de 200 ohms, pois os Microcontroladores PIC são especificados para uma corrente máxima de saída de 25 mA em 5 Volts de alimentação (ref. 7) .
O valor escolhido foi então de 200 ohms.
Faz-se também necessário acrescentar um diodo zener entre o gate do mos-fet e a porta de saída do PIC, para evitar que spikes rápidos de tensão venham a atingir o microcontrolador, o que provocaria a queima do mesmo. O diodo zener (1N4733A) foi inserido entre um resistor de 22 ohms e outro de 180 ohms associados em série.

CONCLUSÃO:
Este software é capaz de produzir na saída de qualquer Microcontrolador um PWM de ciclo ativo variável digitalmente através de 2 botões, com 255 passos de ajuste (256 steps, incluindo o valor 0).
O consumo de memória do processador é inferior a 200 words (cerca de 20% da capacidade de um PIC12F675). A Frequência do PWM gerado gira em torno de 2,2 KHz.
O período medido é de aproximadamente 450 us. O uso de um MOS-FET lógico de potência (IRLZ 44) da International Rectifier garante que elevadas correntes podem ser controladas, o que torna este dispositivo ideal para controle de luminosidade de lâmpadas e de velocidade de motores DC.

REFERÊNCIAS:

(1)- **Ropcke, Ole**. AN654, Aplicattion Note da Microchip (www.microchip.com)
(2)- **Zanco, Wagner.** Microcontroladores PIC, Ed. Érica, 3 Edição
(3)- **Souza, David José**. Desbravando o PIC, 6 Edição, Ed. Érica
(4)- **Pereira, Fabio**. Microcontroladores PIC Técnicas Avançadas, Ed. Érica.
(5)- **Fambrini, Francisco** Apostila sobre Linguagem Assembly, INTEP, edição do Autor.
(6)- **International Rectifier**, Datasheet do transistor IRLZ 44
(7)- **Microchip**, Datasheet do microcontrolador PIC12F6xx

```
;*****************************************************
;            PWM BOTOES - PWM_BOT.ASM                 *
;         PWM por software para PIC12F675             *
;          VARIAVEL  PWMDESIRED CONTROLA              *
;                    O NIVEL DE PWM                   *
;        Escrito em Assembly em 15/Fev/2006           *
;;*****************************************************
#include <p12f675.inc>

__CONFIG 314Ch

#define BANK0  BCF STATUS,RP0
#define BANK1  BSF STATUS,RP0

;*****************************
;variaveis do programa
;*****************************
CBLOCK 0X20
STACKW
STACKS
COUNTER
COUNTER2
PWMDESIRED
PWMMAX
PWMHELP
MAX     ;VALOR MAXIMO DO PWMDESIRED
MIN     ;VALOR MINIMO
FILTRO1  ;filtros dos botoes
FILTRO2
FLAGS
ENDC
;*****************************
; Constantes usadas no programa:
    PWMADJUSTVAL EQU .22
    PWMMAXVAL EQU .29
;*****************************
#define LDR    GPIO,0
#define BT1    GPIO,1
#define BT2    GPIO,2
#define JUMPER GPIO,4
#define LED    GPIO,5

org 0x00
goto power_on

;***********************************
;Endereco inicial da interrupcao
;***********************************
org 0x04
btfsc TMR0,0
GOTO PwmInt

PwmInt:
movwf STACKW
SWAPF STACKW,F
```

```
SWAPF STATUS,W
MOVWF STACKS
BCF INTCON,T0IF
BTFSC LED
GOTO LOWPULSE

HIGHPULSE:
COMF PWMDESIRED,W
MOVWF PWMHELP
ADDWF PWMMAX,F
BTFSS STATUS,C
GOTO HIGHIMPINT

HIGHIMPSHRT:
MOVF PWMMAX,W
ADDWF PCL,F
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BSF LED
BCF LED

INCF COUNTER,F
COMF PWMHELP,W
ADDLW PWMADJUSTVAL+5
MOVWF TMR0
GOTO LOWIMPINT2

HIGHIMPINT:
ADDLW PWMADJUSTVAL
MOVWF TMR0
```

```
HIGHIMPINT2:
BSF LED
INCF COUNTER,F

MOVLW PWMMAXVAL-1
MOVWF PWMMAX
SWAPF STACKS,W
MOVWF STATUS
SWAPF STACKW,W
RETFIE
;*****************************

LOWPULSE:
COMF PWMHELP,W
ADDWF PWMMAX,F
BTFSS STATUS,C
GOTO LOWIMPINT

LOWIMPSHRT:
MOVF PWMMAX,W
ADDWF PCL,F
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BCF LED
BSF LED

COMF PWMDESIRED,W
MOVWF PWMHELP
ADDLW PWMADJUSTVAL+5
MOVWF TMR0
GOTO HIGHIMPINT2
```

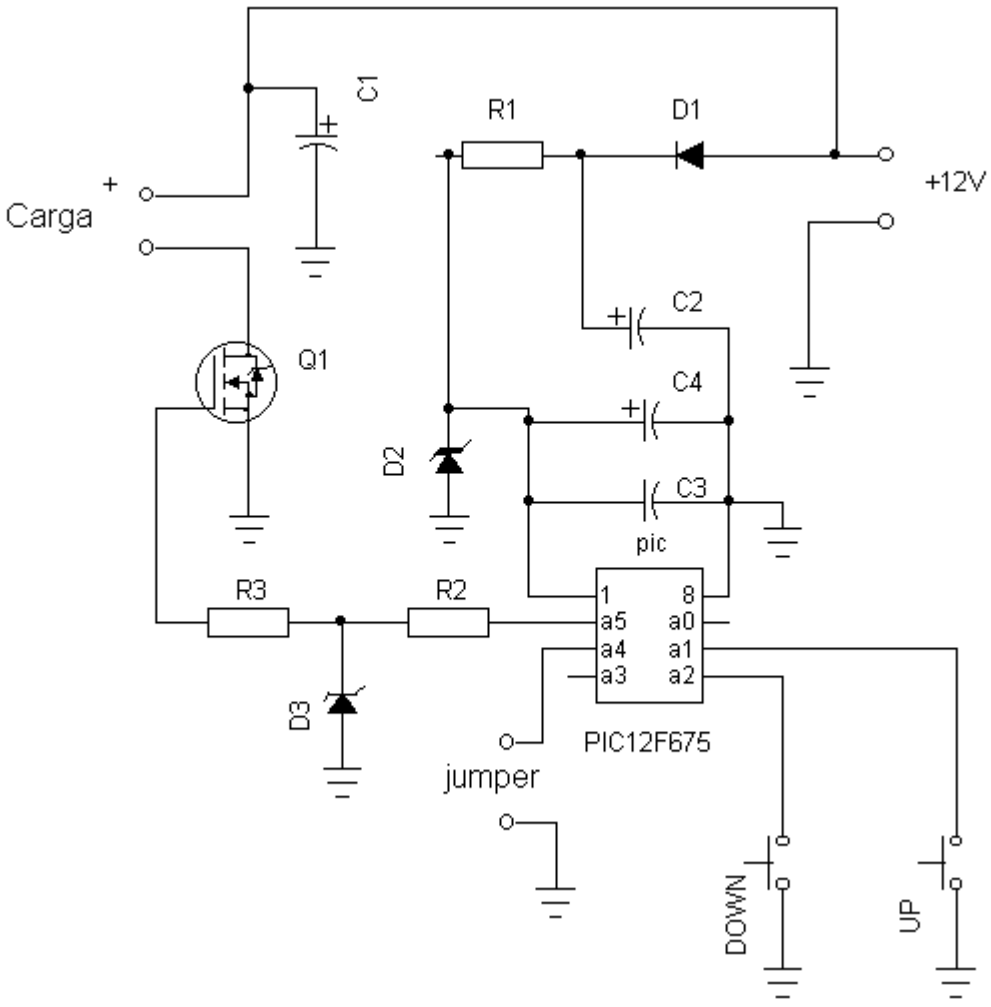
```
LOWIMPINT:
ADDLW PWMADJUSTVAL
MOVWF TMR0

LOWIMPINT2:
BCF LED
MOVLW PWMMAXVAL
MOVWF PWMMAX
SWAPF STACKS,W
MOVWF STATUS
SWAPF STACKW,W
RETFIE


;*****************************************************************
;         inicio do programa propriamente
;*****************************************************************

power_on:           ;ajustes iniciais

clrf TMR0
CLRF PWMDESIRED
BCF LED
MOVLW PWMMAXVAL
MOVWF PWMMAX

;*****************************************************************
;                 Configuração do PIC
;*****************************************************************
BANK1               ; ajusta oscilador interno para 4MHz
CALL 3FFh
MOVWF OSCCAL

BANK0
CLRF GPIO
MOVLW 07h
MOVWF CMCON          ; desliga comparadores analógicos

BANK1
CLRF ANSEL   ;todos I/Os digitais
MOVLW 1Fh
MOVWF TRISIO ;configura as saidas e entradas
MOVLW 16h
MOVWF WPU    ;configura os resitores de pull-up

MOVLW B'00000000'   ; ajusta o OPTION_REG, TMR0 com divisor 1:2
MOVWF OPTION_REG    ; habilita os pull-up

MOVLW B'10100000'   ;ajusta o INTCON
MOVWF INTCON        ;liga interrup do TMR0 e liga pull-ups

BANK0
;*****************************************

Idle:
clrwdt
```

```
btfss COUNTER,07h
goto Idle
bcf COUNTER,07h

;**********************
;Controle do PWM
;**********************
MOVLW .0
MOVWF PWMDESIRED  ;valor inicial do PWM
MOVWF MIN         ;valor minimo do pwm
MOVLW .250        ;valor maximo do pwm
MOVWF MAX


CONTROLE:
clrwdt
;*******************
;Checa o botao 1:
;*******************
BTFSS BT1
GOTO AUMENTAR
;*******************
;Checa o botao 2:
;*******************
BTFSS BT2
GOTO DIMINUIR
goto CONTROLE
;*******************

;*********************************************************************
AUMENTAR:
MOVF MAX,W              ; verifica se o valor de pwmdesired ja esta no maximo
XORWF PWMDESIRED,W
BTFSC STATUS,Z
GOTO CONTROLE           ; se pwmdesired=MAX volta para CONTROLE
INCF PWMDESIRED,1       ; caso contrario, incrementa a variavel pwmdesired
CALL DELAY
GOTO CONTROLE
;*********************************************************************

;*********************************************************************
DIMINUIR:
MOVF MIN,W
XORWF PWMDESIRED,W
BTFSC STATUS,Z
GOTO CONTROLE
DECF PWMDESIRED,1
CALL DELAY
GOTO CONTROLE
;*********************************************************************

;*************************
; Rotina de delay
;*************************
DELAY:
movlw .30
```

```
movwf FILTRO2
DL1:
clrwdt
movlw .255
movwf FILTRO1
DL2:
clrwdt
NOP
DECFSZ FILTRO1,F
GOTO DL2
DECFSZ FILTRO2,F
GOTO DL1
RETURN
;*************************
END
```

LISTA DE MATERIAL

Circuitos Integrados	
c.i. 1	PIC 12F675 - Microcontrolador Microchip
Resistores	todos 1/8 Watt
R1	470 ohms
R2	22 ohms
R3	180 ohms
Capacitores	
C1	470 uF, 25V, cap eletrolítico
C2	220 uF, 25V, cap eletrolitico
C3	100nF, 16V, cap. cerâmico disco
C4	100uF, 16 V, cap eletrolitico
Diodos	
D1	1N4007, diodo de Silicio
D2	1N4733A, diodo zener 5.1 V, 1 watt
D3	1N4733A, diodo zener 5.1 V, 1 watt
chaves	
S1 e S2	chaves tipo push button para circuito impresso
Transistor	
Q1	Transistor Hex Power Mos-Fet IRLZ 44 N



Controlador PWM com PIC12F675