

CONTROL MOTOR DE CONTINUA

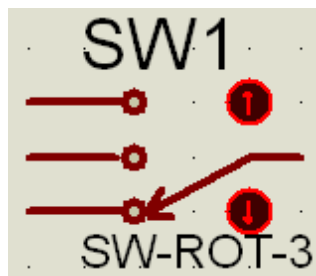
Lo que queremos hacer es un circuito que controle la velocidad de estos tipos de motores, la velocidad sera controlada por medio de una rueda que tendra tres estados, como si fueran tres interruptores, estas combinaciones haran que el motor vaya al 100% de su capacidad, otra posición hara que vaya al 75%, y la otra posición hara que vaya al 25%.

Frecuentemente, se intenta controlar un motor DC mediante una resistencia variable conectada a un transistor. Si bien este sistema funciona, genera gran cantidad de calor y perdidas de potencia. Controlando la velocidad de un motor DC mediante la modulación por ancho de pulso se elimina ese problema. Cuanto más ancho es el pulso, mayor es la velocidad del motor y viceversa.

Como nosotros vamos a hacer el control de este motor mediante un pic, no podremos sacar del pic una señal distintas a las establecidas para la tecnología TTL, con lo que jugaremos con la duracion de los estados altos o bajos que salgan de este, es decir, según dejemos que salga del pic un estado alto un determinado tiempo (estando el tiempo restante en estado bajo), haremos que vaya mas o menos rápido. Por ejemplo si queremos que el motor vaya al 100% de la velocidad que tenga le daremos un estado alto todo lo que dure su periodo y asi tendra de media de tension la maxima, que es la tension que necesita para ir al 100% de su capacidad, estos motores toman como tension la media de entrada. Este control se efectua dependiendo de la duracion del periodo que tenga el motor. El que vamos a utilizar tiene un periodo de 10ms, y la tabla que efectuamos es la siguiente:

Posición uno	
100%	
Posición dos	
75%	
Posición tres	
25%	

Estas posiciones se refieren a las posiciones de la rueda



Siendo la posición uno la posición de mas arriba(100%) y la tres la de abajo(25%)

Para poder controlar los pulsos que salen del pic al motor hemos hecho un programa en MPLab con el que controlaremos la velocidad:

;Motor en continua

,*****

List p=16f84

include"P16f84.inc"

PDel0 equ 0x15 ; nombramos las variables

PDel1 equ 0x16

PDel2 equ 0x17

PDel3 equ 0x18

PDel4 equ 0x19

PDel5 equ 0x20

PDel6 equ 0x21

PDel7 equ 0x22

PDel8 equ 0x23

PDel9 equ 0x24

org 0x00 ; punto de origen

goto Inicio

org 0x05; saltamos el vector de interrupción

,*****

Creamos las demoras para controlar el tiempo que va estar el pulso(1) o el pulso (0)

,*****

DEM25A movlw .7 ; 1 set numero de repeticion (B)

movwf PDel0 ; 1 |

PLoop1 movlw .88 ; 1 set numero de repeticion (A)

movwf PDel1 ; 1 |

PLoop2 clrwdt ; 1 clear watchdog

decfsz PDel1, 1 ; 1 + (1) es el tiempo 0 ? (A)

```
goto PLoop2 ; 2 no, loop
decfsz PDeI0, 1 ; 1 + (1) es el tiempo 0 ? (B)
goto PLoop1 ; 2 no, loop
PDeLL1 goto PDeLL2 ; 2 ciclos delay
PDeLL2 clrwdt ; 1 ciclo delay
return ; 2+2 Fin.
```

```
,*****
```

```
DEM100 movlw .8 ; 1 set numero de repeticion (B)
movwf PDeI8 ; 1 |
PLoop9 movlw .249 ; 1 set numero de repeticion (A)
movwf PDeI9 ; 1 |
PLoop10 clrwdt ; 1 clear watchdog
clrwdt ; 1 ciclo delay
decfsz PDeI9, 1 ; 1 + (1) es el tiempo 0 ? (A)
goto PLoop10 ; 2 no, loop
decfsz PDeI8, 1 ; 1 + (1) es el tiempo 0 ? (B)
goto PLoop9 ; 2 no, loop
PDeLL5 goto PDeLL6 ; 2 ciclos delay
PDeLL6 clrwdt ; 1 ciclo delay
return ; 2+2 Fin.
```

```
,*****
```

```
DEM75A movlw .6 ; 1 set numero de repeticion (B)
movwf PDeI4 ; 1 |
PLoop5 movlw .249 ; 1 set numero de repeticion (A)
movwf PDeI5 ; 1 |
PLoop6 clrwdt ; 1 clear watchdog
```

```

clrwdt ; 1 ciclo delay

decfsz PDe15, 1 ; 1 + (1) es el tiempo 0 ? (A)

goto PLoop6 ; 2 no, loop

decfsz PDe14, 1 ; 1 + (1) es el tiempo 0 ? (B)

goto PLoop5 ; 2 no, loop

clrwdt ; 1 ciclo delay

return ; 2+2 Fin.

```

Inicio

```

bsf STATUS, RP0; vamos al banco correspondiente con los TRIS

movlw b'00000111' ; definimos entradas y salidas en TRISA

movwf TRISA

clrf TRISB

bcf STATUS, RP0 ;cambio de banco

clrf PORTB

btfss PORTA,0

goto velo_100 ; vamos a una de las rutinas que hemos creado( rutina del 100% de ;velocidad)

btfss PORTA,1

goto velo_75

btfss PORTA,2

goto velo_25

goto Inicio

velo_100

bsf PORTB,0

call DEM100 ;tiempo que va a estar a estado alto

btfsc PORTA,0

```

```

goto Inicio

goto velo_100

velo_75

bsf PORTB,0

call DEM75A ; tiempo que va a estar a estado alto (7,5 ms)

bcf PORTB,0

call DEM25A ; tiempo que va estar a estado bajo ( 2,5ms)

btfsc PORTA,1

goto Inicio

goto velo_75

velo_25

bsf PORTB,0

call DEM25A ; tiempo a estado alto( en este caso 2,5 ms)

bcf PORTB,0

call DEM75A

btfsc PORTA,2 ; tiempo a estado bajo ( 7,5 ms)

goto Inicio

goto velo_25

end ; fin del programa

```

Este programa solo funcionara para un tipo de interruptor como el que ponemos en el esquema, ya que si pusieramos tres interruptores de los normales, el programa entraria en error, o funcionamiento indeterminado.

Para simularlo hemos utilizado el programa Proteus, y el esquema es el siguiente:

T

7.5ms

2.5ms

Estado

bajo

7.5ms

2.5ms

T

V

Estado

alto

Duracion completa del periodo 10 ms