
Laboratorio de Diseño de Robots Móviles
Practica No. 5
Generación de PWM

Objetivo: Utilizar el sistema de PWM de un microcontrolador para controlar la operación de los motores de un robot.

Desarrollo: Para cada uno de los siguientes apartados, realizar los diseños electrónicos y programas que se piden.

Duración: Dos semanas.

1. Busque en la ayuda del compilador CCS C información sobre las siguientes funciones las cuales configuran pines del puerto c del PIC como salidas PWM:

```
setup_ccp1(CCP_PWM)
setup_ccp2(CCP_PWM)

set_pwm1_duty(duty)
set_pwm2_duty(duty)

setup_timer_2(T2_DIV_BY_1, 127, 10)
enable_interrupts(INT_TIMER2)
```

2. Tomando como base el programa en C que se encuentra en el apéndice 1 controle la velocidad de los motores del robot móvil que se ha construido en las prácticas anteriores. Conecte las líneas de la etapa de potencia que controlan los motores izquierdo y derecho a los pines 1 y 2 del puerto C los cuales generan las salidas PWM.

Apéndice 1

```
#include <16f877.h>
#device ADC=8
#include <stdlib.h>

#fuses HS,NOPROTECT
#use delay(clock=2000000)
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1FFF, 0x1FFF void loader16F877(void){}

// Inicializa puertos
inicializa_puertos(){

    setup_ccp1(CCP_PWM);
    setup_ccp2(CCP_PWM);
}

// Inicializa interrupciones
```

```
inicializa_temporizadores(){  
  
    // The cycle time will be (1/clock)*4*t2div*(period+1)  
    // In this program clock=10000000 and period=127 (below)  
    setup_timer_2(T2_DIV_BY_1, 127, 10);  
}
```

```
// Se indica el tamaño del periodo positivo del pwm  
mv(long duty){  
  
    // Se indica la direccion de movimiento de las llantas  
    output_d(0x0a);  
  
    // Se indican los tiempos del pwm  
    set_pwm1_duty(duty);  
    set_pwm2_duty(duty);  
}
```

```
// Programa principal  
void main(){  
  
    char linea[20];  
    long duty;  
  
    // Se inicializan las interrupciones  
    inicializa_temporizadores();  
  
    // Se inicializan los puertos  
    inicializa_puertos();  
  
    // Loop infinito  
    while(TRUE){  
  
        // Se lee el tamaño del periodo positivo  
        printf("\n\r-> ");  
        gets(linea);  
        duty=atol(linea);  
  
        // Se indica el tamaño del periodo positivo del pwm  
        mv(duty);  
  
    }  
}
```