
Laboratorio de Diseño de Robots Móviles

Practica No. 6

Temporizadores

Objetivo: Utilizar el sistema de temporizadores de un microcontrolador, para capturar las formas de onda generadas por los encodificadores de los motores de un robot móvil.

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

Duración: 2 semanas.

1. Busque en la ayuda del compilador CCS C información sobre las siguientes funciones de los temporizadores:

```
setup_timer_0()
setup_timer_1()
setup_timer_2()
enable_interrupts()
```

2. Tomando como base el programa en C que se encuentra en el apéndice 1 varíe los tiempos en los cuales ocurren las interrupciones de los temporizadores.
3. Coloque a un lado de las llantas un fondo negro y blanco para detectar cuanto han girado estas usando sensores infrarrojos, como se muestran en la figura 1. Conecte estos encodificadores a las pines C0 y A4 del PIC. Programe los temporizadores T0 y T1 con relojes externos dados por las líneas de los encodificadores, como se muestra en el apéndice 2, el cual muestra cuando se detecta una transición del fondo negro y blanco.

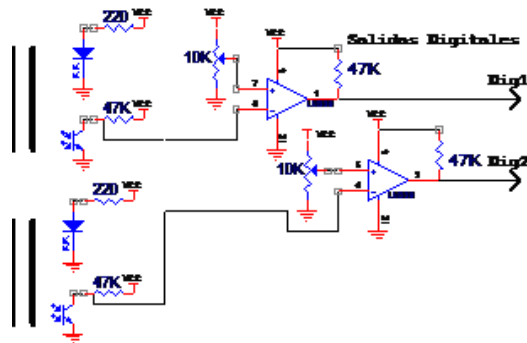


Figura 1. Generación de una señal digital con sensores reflectivos

4. Utilice el temporizador T2 para mostrar los contadores de los temporizadores T0 y T1, como se muestra en el apéndice 3.

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){}

// Esta interrupcion es llamada en forma automatica por el TIMER0.
#INT_TIMER0
void timer0() {
    static long i=0,j=0;
```

```

set_timer0(0x00);

i++;
if(i>1024){
    j++;
    printf("T0 %ld\n\r",j);
    i=0;
    if(j>100)j=0;
}
}

// Esta interrupcion es llamada en forma automatica por el TIMER1.
#INT_TIMER1
void wave_timer1() {
    static long i=0,j=0;

    set_timer1(0xF000);

    i++;
    if(i>1024){
        j++;
        printf("T1 %ld\n\r",j);
        i=0;
        if(j>100)j=0;
    }
}

// Esta interrupcion es llamada en forma automatica por el TIMER2.
#INT_TIMER2
void wave_timer2() {
    static long i=0,j=0;

    i++;

    if(i>4096){
        j++;
        printf("T2 %ld\n\r",j);
        i=0;
        if(j>100)j=0;
    }
}

// Inicializa interrupciones
inicializa_interrupciones(){

    setup_timer_0(T1_INTERNAL|T1_DIV_BY_1); // setup interrupt 0
    enable_interrupts(INT_TIMER0);

    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1); // setup interrupt 1
    enable_interrupts(INT_TIMER1);

    setup_timer_2( T2_DIV_BY_1, 128, 10); // setup interrupts 2
    enable_interrupts(INT_TIMER2);

    enable_interrupts(GLOBAL);
}

void main() {

    inicializa_interrupciones();

    while(TRUE);
}

```

```
}
```

Apéndice 2

```
#include <16f877.h>
#device ADC=8
//#include <stdio.h>
#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){}

// Definicion de Constantes
#define NUM_PULSES_0 0xFF
#define NUM_PULSES_1 0xFFFF

//Variables Globales
static long CntRight=0,CntLeft=0;

// Esta interrupcion es llamada en forma automatica por el TIMER0.
#INT_TIMER0
void timer0() {
    static long i=0;

    CntLeft++;
    printf("T0 %x\n\r",CntLeft);
    set_timer0(NUM_PULSES_0);
}

// Esta interrupcion es llamada en forma automatica por el TIMER1.
#INT_TIMER1
void wave_timer1() {
    static long i=0;

    CntRight++;
    printf("T1 %ld\n\r",CntRight);
    set_timer1(NUM_PULSES_1);
}

// Inicializa interrupciones
inicializa_interrupciones(){

    set_timer1(NUM_PULSES_1);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
    enable_interrupts(INT_TIMER1);

    set_timer0(NUM_PULSES_0);
    setup_timer_0(RTCC_DIV_1|RTCC_EXT_L_TO_H);
    enable_interrupts(INT_TIMER0);

    enable_interrupts(GLOBAL);
}

void main(){

    inicializa_interrupciones();

    while(TRUE);
}
```

Apéndice 3

```
#include <16f877.h>
#include <adc.h>
#include <stdlib.h>

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

// Definicion de Constantes
#define NUM_TIMES_T2 256
#define NUM_SECOND 14

// Esta interrupcion es llamada en forma automatica por el TIMER2.
#define INT_TIMER2
void wave_timer2() {
    static long k=0,l=0,m=0;
    long i,j;

    k++;
    i=get_timer0();
    j=get_timer1();

    if(k>NUM_TIMES_T2){
        set_timer0(0);
        set_timer1(0);
        k=0;
        l++;
        if(l>NUM_SECOND){
            printf("T2 Temp. %ld motor Iz. %lu De. %lu\n\r",m,i,j);
            l=0;
            m++;
        }
    }
}

/* Inicializa interrupciones*/
void inicializa_interrupciones(){

    set_timer1(0);
    setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);

    set_timer0(0);
    setup_timer_0(RTCC_DIV_1|RTCC_EXT_L_TO_H);

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

    enable_interrupts(GLOBAL);
}

// Programa Principal
void main(){

    char linea[20];
    long duty;

    inicializa_interrupciones();

    while(TRUE);
}
```