

## Práctica 5

### • Interrupciones temporizadas y E/S mapeada a memoria

**Objetivo:** Uso de temporizadores para la implementación de un reloj simple con acceso a un puerto como indicador.

**Material:**

- Programas TCC, TASM, TLINK y MKBINTJ
- Tarjeta T-Juino.
- Programa Terminal.
- Protoboard y compuertas lógica (según diseño).

**Equipo:** - Computadora Personal

**Teoría:** \* \* \* Teoría sobre los vectores de interrupción 08h y 1Ch de la PC \* \* \*

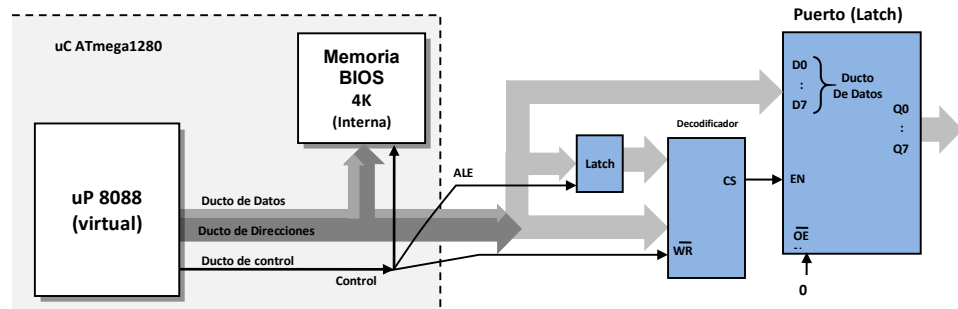


Figura 1. Esquema de puerto de salida mapeado en memoria.

**Desarrollo:** Crear un programa basado en los archivos Timer.h, Timer.c y Prac5.c

#### Listado 1:Timer.h

```
unsigned char TimerSecFlag( void );
void Timer_Ini( void );
```

#### Listado 2:Timer.c

```
#define INTR 8 /* Utilizar el vector adecuado */
volatile static unsigned char SecFlag; /*Bandera para indicar el Seg. */

/* Manejador de Interrupcion: ocurre 20 veces por segundo */
void interrupt ManejadorISR( void ){
    static unsigned char Cont = 0;
    Cont++; /* Conteo de Ticks */
    if(Cont == 20){
        SecFlag=1;
        Cont = 0;
    }
}

unsigned char TimerSecFlag( void ){
    if( SecFlag == 1 ){
        SecFlag = 0;
        return 1;
    }else
        return 0;
}

void Timer_Ini( void ){
    SecFlag=0;

    /* Inicializarvector --faltapokew(WORD segment, WORD offset, WORD value) -- */
    pokew(0x0,INTR*4,offset de la ISR); /* Determinar offset */
    pokew(0x0,INTR*4+2,_CS);
}
```

## Listado 3: Prac5.c

```

#include "Timer.h"
#define BYTE unsigned char
void UpdateClock( void );
void DisplayClock( void );
void PrintDec2d( BYTE );
void Delay( BYTE );

BYTE seg=0; /* para el manejo de los segundos */

void main(){
    Timer_Ini(); /* inicializaTimer de 1 segundo */
    while(1){
        /* Verificación para actualizar el reloj */
        if(TimerSecFlag()){ /* Ha pasado un segundo? */
            UpdateClock(); /* actualiza y muestra el reloj */
            DisplayClock(); /* desplegar reloj en 0,0 */
            /* indicar en puerto */
        }
        /* otras cosas por hacer */
        Delay(100);
        putchar('.'); /* Imprime */
    }

    void UpdateClock( void ){
        seg=(++seg%60);
        /*falta minutos , horas etc. etc. */
    }

    void DisplayClock( void ){
        /* imprimir reloj en formato "hh:mm:ss" */;
        PrintDec2d(seg); /* solo se esta desplegando segundos */
    }

    void Delay( BYTE i){
        while(--i);
    }

    void PrintDec2d( BYTE dato){
        putchar( dato/10 +'0'); /* Imprime decenas */
        putchar( dato%10 +'0'); /* Imprime unidades */
    }

```

**Actividades** (diseño e implementación)Hardware:

- Circuito para incluir a T-Juino un puerto de salida mapeado en memoria en la dirección XXXXH.

Software:

- Realice los cambios necesarios a las funciones **UpdateClock()** y **DisplayClock()**. La función **UpdateClock()** tiene como tarea actualizar el reloj; es decir una vez que es ejecutada esta incrementa los segundos para entonces actualizar los minutos, y horas del reloj. Para el caso de **DisplayClock** tiene la tarea de desplegar en pantalla el reloj en el formato militar (24 Hrs) "**hh:mm:ss**".
- Diseñar e implementar la función **SetClock()** la cual recibe como parámetro la hora, minuto y segundo (formato 24 Hrs) con el fin de inicializar el reloj.

```
void SetClock( BYTE Hora, BYTE minutos, BYTE segundos)
```

- Diseñar e implementar la secuencia apropiada para escribir un byte a puerto de salida mapeado en la dirección XXXXH de memoria.

**Comentarios y Conclusiones.****Bibliografía**