

Práctica 8a

Programación del uC en lenguaje C y comunicación serie.

Objetivo: Mediante esta práctica el alumno aprenderá el uso básico de la programación en lenguaje C con las herramientas AVR Studio y WinAVR. Para ello el alumno implementará los procedimientos comunes para inicializar y operar el puerto serie del microcontrolador.

Equipo:

- Computadora Personal
- Módulo T-Juino

Teoría:

- Programación en lenguaje C en microcontroladores
- Manejo del Periférico de Comunicación Serie 0 (UART0) del microcontrolador ATmega1280/2560

Descripción:

Haciendo uso de los pasos descritos en la Practica 6 crear un nuevo proyecto pero esta vez para hacer uso del lenguaje C con base en el **Listado 1** (Prac8.c).

Listado 1: Prac8.c

```
#include <avr/io.h>

void delay( void ); /* funcion prototype de Delay */

int main( void )
{
    /* configurar de salida el bits 7 de PORTB */
    DDRB = (1<<PB7); /* DDRB= 1000-0000 */

    while(1){

        PORTB |= ( 1 << PB7 ); /* encender LED */
        delay();

        PORTB &= ~( 1 << PB7 ); /* apagar LED */
        delay();

    }

}

void delay( void)
{
    volatile unsigned int i;

    for(i=0;i<0xffff;i++);
}
```

El proyecto deberá ser compilado y luego cargado al módulo para después verificar su operación correcta que es encender y apagar el LED L continuamente.

Actividades a realizar:

1. Calcular el periodo total de la función **Delay (void)** del Listado 1 con optimización *Os*.
2. Implementar las siguientes funciones para la comunicación serie mediante encuesta (polling):
 - a) void **UART0_Init** (uint16_t mode): Función para inicializar el puerto serie del ATmega1280/2560 según el valor del parámetro modo. Si mode es 0 entonces la inicialización es **9600,8,N,1** de lo contrario **19200,8,N,1**.
 - b) char **UART0_getchar** (void): Función que retorna el byte recibido por el puerto serie UART0.
 - c) void **UART0_putchar** (char data): Función que transmite un byte por el puerto serie UART0.
 - d) void **UART0_gets** (char *str): Función que retorna una cadena mediante **UART0_getchar**, la cadena se retorna en el apuntador str.
 - e) void **UART0_puts** (char *str): Función que transmite una cadena mediante **UART0_putchar**.
 - f) void **itoa** (char* str, uint16_t number, uint8_t base): Función que convierte un número de 16 bits a su representación alfanumérica en la base dada, y la retorna en el apuntador str.
 - g) unsigned int **atoi** (char *str): Función que convierte una cadena numérica (de base 10) y retorna su valor numérico en 16 bits.

Estas funciones deberán estar en su propio modulo, con sus prototipos y definiciones dadas por **UART.h** y **UART.c** respectivamente. Probar las funciones elaboradas con el Listado 2.

Listado 2: Prac8.c

```
#include <avr/io.h>
#include "UART.h"

int main( void )
{
    char cad[20];
    uint16_t num;
    UART0_Init(1);
    while(1) {
        UART0_getchar();
        UART0_puts("\n\rIntroduce un número:\n\r");
        UART0_gets(cad);
        num = atoi(cad);
        itoa(cad,num,16);
        UART0_puts("\n\rHex:");
        UART0_puts(cad);
        itoa(cad,num,2);
        UART0_puts("\n\rBin:");
        UART0_puts(cad);
    }
}
```

Comentarios y Conclusiones.

Bibliografía.