

Práctica 6

Programación del uC del periférico de comunicación serie.

Objetivo: Mediante esta práctica el alumno aprenderá el uso básico para inicializar y operar, bajo un esquema de interrupciones, el puerto serie del microcontrolador.

Equipo: - Computadora Personal

- Módulo T-Juino

Teoría: - Manejo del Periférico de Comunicación Serie 0 (UART0) del microcontrolador ATmega1280/2560

- Secuencias de escape ANSI.

Actividades a realizar:

1. Implementar las siguientes funciones:

- a) void **UART_putchar**(uint8_t com, char data):
Función que coloca el dato a enviar por el periférico.
- b) uint8_t **UART_available**(uint8_t com):
Función que retorna 1 si existe dato disponible en el periférico.
- c) char **UART_getchar**(uint8_t com): Función que retorna el dato recibido por el periférico. Si no existe, entonces espera hasta recibir uno.
- d) void **UART_gets**(uint8_t com, char *str)
Función que retorna una cadena haciendo uso de *UART_getchar(uint8_t com)*, la cadena se retorna en el apuntador *str*.
- e) void **UART_puts**(uint8_t com, char *str)
Función que imprime una cadena mediante *UART_putchar(uint8_t com)*.
- f) void **itoa**(uint16_t number, char* str, uint8_t base)
Función que convierte una numero de 16 bits a su representación ASCII en la base especificada.
- g) uint16_t **atoi**(char *str)
Función que convierte una cadena de un valor decimal a un numero entero de 16 bits.
- h) **UART_Ini**(uint8_t com, uint16_t baudrate, uint8_t size,
uint8_t parity, uint8_t stop)

Función que inicializa el periférico del UART en un esquema de interrupciones. Y la configuración es dada por los parámetros, donde:

- **com**: representa el número de UART a configurar. Considerar los cuatro posibles puertos.
- **baudrate**: representa la velocidad en Baud de configuración, puede ser no estándar.
- **size**: representa el número de bits de los datos con los que operará el UARTx. Considerar de 5 a 8 bits.
- **parity**: representa el tipo de paridad con los que operará el UARTx. Considerar 0: No paridad, 1: impar, 2: par.
- **stop**: representa el número de bits de paro con los que operará el UARTx. Considerar 1 ó 2.

4. En base a las secuencias de escape, implementar las siguientes funciones:

a) void **UART_clrscr**(uint8_t com)

Funcion que limpia la terminal mediante la secuencia de escape.

b) void **UART_setColor**(uint8_t com, uint8_t color)

Función que envía la secuencia de escape para configurar el color del texto que se desplegará en la terminal.

c) void **UART_gotoxy**(uint8_t com, uint8_t x, uint8_t y)

Función que posiciona el cursor en la terminal en la coordenada x,y que lleguen como parámetro, utilizando la secuencia de escape.

2. Utilizar el código de repositorio dado para comprobar el funcionamiento de las funciones.

Comentarios y Conclusiones.

Bibliografía.