Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**MICROCONTROLADORES**

**Práctica No. 6**

**Programación del Microcontrolador del Periférico de Comunicación Serie**

**Docente:** Castro Gonzalez Ricardo

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 01261509

# Objetivo:

Mediante esta práctica el alumno aprenderá el uso básico para inicializar y operar el puerto serie del microcontrolador

# Material:

* **Computadora personal**

# Teoría:

* **Manejo del Periférico de Comunicación Serie 0 (UART0)b del microcontrolador ATMega2560**

El manejo del periférico de comunicación serie UART0 en el ATMega2560 es una parte esencial para establecer comunicación serie con otros dispositivos, como módulos GPS, otros microcontroladores, sensores y más.

Para poder utilizar el UART0, es necesatio configurar los parametros necesarios, como lo son, la velocidad de transmisión (baudrate), numero de bits de datos, bits de paro, si hay paridad y entre otras configuraciones. Todo esto se logra editando los registros necesarios, ejemplos:

void UART\_Init() {

// Configurar la velocidad de transmisión a 9600 bauds

UBRR0H = (unsigned char)(103 >> 8);

UBRR0L = (unsigned char)(103);

// Habilitar la transmisión y recepción UART0

UCSR0B = (1 << TXEN0) | (1 << RXEN0);

// Establecer formato de datos: 8 bits de datos, sin paridad, 1 bit de parada

UCSR0C = (3 << UCSZ00);

}

* **Para enviar y recibir datos pueden ser usadas las siguientes funciones**

void UART\_Transmit(unsigned char data) {

// Esperar hasta que el búfer de transmisión esté vacío

while (!(UCSR0A & (1 << UDRE0)));

// Colocar el dato en el registro de transmisión

UDR0 = data;

}

unsigned char UART\_Receive() {

// Esperar hasta que se reciba un dato

while (!(UCSR0A & (1 << RXC0)));

// Devolver el dato recibido

return UDR0;

}

* **Secuencias de escape ANSI**

Los códigos de escape se utilizan para dar formato a la salida en la terminal de texto y se basan del estandar ANSI x3.64 (tambien llamado ECMA-48)

|  |  |
| --- | --- |
| **Secuencia** | **Acción** |
| ESC [ *n* A | Desplaza el cursor hacia arriba *n* filas. Si el cursor se encontraba en la parte superior de la pantalla, no tenía ningún efecto. Si no se especificaba *n* el cursor subía una fila. |
| ESC [ *n* B | Desplaza el cursor *n* filas hacia abajo. Al igual que con el desplazamiento hacia arriba, si el cursor se encontraba en la parte inferior de la pantalla el cursor no se movía, y si no se especificaba *n* bajaba una fila. |
| ESC [ *n* C | Mueve el cursor *n* columnas hacia la derecha. Si el cursor se encuentra en la última columna este comando no tiene efecto. Si no se especifica *n* el desplazamiento es de una columna. |
| ESC [ *n* D | Mueve el cursor *n* columnas a la izquierda, salvo que se encuentre en la primera columna, en cuyo caso no tiene efecto. Si *n* no se especifica toma el valor de 1. |
| ESC [ *n*; *m* f | Mueve el cursor a la fila *n* y columna *m*. Si *n* no se especifica el cursor se mueve a la primera fila. |
| ESC [ *n* J | Borra parte de la pantalla. Si *n* vale 0 se borra desde el cursor hasta el final de la pantalla. En caso de que *n* valga 1 se borra hasta el principio de la pantalla desde la posición del cursor. Si *n* vale 2 se borra toda la pantalla (y utilizando ANSI.SYS de MS-DOS el cursor va al principio de la pantalla) |
| ESC [ *n* K | Borra parte de la línea. Si *n* es cero, desde el cursor al final de la línea, en caso de que valga 1 se borra hasta el principio. Si *n* vale 2 se borra toda la línea. |
| ESC [ *código*; *parámetro* [; *parámetro* ] p | Con este comando se redefine el comportamiento de una tecla. |
| ESC [ s | Guarda la posición actual del cursor. |
| ESC [ u | Coloca el cursor en la posición guardada anteriormente. |
| ESC [ *b* ; *fg* ; *bg* m | Establece la intensidad, el color del primer plano y el color de fondo del texto. Más ejemplos de códigos y colores en <http://softwarelivre.org/terceiro/blog/a-visual-cheat-sheet-for-ansi-color-codes> [Archivado](https://web.archive.org/web/20180922064147/http:/softwarelivre.org/terceiro/blog/a-visual-cheat-sheet-for-ansi-color-codes) el 22 de septiembre de 2018 en [Wayback Machine](https://es.wikipedia.org/wiki/Wayback_Machine" \o "Wayback Machine). |

# Conclusiones y comentarios:

El manejo del puerto serie UART0 en el microcontrolador ATMega2560 y el conocimiento de las secuencias de escape ANSI son competencias clave para los estudiantes que buscan desarrollar proyectos que involucren comunicación serie y presentación de datos en terminales de texto.

Bibliografía y Referencias

Wikipedia. (s.f.). Código escape ANSI. Wikipedia. https://es.wikipedia.org/wiki/C%C3%B3digo\_escape\_ANSI