

Materia:

Microprocesadores y
microcontroladores.



Reporte#9

Programación del uC del periférico
de comunicación serie utilizando
interrupciones.

Alumno:

Montoya Valdivia Omar
Antonio: 1252892

Profesor:

Jesús García

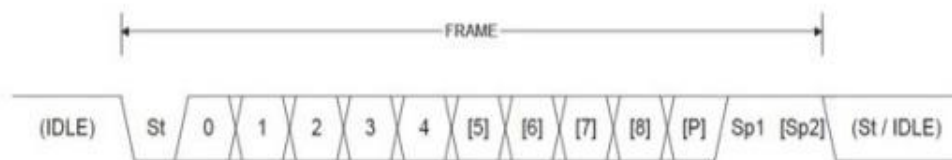
1) Manejo del Periférico de Comunicación Serie 0 (UART0) del microcontrolador

UART es un módulo de hardware que traduce los datos de paralelo a serial para ser transmitidos, las UARTs son usadas comunmente en conjunto con estándares de comunicación como EIA, RS-232, RS-422 o RS-485. la designación "universal" indica que el formato de los datos y las velocidades de transmisión pueden ser configuradas.

La gran mayoría de las familias de AVR poseen al menos una interfaz USART, la cual sirve para enviar y recibir datos seriales desde el microcontrolador a un dispositivo (una PC, por ejemplo) o a otro AVR. Esta interfaz usa solo tres pines (RXD, TXD y GND) enviando/recibiendo los datos usando una frecuencia pre-acordada y nos servirá en el resto del tutorial para hacer el debug de nuestros programas a modo de log a nuestras PC usando un programa de emulación de terminal.

- Formato de la trama

En comunicaciones una trama es una unidad de envío de datos, y el formato de la misma para UART es la siguiente:



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (Rx/D or Tx/D). An IDLE line must be high.

- Paridad

Los códigos de paridad se usan en Telecomunicaciones para detectar, y en algunos casos corregir errores en la transmisión. Para ellos se añade en origen un bit extra llamado bit de paridad a los n bits que forman el carácter original. Este bit de paridad se determina de forma que el número total de bits 1 a transmitir sea par (código de paridad par) o impar (código de paridad impar).

Código de paridad par El bit de paridad será un "0" si el número total de 'unos' a transmitir es "par" (bit "1" para un número "impar" de 'unos').

Código de paridad impar El bit de paridad será un "0" si el número total de 'unos' es "impar" (bit "1" para un número "par" de 'unos').

Configuración

1. Establecer la velocidad de transmisión en emisor y receptor debe ser la misma para poder realizar la comunicación.
2. Establecer el número de bits de datos que deben ser enviados.
3. Mantener el buffer listo, si es una transmisión cargarlo con el dato a transmitir, si es una recepción almacenar el dato recibido para poder recibir más datos sin perder información
4. Por último, habilitar el transmisor/receptor de acuerdo con el uso que se le desee dar.

Modos de Operación

Asynchronous Normal Mode En este modo de comunicación, el dato es transmitido/recibido asincrónicamente, el dato es transferido a la tasa de transmisión seleccionada en el registro UBRR.

Asynchronous Double Speed Mode Este es el modo de transmisión de mas alta velocidad en para comunicación asincrónica. En este modo se establecen la tasa de baudio de manera similar a como se hace en el modo normal, con la diferencia que la velocidad de transmisión será del doble de la que se configure en el registro UBRR. Estableciendo el bit U2X del registro UCSRA se puede duplicar la tasa de transferencia, este bit solo tiene efecto para la operación asíncrona.

Synchronous Mode Esta es la operación USART del AVR, cuando el modo síncronico está en uso (UMSEL = 1 en el registro UCSRC), el pin XCK será usado como entrada de reloj (Slave) o salida de reloj (Master).

Generación de la tasa de Baudios

Esta tasa se establece en el registro de 16 bits UBRR:

Bit	15	14	13	12	11	10	9	8	
	URSEL	—	—	—	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Dado que el microcontrolador AVR es de 8 bits cada registro tendrá un tamaño de 8 bits, por lo tanto el registro UBRR estará compuesto de dos registros UBRRH (high) y UBRRL (low) de manera similar al registro de 16 bits del ADC (ADCH y ADCL). El "USART Baud Rate Register" (UBRR) y el contador (down-counter) conectado funcionan como un prescaler programable y así generar la tasa de baudios, el downcounter, funciona

con el reloj del sistema (FOSC), y es cargado con el valor de UBRR cada vez que ha llegado a cero o cuando el registro UBRRRL ha sido modificado, un reloj es generado cada vez que el contador llega a cero. Este reloj es la salida del generador de tasa de baudios (= FOSC/(UBRR+1)). El transmisor divide este reloj en 2, 8, o 16 dependiendo del modo seleccionado, a continuación aparecen las formulas usadas para calcular la tasa de baudios y el valor de UBRR.

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Estableciendo el número de bit de datos

El tamaño de dato usado por USART se establece en los bits UCSZ2:0 en el registro UCSRC, el emisor y el receptor usan la misma configuración para su operación, es importante tener en cuenta que cambiar las configuraciones durante su operación normal puede "corromper" los datos.

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Estableciendo el número de bits de parada.

El bit USBS del registro UCSRC selecciona el número de bits de parada que insertará el transmisor, el receptor ignorará esta configuración.

USBS	Stop Bit(s)
0	1-bit
1	2-bit

Secuencias de escape ANSI

La secuencia de escape ANSI es una secuencia de caracteres ASCII, los dos primeros son el carácter 27 "Escape" ASCII (1Bh) y el carácter de corchete izquierdo "[" (5Bh). El carácter o caracteres que siguen a los caracteres de escape y corchete izquierdo especifican un código alfanumérico que controla una función de teclado o pantalla.

Algunas de ellas son:

Posicionar el cursor en pantalla

`Esc[Line;ColumnH`
`Esc[Line;Columnf`

Cursor Position:

Moves the cursor to the specified position (coordinates).

If you do not specify a position, the cursor moves to the home position at the upper-left corner of the screen (line 0, column 0). This escape sequence works the same way as the following Cursor Position escape sequence.

Para controlar gráficos en pantalla

Esc[Value;...;Valuem **Set Graphics Mode:**
Calls the graphics functions specified by the following values. These specified functions remain active until the next occurrence of this escape sequence. Graphics mode changes the colors and attributes of text (such as bold and underline) displayed on the screen.

Text attributes	
0	All attributes off
1	Bold on
4	Underscore (on monochrome display adapter only)
5	Blink on
7	Reverse video on
8	Concealed on

0	All attributes off
1	Bold on
4	Underscore (on monochrome display adapter only)
5	Blink on
7	Reverse video on
8	Concealed on

Foreground colors	
30	Black
31	Red
32	Green
33	Yellow
34	Blue
35	Magenta
36	Cyan
37	White

30	Black
31	Red
32	Green
33	Yellow
34	Blue
35	Magenta
36	Cyan
37	White

Background colors	
40	Black
41	Red
42	Green
43	Yellow
44	Blue
45	Magenta
46	Cyan
47	White

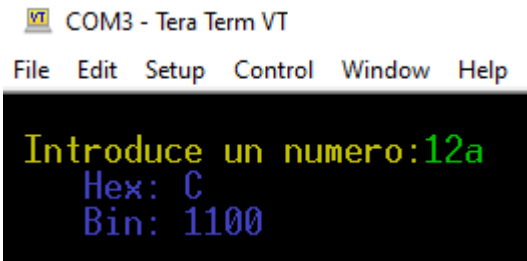
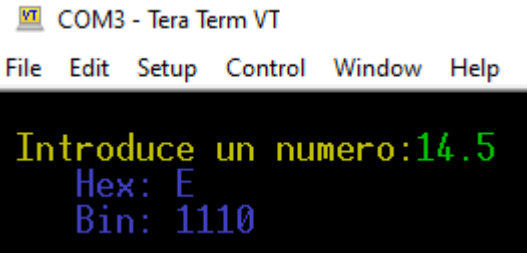
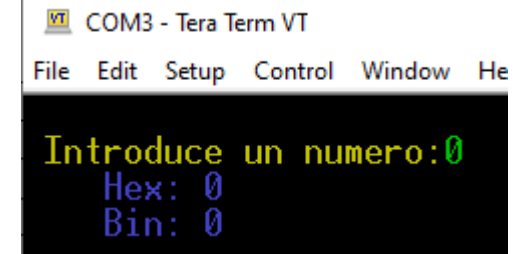
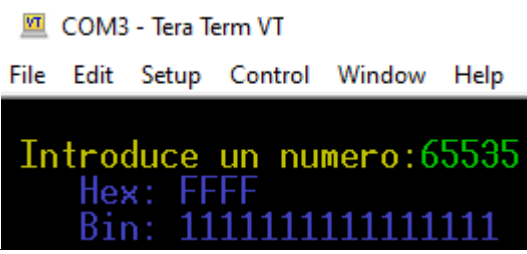
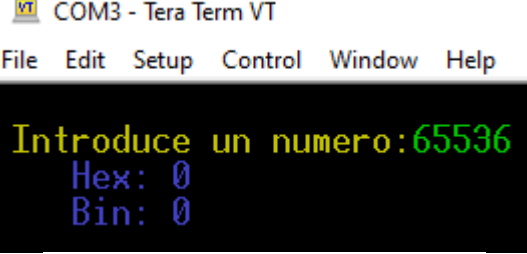
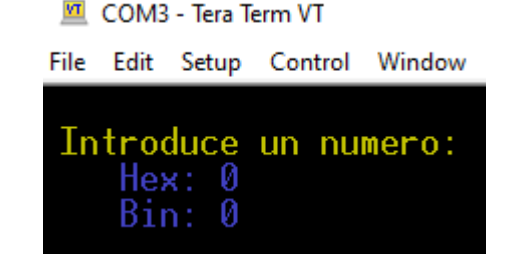
40	Black
41	Red
42	Green
43	Yellow
44	Blue
45	Magenta
46	Cyan
47	White

Parameters 30 through 47 meet the ISO 6429 standard.

Para limpiar la pantalla

Esc[2J	Erase Display: Clears the screen and moves the cursor to the home position (line 0, column 0).
---------------	--

Pruebas realizadas

Prueba requerida	Prueba obtenida
Entrada: "12a\n" Hex->"C" Bin->"1100"	
Entrada: "14.5\n" Hex->"E" Bin->"1110"	
Entrada: "0\n" Hex->"0" Bin->"0"	
Entrada: "65535\n" Hex->"FFFF" Bin->"1111 ... 1111"	
Entrada: "65536\n" Hex->"0" Bin->"0"	
Entrada: "\n" Hex->"0" Bin->"0"	

Entrada: Presionar '1' por 4 seg, presionar BACKSPACE por 5 seg, e introducir "\n"

Hex-> "2"

Bin-> "10"

Presionando 1 por 4 segundos:

The screenshot shows a Tera Term window titled "COM3 - Tera Term VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The main terminal area has a black background. It displays the prompt "Introduce un numero:" in yellow. Following the prompt are two lines of green text, each containing 60 'A' characters. The first line overflows onto the second line, demonstrating a buffer overflow.

Presionando backspace por 5 segundos y luego el 2

[illegible]

Conclusión:

Aprendí el uso de las interrupciones, como invocarlas y que hacer en caso de que se presente alguna. Comprendí el uso del puerto serié UART0, como configurarlo y por supuesto todos los registros de control de él mismo. Además, aprendí las secuencias ASCII para una terminal para cambiar colores, posicionamiento de cursor y como limpiar la pantalla.

Además, comprendí el funcionamiento de los registros de control de los UART y como interactuar con ellos de una manera más sencilla usando las directivas que me ofrece `avr/io.h`. Aprendí a transmitir y recibir datos mediante comunicación serial usando el UART0 del arduino, y por supuesto inicializar cualquier puerto serial del arduino usando la forma más eficiente usando interrupciones.

Referencias:

(2020). Retrieved 28 April 2020, from <http://ascii-table.com/ansi-escape-sequences-vt-100.php>

ASCII Escape Sequences. (2020). Retrieved 28 April 2020, from <https://vt100.net/docs/vt510-rm/chapter12.html>