Universidad Autónoma de Baja California

Microcontroladores y microprocesadores

Practica 9.

Alumno: Caudillo Sanchez Diego

Matricula: 1249199

Docente: Jesus Adan Garcia Lopez

Fecha de entrega: 30 de abril de 2020

Teoria

Manejo del Periférico de Comunicación Serie 0 (UART0) del microcontrolador ATmega 1280/2560

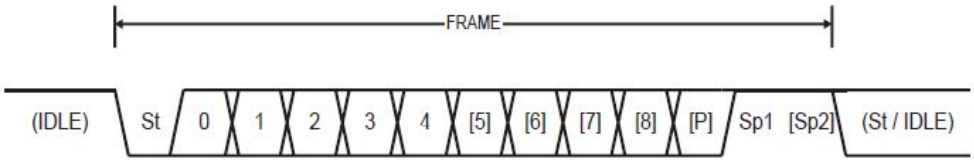
UART es un módulo de hardware que traduce los datos de paralelo a serial para ser transmitidos, las UARTs son usadas comúnmente en conjunto con estándares de comunicación como EIA, RS-232, RS-422 o RS-485. la designación "universal"indica que el formato de los datos y las velocidades de transmisión pueden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

¿Cómo configurar el UART?

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 mas 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 más 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 sólo tiene efecto para la operación asíncrona.

Synchronous Mode

Esta es la operación USART del AVR, cuando el modo sincrónico 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	
	–	–	–	–	UBRR[11:8]				UBRRHn
	UBRR[7:0]								UBRRLn
	7	6	5	4	3	2	1	0	
Read/Write	R	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 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 *down counter*, 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 *UBRRL* 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 fórmulas 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 (U2Xn = 0)	$BAUD = \frac{f_{OSC}}{16(\boldsymbol{UBRR}n+1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{OSC}}{8(\boldsymbol{UBRR}n+1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(\boldsymbol{UBRR}n+1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{2BAUD} - 1$

Estableciendo el numero de bits 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.

UCSZn2	UCSZn1	UCSZn0	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 paro

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

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

Secuencia de escape ANSI

Las secuencias de escape ANSI permiten enviar información de control a la consola para cambiar los atributos del texto representado. Así es posible seleccionar:

- El estilo del texto: normal, claro, subrayado, parpadeante, inverso y oculto
- El color del fondo
- El color del texto

El funcionamiento es muy sencillo, entre la información que se envía de salida a la consola, se incluyen las secuencias de escape dando instrucciones de cómo se ha de representar el texto a continuación.

Por ejemplo, si con printf se vuelca "Palabra %sresaltada%s" insertando en el primer %s la cadena de control para hacer que el texto sea verde, el texto que se escribe a continuación ('resaltada') tendrá color verde. En el último %s se debería introducir la secuencia de escape para volver a la normalidad. De no ser así, toda salida posterior sería de color verde.

La sintáxis sería: "\033[x;xx;xxm" dónde cada 'x' representa un dígito.

Comentarios y conclusiones

Con la elaboración de esta práctica se aprendió a mandar y recibir datos mediante las interrupciones. Asi como la manera y proceso en que trabaja, debido a que vamos observando paso a paso como configurarlo y que bits son los que debemos estar atentos al momento de manejar datos, ya sea para recibir o enviar.

Bibliografia

Datasheet Atmel 8-bit AVR microcontroller 1280/2560
https://en.wikipedia.org/wiki/ANSI_escape_code