

UART, interrupciones ARM

FIUSAC. Electrónica 5



UART



- ❑ Puerto serial que permite al microcontrolador comunicarse con otros dispositivos
- ❑ Cada UART tiene un registro de control de velocidad de baudios, el cual permite seleccionar la velocidad de transmisión
- ❑ En la Tiva, existe únicamente 1 bit de inicio, de 5 a 8 bits de datos, y para el bit de parada pueden ser 1 ó 2 bits.
- ❑ La velocidad de información (ancho de banda), está definida por el valor de datos o la información transmitida por segundo.
- ❑ El ancho de banda en un canal serial, es la velocidad en baudios (bits/seg), dividido por 10,



RS232




- ❑ Es un estándar, y usa un conector DB25.
- ❑ El estándar EIA-574 usa los niveles de voltaje del RS232 y el conector DB9
- ❑ Para implementar un canal bidireccional se necesitan de las terminales TxD, RxD y SG
- ❑ RS232 es un protocolo NRZ (non-return-to-zero)
- ❑ El tiempo en el RS232 es generado automáticamente por el UART




Comunicación asíncrona



- ❑ El transmisor tiene 16 elementos FIFO y 10 bits en el registro Shift
- ❑ Para sacar los datos usando el UART, el software se asegurará que la transmisión FIFO no esté llena (esperará si TXFF es 1), y luego escribir en el registro de datos (UART0_DR_R)
- ❑ Los bits son enviados en este orden: inicio, b0, b1, b2, b3, b4, b5, b6, b7, parada. Donde b0 es LSB y b7 es MSB
- ❑ En el registro de datos, sólo se escribe. El registro de transmisión está en la misma dirección del registro de datos, pero éste se especializa en recibir.

- 
- ❑ Cuando un byte se escribe en UART0_DR_R, este se coloca en la transmisión FIFO.
 - ❑ Byte por byte, el UART obtiene información del FIFO y carga esto en el registro de transmisión.
 - ❑ Se envía la trama (10 bits), 1 bit a la vez, a la velocidad especificada por el registro de velocidad de baudios.
 - ❑ Si existiera algún dato en FIFO o en el registro shift, se escribe UART0_DR_R, una nueva trama espera un momento a que las tramas anteriores sean transmitidas.
 - ❑ FIFO garantiza que los datos son transmitidos en el orden en que fueron escritos.
 - ❑ El hardware del puerto serial, está controlado por un reloj 16 veces más rápido que la velocidad de baudios

- 
- ❑ La recepción de tramas es un poco más complicada, porque se debe sincronizar el registro de recepción shift con los datos entrantes.
 - ❑ La porción de recepción del UART incluye un dato UoRx (pin de entrada) con niveles lógicos digitales
 - ❑ Existen diferentes bits estatus en la recepción:
 - ❑ Recepción FIFO con bandera vacía, RXFE, está limpia y el nuevo dato de entrada está en la recepción FIFO
 - ❑ Recepción FIFO con bandera llena, RXFF, lo cual indica que el registro está lleno

Activación UART

- ❑ Encender el reloj del UART en el registro `SYSCCTL_RCCUART_R`
- ❑ Habilitar la transmisión y recepción de los pines como señales digitales.
- ❑ La función alternativa para estos pines también debe seleccionarse
- ❑ El estatus de los 2 FIFOs se ajusta en el registro `UARTo_FR_R`:
 - ❑ Se activa la bandera `BUSY` mientras la transmisión no ha terminado de enviar los bits
- ❑ El registro de control `UARTo_CTL_R` contiene los bits que encienden la UART
- ❑ `TXE` es el bit que habilita la transmisión, `RXE` el de la recepción
- ❑ Se coloca en 1 a `TXE`, `RXE` y `UARTEN`, para activar el dispositivo



Sincronización I/O



- ❑ Latencia es el tiempo entre que un dispositivo indica que un servicio se requiere y el tiempo de que el servicio es iniciado.
- ❑ Incluye retrasos en hardware digital y en el software
- ❑ Para un dispositivo de entrada, la latencia de software es el tiempo entre la entrada de un nuevo dato y la lectura del mismo por el software
- ❑ Para un dispositivo de salida, es el tiempo entre que el software envía el dato y es recibido por el dispositivo exterior.
- ❑ La latencia de software cuando se usa un ADC, es cuando se supone que el ADC inicia y cuando realmente inicia a trabajar.




Interrupciones




- Propósito: permitir al microcontrolador interactuar con dispositivos externos.
- Existen 5 mecanismos para sincronizar el microcontrolador con los puertos:
 1. Blind cycle: método en donde el software espera un tiempo determinado y luego lee el dato de un dispositivo. Se le llama blind porque no existe información sobre el puerto de salida, y no reporta nada al micro. Se aconseja utilizarlo cuando la velocidad del puerto es lenta y predecible.
 2. Busy wait o polling: es un loop que chequea el estado de los puertos, en espera del estado de realizado.
 - a. Para un dispositivo de entrada, el software espera hasta que el dispositivo tiene un dato nuevo, y luego lo lee desde el puerto.
 - b. Para uno de salida, el software escribe el dato, los triggers lo mantienen en espera hasta que el dispositivo terminó de recibirlo.

- 
- 
3. Interrupción: utiliza el hardware para provocar una ejecución especial en el software.
 - a. Con un dispositivo de entrada, el hardware solicita un servicio de interrupción, cuando éste tiene un nuevo dato. El software activa el servicio de interrupción y lee lo que está en el I/O y lo almacena en la RAM global.
 - b. Para uno de salida, el hardware solicita una interrupción cuando el dispositivo de salida está ocioso (no en uso). El software interrumpe y el servicio de interrupción actúa con una función especial.
 4. Periodic polling: usa una interrupción de reloj periódicamente para chequear el estado de los I/O's, y dependiendo de lo indicado en el software, estas serán las acciones a seguir. Se utilizan cuando el dispositivo no soporta solicitud de interrupción directamente
 - a. Con un I/O de entrada, la bandera de interrupción se activa cuando hay un nuevo dato y lo guarda en la RAM global.
 - b. Con un I/O de salida, la bandera de interrupción se activa cuando el dispositivo está ocioso. Y los datos se colocan en la estructura global, y se escribe.



5. DMA: es una interface que transfiere datos directamente entre memorias.

- a. Con un dispositivo de entrada, el hardware solicitará una transferencia de DMA cuando el dispositivo tiene un nuevo dato.
 - b. Con uno de salida, el hardware solicitará una transferencia de DMA cuando el dispositivo esté ocioso.
- 



Consideraciones del hardware

- El hardware puede tener 3 posibles estados:
 - Ocioso: dispositivo deshabilitado o inactivo. No ocurren ingresos o egresos de datos.
 - Ocupado
 - Listo
- Para cualquiera de los casos, es la bandera la que indica el estado del mismo.



Conceptos de interrupción



- ❑ Interrupción: transferencia automática o ejecución de software en respuesta a un evento de hardware que es asíncrono con la ejecución normal del software.
- ❑ Trigger: evento de hardware, el cual puede estar ocupado o listo para la transición con los dispositivos de entrada – salida.
- ❑ Thread: definido como el camino de acción o el software a ejecutar. La ejecución de un servicio de interrupción se llama background thread. Se termina cuando se encuentra la instrucción BX LR. Un nuevo thread es creado para cada solicitud de interrupción.
- ❑ Proceso: definido por la acción que el software ejecuta.
- ❑ ARM (disarm): deshabilitar la fuente de interrupción
- ❑ Enable (disable) permite una interrupción a la vez, esto se logra cuando se coloca en 1 el bit I de PRIMEASK