

1. UART

Tenemos la versión del micro de 100 pines, entonces tenemos 5 UART's disponibles.

1.1. Pines UART

Sacado de la sección 5.1 Signal Multiplexing and PIN assignment del datasheet del micro. Tenemos acceso a todas las UART del micro en los pines de la tabla.

UART	RTS	CTS	TX	RX
0	A3(2)/A17(3)/B2(3)/D4(3)	A0(2)/A16(3)/B3(3)/D5(3)	A2(2)/A14(3)/B17(3)/D7(3)	A1(2)/A15(3)/B16(3)/D6(3)
1	E3(3)/C1(3)	E2(3)/C2(3)	E0(3)/C4(3)	E1(3)/C3(3)
2	D0(3)	D1(3)	D3(3)	D2(3)
3	B12(2)/E7(3)/B8(3)/C18(3)	B13(2)/E6(3)/B9(3)/C19(3)	E4(3)/B11(3)/C17(3)	E5(3)/B10(3)/C16(3)
4	E27(3)/C12(3)	E26(3)/C13(3)	E24(3)/C15(3)	E25(3)/C14(3)

El terminal TX de la UART0 sale por los pines PTA2, PTA14, PTB17, PTD7.

2. SPI

El proceso de leer/escribir (como master), es un proceso de escritura y lectura simultaneo. Para leer un dato (del perifero que sea) tengo que escribir algo a la salida (si fuera necesario, data, sino algo dummy) para hacer circular la data en el sistema. Analogia del engranaje de la bicicleta.

2.1. Pines SPI

- **DSPI_PCSn:** Señal que indica, desde el master, a que slave le mando data
- **DSPI_SCK:** Señal de clock. Comanda la comunicación del sistema
- **DSPI_SIN:** Señal de entrada.
- **DSPI_SOUT:** Señal de salida

2.2. Driver SPI

El driver funciona con SysTick?

2.2.1. Que debe hacer mi driver?

- Inicializar el perifero (Que SPI quiero inicializar)
- setear el clock (por cada SPI)
- leer: de cual slave leo?(por cada SPI)
- escribir: Tengo que decirle a que slave hablarle (por cada SPI)