Exercises

1. Examine the reference manual for your board to determine the answers to the following questions.
   1. Which pins can be used for UART?

UART0 RX: P4\_0

UART0 TX: P4\_1

UART1 RX: P0\_4, P3\_0

UART1 TX: P0\_5, P3\_1

* 1. Which pins can be used for I2C?

I2C0 SCL: P4\_0

I2C0 SDA: P4\_1

I2C1 SCL: P0\_4, P3\_0

I2C1 SDA: P0\_5, P3\_1

* 1. Which pins can be used for SPI?

SPI0 MOSI: P4\_0

SPI0 MISO: P4\_1

SPI0 SCK: P4\_2

SPI0 SS0: P4\_3, SS1: P0\_0, SS2: P0\_1, SS3: P0\_2

SPI1 MOSI: P0\_4, P3\_0

SPI1 MISO: P0\_5, P3\_1

SPI1 SCK: P0\_6

SPI1 SS0: P0\_7, SS1: P3\_4, SS2: P3\_5, SS3: P3\_6

1. Show the register settings needed to configure UART0 to transmit and receive at 9600 baud, eight data bits (LSB first), one stop bit and no parity. Enable an interrupt to indicate that the module has accepted data.

Set GPIO bits to their alternate functions and as outputs / inputs respectively.

Set baud rate to 9600: divider = 24 MHz/(4\*9600Hz)-1 = 624, which in hex is 0x0270. Load a CLK\_DIVIDER with 0x0270 with bit 31 set for enable. Set CLK\_SELECT03 to select the divider which is being used and clock the module.

In SCB1\_CTRL, set SCB\_MODE to SCB\_MODE\_UART (to put the serial comms module in UART mode) and SCB\_OVS to 3 (4 – 1, sets the oversample rate to 4x).

Clear SCB1\_UART\_CTRL to put in standard UART submode.

In both SCB1\_RX\_CTRL and SCB1\_TX\_CTRL, set SCB\_ENABLED\_MASK and SCB\_DATA\_WIDTH to 7 (8 data bits - 1).

In SCB1\_INTR\_RX\_MASK, set SCB\_NOT\_EMPTY\_MASK to enable an interrupt if the receive FIFO isn’t empty (i.e. there is data to be read).

1. Show the register settings needed to configure I2C as a manager with eight data bits (MSB first).

Set up similar to UART, except

Configure GPIO as DM\_0\_Z.

Clock the module at a high frequency.

In SCB0\_CTRL, set SCB\_MODE to SCB\_MODE\_I2C.

In SCB0\_I2C\_CTRL, set as a manager with SCB\_MANAGER\_MODE\_MASK and configure the oversampling.

Enable the module by setting SCB\_ENABLED\_MASK in SCB0\_CTRL.