Four conditions must be true simultaneously for an interrupt to occur:

• The hardware peripheral’s interrupt arm bit must be set (by software).

• The hardware peripheral’s interrupt flag must be set (by hardware).

• The interrupt has a higher priority than any executing ISR, and the PRIMASK register is 0 (interrupts are enabled).

• The interrupt source is enabled in the Nested Vectored Interrupt Controller (NVIC).

TODO: List

1. To use interrupts with the serial communication interface.

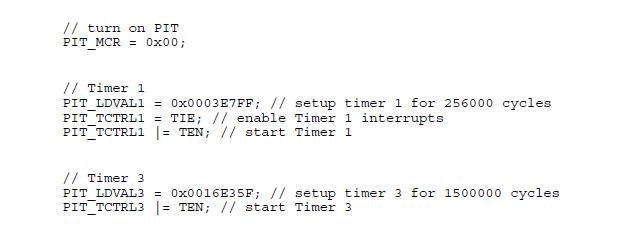
Global interrupt enable

NVICISERx🡪

Vector tables- ask about

2. To set up a periodic timer.

* There are multiple timers, apart from MCR each timer has a dedicated reg
* LDVAL trigger = (period / clock period) -1
* Eg of code
* Bus clock speedo



* PIT\_MCR MDIS -> enables module
* PIT\_LDVAL TSV-> start value for timer to count down (see equation above) (is setup)
* PIT\_TCTRL TIE -> enables interrupts
* PIT\_TCTRL TEN-> starts timer
* PIT\_TFLG TIF -> timer interrupt flag (write 1 to clear)

3. To implement a real-time clock function.

* Use 18pf load of capacitors (apparently in parallel)- RTC\_CR
* RTC\_CR (OSCE)Oscillator enable
* RTC\_LR (crl)-> set to zero to lock crtl reg
* RTC\_SR (TCE) -> enables the 1sec rtc timer
  + Bit must be zero in order to write to
  + Will be set when there is an error SR[TOF] or SR[TIF]
* RTC\_IER(TSIE) -> enables an interrupt every second
  + TODO: investigate what else to do for interrupt
  + Non-ipr: 2, ipr: 16
* Do not think we need compensation for 1 sec

4. To set up a software interface to timer functions.

5. To expand the implementation of the Tower serial protocol.