

Steps for Interrupt Configuration

1. At the start of your program you should call the InstallTable function. This function is responsible to install your interrupt table
 - a. Use the following command: `bl Interrupt_Install_Table`
 - b. This function should be the same as the one in the exercise. Make sure you push all your registers to the stack.
2. After that you need to enable IRQ line in CPSR register and IRQ table
 - a. Update timer value. This will be current time + delay
 - b. For IRQ
 - i. Load the value in 0x3F00B210 and put it in r0
 - ii. Move 10 to r1 // 10 since we are enabling IRQ line 1 and 3 you can also enable only 1
 - iii. Store the value of r1 in r0
 - c. Disable all other interrupts
 - i. Load the value in 0x3F00B214 and put it in r0
 - ii. Move 0 to r1
 - iii. Store the value of r1 in r0
 - d. For cpsr_c register
 - i. `mrs r0,cpscr`
 - ii. `bic r0, #0x80`
 - iii. `msr cpsr_c, r0`
3. For the IRQ function that should be executed when the interrupt is executed, you should do the following:
 - a. Test if timer1 did the interrupt
 - i. Load the values stored in 0x3F00B204 to r1
 - ii. Test bit 2
 - iii. If result is zero go to e
 - b. Check if the game was paused
 - i. You should have a label in memory where you store in it if the game is paused or not
 - ii. If paused you go to e
 - c. If a,b,c are all valid you draw your value pack.
 - d. Enable CS timer Control
 - i. Load the value stored in 0x3F003000
 - ii. Put 1 in bit 1 and the rest are zeroes
 - e. Update time in C1
 - f. Repeat (2)
 - g. Then `subs pc, lr, #4`