```
AREA    InterruptStuff, CODE, READONLY

/* Interrupt Request Handler */

/*
    This interrupt request handler will fully preserve the context
    of the original program it has been called from

        MRS = Move CPSR to a RX
        MSR = Move RX to CPSR
*/

irqhan
--------------------------------------------------------------------------------
        /* 1 - Change into System Mode */
--------------------------------------------------------------------------------
        MRS R2, CPSR                            // load CPSR into register
        BIC R2, R2, #0x1F                       // clear the mode field
        ORR R2, R2, #0x1F                       // set system mode
        MSR CPSR_c, R2                          // (privilege needed to return to IRQ mode)

--------------------------------------------------------------------------------
        /* 2 - Get previous SP and LR */
--------------------------------------------------------------------------------
        MOV R0, SP                              // retrieve user mode SP
        MOV R1, LR                              // retrieve user mode LR

--------------------------------------------------------------------------------
        /* 3 - Change into IRQ Mode */
--------------------------------------------------------------------------------
        MRS R2, CPSR_c                          // extract CPSR to R2
        BIC R2, R2, #0x1F                       // clear the mode field
        ORR R2, R2, #0x12                       // set bits for IRQ mode
        MSR CPSR_c, R2                          // store new mode to CPSR

--------------------------------------------------------------------------------
        /* 4 - Store SPSR, PC, LR, SP onto Program Stack */
--------------------------------------------------------------------------------
        MRS R2, SPSR                            // extract SPSR to R2
        STR R2, [R0, #4]                        // store the spsr(CPSR) of program to SP

        STR LR, [R0, #8]                        // store PC
        STR R1, [R0, #12]                       // store LR
        STR R0, [R0, #16]                       // store SP
        ADD R0, #16                             // update stack pointer to "true" value

--------------------------------------------------------------------------------
        /* 5 - Store Registers R3 - R12 (Unchanged) onto Program Stack */
--------------------------------------------------------------------------------
        STMFD R0!, {R3-R12}                     // store register contents on the stack

--------------------------------------------------------------------------------
        /* 6 - Get original R0 - R2 values and push onto stack */
--------------------------------------------------------------------------------
        LDMFD SP!, {R3-R5}                      // take R0-R2 off of IRQ stack
        STMFD R0!, {R3-R5}                      // store onto programs stack

        MOV R0, R3                              // move register contents back to original pos
        MOV R1, R4                              // overwriting LR of user mode
        MOV R2, R5;                             // overwriting SP of user mode

        LDMFD SP!, {R3-R5}                      // restore registers to condition before context
                                                // storing began
                                                // * system context storing done #*

                //clear the interrupt here, and perhaps other Irq functionality

        LDMFD   SP!,{PC}^                       // return from interrupt, restoring pc from lr
                                                // and also restoring the CPSR
```