**Main.s Source Code**: Kyler Fillerup & Casey Nordgran – Group Partnership #32

```
.equ    STACK_TOP, 0x20000800
.equ    SYSREG_SOFT_RST_CR, 0xE0042030

        .section .int_vector,"a",%progbits @ First linker code section
        .global  _start                 @ Linker entry point
_start:
        .word    STACK_TOP, main
        @ End of int_vector section

        @ Standard text section
        .text
        .syntax  unified
        .thumb

        .type    main, %function
main:
        @ Load   SYSREG_SOFT_RST_CR address
        movw     r0, #:lower16:SYSREG_SOFT_RST_CR
        movt     r0, #:upper16:SYSREG_SOFT_RST_CR
        @ Reset  GPIO hardware
        ldr      r1, [r0, #0]
        orr      r1, #0x4000
        str      r1, [r0, #0]
        @ Take GPIO hardware out of reset
        ldr      r1, [r0, #0]
        mvn      r2, #0x4000 @ move bitwise negation of 0x4000 into r2
        and      r1, r2
        str      r1, [r0, #0]

        mov      r0, #24
        bl       initGPIO @ Call initGPIO in gpio.s to initialize GPIO 24

                 mov      r0, #0
        bl       setGPIO @ Call setGPIO in gpio.s to write 0 to GPIO output register

                 mvn      r0, #1
                 ror      r0, #1 @ set r0 to all 1's except for pin 31
                 add      r4, r0, #0  @ r4 is set to all 1's except pin 23
                 ror      r4, #8 @ r4 is used as reference to known when
                                                    @ to reset the zero in r0 back to pin 31
loop:
        bl       setGPIO         @ enter setGPIO
        ror   r0, #1 @ rotate the all bits right one
        cmp                      r0, r4
                 bne             loop
                 ror             r0, #23 @ if r0 has low bit on pin 23, rst to pin 31
        b                loop
        .end
```