In order to make the program work with the interrupt, I need to activate the compare mode functionality of the general purpose timer, as well as the timer mode. I will use the timer mode to counter up constantly and to reset the value, as well as the compare mode so i can compare until it reaches 1 second. I need to turn on the interrupt bit for this.

Clock module: 0x44E00000
Turn on clock by writing 0x2 at offset 88h

CM_DPLL: 0x44E0_0500
To enable 32khz clock, write 0x2 at offset 10h

DMtimer4: 0x4804_4000

Enable emulation mode running by writing 0x2 at offset 10h
Turn clock on by writing 0x2 at offset 38h
To enable overflow interrupt, write 0x2 at offset 2Ch
To set a clock for 1s, write 0xFFFF8000 at offset 3Ch
To reload the clock for 1s, write 0xFFFF8000 at offset 40h

INTC : 0x48200000
DMTIMER 4 is bit 92, which is bit 28 (0x10000000) of INTC_MIR_CLR2 at offset E8h

High level algorithm

Mainline:

1. Create stacks
2. Turn on GPIO1 Clock
3. Declare the registers to be used
4. Set LED to off
5. Declare LED as an output
6. Set falling detect on input register
7. Enable the interrupt on input
8. Enable timer 4 interrupt on INTC #92
9. Turn on timer 4 clock
10. Set timer 4 clock mux to 32KHz
11. Initialize timer count and load registers for overflow interrupt
12. Let interrupt controller receive interrupt signals
13. Let processor receive interrupt

LOOP:

1. Infinite loop

INT_DIRECTOR:

1. Save registers
2. Check if button is pressed
3. If yes, Go to button service if pressed
4. If not, check if timer 4 interrupt
5. If timer interrupt, check overflow.
6. If overflow, toggle LED
7. If not, restore registers and return

BUTTON_SVC:

1. Turn off interrupt request
2. Turn LED on
3. Start timer and set auto reload
4. Reset intc to allow new interrupts
5. Restore registers and return to infinite loop

LED:

1. Turn off timer 4 overflow interrupt
2. Set GPIO1_21 to output low.
3. Reset intc to allow new interrupts
4. Restore registers and return to wait loop

Low level algorithm

Mainline:

1. Create stacks
    a. Create stack pointer for SVC mode
    b. Switch to IRQ mode (#0x12)
    c. Create stack pointer for IRQ mode
    d. Go back to SVC mode (#0x13)
2. Turn on GPIO1 Clock
    a. Write 0x02 to 0x44E000AC
3. Set GPIO1_21 for LED off (low)
    a. Write 0x00200000 to 0x4804C190
4. Set GPIO1_21 as output
    a. Load 0x4804C134
    b. Read data in 0x4804C134
    c. Modify it with 0xFFDFFFFF
    d. Store it in 0x4804C134
5. Set falling detect on input register

    a. Load 0x4804C14C
    b. Read data in 0x4804C14C
    c. Modify it with 0x20000000
    d. Store it in 0x4804C14C
6. Enable the interrupts
    a. Load 0x482000E8 (INTC_MIR_CLEAR3)
    b. Unmask INTC_INT_98, GPIOINT1A
    c. Write value back
    d. Load 0x482000C8 (INTC_MIR_CLEAR2)
    e. Unmask INTC_INT_92, timer 2 interrupt
    f. Write value back
7. Let processor receive interrupt
    a. Copy CPSR
    b. Clear bit 7
    c. Write back to CPSR

LOOP:

1. Go to LOOP

INT_DIRECTOR:

1. Push registers on stack
2. Check interrupt
    a. Was button pushed?
        i. Load address 0x482000F8 (INTC-PENDING_IRQ3)
        ii. Read the value in
        iii. Check if the value in bit 2 is low
            1. Check timer interrupt
        iv. Load address 0x4804C02C (GPIO1_IRQSTATUS_0)
        v. Check if the value in bit 29 is high
            1. If yes, go to BUTTON_SVC
        vi. Else enable new IRQ input from INTC
            1. Write 0x1 to 0x48200048 (INTC_CONTROL)
        vii. Restore registers and return to LOOP
    b. Timer overflow?
        i. Load address 0x482000D8 (INTC_PENDING_IRQ2)
        ii. Read value in
        iii. Compare with 0x10000000
        iv. If no, enable new IRQ response in INTC
            1. Write 0x1 to 0x48200048 (INTC_CONTROL)
            2. Restore registers and return
        v. If yes, check for overflow
            1. Load address 0x48040028
            2. Compare with 0x2

3. If bit is 1, go to LED
4. If bit is 0, enable new IRQ input from INTC
    a. Write 0x1 to 0x48200048 (INTC_CONTROL)
5. Restore registers and return to LOOP

BUTTON_SVC:

1. Load value 0x20000000 to turn off GPIO1_29 interrupt request
2. Write the value to 0x4804C02C (GPIO1_IRQSTATUS_0)
3. Load address 0x48200048 (INT_CONTROL)
4. Load value 01 to clear to bit 0
5. Write it back
6. Load address 0x4804C194 (GPIO1_SETDATAOUT)
7. Load value 0x00200000 to turn on LED
8. Write the value into the register
9. Set counter
10. Go to LOOP2

LED: