Part 1

High level algorithm:

Start program
    Initialize GPIO clocks for GPIO1
    Turn on LED0 (GPIO1_21) and LED3 (GPIO1_24)
        Delay for 2 seconds
    Turn off LED0 and LED3
    Turn on LED1 (GPIO1_22) and LED2 (GPIO1_23)
        Delay for 2 seconds
    Turn off LED1 and LED2
Loop back to pulse LEDs 10 times
End program

Low level algorithm:

Start program
Write 0x02 to 0x44E000AC to initialize clock module GPIO1
Write 0x01E00000 to 0x4804C190 to initialize to zero
Write 0xFE1FFFFF to 0x4804C134 to enable outputs
Repeat
    Write 0x01200000 to 0x4804C194 to turn on LED0 and LED3
        Delay loop using 0x3B9ACA00 as counter
    Write 0x01200000 to 0x4804C190 to turn off LED0 and LED3
    Write 0x00C00000 to 0x4804C194 to turn on LED1 and LED2
        Delay loop using 0x3B9ACA00 as counter
    Write 0x00C00000 to 0x4804C190 to turn off LED1 and LED2
End repeat
End program

Part 2

High level algorithm:

Start program
    Initialize GPIO clocks for GPIO1
    Initialize GPIO1_21 to GPIO1_24 as logic low outputs
    Initialize GPIO1_29 to handle button
    Initialize INTC
    Enable IRQ processor by clear bit 7
    Wait for interrupt by loop
Repeat
    If button is pushed rotate LEDs
    Else continue to wait

    If button is pushed again turn off LEDs
    Else continue to wait
End repeat
End program

Low level algorithm:

Write 0x02 to 0x44E000AC to initialize clock module GPIO1
Write 0x01E00000 to 0x4804C190 to initialize to zero
Write 0xFE1FFFFF to 0x4804C134 to enable outputs
Use 0x14C (falling edge) and 0x34 (IRQ status) for button initialization
Use =0x482000E8 and 0x04 to initialize INTC
Enable IRQ processor by clear bit 7
Repeat until button is pushed
    If button pushed continue program
End repeat
Repeat until button is pushed
    Rotate LEDs
    If button pushed turn off LEDs and move to previous loop
End repeat
End program

Part 3

High level algorithm

Initialize INTC - for button and timer 7
Turn on timer 7 clock
Set timer 7 functional clock to 32.768 KHz clock
Initialize timer registers for desired count

If button pushed
    Start timer 7 and set for auto reload
    Turn on LED rotation (will contain turning off timer 7 overflow and enable new interrupt)
    If timer interrupt
    Update LED rotation
    Else continue to wait
Else continue to wait
If button pushed
    Turn off timer and LED rotation
    Return to wait loop
Else continue to wait
Low level algorithm

*Since this low level algorithm contains much of the last two algorithms. I thought it would benefit myself much more if I just included the specifics for the timer

Start program
    Intialize SVC
    Initialize GPIO clock module
    Initialize outputs
    Initialize button detection
    Initialize INTC write 0x80000000 to 0x482000C8
    Turn on timer 7 0x2 to 0x44E0007C and 0x44E00504
    Initialize timer 7 registers 0x2 to 0x4804A002C and load count value 0xFFFF0000 to 0x4804A0040 and 0x4804A003C
    Repeat
        If button pushed or timer overflow cause interrupt
            Create a subroutine for checking the timer
        Everything else should follow the same algorithm of part 2
    End repeat
End program