**PART 2 ALGORITHMS**

**HIGH LEVEL:**

MAINLINE

1. Set up stacks for SVC and IRQ modes
2. Load LED\_STATUS pointer to track LEDs ON/OFF
3. Enable GPIO1 module clock
4. Enable GPIO2 module clock
5. Set GPIO1\_21-24 for LEDs off (logic low)
6. Set GPIO1\_21-24 as outputs
7. Set up falling edge detection on GPIO2\_22 (push button) and generate interrupt
8. Enable GPIO2\_22 as interrupt source
9. Initialize INTC and button interrupt on INTC #32
10. Enable IRQ input by clearing bit 7 in CPSR
11. Wait for interrupt in either LED ON (LED\_CYCLE) or LED OFF (WAIT) loops

INT\_DIRECTOR:

1. Save used SVC registers, link register
2. Check if interrupt from button
3. IF NO enable new IRQ, restore saved registers and return

ELSE go service button interrupt BUTTON\_SVC

BUTTON\_SVC:

1. Turn off GPIO2\_22 interrupt
2. Check if LED cycle in ON or OFF

IF OFF, update LED\_STATUS = 1 (LED cycle ON)

ELSE update LED\_STATUS = 0 (LED cycle OFF) and turn off LEDs

1. Enable INTC for new interrupt
2. Restore saved registers and return to LED ON or LED OFF loops

LED\_CYCLE:

REPEAT

IF LED0 is on, turn on LED1

ELSE IF LED1 is on, turn on LED2

ELSE IF LED2 is on, turn on LED3

ELSE turn on LED0

Wait 1 second

Check if LED\_STATUS changed

IF YES, enter LED OFF loop

UNTIL loss of power (endless loop)

**LOW LEVEL:**

MAINLINE

1. Set up stacks for SVC and IRQ modes

Load SVC stack pointer

Point to top of stack

Switch to IRQ mode

Load IRQ stack pointer

Point to top of stack

Switch to SVC mode

1. Load LED\_STATUS pointer to track LEDs ON/OFF
2. Enable GPIO1 module clock

Write 0x2 to CM\_PER\_GPIO1\_CLKCTRL at 0x44E00000 + 0xAC

1. Enable GPIO2 module clock

Write 0x2 to CM\_PER\_GPIO2\_CLKCTRL at 0x44E00000 + 0xB0

1. Set GPIO1\_21-24 for LEDs off (logic low)

Write 0x01E00000 to GPIO1\_CLEARDATAOUT at 0x4804C000 + 0x190

1. Set GPIO1\_21-24 as outputs (RMW)

Load word from GPIO1\_OE at 0X4804C000 + 0x134

AND word with 0xF1EFFFFF

Write result back to GPIO1\_OE

1. Set up falling edge detection on GPIO2\_22 (push button) and generate interrupt (RMW)

Load word from GPIO2\_FALLINGDETECT at 0x481AC000 + 0x14C

Set bit 22 by ORing with 0x00400000

Write result back to GPIO2\_FALLINGDETECT

1. Enable GPIO2\_22 as interrupt source

Write 0x00400000 to GPIO2\_IRQSTATUS\_0 at 0x481AC000 + 0x34

1. Initialize INTC and button interrupt on INTC #32

Write 0x1 to INTC\_MIR\_CLEAR1 at 0x48200000 + 0xA8

1. Enable IRQ input by clearing bit 7 in CPSR

MRS R3, CPSR

BIC R3, #0x80

MSR cpsr\_C, r3

1. Wait for interrupt in either LED ON (LED\_CYCLE) or LED OFF (WAIT) loops

WAIT:

REPEAT

Load word from LED\_STATUS

Check bit 0 for LED cycle status

IF bit0=1, go to LED\_CYCLE

UNTIL loss of power (B WAIT)

INT\_DIRECTOR:

1. Save used SVC registers (R0-R9), link register (R14)

STMFD SP!, {R0-R9, LR}

1. Check if interrupt from button (INT #32)

Load word from INTC\_PENDING\_IRQ1 at 0x48200000 + 0xB8

TEST word with INT #32 bitmap (0x00000001)(bit 0) if interrupt from GPIOINT2A

IF NO, Write 0x1 to INTC\_CONTROL at 0x48200000 + 0x48

Restore registers (LDMFD SP!, {R0-R9, LR})

Return to mainline at next instruction after IRQ (SUBS PC, LR, #4)

ELSE load word from GPIO2\_IRQSTATUS\_0 at 0x481AC000 + 0x2C

TEST word with bit 22 (0x00400000) if interrupt from button press

1. IF NO, enable new IRQ, restore saved registers and return

Write 0x1 to INTC\_CONTROL at 0x48200000 + 0x48

Restore registers (LDMFD SP!, {R0-R9, LR})

Return to mainline at next instruction after IRQ (SUBS PC, LR, #4)

ELSE go service button interrupt BUTTON\_SVC

BNE BUTTON\_SVC

BUTTON\_SVC:

1. Turn off GPIO2\_22 interrupt

Write 0x00400000 to GPIO2\_IRQSTATUS\_0 at 0x481AC000 + 0x2C

1. Check if LED cycle is ON or OFF

Load word from LED\_STATUS

TEST word with bit 0

IF OFF, update LED\_STATUS = 1 (LED cycle ON)

Write 0x1 to LED\_STATUS

Write 0x1 to INTC\_CONTROL at 0x48200000 + 0x48

Restore registers (LDMFD SP!, {R0-R9, LR})

Return to mainline at next instruction after IRQ (SUBS PC, LR, #4)

ELSE update LED\_STATUS = 0 (LED cycle OFF) and turn off LEDs

Write 0x0 to LED\_STATUS

Write 0x01E00000 (bits 21-24) to GPIO1\_CLEARDATAOUT at 0x4804C000 + 0x190

1. Enable INTC for new interrupt

Write 0x1 to INTC\_CONTROL at 0x48200000 + 0x48

1. Restore saved registers and return to LED ON or LED OFF loops

Restore registers (LDMFD SP!, {R0-R9, LR})

Return to mainline at next instruction after IRQ (SUBS PC, LR, #4)

LED\_CYCLE:

REPEAT

Read word from GPIO1\_SETDATAOUT at 0x4804C000 + 0x194

Mask bits 0-20 an 25-31 to check only LED bits (21-24) by ANDing 0x01E00000 with word

Compare masked word with LED0 (GPIO1\_21) on with bitmap 0x00200000

IF LED0 ON (Z flag set), load 0x00400000 to turn on GPIO1\_22 (LED1)

ELSE IF ALL LED OFF (N flag set), load 0x00200000 to turn on GPIO1\_21 (LED0)

Compare masked word with LED2 (GPIO1\_23) on with bitmap 0x00800000

IF LED1 ON (N flag set), load 0x00800000 to turn on GPIO1\_23 (LED2)

ELSE IF LED2 ON (Z flag set), load 0x01000000 to turn on GPIO1\_24 (LED3)

ELSE LED3 ON, load 0x00200000 to turn on GPIO1\_21 (LED0)

Write loaded bitmap to GPIO1\_SETDATAOUT at 0x4804C194

Load 0x00200000 to set delay at 1 second

REPEAT

Subtract #1 from loaded word and update flags

UNTIL loaded word==0

UNTIL loss of power (endless loop)