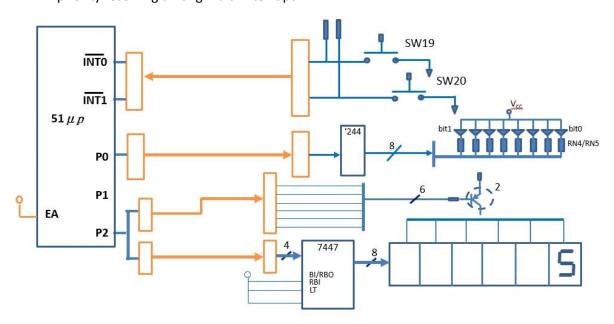
Microprocessor Lab 6 Report

0416106 彭敬樺 0416109 周才錢

Subject and Goal:

This lab is about using μ -Vision 51IDE residing on MegaWin82G516 to:

- operations with two external interrupts and two timer interrupts
- built-in counter/timer setup and control for timing application
- time-up event detection by timer interrupt
- key entry event by external interrupt
- priority resolving among multi-interrupt



Preparations:

- Power cable and required connection from the output to the led input is established.
 Port P0 will control 8x1 discrete LED module. Post P1 and P2 will control 7-segment power module and six 7-seg LED DIGs respectively
- Check the correctness and check if there are any defective on the board by activating all 8x1 discrete LED module and 7-segment LED module using static/dynamic pattern display.

Operating Procedure:

- Jumper-wiring for circuit setup
- Check the 8x1 discrete LED module and 7-segment LED module to see if it's working or not by running code to turn all the light on.
- Code preparation
- Task execution:
 - Start IDE51 emulation,
 - Start execution and troubleshooting if necessary.

Code Preparation:

;	jmp TMR1
; (1) 4 interrupt events in action	start:
; (2) 1x8 LEDs	mov sp, #50H
; normal ON-OFF \square OFF-ON	mov TMOD, #11H
; INT0 □ L2R scanning	mov TH0, #>(65536-10000)
; INT1 □ R2L scanning	mov TL0, #<(65536-10000)
; (3) 7-seg LEDs	mov TH1, #>(65536-50000)
; DIG6-5: timing by second (TF1)	mov TL1, #<(65536-50000)
; DIG4-1: timing by 10ms (TF0)	mov IE, #8FH
; (4) priority	mov IP, #0AH
; priority order of the 4	mov 30H, #0
; interrupts ???	mov 31H, #0
; =====================================	mov 32H, #0
; port0: 1x8 discrete LEDs control	mov 33H, #0
; port1: power-SW control for	mov 34H, #0
; 7-seg LED module	mov 35H, #0
; port 2: pattern control for	clr OH
; timing display on 7-seg LEDs	clr 1H
; 30H: TMR1 10-2sec-register	mov 40H, #0FFH
; 31H: TMR0 sec-register	mov 41H, #0FFH
; 32H: TMR0 10-2sec-register	
; 33H: TMR0 10-2sec time-up flag	setb TR0
; 34H: TMR1 10-2sec time-up flag	setb TR1
; 35H: TMR1 counter: 0-19	;mov P1, #0H
; 36H: normal pattern for 1x8 LEDs	;mov P2, #0H
; BITMAP 0H: INTO flag	clr C
; BITMAP 1H: INT1 flag	mov 2H, C
; BITMAP 2H: left-to-right Cy	mov 3H, C
; BITMAP 3H: right-to-left Cy	display_go:
; 40H: 1x8 LED pattern for INT0	
; 41H: 1x8 LED pattern for INT1	normcycle:
org 0	mov 36H, #0FH
jmp start	mov R4, #2
org 03H	mov R5, #14
jmp INT0	halfcycle:
org 0BH	mov P0, 36H
jmp TMR0	int_in_session:
org 13H	mov R6, #30
jmp INT1	DIG6: mov A, 30H
org 1BH	mov B, #10

mov P1, #0DFH mov P2, A call delay mov A, 40H DIG5: mov A, B mov P1, #0EFH mov P2, A call delay mov C, 2H mov P2, A call delay mov A, 31H mov B, #10 div A, B mov P1, #0FFH mov P2, A call delay mov P0, A div A, B mov P1, #0FFH mov P2, A call delay mov P1, #0FFH mov P2, A call delay mov P2, A call delay mov P3, #0FFH mov P4, #0FFH mov P5, A call delay mov P1, #0FFH mov P4, A call delay mov P5, #0FFH mov P5, A call delay mov P6, #0FFH mov P7, #0FFH mov P8, #10 div A, B mov P1, #0FFH mov P1, #0FFH mov P1, #0FFH mov P2, A call delay mov B, #10 div A, B mov P1, #0FFH mov P1, #0FFH mov P2, A call delay mov P1, #0FFH mov P2, A call delay mov P3, #10 div A, B mov P1, #0FFH mov P2, A call delay djnz R4, midway; Q1 pinp halfcycle; Q2 call delay djnz R6, DIG6 imov A, 41H mov A, 41H mov C, 3H mov T10, #2(65536-10000) mov P0, A mov P0, A mov P0, A mov P1, #00, ext1 mov P0, A mov P1, #00, ext1 mov A, 3H, #0 mov P0, A mov A, 32H mov P0, A mov A, 32H mov P0, A mov P0, A mov A, 32H mov P0, A mov A, 32H mov P0, A mov P0, A mov A, 32H mov P0, A mov A, 32H mov P0, A mov P1, #00, ext1 mov A, 41H, Cipic A, #100, ext1 mov B, #10 mov B,	div A, B	clr 1H
call delay DIG5: mov A, B mov P1, #0EFH mov P2, A call delay DIG4: mov A, 31H mov B, #10 div A, B mov P1, #0F7H mov P2, A call delay DIG3: mov A, B mov P1, #0F8H mov P2, A call delay DIG3: mov A, B mov P1, #0F8H mov P2, A call delay DIG3: mov A, B mov P1, #0F8H mov P2, A call delay DIG3: mov A, 32H mov P2, A call delay Dig2: mov A, 32H mov P3, #10 div A, B mov P1, #0F0H div A, B mov P1, #0F0H mov P2, A call delay Dig1: mov A, B mov P1, #0F0H mov P2, A call delay dinz R5, midway mov 36H, #0F0H mov P2, A call delay DIG1: mov A, B mov P1, #0FEH mov P2, A call delay dinz R4, midway; Q1 DIG1: mov A, B mov P1, #0FEH mov P2, A call delay dinz R4, midway; Q1 pmp halfcycle; Q2 call delay dinz R6, DIG6 ;mov 36H, #0F0H int1test: ; satisfaction? ret TMR0: push PSW push A mov T10, #>(65536-10000) rrc A mov 3H, C pop PSW inc 32H mov P0, A mov P0, A mov A, 32H cjne A, #100, ext1	mov P1, #0DFH	intOtest:
DIG5: mov A, B	mov P2, A	jnb 0H, norm_cont
mov P1, #0EFH mov P2, A call delay DIG4: mov A, 31H mov B, #10 mov P0, A div A, B mov P1, #0F7H mov P2, A call delay mov P0, A div A, B mov P1, #0F7H mov P2, A call delay DIG3: mov A, B mov P1, #0F8H mov P2, A call delay DIG3: mov A, B mov P1, #0F8H mov P2, A call delay Dig2: mov A, 32H mov B, #10 div A, B mov P1, #0F0H div A, B mov P1, #0F0H mov P2, A call delay Dig1: mov A, 32H mov P1, #0F0H mov P2, A call delay mov P1, #0F0H mov P2, A call delay DIG1: mov A, B mov P1, #0F0H mov P2, A call delay DIG1: mov A, B mov P1, #0F0H mov P2, A call delay DIG1: mov A, B mov P1, #0F0H mov P2, A call delay DIG1: mov A, B mov P1, #0F0H mov P2, A call delay djnz R4, midway; Q1 jmp display_go midway: jmp halfcycle; Q2 call delay djnz R6, DIG6 mov P2, A call delay djnz R6, DIG6 mov P3, A jmp display_go midway: jmp halfcycle; Q2 call delay djnz R6, DIG6 mov A, 41H TMR0: push PSW push A mov C, 3H mov TH0, #>(65536-10000) mov TL0, #<(65536-10000) mov TL0, #<(65536-10000) setb TR0 inc 32H mov P0, A mov A, 32H cjne A, #100, extl	call delay	mov A, 40H
mov P2, A rlc A mov 2H, C	DIG5: mov A, B	push PSW
call delay mov 2H, C DIG4: mov A, 31H pop PSW mov B, #10 mov P0, A div A, B mov 40H, A mov P1, #0F7H jb 2H, int0_in_session mov P2, A clr 0H call delay jmp norm_cont DIG3: mov A, B int1_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 DIG1: mov A, B jmp display_go mov P2, A jmp balfcycle ; Q2 call delay delay: ; ? how long should the inc 32H, #0F0H ; delay be for visual int1test: ; satisfaction? jmb 1H, int0test ret mov A, 4HH TMR0: push PSW mov T0, #<(65536-10000)	mov P1, #0EFH	mov C, 2H
DIG4: mov A, 31H mov B, #10 div A, B mov P1, #0F7H mov P2, A call delay DIG3: mov A, B mov P1, #0FBH mov P2, A call delay mov P2, A call delay mov P3, #0FBH mov P4, #0FBH mov P4, #0FBH mov P5, A call delay mov B, #10 div A, B mov P1, #0FDH mov P1, #0FDH div A, B mov P1, #0FDH mov P2, A call delay dipa R4, halfcycle; Q0 DIG1: mov A, B mov P1, #0FEH mov P2, A call delay dipa R4, midway; Q1 pimp display_go mov P1, #0FEH midway: mov P2, A call delay dipa R4, midway; P1 imp display_go mov P1, #0FEH midway: mov P3, #16 mov P4, #0FOH intlest: intlest:	mov P2, A	rlc A
mov B, #10 mov P0, A div A, B mov 40H, A mov P1, #0F7H jb 2H, int0_in_session mov P2, A clr 0H call delay jmp norm_cont DIG3: mov A, B int1_in_session: mov P1, #0FBH int0_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 pim display_go midway: mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay ; ? how long should the delay djnz R6, DIG6 delay ; ? how long should the delay be for visual int1test: jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #><(65536-10000)	call delay	mov 2H, C
div A, B mov 40H, A mov P1, #0F7H jb 2H, int0_in_session mov P2, A clr 0H call delay jmp norm_cont DIG3: mov A, B int1_in_session: mov P1, #0FBH int0_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 85, #14 div A, B mov 85, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 plf: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay djnz R6, DIG6 ;mov 36H, #0F0H ; delay be for visual intltest: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) mov TH0, #>(65536-10000) mov TH0, #>(65536-10000) mov P0, A mov A, 32H mov P0, A mov A, 32H	DIG4: mov A, 31H	pop PSW
mov P1, #0F7H mov P2, A call delay DIG3: mov A, B mov P1, #0FBH mov P2, A call delay DIG3: mov A, B mov P1, #0FBH mov P2, A call delay Dig2: mov A, 32H mov B, #10 div A, B mov P1, #0FDH mov P2, A call delay Mov P1, #0FDH mov P2, A call delay DIG1: mov A, B mov P1, #0FDH mov P2, A call delay DIG1: mov A, B mov P1, #0FEH mov P2, A call delay DIG1: mov A, B mov P1, #0FEH midway: mov P2, A call delay djnz R4, midway ; Q1 jmp display_go midway: jmp halfcycle ; Q2 call delay djnz R6, DIG6 jmov 36H, #0F0H jmp halfcycle ; Q2 call delay djnz R6, DIG6 jmov 36H, #0F0H jmp halfcycle ; Q2 call delay djnz R6, DIG6 jmov A, 41H TMR0: push PSW push A mov C, 3H mov TH0, #>(65536-10000) mov 3H, C pop PSW mov P0, A mov A, 32H mov A, 32H mov A, 32H cjne A, #100, ext1	mov B, #10	mov P0, A
mov P2, A clr 0H call delay jmp norm_cont DIG3: mov A, B int1_in_session: mov P1, #0FBH int0_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 call delay djnz R4, midway ; Q1 mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the intltest: ; delay be for visual intltest: ; satisfaction? ret TMR0: push PSW push A mov A, 41H push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TH0, # mov TL0, #<(65536-10000)	div A, B	mov 40H, A
call delay jmp norm_cont DIG3: mov A, B int1_in_session: mov P1, #0FBH int0_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 jmp display_go midway: mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the ginz R6, DIG6 delay: ; ? how long should the intltest: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) mov A, 3H mov TL0, #<(65536-10000)	mov P1, #0F7H	jb 2H, int0_in_session
DIG3: mov A, B	mov P2, A	clr 0H
mov P1, #0FBH int0_in_session: mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 call delay djnz R4, midway ; Q1 mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay ; ? how long should the int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) mov C, 3H mov TH0, #>(65536-10000) mov 3H, C setb TR0 mov P0, A mov A, 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	call delay	jmp norm_cont
mov P2, A jmp int_in_session call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 call delay djnz R4, midway ; Q1 mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the ; mov 36H, #0F0H ; delay be for visual intltest: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rcc A mov TH0, #>(65536-10000) mov 3H, C setb TR0 pop PSW inc 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	DIG3: mov A, B	int1_in_session:
call delay norm_cont: Dig2: mov A, 32H djnz R5, midway mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 pimp display_go midway: mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rcc A mov TH0, #<(65536-10000)	mov P1, #0FBH	int0_in_session:
Dig2: mov A, 32H mov B, #10 div A, B mov P1, #0FDH mov P2, A call delay djnz R4, midway; Q1 DIG1: mov A, B mov P1, #0FEH midway: mov P2, A call delay djnz R4, midway; Q1 DIG1: mov A, B mov P1, #0FEH midway: mov P2, A call delay djnz R6, DIG6 call delay djnz R6, DIG6 jmp halfcycle; Q2 call delay djnz R5, midway mov P2, A djnz R4, midway; Q1 jmp display_go midway: jmp halfcycle; Q2 call delay delay: ; ? how long should the ; satisfaction? ret TMR0: push PSW push A mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rrc A mov TH0, #>(65536-10000) mov 3H, C pop PSW inc 32H mov P0, A mov A, 32H cjne A, #100, ext1	mov P2, A	jmp int_in_session
mov B, #10 mov 36H, #0F0H div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 call delay jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay djnz R6, DIG6 ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rrc A mov TH0, #<(65536-10000)	call delay	norm_cont:
div A, B mov R5, #14 mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 DIG1: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	Dig2: mov A, 32H	djnz R5, midway
mov P1, #0FDH ;djnz R4, halfcycle ; Q0 mov P2, A djnz R4, midway ; Q1 DIG1: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the jmov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rcc A mov TL0, #<(65536-10000)	mov B, #10	mov 36H, #0F0H
mov P2, A djnz R4, midway; Q1 DIG1: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle; Q2 call delay djnz R6, DIG6 delay: ;? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A push A mov C, 3H mov TH0, #>(65536-10000) rcc A mov TL0, #<(65536-10000)	div A, B	mov R5, #14
call delay djnz R4, midway ; Q1 DIG1: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the jmz R6, DIG6 delay: ; ? how long should the ; mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	mov P1, #0FDH	;djnz R4, halfcycle ; Q0
DIG1: mov A, B jmp display_go mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay delay: ; ? how long should the delay: ; ? how long should the ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	mov P2, A	
mov P1, #0FEH midway: mov P2, A jmp halfcycle ; Q2 call delay djnz R6, DIG6 delay: ;? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	call delay	djnz R4, midway ; Q1
mov P2, A jmp halfcycle ; Q2 call delay djnz R6, DIG6 delay: ; ? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000) mov 3H, C setb TR0 pop PSW inc 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	DIG1: mov A, B	jmp display_go
call delay djnz R6, DIG6 delay: ;? how long should the ;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	mov P1, #0FEH	midway:
djnz R6, DIG6 delay: ; ? how long should the ; delay be for visual ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rc A mov TL0, #<(65536-10000)	mov P2, A	jmp halfcycle ; Q2
;mov 36H, #0F0H ; delay be for visual int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	call delay	
int1test: ; satisfaction? jnb 1H, int0test ret mov A, 41H TMR0: push PSW push A mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	djnz R6, DIG6	delay: ;? how long should the
jnb 1H, int0test ret mov A, 41H TMR0: push PSW push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	;mov 36H, #0F0H	; delay be for visual
mov A, 41H push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000) mov 3H, C pop PSW mov P0, A mov P0, A mov A, 32H mov 41H, A rine A, #100, ext1	int1test:	; satisfaction?
push PSW push A mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	jnb 1H, int0test	ret
mov C, 3H mov TH0, #>(65536-10000) rrc A mov TL0, #<(65536-10000)	mov A, 41H	TMR0: push PSW
rrc A mov TL0, #<(65536-10000) mov 3H, C setb TR0 pop PSW inc 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	push PSW	push A
mov 3H, C setb TR0 pop PSW inc 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	mov C, 3H	mov TH0, #>(65536-10000)
pop PSW inc 32H mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1	rrc A	mov TL0, #<(65536-10000)
mov P0, A mov A, 32H mov 41H, A cjne A, #100, ext1		setb TR0
mov 41H, A cjne A, #100, ext1	pop PSW	inc 32H
· · · · · · · · · · · · · · · · · · ·	mov P0, A	mov A, 32H
jb 3H, int1_in_session mov 32H, #0	mov 41H, A	cjne A, #100, ext1
	jb 3H, int1_in_session	mov 32H, #0

```
PSW
    inc
          31H
                                                       pop
    mov A, 31H
                                                       reti
          A, #60, ext1
                                                  INT0: jb
                                                               0H, intext0
    cine
          31H, #0
                                                  push A
    mov
                                                             PSW
            Α
ext1: pop
                                                       push
           PSW
                                                       setb
                                                             0H
    pop
    reti
                                                            PSW
                                                       pop
                                                       pop
                                                            Α
                                                  intext0:
TMR1: push PSW
   push A
                                                       reti
                                                  INT1: jb
        TH1, #>(65536-50000)
                                                               1H, intext1
   mov
        TL1, #<(65536-50000)
                                                  push A
   mov
    setb TR1
                                                            PSW
                                                       push
          35H
                                                       setb
                                                            1H
    inc
    mov
          A. 35H
                                                       pop
                                                            PSW
    cjne A, #20, ext2
                                                       pop
                                                            Α
    mov
          35H, #0
                                                  intext1:
          30H
    inc
                                                       reti
    mov A, 30H
    cine A, #60, ext2
                                                       end
    mov 30H, #0
 ext2: pop
             A
```

Observation:

- The code is running well, but some parts of the prepared code need to be fixed to prevent error popup. The addition of delay function will also let user see the sequence easier. All the wanted sequence react to the correct button being pressed in the switch.
- Simple execution flow description: Firstly, calculate the timer value at one point of the time by using the built-in counter in 89c51. The value that will be displayed then has to be calculated in BCD value before being feed into the port that will be responsible for the number display. Next, display the value to appropriate location of the LED module. The next part is the module to control the behavior of 8x1 LED segment if some input is detected or not.
- The bit 0 and 1 in BITMAP is used to handle and temporarily saved the condition of whether an interrupt for the 7-segment LED module is detected or not, when interrupt has been handled, then the value will be reset.
- The bit 2 and 3 in BITMAP is used to handle and temporarily saved the condition of whether an interrupt for the 8x1 LED module is detected or not, when interrupt has been handled, then the value will be reset.
- 40H and 41H is the one that holds the LED pattern for the 8x1 LED module
- Push pop of PSW when handling interrupt is unnecessary due to the push of Acc prior and the pop of Acc post the PSW push pop instruction

- The code written above is not a really bad way of arrangement if the goals is only to write a functioning code. However, by managing each wanted jobs into separate function, the code will be easier to read, edit, and implement in future modification

Modified code:

- Modified code with loop applied to the scanning body:

```
org 0BH
; (1) 4 interrupt events in action
                                                   jmp TMR0
; (2) 1x8 LEDs
                                                   org 13H
; normal ON-OFF? OFF-ON
                                                       jmp Control_int1
  Control_int0 ? L2R scanning
                                                        org 1BH
  Control_int1 ? R2L scanning
                                                       jmp TMR1
; (3) 7-seg LEDs
                                                   start:
; DIG6-5: timing by second (TF1)
                                                        mov sp, #50H
; DIG4-1: timing by 10ms (TF0)
                                                        mov TMOD, #11H
; (4) priority
                                                               TH0, #11011000b;#>(65536-
                                                   10000)
; priority order of the 4
                                                        mov
                                                                TL0, #11110000b;#<(65536-
  interrupts ???
                                                   10000)
                                                               TH1, #00111100b;#>(65536-
                                                        mov
                                                   50000)
; port0: 1x8 discrete LEDs control
                                                                TL1, #10110000b;#<(65536-
                                                        mov
; port1: power-SW control for
                                                   50000)
    7-seg LED module
                                                        mov IE, #8FH
; port 2: pattern control for
                                                        mov IP, #0AH
    timing display on 7-seg LEDs
                                                        mov 30H, #0
; 30H: TMR1 10-2sec-register
                                                        mov 31H, #0
; 31H: TMR0 sec-register
                                                        mov 32H, #0
; 32H: TMR0 10-2sec-register
                                                        mov 33H, #0
; 33H: TMR0 10-2sec time-up flag
                                                        mov 34H, #0
; 34H: TMR1 10-2sec time-up flag
                                                        mov 35H, #0
; 35H: TMR1 counter: 0-19
                                                        clr 0H
; 36H: normal pattern for 1x8 LEDs
                                                   clr
                                                       1H
; BITMAP 0H: Control_int0 flag
                                                   mov 40H, #0FFH
; BITMAP 1H: Control_int1 flag
                                                   mov 41H, #0FFH
; BITMAP 2H: left-to-right Cy
; BITMAP 3H: right-to-left Cy
                                                   setb TR0
; 40H: 1x8 LED pattern for Control_int0
                                                        setb TR1
; 41H: 1x8 LED pattern for Control_int1
                                                        ;mov P1, #0H
org 0
                                                        ;mov P2, #0H
jmp start
                                                        clr C
org 03H
                                                        mov 2H, C
jmp Control_int0
                                                        mov 3H, C
```

display_go:	;mov 36H, #0F0H
	int1test:
normcycle:	jnb 1H, int0test
mov 36H, #0FH	mov A, 41H
mov R4, #2	push PSW
mov R5, #14	mov C, 3H
halfcycle:	rrc A
mov P0, 36H	mov 3H, C
int_in_session:	pop PSW
mov R6, #30	mov P0, A
DIG6: mov A, 30H	mov 41H, A
mov B, #10	jb 3H, int1_in_session
div AB	clr 1H
mov P1, #0DFH	intOtest:
mov P2, A	jnb 0H, norm_cont
call delay	mov A, 40H
DIG5: mov A, B	push PSW
mov P1, #0EFH	mov C, 2H
mov P2, A	rlc A
call delay	mov 2H, C
DIG4: mov A, 31H	pop PSW
mov B, #10	mov P0, A
div AB	mov 40H, A
mov P1, #0F7H	jb 2H, int0_in_session
mov P2, A	clr 0H
call delay	jmp norm_cont
DIG3: mov A, B	int1_in_session:
mov P1, #0FBH	int0_in_session:
mov P2, A	jmp int_in_session
call delay	norm_cont:
Dig2: mov A, 32H	djnz R5, midway
mov B, #10	mov 36H, #0F0H
div AB	mov R5, #14
mov P1, #0FDH	;djnz R4, halfcycle ; Q0
mov P2, A	
call delay	djnz R4, midway ; Q1
DIG1: mov A, B	jmp display_go
mov P1, #0FEH	midway:
mov P2, A	jmp halfcycle ; Q2
call delay	
djnz R6, DIG6	delay: push 2
-	- *

push 3	setb TR1
mov R2, #2	inc 35H
dd1: mov R3, #250	mov A, 35H
djnz R3,\$	cjne A, #20, ext2
djnz R2, dd1	mov 35H, #0
pop 3	inc 30H
pop 2	mov A, 30H
ret	cjne A, #60, ext2
	mov 30H, #0
TMR0: push PSW	ext2: pop 0E0H;A
push 0E0H;A	pop PSW
mov TH0, #11011000b;#>(65536-	reti
10000) mov TL0, #11110000b;#<(65536-	Control_int0: jb 0H, intext0
mov TL0, #11110000b;#<(65536-10000)	push 0E0H;A
setb TR0	push PSW
inc 32H	setb 0H
mov A, 32H	pop PSW
cjne A, #100, ext1	pop 0E0H;A
mov 32H, #0	intext0:
inc 31H	reti
mov A, 31H	Control_int1: jb 1H, intext1
cjne A, #60, ext1	push 0E0H;A
mov 31H, #0	push PSW
ext1: pop 0E0H;A	setb 1H
pop PSW	pop PSW
reti	pop 0E0H;A
	intext1:
TMR1: push PSW	reti
push 0E0H;A	
mov 50000) TH1, #00111100b; #>(65536-	end
mov 50000) TL1, #10110000b; #<(65536-	,

-