

UNIT 7

EXERCISE 8051 INSTRUCTIONS

2012학년 2학기

마이크로 프로세서 실습

■ 단일루프 Delay

Example1_Delay1

```
                ORG      8000H
                MOV      R0,#0FFH
DELAY:          DJNZ     R0,DELAY
```

Cycle Calculations

➡ 1 Cycle

➡ 2 x R0 Cycle

Total Cycle =
1 + 2 x 255 = 511 Cycle

■ 이중루프 Delay

Example1_Delay2

```
ORG      8000H

MOV      R1,#0FFH
DELAY1:  MOV      R0,#0FFH
DELAY2:  DJNZ      R0,DELAY2
          DJNZ      R1,DELAY1
```

Cycle Calculations


$$(1 + 2 \times R0 + 2) \times R1$$

$$\begin{aligned} \text{Total Cycle} &= \\ 1 + ((1 + 2 \times 255 + 2) \times 255) \\ &= 130816 \text{ Cycle} \end{aligned}$$

■ 삼중루프 Delay

Example1_Delay3

```
ORG      8000H

MOV      R2,#0FFH
DELAY3:  MOV      R1,#0FFH
DELAY2:  MOV      R0,#0FFH
DELAY1:  DJNZ     R0,DELAY1
         DJNZ     R1,DELAY2
         DJNZ     R2,DELAY3
```

Cycle Calculations

$$(1 + (1 + 2 \times R0 + 2) \times R1 + 2) \times R2$$

Total Cycle =
 $1 + (1 + (1 + 2 \times 255 + 2) \times 255 + 2) \times 255$
= 33358591 Cycle

■ 사중루프 Delay (1초 만들기)

Example1_Delay4

```
ORG      8000H

        MOV      R2,#08H
DELAY3:  MOV      R1,#0FFH
DELAY2:  MOV      R0,#0E0H
DELAY1:  DJNZ     R0,DELAY1
        DJNZ     R1,DELAY2
        DJNZ     R2,DELAY3

        MOV      R4,#07H
ADJUST2: MOV      R3,#6CH
ADJUST1: DJNZ     R3,ADJUST1
        DJNZ     R4,ADJUST2
        NOP
```

Cycle Calculations

$$1 + (1 + (1 + 2 \times R0 + 2) \times R1 + 2) \times R2$$

$$1 + (1 + 2 \times R3 + 2) \times R4$$

$$\text{Total Cycle} = 920065 + 1534 + 1 = 921600$$

$$11059200\text{Hz} / 12 = 921600$$

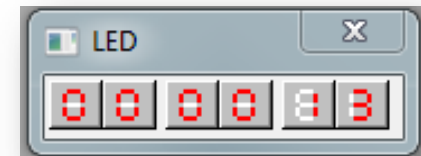
(1 instruction = 12 cycle)

■ 카운터

+ 1초에 1씩 증가하여 60초마다 Port Diode에 왼쪽으로 시프트 하는 카운터 제작

- ASSEMBLE CODE / LED / Port Diode

ORG	8000H	CONDITION:	CJNE	R5,#59H,SEC	
LEDSEG	EQU	OFFC1H	MOV	A,R6	
	CALL	SECCLEAR	MOV	P1,A	
			RL	A	
	MOV	R6,#11111110B	MOV	R6,A	
	MOV	P1,R6	MOV	R5,#00H	
			CALL	SECCLEAR	
			JMP	START	
START:	MOV	R2,#0008H			
DELAY3:	MOV	R1,#00FFH	SEC:	MOV	A,R5
DELAY2:	MOV	R0,#00E0H		ADD	A,#1H
DELAY1:	DJNZ	R0,DELAY1		DA	A
	DJNZ	R1,DELAY2		MOV	R5,A
	DJNZ	R2,DELAY3		CALL	DELAY4
				CALL	DISSEC
				JMP	START
ADJUST2:	MOV	R4,#003H			
ADJUST1:	MOV	R3,#0FAH	DELAY4:	NOP	
	DJNZ	R3,ADJUST1		NOP	
	DJNZ	R4,ADJUST2		RET	
	NOP				
	NOP		SECCLEAR:	MOV	A,#00H
	NOP			CALL	DISSEC
				RET	
			DISSEC:	MOV	DPTR,#LEDSEG
				MOVX	@DPTR,A
				RET	
			END		



- 60 sec 마다 리셋



- 60 sec 마다 왼쪽으로 시프트

■ ASSEMBLE CODE _ Counter

```
                ORG      8000H

LEDSEC  EQU     OFFC1H
                LCALL    SECCLEAR

                MOV      R6,#11111110B
                MOV      P1,R6

START:  MOV      R2,#08H
DELAY3: MOV      R1,#0FFH
DELAY2: MOV      R0,#0E0H
DELAY1: DJNZ     R0,DELAY1
        DJNZ     R1,DELAY2
        DJNZ     R2,DELAY3
```

; 프로그램과 데이터의 시작번지 지정

; SEC 표시 주소

; SEC 표시 초기화

; Port Diode 초기화 (0 : 점등, 1 : 소등)

; 초기 Port Diode 점등

; 1초 타이밍 루틴 시작 Label

■ ASSEMBLE CODE _ Counter

```
        MOV    R4,#003H
ADJUST2: MOV    R3,#0FAH
ADJUST1: DJNZ   R3,ADJUST1
        DJNZ   R4,ADJUST2
        NOP
        NOP
        NOP

CONDITON: CJNE  R5,#59H,SEC
        MOV    A,R6
        MOV    P1,A
        RL     A
        MOV    R6,A
        MOV    R5,#00H
        LCALL  SECCLEAR
        LJMPL  START
```

; 추가적으로 필요한 주기 조절

; 초 단위 비교 (59와 비교) 틀리면 SEC로 분기

; 1분 마다 Port Diode 점등
; Port Diode 왼쪽으로 시프트

; 초 단위 카운터 초기화
; 00:00:59 -> 00:00:00
; START로 분기

■ ASSEMBLE CODE _ Counter

```
SEC:    MOV  A,R5
        ADD  A,#1H
        DA   A
        MOV  R5,A
        LCALL DELAY4
        LCALL DISSEC
        LJMP START
```

```
DELAY4: NOP
        NOP
        RET
```

; 초 단위 1씩 증가
; 십진수 표현

; 주기 조절을 위한 DELAY4로 분기
; 초 표시 루틴 호출
; START로 분기

; 59초가 아닌 경우와 맞는 경우의
Delay 시간을 갖게 하기 위한 Delay 루틴

■ ASSEMBLE CODE _ Counter

```
SECCLEAR: MOV    A,#00H  
          LCALL  DISSEC  
          RET
```

; 초 표시 변수 초기화
; 초 표시 루틴 호출

```
DISSEC:   MOV    DPTR,#LEDSEC  
          MOVX   @DPTR,A  
          RET
```

; 초 표시

```
END
```

; 어셈블러 작업 종료

■ ASSEMBLE CODE _ Counter

```
ORG      8000H

LEDSEC   EQU   OFFC1H
LCALL    SECCLEAR

MOV      R6,#11111110B
MOV      P1,R6

START:   MOV      R2,#08H
DELAY3:  MOV      R1,#0FFH
DELAY2:  MOV      R0,#0E0H
DELAY1:  DJNZ     R0,DELAY1
         DJNZ     R1,DELAY2
         DJNZ     R2,DELAY3
```

■ Cycle Calculations

- $1 + ((1 + (1 + 2 \times R0 + 2) \times R1 + 2) \times R2)$
- = 920065 Cycle이 걸림
- 1초는 921600 Cycle 이므로 1535 Cycle이 더 필요하다.

■ ASSEMBLE CODE _ Counter

```
        MOV    R4,#003H
ADJUST2: MOV    R3,#0FAH
ADJUST1: DJNZ   R3,ADJUST1
        DJNZ   R4,ADJUST2
        NOP
        NOP
        NOP

CONDITON: CJNE  R5,#59H,SEC
        MOV    A,R6
        MOV    P1,A
        RL     A
        MOV    R6,A
        MOV    R5,#00H
        LCALL  SECCLEAR
        LJMPL  START
```

■ Cycle Calculations

- $1 + (1 + 2 \times R3 + 2) \times R4$
=> 1510 + 3(NOP) Cycle이 걸림
- 나머지 22Cycle은 1초를 표시하는 루틴에서 처리함

■ ASSEMBLE CODE _ Counter

```
SEC:    MOV  A,R5
        ADD  A,#1H
        DA   A
        MOV  R5,A
        LCALL DELAY4
        LCALL DISSEC
        LJMP START
```

```
DELAY4: NOP
        NOP
        RET
```

■ Cycle Calculations

+ Case 1

CJNE(59가 아니면) -> SEC로 분기 -> ADD

-> DA -> MOV -> DISSEC -> MOV -> MOVX

-> RET -> START로 분기

+ Case1 Cycle = 22 Cycle

■ ASSEMBLE CODE _ Counter

```
SECCLEAR: MOV    A,#00H
           LCALL  DISSEC
           RET

DISSEC:    MOV    DPTR,#LEDSEC
           MOVX   @DPTR,A
           RET

END
```

■ Cycle Calculations

+ Case 2

CJNE(590이면) -> MOV -> MOV -> RL -> MOV

-> MOV -> SECCLEAR로 분기 -> MOV ->

DISSEC로 분기 -> MOV -> MOVX -> RET -> RET

-> START로 분기

+ Case2 Cycle = 22 Cycle

$$\begin{aligned} \text{+ Total Cycle} &= 920065 + 1513 + 22 \\ &= 921600 \end{aligned}$$

THANK YOU