

3부

함께 실험하기

실험을 하는 목적이 무엇이며, 어떻게 해야 할까요?

- 프로그램 작성법 이해
- 어셈블리의 이해
- 회로의 동작 제어
- 응용 회로

제9장 실험에 들어가기

1. 제품 개발 순서

- 기획, 수요조사, 설계(하드웨어+소프트웨어), 시험생산(시제품), TEST, 생산, 판매
- 제품 개발 시 하드웨어 비중보다 소프트웨어 비중이 더 커짐
: 개발비용 감소, 융통성 많음, 생산비용 감소
- 하드웨어의 역할을 소프트웨어로 구현 --> 마이크로프로세서

마이크로프로세서는 하드웨어를 소프트웨어로 구현하는 것으로
하드웨어와 소프트웨어 두 가지 다 알아야하는 분야 임.

2. Software Development Process

- 1) **Design** : block diagram, flow chart
 - 2) **Coding** : editor (source program : high level language)
 - 3) **Translation** : source program -> object code (low/machine code)
compiler, assembler,
linker
loader
 - 4) **Testing** :
Software test : simulator
Software/Hardware test : emulator, ICE(in circuit emulator)
MDS(microprocessor development systems)
 - 5) **Debugging** : Step, Break points, Modify register
-

3. Source program 작성법

Source program은 high level language 이며, 따라서 특정한 규칙에 맞도록 작성해야 한다.

1) 기본형식

label	mnemonic	operand	comment
-------	----------	---------	---------

- ① label : source 문장의 첫 번째 행에서 시작해야함
처음은 반드시 문자여야 함
사용 가능 문자 31자(A~Z, a~z, 0~9, _ , ?)
- ② mnemonic : operation code로 첫 번째 행에서는 사용하면 안됨
instruction set에서 주어지는 것
assembler directives (pseudo instructions)
- ③ operand : mnemonic에 따르는 symbol, 상수로 1개 이상 시는 ';'로 구분, 상수를 변수로 지정할 경우는 첫 문자는 반드시 문자여야 함.
- ④ comment : assemble 되지 않으며, 작성자가 임의로 사용하는 영역
기호는 ';'으로 이 기호 뒤의 한 줄 문장은 comment로 처리됨.

☞ 참고: operand의 상수는 16진, 10진, 2진, character 등을 사용할 수 있으며, 연산도 가능하다.

type	표 현	예
10진수	D'10진수', .10진수	D'123', .234
16진수	H'16진수', 0x16진수, 16진수H	H'123', 0x3F, 0F4H^,78
2진수	B'2진수'	B'1010'
문자	'문자'	'w'

표 1. MPASM에서의 진수 표현

☞ 주의점 : 16진수 표현 중 16진수H 는 16진수의 첫 시작 값이 문자이면 0 을 앞에 추가해야 함, 0이 없으면 변수 값이 되어 'UNDEFINED

VARIABLE' 이라는 error message가 나옴.(16진 숫자 E5는 E5H가 아니고 0E5H임)

2) 기본적으로 사용되는 directive

- ① LIST : list file option으로 특히 마이크로프로세서의 TYPE를 선언함.

```
list p=<type>
```

- ② PROCESSOR : 마이크로프로세서의 TYPE를 선언함. LIST와 동일

```
processor 16f84
```

- ③ ORG : origin의 의미로 프로그램이 수행되는 시작번지를 의미한다.

```
<label> org < expression >
```

- 프로세서의 종류에 따라서 전원 투입 시 시작되는 번지가 다르다.
- 전원 투입 시 마이크로프로세서는 초기화가 되며, 이를 reset이라고 함.
- 본 실험에 사용하는 마이크로프로세서 reset 후 0000 번지부터 시작함.
- 따라서 프로그램 시작 위치에 ORG 0000 이라고 써 주어야 한다.

- ④ INCLUDE : 주어진 file를 삽입하여 사용하라.

```
include <<include_file>> : 첫 번째 열부터 시작함  
include "c:\mr_pic\16f84.h" OR <PIC16F84.h>
```

- ⑤ EQU : equal의 의미로 사용되며, 선언된 label 값이 expression이 된다.

```
<label> equ <expr> : 첫 번째 열부터 시작함
```

- 프로그램을 일반화 시켜서 작성 시에 도움이 많이 된다.
- 따라서 EQU를 많이 선언할 것.

- ⑥ END: end의 의미로 compiler에게 번역을 끝내라는 의미로 사용된다.

-
- 항상 source program의 끝에 위치해야한다.
 → 이 명령어 다음에 프로그램이 들어 있어도 번역하지 않는다.

⑦ IF: ELSE: ENDIF : if 문의 시작과 끝을 나타냄

```
if <expr> ~
    else if OR else ~
endif
```

⑧ MACRO: ENDM : macro 정의 시작과 끝을 나타냄

```
<label> macro <<arg>,,,<arg>> ~ endm
```

⑨ EXITM : macro 문에서 빠져나올 때 사용

예) test macro num, count

```
if num==count
    movlw 1
    exitm
else
    movlw 0
endif
    movlw 4
endm
```

• PC 컴퓨터상에서 여러분이 작성한 프로그램이 어떻게 변환되어 수행되는가를 알아보도록 합시다.

-source program, compiler, assembler, linker, object code, 수행
