

전자지향 프로그래밍  $\Rightarrow$  C 언어..

임베디드 (IoT) 개발 / 컴퓨터 개발

$\updownarrow$

$\Rightarrow$  H/W 제어  
 $\Rightarrow$  H/W 드라이버 만들기.

범용

웹, 알고리즘 (X)

코딩 방법 (O)

Win 7 OS

32bit

64bit

RAM 용량  
아무?

4GB

1bit 0, 1

2bit 00, 01, 10, 11

3bit 8 EA

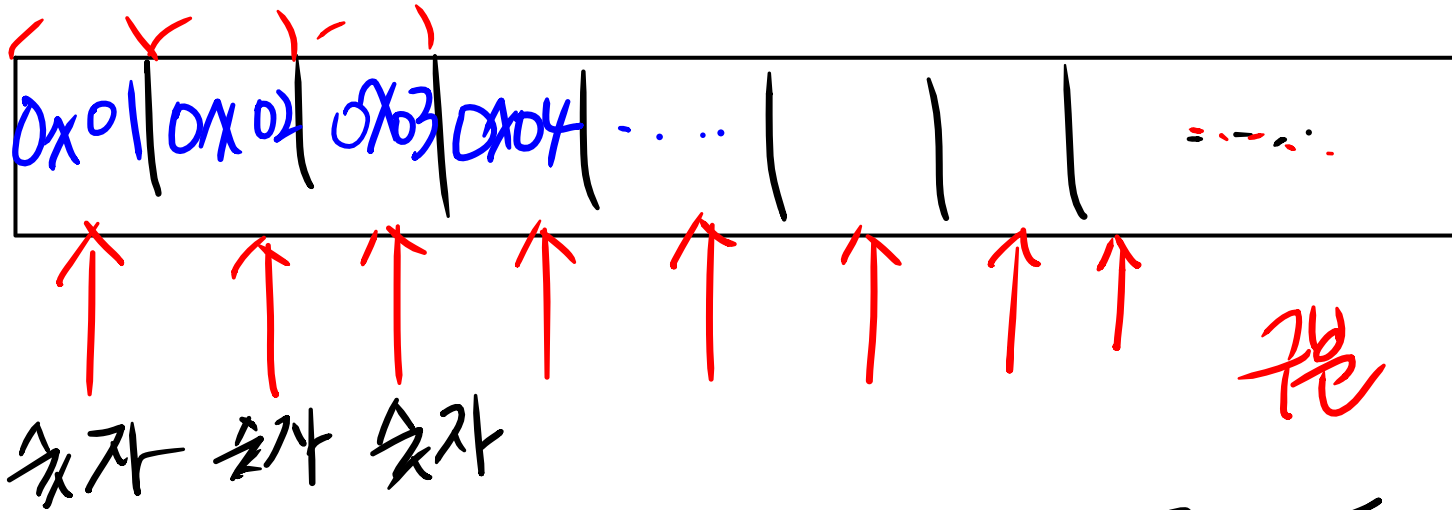
4bit 16 EA

nbit  $2^n$

$2^{32}$

Memory Address  
길이, 32)

Memory  
1 byte = 8 bit



Win 32 bit OS  
주소값의 범위

`0x00000000`  
~ `0xffffffff`

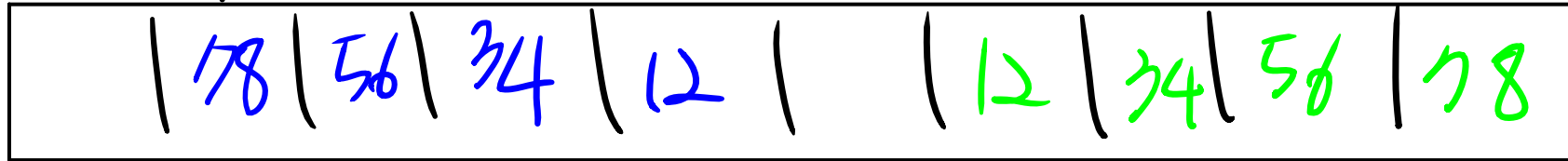
Memory 주소 특징

- ① 연속적이다.
- ② 고유하다.

0x12345678    값 (data)

MSB    LSB

Memory



0x01 0x02 0x03 ...

0x10

저장

Data (값)이 Memory에 저장되는 방식 (방향)

낮은 주소부터

높은 주소 저장 (작)

Little-Endian

..

높은 주소 저장 (큰)

Big-Endian

Byte-order

2 p32b

메모리 1		
주소: 0x012FFA9C		
0x012FFA9C	cc	?
0x012FFA9D	cc	?
0x012FFA9E	cc	?
0x012FFA9F	cc	?
0x012FFAA0	cc	?
0x012FFAA1	cc	?
0x012FFAA2	cc	?
0x012FFAA3	cc	?
0x012FFAA4	cc	?
0x012FFAA5	cc	?
0x012FFAA6	cc	?
0x012FFAA7	cc	?
0x012FFAA8	cc	?
0x012FFAA9	cc	?
0x012FFAAA	cc	?

8 p32b

메모리 1		
주소: 0x012FFA9C		
0x012FFA9C	a8	?
0x012FFA9D	fa	?
0x012FFA9E	2f	/
0x012FFA9F	01	.
0x012FFAA0	cc	?
0x012FFAA1	cc	?
0x012FFAA2	cc	?
0x012FFAA3	cc	?
0x012FFAA4	cc	?
0x012FFAA5	cc	?



2 data

메모리 1		
주소: 0x012FFAA8		
0x012FFAA8	78	x
0x012FFAA9	56	V
0x012FFAAA	34	4
0x012FFAAB	12	.
0x012FFAAC	cc	?
0x012FFAAD	cc	?
0x012FFAAE	cc	?
0x012FFAAF	cc	?

0x012ffa8

자료형

⇒ Data type

⇒ Address type

⇒ Address size

⇒ Cpu가 메모리를 바라보는 메모리 양  
(크기)

⇒ Cpu가 메모리의  
Read/write를 할지?

⇒ Cpu가 바라보는 메모리 유효범위

이름	값	형식
*p16b	0x5678 '噸'	wchar_t
*p32b	0x12345678	unsigned int
*p8b	0x78 'x'	unsigned char
p16b	0x00eff8d0 L"噸ㄴ첼첼@  iOa\x1"	wchar_t *
p32b	0x00eff8d0 (0x12345678)	unsigned int *
p8b	0x00eff8d0 <문자열에 잘못된 문자가 있습니다...>	unsigned char *

Microsoft Visual Studio 디버그 X + v

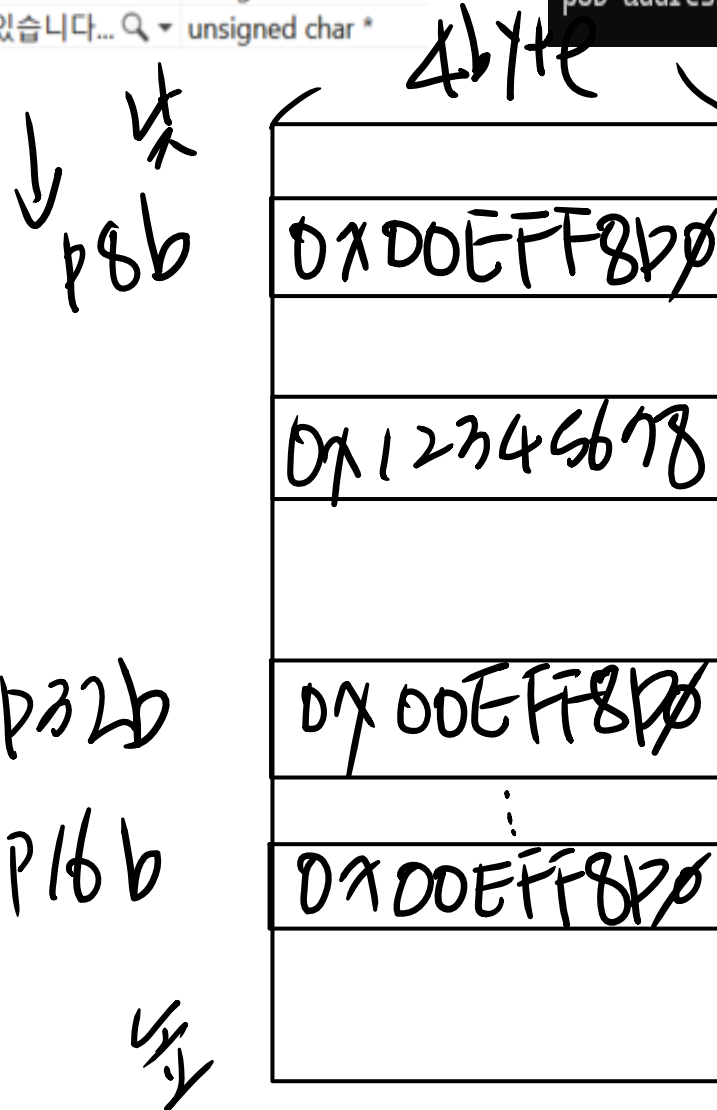
p32b address : 012FFCDC , p32b data : 12345678

p16b address : 012FFCDC , p16b data : 5678

p8b address : 012FFCDC , p8b data : 78

*2 data*

주소	0x00EFF8D0
0x00EFF8D0	78 x
0x00EFF8D1	56 v
0x00EFF8D2	34 4
0x00EFF8D3	12 .
0x00EFF8D4	cc ?
0x00EFF8D5	cc ?
0x00EFF8D6	cc ?
0x00EFF8D7	cc ?
0x00EFF8D8	65 e
0x00EFF8D9	32 2
0x00EFF8DA	45 E
0x00EFF8DB	17 .



*0x00EFF8D0 ~ 3*

임베디드 SW 개발

(컴퓨터) ⇒ ex) BIOS

COE.

프로세스?

Main 함수

① 프로그램의 시작과 끝

② 프로그램의 처리 (기능, 변수)

③ 프로그램의 실행

AP



```

int a;
int b, c;
double da;
char ch;

```

```

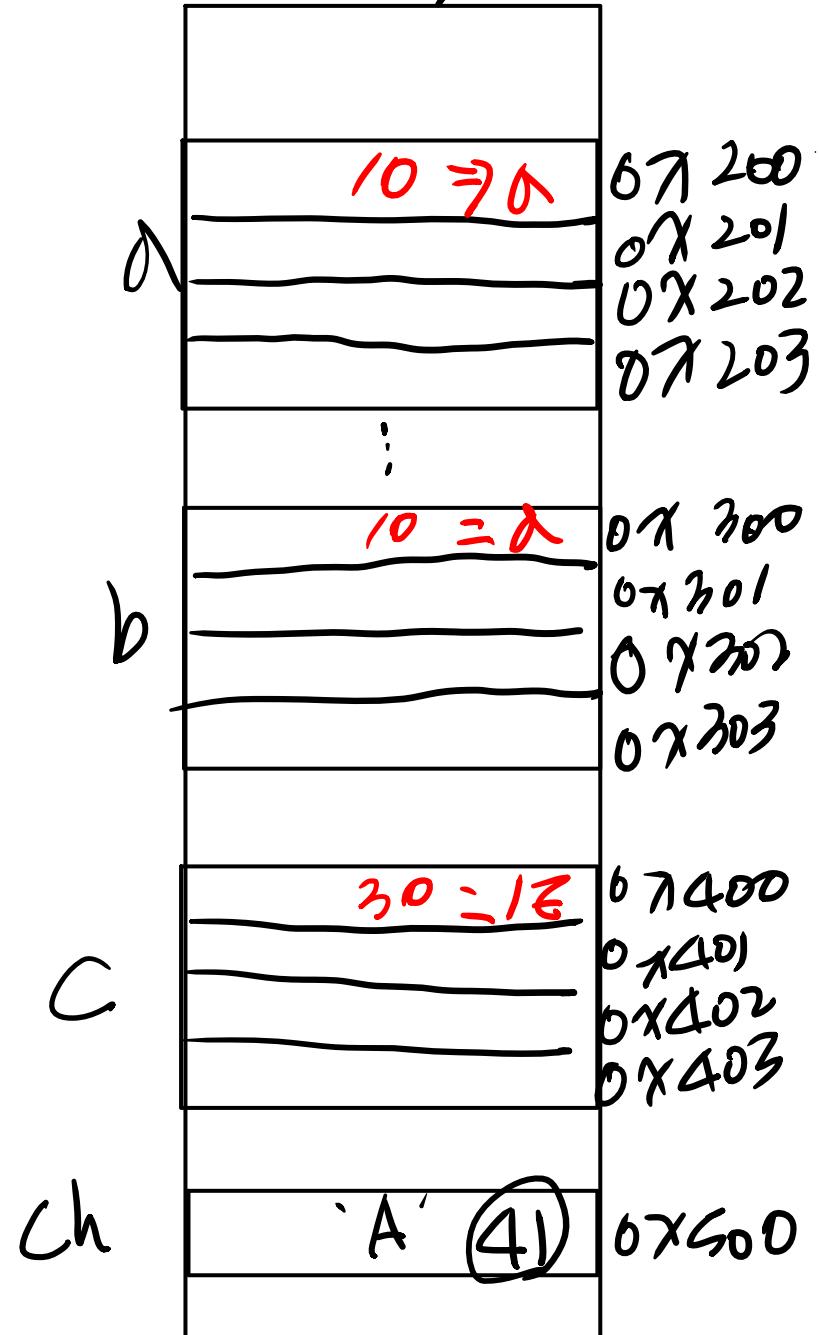
a = 10;
b = a;
c = a + 20;
da = 3.5;
ch = 'A';

```

↑  
변수 값

↑  
변수 or 상수 값

Memory



자료형

크기 (byte)

비고 (서식 문자)

Char  
Short  
int  
float  
double  
long  
long long

1

2

4

4

8

4

8

문자 %c

) 정수형 %hd  
%d

) 실수형 %f  
%lf

) 정가정 %ld

%lld

문자 'A'

문자열 "A"

⇒ (주소) 상수

```
int a = 10, b = 20;
```

```
(a == b)
```

```
char array1[]
```

```
= "banana";
```

```
char array2[]
```

```
= "apple";
```

```
array1 == array2
```

주요 상수 구조 set

1. 문자열

2. 배열

3. 함수

fruit = "banana"

$$O = \overset{==}{O} + O$$

변수

자료형 int, double

문자 if, for  
while

sizeof  
typedef  
volatile

식별자

int age

scanf ("서식문자", &변수명);