- **날　　짜** : 2014년 2월 18일
- **제　　목** : 16bit MCU의 Upper Byte ROM 배열 사용법
- **문서번호** : KR-ES-0205
- **작 성 자** : 지영우 차장 ( wayne.ji@microchip.com , Senior FAE )

## < Engineering Issue 내용 >

### 1. 관련 Device : Microchip 16bit MCU ( PIC24/PIC30/PIC33 )

### 2. 개요

Microchip 16bit MCU는 Program 명령어 구조가 24bit(3byte)로 되어 있기 때문에 일반적인 ROM배열을 사용할 경우, 16bit 영역만을 사용하고, 상위 8bit는 사용하기 힘든 구조로 되어 있다. 이러한 이유로 ROM 배열을 사용할 때 프로그램 메모리의 낭비를 가지고 올 수 있는데 이를 방지하기 위해서는 Compiler(XC16)에서 제공하는 "__pack__upper__byte"라는 수식어를 이용하여 Program Memory(24bit) 전 영역을 사용할 수 있다.

### 3. 사용 예

__pack_upper_byte char PackUpper_Array[] = { 'P', 'a', 'c', 'k', 'U', 'p', 'p', 'e', 'r', 'A', 'r', 'r', 'a', 'y' };

또는

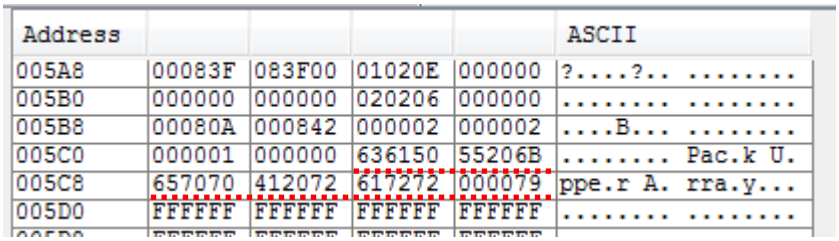
__pack_upper_byte char PackUpper_Array[] = "Pack Upper Array₩n";

### 4. 예제 코드

```
#include <stdio.h>
__pack_upper_byte char PackUpper_Array[] = "Pack Upper Array";
const char ROM_Array[] = { 'R', 'O', 'M', ' ', 'A', 'r', 'r', 'a', 'y' };
unsigned char RAM_Array[] = { 'R', 'A', 'M', ' ', 'A', 'r', 'r', 'a', 'y' };

int main()
{
    int i=0;
    while (i < sizeof(PackUpper_Array)) {
        printf("%c", PackUpper_Array[i]);
        ++i;} printf("₩n");

    i = 0;
    while (i < sizeof(ROM_Array)) {
        printf("%c", ROM_Array[i]);
        ++i;} printf("₩n");

    i = 0;
    while (i < sizeof(RAM_Array)) {
        printf("%c", RAM_Array[i]);
        ++i; } printf("₩n");

    return 0;
}
```

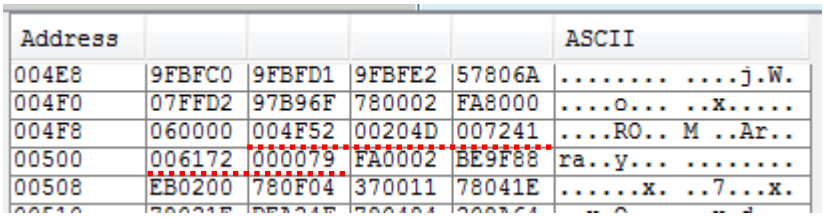
**5. Program Memory 비교 ( “__pack_upper_byte” vs “const” )**

**<** __pack_upper_byte char PackUpper_Array[] = "Pack Upper Array"; **>**



| Address | | | | | ASCII |
|---|---|---|---|---|---|
| 005A8 | 00083F | 083F00 | 01020E | 000000 | ?....?.. ........ |
| 005B0 | 000000 | 000000 | 020206 | 000000 | ........ ........ |
| 005B8 | 00080A | 000842 | 000002 | 000002 | ....B... ........ |
| 005C0 | 000001 | 000000 | 636150 | 55206B | ........ Pac.k U. |
| 005C8 | 657070 | 412072 | 617272 | 000079 | ppe.r A. rra.y... |
| 005D0 | FFFFFF | FFFFFF | FFFFFF | FFFFFF | ........ ........ |

**<** const char ROM_Array[] = { 'R', 'O', 'M', ' ', 'A', 'r', 'r', 'a', 'y' }; **>**



| Address | | | | | ASCII |
|---|---|---|---|---|---|
| 004E8 | 9FBFC0 | 9FBFD1 | 9FBFE2 | 57806A | ........ ....j.W. |
| 004F0 | 07FFD2 | 97B96F | 780002 | FA8000 | ....o... ..x..... |
| 004F8 | 060000 | 004F52 | 00204D | 007241 | ....RO.. M ..Ar.. |
| 00500 | 006172 | 000079 | FA0002 | BE9F88 | ra..y... ........ |
| 00508 | EB0200 | 780F04 | 370011 | 78041E | ......x. ..7...x. |