

Preprocessing (2K)

Copyright (c) 2014 - 2018 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Based on Embedded Software in C for an ARM Cortex M
<http://users.ece.utexas.edu/~valvano/Volume1/>

Structure Declarations (1)

```
#define TriggerPendSV()      { NVIC_INT_CTRL_R = 0x10000000; }
#define SetPA5(x)           { GPIO_PORTA_DATA_R = (GPIO_PORTA_DATA_R & ~0x20) | (x) };
#define Wait(t)             { int wait; for (wait = 0; wait < (t); wait++) { } };
#define CONVERT4BPP(c)     ( ((c) << 12) | ((c) << 7) | ((c) << 1) )
```

Macro

```
#ifdef ABC
... /* ABC */
#ifndef DEF
... /* ABC and not DEF */
#else
... /* ABC and DEF */
#endif
... /* ABC */
#else
... /* not ABC */
#endif HIJ
... /* not ABC but HIJ */
#endif
... /* not ABC */
#endif
```

```
#ifdef ABC
... /* ABC */
#ifndef DEF
... /* ABC and not DEF */
#else
... /* ABC and DEF */
#endif
... /* ABC */
#else
... /* not ABC */
#endif HIJ
... /* not ABC but HIJ */
#endif
... /* not ABC */
#endif
```

Conditional Compilation

```
#define Debug 1

int Sub(int j){ int i;
#ifdef Debug
    PORTC |= 0x80; /* PC7 set when Sub is entered */
#endif
    i=j+1;
#ifdef Debug
    PORTC &= ~0x80; /* PC7 cleared when Sub is exited */
#endif
    return(i);
}

void Program() {
    int i;
#ifdef Debug
    PORTC |=0x40; /* PC6 set when Program is entered */
#endif
    i=Sub(5);
    while(1) { PORTB=2; i=Sub(i);}
}

void ProgB() { int i;
    i=6;
#ifdef Debug
    PORTC &= ~0x40; /* PC6 cleared when Sub is exited */
#endif
}
```

Inline Assembly

```
__asm void  
Delay(unsigned long ulCount) {  
    subs    r0, #1  
    bne     Delay  
    bx     lr  
}
```

```
void Delay(unsigned long ulCount) {  
    __asm ( "    subs    r0, #1\n"  
           "    bne     Delay\n"  
           "    bx     lr\n"           );  
}
```

References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun
- [5] “A Whirlwind Tutorial on Creating Really Teensy ELF Executables for Linux”
<http://cseweb.ucsd.edu/~ricko/CSE131/teensyELF.htm>
- [6] <http://en.wikipedia.org>
- [7] <http://www.muppetlabs.com/~breadbox/software/tiny/teensy.html>
- [8] <http://csapp.cs.cmu.edu/public/ch7-preview.pdf>