Variables (2D)

Young Won Lim
3/28/18

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/

Initialization

```
short MyVariable;                /* variable allows read/write access */
const short MyConstant=50;       /* constant allows only read access */
#define fifty 50

void main(void) {
    MyVariable=50;               /* write access to the variable */
    OutSDec(MyVariable);         /* read access to the variable */
    OutSDec(MyConstant);         /* read access to the constant */
    OutSDec(50);                 /* "50" is a literal */
    OutSDec(fifty);              /* fifty is also a literal */
}
```

Global Variables

```
long TheGlobal;                    /* a regular global variable*/

void main(void) {
  TheGlobal = 1000;
}


    LDR R0,=1000
    LDR R1,=TheGlobal              // address
    STR R0,[R1]
```

Static Variables

```
static short TheGlobal;                /* a static global variable*/

void main(void) {
  TheGlobal = 1000;
}


void main(void) {
  static stort TheLocal;               /* a static local variable*/

  TheLocal = 1000;
}
```

Static Local Variables

```
void function1(void) {
  static short TheCount;
  TheCount = TheCount+1;
}

void function2(void) {
  static short TheCount;
  TheCount = TheCount+1;
}
```

Volatile

```
volatile unsigned long Time;

void SysTick_Handler(void) {       /* every 16ms */
  Time = Time+1;
}

void main(void){
  SysTick_Init();
  Time = 0;
  while (Time<100) { };          /* wait for 100 counts of the 16 ms timer*/
}


Time = 0;
while (Time<100) { };
```

Automatic

```
unsigned char data[100];
#define GPIO_PORTA_DATA_R       (*((volatile unsigned long *) 0x400043FC))

void Collect(void) {
  short i;
  for (i=0;i<100;i++) {                           /* collect 100 measurements */
    data[i] = GPIO_PORTA_DATA_R;      /* collect ith measurement */
  }
}


int *BadFunction(void) {
   int z;
   z = 1000;
   return (&z);                                   // returning the address of an automatic variable
}
```

Automatic Variables

```
void fun(void) {
    long y1,y2,y3;              /* 3 local variables*/
    y1 = 1000;
    y2 = 2000;
    y3 = y1+y2;
}
```

```
fun SUB SP,#12                  ; allocate3 local variables
;  y1 = 1000
    LDR R0,=1000
    STR R0,[SP,#0]
; y2 = 2000
    LDR R0,=2000
    STR R0,[SP,#4]
; y3 = y1+y2
    LDR R0,[SP,#0]      ; y1
    LDR R1,[SP,#4]      ; y2
    ADD R2,R0,R1
    STR R2,[SP,#8]      ;set y3
    ADD SP,#12          ;deallocate
    BX  LR
```

Constant Local

```
short TheGlobal;                        /* a regular global variable*/

void main(void){
    const short TheConstant=1000;       /* a constant local*/

    TheGlobal=TheConstant;
}
```

External

```
extern short ExtGlobal;                /* an external global variable*/

void main(void){
    ExtGlobal=1000;
}
```

Variable Scope

```
unsigned char x;                          /* a regular global variable*/

void sub(void) {
    x=1;
    {   unsigned char x;                  /* a local variable*/
        x=2;
        {   unsigned char x;              /* a local variable*/
            x=3;
            PORTA=x;
        }
        PORTA=x;
    }
    PORTA=x;
}
```

Scope

```
void sub(void) {
    int x;               /* a valid local variable declaration */
    x=1;

    int y;               /* This declaration is improper */
    y=2;
}
```

Declaration

Declaration	Comment	Range
unsigned char uc;	8-bit unsigned number	0 to +255
char c1,c2,c3;	three 8-bit signed numbers	-128 to +127
unsigned int ui;	32-bit unsigned number	0 to +4294967296
int i1,i2;	two 32-bit signed numbers	-2147483648L to 2147483647L
unsigned short us;	16-bit unsigned number	0 to +65535
short s1,s2;	two 16-bit signed numbers	-32768 to +32767
long l1,l2,l3,l4;	four signed 32 bit integers	-2147483648L to 2147483647L
unsigned long ui;	32-bit unsigned number	0 to +4294967296
float f1,f2;	two 32-bit floating numbers	not recommended
double d1,d2;	two 64-bit floating numbers	not recommended

Storage Classes

Modifier	Comment
auto	automatic, allocated on the stack
extern	defined in some other program file
static	permanently allocated
register	attempt to implement an automatic using a register instead of on the stack

Modifiers

Modifier	Comment
volatile	can change value by means other than the current program
const	fixed value, defined in the source code and can not be changed during execution
unsigned	range starts with 0 includes only positive values
signed	range includes both negative and positive values

Const Modifier

```
void LegalFuntion(short in) {
  while (in) {
    UART_OutChar(Ret);
    in--;
  }
}

void NotLegalFuntion(const short in) {
  while (in){
    UART_OutChar(13);
    in--;                                          // this operation is illegal
  }
}

void NotLegalFuntion2(void) {
  const short count=5;
  while (count) {
    UART_OutChar(13);
    count--;                                       // this operation is illegal
  }
}
```

Promotion

```
char            x;          /* signed 8 bit global */
unsigned short  y;          /* unsigned signed 16 bit global */

void sub(void) {
   y=y+x;
                            /* x treated as unsigned even though defined as signed */
}
```

Initialization of Global Variables

```
short I;                    /* 16 bit global */
const short J=96;           /* 16 bit constant */
#define K 97;

void main(void) {
    I=J;
    I=K;
}
```

```
/* poor style */            /* good style */
int I=95;                   int I;
void main(void) {           void main(void) {
                               I=95;
}                           }
```

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