Literals (2C)

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/

Decimal Number Ranges

type	range	precision	examples
unsigned char	0 to 255	8 bits	0 10 123
char	-128 to 127	8 bits	-123 0 10 +10
unsigned int	0 to 4294967295	32 bits	0 2000 2000 50000000L
int	-2147483648 to 2147483647	16 bits	-1000 0 1000 +20000
unsigned short	0 to 65535U	16 bits	0 2000 2000U 50000U
short	-32768 to 32767	16 bits	-1000 0 1000 +20000
long	-2147483648 to 2147483647	32 bits	-123456L 0L 1234567L
unsigned long	0 to 4294967295	32 bits	0L 12345678L

Decimal Number Ranges

type	6811/6812	Cortex M
unsigned char	8 bits	8 bits
char	8 bits	8 bits
unsigned int	16 bits	32 bits
int	16 bits	32 bits
unsigned short	16 bits	16 bits
short	16 bits	16 bits
long	32 bits	32 bits

Decimal Number Ranges

```
short           I;      // 16-bit
unsigned short  J;      // 16-bit
char            K;      // 8-bit
unsigned char   L;      // 8-bit
long            M;      // 32-bit

void main(void) {
   I=97;        /* 16 bits   0x0061       */
   J=97;        /* 16 bits   0x0061       */
   K=97;        /* 8 bits    0x61         */
   L=97;        /* 8 bits    0x61         */
   M=97;        /* 32 bits   0x00000061    */
}
```

Octal Number Ranges

type	range	precision	examples
unsigned char	0 to 0377	8 bits	0 010 0123
char	-0200 to 0177	8 bits	-0123 0 010 +010
unsigned short	0 to 0177777	16 bits	0 02000 0150000U
short	-0100000 to 077777	16 bits	-01000 0 01000 +020000
long	-020000000000 to 017777777777	32 bits	-01234567L 0L 01234567L

Hexdecimal Numbers

Hex Digit	Decimal Value	Binary Value
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A or a	10	1010
B or b	11	1011
C or c	12	1100
D or d	13	1101
E or e	14	1110
F or f	15	1111

Hexa Number Notations

environment	binary format	hexadecimal format	decimal format
Freescale assembly language	%01111010	$7A	122
Intel and TI assembly language	01111010B	7AH	122
C language	-	0x7A	122

Hexadecimal Number Ranges

type	range	precision	examples
unsigned char	0x00 to 0xFF	8 bits	0x01 0x3a 0xB3
char	-0x80 to 0x7F	8 bits	-0x01 0x3a -0x7B
unsigned short	0x0000 to 0xFFFF	16 bits	0x22 0xabcd 0xF0A6
short	-0x8000 to 0x7FFF	16 bits	-0x1234 0x0 +0x7abc
long	-0x80000000 to 0x7FFFFFFF	32 bits	-0x1234567 0xABCDEF

Character Literals

```
short                   I;
unsigned short          J;
char                    K;
unsigned char           L;
long                    M;

void main(void) {
   I   ='a';        /* 16 bits 0x0061 */
   J   ='a';        /* 16 bits 0x0061 */
   K   ='a';        /* 8 bits 0x61 */
   L   ='a';        /* 8 bits 0x61 */
   M   ='a';        /* 32 bits 0x00000061 */
}
```

Character Literals

```
char *pt;
extern void Foo(char *p);

void main (void) {
    pt  ="Jon";      /* pointer to the string, address */
    Foo(pt);         /* passes the pointer not the data itself */
}


char letter, *pt;

void main(void){
    pt          ="A";          /* pointer to the string, address */
    letter      ='A';  /* the data itself ('A' ASCII 65=$41) */
}
```

Escape Sequence

sequence	name	value
\n	newline, linefeed	0x0A = 10
\t	tab	0x09 = 9
\b	backspace	0x08 = 8
\f	form feed	0x0C = 12
\a	bell	0x07 = 7
\r	return	0x0D = 13
\v	vertical tab	0x0B = 11
\0	null	0x00 = 0
\"	ASCII quote	0x22 = 34
\\	ASCII back slash	0x5C = 92
\'	ASCII single quote	0x27 = 39

octal-escape-sequence:
\ octal-digit
\ octal-digit octal-digit
\ octal-digit octal-digit octal-digit

hexadecimal-escape-sequence:
\x hexadecimal-digit
hexadecimal-escape-sequence hexadecimal-digit

```
printf("\tJon\n");
printf("\11Jon\12");
printf("\011Jon\012");
```

```
printf("\tJon\n");
printf("\tJon\n");
```

https://stackoverflow.com/questions/14807036/printing-the-octal-number-through-printf-command

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