



Shakthi Kannan

GNU Free Documentation License

May 2007

Embedded C Program

```
void main ()  
{  
    while (1)  
    {  
        /* just do it */  
    }  
}
```

Data Types and Sizes

Type	Size (byte)	Range
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	-128 to 127
int	4	-2147483648 to 2147483647
short int	2	-32768 to 32767
unsigned int	4	0 to 4294967295
signed int	4	-2147483648 to 2147483647
long int	4	-2147483648 to 2147483647
unsigned long int	4	0 to 4294967295
signed long int	4	-2147483648 to 2147483647

Bitwise operations

AND &

OR |

Exclusive OR ^

Left Shift <<

Right Shift >>

1's Complement ~

Test Bit(s)

```
if (*ADDRESS & MASK)
```

Test Bit(s)

```
/* Hardware register definition */
```

```
typedef volatile unsigned int AT91_REG;
```

```
/* (PIOB) Pin Data Status Register */
```

```
#define AT91C_PIOB_PDSR ((AT91_REG *) 0xFFFFF63C)
```

```
/* Pin Controlled by PB1 */
```

```
#define AT91C_PIO_PB1 ((unsigned int) 1 << 1)
```

Source: u-boot-1.1.1/include/asm-arm/arch-at91rm9200/AT91RM9200.h

```
if (*AT91C_PIOB_PDSR & AT91C_PIO_PB1)
```

Source: u-boot-1.1.1/board/at91rm9200dk/at91rm9200dk.c

Set Bit(s)

```
*ADDRESS |= MASK;
```

Set Bit(s)

```
#define AT91C_PIOC_ASR ((AT91_REG *) 0xFFFFF870) /* (PIOC) Select A Register */

#define AT91C_PIO_PC0 ((unsigned int) 1 << 0) /* Pin Controlled by PC0 */
#define AT91C_PC0_BFCK ((unsigned int) AT91C_PIO_PC0) /* Burst Flash Clock */

#define AT91C_PIO_PC1 ((unsigned int) 1 << 1) /* Pin Controlled by PC1 */

/* Burst Flash Ready */
#define AT91C_PC1_BFRDY_SMOE ((unsigned int) AT91C_PIO_PC1)

#define AT91C_PIO_PC3 ((unsigned int) 1 << 3) /* Pin Controlled by PC3 */

/* Burst Flash Address Advance / SmartMedia Write Enable */
#define AT91C_PC3_BFBAA_SMWE ((unsigned int) AT91C_PIO_PC3)
```

Source: u-boot-1.1.1/include/asm-arm/arch-at91rm9200/AT91RM9200.h

```
*AT91C_PIOC_ASR = AT91C_PC0_BFCK |
                  AT91C_PC1_BFRDY_SMOE |
                  AT91C_PC3_BFBAA_SMWE;
```

Source: u-boot-1.1.1/board/at91rm9200dk/at91rm9200dk.c

Clear Bit(s)

```
*ADDRESS &= ~MASK;
```

Clear Bit(s)

```
immap->im_cpm.cp_rccr &= ~0x00F3FFFF;
```

Source: u-boot-1.1.1/examples/mem_to_mem_idma2intr.c

immap is of type struct immap_t defined in u-boot-1.1.1/include/asm-ppc/immap_8260.h

im_cpm is of type struct comm_proc (typedef to cpm8260_t) in u-boot-1.1.1/include/asm-ppc/immap_8260.h

cp_rccr is of type uint in struct comm_proc in u-boot-1.1.1/include/asm-ppc/immap_8260.h

Toggle Bit(s)

```
*ADDRESS ^= MASK;
```

Toggle Bit(s)

```
#define CFG_REGS_BASE    0xFA000000
```

Source: u-boot-1.1.1/include/configs/RPXsuper.h

```
t_rpx_regs *regs = (t_rpx_regs*) CFG_REGS_BASE;
```

t_rpx_regs is of type struct tt_rpx_regs in u-boot-1.1.1/board/rpxsuper/rpxsuper.h

```
#define MII_MDCK    0x02
```

```
regs->bcsr4 ^= MII_MDCK;
```

Source: u-boot-1.1.1/board/rpxsuper/mii_phy.c

Write-only registers

Read-modify-write to write-only registers:

```
unsigned int temp;  
  
TEMP = *ADDRESS;  
  
TEMP |= MASK;  
  
*ADDRESS = TEMP;
```

Logical operations

Logical AND

&&

Logical OR

||

```
if ((hw->mac_type < e1000_82543) && (hw->report_tx_early == 1))
```

Source: u-boot-1.1.1/drivers/e1000.c

```
while (inw (I2C_STAT) || (inw (I2C_CON) & I2C_CON_MST)) {
```

Source: u-boot-1.1.1/drivers/omap1510_i2c.c

Loops

while ()
do ... while ()
for ()

```
while ((com_port->lsr & LSR_THRE) == 0);
```

Source: u-boot-1.1.1/drivers/ns16550.c

com_port is of type NS16550_t defined in u-boot-1.1.1/include/ns16550.h

```
#define LSR_THRE    0x20    /* Xmit holding register empty */
```

Source: u-boot-1.1.1/include/ns16550.h

Pointers

```
int *p, x;  
p = &x;
```

```
unsigned char *out_port;  
out_port = 0x1010;  
*out_port = 0xaa;
```


Pointers to functions

```
void (*fp) ();  
void foobar ();
```

```
fp = foobar;
```

```
(*fp)();
```

Structures

```
typedef struct _foobar {  
    int x;  
    int y;  
} foobar;  
  
foobar z;
```

Structure Overlays

```
typedef struct _AT91S_TC {
    AT91_REG    TC_CCR;    /* Channel Control Register */
    AT91_REG    TC_CMR;    /* Channel Mode Register */
    AT91_REG    Reserved0[2]; /* */
    AT91_REG    TC_CV;    /* Counter Value */
    AT91_REG    TC_RA;    /* Register A */
    AT91_REG    TC_RB;    /* Register B */
    AT91_REG    TC_RC;    /* Register C */
    AT91_REG    TC_SR;    /* Status Register */
    AT91_REG    TC_IER;    /* Interrupt Enable Register */
    AT91_REG    TC_IDR;    /* Interrupt Disable Register */
    AT91_REG    TC_IMR;    /* Interrupt Mask Register */
} AT91S_TC, *AT91PS_TC;
```

Source: u-boot-1.1.1/include/asm-arm/arch-at91rm9200/AT91RM9200.h

Unions

```
union foo {  
    char a;  
    int b;  
    long c;  
} foobar;
```

```
foobar.c = 0xdeadbeef;
```

```
foobar.b = 0xdead;
```

```
foobar.a = 0xde;
```

Bitfields

Compiler dependant.

```
struct foo {  
    unsigned int bit0: 1;  
    unsigned int bit1: 1;  
    unsigned int bit2: 1;  
    unsigned int bit3: 1;  
    unsigned int bit4: 1;  
    unsigned int bit5: 1;  
    unsigned int bit6: 1;  
    unsigned int bit7: 1;  
} bits;
```

```
#define PORTB (bits *) 0x1010  
PORTB.bit0 = 1;
```

Endianness

```
#include <stdio.h>
int main (void)
{
    unsigned int x = 0x55aa;
    char *p;
    p = (char *) &x;

    printf ("%p = %x\n", p, *p);
    printf ("%p = %x\n", p+1, *(p+1));

    return 0;
}
```

Packed Structure

```
struct usb_endpoint_descriptor {  
    u8 bLength;  
    u8 bDescriptorType;    /* 0x5 */  
    u8 bEndpointAddress;  
    u8 bmAttributes;  
    u16 wMaxPacketSize;  
    u8 bInterval;  
} __attribute__((packed));
```

Source: u-boot-1.1.1/include/usbdescriptors.h