



64KB Data Flash Application Notes

Features

- The 64KB on-chip data flash can be read by direct external addressing mode (by MOVX instruction)
- To read the 64KB on-chip data flash is similar to read external RAM (set OME=0,DFEN=1)
- To program/page erase/protect the 64KB on-chip data flash by used SyncMOS provide ISP function (set FAU0=1 first)

System Control Register (SCONF,\$BF)

bit7				bit0			
WDR	R	R	R	DFEN	ISPE	OME	ALEI
0	0	0	0	0	0	0	0

When used the direct external-addressing mode, the read data area was define by OME and DFEN value

DFEN	OME	Address of MOVX below 768	Address of MOVX over 768
0	0	External memory	External memory
0	1	Internal RAM	External memory
1	0	Internal data flash	Internal data flash
1	1	Internal RAM	Internal data flash

DFEN (64K Data Flash Enable Bit)

- =1: The 64K Data Flash zone enable.
- =0: The 64K Data Flash zone disable.

OME

- =1: The 768 bytes on-chip expanded RAM zone enable.
At direct external addressing mode, when the address range from 0000 to 02FF is access the 768 bytes on-chip expanded RAM, when address >02FF is access to the data flash ROM (DFEN=1) or external memory (DFEN=0).
- =0: The 768 bytes on-chip expanded RAM zone disable
At direct external addressing mode, the address from 0000 to FFFF is access to the data flash ROM (DFEN=1) or external memory (DFEN=0).

ISPE

- =1: enable ISP function
- =0: Disable ISP function

ISP Register (ISPC, \$F7)

bit7						bit0	
START	R	FAU0	R	R	R	ISPF1	ISPF0
0	0	0	0	0	0	0	0

FAU0: FA16 bit

- =0, select 64KB program flash
- =1, select 64KB data flash

ISPF[1..0]: ISP Function Select

- =00: Byte program



July 2003

For SM59264 used

=01:Chip protects

=10:Page erase

=11:Chip erase

Read/write the Data flash Subroutine (Used Keil C)

Example program:

. Main program:

```
#include <reg52.h>
extern void Gotolsp(void);
void read_target();
void read_target1();
xdata char target[256] _at_ 0x0400; //set OSD array from 0x10400 to 0x104ff
xdata char target1[256] _at_ 0x0500; //set OSD array from 0x10500 to 0x105ff
sfr sconf=0xBF;

void DelayX10ms(int count) //Delay time=count*10ms
{
    unsigned int l,m,n;

    for(l=0;l<count;l++)
        for(m=0;m<10;m++)
            for(n=0;n<120;n++)
                ;
}

void read_target(void)
{
    unsigned char i;
    for(i=0;i<20;i++) //Read OSD data from 0x10400 to 0x104ff
    {
        P0 = P0 ^ 0x40; //set p06 flash
        P3 = target[i]; //Output OSD data to Port3
        DelayX10ms(100); //Delay 500 ms
    }
}

void read_target1(void)
{
    unsigned char i;
    for(i=0;i<20;i++) //Read OSD data from 0x10500 to 0x105ff
    {
        P0 = P0 ^ 0x80; //set p07 flash
        P3 = target1[i]; //Output OSD data to Port2
        DelayX10ms(100); //Delay 500 ms
    }
}

void main( void ) //Main program
{
    sconf = 0x08; //enable 64k data flash area(dfen=1)

    if ((P2 & 0x20) == 0) //ISP detect.If P2.5=0 go to FC00H
        Gotolsp(); // enter ISP boot code for assemble code */

    while(1)
    {
        read_target();
        read_target1();
    }
}
```



Font program:

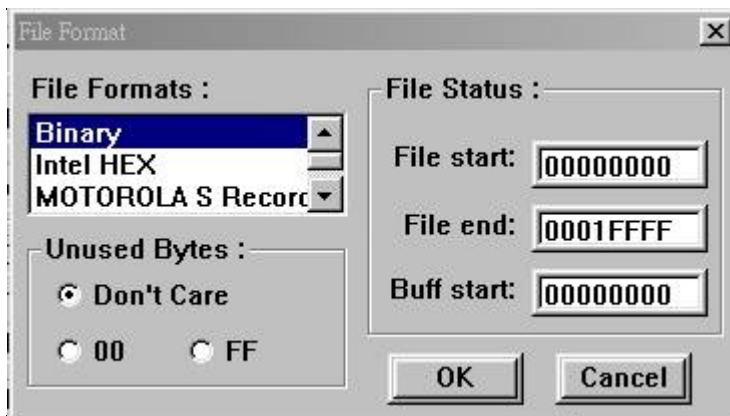
```

#include <stdio.h>
char code dummy[1024]=          /* dummy for expand ram used */
{
    0x00
};
char code target1[] =          //target will put on data flash from address 0x10400
{0x89,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f,
 0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x1a,0x1b,0x1c,0x1d,0x1e,0x1f,
 0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x2a,0x2b,0x2c,0x2d,0x2e,0x2f,
 0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3a,0x3b,0x3c,0x3d,0x3e,0x3f,
 0x40,0x41,0x42,0x43,0x44,0x45,0x46,0x47,0x48,0x49,0x4a,0x4b,0x4c,0x4d,0x4e,0x4f,
 0x50,0x51,0x52,0x53,0x54,0x55,0x56,0x57,0x58,0x59,0x5a,0x5b,0x5c,0x5d,0x5e,0x5f,
 0x60,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69,0x6a,0x6b,0x6c,0x6d,0x6e,0x6f,
 0x70,0x71,0x72,0x73,0x74,0x75,0x76,0x77,0x78,0x79,0x7a,0x7b,0x7c,0x7d,0x7e,0x7f,
 0x80,0x81,0x82,0x83,0x84,0x85,0x86,0x87,0x88,0x89,0x8a,0x8b,0x8c,0x8d,0x8e,0x8f,
 0x90,0x91,0x92,0x93,0x94,0x95,0x96,0x97,0x98,0x99,0x9a,0x9b,0x9c,0x9d,0x9e,0x9f,
 0xa0,0xa1,0xa2,0xa3,0xa4,0xa5,0xa6,0xa7,0xa8,0xa9,0xaa,0xab,0xac,0xad,0xae,0xaf,
 0xb0,0xb1,0xb2,0xb3,0xb4,0xb5,0xb6,0xb7,0xb8,0xb9,0xba,0xbb,0xbc,0xbd,0xbe,0xbf,
 0xc0,0xc1,0xc2,0xc3,0xc4,0xc5,0xc6,0xc7,0xc8,0xc9,0xca,0xcb,0xcc,0xcd,0xce,0xcf,
 0xd0,0xd1,0xd2,0xd3,0xd4,0xd5,0xd6,0xd7,0xd8,0xd9,0xda,0xdb,0xdc,0xdd,0xde,0xdf,
 0xe0,0xe1,0xe2,0xe3,0xe4,0xe5,0xe6,0xe7,0xe8,0xe9,0xea,0xeb,0xec,0xed,0xee,0xef,
 0xf0,0xf1,0xf2,0xf3,0xf4,0xf5,0xf6,0xf7,0xf8,0xf9,0xfa,0xfb,0xfc,0xfd,0xfe,0xff,
};
int code target[] =          //target will put on data flash from address 0x10500
{0x1234,0x5678,0x9012,0x3456,0x7890,0x0500,0x0600,0x0700,0x0800,0x0900,0x0a00,0x0b00,0x0c00,0x0d00,0x0e00,0x0f00,
 0x0001,0x0101,0x0201,0x0301,0x0401,0x0501,0x0601,0x0701,0x0801,0x0901,0x0a01,0x0b01,0x0c01,0x0d01,0x0e01,0x0f01,
 0x0002,0x0102,0x0202,0x0302,0x0402,0x0502,0x0602,0x0702,0x0802,0x0902,0x0a02,0x0b02,0x0c02,0x0d02,0x0e02,0x0f02,
 0x0003,0x0103,0x0203,0x0303,0x0403,0x0503,0x0603,0x0703,0x0803,0x0903,0x0a03,0x0b03,0x0c03,0x0d03,0x0e03,0x0f03,
 0x0004,0x0104,0x0204,0x0304,0x0404,0x0504,0x0604,0x0704,0x0804,0x0904,0x0a04,0x0b04,0x0c04,0x0d04,0x0e04,0x0f04,
 0x0005,0x0105,0x0205,0x0305,0x0405,0x0505,0x0605,0x0705,0x0805,0x0905,0x0a05,0x0b05,0x0c05,0x0d05,0x0e05,0x0f05,
 0x0006,0x0106,0x0206,0x0306,0x0406,0x0506,0x0606,0x0706,0x0806,0x0906,0x0a06,0x0b06,0x0c06,0x0d06,0x0e06,0x0f06,
 0x0007,0x0107,0x0207,0x0307,0x0407,0x0507,0x0607,0x0707,0x0807,0x0907,0x0a07,0x0b07,0x0c07,0x0d07,0x0e07,0x0f07,
 0x0008,0x0108,0x0208,0x0308,0x0408,0x0508,0x0608,0x0708,0x0808,0x0908,0x0a08,0x0b08,0x0c08,0x0d08,0x0e08,0x0f08,
 0x0009,0x0109,0x0209,0x0309,0x0409,0x0509,0x0609,0x0709,0x0809,0x0909,0x0a09,0x0b09,0x0c09,0x0d09,0x0e09,0x0f09,
 0x000a,0x010a,0x020a,0x030a,0x040a,0x050a,0x060a,0x070a,0x080a,0x090a,0x0a0a,0x0b0a,0x0c0a,0x0d0a,0x0e0a,0x0f0a,
 0x000b,0x010b,0x020b,0x030b,0x040b,0x050b,0x060b,0x070b,0x080b,0x090b,0x0a0b,0x0b0b,0x0c0b,0x0d0b,0x0e0b,0x0f0b,
 0x000c,0x010c,0x020c,0x030c,0x040c,0x050c,0x060c,0x070c,0x080c,0x090c,0x0a0c,0x0b0c,0x0c0c,0x0d0c,0x0e0c,0x0f0c,
 0x000d,0x010d,0x020d,0x030d,0x040d,0x050d,0x060d,0x070d,0x080d,0x090d,0x0a0d,0x0b0d,0x0c0d,0x0d0d,0x0e0d,0x0f0d,
 0x000e,0x010e,0x020e,0x030e,0x040e,0x050e,0x060e,0x070e,0x080e,0x090e,0x0a0e,0x0b0e,0x0c0e,0x0d0e,0x0e0e,0x0f0e,
 0x000f,0x010f,0x020f,0x030f,0x040f,0x050f,0x060f,0x070f,0x080f,0x090f,0x0a0f,0x0b0f,0x0c0f,0x0d0f,0x0e0f,0x0f0f,
};

```

Load file rule before program (used HiLo All-11)

1. Load main program to Code Flash area





2. Load font Program to Data Flash area

