制作 keil 的 flash 下载算法

By Wendell.li

最近在使用 NXP 的 RT1052 进行一些工作的开发，这个 MCU 是没有 Flash 的，需要用户自己选定 flash，开始使用 NXP 退出的官方 demo 板进行调试，使用官方提供的历程下载到 flash 中，但是查了一下板载 flash 的价格果断放弃，大家可以查查价格，结合自己的情况选则了华邦的 W25Q64JV 这颗 norflash。然后悲催的事情就开始了，不知道如何通过 keil 下载进去，经过摸索终于实现了这个功能。下面记录一下具体的操作步骤。很少这样写东西也得不好大家多包涵;

一、 实现 flash 的基本操做

实现 flash 的基本功能，包含获取 ID，擦除整片 flash、读操作、写操作、擦除扇区操作等，此部分的内容可以参考正点原子或者野火的 STM32 的历程中有相关的说明，下面按照步骤写一下实现过程。

1. SPI 以及 GPIO 的初始化，以及发送函数

```
void FingerTech_Spigpioinit(void)
{
    gpio_pin_config_t  fingertech_led = {kGPIO_DigitalOutput,0,kGPIO_NoIntmode};
    IOMUXC_SetPinMux(IOMUXC_GPIO_SD_B1_06_LPSPI2_PCS0,0U);
    IOMUXC_SetPinMux(IOMUXC_GPIO_SD_B1_07_LPSPI2_SCK,0U);

    IOMUXC_SetPinMux(IOMUXC_GPIO_SD_B1_08_LPSPI2_SD0,0U);

    IOMUXC_SetPinMux(IOMUXC_GPIO_SD_B1_09_LPSPI2_SDI,0U);

    IOMUXC_SetPinConfig(IOMUXC_GPIO_SD_B1_06_LPSPI2_PCS0,0x10B0u);

    IOMUXC_SetPinConfig( IOMUXC_GPIO_SD_B1_07_LPSPI2_SCK,0x10B0u);

    IOMUXC_SetPinConfig( IOMUXC_GPIO_SD_B1_08_LPSPI2_SD0,0x10B0u);

    IOMUXC_SetPinConfig( IOMUXC_GPIO_SD_B1_09_LPSPI2_SDI,0x10B0u);
```

```
13.
14.     IOMUXC_SetPinMux( IOMUXC_GPIO_SD_B1_10_GPIO3_IO10,0U);

15.     IOMUXC_SetPinMux( IOMUXC_GPIO_SD_B1_11_GPIO3_IO11, 0U);
16.     IOMUXC_SetPinConfig( IOMUXC_GPIO_SD_B1_10_GPIO3_IO10,0x10B0u);

17.
18.     IOMUXC_SetPinConfig(IOMUXC_GPIO_SD_B1_11_GPIO3_IO11,0x10B0u);

19.     GPIO_PinInit(GPIO3,10u,&fingertech_led);
20.     GPIO_PinInit(GPIO3,11u,&fingertech_led);
21.     GPIO_PinWrite(GPIO3,10u,1);
22.     GPIO_PinWrite(GPIO3,11u,1);
23. }
24. void FingerTech_SpiConfigInit(void)
25. {
26.     lpspi_master_config_t masterConfig;
27.     uint32_t srcClock_Hz;
28.     /*Set clock source for LPSPI*/
29.     CLOCK_SetMux(kCLOCK_LpspiMux, EXAMPLE_LPSPI_CLOCK_SOURCE_SELECT);
30.     CLOCK_SetDiv(kCLOCK_LpspiDiv, EXAMPLE_LPSPI_CLOCK_SOURCE_DIVIDER);
31.
32.     masterConfig.baudRate = (18000000U);
33.     masterConfig.bitsPerFrame = 8;
34.     masterConfig.cpol = kLPSPI_ClockPolarityActiveHigh;
35.     masterConfig.cpha = kLPSPI_ClockPhaseFirstEdge;
36.     masterConfig.direction = kLPSPI_MsbFirst;
37.
38.     masterConfig.pcsToSckDelayInNanoSec = 1000000000 / masterConfig.baudRate;
39.     masterConfig.lastSckToPcsDelayInNanoSec = 1000000000 / masterConfig.baudRate;
40.     masterConfig.betweenTransferDelayInNanoSec = 1000000000 / masterConfig.baudRate;
41.
42.     masterConfig.whichPcs = kLPSPI_Pcs0;            //Ê¹ÓÃÆ¬Ñ¡0
43.     masterConfig.pcsActiveHighOrLow = kLPSPI_PcsActiveLow;//0±íÊ¾Ñ¡ÖÐ
44.
45.     masterConfig.pinCfg = kLPSPI_SdiInSdoOut;
46.     masterConfig.dataOutConfig = kLpspiDataOutTristate;//È¡ÏûÑ¡ÖÐºó±£³Ö¸ß×èÌ¬
47.
48.     srcClock_Hz = LPSPI_MASTER_CLK_FREQ;
49.     LPSPI_MasterInit(LPSPI2, &masterConfig, srcClock_Hz);
50.     LPSPI_Enable(LPSPI2,true);
51. }
52.
53. //spi 读写函数
```

```
54. void FingerTech_SpiWRDate(uint8_t* wdata,uint8_t* rdata,size_t size)
55. {
56.     lpspi_transfer_t fingertechmasterXfer;
57.     fingertechmasterXfer.txData = wdata;
58.     fingertechmasterXfer.rxData = rdata;
59.     fingertechmasterXfer.dataSize = size;
60.     fingertechmasterXfer.configFlags = kLPSPI_MasterPcs0 | kLPSPI_MasterPcsContinuous | kLP
    SPI_SlaveByteSwap;
61.     LPSPI_MasterTransferBlocking(LPSPI2, &fingertechmasterXfer);
62. }
63.
```

2. flash 相关操作函数

```
1.  //写允许函数
2.  void FingerTech_W25Q64JV_EnableWrite(void)
3.  {
4.      uint8_t WriteEnableCmd = W25Q64_WRITEENABLE_CMD;
5.      uint8_t ReadEnableCmd = 0x00;
6.      FingerTech_SpiWRDate(&WriteEnableCmd,&ReadEnableCmd,1);
7.  }
8.
9.  //写禁止函数
10. void FingerTech_W25Q64JV_DisableWrite(void)
11. {
12.     uint8_t WriteEnableCmd = W25Q64_WRITEDISABLE_CMD;
13.     uint8_t ReadEnableCmd = 0x00;
14.     FingerTech_SpiWRDate(&WriteEnableCmd,&ReadEnableCmd,1);
15. }
16. //读 ID
17. uint32_t FingerTech_ReadW25Q64JVID(void)
18. {
19.     uint32_t flashid=0;
20.     uint8_t readidcmd[6]={0x90,0x00,0x00,0x00,0xff,0xff};
21.     uint8_t readidrxbuf[6];
22.     FingerTech_SpiWRDate(readidcmd,readidrxbuf,6);
23.     flashid |= ((readidrxbuf[4]<<8)|(readidrxbuf[5]));
24.     return flashid;
25. }
26.
27. //等待忙
28. void FingerTech_W25Q64JV_WAITBUSY(void)
29. {
30.     uint8_t isbusy = 0;
```

```
31.     do{
32.         isbusy = FingerTech_W25Q64JV_ReadSTAREG1();
33.     }while(isbusy);
34. }
35. //读状态寄存器
36. uint8_t FingerTech_W25Q64JV_ReadSTAREG1(void)
37. {
38.     uint8_t WriteEnableCmd[2] = {W25Q64_READSTAREG1_CMD,0xff};
39.     uint8_t ReadEnableCmd[2] = {0x00,0x00};
40.     FingerTech_SpiWRDate(WriteEnableCmd,ReadEnableCmd,2);
41.     return (ReadEnableCmd[1]&0x01);
42. }
43. //写状态寄存器
44. void FingerTech_Write_SR(uint8_t sr)
45. {
46.     uint8_t WriteEnableCmd[2] = {W25Q64_WRITESTAREG1_CMD,0xff};
47.     uint8_t ReadEnableCmd[2] = {0x00,0x00};
48.     WriteEnableCmd[1] = sr;
49.     FingerTech_SpiWRDate(WriteEnableCmd,ReadEnableCmd,2);
50. }
51.
52. //擦除整片 flash
53. void FingerTech_W25Q64JV_Erasechip(void)
54. {
55.     uint8_t WriteEnableCmd = W25Q64_ERASECHIP_CMD;
56.     uint8_t ReadEnableCmd = 0x00;
57.     FingerTech_W25Q64JV_EnableWrite();
58.     FingerTech_W25Q64JV_WAITBUSY();
59.     FingerTech_SpiWRDate(&WriteEnableCmd,&ReadEnableCmd,1);
60.     FingerTech_W25Q64JV_WAITBUSY();
61. }
62.
63. //擦除一个扇区
64. void FingerTech_W25Q64JV_EraseSector(uint32_t desaddress)
65. {
66.     uint8_t txcmd[4]={W25Q64_ERASESECTOR,0x00,0x00,0x00};
67.     uint8_t rxcmd[4]={0x00,0x00,0x00,0x00};
68.     desaddress*=4096;
69.     txcmd[1] = (uint8_t)(desaddress>>16);
70.     txcmd[2] = (uint8_t)(desaddress>>8);
71.     txcmd[3] = (uint8_t)(desaddress);
72.     FingerTech_W25Q64JV_EnableWrite();
73.     FingerTech_W25Q64JV_WAITBUSY();
74.     FingerTech_SpiWRDate(&txcmd[0],&rxcmd[0],4);
```

```
75.     FingerTech_W25Q64JV_WAITBUSY();
76. }
77.  //无校验写入
78. void Fingertech_Flash_Write_NoCheck(uint8_t* pBuffer,uint32_t WriteAddr,uint16_t NumByteToW
    rite)
79. {
80.     uint16_t pageremain;
81.     pageremain=256-WriteAddr%256;
82.     if(NumByteToWrite<=pageremain)
83.         pageremain=NumByteToWrite;
84.     while(1)
85.     {
86.         FingerTech_W25Q64JV_WritePage(pBuffer,WriteAddr,pageremain);
87.         if(NumByteToWrite==pageremain)
88.             break;
89.         else
90.         {
91.             pBuffer+=pageremain;
92.             WriteAddr+=pageremain;
93.
94.             NumByteToWrite-=pageremain;
95.             if(NumByteToWrite>256)
96.                 pageremain=256;
97.             else pageremain=NumByteToWrite;
98.         }
99.     };
100. }
101.
102.   //写 flash
103. void Fingertech_Flash_Write(uint8_t* pBuffer,uint32_t WriteAddr,uint16_t NumByteToWrite)
       {
104.     uint32_t secpos;
105.     uint16_t secoff;
106.     uint16_t secremain;
107.     uint16_t i;
108.
109.     secpos=WriteAddr/4096;
110.     secoff=WriteAddr%4096;
111.     secremain=4096-secoff;
112.
113.     if(NumByteToWrite<=secremain)
114.         secremain=NumByteToWrite;
115.     while(1)
116.     {
```

```
        FingerTech_W25Q64JV_readPage(SPI_FLASH_BUF,secpos*4096,4096);
        for(i=0;i<secremain;i++)
        {
            if(SPI_FLASH_BUF[secoff+i]!=0XFF)
                break;
        }
        if(i<secremain)
        {
            FingerTech_W25Q64JV_EraseSector(secpos);
            for(i=0;i<secremain;i++)
            {
                SPI_FLASH_BUF[i+secoff]=pBuffer[i];
            }
            Fingertech_Flash_Write_NoCheck(SPI_FLASH_BUF,secpos*4096,4096);
        }
        else
            Fingertech_Flash_Write_NoCheck(pBuffer,WriteAddr,secremain);
        if(NumByteToWrite==secremain)
            break;
        else
        {
            secpos++;
            secoff=0;
            pBuffer+=secremain;
            WriteAddr+=secremain;
            NumByteToWrite-=secremain;
            if(NumByteToWrite>4096)secremain=4096;
            else secremain=NumByteToWrite;
        }
    };
}
//写一页数据
void FingerTech_W25Q64JV_WritePage(uint8_t* pbuf,uint32_t desaddress,uint32_t len)
{
    uint8_t txcmd[4+W25X_PAGE_SIZE]={W25Q64_ERASESECTORPAGEPROGRAMCMD,0x00,0x00,0x00};
    uint8_t rxcmd[4+W25X_PAGE_SIZE]={0x00,0x00,0x00,0x00};
    uint32_t writeaddr = 0;
    FingerTech_W25Q64JV_WAITBUSY();
    writeaddr = desaddress ;
    txcmd[1] = (uint8_t)(writeaddr>>16);
    txcmd[2] = (uint8_t)(writeaddr>>8);
    txcmd[3] = (uint8_t)(writeaddr);
    memcpy(&txcmd[4],pbuf,len);
    FingerTech_W25Q64JV_EnableWrite();
```

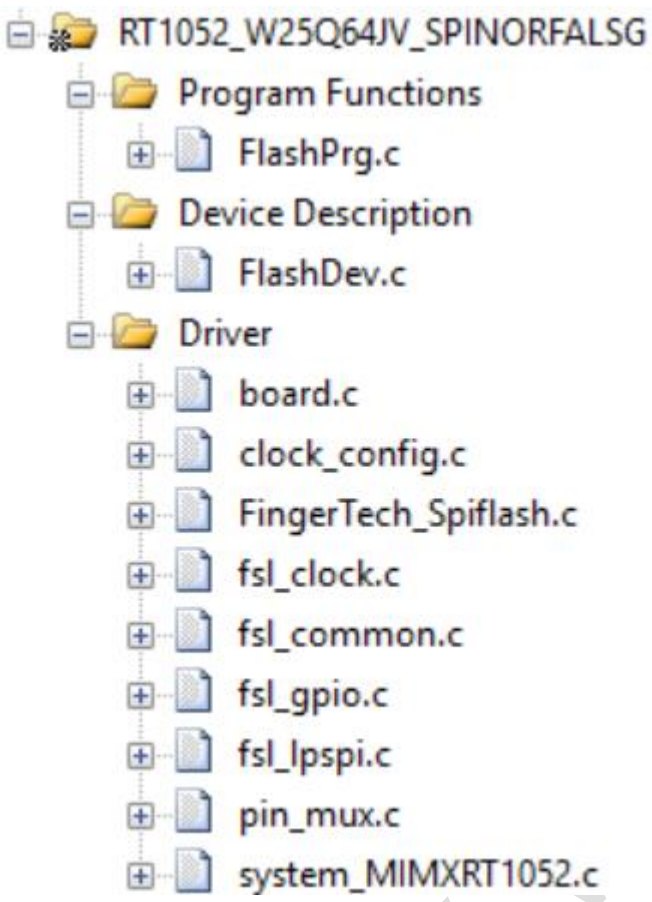
```
161.     FingerTech_SpiWRDate(&txcmd[0],&rxcmd[0],4+W25X_PAGE_SIZE);
162.     FingerTech_W25Q64JV_WAITBUSY();
163. }
164. 
165. //读一页数据，最大读一个扇区
166. void FingerTech_W25Q64JV_readPage(uint8_t* pbuf,uint32_t desaddress,uint32_t len)
167. {
168.     uint8_t txcmd[4100]={W25Q64_READPAGECMD,0x00,0x00,0x00};
169.     uint8_t rxcmd[4100]={0x00,0x00,0x00,0x00};
170.     uint32_t readddr = 0;
171. 
172.     readddr = desaddress ;
173.     txcmd[1] = (uint8_t)(readddr>>16);
174.     txcmd[2] = (uint8_t)(readddr>>8);
175.     txcmd[3] = (uint8_t)(readddr);
176.     memset(&rxcmd[0],0x00,4100);
177.     FingerTech_SpiWRDate(&txcmd[0],&rxcmd[0],4+len);
178.     memcpy(pbuf,&rxcmd[4],len);
179.     //FingerTech_W25Q64JV_WAITBUSY();
180. }
```

二、 对接 keil 下载接口

与 keil 的对接工作主要是对接擦除整片 flash 以及擦除扇区等接口，具体的方法如下：

1. 拷贝 keil 安装目录 falsh 文件夹下的_Template 工程，我的路径为 C:\Keil_v5\ARM\Flash。
2. 编译会发现提示缺少 FlashOS.h 文件，在刚才的那个路径下拷贝到_Template 目录下，并修改去掉引用 FlashOS.h 的..，编译通过后生产一个 NewDevice.flm 文件；
3. 将操作 flash 的函数以及对应的依赖文件都拷贝到这个目录下面来，并添加进工程，我的目录结构如下：

make by vv



4. 填充 FlashDev.c 中 FlashDevice 结构体中的变量,包含设备名、类型、大小、起始地址、写入页大小等内容,如下:

```
struct FlashDevice const FlashDevice  =  {
   FLASH_DRV_VERS,             // Driver Version, do not modify!
   "FingerTech_Flash", // Device Name
   EXTSPI,                     // Device Type
   0x60000000,                 // Device Start Address
   0x00800000,                 // Device Size (8MB)
   256,                        // Programming Page Size
   0,                          // Reserved, must be 0
   0xff,                       // Initial Content of Erased Memory
   1000,                        // Program Page Timeout 300 mSec
   10000,                       // Erase Sector Timeout 5000 mSec
// Specify Size and Address of Sectors
   0x001000, 0x000000,         // Sector Size 4kB (4096 Sectors)
   SECTOR_END
};
```

5. 填充 FlashPrg.c 中的函数接口，我的校验部分写的不好大家可以自己实现。

```c
#include "FlashOS.H"        // FlashOS Structures
#include "pin_mux.h"
#include "fsl_gpio.h"
#include "FingerTech_Spiflash.h"
#include "fsl_lpspi.h"
#include "board.h"

uint32_t base_adr;        // Base Address

/*
 *  Initialize Flash Programming Functions
 *    Parameter:      adr:  Device Base Address
 *                    clk:  Clock Frequency (Hz)
 *                    fnc:  Function Code (1 - Erase, 2 - Program, 3 - Verify)
 *    Return Value:   0 - OK,  1 - Failed
 */

int Init (unsigned long adr, unsigned long clk, unsigned long fnc) {
    /* Add your Code */
    BOARD_ConfigMPU();
    BOARD_InitPins();
    BOARD_BootClockRUN();
    FingerTech_Spigpioinit();
    FingerTech_SpiConfigInit();
    FingerTech_LedInit();
    base_adr = adr;
    return 0;
}

/*
 *  De-Initialize Flash Programming Functions
 *    Parameter:      fnc:  Function Code (1 - Erase, 2 - Program, 3 - Verify)
 *    Return Value:   0 - OK,  1 - Failed
 */
int UnInit (unsigned long fnc) {

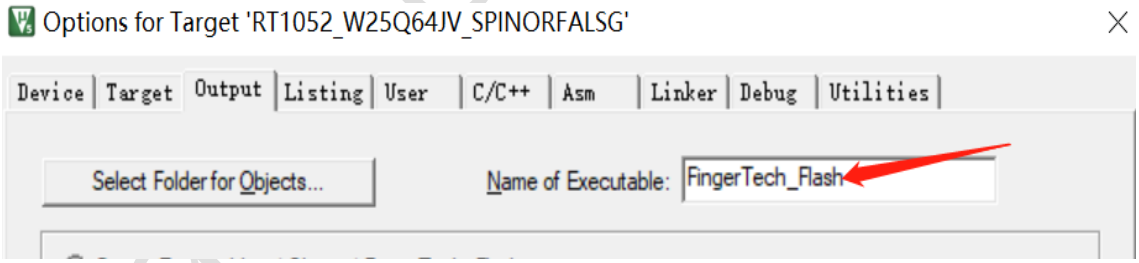
  return (0);
}
/*
 *  Erase complete Flash Memory
 *    Return Value:   0 - OK,  1 - Failed
 */
int EraseChip (void) {

```

```
45.   FingerTech_W25Q64JV_Erasechip();          //²Á³ýflash
46.   return (0);                                          /* Finished without Errors */
47. }
48. /*
49.  *  Erase Sector in Flash Memory
50.  *    Parameter:      adr:  Sector Address
51.  *    Return Value:   0 - OK,  1 - Failed
52.  */
53. int EraseSector (unsigned long adr) {
54.
55.   FingerTech_W25Q64JV_EraseSector(adr - base_adr);
56.   return (1);                                          /* Finished without Errors */
57. }
58. /*
59.  *  Blank Check Checks if Memory is Blank
60.  *    Parameter:      adr:  Block Start Address
61.  *                    sz:   Block Size (in bytes)
62.  *                    pat:  Block Pattern
63.  *    Return Value:   0 - OK,  1 - Failed
64.  */
65.
66. int BlankCheck (unsigned long adr, unsigned long sz, unsigned char pat) {
67.
68.   return (0);                                          /* Always Force Erase */
69. }
70. /*
71.  *  Program Page in Flash Memory
72.  *    Parameter:      adr:  Page Start Address
73.  *                    sz:   Page Size
74.  *                    buf:  Page Data
75.  *    Return Value:   0 - OK,  1 - Failed
76.  */
77.
78. int ProgramPage (unsigned long adr, unsigned long sz, unsigned char *buf) {
79.  // FingerTech_W25Q64JV_WritePage(buf,(adr - base_adr),sz);
80.     Fingertech_Flash_Write(buf,(adr - base_adr),sz);
81.   return (0);                                          /* Finished without Errors */
82. }
83. unsigned long Verify (unsigned long adr,   unsigned long sz,unsigned char *buf)
84. {
85.     uint8_t readbuf[256];
86.     uint32_t len;
87.     uint32_t count = 0;
88.     uint32_t readcount = 0;
```

```
    uint32_t readaddrs =0;
    if((sz%256)==0)
    {
        readcount = sz/256;
    }
    else
    {
        readcount = sz/256 + 1;
    }
    readaddrs = (adr - base_adr);
    for(count=0;count<readcount;count++)
        {
            FingerTech_W25Q64JV_readPage(readbuf,(readaddrs+count*256),256);
            for(len=0;len<256;len++)
            {
                if(buf[len+count*256] != readbuf[len])
                {
                  return count*256 + adr + len;
                }
            }
        }
        return adr+sz;
    }
```

6. 修改一下工程输出的名称，编译后可以看到对应的 flm 文件



7. 将生成的 flm 文件拷贝到 C:\Keil_v5\ARM\Flash\目录下，就能在 keil 中看到并使用，如下是使用效果；

