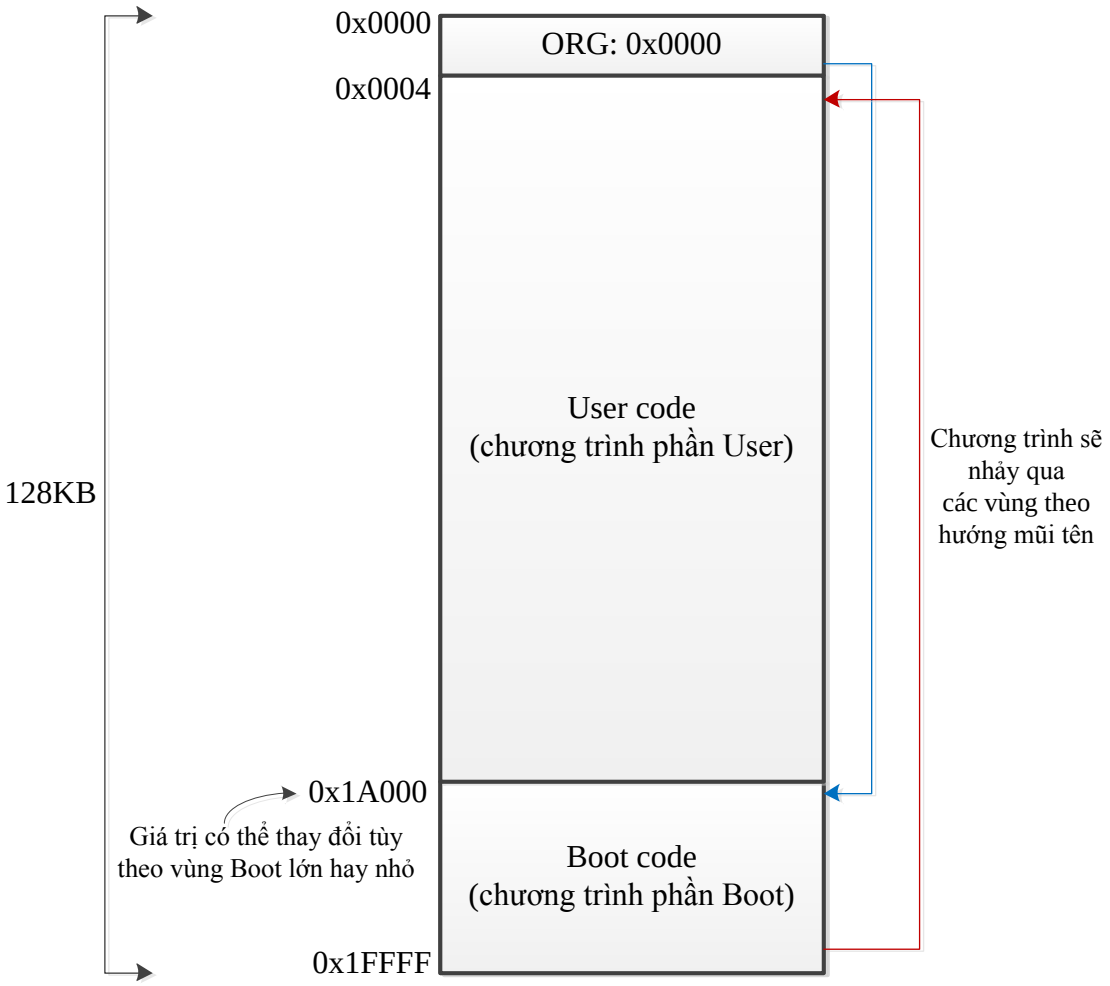
Trang chủ: www.icdrec.edu.vn
E-mail: info@icdrec.edu.vn
Điện thoại: (84-8).37242171 - (84-8).37242172
Email kinh doanh: sg8v1.sales@icdrec.edu.vn
Email hỗ trợ: sg8v1.support@icdrec.edu.vn

SG8V1 Bootloader Application Note

SG8V1_Bootloader_AN1402

Ngày cập nhật	Thông tin cập nhật	Phiên bản
13/10/2014	Mô tả chính	Rev 1.0

Để ứng dụng phần boot loader của SG8V1 người sử dụng phải hiểu được cách phân chia bộ nhớ chương trình thành các phần như hình vẽ bên dưới:



Để có thể thực hiện được chương trình như sơ đồ ở trên, người sử dụng phải tạo 2 project, một project chứa chương trình phần Boot và một project chứa chương trình phần User.

1. Phần Boot

Trong phần Boot code này, người sử dụng sẽ thực hiện các bước sau:

1.1. Tạo project phần Boot

Người sử dụng tạo một project để viết chương trình cho phần Boot

1.2. Tạo tập tin Linker cho phần Boot

Người sử dụng tạo tập tin Linker có tên có tên **BootLinker.x** trong thư mục project và biên soạn nội dung như đoạn mã bên dưới:

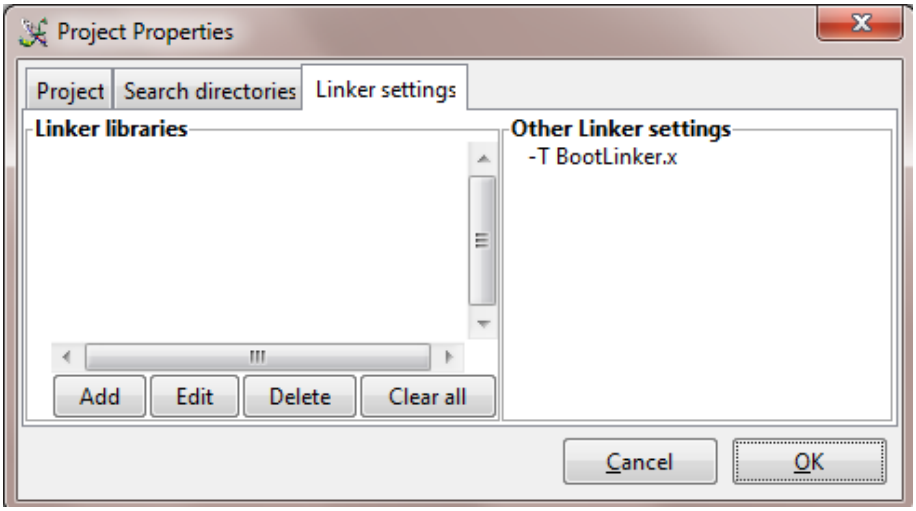
```
OUTPUT_FORMAT("elf32-sg8","elf32-sg8","elf32-sg8")
OUTPUT_ARCH(sg8:1)
__stack_size__ = DEFINED(__stack_size__) ? __stack_size__ : 128;
MEMORY
{
  text   (rx)   : ORIGIN = 0x0, LENGTH = 0x20000
  data   (rw!x) : ORIGIN = 0x800061, LENGTH = 0x3F9F
}
SECTIONS
{
    __startup 0x0000:
    {
       *(__startup)
       KEEP (*(__startup))
       *(__startup.*)
    } > text
    .rel.text      :
    {
       *(.rel.text)
       *(.rel.text.*)
    }
    .rela.text     :
    {
       *(.rela.text)
       *(.rela.text.*)
    }
    .rel.bss       : { *(.rel.bss)  }
    .rela.bss      : { *(.rela.bss) }

    .text  0x1A000: /*Dia chi bat dau vung Boot*/
    {
       *(.init)
       KEEP (*(.init))
       *(__program_main)
       KEEP (*(__program_main))
       *(__program_main.*)
```

```
    *(.text)
    . = ALIGN(2);
    *(.text.*)
    . = ALIGN(2);
     _etext = . ;
  }  > text
  .rodata   :
    {
          . = ALIGN(2);
          *(.rodata)
          . = ALIGN(2);
          *(.rodata.*)
  }  > text

  .register  :
  {
    *(.register)
    KEEP (*(.register))
  } > data
  .data : AT (ADDR (.rodata) + SIZEOF(.rodata))
  {
     PROVIDE (__data_start = .) ;
    *(.data)
    *(.data*)
     PROVIDE (__data_end = .) ;
  }  > data

  .bss   : AT (ADDR (.bss) )
  {
     PROVIDE (__bss_start = .) ;
    *(.bss)
    *(.bss*)
     PROVIDE (__bss_end = .) ;
  }  > data

  .stack  :
  {
        PROVIDE (__stack_start = .) ;
       . = __stack_size__;
        PROVIDE (__stack_end = .) ;
  }  > data
  __stack_end = 0x800061 + 0x3F9F -1;
  __data_load_start = LOADADDR(.data)>>1;
  __data_load_end = (__data_load_start + SIZEOF(.data))>>1;
}
```

Chú ý: Phần địa chỉ có tô màu: 0x1A000 là địa chỉ bắt đầu vùng Boot. Vì đây là một ví dụ, nên trong các ứng dụng cụ thể người sử dụng có thể thay đổi giá trị địa chỉ này một cách hợp lý.

Sau khi tạo xong tập tin Linker, để project liên kết được với tập tin **BootLinker.x** người sử dụng phải thực hiện các thiết lập như sau:

Vào **Project → Properties** màn hình hiển thị cửa sổ **Project Properties** như hình bên dưới:



Chọn tab **Linker Settings**, ta biên soạn vào ô **Other Linker Settings** với cấu trúc như sau:

-T BootLinker.x

Sau đó nhấn OK.

1.3. Viết chương trình cho phần Boot

Trong chương trình phần Boot, người sử dụng có thể tạo 2 chế độ như sau:

+ Chế độ 1: Load (nạp) chương trình cho phần User, các hàm được hỗ trợ trong thư viện **flash.h**

+ Chế độ 2: Nhảy đến vùng User khi đã có sẵn chương trình trong phần User

Để biết vào chế độ nào thì chương trình của người sử dụng viết là kiểm tra cờ Boot, cờ Boot này chúng ta có thể lưu vào EEPROM hoặc bộ nhớ Flash của SG8V1 (hỗ trợ 3KB). Nếu cờ Boot được Set, thì chương trình thực hiện chế độ 1. Nếu cờ Boot không Set thì chương trình thực hiện chế độ 2.

Ví dụ: Thực hiện đoạn mã Boot:

```
void main(void)
{
    /*
    Kiểm tra cờ Boot
    */
    BOOT_BUF = read_ee(BOOT_ADD);
    if(BOOT_BUF ==1) /* Chế độ 1*/
    {
        /*
        Nhận đoạn mã để Load vào bộ nhớ chương trình
        Ví dụ: Nhận từ SD card
        */
        if(FATInit() ==1)
        {
            /*
            Nạp chương trình vào bộ nhớ chương trình
            */
            LoadCodeFromSD();
            fputs("Ket thuc program\r\n",STREAM_UART_3);

            /*
            Xóa cờ Boot
            */
            BOOT_BUF = 0;
            write_ee(BOOT_ADD, BOOT_BUF);
        }
        else
        {
            fputs("SD card Error\r\n",STREAM_UART_3);
        }
        __delay_ms(500);
        //Sg8v1Reset();
        __asm
        RST
        __endasm;
    }
    fputs("No load code from SD card\r\n",STREAM_UART_3);
    i2c_close();    //Close I2C
    spi_close();    //Close SPI
    uart3_close();  //Close UART
    __delay_ms(1000);

    /*
    Chế độ 2: Nhảy đến địa chỉ ứng dụng
    */
    __asm
    LJMP 0x04
    __endasm;
}
```

Chú ý: Lệnh nhảy phải đặt trực tiếp trong hàm main để tránh trường hợp tràn Stack, ta thực hiện lệnh nhảy như sau:

```
__asm
LJMP 0x04
__endasm;
```

2. Phần User

Trong phần User người sử dụng thực hiện các bước sau:

2.1. Tạo project ứng dụng

Người sử dụng tạo một project để viết chương trình cho phần ứng dụng

2.2. Tạo tập tin Linker cho phần User

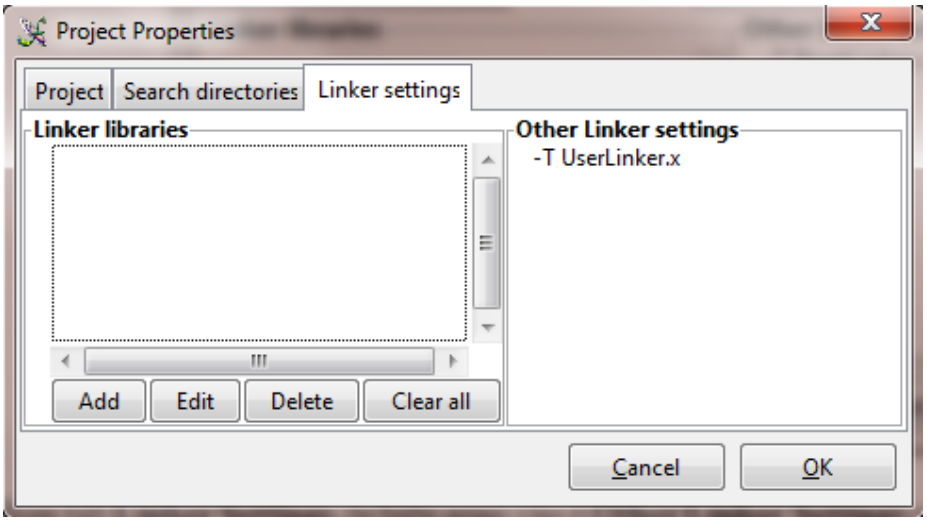
Người sử dụng tạo tập tin Linker có tên có tên **UserLinker.x** trong thư mục project và biên soạn nội dung như đoạn mã bên dưới:

```
OUTPUT_FORMAT("elf32-sg8","elf32-sg8","elf32-sg8")
OUTPUT_ARCH(sg8)
__stack_size__ = DEFINED(__stack_size__) ? __stack_size__ : 128;
MEMORY
{
  text   (rx)   : ORIGIN = 0x08, LENGTH = 0x19FF8
  data   (rw!x) : ORIGIN = 0x800061, LENGTH = 0x3F9F
}
SECTIONS
{
    __startup 0x08:
    {
       *(__startup)
       KEEP (*(__startup))
       *(__startup.*)
    } > text
     __vector_1   0x10:
    {
       *(__vector_1)
       KEEP (*(__vector_1))
       *(__vector_1.*)
    } > text
     __vector_2   0x20:
    {
       *(__vector_2)
       KEEP (*(__vector_2))
       *(__vector_2.*)
    } > text
    /* Internal text space or external memory.  */
    .text   :
    {
       *(__program_startup)
       *(__program_startup.*)
       *(.init)
       KEEP (*(.init))
       *(.init.*)
       *(__program_main)
```

```
    KEEP (*(__program_main))
    *(__program_main.*)
    . = ALIGN(2);
    *(.text)
    . = ALIGN(2);
    *(.text.*)
    . = ALIGN(2);
     _etext = . ;
  }  > text
  .rodata   :
    {
          . = ALIGN(2);
          *(.rodata)
          . = ALIGN(2);
          *(.rodata.*)
  }  > text
  .register :
  {
    *(.register)
    KEEP (*(.register))
  } > data
  .data : AT (ADDR (.rodata) + SIZEOF(.rodata))
  {
     PROVIDE (__data_start = .) ;
    *(.data)
    *(.data*)
     PROVIDE (__data_end = .) ;
  }  > data

  .bss :
  {
     PROVIDE (__bss_start = .) ;
    *(.bss)
    *(.bss*)
     PROVIDE (__bss_end = .) ;
  }  > data

  .stack :
  {
        PROVIDE (__stack_start = .) ;
       . = __stack_size__;
  }  > data
  __stack_end = 0x800061 + 0x3F9F -1;
  __data_load_start = LOADADDR(.data);
  __data_load_end = (__data_load_start + SIZEOF(.data));
}
```

Sau khi tạo xong tập tin Linker, để project liên kết được với tập tin **UserLinker.x** người sử dụng phải thực hiện các thiết lập như sau:

Vào **Project → Properties** màn hình hiển thị cửa sổ **Project Properties** như hình bên dưới:



Chọn tab **Linker Settings**, ta biên soạn vào ô **Other Linker Settings** với cấu trúc như sau:

-T UserLinker.x

Sau đó nhấn OK.

2.3. Viết chương trình ứng dụng cho phần User

Tùy vào ứng dụng mà người sử dụng thực hiện theo yêu cầu của mình. Thực hiện giống như một project bình thường không có chương trình boot, nhưng chỉ thêm phần 2.2