



苏州大学

SOOCHOW UNIVERSITY



第4章 GPIO及程序框架

4.6

第一个汇编语言工程：控制小灯闪烁



苏州大学

SOOCHOW UNIVERSITY

arm



4.6

第一个汇编语言工程： 控制小灯闪烁



4.6

第一个汇编语言工程：控制小灯闪烁

汇编特点

汇编语言编程给人的第一种感觉就是难，但掌握基本的汇编语言编程方法对嵌入式学习十分有益。

本节目的

期望通过一个规范的样例，使学习者在很短的时间内理解一个汇编工程。



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编工程的文件组织

▼ MSP432_Light(asm) [Active - Debug]	工程名
> Binaries	编译链接生成的二进制代码文件
> Includes	系统包含文件（自动生成）
> 01_Doc	<文档文件夹>
> 02_CPU	<内核相关文件>
▼ 03_MCU	<MCU相关文件夹>
▼ Linker_File	<链接文件夹>
> msp432p401r.lds	链接文件
▼ MCU_drivers	<芯片底层驱动构件文件夹>
▼ gpio	<GPIO底层构件文件夹>
> gpio.inc	GPIO底层构件头文件
> gpio.S	GPIO底层构件源文件
> startup	初始化以及启动相关文件夹
▼ 04_UserBoard	<用户板构件文件夹>
▼ light	<小灯构件文件夹>
> light.inc	小灯构件头文件
> light.S	小灯构件源文件
▼ 05_SoftComponent	<软件构件文件夹>
common	<通用代码文件夹>
▼ 06_NosPrg	<无操作系统工程主程序文件夹>
> include.S	总头文件
> main.S	主函数
> Debug	<工程输出文件夹>（编译链接自动生成）

（1）MCU相关头文件夹

（2）底层驱动构件文件夹

（3）Debug工程输出文件夹

（4）程序文件夹



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编工程的文件组织

- ✓ 汇编工程仅包含一个汇编主程序文件，该文件名固定为main.S。
- ✓ 汇编程序的主体是程序的主干，要尽可能简洁、清晰、明了，**程序中的其余功能，尽量由子程序去完成**，主程序主要完成对子程序的循环调用。



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编工程的文件组织

工程文件中主要包含的内容：

(1) 工程描述：工程名、程序描述、版本、日期等。若调试过程有新的体会，也可在此添加。**目的是为将来自己使用，或为同组开发提供必要的备忘信息。**

(2) 总头文件：声明全局变量和包含主程序文件中需要的头文件、宏定义等。

(3) 主程序：主程序一般包括**初始化与主循环**两大部分。初始化包括堆栈初始化、系统初始化、I/O端口初始化、中断初始化等。主循环是程序的工作循环，根据实际需要安排程序段，但一般不宜过长，建议不要超过100行，具体功能可通过调用子程序来实现，或由中断程序实现。

(4) 内部直接调用子程序：**若有不单独存盘的子程序，建议放在此处。**这样在主程序总循环的最后一个语句就可以看到这些子程序。每个子程序不要超过100行。若有更多的子程序请单独存盘，单独测试。



苏州大学

SOOCHOW UNIVERSITY

arm



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编语言GPIO构件及使用方法

头文件gpio.inc

汇编源程序文件gpio.s

功能与C语言GPIO构件一致



4.6

【第一个汇编语言工程：控制小灯闪烁】GPIO构件的头文件gpio.inc

```
#=====
#文件名称: gpio.inc
#功能概要: MSP432 GPIO底层驱动构件（汇编）头文件
#版权所有: 苏州大学嵌入式中心(sumcu.suda.edu.cn)
#版本更新: 2017-11-10 V1.0;
#=====
#端口号地址偏移量宏定义
.equ PT1,(0<<8)
.equ PT2,(1<<8)
.equ PT3,(2<<8)
.equ PT4,(3<<8)
.equ PT5,(4<<8)
.equ PT6,(5<<8)
.equ PT7,(6<<8)
.equ PT8,(7<<8)
.equ PT9,(8<<8)
.equ PT10,(9<<8)
#GPIO引脚方向宏定义
.equ GPIO_IN,(0)
.equ GPIO_OUTPUT,(1)

#端口基地址宏定义(只给出PT1,PT2的地址, 其他由此计算)
.equ PT1_BASE_PTR,0x40004C00    @PT1的地址
.equ PT2_BASE_PTR,0x40004C01    @PT2的地址
#=====
# 函数名称: gpio_init
# 函数返回: 无
# 参数说明: r0:(端口号|(引脚号)),例:(PT2(0))表示2口0脚,头文件中有宏定义
#           r2:引脚方向 (0=输入, 1=输出,可用引脚方向宏定义)
#           r3:端口引脚初始状态 (0=低电平, 1=高电平)
# 功能概要: 初始化指定端口引脚作为GPIO引脚功能, 并定义为输入或输出, 若是输出,
#           还指定初始状态是低电平或高电平
#=====
其他函数略
```

- 端口号地址偏移量宏定义
- GPIO引脚方向宏定义
- 端口基地址宏定义
- 初始化函数gpio_init



4.6

【第一个汇编语言工程：控制小灯闪烁】GPIO构件的汇编源程序gpio.s

```
#-----  
#文件名称: gpio.s  
#功能概要: MSP432 GPIO底层驱动构件（汇编）程序文件  
#-----  
  
#include "gpio.inc"  
#-----以下为内部函数存放处-----  
#-----  
#函数名称: gpio_port_pin_resolution  
#函数返回: 无  
#参数说明: r0: 端口号|引脚号,例:(PT2|(0))表示2口0脚,头文件中有宏定义  
#功能概要: 将传进参数r0进行解析,得出具体端口号与引脚号（如: (PT2|(0)  
#           解析为PT2与0,并将其分别赋值给r0与r1）。  
#-----  
#-----内部函数结束-----  
  
#-----以下为外部接口函数-----  
#-----  
# 函数名称: gpio_init  
# 函数返回: 无  
# 参数说明: r0:(端口号|引脚号),例:(PT2|(0),头文件中有宏定义  
#           r2:引脚方向（0=输入,1=输出,可用引脚方向宏定义）  
#           r3:端口引脚初始状态（0=低电平,1=高电平）  
# 功能概要: 初始化指定端口引脚作为GPIO引脚功能,并定义为输入或输出。若是输出,  
#           还指定初始状态是低电平或高电平  
#-----  
(其他函数略)
```

➤ 内部函数：

gpio_port_pin_resolution

➤ 外部接口函数：

gpio_init



苏州大学

SOOCHOW UNIVERSITY

arm



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编语言Light构件及使用方法

头文件light.inc

汇编源程序文件light.s

用于控制指示灯的亮或暗



4.6

【第一个汇编语言工程：控制小灯闪烁】Light构件的头文件light.inc

```
#=====
#文件名称: light.inc
#功能概要: 小灯驱动程序文件
#=====

#include "gpio.S"
#指示灯端口及引脚定义
.equ LIGHT_RED,(PT2|(0))    @红色RUN灯使用的端口/引脚
.equ LIGHT_GREEN,(PT2|(1))  @蓝色RUN灯使用的端口/引脚
.equ LIGHT_BLUE,(PT2|(2))   @绿色RUN灯使用的端口/引脚
#灯状态宏定义（灯亮、灯暗对应的物理电平由硬件接法决定）
.equ LIGHT_ON,1             @灯亮
.equ LIGHT_OFF,0            @灯暗
```

- 指示灯端口及引脚定义
- 灯状态宏定义



4.6

【第一个汇编语言工程：控制小灯闪烁】Light构件的汇编源程序light.s

```
#=====
#文件名称: light.s
#功能概要: 小灯驱动程序文件
#=====
#include "light.inc"

#=====
# 函数名称: light_init
# 函数返回: 无
# 参数说明: r0:(端口号)|(引脚号),例:(PT2|(0))表示2口0脚,头文件中有宏定义
#           r3:设定小灯状态。由light.inc中宏定义。
# 功能概要: 指示灯驱动初始化。
#=====

#=====
# 函数名称: light_control
# 函数返回: 无
# 参数说明: r0:(端口号)|(引脚号),例:(PT2|(0))表示2口0脚,头文件中有宏定义
#           r3:设定小灯状态。由light.inc中宏定义。
# 功能概要: 控制指示灯亮暗。
#=====

#=====
# 函数名称: light_change
# 函数返回: 无
# 参数说明: r0:(端口号)|(引脚号),例:(PT2|(0))表示2口0脚,头文件中有宏定义
# 功能概要: 切换指示灯亮暗。
#=====
```

➤ 小灯初始化程序：

light_init

➤ 控制小灯亮暗程序：

light_control

➤ 切换小灯亮暗程序：

light_change



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编语言Light构件的使用方法

如何使用
汇编驱动
构件呢？

01

由芯片的哪个引脚来控制小灯

02

是高电平点亮还是低电平点亮



4.6

【第一个汇编语言工程：控制小灯闪烁】汇编语言Light构件的使用方法

例如，小灯由9引脚控制，高电平点亮，使用步骤如下：

小灯汇编 点亮步骤

(1) 在 **light.inc** 文件中给小灯起名字，并确定与 **MCU** 连接的引脚，进行宏定义：

```
.equ LIGHT_BLUE,(PT2|(2))    @蓝色RUN灯使用的端口/引脚
```

(2) 在 **light.inc** 文件中小灯亮、暗进行宏定义，方便编程：

```
equ LIGHT_ON,0    @灯亮
```

```
equ LIGHT_OFF,1   @灯暗
```

(3) 在 **main** 函数中初始化 **LED** 灯的初始状态：

```
ldr r0,=RUN_LIGHT_BLUE    @r0指明端口和引脚（用=是因为宏常数>=256,且用ldr)
```

```
mov r3,#LIGHT_OFF         @r3指明引脚的初始状态
```

```
bl light_init              @调用小灯初始化函数
```

(4) 在 **main** 函数中点亮小灯：

```
bl light_change            @相等，则调用小灯亮暗转变函数
```



4.6

【第一个汇编语言工程：控制小灯闪烁】汇编语言Light测试工程主程序

```
#=====
#文件名称: main.s
#功能概要: 汇编编程控制小灯闪烁
#版权所有: 苏州大学嵌入式中心(sumcu.suda.edu.cn)
#版本更新: 2017-11-10 V1.0;
#=====
#include "include.S"

#start 主函数定义开始
.section .text.main
.global main          @定义全局变量，在芯片初始化之后调用
.align 2              @指令对齐
.type main function   @定义主函数类
.align 2
#end 主函数定义结束

main:
.equ _NVIC, 123
cpsid i               @关闭总中断
#小灯初始化, r0,r3是light_init的入口参数
ldr r0,=LIGHT_BLUE   @r0指明端口和引脚（用=是因为宏常数>=256,且用ldr）
mov r3,#LIGHT_ON      @r3指明引脚的初始状态
bl light_init          @调用小灯初始化函数
cpsie i               @开总中断
#主循环开始

=====
main_loop1:
ldr r4,=RUN_COUNTER_MAX @取延时值到r4
mov r5,#0               @从零计数
loop:
add r5,#1               @加1计数
cmp r4,r5               @r4值与r5值比较
bne loop                @不相等，则跳转loop
bl light_change          @相等，则调用小灯亮暗转变函数
bne main_loop1          @跳转main_loop1
#主循环结束

=====
.end
```

因为该工程需要调用light构件的接口函数，所以在include.S文件中需要包含light.S。

- 首先调用light_init函数，初始化所需的指示灯。注意初始化时，要让每一盏灯初始状态为“暗”。
- 随后，通过light_control函数控制指示灯亮、暗。通过变量的递增并且设置频率后，就能够在程序运行中，比较明显的看到指示灯对应的小灯闪烁的现象。



苏州大学

SOOCHOW UNIVERSITY

arm



4.6 【第一个汇编语言工程：控制小灯闪烁】汇编工程运行过程

main函数之前的运行

main函数之后的运行

当MSP432芯片内电复位或热复位后，系统程序的运行过程



苏州大学

SOOCHOW UNIVERSITY

arm



4.6 【第一个汇编语言工程：控制小灯闪烁】mian函数之前的运行

mian函数之前的运行过程和4.5节c语言控制小灯闪烁的运行过程一样，所以具体的过程可以参考4.5节的加以体会和理解。



4.6 【第一个汇编语言工程：控制小灯闪烁】main函数之后的运行

首先进入main函数后先对所用到的模块进行初始化，比如小灯端口引脚的初始化，小灯引脚复用设置为GPIO功能，设置引脚方向为输出，设置输出为高电平，这样蓝色小灯就可以被点亮。



当某个中断发生后，MCU将转到中断向量表文件isr.s所指定的中断入口地址处开始运行中断服务程序（ISR，Interrupt Service Routine）



苏州大学

SOOCHOW UNIVERSITY

arm

谢谢!

