零死角玩转STM32

构建库函数雏形

淘宝：firestm32.taobao.com

论坛： www.firebbs.cn

主讲内容

01 GPIO寄存器结构体定义

02 GPIO置位复位函数

03 GPIO初始化结构体和初始化函数

参考资料:《零死角玩转STM32》

“自己写库—构建库函数雏形”章节

GPIO寄存器结构体定义

GPIO寄存器结构体定义

代码清单 9-1 封装寄存器列表

```
1  //寄存器的值常常是芯片外设自动更改的，即使 CPU 没有执行程序，也有可能发生变化
2  //编译器有可能会对没有执行程序的变量进行优化，所以用 volatile 来修饰寄存器变量
3
4  //volatile 表示易变的变量，防止编译器优化，
5  #define      __IO    volatile
6  typedef unsigned int uint32_t;
7  typedef unsigned short uint16_t;
8
9  // GPIO 寄存器结构体定义
10 typedef struct
11 {
12      __IO uint32_t CRL;        // 端口配置低寄存器，      地址偏移 0X00
13      __IO uint32_t CRH;        // 端口配置高寄存器，      地址偏移 0X04
14      __IO uint32_t IDR;        // 端口数据输入寄存器，    地址偏移 0X08
15      __IO uint32_t ODR;        // 端口数据输出寄存器，    地址偏移 0X0C
16      __IO uint32_t BSRR;       // 端口位设置/清除寄存器，地址偏移 0X10
17      __IO uint32_t BRR;        // 端口位清除寄存器，      地址偏移 0X14
18      __IO uint32_t LCKR;       // 端口配置锁定寄存器，    地址偏移 0X18
19 } GPIO_TypeDef;
```

GPIO寄存器结构体定义

```
1  /*片上外设基地址  */
2  #define PERIPH_BASE            ((unsigned int)0x40000000)
3
4  /*APB2 总线基地址 */
5  #define APB2PERIPH_BASE        (PERIPH_BASE + 0x10000)
6  /* AHB 总线基地址 */
7  #define AHBPERIPH BASE         (PERIPH BASE + 0x20000)
8
9  /*GPIO 外设基地址*/
10 #define GPIOA_BASE             (APB2PERIPH_BASE + 0x0800)
11 #define GPIOB_BASE             (APB2PERIPH_BASE + 0x0C00)
12 #define GPIOC_BASE             (APB2PERIPH_BASE + 0x1000)
13 #define GPIOD_BASE             (APB2PERIPH_BASE + 0x1400)
14 #define GPIOE_BASE             (APB2PERIPH_BASE + 0x1800)
15 #define GPIOF_BASE             (APB2PERIPH_BASE + 0x1C00)
16 #define GPIOG_BASE             (APB2PERIPH_BASE + 0x2000)
17
18 /*RCC 外设基地址*/
19 #define RCC_BASE       (AHBPERIPH_BASE + 0x1000)
```

GPIO寄存器结构体定义

代码清单 9-2 指向外设首地址的结构体指针

```
1 // GPIO 外设声明
 2 #define GPIOA                    ((GPIO_TypeDef *) GPIOA_BASE)
 3 #define GPIOB                    ((GPIO_TypeDef *) GPIOB_BASE)
 4 #define GPIOC                    ((GPIO_TypeDef *) GPIOC_BASE)
 5 #define GPIOD                    ((GPIO_TypeDef *) GPIOD_BASE)
 6 #define GPIOE                    ((GPIO_TypeDef *) GPIOE_BASE)
 7 #define GPIOF                    ((GPIO_TypeDef *) GPIOF_BASE)
 8 #define GPIOG                    ((GPIO_TypeDef *) GPIOG_BASE)
 9
10
11 // RCC 外设声明
12 #define RCC                      ((RCC_TypeDef *) RCC_BASE)
13
14 /*RCC 的 AHB1 时钟使能寄存器地址,强制转换成指针*/
15 #define RCC_APB2ENR       *(unsigned int*)(RCC_BASE+0x18)
```

GPIO置位和复位函数

代码清单 9-5 GPIO 置位函数与复位函数的定义

```
1  /**
2    *函数功能：设置引脚为高电平
3    *参数说明：GPIOx:该参数为 GPIO_TypeDef 类型的指针，指向 GPIO 端口的地址
4    *          GPIO_Pin:选择要设置的 GPIO 端口引脚，可输入宏 GPIO_Pin_0-15，
5    *                   表示 GPIOx 端口的 0-15 号引脚。
6    */
7  void GPIO_SetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
8  {
9      /*设置 GPIOx 端口 BSRR 寄存器的第 GPIO_Pin 位，使其输出高电平*/
10     /*因为 BSRR 寄存器写 0 不影响，
11       宏 GPIO_Pin 只是对应位为 1，其它位均为 0，所以可以直接赋值*/
12
13     GPIOx->BSRR = GPIO_Pin;
14 }
15
16 /**
17   *函数功能：设置引脚为低电平
18   *参数说明：GPIOx:该参数为 GPIO_TypeDef 类型的指针，指向 GPIO 端口的地址
19   *          GPIO_Pin:选择要设置的 GPIO 端口引脚，可输入宏 GPIO_Pin_0-15，
20   *                   表示 GPIOx 端口的 0-15 号引脚。
21   */
22 void GPIO_ResetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
23 {
24     /*设置 GPIOx 端口 BRR 寄存器的第 GPIO_Pin 位,使其输出低电平*/
25     /*因为 BRR 寄存器写 0 不影响，
26     宏 GPIO_Pin 只是对应位为 1，其它位均为 0，所以可以直接赋值*/
27
28     GPIOx->BRR = GPIO_Pin;
29 }
```

GPIO置位和复位函数

代码清单 9-7 选择引脚参数的宏

```
1  /*GPIO 引脚号定义*/
2  #define GPIO_Pin_0                 (uint16_t)0x0001)  /*!< 选择 Pin0 (1<<0) */
3  #define GPIO_Pin_1                 ((uint16_t)0x0002)  /*!< 选择 Pin1 (1<<1)*/
4  #define GPIO_Pin_2                 ((uint16_t)0x0004)  /*!< 选择 Pin2 (1<<2)*/
5  #define GPIO_Pin_3                 ((uint16_t)0x0008)  /*!< 选择 Pin3 (1<<3)*/
6  #define GPIO_Pin_4                 ((uint16_t)0x0010)  /*!< 选择 Pin4 */
7  #define GPIO_Pin_5                 ((uint16_t)0x0020)  /*!< 选择 Pin5 */
8  #define GPIO_Pin_6                 ((uint16_t)0x0040)  /*!< 选择 Pin6 */
9  #define GPIO_Pin_7                 ((uint16_t)0x0080)  /*!< 选择 Pin7 */
10 #define GPIO_Pin_8                 ((uint16_t)0x0100)  /*!< 选择 Pin8 */
11 #define GPIO_Pin_9                 ((uint16_t)0x0200)  /*!< 选择 Pin9 */
12 #define GPIO_Pin_10                ((uint16_t)0x0400)  /*!< 选择 Pin10 */
13 #define GPIO_Pin_11                ((uint16_t)0x0800)  /*!< 选择 Pin11 */
14 #define GPIO_Pin_12                ((uint16_t)0x1000)  /*!< 选择 Pin12 */
15 #define GPIO_Pin_13                ((uint16_t)0x2000)  /*!< 选择 Pin13 */
16 #define GPIO_Pin_14                ((uint16_t)0x4000)  /*!< 选择 Pin14 */
17 #define GPIO_Pin_15                ((uint16_t)0x8000)  /*!< 选择 Pin15 */
```

GPIO初始化结构体

代码清单 9-9 定义 GPIO 初始化结构体

```
1 typedef struct
2 {
3     uint16_t GPIO_Pin;        /*!< 选择要配置的 GPIO 引脚 */
4
5     uint16_t GPIO_Speed;      /*!< 选择 GPIO 引脚的速率 */
6
7     uint16_t GPIO_Mode;       /*!< 选择 GPIO 引脚的工作模式 */
8 } GPIO_InitTypeDef;
```

GPIO初始化结构体

代码清单 9-9 定义 GPIO 初始化结构体

```
1 typedef struct
2 {
3     uint16_t GPIO_Pin;        /*!< 选择要配置的 GPIO 引脚 */
4
5     uint16_t GPIO_Speed;      /*!< 选择 GPIO 引脚的速率 */
6
7     uint16_t GPIO_Mode;       /*!< 选择 GPIO 引脚的工作模式 */
8 } GPIO_InitTypeDef;
```

GPIO初始化结构体

代码清单 9-10 GPIO 枚举类型定义

```
1  /**
2    * GPIO 输出速率枚举定义
3    */
4  typedef enum
5  {
6      GPIO_Speed_10MHz = 1,          // 10MHZ         (01)b
7      GPIO_Speed_2MHz,               // 2MHZ          (10)b
8      GPIO_Speed_50MHz               // 50MHZ         (11)b
9  } GPIOSpeed_TypeDef;
10
11 /**
12   * GPIO 工作模式枚举定义
13   */
14 typedef enum
15 {
16     GPIO_Mode_AIN = 0x0,             // 模拟输入      (0000 0000)b
17     GPIO_Mode_IN_FLOATING = 0x04,    // 浮空输入      (0000 0100)b
18     GPIO_Mode_IPD = 0x28,            // 下拉输入      (0010 1000)b
19     GPIO_Mode_IPU = 0x48,            // 上拉输入      (0100 1000)b
20
21     GPIO_Mode_Out_OD = 0x14,         // 开漏输出      (0001 0100)b
22     GPIO_Mode_Out_PP = 0x10,         // 推挽输出      (0001 0000)b
23     GPIO_Mode_AF_OD = 0x1C,          // 复用开漏输出  (0001 1100)b
24     GPIO_Mode_AF_PP = 0x18           // 复用推挽输出  (0001 1000)b
25 } GPIOMode_TypeDef;
```

GPIO初始化结构体

STM32F103 GPIO 引脚工作模式真值表分析										
			十六进制	二进制							
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
					上/下拉		输入/出	具体工作模式对照寄存器说明			
GPIOMode_TypeDef	GPIO_Mode_AIN	模拟输入	0X00	0	0	0	0	0	0	0	0
	GPIO_Mode_IN_FLOATING	浮空输入	0X04	0	0	0	0	0	1	0	0
	GPIO_Mode_IPD	下拉输入	0X28	0	0	1	0	0	0	0	0
	GPIO_Mode_IPU	上拉输入	0X48	0	1	0	0	0	0	0	0
	GPIO_Mode_Out_OD	开漏输出	0X14	0	0	0	1	0	1	0	0
	GPIO_Mode_Out_PP	推挽输出	0X10	0	0	0	1	0	0	0	0
	GPIO_Mode_AF_OD	复用开漏输出	0X1C	0	0	0	1	1	1	0	0
	GPIO_Mode_AF_PP	复用推挽输出	0X18	0	0	0	1	1	0	0	0
这8个宏的高4bit可随意设置，只要能在程序上帮助判断出模式即可，真正写到寄存器的值是bit2和bit3											

图 9-6 GPIO 引脚工作模式真值表分析

如果但从这些枚举值的十六进制来看，很难发现规律，转化成二进制之后，就比较容易发现规律。bit4 用来区分端口是输入还是输出，0 表示输入，1 表示输出，bit2 和 bit3 对应寄存器的 CNF_Y[1:0]位，是我们真正要写入到 CRL 和 CRH 这两个端口控制寄存器中的值。bit0 和 bit1 对应寄存器的 $MODE_Y$[1:0]位，这里我们暂不初始化，在 GPIO_Init()初始化函数中用来跟 GPIOSpeed 的值相加即可实现速率的配置。有关具体的代码分析见 GPIO_Init()库函数。其中在下拉输入和上拉输入中我们设置 bit5 和 bit6 的值为 01 和 10 来以示区别。

GPIO初始化结构体

代码清单 9-11 使用枚举定义的 GPIO 初始化结构体

```
1  /**
2    * GPIO 初始化结构体类型定义
3    */
4  typedef struct
5  {
6      uint16_t GPIO_Pin;              /*!< 选择要配置的 GPIO 引脚
7                                          可输入 GPIO_Pin_ 定义的宏 */
8
9      GPIOSpeed_TypeDef GPIO_Speed;   /*!< 选择 GPIO 引脚的速率
10                                         可输入 GPIOSpeed TypeDef 定义的枚举值
*/
11
12     GPIOMode_TypeDef GPIO_Mode;     /*!< 选择 GPIO 引脚的工作模式
13                                         可输入 GPIOMode_TypeDef 定义的枚举值
*/
14 } GPIO_InitTypeDef;
```

GPIO初始化函数

GPIO_Init（ ）函数具体可参考

程序源码里面的配套例程

或者零死角教程里面的程序讲解

THANKS

论坛： www.firebbs.cn

淘宝：firestm32.taobao.com