

【STM32】STM32端口复用和重映射（AFIO辅助功能时钟）

原创

Yngz_Miao

2018-04-06 18:32:56

15296

★ 收藏

46

版权

分类专栏：[《嵌入式》STM32开发笔记](#)

STM32F1xx官方资料：

《STM32中文参考手册V10》-第8章通用和复用功能IO(GPIO和AFIO)

端口复用功能

端口复用的定义

STM32有许多的内置外设（如串口、ADC、DCA等等），这些外设的外部引脚都是和GPIO复用的。也就是说，一个GPIO如果可以复用为内置外设的功能引脚，那么当这个GPIO作为内置外设使用的时候，就叫复用。详细的可以参考《STM32F103ZET6数据手册》p30的内容，表格的倒数第二栏就表示端口复用功能。

比如说，STM32的串口1的引脚对应的I/O位PA9、PA10。而PA9、PA10默认的功能都是GPIO，所以说当PA9、PA10引脚作为串口1使用的时候就是端口复用。

G12	F10	E1	37	63	96	PC6	I/O	FT	PC6	I2S2_MCK/ TIM8_CH1/SDIO_D6	TIM3_CH1
F12	E10	E2	38	64	97	PC7	I/O	FT	PC7	I2S3_MCK/ TIM8_CH2/SDIO_D7	TIM3_CH2
F11	F9	E3	39	65	98	PC8	I/O	FT	PC8	TIM8_CH3/SDIO_D0	TIM3_CH3
E11	E9	D1	40	66	99	PC9	I/O	FT	PC9	TIM8_CH4/SDIO_D1	TIM3_CH4
E12	D9	E4	41	67	100	PA8	I/O	FT	PA8	USART1_CK/ TIM1_CH1 ⁽⁷⁾ /MCO	
D12	C9	D2	42	68	101	PA9	I/O	FT	PA9	USART1_TX ⁽⁷⁾ / TIM1_CH2 ⁽⁷⁾	
D11	D10	D3	43	69	102	PA10	I/O	FT	PA10	USART1_RX ⁽⁷⁾ / TIM1_CH3 ⁽⁷⁾	
C12	C10	C1	44	70	103	PA11	I/O	FT	PA11	USART1_CTS/USBDM CAN_RX ⁽⁷⁾ /TIM1_CH4 ⁽⁷⁾	
B12	B10	C2	45	71	104	PA12	I/O	FT	PA12	USART1_RTS/USBDM CAN_TX ⁽⁷⁾ /TIM1_ETR ⁽⁷⁾	

那么，什么时候端口是默认功能，什么时候端口是复用功能呢？

STM32时钟系统的配置除了初始化的时候在system_stm32f10x.c中的SystemInit函数中外，其他的配置主要在stm32f10x_rcc.c文件中，所以GPIO等等外设的时钟使能函数都是在此文件中。同时我们通过函数名可以得到规律：GPIOA-GPIOC是挂载在APB2下面，TIM2-TIM4是挂载在APB1下面，DMA是挂载在AHB下面。所以调用函数的名称是需要根据这个来确定的。

端口复用初始化过程

接下来看一下端口复用初始化过程的步骤，拿串口1为例：

1、GPIO端口时钟使能。要使用到端口复用，首先是要使能端口的时钟了；

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
```

2、复用的外设时钟使能。比如要将PA9、PA10引脚复用成串口，必须也要使能串口时钟；

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
```

3、端口模式配置。在I/O复用位内置外设功能引脚的时候，必须设置GPIO端口的模式。至于在复用功能下，GPIO的模式怎么设置，可以查看手册《STM32中文参考手册》p110的内容。这里拿USART1为例，进行配置，要配置全双工的串口1，TX引脚需要推挽复用输出，RX引脚需要浮空输入或者上拉输入；

表21 USART

USART引脚	配置	GPIO配置
USARTx_TX	全双工模式	推挽复用输出
	半双工同步模式	推挽复用输出
USARTx_RX	全双工模式	浮空输入或带上拉输入
	半双工同步模式	未用，可作为通用I/O
USARTx_CK	同步模式	推挽复用输出
USARTx_RTS	硬件流量控制	推挽复用输出
USARTx_CTS	硬件流量控制	浮空输入或带上拉输入

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9; //PA.9//复用推挽输出
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP; //复用推挽输出
GPIO_Init(GPIOA, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10; //PA10 PA.10 浮空输入
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING; //浮空输入
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

总而言之，使用复用功能的时候至少要使能2时钟：GPIO时钟使能、复用的外设时钟使能。同时还要初始化GPIO以及复用外设功能（端口模式配置）。

端口重映射

端口重映射的定义

为了使不同的器件封装的外设I/O功能数量达到最优，可以把一些复用功能重新映射到其他的引脚上。STM32中有许多的内置外设的输入、输出引脚都具有重映射（Remap）的功能。

我们知道，每个内置外设都有若干个输入、输出引脚，一般这些引脚的输出端口都是固定不变的，为了更好地安排引脚的走向和功能，在STM32中引入了外设引脚重映射的概念，即一个外设的引脚除了具有默认的端口之外，还可以通过设定重映射寄存器的方式把这个外设的引脚映射到其他的端口。

简单讲，就是把引脚的外设功能映射到其他的引脚上，但不是可以随便映射的，具体的对应关系参考《STM32F103ZET6数据手册》p30的内容，表格的最后一栏就表示端口重映射功能。

这里同样用串口1为例来说明。

Pins						Pin name	Type ⁽¹⁾	I / O Level ⁽²⁾	Main function ⁽³⁾ (after reset)	Alternate functions	
BGA144	BGA100	WLCSP64	LQFP64	LQFP100	LQFP144					Default	Remap
A6	A6	B4	56	90	134	PB4	I/O	FT	NJTRST	SPI3_MISO	PB4 / TIM3_CH1 SPI1_MISO
B6	C5	A5	57	91	135	PB5	I/O		PB5	I2C1_SMBA/ SPI3_MOSI I2S3_SD	TIM3_CH2 / SPI1_MOSI
C6	B5	B5	58	92	136	PB6	I/O	FT	PB6	I2C1_SCL ⁽⁷⁾ / TIM4_CH1 ⁽⁷⁾	USART1_TX
D6	A5	C5	59	93	137	PB7	I/O	FT	PB7	I2C1_SDA ⁽⁷⁾ / FSMC_NADV / TIM4_CH2 ⁽⁷⁾	USART1_RX
D5	D5	A6	60	94	138	BOOT0	I		BOOT0		
C5	B4	D5	61	95	139	PB8	I/O	FT	PB8	TIM4_CH3 ⁽⁷⁾ /SDIO_D4	I2C1_SCL/

可以看出，我们可以将**串口1**重映射到**PB6、PB7**引脚上。

端口重映射初始化过程

接下来看一下端口重映射初始化过程的步骤，拿串口1为例，除了之前使能复用功能的2个时钟之外，还需要使能AFIO功能时钟，然后调用重映射函数：

- 1、GPIO端口时钟使能。要使用到端口复用，首先是要使能端口的时钟了；

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
```

- 2、复用的外设时钟使能。比如要将PB6、PB7引脚复用成串口，必须也要使能串口时钟；

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
```

- 3、使能AFIO时钟。重映射必须使能AFIO时钟；

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);
```

- 4、开启重映射；

```
GPIO_PinRemapConfig(GPIO_Remap_USART1, ENABLE);
```

这样，就将串口1的TX和RX引脚映射到PB6、PB7引脚上面了。至于哪些功能可以重映射，除了查看中文参考手册之外，还可以从GPIO_PinRemapConfig函数入手查看第一个入口参数的取值范围的值。stm32f10x_gpio.h中定义了一些宏定义的标识符：

```
#define GPIO_Remap_SPI1          ((uint32_t)0x00000001)  /*!< SPI1 Alternate Function mapping
#define GPIO_Remap_I2C1          ((uint32_t)0x00000002)  /*!< I2C1 Alternate Function mapping
#define GPIO_Remap_USART1       ((uint32_t)0x00000004)  /*!< USART1 Alternate Function mappin
#define GPIO_Remap_USART2       ((uint32_t)0x00000008)  /*!< USART2 Alternate Function mappin
#define GPIO_PartialRemap_USART3 ((uint32_t)0x00140010)  /*!< USART3 Partial Alternate Functio
#define GPIO_FullRemap_USART3   ((uint32_t)0x00140030)  /*!< USART3 Full Alternate Function m
#define GPIO_PartialRemap_TIM1   ((uint32_t)0x00160040)  /*!< TIM1 Partial Alternate Function
```

```
#define GPIO_FullRemap_TIM1      ((uint32_t)0x001600C0)  /*!< TIM1 Full Alternate Function map

#define GPIO_PartialRemap1_TIM2  ((uint32_t)0x00180100)  /*!< TIM2 Partial1 Alternate Function ma
#define GPIO_PartialRemap2_TIM2  ((uint32_t)0x00180200)  /*!< TIM2 Partial2 Alternate Function ma
#define GPIO_FullRemap_TIM2      ((uint32_t)0x00180300)  /*!< TIM2 Full Alternate Function mappin
#define GPIO_PartialRemap_TIM3   ((uint32_t)0x001A0800)  /*!< TIM3 Partial Alternate Function map
#define GPIO_FullRemap_TIM3      ((uint32_t)0x001A0C00)  /*!< TIM3 Full Alternate Function mappin
```

可以看出，USART1只有一种重映射，而USART3存在部分重映射和完全重映射。所谓部分重映射就是部分引脚和默认的是一样的，完全重映射就是所有引脚都映射到了新的引脚。可以查看《STM32中文参考手册》p119的内容查看部分重映射和完全重映射的内容。而在之前最后开启重映射的函数中，根据第一个参数，来确定是部分重映射还是全部重映射。

AFIO辅助功能时钟

之前在端口重映射的时候，讲到要使能AFIO辅助功能时钟。那么什么时候需要开启（使能）呢？

对寄存器AFIO_MAPR、AFIO_EXTICRx和AFIO_EVCR进行读写操作前，应当首先开启AFIO时钟。

- AFIO_MAPR：配置复用功能重映射
- AFIO_EXTICRx：配置外部中断线映射
- AFIO_EVCR：配置事件输出



微信号：Yngz_Miao
邮箱：yngzMiao@163.com
博客：blog.csdn.net/qq_38410730

https://blog.csdn.net/qq_38410730

点赞 13 评论 2 分享 收藏 46 手机看 打赏 ... 关注