零死角玩转STM32

I2C—读写EEPROM

淘宝：firestm32.taobao.com

论坛： www.chuxue123.com

扫描进入淘宝店铺

主讲内容

01 I2C协议简介

02 STM32的I2C特性及架构

03 I2C初始化结构体详解

04 I2C—读写EEPROM实验

参考资料:《零死角玩转STM32》

“I2C—读写EEPROM”章节

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_ClockSpeed

设置I2C的传输速率，在调用初始化函数时，函数会根据我们输入的数值经过运算后把时钟因子写入到I2C的时钟控制寄存器CCR。而我们写入的这个参数值不得高于400KHz。

实际上由于CCR寄存器不能写入小数类型的时钟因子，影响到SCL的实际频率可能会低于本成员设置的参数值，这时除了通讯稍慢一点以外，不会对I2C的标准通讯造成其它影响。

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_Mode

选择I2C的使用方式，有I2C模式(I2C_Mode_I2C)和SMBus主、从模式(I2C_Mode_SMBusHost、 I2C_Mode_SMBusDevice) 。

I2C不需要在此处区分主从模式，直接设置I2C_Mode_I2C即可。

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_DutyCycle

设置I²C的SCL线时钟的占空比。该配置有两个选择，分别为低电平时间比高电平时间为2：1 (I2C_DutyCycle_2)和16：9 (I2C_DutyCycle_16_9)。

其实这两个模式的比例差别并不大，一般要求都不会如此严格，这里随便选就可以了。

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_OwnAddress1

配置STM32的I2C设备自己的地址，每个连接到I2C总线上的设备都要有一个自己的地址，作为主机也不例外。地址可设置为7位或10位(受下面I2C_AcknowledgeAddress成员决定)，只要该地址是I2C总线上唯一的即可。

STM32的I2C外设可同时使用两个地址，即同时对两个地址作出响应，这个结构成员I2C_OwnAddress1配置的是默认的、OAR1寄存器存储的地址，若需要设置第二个地址寄存器OAR2，可使用I2C_OwnAddress2Config函数来配置，OAR2不支持10位地址。

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_Ack_Enable

配置I²C应答是否使能，设置为使能则可以发送响应信号。一般配置为允许应答(I2C_Ack_Enable)，这是绝大多数遵循I²C标准的设备的通讯要求，改为禁止应答(I2C_Ack_Disable)往往会导致通讯错误。

I2C—读写EEPROM

I2C初始化结构体详解

```
1 typedef struct {
2     uint32_t I2C_ClockSpeed; /*!< 设置 SCL 时钟频率，此值要低于 40 0000*/
3     uint16_t I2C_Mode;       /*!< 指定工作模式，可选 I2C 模式及 SMBUS 模式 */
4     uint16_t I2C_DutyCycle; /*指定时钟占空比，可选 low/high = 2:1 及 16:9 模式*/
5     uint16_t I2C_OwnAddress1;      /*!< 指定自身的 I2C 设备地址 */
6     uint16_t I2C_Ack;                  /*!< 使能或关闭响应(一般都要使能) */
7     uint16_t I2C_AcknowledgedAddress; /*!< 指定地址的长度，可为 7 位及 10 位 */
8 } I2C_InitTypeDef;
```

- I2C_AcknowledgeAddress

选择I2C的寻址模式是7位还是10位地址。这需要根据实际连接到I2C总线上设备的地址进行选择，这个成员的配置也影响到I2C_OwnAddress1成员，只有这里设置成10位模式时，I2C_OwnAddress1才支持10位地址。

配置完这些结构体成员值，调用库函数I2C_Init即可把结构体的配置写入到寄存器中。

THANKS

论坛： www.chuxue123.com

淘宝：firestm32.taobao.com

扫描进入淘宝店铺