零死角玩转STM32

I2C—读写EEPROM

淘宝：firestm32.taobao.com

论坛： www.firebbs.cn

扫描进入淘宝店铺

主讲内容

参考资料:《零死角玩转STM32》

“I2C—读写EEPROM”章节

I2C—读写EEPROM

STM32的I2C特性及架构

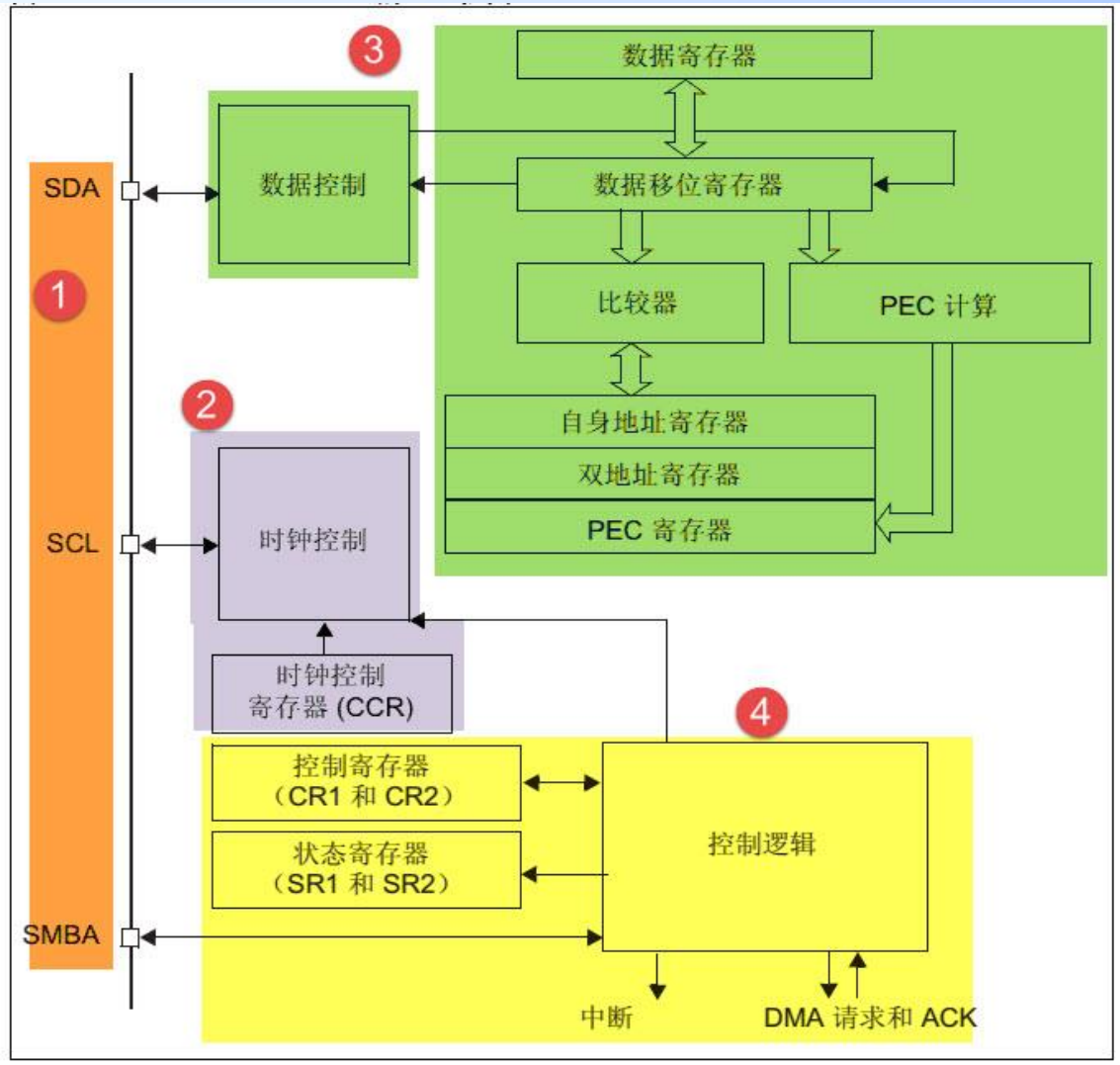
软件模拟协议：使用CPU直接控制通讯引脚的电平，产生出符合通讯协议标准的逻辑。

硬件实现协议：由STM32的I2C片上外设专门负责实现I2C通讯协议，只要配置好该外设，它就会自动根据协议要求产生通讯信号，收发数据并缓存起来，CPU只要检测该外设的状态和访问数据寄存器，就能完成数据收发。这种由硬件外设处理I2C协议的方式减轻了CPU的工作，且使软件设计更加简单。

STM32的I2C外设可用作通讯的主机及从机，支持100Kbit/s和400Kbit/s的速率，支持7位、10位设备地址，支持DMA数据传输，并具有数据校验功能。

I2C—读写EEPROM

STM32的I2C架构剖析



- 通讯引脚
- 时钟控制逻辑
- 数据控制逻辑
- 整体控制逻辑

I2C—读写EEPROM

1.通讯引脚

STM32芯片有多个I2C外设，它们的I2C通讯信号引出到不同的GPIO引脚上，使用时必须配置到这些指定的引脚，以《STM32F10x规格书》为准。

引脚	I2C编号	
	I2C1	I2C2
SCL	PB5/PB8(重映射)	PB10
SDA	PB6/PB9(重映射)	PB11

I2C—读写EEPROM

2.时钟控制逻辑

SCL线的时钟信号，由I^2C接口根据时钟控制寄存器(CCR)控制，控制的参数主要为时钟频率。

- 可选择I2C通讯的“标准/快速”模式，这两个模式分别I2C对应100/400Kbit/s的通讯速率。
- 在快速模式下可选择SCL时钟的占空比，可选$T_{low}/T_{high}=2$或$T_{low}/T_{high}=16/9$模式。
- CCR寄存器中12位的配置因子CCR，它与I2C外设的输入时钟源共同作用，产生SCL时钟。STM32的I2C外设输入时钟源为PCLK1。

I2C—读写EEPROM

计算时钟频率：

标准模式：

$T_{high}=CCR*T_{PCKL1}$　　$T_{low}=CCR*T_{PCLK1}$

快速模式中$T_{low}/T_{high}=2$时：

$T_{high}=CCR*T_{PCKL1}$　　$T_{low}=2*CCR*T_{PCKL1}$

快速模式中$T_{low}/T_{high}=16/9$时：

$T_{high}=9*CCR*T_{PCKL1}$　　$T_{low}=16*CCR*T_{PCKL1}$

例如，我们的PCLK1=36MHz，想要配置400Kbit/s的速率，计算方式如下：

PCLK时钟周期：	TPCLK1 = 1/36000000
目标SCL时钟周期：	TSCL = 1/400000
SCL时钟周期内的高电平时间：	THIGH = TSCL/3
SCL时钟周期内的低电平时间：	TLOW = 2*TSCL/3
计算CCR的值：	CCR = THIGH/TPCLK1 = 30

计算出来的CCR值写入到寄存器即可。

I2C—读写EEPROM

3.数据控制逻辑

I2C的SDA信号主要连接到数据移位寄存器上，数据移位寄存器的数据来源及目标是数据寄存器(DR)、地址寄存器(OAR)、PEC寄存器以及SDA数据线。

- 当向外发送数据的时候，数据移位寄存器以“数据寄存器”为数据源，把数据一位一位地通过SDA信号线发送出去；
- 当从外部接收数据的时候，数据移位寄存器把SDA信号线采样到的数据一位一位地存储到“数据寄存器”中。

I2C—读写EEPROM

4.整体控制逻辑

整体控制逻辑负责协调整个I2C外设，控制逻辑的工作模式根据我们配置的“控制寄存器(CR1/CR2)”的参数而改变。

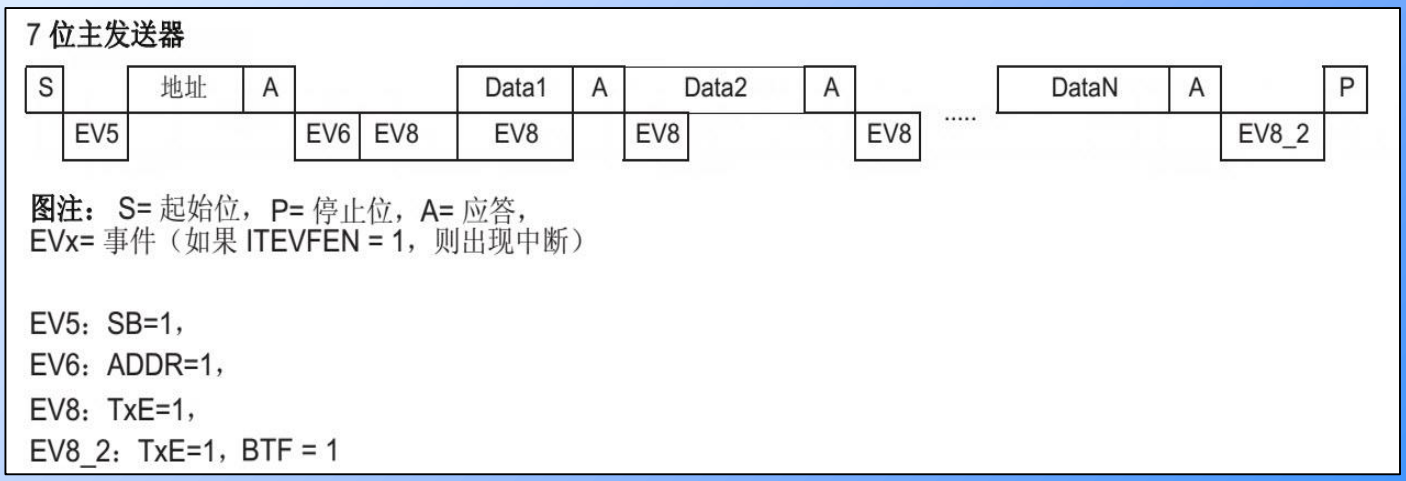
在外设工作时，控制逻辑会根据外设的工作状态修改“状态寄存器(SR1和SR2)”，只要读取这些寄存器相关的寄存器位，就可以了解I2C的工作状态。

I2C—读写EEPROM

STM32的I2C通讯过程

使用I2C外设通讯时，在通讯的不同阶段它会对“状态寄存器(SR1及SR2)”的不同数据位写入参数，通过读取这些寄存器标志来了解通讯状态。

1.主发送器



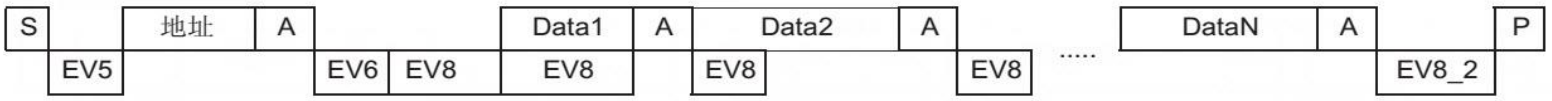
主发送器通讯过程

可使用STM32标准库函数来直接检测这些事件的复合标志，降低编程难度。

I2C—读写EEPROM

1.主发送器通讯过程

7 位主发送器



图注： S= 起始位，P= 停止位，A= 应答，
EVx= 事件（如果 ITEVFEN = 1，则出现中断）

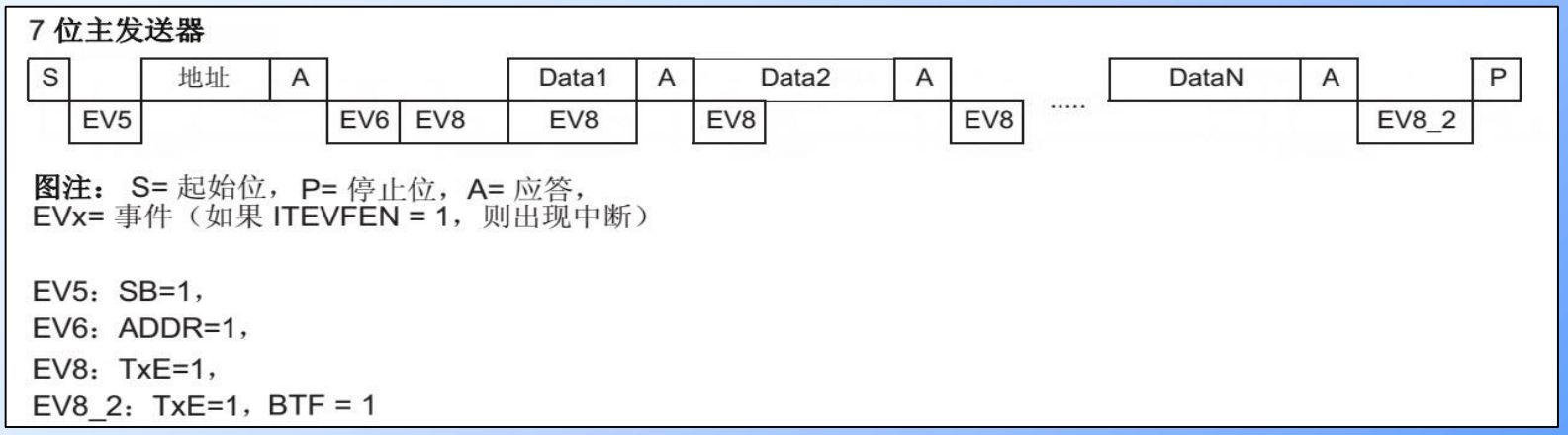
EV5：SB=1，
EV6：ADDR=1，
EV8：TxE=1，
EV8_2：TxE=1，BTF = 1

- 控制产生起始信号(S)，当发生起始信号后，它产生事件“EV5”，并会对SR1寄存器的“SB”位置1，表示起始信号已经发送；

- 发送设备地址并等待应答信号，若有从机应答，则产生事件“EV6”及“EV8”，这时SR1寄存器的“ADDR”位及“TXE”位被置1，ADDR 为1表示地址已经发送，TXE为1表示数据寄存器为空；

I2C—读写EEPROM

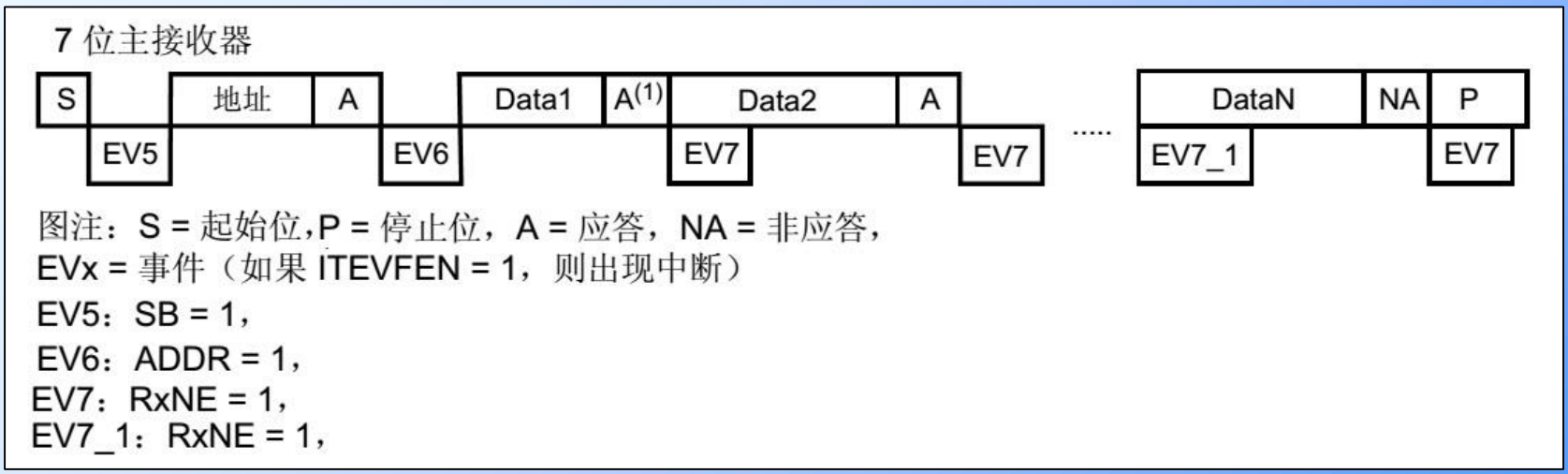
1.主发送器通讯过程



- 往I2C的“数据寄存器DR”写入要发送的数据，这时TXE位会被重置0，表示数据寄存器非空，I2C外设通过SDA信号线一位位把数据发送出去后，又会产生“EV8”事件，即TXE位被置1，重复这个过程，可以发送多个字节数据；
- 发送数据完成后，控制I2C设备产生一个停止信号(P)，这个时候会产生EV2事件，SR1的TXE位及BTF位都被置1，表示通讯结束。

I2C—读写EEPROM

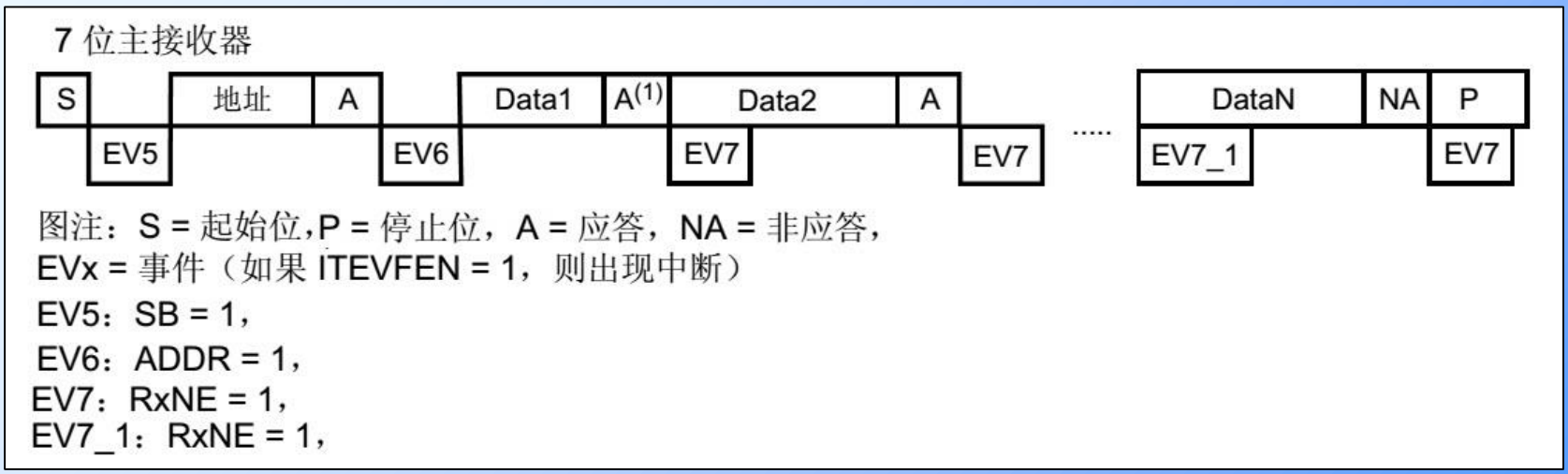
2.主接收器



- 起始信号(S)是由主机端产生的，控制发生起始信号后，它产生事件“EV5”，并会对SR1寄存器的“SB”位置1，表示起始信号已经发送；
- 发送设备地址并等待应答信号，若有从机应答，则产生事件“EV6”这时SR1寄存器的“ADDR”位被置1，表示地址已经发送。

I2C—读写EEPROM

2.主接收器



- 从机端接收到地址后，开始向主机端发送数据。当主机接收到这些数据后，会产生“EV7”事件，SR1寄存器的RXNE被置1，表示接收数据寄存器非空，读取该寄存器后，可对数据寄存器清空，以便接收下一次数据。此时可以控制I2C发送应答信号(ACK)或非应答信号(NACK)，若应答，则重复以上步骤接收数据，若非应答，则停止传输；
- 发送非应答信号后，产生停止信号(P)，结束传输。

THANKS

论坛： www.chuxue123.com

淘宝：www.firebbs.cn

扫描进入淘宝店铺