

STM32Cube高效开发教程（基础篇）

第17章 I2C接口通信

王维波

中国石油大学（华东）控制科学与工程学院

STM32Cube高效开发教程（基础篇）

作者：王维波，鄢志丹，王钊

人民邮电出版社

2021年9月出版

如果有读者需要本书课件的PPT版本用于备课，可以给作者发邮件免费获取，并可加入专门的教学和技术交流QQ群

邮箱：wangwb@upc.edu.cn



17.1 I2C总线和通讯协议

17.2 I2C的HAL驱动程序

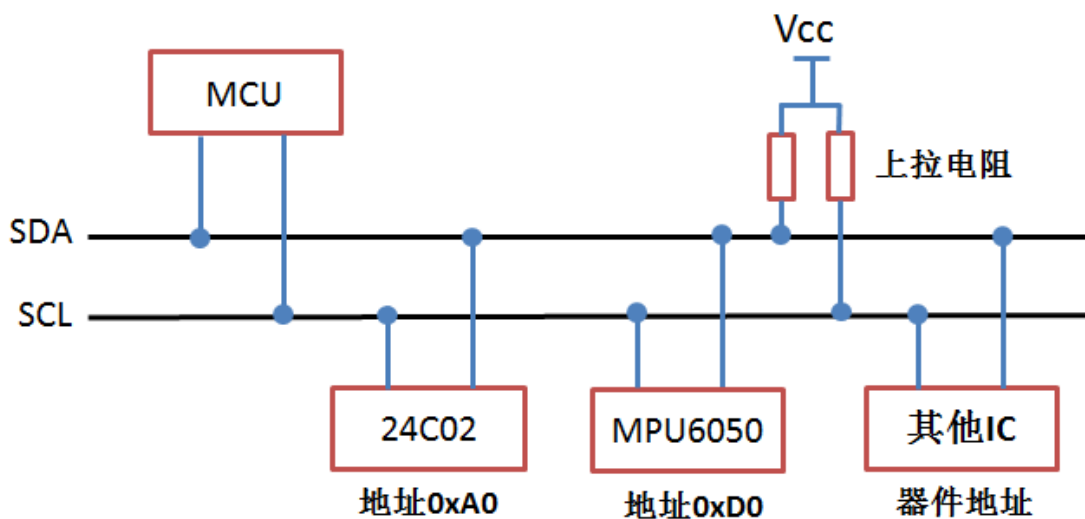
17.3 EEPROM芯片24C02

17.4 读写I2C示例

17.1.1 I2C总线结构

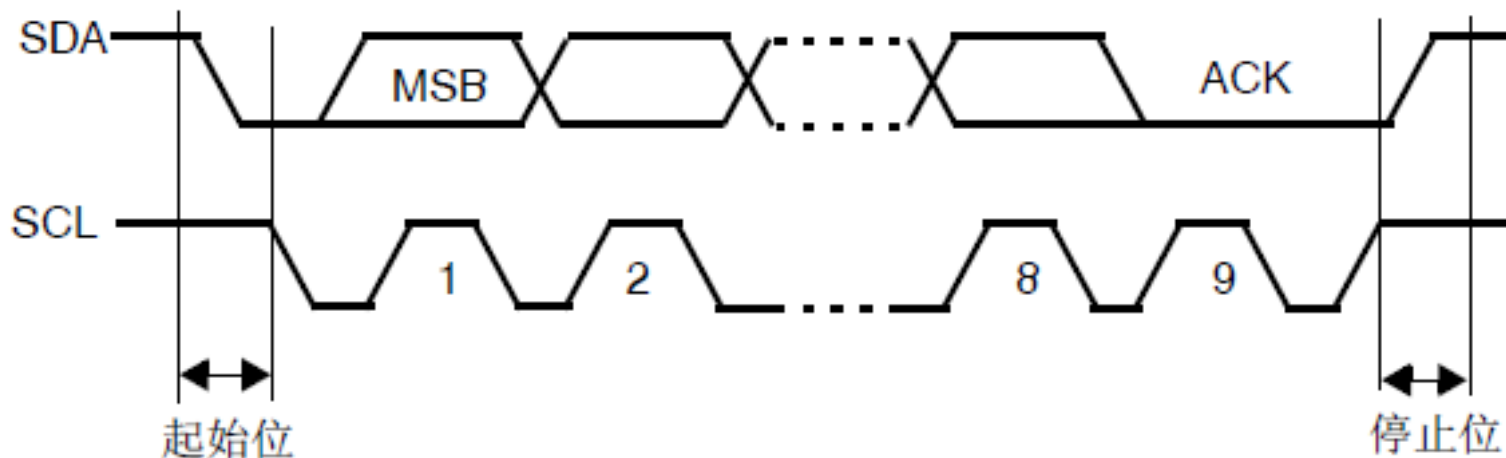
I2C接口只有2根信号线，即**双向串行数据线SDA**和**时钟信号线SCL**。I2C是一种多设备总线，一个I2C总线上可以挂载多个设备，每个设备有1个地址。

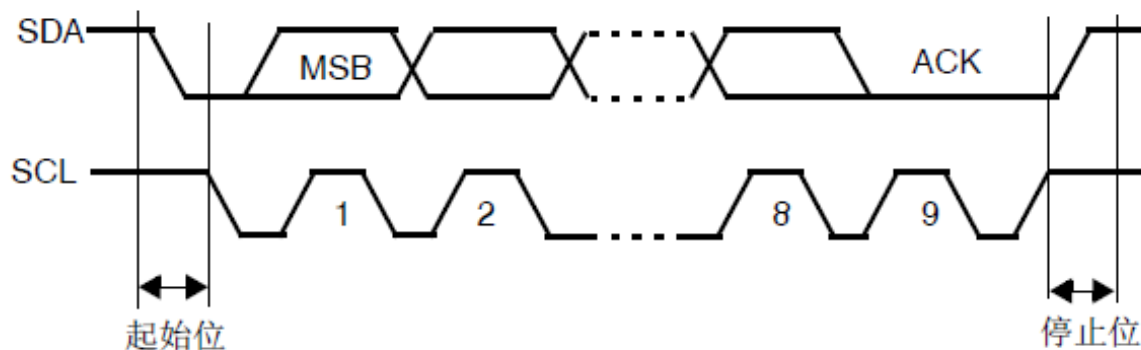
I2C通讯有标准模式和快速模式，**标准模式传输速率为100 kbps**，**快速模式为400 kbps**。



17.1.2 I2C总线通讯协议

I2C通讯总是由主机启动，每次通讯过程由起始信号开始，由停止信号结束。一个数据包是8个位，每个数据包后有一个应答位（ACK）或非应答位（NACK）





- **起始位**：当SCL是高电平时，SDA的下跳沿就是起始位
- **停止位**：当SCL为高电平时，SDA的上跳沿就是停止位
- **数据位**：在SCL的一个时钟周期内传输一个数据位。在SCL为低电平时，发送设备更新SDA的电平；当SCL为高电平时，接收设备读取SDA的电平就是有效的一位数据
- **数据包**：I2C数据通讯一个数据包总是8个位
- **应答信号**：发送设备在第9个SCL时钟周期采集接收设备的应答信号。若到的SDA为低电平，就是应答信号ACK，如果采集的SDA是高电平，就是非应答信号NACK

17.1 I2C总线和通讯协议

17.2 I2C的HAL驱动程序

17.3 EEPROM芯片24C02

17.4 读写I2C示例

17.2.1 阻塞式数据传输

函数名	功能描述
HAL_I2C_IsDeviceReady()	检查某个从设备是否准备好了I2C通讯
HAL_I2C_Master_Transmit()	作为主设备向某个地址的从设备发送一定长度的数据
HAL_I2C_Master_Receive()	作为主设备从某个地址的从设备接收一定长度的数据
HAL_I2C_Slave_Transmit()	作为从设备发送一定长度的数据
HAL_I2C_Slave_Receive()	作为从设备接收一定长度的数据
HAL_I2C_Mem_Write()	向某个从设备的指定存储地址开始写入一定长度的数据
HAL_I2C_Mem_Read()	从某个从设备的指定存储地址开始读取一定长度的数据

17.2.2 中断方式数据传输

函数名	函数功能描述	关联的回调函数
HAL_I2C_Master_Transmit_IT()	主设备向某个地址的从设备发送一定长度的数据	HAL_I2C_MasterTxCpltCallback()
HAL_I2C_Master_Receive_IT()	主设备从某个地址的从设备接收一定长度的数据	HAL_I2C_MasterRxCpltCallback()
HAL_I2C_Master_Abort_IT()	主设备主动终止中断传输过程	HAL_I2C_AbortCpltCallback()
HAL_I2C_Slave_Transmit_IT()	作为从设备发送一定长度的数据	HAL_I2C_SlaveTxCpltCallback()
HAL_I2C_Slave_Receive_IT()	作为从设备接收一定长度的数据	HAL_I2C_SlaveRxCpltCallback()
HAL_I2C_Mem_Write_IT()	向某个从设备的指定存储地址开始写入一定长度的数据	HAL_I2C_MemTxCpltCallback()
HAL_I2C_Mem_Read_IT()	从某个从设备的指定存储地址开始读取一定长度的数据	HAL_I2C_MemRxCpltCallback()
所有中断方式传输函数	中断方式传输过程出现错误	HAL_I2C_ErrorCallback()

17.2.3 DMA方式数据传输

函数名	函数功能描述	关联的回调函数
HAL_I2C_Master_Transmit_DMA()	向某个地址的从设备发送一定长度的数据	HAL_I2C_MasterTxCpltCallback()
HAL_I2C_Master_Receive_DMA()	从某个地址的从设备接收一定长度的数据	HAL_I2C_MasterRxCpltCallback()
HAL_I2C_Slave_Transmit_DMA()	作为从设备发送一定长度的数据	HAL_I2C_SlaveTxCpltCallback()
HAL_I2C_Slave_Receive_DMA()	作为从设备接收一定长度的数据	HAL_I2C_SlaveRxCpltCallback()
HAL_I2C_Mem_Write_DMA()	向某个从设备的指定存储地址开始写入一定长度的数据	HAL_I2C_MemTxCpltCallback()
HAL_I2C_Mem_Read_DMA()	从某个从设备的指定存储地址开始读取一定长度的数据	HAL_I2C_MemRxCpltCallback()

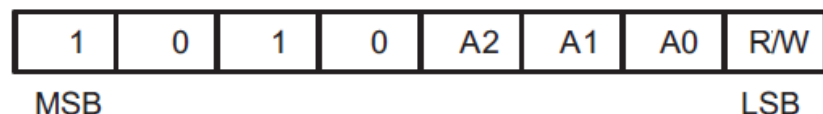
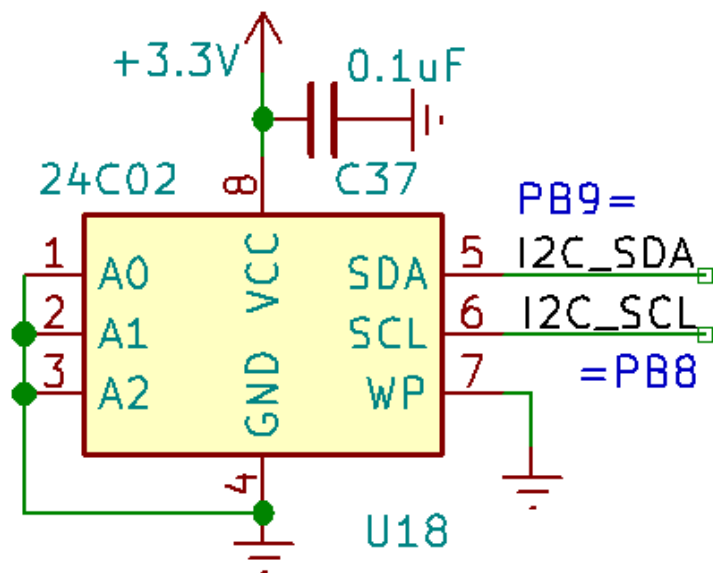
17.1 I2C总线和通讯协议

17.2 I2C的HAL驱动程序

17.3 EEPROM芯片24C02

17.4 读写I2C示例

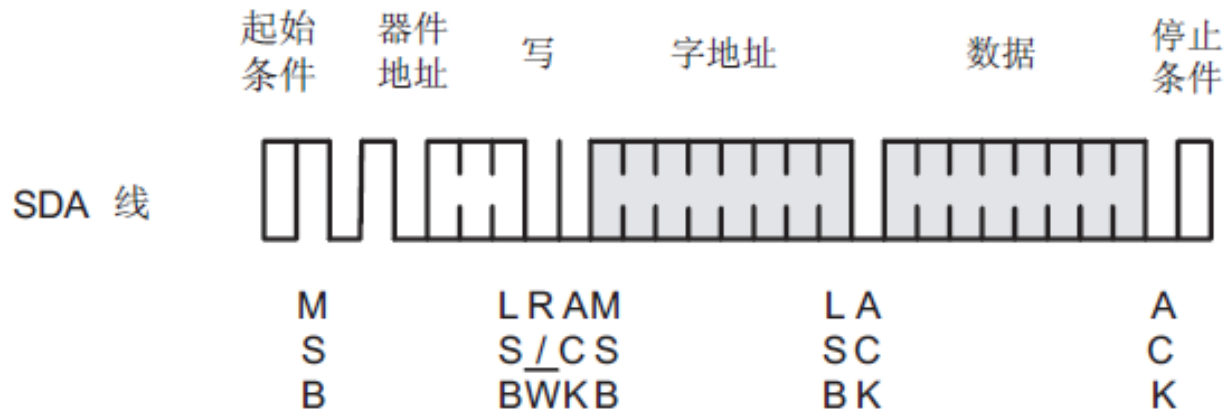
17.3.1 接口和通讯协议



24C02的设备地址组成

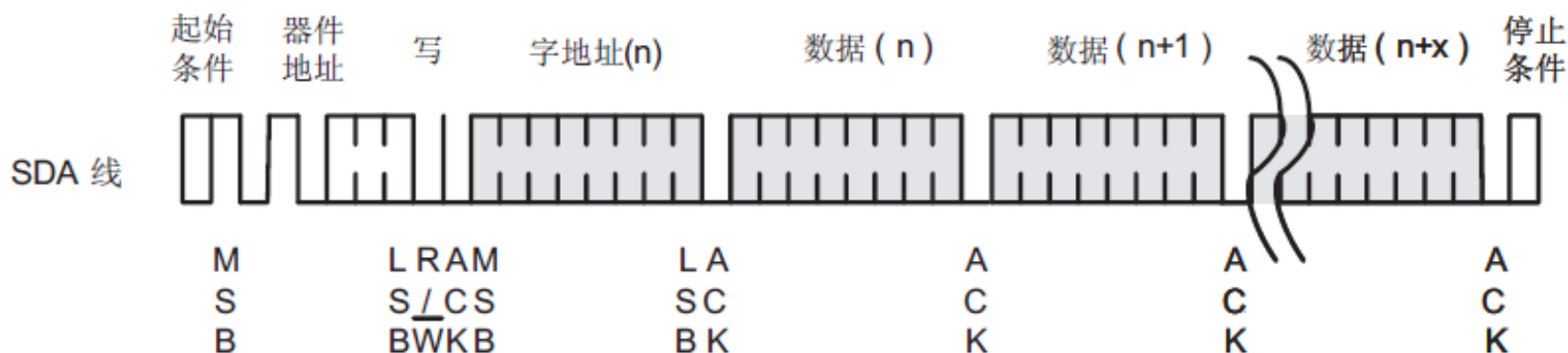
24C02的写操作地址为**0xA0**，读操作地址为0xA1。HAL驱动程序函数中需要传递I2C从设备的地址时，都使用写操作地址

1. 写一个字节数据



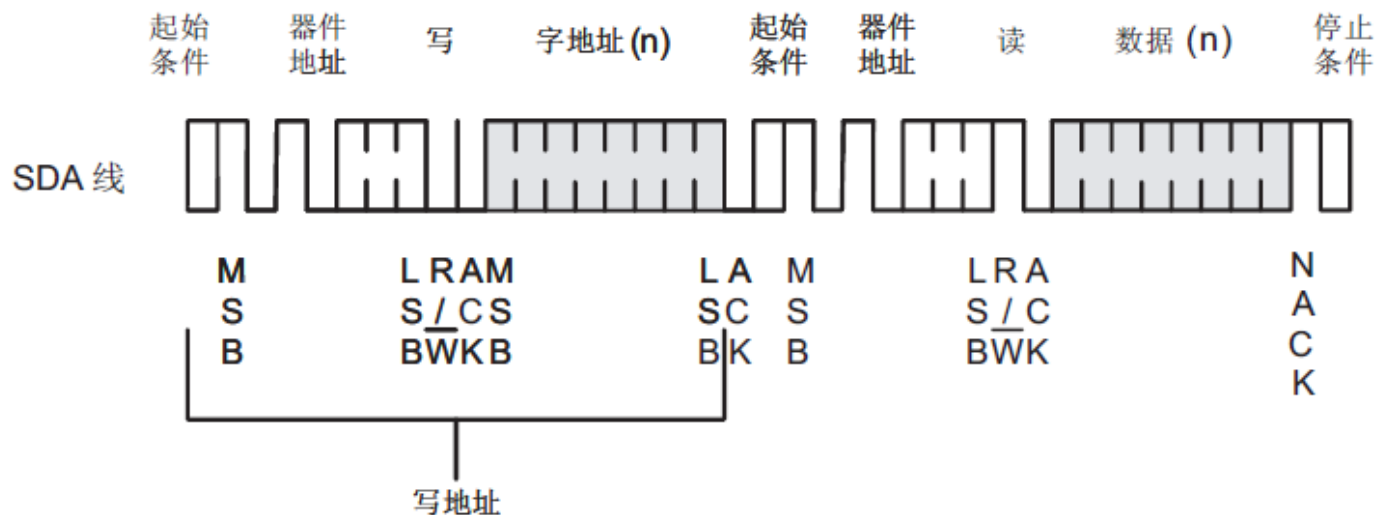
- 主机发送起始信号，然后发送器件的写操作地址
- 24C02应答ACK后，主机再发送8位地址，这是24C02内存存储单元的地址。8位地址的范围是0至255，也就是24C02内256个字节存储单元的地址
- 24C02应答ACK后主机再发送需要写入的1个字节的数据
- 从机接收完数据后应答ACK，主机发停止信号结束传输

2. 连续写多个字节数据



- 24C02内部存储区域按页划分，每页8个字节，所以256字节的存储单元分为32个页，页的起始地址是 $8 \times N$
- 在连续写数据时，如果数据起始地址在页的起始位置，则一次最多可写8个字节的数据。如果数据存储起始地址不在页的起始位置，要注意一次写入的数据不要超过页的边界

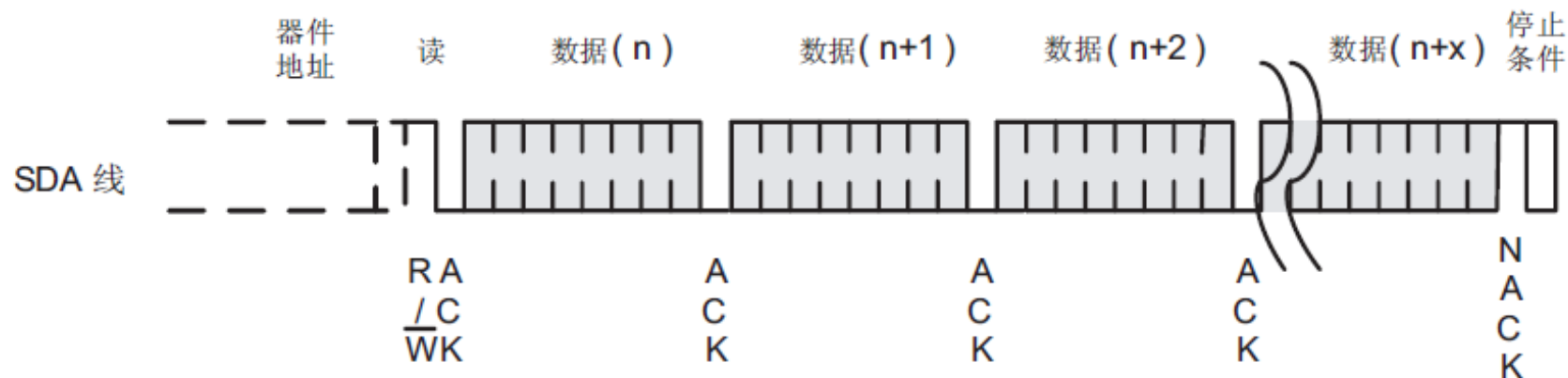
3. 读一个字节数据



可以从24C02的任何一个存储位置读取一个字节的數據。

主设备先进行一次写操作，写入需要读取的存储单元的地址，然后再进行一次读操作，读取的1字节数据就是所指定的存储地址的存储内容。

4. 连续读多个字节数据



可以从24C02一次性连续读取多个字节的数据，且读取数据时不受页边界的影响，也就是读取数据的长度可以超过8字节。

17.3.2 驱动程序设计

24C02的驱动程序文件包括头文件24cxx.h和源程序文件24cxx.c，封装了24C02常用的操作功能。

代码较长，看源程序和教材讲解

17.1 I2C总线和通讯协议

17.2 I2C的HAL驱动程序

17.3 EEPROM芯片24C02

17.4 读写I2C示例

17.4.1 示例功能和CubeMX项目设置

■ **I2C Speed Mode**，速度模式。可选标准模式（Standard Mode）或快速模式（Fast Mode）。

■ **I2C Clock Speed(Hz)**，I2C时钟速度。标准模式最高100 kHz，快速模式最高400 kHz。

I2C1 Mode and Configuration

Mode

I2C I2C

Configuration

Reset Configuration

✓ NVIC Settings ✓ DMA Settings ✓ GPIO Settings

✓ Parameter Settings ✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

Master Features	
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Slave Features	
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

I2C1接口模式和参数设置

17.4.2 程序功能实现

主程序在LCD上显示一个模拟菜单，通过写入和读出数据，验证读写功能正确。

```
[1]KeyUp   = Write a number  
[2]KeyDown = Read the number  
[3]KeyLeft  = Write a string  
[4]KeyRight= Read the string
```

程序较长，看源代码和教材讲解

运行测试

```
Demol7_1:I2C Interface
24C02:EEPROM, 256 bytes
  8 bytes/page, 32 pages
I2C Device Address=0xA0
Device is ready.

[1]KeyUp   = Write a number
[2]KeyDown = Read the number
[3]KeyLeft = Write a string
[4]KeyRight= Read the string

Write string from Page 2:
  University of Petroleum

** Reselect menu or reset **
```

写入一个字符串

```
Demol7_1:I2C Interface
24C02:EEPROM, 256 bytes
  8 bytes/page, 32 pages
I2C Device Address=0xA0
Device is ready.

[1]KeyUp   = Write a number
[2]KeyDown = Read the number
[3]KeyLeft = Write a string
[4]KeyRight= Read the string

Read string from Page 2:
  University of Petroleum

** Reselect menu or reset **
```

读取一个字符串

练习任务

开发板上有一个MPU6050芯片，这是一个集成了三轴加速度计、三轴陀螺仪和温度测量的传感器，使用I2C输出接口，连接到STM32F407的I2C1接口，**器件地址为0xD0**。

编程读取MPU6050的加速度、陀螺仪、温度传感器的原始数据，在LCD上显示。

