

## 概述

I2C 总线用于连接微控制器及其外围设备，具有接口线少、控制简单、通信速率较高等优点，广泛应用于微控制器、LCD 驱动器、触摸屏、存储器、键盘等接口。

适用范围	
SPC1125 系列	SPC1125, SPC1128
SPC1168 系列	SPC1155, SPC1156, SPC1158, SPC1168, SPD1148, SPD1178, SPD1188, SPD1163, SPM1173
SPC2168 系列	SPC2168, SPC2165, SPC2166, SPC1198
SPC1169 系列	SPC1169, SPD1179, SPD1176
SPC2188 系列	SPC1185, SPC2188

# 目录

<b>1</b>	<b>特性</b> .....	<b>7</b>
<b>2</b>	<b>功能描述</b> .....	<b>8</b>
<b>3</b>	<b>功能实例</b> .....	<b>8</b>
3.1	Bulk 传输模式 .....	8
3.1.1	实例 1: 主机发送与接收 .....	8
3.1.2	实例 2: 从机发送与接收 .....	9
3.2	Poll 传输模式 .....	10
3.2.1	实例 3: 主机发送与接收 .....	10
3.2.2	实例 4: 从机发送与接收 .....	11
3.3	中断传输模式 .....	11
3.3.1	实例 5: 主机发送与接收 .....	11
3.3.2	实例 6: 从机发送与接收 .....	12
<b>4</b>	<b>注意事项</b> .....	<b>14</b>
4.1	上拉电阻的计算.....	14
4.1.1	计算 Rmax.....	14
4.1.2	计算 Rmin.....	15
4.2	数据保持时间配置.....	16
4.3	时间延展 .....	16

## 图片列表

图 2-1: 三种模式数据传输格式 .....	8
图 4-1: I2C 连接示意图 .....	14
图 4-2: I2CSDAHOLD 数据保持时间配置 .....	16
图 4-3: 时钟延展 .....	17

SPIN TROL

## 表格列表

表 1-1: 各产品支持的速度模式 .....	7
表 1-2: 各型号 MCU I2C 支持的标准兼容性 .....	7
表 3-1: 实例 1 代码路径 .....	9
表 3-2: 实例 2 代码路径 .....	10
表 3-3: 实例 3 代码路径 .....	10
表 3-4: 实例 4 代码路径 .....	11
表 3-5: 实例 5 代码路径 .....	12
表 3-6: 实例 6 代码路径 .....	13

SPIN TROL

## 版本历史

版本	日期	作者	状态	变更
A/0	2023-09-01	X.He	Outdated	1. 首次发布。
C/0	2024-08-11	LemengZhou	Released	1. 修改为全系列通用文档。

SPIN  
TROL

## 术语或缩写

术语或缩写	描述
MCU	Micro Control Unit
I2C	Inter-Integrated Circuit

SPIN TROL

# 1 特性

I2C 单元有以下特点：

- 支持多种速度模式，各型号芯片支持的速度模式如下表 1-1 所示：

表 1-1: 各产品支持的速度模式

产品型号	支持的速度模式				
	标准模式 (100kb/s)	快速模式 (400kb/s)	快速模式 Plus (1Mb/s)	高速模式 (2Mb/s 或 3.4Mb/s)	超快速模式 (5Mb/s)
SPC1168 系列, SPC1125 系列, SPC2168 系列	支持	支持	不支持	2Mb/s	不支持
SPC1169 系列	支持	支持	支持	3.4Mb/s	不支持
SPC2188 系列	支持	支持	支持	3.4Mb/s	支持

- 各型号芯片支持的标准兼容性，如下表 1-2 所示：

表 1-2: 各型号 MCU I2C 支持的标准兼容性

标准	MCU 型号
Version 2.1 of the I2C-bus specification	SPC1125 系列, SPC1168 系列, SPC2168 系列, SPC1169 系列
I2C-Bus Specification and User Manual Rev6	SPC2188 系列

- 支持 7 位或者 10 位地址寻址；
- 支持 7 位或者 10 位地址模式下的混合格式（又名组合格式）传输；
- 两个独立的发送和接收 FIFO，每个 FIFO 为 16 字节深度；
- SPC2188、SPC1185 型号芯片支持时钟延展功能；

## 2 功能描述

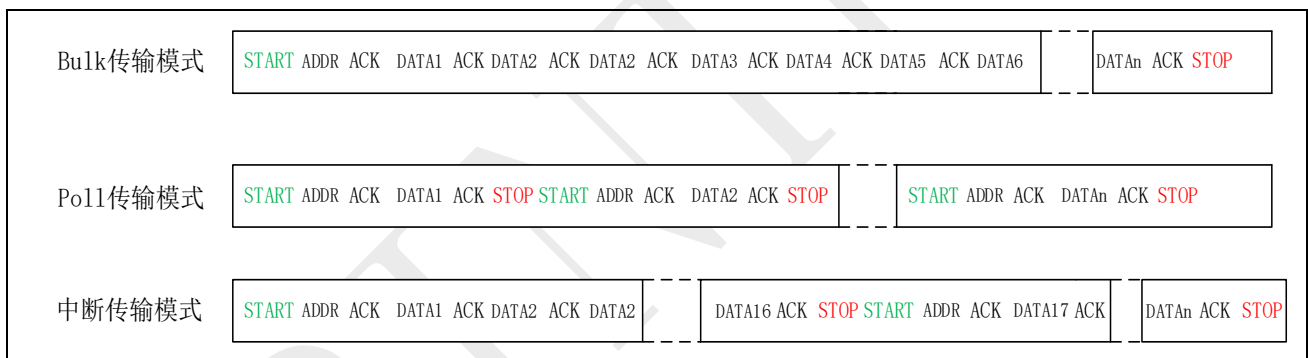
I2C 总线传输具有三种传输模式：

- Bulk 传输模式；
- Poll 传输模式；
- 中断传输模式；

数据传输格式如图 2-1 所示，其中：

- START 信号：时钟为高电平时数据线由高变低；
- STOP 信号：时钟为高电平时数据线有低变高；
- Bulk 传输模式：所有数据一次性传输，只发一次 START 信号、STOP 信号、从设备地址；
- Poll 传输模式：每发一个数据，会发一次 START 信号、STOP 信号、从设备地址；
- 中断传输模式：每发送 FIFO 深度的数据，会发送一次 START 信号、STOP 信号、从设备地址；

图 2-1：三种模式数据传输格式



后续小节通过不同实例，来说明本司各型号 MCU 在各传输模式下的使用方法。

## 3 功能实例

### 3.1 Bulk 传输模式

#### 3.1.1 实例 1：主机发送与接收

##### 3.1.1.1 功能需求

I2C 作为主机，并使用 Bulk 传输模式进行数据的发送和接收。

- 作为 I2C 主机设备；
- 使用 Bulk 传输模式；
- 传输速率为 400kb/s；



### 3.1.1.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 使能探测 STOP 信号中断，通过判断是否进入中断来判断通信是否完成；
- 初始化 I2C 为主机设备，并初始化速度为 400K；
- 使能 I2C 设备；
- 使用 Bulk 模式发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-1。

表 3-1: 实例 1 代码路径

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\ I2C_Master_Bulk_Polling_TxRx

[1] 对于 SDK 提供该实例 Demo，其中 MASTER\_TRANSMIT 为发送接收控制宏，其为 1 时进行数据发送，为 0 时进行数据接收。

## 3.1.2 实例 2: 从机发送与接收

### 3.1.2.1 功能需求

I2C 作为从机，并使用 Bulk 传输模式进行数据的发送和接收。

- 作为 I2C 从机设备；
- 使用 Bulk 传输模式；
- 传输速率为 400kb/s；
- 

### 3.1.2.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 使能探测 STOP 信号中断，通过判断是否进入中断来判断通信是否完成；
- 初始化 I2C 为从机设备，并初始化速度为 400K；
- 设置 I2C 地址模式以及设备地址；
- 当为接收时需要给从机发送读请求；
- 使能 I2C 设备；
- 使用 Bulk 模式发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如

- 表 3-2。

SPIN TROL

**表 3-2: 实例 2 代码路径**

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\ I2C_Slave_Bulk_Polling_TxRx

[1] 对于 SDK 提供该实例 Demo，其中 SLAVE\_TRANSMIT 为发送接收控制宏，其为 1 时进行数据发送，为 0 时进行数据接收。

## 3.2 Poll 传输模式

### 3.2.1 实例 3: 主机发送与接收

#### 3.2.1.1 功能需求

I2C 作为主机，并使用 Poll 传输模式进行数据的发送和接收。

- 作为 I2C 主机设备；
- 使用 Poll 传输模式；
- 传输速率为 400kb/s；
- 

#### 3.2.1.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 使能探测 STOP 信号中断，通过判断是否进入中断来判断通信是否完成；
- 初始化 I2C 为主机设备，并初始化速度为 400K；
- 设置 I2C 地址模式及设备地址；
- 当为接收时需要给从机发送读请求；
- 使能 I2C 设备；
- 使用 Poll 模式发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-3。

**表 3-3: 实例 3 代码路径**

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\ I2C_Slave_Polling_TxRx

[1] 对于 SDK 提供该实例 Demo，其中 MASTER\_TRANSMIT 为发送接收控制宏，其为 1 时进行数据发送，为 0 时进行数据接收。

## 3.2.2 实例 4：从机发送与接收

### 3.2.2.1 功能需求

I2C 作为从机，并使用 Poll 传输模式进行数据的发送和接收。

- 作为 I2C 从机设备；
- 使用 Poll 传输模式；
- 传输速率为 400kb/s；

### 3.2.2.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 使能探测 STOP 信号中断，通过判断是否进入中断来判断通信是否完成；
- 初始化 I2C 为从机设备，并初始化速度为 400K；
- 设置 I2C 地址模式及设备地址；
- 使能 I2C 设备；
- 使用 Poll 模式发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-4。

表 3-4：实例 4 代码路径

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\ I2C_Slave_Polling_TxRx

[1] 对于 SDK 提供该实例 Demo，其中 SLAVE\_TRANSMIT 为发送接收控制宏，其为 1 时进行数据发送，为 0 时进行数据接收。

## 3.3 中断传输模式

### 3.3.1 实例 5：主机发送与接收

#### 3.3.1.1 功能需求

I2C 作为主机，并使用中断传输模式进行数据的发送和接收。

- 作为 I2C 主机设备；
- 使用中断传输模式；
- 传输速率为 400kb/s；

### 3.3.1.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 初始化 I2C 为主机设备，并初始化速度为 400K；
- 设置 I2C 地址模式以及设备地址；
- 当为接收时需要给从机发送读请求；
- 设置发送或者接收 FIFO 的阈值；
- 使能发送与接收中断；
- 使能 I2C 设备；
- 在中断服务函数中根据发送与接收 FIFO 的状态进行发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-5：

表 3-5: 实例 5 代码路径

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\ I2C_Master_INT_TxRx

[1] 对于 SDK 提供该实例 Demo，其中 MASTER\_TRANSMIT 为发送接收控制宏，其为 1 时进行数据发送，为 0 时进行数据接收。

## 3.3.2 实例 6: 从机发送与接收

### 3.3.2.1 功能需求

I2C 作为从机，并使用中断传输模式进行数据的发送和接收。

- 作为 I2C 从机设备；
- 使用中断传输模式；
- 传输速率为 400kb/s；

### 3.3.2.2 功能实现

- 初始化系统时钟，UART 调试口以及初始化 I2C 的 GPIO；
- 初始化 I2C 为从机设备，并初始化速度为 400K；
- 设置 I2C 地址模式以及设备地址；
- 设置发送或者接收 FIFO 的阈值；
- 使能 I2C 设备；
- 使能发送与接收中断
- 在中断服务函数中根据发送与接收 FIFO 的状态进行发送或者接收数据；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-6。

表 3-6: 实例 6 代码路径

MCU 产品型号	代码路径
所有型号	SDK 目录\0_Examples\I2C_Slave_INT_TxRx

[1] 对于 SDK 提供该实例 Demo, 其中 SLAVE\_TRANSMIT 为发送接收控制宏, 其为 1 时进行数据发送, 为 0 时进行数据接收。

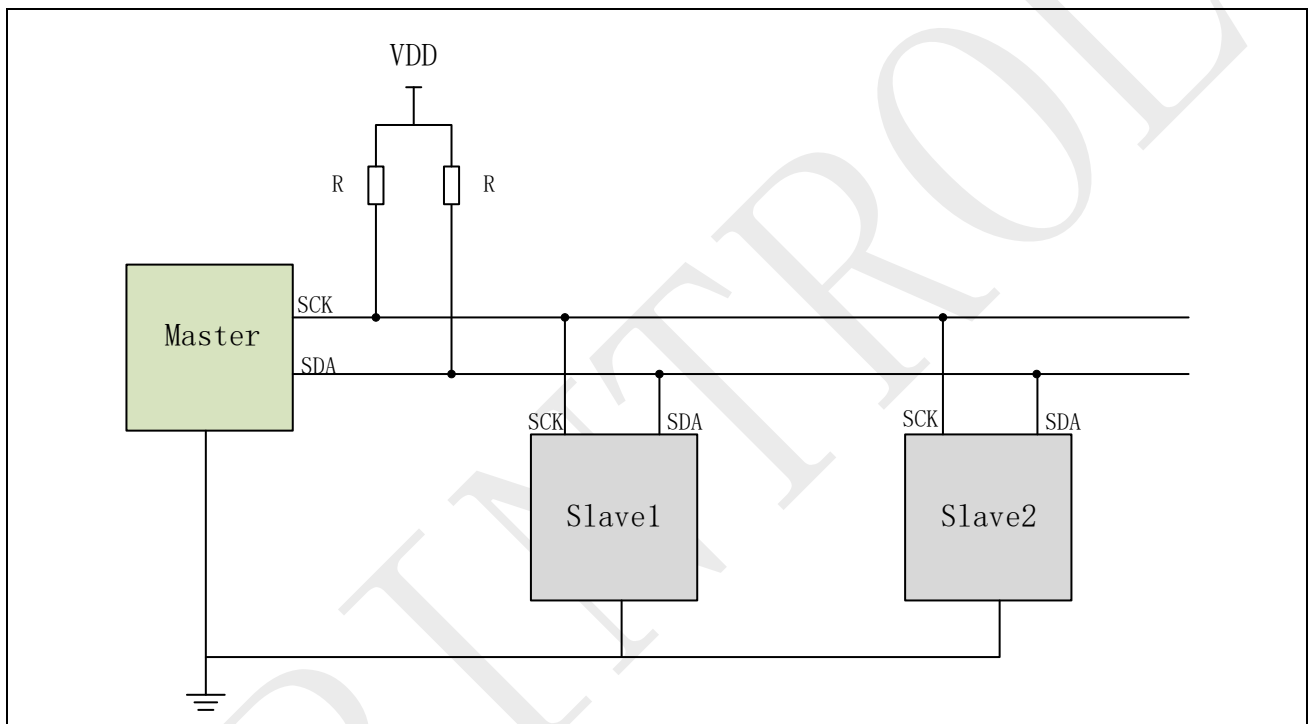
SPIN TROL

## 4 注意事项

### 4.1 上拉电阻的计算

在非超快速模式下，I2C 接口应采用开漏输出方式。此方式需要连接上拉电阻，用以具备输出高电平的能力。另外需要注意的是，I2C 在通信的过程不仅与上拉电阻的选择有关，还与 I2C 的通信的距离有关。I2C 总线适用于短距离通信，为确保主从设备通信的信号质量，两设备之间需要共地用来确保两设备一致的基准电平，否则在通信过程中容易产生毛刺干扰通信数据。连接方式如图 4-1 所示：

图 4-1: I2C 连接示意图



虽然，可以通过减小上拉电阻来提高驱动能力，来解决在实际应用当中时钟波形具有如下非预期问题：

- 时钟的频率低于理论的时钟频率；
- 时钟的上升比较迟缓；

但是，上拉电阻并不是越小越好，产品应根据实际情况，计算出上拉电阻的最小值  $R_{min}$  和最大值  $R_{max}$ 。

#### 4.1.1 计算 $R_{max}$

对于  $R_{max}$  计算方式如下：

1. 确定 I2C 的 IO 特性，以如下参数举例：
  - a)  $V_{DVDD} = 5V$ ；
  - b) I2C 的引脚的达到最小高电平阈值为  $V_{DVDD} - 0.4V$ ；

- c) 达到最大低电平的阈值为 0.5V;
2. 计算达到低电平阈值  $V_t$  所需的时间  $t_1$ :
- a) 某时刻电压的计算表达式可以表示为:

$$V_t = V_{DVDD} \left(1 - e^{-\frac{t}{RC}}\right)$$

其中  $t$  为自上电开始至达到  $V_t$  所需的时间,  $RC$  是时间常数。

- b) 以  $V_{DVDD}=5V$  为例 (以实际测量为准) 计算到达最大低电平阈值时间  $t_1$ :

$$V_{t_1} = 0.5 = V_{DVDD5} \left(1 - e^{-\frac{t_1}{RC}}\right)$$

$$\text{则 } t_1 = 0.0833816 * RC。$$

3. 计算达到最小高电平阈值时间  $t_2$ :

$$V_{t_2} = V_{DVDD} - 0.4 = V_{DVDD} \left(1 - e^{-\frac{t_2}{RC}}\right)$$

$$\text{则 } t_2 = 2.5257286 * RC。$$

4. 计算  $V_{t_1}$  到达  $V_{t_2}$  时间  $T$ :

$$T = t_2 - t_1 = 2.442347 * RC$$

5. 确定 SCL/SDA 上升时间  $t_r$  最大值和预估总线电容  $C_b$ :

- a) 标准模式  $t_r$  时间为 1000ns, 快速模式  $t_r$  时间为 300ns, 快速 plus 模式  $t_r$  时间为 120ns。  
 b) 标准模式  $C_b$  电容为 400pF, 快速模式  $C_b$  电容为 400pF, 快速模式 plus  $C_b$  电容为 550pF。

6. 计算上拉电阻的最大值  $R_{max}$ :

$$R_{max} = \frac{t_r}{2.442347 * C_b}$$

#### 4.1.2 计算 $R_{min}$

对于  $R_{min}$  计算方式如下:

$$R_{min} = \frac{V_{DVDD} - V_{OL}}{I_{OL}}$$

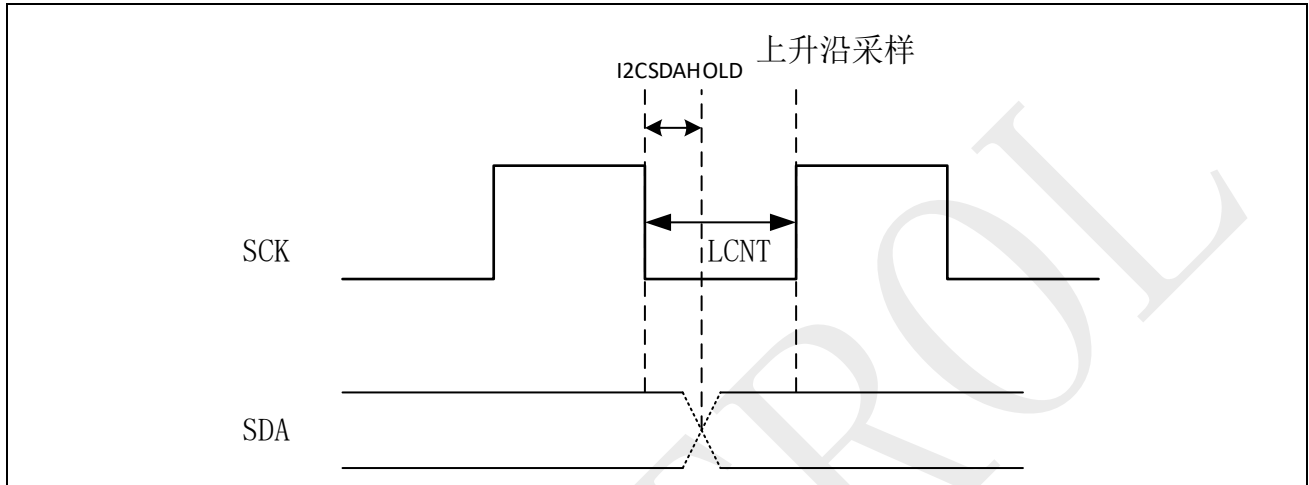
其中  $V_{OL}$  为最大低电平阈值 0.4,  $I_{OL}$  为引脚的额定最小电流 (根据设置引脚的输出强度不同而不同, 当 STRENGTH 为 1 时, 最小电流为 26.1mA)。



## 4.2 数据保持时间配置

在 SPINTROL 的 I2C 控制器中提供了 I2CSDAHOLD 寄存器配置，方便用户根据自己的实际情况灵活调节数据在低电平哪个时刻进行发送数据，但 I2CSDAHOLD 的配置不能超过实际 I2C 时钟的 LCNT 低电平时间，如图 4-2 所示。

图 4-2: I2CSDAHOLD 数据保持时间配置



I2CSDAHOLD 为时钟下降沿开始到数据开始发送的时间，说明如下：

- 当 I2C 设备进行发送时，可以设置 I2CSDAHOLD 的值进行调节数据发送的时间；
- 当 I2C 设备进行接收时，设置 I2CSDAHOLD 时间无效；

对于 I2CSDAHOLD 的设置，IIC 设计指标具有如下规定规定：

- 在标准模式下 I2CSDAHOLD 时间不得低于 50us；
- 快速模式下 I2CSDAHOLD 时间不得低于 0.6us；
- 快速模式 Plus 模式下 I2CSDAHOLD 时间不得低于 0.26us；
- 高速模式下 I2CSDAHOLD 时间不得低于 160ns；
- 超快模式下 I2CSDAHOLD 时间不得低于 50ns；

## 4.3 时间延展

### 适用范围

SPC2188 系列

可以通过 CLKSTRETCH 寄存器，以配置是否使能时钟延展功能。I2C 的时钟延展功能具体表现为，作为从机时，在以下两种状态下会拉低时钟，从而主动终止数据传输。

- 当从机发送数据，且发送 FIFO 为空；
- 当从机接收数据，且接收 FIFO 为满；

且当以上状态不存在时，从机将释放时钟。

如图 4-3 所示，当  $t_1$  时刻从机进行接收数据，但接收 FIFO 已经满不能接收数据时，就会触发时钟延展功能，从机会把时钟拉低，数据信号维持触发前的状态。直到接收 FIFO 不为满时，从机将释放时钟，从而恢复数据传输。

图 4-3: 时钟延展

