

基于Cortex-M3的精密模拟微控制器(ADuCxxx系列)上的I²C兼容接口

作者: Bridget Dowling

简介

本应用笔记介绍如何利用ADI公司基于Cortex-M3的精密模拟微控制器(ADuCxxx系列)实现I²C兼容(内部集成)接口的硬件主机和从机。

该应用笔记还包含展示主机和从机采用I²C接口互相通信的代码段实例。实例包括

- 主机发送和接收
- 从机发送和接收
- 从模式下的DMA传输(发送和接收)
- 主模式下的DMA传输(发送和接收)

请参考 AN-1159 随附代码压缩文件，您可以从 <http://www.analog.com> 下载随附代码。

I²C总线的主要特性如下

- 只需要两条总线线路：串行数据线(SDA)和串行时钟线(SCL)。两条线路均为双向，即主机和从机均可以用作发射器或接收器。
- 一个I²C主机可以与多个从机通信。每个从机都有一个唯一的地址，因此即使在多从机环境中也始终存在一对一的主机/从机关系。
- 仲裁功能允许同一I²C总线上有多个主机。
- 主机和从机的发送与接收速度可高达400 kbps。
- 片内滤波可抑制SDA和SCL线上的50 ns以下尖峰，保护数据完整性。

图1显示了多个设备的典型I²C连接。

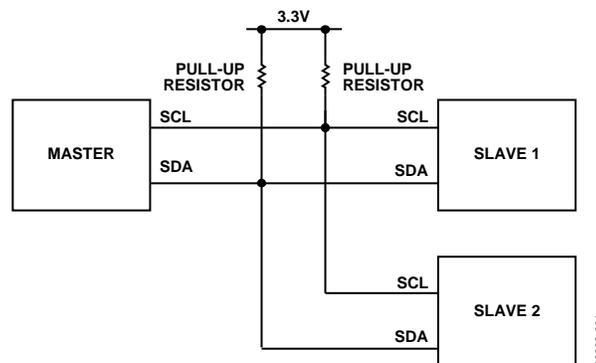


图1. 单主机、多从机I²C框图

目录

简介.....	1	从机接收.....	10
修订历史.....	2	主机接收.....	11
I ² C基础.....	3	从机发送.....	12
典型的I ² C时序图.....	6	DMA模式、主机发送.....	13
在基于Cortex-M3的微转换器上的I ² C方案.....	7	DMA模式、主机接收.....	14
I ² C低电平功能.....	8	DMA模式、从机接收和发送.....	15
主机发送.....	9	配套代码.....	16

修订历史

2012年9月—修订版0：初始版

I²C基本原理

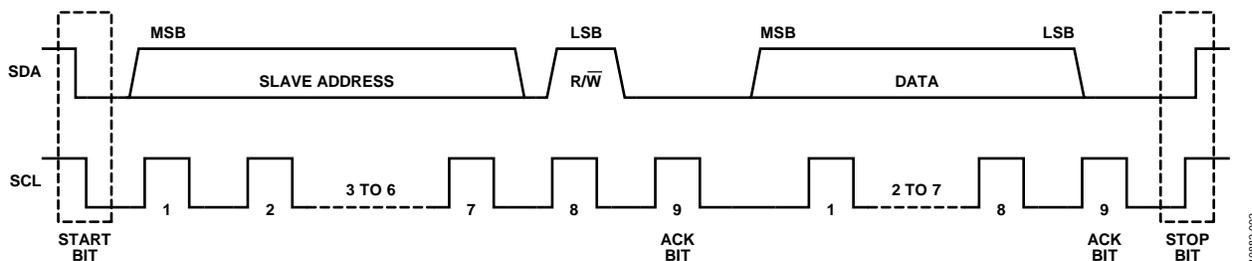


图2. 典型I²C传输序列

I²C接口概述

I²C是最初由Philips半导体(现为NXP半导体)开发的一种双线串行通信系统，允许多个主机和多个从机通过两条线(SCL和SDA)相连。在I²C接口中，至少有一个主机和一个从机。

SCL信号控制主机与从机之间的数据传输。每传输一个数据位，就必须产生一个时钟脉冲。SCL信号始终是从主机传送到从机。不过，如果从机尚未准备好开始下一次传输，则从机能够将此线拉低。这称为时钟延展功能。

SDA信号用于发送或接收数据。SDA输入在SCL高电平周期中必须保持稳定。当SCL为高电平时SDA线路的转换被视为一个启动或者停止条件。图2所示为典型的传输序列。

起始条件

I²C接口的典型数据传输序列从起始条件开始。起始条件是指在SCL线为高电平时，SDA线上发生的高电平至低电平跃迁(见图3)。起始条件的产生始终由主机负责。在SCL线的高电平周期中，SDA线只应在发生起始(和停止)条件时改变。在正常数据传输期间(包括从机寻址)，SDA线上的数据在SCL线的高电平周期中必须保持稳定。

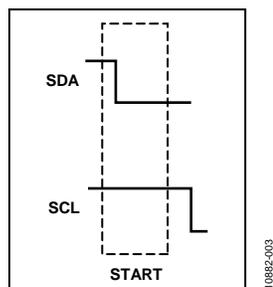


图3. I²C的起始条件

停止条件

数据传输序列由停止条件终止。停止条件是指在SCL为高电平时，SDA线上发生的低电平至高电平跃迁(见图4)。

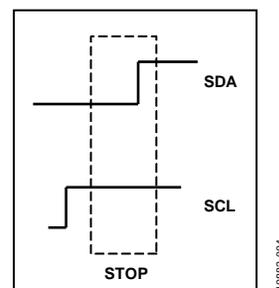


图4. I²C的停止条件

停止条件始终由主机产生。主机一旦认为数据序列已结束，或者接收到来自从机的NACK，就会发送停止条件。接收到停止条件后，从机复位到再次等待从机地址的状态。

I²C接口可以配置为在发生停止条件时产生中断。

从机地址

发生起始条件之后，主机以最高有效位(MSB)优先方式在SDA线上发送一个字节，同时发送8个SCL脉冲。此字节的前7位为7位从机地址。只有当此7位地址与从机的地址(或四个从机地址中的一个)匹配时，从机才会响应主机。第8位(最低有效位LSB)为R/W状态位，决定消息的方向。如果此位为低电平，则主机对选定的从机写入数据。如果此位为1，则主机期待接收来自从机的数据。两种情况下主机都会产生时钟。

如果从机接收到正确的地址，即来自主机的7个MSB与I2CADR0存储器映射寄存器(MMR)的7个MSB匹配，则从机返回一个有效ACK，将SCL线拉低，并设置I2CSSTA中的标志位。

AN-1159

从机通过硬件自动完成I²C从机寻址的所有处理，而主机则负责输出正确的从机地址。

应答(ACK)/不应答(NACK)

如果从机地址与主机发送的地址匹配，从机会自动发送一个应答(ACK)信号，否则会发送一个不应答(NACK)信号。ACK指SDA线在第9个时钟脉冲时为低电平。NACK指SDA线在第9个时钟脉冲时为高电平(见图5)。

在数据传输期间，ACK或NACK始终由接收器产生。请注意，ACK所需的时钟脉冲始终由主机产生。在ACK时钟脉冲期间，发送装置必须释放SDA线(高阻)。为产生有效的ACK，接收器必须将SDA线拉低。

ACK和NACK均通过硬件在各字节接收结束时自动产生。

如果主机从一个从机接收器接收到NACK(从机不响应从机地址或所传输的数据)，主机应产生停止条件，中止传输(参见“数据传输”部分)。

如果从机为发射器，则在从机发送完最后一个字节之后，主机接收器应产生一个不应答(NACK)信号，向从机表明

数据序列已结束。从机一旦接收到NACK，就会释放SDA线，以便主机能够产生停止条件。

从机也可配置为强制一个NACK。

数据传输

在I²C中断服务程序(ISR)中，或者在查询实现中，从机根据主机发送的R/W位的状态决定是否发送或接收。然后，从机在主机发送的每个时钟脉冲上发送或接收一位。主机负责提供9个时钟脉冲(8个用于数据，1个用于ACK)，以便从机与主机之间收发数据。从机每发送或接收一个有效数据字节时，I²C中断位就会置1。

同样要注意，在从机为发射器、主机为接收器的系统中，当从机发送完最后一个字节之后，主机接收器应发送NACK，向从机表明数据序列已结束。从机一旦接收到NACK，就会释放SDA线，以便主机能够产生停止条件。

如果一个主机想要中止数据传输，或者中断总线上另一个主机的数据传输，它可以先发送一个起始条件，再发送一个停止条件。

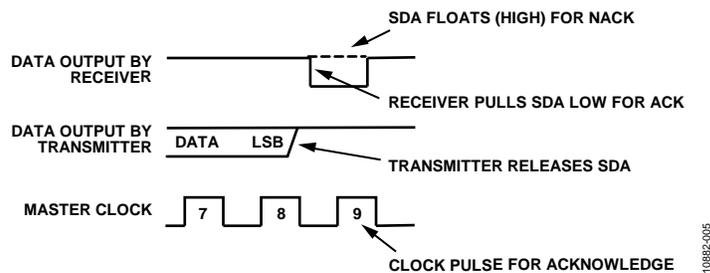


图5. I²C总线上的应答(ACK)和不应答(NACK)

重复起始条件

重复起始条件是指向从机发送第二个起始条件，且第一个和第二个起始条件之间没有发送停止条件。它允许主机通过改变 R/\overline{W} 位逆转传输方向，但不必放弃对总线的控制权。

图6所示为一个传输序列示例。这通常在第一个数据发送至设置读取寄存器地址的器件时使用。当接收到“重复起始条件加从机地址”时，会产生一个中断。使用I2CxSSTA MMR的状态位可以区分是“重复起始条件加从机地址”还是“起始条件加从机地址”。

ADuCxxx系列能在主机模式下使用图6显示的序列直接产生一个I²C重复起始序列。

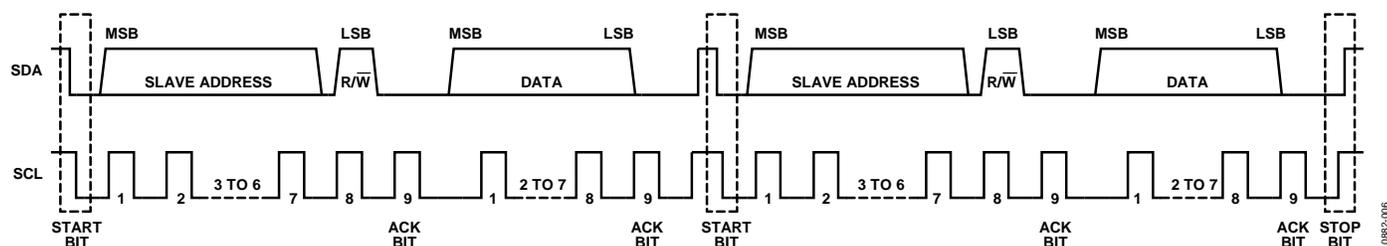


图6. I²C重复起始序列

```
I2cFifoFlush(MASTER, ENABLE);
I2cFifoFlush(MASTER, DISABLE);
NVIC_DisableIRQ(I2CM_IRQn);
I2cMWrCfg(0xA0); //设置为发送从机地址
I2cTx(MASTER, ucStartAddress); //发送从机地址
while ((I2cSta(MASTER)& I2CMSTA_TXFSTA_EMPTY) != I2CMSTA_TXFSTA_EMPTY){} {}
//等待Tx FIFO清空
I2cMRdCfg(0xA0, ucLength, DISABLE);
//发送读取条件, 该功能设置读取位
NVIC_EnableIRQ(I2CM_IRQn);
```

时钟延展

在I²C通信中，主机决定时钟速度。与RS-232不同，I²C总线提供明确的时钟信号，因此主机和从机无需精确同步到预定的波特率。

但在某些情况下，I²C从机无法以主机给出的时钟速度工作，需要放慢速度。这是通过一种称为“时钟延展”的机制来实现的。

如果I²C从机需要降低总线速度，它可以限制时钟速度。另一方面，主机在释放时钟线以使其回到高阻状态后，需要回读时钟信号，直到时钟线确实已变为高电平。

AN-1159

典型I²C时序图

表1和表2显示了基于Cortex-M3的精密模拟微转换器上的I²C总线典型时序。根据I²C总线规范，每条I²C总线线路上的容性负载Cb最大为400 pF。

注意SCL和SDA上的内部上拉电阻应该在软件中禁用。

表1. 快速模式下I²C时序(400 kHz)

参数	描述	最小值	最大值	单位
t_L	时钟低电平脉冲宽度	1300		ns
t_H	时钟高电平脉冲宽度	600		ns
t_{SHD}	起始条件保持时间	600		ns
t_{DSU}	数据建立时间	100		ns
t_{DHD}	数据保持时间	0		ns
t_{RSU}	重复起始建立时间	600		ns
t_{PSU}	停止条件的建立时间	600		ns
t_{BUF}	一个结束条件和起始条件之间的总线空闲时间	1.3		μ s
t_R	时钟和数据的上升时间	$20 + 0.1 C_b$	300	ns
t_F	时钟和数据的下降时间	$20 + 0.1 C_b$	300	ns
t_{SUP}	尖峰抑制脉宽	0	50	ns

表2. 标准模式下I²C时序(100 kHz)

参数	描述	最小值	最大值	单位
t_L	时钟低电平脉冲宽度	4.7		μ s
t_H	时钟高电平脉冲宽度	4.0		μ s
t_{SHD}	起始条件保持时间	4.7		μ s
t_{DSU}	数据建立时间	250		ns
t_{DHD}	数据保持时间	0		μ s
t_{RSU}	重复起始建立时间	4.0		μ s
t_{PSU}	停止条件的建立时间	4.0		μ s
t_{BUF}	一个结束条件和起始条件之间的总线空闲时间	4.7		μ s
t_R	时钟和数据的上升时间		1	μ s
t_F	时钟和数据的下降时间		300	ns

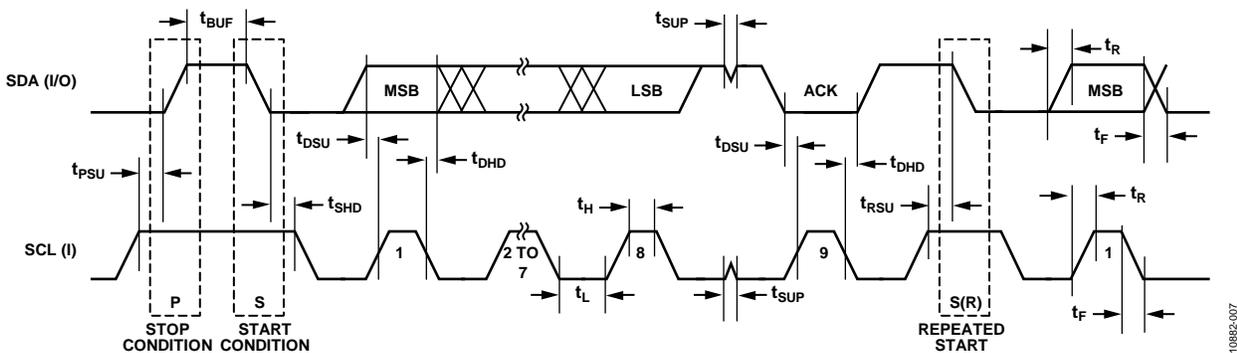


图7. I²C兼容接口时序

在基于CORTEX-M3的微转换器上的I²C方案

基于Cortex-M3的ADuCxxx系列产品包含了全硬件主机和从机I²C端口。

I²C外设接口一共包含19个寄存器，9个用于主机，9个用于从机，一个供主机和从机共用。表3总结了上述情况。

表3. I²C寄存器

寄存器	主机	从机	共享
接收	I2CMRX	I2CSRX	I2CFSTA
发射	I2CMTX	I2CSTX	I2CFSTA
状态	I2CMSTA	I2CSSTA	
控制	I2CMCON	I2CSCON	
地址/ID	I2CADR0/ I2CADR1	I2CID0/I2CID1/ I2CID2/I2CID3	
时钟	I2CDIV		
其它	I2CMRXCNT I2XMCRXCNT	I2CALT	

设备用户手册中描述了所有寄存器和数据位。

应用笔记中更加详细地解释了部分具体功能。

配置通信速度

I2CDIV一个16位寄存器，包含两个8位值：HIGH和LOW。寄存器中的值设置了I²C总线的速度。根据下列公式进行设置：

$$f_{I2CSCL} = f_{PERIPH} / (LOW + HIGH + 3)$$

其中：

f_{PERIPH} = 是I²C外设时钟。

HIGH = I2CDIV[15:8]。I²C总线时钟的高电平时长由以下公式决定：(HIGH + 2) ÷ I²C外设时钟。

LOW = I2CDIV[7:0]。I²C总线时钟的低电平时长由以下公式决定：(LOW + 1) ÷ I²C外设时钟。

因此，在100 kHz的情况下，I²C外设时钟为16 MHz、LOW = 0x4F、HIGH = 0x4E；400 kHz时，LOW = 0x13、HIGH = 0x12。

ADuCM360规格

ADuCM360上的I²C外设时钟为

$$f_{PERIPH} = f_{UCLK} \div (CLKSYS DIV \times I2CCLK)$$

其中：

UCLK为系统时钟：16 MHz。

CLKSYS DIV为1或2，取决于CLKSYS DIV[0]位的设置。

I2CCD是时钟分频值，由CLKCON1[8:6]位设置一个1到7之间的值。

I²C外设时钟默认为禁用，应该首先在CLKDIS[2]中启用。这是为了优化元件的功耗。更多信息请参考AN-1111应用笔记使用ADuCM360/ADuCM361时将功耗最小化的选项。

下列流程图假设I²C外设时钟已经被启用和配置。

FIFO的使用

根据I²C功能，I²C硬件接口包括四个2字节FIFO。

- 主机接收
- 主机发送
- 从机接收
- 从机发送

以上每个都包含一个移位寄存器将字节的最后一位保持为接收/发送。

发送FIFO

要发送数据，必须加载I2CSTX/I2CMTX寄存器。

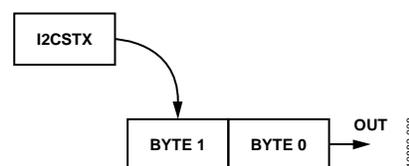


图8. 发送FIFO

写入一个字节到发送寄存器相当于写入FIFO的字节1(见图8)。

- 如果字节0为空，字节1中的字节将被自动推入字节0。该操作对用户是透明的。I2CFSTA寄存器表示FIFO中有一个字节。
- 如果字节0已满，该字节存于字节1中。对Tx再次写入将覆盖字节1中的内容。

字节0中的内容转入移位寄存器(OUT)也对用户透明，但是要求从机端有一个有效的通信。

FIFO在其未满载时将TXREQ位设为(I2CMSTA[2]/I2CSSTA[2])。这些位在使能时会生成中断。

I2CFSTA寄存器中的发送清空位置1将清空FIFO。

接收FIFO

接收数据时，数据到达字节0。

- 如果字节1为空，字节0中的数据将自动移入字节1。
- 如果字节1已满，该数据将留在字节0中，直到读取I2CSRX(相当于读取字节1)。
- 如果在FIFO已满载时有其它数据到达，从机将针对该数据发送一个NACK，并且将I2CSSTA[4]置1。(或者将主机I2CMSTA[9]置1)

FIFO在其未满载时将RXREQ位设为(I2CMSTA[3]/I2CSSTA[3])。这些位在使能时会生成中断。

AN-1159

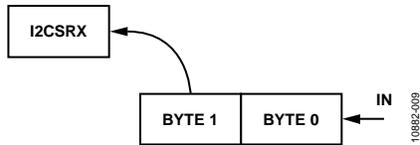


图9. 接收FIFO

DMA通道的使用

DMA控制器中的四个通道分配至I²C接口：

- 主机接收
- 主机发送
- 从机接收
- 从机发送

如果发送时发送FIFO中有空间或者接收时接收FIFO中有一个字节，I²C外设将产生一个DMA请求。

表4. I²C低层次功能

主机配置

<code>int I2cMCfg(int iDMACfg, int iIntSources, int iConfig);</code>	配置I ² C主机
<code>int I2cBaud(int iHighPeriod, int iLowPeriod);</code>	配置I ² C波特率
<code>int I2cMWrCfɡ(unsignd int uiDevAddr);</code>	配置从机地址
<code>int I2cMRdCfɡ(unsignd int uiDevAddr, int iNumBytes, int iExt);</code>	配置从机地址、读取字节数目
<code>int I2cMRdCnt(void);</code>	读取主机接收字节的计数器

从机配置

<code>int I2cSCfg(int iDMACfg, int iIntSources, int iConfig);</code>	配置I ² C从机
<code>int I2cSIDCfɡ(int iSlaveID0, int iSlaveID1, int iSlaveID2, int iSlaveID3);</code>	配置从机地址
<code>int I2cSGCallCfɡ(int iHWGCallAddr);</code>	设置硬件广播

常见功能

<code>int I2cRx(int iMode);</code>	读取从机或主机的Rx寄存器
<code>int I2cTx(int iMode, int iTx);</code>	写入从机或主机的Tx寄存器
<code>int I2cStr(int iMode, int iStretch);</code>	配置时钟延展
<code>int I2cFifoFlush(int iMode, int iFlush);</code>	清除从机或主机的Tx FIFO
<code>int I2cSta(int iMode);</code>	读取从机或主机的状态

当所有存储器中的字节都已经传输至发送FIFO或者分配给DMA通道的存储器已满时，DMA传输结束，产生中断。

I²C状态位TXREQ和RXREQ在设置为DMA模式时不产生中断；然而，显示传输结束的TCOMP位仍然能产生一个中断，在DMA模式中使用。

I²C低层次功能

为了简化代码开发，I2cLib中提供了一套低层次功能。表4列出了这些功能。CD/DVD文档文件夹中对这些功能进行了更加具体的解释。

每种情形的流程图都是基于这些低层次功能。

主机发送

在发送字节前，必须首先将数据载入发送FIFO。I2CADR0寄存器中必须指定从机的地址。对于数据写入，地址寄存器中的写操作(\overline{W})位必须置0。写入I2CADR0寄存器会自动产生一个起始条件。

当发送FIFO为空或未满，且I2CMSTA的第2位字节置为1时，在发送字节的第一个时钟产生一个I²C中断，表示主机刚刚发送了一个字节。因此，用户可以向FIFO添加一个字节。可以在I2CMSTA[1:0]或I2CFSTA寄存器中查询FIFO的状态。

如果启动传输时FIFO中只有一个字节，则第一个I²C中断发生在所发送地址的第一个时钟脉冲上。如果FIFO中有两个字节，则中断发生在所发送的第一个字节的第一个时钟脉冲上。

如果发送FIFO未保持满状态，将在每个时钟边缘产生发送

中断。在最后一个字节的最后一位被发出后，TXUR位被设为1。如果在TXUR位被置为1的 $\frac{1}{2}$ 个时钟周期内没有数据置于FIFO内，即在从机确认位期间时，主机将自动结束传输。

在最后一个字节发送后的5.1 μ s时将自动产生终止条件。

I2CMCON MMR (I2CMCON[5])中的TXREQ中断使能位需要被清零，否则将持续产生发送中断，直到产生终止条件。

如果使能TCOMP位(I2CMCON[8])，在终止条件产生时，也将产生一个中断。这使得I²C外设能够安全关闭，例如在进入低功耗模式之前。

图10所示为主机响应从机数据请求的示例。

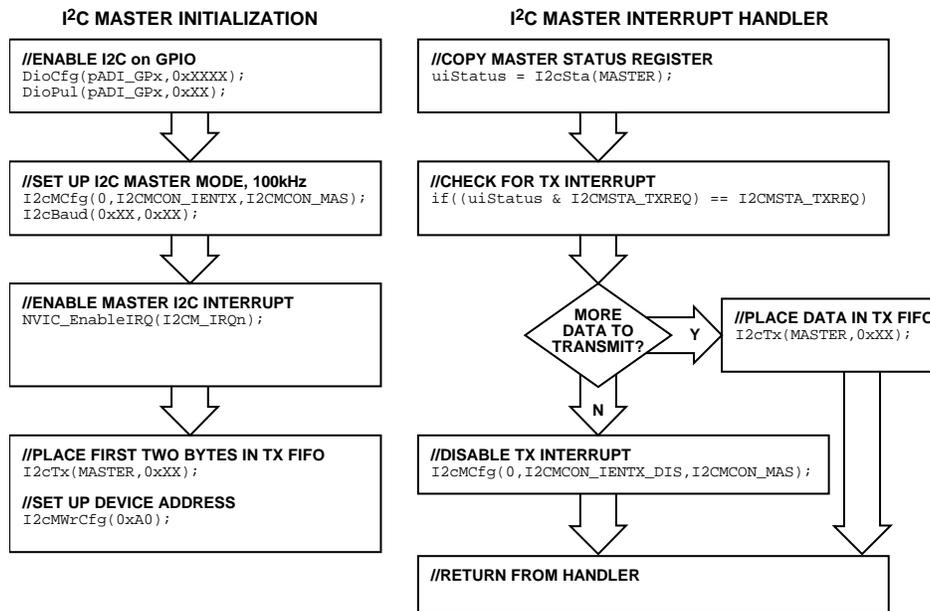


图10. 主机发送流程图

10882-010

AN-1159

从机接收

当I²C从机收到数据时，将在收到每个字节的第八个时钟后产生一个中断。如果FIFO在收到第三个字节收到之前未被读取，RXOF、I2CSSTA[4]位被置为1，表示接收FIFO溢出。此时可以读取FIFO。如果在第九个时钟的上升沿之前FIFO未被读取，从机接口将自动发送一个NACK。

要从FIFO读取数据，应使用I2CRX寄存器。RXREQ、I2CSSTA[3]表示从机收到了数据。只有读取I2CSRX才能使

此位清零。如果将I²C中断使能，将连续产生中断，且将RXREQ设为1。

发送完最后一个数据之后，主机自动发送一个停止条件。从机检测到一个终止条件，将I2CSSTA[10]设为1。如果IENSTOP、I2CSCON[8]设为1，该位可以产生一个中断。

图11显示了从机接收主机字节的流程图。图12显示了当状态位被设为1且当中断发生时的状态。

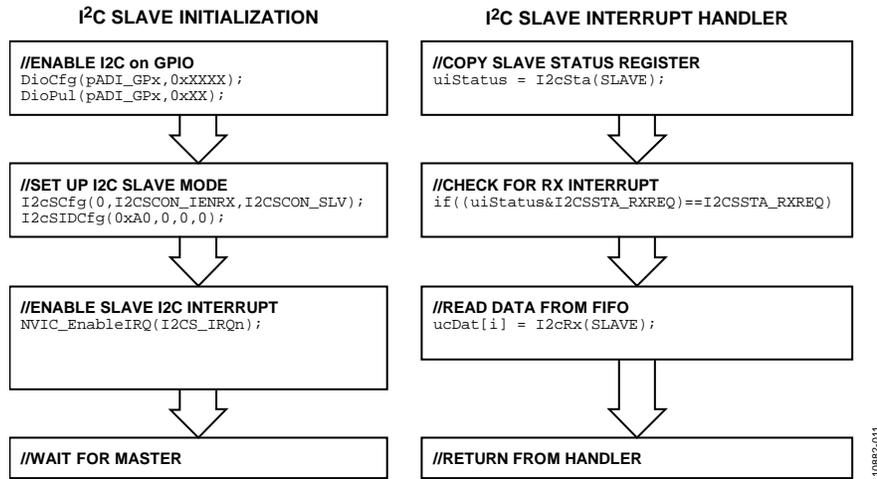


图11. 从机接收流程图

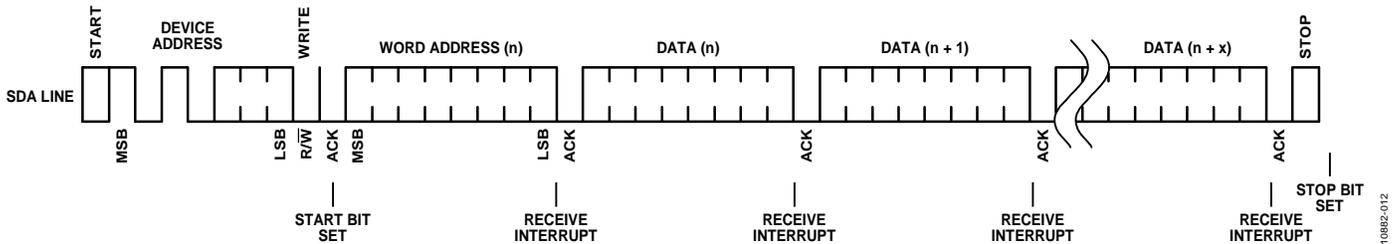


图12. 从机接收示例

主机接收

在主机模式下，要从从机读取数据，可以使用类似的方法。首先，通过I2CMRXCNT寄存器配置要读取的字节数。其值表示要从从机读取的字节数加1。这是一个8位寄存器，因此一次可以接收256个字节。对于更大的传输，该寄存器还提供一个扩展选项。I2CMCRXCNT反应了当前主机接收到的字节数目。

为了开始接收数据，I2CADR0寄存器中的读操作(R)位应置1。这将启动传输，并利用I2CADR0寄存器所设置的地址和R/W位产生起始条件。接收到每个字节后(第9个时钟脉冲ACK或NACK之后)，会产生一个中断。RXREQ、

I2CMSTA[3]置1，表示刚刚接收到一个字节。只有读取I2CMRX才能使此位清零。

当主机不需要接收更多数据时，它针对最后接收到的字节自动产生NACK。这相当于通知从机停止传输字节，以便主机能够产生停止条件。

如果没有及时读取所接收到的数据，并且FIFO已满，则主机将针对所接收到的额外数据提供一个NACK。

图13显示了主机从从机接收字节的流程图。

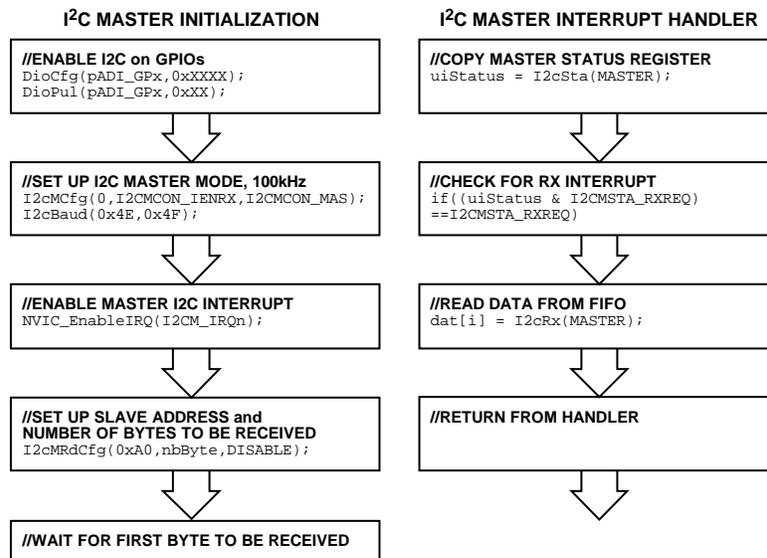


图13. 主机接收流程图

10882-013

AN-1159

从机发送

每次请求发送数据时，从机就会产生一个中断，第一个中断发生于对地址作出应答(ACK)之后，即在发送FIFO的字节0时。数据需要预载到从机发送FIFO中，否则主机的第一个读取请求将产生NACK。如果FIFO预载有两组数据，则在地址作出应答(ACK)后产生一个中断，然后在对所发送的每个字节作出应答(ACK)后产生一个中断。如果FIFO仅预载一组数据，则在地址作出应答(ACK)后产生两个中断，发送第一个数据后FIFO清空。

发送完一个字节后，只要主机继续请求数据，就会产生一个中断。

当发送FIFO中有空间或者每次有一个字节发送至主机时，TXREQ、I2CSSTA[2]置1。当没有数据发送时，发送中断可被禁用并且在检测到终止条件时(I2CSSTA[10])可以重新使能。如果IENSTOP、I2CSCON[8]设为1，终止检测可以产生一个中断。

图14显示了从机对主机的数据请求作出回应的实例。图15显示了当状态位被设为1且当中断发生时的状态。

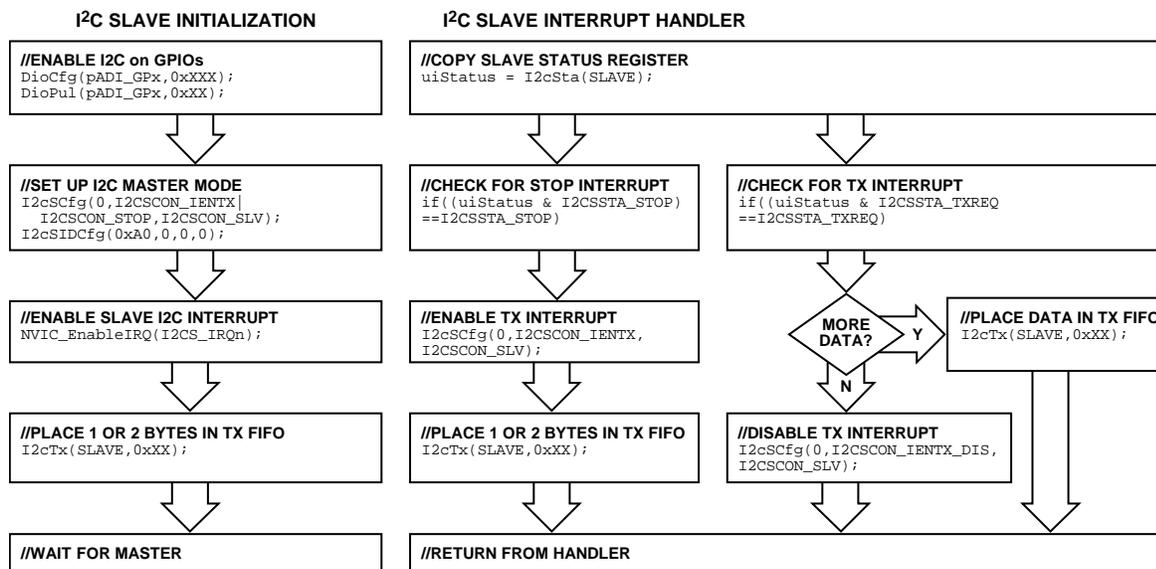


图14. 从机发送流程图

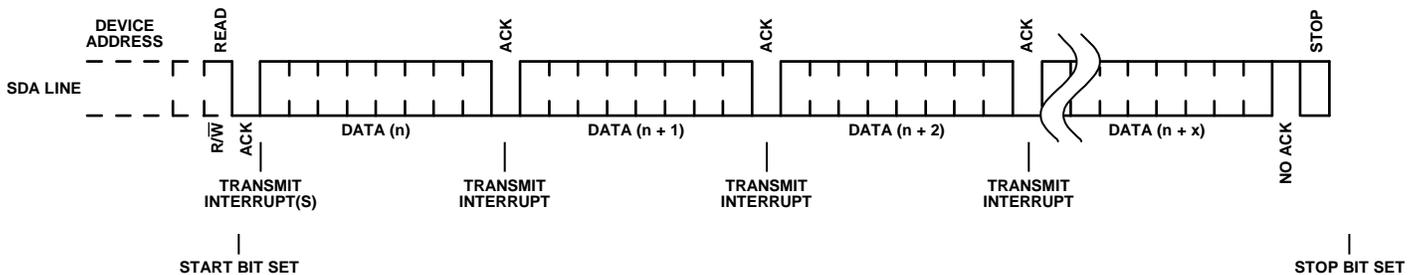


图15. 从机发送示例

DMA模式、主机发送

带有从机地址的主机传输将初始化I²C传输。当通过设置I2CMCON MMR中的TXDMA来配置DMA发送模式时，在I²C外设完全配置时传输将开始，即当DMA控制器进行配置和使能时，开始传输波特率、从机地址和DMA发送请求。I²C主机发送DMA通道应该在开始传输之前的任何时候在NVIC中使能。

当所有字节都从存储器中转入FIFO时，DMA传输完成；这表示FIFO中存储的最后两个字节和倒数第三个字节被传

输，如图17所示。之后在DMA控制器中的DMA通道将被自动禁用。然而，当FIFO未滿时，I²C外设将发送一个DMA请求至DMA控制器并产生一个中断。因此，在DMA控制器中应该屏蔽I²C请求以避免多次中断(DMARMASKSET)。

DMA控制器需要重新配置/重新使能来开始新的传输。

图16显示了I²C主机进行DMA传输的流程图。

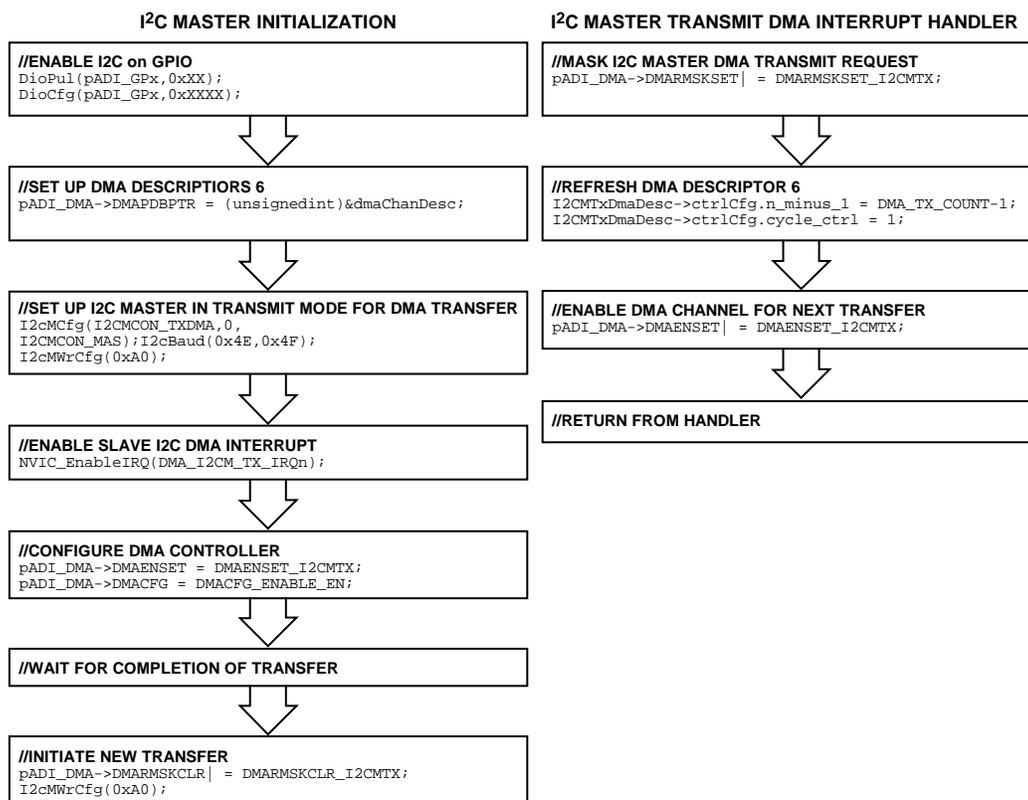


图16. 主机进行DMA传输流程图

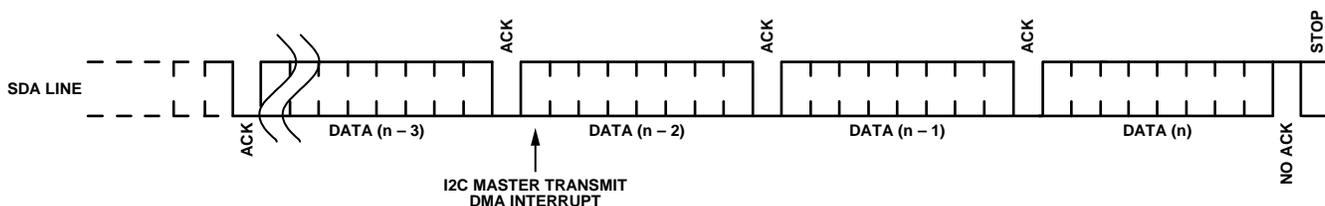


图17. 主机发送DMA中断

AN-1159

DMA模式、主机接收

可以在I2CMCON MMR中对RXDMA进行设置来使能I²C外设中的DMA传输。当DMA控制器被配置和使能且I²C外设被完全配置时，传输将开始，即开始传输波特率、从机地址和DMA发送请求。I²C主机发送DMA通道应该在开始传输之前的任何时候在NVIC中使能。

当DMA控制器预期的所有字节都被接收时，DMA传输完成。I²C主机I2CMRXCNT MMR应该采用同样数目的字节进行配置。

当DMA传输完成时，DMA控制器中的相应通道将自动被禁用。DMA控制器需要重新配置和重新使能来开始新的传输。新传输的初始化可以通过在I2CADR0 MMR中再次写入从地址来完成。

图18显示了主机从从机接收数据的DMA传输流程图。

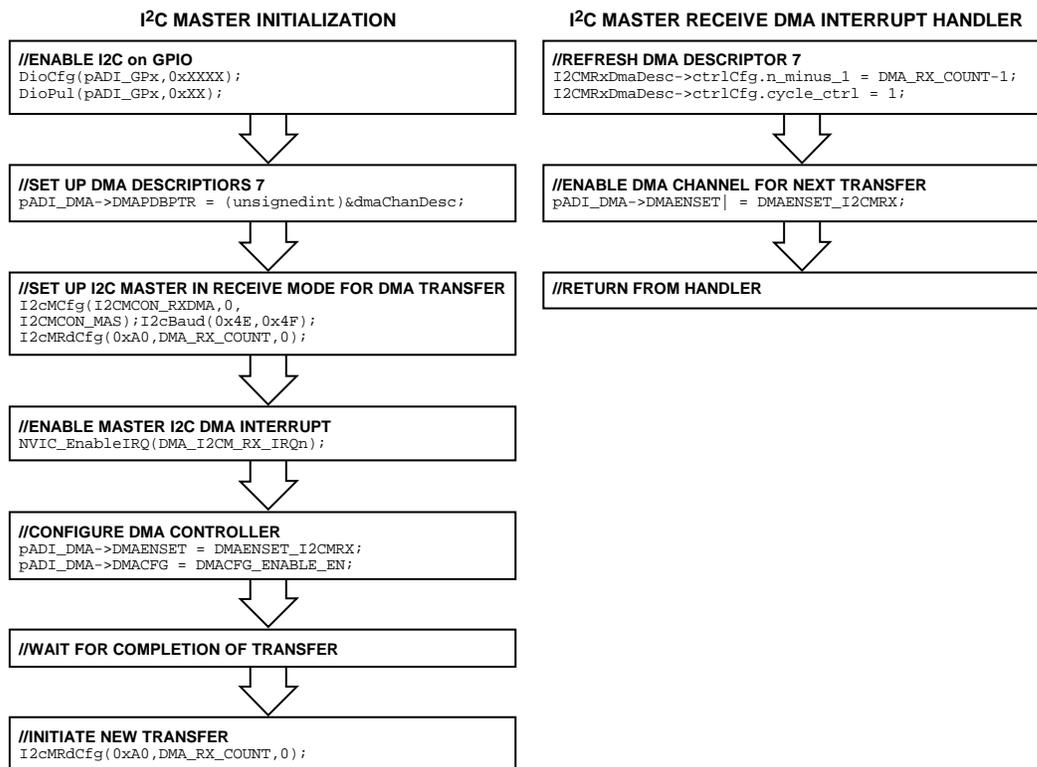


图18. 主机接收DMA传输流程图

10882-018

DMA模式、从机接收和发送

可以在I2CSCON MMR中对TXDMA和/或RXDMA进行设置来使能I²C外设中的DMA传输。I²C外设应该在从机模式中被配置并且应该禁用I²C中断。NVIC和DMA控制器需要进行配置。DMA传输的初始化可以通过接收正确的I²C地址来完成。只有数据字节会被传输入存储器。当DMA传输完成时，DMA控制器中的相应通道将自动被禁用。DMA控

制器需要重新配置来开始新的传输。图19显示了从机发送和接收的DMA传输流程图。

请注意与主机发送类似，在传输第三个至最后一个字节后，发送模式下的DMA传输将结束。

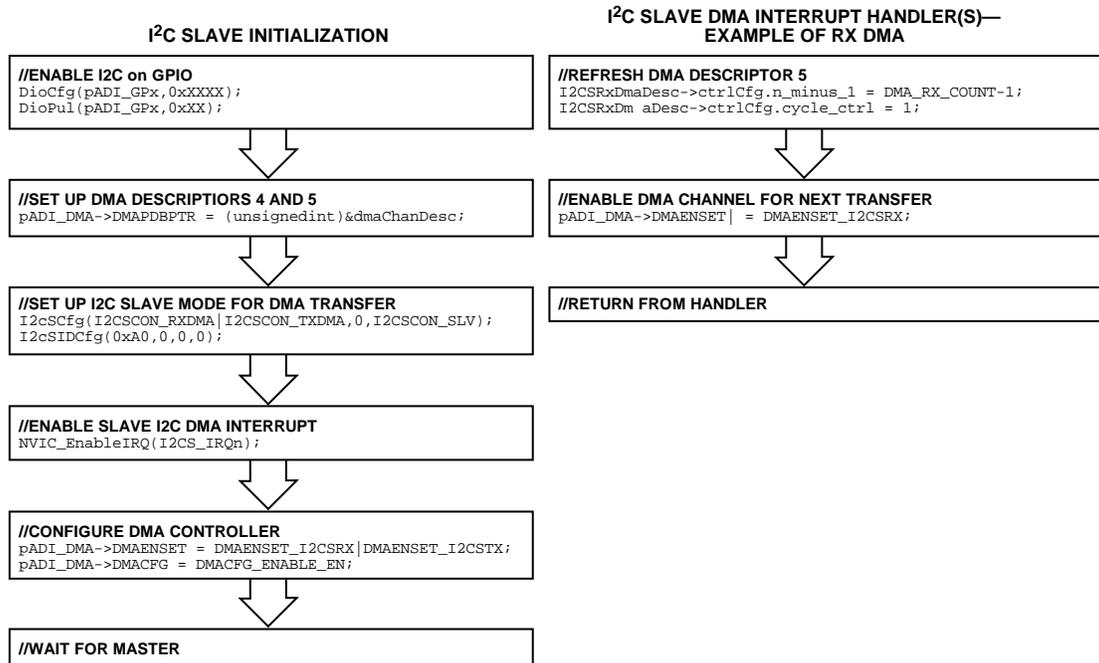


图19.从机DMA传输流程图

AN-1159

配套代码

表5中提供了配套代码的列表和描述。

表5. 配套代码

模式/流程图	代码段实例/工具
主机发送	I2Cmaster.c
从机接收	I2Cslave.c
主机接收	I2Cmaster.c
从机发送	I2Cslave.c
DMA模式、主机发送	I2CmasterDMA.c
DMA模式、主机接收	I2CmasterDMA.c
DMA模式、从机接收和发送	I2CslaveDMA.c

I²C指最初由Philips Semiconductors(现为NXP Semiconductors)开发的一种通信协议。