

文档名: LKS08X I2C 模块采用 DMA 传输使用技巧

作者: 邓廷

编号:

日期: 2019.03.05

内容: LKS08X I2C 模块采用 DMA 传输使用技巧

LKS08X 系列芯片, 集成了一组 I2C 模块。

- I2C 网络, 传输速率需是提前协商且遵循同一波特率, 各从设备地址也需提前确定。
- 只有在 DMA 方式下, I2C 从设备的硬件地址比较才有意义。此时, 硬件地址比较中断可开启也可屏蔽, 看用户自己需求。若开启中断, 地址匹配则会产生中断, 软件可中断处理程序里面处准备好发送数据, 然后继续本次传输; 若关闭中断, 地址匹配后直接开始数据的传输。

#### **I2C 模块主模式下 DMA 发送, 基本传输过程:**

- 配置 I2C 模块时钟 (SYS\_CLK\_FEN&SYS\_CLK\_DIV0), 从模块的硬件地址 (I2C\_ADDR)、读写控制 (I2C\_SCR[2]), 传输控制 (SCR[2]), I2C 相关中断 (I2C\_CFG&I2C\_BCR), DMA 传输使能 (I2C\_BCR[5]), 数据大小 (I2C\_BCR[3:0]) 以及 DMA 寄存器。
- 触发 START (I2C\_MSCR[0]), 开始传输地址+写。若返回 ACK, 表明此时从设备响应本次传输; 若返回 NACK, 表明无对应地址的从设备/从设备没有准备好, 无法完成本次传输。
- 响应本次传输时, 开始发送数据。每发送一个字节, 等待从设备反馈。若是 ACK, 表明可以继续传输; 若是 NACK, 表明从无法继续接收。此时, 将产生 NACK 中断, 根据 SCR[0] 可以判断本批次数据传输是否完成, 同时也可以检查 DMA 寄存器的值判断。
- 触发 STOP, 完成本次传输。无论是否响应本次传输, 主设备均要发出 STOP。

主模式下 DMA 发送。触发 START, 无论从设备的状态如何, 硬件都将预取第一个字节。若从设备无法完成本次传输, 那么主设备需停止本次传输, 同时关闭对应 DMA 通道并重新开启, 其它 DMA 配置也需重新配置。

主模式下 DMA 发送。DMA 必定先把最后一个字节发给主设备, 然后主设备发送给从设备。那么, 传输结束的标志是 I2C 传输完毕。因此, I2C 的 STOP 中断可作为本次传输的结束标志。

#### **I2C 模块主模式下 DMA 接收, 基本传输过程:**

- 配置 I2C 模块时钟 (SYS\_CLK\_FEN&SYS\_CLK\_DIV0), 从模块的硬件地址 (I2C\_ADDR)、读写控制 (I2C\_SCR[2]), 传输控制 (SCR[2]), I2C 相关中断 (I2C\_CFG&I2C\_BCR), DMA 传输使能 (I2C\_BCR[5]), 反馈控制 (SCR[4]), 数据大小 (I2C\_BCR[3:0]) 以及 DMA 寄存器。
- 触发 START (I2C\_MSCR[0]), 开始传输地址+读。若返回 ACK, 表明此时从设备响应本次传输; 若返回 NACK, 表明无对应地址的从设备/从设备没有准备好, 无法完成本次传输。
- 响应本次传输时, 开始接收数据。每接收一个字节, 主设备将根据 SCR[4] 的设置 (此时都应是 ACK), 反馈给从设备。完全接收完毕后。产生完成中断。
- 触发 STOP, 完成本次传输。无论是否响应本次传输, 主设备均要发出 STOP。

主模式下 DMA 接收。触发 START, 若从设备无响应。主设备不会发送 DMA 预取。后续, 主

设备可再次触发 **START**，无需重置 **DMA** 寄存器。

主模式下 **DMA** 接收。主设备接收从设备的数据，然后 **DMA** 才开始搬移数据。那么，传输结束的标志是 **DMA** 传输完毕。因此，**DMA** 中断可作为本次传输的结束标志。因为，**DMA** 传输一个字节耗时很短，且跟随在 **I2C** 的 **STOP** 中断后面。方便软件统一处理，可以用 **I2C** 的 **STOP** 中断做完成标志，但建议在 **I2C** 的 **STOP** 中断函数中，检测 **DMA** 的完成标志位。

#### **I2C 模块从模式下 DMA 发送，基本传输过程：**

- 配置 **I2C** 模块时钟 (**SYS\_CLK\_FEN&SYS\_CLK\_DIV0**)，从模块的硬件地址 (**I2C\_ADDR**)、开启硬件地址比较 (**I2C\_ADDR[7]**)，**I2C** 相关中断 (**I2C\_CFG&I2C\_BCR**，特别是硬件地址比较中断需开启)。
- 触发 **START** (**I2C\_MSCR[0]**)，开始传输地址+读。若从设备地址匹配上，产生中断。从设备若没准备好数据，配置 **SCR[4]** 为 0，返回 **NACK**。若从设备准备好要发送的数据，配置 **DMA**，**BCR[4]** 为 1 (软件协助硬件预取第一发送的字节)，配置好 **SCR[4]** 为 1 (接收主设备请求)，传输方向 **SCR[2]** 为 1 (发送)。开始传输
- 发送完毕本批次数据，产生完成中断。
- 接收 **STOP**，完成本次传输。无论是否响应本次传输，主设备均要发出 **STOP**。

从模式下 **DMA** 发送。一般，从设备准备好接收 **START** 即可。硬件地址匹配成功后，根据主设备的需求，决定从设备是接收还是发送。若是发送，从设备也需要预取，此时可以通过软件协助硬件完成预取第一个发送的数据。若从设备暂时无法实现传输，返回 **NACK** 结束本次传输。

从模式下 **DMA** 发送。**DMA** 必定先把最后一个字节发给从设备，然后从设备发送给主设备。那么，传输结束的标志是 **I2C** 传输完毕。因此，**I2C** 的 **STOP** 中断可作为本次传输的结束标志。

#### **I2C 模块从模式下 DMA 接收，基本传输过程：**

- 配置 **I2C** 模块时钟 (**SYS\_CLK\_FEN&SYS\_CLK\_DIV0**)，从模块的硬件地址 (**I2C\_ADDR**)、开启硬件地址比较 (**I2C\_ADDR[7]**)，**I2C** 相关中断 (**I2C\_CFG&I2C\_BCR**，特别是硬件地址比较中断需开启)。
- 触发 **START** (**I2C\_MSCR[0]**)，开始传输地址+写。若从设备地址匹配上，产生中断。从设备若没准备好接收，配置 **SCR[4]** 为 0，返回 **NACK**。若从设备准备好，配置 **DMA**，配置好 **SCR[4]** 为 1 (接收主设备请求)，传输方向 **SCR[2]** 为 0 (接收)。开始传输
- 接收完毕本批次数据，产生完成中断 (**SCR[0]** 会被置 1)。
- 接收 **STOP**，完成本次传输。无论是否响应本次传输，主设备均要发出 **STOP**。

从模式下 **DMA** 接收。一般，从设备准备好接收 **START** 即可。硬件地址匹配成功后，根据主设备的需求，决定从设备是接收还是发送。若是接收，配置好 **DMA** 等即可。若从设备暂时无法实现传输，返回 **NACK** 结束本次传输。

从模式下 **DMA** 接收。从设备接收主设备的数据，然后 **DMA** 才开始搬移数据。那么，传输结束的标志是 **DMA** 传输完毕。因此，**DMA** 中断可作为本次传输的结束标志。因为，**DMA** 传输一个字节耗时很短，且跟随在 **I2C** 的 **STOP** 中断后面。方便软件统一处理，可以用 **I2C** 的 **STOP** 中断做完成标志，但建议在 **I2C** 的 **STOP** 中断函数中，检测 **DMA** 的完成标志位。