



TWI 典型的传输过程图

图中所示的 TWI 传输过程为：

1. TWI 传输的第一步是发送 **START**。通过往 **TWCR** 寄存器写入特定值，指示 TWI 硬件发送 **START** 信号。写入的值将在随后详细说明。在写入的值中要置位 **TWINT**，这非常重要，往 **TWINT** 位写“1”会清零该位。**TWCR** 寄存器的 **TWINT** 置位期间 TWI 不会启动任何操作。一旦软件清零 **TWINT** 位，TWI 模块立即启动 **START** 信号的传送。
2. 当 **START** 状态发送完毕，**TWCR** 的 **TWINT** 标志位会被置位，**TWSR** 更新为新的状态码，表示 **START** 信号成功发送。
3. 应用程序查看 **TWSR** 的值，确定 **START** 状态已经成功发送。如果 **TWSR** 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，程序将 **SLA+W** 的值载入到 **TWDR** 寄存器中。**TWDR** 寄存器可同时在地址和数据中使用。随后软件往 **TWCR** 寄存器写入特定值，指示 TWI 硬件发送 **TWDR** 中的 **SLA+W** 的值。写入的值将在随后详细说明。在写入的值中要置位 **TWINT**，来清零 **TWINT** 标志位。**TWCR** 寄存器的 **TWINT** 置位期间 TWI 不会启动任何操作。一旦软件清零 **TWINT** 位，TWI 模块立即启动地址包的传送。
4. 当地址包发送完毕后，**TWCR** 的 **TWINT** 标志位会被置位，**TWSR** 更新为新的状态码，表示地址包成功发送。状态码同样会反映从机是否响应该地址包。
5. 应用程序查看 **TWSR** 的值，确定地址包已成功发送，收到的 **ACK** 为期望值。如果 **TWSR** 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，程序将 **Data** 的值载入到 **TWDR** 寄存器中。随后软件往 **TWCR** 寄存器写入特定值，指示 TWI 硬件发送 **TWDR** 中的 **Data** 的值。写入的值将在随后详细说明。在写入的值中要置位 **TWINT**，来清零 **TWINT** 标志位。**TWCR** 寄存器的 **TWINT** 置位期间 TWI 不会启动任何操作。一旦软件清零 **TWINT** 位，TWI 模块立即启动数据包的传送。
6. 当数据包发送完毕后，**TWCR** 的 **TWINT** 标志位会被置位，**TWSR** 更新为新的状态码，表示数据包成功发送。状态码同样会反映从机是否响应该数据包。
7. 应用程序查看 **TWSR** 的值，确定数据包已成功发送，收到的 **ACK** 为期望值。如果 **TWSR** 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，软件往 **TWCR** 寄存器写入特定值，指示 TWI 硬件发送 **STOP** 信号。写入的值将在随后详细说明。在写入的值中要置位 **TWINT**，来清零 **TWINT** 标志位。**TWCR** 寄存器的 **TWINT** 置位期间 TWI 不会启动任何操作。一旦软件清零 **TWINT** 位，TWI 模块立即启动 **STOP** 信号的传送。需要注意的是，在 **STOP** 信号发送完毕之后 **TWINT** 不会被置位。

尽管示例比较简单，但它包含了 TWI 数据传输过程中的所有规则。总结如下：

- 当 TWI 完成一次操作并等待应用程序的反馈时，**TWINT** 标志置位。**SCL** 时钟线会被一直