

# MSP430 与 I2C 总线接口技术

作者：周华 徐华 朱均

**摘要:** 分析了 MSP430 单片机 I/O 端口的结构特点, 提出了适合 MSP430 特点的 I2C 总线接口方案。结合实际给出了与常用 I2C 器件 AT24C02 的接口实例。

## 引言

MSP430 单片机与传统的 51 单片机在结构上有很大的区别。其中之一就是在 MSP430 的外围接口电路中, 没有提供像 51 那样控制外设读、写、地址锁存信号的硬件电路。与这种接口电路相适应, MSP430 更倾向使用 I2C 总线以及 ISP 等基于串行接口的外围器件。另一方面, 随着 I2C 技术的发展和成熟, 其硬件结构简单、高速传输、器件丰富等特点使该类器件的应用越来越广泛。因此研究新型单片机 MSP430 与 I2C 总线接口技术有着重要的意义。本文针对这一问题进行研究, 分析研究了 MSP430 与 I2C 总线接口方法, 提出了高效的接口方案。

## MSP430 单片机 I/O 端口控制特点

与 8031 单片机相比, MSP430 的 I/O 端口的功能要强大的多, 其控制的方法也更为复杂。MSP430 的 I/O 端口可以实现双向的输入、输出; 完成一些特殊功能: 驱动 LCD、A/D 转换、捕获比较等; 实现 I/O 各种中断。MSP430 采用了传统的 8 位端口方式保证其兼容性, 即每个 I/O 端口控制 8 个 I/O 引脚。为了实现 I/O 端口每一个引脚的复杂控制, MSP430 中的每个 I/O 口都对应一组 8 位的控制寄存器(如图 1)。寄存器中的每一位对应一个 I/O 引脚, 实现对该引脚的独立控制。寄存器的功能和数目是由该 I/O 口所能完成的功能以及类型确定的。

图 1 为 MSP430 的一个 I/O 端口的控制结构示意图。对于最基本的只能完成输入、输出功能的 I/O 端口其控制寄存器只有 3 个。其中, 输入寄存器保存输入状态; 输出寄存器保存输出的状态; 方向寄存器控制对应引脚的输入、输出状态。本文中用来实现 I2C 总线接口的 P6.6、P6.7 都属于这类的端口。此外, 有些 I/O 端口不但可以用作基本的输入输出, 而且可以用作其他用途, 比如可以作为 LCD 的驱动控制引脚。这类端口的控制功能寄存器实现引脚功能状态的切换。再者, 有一类端口不但可以完成上述两种端口的功能, 而且可以实现中断功能。该类端口拥有图 1 中所有的寄存器, 中断触发的方式以及中断的屏蔽性都可以通过相应的寄存器控制。本文中使用的 P2.0 就属于该类端口, 利用它来接收 LM92 发出的中断。通过上述的控制结构, MSP430 的 I/O 端口可以实现很丰富的功能。不仅如此, 其中一些 I/O 口还可以与 MSP430 中的特殊模块相结合完成更为复杂的工作。如与捕获比较模块相结合可以实现串行通信, 与 A/D 模块结合实现 A/D 转换等。此外, MSP430 I/O 端口的电器特性也十分突出, 几乎所有的 I/O 口都有 20mA 的驱动能力, 对于一般的 LED、蜂鸣器可以直接驱动无需辅助电路。许多端口内部都集成了上拉电阻, 可以方便与外围器件的接口。

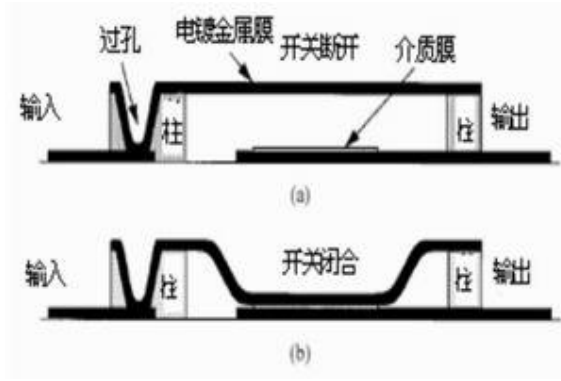


图 1 MSP430 的 I/O 端口控制结构示意图

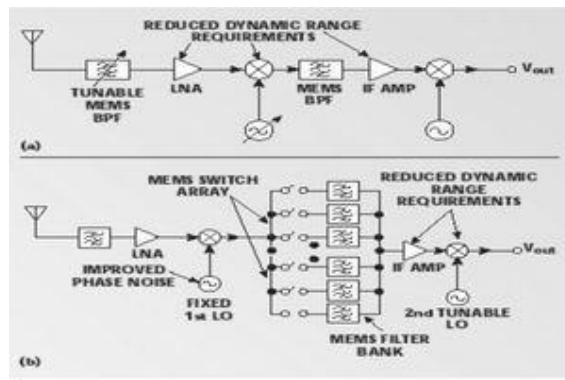


图 2 MSP430x41x 与 I2C 总线器件 LM92、AT2402 接口电路原理图

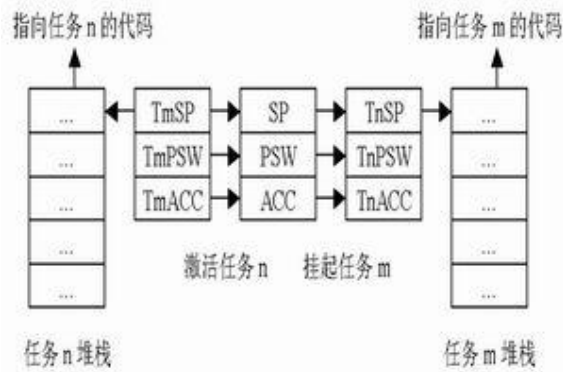


图 3 I2C 总线基本数据操作的时序

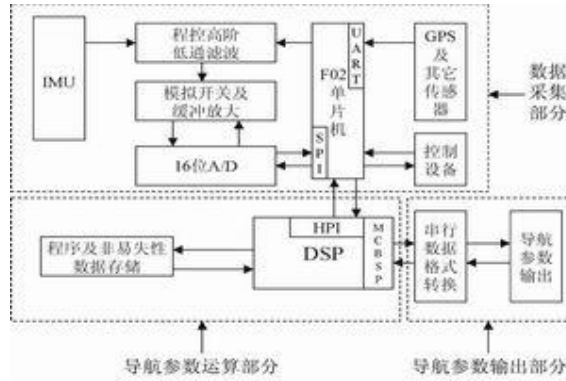


图 4 AT2402 读取指定字节数据的 I2C 总线时序图

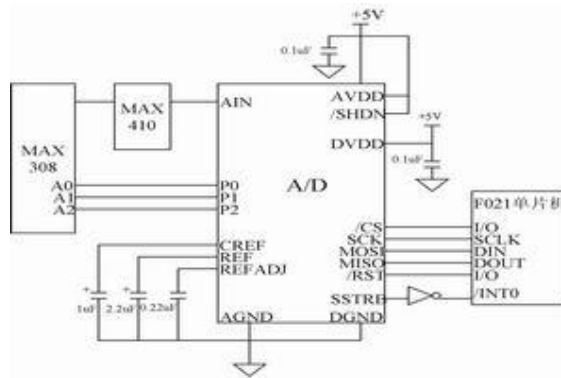


图 5 LM92 读取温度数据的 I2C 总线时序图

### MSP430 与 I2C 总线器件接口

MSP430 与 I2C 总线器件接口电路示于图 2,使用 41 系列单片机的 P6.6 产生 I2C 总线的时序同步信号;使用 P6.7 完成 I2C 总线的串行数据输入输出;利用 P2.0 接收 LM92 产生的中断信号。基于 I2C 总线规范,通过对 LM92 的 A0、A1 和 AT240 的 A0、A1、A2 设定不同的器件地址,两个器件可以共用 SCL、SDA。

与 8031 不同, MSP430 没有位空间,也没有专门执行位操作的控制电路。MSP430 中有关位操作的指令都是通过逻辑运算实现的。例如:

BISB #01000010B,P1OUT; 将 P1.6 和 P1.1 置位

XORB #01000010B,P1OUT; 逻辑或运算

该例中的置位指令 BISB 是用原操作数(01000010)与目的操作数(P1OUT)做逻辑或运算得到的。因此该命令与第二行的指令是等效的。虽然,这样的控制方法比起 8031 略显复杂,但它的控制能力有所增强。这种方式可以同时控制多个端口位。

众所周知,实现 I2C 总线协议主要是控制 SDA、SCL 使其产生协议所规定的各种时序。要控制 P6.7、P6.6 产生 I2C 总线要求的各种时序,就要频繁使用到输入、输出以及方向寄存器。而要减少代码的量,简化接口控制,最直接的方法就是减少有关寄存器操作次数。要实现这一想法需要软硬件结合,充分利用 I/O 口的特点以及 I2C 总线协议的特点。

从图 3 的基本数据操作时序可以发现:I2C 总线在无数据传输时均处于高电平状态;SDA 引脚是数据的输入输出端,它的状态变化最为复杂,控制它需要频繁的使用 P6IN、P6OUT、P6DIR 三个寄存器。

图 2 中的 R1、R2 是上拉电阻,其阻值由选用的 I2C 总线器件的电器特性确定。在本文中这两个电阻不但起上拉的作用,还有助于解决第一个问题。当 P6.6、P6.7 处于接收状态时,上拉电阻可以将该点的电平拉升为 VCC,从而确保总线空闲时有稳定的高电平。

方向寄存器相应位为输入时，就等于给 I2C 从器件发送了逻辑‘1’。将对应的方向控制位设为输出，然后输出寄存器相应位置为‘0’就可以实现发送逻辑‘0’。再进一步，如果将输出寄存器对应位设为‘0’，只控制方向寄存器的变化就可以发送两种逻辑电平。这样，在发送数据时只需要控制方向寄存器。对于 SDA 需要频繁切换输入输出状态的特点，本方法可以减少 15%左右的代码量，并使程序更清晰。这样就为第二个问题找到了很好的解决方法。

### I2C 总线控制时序的实现

I2C 总线中的各种操作都是由这些基本操作组合完成的。由于 I2C 总线器件的类型、功能、结构不尽相同，因此每一种器件具体控制时序有所区别。图 4 是 AT2402 读取指定字节数据控制时序。从图中可以看出一个读取操作中要使用到起始、发送字节、处理回应、接收字节、停止这些基本操作。对于 AT2402 还有其他控制的时序，如字节写时序、数据页读时序、地址读取时序等等。对于不同的功能时序，可以通过子程序的调用来实现。

LM92 是一种高精度的温度传感器，它也采用 I2C 总线方式控制。图 5 是该器件读取温度数据的时序。因为它的功能和结构与 AT2402 有很大的区别，所以二者控制时序不尽相同。如图 4 和图 5，虽然都是实现读取操作，但是二者时序差别很大，LM92 的控制时序明显要复杂的多。不过仔细分析可以看出这些时序也都是由一些基本操作组合实现的。这样就可以在上述方法的基础上完善 LM92 所需要的基本操作子程序，进而根据时序需要安排子程序实现对 LM92 的各种控制。

综上所述，要实现 I2C 总线的控制时序，需要仔细分析各种器件的时序要求及特点，构建所有的基本操作，并按时序要求合理安排基本操作。

### 结语

应用上述的设计方法和电路，实现了 MSP430 与 I2C 总线器件的接口，很好的控制 AT2402 和 LM92，达到了预期的目标。实践证明该方法对实现 I2C 总线器件控制非常有效，而且使用该方法编制的程序代码量小，执行效率高。该方法为 MSP430 与 I2C 总线接口提供了一种可行的方案。

### 参考文献：

1. Brian Merritt. I\_C Interfacing of the MSP430 to a 24xx Series EEPROM([R]). Texas, U.S.A. : Texas Instruments Incorporated, 2000.12.5-7.

本文内容来自互联网，著作权归原作者所有。由电子零件城 (<http://www.epcity.com/>) 整理并制作成 PDF 文件，仅供个人学习之用，不得用于任何商业目的，否则后果自负。如果您认为本 PDF 文件侵犯了您的任何权利，请来信 [epcity@epcity.com](mailto:epcity@epcity.com) 通知，本站立即删除。

搜集整理：电子零件城-笨笨兔 (QQ: 154502842)      2004-04-10