

CAN 控制器 MCP2510 及其应用

作者：第二炮兵工程学院 武翠琴 杨金岩 李艾华 来源：《国外电子元器件》

摘要：MCP2510 是 MT 公司生产的一种带 SPI 接口的 CAN 控制器，主要用于简化 CAN 接口应用设计。文中介绍了 MCP2510 的特点、结构和基本工作原理，给出了其应用接口电路及 SPI 接口协议的软件实现方法。

关键词：： MCP2510 CAN 接口 MCU

1 概述

MCP2510 是一种带有 SPI 接口的 CAN 控制器，它支持 CAN 技术规范 V2.0A/B；并能够发送接收标准的和扩展的信息帧，同时具有接收滤波和信息管理的功能。MCP2510 通过 SI 接口与 MCU 进行数据传输，最高数据传输速率可达 5Mb/s，MCU 可通过 MCP2510 与 CAN 总线上的其它 MCU 单元通讯。MCP2510 内含三个发送缓冲器、二个接收缓冲器。同时还具有灵活的中断管理能力，这些特点使得 MCU 对 CAN 总线的操作变得非常简便。

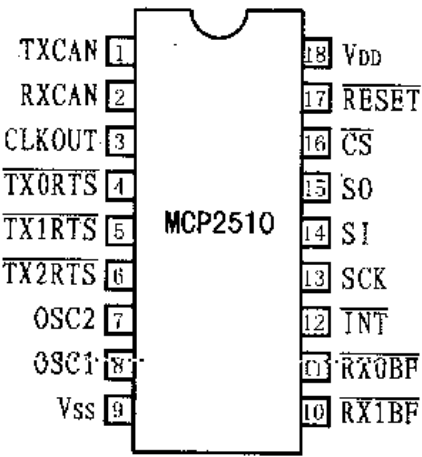


图 1 MCP2510 的引脚图

MCP2510 的主要特点如下：

- 支持 CANV2.0A/B；
- 具有 SPI 接口，支持 SPI 模式 0，0 和 1，1；
- 内含 3 个发送缓冲器和 2 个接收缓冲器，可对其优先权进行编程；
- 具有 6 个接收滤波器，2 个接收滤波器屏蔽；
- 具有灵活的中断管理能力；
- 采用低拉功耗 CMOS 工艺技术，其工作电压范围为 3.0~5.5V；有效电流为 5mA，维持电流为 10μA；
- 工作温度范围为-40~+125℃。

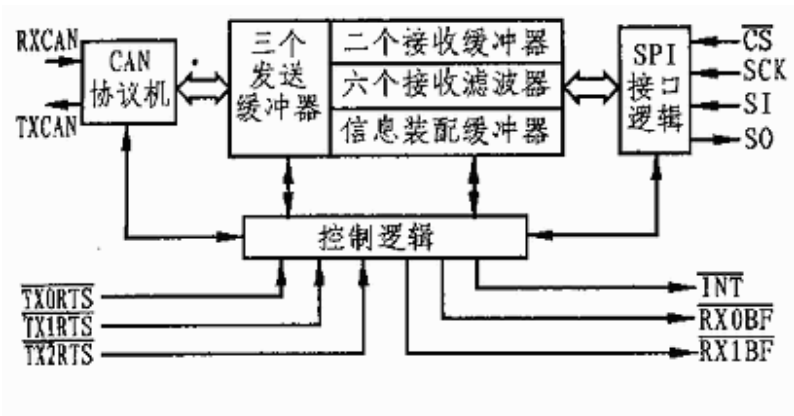


图 2 MCP2510 内部结构框图

2 管理功能

MCP2510 有 PDIP、SOIC 和 TSSOP 三种封装形式，有 18 个引脚。图 1 为 18 脚 PDIP 封装形式，有 18 个引脚。图 1 为 18 脚 PDP 封装 MCP2510 的引脚排列图。表 1 所列是各引脚功能的详细描述。

表 1 MCP2510 的引脚功能

引 脚	名 称	功 能
1	TXCAN	去 CAN 总线的发送输出引脚
2	RXCAN	来自 CAN 总线的接收输入引脚
3	CLKOUT	具有可编程预分频的时钟输出引脚
4 5 6	TX0RTS TX1RTS TX2RTS	发送缓冲器发送请求端，或作数字输入端
7 8	OSC2 OSC1	振荡器输出端、输入端
9	Vss	地端
10 11	RX1BF RX0BF	接收缓冲器中断引脚，或作数字输出端
12	INT	中断输出引脚
13	SCK	SPI 接口的时钟输入端
14	SI	SPI 接口的数据输入端
15	SO	SPI 接口的数据输出端
16	CS	SPI 接口的片选输入端
17	RESET	复位端
18	VDD	电源端

3 结构及工作原理

图 2 是 MCP2510 的内部结构框图，其中 CAN 协议机负责与 CAN 总线的接口，SPI 接口逻辑用于实现同 MCU 的通讯，而寄存、缓冲器组与控制逻辑则用来完成各种方式的设定和操作控制。现结合其工作过程将各部分的功能、原理作一介绍。

3.1 收发操作

MCP2510 的发送操作通过三个发送缓冲器来实现。这三个发送缓冲器各占据 14 个字节的 SRAM。第一字节是控制寄存器 TXBNCTRL，该寄存器里的内容设定了信息发送的条件，且给出了信息的发送状态；第二至第六字节用来存放标准的和扩展的标识符以及仲裁信息；最后八个字节则用来存放待发送的数据信息。在进行发送前，必须先对这些寄存器进行初始化。

3.2 中断管理

MCP2510 有 8 个中断源，包括发送中断、接收中断、错误中断及总线唤醒中断等。利用中断使能寄存器 CANINTE 和中断屏蔽寄存器 CANINTF 可以方便地实现对各种中断的有效管理。当有中断发生时，INT 引脚变为低电平并保持在低电平，直到 MCU 清除中断为止。

3.3 错误检测

CAN 协议具有 CRCF 错误、应答错误、形式错误、位错误和填充错误等检测功能。MCP2510 内含接收出错计数器（REC）和发送出错计数器（TEC）两个错误计数器。因而对于网络中的任何一个节点来说，都有可能因为错误计数器的数值不同而使其处于错误-激活、错误-认可和总线-脱离三种状态之一。

4 应用实例

笔者利用 MCP2510 和 CAN 总线收发器 82C250 构成了一个 CAN 总线分布式温湿度测控网络。该系统包括一个主控制器和多个节点控制器（温湿度测控单元），这种网络拓扑结构采用了总线式结构，且结构简单、成本低，采用无源抽头连接。因而系统的可靠性较高。其信息传输采用 CAN 通信协议，通信介质采用双绞线。由于 CAN 总线是基于发送报文的编码，它不对 CAN 控制节点进行编码，故系统的可扩充性比较好，同时增删 CAN 总线上的控制节点不会对系统的其余节点造成任何影响。

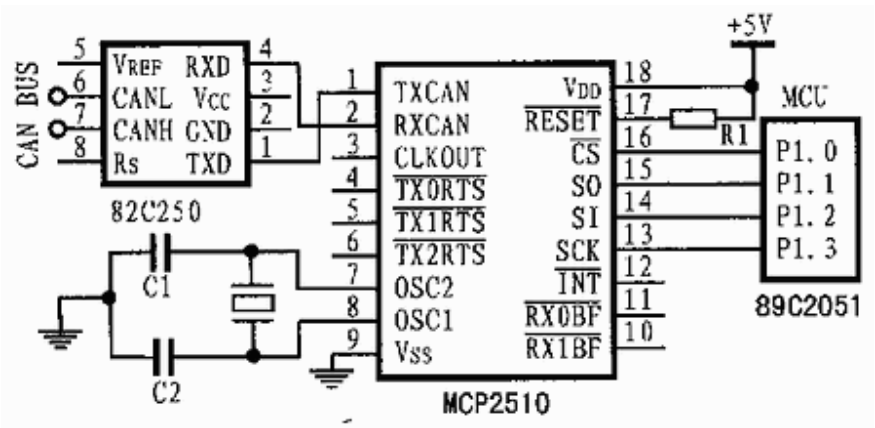


图 3 MCP2510 接口电路

节点控制器的 MCU 可选用具有 SPI 接口的微处理器，也可采用不带 SPI 接口的微处理器。本系统采用的是不带 SPI 接口的 89C2051 微处理器，89C2051 可通过 4 条普通的 I/O 线与 CAN 控制器的 SPI 接口直接相连，并可用软件算法来实现 SPI 接口协议。CAN 总线收发器 82C250 则作为 MCP2510 与物理总线的接口。如果需要进一步提高系统的抗干扰能力，可在 MCP2510 和 82C250 之间再加一个光电隔离器。MCP2510 与 MCU 和 CAN 收发器的实际接收如图 3 所示。

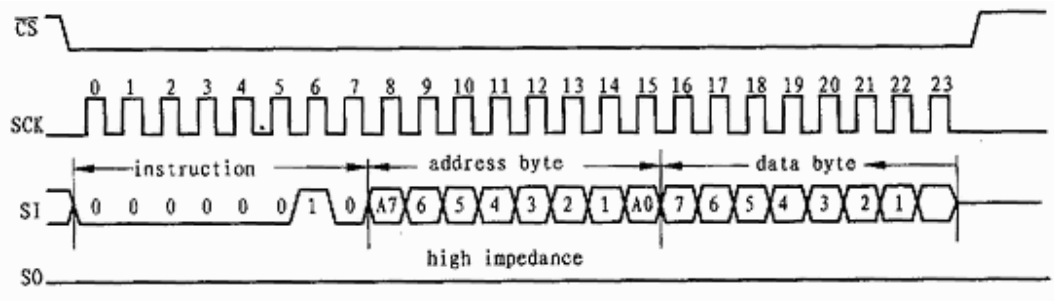


图 4 SPI 写命令时序图

5 SPI 接口协议的软件实现

在使用 CAN 控制器 MCP2510 时，需要对 MCP2510 进行初始化以及对 CAN 总线上的数据进行收发操作。与其它 CAN 控制不同的是，读、写 MCP2510 的发送和接收缓冲器必须通过 SPI 接口协议的读写命令来实现。SPI 写命令时序如图 4 所示。写指令首先被发送到 MCP2510 的 SI 引脚，并在 SCK 的上升沿锁存每个数据位，然后发送地址和数据。执行完毕指令后，数据被写进指定的地址单元中，再通过 SPI 接口协议的写命令来设置发送位以启动发送。

SPI 接口协议的读命令时序图如图 5 所示。操作时，首先将读指令和地址发送到 MCP2510 的 SI 引脚，并在 SCK 的上升沿锁存每个数据位。同时把存贮在这个地址单元中的数据在 SCK 的下降沿输出到 SO 引脚。当执行读写操作时，CS 引脚应始终保持在低电平。

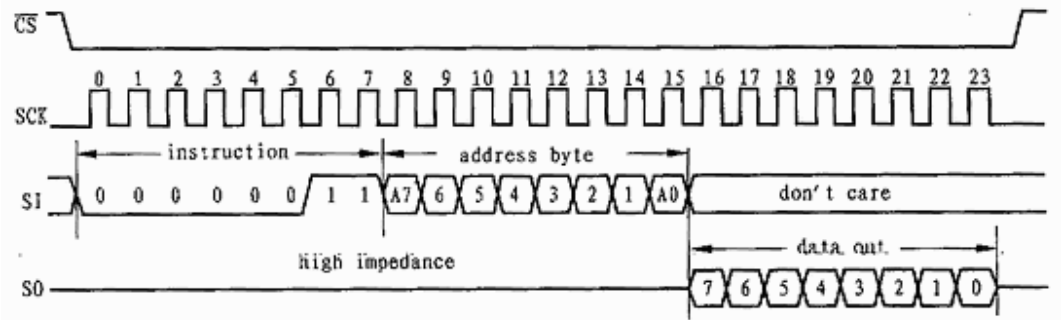


图 5 SPI 读命令时序图

用软件编程实现 SPI 接口协议的汇编语言程序如下：

SPI 接口写子程序：

```
MES_WRITE: CLR C ; 清除进位标志 C

MOV R3, #8 ; 置循环次数 R3 为 8

WRIT1: CLR SCK ; 置 SCK 为低电平

NOP

RLC A ; A 中存放写指令或地址、数据，带进位标志左移一位

MOV SI, C ; A 中的最高位送 SI

NOP

SETB SCK ; 置 SCK 为高电平，此时 A 中的最高数据被锁存

NOP

DJNZ R3, WRIT1 ; 循环 8 次

RET
```

SPI 接口读子程序：

MES_READ: CLR ACC ; 清除 A 中内容

CLR C ; 清除进位标志 C

MOV R6, #8 ; 置循环次数 R6 为 8

READ1: CLR SCK ; 置 SCK 为低电平

NOP

MOV C, SO ; SO 线输出的数据送给进位标志 C

RLC A ; A 带进位标志左移一位

SETB SCK; 置 SCK 为高电平

NOP

DJNZ R6, READ1; 循环 8 次后, SO 线输出的数据送到 A

RET