

基于 STM32 嵌入式系统的智能控制网络终端设计

智能网络终端是实现智能化管理的嵌入式终端设备。通常具有安全门禁系统和自动化控制的基本功能。本文重点介绍基于 STM32 的智能嵌入式终端的网络控制功能的设计和实现。

1 智能网络通信实现方案

将嵌入式系统与 Internet 连接起来实现远距离信息获取和控制功能的本质是嵌入式系统本身能够实现 TCP / IP 网络通信协议。该协议的解决方案总体上可分为两大类。第一类是直接在嵌入式设备上实现 TCP / IP，使之直接连上 Internet。这实质上是由 MCU 及内部固化 TCP / IP 协议的芯片组成应用系统的核心，MCU 可以直接拨号上网，这种方法的硬件电路相对简单，也不需要中间环节的支持。但是由于使用了 TCP / IP 协议芯片，因此需要大容量的程序存储器，而且要求 MCU 有较高的运行速度。同时，应用程序设计师还必须熟悉 TCP / IP 协议，且软件设计复杂，工作量大。目前，此方案的典型代表有 SX-stack、Sciko 公司的 S7600A 芯片；第二类是使嵌入式设备经过通信转换后，再通过公用的 TCP / IP 转接口(网关 Gateway)与 Internet 相连。目前有以下几种方式：

- (1)通过使用独立于微控制器 MCU 的专用的网络接口芯片来完成单片机与网关间的协议转换，并以此作为链接到 Internet 的桥梁。例如韩国 WIZnet 公司的 W5100 芯片；
- (2)用单片机控制以太网网卡进行数据传输，此时必须加载 TCP / IP 协议到单片机中；
- (3)用代理协议来完成与网关间的协议转换。

本设计采用独立于微控制器 MCU 的专用网络接口芯片来实现 TCP / IP 协议，微处理器选择 STM32F103VB，网络芯片选择 W5100。二者通过 SH 接口进行连接，其连接示意图如图 1 所示。

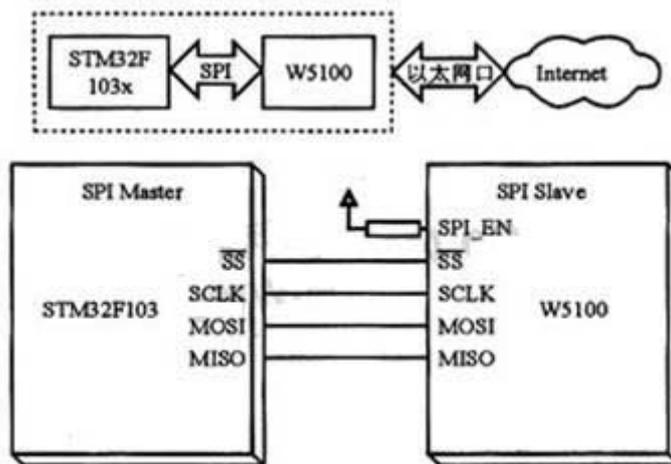


图1 STM32F103与W5100的连接示意图

2 硬件平台的设计

2.1 总体设计

本系统的硬件可分为采集模块、处理模块、键盘和显示模块、传输模块、控制模块和供电模块等。其中采集模块负责采集信息；处理模块负责控制整个系统的工作，处理和存储采集到的信息，控制模块则可以进行相应的控制；键盘和显示模块负责显示相关信息以提示现场人员操作；网络传输模块负责信息与数据服务器进行信息交互；供电模块负责为整个系统供电。图 2 所示是系统的结构框图。通过这种结构可以自主开发出一套高性价比、低成本、低功耗、开发周期短、运行速度快、可靠性高、功能齐全、扩展性强的网络控制终端。



图2 系统结构框图

2.2 硬件实现

本系统的处理器模块选用 ST 公司刚刚推出不久的、基于 ARM Cortex™-M3 内核的新一代嵌入式 STM32 芯片。这是一款专为嵌入式应用而开发的内核。它带有把中断之间延迟降到 6 个 CPU 周期的嵌套向量中断控制器，允许在每一个写操作中修改单个数据位的独立位操作，可进行分支指令预测、单周期乘法、硬件除法且有高效的 Thumb 2 指令集。

这些改良技术使 Cortex-M3 内核具有优异的代码密度、实时性和低功耗性能。所有这些新功能都同时具有目前最优的功耗水平，非常适合应用于长时间工作的终端控制系统。

本系统由+12 V 电源输入，同时经过 LM2575 和 78L09 稳压后分别得到 5 V 和 9 V 电压，5 V 再经过 SPX1117M3-3.3 电压稳压转换后得到 3.3 V，这样 3.3 V、5 V 和 9V 电压即能够很好地满足内核、外设和外部电路的供电。系统时钟源可采用外部晶振，内部 PLL 电路可以调整系统时钟，使系统运行速度更快。为了提供性能优越的电源监控性能，这里选取了专门的 MAX811 系统监视复位芯片，该芯片可以通过手动控制系统的复位，同时还可以实时监控系统电源，一旦系统电源低于系统复位阈值，电路中的 MAX811 将产生一个 140ms 的复位脉冲信号来对系统进行复位。

系统网络模块选用的 W5100 芯片是一款多功能的单片网络接口芯片。它内部集成有 10 / 100Mbps 以太网控制器，可支持自动应答(全双工 / 半双工模式)，主要用于高集成、高稳定、高性能和低成本嵌入式系统。使用 W5100 可以实现没有操作系统的 Internet 连接。W5100 与 IEEE802.310BASE-T 和 802.3u 100BASE-TX 兼容。此外，W5100 内部集成有全硬件的、经过多年市场验证的 TCP / IP 协议栈、以太网介质传输层(MAC)和物理层(PHY)。其硬件 TCP / IP 协议可支持 TCP、UDP、ICMP、IGMP、IPv4、ARP、PPPoE 和 IGMP。同时可支持 4 个独立的端口(Socket)通信，其内部 16 K 字节的发送，接收缓冲区可快速进行数据交换，其最大通信速率可达 25 Mbps。利用 W5100 提供的多种总线(并行总线和 SPI 总线)接口方式可以很方便地与各种 MCU 连接。可以说，W5100 器件的推出，大大简化了硬件电路设计，并可使微控制器系统在没有操作系统的支持下实现单芯片接入 Internet 的理想。一般情况下，只需设置寄存器和存储器，就可以通过 W5100 芯片进行 Internet 连接。

在设计中，STM32F103 与 W5100 可通过 SPI 接口进行连接。STM32F103 作为 SPI 主设备，W5100 作为 SPI 从设备，并由 STM32F103 为 W5100 提供通信时钟，二者的接口连接图如图 3 所示。

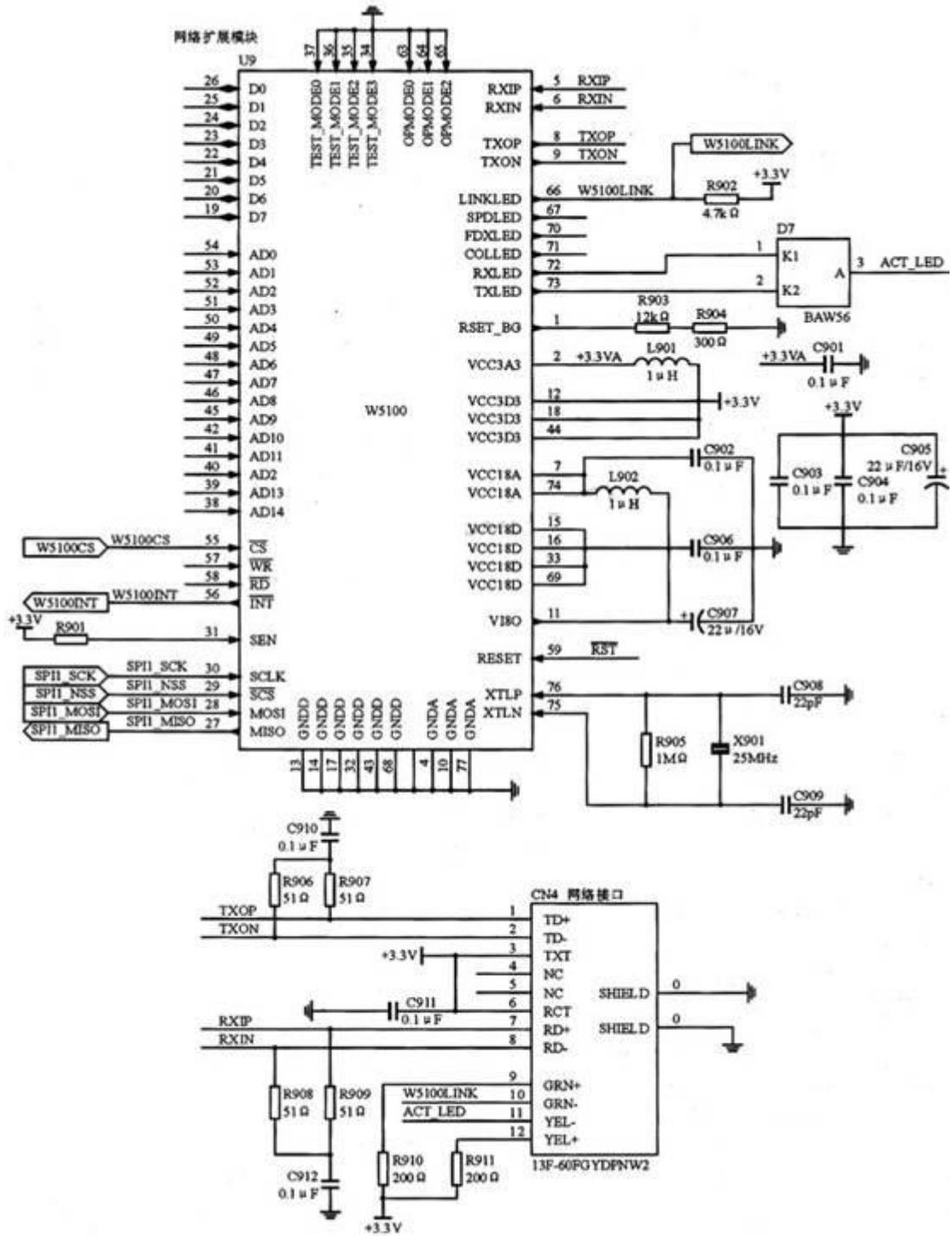


图3 W5100与STM32F103的接口电路

但是，STM32F103 在与 W5100 以 SPI 方式连接时，应注意以下几点：

- (1) W5100 的 SEN 引脚必须通过一个 10 kΩ 的电阻接到高电平，以选 W5100 的 SPI 接口方式；

(2)W5100 的 Do~D7、Ao~A14 及 CS、WR、RD 可以悬空，只使用 SCLK、SCS、MOSI 和 MISO 四根信号线；

(3)W5100 的 INT 引脚是中断输出，MCU 需要根据该信号来判断 W5100 的中断状态；

(4)LINKLED 是 W5100 输出的以太网物理层信号，MCU 需要根据该信号判断以太网的联接是否正常。该信号一般需要上拉输入到 MCU；

(5)W5100 的 SPDLED、FDXLED、COLLED 可以根据选择是否需要输入到 MCU。

另外，系统中的存储器模块可以选用 ST 公司的 64 MB 的串行代码存储闪存 M25P64。该器件的数据传输时钟频率为 50 MHz，数据读取吞吐量为 50 MB / s，其简单的 SPI 串行外设接口可简化系统的设计。本硬件平台使用两片 M25P64 级联，可存储 128 MB 的信息，图 4 所示是其存储电路，它完全可以满足平台信息采集存储的需求。为方便人员察看，系统中的 LCD 屏可选用(114.0 mm×64.0 mm)。

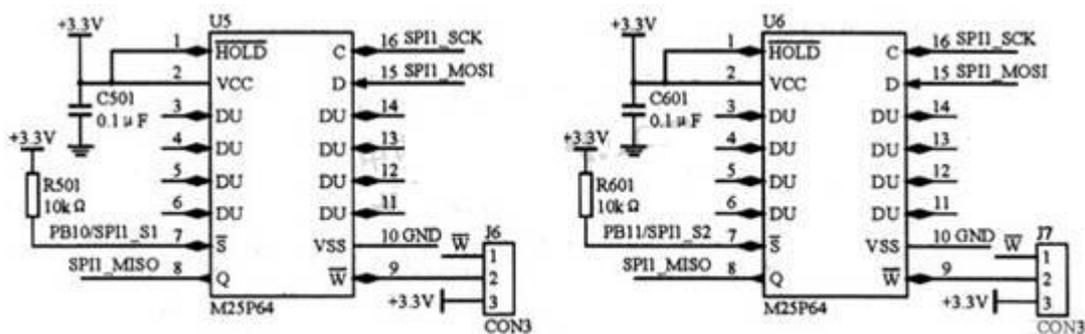


图4 M25P64存储器电路

3 W5100 的驱动程序设计

W5100 的驱动程序主要完成初始化和端口的数据通信等功能。

通过 SPI 接口来配置 W5100 的寄存器一般具有固定的命令格式。根据 SPI 协议，SPI 设备之间只有两条数据线。因此需要定义操作码(OP_Code)。W5100 使用两种操作码：读操作码和写操作码。除了这两种代码，它将忽略和不响应其它的操作码。在 SPI 模式下，W5100 只对“完整的 32 位数据流”进行操作。这个 32 位的数据流包括 1 个字节的操作码、2 个字节的地址码和 1 个字节的数据。其操作码、地址和数据字节的传输都是高位(MSB)在前、低位(LSB)在后。也就是说，SPI 数据的第一位是操作码域的 MSB，最后一位是数据域的 LSB。W5100 的 SPI 数据格式如表 1 所列。

表1 W5100的SPI数据格式

命令	操作码域	地址域	数据域
写操作	0xF0	2字节	1字节
读操作	0x0F	2字节	1字节

初始化 W5100 主要是设置 W5100 的公共寄存器，包括 MR、IMR、RTR、RCR、GAR、SUBR、SHAR、SIPR、RMSR 和 TMSR。一般在使用 W5100 之前，都要对 W5100 进行初始化。

3.1 数据通信的建立

TCP 是面向连接的通信方式，它首先必须建立连接，然后才能利用 IP 地址和端口号进行数据通信。TCP 有两种建立连接的方式，一是通过服务器模式(被动打开)等待连接请求；二是通过客户模式(主动打开)发送连接请求给服务器。本例采用 TCP 客户模式。在建立 TCP 连接之前，一般都需要初始化端口，包括设置端口号、设置 W5100 为 TCP 模式和写入 OPEN 命令。端口初始化主要配置端口 o 的相关寄存器，包括：So_PORT、So_MR 和 So_CR。

图 5 是 W5100 在 TCP 客户模式的处理流程。

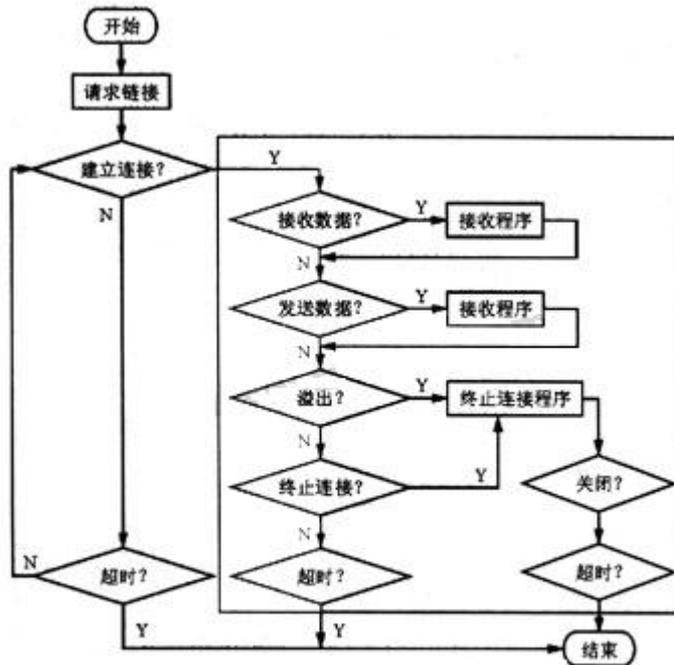


图5 W5100在TCP客户模式的处理流程

3.2 中断处理

在处理 W5100 的中断时，首先应访问 W5100 的中断寄存器(IR)，可用 MCU 通过访问 IR 获得产生中断的来源。任何中断源都可以被中断寄存器(IMR)的相应位所屏蔽，因此，若要使用某个中断源，先要置位该中断源在 IMR 中的相应位，这样，当 IR 中对应位置位时，才会产生中断。此后当中断产生时，即进入中断处理程序。对于每个中断事件，处理方式可由用户自己定义。

假如使用的是端口 0 中断。可在 W5100 的初始化程序中先将 IMR 中的 IM_IR 0(端口 0 中断屏蔽位)置位。这样，当端口 0 中断发生时(IM_IR0=1, So_INT=1)，系统将开始读端口 0 中断寄存器(So_IR)，在 W5100 的端口寄存器中，主要设置有建立连接(CON)、终止连接(DIS

CON)、数据发送完成(SEND_OK)、接收数据(RECV)和超时(TIMEOUT)等中断事件。

3.3 数据接收

当端口产生接收中断时,可调用接收函数 `S_rx_process(SOCKET s)`将端口接收到的数据缓存到 `Rx_buffer` 数组中,并返回接收的数据字节数。当读完所有的数据后,可将接收存储器读指针寄存器(`So_RX_RD`)的值加上读取的数据长度,然后再写入 `So_RX_RD`,最后向端口 `o` 的命令寄存器(`So_CR`)写入 `RECV` 命令,以等待下次接收数据。但要注意在计算实际物理偏移量 `rx_offset` 时, `S_RX_SIZE` 必须保证和在初始化代码中定义接收缓存区的大小一致。

3.4 数据发送

通过 `Socket` 发送数据时,首先把要发送的数据缓存在 `Tx_buffer` 中。此外,在发送数据时,还需先检查发送缓存区的剩余空间的大小,然后控制发送数据的字节数。端口发送缓存区的大小由发送存储器空间寄存器(`TMSR`)确定。在数据发送处理过程中,剩余空间的大小将因写入数据而减少,发送完成后又自动增加。把 `Tx_buffer` 的数据完全写入端口的发送数据缓存区后,可将端口传输写指针寄存器(`Sn_TX_WR`)中的值加上写入的数据长度,再写入 `Sn_Tx_WR`,以指示发送数据的长度,最后在命令寄存器(`Sn_CR`)中写入 `SEND` 命令,以启动发送。

4 结束语

本文对控制终端设计提出了一些新颖的设计思路,利用该思路可实现现场和网络的混合控制。该系统创新处主要包括如下几点:

(1)具有 `USB \ UART` 等输入输出端口,可用于现场测控;

(2)可通过以太网接口实现网络传输、远程网络测控和信息资源共享等功能;

(3)可通过大容量的存储器来进行大容量的数据采集，并将其存储在片上处理；

(4)具有 LCD 显示单元，能够实现数据和状态信息的现场显示，使用户现场操作更加简单。