

文章编号: 1005-8451 (2009) 12-0047-04

## 基于 Parallel I/O 模式的 Lon-232 网关设计

冯道宁, 伍咏晖

(广西机电职业技术学院 电器工程系, 南宁 530007)

**摘要:** 介绍一种基于 Neuron 芯片和单片机 89C51 联合作为通信控制器的 RS232-Lonworks 的网关设计, 并给出详细的硬件电路图, 着重介绍在网关中 parallel I/O 通讯方式的选择和通讯协议的实现过程, 以及神经元模块和 89C51 单片机的程序编写流程。

**关键词:** 网关; Lon Works; 双 CPU; Parallel; Neuron

中图分类号: TP273 文献标识码: A

### Design of Lonworks-232 gateway based on Porellel I/O mode

FENG Dao-ning, WU Yong-hui

(Department of Mechanical and Electrical Technology, Guangxi Vocational College, Nanning 530007, China)

**Abstract:** It was introduced a design of Lon-232 gateway based on the Neuron chip and singl chip 89C51, demonstrated hardware circuit diagram. It was concentrated on the chose of parallel I/O communication mode, way to implement protocol of communication and program of the neuron and 89C51.

**Key words:** gateway; LonWorks; dual CPU; Parallel; Neuron

LonWorks 总线是已广泛应用于铁路交通运输、楼宇、能源等多个领域。网桥可以在串行通讯与 Lon 网之间建立了连接通道, 完成上位主机或其他设备与 Lon 网之间的数据通信, 提高设备的通用性和易用性。LONWORKS 技术的核心是 Neuron 芯片或称为神经元模块, 它是网关的核心硬件, 神经元芯片在大多数 Lon 网关中是一个独立的处理器。若需要使网关具备更强的信号处理能力或 I/O 通道, 可采用其它处理器来处理并由神经元芯片交换数据, 此时神经元芯片只完成通信功能。

### 1 双 CPU 的硬件设计

网关采用神经元 Neuron 芯片和单片机 89C51 联合作为通信控制器, 并对双 CPU 编程, 使两者协同工作。串行端口由单片机引出, 并搭配电平转换芯片和外围电路。节点结构如图 1。

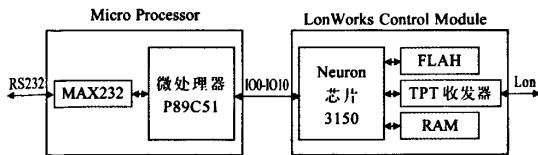


图 1 节点结构图

收稿日期: 2009-04-15

作者简介: 冯道宁, 讲师; 伍咏晖, 讲师。

单片机负责串口通信的收发控制。Lon 接口则仍然由神经元芯片引出, 通过专用收发器接入 Lon 网络。神经元芯片负责 Lon 网络上的报文收发控制, 以及对数据的打包和解包工作。单片机和神经元芯片之间通过并行方式交换数据。通过双 CPU 的协同工作实现 LonWorks 到串口的网关。硬件连接如图 2。

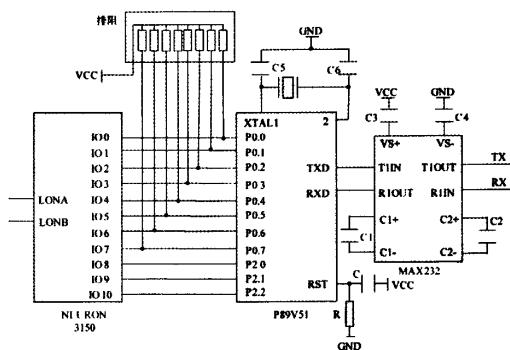


图 2 硬件连接图

### 2 软件设计

软件设计是整个网关设计的重点, 包括单片机与神经元芯片通讯方式的选择, 通讯协议的实现, 还有如何对神经元芯片和单片机进行编程 4 个

方面。

## 2.1 Neuron 的并行通信方式

如上图 2 在单片机和神经元芯片之间通过神经元的并行通信 (Parallel I/O) 方式交换数据。神经元芯片的并口通信方式有其固定的通信协议和方式，单片机要和神经元芯片通过这种方式通信，则要在程序中模拟神经元并口通信的时序和协议。因此，在编写软件前首先要选择神经元芯片 IO 的并口通信模式。神经元芯片提供的并行双向 I/O 对象有 Muxbus I/O 对象和 Parallel I/O 对象两种，前者既可与处理器又可与其它设备相连接，后者适合与处理器进行连接。Parallel I/O 对象使用全部的 11 个 I/O 管脚作为具有握手信号的 8 bit 平行接口。其中 IO0-IO7 是 8 bit 双向数据总线，IO8-IO10 是控制信号线。该接口允许数据以最高 3.3 Mb/s 的速率传输。选用并行通信 Parallel I/O 方式有 2 个原因：首先，当神经元芯片接到一个微处理器上或计算机系统的总线上时，对于一个已经存在的基于处理器的系统，该接口可以将神经元芯片作为一个通信芯片使用，提供更多的应用性能，或者提供给更多的存储区。其次，对于应用级网关，2 个神经元芯片通过并行接口背对背连接，生成 2 个收发器接口，用来从一个系统向另一个系统传送数据包。

Parallel I/O 接口方式是双向的，其方向（读/写）由说明为 Master 的设备控制。当使用该接口时，神经元芯片可以是 Master，也可以是 slave。并行接口 (Parallel IO) 对象提供 3 种不同的并行接口的配置：Master、slave A 和 slave B。Master 和 slave A 方式一般用于普通并行接口和用于神经元芯片到神经元芯片的通信。考虑到单片机的处理能力较强，且在本应用中，单片机和神经元芯片是独立的点对点通信，因此，在节点中选用了 slave A 模式，使单片机成为主 CPU，而神经元芯片为从 CPU。

## 2.2 传递协议

在 slave A 模式中，IO8 为片选信号线 (~CS)，IO9 为读写信号线 (R/~W)，IO10 为握手信号线 (HS)。在通信过程中，单片机首先通过片选信号线，选中神经元芯片，然后等待神经元芯片的握手信号 (HS) 响应。单片机控制读写信号线 (R/~W) 的状态。根据读写信号线 (R/~W) 的状态确定当

前是要读神经元芯片还是向神经元芯片写数据。当读写信号线 (R/~W) 处于读的状态时，则神经元芯片拥有写令牌，在选中的情况下，可以向总线上写数据，单片机则根据时序，按字节读取。

主 CPU 和从 CPU 之间的数据传输通过虚拟的写令牌传递协议 (virtual write token-passing protocol) 得以实现。主 CPU 和从 CPU 交替地获得写令牌 (write token)，实现双向通讯。拥有写令牌的一方可以写数据（不超过 255 byte），或者不写任何数据传送一个空令牌。传送的数据要遵从一定的格式，即在要传送的数据前面加上命令码和传送数据的长度，命令码有 CMD\_XFER (写数据)，CMD\_NULL (传递空令牌)，CMD\_RESYNC (要求从机同步)，CMD\_ACKSYNC (确认同步) 4 种，最后以 EOM 字节结束。

根据以上协议，写数据和传递空令牌的格式分别为：

CMD_XFER	Length	Data	EOM
CMD_NULL	EOM		

图 3 写数据和传递空令牌的格式

通信以前，主从 CPU 之间先建立握手信号，即 HS 信号有效（这由 3150 的固件自动实现），然后，主 CPU 再送一个 CMD\_RESYNC 命令，要求从 CPU 同步，从 CPU 接收到这个信号以后，则发送 CMD\_ACKSYNC，表示已经同步，可以进行通信。

## 2.3 神经元模块软件设计

此部分应用程序采用 Neuron C 语言编写，主要起 Lon 网接口作用，向下与 51CPU 进行并口通讯，向上与 Lon 网利用 LonWorks 收发器进行通讯。将 3150 的 IO 定义为并口 (Parallel IO) 对象类型，并使其工作在 A 工作方式 (Slave A Mode)。

对 Parallel IO 对象进行显式配置的 Neuron C 语句和说明如下：

IO\_0 parallel slave| slave\_b| master io-object-name；

IO\_0 — 指定管脚 IO\_0。并行方式要求用到神经元芯片的所有 11 个 IO 管脚，且必须指定管脚 IO\_0 定义。

slave| slave\_b| master — 指定 slave、slave\_b 或 master 模式。根据应用需求选用不同的模式定义。

io-object-name 由用户为该并行 IO 对象指定的名字。

当将神经元的 IO 配置为 Parallel IO 对象时, 就不能再将其说明为其它 IO 对象。

为了使用神经元芯片的并行 IO 对象, `io_in()` 和 `io_out()` 函数需要一个指向 `parallel_io_interface` 结构的指针:

```
struct parallel_io_interface
{
    ...
    unsigned int length;
    unsigned int data[DATA_SIZE];
} pio; // 并行对象接口数据结构
```

上面的结构必须被说明。用一个相关的 `DATA_SIZE` 定义来表明数据传输中所期待的最大的数据缓冲区。在 `io_out()` 函数中, `length` 为输出的字节数, 它是由应用程序设定的。在 `io_in()` 中, `length` 为期待输入的字节数。如果实际输入的字节数大于 `length`, 则输入数据流无效, 且 `length` 被设为 0; 实际输入的字节数小于等于 `length` 时, `length` 值为实际读入的字节数。长度域必须在调用 `io_in()` 或 `io_out()` 之前设置。`length` 和 `DATA_SIZE` 最大值为 255。并行通信是神经元芯片 IO 的一种典型用法, 因此, 神经元部分的虚拟写令牌传送协议和内部实现程序在固件中都有详细定义。在写应用程序时, 要使用并行通信, 只要调用 Neuron C 中提供的以下相关函数和事件, 即可方便的实现并行通信。(以下说明中, `parallel-io-object-name` 指所定义的并行对象的名字。)

下面给出其具体实现的代码示例:

```
IO_0 parallel slave DataBus; // 定义并口对象 DataBus
typedef struct
{
    unsigned int length;
    unsigned int data[DATA_SIZE];
} pio; // 定义并口通信结构
far pio pin,pout; // 定义并口输入输出数据缓冲区
msg_tag node3; // 定义报文标签
when(msg_arrives(msgcode)) // 网
```

络上有报文到达

```
{
    pout.length=msg_in.len+1;
    memcpy(pout.data,msg_in.data,msg_in.len); // 将报文放入缓冲区
    io_out_request(DataBus);
    // 向并口请求写令牌
}
when(io_in_ready(DataBus))
// 并口有数据到达
{
    pin.length=DATA_SIZE;
    io_in(DataBus,&pin);
    // 读取并口数据至缓冲区
}
when(io_out_ready(DataBus))
// 3150 拥有写令牌
{
    io_out(DataBus,&pout);
}
..... // 其它程序略
```

## 2.4 单片机的程序结构

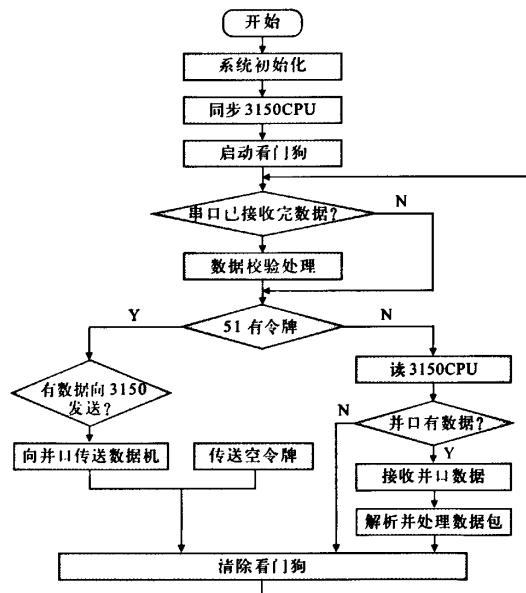


图 4 单片机程序流程图

单片机部分的程序用 C51 来编写, 如前所述

# 自动售检票设备液晶显示屏定时开关控制的开发与应用

曾震宇, 汤健

(广州市地下铁道总公司 运营事业总部, 广州 510310)

**摘要:** 根据液晶显示屏的原理和控制方法, 编制一个定时控制液晶显示器关闭/开启的控制软件, 应用在广州地铁的 AFC 系统设备上, 以此达到延长液晶显示屏的使用寿命, 节能降耗的效果。

**关键词:** 城市轨道交通; 自动售检票系统设备; 液晶显示 (LCD) 屏; 定时控制开/关

中图分类号: TU311.2 文献标识码: A

## Development and application of LCD screen's switch control in AFC equipment

ZENG Zhen-yu, TANG Jian

(Operation Division, Guangzhou Metro Corporation, Guangzhou 510310, China)

**Abstract:** On basis of the LCD screen theory and control method, it was aimed to design a software to control the LCD screen's time turning on and off, which was applied in the AFC System of Guangzhou Metro so as to prolong the LCD screen's life span and reduce wastage.

**Key words:** intercity rail transit; AFC; LCD; time turning on and off

自动售检票系统(简称 AFC)主要分为中央系统和站级系统。中央系统主要是由中央计算机组成, 实现数据的采集、监控和统计管理等, 是自动售检票系统的核心系统。站级系统主要由自动售票机、闸机、自动验票机、半自动售票机和车站

收稿日期: 2009-09-28

作者简介: 曾震宇, 工程师; 汤健, 助理工程师。

计算机组成, 直接面向乘客的一套服务设备。液晶显示屏已经在 AFC 系统中广泛应用, 是设备与乘客“交流”的渠道, 其服务质量直接影响设备对乘客的服务水平。

## 1 地铁 AFC 设备液晶显示屏的运行情况分析

(1) 目前地铁 AFC 设备是连续 24 h 和不断电

主从 CPU 通过虚拟的写令牌传送协议通信。神经元的并行模式由芯片内部固件程序实现, 程序代码较为简单。而单片机部分的程序则相对复杂, 要在应用程序中模拟 3150 的写令牌传送协议。根据神经元在 Slave A 模式的定义, 单片机通信程序主要完成 4 个工作: 与 3150 同步; 向 3150 写数据; 从 3150 读数据; 传送空令牌给 3150。此外, 单片机除了通过并口和 3150 通信外, 还要通过串口和下位机通信, 进行串口数据接收和数据处理。串口部分的数据接收主要通过串口中断程序完成。数据接收完后, 在主程序中进行数据处理。单片机主程序的流程如图 4。

## 3 结束语

在国产某型列车内的车厢代理网关以及主网

关均使用了该方案开发 Lon-232 网关的产品, 经过实际测试其数据传输速率和其稳定性完全能满足列车通讯网络的要求。

### 参考文献:

- [1] 凌志清. 从神经元芯片到控制网络[M]. 北京: 北京航空航天大学出版社, 2002.
- [2] 薛立军. LonWorks 总线双应用 CPU 智能节点设计[J]. 电工技术, 2003 (5).
- [3] Neuron C Programmer's Guide[M]. Echelon Corporation 2002.
- [4] 郝凤涛. 基于 AT89C52 单片机 LonWorks-RS232 协议网关的设计与实现[J]. 仪器仪表用户, 2008 (1).
- [5] 冯艳娜. 基于 Lonworks 总线的网关工具的设计[J]. 工矿自动化, 2008 (5).
- [6] 罗隆. 基于 ATmega128 的以太网与 Lon 总线的网关设计与实现[J]. 可编程控制器与工厂自动化, 2006 (10).