

基于树莓派的多串口多总线服务器设计

工业控制中,各设备的信号采集和监控只靠串口总线难以实现扩展,要将现场控制网络和信息网络相连,就需要解决串口通信协议和因特网通信协议的转换问题,即把原有设备转换为具备网络接口的外设,这样可以将传统串行链路上的数据传输到信息网络上,而无需更换原有设备。如此,可以提高原有设备利用率、增加多终端连接数、节约成本、简化布线的复杂度及延长通信距离。

近年来,因信息化和物联网发展的需要,串口服务器大量涌现,它们不占用主机资源,且具有终端服务器的功能。不过,设备体积庞大、价格昂贵、串口不易裁剪或扩展、传输大量数据帧时丢包、参数配置繁杂等问题也随之出现。

本设计采用的串口服务器的核心设备--树莓派,是一款体积小、价格便宜但功能非常强大的平台,可通过多种通信方式接入互联网,支持多种完整网络协议,结合 USB-hub 及 USB/串口转换器使用,可保证串口服务器使用方便,并实现实时、准确、长时间稳定的数据传输。

1 总体设计

根据当今工业现场数据传输需求,应具备的技术指标和设置参数如下。

1.1 技术指标

- ①主控芯片: 32 位 700 MHz 的 ARM1176JZF-S 处理器;
- ②操作系统: Linux;
- ③支持协议: TCP/IP、UDP、HTTP、FTP;
- ④网络接口: 10/100 Mbps 自适应以太网接口;
- ⑤串口类型: 1~16 个 RS 232/RS485/RS422 接口, 1 个 TTL 电平串口;
- ⑥参数配置方式: WEB 浏览器配置;
- ⑦会话数: 支持多连接, 满足 5 个以内用户同时管理一个模块设备;
- ⑧工作模式: TCP 客户端、TCP 服务器、UDP 广播模式;
- ⑨指示灯: 电源指示灯、通信指示灯;
- ⑩供电电压: 5 V。

1.2 设置参数

该设备选择 WEB 浏览器设置方式,内置 WEB 服务器,用户可进行 IP 地址、串口通信参数、工作模式等的管理和配置。以下主要介绍串口通信和工作模式参数的设置。

(1) 串口通信参数

- ①波特率为 2 400~115 200 bps;
- ②数据位为 6/7/8/9;
- ③校验位为 None/Even/Odd;
- ④停止位为 1/1.5/2;
- ⑤成帧长度为 1~15000 字节;
- ⑥成帧间隔为 30~500 位。

其中，成帧长度指每一帧接收数据的长度，接收端缓冲区一旦接收到该长度的数据就转发出去；成帧间隔就是所谓的串口超时，超时时间 $T(s)$ 、成帧间隔 $N(\text{位})$ 、波特率 $B(b/s)$ 的关系如下。一旦超时，不管已有数据长度是不是达到成帧长度，接收端缓冲区就将已有数据组帧转发出去。

$$T = N/B \quad (1)$$

(2) 工作模式参数

- ① 单个串口支持会话数为 1~5 个，每个会话模式间相互独立；
- ② TCP 服务器模式为绑定端口范围为 1025~65535；
- ③ TCP 客户端模式为需配置目标服务器 IP、端口及自动重连时间，其中自动重连时间指网络连接中断或者服务器异常时，重新发起主动连接的时间间隔，循环自动重连直到正常连接上为止；
- ④ UDP 广播模式为需配置本地端口、目标服务器 IP 和目标端口。

2 硬件设计

此串口服务器根据以上指标选用最新的树莓派、USB-hub 及北京世纪联信公司生产的 USB/四串口转换器组合而成。

2.1 硬件结构组成

串口服务器硬件结构组成如图 1 所示，树莓派具有两个 USB 接口、一个 TTL 串口及一个 10/100 MHz 自适应以太网接口。一个 USB 接口通过 USB-hub 和 n (可据情况选择，最多时 $n=4$) 个 USB/串口转换器扩展出 $4n$ 个 RS232/RS422/RS485 串口，另一个可外接无线 USB 网卡，从而实现局域网或广域网中有线或无线数据传输。

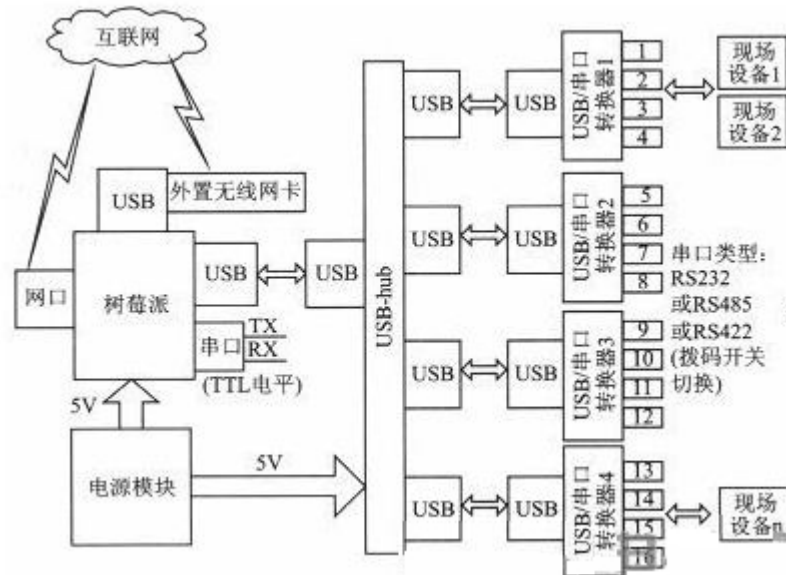


图 1 硬件结构框图

2.2 树莓派简介

树莓派(Raspberry Pi)是一款基于 Linux 系统的只有一张信用卡大小的单板计算机，配备一枚博通生产的 700 MHz ARM 架构 BCM2835 处理器，512 MB 内存，使用 SD 卡当作存储介质，操作系统采用开源的 Linux 系统，提供并支持 Python 作为主要编程语言。

2.3 USB-hub 简介

USB-hub(USB 集线器)是一个将多个 USB 设备连接到计算机上的 USB 接口或另一个 USB 集线器上某时候，它们都来自于主板内部一个或者两个主 USB 接口，而不是相互独立的硬件。一个 USB 接口、一个 USB 集线器和若干个外围设备可以构建一个 USB 网络。本系统选用的 USB-hub 由一个 USB 接口扩展出 4 个 USB 接口。

2.4 USB/串口转换器简介

USB/串口转换器选用北京世纪联信 LENSYS-USB2COM-4 模块。它是一种导轨安装式串口扩展模块，实现了用 USB 口扩展 4 串口的功能，可以通过 USB 快速扩展四个计算机串口，通过拨码开关设定 RS 232/422/485 接口方式，还可以实现 4 个 RS 232 转 4 个 RS422/485 的功能。通过该模块能够可靠、实时、便捷地完成工业现场 RS232/485/422 信号的传输和转换，广泛应用于石油天然气、水利、电力调度、市政调度等行业。

3 软件设计

此串口服务器软件设计基本结构如图 2 所示，客户端只是 WEB 浏览器，软件设计则主要体现为服务器端的网页展现、业务逻辑和数据存储。

WEB 服务器负责串口服务器通信参数的手动配置和存储、通信程序的手动启停等;应用服务器负责实现 TCP/IP 网络与串行接口设备的互通，完成 TCP/IP 协议格式的数据与串行数据间的相互转换;数据库主要包含用户管理数据表、串口参数配置数据表、工作模式配置数据表。

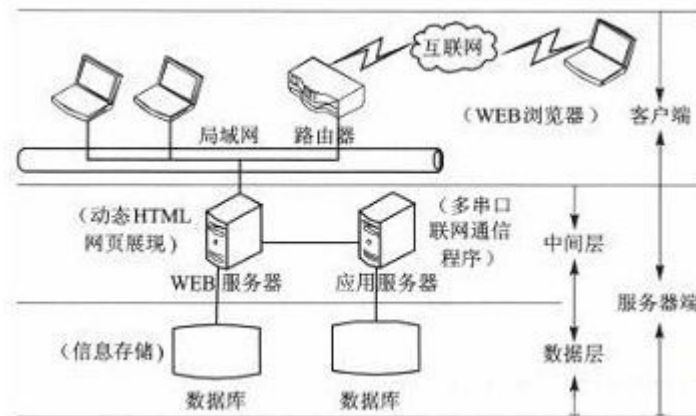


图 2 软件设计基本结构图

下面介绍串口服务器软件设计中 3 个主要任务：WEB 服务器的设计、WEB 配置页面的设计和串口联网通信程序的设计。

3.1 WEB 服务器的设计

由于串口数量较多，所以在树莓派中植入 WEB 服务器，由 WEB 服务器将用户的请求转换为对后台数据的查询或更新，并将友好的图形界面在浏览器上展示给用户，实现简便统一的远程通信参数配置、设备管理、用户管理、监控串口服务器运行状态。

本串口联网服务器选择树莓派自带的 Python 作为开发语言，采用小巧而灵活的 web.py 开发框架提供 HTTP 服务，其轻量级满足采集系统对数据传输模块的要求，可以提高系统的使用效率，WEB 服务器设计流程如图 3 所示。

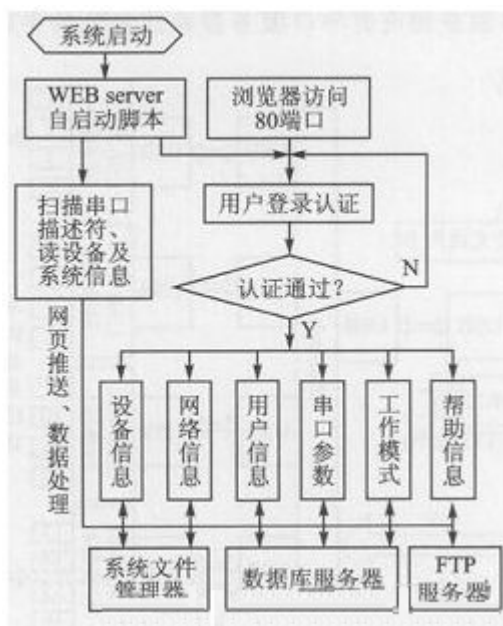


图 3 WEB 服务器设计流程

设备上电或系统重启后，WEB 服务自动启动，并初始化 WEB 服务器所需显示和配置的信息，用户通过浏览器访问系统指定端口进行登录认证即可访问参数配置页面。其中，设备信息和网络信息通过系统文件管理器进行查看和配置，用户信息、串口参数和工作模式信息通过数据库进行读写，帮助信息则通过 FTP 服务器进行串口服务器说明手册、公司服务信息等文档下载。

3.2 WEB 配置页面设计

WEB 配置页面即 WEB 服务器的展示层，如图 4 所示，扫描出的串口数表明该串口服务器带有 16 个串口，每个串口配置相互独立。



图 4 WEB 配置界面

由于配置程序设计采取工作模式配置与串口参数配置相关联，如果选用前 3 个串口，则工作模式配置中只显示此 3 个串口的工作模式配置界面，如图 5 所示。



图 5 工作模式配置界面

以上配置方式可方便实现批量配置，配置完成后可生成配置信息查看界面，如图 6 所示，该界面还可进行选中串口的通信参数、工作模式的编辑和删除。



图 6 配置信息查看界面

3.3 串口联网通信程序设计

WEB 服务器完成配置工作后，即可启动通信程序进行数据通信工作，具体实现流程如图 7 所示。此程序在设备上电或系统重启后自动启动，读取用户配置的串口通信和工作模式参数后，进入通信程序无限循环，图中进程数即现场应用所配置的串口总数。其中 N 为串口服务器串口数， $M(M \leq N)$ 为 N 个串口中使用的串口数； n 为各自编号， l 为进程编号； S 为所配会话数， s 为线程编号。

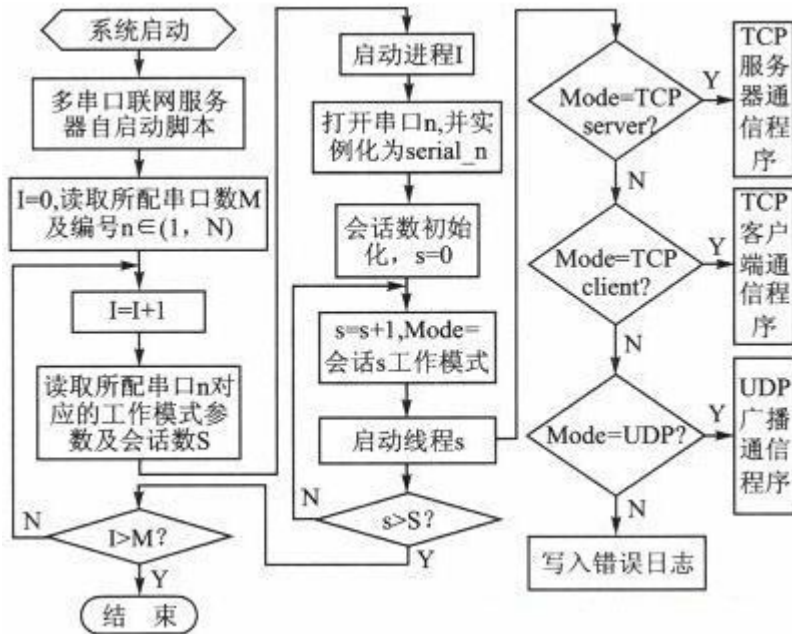


图 7 串口服务器通信程序工作流程

3 种工作模式中，TCP 服务器和 TCP 客户端同属 TCP 协议传输程序，其数据帧收发处理过程都采用同一种思想，下面介绍 TCP、UDP 协议传输程序和数据帧转发程序的设计：

3.3.1 TOP 协议传输程序设计

TCP 协议传输程序涉及到服务器端和客户端的设计。

服务器端程序设计：

- ① 建立一个 **socket**，选择类型 INET 及 TCP 连接方式；
- ② 读取配置端口进行绑定监听，等待客户端主动连接；
- ③ 设置监听队列大小；
- ④ 进入一个无限循环，使用 **accept()** 等待客户连接，返回的新连接对应于客户端 IP，建立通信信道；
- ⑤ 进入无限子循环，通过 **sendall()** 及 **recv()** 进行读写操作。

客户端程序设计：

- ① 建立一个 **socket**，选择类型及连接方式同服务器端；
- ② 读取配置所配置远程服务器 IP 及端口；
- ③ 进入无限循环，使用 **connect()** 连接远程服务器，若连不上，达到所配置超时间隔后再次重连；
- ④ 连接成功后进入无限子循环，通过 **sendall()** 及 **recv()** 进行读写操作。

3.3.2 UDP 协议传输程序设计

UDP 传输称为无连接传输，不存在 TCP 中的三次握手和错误重传机制，其传输程序需同时读取所配置的本地 IP 和端口、远程 IP 和端口，建立数据报形式的 **socket** 后可同时作为发送端和接收端。作为发送端时，**sendto()** 发送地址为远程接收端 IP 和端口，作为接收

端时公开本地 IP 和端口, `recvfrom()` 等待远程发送端的数据到来, 可以接收任何地址发送过来的数据包。

3.3.3 数据帧转发程序设计

串口服务器功能模块实现了串行链路数据与以太网数据转换的功能: 一方面, 接收来自串行链路的数据帧, 并将其转化为以太网链路数据帧后发出; 另一方面, 接收来自以太网链路的数据帧, 并将其转化为串行链路数据帧后发出。以上两个通信过程并行执行, 在程序设计中采取多线程实现方式。

(1) 网络数据的接收

每种通信模式下, 网络数据的接收都在指定的回调函数中实现, TCP 通信接收函数为 `recv()`, 而 UDP 通信中为 `recvfrom()`。当数据帧长度积累到指定的接收缓冲区大小或达到串口超时还不足指定数据帧长度, 立即调用实例化后的串口发送函数 `write()`, 即可将缓冲区中接收到的数据通过串口转发。

(2) 网络数据的发送

接收串口数据时调用串口接收函数 `read()`, 当数据帧长度达到串口接收缓冲区大小或串口超时, 立即调用网络发送函数把该缓冲区中的数据帧通过网口转发, 其中 TCP 通信发送函数为 `sendall()`, 而 UDP 通信为 `sendto()`。

结语

经全面测试合格后, 目前本串口服务器已经成功应用到工业现场, 由此可见, 本串口服务器支持高频转发、大数据帧实时转发, 可长期、全天候稳定运行。