

基于 ARM 的全自动数据采集系统的设计与实现

罗 铿, 平佳伟

(上海航天电子技术研究所, 上海 201100)

摘要: 为满足环境恶劣、人烟稀少地区数据自动采集的要求, 提出并实现了一种基于 ARM 的全自动数据采集系统的方案; 设计的主控板硬件电路以 AM3352 为核心, 配备数字量和模拟量采集模块, 扩展以太网、WIFI、RS232/485 等外设接口; 开发的主控软件在嵌入式 Linux 操作系统中实现全自动数据采集控制, 开发的基于 Android 的手机 APP 可与主控软件实时通信, 设置采集系统的各项参数; 测试结果表明, 开发的全自动数据采集系统具有操作方便、采样精度高以及运行稳定等优点, 可满足实际应用。

关键词: ARM; Linux; Android; APP; 自动数据采集

Design and Implementation of Automatic Data Acquisition System Based on ARM

Luo Keng, Ping Jiawei

(Shanghai Aerospace Electronic Technology Institute, Shanghai 201100, China)

Abstract: A scheme of automatic data acquisition system based on ARM is proposed and realized to meet the requirements of automatic data acquisition in poor environment and sparsely populated areas; The hardware circuit is designed with AM3352 as the core, equipped with digital and analog data acquisition module, extended with Ethernet, WIFI, RS232/485 interface; The developed software in Linux embedded system can realize automatic data acquisition and control; The developed APP based on Android can communicate with the software in real time and set various parameters to acquisition system; The test results show that the developed automatic data acquisition system which is easy to operate, running stable and provides high sampling accuracy, meets the practical application.

Keywords: ARM; Linux; Android; APP; automatic data acquisition

0 引言

某公司需要开发一套数据采集系统, 主要应用于环境监测, 部署在高山、冰川、海岛、沙漠、原始森林等环境恶劣、人烟稀少的地区, 实现区域内监测数据的采集、处理及存储, 并将数据发送至卫星。由于部署环境偏僻, 不方便维护和修改配置, 该系统必须具有很高的可靠性, 很强的错误处理能力, 断电自重启、自运行等功能。

为了满足该项目的需求, 采用以 ARM 处理器为核心, 结合移植的 Linux 操作系统, 开发了一套集嵌入式控制、高速数据采集、实时处理于一体的数据采集系统。该系统具有 16 路同步 16bit 精度模拟采集通道, 16 路数字量采集通道, 以及 RS232 和 RS485 等接口用于外接其他传感器或设备, 能对各类现场信号进行采集、分析处理和存储; 该系统具有上电自运行, 出错自重启等错误处理和自恢复功能; 该系统仅在用户设定的时间段内启动各功能模块进行数据采集、发送, 其他大部分时间处于待机状态能够保持较低的功耗。

同时开发 APP 软件, 可通过移动客户端与系统连接, 实现参数实时设置、数据实时显示等功能, 在偏僻地区部署系统时只需携带手机即可对系统进行重新设置, 满足系统的便携性需求。

1 系统框架设计

系统主要由通信模块、主控板以及客户端组成, 框架如图 1 所示。

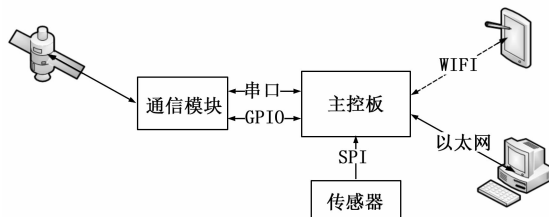


图 1 系统框架

主控板为整套系统的核心, 搭载板上系统, 运行主控软件, 接收来自传感器的监测数据, 与通信模块进行实时交互, 与后端计算机或移动客户端进行实时通信; 通信模块接收主控板的指令, 向卫星发送数据; 计算机或移动客户端接收主控板的数据进行实时显示, 并实时设置主控板的各项参数。

2 硬件设计

为满足项目需求, 以 AM3352 为核心设计了主控板硬件电路^[1-3], 同时包括电源、复位电路、时钟电路、128 MB DDR2 SDRAM、256 MB NAND FLASH、SD 卡接口、RS232 接口、RS485 接口、以太网接口、WIFI 模块、AD 采集模块、数字 IO 模块, 框架如图 2 所示。

微处理器采用 TI 公司的 Cortex-A8 32 位 RISC 处理器 AM3352, 工作频率为 800 MHz, 最高频率达到 1 GHz, 具有

收稿日期: 2015-10-11; 修回日期: 2015-11-09。

作者简介: 罗 铿(1986-), 男, 浙江舟山人, 硕士研究生, 主管设计师, 主要从事嵌入式测控自动化设备方向的研究。

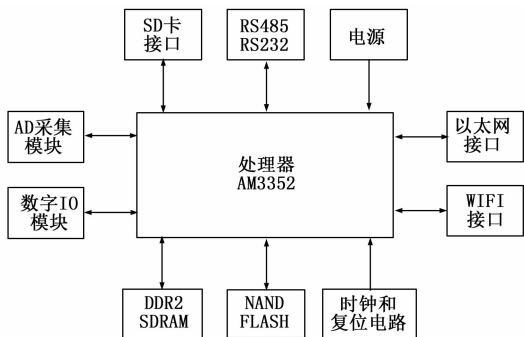


图 2 主控板硬件框架

NEON SIMD 协处理器、单错检测（奇偶校验）的 32 KB/32 KB L1 指令/数据高速缓存、含纠错码（ECC）的 256 KB L2 高速缓存，支持 Linux 操作系统，具有丰富的外设接口，使得这款处理器在兼备高效处理能力、可靠性以及低功耗。

2.1 AD 模块

AD 模块由多路复用电路、衰减电路、AD 电路组成，其原理如图 3 所示。

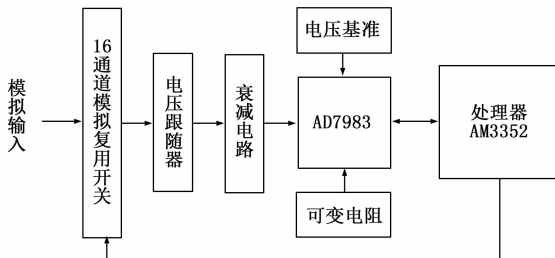


图 3 AD 模块框图

AD 芯片采用 ADI 公司的 16 位单通道采样芯片 AD7983，1.33 MSPS 吞吐速率，伪差分模拟输入范围 0 V 到 VREF，支持 SPI 串行接口，典型值 10.5 mW 低功耗，采用 2.5 V 单电源供电，兼容多种电平逻辑接口。如图 4 所示，AD7983 的基准电压为 4 V，其满量程范围为 0~4 V，故前段经过衰减电路将输入电压范围由 -5~+5 V 调整为 0~4 V，装换公式为 $V_{ad} = \frac{2}{5}V_{in} + 2$ 。

SPI 控制接口采用三线式，将引脚 SDI 拉高；CS 片选信号连接引脚 CNV，上升沿启动转换，但 CNV 必须在最小转换时间逝去前返回低电平，启动转换后，并强制 SDO 进入高阻态，直至转换完成，SDO 从高阻态变为低阻态，数据位则在

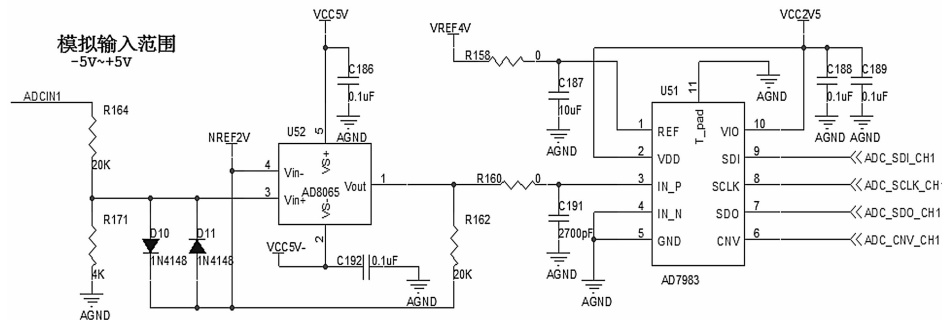


图 4 AD 电路原理图

随后的 SCK 下降沿逐个输出，MSB 优先，时序如图 5 所示。

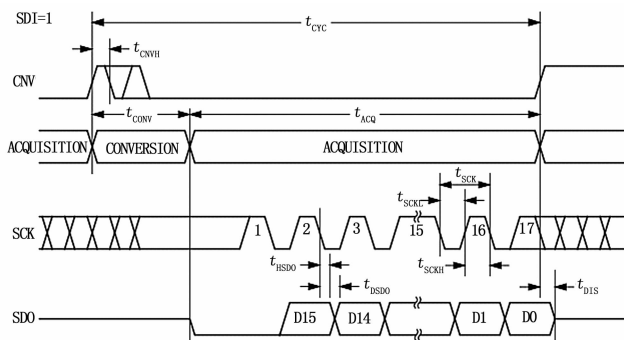


图 5 AD 数据采集时序图

2.2 WIFI 模块

WIFI 模块采用小型 WIFI232 模块，支持 802.11 b/g/n 无线标准，UART 数据通讯接口，多种工作模式可配置，内置天线，具有超小尺寸低功耗的特点。

WIFI 模块采用 MTK 公司的 802.11 b/g/n 标准 WIFI 芯片 MT5931，集成功率放大器（PA）、低噪声放大器（LNA）及收/发开关（T/R Switch）于单芯片，支持最先进的 Wi-Fi 规格，包括 STBC、Wi-Fi Direct 以及 Hotspot 模式。WIFI 芯片通过单片机扩展 UART 接口，以便于接入其他控制器，简化通信接口驱动，可直接使用异步串行通信。连接引脚 VCC、GND、RXD 和 TXD。

2.3 以太网模块

以太网模块采用 MICREL 公司的千兆以太网 PHY 芯片 KSZ9031RNI，该芯片支持 10 M/100 M/1000 Mbps 以太网收发，提供 RGMII 接口，使用 3.3 V 和 1.2 V 电源供电，网络插座采用内置网络变压器的 HY91130AE。以太网模块原理如图 6 所示。

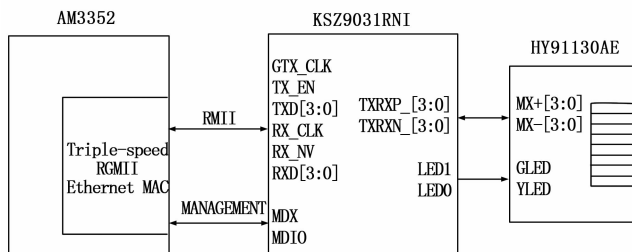


图 6 以太网电路原理图

3 软件设计

全自动数据采集系统主要包括 3 个软件：主控软件、客户端软件、App 软件。

主控软件运行在主控板上，是整个系统的核心。客户端软件可运行在 Windows 系统、Linux 系统的计算机上，主要用于设置主控软件的各个参数并显示接收到的数据。App 软件可运行在 Android 系统的手机上，实现与客户端软件相同的功能，同时具备更高的灵活性与便携性。主控软件和

客户端软件、APP 软件通过网络连接, 可实时通信, 延时在 1 秒内。

3.1 主控软件设计

开发环境: 以一台安装 Ubuntu12.04 桌面版系统的计算机作为宿主开发环境, 通过网线连接移植了 Linux3.2.0 系统的主控板, 使用 NFS 来挂载共享目录。在宿主机安装 QT4.7 开发工具, 采用交叉编译工具链 arm-none-linux-gnueabi-g++, 编写完成的程序通过共享目录拷贝到主控板上, 在主控板上进行调试^[4]。

主控软件包含主程序和 AD 子程序、参数子程序、通信子程序 3 个子程序。

1) 主程序负责完成逻辑控制, 实时检测是否有新的参数传入并设置, 配置通信模块的各项参数, 在到达设定的采样时间调用 AD 子程序, 到达设定的发送时间调用通信子程序等功能;

2) 参数子程序与上位机软件建立网络连接, 实时获得设置的参数, 并保存在 SD 卡中;

3) AD 子程序通过控制主控板上外接的 AD 模块获得采样数据, 并保存在 SD 卡中;

4) 通信子程序接收来自通信模块的数据, 解析 GPS 时间并完成校时;

主控软件的运行流程如图 7 所示。

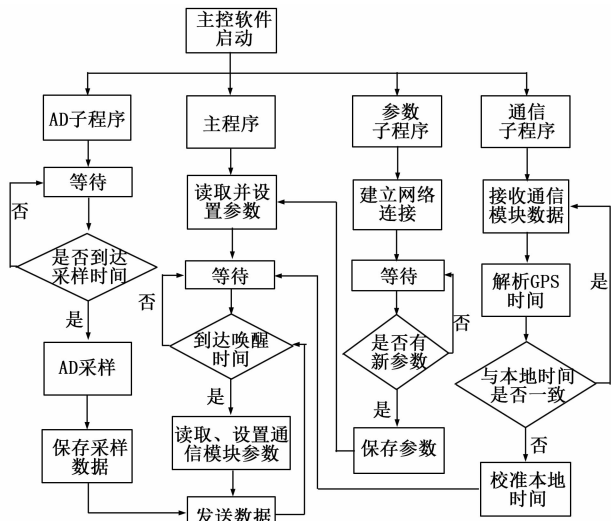


图 7 运行流程图

主控软件可在主控板上电时自动启动, 方法是在主控板的 Linux 系统目录/etc/init.d/下面增加添加自命名为 S91dcp.sh 的脚本文件, 文件内容如下:

```
! /bin/sh
/home/build-Debug_m3352-ARMQT-Debug/Debug_m3352 &
```

其中第二行就是指主控软件在系统中的实际位置。

3.2 客户端软件设计

客户端软件通过 TCP/IP 协议与主控软件建立连接进行通信, 本文将主控软件设计为 Server 端, 将客户端软件设计为 Client 端, 主控软件运行时, 客户端软件可随时连接或断开连接。

开发完成的客户端软件具有与主控板实时通信、数据包收

发、通信模块状态实时显示、AD 采样数据实时显示、参数实时设置等功能, 界面如图 8 所示。

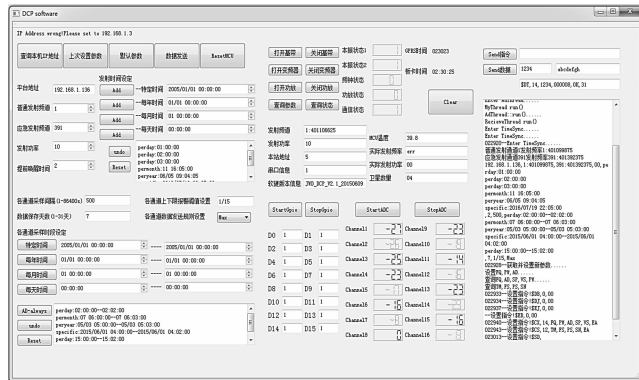


图 8 客户端软件界面图

用户在客户端软件上设置了平台地址、发射频道、发射频率、发射功率、发射时间、数据保存时间、AD 采样时间段等多项参数后, 将这些参数整合成一个数据包发送给主控板, 主控软件对数据包进行解析并实时设置。客户端软件收到主控软件发来的数据包, 解析后实时显示各个模块的数据, 同时, 客户端软件可以发送控制指令到主控软件, 再由主控软件转设置通信模块, 实现对通信模块的实时控制。整个客户端软件功能如图 9 所示。

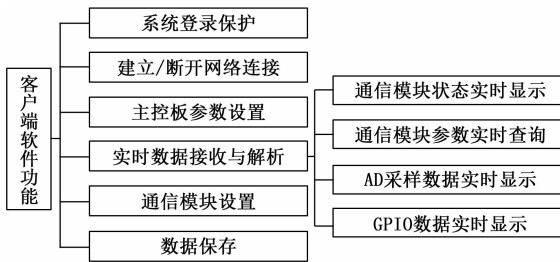


图 9 客户端功能图

主控软件将所有信息组成一个包, 每 1 秒向客户端软件发送一次, 客户端软件收到这个数据包后, 根据两个软件的通信协议(见表 1)解析出各个模块的数据。

表 1 主控软件与客户端软件通信协议

帧头	长度/字节	对应内容
BoardTime;	8	主控板本地时间
Gpio;	10	时钟、频钟、功放等状态
GPOTAG;	40	16 路 TTL 接口采样数据
ADCvalue;	100	16 路 AD 模块采样数据
\$ GPRMC	6	卫星时间
\$ DT	25	通信模块返回数据
\$ WS	50	通信模块参数 1
\$ RP	50	通信模块参数 2

3.3 APP 软件设计

项目在 Android 手机上开发了相应的 APP 软件, 可无线连接到主控板, 实现远程控制与显示, 大幅度提升系统的灵活性与便携性。

APP 软件开发的方式与主控软件开发方式类似, 以一台安装 Windows7 操作系统的计算机作为宿主开发环境, 通过 USB 线连接 Android 手机。在宿主机中安装 QT5.2 开发工具,

安装并配置 JDK1.8、Apache1.9、Android SDK、Android NDK 这 4 个软件，配置完成后即可编译生成后缀为 Apk 的 Android 程序，然后在手机上进行调试^[5-6]。

项目采用的软件开发工具 QT 是一种跨平台的开发工具，因此只需对客户端软件进行部分修改，绘制 APP 软件界面，在配置好的环境下重新编译，即可生成 APP 软件。项目开发的 APP 软件具有与客户端软件完全相同的功能，由于手机屏幕较小无法同时显示所有功能，分别绘制了参数设置界面、参数和状态查询界面、实时通信界面，这 3 个界面可实时切换，便于用户使用。

4 系统测试

本文设计与实现的系统需要在恶劣环境下长期工作，为了验证系统的可靠性和实用性，进行了大量的数据采集测试、应急模式测试和稳定性测试。

4.1 数据采集测试

利用标准信号源提供信号，接到 ADC 的 16 个通道上，通过主控板的 AD 子程序读取采样数据，转发到客户端软件进行显示，结果如表 2 所示。

表 2 ADC 数据采集测试结果 V

真实值	测量值	误差
-5	-4.994	0.006
-4	-3.998	0.002
-3	-2.997	0.003
-2	-1.998	0.002
-1	-0.998	0.002
0	0	0
1	0.998	0.002
2	1.998	0.002
3	2.997	0.003
4	3.997	0.003
5	4.993	0.007

由表 2 可以看出，获得的采样值与标准信号源的值误差在 0.01V 以内，采样率误差较小，系统可以准确地采集数据，且数据转换精度高、误差小，符合使用要求。

4.2 应急模式测试

应急模式是系统监测到某个采样值超过设定的阈值时紧急启动的一种机制，触发后立刻进入发射模式，将当天的采样数

据发送出来。

模拟实际情况进行应急模式测试，在系统平稳运行采集数据时，给通道一个超过阈值的信号，观察客户端软件发现系统立刻进入了应急模式，完成了数据发送，事后通过回看系统日志也证实在该时刻系统进入了应急模式。在不同采样时间段分别进行应急模式测试，从触发阈值到进入应急模式均在 1 秒内，大量测试表明应急模式满足实际应用需求。

4.3 系统稳定性测试

稳定性是系统最重要的指标，设置好各项参数后，启动系统并进行了 72 小时不间断测试，在每天固定时间启动 AD 子程序进行数据采集，每天固定时间进行数据发送，每 1 小时进行一次系统校时确保与卫星时间同步。

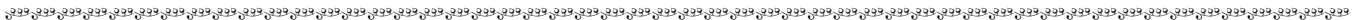
通过分析操作日志以及观察客户端的实时数据，发现系统运行稳定，功能完善，在 72 小时内未出现错误，达到了预期的效果。

5 结论

本文提出了一种基于 ARM 和 Linux 的通用数据采集系统方案，设计了主控板卡硬件电路，搭建了嵌入式 Linux 系统采集平台，实现了 16 路精确定时模拟量采样、16 路数字量采样，具备 RS232/485 接口、以太网通信功能，开发的系统软件运行稳定，客户端软件具备良好人机交互界面，扩展的手机 APP 软件功能完善，组建完成的系统功能完善、运行稳定，在工农业生产监控、地质水文环境监测、森林灾害预防等领域有着广阔的应用前景。

参考文献:

[1] 柯新宇. 基于 ARM 的数据采集卡研制 [D]. 武汉: 华中科技大学, 2008.
 [2] 韩雪川. 基于 ARM 嵌入式 Linux 的数据采集监控终端设计 [D]. 哈尔滨: 哈尔滨工程大学, 2010.
 [3] 闫广续, 袁纵横, 等. 基于 ARM 嵌入式 Linux 的数据采集系统设计 [J]. 计算机测量与控制, 2015, 23 (5): 1724-1727.
 [4] 陆文周. Qt5 开发及实例 [M]. 北京: 电子工业出版社, 2014.
 [5] 王 森. 一种基于 Android 的远程控制工具的设计与实现 [D]. 西安: 西安电子科技大学, 2012.
 [6] 王 峰, 宣伯凯, 等. 基于 Android 的家庭移动医疗监护系统的设计 [J]. 计算机测量与控制, 2015, 23 (5): 1586-1588.



(上接第 158 页)

[9] 吴 迪, 曹 洁, 王进花. 基于自适应高斯混合模型与静态听觉特征融合的说话人识别 [J]. 光学精密工程, 2013, 21 (6): 1598-1604.
 [10] 马义德, 邱秀清, 陈昱莅, 等. 改进的基于听觉掩蔽特性的语音增强 [J]. 电子科技大学学报, 2008, 37 (2): 255-25.
 [11] 茅正冲, 王正创, 龚 熙. 一种低信噪比下的说话人识别算法研究 [J]. 计算机应用与软件, 2014, 31 (12): 218-220, 251.
 [12] Kitaoka N, Yamamoto K, Kusamizu T, et al.. Development of VAD evaluation framework CENSREC-1-C and investigation of relationship between VAD and speech recognition performance [A]. Automatic Speech Recognition & Understanding [C], Kyoto,

Japan, 2007: 607-612.
 [13] 李振静, 王国胤, 杨 勇, 等. 基于谱熵噪声估计的改进谱减法 [J]. 计算机工程, 2009, 35 (18): 164-166.
 [14] 杜志然, 周 萍, 景新幸, 等. 基于谱熵的耳语音增强研究 [J]. 传感器与微系统, 2012, 31 (6): 69-72.
 [15] 蒋 晔, 唐振民. GMM 文本无关的说话人识别系统研究 [J]. 计算机工程与应用, 2010, 46 (11): 179-182.
 [16] 赵立辉, 毛 竹, 霍春宝, 等. 基于 GMM-SVM 的说话人识别系统研究 [J]. 工矿自动化, 2014, 40 (5): 49-53.
 [17] 侯 珏, 刘 轶, 郑 方, 等. 基于 VP 树结构的多层匹配算法在哼唱识别中的应用 [J]. 清华大学学报 (自然科学版), 2009, 49 (S1): 1419-1424.