

基于 DM3730 的阳光长跑终端系统^①

龚泽挚, 郑雅羽, 俞帅东

(浙江工业大学 信息工程学院, 杭州 310000)

摘要: 为解决高校传统长跑具有管理、监督困难, 学生参与积极性低等问题, 设计和实现了一种嵌入式阳光长跑终端系统. 系统硬件平台选用 TI 的基于 ARM 架构的 DM3730 处理器, 以 Linux 操作系统作为软件平台. 提出一种基于身份证和指纹的二次身份识别技术, 基于 Qt 设计人机交互界面, 使用 SQLite 数据库记录学生长跑信息, 采用 TCP 协议实现长跑信息与远程服务器交互. 系统测试表明: 学生身份识别速度快、准确率高, 人机交互友好, 使得长跑管理智能化、学生参与自主化, 具有很高的应用前景和市场价值.

关键词: 阳光长跑; DM3730; Linux; 身份识别; Qt; SQLite; TCP

Sunshine Long-distance Race Terminal System Base on DM3730

GONG Ze-Zhi, ZHENG Ya-Yu, YU Shuai-Dong

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310000, China)

Abstract: To solve some problems of traditional long-distance race, such the difficulty of management and supervision, the low enthusiasm of students, an embedded sunshine long-distance race terminal system is designed and implemented. DM3730 processor based on ARM is chose as the hardware platform and Linux is used as software platform. A new technology about twice identification based on identity card and fingerprint is presented, user running information is stored in SQLite databases, interaction with the remote server is based on Transmission Control Protocol(TCP), and Qt is applied to construct Graphics User Interface(GUI). The testing results show that system has a high speed and precision of identification, an operation of GUI is also very friendly, management gets intelligent, running is held without supervision at any time. Higher application prospect and market value is expected.

Key words: sunshine long-distance race; DM3730; Linux; identification; Qt; SQLite; TCP

1 引言

近年来, 学生体质健康呈现严重下降趋势. 教育部从 2007 年起开展了全国亿万学生阳光体育长跑活动. 实践发现, 当前的长跑管理模式都存在着人力资源消耗大, 管理、监督困难, 学生积极性低等问题^[1,2]. 为此, 在 2010 年引进了一套“校园阳光长跑智能自主管理系统”终端机, 通过刷指纹来记录学生长跑信息. 然而, 在实际运行过程中该机易出现诸如: 死机、身份识别出错等各种问题, 导致其最终被市场淘汰.

为解决当前长跑终端系统存在的普遍问题, 本文设计并实现了一种基于嵌入式 ARM 的阳光长跑终端系统. 选择 TI 的基于 ARM 架构的 DM3730 高性能处

理器作为硬件平台, 以 Linux2.6.32 作为系统的软件平台. 提出了一种全新的基于身份证和指纹的二次身份识别技术, 利用身份证刷卡模块提取 ID, 通过 ID 直接获取学生已存指纹信息, 并完成与当前提取的指纹信息的比对, 真正做到二次身份识别.

同时, 为实现智能化管理与监督, 在系统中使用 SQLite 数据库实现对当前节点学生长跑信息的本地存储, 利用 TCP 协议完成与远程服务器对长跑信息的交互. 为给予学生更好的用户体验, 基于 Qt 技术开发了友好的用户界面, 包括学生信息界面和管理员界面, 并带有语音提示功能. 系统不仅具有日志管理方式, 方便设备异常时的维护, 还具有死机自启动功能, 避

^① 收稿时间:2013-07-25;收到修改稿时间:2013-08-21

免设备一直处于非工作状态。

本文将主要介绍该终端系统核心软件的设计与实现, 并给出系统实际运行时的测试结果。

2 系统硬件架构设计

系统硬件架构如图 1 所示, 将 TI 的 DM3730 双核 (ARM Cortex-A8 Core 和 C64x DSP Core) 高性能处理器作为硬件核心, 其 ARM 端的工作主频可以达到 1GHz, 为系统运行提供了高速保障^[3]。外围主要由数据存储模块、指纹采集模块、身份证 ID 提取模块、显示模块、以及串口、USB 等通信模块组成。

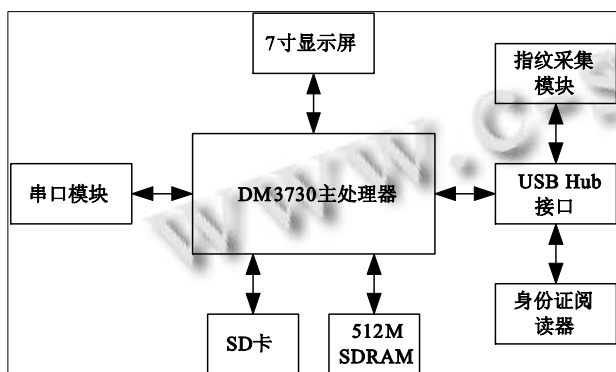


图 1 终端系统硬件结构图

3 系统软件设计与实现

3.1 嵌入式 Linux 软件开发平台搭建

嵌入式 Linux 操作系统要正常运行需要包括 Bootloader、Linux kernel 和文件系统三个部分^[4]。TI 针对 DM3730 处理器已经提供了开源的软件开发包 (DVSDK)。本文以 TI 的 DVSDK4.0.1 为基础, 对 x-loader、U-boot、Linux kernel 和文件系统做部分修改和移植, 并在 Ubuntu11.04 下执行相应命令完成源码的配置与编译, 依次生成对应的镜像文件 MLO、uboot.bin、uImage 和 rootfs.tar.gz, 并将其制作成 SD 卡。

至此, 整个长跑终端的操作系统环境和交叉编译环境搭建完成, 为下一步应用程序的开发打下基础。

3.2 终端系统应用软件架构设计

针对长跑系统的业务需求, 本文对系统进行了模块化的设计。Linux 是一个支持多任务的操作系统, 支持多进程和多线程, 本文将每个核心模块设计成一个独立的进程。

阳光长跑终端系统软件的总体架构设计如图 2 所示, 其主要有以下核心进程组成: 系统进程、指纹进

程、身份证刷卡功能、身份认证功能以及 Qt 进程。

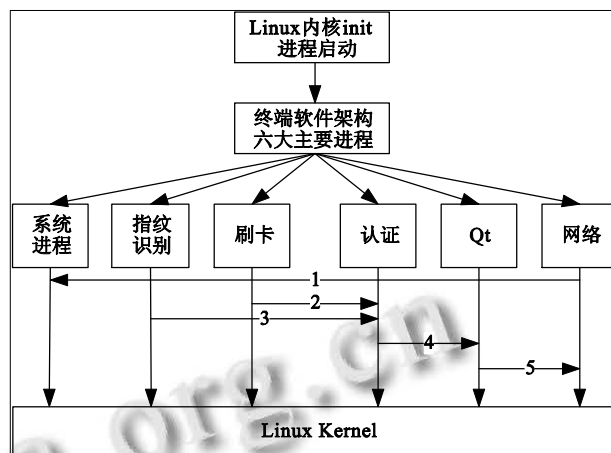


图 2 系统软件总体架构图

3.3 核心软件模块的实现

3.3.1 Linux 下进程间通信

对于 Linux 这种支持多进程的系统, 实现进程之间的通信 (IPC) 是非常重要的, 甚至是必不可少的, 比如本文中提到的身份证刷卡进程会将提取的 ID 发往认证进程, 就需要使用到进程间通信。Linux 中常用进程间通信方式 (IPC) 有: 管道 (Pipe)、信号 (Signal)、消息队列 (Message Queue)、共享内存 (Shared Memory)、信号量 (Semaphore) 以及套接字 (Socket)^[5]。本文主要基于共享内存和 Unix 套接字作为进程间通信的主要方式, 并对其做进一步的封装以适合本终端系统的使用, 其中信号量又被用于实现进程间的同步与互斥。

(1) 基于共享内存的通信方式

共享内存方式是 Linux 中非常高效的进程间通信方式, 由于多个进程共享同一片内存区域, 每当有进程改变共享内存区域的数据时, 就相当于直接改变了共享该内存区域的其他进程的数据, 不需要再进行拷贝或者其他操作。

本文中, 共享内存维护的数据主要包括静态参数和动态参数, 静态参数如终端设备 IP、ID 等。动态参数用于记录进程运行的状态信息。本文设计结构体 ShmParamComm_group 用来表示进程间的共享内存区。

(2) 基于本地 Socket 的通信方式

本地 Socket 也称为 Unix LocalSocket, 是一种数据量小、发送频率低、全双工、一对多的通信方式。本

地 Socket 的服务端接收数据, 客户端发送数据^[6]. 本文为六个进程各自分别建立一个 Socket 服务器端和多个 Socket 客户端, 类似于 C/S 架构. 对任意的 Socket 客户端和服务端都以结构体进行封装, 分别定义为 UCSocket 和 USSocket. 在当前进程中对上述的服务器和多个客户端结构体进行再封装为 IPC_USocketSet, 每个进程通过该 Socket 集合来保持整个进程与其余进程间的通信. 进程间数据流的交互收发通过 recvfrom 和 sendto 函数来完成, 数据交互如图 3 所示.

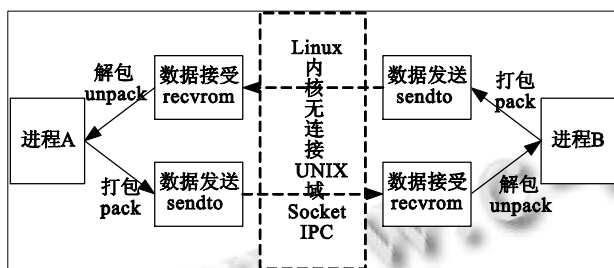


图 3 Socket 本地进程间的数据交互协议

其次, 在 Socket 进程间通信的模块中, 往往由于多个客户端在向服务器发送数据包, 因此服务器需要通过解析数据包来获取客户端进程的 ID 以及相关的数据类型.

基于共享内存和本地 Socket 的进程间通信方式的实现为开发其余进程打下了基础.

3.3.2 系统进程

本文中的系统进程 System Process, 主要实现以下两部分内容:

(1) 创建了系统日志文件, 系统日志文件主要用于记录当前终端设备运行过程中产生的特殊信息, 以及设备出现异常时的错误信息. 系统日志文件方便开发人员对设备进行维护与更新. 本文采用以下宏定义来实现整个日志文件的创建以及写操作.

(2) 本文基于共享内存的通信方式, 实现进程心跳包检测机制. 系统进程负责对其余进程进行心跳监听, 使得系统遇到异常情况时可以自启动, 不会让设备一直处于非工作状态.

3.3.3 身份识别技术的实现

本文采用基于身份证刷卡和指纹识别的二次身份识别技术, 以确保用户身份识别做到 0 失误, 从而进一步确保用户长跑信息的准确度. 本文中的身份识别过程主要包括指纹采集进程、身份证 ID 提取进程以及

最终的身份认证进程.

(1) 身份证 ID 提取进程

身份证阅读器作为系统的外设, 通过 USB 转串口与主处理器进行交互. 在 Linux 系统中, “万物皆文件”^[7], 本系统中该 USB 转串口设备对应的设备文件为 /dev/ttyUSB0. 阅读器的操作流程如下所示:

① 阅读器初始化函数 init_serial(): 主要完成 open 设备文件, 设置串口波特率为 115200bsp, 数据返回模式为单字节.

② 查询卡是否存在的函数 SAM_find_card(): 向阅读器依次发送命令字 0x20、0x01, 使设备一直处于读卡状态. 如果有卡识别到则返回身份证卡的 SIIN 号, 否则就一直查询.

③ 读取身份证 ID 的函数 SAM_get_user_info(): 向阅读器依次发送命令字 0x30、0x01, 阅读器将以结构体 IDCardData 的封装形式返回用户身份证的所有信息, 从该结构体中可以提取到卡的 ID, 由于 ID 号是采用 Unicode 编码, 还需要转为 ASCII 码.

④ 在读取到身份证 ID 后, 将通过本地 Socket 进程间通信方式将 ID 发往认证进程, 需要调用函数 client_send_msg()来实现.

身份证 ID 提取进程会一直处于 While 循环中, 并做心跳包加 1 操作, 实时查询是否有卡可读.

(2) 指纹采集进程

指纹信息采集器通过 USB 接口与主处理器直接通信, 本系统统一采用右手中指作为指纹信息采集点. 指纹原始图像信息一般数据量较大且信息冗余过多, 需要对原始图像作进一步处理获得其特征节点(Node)的特征数据(Feature Date)^[8,9]. 本文指纹节点数最大为 68 个, 指纹特征数据最大可占 408Byte 的存储空间. 指纹采集过程主要分为以下步骤:

① 指纹采集器的相关硬件初始化 dpfpdd_init(), 调用 get_reader_name()返回一个操作 Handle(句柄).

② 指纹捕获函数 finger_capture(), 其主要通过调用 get_pimage_data()函数完成对指纹图像数据的捕获以及 get_fmd()函数完成特征数据的提取.

③ 指纹特征数据的提取成功后, 依旧调用 client_send_msg()函数发往数据到身份认证进程, 做进一步处理.

同理, 该进程也会一直处于 While 循环中, 并做心跳包加 1 操作, 实时等待有指纹信息可以捕获.

(3) 身份认证进程

身份认证进程, 作为一个 Socket 本地 Server, 在轮询接收本地其他进程 Client 发来的数据信息, 包括用户的身份证 ID 号和指纹特征数据. 在该进程中实现身份二次识别认证技术, 其详细实现原理如下:

① 身份认证进程使用 `Usr_Info` 结构体维护一个用户的长跑节点信息, 包括指纹信息、身份证 ID、认证通过时的时间信息以及认证标志位.

② 本文设计的身份认证过程需要严格按照先刷卡, 后指纹的操作流程. 但是用户往往存在一定的操作失误, 本文根据实际情况总结出如图 4 所示的用户可能的操作状态流程图.

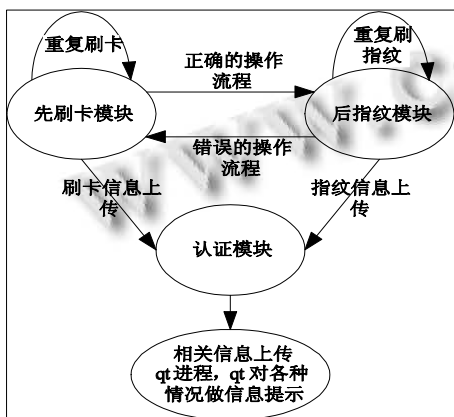


图 4 用户操作各种状态流程图

在正确完成当前用户身份证 ID 和指纹信息的提取之后, 调用 SQLite 数据库的 API 函数, 通过用户 ID 获取本地数据库的指纹信息, 调用 `finger_compare()` 函数确认是否与当前捕获的指纹信息相匹配. 最后, 用户长跑信息将以 `Usr_info` 结构体的数据封装形式通过 Socket 本地进程间通信方式发送到 Qt 进程.

对于图 5 中可能存在的误操作过程, 本文设计出如表 1 所示的用户操作相应提示信息, 其中 `Usr_info` 中的 `set_flag`(认证状态标志位)用来标定当前操作流程的模式, Qt 进程将依据此标志位给出相应的界面与语音提示, 以帮助用户完成一次正确的操作.

表 1 用户操作相应提示信息

操作流程	相关提示语	认证标志位 <code>set_flag</code>
刷卡→指纹	提取成功, 谢谢	<code>VERIFY_ALL_SUCCESS</code>
刷卡→刷卡	请再刷指纹, 谢谢	<code>VERIFY_CARD_REPEAT</code>
指纹→刷卡	请先刷卡, 谢谢	<code>VERIFY_FINGER_REPEAT</code>
指纹→指纹	请先刷卡, 谢谢	<code>VERIFY_FINGER_REPEAT</code>



图 5 基于 Qt 设计的用户界面

本文实现的二次身份识别技术, 不但保证了身份识别的速度, 也大大降低了身份识别的错误率, 友好的操作提示信息也解决了由于误操作而引起的用户长跑信息记录错误.

4 基于Qt的软件模块实现

4.1 人机交互界面

Qt 是诺基亚开发的一种跨平台的 C++ 图形用户界面应用程序框架, 其使用信号与槽(signal/slot)的对象间通信机制^[10], 本文选用的 Qt 版本号为 4.6.2.

本文开发的人机交互界面分为用户长跑信息显示界面和管理员界面. 前者用于显示当前用户在该节点处的长跑信息以及相关操作的提示信息, 如图 5 所示, 其中操作提示信息通过 Socket 本地进程间通信方式从身份认证进程获得. 而后者主要用于设置相关终端设备参数, 查询和管理用户信息, 修改设备登陆密码, 如图 6 所示.



图 6 基于 Qt 设计的管理员界面

4.2 基于 Qt 的 SQLite 数据库

SQLite 是一种常用于嵌入式 Linux 设备的轻量级

数据库,它所占用的资源非常低^[11].而Qt对该数据库做了QtSql类的封装,加快了对数据库业务的开发.

本文创建一个SQLite数据库data.db,该数据库主要用于存储用户的长跑信息以及设备信息.在该数据库中需要建立四张数据表:用户信息表user,管理员信息表manger,终端设备公共参数选项表com_param,终端私有属性参数表pri_param.

Qt对数据库的操作主要由QSqlQuery类来完成,调用其成员函数执行SQLite语句完成插入、更新和删除等操作.

```
QSqlQuery query;
query.exec(QString("insert into student (id,name) values
(3,ChenYun)"));
query.exec(QString("update student set name ='xiaogang'
where id = 0"));
query.exec(QString("delete from student where id = 3"));
```

4.3 基于Qt的网络通信

Qt对网络通信做了QtNetwork类的封装,其使用TCP协议^[12]用于完成数据的传输,根据自己的网络通信业务需求,需要基于该协议设计适合于本系统的数据包格式,类似于HTTP.本文制定的数据包协议由一个包头Header和一个xml数据文件组成,其中Header中定义如下3个请求命令字:

CMD_INIT: 请求远程服务器传输数据库信息,用于完成终端设备的本地数据库初始化;

CMD_PUT: 请求向服务器发送当前用户长跑节点信息,写入服务器端的数据库;

CMD_UPDATE: 系统在每日凌晨会主动请求服务器完成本地数据库信息的更新.

Qt网络客户端的创建以及与远程服务器的交互只需实现基于TCP的TcpServer类.TcpServer创建的客户端会向服务器发送自定义的数据包请求事件处理,然后开始监听服务器端的socket连接.当Qt客户端监听到服务器端发回数据信息后调用TcpThread类建立网络通信进程,通过TcpThread将接收到的信息通过信号和槽的机制传递给GUI主线程,主线程对数据做进一步解析与操作,如实现本地的SQLite数据库的同步等.

5 系统测试及应用情况

在完成以上所有核心软件模块的编码后,需要编

写Makefile将各个源代码通过交叉编译器编译为可以在DM3730上运行的可执行文件.编译出来的可执行文件分别为:card、finger、verify、qt以及sys,考虑到各个进程需要满足一定的启动顺序,故编写进程启动脚本/etc/init.d/app_start.sh,如下所示:

```
#!/bin/sh
ulimit -c unlimited
/usr/share/run_bin/sys &
/usr/share/run_bin/qt -qws &
/usr/share/run_bin/verify &
/usr/share/run_bin/card &
/usr/share/run_bin/finger &
```

本文设计的长跑终端系统实际测试运行时的人机交互界面分别如图7、8所示.其中图7显示的是管理员操作界面,图8显示的是当前学生长跑到此终端处的时间信息.



图7 实际运行时的管理员操作显示界面

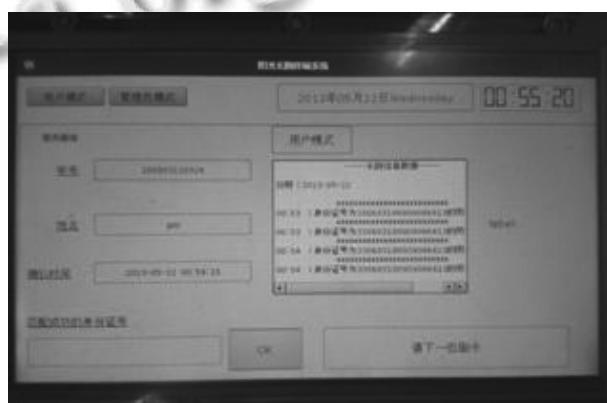


图8 实际运行时的学生长跑信息显示界面

其中,为验证本系统具有快速、准确的身份识别特性,对本地SQLite数据库已存储1万条学生身份信息

在识别学生身份的过程中分别测试其身份证刷卡时间 T1、指纹提取时间 T2 以及一次完整的身份识别总时间 T。其中表 2 给出的是 10 组对应的测试结果。

表 2 身份识别各过程的时间消耗表(单位: 毫秒)

T1	T2	T
641.1	2555.1	6313.5
641.3	2555.9	6401.2
641.3	2535.1	6545.7
641.3	2507.7	6397.2
641.4	2566.2	6604.9
641.4	2565.9	6814.3
641.4	2559.2	6544.5
641.4	2543.6	6504.0
631.4	2557.8	6224.1
641.6	2502.9	6514.1

以上测试数据结果表明本系统各模块运行相对稳定, 学生完成一次身份识别的过程平均耗时约为 6.5 秒, 实现了快速识别的特性。其次, 经重复测试多次后, 统计出系统身份识别的准确率达到 99.9%, 实现了高准确率的识别特性。

本文开发的阳光长跑终端系统已与校体军部相合作进行了 3000 米长跑的试运行, 在学校 4 个不同的测试点分别放置 4 台长跑终端设备。在实际运行的过程中, 基本解决了以前的同类设备所存在的死机、识别易出错等问题。后期, 本长跑终端系统将会作为核心终端运用到本校的阳光长跑运动中。

6 结语

本文以嵌入式 Linux 为基础, 设计并实现了一套基于 DM3730 处理器的阳光长跑终端系统, 解决了传统长跑模式存在的弊端。系统实现的基于身份证与指纹的二次识别技术, 具有快速、准确的特性, 整个身份识别的过程耗时不到 10 秒, 且识别的准确率达到 99.9%; 开发的多功能人机交互界面给予了用户良好的设备体验性; 终端系统配合远程服务器和后台 Web

的开发, 实现长跑信息的智能化、网络化管理。终端设备后期进一步开发考虑运行 Android 系统, 将其与移动互联网结合起来。本套长跑终端设备可广泛应用于高校、社区等地, 具有很高的应用和 market 价值。

参考文献

- 张健, 曾必贵. 我国高校阳光长跑活动现状的研究. 运动精品, 2011.
- 周丽萍. 阳光体育运动的现状挑战与突破. 中国体育报, 2011.
- 倪旭翔, 计春雷. Cortex-A8 嵌入式系统开发与实践. 上海: 中国水利水电出版社, 2011.
- 陈永强, 陶品, 王笃强. 嵌入式 Linux 移植. 实验室研究与探索, 2012, 31(9): 67-69.
- Stevens WR, Rago SA. 尤晋元, 张亚英, 戚正伟译. UNIX 高级环境编程. 北京: 人民邮电出版社, 2012.
- 杨雷刚. 基于达芬奇平台的公交车载视频系统研究与设计 [学位论文]. 浙江: 浙江工业大学, 2012.
- 郑强. Linux 驱动开发入门与实战. 北京: 清华大学出版社, 2011.
- Khalil MS, Khan MK, Razzak MI. Co-occurrence matrix features for fingerprint verification. Proc. of 2011 IEEE International Conference on Anti-Counterfeiting, Security and Identification. 2011. 43-46.
- Zhou WN, Han JZ, Xiao Y, Yan WW. Fingerprint verification based on wavelet and edge detection. The Ninth International Conference on Electronic Measurement & Instruments. 2009. 1001-1004.
- 赵莹, 徐大平. 基于 Qt/Embedded 的嵌入式控制界面开发. 微计算机信息, 2009, 25(4): 36-38.
- 解辉, 徐玉斌, 李建伟. 基于 SQLite 的嵌入式数据采集系统的研究与设计. 计算机与数字工程, 2008, 36(6): 91-94.
- 陈霞. 基于 S3C6410 和 Qt4 的网络视频监控系统的研究与设计. 电脑知识与技术, 2013, 9(2): 383-385.