

基于 ARM 和嵌入式 Linux 的智能小车技术实践^①

林粤伟¹, 牟森¹, 姜海君^{1,2}

¹(青岛科技大学 信息科学技术学院, 青岛 266061)

²(北京邮电大学 电子工程学院, 北京 100876)

摘要: 论述基于 ARM 和嵌入式 Linux 的 WiFi 无线遥控智能小车的技术实践. 介绍技术实践的过程、方法、经验体会. 采用 ARM 11 芯片作为小车的主控制器. 小车上的 WiFi 模块承担无线通信功能并接收来自远程 Android 智能手机的遥控指令. 智能手机可通过“重力感应”或传统“按键”两种模式向小车发送遥控指令. ARM 控制器接收到指令后通过 GPIO 接口控制小车的电机驱动小车的四个轮子做出相应动作(前进、后退、转弯、旋转等), 并通过 PID 算法控制小车走直线. 小车可采集视频并运行计算机视觉算法进行实时人脸检测.

关键词: ARM; 嵌入式 Linux; 无线通信; 机器人; 计算机视觉

引用格式: 林粤伟, 牟森, 姜海君. 基于 ARM 和嵌入式 Linux 的智能小车技术实践. 计算机系统应用, 2018, 27(8): 241-246. <http://www.c-s-a.org.cn/1003-3254/6510.html>

Engineering Practice of Smart Mini Car Based on ARM and Embedded Linux

LIN Yue-Wei¹, MU Sen¹, JIANG Hai-Jun^{1,2}

¹(College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)

²(School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The engineering practice of mini smart car with WiFi wireless remote control function based on ARM and embedded Linux is discussed. The process, method, and experience of engineering practice are introduced. The ARM11 chip is selected as the central control CPU. The WiFi wireless communication module on smart car is used to communicate with remote Android smart phone to receive controlling commands. Controlling commands are sent by the smart phone through the novel ‘gravity sensing’ or traditional ‘key pressing’ mode. Once the commands are received by the ARM chip, it will control the motors via GPIO and drive four wheels to make according actions (moving forward/backward, turning and rotating), and force the car to move in a straight line via PID algorithm. The video data can be collected by camera and the real-time human faces detection is implemented by computer vision algorithms.

Key words: ARM; embedded Linux; wireless communication; robot; computer vision

引言

随着经济与科技的快速发展, 机器人技术正在逐步普及, 在各行各业得到了较多应用. 在社会上的部分岗位中, 机器人已经可以完全替代人类. 当前对于四轮简易机器人, 即智能小车的开发仍是大学生创新创业训练、各类大赛的热点题目, 相关技术资料较为丰富,

文献[1]对无线遥控智能小车进行了介绍. 对于小车的轮子电机控制硬件设计, 在文献[2]中给出了描述. 对于小车的 PID 控制软件设计, 在文献[3]中给出了介绍. 本文借鉴智能小车相关文献和大赛, 如恩智浦杯^[4]、博创杯等比赛成果用到的开发技术, 进行具有计算机视觉人脸识别功能的无线遥控智能小车的技术实践. 本文

① 基金项目: 青岛科技大学大学生创新创业训练计划项目 (201710426023); 青岛科技大学博士科研基金项目 (210-010022614)

Foundation item: Project for College Students Innovation Training of Qingdao University of Science and Technology (201710426023); Project for Doctor of Qingdao University of Science and Technology (210-010022614)

收稿时间: 2017-09-17; 修改时间: 2017-09-30; 采用时间: 2018-02-07; csa 在线出版时间: 2018-07-28

采用基于 ARM^[5]和嵌入式 Linux 系统的方案实现属于简易机器人的四轮智能小车,较传统单片机的解决方案,其智能性和扩展性较高.较同类基于 ARM 和嵌入式 Linux 的解决方案,其主要特点在于基于 OpenCV 实现了一定的人脸识别功能.

本文中手机可以通过传统“按键动作”发送命令,也可以通过较为新颖的“重力感应动作”发送命令,提高了无线遥控的用户体验和乐趣.在手机和小车之间组建无线网络,通过 Android 智能手机的 APP 发送无线遥控命令,手机可以通过按键动作发送命令也可以通过重力感应动作发送命令.重力感应指的是手机上的重力传感器能够根据手机的不同摆放状态获得不同的数据,ARM 芯片通过 WiFi 模块接收命令并进行相应的操作控制,如小车的前进、后退、转弯、旋转、停车等.

1 关键技术

1.1 智能小车系统架构

如图 1,无线遥控小车架构由车轮、小车底板、电机、电池、OK6410 ARM 开发板^[6](带 S3C6410 处理器)、WiFi 模块 (SDIO 接口)、USB 摄像头、速度传感器、Android 手机客户端组成.

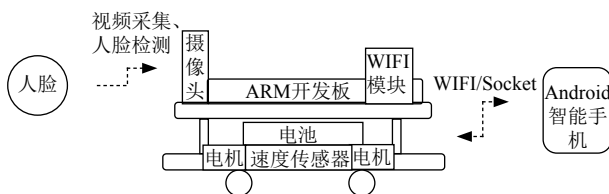


图 1 智能小车系统架构

1.2 关键技术概述

使用的关键技术包括:基于重力传感器的智能手机无线遥控、WiFi/Socket 套接字^[7]通信、计算机视觉人脸检测算法、PID 控制算法^[8]等.

如图 2,基于 Android 智能手机中的重力传感器实现重力感应^[9],需要首先得到 x 、 y 、 z 三轴的加速度的值.该值包含了地心引力的影响,如当将手机放在水平桌面上时, x 轴默认值为 0, y 轴默认值为 0, z 轴默认值为 9.81.将手机向左倾斜时, x 轴的正值.将手机向右倾斜时, x 轴的值为负.将手机向前倾斜时, y 轴的值为负.将手机向后倾斜时, y 轴的正值.

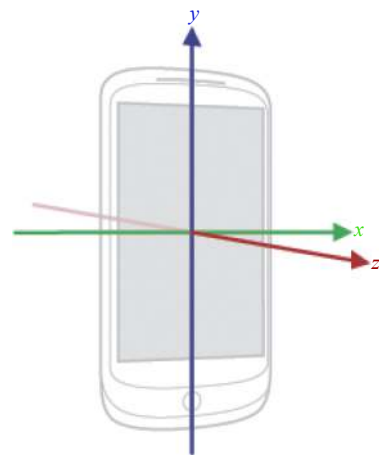


图 2 传感器的坐标轴

利用传感器技术获得手机状态后,还要将与状态对应的命令通过专门的通信技术发送给小车(如手机左倾代表遥控小车左转).ARM 开发板与智能手机之间采取 C/S 架构的 WiFi/Socket (套接字) 技术进行互联通信.其中 ARM 板上运行着 Socket 服务器,负责驱动 WiFi 模块并获取 Android 智能手机客户端发送来的数据命令包,然后把命令解析出来,通过 GPIO 端口把控制信息传送给小车底板,小车底板中的驱动电路接收到控制信号,做出相应的操作驱动电机改变车轮运动状态,从而达到无线遥控小车运动的目的.ARM 小车上运行 Socket 服务端程序,设置 WiFi 模块为热点模式,手机上安装 Socket 通信客户端 APP,使手机连接到 ARM 小车 WiFi 热点,打开手机 APP,设置相应的服务端 IP 和端口号,设置发送的命令.控制端手机 APP(Socket 通信客户端)发送命令数据包,手机 APP 发送指令数据包分为两种方式:屏幕按键、重力感应.屏幕按键操作是用户按住 APP 的前、后、左、右键发送相应的数据包命令;重力感应操作是用户在手机 APP 上选择重力感应模式,通过把手机前倾、后倾、左倾、右倾完成数据包命令的发送.ARM 小车服务端接收到数据包后,对数据包进行解析,然后执行相应的操作,如前进、后退、左转、右转、左旋、右旋、停车等.

人脸识别用到的人脸检测算法是采用 Adaboost 学习算法^[10],它把一些比较弱的分类方法级联整合在一起,组合出新的很强的分类措施.采用比例积分微分 (Proportion Integral Differential, PID) 控制算法控制小车的车轮转速,确保其直线行走和准确转弯.关于人脸

检测算法和 PID 控制算法将在软件设计与实现小节详细描述。

2 硬件设计过程与方法

2.1 关键器件选型

小车采用四个驱动车轮,四个轮子都安装了直流电机,大大增加了地面的阻力。四个轮子和电机都是同种型号的,各个轮子几乎没有差别,在相同供电的情况下,小车运动时,小车的重力中心在小车底盘中心,由于四个轮子都可以单独控制,小车运动时很难跑偏。当然四个驱动轮使得小车有更快的速度,当小车停止时,

由于四个轮子都不转,直接增大了摩擦力,小车很快就能停下,这个小车底盘性能较好,满足本小车的需求^[11]。四驱小车速度达到了预期要求,在小车执行直行、转弯、旋转等动作时更便于控制。

S3C6410 芯片是三星公司的 ARM11 架构处理器,本文选用的中国飞凌公司的 OK6410 开发板采用了 S3C6410 为主控芯片,功能强大,便于用户二次开发^[3]。WiFi 模块较蓝牙模块的通信距离更远,更适合无线遥控,因此选用飞凌公司基于 RTL8189ES 型号的 WiFi 模块^[6],该模块通过 SDIO 接口与板子连接。选用普通 USB 接口的摄像头,通过 USB 口与板子连接。

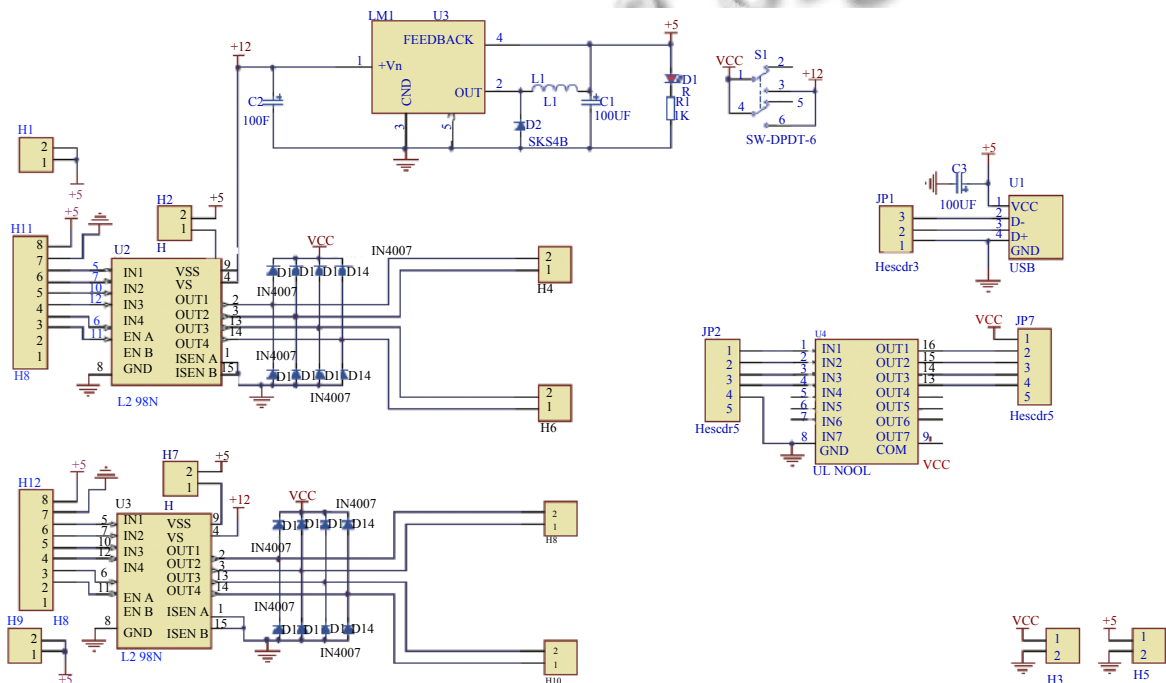


图3 双 L298N 原理图

2.2 电机及其驱动电路

选择了 4 个 130 直流电机,130 电机惯量适中。小车的轮子由四个电机构成,且每个电机都可以单独控制,即四轮驱动。小车的转向是通过轮子的差速实现的,即左右两个轮子存在速度差时,小车就能实现转弯。假如左边两个轮子的速度比右边两个轮子的快,那么小车会偏向右;同理,左边两个轮子比右边两个轮子速度慢,那么小车就会偏左。由于四个轮子的 130 电机需要差速转弯,转向时轮子与地面有很大的阻力,使用的电流比较大,实际测量,小车在走直线时需要 600~850 mA 的电流,转向时需要提供的电流约为 1.2~1.5 A。

130 直流电机工作电压在 3~9 V 之间,实际选取 6 V 左右,此时小车状态较好。

四个直流电机选用双 L298N 电机驱动模块^[8]驱动,两个电机分别驱动左右两对轮子。L298N 芯片是 ST 公司制作的,并且工艺稳定性高,其原理图如图 3,能够直接驱动 4 路 3~16 V 直流电机。电机驱动电路输入接口的 GND 与 OK6410 开发板的 GND 连在一起,电机供电电池的 GND 和驱动电路输出接口的 GND 连在一起,保证逻辑电平的统一。制作了专门的排线以连接 ARM 开发板与电机驱动电路。

电机驱动电路的输入端有 8 个管脚 (IN1~IN8),分

别连接 OK6410 板子的 8 个引出的预留给用户的 GPIO 管脚, 这 8 个 GPIO 管脚分别连接驱动电路的 IN1~IN8 输入管脚, 对应驱动电路的 8 路输出端口 OUT1~OUT8, 以控制车轮旋转方向(使轮子正转、反转、不转)。基于 S3C6410 处理器的定时器内部中断, 由软件模拟产生 GPIO 管脚上的 PWM 波, 可为左右两对车轮的电机提供 PWM 信号, 以控制车轮转速(在左右两对轮子之间形成速度差, 使小车利用差速转弯)。除了上述 8 个作为输出端口的 GPIO 管脚, 板子还有另外 2 个 GPIO 管脚作为输入端口, 分别连接速度传感器的输出管脚, 以达到测速目的。

如表 1, 驱动电路输出端的 8 路输出端口分别连接 4 个车轮的电机(每个电机需要 2 根接线, 4 个电机共需 8 根线, 分别接 8 个输出端口)。以左上电机为例: 当 OUT1、OUT2 同为高电平或低电平时, 左上电机不转; 当 OUT1 为低电平, OUT2 为高电平时, 左上电机正转; 当 OUT1 为高电平, OUT2 为低电平时, 左上电机反转。其他 3 个电机的控制以此类推。

表 1 驱动模块输出端口与电机接线

电机	直流电机接线 1	直流电机接线 2
小车左上电机	OUT1	OUT2
小车左下电机	OUT3	OUT4
小车右上电机	OUT5	OUT6
小车右下电机	OUT7	OUT8

3 软件设计过程与方法

在 PC 机安装虚拟机软件, 虚拟机内安装的是 Ubuntu 12.04, 安装 ARM 交叉编译器、QT Creator IDE(用于开发 QT 程序, QT 是嵌入式 Linux 系统的一种图形化用户界面, 类似于 PC Windows 系统中的桌面^[6])。OK6410 板子中烧录 Linux 3.0.1 版本的内核。

3.1 GPIO 驱动程序与车轮控制

三星 S3C6410 芯片包含 187 个多功能输入/输出端口管脚, 而小车的控制采用 OK6410 开发板预留的用户端口中的 10 路 GPIO, 分别为 GPC0~GPC7(输出由软件模拟的 PWM 波)、GPP8~GPP9(输入端口, 分别检测由速度传感器产生的关联左右两对轮子转速的外部中断)。以 GPC 寄存器的配置为例, GPCCON、GPCDAT、GPCPUD、GPCCONSLP、GPCPUDSLP, 这五个控制寄存器都属于端口 GPC 控制寄存器。寄存器 GPC 的端口都有多种的配置方式, 包括输入、输出、外部中断等, 在小车控制中需要把 GPIO 端口配置为输出模式, 具体配置见表 2。此时 GPC7~0 的管脚

状态与 GPCDAT 的低八位[7:0]相应的位状态相同, GPP9、GPP8 的管脚状态与 GPPDAT 的中间两位[9:8]相应的位状态相同^[12]。

表 2 GPCCON 寄存器配置

管脚	GPCCON[31:0]	寄存器值(含义)	电机驱动模块输入端口
GPC0	[3:0]	0001(输出)	IN1
GPC1	[7:4]	0001(输出)	IN2
GPC2	[11:8]	0001(输出)	IN3
GPC3	[15:12]	0001(输出)	IN4
GPC4	[19:16]	0001(输出)	IN5
GPC5	[23:20]	0001(输出)	IN6
GPC6	[27:24]	0001(输出)	IN7
GPC7	[31:28]	0001(输出)	IN8

在 GPIO 的驱动中, 首先要编写驱动的入口函数和卸载函数^[13], 驱动的入口函数为 mygpio_init(), 卸载函数为 mygpio_exit(), 打开设备的函数为 mygpio_open(), 控制设备的函数为 mygpio_ioctl()。在 mygpio_open() 函数中, 首先要进行地址映射, 内核中访问 IO 内存之前, 只知道它们的物理地址, 不能使用软件直接访问, 调用 ioremap() 将设备物理地址映射为虚拟地址, 然后根据虚拟地址范围去访问 IO 资源。mygpio_ioctl() 函数可以控制每一个车轮的转动状态。

小车的实际速度有时不稳定, 不能直接线性调速, 且小车的四个轮子的直流电机也存在微小的差异, 使得小车轮子之间的转速有差异。为此, 本设计采用 PID(Proportion Integral Differential, 比例积分微分)控制算法控制小车的车轮转速, 确保其直线行走和准确转弯。采用定时器来模拟四路 PWM(Pulse Width Modulation, 脉冲宽度调制)来控制四个直流电机的转速。PID 一般包括两种: 位置式 PID 和增量式 PID, 在小车控制中一般采用增量式, 本文采用的就是这种方式。

计算机控制是经过采样后的数字化控制, 这种控制必须按照采样时间点的偏差计算控制量, 无法输出连续的模拟控制量, 因此积分项和微分项是不能被直接使用的, 而是要先进行处理, 使之离散化。其离散化处理的方法为: 以 T 为采样周期, k 为采样序号, 则离散采样时间 kT 对应着连续时间 t , 用求和代替积分, 用后向差分代替微分, 则离散化的 PID 表达式近似如下:

$$u_k = K_p [e_k + T/T_i \times \sum_{j=0}^k e_j + T_d \times (e_k - e_{k-1})/T] \quad (1)$$

式(1)中, e_k 为输入的控制误差, u_k 为输出, K_p 为比例

系数, T_i 与积分时间相关, T_d 为微分时间常数. 所谓增量式 PID 是指数字控制器的输出只是控制量的增量, 增量式 PID 控制算法可以得到控制器的第 $k-1$ 个采样时刻的输出值为:

$$u_{k-1} = K_p [e_{k-1} + T/T_i \times \sum_{j=0}^{k-1} e_j + T_d \times (e_{k-1} - e_{k-2})/T] \quad (2)$$

将 (1) 与 (2) 相减并整理, 就可以得到增量式 PID 控制算法公式为:

$$\begin{aligned} \Delta u_k &= u_k - u_{k-1} \\ &= K_p [e_k - e_{k-1} + T/T_i \times e_k + T_d(e_{k-2} - e_{k-1} + e_{k-2})/T] \\ &= A e_k + B e_{k-1} + C e_{k-2} \end{aligned} \quad (3)$$

其中

$$\begin{aligned} A &= K_p(1 + T/T_i + T_d/T) \\ B &= K_p(1 + 2T_d/T) \\ C &= K_p T_d/T \end{aligned} \quad (4)$$

由式 (3) 可以看出, 如果计算机控制系统采用恒定的采用周期 T , 只要确定了 A 、 B 、 C 的值, 并使用时间上前后三次测量的目标速度与实际速度的偏差值 e_k 、 e_{k-1} 、 e_{k-2} , 就可以求出控制量的增量 Δu_k . 小车的车轮转速采样周期为 100 ms, 由 PID 算法得出 A 、 B 、 C 三个参数, 按照 Δu_k 来增大或减小输出 PWM 波的占空比, 从而调整车轮转速.

3.2 视频采集与人脸识别

摄像头与 ARM 开发板通过 USB 口连接. 在 ARM 开发板上, 采用 V4L2 (Video for Linux 2) API^[14] 进行打开摄像头等操作, 并基于 Open CV 2.0 开源计算机视觉代码包^[15] 处理视频数据. 使用 QT 界面程序显示采集到的视频并进行人脸检测.

开发时发现 Open CV 的打开摄像头的接口函数在 ARM 开发板上不能正常运行, 故采用 Linux 中的 V4L2 接口来配合编写摄像头程序, 其中包括: 打开摄像头、初始化摄像头、采集视频帧、停止采集视频帧、卸载摄像头设备、获取视频帧、关闭摄像头等. 获取到摄像头视频后, 把视频进行压缩, 从而提高人脸检测的速度, 在视频流中进行人脸检测, 然后恢复原图的大小, 使用 Open CV 函数进行图像格式转换并把视频显示在 QT 界面程序中. 在人脸检测中, 如果检测到人脸, 则在图像中以人脸为中心, 围绕着人脸画出一个方框来标注人脸, 如图 4.

利用 Open CV 中支持的人脸检测方法实现人脸识别: 首先利用样本的 Haar 特征进行分类训练, 获得级

联 boosted 分类器, 进而获得级联分类器类 CascadeClassifier, 使用该类对视频流中的目标进行检测. 检测时需要加载 XML 分类器文件 ("haarcascade_frontalface_alt.xml"). 在程序中构造了一个 detectAndDisplay() 函数, 对在视频流中读取的每一帧图像进行检测. 该函数主要是将每一帧图像转换为灰度图像, 然后进行直方图均衡化, 然后找出人脸区域; 最后显示人脸目标: 定出人脸区域后, 以人脸为中心画出一个方框, 如图 4.

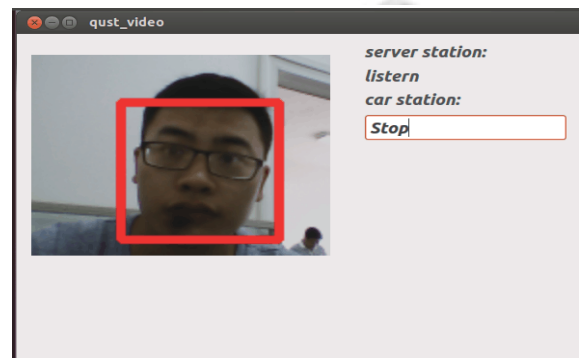


图 4 人脸识别运行界面

3.3 软件主流程

软件主流程如图 5, 整体为 C/S 架构, 小车做服务端, 手机做客户端, 两者之间采取 WiFi/Socket 技术进行通信. 其中小车部分为多线程结构, 主线程负责视频采集与人脸识别, 次线程负责小车控制.

4 实践遇到的主要问题、解决方法和经验体会

首先, 使用 OpenCV 的函数调用摄像头在虚拟机中的 Linux 中可以正常运行, 没有问题存在, 但是当把程序移植到 ARM 开发板的嵌入式 Linux 系统中时, 会出现段错误, 主要是 OpenCV 的相关函数出现问题. 比如其中一个出现问题的函数是 cvCreateCameraCapture(), 调试时发现运行到这里就出现错误了. 采取的措施是避开使用 OpenCV 的函数打开摄像头, 采用 V4L2 来打开摄像头, 使用 OpenCV 的函数来处理图像, 问题成功解决.

另外, 在 QT 调用 OpenCV 的功能调试时调试时间较长, 发现导致问题的因素大都是某些细节没有处理好, 但初次实践者较易忽略. 笔者经过实践解决了问题, 总结经验如下: 在 QT 程序中, 需要把 OpenCV 的动态链接库路径加入到工程中, 并在 ARM 开发板中把所有的链接库 (*.so) 文件添加, 此时需要保证路径和

PC 机上的一致. 如果出现在 ARM 上运行的程序提示

找不到链接库, 可能是由于链接库路径太长.

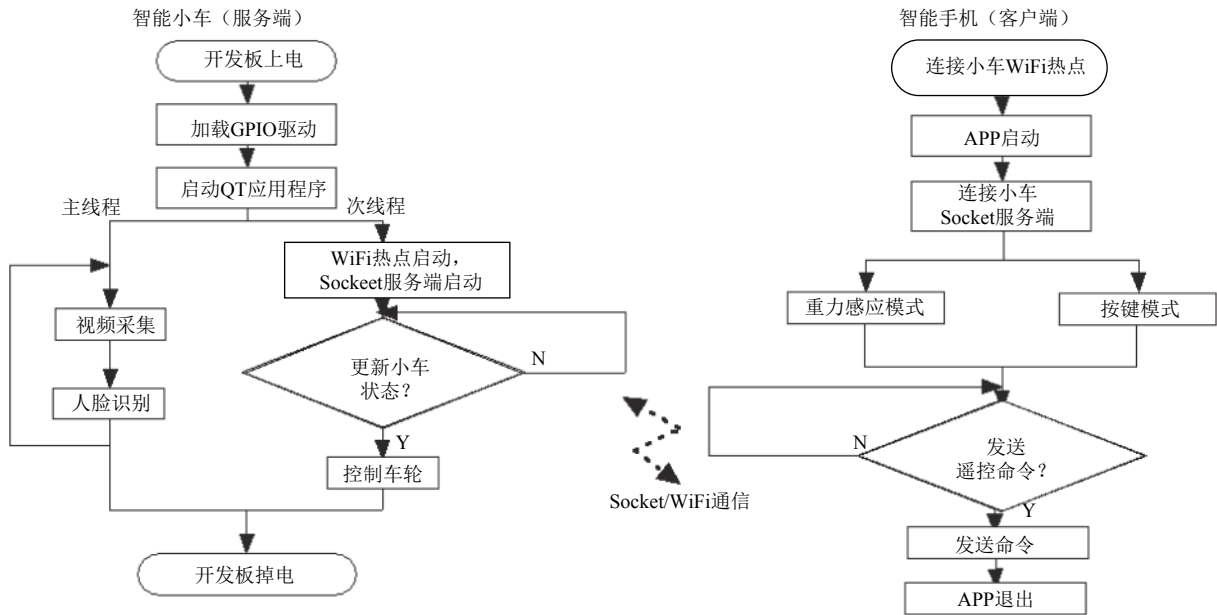


图5 C/S 架构智能小车主程序流程图

5 结论

本文研制的小车实物如图6. 经过实测, 小车各项功能达到了预期设计目标, 运行稳定, 小车可被手机无线遥控, 并支持视频采集与人脸检测, 有效遥控距离为60米左右, 在防爆、救援、养老等领域具有较好的应用价值. 比如, 小车可在人类视野内被遥控进入危险的人类不便进入的区域, 以进行识别、救援、排爆等任务. 将来考虑实现其他计算机视觉算法, 并增加语音识别等功能.



图6 智能小车实物照片

参考文献

- 1 朱丹峰, 葛主冉, 林晓雷. 基于 Android 平台的无线遥控智能小车. 电子器件, 2013, 36(3): 408-412. [doi: 10.3969/j.issn.1005-9490.2013.03.027]
- 2 宋洪慧, 李春蒂, 朱江. 智能小车直流电机驱动电路设计. 机电技术, 2015, (6): 84-87. [doi: 10.3969/j.issn.1672-4801.

2015.06.028]

- 3 隋妍. 基于数字 PID 的智能小车的控制. 数字技术与应用, 2012, (6): 3, 6.
- 4 全国大学生智能汽车竞赛秘书处. 全国大学生“恩智浦杯”智能汽车竞赛官网. <http://www.smartcar.au.tsinghua.edu.cn>.
- 5 ARM Limited. ARM architecture reference manual. ARM Limited, 2005: 23-37.
- 6 飞凌嵌入式. OK6410-A Linux 使用手册. <http://www.forlinux.com>.
- 7 Stevens WR, Fenner B, Rudoff AM. UNIX 网络编程: 卷 1: 套接字联网 API. 北京: 人民邮电出版社, 2015.
- 8 刘金琨. 先进 PID 控制 MATLAB 仿真. 3 版. 北京: 电子工业出版社, 2011.
- 9 丁山. Android 程序设计教程. 北京: 机械工业出版社, 2015.
- 10 曹莹, 苗启广, 刘家辰, 等. AdaBoost 算法研究进展与展望. 自动化学报, 2013, 39(6): 745-758.
- 11 迎创电子. 初始 ZK-4WD 小车机器人. <http://www.mcustudio.com.cn>.
- 12 飞凌嵌入式. OK6410-A 开发板硬件手册. <http://www.forlinux.com>.
- 13 宋宝华. Linux 设备驱动开发详解. 2 版. 北京: 人民邮电出版社, 2010.
- 14 王飞, 孔聪. 基于 V4L2 的 Linux 摄像头驱动的实现. 电子科技, 2012, 25(2): 86-87, 92. [doi: 10.3969/j.issn.1001-9146.2012.02.022]
- 15 Laganieri R. OpenCV2 计算机视觉编程手册. 张静, 译. 北京: 科学出版社, 2013.