2018 Synopsys ARC杯海峡两岸

电子设计竞赛技术论文

论文题目：

智能公交站牌

参赛单位：西安电子科技大学

队伍名称：武陵酒参赛队

指导老师：李康老师

参赛队员：郭嘉敏 张永薪 黄 飘

完成时间：2018年 05月25日

# 基本情况表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 队伍名称 | 武陵酒 | | | 单位名称 | | 西安电子科技大学 | |
| 项目名称 | 智能公交站牌 | | | | | | |
| 项目负责人 | 郭嘉敏 | | | 联系方式 | | | 18392566646 |
| 指导老师 | 李康 | | | 职务 | | | 副教授 |
| 参赛  队员  信息 | 姓名 | 学历 | 证件号码 | | 专业 | | 分工情况 |
| 郭嘉敏 | 硕士在读 | 140321199310080019 | | 微电子学与固体电子学 | | 硬件、软件 |
| 张永薪 | 硕士在读 | 410823199409300056 | | 集成电路系统设计 | | 硬件、软件 |
| 黄飘 | 硕士在读 | 42098419951016532X | | 集成电路系统设计 | | 软件、论文 |
| 项目时间 | 2018年 1月16日 - 2018年 5月25日 | | | | | | |
| 队伍简介 | 本队伍由品学兼优、踏实努力的西安电子科技大学北校区新科509实验室的三名学硕组成 | | | | | | |
| 参与项目 | TFET新型半导体器件研究（进行中），安防系统研究（进行中） | | | | | | |
| 获奖情况  （校级及  以上） |  | | | | | | |
| 研究专长 | FPGA、 | | | | | | |

# 摘 要

当今世界，科技日新月异，生活质量日益提高，交通便利，出行方便，但是也有很多不便利的地方，比如公交车。等公交车的人需要花费较长的时间，有时甚至等到拥挤的公交车。快节奏的生活需要更便利的智慧交通。因此，一款能够提前知道公交车在哪一站并能反应车拥挤程度的公交站牌可以为候车人提供便利、节省时间。相比公交车而言，大部分人都喜欢乘坐地铁，不仅仅是因为地铁准时、快捷，还有一部分原因是因为地铁更智能方便。等地铁的时候能够具体知道下一班地铁大约几分钟后会到达。反观公交车，很多人都等过半个小时甚至更长时间的公交车。或者好不容易等来的公交车，发现满载而错过了选择其他交通方式的机会。如果公交站牌能够类似地铁显示车到站的位置，以及公交车的拥挤程度，可以给候车人提供很多便利。智能公交站牌就实现了这样的功能。

目前大部分城市里面的公交车站牌是线路指示牌，仅提供路线引导功能。少部分地区使用电子站牌，采用LCD显示屏，集显示车位置和打广告于一身。但LCD显示屏价格较高，导致电子屏站牌成本高，难以得到推广。本论文的智能公交站牌以ARC处理器为核心，借助WIFI模块以及机智云实现公交车与站牌之间的通信，并使用三色LED灯在公交站牌上显示公交车位置和车拥挤程度信息。

关键词：智慧交通 公交站牌 ARC处理器 WiFi 机智云

# **ABSTRACT**

In today's world, science and technology are changing rapidly.With our country development,the level of people life increases continuously ,people's requestis more and more high to the living quantity. Traffic developed and more convenient. However, bus is a transpotation which is very inconvenient for a lot of people. It takes a long time for people to wait for the bus, Sometimes even waiting for the crowded bus. fast-paced life requires more convenient and more intelligent transportation.Therefore,a type of bus stop, which shall show in advance the bus location and crowed level, can save time for waiting people. Compared to the bus, most people like to take the subway. It's not just because the subway has the merit of convenience, shortcut, on time, vast capacity,  but also because it's more intelligent and convenient. When waiting for the subway, you can know exactly when the next train will arrive .As for bus, many people wait for more than half an hour or more. Another example is when the bus is approaching , you find it full load and miss the opportunity to choose other vehicles. If the bus stop sign can show the location of the bus and the degree of crowding,which can provide a lot of convenience to the waiting people.The [intelligent](http://dict.youdao.com/w/intelligent/#keyfrom=E2Ctranslation) bus stop implement such a feature.

At present, the bus stop sign in most cities is of the line indicator, which provides the route guidance only. Few areas use electronic bus stop sign with LCD screen,which display bus location and advertise. However, the high price of LCD screen makes it difficult to promote the high cost of electronic screen station. The intelligent bus stop in this paper is based on the ARC processor,and  communicates with bus by wifi and Gizwits.Finanlly, three color LED lights display bus position and  degree of crowding information on bus stop sign.

**Keywords: Intelligent Transportation Bus Stop Sign ARC processor Wifi Gizwits**

# 目 录

基本情况表 ii

摘 要 iii

**ABSTRACT** iv

目 录 V

第一章 方案论证 1

1.1项目概述 1

1.2资源评估 1

1.3预期结果 2

1.4项目实施评估 2

1.5补充说明 2

第二章 作品难点与创新 4

2.1作品难点分析 4

2.2创新性分析 4

2.3小结 4

第三章 系统结构与硬件实现 5

3.1系统原理分析 5

3.2 系统结构 5

3.3硬件实现 6

3.4 小结 8

第四章 软件设计流程及实现 9

4.1软件设计流程 9

4.2软件实现 10

4.2.1算法一 11

4.2.2算法二 22

4.2.3算法三 26

4.3小结 30

第五章 系统测试与分析 31

5.1系统测试指标 31

5.2 测试环境 31

5.2.1验证开发平台 31

5.2.2测试方案 32

5.3测试结果 32

5.3结果分析 35

第六章 总 结 36

参考文献 37

# 第一章 方案论证

## 1.1项目概述

该项目是基于ARC EM处理器的智能公交车站牌，可以显示车的位置信息和车的拥挤程度信息。为候车人提供方便，节省时间，并实现推广，响应智慧交通。硬件方面采用ARC EM处理器、三色LED灯、WiFi、红外对射等，用C语言编写程序控制硬件的工作。亮点：车和站牌之间采用WiFi并连接到机智云上通信。通过三色LED灯显示车的位置和拥挤程度，相比电子屏站牌，更方便且成本低。公交车接近智能公交站牌A时，车上的WiFi和A站牌上的WiFi相互感应，A站牌即获取到车即将到本站的信息。车门上装有红外对射模块，用来记录车载人数。WiFi将到站信息以及车载人数信息传到机智云，各个站牌上的WiFi模块再向机智云下载这些信息，并通过LED流水灯显示车的位置，并以灯不同的颜色显示公交车的拥挤程度。

## 1.2资源评估

所需模块有，三色LED灯六个，红外对射模块一个，WiFi模块六个， SD卡一个，连接线若干，三个ARC EM板，亚克力板四片，胶水一瓶。每个ARC EM板的接口有用到两个UART接口，两个或三个GPIO接口。两个ARC板用作gpio接口的接三色LED灯模拟公交站牌，每个板使用两个gpio接口和两个三色LED灯连接，两个uart接口和WiFi模块连接，用来接收机智云传来的数据。另一个ARC板用作模拟公交车，两个gpio接口和红外对射连接，一个用来计上车人数，另一个用来计下车人数。两个uart接口和两个WiFi模块连接，用来给机智云分别发送位置数据和红外对射模块的数据。所有外设采购花费约两百，资金来源为队员均摊。

## 1.3预期结果

通过对各项功能的设计研究和不断测试，预期达到的效果：

1）车从A站起步，B站灯牌显示A站（第二个灯）灯亮。C站牌显示A站（第二个灯）灯亮。

2）红外对射前门模块统计上车人数，红外对射后门模块统计下车人数，前门人数减去后门人数表示车上人数。若车上人数在小段范围，则B站和C站灯牌显示人数的三色LED灯亮绿灯；若车上人数在中段范围，则B站和C站灯牌显示人数的三色LED灯亮黄灯；若车上人数在大段范围，则B站和C站灯牌显示人数的三色LED灯亮红灯。

3）车到达B站后，C站灯牌显示B站（第三个灯）灯亮。

4）此时，B站和A站均不显示灯亮，因为灯牌显示的是来这条线路上的车信息，已经离去的车信息不显示。

## 1.4项目实施评估

开始是队长提出项目想法，各成员开始收集资料，并进行各控制模块划分，评估硬件需求。寻求导师建议，并设计电路结构图以及模块连接。中期进行模块内部程序编写，调试；后期，各模块组装，基础模型形成；并进一步测试，改进，增强稳定性；最终，作品外部美化，包装，作品形成。

难点：机智云的设备联动，灯牌显示硬件选择，最后系统整合，演示时ARC板资源紧缺。

工作分配：张永薪和郭嘉敏辅助各模块程序编写，包括C语言，AT指令以及机智云协议，还有硬件的设计与串口连接。黄飘负责论文撰写，展示PPT制作，成品测试优化，外部设计美化，以及部分软件代码。最终在大家共同努力下，项目得以在规定时间内完成。

## 1.5补充说明

导师李康主要致力于嵌入式系统方法学，在整个项目完成过程当中给予很多很有用的建议，并提供了实验室和工具，对我们帮助很大。

# 第二章 作品难点与创新

## 2.1作品难点分析

作品难点主要体现在机智云的设备联动，考虑公交车实际中多辆的情况后，选择使用机智云，在对机智云进行开发的过程中遇到了最难的问题，设备联动，云端联动规则的设定，以及机智云的协议，学习了一段时间。第二个难题是智能公交站牌显示的选择，LCD屏幕成本高，驱动复杂，最终综合考虑选择了三色LED灯串。

## 2.2创新性分析

第一点是SD卡的使用，将工作所需程序全部下载到SD卡后，将SD卡插入板子中，按下复位键后板子就可以脱离电脑独立工作。

第二个创新是公交车和公交站牌之间使用WiFi和机智云通信，不受距离显示，实时通信。

第三个创新点是，三色LED灯的使用，直观成本低，容易推广。

## 2.3小结

本设计有较大的现实意义，若经过后续改良可以直接应用于实际生产生活中。虽然本设计在复杂度上有所欠缺，但是若经过后续加工整理可以极大的方便人们的生产生活，具有一定的现实意义。

# 第三章 系统结构与硬件实现

## 3.1系统原理分析

用作车的ARC板计为D板，用作公交站牌的两个ARC板分别计为B、C板。D板的两个uart串口接连个WiFi，一个为机智云固件，一个为AT指令固件，两个gpio接口分别接两个红外对射模块。由于板子有限，用手机模拟A公交站。当D板上的AT指令固件WiFi扫到手机开热点所发出的rssi信息后，AT指令固件WiFi检测到rssi后，本代码设计的是rssi小于50，即表示车到站。检测到车到站的信息后，红外对射模块开始统计人数，一个红外对射模块模拟前门上车，一个红外对射模块模拟后门下车，用前门人数减去后门人数，即为车上的人数。车的人数在小段范围内是上报数据点1，中断范围内上报数据点2，大段范围内上报数据点3。最后，机智云固件WiFi向机智云上报两个数据点，一个是站点序号，另一个是人数范围代表的数字。代表站牌B和站牌C的ARC板上的WiFi固件接收来自机智云的数据。处理数据后，控制LED亮相应的站点，并根据车上人数的范围选择对应的区域显示不同的颜色。

## 3.2 系统结构

整个系统主由站牌车和机智云三大部分构成，结构如下图：



图3.1 智能公交站牌系统结构图

机智云平台为开发者提供了自助式智能硬件开发工具与开放的云端服务。为了降低开发者的开发门槛，缩短开发周期，降低开发资源投入，机智云推出了代码自动生成服务。云端会根据产品定义的数据点生成对应产品的设备端代码

## 3.3硬件实现

⑴WiFi模块：

1. 原理介绍 模块核心处理器 ESP8266 在较小尺寸封装中集成了业界领先的 Tensilica L106 超低功耗 32 位微型 MCU，带有 16 位精简模式，主频支持 80 MHz 和 160 MHz，支持 RTOS，集成 Wi-Fi MAC/ BB/RF/PA/LNA，板载天线。支持标准的 IEEE802.11 b/g/n 协议，完整的 TCP/IP 协议栈。用户可以使用该模块为现有的设备添加联网功能，也可以构建独立的网络控制器。ESP8266 是高性能无线 SOC，以最低成本提供最大实用性，为 WiFi 功能嵌入其他系统提供无限可能。
2. 特点

* 802.11 b/g/n
* 内置Tensilica L106 超低功耗 32 位微型 MCU，主频支持 80 MHz 和160 MHz，支持 RTOS
* 内置10 bit高精度ADC
* 内置TCP/IP协议栈
* 内置TR 开关、balun、LNA、功率放大器和匹配网络
* 内置PLL、稳压器和电源管理组件，802.11b 模式下+20 dBm的输出功率
* A-MPDU 、 A-MSDU 的聚合和 0.4 s的保护间隔
* WiFi @ 2.4 GHz，支持 WPA/WPA2 安全模式
* 支持AT远程升级及云端OTA升级
* 支持 STA/AP/STA+AP 工作模式
* 支持 Smart Config 功能（包括 Android 和 iOS 设备）
* HSPI 、UART、I2C、I2S、IR Remote Control、PWM、GPIO
* 深度睡眠保持电流为 10 uA，关断电流小于 5 uA
* 2 ms 之内唤醒、连接并传递数据包
* 待机状态消耗功率小于1.0 mW (DTIM3)
* 工作温度范围：-20℃- 85℃

⑵红外对射模块：

1. 参数 主要芯片：LM393、对射式红外头  
    工作电压：直流5伏
2. 特点：  
   1、具有信号输出指示。  
   2、单路信号输出。  
   3、输出有效信号为低电平。  
   4、灵敏度不可调。  
   5、可用于工件计数、电机测速。。。。  
   6、电路板输出开关量

⑶色LED灯：

1. 参数

尺寸：25\*21mm 重量：4g 颜色：全彩 红绿蓝三基色 亮度：高亮

电压：5V

输入：数字电平

接口：Arduino 4P

接口平台：Arduino、单片机

b.特性

三基色原理显示多重颜色

通过PMW控制实现全彩显示

Arduino PMW接口可直接驱动

## 3.4 小结

本设计使用ARC EM Starter Kit开发板上的Pmod2，Pmod3和Pmod4接口，与LED RGB连接。Pmod1与WiFi连接，Pmod5与第四个LED RGB连接。产品连接效果图如下：

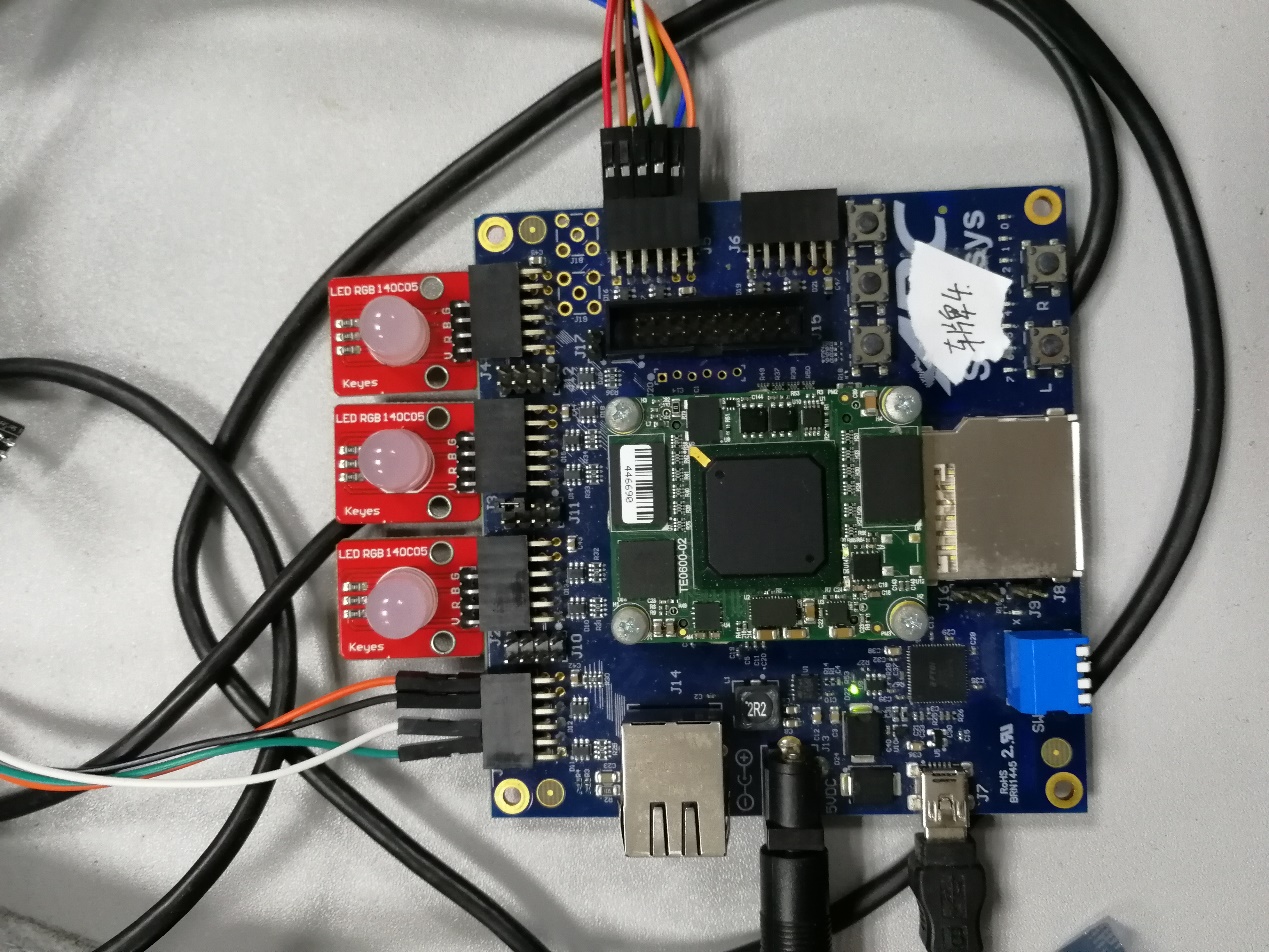


图3.2 智能公交站牌产品连接效果图

# 第四章 软件设计流程及实现

## 4.1软件设计流程

软件的设计流程主要包括两个部分：一个是公交车到站以及上车下车模拟部分的编写，另一个是站点灯显示部分的编写。小车函数和站点函数的流程图如4.1和4.2.



图4.1 站点函数基本流程图



图4.2 小车函数基本流程图

## 4.2软件实现

软件代码部分主要由三个部分组成，一个小车的控制代码，还有两个分别是站点2和站点3的代码，具体实现见下面的代码部分。

### 4.2.1算法一

static void delay\_ms( int z) //1ms

{

uint32\_t volatile x,y;

for(x=1400;x>0;x--)

for(y=z;y>0;y--);

}

void esp8266\_init(void)

{

// set pmod mux and uart map

set\_uart\_map(0xe4);

set\_pmod\_mux(PM1\_UR\_UART\_0|PM1\_LR\_GPIO\_A);

// Get devices structure pointer

esp8266\_uart = uart\_get\_dev(ESP8266\_UART\_ID);

esp8266\_rst = gpio\_get\_dev(ESP8266\_RST\_PORT);

int32\_t ercd;

// Open devices

esp8266\_rst->gpio\_close();

if((ercd = esp8266\_rst->gpio\_open(ESP8266\_RST\_DIR) ) == E\_OK)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open successfully!!\r\n");

}

else if(ercd == E\_OPNED)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed,because opend with different direction!!\r\n");

esp8266\_rst->gpio\_control(GPIO\_CMD\_SET\_BIT\_DIR\_OUTPUT,(void \*)(ESP8266\_RST\_MASK));

}

else

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed, unknow why!!\r\n");

}

esp8266\_uart->uart\_close();

if(esp8266\_uart->uart\_open(ESP8266\_UART\_BAUDRATE) != E\_OK)

{

EMBARC\_PRINTF("ESP8266\_UART open failed!!\r\n");

}else

{

EMBARC\_PRINTF("ESP8266\_UART open successfully!!\r\n");

}

//esp8266\_uart->uart\_control(UART\_CMD\_SET\_RXCB,esp8266\_uart\_rxcb);

//esp8266\_uart->uart\_control(UART\_CMD\_SET\_RXINT,CONV2VOID(1));

}

int main(void)

{

cpu\_lock();

//Board initialization;

board\_init();

//esp8266 initialization;

esp8266\_init();

cpu\_unlock();

uint8\_t StaNum=0;

uint8\_t gjm=0;

uint8\_t i=0;

uint8\_t j=0;

uint8\_t Crowd=0;

uint32\_t crowd=0;

uint16\_t rd\_cnt = 0;

uint8\_t read\_data[MAX\_READ\_CNT];

uint32\_t rd\_avail;

while(1) //机智云设备上线初始化

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01:

{

mcu2wifi\_product\_info();

break; //MCU传入产品的Key与Secret

}

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

j++;

break;

}

case 0x06: break;

case 0x07:

{

mcu2wifi\_heartbeat(); //MCU回复心跳包

j++;

break;

}

case 0x0a: break;

case 0x0c: break;

case 0x0d:

{

mcu2wifi\_wifi\_statu(read\_data);

i++;

break; //wifi将最新的工作状态推送至MCU

}

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

if(i == 4 || j==1) break;

}

while(gjm!=1)

{

gjm=lookforrssi1(); //当rssi满足条件，返回gjm=1，跳出循环，进行下一步对于机智云的操作

StaNum=1; //得到此时的公交所在站点，衡量水平为StaNum，此时值为1。

}

crowd=ct2crowd(); //通过红外脉冲模块计算车载的拥挤程度，衡量水平为crowd

while(1) //ARC板子读取wifi发送的数据函数,MCU完成回复，必不可少。

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07:

{

mcu2wifi\_heartbeat(); //MCU回复心跳包

break;

}

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

//crowd=ct2crowd(); //通过红外脉冲模块计算车载的拥挤程度，衡量水平为crowd

break; //得到此时的StaNum与crowd，跳出while循环

}

EMBARC\_PRINTF("StaNum=%d,crowd=%d\r\n",StaNum,crowd);

mcu2wifi\_dev\_report\_status(StaNum,crowd); //将数据传入云端

delay\_ms(10000);

/\*————————————————————————第一个公交站牌的数据上传由小车完成，第二个，第四个站牌由云端读入改变数据，小灯发生相应改变。——————————\*/

while(gjm!=2)

{

gjm=lookforrssi2(); //当rssi满足条件，返回gjm=2，跳出循环，进行下一步对于机智云的操作

StaNum=2; //得到此时的公交所在站点，衡量水平为StaNum，此时值为2。

}

crowd=ct2crowd(); //通过红外脉冲模块计算车载的拥挤程度，衡量水平为crowd

while(1) //ARC板子读取wifi发送的数据函数,MCU完成回复，必不可少。

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07:

{

mcu2wifi\_heartbeat(); //MCU回复心跳包

break;

}

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

break; //得到此时的StaNum与crowd，跳出while循环

}

EMBARC\_PRINTF("StaNum=%d,crowd=%d\r\n",StaNum,crowd);

mcu2wifi\_dev\_report\_status(StaNum,crowd); //将数据传入云端

delay\_ms(10000);

/\*————————————————————————第二个公交站牌的数据上传由小车完成，第二个，第四个站牌由云端读入改变数据，小灯发生相应改变。——————————\*/

while(gjm!=3)

{

gjm=lookforrssi3(); //当rssi满足条件，返回gjm=2，跳出循环，进行下一步对于机智云的操作

StaNum=3; //得到此时的公交所在站点，衡量水平为StaNum，此时值为2。

}

crowd=ct2crowd(); //通过红外脉冲模块计算车载的拥挤程度，衡量水平为crowd

while(1) //ARC板子读取wifi发送的数据函数,MCU完成回复，必不可少。

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07:

{

mcu2wifi\_heartbeat(); //MCU回复心跳包

break;

}

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

break; //得到此时的StaNum与crowd，跳出while循环

}

EMBARC\_PRINTF("StaNum=%d,crowd=%d\r\n",StaNum,crowd);

mcu2wifi\_dev\_report\_status(StaNum,crowd); //将数据传入云端

delay\_ms(10000);

/\*————————————————————————第三个公交站牌的数据上传由小车完成，第四个站牌由云端读入改变数据，小灯发生相应改变。——————————\*/

while(gjm!=4)

{

gjm=lookforrssi4(); //当rssi满足条件，返回gjm=2，跳出循环，进行下一步对于机智云的操作

StaNum=4; //得到此时的公交所在站点，衡量水平为StaNum，此时值为2。

}

crowd=ct2crowd(); //通过红外脉冲模块计算车载的拥挤程度，衡量水平为crowd

while(1) //ARC板子读取wifi发送的数据函数,MCU完成回复，必不可少。

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07:

{

mcu2wifi\_heartbeat(); //MCU回复心跳包

break;

}

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

break; //得到此时的StaNum与crowd，跳出while循环

}

EMBARC\_PRINTF("StaNum=%d,crowd=%d\r\n",StaNum,crowd);

mcu2wifi\_dev\_report\_status(StaNum,crowd); //将数据传入云端

delay\_ms(10000);

/\*————————————————————————第四个公交站牌的数据上传由小车完成，第四个站牌由云端读入改变数据，小灯发生相应改变。——————————\*/

return E\_SYS;

}

### 4.2.2算法二

static void delay\_ms( int z) //1ms

{

uint32\_t volatile x,y;

for(x=1400;x>0;x--)

for(y=z;y>0;y--);

}

void esp8266\_init(void)

{

// set pmod mux and uart map

set\_uart\_map(0xe4);

set\_pmod\_mux(PM1\_UR\_UART\_0 | PM1\_LR\_GPIO\_A| PM2\_I2C\_HRI| PM3\_GPIO\_AC| PM4\_I2C\_GPIO\_D| PM5\_UR\_GPIO\_C | PM5\_LR\_UART\_2 | PM6\_UR\_SPI\_M0 | PM6\_LR\_GPIO\_A );

// Get devices structure pointer

esp8266\_uart = uart\_get\_dev(ESP8266\_UART\_ID);

esp8266\_rst = gpio\_get\_dev(ESP8266\_RST\_PORT);

int32\_t ercd;

// Open devices

esp8266\_rst->gpio\_close();

if((ercd = esp8266\_rst->gpio\_open(ESP8266\_RST\_DIR) ) == E\_OK)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open successfully!!\r\n");

}

else if(ercd == E\_OPNED)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed,because opend with different direction!!\r\n");

esp8266\_rst->gpio\_control(GPIO\_CMD\_SET\_BIT\_DIR\_OUTPUT,(void \*)(ESP8266\_RST\_MASK));

}

else

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed, unknow why!!\r\n");

}

esp8266\_uart->uart\_close();

if(esp8266\_uart->uart\_open(ESP8266\_UART\_BAUDRATE) != E\_OK)

{

EMBARC\_PRINTF("ESP8266\_UART open failed!!\r\n");

}else

{

EMBARC\_PRINTF("ESP8266\_UART open successfully!!\r\n");

}

}

int main(void)

{

cpu\_lock();

//Board initialization;

board\_init();

//esp8266 initialization;

esp8266\_init();

cpu\_unlock();

uint8\_t StaNum=0;

uint8\_t Crowd=0;

uint16\_t rd\_cnt = 0;

uint8\_t read\_data[MAX\_READ\_CNT];

uint32\_t rd\_avail;

setupAP2();

while(1) //ARC板子读取wifi发送的数据函数

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

if(read\_data[9] == 01)

{

StaNum=read\_data[10];

board\_delay\_ms(500,0);

}

else if(read\_data[9] == 02)

{

Crowd=read\_data[11];

board\_delay\_ms(500,0);

}

EMBARC\_PRINTF("SN=%d,CR=%d\r\n",StaNum,Crowd); //临时显示debug

LEDRGB(StaNum,Crowd); //云端数据改变对应引起指示灯的变化

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07: mcu2wifi\_heartbeat();break; //MCU回复心跳包

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

}

return E\_SYS;

}

### 4.2.3算法三

static void delay\_ms( int z) //1ms

{

uint32\_t volatile x,y;

for(x=1400;x>0;x--)

for(y=z;y>0;y--);

}

void esp8266\_init(void)

{

// set pmod mux and uart map

set\_uart\_map(0xe4);

set\_pmod\_mux(PM1\_UR\_UART\_0|PM1\_LR\_GPIO\_A);

// Get devices structure pointer

esp8266\_uart = uart\_get\_dev(ESP8266\_UART\_ID);

esp8266\_rst = gpio\_get\_dev(ESP8266\_RST\_PORT);

int32\_t ercd;

// Open devices

esp8266\_rst->gpio\_close();

if((ercd = esp8266\_rst->gpio\_open(ESP8266\_RST\_DIR) ) == E\_OK)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open successfully!!\r\n");

}

else if(ercd == E\_OPNED)

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed,because opend with different direction!!\r\n");

esp8266\_rst->gpio\_control(GPIO\_CMD\_SET\_BIT\_DIR\_OUTPUT,(void \*)(ESP8266\_RST\_MASK));

}

else

{

EMBARC\_PRINTF("ESP8266\_RST\_PORT open failed, unknow why!!\r\n");

}

esp8266\_uart->uart\_close();

if(esp8266\_uart->uart\_open(ESP8266\_UART\_BAUDRATE) != E\_OK)

{

EMBARC\_PRINTF("ESP8266\_UART open failed!!\r\n");

}else

{

EMBARC\_PRINTF("ESP8266\_UART open successfully!!\r\n");

}

}

int main(void)

{

cpu\_lock();

//Board initialization;

board\_init();

//esp8266 initialization;

esp8266\_init();

cpu\_unlock();

uint8\_t StaNum=0;

uint8\_t Crowd=0;

uint16\_t rd\_cnt = 0;

uint8\_t read\_data[MAX\_READ\_CNT];

uint32\_t rd\_avail;

setupAP4();

while(1) //ARC板子读取wifi发送的数据函数

{

esp8266\_uart->uart\_control(UART\_CMD\_GET\_RXAVAIL,(void \*)(&rd\_avail));

rd\_cnt = (MAX\_READ\_CNT > rd\_avail)?rd\_avail:MAX\_READ\_CNT;

board\_delay\_ms(100,0);

if(rd\_avail>0)

{

esp8266\_uart->uart\_read((void \*)read\_data,rd\_cnt);

for(uint8\_t i=0;i<rd\_cnt;i++) EMBARC\_PRINTF("%2x ",read\_data[i]);

EMBARC\_PRINTF("\r\n");

//判断是否为合法数据包，合法数据包以固定包头0xFFFF开始。

if(read\_data[0]==0xFF && read\_data[1]==0xFF)

{

sn = read\_data[5];

switch(read\_data[4]) //读取命令位的代码

{

case 0x01: mcu2wifi\_product\_info();break; //MCU传入产品的Key与Secret

case 0x03: //Wifi控制设备，MCU读入改变的数据点

{

if(read\_data[9] == 01)

{

StaNum=read\_data[10];

board\_delay\_ms(500,0);

}

else if(read\_data[9] == 02)

{

Crowd=read\_data[11];

board\_delay\_ms(500,0);

}

EMBARC\_PRINTF("SN=%d,CR=%d\r\n",StaNum,Crowd); //临时显示debug

LEDRGB(StaNum,Crowd); //云端数据改变对应引起指示灯的变化

mcu2wifi\_wifi\_ctrl\_dev(read\_data); //MCU回复wifi的控制指令

break;

}

case 0x06: break;

case 0x07: mcu2wifi\_heartbeat();break; //MCU回复心跳包

case 0x0a: break;

case 0x0c: break;

case 0x0d: mcu2wifi\_wifi\_statu(read\_data);break; //wifi将最新的工作状态推送至MCU

default : EMBARC\_PRINTF("Undefine commond or error!!!\r\n");break;

}

}

}

}

return E\_SYS;

}

…

## 4.3小结

代码部分主要为C语言，在学习了AT指令和机智云协议后，对公交车部分还有站点部分的代码进行编写，有效实现了功能。

# 第五章 系统测试与分析

## 5.1系统测试指标

车到站的距离以及车上的人数。车到站的距离是通过车的WiFi发送rssi能否被站牌上的WiFi感应到衡量。车上的人数是车上上车红外模块扫到的人数减去车上下车红外模块扫到的人数。人数多、少或适中是通过程序设定的值来衡量的，十个人以内即是少，二十人以上即是多，中间范围内为适中

## 5.2 测试环境

模拟交通环境，为了实验室效果明显，本次测试在一个大小10平米的空间内，用人的行动模拟车走动，在空间的四角标好站点。用卡片阻挡红外对射模块模拟人上车下车。

### 5.2.1验证开发平台

1. 硬件平台：

ARC EM处理器是一个32位处理器内核，采用三级流水线结构，使功耗效率和面积效率都达到最佳化。这一系列处理器是ARC中功耗最低的，同时其可配置性能够针对性能和功耗进行优化，定制指令能够整合专有硬件以及广泛的生态系统支持。

该处理器主要有以下特点：

①支持ARCv2指令集，能混合执行16/32位指令，能最优化程序代码密度。

②支持用户模式和内核模式。

③支持最多64个寄存器，有效提升执行速度和优化程序大小。

④存储器寻址方式灵活简单，执行效率高。

⑤支持最多240个外部中断和16个优先级，支持快速中断和寄存器文件自动切换。

⑥支持指令和数据高速缓存。

⑦支持指令和数据紧密耦合存储器。

⑧丰富的可配置性。

⑨强大的用户扩展性。

⑩高效的低功耗机制，①结构级的门控时钟：在正常运行时，处理器会自动关闭没有使用的模块时钟；②多种低功耗模式：SLEEP指令可使处理器进入不同等级的低功耗模式，包括多种关闭时钟和关闭电源的模式；③支持动态的电压频率调整 。

1. 软件平台：

embARC是为ARC处理器，特别是ARC EM处理器的开发而提供的一个开源软件平台，包含大量的软件资源和说明文档，以帮助用户基于ARC处理器快速开发丰富的上层应用程序。例如，在embARC上可以下载针对ARC处理器在IOT领域开发的开源软件包，包括底层驱动、操作系统和中间件等。

embARC不仅提供了与FreeRTOS和Contiki OS等操作系统的接口，在embARC上开发的上层应用程序也可以很好地在GNU工具链或MetaWare工具链中编译和调试。

### 5.2.2测试方案

1、将各站点位置排好，各站点板子插电源，并触发初始化。

2、车出发经过第一个站点安远门，用卡片档上车红外对射模块，模拟上车，上车六个人，观察站牌灯变化。

3、车继续前进经过第二个站点安定门，继续模拟上车九个人，观察灯牌变化。

4、车继续前进经过第三个站点长乐门，继续模拟上车，超过二十人，观察灯牌变化。

5、车继续前进经过第四个站点永宁门，用卡片档下车红外对射模块，模拟下车，下车到个位数人，观察灯牌变化。

## 5.3测试结果

车到安远门站时，灯牌对应安远门的第一个灯亮，且灯的颜色为绿色；



图5.1 小车到第一站时站牌的显示

车到安定门站时，灯牌对应安定门的第二个灯亮，前灯灭，灯的颜色为黄色；



图5.2 小车到第二站时站牌的显示

车到长乐门站时，灯牌对应长乐门的第三个灯亮，前灯灭，且灯的颜色为红色；



图5.3 小车到第三站时站牌的显示

车到永宁门站时，灯牌对应永宁门的第四个灯亮，前灯灭，且灯的颜色为绿色。



图5.3 小车到第三站时站牌的显示

## 5.3结果分析

经过WiFi间的感应，以及上车下车人数的模拟，整的来说达到了预期的效果。智能公交站牌实现了车位置的报站以及车拥挤人数的灯色显示。

# 第六章 总结展望

本设计主要目的在于利用ARC处理器实现公交站牌的报站车拥挤程度的显示。到目前为止，通过对ARC处理器，红外对射模块，LED RGB，WiFi模块的深入研究，完成了硬件的拼接，软件的编写，并且软件利用C语言进行编程，增加了程序的可移植性和灵活性。

归纳起来主要做了如下几方面的工作：1、对WiFi模块、红外对射模块、LED RGB各种模块的工作原理进行学习，并进行测试；2、对ARC处理器有了深入的理解，学会了利用各种接口增加外设；3、利用C语言进行程序设计，实现各个模块功能。4、机智云设备联动的实现，协议的学习和代码AT指令编写。根据上面论述结合测试数据可以看出本次设计基本完成了设计任务和要求。

过月余的忙碌，产品终于成型，在竞赛中我们团队学到了很多东西，团结合作，细致分工，查询资料，自学能力得到提高，深入了解了嵌入式系统的开发与编程，学会了熟练使用ARC EM处理器，掌握了AT指令和机智云协议的编写，学习了如何进行系统设计及相关技巧，为今后的工作和学习奠定了坚实的基础。

本产品的意义所在：响应智慧城市号召，为智慧城市交通做出一份贡献，为人们提供便利。

随着科技的发展，智慧城市共建，智慧交通深入，未来还有很多需要努力和改进的地方：

1.引入语音技术，在机智云中加入语音系统，是智能公交车的受众面更广，为盲人也提供服务；

2.引入充电技术，候车的时候为候车人提供充电服务；

3.资金允许的情况下，引入触屏屏幕，为候车人提供娱乐性服务。MCU处理器可以实现多辆车的信息查询；

4.提供交通换乘信息，更大程度为乘客提供便利，使得城市交通系统更加智能化。

# 参考文献

[1] 雷鑑铭.ARC EM处理器嵌入式系统开发与编程[M].北京：机械工业出版社，2015.10.

[2] 孟庆洪，侯宝稳.ARM嵌入式系统开发与编程[M].北京：清华大学出版社，2011:1-3.

[3] 马文华.嵌入式系统设计与开发[M].北京：科学出版社，2011.6.

[4] 沈建华，郝立平.ARM Cortex-M0+微控制器原理与应用—基于Atmel SAM D20 系列[M].北京：北京航空航天大学出版社，2014:34-63，100-155.

[5] 刘晓辉，夏建生.实用电子元器件与电路基础（第二版）.北京：电子工业出版社，2009.35-63.

[6] 胡汉才．单片机原理及其接口技术[M]（第2版）.北京：清华大学出版社．2004.49-77．

[7] 杨欣．电子设计从零开始[M].北京:清华大学出版社,2005.28-102.

[8] 谭浩强．C程序设计[M]（第三版）.北京：清华大学出版社．2005．37-65．

[9] Synopsys ARC.Design Ware ARC EM Databook[EB/OL].Version 5756-034,December 2014.http://www.synopsys.com/IP/ProcessorIP/ARCprocessors/Pages/default.aspx.

[10] Donald A.Neamen.Electronic Circuits Analysis and Design ,2nd ed.McGraw-Hill Companies，2001.

[11] Intel: Benjamin Jun，Paul Kocher．The lntel Random Number Generator[J]．White Paper Prepared for lntel Corporation，April 22，l999：4-5．