**自行车租赁监管系统**

**开发记录文档**

**计算机科学与技术学院**

**2020.6**

#### 目录

[1. 车棚终端硬件程序开发 3](#_Toc23462)

[1.1. 设计 3](#_Toc15849)

[1.1.1. 功能流程简介 3](#_Toc15143)

[1.1.2. 相关代码 3](#_Toc24520)

[1.1.3. 逻辑流程图 9](#_Toc17287)

[1.1.4. 详细设计 10](#_Toc28500)

[1.2. 调试 13](#_Toc17153)

[1.2.1. 实际与界面交流运行图 13](#_Toc28206)

[1.2.2. 问题及解决 14](#_Toc14267)

[2. 自行车终端程序开发 15](#_Toc198)

[2.1. 设计 15](#_Toc24514)

[2.1.1. 界面设计图 15](#_Toc21222)

[2.1.2. 相关代码 16](#_Toc19934)

[2.1.4. 逻辑流程图 18](#_Toc4539)

[2.2. 调试 19](#_Toc20998)

[2.2.1. 程序运行截图 19](#_Toc12017)

[2.2.2. 出现问题和解决 21](#_Toc14974)

[3. 车辆信息管理页面 22](#_Toc32218)

[3.1. 设计 22](#_Toc23274)

[3.1.1. 界面设计 22](#_Toc706)

[3.1.2. 相关代码 22](#_Toc28880)

[3.1.3. 插入数据流程图 26](#_Toc19925)

[3.1.4. 显示所有的数据流程图 27](#_Toc31078)

[3.1.5. 查询车辆信息流程图 28](#_Toc17988)

[3.2. 调试 29](#_Toc27298)

[3.2.1 运行截图 29](#_Toc12177)

[3.2.2. 问题 32](#_Toc4574)

# 车棚终端硬件程序开发

## 设计

### 功能流程简介

车棚终端通过光敏电阻控制灯的开关，可以在天黑得时候达到自动开灯的功能。同时通过对IC卡的读写，提供了充值扣款的功能，同时向IC卡中供自行车终端读取的时间，写入的时间将决定自行车车锁开启的时间。通过这样将两个系统连接在一起，构成一个完整的自动化租车系统。

### 相关代码

**光敏控制灯的核心代码：**

#define LED1 P1\_0 //定义P1.0口为LED1控制端

#define DATA\_PIN P1\_5 //定义P1.5口为传感器输入端

void InitLed(void)

{

P1DIR |= 0x01; //P1.0定义为输出口

}

InitLed();

P1DIR &= ~0x20; //P1.5定义为输入口

if(DATA\_PIN == 1) //当光敏电阻处于黑暗中时P0.5高电平,LED1熄灭

{

LED1 = 0;

Delay\_1ms(1000);

}

else

{

LED1 = 1; //检测到光线时P0.5为低电平LED1亮

Delay\_1ms(1000);

}

**充值扣款的核心代码：**

void IC\_test()

{

uchar find=0xaa;

while(1)

{

//16进制转ASC码

char i;

char Card\_Id[8]; //存放32位卡号

char Card\_type[8];

ret = PcdRequest(0x52,ucTagType);//寻卡

if(ret != 0x26)

ret = PcdRequest(0x52,ucTagType);

if(ret != 0x26)

find = 0xaa;

if((ret == 0x26)&&(find == 0xaa))

{

for(i=0;i<4;i++)

{

Card\_type[i\*2]=asc\_16[ucTagType[i]/16];

Card\_type[i\*2+1]=asc\_16[ucTagType[i]%16];

}

UartSend\_String(Card\_type,4);

if(PcdAnticoll(ucTagType) == 0x26)//防冲撞

{

UartSend\_String("The Card ID is: ",16);

//16进制转ASC码

for(i=0;i<4;i++)

{

Card\_Id[i\*2]=asc\_16[ucTagType[i]/16];

Card\_Id[i\*2+1]=asc\_16[ucTagType[i]%16];

}

UartSend\_String(Card\_Id,8);

UartSend\_String("\n",1);

ret=PcdSelect(ucTagType);//选卡

if(ret==0x26){

UartSend\_String("select sucess",13);

ret=PcdAuthState(0x60,1,DefaultKey,ucTagType);

if(ret==0x26){

UartSend\_String("auth 1 sucess",13);

break;

}

}

}

find = 0x00;

}

}

}

**主函数：**

void main()

{

InitIO();

PcdReset();

M500PcdConfigISOType('A');//设置工作方式

InitLed();

P1DIR &= ~0x20; //P1.5定义为输入口

while(1)

{

if(DATA\_PIN == 1) //当光敏电阻处于黑暗中时P0.5高电平,LED1熄灭

{

LED1 = 0;

Delay\_1ms(1000);

}

else

{

LED1 = 1; //检测到光线时P0.5为低电平LED1亮

Delay\_1ms(1000);

}

if(RXTXflag == 1) //接收状态

{

if(temp != 0)

{ //接收状态指示

if((temp!='#')&&(datanumber<50)) //’＃‘被定义为结束字符，最多能接收50个字符

{

Recdata[datanumber++] = temp;

}

else

{

RXTXflag = 3; //进入发送状态

}

if(datanumber == 50)

RXTXflag = 3;

temp = 0;

}

}

if(RXTXflag == 3) //发送状态

{

UartSend\_String("send:",5);

U0CSR &= ~0x40; //不能收数

UartSend\_String(Recdata,datanumber);

UartSend\_String("\n",1);

U0CSR |= 0x40; //允许接收

if(Recdata[0]=='0'){

char mm[50];

int i,j=1;

for(i=0;i<datanumber;i++){

mm[i]=Recdata[j];

j++;

}

datanumber--;

U0CSR &= ~0x40;

UartSend\_String("down",4);

UartSend\_String("\n",1);

UartSend\_String(mm,datanumber);

UartSend\_String("\n",1);

U0CSR |= 0x40;

Money[0]=String2Int(mm);

IC\_test(); //检测IC卡

ret=PcdValue(0xc0,1,Money);

MFRC522\_Read\_Wallet(1,pdata);

if(ret==0x26){

UartSend\_String("yue sucess",10);

}

PcdWrite(2,RFID1);

ret=PcdValue(0xc1,2,Money);

if(ret==0x26)

{

MFRC522\_Read\_Wallet(2,pdata);

}

}

else if(Recdata[0]=='1'){

char mm[50];

int i,j=1;

for(i=0;i<datanumber;i++){

mm[i]=Recdata[j];

j++;

}

datanumber--;

UartSend\_String("up",2);

UartSend\_String("\n",1);

UartSend\_String(mm,datanumber);

UartSend\_String("\n",1);

Money[0]=String2Int(mm);

IC\_test(); //检测IC卡

ret=PcdValue(0xc1,1,Money);

if(ret==0x26){

ret=PcdRead(1,pdata);

if(ret==0x26){

for(i=0;i<32;i++)

{

Card\_mess[i\*2]=asc\_16[pdata[i]/16];

Card\_mess[i\*2+1]=asc\_16[pdata[i]%16];

}

UartSend\_String(Card\_mess,32);

UartSend\_String("\n",1);

}

}

MFRC522\_Read\_Wallet(1,pdata);

if(ret==0x26){

UartSend\_String("yue sucess",10);

}

}

else if(Recdata[0]=='c'){

IC\_test(); //检测IC卡

ret=PcdWrite(1,RFID1);

if(ret==0x26){

UartSend\_String("clear sucess",12);

ret=PcdRead(1,pdata);

if(ret==0x26){

int i;

for(i=0;i<32;i++)

{

Card\_mess[i\*2]=asc\_16[pdata[i]/16];

Card\_mess[i\*2+1]=asc\_16[pdata[i]%16];

}

UartSend\_String(Card\_mess,32);

UartSend\_String("\n",1);

}

MFRC522\_Read\_Wallet(1,pdata);

}

}

RXTXflag = 1; //恢复到接收状态

datanumber = 0; //指针归0

memset(Recdata, 0, sizeof(Recdata));

}

}

}

**中断函数：**

#pragma vector = URX0\_VECTOR

\_\_interrupt void UART0\_ISR(void)

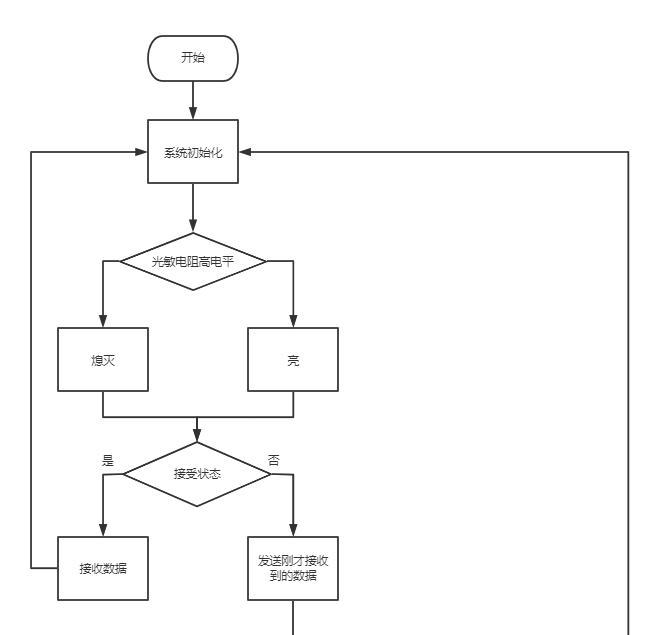
{

URX0IF = 0; //清中断标志

temp = U0DBUF;

}

### 逻辑流程图



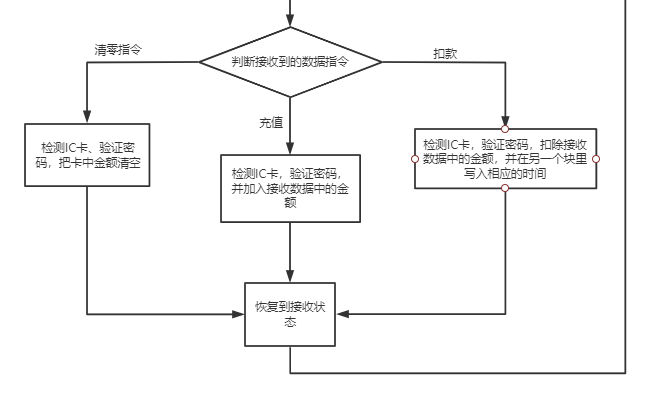


图1 逻辑流程图

### 详细设计

**相关函数作用及其参数：**

MFRC522\_Read\_Wallet(uchar block,unsigned char \*pData)

函数名：MFRC522\_Read\_Wallet

描 述：读出钱包的金额

说 明：

参 数：block 钱包的位置（第几块）,pData 卡的内容

返回值：成功返回MI\_OK

Transform\_16\_10(uint num)

函数名：Transform\_16\_10

描 述：16进制装换成10进制

说 明：

参 数：num 要转换的数据

返回值：无

PcdBakValue(unsigned char sourceaddr, unsigned char goaladdr)

功 能：备份钱包

参数说明: sourceaddr[IN]：源地址

goaladdr[IN]：目标地址

返 回: 成功返回MI\_OK

PcdValue(unsigned char dd\_mode,unsigned char addr,unsigned char \*pValue)

/功 能：扣款和充值

参数说明: dd\_mode[IN]：命令字

0xC0 = 扣款

0xC1 = 充值

addr[IN]：钱包地址

pValue[IN]：4字节增(减)值，低位在前

返 回: 成功返回MI\_OK

PcdRead(unsigned char addr,unsigned char \*pData)

功 能：读取M1卡一块数据

参数说明: addr[IN]：块地址

pData[OUT]：读出的数据，16字节

返 回: 成功返回MI\_OK

PcdWrite(unsigned char addr,unsigned char \*pData)

功 能：写数据到M1卡一块

参数说明: addr[IN]：块地址

pData[IN]：写入的数据，16字节

返 回: 成功返回MI\_OK

PcdAuthState(unsigned char auth\_mode,unsigned char addr,unsigned char \*pKey,unsigned char \*pSnr)

功 能：验证卡片密码

参数说明: auth\_mode[IN]: 密码验证模式

0x60 = 验证A密钥

0x61 = 验证B密钥

addr[IN]：块地址

pKey[IN]：密码

pSnr[IN]：卡片序列号，4字节

返 回: 成功返回MI\_OK

PcdSelect(unsigned char \*pSnr)

功 能：选定卡片

参数说明: pSnr[IN]:卡片序列号，4字节

返 回: 成功返回MI\_OK

PcdAnticoll(unsigned char \*pSnr)

功 能：防冲撞

参数说明: pSnr[OUT]:卡片序列号，4字节

返 回: 成功返回MI\_OK

PcdRequest(unsigned char req\_code,unsigned char \*pTagType)

功 能：寻卡

参数说明: req\_code[IN]:寻卡方式

0x52 = 寻感应区内所有符合14443A标准的卡

0x26 = 寻未进入休眠状态的卡

pTagType[OUT]：卡片类型代码

0x4400 = Mifare\_UltraLight

0x0400 = Mifare\_One(S50)

0x0200 = Mifare\_One(S70)

0x0800 = Mifare\_Pro(X)

0x4403 = Mifare\_DESFire

返 回: 成功返回MI\_OK

## 调试

### 实际与界面交流运行图



图2 运行图1

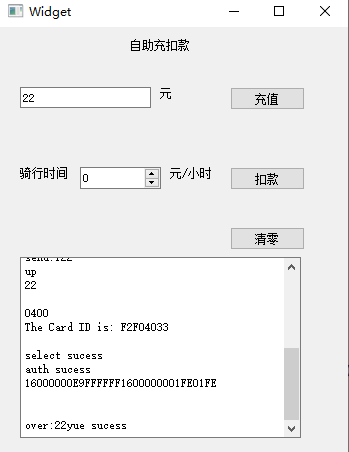


图3 运行图2

### 问题及解决

主要问题是RC522的充值和扣款程序都没有相应的示例和文档，不清楚操作原理，想要读到写到IC卡中的金额更是没有相应的函数，只好通过查阅大量的资料和相关的代码，但是由于大部分是基于其他的单片机开发的，代码差别过大，花了不少的时间才搞懂了代码的运行过程原理和相应参数的意义和传递过程。并且写了读取IC卡余额的函数，以至于可以简单调用程序进行IC卡余额的读取。

# 自行车终端程序开发

## 设计

### 相关功能简介

自行车终端可以读取到IC卡中存储的时间，并把时间参数传给定时器，模拟车锁打开，定时器启动，定时器运行相应时间后模拟车锁关闭。同时在定时器运行过程中将温湿度传感器获取到的温湿度发送到LCD上显示。

## 相关代码

##### 主函数：

void main()

{

InitIO();

InitLed();

LCD\_Init(); //oled 初始化

LCD\_CLS(); //屏全亮

LCD\_welcome();

InitT1();

PcdReset();

M500PcdConfigISOType('A');//设置工作方式

while(1){

IC\_test();

ret=MFRC522\_Read\_Wallet(2,pdata);

if(ret==0x26){

while(1){

if(IRCON>0){

IRCON=0;

uchar temp[3];

uchar humidity[3];

uchar strTemp[15]="T:";

uchar strHumidity[5]="H:";

memset(temp, 0, 3);

memset(humidity, 0, 3);

DHT11(); //获取温湿度

//将温湿度的转换成字符串

temp[0]=wendu\_shi+0x30;

temp[1]=wendu\_ge+0x30;

humidity[0]=shidu\_shi+0x30;

humidity[1]=shidu\_ge+0x30;

strcat(strTemp,temp);

strcat(strTemp," ");

strcat(strHumidity,humidity);

strcat(strTemp,strHumidity);

LCD\_P8x16Str(0, 4,strTemp);

if(++count>=money){

count=0;

LED1=!LED1;

T1CTL = 0x0c;

break;

}

}

}

}

}

}

##### 读取时间函数：

char MFRC522\_Read\_Wallet(uchar block,unsigned char \*pData)

{

uchar status=MI\_ERR;

//uint money; //最后的结果

status = PcdRead(block,pData); //读块

if(status==0x26){

money=pData[1];

money<<=8; //扩大8倍

money+=pData[0];

money=Transform\_16\_10(money);

Money[0]=money;

char yue[20];

Int2String(money,yue);

//UartSend(block);

UartSend\_String("over:",5);

UartSend\_String(yue,strlen(yue));

UartSend\_String("\n",1);

LED1=!LED1;

T1CTL = 0x0d; //启动定时器

}

//PcdValue(0xc0,2,Money);

return status;

}

### 2.1.4. 逻辑流程图

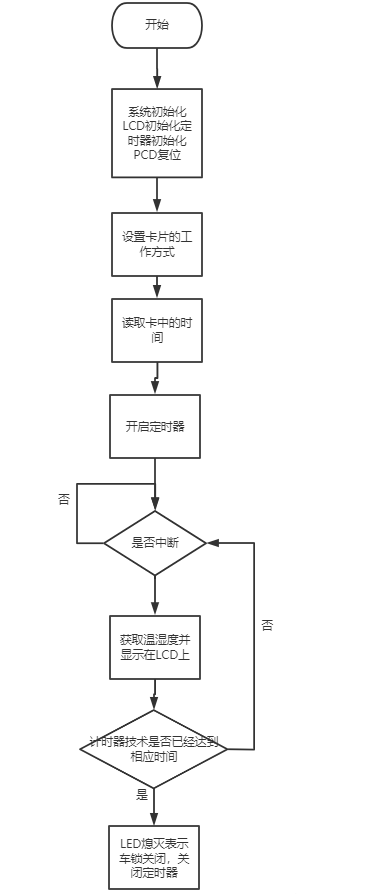


图4 逻辑流程图

## 调试

### 程序运行截图

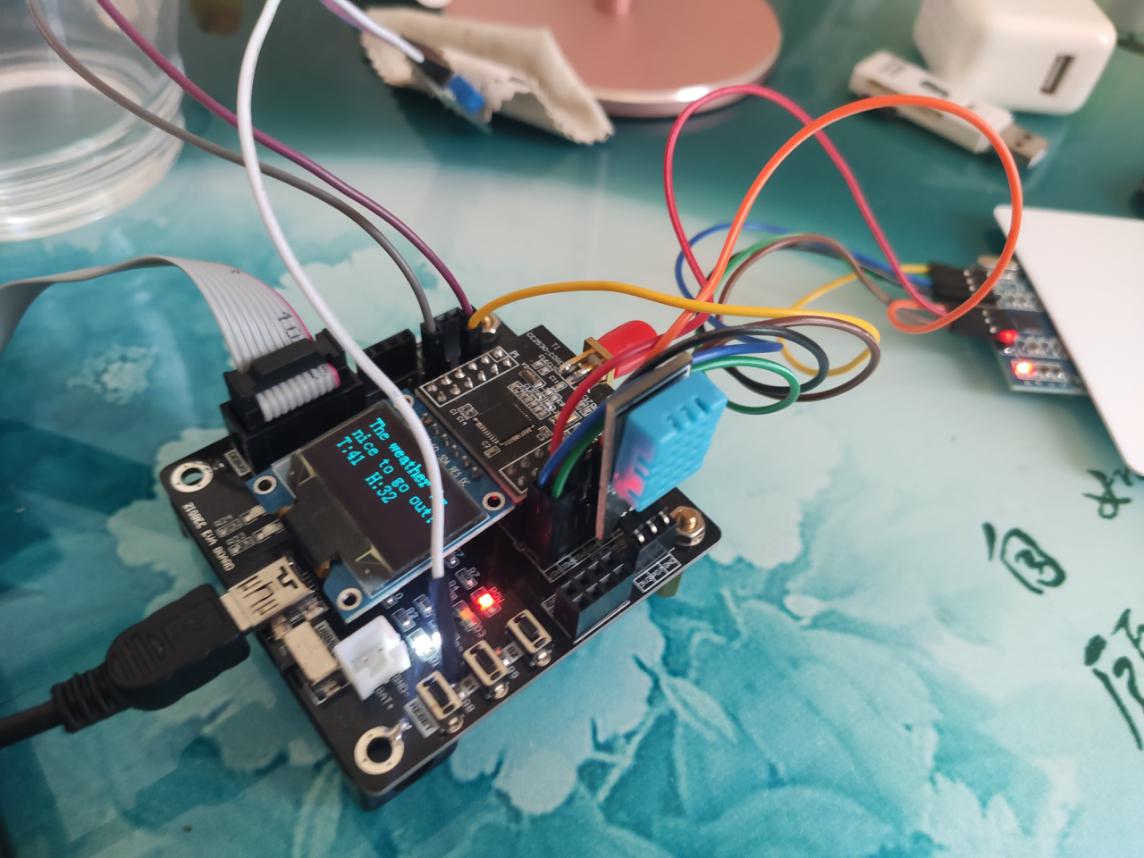


图9 LCD显示当前温湿度

### 出现问题和解决

出现的主要问题就是系统的并发性，由于温湿度需要一个循环函数一直读取和在LCD上显示，同时还要再循环里不断检测IC卡和读取卡内的数据，同时还要判断定时器的时间。最后在判断定时器是否发生中断的函数中加入了温湿度传感器相关的代码，因为定时器开启时大概每0.5秒发生一次中断，所以可以实现相关的功能，同时对IC卡的检测，数据的读取等方面也没有影响，提高了系统的并发程度。