# 基于 C# 的射频卡读写原理及实现

张可可[1]，熊庆宇[2]
（重庆大学自动化学院，重庆 400044）

**摘 要**：介绍了射频卡的硬件结构和工作原理，给出了一套对射频卡进行数据采集和实时处理的软件设计方案，并采用 C# 语言编写了关键的程序代码。

**关键词**：射频卡；读卡器；定时器；动态链接库
**中图分类号**：TP391 **文献标识码**：B **文章编号**：1674-7720(2010)12-0026-03

## Principles of reading and writing of RF card and development with C# program language

ZHANG Ke Ke[1], XIONG Qing Yu[2]
(Automation College of ChongQing University, Chongqing 400044, China)

**Abstract**: It gives a set of data acquisition and real-time processing software design of RF card on the basis of introducing the hardware structure and working principle about RF card in detail, and gives the code with C# program language.

**Key words**: RF card; card reader; timer; dynamic link library

射频卡又称非接触式 IC 卡，它将 RFID 和 IC 技术完美结合，使卡片能够在不需要电源及与读卡器不接触的情况下正常工作。目前射频卡已经广泛使用在社会生活的各个领域，如银行卡、企业一卡通系统等。由于射频卡具有使用人群的密集性以及使用时间不确定性的特点，就要求读卡器能够对射频卡进行实时准确的数据采集并通过数据线把采集到的数据传送给计算机，通过特定的处理软件进行快速处理，并将处理结果反馈回射频卡，从而实现计算机与射频卡信息的双向交互，满足人们特定的要求。本文通过定时器技术实现对射频卡信息的实时采集和交互处理，利用定时器的定时触发功能实现对射频卡读写函数的全天候循环调用，减轻系统的负载、优化系统的进程、提高系统的稳定性，从而保持计算机和射频卡协同高效地工作。

## 1 射频卡硬件结构与工作原理

本课题中射频卡采用业界广泛使用的由荷兰飞利浦公司生产的 M1 卡，M1 卡主要有射频天线和 ASIC 两部分组成，如图 1[1]。射频天线是由特制的磁感线圈绕制而成，用来接收读卡器发出的固定频率的电磁波。ASIC 主要由高速射频 RF 接口、数据读写控制单元、存储工具 EEPROM 构成。当读卡器对射频卡进行读写操作时，读卡器会持续发出一组频率固定的电磁波，电磁波的频率与 M1 卡内置的 LC 谐振模块的谐振频率相同，从而造成 LC 谐振模块发生共振，使谐振电路的电容内产生电荷，这个电容通过特殊的传输装置单向传输到另外一个电容聚集起来。当积累的电荷电压达到 2 V 时，此电荷实际上可以作为一个电源向卡内的各种电路装置供电，从而实现读卡器对射频卡的读写操作。

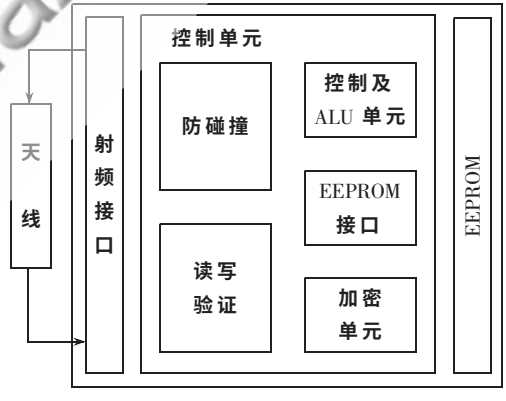


图 1 射频卡的结构图

高速射频 RF 接口的主要功能是用来接收通过 LC 谐振电路产生的电源电压以及谐振电路本身的复位信号和时钟信号。数据读写控制单元的主要功能是对射频接口传递的数据进行调制和解密并对数据按照特定的

步骤与读卡器进行数据的交互处理。读卡器与计算机连接的串口初始化成功后，就开始在读卡器射频感应的工作范围内寻找射频卡[2]。如果同时感应到多张射频卡，读卡器会启动反冲突机制控制模块选定其中的一张。选定要处理的卡之后，读写器就确定要访问的扇区号，并对该扇区密码进行密码校验，在3次相互认证之后就可以通过加密流进行通讯，对读卡器进行读写操作，操作成功后启动报警控制模块，提示操作成功，同时挂起该张卡。EEPROM是射频卡的存储单元，用来保存读卡器写入的信息。M1射频卡存储空间是8 KB。存储空间分为16个扇区，每个扇区又分为4个块，每个块内存大小为16 B。64个块按物理排序命名，序号从0块一直到63块。其中0块保存的是射频卡的序列号，出厂时由厂家直接写入，不能更改。另外，每个扇区的第4块是该扇区的密码存储块，其中包括两套密码以及密码读取控制字节。其余3块是数据块，可以存储数据并进行相应数据操作[3]，如图2所示。


图2 射频卡的存储空间结构图

## 2 软件设计流程与程序实现

基于射频卡使用环境及对数据处理实时性的特殊要求，必然要求读卡器处于一种不间断的监测状态，能够对进入读卡器感应区域的射频卡进行快速稳定的数据采集，并把这种处理结果实时传输给相连的计算机，通过专门的软件进行信息的交互处理[4]。考虑到这些要求，在软件的设计过程中使用C#中的timer控件来满足这种要求，利用Timer控件的定时激发功能，使读卡器能够不间断地检测是否有卡进入感应区域。首先初始化串口，保证读卡器和计算机的正常连接，如果读卡器没有检测到有射频卡处于工作区，就一直保持检测状态。如果读卡器检测到工作范围内有卡，就按照正常读写操作流程(如图3)对射频卡进行操作，一张卡操作完成后，读卡器会自动报警提示操作成功并挂起这张卡。在这种情况下，除非把这张卡移除工作区，否则读卡器将无法继续正常工作。

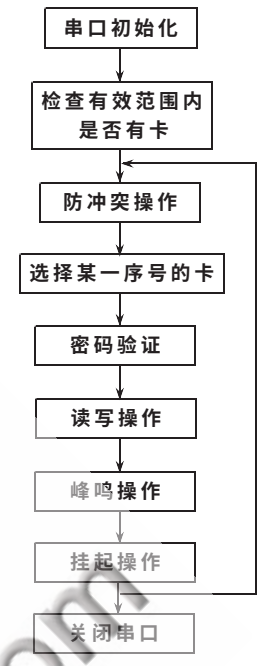


图3 射频卡读写操作流程

在分析了射频卡的工作原理和软件流程后，本文用C#语言来编写具体的程序代码，C#是微软公司发布的一种面向对象的、运行于.NET Framework之上的高级程序设计语言[5]。为了便于产品的开发，厂家已经附带给出了开发射频卡程序所需要的动态连接库。C#语言可以直接调用给动态连接库，只需要在程序中加以引用说明即可。本课题采用的读卡器为双面D8读卡器，附带的动态链接库文件为dcrf32.dll。此文件中包含了常用的射频卡读写操作等系列函数。实现对射频卡写数据操作的部分关键代码如下：

```
…
[DllImport("dcrf32.dll")]
public static extern int dc_init(short port, int baud);
[DllImport("dcrf32.dll")]
public static extern short dc_request (int icdev, char _Mode, ref uint TagType);
[DllImport("dcrf32.dll")]
public static extern short dc_select(int icdev, uint _SecNr, byte[] _Size);
[DllImport("dcrf32.dll")]
public static extern short dc_authentication (int icdev, int _Mode, int _SecNr);
[DllImport("dcrf32.dll")]
public static extern int dc_beep (int icdev, short _Msec);
[DllImport("dcrf32.dll")]
public static extern int dc_pro_halt(int icdev);
[DllImport("dcrf32.dll")]
public static extern short dc_exit(int icdev);
private void Card_Read();
{
    _icdev=dc_init(Form3.Com, Form3.botelv);
                                        //串口初始化
    if (_icdev <= 0)
    {MessageBox.Show("串口初始化失败！");
        return;
    }
```

```
byte[] name = System.Text.Encoding.Default.GetBytes
                (txtN.Text);
byte[] sex = System.Text.Encoding.Default.GetBytes
                (txtS.Text);
if (name.Length > 16)
{
    MessageBox.Show("超过规定的数据长度,写入
                                    失败");
    txtN.Text = null;
    return;
}
if (sex.Length > 16) //数据长度检测
{
    MessageBox.Show("超过规定的数据长度,写入
                                    失败");
    txtS.Text = null;
    return;
}
int st;
ulong icCardNo = 0;
char tt = (char)0;
st = dc_card(IcDev, tt, ref icCardNo); //寻卡操作
if (st ! = 0)
{   txtCardId.Text = "";
    txtN.Text = "";
    txtS.Text = "";
    MessageBox.Show("寻卡失败！");
    return;
}
int sector = 0;
st = dc_authentication(IcDev, 0, sector); //密码验证
if (st ! = 0)
{
    MessageBox.Show("验证密码失败！");
    return;
}
try
{
    string dt = txtS.Text;
    st = dc_write(IcDev, 2, dt);    //射频卡写操作
    dc_beep(IcDev, 10);             //蜂鸣操作
    st=dc_halt(IcDev) ;             //挂起操作
    MessageBox.Show("修改成功");
}
catch
{
MessageBox.Show("更改卡中信息失败");
}
dc_exit(IcDev); //关闭串口
}
private void timer1_Tick(object sender, EventArgs e)
                                    //定时器操作
{
    timer1.Interval = 1000;
    timer1.Start();
    Card_Read();
}
```

实际应用结果表明，采用 C# 语言结合定时器的特有功能编写的射频卡读写控制程序运行稳定，能够很好地满足工作现场的需要。在读卡器对射频卡进行读写操作的同时，并不影响软件系统其他模块的操作，具有较强的实用意义。

参考文献

[1] 杨瑞,彩虹.射频卡多线程读写原理及其实现[J].计算机与信息技术,2006(2):1-3.

[2] 苏明强,刘伟.高性价比的 MiFARE 卡读写模块的设计.[J].微计算机信息,2006,22(5-2):1-2.

[3] 何泉,曹刚. 基于射频识别技术的联机型门禁系统设计[J].微计算机信息, 2008,24(5-2):201-202.

[4] 刘继平,谭耀辉. 基于 MSP430 单片机的实用射频卡读卡电路设计[J].现代电子技术,2008(10):171-173.

[5] 王卓人,邓晋钧,刘宗祥. IC 卡的技术与应用. 北京：电子工业出版社,1999:49.

(收稿日期:2010-02-25)

作者简介：

张可可,男,1985 年生,硕士,主要研究方向:射频及软件开发。

熊庆宇,男,1963 年生,教授,主要研究方向:神经网络及模糊控制。