中山大学移动信息工程学院

物联网技术实验报告

旦增罗布

RFID读写器上位机软件的设计与实现

小组成员:

旦增罗布 14353049

白冰 14353002

陈阳 14353038

旦真 13354052

指导教师：

胡建国

目 录

[第一章 RFID读写器上位机软件设计概要（白冰） 2](#_Toc484757185)

[1.1 RFID读写器上位机软件设计概述 2](#_Toc484757186)

[1.2 RFID读写器上位机软件设计安排 2](#_Toc484757187)

[第二章 RFID读写器上位机软件详细设计过程 2](#_Toc484757188)

[2.0 界面设计（陈阳） 2](#_Toc484757189)

[2.1 开启设备（陈阳） 2](#_Toc484757190)

[2.2 获取卡片信息(旦真) 2](#_Toc484757191)

[2.3 Tab选项卡（旦真） 2](#_Toc484757192)

[2.4 LED设置 （旦增罗布） 2](#_Toc484757193)

[2.5 读写扇块区（旦增罗布） 3](#_Toc484757194)

[2.6 电子钱包（白冰） 7](#_Toc484757195)

[2.7 上机网管 7](#_Toc484757196)

[2.8 历史记录 7](#_Toc484757197)

[第三章 RFID读写器上位机软件功能测试 7](#_Toc484757198)

[3.1 开启设备(陈阳) 7](#_Toc484757199)

[3.2 获取卡片信息（旦真） 7](#_Toc484757200)

[3.3 读写扇区（旦增罗布） 8](#_Toc484757201)

[3.4 电子钱包（白冰） 8](#_Toc484757202)

[第四章 RFID读写器上位机软件功能创新（旦增罗布） 8](#_Toc484757203)

[第五章 RFID读写器上位机软件项目总结 8](#_Toc484757204)

下面仅描述个人完成的内容。

# 第一章 RFID读写器上位机软件设计概要（白冰）

## 1.1 RFID读写器上位机软件设计概述

## 1.2 RFID读写器上位机软件设计安排

# 第二章 RFID读写器上位机软件详细设计过程

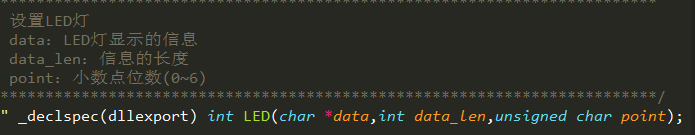
## 2.0 界面设计（陈阳）

## 2.1 开启设备（陈阳）

## 2.2 获取卡片信息(旦真)

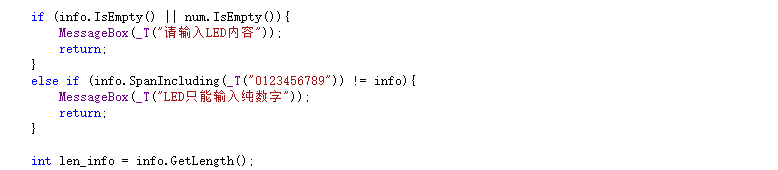
## 2.3 Tab选项卡（旦真）

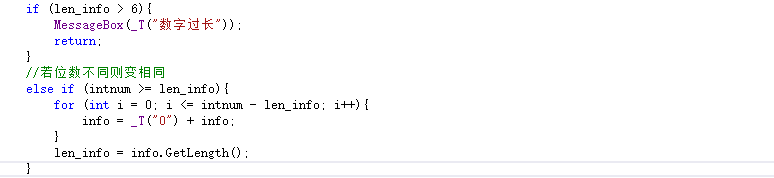
## 2.4 LED设置 （旦增罗布）

LED设置是通过LED函数实现，所以只要获取满足相应类型的数据作为LED函数参数即可实现。

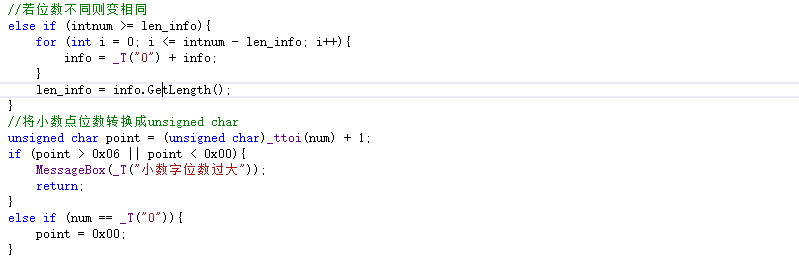
1.LED灯显示的信息data的获取：

1.从信息编辑框中获取CString类型的数据，之后判断是否满足非空、为纯数字、位数小于6这三个条件 ，不满足其一任何一项则报错并跳出响应事件。代码如下：

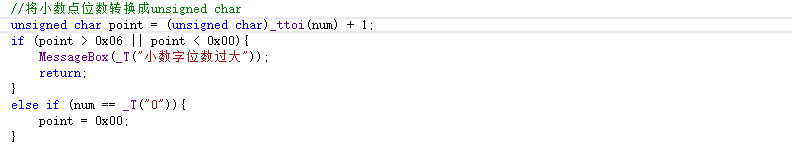




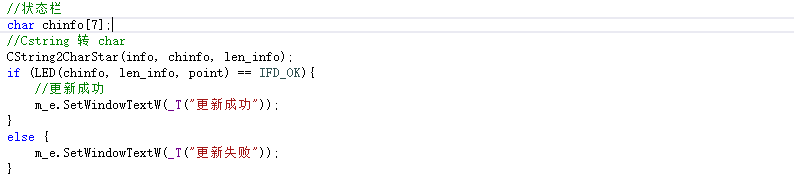
1. 从小数位编辑框中获取CString类型小数位数，之后判断是否满足非空、是否与data的位数一致的两个条件。如果满足非空但是位数显示大于data位数，则给data前置补零到相等为止，代码见下：



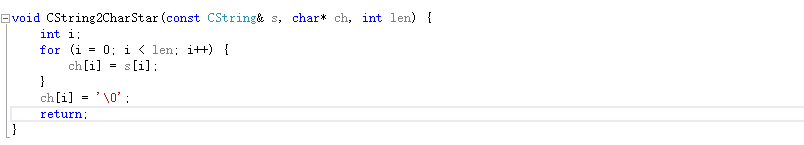
之后将CString类型小数位转成unsigned char类型并要求不超过6，超过则报错并return，代码见下：



最后再满足以上条件后CSring类型的data转为unsigned char并作为LED函数的参数，通过返回值来确定是否操作成功，代码见下：



从CString到unsigned char的转变需要函数CString2CharStar，见下：

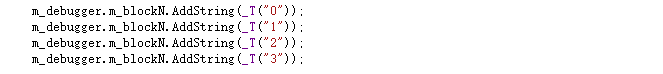


## 2.5 读写扇块区（旦增罗布）

这里有三个事件响应要写读块、写块、读扇区。这里需要了解扇区是什么东西，在TA提供的PPT中有提到，所以不赘述。然后是组合框初始化，怎样让我们有选项。

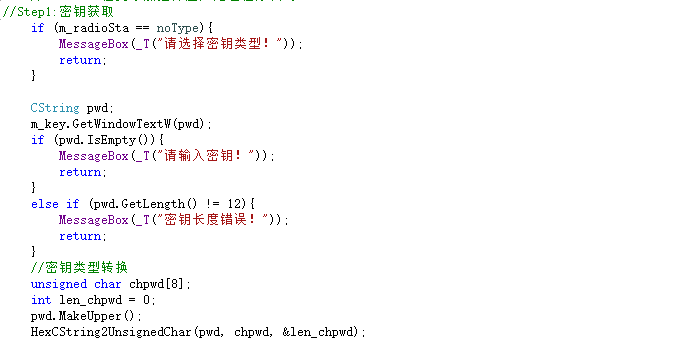
首先讲下组合框初始化：

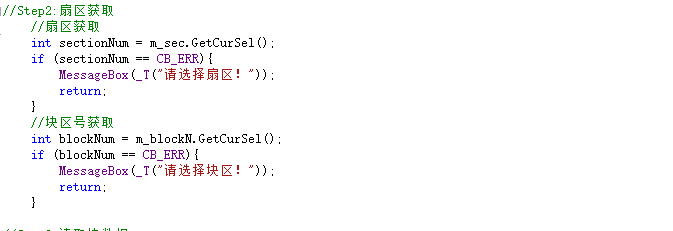
由于是初始化的时候便有选项提供给我们，所以逻辑上需要定义在xxx+dlg.cpp文件中，然后直接调用AddString即可实现，如下图中为对块区组合框的初始化，扇区的为同理，m\_debugger为Debugger类的实体：



1. 读块：读块操作主要需要调用read\_block函数，需要参数扇区号、块区号、密钥类型、密钥、以及两个返回数据。所以首先需要满足扇区选择、块区选择、密钥类型选择、密钥填入操作，若为满足前几样操作将提示并结束当前单机操作。其中需要说明的是密钥类型我们通过在Debugger.h中定义一个公开枚举类型成员RadioStatus,有三种选项noType, TypeA, TypeB,用这三种来确定是否选择密钥。其他判断条件简单并不需要赘述。

这段程序分三步：

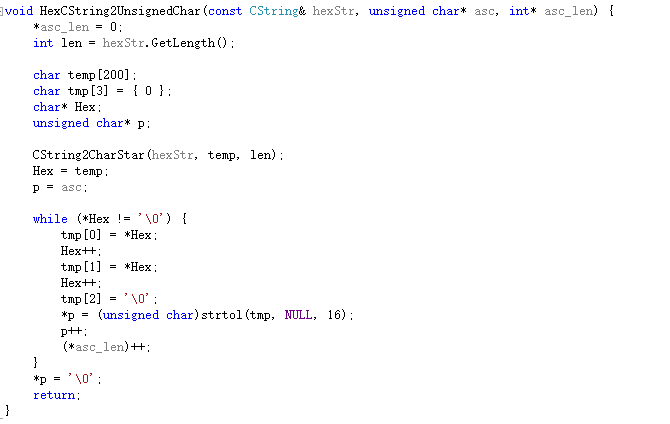






最后需要将获取的块数据从unsigned char类型转换成CString类型，并分别对应到相应的框。

其中第一步中涉及密钥类型的转换是从16进制CString到unsigned char的，其代码见下：



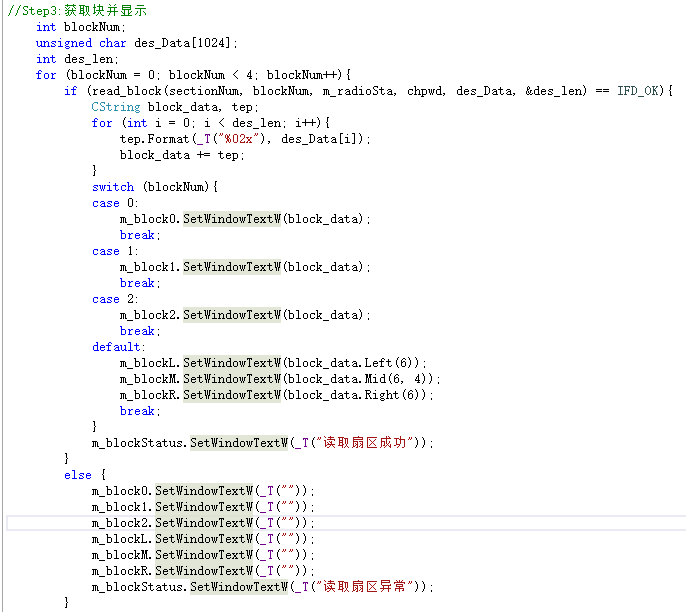
1. 写块：写块部分调用函数write\_block函数，参数同上。这部分的前几个部分与读块的前几个部分是一样的，密钥类型、密钥、选扇区、选块区。即同上分为三个步骤而只有第三个步骤与上不同，逻辑部分在TA给的PPT中比较详细了。由于其中当写块为3时的操作TA并未讲清楚，所以本人选择了参考被人的代码。

TA要求的是块三的三个框都为disable但是这里又可以输入数据，前后莫名矛盾，所以选择将第一个框变为able，后两个变为disable.



最后通过write\_block的返回结果判断是否成功。

3读扇区：读扇区调用read\_block函数，与读块逻辑上是一致的，区别在于读扇区时并不需要选择块区，块区是从0~3循环一遍刚好读取各个块的数据，不同步骤为第三步，见下：



上述中的类型转换就不赘述了，涉及从16进制CString到unsigned char的转换。

## 2.6 电子钱包（白冰）

## 2.7 上机网管

## 2.8 历史记录

# 第三章 RFID读写器上位机软件功能测试

## 3.1 开启设备(陈阳)

。。。。。

## 3.2 获取卡片信息（旦真）

。。。。。

## 3.3 读写扇区（旦增罗布）

## 3.4 电子钱包（白冰）

# 第四章 RFID读写器上位机软件功能创新（旦增罗布）

功能创新部分想了挺久，然后有一次去网咖打游戏，他们的上位机可以做到余额读取并有女性的提示音，受此启发，我们觉得在各个部分加入提示音用户体验会更好。

本次实验中用到三中不同提示音，正确时出发的，结果发生错误时出发的，过程中发生错误时出发三种。

实现：实现是通过Windows提供的Windows Multimedia API接口中的sndPlaySound函数实现，具体过程如下：

1.连接器中导入WinMM.lib库，然后包含mmsystem.h头文件



WinMM.lib库一般位于:

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.1A\Lib\WinMM.Lib

1. 调用bool sndPlaySound(LPCSTR lpszSound,UINT fuSound);

其中lpszSound是所要播放的音乐格式需要.wav；fuSound是播放的标志主要有同步播放，异步播放，循环播放等选择，项目中使用异步播放SND\_ASYNC。如下：



3.目前比较常用PlaySound函数来播放音频文件，由于我们项目中只需要简单的提示音所以sndPlaySound函数就可以已经很好了。

# 第五章 RFID读写器上位机软件项目总结

这次实验得到的知识很多，也获取了许多经验。

首先对字符类型转换能力有了很大的提升，因为再实验过程中涉及这部分的内容很多，每次调用一个函数都需要转换字符类型。CString与unsigned char 类型之间的转换、CString与long类型之间的转换。

其次对开发MFC及Windows的程序起到了了很大的了解及入门的作用，有很多人说MFC是时代淘汰的东西，我觉得即便是现在很少有人在用，但其价值依然存在，MFC加深了我对面向对象及图形化编程的认识。

图形化编程大大地减少了创建类对象及类变量等的操作时间，并且让开发变得很轻松，对多人编程也很友好，自己做的工作可以很清楚地展现出来。

这次实验小组分工很简单，每个人的部分也很简单，但是个人觉得还是需要我们自己从零开始做到自己的部分，因为MFC挺有趣，难度也不是很大，当然最后数据库部分由于小组中没人想做所以就并没有实现。

这次的项目中还有许多需要提升的地方，如果以后有机会择日必定再改进。

本次实验中最初遇到的最大麻烦是类型转换，以及其导致的程序崩坏，在看玩TA提供的类型转换后好好研究发现了其中的规律，最终解决了转换问题。