# 增加读安全模块接口函数供对象管理层调用，支持读esam属性

* 1. 功能要求

通信读取F100 Esam属性，通过对象管理层散转到安全模块。安全模块提供接口函数供对象管理层调用。

* 1. 修改设计方案：
* 增加接口函数如下：

*uchar8 InF\_Get\_SafeModule（ulong32 V\_ulOAD， uchar8 \*pV\_ucDataBuf，ushort16 \*pV\_usDataLen,ushort16 \*P\_usDataEncode）*

*输入参数：*

*V\_ulOAD:读取的OAD；*

*\*pV\_usDataLen：读取数据缓存长度；*

*输出参数：*

*\*pV\_ucDataBuf:读取数据；*

*\*pV\_usDataLen:读取数据实际长度；*

*\*P\_usDataEncode:数据内部编码（第二阶段使用）；*

*返回参数：C\_OK正确，其他返回具体错误（统一定义）*

* 接口函数内部调用安全模块内部读esam属性函数：

*ushort16 S\_ReadEsam\_698(uchar8 \*P\_Data,ushort16 \*P\_ucLen,uchar8 Item,uchar8 V\_ucClientAddr)。该模块功能函数修改如下：*

**注1：esam属性15：**

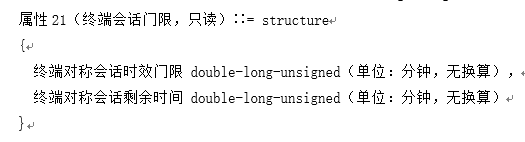
*C:\Users\yblx\AppData\Local\Temp\1631513098(1).png红外认证剩余时间代表2个含义：主站来读代表红外认证剩余时间；终端来读代表终端身份认证时效剩余时间。安全模块通过标识码来区分如下:*

*C:\Users\yblx\AppData\Local\Temp\1631512511(1).png*

*通信模块来读，根据是主站还是终端来读，重新赋值标识码。*

**注2：esam属性6和属性21在安全模块的特殊处理修改：**

**C:\Users\yblx\AppData\Local\Temp\1631522421(1).png**

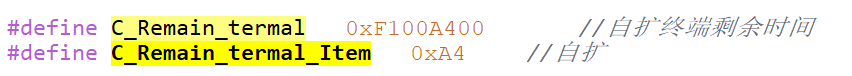
****

安全模块先读出来esam内部的会话剩余时间，再根据通信的建立应用链接剩余时间做逻辑判断：



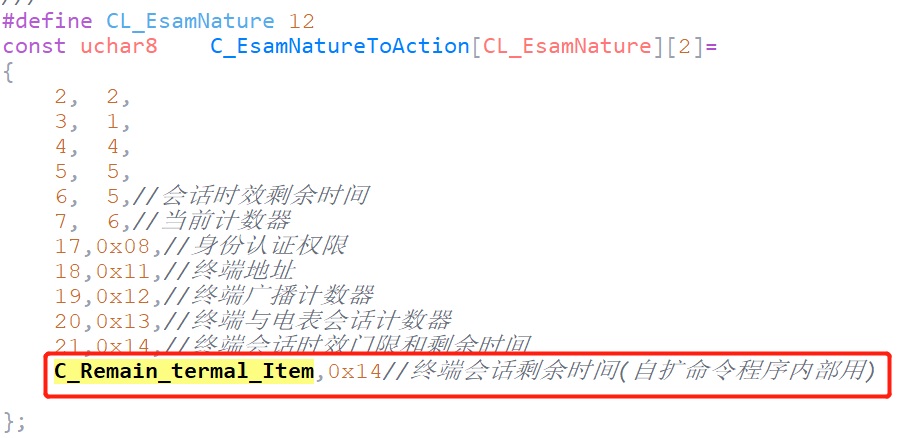
建议该逻辑判断放在通信模块，esam只应答esam内部的应用链接时间。

**注3：之前自扩的读取终端对称会话剩余时间标识码宏定义应该按照698格式规范修改**

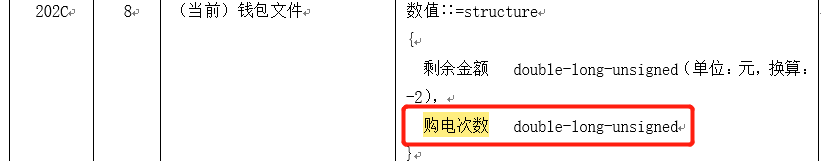
****

**改成 #define C\_Remain\_termal 0xF1001502 /\*自扩终端剩余时间\*/**

**安全模块根据标识码返回数据及长度，不适用自扩特殊处理。**

****

**注4：增加钱包文件的购电次数的读取**

****

**安全模块直接读esam，不存储数据层**

**注：对象管理层/通信负责将钱包文件拆分成读电能模块接口函数和安全模块接口函数？**

* 1. 评审结果

通过

# 安全模块读剩余金额、透支金额改成读对象管理层接口函数

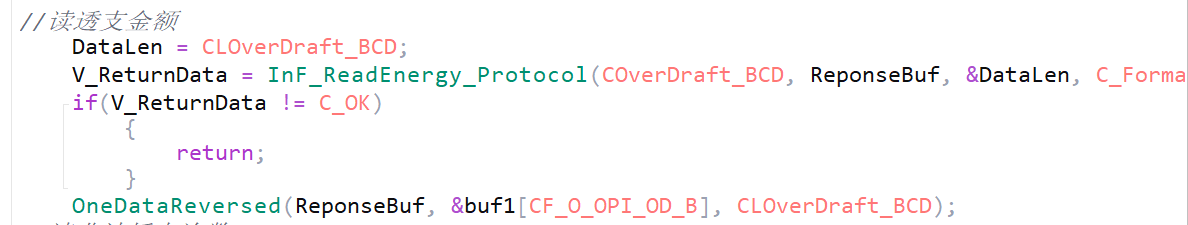
* 1. 功能要求

1：退费前，要读出数据层剩余金额，和esam内部的金额先同步。再判断是否允许退费；

2：充值、退费前，先读出数据层剩余金额，和esam内部的金额先同步；

3：刷新ESAM中运行参数（插入用户卡、~~数据回抄参数运行信息文件（645）~~），读电表的透支金额。

* 1. 修改设计方案：
* void Updata\_ESAM\_OpParm(uchar8 MsgNum)函数中：原程序中会读取透支金额的BCD格式

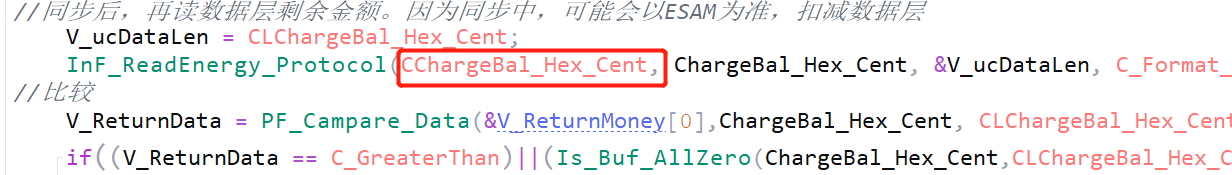


改成读对象管理层接口函数：void InF\_GetData\_ObjectManage（ulong32 V\_ulOAD， uchar8 \*pV\_ucDataBuf，ushort16 \*pV\_usDataLen,ushort16 \*P\_usDataEncode）；



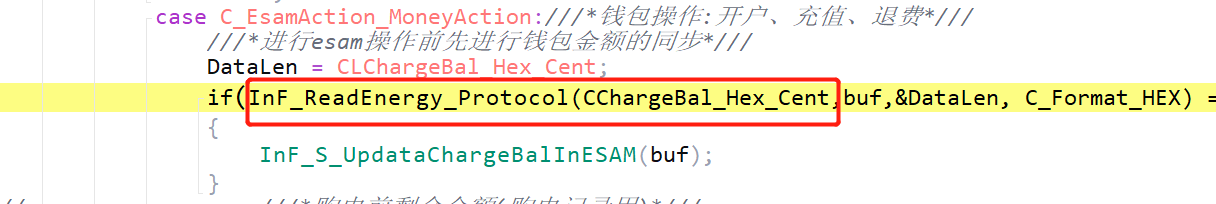
宏定义需要根据电能模块修改做相应调整，电能模块回的透支金额是HEX格式，需要安全模块转换成BCD格式写到esam。

* uchar8 DV\_Return\_Money\_698(uchar8 \*V\_ReturnMoney)函数中读剩余金额：



需要改成读对象管理层接口函数。

* uchar8 INF\_S\_ActionEsam\_698(ulong32 OAD,uchar8 \*p\_Data,ushort16 \*Len,uchar8 Action,uchar8 V\_ucClientAddr)函数中，充值、退费分支会读取数据层剩余金额，与esam剩余金额进行同步：



需要改成读对象管理层接口函数。

* 1. 评审结果

通过

# 合并读取钱包接口函数

* 1. 功能要求

插入用户卡和参数预置卡都需要读取钱包文件：

* 1. 修改设计方案：
* 接口函数合并：。

用户卡读取钱包文件函数接口：

uchar8 InF\_SC\_Get\_MFile\_PreSetCard(StrCardParse \*pV\_CardParse)

{}

参数预置卡中读出钱包文件函数接口：

uchar8 InF\_SC\_Get\_MFile\_UserCard(StrCardParse \*pV\_CardParse)

可以合并为一个接口函数如下：

uchar8 InF\_SC\_Get\_MFile(StrCardParse \*pV\_CardParse)

函数功能：根据入参的结构体StrCardParse里面的卡类型来区分是什么卡，读取相应卡片的钱包文件。

* 1. 评审结果

通过

# 增加判断当前密钥状态接口函数

* 1. 功能要求

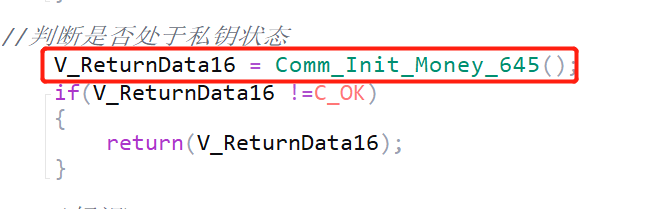
钱包初始化操作，电表需要判断当前电表处于的密钥状态：公钥还是私钥。

* 1. 修改设计方案：
* 增加uchar8 InF\_JudgeKeyStatus(void)接口函数：

直接读取esam中的密钥状态，判断当前处于公钥还是私钥。

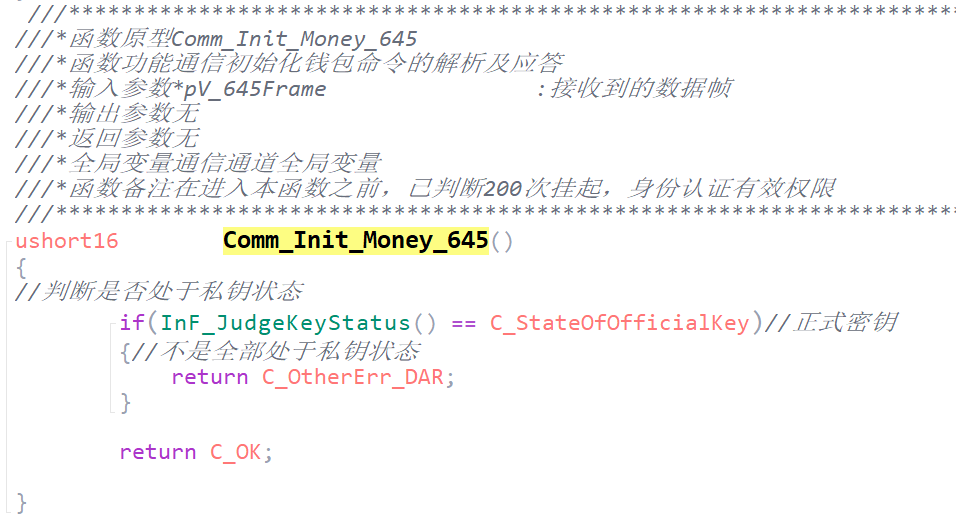
安全模块判断当前密钥状态也调用此接口函数（原通信有此函数）。

* ushort16 S\_InitMoney\_698(uchar8 \*P\_Data,ushort16 \*P\_ucLen)函数中，判断当前电表密钥状态调用了如下函数：



改成调用新增的接口函数，其他模块也调用此接口函数。

* 删除ushort16 Comm\_Init\_Money\_645()此函数：



* 1. 评审结果

通过

# 数据翻转提取公共函数

* 1. 功能要求

数据翻转函数，提取为公共函数。

* 1. 修改设计方案：
* 安全模块中：

void OneDataReversed(uchar8 \*pSBuf, uchar8 \*pDBuf,uchar8 Len)

与通信函数中：

void SF\_OneDataReversed(uchar8 \*pSBuf, uchar8 \*pDBuf,uchar8 Len)

功能一致；提取出来公共函数；

安全模块所有调用OneDataReversed（）的地方都改成调用公共函数

* 1. 评审结果

通过

# Esam操作类接口函数调整---参数更新返回符合698协议的数据

* 1. 功能要求

Esam的操作功能。

* 1. 修改设计方案：
* 修改接口函数如下：

uchar8 INF\_S\_ActionEsam\_698(*ulong32 V\_ulOAD， \*pV\_ucDataBuf，ushort16 \*pV\_usDataLen*, ~~uchar8 Action,~~ *\*pV\_ucDataOutBuf，ushort16 \*pV\_usDataOutLen*,uchar8 V\_ucClientAddr)

*输入参数：*

*V\_ulOAD:操作的OAD；*

*pV\_ucDataBuf：操作输入参数*

*\*pV\_usDataLen：操作数据长度；*

*uchar8 V\_ucClientAddr：主站还是终端*

*输出参数：*

*\* pV\_ucDataOutBuf:返回数据；*

*\* pV\_usDataOutLen:返回数据实际长度；*

*返回参数：C\_OK正确，其他返回具体错误（统一定义）*

注1： 接口函数入参调整，去掉uchar8 Action这个入参；增加输出参数：*\* pV\_ucDataOutBuf：*返回数据指针和*\* pV\_usDataOutLen：*返回数据长度指针。

注2：参数更新返回给通信的698格式的数据

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项（标识码） | Esam格式 | 698格式 | 备注 |
| 表号 | 大端 | 小端 |  |
| 客户编号 | 大端 | 小端 | 没有动态库，上位机不支持 |
| 电流互感器变比 | 3字节BCD格式大端 | 4字节hex格式 小端 |  |
| 电压互感器变比 | 3字节BCD格式大端 | 4字节hex格式 小端 |  |
| 报警金额1、2 | 4字节BCD大端 | 4字节hex小端 |  |
| 两套费率表切换时间 | 5字节BCD  年月日时分、大端 | 7字节  Data\_time\_s格式（小端） |  |
| 两套阶梯表切换时间 | 5字节BCD  年月日时分、大端 | 7字节  Data\_time\_s格式（小端） |  |
| 当前费率表 | --- | --- | 没有动态库，上位机不支持 |
| 备用套费率表 | 4字节BCD | 4字节hex | 补齐12个费率 |
| 当前阶梯表 | --- | --- | 没有动态库，上位机不支持 |
| 备用阶梯表 | 4字节BCD+4字节BCD+月日时BCD | 4字节hex+4字节hex+月日时HEX（顺序月、日、时） | 数据长度不够，动态库不能生成数据 |

* 1. 评审结果

通过

# 密钥状态不存EE

* 1. 功能要求

密钥状态直接读esam，EE中不保存。

数据层不存储密钥状态

C:\Users\yblx\AppData\Local\Temp\1631695270(1).png

* 1. 修改设计方案：
* 删除void InF\_S\_GetKeyStatus\_ReworkKeyStatus(void)函数：。

函数功能：同步EE和ESAM中密钥状态。

* 1. 评审结果

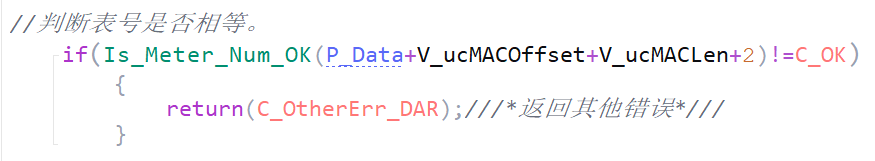
通过

# 判断表号是否相等函数

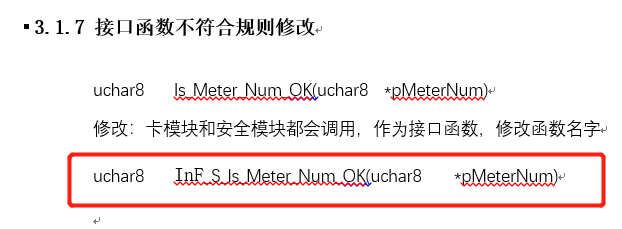
* 1. 功能要求

开户充值退费，需要判断上位机下发的表号和电表数据层的表号是否一致。

* 1. 修改设计方案：
* ushort16 S\_Purchase\_ReturnFile\_698(uchar8 \*P\_Data,ushort16 \*P\_ucLen)函数中调用判断表号是否相等函数：



此接口在安全模块中做了修改如下：



故调用做了修改。

* 1. 评审结果

通过

# esam类属性14红外认证时效门限（身份认证时效门限）、属性15红外认证剩余时间（终端身份认证剩余时间）数据层存储4字节

* 1. 功能要求

身份认证时效门限采用原红外认证时效门限且只能通过主站进行设置

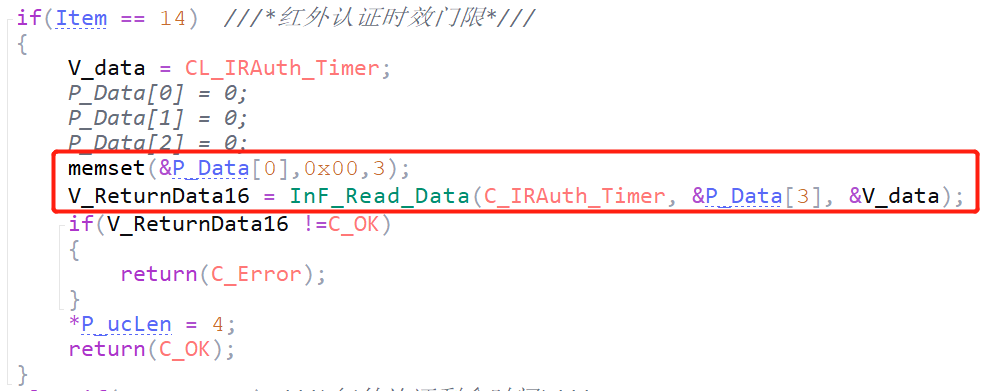
客户机地址为主站时，抄读红外认证剩余时间为原红外认证剩余时间T1。客户机地址为终端时，抄读红外认证剩余时间为当前终端维持的身份认证剩余时间T2。

* 1. 修改设计方案：

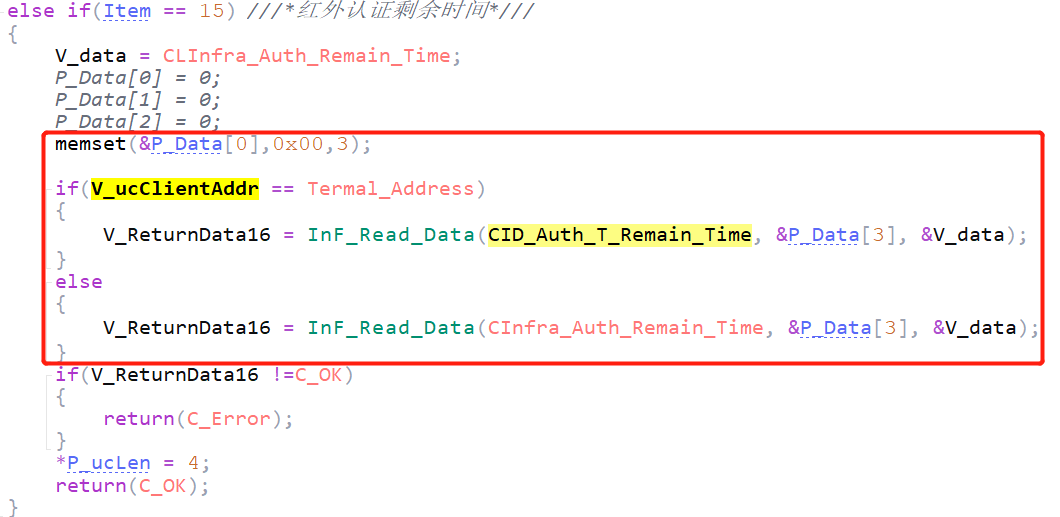
**安全模块：**

* ushort16 S\_ReadEsam\_698(uchar8 \*P\_Data,ushort16 \*P\_ucLen,uchar8 Item,uchar8 V\_ucClientAddr)函数中：

**注1 读红外时效门限，直接读取数据层数据，不进行数据处理**



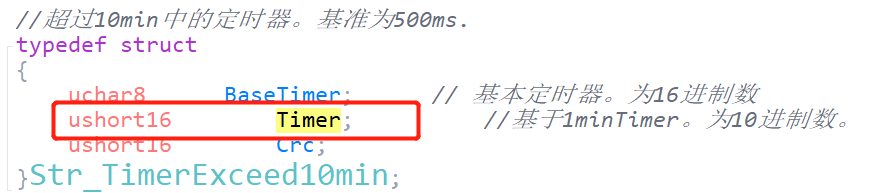
**注2 读红外剩余时间，直接读取数据层数据，不进行数据处理**。



* uchar8 INF\_S\_ActionEsam\_698(ulong32 OAD,uchar8 \*p\_Data,ushort16 \*Len,uchar8 Action,uchar8 V\_ucClientAddr)函数中，红外认证成功处理通信定时器，移到通信模块中：



**通信模块修改**



void DealWith\_ID\_Auth\_T\_Timer(uchar8 Method,uchar8 \*pVTimer)

void DealWith\_IR\_Auth\_Timer(uchar8 Method,uchar8 \*pVTimer)

函数和结构体都需要调整，定时器是4字节。

* 1. 评审结果

通过

# 数据颠倒接口函数合并

* 1. 功能要求

数据颠倒。

* 1. 修改设计方案：
* 安全模块中：

函数中：调用void OneDataReversed(uchar8 \*pSBuf, uchar8 \*pDBuf,uchar8 Len)数据颠倒的地方，统一调用公共函数PF\_ OneDataReversed(uchar8 \*pSBuf, uchar8 \*pDBuf,uchar8 Len)

* 1. 评审结果

通过

# 功能

* 1. 功能要求

插。

* 1. 修改设计方案：
* Card\_Parse（）函数中：。

uchar8 INF\_JudgeVolage\_Relay(void)

* 1. 评审结果

通过

# 功能

* 1. 功能要求

插。

* 1. 修改设计方案：
* Card\_Parse（）函数中：。

uchar8 INF\_JudgeVolage\_Relay(void)

* 1. 评审结果

通过