API说明

所有函数接口的参数和返回值都用json字符串表示

1 初始化读写器

/\*

BSTR InitReader(BSTR INPARAMS);

\*/

INPARAMS:格式为 {"PhyUrl":"串口号或ip地址","AntPortNumber":"天线端口数量"}；举例：{"PhyUrl":"com3","AntPortNumber":"4"}

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

2 参数设置

\*/

BSTR ParamSet(BSTR INPARAMS);

/\*

INPARAMS:主体格式为：{"ParamType":"参数类型","ParamValue":"参数值"}；根据不同的参数具体格式会不同，目前支持的参数有：

2.1，设置gen2session，具体格式为{"ParamType":"Potl/Gen2/Session","ParamValue":"session值"},session的合法值为0-3。

例如：{"ParamType":"Potl/Gen2/Session","ParamValue":"0"}

2.2，设置功率，具体格式为{"ParamType":"Rf/Antpower","ParamValue":[{"AntId":"天线id","ReadPower":"读功率值","WritePower":"写功率值"},{},...]}；

天线id从1开始编号，读功率值和写功率值对于发卡机的合理范围为500-2300。当读写器有多个天线端口的时候，ParamValue可能含有多个天线的设置。

例如：

{"ParamType":"Rf/Antpower","ParamValue":[{"AntId":"1","ReadPower":"2000","WritePower":"2000"},{"AntId":"2","ReadPower":"3000","WritePower":"2000"},{"AntId":"3","ReadPower":"1000","WritePower":"1000"},{"AntId":"4","ReadPower":"3000","WritePower":"3000"}]}

2.3，设置过滤器，具体格式为{"ParamType":"Tag/Filter","ParamValue":{"Bank":"bank值","StartAddress":"起始地址","FilterData":"过滤匹配的数据","IsInvert":"是否匹配"}}

bank值范围为1-3；起始地址从0开始，以bit为单位；过滤匹配的数据为2进制表示的字符串，例如00001111这样的字符串。是否匹配字段为false的时候表示匹配，为true表示不匹配。

例如：

{"ParamType":"Tag/Filter","ParamValue":{"Bank":"1","StartAddress":"32","FilterData":"0001000100010001","IsInvert":"false"}}

2.4, 设置gen2q，具体格式为{"ParamType":"Potl/Gen2/Q","ParamValue":"q值"},q的合法值为-1到15，-1表示自动。

例如：{"ParamType":"Potl/Gen2/Q","ParamValue":"-1"}

2.5, 设置tagencoding，具体格式为{"ParamType":"Potl/Gen2/Tagencoding","ParamValue":

"tagencoding值"}, tagencoding的合法值为0到3 0:fm0,1:m2,2:m4,3:m8

例如：

{"ParamType":"Potl/Gen2/Tagencoding","ParamValue":"3"}

2.6，设置maxepclen，具体格式为

{"ParamType":"Potl/Gen2/Maxepclen","ParamValue":"Maxepclen值"},maxepclen的值96,496

例如：{"ParamType":"Potl/Gen2/Maxepclen","ParamValue":"496"}

2.7，设置嵌入读数据，具体格式为

{"ParamType":"Tag/Embededdata","ParamValue":{"Bank":"bank值","StartAddress":"起始地址","ByteCnt":"数据长度","Accesspwd":"访问密码"}}

bank值范围为1-3；起始地址从0开始，bytecnt指读多少字节，Accesspwd是访问密码

例如：

{"ParamType":"Tag/Embededdata","ParamValue":{"Bank":"1","StartAddress":"2","ByteCnt":"6","Accesspwd":"11111111"}}

2.8,设置invpotl 具体格式为{"ParamType":"Tag/Invpotl","ParamValue":{"Protocol":"协议值","Potlcnt":"个数"}},协议值 NONE，30；ISO180006B，30；GEN2，30，ISO180006B\_UCODE,20； IPX64,30； IPX256，40；

例如：

{"ParamType":"Tag/Invpotl","ParamValue":{"Protocol":"GEN2,30;ISO180006B,30","Potlcnt":"2"}}

2.9,设置Reader/Is/Chk/Ant 具体格式为{"ParamType":"Reader/Is/Chk/Ant",

"ParamValue":"Reader/Is/Chk/Ant值"},检测天线值 Is\_chk\_ant的值 true 或者flase

例如：

{"ParamType":"Reader/Is/Chk/Ant","ParamValue":"true"}

2.10，设置reader\_ip 具体格式为{"ParamType":"Reader/IP","ParamValue":[{"Ip":"Ip地址","Mask":"mask地址","Gateway":"网关"}]}；

例如：

{"ParamType":"Reader/IP","ParamValue":{"Ip":"192.168.1.100","Mask":"255.255.255.0","Gateway":"192.168.1.1"}}

2.11，设置frequency\_region 具体格式为

{"ParamType":"Frequency/Region","ParamValue":"region值"},region

的NONE,NA,EU,EU2,EU3, KR,PRC,PRC2,OPEN.

例如：

{"ParamType":"Frequency/Region","ParamValue":"NA"}

2.12,设置frequency\_hoptable 具体格式为

{"ParamType":"Frequency/Hoptable","ParamValue":"Hoptable频点表"}； hoptable 是

902250,902750，......;

例如：

{"ParamType":"Frequency/Hoptable","ParamValue":"925750,920250,925250,925750"}

2.13,设置gen2\_blf 具体格式为{"ParamType":"Potl/Gen2/Blf","ParamValue":"gen2\_blf值"},

例如：

{"ParamType":"Potl/Gen2/Blf","ParamValue":"40"}

2.14,设置gen2\_writemode 具体格式为

{"ParamType":"Potl/Gen2/Writemode","ParamValue":"writemode值"}, writemode 值0为word write 1为block write 2 为先block write后word write。

例如：

{"ParamType":"Potl/Gen2/Writemode","ParamValue":"0"}

2.15,设置gen2\_target 具体格式为{"ParamType":"Potl/Gen2/Target","ParamValue":"target值"}, target值0表示A，1表示B，2表示A->B,3表示B->A

例如：

{"ParamType":"Potl/Gen2/Target","ParamValue":"0"}

2.16,设置uniquebyant 具体格式为

{"ParamType":"Tagdata/Uniquebyant","ParamValue":"uniqueby值"}, true表示天线唯一，false表示天线不唯一

例如：

{"ParamType":"Tagdata/Uniquebyant","ParamValue":"true"}

2.17,设置uniquebyemddata 具体格式为{"ParamType":"Tagdata/Uniquebyemddata

","ParamValue":"uniquebyemddata值"}, true表示嵌入数据唯一，false表示嵌入数据不唯一

例如：

{"ParamType":"Tagdata/Uniquebyemddata ","ParamValue":"true"}

2.18,设置tagdata\_recordhighestrssi 具体格式为

{"ParamType":"Tagdata/Recordhighestrssi","ParamValue":"recordhighestrssi值"}, true表示记录最高rssi，false表示不记录

例如：

{"ParamType":"Tagdata/Recordhighestrssi","ParamValue":"true"}

2.19，设置hoptime,具体格式为{"ParamType":"RF/Hoptime","ParamValue":"hoptime值"},

例如：

{"ParamType":"RF/Hoptime","ParamValue":"0"}

2.20,设置 lbt\_enable 具体格式为{"ParamType":"RF/Lbt/Enable","ParamValue":"enable"值}, true表示启用，false表示禁用

例如：

{"ParamType":"RF/Lbt/Enable","ParamValue":"true"}

2.21，设置Iso180006b\_blf 具体格式为

{"ParamType":"Polt/ISO180006B/Blf","ParamValue":"1so180006b\_blf"值}, true表示启用，false表示禁用

例如：

{"ParamType":"Polt/ISO180006B/Blf","ParamValue":"true"}

2.22,设置gen2\_tari ,具体格式为{"ParamType":"Potl/Gen2/Tari","ParamValue":"gen2\_tari值"}, 0:Tari of 25 microseconds;1:Tari of 12.5 microseconds;2:Tari of 6.25 microseconds

例如：

{"ParamType":"Potl/Gen2/Tari","ParamValue":"0"}

2.23,设置trans\_timeout 具体格式为{"ParamType":"Trans/Timeout","ParamValue":"hoptime值"},

例如：

{"ParamType":"Trans/Timeout","ParamValue":"200"}

2.24,设置tag\_embsecureread 具体格式为

{"ParamType":"Tag/Embsecureread","ParamValue":{"Tagtype":"tagtype值","Pwdtype":"pwdtype值","ApIndexStartBitsInEpc":"apindexstartbitsinepc值

","ApIndexBitsNumInEpc":"apindexbitsnumbinepc值","Bank":"bank值","Address":"起始地址","Blkcnt":"数据长度","Accesspwd":"访问密码"}}

例如：

{"ParamType":"Tag/Embsecureread","ParamValue":

{"Tagtype":"0","Pwdtype":"0","ApIndexStartBitsInEpc":"2","ApIndexBitsNumInEpc":"6","Bank":"1","Address":"0","Blkcnt":"6","Accesspwd":"11111111"}}

2.25,设置transmit\_mode 具体格式为

{"ParamType":"Transmit/Mode","ParamValue":"transmit\_mode值"},

例如：

{"ParamType":"Transmit/Mode","ParamValue":"0"}

2.26,设置powersave\_mode 具体格式为

{"ParamType":"Powersave/Mode","ParamValue":"powersave\_mode值"},

例如：

{"ParamType":"Powersave/Mode","ParamValue":"0"}

2.27，设置tag\_search\_mode 具体格式为

{"ParamType":"Tag/Search/Mode","ParamValue":"tag\_search\_mode值"},

例如：{"ParamType":"Tag/Search/Mode","ParamValue":"0"}

2.28，设置polt\_iso180006b\_modulation\_depth 具体格式为

{"ParamType":"Polt/ISO180006B/Modulation/Depth","ParamValue":"depth值"},

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

例如：

{"ParamType":"Polt/ISO180006B/Modulation/Depth","ParamValue":"0"}

3 盘点函数

BSTR TagInventory(BSTR INPARAMS);

INPARAMS:格式为：{"UseAnts":"使用的天线","TimeOut":"超时时间"}；使用的天线用类似1234这样的字符串表示，1234表示同时使用天线1，2，3，4；超时时间的有效范围为50-65535，

单位毫秒，一般采用一个天线给150ms的方式

返回值：格式为：{"ErrCode":"错误代码","TagCount":"标签数量","TagItems":[{"EpcId":"epc码","EmbededData":"附加数据","AntennaID":"天线id","ReadCnt":"读次数","Frequency":"频点","protocol":"协议","RSSI":"信号强度"},{},...]}

错误代码为0表示成功，其他值为出错；标签数量表示读到的标签个数；epc码为16进制表示的epc数据；附加数据为选择在盘存中嵌入读指令时读到的其他bank数据，一般不使用这种方式时，此字段为null；天线id表示从那个天线读到标签；读次数

表示标签在此次盘存过程中被读到的总次数；频点表示标签是从哪个频点读到的；协议表示标签遵从的空中接口协议；信号强度表示读写器接收到标签信号的实际强度。

例如：

{"UseAnts":"1","TimeOut":"150"}

4 初始化标签函数

BSTR WriteTagEpc(BSTR INPARAMS);

INPARAMS:格式为：{"OpAnt":"操作天线","EpcId":"要写入的epc码","AccessPassword":"访问密码","TimeOut":"超时时间"}

操作天线表示要使用哪个天线来写；要写入的epc码为需要写入的epc码，字符串的长度必须是4的倍数；访问密码固定为8个字符，如果不需要使用访问密码则设置为00000000；超时时间的有效范围为50-65535，一般设置为1000

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

例如：

write tag

{"OpAnt":"1","EpcId":"1111222233334444555566661111222233334444555566661111222233334444555566661111222233334444555566661111222233334444555566664567","AccessPassword":"00000000","TimeOut":"1000"}

5 获取单标签数据

BSTR GetTagData(BSTR INPARAMS);

INPARAMS:格式为：{"OpAnt":"操作天线","Bank":"bank值","StartAddress":"起始地址","BlockCount":"要读的块数","AccessPassword":"访问密码","TimeOut":"超时时间"}

操作天线表示要使用哪个天线来写；bank值有效范围为0-3；起始地址从0编址，以块（16bit）为单位；要读的块数为实际需要读的块数；访问密码固定为8个字符，如果不需要使用访问密码则设置为00000000；

超时时间的有效范围为50-65535，一般设置为1000

返回值：{"ErrCode":"错误代码"，"BankData":"读到的数据"}；错误代码0表示执行成功，其他值表示出错；读到的数据是用16进制字符串表示的数据，当读出错的时候，此字段为null

例如：

{"OpAnt":"1","Bank":"1","StartAddress":"2","BlockCount":"6","AccessPassword":"00000000","TimeOut":"1000"}

6 写单标签数据

BSTR WriteTagData(BSTR INPARAMS);

INPARAMS:格式为：{"OpAnt":"操作天线","Bank":"bank值","StartAddress":"起始地址","WriteData":"要写入的数据","AccessPassword":"访问密码","TimeOut":"超时时间"}

操作天线表示要使用哪个天线来写；bank值有效范围为0-3；起始地址从0编址，以块（16bit）为单位；要写入的数据为16进制字符串表示的数据，长度必须是4的倍数；访问密码固定为8个字符，如果不需要使用访问密码则设置为00000000；

超时时间的有效范围为50-65535，一般设置为1000

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

INPARAMS：作为保留参数使用，现在可以随便传入一个字符串。

返回值：{"ErrCode":"错误代码"}； 。

{"OpAnt":"1","Bank":"1","StartAddress":"2","WriteData":"00009999","AccessPassword":"00000000","TimeOut":"1000"}

7 关闭读写器

BSTR CloseReader(BSTR INPARAMS);

例如：CloseReader(“”);

8 锁标签

BSTR LockTag(BSTR INPARAMS);

INPARAMS:格式为：{"OpAnt":"操作天线","LockObjects":"lockobjects值","LockTypes":"locktypes值","AccessPassword":"访问密码","TimeOut":"超时时间"}

操作天线表示要使用哪个天线来写；锁定的区域，要锁定多个区域使用字符串LOCK\_OBJECT\_KILL\_PASSWORD|LOCK\_OBJECT\_ACCESS\_PASSWD|LOCK\_OBJECT\_BANK1|LOCK\_OBJECT\_BANK2|LOCK\_OBJECT\_BANK3；锁定类型，可以使用|进行多个锁定操作，锁定类型有如下：

KILL\_PASSWORD\_UNLOCK,

KILL\_PASSWORD\_LOCK, //销毁密码密码锁定

KILL\_PASSWORD\_PERM\_LOCK//销毁密码永久锁定

ACCESS\_PASSWD\_UNLOCK,

ACCESS\_PASSWD\_LOCK//访问密码密码锁定

ACCESS\_PASSWD\_PERM\_LOCK//访问密码永久锁定

BANK1\_UNLOCK,

BANK1\_LOCK//bank1密码锁定

BANK1\_PERM\_LOCK,//bank1永久锁定

BANK2\_UNLOCK,

BANK2\_LOCK, //bank2密码锁定

BANK2\_PERM\_LOCK,//bank2永久锁定

BANK3\_UNLOCK,

BANK3\_LOCK,//bank3密码锁定

BANK3\_PERM\_LOCK, //bank3永久锁定

访问密码固定为8个字符，如果不需要使用访问密码则设置为00000000；

超时时间的有效范围为50-65535，一般设置为1000

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

例如：

{"OpAnt":"1","LockObjects":"LOCK\_OBJECT\_BANK1|LOCK\_OBJECT\_BANK3","LockTypes":"BANK1\_UNLOCK|ACCESS\_PASSWD\_LOCK","AccessPassword":"11111111","TimeOut":"1000"}

9 销毁标签

BSTR KillTag(BSTR INPARAMS);

INPARAMS:格式为：{"OpAnt":"操作天线","AccessPassword":"访问密码","TimeOut":"超时时间"}

操作天线表示要使用哪个天线来写；

访问密码固定为8个字符，如果不需要使用访问密码则设置为00000000；

超时时间的有效范围为50-65535，一般设置为1000

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

例如：{"OpAnt":"1","AccessPassword":"11111111","TimeOut":"1000"}

10 获取参数

BSTR ParamGet(BSTR INPARAMS);

返回值：{"ErrCode":"错误代码",return返回值}；错误代码0表示执行成功，其他值表示出错。

return返回值 主体格式为：{"ParamType":"参数类型","ParamValue":"参数值"}；根据不同的参数具体格式会不同

例如：具体格式为{"ParamType":"Rf/Antports/Vswr","ParamValue":["AntCnt":"antcount值

","AntVSWRS":[{"Antdid":"antid","Vswr":"vswr"},{}...]}；

INPARAMS:格式为：{"ParamType":"参数名称"}

参数名称：

{"ParamType":"Potl/Gen2/Session"}

{"ParamType":"Potl/Gen2/Tagencoding"}

{"ParamType":"Potl/Gen2/Maxepclen"}

{"ParamType":"Rf/Antpower"}

{"ParamType":"Rf/Maxpower"}

{"ParamType":"Rf/Minpower"}

{"ParamType":"Tag/Filter"}

{"ParamType":"Tag/Embededdata"}

{"ParamType":"Tag/Invpotl"}

{"ParamType":"Reader/Conn/Ants"}

{"ParamType":"Reader/Available/Antports"}

{"ParamType":"Reader/Is/Chk/Ant"}

{"ParamType":"Reader/Version"}

{"ParamType":"Reader/Ip"}

{"ParamType":"Frequency/Region"}

{"ParamType":"Frequency/Hoptable"}

{"ParamType":"Potl/Gen2/Blf"}

{"ParamType":"Potl/Gen2/Writemode"}

{"ParamType":"Potl/Gen2/Target"}

{"ParamType":"Tagdata/Uniquebyant"}

{"ParamType":"Tagdata/Uniquebyemddata"}

{"ParamType":"Tagdata/Recordhighestrssi"}

{"ParamType":"Rf/Temperature"}

{"ParamType":"Rf/Hoptime"}

{"ParamType":"Rf/Lbt/Eanble"}

{"ParamType":"Rf/Supportedregions"}

{"ParamType":"Potl/Supportedprotocols"}

{"ParamType":"Potl/ISO180006B/Blf"}

{"ParamType":"Potl/Gen2/Tari"}

{"ParamType":"Trans/Timeout"}

{"ParamType":"Tag/Embsecureread"}

{"ParamType":"Transmit/Mode"}

{"ParamType":"Powersave/Mode"}

{"ParamType":"Tag/Search/Mode"}

{"ParamType":"Potl/ISO180006B/Modulation/Depth"}

{"ParamType":"Potl/ISO180006B/Delimiter"}

{"ParamType":"Rf/Antports/Vswr"}

11 设置gpo状态

BSTR SetGPO(BSTR INPARAMS)

INPARAMS:格式为：{"GpoId":"Gpo编号（从1开始）","GpoState ":"Gpo状态（0或者1）"}

例如：{" GpoId ":"1"," GpoState ":"1"}

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

12 获取gpi状态

BSTR GetGPI(BSTR INPARAMS)

INPARAMS:格式为：{"GpiId":"Gpi编号（从1开始）"}

返回值：{"ErrCode":"错误代码", “GpiState”:”Gpi状态（0或者1）”}；错误代码0表示执行成功，其他值表示出错，如果出错则没有GpiState字段

13 开始异步清点标签

BSTR StartInventory(BSTR INPARAMS)

INPARAMS:格式为：{"UseAnts":"使用的天线"," ReadDur":"读时长",” ReadInterval”:”读周期”}使用的天线用类似1234这样的字符串表示，1234表示同时使用天线1，2，3，4；读时长的有效范围为50-65535，单位为毫秒；读周期取值一般为：ReadDur+600，单位为毫秒，一般采用一个天线给100ms的方式。在开始异步清点后，读写器以ReadInterval为周期，每次清点标签的时间长度为ReadDur。当读到标签后会以事件的形式通知调用方，事件的回调函数形式为：void FindTag(BSTR tag), 参数tag为读到的单个标签信息，每读到一个标签都会产生一个通知事件，tag格式为：{"EpcId":"epc码","EmbededData":"附加数据","AntennaID":"天线id","ReadCnt":"读次数","Frequency":"频点","protocol":"协议","RSSI":"信号强度"}。在异步清点操作期间其它函数都不可以调用。

返回值：{"ErrCode":"错误代码"}；错误代码0表示执行成功，其他值表示出错

14 停止异步清点标签

BSTR StopInventory(BSTR INPARAMS);

INPARAMS:保留给未来使用，传入空字符串“”即可

在调用了StartInventory方法后在必须要调用StopInventory方法才能释放相关资源。在调用StopInventory后可以再调用其它函数。

15部署说明

请将lib目录拷贝到选定的安装目录，winxp系统双击“register.bat”完成控件注册，windonws7和windows8要以管理员选项运行注册程序。用ie打开tesModuleAPI2.html测试网页