

***UHFREADER288MP.DLL 动态连接库使用***  
**手册**

**V2.0**

# 目录

目录.....	2
1.介绍.....	4
2.应用程序集成.....	5
3.编程接口.....	6
3.1 连接/关闭读写器.....	6
3.1.1 AutoOpenComPort 【自动连接串口】 .....	6
3.1.2 OpenComPort 【连接指定串口】 .....	7
3.1.3 CloseComPort 【关闭所有串口】 .....	8
3.1.4 CloseSpecComPort 【关闭指定串口】 .....	9
3.1.5 OpenNetPort 【打开 TCP 连接】 .....	10
3.1.6 CloseNetPort 【关闭 TCP 连接】 .....	11
3.2. 18000-6C 标准命令.....	12
3.2.1. Inventory_G2 【询查命令】 .....	12
3.2.2. InventoryMix_G2 【混合询查命令】 .....	14
3.2.3. InitRFIDCallBack 【初始化回调】 .....	17
3.2.4. ReadData_G2 【读数据命令】 .....	18
3.2.5. WriteData_G2 【写数据命令】 .....	19
3.2.6. BlockWrite_G2 【块写命令】 .....	20
3.2.7. ExtReadData_G2 【扩展读数据命令】 .....	21
3.2.8. ExtWriteData_G2 【扩展写数据命令】 .....	22
3.2.9. WriteEPC_G2 【广播写 EPC 号命令】 .....	23
3.2.10. BlockErase_G2 【块擦除命令】 .....	24
3.2.11. Lock_G2 【设定存储区读写保护状态命令】 .....	25
3.2.12. KillTag_G2 【销毁标签命令】 .....	27
3.2.13. SingleTagInventory_G2 【单张标签查询命令】 .....	28
3.3. 缓存操作命令.....	29
3.3.1. InventoryBuffer_G2 【缓存询查命令】 .....	29
3.3.2. ReadBuffer_G2 【获取缓存数据命令】 .....	31
3.3.3. ClearBuffer_G2 【清空缓存数据命令】 .....	32
3.3.4. GetBufferCnt_G2 【获取缓存标签数量命令】 .....	33
3.3.5. SetSaveLen 【设置缓存的 EPC/TID 长度命令】 .....	34
3.3.5. GetSaveLen 【获取缓存的 EPC/TID 长度命令】 .....	35
3.4. NXP 标签命令.....	36
3.4.1. SetPrivacyByEPC_G2 【单张读保护设置命令】 .....	36
3.4.2. SetPrivacyWithoutEPC_G2 【多张读保护设置命令】 .....	37
3.4.3. ResetPrivacy_G2 【解锁读保护命令】 .....	38
3.4.4. CheckPrivacy_G2 【检测读保护命令】 .....	39
3.4.5. EASConfigure_G2 【EAS 报警配置命令】 .....	40

3.4.6. EASAlarm_G2 【报警探测命令】 .....	41
3.5. Impinj Monza4 QT 标签命令 .....	42
3.5.1. GetMonza4QTWorkParamter_G2 【读取 Monza4QT 工作参数命令】 .....	42
3.5.2. SetMonza4QTWorkParamter_G2 【设置 Monza4QT 工作参数命令】 .....	43
3.5.3. Inventory_QT_G2 【QT 询查命令】 .....	43
3.5.4. SetTagCustomFunction_G2 【标签自定义命令】 .....	46
3.6. 18000-6B 标准命令 .....	47
3.6.1. InventorySingle_6B 【单张询查命令】 .....	47
3.6.2. InventoryMultiple_6B 【条件询查命令】 .....	48
3.6.3. ReadData_6B 【读数据命令】 .....	49
3.6.4. WriteData_6B 【写数据命令】 .....	50
3.6.5. CheckLock_6B 【锁定检测命令】 .....	51
3.6.6. Lock_6B 【锁定命令】 .....	52
3.7. 读写器自定义命令 .....	53
3.7.1. GetReaderInformation 【获取读写器信息命令】 .....	53
3.7.2 SetAddress 【设置读写器地址命令】 .....	55
3.7.3 SetInventoryScanTime 【询查命令最大响应时间设置命令】 .....	56
3.7.4. SetRfPower 【功率设置命令】 .....	57
3.7.5. SetRegion 【工作频率设置命令】 .....	58
3.7.6. SetBaudRate 【波特率设置命令】 .....	59
3.7.7. BuzzerAndLEDControl 【声光控制命令】 .....	60
3.7.8. SetAntennaMultiplexing 【天线配置命令】 .....	61
3.7.9. SetBeepNotification 【蜂鸣器设置命令】 .....	62
3.7.10. SetRelay 【继电器控制命令】 .....	63
3.7.11. GetGPIOStatus 【读取 GPIO 状态命令】 .....	64
3.7.12. SetGPIO 【GPIO 控制命令】 .....	65
3.7.13. GetSeriaNo 【获取读写器序列号命令】 .....	66
3.7.14. SetCheckAnt 【天线检测命令】 .....	67
3.7.15. WriteRfPower 【设置写功率命令】 .....	68
3.7.16. ReadRfPower 【读取写功率命令】 .....	69
3.7.17. RetryTimes 【写重复次数设置命令】 .....	70
3.7.18. SetReadMode 【实时模式设置命令】 .....	71
3.7.19. SetReadParameter 【设置实时模式工作参数命令】 .....	72
3.7.20. GetReadParameter 【读取实时模式工作参数命令】 .....	74
3.7.21. SetDRM 【设置 DRM】 .....	75
3.7.22. GetDRM 【读取 DRM】 .....	76
3.7.23. GetReaderTemperature 【读取温度】 .....	77
3.7.24. MeasureReturnLoss 【测量回损】 .....	78
3.7.25. SetNotificationPulseOutput 【通知输出端口命令】 .....	79
3.7.26. ReadActiveModeData 【获取主动上传数据】 .....	80
附录 1. 返回值错误代码表 .....	81
附录 2. ErrorCode 错误代码表 .....	82



## 1.介绍

该开发包支持 C, C++ 和其他能够调用 C 函数接口的语言。如 delphi , C#, VB6.0,VB.NET 等。该开发包目前只支持 windows 操作系统（分 32 位和 64 位）。

上位机应用程序通过 UHFREADER288MP.DLL 操作 R2000 系列 18000-6C、18000-6B 格式电子标签读写器，支持上位机应用程序同时在多个通讯端口打开多个读写器。

R2000 系 列 电 子 标 签 读 写 器 包 括 UHFREADER288MP、UHFREADER288M、UHFReader82 等型号。

## 2.应用程序集成

该开发包由以下个文件组成：

文件	使用
UHFREADER288MP.dll	DLL 动态链接库，包含所有 API 接口
UHFREADER288MP.lib	VC++工程 静态链接库
UHFREADER288MP.h	C/C++ 工程头文件，函数定义
Reader.h	C/C++ 附加头文件
ThreadClass.h	C/C++ 附加头文件
RWDev.cs	C# 函数定义类文件
UHFREADER288MP_DLL_Head.pas	Delphi 函数定义头文件

编程语言	说明
VC++	UHFREADER288MP.lib, UHFREADER288MP.h, Reader.h, ThreadClass.h 拷贝到工程目录，并加入到工程。把 UHFREADER288MP.dll 拷贝到 exe 文件输出目录。
C#	RWDev.cs 拷贝到工程目录，并加入到工程。把 UHFREADER288MP.dll 拷贝到 exe 文件输出目录。
Delphi	UHFREADER288MP_DLL_Head.pas 拷贝到工程目录，并加入到工程。把 UHFREADER288MP.dll 拷贝到 exe 文件输出目录。

## 3.编程接口

### 3.1 连接/关闭读写器

#### 3.1.1 AutoOpenComPort【自动连接串口】

定义	int AutoOpenComPort(int* port, BYTE* ComAdr, BYTE baud, int* FrmHandle);		
说明	自动识别并打开与读写器连接的串口，该功能只支持 com1-com9 之间的串口。		
参数	名称	类型	备注
	port	Int*	返回与读写器连接的串口号。
	ComAdr	BYTE*	输入读写器地址，如果在不清楚真实地址的情况下，可以输入广播地址 0xFF，函数调用成功后，此参数将返回真实地址。
	baud	BYTE	串口通讯波特率。0-9600；1-19200；2-38400；5-57600；6-115200。
	FrmHandle	Int*	返回与读写器连接的句柄，此后所有 API 操作都需要此句柄。
返回 (int)	成功：0；失败：非 0；（查看返回值错误代码表）		
参考代码	<pre>int port=0; BYTE address=255;//广播地址 BYTE baud=5;//57600 int FrmHandle=-1; int fCmdRet =AutoOpenComPort(&amp;port,&amp;address, baud, &amp;FrmHandle);</pre>		

### 3.1.2 OpenComPort【连接指定串口】

定义	int OpenComPort(int port, BYTE* ComAdr, BYTE baud, int* FrmHandle);		
说明	打开与读写器连接的串口。		
参数	名称	类型	备注
	port	Int	与读写器连接的串口号。1-COM1;2-COM2...
	ComAdr	BYTE*	输入读写器地址，如果在不清楚真实地址的情况下，可以输入广播地址 0xFF，函数调用成功后，此参数将返回真实地址。
	baud	BYTE	串口通讯波特率。0-9600；1-19200；2-38400；5-57600；6-115200。
	FrmHandle	Int*	返回与读写器连接的句柄，此后所有 API 操作都需要此句柄。
返回 (int)	成功：0；失败：非 0；（查看返回值错误代码表）		
参考代码	<pre> int port=1;//打开串口 com1 BYTE address=255;//广播地址 BYTE baud=5;//57600 int FrmHandle=-1; int fCmdRet =OpenComPort(port,&amp;address, baud, &amp;FrmHandle); if(fCmdRet==0) {     //连接成功     ..... } </pre>		



### 3.1.3 CloseComPort 【关闭所有串口】

定义	int CloseComPort();			
说明	撤销所有串口和读写器的连接并释放相应资源。			
参数 (无)	名称	类型	方向	备注
返回 (int)	成功：0 ；失败：非 0 ；（查看返回值错误代码表）			
参考代码				

### 3.1.4 CloseSpecComPort【关闭指定串口】

定义	int CloseSpecComPort(int FrmHandle);		
说明	撤销指定串口和读写器的连接并释放相应资源。		
参数	名称	类型	备注
	FrmHandle	int	与读写器连接的句柄，打开串口 API 获得的句柄。
返回 (int)	成功：0 ；失败：非 0 ；（查看返回值错误代码表）		
参考代码			

### 3.1.5 OpenNetPort【打开 TCP 连接】

定义	Int OpenNetPort(int Port,LPCTSTR IPAddr, BYTE*ComAdr, int *Frmhandle);		
说明	打开与读写器的 TCP 连接。		
参数	名称	类型	备注
	port	int	与读写器连接的端口号。
	IPAddr	LPCTSTR	读写器的 IP 地址。
	ComAdr	BYTE*	输入读写器地址，对于 TCP 连接，可直接输入广播地址 255。
	FrmHandle	Int*	返回与读写器连接的句柄，此后所有 API 操作都需要此句柄。
返回 (int)	成功：0 ；失败：非 0 ；（查看返回值错误代码表）		
参考代码	<pre>int port=6000; LPCTSTR ipAddr ="192.168.1.190"; BYTE ComAdr =255;//广播地址 int FrmHandle=-1; int fCmdRet =OpenNetPort(port, ipAddr ,&amp;ComAdr, &amp;FrmHandle); if(fCmdRet==0) {     //连接成功     ..... }</pre>		

### 3.1.6 CloseNetPort【关闭 TCP 连接】

定义	int CloseNetPort(int FrmHandle);		
说明	关闭与读写器的 TCP 连接并释放相应资源。		
参数	名称	类型	备注
	FrmHandle	int	与读写器连接的句柄，打开 TCP 连接获得的句柄。
返回 (int)	成功：0；失败：非 0；（查看返回值错误代码表）		
参考代码			

## 3.2. 18000-6C 标准命令

### 3.2.1. Inventory\_G2【询查命令】

定义	int Inventory_G2(BYTE * ComAddr,BYTE QValue,BYTE Session,BYTE MaskMem,BYTE *MaskAdr,BYTE MaskLen,BYTE *MaskData,BYTE MaskFlag,BYTE AdrTID,BYTE LenTID,BYTE TIDFlag,BYTE Target,BYTE InAnt,BYTE Scantime,BYTE FastFlag,BYTE *pEPCList,BYTE *Ant,int *Totalen, int *CardNum,int FrmHandle);			
说明	检查有效范围内是否有符合协议的电子标签存在。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	QValue	BYTE	[in]	1 个字节。 bit7：统计数据包标志。 0 – 询查结束后，不发送询查过程统计数据包； 1 – 询查结束后，发送询查过程统计数据包。 Bit5：快速 ID 查询标志位，仅支持 Impinj Monza 系列部分标签，用于快速获取标签的 EPC 和 TID。 0 – 不启用； 1 – 启用。 Bit4 ~ bit0：询查 EPC 标签时使用的初始 Q 值， 注：1、Q 值的设置应为场内的标签数量约等于 2 的 Q 次方。Q 值的范围为 0 ~ 15，若命令中出现了其它值，将返回参数出错的消息。 2、带统计包返回查询速度比不带慢。
	Session	BYTE	[in]	1 个字节，询查 EPC 标签时使用的 Session 值。 0x00：Session 使用 S0； 0x01：Session 使用 S1； 0x02：Session 使用 S2； 0x03：Session 使用 S3。 0xff：读写器自动配置 Session（仅对 EPC 询查有效） 建议单张或者少量标签查询选 S0。
	MaskMem	BYTE	[in]	1 个字节，掩码区。 1 – EPC 存储区； 2– TID 存储区； 3–用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节，掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节，掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍，则 n 为[MaskLen/8]取整再加 1。不够的在低位补 0。

	MaskFlag	BYTE	[in]	1 个字节, 掩码标志位。0 – 不掩码; 1 – 掩码。
	AdrTID	BYTE	[in]	1 个字节, 查询 TID 区的起始地址。
	LenTID	BYTE	[in]	1 个字节, 查询 TID 区的数据字数, 范围 0~15。
	TIDFlag	BYTE	[in]	1 个字节, TID 查询标志位。 0 – 读 EPC ; 1 – 读 TID。
	Target	BYTE	[in]	1 个字节, 查询标签的 Target 值。 0 – Target A ; 1 – Target B。
	InAnt	BYTE	[in]	1 个字节, 本次要进行查询的天线号。 0x80 – 天线 1 ; 0x81 – 天线 2 ; 0x82 – 天线 3 ; 0x83 – 天线 4 ; <b>单口读写器固定 0x80。</b>
	Scantime	BYTE	[in]	1 个字节, 本次命令查询的最大时间, 范围 0–255(*100ms), 当时间为 0 没, 表示口有限制。
	FastFlag	BYTE	[in]	1 个字节, 快速查询的标志位。 0 – 不启用, 不需要指定 Target, Inant, Scantime 3 个参数, 默认 0 即可 ; 1 – 启用, 此时要指定 Target, Inant, Scantime 3 个参数。
	pEPCList	BYTE*	[out]	返回的标签数据, 格式为 : EPC/TID 长度+EPC/TID 号+RSSI。当有多张标签时, 由若干个同格式数据按顺序排列返回。
	Ant	BYTE*	[out]	1 个字节, 查询到标签的天线。 4 口读写器 每个 bit 位表示天线号, 如 0x04, 二进制为 0000 0100, 表示天线 3 查询到该标签。 12 口读写器, 按数字 0–11 分别对应天线 1 到天线 12。
	Totallen	int*	[out]	pEPCList 收到数据的总长度。
	CardNum	int*	[out]	本次查询到的标签张数。 <b>注 : 当使用回调方式返回时, Totallen 和 CardNum 均为 0。</b>
	FrmHandle	int	[in]	连接读写器返回的句柄。
<b>返回 (int)</b>	0x01: 命令执行结束, 同时返回查询到的电子标签数据; 0x02: 查询时间结束, 命令执行强制退出, 同时返回已查询到的标签数据 0xF8: 天线连接检测错误, 当前天线连接可能已经断开; 其余值, 请查看返回值错误代码表。			
<b>参考代码</b>	无。			

### 3.2.2. InventoryMix\_G2【混合查询命令】

定义	int InventoryMix_G2(BYTE * ComAddr,BYTE QValue,BYTE Session,BYTE MaskMem,BYTE *MaskAdr,BYTE MaskLen,BYTE *MaskData,BYTE MaskFlag,BYTE ReadMem,BYTE* ReadAdr,BYTE ReadLen,BYTE* Psd,BYTE Target,BYTE InAnt,BYTE Scantime,BYTE FastFlag,BYTE *pEPCList,BYTE *Ant,int *Totallen, int *CardNum,int FrmHandle);			
说明	检查有效范围内是否有符合协议的电子标签存在, 并在查询到标签EPC号的同时读取指定的标签数据。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	QValue	BYTE	[in]	1 个字节。 bit7 : 统计数据包标志。 0 – 查询结束后, 不发送查询过程统计数据包 ; 1 – 查询结束后, 发送查询过程统计数据包。 bit6 ~ bit0 : 查询 EPC 标签时使用的初始 Q 值, <b>注: 1、Q 值的设置应为场内的标签数量约等于 2 的 Q 次方。Q 值的范围为 0 ~ 15, 若命令中出现了其它值, 将返回参数出错的消息。</b> <b>2、带统计包发送速度比不带慢。</b>
	Session	BYTE	[in]	1 个字节, 查询标签时使用的 Session 值。 0x00 : Session 使用 S0 ; 0x01 : Session 使用 S1 ; 0x02 : Session 使用 S2 ; 0x03 : Session 使用 S3。 <b>建议单张或者少量标签查询选 S0。</b>
	MaskMem	BYTE	[in]	1 个字节, 掩码区。 1 – EPC 存储区 ; 2 – TID 存储区 ; 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节, 掩码的起始 <b>位</b> 地址 (单位 : Bits)。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节, 掩码的 <b>位</b> 长度 (单位 : Bits) 。
	MaskData	BYTE*	[in]	n 个字节, 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍, 则 n 为 [MaskLen/8] 取整再加 1。不够的在低位补 0。
	MaskFlag	BYTE	[in]	1 个字节, 掩码标志位。0 – 不掩码 ; 1 – 掩码。
	ReadMem	BYTE	[in]	1 个字节, 选择要读取的存储区, 0x00 – 保留区 ; 0x01 – EPC 存储区 ; 0x02 – TID 存储区 ; 0x03 –用户存储区。其他值保留。若命令中出现了其它值, 将返回参数出错的消息。

	ReadAdr	BYTE*	[in]	2 个字节，要读取存储数据的起始字地址，高字节在前。
	ReadLen	BYTE	[in]	1 个字节，要读取存储数据的字长度。
	Psd	BYTE*	[in]	4 个字节，标签的访问密码。
	Target	BYTE	[in]	1 个字节，查询标签的 Target 值。 0 – Target A； 1 – Target B。
	InAnt	BYTE	[in]	1 个字节，本次要进行查询的天线号。 0x80 – 天线 1；0x81 – 天线 2； 0x82 – 天线 3；0x83 – 天线 4； 单口读写器固定 0x80。
	Scantime	BYTE	[in]	1 个字节，本次命令查询的最大时间，范围 0–255(*100ms)，当时间为 0 时，表示没有限制。
	FastFlag	BYTE	[in]	1 个字节，快速查询的标志位。 0 – 不启用，不需要指定 Target, Inant, Scantime 3 个参数，默认 0 即可； 1 – 启用，此时要指定 Target, Inant, Scantime 3 个参数。
	pEPCList	BYTE*	[out]	返回的标签数据，格式为： 滚码+EPC/数据长度+EPC/数据+RSSI。当有多张标签时，由若干个同格式数据按顺序排列返回。滚码最高位为 0 表示当前为 EPC，最高位为 1 表示存储数据，同张标签数据先上传 EPC 号，再上传存储数据，低 7 位如果相邻，可确认是同张数据，不相邻非同张标签数据，滚码低 7 位从 0 到 127，每上传一次序号加一，超过 127，从 0 重新开始
	Ant	BYTE*	[out]	1 个字节，查询到标签的天线。 4 口读写器，每个 bit 位表示天线号，如 0x04，二进制为 0000 0100，表示天线 3 查询到该标签。 12 口读写器，按数字 0–11 分别对应天线 1 到天线 12。
	Totallen	int*	[out]	pEPCList 收到数据的总长度。
返回 (int)	CardNum	int*	[out]	本次查询到的标签张数。 注：当使用回调方式返回时，Totallen 和 CardNum 均为 0。
	FrmHandle	int	[in]	连接读写器返回的句柄。
参考代码	无。			





### 3.2.3. InitRFIDCallBack【初始化回调】

定义	void InitRFIDCallBack(pRFIDCallBack pUIDback,BOOL isBackUID,int FrmHandle);			
说明	初始化回调函数 初始化此API后 询查命令和混合询查命令得到的标签可通过回调方式上传。			
参数	名称	类型	方向	备注
	pUIDback	pRFIDCallBack	[in]	回调接口指针 typedef struct _RFIDTag { BYTE PacketParam;//数据包类型标志 BYTE LEN;//长度 LPCTSTR UID; //EPC或UID BYTE RSSI; BYTE ANT; int FrmHandle; }MyStruct,*RFIDTagStruct; Typedefvoid(CALLBACK*pRFIDCallBack)( RFIDTagStruct lpRfid, INT32 nEvt);
	isBackUID	BOOL	[in]	保留。
	FrmHandle	DWORD	[in]	连接读写器返回的句柄。
返回	无			
参考代码	参考 <a href="#">编程示例WC#W多天线测试WMutipleAntDemo</a>			

### 3.2.4. ReadData\_G2【读数据命令】

定义	int ReadData_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE Mem, BYTE WordPtr, BYTE Num, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, BYTE *Data, int *Errorcode, int FrmHandle);			
说明	该命令读取标签的整个或部分保留区、EPC存储器、TID存储器或用户存储器中的数据。从指定的地址开始读，以字为单位。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字，标签的EPC号。
	ENum	BYTE	[in]	1个字节，标签的EPC号字长度，当Enum=255时，使用掩码方式。
	Mem	BYTE	[in]	1个字节，选择要读取的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值，将返回参数出错的消息。。
	WordPtr	BYTE	[in]	1个字节，要读取的起始字地址。
	Num	BYTE	[in]	1个字节，要读取的字长度。
	Password	BYTE*	[in]	4个字节，标签的访问密码。
	MaskMem	BYTE	[in]	1个字节，掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2个字节，掩码的起始位地址（单位：Bits）。范围0~16383。
	MaskLen	BYTE	[in]	1个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节，掩码数据。n等于MaskLen/8。如果MaskLen不是8的整数倍，则n为[MaskLen/8]取整再加1。不够的在低位补0。
	Data	BYTE*	[out]	Num个字，读取到的标签数据。
	ErrorCode	int *	[out]	返回值为0xFC时，返回标签错误信息，请查看ErrorCode错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.2.5. WriteData\_G2【写数据命令】

定义	int WriteData_G2(BYTE *ComAdr, BYTE * EPC, BYTE Wnum, BYTE Enum, BYTE Mem, BYTE WordPtr,BYTE*Writedata,BYTE* Password, BYTE MaskMem,BYTE*MaskAdr,BYTE MaskLen,BYTE * MaskData,int * errorcode,int FrmHandle);			
说明	该命令可以一次性往保留内存、EPC存储器、TID存储器或用户存储器中写入若干个字。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	Wnum	BYTE	[in]	1个字节， 要写入的字数， 当Enum=255时， 使用掩码方式。
	Enum	BYTE	[in]	1个字节， 标签的 EPC 号字长度。当 Enum=255 时， 使用掩码方式。
	Mem	BYTE	[in]	1个字节， 选择要写入的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值，将返回参数出错的消息。。
	WordPtr	BYTE	[in]	1个字节， 要写入的起始字地址。
	Writedata	BYTE*	[in]	WNum 个字节， 要写入的数据。
	Password	BYTE*	[in]	4个字节， 标签的访问密码。
	MaskMem	BYTE	[in]	1个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍，则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.2.6. BlockWrite\_G2【块写命令】

定义	int BlockWrite_G2(BYTE *ComAdr, BYTE * EPC, BYTE Wnum, BYTE Enum, BYTE Mem, BYTE WordPtr, BYTE*Writedata, BYTE* Password, BYTE MaskMem, BYTE*MaskAdr, BYTE MaskLen, BYTE * MaskData, int * errorcode, int FrmHandle);			
说明	该命令可以一次性往保留内存、EPC存储器、TID存储器或用户存储器中写入若干个字。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	Wnum	BYTE	[in]	1个字节， 要写入的字数。
	Enum	BYTE	[in]	1个字节， 标签的 EPC 号字长度。当 Enum=255 时， 使用掩码方式。
	Mem	BYTE	[in]	1个字节， 选择要写入的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值， 将返回参数出错的消息。。
	WordPtr	BYTE	[in]	1个字节， 要写入的起始字地址。
	Writedata	BYTE*	[in]	WNum个字节， 要写入的数据。
	Password	BYTE*	[in]	4个字节， 标签的访问密码。
	MaskMem	BYTE	[in]	1个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍， 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
返回 (int)	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
参考代码	无。			

### 3.2.7. ExtReadData\_G2【扩展读数据命令】

定义	int ExtReadData_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE Mem, BYTE* WordPtr, BYTE Num, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, BYTE *Data, int *Errorcode, int FrmHandle);			
说明	该命令读取标签的整个或部分保留区、EPC存储器、TID存储器或用户存储器中的数据。从指定的地址开始读，以字为单位。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字，标签的EPC号。
	ENum	BYTE	[in]	1个字节，标签的EPC号字长度，当Enum=255时，使用掩码方式。
	Mem	BYTE	[in]	1个字节，选择要读取的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值，将返回参数出错的消息。。
	WordPtr	BYTE*	[in]	2个字节，要读取的起始字地址，高字节在前。
	Num	BYTE	[in]	1个字节，要读取的字长度。
	Password	BYTE*	[in]	4个字节，标签的访问密码。
	MaskMem	BYTE	[in]	1个字节，掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2个字节，掩码的起始位地址（单位：Bits）。范围0~16383。
	MaskLen	BYTE	[in]	1个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节，掩码数据。n等于MaskLen/8。如果MaskLen不是8的整数倍，则n为[MaskLen/8]取整再加1。不够的在低位补0。
	Data	BYTE*	[out]	Num个字，读取到的标签数据。
	ErrorCode	int *	[out]	返回值为0xFC时，返回标签错误信息，请查看ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.2.8. ExtWriteData\_G2【扩展写数据命令】

定义	int WriteData_G2(BYTE *ComAdr, BYTE * EPC, BYTE Wnum, BYTE Enum, BYTE Mem, BYTE* WordPtr,BYTE*Writedata,BYTE* Password, BYTE MaskMem,BYTE*MaskAdr, BYTE MaskLen,BYTE * MaskData,int * errorcode,int FrmHandle);			
说明	该命令可以一次性往保留内存、EPC存储器、TID存储器或用户存储器中写入若干个字。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	Wnum	BYTE	[in]	1个字节， 要写入的字数。
	Enum	BYTE	[in]	1个字节， 标签的 EPC 号字长度。当 Enum=255 时， 使用掩码方式。
	Mem	BYTE	[in]	1个字节， 选择要写入的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值， 将返回参数出错的消息。。
	WordPtr	BYTE*	[in]	2个字节， 要写入的起始字地址。
	Writedata	BYTE*	[in]	WNum个字节， 要写入的数据。
	Password	BYTE*	[in]	4个字节， 标签的访问密码。
	MaskMem	BYTE	[in]	1个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍， 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
返回 (int)	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
参考代码	无。			

### 3.2.9. WriteEPC\_G2【广播写 EPC 号命令】

定义	int WriteEPC_G2(BYTE * ComAddr,BYTE *Password, BYTE *EPC, BYTE ENum, int *Errorcode, int FrmHandle);			
说明	该命令向电子标签写入EPC号。写入的时候，天线有效范围内只能有一张电子标签。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Password	BYTE*	[in]	4个字节，标签的访问密码。
	EPC	BYTE*	[in]	Enum 个字节，要写入的 EPC 号。
	ENum	BYTE	[in]	1 个字节，写入的标签字长度。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			



### 3.2.10. BlockErase\_G2【块擦除命令】

定义	int BlockErase_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE Mem, BYTE WordPtr, BYTE Num, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, int *Errorcode, int FrmHandle);			
说明	该命令可以擦除标签的保留内存、EPC存储器、TID存储器或用户存储器的若干字。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	ENum	BYTE	[in]	1 个字节， 标签的 EPC 号字长度， 当 Enum=255 时， 使用掩码方式。
	Mem	BYTE	[in]	1 个字节， 选择要擦除的存储区， 0x00 – 保留区； 0x01 – EPC 存储区； 0x02 – TID 存储区； 0x03 –用户存储区。其他值保留。若命令中出现了其它值， 将返回参数出错的消息。。
	WordPtr	BYTE*	[in]	1 个字节， 要擦除的起始字地址。
	Num	BYTE	[in]	1 个字节， 要擦除的字长度。
	Password	BYTE*	[in]	4 个字节， 标签的访问密码。
	MaskMem	BYTE	[in]	1 个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍， 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 ErrorCode 错误代码表。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.2.11. Lock\_G2【设定存储区读写保护状态命令】

定义	int Lock_G2(BYTE * ComAddr,BYTE *EPC, BYTE ENum, BYTE select, BYTE setprotect, BYTE *Password,BYTE MaskMem,BYTE *MaskAdr,BYTE MaskLen,BYTE *MaskData, int *Errorcode,int FrmHandle);			
说明	该命令可以设定保留区为可读写、永远可读写、带密码可读写、永远可读写，可以分别设定 EPC 存储器、TID 存储器 and 用户存储器为可写、永远可写、带密码可写、永远不可写。EPC 存储器、TID 存储器 or 用户存储器是永远可读的。而且，TID 存储器是只读的，永远都不可写。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字，标签的EPC号。
	ENum	BYTE	[in]	1个字节，标签的EPC号字长度，当Enum=255时，使用掩码方式。
	select	BYTE	[in]	1个字节， 0x00 – 控制 Kill 密码读写保护设定。 0x01 – 控制访问密码读写保护设定。 0x02 – 控制 EPC 存储器读写保护设定。 0x03 – 控制 TID 存储器读写保护设定。 0x04 – 控制用户存储器读写保护设定。 其它值保留，若出读写器接收到了其他值，将返回参数出错的消息。
	setprotect	BYTE	[in]	1个字节， 当 Select 为 0x00 或 0x01，SetProtect 值代表的意义如下： 0x00 – 设置为可读写 0x01 – 设置为永远可读写 0x02 – 设置为带密码可读写 0x03 – 设置为永远不可读写 当 Select 为 0x02、0x03、0x04 时，SetProtect 值代表的意义如下： 0x00 – 设置为可写 0x01 – 设置为永远可写 0x02 – 设置为带密码可写 0x03 – 设置为永远不可写。
	Password	BYTE*	[in]	4个字节，标签的访问密码。
	MaskMem	BYTE	[in]	1个字节，掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 – 用户存储区。
	MaskAdr	BYTE*	[in]	2个字节，掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节，掩码数据。n 等于 MaskLen/8。如果

				MaskLen 不是 8 的整数倍，则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
<b>返回 (int)</b>	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
<b>参考代码</b>	无。			

### 3.2.12. KillTag\_G2【销毁标签命令】

定义	int KillTag_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE *Killpwd, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, int *Errorcode, int FrmHandle);			
说明	该命令用来销毁标签。标签销毁后，永远不会再处理读写器的命令。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字，标签的EPC号。
	ENum	BYTE	[in]	1个字节，标签的EPC号字长度，当Enum=255时，使用掩码方式。
	Killpwd	BYTE*	[in]	4个字节，标签的杀死密码。
	MaskMem	BYTE	[in]	1个字节，掩码区。 1 - EPC 存储区； 2 - TID 存储区； 3 -用户存储区。
	MaskAdr	BYTE*	[in]	2个字节，掩码的起始位地址（单位：Bits）。范围0~16383。
	MaskLen	BYTE	[in]	1个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n个字节，掩码数据。n等于MaskLen/8。如果MaskLen不是8的整数倍，则n为[MaskLen/8]取整再加1。不够的在低位补0。
	ErrorCode	int *	[out]	返回值为0xFC时，返回标签错误信息，请查看ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.2.13. SingleTagInventory\_G2【单张标签查询命令】

定义	int SingleTagInventory_G2(BYTE * ComAddr, BYTE* EPC, int *EPCLength, int *CardNum,int FrmHandle);			
说明	检查有效范围内是否有符合协议的电子标签存在。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[out]	EPCLength+2 个字节， 返回标签的 EPC 号, 格式为天线号+EPC 号+RSSI
	EPCLength	int*	[out]	1 个字节，返回的 EPC 号长度。
	CardNum	int*	[out]	1 个字节，返回的标签张数。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.3. 缓存操作命令

#### 3.3.1. InventoryBuffer\_G2【缓存查询命令】

定义	int InventoryBuffer_G2(BYTE * ComAddr,BYTE QValue,BYTE Session,BYTE MaskMem,BYTE *MaskAdr,BYTE MaskLen,BYTE *MaskData,BYTE MaskFlag,BYTE AdrTID,BYTE LenTID,BYTE TIDFlag,BYTE Target,BYTE InAnt,BYTE Scantime,BYTE FastFlag ,int * BufferCount,int *CardNum,int FrmHandle);			
说明	检查有效范围内是否有符合协议的电子标签存在。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	QValue	BYTE	[in]	1 个字节。 查询 EPC 标签时使用的初始 Q 值, 注:Q 值的设置应为场内的标签数量约等于 2 的 Q 次方。Q 值的范围为 0~15, 若命令中出现了其它值, 将返回参数出错的消息。
	Session	BYTE	[in]	1 个字节, 查询 EPC 标签时使用的 Session 值。 0x00: Session 使用 S0; 0x01: Session 使用 S1; 0x02: Session 使用 S2; 0x03: Session 使用 S3。 0xff: 读写器自动配置 Session (仅对 EPC 查询有效) 建议单张或者少量标签查询选 S0。
	MaskMem	BYTE	[in]	1 个字节, 掩码区。 1 – EPC 存储区; 2– TID 存储区; 3–用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节, 掩码的起始位地址 (单位: Bits)。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节, 掩码的位长度 (单位: Bits)。
	MaskData	BYTE*	[in]	n 个字节, 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍, 则 n 为 [MaskLen/8]取整再加 1。不够的在低位补 0。
	MaskFlag	BYTE	[in]	1 个字节, 掩码标志位。0 – 不掩码; 1 – 掩码。
	AdrTID	BYTE	[in]	1 个字节, 查询 TID 区的起始字地址。
	LenTID	BYTE	[in]	1 个字节, 查询 TID 区的数据字数, 范围 0~15。
	TIDFlag	BYTE	[in]	1 个字节, TID 查询标志位。 0 – 读 EPC; 1 – 读 TID。
	Target	BYTE	[in]	1 个字节, 查询标签的 Target 值。 0 – Target A; 1 – Target B。

	InAnt	BYTE	[in]	1 个字节，本次要进行查询的天线号。 0x80 –天线 1；0x81 –天线 2； 0x82 –天线 3；0x83 – 天线 4； <b>单口读写器固定 0x80。</b>
	Scantime	BYTE	[in]	1 个字节，本次命令查询的最大时间，范围 0–255(*100ms)，当时间为 0 时，表示没有限制。
	FastFlag	BYTE	[in]	1 个字节，快速查询的标志位。 0 – 不启用，不需要指定 Target,Inant, Scantime 3 个参数，默认 0 即可； 1 – 启用，此时要指定 Target,Inant,Scantime 3 个参数。
	BufferCount	int*	[out]	2 个字节，缓存中记录的标签总数，相同 EPC/TID 数据的标签将被视为同一张标签。若未清空缓存，标签数量为多次询查操作的数量累加。
	CardNum	int*	[out]	2 个字节，本次询查中读取标签的次数，不区分是否多次读取同一张标签。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0x01: 命令执行结束，同时返回询查到的电子标签数据； 0x02: 询查时间结束，命令执行强制退出，同时返回已询查到的标签数据 0xF8: 天线连接检测错误，当前天线连接可能已经断开； 其余值，请查看返回值错误代码表。			
参考代码	无。			

### 3.3.2. ReadBuffer\_G2【获取缓存数据命令】

定义	int ReadBuffer_G2(BYTE *ComAdr, int *Totallen, int *CardNum, BYTE *pEPCList, int FrmHandle);			
说明	该命令用于获取读写器存储区中所有标签信息。响应完成后，缓存中的数据并不丢失，可以多次提取。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Totallen	int*	[out]	读取到的 pEPCList 长度。
	CardNum	int *	[out]	读取到的标签张数。
	pEPCList	BYTE*	[out]	Totallen 个字节，读取到的标签数据，由多个标签组成，每个标签的格式为： ANT+LEN+EPC/TID+RSSI+Count;
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			



### 3.3.3. ClearBuffer\_G2【清空缓存数据命令】

定义	int ClearBuffer_G2(BYTE *ComAdr, int FrmHandle);			
说明	该命令用于清除缓存查询中的缓存数据。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.3.4. GetBufferCnt\_G2【获取缓存标签数量命令】

定义	int GetBufferCnt_G2(BYTE *ComAdr, int *Count, int FrmHandle);			
说明	该命令用于获取缓存区中的标签数量。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Count	int*	[out]	缓存区中的标签数量。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.3.5. SetSaveLen 【设置缓存的 EPC/TID 长度命令】

定义	int SetSaveLen(BYTE *ComAdr, BYTE SaveLen, int FrmHandle);			
说明	该命令用于设置带缓存查询时存储的 EPC/TID 最大长度。如果当前缓存中有标签数据时，设置长度后缓存中的数据将被清空。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	SaveLen	BYTE	[in]	1 个字节，规定缓存 EPC/TID 的最大长度。 0x00 – 128bit 长度，即 16 个字节。 0x01 – 496bit 长度，即 62 个字节。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.3.5. GetSaveLen 【获取缓存的 EPC/TID 长度命令】

定义	int GetSaveLen(BYTE *ComAdr, BYTE *SaveLen, int FrmHandle);			
说明	该命令用于设置带缓存查询时存储的 EPC/TID 最大长度。如果当前缓存中有标签数据时，设置长度后缓存中的数据将被清空。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	SaveLen	BYTE*	[out]	1 个字节，规定缓存 EPC/TID 的最大长度。 0x00 – 128bit 长度，即 16 个字节。 0x01 – 496bit 长度，即 62 个字节。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.4. NXP 标签命令

#### 3.4.1. SetPrivacyByEPC\_G2【单张读保护设置命令】

定义	int SetPrivacyByEPC_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, int *Errorcode, int FrmHandle);			
说明	该命令根据电子标签的 EPC 号, 对标签设置读保护, 使得电子标签不能被任何命令读写, 对标签进行查询操作, 也无法得到电子标签的 EPC 号。仅对 NXP 的 UCODE EPC G2X 标签有效。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字, 标签的EPC号。
	ENum	BYTE	[in]	1 个字节, 标签的 EPC 号字长度, 当 Enum=255 时, 使用掩码方式。
	Password	BYTE*	[in]	4 个字节, 标签的访问密码。
	MaskMem	BYTE	[in]	1 个字节, 掩码区。 1 – EPC 存储区; 2 – TID 存储区; 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节, 掩码的起始位地址 (单位: Bits)。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节, 掩码的位长度 (单位: Bits)。
	MaskData	BYTE*	[in]	n 个字节, 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍, 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int*	[out]	返回值为 0xFC 时, 返回标签错误信息, 请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无。			

### 3.4.2. SetPrivacyWithoutEPC\_G2【多张读保护设置命令】

定义	int SetPrivacyWithoutEPC_G2(BYTE * ComAddr, BYTE *Password, int *Errorcode, int FrmHandle);			
说明	该命令可以为有效范围内的电子标签设定读保护。这个命令与前面一个任务的区别是 当有效范围内存在多张标签的时候 无法知道这个命令操作的是哪一张电子标签。如果要对多张标签进行操作，则标签的访问密码最好是相同的。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Password	BYTE*	[in]	4 个字节，标签的访问密码。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.4.3. ResetPrivacy\_G2【解锁读保护命令】

定义	int ResetPrivacy_G2(BYTE * ComAddr, BYTE *Password, int *Errorcode, int FrmHandle);			
说明	该命令用来给设置了读保护的标签解锁。用这个命令时, 天线有效范围内只能放置一张要被解锁的电子标签。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Password	BYTE*	[in]	4 个字节, 标签的访问密码。
	ErrorCode	int *	[out]	返回值为 0xFC 时, 返回标签错误信息, 请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无。			

### 3.4.4. CheckPrivacy\_G2【检测读保护命令】

定义	int CheckPrivacy_G2(BYTE * ComAddr, BYTE * readpro, int *Errorcode, int FrmHandle);			
说明	<p>该命令不能测试标签是否支持读保护锁定，只能测试标签是否被读保护锁定。对于不支持读保护锁定的电子标签，一致认为没有被锁定。</p> <p>这个命令只能对单张电子标签进行操作，确保天线有效范围内只存在一张电子标签。</p>			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	readpro	BYTE*	[in]	1 个字节， 0x00 – 电子标签没有被设置为读保护。 0x01 – 电子标签被设置读保护。 说明：对于不支持读保护设定的标签，认为没有被设置读保护。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			



### 3.4.5. EASConfigure\_G2【EAS 报警配置命令】

定义	int EASConfigure_G2(BYTE * ComAddr,BYTE *EPC,BYTE ENum,BYTE *Password,BYTE EAS, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, int *Errorcode,int FrmHandle);			
说明	对电子标签的 EAS 状态位进行设置或复位。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	ENum	BYTE	[in]	1 个字节， 标签的 EPC 号字长度， 当 Enum=255 时， 使用掩码方式。
	Password	BYTE*	[in]	4 个字节， 标签的访问密码。
	EAS	BYTE	[in]	1 个字节， 0x00 – 表示设置为关闭 EAS 报警； 0x01 – 表示设置为打开 EAS 报警。
	MaskMem	BYTE	[in]	1 个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍， 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.4.6. EASAlarm\_G2【报警探测命令】

定义	int EASAlarm_G2(BYTE * ComAddr,int *Errorcode, int FrmHandle);			
说明	该命令检测电子标签的 EAS 报警。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 <b>ErrorCode</b> 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

## 3.5. Impinj Monza4 QT 标签命令

### 3.5.1. GetMonza4QTWorkParamter\_G2【读取 Monza4QT 工作参数命令】

定义	int GetMonza4QTWorkParamter_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, BYTE *QTcontrol, int *Errorcode, int FrmHandle);			
说明	这个命令用于读取标签当前设置的工作参数。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字， 标签的EPC号。
	ENum	BYTE	[in]	1 个字节， 标签的 EPC 号字长度， 当 Enum=255 时， 使用掩码方式。
	Password	BYTE*	[in]	4 个字节， 标签的访问密码。
	MaskMem	BYTE	[in]	1 个字节， 掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节， 掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节， 掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节， 掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍， 则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	QTcontrol	BYTE*	[out]	1 个字节， 标签工作参数。 Bit0：标签当前使用的镜像页。 Bit0 = 0 时表示 private； Bit0 = 1 时表示 public。 Bit1：标签是否使能距离保护。 Bit1 = 0 时表示不使能； Bit1 = 1 时表示使能。 其他值保留。
	ErrorCode	int *	[out]	返回值为 0xFC 时， 返回标签错误信息， 请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.5.2. SetMonza4QTWorkParamter\_G2【设置 Monza4QT 工作参数命令】

定义	int SetMonza4QTWorkParamter_G2(BYTE * ComAddr, BYTE *EPC, BYTE ENum, BYTE QTcontrol, BYTE *Password, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, int *Errorcode, int FrmHandle);			
说明	这个命令用于设置标签当前的工作参数。仅对 Impinj 的 Monza 4QT 标签有效。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	EPC	BYTE*	[in]	ENum个字，标签的EPC号。
	ENum	BYTE	[in]	1 个字节，标签的 EPC 号字长度，当 Enum=255 时，使用掩码方式。
	QTcontrol	BYTE	[in]	1 个字节， 标签工作参数。 Bit0：标签当前使用的镜像页。 Bit0 = 0 时表示 private； Bit0 = 1 时表示 public。 Bit1：标签是否使能距离保护。 Bit1 = 0 时表示不使能； Bit1 = 1 时表示使能。 其他值保留。
	Password	BYTE*	[in]	4 个字节，标签的访问密码。
	MaskMem	BYTE	[in]	1 个字节，掩码区。 1 – EPC 存储区； 2 – TID 存储区； 3 –用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节，掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节，掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍，则 n 为[MaskLen/8] 取整再加 1。不够的在低位补 0。
	ErrorCode	int *	[out]	返回值为 0xFC 时，返回标签错误信息，请查看 ErrorCode 错误代码表。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.5.3. Inventory\_QT\_G2【QT 询查命令】

定义	int Inventory_QT_G2(BYTE * ComAddr, BYTE QValue, BYTE Session, BYTE Target, BYTE InAnt, BYTE Scantime, BYTE FastFlag, BYTE *pEPCList, BYTE *Ant, int *TotalLen, int *CardNum, int FrmHandle);
说明	检查有效范围内是否有符合协议的电子标签存在。

参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	QValue	BYTE	[in]	1 个字节, 查询 EPC 标签时使用的初始 Q 值, 注: Q 值的设置应为场内的标签数量约等于 2 的 Q 次方。Q 值的范围为 0 ~ 15, 若命令中出现了其它值, 将返回参数出错的消息。
	Session	BYTE	[in]	1 个字节, 查询 EPC 标签时使用的 Session 值。 0x00: Session 使用 S0; 0x01: Session 使用 S1; 0x02: Session 使用 S2; 0x03: Session 使用 S3。 建议单张或者少量标签查询选 S0。
	Target	BYTE	[in]	1 个字节, 查询标签的 Target 值。 0 - Target A; 1 - Target B。
	InAnt	BYTE	[in]	1 个字节, 本次要进行查询的天线号。 0x80 - 天线 1; 0x81 - 天线 2; 0x82 - 天线 3; 0x83 - 天线 4; 单口读写器固定 0x80。
	Scantime	BYTE	[in]	1 个字节, 本次命令查询的最大时间, 范围 0-255(*100ms), 当时间为 0 时, 表示没有限制。
	FastFlag	BYTE	[in]	1 个字节, 快速查询的标志位。 0 - 不启用, 不需要指定 Target, Inant, Scantime 3 个参数, 默认 0 即可; 1 - 启用, 此时要指定 Target, Inant, Scantime 3 个参数。
	pEPCList	BYTE*	[out]	返回的标签数据, 格式为: EPC/TID 长度+EPC/TID 号+RSSI。当有多张标签时, 由若干个同格式数据按顺序排列返回。
	Ant	BYTE*	[out]	1 个字节, 查询到标签的天线。 4 口读写器 每个 bit 位表示天线号, 如 0x04, 二进制为 0000 0100, 表示天线 3 查询到该标签。 12 口读写器, 按数字 0-11 分别对应天线 1 到天线 12。
	Totallen	int*	[out]	pEPCList 收到数据的总长度。
	CardNum	int*	[out]	本次查询到的标签张数。 注: 当使用回调方式返回时, Totallen 和 CardNum 均为 0。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0x01: 命令执行结束, 同时返回查询到的电子标签数据; 0x02: 查询时间结束, 命令执行强制退出, 同时返回已查询到的标签数据 0xF8: 天线连接检测错误, 当前天线连接可能已经断开;			

	其余值，请查看返回值错误代码表。
参考代码	无。

### 3.5.4. SetTagCustomFunction\_G2【标签自定义命令】

定义	int SetTagCustomFunction(BYTE * ComAddr, BYTE *InlayType, int FrmHandle);			
说明	该命令用于设置读写器启动某些类型标签的自定义工作机制，以实现标签特定功能。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	InlayType	BYTE*	[in/out]	1 个字节， 标签类型，取值范围 0~254。 0x00 – 表示不指定标签类型。 0x01 – 为启动Monza4QT标签的Peek功能（标签状态临时从public转换到private），它将影响标签的读数据、写数据、块写数据、写保护字和写EPC号等操作。 其余值保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

## 3.6. 18000-6B 标准命令

### 3.6.1. InventorySingle\_6B 【单张查询命令】

定义	int InventorySingle_6B(BYTE *address, BYTE *Ant, BYTE* ID_6B, int FrmHandle);			
说明	该命令只能查询单张电子标签。如果多张标签同时处于天线有效范围内，可能无法查询到电子标签。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Ant	BYTE*	[out]	1 个字节，0x04，二进制为 0000 0100，表示天线 3 查询到该标签；0x08，二进制为 0000 1000，表示天线 4 查询到该标签。
	ID_6B	BYTE*	[out]	10 个字节，标签的 UID 号，第 1 个字节是 UID 长度，为 0x08。第 2~9 个字节为标签的 UID 号，低字节在前。第 10 个字节是 RSSI 值。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			



### 3.6.2. InventoryMultiple\_6B【条件查询命令】

定义	int InventoryMultiple_6B(BYTE * ComAddr, BYTE Condition, BYTE StartAddress, BYTE mask, BYTE *ConditionContent, BYTE *Ant, BYTE* ID_6B, int *Cardnum, int FrmHandle);			
说明	该命令按照给定的条件查询电子标签。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Condition	BYTE	[in]	1 个字节，选择标签的要求。 0x00 – 等于条件； 0x01 – 不等于条件； 0x02 – 大于条件； 0x03 – 小于条件。
	StartAddress	BYTE	[in]	1 个字节，用于比较的标签起始地址。
	mask	BYTE	[in]	1 个字节，用来指定要比较的数据。Mask 的每一位对应 ConditionContent 的一个字节。Mask 的最高位对应 (bit7) 对应 ConditionContent 的最左边的一个字节。Mask 的最低位对应 (bit0) 对应 ConditionContent 的最右边一字节。Mask 相应位为 1，表示要把该位在 ConditionContent 中对应的字节与标签中对应的字节进行比较;为 0 则不比较。
	ConditionContent	BYTE*	[in]	8 个字节。用于比较的条件，即用来比较的数据。
	Ant	BYTE*	[out]	1 个字节，查询到标签的天线，如 0x04，二进制为 0000 0100，表示天线 3 查询到该标签。
	ID_6B	BYTE*	[out]	Cardnum*10 个字节，电子标签的 UID。每个 UID 是 10 个字节，第一个字节是 UID 长度，为 0x08，后 8 个字节为标签的 UID 号，最后个为 RSSI。
	Cardnum	int*	[in]	1 个字节，读到的标签张数。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0x15：命令执行结束，同时返回查询到的电子标签数据； 0x16：查询时间结束，命令执行强制退出，同时返回已查询到的标签数据 0xF8：天线连接检测错误，当前天线连接可能已经断开。 其余值，请查看返回值错误代码表。			
参考代码	无。			

### 3.6.3. ReadData\_6B【读数据命令】

定义	int ReadData_6B(BYTE * ComAddr, BYTE* ID_6B, BYTE StartAddress, BYTE Num, BYTE *Data, int *Errorcode, int FrmHandle);			
说明	该命令用来从电子标签的某个指定地址开始读若干个字节。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ID_6B	BYTE*	[in]	8 个字节， 标签的 UID 号
	StartAddress	BYTE	[in]	1 个字节。读数据的起始地址。地址范围为 0 ~ 223。如果地址超出 223, 将返回参数错误信息。
	Num	BYTE	[in]	1 个字节， 要读的数据字节数。 Num 的范围是 1 ~ 32。如果 StartAddress +Num 大于 224、或是 Num 超出 32、或 Num 为 0, 读写器将返回参数错误信息。
	Data	BYTE*	[out]	Num 个字节， 读取到的标签数据。
	Errorcode	int*	[out]	1 个字节， 保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.6.4. WriteData\_6B【写数据命令】

定义	int WriteData_6B(BYTE * ComAddr, BYTE *ID_6B, BYTE StartAddress, BYTE *Writedata, BYTE Writedatalen, BYTE *writtenbyte, int *Errorcode, int FrmHandle);			
说明	该命令用来从电子标签的某个指定地址开始读若干个字节。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ID_6B	BYTE*	[in]	8 个字节， 标签的 UID 号
	StartAddress	BYTE	[in]	1 个字节。写数据的起始地址。地址范围为 0 ~ 223。如果地址超出 223, 将返回参数错误信息。
	Writedata	BYTE*	[in]	Writedatalen 个字节， 要写入的标签数据
	Writedatalen	BYTE	[in]	1 个字节， 要写入的数据字节数。
	writtenbyte	BYTE*	[out]	1 个字节， 保留。
	Errorcode	int*	[out]	1 个字节， 保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

### 3.6.5. CheckLock\_6B【锁定检测命令】

定义	int CheckLock_6B(BYTE * ComAddr, BYTE *ID_6B, BYTE StartAddress, BYTE *ReLockState, int *Errorcode, int FrmHandle);			
说明	该命令用来检测指定的字节是否锁定。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ID_6B	BYTE*	[in]	8 个字节，标签的 UID 号
	StartAddress	BYTE	[in]	1 个字节。要检测是否被锁定的字节的地址。地址范围为 0~ 223。如果地址超出范围，将返回参数错误信息。
	ReLockState	BYTE*	[out]	1 个字节， 0x00 – 未锁定； 0x01 – 已经锁定，不能再次锁定。
	Errorcode	int*	[out]	1 个字节，保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无。			

### 3.6.6. Lock\_6B【锁定命令】

定义	int Lock_6B(BYTE * ComAddr, BYTE *ID_6B, BYTE StartAddress, int *Errorcode, int FrmHandle);			
说明	该命令用来检测指定的字节是否锁定。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ID_6B	BYTE*	[in]	8 个字节， 标签的 UID 号
	StartAddress	BYTE	[in]	1 个字节。要锁定的字节的地址。地址范围为 0~223。如果地址超出范围 将返回参数错误信息。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无。			

## 3.7. 读写器自定义命令

### 3.7.1. GetReaderInformation 【获取读写器信息命令】

定义	int GetReaderInformation(BYTE* ComAdr, BYTE* VersionInfo, BYTE* ReaderType, BYTE* TrType, BYTE* dmaxfre, BYTE* dminfre, BYTE* powerdBm, BYTE* ScanTime, BYTE* Ant, BYTE* BeepEn, BYTE* OutputRep, BYTE* CheckAnt, int FrmHandle);			
说明	获得读写器的信息，这其中包括读写器地址 (ComAdr) 和读写器软件版本 (VersionInfo) 的信息等多项信息			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	1 个字节，使用连接读写器返回的读写器地址。
	VersionInfo	BYTE*	[out]	2 个字节，读写器版本信息。第 1 个字节为版本号，第 2 个字节为子版本号柄。
	ReaderType	BYTE*	[out]	1 个字节，读写器类型代码。
	TrType	BYTE*	[out]	1 个字节，读写器支持协议。
	dmaxfre	BYTE*	[out]	1 个字节，Bit7–Bit6 用于频段设置用；Bit5–Bit0 表示当前读写器工作的最大频点。
	dminfre	BYTE*	[out]	1 个字节，Bit7–Bit6 用于频段设置用；Bit5–Bit0 表示当前读写器工作的最小频点。
	powerdBm	BYTE*	[out]	1 个字节，读写器的输出功率。范围是 0 到 30，单位 dBm。
	ScanTime	BYTE*	[out]	1 个字节，查询时间。读写器收到查询命令后，在查询时间内，会给上位机应答。
	Ant	BYTE*	[out]	1 个字节，天线配置信息。
	BeepEn	BYTE*	[out]	1 个字节，蜂鸣器开关信息。
	OutputRep	BYTE*	[out]	1 个字节，通知输出端口参数。
	CheckAnt	BYTE*	[out]	1 个字节，天线检测信息。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

注：频段设置如下表：

MaxFre(Bit7)	MaxFre(Bit6)	MinFre(Bit7)	MinFre(Bit6)	FreqBand
0	0	0	1	Chinese band2
0	0	1	0	US band
0	0	1	1	Korean band
0	1	0	0	EU band
0	1	1	0	Ukraine band

1	0	0	0	Chinese band1
1	1	0	0	US band3

### 3.7.2 SetAddress 【设置读写器地址命令】

定义	int SetAddress(BYTE * ComAddr, BYTE ComAdrData, int FrmHandle);			
说明	执行该命令后，读写器将会把读写器地址改为用户给定的值，并把这个值写入 EEPROM 保存。出厂时默认值是 0x00。允许用户的修改范围是 0x00~0xFE。当用户写入的值是 0xFF 时，读写器将会自动恢复成默认值 0x00。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ComAdrData	BYTE	[in]	1 个字节，要修改的新地址。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			



### 3.7.3 SetInventoryScanTime【查询命令最大响应时间设置命令】

定义	int SetInventoryScanTime(BYTE *address, BYTE ScanTime, int FrmHandle);			
说明	该命令用于设置查询命令的最大响应时间。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ScanTime	BYTE	[in]	1 个字节, 查询命令响应时间, 范围 0 到 255。 设置为 0 时, 该参数无效, 一直等待, 直到该轮读完。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无			

### 3.7.4. SetRfPower【功率设置命令】

定义	int SetRfPower(BYTE *address, BYTE PowerDbm, int FrmHandle);			
说明	设置读写器功率。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	PowerDbm	BYTE	[in]	1 个字节，读写器功率，范围 0~30dBm。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

### 3.7.5. SetRegion【工作频率设置命令】

定义	int SetRegion(BYTE *address, BYTE dmaxfre, BYTE dminfre, int FrmHandle);			
说明	该命令设置读写器工作地上限频率，下限频率。上限频率必须大于或等于下限频率。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	dmaxfre	BYTE	[in]	1 个字节，Bit7–Bit6 用于频段设置用；Bit5–Bit0 表示当前读写器工作的最大频点。
	dminfre	BYTE	[in]	1 个字节，Bit7–Bit6 用于频段设置用；Bit5–Bit0 表示当前读写器工作的最小频点。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

注：频段设置如下表：

MaxFre(Bit7)	MaxFre(Bit6)	MinFre(Bit7)	MinFre(Bit6)	FreqBand
0	0	0	1	Chinese band2
0	0	1	0	US band
0	0	1	1	Korean band
0	1	0	0	EU band
0	1	1	0	Ukraine band
1	0	0	0	Chinese band1
1	1	0	0	US band3

各频段计算公式：

Chinese band2:  $F_s = 920.125 + N * 0.25 \text{ (MHz)}$  其中  $N \in [0, 19]$ 。

US band:  $F_s = 902.75 + N * 0.5 \text{ (MHz)}$  其中  $N \in [0, 49]$ 。

Korean band:  $F_s = 917.1 + N * 0.2 \text{ (MHz)}$  其中  $N \in [0, 31]$ 。

EU band:  $F_s = 865.1 + N * 0.2 \text{ (MHz)}$  其中  $N \in [0, 14]$ 。

Ukraine band:  $F_s = 868.0 + N * 0.1 \text{ (MHz)}$  其中  $N \in [0, 6]$ 。

Chinese band1:  $F_s = 840.125 + N * 0.25 \text{ (MHz)}$  其中  $N \in [0, 19]$ 。

US band3:  $F_s = 902 + N * 0.5 \text{ (MHz)}$  其中  $N \in [0, 52]$ 。

### 3.7.6. SetBaudRate【波特率设置命令】

定义	int SetBaudRate(BYTE * ComAddr, BYTE baud, int FrmHandle);			
说明	设置读写器通讯波特率			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	baud	BYTE	[in]	1 个字节，波特率。 0 – 9600bps; 1 – 19200bps; 2 – 38400bps; 5 – 57600bps; 6 – 115200bps.
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无			

### 3.7.7. BuzzerAndLEDControl【声光控制命令】

定义	int BuzzerAndLEDControl(BYTE * ComAddr, BYTE AvtiveTime, BYTE SilentTime, BYTE Times, int FrmHandle);			
说明	该命令用来控制LED灯和蜂鸣器按一定规律闪烁和鸣叫。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	AvtiveTime	BYTE	[in]	1 个字节，LED 灯亮和蜂鸣器鸣叫时间 (ActiveT*50ms)，默认值为零。0<=ActiveT<=255。
	SilentTime	BYTE	[in]	1 个字节，LED 灯和蜂鸣器静默时间 (SilentT *50ms)，默认值为零。0<=SilentT <=255。 .
	Times	BYTE	[in]	1 个字节，LED 灯亮和蜂鸣器鸣叫次数(0<=Times<=255) 默认值为零。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无			

### 3.7.8. SetAntennaMultiplexing【天线配置命令】

定义	int SetAntennaMultiplexing(BYTE *ComAdr, BYTE Ant, int FrmHandle);			
说明	该命令用于4口读写器天线配置。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Ant	BYTE	[in]	1 个字节，天线配置信息。天线至少要配置一个。 Bit0：天线 1 配置位。 Bit0=0 天线 1 关闭；Bit0=1 天线 1 打开。 Bit1：天线 2 配置位。 Bit1=0 天线 2 关闭；Bit1=1 天线 2 打开。 Bit2：天线 3 配置位。 Bit2=0 天线 3 关闭；Bit2=1 天线 3 打开。 Bit3：天线 4 配置位。 Bit3=0 天线 4 关闭；Bit3=1 天线 4 打开。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无			

### 3.7.9. SetBeepNotification 【蜂鸣器设置命令】

定义	int SetBeepNotification(BYTE *ComAdr,BYTE BeepEn, int FrmHandle);			
说明	该命令用于设置蜂鸣器开关。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	BeepEn	BYTE	[in]	1 个字节, 0x00 – 蜂鸣器关; 0x01 – 蜂鸣器开。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无			

### 3.7.10. SetRelay【继电器控制命令】

定义	int SetRelay(BYTE *ComAdr, BYTE RelayTime, int FrmHandle);			
说明	该命令用于控制继电器闭合。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	RelayTime	BYTE	[in]	1 个字节，继电器闭合时间 (1~255)*50ms。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			



### 3.7.11. GetGPIOStatus 【读取 GPIO 状态命令】

定义	int GetGPIOStatus(BYTE *ComAdr, BYTE *OutputPin, int FrmHandle);			
说明	该命令用于读取通用GPIO口输出，默认输出为高电平。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	OutputPin	BYTE*	[out]	1 个字节，Bit0 代表 IN1 引脚状态，Bit4–Bit5 分别代表 Out1–Out2 状态其他位保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

### 3.7.12. SetGPIO【GPIO 控制命令】

定义	int SetGPIO(BYTE *ComAdr, BYTE OutputPin, int FrmHandle);			
说明	该命令用于设置GPIO口输出状态。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	OutputPin	BYTE	[in]	1 个字节，GPIO 口 (Out1-Out2 引脚)输出状态。Bit0-Bit1 分别控制 Out1-Out2 引脚，Bit2-Bit7 保留。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

### 3.7.13. GetSeriaNo【获取读写器序列号命令】

定义	int GetSeriaNo(BYTE * ComAddr, BYTE* SeriaNo, int FrmHandle);			
说明	该命令用于获取读写器4字节的唯一序列号。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	SeriaNo	BYTE*	[out]	4 字节，读写器序列号。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看返回值错误代码表。			
参考代码	无			

### 3.7.14. SetCheckAnt【天线检测命令】

定义	int SetCheckAnt(BYTE *address, BYTE CheckAnt, int FrmHandle);			
说明	该命令用于设置读写标签的过程中是否进行天线检测。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	CheckAnt	BYTE	[in]	1 字节， 天线检测开关。 0x00 – 关闭读写标签过程中的天线检测。 0x01 – 开启读写标签过程中的天线检测。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看返回值错误代码表。			
参考代码	无			

### 3.7.15. WriteRfPower 【设置写功率命令】

定义	int WriteRfPower(BYTE * ComAddr, BYTE PowerDbm, int FrmHandle);			
说明	该命令用来设置写标签时的功率。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	PowerDbm	BYTE	[in]	1 个字节，读写器的写功率， 0-30 ； bit7=0 – 不使能 ； bit7=1 – 使能；。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.16. ReadRfPower【读取写功率命令】

定义	int WriteRfPower(BYTE * ComAddr, BYTE PowerDbm, int FrmHandle);			
说明	该命令用来读取写标签时的功率。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	PowerDbm	BYTE*	[out]	1 个字节，读写器的写功率， 0-30 ； bit7=0 – 不使能 ； bit7=1 – 使能；。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.17. RetryTimes【写重复次数设置命令】

定义	int RetryTimes(BYTE * ComAddr, BYTE *Times, int FrmHandle);			
说明	该命令用来读取写标签时的功率。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Times	BYTE*	[out]	1 个字节, bit7=1 – 设置重写次数; bit7=0 – 读取重写次数; 范围 0-7。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看错误代码表。			
参考代码	无			

### 3.7.18. SetReadMode【实时模式设置命令】

定义	int SetReadMode(BYTE *ComAdr,BYTE ReadMode ,int FrmHandle);			
说明	该命令用来设置读写器实时查询模式。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	ReadMode	BYTE	[in]	1 个字节, 0 – 应答模式; 1 – 实时查询模式; 2 – 触发模式。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看错误代码表。			
参考代码	无			



### 3.7.19. SetReadParameter 【设置实时模式工作参数命令】

定义	int SetReadParameter(BYTE *ComAdr, BYTE *Parameter, BYTE MaskMem, BYTE *MaskAdr, BYTE MaskLen, BYTE *MaskData, BYTE MaskFlag, BYTE AdrTID, BYTE LenTID, BYTE TIDFlag, int FrmHandle);			
说明	该命令用来设置读写器实时查询模式下的工作参数。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Parameter	BYTE*	[in]	<p>5 个字节,</p> <p>Parameter[0]: 自动实时查询的标签类型。若命令中出现了其它值, 将返回参数出错的消息。</p> <p>0 – EPC C1G2 (ISO18000-6C) 标签。</p> <p>1 – ISO18000-6B 标签。</p> <p>Parameter[1]: 读写器自动实时查询标签的间隔时间。</p> <p>0x00 – 10ms ;</p> <p>0x01 – 20ms ;</p> <p>0x02 – 30ms ;</p> <p>0x03 – 50ms ;</p> <p>0x04 – 100ms。</p> <p>Parameter[2]: 标签过滤时间。以 1s 为单位, 参数范围 0 ~ 255s。本参数表示, 在自动实时查询模式下, 读写器在过滤时间内如果多次读到同一个标签, 只会上传一次。如果该参数为 0, 则表示不启用标签过滤功能。</p> <p>Parameter[3]: 1 个字节, 查询 EPC 标签时使用的初始 Q 值。Q 值的设置应为场内的标签数量约等于 2Q。Q 值的范围为 0 ~ 15, 若命令中出现了其它值, 将返回参数出错的消息。</p> <p>Parameter[4]: 查询 EPC 标签时使用的 Session 值。</p> <p>0x00 : Session 使用 S0 ;</p> <p>0x01 : Session 使用 S1 ;</p> <p>0x02 : Session 使用 S2 ;</p> <p>0x03 : Session 使用 S3。</p> <p>0xff : 读写器自动配置 Session (仅对 EPC 查询有效) 。</p>
	MaskMem	BYTE	[in]	<p>1 个字节, 掩码区。</p> <p>1 – EPC 存储区 ;</p> <p>2 – TID 存储区 ;</p>

				3-用户存储区。
	MaskAdr	BYTE*	[in]	2 个字节，掩码的起始位地址（单位：Bits）。范围 0~16383。
	MaskLen	BYTE	[in]	1 个字节，掩码的位长度（单位：Bits）。
	MaskData	BYTE*	[in]	n 个字节，掩码数据。n 等于 MaskLen/8。如果 MaskLen 不是 8 的整数倍，则 n 为[MaskLen/8]取整再加 1。不够的在低位补 0。
	AdrTID	BYTE	[in]	1 个字节，查询 TID 区的起始字地址。
	LenTID	BYTE	[in]	1 个字节，查询 TID 区的数据字数，范围 0~15。
	TIDFlag	BYTE	[in]	1 个字节，TID 查询标志位。 0 – 读 EPC； 1 – 读 TID。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看错误代码表。			
参考代码	无			

### 3.7.20. GetReadParameter【读取实时模式工作参数命令】

定义	int GetReadParameter (BYTE *ComAdr,BYTE* Parameter,int FrmHandle);			
说明	该命令用来设置读写器实时查询模式。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	Parameter	BYTE*	[out]	Parameter 依顺序为： ReadMode：1 个字节，读写器当前工作模式。 TagProtocol：1 个字节，自动实时查询模式的标签类型。 ReadPauseTime：1 个字节，读写器自动实时查询标签的间隔时间。 FilterTime：1 个字节，自动实时查询模式的标签过滤时间。 QValue：1 个字节，自动实时查询 EPC 标签时使用的初始 Q 值。 Session：1 个字节，自动实时查询 EPC 标签时使用的 Session 值。 MaskMem、MaskAdr、MaskLen、MaskData：自动实时查询模式下读写器查询 EPC C1G2 标签掩码条件参数。其中 MaskMem 为一个字节；MaskAdr 为两个字节，高字节在前；MaskLen 为一个字节；MaskData 固定为 32 个字节，超过 MaskLen 指示的内容为零。 AdrTID：自动实时查询 TID 区的起始地址。 LenTID：自动实时查询 TID 区的数据字数。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败，请查看错误代码表。			
参考代码	无			

### 3.7.21. SetDRM 【设置 DRM】

定义	int SetDRM(BYTE * ComAddr, BYTE DRM, int FrmHandle);			
说明	该命令用来设置 DRM 状态			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	DRM	BYTE	[in]	1 字节， DRM 状态 0 – 关闭； 1 – 打开。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.22. GetDRM【读取 DRM】

定义	int GetDRM(BYTE * ComAddr, BYTE* DRM, int FrmHandle);			
说明	该命令用来读取 DRM 状态			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	DRM	BYTE*	[out]	1 字节， DRM 状态 0 – 关闭； 1 – 打开。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.23. GetReaderTemperature【读取温度】

定义	int GetReaderTemperature (BYTE * ComAddr, BYTE* PlusMinus,BYTE* Temperature, int FrmHandle);			
说明	该命令用来读取读写器当前温度。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	PlusMinus	BYTE*	[out]	1 字节， 温度状态， 0 - 零下； 1 - 零上。
	Temperature	BYTE*	[out]	1 字节， 温度数值， 单位℃
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.24. MeasureReturnLoss 【测量回损】

定义	int MeasureReturnLoss(BYTE * ComAddr, BYTE* TestFreq, BYTE Ant, BYTE* ReturnLoss,int FrmHandle);			
说明	该命令用来测量读写器指定端口的回波损耗。			
参数	名称	类型	方向	备注
	ComAddr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	TestFreq	BYTE*	[in]	4 字节， 测量频点。
	Ant	BYTE	[in]	1 字节， 测量天线端口， 0 – 天线 1 ； 1 – 天线 2。 2 – 天线 3 ； 3 – 天线 4。
	ReturnLoss	BYTE*	[out]	1 字节， 回波损耗， 单位 dB。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败， 请查看错误代码表。			
参考代码	无			

### 3.7.25. SetNotificationPulseOutput【通知输出端口命令】

定义	int SetNotificationPulseOutput(BYTE *ComAdr, BYTE OutputRep, int FrmHandle);			
说明	该命令用于设置当查询到标签时是否在对端口输出一个2ms的低脉冲。			
参数	名称	类型	方向	备注
	ComAdr	BYTE*	[in/out]	使用连接读写器返回的读写器地址。
	OutputRep	BYTE	[in]	1 个字节, Bit0 : 对 应 天 线 1 和 Out1 引 脚。Bit0=1 时, 当天线 1 查询到标签 时, Out1 引脚输出一个 2ms 的低脉 冲; Bit0=0 时, 无信号输出。 Bit1 : 对 应 天 线 2 和 Out2 引 脚。Bit1=1 时, 当天线 2 查询到标签 时, Out2 引脚输出一个 2ms 的低脉 冲; Bit1=0 时, 无信号输出。 Bit2 : 对 应 天 线 3 和 Out3 引 脚。Bit2=1 时, 当天线 3 查询到标签 时, Out3 引脚输出一个 2ms 的低脉 冲; Bit2=0 时, 无信号输出。 Bit3 : 对 应 天 线 4 和 Out4 引 脚。Bit3=1 时, 当天线 4 查询到标签 时, Out4 引脚输出一个 2ms 的低脉 冲; Bit3=0 时, 无信号输出。 其它位保留, 默认值为 0。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无			



### 3.7.26. ReadActiveModeData 【获取主动上传数据】

定义	int ReadActiveModeData(BYTE * LoadData, int *Datalength, int FrmHandle);			
说明	该命令用于获取主动模式或实时模式下获取读写器上传的数据。			
参数	名称	类型	方向	备注
	LoadData	BYTE*	[out]	Datalength 个字节, 上传的裸数据, 格式查看用户手册。
	Datalength	BYTE*	[out]	LoadData的字节长度。
	FrmHandle	int	[in]	连接读写器返回的句柄。
返回 (int)	0:表示成功。不等于 0:表示失败, 请查看返回值错误代码表。			
参考代码	无			

## 附录 1. 返回值错误代码表

错误代码	描述
0x00	无错误，表示 API 调用成功。
0x01	查询时间结束前返回。
0x02	指定的查询时间溢出。
0x05	访问密码错误。
0x09	销毁密码错误。
0x0A	销毁密码不能为全 0。
0x0B	电子标签不支持该命令。
0x0C	对该命令，访问密码不能为 0。
0x0D	电子标签已经被设置了读保护，不能再次设置。
0x0E	电子标签没有被设置读保护，不需要解锁。
0x10	有字节空间被锁定，写入失败。
0x11	不能锁定。
0x12	已经锁定，不能再次锁定。
0x13	参数保存失败,但设置的值在读写模块断电前有效。
0x14	无法调整。
0x15	查询时间结束前返回。
0x17	本条消息之后，还有消息。
0x18	读写模块存储空间已满。
0x19	电子不支持该命令或者访问密码不能为 0
0xF8	天线检测错误。
0xF9	命令执行出错。
0xFA	有电子标签，但通信不畅，无法操作。
0xFB	无电子标签可操作。
0xFC	电子标签返回错误代码。
0xFD	命令长度错误。
0xFE	不合法的命令。
0xFF	参数错误。
0x30	通讯错误。
0x33	通讯繁忙，设备正在执行其他命令。
0x35	端口已打开
0x37	无效句柄。

## 附录 2. ErrorCode 错误代码表

错误代码	描述
0x00	其它错误，全部捕捉未被其它代码覆盖的错误。
0x03	存储器超限或不被支持的 PC 值，规定存储位置不存在或标签不支持 PC 值。
0x04	存储器锁定，存储位置锁定永久锁定，且不可写入。
0x0B	电源不足，标签电源不足，无法执行存储写入操作。
0x0F	非特定错误，标签不支持特定错误代码。