

MOTEC 驱动器函数库使用手册

Version 2.1

MOTEC(中国)营业体系

2017-05-31

(本手册适用于 MOTEC 步进/直流/交流伺服驱动器)

目 录

1. 函数库说明.....	4
2. 函数库安装.....	6
3. 建立自己的应用.....	6
4. 函数描述.....	11
4.1 IsOpenCom.....	11
4.2 OpenCom.....	11
4.3 CloseCom.....	11
4.4 SetTimeCont.....	12
4.5 AddChoose.....	12
4.6 InputAddress.....	12
4.7 Base.....	12
4.8 EchoTesting.....	13
4.9 SendCommand.....	13
4.10 PutParameter.....	13
4.11 GetParameter.....	14
4.12 SaveParameter.....	14
4.13 GetSerialNumber.....	14
4.14 GetVersionInformation.....	14
4.15 GetEncoderCounterPerRound.....	14
4.16 PutEncoderCounterPerRound.....	15
4.17 GetDigitalOutput.....	15
4.18 PutDigitalOutput.....	15
4.19 GetDigitalInput.....	16
4.20 GetAnalogInput.....	16
4.21 GetAnalogOutput.....	16
4.22 PutAnalogOutput.....	17
4.23 GetMotorEnableStatus.....	17
4.24 PutMotorEnableStatus.....	17
4.25 GetOperationMode.....	17
4.26 PutOperationMode.....	18
4.27 GetControlMode.....	18
4.28 PutControlMode.....	18
4.29 GetSProfileParameter.....	19
4.30 PutSProfileParameter.....	19
4.31 GetTProfileParameter.....	19
4.32 PutTProfileParameter.....	20
4.33 MoveAbsolute.....	20
4.34 MoveRelative.....	20
4.35 Go.....	20
4.36 Stop.....	21
4.37 QuickStop.....	21

4.38	GetProfileType.....	21
4.39	PutProfileType.....	21
4.40	GetStopDecelleration.....	22
4.41	PutStopDecelleration.....	22
4.42	GetStatusFlag.....	22
4.43	GetPosition.....	22
4.44	GetVelocity.....	23
4.45	GetCurrent.....	23
4.46	ResetPosition.....	23
4.47	PutVelocitySetpoint.....	23
4.48	PutCurrentSetpoint.....	24
4.49	TrigDataLog.....	24
4.50	GetDataLogChnA.....	24
4.51	GetDataLogChnB.....	24
4.52	GetDataLogChnC.....	25
4.53	GetDataLogChnD.....	25
4.54	GetLogDataNumber.....	25
4.55	PutLogDataNumber.....	25
4.56	PutDataLoggingSetting.....	26
4.57	GetDataLoggingSetting.....	26
4.58	GetloggingCompleteFlag.....	27
4.59	PutBrakeAction.....	27
4.60	GetBrakeStatus.....	27
4.61	ClearErrors.....	28
4.62	GetError.....	28
4.63	GetErrorDescriptionID.....	28
4.64	GetErrorDescription.....	28
4.65	SeekLimit.....	29
4.66	SeekIndex.....	29
4.67	Homming.....	29
4.68	Jog.....	29
4.69	ClearMotionFlags.....	30
4.70	GetDoneFlag.....	30
4.71	EnableDoneFlag.....	30
5.	联系方式.....	31



注意：使用 **OpenCom** 函数打开串口成功后，必须使用 **EchoTesting** 函数，确定驱动器的校验方式，否则可能无法正常通讯！

1. 函数库说明

MotecDriverCmd 函数库是 MOTEC 提供的用于上位机为 PC 机，用于网络运行模式时上位机应用程序的编程。用户可以用 VC、VB、C#、Delphi 等编程工具进行应用程序的编写。无论是 RS232 还是 RS485 都可以执行多台 MOTEC 驱动器的联网工作。利用 MotecDriverCmd，用户能方便快捷的构建自己的应用，而无需从底层的串口操作开始。

其串口设置为 8 个数据位，一个起始位，一个停止位，无校验。

下面的内容先对函数做一简单的说明，具体的函数说明请参见第 4 章节的说明。

表 1. 函数库函数说明

序号	函数名称	函数功能描述
1	IsOpenCom	判断端口是否打开
2	OpenCom	打开串口
3	CloseCom	关闭串口
4	SetTimeCont	设置时间常数
5	Addchoose	地址标志
6	InputAddress	获得连接上的 MOTEC 驱动器台数
7	Base	获得连接上的 MOTEC 驱动器地址
8	EchoTesting	发送通讯测试指令
9	SendCommand	发送命令，并接收返回值
10	PutParameter	设置驱动器参数表参数
11	GetParameter	获取驱动器参数表参数
12	SaveParameter	将驱动器 RAM 中的参数表保存到 FLASH 中
13	GetSerialNumber	读取 32 位控制器序列号
14	GetVersionInformation	读取控制器硬件和软件版本信息
15	GetEncoderCounterPerRound	获取编码器每转计数
16	PutEncoderCounterPerRound	设置编码器每转计数
17	GetDigitalOutput	获取数字输出口状态
18	PutDigitalOutput	设置数字输出口状态
19	GetDigitalInput	获取数字输入口状态
20	GetAnalogInput	获取模拟输入口状态
21	GetAnalogOutput	获取模拟输出口状态
22	PutAnalogOutput	设置模拟输出口状态
23	GetMotorEnableStatus	使能/释放电机
24	PutMotorEnableStatus	设置电机状态(使能/释放)
25	GetOperationMode	获取驱动器操作模式

26	PutOperationMode	设置驱动器操作模式
27	GetControlMode	获取驱动器控制模式
28	PutControlMode	设置驱动器控制模式
29	GetSPProfileParameter	获取 S 曲线轨迹参数
30	PutSPProfileParameter	设置 S 曲线轨迹参数
31	GetTPProfileParameter	获取 T 曲线轨迹参数
32	PutTPProfileParameter	设置 T 曲线轨迹参数
33	MoveAbsolute	设置绝对运动目的地的位置
34	MoveRelative	设置相对运动的运动距离
35	Go	开始运动
36	Stop	停止运动
37	QuickStop	紧急停止运动
38	GetProfileType	获取点到点运动轨迹类型
39	PutProfileType	设置点到点运动轨迹类型
40	GetStopDeceleration	获取停止运动减速度
41	PutStopDeceleration	设置停止运动减速度
42	GetStatusFlag	获取驱动器状态信息
43	GetPosition	获取电机位置信息
44	GetVelocity	获取电机速度信息
45	GetCurrent	获取电机电流信息
46	ResetPosition	电机位置归零
47	PutVelocitySetpoint	在速度控制模式下，设置速度设定值
48	PutCurrentSetpoint	在电流控制模式下，设置电流设定值
49	TrigDataLog	触发数据记录，根据所设定的数据记录参数，从 0 开始记录数据；
50	GetDataLogChnA	获取通道 1 中所记录的数据
51	GetDataLogChnB	获取通道 2 中所记录的数据
52	GetDataLogChnC	获取通道 3 中所记录的数据
53	GetDataLogChnD	获取通道 4 中所记录的数据
54	GetLogDataNumber	获取数据记录长度
55	PutLogDataNumber	设置数据记录长度
56	PutDataLoggingSetting	设置数据记录参数
57	GetDataLoggingSetting	获取数据记录参数
58	GetloggingCompleteFlag	获取数据记录完成标志
59	PutBrakeAction	启动/释放抱闸
60	GetBrakeStatus	获取抱闸状态
61	ClearErrors	清除控制器故障信息
62	GetError	读取控制器的故障代码
63	GetErrorDescriptionID	获取故障状态编号
64	GetErrorDescription	根据故障编号获取故障信息
65	SeekLimit	启动电机运动，直至到达限位开关
66	SeekIndex	启动电机寻找 Z 脉冲
67	Homming	启动电机回原点

68	Jog	使电机按 Jog 方式运动
69	ClearMotionFlags	清除运动完成和接近完成标志
70	GetDoneFlag	获取运动完成和接近完成标志
71	EnableDoneFlag	使能/取消 运动完成标志和接近完成标志

2. 函数库安装

直接注册 MotecDriverCmd.ocx 即可。

问：关于 Win7 系统下注册 ocx 组件，无法注册，注册失败，这里提供解决方案。

答：win7 系统下会涉及“管理员身份”，是很多 OCX 组件无法使用的原因。那如何正确注册呢。以下提供详细步骤，其实很简单，但是只能手动注册。

进入“开始”->“所有程序”->“附件”->“命令提示符”(cmd.exe),这里比较关键，必须进行右击 cmd.exe,“以管理员身份……”进入该 cmd.exe,即可进入管理员身份来注册 ocx。然后需要手动输入命令。

具体输入命令如下：regsvr32 "D:\XX \ XX.ocx".即可完成注册(此处 XX 表示所在的文件夹和文件名)。

3. 建立自己的应用

成功注册函数库之后，用户便可以进行应用程序的编写。以下以 visual C++为例说明如何利用 MOTEC 提供的函数库进行应用程序的编程。

步骤：

1) 启动 Microsoft Visual C++6.0，单击[File]菜单，选择[New]，在 Projects 选项卡下，选择 MFC AppWizard(exe),在右侧的[Project name]文本框中，输入工程名：Sample，如图 3.1 所示。

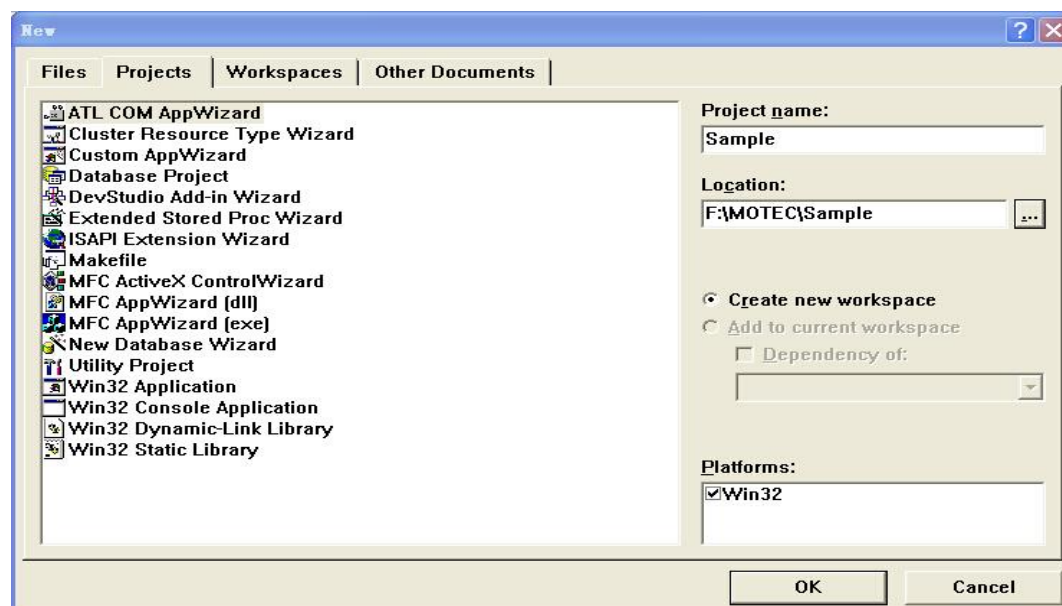


图 3.1MFC Step1

2) 单击[OK]按钮, 出现如图 3.2 所示对话框, 选择 Dialog based 选项。

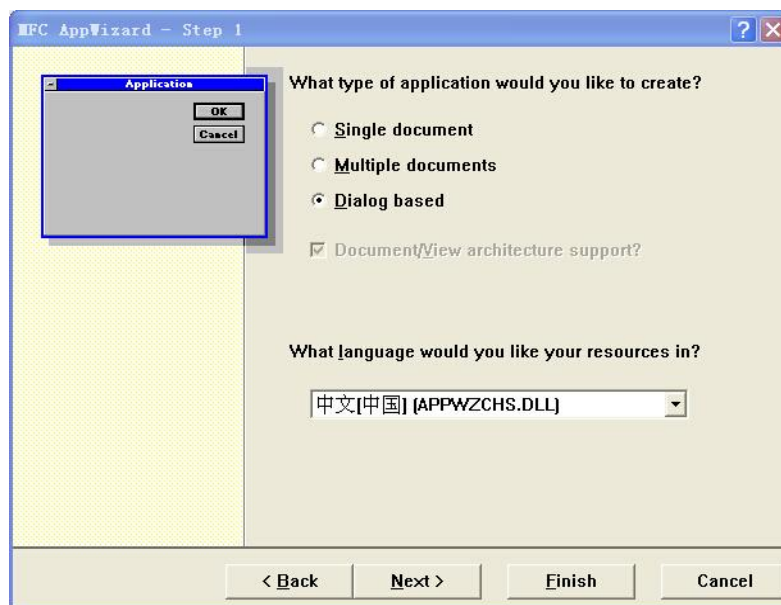


图 3.2MFC Step2

3) 单击[Finish]按钮, 出现如图 3.3 所示对话框, 单击[OK]按钮, 创建完成一个新工程: Sample。

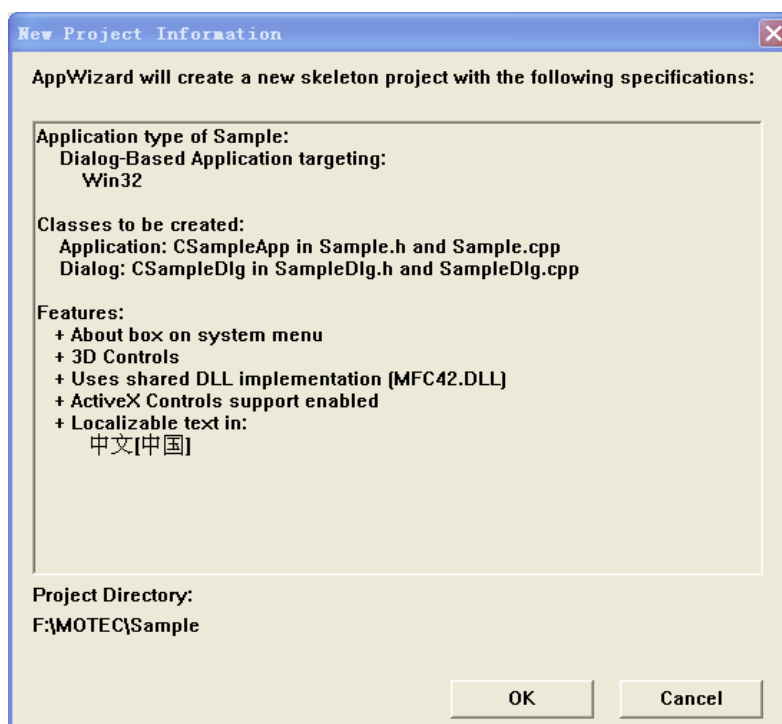


图 3.3MFC Step3

4) 完成注册 MotecDriverCmd 控件后, 插入 MotecDriverCmd 控件的方法是, 选择[Project\Add To Project\Components and Control...]菜单项, 将显示如图 3.4 所示的对话框。

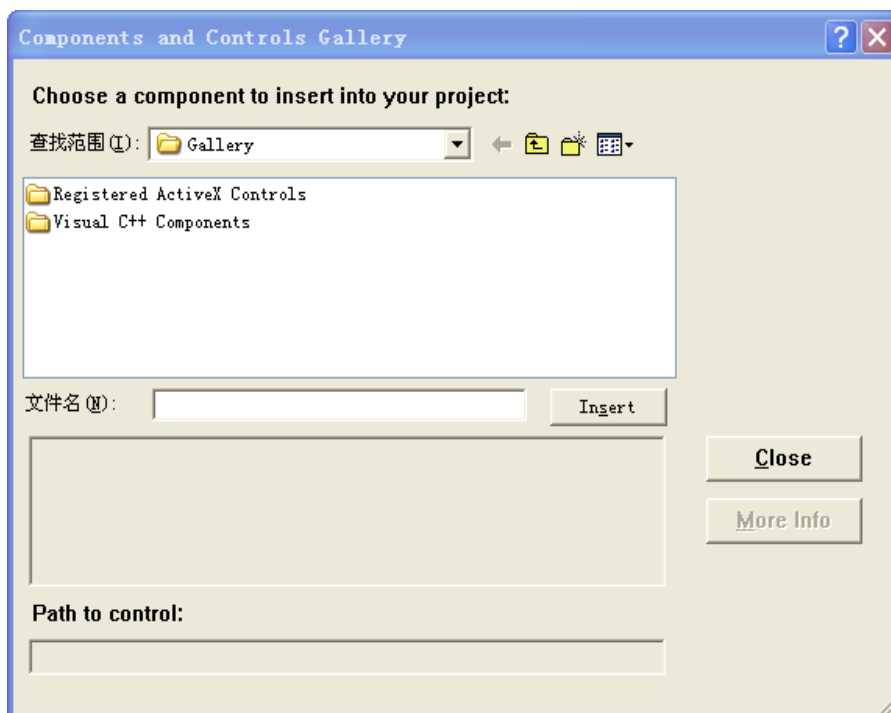


图 3.4 添加控件 Step1

5) 在此对话框中，双击“Registered ActiveX Controls”目录，并在此目录下找到CmotecDriverCmd 控件并选中，如图 3.5 所示。

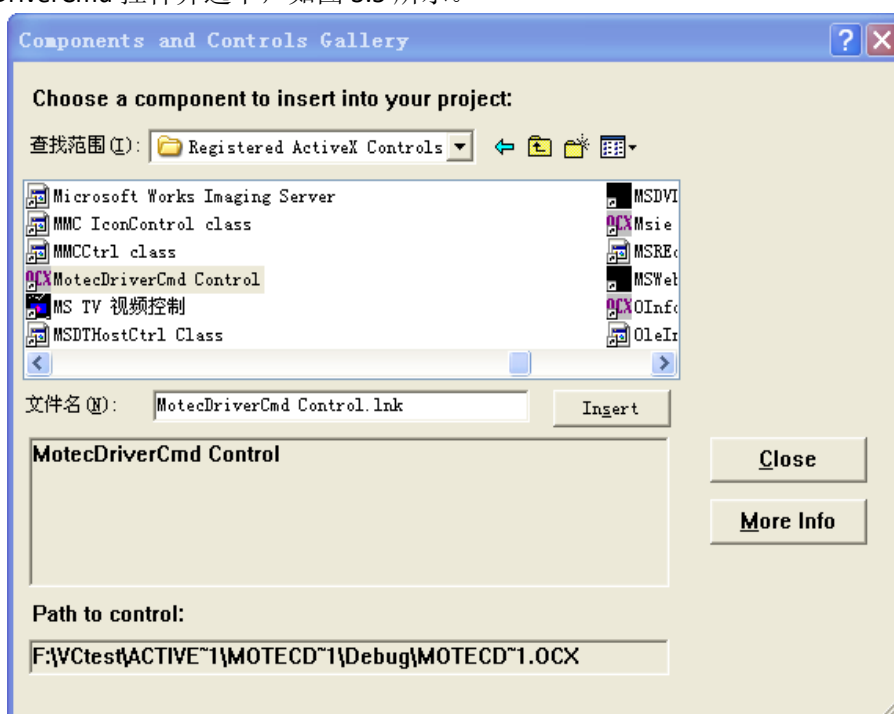


图 3.5 添加控件 Step2

6) 单击[insert]按钮，并单击如图 3.6 所示对话框上的[确认]按钮。

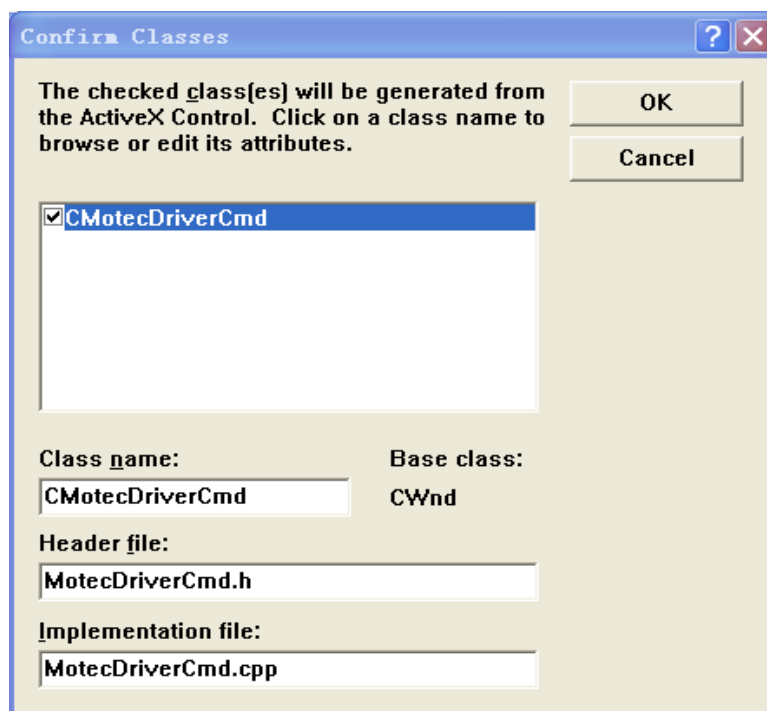


图 3.6 添加控件 Step3

插入控件后，会在该工程中为该控件生成一个类，这里就为 CmotecDriverCmd 控件生成了一个类，类名为 CmotecDriverCmd，同时，在工具箱上也增加了 CmotecDriverCmd 控件的图标，如图 3.7 所示，单击该图标，在对话框资源上拖放一块合适的区域就可以放置一个 CmotecDriverCmd 控件。

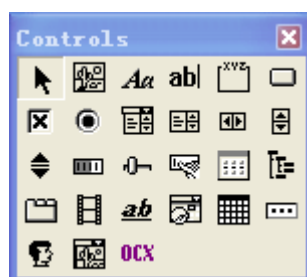


图 3.7 工具箱上添加的 CmotecDriverCmd 控件图标

7) 调用控件函数流程和部分代码如下：

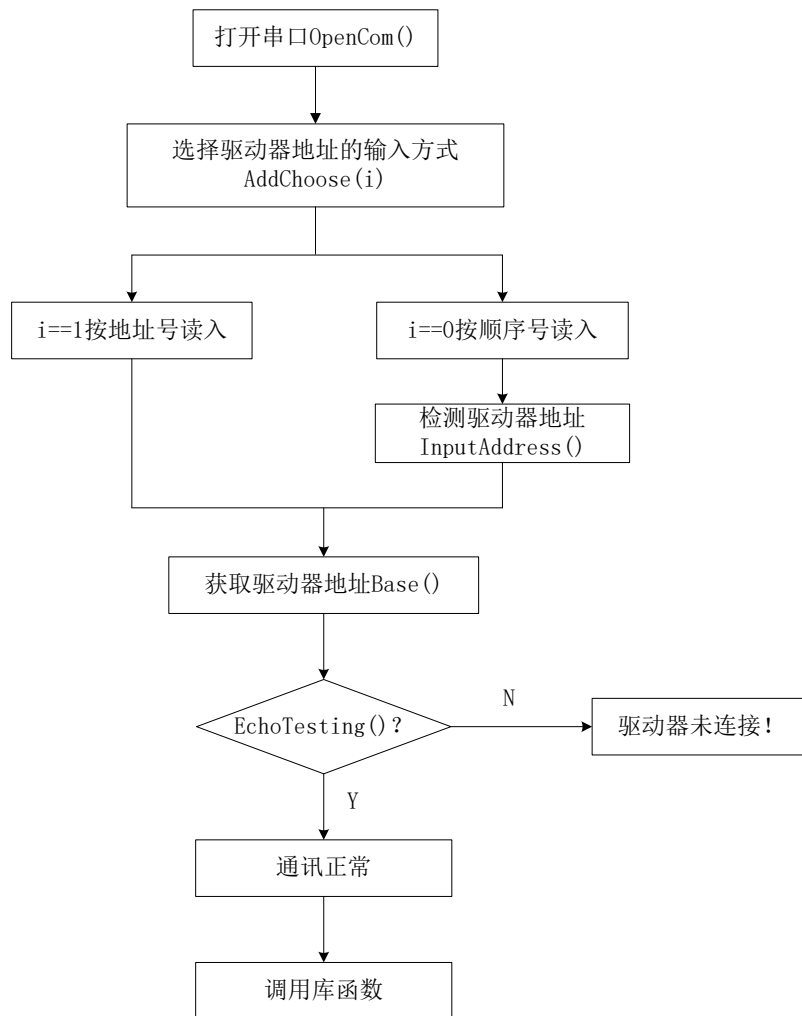


图 3.8 调用控件函数基本流程图

建立通讯时，首先调用 `OpenCom()` 函数打开串口，串口打开后，调用 `AddChoose(i)` 选择驱动器地址的输入方式，当 `i==1` 时，按照地址号方式读入地址；当 `i==0` 时，按照顺序号方式读入地址，通过调用 `InputAddress()` 检测连接到的驱动器地址，接着调用 `Base()` 函数获取驱动器地址，最后检测 `Echotesting()`，返回值为真表示通讯正常，否则驱动器未连接上。通讯连接成功后，就可以按照库函数使用说明使用库函数了。

注：驱动器的地址有两种输入方式：顺序号和地址号。若原有 3 个驱动器，地址为 2、7、8，则对应序号为 0、1、2，此序号根据地址排序。

代码举例：

```
m_cMotecDriverCmd.OpenCom(3,115200,8,0,0);//打开串口,3 为端口号, 115200 为波特率  
m_cMotecDriverCmd.AddChoose(0);//选择顺序号读入地址  
m_cMotecDriverCmd.InputAddress(&nrcpar);//nrcpar 为连接上的驱动器台数, 并检测驱动器地址  
m_cMotecDriverCmd.Base(0);//输入 0 代表选择顺序号为 0 的驱动器  
if(m_cMotecDriverCmd.EchoTesting()) //测试通讯是否正常
```

```
{
    nRecvStatus = m_cMotecDriverCmd.GetParameter (6,&nRpar);//获取参数表中的参数序号 6 对
    应的参数值，RecvStatus 为接收状态
}
```

4. 函数描述

4.1 IsOpenCom

函数：BOOL IsOpenCom(void)			
功能：	判断端口是否打开		
参数	没有参数		
返回值：	0- 端口未打开； 1- 端口打开；		

4.2 OpenCom

函数：BOOL OpenCom (int ComNo,DWORD Baud,BYTE Data,BYTE Parity,BYTE Stop)			
功能：	打开串口		
参数	参数类型	参数类型描述	参数内容
输入参数 1：	int	32 位有符号数	端口号
输入参数 2：	DWORD	32 位无符号数	波特率
输入参数 3：	BYTE	8 位无符号数	数据位
输入参数 4：	BYTE	8 位无符号数	校验位 0:无校验 1:奇校验 2: 偶校验
输入参数 5：	BYTE	8 位无符号数	停止位 0: 1 个停止位 1: 1.5 个停止位 2: 2 个停止位
返回值：	0- 失败； 1- 成功；		

注释：MOTEC 系列驱动器串口设置为 8 个数据位，一个停止位，无校验，如串口为 1.波特率为 115200，则 OpenCom(1,115200,8,0,0)

4.3 CloseCom

函数：void CloseCom (void)			
功能：	关闭串口		
参数	没有参数		
返回值：	没有返回值		

4.4. SetTimeCont

函数: BOOL SetTimeCont (DWORD rt,DWORD rm,DWORD rc,DWORD wm,DWORD wc)			
功能:	设置时间常数 单位: 毫秒(ms)		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	DWORD rt	32 位无符号数	读时间间隔超时
输入参数 2:	DWORD rm	32 位无符号数	读时间系数
输入参数 3:	DWORD rc	32 位无符号数	读时间常量
输入参数 4:	DWORD wm	32 位无符号数	写时间系数
输入参数 5:	DWORD wc	32 位无符号数	写时间常量
返回值:	0- 失败; 1- 成功;		

注: 读时间间隔超时是指以毫秒为单位设置通信线路上两个字符到达之间最大时间间隔; 读总超时时间= $rm \times$ 要求读的字节数 + rc ; 写总超时时间= $wm \times$ 要求写的字节数 + wc ; 当串口打开后, 再设置时间常数; 若未设置时间常数, 默认情况下: $rt=100$, $rm=5$, $rc=20$, $wm=5$, $wc=20$ 。

4.5 AddChoose

函数: BOOL AddChoose(BOOL adChoose)			
功能:	地址标志		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	BOOL adChoose	布尔型	写入地址标志的参数
备注:	0- 驱动器顺序号; 1- 驱动器地址;		
返回值:	0- 驱动器顺序号; 1- 驱动器地址;		

注释: 和 Base(ushort index)配合使用, AddChoose(0)表示当使用 Base(ushort index)中的 index 代表连接的驱动器的序号; AddChoose(1)表示当使用 Base(ushort index)中的 index 代表连接的驱动器的真实地址

4.6 InputAddress

函数: BOOL InputAddress (ushort driverNum)			
功能:	获得连接上的 MOTEC 驱动器台数		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort driverNum	16 位无符号数	驱动器的台数
返回值:	0- 没有检测到驱动器; 1- 检测到驱动器;		

4.7 Base

函数: BOOL Base (ushort index)	
功能:	设定需要通讯的 MOTEC 驱动器地址

参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort index	16 位无符号数	写入驱动器的地址或序号
返回值:	0- 写入失败; 1- 写入成功;		

注释: 和 AddChoose (BOOL adChoose)配合使用, AddChoose(0)表示当使用 Base(ushort index) 中的 index 代表连接的驱动器的序号; AddChoose(1)表示当使用 Base(ushort index)中的 index 代表连接的驱动器的真实地址

举例: 如需要和地址为 2 的驱动器通讯, 则 AddChoose (1) ;Base(2);

4.8 EchoTesting

函数: BOOL EchoTesting (void)			
功能:	发送通讯测试指令		
参数	没有参数		
返回值:	0- 表示通讯不正常; 1- 表示通讯正常;		

打开串口 (OpenCom) 成功后, 必须使用 EchoTesting 函数, 确定驱动器的校验方式

4.9 SendCommand

函数: BOOL SendCommand (uchar command,ushort sParameter1, ushort sParameter2, ushort rParameter1, ushort rParameter2)			
功能:	发送命令, 并接收返回值		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	uchar command	8 位无符号数	发送命令
输入参数 2:	ushort sParameter1	16 位无符号数	发送参数 1
输入参数 3:	ushort sParameter2	16 位无符号数	发送参数 2
接收参数 4:	ushort rParameter1	16 位无符号数	接收参数 1
接收参数 5:	ushort rParameter2	16 位无符号数	接收参数 2
返回值:	0- 发送失败; 1- 发送成功;		

4.10 PutParameter

函数: int PutParameter(ushort parNumber, ushort parData)			
功能:	设置驱动器参数表参数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort parNumber	16 位无符号数	参数表中的参数序号
输入参数 2:	ushort parData	16 位无符号数	写入的参数值
返回值:	3-设置参数越界, 设置不成功; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.11 GetParameter

函数: int GetParameter(ushort parNumber, ushort parData)			
功能:	获取驱动器参数表参数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort parNumber	16 位无符号数	参数表中的参数序号
输出参数 2:	ushort parData	16 位无符号数	读取的参数值
返回值:	3-参数序号越界; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.12 SaveParameter

函数: int SaveParameter(void)			
功能:	将驱动器 RAM 中的参数表保存到 FLASH 中		
参数	没有参数		
返回值:	1-保存失败; 0-成功;		

4.13 GetSerialNumber

函数: int GetSerialNumber(uint serialNum)			
功能:	读取 32 位控制器序列号		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	uint serialNum	32 位无符号数	32 位控制器序列号
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.14 GetVersionInformation

函数: int GetVersionInformation (ushort hardwareVersion, ushort softwareVersion)			
功能:	读取控制器硬件和软件版本信息		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort hardwareVersion	16 位无符号数	驱动器硬件版本信息
输出参数 2:	ushort softwareVerion	16 位无符号数	驱动器软件版本信息
备注:	如版本信息为 10101, 则表示版本号为 1.01.01		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.15 GetEncoderCounterPerRound

函数: int GetEncoderCounterPerRound (uint encoderCountPerRound)			
---	--	--	--

功能:	获取编码器每转计数		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	uint encoderCountPerRound	32 位无符号数	电机编码器每转计数
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.16 PutEncoderCounterPerRound

函数: int PutEncoderCounterPerRound (uint encoderCountPerRound)			
功能:	设置编码器每转计数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	uint encoderCountPerRound	32 位无符号数	电机编码器每转计数
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.17 GetDigitalOutput

函数: int GetDigitalOutput (ushort digOutNum, ushort digOutStatus)			
功能:	获取数字输出口状态		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort digOutNum	16 位无符号数	输出口编号
输出参数 2:	ushort digOutStatus	16 位无符号数	输出口状态
备注	输出口编号从 0 到 6, 如果编号是 0, 则输出口状态参数中获取的数据是输出口 1 到 6 的状态, bit0 代表输出口 1, bit5 代表输出口 6 的状态。如果输出口编号是 1 到 6 的数字, 则输出口状态参数中获取的是对应输出口编号的状态。0 表示光耦断开, 1 表示光耦导通。		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.18 PutDigitalOutput

函数: int PutDigitalOutput (ushort digOutNum, ushort digOutStatus)			
功能:	设置数字输出口状态		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort digOutNum	16 位无符号数	输出口编号
输入参数 2:	ushort digOutStatus	16 位无符号数	输出口状态
备注	输出口编号从 0 到 6, 如果编号是 0, 则输出口状态参数中设置的数据是输出口 1 到 6 的状态, bit0 代表输出口 1, bit5 代表输出口 6 的状态。如果输出口编号是 1 到 6 的数字, 则输出口状态参数中设置的是对应输出口编号的状态。0 表示光耦断开, 1 表示光耦导通。		
返回值:	2-通讯超时, 驱动器没有返回信息;		

	1-有返回值，校验出错，数据不可信； 0-成功；
--	-----------------------------

4.19 GetDigitalInput

函数: int GetDigitalInput (ushort digInNum, ushort digInStatus)			
功能:	获取数字输入口状态		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort digInNum	16 位无符号数	输入口编号
输出参数 2:	ushort digInStatus	16 位无符号数	输入口状态
备注	输入口编号从 0 到 12，如果编号是 0，则输入口状态参数中设置的数据是输出口 1 到 6 的状态，bit0 代表输出口 1，bit11 代表输入口 12 的状态。如果输入口编号是 1 到 12 的数字，则输入口状态参数中设置的是对应输入口编号的状态。0 表示光耦断开，1 表示光耦导通。		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.20 GetAnalogInput

函数: int GetAnalogInput (ushort analogIn1, ushort analogIn2)			
功能:	获取模拟输入口状态		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort analogIn1	16 位无符号数	模拟输入 1 数值
输出参数 2:	ushort analogIn2	16 位无符号数	模拟输入 2 数值
备注	模拟输入口的数值从-1000 到+1000 代表电压-10VDC 到+10VDC；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.21 GetAnalogOutput

函数: int GetAnalogOutput (ushort analogOut1, ushort analogOut2)			
功能:	获取模拟输出口状态		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort analogOut1	16 位无符号数	模拟输出 1 数值
输出参数 2:	ushort analogOut2	16 位无符号数	模拟输出 2 数值
备注	模拟输出口的数值从-1000 到+1000 代表电压-10VDC 到+10VDC；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.22 PutAnalogOutput

函数: int PutAnalogOutput (ushort analogOut1, ushort analogOut2)			
功能:	设置模拟输出状态		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort analogIn1	16 位无符号数	模拟输入 1 数值
输入参数 2:	ushort analogIn2	16 位无符号数	模拟输入 2 数值
备注	模拟输入口的数值从-1000 到+1000 代表电压-10VDC 到+10VDC;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.23 GetMotorEnableStatus

函数: int GetMotorEnableStatus (ushort motorStatus)			
功能:	使能/释放电机		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort motorStatus	16 位无符号数	电机使能状态
备注	0 表示电机释放, 1 表示电机使能;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.24 PutMotorEnableStatus

函数: int PutMotorEnableStatus (ushort motorStatus)			
功能:	设置电机状态(使能/释放)		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort motorStatus	16 位无符号数	电机使能状态
备注	0 表示电机释放, 1 表示电机使能;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.25 GetOperationMode

函数: int GetOperationMode (ushort OpMode)			
功能:	获取驱动器操作模式		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort OpMode	16 位无符号数	驱动器操作模式
备注	0- 网络模式; 1- 脉冲/方向模式; 2- 模拟信号模式; 3- 独立可编程模式;		

返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；
------	--

4.26 PutOperationMode

函数: int PutOperationMode(ushort OpMode)			
功能:	设置驱动器操作模式		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort OpMode	16 位无符号数	驱动器操作模式
备注	0- 网络模式； 1- 脉冲/方向模式； 2- 模拟信号模式； 3- 独立可编程模式；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.27 GetControlMode

函数: int GetControlMode (ushort CoMode)			
功能:	获取驱动器控制模式		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort CoMode	16 位无符号数	驱动器控制模式
备注	0- 电流控制模式； 1- 速度控制模式； 2- 位置控制模式；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.28 PutControlMode

函数: int PutControlMode (ushort CoMode)			
功能:	设置驱动器控制模式		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort CoMode	16 位无符号数	驱动器控制模式
备注	0- 电流控制模式； 1- 速度控制模式； 2- 位置控制模式；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.29 GetSProfileParameter

函数: int GetSProfileParameter (ushort SMaxVel, ushort SMaxAcc, ushort SMaxJerk)			
功能:	获取 S 曲线轨迹参数		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort SMaxVel	16 位无符号数	S 曲线最大速度(单位 RPM)
输出参数 2:	ushort SMaxAcc	16 位无符号数	S 曲线最大加速度(单位 R/S^2)
输出参数 3:	ushort SMaxJerk	16 位无符号数	S 曲线最大加加速度(单位 R/S^3)
备注:	S 曲线和 T 曲线轨迹参数中的最大速度为同一个参数, 是参数表中第 X 号参数;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.30 PutSProfileParameter

函数: int PutSProfileParameter (ushort SMaxVel, ushort SMaxAcc, ushort SMaxJerk)			
功能:	设置 S 曲线轨迹参数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort SMaxVel	16 位无符号数	S 曲线最大速度(单位 RPM)
输入参数 2:	ushort SMaxAcc	16 位无符号数	S 曲线最大加速度(单位 R/S^2)
输入参数 3:	ushort SMaxJerk	16 位无符号数	S 曲线最大加加速度(单位 R/S^3)
备注:	S 曲线和 T 曲线轨迹参数中的最大速度为同一个参数, 是参数表中第 X 号参数;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.31 GetTProfileParameter

函数: int GetTProfileParameter (ushort TMaxVel, ushort TAcc, ushort TDcc, ushort TReVel)			
功能:	获取 T 曲线轨迹参数		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort TMaxVel	16 位无符号数	T 曲线最大速度(单位 RPM)
输出参数 2:	ushort TAcc	16 位无符号数	T 曲线加速度(单位 R/S^2)
输出参数 3:	ushort TDcc	16 位无符号数	T 曲线减速度(单位 R/S^2)
输出参数 4:	ushort TReVel	16 位无符号数	T 曲线最大反转速度(单位 RPM)
备注:	S 曲线和 T 曲线轨迹参数中的最大速度为同一个参数, 是参数表中第 X 号参数;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.32 PutTProfileParameter

函数: int PutTProfileParameter (ushort TMaxVel, ushort TAcc, ushort TDcc, ushort TReVel)			
功能:	设置 T 曲线轨迹参数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort TMaxVel	16 位无符号数	T 曲线最大速度(单位 RPM)
输入参数 2:	ushort TAcc	16 位无符号数	T 曲线加速度(单位 R/S^2)
输入参数 3:	ushort TDcc	16 位无符号数	T 曲线减速度(单位 R/S^2)
输入参数 4:	ushort TReVel	16 位无符号数	T 曲线最大反转速度(单位 RPM)
备注:	S 曲线和 T 曲线轨迹参数中的最大速度为同一个参数, 是参数表中第 X 号参数;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.33 MoveAbsolute

函数: int MoveAbsolute (int moveDestination)			
功能:	设置绝对运动目的位置		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	int moveDestination	32 位有符号数	运动目的位置
备注:	设置完运动距离后需要调用 Go 命令启动点到点位置运动;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.34 MoveRelative

函数: int MoveRelative (int moveDistant)			
功能:	设置相对运动的运动距离		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	int moveDistant	32 位有符号数	运动距离
备注:	设置完运动距离后需要调用 Go 命令启动点到点位置运动;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.35 Go

函数: int Go (void)			
功能:	开始运动		
参数	没有参数		
备注:	Go 指令需要和 moveAbsolute 或 moveRelative 指令一起使用;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信;		

	0-成功;
--	-------

4.36 Stop

函数: int Stop (void)			
功能:	停止运动		
参数	没有参数		
备注	停止运动的过程由停止运动减速度控制;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.37 QuickStop

函数: int QuickStop (void)			
功能:	紧急停止运动		
参数	没有参数		
备注	紧急停止运动的过程由紧急停止类型和紧急停止减速度控制;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.38 GetProfileType

函数: int GetProfileType (ushort profileType)			
功能:	获取点到点运动轨迹类型		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort profileType	16 位无符号数	轨迹类型
备注	0- T 曲线轨迹; 1- S 曲线轨迹;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.39 PutProfileType

函数: int PutProfileType (ushort profileType)			
功能:	设置点到点运动轨迹类型		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort profileType	16 位无符号数	轨迹类型
备注	0- T 曲线轨迹; 1- S 曲线轨迹;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信;		

	0-成功;
--	-------

4.40 GetStopDecelleration

函数: int GetStopDecelleration (ushort stopDcc, ushort quickStopDcc)			
功能:	获取停止运动减速度		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort stopDcc	16 位无符号数	停止运动减速度(单位 R/S ²)
输出参数 2:	ushort quickStopDcc	16 位无符号数	急停减速度(单位 R/S ²)
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.41 PutStopDecelleration

函数: int PutStopDecelleration (ushort stopDcc, ushort quickStopDcc)			
功能:	设置停止运动减速度		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort stopDcc	16 位无符号数	停止运动减速度(单位 R/S ²)
输入参数 2:	ushort quickStopDcc	16 位无符号数	急停减速度(单位 R/S ²)
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.42 GetStatusFlag

函数: int GetStatusFlag (ushort status)			
功能:	获取驱动器状态信息		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort status	16 位无符号数	驱动器状态标志位
备注			
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.43 GetPosition

函数: int GetPosition (int actualPosition, int setpointPosition)			
功能:	获取电机位置信息		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	int actualPosition	32 位有符号数	驱动器位置实际值(单位: 脉冲)
输出参数 2:	int setpointPosition	32 位有符号数	驱动器位置设定值(单位: 脉冲)
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信;		

	0-成功;
--	-------

4.44 GetVelocity

函数: int GetVelocity (float actualVelocity, float setpointVelocity)			
功能:	获取电机速度信息		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	floatactualVelocity	浮点数	驱动器速度实际值(单位:RPM)
输出参数 2:	floatsetpointVelocity	浮点数	驱动器速度设定值(单位:RPM)
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.45 GetCurrent

函数: int GetCurrent (float actualCurrent, float setpointCurrent)			
功能:	获取电机电流信息		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	longactualCurrent	整数	电机电流实际值(单位:毫安)
输出参数 2:	longsetpointCurrent	整数	电机电流设定值(单位:毫安)
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.46 ResetPosition

函数: int ResetPosition (void)			
功能:	电机位置归零		
参数	没有参数		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.47 PutVelocitySetpoint

函数: int PutVelocitySetpoint (float velocitySetpoint)			
功能:	在速度控制模式下, 设置速度设定值		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	floatvelocitySetpoint	浮点数	电机速度设定值(单位:RPM)
备注	电机速度设定值精确到小数点后面 1 位;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.48 PutCurrentSetpoint

函数: int PutCurrentSetpoint (short currentSetpoint)			
功能:	在电流控制模式下, 设置电流设定值		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	shortcurrentSetpoint	16 位有符号数	电机电流设定值(单位:mA)
返回值:	0-失败 1-成功		

4.49 TrigDataLog

函数: int TrigDataLog(void)			
功能:	触发数据记录, 根据所设定的数据记录参数, 从 0 开始记录数据;		
参数	无参数		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.50 GetDataLogChnA

函数: int GetDataLogChnA (ushort i, ushort j, ushort data1, ushort data2)			
功能:	获取通道 1 中所记录的数据		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort i	16 位无符号数	需要获取数据的位置编号
输入参数 2:	ushort j	16 位无符号数	需要获取数据的位置编号
输出参数 3:	ushort data1	16 位无符号数	对应 i 位置的数据
输出参数 4:	ushort data2	16 位无符号数	对应 j 位置的数据
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.51 GetDataLogChnB

函数: int GetDataLogChnB (ushort i, ushort j, ushort data1, ushort data2)			
功能:	获取通道 2 中所记录的数据		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort i	16 位无符号数	需要获取数据的位置编号
输入参数 2:	ushort j	16 位无符号数	需要获取数据的位置编号
输出参数 3:	ushort data1	16 位无符号数	对应 i 位置的数据
输出参数 4:	ushort data2	16 位无符号数	对应 j 位置的数据
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信;		

	0-成功;
--	-------

4.52 GetDataLogChnC

函数: int GetDataLogChnC (ushort i, ushort j, ushort data1, ushort data2)			
功能:	获取通道 3 中所记录的数据		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort i	16 位无符号数	需要获取数据的位置编号
输入参数 2:	ushort j	16 位无符号数	需要获取数据的位置编号
输出参数 3:	ushort data1	16 位无符号数	对应 i 位置的数据
输出参数 4:	ushort data2	16 位无符号数	对应 j 位置的数据
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.53 GetDataLogChnD

函数: int GetDataLogChnD (ushort i, ushort j, ushort data1, ushort data2)			
功能:	获取通道 4 中所记录的数据		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort i	16 位无符号数	需要获取数据的位置编号
输入参数 2:	ushort j	16 位无符号数	需要获取数据的位置编号
输出参数 3:	ushort data1	16 位无符号数	对应 i 位置的数据
输出参数 4:	ushort data2	16 位无符号数	对应 j 位置的数据
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.54 GetLogDataNumber

函数: int GetLogDataNumber (ushort logDataNumber)			
功能:	获取数据记录长度		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort logDataNumber	16 位无符号数	数据记录长度, 最大 2000
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.55 PutLogDataNumber

函数: int put_LogDataNumber (ushort logDataNumber)	
功能:	设置数据记录长度

参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort logDataNumber	16 位无符号数	数据记录长度, 最大 2000
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.56 PutDataLoggingSetting

函数: int PutDataLoggingSetting (ushort loggingEnable, ushort samplingTime, ushort trigger, ushort trigLevel, ushort trigMode, ushort posFactor)			
功能:	设置数据记录参数		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort loggingEnable	16 位无符号数	0- 数据记录无效; 1- 数据记录使能
输入参数 2:	ushort samplingTime	16 位无符号数	数据记录采样时间, 单位 0.1ms
输入参数 3:	ushort trigger	16 位无符号数	用于触发数据记录的变量, 0 到 19 分别表示数据记录通道里定义的变量;
输入参数 4:	ushort trigLevel	16 位无符号数	数据记录功能触发界限值
输入参数 5:	ushort trigMode	16 位无符号数	数据记录触发方式: 0-下降沿触发; 1-上升沿触发;
输入参数 6:	ushort posFactor	16 位无符号数	数据记录功能记录位置的位置因子, 数据记录的位置信息为: 位置信息 = 实际位置信息 × Pr.206/每圈脉冲数
备注	当为上升沿触发时, 实际值要高于输入参数 4 所设置值, 开始触发 当为下降沿触发时, 实际值要低于输入参数 4 所设置值, 开始触发		
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.57 GetDataLoggingSetting

函数: int GetDataLoggingSetting (ushort loggingEnable, ushort samplingTime, ushort trigger, ushort trigLevel, ushort trigMode, ushort posFactor)			
功能:	获取数据记录参数		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort loggingEnable	16 位无符号数	0- 数据记录无效; 1- 数据记录使能
输出参数 2:	ushort	16 位无符号数	数据记录采样时间, 单位 0.1ms

	samplingTime		
输出参数 3:	ushort trigger	16 位无符号数	用于触发数据记录的变量, 0 到 19 分别表示数据记录通道里定义的变量;
输出参数 4:	ushort trigLevel	16 位无符号数	数据记录功能触发界限值
输出参数 5:	ushort trigMode	16 位无符号数	数据记录触发方式: 0-下降沿触发; 1-上升沿触发;
输出参数 6:	ushort posFactor	16 位无符号数	数据记录功能记录位置的位置因子, 数据记录的位置信息为: 位置信息 = 实际位置信息 × Pr.206/每圈脉冲数
备注	当为上升沿触发时, 实际值要高于输入参数 4 所设置值, 开始触发 当为下降沿触发时, 实际值要低于输入参数 4 所设置值, 开始触发		
返回值:	3-位置数据超过长度; 2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.58 GetloggingCompleteFlag

函数: int GetloggingCompleteFlag(ushort flag)			
功能:	获取数据记录完成标志		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort flag	16 位无符号数	0- 数据记录尚未完成 1- 数据记录已经完成
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.59 PutBrakeAction

函数: int PutBrakeAction(ushort brakeStatus)			
功能:	启动/释放抱闸		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort brakeStatus	16 位无符号数	需要获取数据的位置编号
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.60 GetBrakeStatus

函数: int GetBrakeStatus(ushort brakeStatus)	
功能:	获取抱闸状态

参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort brakeStatus	16 位无符号数	0- 抱闸处于抱紧状态 1- 抱闸处于释放状态 2- 没有定义抱闸输出
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.61 ClearErrors

函数: int ClearErrors (void)			
功能:	清除控制器故障信息		
参数	无参数		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.62 GetError

函数: int GetError (uint errorCode)			
功能:	读取控制器的故障代码		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	uint errorCode	32 位无符号数	故障代码
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.63 GetErrorDescriptionID

函数: int GetErrorDescriptionID(uint errorCode, VARIANT errorDescriptionID)			
功能:	获取故障状态编号		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	uint errorCode	32 位无符号数	故障代码
输出参数 2:	VARIANT errorDescriptionID	double dblVal	故障状态编号
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.64 GetErrorDescription

函数: int GetErrorDescription(ushort errorDescriptionID, BSTR bstrVal errorDescription)			
功能:	根据故障编号获取故障信息		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	ushort errorDescriptionID	16 位无符号数	故障状态编号

输出参数 2:	BSTR bstrVal errorDescription	BSTR bstrVal	故障状态描述
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.65 SeekLimit

函数: int SeekLimit (void)			
功能:	启动电机运动，直至到达限位开关		
参数	无参数		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 6:		

4.66 SeekIndex

函数: int SeekIndex (void)			
功能:	启动电机寻找零位脉冲		
参数	无参数		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.67 Homming

函数: int Homming (void)			
功能:	启动电机回原点		
参数	无参数		
返回值:	3-位置数据超过长度； 2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.68 Jog

函数: int Jog (short jogVelocity)			
功能:	使电机按 Jog 方式运动		
参数	参数类型	参数类型描述	参数内容
输入参数 1:	short jogVelocity	16 位有符号数	运动速度(单位 RPM)
备注	jogVelocity 为有符号数，正值代表正向运动，负值代表反向运动；		
返回值:	2-通讯超时，驱动器没有返回信息； 1-有返回值，校验出错，数据不可信； 0-成功；		

4.69 ClearMotionFlags

函数: int ClearMotionFlags (void)			
功能:	清除运动完成和接近完成标志		
参数	参数类型	参数类型描述	参数内容
参数:	无参数		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.70 GetDoneFlag

函数: int GetDoneFlag (ushort doneFlag, ushort nearDoneFlag)			
功能:	获取运动完成和接近完成标志		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort doneFlag	16 位无符号数	0- 运动尚未完成 1- 运动完成
输出参数 2:	ushort nearDoneFlag	16 位无符号数	0- 运动尚未到达接近完成 1- 运动到达接近完成
备注	运动完成标志和运动接近完成标志只在位置控制模式下有效;		
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

4.71 EnableDoneFlag

函数: int EnableDoneFlag (ushort statusEnableDone, ushort statusEnableNearDone)			
功能:	使能/取消 运动完成标志和接近完成标志		
参数	参数类型	参数类型描述	参数内容
输出参数 1:	ushort statusEnableDone	16 位无符号数	0- 运动完成标志无效 1- 运动完成标志有效
输出参数 2:	Ushort statusEnableNearDone	16 位无符号数	0- 接近完成标志无效 1- 接近完成标志有效
返回值:	2-通讯超时, 驱动器没有返回信息; 1-有返回值, 校验出错, 数据不可信; 0-成功;		

5. 联系方式

Website: <http://www.motec365.com.cn>;

地址: 北京市通州区环科中路 17 号 11B;

服务热线: 010-56298855-666;

Email: motecSupport@sina.com;