MACC 系列控制器 DLL 开发说明

拟制	日期	版本
	2022-9-30	T1

修改记录：

修改人员	日期	版本	修改内容
李建权	2023.9.22	T1.1	增加获取固件版本函数和 517 回复码说明

目录

概述

规定VCC3500控制器与控制端的通讯方式、MACC DLL的使用方法

DLL 说明

控制器通讯方式：100M以太网全双工通讯

开发系统配置：64 位操作系统

开发语言：C++

使用步骤

步骤 1：准备 MACC.h、MACC.lib、MACC.dll；

步骤 2：在工程中引用头文件 MACC.h 和 MACC.lib，并把 MACC.dll 放到程序运行目录下；

步骤 3：在程序中使用 MACC_initial 与控制器建立连接；

步骤 4：开始其他操作；

DLL 方法概览

序号	名称	作用	备注
1	MACC_initial	与控制器建立连接,并初始化控制器参数	
2	MACC_close	断开与控制器的数据连接	
3	MACC_get_probe_status	读取探针模块状态字	
4	MACC_get_position	读取光栅尺位置	
5	MACC_get_motion_status	读取运动模块状态字	
6	MACC_get_axis_status	读各轴状态	
7	MACC_get_joystick_Msg	读取手操器信息	
8	MACC_go_home	回原点运动	
9	MACC_single_move_to	单轴点位运动	
10	MACC_XYZ_move_to	XYZ 直线插补到指定位置	
11	MACC_arc_move_xyz	空间圆弧运动	
12	MACC_set_retract_parameter	设置探测回退参数	
13	MACC_touch_move	探针采点运动指令	
14	MACC_get_probe_capture_position	读取探针锁存值	
15	MACC_jog_start	示教运动启动	
16	MACC_change_speed	示教运动变速	
17	MACC_jog_stop	停止示教运动	
18	MACC_stop	停止运动	
19	MACC_disable_joystick	禁止手操器控制	
20	MACC_enable_joystick	允许手操器控制	
21	MACC_enable_probe	使能探针	
22	MACC_disable_probe	忽略探针	
23	MACC_get_input	读通用输入口状态	
24	MACC_set_output	写通用输出口状态	
25	MACC_get_output	读通用输出口状态	
26	MACC_get_profile	读取调试软件设定的各轴最	

		大速度和加速度	
27	MACC_get_soft_limit	读取轴的软限位	
28	MACC_get_last_err	获取最后的错误信息	
29	MACC_clear_err	清除错误	
30	MACC_log_enable	使能日记功能	
31	MACC_get_serialno	获取控制器序列号	
32	MACC_compare_password	比较输入的密码信息	
33	MACC_set_probe_head_pos	测座转角命令	
34	MACC_get_probe_head_pos	获取当前测座角度	
35	MACC_machine_error_compensation	机械误差补偿使能	
36	MACC_temperature_compensation_enable	温度误差补偿使能	
37	MACC_get_part_temperature	获取工件温度	
38	MACC_get_firmware	获取固件版本	

返回值说明

所有的 DLL 方法的返回值都是 int 类型，具体含义如下表

返回值说明表

返回值	说明	备注
0	操作成功	
1	Flash 错误：NAND_STATUS_PASSED	
2	Flash 错误：NAND_STATUS_FAILED	
8	Flash 错误：NAND_STATUS_NOT_FOUND	
16	Flash 错误：NAND_STATUS_DEVBUSY	
31	网络连接异常	
32	Flash 错误：NAND_STATUS_DEVWRPROTECT	
64	Flash 错误：NAND_STATUS_WAITTIMEOUT	
128	Flash 错误：NAND_STATUS_NOT_FOUND	
256	Flash 错误：NAND_STATUS_READ_ECC_ERROR_CORRECTED	
512	Flash 错误：NAND_STATUS_READ_ECC_UNCORRECTABLE_ERROR	
513	Flash 错误：地址错误	
514	Flash 错误：数据长度错误	
515	Flash 错误：坏块错误	
516	Flash 错误：校验错误	
517	控制器未解密	
4096	缓冲区队列为空	
4097	缓冲区队列已满	
4098	缓冲区队列忙碌	
4099	缓冲区队列空闲	
4100	缓冲区队列操作成功	
4101	缓冲区队列操作错误	
4102	网络通信数据长度错误	
4103	网络通信数据长度超限	

4104	网络通信超时	
4105	网络通信数据校验错误	
4106	网络通信通讯帧头错误	
4110	任务线程操作失败	
4115	该命令不支持,请升级控制器固件版本	
4131	机器坐标系空闲状态	
4133	机器坐标系处于运动状态	
4135	机器坐标系任务已经执行完成	
4352	机器轴空闲状态	
4354	机器轴处于运动状态	
4356	该轴不存在	
4357	轴未使能	
4358	控制模式错误	
4359	正限位报警	
4360	负限位报警	
4361	伺服未使能	
4362	目标位置距离太小	
4368	位置跟随超差报警	
4372	驱动器报警	
4373	软限位报警	
4384	回零运动运行中	
4385	回零运动执行完成	
4386	回零运动负限位搜索错误	
4387	回零运动正限位搜索错误	
4388	回零运动原点信号异常	
4389	回零运动索引信号异常	
4390	回零运动轴未使能	
4391	机器未回零	
4392	回零运动偏移错误	
4393	回零运动捕获成功	
4394	回零运动捕获成功且运动结束	
4416	急停输入有效	
4417	编码器 A 相报警	
4418	编码器 B 相报警	
4419	编码器 C 相报警	
4420	限位输入已有效，请先移动机器脱离限位	
4421	正限位未使能	
4422	负限位未使能	
4432	探针已触发	
4433	探针异常触发	
4435	测量运动搜索失败	
4436	自动探测回退结束后未到达目标位置	
4437	自动探测回退结束后探针未复位	

4448	不支持该参数,请升级控制器固件版本	
4449	参数写入 FPGA 失败	
4464	DLL 参数错误，不能为 0 值	
4465	DLL FIFO 错误	
4466	DLL FIFO 缓存已满错误	
4467	DLL FIFO 缓存已空错误	
4468	DLL FIFO 缓存可执行帧错误	
4480	正向软限位报警	
4481	负向软限位报警	
4512	圆弧半径不能为 0	
4513	圆弧旋转方向不匹配	
4514	圆弧两点距离大于直径	
4515	圆弧两点不在同一平面上	
4516	圆弧平面取值不匹配	
4517	圆弧两点到圆心距离不相等，圆弧错误	
4518	圆弧弦高误差不匹配	
4519	空间圆弧取值不匹配，为一条线	
4520	圆弧计算圆心角错误	

DLL 方法解析

1. **int MACC_initial(void)**

函数功能：

该函数试图与控制器建立通讯连接， 如果通讯建立， 将初始化控制器。如果返回值非0，请查返回值说明表获得返回值含义。

2. **int MACC_close(void)**

函数功能：

断开与控制器的数据连接等操作。

3. **int MACC_get_probe_status(unsigned int *status)**

函数功能：

读取探针模块状态字；该函数用来检测探针数据锁存标记等。

参数说明：

计数模块状态字表

位编号	位定义	取值说明	Word
Bit0	探针锁存标记	0：锁存标记未触发；1：锁存标记已触发	0x0001
Bit1	探针状态	0：探针未触发；1：探针已触发	0x0002
Bit2	探针使能状态	0：探针未使能；1：探针已使能	0x0004
Bit3	测座轴驱动报警	0：正常；1：报警	0x0008
Bit4	测座未安装	0：正常；1：报警	0x0010
Bit5	测头未安装	0：正常；1：报警	0x0020
Bit6	测座运动完成状态	0：未完成；1：已完成	0x0040

Bit7	测座锁紧异常	0：正常；1：报警	0x0080
Bit8	与 PHC 测座控制器通讯状态	0：正常；1：报警	0x0100
Bit9	测头故障报警(SC80)	0：测头正常；1：测头碰撞	0x0200
Bit10	测头压深超限报警(SC80 硬件报警)	0：测头正常；1：测头压深超限	0x0400
Bit11	探针释放到位状态(SC80)	0：未到位；1：释放到位	0x0800
Bit12	探针夹持到位状态(SC80)	0：未到位；1：夹持到位	0x1000
Bit13	测头压深报警（SC80 软设置报警）	0：测头正常；1：测头压深报警	0x2000

另请参考： MACC_get_probe_capture_position

4. int MACC_get_position(double *scale);

函数功能：

读取各轴的机械坐标；控制器电源开启后该位置无意义，成功回原点后该位置为相对原点位置。

参数说明：

scale所指向的数组长度不得小于8，单位为毫米，数组索引标号0-7分别对应X，Y，Z，R等8个轴的数据。

5. int MACC_get_motion_status(unsigned int *status)

函数功能：

读取运动模块状态字。运动控制的重点在于运动模块状态字的理解,运动模块状态字含义丰富，是运动控制状态之窗，可以说，对运动控制理解和灵活运用的熟练程度与软件开发质量、控制精度以及控制器使用效果直接相关。如果测量软件有运行日志记录功能，在异常出现时该状态字应该被记录，作为硬件异常诊断的有力依据。如果不具备，应保留一个菜单在必要时弹出该状态字内容。

参数说明：

运动模块状态字表

位编号	位定义	取值说明	16 进制值
Bit0	所有轴停止标记	0-有轴尚处于运动中， 1-所有轴运动均已停止；该位由 bit4-bit7“逻辑与”操作而来	0x0001
Bit1	出错标记	0-无出错信息或出错信息已读取， 1-有新的出错信息发生	0x0002
Bit2	保留	保留	
Bit3	操纵杆使能状态	1：操纵杆已使能， 0-操纵杆已被禁止	0x0008
Bit4	X 轴状态	1：停止状态， 0－运动状态	0x0010
Bit5	Y 轴状态	1：停止状态， 0－运动状态	0x0020
Bit6	Z 轴状态	1：停止状态， 0－运动状态	0x0040
Bit7	U 轴状态	1：停止状态， 0－运动状态	0x0080

Bit8	测量模式	1： 3D 测量模式， 0-2D 测量模式	0x0100
Bit9	探针使能	1：探针已使能， 0-探针未使能	0x0200
Bit10	紧急停止开关状态	1－紧急停止开关已触发， 0－未触发	0x0400
Bit11	自动探测异常标记	1：自动探测异常， 0：自动探测无异常	0x0800
Bit12	操纵杆采点异常标记	1：操纵杆采点异常， 0：操纵杆采点无异常	0x1000
Bit13	探针状态	1－安装就绪， 0－探针未安装或者探针已触发	0x2000
Bit14	探针异常触发标志	1－探针使能后，在非探针采点运动（点位运动、插补运动、 jog 运动）时触发，当探针复位后该标记位自动清 0 0－未出现异常触发现象	0x4000
Bit15	采点运动标记	0- 无探针采点运动或探针采点运动已结束 1- 探针采点运动正在进行中	0x8000

Bit0: 用来判断直线插补运动是否完成，或者用来检测所有轴是否均已停止运动，它由bit4-bit7逻辑与操作而来。

Bit1: 出错标记，当API函数操作错误、系统错误（见MACC_get_last_err所获得出错代码信息）时，该标记位置1。调用MACC_get_last_err读取出错代码后该标记为清0。

BIT3：操纵杆使能状态：成功使能后该标记为1，操纵杆禁止后该标记被清0。

BIT4： X轴停止状态标记位，当X轴停止运动（包括JOG运动change_speed到0）时该标记为1，运动时为0。

BIT5~7： Y/Z/U轴停止状态标记位，作用与X轴相同。

BIT9：探针使能标记，探针被使能为1，探针被禁止为0。

BIT10：紧急停止开关状态，紧急停止开关触发后为1，复位后为0。

BIT11：自动探测异常标记，成功调用MACC_touch_move函数后该标记清0，机台将朝向工件运动，在接触到工件后减速、并回退到调用MACC_touch_move函数的位置，一个完整的探针测量运动成功结束，该标记保持为0；当异常发生时，采点运动结束后该标记为1，异常现象和原因请参考MACC_get_last_err。异常处理：用操纵杆或JOG运动将探针移到安全的区域，排除问题后再进行测量。MACC_clear_motion_err可清除该标记。

BIT12：操纵杆采点异常标记，正常情况下，使用操纵杆进行探针采点编程时，探针碰撞后将自动回退，当异常发生时该标记被置1。异常处理：用操纵杆或JOG运动将探针移到安全的区域，排除问题后再进行测量。MACC_clear_motion_err可清除该标记。

BIT13：探针状态，当探针未安装、探针触发时该标记为0，探针已安装且就绪状态时该标记为1，实时的反映了探针当前的状态。

BIT14：探针异常触发标志，在进行单轴、多轴点到点运动、 JOG运动时，如果探针触发则被控制器判定为“探针异常触发”，该标志位置1，这些运动被立即停止，探针复位后该标志位自动清0。MACC_clear_motion_err也可清除该标记。探针异常触发后如果探针使能，应禁止探针后下达运动指令使探针离开工件。

BIT15：自动探测标志，成功调用MACC_touch_move函数后该位置1，机台自动探测；采点成功并回退到调用MACC_touch_move的位置后，该位清0；异常出现时所有轴立即停止运动，新的touch_move指令成功下达时该标志清0。见BIT11说明

6. int MACC_get_axis_status(unsigned int *status)

函数功能：

读取各轴的轴状态字

参数说明：

无论控制器是否具备U或其他轴， status所指向的数组长度均不得小于8。

status[0]： X轴状态字

status[1]： Y轴状态字

status[2]： Z轴状态字

status[n]： N轴状态字

依此类推，最多支持8轴状态

轴状态字表

位编号	位定义	取值说明	16 进制值
Bit0-4	保留	保留	
Bit5	正限位状态	0-未触发,1-已触发	0x0020
Bit6	负限位状态	0-未触发,1-已触发	0x0040
Bit7	原点开关状态	0-未触发,1-已触发	0x0080
Bit8/Bit9	保留	保留	
Bit10	运动状态	0-运动中， 1-已停止	0x0400
Bit11	回原点标记	0-未回原点或者回原点失败 ，1-已成功回原点	0x0800
Bit12	回零运动状态	0-不处于回零运动，1-回零运动中	0x1000
Bit13	保留	保留	
Bit14	找 RI 失败	0-无异常， 1－找光学尺 RI 失败	0x4000
Bit15	跟随误差超差	0-未出现跟随误差超差， 1－跟随误差超过设定值	0x8000

该寄存器对运动控制器的理解和使用比较重要，各 BIT 介绍如下：

Bit0-4、 bit8-9、 bit12-13：保留未用，读为 0。

Bit5 / Bit6 / Bit7：该位实时反映正限位 / 负限位 / 原点开关状态； 0 表示未触发，1 表示已触发。

Bit10：停止状态为 1，运动状态为 0。 轴的运动控制命令下达后该位立即清 0，如果运动距离非常短（为 1 个运动当量或者为 0），运动状态也会保持为 0 若干微秒，尽管应用软件未必能捕捉到。因此应用软件发送移动距离不为 0 的运动命令后，最短时间内查询运动状态，即使移动还没有发生，但读取的运动状态也一定为“运动”状态。

Bit11： 0－未回原点或者回原点失败； 1－已成功回原点。控制器上电后为 0；MACC_go_home 函数调用后首先清该位为 0，然后执行找原点运动。运动结束后应测试该位以确定找原点是否成功。MACC_initial 和 MACC_close 函数对该位无影响，因此测量软件启动、关闭对该标记位无影响。若软件启动 MACC_initial 成功后，调用该函数检测到 bit11 为 1，表明机器已回原点，可以立即投入使用， 缩短软件从启动到投入正式使用的时间，提高仪器工作效率。

Bit14：控制器上电后为 0，若当前轴启动回原点，机台在正负限位开关之间往返一次仍然未找到 RI，则该标记被置位，同时 Last_err 将记录错误代码。

Bit15：控制器上电后为 0，当前轴点位运动时跟随误差超过设定值， 当前轴将立即停止， 同时该位置 1，在执行新的任务前需要调用 MACC_clear_err()清除错误。

7. int MACC_get_joystick_Msg(unsigned int *message)

函数功能：

读取手操器按钮状态信息。

参数说明：

手操器状态表

位编号	按键定义	操作说明	对应 16 进制数
0	KEY_GOTO	拐点设置	0x0001
1	KEY_DELETE_POINT	删除探针采点	0x0002
2	KEY_PROBE_ENABLE	探针忽略/使能	0x0004
3	KEY_F1	F1 功能键	0x0008
4	KEY_F2	F2 功能键	0x0010
5	KEY_F3	F3 功能键	0x0020

注:即使摇杆锁定状态，应用软件也能获取到键盘状态的改变。

8. int MACC_go_home(int axis)

函数功能：

令指定轴回原点

参数说明：

axis：轴编号， 0-X轴， 1-Y轴， 2-Z轴，依次类推，最大数值为7

9. int MACC_single_move_to(int axis,double speed,double target_pos)

函数功能：

命令指定轴以绝对坐标方式定位到位置为target_pos。

Single_move_to成功执行的条件：

1. 当前轴已处于停止状态
2. 如果探针已使能，探针应该处于未触发状态
3. 如果手操器已使能， 手操器操纵杆应处于未扳动状态

参数说明：

axis： 轴编号， 0-X轴， 1-Y轴， 2-Z轴，依次类推，最大数值为7；

speed：单位为mm / s；

target_pos： 运动到终点时机械坐标位置， 绝对坐标方式， 单位为： mm。

10. int MACC_XYZ_move_to(double speed,double *target_pos)

函数功能：

令 XYZ 轴按照直线插补运动规律运动到某处。

直线插补运动状态查询请参考 MACC_get_motion_status，运动模块状态字

BIT0 最适合用来判断直线插补运动是否完毕， BIT14 作为辅助判断标记。

XYZ_move_to 成功执行的条件：

1. 当前轴已处于停止状态
2. 如果探针已使能，探针应该处于未触发状态
3. 如果手操器已使能， 手操器操纵杆应处于未扳动状态

参数说明:

speed:直线插补运动矢量速度。当计算出来某一轴的分速度超过该轴的最大速度值时，各轴的运动速度将等比例降低，确保各轴安全运动，且运动轨迹的精度保持不丢失。

举例说明： Z 轴运动速度一般远低于 X/Y 轴， X/Y 轴运动距离比 Z 轴大时，仪器以比大的矢量速度运动，确保效率。当 Z 轴运动距离比较大时，Z 轴的理论分速度超过了 Z 轴最

大速度，各轴运动速度将自动降低，确保 Z 轴能可靠运行。

target_pos[0]:X 轴终点绝对位置

target_pos[1]:Y 轴终点绝对位置

target_pos[2]:Z 轴终点绝对位置

11. int MACC_arc_move_xyz(double x_start,double y_start, double z_start, double x_mid,double y_mid, double z_mid, double x_end,double y_end, double z_end, double synVel, unsigned int circleType)

函数功能：

空间三点圆弧运动

参数说明：

x_start、y_start、z_start 为圆弧起点坐标，

x_mid、y_mid、z_mid 为圆弧中间任意点坐标，

x_end、x_end、x_end 为圆弧终点坐标；

synVel 为圆弧运动线速度

circleType 表示曲线类型，0 为圆弧，1 为整圆

12. int MACC_set_retract_parameter(double retract_speed,double retract_distance)

函数功能：

设置探针采点回退参数。该参数已通过控制器调试软件设置并保存于控制器之内。测量软件可覆盖此设置，但测量软件的设置不会被保存，电源重新启动将恢复调试软件设定值。

参数说明：

retract_speed： 探针采点后回退速度，单位 mm/s

retract_distance：回退距离，单位 mm。

13. int MACC_touch_move(double touch_speed, double search_radius, double *target_pos, double *ijk)

函数功能：

启动自动探测采点运动。探针采点运动用于测量软件执行已编好的测量程序， 获取一个碰撞点坐标。使用 MACC_XYZ_move_to 定位到碰撞点附近，然后调用该函数启动采点运动。

成功执行的条件：

1． 探针已经使能并且已就绪

2． 所有轴处于停止状态。

3． 操纵杆已禁止。

该函数成功调用后发生如下事情：

1. 运动模块状态字 Bit11（ 自动探测失败标记） 设置为 0

2. 运动模块状态字 bit15 设置为 1

3. 控制器使机器从调用该函数的地方开始，以 touch_speed 沿搜索方向矢量运动，在此过程中接触到工件后，探针将减速停止然后按照原方向回退 MACC_set_retract_parameter 设定的回退距离，采点运动成功结束。如果到达理论碰撞点后探针没有接触到工件，将继续向前运动（搜索工件），如果超出“理论碰撞点”的距离大于“搜索距离”，机台将停止运动，采点运动结束；运动模块状态字 Bit11 将被置 1，同时 Last_err 代码被设置。

无论成功结束还是异常结束， 运动模块状态字 bit15 都会清零。

参数说明:

touch_speed：采点速度，单位 mm/s，通常为 0.5-20mm/s

search_distance：搜索距离，由于待测工件尺寸偏差和夹具精度原因，自动测量时理论碰撞点总与实际碰撞点存在偏差，因此探针需要在理论碰撞点之后一定范围内（搜索距离）搜索工件。

target_pos[0]： X 轴理论碰撞点坐标，单位 mm

target_pos[1]： Y 轴理论碰撞点坐标，单位 mm

target_pos[2]： Z 轴理论碰撞点坐标，单位 mm

ijk[0]： 搜索方向矢量 i

ijk[1]： 搜索方向矢量 j

ijk[2]： 搜索方向矢量 k

14. int MACC_get_probe_capture_position(double *pos_x,double *pos_y,double *pos_z,double *pos_r,double *pos_i,double *pos_j,double *pos_k)

函数功能：

读取探针锁存的 X / Y / Z / R 坐标和探测点的表面法向量 ijk。 当 get_probe_status 检测到计数模块状态字 bit0（探针锁存标记）为 1 时，说明探针有更新新的锁存数据，探针忽略状态时被触发，不会锁存数据。

参数说明:

pos_x： X 轴碰撞点坐标，单位 mm

pos_y： Y 轴碰撞点坐标，单位 mm

pos_z： Z 轴碰撞点坐标，单位 mm

pos_r： R 轴碰撞点坐标，单位 mm

pos_i： 表面法向量 i

pos_j： 表面法向量 j

pos_k： 表面法向量 k

15. int MACC_jog_start(int axis)

函数功能：

使指定轴开始 JOG 模式运动，该函数指定了本次启动的 jog 运动所允许的最大速度。该函数调用之后 ，MACC_jog_stop 调用之前 ，可连续调用 MACC_change_speed 改变运动 速度和方向。Jog 功能特别适用于使用鼠标控制机台运动的情形。

假如首次启动 JOG 运动，或者先前启动的 JOG 运动已被 MACC_ stop 停止， 需要再次启动 JOG 运动时，必须具备如下条件：

1． 操纵杆已被 disable 或者操纵杆已使能但没有被扳动；

2． 没有一个轴处于单轴、多轴点到点运动状态；

3． 如果探针已被使能，探针应该处于未触发状态；

如果某轴已经处于 jog 运动，其它轴当然具备 jog 运动条件。

如果探针已使能，且调用该函数时探针处于触发状态，该操作将被忽略并返回“ 操作出错”。

参数说明：

axis： 轴编号， 0-X 轴， 1-Y 轴， 2-Z 轴,3-U 轴，依次类推，最大数值为 7；

16. int MACC_change_speed(int axis,double curnt_speed)

函数功能：

jog_start 函数调用之后、 MACC_ stop 调用之前，连续不断调用该函数以达到改变 jog 运动速度和方向的目的。

该函数执行的条件：指定轴已使用 jog_start 设置好并启动 jog 运动。

参数说明:

axis： 轴编号， 0-X 轴， 1-Y 轴， 2-Z 轴,3-U 轴，依次类推，最大数值为 7；

curnt_speed：目标速度。 该参数为正时，机台朝正向运动，为负时朝负向运动， 该参数数值和方向突然变化时，控制器将会使速度按照梯形曲线加减速变化； 该参数绝对值大于 jog_speed_max 时， 速度限制为 jog_speed_max。调用后轴移动速度将一直保持，直到限位被触发。如果 MACC_change_speed(n,0)，轴将处于“速度为 0 的 jog 状态”， 除非使用 MACC_jog_stop 命令。

17. int MACC_jog_stop(void)

函数功能：

令所有处于 JOG 状态的轴减速停止。该函数任何时候调用、重复调用都不会报错。该函数一定要与 jog_start 配套使用，比如：在鼠标左键按下时调用 MACC_jog_start，在鼠标左键松开时调用 MACC_jog_stop. Change_speed 到 0 可以使轴处于停止运转状态，但轴仍然处于“速度为 0 的 JOG”运动状态，使用 jog_stop 之前，发送运动命令将会拒绝执行并以出错返回。

18. int MACC_stop(int mode)

函数功能：

令所有轴停止运动， 通常用来中止一个已经下达的任何运动指令，但如果是 JOG 运动，还需要使用 MACC_jog_stop 命令。

参数说明:

mode： 停止模式：

0－减速停止，如果正在高速运动，该函数调用后机台将滑行一段之后停止；

1－立即停止，该函数返回后该轴便处于停止状态，由于立即停止会导致产生较大冲击，可能导致机台损坏，如果机台比较大运行速度比较快， 非紧急情况不要使用立即停止。

19. int MACC_disable_joystick(void)

函数功能：

通知控制器， 无论任何轴处于任何状态， 扳动操纵杆都不要让机台产生运动。 该操作常用于：

1. 用户编制的自动测量程序启动前。

2. 自动对焦或其它不希望操作者通过操纵杆控制机台运动的情形。

无论操纵杆是否使能， MACC_disable_joystick 多次调用该函数无影响。

20. int MACC_enable_joystick(void)

函数功能：

通知控制器， 允许操纵杆控制机台运动。控制器连接 IMO 手操器时，可通过面板上的 Unlock 键解锁操纵盒，从而使能操纵杆。 该函数执行成功的条件：

1. 操纵盒已校正并且已连接到控制器。

2. 调用该函数时操纵盒处于未被扳动状态。

允许操纵盒控制后，操纵盒上高低速切换按钮才会被响应。其他按钮不受操纵杆是否使能

的限制。

操纵杆使能后， 如果所有轴空闲，扳动操纵杆， 机台移动且速度由操纵杆控制。所有轴停止后，可以执行点位运动、 JOG 运动、探针采点运动；如果这些运动正在进行，操纵杆将暂时屏蔽，直到所有轴停下来。为了避免测量程序自动运行时，操纵杆产生干扰（比如运动的停顿瞬间），自动运行启动前应该用 MACC_disable_joystick 禁止操纵杆， 在需要人工干预的地方或者测量程序运行完毕再使能操纵杆。扳动操纵杆移动机台，当探针触发时机台将自动减速、停止，并按照设定的回退速度和回退距离回退。另请参考:MACC_disable_joystick、 MACC_touch_move。

参数说明:

low_speed 和 high_speed 参数保留； 该速度由用户在“调试软件”界面设置。

21. int MACC_enable_probe(void)

函数功能：

使能探针。 探针使能与屏蔽在控制器中可以设置，并存储在控制器 flash 内。探针使能后，单轴点到点运动、多轴直线插补和圆弧插补、回原点运动以及 JOG 运动时触发探针控制器将立即停止。因此，在使用探针交换架或类似操作时，需暂时忽略探针，直到测头交换完毕再次使能探针。

参数说明:

无。

22. int MACC_disable_probe(void)

函数功能：

忽略探针。

参数说明:

无。

23. int MACC_get_input(unsigned int *input)

函数功能：

读取输入口电平状态。 MACC 控制器具备 4 个通用输入口。

参数说明：

输入口状态位定义表

位编号	位定义	取值说明	16 进制值
Bit0	通用输入 0	0－低电平， 1－高电平	0x0001
Bit 1	通用输入 1	0－低电平， 1－高电平	0x0002
Bit 2	通用输入 2	0－低电平， 1－高电平	0x0004
Bit 3	通用输入 3	0－低电平， 1－高电平	0x0008
Bit 4- Bit 31	保留		

24. int MACC_set_output(unsigned int output)

函数功能：

整体设置输出口状态。MACC 控制器具备 5 个通用输出口。

参数说明：

output 状态位定义表

位编号	位定义	取值说明	16 进制值
Bit 0	Out 0	0－输出导通，负载加电， 1－输出截止，负载断电	0x0001
Bit 1	Out 1	0－输出导通，负载加电， 1－输出截止，负载断电	0x0002
Bit 2	Out 2	0－输出导通，负载加电， 1－输出截止，负载断电	0x0004
Bit 3	Out 3	0－输出导通，负载加电， 1－输出截止，负载断电	0x0008
Bit 4	Out 4	0－输出导通，负载加电， 1－输出截止，负载断电	0x0010

25. int MACC_get_output(unsigned int *output)

函数功能：

读取输出口电平状态。 MACC 控制器具备 5 个通用输出口。

26. int MACC_get_profile(double *max_acc,double *max_speed)

函数功能：

读取各轴最大加速度和最大速度

参数说明：

max_acc和max_speed所指向的数组长度不得小于8。数组索引标号0-7分别对应X，Y，Z，R等8个轴的数据

max_acc数据的单位为mm/s^2，max_speed数据的单位为mm/s

27. int MACC_get_soft_limit(int axis,double *pos_limit,double *neg_limit)

函数功能：

获取指定轴正、负限位开关相对于原点的位置，即获取轴安全运动范围。

参数说明：

axis：轴编号，0-X 轴，1-Y 轴，2-Z 轴,3-U 轴；

pos_limit/neg_limit：正、负软件限位坐标位置，单位 mm

28. int MACC_get_last_err(unsigned int *err_code)

函数功能：

当 MACC_get_motion_status(unsigned int *status)读取到运动模块状态字 bit1 为 1（对应 16 进制数为 0x0002） 时，说明有出错现象产生且未被读取。该函数用于读取最后一次的出错信息代码。该函数调用后模块状态字 bit1 自动清 0。

参数说明：

err_code：错误信息代码，具体含义请对照“ 返回值说明表”。

29. int MACC_clear_err()

函数功能：

清除控制器目前出现的错误；

参数说明：

没有参数。

30. int MACC_log_enable(bool enableVal)

函数功能：

使能日志记录

参数说明：

enableVal 为 true--记录操作日志，false--不记录操作日志

31. int MACC_get_serialno(char* serialVal);

函数功能：

获取控制器序列号。

参数说明：

serialVal 表示用于存储返回的序列号信息的数组，长度为 32；

32. int MACC_compare_password(char* pwd, unsigned short len_pwd)

函数功能：

比较加密信息，成功则允许运动操作，否则禁止运动。

参数说明：

pwd 表示密码数组；

len_pwd 表示密码数组长度；

33. int MACC_set_probe_head_pos(double angle_a, double angle_b)

函数功能：

使测座转动至 A 角为 angle_a，B 角为 angle_b 的角度。该功能仅对自动测座起作用，如 CZ10T、PH10T 等。转动过程中，MACC_get_probe_status 获取值的 bit6 为 0，转角完成后 bit6 为 1。

参数说明：

err_code：错误信息代码，具体含义请对照“ 返回值说明表”。

34. int MACC_get_probe_head_pos(double *angle_a, double *angle_b)

函数功能：

获取测座当前的 A 角和 B 角；

参数说明：

angle_a 和 angle_b 分别表示接收返回值变量的指针。

35. int MACC_machine_error_compensation(bool enableVal, double *tool_offset)

函数功能：

使能机械误差补偿。在使用该功能之前需要在 MACC 的调试软件的误差补偿模块录入误差数据，否则即使使能了补偿功能也无效。

参数说明：

enableVal 为 true—使能补偿，false—忽略补偿

tool_offset 是长度为 3 的数组，表示探针球心的 X、Y、Z 三个方向上的偏置

36. int MACC_temperature_compensation_enable(bool enableVal);

函数功能：

使能/忽略温度补偿功能。

参数说明：

enableVal 为 true—使能补偿，false—忽略补偿

37. int MACC_get_part_temperature(double *tempVal)

函数功能：

获取工件的温度。使用该功能需要在控制器端接入温度传感器，并在 MACC 调试软件的温度监测模块录入温度通道匹配才能生效。

参数说明：

tempVal 表示接收返回值的指针；

38. int MACC_get_firmware(char* FVal, char* DVal)

函数功能：

获取控制器的固件版本。

参数说明：

FVal 表示 F 固件的版本号， 用 ASCII 解码；

DVal 表示 D 固件的版本号， 用 ASCII 解码；