OMRON

可编程多轴运动控制器

应用指南

激光应用篇

CK3M-CPU1□1
CK3W-AX2323□
CK3W-GC2200

SBCE-CN5-504A

声明

- 严禁擅自对本手册的部分或全部内容进行影印、复制或转载。
- 因产品改良的关系，本手册记载的产品规格等有时可能会不经预告而变更，恕不事先通知。
- 本手册内容力求尽善尽美，如有不明或错误之处等，烦请联系本公司分部或营业所。届时，请一并告知卷末记载的手册编号。

商标

- Microsoft、Windows、Excel、Visual Basic 是美国 Microsoft Corporation 在美国及其他国家的注册商标或商标。
- EtherCAT®是德国 Beckhoff Automation GmbH 提供许可的注册商标，相关知识产权由倍福公司所有。

本手册中记载的其它公司名称、产品名称为各公司的商标或注册商标。

著作权

- 屏幕截图的使用已获得微软的许可。
- 本产品已安装第三方软件。关于软件的许可和著作权，请参考 http://www.fa.omron.co.jp/nj_info_e/。

目录构成

目录

第 4 章 各种设定的自定义方法

相关手册

为安全使用系统，请务必获取系统构成机器和设备的手册、使用说明书等资料，在充分确认和理解手册、使用说明书上记载的内容（包括“安全注意事项”、“安全要点”等安全相关的注意事项以及“使用注意事项”）的基础上使用。
欧姆龙株式会社（以下称欧姆龙）、美国 Delta Tau Data Systems 公司（以下称 DT 公司）的手册如下所示。

制造商	手册编号	型号	手册名称
欧姆龙	SBCE-CN5-435	CK3M-CPU□□□ CK3W-AX2323□ CK3W-GC2200	可编程多轴运动控制器 用户手册 硬件篇
DT 公司	SBCE-CN5-404	—	Power PMAC 用户手册
DT 公司	SBCE-CN5-405	—	Power PMAC 软件基准手册
DT 公司	SBCE-CN5-439	—	Power PMAC IDE 用户手册(V4)

修订记录

修订符号附记在封面和封底下面记载的手册编号的末尾。

修订符号	修订年月	修订理由
A	2021 年 5 月	第一版

术语和定义

术语	说明和定义
PMAC	Programmable Multi Axis Controller 的缩写。
Power PMAC IDE	该计算机软件用于设定控制器、创建用户程序、监视。
AC 伺服电机	以交流信号输入驱动的伺服电机。通过交流信号的输入，在电机内部形成旋转磁场，使电机的转子旋转。
伺服放大器	伺服电机的控制设备。收到来自控制器的指令，通过功率设备切换电源，输出动力信号驱动电机。
Direct PWM	这是 Power PMAC 和伺服放大器的通信方式，是本公司的特色。Power PMAC 直接发送伺服放大器功率设备的 ON/OFF 信号，而伺服放大器则将电流反馈值发送至 Power PMAC。这样，可以在 Power PMAC 中形成电流回路。
电角	指用于旋转 AC 伺服电机的旋转磁场的角度。
换相控制	指产生具有所需电角的旋转磁场，使电机旋转。
相位基准	电角的原点。进行换相控制时，首先需要建立相位基准（相位基准搜索）。
正弦波编码器	可输出 1VPP 的 SIN/COS 波形的编码器。
扫描振镜	通过控制激光反射镜的旋转角来扫描激光照射位置的控制设备。可利用反射镜的高响应性能进行高速轨迹控制。
SL2-100	与数字扫描振镜进行通信的协议之一。
控制指令	该指令可设定扫描振镜向控制器发送数据时的数据类型。
XY 滑台	向 X 轴方向、Y 轴方向运动的定位滑台。与扫描振镜相比，响应速度较慢，但可动范围大，因此可对更大的区域进行扫描。
MOTF	Motion On The Fly 的缩写。该算法可将位置指令分为高频分量和低频分量，并将它们分配给具有不同响应速度的驱动器。在本书中，高频分量分配到扫描振镜，低频分量分配到 XY 滑台。
脉冲激光	以特定时间间隔反复闪烁的激光。与之相应，不间断输出的设备称为连续光振荡激光。
Q 开关	脉冲激光的振荡方式之一。从外部输入用于振荡的脉冲，输出脉冲激光。
TCR	Trigger Output by Commanded distance for Rapid Processing 的缩写。根据 CK3W-GC□2□□的距离触发激光输出的功能。

注意事项

- 构建实际系统时，请确认构成系统的各机器和设备的规格，确保额定值和性能有一定的余量，并采取安全措施，例如构建安全回路，确保在发生故障时也能将危险控制在最低。
- 为安全使用系统，请先获取各系统构成机器和设备的手册、使用说明书等资料，在充分确认和理解手册、使用说明书上记载的内容（包括“安全注意事项”、“安全要点”等安全相关的注意事项以及“使用注意事项”）的基础上使用。
- 系统需要符合的标准、法规或规定请客户自行确认。
- 未经欧姆龙株式会社允许，禁止复印、复制或重新分发本资料的部分或全部。
- 本资料的内容是截至 2021 年 5 月的最新信息。
  因产品改良的关系，本资料的记载内容有时可能会不经预告而变更。

本资料中使用的标记含义如下。

使用注意事项

表示为了防止产品出现动作不良、误动作或严重影响其性能、功能，应该实施或避免的行为。

参考

希望根据需要阅读的项目。
了解后有助于使用的信息以及使用时可参考的内容。

图标说明

● 符号表示强制。
具体内容如 ● 的内部图标和文字叙述所示。
左图表示“一般强制事项”。

1

概要

本章介绍本资料的概要。

1-1 概要

本资料介绍了在欧姆龙产可编程多轴运动控制器 CK3M-□□□□（以下称控制器）上构建激光加工系统的步骤。

⚠ 注意

本资料的使用范围为列入连接目标的设备的连接确认。使用本资料中未记载的指令或系统构成时，请先获取各系统构成机器和设备的手册、使用说明书等资料，在确认“安全注意事项”、“安全要点”等安全相关的注意事项后，再开展作业。

2

设备构成

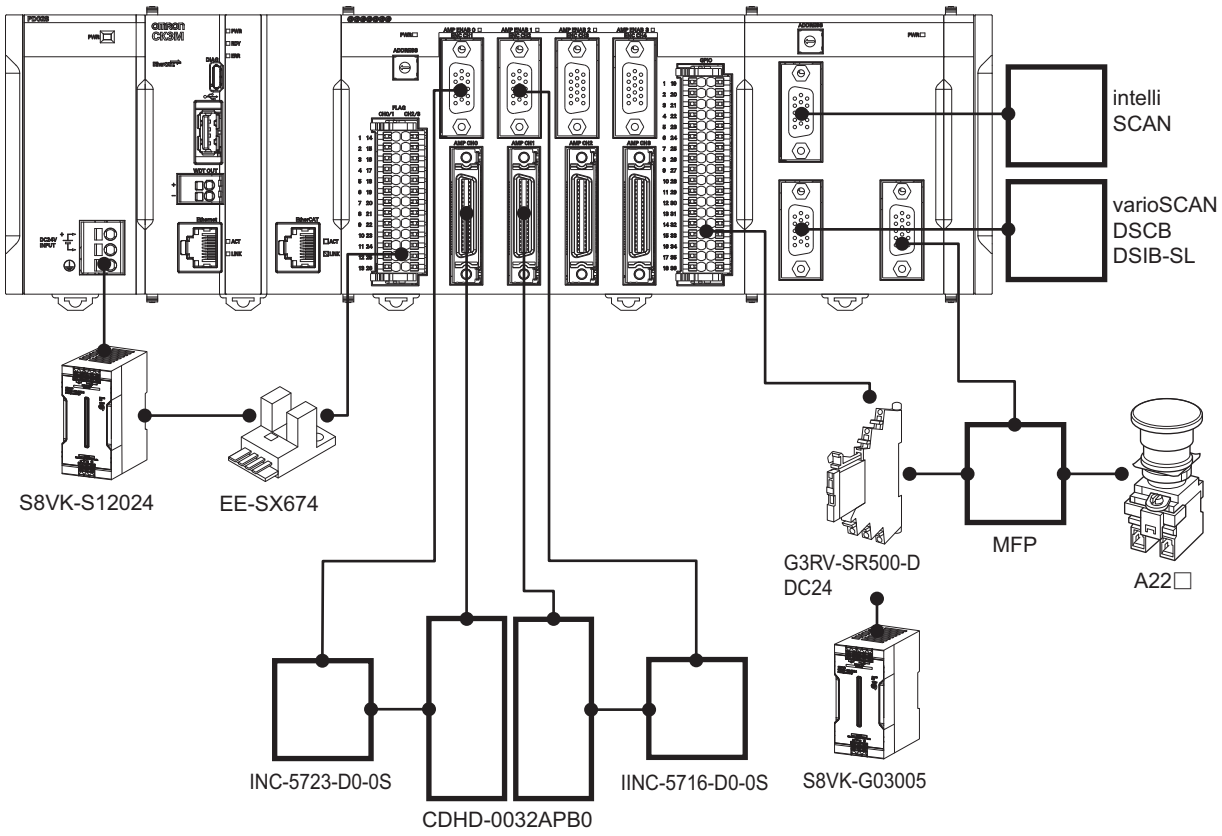
本章介绍设备构成。

2-1 设备构成

用于重现本资料中连接步骤的构成设备如下。

制造商	名称	型号	版本
欧姆龙	可编程多轴运动控制器 CPU 单元	CK3M-CPU1□1	Ver.2.6.1 以上
欧姆龙	可编程多轴运动控制器 轴接口单元	CK3W-AX2323N*1	—
欧姆龙	可编程多轴运动控制器 振镜接口单元	CK3W-GC2200	—
欧姆龙	可编程多轴运动控制器 电源供给单元	CK3W-PD048	—
欧姆龙	可编程多轴运动控制器 端盖	CK3W-TER11	—
欧姆龙	可编程多轴运动控制器 Direct PWM 电缆	CK3W-CAAD036A	—
欧姆龙	可编程多轴运动控制器 串行编码器电缆	CK3W-CAES03A	—
欧姆龙	可编程多轴运动控制器 激光接口单元电缆	CK3W-CAG03A	—
欧姆龙	开关电源(24V DC)	S8VK-S12024	—
欧姆龙	开关电源(5V DC)	S8VK-G03005	—
欧姆龙	开关电源(15V DC)	S8FS-G10015CD	—
欧姆龙	固态继电器	G3RV-SR500-D DC24	—
欧姆龙	微型光电传感器	EE-SX674	—
Servotronix	Direct PWM 放大器	CDHD-0032APB0*1	—
Maxphotonics	激光振荡器	MFP	—
INNO6	直线电机（Y 轴）	INC-5716-D0-0S	—
INNO6	直线电机（X 轴）	INC-5723-D0-0S	—
HEIDENHEIN	直线编码器	LIDA48*1	—
SCANLAB	扫描振镜	intelliSCAN	—
SCANLAB	fθ 镜头	—	—
SCANLAB	动态对焦单元	varioSCAN	—
—	Windows 计算机	—	—
DT 公司	Power PMAC 设定工具	Power PMAC IDE	Ver.4.5 以上

*1. 本书的前提是以 Direct PWM 或正弦波编码器方式控制 XY 滑台。要采用其他控制方式时，请参考其他启动指南，变更「3-4 电机和编码器的设定(P.3-21)」的设定内容。


intelli
SCAN
varioSCAN
DSCB
DSIB-SL
S8VK-S12024
EE-SX674
G3RV-SR500-D
DC24
MFP
A22□
INC-5723-D0-0S
CDHD-0032APB0
IINC-5716-D0-0S
S8VK-G03005

3

连接步骤

本章介绍如何与构建激光应用的各个设备连接。前提是控制器处于出厂时的初始设定状态。

3-1 作业流程

构建激光应用的步骤如下所示。

「3-2 控制器的设定准备(P.3-4)」	进行控制器的设定准备。
▼	
「3-2-1 新建项目(P.3-4)」	
▼	
「3-2-2 控制器的初始设定(P.3-5)」	
▽	
「3-3 设定 MOTF(P.3-7)」	设定 MOTF。
▼	
「3-3-1 编程(P.3-7)」	
▼	
「3-3-2 确定过滤器系数(P.3-13)」	
▽	
「3-4 电机和编码器的设定(P.3-21)」	设定电机和编码器。
▼	
「3-4-1 各设备的配线(P.3-21)」	
▼	
「3-4-2 编程(P.3-22)」	
▼	
「3-4-3 电流回路的调谐(P.3-25)」	
▼	
「3-4-4 建立相位基准(P.3-27)」	
▼	
「3-4-5 开放回路测试(P.3-28)」	
▼	
「3-4-6 位置回路的自动调谐(P.3-29)」	
▼	
「3-4-7 位置回路的交互调谐(P.3-30)」	
▼	
「3-4-8 Y 轴的调谐(P.3-34)」	
▼	
「3-4-9 确认调谐结果(P.3-34)」	
▽	
「3-5 设定超程限制开关和原点(P.3-37)」	设定各种传感器。
▼	
「3-5-1 各设备的配线(P.3-37)」	
▼	
「3-5-2 编程(P.3-37)」	
▼	
「3-5-3 确认设定（超程限制开关）(P.3-39)」	
▼	
「3-5-4 设定原点(P.3-40)」	
▽	
「3-6 设定扫描振镜(P.3-42)」	设定扫描振镜。
▼	
「3-6-1 各设备的配线(P.3-42)」	
▼	
「3-6-2 编程(P.3-43)」	

▼	
「3-6-3 控制指令的设定(P.3-45)」	
▼	
「3-6-4 设定的确认(P.3-47)」	
▼	
「3-6-5 MOTF 的上机验证(P.3-48)」	
▽	
「3-7 设定激光振荡器(P.3-49)」	设定激光振荡器。
▼	
「3-7-1 各设备的配线(P.3-49)」	
▼	
「3-7-2 引导激光的输出(P.3-49)」	
▼	
「3-7-3 编程(P.3-51)」	
▼	
「3-7-4 设定激光振荡器(P.3-54)」	
▼	
「3-7-5 设定的确认(P.3-55)」	

3-2 控制器的设定准备

做好控制器的设定准备。
请事先将 Power Programmable Multi Axis Controller IDE 安装到电脑上。

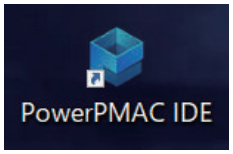
3-2-1 新建项目

按照以下步骤新建项目。

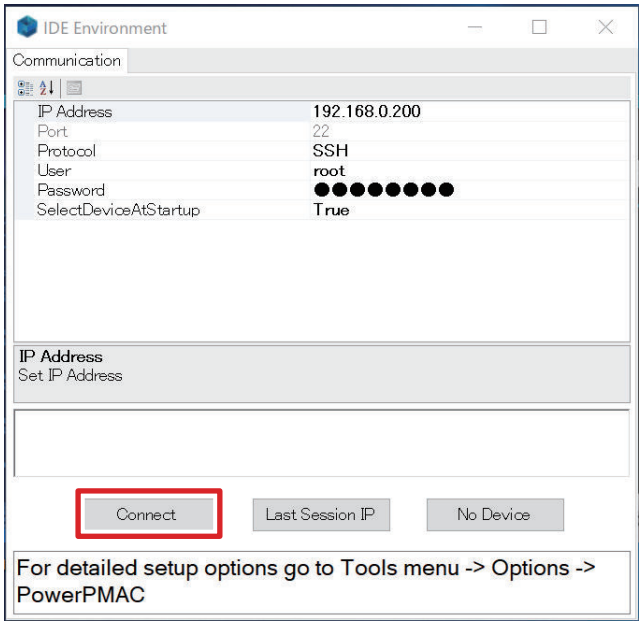
1 通过 Ethernet 电缆连接控制器和计算机。

2 接通控制器的电源。

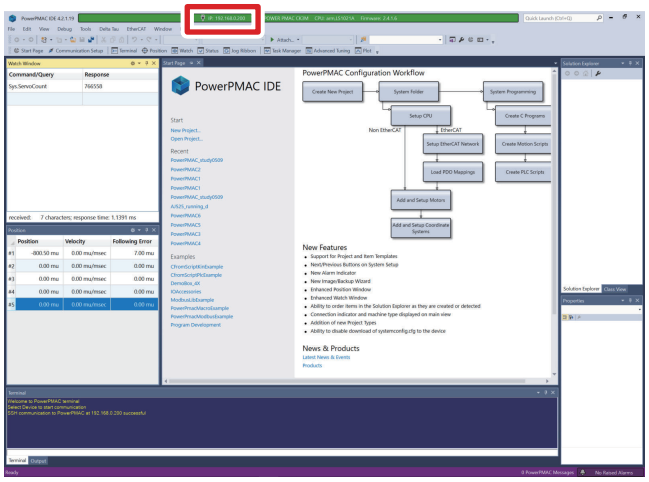
3 启动 Power PMAC IDE。
- 启动时，如果显示访问权限确认对话框，请选择启动。



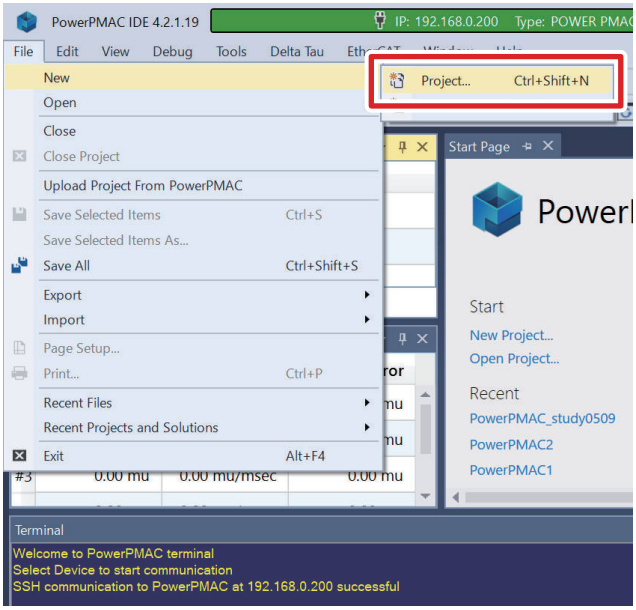
4 将显示 Communication 画面，请指定连接目标控制器的 IP 地址，然后单击［Connect］按钮。
- 控制器的默认 IP 地址为“192.168.0.200”。
- 如有需要，请将 Windows 的 IP 地址变更为“192.168.0.X”。



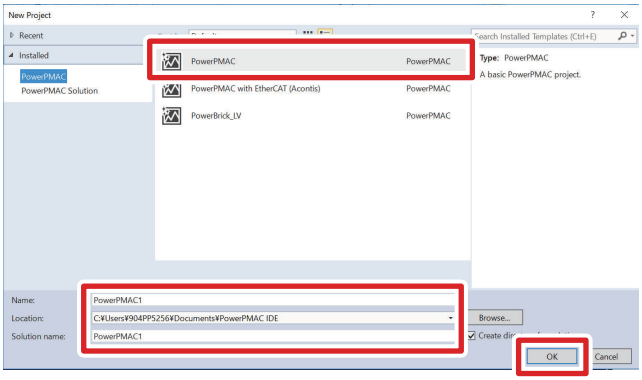
5 启动 Power PMAC IDE，并与控制器变为在线状态。



6 选择［File］菜单中的［New］–［Project］。



7 如右图所示，在选择［Power PMAC］后，输入任意项目名称和保存位置，然后单击［OK］按钮。



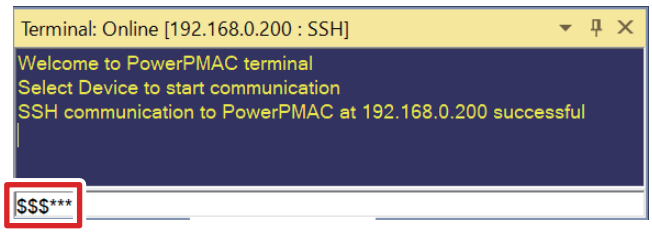
3-2-2 控制器的初始设定

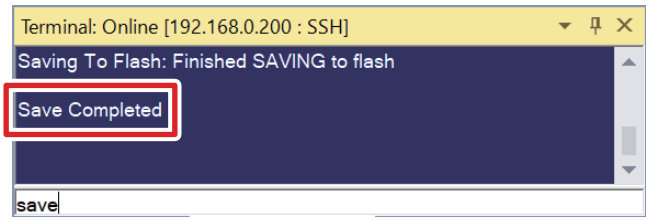
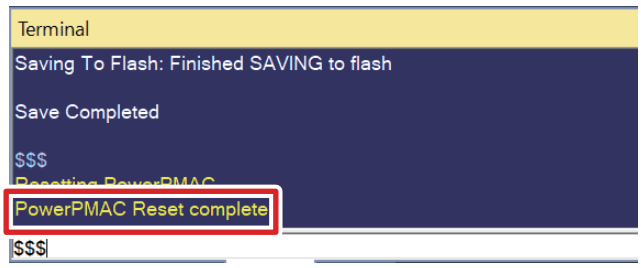
按照以下步骤进行控制器的初始设定。

使用注意事项

初始设定后，存储器将全部被清除，因此如果控制器中还有需要的数据，请先保存数据。

1 通过 Terminal（终端）输入［$$$***］指令，将控制器恢复为出厂状态。



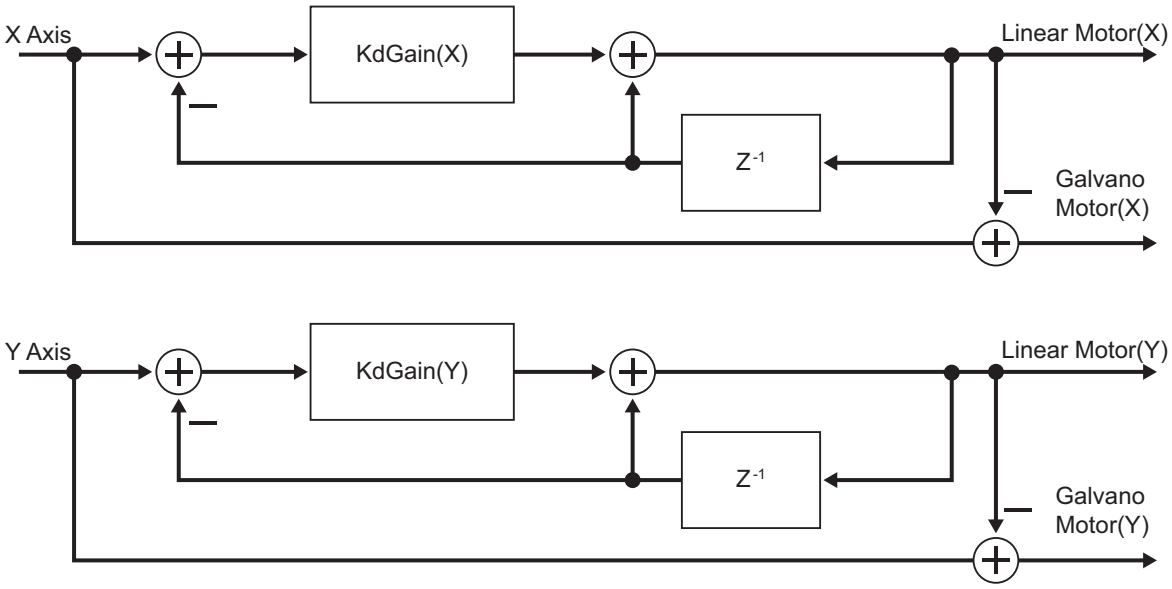
2	在 Power PMAC IDE 的 Terminal（终端）中输入［save］指令。 • 结束后，将在 Terminal（终端）中显示“Save Completed”。	
3	在 Power PMAC IDE 的 Terminal（终端）中输入［$$$］指令。 • 结束后，将在 Terminal（终端）中显示“PowerPMAC Reset completed”。	

3-3 设定 MOTF

3-3-1 编程

设定 MOTF。在本书中，使用 EMA（指数平滑移动平均）的低通滤波器对低频分量和高频分量进行排序。
以下表示 EMA 的输入输出方程式。其中，u(n)为时刻 n 时的输入信号，y(n)为输出信号。
y(n)=KdGain*u(n)+(1-KdGain)*y(n-1)

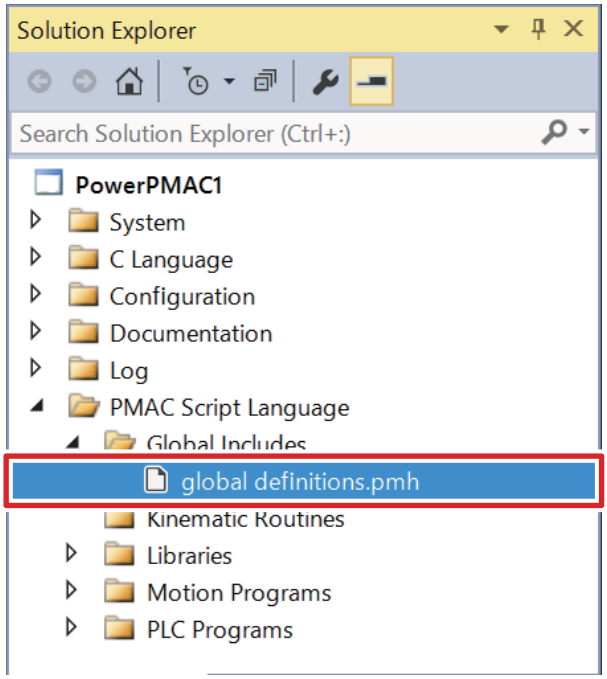
以下表示 MOTF 的框线图。将某一轴的指令中，通过低通滤波器后的低频分量输入到电机，将剩余的移动量输入到扫描振镜。这样即可实现高行程和高响应的兼顾。



在上面，KdGain 设定为 0 到 1 之间的值。想要增大低通滤波器的截止频率时，请将 KdGain 设定为更大的值。
在本书中，设备上的轴与电机编号之间的关系如下表所示。

设备上的轴		电机编号
机构	方向	
XY 滑台	X 轴	#1
	Y 轴	#2
扫描振镜	X 轴	#3
	Y 轴	#4
对焦单元	Z 轴	#5
TCR 计算用的虚拟轴		#6

1 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［global definitions.pmh］。

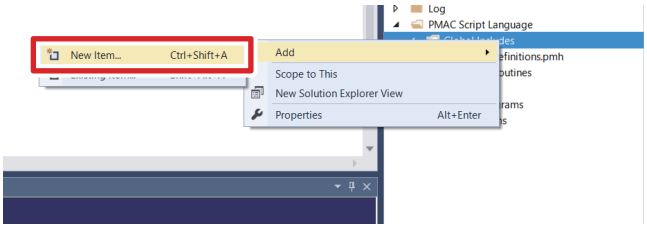


2 将右侧的文本写入到 global definitions.pmh 中。

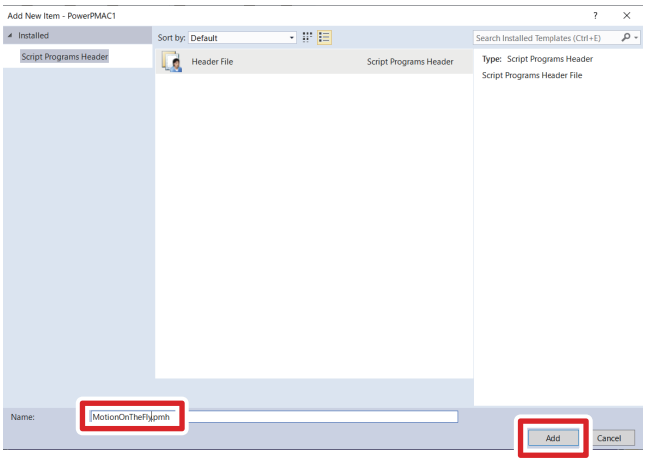
- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

```
Sys.WpKey=$AAAAAAAA
//Segmentation Time
Coord[1].SegMoveTime=0.1;//*1

//Global Clock Configurations
Gate3[0].PhaseFreq=10000;//*2
Gate3[0].ServoClockDiv=0;//*3
Sys.PhaseOverServoPeriod=1;//*4
Sys.ServoPeriod=0.1;//*5
Gate3[0].PhaseServoDir=0;//*6
Gate3[1].PhaseServoDir=3;//*6

//ADC settings
Gate3[0].AdcEncCtrl=$3fffc000
```

3 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］，从［Add（添加）］中选择［New Item（添加项）］。



4 在［Name（名称）］栏中输入「MotionOnTheFly.pmh」，单击［Add（添加）］按钮。



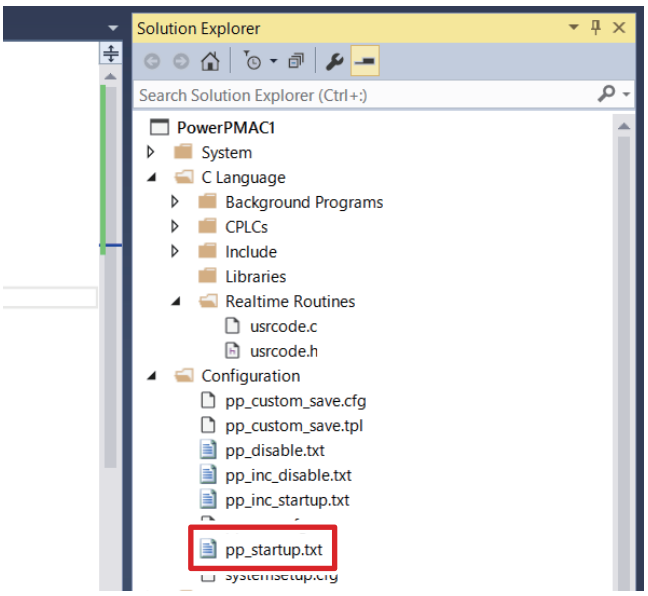
5 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［MotionOnTheFly.pmh］。

6 将右侧的文本写入到 MotionOnTheFly.pmh 中。

- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

```
//LPF's Gain
global KdgainX=0.00045;//*1
global KdgainY=0.00045;//*2

//Scale Factors of Galvo Scanner
global GalvoSfX=-503316*4096/(170*0.374);//*3
global GalvoSfY=-503316*4096/(170*0.374);//*4

//Settings of Coordinate System
&1;//*5
#1->I;//*5
#2->I;//*5
#3->I;//*5
#4->I;//*5
```

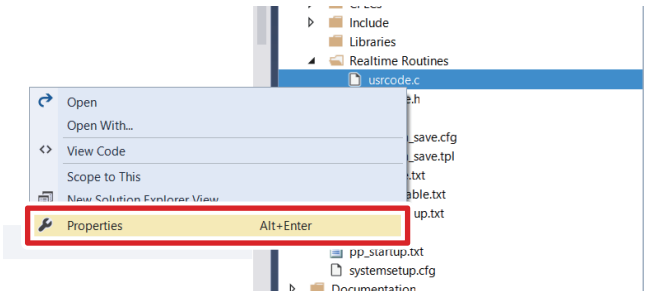
7 打开 Solution Explorer（解决方案资源管理器）的［Configuration］-［pp_startup.txt］。



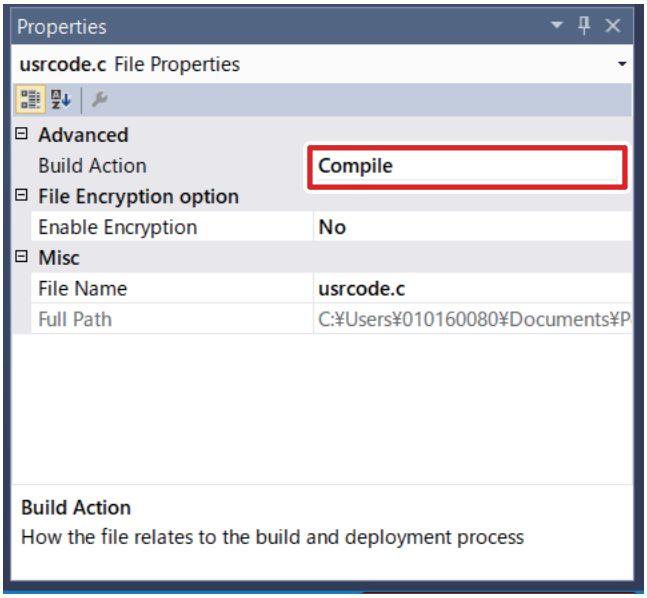
8 将右侧的文本写入到 pp_startup.txt 中。

```
//Enable CfromScript
UserAlgo.CFunc=1;
```

9 右击 Solution Explorer（解决方案资源管理器）的［C Language］–［Realtime Routines］–［usercode.c］，选择［Properties］。



10 在［Properties］窗口的［Build Action］中选择［Compile］。



11 打开 Solution Explorer（解决方案资源管理器）的［C Language］–［Realtime Routines］–［usercode.c］。

12 删除右侧的 CfromScript 函数定义。

```
double CfromScript(double cfrom_type,double arg2,
double arg3,double arg4,double arg5,double arg6,d
ouble arg7,struct LocalData *Ldata)
{
    int icfrom_type=(int)cfrom_type;
    double*C,*D,*L,*R,rtn;//C,D,R-only needed if
doing Kinematics

    C=GetCVarPtr(Ldata);//Only needed if doing Ki
nematics
    D=GetDVarPtr(Ldata);//Only needed if doing Ki
nematics
    L=GetLVarPtr(Ldata);//Only needed if using Ld
ata or Kinematics
    R=GetRVarPtr(Ldata);//Only needed if doing Ki
nematics
    rtn=-1.0;
    return rtn;
}
```

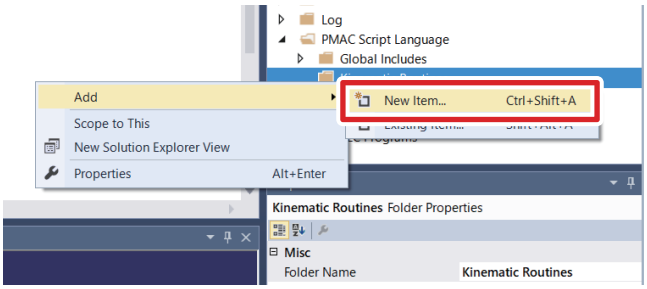
13 将右侧的文本写入到 usercode.c 中。

```
static double prevLpfXPos=0;
static double prevLpfYPos=0;

double CfromScript(double kinTypeDouble,double ar
g2,double arg3,double arg4,double arg5,double arg
6,double arg7,LocalData *Ldata)
{
 int kinType=(int)kinTypeDouble;
 double deltaXPos,deltaYPos;
 double *L=GetLVarPtr(Ldata);
 double *C=GetCVarPtr(Ldata);

 switch(kinType)
 {
  case Forward_Kinematics_State:
  {
   _KinPosAxisX=_KinPosMotor1+(_KinPosMotor3/Galv
oSfX);
   _KinPosAxisY=_KinPosMotor2+(_KinPosMotor4/Galv
oSfY);
   if(Ldata->Status & 0x40)
   {
    prevLpfXPos=_KinPosMotor1;
    prevLpfYPos=_KinPosMotor2;
   }
   break;
  }
  case Inverse_Kinematics_State:
  {
   deltaXPos=_KinPosAxisX-prevLpfXPos;
   deltaYPos=_KinPosAxisY-prevLpfYPos;
   _KinPosMotor1=prevLpfXPos+(KdgainX*deltaXPos);
   _KinPosMotor2=prevLpfYPos+(KdgainY*deltaYPos);
   _KinPosMotor3=(_KinPosAxisX-_KinPosMotor1)*Gal
voSfX;
   _KinPosMotor4=(_KinPosAxisY-_KinPosMotor2)*Gal
voSfY;
   prevLpfXPos=_KinPosMotor1;
   prevLpfYPos=_KinPosMotor2;
   break;
  }
 }
 return 0;
}
```

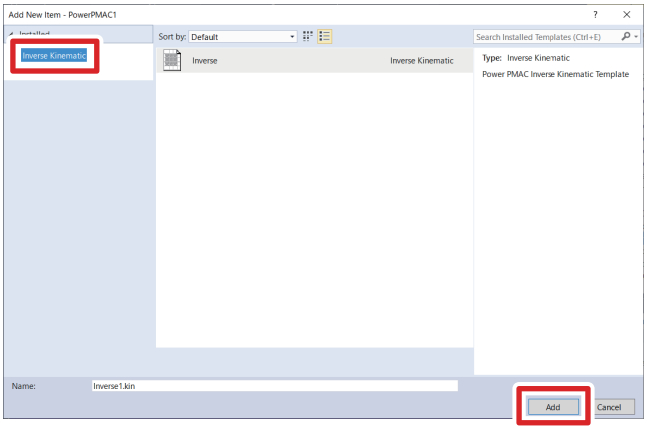
14 在 usercode.c 文件的开头处添加右侧的文本。

```
#define Forward_Kinematics_State 0
#define Inverse_Kinematics_State 1

#define _KinPosAxisX *(C+6)
#define _KinPosAxisY *(C+7)
#define _KinPosMotor1 *(L+1)
#define _KinPosMotor2 *(L+2)
#define _KinPosMotor3 *(L+3)
#define _KinPosMotor4 *(L+4)
```

15 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］-［Kinematics Routines］，从［Add（添加）］中选择［New Item（添加项）］。



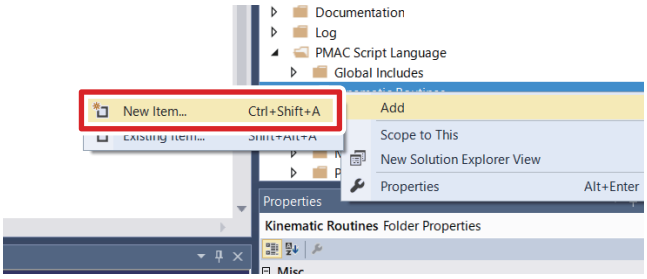
16 选择［Inverse Kinematic］，单击［Add（添加）］按钮。



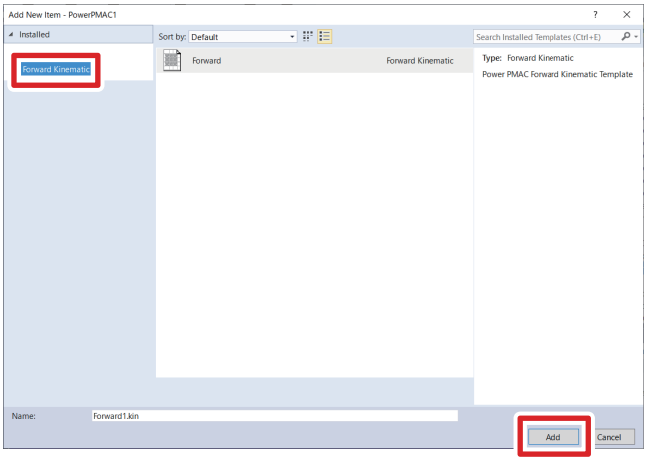
17 将右侧的文本写入到 Inverse1.kin 中。

```
open inverse(1)

local ret;
ret=CfromScript(1,0,0,0,0,0,0);

close
```

18 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］-［Kinematics Routines］，从［Add（添加）］中选择［New Item（添加项）］。



19 选择［Forward Kinematic］，单击［Add（添加）］按钮。

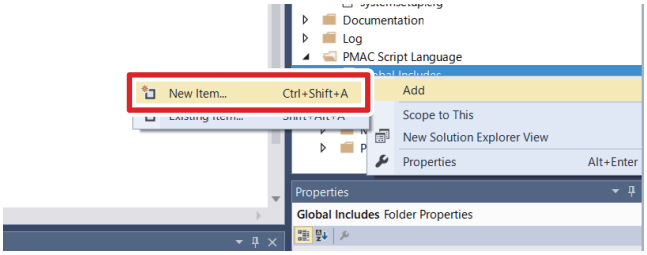


20	将右侧的文本写入到 Forward1.kin 中。	`open forward(1)`
		
		`local ret;`
		`ret=CfromScript(0,0,0,0,0,0,0);`
		
		`close`

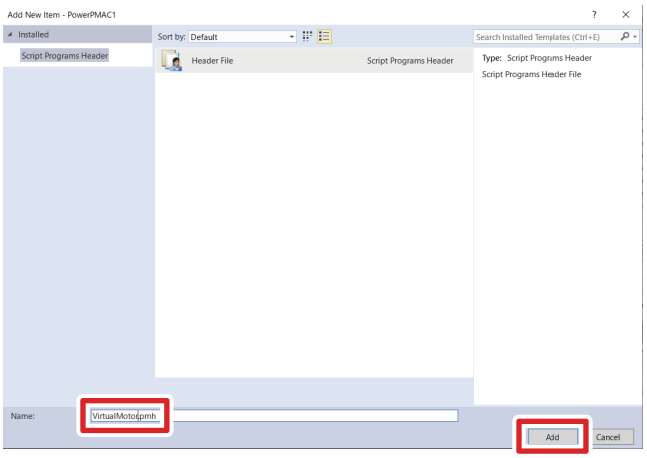
3-3-2 确定过滤器系数

调整用于 MOTF 的过滤器的参数 KdGain。

1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



2 在［Name（名称）］栏中输入「VirtualMotor.pmh」，单击［Add（添加）］按钮。



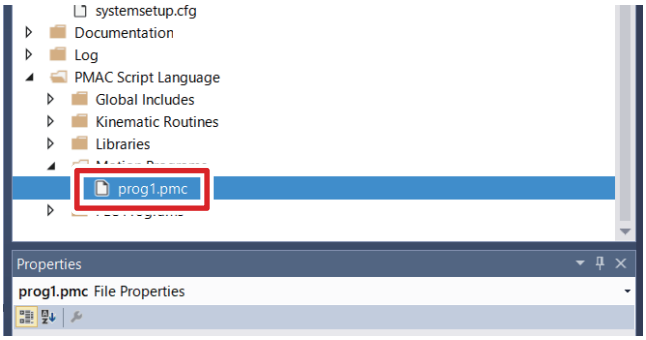
3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［VirtualMotor.pmh］。

4 将右侧的文本写入到 VirtualMotor.pmh 中。

```
EncTable[1].type=1
EncTable[1].index1=0
EncTable[1].index2=0
EncTable[1].index3=0
EncTable[1].index4=0
EncTable[1].pEnc1=Sys.udata[10].a
EncTable[1].pEnc =Sys.udata[10].a;
EncTable[1].ScaleFactor=1;
Motor[1].pDac=Sys.udata[10].a
Motor[1].pEnc =EncTable[1].a
Motor[1].pEnc2=EncTable[1].a
Motor[1].pLimits=0
Motor[1].pAmpFault=0
Motor[1].Ctrl=Sys.PosCtrl
Motor[1].ServoCtrl=1
Motor[1].MaxSpeed=0
Motor[1].FatalFeLimit=0

EncTable[2].type=1
EncTable[2].index1=0
EncTable[2].index2=0
EncTable[2].index3=0
EncTable[2].index4=0
EncTable[2].pEnc1=Sys.udata[11].a
EncTable[2].pEnc=Sys.udata[11].a;
EncTable[2].ScaleFactor=1;
Motor[2].pDac=Sys.udata[11].a
Motor[2].pEnc=EncTable[2].a
Motor[2].pEnc2=EncTable[2].a
Motor[2].pLimits=0
Motor[2].pAmpFault=0
Motor[2].Ctrl=Sys.PosCtrl
Motor[2].ServoCtrl=1
Motor[2].MaxSpeed=0
Motor[2].FatalFeLimit=0
```

```
EncTable[3].type=1
EncTable[3].index1=0
EncTable[3].index2=0
EncTable[3].index3=0
EncTable[3].index4=0
EncTable[3].pEnc1=Sys.udata[12].a
EncTable[3].pEnc=Sys.udata[12].a
EncTable[3].ScaleFactor=1
Motor[3].pDac=Sys.udata[12].a
Motor[3].pEnc=EncTable[3].a
Motor[3].pEnc2=EncTable[3].a
Motor[3].pLimits=0
Motor[3].pAmpFault=0
Motor[3].Ctrl=Sys.PosCtrl
Motor[3].ServoCtrl=1
Motor[3].MaxSpeed=0
Motor[3].FatalFeLimit=0

EncTable[4].type=1
EncTable[4].index1=0
EncTable[4].index2=0
EncTable[4].index3=0
EncTable[4].index4=0
EncTable[4].pEnc1=Sys.udata[13].a
EncTable[4].pEnc=Sys.udata[13].a
EncTable[4].ScaleFactor=1
Motor[4].pDac=Sys.udata[13].a
Motor[4].pEnc=EncTable[4].a
Motor[4].pEnc2=EncTable[4].a
Motor[4].pLimits=0
Motor[4].pAmpFault=0
Motor[4].Ctrl=Sys.PosCtrl
Motor[4].ServoCtrl=1
Motor[4].MaxSpeed=0
Motor[4].FatalFeLimit=0
```

5 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Motion Programs］下面的［prog1.pmc］。

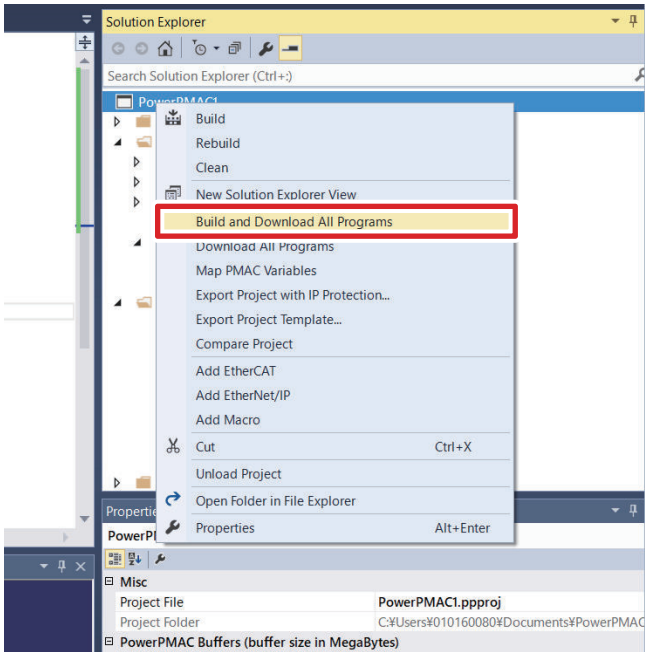


6 将加工时使用的程序写入到 prog1.pmc 中。
右侧的文本为加工程序的一个示例。

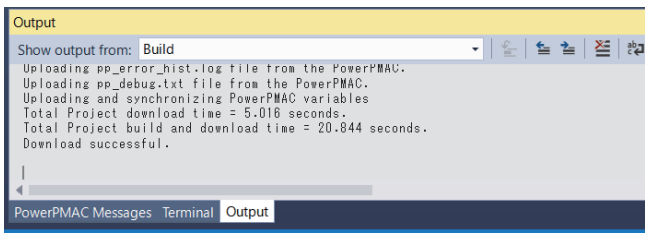
```
open prog 1

coord[1].FeedTime=1000;
Coord[1].MaxFeedRate=180;

ta(0.1);
td(0.1);
ts(0);
F(100);

abs;
linear;

X(50) Y(50);
X(-50) Y(50);
X(-50) Y(-50);
X(50) Y(-50);
X(50) Y(50);
X(0) Y(0);

close
```

7 项目的下载

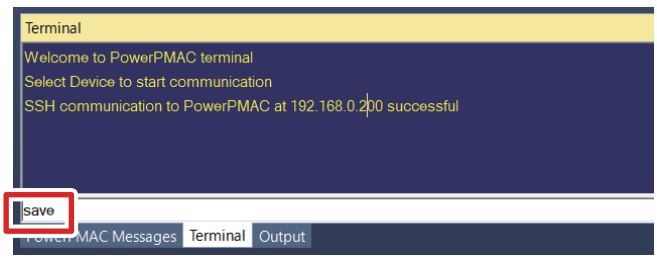
右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。



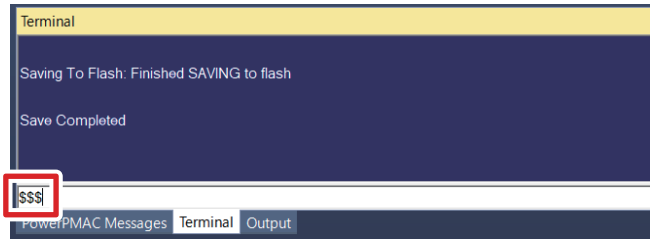
8 确认 Output（输出）没有异常。
- 传送失败时，请通过 Output（输出）确认错误的内容。



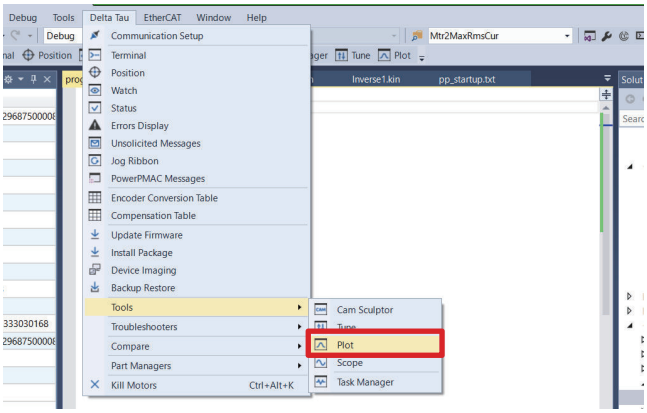
9 在 Power PMAC IDE 的 Terminal（终端）中输入［save］指令。
- 结束后，将在 Terminal（终端）中显示“Save Completed”。



10 在 Power PMAC IDE 的 Terminal（终端）中输入［$$$］指令。

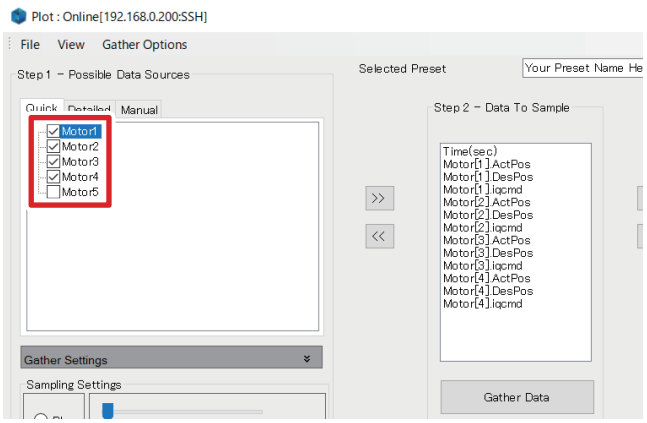


11 在［Delta Tau］菜单的［Tools（工具）］中选择［Plot（绘图）］。



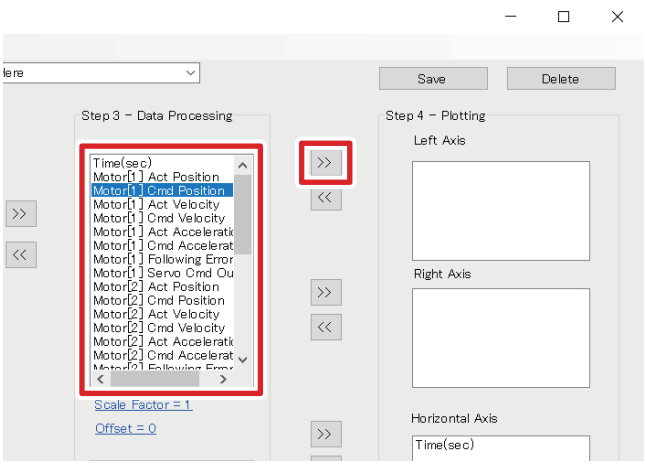
12 勾选［Step1 – Possible Data Sources（第 1 步-可能的数据源）］中以下项目的复选框。

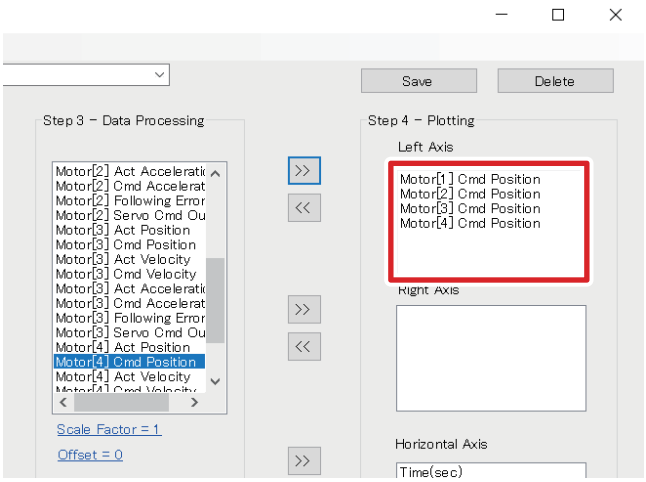
- ［Motor1］
- ［Motor2］
- ［Motor3］
- ［Motor4］



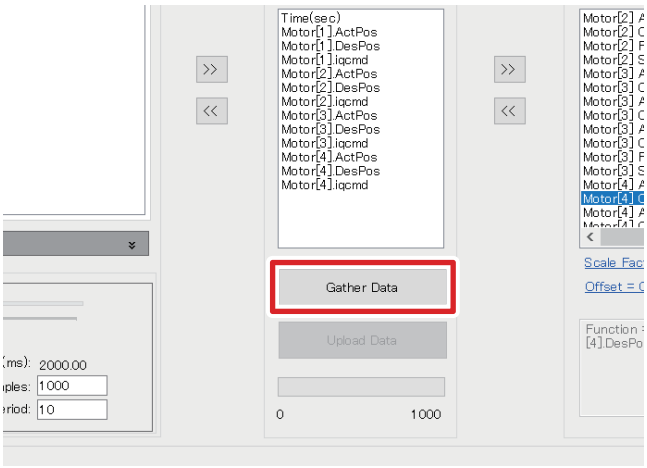
13 选择［Step3 – Data Processing（第 3 步-数据处理）］中的以下项目，单击［>>］按钮。

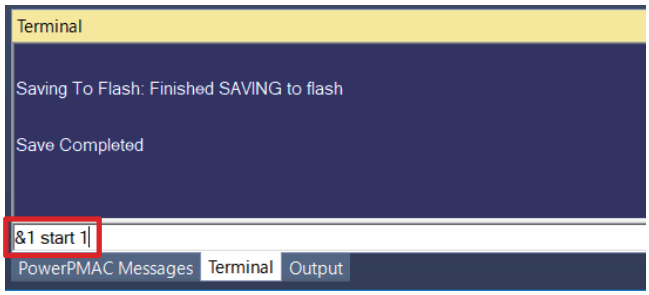
- ［Motor[1] Cmd Position］
- ［Motor[2] Cmd Position］
- ［Motor[3] Cmd Position］
- ［Motor[4] Cmd Position］



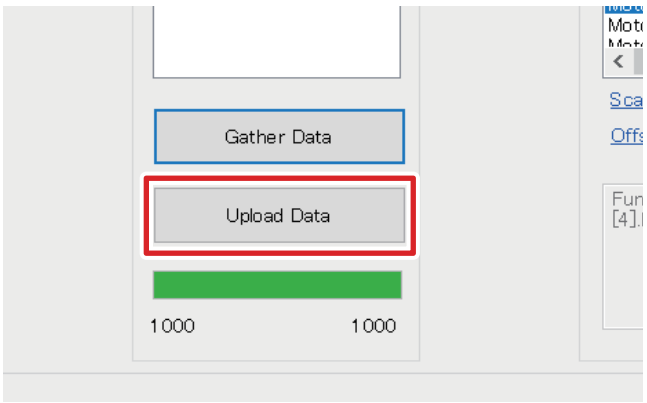


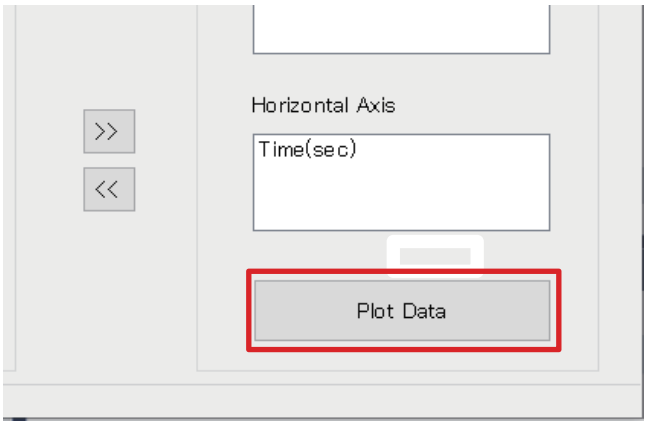
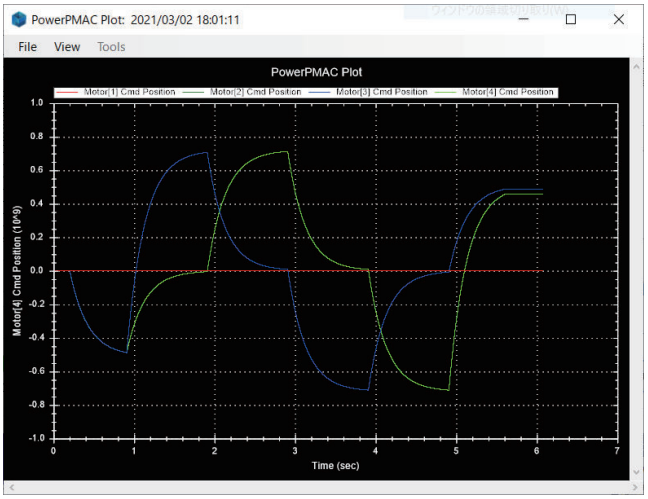
14 单击［Gather Data（收集数据）］按钮，在 Power PMAC IDE 的 Terminal（终端）中输入［&1 enable］和［&1 start 1］指令。





15 单击［Upload Data（上载数据）］按钮。



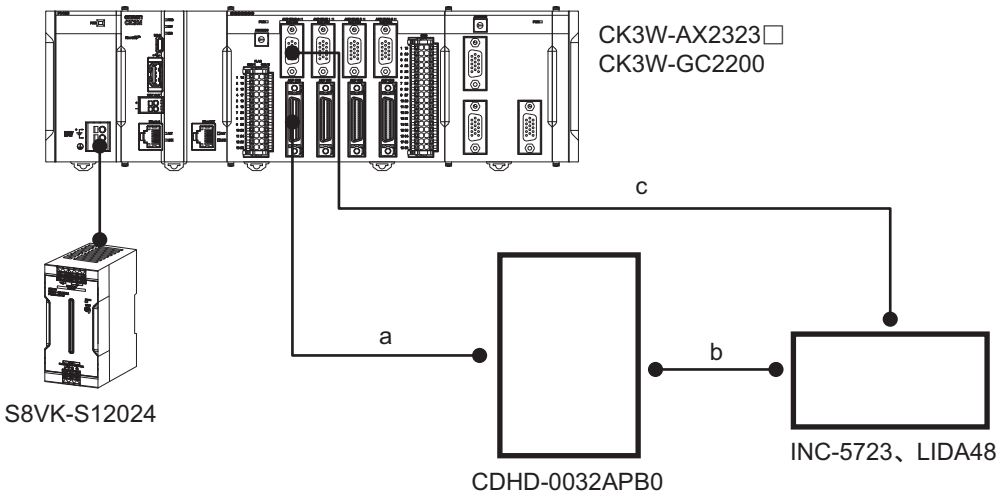
16	单击［Plot Data（绘图数据）］按钮。	
17	如右侧所示，确认显示各轴的位置。	
18	Motor[3] Cmd Position、Motor[4] Cmd Position 中的任意一个超过扫描振镜的指令最大值/最小值时，应打开［global definitions.pmh］，对 KdgainX、KdgainY 进行编辑。然后，再次执行步骤 7～17。	
19	如果步骤 18 中过滤器的响应没有问题，则右击 Solution Explorer（解决方案资源管理器）上的［VirtualMotor.pmh］，选择［Delete］。	

3-4 电机和编码器的设定

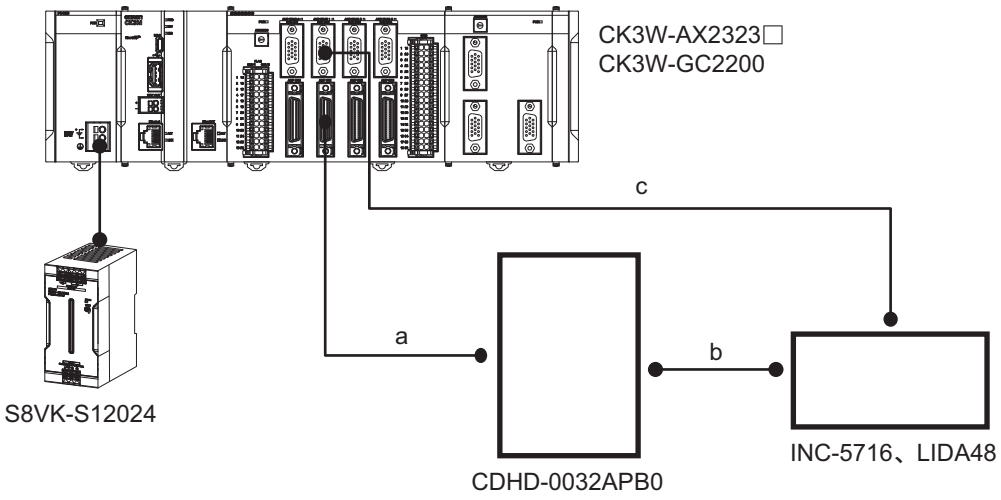
3-4-1 各设备的配线

介绍 CK3M 和伺服驱动器、直线电机的配线方法。

• X 轴的配线



• Y 轴的配线



按照以下要领连接上图 a、b、c 位置。

a. 控制器与伺服放大器的连接
用以下专用电缆连接 CK3W-AX2323□的放大器连接器和伺服放大器的 C2 连接器。

制造商	名称	型号	长度
欧姆龙	Direct PWM 电缆	CK3W-CAAD009A	0.9m
		CK3W-CAAD018A	1.8m
		CK3W-CAAD036A	3.6m

b. 伺服放大器和伺服电机的连接
在伺服放大器的 P2 连接器上连接直线电机的 Motor Connector。

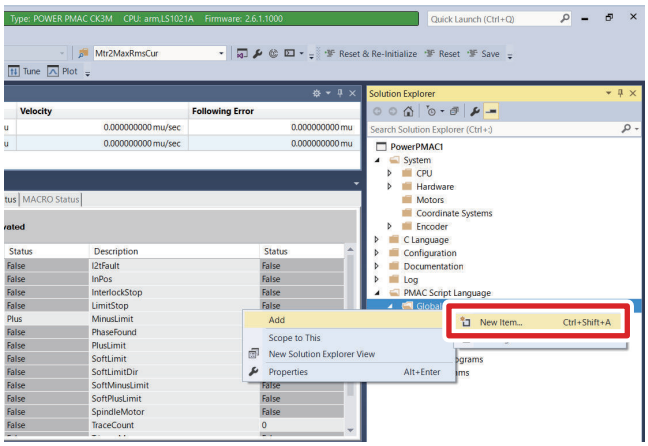
c. 控制器与伺服电机的连接
用专用电缆 CK3W-CAEA03A 连接 CK3W-AX2323□的编码器连接器和直线编码器的连接器。

CK3W-AX2323			LIDA48	
Signal	Pin		Signal	Pin
Sinusoidal Encoder SIN+	1		A+	1
Sinusoidal Encoder SIN-	6		A-	9
Sinusoidal Encoder COS+	2		B+	3
Sinusoidal Encoder COS-	7		B-	11
Encoder Power Supply (+5VDC)	11		Up	4
Encoder Power Supply (GND)	13		0V	2

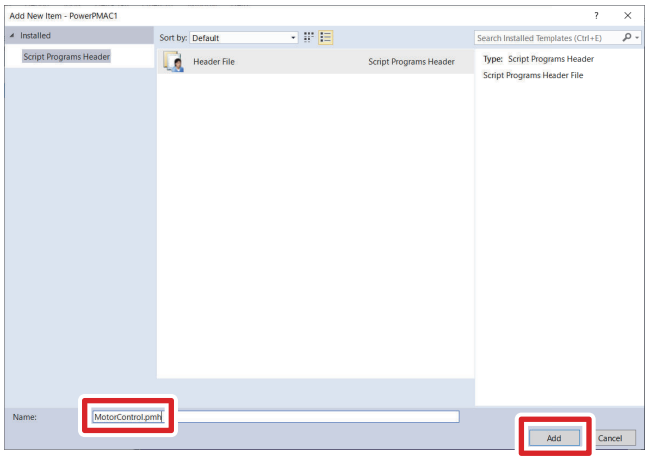
3-4-2 编程

对控制伺服放大器、直线电机的设定进行编程。

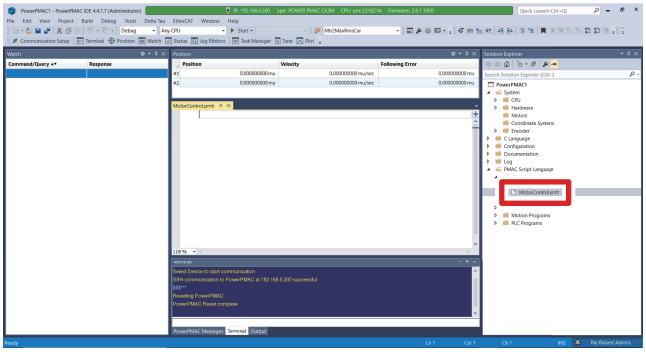
1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



2 在［Name（名称）］栏中输入「MotorControl.pmh」，单击［Add（添加）］按钮。



3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［MotorControl.pmh］。



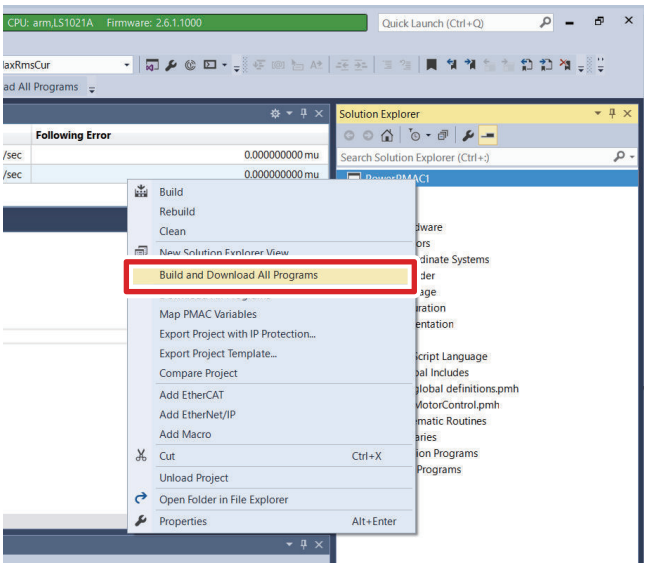
4 将下面的文本写入到 MotorControl.pmh 中。

- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

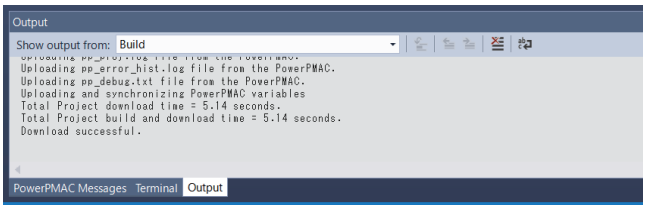
```
Sys.WpKey=$AAAAAAAA

//Servo Task Configurations
Motor[1].ServoCtrl=1//*1
Motor[2].ServoCtrl=1//*1
Motor[1].pEnc=EncTable[1].a//*2
Motor[2].pEnc=EncTable[2].a//*2
Motor[1].pEnc2=EncTable[1].a//*2
Motor[2].pEnc2=EncTable[2].a//*2
Motor[1].EncType=6 //*3
Motor[2].EncType=6 //*3
Motor[1].PosUnit=3 //*4
Motor[2].PosUnit=3 //*4
EncTable[1].Type=1//*5
EncTable[2].Type=1//*5
EncTable[1].pEnc=Gate3[0].Chan[0].ServoCapt.a//*6
EncTable[2].pEnc=Gate3[0].Chan[1].ServoCapt.a//*6
EncTable[1].ScaleFactor=1.0//*7
EncTable[2].ScaleFactor=1.0//*7
Motor[1].AmpFaultLevel=1//*8
Motor[2].AmpFaultLevel=1//*8
Motor[1].FatalFeLimit=1.0//*9
Motor[2].FatalFeLimit=1.0//*9

//Phase Task Configurations
Motor[1].PhaseCtrl=4;//*10
Motor[2].PhaseCtrl=4;//*10
Motor[1].pPhaseEnc=Gate3[0].Chan[0].PhaseCapt.a//*11
Motor[2].pPhaseEnc=Gate3[0].Chan[1].PhaseCapt.a//*11
Motor[1].PhasePosSf=2048*(0.020/16384)/36//*12
Motor[2].PhasePosSf=2048*(0.020/16384)/36//*12
Motor[1].PhaseOffset=683//*13
Motor[2].PhaseOffset=683//*13
Motor[1].pAdc=Gate3[0].Chan[0].AdcAmp[0].a//*14
Motor[2].pAdc=Gate3[0].Chan[1].AdcAmp[0].a//*14
Motor[1].AdcMask=$FFFF0000//*15
Motor[2].AdcMask=$FFFF0000//*15
Motor[1].pDac=Gate3[0].Chan[0].Pwm[0].a//*16
Motor[2].pDac=Gate3[0].Chan[1].Pwm[0].a//*16
Motor[1].PhaseFindingDac=Motor[1].I2tSet/2//*17
Motor[2].PhaseFindingDac=Motor[2].I2tSet/2//*17
Motor[1].PhaseFindingTime=3000/(2*Sys.ServoPeriod*(Sys.RtIntPeriod+1))//*18
Motor[2].PhaseFindingTime=3000/(2*Sys.ServoPeriod*(Sys.RtIntPeriod+1))//*18
```

```
//PWM Output Configurations
Motor[1].PwmSf=0.95*16384//*19
Motor[2].PwmSf=0.95*16384//*19
Motor[1].MaxDac=9.0*32768*COSD(30)/11.25//*20
Motor[2].MaxDac=9.0*32768*COSD(30)/11.25//*20
Motor[1].I2tSet=3.0*32768*COSD(30)/11.25//*21
Motor[2].I2tSet=3.0*32768*COSD(30)/11.25//*21
Motor[1].I2tTrip=(Motor[1].MaxDac*Motor[1].MaxDac-Motor[1].I2tSet*Motor[1].I2tSet)*
2//*22
Motor[2].I2tTrip=(Motor[2].MaxDac*Motor[2].MaxDac-Motor[2].I2tSet*Motor[2].I2tSet)*
2//*22
Gate3[0].Chan[0].PwmDeadTime=15//*23
Gate3[0].Chan[1].PwmDeadTime=15//*23
Gate3[0].Chan[0].PwmFreqMult=2//*24
Gate3[0].Chan[1].PwmFreqMult=2//*24
Gate3[0].Chan[0].PackOutData=0//*25
Gate3[0].Chan[1].PackOutData=0//*25

//Current Loop Configurations
Gate3[0].AdcAmpStrobe=$fffffc;//*26
Gate3[0].AdcAmpHeaderBits=2;//*27
Gate3[0].AdcAmpClockDiv=5;//*28
Gate3[0].AdcEncClockDiv=5//*28
Gate3[0].Chan[0].PackInData=0//*29
Gate3[0].Chan[1].PackInData=0//*29

//Sinusoidal Encoder Settings
Gate3[0].Chan[0].AtanEna=1//*30
Gate3[0].Chan[1].AtanEna=1//*30
Gate3[0].Chan[0].EncCtrl=3//*31
Gate3[0].Chan[1].EncCtrl=3//*31
Motor[1].PosSf=0.020/16384//*32
Motor[2].PosSf=0.020/16384//*32
Motor[1].Pos2Sf=Motor[1].PosSf//*32
Motor[2].Pos2Sf=Motor[2].PosSf//*32
```

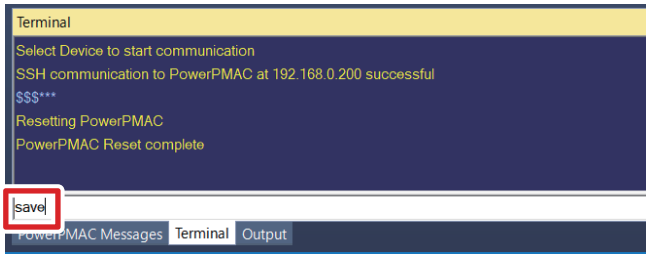
5 右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。



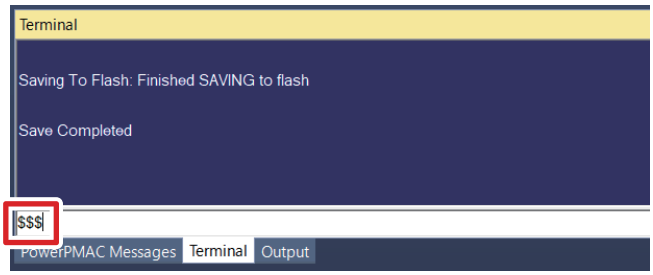
6 确认 Output（输出）没有异常。



7 在 Power PMAC IDE 的 Terminal（终端）中输入［save］指令。



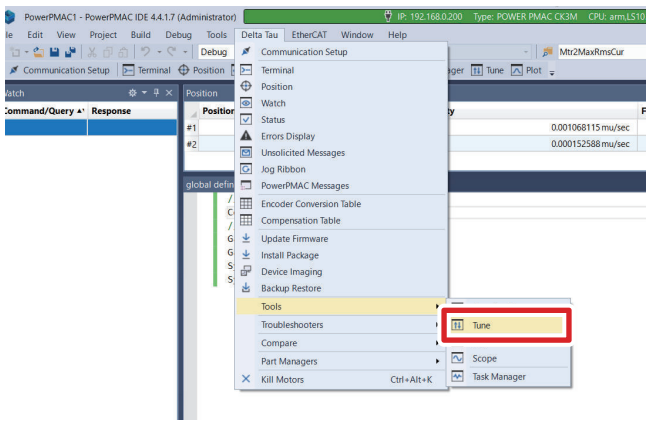
8 在 Power PMAC IDE 的 Terminal（终端）中输入［$$$］指令。



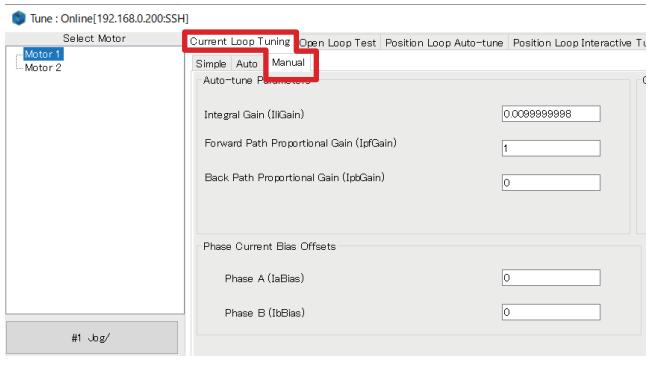
3-4-3 电流回路的调谐

对伺服放大器的电流回路进行调谐。

1 在［Delta Tau］菜单的［Tools（工具）］中选择［Tune］，打开 Tune 画面。

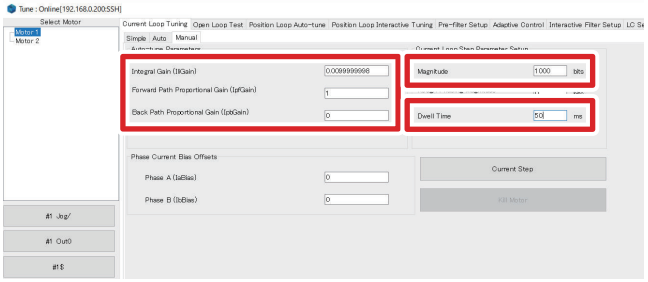


2 在 Tune 画面中选择［Current Loop Tuning］的［Manual］。



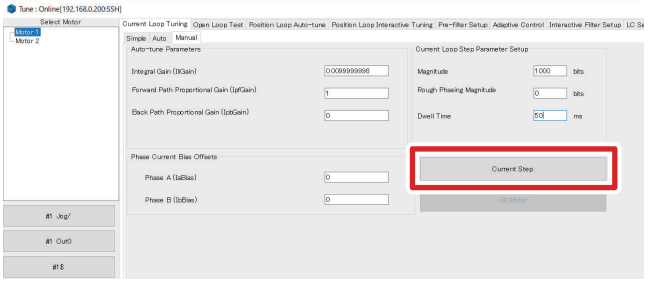
3 设定以下参数。

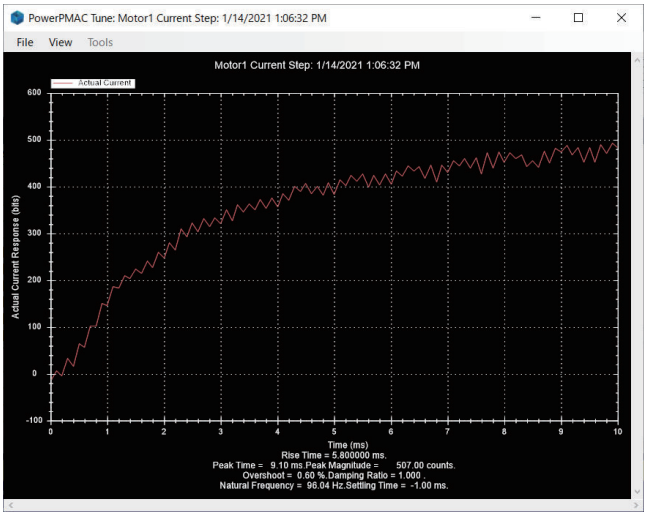
［IIiGain］: 0.0099999998（默认）
［IpfGain］: 1（默认）
［IpbGain］: 0
［Magnitude］: 1000bits
［Dwell Time］: 50ms



4 单击［Current Step］按钮。

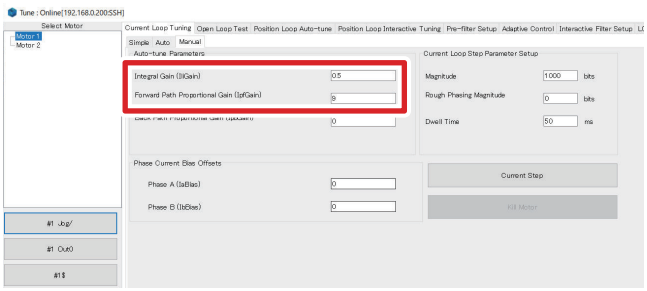
- 显示电流的阶跃响应。

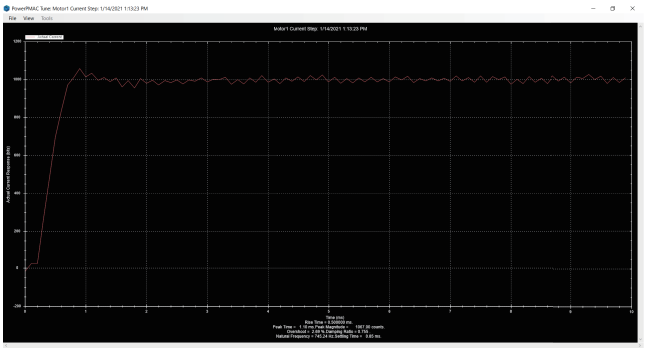




5 调整［IIiGain］和［IpfGain］，直至获得期待的响应特性。

- 启动响应较慢时，增大［IIiGain］。
- 过冲或振动较大时，增大［IpfGain］。
- 各个增益请从较小的值开始逐步增大。





3-4-4 建立相位基准

建立直线电机的相位基准。

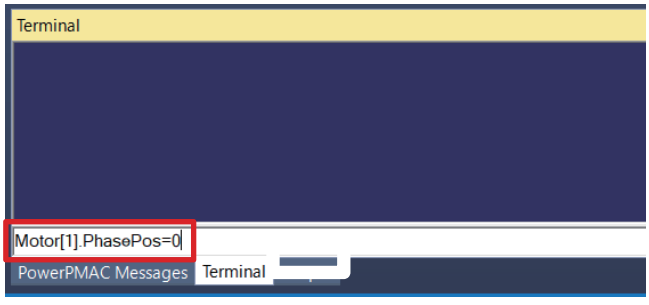
1 在 Power PMAC IDE 的 Terminal（终端）中输入［#1 out 0］指令。

2 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[1].IbBias=500］指令。

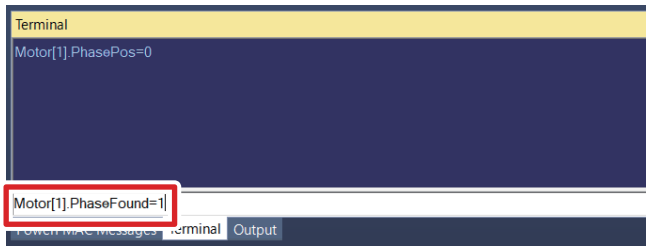
3 手动操作 X 轴，确认已被伺服锁定。

- 未被伺服锁定时，请逐步增大 Motor[1].IbBias 的值。

4 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[1].PhasePos=0］指令。



5 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[1].PhaseFound=1］指令。



6 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[1].IbBias=0］指令。

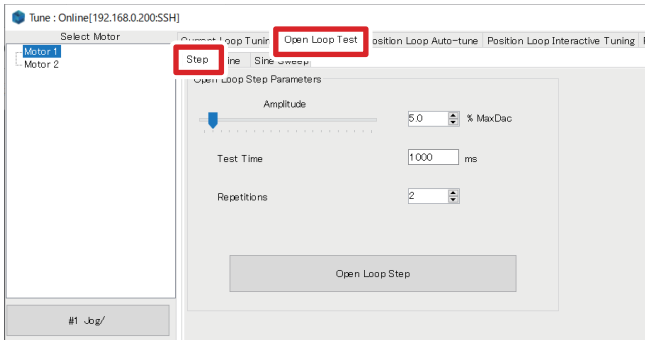
7 在 Power PMAC IDE 的 Terminal（终端）中输入［#1 k］指令。

8 对轴 2（Y 轴）重复执行步骤 1～7。

3-4-5 开放回路测试

以开放回路方式运行直线电机并确认程序是否正确。

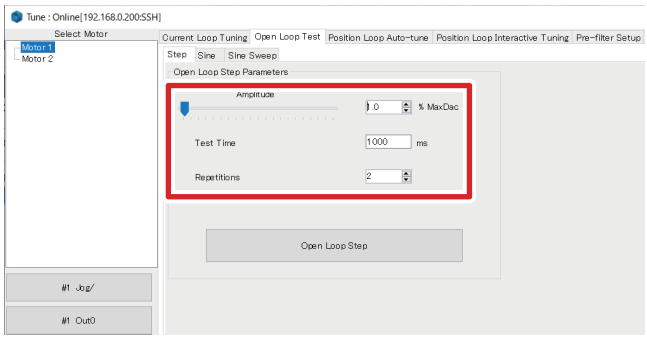
1 在［Delta Tau］菜单的［Tools（工具）］中选择［Tune］，打开 Tune 画面，再选择［Open LoopTest］的［Step］。



2 设定以下调谐参数。

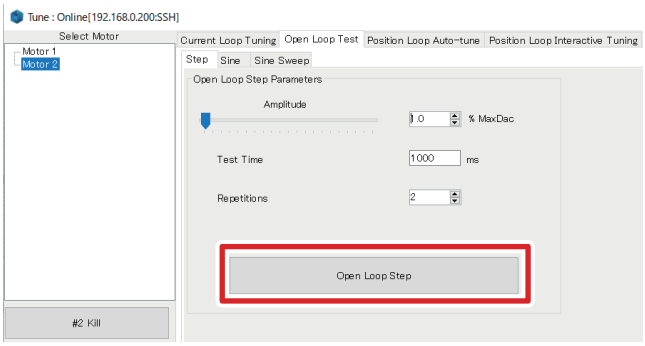
［Amplitude］：1.0%*1
［Test Time］：1,000ms
［Repetitions］：2

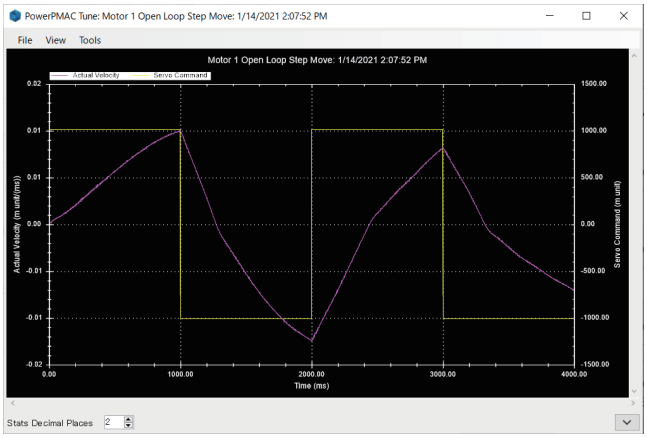
*1. 电机不旋转时，请设定为较大的值。



3 单击［Open Loop Step］按钮。

- 电机将往复运动，并显示右侧所示的测试结果。
- 电机不旋转时，请将［Test Amplitude］变更为较大的值。
- 此次的测试结果为［Test Amplitude］等于 8.0%时的情况。

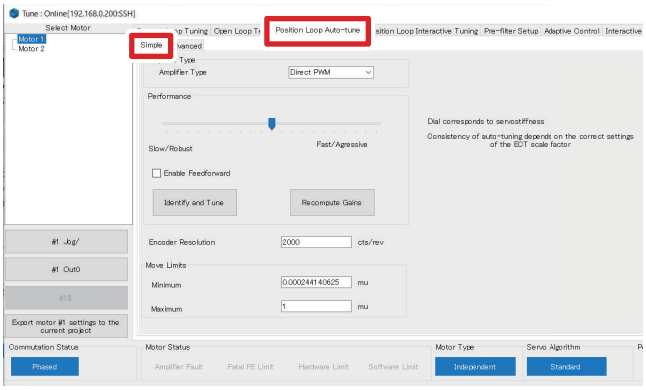




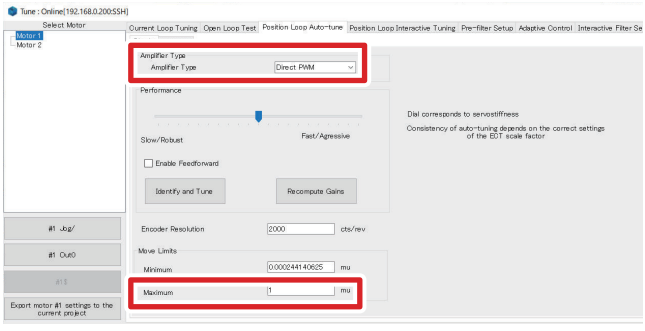
3-4-6 位置回路的自动调谐

对直线电机进行位置回路的自动调谐。

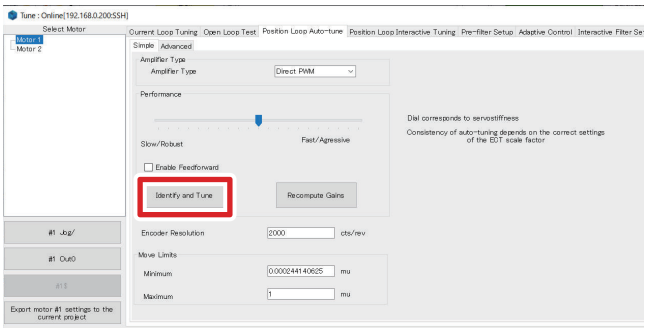
1 在 Tune 画面中选择［Position Loop Auto-tune］的［Simple］。



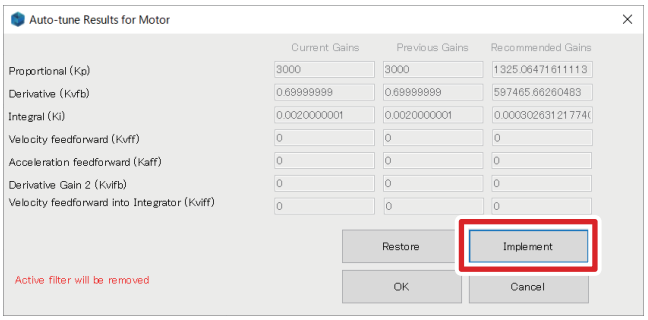
2 设定以下参数。
［Amplifier Type］: Direct PWM
［Maximum］: 1mu(1mm)



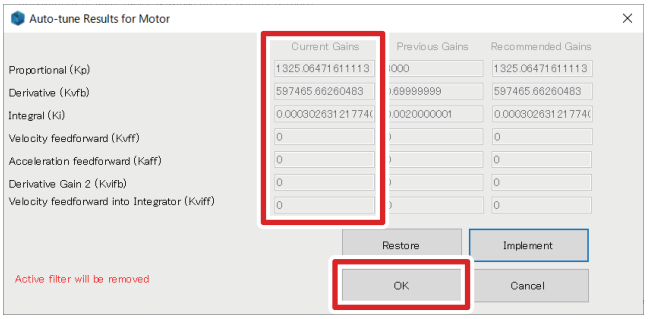
3 单击［Identify and Tune］按钮。



4 显示右侧画面后，单击［Implement］按钮。



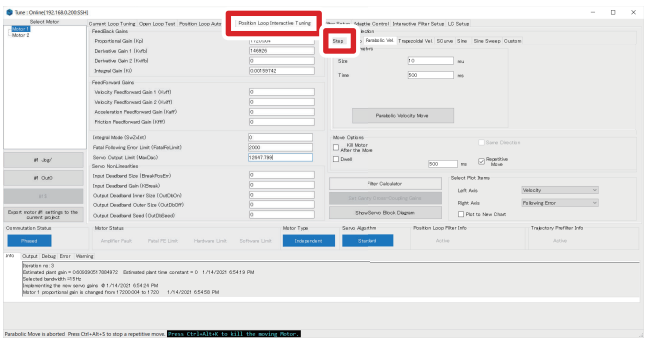
5 确认［Current Gains］中已反映［Recommended Gains］的值，然后单击［OK］按钮。



3-4-7 位置回路的交互调谐

对直线电机进行位置回路的交互调谐。

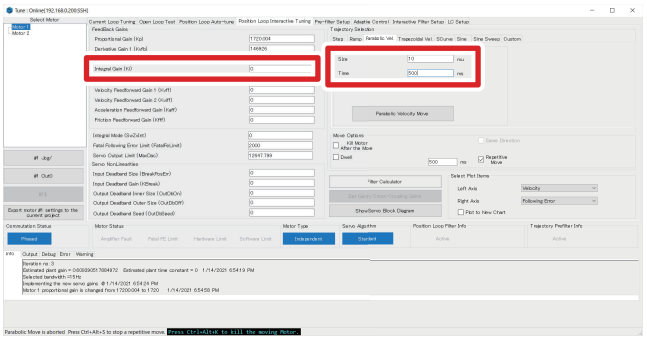
1 在 Tune 画面中选择［Position Loop Interactive Tuning］的［Step］。



2 设定以下参数。

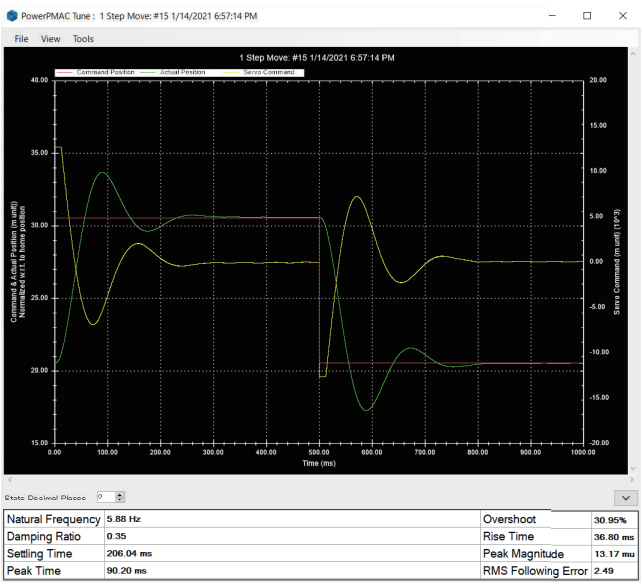
［FeedBack Gains］
［Integral Gains(Ki)］: 0

［Move Parameters］
［Size］: 10mu
［Time］: 1,000ms



3 单击［Step Move］按钮。

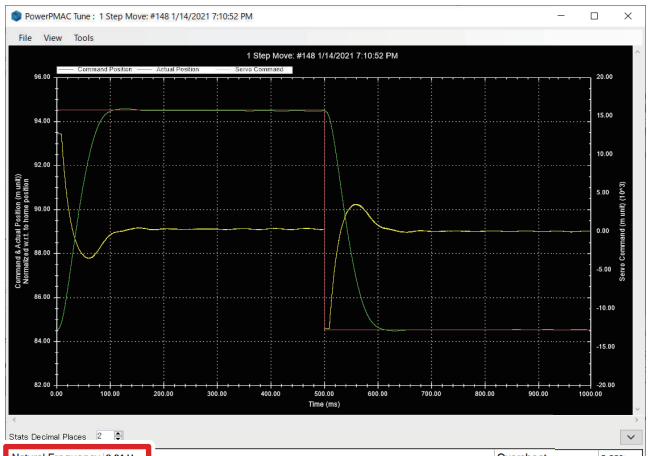
4 确认阶跃响应特性。



Natural Frequency	5.88 Hz	Overshoot	30.95%
Damping Ratio	0.35	Rise Time	36.80 ms
Settling Time	206.04 ms	Peak Magnitude	13.17 mu
Peak Time	90.20 ms	RMS Following Error	2.49

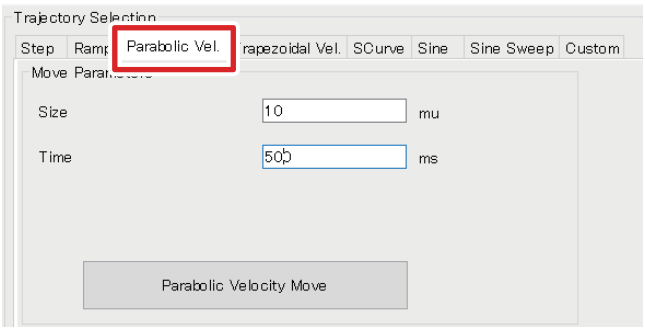
5 为获得期待的响应特性，对［Proportional Gain(Kp)］、［Derivative Gain 1(Kvfb)］进行调整。

- 启动响应较慢时，增大［Proportional Gain(Kp)］。
- 过冲或振动较大时，增大［Derivative Gain 1(Kvfb)］。
- 稳态偏差较大时，增大［Integral Gain 1(Ki)］。
- 各个增益请从较小的值开始逐步增大。
- 调整各参数，当［Natural Frequency］的值不再继续增大时，结束调谐。

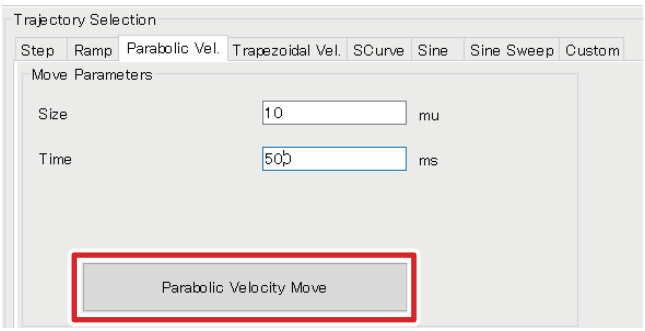


Natural Frequency	9.64 Hz	Overshoot	0.89%
		Rise Time	57.80 ms
Settling Time	81.75 ms	Peak Magnitude	10.03 mu
Peak Time	128.40 ms	RMS Followng Error	2.46

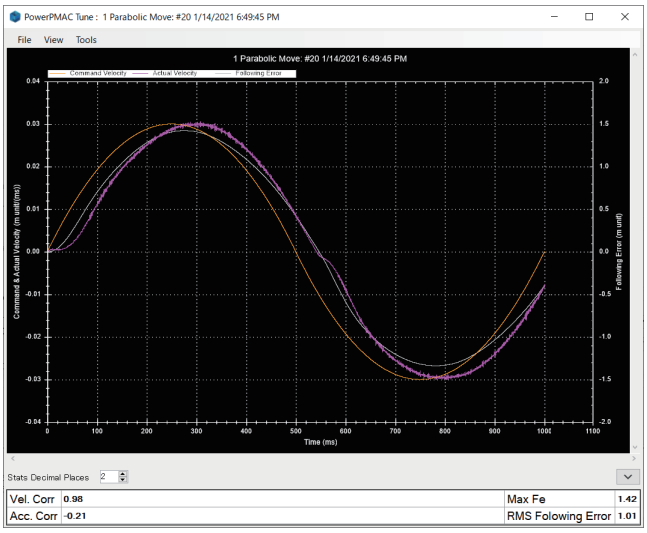
6 选择［Parabolic Vel.］，设定以下参数。
［Size］: 10mu(10mm)
［Time］: 500ms
［Left Axis］: Velocity
［Right Axis］: Following Error



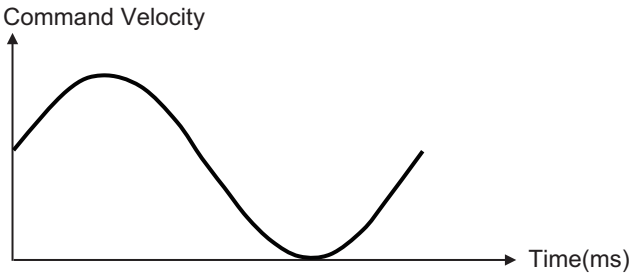
7 单击［Parabolic Velocity Move］按钮。

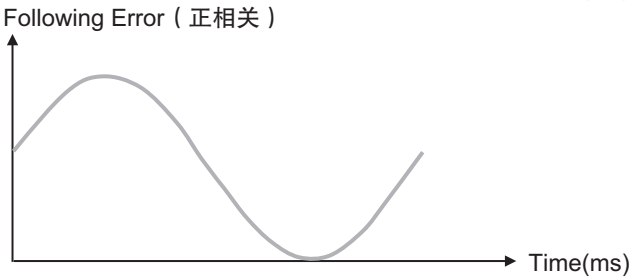


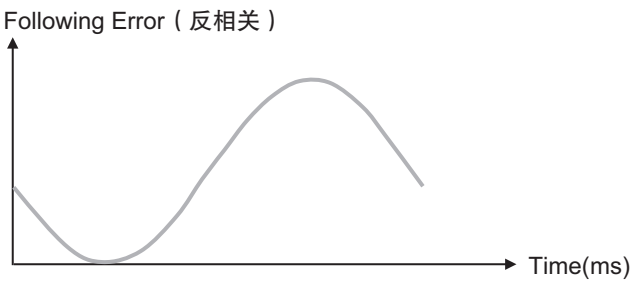
8 确认速度的抛物线响应特性。



9 ［Following Error］与速度呈正相关时，增大［Kvff］。呈反相关时，减小［Kvff］。

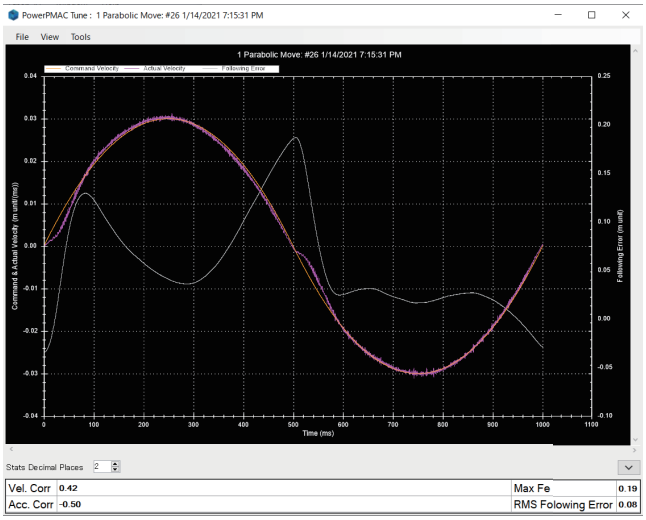




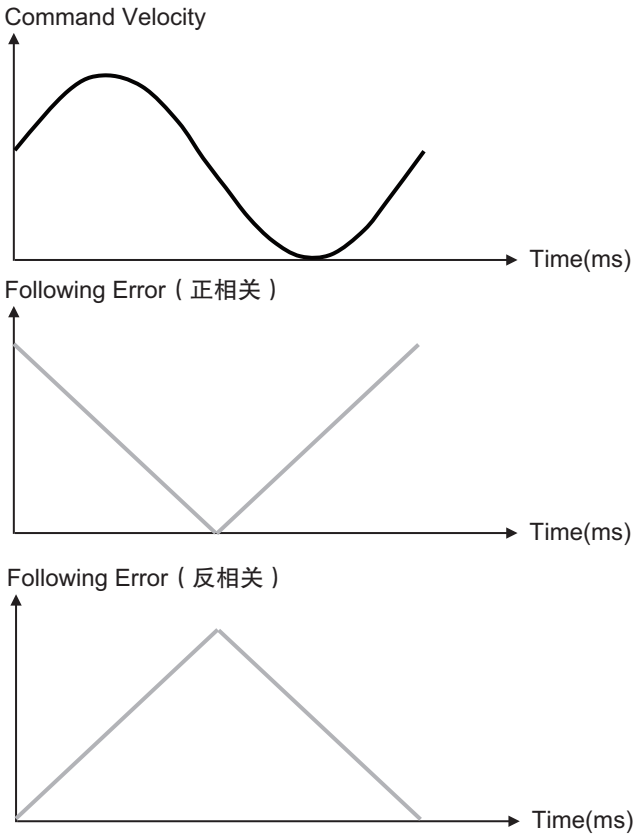


10 再次单击［Parabolic Velocity Move］按钮。

- 反复以上操作，直至［Following Error］与速度不再相关。



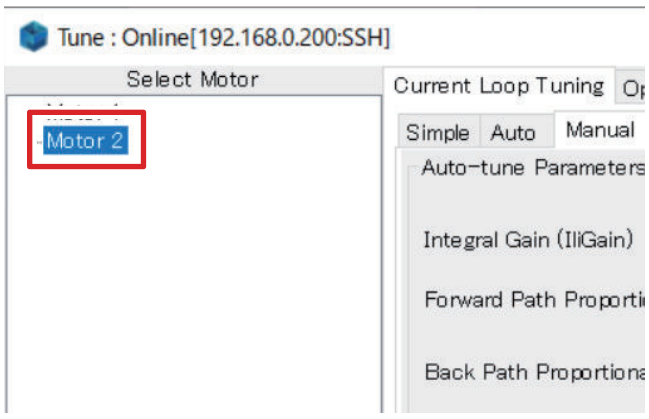
11 同样，［Following Error］与［Acceleration］（加速度）或摩擦具有相关性时，也需要增减［Kaff］、［Kfff］的值。



3-4-8 Y 轴的调谐

以同样的方式对 Y 轴进行调谐。

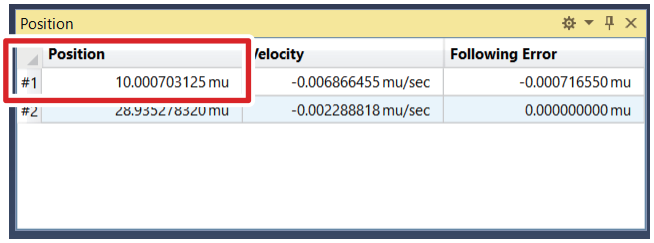
1 在 Tune 画面的［Select Motor］中选择［Motor2］。



2 分别执行电流回路的调谐、开放回路测试、位置回路的自动调谐、位置回路的交互调谐中描述的内容。

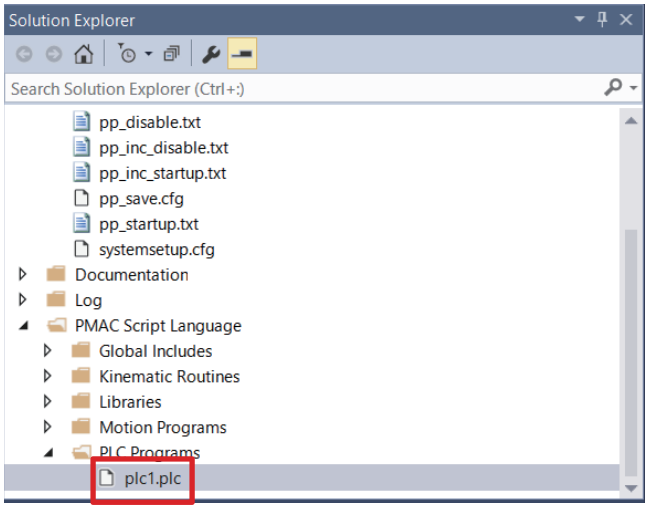
3-4-9 确认调谐结果

确认调谐结果正确，然后反映到项目文件中。

1	在 Power PMAC IDE 的 Terminal（终端）中输入［#1 hmz］指令。 然后，输入［#1 j=10］指令。	
2	确认 X 轴处于动作状态。同时，确认 Position 中［#1］的［Position］值接近 10.0。	
3	在 Power PMAC IDE 的 Terminal（终端）中输入［#1 k］指令，停止电机。	
4	对 Motor[2]重复执行步骤 1～3。	
5	打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［MotorControl.pmh］。	

6	将调谐得到的增益添加到 MotorControl.pmh 中。	`Motor[1].IiGain=***`
		`Motor[2].IiGain=***`
		`Motor[1].IpfGain=***`
		`Motor[2].IpfGain=***`
		`Motor[1].Servo.Kp=***`
		`Motor[2].Servo.Kp=***`
		`Motor[1].Servo.Kvfb=***`
		`Motor[2].Servo.Kvfb=***`
		`Motor[1].Servo.Ki=***`
		`Motor[2].Servo.Ki=***`
		`Motor[1].Servo.Kaff=***`
		`Motor[2].Servo.Kaff=***`
		`Motor[1].Servo.Kvff=***`
		`Motor[1].Servo.Kfff=***`

7	打开 Solution Explorer（解决方案资源管理器）的［Configuration］下面的［pp_startup.txt］。	
8	写入右侧所示的相位搜索执行指令。	`enable plc PhaseSearch`

9	打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［PLC Programs］下面的［plc1.plc］。	

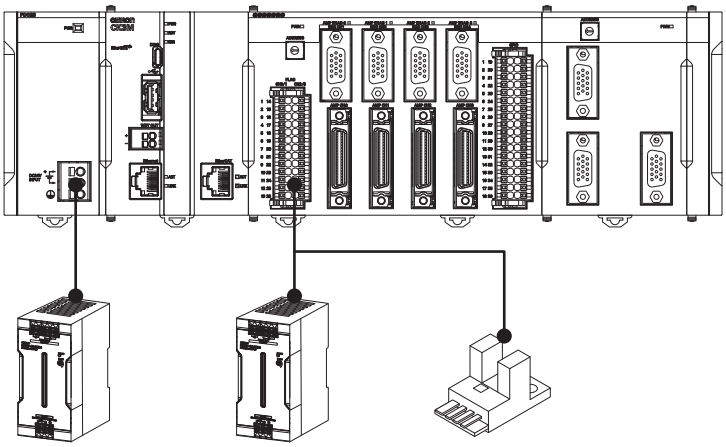
10	将右侧的程序添加到 plc1.plc 中。	`open plc PhaseSearch`
		
		`P0=Sys.Time+1.0;`
		`while(P0>Sys.Time){};`
		`Motor[1].PhaseFindingStep=1;`
		`while(Motor[1].PhaseFindingStep!=0){};`
		`Motor[2].PhaseFindingStep=1;`
		`while(Motor[2].PhaseFindingStep!=0){};`
		
		`disable plc PhaseSearch`
		
		`close`

11	右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。 • 如步骤 5、6 所示，通过在 MotorControl.pmh 中写入增益，可将增益作为程序下载到 PMAC。 • 如步骤 7、8 所示，通过在 pp_startup.txt 中写入相位搜索执行指令，在重新接通电源或重置后，将自动执行相位搜索，并将 Motor[1]、Motor[2]设为有效。	

3-5 设定超程限制开关和原点

3-5-1 各设备的配线

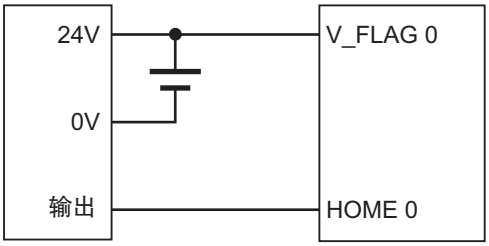
介绍 CK3M 和各种传感器的配线方法。



EE-SX674 分别连接到 CK3W-AX2323N 的以下 FLAG 端子。

- PLIM0: X 轴的正方向超程限制开关
- NLIM0: X 轴的负方向超程限制开关
- PLIM1: Y 轴的正方向超程限制开关
- NLIM1: Y 轴的负方向超程限制开关

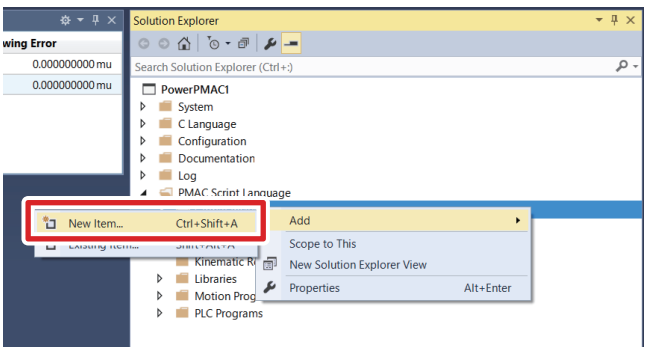
以下是 HOME0 的配线示例。



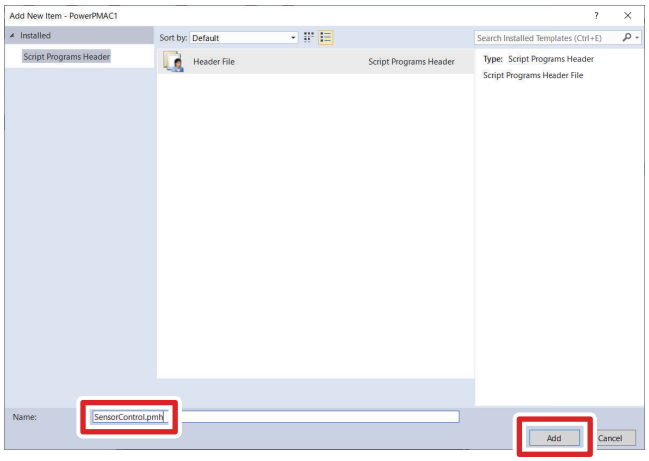
3-5-2 编程

进行超程限制开关的设定。

1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



2 在［Name（名称）］栏中输入「SensorControl.pmh」，单击［Add（添加）］按钮。



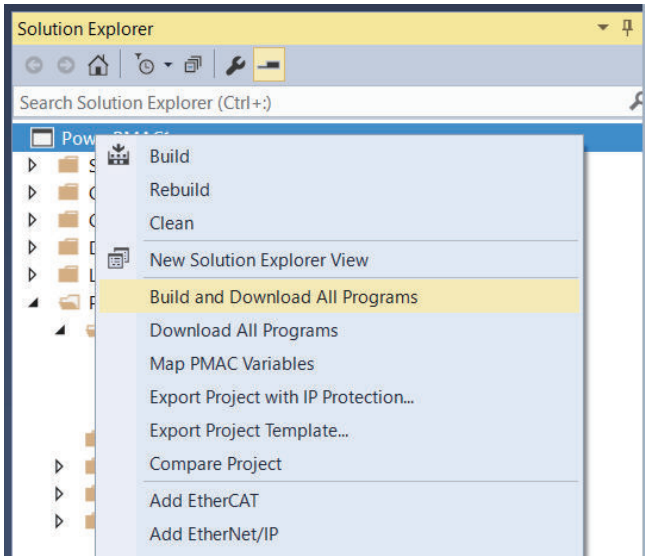
3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［SensorControl.pmh］。

4 将右侧的文本写入到 SensorControl.pmh 中。

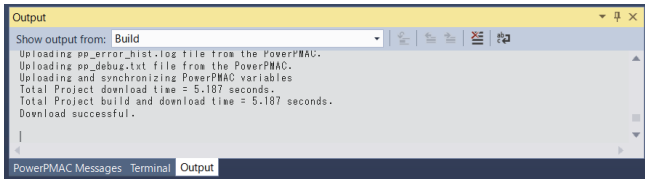
- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法 (P.4-1)」。

```
Sys.WpKey=$AAAAAAAA

//Overtravel Limit Switch Configurations
Motor[1].pLimits=Gate3[0].Chan[0].Status.a;//
*1
Motor[2].pLimits=Gate3[0].Chan[1].Status.a;//
*1
Motor[1].LimitBits=9;//*2
Motor[2].LimitBits=9;//*2
```

5 右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。

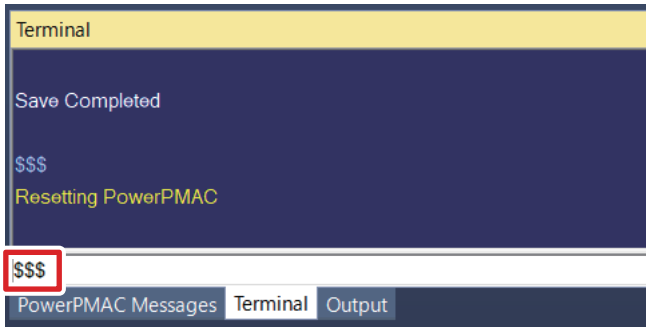


6 确认 Output（输出）没有异常。



7 在 Power PMAC IDE 的 Terminal（终端）中输入［save］指令。

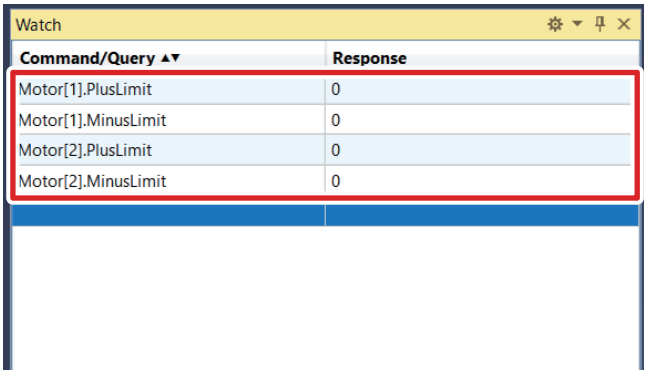
8 在 Power PMAC IDE 的 Terminal（终端）中输入［$$$］指令。



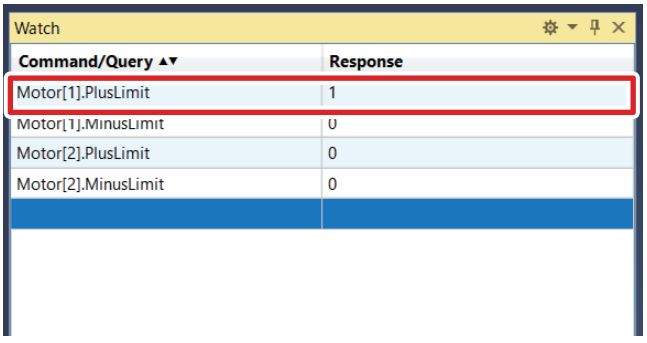
3-5-3 确认设定（超程限制开关）

确认超程限制开关的设定是否正常动作。

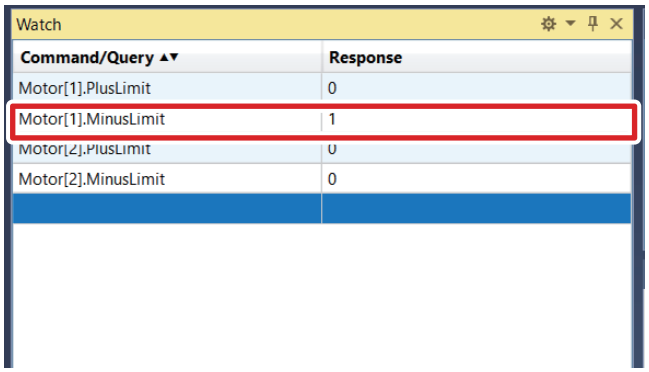
1 确认以下变量全部为 0。
- 「Motor[1].PlusLimit」
- 「Motor[1].MinusLimit」
- 「Motor[2].PlusLimit」
- 「Motor[2].MinusLimit」



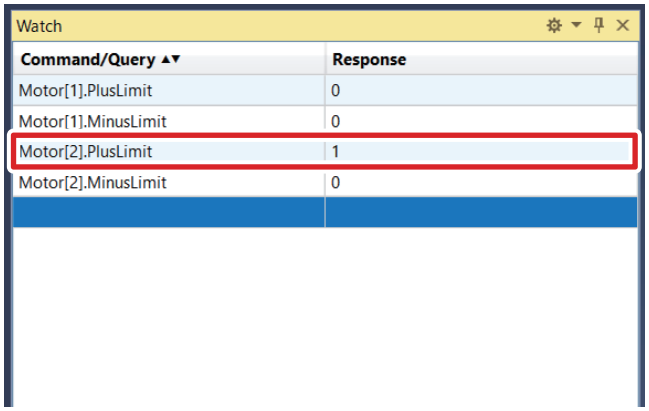
2 确认 X 轴的正方向超程限制开关「Motor[1].PlusLimit」为 1。



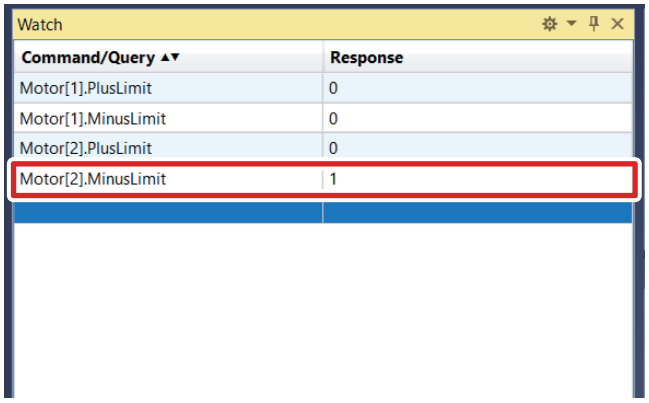
3 确认 X 轴的负方向超程限制开关「Motor[1].MinusLimit」为 1。



4 确认 Y 轴的正方向超程限制开关「Motor[2].PlusLimit」为 1。



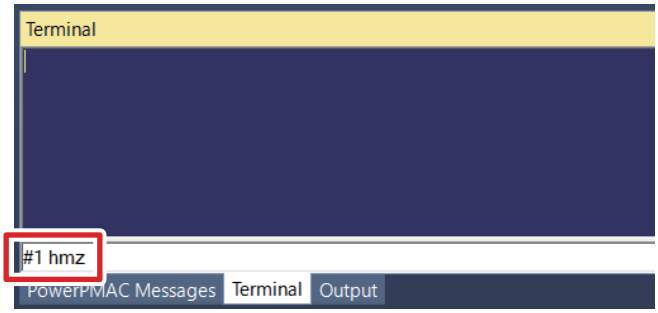
5 确认 Y 轴的负方向超程限制开关「Motor[2].MinusLimit」为 1。



3-5-4 设定原点

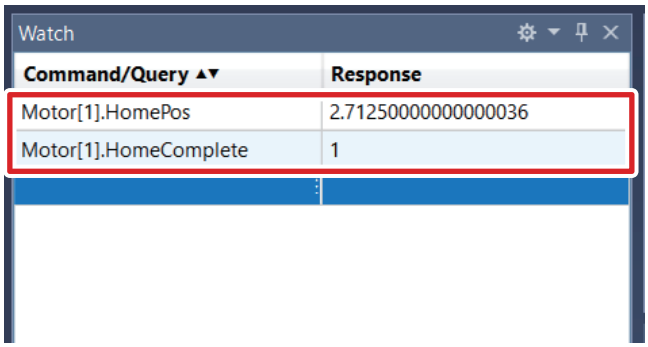
进行原点设定。

1 将 X 轴移动至原点位置，在 Power PMAC IDE 的 Terminal（终端）中输入［#1 hmz］指令。

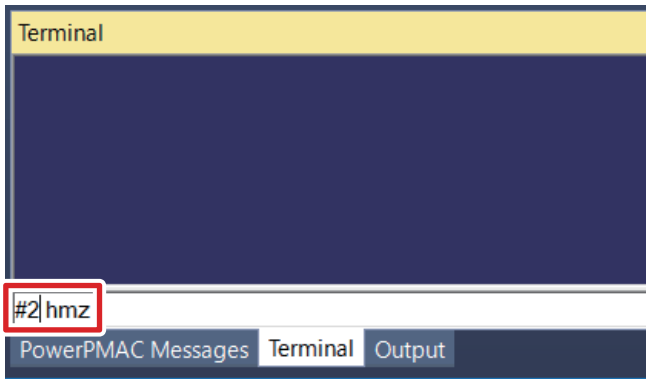


2 确认已正确反映以下变量。

- 「Motor[1].HomePos」：原点触发位置
- 「Motor[1].HomeComplete」：1

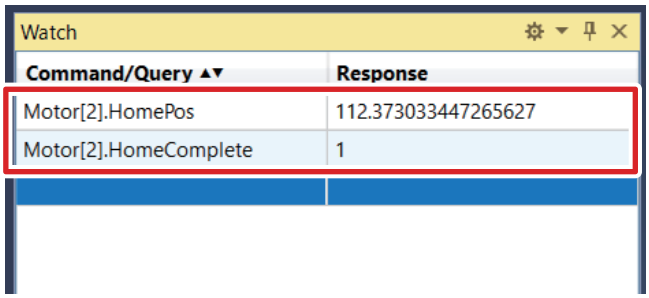


3 将 Y 轴移动至原点位置，在 Power PMAC IDE 的 Terminal（终端）中输入［#2 hmz］指令。



4 确认已正确反映以下变量。

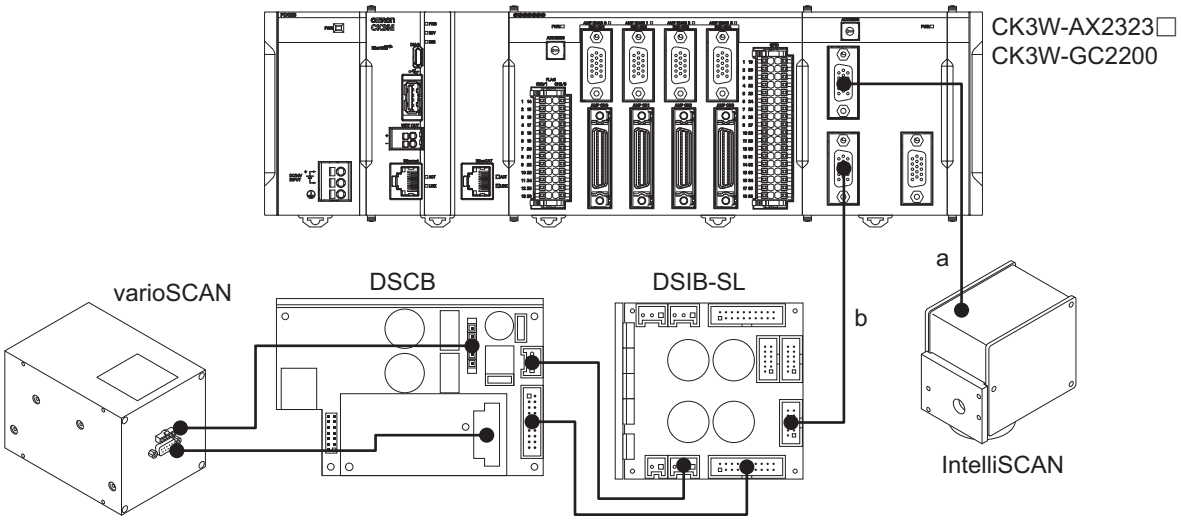
- 「Motor[2].HomePos」：原点触发位置
- 「Motor[2].HomeComplete」：1



3-6 设定扫描振镜

3-6-1 各设备的配线

介绍 CK3M 和扫描振镜的配线方法。



按照以下要领连接上图 a、b 位置。

a. 控制器和扫描振镜的连接
用专用电缆 CK3W-CAG03A 连接 CK3W-GC2200 的 XY 连接器和 intelliSCAN 的 SL2-100 连接器。

CK3W-GC2200			IntelliSCAN	
Signal	Pin		Signal	Pin
XY-IN+	1		DATA OUT+	5
XY-IN-	6		DATA-OUT-	9
XY-OUT+	2		DATA-IN+	1
XY-OUT-	7		DATA-IN-	6

b. 控制器和动态对焦单元的连接
用专用电缆 CK3W-CAG03A 连接 CK3W-GC2200 的 Z 连接器和 DSIB-SL 的 DIGITAL IN 连接器。

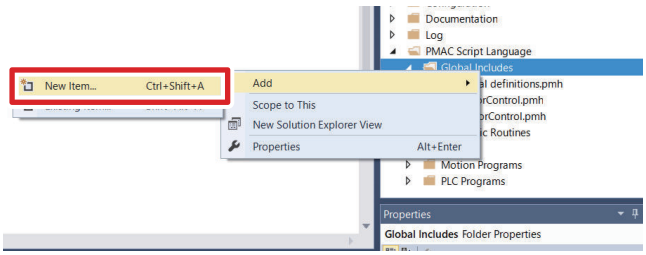
CK3W-GC2200	
Signal	Pin
XY-IN+	1
XY-IN-	6
XY-OUT+	2
XY-OUT-	7

DSIB-SL	
Signal	Pin
DATA OUT+	9
DATA-OUT-	8
DATA-IN+	1
DATA-IN-	2

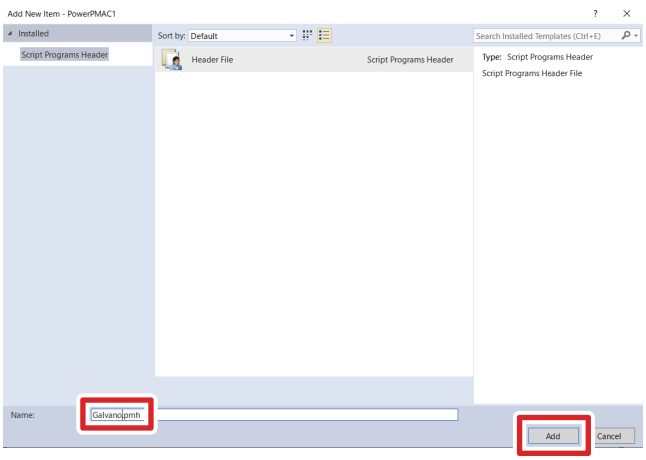
3-6-2 编程

对扫描振镜的设定进行编程。

1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



2 在［Name（名称）］栏中输入「Galvano.pmh」，单击［Add（添加）］按钮。



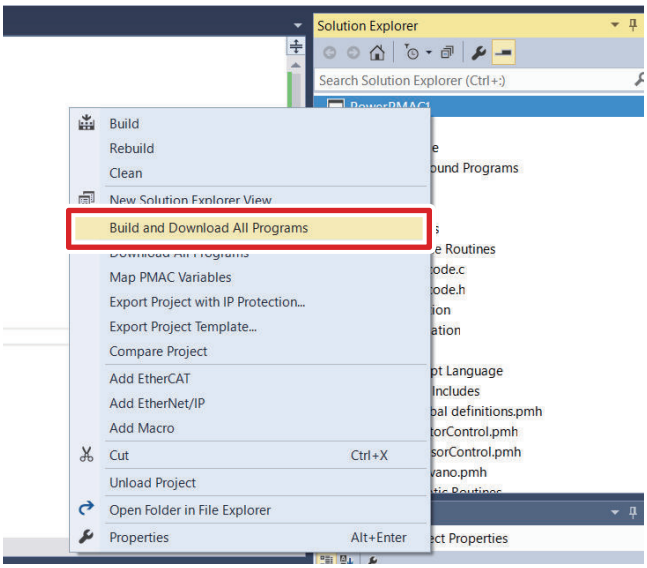
3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［Galvano.pmh］。

4 将右侧的文本写入到 Galvano.pmh 中。

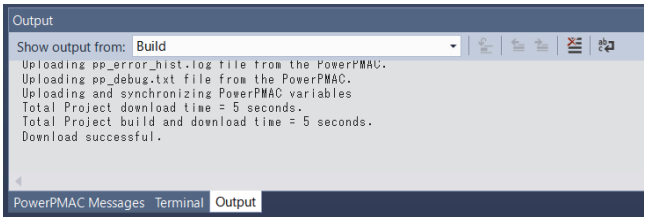
- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

```
Sys.WpKey=$AAAAAAAA

//SL2-100 Configurations
Gate3[1].SerialEncCtrl=$82800//*1

//Servo Task Configurations
Motor[3].ServoCtrl=1//*2
Motor[4].ServoCtrl=1//*2
Motor[5].ServoCtrl=1//*2
Motor[3].Ctrl=Sys.PosCtrl//*3
Motor[4].Ctrl=Sys.PosCtrl//*3
Motor[5].Ctrl=Sys.PosCtrl//*3
Motor[3].MaxSpeed=0;//*4
Motor[4].MaxSpeed=0;//*4
Motor[5].MaxSpeed=0;//*4

//Motor Output Configurations
Motor[3].MaxPos=0//*5
Motor[4].MaxPos=0//*5
Motor[5].MaxPos=0//*5
Motor[3].MinPos=0//*6
Motor[4].MinPos=0//*6
Motor[5].MinPos=0//*6
Motor[3].pDac=Gate3[1].Chan[0].Dac[0].a//*7
Motor[4].pDac=Gate3[1].Chan[1].Dac[0].a//*7
Motor[5].pDac=Gate3[1].Chan[2].Dac[0].a//*7
Motor[3].pLimits=0//*8
Motor[4].pLimits=0//*8
Motor[5].pLimits=0//*8
Motor[3].FatalFeLimit=0//*9
Motor[4].FatalFeLimit=0//*9
Motor[5].FatalFeLimit=0//*9

//Encoder Input Configurations
Motor[3].pEnc=EncTable[3].a//*10
Motor[4].pEnc=EncTable[4].a//*10
Motor[5].pEnc=EncTable[5].a//*10
Motor[3].pEnc2=EncTable[3].a//*11
Motor[4].pEnc2=EncTable[4].a//*11
Motor[5].pEnc2=EncTable[5].a//*11
EncTable[3].Type=1//*12
EncTable[4].Type=1//*12
EncTable[5].Type=1//*12
EncTable[3].pEnc=Gate3[1].Chan[0].SerialEncDataA.
a//*13
EncTable[4].pEnc=Gate3[1].Chan[1].SerialEncDataA.
a//*13
EncTable[5].pEnc=Gate3[1].Chan[2].SerialEncDataA.
a//*13
EncTable[3].ScaleFactor=1/4096//*14
EncTable[4].ScaleFactor=1/4096//*14
EncTable[5].ScaleFactor=1/4096//*14

//Enable Motor 6
Motor[6].ServoCtrl=1//*2
Motor[6].pDac=Sys.Udata[0].a
```

5 右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。



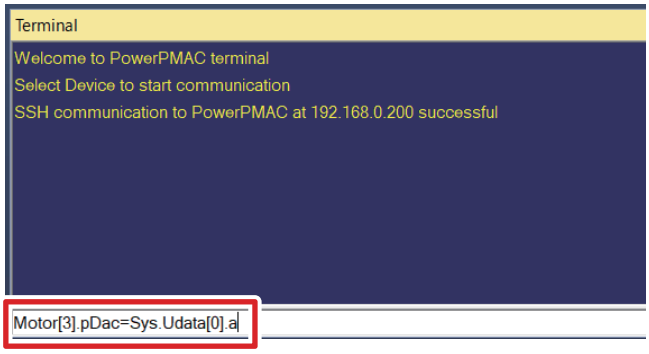
6 确认 Output（输出）没有异常。



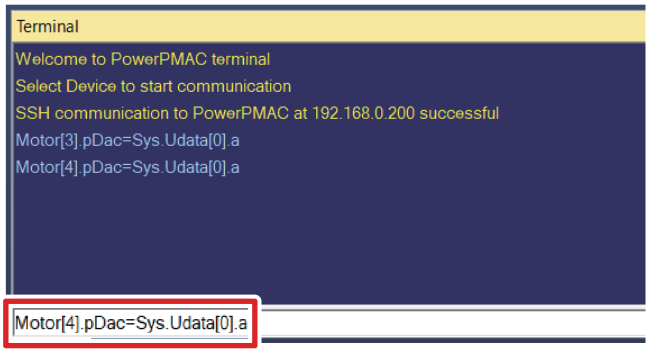
3-6-3 控制指令的设定

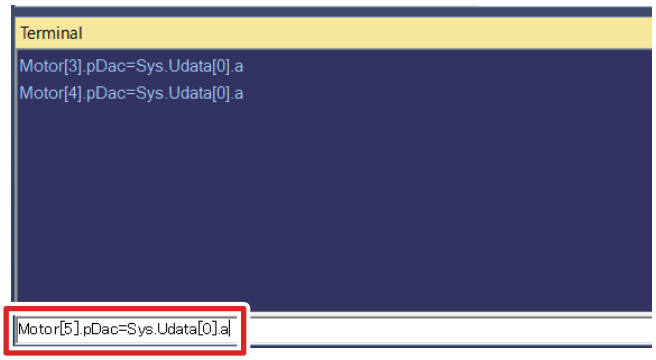
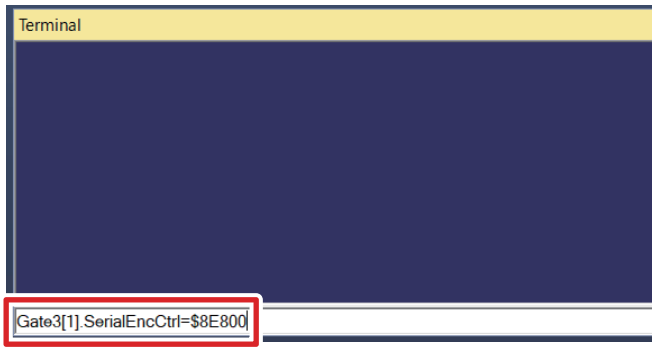
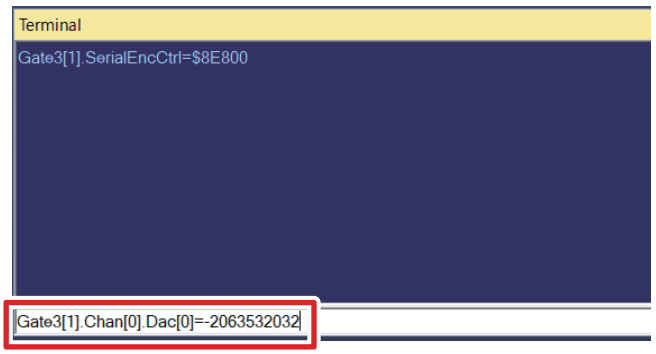
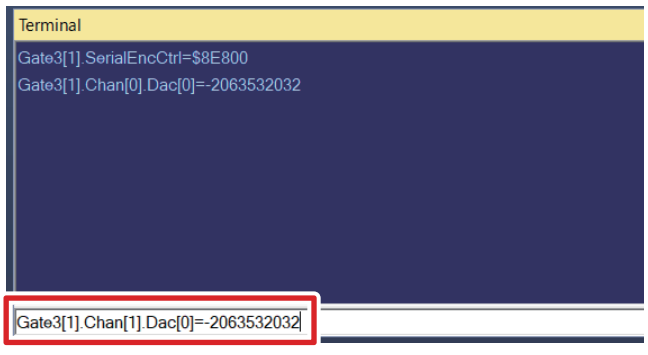
将扫描振镜的反馈信号设定为当前位置。

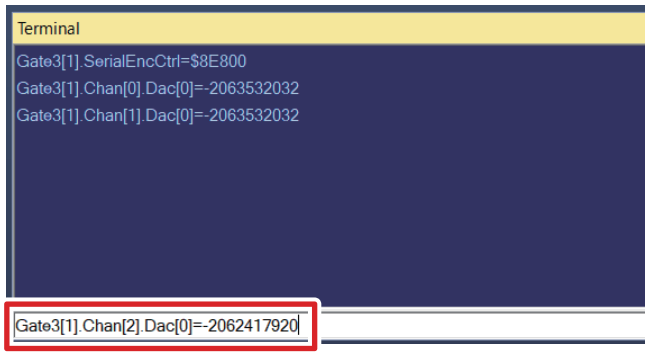
1 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[3].pDac=Sys.Udata[0].a］指令。



2 在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[4].pDac=Sys.Udata[0].a］指令。

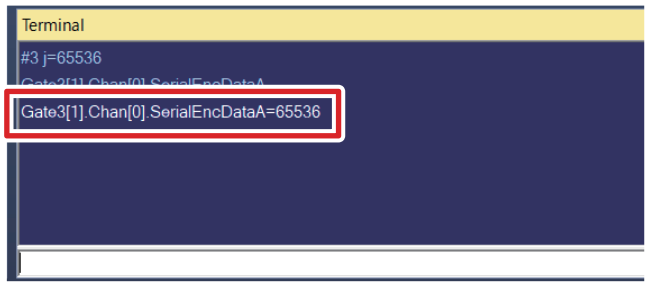


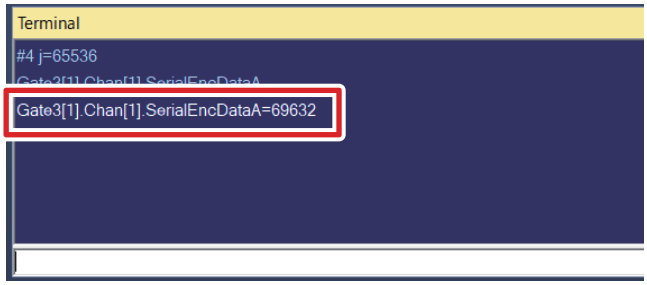
3	在 Power PMAC IDE 的 Terminal（终端）中输入［Motor[5].pDac=Sys.Udata[0].a］指令。	
4	在 Power PMAC IDE 的 Terminal（终端）中输入［Gate3[1].SerialEncCtrl=$8E800］指令。	
5	在 Power PMAC IDE 的 Terminal（终端）中输入［Gate3[1].Chan[0].Dac[0]=-2063532032］指令。	
6	在 Power PMAC IDE 的 Terminal（终端）中输入［Gate3[1].Chan[1].Dac[0]=-2063532032］指令。	

7	在 Power PMAC IDE 的 Terminal（终端）中输入［Gate3[1].Chan[2].Dac[0]=-2062417920］指令。	
8	在 Power PMAC IDE 的 Terminal（终端）中输入［$$$］指令。	

3-6-4 设定的确认

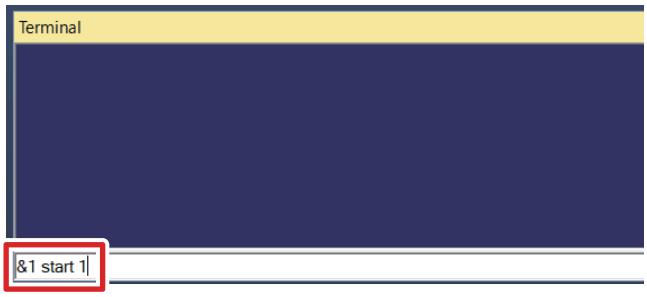
确认扫描振镜的设定是否正确。

1	在 Power PMAC IDE 的 Terminal（终端）中输入［#3 j=65536］指令。	
2	确认接近 Gate3[0].Chan[0].SerialEncDataA=65536。	

3	在 Power PMAC IDE 的 Terminal（终端）中输入［#4 j=65536］指令。	
4	确认接近 Gate3[1].Chan[1].SerialEncDataA=65536。	
5	在 Power PMAC IDE 的 Terminal（终端）中输入［#5 j=65536］指令。	
6	确认接近 Gate3[0].Chan[2].SerialEncDataA=65536。	

3-6-5 MOTF 的上机验证

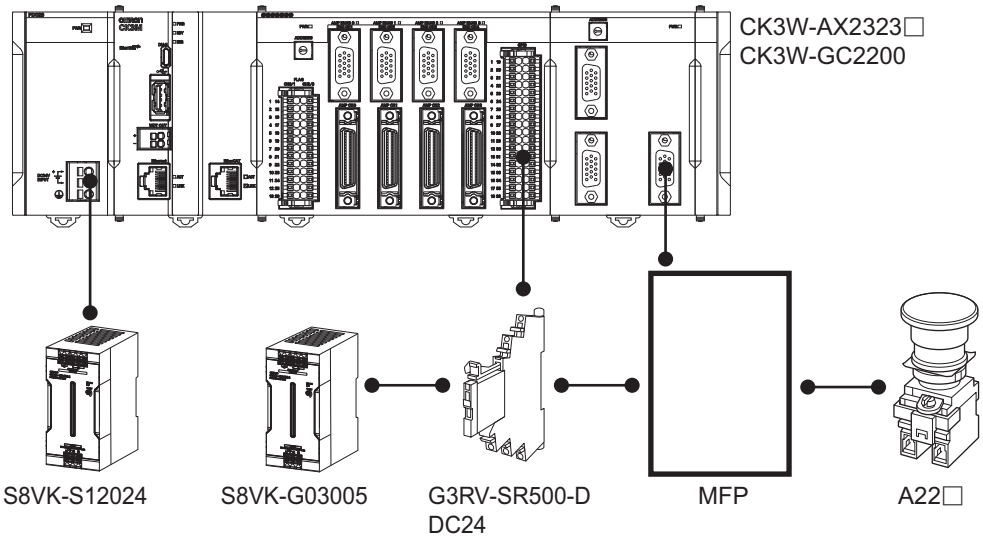
上机确认 MOTF 的设定是否正常动作。

1	在 Power PMAC IDE 的 Terminal（终端）中输入［&1 start 1］指令。	
2	确认 X、Y 轴的直线电机和扫描振镜同步动作。	

3-7 设定激光振荡器

3-7-1 各设备的配线

介绍 CK3M 和激光振荡器的配线方法。



分别按以下说明连接 CK3W-AX2323N 的 GPIO 端子和 MFP。

- OUT 0: MFP 的 PIN5
- OUT 1: MFP 的 PIN6
- OUT 2: MFP 的 PIN7
- OUT 3: MFP 的 PIN8
- OUT 4: MFP 的 PIN9
- OUT 5: MFP 的 PIN18
- OUT 6: MFP 的 PIN22

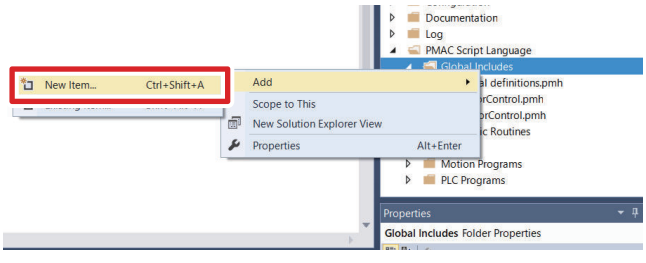
分别按以下说明连接 CK3W-GC2200 的 LASER 端子和 MFP。

- OUT 0: MFP 的 PIN20
- OUT 1: MFP 的 PIN21

3-7-2 引导激光的输出

确认引导激光是否正常输出。

1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



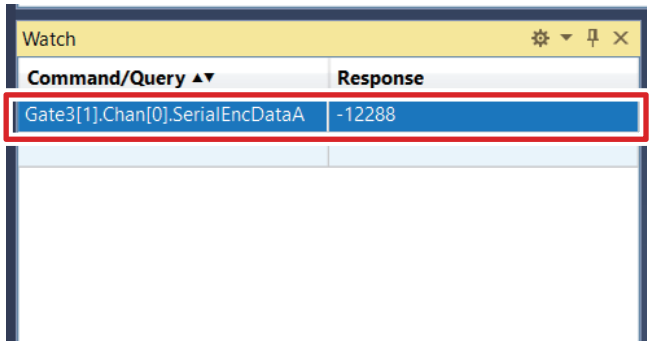
2 在［Name（名称）］栏中输入「LaserDefinitions.pmh」，单击［Add（添加）］按钮。

3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［LaserControl.pmh］，补充右侧所属的文本。

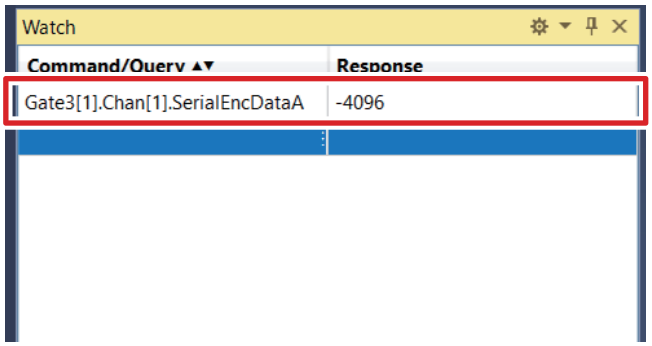
```
ptr LaserPower->Gate3[0].GpioData[0].16.4
ptr LaserLatch->Gate3[0].GpioData[0].20.1
ptr MasterOscillator->Gate3[0].GpioData[0].21
.1
ptr GuideLaser->Gate3[0].GpioData[0].22.1
```

4 在 Power PMAC IDE 的 Terminal（终端）中输入［GuideLaser=1］指令，确认有引导激光输出。

5 在 Power PMAC IDE 的 Terminal（终端）中输入［#3 j=0］指令，确认接近 Gate3[1].Chan[0].SerialEncDataA=0。



6 在 Power PMAC IDE 的 Terminal（终端）中输入［#4 j=0］指令，确认为 Gate3[1].Chan[1].SerialEncDataA=0。



7 确认 XY 滑台的中央有引导激光输出。

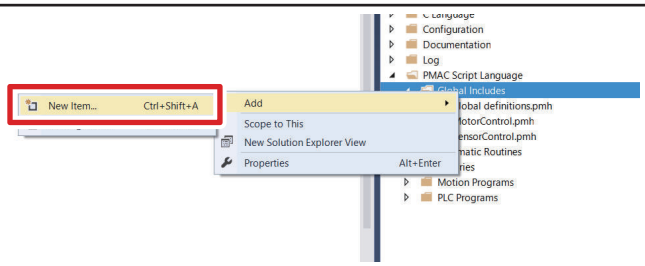
8 在 Power PMAC IDE 的 Terminal（终端）中输入［#5 j={位置}］指令，确认引导激光的光斑直径是否合适。

9 在 Power PMAC IDE 的 Terminal（终端）中输入［GuideLaser=0］指令，确认没有引导激光输出。

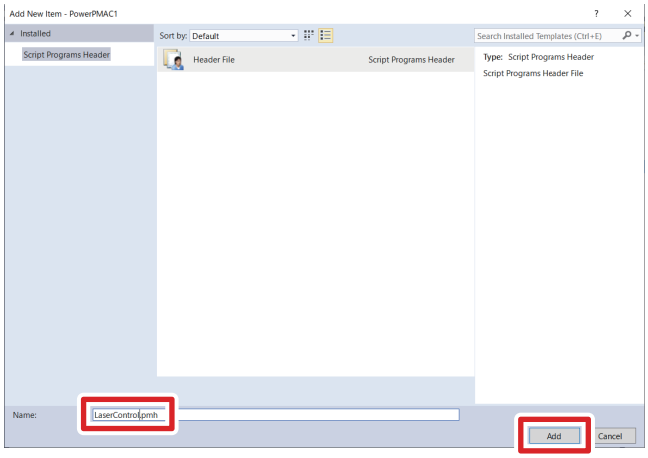
3-7-3 编程

对激光振荡器的设定进行编程。

1 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。



2 在［Name（名称）］栏中输入「LaserControl.pmh」，单击［Add（添加）］按钮。

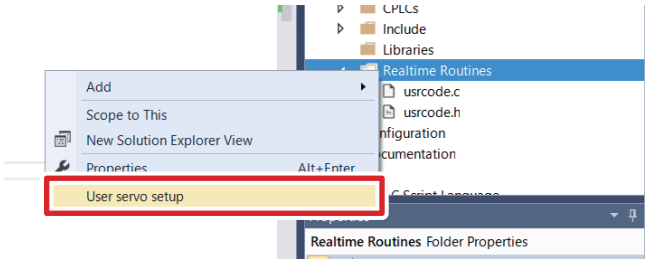


3 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］下面的［LaserControl.pmh］，补充右侧所属的文本。

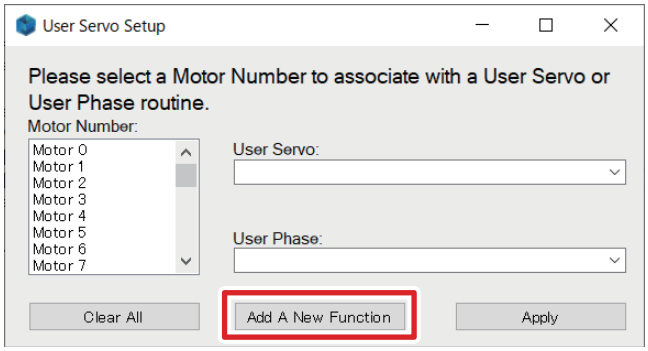
- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

```
Sys.WpKey=$AAAAAAAA

//Q-Switch Configurations
Gate3[1].Chan[0].CompA=$8000D000//*1
Gate3[1].Chan[1].CompA=$0//*2

//Laser Output Configurations
Gate3[1].Chan[2].CompA=$FFF00//*3
```

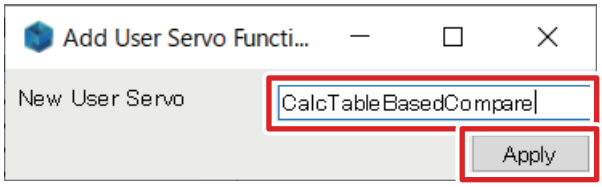
4 右击 Solution Explorer（解决方案资源管理器）的［C Language］–［Realtime Routines］，选择［User servo setup（用户伺服设置）］。



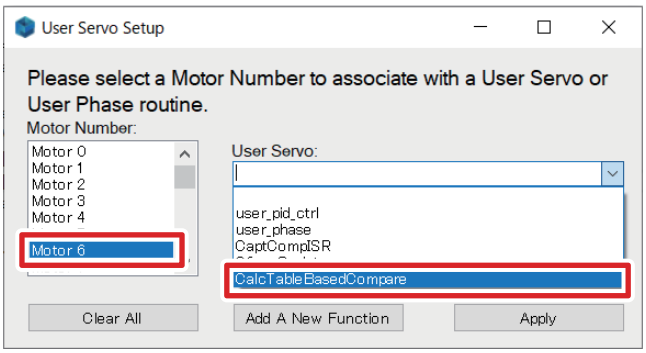
5 单击［Add a New Function（用户伺服设置）］按钮。



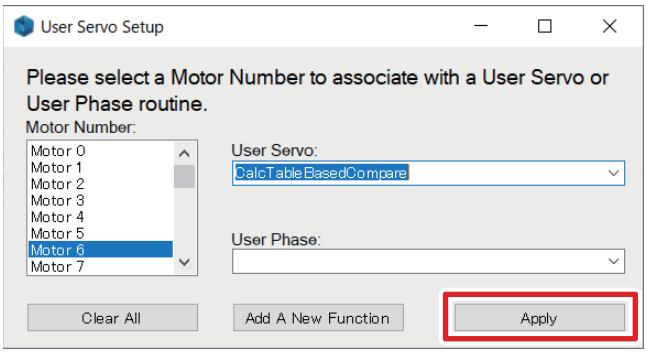
6 输入「CalcTableBasedCompare」，单击［Apply（应用）］按钮。



7 选择［Motor 6］，从［User Servo（用户伺服）］中选择［CalcTableBasedCompare］。



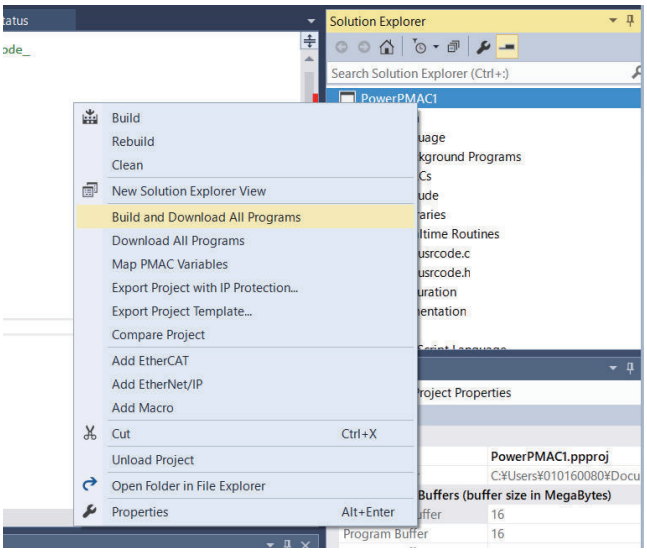
8 单击［Apply（应用）］按钮。



9 单击 Solution Explorer（解决方案资源管理器）的［C Language］–［Realtime Routines］–［usercode.c］，添加右侧的函数。

```
static double X_old=0;
static double Y_old=0;
static double SumDistance=0;

double CalcTableBasedCompare(struct MotorData*Mpt
r)
{
 volatile GateArray3 *MySecondGate3IC;
 MySecondGate3IC=GetGate3MemPtr(1);

 double X_now;
 X_now=pshm->Motor[1].DesPos+pshm->Motor[3].DesPo
s;
 double Y_now;
 Y_now=pshm->Motor[2].DesPos+pshm->Motor[4].DesPo
s;
 double distance;
 distance=sqrt((X_old-X_now)*(X_old-X_now)+(Y_old
-Y_now)*(Y_old-Y_now));
 if(DistanceCountOn)SumDistance+=distance;

 X_old=X_now;
 Y_old=Y_now;
 MySecondGate3IC->Chan[0].CompB=SumDistance;

 return 0;
}
```

10 右击 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Global Includes］，从［Add（添加）］中选择［New Item（添加项）］。然后，在［Name（名称）］栏中输入「TCRConfigurations.pmh」，单击［Add（添加）］按钮。

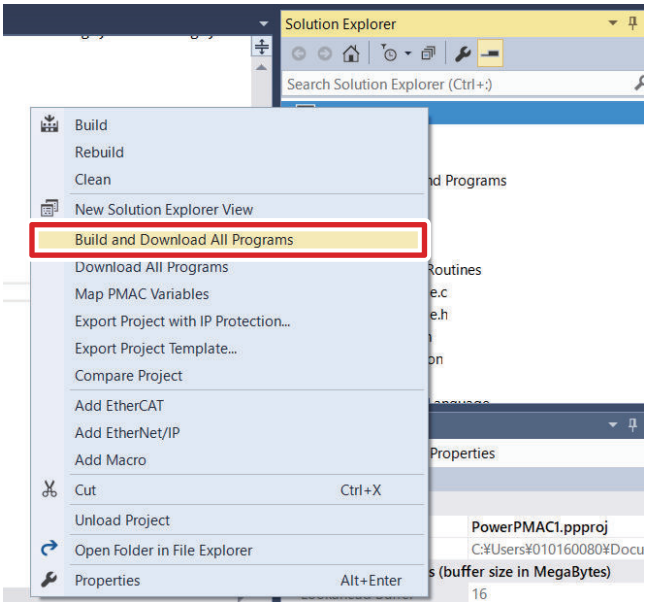


11 将右侧的文本写入到 TCRConfigurations.pmh 中。

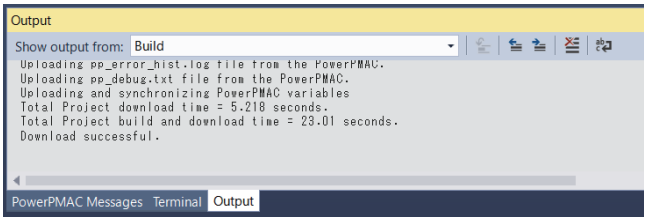
- 关于带注释设定的内容，请参考「第 4 章 各种设定的自定义方法(P.4-1)」。

```
global DistanceCountOn
Gate3[1].Chan[1].CompB=50.0*1.414//*1
Gate3[1].Chan[1].CompB=50.0*1.414+400//*2
//Gate3[1].Chan[1].CompB=****
Gate3[1].Chan[2].CompB=$80000000//*3
```

12 右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。



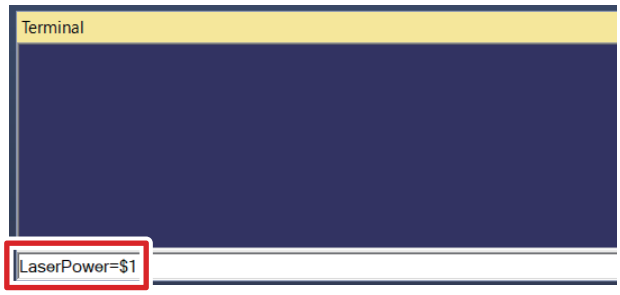
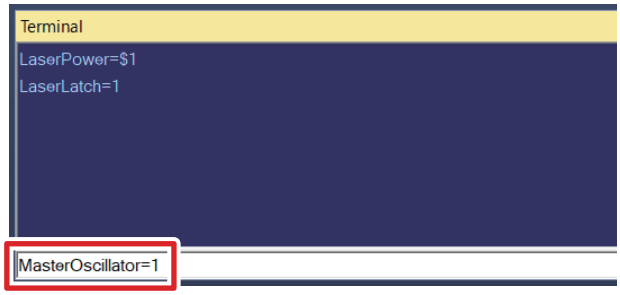
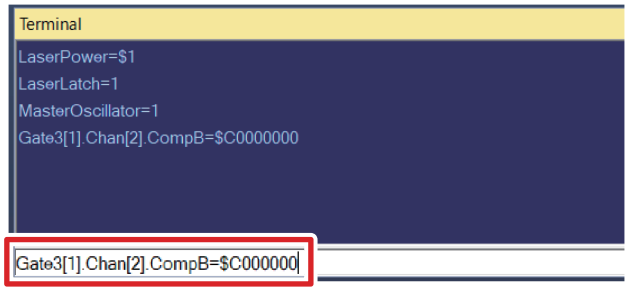
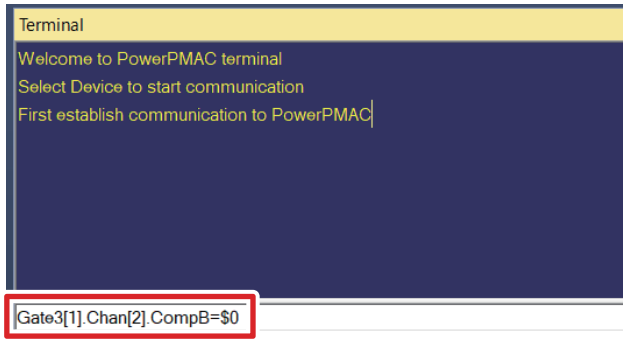
13 确认 Output（输出）没有异常。



3-7-4 设定激光振荡器

进行激光振荡器的功率设定和激光输出所需的事先准备。

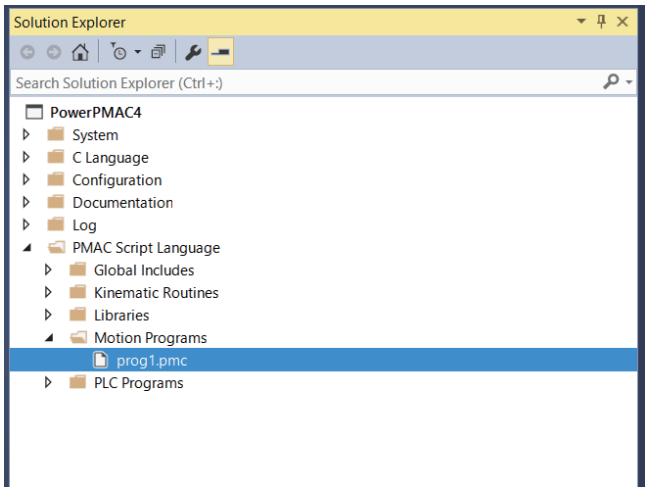
1 确认紧急停止按钮开关处于 OFF 状态。

2	在 Terminal（终端）中输入［LaserPower=$1］指令。	
3	在 Terminal（终端）中输入［LaserLatch=1］指令。	
4	在 Terminal（终端）中输入［MasterOscillator=1］指令。	
5	在 Terminal（终端）中输入［Gate3[1].Chan[2].CompB=$C000000］指令。	
6	确认有激光输出。	
7	在 Terminal（终端）中输入［Gate3[1].Chan[2].CompB=$0］指令。	

3-7-5 设定的确认

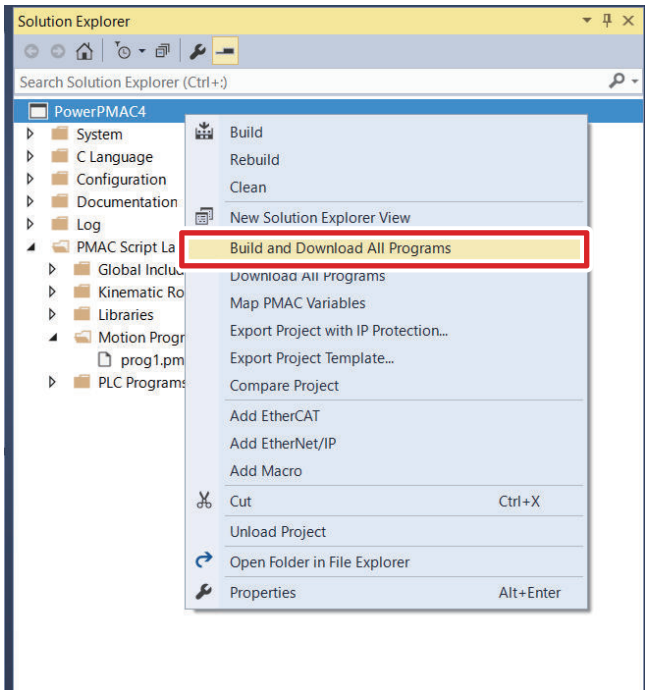
通过是否可以实际进行加工，确认是否已正确按本书进行设定。

1 打开 Solution Explorer（解决方案资源管理器）的［PMAC Script Language］–［Motion Programs］下面的［prog1.pmc］。



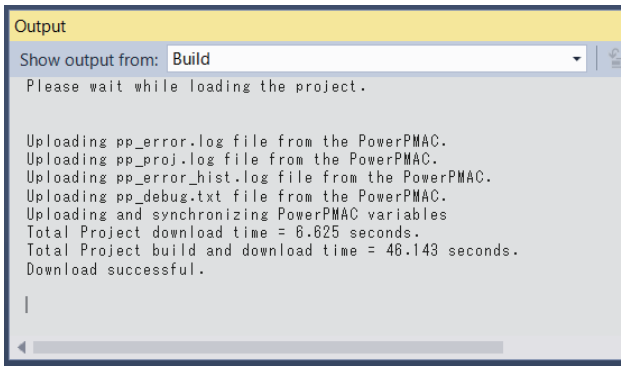
2 将加工时使用的程序写入到 prog1.pmc 中。

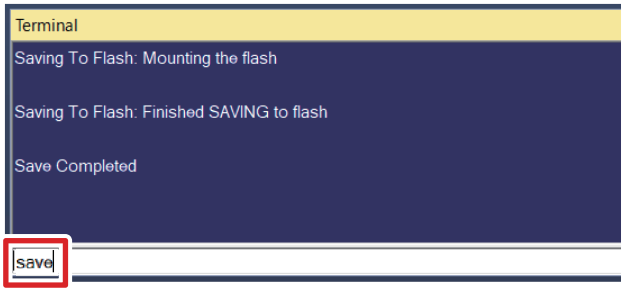
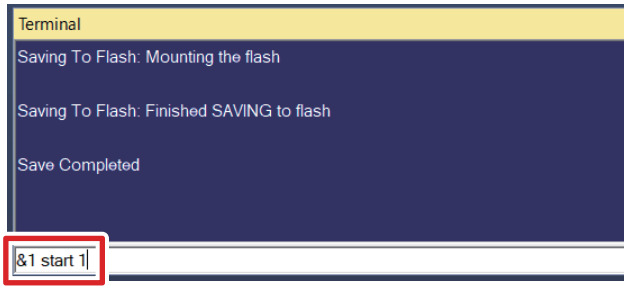
3 右击 Power PMAC IDE 画面右上方 Solution Explorer（解决方案资源管理器）的项目名称，选择［Build and Download All Programs（构建并下载所有程序）］，执行链接&下载。



4 确认 Output（输出）没有异常。

- 传送失败时，请通过 Output（输出）确认错误的内容。



5	在 Power PMAC IDE 的 Terminal（终端）中输入［save］指令。 • 结束后，将在 Terminal（终端）中显示“Save Completed”。	
6	在 Power PMAC IDE 的 Terminal（终端）中输入［&1 start 1］指令。	
7	确认能正确加工。	

4

各种设定的自定义方法

本章介绍各种设定的自定义方法。

4-1 globaldefinitions.pmh

各种时钟或任务周期的设定文件。

No.	参数	说明
*1	Coord[1].SegMoveTime	以 ms 为单位指定圆弧插补等轨迹的插补时间。此值越小，越接近准确的轨迹，但计算量会增大，CPU 的计算负载将变大。
*2	Gate3[0].PhaseFreq	指定相位时钟的频率。此值越大，电流回路等处理的精度越高，但计算量会增大，CPU 的计算负载将变大。
*3	Gate3[0].ServoClockDiv	指定伺服时钟的分频比。此值越小，位置回路等处理的精度越高，但计算量会增大，CPU 的计算负载将变大。 伺服时钟频率将根据本设定值变为以下值。 伺服时钟频率 =相位时钟频率/(Gate3[0].ServoClockDiv+1)
*4	Sys.PhaseOverServoPeriod	设定以下所示的值。 Sys.PhaseOverServoPeriod =1/(Gate3[0].ServoClockDiv+1)
*5	Sys.ServoPeriod	设定以下所示的值。 Sys.ServoPeriod =1000*(Gate3[0].ServoClockDiv+1)/Gate3[0].PhaseFreq
*6	Gate3[x].PhaseServoDir	指定由哪个单元提供相位时钟和伺服时钟。

4-2 MotionOnTheFly.pmh

MOTF 功能相关的设定文件。

No.	参数	说明
*1	KdgainX	指定 X 轴上用于 MOTF 的过滤器系数。此值越小，扫描振镜的动作范围越大。通过将此值设定为足够小的值，尽可能扩大扫描振镜的控制范围，可以获得更快的响应。
*2	KdgainY	指定上述 Y 轴的过滤器系数。
*3	GalvoSfX	指定相对于 X 轴中激光移动距离（指令单位系统）的振镜电机指令位置（位）。（单位: 位/指令单位）
*4	GalvoSfY	指定上述 Y 轴的值。
*5	&1 #1->I; #2->I; #3->I; #4->I;	将轴 1、2、3 和 4 分配至坐标系 1 作为反向运动轴。要变更轴编号时或要将其他轴分配到坐标系时，请添加此部分。

4-3 MotorControl.pmh

直线电机控制相关的设定文件。

No.	参数	说明
*1	Motor[x].ServoCtrl	将执行伺服控制的 x 轴设定为 1。本书中使用 6 轴，要使用更多的轴时，请将 Motor[y].ServoCtrl(y>6)也设定为 1。
*2	Motor[x].pEnc, Motor[x].pEnc2	指定要参考的编码器表，作为 x 轴的回路反馈。在本书中，分配了与轴具有相同索引的编码器表。
*3	Motor[x].EncType	指定 x 轴的编码器种类。本书以正弦波编码器为对象，因此指定为 6。
*4	Motor[x].PosUnit	指定 x 轴的指令单位。本书中以 mm 为单位，因此设定为 3。
*5	EncTable[x].Type	指定编码器表的计算方式。在本书中，为了读取 32 位反馈值，所以设定为 1。
*6	EncTable[x].pEnc	指定编码器表 x 参考的寄存器。本书中获取由 CK3W-AX2323N 单元计算得到的反馈值。
*7	EncTable[x].ScaleFactor	指定将 CK3W-AX2323N 单元（24 位）的数据获取至编码器表（32 位）时的转换系数 $1/2^8$。
*8	Motor[x].AmpFaultlevel	设定将放大器故障设为高电平有效还是低电平有效。本书中设定为高电平有效。
*9	Motor[x].FatalFeLimit	指定在 x 轴上停止运行的偏差量。要提高安全性，请设定为较小的值。反之，如果发生偏差异常而无法开始运行，则请设定为较大的值。
*10	Motor[x].PhaseCtrl	将执行电流回路的 x 轴设定为 4。在本书中，XY 滑台上使用 Direct PWM 控制方式，要使用模拟电压控制或 EtherCAT 控制等时，请设定为 0。
*11	Motor[x].pPhaseEnc	指定用于获取电机反馈位置的寄存器。
*12	Motor[x].PhasePosSf	指定将 No.11 获取的当前位置转换为电角（0～2048）时的转换系数。具体按以下公式计算。 $PhasePosSf = \frac{2048 \times A}{B}$ 这里，A 指定为 Gate3[0].Chan[x].PhaseCapt 的每个脉冲的长度。由于 LIDA48 为每个正弦波周期 0.02mm，且 PMAC 的插值分割数为 16384，因此指定为 0.02/16384。 此外，B 指定为电机的电角旋转 1 圈的长度。本书中使用的直线电机其 ECL(Electrical Cycle Length)为 36mm，因此设定为 36。
*13	Motor[x].PhaseOffset	指定各相电机的电角（0～2048）相位差。使用三相电机时，为 683 或-683。
*14	Motor[x].pAdc	指定用于读取来自伺服放大器的反馈电流值的寄存器。本书中使用从 Direct PWM 接口读取的值。
*15	Motor[x].AdcMask	指定使用 No.14 寄存器中的哪一个位作为反馈值。该值根据 AD 变流器的规格进行设定，该 AD 变流器安装在伺服放大器上，用于读取电流值。
*16	Motor[x].pDac	指定写入控制器指令输出的寄存器的起始地址。这里的输出位置为 CK3W-AX2323N 的 U 相/V 相/W 相输出用寄存器，所以指定为开头的 U 相(Gate3[0].Chan[0].Pwm[0])。
*17	Motor[x].PhaseFindingDac	指定进行相位基准搜索时使用的输出的大小。如果相位基准搜索无法完成，可能是因为输出不足，请增大该值。此外，如果在相位基准搜索中出现 I2TFault 状态异常，则输出电流超过额定值，因此请减小该值。

No.	参数	说明
*18	Motor[x].PhaseFindingTime	指定相位基准搜索的保持时间的大小。如果相位基准搜索无法完成，请将该值设定为较大的值。此外，如果在相位基准搜索中出现 I2TFault 状态异常，则输出电流超过额定值，因此请减小该值。
*19	Motor[x].PwmSf	指定 Direct PWM 输出的系数。满量程为 16384。通常设定为小于满量程的 95%，以便 PWM 波形的占空比不会达到 100%。 请根据使用的伺服放大器的规格设定本参数。
*20	Motor[x].MaxDac	指定电机及伺服放大器的连续额定电流值。请按以下公式计算。 $MaxDac=\frac{\cos 30°\times 32767\times \text{连续额定电流}}{\text{反馈电流的满量程电流}}$ 由于这次使用的设备在 X 轴和 Y 轴上的伺服放大器的连续额定电流都较小，因此使用伺服放大器的连续额定电流。 CDHD-0032APB0: 9A rms INC-5716D0-0S: 9.88A rms INC-5723D0-0S: 14.82A rms 此外，根据 CDHD-0032APB0 的规格，反馈电流的满量程采用 11.25Arms。
*21	Motor[x].I2tSet	指定电机及伺服放大器的短时额定电流值。请按以下公式计算。 $I2tSet=\frac{\cos 30°\times 32767\times \text{短时间额定电流}}{\text{反馈电流的满量程电流}}$ 由于这次使用的设备在 X 轴和 Y 轴上的伺服放大器的连续额定电流都较小，因此使用伺服放大器的短时额定电流值。 CDHD-0032APB0: 3A rms INC-5716D0-0S: 3.08A rms INC-5723D0-0S: 4.62A rms
*22	Motor[x].I2tTrip	指定电机及伺服放大器的短时额定电流值的容许时间。请按以下公式计算。根据 CDHD-0032APB0 的规格，容许时间设为 2 秒。 $I2tTrip=(MaxDac^2+IdCmd^2-I2tSet^2)\times$容许时间（秒）
*23	Gate3[0].Chan[x].PwmDeadTime	将 PWM 信号的死区时间指定为 800 纳秒。死区时间按以下公式计算。 DeadTime=0.0533μs×Gate3[0].Chan[x].PwmDeadTime 请根据伺服放大器的规格设定本值。
*24	Gate3[0].Chan[x].PwmFreqMult	将 PWM 输出频率设定为 15kHz。PWM 频率按以下公式计算。 $f_{PWM}=\frac{PwmFreqMult+1}{2}\times \text{相位频率}$ 请根据伺服放大器的最高输入频率，在 40kHz 以下的范围内设定 PWM 频率。
*25	Gate3[0].Chan[x].PackOutData	为了分两步完成对 U/V/W 相输出寄存器的写入，并实现高效的算法，将压缩格式设定为有效。 使用 CK3W-AX2323N 时，请勿变更该值。
*26	Gate3[0].AdcAmpStrobe	指定 ADC Strobe Word（在 Direct PWM 规格中，从控制器发送至伺服放大器的控制指令）。请根据伺服放大器的规格变更本值。
*27	Gate3[0].AdcAmpHeaderBits	指定反馈电流值数据中包含的标题位数。请根据伺服放大器的规格变更本值。
*28	Gate3[0].AdcAmpClockDiv	指定通过伺服放大器读取电流数据的 AD 变流器的时钟信号频率。
*29	Gate3[0].Chan[x].PackInData	为了分两步完成从 U/V/W 相输入寄存器的读取，并实现高效的算法，将压缩格式设定为有效。 使用 CK3W-AX2323N 时，请勿变更该值。

No.	参数	说明
*30	Gate3[0].Chan[x].AtanEna	使用正弦波编码器时设为有效。使用脉冲编码器或串行编码器时请设定为 0。
*31	Gate3[0].Chan[x].EncCtrl	设定编码器的解码方式。本书中，设定为 4 倍递增 AB 相 CW 解码。
*32	Motor[x].PosSf, Motor[x].Pos2Sf	指定轴接口单元计算的反馈值与指令位置之间的转换系数。 使用正弦波编码器时，脉冲数为每周期 16834，此外，使用的正弦波编码器为每周期 0.020mm，因此设定为 0.020/16384。

4-4 SensorControl.pmh

超程限制开关相关的设定文件。

No.	参数	说明
*1	Motor[x].pLimits	指定超程限制开关的参照处寄存器。
*2	Motor[x].LimitBits	指定要参照超程限制开关的参照处寄存器(No.1)中的哪个位。

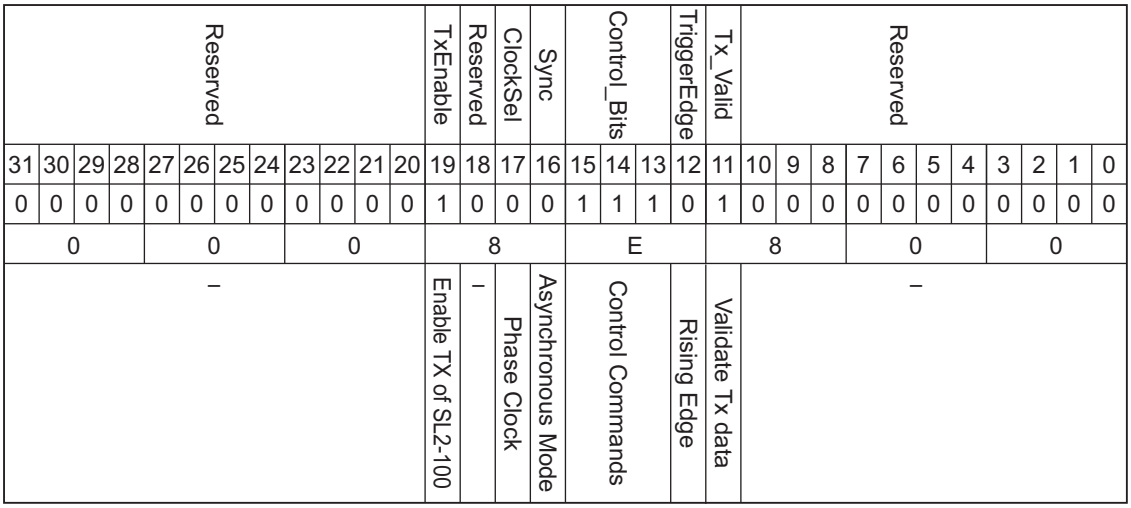
4-5 Galvano.pmh

扫描振镜的控制相关的设定文件。

No.	参数	说明
*1	Gate3[1].SerialEncCtrl	按下图所示设定。
*2	Motor[x].ServoCtrl	将执行伺服控制的 x 轴设定为 1。本书中使用 6 轴，要使用更多的轴时，请将 Motor[y].ServoCtrl(y>6)也设定为 1。
*3	Motor[x].Ctrl	在振镜控制中，为了进行位置控制，需设定 Sys.PosCtrl。
*4	Motor[x].MaxSpeed	指定振镜控制的最大速度。这里为了将最大速度监视设为无效，所以指定为 0。
*5	Motor[x].MaxPos	指定振镜控制的指令位置的最大值。这里为了设为无效，所以指定为 0。
*6	Motor[x].MinPos	指定振镜控制的指令位置的最小值。这里为了设为无效，所以指定为 0。
*7	Motor[x].pDac	指定写入控制器指令输出的寄存器的起始地址。这里的输出位置为 CK3W-GC2200 的 XY 输出用寄存器，所以指定为 Gate3[1].Chan[x].Dac[0]。
*8	Motor[x].pLimits	设定振镜控制的超程限制。这里为了设为无效，所以指定为 0。
*9	Motor[x].FatalFeLimit	指定在 x 轴上停止运行的偏差量。在扫描振镜上，为了将此设定设为无效，所以指定为 0。
*10	Motor[x].pEnc	指定要参考的编码器表，作为 x 轴的位置回路反馈。在本书中，分配了与轴具有相同索引的编码器表。
*11	Motor[x].pEnc2	指定要参考的编码器表，作为 x 轴的速度回路反馈。在本书中，分配了与轴具有相同索引的编码器表。
*12	EncTable[x].Type	指定编码器表的计算方式。在本书中，为了读取 32 位反馈值，所以设定为 1。
*13	EncTable[x].pEnc	指定编码器表 x 参考的寄存器。在本书中，输出有 CK3W-GC2200 单元计算的直线插补值。
*14	EncTable[x].ScaleFactor	指定将 SL2-100 数据（20 位）获取至编码器表（32 位）时的转换系数 $1/2^{12}$。

Reserved												TxEnable	Reserved	ClockSel	Sync	Control_Bits			TriggerEdge	Tx_Valid	Reserved										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0				0				0				8				2				8				0				0			
–												Enable TX of SL2-100	–	Phase Clock	Asynchronous Mode	Command Position			Rising Edge	Validate Tx data	–										

Command Position时



Control Commands时

4-6 LaserControl.pmh

激光振荡器的控制相关的设定文件。

No.	参数	说明
*1	Gate3[1].Chan[0].CompA	按如下所示进行设定。
*2	Gate3[1].Chan[1].CompA	设定相对于 SL2-100 通信数据的 PMAC 内部时钟的延迟时间。本书中设定为 0。
*3	Gate3[1].Chan[2].CompA	按如下所示进行设定。

DutyCycle												PWMPeriod												Reserved							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8				0				0				0				D				0				0				0			
Duty Ratio=50%												PWM Frequency=30kHz												–							

Gate3[1].Chan[0].CompA的设定

Reserved												PulseCount												Reserved							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0				0				0				F				F				F				0				0			
												Continuous Pulse Outputs																			

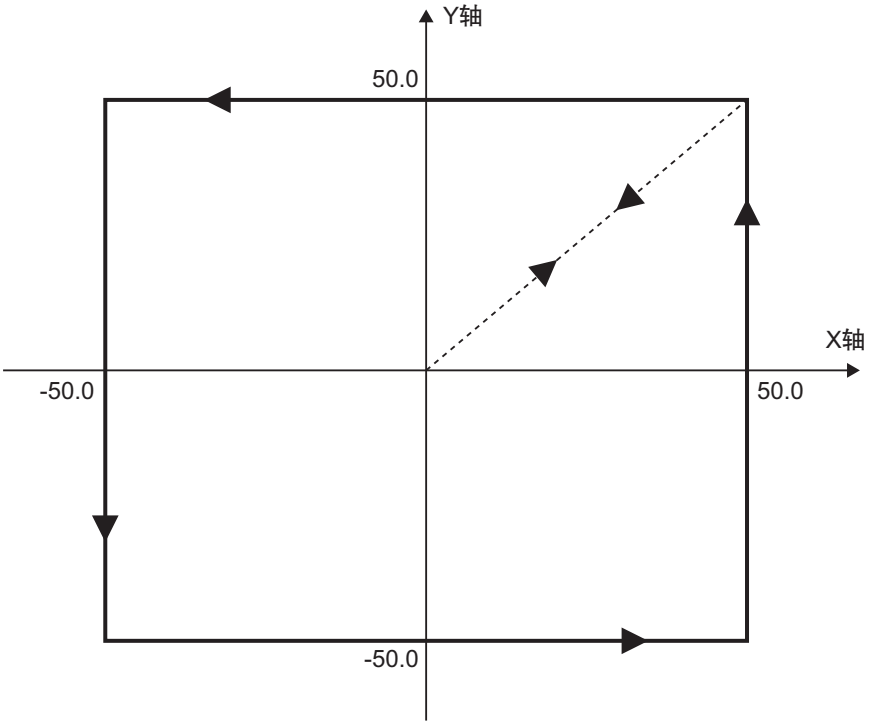
Gate3[1].Chan[2].CompA的设定

4-7 TCRConfigurations.pmh

激光的 ON/OFF 条件设定文件。

No.	参数	说明
*1	Gate3[1].Chan[1].CompB	设定激光从 OFF 变为 ON 期间的里程数。
*2	Gate3[1].Chan[1].CompB	设定激光从 ON 变为 OFF 期间的里程数。 之后，添加激光输出的切换条件。
*3	Gate3[1].Chan[2].CompB	按如下所示进行设定。

本书中介绍的前提条件如下所示，是以激光从原点输出到正方形上，然后再返回原点为条件的。



CompareEnable	ClearTable	CompClkSel	Reserved	CompOutWrite	CompOutPol	Reserved
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8				0				0				0				0				0				0				0			
Compare Enabled	Normal	Servo Clock	–	Normal	H active	–

Gate3[1].Chan[2].CompB的设定

购买欧姆龙产品的客户须知

承诺事项

承蒙对欧姆龙株式会社 (以下简称 “ 本公司 ”) 产品的一贯厚爱和支持,藉此机会再次深表谢意。

如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

1. 定义

本承诺事项中的术语定义如下。

(1) “ 本公司产品 ”: 是指 “ 本公司 ” 的 FA 系统机器、通用控制器、传感器、电子 / 结构部件。

(2) “ 产品目录等 ”: 是指与 “ 本公司产品 ” 有关的欧姆龙综合产品目录、FA 系统设备综合产品目录、安全组件综合产品目录、电子 / 机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。

(3) “ 使用条件等 ”: 是指在 “ 产品目录等 ” 资料中记载的 “ 本公司产品 ” 的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。

(4) “ 客户用途 ”: 是指客户使用 “ 本公司产品 ” 的方法,包括将 “ 本公司产品 ” 组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。

(5) “ 适用性等 ”: 是指在 “ 客户用途 ” 中 “ 本公司产品 ” 的 (a) 适用性、(b) 动作、(c) 不侵害第三方知识产权、(d) 法规法令的遵守以及 (e) 满足各种规格标准。

2. 关于记载事项的注意事项

对 “ 产品目录等 ” 中的记载内容,请理解如下要点。

(1) 额定值及性能值是在单项试验中分别在各条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。

(2) 提供的参考数据仅作为参考,并非可在该范围内一直正常运行的保证。

(3) 应用示例仅作参考,不构成对 “ 适用性等 ” 的保证。

(4) 如果因技术改进等原因,“ 本公司 ” 可能会停止 “ 本公司产品 ” 的生产或变更 “ 本公司产品 ” 的规格。

3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

(1) 除了额定值、性能指标外,使用时还必须遵守 “ 使用条件等 ”。

(2) 客户应事先确认 “ 适用性等 ”,进而再判断是否选用 “ 本公司产品 ”。“ 本公司 ” 对 “ 适用性等 ” 不做任何保证。

(3) 对于 “ 本公司产品 ” 在客户的整个系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。

(4) 使用 “ 本公司产品 ” 时,客户必须采取如下措施: (i) 相对额定值及性能指标,必须在留有余量的前提下使用 “ 本公司产品 ”,并采用冗余设计等安全设计 (ii) 所采用的安全设计必须确保即使 “ 本公司产品 ” 发生故障时也可将 “ 客户用途 ” 中的危险降到最小程度、(iii) 构建随时提示使用者危险的完整安全体系、(iv) 针对 “ 本公司产品 ” 及 “ 客户用途 ” 定期实施各项维护保养。

(5) 因 DDoS 攻击(分布式 DoS 攻击)、计算机病毒以及其他技术性有害程序、非法侵入,即使导致 “ 本公司产品 ”、所安装软件、或者所有的计算机器材、计算机程序、网络、数据库受到感染,对于由此而引起的直接或间接损失、损害以及其他费用,“ 本公司 ” 将不承担任何责任。

对于 (i) 杀毒保护、(ii) 数据输入输出、(iii) 丢失数据的恢复、(iv) 防止 “ 本公司产品 ” 或者所安装软件感染计算机病毒、(v) 防止对 “ 本公司产品 ” 的非法侵入,请客户自行负责采取充分措施。

(6) “ 本公司产品 ” 是作为应用于一般工业产品的通用产品而设计生产的。除 “ 本公司 ” 已表明可用于特殊用途的,或已经与客户有特殊约定的情形外,若客户将 “ 本公司产品 ” 直接用于以下用途的,“ 本公司 ” 无法作出保证。

(a) 必须具备很高安全性的用途 (例: 核能控制设备、燃烧设备、航空 / 宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)

(b) 必须具备很高可靠性的用途 (例: 燃气、自来水、电力等供应系统、24 小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)

(c) 具有苛刻条件或严酷环境的用途 (例: 安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)

(d) “ 产品目录等 ” 资料中未记载的条件或环境下的用途

(7) 除了不适用于上述 3.(6)(a) 至 (d) 中记载的用途外,“ 本产品目录等资料中记载的产品 ” 也不适用于汽车 (含二轮车,以下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

4. 保修条件

“ 本公司产品 ” 的保修条件如下。

(1) 保修期限　自购买之日起 1 年。(但是,“ 产品目录等 ” 资料中有明确说明时除外。)

(2) 保修内容　对于发生故障的 “ 本公司产品 ”,由 “ 本公司 ” 判断并可选择以下其中之一方式进行保修。

(a) 在本公司的维修保养服务点对发生故障的 “ 本公司产品 ” 进行免费修理 (但是对于电子、结构部件不提供修理服务。)

(b) 对发生故障的 “ 本公司产品 ” 免费提供同等数量的替代品

(3) 当故障因以下任何一种情形引起时,不属于保修的范围。

(a) 将 “ 本公司产品 ” 用于原本设计用途以外的用途

(b) 超过 “ 使用条件等 ” 范围的使用

(c) 违反本注意事项 “3. 使用时的注意事项 ” 的使用

(d) 非因 “ 本公司 ” 进行的改装、修理导致故障时

(e) 非因 “ 本公司 ” 出品的软件导致故障时

(f) “ 本公司 ” 生产时的科学、技术水平无法预见的原因

(g) 除上述情形外的其它原因,如 “ 本公司 ” 或 “ 本公司产品 ” 以外的原因 (包括天灾等不可抗力)

5. 责任限制

本承诺事项中记载的保修是关于 “ 本公司产品 ” 的全部保证。对于因 “ 本公司产品 ” 而发生的其他损害,“ 本公司 ” 及 “ 本公司产品 ” 的经销商不负任何责任。

6. 出口管理

客户若将 “ 本公司产品 ” 或技术资料出口或向境外提供时,请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则,“ 本公司 ” 有权不予提供 “ 本公司产品 ” 或技术资料。

IC321GC-zh
202107

注: 规格如有变更, 恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司
http://www.fa.omron.com.cn　　咨询热线: **400-820-4535**