项目七 运动控制系统应用案例（一）

7.3手轮对刀的运动控制

手脉轮运动控制编程

我们知道，数控机床加工工件之前必须对刀，否则数控系统无法识别编程人员设定的工件坐标系，也就无法进行正常加工。

手轮可以实现对刀，通过手轮选择不同的轴跟随手轮运动，同时可以选择以不同的速度比例跟随，最终实现对刀。

手轮对刀的主从运动控制程序主要实现以下功能：

1．程序界面可以实现运动控制器初始化、清除状态、位置清零、启动和停止手轮的功能。

2．程序实现手轮与从轴的电子齿轮运动，其中手轮编码器为主轴，通过手轮可以改变从轴的轴号和速度。

3．能从界面设定离合区位移，并将主从轴的位移脉冲数显示在界面上。

手脉轮运动控制编程

1 新建MFC项目，并调用运动控制器函数库

在Visual Studio中新建项目工程，项目名称为“MPG”。创建过程可参考Jog运动控制，只需将“Jog”改为“MPG”。

2 调用运动控制器函数库

（1）配置驱动器和运动控制器

（2）调用库及配置文件

（3）添加库文件

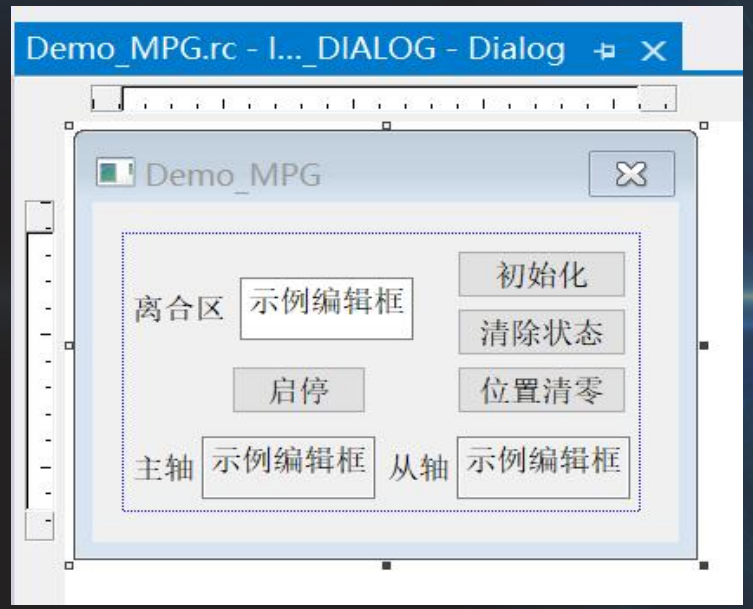
（4）添加头文件

以上几步均可参考Jog运动控制

手脉冲运动控制编程

3 设计界面

根据需要设计程序界面，如图所示，并修改控件属性。



界面设计

手脉轮运动控制编程

4 代码实现

（1）在Demo_MPGDlg.cpp文件中对手脉的轴号进行宏定义，手脉轮外接在端子板上，轴号是固定的，为11。

手脉轮外接在端子板上，轴号是固定的，为11，同时，定义一个BOOL类型的全局变量flag，作为判断手轮功能启动和停止的标志位，flag初始值为FALSE，默认手轮功能是关闭状态。

```
#define  masterAxis  11          //宏定义手轮轴号作为主轴
BOOL flag = FALSE;  //定义全局变量，作为判断手轮启动和停止的标志位
```

手脉轮运动控制编程

4 代码实现

（2）手脉轮实验中，需要初始化、状态清除和位置清零就可以。

使用手轮时，电机轴伺服使能、伺服关闭通过手轮的拨档控制，所以不用通过按钮去实现使能和关闭功能。这里只需要初始化、状态清除和位置清零三个按钮。

初始化、状态清除的程序参考Jog运动控制部分完成，位置清零按钮代码如下。

```
void CDemoMPGDlg::OnBnClickedButtonZeroPos()
{
        short sRtn;//返回值变量
        sRtn = GT_Stop(0x0f, 0x0f);                    //位置清零之前，电机轴需要先停止运动，将1-4轴全部停止
        sRtn = GT_ZeroPos(1,4);                        //将1-4轴位置清零
}
```

手脉轮运动控制编程

4 代码实现

（3）读取编码器位置以及手脉轴选和倍率。

在Demo_MPGDlg.cpp文件中定义获取手脉编码器、轴选和倍率的函数，并在Demo_MPGDlg.h头文件中进行声明。该函数主要实现两个功能：一是读取手脉轮编码器脉冲位置，并将其显示在界面上；二是读取手脉轮轴选和倍率的DI信号，根据读取的DI数据，调用判断轴号和倍率函数。

```
void CDemoMPGDlg::mpgGetPosDi()
{      //读取编码器位置以及手脉轴选和倍率
       short sRtn;      //返回值变量
       double mpgPos;           //手脉轮编码器位置变量
       long DiValue;            //轴选和倍率IO变量
       CString strVal;          //CString类型字符串变量
       while (flag)     //当条件为真时，执行循环体
       {
              sRtn = GT_GetEncPos(11, &mpgPos, 1);          //读取辅助编码器的位置
              strVal.Format(_T("%f"), mpgPos);              //将读取到的编码器数值从double类型转换为CString类
              SetDlgItemText(IDC_Mas_Pos, strVal); //将strVal变量的值用ID为IDC_Mas_Pos的Edit Control控件显示
```

手脉轮运动控制编程

4 代码实现

```
        sRtn = GT_GetDi(MC_MPG, &DiValue);        //轴选和倍率
        mpgSelectAxis(DiValue);                   //调用判断轴选和倍率函数
        sRtn = GT_ClrSts(1, 4);   //清除1-4轴异常报警，这里是清除限位触发标志
        MSG msg;      //MSG是Windows程序中的结构体。在Windows程序中，消息是由MSG结构体来表示的。
        if (PeekMessage(&msg, (HWND)NULL, 0, 0, PM_REMOVE))
        {
            ::SendMessage(msg.hwnd, msg.message, msg.wParam, msg.lParam);
        }
    }
}
```

手脉轮运动控制编程

4 代码实现

sRtn = GT_ClrSts(1, 4);是为了电机轴触发限位信号后清除限位触发标志的。

在单线程程序中，当while循环无法自动改变循环条件时，程序陷入死循环，此时界面便无法操作，为了解决这个问题，通过如下代码不断检测窗口消息、向窗口发送消息，并且PeekMessage处理后，消息从队列里除掉。

```
MSG msg; //MSG是Windows程序中的结构体。在Windows程序中，消息是由MSG结构体来表示的。
if (PeekMessage(&msg, (HWND)NULL, 0, 0, PM_REMOVE))
{
        ::SendMessage(msg.hwnd, msg.message, msg.wParam, msg.lParam);
}
```

手脉轮运动控制编程

4 代码实现

（4）轴号选择以及倍率变换函数

在Demo_MPGDlg.cpp源文件中定义void CDemo_MPGDlg::mpgSelectAxis(long lGpiValue)函数，在此函数中通过传入手脉轴号以及倍率的值选择相应的电机轴作为从动轴、设置从动轴比例，同时在切换当前从轴时关闭上一次选中的轴号；同样该函数需要在Demo_MPGDlg.h头文件中进行声明。

```
void CDemoMPGDlg::mpgSelectAxis(long diValue)
{      //判断轴选和倍率
            short sRtn;             //返回值变量
            short slaveAxis = 0;            //从轴轴号
            long slaveEvn = 1;              //从轴传动比系数 必须初始化 否则切换倍率时会中断
            switch (diValue & 0x0f)       //获取轴号
            {
                case 0x0e:        //0000 1110:14
                        slaveAxis = 1;        //选中1轴
                        sRtn = GT_Stop(1 << (2 - 1), 1 << (2 - 1));        //将2轴停止运动
                        break; //结束语句段，跳出switch语句
```

手脉轮运动控制编程

4 代码实现

```
case 0x0d:              //0000 1101:13
                        slaveAxis = 2;          //选中2轴
                        sRtn = GT_Stop((1 << (1 - 1)) | (1 << (3 - 1)), (1 << (1 - 1)) | (1 << (3 - 1)));          //将1、3轴
停止运动
                        break;      //结束语句段，跳出switch语句
                case 0x0b:              //0000 1011:11
                        slaveAxis = 3;          //选中3轴
                        sRtn = GT_Stop((1 << (2 - 1))| (1 << (4 - 1)), (1 << (2 - 1)) | (1 << (4 - 1)));          //将2、4轴
停止运动
                        break;      //结束语句段，跳出switch语句
                case 0x07:              //0000 0111:7
                        slaveAxis = 4;          //选中4轴
                        sRtn = GT_Stop(1 << (3 - 1), 1 << (3 - 1));             //将3轴停止运动
                        break;      //结束语句段，跳出switch语句
                default:    //将1到4轴伺服关闭
                        sRtn = GT_AxisOff(1);
                        sRtn = GT_AxisOff(2);
                        sRtn = GT_AxisOff(3);
                        sRtn = GT_AxisOff(4);
                }
```

手脉轮运动控制编程

4 代码实现

```
    switch (diValue & 0x70)//获取倍率
    {
        case 0x60:slaveEvn = 1; break;          //110 0000:96，从轴传动比系数设为1
        case 0x50:slaveEvn = 10; break;         //101 0000:80，从轴传动比系数设为10
        case 0x30:slaveEvn = 100; break;        //011 0000:48，从轴传动比系数设为100
    }
    if (slaveAxis)   //当从轴轴号不为0时执行if语句
    {
        sRtn = GT_AxisOn(slaveAxis);            //使能选中轴
        mpgGearMotion(slaveAxis, slaveEvn); //调用从轴运动控制程序
    }
}
```

将读到的DI信号做处理，分别得到轴选和倍率，然后依次判断选中的轴号和倍率。

将diValue和0x0f（0000 1111）进行按位与运算，则只保留低四位有效，将diValue和0x70(0111 0000)进行按位与运算，得到5~7位的数据，我们根据这个设置倍率。

手脉轮运动控制编程

4 代码实现

（5） 从轴运动程序

在Demo_MPGDlg.cpp源文件中，添加void CDemoMPGDlg::mpgGearMotion(short SlaveAxis, long SlaveEvn)函数，实现从轴运动模式、跟随方式、传动比和离合区设置等操作，其中离合区参数从界面获取，将从轴编码器脉冲位置显示在界面上。同样需要在Demo_MPGDlg.h头文件中进行声明。

```
void CDemoMPGDlg::mpgGearMotion(short SlaveAxis, long SlaveEvn)
{
	short sRtn;	//返回值变量
	long masterEvn = 1;	//主轴传动比系数
	double slaPos;//从轴位置变量
	CString strVal; //CString类型字符串变量
	sRtn = GT_PrfGear(SlaveAxis);		//设置从轴运动模式位电子齿轮模式
	//设置从轴，跟随主轴编码器
	sRtn = GT_SetGearMaster(SlaveAxis, masterAxis, GEAR_MASTER_ENCODER);
```

手脉轮运动控制编程

4 代码实现

```
    sRtn = GT_SetGearRatio(SlaveAxis, masterEvn, SlaveEvn, slope);    //设置从轴的传动比和离合区
    sRtn = GT_GearStart(1 << (SlaveAxis - 1));        //启动从轴

    sRtn = GT_GetEncPos(SlaveAxis, &slaPos, 1);    //读取从轴编码器的位置
    strVal.Format(_T("%f"), slaPos);          //格式转换，将slaPos变量从short类型转换为CString类型
    SetDlgItemText(IDC_Sla_Pos, strVal);   //将strVal变量值用ID为IDC_Sla_Pos的Edit Control控件显示
    UpdateWindow();  //更新窗口
}
```

手脉轮运动控制编程

4 代码实现

（6）启停按钮程序

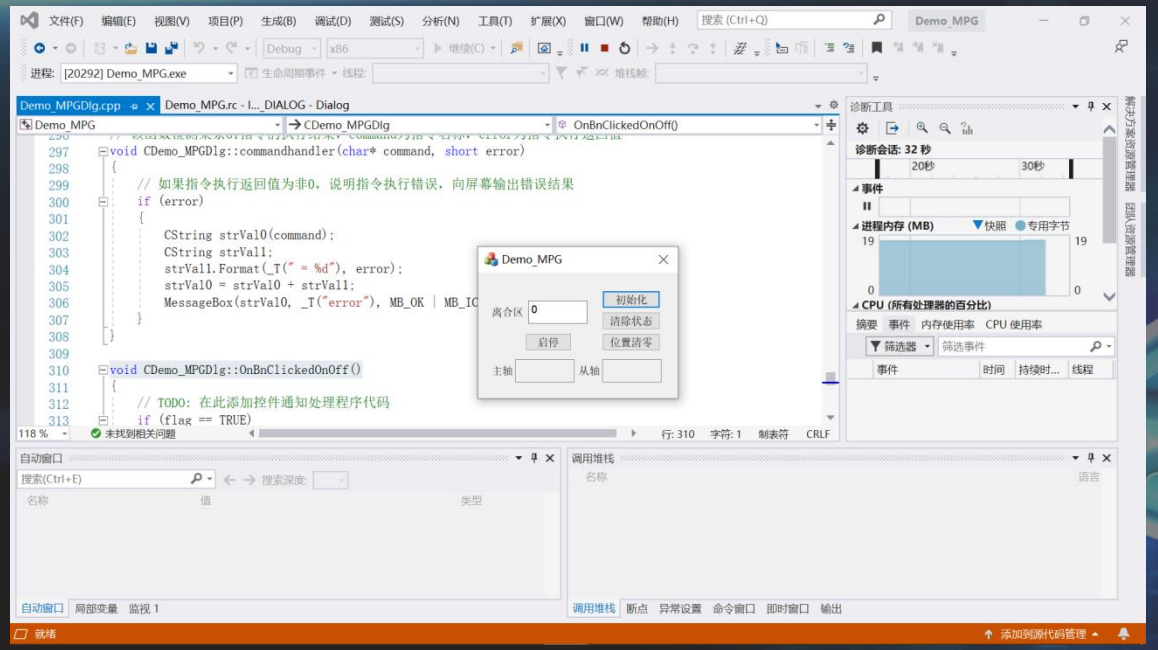
默认情况下，手脉轮启动和停止的标志位变量是FALSE状态，单击按钮，启动手脉轮功能，标志位变为TRUE状态，同时调用读取手脉轮编码器和DI信号的函数；再按一次按钮，标志位再变为FALSE状态，同时将所有电机轴停止运动，伺服使能关闭。

```
void CDemoMPGDlg::OnBnClickedOnOff() {
        if (flag == TRUE)    {
                short sRtn;             //返回值变量
                flag = FALSE;           //将标志位复位为FALSE状态，停用手脉
                sRtn = GT_Stop(0x0f, 0x0f);  //将1-4轴停止运动
                sRtn = GT_AxisOff(1);          //1轴伺服使能关闭
                sRtn = GT_AxisOff(2);          //2轴伺服使能关闭
                sRtn = GT_AxisOff(3);          //3轴伺服使能关闭
                sRtn = GT_AxisOff(4);          //4轴伺服使能关闭    }
        else {
                flag = TRUE;            //将标志位置位为TRUE状态,启用手脉
                mpgGetPosDi();     //调用获取手轮编码器位置和DI信号函数
        }
}
```

手脉轮运动控制编程

5、 程序调试

检查代码无误，生成解决方案，对代码进行调试，如图所示。



运行程序，填写合适的离合区位移，使用手脉改变轴号和倍率，转动手脉编码器，观察轴运动状态，查看主轴、从轴的位移脉冲数。

谢谢