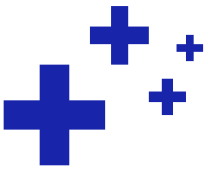
Plugin 使用手册

安装使用本产品前请熟读本手册，并充分理解其内容。

请指定保管人员安全地保存在指定位置以便随时能阅读。



上海铼钠克数控科技有限公司

Shanghai Lynuc CNC Technology Co., Ltd.

概述

关于本手册

- 手册名称　Plugin 使用手册
- 文档类型　说明文档
- 版本　Ver 1.0

本手册的阅读对象

本手册面向：

- 软件工程师/产品技术人员/技术服务人员/产品使用人员

操作前提

读者应：

- 熟悉本手册中的相关概念
- 受过铼钠克控制装置操作方面的培训

改版说明

版本	发布日期	修订说明
V1.0	2019/6/11	首次发布

目 录

1. Plugin 简介

Plugin 是一种用户使用 Qt 工具二次开发的自定义界面，运行在 LYNUC 控制器。

LYNUC 系统支持客户使用 Plugin 进行二次开发，实现系统本身没有而客户想要实现的特定功能。用户只需要开发自己关注的功能界面，然后通过调用封装好的二次开发接口，来调用底层数据，调用成功后将值返回给客户。

本例程为客户开发 Plugin 提供了一个范例，客户可以以此为模板来编程，实现自己想要的功能。

2. 例程

本例程实现了模式切换和调用程序的简单功能，为客户实现自己功能提供了一个简单的模板。

2.1 获取例程

例程存放在百度云，读者可自主获取。

百度云链接：https://pan.baidu.com/s/16sWTr9nTTvwbf_dYWCnDgQ

提取码：7cx4

2.2 例程简介

如下所示：三个程序，两个.so 文件和一个说明文档

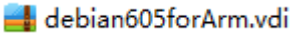
图 2-1 例程内容

以下为上图中各文件的内容功能说明：

1. debian605forArm：配置虚拟机环境所需的虚拟硬盘文件，后缀为.vdi。
2. VirtualBox-5.2.2-119230-Win：虚拟机程序的安装包，来创建 Linux 环境，生成.so 文件。
3. MC2PlugInDemo：示例程序的源码。
4. libMC2PlugInDemoN3.so：最终生成的.so 文件，用来放入 N3 控制器验证功能。
5. libMC2PlugInDemox86.so：最终生成的.so 文件，用来放入 N5,U5,U5E 控制器验证功能。

说明

debian605 解压后是 vdi 后缀文件：



3. Windows 编译

3.1 编辑环境

编译环境：VS2010+Qt4.8.7

验证程序逻辑是否 ok，环境配置如下:

依赖库：MC2PlugInBase.lib

头文件：MC2PlugInBase.h　MC2PlugInGuiConfig.h

3.2 VS 编译具体过程

1. 用 VS2010 打开 MC2PlugInDemo.pro 文件
2. 增加预处理定义：项目-->属性-->c/c++>预处理器：MC2PLAGIN_LIB
3. 增加库文件：链接器-->常规-->附加库目录：MC2PlugInBase.lib；

   链接器-->输入-->附加依赖项：MC2PlugInBase.lib

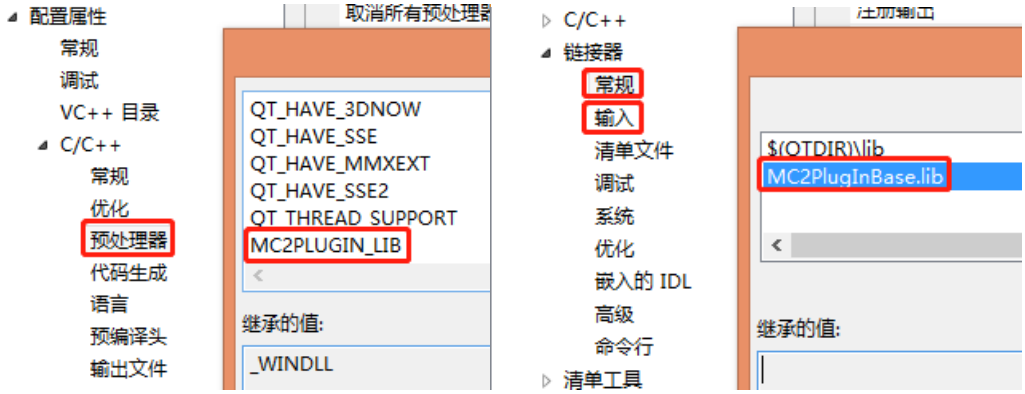


图 3-1 增加库文件

4. 编译程序，下面会提示”正在创建库”，” 成功一个”，说明程序正常

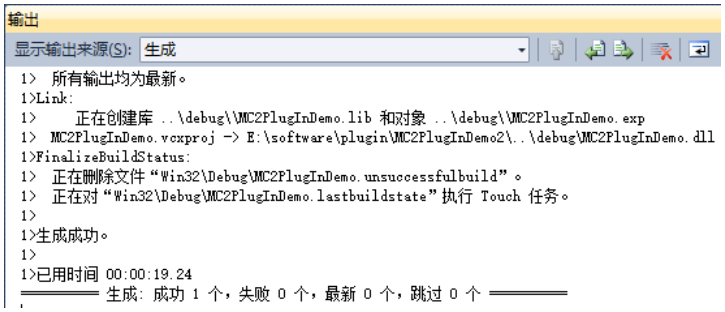


图 3-2　编译程序

4. Virtual Box 虚拟机编译.so 文件

4.1 VirtualBox 简介

VirtualBox 是一款开源虚拟机软件，可虚拟的系统包括 Windows、Mac OS X、Linux、IBM OS2 甚至 Android 等操作系统，使用者可以在 VirtualBox 上安装并且运行上述的这些操作系统。

使用 VirtualBox 可以生成.so 文件，用来放入 LYNUC 数控系统中，作为一个子画面显示。

4.2 生成.so 文件

1. 安装 Virtual Box
2. Virtual Box 新建虚拟电脑，类型：Linux，版本：Debian（32bit）

图 4-1 新建虚拟电脑

3. 下一步内存选默认就 ok

   虚拟硬盘-->使用已有的虚拟硬盘文件 debian605forArm.vdi

4. 设置-->网络-->端口转发，主机端口号输入：123（连接 Xshell 需用到）

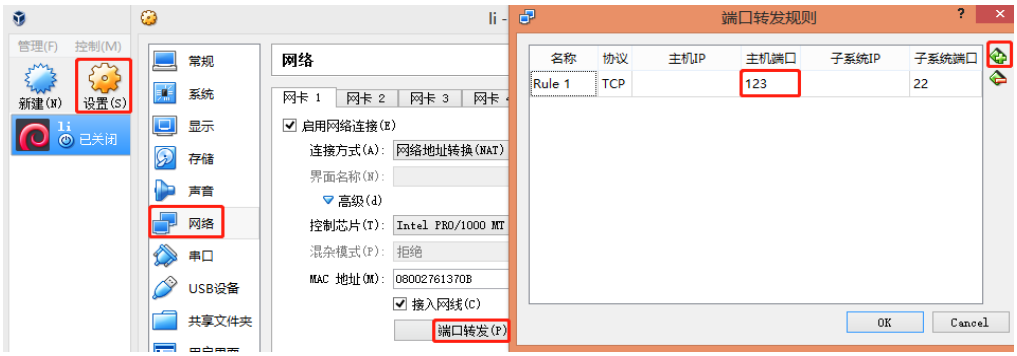
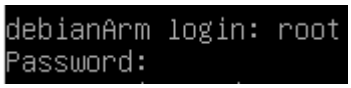


图 4-2 设置网络

5. 然后点击启动，login：root

password：passwd



6. 通过 Xmanager—Xshell 把 MC2PlugInDemo2 文件夹放入虚拟机/home/Lynuc2 目录

7. 先进到 Demo 目录下，然后输入下面代码生成 makefile 文件

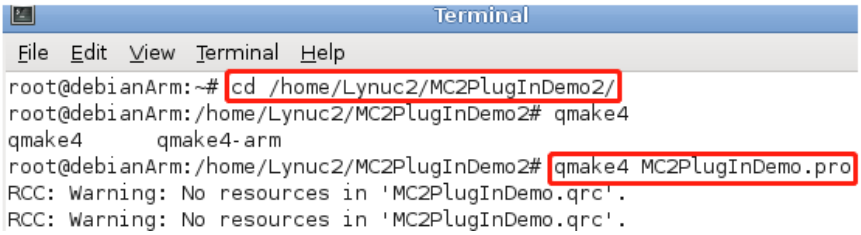


图 4-3 生成 makefile

说明

N5/U5/U5E 输入：qmake4 MC2PlugInDemo.pro

N3 输入：qmake4-arm MC2PlugInDemo.pro

8. 接着输入 make 生成.so 文件

```
root@debianArm:/home/Lynuc2/MC2PlugInDemo2# make
```

9. 结尾显示下图所示说明.so 文件生成成功，放在/home/Lynuc/UI/Plugins 目录

```
mv -f libMC2PlugInDemo.so ../../Lynuc/UI/Plugins/
make[1]: Leaving directory `/home/Lynuc2/MC2PlugInDemo2'
root@debianArm:/home/Lynuc2/MC2PlugInDemo2#
```

图 4-4 生成.so 文件

5. 在控制器上运行

把生成的.so 文件放进控制器，来验证功能是否正常。

1. 修改读写权限

修改磁盘读写权限	mount -o remount,rw /
登陆超级权限 用户名	su
登陆超级权限 密码	passwd

2. 将生成的 libMC2PlugInDemo.so，放入控制器/home/Lynuc/UI/Plugins 目录下

3. 重启后验证功能和界面是否都正常。

图 5-1 最终效果

附录 A　Xshell 连接虚拟机

Xmanager 是一款小巧、便捷的浏览远端 X 窗口系统的工具。在工作中经常使用 Xmanager 来登录远端的 Solaris 系统，在 X 窗口系统上作图形化的操作。就像运行在 PC 上的任何 Windows 应用程序一样，它可以无缝拼接到 UNIX 应用程序中。在 UNIX/Linux 和 Windows 网络环境中，Xmanager 是最好的连通解决方案。它是一个一站式解决方案，这个软件包含有以下一些产品：Xmanager 3D(OpenGL)，Xshell，Xftp 和 Xlpd。

Xshell 连接虚拟机步骤如下：

1. 先要启动虚拟机环境，输入账号：root 和密码：passwd

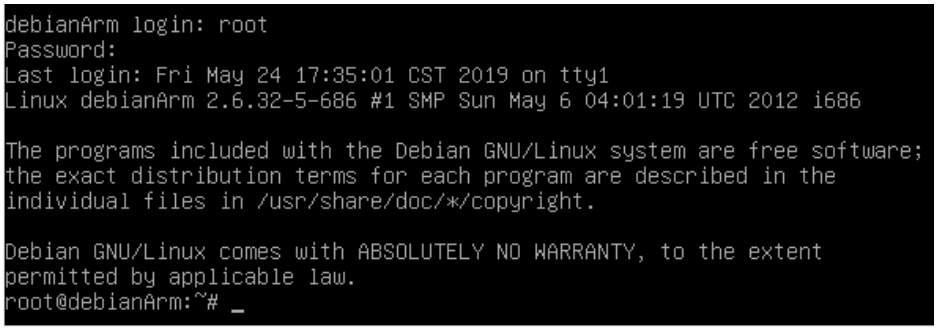


图 A-1 启动环境

2. 然后打开 Xmanager-Xshell，点击新建，名称随意，协议：SSH，主机：127.0.0.1，端口号：123（与虚拟机端口号一致）

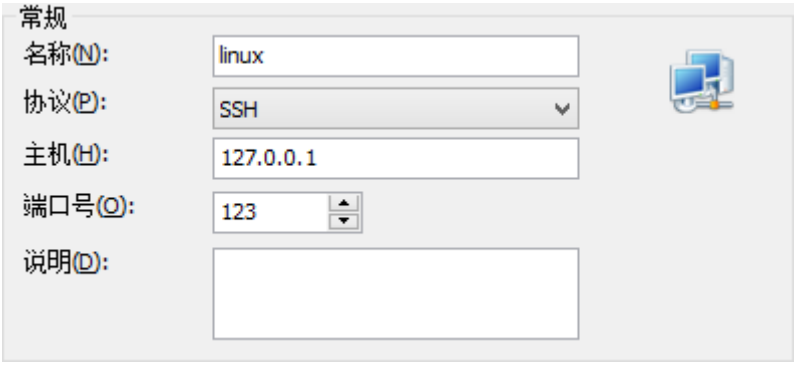


图 A-2 新建项目

3. 点击连接，输入用户名：root，密码：passwd

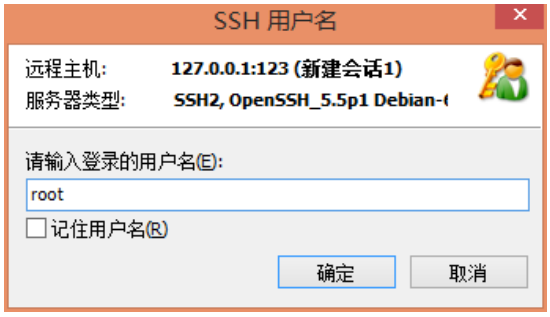


图 A-3 输入账号

图 A-4 输入密码

4. 显示以下结果时说明连接成功

图 A-5 查看是否连接成功

附录 B 代码详细说明

B.1 MC2PluginMenuData

PlugIn 菜单属性结构体，定义及说明如下：

```
typedef struct
{
    int nId;                    //菜单 Id 号
    int nParentId;              //父菜单 Id 号
    int nPosition;              //菜单位置
    char szName[32];            //菜单名称
    int nEnable;                //菜单使能
    int nVisible;               //菜单可见
    int nCheckable;             //菜单是否可选中
    int nChecked;               //菜单是否已选中
    int nRotatable;             //菜单是否可旋转
} MC2PluginMenuData;
```

nId：菜单 Id 号，自动生成，不能修改。

nParentId：父菜单 Id 号，编程时自定义，确定 PlugIn 菜单结构。取值可以为已生成 PlugIn 菜单的 Id 号或者头文件中定义的宏，如下，

```
MC2PLUGIN_MENU_TOP              //Top Menu
MC2PLUGIN_MENU_POSITION         //Position
MC2PLUGIN_MENU_OFFSET           //Offset
MC2PLUGIN_MENU_PROGRAM          //Program
MC2PLUGIN_MENU_EDIT             //Edit
MC2PLUGIN_MENU_INFORMATION      //Information
```

取值为 MC2PLUGIN_MENU_TOP，表示该菜单为顶级菜单节点，可以在其下添加一级、二级、三级菜单。如果取值为其它宏，表示该菜单为对应模块下的一级菜单。如果取值为已生成 PlugIn 菜单的 Id 号，表示该菜单为已生成菜单的子菜单。

nPosition：菜单位置，取值范围 0~14。

szName[32]：菜单名称。

nEnable、nVisible、nCheckable、nCheckedd、nRotatable：菜单属性，0 表示否，1 表示是。

B.2 MC2PluginOP

PlugIn 函数指针结构，包含了各种操作函数。结构定义如下，

```
typedef struct
{
    char * (*getName)();
    char * (*getVersion)();
    int  (*initMenu)(MC2PluginMenuData  menuData[MAX_PLUGIN_MENU_COUNT],
int nLang);
```

```
    void (*doMenuAction)(int nId);
    void (*changeModule)(int nModule);
    void (*changeMode)(int nMode);
    void (*changeStatus)(int nStatus);
    void * (*queryParent)(int nId);
} MC2PluginOP;
```

结构中对应的函数都在 PlugIn 编程时定义，然后通过接口 MC2PluginInit 传给 LynucMC2，queryParent 除外，由 LynucMC2 定义，并传入。

B.3 接口函数说明

声明为：int MC2PluginInit(MC2PluginOP * op)。PlugIn 编程时定义。将编写好的函数指针赋给结构体 op。LynucMC 调用该接口获取 PlugIn 中编写的函数指针，同时给函数指针 queryParent 赋值。

int MC2PluginGetMacroData(int nMacroNo, MC2PluginMacroData * data)
获取宏变量

int MC2PluginExecuteNCCode(char * pFileName);
执行，LynucMC2 切换到 Program 模块，并且将 NC 文件 pFileName 打开。

int MC2PluginGetExecuteFileName(char * szFileName);
获取当前加工的 NC 文件名。

int MC2PluginGetCurNCName(char * pszCurNC);
获取当前 NC 子程序的文件名

int MC2PluginMenuChange(int nMenuId);
用于模块切换，当 PlugIn 的菜单存在顶级菜单时（即父菜单 Id 为-1），使用该接口将 LynucMC2 切换到 PlugIn 模块，此时 nMenuId 为顶级菜单 Id 号，LynucMC2 从 PlugIn 模块返回时，也调用该接口，此时 nMenuId 为-1。

int MC2PluginMaxFileSize();
获取 NC 文件最大长度，单位：字节。

int MC2PluginSettingLevel();
获取当前登录级别，0：未登录；1：普通用户；2：机床厂；3：GSPEED。

int MC2PluginSetSingleMenuEnabled(int nMenuId, bool enabled);
设置单个菜单使能。

int MC2PluginSetMenuVisible(int nMenuId, bool visible);
设置菜单可见。

int MC2PluginSetMenuName(int nMenuId, const QString & name);
设置菜单名称。

int MC2PluginSetSettingData(int macro, MC2PluginSETTINGDATA data);
设置设定参数值。

int MC2PluginSaveSettingData();
保存设定参数。

int MC2PluginSetMenuEnabled(bool enabled);
设置整个菜单系统使能。

B.4 实现接口 MC2PluginInit(MC2PluginOP * op)

示例代码如下：

```
op->initMenu = demoInitMenu;
op->doMenuAction = demoDoMenuAction;
op->changeModule = demoChangeModule;
op->changeMode = demoChangeMode;
op->changeStatus = demoChangeStatus;
demoQueryParent = op->queryParent;
```

函数指针 demoInitMenu、demoDoMenuAction、demoChangeModule、demoChangeMode、demoChangeStatus 都是 PlugIn 编程实现，demoQueryParent 是 LynucMC2 通过变量 op 传入，供 PlugIn 编程调用。

B.5 实现 MC2PluginOP 中声明的函数

char * getName();
获取 PlugIn 名称，可以不用实现。

char * getVersion();
获取 PlugIn 版本号，可以不用实现。

int (*initMenu)(MC2PluginMenuData menuData[MAX_PLUGIN_MENU_COUNT],
int nLang);
初始化菜单，菜单最大数为 256，nLang 用来实现多国语支持。示例代码如下：

```
menuData[0].nParentId = MC2PLUGIN_MENU_PROGRAM;
strcpy(menuData[0].szName, "DEMO");
menuData[0].nEnable = TRUE;
menuData[0].nVisible = TRUE;
menuData[0].nCheckable = TRUE;
menuData[0].nChecked = false;
menuData[0].nRotatable = FALSE;
menuData[0].nPosition = 4;
```

```
menuData[1].nParentId = -1;

menuData[2].nParentId = menuData[1].nId;
strcpy(menuData[2].szName, "ACTION1");
menuData[2].nEnable = TRUE;
menuData[2].nVisible = TRUE;
menuData[2].nCheckable = TRUE;
menuData[2].nChecked = TRUE;
menuData[2].nRotatable = TRUE;
menuData[2].nPosition = 1;
```

说明

返回值：生成菜单数量。
返回值必须和实际菜单个数相等，不然会导致 LynucMC2 崩溃。

void (*doMenuAction)(int nId)
菜单点击相应函数，LynucMC2 点击菜单时，会触发该函数，在该函数内实现菜单点击的操作。

void (*changeModule)(int nModule);
模块切换函数，LynucMC2 模块切换时，会触发该函数，该函数实现模块切换时对应的操作。

void (*changeMode)(int nMode)
模式切换函数，LynucMC2 模式切换时，会触发该函数，该函数实现模式切换时对应的操作。

void (*changeStatus)(int nStatus)
状态切换函数，LynucMC2 状态切换时，会触发该函数，该函数实现状态切换时对应的操作。

void * (*queryParent)(int nId);
获取父窗口指针，由 LynucMC2 定义，并传入，PlugIn 使用该函数获取父窗口指针。

附录 C 免责声明

支持正版，人人有责。安装或者使用公司购买的正版软件以外的其它软件，属于个人行为，个人应负全部法律责任。

上海铼钠克数控科技有限公司

地址：中国上海市闵行区都会路 2338 弄 30-31 号
邮编：201108
电话：+86 21 61837766
传真：+86 21 60720487
网址：http://www.lynuc.cn