#### 项目四 软件界面基础开发

# 指示灯控制程序编程



工业自动化设备常常会安装三色指示灯,不同的颜色有不一样的意义,技术员可以根据指示灯的颜 色判断设备运行状况。本任务为使用 MFC 制作一个基于指示灯的人机交互界面,实现触发和复位时指 示灯有不同的指示状态,如图 4-16 所示。



1. 创建一个基于对话框的 MFC 项目。

2. 在自动生成的对话框模板中, 删除"TODO:在此放置对话框控件。"静态文本控件"、"确定" 按钮和"取消"按钮。添加两个编辑框 Edit Control 控件, 其 ID 分 别为 IDC\_EDIT1、IDC\_EDIT2; 再 添加一个按钮, 将 Caption 属性改为复制。此时的对话框模板如图 4-17 所示。

示例编辑框	
示多性職種權	
要制	

图 4-17 窗口界面

3. 选中编辑框 IDC\_EDIT1, 右键为其添加 CString 类型的控件变量 edit1, 如图4-18 所示。

19/ <b>4</b>	10(1)	41	把1年期間(Y)	
Herei.	IDC_ED(T)	•	LOIT.	
	(T)(89)(8)		當府(N)	
	đ.		edit1[	
	id(Fi(A)		2.00 M (2017)	
	public	r.	CString	
	(EMI(M)			

4. 选中按钮,右击选择属性,选择通知消息事件中的 BN\_CLICKED,在右面写入"OnBtnClickCopy", 如图 4-19 所示。按回车进入按钮单击事件的函数体。

IDC_BUTTON1 (Button Control) IBut	-	
H H A /		
CEditControlDig		
BCN_DROPDOWN		
BCN_HOTITEMCHANGE		
BN_CLICKED	OnBtnClickCopy	
BN_DOUBLECLICKED		
BN_KILLFOCUS		
BN_SETFOCUS		
NM_CUSTOMDRAW		
NM_GETCUSTOMSPLITRECT		
NM THEMECHANGED		

5. 将按钮单击事件函数 CEditControlDlg::OnBtnClickCopy()代码修改为如下: void CEditControlDlg::OnBtnClickCopy()

// TODO: 在此添加控件通知处理程序代码 CString str; UpdateData(TRUE); //刷新控件的值 //获得编辑框1的内容 str = edit1; UpdateData(FALSE); //停止刷新控件的值 //将复制的内容复制到编辑框2 SetDlgItemText(IDC\_EDIT2, str);

6.运行程序,弹出结果对话框,在编辑框1中输入"运动状态",单击复制按钮,界面如图4-20 所示。

3		- 🗆 🗙	
	运动状态		
	运动状态		
	宣わ」		
	图 4-20 运行程序界面		

# 1. 创建一个基于对话框的MFC项目。 (1) 打开VS2019,选择"创建新项目"。 (2)选择"MFC应用",单击"下一步"。

#### Visual Studio 2019

#### 打开最近使用的内容(R)

使用 Visual Studio 时,你打开的任何项目、文件夹成文件都將显示在此处供你快速访问。 可固定任何旗ૂ繫打开的时象,使其始终位于列来顶部。

开始使	用
*	克隆存储库(C) 从 GitHub 版 Azure DevOps 等联机存储库获 代码
Ъ	打开项目或解决方案(P) 打并本地 Visual Studio 调目或 .sln 文件
2	打开本地文件夹(F) 导航和周期任何文件夹中的代码

创建新项目(N)

出接具有代码基架的项目模板以开始

继续但无需代码(W) →

\*7

创建新项目

間 Windows 桌面应用程序

凹 Windows 桌面向导

最近使用的项目模板(R)

p-搜索模板(Alt+S)(S) 全部清除(C) C++ Windows . 桌面 CD
Windows 桌面应用程序
具有在 Windows 上运行的图形用户界面的应用程序的项目。 C++ C++ Windows 桌面 C++ ■\*\*\* 空白应用(通用 Windows - C++/CX) ↔→ 用于没有预定义控件或布局的单页通用 Windows 平台(UWP)应用的项目。 C++ XAML Windows Xbox UWP 卓面 MFC 应用 生成具有在 Windows 上运行的复杂用户界面的应用。 C++ Windows 貞面 **一** 共享项目 0 使用"共享条目"项目在多个项目之间共享文件。 C++ Windows Android iOS Linux 桌面 控制台 库 UWP 游戏 移动 ATL 项目 使用活动 使用活动模板库(ATL)创建小型快速的组件对象模型(COM)对象。 C++ Windows 桌面

上一步(B) 下一步(N)

#### 1. 创建一个基于对话框的MFC项目。

(3) 在 "项目名称" 中对项目进行命名, 此处我们以ioShow为例。命名完成后单击创建。

(4) 将"应用程序类型"改为"基于对话框",单击"完成"。

(5) 在菜单栏中选择"视图"→"其他窗口"→"资源视图",单击后出现以下对话框。在"资源 视图中"选择中"ioShow.rc"→"Dialog"→"IDD\_IOSHOW\_DIALOG",双击后出现编辑界面。



#### 1. 创建一个基于对话框的MFC项目。

(6) 在菜单栏中选择"视图"→"工具箱",单击后出现以下对话框。

此对话框内都为可直接使用的界面控件,在制作界面时,只需 要将控件拖动至界面编辑区域即可使用。

工具箱		▼ -¤ X
搜索工具	具箱	ρ.
▲ 对话	框编辑器	<u>^</u>
k	指针	
	Button	
×	Check Box	
abi	Edit Control	
<b>-</b> ++++++++++++++++++++++++++++++++++++	Combo Box	
	List Box	
[ <u>*vz</u> ]	Group Box	
⊙	Radio Button	
_4.3	Static Text	
	Picture Control	
<b>A</b> F	Horizontal Scro <mark>ll B</mark> ar	
•	Vertical Scroll Bar	
<del>.</del> 0-	Slider Control	
\$	Spin Control	
<b>DOC</b>	Progress Control	
Ĩ.	Hot Key	
	List Control	
Tr=	Tree Control	
<b>—</b>	Tab Control	
*	Animation Control	
2.0	Rich Edit 2.0 Control	1000
1/1/1	Date Time Picker	
囲	Month Calendar Control	

2. 在自动生成的对话框模板中, 删除"TODO: 在此放置对话框控件。"静态文本控件、"确定"按钮和"取消"按钮。添加两个按钮, 将Caption属性分别改为"IO触发"、"IO复位"; 再添加一个 Picture Control控件, 修改其ID为IDC\_Led。

3. 准备两张IO指示灯的图片放到工程的res文件夹中, 如图所示。



4. 在VS界面资源视图中的"ioShow.rc"节 点上点右键,选择"添加资源"示,弹出添加资源 对话框。

然后在左侧的资源类型中选择"lcon",点按 钮"导入",显示一个文件对话框,我们选择res 文件夹中的LedOff.ico图片文件,导入成功后会在 Resource View的ioShow.rc\*节点下的"lcon"文 件下出现一个新的子节点"IDI\_ICON1",这里采 用默认ID不修改。以同样的方法将res文件夹中的 LedOn.ico图片导入,默认ID为"IDI\_ICON2", 如图所示。



5. 选中Picture Control控件,在图片控件的属性页中有一个Type属性,因为要加载的是图标图片,所以Type属性选择Icon。在图片控件的Image属性的下拉列表中选择导入的图标IDI\_ICON1。此时,界面如图所示。



6. 选中"IO触发"按钮,右击选择属性,选择通知消息事件中的BN\_CLICKED,在右面写入 "BtnChangelO",按回车进入按钮单击事件的函数体CioShowDlg::BtnChangelO(),修改为如下:

```
void CioShowDlg::BtnChangeIO()
      // TODO: 在此添加控件通知处理程序代码
      //从资源加载图标
      HICON hIcon = NULL;
      hIcon = AfxGetApp()->LoadIcon(IDI_ICON2); //获取加载IDI ICON2
图标的句柄
 //为Picture Control 控件加载图标
      IO Status. SetIcon(hIcon);
```

7. 选中"IO复位"按钮,右击选择属性,选择通知消息事件中的BN\_CLICKED,在右面写入 "Btn\_LedOff",按回车进入按钮单击事件的函数体CioShowDlg::Btn\_LedOff(),修改为如下:

```
void CioShowDlg::Btn_LedOff()
```

// TODO: 在此添加控件通知处理程序代码

```
//从资源加载图标
HICON hIcon = NULL;
hIcon = AfxGetApp()->LoadIcon(IDI_ICON1); //获取加载IDI_ICON1图标的
句柄
//为Picture Control控件加载图标
IO Status. SetIcon(hIcon);
```

二. 指示灯控制程序编程与调试

8. 运行程序, 弹出结果对话框, 单击 "IO触发"按钮, IO指示灯变亮, 如图所示。再单击 "IO复位"按钮, IO指示灯变为初始状态。



# THANK YOU!