wincc 的 N 个经典问题解答

1：如何触发计算机扬声器的声音？

答： 编写如下 C-Action:

```
#pragma code("kernel32.dll");
BOOL Beep(DWORD dwFreq,DWORD dwDuration);
#pragma code();
Beep(500,500);
```

2：如何通过 C 脚本来确定报警信息？

答： 首先必须在画面中插入报警控件，可以用如下两种方式来确认信息：

（1）、确认单条信息

4 版本和高于此版本的 WinCC

```
BOOL OnBtnSinglAckn（char*lpszPictureName,char*lpszObjectName）
```

5 版本和高于此版本的 WinCC

```
BOOL AXC_OnBtnSinglAckn（char*lpszPictureName,char*lpszObjectName）
```

（2）、确认报警窗口所有可见的报警

4 版本和低于此版本的 WinCC

```
BOOL OnBtnVisibleAckn（char*lpszPictureName,char*lpszObjectName）
```

5 版本和高于此版本的 WinCC

```
BOOL AXC_OnBtnVisibleAckn（char*lpszPictureName,char*lpszObjectName）
```

3：如何在 WinCC 中读取系统时间？

答： 通过如下 C-Action：

```
#pragma code("kernel32.dll");
Void GetLocalTimes(SYSTEMTIME*lpst);
#pragma code();
SYSTEMTIME time;
GetLocalTime(&time);
SetTagWord("Varname",time.wYear);
SetTagWord("Varname",time.wMonth);
SetTagWord("Varname",time.wDayOfWeek);
SetTagWord("Varname",time.wDay);
SetTagWord("Varname",time.wHour);
SetTagWord("Varname",time.wMinute);
SetTagWord("Varname",time.wSecond);
SetTagWord("Varname",time.wMilliseconds);
```

4：如何经 Windows 对话框设置日期时间？

答： 通过调用 Windows 对话框实现。具体如下：

```
#include"apdefap.h"
void onClick(char*lpszPictureName,char*lpszObjectName,
char*lpszPropertyName)
```

{ProgramExcute("c:\\win98\\control.exe timedate.cpl");}

其中执行的程序路径，需根据具体情况填写。

5：如何在 WinCC 中调用 SQL 语言？

答： 1、创建一个 SQL 文件，此文件在 ISQL 中建立，文件内包含所要执行的 SQL 语句。Windows 对话框实现。具体如下：

2、在 WinCC 中用 C Script 调用上述 SQL 文件，如下所示：

```
#include"apdefap.h"
void OnLButtonDown(char* lpszPictureName,
char* lpszObjectName,
char* lpszPropertyName,
UINT nFlags,int x,int y)
{
char*a="c:\\siemens\\common\\SQLANY\\ISQL-q-b-c
UID=DBA;PWD=SQL;DBF=E:\\testsql\\testsqlRT.DB;
DBN=CC_testsql_99-12-03-12:48:26R;READ
E:testsql\\test.sql";
Printf("%s\r\n",a);
ProgramExcute(a);
}
```

下面是一个简单的 SQL 文件内容：

```
select *from pde#hd#t#test;
output to e:\\test2.txt FORMAT ascii
```

注意：文件名及路径中不要带空格。

6：如何整点启动归档？

答： 在"Globle Script"下的 Project function 编写程序函数：cyclicarchive

```
BOOL cyclicarchive()
{
#pragma code("kernel.dll");
void GetLocalTime(SYSTEMTIME*lpsz);
#pragma code();
SYSTEMTIME time;
Int t1;
GetLocalTime(&time);
T1=time.wMinute;
If(t1==00)
{
SetTagBit("startarchive",1);
Return(BOOL) (GetTagBit("startarchive"));
}
}
```

在 Tagloging 中的"Properties of process tag"中的"archive tag"Tab 下的 Archive type 选择 Cycle-selective，在"Event"标签下的"StartEvent"内选择 cyclicarchive 函数。

7：How can I set and reset a WinCC variable by mouse click with C script?

答： The following function shows how you can alternately set and reset a WinCC variable by mouse click.

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
BOOL z;
z=GetTagBit("MyBitVariable");
if (z==0)
SetTagBit("MyBitVariable",1);
else
SetTagBit("MyBitVariable",0);
}
```

8：How can I program a waiting function (Sleep) in WinCC?

答： The following sample program shows how the "Sleep" is used.

```
#pragma code("Kernel32.dll")
void Sleep(int milliseconds);
#pragma code()
Sleep(1000); //time specification in milliseconds
```

Warning:

If you use Sleep(), processing the C script is interrupted for the time indicated. Requests for the interrupted function cannot be processed during this time.

9：How can I output a SIMATIC timer minutes and seconds in WinCC?

答： If you want to output a SIMATIC Timer in minutes and seconds in WinCC, then in WinCC please use a static text in the Graphics Designer to which you interface the following action:

```
char* _main(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
char *p;
DWORD hilf;
int min, sec;
p=SysMalloc(10);
hilf=GetTagDWord("Time");
min=hilf/60000;
sec=hilf%60000/1000;
sprintf(p,"%d min %d sec",min,sec);
return p;
}
```

The "Time" variable linked to the I/O field must have the following propert

ies:

•Data type "32-bit value without sign"

•Format adaptation "DwordToSimaticBCDTimer"

•Address: data area "Times" and addressing "Word"

10：快捷地切换画面

通常要将所有的设备都显示在一张画面里是不可能的，所以将设备按照处理工艺的功能步骤分级在多张画面内，以一个污水处理厂为例分为电泳线、前处理线、生化线、加药线等，之间的切换使用按钮的鼠标动作来实现。这对于用 WinCC 现成的鼠标动作来实现是很简单的，但是不是要在每张图上都使用相同数量且位置排列顺序一致的按钮呢？(出于对操作的一致性考虑，不能让操作人员在不同的图上，不同的位置找想要操作的按钮)这个问题的解决我们使用 WinCC 的脚本编程，在按钮动作中调用它的内部函数来实现。首先，组态一幅背景画面，其中包括要显示的静态文本、OLE(例如，时钟)、所有的图形切换按钮及推出关机按钮。第二，在背景画面中插入智能对象(Smart Object)中的画面窗口(Picture Window)，并且使其的尺寸与其分级画面相同。第三，在相应的切换按钮的属性->事件->鼠标动作中编写如下 C 语言脚本代码：

```
#include\"apdefap.h\"
void onClick(char *lpsz PictureName,
char *lpsz ObjectName,char *lpsz PropertyName)
{
SetPictureName(\"背景画面\", \"画面窗口\", \"电泳处理线\");
} //\"背景画面\"即为始终显示地静态背景的属性名,\"画面窗口\"即为在背景画//面中插入地画面窗口(Picture
Window)属性名，\"电泳处理　　线\"即为分级画面//的名称
```

这样就能方便快捷地切换画面。

11：必须始终显示的报警记录，用画面颜色闪烁来提示操作者有故障发生，可以用同样的 C 语言脚本(当然不是加在鼠标动作中)在背景画面的底部留下一条类似 Windows 状态栏的报警栏，当出现故障报警的时候在报警栏显示最近一条报警记录，操作人员可以利用按钮切换到主报警记录画面了解故障的完全信息。

12：单个部件的组态

WinCC 在其内部的图库里集成很多的图形对象，如水箱、电机、阀门等等，对于污水处理用的最多的是水池，搅拌机等等。为了使组态画面贴近实际，能够更好的反应现场工况，图库里的元素并不能满足要求。我们可以利用 WinCC 的画图工具自己绘制适应实际的图形元素，并且使用 C 脚本使其产生动画效果。

对于调节池，反应池等可以用矩形来表示，以图形的填充高度来示意其液位的高度。这里要注意的是一些数值地转化，由于一些仪表例如 E+H 的超声波液位计，是以 4～20 毫安的电流信号进入 PLC，对于表的参数设定好以后 4 毫安就对应相应池子的液位最低点，而 20 毫安对应于池子的液位最高点。所以在显示数字高度的时候要将 PLC 相对 4、20 毫安的数值转化为实际池子的高度范围例如，0.5~5.5 米。但是对于图形填充的时候是按照百分数来表示的所以还要将 PLC 数值转化为 0～100%的度量范围。

搅拌机的组态可以使用 C 脚本使其产生动画，在其运行时产生视觉上的旋转。实际上搅拌机的图形是用两个部分椭圆组成的，在椭圆的属性->几何->半径中加入代码如下：

```
#include \"apdefap.h\"
```

```
long _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
static int x=25,y,z;//x 为半径的初始值为 25，y 为切换变量，z 为搅拌机运
行状态变量
z=GetTagBit(\"202 搅拌机运行\"); //取得 PLC 搅拌机的运行状态
if(z!=0)
{
switch(y)
{
case 0:
x=x-4;
if(x<=0)
y=1;
break;
default :
x=x+4;
if(x>=25)
y=0;
break;
}
}
return x;
}
```

13：语音报警的组态

在工业现场安全是极其重要的，无论从那个角度讲我们应该利用一切手段减少故障的发生。在故障已经发生的时候，应该在第一时间以多种方式通知操作人员有故障发生。现代微处理计算机的处理速度可以完全胜任对于图形，语音地同时处理。所以我们可以利用计算机的声卡和音箱在有故障发生的时候产生语言报警，但是问题是 WinCC 本身并不能产生语音的功能，而且其内部的上千个函数也没有提供处理声音的函数。解决这个问题的方法归功于微软开发的 WinCC 与操作系统地完美结合，因为在 WinCC 中可以直接调用 Windows 的 API 函数。实现的具体 C 脚本代码如下：

```
long _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#pragma code(\"Winmm.dll\")
void WINAPI PlaySoundA(char *pszSound,char *hmode,DWORD dwflay);
#pragma code()
if(GetTagBit(\"112 排泥备泵故障\"))
PlaySoundA(\"d:\\\\\\\\\winnt\\\\\\\\media\\\\\\\\Mircosoft sound.wav\
",NULL,8);
return 1020;
}
```

有一个值得主意的问题是此段代码的加入点，通过反复多次的实践得出结论，即这个点必须加在始终显示于屏幕上的任何图形元素的属性中，这样才能达到语言报警的预期效果。

14：当为 WinCC 指定 PC 名时应注意什么?

解答:计算机名不能包含特殊字符如空格、退格和下划线，并前 13 个字符必须是唯一的。由于操作系统的原因，名字的长度不能超过 15 个字符。因此推荐只使用 a to z, A to Z, 0 to 9 这些字符。必须以子母开头。

从 SIMATIC WinCC V6.0 起，有如下限制：计算机名可包含下划线。然而，当使用 DNS 主机名时，不能使用下划线。

15：和 plc 用 S7 通讯为什么产生连接错误代码：D801？

解答:这个错误代码表示 WinCC 变量地址错误，检查每个变量的地址和通讯连接是否正确，如果变量的地址不属于控制器的地址范围，则会报这个错误代码。

16：和 plc 用 S7 通讯怎样清除错误代码 8304？

解答:WinCC 运行时的画面不再更新，WinCC 的日志文件中有错误代码 8304，这个错误码表明 WinCC 和 S7 连接出现问题。

解决方法:

停止 AS 站的循环读服务，在"WinCC Explorer > 变量管理 > SIMATIC S7 PROTOCOL SUITE"
中。右键单击出现问题的 S7 连接，选择"系统参数"，清除复选框“周期管理>通过 PLC”的选择。

17：在动态值域对话框中怎样才能按逻辑连接两个因变量到一个结果?

解答:由于在 WinCC 动态值域对话框中的布尔数学体系运算符，您可以根据 C 脚本惯例运用这些符号。

下面是符号及其意思的概括:

与 位比较 &
与 表达式比较 &&
或 位比较 |
或 表达式比较 ||
非 !

通过 Object > Properties > Dynamic 进入 WinCC 动态值域对话框 ,> > 并右击"Dynamic Value Ranges dialog" 按照需要输入表达式即可。

18：WinCC 图形编辑器中是否存在通过鼠标点击达到增加/减少变量值的对象？

解答：可以使用 OCX“SpinButton”实现这个功能。下面描述了如何合并和联系这个对象。

在 WinCC 图形编辑器中，插入一个“Control”类型的小对象。在打开的窗口中选择进入 Microsoft Forms 2.0 SpinButton 并以 OK 来确认。

右击打开对象“SpinButton”的属性。在控制属性中使用您想要的变量来连接“Value”属性。

但要注意:

使用第三方的 ActiveX 控件会导致错误(例如内存丢失，性能降低，系统阻塞)。" 软件

应用者应该对由于使用第三方 ActiveX 控件而造成的任何问题负责。

ActiveX 控件事件连接 C 脚本

如果连接 C 脚本到 ActiveX 控件事件，那么应该确认这个事件名至少 5 个字符长。如果这个事件名少于 5 个字符长，那么 C 脚本不被执行。

19：重新启动后，不使用登录窗口如何以一个缺省用户的身份自动登录？怎样确保运行期间有个缺省用户始终处于登入状态？即使另外一个用户已经预先退出。

解答：

重新启动以及在运行期间,您希望 HMI 系统达到最小的实用性而不使用登录窗口。然而，对于高级操作，登录功能应当保留。此外，如果没有用户 登录，则有一个缺省的用户自动登录。缺省用户的权限可以在用户管理器中根据需要设定。

可使用下面的 C 脚本执行此项功能，请按照下列步骤进行：

将附件中的函数“Silentlogin.pas”复制到项目中的“PAS”子文件夹中。

在项目中选择 Global Script > C Editor > Actions > Global Actions 并打开全局动作 (Global Action)“Silentlogin.pas”。

在“PWRTSilentLogin (“Login”,“Password”);”一行中，用缺省的用户名替换用户“Login”，用 缺省用户的口令代替“Password”。

编译和保存 C 脚本。

用变量@CurrentUser 设置一个变量触发器，周期选择为“ 2 s”。这确保了系统不会因脚本而负荷过重。在所述的例子中，变量 @CurrentUser(包含当前登录的用户)每隔 2 秒钟被询问一次察看有无变化。只有当用户变化时才调用脚本，例如当前用户退出时。

确保已经在计算机的属性“Startup”选项卡上激活了“Global Script Runtime”。

除此之外，SIMATIC PCS 7(SIMATIC PCS 7 V6.0 SP1 及更高版本)用户还需执行下列步骤：

将系统画面@Welcome.pdl 复制到一个安全的位置，以便可以恢复原始画面。

使用 Graphics Designer 打开画面@Welcome.pdl。

打开保存在选定画面中的 C 脚本，(右击)> Event > Picture Object > Miscellaneous > Open Picture)。

注释掉“PASSLoginDialog(Screen);”一行，以 // 作为注释的前缀。 编译 C 脚本。 保存系统画面@Welcome.pdl。

20：如何进行 WinCC 和 S7 之间的时间同步?

回答:下面的方法只能进行时间设置而不能完成时间同步的功能。因为必须考虑到延迟，比如报文的处理时间，C 脚本的运行时间等，因此这个方法在精度方面不如真正的时间同步精确。

1.创建一个数据块 ，其中有一个 "DATE_AND_TIME" 类型的变量和一个布尔变量。当元素"Flag" 在 WinCC 中被置 1，程序就调用系统函数 SFC0 "SET_CLK"。D B1 中的触发变量" Flag" (DB1.DBX 8.0)在 WinCC 脚本中被 SetTagBitWait("DB1_FLAG",TRUE) 置 1。因此仅当此 C 脚本在 WinCC 中被调用时,时间才被设置。SFC0 必须先添加到 Step7 程序的块文件夹中。 在此段程序中，DB 块中的各个时间变量被写到作为 SFC0 参数的"DATE_AND_TIME"类型的本地变量"DateAndTime"中。 用 "SET_CLK" 设置完时钟后，触 发变量"Flag"被复位。

注意： 在此段程序中，本地变量"DateAndTime"存储在以 0 为起始地址的本地数据堆栈中。如果不得已要把这个变量分配到别的地址，同 时对传送指令参数化时，必须要考虑到地址

分配的问题。

2.为“年”新建一个 "Unsigned 8-bit value"类型的变量，对其进行格式变换 ByteTo BCDByte，然后为其在 DB 块中选择相应的字节地址： 在 WinCC 中创建剩下的变量。 选择 "Unsigned 16-bit value" 类型然后改变格式为 WordToBCDByte ，然后在 DB 块中选择实际的字节地址。

3.最后,在 WinCC 中创建一个 C 脚本来读取系统时间并进行拆分，然后把它们写到 DB1 中。 S7 PLC 中的时钟设置是被脚本中的"DBI_FLAG"变量触发的。

21：WINCC--如何在程序中动态修改用户密码

1、点击“开始”--》“设置”--》“控制面板”--》“管理工具”--》“数据源（ODBC）”，打开 ODBC 数据源管理器

2、在用户 DSN 页面的用户数据源中找到与当前项目所关联的一项，其名称为“CC_项目名_项目建立日期时间”，记下该项的名称

3、在脚本中加入以下代码：

```
#pragma code("UseGen.dll")
#include "USEGENAP.H"
#pragma code()
LPCMN_ERROR err; //定义的 LPCMN_ERROR 型变量,在函数调用中需使用
if (PWGENConnect("CC_ass_04-09-21_16:35:22",err)) //建立与数据库的联接,其中 CC_ass_04-09-21_16:35:22 用第二步中记下的名称取代
{
if (PWGENChangePassword(GetTagChar("user"),GetTagChar("oldpassword"),GetTagChar("password"),err)) //修改密码,user,oldpassword,password 分别为存贮用户名,原密码,新密码的内部变量,类型为文本变量 8 位字符集
{
//密码修改成功后的操作,如给用户提示等
}
}
PWGENDisconnect(err); //断开与数据库的联接
```

4、编译运行程序

5、工作完成

6、与用户管理相关的函数定义存贮在 APPLIB 目录下的 USEGENAP.H 文件中，可根据上面示例自行完成添加用户，修改权限等功能。

22：WinCC6.0 中归档时和 5.1 版本为什么不同了？

WinCC V6.0 的后台数据库采用了 MS SQL Server 2000,所以归档方式与 V5.1 有所不同，它的运行数据存放在数据片段（segment）当中，工程师可以根据尺寸需求组态最大容量或根据时间周期启动新的数据库归档片段。将归档数据连续的写入数据库，单个数据片段的尺寸到达或者时间界限到达时，系统会自动开启另一个数据片段进行归档。当数据片段的总体尺寸达到最大时，最早的数据片段就会被覆盖，重新开始新的归档。

23：WinCC6.0 中如何设定归档周期？

WinCC V6.0 版本中的快慢速归档的归档周期界限可以由用户自行设定，该参数在快速归档属性的第三个标签项中设置。

24：如何计算慢速归档数据库的尺寸？

慢速归档时一条变量归档记录占用 32 字节的空间，每个变量以 2 分钟为归档周期，一周之内会产生 5040 条记录，若有 5000 个变量的归档，则单个数据片段的大小计算为：

32×5000×5040＝806400000 byte ==> 约等于 800MB

考虑到留出 20%的余量，设定单个数据片段为 1G

所有数据归档期限是两个月，因此所有段的尺寸为单个片段尺寸乘以单个片段的个数，即：1GB×9＝9GB

25：如何计算快速速归档数据库的尺寸？

快速归档时一条变量归档记录占用 3 字节的空间，每个变量以 2 秒钟为归档周期，一周之内会产生 302400 条记录，若有 50 个变量的归档，则单个数据片段的大小计算为：

3×50×302400＝45360000 byte ==> 约等于 46MB

考虑到留出 20%的余量，设定单个数据片段为 60MB

所有数据归档期限是两个月，因此所有段的尺寸为单个片段尺寸乘以单个片段的个数，即：60MB×9＝540MB

26：所有的归档变量都可以计算出它占用的数据库大小吗？

只有周期连续归档的数据才能定量的计算其占用的数据库尺寸，因此当您对应设定的时间期限计算并设置数据库尺寸大小时，需要考虑其他数据归档类型的数据，留出相应的余量。

27：WinCC V5.1 中文版的安装要求是什么？

1）WinCC V5.1 亚洲版只有 V5.1 这一个版本，不再有后继版本，WinCC V5.1 亚洲版应安装在 Windows 2000 SP2 操作系统上

2）WinCC 的语言版本应和操作系统的语言版本相对应，不建议将中文 WinCC 装在英文操作系统上

28：有没有快捷的方法如何将 WinCC 的实时数据通过 OPC DA 记录到 MS Access、MS SQL Server 和 Oracle 数据库中？

可以使用 WinCC Industrial Data Bridge 将 WinCC 的实时数据通过 OPC DA 记录到 MS Access、MS SQL Server 和 Oracle 数据库中，但是需要授权，分为以 128、 512、2K 和 10K

29：WinCC 能提供的最高变量刷新速度是多少？

对于一般的网络通讯方式来说，WinCC 能提供的最高刷新速度是 250 毫秒，但 WinCC 采用 RawData 归档数据链接的方式可以实现对 S7-400PLC 的高速数据采集。

30：如何实现 WinCC 高速数据采集？

WinCC 采用 RawData 归档数据链接的方式可以实现对 S7-400PLC 的高速数据采集。原理是 PLC 将每个循环周期所采集的过程值（或 PLC 以其他方式得到的数据或数据包）以一定的顺序存放在具有一定的格式的 DB 块中，当到达一定的数量后，PLC 可以调用系统功能块 SFB37（AR_Send）将这个 DB 块主动地发送给 WinCC，然后 WinCC 会在后台自动调用标准化 DLL 来拆解数据，并将其按时间顺序保存在数据库中。在 WinCC 的过程画面中，可以使用在线趋势控件或在线表格控件来查看所采集的数据。

由于是批量传送，可以有效地提高通讯效率，使高速数据采集成为可能，而这时所谓的采集频率就取决于你对保存在 DB 块中的各过程值间的时间间隔的定义。可以定义的最小的时间间隔是 1 毫秒。但如果是 PLC 每个循环周期采样一次，那么定义的时间间隔应大于 PLC 循环周期。DB 块的最大尺寸是 16KB。

技巧：可以考虑使用多 DB 块进行缓冲并添加程序控制 DB 块的写入和发送顺序，能够实现连续的采集，但要充分考虑 CPU 的负载和循环周期。

31：WinCC 高速数据采集的前提条件是什么？

1） WinCC 的版本为 V5.1 或更高

2） S7-400 系列 CPU

3） WinCC 站与 S7 400 站建立 S7 连接（包括 MPI,ProfiBus,TCP/IP,工业以太网都可以实现）

32：为什么我无法从 WinCC 里调用 STEP 7 变量？

从 WinCC 里调用 STEP 7 变量的前提条件是，WinCC 的项目文件必须是集成在 STEP 7 项目中的。

在安装所有 Simatic 软件前，请查阅软件的安装注意事项，确定操作系统与软件的兼容性。该文档一般位于：CD\Documents\<语言版本>\InstallNotes.chm。

要使用 WinCC 与 STEP 7 的集成功能，WinCC 和 STEP 7 必须安装在同一台计算机上，必须在安装 WinCC 之前安装 STEP 7。STEP 7 与 WinCC 的版本必须一致。

33：如何把现成的 WinCC 项目集成到 STEP 7 项目中？

如果你在一台计算机上已经安装了兼容的 WinCC 和 STEP 7，并且有了一个单独使用的 WinCC 项目，想把它集成到一个已有的 STEP 7 项目中去。那么，你必须先添加与集成相关的 WinCC 组件。把 WinCC 光盘放入光驱，并启动 WinCC 的安装程序。添加与集成相关的 WinCC 组件。

按如下步骤把已有 WinCC 项目文件插入 STEP 7 项目文件

1.在 STEP 7 项目文件中插入一个 OS 站，然后把它改名为已有的 WinCC 项目名称。

2.在 STEP 7 项目里删除因仅插入 OS 站而产生的 WinCC 项目文件，其位置在 STEP 7 项目文件夹里的 wincproj 文件夹下，例如：d:\siemens\STEP7\S7proj\STEP7_Integration\wincproj\'Name of the OS'

3.最后在项目复制器里用'Save as'把已有的 WinCC 项目文件存储到 STEP 7 项目路径下。项目复制器位于开始菜单项"Start > SIMATIC > WinCC > Tools"。

注意：对于集成 STEP 7 项目里的 WinCC 项目，你也可以使用压缩工具（Packer）来归档 WinCC 项目。

34：wincc 怎样和 s7plc-sim 连接？

要访问 PLCSIM 模拟软件，必须按如下方法操作，按照以下的顺序来安装程序：

STEP 7 V 5.x

PLCSIM V4.x 以上

WinCC V5.x

选择用户自定义安装。 对于 SIMATIC WinCC V5

SP1 及以下版本的用户,在“通讯”组件下，必须选择“S7Dos”和“对象管理器”组件。

对于 SIMATIC WinCC V5

SP2 及更高版本的用户，“S7Dos”是自动安装的。就是说不再需要在“用户自定义安装”下选择该选项。在 PG/PC 界面上做如下设置：在控制面板中双击“设置 PG/PC 界面”。在“应用程序的访问点”域中，选择“MPI

(WinCC)”。在“使用的界面参数”域中选择“<无>”。启动应用程序 STEP 7：

启动 SIMATIC 管理器

启动 PLCSIM

打开要模拟的项目或组态一个项目。

在项目中添加一个 OS。

在 PLCSIM 中加载项目。

启动 WinCC 并创建一个新项目或打开已有的项目。

WinCC

添加“SIMATIC S7 PROTOCOL SUITE”到变量管理器。

在 MPI 下添加一个新连接。

右击该连接然后选择“属性”。

点击“属性”按钮。

在“连接”标签中指定 MPI 地址和已在 STEP 7 中组态好的 CPU 的插槽。

确认所做的指定。

激活 WinCC 项目。

进入“开始 > SIMATIC > WinCC > 工具 > 通道诊断”。在“通道连接”下可以显示连接的状态。

如果连接没能设好：在 WinCC 资源管理器中的“系统参数 - MPI > 单元”中，启用选项“自动设置”。

在“系统参数 - MPI > 通道”中，取消“使用 PLC 的循环读取服务”选项。

wincc 和西门子 PLC 仿真通讯需要 PLCSIM ，先将 PLCSIM

运行，建立一个 WINCC 项目，使用 MPI 通讯即可，说白了跟 MPI 通讯一样，只是 PLC 是用 PLCSIM 在计算机上模拟的

但是有一点要注意的是，这种方式下不能使用输入区（I 区），凡是有输入的地方都要用 M 区来代替

可以使用输入区（I 区），只是有些特殊功能不能用。

35：WINCC-如何使用自定义的对话框实现用户登录？

在登录按钮中加处以下脚本:

```
#pragma code("useadmin.dll")
#include "PWRT_API.H"
#pragma code()
if (PWRTSilentLogin("username", "PassWord"))
{
//登录成功后的处理
}
```

其中"USERNAME","PASSWORD"可以用存储用户名和密码的变量替换.如:

```
PWRTSilentLogin(GetTagChar("user"), GetTagChar("PassWord"))
```

当登录成功时,函数返回值为真;如登录失败,则返回值为假。

36：WINCC-如何使用自定义的对话框实现用户退出？

实现登陆：

```
#pragma code(""useadmin.dll)
#include "PWRT_api.h"
#Pragma code()
PWRTlogin('1');
```

实现退出：

```
#pragma code(""useadmin.dll)
#include "PWRT_api.h"
#Pragma code()
PWRTlogout();
```

37：如何把 GIF 图片放到 wincc 中？

首先插入 aniGIF.ocx 这个 ole 控件，然后双击它，在 GIF 属性中选择您需要显示的 GIF 图片就可以了。

38：如何将低版本创建的项目移植到 WinCCV6.0 中？

将项目移植到 WinCC V6.0 的工作步骤：

WinCC V6.0 与其以前的版本相比在数据组织方面有着显著的不同。为了使在 WinCC V5.0 Service Pack2 或 WinCC V5.1 中创建的项目在 WinCC V6.0 中也能工作，项目数据必须首先通过移植作相应的调整。为此，WinCC V6.0 提供了一个项目移植器，用于自动移植项目的组态数据、运行系统数据和归档数据在移植之前，建议为原版本的项目做一个备份。与此有关的信息参见 WinCC 信息系统中的主题“使用 WinCC”>“使用项目”>“复制和归档项目”。

已归档的文件:如果必须访问先前版本的归档数据，则必须将归档移植到 WinCC V6.0。请使用项目移植器移植归档数据和 dBASE III 归档。

多用户项目:为了使利用 WinCC V5.0 SP2 或 V5.1 所创建的多用户项目在 WinCC V6.0 中能够正常工作，可进行如下操作：

移植系统中所有服务器上的单个多用户项目。如果原来的项目使用了多客户机，则分别单独移植多客户机的项目数据。正常操作中的冗余系统,不用取消激活操作就可在冗余系统中对项目进行升级。此时，将按规定的次序升级服务器、客户机和多客户机。章节“在正常操作中升级冗余系统”中提供了有关的详细说明。

警惕:为了不影响系统操作，必须遵守所描述的步骤次序，且完成所有步骤时不能有任何长时间的中断。

早于 WinCC V5.0 Service Pack 2 的 WinCC 版本:对于早于 WinCC V5.0 SP2 的 WinCC 版本所创建的项目，必须一步一步地进行移植,将系统先升级到 WinCC V5.1，并移植项目。安装 WinCC V6.0，并使用项目移植器移植项目。

39：wincc6.0 中支持 ab plc 的驱动吗？

WinCC V6.0 将不再提供下列通讯通道：

Allen Bradley DH DH+ DH485
Allen Bradley Serial DF1
Applicom Multi Protocol Interface
GE Fanuc SNP SNPX
Mitsubishi FX
Modbus Protocol Suite

Modbus Serial
SIMATIC S5 PMC Ethernet
SIMATIC S5 PMC Profibus
SIPART
可以用 OPC 来替代。某些通道需附加件的支持。

40：WINCC 的授权坏了，显示“Authorization SIK/SIMATIC WINCC RT 128 PowerTags is faulty.”，重新安装显示已经存在此授权，请问要怎么处理？

可以询问原来的销售商索要一个激活码，可以在 authorsw 中 manageauthorization 中右键单击选择“recover autrorization”输入激活码激活授权即可。

41:用 C 编程解决授权点数不够用的问题(acsun 提供)

当需要的工艺参数超过 WinCC 版本限制的 Tag 数目,可以用 C 语言编程实现多个工艺参数打包成一个 Tag 传送.例如某个配料称重系统有 146 个参数超过了 WinCC 的 128 个 Tag 的限制就可以用 C 语言编程决这一问题而不需要购买更高的授权.

基本思想就是把多个参数在下位机内存中连续排列然后在WinCC中定义一个Tag它的长度是多个参数之和取得这个 Tag 后编程将其分成多个参数

例如下位机有两个参数 LTN44001 和 LTN44023 都是 16 位整数分别存放在 DD99.DW146 和 DD99.DW148

在 WinCC 中定义一个外部 Tag 命名为 PackageTag 类型为 32 位整数并联地址为 DD99.DBD146 再定义两个内部 tag 名为 LTN44001 和 LTN44023 在 Global Script 全局脚本中 C 语言编程如下

```
Union
{
Long Dword
Int Word[2]
}union
Union.Dword=GetTagDword("PackageTag")
SetTagWord("LTN44001",Union.Word[0])
SetTagWord("LTN44023",Union.Word[0])
```

这样两个参数 LTN44001 和 LTN44023 就通过一个 Tag 传送上来了理论上只要下位机内存足够可以传送任意数量的参数而不受 WinCC 版本外部 Tag 数目的限制.

42:如何在 WinCC 里用 C 语言调用 SQL 语言？

1、创建一个 SQL 文件。

此文件在 ISQL 中创建，文件内容是所希望执行的 SQL 语句。

2、在 WinCC 的 C Script 中编写程序调用此 SQL 文件，如以下程序所示：

```
#i nclude "apdefap.h"

void OnLButtonDown(char* lpszPictureName,

char* lpszObjectName,

char* lpszPropertyName,
```

```
UINT nFlags, int x, int y)

{

char*a="C:\\SIEMENS\\Common\\SQLANY\\ISQL-q-b-c

UID=DBA;PWD=SQL;DBF=E:\\testsql\\testsqlRT.DB;

DBN=CC_testsql_99-12-03_12:48:26R;

READ

E:\\testsql\\test.sql";

printf("%s\r\n",a);

ProgramExecute(a);

}
```

下面是一个简单的 SQL 文件内容：

```
select * from pde#hd#t#test;

output to E:\\test2.txt FORMAT ascii
```

注意:文件名及路径中不要带空格。

43：如何整点启动归档？

在"Global Script"下的 Project functions 编写函数：cyclicarchive

```
BOOL cyclicarchive()

{

#pragma code ("kernel32.dll");

void GetLocalTime (SYSTEMTIME* lpst);

#pragma code();

SYSTEMTIME time;

Int t1;

GetLocalTime(&time);
```

```
t1=time.wMinute;

if(t1==00)

{

SetTagBit("startarchive",1);

return(BOOL)(GetTagBit("startarchive"));

}

}
```

在 Tagloging 中的"Properties of process tag"中的"Archive Tag"tab 下的 Archiving type 选择 Cycle-selective，在"Event"标签下的"Start Event"内选择 cyclicarchive 函数。

44：如何在按键组合被禁用的情况下，从 WinCC 运行环境进入 WinCC Control Center？

最好是做一个按钮，该按钮需要用用户权限保护，在该钮中编写如下 C-action：

低于 WinCC 5.0 版本：

```
#pragma code ("user32.dll");

BOOL SetForegroundWindow(HWND);

#pragma code();

HWND handle;

handle=FindWindow("MCPFrameWndClass",NULL);

If (!SetForegroundWindow(handle))

Printf ("\r\n SetForeground fails");
```

WinCC 5.0 版本以及更高的版本：

```
#pragma code("user32.dll");

BOOL SetForegroundWindow(HWND);

#pragma code();

HWND handle;

handle=FindWindow("WinCCExplorerFrameWndClass",NULL);
```

```
If (!SetForegroundWindow(handle))

Printf ("\r\n SetForeground fails");
```

45：WinCC 如何实现鼠标 OnMouseOVer 事件?

用 WINDOWAPI 函数 GetCusorPos 获取当前鼠标位置,用 GetWindowRect 函数获取窗口位置，两值相减得鼠标在 WINCC frame 上的相对位置。用全局脚本（设定为 1s 定时刷新），然后获取要 OnMouseOver 事件的物体的位置，并与鼠标位置相比较，如一致则触发自己定义的动作。

```
#i nclude "apdefap.h"

int gscAction( void )
{
#pragma code("user32.dll");
BOOL GetCusorPos(POINT lpPoint);  //获取鼠标的位置（绝对位置-对应屏幕分辨率）
BOOL GetWindowRect(HWND hwnd,LPRECT lpRect); //获取窗体┘位置
#pragma code();
POINT pPos;
RECT rRec;
HWND hwnd;
BOOL bRet,bRet2;
long lLeft,lTop,lWidth,lHeight;
long lX,lY;
char szStr[100];

hwnd=FindWindow(NULL,"WinCC-Runtime - ");  //如语言为中文应为"WinCC 运行系统-"

if (hwnd==0) {printf("\r\nError! WinCc Handle is %d",hwnd);goto over;}

bRet=GetCursorPos(&pPos);
if (bRet==0) goto over;

bRet2=GetWindowRect(hwnd,&rRec);
if (bRet2==0) goto over;
lX=pPos.x-rRec.left;   //鼠标 对{置
lY=pPos.y-rRec.top;   //鼠标 ¶位置
//如果为非全屏模式，需将上述数值中的高 height 判断减去标题栏的宽度
//printf("The Current Cusor Pos is x:%d,y:%d\r\n",pPos.x,pPos.y);
//printf("The Cusor Pos in Window is x:%d,y:%d\r\n",pPos.x-lLeft2,pPos.y-lTop2);

lLeft=GetLeft("NewPdl0.Pdl","Text1");      //Return - Type :long int
lTop=GetTop("NewPdl0.Pdl","Text1");      //Return - Type :long int
```

```
lWidth=GetWidth("NewPdl0.Pdl","Text1");       //Return - Type :long int
lHeight=GetHeight("NewPdl0.Pdl","Text1");       //Return - Type :long int

if ((lX>=lLeft)&&(lY>=lTop)&&(lX<=lLeft+lWidth)&&(lY<=lTop+lHeight)) {

sprintf(szStr,"%d,%d",lX,lY);
SetText("NewPdl0.Pdl","Text1",szStr);       //Return - Type :char*

}
//printf("The Text1 Pos is x:%d,y:%d\r\n",lLeft2,lTop-lTop2);
over:
return 0;
}
```

46：如何实现 ASP 与 WinCC V6 数据库 sql server 2000 的连接?

WinCC V6 用 ODK 获取 DSN 名，{后生成网址字符串，再用 shellExecuteA 函数打开网页

```
 //-------------------------------------------------------------------------
------
#pragma code("shell32.dll")
long ShellExecuteA(HWND,LPCTSTR,LPCTSTR,LPCTSTR,LPCTSTR,int);
#pragma code()

char dsnStr[100];
HWND hwnd;
sprintf(dsnStr,"http://dcount/test.asp?dsnStr=%s",GetDSN(1));       //Return-Typ
e: LPCTSTR
printf("\r\n%s",dsnStr);
hwnd=FindWindow(NULL,"WinCC - Runtime ");
ShellExecuteA(hwnd,"open",dsnStr,NULL,"C:\\",SW_SHOWNORMAL);
//-------------------------------------------------------------------------
--------
```

ASP 方面需要注意的是连接字符串的形式与普通 acess ODBC 不太一样，而且需要用户名和密码

由于 WINCC 封装的 sa 用户的密码暂时不知道，所以必须用 SQL Server EnterPrise Manage r 建立自己的用户，添加 system Administrator 权限就可以了！否则会出现错误提示

```
Microsoft OLE DB Provider for SQL Server 错误 '80040e4d'
Login failed for user 'sa'.

'-------------------------------------------------------------------------
--------
dsnStr=request("dsnStr")
set conn=server.createobject("adodb.connection")
set rs=server.createobject("adodb.recordset")
conn.Provider = "sqloledb"
```

```
conn.open "Server=DCOUNT\WINCC;Database=" & dsnStr & ";UID=dcount;pwd=;"
rs.open "select * from test",conn,1,1
response.write rs.recordcount
do while not rs.eof
      response.write rs("f1") & "---" & rs("f2") & "---" & rs("f3")
      rs.movenext
loop
rs.close
conn.close
```

47：如何在退出 WINCC 监控时直接关闭电脑?

1.利用动态向导

2.在脚本中加入 DMExitWinCCEx (DM_SDMODE_SYSTEM);也可以

48：如何通过 WINCC API 函数读出当前报警消息?

1、使用 GMsgFunction 中读出当前报警信息的 ID（dwMsgNr）；

2、使用 MSRTGetMsgCSData 函数读出该报警信息 ID 对应文本库 TEXTLIB 中的文本 ID（dwTextID1）；

3、使用 MSRTGetMsgText 函数读出该文本 ID 的文本。

信息到达处理：如果读取文本成功，则置文本变量 MSG。

信息离去处理：如果 MsgNr 与上一次相同，则复位 MSG，如果不是，则继续保持信息。

具体函数说明请看 ODK 文档

```
BOOL GMsgFunction( char* pszMsgData)
{
#pragma code("msrtcli.dll")
#i nclude "msrtapi.h"
#pragma code();

MSG_TEXT_STRUCT tMeld;
MSG_CSDATA_STRUCT sM;
CMN_ERROR  err;
BOOL bRet;
DWORD dwTextID1;
DWORD dwMsgNum;
char  szMsg[255];
#define TAG_MSG "MSG"

  MSG_RTDATA_STRUCT mRT;
  memset( &mRT, 0, sizeof( MSG_RTDATA_STRUCT ) );

  if( pszMsgData != NULL )
  {

    printf( "Meldung : %s \r\n", pszMsgData );
    sscanf( pszMsgData,  "%ld,%ld,%04d.%02d.%02d,%02d:%02d:%02d:%03d,%ld, %ld,
```

```
%ld, %d,%d",
       &mRT.dwMsgNr,                    // Meldungsnummer
       &mRT.dwMsgState,                     // Status MSG_STATE_COME, .._GO, .._Q
UIT, .._QUIT_SYSTEM
       &mRT.stMsgTime.wYear,              // Tag
       &mRT.stMsgTime.wMonth,              // Monat
       &mRT.stMsgTime.wDay,             // Jahr
       &mRT.stMsgTime.wHour,              // Stunde
       &mRT.stMsgTime.wMinute,             // Minute
       &mRT.stMsgTime.wSecond,              // Sekunde
       &mRT.stMsgTime.wMilliseconds,       // Millisekunde
       &mRT.dwTimeDiff,                   // Zeitdauer der anstehenden Meldung
       &mRT.dwCounter,                   // Interner Meldungsz?hler
       &mRT.dwFlags,                    // Flags( intern )
       &mRT.wPValueUsed,
       &mRT.wTextValueUsed );

//*************************************************************************************
*********code for dcount
if (mRT.dwMsgState==MSG_STATE_COME)//信息到达处理
{
       dwMsgNum=mRT.dwMsgNr;
       printf("\r\nThe Alarm Message No is %d !\r\n",dwMsgNum);
       bRet=MSRTGetMsgCSData(dwMsgNum,&sM,&err);
       if (bRet==TRUE)
        {
             dwTextID1=sM.dwTextID[0];
             printf("\r\nThe TextID of The MessageNr %d  is %d !\r\n",dwMsgNum,d
wTextID1);
             bRet=MSRTGetMsgText(1,dwTextID1,&tMeld,&err);
             if (bRet==TRUE)
                   {
                   sprintf(szMsg,"%s",tMeld.szText);
                   printf("\r\nThe Text of TextID  %d  is %s !\r\n",dwTextID1,sz
Msg);
                   }

       }

}
if (mRT.dwMsgState==MSG_STATE_GO)//信息离去处理
{
       dwMsgNum=mRT.dwMsgNr;
       if (dwMsgNum==GetTagDWord(TAG_MSG_NR)) SetTagChar(TAG_MSG,"");
```

```
}
//*************************************************************************
*********code for dcount

    // Meldungsdaten einlesen


      // Prozesswerte lesen, falls gew???????§???1nscht
    }

  printf("Nr : %d, St: %x, %d-%d-%d %d:%d:%d.%d, Dur: %d, Cnt %d, Fl %d\r\n" ,
  mRT.dwMsgNr, mRT.dwMsgState, mRT.stMsgTime.wDay, mRT.stMsgTime.wMonth, mRT.st
MsgTime.wYear,
  mRT.stMsgTime.wHour, mRT.stMsgTime.wMinute, mRT.stMsgTime.wSecond, mRT.stMsgT
ime.wMilliseconds, mRT.dwTimeDiff,
  mRT.dwCounter, mRT.dwFlags ) ;

SetTagChar(TAG_MSG,szMsg);

   return( TRUE );
}
```

49：如何实现用户登陆日志（wincc 中用 c 脚本实现?)(柳树成林原创)

用户登陆日志：（包括用户的登陆退出信息，以便查询在什么时间段是哪个用户在使用这个监控软件）

```
#i nclude "apdefap.h"

int gscAction( void )
{
#pragma code("kernel32.dll")
VOID GetLocalTime(LPSYSTEMTIME lpSystemTime);
#pragma code()

char*  username;
char buf[128];
static char preuser[128];
unsigned a,b,c,d,e,f;
FILE* fp;
SYSTEMTIME sysTime;

//读取系统时间,并且复制给变量 a,b,c,d,e,f
GetLocalTime(&sysTime);
```

```
a=sysTime.wHour;
b=sysTime.wMinute;
c=sysTime.wSecond;
f=sysTime.wYear;
e=sysTime.wMonth;
d=sysTime.wDay;

//得到当前用户名称
username = GetTagChar("@CurrentUser");
fp= fopen("c:\\wincclog.txt", "a+");
if(strcmp(username, preuser)!=0)    //如果当前用户名称和前一个用户名不同
{
       if((strcmp(username, "")  != 0)&&(strcmp(preuser, "") == 0))     //如果
当前用户名称不空同时前一个用户名为空

       {
                 sprintf(buf, "用户:%s\t 登陆时间是：\t %d-%d-%d,%d-%d-%d\n", use
rname,a,b,c,d,e,f);
                 fputs(buf, fp);
        }
       else
       {
               if((strcmp(username, "") == 0)&&(strcmp(preuser, "")  !=
0))     //如果当前用户名称为空同时前一个用户名不空

              {
                        sprintf(buf, "用户:%s\t 退出时间是：\t %d-%d-%d,%d-%d-%
d\n", preuser,a,b,c,d,e,f);
                        fputs(buf, fp);
               }

               else
              {
                        sprintf(buf, "用户:%s\t 退出时间是：\t %d-%d-%d,%d-%d-%
d\n", preuser,a,b,c,d,e,f);
                        fputs(buf, fp);
                        sprintf(buf, "用户:%s\t 登陆时间是：\t %d-%d-%d,%d-%d-%
d\n", username,a,b,c,d,e,f);
                        fputs(buf, fp);
               }
        }
}

strcpy(preuser, username);
fclose(fp);
```

```
return 0;
}
```

50：在 wincc 画面编辑器里注册的 OCX 控件，由于开发时没有考虑到 wincc 标准控件中具备层次哪个属性！所以不能够通过画面编辑器里的菜单进行设置 OCX 控件的层次关系！

建议不要把 wincc 中注册的 OCX 控件和 wincc 本身的标准控件重叠放置，否则 wincc 本身的标准控件将被覆盖！