

学习目的：搭建一个简单的 Allegro Skill 调试环境，写出第一个 HELLO WORLD 程序。

1，首先选择一款适合自己的文本编辑器，免费的有 SciTE, Notepad ++, 收费的有 Ultra Edit, CodeWright. 以上几款软件均有语法高亮文件。推荐使用 SciTE, Notepad ++。

2，在 PCBENV 目录下新建一个文本文件，改名为 test.il，后面范例中的编写都在这个文件中完成，如果不知道 PCBENV 在哪里，参考以下方法：

1. 通过查看电脑环境变量中的 HOME 变量值。例如 HOME 变量为 d:\，对应的目录应为 d:\pcbenv

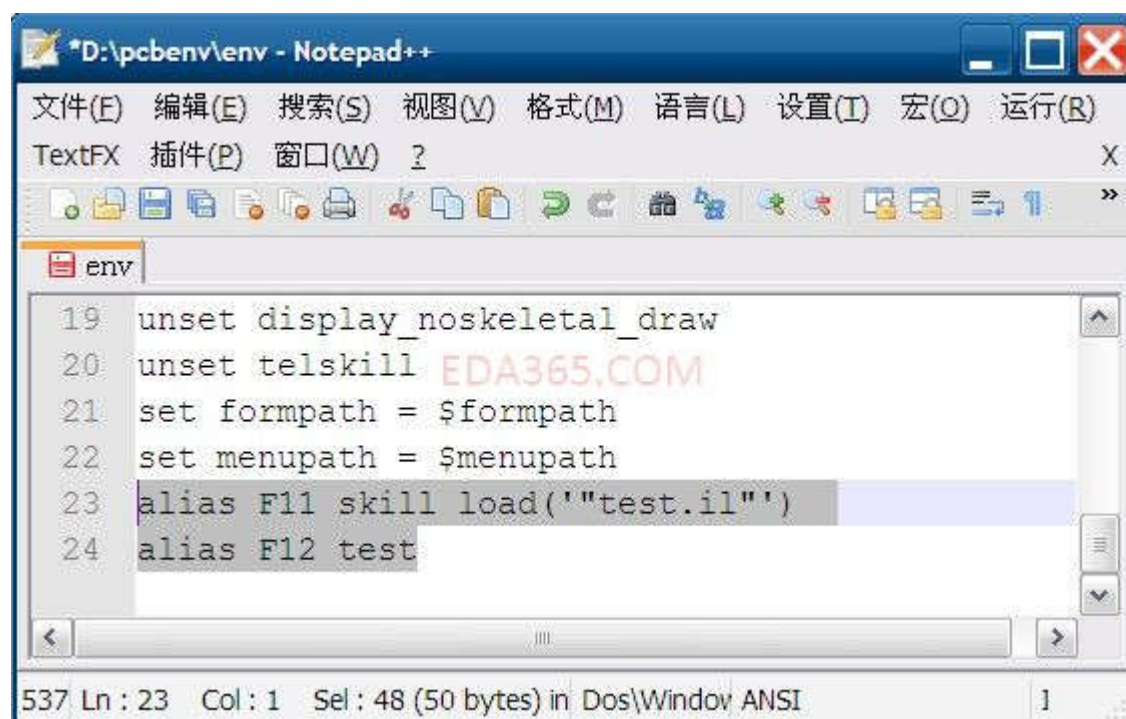
2. 直接在 ALLEGRO 命令行输入 echo \$localenv 也可以得到实际的环境变量目录。

3，注册快捷键以方便调试，编辑 ENV 文件, 添加如下两行，F11 为载入 test.il, F12 为执行 test 命令。

```
alias F11 skill load('test.il')
```

```
alias F12 test
```

如图：

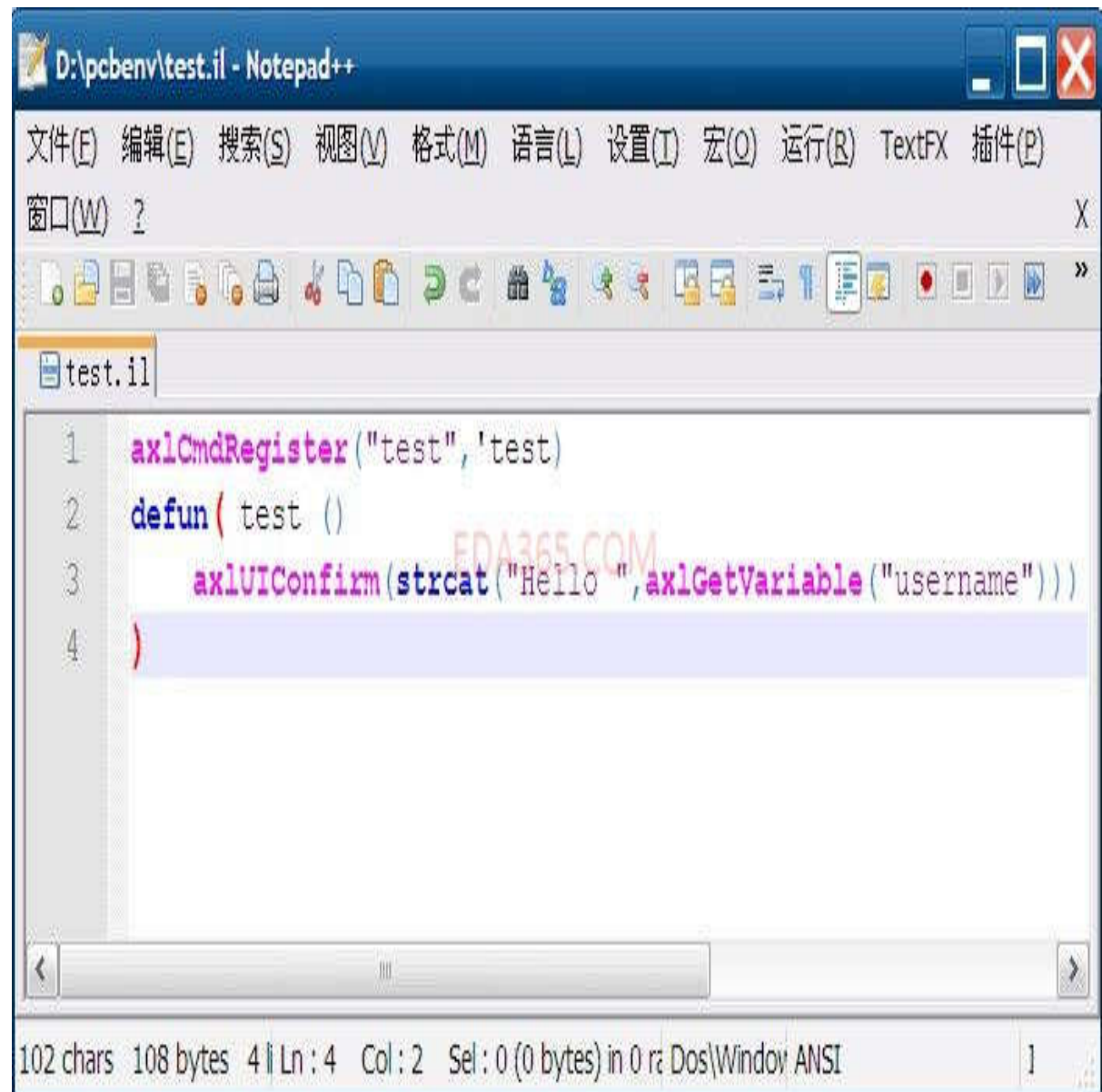


以后，我们就可以在 test.il 文件中书写代码，保存后，在 Allegro 中按 F11 就会载入，然后按 F12 就可以执行命令 test, 后面会讲到这个 test 命令是如何注册的。

4，来完成第一个 Hello World 程序。

用文本编辑器打开 test.il 文件，将以下蓝色部分代码复制到 test.il 文件中并

保存，



The screenshot shows a Notepad++ window titled "D:\pcben\test.il - Notepad++". The menu bar includes "文件(F)", "编辑(E)", "搜索(S)", "视图(V)", "格式(M)", "语言(L)", "设置(I)", "宏(O)", "运行(R)", "TextFX", and "插件(P)". The toolbar contains various icons for file operations and editing. The editor window shows a file named "test.il" with the following code:

```
1  axlCmdRegister("test", 'test')
2  defun( test ()
3      axlUIConfirm(strcat("Hello ", axlGetVariable("username")))
4  )
```

The status bar at the bottom indicates "102 chars 108 bytes 4 | Ln: 4 Col: 2 Sel: 0 (0 bytes) in 0 rz Dos\Windows ANSI".

<-----我是分隔线----->

```
axlCmdRegister("test", 'test')
```

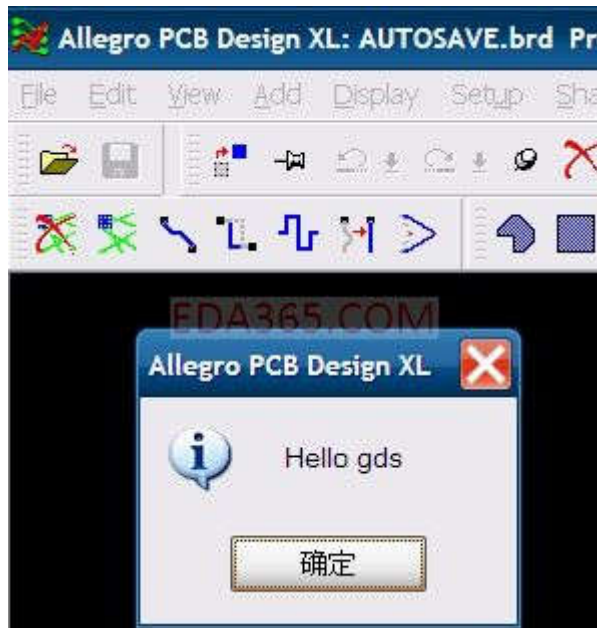
```
defun( test ()
```

```
    axlUIConfirm(strcat("Hello ", axlGetVariable("username")))
```

```
)
```

<-----我是分隔线----->

回到 allegro 界面，按 F11, 再按 F12. 看下执行效果。



这个 Hello World 程序用到了 `axlUIConfirm`, `strcat`, `axlGetVariable` 这几个函数。

`axlGetVariable("username")` 来获取 Username

`strcat`: 在本例中将 Hello 这个字符串与用户名合并成一个字符串。

`axlUIConfirm`: 弹出提示框，内容为刚才 `strcat` 合并后的字符串。

今天先到这里，大家先把这个 Hello world 程序调试出来。另外这个系列教程是让大家入门的，大家觉得写的浅也不要责怪。

有时间可以先看下以下帮助文件：

[X:\Cadence\SPB\\_15.7\doc\sklanguser\sklanguser.pdf](#)

[X:\Cadence\SPB\\_15.7\doc\sklangref\sklangref.pdf](#)

[X:\Cadence\SPB\\_15.7\doc\algroskill\algroskill.pdf](#)

十天学会 Skill 之第二天

By deargds

<http://www.eda365.com>

学习目的：了解 SKILL 的一些基础，变量，函数，以及流程控制。

一，我们先了解以下几种数据类型，

整数：integer 例如：2, 8, 6

浮点数：float 例如：2.3, 9.3265, 0.5, .8,

字符串：string 例如：“Hello!”

原子类型:atom 例如：t, nil, unbound

链表：list 例如:`list(1,2,3)` '(1,2,3)

给变量赋值，除了使用=之外，还可以使用 `set` 函数.

`set(a,"hello")`

数据类型可使用 `type` 函数来查询

例如:`a = 1`

那么 `type(a)`的返回值为 `fixnum`

`a = "string"`

`type(a)`返回值为 `string`

另外链表类型非常有用，在 skill 中有一系列的函数是针对链表操作的,示例如下:

`a = list(5,6,7)`;定义链表 a，内容为 5 6 7.

如果想得到链表中第一个值，使用 `car` 函数即可，`car(a)`的结果为 5

另外还有 `append1`,`cons`, `remove`,`reverse`,`nth` 等许多函数，大家自己先看 HELP 吧。

X:\Cadence\SPB\_15.7\doc\sklanguser\sklanguser.pdf

其它数据类型如 `symbol`,`defstruct`,`array` 等请参考

X:\Cadence\SPB\_15.7\doc\sklanguser\sklanguser.pdf

二，函数的定义

1.通常使用 `defun` 或者 `procedure` 来自定义一个函数，两者的作用是一样的，区别只在乎在书写格式的差别

`defun(test ()`

`;you can do something`  
`)`

`procedure(test()`

`;you can do something`  
`)`

上面两句同样是定义了一个 `test` 函数,`defun` 定义时在函数名后要有一个空格。

2.局部变量的定义，使用 `let` 或 `prog` 函数，如果不需要在函数内使用跳转函数或者返回值的

话使用 `let` 即可。

```
defun(test ()  
  let((a)  
    ;you can do something  
  )  
)
```

上面示例中使用 `let` 定义了 `a` 为局部变量，其它函数无法访问。

### 3.函数的参数类型@rest @optional @key

`@rest` 不确定数目的参数，可以有多个参数。

```
defun( test ( @rest a )  
  println(a)  
)
```

`@optional` 可选择的参数，在调用该函数时，可以忽略此参数。

```
defun(test (@optional a )  
  println(a)  
)
```

`@key` 指定参数的固定值。

```
defun(test (a b @key (c 3))  
  println(c)  
)
```

ps:`@optional` 与 `@key` 不能同时出现在函数参数定义中。

## 三，流程控制

1.条件判断 `if` `when` `cond` `unless`，  
双向判断

```
if( 3 > 1 then  
  println("You know too much!")  
else  
  println("hehe")  
)
```

```
when( 3 > 1  
  println("You know too much!")  
)
```

```
unless( 3 < 1  
  println("You know too much!")  
)
```

;多向判断

```
cond(  
    (2 > 0 ,println(2))  
    (3 < 0 ,println(3))  
)
```

2.循环语句 while for foreach

;while 示例

i = 0;定义 i 的初值为 0

```
while((i <= 10)  
    i = i + 1;自加一次  
    println(i)  
)
```

;for 示例

sum = 0;定义 sum 的初值为 0

```
for( i 1 10  
    sum = sum + i;自加一次  
    println(sum)  
)
```

;foreach 语句，方便用来循环链表类型

```
foreach( x '(1 2 3 4)  
    println(x)  
)
```

3.选择语句 case caseq

i = 5 + 6

```
case( i  
    (5 println("no"))  
    (11 println("yes"))  
    (t 'Other)  
)
```

上面简单介绍需要了解的基本知识，另外想学好 SKILL,帮助文件是一定要看的，这个教程只是帮大家入门，不可能过多地去重复教程上已经有的内容。

X:\Cadence\SPB\_15.7\doc\sklanguser\sklanguser.pdf

X:\Cadence\SPB\_15.7\doc\sklangref\sklangref.pdf

学习目的：了解 Skill 与 Allegro 交互接口，创建一个简单的窗体实例。

一，了解 Skill 中常用的一些交互函数

`axlUIYesNo("Are you ready?");`弹出 YES NO 选择框

`axlUIYesNoCancel("Are you ready?");`弹出 YES NO CANCEL 三态选择框

以上函数会根据用户所选结果返回一个值，我们可以判断这个返回值来进行相关的操作。



`axlUIMultipleChoice("Pick a choice" list("a" "b" "c"));`多项选择



`axlUIPrompt("Enter Your name" "Cat" );`提示用户输入内容并返回

`axlUIConfirm("Hello" );`弹出提示框



以上两个函数组合一下执行

`axlUIConfirm(axlUIPrompt("Enter Your name" "Cat" ))`

`axlMeterCreate;`创建进度条，需与其它函数配合使用

`axlMeterUpdate`  
`axlMeterDestroy`

创建进度条示例：

```
defun(test ()  
    axlMeterCreate("test", "", t);创建进度条  
    for(n 1 4  
        sleep(1);延迟 1 秒  
        axlMeterUpdate(n*25 "test");更新进度  
    )  
    axlMeterDestroy();关闭进度条  
)
```



弹出 NET 选择框

```
axlUIDataBrowse('NET' (RETRIEVE_NAME) "test" t)
```

建议查看下 `axlUIDataBrowse` 函数的详细说明

二，创建程序交互窗口

Skill 的窗体是通过对应的 form 文件来进行描述的，每个不同的窗体对应不同的 FORM 文件，里面包括了窗体的属性，每个控件的位置，大小，相关的属性等描述。

FORM 文件内容示例：

```
FILE_TYPE=FORM_DEFN VERSION=2
```

```
FORM
```

```
FIXED
```

```
PORT 42 5
```

```
HEADER "Progress Meter"
```

```
TITLE
```

```
TEXT
```

```
TLOC 1 3
```

```
INFO progressText 40
```

```
ENDTEXT
```



```
FIELD bar
FLOC 1 1
PROGRESS 41 3
ENDFIELD
```

```
ENDTILE
```

```
ENDFORM
```

SKILL 内置了一些 WIN32 控件，如按钮，文本框，单选，复选，列表，树状列表，下拉列表，进度条等，  
在 FORM 文件中，每个控件都有自己的 FIELD 名称，可以通过 FIELD 来访问该控件以获取和设置控件的值。

在编写好 Form 文件后，可以通过 `axlFormTest()` 函数来进行测试，

例如

```
axlFormTest("C://Cadence//SPB_15.7//share//pcb//text//forms//createsp  
litplane.form")
```

实际的 Skill 程序中为了方便，通常都会使用动态生成 Form 文件的方法来创建窗体，结合前面了解过的文件输入输出端口，比较容易实现。

示例如下：



```
defun( test ()
    f = "/test.form"; 定义 test.form 文件
    p = outfile(f "w")
    fprintf(p "FILE_TYPE=FORM_DEFN VERSION=2\n")
    fprintf(p "FORM\n")
    fprintf(p "FIXED\n")
    fprintf(p "PORT 22 8\n")
    fprintf(p "HEADER \"TEST\"\n")
    fprintf(p "TILE\n")
    fprintf(p, "FIELD tip\n\n"); 开始写入文本框定义
    fprintf(p, "FLOC 1 1\n"); 定义坐标位置
    fprintf(p, "STRFILLIN 15 40\n"); 控件尺寸
```

```

fprintf(p, "ENDFIELD\n\n")
fprintf(p "FIELD pop\n");开始写入按钮 POP 定义
fprintf(p "FLOC 1 4\n")
fprintf(p "MENUBUTTON \"Pop\" 8 3\n")
fprintf(p "ENDFIELD\n")
fprintf(p "FIELD close\n");开始写入按钮 EXIT 定义
fprintf(p "FLOC 10 4\n")
fprintf(p "MENUBUTTON \"Exit\" 8 3\n")
fprintf(p "ENDFIELD\n")
fprintf(p "ENDTITLE\n")
fprintf(p "ENDFORM\n")
close(p);test.form 文件写入结束

h = axlFormCreate((gensym) f nil 'test_call t);调用
axlFormCreate 函数根据 test.form 文件创建窗体，并关联窗体回调函数
test_call

axlFormDisplay(h);显示窗体
)

(defun( test_call (h)
  case( h -> curField ;获取当前窗体激活的控件
    ("pop"
      axlUIConfirm(axlFormGetField(
h "tip")));获取 FIELD 为 tip 的控件内容，并弹出
    )
    ("close"
      axlFormClose(h);关闭窗体
      deleteFile(f);删除生成的 FORM
文件
    )
  )
)
)

```

在回调函数中经常用到的函数及属性

`axlFormSetField`;设置控件值

`axlFormGetField`;获取控件值

`axlFormSetFieldEditable`;禁用/允许控件

`axlFormSetFieldVisible`;隐藏/显示控件

`formHandle -> curField`;返回窗体当前活动的控件。

`formHandle -> curValue`;返回窗体当前活动控件的值，类型与 FORM 文件中所定义类型一致，也可以通过 `axlFormCreate` 定义为 string 类型。

另 外 在 帮 助 文 件

X:\Cadence\SPB\_15.7\share\pcb\examples\skill\form\basic 目录下有非常详细的 FORM 文件示例，可以参考一下。

十天学会 Skill 之第三天

By deargds

<http://www.eda365.com>

学习目的：如何操作，读写文件。

## 一，创建，删除目录等

1. 在当前打开的目录下创建 test 子目录，也可以使用绝对路径。

```
createDir("./test")
```

```
createDir("c:/test")
```

2. 删除目录，也可以使用绝对路径。

```
deleteDir("./test")
```

```
deleteDir("c:/test")
```

3. 检查目录是否存在

```
isDir("c:/test")
```

4. 检查目录是否可读写

```
isReadable("c:/test")
```

```
isExecutable("c:/test")
```

5. 同样也可以用 isReadable, isExecutable 来判断文件是否可读写。

c:/test/1.txt 实际不存在，所以以下执行结果返回为 nil, 所以在读写文件之前我们可以用来作判断。

```
isReadable("c:/test/1.txt")
```

```
isExecutable("c:/test/1.txt")
```

ps: 另外还有 isFileName, isFile, isLargeFile, isExecutable 等相关函数请参考帮助

## 二，如何读取文件

1. 首先要定义一个文件输入端口类型

c:/test/1.txt 文件的实际内容如下：

```
<----c:/test/1.txt---->
```

```
Monday
```

```
Tuesday
```

```
Wednesday
```

```
Thursday
```

```
Friday
```

```
Saturday
```

```
Sunday
```

```
<----c:/test/1.txt---->
```

现在我们将每行的内容读出并在 ALLEGRO 界面中弹出提示。

; 首先使用 infile 定义一个文件输入端口类型 weekport,

```
weekport = infile("c:/test/1.txt")
;现在用 gets 对 weekport 进行读取
gets(i, weekport)
;现在文件的第一行就读到变量 i 中了，下面直接调用 AXL 函数弹出
axlUIConfirm(i)
```



```
;使用 while 循环读出每一行,直到文件结尾
while( gets(i, weekport)
axlUIConfirm(i);会弹出每一行的内容
)
;在读取结束后，要记得使用 close 函数关闭端口
close(weekport)
```

### 三，如何写入文件

与读取类似，写入文件需要用 outfile 定义一个写入端口

```
;写入到"c:/test/1.txt" 中，如果是添加模式的话 outfile 要用参数"a"
wport = outfile("c:/test/1.txt")
;写入端口
fprintf(wport, "Hello, world!")
;关闭端口
close(wport)
;c:/test/1.txt 文件中就已经写入 Hello world 这行内容了。
```

以上大概介绍了读写文件的一般流程，大家自己可以对照帮助看一下相关函数的用法，特别是格式化输出函数 `fprintf`。

十天学会 Skill 之第四天

By deargds

<http://www.eda365.com>

学习目的：了解 AXL 函数，Allegro PCB 用户模型。

## 一，认识 AXL 函数

AXL 函数就是 Allegro 提供的一系列设计接口函数，通过这些函数，我们能够对设计文件及用户接口进行互动操作，来实现各种不同的功能。

### 1.部分 AXL 示例:

`axlUIConfirm("Hello,world!");`弹出提示框

`axlUIControl('screen');`获取屏幕分辨率

`axlUIDataBrowse('NET '(RETRIEVE_NAME) "hi" t);`弹出 NET 选择框

`axlVisibleGet();`获取当前所有显示的层面

`axlCNSEcsetCreate("Test");`创建名为 test 的 ECSET

`axlChangeWidth(line_id, 20.0);`改变线宽

...

### 2.什么是 DBID

设计文件就是一个储存了实际设计信息与逻辑信息的数据库，其中的每一个对象都有自己唯一的 ID，我们可以通过对这些 ID 来操作对象，跟 DBID 相关的一些函数

`axlIsDBIDType(ID);`判断给定的变量是否是 DBID 类型

`axlDbidName(ID);`返回 DBID 的名称

在获取一个对象的 DBID 后，我们可以获得该对象的一些属性，比如层面，端点坐标，半径，中心点，宽度等等，不同的 DB 类型有不同的属性。

示例:获取当前设计中的所有 NET 的名称列表,并保存到当前目录下的 allnet.txt 中,参考上次写文件的示例。

`all = axlDBGetDesign() -> nets;`获取所有 NET 的 DBID

`wport = outfile("./allnet.txt");`定义输入端口

`foreach(n all`

`fprintf(wport,n -> name);`获得 NETNAME

)

`close(wport);`关闭端口

## 二，Allegro PCB 用户模型

PCB 的设计文件是一个包含了所有设计信息的数据库，Allegro PCB 用户模型封装了数据库的一些信息，我们可以通过一些对象来获取我们需要得到的内容。

示例:

获取所有 component

`axlDBGetDesign() -> components`

获取所有 net

`axlDBGetDesign() -> nets`

获取所有 matchgroup

```
axlDBGetDesign() -> matchgroup
```

如果要访问其它一些对象, 如 TEXT,CLINE,VIA 等, 则可以使用 Selection and Find Functions.

示例:

```
axlClearSelSet()
```

```
axlSetFindFilter( ?enabled (list "noall" "invisible" "clinesege") , ?onButtons (list "clinesege"));使  
用 AXL 函数 axlSetFindFilter 设置 FIND 过滤条件
```

```
all_cline = axlGetSelSet(axlAddSelectAll());全选,当前设计中所有的 clinesege 放入 all_cline 变  
量中。
```

```
;下面利用 foreach 循环来访问每一个 clinesege,
```

```
foreach(i all_cline
```

```
when(i->width < 4 ;判断线宽是否小于 4,如果小于 4 则高亮
```

```
axlHighlightObject(i) ;高亮 i
```

```
)
```

```
)
```

## 十天学会 Skill 之第五天

By deargds

<http://www.eda365.com>

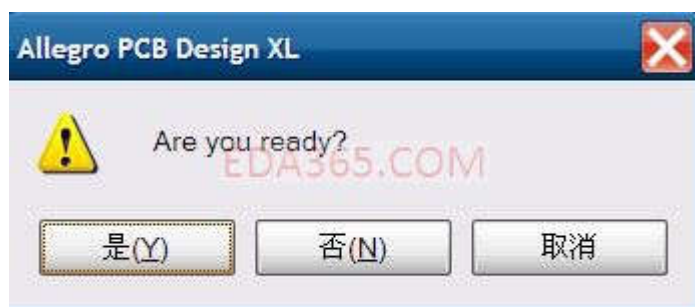
学习目的：了解 Skill 与 Allegro 交互接口，创建一个简单的窗体实例。

一，了解 Skill 中常用的一些交互函数

`axlUIYesNo("Are you ready?");`弹出 YES NO 选择框

`axlUIYesNoCancel("Are you ready?");`弹出 YES NO CANCEL 三态选择框

以上函数会根据用户所选结果返回一个值，我们可以判断这个返回值来进行相关的操作。



`axlUIMultipleChoice("Pick a choice" list("a" "b" "c"));`多项选择



`axlUIPrompt("Enter Your name" "Cat" );`提示用户输入内容并返回

`axlUIConfirm("Hello" );`弹出提示框



以上两个函数组合一下执行

```
axlUIConfirm(axlUIPrompt( "Enter Your name" "Cat" ))
```

`axlMeterCreate`;创建进度条，需与其它函数配合使用

`axlMeterUpdate`

`axlMeterDestroy`

创建进度条示例：

```
defun(test ()
    axlMeterCreate("test", "", t);创建进度条
    for(n 1 4
        sleep(1);延迟 1 秒
        axlMeterUpdate(n*25 "test");更新进度
    )
    axlMeterDestroy();关闭进度条
)
```

弹出 NET 选择框

```
axlUIDataBrowse('NET ' (RETRIEVE_NAME) "test" t)
```

建议查看下 `axlUIDataBrowse` 函数的详细说明

## 二，创建程序交互窗口

Skill 的窗体是通过对应的 form 文件来进行描述的，每个不同的窗体对应不同的 FORM 文件，里面包括了窗体的属性，每个控件的位置，大小，相关的属性等

描述。

FORM 文件内容示例：

```
FILE_TYPE=FORM_DEFN VERSION=2
FORM
FIXED
PORT 42 5
HEADER "Progress Meter"

TILE

TEXT
TLOC 1 3
INFO progressText 40
ENDTEXT

FIELD bar
FLOC 1 1
PROGRESS 41 3
ENDFIELD

ENDTILE

ENDFORM
```

SKILL 内置了一些 WIN32 控件，如按钮，文本框，单选，复选，列表，树状列表，下拉列表，进度条等，

在 FORM 文件中，每个控件都有自己的 FIELD 名称，可以通过 FIELD 来访问该控件以获取和设置控件的值。

在编写好 Form 文件后，可以通过 `axlFormTest()` 函数来进行测试，

例如

```
axlFormTest("C://Cadence//SPB_15.7//share//pcb//text//forms//createsp  
litplane.form")
```

实际的 Skill 程序中为了方便，通常都会使用动态生成 Form 文件的方法来创建窗体，结合前面了解过的文件输入输出端口，比较容易实现。

示例如下：

```
defun( test ()
    f = "/test.form"; 定义 test.form 文件
    p = outfile(f "w")
    fprintf(p "FILE_TYPE=FORM_DEFN VERSION=2\n")
    fprintf(p "FORM\n")
    fprintf(p "FIXED\n")
    fprintf(p "PORT 22 8\n")
    fprintf(p "HEADER \"TEST\"\\n")
    fprintf(p "TILE\n")
    fprintf(p, "FIELD tip\\n\\n"); 开始写入文本框定义
    fprintf(p, "FLOC 1 1\\n"); 定义坐标位置
    fprintf(p, "STRFILLIN 15 40\\n"); 控件尺寸
    fprintf(p, "ENDFIELD\\n\\n")
    fprintf(p "FIELD pop\\n"); 开始写入按钮 POP 定义
    fprintf(p "FLOC 1 4\\n")
    fprintf(p "MENUBUTTON \"Pop\" 8 3\\n")
    fprintf(p "ENDFIELD\\n")
    fprintf(p "FIELD close\\n"); 开始写入按钮 EXIT 定义
    fprintf(p "FLOC 10 4\\n")
    fprintf(p "MENUBUTTON \"Exit\" 8 3\\n")
    fprintf(p "ENDFIELD\\n")
    fprintf(p "ENDTILE\\n")
    fprintf(p "ENDFORM\\n")
    close(p); test.form 文件写入结束

    h = axlFormCreate((gensym) f nil 'test_call t); 调用
axlFormCreate 函数根据 test.form 文件创建窗体，并关联窗体回调函数
test_call

    axlFormDisplay(h); 显示窗体
)

defun( test_call (h)
```

```

        case( h -> curField ;获取当前窗体激活的控件
            ("pop"
                axlUIConfirm(axlFormGetField(
h "tip"));获取 FIELD 为 tip 的控件内容，并弹出
            )
            ("close"
                axlFormClose(h);关闭窗体
                deleteFile(f);删除生成的 FORM
文件
            )
        )
    )
)

```

在回调函数中经常用到的函数及属性

[axlFormSetField](#);设置控件值

[axlFormGetField](#);获取控件值

[axlFormSetFieldEditable](#);禁用/允许控件

[axlFormSetFieldVisible](#);隐藏/显示控件

formHandle -> curField;返回窗体当前活动的控件。

formHandle -> curValue;返回窗体当前活动控件的值，类型与 FORM 文件中所定义类型一致，也可以通过 axlFormCreate 定义为 string 类型。

另 外 在 帮 助 文 件  
X:\Cadence\SPB\_15.7\share\pcb\examples\skill\form\basic 目录下有非常  
详细的 FORM 文件示例，可以参考一下。