Скрипт-проект Altium Designer

Владислав Суходольский, эксперт по Altium компании «Бюро ESG»,доцент кафедры микроэлектроники СПБГЭТУ «ЛЭТИ»

Скрипт-проект Altium Designer (<имя проекта>. PrjScr) представляет собой текстовое программное приложение, использующее так называемый интерфейс программирования приложений — API (Application Programming Interface) для управления процессами и модификации объектов в открытых проектах других видов (PCB, FPGA и др.).

Скрипт-система Altium Designer позволяет формировать программное приложение (скрипт) двух видов:

1. Script Unit — скрипт-единица — программное приложение, управляющее модификацией глобальных настроек (DXP Preferences), объектов в схемных документах, библиотеках или документе печатной платы выбранного активного проекта. В простейшем случае — это текстовое сообщение, выводимое на экран монитора поверх документа активного проекта.
2. Script Form — скрипт-форма — программное приложение, включающее диалоговое окно, которое пользователь программирует для управления работой собственно скрипт-программы, включая в него виртуальные кнопки и другие экранные функции.

Для скрипт-программирования доступны шесть языков высокого уровня:

- Borland Delphi Pascal — (DelphiScript);
- Java — (JScript);
- Visual Basic — (VBScript);
- Enable Basic — (EnableBasicScript);
- TCL — язык программирования TCL;
- Query Language — язык запросов Altium Designer.

В частности, входящая в поставку программного комплекса Altium Designer система формирования скриптов на языке Delphi включает следующие варианты скрипт-проектов:

- General Scripts — скрипты общего назначения — вывод на экран текстовых надписей, изображений, модификация ini-файлов, архивация документов и др.;
- DXP Scripts — скрипты, управляющие настройками DXP-платформы;
- PCB Scripts — скрипты, управляющие модификацией объектов в документах печатной платы;
- SCH Scripts — скрипты, управляющие модификацией объектов в схемных документах;
- Processes Scripts — скрипты, управляющие процессами Altium Designer;
- Workspace Manager Scripts — скрипты, управляющие модификацией объектов рабочего пространства проектов.

В версиях программного комплекса по Summer 09 в поставку, в директорию ...\Examples\Scripts\ включалась «Галерея примеров» из более чем 270 примеров скрипт-программ, позволяющих освоить систему скрипт-программирования в среде Altium Designer. В новых версиях, начиная с AD 10, данные примеры располагаются на интернет-сайте производителя www.altium.com.

Прежде всего ознакомимся с терминологией, используемой в скрипт-системе Altium Designer.

- **API** (**Application Programming Interface**) — так называемый интерфейс программирования приложений, используемый для модификации объектов в документах проектов;
- интерфейсы **API** автоматически доступны для использования в скриптах;
- **Altium Designer Runtime Library** — библиотека интерфейсов **API**;
- каждая программа-редактор Altium Designer имеет свой интерфейс **API**. Например, редактор печатных плат имеет интерфейс **PCB API**, редактор схем — интерфейс **Schematic API** и т.д.;
- **Object Model** — модель объекта, используемая интерфейсом **API**, состоит из интерфейсов объектов (**Object Interface**);
- **Object Interface** — интерфейс объекта. Состоит из методов (**Methods**) и свойств (**Properties**) и служит для отображения объектов, используемых в скриптах с целью извлечения и модификации данных в открытых проектных документах;
- **Methods** (Методы) — действия, которые может выполнять интерфейс объекта;
- **Properties** (Свойства) интерфейса объекта отображают данные, содержащиеся в объекте, представляемом интерфейсом;
- **Script Form** (Скрипт-форма) — диалоговое окно, программируемое пользователем с целью управления функциями скрипт-программы;
- **Components** (Компоненты) — видимые в окне скрипт-формы объекты, помещаемые в форму из плавающей панели **Tool Palette** и служащие для управления работой скрипт-приложения. Называются также **Controls** (Управляющие воздействия);
- **Events** (События) — воздействия, на которые реагирует компонент в скрипт-форме;
- **Processes** (Процессы) — командные строки, применяемые для исполнения команд скрипт-приложения.

Формирование и исполнение скрипт-единицы

1. Командой главного меню **File** → **New** → **Project** → **Script Project** образовать новый скрипт-проект (рис. 1).

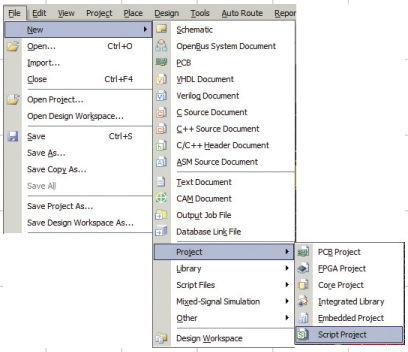

Рис. 1. Образование нового скрипт-проекта

2. Щелчком правой кнопки на имени проекта Script_Project_1.PrjScr в панели **Projects** активизировать контекстное меню и командой **Save Project As** сохранить проект с новым именем.
3. Активизировать контекстное меню аналогично второму пункту и указать в нем цепочку подкоманд формирования скрипт-единицы (в нашем случае на языке программирования Delphi) **Add New to Project** → **Delphi Script Unit** (рис. 2).
4. В панели *Projects* к дереву скрипт-проекта присоединяется узел EditScript1.pas; одновременно в главном окне программы открывается одноименное поле для записи текста.

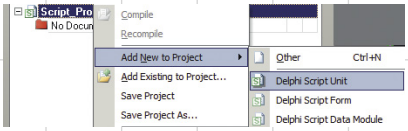

Рис. 2. Команда образования скрипт-единицы

5. Предположим, что мы хотим вывести на экран приветствие: «Добро пожаловать в скрипт-систему Altium Designer!» Для этого нужно ввести с клавиатуры в поле EditScript1.pas следующий текст скрипт-единицы:

```
Procedure ShowAMessage;
Var
DefaultMessage;
Begin
DefaultMessage := 'Добро пожаловать в скрипт-систему Altium Designer!';
ShowMessage(DefaultMessage);
End;
```

6. Командой **File** → **Save As** сохранить текстовый файл скрипт-единицы с новым именем, например, Welcome_to_AD.pas.
7. Для исполнения скрипт-единицы — вывода на экран приветствия — выполнить команду главного меню **DXP** → **Run Script**,

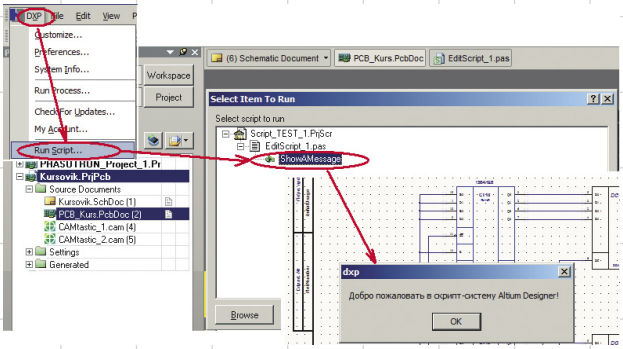

Рис. 3. Исполнение скрипт-единицы

после чего в диалоговом окне выбора единицы для исполнения Select Item To Run указать имя процедуры ShowAMessage (рис. 3).

8. Поверх открытого документа активного PCB-проекта в окне DXP выводится запрограммированное сообщение.

Формирование и исполнение скрипт-формы

Образуем скрипт-форму для вывода на экран того же приветствия «Добро пожаловать в скрипт-систему Altium Designer!» под управлением двух экранных кнопок: **Display** (Показать) и **Close** (Закрыть). Для образования скрипт-формы в составе активного скрипт-проекта следует:

1. Щелчком правой кнопки на имени проекта активизировать контекстное меню, указать в нем команду **Add New To Project** и выбрать подкоманду **Delphi Script Form**. К дереву документов проекта присоединяется новый узел EditScript1.pas.
2. Командой меню **File** → **Save As** сохранить документ скрипт-формы с новым именем, например, Welcome_Form.pas.
3. В главном окне программы появляются две вкладки: **Code** — поле для составления текста программного приложения и **Form** — поле редактирования

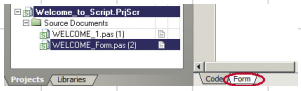

Рис. 4. Дерево проекта и вкладки главного окна программы

состава функций диалогового окна скрипт-формы. Выбираем вкладку **Form** (рис. 4) и переходим к редактированию диалогового окна скрипт-формы.

4. Кнопкой **Script** в обрамлении главного окна программы активизировать панель «Инспектор объектов» — **Object Inspector** (рис. 5).
5. На вкладке **Properties** панели Инспектора выполним следующие настройки:

- в строку **Caption** (Заголовок) вводим имя для диалогового окна скрипт-формы, например Welcome, которое сразу отображается в заголовке окна;
- в строку **Name** вводим имя процедуры исполняемого программного приложения WelcomeForm1; после нажатия клавиши **Enter** в верхнем поле панели Инспектора отображается строка с именем процедуры:

WelcomeForm1:TWelcomeForm1

6. В нашем примере необходимо внести в диалоговое окно скрипт-формы две экранные кнопки. Для этого:

- кнопкой **Script** в правом нижнем углу главного окна программы активизируем контекстное меню и открываем панель рабочего пространства **Tool Palette** (рис. 6, правая панель), в которой размещены более ста пиктограмм инструментов управления для скрипт-формы (компонентов);
- выбираем в панели **Tool Palette** компонент TButton, перемещаем курсор в окно скрипт-формы и фиксируем кнопку Button1 в поле окна щелчком левой кнопки мыши;
- повторяем это действие еще раз и помещаем в окно скрипт-формы вторую кнопку — Button2 (рис. 6, в центре).

7. Далее следует присвоить кнопкам имена и обозначить выполняемые ими функции. Для этого:

- выбрать курсором кнопку Button1 (изображение кнопки отмечается значками-«прищепками»);

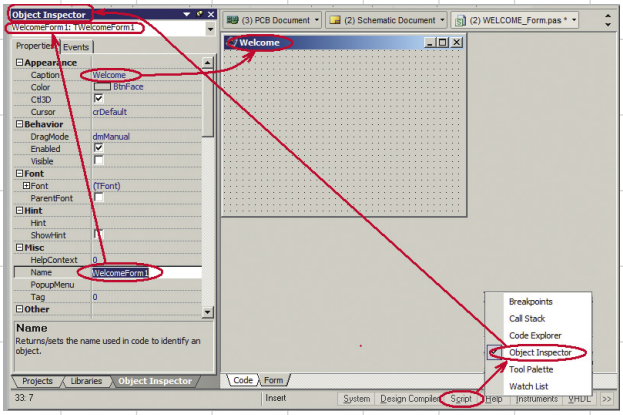

Рис. 5. Редактирование свойств скрипт-формы

Бюро ESG
Altium
Altium Designer
Новое поколение САПР для электронного проектирования

Реклама
Единая среда проектирования
АКЦИЯ
«Специальное предложение по переходу с P-CAD на Altium Designer 2013»
Только до 30 июня 2013 года!
Справки по телефону +7 812 496 6929
ООО "Бюро ЕСГ"
Авторизованный реселлер компании Altium
Санкт-Петербург,
ул. Белоостровская 28
т. +7 (812) 496-6929
ф. +7 (812) 496-5272
Email: esg@esg.spb.ru
Internet: www.esg.spb.ru

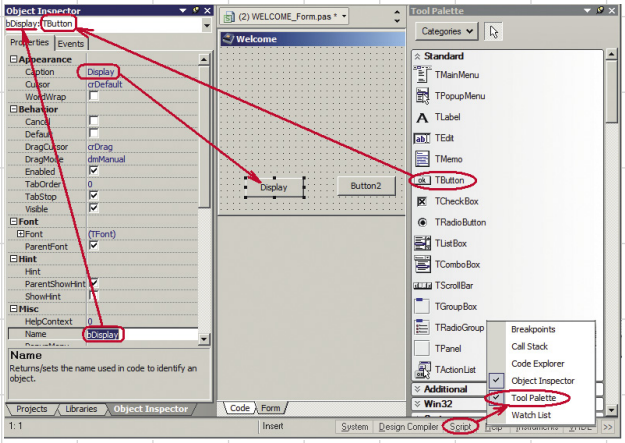

Рис. 6. Назначение инструментов управления для скрипт-формы

- в панели **Object Inspector** (рис. 6, левая панель) отображаются свойства кнопки;
- в строку **Caption** ввести имя кнопки — **Display**;
- в строку **Name** ввести обозначение функции кнопки: bDisplay — в верхнем поле панели Инспектора отображается строка **bDisplay:TButton**;
- повторить все действия для второй кнопки, указав при этом имя Close и функцию bClose.

8. Теперь необходимо определить процедуры, которые выполняются по щелчку кнопок управления. Для этого:
- выбрать курсором кнопку **Display** и перейти в панели Инспектора на вкладку **Events** (События);
- выбрать на вкладке **Events** (События Mouse → OnClick (по щелчку мыши) и двойным щелчком в поле с именем функции bDisplayClick открыть вкладку **Code** в главном окне программы;
- на вкладке **Code** открываются составленные программой при редактировании свойств кнопок процедуры обработки событий — щелчков на кнопках **Display** и **Close** (рис. 7);
- тела процедур пока пусты, поскольку действия, производимые щелчком по каждой из кнопок, еще не запрограммированы;
- вводим в текст на экране тело каждой из двух процедур: для первой процедуры — это оператор ShowMessage — показать сообщение, а для второй — оператор Close (Закрыть), в результате чего текст на экране приобретает вид:

```
procedure TWelcomeForm1.
bDisplayClick(Sender:
TObject);
begin
ShowMessage('Добро
пожаловать в скрипт-систему
Altium Designer!');
end;

procedure TWelcomeForm1.
bCloseClick(Sender: TObject);
begin
Close;
end;
```

9. В текст скрипт-приложения необходимо еще включить процедуру, дающую старт исполнению скрипт-формы по команде **DXP → Run Script:**

```
Procedure RunWelcome_Form;
Begin
WelcomeForm1.ShowModal;
End;
```

10. Сохраним отредактированный файл скрипт-формы и командой **DXP → Run Script** запускаем ее исполнение:
- в открывающемся окне выбора скрипта для исполнения (рис. 8) выбираем процедуру Welcome_Form.pas>RunWelcome_Form;
- по щелчку на **OK** открывается диалоговое окно скрипт-формы Welcome;
- по щелчку на кнопке **Display** открывается окно DXP с приветствием: «Добро пожаловать в скрипт-систему Altium Designer!»;
- по щелчку на **OK** окно DXP с приветствием закрывается, но исполнение скрипт-формы на этом не заканчивается: повторным щелчком по кнопке **Display** вывод приветствия может быть повторен;
- если окно с приветствием закрыто, щелчком по кнопке **Close** исполнение скрипт-формы прекращается.

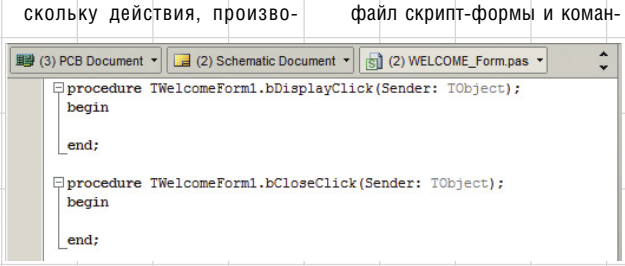

Рис. 7. Заготовка процедур обработки событий

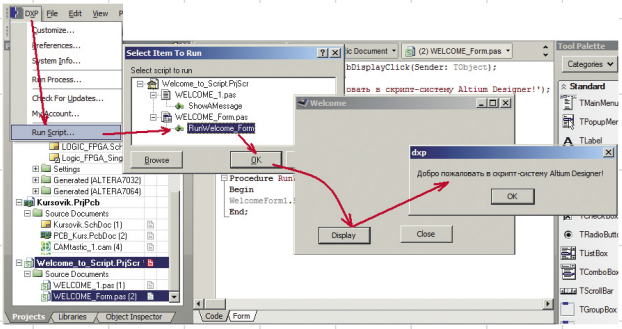

Рис. 8. Исполнение скрипт-формы

Покажем еще один пример выполнения скрипт-формы из «Галереи примеров», поставляемой с версией Altium Designer Summer 09, — редактирование диаметров отверстий на печатной плате, аналогичное выполнению в Altium Designer в панели PCB с использованием функции **Hole Size Editor**:

1. Откроем файл скрипт-проекта, расположенный в памяти компьютера по адресу: …\Examples\Scripts\DelphiScript Scripts\PCB\Hole Size Editor\Hole Size Editor.PrjScr. Дерево проекта включает две скрипт-формы и одну скрипт-единицу:
- HSForm.pas — главное скрипт-приложение, код которого содержит все основные процедуры редактирования, взаимодействующие через интерфейс ***PCB API*** с объектами PCB-документа;
- HSEditForm.pas — скрипт-форма, активизируемая из главной скрипт-формы HSForm.pas и осуществляющая непосредственное изменение диаметров отверстий;
- Customsort.pas — скрипт-единица, также активизируемая из главной скрипт-формы HSForm.pas и осуществляющая сортировку отверстий по размеру.

Программный код упомянутых скриптов не приводится, поскольку для его понимания от читателя требуется детальное знакомство

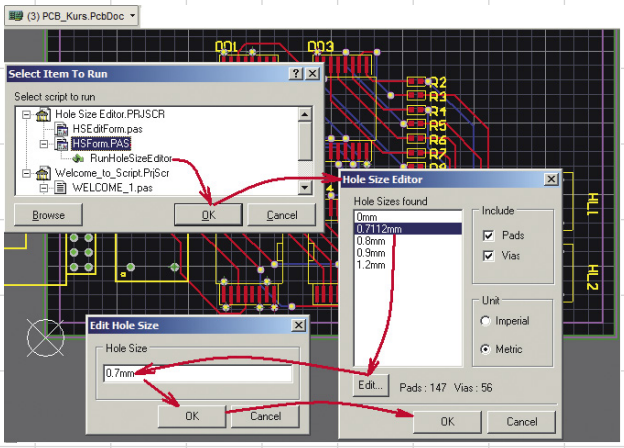

Рис. 9. Редактирование диаметров отверстий в PCB-документе

с языком программирования DelphiScript, а научить ему в рамках нашей статьи не представляется возможным.

2. Активизируем документ печатной платы одного из PCB-проектов, доступных в панели Projects, и командой **DXP → Run Script** запускаем исполнение скрипт-формы.
3. В окне **Select Item To Run** (рис. 9) указываем пункт RunHoleSizeEditor. По щелчку на OK открывается диалоговое окно **Hole Size Editor** скрипт-формы HSForm.pas. Видим, что окно насыщено большим количеством компонентов (инструментов управления):

- в поле **Hole Sizes Found** (Найденные размеры отверстий) — компонент TListBox — отображен список размеров отверстий на печатной плате;
- в поле **Include** (Включить) — компонент TGroupBox — расположены гнезда с флажками включения в список отверстий контактных площадок (Pads) и переходных отверстий (Vias) — компоненты TCheckBox;
- в поле **Unit** (Единицы измерения) расположены «радиокнопки» выбора дюймовой (⦿ Imperial) и метрической систем единиц измерения (⦿ Metric) — компоненты TRadioButton;
- кнопки **Edit** (редактировать), **OK** и **Cancel** — знакомые нам по предыдущему примеру компоненты TButton;
- в нижней части окна размещаются сведения о полном числе отверстий на плате — компоненты TLabel.

4. Выберем в списке один из размеров (отверстия данного размера отмечаются на плате цветными метками селекции) и кнопкой **Edit** запустим редактирование.
5. В открывающемся окне **Edit Hole Size** скрипт-формы HSEditForm.pas изменяем диаметр отверстия и щелчками на **OK** последовательно закрываем окна обеих скрипт-форм, а с ними и операцию редактирования.

В заключение отметим, что здесь нам удалось получить лишь поверхностное знакомство с возможностями скрипт-системы Altium Designer. В данном изложении нет возможности рассмотреть такие вопросы, как синтаксис языков программирования скрипт-приложений, функции ***API***-интерфейса, модели объектов скрипт-системы, различные виды скриптов. Подробное рассмотрение всех этих вопросов можно найти в ХЕЛП-документах или документах «Центра знаний» (Knowledge Center) на сайте фирмы-производителя www.altium.com*.

По существу, скрипт-система представляет собой органически встроенный элемент клиент-серверной технологии, лежащей в основе всех функций Altium Designer, и является мощным средством управления, повышающим эффективность САПР при выполнении проектной работы и редактировании проектов и библиотек компонентной базы.

* Основные из этих документов:

- TU0121 Getting Started with Scripting. pdf
- TU0125 Building Script Projects. pdf
- GU0120 A Tour of the Scripting System. pdf
- TR0120 DelphiScript Reference. pdf
- GU0117 Using Altium Designer RTL.pdf
- TR0123 Script Examples Gallery Reference.pdf