

Алексей Бодягин

Методы интеграции изображения с IP-видеокамер в экранную форму SCADA-системы

В настоящее время видеонаблюдение широко используется практически во всех сферах деятельности человека. Видеоконтроль применяется на производстве для наблюдения за ответственными технологическими участками, в охранных системах, и, безусловно, он нашёл широкое применение в современных решениях для «умного» дома.

В настоящее время практически каждая комплексная система автоматизации инженерных систем здания включает в себя систему видеонаблюдения или нуждается в её внедрении. Это особенно актуально для проектов по автоматизации коттеджных поселков и элитных квартир, где конечные заказчики системы формируют технические требования, исходя из порой фантастических представлений о наборе функций «умного» дома. Такие проекты немыслимы без систем визуального контроля за территорией и внутренними помещениями.

Выбор видеокамеры

Сегодня на рынке систем видеонаблюдения широкое распространение получили IP-видеокамеры. Они представляют собой аналоговые камеры с интегрированным модулем преобразования аналогового сигнала в цифровой для передачи по протоколу TCP/IP. Они обладают всеми основными характеристиками аналоговых видеокамер, за исключением некоторых особенностей. Так, условия стандарта передачи данных Ethernet накладывают определённые ограничения, в первую очередь на длину сегмента сети. Предельное расстояние — порядка 100 м, хотя с принятием специальных мер это значение можно увеличить. Существуют также управляемые IP-видеокамеры и варианты IP-видеокамер с использованием беспроводного стандарта передачи данных Wi-Fi. Это может быть удобно в случае невозможности прокладки коммуникаций на объекте, при установке камеры на подвижном объекте либо в зоне действия развёрнутой сети Wi-Fi. К достоинствам IP-видеокамер относятся отсутствие в системе плат видеозахвата и дорогостоящего специализированного программного обеспечения; к недостаткам — как правило, худшие технические характеристики (чувствительность и разрешающая способность) по сравнению с аналоговыми камерами, а также высокая стоимость. Существуют IP-видеокамеры, не уступающие аналоговым по своим характеристикам, но их стоимость особенно высока. Несмотря на это использование IP-ви-

деокамер может быть оправданно в случае их монтажа на объекте с развёрнутой сетью Ethernet, имеющей необходимый запас по пропускной способности. Несомненно, что прогресс в производстве этого типа оборудования приведёт к улучшению характеристик и снижению его стоимости.

Главное достоинство IP-видеокамер — простота интеграции в существующую инфраструктуру как на физическом уровне (быстрое подключение к сети Ethernet), так и на прикладном уровне программного обеспечения. Практически все современные IP-видеокамеры имеют встроенный Web-интерфейс, позволяющий управлять камерой и просматривать изображение через Web-браузер, и ActiveX-объект, выступающий в роли своеобразного драйвера, транслирующего изображение с камеры в экранную форму, где он размещён.

Интеграция со SCADA-системами

Наличие ActiveX-объекта значительно облегчает задачу просмотра изображения с IP-видеокамеры на экранной форме SCADA-системы. Практически все современные SCADA-пакеты поддерживают импорт ActiveX-объектов на свои инструменты визуализации и обеспечивают выполнение инструкций на различных языках программирования, таких как Visual Basic for Applications, VBScript и JavaScript.

В данной статье все примеры использования SCADA-системы выполнены на основе программного продукта ICONICS GENESIS32, поставляемого компанией ПРОСОФТ и обладающего описанными функциями.

Перед началом работы с ActiveX-объектом его необходимо зарегистрировать в системе. Регистрация осуществляется с помощью команды `regsvr32`. В описанном примере данная команда имеет вид `regsvr32 c:\xplug.ocx` (все примеры, представленные в статье, приведены для семейства операционных систем MS Windows, совместимых с GENESIS32).

Запустим компонент GraphWorX и создадим новую экранную форму. Далее разместим на ней ActiveX-объект используемой IP-видеокамеры (в описанных в статье примерах используется видеокамера компании Advantech ADAM-6090LE). Для этого выберем команду Insert New Object меню Edit и создадим из библиотеки нашего ActiveX-Control новый объект экранной формы (класс объекта в браузере имеет имя Gif89, рис. 1). Казалось бы,

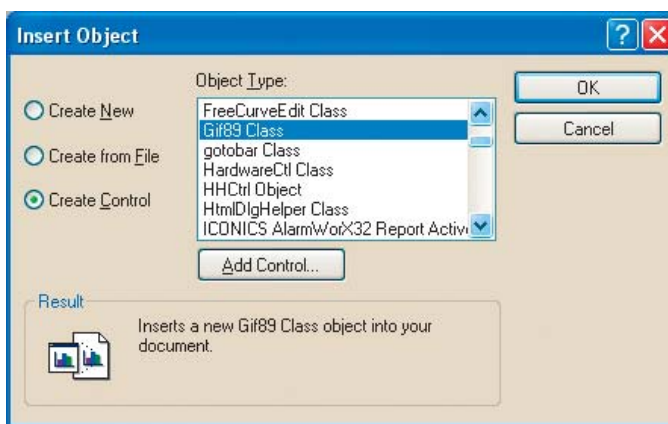


Рис. 1. Окно импорта OLE-объектов в экранную форму GraphWorX

задача решена, но после перехода в режим исполнения импортированный объект не отображает изображение с видеокamеры.

Дело в том, что после инициализации в режиме исполнения ActiveX-объект не транслирует изображение с видеокamеры до тех пор, пока не будет выполнен его метод Play (у многих других IP-видеокamер данный метод называется аналогично). Данная архитектура объекта вполне оправдана: до тех пор пока трансляция изображения с видеокamеры не инициирована оператором, ActiveX-объект не опрашивает видеокamеру и тем самым не создаётся избыточный трафик.

Выполнение метода объекта с помощью скрипта не представляет собой сложной задачи. На языке VBA эта инструкция будет выглядеть так: ThisDisplay.Gif89a1.Play (Gif89a — имя ActiveX-Control в библиотеке нашего объекта). Но тут мы сталкиваемся с новой проблемой, свойственной многим ActiveX-объектам: при подключении библиотеки нашего объекта начинают некорректно выполняться инструкции выбранного языка программирования. Причина в том, что производители оборудования создают ActiveX-объекты для корректной работы Web-интерфейса на стороне клиента или для своего программного обеспечения.

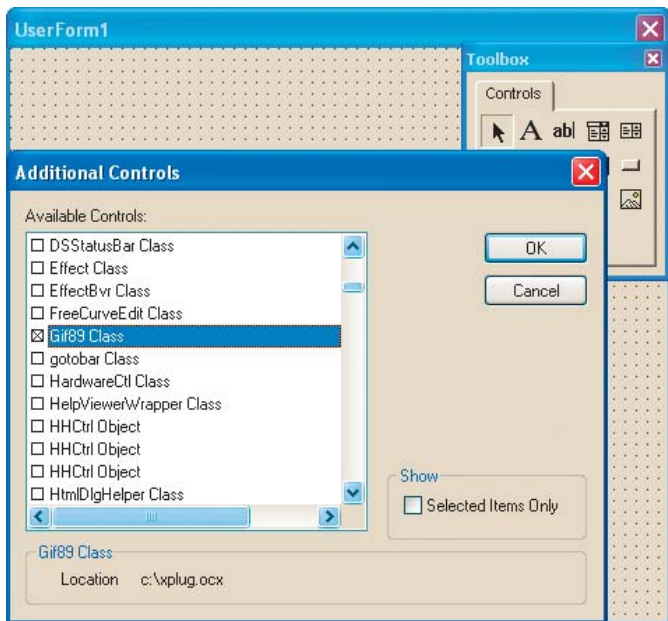


Рис. 2. Окно добавления объектов на панель Toolbox редактора VBA

Способов интеграции изображения с IP-видеокamеры может быть много. В данной статье рассмотрены три метода: создание экранной формы VBA, создание собственного ActiveX-объекта и создание собственного Web-интерфейса.

ИНТЕГРАЦИЯ С СОЗДАНИЕМ ЭКРАННОЙ ФОРМЫ VBA

Данный метод достаточно прост в реализации. Необходимо открыть редактор VBA компонента GraphWorX и создать объект класса UserForm (команда UserForm меню Insert). С помощью панели Toolbox нужно подключить библиотеку нашего ActiveX-объекта (команда Additional Controls меню Tools, рис. 2) и импортировать его с панели на VBA-форму. Для выполнения данного действия необходимо, чтобы подключаемая библиотека уже была зарегистрирована в операционной системе.

В текстовом коде объекта UserForm напишем следующее:

```
Private Sub UserForm_Activate()  
UserForm1.Gif89a1.RemoteHost = «192.168.2.1»  
UserForm1.Gif89a1.RemotePort = 8481  
UserForm1.Gif89a1.Timeout = 5  
UserForm1.Gif89a1.AuthType = 1  
UserForm1.Gif89a1.PreviewFrameRate = 1  
UserForm1.Gif89a1.PreviewWidth = 320  
UserForm1.Gif89a1.PreviewHeight = 240  
UserForm1.Gif89a1.Width = 320  
UserForm1.Gif89a1.Height = 240  
UserForm1.Gif89a1.Play  
End Sub  
Private Sub UserForm_Terminate()  
UserForm1.Gif89a1.Stop  
End Sub
```

Процедура UserForm_Activate выполняется при иницировании созданной VBA-формы. В ней задаётся IP-адрес видеокamеры, устанавливаются размеры изображения и выполняется метод Play, начинающий трансляцию видеоизображения с камеры. Процедура UserForm_Terminate выполняется при закрытии формы и инициирует метод Stop, прекращающий трансляцию. В данном примере процедуру UserForm_Terminate можно опустить, поскольку при закрытии VBA-формы опрос прекращается автоматически, так как закрывается и сам ActiveX-объект.



Рис. 3. Результат вывода изображения с IP-видеокamеры в экранную форму GraphWorX

В VBA-коде объекта ThisDisplay (экранная форма GraphWorX) необходимо написать следующий код:

```
Private Sub GwxDisplay_PostRuntimeStart()
UserForm1.Show
End Sub
```

Данная процедура выполняется автоматически после перехода экранной формы в режим исполнения и иницирует объект UserForm1 (созданную нами VBA-форму, внешний вид которой представлен на рис. 3).

Запустить выполнение этой процедуры можно также по нажатию кнопки или по другому событию объекта ThisDisplay.

Описанный метод имеет ряд недостатков, главными из которых является использование нового окна Windows и невозможность выполнения VBA-инструкций при публикации проекта в Web (для этой задачи у ICONICS имеется специальный программный продукт WebHMI).

СОЗДАНИЕ СОБСТВЕННОГО АКТИВХ-ОБЪЕКТА

Довольно распространённым методом решения данной задачи является создание собственного объекта ActiveX-Control, который будет использоваться в качестве контейнера для ActiveX-объекта IP-видеокамеры.

Такой объект можно создать в различных средах программирования. В данном примере мы будем использовать Visual Basic 6.0 из состава пакета Visual Studio 6.0.

При создании нового проекта необходимо выбрать тип ActiveX Control (рис. 4). Далее необходимо импортировать в форму объекта ActiveX-компонент IP-видеокамеры. Для этого нужно с помощью команды Components меню Project подключить библиотеку данного объекта аналогично тому, как мы делали это в предыдущем пункте, а затем появившийся на панели ToolBox объект импортировать в форму создаваемого ActiveX-Control. Для выполнения описанного действия подключаемая библиотека должна быть предварительно зарегистрирована в системе.

В окне текстового кода создаваемого объекта пишем:

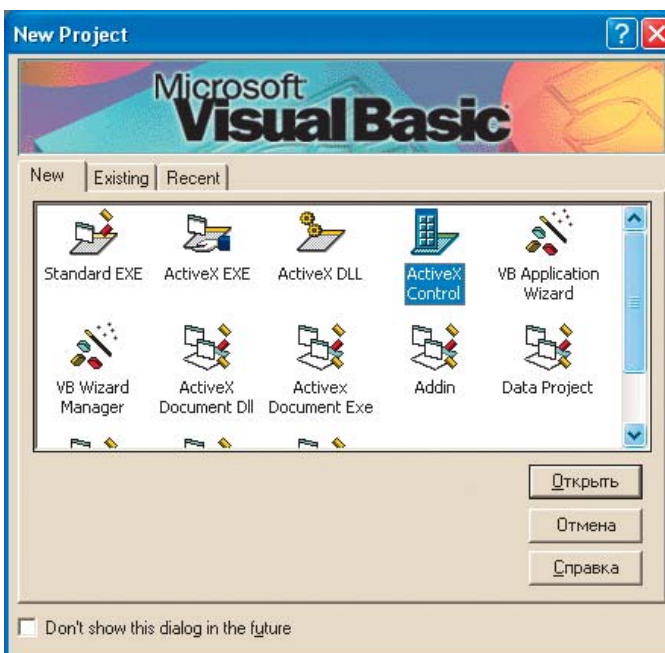


Рис. 4. Окно выбора типа создаваемого проекта в MS Visual Basic 6.0

```
Public IP As String
Public Sub Play()
UserControl.Gif89a1.RemoteHost = IP
UserControl.Gif89a1.Play
End Sub
Public Sub Stopplay()
UserControl.Gif89a1.Stop
End Sub
Private Sub UserControl_Initialize()
UserControl.Gif89a1.RemotePort = 8481
UserControl.Gif89a1.Timeout = 5
UserControl.Gif89a1.AuthType = 1
UserControl.Gif89a1.PreviewFrameRate = 1
UserControl.Gif89a1.PreviewWidth = 320
UserControl.Gif89a1.PreviewHeight = 240
UserControl.Gif89a1.Width = 320
UserControl.Gif89a1.Height = 240
End Sub
```

Строки Public IP As String, Public Sub Play() и Public Sub Stopplay() объявляют свойство IP и методы начала и окончания трансляции изображения соответственно.

При инициализации созданного нами объекта (процедура UserControl_Initialize()) задаются значения свойств ActiveX-Control видеокамеры (объект Gif89a1).

Теперь скомпилируем созданный нами объект (команда Build Outputs меню Tools/Publish) и зарегистрируем его в операционной системе с помощью команды regsvr32 (в моём примере команда имеет вид regsvr32 c:\IPCam.ocx).

Запустим компонент GraphWorX, создадим экранную форму и импортируем в неё скомпилированный ActiveX-объект, как это описано в разделе «Интеграция со SCADA-системами». В VBA-коде объекта ThisDisplay (экранная форма GraphWorX) необходимо написать следующее:

```
Private Sub GwxDisplay_PostRuntimeStart()
ThisDisplay.Cam1.IP = «192.168.2.1»
ThisDisplay.Cam1.Play
End Sub
Private Sub GwxDisplay_PreRuntimeStop()
ThisDisplay.Cam1.Stopplay
End Sub
```

Процедура GwxDisplay_PostRuntimeStart() выполняется автоматически после перехода экранной формы в режим исполнения. Она присваивает свойству нашего ActiveX-объекта IP-адрес видеокамеры и иницирует выполнение метода Play. Данный метод присваивает значение переменной IP соответствующему свойству исходного объекта и иницирует его метод Play, включающий трансляцию видеоизображения.

Процедура GwxDisplay_PreRuntimeStop() выполняется автоматически перед переходом экранной формы в режим разработки. Она иницирует выполнение метода Stopplay созданного нами объекта. Данный метод выполняет процедуру Stop ActiveX-объекта видеокамеры, прекращающую трансляцию изображения.

Запустить выполнение этих процедур можно также по нажатию кнопки или по другому событию объекта ThisDisplay.

Результат данного примера представлен на рис. 3.

Описанный метод обладает высокой универсальностью и может быть использован как для преодоления проблем несовместимости ActiveX-объекта и рабочей среды, куда необходимо его интегрировать, так и для разработки собственных библиотек объектов ActiveX-Control.

ИНТЕГРАЦИЯ ПОСРЕДСТВОМ СОЗДАНИЯ СОБСТВЕННОГО WEB-ИНТЕРФЕЙСА

Рассмотрим другой способ вывода изображения с IP-видеокамеры в экранную форму, который также может быть использован для решения других подобных задач интеграции. Суть метода заключается в разработке собственного Web-интерфейса и просмотра созданной Web-страницы в экранной форме SCADA-системы с помощью стандартного ActiveX-компонента Web Browser.

Как уже было упомянуто, IP-видеокамера имеет Web-интерфейс, но он представлен в виде нескольких Web-страниц, и изображение с видеокамеры занимает лишь часть окна (рис. 5), поэтому целесообразно разработать собственный интерфейс. Для этого создадим текстовый файл с расширением *.htm и поместим в него следующий html-код:

```
<html>
<SCRIPT LANGUAGE=«JavaScript»>
function Init() {
    cam.RemoteHost = «192.168.2.1»
    cam.RemotePort = 8481
    cam.Timeout = 5
    cam.AuthType = 1
    cam.PreviewFrameRate = 1
    cam.PreviewWidth = 320
    cam.PreviewHeight = 240
    cam.Play() }
</SCRIPT>
<body bgcolor=#e4ebf1 onload=«Init()»>
    <OBJECT ID=«cam» WIDTH=320 HEIGHT=240
    CLASSID=«CLSID:31150A86-0BBA-409f-BEB4-F3922D10BF34»
    </OBJECT>
</body>
</html>
```

Принцип работы приведённого кода следующий. После загрузки созданной Web-страницы выполняется функция Init, которая задаёт значения свойств ActiveX-объекта камеры и выполняет метод Play. Определение объекта осуществляется через идентификатор класса CLSID:31150A86-0BBA-409f-BEB4-F3922D10BF34 (в моём случае), с которым он зарегистрирован в реестре операционной системы (идентификатор можно определить из редактора системного реестра, который запускается командой regedit). Библиотека ActiveX-компонента должна быть предварительно зарегистрирована в системе.

Сохраним текстовый файл под именем IPCam.htm. Итак, мы создали Web-страницу с размерами заданного нами ActiveX-объекта, после загрузки которой автоматически начинается трансляция изображения с IP-видеокамеры. Теперь создадим экранную форму SCADA-системы и отобразим на ней созданный Web-интерфейс.

Для этого запустим компонент GraphWorX, создадим экранную форму и импортируем на неё ActiveX-объект



Рис. 5. Встроенный Web-интерфейс IP-видеокамеры

класса Web Browser. Процедура импорта объектов на экранную форму описана в разделе Интеграция со SCADA-системами.

В VBA-коде объекта ThisDisplay напишем следующее:

```
Private Sub GwxDisplay_PostRuntimeStart()
ThisDisplay.WebBrowser1.Navigate («c:\IPCam.htm»)
End Sub
```

Процедура GwxDisplay_PostRuntimeStart выполняется сразу после перехода экранной формы в режим исполнения и загружает в ActiveX-объект Web-браузера созданную нами страницу Web-интерфейса IP-видеокамеры. Инициировать выполнение этой процедуры можно также по нажатию кнопки или по другому событию объекта ThisDisplay.

Видеоизображение, полученное описанным методом, представлено на рис. 6.

РЕЗУЛЬТАТЫ

В статье были рассмотрены некоторые методы интеграции изображения с IP-видеокамеры в экранную форму SCADA-системы на примере GENESIS32. Реализация приведённых алгоритмов в других SCADA-пакетах имеет незначительные различия. Необходимое требование — поддержка ActiveX-компонентов и языков программирования инструкций (скриптов). Описанные трудности подключения видеокамер с помощью ActiveX-объектов характерны и для иного оборудования. Предложенные методы позволяют решить данные проблемы, а также дают общее представление о различных вариантах работы SCADA-систем с объектами типа ActiveX-Control. В своей статье я попытался систематизировать методы работы с ActiveX-объектами на примере задачи вывода изображения с видеокамеры на экранную форму SCADA-системы. Описанные примеры детализированы, благодаря чему могут быть использованы как руководство при разработке собственных проектов. Действия, связанные с конфигурированием внешнего вида экранных форм, были намеренно опущены, поскольку предполагается, что любой инженер, даже незнакомец со средствами программирования, сможет построить его с помощью интерфейса окна свойств.

Рассмотренные функции интеграции изображения с видеокамеры в SCADA-систему весьма актуальны и реализуют возможность создания систем диспетчеризации качественно нового уровня. ●

Автор — сотрудник фирмы ПРОСОФТ

Телефон: (495) 234-0636

E-mail: bodyagin@prosoft.ru