

Татьяна Кузьмина

## Подключение модулей ввода/вывода Advantech к SCADA-системе ICONICS GENESIS32 с помощью ActiveX-компонентов

Существует несколько методов подключения оборудования к SCADA-системам. Самыми распространенными из них являются использование специальных драйверов, разработанных для конкретной SCADA-системы, а также применение универсальных OPC-серверов. К сожалению, разработчики программного обеспечения SCADA предоставляют драйверы только для небольшого ряда продуктов самых известных фирм, а использование OPC не всегда доступно и при этом является недешёвым решением для небольших систем.

Компромиссом в данной ситуации является возможность использования библиотек элементов управления ActiveX. Производитель оборудования предоставляет для своих устройств библиотеку элементов управления, которая может использоваться в любых SCADA-системах, поддерживающих технологию ActiveX. Работа с элементами управления при этом осуществляется с помощью стандартных языков программирования или упрощённых скриптоподобных функций.

### Технология ActiveX

Изначально технология ActiveX разрабатывалась для использования в области мультимедиа-приложений в сети Интернет, но впоследствии благодаря ряду преимуществ она нашла своё применение в SCADA-системах.

Элементы управления ActiveX представляют собой универсальные готовые компоненты, которые интегрируются в приложения и позволяют выполнять различные предопределённые действия. Технология ActiveX возникла в процессе модификации элементов управления OLE (управляющих элементов OLE), которые в свою очередь базируются на технологиях COM и DCOM. Это позволяет элементам взаимодействовать как локально, так и в сетевых приложениях. Важной особенностью элементов управления ActiveX является возможность работать совместно друг с другом и другими приложениями независимо от языка программирования, с помощью которого они были созданы.

Интерфейс элементов управления ActiveX представлен свойствами (properties), методами (methods) и событиями (events). Использование свойств позволяет отображать или изменять характеристики объекта ActiveX. Метод — это действие или функция, которые способен выполнять элемент, а событие — это внешнее условие, на которое может реагировать элемент ActiveX и при этом выполнять какое-либо предопределённое действие.

В статье рассматриваются основные принципы работы с элементами управления ActiveX в SCADA-системах на примере подключения модуля аналогового ввода Advantech к SCADA-системе ICONICS GENESIS32.

### Установка библиотеки элементов ActiveX для модулей ввода/вывода Advantech

Для плат и модулей ввода/вывода серий PCI-17xx, MIC-37xx и USB-47xx компания Advantech не предоставляет OPC-серверов, а SCADA-системы различных производителей не включают в себя драйверы для работы с ними.

Разработка проекта возможна благодаря библиотеке ActiveX-компонентов **ActiveDaq Pro**, которая доступна для свободного скачивания с сайта производителя ([www.advantech.com.tw/support](http://www.advantech.com.tw/support)). Перед началом работы необходимо установить данную библиотеку, а также драйвер для используемого модуля ввода-вывода, который прилагается в его комплекте поставки.

*Замечание.* При установке библиотеки **ActiveDaq Pro** необходимая информация о компонентах ActiveX будет автоматически внесена в регистр Windows. Регистрацию можно осуществить также с помощью команды **RegSvr32**. (Выберите меню **Start->Run**, в командной строке наберите **CMD**, после чего введите **REGSVR32 "C:\<путь к OCX control>\<Имя ActiveX Control>.OCX"**.)

Библиотека **ActiveDaq Pro** включает в себя такие компоненты, как **AI Control**, **AO Control** и **DIO Control** (для системы аналогового и дискретного ввода/вывода), **Thermo Control** (для модулей подключения термодатчиков), **Counter Control** (для плат счетчиков/таймеров), **Pulse Control** (для модулей с частотным/импульсным выходом).

### Рассматриваемый пример

Чтобы проиллюстрировать удобство и простоту использования готовых ActiveX-элементов, рассмотрим в качестве примера следующую задачу. Имеется многоканальный модуль аналогового ввода, необходимо создать проект в SCADA-системе, позволяющий осуществлять:

- выбор устройства из списка подключённых;
- выбор канала, с которого будет считываться информация;
- получение данных и их отображение в текстовом блоке и на динамическом графике.

В данном примере рассматривается система аналогового ввода, поэтому будет использоваться элемент ActiveX **ActiveDAQ Pro AI Control (AdvAI)**. Для решения поставлен-

ной задачи необходимы следующие атрибуты объекта AdvAI:

**Методы (Methods)**

**SelectDevice** Позволяет выбрать устройство из списка установленных в системе, которое поддерживает функции аналогового ввода

**Свойства (Properties)**

**DeviceNumber** Отображает или устанавливает номер выбранного устройства

**DeviceName** Отображает имя выбранного устройства

**ChannelNow** Отображает или устанавливает текущий канал аналогового ввода

**DataAnalog** Отображает значения сигнала на канале аналогового ввода с номером ChannelNow

*Замечание.* Полный список методов, свойств и событий для ActiveX-элементов доступен в документации ActiveDaq Pro.

**СОЗДАНИЕ ПРОЕКТА В SCADA-СИСТЕМЕ ADVANTECH STUDIO**

Перед тем как рассмотреть принципы внедрения элементов управления ActiveX в ICONICS GENESIS32, необходимо сказать пару слов об их использовании в программе Advantech Studio. Интеграция элементов управления в данную SCADA-систему происходит по несколько упрощённой схеме – с помощью специальных скриптоподобных функций, которые представлены в таблице 1.

*Замечание.* Приставка *str* указывает на то, что переменная должна быть строковой.

Таблица 1

Список функций Advantech Studio

Функция	Назначение
XGet(strName, strProperties)	Отображает значение свойства <strProperties> элемента ActiveX с именем <strName>.
XSet(strName, strProperties, Value)	Устанавливает значение <Value> в свойство <strProperties> ActiveX-объекта с именем <strName>.
XRun(strName, strMethod, Parameter1, Parameter2, ..., ParameterN)	Реализует метод <strMethod> ActiveX-элемента с именем <strName> в соответствии с заданными параметрами <Parameter1>, <Parameter2>, ..., <ParameterN>.

Элемент управления ActiveX добавляется в проект с помощью пункта меню **Insert-ActiveX object** или кнопки на **Панели Инструментов OX** ( ).

Для дальнейшей работы требуется введение любого имени этого элемента в поле **NAME** в окне свойств, доступном при двойном щелчке мышью на появившемся значке **Advantech ActiveDAQ Pro AI Control** (рис. 1).

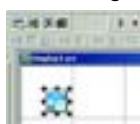


Рис. 1. Значок Advantech ActiveDAQ Pro AI Control

В данном окне свойств доступны также настройки ActiveX-объекта (тип устройства, настройка параметров каналов и т.д.), а также список свойств, методов и событий для этого элемента.

Функции, позволяющие работать с элементом, заносятся в лист вычислений **Math** (рис. 2) в поле **Expression**. Синтаксис имеет вид, например, **XGet("AdvAI-N1", "DeviceNumber")**, где **AdvAI-N1** — имя ActiveX-компонента, **DeviceNumber** — свойство, позволяющее отобразить номер выбранного уст-

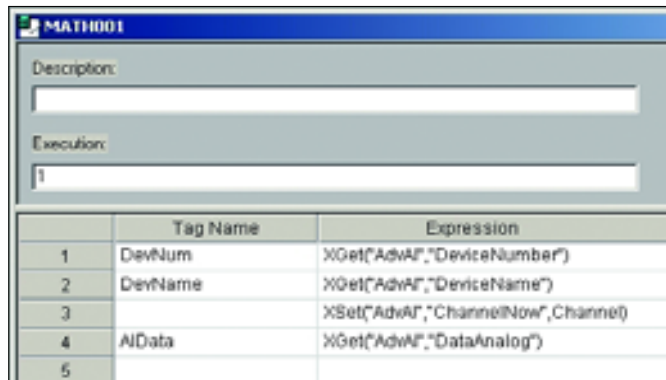


Рис. 2. Лист вычислений Math

ройства. Прописав в поле **Tag Name** переменную (tag), мы получаем возможность использовать значение функции в дальнейших вычислениях или при построении графиков.

**СОЗДАНИЕ ПРОЕКТА В SCADA-СИСТЕМЕ ICONICS GENESIS32**

Интеграция элементов управления ActiveX в SCADA-систему ICONICS GENESIS32 имеет более сложную схему, поэтому целесообразно рассмотреть данный процесс более подробно. Обращение к ActiveX осуществляется с помощью скриптов VBscript, Jscript или VBA. В данном руководстве рассмотрен пример создания кода VBA.

**Создание проекта**

Создадим новую экранную форму в GraphWorX32. Введите ActiveX-компонент Advantech ActiveDAQ Pro AI Control с помощью кнопки на **Панели инструментов** либо через меню **Правка — Вставить новый объект** (рис. 3).

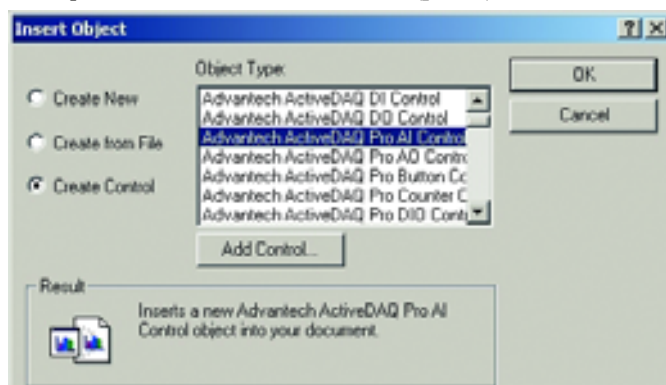


Рис. 3. Добавление элемента управления ActiveDAQ Pro AI Control

**Выбор устройства**

Настроим возможность выбора устройства из списка подключённых.

Добавьте на рабочий лист объект **Кнопка**. В диалоговом окне объекта **Инспектор свойств** (рис. 4) введите текст «Выбрать устройство», который будет отображаться на кнопке, а также выберите из выпадающего списка действие **Выполнить сценарий VBA**. Нажмите кнопку **Создать** для создания нового сценария VBA.

Введите имя сценария **SelectDevice** в пункте **Имя сценария**, после чего нажмите **ОК**. Программа создаст модуль **GwxSelectDevice\_Main** и откроет редактор VBA. Для того чтобы с помощью данного сценария работать с ActiveX-компонентом, необходимо создать новый экземпляр класса AdvAI, назовем его AdvAI1. Для выбора устройства исполь-

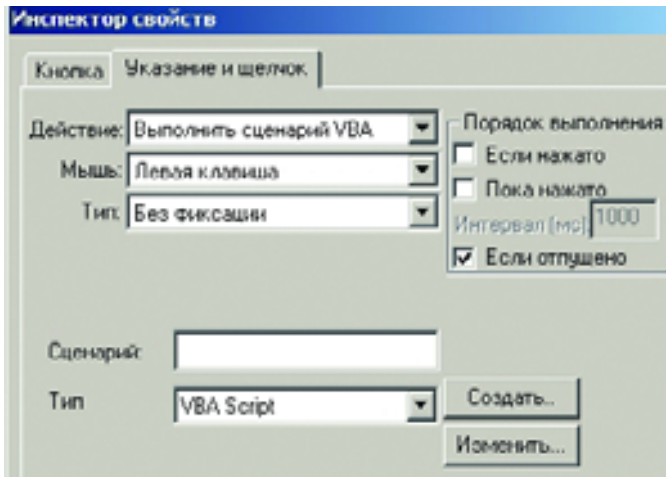


Рис. 4. Диалоговое окно свойств объекта Кнопка

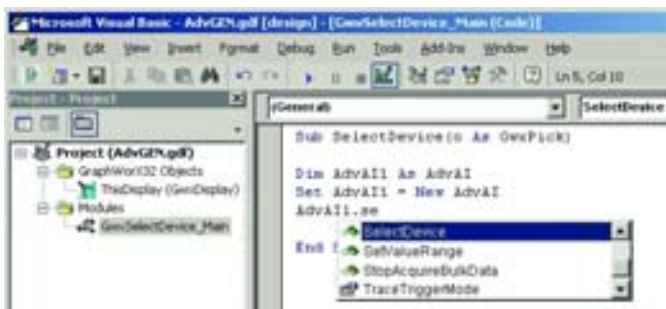


Рис. 5. Написание сценария в редакторе VBA

уем метод **SelectDevice** (рис. 5). Код VBA при этом будет иметь вид:

```
Sub SelectDevice(o As GwxBlick)
Dim AdvAI1 As AdvAI
Set AdvAI1 = New AdvAI
AdvAI1.SelectDevice
End Sub
```

Настроим текстовые блоки для вывода информации о номере и названии выбранного устройства. Для этого воспользуемся свойствами **DeviceNumber** и **DeviceName**.

Добавьте на рабочий лист объект **Значение параметра** (кнопка на Панели инструментов). В поле **Источник данных** диалогового окна **Инспектор свойств** укажите локальную переменную GraphWorX32 **~~DeviceNumber~~** (рис. 6). Выберите запрашиваемый и отображаемый типы данных **FLOAT** (действительное) и формат данных **xxx** (целое число).

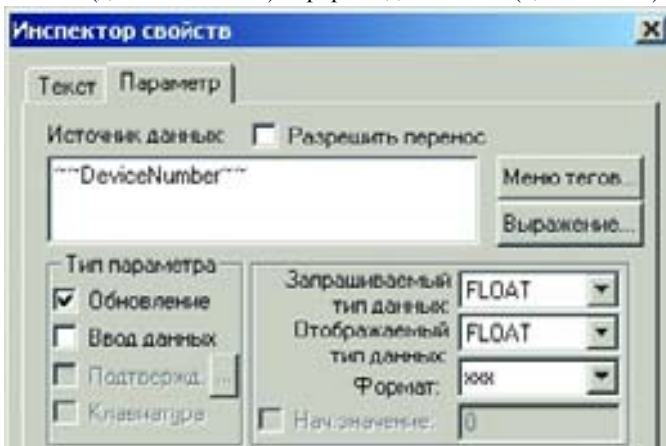


Рис. 6. Добавление источника данных для объекта Значение параметра

Перейдите к VBA-коду, выполняемому при нажатии кнопки **Выбрать устройство**.

*Замечание.* Для открытия редактора VBA можно использовать клавиши быстрого доступа **Alt+F11**.

Определяем новую переменную VBA, принадлежащую классу GwxPoint, устанавливаем взаимосвязь с локальной переменной GraphWorX **~~DeviceNumber~~** и используем свойство **DeviceNumber** для получения номера устройства и отображения его в локальной переменной.

Код VBA будет иметь вид:

```
Dim DeviceNumber As GwxPoint
Set DeviceNumber =
ThisDisplay.GetPointObjectName("~~DeviceNumber~~")
DeviceNumber.Value = AdvAI1.DeviceNumber
```

Для настройки текстового блока с названием устройства по аналогичной схеме конфигурируем объект **Значение параметра** (кнопка на Панели инструментов) с локальной переменной **~~DeviceName~~** типа String (строка) и устанавливаем связь с переменной VBA **DeviceName**. Код VBA будет иметь вид:

```
Dim DeviceName As GwxPoint
Set DeviceName =
ThisDisplay.GetPointObjectName("~~DeviceName~~")
DeviceName.Value = AdvAI1.DeviceName
```

Общий код сценария VBA, выполняемого при нажатии кнопки **Выбрать устройство**:

```
Sub SelectDevice(o As GwxBlick)
Dim AdvAI1 As AdvAI, DeviceName As GwxPoint, DeviceNumber As GwxPoint
Set AdvAI1 = New AdvAI
Set DeviceName =
ThisDisplay.GetPointObjectName("~~DeviceName~~")
Set DeviceNumber =
ThisDisplay.GetPointObjectName("~~DeviceNumber~~")
AdvAI1.SelectDevice
DeviceName.Value = AdvAI1.DeviceName
DeviceNumber.Value = AdvAI1.DeviceNumber
End Sub
```

Протестируем проект с помощью режима **Исполнение** (рис. 7).

### Выбор канала

Настроим возможность выбора рабочего канала, а также отображение в текстовом блоке значения сигнала на этом канале.

Откройте редактор VBA. В древовидном списке с левой стороны окна двойным нажатием на элемент **ThisDisplay** откройте окно редактора кода. В этом окне в выпадающем списке в левом углу выберите **GwxDisplay**, а в выпадающем

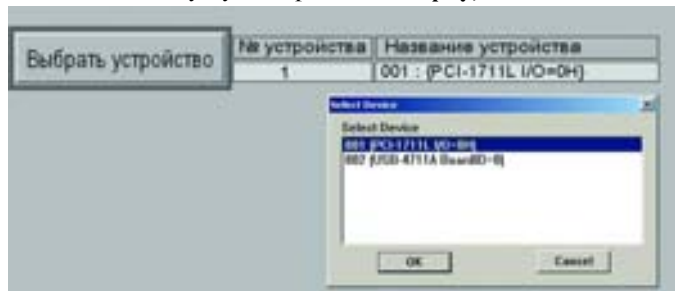



Рис. 7. Тестирование проекта в режиме Исполнение

списке действий в правом углу выберите **DataEntryValueEntered**:

```
Private Sub GwxDisplay_DataEntryValueEntered(ByVal dataEntry As Object)
```

Для корректной работы программы необходимо задать номер устройства, которое выбирается с помощью кнопки **Выбрать устройство**. Используем для этого ранее созданную локальную переменную GraphWorX32 **DeviceNumber**. Определим новую переменную VBA **DevNum**, привяжем ее к локальной переменной **DeviceNumber**:

```
Dim DevNum As GwxPoint
Set DevNum = _
ThisDisplay.GetPointObjectFromName(«DeviceNumber»)
AdvAI1.DeviceNumber = DevNum.Value
```

Выберем канал этого устройства, для чего используем свойство объекта ActiveX **ChannelNow**. Создадим блок **Значение параметра** (кнопка  на **Панели инструментов**). В диалоговом окне **Инспектор свойств** установите в поле **Источник данных** локальную переменную **InsertChannel** тип **Float** (действительное), разрешите ввод данных с помощью опции **Ввод данных**.

Создадим новую переменную VBA **InsertChannel**, определим её как экземпляр класса **GwxPoint**, а также свяжем её с локальной переменной **InsertChannel**:

```
Dim InsertChannel As GwxPoint
Set InsertChannel = _
ThisDisplay.GetPointObjectFromName(«InsertChannel»)
AdvAI1.ChannelNow = InsertChannel.Value
```


Общий код сценария VBA будет выглядеть следующим образом:


```
Private Sub GwxDisplay_DataEntryValueEntered(ByVal dataEntry As Object)

Dim DevNum As GwxPoint, InsertChannel As GwxPoint
Set DevNum = _
ThisDisplay.GetPointObjectFromName(«DeviceNumber»)
Set InsertChannel = _
ThisDisplay.GetPointObjectFromName(«InsertChannel»)

AdvAI1.DeviceNumber = DevNum.Value
AdvAI1.ChannelNow = InsertChannel.Value

End Sub
```

Выведем данные из выбранного канала в текстовый блок с помощью свойства объекта ActiveX **DataAnalog**. Так как данные должны периодически обновляться, необходимо использовать GENESIS32 ActiveX-компонент **Таймер** (Iconics AxTimer Control Properties). Добавим данный компонент через пункт меню **Правка-Вставить новый объект** или с помощью кнопки  на **Панели инструментов**. Щёлкнем два раза на таймере, установив значение интервала времени 100 мс.

Добавим на рабочий лист блок **Значение параметра** (кнопка  на **Панели инструментов**), в котором будут показываться текущие данные из канала аналогового ввода. Определим

локальную переменную **AnalogData** (тип String). Нам потребуется свойство объекта ActiveX **DataAnalog** для получения данных из текущего канала. В редакторе VBA создадим процедуру **Private Sub AxTimer1\_Timer()** (аналогично созданной ранее процедуре **GwxDisplay\_DataEntryValueEntered**).

Определим новую переменную VBA **AnalogData** и свяжем её с локальной переменной **AnalogData**:


```
Private Sub AxTimer1_Timer()

Dim AnalogData As GwxPoint
Set AnalogData = _
ThisDisplay.GetPointObjectFromName(«AnalogData»)
AnalogData.Value = AdvAI1.DataAnalog

End Sub
```

## Построение графика

Встроенный в ICONICS GENESIS32 компонент **Тренд** (график) позволяет отображать данные от OPC-тега. Поэтому необходимо привязать полученные данные аналогового ввода к OPC. Для этого создадим новый OPC-тег с помощью программы **OPC Simulator** типа **Float** и без определения функции, назовём его, например, **tag1**. Создадим Кнопку **Построение графика**, в диалоговом окне **Инспектор Свойств** укажем действие **Передать значение**, поставим признак **Пока нажато** и интервал 100 мс. Установим опцию **Начальное состояние — нажато**. В поле **Источник данных** возьмём **tag1** из меню тегов, а в поле **Пока нажато** — локальную переменную **AnalogData** (рис. 8).

Добавим **Тренд** с помощью кнопки  на **Панели инструментов**. Установим необходимые диапазоны осей координат графика и привяжем к нему созданный OPC-тег **tag1** через закладку **Перья->теги OPC**.

Протестируем проект с помощью режима **Исполнение** (рис. 9).

## ЗАКЛЮЧЕНИЕ

Использование элементов управления ActiveX значительно облегчает процесс разработки проекта. Благодаря универсальности и завершённости компонент легко интегрируется как готовый блок, позволяющий получать данные, изменять настройки и удалённо производить мониторинг состояния устройства. Управлять параметрами данного блока можно с

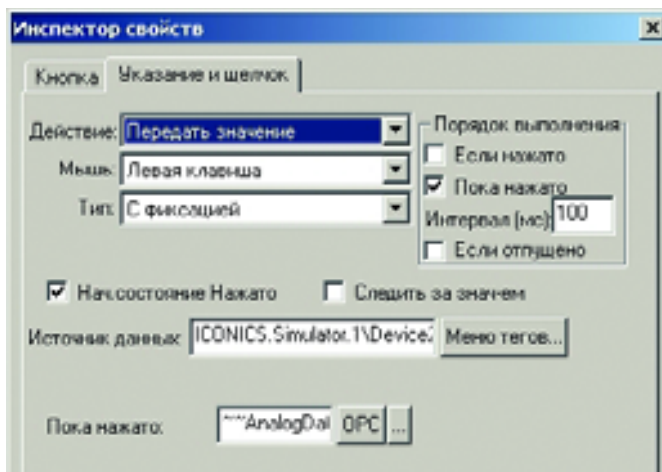


Рис. 8. Добавление локальной переменной в диалоговом окне Инспектора Свойств объекта Кнопка



Рис. 9. Тестирование готового проекта в режиме Исполнение

помощью любого доступного языка программирования: Visual Basic, Visual C++, Delphi, VB.Net и т.д. Это не только сокращает время разработки проекта, но и позволяет исключить вероятность возникновения ошибок при написании сложного кода.

Принципы интеграции элементов управления являются универсальными, поэтому описанные в данной статье методы и советы могут быть использованы при работе с любой SCADA-системой, поддерживающей технологию ActiveX. ●

**Автор — сотрудник фирмы ПРОСОФТ**

**Телефон: (495) 234-0636**

**E-mail: info@prosoft.ru**

НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ

**Новости ISA**

18-19 апреля 2008 года в Мадриде (Испания) была проведена конференция руководителей 12-го округа ISA (<http://isaeur.org>), объединяющего в своих рядах специалистов из Европейского, Ближневосточного и Африканского регионов. Во второй день работы конференции состоялось ежегодное заседание Исполкома округа 12. В работе конференции приняли участие представители 9 стран: Великобритании, Испании, Италии, Нидерландов, Франции, Ирландии, США, Российской Федерации, Португалии. Делегация Российской секции ISA, состоявшая из шести профессоров (Оводенко А.А. — глава делегации, Боер В.М., Шепета А.П., Павлов И.А., Лосев К.В., Бобович А.В.), приняла участие в работе конференции. Программа конференции была очень насыщенной: руководителями ISA и округа 12 было сделано 15 презентаций. В работе конференции приняли участие президент ISA госпожа Kim Miller Dunn (США) и её предшественник на посту президента ISA господин Gerald Wilbanks (США).

Большая делегация представляет округ 12 на ежегодном летнем собрании ISA в городе West Palm Beach (США) в конце мая: Александр Бобович (Россия), Pino Zani (Италия), Billy Walsh, Kevin Dignam, Declan Lordan (все — Ирландия).

Очередная ежегодная конференция округа 12 состоится в Нидерландах в апреле-мае 2009 года. Окончательное решение о дате и месте проведения конференции будет принято в октябре в Хьюстоне (США) во время работы ежегодного собрания ISA.

19 апреля в Мадриде были объявлены победители IV Европейского конкурса ISA на лучшую студенческую научную работу (ESPC-2008). Победителями стали представители 5 стран: Италии, Испании, США, Кении и Российской Федерации. Огромного успеха в очередной раз добились студенты и аспиранты Санкт-Петербургского государственного университета аэрокосмического приборостроения (ГУАП): золо-



**На конференции руководителей 12-го округа ISA в Мадриде**



**Ректор ГУАП А.А. Оводенко**

тых медалей удостоены аспирант Евгений Бакин и студент Георгий Кулумчев (недавно получивший приз за лучший студенческий проект в области промышленной автоматизации на выставке «ПТА Санкт-Петербург 2008»); серебряными медалями награждены выпускница ГУАП 2008 года Ирина Кипяткова и студент Симон Одундо; бронзовыми медалями награждены студенты Алексей Тыртычный, Иван Спиндзак и Константин Гурнов.

Президентом Российской секции ISA на 2009 год впервые избрана женщина — член-корреспондент РАН, доктор технических наук, профессор, декан факультета интеллектуальных систем управления и нанотехнологий ГУАП Чубраева Лидия Игоревна.

29 апреля 2008 года в Москве в Екатерининском зале Кремля Президент Российской Федерации Владимир Владимирович Путин вручил государственные награды 54 выдающимся деятелям науки, культуры, образования, представителям промышленности и сельского хозяйства, военнослужащим. Орденом «За заслуги перед Отечеством» IV степени награждён ректор ГУАП, глава представительства ISA в Российской Федерации Оводенко Анатолий Аркадьевич. Выступая на церемонии награждения, В.В. Путин сказал: «Высокие награды Родины вручаются сегодня и за мирные свершения в науке, медицине и образовании, за трудовые успехи на производстве. Из таких достижений и складывается экономическая и технологическая мощь страны, её весомый интеллектуальный багаж, благополучие граждан.

Биография каждого из вас достойна стать примером, и в первую очередь для молодёжи, но думаю, что самым убедительным для неё является, прежде всего, авторитет учителя, наставника, педагога». ●