

# ИССЛЕДОВАНИЕ ПРЕИМУЩЕСТВ ИНСТРУМЕНТАЛЬНЫХ ДРАЙВЕРОВ PXI

## EXPLORING THE ADVANTAGES OF PXI INSTRUMENT DRIVERS

Стивен Питерсон (Stephen Peterson), Agilent Technologies

Теоретически модульная система, предлагающая « типовые блоки » в виде модулей и соответствующие драйверы, позволяет создать практически любой прибор. Тем не менее, при практическом создании высококачественного прибора можно столкнуться с серьезными трудностями в связи с необходимостью учитывать всевозможные детали — от ожидания установления сигнала до расчета поправочных коэффициентов для всех модулей.

Инструментальный драйвер должен обладать встроенным « интеллектом », необходимым для управления модулями, входящими в состав многомодульного прибора. Настоящая статья описывает преимущества таких инструментальных драйверов на примере PXI. В общем это преимущество можно описать одним словом: простота. Вы увидите, что простота инструментального драйвера помогает сфокусироваться на функциональности самого прибора, например, на функциях создания сигнала или выполнения измерений, а не на тонкостях управления каждым отдельным модулем.

### ОПРЕДЕЛЕНИЕ «ИНСТРУМЕНТАЛЬНОГО ДРАЙВЕРА»

В качестве рабочего определения можно предложить следующее — инструментальный драйвер представляет собой унифицированный набор программных утилит, который управляет несколькими модулями как единым прибором. В общем, драйвер модульного прибора подобен драйверу традиционного автономного прибора, например, генератора сигналов. И в модульном, и в автономном приборе драйвер позволяет пользователю сосредоточиться на решении проблемы — например, на измерении параметров тестируемого устройства, — а не на настройке нескольких модулей в попытке заставить их работать, как единый прибор.

Инструментальные драйверы позволяют воспользоваться преимуществами функциональности на уровне прибора и гибкостью модульной системы, что включает создание новых приборов путем до-

```
IAGM9300 driverM9300 = newAgM9300();
driverM9300.Initialize("PXI24::0::0::INSTR", false, false,
"simulate=false");
IAGM9301 driverM9301 = newAgM9301();
driverM9301.Initialize("PXI25::0::0::INSTR", false, false,
"simulate=false");
```

// повторите приведенные выше операции для каждого дополнительного модуля

Листинг 1. Фрагмент кода на языке C#, демонстрирующий подключение двух отдельных модулей

```
IAGM938x driver = newAgM938x();
driver.Initialize("PXI24::0::0::INSTR;PXI25::0::0::INSTR;PXI26::0::0::INS
TR;PXI27::0::0::INSTR", false, false, "simulate=false");
```

Листинг 2. Этот листинг выполняет подключение всех модулей в приборе PXI

```
IAGM938x driver = newAgM938x();
driver.Initialize("M9381A VSG", false, false, "simulate=false");
```

Листинг 3. Определение логического прибора дополнительно упрощает его подключение



## Agilent Technologies

бавления или удаления модулей. Это может исключить потребность покупки нескольких однотипных устройств.

### ПРЕИМУЩЕСТВА ПРОСТОТЫ

По сравнению с установкой, подключением и использованием набора модулей, работа на уровне прибора выглядит значительно проще. Впрочем, подобно драйверу отдельного модуля, драйвер прибора позволяет воспользоваться гибкостью модульной системы за счет поддержки различных конфигураций модулей.

Проиллюстрировать эти концепции нам помогут два многомодульных прибора формата PXI: источник немодулированного сигнала (CW) и векторный генератор сигналов (VSG). Оба прибора включают инструментальный драйвер IVI, а также драйвер IVI для модуля опорного сигнала внутри источника немодулированного сигнала или векторного генератора сигналов.

### ПРОСТАЯ УСТАНОВКА

Если драйверы отдельных модулей не поставляются в виде единого пакета, вам придется скачать, установить и настроить по одному драйверу на каждый модуль. И хотя обычно это не создает серьезных проблем, такая процедура может оказаться достаточно утомительной. Этот процесс определенно более сложен и трудоемок, чем установка инструментального драйвера: после однократной загрузки и однократной установки драйвер готов к работе.

### ПРОСТОЕ ПОДКЛЮЧЕНИЕ ПРИБОРА

Подключение инструментального драйвера выполняется почти так же просто, как и подключение драйвера модуля. Например, подключение драйвера модуля на языке C# может выглядеть примерно так, как в листинге 1. В отличие от этого, подключение инструментального драйвера выполняется проще, как показано в листинге 2.

Эти листинги помогают проиллюстрировать рекуррентный характер подключения модулей по сравнению с подключением одного прибора: одна инициализация инструментального драйвера против нескольких инициализаций драйверов отдельных модулей.

При подключении инструментального драйвера есть один важный момент, о котором не следует забывать: нужно указать все модули, входящие в состав прибора. Для многократного применения эту процедуру можно упростить, определив логический прибор в хранилище конфигураций IVI и использовав имя логического прибора вместо адресов модулей, как показано в листинге 3.

Каждый из указанных здесь приборов предлагает программную переднюю панель (SFP), которая, помимо прочего, позволяет сохранять «подключение прибора» в хранилище конфигураций IVI, связывая его с Логическим именем IVI. Пример такого экрана приведен на рис. 1.

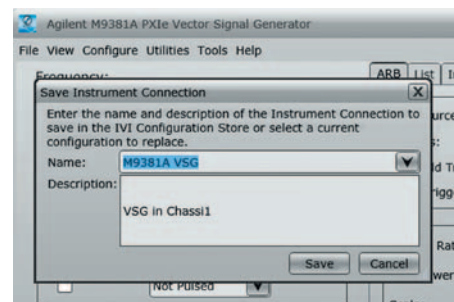


Рис. 1. Программная передняя панель векторного генератора сигналов формата PXI позволяет сохранить конфигурацию прибора

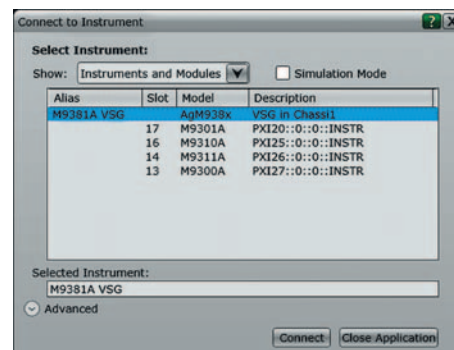


Рис. 2. Диалоговое окно подключения показывает отдельные модули, входящие в логическое имя

Если определение прибора находится в хранилище конфигураций IVI, диалоговое окно подключения программной передней панели предлагает в качестве опции подключения логическое имя, как показано на рис. 2. Кроме того, логическое имя можно использовать через вызов инициализации IVI, подобно тому, как показано в листинге 3.

## ПРОСТАЯ РАБОТА С ПРИБОРОМ

В конечном итоге именно удобство применения инструментального драйвера является основным атрибутом, отличающим его от драйверов модулей. И хотя драйверы модулей, скорее всего, будут содержать приложение для управления отдельными модулями, ответственность за объединение функций нескольких модулей по-прежнему лежит на пользователе. В отличие от этого, инструментальный драйвер позволяет сразу же использовать прибор, что зачастую дополняется специальным приложением, управляющим всеми модулями прибора. В результате инструментальный драйвер, как правило, сокращает «время до первого измерения».

На рис. 3 показан пример программной передней панели для векторного генератора сигналов в формате PXI и соответствующего инструментального драйвера IVI. Как видно на этом снимке экрана, пользователь может сконцентрироваться на решаемой задаче — на настройке частоты, мощности, модуляции и т.д., — а не на взаимодействии модулей.

## УСТАНОВКА ЧАСТОТЫ И УРОВНЯ МОЩНОСТИ

Другим аспектом удобства применения является программирование модулей, которое выполняется проще при использовании инструментального драйвера. В качестве наглядных примеров давайте рассмотрим установку частоты и уровня мощности векторного генератора сигналов. При использовании инструментального драйвера настройка частоты выполняется всего двумя операторами (листинг 4).

Для оптимизации производительности приведенный в качестве примера драйвер IVI использует принцип «установки и применения», потому что установка разных параметров может потребовать вычисления поправочных коэффициентов и изменения настроек оборудования. Эти операции можно отложить до момента вызова «Apply()», что позволяет выполнить взаимозависимые вычисления и изменения за один шаг.

Установка уровня мощности немного сложнее установки частоты в связи с некоторыми ограничениями, накладываемыми на отдельные модули векторного генератора сигналов:

- выходной модуль источника имеет достаточно грубый шаг перестройки амплитуды (например, 0,25 дБ);
- если включен модуль автоматической регулировки уровня (APU), шаг пере-

```
driver.RF.Frequency = 10e6; // 10 МГц
driver.Apply();
```

Листинг 4. Для настройки выходной частоты 10 МГц достаточно двух строк кода

```
driver.RF.Level = -10; // -10 дБм
driver.ALC.Enabled=true;
driver.Apply();
```

Листинг 5. Для настройки уровня выходной мощности на -10 дБм и включения APU достаточно всего трех строк кода

```
driver.RF.Frequency = 1.01e9; // 1.01 ГГц
driver.RF.Level = -10; // -10 дБм
driver.RF.OutputEnabled = true;
driver.Apply();
```

Листинг 6. Один и тот же код можно использовать для настройки немодулированного сигнала в любой аппаратной конфигурации



Рис. 3. Программная передняя панель упрощает и ускоряет настройку ключевых параметров векторного генератора сигналов

стройки амплитуды становится существенно меньше (например 0,02 дБ);

- если включена модуляция, модуль модулятора тоже имеет очень малый шаг перестройки амплитуды (например 0,02 дБ).

Кроме того, реальные параметры внутреннего аттенюатора и усилителя могут отличаться от каскада к каскаду и от модуля к модулю. В результате значения этих параметров надо считывать из модуля: реальные значения определяются во время заводской калибровки и сохраняются в энергонезависимой памяти модуля.

Поскольку инструментальный драйвер автоматически все это учитывает, настройку уровня мощности можно выполнить всего тремя операторами, как показано в листинге 5.

## БОЛЬШЕЕ УДОБСТВО ПРИМЕНЕНИЯ

Процедуры настройки частоты и мощности относятся к достаточно простым случаям настройки. Другие более сложные варианты настройки могут быть следующими:

- в векторном анализаторе сигналов (VSA) частотные характеристики всех модулей должны объединяться и корректировать измеряемые значения, возвращаемые каждым модулем дигитайзера;
- почти во всех приборах приходится вычислять время, необходимое сигнальному тракту для достижения стабильного состояния после изменения аппаратных настроек (т.е., время установления).

Кроме того, конфигурация зависит от опциональных возможностей модулей прибора. Например, упомянутые выше источник немодулированного сигнала и векторный генератор сигналов могут комплектоваться синтезатором на 3 или 6 ГГц.

## РАСШИРЕННАЯ НАСТРОЙКА СИСТЕМЫ

Кроме того, инструментальные драйверы позволяют в полной мере воспользоваться основным достоинством модульных систем, а именно возможностью приобретения аппаратных модулей только с необходимой функциональностью. Например, использованный нами в качестве примера источник немодулированного сигнала включает три модуля: модуль опорного сигнала, синтезатор и ВЧ выход. Добавление модулятора превращает его в векторный генератор сигналов, а инструментальный драйвер берет на себя заботу о соответствующей настройке модулей и обеспечении их взаимодействия.

Это иллюстрируется листингом 6: одни и те же команды устанавливают уровень мощности немодулированного сигнала частотой 1,01 ГГц на -10 дБм для источника немодулированного сигнала и для векторного генератора сигналов. Для выполнения этой же операции с помощью драйверов модулей пользователю пришлось бы сначала определить, установлен ли модулятор, а затем соответствующим образом настроить сигнальный тракт и поправочные коэффициенты.

## КОГДА НЕ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ИНСТРУМЕНТАЛЬНЫЙ ДРАЙВЕР

А теперь одно предостережение: типичный инструментальный драйвер не дает того уровня гибкости, которого можно достичь за счет применения драйверов отдельных модулей. Например, если пользователь вставит между синтезатором и модулятором векторного генератора сигналов преобразователь частоты, инструментальный драйвер может не суметь рассчитать входную частоту модулятора.

В некоторых случаях инструментальные драйверы допускают определенный уровень адаптации. Например, использованный здесь драйвер IVI предлагает перехват управления на уровне модулей, а также таблицу потерь в тестовой оснастке, позволяющую указать характеристики частоты и мощности, которые будут учтены при расчете параметров настройки. Впрочем, даже такие функции не могут охватить все возможные аппаратные конфигурации.

## ЗАКЛЮЧЕНИЕ

Во многих приложениях инструментальный драйвер упрощает большинство операций, которые сложно выполнить с помощью драйверов отдельных модулей: установка, подключение и применение. Для пользователя это сокращает время, уходящее на создание прибора, и позволяет уделять больше времени повседневной работе. ☑

*This article focuses on the advantages of «smart» instrument drivers using PXI as an example.*