

LabVIEW В USB-ЛАБОРАТОРИИ

Афонский А.А., Суханов Е.В.

Два года назад мы уже писали об использовании комплекта разработчика программного обеспечения (SDK) для цифрового запоминающего осциллографа АКТАКОМ АСК-3106 (см. КИПиС, № 6, 2003 г.). В той статье мы пришли к выводу, что комплект разработчика (набор системных драйверов, библиотек, документации и примеров программ) — хорошая альтернатива фирменному ПО в случаях, когда требуется повышенная эффективность или нестандартная обработка измерений. Однако при этом пользователь должен иметь квалификацию хорошего программиста. Выпущенный недавно новый вариант этого SDK, кажется, может существенно расширить «круг избранных».

ПРЕИМУЩЕСТВА ТЕХНОЛОГИИ LabVIEW

Дело в том, что в этом комплекте реализована поддержка технологий LabVIEW. Тому, кто уже работал с этим замечательным продуктом известной американской компании National Instruments, можно уже больше ничего не объяснять. Для остальных же кратко поясню, о чем идет речь. LabVIEW — графическая среда разработки для технических приложений. Программы в ней создаются не в виде текстовых модулей, а в виде привычных для каждого

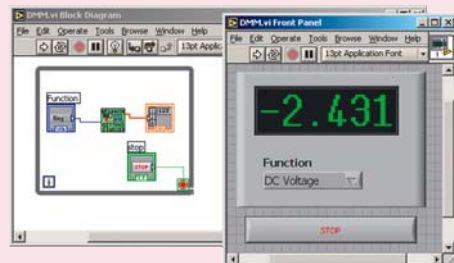


Рис. 1. Программа цифрового мультиметра в среде LabVIEW

инженера «электрических» схем. Только по проводам на этих схемах вместо электрических сигналов передаются программные данные, а вместо электроприборов — функции программы. Впрочем, приборы на этих блок-диаграммах тоже, как правило, присутствуют: и «настоящие виртуальные» — реальные устройства, подключаемые к компьютеру, и «совсем виртуальные» — индикаторы и управляющие элементы программы — «лампочки и кнопки» ее пользовательского интерфейса. Вот так может выглядеть в LabVIEW полнофункциональная программа цифрового мультиметра (рис. 1).

На этой блок-диаграмме центральный элемент — подмодуль устройства, поставляемый производителем оборудования в составе «драйвера LabVIEW».

Все тонкости работы с аппаратурой спрятаны там, пользователю видны только входные коннекторы управления и выходные коннекторы результатов измерений.

ИЗМЕНЕНИЯ НА УРОВНЕ БИБЛИОТЕКИ ПРОТОКОЛА

Вообще, высокий уровень абстракции — традиция драйверов приборов для LabVIEW. Новый комплект разработчика для АСК-3106 также претерпел изменения в этом направлении: в его базовой библиотеке (обычная динамичес-

кая библиотека Windows, не связанная с особенностями LabVIEW) появились функции, полностью осуществляющие наиболее общие операции работы с прибором. В более ранних версиях SDK пользователю предлагалось реализовывать их в программе самостоятельно. Сравните, например, два варианта пользовательской программы, выполняющей аналогичные действия (см. врезку).

Из изменений, не имеющих прямого отношения к применению LabVIEW, надо также отметить то, что теперь из одной библиотеки можно открыть од-

В предыдущей версии:

```

BYTE status, tmpnon;
double
    min_nonius,
    max_nonius;
unsigned int zero = 0, deadline;
int i, res;
static int started=0;

res = -1;

if (!started) {
    scopeData_ptr->start = 0;
    if (scopeData_ptr->triggerMode==TRIGGER_AUTO)
        scopeData_ptr->triggerMode = TRIGGER_NORMAL;
    ResetScope (0);
    signal_size =
        scopeData_ptr->trgDelay + scopeData_ptr->postTrgLength;
    if (signal_size > DATA_DIM) signal_size = DATA_DIM;
    StartRegistration (1);
    if (only_start) return started = 1;
    do {
        Application->ProcessMessages();
        if (!start_register || exit_flag) goto ERR;
        ReadStatus (&status);
    } while (!(status & STATUS_START_READ_DATA) && status);
    deadline = 1000*(scopeData_ptr->trgDelay+100) +
        timebaseTab[scopeData_ptr->timeBase];
    if (deadline < 55) deadline = 55;
} else { deadline = -1; zero=1; }
do {
    ReadStatus (&status);
    Application->ProcessMessages();
    if (!start_register || exit_flag) goto ERR;
    if ((trg_mode==TRIGGER_AUTO) &&
        (scopeData_ptr->triggerMode != TRIGGER_AUTO) &&
        (status & STATUS_DELAY_END) && !(status & STATUS_TRIGGER)) {
        if (!zero) zero = GetTickCount ();
        else if (GetTickCount() - zero > deadline) {
            SetTriggerMode ( TRIGGER_AUTO );
            zero = GetTickCount ();
        }
    }
    if ((scopeData_ptr->triggerMode == TRIGGER_AUTO) &&
        (GetTickCount () - zero > 2000*signal_size*
        timebaseTab[scopeData_ptr->timeBase])) status = 0;
} while (status & (STATUS_START_READ_DATA|STATUS_DELAY_END));
ReadTimeZoom (&tmpnon);
if (!GetMinMaxTimeZoom (&min_nonius, &max_nonius))
    good_nonius = ((max_nonius - min_nonius) > 40);
StartRegistration (0);
if (start_register) {
    if (good_nonius && (scopeData_ptr->timeBase <= TIMEBASE_STROB)) {
        nonius = (double)((tmpnon - min_nonius)/(max_nonius - min_nonius));
        nonius += GetParity ();
    } else nonius = 0;
    ReadData (BufferA, BufferB, signal_size);
}
res = 0;
ERR:
scopeData_ptr->triggerMode = trg_mode;
started = 0;
    
```

В новой версии:

```

if (!ACK3106_ResetScope (NULL))
    err = ACK3106_ReadWaveform (
        instrumentHandle, -1, -1,
        waveformArrayA,
        waveformArrayB,
        sactualPoints,
        sinitialX, &sIncrement
    );
    
```

новременно до 32 приборов АСК-3106. Раньше же для управления каждым прибором требовалось создавать отдельный процесс Windows.

ИНСТРУМЕНТ LabWindows/CVI – БИБЛИОТЕКА НА СИ И ФУНКЦИОНАЛЬНОЕ ДЕРЕВО

Рассмотрим подробно состав нового SDK. Как уже было сказано, в основе его лежит динамическая библиотека, реализующая все основные функции работы с прибором. Эта библиотека — самый

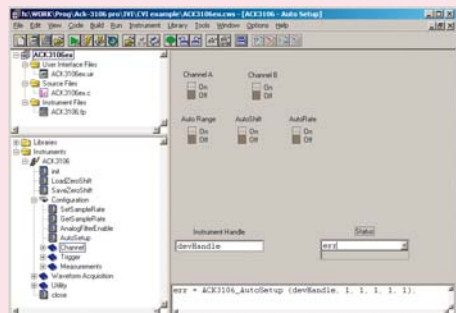


Рис. 2. Использование функционального дерева инструмента в LabWindows/CVI

нижний уровень из доступных пользователю. Все заботы об использовании библиотек интерфейсов связи и системных драйверов эта библиотека берет на себя, поэтому в дальнейшем именно ее мы будем называть библиотекой драйвера, а о более низких уровнях забудем.

Эта библиотека сопровождается заголовочным файлом на языке Си и файлом функционального дерева для LabWindows/CVI. LabWindows — еще одна среда разработки National Instruments, позволяющая использовать те

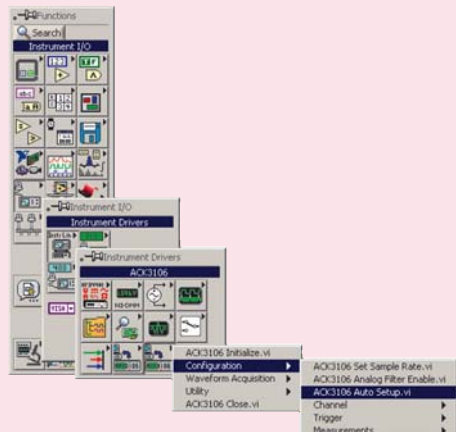


Рис. 3. Палитра функций LabVIEW для АСК-3106

же драйвера, модули и функции, что и LabVIEW, но программы в ней пишутся на обычном языке Си. Такой подход лучше всего подходит разработчикам, уже хорошо владеющим навыками традиционного программирования, но желающим также использовать и все наработки National Instruments в области анализа сигналов, пользовательского интерфейса или стандартных протоколов обмена данными.

Функциональное дерево — иерархическая структура, содержащая все функции, экспортируемые библиотекой драй-

вера, с полным описанием классов, функций и всех используемых параметров. На практике, для того, чтобы вставить функцию драйвера в программу, пользователь выбирает из меню нужный пункт и просто заполняет поля в появляющемся диалоговом окне (рис. 2).

ПАЛИТРА LabVIEW – ФУНКЦИИ

Функциональное дерево LabWindows/CVI полностью повторяется в LabVIEW в виде палитры функций (рис. 3). Процесс создания приложения с ее помощью отличается от традиционного программирования (в LabWindows/CVI) только заменой текстового описания алгоритма графической блок-диаграммой.

Пример блок-диаграммы, использующей функции библиотеки драйвера АСК-3106 таким образом, представлен на рис. 4. Как видите, при всей наглядности графического представления алгоритмов, программа может получаться довольно сложной. К счастью, в комплект разработчика входят дополнительные средства, позволяющие еще

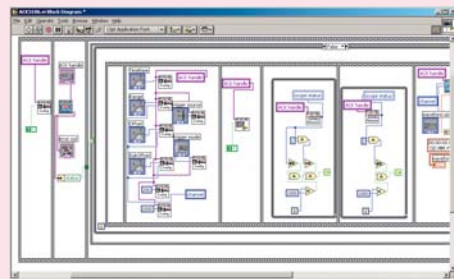


Рис. 4. Блок-диаграмма приложения АСК-3106 на основе функций библиотеки драйвера

более упростить создание приложений. Поэтому не будем далее разбирать построение блок-диаграмм из отдельных функций, а перейдем сразу к подмодулям и экспресс-инструментам.

ПОДМОДУЛЬ И ЭКСПРЕСС-ИНСТРУМЕНТ LabVIEW

Замечательное свойство LabVIEW — это его модульность. Если у вас уже есть готовое приложение для работы с прибором, далее вы можете использовать его как готовую функцию в следующем приложении более высокого уровня. Для АСК-3106 в комплект разработчика уже входит готовый подмодуль, реализующий полный цикл управления и измерения. На входы этого подмодуля можно завести всевозможные элементы управления осциллогра-

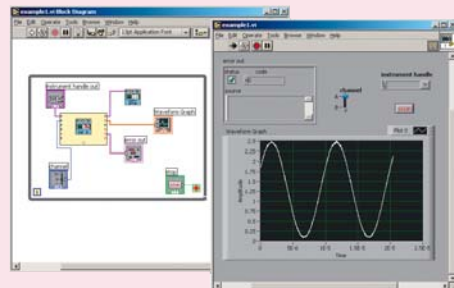


Рис. 5. Использование подмодуля АСК-3106

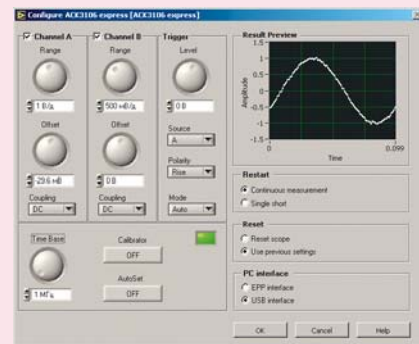


Рис. 6. Диалоговая панель настройки экспресс-инструмента АСК-3106

фом (ручки выбора диапазонов каналов, переключатели типов входов, регулятор скорости развертки и т. д.), а на его выходе получить готовый блок состояния прибора и массивы осциллограмм. Так как любой из входов можно оставить неподключенным (в этом случае подмодуль просто будет использовать значение параметра по умолчанию) и использовать готовую функцию автонастройки на сигнал, то готовое приложение, в автоматическом режиме снимающее и отображающее на экране осциллограммы, можно сделать совсем простым. Его блок-диаграмму и панель пользовательского интерфейса вы можете увидеть на рис. 5.

Близким к подмодулю в LabVIEW является сравнительно новое понятие «экспресс-инструмента». Фактически, это обычный подмодуль, параметры которого задаются специально созданным

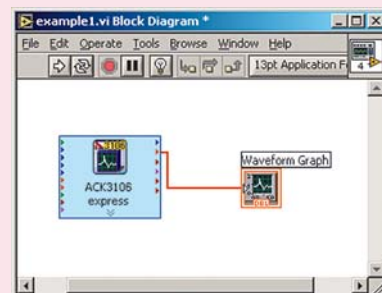


Рис. 7. Блок-диаграмма приложения, использующего экспресс-инструмент АСК-3106

для него диалоговым окном (рис. 6). С использованием экспресс-инструмента блок-диаграмма может быть собрана буквально из пары элементов (рис. 7).

ПРИЛОЖЕНИЕ. ПРОЕКТ ДЛЯ BORLAND C++ BUILDER

Но, конечно, со всеми упрощающими нововведениями в новом комплекте разработчика не исключена возможность использования традиционных способов программирования. Библиотека драйвера снабжается описанием не только в виде функционального дерева, но и в виде обычного файла справки Windows, а также примерами программирования в средах разработки Borland C++ Builder и MS Visual C++ без использования LabVIEW или LabWindows/CVI. Впрочем, об этом мы уже писали два года назад...