

LabVIEW ИНСТРУМЕНТАЛЕН ДРАЙВЕР ЗА УПРАВЛЕНИЕ НА ИЗМЕРВАТЕЛЕН МОДУЛ ПО ПАРАЛЕЛНИЯ ПОРТ

Георги Тодоров Николов

Технически Университет -София , ФЕТТ, кат."Електронна техника", Tel.: (+359 2) 965 26-40, E-mail: gnikolov@vmei.acad.bg

Nikolov G., A LabVIEW Instrument Driver Controlling Measurement Unit via Parallel Port. This paper presents a LabVIEW Instrument Driver controlling parallel port device with low level hardware functions like Pico's ADC-11. The presented ADC-11 instrument driver is created following the recommendations of National Instruments. The paper describes the structure, hierarchy, organization of instrument driver as well as functional body components and the integration of these components. Some specific routines are explained in more details. In the end of presentation the application example demonstrates a common use of the instrument driver and ADC-11 to create virtual measurement instruments.

1. ВЪВЕДЕНИЕ

Приложението на езиците за графично програмиране (*graphical programming*) в областта на измервателната техника многократно ускори скоростта на разработването и обновяването на управляващия софтуер в сравнение с традиционното текстово програмиране. Един от най-разпространените представители на графичните програмни среди, явяващ се като един вид стандарт за управление на измервателна апаратура, е продуктът LabVIEW на фирмата National Instruments (NI). Тази среда поддържа библиотека състояща се от повече от 1600 инструментални драйвери за управление на измервателни уреди. Допълнително производителите на измервателна апаратура предлагат самостоятелно разработени драйвери за продуктите си, предназначени за работа в LabVIEW. Огромният брой разнородна измервателна апаратура налага утвърждаване на концептуален модел, който да подпомага както разработчиците на инструментални драйвери, така и програмистите използващи тези драйвери за създаване на потребителски приложения. Един такъв модел пред назначен за графичен начин на управление е предложен от фирмата National Instrument [2], [3] и е приет от участниците в алианса VXI Plug&Play с наименованието VISA модел на инструментален драйвер [1], [2], [3]. В този VISA модел е унифициран достъпа до измервателната апаратура по стандартните интерфейси GPIB, RS-232/485, VXI и PXI. За съжаление обаче, в модела не е заложен механизъм за управление на модули осъществяващи връзка с компютър по паралелния порт.

Известно е, че такива модули съществуват [7], [8], [9], [11] и успешно могат да намерят приложение при изграждане на измервателни системи, като изпълняват функции не изискващи сериозни метрологични показатели.

В настоящия доклад е представен инструментален драйвер, работещ в средата LabVIEW, предназначен за управление на измервателен модул ADC-11, продукт на фирмата Pico Technologies. По същество модула ADC-11

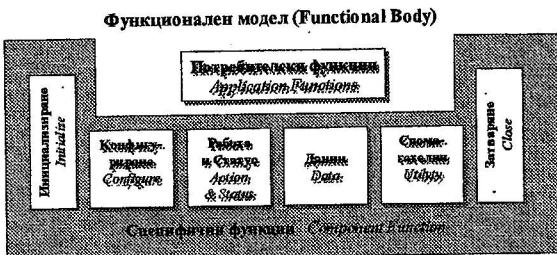
представлява единадесет канален десет битов АЦП, който се управлява директно от компютър по паралелния порт. Инструменталният драйвер е разработен съгласно инструкциите и препоръките на фирмата National Instruments за създаване на подобни приложения, така че успешно може да бъде интегриран в измервателна система заедно с разпространените драйвери за управление на измервателни средства по стандартизираните интерфейси като GPIP, RS-232/485, VXI или PXI.

2. ФУНКЦИОНАЛЕН МОДЕЛ НА ИНСТРУМЕНТАЛЕН ДРАЙВЕР

Инструменталните драйвери са предназначени за осигуряване на интуитивна интерактивна връзка между управлявания инструмент и потребителя, посредством виртуален лицев панел. Освен това тяхната архитектура трябва да бъде организирана по такъв начин, че да позволява гъвкаво включване на отделни обособени части на драйвера в програмния код на разработвана потребителска програма т.е да осигури възможност отделни функции за управление на измервателния уред да участват в цикли, масиви, математически функции и т.н.

Стандартният функционален модел на инструментален драйвер, показан на фиг.1 е описан в редица литературни източници ([1], [2], [3]), според които модулите изпълняващи специфичните функции се разделят на следните подгрупи:

- ♦ *Инициализиране (Initialize)* – модул, чрез който се стартира управлението на измервателния уред;
- ♦ *Конфигуриране (Configuration)* – тук са групирани функциите чрез които



Фиг. 1

уреда може да бъде приведен в зададен от потребителя режим на работа.

♦ *Работа и статус (Action/Status)* – тази подгрупа обединява модули управляващи самия измервателен процес и модули обслужващи заявките за

определяне на текущото състояние (статуса) на уреда.

- ♦ *Данни (Data)* – тук се намират функции определящи формата и вида на обменяните с измервателното устройство данни.
- ♦ *Спомагателни (Utility)* – това са модули изпълняващи функции, които не вземат пряко участие в измерването (самотестване, калибровка и др.).
- ♦ *Затваряне (Close)* – модул, извършващ необходимите процедури за затваряне на софтуерната връзка и освобождаване на системните ресурси.

За улеснение на проектантите и потребителите, в стандарта за LabVIEW инструментални драйвери са включени и няколко допълнителни средства. Това са програмни приложения за първоначално запознаване (*Getting Started*), йерархично дърво на модела (*VI Tree*) и пример за приложение на драйвера (*Application Example*).

3. ОРГАНИЗАЦИЯ НА ИНСТРУМЕНТАЛНИЯ ДРАЙВЕР ЗА ADC-11

Информацията илюстрираща накратко хардуерните възможности на измервателния модул ADC-11 е представена в таблица 1.

Разделителна способност (Resolution)	10 bits
Брой аналогови входове (Number of input channels)	11
Обхват на входното напрежение (Input voltage range)	0 - 2.5 V
Максимална честота на стробиране (Sampling rate) ¹⁾	15kS/s (15,000 samples per second)
Вградена защита на входа (Input overvoltage protection)	± 30V
Входен импеданс (Input impedance)	>1MΩ
Напрежение от цифровия изход (Digital output voltage)	typically 3-5 volts ²⁾
Изходен импеданс (Digital output impedance)	approx 1-3k ohms ³⁾
Входен куплунг (Input connector)	25 way female D-type
Изходен куплунг (Output connector)	25 way male D-type (to PC printer port)

1) измерена за 100MHz Sampling

2) Зависи от типа на компютъра и товара

3) Зависи от типа на компютъра

Особеност и основна трудност при разработката на инструменталния драйвер е, че фирмата производител на модула ADC-11, не предоставя информация нито за начина на свързване между входните измервателни канали и изхода на модула към компютъра, нито за командите на които тези канали се подчиняват. Поради тази причина, въпреки че съществуват решения за директно управление на паралелния порт от средата LabVIEW за конкретния случай те са неприложими.

Осъществяване на връзка между хардуера на модула и графичната среда за програмиране може да се осъществи като се използват драйверите предлагани от Pico Technology за управление на ADC-11 посредством текстови програмни езици. Тези драйвери са разположени в динамична библиотека adc1132.dll, която може да се придобие свободно от Web-сайта на фирмата (www.picotech.com/drivers.html). Извикването на текстово базираните функции, съставящи динамичната библиотека, от средата за графично програмиране LabVIEW се осъществява чрез функцията Call Library Function.

Спазвайки препоръките на National Instruments, разработените софтуерни модули, изграждащи драйвера се запазват в библиотека (в случая ADC11.lib), като първи трябва да фигурира модула илюстриращ структурата на драйвера или така нареченото йерархично дърво на модела – ADC11 VI Tree.vi (фиг.2).

Този модул представлява неизпълним код, като предназначението му е да запознае потребителя с архитектурата и организацията на драйвера.

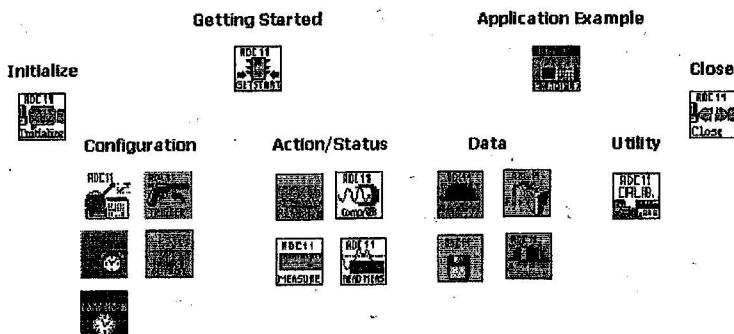
Както се вижда от фигурата драйвера се състои от 18 софтуерни модула разположени съгласно функционалния модел в съответните подгрупи.

♦ **Инициализиране (Initialize)** Тази подгрупа се състои от един модул - ADC11 Initialize .vi. Той инициализира драйвера, като адресира паралелния порт към който е присъединен измервателния модул. Допълнително е разработен код за генериране на грешки (Error Messages), при неправилно адресиране.

♦ **Конфигуриране (Configuration)** Модулите за конфигуриране на измервателния процес са пет и изпълняват следните функции:

ADC11 Dig Output vi – установява цифровия изход във високо ниво. Генериране на периодични импулси или импулсна последователност може да се постигне чрез написване на допълнителен код (поставянето на модула в структура, цикъл и т.н.)

ADC-11 PICO Technology



Фиг. 2

ADC11 Config Trigger.vi – конфигурира синхронизация по избран канал. ADC11 започва измерване, едва когато в канала постъпи сигнал отговарящ на избраното ниво и направление. Позволява и генериране на синхроимпулс след зададено закъснение (в μ s).

ADC11 Set Interval and Channel.vi – конфигурира стъпката за дискретизация във времето (в μ s), само за един избран канал.

ADC11 Total Time.vi – задава броя отчети и интервала необходим за придобиването им (в μ s).

ADC11 Set scanning function.vi – задават се стъпката (в μ s), каналите и реда на включване на каналите при сканиращо измерване.

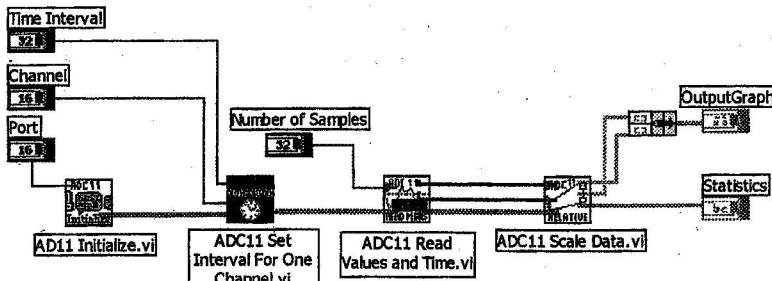
♦ **Работа и статус (Action/Status)** Четирите модула от тази подгрупа изпълняват следните функции.

ADC11 Read Single Channel .vi - извлича стойността на напрежението подадена на един от каналите в брой дискрети. Не дава информация за времето на отчитане.

ADC11 Read Value and Time.vi – отчита напрежението (в брой дискрети) и времето на отчет в (в μ s) само за един канал.

ADC11 Read Values and Times.vi – отчита напрежението (в брой дискрети) и времето на отчет (в μs) за групата канали конфигурирани с функцията **ADC11 Set scanning function.vi**

ADC11 Read Scanning Results.vi – задава се броя и началното време (в μs) от където да се извлекат отчетите.



Фиг. 3

♦ **Данни (Data)**

ADC11 Sent to File.vi – подготвя файл за запис на получените резултати

ADC11 Get from File.vi – извлича запазени чрез предходната функция данни от файл и ги подготвя за изобразяване.

ADC11 Set Arrays.vi – подрежда данните от извършено сканиране в отделни масиви според реда и броя на използвани канали.

ADC11 Scale Data.vi – преобразува резултатите от измерването (които са в брой дискрети и в μs) в избран от потребителя формат (V , mV , V_{rms} и др.)

♦ **Спомагателни (Utility)** Единствения модул от тази подгрупа **ADC11 Calibrate.vi** служи за определяне на калибровъчен коефициент. Тъй като в модула не е вграден еталонен опорен източник, необходимо е той да бъде подведен външно към избран канал, като се въведе и еталонната му стойност.

♦ **Затваряне (Close) ADC11 Close.vi**. Този модул завършва изпълнението на програмата, като освобождава заетия от модула паралелен порт.

ADC11 Getting Started.vi – служи за първоначално запознаване с работата на измервателното средство.

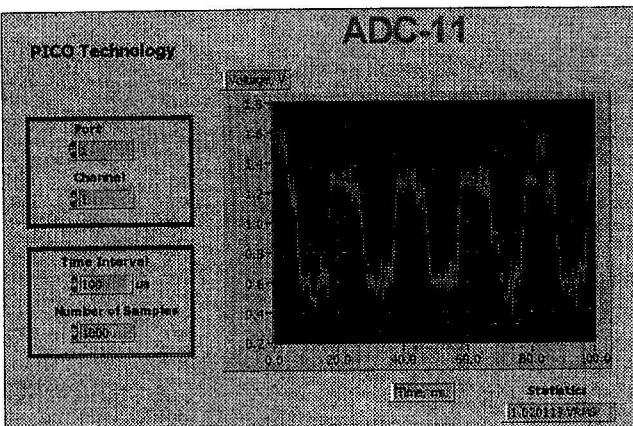
ADC11 Application Example.vi – показано е примерно приложение на инструменталния драйвер, като с помощта на отделните модули и допълнителни функции е съставена цялостна програма. Функционалната диаграма на приложението е показана на фиг.3, а лицевия панел на виртуалния инструмент на фиг. 4.

ЗАКЛЮЧЕНИЕ

Развитието на компютърните технологии постави нови предизвикателства пред инженерите и разработчиците работещи в областите на автоматизираните системи за тестване, измерване и контрол. Това е особено валидно за интегрирация софтуер, където концепциите на графичното програ-

мирание налагат коренна промяна в начин на мислене и изобщо в подхода, който трябва да се приложи от проектантите и разработчиците при изграждане на нови или обновяване на стари измервателни системи.

В настоящото изложение е представен инструментален драйвер за многоканален из-



Фиг. 4

мервателен модул ADC-11 на фирмата Pico Technology, управляван по паралелния порт на компютъра. Драйверът е предназначен за работа в графичната програмна среда LabVIEW, като при разработката са спазени изискванията и препоръките на фирмата National Instruments, съгласно стандарта за изготвяне на подобни приложения.

Използваният за разработката подход е приложим при проектиране и създаване на LabVIEW инструментални драйвери и за други измервателни модули управлявани по паралелния порт. Една такава разработка би позволила интегрирането на измервателни средства с ограничен брой хардуерни функции в една среда с прецизна, професионална апаратура.

4. ЛИТЕРАТУРА

- [1] R. Jamal, H. Pichnik "LabVIEW Applications and Solutions", Prentice Hall, 1999.
- [2] Noel A., "Developing a LabVIEW Instrument Driver", NI, App. Note 006, Austin, May 1998
- [3] Noel A., "LabVIEW Instrument Driver Standards", NI, App. Note 111, Austin, 1998
- [4] M. Rowe "What Does a Driver Do For You Anyway?", Test & Measurement World, March 1999.
- [5] K. Shah "Using IVI Drivers in LabVIEW" NI, Application Note 140, August 1999.
- [6] "Application Programs, Instruments Drivers and VXIPlug&Play Overview", VXI Technology, Technical Note 1998
- [7] Seán O'Leary, "Control the AD7710/11/12/13 from the PC", Analog Devices, 8/1994
- [8] Flite Electronics Ltd, "DSO-2102 Parallel Port Digital Storage Oscilloscope", Data Sheet, (<http://www.flite.co.uk/test/dso-2102.html>)
- [9] Intelligent Instrumentation, "DASport Technical Data", Texas Instruments Company (<http://www.instrument.com/pci/daspport.asp>)
- [10] Archana Shrotriya, "Writing Win32 Dynamic Link Libraries (DLLs) and Calling Them from LabVIEW", National Instruments, Application Note 087, June 1996.
- [11] MAXIM, "Automated Test Equipment on a Budget", May 2001, www.maxim-ic.com.