

Програмна реализация на система за експериментални изследвания на обекти в реално време на базата на LabJack U6 Pro

Георги Кръстев

Program Implementation of a System for Real Time Experimental Investigations of Objects with the Involvement of LabJack U6 Pro: The paper presents the Data Recorder program implementation of a system for real time experimental investigations of objects with the involvement of LABJACK U6 PRO.

Key words: Automation of scientific research, Data acquisition systems, Computer Systems and Technologies

ВЪВЕДЕНИЕ

LabJacks са USB/Ethernet базирани устройства за измерване и автоматизация, които осигуряват аналогови и цифрови входове/изходи, и др. [2, 3]. Устройствата осигуряват лесен за използване интерфейс между компютрите и физическия свят.

Разработеното програмно осигуряване е за устройство LABJACK U6 PRO и се нарича Data Recorder. Разработката е на базата на Qt на Nokia [4] и представлява част от разработената в Русенския университет мобилна изследователска лаборатория за целите на техническата диагностика на транспортни средства [1].

СТРУКТУРА НА DATA RECORDER – ОСНОВНИ ИНТЕРФЕЙСНИ КЛАСОВЕ

На фиг. 1 е показана UML диаграма на класовете на приложението Data Recorder в обобщен вид. Дадени са само основните класове на интерфейсно ниво. Асоциациите между тях са описани на български език, с цел по-лесно изясняване на предназначението на отделните класове и техните отношения.

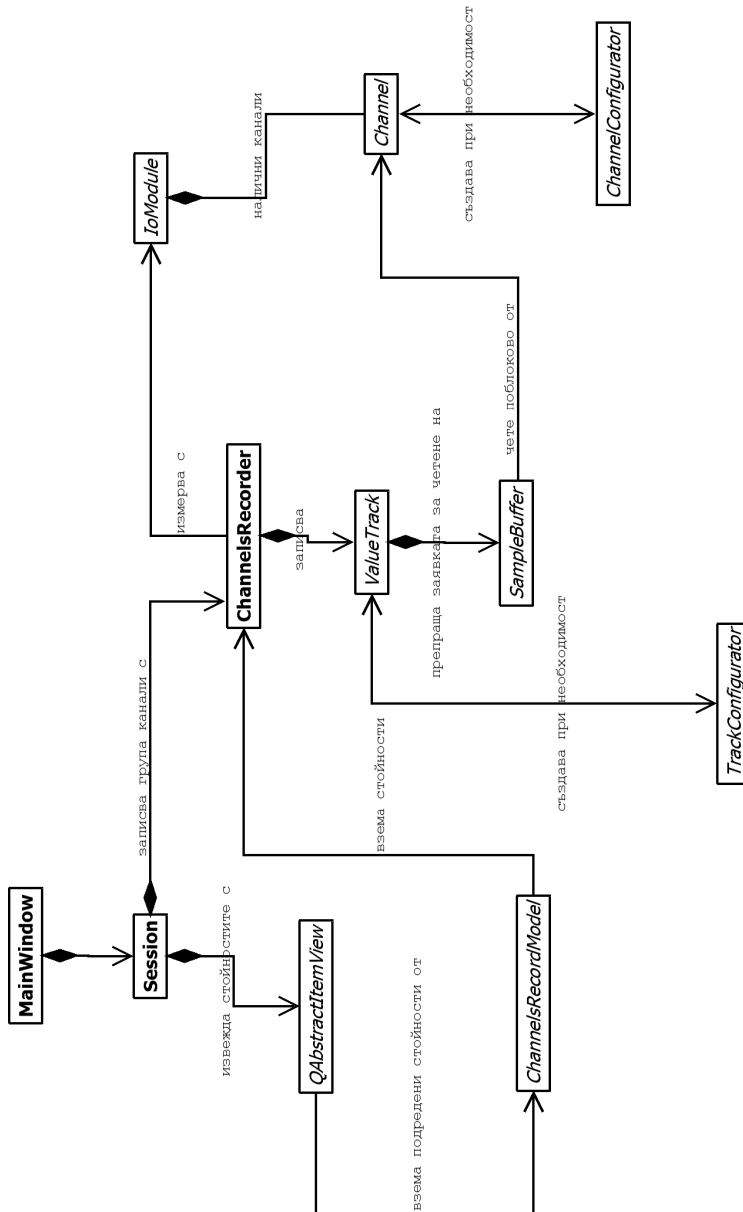
Клас MainWindow представя основния прозорец на приложението. Чрез главно меню и бутони с инструменти, поддържани от този клас, потребителят извиква основните функции – зареждане на конфигурации и данни от файл, създаване на нова конфигурация, стартиране и спиране на измерването и т.н. Той предава командите към класа Session.

Клас Session се грижи за изпълнение на основните операции, като синхронизира измерването, визуализацията, записа на данните, зареждането на данни в буфера. Той гарантира например забраняването на зареждане на данни от файл по време на измерване. При зареждане инициализира компонентите на системата с заредените стойности.

<interface>QAbstractItemView е вграден в Qt и заедно с **<interface>ChannelsRecordModel**, който е наследник на **QAbstractItemModel**, позволяват прегледа на данните по различни начини. Различни наследници на **QAbstractItemView** могат да визуализират едни и същи данни по съществено различни начини. **QAbstractItemModel** позволява, данните от произволен източник да бъдат преставени в модел различен от оригиналния. Тази технология е известна като Model-View. Тя е част от Model-View-Controller, но тъй като тук потребителят не влияе на измерените данни, се използва само част от нея.

Клас ChannelsRecorder съдържа няколко „записващи ленти“ за запис на измерените данни. Грижи си за синхронното извличане на данни от буферите на всички канали, тъй като поради спецификата на поточния режим, данните от каналите не могат да бъдат извличани независимо. Той включва командите към

измервателния модул и съдържа информация от вида [пътка, номер на отчет] и желания брой отчети. Има и метод за изчисляване на необходимия брой отчети от желано време на записване. Стартира отделна нишка, така че по време на обмен потребителският интерфейс да не замръзва.

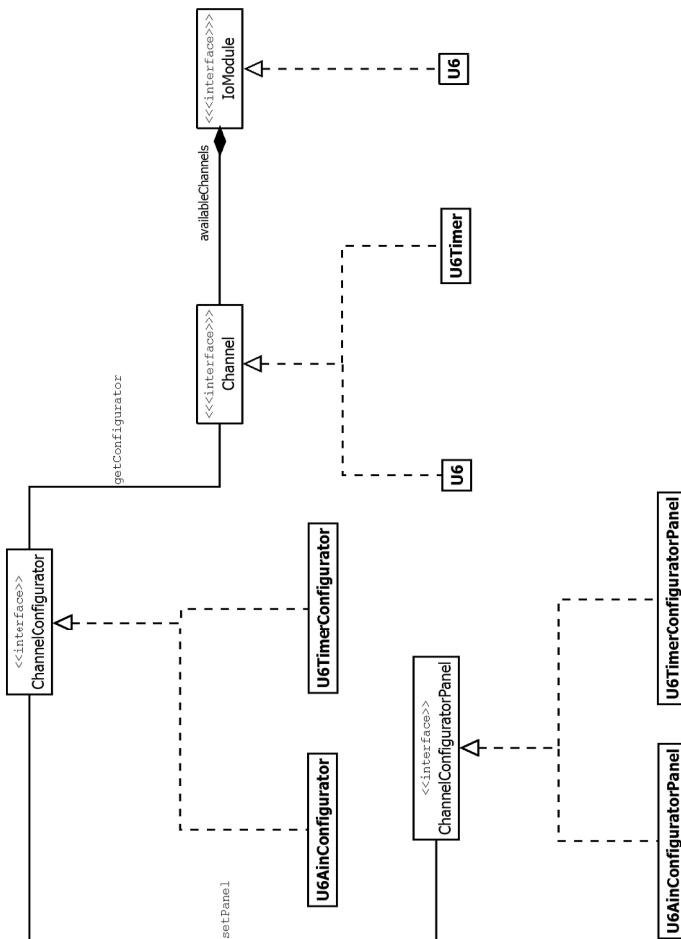


Фиг. 1. Обобщена класова диаграма на Data Recorder

<interface>ValueTrack съдържа информация за типа величината, обхвата и как от данните в буфера, да бъде получена реалната стойност. В зависимост от реализацията си, създава диалог със специфични настройки за величината, която се записва. Съдържа и буфера SampleBuffer.

<interface>SampleBuffer – тук се запомнят данните по време на отчитане. Буферът извиква метода за извличане на определен брой отчети от измервателния модул, като подава указател към свободна област. Осигуряването на паметта е грижа на буфера, а броят на отчетите за четене идва от ChannelRecorder.

<interface>IoModule – описва конфигурацията на модула, наличните канали и информацията на следените канали. Създава и манипулатора, с който се обменят данни с измервателния модул (фиг. 2). Притежава методи за задаване на размер на буфер, задаване на каналите за сканиране, честота на отчетите и всичко което засяга работата на модула като цяло.



Фиг. 2. Класове реализиращи IoModule и Channel за U6. Показани са и класовете за специфични да типа канал диалози

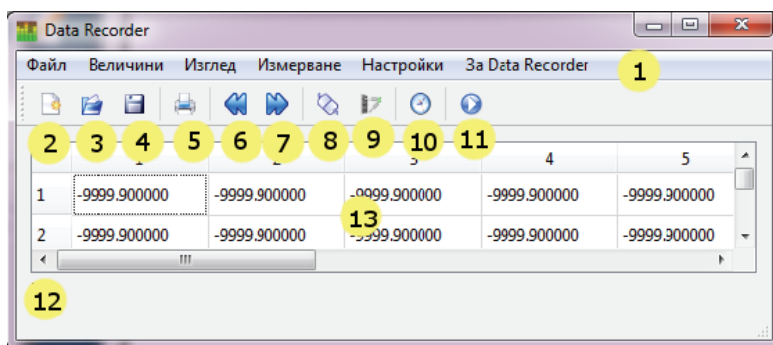
<interface>**Channel** – наследниците му съдържат настройките, специфични за всеки тип канал – аналогов вход, таймер и т.н. Създава при извикване на виртуалния метод `creatConfigurator` и специфичен за типа на канала диалог за конфигурация - **ChannelConfigurator**.

Препредава командата за четене на определен брой стойности от `IoModule`.

СТРУКТУРА И ОРГАНИЗАЦИЯ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

Потребителският интерфейс е стандартен за работа с Windows приложения. Главният екран е показан на фиг. 3 и съдържа:

- Главно меню (1) – оттук се извикват всички функции на програмата.
- Лента с инструменти за бърз достъп до важните функции (2-11).
- Поле за визуализация на измерването (13). В случая е показан буфера.
- Поле за състоянието/status bar (12) – тук се извеждат извършваните операции и съобщенията към потребителя.



Фиг. 3. Главен екран на приложението Data Recorder (намален)

ЗАКЛЮЧЕНИЕ

Проектираното приложение Data Recorder е за измервателен модул LABJACK U6 PRO и е реализирано в програмната среда QtCreator. Осигурява следната функционалност: конфигуриране на измервателните канали; поддържа работа в поточен режим на обмен на данни; визуализирането е таблично и графично; запазва получените данни и конфигурацията на каналите във файлове.

На базата на резултатите от процеса на реализация и тестването на програмата Data Recorder, може да се заключи че основните задачи са изпълнени и системата е работоспособна и приложима в практиката.

ЛИТЕРАТУРА

- [1] Кръстев, Г., Цв. Георгиев, Б. Христов, Х. Авакян. Мобилна изследователска лаборатория. Международна конференция Автоматика и информатика'09, София
- [2] LabJack U6 (Pro) User's Guide Revision 0.99, LabJack Corp. 2009 г.
- [3] <http://labjack.com>
- [4] <http://qt.nokia.com/files/pdf/qt-4.6-whitepaper>

За контакти:

Доц. д-р Георги Кръстев, Катедра "Компютърни системи и технологии", Русенски университет "Ангел Кънчев", тел.: 082-888 672, e-mail: gkrastev@ecs.ru.acad.bg

Докладът е рецензиран.