

ГРАФИЧЕН ИНТЕРФЕЙС ЗА ARDUINO UNO

Слави Димитров¹, Здравко Илиев²

¹ Минно-геоложки университет "Св. Иван Рилски", 1700 София, E-mail: shadowmana_bg@yahoo.com

² Минно-геоложки университет "Св. Иван Рилски", 1700 София, E-mail: iliev@mgu.bg

РЕЗЮМЕ. Разработен е графичен интерфейс за Arduino UNO. Той позволява наблюдение на цифровите и аналоговите входове на микроконтролера. Може да се използва в учебния процес и практиката.

ARDUINO UNO GRAPHICAL INTERFACE

Slavi Dimitrov¹, Zdravko Iliev²,

¹ University of Mining and Geology "St. Ivan Rilski", 1700 Sofia, e-mail: shadowmana_bg@yahoo.com

² University of Mining and Geology "St. Ivan Rilski", 1700 Sofia,, e-mail: iliev@mgu.bg

ABSTRACT. Graphical interface for Arduino UNO has developed. It allows monitoring of digital and analog inputs of the microcontroller. Can be used in teaching and practice.

Въведение

В последните години се наблюдава тенденция за все по-широкото използване на хардуер и софтуер с отворен код. Причините за това са много, като най-важните са: добрите характеристики на предлаганите продукти, които в повечето случаи не отстъпват на скъпоплатени аналози, ниската цена, липсата на необходимост от заплащане на лицензионни такси, големият обем свободно достъпна информация, неограниченото споделяне на опита с други разработчици и др.

От няколко години катедра "Автоматизация на минното производство" при МГУ "Св. Иван Рилски" използва платформата с отворен код Arduino както в учебния процес (З.Илиев, Д.Ташева, 2010), така и при разработки с определена практическа насоченост (З.Илиев, Д. Ташева, Н. Иванов, Т. Тодоров, 2011). Натрупаният опит показва, че тя се усвоява лесно и е подходяща за реализация на разработки с малка и средна степен на сложност, изискващи контрол и управление на ограничен брой входни и изходни сигнали, свързани с реализиране на алгоритми, нямащи специални изисквания по отношение на бързодействието. В същото време беше отчетено, че стандартната развойна среда не притежава интерфейс, позволяващ по удобен начин да се визуализират състоянията на входните сигнали към микроконтролера. Това се явява съществен проблем, когато платформата се използва за онагледяване на протичащите процеси при провеждането на лабораторните упражнения или като средство за събиране и начална интерпретация на данни

от реални обекти. Подобен интерфейс би се явил и удобно средство, подпомагащо тестването и настройката на специализирани устройства в процеса на тяхното проектиране. Във връзка с това беше поставена задачата да се проучи наличния и при необходимост да се разработи специализиран софтуер, който по удобен начин да предоставя на потребителя в цифров и графичен вид информация за състоянието на аналоговите и цифровите входове на платка с микроконтролер от серията Arduino Uno (<http://arduino.cc>).

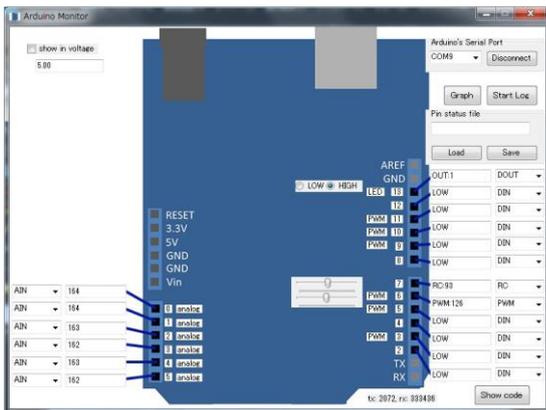
Съществуващи разработки

Широкото използване на платформата Arduino, както и обективната необходимост от усъвършенстване на потребителския интерфейс са довели до разработване на множество приложения, насочени към неговото подобряване. Повечето от тях са с отворен код. Приложенията са предназначени за работа под управление на различни операционни системи: Microsoft Windows, Linux, MacOS, OSX и др. Тук са представени някои от тях, които се характеризат с най-добри функционални възможности:

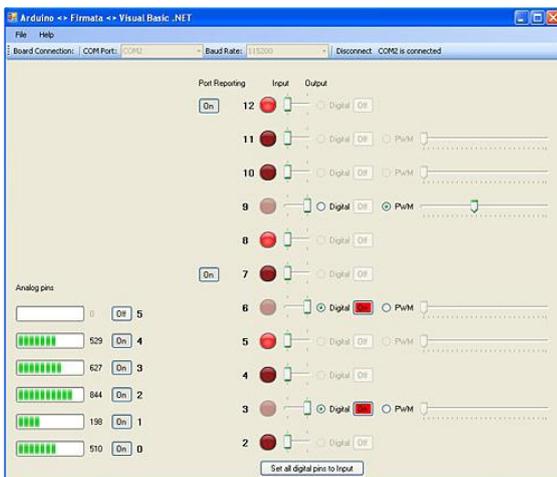
"**ArduinoMonitor**", версия 0.1 (Mitsunaga N.) е разработен с използване на библиотека "Sharpduino" (фиг.1). Позволява управление и наблюдение на състоянието на пиновете на микроконтролера. Предвидена е възможността за запис на числовата информация в CSV файл, а на графиките в PNG формат. Има възможност за записване и зареждане на конфигурация в XML формат.

“ArduinoFirmataVB” и “ArduinoFirmataVBExtended” са .NET framework библиотеки (Andrew Craigie), които съдържат методи за изпращане и получаване на съобщения чрез протокола Firmata (V2.0). Осигуряват връзката към серийния порт и имат вградени методи за задаване на името на порта и скоростта на предаване на данните.

Състоянията на пиновете се индикират с помощта на LED графични елементи, а резултатите от аналого-цифровото преобразуване се показват визуално чрез лента за развитие и в числов вид. Вградени са бутони за включено/изключено, които позволяват да се активира или дезактивира извеждането на информацията за даден пин (фиг. 2).



Фиг.1. “ArduinoMonitor”, версия 0.1.

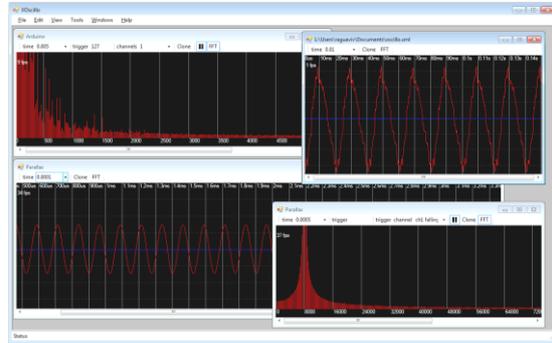


Фиг.2. “ArduinoFirmataVB”

“As3Glue” е реализирано приложение под Flash (Bjoern Hartmann, Kamperman. K), представляващо ActionScript 3.0 библиотека, която активира комуникацията между Flash/Flex/AIR приложения и Arduino платки. Дава възможност да се наблюдават цифровите и аналоговите пинове на Arduino, без да има нужда от допълнително програмиране. Позволява използването на няколко микроконтролера едновременно. Добавена е поддръжка за Arduino Mega.

“xoscillo” (Raul Aguaviva) представлява мулти-платформен (Microsoft Windows, Linux) софтуер с функции на осцилоскоп и логически анализатор (фиг.3). Дава

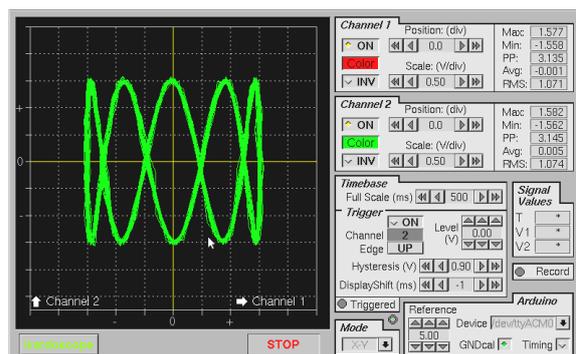
възможност за панорамна гледка, зареждане и записване на графики, приближаване и отдалечаване на графиката, показване на няколко сигнала едновременно, честотен анализ (FFT (Fast Fourier Transform)), филтриране (нисчестотен филтър), автоматично разпознаване на хардуера и някои допълнителни инструменти като декодиране на форми, изчисляване на главна честота по няколко метода. Поддържа Arduino и Parallax USB oscilloscope.



Фиг.3. “xoscillo”

“Arduinoscope” (David Konsumer) е базиран на “Poorman’s oscilloscope”. Включва и логически анализатор. Позволява пауза и запис на графиката и има възможност за настройка на броя на наблюдаваните пинове.

“Ixdaroscope” (Oskar Leuthold) е софтуерна имплементация на осцилоскоп под Linux (фиг.4). Поддържа Arduino Uno (или ATmega X8). Комуникацията може да бъде през сериен порт или по USB порт. Може да показва 2 канала с разделителна способност по вертикала - от 2 mV до 10 V на деление. Позволява измерване на сигнала с честота около 3000 опробвания в секунда за канал, моментна “снимка” на данните, графично изобразяване на записани данни от файл, определяне на основни характеристики на сигнала като минимална и максимална стойност, математическо очакване, дисперсия и др.



Фиг.4. “Ixdaroscope”.

“Poorman’s oscilloscope” (Arduino + Processing) представя визуално аналогов сигнал от Arduino чрез Processing (Sofian Audry). Разделителната способност е 10 bit, но честотата на сканиране е сравнително ниска. Принципът на работа е свързан с изпращане на прочетените стойности от аналогов pin 0 по серийна комуникация чрез код в Processing. Има възможност за приближаване и отдалечаване на графиката.

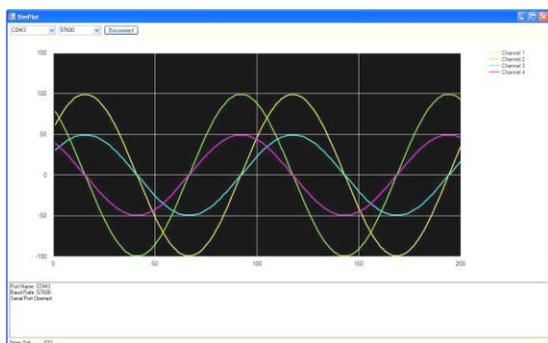
“Grapher” (gioblu) е разработен в средата Processing. Показва до 6 аналогови или цифрови входове от Arduino.

“Oscilloscope/Logic Analyzer” (Jonathan Oxeer, Hugh Blemings) е устройство, което използва данни от USB комуникацията на Arduino с компютър (фиг.5). Програмата му е написана на Processing. Поддържа Microsoft Windows, Linux и MacOS. Прочитането на един пин става за около 100 μ s.



Фиг.5. “Oscilloscope/Logic Analyzer”

“SimPlot” (Negtronics) е инструмент за изчертаване на данни от микроконтролер в реално време (фиг.6.). Приема данните чрез сериен порт и ги изчертава на екрана на компютъра. Другите му особености са - 4 канален, 16 битово преобразуване със знак, настройваеми дължина по X и мащаб по Y.



Фиг.6. “SimPlot”

Направеното проучване показва, че разработките са с различна степен на сложност, обхватност и бързодействие. В същото време нито една то тях не покрива изцяло поставените изисквания към необходимия интерфейс:

- условия за лесна и интуитивна работа на потребителя с него (изискване продиктуване от ориентацията му за учебни цели);
- възможност за конфигуриране на произволен набор от входни изходни канали, които да се наблюдават в отделни диалогови прозорци;
- съчетаване на цифрова и графична интерпретация на данните с възможност за динамичен избор на типа и измервателната скала.

Поради това се пристъпи към разработване на специализиран програмен продукт за визуализация на

състоянието на цифровите и аналоговите входове и изходи на микроконтролер от серията Arduino Uno

Същност на разработката

Изграденото софтуерно приложение е с име “ArduinOscilloscopeOut”. Реализирано е на платформата Microsoft .NET Framework 4.0 чрез езика за програмиране Visual Basic .NET. Средата за разработка е Microsoft Visual Studio 2010. Поддържаната операционна система за момента е Microsoft Windows XP. Visual Basic приложението използва библиотека “sharpduino”. Тя позволява да се управлява Arduino чрез протокола Firmata (<http://arduino.cc/en/Reference/Firmata>).

Текущата версия 0.2a е тествана със StandardFirmata 2.3 sketch от Arduino IDE 1.00 (Valsamidis, T). Целта е да се опрости връзката на .NET програмите към хардуера. Кодът е под свободен лиценз, което позволява да се добавят нови разширения и да се правят промени. Характерната за повечето подобни протоколи относително висока сложност е “скрита” от крайния потребител, а платката е настроена постоянно да изпраща съобщения за обновяване на стойностите на аналоговите и цифровите пинове, които се обработват от класа ArduinoUno – метод readAnalog.

Връзката между персоналния компютър и Arduino Uno се осъществява по USB кабел чрез протокол Firmata. Той е специализиран за комуникация с микроконтролери. Съвместим е с всякакъв софтуер. Проектиран е така, че да може да бъде разширяван и да се добавят нови функционалности. Поддържа до 16 аналогови и до 128 цифрови пина (<http://firmata.org>). Позволява да се пише потребителски firmware без да е необходимо да се създава нов протокол и обекти за използваната програмна среда.

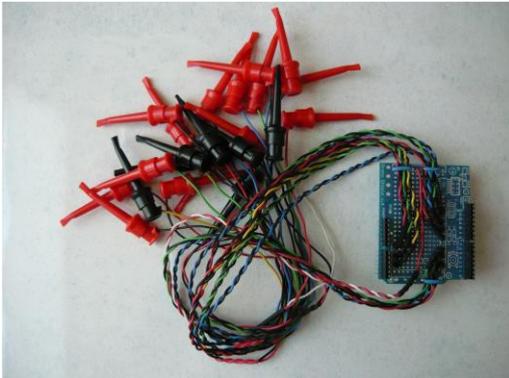
С цел по-лесно осъществяване на връзките на външните устройства с пиновете на микроконтролера, е разработен хардуерен модул (shield), представен фиг.7. Той се включва към Arduino Uno, като се монтира върху съществуващите рейки.

В основния диалогов прозорец (фиг.8) е предвидено меню със следните елементи - **File**, **Visualization - Analog Inputs**, **Visualization - Digital Inputs**, и **About**.

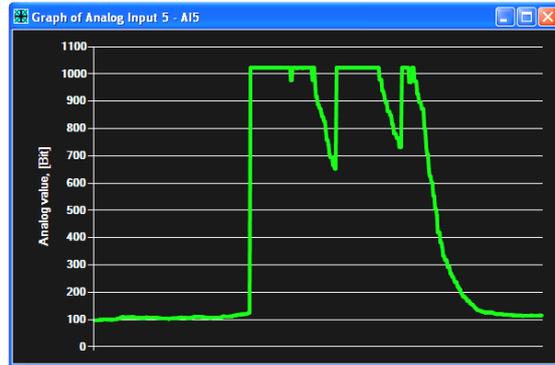
Чрез подменюто, подчинено на **File** се активира и деактивира връзката към платката и се осъществява изход от програмата.

До всички компоненти на менюто без **About** е осигурен бърз достъп (shortcuts) чрез «горещи клавиши».

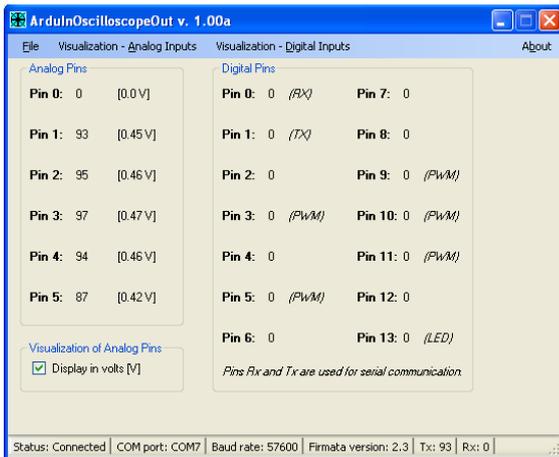
Всеки елемент и съответните им поделементи имат преки пътища чрез клавиатурата. Отбелязани са до името на елемента/поделемента.



Фиг.7. Хардуерен модул.



Фиг.9. Графичен екран.



Фиг.8. Основен екран на "ArduinOscilloscopeOut"

File има поделементи - **(Ctrl+N)**, **Disconnect (Ctrl+W)** и **Exit (Ctrl+Q)**. Функциите им съответно са: **Connect** - осъществяване на връзка към платката (отделен модален прозорец). **Disconnect** - прекъсване на връзката, **Exit** - изход от програмата.

Основната функция на приложението "**ArduinOscilloscopeOut**" е да визуализира състоянието на цифровите и аналоговите входове на микроконтролера от платката Arduino Uno. Тази информация в числов вид се извежда в основния диалогов прозорец. За аналоговите входове и изходи потребителят има възможност да определи дали желае да наблюдава 10-битовите цифрови кодове, получени в резултата на аналого-цифровото преобразуване или преобразуваните в напрежение стойности в границите от 0 до 5 V.

Освен тази основна информация в долната статус лента се показват състоянието на връзката, COM портът към който е включена платката и скоростта на комуникация (по подразбиране - 57600 bps).

Достъп до графичните режими за представяне на цифровите и аналоговите входове се осъществява съответно чрез елементите на менюто **Visualization – Digital Inputs** и **Visualization - Analog Inputs**. Всеки селектиран аналогов или цифров пин се представя в отделен графичен екран (фиг.9).

Харктерна особеност на разработката е пълната свобода на потребителя да конфигурира по свое желание броя и извежданите величини в графичните екрани.Тя позволява лесна съпоставка на графичните екрани. Това е много удобно в случай, че потребителя желае да наблюдава само определени входове, да прави сравнения и логически анализ в произволни визуални комбинации. Възможна е и паралелна визуализация едновременно на цифрови и аналогови входове в произволни конфигурации.Това е възможност, която не е предвидена в известните до момента подобни приложения.

С цел осигуряване на бърза и лесна инсталация са използвани инструментите на Microsoft Visual Studio 2010 за създаване на инсталационни файлове. Цялата инсталация отнема около 1-2 минути, в зависимост от ресурсите и възможностите на персоналния компютър. Файловете, необходими за правилното функциониране на приложението са "ArduinOscilloscopeOut.exe" - изпълним файл, "Sharpduino.dll" - библиотека за връзка към Arduino Uno и "ArduinOscilloscopeOut_instructions.txt" - кратки инструкции за ползване. Тези файлове се копират автоматично при инсталацията. Самата инсталация не прави допълнителни настройки в системата и регистрите на Microsoft Windows.

Заклучение

Чрез разработената програмна система "ArduinOscilloscopeOut" е реализирана поставената цел. Изграден е максимално опростен графичен интерфейс, чрез който лесно може да се следят състоянията на аналоговите и цифровите входове на Arduino Uno. Приложимостта не е свързана с конкретна задача или тип разработка. Може да се използва както в процеса на обучение, така и при разработване и тестване на практически ориентирани приложения, базирани на този микроконтролер.

Разработката може да бъде развита в бъдеще в следните насоки: поддържане на други операционни системи и други платки от фамилията Arduino (Leonardo, Mega 2560, LilyPad, Mega ADK, Fio, Pro, BT, Nano, Mini, Pro Mini и т.н.), използване на пълните възможности на "sharpduino" за управление на пиновете на микроконтролера, както и съхранение на информацията във файлове.

Литература

3. Илиев, Д. Ташева, Използване на Open Source Hardware платформа в учебния процес, Годишник на МГУ «Св. Иван Рилски», сборник доклади том 53, свитък III, стр. 180-185, София, октомври 2010 г., ISSN: 1312-1820
3. Илиев, Ташева Д., Иванов Н., Тодоров Т. Изследване на колебанията в металната конструкция при работа на багер ERs-710, Национална научно-техническа конференция с международно участие БУЛКАМК'11, стр.51-56. София, 6-7 октомври 2011 г. ISSN 1314-4537.
- Audry, S., Accrochages, "Poorman's oscilloscope (Arduino + Processing)", <http://accrochages.drone.ws/en/node/90>
- Craigie, A., "Arduino Firmata VB project", <http://acraigie.com/programming/firmatavb>
- Firmata, "Firmata protocol", <http://firmata.org>
- gioblu, "Grapher project", <http://arduino.cc/forum/index.php?topic=61236.0>
- Hartmann, B., "as3glue library", <http://code.google.com/p/as3glue>
- Kamperman, K., "ArduinoFlashcommunicationAS3 - As3Glue-bundle", <http://kasperkamperman.com/blogarduino/arduino-flash-communication-as3>
- Konsumer, D., "arduinoscope project", <http://code.google.com/p/arduinoscope>
- Leuthold, O., "Ixardoscope project", <http://sourceforge.net/projects/ixardoscope>
- Mitsunaga, N., "ArduinoMonitor project", <http://n.mtng.org/ele/arduino/arduinomon.html>
- Negtronics, "SimPlot project", <http://negtronics.com/simplot>
- Valsamidis, T., "sharpduino library", <http://code.google.com/p/sharpduino>