

ИЗПОЛЗВАНЕ НА PINGUINO В ОБУЧЕНИЕТО ПО ВГРАДЕНИ СИСТЕМИ

Георги ПЕТРОВ*, Николай ДИМИТРОВ**

Нов български университет, София 1618, бул. „Монтевидео“ №21, бл. 2, ет. 6, офис 609, E-mail: gpetrov@nbu.bg, nikoolayy1@gmail.com

Резюме. Този доклад представя резултати от експерименталното внедряване на програмируеми микроконтролери PIC модули Pinguino в обучението по вградени системи, комуникационни контролери и общо програмиране в департамент “Телекомуникации” на НБУ. Сравнени са възможностите и ценовите показатели на развойни системи, като Pinguino, Arduino, Olimexino и други ползващи затворен код. Обсъдени са възможностите за реализация на бързо приложно обучение на възможно най-ниски цени за едно работно място.

Ключови думи: програмиране с Pinguino.

Abstract. This report is presenting results from experimental implementation of programmable PIC microcontroller modules Pinguino in education of embedded systems, communicational controllers and general programing at department of Telecommunications in NBU. The capabilities and the price of development systems are compared with Pinguino, Arduino, Olimexino and others that are proprietary closed systems. The opportunities for realization of fast practical education for the lowest possible cost is achived with Pinguino MX220 modules.

Key words: programming with Pinguino.

УВОД

Развитието на съвременните системи по автоматизация, управление и вградени приложения води до все по-масова употреба на разнообразни комуникационни и други специализирани микроконтролери и микропроцесори. Това наред с развитието на мрежовите приложения поставя сериозно предизвикателство пред учебните програми в широка гама специалности, като: автоматика, индустриална информатика, вградени системи, комуникации, телематика, управление и др., като основната трудност е да се инкорпорират в различна степен учебни дисциплини и курсове свързани с употребата, разработката и вграждането на подобни микроконтролери от

различен тип. До скоро на пазара изобилстваше от голям брой развойни китове предимно разработвани от производителите на чипове или трети доставчици. Въпреки това крайната цена за внедряване на подобни системи в учебните програми, като финанси, технологии и време за обучение беше съществена. През последните години с развитието на самите микроконтролери и разширяване на техните възможности относно опции позволяващи препрограмирането на системите и увеличаване, както на обемът оперативна и програмна памет, интерфейсите и разрядността им доведе до по-масовото развитие на препрограмируеми системи от високо ниво. Тези системи разполагат със стандартни интерфейси: RS232, USB, Ethernet [1] и се разработват от нови компании, като проекти с отворен код, при които финалният продукт не изисква наличието на скъп програмиращ хардуер у всеки програмист и любител. Тези системи са фабрично заредени с миниатурна микро програма наричана bootloader, чиято цел е да позволи на потребителя да препрограмира системата без да се налага ползването на стандартен фирмен програматор. Наред с това тези системи получиха допълнително развитие с по-масовото им разработване, като софтуер с отворен код и безплатен софтуер, което ги направи привлекателни, лесни за усвояване и най-важното, достатъчно евтини така че масовият потребител, учащ се и преподавател да може да ги използва максимално ефективно. Системите за които става въпрос в този доклад са разновидност на една цяла група микроконтролерни модулни приложения, сред които са: *ARDUINO*, *MAPLE*, *PINGUINO*.

Защо XXX-ino?

Когато говорим за избор на обучителна платформа за масово приложение в университетите, коледите и училищата, където микроконтролерите не са основният фокус на обучението следва да се съобразим с редица важни ограничения формиращи крайната цена на обучението. Изборът не е проста функция на личните предпочитания на преподавателя, наличното проектно финансиране и най-вече удобство и кониоктура с която преподавателският и академичен състав е свикнал. Следва да се съобразим с онези фактори, които ще направят нашето обучение ефективно, разнообразно и привлекателно, достатъчно бързо и отговарящо на съвременните тенденции в развитието на вградените компютърни системи, така че обучените да добият комплексен набор от умения и познания, с които те да станат максимално независими разработчици от висок клас и при това за кратко време. Тук основният въпрос е как да се постигне бързо крайно приложение на възможно по-ниска цена, а когато става въпрос за студенти това също значи и евтина платформа за разработка, достатъчно надеждна и позволяваща развитие и разработка с допълнителни модули (самоделни или готови). Класическият подход ползван до преди 2-3 години беше ползването на учебни китове, най-вече произведени от конкретният производител на микроконтролери: Atmel, Microchip, Analog Devices [2] и др. които въпреки ниската си крайна цена (т.н. микро китове) системно изискваха наличието на програмиращ хардуер, чиято цена често надвишаваше три-четирикратно цената на китовете, а също така повечето от тях се разпространяваха със софтуер работещ основно под ОС Windows. Това означаваше, че за функционалното оборудване на един стандартен учебен курс от 15 работни места (компютърно базирани работни места по стандарт в учебна зала) единичната инвестиция за даден курс, заедно със софтуерните пакети надвишаваше 1400лв. за хардуера, като софтуерът в зависимост от исканият краен резултат можеше да бъде безплатен или надвишаващ 500\$ за 1 бр. [3] Типичният пример, който може да бъде даден е софтуерът на компания CCS (Custom Computer Services, Inc.). От друга страна когато говорим за микроконтролери, и особено за достъпни системи и крайни продукти правени с тях, както цена така и по сложност в по-голяма част от случаите обучението се базира на PIC или AVR (8, 16, 32 битови контролери), както и някои разновидности с ARM.



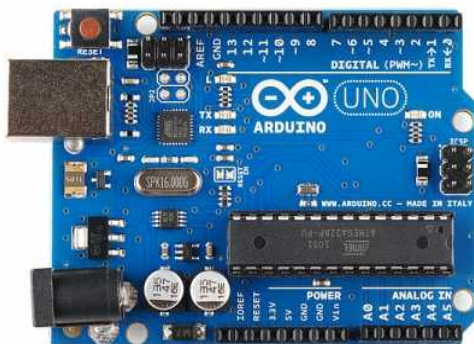
Фигура 1. Типичен скъп приложно ориентиран кит за разработка с 16 битови PIC контролери на компанията CCS, това е представител на старата генерация развойни модули.

Този софтуер макар и бесплатно разпространяван от повечето производители, като AVR Studio, uVision Keil или MPLAB със съответните C/C++ компилатори за различните версии контролери не може да бъде директно приложен за постигане на бързи и задоволителни резултати в обучението. Персоналният опит на авторите в НБУ показва, че миграцията между тези развойни системи в условия на намален учебен хорариум и бюджет не води до съществени преимущества в самото обучение, и по-специално крайният резултат е незадоволителен. Крайният резултат в учебните програми на НБУ е постигането на такова ниво на знания и умения при което студентите сами да могат да създават функционално пълни приложения (не конкретни програмни задачи) и при това да могат да навлизат в детайлите на съответните системи при тяхно желание или когато професионалният им живот го наложи. Това предполага, че ползването на фабрично предлаганите интегрирани среди за разработка е неефективно и често напълно безполезно. (Студентите не могат да програмират ефективно на C без да познават архитектурата на контролерите, основите на електрониката и респективно управлението на хардуер с асемблерни програми, а също така това изисква и задълбочени познания за конкретния C синтаксис на всеки един производител.) Съвременните изисквания изискват кросплатформеност, ниска крайна цена, възможност за зареждане на малки до големи приложения, мрежова свързаност и ред други технически изисквания и не на последно място ниска консумация на електричество. Естествено 8 битовите контролери имат и ще продължават да имат място в подобни минимистични системи с ниска консумация, обаче не можем да отречем, че мрежовият подход на новите индустриални и информационни приложения изисква хардуер позволяващ по-високо бързодействие, многозадачност, мултиинтерфейсна поддръжка, възможност за свързване през безжични и жични комуникационни интерфейси, поддръжка на локални памети с голям обем (FLASH карти и дискове), което прави 8 битовите контролери практически неприложими. От друга страна програмирането на

асемблер е добро начало за професионалисти, чиято основна работа ще бъде свързана със системното програмиране и дизайн на ниско ниво. Това обучение считам за особено важно, на за жалост притиснати от хорариума и финансовите ограничения подобни курсове са допустими само в държавните и други големи учебни програми. Проблемът тук не е само хорариума, но и цената на цялата развойна среда: софтуер, хардуер и програматор, което дефакто прави тези учебни модули и методи трудно приложими и често непостижими за самостоятелна работа на отделните студенти поради високата цена. В близкото минало частична рудекция беше постигната с ползването на gcc компилаторите в Linux където отдавна се предлагат безплатни пакети с библиотеки за програмиране на AVR контролери. Въпреки това този метод не е модерен! Интересен и дискутируем е въпроса какво ще наричаме модерен учебен подход? Това е съвкупност от множество неща, често трудно дефинирани в началния момент, когато решаваме да създаваме учебни дисциплини отговарящи на новите тенденции. А когато фокусът е бързо, евтино и приложно, нещата добиват съвсем различен характер.

Arduino

Преди Arduino съществуват множество опити за направа на крос платформени развойни системи за микроконтролери с отворен код. Обаче едва Arduino достига такава степен на интеграция – хардуер и софтуер + приложност, че да направи внедряването на цифровите учебни системи достъпно дори в детските градини (Фигура 1). Според създателите му: „Ардуино е малка платка с огромни възможности и отворен код. Снабдена с мощни ATmega микроконтролери, Ардуино изпълнява ролята на "мозъка" на роботи и интерактивни проекти. Ардуино може да "усеща" заобикалящата го свят с помощта на различни сензори (напр. за звук, светлина, докосване, движение и др.) и да реагира на промени в заобикалящата среда (напр. като задвижва електромотори и командва уреди издаващи звук или светлина).“



Фигура 2. Arduino Uno досега това е най-разпространеният и лесен за употреба кит позволяващ разработка с контролерите на Atmel.

Самият проект започва в гр. Иврея, Италия (към фирма Оливети). Проектът стартира през 2005г., като идея да се създаде малка платка за управление на работи и устройства предимно за студентска разработка. През 2012г. на пазара са продадени над 320,000 платки. Основатели на компанията са Massimo Banzi и David Cuartielles кръстили своя проект на името на Arduino от Иврея, основна историческа личност в града. *Буквално думата значи добър приятел.* Самият проект включва създаването на напълно интегрирани и крос платформена среда за разработка с отворен код. Самият код е написан от колумбийски студент Hernando Barragán, като често от магистърската му теза в института Interaction Design в Иврея под ръководството на M.Banzi и C.Reas.

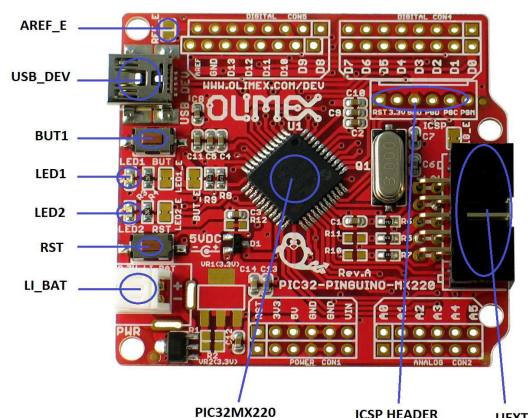
Pinguino

Pinguino е развойна система за PIC микроконтролери, които поддържат вътрешно програмиране (8 и 32 битови версии за момента). Българската компания OLIMEX предлага най-евтиният възможен вариант на Pinguino базова развойна платформа MX220. Системата притежава процесорно ядро **40 MHz MIPS32® M4K®**, MIPS16eR режим на работа до 40% по-малък размер на кода от Arduino поради специфика на компилатора без ползване на библиотечни макро дефиниции. Скорост на обработка на цифровите инструкции е 1.56 DMIPS/MHz, процесорът поддържа ефективно програмиране на C и асемблер, важно свойство е наличието на 32x16 умножител за един такт и двутактов 32x32 битов умножител. Важно свойство на Pinguino е че създателите му гарантират абсолютна поддръжка в режим CDC серийна USB2.0 емуляция, което не е постигнато в Arduino поради ползването на споделен сериен порт за програмиране и комуникация през FTDI чип. В таблица 1 са показани сравнителни характеристики на трите платформи с отворен код.

Таблица 1. Сравнение на трите развойни системи за разработка и обучение в стил Arduino

	PIC32 PINGUINO-MX220	ARDUINO-UNO	Maple ARM STM32F103RB
CPU	32 bit	8 bit	32 bit
CLOCK	40MHz	20 MHz	72 MHz
FLASH	32 KB	32 KB	128 KB
RAM	8KB	1 KB	20KB
USB	YES	NO	YES
LOW PWR	YES	NO	YES
GPIO MAX	40 MHz	5 MHz	40 MHz
GPIO PIN	13	14	39
ADC	1.1 MSPS	15 KSPS	1 MSPS
ADC	10 bit	10 bit	12 bit
DMA	YES 4 CH	NO	YES 7 CH
UEXT	YES	NO	NO

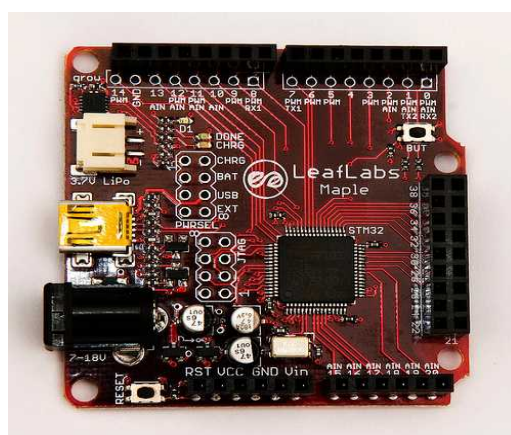
PRICE	20 BG	42 BG	61 BG
-------	-------	-------	-------



Фигура 3. Pinguino MX220 най-евтиният и достъпен развоен кит готов за употреба от всеки.

MAPLE

Що се отнася до възможностите за обучение по програмиране на ARM процесори не може да не споменем Maple - 72 MHz ARM cortex чип осдобен подходящ при обучение по програмиране за мобилни и фиксирани приложения с понижена консумация, преносима техника и други стандартни продукти. Различното в този ARM кит е средата за програмиране, която е доста по-дружелюбна от класическите uVision C компилатори, при които разработчика трябваше да познава поне 80% от архитектурата на системата [4]. За тези които харесват Arduino, Maple е технически съвместим продукт.



Фигура 4. Maple кит за разработка с Cortex-M3 контролери.

Заклучение

Краткият ни опит с Pinguino показва, че развойната среда копираща частично Arduino, но включвайки по-голям набор интерфейсни функции и поддръжката на пълноскоростни USB 2.0 библиотеки съчетано със свръх ниската цена от 24лв за краен продукт с кабел за програмиране прави MX220 платката идеална за начинаещи и напреднали ученици и студенти, а също така и подходяща за професионалисти от широк спектър на хардуерното и софтуерно производство. Ползването на Питон, като основа на средата и някои особености на компилатора правят кода на Pinguino значително по-малък и бърз от този на конкурентните платформи. Също така и разрядността на контролера, както и двуту базови работни честоти в които се предлагат другите модули 40-80MHz дават широка перспектива за дълговременното му ползване в широк спектър приложения от елементарна роботика, автоматизация до малки мултимедийни системи. Едновременно с това базовият 32 битов модул не би могъл да замени 8 битовият еквивалент, който препоръчваме при по-специализирани учебни курсове засягащи детайлно архитектурата на микроконтролерите. От 2012г. Pinguino се предлага с поддръжка и на платформа с PIC 18FXXX. Следва да отчетем факта, че за ефективно и пълно обучение бъдещите специалисти следва да преминат цикъл лекции върху разнообразни платформи и архитектури, като неминуемо бъде засегнато програмирането на ниско ниво, ползването на специфични C компилатори предлагани за високо надеждни приложения и Linux базирани PIC платформи [5]. Едновременно с това Pinguino е първата напълно отворена и евтина ресна за изработка дори в домашни условия платформа с отворен код и хардуер, което е ключовият аспект на съвременните вградени системи подсиуряващо бъдещата им висока надеждност, отстраняемост на грешките, достъпност, прозрачност и кросплатформеност, която фокусира разработчиците над конкретното приложение, а не върху конкретният хардуер. Можем да считаме, че така появилата се система ще продължи своето развитие и поддръжка на повече контролери и езици за програмиране - basic в най-близко бъдеще.

ЛИТЕРАТУРА

- [1] Michael McRoberts, *“Begining Arduino”*, 2010
- [2] Т.Стефанова, Г.Петров, П.Цветков, И.Богомилов, *“Приложение на обучителни модули в практическото обучение по аналогова електроника”*, Метрология 2009, ISSN 1313-9126
- [3] П.Цветков, Г.Петров, П.Илиев, *“Software development environments for image acquisition and processing”*, Challenges in higher education and research in the 21st century Созопол 2006 – ISBN-10: 954-580-206-5
- [4] Morgan Kaufmann, *„ARM System Developer's Guide: Designing and Optimizing System Software (Computer Architecture and Design)“*, 2004
- [5] O'Reilly, *„Programming Embedded Systems: With C and GNU Development Tools“*, 2006