

Универзитет у Крагујевцу  
Факултет инжењерских наука



## Семинатски рад из предмета Архитектура рачунарских система

Тема:  
Arm 64 pin  
LPC 2148

Студенти:  
Јован Петровић 556/2015  
Александра Нешић 568/2015  
Софија Јаковљевић 574/2015

Предментни професор:  
Александар Пеулић

## САДРЖАЈ

1. Увод.....	2
2. Архитектура .....	3
2.1 MikroBoard.....	3
2.2 LPC2148 .....	8
2.3 UNI-DS6.....	21
3. Пројектни задатак .....	25
4. Закључак .....	29
5. Литература.....	29
6. Прилог кодови.....	29

## **1.Увод**

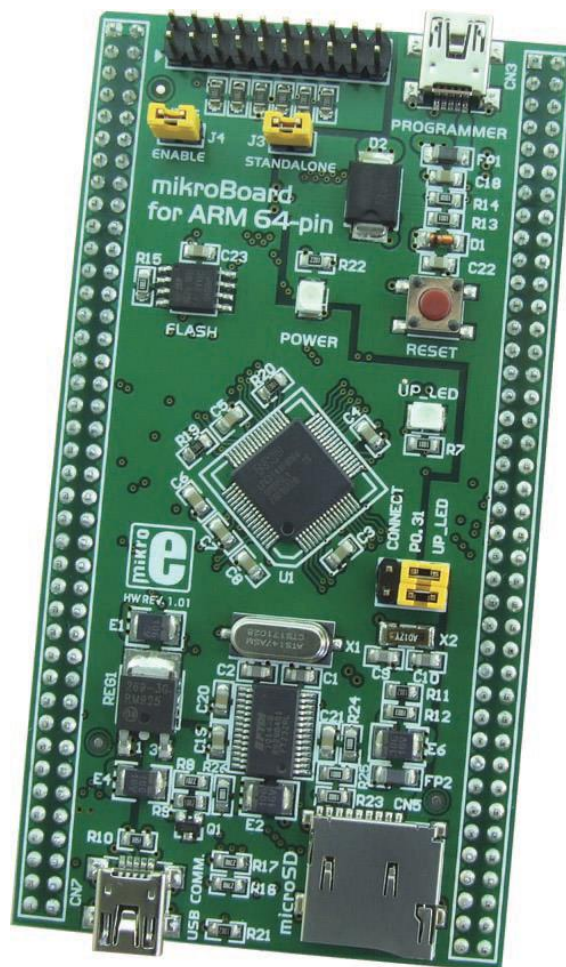
У овом раду ћемо приказати, коришћењем развојног система Uni DS6 и микроконтролера Arm 64 pin који у себи садржи процесор LPC 2148, паљење и гашење диода. Креирање .hex фајла ћемо реализовати преко програма Keil  $\mu$ Vision5, а покретање преко програма Flash Magic.

## 2. Архитектура

### 2.1 MikroBoard

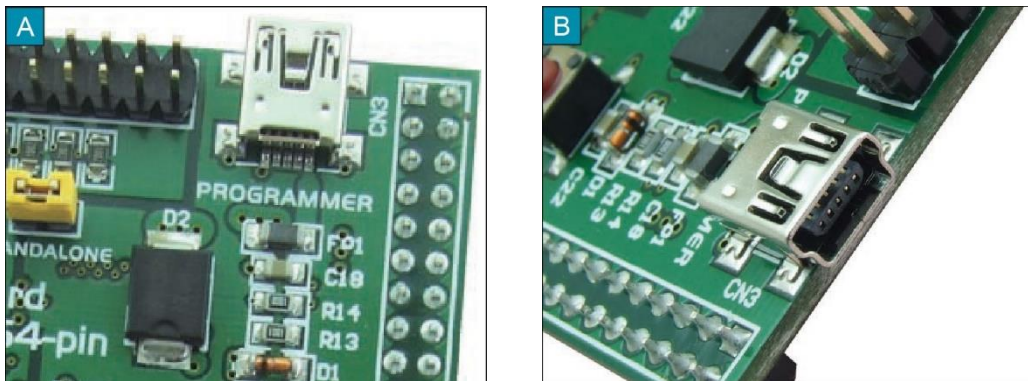
#### 2.1.1 Опште одлике

MikroBoard плоча је намењена за коришћење уз развојну плочу EasyARM v6 али се такође може користити као засебан уређај. На њој се може наћи један микроконтролер из широке понуде, у овом случају ARM 64. На плочи се поред микроконтролера налазе и USB конекторима, флеш модул, улаз за microSD картицу, USB UART, регулатор напона и прикључци који омогућавају повезивање са развојним системом.

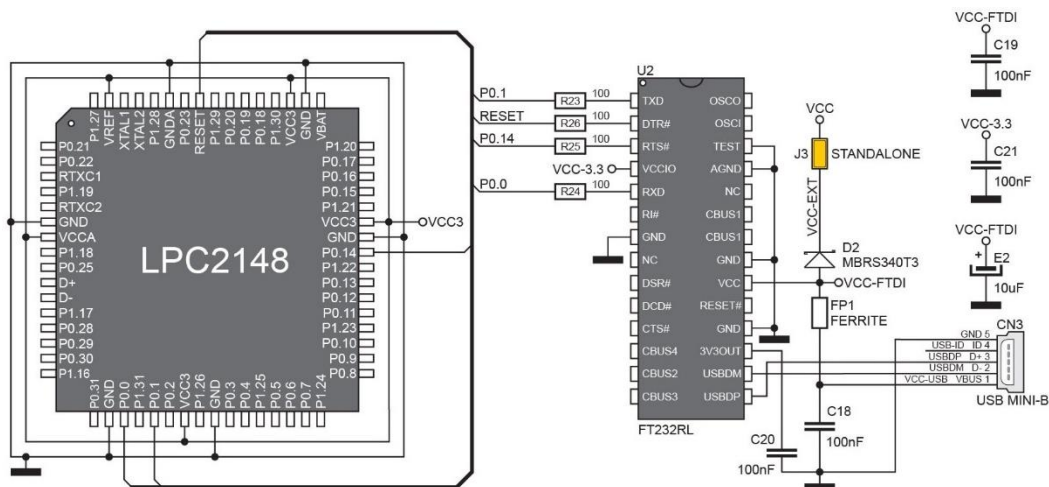


## 2.1.2 Програмирање микроконтролера

Микроконтролер се програмира помоћу bootloader-а или JTAG програматора. Коришћење bootloader-а је омогућено помоћу кода који се учитава у микроконтролер. Да би се микроконтролер испрограмирао помоћу bootloader-а потребно је повезати плочу са рачунаром помоћу CN3 конектора и USB кабла. Рачунар шаље .hex код микроконтролеру помоћу неког од програма, у овом случају програма *FlashMagic*.



USB прикључак са ознаком CN3 се повезује са UART модулом уграђеним у микроконтролер путем модула FTDI (FT232RL).



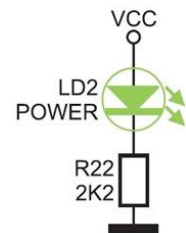
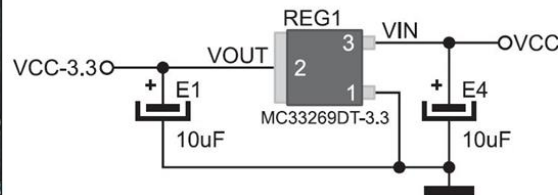
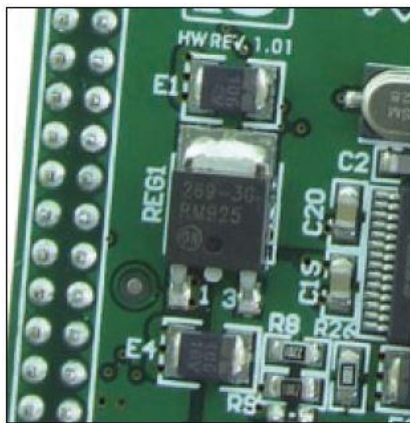
Када се MikroBoard користи као засебан уређај, потребно је поставити џампер J3 на плочи. Уколико је плоча повезана са EasyARM v6 развојним системом, џампер J3 треба скинути.

Микроконтролер се може програмирати уз помоћ JTAG програматора. JTAG програмтор се такође може користити за тестирање рада микроконтролера.

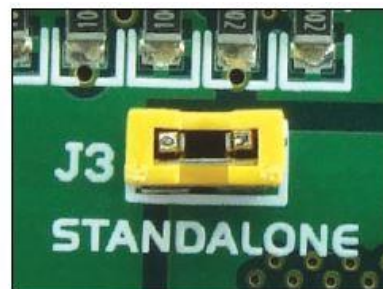
Да би се JTAG програматор користио, треба поставити џампер на позицију J4- ENABLE. Када се JTAG програматор не користи, треба уклонити џампер са позиције J4.

### 2.1.3 Регулатор напона

Микроконтролер ради под напоном од 3.3V, док се плоча напаја са напоном од 5V преко USB кабла.

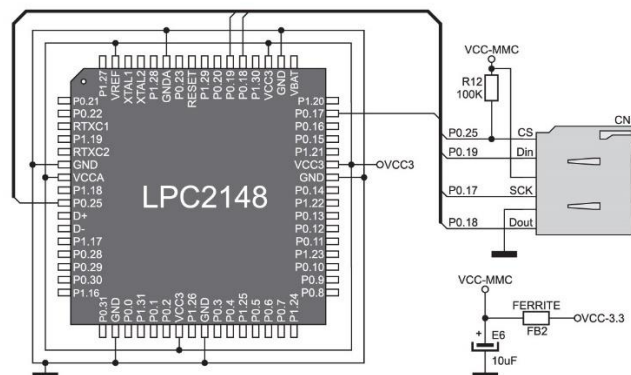
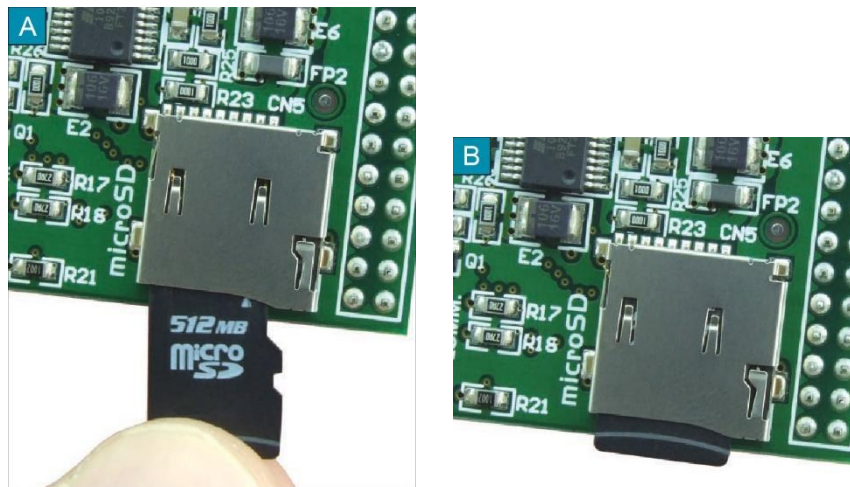


У случају да се MikroBoard напаја са развојног система EasyARM v6, функција регулатора напона се не мења, само је потребно уклонити џампер J3-STANDALONE.



## 2.1.4 MicroSD прикључак

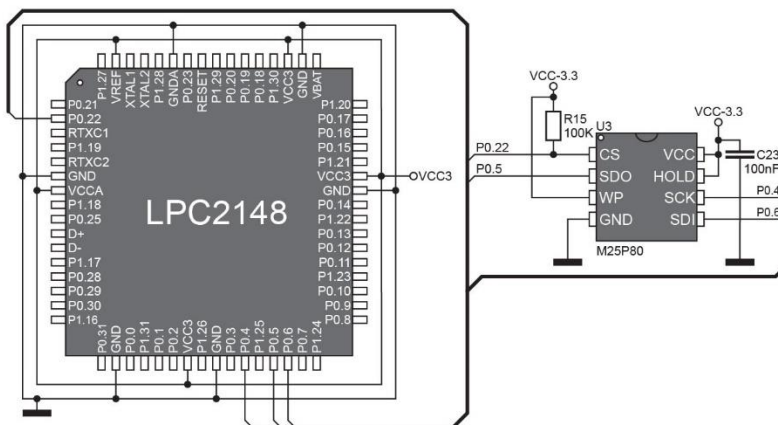
На плочи постоји прикључак обележен са CN5 који омогућава коришћење microSD картице. Прикључена microSD картица даје микроконтролеру више простора за смештање меморије. За комуникацију између microSD картице и микроконтролера се користи SPI.





### 2.1.5 Флеш модул

Флеш модул пружа додатних 8Mb флеш меморије која је повезана са микроконтролером преко SPI.

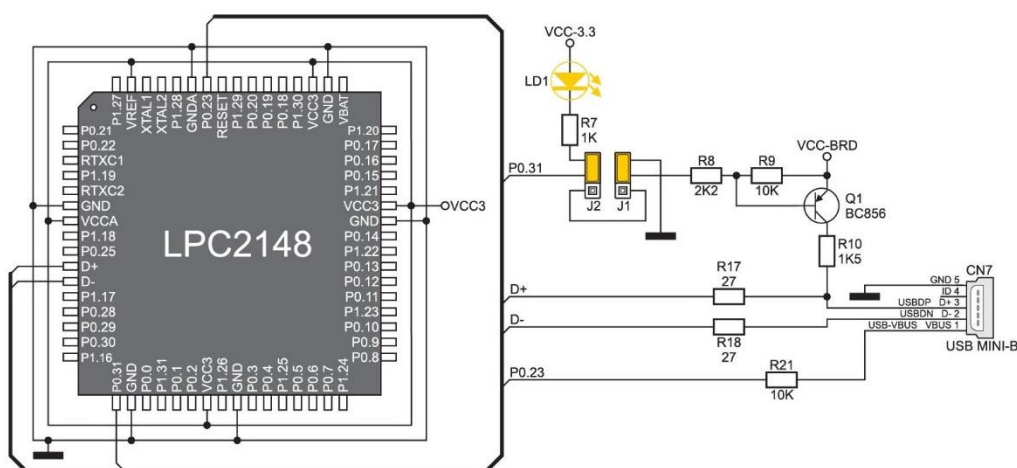
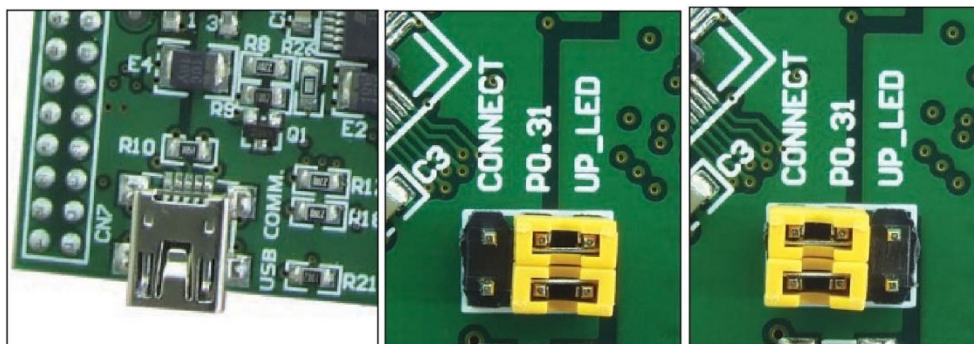


### 2.1.6 USB комуникација

USB прикључак CN7 дозвољава приступ USB уређајима микроконтролеру. Приликом повезивања, џампери J1 и J2 се користе за одређивање функција повезивања (UP\_LED или „мека конекција”). У случају када су џампер J1 и J2 постављени на UP\_LED позицију, LED диода на плочи означена са UP\_LED (LD1) сија само када се USB уређај конфигурише. Ако су J1 и J2 постављени



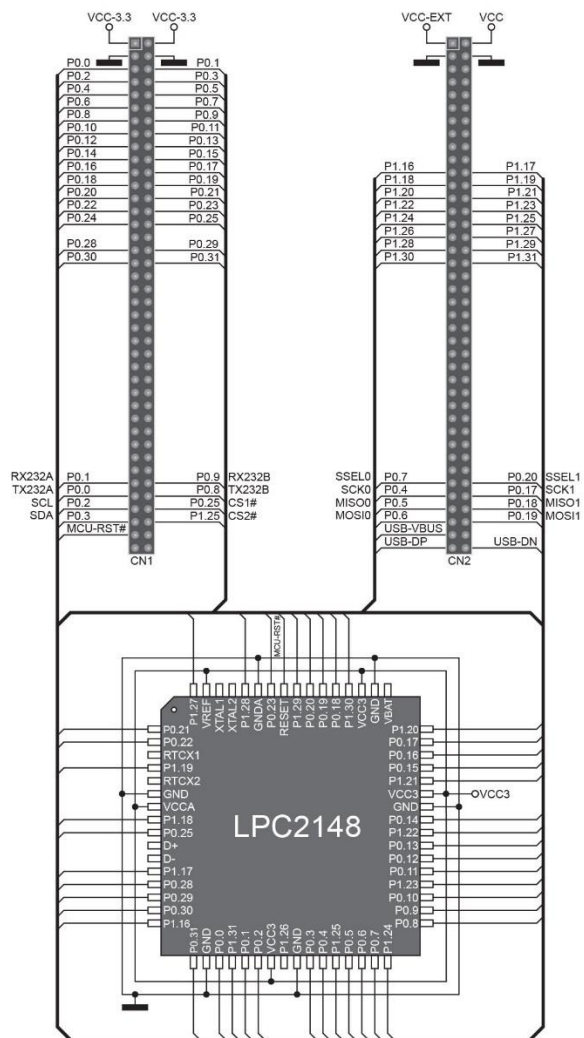
на CONNECT положај, спољни отпорници од  $1500\Omega$  се дају на коришћење софтверу, тако реализујући функцију „меког повезивања”.



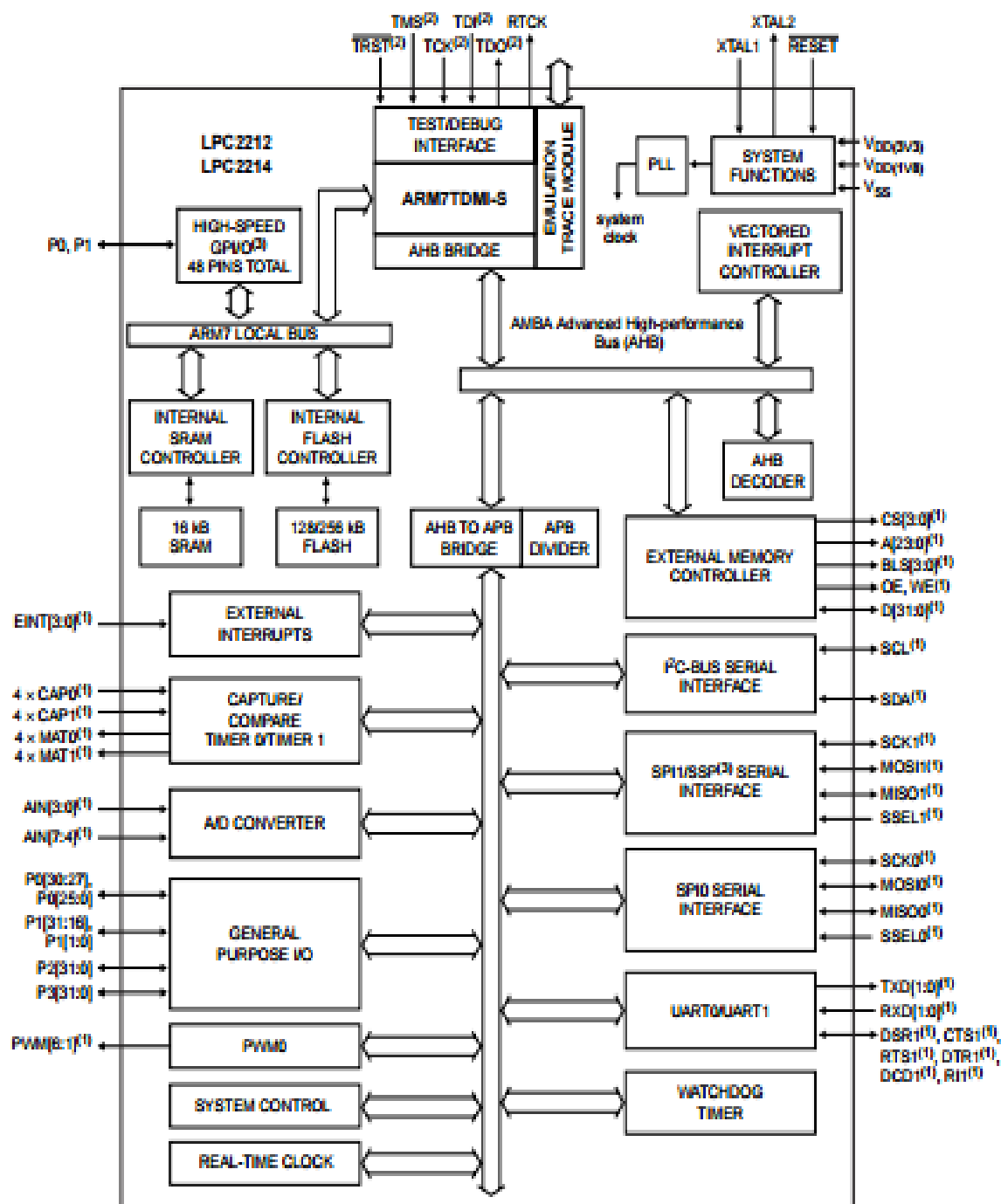
## 2.2 LPC2148

### 2.2.1 Општи опис и карактеристике

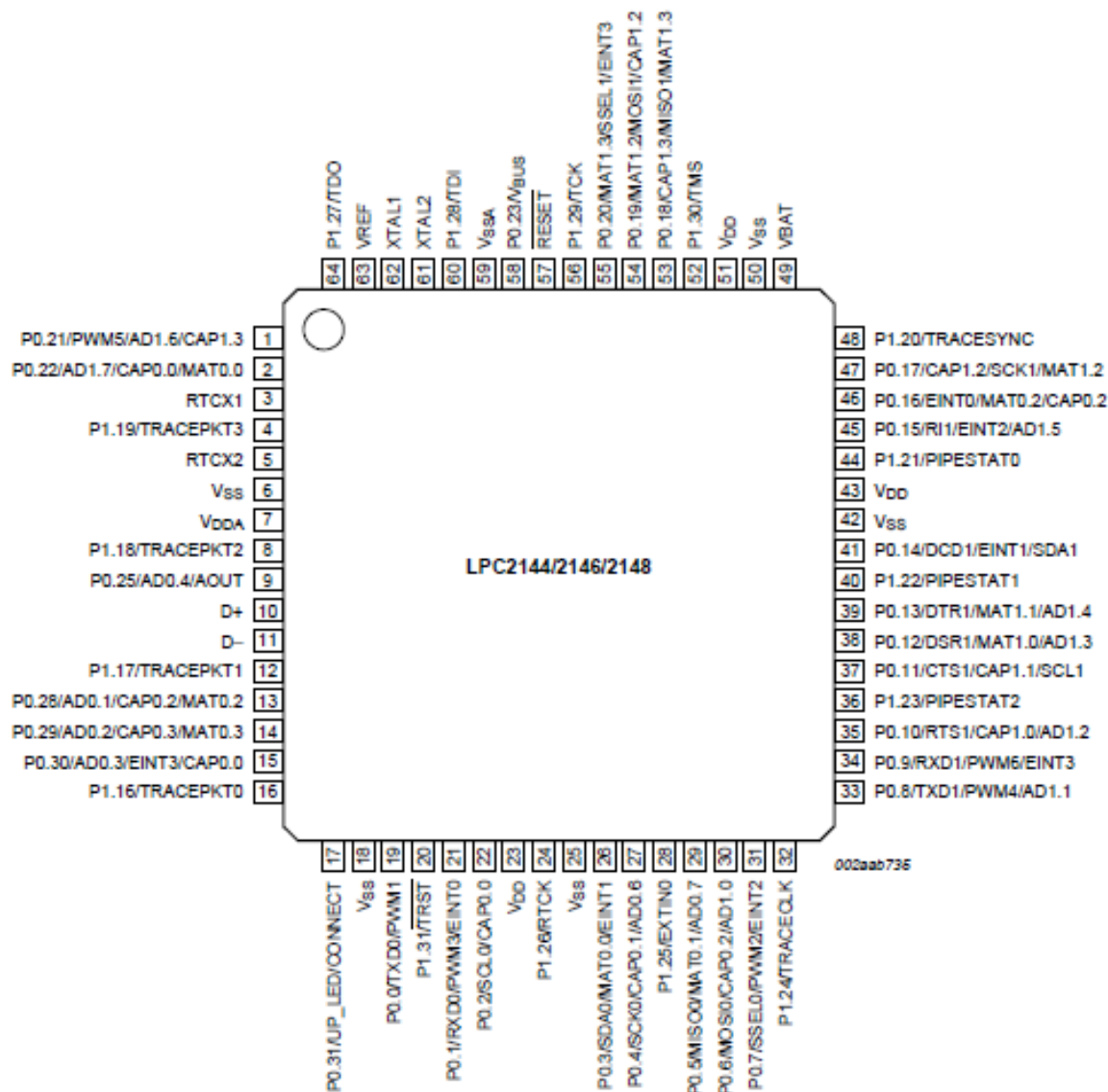
LPC2148 се заснива на 32-битном ARM7TDMI-S микроконтролеру у LQFP64 пакету. Има 40kB статичког RAM-а на чипу и 512kB меморије флеш програма на чипу. Интерфејс ширине 128 бита омогућава брзину од 60MHz за извршавање операција. Поседује и системско и апликативно програмирање преко boot-loader софтвера на чипу. Флеш програмирање траје 1ms за 256-битну линију. Брисање једног сектора или целог чипа траје 400ms. Контролер уређаја подржава USB са 2kB RAM-а. RTC (real time clock) има малу потрошњу, независно напајање и улаз за клик од 32KHz.



## 2.2.2 Блок дијаграм



### 2.2.3 Пинови



## 2.2.4 Функционални опис

### 2.2.4.1 Преглед архитектуре

ARM7RDMI-S је 32-битни микропроцесор опште намене, који нуди високу перформансу и малу потрошњу енергије. ARM архитектура је базирана на RISC (Reduced Instruction Set Computers) принципима, и сет инструкција и сродни механизми декодирања су много једноставнији него они на микропрограмираним CISC (Complex Instruction Set Computers) рачунарима. Ова једноставност резултује високим протоком инструкција и импресивним одзивом прекида у реалном времену од малог и исплативог процесорског језгра. Pipeline технике (техника која имплементира форму паралелности у раду процесора, која омогућава бржи CPU проток-број задатих инструкција које се извршавају у јединици времена), омогућавају да сви делови процесорских и меморијских система могу да раде континуирано. Типично, док се једна инструкција извршава, њен наследник се декодира, а трећа инструкција се узима из меморије.

ARM7RDMI-S процесор такође користи јединствену архитектонску стратегију познату као Thumb, која га чини идеално опремљеним за апликације великог обима са меморијским ограничењима, или апликације где је збијеност кода проблем. Кључна идеја Thumb-а је супер-редуковани сет инструкција. У суштини, ARM7RDMI-S процесор има два сета инструкција:

- Стандардни 32-битни ARM сет
- 16-битни Thumb сет

Инструкција дужине 16 бита Thumb-овог сета омогућава приступ двострукој збијености стандардног ARM кода док задржава већину ARM-ових предности у перформансама над традиционалним 16-битним процесором користећи 16-битне регистре. Ово је могуће зато што Thumb-ов код ради на истом 32-битном регистарском сету као и ARM код. Thumb-ов код може обезбедити до 65 % ARM-ове величине кода, и 160 % перформанси еквивалентног ARM процесора повезаног на 16-битни меморијски систем.

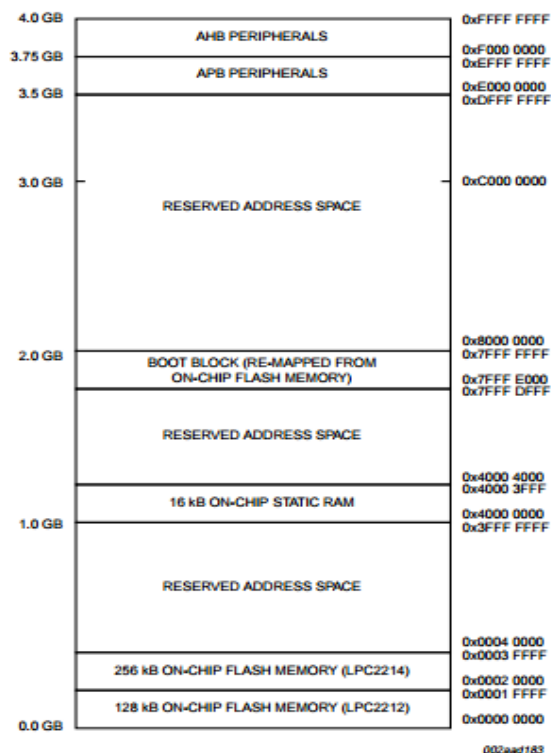
### 2.2.4.2 Интегрисана флеш програмабилна меморија

LPC2148 садржи 512kB систем флеш меморије. Ова меморија може да се користи за смештање и кодова и података. Програмирање флеш меморије може се обавити на неколико начина. Може бити програмирана In-System преко серијског порта. Апликативни програм такође може брисати и/или програмирати флеш док је апликација покренута, дозвољавајући висок степен флексибилности за смештање података надоградње фирмњаре-а, итд. Када се користи bootloader на чипу, 248kB флеш меморије је доступно за кориснички код. LPC2148 флеш меморија обезбеђује минимално 100000 бриши/пиши циклуса и 20 година задржавања података.

### 2.2.4.3 Интегрисани статички RAM

Интегрисани статички RAM се може користити за код и/или смештање података. LPC2148 обезбеђује 32kB статичког RAM-а.

### 2.2.4.4 Меморијска мапа



#### ***2.2.4.5 Прекидачки контролер***

Векторски прекидачки контролер (VIC – Vectored Interrupt Controller) прима све захтеве прекида и убацује и категоризује их као FIQ(Fast Interrupt reQuest), векторски IRQ (Interrupt Request) и не-векторски IRQ дефинисан програмабилним подешавањима. FIQ има највећи приоритет, затим векторски IRQ, а најмањи приоритет има не-векторски IRQ.

#### ***2.2.4.6 Блок повезивања пина***

Блок повезивања пина дозвољава селектованим пиновима микроконтролера да имају више од једне функције. Конфигурациони регистри контролишу мултиплексере да дозволе повезивање између пина и периферија на чипу. Периферије прво треба да буду повезане на одговарајуће пинове да би биле активирани, и пре него што је неки повезани прекид омогућен.

Активност било које дозвољене периферне функције која није повезана на одговарајући пин треба се сматрати недефинисаном. Пин контролни модул садржи 3 регистра.

#### ***2.2.4.7 Паралелни, опште намене и брзи U/I***

Пинови уређаја који нису повезани на одређену периферну функцију контролишу се паралелним U/I регистрама. Пинови могу бити динамички конфигурисани као улази или излази. Одвојени регистри дозвољавају подешавања или брисање било ког броја излаза истовремено. Вредност излазног регистра се може читати уназад, као и тренутно стање пинова порта. Сви U/I се постављају за улазе после ресетовања.

LPC2148 уводи убрзане GPIO функције у односу на претходне LPC2000 уређаје:

- GPIO регистри су пребачени на ARM магистралу због најбржег могућег I/O тајминга.



- 'Маскирани регистри' дозвољавају рад са одабраном групом битова са портова, остављајући тако остале битове непромењене.
- Сви GPIO регистри су адресабилни.
- Укупна вредност са портова може бити написана у једном упуству.

#### **2.2.4.8 10-битни ADC**

LPC2148 садржи две 10-битне узастопне апроксимације ADC. Укупан број слободних ADC улаза за LPC2148 је 14.

#### **2.2.4.9 UART-i**

LPC2148 садржи два UART-а. Поред стандардних преносних и пријемних линија података, UART1 такође омогућава потпуну контролу модема руковања интерфејсом.

#### **2.2.4.10 I<sup>2</sup>C магистрала**

I<sup>2</sup>C-магистрала је двосмерна магистрала за интер-IC контролу коришћењем само две жице: серијска clock линија (SCL), и серијска линија података (SDA). Сваки уређај се препознаје по јединственој адреси и може да ради и само као пријемник уређаја (нпр. LCD драјвер или предајник са способношћу да и прима и шаље информације (као меморија)). Предајници и/или пријемници могу да раде и у мастер или славе режиму, у зависности од тога да ли чип мора да покрене пренос података или је само адресиран. I<sup>2</sup>C-магистрала је мулти-мастер магистрала; може се контролисати више од једне мастер магистрале повезане са њом. I<sup>2</sup>C -магистрала имплементирана у LPC2148 подржава брзину до 400kbit/s (Брза I<sup>2</sup>C -магистрала).

#### **2.2.4.11 SPI (серијски периферни интерфејс)**

LPC2148 садржи један SPI контролер. SPI је потпуни двоструки серијски интерфејс, дизајниран да може да подржи више master-a и slave-a

конектованих на задату магистралу. Само један мастер и један славе могу да комуницирају на интерфејсу током преноса података. Током преноса података мастер увек шаље бајт података slave-у, а slave увек шаље бајт података master-у.

#### ***2.2.4.12 SSP контролер***

SSP је контролер способан да ради на SPI, 4-жичном SSI, или Microwire магистрали. Може да комуницира са више master-а и slave-а на магистрали. Само један master и један slave могу да комуницирају на магистрали приликом преноса података. Преноси података су потпуни дуплекс, са оквирима од 4 до 16 битова података који путују од master-а до slave-а и од slave-а до master-а. Како SSP и SPI1 периферије деле исте физичке пунове, није могуће да буду активни у исто време. Апликација може да пребаци у лету са SPI1 на SSP и назад.

#### ***2.2.4.13 Тајмери опште намене***

Тајмер/Бројач је дизајниран да броји циклусе периферног клока (PCLK) или клока обезбеђеног споља и опционо генерише прекиде или изводи друге радње у одређеним вредностима тајмера, у зависности од четири match регистра. Такође садржи четири улаза за хватање да улови вредност тајмера када се мења улазни сигнал, опционо генерише прекид. Више пинова се може одабрати да изводе једно хватање или функцију упаривања, обезбеђујући апликацију са 'или' и 'и', као и функцију 'broadcast' међу њима.

#### ***2.2.4.14 Watchdog тајмер***

Намена овог тајмера је да ресетује микроконтролер у одређеном временском интервалу ако уђе у погрешно стање. Када је укључен, тајмер ће генерисати ресетовање система ако кориснички систем падне на читавање у оквиру одређеног времена.

#### ***2.2.4.15 Clock у реалном времену***

Clock је дизајниран да обезбеди скуп бројача за мерење времена када је одабран нормалан начин рада, или начин рада мировања. Направљен је да користи мало енергије што га чини погодним за уређаје који се пуне уз помоћ батерије где процесор не ради константно.

#### ***2.2.4.16 Ширина импулса модулятора***

Модулятор је базиран на стандардном блок тајмеру и наслеђује све његове карактеристике, мада је само функција модулятора постављена на излаз LPC2148. Тајмер је дизајниран да броји циклусе периферног clock-а и опционо генерише прекиде или извршава друге радње када се одређене вредности на тајмеру јаве, базиран на слагању седам регистара. Функција модулятора је такође базирана на слагању регистара. Способност да одвојено контролише позитивну и негативну ивицу дозвољава модулятору да буде коришћен за више апликација. На пример, више фазни мотор типично обухвата три излаза модулятора која се не преклапају са индивидуалном контролом сва три импулса ширине и позиције. Два регистра која се поклапају, могу да се користе да се обезбеди контролисање једене ивице излаза модулятора. Један регистар контролише брзину циклуса, док други регистар контролише позицију ивице модулятора. Додатна ивица контролише излазе. Три регистра која се поклапају, могу се користити да обезбеде контролисање излаза модулятора са две ивице. Са контролисањем дупле ивице излаза модулатоа, специфични регистри контролишу узлазне и силазне ивице код излаза.

### **2.2.5 Контрола система**

#### ***2.2.5.1 Кристални осцилатор***

Осцилатор подржава кристале у рангу од 1 Mhz до 30 Mhz. Излазна фреквенција осцилатора се назива fosc, а фреквенција clock-а процесора је прослеђена као CCLK за потребе једначина стопа, нпр Fosc и CCLK су једнаки осим ако је PLL покренут и повезан.

### **2.2.5.2 PLL**

PLL прихвата улазну фреквенцију clock-а у рангу од 10 Mhz до 25 Mhz. Улазна фреквенција се множи у распону од 10 Mhz до 60 Mhz са струјно-контролисаним осцилатором (ССО - тренутно контролисани осцилатор). Множилац може бити целобројна вредност од 1 до 32 (у пракси, множилац не може бити већи од 6 за ову породицу микроконтролера због горње границе фреквенције процесора). ССО ради у распону од 156 Mhz до 320 Mhz, тако да постоји додатни делилац у петљи да задржи ССО у оквиру његовог фреквентног опсега док PLL обезбеђује жељену излазну фреквенцију. Излазни делилац може бити постављен да дели са 2, 4, 8 или 16 да произведе излазни clock. обезbeđuje željenu izlaznu frekvenciju. Izlazni delilac može biti postavljen da deli sa 2, 4, 8 ili 16 da proizvede izlazni clock.

### **2.2.5.3 Тајмер ресетовања и буђења**

Ресетовање има два извора на LPC2148: RESET pin и Watchdog Reset. RESET pin је Schmitt trigger улазни пин са додатним филтером грешке. Тврдња ресетовања чипа са било којим извором покреће тајмер буђења, узизивајући да унутрашње ресетовање чипа задржи тврдњу док спољашње ресетовање је напуштено. Осцилатор је покренут, фиксиран борј clock-ова је прошао, и контролер флеша на чипу је завршио своју иницијализацију. Када је унутрашње ресетовање уклоњено, процесор почиње извршавање на адреси 0, која је вектор ресетовања. У том тренутку, сви процесорски и периферни регистри се иницијализују на преодређене вредности. Тајмер за буђење осигурава да су псцилатор и друге аналогне функције потребне за операције чипа у потпуности функционалне, пре него што је процесору дозвољено да изврши инструкције.

#### ***2.2.5.4 Brownout детектор***

LPC2148 укључује 2 фазно надгледање напона на VDD пиновима. Ако овај напон падне испод 2,9 V, BOD потврђује сигнал прекида на VIC-у. Овај сигнал може бити омогућен за прекид, или софтвер може надгледати сигнал уз помоћ читања регистра. Друга фаза ниског напона детекције потврђује ресетовање LPC2148 када напон падне на пиновима VDD испод 2,6 V.

#### ***2.2.5.5 Заштита кода***

Ова карактеристика LPC2148 дозвољава апликацији да контролише да ли сме доћи до отклањања грешака или до заштите од посматрања.

#### ***2.2.5.6 Спољашњи прекидни улази***

LPC2148 укључује до 9 ивица или спољашњих прекидних излаза осетљивих на ниво као пин функције по избору. Када су пинови комбиновани, спољашњи догађаји могу бити обрађени као четри независна прекидна сигнала. Спољашњи прекидни излази могу опционо да се користе да буде процесор из режима гашења.

#### ***2.2.5.7 Контрола мапирања меморије***

Контрола мапирања меморије мења мапирање прекида вектора који се појављују на почетку на адреси 0x0000 0000. Вектори морају бити мапирани на дну чипа flash меморије, или на чипу статичке RAM меморије. Ово омогућава коду који ради на више различитих меморијских локација, да има контролу на прекидима.

#### ***2.2.5.8 Контрола снаге***

LPC2148 подржава 2 мода редуковане снаге: Idle мод и Power-down мод. У пасивном режиму рада, извршење инструкција је суспендовано док се не појави ресет или прекид. Периферне функције настављају рад током пасивног режима и могу генерисати прекиде тако да изазову процесор да настави извршење. Idle мод елиминише моћ која је коришћена од стране самог процесора, меморијских система и сродних контролера, и унутрашњих басева. У Power-down моду, осцилатор је угашен и чип не прима унутрашње сатове. Стање процесора и регистара, периферних регистара, и унутрашњих SRAM вредности су очуване кроз Power-down мод и логички нивои чипа излазних пинова остају статички. Режим Power-down може да се заврши и да се нормалнео функционисање настави уз ресетовање или одређених специфичних прекида који су у стању да функционишу без клокова. Пошто су све динамичке операције чипа суспендоване, Power-down режим смањује потрошњу снаге чипа на скоро нулу.

### ***2.2.5.9 APB бас***

APB делилац дефинише везу између clock-а процесора (CCLK) и clock-а коришћеног од стране периферних уређаја (PCLK). APB делилац служи у 2 сврхе. Прва је да обезбеди периферије са жељеним PCLK помоћу APB баса тако да могу да врше опрације при брзини одабраној од стране ARM процесора. Како би се ово постигло, APB бас може бити успорен на  $\frac{1}{2}$  до  $\frac{1}{4}$  такта clock-а процесора. Зато што APB бас мора радити одговарајуће при покретању (и његов тајминг се не може мењати ако не ради пошто се APB делилац контролних регистара налази на APV басу), уобичајено стање је при ресету за APB бас да ради на брзину  $\frac{1}{4}$  такта процесора clock-а. Друга сврха APB делиоца је да дозволи чување снаге када нека апликација не захтева периферије да се користи при виским тактовима процесора. Зато што је APB делилац конектован на PLL излаз, PLL остаје активан (ако је радио) током Idle мода.

### ***2.2.5.10 Емулација и отклањање грешака***

LPC2148 подржава емулацију и отклањање грешака помоћу JTAG серијског порта. Траг порта дозвољава праћење извршавања програма. Отклањање грешака и функције праћења су мултиплексирани само са GPIO-има на Port-у 1. Ово значи да се све комуникације, тајмер и интерферјс периферије налазе на Port-у 0 и слободне су током развоја фазе отклањања грешака.

#### ***2.2.5.11 Уграђена ICE логика***

Стандардна ARM уграђена ICE логика омогућује подршку отклањања грешака на чипу. Отклањање грешака одређеног система захтева да рачунар на којем се ради отклањање грешака покреће софтвер за отклањање грешака и Embedded ICE протокол конвертовања. Embedded ICE протокол конвертовања конвертује удаљено дебаговање помоћу протокола команди у JTAG податке потребне да се приступи ARM језгру.

#### ***2.2.5.12 RealMonitor***

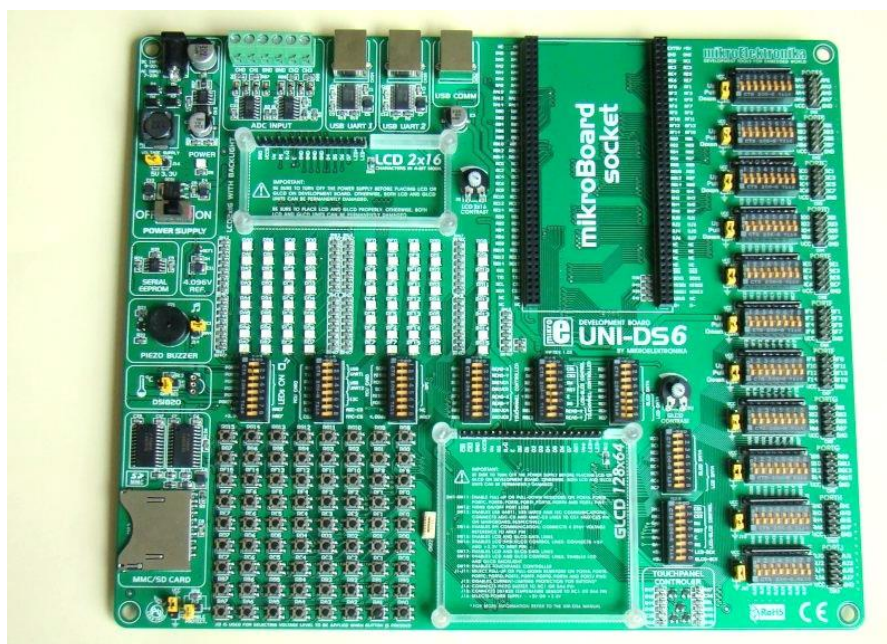
RealMonitor је подесив софтверски модул, развијен од стране ARM Inc., који омогућава дебаговање у реалном времену. То је лагани debug monitor који се покреће у позадини док корисник дебагује предњу апликацију. Комуницира са домаћином користећи DCC ( Debug Communications Channel), који је присутан у EmbeddedICE логици. LPC2148 садржи специфичну конфигурацију RealMonitor софтвера програмираног на flash меморији, на чипу.

## **2.3 UNI-DS6**

Развојни систем UNI-DS6 пружа развојно окружење за програмирање и експериментисање са разним микроконтролерима различитих произвођача. Бројни модули, као што су 128x64 графички LCD дисплеј, 2x16



алфанумерички LCD дисплеј, пиезоелектрична зујалица, USB-UART итд. се налазе на плочи и омогућавају лако симулирање рада циљног уређаја.



Како би се омогућило да развојни систем буде укључен, неопходно је обезбедити одговарајући напон напајања преко AC/DC конектора CN19. Када се развојни систем напаја, потребно је поставити прекидач обележен са POWER SUPPLY на ON позицију.

МикроBoard је дизајниран за постављање микроконтролера на развојни систем. Сваки mikroBoard има интегрисани програматор који се користи за MCU програмирање. За повезивање са развојним системом, mikroBoard користи два 2x40 мушка заглавља. Поред тога, mikroBoard се може користити као самосталан уређај.

UNI-DS6 развојни систем је дизајниран за коришћење са бројним моделима mikroBoard плоча. Сви mikroBoard-и се постављају на универзални mikroBoard прикључак (mikroBoard socket).

MicroBoard на развојном систему користи уграђен програматор за MCU програмирање. Све што треба урадити је повезати microBoard са рачунаром преко USB кабла и инсталирати одговарајући софтвер на рачунару.

UNI-DS6 развојни систем може да комуницира са спољашњим уређајима преко USB конектора коришћењем USB комуникације. USB конектор је директно повезан на пинове микроконтролера коришћењем USB комуникације.

ADC модул се користи за конвертовање аналогног нивоа напона у одговарајућу 12-битну дигиталну вредност. Аналогни напон се добија преко screw терминала CN15 и CN16. Напон испоручен преко VREF пина се користи као референтни напон. Да би се овај напон користио, прекидач 8 на DIP прекидачу SW14 треба да буде постављен на ON позицију.

UNI-DS развојни систем може да чита меморијске картице јер има MMC/SD прикључак. Меморијска картица комуницира са микроконтролером преко пинова микроконтролера који се користе за серијску комуникацију. Да би се успоставила комуникација између меморијске картице и микроконтролера, неопходно је да прекидачи 1, 2, 3 на DIP прекидачу SW14, као и прекидач 8 на DIP прекидачу SW13 буду постављени на ON позицију.

UNI-DS има 72 LED-а који се користе за визуелно указивање на стање сваког У/И пина микроконтролера. Активан LED указује да је на пину активна логичка јединица.

Стање улазних пинова микроконтролера се може променити коришћењем тастера. Притиском на било који тастер када је jumper J13 у положају VCC-

BRD, логичка јединица ће бити примењена на одговарајућем пину микроконтролера.

UNI-DS6 развојни систем поседује конектор за алфанумерички 2x16 LCD дисплеј. Овај конектор је повезан са микроконтролером преко DIP прекидача. Да би се омогућио рад 2x16 ЛЦД дисплеја, потребно је написати програм који дефинише који MCU пинови ће бити коришћени за комуникацију између 2x16 LCD дисплеја и MCU-а.

128x64 графички LCD (GLCD) је повезан са микроконтролером преко DIP прекидача. Има резолуцију екрана 128x64 пиксела, што омогућава да дијаграми, табеле и остали графички садржаји буду приказани. Да би се омогућио 128x64 графички LCD дисплеј, потребно је написати програм који дефинише који MCU пинови ће бити коришћени за комуникацију између 128x64 графичког LCD дисплеја и MCU-а.

Панел осетљив на додир је танак, самолепљив и транспарентан. Поставља се преко GLCD-а. Његова главна функција је да региструје притисак у одређеној тачки дисплеја и проследи координате у виду аналогног напона до микроконтролера.

Дуж десне стране развојног система постоји једанаест 10-пинских конектора повезаних на У/И портове микроконтролера. Pull-up или pull-down отпорници могу бити повезани на У/И портове преко jumper-а J1-J11 и DIP прекидача SW1-SW11.

### 3. Пројектни задатак

У овом пројектном задатку је приказано паљење и гашење диода на портовима 0 и 1.

Програмирање микроконтролера са bootloader-ом користећи програм Flash Magic, корак по корак:

#### 1. Повезивање плоче са рачунаром

Један крај кабла се убаци у било који од слободних USB портова на рачунару, а други крај у CN3 порт на картици.



#### 2. Покретање програма Flash Magic

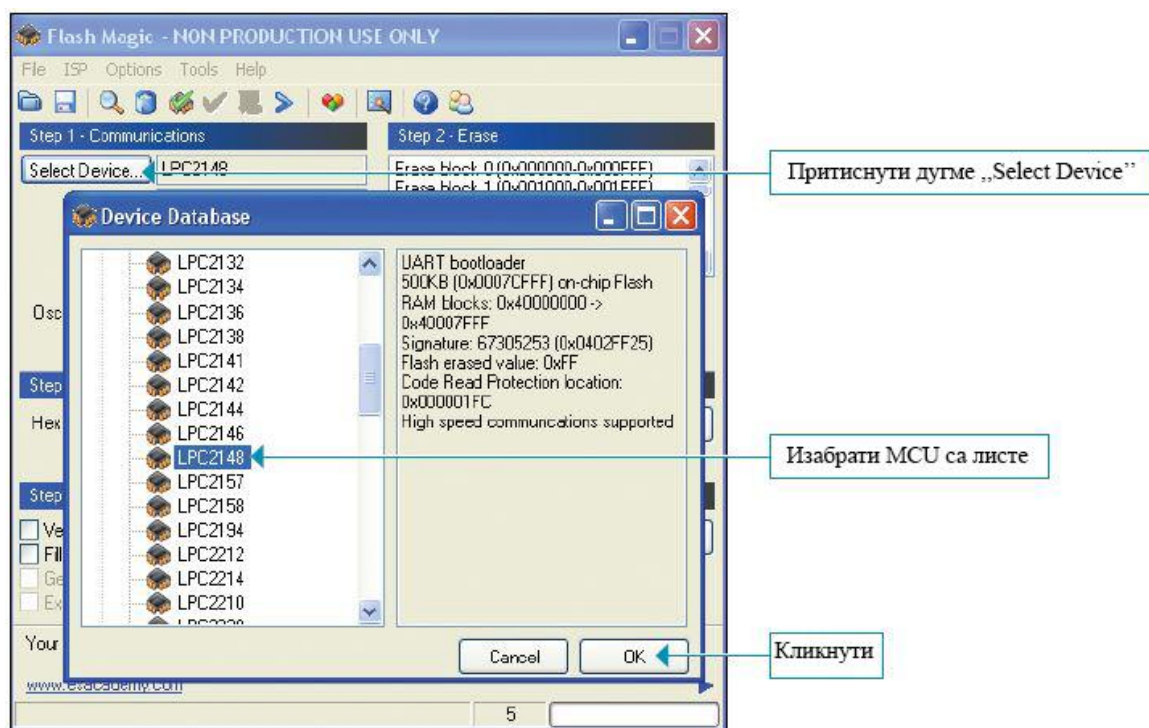
Програм Flash Magic се преузима са везе:

<http://www.flashmagictool.com/download.html&d=FlashMagic.exe> и покреће га.

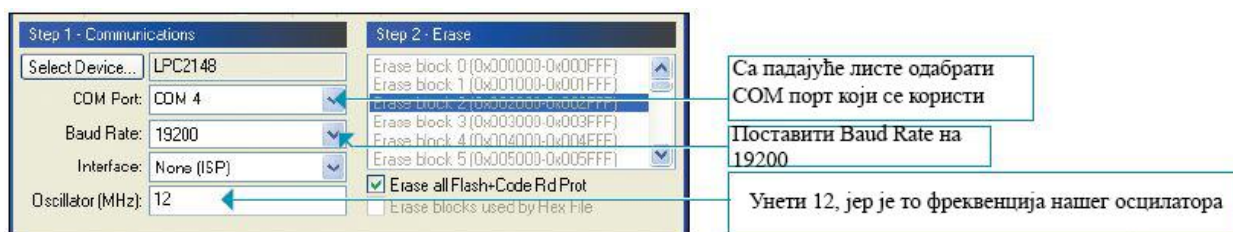
Након инсталације се покреће.



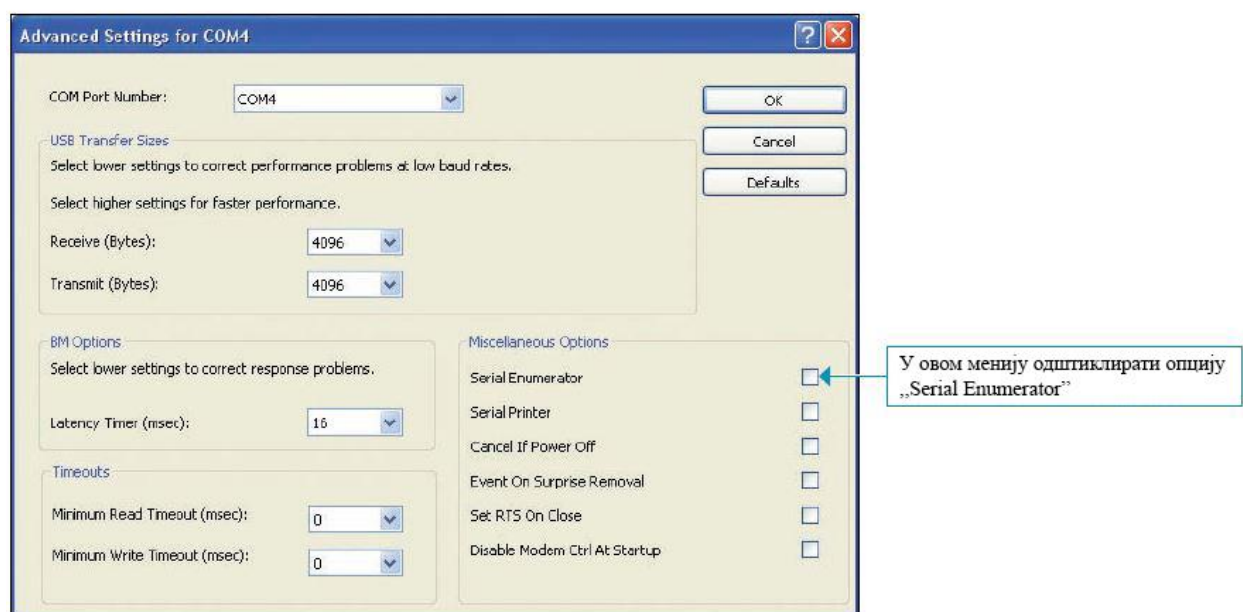
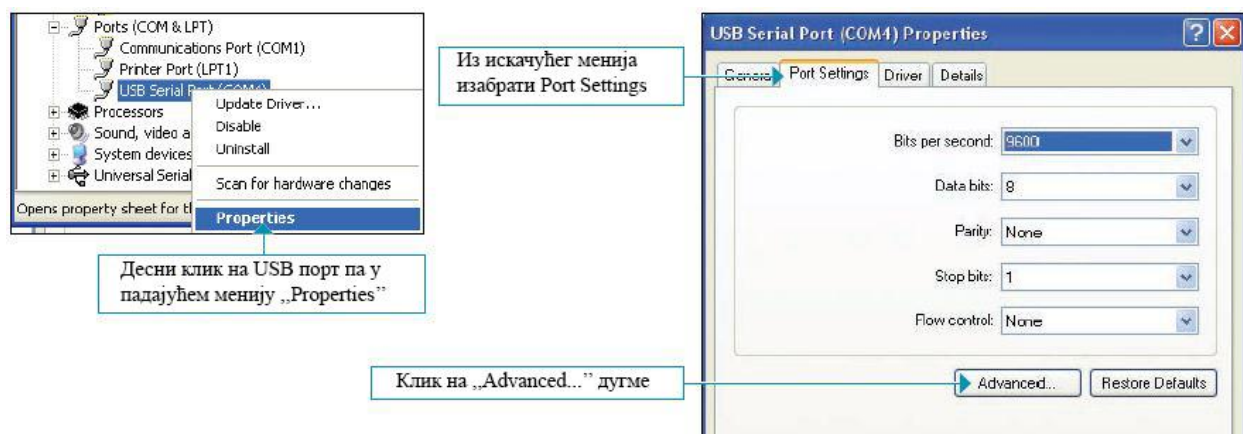
#### 3. Избор микроконтролера



#### 4. Подешавања

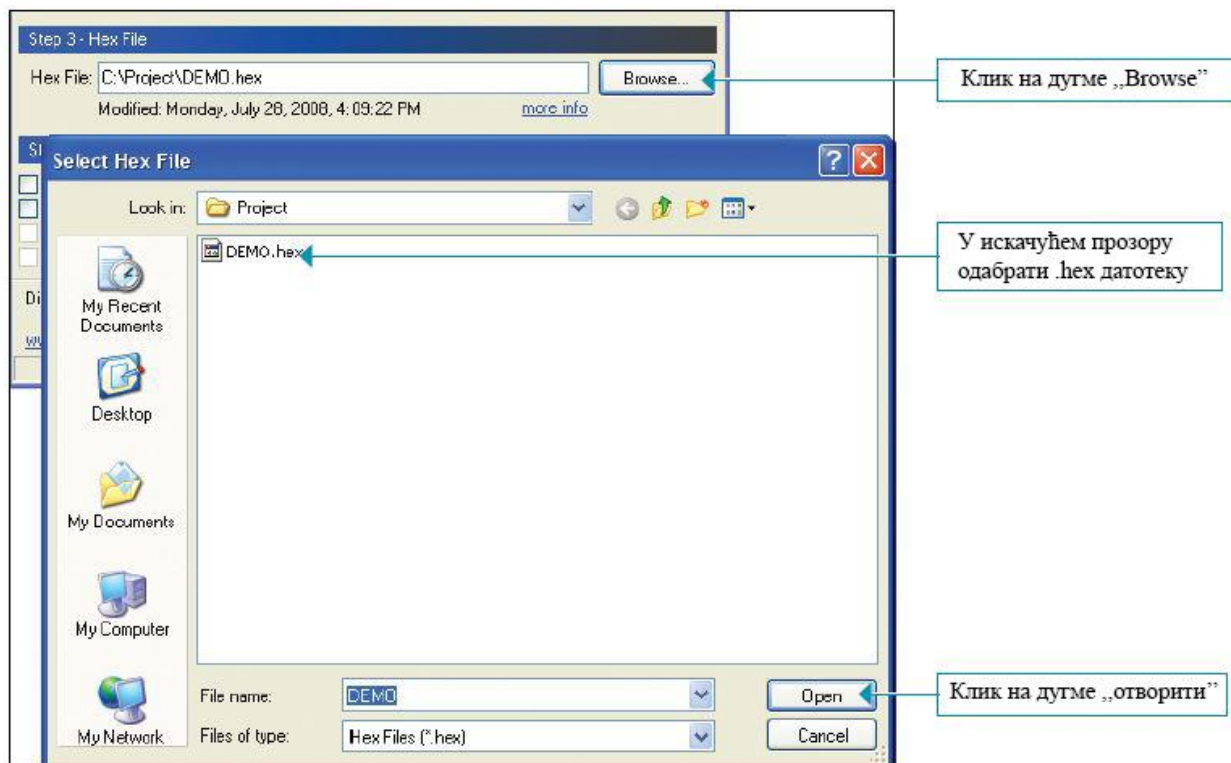


Помоћу Device Manager-а се може видети који се COM порт користи за комуникацију са плочом.

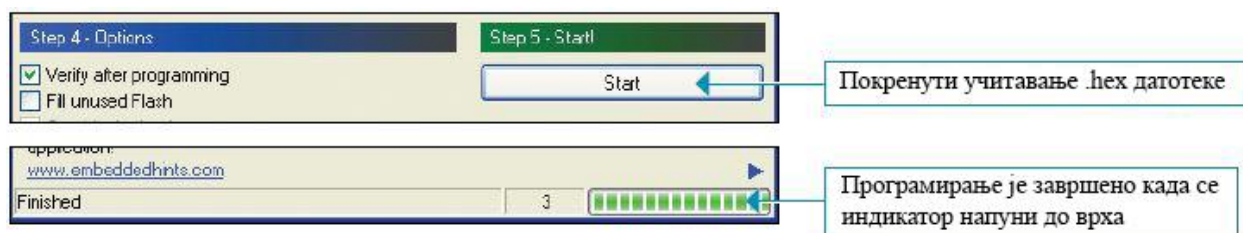




## 5. Бирање .hex датотеке



## 6. Учитавање .hex датотеке





## 4. Закључак

Овај пројекат за улогу има да упозори или обавести када је дошло до опасности, када се догодио одређени квар, приликом провала, или да сигнализира да је неки задатак извршен.

## 5. Литература

-Ауторизован превод са енглеског језика седмог издања књиге *Computer Organization and Architecture Designing for Performance Seventh Edition*, William Stallings

- <http://download.mikroe.com/documents/full-featured-boards/easy/easyarm-v6/mikroboard-arm64-manual-v100.pdf>

- <http://download.mikroe.com/documents/full-featured-boards/universal/unids-v6/unids-v6-manual-v100.pdf>

## 6. Прилог кодови

```
#include <LPC214X.H>
```

```
#include "Utility.h"
```

```
int main (void)
```

```
{
```

```
    PINSEL0 = 0;
```

```
    PINSEL1 = 0;
```

```
    PINSEL2 &= 0x0000000C;
```

```
    PINSEL2 |= 0x00000030;
```

```
    DelayProc(0.2 * CCLOCK);
```

```
IODIR0 = 0xFFFFFFFF;    // PORT0[31..0] defined as Output
IODIR1 = 0xFFFFFFFF;    // PORT1[31..16] defined as Output

while (1)
{
    IOSET0 = 0xFFFFFFFF;
    IOSET1 = 0xFFFFFFFF;
    DelayProc(0.2 * CCLOCK);

    IOCLR0 = 0xFFFFFFFF;
    IOCLR1 = 0xFFFFFFFF;
    DelayProc(0.2 * CCLOCK);
}
```