

СТШ “Никола Тесла”  
Сремска Митровица

# МАТУРСКИ РАД

Предмет: Рачунари у системима  
автоматског управљања

Рад: База Рону програматора

Професор:  
Ђоко Крсмановић

Ученик:  
Далибор Сухањи IVe1

## I Увод

### 1.1 Историјски Развој Рачунара

Рачунање је за човека постала веома важна област са развојем трговине и размене добара. Прво су се развијале разне врсте меморија. Једна од најстаријих меморија је КИПУ, меморија која се састојала од једног хоризонталног канапа, о који су се качили други канапи. Био је веома распрострањен и Палестини, где је и измишљен, а касније се проширио и на друге делове Азије. На ужадима се правили чворићи, на једнаким растојањима од главног ужета. Облик чвора је претстављао цифру (1,2,3...9), док је растојање од главног ужета претстављало бројну вредност(јединица,десетица,хиљада...).

Око 4000-те године пре нове ере је направљено једно од првих помагала у рачунању Абакус. Сличан је данашњој рачунаљци. Прву машину за рачунање је 1623. направио Вилхелм Шикард из Гибингена у Немачкој. Ова машина је била веома примитивна и гломазна, те није нашла неку веома велику примену. Творцем прве рачунске машине, која је могла рачунати и одузимати се сматра Блез Паскал. Ова машина је прављена око 3 године, и по завршетку је добила име “Паскалина”, по њеном творцу. Била је веома компликована за рад и никада није нашла своју примену, јер су је и људи који су били обучени за рад са њом, а било их је мало, сматрали ову машину непријатељском за човека и веома компликованом за рад.

Пошто су машине за рачунање могле само сабирати и одузимати, 1594. године Шкотланђанин Непер је направио алгоритме помоћу којих се операције множења и дељења своде на сабирање и одузимање.

1820. године Чарлс Хавијер Томас је направио први успешан калкулатор који је могао да сабира, одузима, множи и дели. Овај модел је био широко прихваћен као шаблон за прављење свих од тог времена новијих калкулатора и од 1890. године је био толико усавршен, да је могао да памти неколико последњих резултата, као и да их штампа.

Око 1890. године Вилијем Бароуз је издао неколико рачунарских машина које су се користиле у банкама и осталим областима све до око 1970 године, када се појављује масовна производња електронских калкулатора.

1822. године је направљена диференцијала машина која је била аутоматска и програмски контролисана, а затим је исти аутор направио и аналитичку машину. При производњи ове машине је помагала Ада Бајрон, која је правила и програме за ову машину, те се сматра и првим програмером.

Први електронски дигитални рачунар се звао ABC којег је 1939. године, пројектовао Атанасов. Ова машина никада није комплетирана, а пројекат је напуштен 1942. године.

За време II светског рата је направљен рачунар “ENIAC”. Могао је да врши задатке електронском брзином, а програмиран је укључивањем и искључивањем каблова. Програмиран је и по неколико дана, а кварио се у просеку сваких 7 минута. Као улаз и излаз је користио бушене картице. Био је велик 10 x 20 метара, тежио је око 30 тона, имао је 70000 отпорника...

Упоредо са развојем овог рачунара развијан је нови рачунар који је имао неколико десетина пута мање делова и знатно мање димензије. Код овог рачунара је први пут уведен појам унутрашње меморије тј. програми се нису чували на картицама, већ су се чували у унутрашњој меморији, чиме се знатно убрзao процес уноса и исчитавања података .

У циљу смањења величине рачунара и појавом полупроводничке технологије фирмa “Intel” је дошла на идеју да главни мозак рачунара спакује на једну малу плочицу од силицијума или неког другог полупроводничког материјала. У почетку је број компонената који су моглистати на плочицу био око пар стотина елемената, а до данас се повећало на више стотина милиона елемената, највише транзистора. Те плочице (чипови у даљем тексту) су се звали, а и зову се микропроцесори. Први микропроцесор је био 4004, четворобитни микропроцесор. Затим је производња микропроцесора комерцијализована, и после неколико година су се појавиле и друге фирме које су се бавиле производњом микропроцесора.

Рачунари се деле на генерацијекојих до данас има 4 :

1. Рачунари прве генерације (Каррактеристика ових рачунара је била да су прављене од вакуумских цеви као активних елемената. Меморије су углавном биле магнетне траке, имали су веома велику потрошњу, и развијали су веома велику топлоту)

2. Рачунари друге генерације (имали су око 10000 транзистора, и производили су се између 1948. и 1959. године. Код ових рачунара се појављују први програмски језици: Cobol, Fortran, Algol, Lisp ...)

3. Рачунари треће генерације (До појаве ове генерације доводи појава чипова. Појављују се први персонални рачунари. Имали су око пола милиона активних елемената и уместо магнетних трака почињу се користити магнетни дискови).

4. Четврта генерација рачунара се рачуна од појаве IBM компатибилних рачунара и траје све до данас.

## 1.2 Рачунар

По најопштијој подели рачунар се може разложити на неколико делова:

1. Улазне јединице
2. Излазне јединице
3. Централна јединица
4. Меморије

Улазне јединице су уређаји помоћу којих се у рачунар уносе подаци. То су тастатура, миш, скенер, камера, микрофон... Помоћу улазних јединица ми рачунару уносимо команде, податке које нам рачунар захтева...

Излазне јединице су јединице помоћу којих рачунар комуницира са корисником, тј преко излазних јединица рачунар на излазне јединице шаље резултате, пита кориснике за даље инструкције... Неке од излазних јединица су: монитор, штампач, плотер, звучници...

Меморија је део рачунара у којима рачунар складиши податке, међу резултате и резултате рада рачунара. Постоји много врста меморија а неке од њих су : RAM, ROM, EEPROM, CD-ROM, хард диск, флопи диск(дискета),...

Централна јединица је главни део рачунара. Задужење ове јединице је да врши све аритметичке и логичке операције које ми задајемо рачунару. Она контролише све параметре рачунара, додељује ресурсе рачунара одређеној апликацији, контролише проток података...

Спома гледано, рачунар се обавезно сатоји из 4 дела : миш, тастатура, монитор и кућиште. Миш и тастатура су улазне јединице рачунара. Миш претвара померај руке у померај курсора на екрану, и притиском на одређене тастере који се налазе на мишу рачунар врши одређену акцију. Преко тастатуре ми у рачунар уносимо податке као што су слова, бројеви. Притиском одређене комбинације тастера рачунар врши одређену, унапред дефинисану функцију. Монитор је излазна јединица рачунара. Преко монитора нас рачунар извештава о резултатима послова које је рачунар за нас извршио и поставља нам питања о неким стварима у току његовог рада. У кућишту рачунара се налази централна јединица, слотови за експанзију функција рачунара, уређаји за контролу кретања података. Поред ових јединица као стандардни делови се све више убрајају и звучници.

Све видљиве компоненте рачунара се зову хардвер, док се сви подаци и програми у рачунара зову софтвер. Ова подела је веома груба, јер хардвер без софтвера не може да ради, као ни софтвер без хардвера и не могу се одвојити један од другог. Сви подаци у рачунару су у облику јединица и нула тј. у бинарном облику због лакше техничке реализације рачунара. Једна јединица или нула се зове бит. Осам бита се зове байт(By). $1024 \text{ By} = 1\text{kBy}$ ,  $1024\text{kBy} = 1\text{MBy}$ ...

Постоје системски и апликативни софтвер. Системски софтвер је софтвер који управља радом целокупног рачунарског система. То је најчешће оперативни систем рачунара.

Оперативни систем је главни програм који управља радом рачунара и расподељује ресурсе рачунара. Апликативни софтвер су сви подаци који се налазе у рачунару. То су програми, антивирусни софтвер, игрице, документа и фајлови...

### 1.3 Примена Рачунара

Данас се рачунари могу наћи скоро у свакој кући, и користе се скоро у свакој области. У почетку је рачунар првенствено коришћен као помагало у рачунању , али је развојем технике уопште и софтвера примена рачунара нагло расла тј. ширила се. Увођењем миша и GUI (Graphic User Interface, слика 1.), употреба рачунара се повећала поред обраде табела, звука, текста, научних прорачуна и на обраду слика, техничких цртежа, а са развојем софтвера и на разне анимације у техници производње анимираних филмова, репарацији застарелих слика, симулацији рада електротехничких кола ...



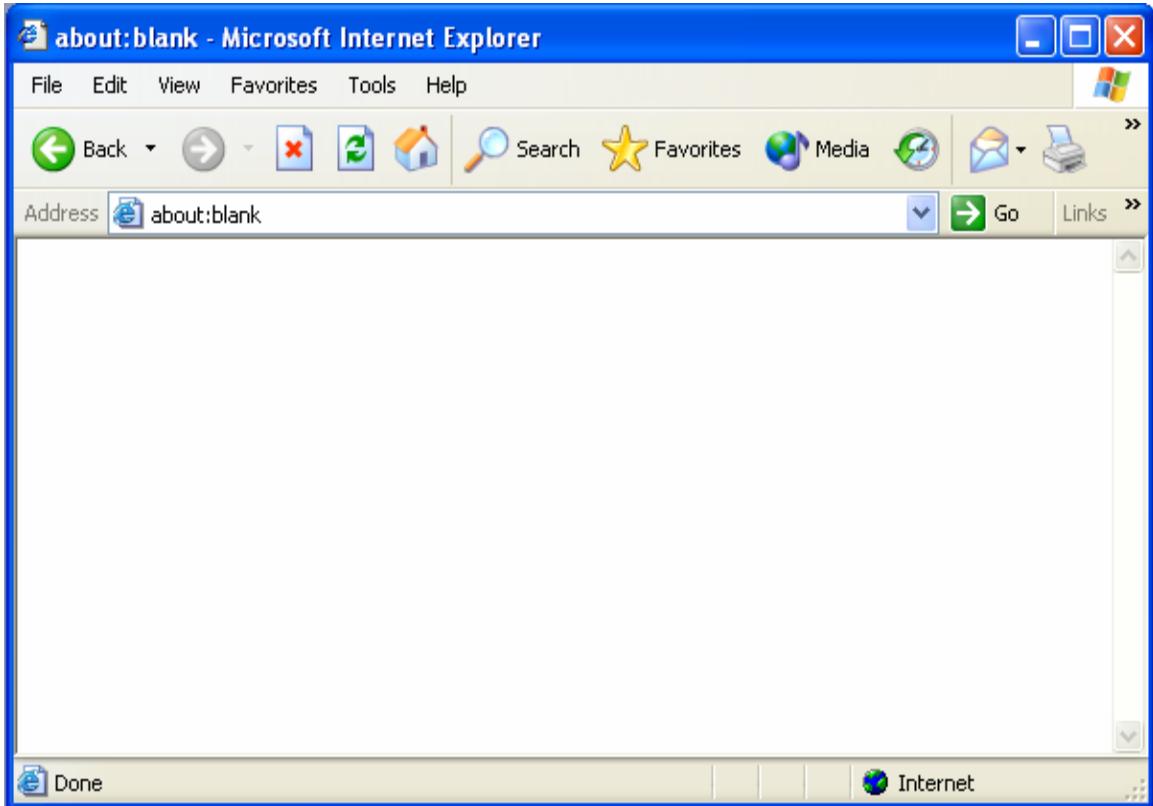
Слика 1. Пример GUI

Неки од програма који се данас користе су:

1. MS WORD је програм који је саставни део програмског пакета MICROSOFT OFFICE и користи се за обраду текста,
2. MS EXCEL је програм који је саставни део горе наведеног програмског пакета и користи се првенствено за обраду табела,
3. ADOBE PHOTOSHOP, AUTO CAD, COREL DRAW, PAINT... су програми који се користе за едитовање и обраду слика,
4. WINDOWS MEDIA PLAYER, WINAMP,... су програми који се користе за обраду аудио и једноставнијих видео података,
5. INTERNET EXPLORER, MOZILA, OPERA .... су програми који се користе за прегледање интернета, што је један од данас најважнијих примена рачунара.

Данас се рачунарски програми најчешће испоручују корисницима у виду програмских пакета-скупа сродних међусобно независних програма, који сви заједно чине једну целину. Један од најчешће коришћених програмских пакета је MICROSOFT OFFICE програмски пакет, који се данас користи за прављење презентација, прављење табела, едитовање текста, прављење интернет страница... Постоји више врста овог пакета, па садржај програма зависи од произвођача, ако су у питању пиратске врсте, до генерације програмског пакета, јер сваки програмски пакет има неких додатака.

Још једна примена рачунара, без чега је готово немогуће замислити рад са рачунаром, је интернет-виртуелна мрежа која повезује све кориснике који су повезани на ову мрежу и служи као нека врста огласне табле, где корисници интернета могу наћи одговор на скоро сва питања која их интересују.



Слика 2. Прозор програма,који се користи за претрагу интернета

Важна улога рачунара је и мултимедија. То је веома велика област рачунарства и у то спадају неке од следећих категорија: издаваштво,географске мапе, ...

Ово су само неке до примена рачунара, а са развојем технике ове границе све више и више достижу појам бесконачности.

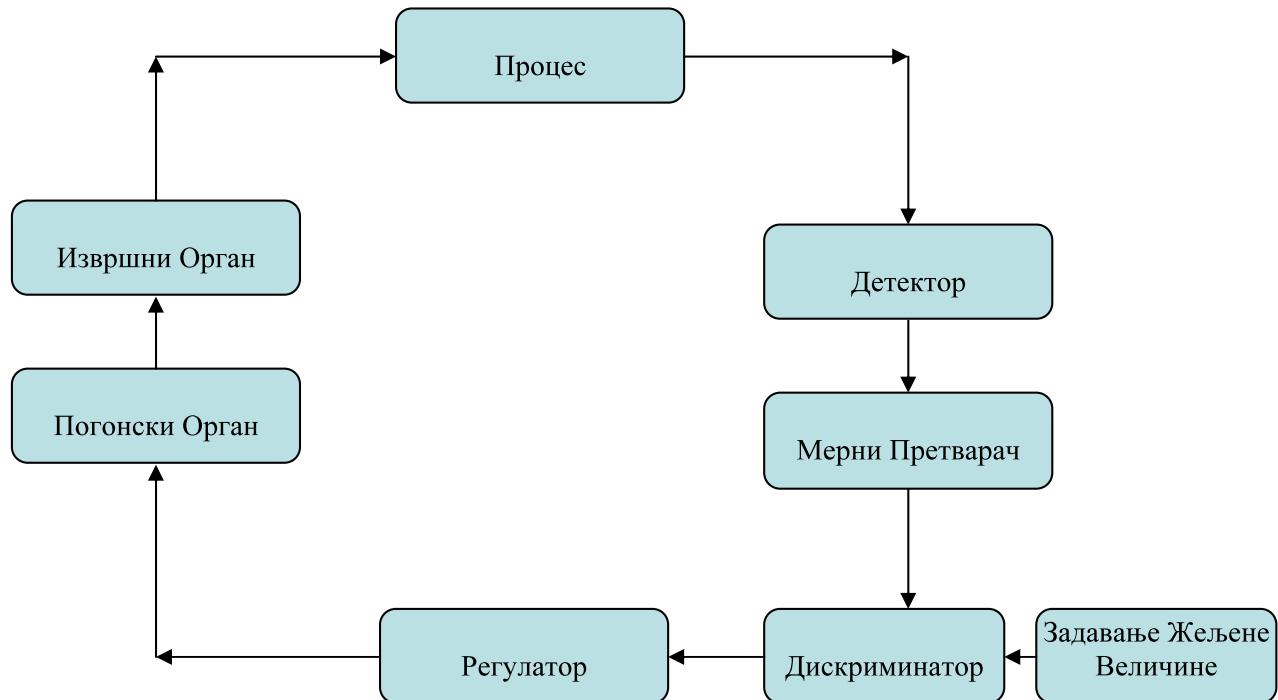
#### 1.4 САУ(СИСТЕМИ АУТОМАТСКОГ УПРАВЉАЊА)

Поред свих горе наведених области, рачунари се могу користити и у САУ (системи аутоматског управљања). То су системи чији елементи међусобно повезани могу да без људске помоћи управљају неким процесом.

У почетку су то били веома примитивни системи ( састојали су се од свега неколико елемената, који су углавном били гломазни и спори, и могли су управљати само једним процесом), док су се развојем технологије ти системи толико побољшали, да могу управљати и са неколико десетина процеса електронском брзином и са веома малим бројем елемената.

**Детектор** је уређај помоћу којег ми неку неелектричну величину конвертујемо у још једну неелектричну величину, која је погодна за претварање у електричну нпр. притисак се претвара прво у силу, па се сила претвара у електрични сигнал помоћу **Мерног претварача**. Из мерног претварача се та добијена величина, која је стандардизована, води у **Дискриминатор**, који одузима тај сигнал од сигнала који се налази у склопу који служи за **Задавање жељене величине**.

Главни део целог склопа је **Регулатор**. Он има 5 функција а то су : задавање задате вредности, дискриминатор „динамичка обрада сигнала,појачање,прилоагођење излаза регулатора улазу извршног елемента. Извршни елемент се састоји од 2 дела и његова улога је да делује на објект управљања.



Слика 3. Блок шема САУ

### 1.5 Рачунари У САУ

Као што је у претходном делу речено, рачунари се могу користити и у системима аутоматског управљања. Ову погодност користе скоро све фабрике на западу, док је код коришћење рачунара у САУ веома ретко. Помоћу рачунара се у фабрици повећава квалитет и број производа по јединици времена, пре свега зато што се повећава брзина, прецизност и број процеса којим се може управљати, јер када процесом управљају људи долази до велике спорости и у одзиву и у брзини реакције на брузу промену параметара ван дозвољене вредности, а и број процеса којим неколико људи може да управља је веома мали.

Да би неки процес могао да буде управљив помоћу рачунара, морамо имати интерфејс и софтвер. Интерфејс је електромеханички део, који нам служи као спона између процеса и рачунара. Рачунар са спољном средином разговара помоћу портова и то у бинарном језику због техничке реализације самог рачунара. Да би неку неелектричну величину увели у рачунар пре свега је треба превести у електрични облик. То се ради помоћу мерних елемената (на блок шеми сај детектор и мерни претварач). Када се то преведе у електрични облик, још није завршен посао, јер је тај сигнал аналоган, а већ је речено да рачунар са спољном средином разговара у дигиталном облику. Пребацивање из аналогног у дигитални облик се врши помоћу аналогно-дигиталних конвертора и тек се онда, ако нема никаквих додатака( комбинационе дигиталне мреже ), уводе у рачунар као улазне величине процеса, уз помоћ којих рачунар директно или индиректно доноси одлуке о деловању на процес. За деловање на процес се користе излази из рачунара . Они су у дигиталном облику, и можемо их по воли претворити у аналогни облик или оставити у дигиталном облику да тако управљају процесом. На стандардним рачунарима имају неколико портова, од којих ћу издвојити 2 а то су паралелни порт и серијски порт. Код паралелног порта се пренос података врши паралелно (сви битови улазе одједном ). Он се састоји од 2 порта- улазног и излазног, од којих можемо увести у порт максимално 8 бита одједним, а извести 8 бита одједном, као излазну комбинацију. Има 25 пинова. Серијски порт користи серијску комуникацију а то је комуникација код које се подаци у рачунар уносе бит по бит. О овом порту ће касније бити више речи. Лоша страна код паралелног пората у односу на серијски је релативно велики број проводника, док је предност већа брзина преноса у односу на серијски порт.

Да би рачунар могао правилно комуницирати са спољном средином, морамо имати програм рада (софтвер). Он се пише у неком од програмских језика ( Pascal , Basic , C , C++ ,Fortran ...). У овом раду ће се за програмирање користити програмски језик PIC Basic. Да би програм могао да правилно контролише процес програм мора у току управљања процесом пазити о следећим параметрима: стању исправности улазних и излазних уређаја, да уз помоћ неких формула у зависности од улазних величина проверава добијене улазне резултате и приказује грешку, да у сваком тренутку приказује оператору стање процесних величина и променљивих, графички и/или текстуално, да стално прати параметре процеса...

### **1.6 Микроконтролери**

Са развојем полупроводничке технологије и микропроцесора су се паралелно развијали и микроконтролери. То су електронски склопови који самостално могу да управљају процесом. У себи имају програм који управља излазним портом/портовима. Као и рачунари и микроконтролери да би управљали спољном средином морају имати и интерфејс који га повезује са спољном средином.

### **1.7 EEPROM**

У току програмирања микроконтролера нам се може десити да програм не можестати у интерну меморију микроконтролера. Да би се тај проблем превазишао морамо тај програм или смањити што је некад немогуће, или га ставити у спољну меморију Та спољна меморија може бити и EEPROM. Ова врста меморије се користи да се у њу запмте параметри процеса нпр. код телевизора се чува подешеност неких елемената, а код мобилних телефона се чува PIN код. Помоћу ове базе се програмира и ова меморија.

## 2. Теоретски увод у матурски рад

### 2.1 Микроконтролери

Врло често се сусрећемо а ситуацијом да нам при регулисању неких процеса није могуће извршити регулацију са аналогним електронским колима или са рачунаром због услова у којима се опрема налази , нпр. на веома малом простору, те се робусна аналогна кола или рачунари у том случају никако не могу користити. Зато се користе микроконтролери,који су најчешће у облику чипа, и заузимају веома мали простор. Такође у многим случајевима је хардвер за управљање процесом којим управља микроконтролер мањи.

Микроконтролери су дигитални уређаји који могу да контролишу неке процесе. Контролисање врше преко портова у који уводе улазне синале, који им говоре о тренутном стању процеса и помоћу којих шаљу излазне сигнале, који управљају спољним процесом. Ти улази и излази су у дигиталном облику, тј. раде са сигналима који имају само два стања: логичку нулу(0V), и логичку јединицу (5V), ако је у питању TTL стандард, што се најчешће и користи. Да би се нека неелектрична величина прилагодила улазима, прво се мора претворити у електрични сигнал, а онда из електричног сигнала који може бити аналоган, у облику фреквенције,или дигиталном , претвара у дигитални сигнал.Овај сигнал може да буде једно и вишебитни, а максималну дужину сигнала одређује број улазних битова микроконтролера.

Да би микроконтролер знао на који начин да управља спољном средином,морамо да му напишемо програм.Он се пише у неком од програмских језика , пребацује у низ нула и јединица, јер микроконтролер разуме само тај језик, и уписује у меморију микроконтролера.

Постоје велики број архитектура микроконтролера,а најчешће се састоје од:

1. управљачке јединице(ова јединица је јединица која служи за извођење свих логичких и аритметичких операција)
2. меморија (јединица у коју се уписује програм, и подаци добијени у току рада микроконтролера ...)
3. улазног дела (-је јединица која служи за пријем података из спољне средине и отпремање у регистре ,одакле се у току обраде сигнала узимају и обрађују)
4. излазног дела (-је јединица у коју се привремено смештају подаци који се шаљу на излаз и одакле их спољна кола узимају)
5. регистра (то су меморијски модули у које се привремено смештају подаци у току рада микроконтролера)
6. магистрала (то су линије помоћу којих се врши пренос података и инструкција између делова микроконтролера).

### 2.2 Меморије

Меморије су електронска кола која имају особину да трајно или привремено памте одређену количину података у дигиталном облику, и најчешће се врши подела по количини меморије коју могу да примити.

Постоје неколико врста меморија а то су:

1. RAM(меморија у коју се подаци могу произвољно уписивати и ишчитавати, али се по нестанку напајања губе сви подаци који су били уписаны у меморији)
2. ROM(меморија из које се подаци могу само ишчитавати али се по нестанку струје подаци не губе)
3. EEPROM меморија...

Меморије се најчешће сastoјe од дигиталних кола који се зову флип-флопови,који привремено могу памтити један или два бита, а комбинацијом флип-флопова и логичких кола се количина података која се може уписати у меморију теоретски може повећати бесконачно пута.

### 2.3 Програматори

Упис програма у микроконтролере се најједноставније врши уз помоћ рачунара.За то служе уређаји који се зову програматори. Постоји веома велики број микроконтролера, који при том имају различите распореде контаката за разговор са спољном средином, те се за сваки микроконтролер мора користити други програматор. Да би се избегао велики број програматора користи се универзални програматор, који за неколико микроконтролера има само једну, или неколико прикључака, а сам уређај се контролише помоћу рачунбарског програма. Поред микроконтролера овај уређај може да пуни и меморију, а са рачунаром се повезује помоћу портова(серијски,паралелни,...). У оквиру овог матурског рада ћемо се упознати са серијским универзалним програматором и при том користити микроконтролер PIC 16F84 и EEPROM

меморију 2404. То је универзални програматор који може да програмира велики број микроконтролера и уписује поратке у разне врсте меморија. Састоји се од две дела:

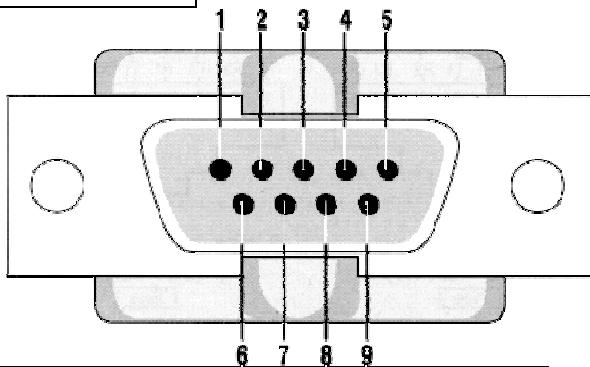
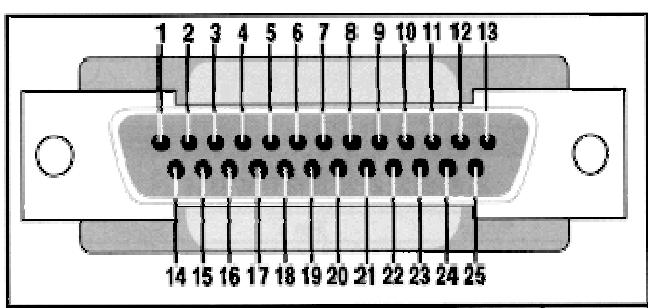
1. базе која врши стандардизацију серијског порта са микроконтролерима и меморијама,
2. модула који се наставља на базу који прилагођава распоред контаката са конектора базе распореду контаката на микроконтролеру,  
а принцип рада базе и два модула која се у оквиру матурских радова обрађују биће дата у неком од следећих поглавља.

### 3 Принцип рада

#### 3.1 Серијски порт

Серијски порт рачунара користи серијску комуникацију. То је врста комуникације код које се подаци преносе бит по бит. Серијски порт има два напонска нивоа: логичку нулу(3 до 25 волти) и логичку јединицу (-25 до -3 волте). На рачунару се могу наћи портови са 9 и 25 контаката чији је изглед дат на слици:

DSUB-25	DSUB-9	Signal	Opis signala
pin 2	pin 3	TD	Transmit Data
pin 3	pin 2	RD	Receive Data
pin 4	pin 7	RTS	Request ToSend
pin 5	pin 8	CTS	Clear To Send
pin 6	pin 6	DSR	Data Set Ready
pin 7	pin 5	SG	Signal Ground
pin 8	pin 1	CD	Carrier Detect
pin 20	pin 4	DTR	Data Terminal Ready
pin 22	pin 9	RI	Ring Indikator



Signal	Opis signala	Značenje signala
TD	Transmit Data	Serijski izlaz podataka(TXD)
RD	Receive Data	Serijski ulaz podataka(RXD)
RTS	Request ToSend	Indicira da je modem spreman za razmenu podataka
CTS	Clear To Send	Kada modem detektuje signal "Carrier" od modema sa druge strane linije, onda ona postaje aktivna
DSR	Data Set Ready	DCE signalizira da je spreman za rad
SG	Signal Ground	Masa
CD	Carrier Detect	DCE javlja da je veza uspostavljena
DTR	Data Terminal Ready	Indicira DCE uređaju da je DTE spreman
RI	Ring Indikator	Signalizira da je detektovan signal "zvona" na telefonskoj liniji

Контакти и њихово значење су дати у претходним табелама.

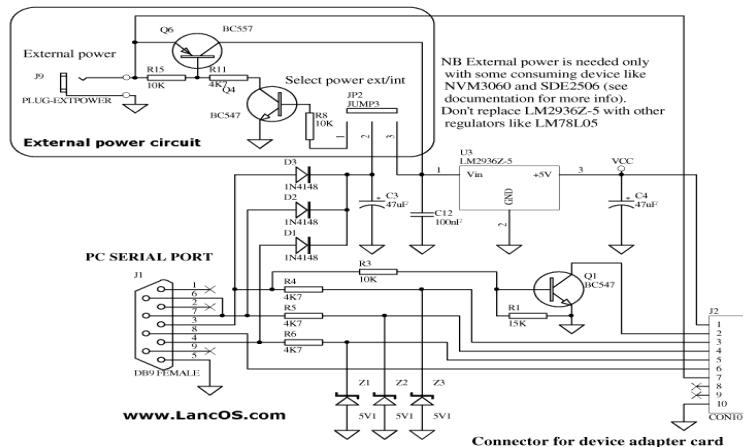
Пренос података почиње са старт битом. После тог бита долазе битови који садрже податке и бит који служи за детекцију евентуалне грешке у преносу. Комуникација(циклус) се завршава са стоп битом.

Серијски пренос почиње тако да предајна страна шаље пријемној страни и гннал RTS- захтев за слањем док пријемник ако је спреман за пријем одговара сигналом CTS-спреман за пријем. Предајник затим шаље податке пријемнику. Након пријема података пријемник проверава да ли су подаци примљени без грешке и за то време јавља преданику да је заузет. Ако су подаци примљени без грешке, пријемник шаље предајнику сигнал потврде(ACK Acknowledgement – ASCII 6), а у случају грешке шаље сигнал негативне потврде NAK(ASCII 21). Заисно од употребљеног протокола одговор на грешку је највероватније поновно слање сигнала. Протокол за пренос се може извршити хардверски и софтверски. За реализацију хардверског протокола је потребно постојање линије RTS и CTS и линија за пренос битова поруке. Софтверска реализација користи уместо споменутих сигнала асции знакове XON и XOFF, због чега је уместо три линије потребна само једна. Код овог протокола пријемници и на предаји и на пријему проверавају сваки приспели знак утврђујући да ли су то подаци или претходно описани знаци. Страна која не може да прими карактер као индикатор тог стања другој страни шаље XOFF знак.

### 3.2 База

База програматора је уређај на који се прикључују сви уређаји који програмирају микроконтролере. Она се прикључује на серијски порт рачунара тј. комуникацију са ралунаром врши помоћу серијског порта.

База програматора се састоји од неколико модула, који ће бити описани у даљем тексту. Отпорници R4, R5, R6, R8 служе за ограничење струје и заштиту од струје кратког споја. Отпорници R11, R15 и R1, R3 чине два разделника напона који нам дају стабилне напоне на базама транзистора и тиме обезбеђују да напони на базама буду нижи од напона на колектору (нпн) и напона на емитеру (пнп), чиме се омогућује да PN спојеви могу провести када струја протече кроз отпорнике. Зенер диоде које имају пробојни напон од 5 волти и служе да обезбеде напон од 5 волти на улазу даљих модула јер сви уређаји који се прикључују на базу раде у TTL стандарду. Транзистор Q1 служи као инвертор са отвореним колектором, те имамо инвертовани трећи контакт на конектору. Остале два транзистора служе као прекидачи и принцип рада ће бити дат у даљем тексту. Стабилизатор напона служи да обезбеди напајање од 5 волти за остале елементе програматора.



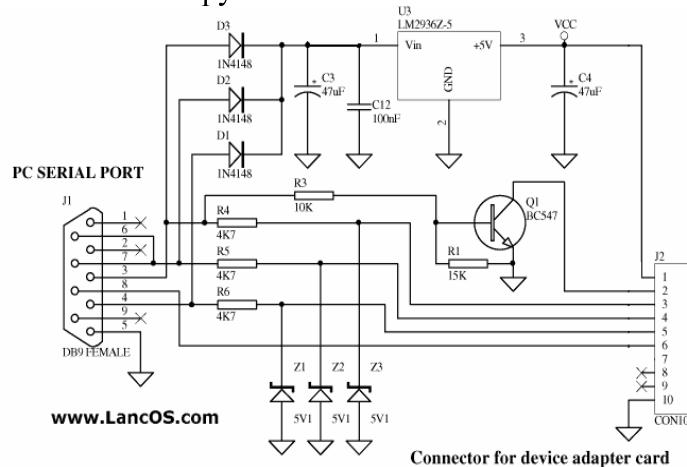
Слика 1.Шема базе програматора

Модул напајања се састоји од 2 транзистора, 3 отпорника, стабилизатора напона и ектсерног извора напајања. Да би екстерно напајање имало ефекта цампер мора да спаја контакте 1 и 2. Када је напон на бази првог транзистора(нпн) позитиван (5-25 волти са серијског порта), струја тече кроз транзистор од колектора ка емитеру и на разделнику напона (бази другог транзистора) се јавља напон који је мањи од напона на емитеру , те се струја од емитера креће према колектору, а преко којектора према стабилизатору напона, који ако је тај напон већи од 5 волти тај напон стабилизује на 5 волти ,чиме је постигнут напон у TTL стандарду(највећи проблем у серијској комуникацији је шо је напон логичке нуле позитиван и износи и до 25 волти, а логичке јединице негативан и износи око -25 волти), те се преко првог контакта на конектору разноси на остале елементе програматора, и стандардизација напона је компликованија. Ако је напон на бази негативан или мањи од напона екстерног напајања транзистори не могу да проводе јер PN спојеви нису могли да пробију напонску барјеру, али како год било, са екстерног напајања се напон води директно на 7. контакт конектора и користе га неки од елемената који су најчешће у C-MOS стандарду.

Са 3. пина серијског порта се сигнал води на 3. контакт конектора али се води и на разделник напона који на бази даје стални напон који је мањи од напона на колектору. Са шеме видимо да је колектор отворен. Контакт 2 се спаја на напајање од 5 волти преко отпорника у следећем модулу, чиме се на колектор доводи напајање , и када је напон на бази мањи од напона на колектору и за 0,7 волти већи од напона на колектору транзистор води и на другом контакту имамо напон логичке нуле, док на терћем имамо напон логичке јединице. Ако серијски порт даје напон логичке нуле, на 3. контакту конектора имамо напон логичке нуле, а пошто је напон на бази транзистора мањи од напона на емитеру, транзистор не проводи те на 2. контакту конектора имамо напон логичке јединице. Контакти 2 и 3 на конектору служе да ресетују микроконтролер или меморију.

Контакт 4 на конектору са 2 и 7 пина серијског порта доводи тактни сигнал на четврти контакт конектора. Са 4. пина серијског порта се доводе подаци који се у следећем модулу уписују у чипове. Код 3, 4, и 5. контакта отпорници служе као заштита од прекорачења струје, а зенер диоде служе да обезбеде напон у TTL стандарду од 5 волти и 0 волти.

У току овог матурског рада нисмо правили модул напајања, те су контакти цампера 2 и 3 кратко спојени, и не доводи се екстерно напајање на 7. pin конектора. Та шема је дата на слици 2, а у табели 1 је дат распоред пинова на конектору.



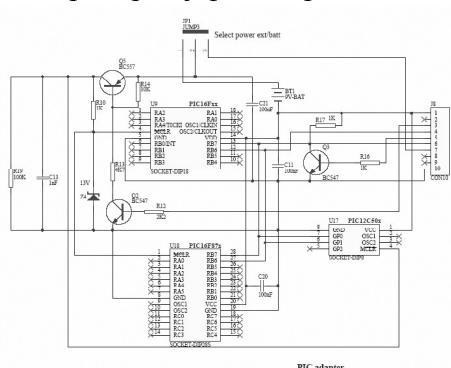
Слика 2. права шема базе програматора

Број контакта	Намена
1	VCC 5V
2	Not Rst
3	Rst
4	Clk
5	Data In
6	Data Out
7	VCC 9 V
8	Not in use
9	Not in use
10	GND

Табела 1. Распоред контаката на конектору

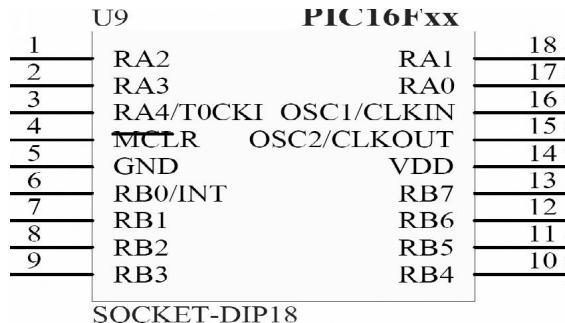
### 3.3PIC Програматор

Ово је модул којим ми програмирајо микроконтролер. На себи има гнездо у које се ставља микроконтролер, електронске делове, као и приклучак екстерног напајања. Са 1. контакта конектора се доводи напајање од 5 волти. Са трећег се доводи сигнал за ресетовање, преко транзистора који ради исто као модул напајања код базе и расподељује се на контакте свих макроконтролера, јер се преко ове карте могу уписивати на три места у зависности од типа микроконтролера. Шема модула је дата на слици 3.



Слика 3. Шема модула за програмирање PIC микроконтролера

Помоћу цампера се одређује да ли хоћемо имати екстерно напајање, или ћемо користити напајање са базе. У овом матурском раду се неће користити гнездо за микроконтролере са 8 и 20 пинова. Такође, не користи се транзистор Q2, те је и принцип рада мало модификован- ако доведемо логичку нулу, ресетоваћемо микроконтролер, јер ће тад транзистор да проведе, а логичком јединицом одржавамо стање да се микроконтролер не ресетује. Пинови микроконтролера који ћемо програмирати у овом раду је дат на слици 4. Уписивање података у микроконтролер се врши помоћу 5. и 6. контакта конектора а преко 10 контакта конектора се доводи маса у модул.



Слика 4. Пинови на микроконтролеру PIC 16F84

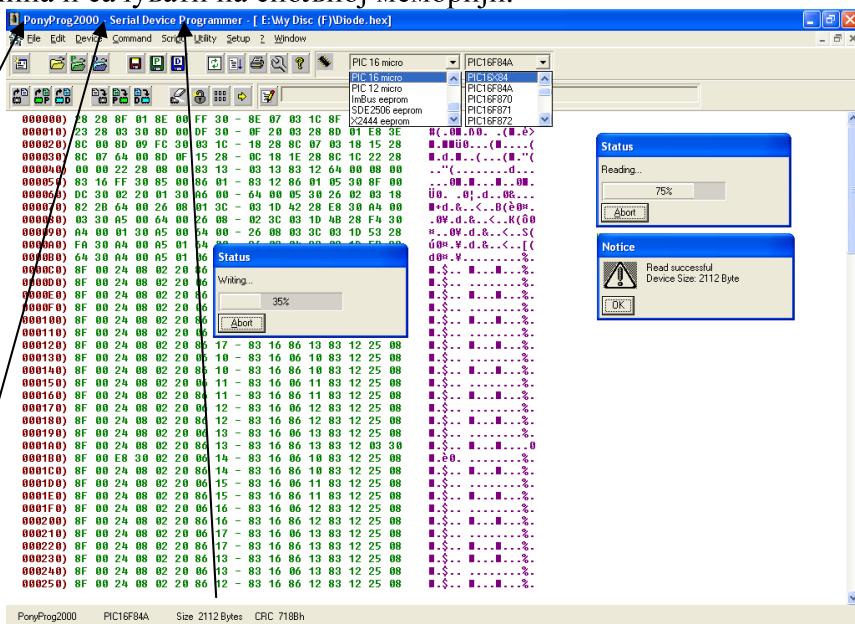
### 3.4 Програм за програмирање микроконтролера

Да би уписали програм у микроконтролер морамо користити серијски порт рачунара. То значи да нам програми које уписујемо морају бити у бинарном облику. У микроконтролер се зато упуisuју тзв. хексафајлови. Они се састоје од комбинација нула и јединица. То су инструкције које микроконтролер у себи преводи и обрађује их.

У овом матурском раду смо користили програм Pony Prog 2000, који смо нашли на истој адреси на којој смо нашли и шеме([www.lancos.com](http://www.lancos.com)). Овај матурски рад је интерфејс за тај програм. Он је у зипованом облику и после инсталације програма треба отићи у мениј SETUP и извршити калибрацију и подешавања програма за уређај.

После извршеног подешавања програма прикључује се уређај и у подешавању се покрене тест и ако је све у реду можемо почети са радом. Прво се одреди чип који пунимо или програмирамо. То се налази у линији алата где се прво одреди којој групи припада, а онда се одређује које интегрално коло програмирамо. Отвори се хексафайл који се хоће уписати и у менију COMMAND где одаберемо опцију WRITE ALL. После одређеног кратког временског периода долази информација да је садржај који смо желели уписан у чип. У овом програму се може и ишчитавати садржај чипа и сачувати на спољној меморији.

Слика 5. Пример рада у програму Pony Prog 2000



Ишчитавање садржаја микроконтролера се врши менију COMMAND са командом READ ALL.

### 3.5 PIC BASIC

У програму који смо користили можемо само уписивати и ишчитавати податке из микроконтролера и меморија. Да бисмо програмирали уређај у том облику морали би уложити веома много труда ,уз велику могућност грешке. Зато се користи програмски језик PIC BASIC који служи за програмирање ових врста микроконтролера. После писања програма у овом програмском језику он се компајлира, тестира на грешке, и тако се пребацује у низ нула и јединица које разме микроконтролер тзв. Хексафајлове.

Микроконтролери могу да расподеле своје портove тако да им сви делови буду улазни , да су сви излазни, а сваки бит порта се може независно од других наместити да ради као улазни или излазни. Да би наместили да је сваки порт улазни користимо команду:

TRISA = \$FF ' Svi pinovi porta A su ulazni

TRISB = \$00 ' Svi pinovi porta B su ulazni.

Постаљање вредности портова се могу урадити на неколико начина:

1. тако што се сваком биту прикачи нека вредност(1 или 0 )

2. задавањем вредности порта

1.

High PORTB.5 ' Upali Led 5

pause j

High PORTB.6 ' Upali Led 6

pause j

High PORTB.7 ' Upali Led 7

pause j

2. PORTB = \$FF

pause 300

PORBTB = \$00

pause 300

После писања програма (програм је дат у прилогу) прелазимо на компајлирање, где се отвара свака евентуална грешка и такав програм се убацује у тзв. Хексафајл. Тај фајл се касније уз помоћ других програма уписује у микроконтролер. Постоји опција Да се може изабрати за који контролер пишемо програм, јер сваки микроконтролер има своје комбинације нула и јединица.

MicroCode Studio - PICBASIC (C:\Matura\diode2.bas)

File Edit View Project Help

16F628

Code Explorer

diode2 Matura

```
286 HIGH PORTB.5      ' Upali Led 5
287 PAUSE j
288
289 HIGH PORTB.3      ' Upali Led 3
290 HIGH PORTB.4      ' Upali Led 4
291 PAUSE j
292
293 LOW PORTB.3       ' Ugasi Led 3
294 LOW PORTB.4       ' Ugasi Led 4
295 PAUSE j
296
297 LOW PORTB.5       ' Ugasi Led 5
298 LOW PORTB.2       ' Ugasi Led 2
299 PAUSE j
300
301 LOW PORTB.6       ' Ugasi Led 6
302 LOW PORTB.1       ' Ugasi Led 1
303 PAUSE j
304
305 LOW PORTB.7       ' Ugasi Led 7
306 LOW PORTB.0       ' Ugasi Led 0
307 PAUSE j
308 PAUSE 1000
309
310 ' Dva puta blinika na kraju ciklusa
311 FOR k = 1 TO 2
312   PORTB = $FF      ' Upali sve LED diode
313   PAUSE 300
314   PORTB = $00      ' Ugasi sve LED diode
315   PAUSE 300
316 NEXT k
```

Слика 6. Пример рада у едитору PIC BASIC-a

Неке од команди су :

IF ... Then : је функција која ради инструкције после Then ако је тачна претпоставка иза команде IF.

Call label: позива неку лабелу која може бити нека функција, потпрограм или само враћање на неку лабелу.

End : функција која означава крај програма или процедуре.

**EEPROM** {Location, } [Constant{, Constant...}] уписује константу у EEPROM меморију.

HIGH, PIN : шаље логичку јединицу на пин порта који је задат.

LOW, PIN : шаље логичку нулу на задати пин.

У овом програмском језику има веома много инструкција, а ово су само неке од најчешће коришћених инструкција.

Ово је веома једноставан програмски језик а програм подржава следеће типове микроконтролера:

16F876(A) and 16F877(A) 16F870, 16F871, 16F873(A), 16F874(A), 16F876(A),  
16F877(A), 16F87, 16F88, 18F242, 18F248, 18F252, 18F258, 18F442, 18F448, 18F452, 18F458,  
18F1220, 18F1320, 18F2220, 18F2320, 18F4220, 18F4320, 18F6620, 18F6720, 18F8620,  
18F8720, 18F2331, 18F2431, 18F4331, 18F4431, 18F6585, 18F8585, 18F6680, 18F8680,  
18F6627, 18F6722, 18F8627, 18F8722, 18F2525, 18F2620, 18F4525, 18F4620, 18F2455,  
18F2550, 18F4455, 18F4550, 18F2420, 18F2520, 18F4420, 18F4520, 18F2439, 18F2539,  
18F4439, 18F4539, 18F2480, 18F2580, 18F4480, 18F4580, 18F2585, 18F2680, 18F4585,  
18F4680, 18F6520, 18F8520, 18F6525, 18F6621, 18F8525 and 18F8621.

#### 4 Закључак

У овом матурском раду смо изучавали област чија будућност тек надолази, а то је област микроконтролера-рачунара у облику чипа. Микроконтролери се због задатака које могу да обаве зову и микропроцесори, јер могу за веома кратко време да обраде веома много података. Микроконтролери се данас најчешће користе за контролисање индустриских постројења и погона, јер ако се правилно направи хардвер и програм, један микроконтролер може управљати скоро целом фабриком, под условом да има доволјан број улазних и излазних портова. Већина данашњих микроконтролера који се сваким даном све више усавршавају, могу да подрже веома захтевне процесе захваљујући пре свега релативно великој количини меморије намењеној за рад са програмима, омогућеном прикључивању спољних меморија, великој брзини обраде и преноса података као и реакцији улазних и излазних портова, а побољшањем дигиталне технике као и технике прављења интегрисаних кола се у будућности све више повећавају перформансе микроконтролера, који ће у веома дogleдно време моћи управљати, уз веома мале трошкове и веома брз одзив, не само постројењима већ и комплетним електросистемима вишемилионских градова. Једина мана микроконтролера је дијалог са корисницима, јер се излазни модули као што су дисплеји, штампачи ... морају накнадно правити а неки од микроконтролера чак и не подржавају ту технологију, али то су најчешће неки старији типови микроконтролера, који се данас више и не користе.

У почетку микроконтролери су били веома спори и могли су да врше само основне аритметичке и логичке операције, али су за веома кратко време брзина преноса код микропроцесора са максимално 2 мегахерца порасли и на неколико гигхерца, а количина меморије је порасла скоро неколико хиљада пута са даљом тенденцијом раста.

Поред микроконтролера у овом матурском раду смо изучавали и област меморија. Меморије су електронска кола која могу да приремено или за стално памте одређену количину података. Данашња количина података које меморије могу да запамте је фасцинантна. Код савремених рачунара количина меморије иде и до неколико стотина гигабајта(за хард дискове), а за електронске ,меморије као што је USB FLASH меморија, количина меморије је и неколико гигабајта на веома малом простору од неколико центиметара квадратних (1 GB=1024MB, 1MB=1024KB, 1KB=41024B, 1B=8 BIT, 1GB=8589934592BIT).

Главна тема матурског рада је била база програматора за упис података у микроконтролер и меморију. Принцип рада базе је веома једноставан и објашњен је у принципу рада.Рачунари су и у овој области (програмирања микроконтролера) преузели вођство, пре свега због једноставности употребе програма који се користе за програмирање као и цене рада рачунара и интрејфјеса који служи у те намене. Начин рада са програмима за програмирање и упис података у микроконтролер је дат у принципу рада.

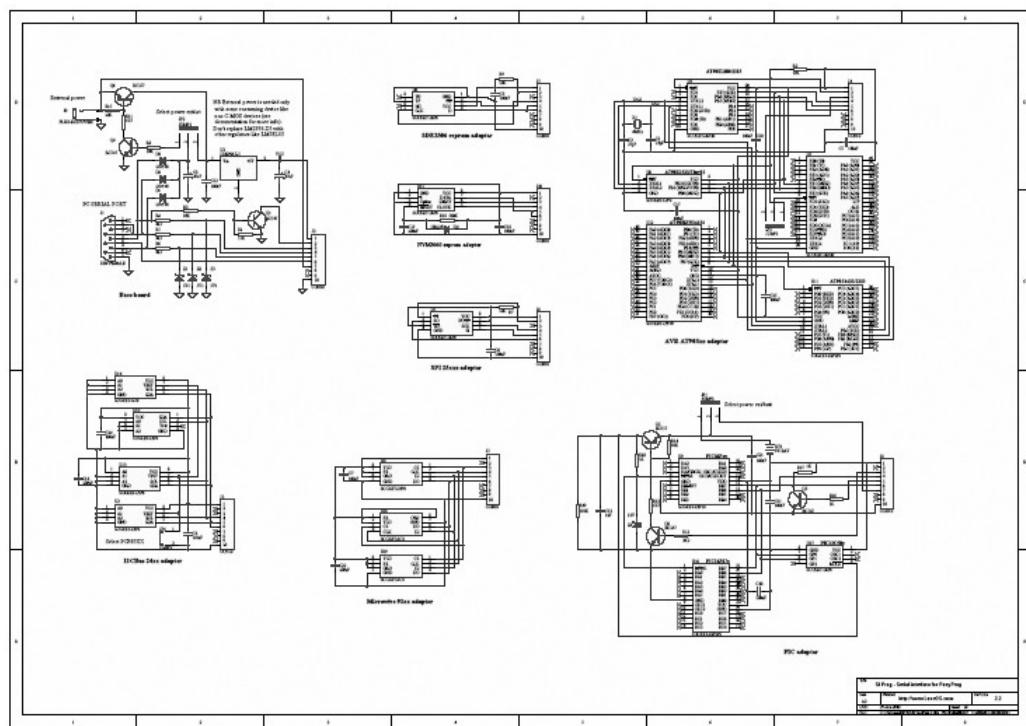
Један од главних предности коришћења микроконтролера је коришћење интегрисане технологије, што значајно смањује простор потребан за модуле микроконтролера. Да би се користила та технологија, користе се веома ригорозни стандарди што код коришћења аналогне технике врло често није случај.

Микроконтролери све своје податке обрађују у бинарном облику тј. помоћу јединица и нула. Пошто је такво програмирање веома тешко користе се програмски језици код којих су команде најчешће скраћенице енглеских речи који означавају команду. Овакви програмски језици се морају пребацити у машински језик, што раде програми који се зову асемблери. Сваки микроконтролер има свој асемблер тј. команде које се користе зависе од типа микроконтролера, али је већина команда сличне синтаксе. Код савремених микропроцесора су команде све исте. Од микропроцесора i8086 су се команде само повећавале, али су све новије генерације микропроцесора подржавале старе асемблерске инструкције. Једна од препрека код коришћења асемблера је велика дужина програма и што се морају познавати неки основни подаци о микропроцесору или микроконтролеру који се програмира, али је извршење програма у асемблерском језику знатно брже него код осталих програмских језика.

## 5 Прилог

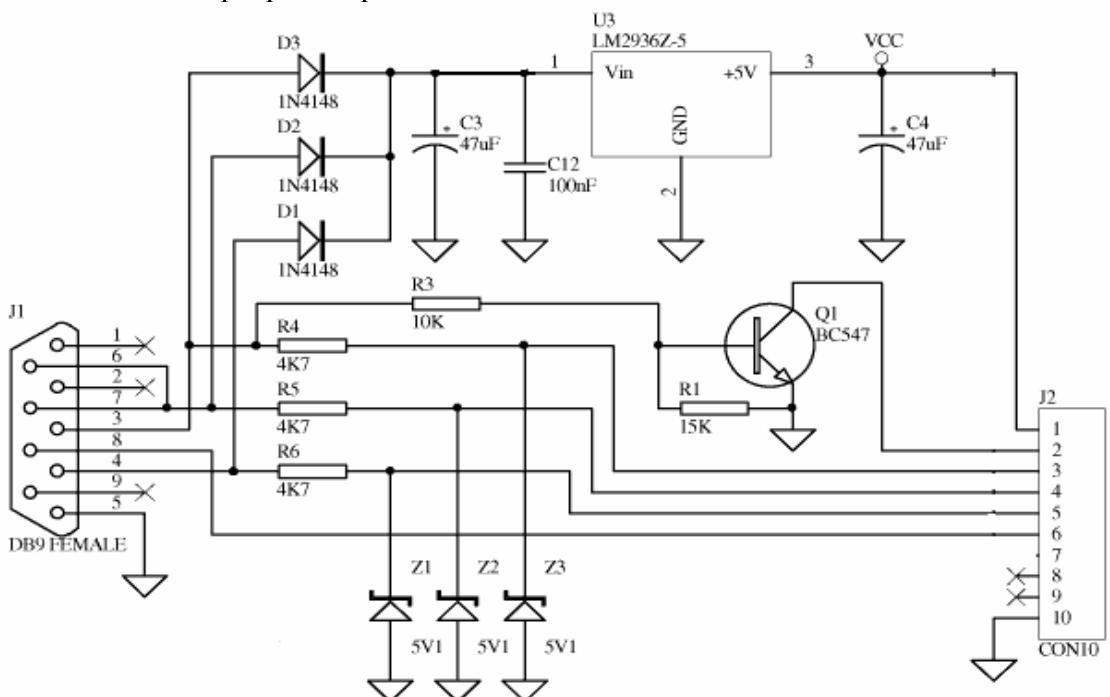
У овом матурском раду смо правили серијски интерфејс за Pony Prog 2000- програм који служи за програмирање микронтролера и пуњење меморије. Шеме које смо користили у овом раду су следеће:

- шема свих модула:



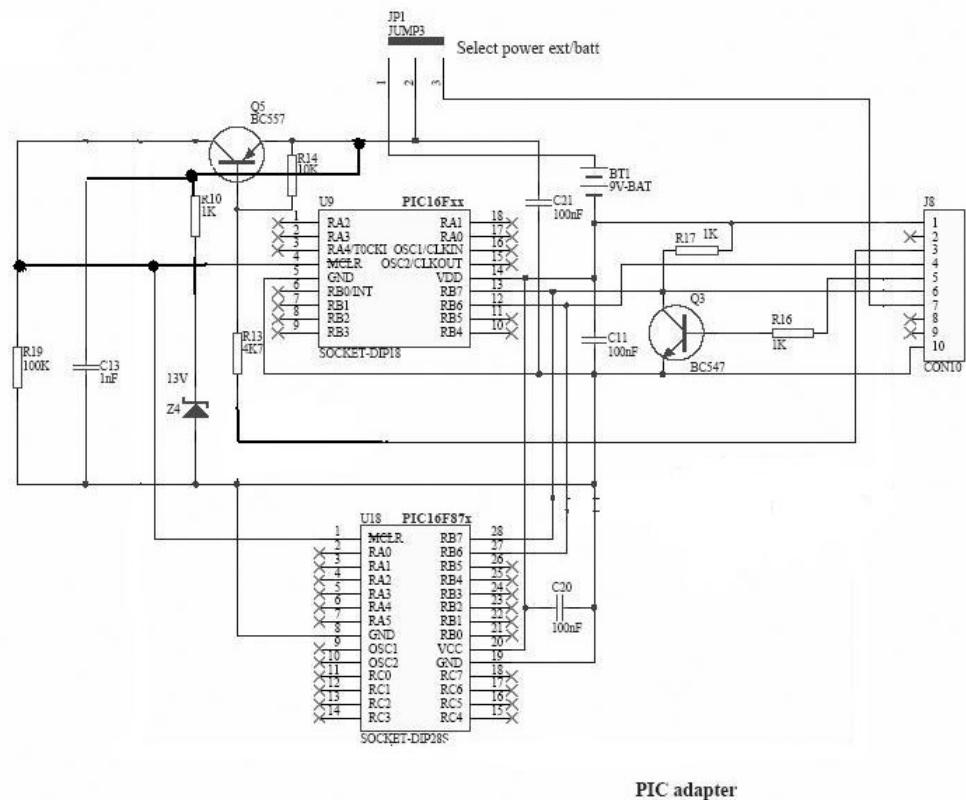
Слика 1. Шема свих модула

- шема базе програматора:



Слика 2. шема базе програматора

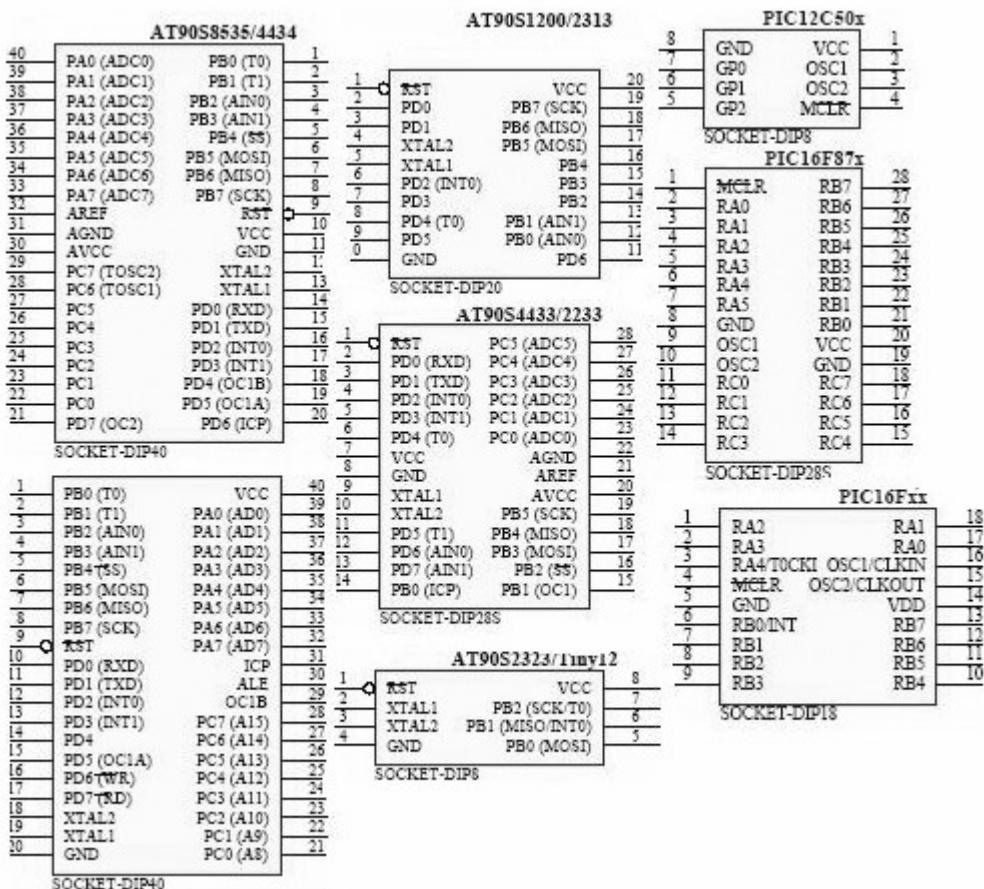
3. шема модула за програмирање PIC 16F84:



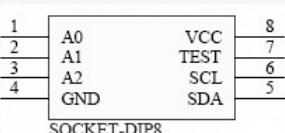
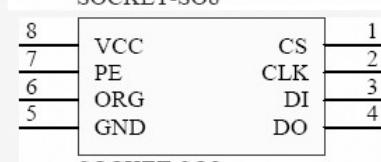
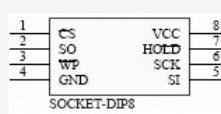
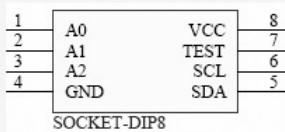
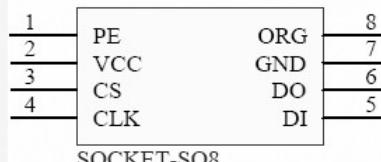
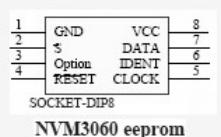
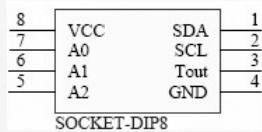
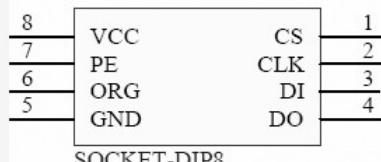
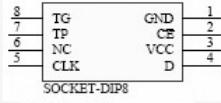
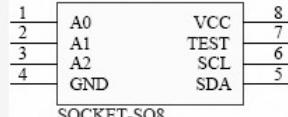
PIC adapter

4. шема меморије

Пинови неких микроконтролера:



Пинови неких микроконтролера



## Microwire 93xx

Неки од интегрсаних кола који се програмирају у склопу целог интерфејса

### Пример програма у програмском језику PIC BASIC

```
*****
***** Name : DIODE.BAS *
***** Autor : Milanovic V. *
***** Napomena : Primer prostog programa *
***** Datum : 21.04.2004 *
***** Verzija : 1.0 *
***** Napomena : Program generise razlicite sekvence na LED
*
*: diodama *
***** i var Byte
***** j var Word
***** k var Byte
TRISA = $FF ' Svi pinovi porta A su ulazni
TRISB = $00 ' Svi pinovi porta B su ulazni
PORTB = $00 ' Ugasi sve LED diode na pocetku
programa
pause 1500
Pocetak:
For i = 1 To 4
-----Pali jednu po jednu diodu i gasi od pocetka-----
If i = 1 Then j = 1000 ' Pauza 1 sekund
If i = 2 Then j = 500 ' Pauza 0.5 sekunda
If i = 3 Then j = 250 ' Pauza 0.25 sekunda
If i = 4 Then j = 100 ' Pauza 0.25 sekunda
High PORTB.0 ' Upali Led 0
pause j
High PORTB.1 ' Upali Led 1
pause j
High PORTB.2 ' Upali Led 2
pause j
High PORTB.3 ' Upali Led 3
pause j
High PORTB.4 ' Upali Led 4
pause j
High PORTB.5 ' Upali Led 5
pause j
High PORTB.6 ' Upali Led 6
pause j
High PORTB.7 ' Upali Led 7
pause j
Low PORTB.0 ' Ugasi Led 0
pause j
Low PORTB.1 ' Ugasi Led 1
pause j
Low PORTB.2 ' Ugasi Led 2
pause j
Low PORTB.3 ' Ugasi Led 3
pause j
Low PORTB.4 ' Ugasi Led 4
pause j
Low PORTB.5 ' Ugasi Led 5
pause j
Low PORTB.6 ' Ugasi Led 6
pause j
Low PORTB.7 ' Ugasi Led 7
Pause 1000 ' Pauza 1 sekunda
-----Pali jednu po jednu diodu i gasi od kraja-----
High PORTB.0 ' Upali Led 0
pause j
High PORTB.1 ' Upali Led 1
pause j
High PORTB.2 ' Upali Led 2
pause j
```



```

Low PORTB.4  ' Ugasi Led 4
High PORTB.5  ' Ugasi Led 5
pause j

Low PORTB.3  ' Ugasi Led 3
High PORTB.4  ' Upali Led 4
pause j

High PORTB.3  ' Upali Led 3
Low PORTB.2  ' Ugasi Led 2
pause j

High PORTB.2  ' Upali Led 2
Low PORTB.1  ' Ugasi Led 1
pause j

High PORTB.1  ' Upali Led 1
Low PORTB.0  ' Ugasi Led 0
pause j

High PORTB.0  ' Ugasi Led 0
Pause 1000  ' Pauza 1 sekunda

PORTB = $00  ' Ugasi sve LED diode
pause 300

'----Pali sa krajeva LED pa ih gasi-----
High PORTB.0  ' Upali Led 0
High PORTB.7  ' Upali Led 7
pause j

High PORTB.1  ' Upali Led 1
High PORTB.6  ' Upali Led 6
pause j

High PORTB.2  ' Upali Led 2
High PORTB.5  ' Upali Led 5
pause j

High PORTB.3  ' Upali Led 3
High PORTB.4  ' Upali Led 4
pause j

Low PORTB.3  ' Ugasi Led 3
Low PORTB.4  ' Ugasi Led 4
pause j

Low PORTB.5  ' Ugasi Led 5
Low PORTB.2  ' Ugasi Led 2
pause j

Low PORTB.6  ' Ugasi Led 6
Low PORTB.1  ' Ugasi Led 1
pause j

Low PORTB.7  ' Ugasi Led 7
Low PORTB.0  ' Ugasi Led 0
pause j
pause 1000

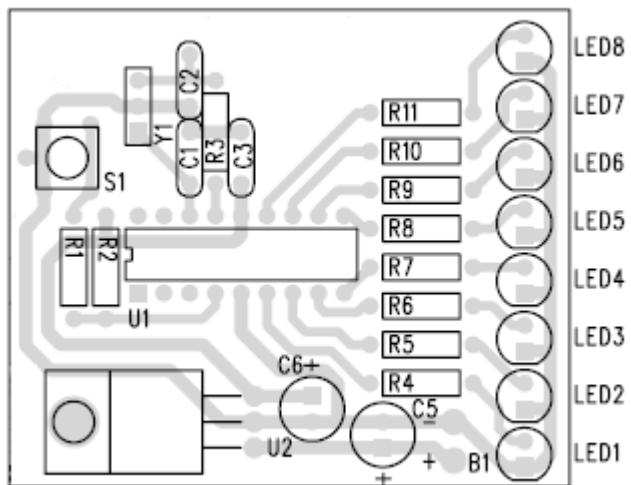
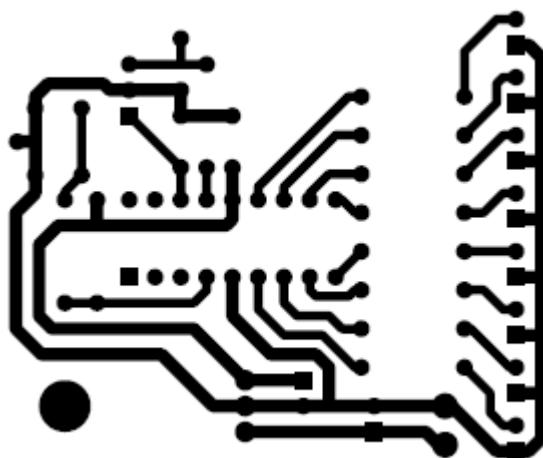
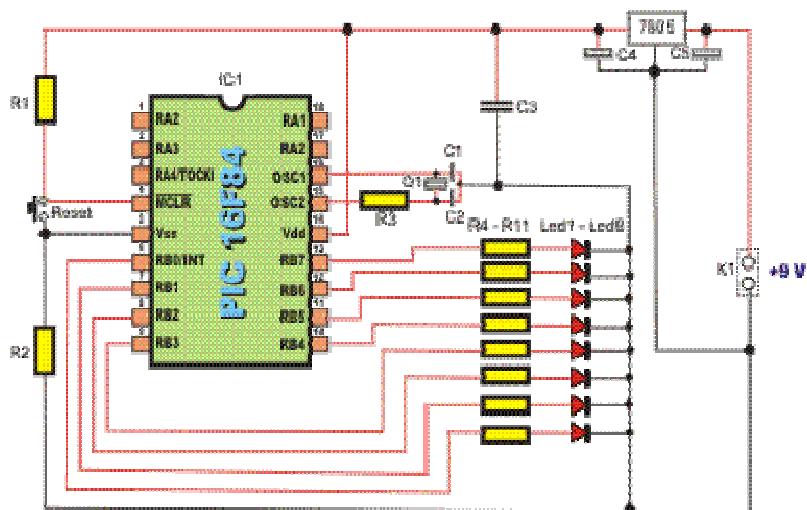
'Dva puta blinka na kraju ciklusa
For k = 1 To 2
    PORTB = $FF  ' Upali sve LED diode
    pause 300
    PORTB = $00  ' Ugasi sve LED diode
    pause 300
Next k
pause 1000

Next i
pause 1000

'-----Tri puta blinkanje dioda-----
For i = 1 To 3
    PORTB = $FF  ' Upali sve LED diode
    pause 200
    PORTB = $00  ' Ugasi sve LED diode
    pause 200
Next i
pause 1000  ' Pauza 1 sekunda
GoTo Pocetak  ' Vraca se na pocetak programa

```

Као подршка овом амтурском раду је првљено и тестно коло за микроконтролер чија је шема, штампана плоча и расоред елементата на штампаној плочи дати на следећим сликама:



## Теорија неких елемената који чине базу

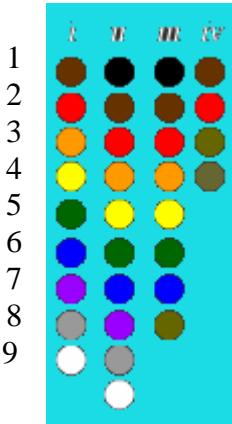
Диода је полупроводнички елемент који се састоји од PN споја и прикључака. Диоде се праве тако што се једном делу полупроводничке плочице додају петовалентне а другом тровалентне примесе. Тиме се унутар полупроводника остварују шупљине и вишак електрона, те се услед тога долази до дифузног кретања, до неке границе и у једном појасу остаје вишак позитивног а у другом негативног наелектрисања, те се јавља потенцијална баријера и изниси од 0,2 до 0,7 V. Она се може повећати али и смањити довођењем спољног напона, у зависности како је диода поларисана. Ако је поларисана директно баријера се смањује и ако је спољни напон већи од напона баријере диода поиње да проводи. Ако се диода инверзно поларише повећава се површина потенцијалне баријере и диода никако не може да проведе. Диоде се због овог својства провођења само у једном смеру користе као усмерачки елементи у претварачима наизменичне у једносмерну струју.

Транзистор је полупроводнички елемент који се у суштини састоји од две диоде. Постоје две врсте транзистора: PNP , NPN. Транзистори се састоје од три дела: колектор, емитер и база. Транзистори се најчешће користе као прекидачи а због својих појачавачких способности користе се у многим аудио-видео уређајима.

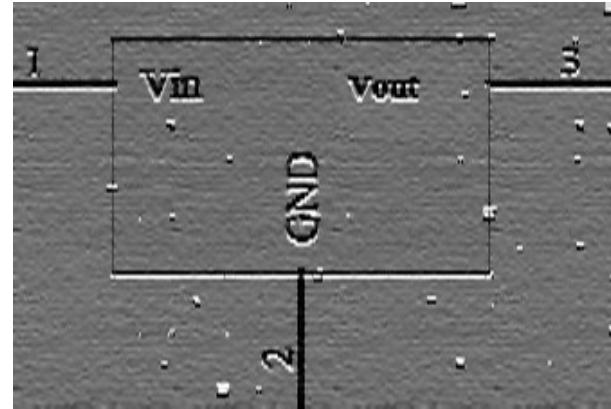
Зенер диода је полупроводнички елемент. Она има способност да при прекорачењу неких напона тнапон који се не сме прекорачити остаје на контактима диоде(при инверзној поларизацији). Увек се поларишу у колу инверзно и најчешће се користе у неким стабилизаторима напона

Отпорник је елеменат који се састоји од тела по којем је намотана жица. Жица која је намотана на тело има неку отпорност при кретању струје. Вредности отпорности могу бити веома мали или велики ,што зависи од врсте проводника који се користи при изради проводника. Отпорници се обележавају на два система: словима и бројевима, или бојама(што је боље решење јер се боје слабо бришу). Систем боја је дат на слици:

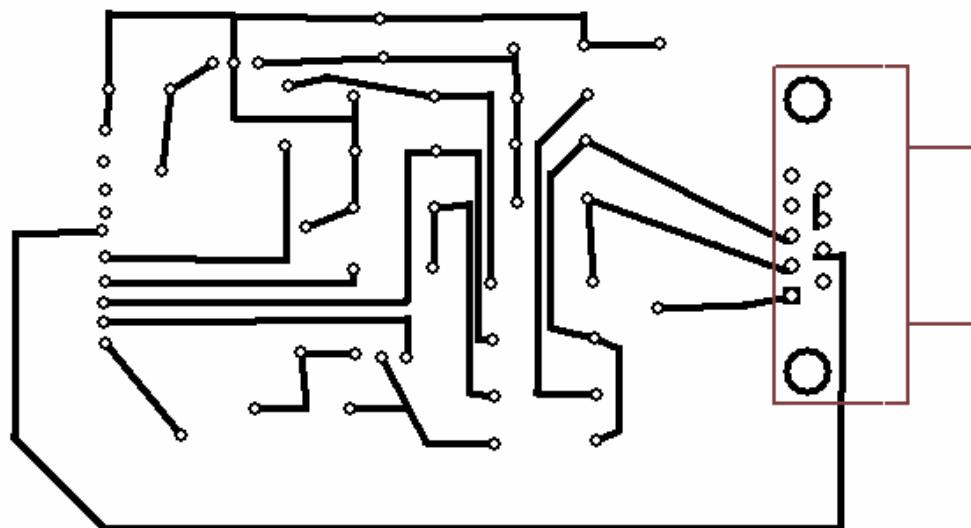
Слка 1 озн.отпорника бојама 1



Стабилизатор напона је електронско коло које служи да нам ограничи напон на неку одређену вредност при прекорачењу одређеног напона. Најпознатије су две врсте 79xx 78xx. Прва врста је стабилизатор напона за негативне напоне а xx је ознака максималног напона који дају. Друга је за позитивне напоне. Код стабилизатора напона ако се напон налази испод одређеног напон не ограничава већ остаје какав јесте нпр. макс. напон је 9 V , а тренутни 2.5V. Пошто напон није прешао максимални напон на излазу је исти као напон на улазу. Стабилизатори напона имају најчешће три контакта: улаз, масу и излазни контакт.



Стабилизатор напона(Контакти)



Изглед штампана плоче одгоре

Подаци о микроконтролеру

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM

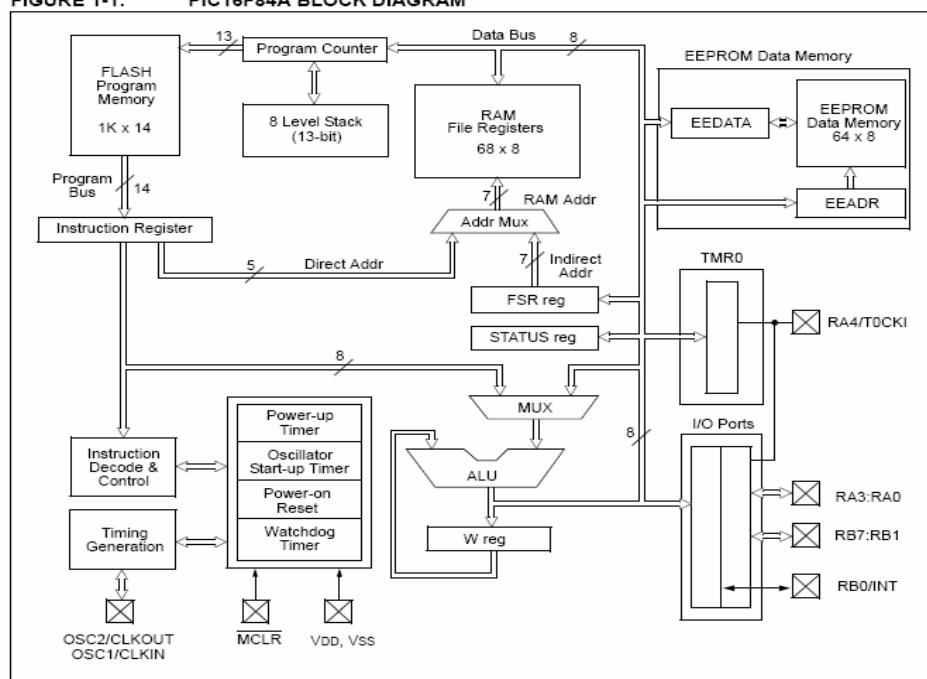


TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	17	19	I/O	TTL	PORTA is a bi-directional I/O port.
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
RB0/INT	6	6	7	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST <sup>(2)</sup>	Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST <sup>(2)</sup>	Interrupt-on-change pin. Serial programming data.
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

## 9.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

Ambient temperature under bias	.....	-55°C to +125°C
Storage temperature	.....	-65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, MCLR, and RA4)	.....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	.....	-0.3 to +7.5V
Voltage on MCLR with respect to Vss <sup>(1)</sup>	.....	-0.3 to +14V
Voltage on RA4 with respect to Vss	.....	-0.3 to +8.5V
Total power dissipation <sup>(2)</sup>	.....	800 mW
Maximum current out of Vss pin	.....	150 mA
Maximum current into VDD pin	.....	100 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD)	.....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD)	.....	± 20 mA
Maximum output current sunk by any I/O pin	.....	25 mA
Maximum output current sourced by any I/O pin	.....	25 mA
Maximum current sunk by PORTA	.....	80 mA
Maximum current sourced by PORTA	.....	50 mA
Maximum current sunk by PORTB	.....	150 mA
Maximum current sourced by PORTB	.....	100 mA

**Note 1:** Voltage spikes below Vss at the MCLR pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the MCLR pin rather than pulling this pin directly to Vss.

**2:** Power dissipation is calculated as follows: P<sub>dis</sub> = V<sub>DD</sub> × {I<sub>DD</sub> - Σ I<sub>OH</sub>} + Σ {(V<sub>DD</sub>-V<sub>OH</sub>) × I<sub>OH</sub>} + Σ (V<sub>OL</sub> × I<sub>OL</sub>).

## High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM

- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

### **Peripheral Features:**

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
- 25 mA sink max. per pin
- 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

### **Special Microcontroller Features:**

- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

### **Pin Diagrams**

### **CMOS Enhanced FLASH/EEPROM**

#### **Technology:**

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @

### **6 Коришћена литература**

1. Примена рачунара у електротехници-Душан Смиљанић,Небојша Петровић,Миомир Филиповић-зунз 2000.
2. Електронски рачунари и компоненте-Живко Тошић,Момчило Ранђеловић,Александар А. Коцић-зунс 2003.
3. Електронска књига фирме Microchip и PDF формату

Сајтови:

1. [www.lancos.com](http://www.lancos.com)
2. [milan.milanovic.org/skola](http://milan.milanovic.org/skola)
3. [www.elektroda.net](http://www.elektroda.net)

Све шеме које су коришћене су проверене програмом Electronics Workbench 5 demo

## Садржај:

<u>1 Увод</u> .....	1
1.1 Историјски развој рачунара.....	1
1.2 Рачунар.....	2
1.3 Примена рачунара.....	3
1.4 Системи аутоматског управљања.....	4
1.5 Рачунари у системима аутоматског управљања.....	5
1.6 Микроконтролери.....	6
1.7 EEPROM.....	6
<u>2 Теоретски увод у матурски рад</u> .....	7
2.1 Микроконтролери.....	7
2.2 Меморије.....	7
2.3 Програматори.....	7
<u>3 Принцип рада</u> .....	9
3.1 Серијски порт.....	9
3.2 База.....	10
3.3 PIC програматор.....	11
3.4 Програм за програмирање микроконтролера.....	12
3.5 PIC Basic.....	13
<u>4 Закључак</u> .....	15
<u>5 Прилог</u> .....	16
<u>6 Коришћена литература</u> .....	25