



**УНИВЕРЗИТЕТ У НОВОМ САДУ
UNIVERSITY OF NOVI SAD
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
FACULTY OF TECHNICAL SCIENCES**



Одсек : Електротехника и рачунарство
Смер : Енергетика, електроника и телекомуникације
Усмерење : Микрорачунарска електроника

ДИПЛОМСКИ – МАСТЕР РАД

Тема:

Краткодометни пакетни радио линк

Ментор:
Др Ласло Нађ, ванр. проф.

Кандидат:
Немања Ђурић, Е10676

Нови Сад, мај 2009. године

Садржај

1.	Увод	2
2.	Анализа задатка	3
2.1.	Радио таласи	3
2.2.	Радио антене	4
2.3.	Модулација сигнала	6
2.3.1	ASK	6
2.3.2	PSK	7
2.3.3	FSK	8
3.	RF примопредајник IA4421	9
3.1.	Кратак опис кола	9
3.2.	SPI интерфејс	11
3.3.	Програмирање уређаја	12
3.4.	Избор антене	14
3.4.1	Мала луп антена	14
3.4.2	Диференцијална IFA антена	15
3.4.3	BIFA антена	16
3.4.4	Упоредни приказ особина антена	16
3.5.	Комуникациони протокол	19
4.	Хардвер уређаја	20
4.1.	Напајање	20
4.2.	Комуникација са рачунаром	21
4.2.1	RS-232 стандард	22
4.3.	Управљачки део уређаја	23
4.4.	RF део уређаја	24
4.5.	Кончан изглед PCB-а	24
5.	Софтвер уређаја	25
5.1.	Програм који управља уређајем	25
5.1.1	Wireless Development Suite	26
5.2.	Кориснички интерфејс	27
6.	Закључак	29
	Литература	30
	Прилог А: Изворни код	31
	Програм који управља уређајем	31
	Чет програм	42

1. Увод

Развој технологије је довео до великог броја интегрисаних кола која у себи обједињују функционалности многих компоненти које су се раније морале одвојено набављати и постављати на плочу. Овај напредак у технологији је довео до драстичног смањења броја компоненти потребних за реализовање коначне апликације. Тако се од 1993. до данас број компоненти у RF (*Radio Frequency*, радио фреквенција) уређајима смањио преко 8 пута, и иако је тај тренд данас знатно успорен, ипак се и даље наставља [1].

IA4421 је управо једно такво коло, способно да додатно олакша развој уређаја и даље смањи време развоја и број компоненти. Поред тога што се одликује великим бројем интегрисаних функција и релативном лакоћом кориштења, особина која га истиче је и цена, знатно нижа од конкуренције. Због свих ових карактеристика IA4421 је био први избор при реализацији рада.

Уређај описан у овом раду користи радио таласе за пренос кратких пакета информација. Користе се радио таласи који спадају у UHF (*Ultra High Frequency*, ултра високе фреквенције) део електромагнетског спектра и сигнал је модулисан FSK (*Frequency Shift Keying*, дигитална фреквенцијска модулација) модулацијом. Као терминали у комуникацији се користе рачунари, а у ту сврху је искориштен серијски порт. Компјутери комуницирају са микроконтролером, који даље комуницира са примопредајником. Детаљнија објашњења свих ових термина, као и други детаљи везани за рад су дати у наставку.

Све скраћенице у овом раду су преузете из енглеског језика и написане су латиницом, с обзиром да су фирме са енглеског говорног подручја водеће у овој области, као и да је већина литературе везане за RF написана на енглеском језику.

Рад обухвата следећа поглавља:

- у првом поглављу је дат кратак увод и преглед рада
- друго поглавље се бави самим задатком, објашњава разлоге избора RF-а за комуникацију, описује основне особине радио таласа и начине преноса података RF путем
- у трећем поглављу је дат опис интегрисаног кола IA4421, које представља срце уређаја описаног у овом дипломском-мастер раду
- у четвртном поглављу је дат опис хардвера, процес дизајна и израде, као и кратки описи компоненти које су искориштене
- у петом поглављу је дат преглед алгоритама и самог софтвера који управља уређајем, као и опис Windows програма који обезбеђује једноставан интерфејс између корисника и RF модула
- у шестом поглављу је дат закључак
- на крају рада је дат списак кориштене литературе

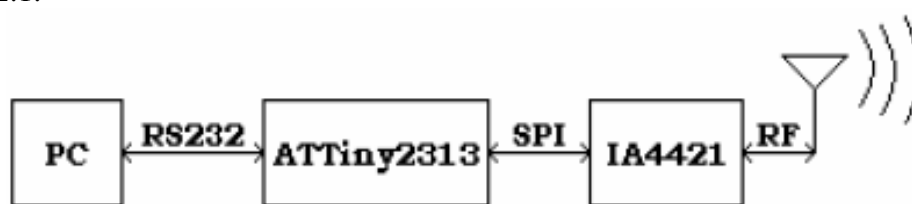
Овај рад представља резултат истраживања на тему бежичне комуникације са циљем ближег упознавања са комуникацијом помоћу RF-а, као веома распрострањеном начину повезивања. Додатни мотив за израду рада је била и чињеница да је ова веома занимљива тема ипак у другом плану на Електроници, смеру са којег је овај пројекат.

2. Анализа задатка

У времену све бржег технолошког развоја телекомуникације јесу, и биће у будућности, једна од најбитнијих области човековог деловања. Без развијених комуникационих веза ствари које узимамо здраво за готово, као што су телефон, радио, телевизија, интернет, не би постојале и начин живота на који смо навикли не би био могућ. Човек је у свом раду дошао до многих комуникационих решења, а једна од најперспективнијих области у том мноштву генијалних изума су свакако бежичне комуникације.

Бежичне комуникације су доживеле огроман развој последњих година и бежични уређаји постају све јефтинији, приступачнији и једноставнији за коришћење. Постоје многи начини преноса информација бежичним путем, као што су пренос помоћу инфрацрвеног зрачења, радио таласа, ласерског снопа, видљиве светлости или акустичне енергије. Поред тога развијени су и разни протоколи који омогућавају и олакшавају овај вид комуникације. Овај дипломски-мастер рад управо покушава да да предлог за један уређај који је заснован на једном од споменутих начина преноса информација.

Задатак који је био постављен је да се направе уређаји који ће омогућавати успостављање бежичне везе. Као прва ствар се намеће питање на који начин, од многих који се користе за бежично повезивање, ће се бежична веза остварити. Изабрана је RF технологија, технологију коју одликује велики обим доступне литературе, доступност и релативно ниска цена компоненти. Истина је да ове особине мање-више поседују и остале технологије, али RF истиче особина да линија видљивости између два примопредајника углавном није обавезан услов везе. Даље је важно одредити и како ће се тим уређајима управљати и кога ће уопште они повезивати. Логичан и доста једноставан избор је употреба PC-а (*Personal Computer*, лични рачунар) за терминале, док ће се наредбе и подаци RF модулима преносити стандардном серијском везом, помоћу протокола RS-232 (*Recommended Standard*, препоручени стандард). Оваквим резоновањем се долази до поједностављене шеме система, приказане на слици 2.1.



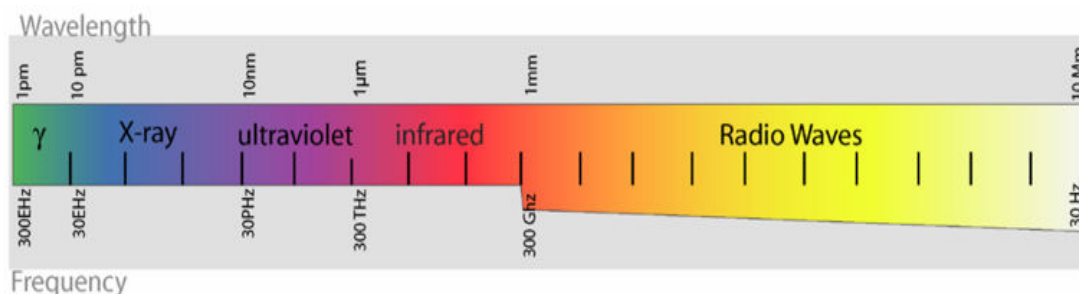
Слика 2.1 Поједностављена шема целог система

Да би се потпуно разумео начин рада RF модула морају се схватити и основне особине радио таласа, као и основни принципи њихове употребе у комуникацијама.

2.1. Радио таласи

Радио таласи су електромагнетски таласи чије место у спектру можете да видите на слици 2.2. Њихово постојање је први предвидео Џејмс Максвел (*James Maxwell*) још 1865. године, док је Хајнрих Херц (*Heinrich Hertz*) 1887. у

лабораторији доказао Максвелове тврдње. Многи изуми су следили који су омогућили искориштавање радио таласа, при чему су међу најзаслужнијим људима за развој радија и употребу радио таласа за комуникације свакако Едвин Армстронг (*Edwin Armstrong*), Никола Тесла и Ђуљелмо Маркони (*Guglielmo Marconi*), поред многих других који су допринели развоју радија [2].



Слика 2.2 Електромагнетски спектар [3]

Радио таласе описујемо таласном дужином или фреквенцијом, при чему су ове две величине једнозначно повезане преко формуле:

$$c = f * \lambda \quad (1)$$

где је:

- c – брзина светлости у вакууму ($\approx 3 * 10^8$ m/s)
- f – фреквенција [Hz]
- λ – таласна дужина [m]

Управо овде се може видети зашто су радио таласи тако подесни за употребу у комуникацијама. Наиме, да би се уопште остварила бежична веза потребна је антена која шаље до и прима поруке од других бежичних уређаја, а управо је величина антене, тачније њене димензије, тесно повезана са таласном дужином радио таласа.

2.2. Радио антене

Један од најбитнијих делова радио кола је свакако антена, која служи за примање и слање радио таласа. Антене су у суштини електрични претварачи, који претварају електричне сигнале, струју или напон, у електромагнетске таласе и обрнуто. Да би антена заједно са улазним колом проосциловала, тј. да би се детектовао валидан сигнал, димензије антене морају бити упоредиве са таласном дужином сигнала који антена детектује. Овде се види зашто су управо радио таласи искориштени у комуникацијама, пошто су њихове таласне дужине између пар милиметара до неколико метара. Антене ових димензија су биле лаке за израду чак и пре једног века када је почињало доба радија и када је технологија била на знатно нижем нивоу од садашњег.

Постоји много врста антена, које се разликују по ефикасности, цени, величини и другим параметрима важним при дизајну система. Основна подела антена се може направити према томе да ли су антене са једним крајем, небалансиране, или су диференцијалне, тј. балансиране. Како већина кола има диференцијалне антенске улазе, уколико се користи небалансирана антена

између антене и улаза се убацује трансформационо *балун* коло (*BALanced-UNbalanced*, балансирано-небалансирано), које прилагођава такву антену [4].

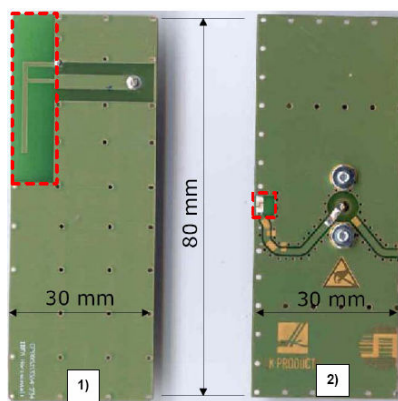
У електроници, у RF уређајима се најчешће користе 3 врсте антена, РСВ (*Printed Circuit Board*, штампана плоча) антена, чип (*chip*) антена и бич (*whip*) антена, свака са својим предностима и манама:

- РСВ антена се реализује директно на штампаној плочи, она је јефтина, могу се постићи добре перформансе, и димензије могу бити јако мале ако се ради са већим фреквенцијама, док су лоше стране употребе РСВ антена велике димензије при ниским учестаностима, као и тешкоћа самосталног дизајнирања мале и ефикасне антене, слика 2.3



Слика 2.3 РСВ антена [4]

- чип антена је знатно мања од других антена, слика 2.4, али је мана то што мало више кошта и често се боље перформансе могу постићи другим типовима антена



Слика 2.4 Упоредни приказ величине РСВ (1) и чип (2) антене [4]

- бич антена има најбоље особине, али је проблем што је гломазна и скупа, а и додатни специјални конектор на РСВ плочи такође увећава цену, слика 2.5



Слика 2.5 Бич антена [5]

После разматрања добрих и лоших особина поменутих врста антена, одлучено је да се за овај рад користи диференцијална РСВ антена. Поред мале цене и величине при високим учестаностима и смањеног броја компоненти због избегавања балун кола, пресудна је била и јако добра и опширна литература за пројектовање РСВ антена. Бројне већ дизајниране антене које су произвођачи RF примопредајника обезбедили су такође помогле при избору.

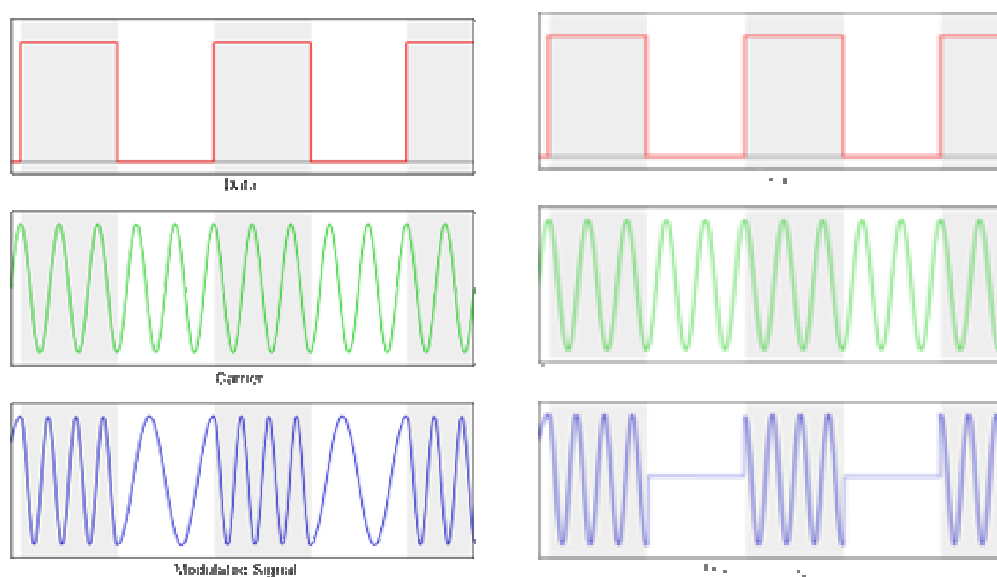
2.3. Модулација сигнала

Модулација сигнала је неизоставан корак како у жичаном тако и у бежичном преносу. Наиме, информација која нам је занимљива и коју желимо да пошаљемо најчешће није у облику који је одговарајући за пренос. Користан сигнал морамо да конвертујемо у неки други облик, тако да при преносу буде мала вероватноћа грешке, да шум и препреке не утичу знатно на сигнал. Још један разлог за модулисање сигнала је да се обезбеди истовремено кориштење већег броја канала за комуникацију. И у крајњој линији, да измењен, модулисан сигнал можемо лако и исправно детектовати на крају везе.

Модулација сигнала представља мењање параметара сигнала носиоца, који је најчешће синусоида, у зависности од модулишућег, корисног сигнала да би се тако добијени модулисани сигнал користио за пренос информација. Како синусни сигнал описујемо амплитудом, фазом и фреквенцијом, разне методе модулације су развијене које мењају управо ове параметре сигнала носиоца. Такође се разликују модулације аналогних и дигиталних сигнала, али како су информације којима се бави овај рад дигиталне, бавићемо се само модулацијом дигиталних сигнала.

2.3.1 ASK

Код ASK (*Amplitude Shift Keying*, дигитална амплитудска модулација) модулације амплитуда сигнала носиоца се мења у зависности од модулишућег сигнала, док фреквенција и фаза остају непромењене, слика 2.6.



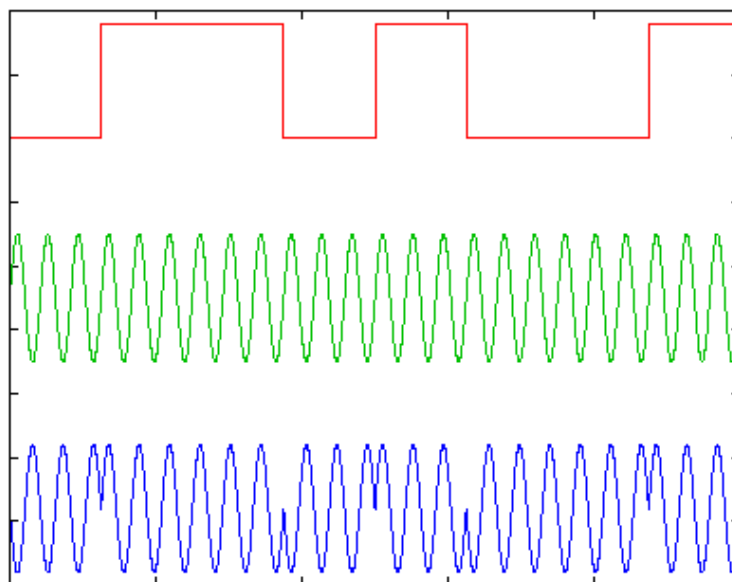
Слика 2.6 Упоредни приказ FSK и ASK, тј. OOK [3]

ASK је најсличнија АМ (*Amplitude Modulation*, амплитудска модулација) модулацији за аналогне сигнале, са разликом да се код ASK користи коначан број амплитудских нивоа за модулацију. Логичка '0' се углавном модулише мањом амплитудом, док се логичка '1' модулише већом амплитудом модулисаног сигнала. Најједноставнија варијација ове модулације је OOK (*On-Off Keying*, модулација упаљено-угашено) где одсуство сигнала одређени период времена значи логичку '0', а присуство логичку '1'.

Самим тим што је најједноставнија и најјефтинија врста модулације, ASK са собом доноси и одређене проблеме. Пошто долази до наглих промена у амплитуди, поготово код OOK, спектрална ефикасност је јако мала, и у фреквентном домену сигнал има јако пуно нежељених компоненти. Такође шум јако утиче на сигнал, суперпонирајући се и мењајући амплитуду модулисаног сигнала. Због тога исправна комуникација јако зависи од присуства шума и вредности SNR (*Signal-Noise Ratio*, однос сигнал-шум) фактора.

2.3.2 PSK

Код PSK (*Phase Shift Keying*, дигитална фазна модулација) модулације, за разлику од ASK, сигналу носиоцу се мења фаза у зависности од модулишућег сигнала, док амплитуда и фреквенција остају непромењени, слика 2.7. Овој модулацији одговара PM (*Phase Modulation*, фазна модулација) у модулацији аналогног сигнала. Најједноставнија је BPSK (*Binary PSK*, бинарна PSK), где се користе две различите фазе за модулацију, мада се потпуно равноправно користе и модулације са већим бројем фаза, као што су QPSK (*Quadrature PSK*, квадатурна PSK) и $\pi/4$ -QPSK.



Слика 2.7 PSK модулација [3]

Ова врста модулације се доста често користи. Неки стандарди за бежични LAN (*Local Area Network*, локална мрежа), нпр. IEEE 802.11b-1999 [6], као и за RFID (*RF IDentification*, идентификација помоћу RF-a) као што је ISO/IEC 14443 [7], користе PSK. Разлог је његова једноставност у односу на неке друге врсте модулација које се користе за примене где су битни велики протоци података.

Разлог због кога нису узимани у обзир примопредајници који користе PSK је просто зато што није нађено прихватљиво PSK интегрисано коло, док је понуда за ASK и FSK примопредајнике заиста огромна.

2.3.3 FSK

Код FSK модулације, за разлику од других поменутих врста модулације, сигнал носилац се модулише тако да му се у зависности од вредности модулишућег сигнала мења фреквенција у времену, док фаза и амплитуда остају непромењене, слика 2.6. Ова модулација одговара FM (*Frequency Modulation*, фреквенцијска модулација) модулацији аналогног сигнала, при чему се код FSK користи коначан број различитих фреквенција. Најједноставнији FSK је BFSK (*Binary FSK*, бинарни FSK), где се користе само две фреквенције за модулацију. Логичка '1' и логичка '0' се кодују двома различитим фреквенцијама, које се налазе једнако удаљене на обе стране од централне учестаности.

Како шум не утиче на фреквенцију модулисаног сигнала, већ на амплитуду, тако је и сигнал који је FSK модулисан отпорнији на сметње, и ређе се појављују грешке у преносу. И поред ове очигледне предности, у већини уређаја се користи OOK модулација уместо FSK. Један од разлога је велика сложеност кола која раде са FSK у односу на OOK, тј. ASK. При трансмисији података, када је бит '0' ASK модулатор не мора уопште ни да ради, ASK системи просечно троше 50% мање енергије. Даље, пошто FSK користи две различите фреквенције за пренос података, а ASK само једну, и ширина опсега је знатно већа код FSK [8].

И поред свих лоших особина које карактеришу FSK, ипак је изабран RF примопредајник који користи FSK модулацију. За овај рад најбитнија је особина FSK-а да је BER (*Bit Error Ratio*, однос лоше примљених и укупно примљених бита) мањи него код ASK модулације, тј. да је мања вероватноћа грешке при преносу [9]. Уколико би се овај уређај продавао вероватно би се користио ASK примопредајник због ниже цене компоненте и једноставније производње, што би сигурно водило и ка већој економској исплативости финалног RF производа.

3. RF примопредајник IA4421

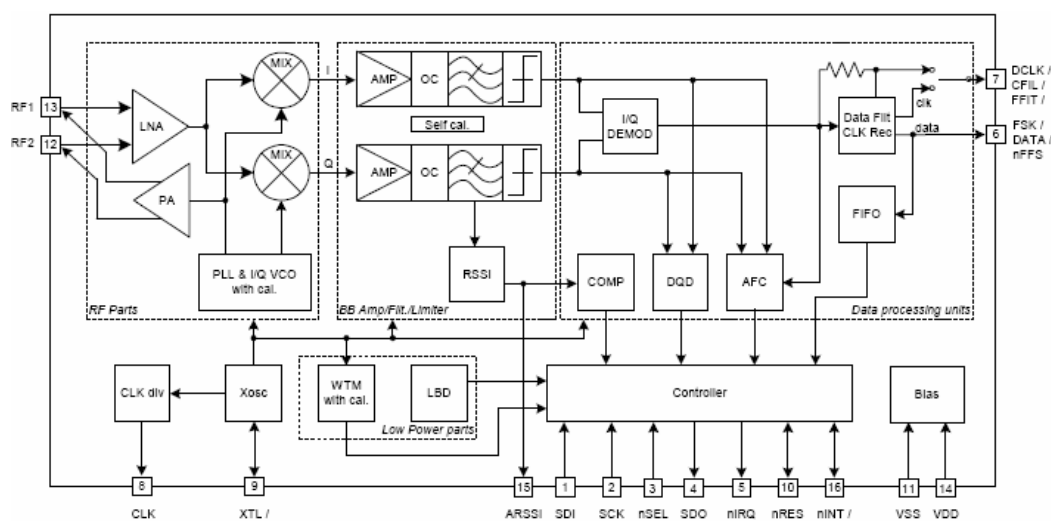
После опсежне претраге за RF примопредајницима, избор је сведен на пар интегрисаних кола која задовољавају захтеве задатка. Како овај RF пројекат ипак није са смера Телекомуникације, главни услови за коначан избор су били да је коло једноставно за употребу, да већ постоје дизајниране антене за дато коло, као и да је коло јефтино и набављиво у Србији.

Последњи захтев је само условно испуњен за сва кола, пошто се кола морају поручити преко једног од дилера електронских компоненти, а затим се чека и више од месец дана до тренутка куповине. Наравно, под условом да тај тренутак уопште и дође. Већина кола која је разматрана је доста једноставна за употребу, а и произвођачи су дали своје предлоге за антене, или су барем објаснили како најлакше доћи до коначног дизајна. На крају, долази се до закључка да је најјефтиније коло и најбоље коло, пошто су сва мање-више слична. При разматрању колико је јефтино коло, треба се у обзир узети и број и цена додатних компоненти потребних за исправан рад кола. Додатне компоненте обухватају кондензаторе, калемове и отпорнике за евентуално прилагођење улазне импедансе антене, као и осцилаторе за давање референтне учестаности.

Коло које се издвојило од свих осталих својим могућностима, као и својом ценом је RF примопредајник IA4421, произвођача Silicon Labs [10].

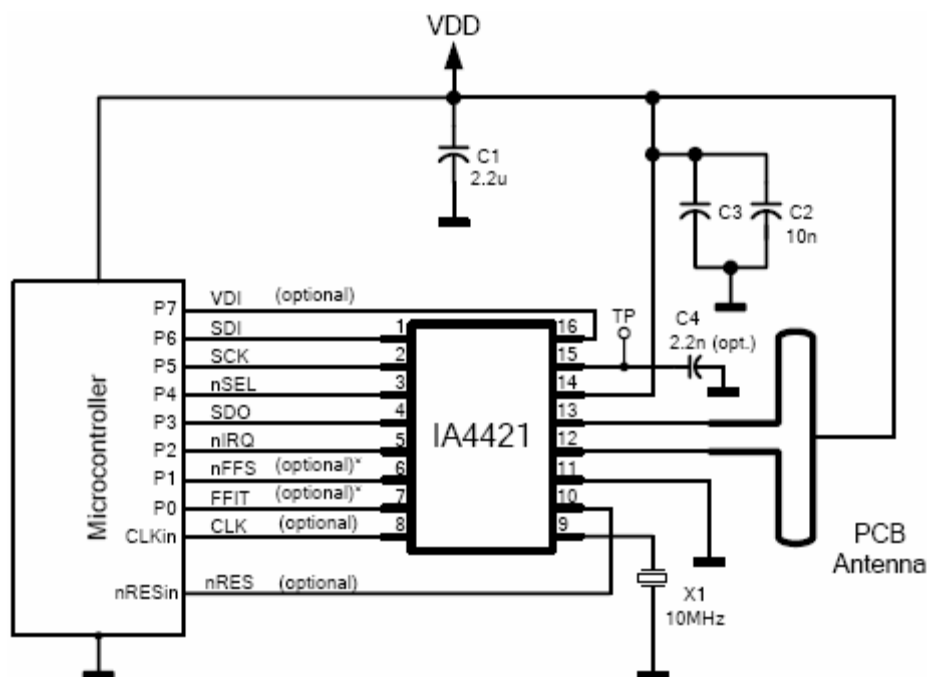
3.1. Кратак опис кола

IA4421 је производ калифорнијске фирме Integration Associates, међутим када је у јуну 2008. фирма Silicon Labs из Тексаса купила IA за 80 милиона долара коло је променило име у SI4421. Ово коло се може набавити по цени од 2.15\$ при куповини 1000 комада [11], што га чини једним од најјефтинијих RF кола у понуди, узимајући у обзир и додатне компоненте и време потребно да се развије комплетан RF уређај. Занимљиво је да многе фирме копирају IA4421, и да се функционално исти чип може пронаћи под разним именима. Тако се уместо IA4421 равноправно може узети TRC102 америчког произвођача RFM, или RF12 кинеске фирме Норе, који су чак јефтинији од оригинала.



Слика 3.1 Блок шема IA4421 [12]

IA4421 је FSK примопредајник са програмабилним фреквенцијским опсегом од 433, 868 и 915 MHz. Захтева напон напајања од 3V. Има велики број интегрисаних функција које смањују број додатних компоненти и олакшавају дизајн. Интегрисане RF функције укључују LNA (*Low Noise Amplifier*, нискошумни појачавач) и PA (*Power Amplifier*, појачавач снаге), IQ демодулатор, PLL (*Phase-Lock Loop*, петља за закључавање фазе), AFC (*Automatic Frequency Control*, аутоматска контрола фреквенције), 16-битни Rx (*Receive*, пријем) FIFO (*First In First Out*, први излази који је први ушао) регистар и два 8-битна Tx (*Transmit*, слање) регистра за податке, слика 3.1. Захваљујући томе за коначну апликацију, поред самог кола, потребни су још само јефтини кристални осцилатор и антена, слика 3.2.

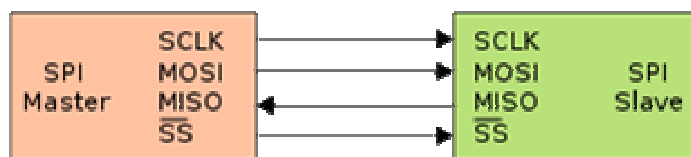


Слика 3.2 Типична апликација са колом IA4421 [12]

За програмирање кола и комуникацију за мастер колом, у овом раду се користи микроконтролер ATtiny2313, на располагању је SPI (*Serial Peripheral Interface*, серијски периферијски интерфејс) интерфејс. Поред SPI постоји и велики број излазних линија које служе да обавесте мастер коло о битним догађајима, као што су крај трансмисије или нова порука. Постоји могућност и давања промењивог CLK (*CLocK*, сат) сигнала мастеру, што смањује број кристалних осцилатора на плочи.

3.2. SPI интерфејс

SPI интерфејс се користи за комуницирање између микроконтролера и примопредајника. SPI интерфејс је *de facto* стандард, што значи да није званичан стандард. Прва је почела да га користи компанија Motorola. Подаци се преносе синхронно и серијски, и може да ради у *full-duplex* начину рада, тј. истовремено слање и примање података. Овај интерфејс је јако популаран јер користи знатно мање линија за комуникацију него паралелни интерфејси, а када се пореди са другим интерфејсима који користе исти или мањи број линија постиже веће брзине протока података. Подаци могу да се преносе брзинама од неколико Mbs (мегабити по секунди) [13]. Подаци се не могу преносити на великим раздаљинама због превеликог шума који ће тада онемогућити комуникацију.



Слика 3.3 SPI интерфејс [3]

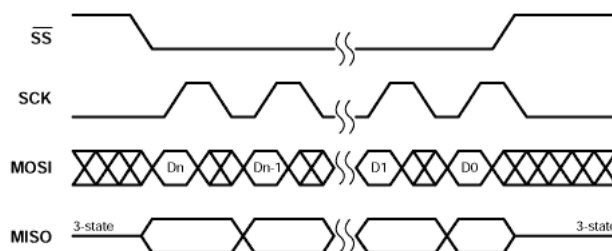
У овом интерфејсу се разликују један мастер (*master*, господар) и један или више слејвова (*slave*, роб), слика 3.3. Интерфејс се још назива и четворожична серијска магистрала, јер се користе четири линије за остварење комуникације. Те четири линије су SCLK (*Serial CLoK*, серијски клок), MOSI (*Master Out Slave In*, мастер излаз, слејв улаз), MISO (*Master In Slave Out*, мастер улаз, слејв излаз) и SS (*Slave Select*, избор слејва), која служи да се одреди којем од слејвова је намењена порука. Преко MOSI линије мастер шаље податке слејву, док преко MISO линије слејв одговара и шаље податке назад мастер колу.

Два параметра, CPOL (*Clock POLarity*, поларитет клока) и CPHA (*Clock PHase*, фаза клока), у потпуности одређују како ће се слати подаци између мастера и слејва. CPOL одређује које је основно стање SCLK сигнала, да ли је на логичкој јединици или на логичкој нули. CPHA одређује да ли се подаци узимају са MOSI линије са прве или са друге ивице клока, као и да ли се подаци на MISO линије мењају на првој или на другој ивици клока. У табели 3.1 су приказане све четири комбинације ових параметара. Код IA4421 оба параметра су једнака '0'.

Табела 3.1 Комбинације параметара CPOL и CPHA

CPOL	CPHA	Опис
0	0	Основно стање SCLK сигнала је '0', подаци са MOSI линије се узимају на растућу ивицу
0	1	Основно стање SCLK сигнала је '0', подаци са MOSI линије се узимају на опадајућу ивицу
1	0	Основно стање SCLK сигнала је '1', подаци са MOSI линије се узимају на опадајућу ивицу
1	1	Основно стање SCLK сигнала је '1', подаци са MOSI линије се узимају на растућу ивицу

Како SPI није званичан стандард, вредности нивоа напона за логичке јединице и нуле нису дефинисане. Поред тога, ни број бита који се преносе, тј. дужина податка који се преноси није дефинисана. Линије се често називају разним именима, тако да се на мастеру MOSI може наћи као SDO, а MISO као SDI. Такође не постоји начин да мастер зна да ли уопште има некога на другој страни везе или прича у празно. Поред свега тога, због своје једноставности, релативно малог броја линија, као и велике брзине протока, SPI се јако често користи за комуникацију на кратким растојањима.



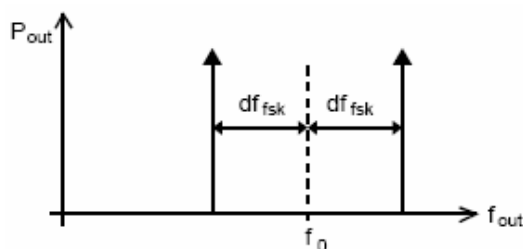
Слика 3.4 Временски дијаграм SPI комуникације за IA4421 [12]

Комуникација отпочиње спуштањем линије SS, која је активна на логичкој нули. Мастер затим мења SCLK између логичке јединице и нуле док год траје комуникација. Подаци се уносе у слејв на растућу ивицу серијског блока, док се на MISO линији мењају на опадајућу ивицу. Комуникација се најзад прекида када SS линија скочи на логичку јединицу, слика 3.4.

3.3. Програмирање уређаја

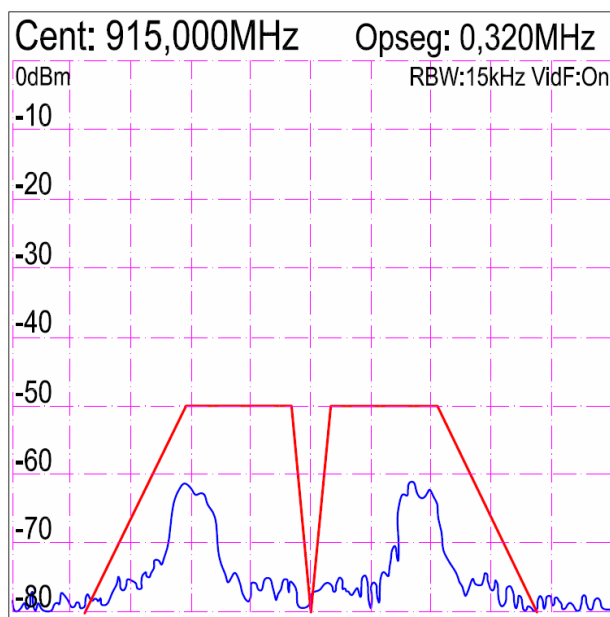
При комуникацији са IA4421 одједном се шаље или прима 16 бита. Код слања наредби први бајт је команда, док други бајт представља параметар команде. Постоји укупно 17 наредби које могу да се пошаљу интегрисаном колу IA4421 преко микроконтролера и које се користе за пажљиво програмирање уређаја. Практично сваки део IA4421 је доступан и подешљив. На располагању су следеће команде:

- **опште подешавање:** измена фреквенцијског опсега, контрола Rx и Tx регистара
- **контрола фреквенције:** подешавање централне фреквенције f_0 за FSK модулацију
- **контрола предајника:** подешавање излазне снаге P_{out} и фреквенцијске девијације df_{fsk} у дискретним корацима од по 15 kHz, слике 3.5 и 3.6



Слика 3.5 Подешавање предајника [12]

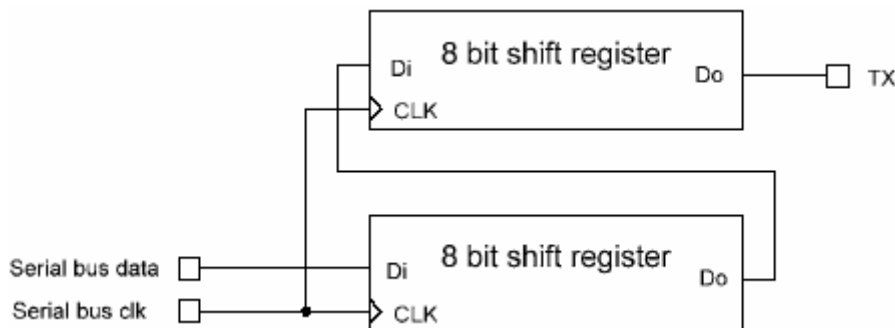
- **контрола потрошње:** укључење и искључење делова кола за слање и примање порука, као и сваког дела приказаног на слици 3.1 понаособ, а у циљу максималног смањења потрошње кола
- **контрола брзине протока:** подешавање брзине протока трансмисије
- **контрола пријемника:** подешавање ширине фреквенцијског опсега улазног филтра (равна црвена линија на слици 3.6), појачања LNA, као и понашања интегрисаних функција VDI (*Valid Data Indicator*, индикатор исправног податка) и RSSI (*Received Signal Strength Indicator*, индикатор снаге долазног сигнала) у зависности од долазног сигнала; помоћу ових функција IA4421 се може искористити и за пријем ASK модулисаног сигнала



Слика 3.6 Фреквенцијски спектар који генерише порука '010101...' [14]

- **контрола филтера података:** подешавање CR-а (*Clock Recovery*, екстракција клока из корисног сигнала), филтра за податке (дигитални или аналогни), као и DQD (*Data Quality Detector*, детектор квалитета података), који служи за детектовање пријема исправног FSK модулисаног сигнала
- **контрола ресета и FIFO регистра:** подешавање FIFO регистра, величине и услова за почетак пуњења истог, тј. подешавање дужине синхронизационе шеме, као и подешавање услова за ресет (*reset*, поновна иницијализација) уређаја
- **контрола синхронизационе шеме:** измена синхронизационе шеме, што се може искористити као јединствена адреса појединачног уређаја
- **команда за читање FIFO регистра:** после ове команде преко MOSI линије се преузима садржај FIFO регистра
- **контрола AFC:** подешавање AFC (*Automatic Frequency Control*, аутоматска контрола фреквенције), који служи за изједначавање Rx и Tx фреквенције тако што се мери и надомешћује разлика између њих; без овог кола Rx и Tx фреквенција би морале бити савршено усклађене и морао би да се користи скуп, веома прецизни осцилатор

- **контрола PLL-а:** подешавање параметара PLL
- **команда за промену вредности Tx регистра:** упис бајта намењеног за слање у први Tx регистар, почетна вредност у оба регистра је 0xAA, слика 3.7



Слика 3.7 Веза два 8-битна Tx регистра [12]

- **контрола бројача за буђење:** подешавање временског периода после кога се дешава буђење (*wake-up*) уређаја
- **команда за рад у економичном режиму:** стављање уређаја у режим рада у коме је успаван, али се повремено буди и проверава да ли има долазећих порука
- **контрола детектора празне батерије и излазног CLK сигнала за мастер коло:** подешавање напонског нивоа напајања за који се мастер обавештава да се батерија испразнила и подешавање фреквенције CLK сигнала који за рад користи мастер коло, у распону од 1 до 10 MHz
- **провера статуса уређаја:** после ове команде преко MOSI линије се чита статусна реч, која садржи стања свих регистара и индикатора

3.4. Избор антене

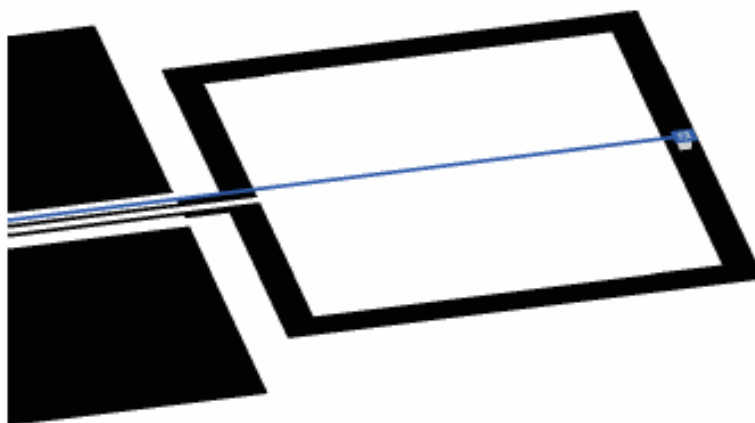
Већ је речено да је за антену изабрана РСВ диференцијална антена. Како постоји велики број таквих антена, избор треба додатно да се сузи. Срећом, произвођач примопредајника је обезбедио велики број готових предлога за овакве антене и смерница за дизајнирање [15][16][17], пошто је самостално дизајнирање антена јако тежак и захтеван посао.

3.4.1 Мала луп антена

Због својих малих димензија и високе импедансе, луп (*loop*, петља) антена је погодна за намену где је потребна јако мала снага и где је величина од интереса. Назив мала луп антена потиче од тога што је обим ове антене мањи од $\frac{1}{4}$ таласне дужине радио таласа који се прима. Луп антена има малу величину, висок фактор добротe Q (између 20 и 60, у зависности од величине и фреквенције), и средњег радијационог појачања (око -10 dB) који зависи од величине. Због своје јако велике импедансе (4-8 k Ω при резонантној учестаности) луп антена захтева малу струју.

Антиена је прилично неосетљива на близину људског тела и на проблеме са паразитним капацитивностима који из тога произилазе. Луп антиена је идеална за краткодометне уређаје напајане батеријом, као што су даљински управљачи.

Због високог Q , антиена је осетљива на раздешавање проузроковано близином металних објеката или варијацијама температуре. Често кориштено решење за овај проблем је смањење Q додавањем отпорника од неколико стотина ома паралелно са антиеном. Ипак, како је отпорност отпорника знатно мања од импедансе антиене, велики део излазне снаге се дисипира преко отпорника уместо да се преноси преко антиене што је велики проблем. Поред тога, додатни отпорник представља и додатни трошак.

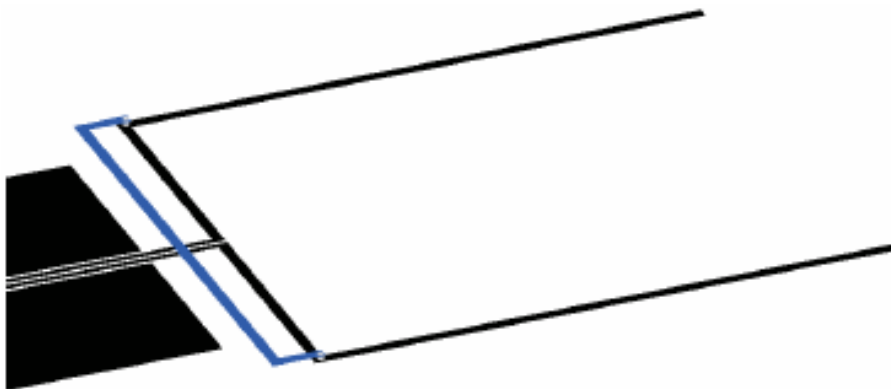


Слика 3.8 Мала луп антиена [16]

На слици 3.8 је приказана мала луп антиена. Танка веза на оси симетрије, повезана преко вије са антиеном служи за DC (*Direct Current*, једносмерна струја) поларизацију Tx излаза, који су отворени колектори.

3.4.2 Диференцијална IFA антиена

Диференцијална IFA (*Inverted F Antenna*, инвертовано F антиена) антиена се добија из небалансиране IFA антиене пресликавањем на GND (*Ground*, маса) област, слика 3.9.



Слика 3.9 Диференцијална IFA антиена [16]

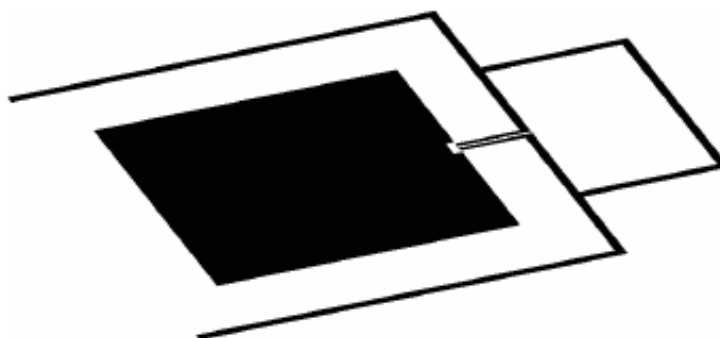
IFA антиена има велику импедансу, иако мању од луп антиене. Због тога се може искористити већа излазна струја при истом напону напајања, и може се

достигнути већа излазна снага. Кориштењем исправне геометрије, може се постићи индуктивна улазна импеданса, а самим тим и резонанција са излазном капацитивношћу примопредајника. Ефикасност радијације је знатно већа у поређењу са луп антеном. Q је мањи од Q луп антене, и због тога је антена мање осетљива на раздешавање. Ипак, у случају краткодометних апликација ($\sim 10\text{m}$) где се не захтева велика снага, ефикасност примопредајника са антенном са већом импедансом је већа. Деградација ефикасности примопредајника се надомешћује бољим радијационим појачањем ($G = 0\text{-}1\text{ dB}$), ефикасношћу антене на вишим учестаностима (10 dB већа него код луп антене), као и бољим потискивањем хармоника ($5\text{-}10\text{ dB}$ боље него код луп антене).

Највећи недостатак ове антене су огромне димензије у поређењу са луп антенном. Стога је IFA антена најбољи избор на вишим фреквенцијама (868 и 915 MHz), где ниже таласне дужине омогућавају већу ефикасност са прихватљивим димензијама антене.

3.4.3 BIFA антена

BIFA (*Back IFA*, заокренута уназад IFA) антена је модификована диференцијална IFA антена, слика 3.10.



Слика 3.10 BIFA антена [16]

У применама где је потребна мала снага, ефикасност трансмисије се може побољшати повећавањем улазне импедансе IFA антене. Мали пораст импедансе се може постићи смањивањем димензије петље.

Димензије коначне апликације се могу смањити постављањем хоризонталних линија, “руку” IFA антене око остатка кола. Остатак кола се може посматрати као велика GND област, и због тога су краће руке довољне за подешавање импедансе антене, која је осетљива на дужину руку.

3.4.4 Упоредни приказ особина антена

Да би се могле упоређивати антене, мора се дефинисати EIRP (*Equivalent Isotropic Radiated Power*, еквивалентна изотропна одаслана снага). EIRP се добија као аритметички производ снаге предајника и појачања антене. EIRP је снага коју би одавала хипотетичка изотропна антена тако да је њен интензитет радијације једнак максимуму интензитета радијације разматране антене. Изотропна антена је теоретски извор таласа који одашиље таласе потпуно

равномерно у свим правцима. Укратко, уколико једна антена има већи EIRP од неке друге, значи да та антена одашиље већу снагу у потребном правцу, тј. већа је усмереност антене. У табели 3.2 је приказан EIRP за описане антене.

Табела 3.2 EIRP за луп и БИФА антenu [16]

ERP _{max} [dBm]	Antenna type	
	Loop	BIFA
915 MHz	-15.3 (-3dB state)	-1.2
868 MHz	-20 (-3dB state)	-1.6
434 MHz	-18.3 (-6dB state)	N/A
315 MHz	-20.4 (-6dB state)	N/A

Уколико познајемо EIRP можемо израчунати и остале параметре битне за радио везу. Из EIRP (у ватима) можемо израчунати густину снаге S на датој раздаљини d :

$$S = \frac{EIRP}{4\pi d^2} \left[\frac{W}{m^2} \right] \quad (1)$$

Ефективна вредност снаге електричног поља E се може израчунати помоћу формуле:

$$E = \sqrt{120\pi S} \left[\frac{V}{m^2} \right] \quad (2)$$

Из једначина (1) и (2) може се одредити домет ако се познаје EIRP предајника и E у близини пријемне антене:

$$d = \frac{\sqrt{30EIRP}}{E} [m] \quad (3)$$

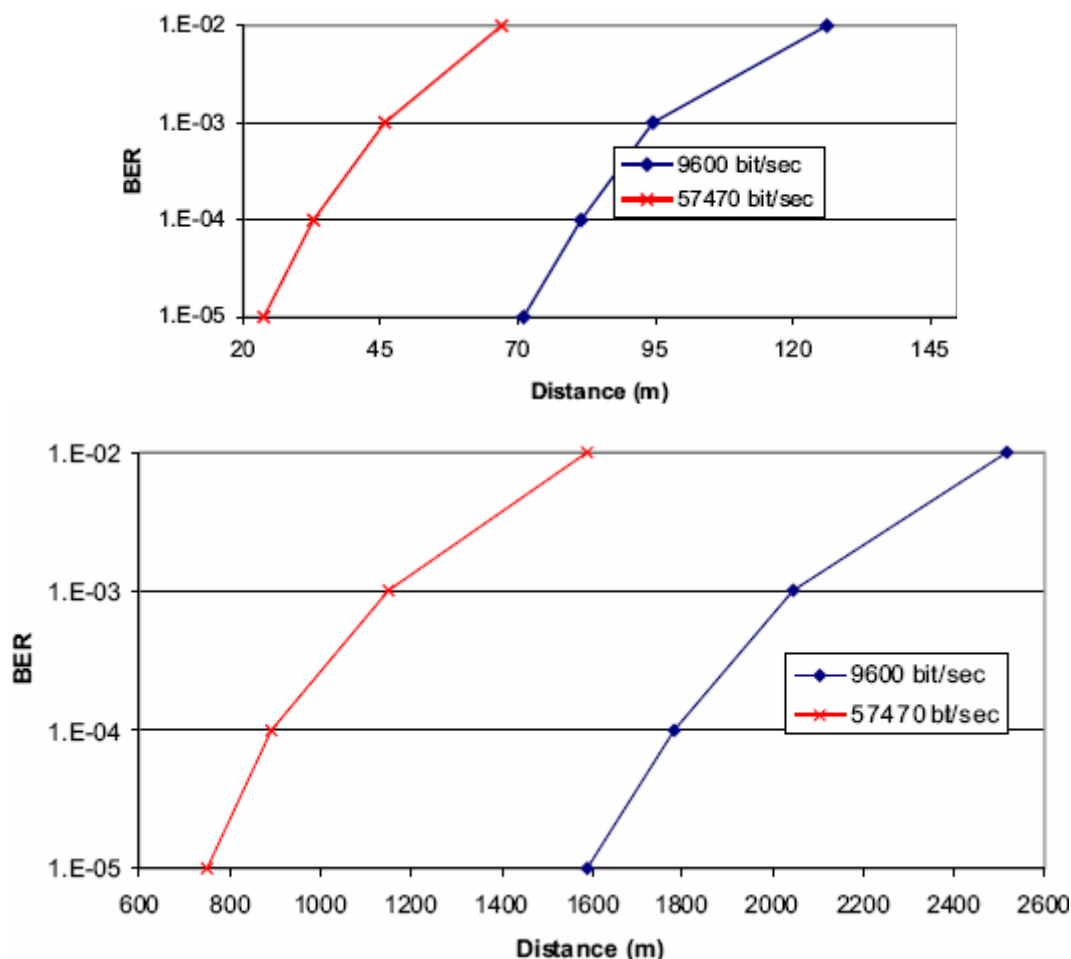
Антене можемо поредити и према томе колико E је потребно да буде у близини пријемника да би се постигла одређена вредност BER-а, табела 3.3.

Табела 3.3 Потребна вредност E при 915 MHz да би се постигле разне вредности BER-а при јаким сметњама, приказане за две различите брзине протока [16]

E _{min} [mV/m] _{r.m.s.} 915 MHz	RX Antenna type			
	Loop		Back IFA	
BER	9600 bps	57470 bps	9600 bps	57470 bps
10 ⁻²	0.24	0.44	0.06	0.09
10 ⁻³	0.31	0.65	0.07	0.13
10 ⁻⁴	0.37	0.91	0.08	0.17
10 ⁻⁵	0.42	1.25	0.09	0.2

Из табеле 3.3 се види да је BIFA антена знатно осетљивија и да се поузданији пренос остварује и при знатно слабијем електричном пољу.

Вредност BER-а се мења са повећањем растојања између предајника и пријемника. Како се растојање повећава сигнал слаби и BER опада, тј. све је већа вероватноћа да ће се сигнал погрешно растумачити. На слици 3.11 је приказана зависност BER-а од растојања за парове пријемних и предајних антена луп-луп и BIFA-BIFA на 915 MHz, за две различите брзине преноса.

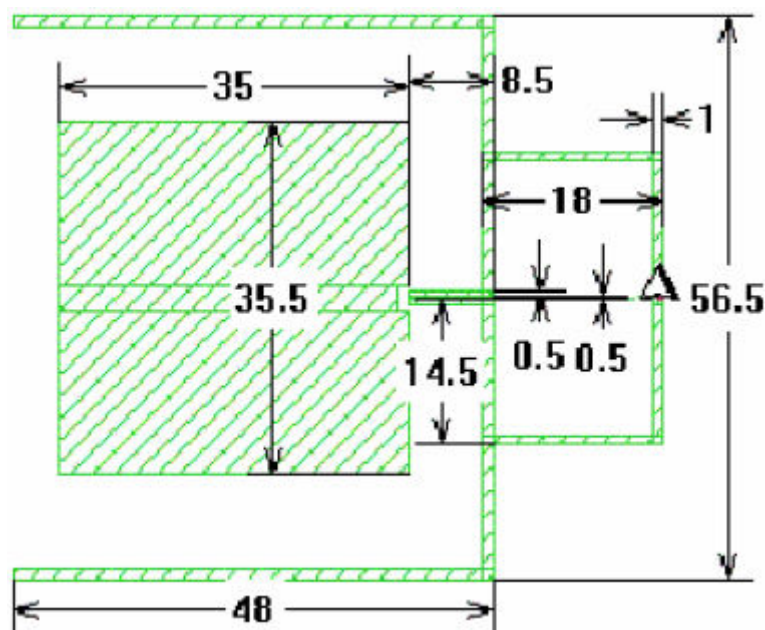


Слика 3.11 Зависност BER-а од растојања за луп-луп (горе) и BIFA-BIFA (доле) [16]

Јасно се види да је BIFA-BIFA антенски пар знатно поузданији и да се могу постићи већи домети него са луп-луп антенским паром.

После свега реченог јасно је да је BIFA антена бољи избор на вишим учестаностима. Не само што на овим учестаностима димензије антене постају прихватљиве за већину апликација, већ је и радио веза знатно поузданија и при већим раздаљинама и брзинама протока.

BIFA антена за фреквенцијски опсег од 915 MHz са детаљно приказаним димензијама, предложена од стране произвођача, је приказана на слици 3.12. Остале антене се могу пронаћи у литератури [15] и [17].



Слика 3.12 BIFA антена за фреквенцијски опсег од 915 MHz (димензије су у mm) [15]

3.5. Комуникациони протокол

При слању пакета мора се пратити одређени протокол да би се на пријемној страни успешно примила порука. На слици 3.13 је приказан исправан начин слања пакета.



Слика 3.13 Редослед слања бајтова [12]

Прво се шаљу један или два бајта наизменично '1' и '0'. За ову преамбулу се може искористити почетна вредност Tx регистра, 0xAA. Ово омогућава примопредајнику да детектује пријем FSK модулисаног сигнала. За ову сврху се користе DQD (*Data Quality Detector*, детектор квалитета података) и DRSSI (*Digital RSSI*, дигитални RSSI). Када ова два сигнала детектују валидан сигнал, пријемник проверава да ли тај сигнал стиже предвиђеном брзином преноса, и мења се VDI (*Valid Data Indicator*, индикатор исправног податка) сигнал у зависности од резултата провере. Затим се шаље синхронизациона шема дужине два бајта, при чему је први бајт програмабилан. Ово може да се искористи за адресирање појединих примопредајника, уколико у комуникацији учествује више од два примопредајника. Када се прими одговарајућа шема, а уз то сигнал VDI показује да је примљен исправан сигнал, почиње да се пуни пријемни FIFO регистар долазећим битима.

4. Хардвер уређаја

Хардвер уређаја чине следећи делови:

- део за напајање, са стабилизатором напона LM317
- комуникациона веза са рачунаром, са прилагођењем напонских нивоа помоћу MAX3232
- управљачки део, са микроконтролером ATtiny2313
- RF део, са примопредајником IA4421

Цео уређај је реализован на једној плочици. Најбољи резултати се постижу уколико се користи плочица дебљине 0.5mm, направљена од материјала који има $\epsilon_r = 4.7$. На плочици се налазе два конектора, конектор за серијску везу са рачунаром, као и конектор за довођење једносмерног нестабилисаног напона, који не сме бити мањи од 6.2V. Физичке границе РСВ-а су минимално 2cm од ивица РСВ антене [6].

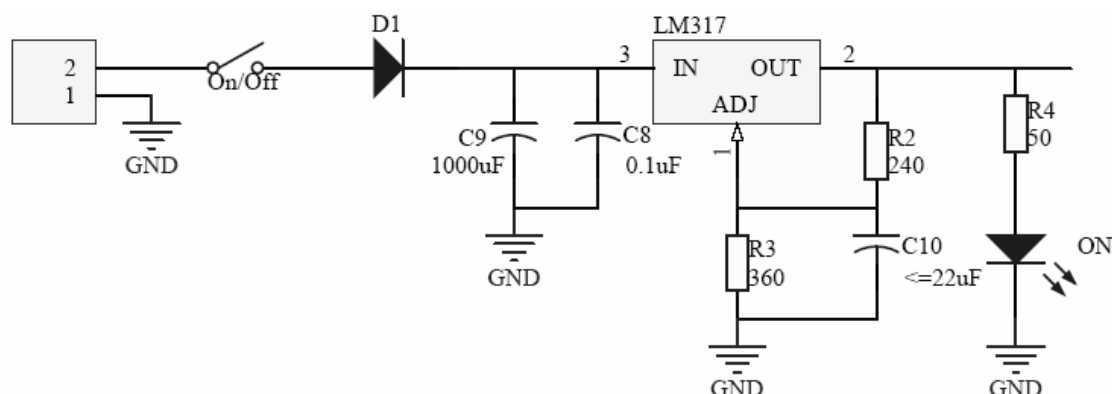
Уређај не сме бити затворен у металном кућишту, пошто би то знатно смањило могућност исправног слања и примања порука.

4.1. Напајање

За напајање уређаја се користи извор једносмерног нестабилисаног напона (адаптер) који даје 13V на излазу, што се даље води на улаз подесивог извора стабилсаног напона LM317, слика 4.1. Напон се подешава помоћу отпорника R_2 и R_3 , по формули:

$$V_{out} = 1.25 * (1 + R_3 / R_2) + I_{adj} * R_3 \quad (1)$$

Струја I_{adj} је реда 50μA. На основу једначине (1) смо изабрали отпорнике R_2 и R_3 тако да LM317 на излазу даје 3.125V, пошто сва искориштена интегрисана кола захтевају напајање од 3V.



Слика 4.1 Напајање уређаја

Користе се још и конектор за адаптер, прекидач за паљење и гашење уређаја, заштитна диода D1 која штити од погрешног повезивања напајања, као и LED (*Light Emitting Diode*, светлећа диода) диода као индикација да је уређај упаљен.

4.2. Комуникација са рачунаром

За остварење серијске везе између компјутера и микроконтролера се користи конвертор напонских нивоа MAX3232.

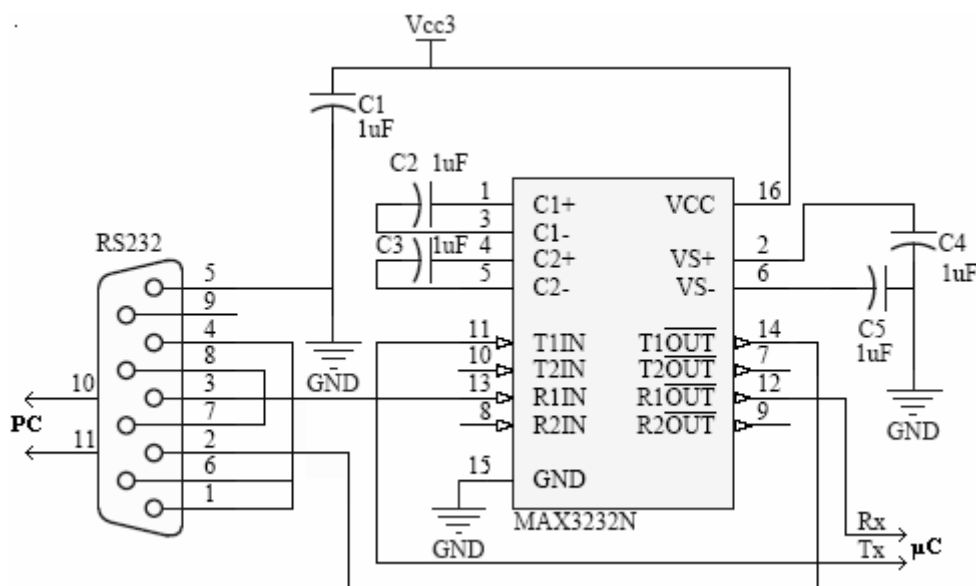
Код RS-232 интерфејса подаци се преносе преко пара проводника. У циљу минимизирања проблема везаних за шум и сметње користи се широк напонски опсег сигнала, за логичку нулу од 5V до 15V, а за логичку јединицу од -15V до -5V. При томе је номинални напонски ниво за логичку јединицу -12V, а ниво од 12V за логичку нулу, али ће пријемник прихватити све нивое од -25V до -3V и од 3V до 25V.

Овде настаје проблем јер ови нивои нису компатибилни са стандардним CMOS напонским нивоима, табела 4.1. Због овог неслагања напонских нивоа, да би се примили подаци преко RS-232 интерфејса потребно је смањити и инвертовати напоне. Када се шаљу подаци преко RS-232, опет је потребно инвертовати и повећати напоне.

Табела 4.1 Поређење RS-232 и CMOS напонских нивоа

RS-232		CMOS	Логика	
-15V ... -5V	<->	+3.5V...+5V	<->	1
+5V ... +15V	<->	0V ... +1.5V	<->	0

MAX3232 je integrisano kolo koje resava gore naznaceni problem. Ovo kolo sadrzi dva drajvera i dva prijemnika za prilagođenje naponskih nivoa. Jako je popularno zbog potrebe za samo jednim naponom napajanja i jer generiše RS-232 naponske nivoe internu. Mora se napomenuti da MAX3232 nije zaduzen za sinhronizaciju niti za dekodovanje RS-232 signala. Njegova jedina uloga je konverzija naponskih nivoa signala. Na slici 4.2 je prikazano okruženje integrisanog kola MAX3232.

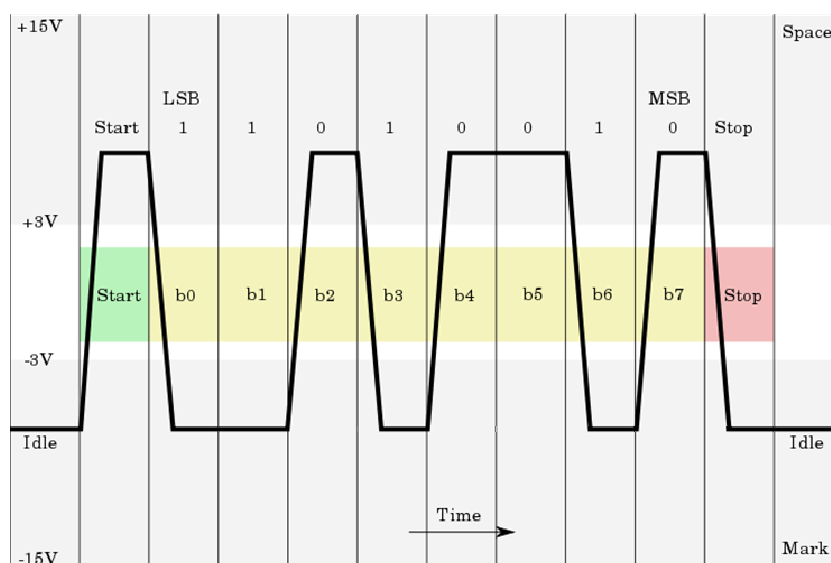


Слика 4.2 Начин повезивања конвертора нивоа MAX3232

4.2.1 RS-232 стандард

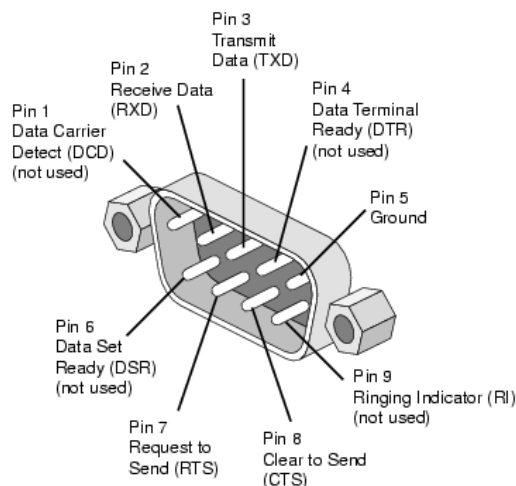
Један од најзначајнијих механизма микроконтролера је серијска комуникација по RS-232 стандарду [18]. Како се пријем и слање података обављају одвојеним линијама могућ је *full-duplex* начин рада.

Пренос отпочиње START битом (логичка '0'), за којим иде податак дужине од 5 до 9 бита, иако се скоро увек користи 8-битна реч. За разлику од спомињане SPI комуникације, прво се шаље LSB (*Least Significant Bit*, бит најмање вредности) бит, па онда даље све до MSB (*Most Significant Bit*, бит највеће вредности) бита. Затим може да иде бит парности, који служи за детектовање лоше примљених података. Пренос се завршава слањем једног или два STOP бита, који су такође логичке нуле, слика 4.3.



Слика 4.3 ASCII карактер 'K' (0x4b) пренет помоћу RS-232 стандарда [3]

Брзина преноса се мери баудима (*baud*), названим по француском инжењеру Емилу Бауду (*Émile Baudot*), и означава број пренетих симбола у секунди. У овом случају, код RS-232 стандарда, то је једнако броју пренетих бита по секунди, пошто један симбол означава само један бит, што не мора увек бити случај. Конектор DB-9 за серијски порт је приказан на слици 4.4.

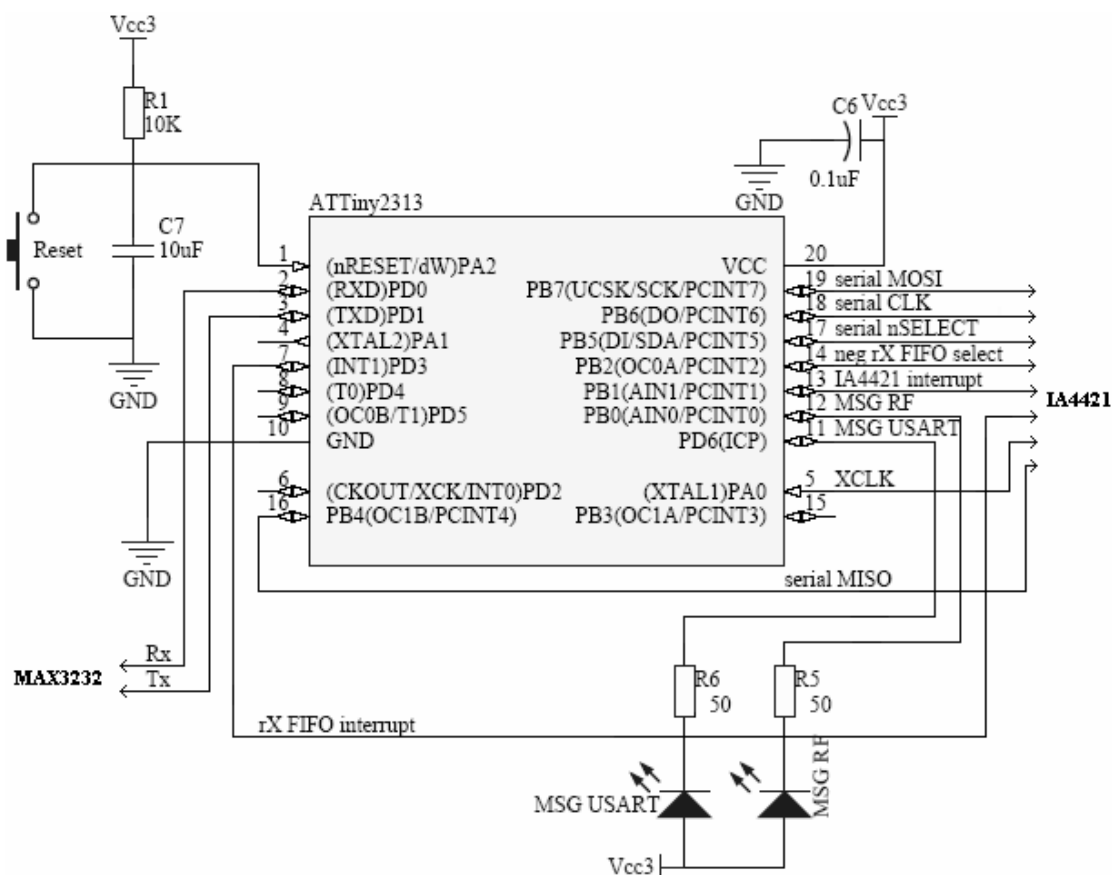


Слика 4.4 DB-9 конектор за серијски порт [3]

4.3. Управљачки део уређаја

За управљање се користи микроконтролер Atmel ATTiny2313. Ово коло је изабрано јер има уграђену подршку за серијску комуникацију и за SPI комуникацију. Поред тога, као и сва друга искориштена интегрисана кола, захтева напон напајања од 3V, што значи да се може искористити само један стабилизатор напона.

Од додатних кола убачено је ресет коло, које враћа уређај у почетно стање, затим конектор за ISP (*In-System Programming*, програмирање у систему) програмирање, као и две сигналне LED диоде које се користе као индикатори битних догађаја, слања PING поруке и пријем туђе PING поруке, слика 4.5.

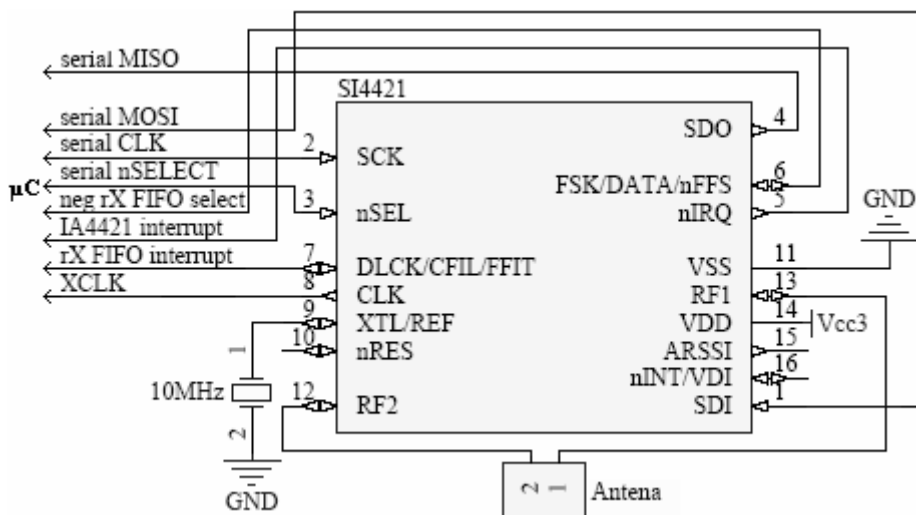


Слика 4.5 Управљачки део уређаја

За програмирање микроконтролера је искориштен бесплатан програм AvrOspII [19], док се шема ISP програматора може наћи у [20] и [21].

4.4. RF deo ypehaja

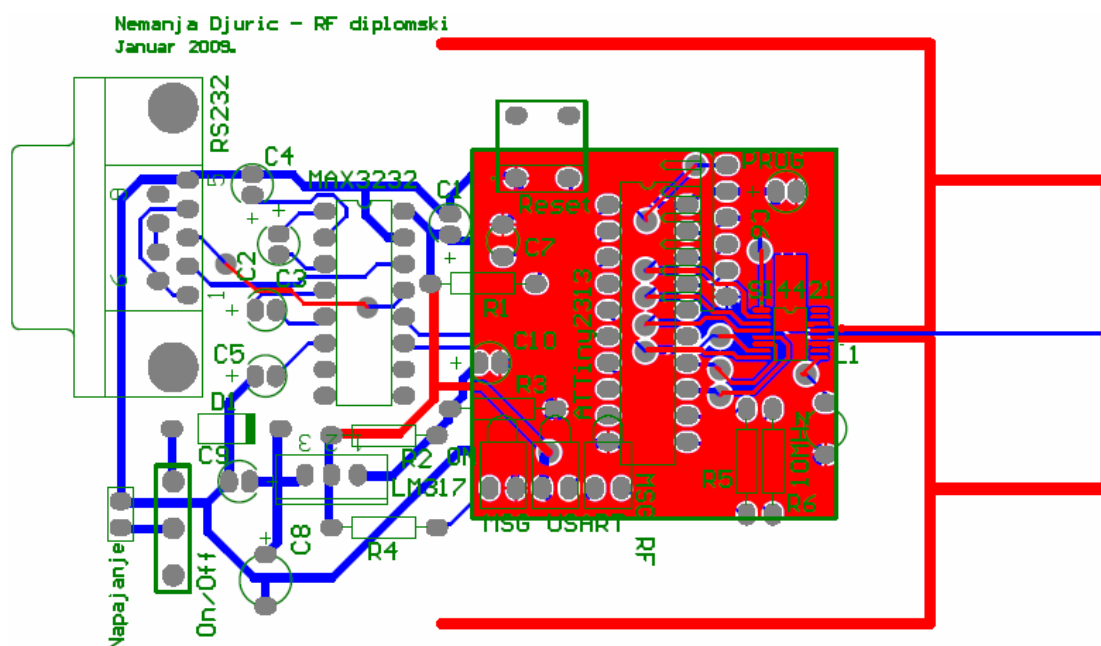
Најбитнија компонента RF дела је примопредајник IA4421, описан у трећем поглављу. Веза IA4421 са остатком кола је приказана на слици 4.6.



Слика 4.6 Окружење IA4421 примопредајника

4.5. Коначан изглед РСВ-а

PCB је урађен двострано, и konačni izgled је приказан на слици 4.7. При дизајнирању су праћени упуту произвођача који се тичу изгледа PCB плочице [14][16]. Металне везе између ножица IA4421 с једне стране, и крајева антене и кристалног осцилатора са друге су направљене тако да буду што је могуће краће. Поред тога и димензије антене и GND области, као и одстојање од краја антене до физичке ивице PCB-а су пажљиво подешени.



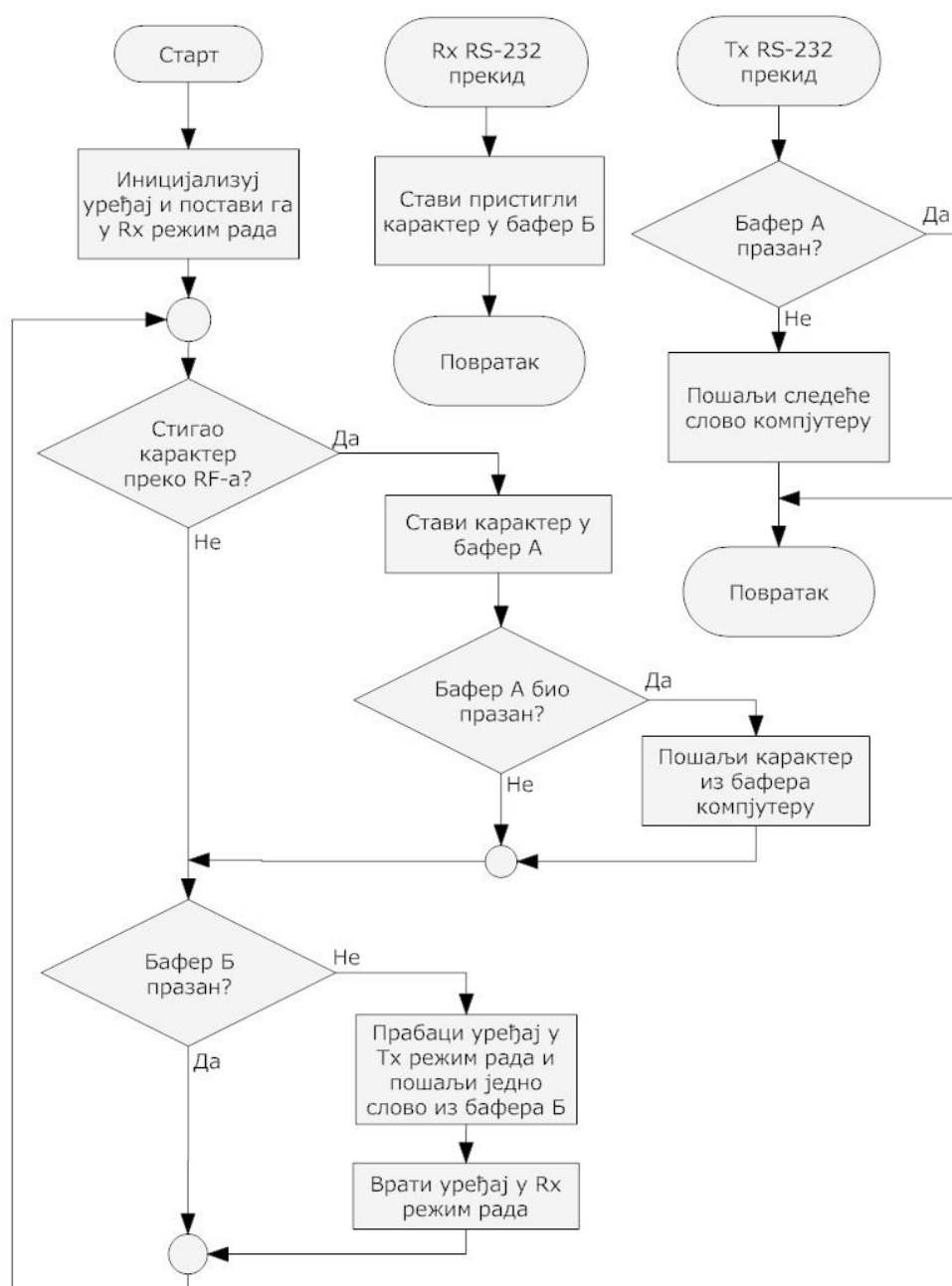
Слика 4.7 Коначан изглед РСВ-а

5. Софтвер уређаја

Написана су два програма за потребе радио линка. Програм који управља самим уређајем писан у С окружењу програма AVR Studio, и програм који представља интерфејс између корисника и уређаја, писан у програму Visual Basic и који је замишљен као чет (*chat*, разговор) програм. Комплетан изворни код (*source code*) се налази у Прилогу А.

5.1. Програм који управља уређајем

Алгоритам на основу којег је програмиран уређај је дат на слици 5.1.



Слика 5.1 Алгоритам рада уређаја

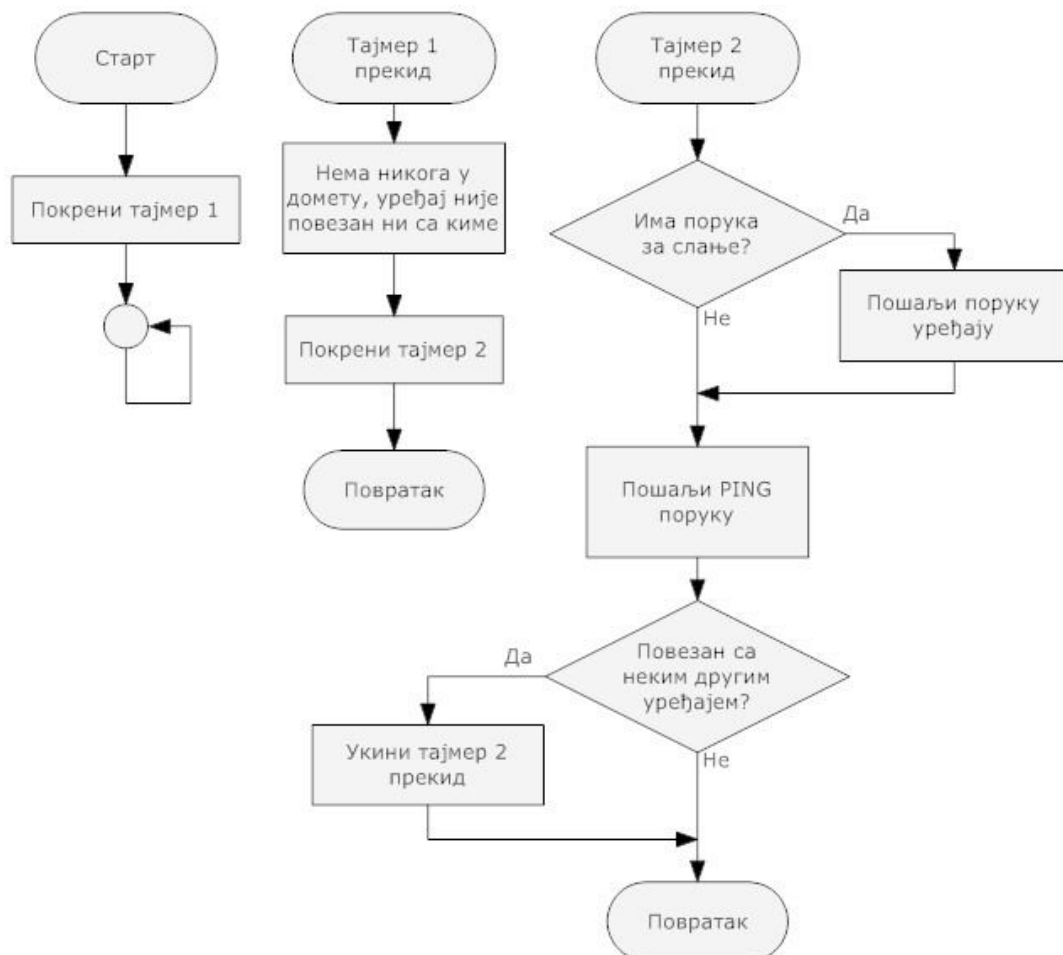
5.1.1 Wireless Development Suite

Као помоћ при програмирању IA4421 се може искористити бесплатни програм WDS (*Wireless Development Suite*), који се може наћи на интернет сајту компаније Silicon Labs [10]. Уз помоћ овог програма се јако једноставно креирају .h фајлови са свим потребним наредбама, слика 5.2.

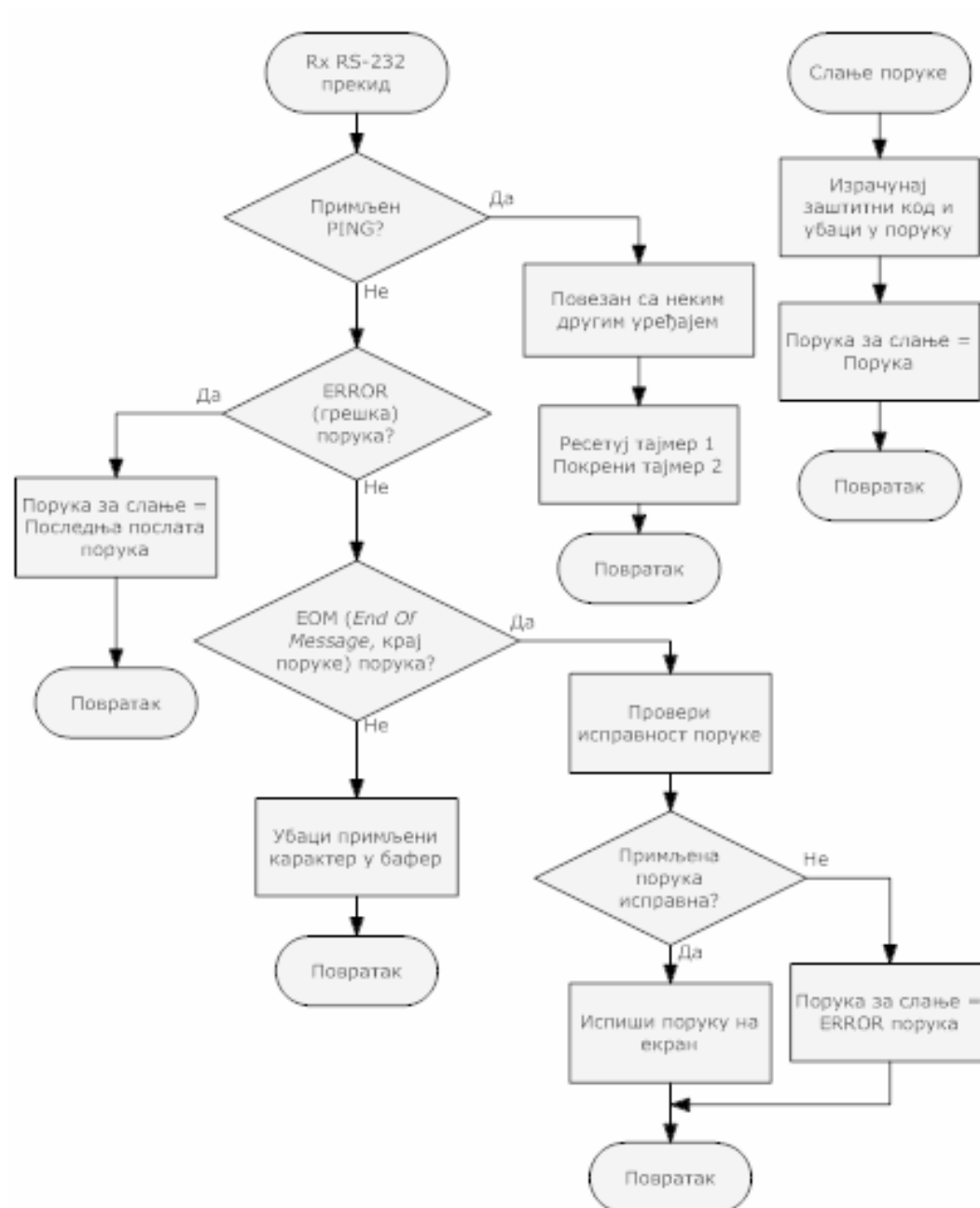
Слика 5.2 Изглед главног прозора програма WDS

5.2. Кориснички интерфејс

Кориснички интерфејс је замишљен као чет програм, помоћу кога би корисници са два одвојена компјутера на које су повезани RF уређаји комуницирали. Алгоритми, на основу којих је програмиран кориснички интерфејс, су приказани на сликама 5.3 и 5.4.



Слика 5.3 Први део алгоритма на основу којег ради GUI



Слика 5.4 Други део алгоритма на основу којег ради GUI

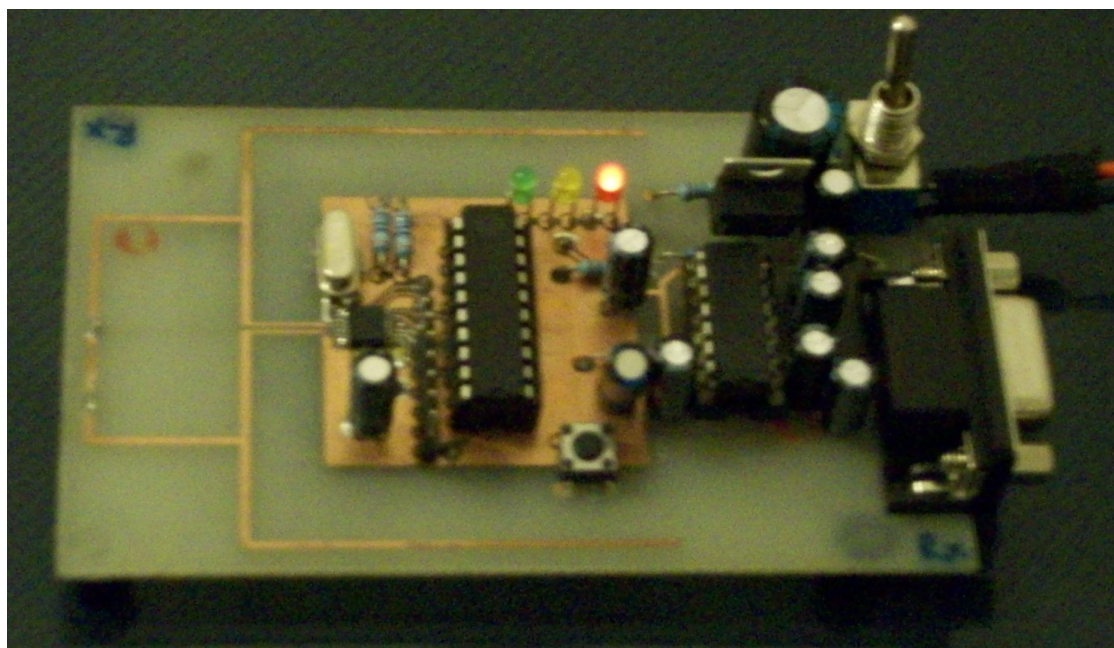
6. Закључак

У овом раду је изложено комплетно практично решење јефтиног и поузданог краткодометног радио линка, реализованог кориштењем микроконтролера ATtiny2313, RF примопредајника IA4421 и додатним компонентама.

IA4421 представља добар и јефтин избор при пројектовању RF комуникацијских уређаја. Својим могућностима знатно олакшава процес развоја и смањује број потребних компоненти. Уређаји засновани на IA4421 који су описани у овом раду су се показали у лабораторијским условима као веома добро решење за комуникацију на кратким раздаљинама.

Могуће примене уређаја су у индустрији, у свим сензорским системима где се мери температура, брзина, ниво воде, притисак, итд., као и у било којој другој примени када је потребан јефтин и поуздан бежични пренос података на кратким растојањима. Уколико би се уређај даље развијао и када би се постигло даље смањење димензија могао би да се искористи и у аутомобилској индустрији у RKE (*Remote Keyless Entry*, даљинско откључавање врата) и TPMS (*Tire-Pressure Monitoring System*, систем за надгледање притиска у гумама) системима.

За крај желим да се захвалим фирми Albatex из Београда, без чије велике помоћи овај рад још увек не би био завршен.



Слика 6.1 Реализован уређај

Литература

- [1] M. Zaser Azizi, *RFIC design for mobile communications*, стр. 33, 2009.
- [2] Thomas H. Lee, *The design of CMOS RF integrated circuits*, Cambridge University Press, 1998.
- [3] <http://en.wikipedia.org>, април 2009.
- [4] Audun Andersen, Texas Instruments, *Antenna Selection Guide*, 2004.
- [5] <http://www.wivia.com>, април 2009.
- [6] <http://standards.ieee.org/getieee802/download/802.11b-1999.pdf>, март 2009.
- [7] Atmel Corporation, *Understanding the Requirements of ISO/IEC 14443 for Type B Proximity Contactless Identification Cards*, новембар 2005.
- [8] Darrell L. Ash, RF Monolithics, Inc., Dallas, Texas, *A comparison between OOK/ASK and FSK modulation techniques for radio links*, 2006.
- [9] John Anthes, RF Monolithics, *OOK, ASK and FSK Modulation in the Presence of an Interfering signal*, март 2004.
- [10] <https://www.silabs.com>, април 2009.
- [11] <http://www.mouser.com>, април 2009.
- [12] Integration Associates, *IA4421 Datasheet*, 2004.
- [13] David Kalinsky, Roee Kalinsky, *Introduction to Serial Peripheral Interface*, http://embedded.com/columns/beginnerscorner/9900483?_requestid=569768, април 2009.
- [14] Cristian Toma, Microchip Technology Inc., *Interfacing the IA4421 Transceiver to PIC® Microcontrollers*, стр. 1, 2008.
- [15] Integration Associates, *IA ISM-AN1, Antenna Selection Guide for the IA4220 and IA4320 ISM Band FSK Transmitter/Receiver Chipset*, јун 2004.
- [16] Integration Associates, *IA ISM-AN2, Antenna Development Guide for the IA4220 and IA4320 ISM Band FSK Transmitter/Receiver Chipset*, мај 2004.
- [17] Integration Associates, *IA ISM-AN6, Antenna Selection Guide for the IA4420 ISM Band FSK Transceiver*, децембар 2005.
- [18] Electronics Industries Association, *EIA Standard RS-232-C Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Data Interchange*, август 1969.
- [19] <http://www.esnips.com/web/AtmelAVR>, април 2009.
- [20] http://atmel.com/dyn/resources/prod_documents/DOC0943.pdf, април 2009.
- [21] http://www.serasidis.gr/circuits/avr_isp/avr_isp.htm, април 2009.

Прилог А: Изворни код

Програм који управља уређајем

Програм је писан у С програмском језику, у програму AVR Studio.

```

                                /*****
                                main.c
                                *****/

#include "utils.h"
#include "USART.h"
#include "IA4421.h"

volatile uchar posl_IA = 0;
volatile uchar posl_RS = 0;
volatile uchar posl_RS_flag = 0;
volatile uchar posl_IA_flag = 0;
volatile uchar first = 1;

ISR(TIMER0_COMPA_vect)           // this interrupt will occur every 0.125ms
{
    if (delay != 0)
        delay--;
}

ISR(USART_TX_vect)
{
    sendUSART();
}

ISR(USART_RX_vect)
{
    posl_IA = UDR;
    posl_IA_flag = 1;
    setRx(posl_IA);
}

ISR(INT1_vect)                   //external interrupt when IA4421's FIFO is full
{
    if (first)
    {
        first = 0;
        return;
    }
    posl_RS_flag = 1;
}
```

```

void initTiny(void)
{
// settings of TIMER0
//-----
// we use Timer to implement a delay of 5ms, needed during tuning of the antenna

    OCR0A = 250;      // on-compare register value (T0 interrupt every 0.25 ms)
    TCNT0 = 0;        // Timer0 initial value (default)
    SBIT(TCCR0A, WGM01); // Clear Timer0 on Compare match mode(CTC)
    SBIT(TCCR0B, CS00);  // TclkT0 = TclkIO / 1 (Timer0 clock is I/O
                        //clock divided by 1)
    SBIT(TIMSK, OCIE0A); // allow TIMER0_COMP interrupt

// settings of FUSE bits
//-----
// the fuse bits can't be programmed here
// instead they are modified during programming of the device
// CKSEL3..0 = 0000; // external clock
// SUT1..0 = 10;     // default value, start-up time bits, slowly rising power
// CKDIV8 = 1;       // Clock Prescaler, Clock Division Factor = 1
// after these settings, external clock and internal system clock of Tiny2313
//will be 1MHz

// settings of GPIO, we use portB(all pins) and portD(only 6) [1 (out), 0 (in)]
//-----
    DDRB = 0xed;      // PB7(MOSI), PB6(SCK), PB5(nSEL IA), PB4(MISO)
    //11101101        // PB3(FFIT transferred to pin 7),
    //PB2(nFIFO Sel), PB1(IA interrupt), PB0(MSG RF diode)
    DDRD = 0x40;      // PD6(MSG UART diode)
    //01000000

// deactivate USART comm with IA4421, active state is logic '0'
    SBIT(PORTB, PB5);  // nSEL = High (inactive)
    SBIT(PORTB, PB2);  // nFFS = High (inactive)

// settings of external interrupts (we use INT1 for indication of full FIFO)
//-----
    SBIT(MCUCR, ISC11);
    SBIT(MCUCR, ISC10); // interrupt will occur on rising edge on INT1 pin
    //SBIT(GIMSK, INT1); // enables ext. pin INT1

// end settings
//-----
    initLED();          // turns off all controlable LEDs
}

int main(void)
{
    initTiny();

```

```

initUSART();
SBIT(SREG, SREG_I);    // allow all interrupts
CBIT(TIMSK, OCIE0A);   // disallow TIMER0_COMP interrupt
initIA4421(); // this goes after general interrupt enable because initIA4421()
                // requires Timer0 interrupt to function properly
readStatusIA4421();
SBIT(GIMSK, INT1);     // enables ext. pin INT1

while (1)
{
    if (posl_IA_flag)
    {
        posl_IA_flag = 0;
        if (posl_IA == PING)           // PING msg
            toggleLED(LED_USART); // toggle LED when
                                   //message from computer is received
    }
    if (posl_RS_flag)
    {
        posl_RS_flag = 0;
        setTx(posl_RS = receiveIA4421());
        if (posl_RS == PING)
            toggleLED(LED_IA);
    }
    sendIA();
}

}

/*****
    ia4421.h
*****/

#ifndef _IA4421_H
#define _IA4421_H

#include "utils.h"

void initIA4421(void);
void sendIA4421(uchar data);
uchar receiveIA4421(void);
void powerDownIA4421(void);
uint readStatusIA4421(void);

#endif

/*****
    ia4421.c
*****/

#include "IA4421.h"
#include "utils.h"

```

```

// IA4421 SPI commands
const long Configuration_cmd      = 0x803B;    // Cload=12.5pF; TX
                                         //registers & FIFO are disabled
const long Power_management_cmd    = 0x8208;    // Everything off, uC clk
                                         //enabled (default)
const long Receiver_control_cmd    = 0x94A1;    // BW=134kHz, DRSSI=-97dBm,
                                         //pin8=VDI, fast VDI
const long Txreg_write_cmd         = 0xB800;
const long FIFO_cmd                = 0xCA81;    // Sync. latch
                                         //cleared, limit=8bits, disable sensitive reset, FIFO enable
const long Data_filter_cmd         = 0xC22C;    // Digital LPF (default)
const long AFC_cmd                 = 0xC4E7;    // Auto AFC (default) // auto/drop 7/8
const long Frequency_cmd           = 0xA7D0;    // Fo=915.000MHz
const long Transmitter_control_cmd = 0x9830;    // df=60kHz, Pmax, normal
                                         //modulation polarity
const long Data_rate_cmd           = 0xC608;    // 38400 bauda//0xC68F;
                                         // 2400Baud          //0xC623// 9579Baud (default)
const long PLL_command             = 0xCC17;    // uC CLK <= 2.5MHz
const long Clock_rate_cmd          = 0xC060;    // uC CLK = 2MHz

#define RF_CLK      PB6                //out
#define RF_SDI      PB7                //out
#define RF_SDO      RBIT(PINB, PINB4)  //in
#define RF_nFFS     PB2                //out
#define RF_SEL      PB5                //out

#define SPI_SCK      RF_CLK
#define SPI_SDI      RF_SDO
#define SPI_SDO      RF_SDI
#define SPI_SEL      RF_SEL

#define SET_USART(x)      SBIT(PORTB, x)
#define CLEAR_USART(x)   CBIT(PORTB, x)
#define TOGGLE_USART(x)  TBIT(PORTB, x)

void SPI_Command(uint);

uchar highSDI(void)
{
    uchar i = PINB;
    return ((i & 0x10) ? 1 : 0);
}

uchar SPI_Read(void)
{
    // written according to diagram 'trc102.pdf', page 15
    uchar i;
    uchar spidata = 0;

```

```
CLEAR_USART(SPI_SDO);

for (i = 0; i < 8; i++)
{
    spidata = spidata << 1;
    if (highSDI())
        spidata |= 0x01;
    else
        spidata &= 0xFE;

    SET_USART(SPI_SCK);
    CLEAR_USART(SPI_SCK);
}
return spidata;
}

uint readStatusIA4421(void)
{
    SPI_Command(0xC4E6); // turn off AFC, in order to get correct status word
    uchar i;
    uint spiStatus = 0;

    CLEAR_USART(SPI_SDO);
    CLEAR_USART(SPI_SEL);

    for (i = 0; i < 16; i++)
    {
        spiStatus <=< 1;

        if (highSDI())
            spiStatus |= 0x01;
        else
            spiStatus &= 0xFE;
        SET_USART(SPI_SCK);
        asm("NOP");
        CLEAR_USART(SPI_SCK);
    }

    SET_USART(SPI_SEL);
    SPI_Command(AFC_cmd); // turn on AFC
    return spiStatus;
}

void SPI_Write(uchar spidata)
{
    // also written according to diagram in 'trc102.pdf', page 15
    uchar i;

    CLEAR_USART(SPI_SDO);
```

```

    for (i = 0; i < 8; i++)
    {
        CLEAR_USART(SPI_SCK);
        if (spidata & 0x80)
            SET_USART(SPI_SDO);
        else
            CLEAR_USART(SPI_SDO);

        SET_USART(SPI_SCK);                // CLK pulse
        spidata = spidata << 1;
    }
    CLEAR_USART(SPI_SCK);
    CLEAR_USART(SPI_SDO);
}

void SPI_Command(uint spicmd)
{
    CLEAR_USART(SPI_SEL);
    SPI_Write ((spicmd & 0xFF00) >> 8);
    SPI_Write ((spicmd & 0x00FF));
    SET_USART(SPI_SEL);
}

void SPI_Write16(uint spicmd)
{
    SPI_Write ((spicmd & 0xFF00) >> 8);
    SPI_Write ((spicmd & 0x00FF));
}

void resetRadio()
{
    SPI_Command(FIFO_cmd);
    SPI_Command(FIFO_cmd | 0x0002);        // FIFO synchron latch re-enable
}

void powerDownIA4421(void)
{
    SPI_Command(0x8201);
}

void initIA4421(void)
{
    SPI_Command(PLL_command);

    //---- configuring the RF link -----
    //---- Send init cmd
    SPI_Command( FIFO_cmd);
    SPI_Command( FIFO_cmd | 0x0002);
    SPI_Command( Configuration_cmd);
}

```

```

    SPI_Command( Frequency_cmd);
    SPI_Command( Power_management_cmd);
    SPI_Command( Receiver_control_cmd);
    SPI_Command( Transmitter_control_cmd);
    SPI_Command( Clock_rate_cmd);

    //---- antenna tuning
    SPI_Command( Power_management_cmd | 0x0020);      // turn on tx
    SBIT(TIMSK,OCIE0A);          // allow TIMER0_COMP interrupt
    delay_ms(5);
    CBIT(TIMSK,OCIE0A);          // disallow TIMER0_COMP interrupt
    //---- end of antenna tuning
    SPI_Command( Power_management_cmd | 0x0080);      // turn off Tx, turn
                                                    //on receiver
    SPI_Command( Configuration_cmd | 0x0040);        // enable the FIFO
    SPI_Command( FIFO_cmd);
    SPI_Command( FIFO_cmd | 0x0002);                // enable synchron latch
}

void sendIA4421(uchar data)
{
    //---- turn off receiver , enable Tx register
    SPI_Command(Power_management_cmd); // turn off transmitter and receiver
    SPI_Command(Configuration_cmd | 0x0080); // Enable the Tx register

    //---- Packet transmission
    // Reset value of the Tx regs are [AA AA], triggers VDI (Valid Data Indicator)
    SPI_Command(Txreg_write_cmd | 0xAA);
    SPI_Command(Txreg_write_cmd | 0xAA);

    //---- Enable Tx
    SPI_Command(Power_management_cmd | 0x0020); // turn on tx
    CLEAR_USART(SPI_SEL);                        // chip select low

    while (!highSDI());
    SPI_Write16(Txreg_write_cmd | 0x2D);          // sync pattern 1nd byte

    while (!highSDI());
    SPI_Write(0xD4);                              // sync pattern 2nd byte

    // transfer of the part of the message
    while (!highSDI()); // wait for SDO, when buffer data <=8 bits
    SPI_Write(data);    // write a byte to tx register

    while (!highSDI());
    SPI_Write(0x00);    // write a dummy byte since the previous byte is
                        //still in buffer
    while (!highSDI()); // wait for the last byte transmission end
    SET_USART(SPI_SEL); // chip select high, end transmission

```

```

    SPI_Command(Power_management_cmd);           // turn off tx
    //---- Turn off Tx disable the Tx register
    SPI_Command(Power_management_cmd | 0x0080);   // turn off Tx, turn
                                                //on the receiver
    SPI_Command(Configuration_cmd | 0x0040);      // disable the Tx register,
                                                //Enable the FIFO
}

uchar receiveIA4421(void)
{
    // this function is called when FFIT pin goes to logical '1'
    // according to 'ia4421.pdf', page 31
    uchar temp;
    CLEAR_USART(RF_nFFS);
    temp = SPI_Read();
    SET_USART(RF_nFFS);
    resetRadio();
    return temp;
}

```

```

/*****

```

```

    USART.h

```

```

*****/

```

```

#ifndef _USART_H
#define _USART_H

```

```

#include "utils.h"

```

```

//modes

```

```

#define RX_BUF    0

```

```

#define TX_BUF    1

```

```

void initUSART(void);
void sendUSART(void);
void sendIA(void);
void setRx(uchar);
void setTx(uchar);

```

```

#endif

```

```

/*****

```

```

    USART.c

```

```

*****/

```

```

#include "utils.h"
#include "USART.h"
#include "IA4421.h"

```

```

// UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS RX & TX

```

```

#define MAX_RX    30           // incoming from RS-232

```

```

#define MAX_TX    20           // incoming from RF

```

```
uchar RxBuffer[MAX_RX], RxPointer = 0, toReceive = 0;
uchar TxBuffer[MAX_TX], TxPointer = 0, toSend = 0;

uchar empty(uchar buf) // buf: 0-RxBuf, 1-TxBuf
{
    if (buf)
    {
        if (TxPointer == toSend) return 1;
    }
    else
        if (RxPointer == toReceive) return 1;
    return 0;
}

void initUSART(void)
{
    // For the proper functioning of USART relevant registers are: UCSRA,
    //UCSRB, UCSRC and UBBRH and UBBRL used to set baud-rate
    //
    // After the following settings USART will have properties:
    //     - asynchronous mode
    //     - 8 data bits
    //     - 1 stop bit
    //     - no parity bit
    //     - baud-rate = 9600bps when fosc = 2MHz

    UBRRH = 0;
    UBBRL = 12;          // 25 for 2400 baud @ 1MHz

    SBIT(UCSRB, RXCIE);    //rx interrupt enable
    SBIT(UCSRB, TXCIE);    //tx interrupt enable
    SBIT(UCSRB, RXEN);     //rx enable
    SBIT(UCSRB, TXEN);     //tx enable
}

void sendUSART(void)
{
    if (empty(TX_BUF))
        return;
    TXB = TxBuffer[toSend++];
    if (toSend == MAX_TX)
        toSend = 0;
}

uchar posalji = 0;
void setTx(uchar byte)
{
    if (TxPointer == toSend)
        posalji = 1;
}
```

```

    TxBuffer[TxPointer++] = byte;
    if (posalji)
    {
        posalji = 0;
        sendUSART();
    }
    if (TxPointer == MAX_TX)
        TxPointer = 0;
}

void setRx(uchar slovo)
{
    RxBuffer[RxPointer++] = slovo;
    if (RxPointer == MAX_RX)
        RxPointer = 0;
}

void sendIA(void)
{
    if (empty(RX_BUF))
        return;
    sendIA4421(RxBuffer[toReceive++]);
    if (toReceive == MAX_RX)
        toReceive = 0;
}

    /*****
        utils.h
    *****/

#ifndef _UTILS_H
#define _UTILS_H

#include <avr/io.h>
#include <avr/interrupt.h>

typedef unsigned char uchar;
typedef unsigned int uint;

#define EOM      200           // end of message
#define PING     201           // ping msg
#define ERROR    202           // error msg
#define EOC      203           // end of connection
#define OOR      204           // out of reach

#define SBIT(reg,n) (reg |= (1 << n))           // set bit
#define CBIT(reg,n) (reg &= ~(1 << n))           // clear bit
#define TBIT(reg,n) (reg ^= (1 << n))           // toggle bit
#define RBIT(reg,n) ((reg & (1 << n) != 0) ? 1 : 0) // read bit

```

```
///// diodes management
#define LED_IA          1
#define LED_USART      0

void toggleLED(uchar LED);
void initLED(void);

///// delays
extern volatile uint delay;
void delay_ms(uint ms);

#endif

                                     /*****
                                     utils.c
                                     *****/

#include "utils.h"
volatile uint delay = 0;

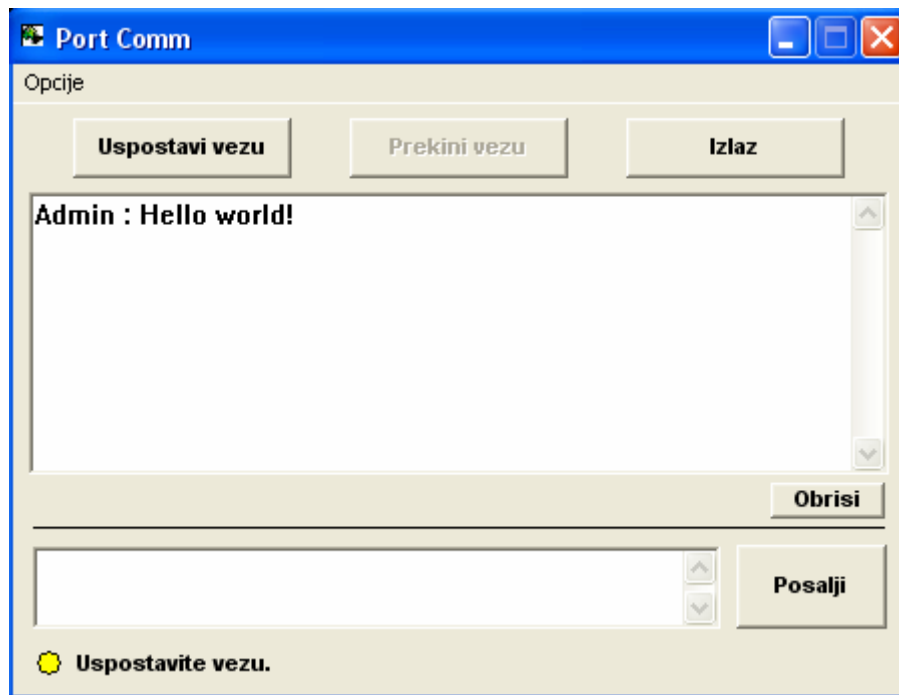
void delay_ms(uint ms)
{
    ms <= 3;    // because interrupt occurs every 0.125 ms
    delay = ms;
    while (delay) {};
}

void initLED(void)
{
    SBIT(PORTD, PD6);
    SBIT(PORTB, PB0);
}

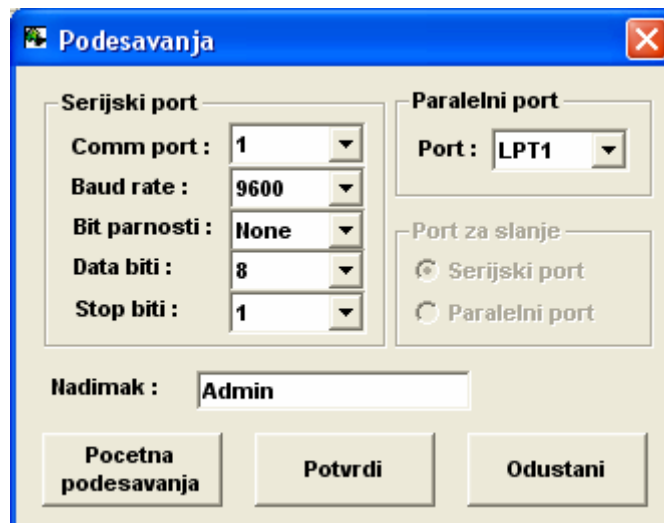
void toggleLED(uchar LED)
{
    if (LED)          // Diode On/Off when message arrives
        TBIT(PORTB, PB0);
    else
        TBIT(PORTD, PD6);
}
```

Чет програм

Програм је писан у програму Visual Basic. Изглед главног прозора програма је дат на слици А.1. Изглед прозора за подешавање је дат на слици А.2.



Слика А.1 Изглед главног прозора чет програма (*Form1.frm*)



Слика А.2 Изглед прозора за подешавање (*Form2.frm*)

```

/*****
Form1.frm
*****/

```

Option Explicit
Dim tekst As String

```
Dim slovo As String
Dim temp As String
Dim za_slanje As String
Dim prva_poruka As Boolean
Dim konektovan As Boolean
Dim poruka_za_slanje As Boolean
Dim duzina_poruke As Byte
Dim provera As Boolean
Dim tmp As String
Dim prekid As Boolean
Dim ime_prijem As Boolean
Public ime_ja As String
Public ime_izm As Boolean
Dim ime_on As String
'konstante:
' End Of Message      = 200
' PING                = 201
' ERROR               = 202
' End Of Connection   = 203
' Out Of Reach        = 204
' Name Message        = 205
Const EOM = 200
Const PING = 201
Const ERROR = 202
Const EOC = 203
Const OOR = 204
Const NM = 205

Private Sub Command5_Click()      'Izlaz
    End
End Sub

Private Sub MSComm1_OnComm()
    ' prvi karakter poruke oznacava duzinu poruke
    If (MSComm1.CommEvent = comEvReceive) Then
        slovo = MSComm1.Input
        If (Asc(slovo) = 0) Then
            Exit Sub
        End If
        If (Asc(slovo) = PING) Then      'ping poruka
            If ((prekid = True) Or (ime_izm)) Then
                za_slanje = Chr(Len(ime_ja) + 1) & Chr(NM) & ime_ja & Chr(EOM) &
Chr(EOM)
                temp = za_slanje
                ime_izm = False
                poruka_za_slanje = True
            End If
            konektovan = True
            prekid = False
        End If
    End If
End Sub
```

```
Shape1.FillColor = &H80FF80
Text2.BackColor = &H80000005
Label1.Caption = "Veza uspostavljena."
Timer500.Enabled = False 'reinicijalizacija timera
Timer20.Enabled = True ' za sledeci poslati ping
TimerPom.Enabled = False
Timer500.Enabled = True
Exit Sub
ElseIf (Asc(slovo) = ERROR) Then
    za_slanje = temp
    poruka_za_slanje = True
    Exit Sub
End If
If (Asc(slovo) = EOM) Then
    If (tekst = "") Then
        Exit Sub
    End If
    If (duzina_poruke <> Len(tekst)) Then
        za_slanje = Chr(ERROR) 'error poruka
        poruka_za_slanje = True
        tekst = ""
    Else
        If (InStr(1, tekst, Chr(NM), vbTextCompare) <> 0) Then
            tmp = ime_on
            ime_on = Right(tekst, Len(tekst) - 1)
            If (tmp <> ime_on) Then
                If (ime_on = "Komp1") Then
                    ime_on = "Komp2"
                ElseIf ((ime_on & " gost") = tmp) Then
                    ime_on = tmp
                Else
                    ispisi (""" & tmp & "" je promenio ime u "" & ime_on & """)
                    If (ime_on = ime_ja) Then
                        ime_on = ime_on & " gost"
                    End If
                End If
            End If
        End If
        tekst = ""
    Else
        If (Not prva_poruka) Then
            Text2.Text = Text2.Text & vbCrLf
        End If
        prva_poruka = False
        Text2.Text = Text2.Text & ime_on & " : " & tekst
        tekst = ""
        Text2.SetFocus
        SendKeys "^{END}", True
        Text1.SetFocus
    End If
```

```
End If
duzina_poruke = 0
provera = True
Else
  If (tekst = "" And provera) Then
    duzina_poruke = Asc(slovo)
    provera = False
    Exit Sub
  End If
  tekst = tekst & slovo
End If
End If
End Sub

Private Sub posalji_poruku(poruka As String)
  Dim broj As Integer
  If (MSComm1.PortOpen) Then
    MSComm1.Output = poruka
  Else
    Text2.BackColor = &HFF
  End If
End Sub

Private Sub ispisi(ispis As String)
  If (Not prva_poruka) Then
    Text2.Text = Text2.Text & vbCrLf
  End If
  Text2.Text = Text2.Text & ispis
  prva_poruka = False
  Text2.SetFocus
  SendKeys "^{END}", True
  Text1.SetFocus
  Text1.Text = ""
End Sub

Private Sub Command1_Click()      'Uspostavi vezu
  On Error GoTo SPI_error:
  MSComm1.Settings = podes
  MSComm1.PortOpen = True
  Command4.Enabled = True
  Command1.Enabled = False
  Text1.SetFocus
  Text2.BackColor = &H80000005
  Label1.Caption = "Trazim uredjaje u dometu ..."
  Timer500.Enabled = True
  Exit Sub
SPI_error:
  MsgBox "Greska prilikom otvaranja serijskog porta.", vbCritical + vbOKOnly
End Sub
```

```
Private Sub Command2_Click()      'Posalji
    If (Text1.Text = "") Then
        Text1.SetFocus
        Exit Sub
    End If
    za_slanje = Chr(Len(Text1.Text)) & Text1.Text & Chr(200) & Chr(200)
    temp = za_slanje
    poruka_za_slanje = True
    ispisi (ime_ja & " : " & Text1.Text)
    If (MSComm1.PortOpen = False) Then
        Text2.BackColor = &HFF
        ispisi (" **** Poslednja poruka nije poslata!!! ****")
        poruka_za_slanje = False
        Exit Sub
    End If
    If (Not konektovan) Then
        Text2.BackColor = &HFF
        ispisi (" **** Poslednja poruka nije poslata!!! ****")
        poruka_za_slanje = False
    Else
        Text2.BackColor = &H80000005
    End If
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If (Len(Text1.Text) = 60 And KeyAscii <> 13) Then
        If (KeyAscii <> 8) Then
            KeyAscii = 0
        End If
        Exit Sub
    End If
    If (KeyAscii = 13) Then
        KeyAscii = 0
        If (Text1.Text = "") Then
            Text1.SetFocus
            Exit Sub
        End If
        za_slanje = Chr(Len(Text1.Text)) & Text1.Text & Chr(EOM) & Chr(EOM)
        temp = za_slanje
        poruka_za_slanje = True
        ispisi (ime_ja & " : " & Text1.Text)
        If (MSComm1.PortOpen = False) Then
            Text2.BackColor = &HFF
            ispisi (" **** Poslednja poruka nije poslata!!! ****")
            poruka_za_slanje = False
            Exit Sub
        End If
    End If
End Sub
```

```
Private Sub Command3_Click()      'Obrisi
    Text2.Text = ""
    Text1.SetFocus
    prva_poruka = True
End Sub

Private Sub Command4_Click()      'Prekini vezu
    MousePointer = vbHourglass
    'MSComm1.Output = Chr(EOC)      'end of connection
    MSComm1.PortOpen = False
    Command4.Enabled = False
    Command1.Enabled = True
    Text1.SetFocus
    MousePointer = vbDefault
    Shape1.FillColor = &HFFFF&
    Label1.Caption = "Uspostavite vezu."
    Timer20.Enabled = False
    TimerPom.Enabled = False
    Timer500.Enabled = False
    konektovan = False
End Sub

Private Sub Form_Load()
    a = 5
    b = 2
    c = 4
    d = 0
    e = 0
    f = 0
    podes = "9600,N,8,1"
    tekst = ""
    MSComm1.CommPort = 1
    ' 2400 baud, no parity, 8 data, and 1 stop bit.
    MSComm1.Settings = podes
    MSComm1.OutBufferSize = 100
    'MSComm1.EOFEnable = True
    MSComm1.RThreshold = 1
    MSComm1.InputLen = 1
    parPort = &H378
    prva_poruka = True
    prekid = True
    poruka_za_slanje = False
    konektovan = False
    Label1.Caption = "Uspostavite vezu."
    Shape1.FillColor = &HFFFF&
    Timer20.Enabled = False
    Timer500.Enabled = False
    TimerPom.Enabled = False
    ime_ja = "Komp1"
```

```
ime_on = "Komp2"
ime_izm = False
ime_prijem = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If (MSComm1.PortOpen = True) Then
        MSComm1.PortOpen = False
    End If
End Sub

Private Sub mnuExit_Click()
    Unload Me
End Sub

Private Sub mnuSettings_Click()
    Form2.Visible = True
    Form1.Enabled = False
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    Text1.SetFocus
    Text1.Text = Chr(KeyAscii)
    SendKeys "{END}", True
End Sub

Private Sub Timer20_Timer()
    If (poruka_za_slanje) Then
        poruka_za_slanje = False
        posalji_poruku (za_slanje)
    End If
    MSComm1.Output = Chr(PING) 'ping poruka
    If (konektovan) Then
        Timer20.Enabled = False
    End If
End Sub

Private Sub Timer500_Timer()
    konektovan = False
    MSComm1.Output = Chr(PING) 'ping poruka
    Timer20.Enabled = True
    TimerPom.Enabled = True
End Sub

Private Sub TimerPom_Timer()
    prekid = True
    Label1.Caption = "Nema nikoga u dometu."
    Shape1.FillColor = &HFF&
    'MSComm1.Output = Chr(OOR)
```

```

Timer500.Enabled = False
TimerPom.Enabled = False
End Sub

```

```

/*****
Form2.frm
*****/

```

```

Private Sub Command1_Click()    'Potvrđi
    Dim i As Byte

    If (Text1.Text <> Form1.ime_ja) Then
        Form1.ime_ja = Text1.Text
        Form1.ime_izm = True
    End If

    If ((a <> Combo1(0).ListIndex) Or (b <> Combo1(1).ListIndex) Or (c <>
    Combo1(2).ListIndex) Or (d <> Combo1(3).ListIndex) Or (e <>
    Combo1(4).ListIndex) Or (f <> Combo1(5).ListIndex)) Then
        If (Form1.MSComm1.PortOpen = True) Then
            MsgBox "Veza ce se prekinuti posto su izmenjeni parametri serijske veze.",
vbInformation
        End If
    Else
        Unload Me
        Exit Sub
    End If
    podes = ""
    For i = 0 To 2
        If (i = 1) Then
            podes = podes & Left(Combo1(i).List(Combo1(i).ListIndex), 1) & ","
        Else
            podes = podes & Combo1(i).List(Combo1(i).ListIndex) & ","
        End If
    Next i
    podes = podes & Combo1(3).List(Combo1(3).ListIndex)
    If (Form1.MSComm1.PortOpen = True) Then
        MousePointer = vbHourglass
        Form1.MSComm1.PortOpen = False
        Form1.Command4.Enabled = False
        Form1.Command1.Enabled = True
        MousePointer = vbDefault
        Form1.Shape1.FillColor = &HFFFFF&
        Form1.Label1.Caption = "Uspostavite vezu."
        Form1.Timer20.Enabled = False
        Form1.Timer500.Enabled = False
    End If
    Form1.MSComm1.Settings = podes
    Form1.MSComm1.CommPort = Val(Combo1(4).List(Combo1(4).ListIndex))

```

```

If (Combo1(5).List(Combo1(5).ListIndex) = "LPT2") Then
    parPort = &H278
Else
    parPort = &H378
End If
a = Combo1(0).ListIndex
b = Combo1(1).ListIndex
c = Combo1(2).ListIndex
d = Combo1(3).ListIndex
e = Combo1(4).ListIndex
f = Combo1(5).ListIndex
Unload Me
End Sub

```

```

Private Sub Command2_Click()    'Odustani
    Unload Me
End Sub

```

```

Private Sub Command3_Click()    'Pocetna podesavanja
    Combo1(0).ListIndex = 4
    Combo1(1).ListIndex = 2
    Combo1(2).ListIndex = 4
    Combo1(3).ListIndex = 0
    Combo1(4).ListIndex = 0
    Combo1(5).ListIndex = 0
End Sub

```

```

Private Sub Form_Load()
    Option4.Value = True
    Option3.Enabled = False
    Option4.Enabled = False
    Combo1(0).ListIndex = a
    Combo1(1).ListIndex = b
    Combo1(2).ListIndex = c
    Combo1(3).ListIndex = d
    Combo1(4).ListIndex = e
    Combo1(5).ListIndex = f
    Text1.Text = Form1.ime_ja
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    Form1.Enabled = True
End Sub

```

```

/*****
Module1.bas
*****/

```

```

Public podes As String
Public a, b, c, d, e, f As Byte

```