

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 531

**SUSTAV ZA AUTOMATIZIRANO  
NAVODNJAVANJE POLJOPROVREDNIH  
POVRŠINA**

Lovro Damiš

Zagreb, lipanj 2012.

Ovdje ide izvornik diplomskog zadatka!!!

*Zahvaljujem mentoru prof.dr.sc. Davoru Petrinoviću što me prihvatio kao svog diplomanta i omogućio mi izradu diplomskog rada u području mog interesa.*

*Zahvaljujem g. Zoranu Lovreku na ideji, svim stručnim savjetima iz područja agronomije i materijalnim sredstvima bez kojih nebi bilo realizacije ovog projekta.*

*Zahvaljujem svojoj obitelji koja je bila najveća podrška svih ovih godina.*

## Sadržaj

1	Uvod .....	1
2	Optimalno navodnjavanje .....	2
3	Zahtjevi i osnovni princip rada .....	5
4	Senzorska jedinica .....	7
4.1	Opis algoritma senzorske jedinice .....	8
4.2	Opis algoritma ugradbenog računala .....	10
5	Centralna jedinica .....	14
6	Izvršna jedinica .....	20
7	Napajanje jedinica i potrošnja .....	25
8	Opis komponenata .....	29
8.1	Watermark senzor .....	29
8.2	Z/f pretvornik .....	31
8.2.1	Postupak dobivanja koeficijenata .....	35
8.3	Primopredajni modul RFM22B .....	38
8.4	Elektromagnetni ventil za vodu .....	48
8.4.1	Ventil s membranom .....	48
8.4.2	Solenoid .....	50
8.5	H-most: BD6222 .....	52
8.6	USB/SPI pretvornik: FT232RL .....	57
8.6.1	Opis rada FT232RL .....	58
8.7	Ugradbeno računalo: ATtiny2313A .....	61
8.7.1	Prekidi .....	63
8.7.2	Takt sustava .....	65
8.7.3	Sleep stanja .....	66
8.7.4	Power Reduction Register (PRR) .....	66
8.7.5	8-bit Timer/Counter sa PWM generatorom .....	67
8.7.6	16-bit Timer/Counter1 .....	69
8.7.7	Električne karakteristike .....	70
9	Zaključak .....	72
10	Literatura .....	74
11	Naslov .....	75
12	Sažetak .....	76

13	Ključne riječi .....	78
14	Privitak.....	79

## 1 Uvod

Navodnjavanje podrazumijeva svaki oblik umjetnog povećavanja sadržaja vode u tlu poljoprivrednih površina. To je stara melioracijska mjera koja je u prošlosti zabilježena već 6000 godina prije Krista. Po svojem navodnjavanju poznati su Egipćani koji su koristili rijeku Nil, a Grci i Rimljani su gradili vrlo sofisticirane sustave za navodnjavanje po Europi i Aziji. Razvitkom crpke i otkrićem novih materijala, nakon Drugog svjetskog rata, navodnjavanje se naglo razvija. Tako do danas imamo 17% obradivih površina koje se navodnjavaju, a na njima se proizvodi gotovo 40% svjetske hrane. U Hrvatskoj je situacija nešto drugačija te se navodnjava tek 0.86% obradivih površina, što je daleko ispod naše prirodne mogućnosti i potrebe. Upravo u tim podacima nalazim motivaciju i interes za izradu automatiziranog sustava za navodnjavanje poljoprivrednih površina (u daljnjem tekstu sustav), kako bi ono bilo dostupnije i kvalitetnije.

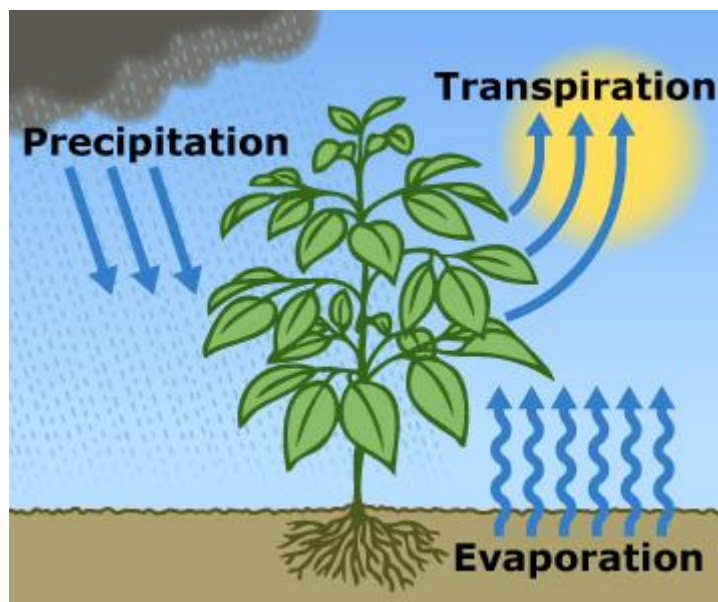
Automatizirano navodnjavanje podrazumijeva samostalan i zaokružen sustav koji na temelju informacija sa senzora signalizira kada i koliko navodnjavati kako bi se određenoj poljoprivrednoj kulturi osigurala optimalna količina vode. Potrošnja optimalne količine vode u tlu donosi dvostruku korist: maksimizaciju biološkog potencijala poljoprivredne kulture uz minimizaciju utrošene vode.

Danas već postoje gotovi sustavi za navodnjavanje, međutim oni su većim dijelom poluautomatski što znači da se navodnjava na temelju proračuna i u vremenskim okvirima. U tim sustavima ne postoji nikakva povratna veza koja bi potrošnju vode regulirala u točno optimalnim uvjetima. Svrha ovog Diplomskog rada je prezentirati potpuno automatizirani sustav za navodnjavanje, koji automatski uspostavlja korelaciju optimalne potrošnje vode (potrebne vlage u tlu) za biološki potencijal određene poljoprivredne kulture.

## 2 Optimalno navodnjavanje

Općenito, navodnjavanje se vremenski programira tako da se nadoknadi deficit vode u tlu. Kako različite poljoprivredne kulture imaju različite zahtjeve za vodom, tako i deficit vode varira s njima. Vrlo važnu ulogu u tome ima i vrsta tla koje se koristi za uzgoj. Spomenuto možemo svesti u matematički oblik prema kojem se planira navodnjavanje. Ključne pojmove koje uvodimo u proračun su evapotranspiracija, koeficijent kulture, efektivne oborine i konačno deficit vode.

Evapotranspiracija je zbroj evaporacije i transpiracije (Slika 1). Evaporacija znači gubitak vode isparivanjem iz tla, a transpiracija je gubitak vode u kemijskim procesima biljke te isparivanjem kroz lišće.



Slika 1 - Evapotranspiracija

Očito je da različite biljke imaju različitu transpiraciju te se za svaku definira koeficijent kulture koji govori količini transpiracije.

Evapotranspiracija neke kulture računa se prema formuli (Formula 1), gdje  $ET_0$  označava referentnu evapotranspiraciju (evapotranspiracija idealnog travnjaka), a  $k_c$  koeficijent kulture.

$$ET_c = k_c * ET_0 \text{ [mm/dan]} \quad (1)$$

Deficit vode dobijemo iz formule (Formula 2).  $O_{ef}$  su efektivne oborine promatranog područja izražene u mm/dan.

$$Dv = ET_c - O_{ef} \text{ [mm/dan]} \quad (2)$$

Gore navedeni proračun deficita vode prihvatljiva je aproksimacija i osnovni čimbenik u projektiranju navodnjavanja. Ipak, da bi se dobili preostali potrebni parametri, moraju se koristiti podatci iz gotovo kompletne meteorološke stanice, što nije uvijek moguće. Uz to, takav podatak može se koristiti isključivo za poluautomatsko navodnjavanje. U ovom istraživanju, deficit vode bit će kontrolni podatak.

Za potpunu automatizaciju navodnjavanja, potrebno je pronaći mjerljive parametre čiji su odzivi dovoljno brzi za regulaciju sustava.

Agronomska struka nas uči da voda u tlu može biti vezana i slobodna. Vezana voda je ona koja se jakim silama drži za čestice u tlu i nije dostupna korijenu biljke. Slobodna voda se drži malim silama za čestice tla, te se može kretati u svim smjerovima. Upravo slobodna voda je ona koju korijenje crpi iz tla i njena postojanost ključni je parametar optimalnog navodnjavanja.

Slobodne vode može biti i previše. Takvo stanje nazivamo stanjem prenatopljenosti i može rezultirati propadanjem korijena. Iz tablice (Tablica 1) možemo očitati optimalne vrijednosti slobodne vode. Ona se iskazuje tlakom što je proporcionalno sili koja je potrebna korijenu da izvuče vodu iz tla. Koriste se jedinice cb (centibari) ili kPa (kilopascali) jer je  $1 \text{ cb} = 1 \text{ kPa}$ .

**Tablica 1** – Sile vode u tlu

Tlak [kPa], [cb]	Napomena
<10	Prenatopljenost
<b>10 - 30</b>	<b>Optimalna vlažnost</b>
600-700	Granica slobodne i vezane vode
>1500	Vlažnost venjenja

Postoje mnogi senzori za mjerenje vlažnosti tla. Jedino *Watermark* senzor, proizvođača Irrometer, može iskazati rezultate u centibarima odnosno



kilopascalima. Takav se senzor može direktno upotrijebiti u automatiziranom sustavu navodnjavanja, održavajući vlažnost tla u optimalnim uvjetima.

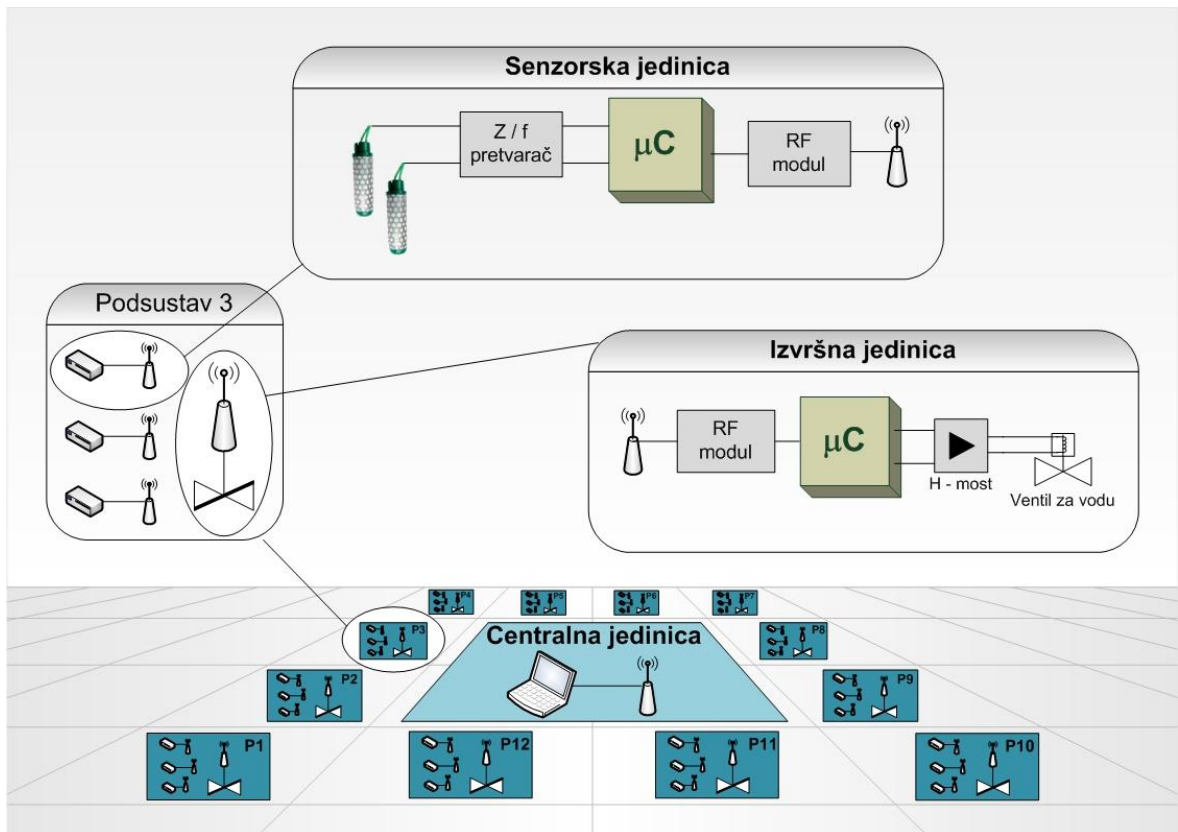
U nastavku bit će objašnjeni zahtjevi takvog sustava, te će se opisati realizacija prototipa sa svim njegovim specifikacijama.

### 3 Zahtjevi i osnovni princip rada

Glavna ideja jest napraviti sustav koji se može integrirati u postojeće sustave navodnjavanja. Kako dostupnost električne energije na poljoprivrednim terenima varira od područja do područja, slijedi logičan zahtjev da novi sustav o kojem je ovdje riječ bude u potpunosti baterijski napajan. Stoga je izbor korištenih komponenti i programiranje sustava upravo takvo da se minimizira potrošnja energije iz baterije. Sklopovske jedinice koje će se ugrađivati moraju biti što manjih dimenzija kako na terenu ne bi smetale uzgoju navodnjavane kulture i njenoj poljoprivrednoj obradi. Također, svi elementi koji se koriste u sustavu, moraju biti izolirani ili otporni na vanjske utjecaje poput temperature, vlage, nečistoće sitne granulacije i sl. Najveća procijenjena udaljenost između dviju jedinica koje bežično komuniciraju jest 1 Km. Zato je potrebno koristiti RF module dovoljno velike snage kako bi se postigao traženi domet.

Sustav je cjelina tri vrste jedinica: Senzorska jedinica, Centralna jedinica i Izvršna jedinica. Svaka se ponaša kao zasebni sklop s mogućnošću bežičnog komuniciranja s ostalim jedinicama. Ukupnost sustava može se podijeliti na podsustave koji su građeni od  $N$  senzorskih jedinica te jedne izvršne jedinice. Na  $M$  podsustava dolazi jedna centralna jedinica. Takav pristup pruža veliku fleksibilnost u gradnji sustava i jednako je efikasan kako za male tako i za velike površine. Prototip će biti sustav za koji vrijedi  $M=1$  i  $N=1$  tj. po jedna od svake jedinice.

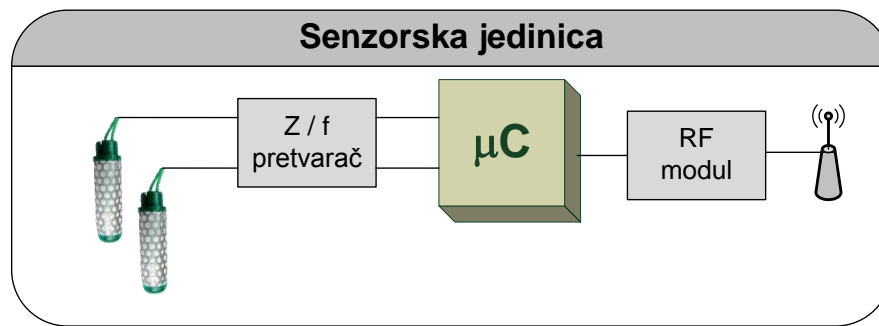
Organizacija sustava prikazana je slikom (Slika 2), a princip rada može se naslutiti iz naziva jedinica: Senzorska jedinica prikuplja podatke o vlažnosti tla, šalje ih u Centralnu jedinicu koja nakon obrade podataka utvrđuje koje ventile i kada treba otvoriti, što se čini preko Izvršne jedinice koja se nalazi uz ventil. Time je ostvarena osnova za potpuno automatsko upravljanje sustavom za navodnjavanje.



**Slika 2 – Organizacija sustava**

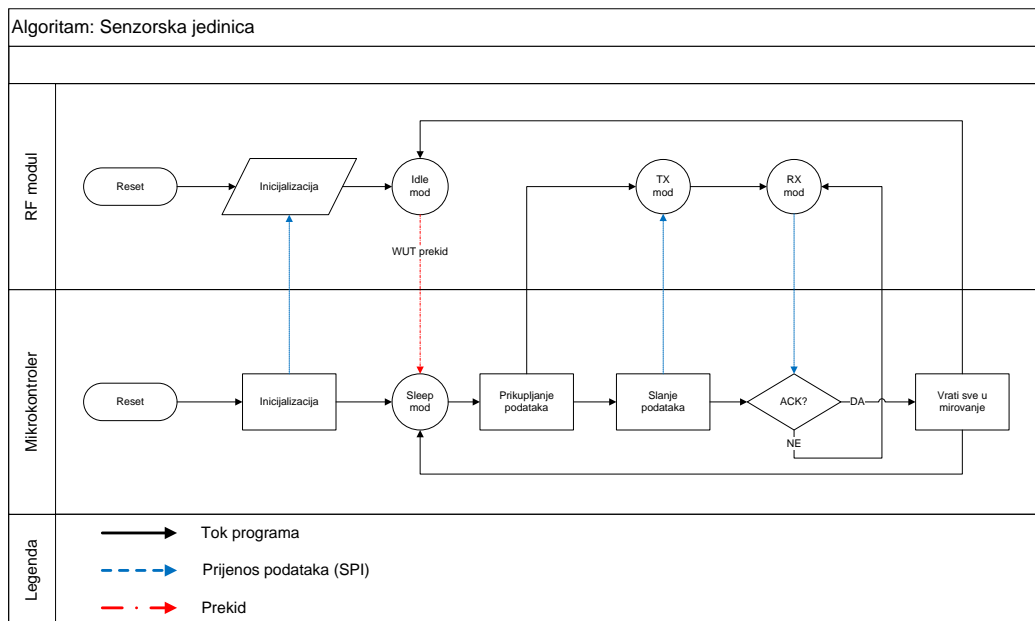
## 4 Senzorska jedinica

Na slici (Slika 3) je prikazan shematski prikaz Senzorske jedinice. Naziv je dobila po svojoj glavnoj funkciji, a to je mjerenje vlage tla pomoću *Watermark* senzora proizvođača Irrometer (na slici lijevo). Predviđena su dva ulaza za senzore tako da se može mjeriti vlažnost tla na dvije različite dubine. Vlažnost tla proporcionalna je impedanciji senzora (izražene su otporne i kapacitivne komponente) koja preko Z/f pretvarača definira frekvenciju na izlazu. Tako je vlažnost tla izražena preko frekvencije koja se kasnije baždari u centibare (cb) ili kilopascale (kPa).



**Slika 3** – Senzorska jedinica

Ugradbeno računalo ( $\mu\text{C}$  na slici) budi se iz *sleep* stanja na zahtjev RF modula, očitava vrijednosti senzora mjereći frekvenciju, formira podatkovni paket te ga preko RF modula šalje u centralnu jedinicu. Komunikacija ugradbenog računala i RF modula prikazana je slikom (Slika 4), a algoritam programa upisanog u ugradbeno računalo slikom (Slika 5).



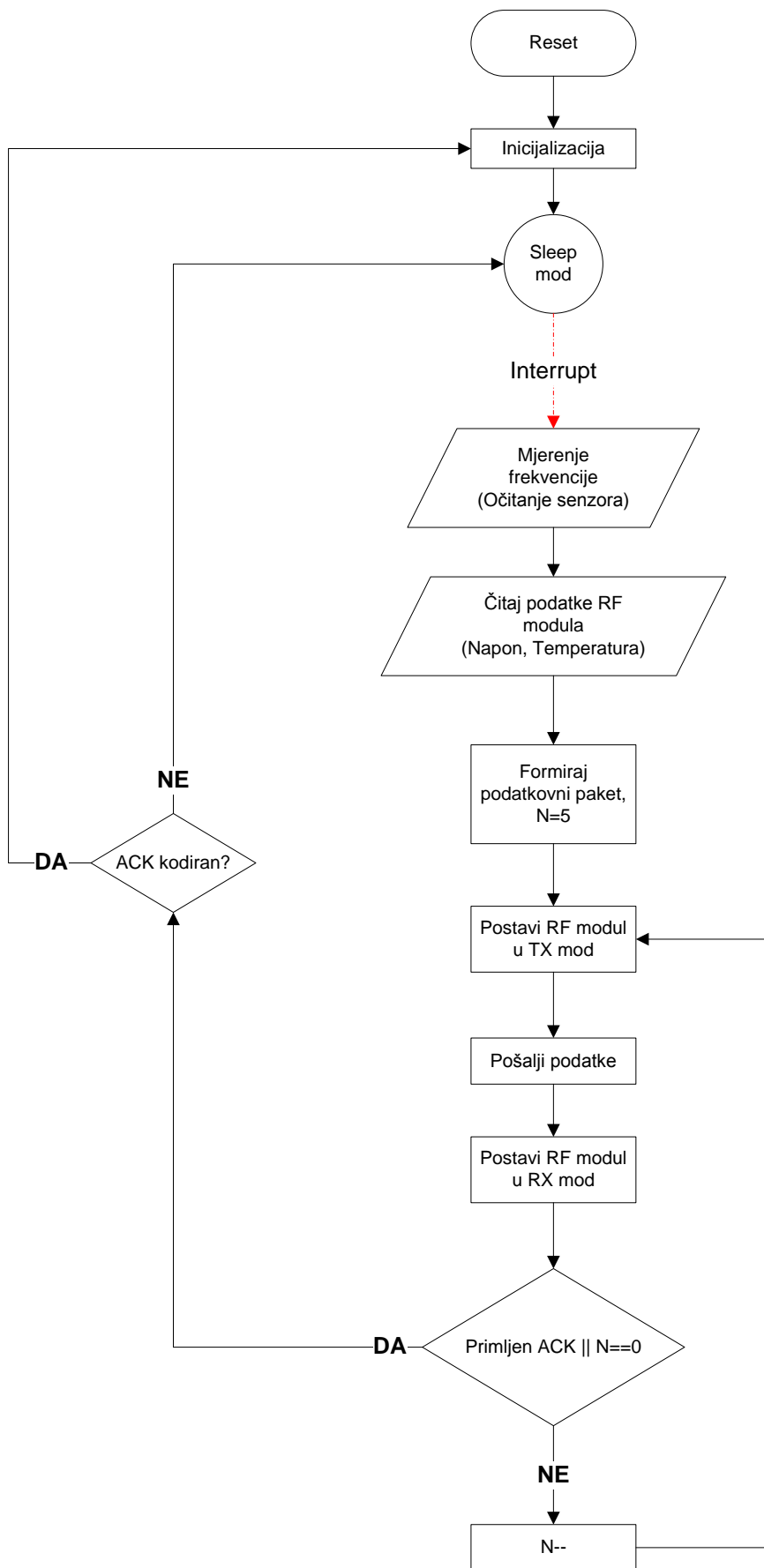
**Slika 4** – Komunikacija ugradbenog računala i RF modula

#### 4.1 Opis algoritma senzorske jedinice

Pri uključenju senzorske jedinice, RF modul pokreće interni sklop za vlastitu inicijalizaciju, a ugradbeno računalo izvršava programiranu inicijalizacijsku rutinu. Kako pretpostavljena inicijalizacija RF modula ne odgovara potrebama sustava, ugradbeno računalo nakon vlastite inicijalizacije programira RF modul. Ugradbeno računalo tada ulazi u *sleep* stanje, a RF modul je u stanju mirovanja, u kojem radi samo WUT vremenski sklop.

Nakon isteka WUT vremenskog sklopa, RF modul postavlja NIRQ izvod u aktivno stanje (logička '0') tj. šalje zahtjev za prekid, a WUT vremenski sklop počinje novo odbrojavanje. Prilikom inicijalizacije, zadani period WUT vremenskog sklopa jest 10 sekundi, a moguća je i promjena perioda što je objašnjeno kasnije.

U trenutku postavljanja prekida, ugradbeno računalo prelazi iz *sleep* stanja u aktivno stanje. Prva slijedeća operacija ugradbenog računala je prikupljanje podataka, po čemu slijede njegova obrada i bežični prijenos, a prije vraćanja u *sleep* stanje, čeka se potvrda o uspješno poslanom podatku koja dolazi iz centralne jedinice.



**Slika 5** – Algoritam ugradbenog računala u Senzorskoj jedinici

## 4.2 Opis algoritma ugradbenog računala

Ponašanje ugradbenog računala prije ulaska u *sleep* stanje objašnjeno je u opisu algoritma senzorske jedinice. Ovdje će se detaljnije opisati ponašanje ugradbenog računala od izlaska iz *sleep* stanja do njegovog ponovnog ulaska.

U trenutku pojave prekida izazvanog RF modulom i to zbog isteka perioda WUT vremenskog sklopa, ugradbeno računalo izlazi iz *sleep* stanja te prelazi u aktivno stanje, što znači da sada svi sklopovi ugradbenog računala dobivaju radni takt te nastavljaju svoj rad. Odmah po uspostavi aktivnog stanja, ugradbeno računalo izvršava prekidnu rutinu prije nego se vrati u glavni tok programa.

U prekidnoj se rutini ponovno inicijaliziraju neki dijelovi ugradbenog računala koji su morali biti deaktivirani za vrijeme *sleep* stanja. Također, dojavljuje se RF modulu da je prekid uspješno detektiran po čemu prekidni signal prelazi u neaktivno stanje (logička '1').

U glavnom programu poziva se rutina za obradu podataka koja poziva pomoćne rutine za dohvat pojedinog podataka. Prvo se pristupa mjerenju i dohvaćanju podataka sa senzora, a potom slijedi dohvat podataka iz RF modula (temperatura i razina napona napajanja). Podaci se formiraju u polje koje čini podatkovni paket spreman za bežični prijenos. Izgled gotovog podatkovnog paketa senzorske jedinice prikazan je slikom (Slika 6), a objašnjenje pojedinog elementa dano je tablicom (Tablica 2). Svaki element predstavlja 8 – bitni podatak (1 bajt), a broj ispod njega lokaciju u polju.

JIB <sub>MSB</sub>	JIB <sub>LSB</sub>	W1 <sub>MSB</sub>	W1 <sub>LSB</sub>	W2 <sub>MSB</sub>	W2 <sub>LSB</sub>	RF_temp	RF_batt	KB
0	1	2	3	4	5	6	7	8

**Slika 6** – Podatkovni paket senzorske jedinice

**Tablica 2 – Legenda (Senzorska jedinica)**

Element	Objašnjenje
JIB	Jedinstveni identifikacijski broj senzorske jedinice. Predstavlja njenu adresu prilikom prozivanja jedinica. Sastoji se od 2 bajta (MSB i LSB)
W1	Očitavanje senzora na kanalu 1 Sastoji se od 2 bajta (MSB i LSB)
W2	Očitavanje senzora na kanalu 2 Sastoji se od 2 bajta (MSB i LSB)
RF_temp	Temperatura izmjerena integriranim temperaturnim senzorom RF modula
RF_batt	Iznos napona napajanja (napon baterije) izmjeran 5-bitnim ADC pretvornikom RF modula
KB	Kontrolni bajt. Sadrži informacije o događajima prilikom prethodnog transfera podatkovnog paketa.

Nakon konačno formiranog podatkovnog paketa, isti pohranjujemo u FIFO registar RF modula. Potom slijedi upis adrese ciljane jedinice (centralna jedinica sa adresom 0x00) te prebacivanje RF modula u TX mod čime započinje bežični prijenos. Po završetku prijensa svih podataka iz FIFO registra, prebacujemo RF modul u RX mod gdje čekamo odgovor o uspješnosti prijensa paketa. Ukoliko odgovor ne bude primljen u vremenu od jedne sekunde, ponavljamo ciklus slanja podatkovnog paketa. Ponavljanje ciklusa može se dogoditi pet puta za redom prije nego senzorska jedinica ne odustane i vrati se u *sleep* stanje.

U slučaju uspješno primljenog odgovora, analiziramo primljeni podatak i ovisno o njegovoj vrijednosti poduzimamo određenu akciju (Tablica 3) prije vraćanja u *sleep* stanje.



**Tablica 3 – Tumačenje odgovora senzorske jedinice**

Kôd odgovora	Akcija
0xAA	<ul style="list-style-type: none"><li>➤ Neutralni odgovor</li><li>➤ Očisti kontrolni bajt (KB=0).</li></ul>
0xDX	<ul style="list-style-type: none"><li>➤ Preprogramiraj snagu odašiljanja RF modula (vrijednost snage sadržana je u X).</li><li>➤ Postavi treći bit kontrolnog bajta (KB =(1&lt;&lt;3))</li></ul>
0(EX	<ul style="list-style-type: none"><li>➤ Preprogramiraj WUT period RF modula (vrijednost perioda sadržana je u X).</li><li>➤ Postavi drugi bit kontrolnog bajta (KB =(1&lt;&lt;2))</li></ul>
Kôd nije primljen	<ul style="list-style-type: none"><li>➤ Postavi nulti bit kontrolnog bajta (KB =(1&lt;&lt;0))</li></ul>

Moguće je da uslijed smetnje ili poremećaja rada senzorske jedinice, ugradbeno računalo detektira lažan prekid. Tada se postavlja prvi bit kontrolnog bajta (KB|=(1<<1)) i ugradbeno računalo se vraća u *sleep* stanje.

Centralna jedinica će tek nakon drugog buđenja senzorske jedinice imati informaciju o uspješnosti (kontrolni bajt - KB) prethodne komunikacije sa senzorskom jedinicom. Ova opcija koristi se u svrhu testiranja sustava.

Kompletna shema (Slika 6) i nacrt za tiskanu pločicu senzorske jedinice, izrađeni su programskim paketom Altium Designer (verzija: Summer 09).



## 5 Centralna jedinica

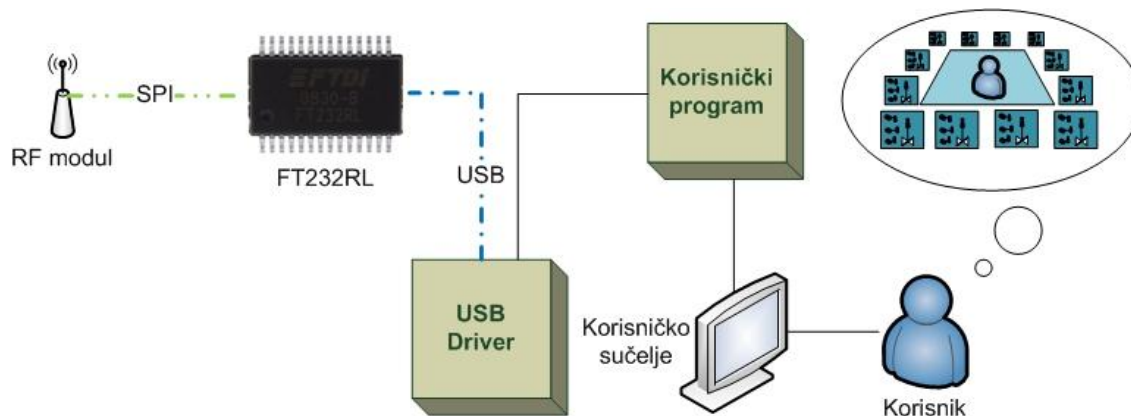


Centralna jedinica podrazumijeva računalo s upravljačkim programom i vanjski periferni sklop koji omogućava komunikaciju s M izvršnih jedinica i M\*N senzorskih jedinica (prema poglavlju: Zahtjevi i osnovni princip rada).

Primopredajni modul koji koristi periferni sklop centralne jedinice isti je kao onaj ugrađen u senzorsku i izvršnu jedinicu, a jedina komunikacija s modulom je preko SPI protokola. Računalo, s druge strane, ne podržava SPI protokol, ali zato svako računalo podržava USB protokol. Za povezivanje RF modula s računalom potrebno je koristiti USB/SPI pretvornik.

Kao USB/SPI pretvornik iskorišten je integrirani krug FT232RL proizvođača FTDI (Future Technology Devices International Limited), koji je po svojoj primarnoj funkciji USB/RS232 pretvornik. Ipak, novija verzija nudi jedan poseban način rada u kojem je moguće računalom direktno upravljati, naponskim stanjima njegovih izvoda (*Bit-Bang mod*). Ta mogućnost iskorištena je za programsku emulaciju SPI protokola.

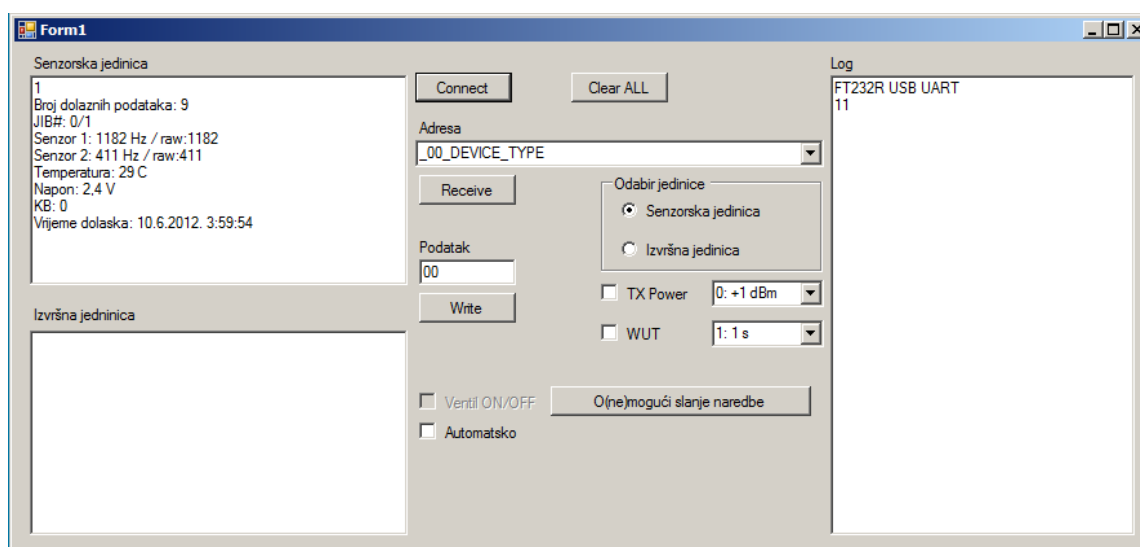
U konačnici centralna jedinica jest lanac redom spojenih sklopovskih i programskih komponenata (Slika 7): RF modul – FT232RL(*Bit-Bang mod*) – USB sabirnica – USB upravljački programski modul (*driver*) – Korisnički program – Korisničko sučelje.



**Slika 7** – Shema centralne jedinice

Korisnički program izrađen je pomoću razvojnog okruženja Microsoft Visual studio 2010 u programskom jeziku C#. Kako je programski jezik C# objektno orijentirani jezik, komunikacija preko vrlo kompleksnog USB protokola ostvarena je koristeći gotove objekte što je uvelike olakšalo i ubrzalo izradu programskog dijela centralne jedinice. Gotovi objekti, prilagođeni za komunikaciju s FT232RL integriranim krugom, dolaze s upravljačkim programom kojeg je moguće besplatno nabaviti preko Internet stranica proizvođača.

Korisnički program izrađen je tako da se u jednom prozoru može pratiti stanje cijelog sustava (Slika 8). Pristigli podaci sa senzorske ili izvršne jedinice mogu se pratiti u istoimenim okvirima. U okviru „Log“ zapisuju se povijest nekih događaja tokom rada korisničkog programa.



**Slika 8** – Korisnički program

U središnjem dijelu korisničkog programa nalazi se kontrolna ploča preko koje upravljamo sustavom. Objašnjenja pojedinih tipki i opcija prikazana su tablicom (Tablica 4).

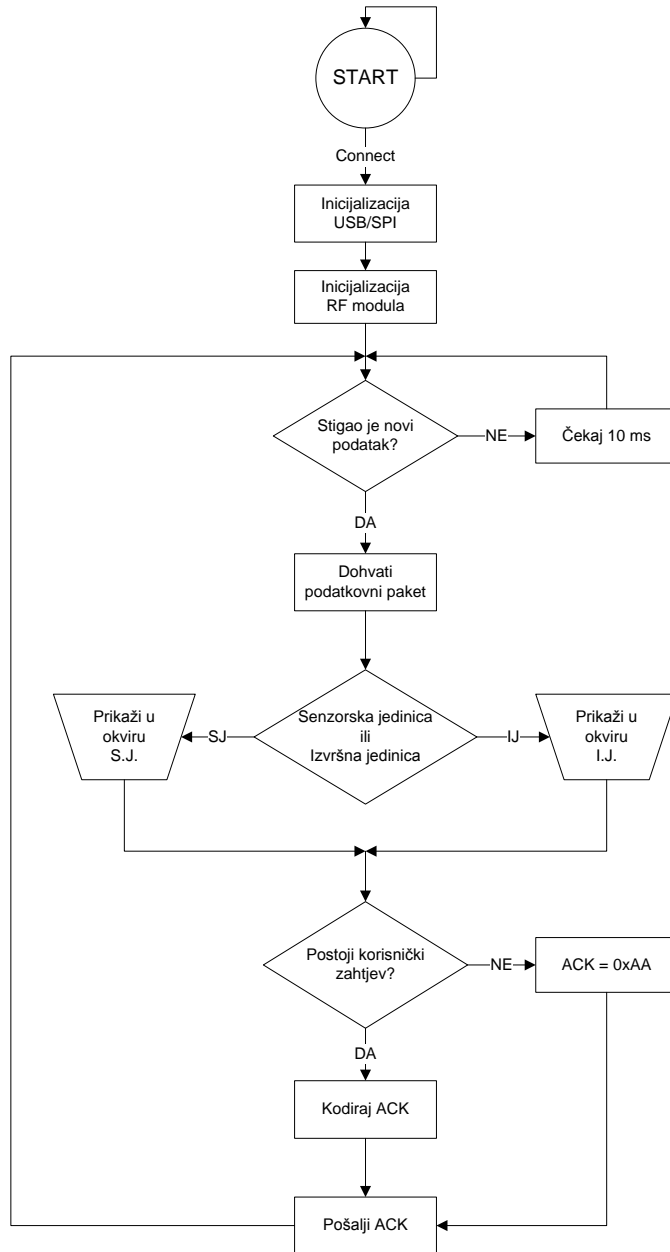
**Tablica 4 – Objašnjenje tipki i opcija**

Naziv tipke/opcije	Objašnjenje
Connect	Pri pokretanju programa kontrolna ploča je neaktivna dok se ne pritisne <i>Connect</i> . Po pritisku dolazi do inicijalizacije USB protokola, a potom se inicijalizira RF modul.
Clear ALL	Briše sadržaj svih okvira ( <i>Log, Senzorska jedinica, Izvršna jedinica</i> ).
Receive	Dohvaća podatak iz registra RF modula, odabranog u padajućem izborniku <i>Adresa</i> te ga ispisuje unutar okvira <i>Log</i> .
Write	Zapisuje podatak, upisan u polje <i>Podatak</i> , u registar RF modula odabran padajućim izbornikom <i>Adresa</i> .
O(ne)mogući slanje naredbe	Uključuje ili isključuje opciju <i>Ventil ON/OFF</i>
Ventil ON/OFF	Kada je omogućena ova opcija tada se ovisno o zastavici šalje impuls za paljenje ili gašenje udaljenog ventila preko izvršne jedinice.
Automatsko	Kada je postavljena zastavica, udaljeni ventil se pali/gasi preko izvršne jedinice ovisno izmjerenoj vrijednosti sa senzora.
TX Power	Ako je postavljena zastavica, podiže se zahtjev za preprogramiranje izlazne snage RF modula, jedinice označene unutar sekcije <i>Odabir jedinice</i> . Nova vrijednost snage, koja se želi preprogramirati, odabire se padajućim izbornikom desno od ove zastavice.
WUT	Ako je postavljena zastavica, podiže se zahtjev za preprogramiranje perioda buđenja RF modula, jedinice označene unutar sekcije <i>Odabir jedinice</i> . Nova vrijednost perioda, koji se želi preprogramirati, odabire se padajućim izbornikom desno od ove zastavice.

Centralna jedinica, nakon podizanja korisničkog programa i pritiska na tipku *Connect*, u *Log* okviru prikazuje poruku o uspješnoj inicijalizaciji svih sklopovskih i programskih modula. U tom trenutku, RF modul nalazi se u RX stanju (stanje prijema) i čeka podatak iz bilo koje jedinice. Kada primi podatak, podiže NIRQ izvod u aktivno stanje (logička '0'), što korisnički program detektira periodičkom rutinom za provjeru. Slijedi dohvat podatkovnog paketa iz FIFO registra RF modula preko USB/SPI protokola. Provjeravaju se nulti i prvi bajt paketa (jedinstveni identifikacijski broj jedinice – JIB) prema kojima se identificira, radi li se o izvršnoj ili senzorskoj jedinici. Senzorske jedinice vrijednost bajta  $JIB_{MSB}$  imaju

jednak nuli, dok izvršne jedinice vrijednost  $JIB_{LSB}$  bajta imaju jednak nuli. Prema tome se podaci paketa obrađuju i zapisuju u odgovarajući okvir korisničkog programa.

Slijedi prebacivanje RF modula u TX stanje i slanje odgovora u jedinicu koja je upravo poslala paket i čeka odgovor. Ako korisnik nije zatražio nikakva preprogramiranja jedinice ili upravljanje ventilima kod izvršne jedinice, centralna jedinica odgovara s neutralnim podatkom 0xAA. U suprotnom, ako postoji zahtjev za slanje naredbe u jedinicu, odgovor centralne jedinice biti će kodirani podatak razumljiv jedinici kojoj se šalje prema odgovarajućim tablicama (Tablica 3 i Tablica 6). Sada se centralna jedinica prebacuje u RX stanje gdje čeka dolazak novog podatka. Na slici (Slika 9) prikazan je pojednostavljeni algoritam centralne jedinice.



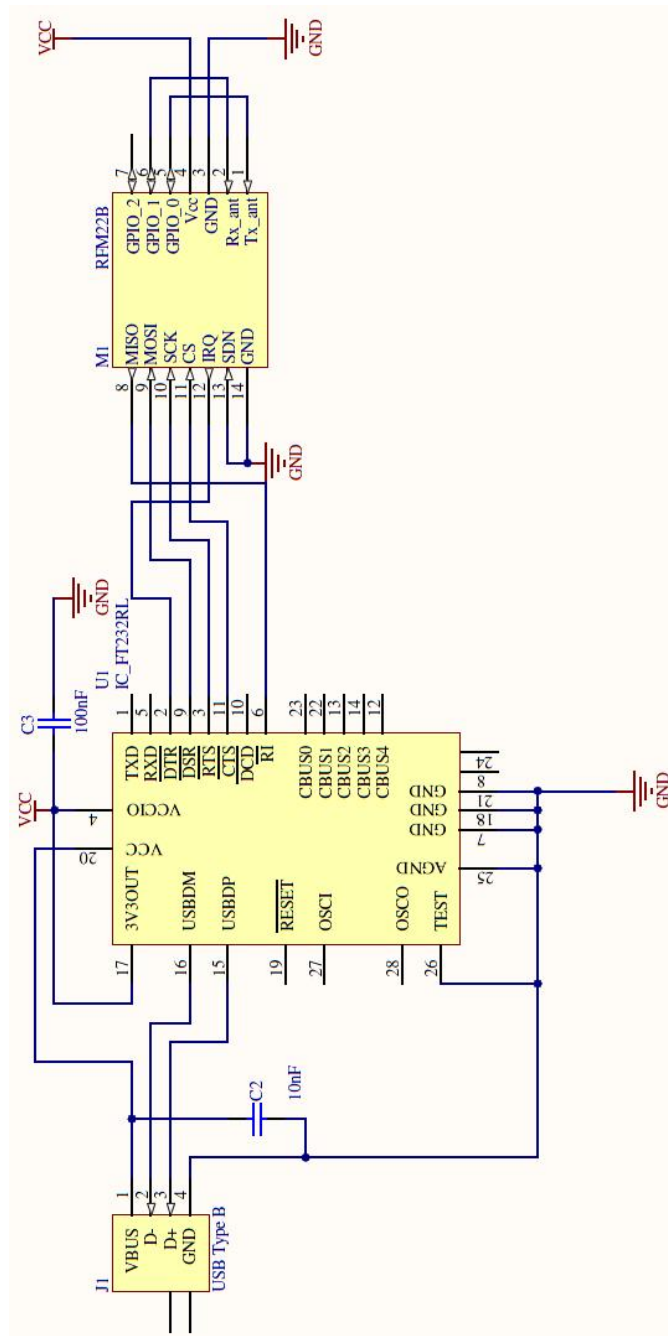
**Slika 9** – Algoritam centralne jedinice

Prilikom preprogramiranja izlazne snage ili WUT vremenskog sklopa, preprogramirana jedinica postavit će odgovarajući bit kontrolnog bajta (KB). U slijedećem javljanju jedinice, isti će biti vidljiv centralnoj jedinici koja tada javlja u *Log* okviru poruku o uspješnosti.

Matematička obrada podataka sa *Watermark* senzora tj. naponska kompenzacija frekvencije objašnjena u poglavlju *Z/f* pretvornik, trenutno je implementirana u centralnoj jedinici. Razlog tome je veća procesorska moć računala. Ista obrada u ugradbenom računalu zauzimala podosta programske

memorije, ali i procesorskog vremena što dovodi do veće potrošnje energije iz baterije. Stoga se u okviru *Senzorska jedinica*, uz oznaku *Senzor X*: prikazuje prvo kompenzirana vrijednost, a potom sirovi, nekompenzirani podatak iz senzorske jedinice (*raw*).

Kompletna shema (Slika 10) i nacrt za tiskanu pločicu sklopovskog dijela centralne jedinice, izrađeni su programskim paketom Altium Designer (verzija: Summer 09).



**Slika 10** – Shema sučelja centralne jedinice

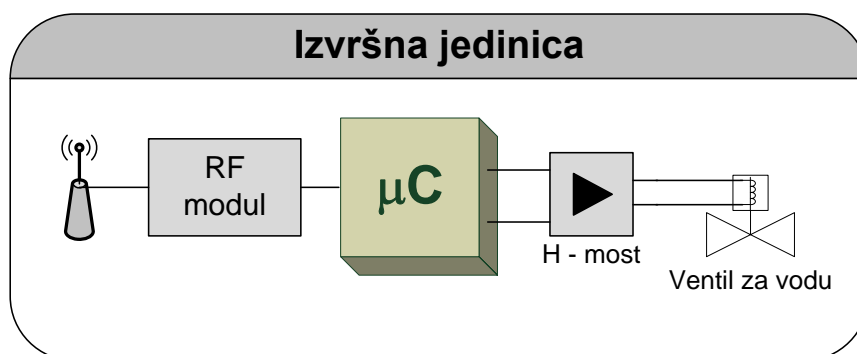


## 6 Izvršna jedinica

Ideja automatskog sustava za navodnjavanje jest potpuno automatizirano zalijevanje poljoprivredne površine. Osim toga, automatizacijom možemo nadomjestiti potrebu za ljudskom radnom snagom. Ta nova snaga su strojevi i upravljački sklopovi. Upravo izvršna jedinica igra glavnu ulogu u zamjeni ljudsko – strojno. Ona upravlja ventilom za vodu, otvarajući ga i zatvarajući na zahtjev iz centralne jedinice. Na ulaz ventila dovodi se voda pod pritiskom iz bilo kojeg izvora, a izlaz je povezan sa sustavom kapaljki koje polako natapaju tlo kada je ventil otvoren.

Shema izvršne jedinice prikazana je slikom (Slika 11). Princip rada sličan je senzorskoj jedinici, osim što ugradbeno računalo ne prikuplja podatke iz okoline već indirektno djeluje na nju upravljajući ventilom za vodu. Izvršna jedinica također je ostvarena kao bežični čvor i koristi isti RF modul kao i ostale dvije vrste jedinica.

Za razliku od ostalih jedinica, izvršna mora upravljati robusnim ventilom za čije je upravljanje potrebna relativno velika snaga. Zbog toga izvršna jedinica koristi dva izvora napajanja. Dvije serijski spojene baterije (2.4 V) za napajanje RF modula i ugradbenog računala i jednu bateriju od 9 V iz koje se uzima energija za upravljanje ventilom.

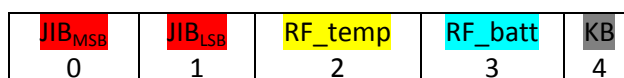


**Slika 11** – Izvršna jedinica

Regulacija toka energije potrebnog za paljenje odnosno gašenje ventila ostvaruje se H-mostom. Korišten je H – most BD222 proizvođača ROHM Semiconductor. Radi se o jedno-kanalnom mostu koji radi na naponu napajanja do 18V i moguć je tok struje od maksimalno 2 A, što zadovoljava potrebe

elektromagnetnog ventila za vodu. H-mostom upravlja ugradbeno računalo i to u PWM načinu rada kako bi se minimizirala potrošnja energije.

Ugradbeno računalo nalazi se u *sleep* stanju sve do isteka WUT vremenskog sklopa RF modula. Tada se prebacuje u aktivno stanje, prikuplja podatke iz RF modula, formira podatkovni paket (Slika 12, Tablica 5) te ga šalje u centralnu jedinicu.



**Slika 12** – Podatkovni paket senzorske jedinice

**Tablica 5** – Legenda (Izvršna jedinica)

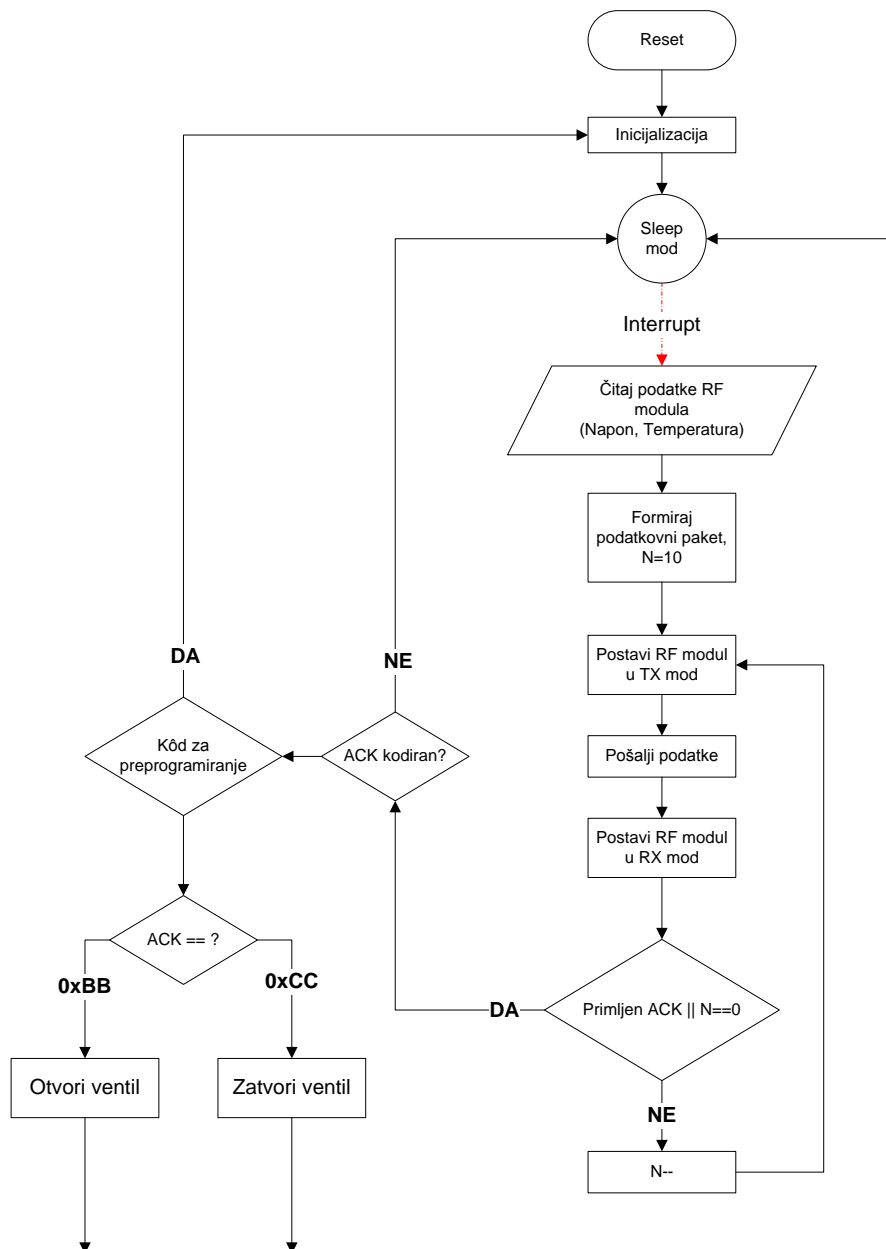
Element	Objašnjenje
JIB	Jedinstveni identifikacijski broj senzorske jedinice. Predstavlja njenu adresu prilikom prozivanja jedinica. Sastoji se od 2 bajta (MSB i LSB)
RF_temp	Temperatura izmjerena integriranim temperaturnim senzorom RF modula
RF_batt	Iznos napona napajanja (napon baterije) izmjeren 5-bitnim ADC pretvornikom RF modula
KB	Kontrolni bajt. Sadrži informacije o događajima prilikom prethodnog transfera podatkovnog paketa.

Ugradbeno računalo od centralne jedinice očekuje kodirani odgovor tj. naredbu za otvaranje ili zatvaranje ventila. Moguć je neutralni odgovor, kada ne poduzima nikakvu akcija i vraća se u *sleep* stanje. Ako ne dobije nikakav odgovor, ponovno šalje podatkovni paket i čeka odgovor, pa tako deset puta, dok ne odustane i vrati se u *sleep* stanje. Za razliku od toga, senzorska jedinica ponavlja ciklus samo pet puta. Kod izvršne jedinice ne smije se dogoditi propust zatvaranja ili otvaranja ventila, pa se opreza radi ovaj ciklus ponavljanja udvostručava. Tumačenje odgovora izvršne jedinice nalazi se u tablici (Tablica 6).

**Tablica 6 – Tumačenje odgovora izvršne jedinice**

Kôd odgovora	Akcija
0xAA	<ul style="list-style-type: none"><li>➤ Neutralni odgovor</li><li>➤ Očisti bitove 0, 2 i 3 kontrolnog bajta</li></ul>
0xBB	<ul style="list-style-type: none"><li>➤ Zahtjev za otvaranje ventila</li><li>➤ Postavi sedmi bit kontrolnog bajta (KB  = (1 &lt;&lt; 7))</li></ul>
0xCC	<ul style="list-style-type: none"><li>➤ Zahtjev za zatvaranje ventila</li><li>➤ Očisti bit 7 kontrolnog bajta</li></ul>
0xDX	<ul style="list-style-type: none"><li>➤ Preprogramiraj snagu odašiljanja RF modula (vrijednost snage sadržana je u X).</li><li>➤ Postavi treći bit kontrolnog bajta (KB  = (1 &lt;&lt; 3))</li></ul>
0(EX	<ul style="list-style-type: none"><li>➤ Preprogramiraj WUT period RF modula (vrijednost perioda sadržana je u X).</li><li>➤ Postavi drugi bit kontrolnog bajta (KB  = (1 &lt;&lt; 2))</li></ul>
Kôd nije primljen	<ul style="list-style-type: none"><li>➤ Postavi nulti bit kontrolnog bajta (KB  = (1 &lt;&lt; 0))</li></ul>

Komunikacija ugradbenog računala i RF modula identična je onoj u senzorskoj jedinici, vidi sliku (Slika 4). Algoritam programa upisanog u ugradbeno računalo izvršne jedinice prikazan je slikom (Slika 13).

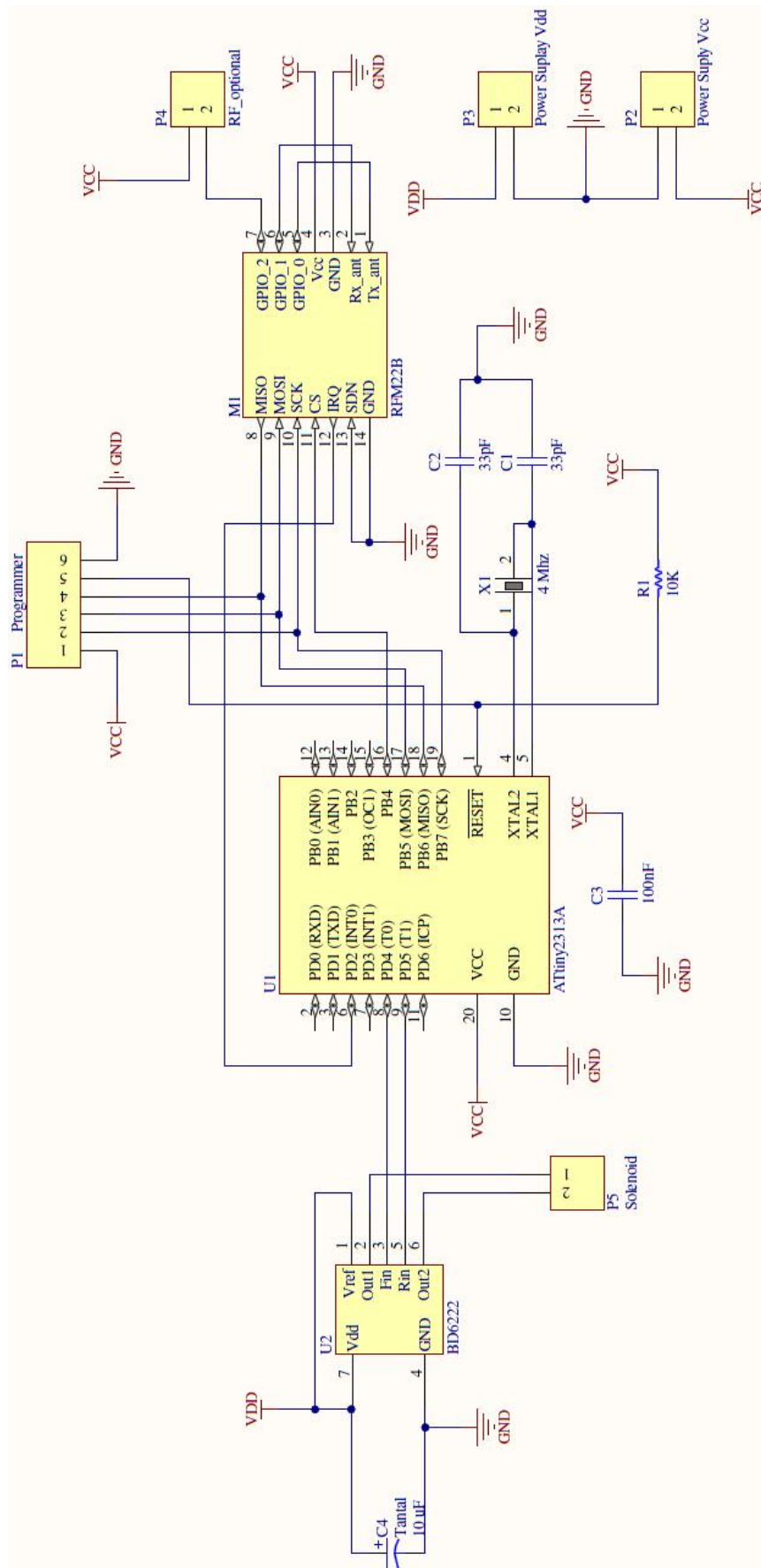


**Slika 13** - Algoritam ugradbenog računala u izvršnoj jedinici

Moguće je da uslijed smetnje ili poremećaja rada izvršne jedinice, ugradbeno računalo detektira lažan prekid. Tada se postavlja prvi bit kontrolnog bajta ( $KB_1 = 1 < 1$ ) i ugradbeno računalo se vraća u *sleep* stanje.

Centralna jedinica će tek nakon drugog buđenja izvršne jedinice imati informaciju o uspješnosti (kontrolni bajt - KB) prethodne komunikacije sa izvršnom jedinicom. Ova opcija koristi se u svrhu testiranja sustava.

Kompletna shema (Slika 14) i nacrt za tiskanu pločicu izvršne jedinice, izrađeni su programskim paketom Altium Designer (verzija: Summer 09).



Slika 14 – Shema izvršne jedinice

## 7 Napajanje jedinica i potrošnja

Prema osnovnim zahtjevima ovog sustava, napajanje udaljenih jedinica mora biti neovisno o dostupnosti električne mreže. Zato je odabrano baterijsko napajanje i to punjivim baterijama sa maksimalnim mogućim kapacitetom uz uvjet da su tipa LSD (*low self-discharge*) - smanjeno samopražnjenje.

Korištene su NiMh punjive baterije proizvođača Camelion zbog dobrog omjera cijene i kapaciteta. Sadrže oznaku „*Always Ready*“ što je jedna od oznaka za LSD tipove punjivih baterija. Fotografija odabrane punjive baterije prikazana je slikom (Slika 15), a tablicom (Tablica 7) prikazane su značajke koje daje proizvođač.



**Slika 15** – Punjiva baterija (AA)

**Tablica 7** – Značajke punjive baterije (AA)

<b>Battery Code:</b>	1.2V, AA		
<b>IEC Code:</b>	AlwaysReady	<b>ANSI Code:</b>	
<b>Size:</b>	AA	<b>Voltage:</b>	1.2V
<b>Capacity:</b>	2500mAh	<b>Chemistry:</b>	Ni-MH
<b>Height:</b>	50.5 mm	<b>Diameter:</b>	14.1 mm
<b>Length:</b>	N/A	<b>Volume:</b>	
<b>Width:</b>	N/A	<b>Weight:</b>	29.0 gram

Nazivna snaga punjive baterije iznosi 1.2 V, a najviša vrijednost minimalnog napona napajanja komponenata iznosi 2 V. Stoga serijski spajamo dvije baterijske ćelije te dobivamo nazivni napon izvora napajanja 2.4 V. Istim se napajaju sve komponente senzorske i izvršne jedinice uz izuzetak solenoida koji za svoj rad treba nešto veće energije, pa se koristi punjiva baterija nazivnog napona 8.4 V (Slika 16). Centralna jedinica napaja se direktno iz USB sabirnice koja ovisi o računalu i neće se razmatrati njena potrošnja jer se pretpostavlja da centralna jedinica ne oskudijeva električnom energijom.



<b>Battery Code:</b>	1.2V, 9V		
<b>IEC Code:</b>	Nickel Metal Hydride	<b>ANSI Code:</b>	
<b>Size:</b>	9V	<b>Voltage:</b>	8.4V
<b>Capacity:</b>	250 mAh	<b>Chemistry:</b>	Ni-MH
<b>Height:</b>	48.5 mm	<b>Diameter:</b>	N/A
<b>Length:</b>	26.2 mm	<b>Volume:</b>	
<b>Width:</b>	15.2mm	<b>Weight:</b>	42.0 gram

**Slika 16** – Punjiva baterija (9V)

U tablici (Tablica 8) nalazi se pregled potrošnje po stanjima svih komponenata koje utječu na potrošnju energije iz baterije. Korištene su maksimalne pretpostavljene vrijednosti, kako bi se mogla aproksimirati ukupna potrošnja za najgori slučaj. Pretpostavljeno vrijeme periodičnog buđenja jedinice iznosi 10 minuta. Nakon isteka perioda, pojedina jedinica se budi ostaje aktivna u prosjeku 2 sekunde i vraća se u mirovanje. Ugradbeno računalo ATtiny2313A senzorske jedinice radi na taktu frekvencije 4 MHz, a izvršne jedinice na taktu frekvencije 8 MHz.

**Tablica 8** – Pregled potrošnje struje komponenata

Komponenta	Stanje	Iznos maksimalne struje	Vrijeme trajanja
Z/f pretvornik: <b>TLC556cd</b>	Aktivan	900 $\mu$ A	2 s
	Sleep	0 A	10 min
Ugradbeno računalo: <b>ATtiny2313A</b>	Aktivan, 4 Mhz	2,5 mA	3 s
	Aktivan, 8 Mhz	7 mA	2 s
	Sleep	2 $\mu$ A	10 min
RF modul: <b>RFM22b</b>	TX	85 mA	2 ms
	RX	18,5 mA	1 s
	Sleep	1 $\mu$ A	10 min
Solenoid i H-most: <b>BD6222</b>	Otvaranje	960 mA	60 ms
	Zatvaranje	480 mA	60 ms
	Sleep	10 $\mu$ A	10 min

Iz tablice (Tablica 8) možemo odrediti prosječnu struju za vrijeme aktivnog stanja  $I_A$ , i prosječnu struju za vrijeme *sleep* stanja  $I_S$  i to za svaku jedinicu posebno. Nakon toga određujemo srednju struju ciklusa prema formuli (Formula 3). Rezultati su izraženi tablicom (Tablica 9).

**Tablica 9** – Srednja potrošnja

Jedinica	Stanje	Prosječna struja	Trajanje	Srednja vrijednost struje
Senzorska jedinica	Aktivno stanje	50 mA	3 s	<b>250 <math>\mu</math>A</b>
	Sleep stanje	3 $\mu$ A	600 s	
Izvršna jedinica (sklopovlje)	Aktivno stanje	60 mA	2 s	<b>202 <math>\mu</math>A</b>
	Sleep stanje	3 $\mu$ A	600 s	
Izvršna jedinica (solenoid)	Aktivno stanje	720 mA	60 ms	<b>82 <math>\mu</math>A</b>
	Sleep stanje	10 $\mu$ A	600 s	

$$I_{sr} = \frac{I_A T_A + I_S T_S}{T_A + T_S} \quad (3)$$

Ako koristimo punjive baterije kapaciteta 2500 mAh za senzorsku i izvršnu jedinicu, odnosno 250 mAh za dio izvršne jedinice koja upravlja solenoidom tada prema vrijednostima u tablici (Tablica 9) možemo aproksimirati vrijeme trajanja punjive baterije (Tablica 10). Za sigurniju procjenu, dobivene vrijednosti smanjimo za oko 30% zbog utjecaja samopražnjenja (Formula 4).

$$T_{uk} = \frac{C_{bat} [mAh]}{I_{sr} [mA]} * 0.7 [h] \quad (4)$$

**Tablica 10** – Procjena trajanja jedinica do ponovnog punjenja

Jedinica	Srednja vrijednost struje	Vrijeme trajanja punjive baterije	Vrijeme trajanja punjive baterije uz samopražnjenje
Senzorska jedinica	250 mA	417 dana	<b>300 dana</b>
Izvršna jedinica (sklopovlje)	202 mA	516 dana	<b>385 dana</b>
Izvršna jedinica (solenoid)	82 mA	150 dana	<b>100 dana</b>



Izvršna jedinica ima nisku vrijednost zbog manjeg kapaciteta punjive baterije od 9 V, ali i zbog stroge aproksimacije u kojoj se podrazumijeva da će se izvršna jedinica nakon svakog buđenja upaliti ili ugasiti ventil. U praksi to neće biti tako, već će se pobuđivanje ventila vršiti najviše jednom dnevno. U tom će slučaju, vrijeme trajanja punjive baterije biti višestruko uvećano.

Možemo zaključiti da će sve komponente sustava imati dovoljno električne energije kroz čitavu sezonu uzgoja poljoprivredne kulture, što je jedan od osnovnih zahtjeva ovog sustava.

## 8 Opis komponenata

### 8.1 Watermark senzor

*Watermark* senzor jedinstveni je senzor koji se zbog svojih prednosti koristi još od 1978. godine. Ono što ga čini boljim od ostalih senzora jest neosjetljivost na vrste tla tj. različite pH vrijednosti i mogućnost mjerenja vlage tla u cb ili kPa.

Senzor je građen uglavnom od gipsa i pijeska koji služe kao filtar za kiselosti i soli koje se nalaze u tlu. Unutar senzora nalaze se dvije elektrode od nerđajućeg metala. Između elektroda nalazi se filtrirana voda iz tla te se senzor ponaša kao otporni element. Cijela struktura senzora spremljena je u rešetkasto metalno kućište od visokootpornog materijala na kiselosti i soli što daje senzoru izuzetnu čvrstoću i izdržljivost. Izgled senzora prikazan je slikom (Slika 17), a potom slijede specifikacije koje daje proizvođač.



**Slika 17** – Watermark senzor

- Mjerni opseg od 0 kPa do 200 kPa
- Visoke čvrstoće i izdržljivosti
- Ne otapa se u solima
- Otporan na smrzavanje
- Kompenziran na utjecaj saliniteta

- Jeftin
- Jednostavan za rukovanje
- Kompatibilan sa AC i DC specijaliziranim čitačima
- Nema potrebe za održavanjem
- Dugi vijek trajanja

Za razliku od ostalih senzora koji mjere vodljivost tla kao parametar vlažnosti tla, *Watermark* senzor oponaša korijen biljke, a informacija koja se može dobiti njegovim očitanjem jest tlak vode na korijen. Ukoliko je tlak manji tada korijenu treba manja sila kako bi povukao vodu iz tla, dok za veći tlak treba veća sila. Stanje velike vlažnosti tla odgovara malom tlaku, a suhoća je popraćena većim vrijednostima tlaka, vidi tablicu (Tablica 1).

Jedini problem koji ostaje kod ovog senzora jest realizacija mjernog sklopa čiji će izlazi biti digitalni i kompatibilni s ugradbenim računalom. Taj je problem riješen *Z/f* pretvornikom koji je detaljno opisan u nastavku.

Skaliranje frekvencije u kPa ili cb vrši se baždarenjem pomoću *Watermark meter* instrumenta, također proizvođača Irrrometer (Slika 18).



**Slika 18** – Watermark meter

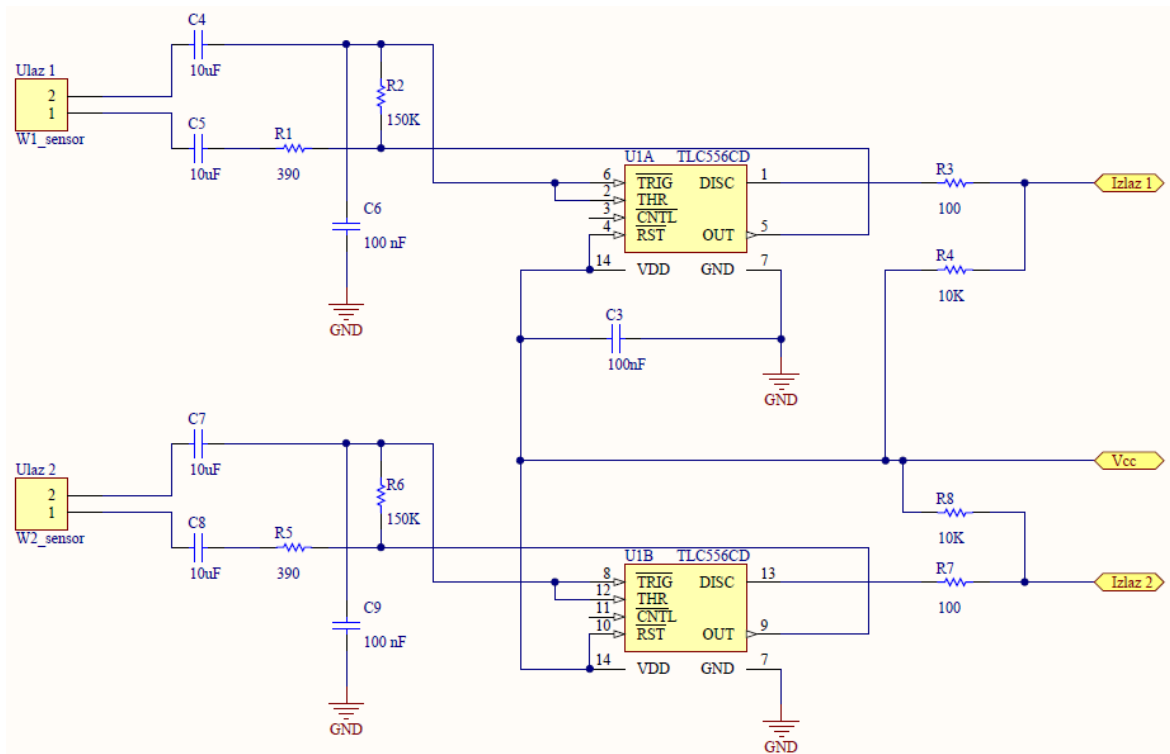
## 8.2 Z/f pretvornik

Z/f pretvornik je sklop koji impedanciju senzora pretvara u frekvenciju koju možemo mjeriti ugradbenim računalom. Pretvornik je realiziran pomoću poznatog generatora takta, NE555 integriranog kruga. Isti, se najčešće koristi za generiranje takta stabilne frekvencije koja se fiksno namješta preko vanjskih RC komponenti. Kako je korišteni senzor upravo RC komponenta, možemo ga koristiti za namještanje frekvencije NE555 integriranog kruga. Promjenom vlažnosti tla, mijenja se impedancija senzora, a time i frekvencija na izlazu. Ugradbenim računalom možemo prebrojati impulse te frekvencije, u nekom okviru, što nam daje cijeli broj proporcionalan frekvenciji. Time smo ostvarili kompletni mjerni lanac od parametra okoline (vlažnost tla) do podataka u ugradbenom računalu.

Senzorska jedinica projektirana je s dva ulaza pa se umjesto integriranog kruga NE555 koristi TLC556CD integrirani krug. TLC556CD jest novija verzija NE555 i u jednom kućištu nalaze se dva jednaka integrirana kruga tj. dva ulaza i dva izlaza. U novijoj verziji, također je smanjen minimalni napon napajanja do 2V što je jedan od osnovnih uvjeta kod odabira komponenata za ovaj sustav.

Na slici (Slika 19) dana je shema sklopa, a formulama (Formula 5, 6 i 7) aproksimiramo frekvenciju za Izlaz 1 (analogno dobivamo formulu za Izlaz 2). Razlika izračunate i stvarne frekvencije ovisi o više parametara: napon napajanja, temperatura, frekvencija.

Utjecaj temperature može se zanemariti. Frekvencija je vrlo nelinearna pa će se izvršiti baždarenje u centibare i/ili kilopascale. Utjecaj promjene napona na frekvenciju je značajan i potrebna je kompenzacija.



Slika 19 – Shema Z/f pretvornika

$$f_o \approx (2 * T_o)^{-1} \quad (5)$$

$$T_o = C_6 * \{(R_1 + Z) || R_2\} * \ln 2 \quad (6)$$

$$Z \approx 2 * X_{C_4} + R_w \quad (7)$$

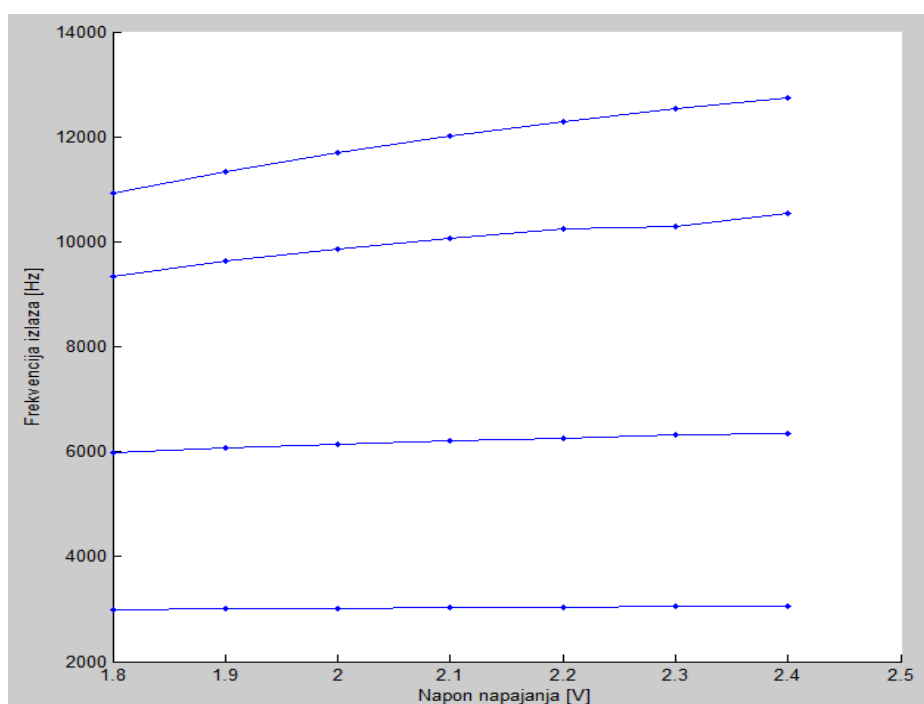
$R_w$  – Otpor senzora

Naponska kompenzacija proračunata je na temelju laboratorijskih mjerenja. Za nekoliko različitih frekvencija izmjerena je njena ovisnost o naponu. Promjena frekvencije vrši se pomoću trimer otpornika koji simulira otpor Watermark senzora. Za vrijeme mjerenja, Z/f pretvornik spojen je na podesivi regulator napona realiziran pomoću LM317 integriranog kruga. Finim trimer otpornikom podešavan je napon od 2.4 V pa do 1.8 V u koracima po 0.1 V. Takvo mjerenje simulacija je realnih uvjeta kod kojih napunjena baterija (NiMh) pada od svoje nazivne vrijednosti do vrijednosti kada je proglašavamo praznom (otprilike 1V po ćeliji).

Tablicom (Tablica 11) dan je pregled mjerenja Z/f pretvornika, a slikom (Slika 20) grafički prikazi na temelju kojih se odlučuje o matematičkom modelu.

**Tablica 11 – Pregled mjerenja**

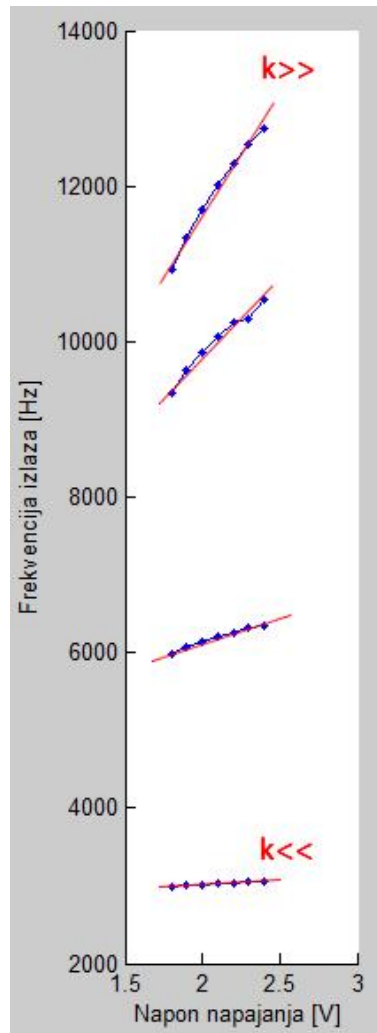
Naziv	Početna frekvencija [Hz]	Maksimalna devijacija [Hz]
Potpuno suhi senzor	51	0
Međufrekvecija 1	909	6
Međufrekvecija 2	3052	69
Međufrekvecija 3	6339	364
Međufrekvecija 4	10542	1197
Međufrekvecija 5	12742	1822
Potpuno mokri senzor	14187	3204



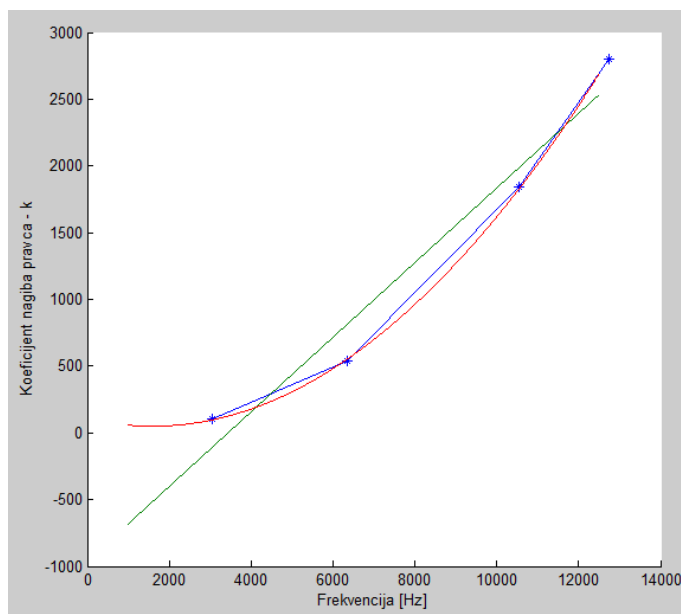
**Slika 20 – Rezultati mjerenja Z/f pretvornika**

Iz slike (Slika 20) možemo primijetiti kako s porastom frekvencije raste i nelinearnost odnosno ovisnost o naponu napajanja. Za potrebe mjerenja vlažnosti tla značajan je tek uski pojas od oko 5 kHz pa do 900 Hz, pa prema tome nema potrebe za mjerenje s visokom preciznošću. Zbog toga, ali i zbog jednostavnosti, nelinearan model lineariziramo tako da krivulju za svaku frekvenciju predstavimo pravcem. Tada je lako je uočiti kako s porastom frekvencije raste i nagib pravca. Sužavanjem X-osi sa slike (Slika 20) možemo vizualno izraziti dokaz o porastu koeficijenta pravaca s frekvencijom (Slika 21).

Sada je potrebno izvršiti aproksimaciju funkcije promjene parametra  $k$  u ovisnosti o frekvenciji. Linearna i kvadratna regresija korištene su kao matematički alat aproksimacije. Na slici (Slika 22) prikazane su obje aproksimacije u odnosu na stvarnu funkciju. Primijećujemo kako je kvadratna regresija dala bolju aproksimaciju, pa linearnu odbacujemo. Proračuni su izvedeni korištenjem programskog paketa Matlab. Koeficijente dobivene kvadratnom regresijom zapisujemo te ih koristimo za kompenzacijsku funkciju realiziranu unutar korisničkog programa centralne jedinice. Postupak dobivenih koeficijenata detaljno je opisan u nastavku.



**Slika 21** – Porast koeficijenta pravca



**Slika 22** – Linearna i kvadraturna regresija

### 8.2.1 Postupak dobivanja koeficijenata

Za potrebe proračuna koeficijenata, dobivenih iz podataka laboratorijskog mjerenja (vidi pravitak, Mjerenja za potrebe kompenzacije), potrebne su nam formule za linearnu regresiju (Formule 8, 9 i 10) i formule za kvadraturnu regresiju (Formule 11, 12, 13 i 14).

$$y = \beta x + \alpha \quad (8)$$

$$\beta = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n (x_i^2) - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \quad (9)$$

$$\alpha = \bar{y} - \beta \bar{x} \quad (10)$$

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 \quad (11)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad (12)$$



$$\vec{y} = X * \vec{a} \quad (13)$$

$$\hat{\vec{a}} = (X^T X)^{-1} X^T \vec{y} \quad (14)$$

U prvom koraku želimo nelinearne vektore frekvencije ( $f_{1,n} - f_{M,n}$ ) linearizirati tj. aproksimirati pravac koristeći linearnu regresiju. Matematički model prema formuli (8) definiran je kao (15). Koristeći formulu (9), možemo izračunati koeficijent pravca za pojedini vektor frekvencije (16).

$$f_N(t) = k_N * U_N(t) \quad (15)$$

$$k_n = \frac{\sum_{i=1}^M U_i (f_{1,n} - f_{i,n})}{\sum_{i=1}^M U_i^2} \quad (16)$$

U slijedećim koracima pokušavamo odrediti ovisnost promjene koeficijenta pravca (dobivenih prethodno) s frekvencijom. Postavljamo dva matematička modela: linearan (17) i kvadraturni (20).

Za linearnu aproksimaciju potrebni su nam koeficijenti  $b$  i  $a$  koje lako dobivamo prema formuli (9) i (10), a koje prilagođene izmjerenim podacima poprimaju oblik (18) i (19).

$$k(f) = b * f + a \quad (17)$$

$$b = \frac{\sum_{i=1}^N k_i f_{1,i} - \frac{1}{N} \sum_{i=1}^N k_i \sum_{i=1}^N f_{1,i}}{\sum_{i=1}^N (f_{1,i}^2) - \frac{1}{N} (\sum_{i=1}^N f_{1,i})^2} \quad (18)$$

$$a = \frac{1}{N} \sum_{i=1}^N k_i - b \sum_{i=1}^N f_{1,i} \quad (19)$$

Kvadraturna aproksimacija podskup je općenite polinomne regresije, pa definiramo Vandermondovu matricu  $X$  (22) prema formuli (12). Skup izmjerenih podataka lakše je prikazati vektorski prema formuli (13) pa dobivamo izraz (21). Iz izraza (21) zanimaju nas  $c$  koeficijenti koje možemo estimirati prema formuli (14), te konačno dobivamo izraz (23) koji je formula za dobivanje koeficijenta

kvadratne jednadžbe. Estimacija podrazumijeva slučaj najmanjih kvadrata (*Least square*).

$$k_i = c_0 + c_1 f_{1,i} + c_2 f_{1,i}^2 \quad (20)$$

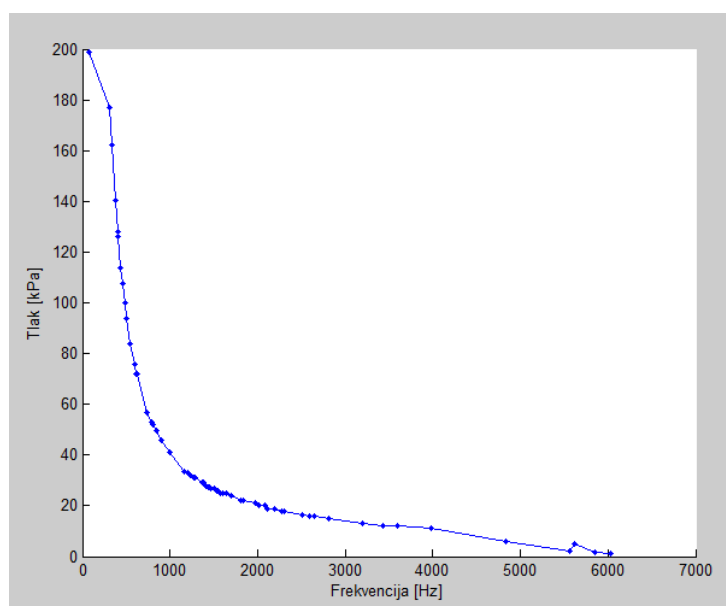
$$\vec{k} = X * \vec{c} \quad (21)$$

$$X = \begin{bmatrix} 1 & f_{1,1} & f_{1,1}^2 \\ \vdots & \vdots & \vdots \\ 1 & f_{1,N} & f_{1,N}^2 \end{bmatrix} \quad (22)$$

$$\vec{c} = (X^T X)^{-1} X^T \vec{k} \quad (23)$$

Na kraju imamo koeficijente linearne aproksimacije a i b te koeficijente kvadraturne aproksimacije c<sub>0</sub>, c<sub>1</sub> i c<sub>2</sub>. Iz grafova funkcija odlučujemo o boljoj aprokimaciji koja se kasnije implementira u sustav za kompenzaciju.

Navedeno rješava problem kompenzacije uslijed osjetljivosti frekvencije s naponom napajanja. Konačna informacija, treba biti izražena u jedinicama tlaka (kPa ili cb). Iz laboratorijskih mjerenja dobivamo tablicu (vidi privitak, Mjerenja za potrebe baždarenja) prema kojoj vršimo baždarenje. Na slici (Slika 23) grafički je prikazana ovisnost kompenzirane frekvencije o tlaku. Informacije o tlaku dobivene su mjerenjem uz pomoć službenog mjernog instrumenta (Slika 18).

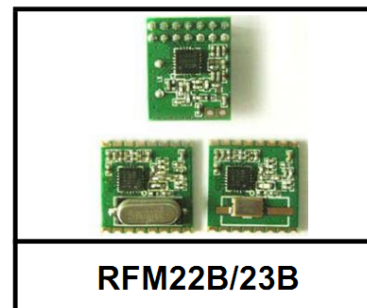


**Slika 23** – Krivulja baždarenja

### 8.3 Primopredajni modul RFM22B

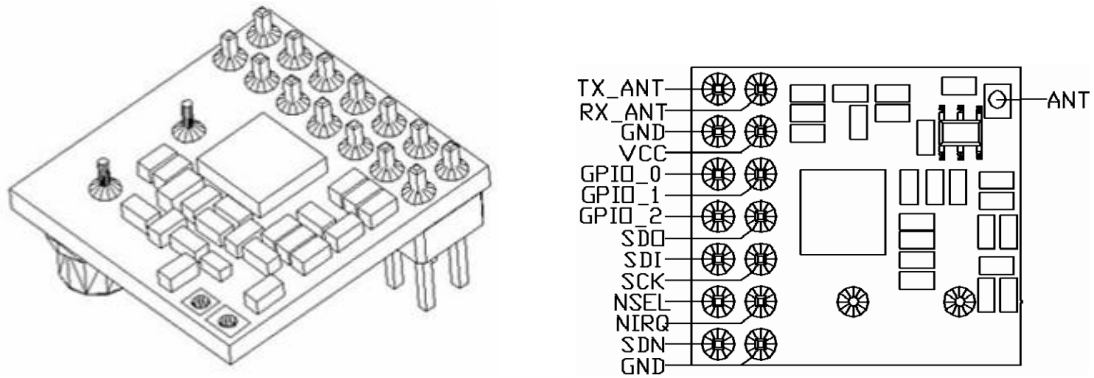
Primopredajni modul RFM22B (u nastavku modul) je najvažnija komponenta i nalazi se u svakoj jedinici. Omogućava komunikaciju i prijenos podataka među jedinicama, a neke njegove posebne mogućnosti iskorištene su svrhu manjeg opterećivanja ugradbenog računala. Jedna takva mogućnost jest periodično buđenje cijele jedinice što omogućava rad jedinice s minimalnom potrošnjom energije. Kratki opis modula prikazan je izvatkom iz službene tehničke dokumentacije proizvođača Hoperf Electronic (Slika 24).

- Frequency Range
  - 433/470/868/915MHz ISM bands
- Sensitivity = -121 dBm
- Output power range
  - +20 dBm Max (RFM22B)
  - +13 dBm Max (RFM23B)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- -40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 14-PIN DIP & 16-PIN SMD package
- Low cost



**Slika 24 – Značajke modula**

Važno je istaknuti kako modul ima nizak napon napajanja (od 1.8V do 3.6V) pa se može napajati iz samo dvije baterije čime je zadovoljen zahtjev niske potrošnje sklopa; Tablicom (Tablica 12) prikazana je potrošnja modula pri različitim modovima rada. Uz to, modul ima programabilno izlazno RF pojačalo koje može izračiti snagu na anteni do +20dBm (100 mW) što pokriva zahtjev za dometom od 1 km.



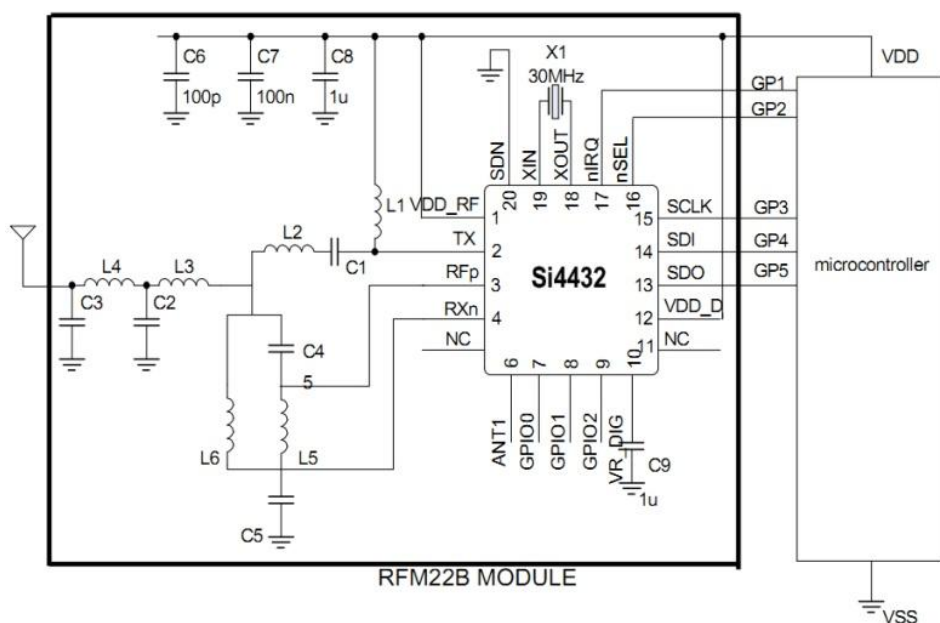
**Slika 25** – Izgled modula (lijevo), opis nožica (desno)

**Tablica 12** – DC karakteristike modula

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Supply Voltage Range	$V_{DD}$		1.8	3.0	3.6	V
Power Saving Modes	$I_{Shutdown}$	RC Oscillator, Main Digital Regulator, and Low Power Digital Regulator OFF	—	15	50	nA
	$I_{Standby}$	Low Power Digital Regulator ON (Register values retained) and Main Digital Regulator, and RC Oscillator OFF	—	450	800	nA
	$I_{Sleep}$	RC Oscillator and Low Power Digital Regulator ON (Register values retained) and Main Digital Regulator OFF	—	1	—	$\mu$ A
	$I_{Sensor-LBD}$	Main Digital Regulator and Low Battery Detector ON, Crystal Oscillator and all other blocks OFF	—	1	—	$\mu$ A
	$I_{Sensor-TS}$	Main Digital Regulator and Temperature Sensor ON, Crystal Oscillator and all other blocks OFF	—	1	—	$\mu$ A
	$I_{Ready}$	Crystal Oscillator and Main Digital Regulator ON, all other blocks OFF. Crystal Oscillator buffer disabled	—	800	—	$\mu$ A
TUNE Mode Current	$I_{Tune}$	Synthesizer and regulators enabled	—	8.5	—	mA
RX Mode Current	$I_{RX}$		—	18.5	—	mA
TX Mode Current —RFM22B	$I_{TX_{+20}}$	txpow[2:0] = 111 (+20 dBm)	—	85	—	mA
TX Mode Current —RFM23B	$I_{TX_{+13}}$	txpow[2:0] = 110 (+13 dBm)	—	30	—	mA
	$I_{TX_{+1}}$	txpow[2:0] = 001 (+1 dBm)	—	18	—	mA

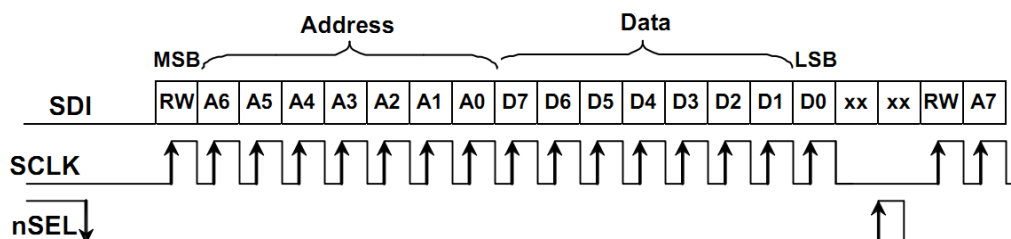
Glavna komponenta modula je integrirani krug Si4432 proizvođača Silicon labs koji koristi arhitekturu i protokol EZRadioPRO®. Integrirani krug nudi mogućnost podešavanja nosača frekvencije od 240 MHz do 940 Mhz, no linija prilagodbe za antenu izvedena je fiksno na modulu i pretpostavlja frekvenciju

nosača od točno 868 Mhz. Svaka devijacija rezultirat će gušenjem signala, smanjenjem dometa i zagrijavanjem modula zbog refleksija na liniji. Shema modula i način povezivanja s ugradbenim računalom prikazani su slikom (Slika 26).



**Slika 26** – Shema modula

Komunikacija s modulom izvedena je preko serijskog SPI protokola. SPI sabirnica podrazumijeva standardne signale: MISO(izlaz podataka iz modula), MOSI(ulaz podataka u modul), SCK(signal takta) i CS(*Chip Select*, omogućavanje prijenosa podataka). Vremenski dijagrami prikazani su na slici (Slika 27). Komunikacija s ugradbenim računalom ostvaruje se u dva koraka. U prvom koraku šalje se 1. bajt koji se sastoji od adrese (bitovi 0 - 6), a MSB (bit 7) je zahtjev za pisanje ili čitanje sa zadane adrese ('0' – čitanje, '1' – pisanje). Svaka adresa (od 0x00 do 0x7F) predstavlja jedan registar veličine 1 bajt. Popis registara i njihovo značenje prikazano je tablicom (Tablica 13).



**Slika 27** – Vremenski dijagrami SPI protokola

U drugom koraku šalje se 2. bajt koji je 8-bitni podatak ukoliko želimo pisati u registar ili proizvoljan podatak ukoliko se čita iz registra. Prije prvog koraka CS signal mora se postaviti u aktivno stanje (logička '0'), a nakon drugo koraka mora se deaktivirati. Moguć je poseban mod rada (Burst write/read) – ako se CS ne deaktivira u drugom koraku, trenutna adresa se povećava za 1, a nadolazeći podaci služe za pisanje ili čitanje u slijedeći registar. Ovaj je mod posebno pogodan kada se želi upisati skup podataka u neki blok registara ili za kontinuirani upis u FIFO modula.

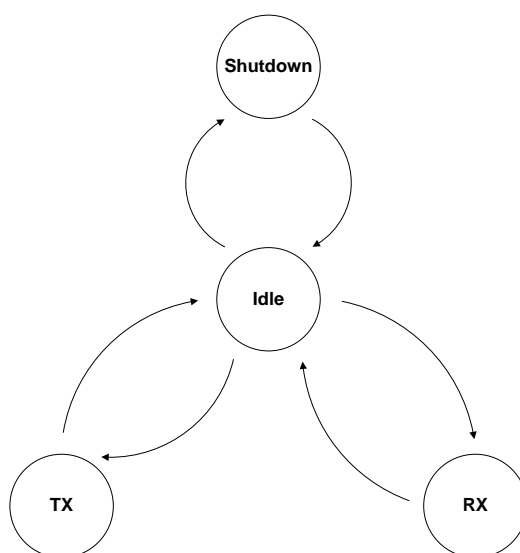
**Tablica 13** – Pregled registara modula

Adresa	R/W	Ime i opis registara	Vrijednost nakon reseta
0	R	Device Type	111
1	R	Device Version	06h
2	R	Device Status	—
3	R	Interrupt Status 1	—
4	R	Interrupt Status 2	—
5	R/W	Interrupt Enable 1	00h
6	R/W	Interrupt Enable 2	03h
7	R/W	Operating & Function Control 1	01h
8	R/W	Operating & Function Control 2	00h
9	R/W	Crystal Oscillator Load Capacitance	7Fh
0A	R/W	Microcontroller Output Clock	06h
0B	R/W	GPIO0 Configuration	00h
0C	R/W	GPIO1 Configuration	00h
0D	R/W	GPIO2 Configuration	00h
0E	R/W	I/O Port Configuration	00h
0F	R/W	ADC Configuration	00h
10	R/W	ADC Sensor Amplifier Offset	00h
11	R	ADC Value	—
12	R/W	Temperature Sensor Control	20h
13	R/W	Temperature Value Offset	00h
14	R/W	Wake-Up Timer Period 1	03h
15	R/W	Wake-Up Timer Period 2	00h
16	R/W	Wake-Up Timer Period 3	01h
17	R	Wake-Up Timer Value 1	—
18	R	Wake-Up Timer Value 2	—
19	R/W	Low-Duty Cycle Mode Duration	00h
1A	R/W	Low Battery Detector Threshold	14h
1B	R	Battery Voltage Level	—
1C	R/W	IF Filter Bandwidth	01h
1D	R/W	AFC Loop Gearshift Override	40h
1E	R/W	AFC Timing Control	0Ah
1F	R/W	Clock Recovery Gearshift Override	03h

20	R/W	Clock Recovery Oversampling Ratio	64h
21	R/W	Clock Recovery Offset 2	01h
22	R/W	Clock Recovery Offset 1	47h
23	R/W	Clock Recovery Offset 0	AEh
24	R/W	Clock Recovery Timing Loop Gain 1	02h
25	R/W	Clock Recovery Timing Loop Gain 0	8Fh
26	R	Received Signal Strength Indicator	—
27	R/W	RSSI Threshold for Clear Channel Indicator	1Eh
28	R	Antenna Diversity Register 1	—
29	R	Antenna Diversity Register 2	—
2A	R/W	AFC Limiter	00h
2B	R	AFC Correction Read	00h
2C	R/W	OOK Counter Value 1	18h
2D	R/W	OOK Counter Value 2	BCh
2E	R/W	Slicer Peak Hold	26h
2F		Reserved	
30	R/W	Data Access Control	8Dh
31	R	EzMAC status	—
32	R/W	Header Control 1	0Ch
33	R/W	Header Control 2	22h
34	R/W	Preamble Length	08h
35	R/W	Preamble Detection Control	2Ah
36	R/W	Sync Word 3	2Dh
37	R/W	Sync Word 2	D4h
38	R/W	Sync Word 1	00h
39	R/W	Sync Word 0	00h
3A	R/W	Transmit Header 3	00h
3B	R/W	Transmit Header 2	00h
3C	R/W	Transmit Header 1	00h
3D	R/W	Transmit Header 0	00h
3E	R/W	Transmit Packet Length	00h
3F	R/W	Check Header 3	00h
40	R/W	Check Header 2	00h
41	R/W	Check Header 1	00h
42	R/W	Check Header 0	00h
43	R/W	Header Enable 3	FFh
44	R/W	Header Enable 2	FFh
45	R/W	Header Enable 1	FFh
46	R/W	Header Enable 0	FFh
47	R	Received Header 3	—
48	R	Received Header 2	—
49	R	Received Header 1	—
4A	R	Received Header 0	—
4B	R	Received Packet Length	—
4C-4E		Reserved	
4F	R/W	ADC8 Control	10h
50-5F		Reserved	
60	R/W	Channel Filter Coefficient Address	00h
61		Reserved	
62	R/W	Crystal Oscillator/Control Test	24h

63-68		Reserved	
69	R/W	AGC Override 1	20h
6A-6C		Reserved	
6D	R/W	TX Power	18h
6E	R/W	TX Data Rate 1	0Ah
6F	R/W	TX Data Rate 0	3Dh
70	R/W	Modulation Mode Control 1	0Ch
71	R/W	Modulation Mode Control 2	00h
72	R/W	Frequency Deviation	20h
73	R/W	Frequency Offset 1	00h
74	R/W	Frequency Offset 2	00h
75	R/W	Frequency Band Select	75h
76	R/W	Nominal Carrier Frequency 1	BBh
77	R/W	Nominal Carrier Frequency 0	80h
78		Reserved	
79	R/W	Frequency Hopping Channel Select	00h
7A	R/W	Frequency Hopping Step Size	00h
7B		Reserved	
7C	R/W	TX FIFO Control 1	37h
7D	R/W	TX FIFO Control 2	04h
7E	R/W	RX FIFO Control	37h
7F	R/W	FIFO Access	—

Kako bi se smanjila efektivna potrošnja energije, ponašanje modula izvedeno je kao automat s konačnim brojem stanja. Mogući prijelazi iz određenog stanja u neko drugo prikazano je shemom na slici (Slika 28).



**Slika 28** – Stanja modula

*Shut down* stanje, posebno je stanje minimalne potrošnje energije u koje se dolazi sklopovski, postavljanjem odgovarajućeg izvoda modula u logičku '1'. U tom



stanju svi blokovi integriranog kruga su ugašeni. U ovom projektu, takvo se stanje ne koristi jer je upravo modul komponenta koja periodički budi cijelu jedinicu, dakle potreban nam je aktivan blok za mjerenje vremena (WUT) u modulu.

U normalnom načinu rada koriste se stanja visoke potrošnje, TX i RX za slanje odnosno primanje podataka, koji automatski prelaze u stanje *Idle*, male potrošnje, po završetku operacije. Prema potrebama korisnika, *Idle* stanje može se podesiti na različite razine aktivnosti modula. U tablici (Tablica 14) prikazane su značajke pojedinih stanja modula. Moguća *Idle* stanja označena su zvjezdicom (\*), a oznaka X znači nebitno.

**Tablica 14 – Značajke stanja modula**

Mode Name	Circuit Blocks								
	Digital LDO	SPI	32 kHz OSC	AUX	30 MHz XTAL	PLL	PA	RX	I <sub>VDD</sub>
SHUT DOWN	OFF (Register contents lost)	OFF	OFF	OFF	OFF	OFF	OFF	OFF	15 nA
STANDBY*	ON (Register contents retained)	ON	OFF	OFF	OFF	OFF	OFF	OFF	450 nA
SLEEP*		ON	ON	X	OFF	OFF	OFF	OFF	1 μA
SENSOR*		ON	X	ON	OFF	OFF	OFF	OFF	1 μA
READY*		ON	X	X	ON	OFF	OFF	OFF	800 μA
TUNING*		ON	X	X	ON	ON	OFF	OFF	8.5 mA
TRANSMIT		ON	X	X	ON	ON	ON	OFF	85 mA
RECEIVE		ON	X	X	ON	ON	OFF	ON	18.5 mA

Za uspješnu komunikaciju između dva modula potrebno je oba ispravno podesiti upisivanjem konfiguracijskih bajtova u određene registre. Da bi se olakšao postupak konfiguracije velikog broja registara, proizvođač Hoperf electronic ponudio je excel kalkulator koji na temelju željenih značajki izbacuje vrijednosti za upis u registre. Tablicom 15 prikazane su odabrane značajke koje su programirane u sva tri modula u sve tri jedinice.

**Tablica 15 – Odabir komunikacijskih značajki modula**

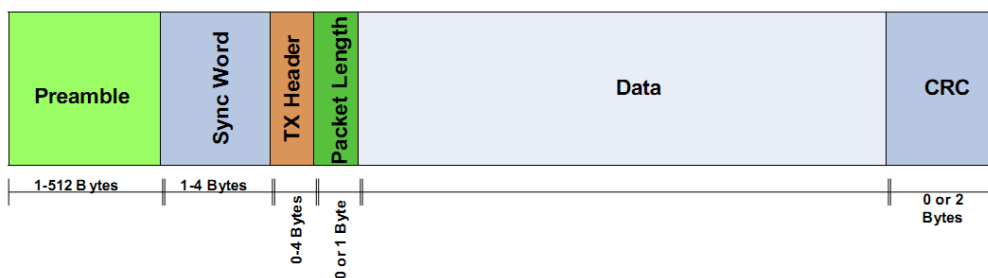
Parametar	Vrijednost
Vrsta modulacije	GFSK
Manchester kodiranje	NE
Nosioč frekvencije	868 Mhz
Devijacija frekvencije	50 kHz
Frekvencija kristalnog oscilatora	30 Mhz
Automatsko upravljanje frekvencijom (AFC)	DA
Brzina prijenosa podataka	100 kbps
Automatsko pakiranje podataka	DA
Mod rada	FIFO
CRC kontrola	DA
CRC tip	CRC16-IBM
Prvo šalji MSB/LSB	MSB
Uključiti zaglavlje u paketu	DA (3,2)
Duljina sinkronizacije	2 byta
Duljina preambule	8 nibbla
Uključiti duljinu podataka u paket	DA
Provjera zaglavlja pri prijemu	DA (3,2)

Korištena je GFSK modulacije zbog svog čisteg spektra u odnosu na FSK modulaciju.

Nositelj frekvencije podešen je u skladu s prilagodnom linijom modula.

Automatsko upravljanje frekvencijom (AFC – Automatic frequency control) je sklop koji omogućava automatsko detektiranje frekvencije nosioca kod prijema, a koji se uslijed propagacije kroz prostor može malo promijeniti.

Mod rada može biti kontinuirani kod kojeg predajni modul kontinuirano šalje analogni signal, a prijemni ga prima. Drugačiji mod rada jest tzv. FIFO mod. Ovdje se podatak upisuje u FIFO registar modula, a koji se potom automatski pakira u paket određen postavkama i šalje prema prijemnom modulu. Nakon odaslanog signala, modul se vraća u Idle stanje. Pakiranje paketa određeno je EZRadioPRO® standardom, a struktura paketa prikazana je na slici (Slika 29). Paket se koristi kao modulacijski signal frekvencijskog modulatora u modulu.



**Slika 29** – Struktura paketa

Osim arhitekture za bežičnu komunikaciju modul ima integrirane slijedeće pomoćne sklopove od kojih su masno otisnuti oni korišteni:

**Smart reset** – prilikom paljenja uređaja postavlja svaki registar na njegovu inicijalnu, tvornički predodređenu vrijednost.

**Microcontroller clock** – sklop za odabir frekvencije i njeno prosljeđivanje na jedan od pomoćnih izvoda (GPIO). Koristi se kao generator takta za ugradbeno računalo čime se uklanja potreba za dodatnim vanjskim rezonatorom.

**General purpose ADC** – 8-bitni analogno - digitalni pretvornik opće namjene sa integriranim referentnim naponom od 1.2 V. Moguće je povezivanje s vanjskim pomoćnim izvodima (GPIO) ili integriranim analognim temperaturnim senzorom.

**Temperature senzor** – Integrirani analogni temperaturni senzor pomoću kojeg dobivamo vrijednu informaciju o temperaturi okoline. Spaja se na interni ADC. Moguć je odabir nekoliko mjernih područja od kojih je odabrano ono od -64 °C do 64 °C s tolerancijom od 0.5 °C. Temperatura se očitava u mV, a stupnjevi se mogu izračunati prema formuli (Formula 24).

**Low battery detector** – zasebni analogno – digitalni pretvornik rezolucije 5-bita čiju vrijednost možemo pročitati u svakom trenutku u odgovarajućem registru. Pročitana vrijednost preračunava se u volte prema formuli (Formula 25). Ista odgovara naponu napajanja modula, a kako je modul napajan direktno iz baterije, možemo pratiti preostali kapacitet baterije. Zato je ovaj sklop od značajne važnosti za svaku jedinicu.

**Wake-up timer (WUT)** – jedini aktivni sklop u Idle stanju. WUT je vremenski sklop koji se koristi za namještanje perioda buđenja jedinice. Moguće je

podešavanje perioda u vrlo velikom rasponu prema formuli (Formula 26), gdje je M 16-bitni registar, a R 5-bitni registar. Sklop radi na taktu frekvencije 32.768 kHz. Pri isteku brojanja, postavlja se prekid, a sklop započinje brojanje od početka.

$$\mathbf{Temperatura = ADC8_{vrijednost} * 0.5 - 64 [^{\circ}C]} \quad (24)$$

$$\mathbf{NaponBaterije = ADC5_{vrijednost} * 50mV + 1.7 [V]} \quad (25)$$

$$\mathbf{WUT = \frac{4 * M * 2^R}{32.768} [ms]} \quad (26)$$

**GPIO configuration** – posebni dio arhitekture modula koji omogućava upravljanje trima izvodima opće namjene. Izvodi mogu biti izlazi ili ulazi ili se na njih prosljeđuje neki interni signal. U jedinicama se koriste za prebacivanje TX ili RX antene.

Antena diversity – sklop koji preko GPIO izvoda omogućava napredno upravljanje prijemnom i predajnom antenom posebnim algoritmima prebacivanja iz jedne u drugu u svrhu rješavanja problema koji se pojavljuje kod višestruke propagacije signala – kada na prijemnu antenu dolazi isti signal iz više smjerova, različitih intenziteta.

RSSI (Received signal strength indicator) – sklop za mjerenje jačine signala za vrijeme prijema. Njegovu vrijednost možemo očitati u svakom trenutku u odgovarajućem registru. Postoji i registar praga RSSI signala, kojim definiramo nivo okidanja prekida.

## 8.4 Elektromagnetni ventil za vodu

Cijela ideja ovog projekta jest u pravom trenutku uključiti ili isključiti navodnjavanje nekog sektora. Protok vode upravljan je elektromagnetnim ventilom preko izvršne jedinice. Korišteni elektromagnetni ventil je SRV-101G proizvođača Hunter (Slika 30).

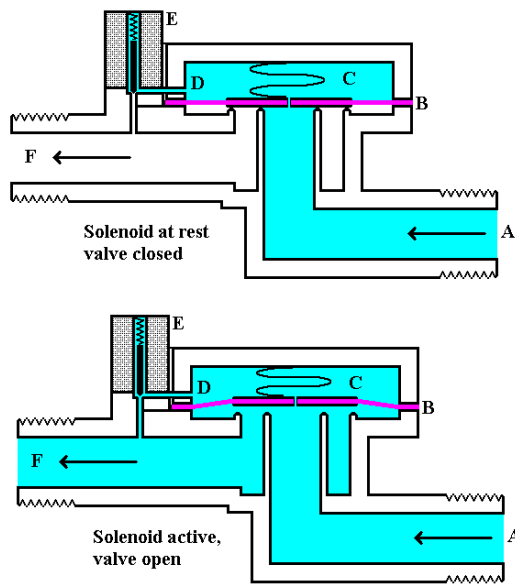


**Slika 30** – Elektromagnetni ventil SRV-101G

Elektromagnetni ventil sastoji se od dva glavna dijela: Ventil s membranom i solenoid.

### 8.4.1 Ventil s membranom

Ventil s membranom omogućava protok vode pod velikim pritiskom uz malu potrebnu silu za otvaranje i zatvaranje ventila. Fina regulacija protoka može se fiksno podesiti zakretanjem osovine na vrhu kojom se regulira pritisak na membranu. Princip rada ventila s membranom prikazan je na slici (Slika 31).



**Slika 31** – Ventil s membranom (princip rada)

Kada je ventil zatvoren, iglica solenoida je spuštena tako da zatvara kanal D. Time su tlakovi u komorama A i C izjednačeni pa sila koja djeluje na membranu B jest sila opruge koja potpuno zatvara ventil.

U trenutku kada se podigne iglica solenoida, komora C se počinje prazniti kroz sada slobodan kanal D. Time je tlak komore C manji od tlaka komore A pa dolazi do djelovanja sile na membranu, suprotne od djelovanja sile opruge. Membrana se lagano podiže propuštajući protok vode iz komore A u komoru F, gdje su iste ulaz i izlaz iz ventila.

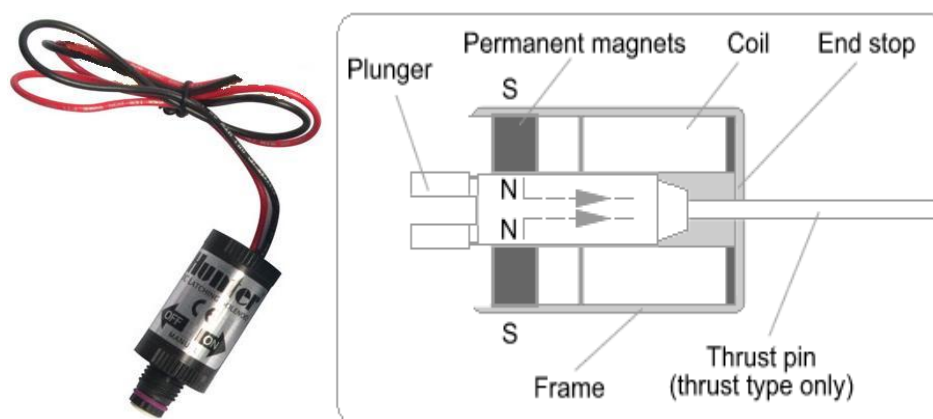
Ventil ima definirani smjer protoka, pa se prema tome mora poštivati orijentacija ventila prilikom ugradnje u sustav navodnjavanja.

Značajke modela SRV-101G prema tehničkoj dokumentaciji proizvođača:

- Maksimalna temperatura rada: 52 °C
- Opseg protoka: od 3.8 L/min do 114 L/min
- Preporučeni opseg tlaka: od 1.4 bar do 10.3 bar

### 8.4.2 Solenoid

Solenoid je bistabilni elektromehanički uređaj. Sastoji se od zavojnice i osovine koji čine elektromagnet, permanentnog magneta i opruge (Slika 32). Permanentni magnet omogućava osovini da zaglavi u jednom krajnjem položaju, dok opruga drži osovinu o drugom krajnjem položaju, nakon otpuštanja permanentnog magneta. Promjena položaja osovine iz jednog položaja u drugi (spuštena ili podignuta iglica solenoida) vrši se pobuđivanjem elektromagneta.



**Slika 32** – Solenoid i princip rada

Elektromagnet se pobuđuje istosmjernom strujom pri naponu od 9 V. Propuštanjem struje u jednom smjeru zaglavljuje osovinu na stranu s permanentnim magnetom, a propuštanje struje u drugom smjeru, zajedno s oprugom stvara silu dovoljnu za otpuštanje osovine s permanentnog magneta i pomicanje u drugo stabilno stanje.

Izmjerena struja prilikom spajanja solenoida na izvor napajanja od 9 V doseže čak 1.6 A što čini preveliku potrošnju u sustavu s baterijskim napajanjem. Da bi se pomicala osovina solenoida, nije potrebna puna snaga  $P_{max}$  (Formula 27). Eksperimentalno su određene minimalne snage paljenja  $P_p$  i gašenja  $P_g$  (Formule 28 i 29). Pri tome paljenje znači podizanje osovine prema permanentnom magnetu, a gašenje podrazumijeva otpuštanje osovine s permanentnog magneta.

$$P_{max} = U * I = 9V * 1.6A = 14.4 W \quad (27)$$

$$P_p = P_{max}(60\%) = P_{max} * 0.6 = 8.64 W \quad (28)$$

$$P_g = P_{max}(30\%) = P_{max} * 0.3 = 4.32 W \quad (29)$$

Iz formula (Formula 28 i 29) možemo izračunati struju paljenja i struju gašenja (Formule 30 i 31):

$$I_p = \frac{P_p}{U} = \frac{8.64 W}{9 V} = 0.96 A \quad (30)$$

$$I_g = \frac{P_g}{U} = \frac{4.32 W}{9 V} = 0.48 A \quad (31)$$

Time smo minimizirali ukupnu potrošnju energije, a valja istaknuti da impuls paljenja i gašenja solenoida iznosi svega 50 ms.

Upravljanje solenoidom ne može se vršiti direktnim spajanjem na ugradbeno računalo. Zato se kao upravljački sklop solenoida koristi H-most BD6222, opisan u slijedećem poglavlju. Propuštanje samo dijela snage u solenoid ostvaruje programski, upotrebom pulsno širinskog modulatora – PWM.



## 8.5 H-most: BD6222

H-most najčešće se koristi za upravljanje motorima, a gdje su potrebne veće snage od one koju mogu dati ugradbena računala. Uz pojačanje snage, što im je primarna funkcija, omogućuju i promjenu smjera vrtnje motora. To se postiže topologijom tranzistora koja slični na slovo H, a od tuda i naziv samog sklopa.

U našem slučaju H-most koristi će se za upravljanje solenoidom, što je elektronički gledano jednako upravljanju motorima. Za upravljanje solenoidom, potrebno je pojačanje napona i struje te mogućnost promjene polariteta, a to je sve ono što H-most može dati.

Korišten je H-most BD6222 proizvođača ROHM Semiconductor. Postoji cijela paleta izvedbi istog proizvođača, verzija 6222 odabrana je zbog najboljeg omjera performansi i cijene gledano za potrebe konkretne izvršne jedinice ovog sustava.

Najvažnije značajke koje je izdvojio proizvođač prikazane su izvatkom na slici (Slika 33).

### ● Features

- 1) Built-in, selectable one channel or two channels configuration
- 2) Low standby current
- 3) Supports PWM control signal input (20kHz to 100kHz)
- 4) VREF voltage setting pin enables PWM duty control
- 5) Cross-conduction prevention circuit
- 6) Four protection circuits provided: OCP, OVP, TSD and UVLO

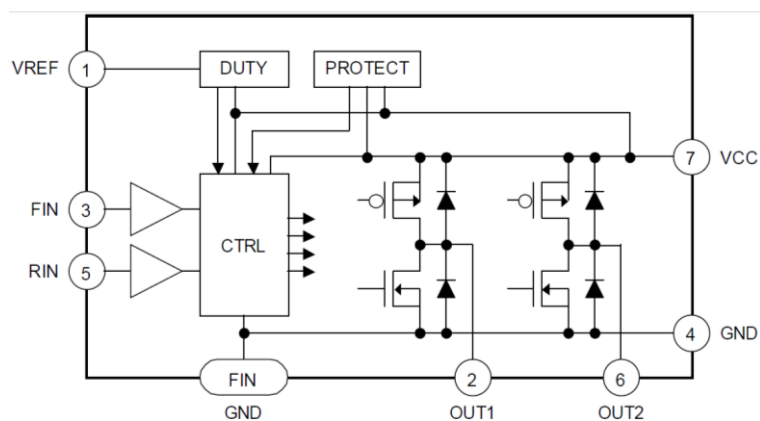
### **Slika 33** – Značajke H-mosta BD6222

U tablici (Tablica 16) prikazane su najvažnije električne karakteristike ovog integrirano kruga, značajne sa strane kompatibilnosti sa sustavom u koji se ugrađuje.

**Tablica 16 – Kompatibilnost BD6222 sa sustavom**

Parametar	Vrijednost	Zahtjev sustava
Maksimalni napon napajanja	18 V	9 V
Maksimalna izlazna struja	2 A	1 A
Temperaturno područje rada	Od -40 °C do 85 °C	Od -20 °C do 40 °C
Struja mirovanja	10 $\mu$ A	odgovara
Minimalni napon logičke '1'	2 V	odgovara
Maksimalni napon logičke '0'	0.8 V	odgovara
Nosioč frekvencije PWM - a	Od 20 do 35 kHz	30 kHz

Shematski prikaz arhitekture integriranog kruga prikazan je na slici (Slika 34), a opis izvoda zapisan je u tablici (Tablica 17).



**Slika 34 – Shema arhitekture BD6222**

**Tablica 17 – Značajke izvoda**

Izvod	Ime	Značajka
1	Vref	Referentni napon, određuje radni ciklus ( <i>Duty-Cycle</i> )
2	Out1	Izlaz 1
3	Fin	Upravljački ulazni izvod (naprijed)
4	GND	Masa
5	Rin	Upravljački ulazni izvod (nazad)
6	Out2	Izlaz 2
7	Vcc	Napon napajanja

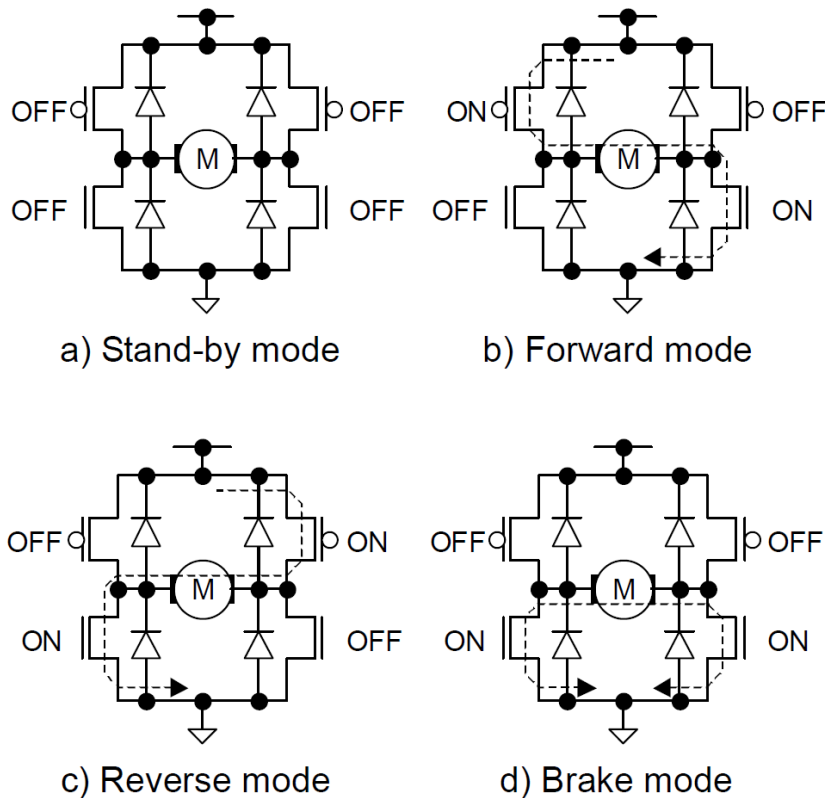
Ovisno o TTL/CMOS ulazima Fin i Rin, ugradbenim računalom upravljamo izlazima snage Out1 i Out2. Tablicom (Tablica 18) prikazana su sva moguća stanja i operacije nad H-mostom BD6222.

**Tablica 18** – Logička stanja BD6222

	FIN	RIN	VREF	OUT1	OUT2	Operation
a	L	L	X	Hi-Z*	Hi-Z*	Stand-by (idling)
b	H	L	VCC	H	L	Forward (OUT1 > OUT2)
c	L	H	VCC	L	H	Reverse (OUT1 < OUT2)
d	H	H	X	L	L	Brake (stop)
e	PWM	L	VCC	H	$\overline{\text{PWM}}$	Forward (PWM control mode A)
f	L	PWM	VCC	$\overline{\text{PWM}}$	H	Reverse (PWM control mode A)
g	H	PWM	VCC	$\overline{\text{PWM}}$	L	Forward (PWM control mode B)
h	PWM	H	VCC	L	$\overline{\text{PWM}}$	Reverse (PWM control mode B)
i	H	L	Option	H	$\overline{\text{PWM}}$	Forward (VREF control)
j	L	H	Option	$\overline{\text{PWM}}$	H	Reverse (VREF control)

Hi-Z u tablici označava stanje visoke impedancije, no ipak između otvorenih tranzistora nalazi se zaporno orijentirana dioda koja ima svoj konačan otpor i struje curenja. X u tablici znači da iznos napona ne utječe na operaciju.

Malim slovima u tablici označene su diskretne kombinacije. Skup od a,b,c i d predstavlja kombinacije osnovnog režima rada H-mosta. Tako postoje i četiri osnovne topologije tj. tokovi struja kroz tranzistore (Slika 35).



**Slika 35** – Osnovne topologije

Topologija a) aktivna je kada su Fin i Rin u logičkoj '0'. U ovom stanju struje ne teku kroz tranzistore. Koristi se kao stanje mirovanja i uštede energije.

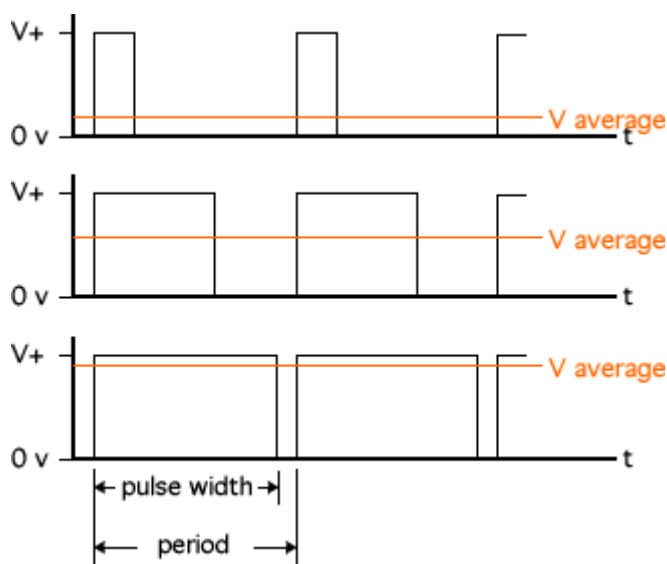
Topologija b) zatvara dva dijagonalna tranzistora, dok su suprotna dva otvorena. Rezultat je tok struje iz izvora, pa kroz zavojnicu motora ili u našem slučaju solenoida pa prema odvodu.

Topologija c) simetrična je topologiji b), a rezultat je tok struje u suprotnom smjeru kroz trošilo u odnosu na tok struje u b) topologiji.

Topologija d) ne koristi se za upravljanje solenoidom.

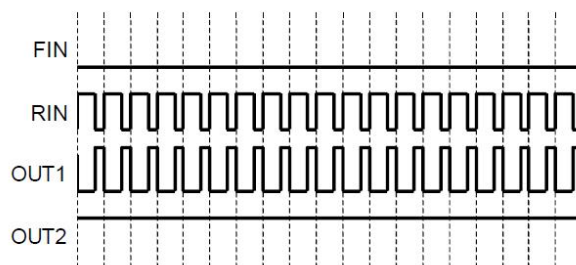
Izmjenjujući topologije b) i c) mijenjamo polaritet napajanja solenoida što rezultira otvaranjem i zatvaranjem ventila za vodu. Ipak, navedene topologije puštaju punu snagu u zavojnicu solenoida pa dolazi do prevelike potrošnje energije. Stoga se topologije b) i c) koriste u PWM načinu rada.

Pulsno – širinska modulacija omogućava kontroliranje toka efektivne energije kroz trošilo tako da se struja malo propušta, a malo ne tokom jednog perioda. Taj osnovni period nazivamo frekvencijski nosač PWM signala. Efektivna energija tada ovisi o odnosu trajanja propuštanja struje prema trajanju nepropuštanja struje kroz trošilo.

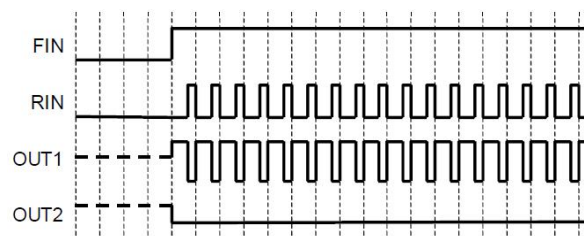


**Slika 36** – Pulsno–širinska modulacija

Sa slike (Slika 36) možemo primijetiti kako je efektivni napon sve manji kada je vrijeme pozitivnog pulsa (propuštanja struje kroz trošilo) sve kraće i obrnuto. Upravo taj princip primjenjuje se za upravljanje solenoidom. Konkretno, ugradbenim računalom programiramo ulaze Fin i Rin na način da ulazom Fin određujemo polaritet solenoida (ovisno o zahtjevu za otvaranje ili zatvaranje), a kroz ulaz Rin puštamo signal iz internog PWM generatora ugradbenog računala. Takav način rada odgovara operacijama f) i g) iz tablice (Tablica 18). Pri tome treba voditi računa o invertiranoj logici u f) načinu rada gdje pozitivni puls znači blokiranje struje. Izgledi vremenskih dijagrama za f) način rada prikazan je slikom (Slika 37), g) način rada slikom (Slika 38).



**Slika 37** – PWM f) način rada



**Slika 38** – PWM g) način rada

Za ispravan i točan rad pri PWM načinima rada, potrebno je poštivati donju i gornju graničnu frekvenciju na kojoj BD6222 može raditi. Zbog tranzitnih integriranog kruga prilikom ulaska i izlaska u neaktivno stanje rada, definirana je donja granična frekvencija od 20 kHz pri kojoj je garantirani uspjeh rada PWM upravljanja jer neće doći do tranzitnih pojava. S druge strane zbog tromosti tranzistora, maksimalan moguća frekvencija iznosi 100 kHz.

Ugradbenim računalom podešena je frekvencija nosioca PWM signala na 33 kHz.

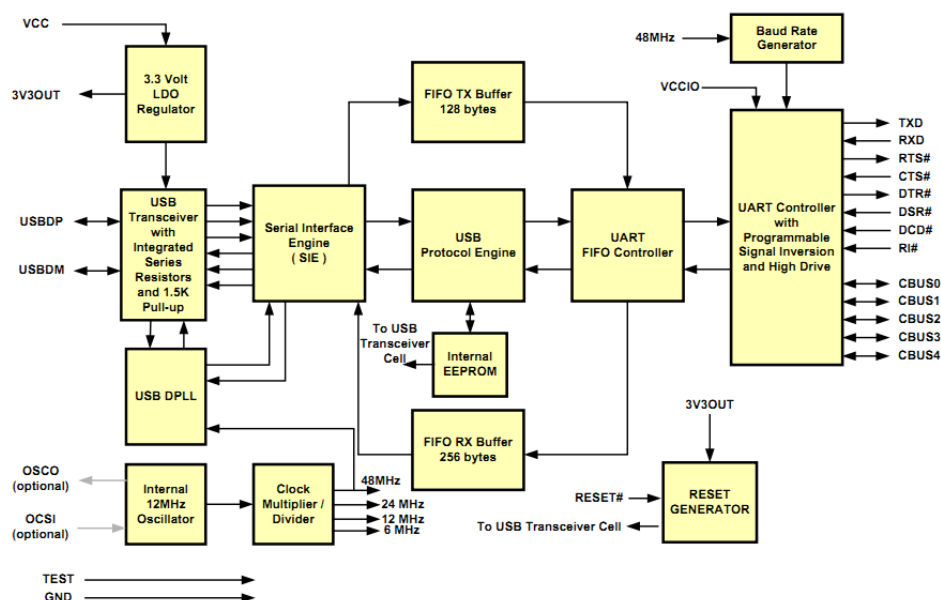
## 8.6 USB/SPI pretvornik: FT232RL

Future technology devices international limited (FTDI) poznati su po razvoju integriranih krugova za ostvarenje različitih sučelja. Najčešće se radi o pretvorbi USB protokola u neki drugi. U centralnoj jedinici potrebno je ostvariti komunikaciju između računala i primopredajnog modula. Pri tome, računalo koristi USB protokol, a RF modul SPI protokol. Idealno bi bilo koristiti FTDI FT2232 integrirani krug koji je projektiran za direktnu pretvorbu USB – SPI. Taj odabir, ipak je odbačen zbog njegove lošije dostupnosti, cijene i dodatnih mogućnosti u odnosu na standardni i široko poznati FTDI FT232RL.

U provjerenoj praksi FT232RL koristi se i vrlo dobro obavlja pretvorbu protokola USB – UART. Svaki FT232RL dolazi s dvije vrste upravljačkih programa (*drivers*): VCP (Virtual COM port drivers) i D2XX (Direct drivers). Uobičajeno se koristi VCP upravljački programa koji simulira standardni RS232 protokol. Sa direktnim D2XX upravljačkim programom moguće je dobiti pristup naprednim opcijama i načinima rada FT232RL integriranog kruga. Jedna takva je i *Bit-Bang* način rada u kojem možemo direktno upravljati naponskim stanjima pojedinih izvoda. Takva mogućnost pruža široku primjenu, a ovdje je iskorištena za emulaciju SPI protokola.

## 8.6.1 Opis rada FT232RL

Na slici (Slika 39) prikazana blok shema integriranog kruga FT232RL.



**Slika 39** – Blok shema FT232RL

Radi se o vrlo sofisticiranom sklopu koji je projektiran za jednostavno korištenje. Tako korisnik uopće ne mora poznavati njegovu arhitekturu već je dovoljno znati ulogu pojedinih izvoda čiji su nazivi također standardizirana i intuitivni (Tablica 19).

**Tablica 19** – Uloga izvoda FT232RL

Naziv izvoda	Uloga
TXD, RXD, RTS#, CTS#, DTR#, DSR#, DCD#, RI#	Standardni izvodi RS232 protokola.
CBUS0 – CBUS4	Konfigurabilni izvodi posebne namjene.
Vcc	Napon napajanja (iz USB-a).
GND, AGND	Potencijal mase.
3V3OUT	Izlaz za napajanje jedinica koje rade na naponu napajanja od 3.3 V.
VCCIO	Priključak za napajanje izlaznih izvoda; Koristi se za namještanje naponskih razina izlaza.
USBDP, USBDM	USB priključci (USB+ i USB-).
OSCI, OSCO	Priključci za spajanje vanjskog oscilatora (opcionalno)
RESET#	Koristi se za vraćanje tvorničkih postavki.
TEST	Postavlja sklopove u testni način rada.
NC	Ovi izvodi nisu korišteni ( <i>Not Connected</i> )

Integrirani krug FT232RL napaja se direktno iz USB sabirnice sa naponom napajanja od 5 V. Diferencijalna USB sabirnica (USB D+ i D-) spajaju se direktno na FT232RL preko odgovarajućih priključaka USBDP (D+) i USBDM (D-) bez ikakve potrebe za vanjskim komponentama. Izlaz 3V3OUT koristi se za napajanje RF modula koji ne može raditi na naponima višim od 3.3 V, pa nisu potrebni dodatni regulatori napona za pogon istog. Za ispravan rad, izvod TEST obavezno mora biti pritegnut prema masi.

Preko upravljačkog programa D2XX, FT232RL postavlja se u asinkroni Bit – Bang način rada u kojem je moguće u svakom trenutku promijeniti ili pročitati stanje sa standardnih izvoda za RS232 protokol (vidi Tablicu xx (gore)). Nije moguće pročitati ili upisati jedan bit za željeni izvod, već se koristi manipulacija odgovarajućeg bajta. Dohvaćanje ili upis istoga osigurano je funkcijama upravljačkog programa FT\_read odnosno FT\_write, uz uvjet da je FT232RL u asinkronom *Bit – Bang* načinu rada. Pojedini bit (D<sub>0</sub> – D<sub>7</sub>) navedenog bajta stanje je izvoda određenog tablicom (Tablica 20).

**Tablica 20** – Interpretacije izvoda

Naziv izvoda	Sklopovska interpretacija	Programska interpretacija
TXD	D <sub>0</sub>	-
RXD	D <sub>1</sub>	-
RTS#	D <sub>2</sub>	CLK
CTS#	D <sub>3</sub>	CS
DTR#	D <sub>4</sub>	MISO
DSR#	D <sub>5</sub>	MOSI
DCD#	D <sub>6</sub>	-
RI#	D <sub>7</sub>	NIRQ

Emulacija SPI protokola izvedena je programski i radi na način da se slijedno šalju paketi poštujući dogovorenu programsku interpretaciju signala iz tablice (Tablica 20 ). Jedan bajt predstavlja vektor vrijednosti stanja izvoda, a jedan paket sastoji se od 20 bajtova. Svaki bajt paketa predstavlja stanje SPI sabirnice u svakom trenutku promjene takta CLK.

Za potrebe programske emulacije SPI protokola stvorene su slijedeće funkcije koje obavljaju gore spomenute operacije (Tablica 21).

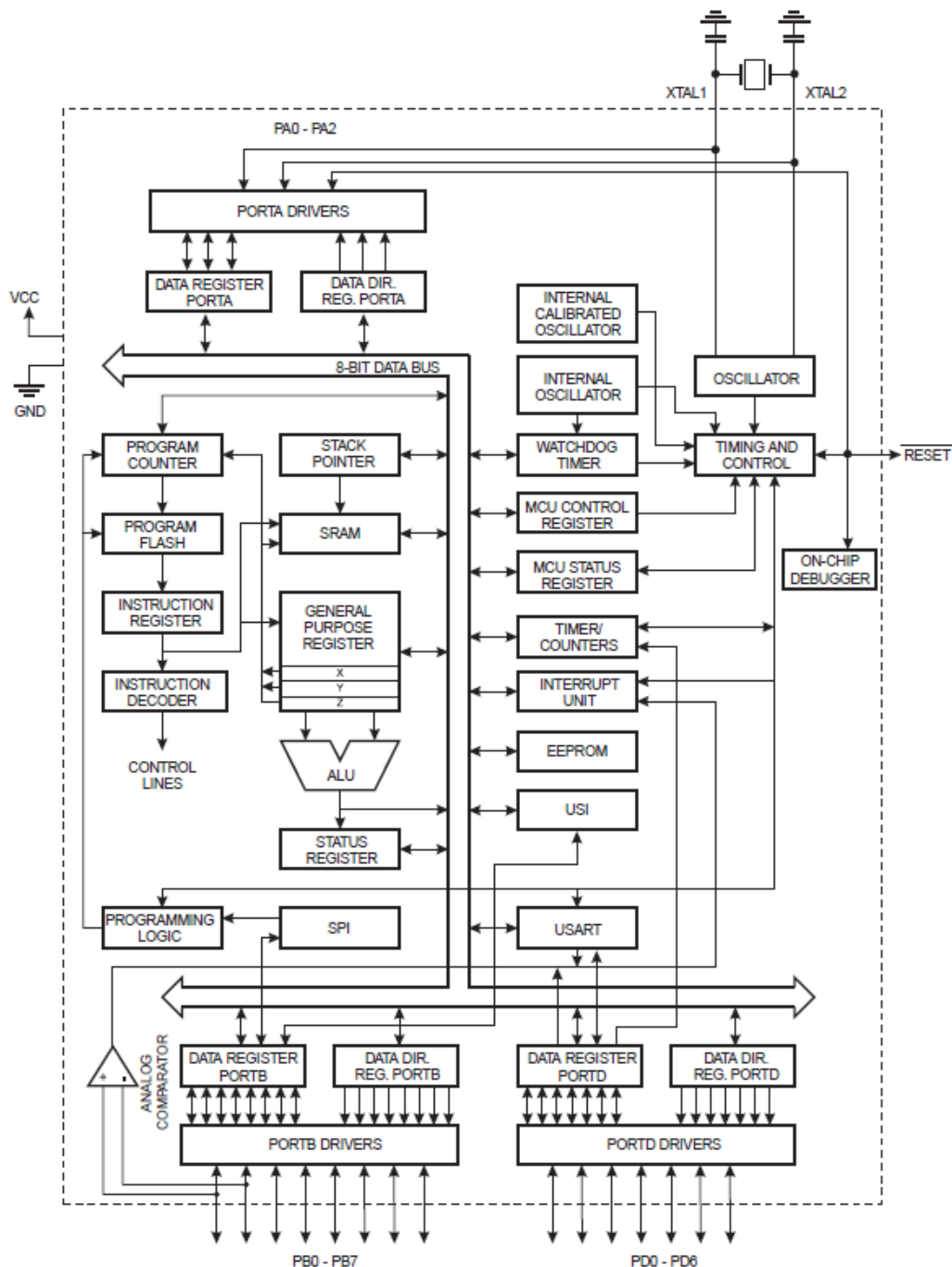


**Tablica 21 – Opis funkcija za emulaciju SPI**

Funkcija	Ulazni parametri/ Izlazni parametri	Uloga
InitializeFTDI	-/-	Inicijalizacija FT232RL integriranog kruga i postavljanje u Bit – Bang način rada.
makeSPIPacket	Podatak, veličina podatka u bitovima/ Paket, veličina paketa u bajtovima	Pretvara podatak u paket
writeSPIPacket	Paket, veličina paketa u bajtovima/ Informacija o uspjehnosti	Emulacija pisanja podataka preko SPI sabirnice
readSPIPacket	-/ Paket, veličina paketa u bajtovima, Informacija o uspjehnosti	Emulacija čitanja podataka preko SPI sabirnice
parseSPIPacket	Paket, veličina paketa u bajtovima/ Podatak	Pretvara paket u podatak

## 8.7 Ugradbeno računalo: ATtiny2313A

Ugradbeno računalo ATtiny2313A je *low-power* CMOS 8 – bitni mikrokontroler baziran na naprednoj RISC arhitekturi (Slika 40). Izvodi instrukcije pri svakom pojedinom taktu. Postiže 1 MIPS pri 1 MHz čime omogućuje korisniku da optimira odnos potrošnje i brzine rada.



**Slika 40** – Arhitektura ATtiny2313A

Nalazi se u senzorskoj i izvršnoj jedinici gdje se većinu vremena nalazi u neaktivnom (*sleep*) stanju sve do buđenje na zahtjev RF modula, kada obavlja obradu podataka prije ponovnog vrćanja u neaktivno stanje.

Sažetak osnovnih značajki prikazan je u tablici (Tablica 22), a u nastavku se nalaze opisi perifernih sklopova, ugradbenog računala, koji se koriste u senzorskoj ili izvršnoj jedinici.

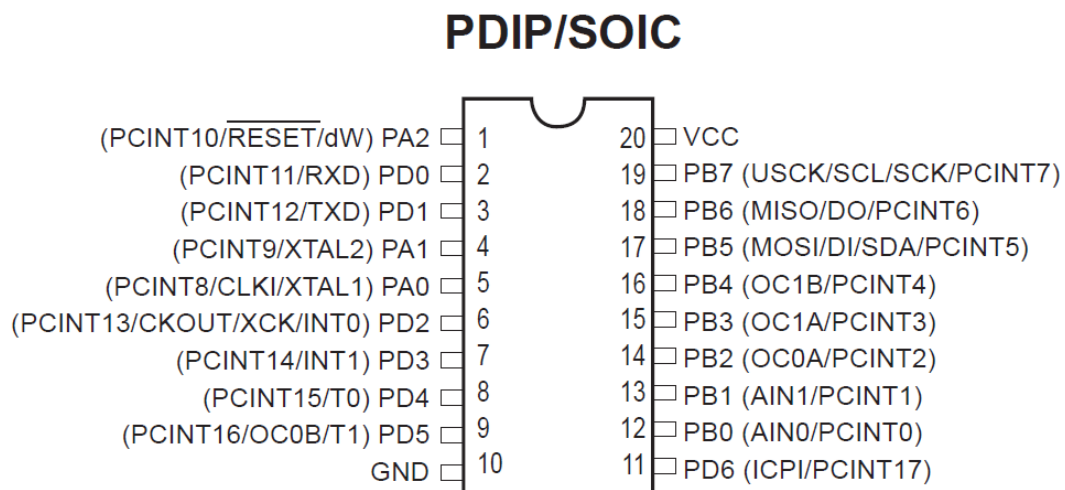
**Tablica 22 – Osnovne značajke Attiny2313A**

<b>High performance, Low Power AVR 8 – Bit Microcontroller</b>
<b>Napredna RISC Arhitektura</b>
- 120 Naprednih instrukcija – Instrukcije se izvršavaju pri svakom pojedinom taktu
- 32 x 8 Registara opće namjene
- Ostvaruje 20 MIPS pri taktu 20 MHz
<b>Podatkovna i <i>non-volatile</i> programska memorija</b>
- 2K Bytes In-System Self programmable flash
- Izdržljivost 10,000 Ciklusa Brisanja/Čitanja
- 128 Bytes In-System Programmable EEPROM
- Izdržljivost: 100,000 Ciklusa Brisanja/Čitanja
- 128 Bytes Internal SRAM
<b>Značajke periferije</b>
- Jedan 8-bit Timer/Counter s odvojenim djelilom i <i>compare</i> modom
- Jedan 16-bit Time/Counter s odvojenim djelilom, <i>compare</i> modom i <i>capture</i> modom
- Četiri PWM kanala
- USI - Univerzalno Serijsko Sučelje
- Full Duplex USART
<b>Posebne značajke mikrokontrolera</b>
- <i>debugWire On - chip Debugging</i>
- <i>In - System Programmable via SPI Port</i>
- <i>Low - power Idle, Power-down, and Standby Modes</i>
- Enhanced Power-on Reset Circuit
<b>I/O and Packages</b>
- 20-pin PDIP, 20-pin SOIC, 20-pad MLF/VQFN
<b>Napon napajanja</b>
- 1.8 - 5.5V
<b>Brzine rada</b>
- 0 - 4 MHz @ 1.8 - 5.5V
- 0 - 10 MHz @ 2.7 - 5.5V
<b>Temperatura rada : -40 C do +85 C</b>
<b>Low Power Consumption</b>

- <i>Active Mode</i>
- 190 uA pri 1.8 i 1MHz
- <i>Idle Mode</i>
- 24 uA pri 1.8V i 1MHz
- <i>Power-down Mode</i>
- 0.1 uA pri 1.8 i +25 C

### 8.7.1 Prekidi

Prekidi su signali koje CPU mikrokontrolera kontinuirano provjerava. Mogu se aktivirati pod utjecajem unutrašnje periferije, ili promjenom vanjskih izvoda ugrađenog računala. Popis svih mogućih prekida prikazan je tablicom (Tablica 23). Njihova aktivacija može utjecati na slijed izvođenja glavnog programa. Svaki prekid ima pripadajući vektor na koji će se preusmjeriti programsko brojilo u slučaju dozvoljene aktivacije prekida. ATtiny2313A podržava nekoliko prekidnih izvora od kojih je od posebne važnosti (za senzorsku i izvršnu jedinicu) vanjski izvor. Vanjski prekid se može postići koristeći dvije vrste izvoda: INTn i PCINTn, gdje je oznaka n broj izvoda prema slici (Slika 41).



**Slika 41** – Pregled izvoda ATtiny2313A

**Tablica 23 – Pregled prekidnih vektora**

Vector No.	Program Address	Label	Interrupt Source
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x0001	INT0	External Interrupt Request 0
3	0x0002	INT1	External Interrupt Request 1
4	0x0003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	0x0004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	0x0005	TIMER1 OVF	Timer/Counter1 Overflow
7	0x0006	TIMER0 OVF	Timer/Counter0 Overflow
8	0x0007	USART0, RX	USART0, Rx Complete
9	0x0008	USART0, UDRE	USART0 Data Register Empty
10	0x0009	USART0, TX	USART0, Tx Complete
11	0x000A	ANALOG COMP	Analog Comparator
12	0x000B	PCINT0	Pin Change Interrupt Request 0
13	0x000C	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x000D	TIMER0 COMPA	Timer/Counter0 Compare Match A
15	0x000E	TIMER0 COMPB	Timer/Counter0 Compare Match B
16	0x000F	USI START	USI Start Condition
17	0x0010	USI OVERFLOW	USI Overflow
18	0x0011	EE READY	EEPROM Ready
19	0x0012	WDT OVERFLOW	Watchdog Timer Overflow
20	0x0013	PCINT1	Pin Change Interrupt Request 1
21	0x0014	PCINT2	Pin Change Interrupt Request 2

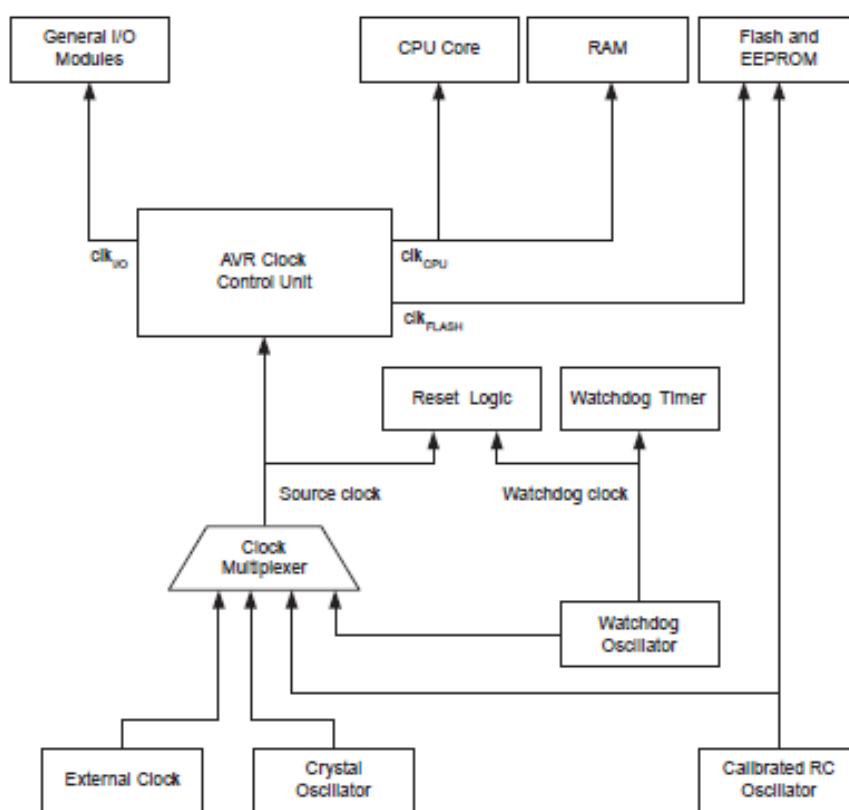
Bilo koja promjena nivoa signala na bilo kojem od PCINT0:16 će uzrokovati aktivaciju prekida. Programski se mora odrediti izvod koji je prouzrokovao prekid te je li prekid aktiviran rastućim ili padajućim bridom signala. Vanjski prekidi mogu se pratiti sinkrono ili asinkrono. Sinkronim praćenjem aktivacija prekida događa se za vrijeme promjene brida I/O takta (Sinkroni prekidi), dok asinkrono praćenje aktivira prekid odmah po promjeni stanja izvoda (Asinkroni prekidi). Potonji se mogu koristiti za buđenje uređaja iz svih *sleep* stanja, a Sinkroni prekidi ne mogu se aktivirati u stanjima gdje je ugašen I/O takt.

Vrijeme izvršavanja prekidne rutine za sve prekide iznosi četiri do pet CPU taktova minimalno. Vrijeme od četiri ili pet taktova signala ovisi o širini programskog brojila. Ako širina programskog brojila nije veća od dva bajta, onda će vrijeme izvršavanja prekidne rutine biti četiri takta minimalno. U slučaju kada je programsko brojilo veće od dva bajta, izvršavanje prekidne rutine trajati će minimalno pet CPU taktova.

Prioritet prekida određen je adresom prekidnog vektora. Prekid s najmanjom adresom prekidnog vektora ima najveći prioritet. Prema tome RESET ima najveći prioritet ispod njega INTO, INT1 itd. Ako se dva prekida dogode u isto vrijeme, izvršiti će se prekid s većim prioritetom.

### 8.7.2 Takt sustava

Slika (Slika 42) prikazuje izvore takta signala i njegovu distribuciju unutar sustava. Nisu potrebne sve distribucijske veze u svakom trenutku. Ako želimo smanjiti potrošnju, takt za periferije koje nam ne trebaju se može ugasiti stavljanjem sustava u različita *sleep* stanja.



**Slika 42** – Distribucija takta

Ugradbeno računalo s AVR arhitekturom pruža različita *sleep* stanja omogućujući korisniku da optimizira potrošnju prema zahtjevima zadatka.

### 8.7.3 Sleep stanja

Tablicom (Tablica 24) prikazana su različita *sleep* stanja i načini preko kojih sustav ponovno uspostavlja aktivno stanje (buđenje ugradbenog računala).

**Tablica 24** – Pregled i značajke *sleep* stanja

Sleep Mode	Active Clock Domains			Oscillators	Wake-up Sources				
	clk <sub>CPU</sub>	clk <sub>FLASH</sub>	clk <sub>I/O</sub>	Main Clock Source Enabled	INT0, INT1 and Pin Change	USI Start Condition	SPM/EEPROM Ready Interrupt	Other I/O	Watchdog Interrupt
Idle			X	X	X	X	X	X	X
Power-down					X <sup>(1)</sup>	X			X
Stand-by				X	X <sup>(1)</sup>	X			X

Ako se dogodi prekid dok je ugradbeno računalo u nekom od *sleep* stanja, inicirat će se buđenje MCU jedinice. Prvo će MCU jedinica biti zaustavljena četiri CPU takta (*start-up* vrijeme), a zatim će se izvoditi prekidna rutina. Po završetku prekidne rutine nastavlja se izvršavati program sa mjesta zadnjeg poziva SLEEP naredbe. Sadržaj registara i SRAM će biti očuvan kada se sustav probudi.

### 8.7.4 Power Reduction Register (PRR)

Power reduction register omogućuje dinamičko smanjenje potrošnje energije u sustavu tako da isključuje pojedine periferije. Kad je takt periferije zaustavljen vrijedi:

- Trenutno stanje periferije je zamrznuto
- U registre zamrznute periferije ne može se čitati ili upisivati
- Resursi zauzeti periferijom će i dalje ostati zauzeti

Periferija se ne smije koristiti ako joj zaustavimo takt. Brišući PRR bit budimo periferiju i stavljamo je u isto stanje prije gašenja. Gašenje periferije se može koristiti u *Idle* i *Active* modu prilikom čega ostvarujemo velike uštede u potrošnji. U svim ostalim modovima rada takt je već zaustavljen.

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	-	-	-	-	PRTIM1	PRTIM0	PRUSI	PRUSART	PRR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Slika 43** – PRR registar

- **Bit 3** – PRTIM1: Power Reduction Timer/Counter1  
Upisivanjem logičke jedinice u ovaj bit gasi Timer/Counter1 modul. Kad je Timer/Counter1 uključen, nastaviti će se tamo gdje je stao prije gašenja.
- **Bit 2** – PRTIM0: Power Reduction Timer/Counter0  
Upisivanjem logičke jedinice gasi Timer/Counter0 modul. Kad je Timer/Counter0 uključen, nastaviti će se tamo gdje je stao prije gašenja.
- **Bit 1** – PRUSI: Power Reduction USI  
Upisivanjem logičke jedinice gasi USI gašenjem takta modula. Kad je USI opet upaljen, USI mora opet biti inicijaliziran prije ponovnog korištenja.
- **Bit 0** – PRUSART: Power Reduction USART  
Upisivanje logičke jedinice gasi USI zaustavljajući takt modulu. Kad je UART opet uključen, USART mora opet biti inicijaliziran prije ponovnog korištenja

### 8.7.5 8-bit Timer/Counter sa PWM generatorom

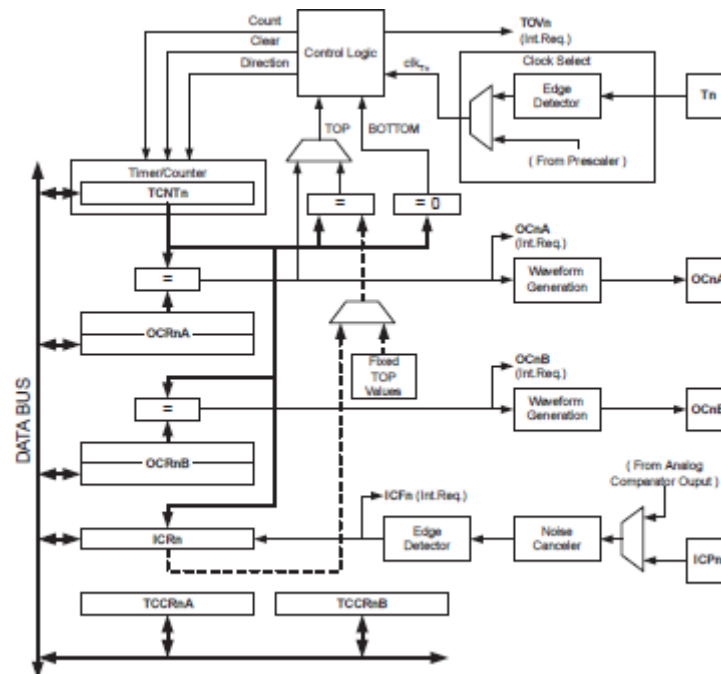
Timer/Counter0 je općenito 8-bitni Timer/Counter sklop, s dvije neovisne *output compare* jedinice koje podržavaju PWM način rada. PWM omogućava





### 8.7.6 16-bit Timer/Counter1

Pojednostavljena blok shema 16-bitnog Timer/Counter sklopa prikazana je slikom (Slika 46).

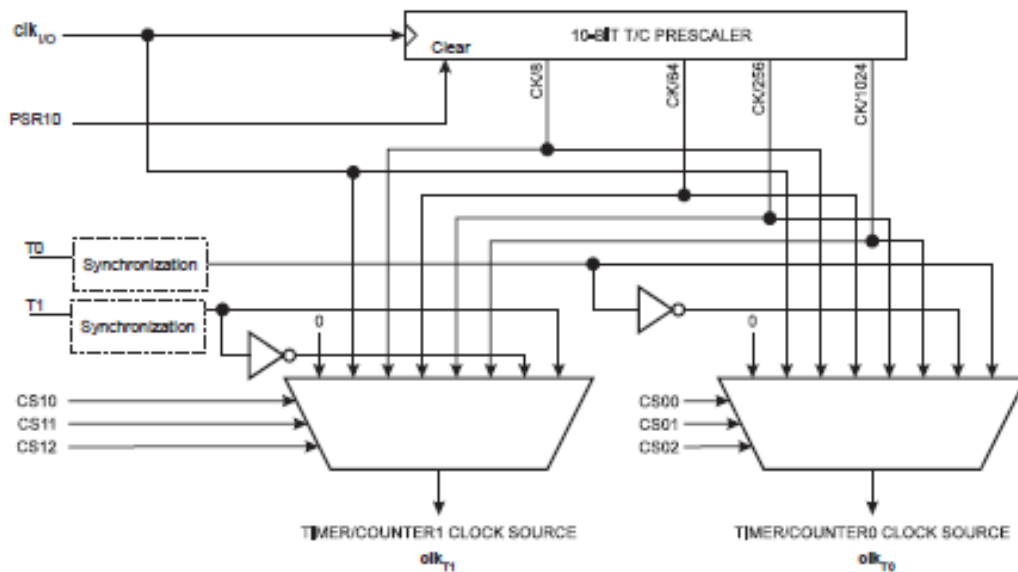


**Slika 46** – Blok shema 16-bitnog Timer/Counter sklopa

Timer/Counter može koristiti unutarnji ili vanjski izvor takta. Neaktivan je kad nije selektiran ni jedan izvor takta. Osnovne značajke:

- 16 bitna arhitektura
- Dvije neovisne *Output compare* jedinice
- Jedna ulazna *capture* jedinica
- *Input capture noise canceler*
- *Clear timer on compare match (CTC)*
- Varijabilni PWM period
- Frekvencijski generator

Svaki blok Timer/Counter sklopa koji postoji u ugradbenom računalo može koristiti takt procesora, čime je omogućena najveća brzina rada ili se alternativno može koristiti djelilo takta (Slika 47).



**Slika 47** – Dijelilo takta za Timer/Counter0 i Timer/Counter1

### 8.7.7 Električne karakteristike

Slikom (Slika 48) prikazane su apsolutno maksimalne vrijednosti. Sve što nadilazi iste, može izazvati trajna oštećenja ugradbenog računala.

Operating Temperature .....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground .....	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage .....	6.0V
DC Current per I/O Pin .....	40.0 mA
DC Current $V_{CC}$ and GND Pins.....	200.0 mA

**Slika 48** – Apsolutno maksimalne vrijednosti

Aproksimacija potrošnje energije ugradbenog računala izvodi se koristeći podatke proizvođača dane tablicom (Tablica 24).

**Tablica 24** – Pregled potrošnje za različita stanja

Symbol	Parameter	Condition	Typ.	Max.	Units
ICC	Power Supply Current	Active 4Mhz, Vcc = 3V	1,3	2,5	mA
		Idle 4Mhz, Vcc = 3V	0,25	0,6	mA
	Power- Down mode	WDT enabled, Vcc = 3V	4	10	uA
		WDT disabled, Vcc = 3V	<0,15	2	uA

## 9 Zaključak

Za potrebe diplomskog rada izrađen je prototip opisanog sustava koji se sastoji od centralne, senzorske i izvršne jedinice. Za svaku jedinicu ručno su izrađene tiskane pločice sa svim komponentama i smještene su u zaštitno kućište. Svaka jedinica testirana je najprije zasebno, a nakon toga i u sklopu prototipa. Postignuti se rezultati u cijelosti poklapaju s očekivanjima opisanim u ovom radu, uz napomenu da su zadovoljeni i svi postavljeni zahtjevi. Preostaje promatranje, kontrola i ispitivanje rada i funkcionalnosti prototipa kroz dulji vremenski period u kojem će se moći provjeriti aproksimacije radnog vijeka baterijski napajanih jedinica.

Osim fizičkog prototipa, razrađene su i neke metode obrade signala korištenih za potrebe kompenzacije i baždarenja senzora. Premda iste daju relativno dobre i zadovoljavajuće rezultate, za pretpostaviti je da postoji prostor za daljnje usavršavanje. Primjerice, u slijedećim verzijama mogli bi se razviti mehanizmi za automatsku kompenzaciju i baždarenje, a funkcije regresije trebale bi biti višeg reda.

Očekivani problemi mogli bi nastupiti prilikom testiranja prototipa u netipičnim atmosferskim uvjetima, zbog nepredvidivih smetnji. Također, ne može se sa sigurnošću predvidjeti ponašanje sustava u punoj konfiguraciji koja podrazumijeva velik broj senzorskih i izvršnih jedinica uz jednu centralnu jedinicu. Iako su unaprijed osmišljeni mehanizmi zaštite od kolizija, moguće je da nakon određene veličine sustava zakažu. U tom slučaju problem će se riješiti drugačijim pristupom ili pak uvođenjem neke nove jedinice (komunikacijski čvorovi).

Centralna jedinica ostvarena je uz pomoć računala koje omogućava fleksibilno korisničko sučelje, koje se može jednostavno mijenjati i nadograđivati. Istovremeno pruža korisniku informacije o radu sustava i omogućava upravljanje sustavom.

Ukoliko se kao problem prezentirane centralne jedinice postavi potreba za neprestanim napajanjem iz električne mreže, isti će se riješiti realizacijom

centralne jedinice kao zasebnog sklopa s ugradbenim računalom, primopredajnim modulom i baterijskim napajanjem.

U narednim verzijama ovog sustava, također će se razmatrati i mogućnost solarnog napajanja uređaja, čime bi se smanjila potreba za periodičkim pregledima instaliranih jedinica.

## 10 Literatura

[1] Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates

[2] Draper, N.R. and Smith, H. Applied Regression Analysis Wiley Series in Probability and Statistics (1998)

[3] Madjar S., Šoštarić J. Navodnjavanje poljoprivrednih kultura. Osijek: Sveučilište J.J. Strossmayer, 2009.

[4] Wikipedia, 24. 5. 2012., *Nickel – metal hydride battery*, [http://en.wikipedia.org/wiki/Nickel%E2%80%93metal\\_hydride\\_battery](http://en.wikipedia.org/wiki/Nickel%E2%80%93metal_hydride_battery)

## **11 Naslov**

Sustav za automatizirano navodnjavanje poljoprivrednih površina

Automatic irrigation system for agricultural surfaces



## 12 Sažetak

Optimizacija uzgoja poljoprivrednih kultura ključni je uvjet za razvoj prehrambene industrije u smislu opće ljudske potrebe za hranom. Sustavi za navodnjavanje u procesu uzgoja poljoprivrednih kultura igraju vrlo značajnu ulogu. Cilj prezentiranog sustava za automatizirano navodnjavanje poljoprivrednih površina (u nastavku sustav) je dodatna optimizacija postojećih sustava za što racionalniju i svrsishodniju potrošnju vode. To se postiže potpunom automatizacijom sustava koji se dijeli na tri vrste sklopova: Senzorska jedinica, Centralna jedinica i Izvršna jedinica. Svaka ima mogućnost bežične komunikacije s drugim jedinicama. Tako senzorska jedinica šalje podatke o vlažnosti tla u centralnu jedinicu koja na temelju obrade podataka otvara ili zatvara ventile navodnjavanja i to preko izvršne jedinice. Mjerenje vlažnosti tla provodi senzorska jedinica korištenjem *Watermark* senzora uz odgovarajuće sklopovlje koje je podvrgnuto kompenzaciji i baždarenju kako bi se dobile precizne i stabilne informacije.

Senzorska i izvršna jedinica lokalno su upravljane ugradbenim računalom ATtiny2313A, a za bežičnu komunikaciju koriste primopredajni modul RFM22B koji ostvaruje domet i preko 1 km.

Centralna jedinica koristi isti primopredajni modul kao i ostale jedinice, ali je upravljana računalom uz korisnički program koji kroz korisničko sučelje omogućuje korisniku nadzor i upravljanje sustavom.

Sustav je projektiran za rad s minimalnom potrošnjom električne energije, pa se svaka udaljena jedinica može napajati korištenjem dvije punjive baterije (1.2 V), uz garanciju neprekidnog rada od 10 mjeseci.

## **Abstract**

Optimization of cultivation of agricultural crops is a key condition for the development of food industry in terms of universal human needs for food. Irrigation in the process of growing agricultural crops plays a very important role. The aim of the presented system for automated irrigation of agricultural surfaces (the system) is further optimization of existing systems and rational and effective use of water. This is achieved by fully automating the system. It is divided into three types of components: the sensor unit, central unit and execution units. Each has the option of wireless communication with other units. Sensor unit sends data on soil moisture in the central unit and then central unit opens or closes irrigation valves on the basis of data processing through the execution units. Sensor unit measures soil moisture using Watermark sensors with appropriate hardware, which is subject to compensation and calibration in order to obtain accurate and stable information.

Sensor unit and executive unit are controlled locally by microcontroller ATtiny2313A, and for wireless communication is used transceiver RFM22B that achieves range over 1 km.

The central unit uses the same transceiver as other units, but is controlled by a computer program which through the user interface allows the user to control and manage the system.

The system is designed to operate with minimal power consumption, so that each remote unit can be powered using two rechargeable batteries (1.2 V), with a guarantee uptime of 10 months.

## 13 Ključne riječi

navodnjavanje, evapotranspiracija, bežična komunikacija, primopredajni modul, senzor vlage tla, kompenzacija, baždarenje, linearna regresija, kvadraturna regresija, ugradbeno računalo, komunikacijski protokoli, elektromagnetski ventil, H-most, punjive baterije

### **Keywords**

irrigation, evapotranspiration, wireless communication, transceiver, soil moisture sensor, compensation, calibration, linear regression, regression quadrature, microcontroller, communication protocols, solenoid valve, H-bridge, rechargeable batteries

## 14 Privitak

- tablice mjerenja za potrebe kompenzacije
- tablice mjerenja za potrebe baždarenja
  
- kodovi senzorske jedinice
- kodovi izvršne jedinice
- kodovi korisničkog programa

## Tablica mjerenja za potrebe kompenzacije

Napon [V]	Frekvencija min [Hz]	Frekvencija max [Hz]	Srednja frek. [Hz]	Devijacija [Hz]
Kratki spoj				
3,29	14178	14196	14187	18
3,11	13971	13992	13981,5	21
2,8	13548	13569	13558,5	21
2,6	13218	13236	13227	18
2,49	13026	13044	13035	18
2,4	12837	12852	12844,5	15
2,31	12630	12648	12639	18
2,21	12384	12405	12394,5	21
2,12	12123	12138	12130,5	15
2	11751	11769	11760	18
1,9	11400	11412	11406	12
1,8	10977	10989	10983	12
			<b>3204</b>	<b>17,25</b>
Međufrekvencija 1				
3,28	906	906	906	0
3,09	909	909	909	0
2,79	909	909	909	0
2,51	909	909	909	0
2,4	909	909	909	0
2,3	912	912	912	0
2,2	912	912	912	0
2,1	912	912	912	0
2	912	912	912	0
1,9	912	912	912	0
1,8	909	909	909	0
			<b>6</b>	<b>0</b>
Međufrekvencija 2				
2,4	3051	3054	3052,5	3
2,3	3045	3048	3046,5	3
2,2	3036	3039	3037,5	3
2,1	3027	3030	3028,5	3
2	3015	3015	3015	0
1,9	3003	3003	3003	0
1,8	2982	2985	2983,5	3
			<b>69</b>	<b>2,142857143</b>
Međufrekvencija 3				
2,4	6315	6363	6339	48

2,3	6318	6324	6321	6
2,2	6255	6270	6262,5	15
2,1	6204	6213	6208,5	9
2	6141	6147	6144	6
1,9	6066	6075	6070,5	9
1,8	5970	5979	5974,5	9
			<b>364,5</b>	<b>14,57142857</b>
	Međufrekvencija 4			
2,4	10530	10554	10542	24
2,3	10192	10407	10299,5	215
2,2	10227	10248	10237,5	21
2,1	10056	10074	10065	18
2	9846	9864	9855	18
1,9	9630	9636	9633	6
1,8	9339	9351	9345	12
			<b>1197</b>	<b>44,85714286</b>
	Međufrekvencija 5			
2,4	12732	12753	12742,5	21
2,3	12519	12534	12526,5	15
2,2	12273	12291	12282	18
2,1	12000	12015	12007,5	15
2	11691	11706	11698,5	15
1,9	11331	11343	11337	12
1,8	10917	10923	10920	6
			<b>1822,5</b>	<b>14,57142857</b>
	Suhi senzor			
2,4	51	51	51	0
2,3	51	51	51	0
2,2	51	51	51	0
2,1	51	51	51	0
2	51	51	51	0
1,9	51	51	51	0
1,8	51	51	51	0
			<b>0</b>	<b>0</b>
	Otvoreni krug			
2,4	48	48	48	0
2,3	48	48	48	0
2,2	48	48	48	0
2,1	48	48	48	0
2	48	48	48	0
1,9	48	48	48	0
1,8	48	48	48	0
			<b>0</b>	<b>0</b>

## Tablica mjerenja za potrebe baždarenja

Očitanje [Hz]	Očitanje [cB/kPa]		Očitanje [Hz]	Očitanje [cB/kPa]
72	199		1470	27
313	177		1500	27
334	162		1530	26
373	140,5		1548	26
400	128		1581	25
411	126		1599	25
436	114		1638	25
458	107,5		1692	24
485	100		1812	22
503	94		1836	22
548	84		1974	21
593	76		2019	20
618	72		2085	20
623	72		2106	19
734	57		2187	19
786	53		2271	18
800	52		2300	18
842	49,5		2513	16,5
903	46		2587	16
993	41		2654	16
1164	33,5		2815	15
1200	33		3192	13
1227	32		3437	12
1278	31		3599	12
1281	31		3987	11
1362	29		4837	6
1374	29		5564	2
1380	29		5619	5
1410	28		5858	1,5
1437	27,5		6029	1
1449	27,5			

## Kodovi senzorske jedinice

```
#define F_CPU 4000000UL
#define SN 0x0001
#define Router 0x0000

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include "mySPI.h"
#include "rfmspi.h"
#include "rf22.h"

uint8_t RF_temp(void);
uint8_t RF_batt(void);
int RunFreqMeter(uint8_t SenzorNo);
int getF(int f);
void Lullaby(void);
void Obrada_podataka(void);
uint8_t Wait_ack(void);

volatile uint16_t N=0;
volatile char TimeOut = 0;
volatile uint8_t mod=0;
uint8_t Swm=0;
uint8_t Err=0;

int main(void)
{
    //Interrupt Setup-----
    GIMSK |= (1<<4);
    GIMSK |= (1<<INT0);

    //Timer1 SetUp-----
    TCCR1B |= (1 << WGM12);           //CTC mod
    TIMSK |= (1 << OCIE1A);          //Enable CTC interrupt

    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    PRR = 0x07;                       //Gasi sve osim Timer1

    PORT_init();                       //SPI i ostalo
    RFM22_init();
    RF_pair(Router,SN);
    RF_dBm(0);
    RF_wut(2);

    spiWrite(0x07,0x20);               //WUT mod

    sei();

    Obrada_podataka();

    while(1){

        Lullaby();                     //Priprema za sleep mod

        if(mod==4) {                   //Ako je RFM inicirao buđenje
            Obrada_podataka();
            Err&=~(1<<1);
        }
    }
}
```



```

    }
    else Err|=(1<<1);
}
}

void Obrada_podataka(void) {
    uint8_t paket[10];
    uint8_t Pokusaj=5;
    int mj;

    //--Stvaranje paketa--
    paket[0] =(uint8_t) (SN>>8);
    paket[1] =(uint8_t) SN;

    mj=RunFreqMeter(1);
    paket[2]=(uint8_t) (mj>>8);
    paket[3]=(uint8_t) mj;

    mj=RunFreqMeter(2);
    paket[4]=(uint8_t) (mj>>8);
    paket[5]=(uint8_t) mj;

    paket[6]=RF_temp();
    paket[7]=RF_batt();

    paket[8]=Err;

    //--Slanje paketa i primanje potvrde--
    while(Pokusaj>0) {
        send(paket,9);
        mj=Wait_ack();
        if(mj==0) { _delay_ms(200); Pokusaj--; continue;}
        else if( mj == 0xAA ) { Err&=~((1<<0)|(1<<2)|(1<<3)); return;}
        else if((mj&0xF0)==0xE0) {RF_wut(mj&0x0F); Err|=(1<<2); return;}
        else if((mj&0xF0)==0xD0) {RF_dBm(mj&0x0F); Err|=(1<<3); return;}

    }

    Err|=(1<<0);
}

uint8_t Wait_ack(void) {
    int k;

    spiWrite(0x0E, 0x02); //2-RX 1-Tx antena
    spiWrite(0x07,0x25); //RX mod, WUT mod

    //--Time-out cca. 1s--
    k=F_CPU/(256*1)-1;
    OCR1A = k;
    TimeOut=0;
    TCNT1=0;
    TCCR1B |= (1 << CS12);

    while(1) {
        if(mod==3) {
            TCCR1B &= ~(1 << CS12);
            if(spiRead(0x4B)>1) return 0;
            return spiRead(0x7F);
        }
    }
}

```

```

        }
        if(TimeOut==1) return 0;
    }
}

uint8_t RF_temp(void){
    //--Očitaj temperaturu--
    spiWrite(0x12,0x20);
    spiWrite(0x0F,0x80);
    _delay_ms(1);
    return spiRead(0x11);
}

uint8_t RF_batt(void){
    //-- Očitaj napon baterije--
    spiWrite(0x07,0x41);
    _delay_ms(1000);
    return spiRead(0x1B);
}

int RunFreqMeter(uint8_t SenzorNo){
    int R1,R2;
    //--Mjerenje frekvencije--
    Swm=SenzorNo;
    return getF(3);    //Okvir = 3Hz
}

int getF(int F_okv){
    int k;

    //--Proračun vrijednosti brojila za zadani okvir--
    k=F_CPU/(256*F_okv)-1;
    OCR1A = k;
    PORTD |= (1 << PIND5);    //Uključi napajanje senzora
    _delay_ms(100);

    //--Brojanje sinkronih impulsa--
    TimeOut=0; N=0;
    if(Swm==1) PCMSK2 |= (1<<PCINT15);
    if(Swm==2) PCMSK2 |= (1<<PCINT14);
    TCNT1=0;
    TCCR1B |= (1 << CS12);
    while(!TimeOut);

    return N*F_okv;    //Vrati rezultat u Hz
}

void Lullaby(void){
    //--Clear TX/RX FIFO--
    spiWrite(0x08,0x03);
    spiWrite(0x08,0x00);
    //--WUT mod--
    spiWrite(0x07, 0x20);
    //--Deaktivacija I/O portova--
    MCUCR|=(1<<PUD);
    DDRD=0;
    DDRB=0;
    sleep_mode();    //Odlazi u sleep mod
}

```

```

ISR(TIMER1_COMPA_vect){
    //--Obrada Time-out-a--
    PCMSK2 &= ~(1<<PCINT15);
    PCMSK2 &= ~(1<<PCINT14);
    TCCR1B &= ~(1 << CS12);
    PORTD &= ~(1 << PIND5);
    TimeOut = 1;
}

ISR(PCINT_D_vect){
    //--Brojanje impulsa--
    if( ((PIND&(1<<PIND4))==0) && Swm==1 ) N++;
    if( ((PIND&(1<<PIND3))==0) && Swm==2 ) N++;
}

ISR(INT0_vect){ //Buđenje procesora na zahtjev WUT-a
    uint8_t I3,I4;
    mod=0;

    PORT_init(); //Ponovna inicijalizacija

    //--Brisanje zastavica interrupt-a--
    I3=spiRead(0x03);
    I4=spiRead(0x04);

    //--Postavljenje moda rada--
    if(I3&0x02) mod|=3;
    if(I4&0x08) mod|=4;
}

```

## Kodovi izvršne jedinice

```
#define F_CPU 4000000UL
#define SN 0x0100
#define Router 0x0000

#define Fin PIND4
#define Rin PIND5
#define SOL PORTD
#define impuls 50
#define P_pali 65
#define P_gasi 15

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include "mySPI.h"
#include "rfmspi.h"
#include "rf22.h"

uint8_t RF_temp(void);
uint8_t RF_batt(void);
void Lullaby(void);
void Obrada_podataka(void);
uint8_t Wait_ack(void);
void SOL_pali(uint8_t P);
void SOL_gasi(uint8_t P);

volatile uint16_t N=0;
volatile char TimeOut = 0;
volatile uint8_t mod=0;
uint8_t Err=0;

int main(void)
{
    //Interrupt Setup-----
    GIMSK |= (1<<INT0);

    //Timer1 SetUp-----
    TCCR1B |= (1 << WGM12); //CTC mod
    TIMSK |= (1 << OCIE1A); //Enable CTC interrupt

    //Timer0 SetUp (PWM)-----
    TCCR0A |= ((1<<COM0B1) | (1<<WGM00) | (1<<WGM01));
    TCCR0B |= (1<<WGM02);
    OCR0A = 127; //cca. f=31 kHz
    OCR0B = 64; //50%

    set_sleep_mode(SLEEP_MODE_PWR_DOWN);

    PORT_init(); //SPI i ostalo
    RFM22_init();
    RF_pair(Router,SN);
    RF_dBm(0);
    RF_wut(2);

    spiWrite(0x07,0x20); //WUT mod

    DDRD|=(1<<PIND6);
```

```

if (spiRead(0x00)==0x08) PORTD|=(1<<PIND6);
else PORTD&=~(1<<PIND6);
_delay_ms(4000);

sei(); //Omoući globalni interrupt

SOL_gasi(P_gasi); //Inicijalno zatvori ventil

Obrada_podataka();

while(1){

    Lullaby(); //Priprema za sleep mod

    if(mod==4) { //Ako je RFM inicirao buđenje
        Obrada_podataka();
        Err&=~(1<<1);
    }
    else Err|=(1<<1);

}

}

void Obrada_podataka(void) {
    uint8_t paket[10];
    int mj;
    uint8_t Pokusaj=10;

    //--Stvaranje paketa--
    paket[0] =(uint8_t) (SN>>8);
    paket[1] =(uint8_t) SN;

    paket[2]=RF_temp();
    paket[3]=RF_batt();

    paket[4]=Err;

    //--Slanje paketa i primanje potvrde--
    while(Pokusaj>0) {
        send(paket,5);
        mj=Wait_ack();
        if(mj==0) { _delay_ms(200);Pokusaj--;continue;}
        else if(mj == 0xBB) {SOL_pali(P_pali); return;}
        else if(mj == 0xCC) {SOL_gasi(P_gasi); return;}
        else if( mj == 0xAA ) { Err&=~((1<<0)|(1<<2)|(1<<3));return;}
        else if((mj&0xF0)==0xE0) {RF_wut(mj&0x0F);Err|=(1<<2);return;}
        else if((mj&0xF0)==0xD0) {RF_dBm(mj&0x0F);Err|=(1<<3);return;}
    }

    Err|=(1<<0);

}

uint8_t Wait_ack(void) {
    //Isti kao kôd Senzorske jedinice!
}

```

```

uint8_t RF_temp(void){
    //Isti kao kôd Senzorske jedinice!
}
uint8_t RF_batt(void){
    //Isti kao kôd Senzorske jedinice!
}
void Lullaby(void){
    //Isti kao kôd Senzorske jedinice!
}

void SOL_pali(uint8_t P){
    //--Proračun energije paljenja--
    if(P<1) P=1;
    if(P>100) P=100;
    //--Pokretanje PWM-a--
    OCR0B = (P*OCR0A)/100;
    sbi(SOL,Fin);
    TCCR0A |= ((1<<COM0B1)|(1<<COM0B0));
    TCCR0B |= (1<<CS00);
    _delay_ms(impuls);
    //--Gašenje PWM-a--
    TCCR0A &= ~(1<<COM0B1)|(1<<COM0B0));
    TCCR0B &= ~(1<<CS00);
    cbi(SOL,Fin);
    cbi(SOL,Rin);

    Err|=(1<<7);    //Postavi indikaciju paljenja
}

void SOL_gasi(uint8_t P){
    //--Proračun energije gašenja--
    if(P<1) P=1;
    if(P>100) P=100;
    //--Pokretanje PWM-a--
    OCR0B = (P*OCR0A)/100;
    cbi(SOL,Fin);
    TCCR0A |= (1<<COM0B1);
    TCCR0B |= (1<<CS00);
    _delay_ms(impuls);
    //--Gašenje PWM-a--
    TCCR0A &= ~(1<<COM0B1);
    TCCR0B &= ~(1<<CS00);
    cbi(SOL,Fin);
    cbi(SOL,Rin);

    Err&=~(1<<7);    //Obriši indikaciju paljenja
}

ISR(TIMER1_COMPA_vect){
    //--Obrada Time-out-a--
    TCCR1B &= ~(1 << CS12);
    TimeOut = 1;
}

ISR(INT0_vect){
    //Isti kao kôd Senzorske jedinice!
}

```

## Kodovi korisničkog programa

```
// Inicijalizacija FTDI integriranog kruga
public FTDI.FT_STATUS InitializeFTDI ()
{
    DataHandler.DoWork += new DoWorkEventHandler(DataHandler_DoWork);
    DataHandler.RunWorkerCompleted += new
RunWorkerCompletedEventHandler(DataHandler_Completed);

    UInt32 ftdiDeviceCount = 0;
    int i;

    // Determine the number of FTDI devices connected to the machine
    ftStatus = myFtdiDevice.GetNumberOfDevices(ref ftdiDeviceCount);
    // Check status
    if (ftStatus != FTDI.FT_STATUS.FT_OK)
    {
        return (ftStatus);
    }
    // If no devices available, return
    if (ftdiDeviceCount == 0)
    {
        ftStatus = FTDI.FT_STATUS.FT_DEVICE_NOT_FOUND;
        return (ftStatus);
    }
    // Allocate storage for device info list
    FTDI.FT_DEVICE_INFO_NODE[] ftdiDeviceList = new
FTDI.FT_DEVICE_INFO_NODE[ftdiDeviceCount];
    // Populate our device list
    ftStatus = myFtdiDevice.GetDeviceList(ftdiDeviceList);
    // Open first device in our list by serial number
    ftStatus =
myFtdiDevice.OpenBySerialNumber(ftdiDeviceList[0].SerialNumber);
    if (ftStatus != FTDI.FT_STATUS.FT_OK)
    {
        return(ftStatus);
    }
    //FTDI je otvoren!!!!-----

    // Set latency timer
    ftStatus = myFtdiDevice.SetLatency(2);
    if (ftStatus != FTDI.FT_STATUS.FT_OK)
    {
        return (ftStatus);
    }
    // Reset the controller
    ftStatus = myFtdiDevice.SetBitMode(0, 0);
    if (ftStatus != FTDI.FT_STATUS.FT_OK)
    {
        return (ftStatus);
    }
    // need to setup bTemp to set inputs and outputs
    byte bTemp;
    byte CSmask = (byte) (Math.Pow(2, CSIndex));
    byte MOSImask = (byte) (Math.Pow(2, MOSIIndex));
    byte MISOmask = (byte) (Math.Pow(2, MISOIndex));
    byte CLKmask = (byte) (Math.Pow(2, CLKIndex));
    byte IRQmask = (byte) (Math.Pow(2, IRQIndex));
    bTemp = (byte) (CSmask | MOSImask | CLKmask);
```

```

// Set synchronous bit bang mode
ftStatus = myFtdiDevice.SetBitMode(bTemp, 4);
if (ftStatus != FTDI.FT_STATUS.FT_OK)
{
    return (ftStatus);
}
// Set baud rate/bit clock settings
//ftStatus = myFtdiDevice.SetBaudRate(3000000);
ftStatus = myFtdiDevice.SetBaudRate(1000000);
if (ftStatus != FTDI.FT_STATUS.FT_OK)
{
    return (ftStatus);
}

return (ftStatus);
}

// Emulacija SPI protokola
public void makeSPIPacket(uint inData, uint inDataLength, ref byte[]
outData, ref uint outDataLength)
{
    int i, k;
    uint j;

    byte[] packet = new byte[0xFF];
    byte[] sBuffer = new byte[0xFF];

    byte MOSI = (byte)(Math.Pow(2, MOSIIndex)); // make mask
    byte CS = (byte)(Math.Pow(2, CSIndex));
    byte CLK = (byte)(Math.Pow(2, CLKIndex));

    k = (int)inDataLength - 1;
    j = 0;

    packet[j++] = CS; // CS inactive
    packet[j++] = (byte)0x00; // CS active

    for (i = k; i >= 0; i--)
    {
        if (((uint)(Math.Pow(2, i)) & inData) > 0)
        {
            packet[j++] = (byte)(0x00 | MOSI); // CS active CLK
inactive
            packet[j++] = (byte)(CLK | MOSI); // CS & CLK active
        }
        else
        {
            packet[j++] = (byte)(0x00); // CS active CLK inactive
            packet[j++] = CLK; // CS & CLK active
        }
    }
    packet[j++] = (byte)0x00; // CS active CLK inactive
    packet[j++] = CS; // CS inactive

    outDataLength = j;
    outData = packet;
}

// Pisanje preko SPI protokola
public void spiWrite(byte addr, byte data)

```



```

{
    byte[] reg = new byte[1000];
    uint N = new uint();
    Int32 Packet;
    Int32 PacketLen;
    PacketLen = 16 * 2 + 4;
    addr = (byte)(addr | 0x80);
    Packet = (addr << 8) | data;
    makeSPIPacket((uint)Packet, 16, ref reg, ref N);
    writeSPIPacket(reg, N); if (ftStatus != FTDI.FT_STATUS.FT_OK)
    MessageBox.Show("Error spiWrite");
    if (N != PacketLen) MessageBox.Show("Error spiWrite N");
    Thread.Sleep(10);
    if ((ftStatus = myFtdiDevice.GetRxBytesAvailable(ref N)) !=
    FTDI.FT_STATUS.FT_OK) { }
    readSPIPacket(ref reg, ref N); if (ftStatus != FTDI.FT_STATUS.FT_OK)
    MessageBox.Show("Error read");
    return (byte)parseSPIPacket(reg, N);
}

// Periodička provjera dolaznog podatka
private void Ring_Tick(object sender, EventArgs e)
{
    int i;
    byte PinState = new byte();
    byte cmd_pwr = new byte();
    byte cmd_wut = new byte();
    byte cmd_wut2 = new byte();
    byte cmd = new byte();
    cmd = 0xAA;
    Ring.Enabled = false;

    spi.checkPinState(ref PinState);

    if ((PinState & (1 << spi.IRQIndex)) == 0)
    {
        if ((spi.spiRead(0x03) & 0x02) != 0)
        {
            FIFO.length = spi.spiRead(0x4B);
            for (i = 0; i < FIFO.length; i++)
            {
                FIFO.data[i] = spi.spiRead(0x7F);
            }
            //BB-paljenje
            //CC-gašenje
            //AA-nita
            //Fx-WUT config
            //Dx-dBm config
            cmd_pwr = (byte)(0xD0 |
(int.Parse(tx_pwr_var.Text.Substring(0, 1))));
            cmd_wut = (byte)(0xF0 | (int.Parse(wut_var.Text.Substring(0,
1))));
            cmd_wut2 = (byte)(0xE0 | (int.Parse(wut_var.Text.Substring(0,
1))));

            if ((FIFO.data[0] & 0xFF) != 0) //Izvršna jedinica
            {
                if (set_tx.Checked && (!set_tx.Enabled) && ij.Checked) {
cmd = cmd_pwr; }
                else if (set_wut.Checked && (!set_wut.Enabled) &&
ij.Checked) { cmd = cmd_wut2; }
            }
        }
    }
}

```

```

else if (ventil.Checked && ventil.Enabled) cmd = 0xBB;
else if (!ventil.Checked && ventil.Enabled) cmd = 0xCC;
else if (Akcija == 2) { cmd = 0xCC; Akcija = 0; }
else if (Akcija == 1) { cmd = 0xBB; Akcija = 0; }
else cmd = 0xAA;

spi.send_byte(cmd, FIFO.data[0], FIFO.data[1]);

if (!groupBox1.Enabled)
{
    if ((FIFO.data[4] & 0x08) != 0) Enable_All(8);
    if ((FIFO.data[4] & 0x04) != 0) Enable_All(4);
}

}
else //Senzorska jedinica
{
    Wm1 = (FIFO.data[2] * 256 + FIFO.data[3]);
    Wm2 = (FIFO.data[4] * 256 + FIFO.data[5]);
    if ((Wm1 > 1000) && Automatsko.Checked) Akcija = 2;
    if ((Wm1 < 100) && Automatsko.Checked) Akcija = 1;

    if (set_tx.Checked && (!set_tx.Enabled) && sj.Checked) {
cmd = cmd_pwr; }
    else if (set_wut.Checked && (!set_wut.Enabled) &&
sj.Checked) { cmd = cmd_wut; }
    else cmd = 0xAA;

    spi.send_byte(cmd, FIFO.data[0], FIFO.data[1]);

    if (!groupBox1.Enabled)
    {
        if ((FIFO.data[8] & 0x08) != 0) Enable_All(8);
        if ((FIFO.data[8] & 0x04) != 0) Enable_All(4);
    }
}

spi.spiWrite(0x0E, 0x02); //2-RX 1-Tx
spi.spiWrite(0x07, 0x05);
FIFO.Ready = true;
}

}
Ring.Enabled = true;
}

```