

# Realizacija multiplatformske bežične mreže za daljinsko upravljanje i nadzor

Slaven Popović, Ljubiša Branković

Direkcija za tehniku

Telekomunikacije RS, a.d. Banja Luka

Banja Luka, Bosna i Hercegovina

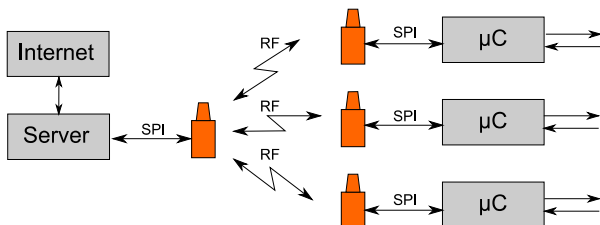
[slaven.popovic@mtel.ba](mailto:slaven.popovic@mtel.ba); [ljubisa.brankovic@mtel.ba](mailto:ljubisa.brankovic@mtel.ba)

*Sadržaj*—Ideja ovog rada je da se u potpunosti omogući nadgledanje procesa (temperature, grijanja, rasvjete, logičkih stanja itd.) bežičnim putem. Pred nas je postavljen zadatak da čovjek putem našeg sistema ima mogućnost da nadzire proces od interesa u svakom trenutku. Osnova ovog sistema je Raspberry Pi platforma koja u ovoj konfiguraciji ima ulogu servera. Na server je spojen Internet i RF modem u ISM (Industrial, Scientific and Medical) radio opsegu koji komunicira sa RF klijentima koji su ugrađeni odnosno postavljeni u objekte čiji procesi, stanja i vrijednosti nas interesuju.

*Ključne riječi*—Daljinski nadzor; RF server; RF klijent; Raspberry Pi; IOS

## I. UVOD

Predmet ovog rada je upotreba bežične mreže za daljinsko slanje i prijem podataka kojim je moguće vršiti upravljanje i nadzor raznim procesima kako u domaćinstvu tako i industriji. Neke od primjena ovog rješenja su očitavanje temperature, vlažnosti, pritiska, raznih brojača, upravljanje grijanjem u prostorijama, upravljanje rasvjetom i slično. Obzirom da na krajnjem nodu u mreži imamo nekoliko ulaznih portova moguće je prikupljanje raznih podataka na daljinu. Takođe centralni dio rjesenja je Raspberry Pi (Single Board Computer) platforma koja ima ulogu servera za RF modeme i servera za komunikaciju prema Internetu. Ova platforma ima skoro sve funkcionalnosti personalnog računara a može se koristiti i za specijalizovane funkcije koje tradicionalno obavlja mikrokontroler koristeći programabilni GPIO interfejs. Na Sl. 1 je data blok šema upravljačkog sistema. Upravljanje se definiše pomoću programskog jezika Python. Kao konzola za upravljanje koristi se Iphone smart telefon sa IOS operativnim sistemom. Konzolni program na RPi platformi za upravljanje napisan je u programskom jeziku Python. Za kreiranje menija i aplikacije na smart telefonu korišten je XCODE IDE.

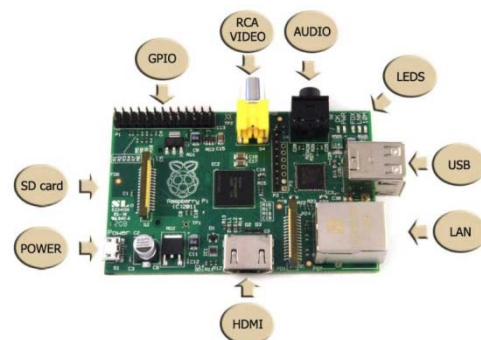


Slika 1. Blok šema upravljačkog sistema

## II. RPI SERVER

Server je realizovan kao Single Board Computer koristeći platformu Raspberry Pi model B [3] (u daljem tekstu RPi) koji na sebi ima ethernet port 100 Mb/s. Karakteristike Raspberry Pi modela B (Sl. 2) su sledeće:

- Broadcom BCM2835 SoC
- 700 MHz ARM1176JZF-S (CPU core)
- Broadcom VideoCore IV grafička procesorska jedinica (GPU)
- 15-pin MIPI camera interface (CSI) connector
- 512 MB RAM memorija
- 2 x USB2.0 porta
- Prošireni 40-pinski GPIO konektor
- Video/Audio izlaz preko 4-polnog džeka od 3.5mm, HDMI ili izvornog LCD (DSI)
- SD / MMC / SDIO card slot
- 10/100 Ethernet (RJ45)
- Niskonaponske periferije: 8 x GPIO, UART, I2C magistrala, SPI magistrala, +3.3V, +5V, masa
- Podržava Debian GNU/Linux, Fedora, Arch Linux, RISC OS i druge.



Slika 2. Raspberry PI model B

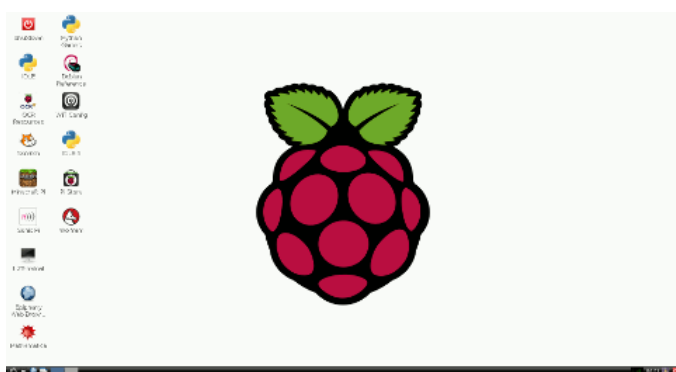
Za proširenje sistema i komunikaciju sa CC1101 čipom korišten je port za proširenje na RPi matičnoj ploči prikazan na Sl. 3.



RPi pin	Naziv	RPi funkcija
1	3,3V	3,3V
25	Gnd	Gnd
19	SPO	GPIO10(MOSI)
24	CLK	GPIO11(SCLK)
21	SPI	GPIO9(MISO)
26	CE	GPIO7(CE1)
22	INT	GPIO25

Slika 3. Pinout i korišteni pinovi

Kao operativni sistem instalirana je Raspbian verzija Linux operativnog sistema (Sl. 4). Raspbian je operativni sistem za RPi koji je baziran na Debianu. Instaliran je na SD karticu kapaciteta 4 GB, što je ujedno i prostor za skladištenje korisničkih podataka programskog koda.



Slika 4. OS Raspbian GUI

Programski jezik korišten za konfiguraciju i komunikaciju sa periferijom je Python verzije 2.7. Python je interpreterski, objektno orjentisani programski jezik, kojeg je 1990. godine prvi razvio Guido van Rossum [1]. Python ne donosi neke nove revolucionarne osobine u programiranju, već na optimalan način ujedinjuje sve najbolje ideje i načela rada drugih programskih jezika. On je jednostavan i snažan istovremeno. Mogućnost lakog portovanja kroz razne platforme dodatno nas je uvjerila da je on pravi izbor za naš rad. Python je besplatan (za akademske ustanove i neprofitnu upotrebu) Open-Source softver sa izuzetno dobrom podrškom, literaturom i dokumentacijom. Za manipulaciju TCP paketima i SPI komunikaciju korištena je korišten je RPIO biblioteka.

### III. KRAJNI NODOVI U BEŽIČNOJ MREŽI

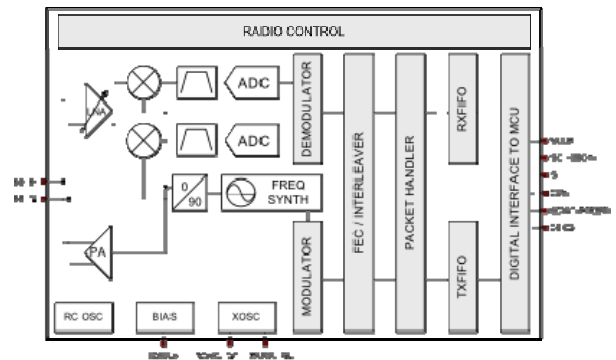
Za bežičnu komunikaciju između nodova i RPi servera korišten je CC1101 čip [6] proizvođača Texas Instruments (bivši Chipcon). CC1100 je RF čip koji radi u ISM (Industrial,

Scientific, Medical) radio bendu. U sebi sadrži prijemnik i predajnik koji su potpuno konfigurabilni korištenjem SPI protokola. Podržava sledeće vrste modulacija:

- GFSK (Gaussian Frequency Shift Keying)
- FSK (Frequency Shift Keying)
- MSK (Minimum Shift Keying)
- ASK (Amplitude Shift Keying)
- OOK (On-Off Keying).

Bitske brzine koje se mogu ostvariti kreću se u rasponu od 1,2 Kbps do 500 Kbps. CC1100 (blok šema data na Sl. 5) u sebi ima hardver za obradu paketa, kao i hardver za analognu obradu primljenog RF signala. Od hardverskih dodataka ima:

- FEC
- CRC
- Data Whitening
- Integrisan analogni senzor temperature
- Programabilan filter propusnog opsega
- Programabilna izlazna snaga do +10 dBm
- Packet Handler (obrada primljenih i poslatih paketa).

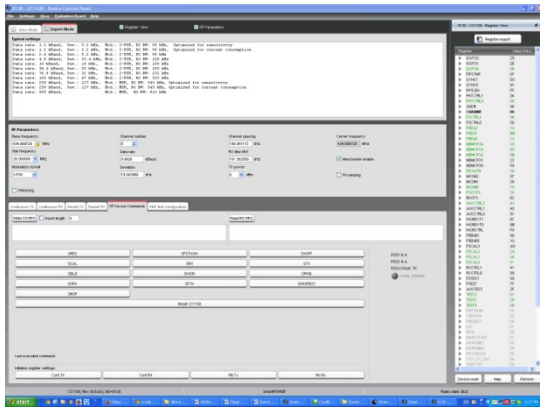


Slika 5. Blok šema CC1101

Tokom razvoja i testiranja optimalnih parametara rada RF djela korišten je razvojni kit CC1101DK433 (Sl. 6) proizvođača Texas Instruments. Za modelovanje optimalne konfiguracije CC1101 čipa i registara zaduženih za RF dio korišten je softver SmartRF Studio 7 (Sl. 7).



Slika 6. Razvojni kit CC1101DK433



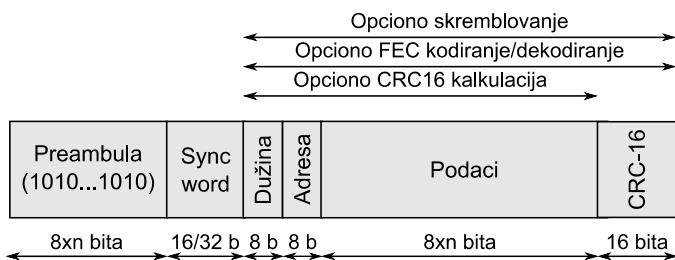
Slika 7. SmartRF Studio 7

Implementirani protokol na RF dijelu nije zahtjevan po pitanju brzine prenosa podataka na relaciji klijent-server, te je kao logičan izbor korištena bitska brzina od 10 Kb/s (Tabela 1).

TABELA I. OSNOVNI PARAMETRI RF DIJELA

Parametar	Vrijednost/Tip
Modulacija	2- FSK
Izlazna snaga	+7dBm
Frekvencija	434,999725 Mhz
Devijacija	19,042 Khz
Bitska brzina	9,926 Kbps
Razmak kanala	199,9511Khz

Kako bi se povećala robusnost i otpornost RF komunikacije na smetnje iskorišten je hardverski modul za "Data Whitening", odnosno za eliminaciju DC komponente u eteru. Takodje primjenjen je FEC algoritam u čipu za detekciju greške. Neophodan uslov za korištenje FEC algoritma je da svi paketi budu iste dužine koja je u ovom slučaju 64 bajta. Ova veličina je uzeta jer je veličina RX/TX bafera upravo 64 bajta.



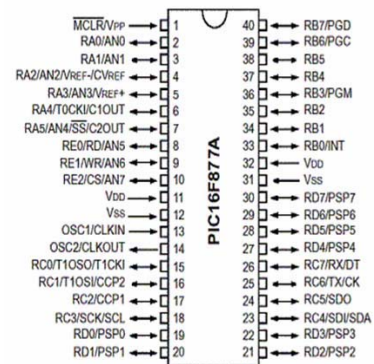
Slika 8. Format paketa za RF komunikaciju

Kako ne bi došlo do istovremenog emitovanja signala u eteru implementiran je CCA (Clear Channel Assessment) algoritam, koji prije svakog emitovanja u provjerava da li je RF kanal slobodan, odnosno da li neko u mreži emituje u tome trenutku. Ukoliko nod registruje emitovanje odlaže slanje za definisano vrijeme i pokušava ponovo da emituje poruku odnosno paket uz ponovno "oluškivanje" etera. Kako bi se smanjilo vrijeme emitavanja paketa moguće je povećati bitsku

brzinu ali to u našem radu nije bilo kritično, tako da smo ostavili skromnu vrijednost od 10 Kb/s koja na obimu od 10 nodova radi bez ikakvih problema.

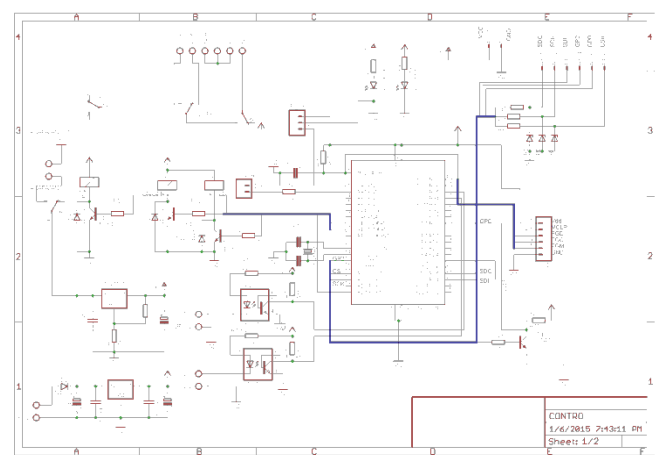
Implementiran je jednostavan protokol za komunikaciju klijent-server (Sl. 8). Na postojeću platformu moguće je implementirati i neki kompleksniji protokol za komunikaciju kao što su ZigBee ili SimpliciTI [2]. U prilog tome ide činjenica da u našoj mreži nemamo potrebe za nodovima u konfiguraciji eksternder/ruter.

Za upravljanje i konfiguraciju krajnjeg noda korišten je mikrokontroler srednje klase, serije uC 16F i to PIC 16F877A (Sl. 9) proizvođača Microchip. Radi se o mikrokontroleru tipične RISC arhitekture sa solidnom listom dostupnih periferija koje su nam na raspolaganju. U našem radu korišten je SPI modul za komunikaciju sa CC1101, A/D konvertor, PWM modul, tajmeri TIMER0 i TIMER1 i Watchdog tajmer.



Slika 9. Izgled PIC mikrokontrolera

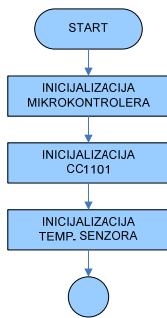
Krajnji nodovi su dizajnirani u programskom paketu EAGLE 6.20 (Sl 10).



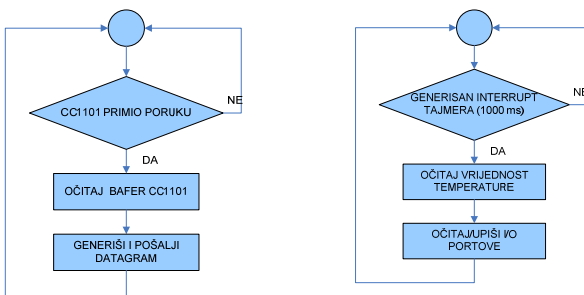
Slika 10. Šema krajnjeg noda

Softver za PIC 16F877 mikrokontroler napisan je u C programskom jeziku [5], korišten je HI TECH pic C kompajler [4] u saradnji sa MICROCHIP MPLAB X IDE razvojnim okruženjem. Na Sl. 11 je dat dijagram toka glavnog programa. Interapt mehanizam na PIC mikrokontroleru (Sl. 12) je iskorišten za obradu paketa i očitavanja odnosno upise u periferne elemente, te je zbog toga glavni program u potpunosti

rasterećen i unutar njega programa vrši se samo konfiguracija registara PIC-a, registara CC1101 i setovanje Watchdog tajmera (Sl. 12).

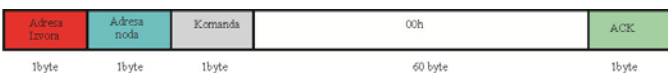


Slika 11. Glavni program

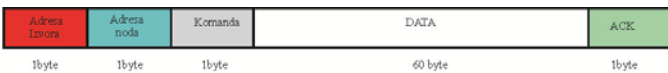


Slika 12. Obrada paketa i očitavanje vrijednosti temperature

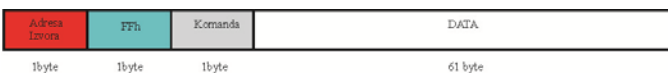
Imajući u vidu da se u našem slučaju radi o komunikaciji tipa point-multipoint nameće se potreba za kreiranjem paketske mreže RF modema, odnosno koristi se paketski dvosmjerni sistem prenosa u konfiguraciji klijent-server. Implementiran je jednostavni protokol za komunikaciju klijent-server (Sl. 13, Sl. 14, Sl. 15).



Slika 13. Paket za komandu nodu



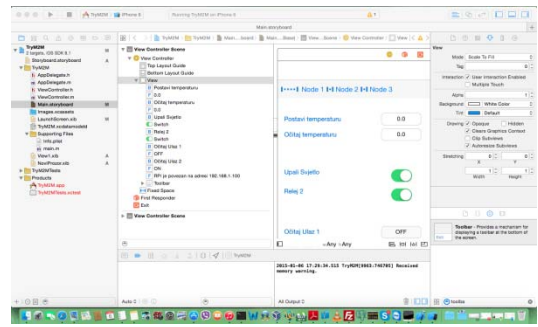
Slika 14. Paket za slanje podataka



Slika 15. Broadcast paket

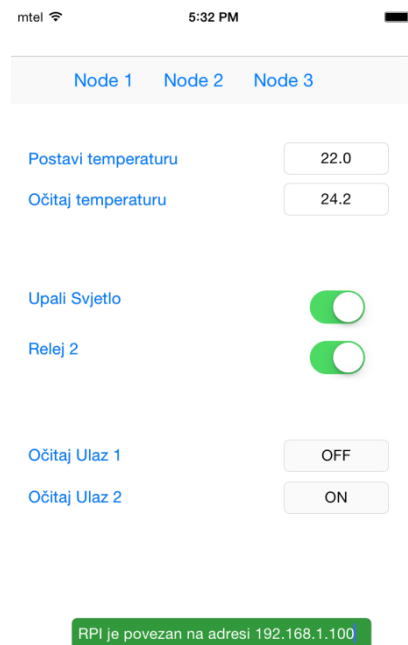
#### IV. IOS APLIKACIJA

Aplikacija je realizovana na IOS Apple platformi korištenjem XCODE 6.1.1 verzije za IOS 8.1. XCODE (Sl. 16) je integrisano razvojno okruženje za OS X i IOS operativni sistem IDE.



Slika 16. XCODE IDE

Grafički interfejs prema korisniku kreiran je koristeći XCODE IDE. Programski jezik u pozadini XCODE-a je Objective-C [7], [8]. Po startovanju aplikacije provjerava se dostupnost RPi servera sa unaprijed definisanom IP adresom i po unaprijed poznatom portu. U našem radu dodjeli smo IP adresu RPi serveru iz privanog opsega IP adresa odnosno 192.168.1.100. Korišteni port je 2012 i to za TCP prenos. U kreiranoj aplikaciji (Sl. 17) ostavljena je mogućnost upravljanja sa tri noda u mreži.

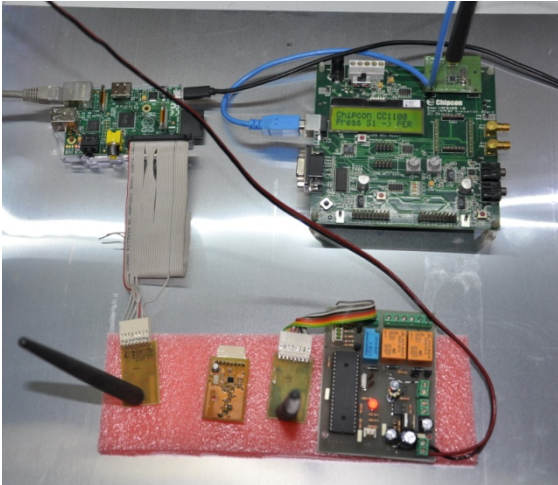


Slika 17. IOS aplikacija za upravljanje

#### V. SISTEM U RADU

Tokom testiranja postavljeno je testno okruženje (Sl. 18) na kome su vršena testiranja softvera i hardvera kao i optimizacija cijelog sistema. Prije nego smo pustili sistem u rad izvršen je niz simulacija za svaku cjelinu pojedinačno koristeći dostupne simulatore (Proteus, XCODE, MPLAB IDLEX) za svaki segment našeg projekta. Na testnoj platformi postavljeno je upravljanje ulaznom kapijom, upravljanje kotlom centralnog

grijanja, mjerenje temperature u prostoriji, PIR senzor i upravljnje rasvjetom u dvije prostorije.



Slika 18. Sistem tokom razvoja i testiranja

## VI. ZAKLJUČAK

U radu je prikazano jedno rješenje bežične mreže za nadzor i upravljanje objektima odnosno procesima na daljinu. Postavljena je solidna platforma koja može da odgovori na mnogo složenije zahtjeve do prikazanog u našem radu. Izazov nam je bio da nekoliko cjelina objedinimo u jedinstven sistem koji će biti u stanju da upravlja procesima sa mobilnog telefona, odnosno preko Interneta. Buduća unapređenja će biti u pravcu podizanja HTTP servera na RPi, razvoj aplikacije za Android operativni sistem, konfiguracija Apache HTTP servera sa statičkom javnom adresom. U trenutnoj verziji nije se vodilo računa o sigurnosti komunikacije i to je nešto što će biti u fokusu interesovanja za neku buduću verziju softvera na RPi platformi.

## LITERATURA

- [1] Learning Python with RaspberryP , Alex Bradbury and Ben Everard, Mart 2014.
- [2] K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons, 2007.
- [3] Raspberry Pi User Guide, Eben Upton , Gareth Halfacree , Septembar 2014.
- [4] HI-TECH C® for PIC10/12/16 User's Guide, Microchip Tehnology 2010.
- [5] Embedded C Programming, Hellebuyck, Chuck 2008.
- [6] Low-Power Sub-1 GHz RF Transceiver CC1101, Texas Instrument Datasheet, 2014.
- [7] Beginning Xcode, James Bucanek , April 3, 2006.
- [8] Programming in Objective-C (6th Edition), Stephen G. Kochan December 13, 2013.

## ABSTRACT

The idea of this paper is to completely enable monitoring of a process (temperature monitoring, heating monitoring, light control, logic states monitoring etc.) using wireless network. Our task was to create system which provides monitoring of a process of interest at any time. Base of this system is a Raspberry Pi platform which is a server in this configuration. The server is connected to the Internet and RF modem. The RF modem works in ISM (Industrial, Scientific and Medical) radio band and communicates with RF clients which are embedded in objects that we want to monitor or control.

### REALIZATION OF THE MULTIPLATFORM WIRELESS NETWORK FOR REMOTE CONTROL AND MONITORING

Slaven Popović, Ljubiša Branković