

IOT MERNO-INFORMACIONI SISTEM ZA PAMETNI PLASTENIK

IOT MEASUREMENT AND INFORMATION SYSTEM FOR SMART GREENHOUSE

Milan Šaš, Fakultet tehničkih nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu prikazan je realizovani merno-informacioni sistem čija primena je u objektima poput staklenika ili plastenika a za cilj ima merenje zadatih fizičkih veličina, ostvari bežični prenos, obradu i prikaz podataka kao i mogućnost automatizacije rada uređaja koji se nalaze u objektu, poput pumpi za vodu, ventila na crevima, grejača, i sličnih uređaja koji mogu da se nađu u objektu. Sistem je projektovan kao „low cost” rešenje i koristi široko dostupne uređaje i senzore.

Ključne reči: SHT35, SHT31, TEMT6000, NRF24L01, Arduino Nano, Raspberry Pi, Python

Abstract – This paper shows the realization of a measuring and information system whose application is in facilities such as greenhouses and aims to measure the physical values, achieve wireless transmission, processing and display of data, as well as the ability to automate the operation of devices located in the building, such as water pumps, valves on hoses, heaters, and similar devices that can be found in the building. The system is designed as a "low cost" solution and uses widely available devices and sensors.

Keywords SHT35, SHT31, TEMT6000, NRF24L01, Arduino Nano, Raspberry Pi, Python

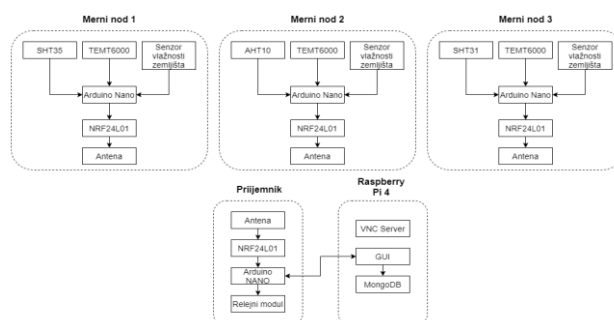
1. UVOD

U današnjem vremenu možemo da primetimo da tehnologija pronalazi primenu u sistemima koji se primenjuju u poljoprivredi a za cilj imaju automatizuju procesa koji se odvijaju.

Sistem koji je projektovan i koji će biti prikazan u ovom radu ima za cilj da pomogne u automatizaciji procesa uzgajanja voća i povrća u stakleniku ili plasteniku.

Automatizacija je sastoji iz merenja vrednosti veličina koje su unapred definisane i regulacije rada uređaja na osnovu izmerenih vrednosti. Veličine koje se mere su temperatura, relativna vlažnosti vazduha, vlažnost zemljišta i nivo osvetljenja.

Uređaji kojima se upravlja su pumpa za vodu, tri ventila, grejalica, osvetljenje i ovlaživač vazduha. Na slici 1 data je blok šema celog sistema.



Slika 1: Blok šema projekta

2. ANALIZA DELOVA SISTEMA

2.1. Merni nod

Merni nod se sastoji od Arduino Nano sistema [1] i NRF24L01 [2] bežičnog modula za prenos podataka. Na slici 2 nalazi se prikaz NRF24L01 bežičnog modula.



Slika 2: NRF24L01 bežični modul

R.B.	Ime senzora	Opseg temperature (°C)	Opseg r.v. vazduha (%)	Komunikacija	Napajanje
1	SHT35	-40 – 125 (0.015 +/- 0.2)	0 – 100 (0.01 +/- 1.5)	I2C	2.4 V – 5.5 V
2	SHT31	-40 – 125 (0.015 +/- 0.3)	0 – 100 (0.01 +/- 2)	I2C	2.4 V – 5.5 V
3	SHT30	-40 – 125 (0.015 +/- 0.3)	0 – 100 (0.01 +/- 3)	I2C	2.4 V – 5.5 V
4	AHT10	-40 – 85 (0.01 +/- 0.3)	0 – 100 (0.024 +/- 2)	I2C	3.3 V – 5.0 V
5	DHT22	-40 – 80 (0.1 +/- 0.5)	0 – 100 (0.1 +/- 2)	One – Wire	3.3 V – 6.0 V
6	AM2320	-40 – 80 (0.1 +/- 0.5)	0 – 99.9 (0.1 +/- 3)	I2C	3.1 V – 5.5 V

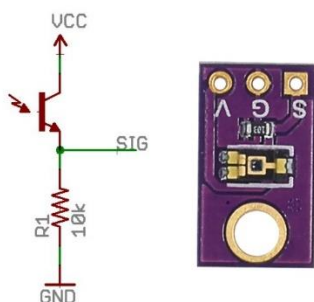
Tabela 1: Specifikacije senzora za merenje temperature i relativne vlažnosti vazduha [3][4][5][6]

Napajanje se vrši pomoću baterije od 9 V a sa dva naponska regulatora postizemo potrebne naponske nivoe od 5 V potrebnog za napajanje mikrokontrolera i senzora i 3.3 V potrebnog za napajanje bežičnog modula. Na sam Arduino su povezani senzori SHT35 [3], SHT31 [3] i AHT10 [4] za merenje temperature i vlažnosti vazduha koji sa Arduinoom komunicira preko I2C (Inter-Integrated Circuit) protokola. U tabeli 1 su date osnovne karakteristike senzora za temperaturu i relativnu vlažnost vazduha. Od zavisnosti od zadatih specifikacija od strane korisnika može se birati senzor ili senzori koji se koriste u sistemu. Kao senzor za merenje nivoa osvetljenosti koristi se senzor TEMT6000 [7] analognog izlaza. Opseg

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Platon Sovilj.

merjenja nivoa osvetljenosti je od 10 lux do 1000 lux a napaja se sa 3.3 V. Sam senzor je realizovan kao naponski razdelnik, a njegov izgled se može videti na slici 3.



Slika 3: TEMT6000 šema (levo) i izgled (desno) senzora

Kao senzor vlažnosti zemljišta koristi se kapacitivni senzor [8] koji daje informaciju preko analognog izlaza u opsegu od 0 V do 3.3 V.

Napajanje se vrši sa 3.3V a na slici 4 prikazan je izgled senzora u verziji 1.2.



Slika 4: Kapacitivni senzor vlažnosti zemljišta (v1.2)

Na svakih 10 sekundi vrši se akvizicija podataka sa senzora, obrada i slanje paketa podataka koji se formatira na sledeći način:

```
nodeID_temp_hum_soil_lux_batlvl
```

gde je:

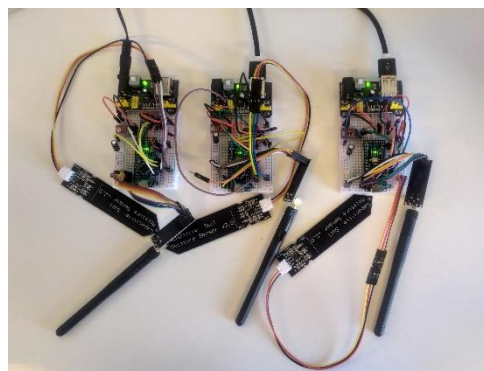
- *nodeID* – ID noda koji šalje podatke
- *temp* – očitana temperatura
- *hum* – očitana vlažnost vazduha
- *soil* – očitana vlažnost zemljišta
- *lux* – očitani nivo osvetljenja
- *batlvl* – preostali procenat baterije

Posle kreiranja paketa podataka koji je potrebno poslati, kreirani paket se prosleđuje bežičnom modulu putem SPI (*Serial Peripheral Interface*) komunikacionog protokola. Komunikacija između prijemnika i predajnika se odvija na 2.4 GHz, koristeći GMSK (*Gaussian Minimum Shift Keying*) modulaciju.

Nakon slanja paketa mikrokontroler i bežični modul ulaze u *sleep* mod kako bi se smanjilo opterećenje na bateriju i produžilo vreme trajanja jedne baterije. Na slici 5 dat je izgled tri merna noda koji se koriste u sistemu.

2.2. Prijemnik

Prijemnik podataka je realizovan pomoću Arduino Nano mikrokontrolera i NRF24L01 bežičnog modula. Sam bežični modul je ovde podešen da radi kao prijemnik i prikuplja podatke koji mu se šalju sa svih mernih nodova.



Slika 5: Prikaz tri merna noda

Pristigle podatke, preko UART (*Universal asynchronous receiver-transmitter*) konekcije, prosleđuje GUI (*Graphical user interface*) programu koji se nalazi na Raspberry Pi [9] platformi. Pored bežičnog modula, na Arduino Nano je povezan i skup od osam relejnih modula koji služe za upravljanje uređajima koji se nalaze u objektu. Upravljanje ovim uređajima se vrši iz GUI-a, koji šalje, preko UART-a, stanja svakog uređaja. Na osnovu poslate poruke mikrokontroler postavlja relejne module u zadata stanja preo svojih digitalnih izlaza. Dodatno, u kod mikrokontrolera je dodata funkcionalnost koja proverava da li je GUI povezan na TX i RX linije tako što na svakih 5 sekundi pošalje poruku kojom proverava prisutnost na drugoj strani. Ukoliko ne dobije odgovor u roku od 10 sekundi, mikrokontroler će postaviti sve uređaje u neaktivno stanje, kako bi sprečio dalji rad bez nadzora. Ova funkcionalnost je korisna u slučaju da se, neočekivano, zatvori program kojim se upravlja ovim sistemom.

2.3. Raspberry Pi 4

Kao centralni deo sistema koristi se Raspberry Pi 4 računar na kome se nalazi GUI aplikacija kojom se upravlja čitavim sistemom i koja će biti opisana u nastavku rada. Sam računar poseduje Broadcom BCM2711 Quad core Cortex-A72 64bit 1.5 GHz procesor [10] i 4 GB RAM memorije. Operativni sistem kao i svi potrebni fajlovi se nalaze na SD kartici od 16 GB. Napajanje se vrši pomoću punjača koji je napravljen za potrebe ovog računara a koji daje 5 VDC/3A na svom izlazu. Pored aplikacije, na RPi se nalazi i MongoDB [11] baza podataka u kojoj se čuvaju svi podaci koji su pristigli od početka rada aplikacije i VNC (*Virtual Network Computing*) [12] koji omogućava korisniku da pristupi RPi računaru sa personalnog računara ili mobilnog telefona. Ovakav način pristupa daje korisniku veću slobodu pri korišćenju celokupnog sistema. Bitno je napomenuti da ovakav pristup zahteva da oba računara budu povezana na istu LAN (*Local Area Network*) i poznavanje IP adrese RPi računara i korisničke lozinke za pristup.

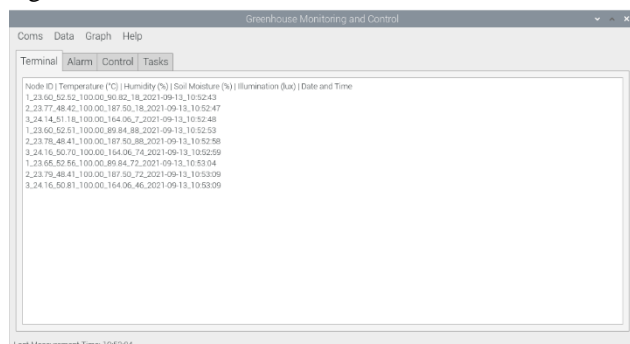
3. GUI APLIKACIJA

Kao što je već rečeno u prethodnom odeljku, na RPi računaru postoji GUI aplikacija kojom se upravlja čitavim sistemom. Sama aplikacija je napisana u Python programskom jeziku i dizajnirana pomoću Qt Designer [13] aplikacije.

Izgled aplikacije podeljen je u četiri celine, odnosno taba. Zadatak aplikacije jeste da prihvata pristigle podatke, obrađuje ih, prikazuje i smešta u bazu podataka. U nastavku rada će biti opisane funkcionalnosti koje se nalaze u svakom tabu.

3.1. Terminal tab

U *Terminal* tabu možemo da pravimo sve poruke koje aplikacija može da prikaže korisniku. To su poruke o novom pristiglom paketu očitanih vrednosti sa nekog mernog noda, uspešnom ili neuspešnom uspostavljanju i započinjanju komunikacije sa prijemnikom, rezultatu postavljanja stanja uređaja i greškama koje mogu da se dogode tokom rada aplikacije. Na slici 6 prikazan je izgled *Terminal* taba.



Slika 6: Izgled Terminal taba

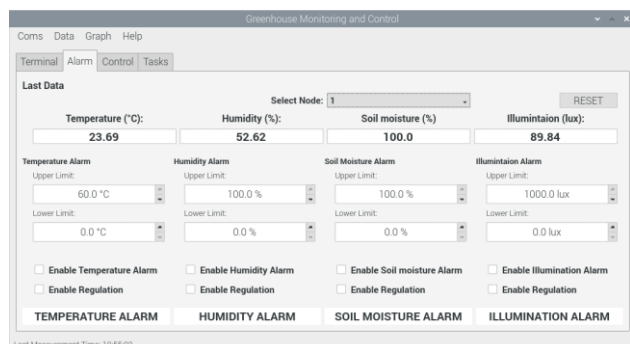
3.2. Alarm tab

U *Alarm* tabu možemo da pratimo poslednje očitane vrednosti sa svakog noda i da, na osnovu tih vrednosti, postavimo alarme za svaku veličinu.

Postavljanje alarma vrši se tako što se izaberu gornja i donja granica za alarm. Ukoliko se poslednja očitana vrednost nalazi van granica zadatog opsega alarm će se aktivirati i polje će biti crvene boje. Ukoliko je poslednja očitana vrednost unutar zadatih granica polje će biti zelene boje.

Ukoliko alarm nije postavljen polje će biti bele boje. Pored mogućnosti postavljanja alarma postoji i mogućnost postavljanja regulacije uređaja na osnovu poslednje očitane vrednosti. Sama regulacije je histerezisnog tipa a za granice histerezisa uzimaju se vrednosti iz polja za granice alarma.

Bitno je napomenuti da u jednom trenutku regulacija može da se obavlja samo na osnovu očitavanja sa jednog mernog noda i da nije moguće postaviti regulaciju na osnovu podataka sa više nodova. Na slici 7 dat je prikaz *Alarm* taba.



Slika 7: Izgled Alarm taba

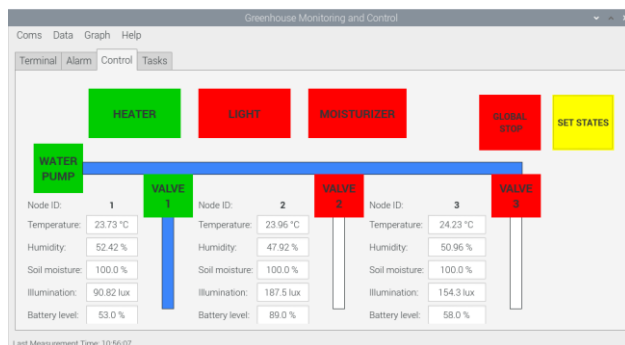
3.3. Control tab

U *Control* tabu imamo mogućnost ručnog postavljanja stanja svakog od uređaja. Stanje uređaja može biti uključen (zelena boja) ili isključen (crvena boja). Kada korisnik odabere stanja svih uređaja potrebno je da izabere opciju *SET STATES* koja prosleđuje sva stanja mikrokontroleru koji daje postavlja releje u zadate pozicije.

Dodatno, kako bi se omogućilo lakše praćenje poslednjih pristiglih podataka, u odeljcima za svaki merni nod prikazuju se poslednji očitani podaci. Ukoliko je potrebno trenutno zaustavljanje svih uređaja to se može učiniti pritiskom na taster *GLOBAL STOP*.

Dodatno, postavljena je svojevrsna zaštita od pogrešnog rukovanja a koja se aktivira kada korisnik greškom želi da uključi pumpu za vodu a ne otvori ni jedan ventil. U tom trenutku se ispisuje poruka upozorenja i pumpa se neće uključiti.

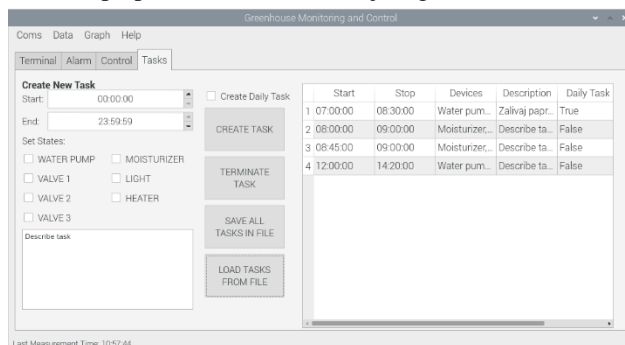
Ukoliko je u tom pokušaju bilo još uređaja kojima se menja stanje to će se uspešno izvršiti. Na slici 8 prikazan je izgled *Control* taba.



Slika 8: Izgled Control taba

3.4. Tasks tab

U *Tasks* tabu korisniku je omogućeno postavljanje zadataka koji treba da se izvrše u toku rada aplikacije. Korisnik može da bira vreme početka i kraja zadatka, stanja svih uređaja tokom trajanja zadatka, pridoda opis zadatka i omogući da se zadatak izvršava svakog dana, u suprotnom on će se izvršiti samo jednom. U tabeli korisnik može jasno da prati sve zadatke koji su postavljeni i da li je i koji zadatak aktivan. Po završetku zadatak, ukoliko se zadatak ne ponavlja i sledeći dan, biće uklonjen iz tabele. Dodatno, korisniku je omogućeno da zadatke koje je postavio sačuva u jedan fajl i da ih iskoristi po potrebi. Na slici 9 dat je izgled *Tasks* taba.



Slika 9: Izgled Tasks taba

3.5. Dodatne opcije

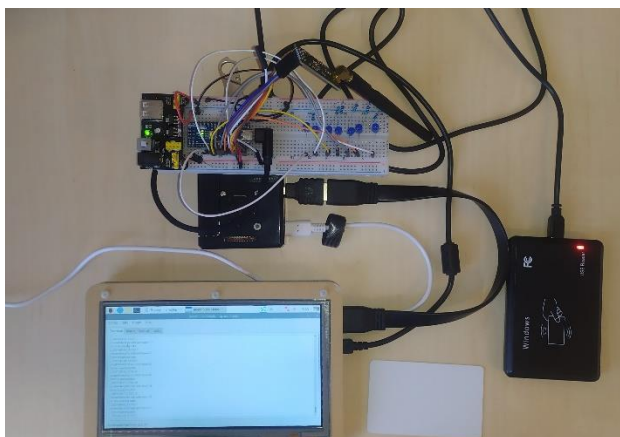
Kako bi se onemogućio pristup aplikaciji od strane neovlašćenih lica implementiran je sigurnosni sistem koji se zasniva na RFID (*Radio-frequency identificacion*) karticama i čitaču za iste. Na slici 10 dat je izgled čitača koji se koristi.



Slika 10: RDIF čitač kartica

Kartice koje se koriste rade na frekvenciji od 125 kHz i na sebi nose već predefinisani kod. U samoj aplikaciji su već implementirani kodovi koji su povezani sa nivoom pristupa koji taj kod omogućava. Postoje dva nivoa pristupa, *user* i *master*. Razlika između dva profila korisnika je u tome što *master* korisnik ima mogućnost logovanja podataka iz baze podataka. Prilikom pokretanja aplikacija je zaključana pa je potrebno očitati karticu korisnika koji pristupa sistemu. Ukoliko, posle tri pokušaja, korisnik ne uspe da očita svoju karticu aplikacija ostaje zaključana a samo *master* korisnik može da otključa aplikaciju koristeći svoju karticu.

U padajućim menijima nalaze se dodatne opcije koje nudi ova aplikacija. To su opcije za podešavanje i pokretanje komunikacije sa prijemnikom, grafički prikaz istorije merenja sa mernog noda, kreiranje log datoteke u koju će biti upisani svi podaci koje se nalaze u izabranom vremenskom opsegu logovanja, brisanje kompletne baze podataka i otvaranje About prozora u kom se nalazi više informacija o samoj aplikaciji. Na slici 11 prikazan je izgled kompletnog sistema.



Slika 11: Izgled kompletnog sistema

3. ZAKLJUČAK

U radu je dato rešenje sistema za merenje i regulaciju i stakleniku ili plasteniku.

Sam sistem je uspešno realizovan i zadovoljava potrebe koje su na početku definisane. Sam sistem može da se proširi kako bi imao više od tri merna noda ali treba uzeti u obzir brzinu obrade jednog paketa podataka kako ne bi dolazilo do zakrčenja podataka.

U slučaju veće mreže mernih nodova potrebno je razmotriti i opciju konfiguracije mreže u *mesh grid* infrastrukturu kako bi se optimizovao put podataka od mernog noda do prijemnika. U daljem razvoju sistema potrebno je projektovati PCB (*Printed Circuit Board*) za svaki merni nod i prijemnik. Sledeći korak jeste razvoj sistema do komercijalnog sistema koji je spreman za upotrebu.

4. LITERATURA

- [1] <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>
- [2] [https://cdn.sparkfun.com/datasheets/Wireless/No rdic/nRF24L01 Product Specification v2 0.pdf](https://cdn.sparkfun.com/datasheets/Wireless/No%20rdic/nRF24L01%20Product%20Specification%20v2%200.pdf)
- [3] <https://pdf1.alldatasheet.com/datasheet-pdf/view/897976/ETC2/SHT35.html>
- [4] <https://www.handsontec.com/dataspecs/sensor/AHT10.pdf>
- [5] <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>
- [6] <https://datasheetspdf.com/pdf-file/952504/Aosong/AM2320/1>
- [7] <https://www.sparkfun.com/datasheets/Sensors/Imaging/TEMT6000.pdf>
- [8] https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0193_Web.pdf
- [9] <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- [10] <https://pdf1.alldatasheet.com/datasheet-pdf/view/1283902/ETC1/BCM2711.html>
- [11] <https://www.mongodb.com>
- [12] <https://www.realvnc.com/en/connect/>
- [13] <https://doc.qt.io/qt-5/qt designer-manual.html>

Kratka biografija:



Milan Šaš rođen je u Beogradu 1997. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Merenje i regulacija odbranio je 2020.god. kontakt: milansas@uns.ac.rs