

# PID regulacija temperature prostorije preko Interneta Android aplikacijom

Mikluc Davorin  
Vojna akademija  
Beograd, Srbija  
miklucd@yahoo.com

Božanić Stefan, Tamara Gajić  
Centar za komandno informacione  
sisteme i informatičku podršku  
Vojska Srbije  
Beograd, Srbija  
zhb.beograd@gmail.com,  
tamara.gajic@vs.rs

Grahovac Darko  
Komanda za obuku  
Vojska Srbije  
Beograd, Srbija  
darkograha@gmail.com

**Sažetak**— Razmena informacija između različitih uređaja i upravljanje istim preko interneta teži da postane neophodna stvar savremenog čoveka. U radu je predstavljena jedna praktičan primer PID regulatora za daljinsko upravljanje i nadzor temperature prostorije preko interneta korišćenjem MQTT protokola. Upotrebom NodeMCU platforme sa WiFi antenom i senzora temperature prikazana je kontrola releja kao i očitavanje trenutne temperature u prostoriji. U radu je predstavljen koncept rada PID regulatora, MQTT protokola kao i njegova praktična realizacija na pomenutom hardveru kojim se upravljanja preko interneta.

**Ključne riječi**—IoT; MQTT; Android; PID

## I. UVOD

U današnje vreme sve je više uređaja, koji sadrže hardver, softver i razne senzore koji su povezani na mrežu, koja im omogućava razmenu podataka. Takva mreža povezanih uređaja naziva se Internet stvari (engl. *Internet of Things*), u nastavku IoT [1] [5]. Jedna od zanimljivih primena su svakako pametni gradovi, gde se koristi mreža senzora, raspoređenih po celom gradu, koja prikuplja podatke, kao što su temperatura, kvalitet vazduha, jačina buke, kao i mnogi drugi parametri. Prikupljeni podaci se obrađuju i koriste za pružanje informacija korisnicima.

Svaki uređaj u mrežnom sistemu je jedinstveno identifikovan na temelju ugrađenog računarskog sistema, i u mogućnosti je da se služi postojećom mrežnom infrastrukturom, zvanom kao Internet [2]. Međutim, uređaji u mreži mogu biti različiti po performansama (hardveru i softveru). Na primer, uređaji koji pomoću senzora prikupljaju podatke i šalju ih na dalju obradu, ne zahtevaju hardver velikih performansi. Jedan od primera uređaja pogodnih za takvu svrhu je uređaj NodeMCU, koji je predstavljen u ovom radu. Važno je posvetiti pažnju i optimizaciji i minimalizaciji kod projektovanja sistema, kako bi se smanjili zahtevi na hardveru, potrošnja električne energije i mrežnog prometa.

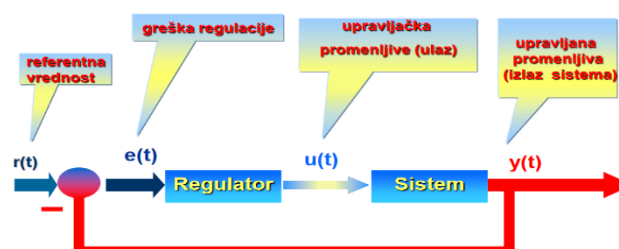
Jedna od najvažnijih stavki kod takvih sistema je komunikacija, odnosno razmena poruka među sistemima [10]. Jedan od najčešće korišćenih protokola u IoT-u je protokol MQTT, čija je namena razmena podataka. Glavna karakteristika ovog protokola je to što je namenjen za korišćenje kod sistema

gde je bitno minimizirati mrežni promet i zahteve hardvera, u uslovima gde je veza nestabilna i gde je mogućnost gubitka poruke velika. Protokol funkcioniše po principu izdavač-pretplatnik, uz slanje poruka putem mrežnog sloja TCP.

U ovom radu demonstriran je rad protokola MQTT tako što je implementirana regulacija temperature prostorije pomoću PID-a. Promena temperature manifestuje se na stanje releja koji simulira grejno telo prostorije. Putem navedenog protokola i očitane temperatura sa temperaturnog senzora sa NodeMCU-a mikrokontrolera može se pročitana temperatura prikazati na Android uređaju. Postoji i mogućnost promene željene vrednosti temperature u prostori koja utiče na uključenost releja tj. grejnog tela. U nastavku će biti opisana PID regulacija, MQTT protokol, korišćeni hardver i biće prikazana Android aplikacija sa opisanim rezultatima testa.

## II. PID REGULACIJA

Svaki sistem automatskog upravljanja (SAU) [11] se odlikuje određenim zakonom ili zakonima upravljanja (Sl. 1). Taj zakon predstavlja matematičku zavisnost na osnovu koje upravljački uređaj obrađuje relevantne (ulazne) signale i generiše odgovarajuća upravljačka dejstva. Najčešća forma ovakvih upravljačkih uređaja se naziva regulator.



Slika 1. Upravljanje sistemom sa zatvorenim povratnom spregom

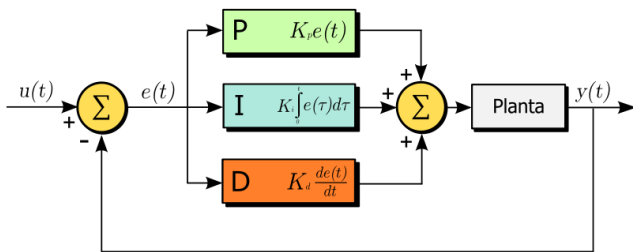
Zadatak povratne sprege je da vraća na ulaz u sistem izmerene izlazne veličine sistema upoređujući ih sa referentnim, željenim vrednostima izlaza. Na taj način se formira greška ili odstupanje stvarnog od željenog ponašanja i ona predstavlja veličinu koju upravljački sistem treba da minimizira.

U odnosu na tip upravljanja dejstvo može linearno zavisiti od greške, zavisiti od njenog integrala ili od prvog izvoda greške po vremenu. U odnosu na pomenuto, postoje: proporcionalni, integralni i diferencijalni regulator. Bez obzira na tip regulatora i način njegove realizacije, osnovni zahtevi za regulaciju sistema su: stabilnost, tačnost i brzina odziva.

Svi ovi regulatori bez obzira na način rada treba da prilikom procesa regulacije obezbede odgovarajuću stabilnost i preciznost, kao i odgovarajuću brzinu regulacije. PID regulacija obezbeđuje da svi ovi zahtevi budu ispunjeni jer obezbeđuje odgovarajuću reakciju na bilo koju promenu regulisane veličine a takođe omogućava i kvalitetnije upravljanje sistemima pri početnom približavanju zadatoj vrednosti.

PID regulator ima tri parametra koja se podešavaju: pojačanje  $K_p$ , integralnu vremensku konstantu  $T_i$  i konstantu diferenciranja  $T_d$  [13]. Prisustvo proporcionalnog, integralnog i diferencijalnog dejstva u ovom regulatoru omogućava dobijanje željenih performansi kao što su: stabilnost, brzina reagovanja, tačnost rada i vreme trajanja prelaznog procesa. Na Sl. 2 prikazana je blok šema PID regulatora, dok je jednačina PID regulacije je opisanja sa:

$$u(t) = K_p + K_i \int_0^t e(t)dt + \frac{K_d(de(t))}{dt} = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right)$$



Slika 2. Blok šema PID regulatora

P regulator predstavlja najjednostavniji tip regulatora koji se opisuje jednačinom:  $u(t) = K_p \cdot e(t)$ . Proporcionalni regulator odlikuje se proporcijalnim područjem koje se definiše kao potrebna procentualna promena ulazne veličine da bi se izlazna veličina promenila za 100%.

I regulator se opisuje jednačinom:  $u(t) = K_i \int_0^t e(t) dt$ . Integralni regulator znatno povećava inertnost sistema, odnosno, sistem sporije reaguje na spoljne uticaje, ali zato u većini slučajeva trajno otklanja grešku rada sistema u stacionarnom stanju. Zbog tih osobina ima masovnu primenu u sistemima kod kojih se upravlja preko releja (da bi se broj naglih aktivacija/deaktivacija releja redukovao i time se produžio njegov radni vek). Negativna osobina ovog tipa regulatora je i destabilizujuće dejstvo u sistemu usled njemu svojstvenog kašnjenja.

D regulator opisuje jednačina:  $u(t) = K_d (de(t))/dt$ . Ovaj diferencijalni regulator se ne upotrebljava samostalno, jer je u ustaljenom režimu rada signal greške konstantan, te je izvod ovog signala jednak nuli. Zbog osobine da je promenljiva  $u(t)$  proporcionalna brzini promene (prvom izvodu) greške u vremenu vidi se da bi D regulator reagovao samo na brze

promene dok spore i dugotrajne promene ne bi prouzrokovale nikakvo dejstvo ovog regulatora. Kombinovanjem sa P i/ili I regulatorom, ovaj regulator dobija na značaju, posebno u prelaznom režimu rada sistema. Njegovo postojanje omogućava bolje praćenje dinamike sistema, jer se njime prati veličina promene greške, a ne samo njena apsolutna vrednost [12]. Uvođenjem diferencijalnog regulatora se povećava stabilnost i brzina reagovanja sistema.

Na osnovu navedenog, vidi se da postoje dva osnovna parametra  $K_p$  i  $T_i$ . Pojačanje  $K_p$  u odsustvu integralnog dejstva predstavlja promenu  $u(t)$  za jediničnu promenu greške  $e(t)$ . Podešavanje integralnog dejstva se vrši integralnom vremenskom konstantom  $T_i$ . U slučaju velikog proporcionalnog područja se pojavljuju spore oscilacije sa velikim amplitudama, a one su posledica smanjene brzine reagovanja usled integralnog dejstva. Uvođenjem D regulacije dobija se brza reakcija sistema na promenu veličine greške, a samim tim se povećava stabilnost i brzina reagovanja sistema.

Najzastupljeniji su PID regulatori izvedeni u mikroprocesorskim uređajima i PID regulatori izvedeni kao standardni moduli u programabilnim logičkim kontrolerima.

U kombinaciji sa impulsnim upravljanjem, PI/PID upravljanje daje mogućnost regulacije temperature sa velikom preciznošću. Koristeći hardversku razvojnu platformu NodeMCU sa integrisanim Wi-Fi modulom dolazi se do softverske realizacije precizne regulacije temperature u vidu korišćenja već postojećih biblioteka namenjenih za realizaciju PID regulaciju (PID\_V1.h biblioteka). Na ovaj način se efikasno i ekonomično realizuje PID regulacije jer se korišćenjem jednog razvojnog okruženja dobija celokupan sistem kojim se može upravljati i sa udaljenog mesta, a pritom se ne troši vreme i resursi da bi se svako kolo PID regulatora hardverski posebno realizovalo.

Za razvoj programskog koda koje je pisano u C++ korišćeno je komercijalno okruženje Arduino IDE verzija 1.8.5.

Posle uključivanja biblioteke u projekat pomoću komande `#include <PID_v1.h>`, potrebno je generisati promenljive:

- Setpoint (željena vrednost koju korisnik želi u prostoriji koja se šalje preko Android aplikacije)
- Input (vrednost sa temperaturnog senzora)
- Output (izlaz od kojeg zavisi uključenost releja)
- $K_p$
- $K_i$
- $K_d$

Potrebno je generisati i objekat tipa PID koji ima ulazne parametre opisane gore:

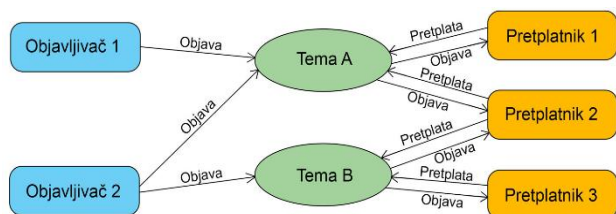
```
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

Kasnije se promenljiva Output koristi kao vrednost kojom se kontroliše stanje releja.

### III. MQTT PROTOKOL

MQTT (engl. *Message Queuing Telemetry Transport*) [3], poznat i po svom starom imenu *MQ Integrator SCADA Device Protocol* je pojednostavljen protokol za komunikaciju između uređaja, baziran na principu izdavač-pretplatnik, koji koristi mrežni protokol TCP/IP [7]. Osmišljen je da bude otvoren, lagan i jednostavan za implementaciju, odnosno da se može koristiti u prostorno ograničenim okruženjima sa ograničenim mogućnostima procesiranja, malim kapacitetom memorije (programske i radne) i sa velikim kašnjenjima. Kao takav, smanjuje zahteve na propusnu moć mreže i predstavlja prihvatljivo rešenje za korišćenje u sistemima sa nestabilnom mrežnom vezom gde prevladavaju male brzine prenosa i gde postoji velika mogućnost za gubitak informacija. Protokol je nastao krajem 1990-ih i početkom 2000-ih godina, a inicijalno je razvijen od strane firme IBM. Danas je to otvoreni protokol.

Princip izdavač-pretplatnik (engl. *publish-subscribe*) [4] je obrazac za razmenu poruka, u kojem pošiljaoci poruka, koji se nazivaju izdavač (engl. *publishers*), ne šalju svoje poruke unapred poznatom primaocu poruke, već ih smeštaju u klase, koristeći teme. Na taj način izdavači ne znaju ko će sve primiti poruku. Pretplatnici (engl. *subscribers*) se pretplaćuju na teme (engl. *topic*) koje ih zanimaju, i tako primaju sve one poruke koje izdavači objavljuju na tu temu. Načelo rada je prikazano na Sl. 3. Pretečom ovog obrasca se smatra sistem za novosti, pod imenom Isis Toolkit, opisan 1987. godine.



Slika 3. Prikaz načela obrasca izdavač-pretplatnik

Poruke koje izdavači objave, potrebno je filtrirati kako bi se poslane odgovarajućim pretplatnicima. Svaki pretplatnik prima određeni podskup poruka, za koje je zainteresovan. Za to je zadužen treći posrednik, koji se naziva broker. Pretplatnici se registruju na brokera i odlučuju na koje se teme žele pretplatiti, a broker zatim sprema poruke koje primi od izdavača i prosleđuje ih pretplatnicima koji su pretplaćeni na tu temu (eng. *store and forward*). Broker može postaviti prioritete poruka u redu poruka, pre nego što ih prosledi dalje. Postoje dva oblika filtriranja:

- 1) Bazirano na temi (engl. *topic-based*)
- 2) Bazirano na sadržaju (engl. *content-based*)

Moguće su i hibridne konfiguracije navedenih oblika, gde se pretplatnici pretplaćuju na određene teme, te postavljaju ograničenja na sadržaj poruka iz tih tema.

Izdavač-pretplatnik obrazac za prenos poruka ima određene prednosti i nedostatke, naspram ostalih oblika prenosa, kao što je na primer tradicionalno korišćeni klijent server (engl. *client-server*) obrazac, a to su:

1. Slaba veza i velika prilagodljivost

2. Nefleksibilno semantičko razdvajanje
3. Poteškoće kod prenosa poruka
4. Problem sa prilagođavanjem
  - Periodi opterećenja (engl. *load surges*)
  - Zastoji (engl. *slowdowns*)
  - Zasićenje usled emitovanja (engl. *IP broadcast storms*)
5. Problem sigurnosti [6]

Protokol MQTT koristi prethodno opisani obrazac objavi-pretplati s pojedinim specifičnostima, za razliku od na primer protokola HTTP, koji koristi obrazac zahtev-odgovor. U središtu se nalazi posrednik (broker), na kojeg se spaja mnoštvo klijenata koji se pretplaćuju na određene teme (engl. *topic*), koje ih zanimaju. Takođe, klijenti se mogu povezati na posrednika i kao izdavači i objavljujući poruke na određene teme, pri čemu ih broker prosleđuje svim klijentima-pretplatnicima koji su pretplaćeni na te teme. Reč je o principu događaja, gde klijent ne mora zahtevati informacije (engl. *pull*) od posrednik, već mu posrednik dostavlja informacije za koje je zainteresovan (engl. *push*), čim se pojavi nešto novo, odnosno čim se desi neki događaj (engl. *event*). Klijent može biti bilo koji uređaj, koji ima implementiran protokol MQTT, od malih ograničenih čvorova u obliku senzorskih sistema s mikrokontrolerom do velikih izdavača. Klijent mora znati za brokera, na kojeg se povezuje putem protokola TCP/IP, pa, ako je reč o pretplatniku, mora znati na koju se temu pretplaćuje.

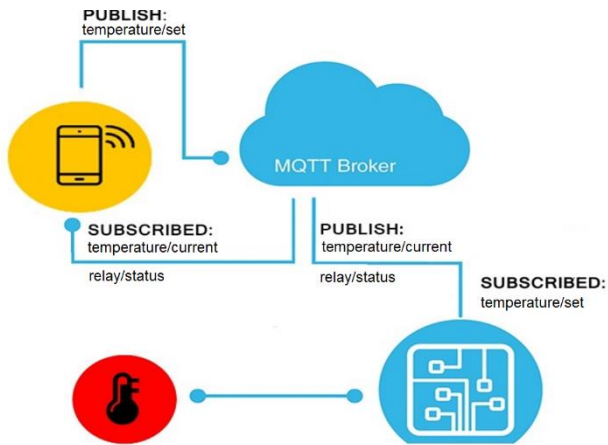
Posrednik (broker) je središnji čvor kod protokola MQTT. Njegova zadatak je da prima objave od izdavača pa ih, pomoću tema i filtriranja, prosleđuje zainteresovanim klijentima-pretplatnicima. Svaki klijent mora znati za posrednika, tako da zna njegovu IP adresu i port, i tek onda ostvaruje trajnu, koja traje sve dok klijent ne zatraži prekid veze, ili dok ne dođe do raskida veze.

Za svrhu ovog rada korišćena je tema „temperature/set“ gde se šalje željena vrednost temperature prostorije. Mikrokontroler se pretplaćuje na tu temu i osluškuje željene temperature korisnika. Mikrokontroler ima dvosmernu komunikaciju. Pročitane vrednosti sa temperaturnog senzora prosleđuje se na kanal „temperature/current“ gde obaveštava sve korisnike Andoird aplikacije o trenutnoj temperaturi. Takođe, NodeMCU šalje i trenutno stanje uključenosti releja na temu „relay/status“. Više o grafičkom prikazu ovih informacija biće u nastavku.

### IV. HARDVERSKA REALIZACIJA

Za implementaciju protokola MQTT je potreban posrednik (broker). Radi jednostavnosti je odabran CloudMQTT posrednik/poslužitelj. Server se nalazi na cloudmqtt.com adresi. Besplatan je do 5 instanci istovremeno i prikladan za ovo testiranje protokola jer se koriste samo 2 instance (mikrokontroler i android uređaj). Taj paket se zove „Cute Cat“. Pored tog paketa CloudMQTT nudi i ostale pakete za više instanci, ali koji se naplaćuju na mesečnom nivou.

Blok šema sistema predstavljena je na Sl. 4.

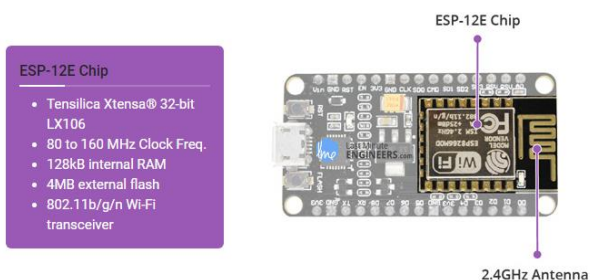


Slika 4. Blok šema sistema

Sistem čini NodeMCU mikrokontroler, jedan temperaturni senzor DS18B20 i jedan relej koja simulira grejno telo.

NodeMCU [8] je povezan sa temperaturnim senzorom. Očitava temperatura sa senzora i tu vrednost objavljuje na temu „temperatura/current“ i preko PID regulacije uključuje i isključuje relej i to stanje releja objavljuje na temu „relay/status“. Korisnička Android aplikacija se pretplaćuje na te teme i te se vrednosti prikazuju korisniku. Svaka promena očitane temperature i stanja releja se manifestuje na aplikaciji. Android aplikacija ima mogućnost promene željene temperature tako što objavljuje vrednost željene temperature na temi „temperature/set“. NodeMCU je pretplaćen na tu temu i u zavisnosti od te vrednosti računica za PID regulaciju se menja koja se na kraju manifestuje promenom stanja na pinu gde se nalazi relej.

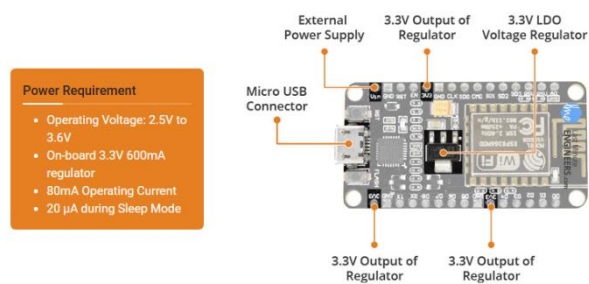
NodeMCU je platforma otvorenog koda (engl. *open-source*) namenjena za modeliranje i izgradnju elektroničkih uređaja. Platforma obuhvata hardver opremljenom ESP-12E modulom koja sadrži čip ESP8266 koji ima Tensilica Xtensa® 32-bit LX106 RISC procesor čiji takt ide od 80 do 160 MHz.



Slika 5. NodeMCU

NodeMCU ima 128 KB RAM memorije i 4MB Flash memorije dovoljno za rad sa velikim stringovima koje čine veb stranice, JSON, XML podaci i sve što se danas koristi u IoT svetu.

ESP8266 integriše 802.11b/g/n HT40 Wi-Fi primopredajnik, tako da može da se poveže sa WiFi mrežom da se komunicira preko Interneta.

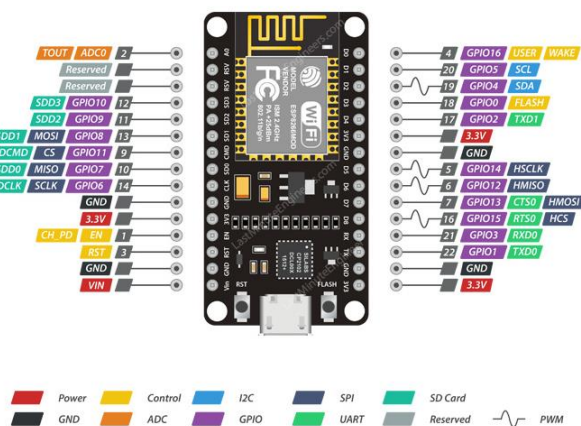


Slika 6. NodeMCU napajanje

Napaja se preko USB-a ili preko spoljnog izvora napajanja i tom slučaju se napaja sa naponom od 7V do 12V (Sl. 6).

ESP8266 NodeMCU ima 17 GPIO (eng. *general-purpose input/output*) pinova (Sl. 7). Različiti pinovi mogu imati različita svojstva i to:

- ADC kanal – 10-bit ADC kanal.
- UART interfejs – UART interfejs se koristi za učitavanje serializacije.
- PWM izlazi – PWM pinovi za kontrolisanje motora.
- SPI, I2C & I2S interfejsi – SPI i I2C interfejsi za povezivanje sa senzorima i periferijama.
- I2S interfejs – I2S interfejs ako postoji potreba za dodavanjem muzike u projektu.



Slika 7. NodeMCU Pinout

Zahvaljujući ESP8266 funkciji multipleksiranja pinova jedan pin GPIO može da funkcioniše kao PWM/UART/SPI.

ESP8266 NodeMCU sadrži dva dugmeta. RST je reset dugme i koristi es da se resetuje ESP8266 čip. Drugo dugme je „flash“ dugme koji služi za preuzimanje prilikom nadogradnje firmvera.

Digitalni vodootporni senzor temperature DS18B20 [9] pruža 9-bitno merenje temperature. Jednožična je komunikacija između DS18B20 temperaturnog senzora i mikrokontrolera (Sl. 8).



Slika 8. DS18B20

Opseg temperature merenja je od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$ . Tačnost merenja je  $\pm 0.5^{\circ}\text{C}$  za temperature od  $-10^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ . Rezolucija senzora je 9-bitna. Napaja se naponom od 3.0V do 5.5V.

Za svrhu ovog rada korišćeni su pinovi D1 (GPIO5) i D2 (GPIO4). Na D1 je povezan DATA žica temperaturnog senzora, a D2 je korišćen kao izlaz i povezan je relej na njega.

## V. SOFTVERSKA PODRŠKA SISTEMA

Korisnička aplikacija koja je pisana u programskom jeziku Java u okruženju Android Studio i za testiranje PID regulacije opisanog sistema instalira je na realnom uređaju Nexus 4 Android telefonu (operativni sistem Lollipop 5.1.1).

Za povezivanje sa brokerom korišćena je biblioteka eclipse.paho.client. Na početku pisanju koda potrebno je promenljive definisati:

```
public MqttAndroidClient client;
final String serverUri =
"tcp://m23.cloudmqtt.com:13634";
final String clientId = "###";
final String username = "###";
final String password = "###";
```

Za uspešno povezivanje sa brokerom potrebno je pozvati komande:

```
client = new MqttAndroidClient(context,
serverUri, clientId);
MqttConnectOptions mqttConnectOptions = new
MqttConnectOptions();
mqttConnectOptions.setAutomaticReconnect(true);
mqttConnectOptions.setCleanSession(false);
mqttConnectOptions.setUserName(username);
mqttConnectOptions.setPassword(password.toCharArray());
```

I konačno potrebno je pozivanje komande za konektovanje sa brokerom koja nadjačava (eng. *override*) dve bitne metode onSuccess i onFailure. Prilikom uspešne konekcije Android klijent se preplaćuje na kanale za osluškivanje vrednosti temperature i stanje releja. Bilo je već govora u glavi broj 3 i 4. To se postiže u metodi onSuccess.

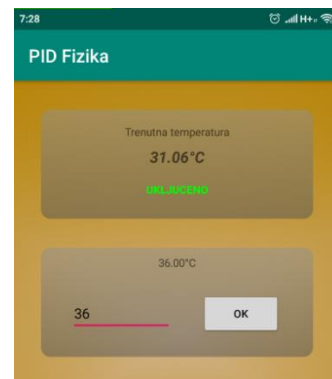
Početni ekran aplikacije posle uspešne konekcije sa brokerom je prikazana je na Sl. 9 gde se grafički prikazuje trenutna temperatura prostorije (u ovom slučaju  $29.75^{\circ}\text{C}$ ), stanje releja tj. grejnog tela (u ovom slučaju isključeno) i pokazuje željenu temperaturu prostorije ( $30^{\circ}\text{C}$ ). Sve ove informacije Android uređaj dobija od cloudmqtt.com servera preko MQTT protokola na kojem je NodeMCU mikrokontroler poslao i šalje na određeni period.

U aplikaciji postoji i mogućnost promene željene temperature prostorije ukucavanjem te vrednosti i pritiskom na dugme „OK“.



Slika 9. Android aplikacija

Kada korisnik promeni željenu temperaturu (Sl. 10) tako da je veća od trenutne, relej se uključuje zbog PID regulacije i takva promena se manifestuje u aplikaciji. Slanje željene temperature se takođe šalje preko MQTT protokola.



Slika 10. Promena željene temperature na  $36^{\circ}\text{C}$

Mikrokontroler i Android aplikacija mogu da šalju i da primaju podatke (dvostrana komunikacija) istovremeno što je pokazano na ovom primeru.

Program za NodeMCU razvojne sisteme je pisan u Arduino IDE 1.8.5 okruženju i korišćene su biblioteke za PID regulaciju (PID\_v1.h), MQTT protokol (ESP8266WiFi.h i PubSubClient.h) i rad sa temperaturnim senzorom (DallasTemperature.h i OneWire.h). Android aplikacija je pisana u Android Studiu 3.2.

## VI. REZULTATI TESTA

Da bi se sistem testirao upotrebljen je MQTT broker m23.cloudmqtt.com koji je besplatan za 5 konekcija. Pet

konekcija su bile sasvim dovoljne za uspešno testiranje sistema (jedna instanca za Android aplikaciju, druga za NodeMCU mikrokontroler). Povremeno se konektovale istovremeno i više instanci Android klijenata i sve instance su videle iste podatke u istom trenutku.

Sistem je testiran 24 sata za vreme kojih je praćena temperatura prostorije i menjano stanje željene temperature. Rezultati su pokazali da nijednog momenta nije došlo do gubitka informacije o temperaturi ili do neočekivane promene stanja releja.

Dalja testiranja bi bila usmerena na zaštitu sistema od neovlašćenih klijenata i brzini odziva sistema.

## VII. ZAKLJUČAK

U ovom radu objašnjen je protokol MQTT prikladan za korišćenje u ograničenim sredinama u mreži IoT. MQTT protokol je prikladan za korišćenje kod međusobne komunikacije više čvorova (engl. *many-to-many*), omogućava prenos poruka među više klijenata putem središnjeg čvora odnosno brokera. Iako postoje mehanizmi za trajnost podataka, MQTT najbolje funkcioniše kao komunikacijska magistrala za podatke. Koristi se protokol TCP za vezu s brokerom. MQTT ne podržava označavanje tipa sadržaja te se poruke mogu koristiti za razne primene, ali klijenti moraju unapred znati tip sadržaja.

Postoje brojni uređaji prikladni za ulogu čvorova u IoT-u, a u ovom radu je opisan NodeMCU uređaj, kao jedan od najpoznatijih i najkorišćenijih uređaja tog tipa. Jednostavnost korišćenja, integrisana WiFi antena, brojni dodaci u vidu hardverskih modula i softverskih programskih biblioteka, odlična podrška i niska cena čine NodeMCU odličnim izborom za korišćenje u IoT-u. U skladu s tim, i u sklopu ovog rada, isproban je MQTT protokol sa očitavanjem temperature, realizacija PID regulatora i prikazivanje vrednosti releja i temperature na korisničkoj Android aplikaciji. Takođe je proverena i mogućnost promene željene temperature koja se manifestuje na uključivanje i isključivanje releja na NodeMCU preko iste te Android aplikacije, i time potvrđena dvostrana komunikacija sa brokerom.

## LITERATURA

- [1] S. Božanić, M. Antonić, "Daljinsko upravljanje i nadzor uređajima preko interneta", Kopaonik, Yu info, March 2017., pp. 242-247

- [2] Dogan Ibrahim: Internet stvari, 2016. godine, Agencija EHO.
- [3] S. Furrer, W. Schott, H. L. Truong, B. Weiss, "The IBM wireless sensor networking testbed", 2nd International Conference on Testbeds & Research Infrastructures for the DEvelopment of NeTworks & CoMMunities (TRIDENTCOM'06), Mar. 2006.
- [4] HWANG, Hyun Cheon; PARK, JiSu; SHON, Jin Gon. Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. Wireless Personal Communications, 2016, 91.4: 1765-1777.
- [5] S. Božanić, S. Sretenović, "Laboratory Example for Controlling Devices in Smart Homes", Bucharest, CERC, May 2017.
- [6] S. Božanić, D. Mikluc and S. Sretenović, "An example of encrypting communication in smart houses", Obrenovac, ICMNEE, September 2017.
- [7] B. A. Forouzan, S. C. Fegan, "TCP/IP Protocol Suite", Mc Graw Hill book, ISBN 0071115838, 3rd edition, New York, 2006.
- [8] <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/> posećeno 10. jan
- [9] <http://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf> posećeno 10. jan
- [10] S. Sretenović, S. Božanić, "Monitoring Air Quality from Location", Bucharest, CERC, May 2017.
- [11] <https://www.automatika.rs/baza-znanja/teorija-upravljanja/osnove-pid-regulatora.html> posećeno 10. jan
- [12] <https://mehatronika.gomodesign.rs/precizna-regulacija-temperature-primenom-pi-regulatora-i-sirinsko-impulsne-modulacije> posećeno 10. jan
- [13] [http://laris.fesb.hr/digitalno\\_vodjenje/text\\_5-5.htm](http://laris.fesb.hr/digitalno_vodjenje/text_5-5.htm) posećeno 10. jan

## ABSTRACT

The exchange of information among different devices and their management over the Internet tends to become necessary thing for a modern man. In this work is presented a practical implementation of PID regulators for remote control and supervising the temperature in the room via the Internet using the MQTT protocol. Using NodeMCU platform with WiFi antenna and the temperature's sensors it is shown the relay's control and scale reading the current temperature in the room. In this work is being shown the concept of work of PID regulators, MQTT protocol and its practical realization of the mentioned hardware, which is being controlled via the Internet.

## PAPER TITLE

Mikluc Davorin, Božanić Stefan, Gajić Tamara i Grahovac Darko