



Uvod

U ovoj laboratorijskoj vježbi ćemo se upoznati sa načinom realizacije komunikacije u okviru ugradbenih sistema. Da bismo omogućili bežičnu komunikaciju po standardu IEEE 802.11, razvojnom sistemu FRDM-KL25Z će biti potrebno dodati WiFi modul. U vježbi će biti demonstrirano i kako se protokol MQTT koristi za upravljanje standardnim ulazima i izlazima razvojnog sistema.

Osnovni pojmovi vezani za komunikaciju u ugradbenim sistemima su pojašnjeni u knjizi [1], kao i na predavanjima [2] i [3], a dokumentacija za Mbed OS API je data na linku [4].

Komunikacija u ugradbenim sistemima

U nastavku će biti pojašnjen način korištenja WiFi modula baziranog na mikrokontroleru ESP8266, kao i struktura sistema koji koristi MQTT protokol za upravljanje standardnim ulazima i izlazima razvojnog sistema FRDM-KL25Z.

WiFi modul na bazi mikrokontrolera ESP8266

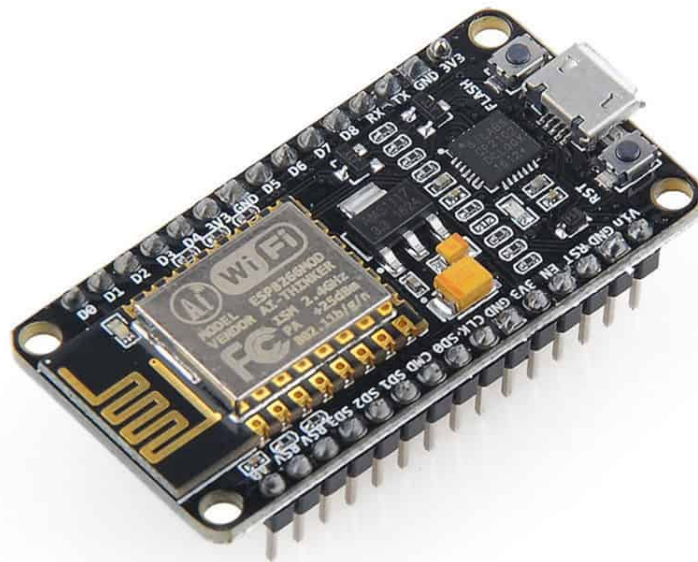
Mikrokontroler ESP8266 je vrlo popularan zbog toga što posjeduje WiFi modul, tako da omogućava realizaciju bežične komunikacije po standardu IEEE 802.11. Mi ćemo ovaj mikrokontroler koristiti u okviru tzv. NodeMCU razvojnog sistema (slika 1). Treba napomenuti da se radi o različitoj arhitekturi u odnosu na ARM sisteme koje koristimo na predmetu Ugradbeni sistemi. Međutim, mi ćemo koristiti već pripremljeni firmware¹, koji omogućava upravljanje ovim razvojnim sistemom putem tzv. AT-komandi². AT (Attention) komande započinju sa karakterima AT, nakon čega slijedi niz karaktera koji definiraju specifičnu komandu. Ukupni set AT komandi koje neki uređaj može izvršiti ovisi o tome o kojem uređaju se radi. Cjelokupan set AT komandi koje razumije mikrokontroler ESP8266 sa AT firmware-om je pojašnjen u [5].

AT komande se mogu zadavati putem serijskog terminala kao što je npr. *Putty*, tako što se unese komanda, koja se završava sa `<cr><lf>`. Uređaj koji izvršava komandu će odgovoriti kao u primjeru 1.

Da bismo mogli direktno komunicirati sa mikrokontrolerom ESP8266, potreban nam je serijski terminal. U okviru ove i budućih laboratorijskih vježbi ćemo za tu svrhu koristiti popularni *Putty*.

¹Za firmware vidjeti <https://en.wikipedia.org/wiki/Firmware>

²Više o AT komandama se može pročitati na https://en.wikipedia.org/wiki/Hayes_command_set



Slika 1: Izgled NodeMCU razvojnog sistema, koji posjeduje WiFi modul.

```
AT+GMR

AT version:1.6.2.0(Apr 13 2018 11:10:59)
SDK version:2.2.1(6ab97e9)
compile time:Jun 7 2018 19:34:26
Bin version(Wroom 02):1.6.2

OK
```

Listing 1: Primjer ispisa na serijskom terminalu pri izvršavanju komande AT+GMR

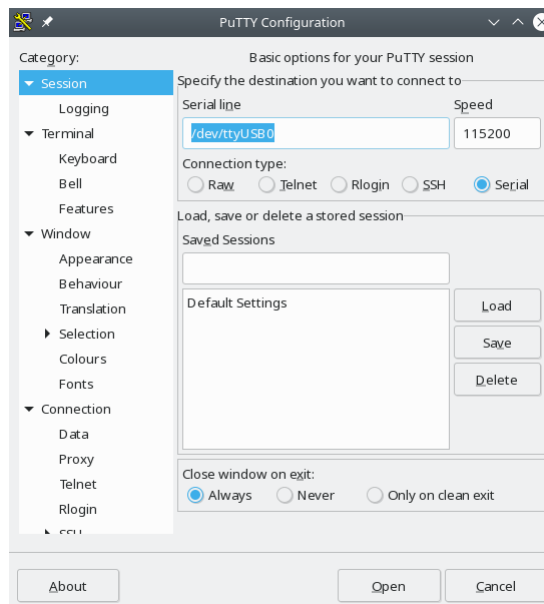
Putty

Putty predstavlja open source aplikaciju koja omogućava korištenje serijske komunikacije, kao i *ssh* (o *ssh* će biti riječi u jednoj od sljedećih laboratorijskih vježbi). Može se preuzeti sa linka <http://www.putty.org/>.

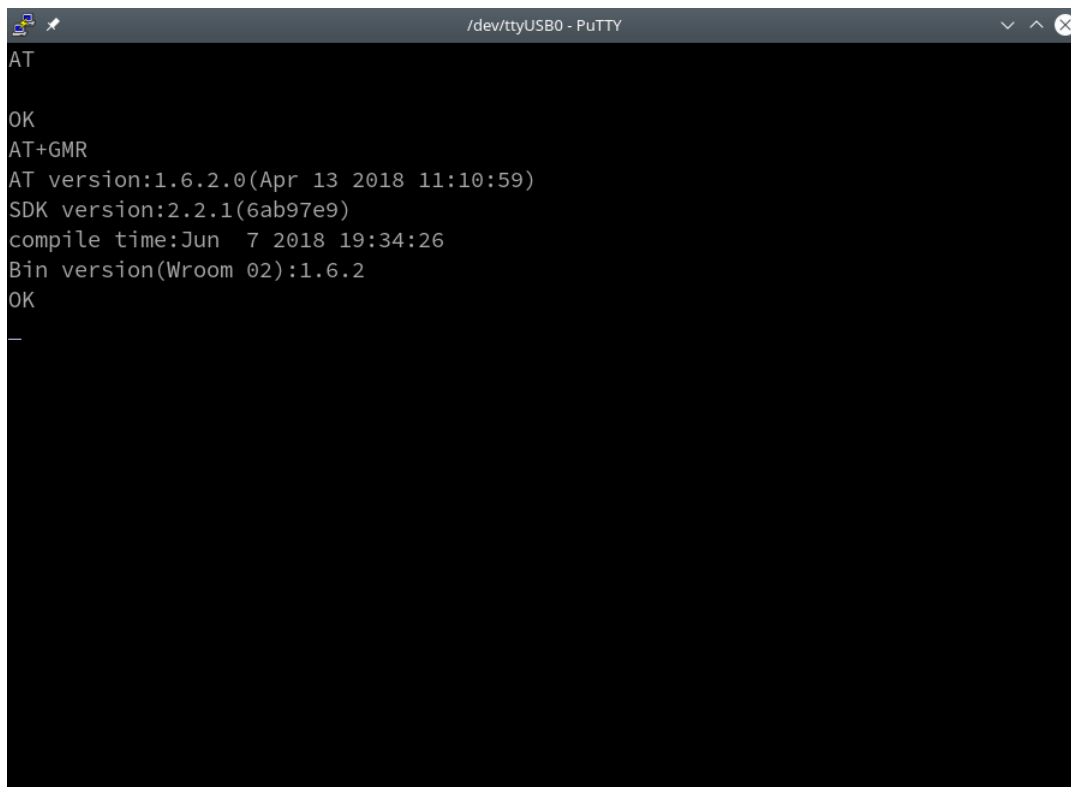
Nakon pokretanja aplikacije, bit će prikazan prozor u okviru koga je moguće podesiti parametre konekcije (slika 2). Voditi računa da se odabere **Serial**, te odgovarajući serijski port i brzina, u skladu sa podešenjima uređaja sa kojim je potrebno ostvariti komunikaciju. Nakon klika na dugme <Open> će biti prikazan prozor serijskog terminala, kao na slici 3.

Putem serijskog terminala se sada mogu zadavati npr. AT komande, a u prozoru će biti prikazan odgovor koji pošalje mikrokontroler ESP8266.

Treba voditi računa da AT firmware mikrokontrolera ESP8266 kao kraj komande očekuje par ASCII znakova <cr><lf> (ne samo <lf>), te je potrebno ili adekvatno podesiti *Putty*, ili ova dva znaka generirati pritiskom na kombinacije tastera <Ctrl>+<M>, te nakon toga <Ctrl>+<J>.



Slika 2: Izgled prozora za podešavanje parametara konekcije.



Slika 3: Izgled prozora serijskog terminala.

Korištenje MQTT protokola

MQTT protokol predstavlja jedan od najčešće korištenih protokola za prenos podataka u IoT sistemima danas. U ovoj vježbi će biti demonstrirano korištenje MQTT protokola za rad sa digitalnim ulazima, digitalnim izlazima, analognim ulazima i PWM izlazima na virtualnom sistemu u Mbed simulatoru, kao i na razvojnom sistemu FRDM-KL25Z. Kao klijentske aplikacije će se koristiti:

- MQTTLens ekstenzija za Google Chrome - jednostavni alat za slanje i prijem MQTT poruka,

- MQTT Dash - aplikacija za Android, koja omogućava kreiranje jednostavnih kontrolnih ploča za slanje i prijem MQTT poruka.

Radi toga je potrebno instalirati MQTTLens ekstenziju u browser, te instalirati sa Google Play-a aplikaciju MQTT Dash. Za iPhone se mogu koristiti slične aplikacije (npr. MQTT Buddy ili MQTTTool).

O MQTT protokolu je bilo riječi u okviru predavanja [3].

MQTT broker

U okviru laboratorijske vježbe ćemo koristiti Mosquitto MQTT broker instaliran na ETF, a podaci za pristup su:

- adresa: 195.130.59.209
- username: ugradbeni
- password: laboratorija

Moguće je koristiti i javni MQTT broker, kao npr. *HiveMQ*:

- adresa: broker.hivemq.com
- username: nema
- password: nema

MQTT Lens

MQTTLens predstavlja ekstenziju za Google Chrome, koja se može instalirati iz Web Store-a.

Nakon instaliranja je potrebno pokrenuti *MQTTLens*, te podesiti parametre konekcije (slika 4).

Add a new Connection

Connection Details

Connection name: HiveMQ

Connection color scheme: [Green bar]

Hostname: tcp:// broker.hivemq.com

Port: 1883

Client ID: lens_INTTMQiqg8FFmGIELVpCqswV8Sk [Generate a random ID]

Session: ☒ Clean Session

Automatic Connection: ☒ Automatic Connection

Keep Alive: 120 seconds

Credentials

Username: Enter username

Password: Enter password

Last-Will: [Dropdown arrow]

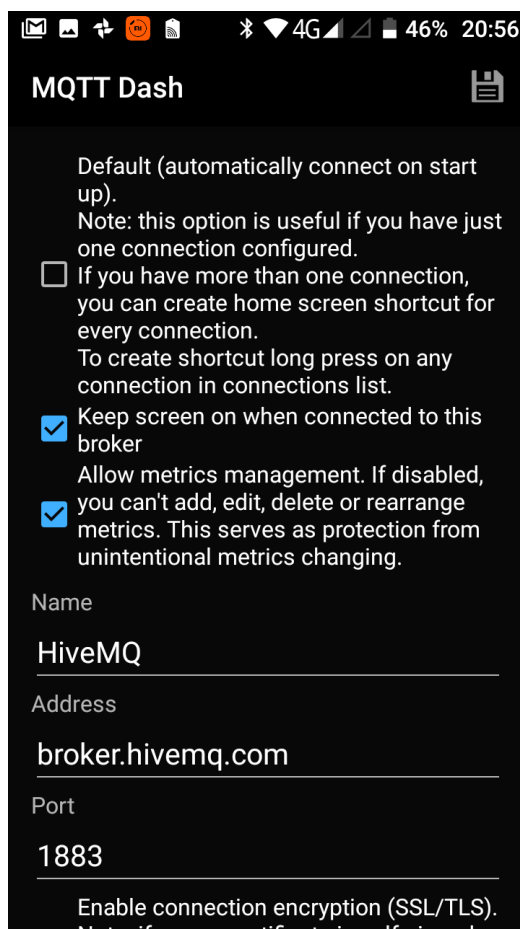
CANCEL SAVE CHANGES

Slika 4: Podešavanje konekcije na MQTT broker u okviru *MQTTLens*.

MQTT Dash

MQTT Dash predstavlja aplikaciju za Android uređaje, koja se može instalirati sa Google Play Store-a. Za iPhone se mogu koristiti slične aplikacije (npr. MQTT Buddy ili MQTTool).

Nakon instaliranja i pokretanja aplikacije *MQTT Dash*, potrebno je podesiti konekciju na MQTT broker. Klikom na simbol + će se pojaviti ekran za unos parametara konekcije, kao na slici 5.



Slika 5: Podešavanje konekcije na MQTT broker u okviru *MQTT Dash*.

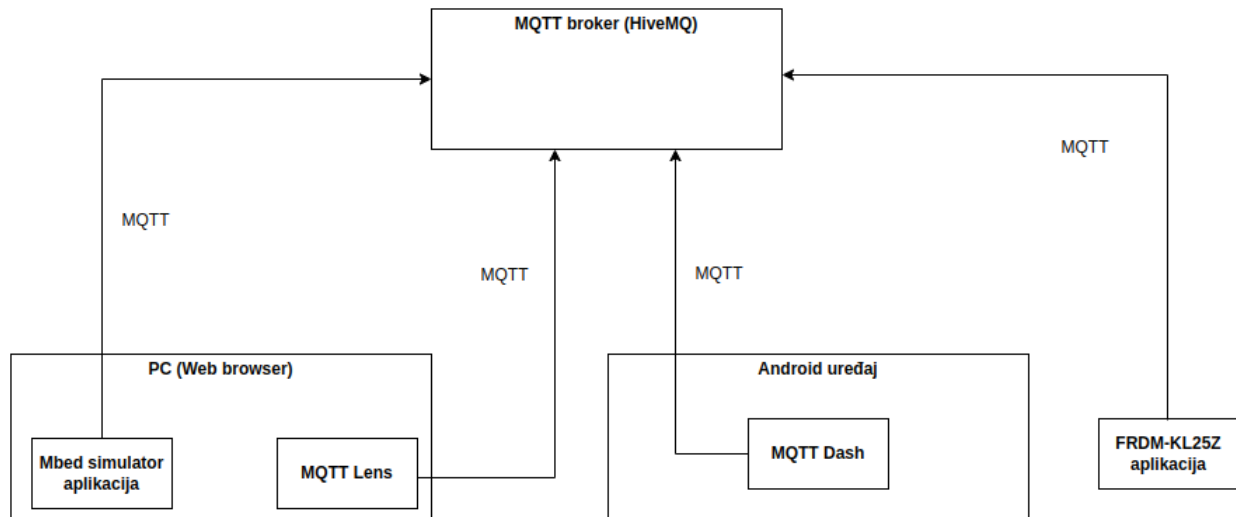
Nakon klika na simbol za snimanje konfiguracije i klika na konekciju HiveMQ, pojaviće se prazan prozor. U okviru ovog prozora se mogu dodavati grafičke kontrole za prikaz i zadavanje vrijednosti.

Pojašnjenje načina rada cjelokupnog sistema

Izgled cjelokupnog sistema je predstavljen na slici 6.

Aplikacija koja se izvršava na Mbed simulatoru, aplikacija koja se izvršava na razvojnom sistemu FRDM-KL25Z, *MQTTLens* i *MQTT Dash* predstavljaju klijente, koji će međusobno razmjenjivati poruke. Te poruke će omogućiti da se na virtualnom sistemu u Mbed simulatoru, kao i na razvojnom sistemu FRDM-KL25Z uključuju i isključuju LED diode (primjer upravljanja digitalnim izlazima), te postavlja intenzitet svjetlosti LED diode (primjer upravljanja PWM izlazom). Poruke koje će postavljati vrijednosti ovih izlaza će se generirati drugim klijentskim aplikacijama (*MQTTLens* i/ili *MQTT Dash*). Isto tako, djelovanje na potencijometar u okviru Mbed simulatora i preko razvojnog sistema FRDM-KL25Z će rezultirati slanjem poruke koju će onda moći procesirati druga dva klijenta (primjer korištenja analognog

ulaza), odnosno pritisak i otpuštanje tastera će rezultirati slanjem poruke na koju će reagirati druga dva klijenta (primjer korištenja digitalnog ulaza).



Slika 6: Podešavanje konekcije na MQTT broker u okviru *MQTT Dash*.

Osnove MQTT protokola su predstavljene na predavanju, te je rečeno da MQTT klijenti međusobno razmjenjuju poruke posredstvom MQTT brokera. Radi toga je na svakom klijentu potrebno podesiti konekciju na MQTT broker, kako je prethodno opisano. Za potrebe ove vježbe će se radi jednostavnosti koristiti MQTT broker HiveMQ, za koji nije potrebno posjedovati korisničko ime i lozinku. Konekcija virtualnog sistema na MQTT broker je već podešena u okviru aplikacije koja će se izvršavati na Mbed simulatoru.

ZADACI

Hint

Mbed simulator se u potpunosti može koristiti za pripremu ove vježbe.

Zadatak 1

Upoznati se sa sadržajem dokumenta [5], u kome je pojašnjen set AT komandi koji poznaje AT firmware za mikrokontroler ESP8266.

U laboratoriji povezati USB kablom NodeMCU razvojni sistem sa računarom, te pokrenuti *Putty*. Konfigurirati brzinu komunikacije od 115200 bps, te upisati oznaku virtualnog serijskog porta na koji je povezan NodeMCU.

Napomena

Ukoliko virtualni serijski port nije prepoznat, na računar je potrebno instalirati driver za CH340(G) USB-to-Serial chip, koji se može pronaći na linku <https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers>

Otvoriti serijsku konekciju, te u terminalu unijeti komandu AT <cr><lf>. Ukoliko NodeMCU modul odgovori stringom OK, to znači da je ostvarena komunikacija.

Hint

Znakovi `<cr><lf>` se mogu generirati pritiskom na kombinacije tastera `<Ctrl>+<M>`, te nakon toga `<Ctrl>+<J>`.

Izvršiti sljedeći niz AT komandi:

- AT+GMR - provjeriti verziju firmware-a na NodeMCU modulu;
- AT+CWMODE? - provjeriti način rada modula (0 - WiFi onemogućen, 1 - Station, 2 - SoftAP, 3 - SoftAP+Station);
- AT+CWMODE=1 - aktivirati Station način rada, kako bi se modul mogao povezati na WiFi mrežu;
- AT+CWLAP - skenirati vidljive WiFi mreže u okolini;
- AT+CWJAP="ETF-WiFi-Guest", "ETF-WiFi-Guest" - povezati modul na mrežu *ETF-WiFi-Guest*;
- AT+CIPSTATUS - provjeriti status konekcije na IP mrežu;
- AT+CIPSTA? - provjeriti parametre konekcije na IP mrežu

Napomena

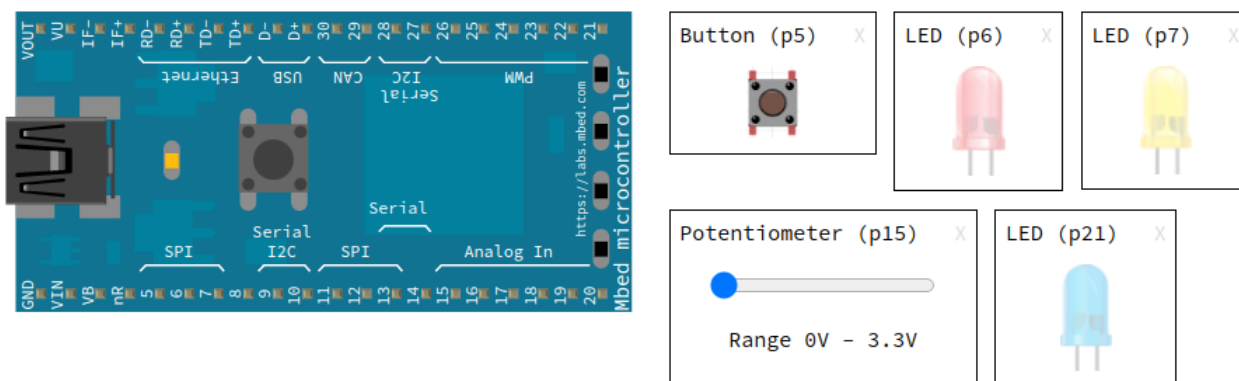
Treba voditi računa da se u okviru terminala cijela AT komanda mora unijeti velikim slovima i bez greške, jer naknadno editiranje nije moguće. Ukoliko se pri unosu komande napravi greška, onda je potrebno unijeti `<cr><lf>`, te nakon što modul odgovori sa ERROR, ponoviti ispravan unos.

Zadatak 2

Mbed simulator

Dodavanje komponenti virtualnom sistemu

U Mbed simulatoru virtualnom sistemu dodati komponente kao na slici 7.



Slika 7: Izgled Mbed simulatora za zadatak 2.

LED na pinu p6 i LED na pinu p7 će biti korištene kao digitalni izlazi kojima se upravlja slanjem poruka sa klijenata *MQTTLens* i *MQTT Dash*. Intenzitet svjetla LED na pinu p21 će se postavljati putem PWM izlaza, a duty cycle će se postavljati klijentima *MQTTLens* i *MQTT*

Dash. Klik na taster na pinu p5 će generirati poruku za druga dva klijenta. Treba napomenuti da će pritisnuti taster rezultirati slanjem poruke sa vrijednošću 1, a otpušten taster slanjem poruke sa vrijednošću 0.

Hint

Da bi se na browseru simulirao pritisak i držanje tastera, potrebno je kliknuti na dugme, te držeći pritisnuto dugme, pointer miša izmaknuti sa tastera. Nakon toga će taster ostati u stanju 1. Za otpuštanje tastera je potrebno kliknuti izvan područja tastera, te držeći pritisnuto dugme, pointer miša pozicionirati na područje tastera i otpustiti dugme, nakon čega će taster preći u stanje 0.

Postavljanje vrijednosti potencimetra će rezultirati slanjem poruke koja sadrži postavljenu vrijednost.

Aplikacija za virtualni sistem u Mbed simulatoru

U Mbed simulator kopirati kôd koji se nalazi u fajlu <https://c2.etf.unsa.ba/mod/resource/view.php?id=89169>. Kako bi se izbjeglo da studenti jedni drugima upravljaju aplikacijama, potrebno je u okviru TEMASUBLED1, TEMASUBLED2, TEMASUBLED3, TEMAPUBPOT i TEMAPUBTAST promijeniti riječ “ugradbeni” u proizvoljni string:

```
#define TEMASUBLED1 "mojproizvoljnistring/led1"
#define TEMASUBLED2 "mojproizvoljnistring/led2"
#define TEMASUBLED3 "mojproizvoljnistring/led3"
#define TEMAPUBPOT "mojproizvoljnistring/potenciometar"
#define TEMAPUBTAST "mojproizvoljnistring/taster"
```

Isto tako, potrebno je u kodu postaviti jedinstveno ime klijenta u linijama `#define MQTT_CLIENT_NAME "MBED_SIMULATOR"` i `#define MQTT_CLIENT_NAME "FRDM_KL25Z"`. Razlog je to što MQTT broker zahtijeva da svaki klijent koji je povezan sa brokerom ima jedinstveno ime, te bi samo jedan od klijenata sa definiranim imenom mogao pristupiti klijentu u jednom trenutku.

Kliknuti na dugme Run, nakon čega će aplikacija biti kompajlirana, pokrenuta, te će se ostvariti konekcija sa MQTT brokerom.

Interakcija sa klijentom MQTTLens

Prikaz vrijednosti potencimetra

U okviru klijenta *MQTTLens*, u polje Subscribe upisati `mojproizvoljnistring/potenciometar` i kliknuti na dugme SUBSCRIBE. Time se klijent *MQTTLens* prijavio na pomenutu temu i sve poruke na tu temu će biti prikazane ispod Subscriptions.

Postaviti proizvoljnu vrijednost potencimetra i pročitati postavljenu vrijednost u okviru Topic: `mojproizvoljnistring/potenciometar`, gdje bi se trebao pojaviti JSON formatirani string kao npr.:

```
{
  "Potenciometar" : 0.45532
}
```


Svaka promjena vrijednosti potenciometra će rezultirati odgovarajućom porukom, koja će biti prikazana u *MQTTLens* klijentu.

Prikaz stanja tastera

U polje **Subscribe** upisati `mojproizvoljnistring/taster` i kliknuti na dugme **SUBSCRIBE**. Time će se klijent *MQTTLens* prijaviti na ovu temu, s tim da će ostati prijavljen i na prethodnu temu (sve dok se prijava na temu ne obriše). Na gore opisani način pritisnuti taster i pročitati stanje tastera u okviru **Topic**: `mojproizvoljnistring/taster`, gdje bi se trebao pojaviti JSON formatirani string, kao npr.:

```
{  
  "Taster" : 1  
}
```

Svaka promjena stanja tastera bi trebala rezultirati odgovarajućom porukom, čiji se sadržaj može vidjeti u klijentu *MQTTLens*.

Prikaz stanja LED dioda

U okviru klijenta *MQTTLens*, u polje **Publish** upisati `mojproizvoljnistring/led1` (slično za `led2`), a u polje **Message** tekst 1 (za uključivanje) ili 0 (za isključivanje), te kliknuti na dugme **PUBLISH**. U okviru Mbed simulatora bi se trebalo promijeniti stanje odgovarajuće LED. Svakim novim slanjem poruke će se stanje odgovarajuće diode promijeniti.

Postavljanje intenziteta osvjetljenja LED diode

U okviru klijenta *MQTTLens*, u polje **Publish** upisati `mojproizvoljnistring/led3`, a u polje **Message** vrijednost između 0.0 i 1.0, te kliknuti na dugme **PUBLISH**. U okviru Mbed simulatora bi se intenzitet osvjetljenja plave LED diode trebao postaviti u skladu sa poslanim Duty Cycle-om. Svaka nova poslana vrijednost će promijeniti intenzitet osvjetljenja plave LED diode.

Interakcija sa klijentom MQTT Dash

Za korištenje klijenta MQTT Dash je potrebno dodati i konfigurisati odgovarajuće grafičke kontrole, kao što je prikazano na slici 9.

Grafičke kontrole se dodaju klikom na dugme "+" u gornjem desnom dijelu prozora, a podešavanje svake od kontrola je prikazano na slikama 1.

Nakon što se sve grafičke kontrole konfiguriraju i snime, vrijednosti grafičkih kontrola "Potenciometar" i "Taster" će početi prikazivati one vrijednosti koje su zadane putem virtualnog sistema u Mbed s imulatoru. Isto tako, klik na grafičku kontrolu LED 1 i LED 2 će izazvati promjenu stanja odgovarajuće LED diode u okviru Mbed simulatora, dok će vrijednost zadana pomoću kontrole PWM LED postaviti intenzitet osvjetljenja plave LED u Mbed simulatoru.

FRDM-KL25Z

Na razvojni sistem FRDM-KL25Z je potrebno spojiti modul NodeMCU, kako je to prikazano na slici 8. Razvojni sistem FDM-KL25Z je potrebno USB kablom povezati sa računarom na uobičajeni način. Modul NodeMCU je potrebno povezati USB kablom sa punjačem napona 5V.

Napomena

Modul NodeMCU se USB kablom ne bi trebao povezati sa računarom, nego sa punjačem iz razloga što se UART koristi za komunikaciju sa razvojnim sistemom FRDM-KL25Z. Ukoliko bi se USB kabl povezao na računar, komunikacija sa razvojnim sistemom FRDM-KL25Z ne bi funkcionirala, jer bi bio inicijaliziran i virtualni serijski port na računar, koji bi zauzeo UART.

U razvojno okruženje Keil Online Studio importovati projekat koji se može preuzeti sa linka <https://drive.google.com/file/d/1cJwZB8fWQZUruHufvLU6DPNRY2OnbEuq/view?usp=sharing>. Ovaj projekat sadrži identičan kôd kakav se izvršava na Mbed simulatoru, sa dvije razlike:

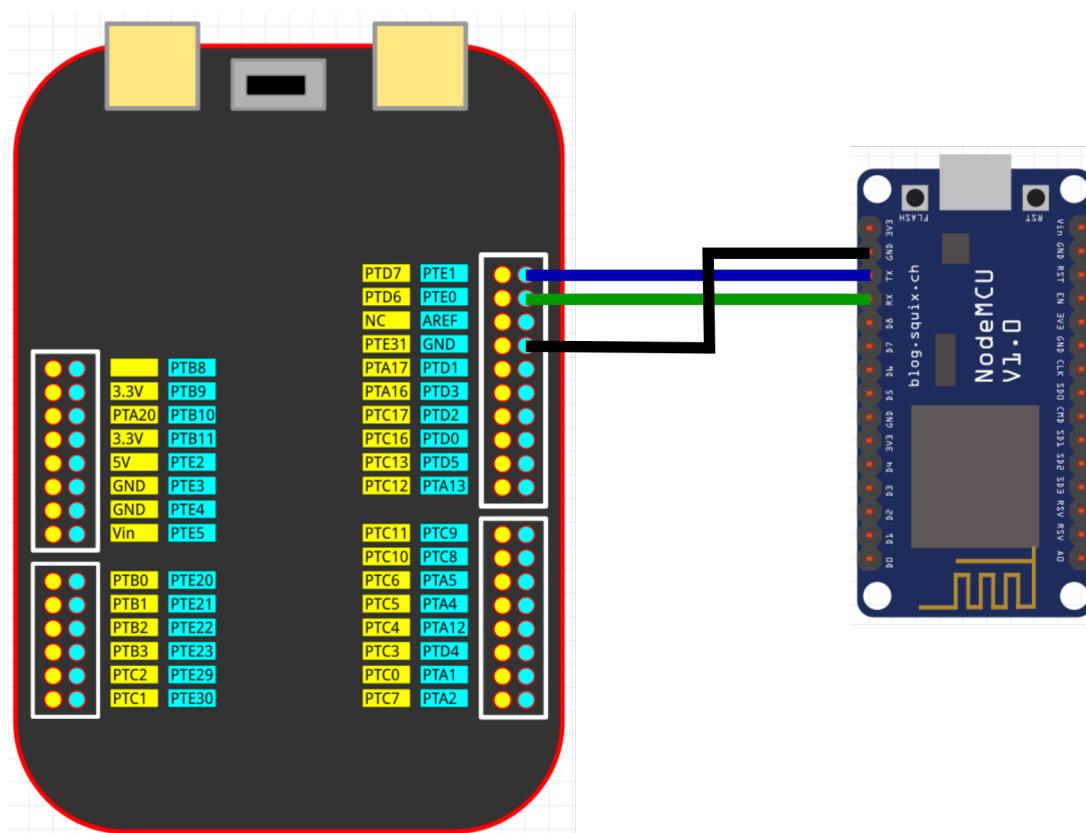
- na Mbed simulatoru i razvojnom sistemu FRDM-KL25Z se koriste različiti pinovi,
- u projekat za razvojni sistem su importovane sve neophodne biblioteke za rad za modulom ESP8266, te ih nije potrebno naknadno dodavati.

Kompajlirati projekat i izvršni fajl prebaciti na sistem FRDM-KL25Z.

Na identičan način kao i sa virtualnim sistemom u Mbed simulatoru testirati razmjenu poruka sa klijentima *MQTTLens* i *MQTT Dash*.

Hint

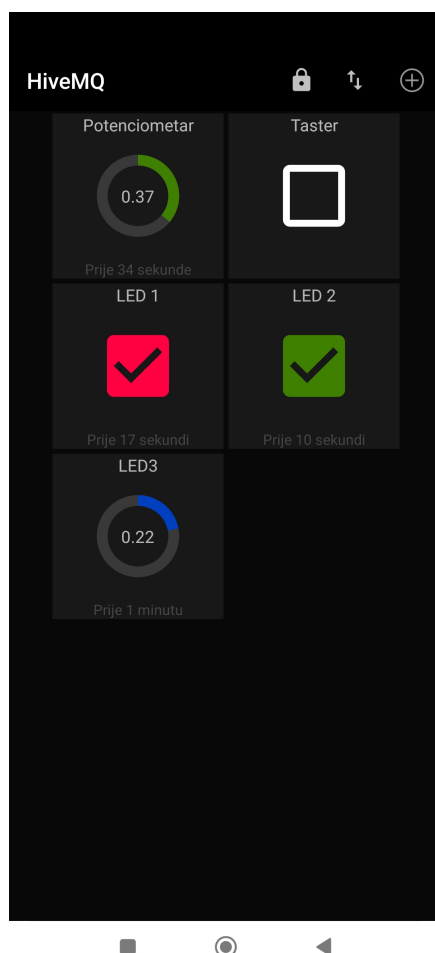
Taster i potencijometar je potrebno povezati sa razvojnim sistemom FRDM-KL25Z, kako bi se moglo testirati slanje podataka sa razvojnog sistema. Eventualno, pritisak na taster se postiže povezivanjem pina D8 razvojnog sistema FRDM-KL25Z direktno na 3,3V, a otpuštanje tastera povezivanjem na 0V.



Slika 8: Povezivanje razvojnog sistema FRDM-KL25Z sa modulom NodeMCU.

Zadatak 3

Kreirati virtualni sistem u Mbed simulatoru i modificirati dati kôd za Mbed simulator, te kôd za razvojni sistem FRDM-KL25Z, tako da se omogući postavljanje proizvoljne boje RGB led diode na sistemu FRDM-KL25Z.



Slika 9: Izgled dashboard-a za zadatak 2 u klijentu *MQTT Dash*.

Literatura

- [1] S. Konjicija, E. Sokić (2019) *Ugradbeni sistemi: Hardverrski aspekti, Elektrotehnički fakultet Univerziteta u Sarajevu, ISBN 978-9958-629-77-8*
- [2] S. Konjicija (2022) *Predavanje Ugradbeni sistemi: Komunikacija (1. dio)*
- [3] S. Konjicija (2022) *Predavanje Ugradbeni sistemi: Komunikacija (2. dio)*
- [4] ARM Holdings (2022) *Mbed OS API Documentation*
- [5] Espressif (2022) *ESP-AT User Guide*

21:53 74%

MQTT Dash

Name
Potenciometar

Topic (sub)
ugradbeni/potenciometar

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>

\$.Potenciometar


☐ Enable publishing

Min **0.0** Max **1.0**

Prefix Postfix

Precision **2**

☒ Display payload value instead of percentage

Progress color


Other settings
☒ QoS(0)
☐ QoS(1)
☐ QoS(2)

Blink tile to draw attention, if the expression

21:53 74%

MQTT Dash

Name
Taster

Topic (sub)
ugradbeni/taster



Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>



\$.taster

☐ Enable publishing

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

On **1** Off **0**

Other settings
☒ QoS(0)
☐ QoS(1)
☐ QoS(2)

Blink tile to draw attention, if the expression evaluates to 'true'.

21:54 74%

MQTT Dash

Name
LED 1

Topic (sub)
ugradbeni/led1

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>



☒ Enable publishing



Topic (pub) - keep empty if the same as sub

☒ Update metric on publish immediately (do not wait for incoming message to update visual state)

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

On **1** Off **0**

Other settings

21:54 74%

MQTT Dash

Name
LED 2

Topic (sub)
ugradbeni/led2

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>



☒ Enable publishing



Topic (pub) - keep empty if the same as sub

☒ Update metric on publish immediately (do not wait for incoming message to update visual state)

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

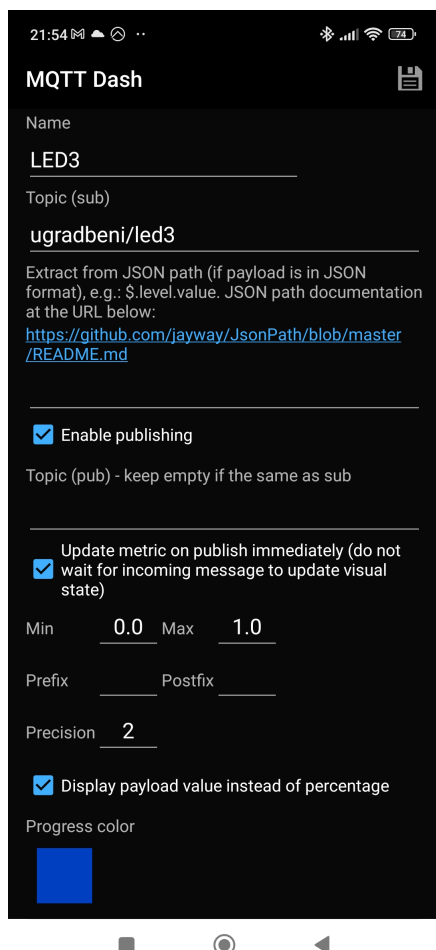
On **1** Off **0**

Other settings

Tablica 1: Podešenja kontrola za zadatak 2.



Tablica 2: Podešenja kontrola za zadatak 2.