

UNIVERZITET U SARAJEVU



ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Laboratorijska vježba 6

Komunikacija - primjer korištenja MQTT protokola

Bazdar Faris

Predmet: Ugradbeni sistemi

Akadska godina: 2019/2020

Sarajevo, april 2020.

1 Zadatak 1

```
#define TEMASUBLEDS "fbazdar/ledice"
#define TEMAPUBPOT "fbazdar/potenciometar"
#define TEMAPUBTAST "fbazdar/taster"

#include "mbed.h"

#define MQTTCLIENT_QOS2 0

#include "easy-connect.h"
#include "MQTTNetwork.h"
#include "MQTTmbed.h"
#include "MQTTClient.h"
#include <string.h>

int arrivedcount = 0;

DigitalIn taster(p7);
DigitalOut led1(p5);
DigitalOut led2(p6);
AnalogIn pot(p15);
PwmOut led3(p21);

char* str;
double pot_value=-1;
bool taster_state=1;

void messageArrived(MQTT::MessageData& md)
{
    MQTT::Message &message = md.message;
    printf("Message arrived: qos %d, retained %d, dup %d,
        packetid %d\r\n", message.qos, message.retained, message
        .dup, message.id);
    printf("Payload %s\r\n", message.payloadlen, (char*)
        message.payload);
    ++arrivedcount;
    str=(char*)message.payload;
    if (!strcmp(str, "led1", sizeof(str))) led1=!led1;
    else if (!strcmp(str, "led2", sizeof(str))) led2=!led2;
    else led3=atof(str);
}

int main(int argc, char* argv[])
{
    printf("Ugradbeni sistemi\r\n");
```

```
printf("Demonstracija koristenja MQTT protokola\r\n\r\n");

NetworkInterface *network;
network = NetworkInterface::get_default_instance();

if (!network) {
    return -1;
}

MQTTNetwork mqttNetwork(network);

MQTT::Client<MQTTNetwork, Countdown> client(mqttNetwork);

const char* hostname = "broker.hivemq.com";
int port = 1883;
printf("Connecting to %s:%d\r\n", hostname, port);
int rc = mqttNetwork.connect(hostname, port);
if (rc != 0)
    printf("rc from TCP connect is %d\r\n", rc);

MQTTPacket_connectData data =
    MQTTPacket_connectData_initializer;
data.MQTTVersion = 3;
data.clientID.cstring = "ugradbeni";
data.username.cstring = "";
data.password.cstring = "";
if ((rc = client.connect(data)) != 0)
    printf("rc from MQTT connect is %d\r\n", rc);

if ((rc = client.subscribe(TEMASUBLEDS, MQTT::QOS2,
    messageArrived)) != 0)
    printf("rc from MQTT subscribe is %d\r\n", rc);

MQTT::Message message;

// QoS 0
char buf[100];
while(1) {
    if (taster_state!=taster) {
        taster_state=taster;
        sprintf(buf, "{\\"Taster\\": %d}", taster.read());
        message.qos = MQTT::QOS0;
        message.retained = false;
        message.dup = false;
        message.payload = (void*)buf;
        message.payloadlen = strlen(buf);
        rc = client.publish(TEMAPUBTAST, message);
```

```
    }
    if (pot_value!=pot) {
        pot_value=pot;
        sprintf(buf, "{\"Potenciometar\": %f}", pot_value);
        message.qos = MQTT::QOS0;
        message.retained = false;
        message.dup = false;
        message.payload = (void*)buf;
        message.payloadlen = strlen(buf);
        rc = client.publish(TEMAPUBPOT, message);
    }
    rc = client.subscribe(TEMASUBLEDS, MQTT::QOS0,
        messageArrived);
    wait(1);
}

}
```

Kod za prvi zadatak

2 Zadatak 2

```
void messageArrived(MQTT::MessageData& md)
{
    MQTT::Message &message = md.message;
    printf("Message arrived: qos %d, retained %d, dup %d,
        packetid %d\r\n", message.qos, message.retained, message
        .dup, message.id);
    printf("Payload %.*s\r\n", message.payloadlen, (char*)
        message.payload);
    ++arrivedcount;

    str=(char*)message.payload;

    if (!strncmp(str,"led1 1",7)) led1 = 1;
    else if (!strncmp(str,"led1 0",7)) led1 = 0;
    else if (!strncmp(str,"led2 1",7)) led2 = 1;
    else if (!strncmp(str,"led2 0",7)) led2 = 0;
    else led3=atof(str);
}
```

Izmjena koda za drugi zadatak

U ovom zadatku bilo je potrebno podesiti kod tako da se može specificirati stanje u kojem se želi da se neka od dioda nalazi. Ovo je urađeno modifikacijom funkcije iz prvog zadatka *messageArrived()*, uvjeti su razdvojeni tako da pokriju sva 4 moguća slučaja komandi koje se mogu zadati putem MQTT klijenta. Postavljen je broj 7 kao ograničenje maksimalnog broja karaktera koji se čitaju pri poređenju stringova funkcijom *strncmp()* da bi se postigla ispravna funkcionalnost.

3 Zadatak 3

```
C12832 lcd(SPI_MOSI, SPI_SCK, SPI_MISO, p8, p11);
...
void messageArrivedForDisplay(MQTT::MessageData& md) // moja
    funkcija za display
{
    MQTT::Message &message = md.message;
    printf("Message arrived: qos %d, retained %d, dup %d,
        packetid %d\r\n", message.qos, message.retained, message
        .dup, message.id);
    printf("Payload %.*s\r\n", message.payloadlen, (char*)
        message.payload);
    ++arrivedcount;
    lcd.cls();
    lcd.locate(0,3);
    lcd.printf("%.*s\r\n", message.payloadlen, (char*)message.
        payload);
}
...
rc = client.subscribe(TEMASUBDISP,MQTT::QOS2,
    messageArrivedForDisplay);
```

Izmjena koda za treći zadatak

U ovom zadatku bilo je potrebno dodati display i omogućiti slanje poruke na isti preko MQTT klijenta. Nakon inicijalizacije displaya, kreirana je funkcija slična funkciji za podešavanje stanja na LED1 i LED2. U principu prvo se resetuje ekran koristeći funkciju `lcd.cls()` te se potom locira lokacija na kojoj se želi vršiti pisanje koristeći funkciju `lcd.locate(0,3)`, i konačno ispisuje se string na display. Korišten je ovakav ispis stringa na display, umjesto učitavanja stringa u neku pomoćnu varijablu najviše iz razloga jer taj način "pamti" string što dovodi do toga da sadržaj na displayu nije tačan ukoliko se idući string unese sa manje karaktera od prethodnog na displayu. Na kraju koda se samo pretplaćuje na specifičnu temu gdje je postavljen display, kako bi se mogla izvršiti komunikacija sa MQTT klijentom.