

Univerzitet u Sarajevu
Elektrotehnički fakultet
Ugradbeni sistemi 2023/24.

Izvještaj za laboratorijsku vježbu 7
Komunikacija - primjer korištenja MQTT protokola

Ime i prezime: **Kanita Kadušić**
Broj index-a: **19327**

Sarajevo, april 2024.

Sadržaj

1	Pseudokod.....	1
1.1	Zadatak 3	1
2	Analiza programskog rješenja	2
2.1	Zadatak 3	2
3	Korišteni hardverski resursi.....	3
3.1	picoETF.....	3
4	Zaključak	4
5	Prilog	5
5.1	Zadatak 3: Izvorni kôd	5

1 Pseudokod

1.1 Zadatak 3

establish connection
MQTT message arrived callback functions for LED control
temperature and humidity reading function
establish connection with the MQTT broker
subscribe to topics

declare Ticker for measuring temperature and humidity

while true
 read button value
 read potentiometer value

2 Analiza programskog rješenja

2.1 Zadatak 3

[1-5] uvoz odgovarajućih biblioteka

[9-25] deklaracija i inicijalizacija odgovarajućih konstanti

[29-39] deklaracija i inicijalizacija potrebnih ulaza i izlaza

[43-47] postavljanje frekvencije odgovarajućih PWM izlaza

[51-61] uspostavljanje Wi-Fi konekcije

[65-93] uspostavljanje povratnih poziva za pristigle poruke

[95-107] poziv odgovarajuće funkcije na osnovu teme

[109-116] funkcija koja periodično, uz pomoć DHT11 senzora, očitava temperaturu i vlažnost

[120-122] uspostavljanje konekcije sa MQTT brokerom

[124-132] supskripcija na teme

[134] deklaracija *Ticker*-a za periodično očitavanje temperature i vlažnosti

[136-137] deklaracija i inicijalizacija potrebnih varijabli

[139-151] neprekidno očitavanje stanja tastera i vrijednosti potencijometra

3 Korišteni hardverski resursi

3.1 picoETF

	<i>Komponenta</i>	<i>Opis</i>	<i>Količina</i>
1	RP2040	mikrokontroler	1
2	LED dioda	digitalni izlaz	3
3	RGB LED dioda	digitalni izlaz	1
4	taster	digitalni ulaz	1
5	potenciometar	ulazna komponenta	1
6	DHT11	ulazna komponenta	1
7	konektor	uvezivanje sistema	
8	USB A – USB Micro	napajanje i komunikacija	1

4 **Zaključak**

Zadaci u okviru Laboratorijske vježbe 7 su bili među najzanimljivijim dosad, a hardverska realizacija tokom laboratorijske vježbe je protekla uredno i bez ikakvih problema.

Prvi zadatak je predstavljao uvod u materiju laboratorijske vježbe, dok je treći zadatak bio nadogradnja drugog zadatka.

Za kraj, nova znanja stečena u okviru Laboratorijske vježbe 7 podrazumijevaju korištenje komunikacije u okviru ugradbenih sistema, tačnije, rad s MQTT protokolom, te s novim hardverskim komponentama, konkretno, DHT11 senzorom - senzorom za mjerenje temperature i vlažnosti.

5 Prilog

5.1 Zadatak 3: Izvorni kôd

```
001: from machine import Pin, ADC, PWM, Timer
002: from time import sleep
003: import network
004: import simple
005: import dht
006:
007: ----
008:
009: WIFI_SSID = "ssid"
010: WIFI_PASSWORD = "password"
011:
012: MQTT_SERVER = "broker.hivemq.com"
013: MQTT_CLIENT = "embedded"
014:
015: MQTT_TOPIC_LED1 = b"embedded/led1"
016: MQTT_TOPIC_LED2 = b"embedded/led2"
017: MQTT_TOPIC_LED3 = b"embedded/led3"
018:
019: MQTT_TOPIC_RGB_RED = b"embedded/rgbRed"
020: MQTT_TOPIC_RGB_GREEN = b"embedded/rgbGreen"
021: MQTT_TOPIC_RGB_BLUE = b"embedded/rgbBlue"
022:
023: MQTT_TOPIC_BUTTON = b"embedded/taster"
024: MQTT_TOPIC_POTENTIOMETER = b"embedded/potenciometar"
025: MQTT_TOPIC_SENSOR = b"embedded/dht11"
026:
027: ----
028:
029: led1 = Pin(4, Pin.OUT)
030: led2 = Pin(5, Pin.OUT)
031: led3 = PWM(Pin(6))
032:
033: rgbRed = PWM(Pin(14))
034: rgbGreen = PWM(Pin(12))
035: rgbBlue = PWM(Pin(13))
036:
037: button = Pin(3, Pin.IN)
038: potentiometer = ADC(Pin(26))
039: sensor = dht.DHT11(Pin(28))
040:
041: ----
042:
```

```

043: led3.freq(1000)
044:
045: rgbRed.freq(1000)
046: rgbGreen.freq(1000)
047: rgbBlue.freq(1000)
048:
049: ----
050:
051: # Uspostavljanje Wi-Fi konekcije
052: wifi = network.WLAN(network.STA_IF)
053: wifi.active(True)
054: wifi.connect(WIFI_SSID, WIFI_PASSWORD)
055:
056: while not wifi.isconnected():
057:     print("Uspostavljanje konekcije...")
058:     pass
059:
060: print("WLAN konekcija uspostavljena")
061: print("IP adresa:", wifi.ifconfig()[0])
062:
063: ----
064:
065: def messageArrivedLed1(topic, msg):
066:     print("Message arrived on topic: ", topic)
067:     print("Payload: ", msg)
068:     led1.value(int(float(msg)))
069:
070: def messageArrivedLed2(topic, msg):
071:     print("Message arrived on topic: ", topic)
072:     print("Payload: ", msg)
073:     led2.value(int(float(msg)))
074:
075: def messageArrivedLed3(topic, msg):
076:     print("Message arrived on topic: ", topic)
077:     print("Payload: ", msg)
078:     led3.duty_u16(int(float(msg)*65535))
079:
080: def messageArrivedRgbRed(topic, msg):
081:     print("Message arrived on topic: ", topic)
082:     print("Payload: ", msg)
083:     rgbRed.duty_u16(int(float(msg)*65535))
084:
085: def messageArrivedRgbGreen(topic, msg):
086:     print("Message arrived on topic: ", topic)
087:     print("Payload: ", msg)
088:     rgbGreen.duty_u16(int(float(msg)*65535))
089:
090: def messageArrivedRgbBlue(topic, msg):
091:     print("Message arrived on topic: ", topic)
092:     print("Payload: ", msg)
093:     rgbBlue.duty_u16(int(float(msg)*65535))
094:

```



```

095: def dispatcher(topic, msg):
096:     if topic == MQTT_TOPIC_LED1:
097:         messageArrivedLed1(topic, msg)
098:     elif topic == MQTT_TOPIC_LED2:
099:         messageArrivedLed2(topic, msg)
100:     elif topic == MQTT_TOPIC_LED3:
101:         messageArrivedLed3(topic, msg)
102:     elif topic == MQTT_TOPIC_RGB_RED:
103:         messageArrivedRgbRed(topic, msg)
104:     elif topic == MQTT_TOPIC_RGB_GREEN:
105:         messageArrivedRgbGreen(topic, msg)
106:     elif topic == MQTT_TOPIC_RGB_BLUE:
107:         messageArrivedRgbBlue(topic, msg)
108:
109: def sensorReport(pin):
110:     sensor.measure()
111:     temp = sensor.temperature()
112:     humidity = sensor.humidity()
113:
114:     publish = str("Temperature: " + str(temp) + "\nHumidity: " + str(humidity))
115:     buf = "{\\"Senzor\\": \n{}}".format(publish)
116:     client.publish(MQTT_TOPIC_SENSOR, buf)
117:
118: ----
119:
120: # Uspostavljanje konekcije sa MQTT brokerom
121: client = simple.MQTTClient(client_id=MQTT_CLIENT, server=MQTT_SERVER, port=1883)
122: client.connect()
123:
124: # Supskripcija na teme
125: client.set_callback(dispatcher)
126: client.subscribe(MQTT_TOPIC_LED1)
127: client.subscribe(MQTT_TOPIC_LED2)
128: client.subscribe(MQTT_TOPIC_LED3)
129: client.subscribe(MQTT_TOPIC_RGB_RED)
130: client.subscribe(MQTT_TOPIC_RGB_GREEN)
131: client.subscribe(MQTT_TOPIC_RGB_BLUE)
132: client.subscribe(MQTT_TOPIC_SENSOR)
133:
134: t = Timer(period=2000, callback=sensorReport, mode=Timer.PERIODIC)
135:
136: button_state = button.value()
137: potentiometer_value = potentiometer.read_u16()
138:

```

```
139: while True:
140:     if button.value() != button_state:
141:         button_state = button.value()
142:         buf = "{{\\\"Taster\\\": {}}}".format(button_state)
143:         client.publish(MQTT_TOPIC_BUTTON, buf)
144:
145:     if potentiometer.read_u16() != potentiometer_value:
146:         potentiometer_value = potentiometer.read_u16()
147:         buf = "{{\\\"Potenciometar\\\": {}}}".format(potentiometer_value)
148:         client.publish(MQTT_TOPIC_POTENTIOMETER, buf)
149:
150:     client.check_msg()
151:     sleep(0.1)
```