



Upravljanje uređajem pomoću mobilnog telefona (MQTT protokol)



Arduino uređaj kao klijent

Upišite u Arduino Uno R4 WiFi skeč nazvan:

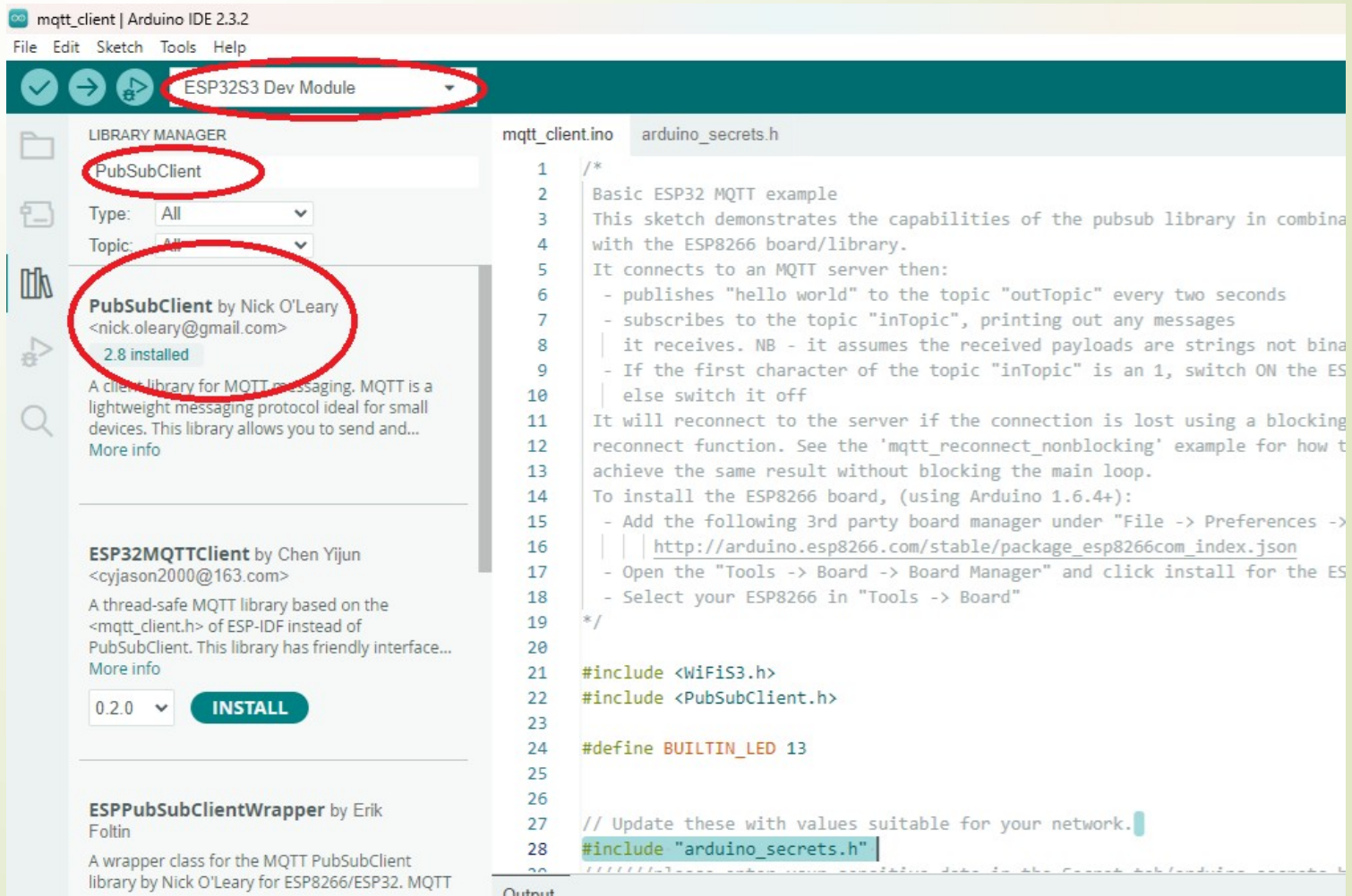
mqtt_cilent.ino

Skeč omogućuje da adruino uređaj funkcionise kao MQTT klijent.

Da bi se skeč uspješno kompajlirao i izvršavao potrebno je instalirati biblioteku: **PubSubClient**

Arduino uređaj kao klijent

Instaliranje PubSubClient biblioteke:



The screenshot shows the Arduino IDE 2.3.2 interface. The 'Tools' menu is open, and 'ESP32S3 Dev Module' is selected. The 'LIBRARY MANAGER' tab is active, displaying a search for 'PubSubClient'. The library by Nick O'Leary is highlighted, showing it is version 2.8 and already installed. Below it, the 'ESP32MQTTClient' library by Chen Yijun is shown with version 0.2.0 and an 'INSTALL' button. The 'ESP8266PubSubClientWrapper' library by Erik Foltin is also visible. The main editor window shows the 'mqtt_client.ino' file with the following code:

```
1  /*
2  Basic ESP32 MQTT example
3  This sketch demonstrates the capabilities of the pubsub library in combina
4  with the ESP8266 board/library.
5  It connects to an MQTT server then:
6  - publishes "hello world" to the topic "outTopic" every two seconds
7  - subscribes to the topic "inTopic", printing out any messages
8  it receives. NB - it assumes the received payloads are strings not bina
9  - If the first character of the topic "inTopic" is an 1, switch ON the ES
10 else switch it off
11 It will reconnect to the server if the connection is lost using a blocking
12 reconnect function. See the 'mqtt_reconnect_nonblocking' example for how t
13 achieve the same result without blocking the main loop.
14 To install the ESP8266 board, (using Arduino 1.6.4+):
15 - Add the following 3rd party board manager under "File -> Preferences ->
16 | | http://arduino.esp8266.com/stable/package_esp8266com_index.json
17 - Open the "Tools -> Board -> Board Manager" and click install for the ES
18 - Select your ESP8266 in "Tools -> Board"
19 */
20
21 #include <WiFiS3.h>
22 #include <PubSubClient.h>
23
24 #define BUILTIN_LED 13
25
26
27 // Update these with values suitable for your network.
28 #include "arduino_secrets.h"
```

Arduino uređaj kao klijent – uloga **loop()** funkcije

U biblioteci **PubSubClient**, funkcija **client.loop()** ključna je za održavanje veze s **MQTT brokerom** i rukovanje dolaznim porukama.

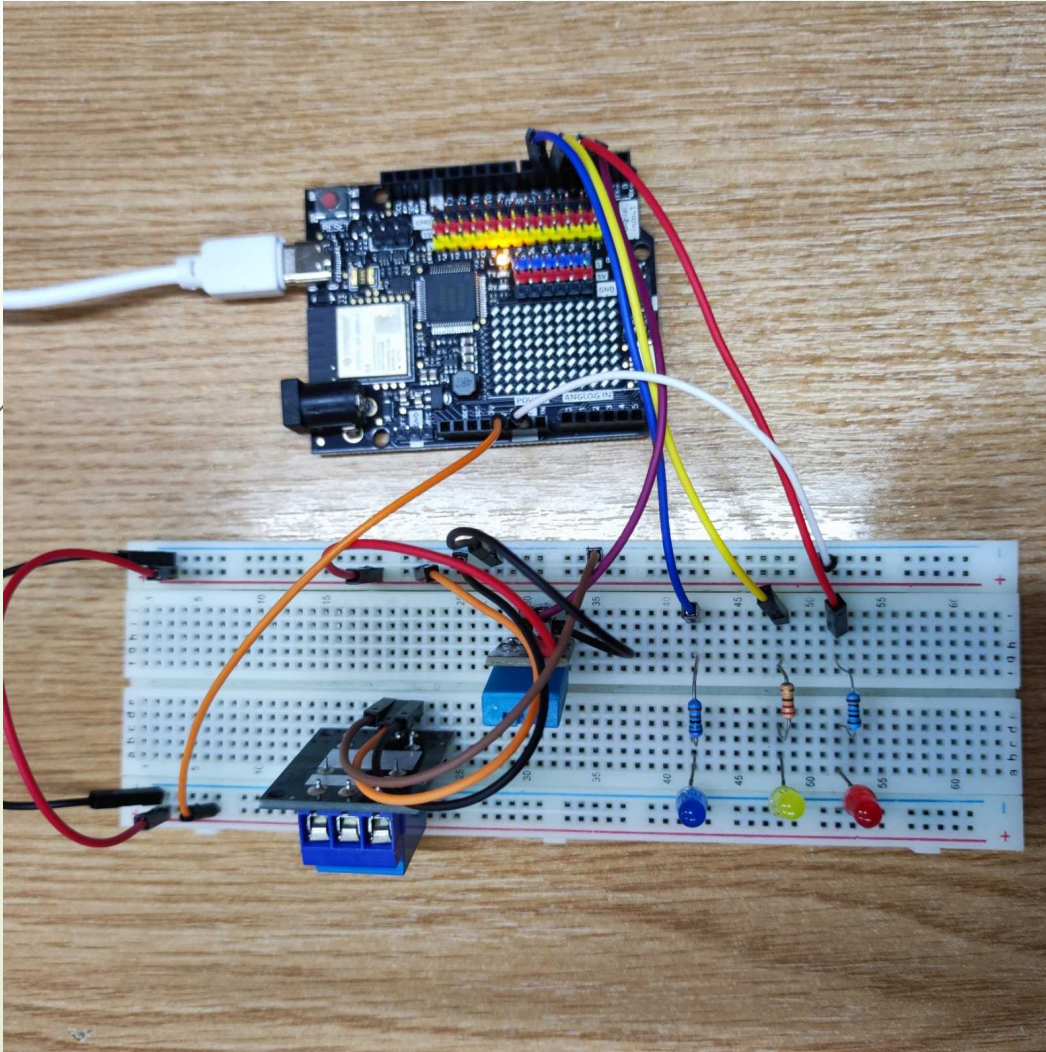
Kada se pretplatite na teme ili objavite poruke koristeći MQTT (Message Queuing Telemetry Transport), mikrokontrolerska ploča mora stalno komunicirati s MQTT brokerom kako bi primala poruke ili slala ažuriranja. Za ovu kontinuiranu komunikaciju odgovorna je funkcija **client.loop()**.

Evo što radi:

- **Održavanje veze:** osigurava da se veza s MQTT brokerom održava živom. MQTT se temelji na trajnoj TCP vezi, a klijent mora slati periodične kontrolne pakete kako bi održao ovu vezu.
- **Rukovanje dolaznim porukama:** Provjerava sve dolazne poruke iz pretplaćenih tema. Kada stignu poruke, funkcija **loop()** ih obrađuje i pokreće odgovarajuće funkcije povratnog poziva koje ste možda definisali u svom kodu.
- **Obrada odlaznih poruka:** Ako postoje poruke u redu čekanja za objavljivanje, funkcija **loop()** se brine za njihovo slanje MQTT brokeru.
- **Keep-Alive Mechanism:** Takođe upravlja mehanizmom za održavanje aktivnosti MQTT protokola, gdje povremeno šalje ping zahtjeve brokeru kako bi se osiguralo da je veza još uvijek aktivna.

Sve u svemu, redovno pozivanje **client.loop()**, obično unutar glavne petlje vašeg programa, bitno je za nesmetan rad MQTT komunikacije. Održava komunikaciju živom, obrađuje dolazne poruke i osigurava da vaš uređaj i dalje reaguje na MQTT mrežu.

Priprema za vježbu



Mikrokontrolerski dizajn

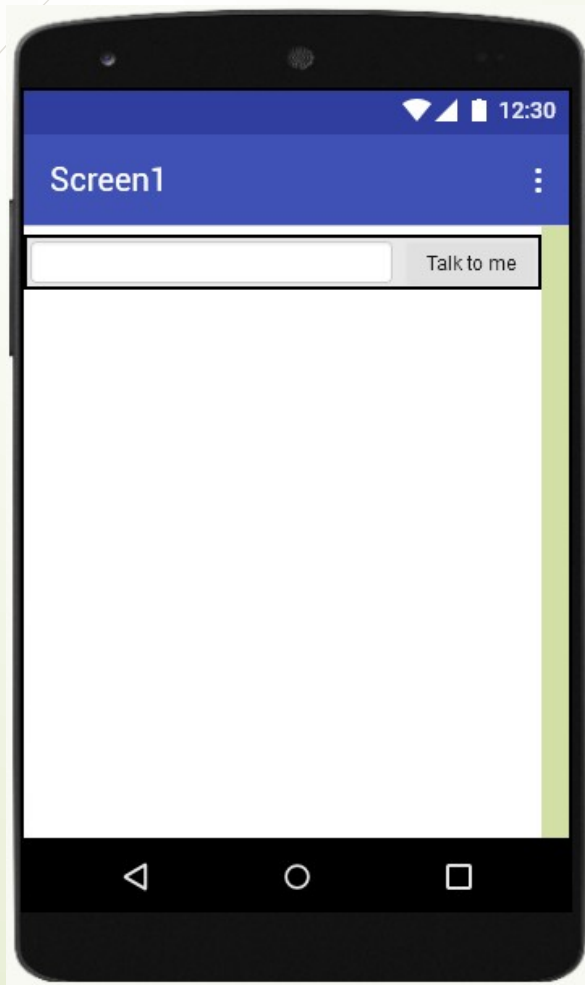
Crvena LED je preko otpornika 220► spojena na pin 6.
Žuta LED je preko otpornika 220► spojena na pin 5.
Plava LED je preko otpornika 220► spojena na pin 3.

Signal Relay-a je spojen na pin A0. Uz to VCC je spojeno na 5V, a GND na GND.

Signal DHT11 senzora povezan je na pin 2. VCC je spojeno na 5V, a GND na GND

Link za preuzimanje MQTT extenzije:
<https://ullisroboterseite.de/android-AI2-PahoMQTT-en.html#down>

Priprema za vježbu



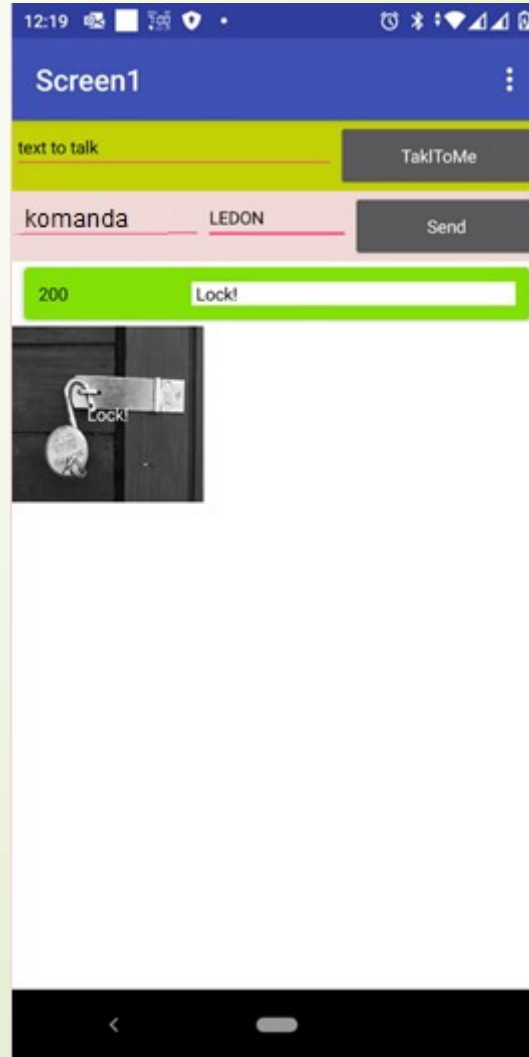
Zajednički ćemo kreirati aplikaciju čiji je izgled prikazan na slici.

Priprema za vježbu



Na osnovnu aplikaciju, iz predhodnog slajda, dodat ćemo i dio koji će omogućiti povezivanje sa našim mikrokontrolerskim dizajnom.

Vježba 1



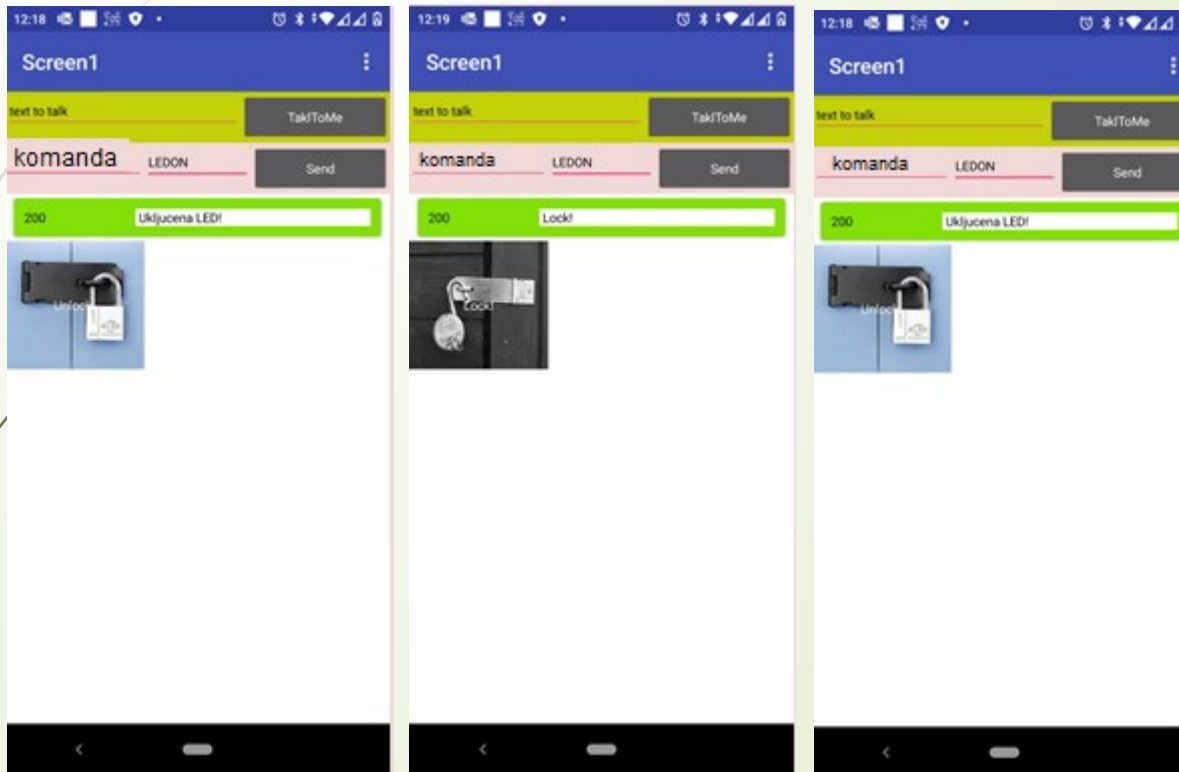
Dopuniti aplikaciju mogućnošću otvaranja brave (odnosno upravljanja rele-om).

Kada se pritisne komandno dugme, sa slikom zaključanog katanca, zatvara se NO kontakt rele-a u mikrokontrolerskom dizajnu, a slika na komandnom dugmetu mobilnog telefona se mijenja u sliku sa otvorenim katancem.

Kada se pritisne komandno dugme, sa slikom otključanog katanca, otvara se NO kontakt rele-a u mikrokontrolerskom dizajnu, a slika na komandnom dugmetu mobilnog telefona se mijenja u sliku sa zatvorenim katancem.

(1 – 0.5 bodova)

Vježba 2



Dopuniti rad sa rele-om, na sljedeći način:

Ukoliko se po zatvaranju kontakta rele-a, u roku od 5 sekundi ne zada komanda za otvaranje kontakta, kontakt se otvara, a komandno dugme ponovo dobija sliku zaključanog katanca.

(1 – 0.5 bodova)

Vježba 3



Dopuniti aplikaciju mogućnošću upravljanja intezitetom sjaja crvene, žute i plave LED, koje su povezane na pinove 3,5 i 6.

Upotrijebiti *Slider* komponente.

(2 – 1 – 0.5 bodova)

Vježba 4



Dopuniti aplikaciju mogućnošću mogućnošću prikazivanja, informacija o temperaturi i vlazi.

Informaciju o temperaturi i vlazi obezbjeđuje senzor temperature i vlage u mikrokontrolerskom dizajnu.

Informacije prikazati kao brožani ispis na komandnom dugmetu, sa odgovarajućom sugestivnom slikom u pozadini, kao i pomoću slider komponenti.

Informacije obezbijediti bez pritiska na komandno dugme i bez bilo kakve manuelne akcije na telefonu

(2 – 1 - 0.5 bodova)

U ovom primjeru koisti se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka](#) (GitHub).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit_Sensor** biblioteka, ona se može preuzeti sa [ovog linka](#) (GitHub).

Vježba 5



Dopuniti aplikaciju sljedećim:

Po pritisku komandnog dugmeta za prikaz temperature, istu prikazati u Farenheit-ima, u trajanju 5 sekundi nakon čega se ponovo vraća ispis u stепенima Celsijus-a.

(1 – 0.5 bodova)

U ovom primjeru koisti se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka](#) (GitHub).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit_Sensor** biblioteka, ona se može preuzeti sa [ovog linka](#) (GitHub).

Vježba 6



Upravlјati LED glasovnim komandama:

Za uključenje LED: "turn on red(blue/yellow)"

Za isključenje LED: "turn off red(blue/yellow)"

Prilikom uključenja LED, ista treba sijati
intezitetom zadanim pomoću slajdera

(1 – 0.5 bodova)

Vježba 7



Upravlјati koračnim motorom glasovnim komandama:

LEFT, RIGHT, FASTER, SLOWER, STOP

Uloga glasovnih dugmadi je jasna iz samog naziva.

Na ekranu mobilnog telefona prikazivati brzinu (kor/s) i smjer okretanja koračnog motora.

(2 – 1 – 0.5 bodova)

Vježba 8



Kada se mobilni telefon primkne uhu, na LCD-u uređaja, u njegovom gornjem redu, ispisati poruku: „Telefon pri uhu!“.

Kada se telefon odmakne od uha, na LCD-u uređaja, u njegovom gornjem redu, ispisati poruku: „Telefon odmaknut“.

(2 - 1 – 0.5 bodova)

U ovom primjeru koristi se I²C LCD. Biblioteka za rad sa njime može se preuzeti sa linka:

https://github.com/johnrickman/LiquidCrystal_I2C