

Univerzitet u Sarajevu
Elektrotehnički fakultet
Ugradbeni sistemi (RI/TK) 2024/25

Dokumentacija

Sistem za detekciju i gašenje požara

Studenti:
Muhamed Džafić
Ismail Mujanović

Sarajevo, juli 2025.

Sadržaj

1	Uvod	2
2	Arhitektura sistema	2
2.1	Fizičko povezivanje komponenti	2
3	Umrežavanje i MQTT protokol	2
3.1	Povezivanje na WiFi mrežu	2
3.2	Povezivanje na MQTT broker	3
3.3	Obrada MQTT poruka	3
4	Ulagani podaci – senzori	3
5	Upravljanje relejem	4
6	Vizuelni prikaz – TFT ekran	4
7	Logika rada	4
8	Potencijalne primjene	5
9	Zaključak	5

1 Uvod

Dokumentacija detaljno opisuje tehničku implementaciju sistema za detekciju i automatsko (ili ručno) gašenje požara korištenjem mikrokontrolera picoETF, senzora i MQTT komunikacije. Sistem je razvijen u MicroPython programskom jeziku i koristi TFT displej za lokalni prikaz, te WiFi mrežu za daljinski nadzor i upravljanje. U nastavku se po koracima objašnjavaju ključni dijelovi implementacije, uz referenciranje na konkretnе funkcije i segmente iz izvornog koda projekta.

2 Arhitektura sistema

Sistem se zasniva na picoETF mikrokontroleru koji upravlja senzorima (senzor za plamen, temperaturni senzor), relejnim aktuatorom za pumpu, kao i TFT displejom za lokalni prikaz stanja. Komunikacija sa udaljenim korisnikom obezbijeđena je putem WiFi konekcije i MQTT protokola. Sistem omogućava dva režima rada: automatski, gdje mikrokontroler sam upravlja pumpom u slučaju požara, i ručni, gdje korisnik putem MQTT poruka upravlja radom pumpe.

2.1 Fizičko povezivanje komponenti

Komponenta	Pin na picoETF
LM35	GP28 (ADC)
Senzor za plamen	GP14 (digitalni ulaz)
Relej	GP13 (digitalni izlaz)
TFT SPI ekran	GP18 (CLK), GP19 (MOSI), GP16 (MISO), GP17 (CS), GP15 (DC), GP20 (RST)

3 Umrežavanje i MQTT protokol

Sistem koristi WiFi konekciju za povezivanje na Internet i ostvarenje dvosmjerne komunikacije sa udaljenim korisnikom putem MQTT (Message Queuing Telemetry Transport) protokola.

3.1 Povezivanje na WiFi mrežu

Povezivanje na WiFi mrežu vrši se pomoću MicroPython `network` modula i funkcije:

```
def connect_wifi(ssid, password):
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        wlan.connect(ssid, password)
        while not wlan.isconnected():
            pass
    print("Povezano:", wlan.ifconfig())
```

Nakon uspješnog povezivanja, uređaj dobija IP adresu što se potvrđuje kroz `wlan.ifconfig()` i spreman je za razmjenu podataka. Korišten je `STA_IF` (station mode) interfejs jer uređaj funkcioniše kao klijent u mreži.

3.2 Povezivanje na MQTT broker

Za MQTT komunikaciju koristi se `MQTTClient` klasa iz `umqtt.simple` biblioteke. Broker koji se koristi u projektu je javni broker "broker.hivemq.com". Uredaj se prijavljuje sa ID-om "pico_plamen" i pretplaćuje na temu:

- picoETF/pumpa – kontrola rada pumpe (ulazna tema)

Takođe, sistem objavljuje podatke na sljedeće teme:

- picoETF/temperatura – trenutačna temperatura (izlazna tema)
- picoETF/status – status plamena (izlazna tema)

3.3 Obrada MQTT poruka

Upravljanje režimom rada pumpe se vrši u funkciji `sub_cb(topic, msg)`, koja se poziva kada stigne nova poruka. Funkcija reaguje samo na poruke sa teme picoETF/pumpa, i to na sljedeći način:

```
def sub_cb(topic, msg):
    global manual_mode
    if topic == TOPIC_PUMPA:
        if msg == b"auto":
            manual_mode = False
        else:
            manual_mode = True
            if msg == b"1":
                relay.value(0)
            elif msg == b"0":
                relay.value(1)
```

Na osnovu pristigle poruke:

- "auto" → sistem se prebacuje u automatski režim, u kojem sam detektuje i reaguje na plamen.
- "1" → u ručnom režimu, uključuje se relej i aktivira pumpa.
- "0" → u ručnom režimu, relej se isključuje i pumpa se gasi.

Relej je povezan na `Pin(13)`, i njegova vrijednost (0 = uključen, 1 = isključen) kontroliše fizičko stanje pumpe.

4 Ulagani podaci – senzori

Dva senzora obezbjeđuju ulazne podatke:

- **Temperaturni senzor (LM35)**: povezan na analogni `Pin(28)`. Čitanje se vrši funkcijom `adc.read_u16()`, a temperatura se konvertuje u °C pomoću izraza:

```
temp_c = raw * 330 / 65535
```

- **Senzor za plamen**: digitalni ulaz `Pin(14)` vraća 0 kada je plamen detektovan. Predstavlja osnovni uslov za aktivaciju pumpnog sistema.

5 Upravljanje relejem

Relej je vezan za Pin(13) i inicijalno se postavlja u stanje isključenosti (`relay.value(1)`). U automatskom režimu se uključuje prilikom detekcije plamena, dok se u ručnom režimu upravlja direktnim MQTT komandama.

6 Vizuelni prikaz – TFT ekran

Sistem koristi ILI9341 TFT displej dimenzija 320×240 piksela, povezan SPI interfejsom. Za rad s ekranom koriste se sljedeće biblioteke:

- ili934xnew – za inicijalizaciju i osnovne funkcije TFT ekrana
- glcdfont, tt14, tt24, tt32 – različiti fontovi za ispis na ekranu

Prikaz temperature i status plamena se ažurira u svakoj iteraciji glavne petlje:

```
display.print(f"Temperatura: {round(temp_c, 2)} C")
```

Ako je plamen detektovan (`flame_status == 0`), ispisuje se crvenim slovima:

```
display.set_pos(10, 40)
display.set_color(color565(255, 0, 0), color565(0, 0, 0))
display.print("PLAMEN DETEKTOVAN!")
```

Ako plamen nije detektovan, ispisuje se neutralna poruka bijelom bojom:

```
display.set_pos(10, 40)
display.set_color(color565(255, 255, 255), color565(0, 0, 0))
display.print("Nema plamena")
```

U slučaju detekcije plamena, iscrtava se i simbolična vatrica pomoću funkcije `nacrtaj_veliku_vatru(x0, y0)`, čija je implementacija data u nastavku:

```
def nacrtaj_veliku_vatru(x0, y0):
    display.fill_rectangle(x0+10, y0+60, 60, 20, color565(255, 0, 0))
    display.fill_rectangle(x0+20, y0+40, 40, 20, color565(255, 69, 0))
    display.fill_rectangle(x0+30, y0+20, 20, 20, color565(255, 215, 0))
    display.fill_rectangle(x0+38, y0+10, 4, 10, color565(255, 255, 0))
```

7 Logika rada

Centralna logika implementirana je unutar beskonačne `while True` petlje. U svakom ciklusu obavljaju se sljedeći koraci:

1. Očitavanje analogne vrijednosti i konverzija u temperaturu.
2. Provjera statusa plamena (`do_flame.value()`).
3. Slanje MQTT poruka (temperature i statusa).
4. Osvježavanje ekrana prikazom trenutnog stanja.

5. U zavisnosti od `manual_mode` i statusa plamena – uključivanje ili isključivanje releja.
6. Provjera pristiglih MQTT poruka funkcijom `mqtt_client.check_msg()`.
7. Pauza od 2 sekunde (`time.sleep(2)`) radi smanjenja učestalosti ciklusa.

Kôd za odlučivanje o statusu izgleda ovako:

```
if flame_status == 0:
    display.print("PLAMEN DETEKTOVAN!")
    nacrtaj_veliku_vatru(...)
    mqtt_client.publish(TOPIC_STATUS, b"PLAMEN DETEKTOVAN!")
    if not manual_mode:
        relay.value(0)
else:
    display.print("Nema plamena")
    mqtt_client.publish(TOPIC_STATUS, b"Nema plamena")
    if not manual_mode:
        relay.value(1)
```

8 Potencijalne primjene

Sistem demonstrira mogućnost kreiranja pametnih autonomnih sistema koji mogu reagovati u realnom vremenu na opasnosti kao što su požari. Moguće primjene uključuju:

- Pametne kuće (nadzor i gašenje u kuhinji)
- Industrijske objekte (automatizovana sigurnosna rješenja)
- IoT platforme za praćenje okoline
- Integraciju sa sistemima za pametne gradove

9 Zaključak

Projekat predstavlja primjenu ugradbenih sistema u oblasti zaštite i nadzora. Kombinacija lokalne obrade podataka, vizuelnog prikaza i bežične komunikacije omogućava pravovremeno detektovanje i reakciju na požarne situacije. Sistem može raditi samostalno u realnom vremenu, kao i biti integriran u širu IoT infrastrukturu putem MQTT protokola.