

Sobna vremenska postaja

Tomaž Čuk

UVOD

Za projekt sem naredil sobno vremensko postajo, ki prikazuje trenutno temperaturo, vlažnost in zračni tlak s pomočjo senzorja. Informacije bom prikazal na ekrančku, poleg tega pa se prenesejo v MQTT server in si jih lahko ogledamo v aplikaciji Home Assistant.

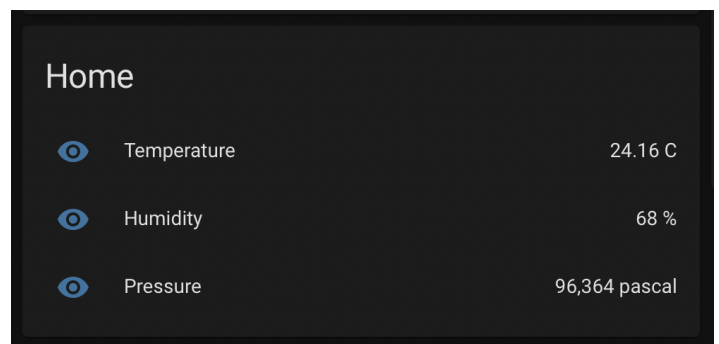
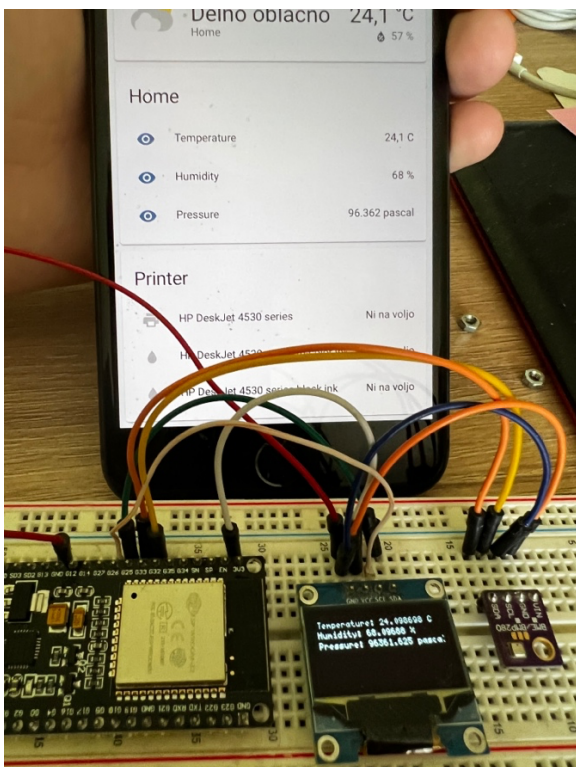
Komponente:

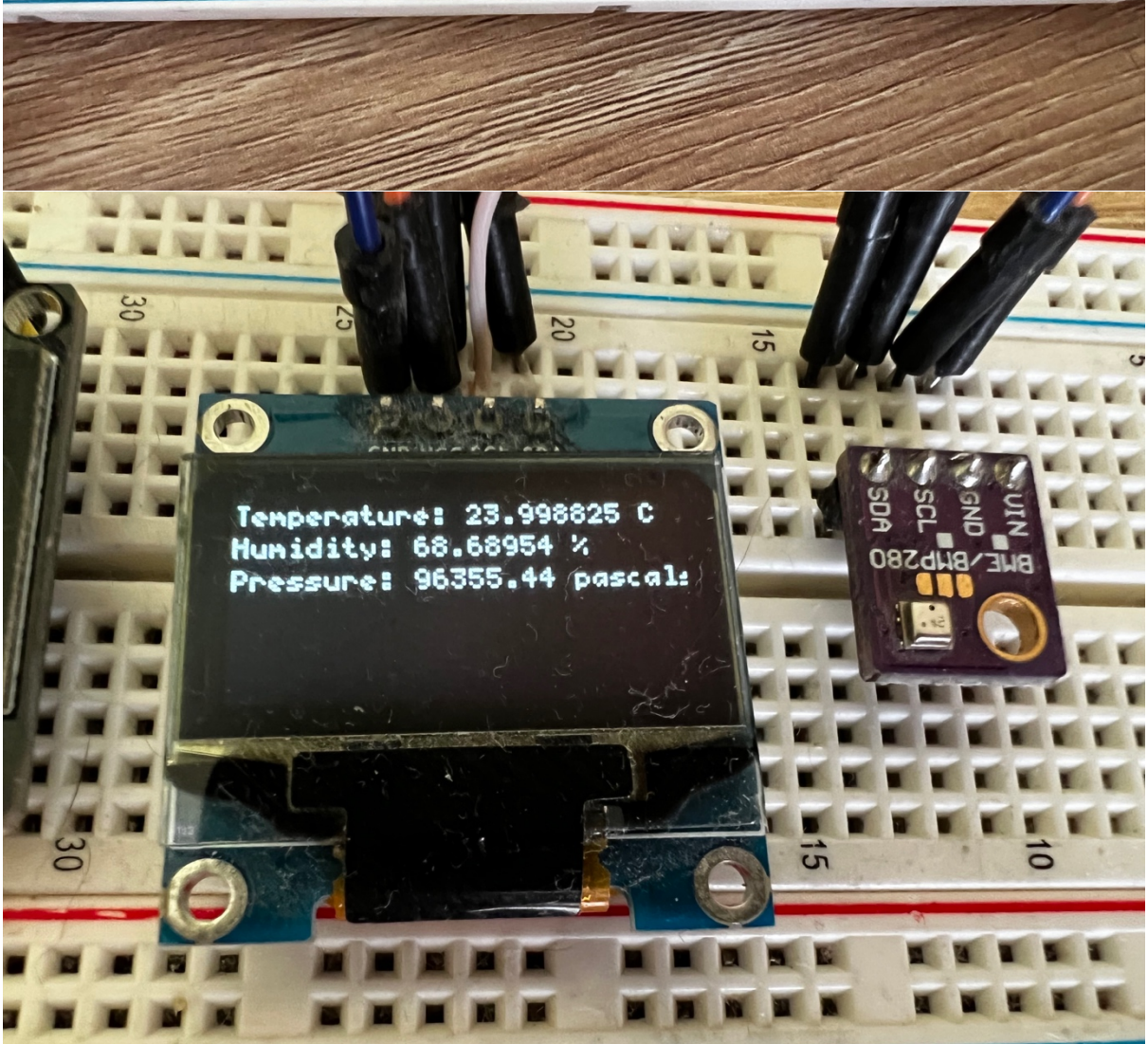
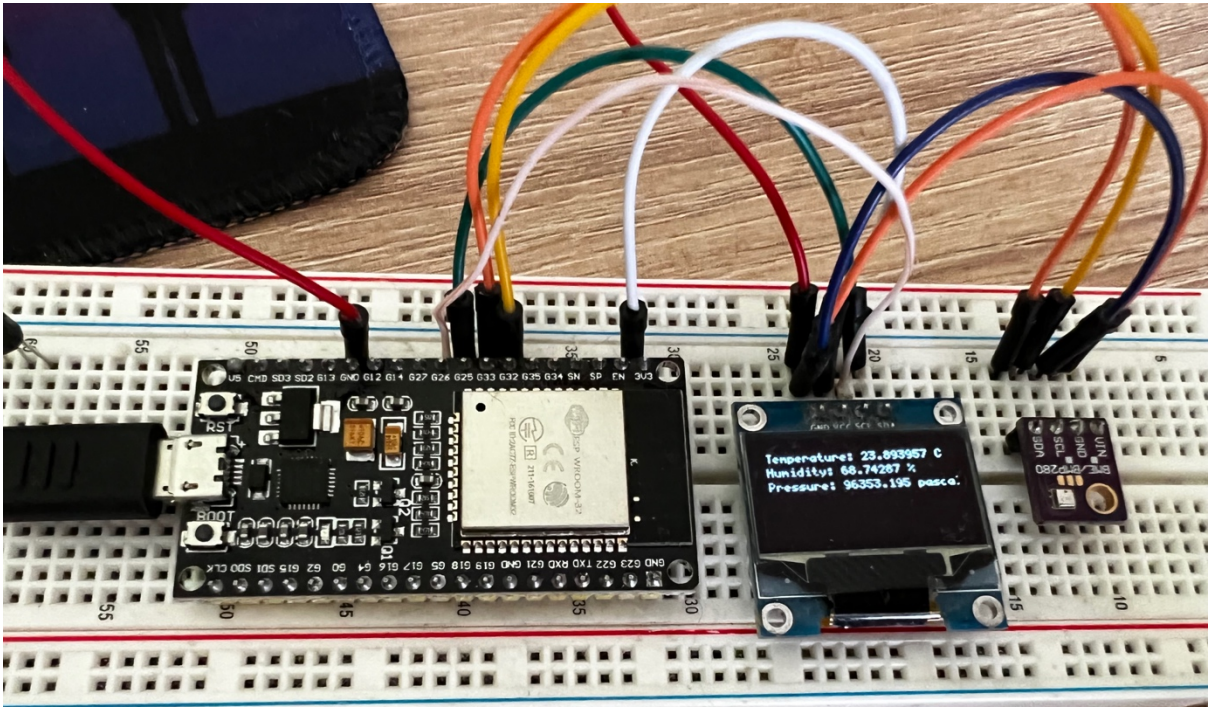
- BME280 senzor za temperaturo, vlažnost in zračni tlak
- Ekran SDD1306
- ESP32 z vgrajenim Wifi modulom
- Raspberry Pi, ki poganja MQTT in Home Assistant server

JEDRO

Odločil sem se, da poskusim programiranje mikrokontrolerov v programskem jeziku Rust. Izbral sem ESP32, saj ima že vgrajen Wifi modul in kar dobro podporo za razvijanje v Rust. Za ekran sem izbral SDD1306, saj sem ga imel že doma od prejšnjih projektov in je bolj služil za debugiranje. Za senzor sem izbral BME280, saj sem ga prav tako že imel doma in je dovolj natančen. Potreboval sem še Raspberry Pi, na katerega sem naložil MQTT server, ki je sprejemal informacije od ESP32, in Home Assistant, ki je bral informacije iz MQTT serverja ter jih lahko vidimo na spletu ali pa preko mobilne aplikacije. S Home Assistant se da veliko stvari avtomatizirati, tako da bi lahko v prihodnosti dodal da se ob prenizki ali previsoki vlažnosti avtomatično vklopi vlažilnik zraka.

Za komunikacijo med mikrokontrolerjem in ekrančkom ter senzorjem sem uporabil I2C, saj potrebuje le 2 žici. Program je sestavljen tako, da se ob zagonu vklopi Wifi in poveže na dostopno točko, nato se vzpostavi povezava z ekranom in senzorjem, opravi se meritev in se prikaže na zaslonu, prav tako pa jo pošlje do MQTT serverja in nazadnje preide v globoko spanje za 1 minuto.





TEŽAVE

Največ težav mi je povzročala sama vzpostavitev delovnega okolja, saj je potrebno namestiti veliko različnih orodij kot na primer ESP-IDF, Rust toolchain za ESP, LLVM/Clang, orodja za flash in monitor.

Dokumentacija je bila slaba, zato sem se bolj kot ne zanašal nad »example« projektih drugih ljudi (najbolj mi je pomagal ta projekt <https://github.com/ivmarkov/rust-esp32-std-demo>)

Bilo je nekaj problemov s flashanjem programa, saj je bila moja verzija mikrokontrolerja malo drugačna od uradne, tako da sem po urah in urah iskanja po internetu ugotovil, da moram držati gumb BOOT in vmes pritisniti gumb RESET in potem še vedno držati BOOT, dokler se flashanje ne začne.

ZAKLJUČEK

Kljub težavam je bilo vseeno zanimivo raziskati, kako zgleda razvijanje mikrokontrolerjev v jeziku Rust, ki ima kar nekaj prednosti na primer sestavljanje različnih knjižnic skupaj po principu vmesnik/razred, ima pa seveda še veliko slabosti, ker tako razvijanje ni razširjeno oz. popularno je podpora in dokumentacija precej slabša v primerjavi z razvijanjem v Cju.