

# Praćenje nivoa kiseonika u krvi i vrednosti pulsa preko MQTT

Dokumentacija

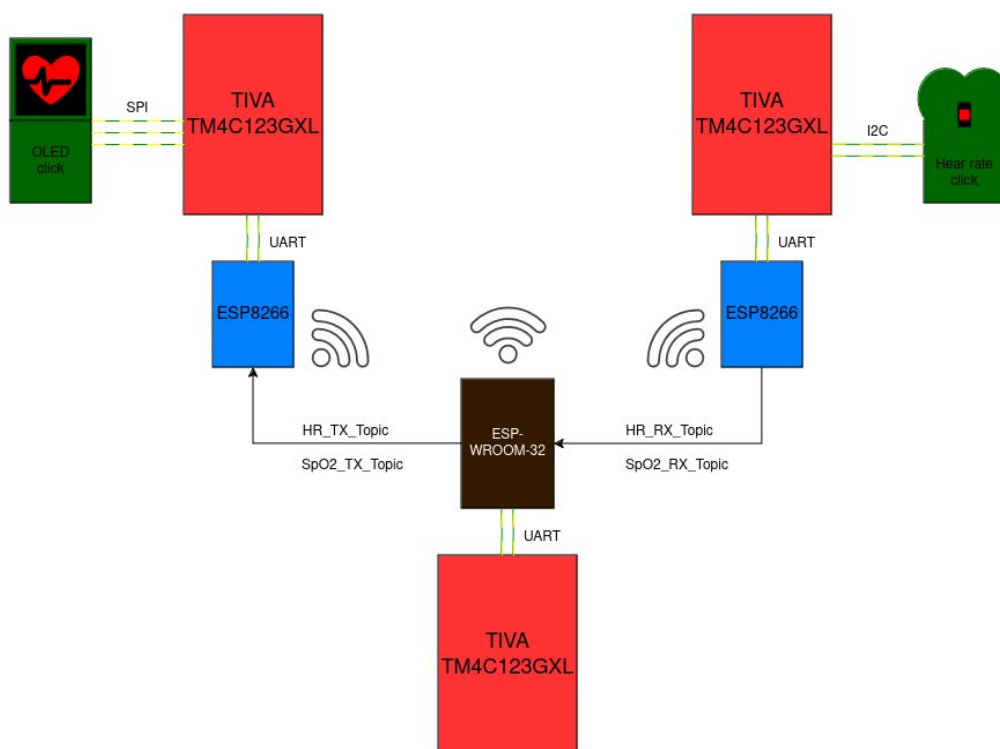
Aleksandar Živković rm 17/23

# Sadržaj

Sadržaj.....	2
Opis sistema.....	3
Broker.....	4
ESP AT drajver.....	4
Opis.....	5
Implementacija.....	6
Parsiranje poruka.....	6
Automatizacija MQTT konfiguraacije.....	7
Slanje i primanje poruka.....	7
Oximetar.....	7
Implementacija.....	8
MeasureTask.....	8
UpdateTask.....	9
OLED Modul.....	10
Implementacija.....	11

## Opis sistema

Cilj ovog projekta bio je razvoj prototipa IoT uređaja za merenje vrednosti kiseonika u krvi (SpO2) i pulsa (Heart Rate), kao i njihov prikaz na OLED ekranu putem MQTT protokola. Sistem se sastoji od tri nezavisne komponente: Broker, Oximeter čitač i Upravljač OLED modulom, o kojima će kasnije biti više rečeno. Sve komponente su implementirane na Tiva C pločici TM4C123GXL sa FreeRTOS, pri čemu je povezivanje na MQTT mrežu ostvareno putem ESP modula koji koristi ESP AT firmware. Dijagram sistema je prikazan na slici 1.



Slika 1: Blok šema sistema

# Broker

MQTT (Message Queuing Telemetry Transport) je standardizovani protokol za komunikaciju IoT uređaja. Dizajniran je kao vrlo jednostavan i lagan protokol za prenos poruka, idealan za uređaje sa ograničenim resursima koji se povezuju preko mreža sa visokim kašnjenjem i niskim propusnim opsegom.

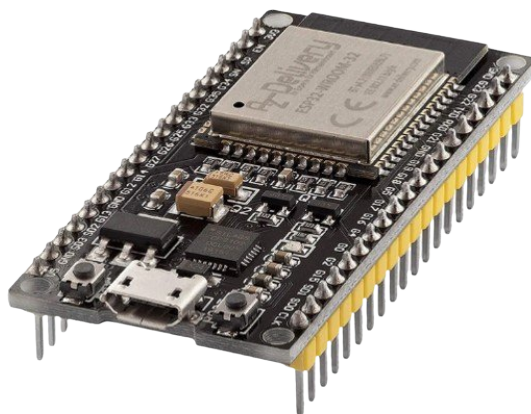
MQTT se zasniva na publish/subscribe mehanizmu, gde klijenti šalju (publish) poruke na određene kanale (topics), dok drugi klijenti osluškiju (subscribe) kanale kako bi primali te poruke. Centralni element sistema je MQTT broker, koji prima poruke od pošiljalaca, filtrira ih i prosleđuje klijentima koji osluškiju date kanale.

Broker komponenta predstavlja koordinator sistema. Njena uloga je da pročitane vrednosti dobijene od Oximetar čitača šalje OLED modulu za prikaz. Komunikacija sistema putem MQTT protokola realizovana je preko četiri kanala: dva za komunikaciju između Oximetra i Brokera, i dva za komunikaciju između Brokera i OLED modula. Ovi kanali su dalje podeljeni u dve grupe: jedna putanja za prenos SpO2 vrednosti i druga za prenos Heart Rate vrednosti, tj. Oximetar -> Broker -> OLED.

Kao host za MQTT korišćen je Eclipse MQTT.

## ESP AT drajver

Sistem je implementiran sa ESP32-WROOM-02 (slika 2) i ESP8266 (slika 3) modulima.



Slika 2: ESP32-WROOM-02 modul



Slika 3: ESP8266 modul

## Opis

ESP AT firmware je softver razvijen za ESP module, kao što su ESP8266 i ESP32, koji omogućava jednostavno povezivanje i komunikaciju putem AT komandi. Ovaj firmware omogućava uređajima da se povezuju na Wi-Fi mreže, uspostavljaju TCP/UDP konekcije i komuniciraju sa serverima ili drugim uređajima preko interneta. Koristeći AT komande, mogu se kontrolisati različite funkcionalnosti ESP modula bez potrebe za programiranjem samog mikrokontrolera.

U set ponuđenih AT komandi spada i funkcije za rad sa MQTT protokolom. Povezivanje ESP modula sa Tiva pločicama postignuto je putem UART serijskog protokola. Da bi se neka komanda izvršila, potrebno ju je poslati na ESP modul preko UART-a sa odgovarajućim parametrima, nakon čega bi se dobio odgovarajući odgovor o statusu poslate poruke.

Na primer, publish komanda izgleda ovako:

- Tiva pločica: `"AT+MQTTPUB=0,"topic","data",1,0"`
- ESP modul: `"OK"`

U ovom primeru, Tiva pločica šalje zahtev za publish "data" na "topic" kanal, a ESP modul uspešno izvršava komandu i odgovara sa "OK". Postoje i drugi tipovi poruka koje ESP može da pošalje, kao što su "ERROR", "SEND OK", "SEND FAIL" ili dobijena poruka nakon publishovanja.

Primer poruke dobijene od strane ESP modula koji osluškuje "test" kanal:

- ESP modul: `"+MQTTSUBRECV:0,"test",6,"poruka""`

U ovom slučaju, ESP modul prima poruku "poruka" na "test" kanal. Prijem ovakvih poruka omogućava Tiva pločici da reaguje na podatke koje prima putem MQTT protokola. Kao što se može videti, pri radu sa ESP AT firmwareom potrebno je osmisliti adekvatan način parsiranja dobijenih poruka, što uključuje i razvoj ESP AT drajvera.

# Implementacija

Glavni zadatak ove implementacije bio je pokrivanje parsiranja dobijenih poruka od strane ESP modula, automatizacija MQTT konfiguracije prilikom pokretanja sistema, kao i olakšavanje slanja i primanja poruka.

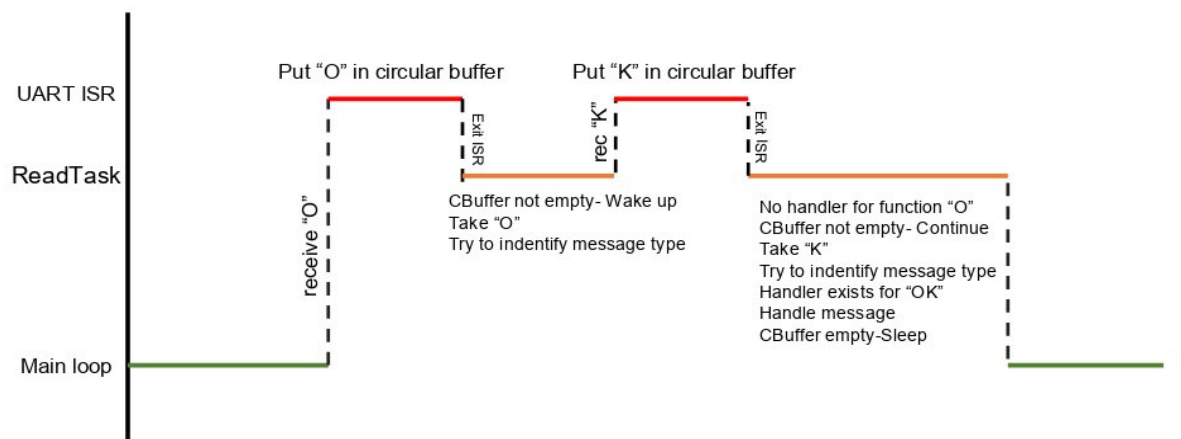
## Parsiranje poruka

Parsiranje je urađeno u vidu jednog FreeRTOS taska nazvanim, *ReadTask*.

Kako se preko UART-a šalje karakter po karakter, kao vid optimizacije, parsiranje poruka je urađeno pomoću Trie datastrukture. Trie datastruktura predstavlja stablo u kojem čvorovi predstavljaju karaktere, a listovi pokazuju na funkcije koje odgovaraju datoj komandi, što se postiže kretanjem od korena do lista stabla. Ovakvim pristupom dobijamo brz odziv na prosleđene komande uz minimalnu potrošnju memorijskih resursa.

Parsiranje poruka dobijenih od ESP modula može biti vrlo opterećujuće za sistem, stoga je implementiran mehanizam odloženog odziva na ISR rutinu. Prilikom prijema karaktera putem UART protokola, karakter se ubacuje u kružni bafer, koji se kasnije obrađuje u *ReadTasku*, tasku sa najvišim prioritetom u sistemu.

*ReadTask* čita dobijene karaktere iz bafera i propušta ih kroz Trie strukturu, pomoću koje se određuje kontekst dobijene poruke i odgovarajući handler. Ukoliko je poruka tipa publish, u handleru za publish, podaci poruke se ubacuju u Queue koja odgovara datom kanalu kojeg ostale komponente sistema osluškiju čekajući na poruku. Ako je poruka tipa OK, tada se OK semafor sistema inkrementira, služeći kao potvrda uspešnosti za operacije koje čekaju potvrdni odgovor. Na slici 4 se može videti tok rada *ReadTaska* u obradi OK poruke.



Slika 4: Šema obrade "OK" poruke

## Automatizacija MQTT konfiguracije

Konfiguracija urađena je u vidu jednog FreeRTOS taska nazvanim, *ESPSetUpTask*.

*ESPSetUpTask* automatizuje konfiguraciju ESP modula za MQTT protokol putem serije AT komandi. Prvo se izvršava resetiranje modula (AT+RST), zatim se postavlja režim rada WiFi modula (AT+CWMODE=1) i vrši povezivanje na WiFi mrežu sa specificiranim SSID i lozinkom (AT+CWJAP="<SSID>","<password>"). Nakon uspešnog povezivanja, modul se konfiguriše za MQTT komunikaciju, uključujući podešavanje postavki modula (AT+MQTTUSERCFG) i uspostavljanje veze sa MQTT hostom na određenoj IP adresi i portu (AT+MQTTCONN). Dalje, modul se postavlja da osluškuje određene MQTT kanale za prijem podataka, kao što su merenja SpO2 i Heart Rate (AT+MQTTSUB).

Sve navedene komande se šalju preko UART protokola, a svaka komanda se proverava da bi se osiguralo uspešno izvršavanje pre prelaska na sledeću. Nakon završetka inicijalizacije, omogućava se slanje poruka.

## Slanje i primanje poruka

Za slanje poruka na određeni kanal implementirana je komanda *bATDSend* kojoj se prosleđuje tip kanala i željeni podatak za slanje. Navedena komanda preko UART protokola šalje odgovarajuću AT komandu za slanje ESP modulu sa prosleđenim parametrima. Bitno je napomenuti da se poruka neće poslati ukoliko se ne dobije potvrda o uspešnosti slanja ili dok drajver nije u potpunosti konfigurisan.

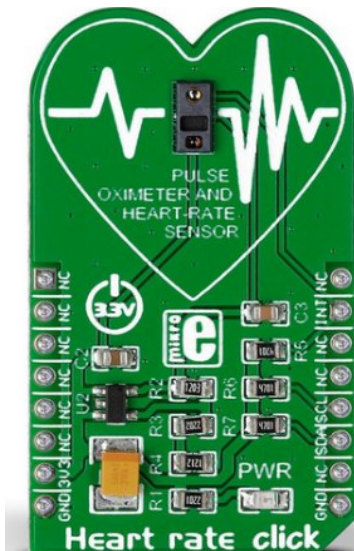
Što se tiče primanja poruka, implementacija je izvedena tako da se pri konfiguraciji AT drajvera proslede dva FreeRTOS Queue-a: jedan za Heart Rate, a drugi za SpO2. Kako su same pročitane vrednosti intidžerskog tipa, očekuje se da Queue-ovi budu tipa int. Ovim pristupom postiže se lako povezivanje ostalih komponenti sistema, jer ne moraju da vrše dodatne operacije parsiranja podataka prilikom čitanja iz Queue-ova.

*\*AT drajver se smatra konfigurisanim kada se kompletno završi ESPSetUpTask task.*

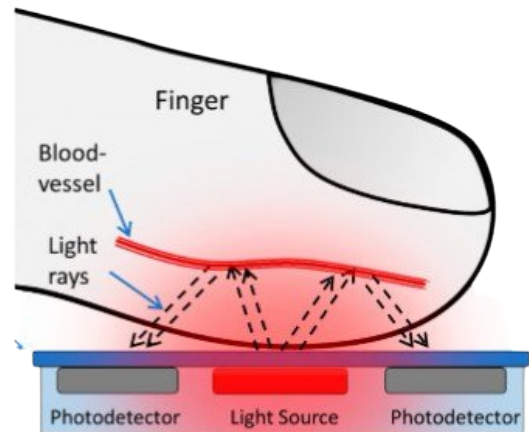
## Oximeter

Za merenja vrednosti Heart Rate i SpO2 korišćen je MirkoE Heart Rate Click (slika 5). Dovoljno je postaviti kažiprst na vrh senzora kako bi se dobile vrednosti pulsa i zasićenosti krvi kiseonikom putem I2C interfejsa.

Ovaj senzor poseduje dve integrisane LED diode, crvenu (RED) i infracrvenu (IR), koje emituju odgovarajuće talasne dužine. Talasne dužine ovih LED dioda su 660 nm i 880 nm, respektivno. Odbijena svetlost se detektuje pomoću crveno/IR foto-detektora, a zatim se meri pomoću delta-sigma 16-bitnog ADC-a sa niskim šumom. (slika 6). Senzor MAX30100 poseduje FIFO bafer kapaciteta 16 reči od 8 bitova u kojem se skladište izmerene vrednosti.



Slika 5: MikroE Heart Rate Click



Slika 6: Merenje vrednosti pomocu LED diode

## Implementacija

Srž implementacije čine dva taska: *MeasureTask* i *UpdateTask*.

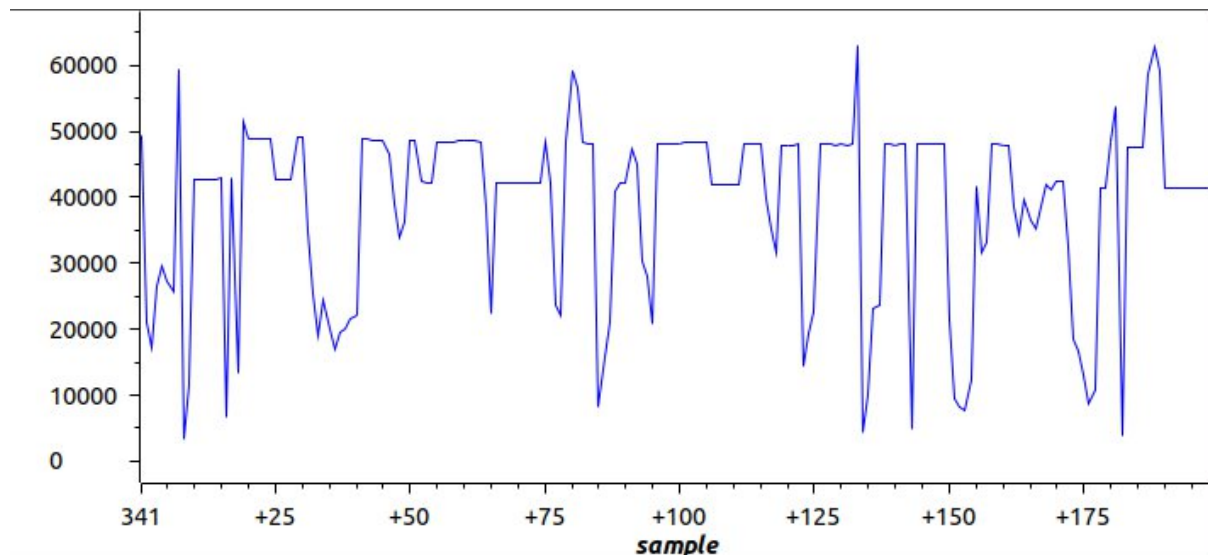
### MeasureTask

Kako iz datog modula dobijamo samo sirove podatke, pre nego što uopšte počnemo da ih koristimo, potrebno ih je prvo obraditi. Glavni cilj ovog taska je prikupljanje i obrada sirovih podataka infracrvenih (IR) i crvenih (RED) zrakova sa senzora. Podaci se obrađuju metodom srednje vrednosti, gde se za određivanje trenutne vrednosti SpO2 i Heart Ratea koristi srednja vrednost poslednja četiri pročitana uzorka.

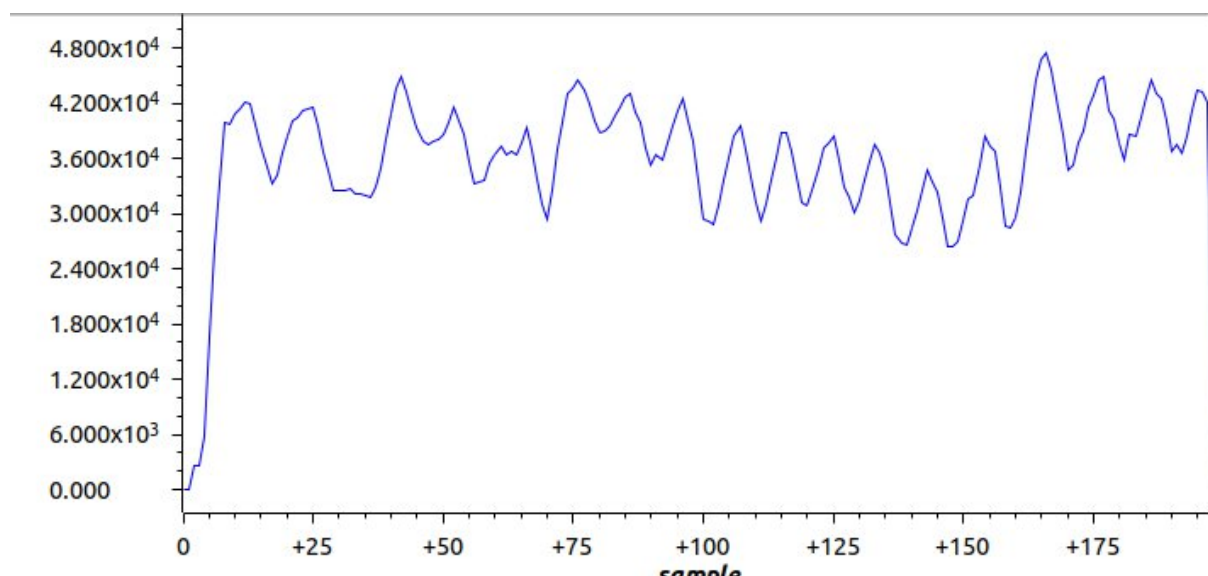
Prikupljanje jednog uzorka vrši se periodičnim čitanjem podataka iz FIFO bafera senzora tokom intervala od 50 milisekundi. Tokom ovog intervala, task akumulira IR i RED vrednosti, računajući prosečne vrednosti pročitanih podataka na kraju intervala.

Nakon prikupljanja, zadatak ažurira bafer uzoraka, uklanjajući najstarije vrednosti i dodajući nove. Kumulativne prosečne vrednosti poslednja četiri uzorka se zatim izračunavaju i šalju putem Queuea za dalje tumačenje u *UpdateTasku*. Na slici 7 može se videti kako izgledaju sirovi podaci IR zraka, a na slici 8 kako izgledaju obrađeni podaci datog zraka.





Slika 7: Izgled sirovog IR zraka



Slika 8: Izgled obrađenog IR zraka

## UpdateTask

*UpdateTask* je zadatak odgovoran za obradu podataka prikupljenih i obrađenih u *MeasureTasku*, sa ciljem izračunavanja Heart Ratea i SpO2.

Dati task čeka na obrađene podatke iz Queuea koje šalje *MeasureTask*. Prilikom obrade pulsa (BPM), task analizira IR podatke kako bi detektovao promene u signalu koje odgovaraju otkucajima srca. Detekcija otkucaja se vrši brojanjem porasta vrednosti u IR podacima, a srednje vreme između otkucaja se koristi za izračunavanje BPMa. BPM se računa fromulom:

$$BPM = \frac{60000}{MeanBeatTimestampValue}$$

Za procenu nivoa kiseonika, task analizira kvadratne sume AC komponenti IR i RED signala. Na osnovu RMS (korena srednje kvadratne vrednosti) odnosa ovih komponenti, task procenjuje SpO2 vrednost. Dodatno, task detektuje prisustvo prsta na senzoru na osnovu praga za IR podatke. Ukoliko IR podaci ne prelaze prag, task resetuje vrednosti BPM i SpO2, signalizirajući odsustvo prsta na senzoru. SpO2 se racuna formulom:

$$SpO_2 = 110 - 14.5 \times RMS$$

## OLED Modul

Za prikaz vrednosti korišćen je MikroE OLED C Click modul (slika 9). Zasnovan je na PSP27801, kvadratnom OLED ekranu dimenzija 25x25mm sa rezolucijom 96x96 piksela i podrškom za punu boju. Ovaj modul koristi grafički drajver SSD1351, koji dolazi sa integrisanim SRAM baferom za prikaz veličine 128x128x18 bita. Komunikacija sa Tiva pločicom ostvaruje se putem standardnog 4-žičnog SPI serijskog interfejsa. Osim SPI pinova, OLED modul takođe poseduje D/C pin koji služi za prebacivanje konteksta pilikom slanja između podataka i komandi. Sa datim modulom moguće je predstaviti 65535 različitih boja (5R, 6G, 5B), a njihov način upisivanja se može videti na slici 10.



Slika 9: MikroE OLED C Click

D7	D6	D5	D4	D3	D2	D1	D0
C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>
B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>

Slika 10: Način upisivanja boje

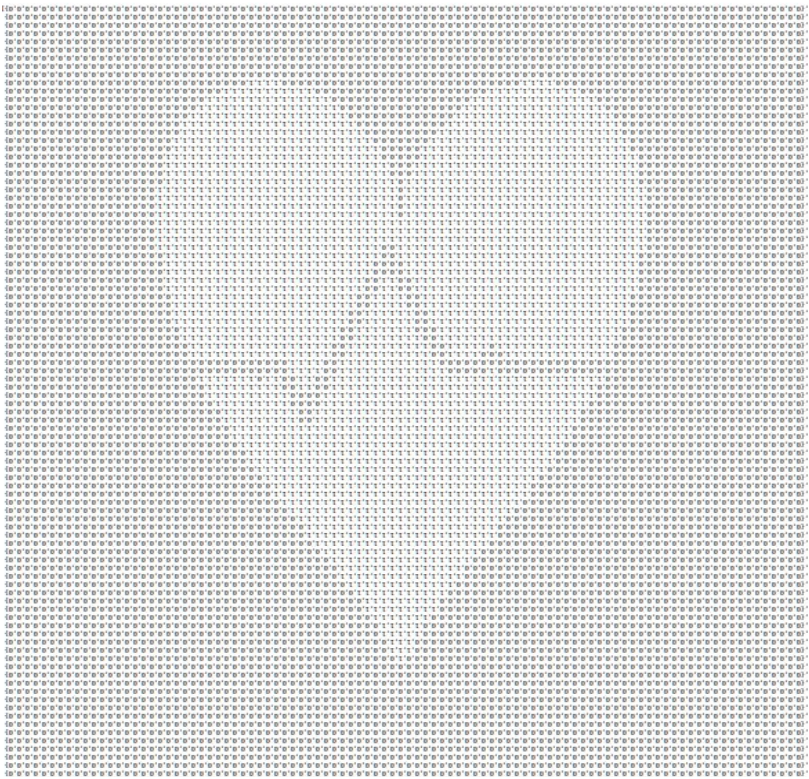
# Implementacija

Prikaz vrednosti se vrši u dva taska: *SpO2Task* i *HeartRateTask*. Prilikom inicijalizacije, na ekranu se iscrtava srce. Umesto matematičkog računanja, koriste se bit mape kako bi se proces prikaza ubrzao uz minimalnu potrošnju memorijskih resursa (slika 11).

Taskovi funkcionišu na sličan način, razlikujući se samo u lokaciji prikaza podataka na ekranu. Puls je prikazan u donjem levom uglu, dok je SpO2 prikazan u donjem desnom uglu (slika 9).

Oba taska čekaju da vrednosti stignu u njihov odgovarajući Queue (SpO2 ili Heart Rate Queue). Kada Oximetar pošalje podatke, te vrednosti se razlažu na cifre jedinica, desetica i stotica. Svaka od tih cifara se zatim upisuje na odgovarajuću poziciju u RAM memoriji OLED ekrana, čime se omogućava njihov prikaz.

Da bi se vrednost upisala u RAM, prvo je potrebno postaviti D/C pin na low. Nakon toga šalju se komande koje definišu pozicije početka i kraja upisivanja kolona i redova. Kada se ove komande pošalju, D/C pin se postavlja na high, a zatim se dalje prosleđeni podaci upisuju u prethodno definisani okvir u RAMu.



Slika 11: Bitmapa prikaza srca