Razvoj ugradbenih sustava

Projekt 

Ak. god. 2024./2025.

Smart Hospital Lobby

*Filip Bačić, Branimir Kuveždić, Martin Hrupec*

Datum predaje: *23. travnja 2025.*

# Sadržaj

[Sadržaj 2](#_Toc196239128)

[Slike 3](#_Toc196239129)

[Dnevnik promjena dokumentacije 4](#_Toc196239130)

[Pojmovnik 5](#_Toc196239131)

[1 Opis projektnog zadatka 1](#_Toc196239132)

[2 Opis sustava 2](#_Toc196239133)

[2.1 Pristup oblikovanju 3](#_Toc196239134)

[2.2 Resursi 3](#_Toc196239135)

[3 Specifikacija zahtjeva 4](#_Toc196239136)

[3.1 Funkcijski zahtjevi 6](#_Toc196239137)

[3.2 Mehanički zahtjevi 6](#_Toc196239138)

[3.3 Cijena 7](#_Toc196239139)

[3.4 Buduće značajke 7](#_Toc196239140)

[4 Detaljnije specifikacije funkcije 8](#_Toc196239141)

[4.1 Implementacija i korisničko sučelje 8](#_Toc196239142)

[4.1.1 Oblikovanje sklopovlja 8](#_Toc196239143)

[4.1.2 Sučelje sklopovlja 9](#_Toc196239144)

[4.1.3 Implementacija 9](#_Toc196239145)

[5 Korištene tehnologije i alati 10](#_Toc196239146)

[6 Zaključak i budući rad 11](#_Toc196239147)

[7 Reference 12](#_Toc196239148)

*Sadržaj se automatski osvježava prema tekstu (desni klik, „Update Field“) uz zadani formata poglavlja.*

# Slike

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Opis promjene/dodatka | Autor(i) | Datum |
| 0.1 | Napravljen predložak. | Hrupec | 17.04.2025. |
| 0.2 | Ažuriranje dokumenta | Bačić | 20.04.2025. |
| 0.6 | Dodavanje dijagrama | Kuveždić | 21.04.2025. |
| 0.7 | Finaliziranje dokumenta | Hrupec | 21.04.2025. |
| 0.8 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Pojmovnik

*ESP32 - Mikroprocesorska razvojna ploča s Wi-Fi i Bluetooth mogućnostima, pogodna za IoT projekte i ugradbene sustave. U ovom projektu koristi se kao središnja jedinica koja upravlja svim perifernim uređajima i komunikacijama*

*Firebase- Cloud platforma tvrtke Google za razvoj aplikacija, korištena ovdje za pohranu i dohvat podataka o pacijentima putem interneta.*

*Firebase Realtime Database - Komponenta Firebasea koja omogućuje sinkronizaciju podataka u stvarnom vremenu između klijenta i baze podataka.*

*FirebaseESP32 - Biblioteka za Arduino okruženje koja omogućuje komunikaciju između ESP32 i Firebasea koristeći REST API*

*Wi-Fi - Bežična mrežna komunikacija koja se koristi za povezivanje uređaja na internet. ESP32 se spaja na Wi-Fi kako bi pristupio Firebaseu.*

*LCD 1602 12C - LCD zaslon s 16 stupaca i 2 reda, s I2C sučeljem koje pojednostavljuje komunikaciju sa samo dvije žice (SDA i SCL). Koristi se za prikaz podataka o pacijentu.*

*12C - Serijski komunikacijski protokol (Inter-Integrated Circuit) koji omogućuje komunikaciju između mikrokontrolera i perifernih uređaja putem dvije žice.*

*LED dioda (indikatorska) - Elektronička komponenta koja svijetli kada je uključena. U projektu se koristi za vizualnu indikaciju statusa unosa: zelena za uspješan, crvena za neuspješan unos.*

*Tipkovnica 4x4 (membranska) - Matrica od 16 tipki povezana u 4 reda i 4 stupca. Omogućuje unos numeričkih i funkcijskih podataka.*

*String - Podatkovni tip u Arduino jeziku za upravljanje tekstualnim podacima, koristi se za pohranu unosa korisnika i podataka o pacijentu.*

*Struktura (struct) - Konstrukt u C/C++ jeziku koji omogućuje grupiranje više varijabli različitih tipova pod jednim imenom. Koristi se za definiranje strukture pacijenta*

*FirebaseData / FirebaseConfig / FirebaseAuth - Objekti biblioteke FirebaseESP32 za upravljanje autentikacijom, konfiguracijom i prijenosom podataka prema Firebaseu.*

*Local database . Lokalna baza podataka definirana unutar koda (niz LocalPatient struktura) koja služi kao sigurnosna kopija ako internet nije dostupan.*

*State machine (stroj stanja) - Programska tehnika u kojoj se sustav nalazi u jednom od unaprijed definiranih stanja i prelazi u drugo stanje na temelju ulaza. Omogućuje jednostavno upravljanje tijekovima rada.*

*millis() - Arduino funkcija koja vraća broj milisekundi od pokretanja programa, koristi se za mjerenje vremena i upravljanje timeoutima.*

*Serial Monitor - Alat u Arduino IDE-u koji omogućuje praćenje ispisa iz ESP32 putem serijske komunikacije (koristan za debugiranje).*

# Opis projektnog zadatka

Cilj projekta **“Smart Hospital Lobby”** je izgraditi ugradbeni sustav koji olakšava i ubrzava proces prijema pacijenata u zdravstvenim ustanovama. Umjesto tradicionalnog poziva po imenu, sustav omogućuje pacijentu da samostalno unese broj svoje zdravstvene kartice. Sustav zatim dohvaća informacije o zakazanom pregledu i prikazuje ih na zaslonu, uz vizualnu potvrdu uspješnosti pomoću LED dioda.

Projekt je razvijen u edukacijske svrhe, ali s realnim potencijalom za implementaciju u bolničke sustave, posebice u uvjetima pandemije, masovnih pregleda ili hitne medicine. Sustav podržava rad i u simuliranom i stvarnom okruženju, osiguravajući dostupnost informacija čak i u slučaju mrežnih prekida.

# Opis sustava

U sklopu projekta **"RUS\_Projekt\_Bačić\_Kuveždić\_Hrupec"**, razvijen je ugradbeni sustav za inteligentno upravljanje čekaonicom u zdravstvenoj ustanovi. Sustav omogućava pacijentima unos broja zdravstvene kartice putem tipkovnice, nakon čega automatski dohvaća i prikazuje relevantne informacije o terminu pregleda i liječniku s Firebase baze podataka ili iz lokalne baze u slučaju nedostupnosti interneta. Informacije se prikazuju na LCD zaslonu, a status operacije se signalizira putem dviju LED dioda – zelene za uspješan dohvat i crvene za grešku ili nedostupan unos.

Cilj sustava je optimizirati proces poziva pacijenata te smanjiti potrebu za ljudskom asistencijom u čekaonicama, posebno u uvjetima povećanog pritiska na zdravstvene ustanove. Sustav je osmišljen tako da podržava rad u stvarnom vremenu (online putem Firebase-a), kao i u offline načinu rada, čime osigurava otpornost na mrežne probleme.

Glavne komponente sustava su:

* **ESP32 razvojna pločica** koja služi kao središnja procesna jedinica.
* **Tipkovnica 4x4** za unos podataka od strane korisnika.
* **LCD zaslon s I2C komunikacijom** za prikaz podataka.
* **Zelena i crvena LED dioda** za vizualnu indikaciju.
* **Firebase baza podataka** za udaljeni pristup informacijama.
* **Lokalna baza podataka (u kôdu)** za offline pristup.

Programsko rješenje implementira stroj stanja (engl. *state machine*) koji upravlja tokovima rada sustava: unos → obrada → prikaz informacija ili greške → povratak. Modularna arhitektura omogućava lako nadograđivanje i fleksibilnu integraciju dodatnih komponenti poput Bluetooth komunikacije, RFID čitača ili naprednog korisničkog sučelja.

Uz stabilan rad, sustav nudi korisnički prihvatljivo sučelje – intuitivni unos i jasan prikaz – što ga čini pogodnim za implementaciju u stvarnim čekaonicama, uz mogućnosti daljnjeg razvoja i integracije u šire bolničke sustave.

## Pristup oblikovanju

Pristup oblikovanju temelji se na modularnosti i otpornosti sustava. Projekt je razvijan s ciljem postizanja funkcionalnosti i pouzdanosti čak i u slučaju nedostupnosti interneta, stoga sustav uključuje lokalnu bazu pacijenata kao sigurnosnu mjeru. Korištena je **višestanja arhitektura (state machine)** za logičko upravljanje sustavom, što omogućuje jasno upravljanje korisničkim interakcijama i jednostavno održavanje.

Razvoj je započeo simulacijom komponenti u **Wokwi emulatoru**, čime se omogućilo brzo testiranje i vizualna provjera funkcionalnosti bez potrebe za stvarnim hardverom. U ovoj fazi naglasak je bio na:

* točnoj emulaciji hardverskih veza (LCD, LED, tipkovnica),
* uspostavi stabilne Wi-Fi veze i povezivanju na Firebase RTDB,
* razvoju sigurnog fallback mehanizma putem lokalne baze podataka.

## Resursi

**Hardverski resursi:**

* ESP32 razvojna pločica s WiFi i GPIO mogućnostima
* LCD 1602 s I2C sučeljem (spojen na GPIO 21 i 22)
* Tipkovnica 4x4 (GPIO pinovi 5, 18, 19, 23, 12, 13, 14, 15)
* LED diode (GPIO 2 za zelenu, GPIO 4 za crvenu)
* Napajanje 5V (preko USB)

**Softverski resursi:**

* Arduino IDE
* Biblioteke: FirebaseESP32, LiquidCrystal\_I2C, Keypad, ArduinoJson
* Simulacijsko okruženje Wokwi (za razvoj bez fizičkog ESP32 hardvera)

# Specifikacija zahtjeva

USE CASE DIJAGRAM

A diagram of a diagram

AI-generated content may be incorrect.

**UC-1: Unos broja kartice**

* **Akter:** Pacijent
* **Opis:** Pacijent unosi broj kartice pomoću numeričke tipkovnice
* **Preduvjet:** Sustav je uključen i u stanju čekanja
* **Postuvjet:** Sustav prepoznaje uneseni broj i prelazi u stanje obrade

**UC-2: Dohvat podataka o pacijentu**

* **Akter:** ESP32
* **Opis:** Sustav dohvaća podatke iz Firebase baze ili lokalne baze u offline načinu
* **Preduvjet:** Broj kartice je unesen
* **Postuvjet:** Informacije su uspješno dohvaćene ili je prikazana poruka o grešci

**UC-3: Prikaz rezultata**

* **Akter:** LCD + LED indikatori
* **Opis:** Rezultati se prikazuju na LCD-u, a LED signalizira ishod
* **Preduvjet:** Podaci su dohvaćeni
* **Postuvjet:** Informacije su prikazane korisniku i čeka se povratna akcija (tipka D)

**UC-4: Reset sustava**

* **Akter:** Pacijent
* **Opis:** Pacijent pritišće ‘D’ za povratak u početno stanje
* **Preduvjet:** Prikaz podataka ili greške je u tijeku
* **Postuvjet:** Sustav je spreman za novi unos

Dijagram prikazuje korisnika (pacijenta) koji koristi sustav putem tipkovnice. Unos broja aktivira dohvat podataka iz baze, a dohvaćeni podaci se prikazuju pacijentu. Prikaz je realiziran putem LCD zaslona i LED indikacije. Ova interakcija se ponavlja ciklički, a cijeli proces je jasan, linearan i prilagođen korisnicima s minimalnim tehničkim znanjem.

SEKVENCIJSKI DIJAGRAM

A diagram of a computer program

AI-generated content may be incorrect.

Sekvencijski dijagram prikazuje redoslijed poruka i poziva između komponenti:

* Pacijent komunicira s tipkovnicom
* Tipkovnica šalje podatke ESP32 mikrokontroleru
* ESP32 traži podatke iz baze
* Podaci se vraćaju i prikazuju na LCD/LED

## Funkcijski zahtjevi

| **Oznaka** | **Zahtjev** |
| --- | --- |
| F-1 | Omogućiti unos broja kartice putem tipkovnice |
| F-2 | Povezati se s Firebase bazom podataka i dohvatiti informacije |
| F-3 | Prikazati informacije o pacijentu na LCD zaslonu |
| F-4 | Upaliti zelenu LED ako je pacijent uspješno pronađen |
| F-5 | U slučaju pogrešnog unosa upaliti crvenu LED i ispisati poruku o grešci |
| F-6 | Omogućiti offline način rada uz lokalnu bazu |
| F-7 | Reset sustava na početno stanje pritiskom tipke D |

## Mehanički zahtjevi

| **Oznaka** | **Zahtjev** |
| --- | --- |
| M-1 | Dimenzije uređaja prilagođene za ugradnju u čekaonički pult |
| M-2 | Tipkovnica mora biti čvrsto fiksirana i lako dostupna korisnicima |
| M-3 | LCD mora biti vidljiv pod različitim kutevima osvjetljenja |
| M-4 | LED indikatori moraju biti jasno vidljivi s udaljenosti od barem 2 metra |
| M-5 | Uređaj mora biti zaštićen od elektrostatskih smetnji i naponskih udara |

## Cijena

Troškovnik komponenata temeljen na cijenama s lokalnih i online dobavljača:

| **Komponenta** | **Jedinična cijena (euro)** |
| --- | --- |
| ESP32 razvojna pločica | 10 |
| LCD 1602 I2C | 8 |
| Tipkovnica 4x4 | 3 |
| 2x LED diode + otpornici | 3 |
| Napajanje (adapter ili USB) | 7 |
| **Ukupno** | 31 |

## Buduće značajke

**Bluetooth povezivost** – za komunikaciju s mobilnim aplikacijama ili medicinskim osobljem.

**RFID čitač** – za automatsko prepoznavanje kartica bez fizičkog unosa.

**Govorno sučelje** – za slabovidne korisnike.

**Integracija s bolničkim informacijskim sustavima** – automatsko ažuriranje termina i soba.

**Analitika korištenja** – statistika ulaza, vremena čekanja, itd.

**Baterijsko napajanje** – za mobilne verzije u privremenim čekaonicama ili terenskoj upotrebi.

# Detaljnije specifikacije funkcije

DIJAGRAM STANJA

A diagram of different types of lines

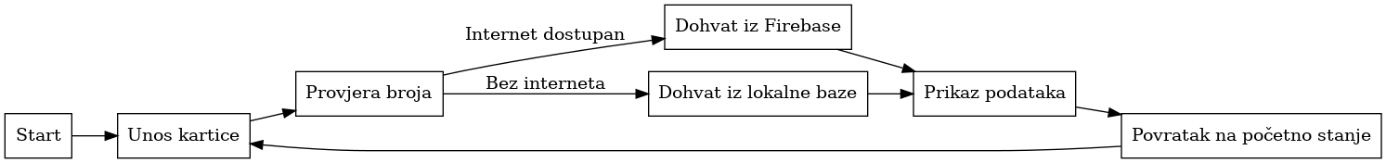
AI-generated content may be incorrect.

Dijagram prikazuje prijelaze između stanja unutar sustava:

1. Čekanje na unos
2. Obrada kartice
3. Prikaz rezultata (ili greške)
4. Povratak na početno stanje

Ovaj dijagram je ključan za razumijevanje kako sustav reagira na korisničke akcije i vanjske događaje.

DIJAGRAM AKTIVNOSTI



Dijagram aktivnosti modelira tok radnji koje se izvršavaju tijekom jedne sesije korištenja: od unosa kartice, preko provjere internetske veze do dohvaćanja i prikaza podataka.

Koristan je za razumijevanje toka podataka i logike odlučivanja (npr. provjera dostupnosti interneta).

## Implementacija i korisničko sučelje

### Oblikovanje sklopovlja

Sklopovlje se temelji na ESP32 kontroleru s GPIO pinovima konfiguriranim za ulazno-izlazne jedinice:

* I2C linije (GPIO 21 i 22) spajaju LCD zaslon.
* GPIO 12–15 i 5, 18, 19, 23 povezani su na tipkovnicu kao digitalni ulazi.
* GPIO 2 i 4 koriste se za LED indikatore.

Napajanje dolazi preko USB priključka (5V), što je dovoljno za rad cijelog sustava bez dodatne regulacije.

Shema je simulirana u [Wokwi](https://wokwi.com/" \t "_new) gdje su svi elementi virtualno povezani i testirani u stvarnom vremenu.

### Sučelje sklopovlja

Komunikacija s LCD zaslonom ostvaruje se putem I2C sabirnice, čime se štedi na broju korištenih pinova. Tipkovnica koristi matricu koja se aktivno skenira, detektira pritiske i prepoznaje unesene znakove. Firebase komunikacija odvija se putem WiFi veze i REST API-ja, uz pomoć biblioteke FirebaseESP32.

### Implementacija

Kod je podijeljen u logičke cjeline:

* **setup()** – inicijalizacija perifernih uređaja i internetske veze.
* **loop()** – upravljanje stanjem sustava, uključujući detekciju unosa, dohvat podataka i prikaz.
* **getPatientData()** – dohvaća podatke iz Firebase baze ili lokalnog niza.
* **displayPatientInfo()** – obrada i prikaz podataka na LCD.

Sustav koristi **state machine** pristup radi lakšeg upravljanja korisničkim interakcijama i održavanja softverske logike.

# Korištene tehnologije i alati

| **Tehnologija / Alat** | | **Opis** |
| --- | --- | --- |
| **ESP32 DevKitC V4** | Glavni mikrokontroler s ugrađenim WiFi modulom. | |
| **Arduino IDE** | Razvojno okruženje korišteno za programiranje ESP32. | |
| **Wokwi Simulator** | Online simulacijsko okruženje za testiranje bez hardvera. | |
| **Firebase Realtime DB** | Cloud baza za pohranu termina pacijenata. | |
| **LiquidCrystal\_I2C** | Biblioteka za rad s LCD zaslonom putem I2C sučelja. | |
| **Keypad Library** | Biblioteka za rad s matricom tipkovnice. | |
| **ArduinoJson** | Parsiranje JSON odgovora iz Firebase baze. | |
| **GitHub** | Verzijsko upravljanje i pohrana izvornog koda. | |

# Zaključak i budući rad

Projekt "Smart Hospital Lobby" uspješno je implementiran kao funkcionalni ugradbeni sustav koji poboljšava iskustvo pacijenata u čekaonicama putem jednostavnog i intuitivnog korisničkog sučelja. Korištenjem ESP32 mikrokontrolera, LCD zaslona i tipkovnice, sustav omogućuje unos broja zdravstvene kartice, dohvat termina iz Firebase baze te offline način rada putem lokalne memorije. Vizualna signalizacija putem LED dioda dodatno pojačava korisničku interakciju.

Kroz razvoj projekta, tim je stekao vrijedna znanja iz područja ugradbenih sustava, IoT tehnologija, obrade podataka i simulacije rada sustava bez fizičkog hardvera. Tehnički izazovi poput nepostojanja internetske veze, prepoznati su i riješeni uvođenjem redundancije kroz lokalnu bazu podataka.

U budućnosti, projekt se može nadograditi dodatnim funkcionalnostima poput Bluetooth povezivanja, mobilne aplikacije, prepoznavanja glasom ili povezivanja s većim bolničkim sustavima. Time se otvaraju mogućnosti za komercijalnu primjenu u stvarnim okruženjima.

# Reference

[1] Arduino Reference <https://www.arduino.cc/reference/en/>

[2] Firebase ESP32 Client Library Documentation <https://github.com/mobizt/Firebase-ESP-Client>

[3] LiquidCrystal\_I2C Library Documentation <https://github.com/johnrickman/LiquidCrystal_I2C>

[4] ArduinoJson Library <https://arduinojson.org/>

[5] Wokwi Documentation <https://docs.wokwi.com>

[6] ESP32 Technical Reference Manual <https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>

Dodatak A:

*U dijelu dodataka potrebno je prema potrebi dati detalje ključnih elemeta projekta.*