

Sveobuhvatni Plan Rada: Migracija Hotel Controller (ESP32)

1. Cilj Projekta

Cilj ovog projekta je potpuna migracija postojeće STM32F4 "Hotel Controller" aplikacije na ESP32 (WT32-ETH01) platformu. Projekat zahtijeva fundamentalnu promjenu arhitekture sa monolitne petlje (HC_Service) na modularni FreeRTOS sistem baziran na "Menadžerima", kako je definisano u originalnom Plan rada.pdf.

Ključni zahtjev je zadržavanje **apsolutne kompatibilnosti (V1)** sa postojećom infrastrukturom, specifično sa HTTP CGI komandama i logikom HTTP odgovora.

2. Ključne Strateške Odluke (Potvrđeno)

Na osnovu naše analize, donecene su sljedeće ključne odluke koje definišu smjer razvoja:

- Hardverska Platforma:** Koristi se isključivo hardverska mapa pinova definisana u postojećem ESP32 projektu (ProjectConfig.h). Originalna mapa pinova iz Plan rada.pdf se ignoriše.
- Strategija Skladištenja:** Eksterni SPI Flash čip se u potpunosti uklanja. Primarni i jedini medij za skladištenje fajlova (Firmware, Slike, Konfiguracije) postaje **uSD kartica (FAT File System)**.
- Izvor Fajlova:** Svi moduli, prvenstveno UpdateManager, moraju biti refaktorisani da čitaju fajlove (npr. "NEW.BIN", "IMG20.RAW") direktno sa uSD kartice, replicirajući logiku iz hotel_ctrl.c.
- Hibridna Lista Adresa:** Implementira se hibridni model keširanja:
 - Izvor (Master):** Fajl CTRL_ADD.TXT na uSD kartici.
 - Keš (Radna Lista):** EEPROM.
 - Okidač:** Nova komanda cad=load (putem novog dugmeta na web stranici) pokreće čitanje sa CTRL_ADD.TXT i upisivanje (keširanje) liste u EEPROM.
 - Logika Pollinga:** LogPullManager čita adrese **isključivo** iz EEPROM keša, što omogućava rad i ako se uSD kartica ukloni.
- HTTP Upload Fajlova:** Stari TFTP server se ne implementira. Umjesto njega, HttpServer će biti nadograđen da podržava HTTP POST upload (/upload-firmware) direktno na uSD karticu, kao i preglednik fajlova (listanje/brisanje).
- HTTP Kompatibilnost Odgovora (V1):** Ovo je kritičan zahtjev. HttpServer **mora** replicirati SSI ponašanje starog sistema.
 - Odgovor na CGI komandu **ne smije** biti text/plain ("OK", "TIMEOUT" itd.).
 - Odgovor **mora** biti text/html, gdje se u HTML šablon (baziran na log.html) na mjesto taga \$!--#t-->\$ umeće **isključivo** jedan od dozvoljenih stringova: "OK", "TIMEOUT", "ERROR", "BUSY", "EMPTY", "DELETED", ili specifični string sa podacima (za cst ili

log=3).

7. **Faza Razvoja:** Projekat ostaje u **Fazi 1** (Debug na SerialO).

3. Definicija Arhitekture (ESP32)

3.1. Centralni Dispečer (Rs485Service)

Sistem je baziran na FreeRTOS zadatku Rs485Service koji djeluje kao dispečer i upravlja pristupom RS485 magistrali sa sljedećim prioritetom:

1. **Najviši:** HttpQueryManager (Korisnički blokirajući upit)
2. **Visok:** UpdateManager (Proces ažuriranja firmvera)
3. **Normalan:** LogPullManager (Redovni polling uređaja)
4. **Nizak:** TimeSync (Broadcast vremena)

3.2. Hardverska Mapa (Potvrđena)

Koriste se pinovi definisani u ProjectConfig.h:

- **RS485:** RX=5, TX=17, DE=33
- **I2C (EEPROM):** SDA=21, SCL=22
- **SPI (za uSD):** SCK=14, MISO=19, MOSI=18, CS=23
- **Ethernet:** Periferija integrisana na WT32-ETH01.
- **Status LED:** 2
- **WLAN Reset:** 32
- **Debug:** SerialO (GPIO 1, 3)

4. Revidirani Plan Implementacije (Sekvence)

Ovaj plan definiše tačne korake za refaktorisanje postojećeg ESP32 koda kako bi se uskladio sa strateškim odlukama.

Sekvenca 1: Promjena Infrastrukture (Hardverska Apstrakcija)

Cilj: Ukloniti drajver za SPI Flash i zamijeniti ga drajverom za uSD karticu.

1. **Zadatak 1.1: Ažuriranje platformio.ini**
 - Ukloniti lib_deps = adafruit/Adafruit SPIFlash.
 - Dodati zavisnosti za FS.h i SD.h (provjeriti da li su dio Arduino core-a).
2. **Zadatak 1.2: Uklanjanje SpiFlashStorage Modula**
 - Izbrisati fajlove SpiFlashStorage.h i SpiFlashStorage.cpp iz projekta.
3. **Zadatak 1.3: Kreiranje SdCardManager Modula (Novi Fajlovi)**
 - Kreirati SdCardManager.h i SdCardManager.cpp.
 - Klasa mora imati Initialize() metodu koja poziva SD.begin(23, SPI, ...) koristeći potvrđene SPI pinove.
 - Implementirati javne metode: bool IsCardMounted(), String ReadTextFile(const char* path), File OpenFile(const char* path, const char* mode), String ListFiles(const char*

path), bool DeleteFile(const char* path).

4. Zadatak 1.4: Ažuriranje main.cpp

- Ukloniti #include "SpiFlashStorage.h".
- Dodati #include "SdCardManager.h".
- Zamijeniti globalni objekat g_spiFlashStorage sa g_sdCardManager.
- Ažurirati setup(): zamijeniti poziv g_spiFlashStorage.Initialize(...) sa g_sdCardManager.Initialize().
- Ažurirati proslijedivanje pokazivača modulima HttpServer i UpdateManager.

Sekvenca 2: Refaktorisanje Mape Skladištenja i Podataka

Cilj: Ažurirati EEPROM mapu i module koji zavise od nje.

1. Zadatak 2.1: Ažuriranje ProjectConfig.h

- Postaviti MAX_ADDRESS_LIST_SIZE na 500.
- Implementirati Dinamičku Mapu:
 - EEPROM_ADDRESS_LIST_START_ADDR mora biti (EEPROM_CONFIG_START_ADDR + sizeof(AppConfig)).
 - EEPROM_ADDRESS_LIST_SIZE mora biti (MAX_ADDRESS_LIST_SIZE * sizeof(uint16_t)).
 - EEPROM_LOG_START_ADDR mora biti (EEPROM_ADDRESS_LIST_START_ADDR + EEPROM_ADDRESS_LIST_SIZE).

2. Zadatak 2.2: Ažuriranje EepromStorage Modula

- Ažurirati ReadAddressList da koristi novi EEPROM_ADDRESS_LIST_SIZE.
- Dodati novu metodu: bool WriteAddressList(uint16_t* listBuffer, uint16_t count).
- Potvrditi da LoggerInit koristi novi, dinamički izračunati EEPROM_LOG_START_ADDR.

Sekvenca 3: Refaktorisanje Glavnih Modula (Logika Aplikacije)

Cilj: Prebaciti logiku UpdateManager-a i HttpServer-a na FAT FS.

1. Zadatak 3.1: Refaktorisanje UpdateManager (Čitanje sa FS-a)

- Promijeniti Initialize() da prima SdCardManager*.
- Prepraviti PrepareSession():
 - Mora provjeriti m_sdCardManager->IsCardMounted().
 - Ukloniti logiku za VERS_INF_OFFSET.
 - Mora koristiti m_sdCardManager->OpenFile(...) da otvorи traženi fajl (npr. "IMG20.RAW").
 - fw_size se postavlja na file.size().
 - (Logika za CRC32 se mora razmotriti - računanje u letu ili čitanje iz .meta fajla).
- Prepraviti SendDataPacket(): Mora čitati podatke koristeći File::read().

2. Zadatak 3.2: Refaktorisanje HttpServer (Handleri i Odgovori)

- Promijeniti Initialize() da prima SdCardManager*.
- Prepraviti HandleFileUpload(): Mora koristiti m_sdCardManager za upis fajla na uSD karticu.
- Kreirati log_html.h fajl koji sadrži log.html sa tagom \$<!--#t-->\$.

- Prepraviti HandleSysctrlRequest():
 - Implementirati SSI logiku odgovora (V1 Kompatibilnost). Svi odgovori se umeću u LOG_HTML šablon.
 - Dodati case za cad=load:
 1. Poziva m_sdCardManager->ReadTextFile("CTRL_ADD.TXT").
 2. Parsira string u uint16_t niz.
 3. Poziva m_eepromStorage->WriteAddressList(...) da kešira listu.
 - Dodati nove handlere za list_files i delete_file (za File Browser).
3. **Zadatak 3.3: Ažuriranje index_html.h (Frontend)**
- Dodati dugme "Ucitaj Listu Kontrolera" (send_event(610)).
 - Dodati novu HTML formu za upload fajlova (poziva /upload-firmware).
 - Dodati novu sekciju (npr. <div>) za prikaz liste fajlova sa uSD kartice (File Browser).

Sekvenca 4: Završna Stabilnost

1. **Zadatak 4.1: Testiranje Faze 1**
 - Sav Serial.println ispis se koristi za debagovanje.
2. **Zadatak 4.2: Buduća Faza 2 (Nije u ovom opsegu)**
 - Kasnija implementacija CommandParser modula za DWIN displej.