

BrewMate

Introducere

BrewMate este un asistent digital portabil creat special pentru pasionații de cafea care explorează cafenelele orașului. Proiectul îmbină plăcerea descoperirii cu tehnologia embedded, oferind o experiență personalizată de jurnalizare și recomandare.

Funcționalități cheie:

- Înregistrează automat cafenelele vizitate și tipurile de cafea consumate
- Salvează datele pe un card SD pentru consultare ulterioară
- Oferă recomandări bazate pe istoricul tău de consum (OpenAI)
- Interfață prietenoasă pentru navigare facilă
- (Extra) Funcție vocală pentru sugestii dinamice de localuri sau sortimente de cafea

Descriere generală

BrewMate este construit ca un sistem embedded interactiv, gândit să funcționeze autonom și să ofere o experiență fluidă utilizatorilor. Dispozitivul combină module hardware și software pentru a înregistra, organiza și recomanda preferințe legate de cafea.

Utilizatorii pot înregistra automat cafenelele vizitate și tipurile de cafea consumate, aceste date fiind salvate pe un card SD pentru consultare ulterioară. Fiecare înregistrare conține numele cafenelei și tipul de cafea comandat.

Înainte de o nouă ieșire, utilizatorii pot vizualiza aceste date direct de pe display, unde interfața le va afișa sugestii filtrate pe baza istoricului.

BrewMate integrează și o funcție vocală inteligentă care utilizează API-ul OpenAI pentru generarea de recomandări. Aceasta analizează preferințele înregistrate și oferă sugestii dinamice privind cafelele noi sau sortimente potrivite gusturilor utilizatorului.

Datele sunt gestionate într-un format structurat și pot fi exportate sau prelucrate în aplicații externe în versiuni viitoare. Sistemul este modular și permite adăugarea de noi funcționalități (ex. analiză statistică, integrare cu aplicații mobile).



Hardware Design

Componente:

- ESP32 WROOM32
- Display TFT SPI 2.4" ST7789
- Modul microSD Card
- Microfon INMP441
- Acumulator Li-ion 18650 + modul TP4056
- Breadboard-uri

Interfețe hardware folosite:

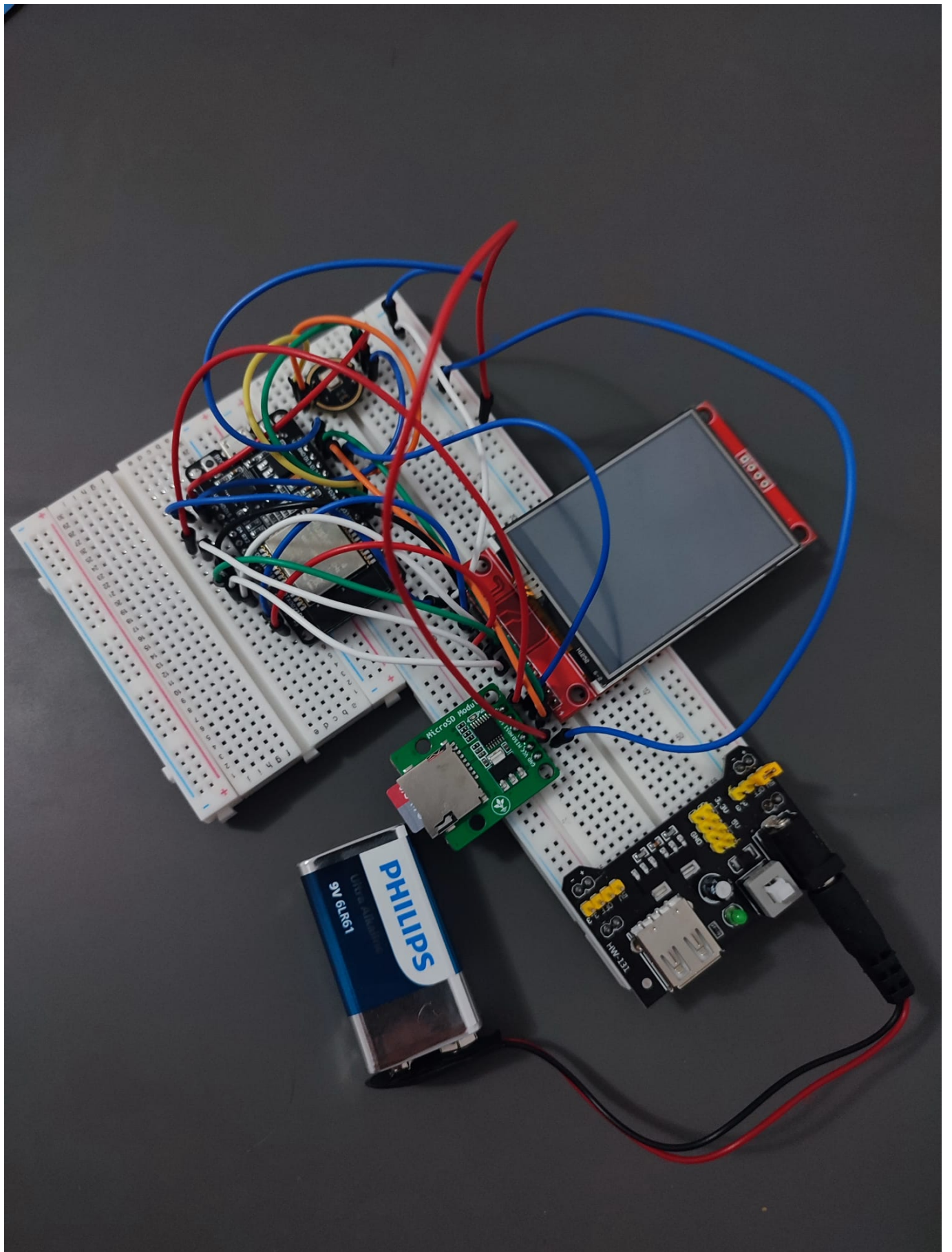
- GPIO – pentru controlul pinilor
- SPI – pentru TFT și SD card
- I2S – pentru captarea sunetului
- Timere – pentru gestionarea timeout-urilor
- Întreruperi – pentru input eficient

Conectivitate:

Modul	Semnal	ESP32 Pin
INMP441	I2S SCK	GPIO33
	I2S WS (LRCL)	GPIO25
	I2S SD	GPIO32
	VCC	3.3V
	GND	GND

Modul	Semnal	ESP32 Pin
SD Card	CS	GPIO2
	MOSI	GPIO13
	MISO	GPIO12
	SCK	GPIO14
	VCC	3.3V
	GND	GND

Modul	Semnal	ESP32 Pin
TFT Display	CS	GPIO5
	MOSI	GPIO23
	SCK	GPIO18
	DC	GPIO21
	RESET	GPIO22
	BL (Backlight)	GPIO4
	VCC	3.3V/5V
	GND	GND



Software Design

Mediu de dezvoltare:

- Thonny
- Platforma de testare: ESP32 + MicroPython

Biblioteci utilizate:

- machine, network, urequests, json, time, gc, os

Implementare:

- Clasa **WorkingSDCard** este responsabilă de gestionarea comunicației SPI software cu cardul SD, folosind transmisie bit cu bit (bit-banging) prin pini digitali. Aceasta implementează funcționalități de inițializare a cardului, trimitere de comenzi standard (CMD0, CMD8, CMD17, etc.), și scriere/citire blocuri de 512 bytes.
- Clasa **SimpleCoffeeFS** acționează ca un mini-sistem de fișiere, folosind un singur sector predefinit de pe cardul SD pentru a salva un jurnal cu ultimele cafele consumate. Structura fiecărei înregistrări conține timestamp-ul, numele cafelei și tipul de cafea. Această clasă gestionează atât încărcarea datelor la pornire, cât și salvarea înregistrărilor noi, menținând un număr limitat de intrări pentru a nu depăși limita de memorie a unui sector.

```
def add_entry(self, shop_name, coffee_type):
    """Add coffee entry and save to SD card"""
    try:
        print(f"Adding coffee entry: {shop_name} - {coffee_type}")

        # Add to memory first
        timestamp = time.ticks_ms()
        entry = {
            'timestamp': timestamp,
            'shop': shop_name,
            'coffee': coffee_type
        }

        self.entries.append(entry)

        # Keep only recent entries to fit in one sector
        if len(self.entries) > self.max_entries_per_sector:
            self.entries = self.entries[-self.max_entries_per_sector:]

        # Save to SD card
        success = self._save_entries()

        if success:
            print(f"☑ Coffee entry saved to SD card!")
        else:
            print(f"⚠ Entry saved to memory but SD write failed")
```

```

    return True # Always return True since we have it in memory

except Exception as e:
    print(f"❌ Error adding entry: {e}")
    return False

```

- Funcțiile pentru randare pe ecran folosesc un font bitmap 8×8 stocat ca dicționar. Caracterele sunt desenate pixel cu pixel direct pe ecranul TFT prin funcții dedicate precum draw_large_char() și draw_large_char_line().
- Pentru conectivitate, există **funcții de conectare la rețeaua WiFi** (connect_wifi) și de **trimitere a cererilor către API-ul OpenAI** (make_openai_request). Acestea permit dispozitivului să obțină sugestii inteligente pe baza jurnalului de cafea, folosind modelul GPT. Cererile sunt trimise în format JSON și răspunsul este extras și afișat într-un mod ușor de parcurs.

```

def connect_wifi():
    """Connect to WiFi"""
    print("Connecting to WiFi...")

    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)

    if not wlan.isconnected():
        print(f"Connecting to {WIFI_SSID}...")
        wlan.connect(WIFI_SSID, WIFI_PASSWORD)

        timeout = 15
        while not wlan.isconnected() and timeout > 0:
            print(f"Waiting... {timeout}s")
            time.sleep(1)
            timeout -= 1

        if wlan.isconnected():
            print("WiFi connected!")
            print(f"IP: {wlan.ifconfig()[0]}")
            return True
        else:
            print("WiFi connection failed!")
            return False
    else:
        print("Already connected")
        return True

```

- Interfața aplicației este implementată prin funcții care gestionează **meniurile, logarea, afișarea intrărilor și recomandările AI**. Funcții precum show_coffee_menu(), log_coffee_entry(), view_coffee_logs() și get_coffee_recommendation() definesc fluxul principal al aplicației, cu o interfață simplă, intuitivă și adaptată limitărilor hardware.

Rezultate Obținute

- Dispozitivul este capabil să logheze cafele offline pe SD Card, păstrând persistența datelor la repornire.
- Răspunsuri corecte și relevante de la OpenAI pe baza jurnalului de cafele.
- Afișare grafică completă: meniu, text, recomandări scrollabile.
- Modularitate bună pentru extensii viitoare (cloud sync, voice, statistici).

Concluzii

BrewMate demonstrează că un sistem embedded poate oferi o experiență interactivă și personalizată, integrând funcții moderne de AI într-o interfață minimalistă. Dispozitivul are potențial real pentru dezvoltare într-un produs comercial pentru pasionații de cafea și entuziaști tech deopotrivă.

Download

Software-ul este disponibil aici: <https://github.com/alexandrufu/BrewMatePM>

Bibliografie/Resurse

Resurse Software:

- <https://techtotinker.com/2021/04/08/024-esp32-micropython-how-to-use-sd-card-in-micropython/>
- <https://randomnerdtutorials.com/getting-started-thonny-micropython-python-ide-esp32-esp8266/>
- https://micropython-stubs.readthedocs.io/en/main/11_install_stubs.html

Resurse Hardware:

- <https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf>
- <https://www.instructables.com/ESP32-Mic-Testing-With-INMP441-and-DumbDisplay/>
- <https://randomnerdtutorials.com/esp32-tft-touchscreen-display-2-8-ili9341-arduino/>
- <https://www.electronicwings.com/esp32/microsd-card-interfacing-with-esp32>
- https://www.youtube.com/watch?v=rq5yPjX_uk
- <https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/rnedelcu/alexandru.trifu2712>



Last update: **2025/05/27 21:12**