

Vizualizare 3D și Web Server pe ESP32

Sisteme cu microprocesoare

Paladi Andrei
Ciobanu Andrei
Timofte Alin-Gabriel

Mai 2025

Cuprins

| | | |
|-----------|---------------------------------|----------|
| 1 | Introducere | 2 |
| 2 | Scopul Proiectului | 2 |
| 3 | Componente Hardware | 2 |
| 4 | Arhitectura Sistemului | 3 |
| 5 | Funcționalitate Software | 4 |
| 6 | Interfața Web | 4 |
| 7 | Testare și Dificultăți | 5 |
| 8 | Ce Am Învățat | 5 |
| 9 | Concluzii | 5 |
| 10 | Bibliografie | 5 |

1 Introducere

Internet of Things (IoT) este un domeniu în plină expansiune care permite interconectarea dispozitivelor fizice prin rețele digitale. Acest proiect urmărește realizarea unei aplicații care combină senzori, procesare și interfață grafică, toate implementate pe un microcontroller ESP32-C3.

2 Scopul Proiectului

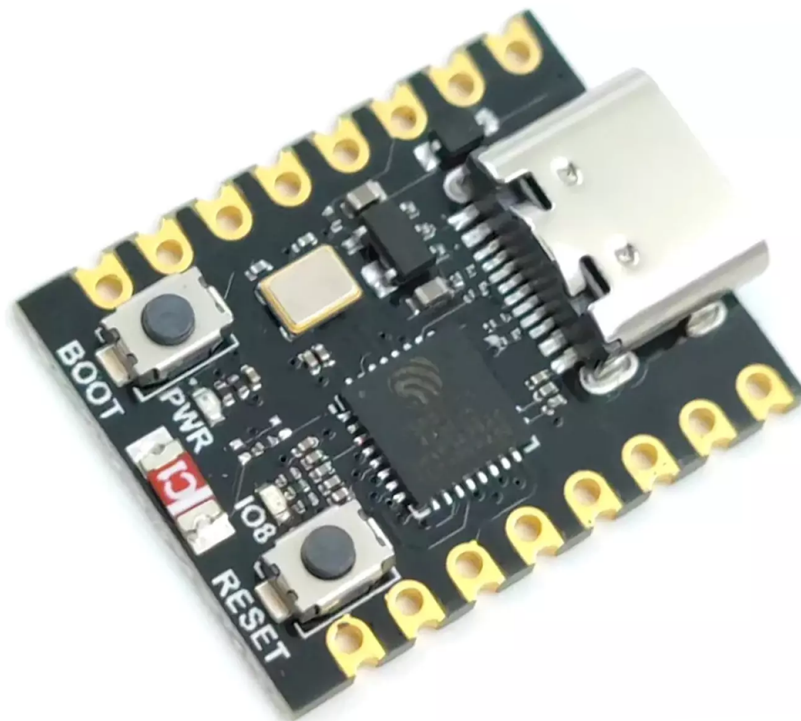
Scopul este de a implementa un sistem embedded care:

- Preia date de la un senzor de accelerație (MPU6500)
- Afășează în timp real un cub 3D rotativ pe un ecran ST7789
- Oferă o interfață web unde este vizualizată mișcarea sub forma unei mingi animate

3 Componente Hardware

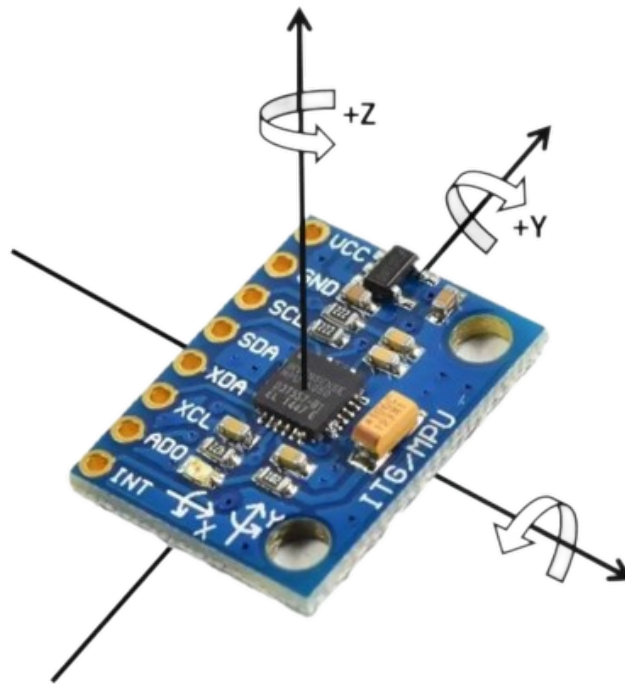
ESP32-C3 Super Mini

Placă de dezvoltare cu procesor RISC-V, suport WiFi, GPIO și consum redus de energie.



MPU6500

Senzor IMU cu accelerometru și giroscop pe 3 axe, comunică prin I2C.



ST7789

Display SPI color, 240x240 pixeli, utilizat pentru randarea grafică 2D/3D.

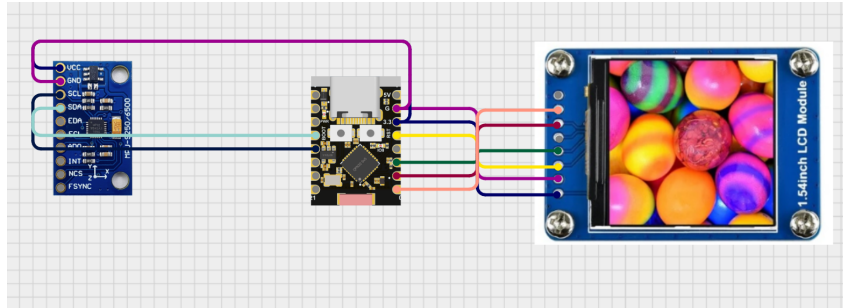


4 Arhitectura Sistemului

Schema bloc:

- Senzor MPU6500 → măsoară accelerația

- **ESP32** → citește datele, le filtrează, le afișează și le trimite pe Web
- **ST7789** → afișează cubul 3D rotativ
- **Browser Web** → afișează mingea animată



5 Funcționalitate Software

Proiectul este scris în C++ folosind Arduino Framework. Codul este împărțit logic între:

- **Inițializare hardware:** senzor, ecran, WiFi
- **Prelucrare date:** filtrare low-pass
- **Redare grafică:** cub 3D pe ecran
- **Interfață Web:** actualizare mingea virtuală

Exemplu de cod: Inițializare MPU și filtrare

```

1 Wire.begin(SDA_PIN, SCL_PIN);
2 mpu.init(calib);
3 mpu.update();
4 mpu.getAccel(&accel);
5
6 filteredX = alpha * accel.accelX + (1 - alpha) * filteredX;
7 filteredY = alpha * accel.accelY + (1 - alpha) * filteredY;
```

Cub 3D și proiecția lui

Cubul este reprezentat de 8 puncte 3D care sunt rotite și proiectate pe ecran.

```

1 float rotx = cube_vertex[i][2] * sin(y) + cube_vertex[i][0] * cos(y);
2 float rotyy = roty * cos(x) - rotz * sin(x);
3 float rotxxx = rotx * cos(z) - rotyy * sin(z);
4 rotxxx *= viewDistance / (viewDistance + rotzzz);
5 rotxxx += xOrigin;
```

6 Interfața Web

Webserverul servește o pagină HTML + CSS + JS care conține o minge ce se mișcă în funcție de accelerație.

Fragment JavaScript

```
1 fetch('/data')
2   .then(r => r.json())
3   .then(data => {
4     velX += data.accelX * accelScale;
5     posX += velX;
6     document.getElementById("ball").style.left = posX + "px";
7   });
```

7 Testare și Dificultăți

Dificultăți întâmpinate:

- Lipsă documentație completă pentru MPU6500 pe ESP32-C3
- Probleme inițiale cu biblioteca ST7789
- Filtrarea zgomotului accelerometrului

Soluții:

- Am folosit biblioteca FastIMU pentru abstractizarea senzorului
- Implementare filtru low-pass pentru reducerea fluctuațiilor

8 Ce Am Învățat

- Inițializarea și utilizarea senzorilor I2C în medii embedded
- Cum funcționează un server web minimal pe microcontroler
- Noțiuni de proiecție 3D pe 2D și transformări matematice
- Sincronizarea taskurilor și desenarea în timp real

9 Concluzii

Am demonstrat că ESP32-C3 poate gestiona eficient un sistem IoT cu vizualizare grafică locală și interfață Web. Sistemul poate fi extins pentru aplicații de detecție mișcare, control gesturi, sau chiar integrare în AR/VR.

10 Bibliografie

- <https://randomnerdtutorials.com>
- <https://github.com/mjs513/FastIMU>
- https://github.com/Bodmer/TFT_eSPI
- <https://docs.espressif.com>

- Video demonstrație/funcționalitate:
https://drive.google.com/file/d/1_Hu4CtCq_ipnwNAGPkaHM4mkgjyMcDfi/view?usp=drive_link
- Video demonstrație/funcționalitate:
https://drive.google.com/file/d/1DvNk_oyfeJiqVnGHw953YaIxfmRQ6-sX/view?usp=drive_link
- **Codul complet pe GitHub (PokeBuddy):**
<https://github.com/AndreiPaladi/Proiect-SM-PokeBuddy>