



## Sisteme Specializate cu Microprocesoare

### Temă proiect

#### **Resurse**

- *Reference Manual*
- *Pinouts*
- *Schematics*
- *Toolkit*

#### **Structură generală a proiectului**

**! Acstea cerințe vor fi respectate de către toți studenții !**

- Notarea
  - **40%** - documentația proiectului, care trebuie să fie concisă și pe baza căreia un cititor să înțeleagă scopul și rezultatele proiectului;
  - **60%** - prezentarea practică.
- Versionare
  - Crearea unui repository pe platforma **Github** unde va fi salvat codul proiectului (headere și fișiere sursă) împreună cu un fișier **README.md** care să descrie pașii de initializare și setup;
  - Un singur branch final, mesaje de commit-uri care cuprind esența modificărilor aduse, structura proiectului (folder **src** pentru implementarea în Keil, folder **doc** pentru documentație și imaginile folosite, folder **gui** pentru codul interfeței grafice, **README**-ul contine numele coechipierilor și descrierea succintă a proiectului, fără arhive, fără numere de versiuni la documentele de tip .docx).
- Documentația
  - Documentația va conține imagini/gif-uri care să reflecte rezultatele obținute/printscreen-uri ce conțin terminalul (va afișa mesaje referitoare la starea curentă a proiectului, pe baza comenzielor date din interfața grafică), fereastra destinată controller-ului de Python și o fereastră cu captura video a plăcii pe parcursul execuției (e.g. **DroidCam**), Figura 3;
  - Secvențe de cod explicate, formatace cu ajutorul plugin-ului **Easy Code Formatter** - fără printscreen-uri de cod;
  - Documentația trebuie să conțină neapărat o schemă bloc și o diagramă de stări cu comentarii

**Exemplu:** Pentru o aplicație în care selectez prin intermediul GUI un anumit LED din LED-ul RGB prezent pe platforma de dezvoltare și îi controliez intensitatea luminoasă, Figurile 1 și 2 exemplifică realizarea acestor diagrame;

- Arhitectură full-scale, specificații, proiectare preliminară, proiectare detaliată, proceduri de testare, identificarea limitărilor, lecții învățate și recomandări pentru viitorii colegi care vor implementa proiecte similare;
- Documentația de proiect va fi în format **.docx**, cu următoarele secțiuni
  - \* **Scopul proiectului** - descriere a contextului practic în care ar putea fi folosit;
  - \* **Configurare** - enumerarea perifericelor care au fost utilizate și modul lor de configurare (la nivel de regiștri), mod de conectare al senzorilor și motivarea acestuia prin schematics;
  - \* Reprezentare grafică a modului de conectare al senzorilor și componentelor electronice pe un breadboard virtual folosind platforma web **Tinkercad**;
  - \* **Setup** - înșiruirea pașilor care trebuie urmăriți pentru rularea proiectului însotiti de printscreens-uri;
  - \* **Probleme întâmpinate** - modul de rezolvare al problemelor întâmpinate și sugestii de îmbunătățire a proiectului rezultat.

- Senzori

- Preluarea datelor de la senzorii asignați temei de proiect, fie prin intermediul perifericului **ADC** în cazul senzorilor analogici, fie prin intermediul perifericului **GPIO** în cazul senzorilor digitali;
- *Utilizarea unui senzor nou în cadrul proiectului reprezintă un **BONUS**. Proiectarea unei componente utile în cadrul proiectului pentru noul senzor;*
- Transmiterea datelor obținute de senzori de la platforma de dezvoltare, pe portul serial către PC, cu ajutorul perifericului **UART**.

- Aplicația host

- Realizarea unei interfețe grafice cu ajutorul limbajului Python, pornind de la scheletul furnizat pe repository-ul laboratorului (**MCULabs - Schelet GUI**), care să prezinte un grafic (barplot) în timp real pentru fiecare senzor de input asociat temei de proiect. Un grafic va afișa valorile în 3 zone de culori, **verde** pentru valori mici, **galben** pentru valori medii și **roșu** pentru valori mari. Cele două praguri, i.e., valori mici-medii, valori medii-mari, vor fi stabilite dinamic interpolând în 3 zone spectrul de valori generate de senzorul respectiv;
- Butoane grafice pentru a trimite comenzi către MCU;
- Alte ferestre pentru a afișa informații utile.

- Aplicația target

- Fiecare echipă va avea de implementat un proiect pornind de la un schelet comun, urmând a fi particularizate configurațiile conform cerințelor de proiect (teme.zip).
- Suta minimă de periferice utilizate:
  - \* **UART** - comunicație serială asincronă;
  - \* **GPIO** - general purpose input/output, lucrând cu LED-ul RGB on-board;
  - \* **PIT** - întreruperi la perioade specifice de timp;
  - \* **ADC** - achiziția datelor de la un senzor analogic;
  - \* **PWM** - controlul motoarelor electrice, management eficient al energiei.
- Prin intermediul unui buton din interfața grafică, va fi controlată direcția secvenței de aprindere a LED-urilor RGB de pe FRDM-KL25Z.

**! Deadline implementare/specificație proiect = 17.01.2024, 22:00 !**

**Documentația** va fi încărcată pe moodle (lucrare "Proiect de semestru" în cursul A4-S1-MICROP).

Documentația va avea o nouă secțiune care va conține link-uri accesibile pentru repo-ul proiectului (Github) precum și pentru prezentările video.

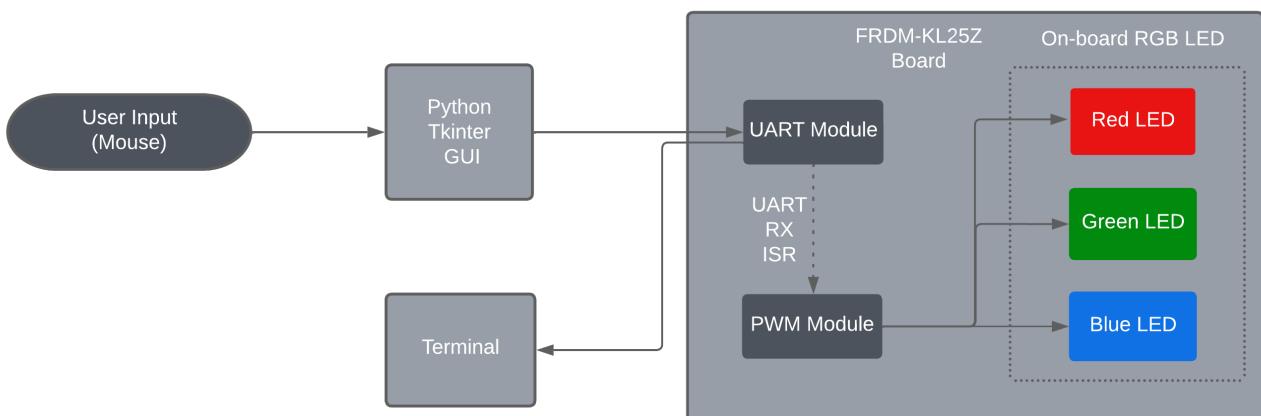


Figure 1: Exemplu de schema bloc a unui sistem

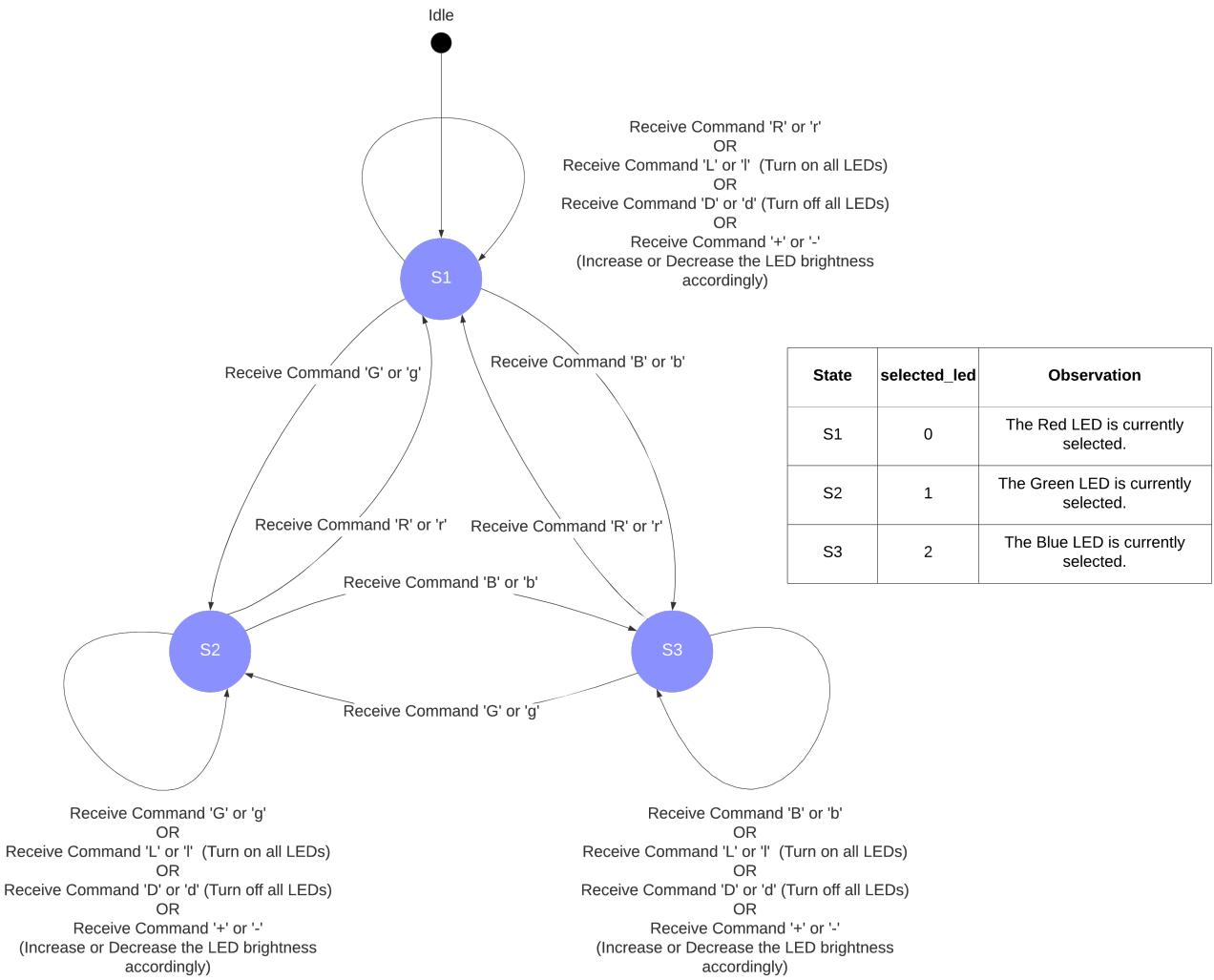


Figure 2: Exemplu diagramă de stări a unui sistem

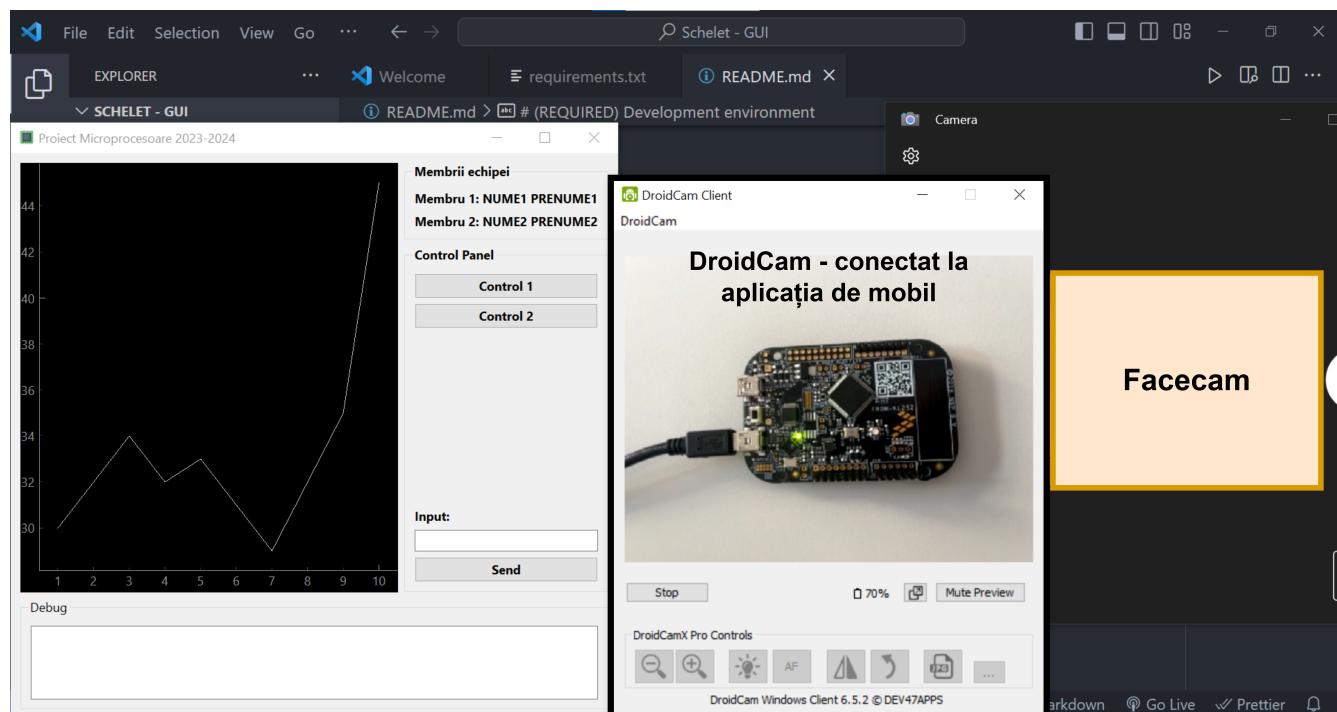


Figure 3: Mod de realizare a înregistrării video-ului în vederea explicării proiectului