

Project Description – Embedded Firmware FSM

1. Project Title | عنوان پروژه

Embedded Firmware Development for Finite-State Machine (FSM) Control

توسعه نرم افزار Embedded برای کنترل ماشین حالت محدود (FSM)

2. Introduction | مقدمه

The aim of this project is to develop an embedded firmware for a microcontroller-based Finite-State Machine (FSM) system. The project covers design, implementation, testing, and simulation, providing a full lifecycle experience in embedded software development.

هدف این پروژه توسعه یک نرم افزار Embedded برای یک سیستم مبتنی بر میکروکنترلر با استفاده از FSM است. پروژه شامل طراحی، پیاده سازی، تست و شبیه سازی بوده و تجربه کامل چرخه توسعه نرم افزار embedded را ارائه می دهد.

3. Tools & Environment | ابزارها و محیط توسعه

- Programming Languages** | زبان ها: C (Embedded), C++, Python
- Microcontroller** | میکروکنترلر: STM32
- IDE & Tools** | محیط توسعه: VS Code, CubeIDE, MATLAB/Simulink, Git/GitHub
- Version Control** | مدیریت نسخه: Git
- Simulation & Debugging** | شبیه سازی و دیباگ: Python, MATLAB, off-target debugging

4. Design & Architecture | طراحی و معماری

- The firmware is modular:
 - main.c handles the main program flow
 - fsm.c and fsm.h implement the FSM logic
- FSM manages states and transitions according to defined triggers
- The system is designed for maintainability and extensibility
- نرم افزار به صورت ماژولار طراحی شده:
 - جریان اصلی برنامه را کنترل می کند main.c
 - را پیاده سازی می کنند FSM منطق fsm.c و fsm.h
- وضعیت ها و انتقال ها را بر اساس ورودی های مشخص مدیریت می کند FSM
- سیستم برای قابلیت نگهداری و توسعه طراحی شده است

5. Features & Functionality | قابلیت ها

- Finite-State Machine implementation with multiple states (Idle, Processing, Error, Complete)
- Interrupt-based input handling
- Hardware-near programming for peripheral control
- Debugging and performance optimization
- Unit testing and firmware validation
- Data visualization using Python

- FSM پیاده‌سازی (Idle، Processing، Error، Complete) با چند وضعیت
- Interrupt مدیریت ورودی‌ها با
- برنامه‌نویسی نزدیک به سخت‌افزار برای کنترل پریفرال‌ها
- دیباگینگ و بهینه‌سازی عملکرد
- Firmware و اعتبارسنجی (Unit Test) تست واحد
- Python تحلیل و نمایش داده‌ها با

6. پیاده‌سازی | Implementation

- Developed firmware modules in **C** for the microcontroller
- FSM implemented using **Finite-State Machine** logic
- Interrupts and register-level programming for input/output
- Python scripts for **simulation, testing, and output visualization**
- Versioning and collaboration handled via Git/GitHub
- برای میکروکنترلر C توسعه ماژول‌های نرم‌افزار در
- با منطق ماشین حالت محدود پیاده‌سازی شد FSM
- و برنامه‌نویسی در سطح رجیستر برای ورودی/خروجی Interrupt استفاده از
- برای شبیه‌سازی، تست و نمایش خروجی‌ها Python اسکریپت‌های
- Git/GitHub مدیریت نسخه و همکاری از طریق

7. نتایج | Results

- Successfully verified FSM transitions under all defined conditions
- Generated simulation outputs and debugged firmware errors
- Python plots visualize state changes and system behavior
- Unit tests confirm stability and correct functionality
- تحت تمام شرایط تعریف‌شده با موفقیت تست شد FSM انتقال‌های
- رفع شد Firmware خروجی‌های شبیه‌سازی تولید و خطاهای
- تغییر وضعیت و رفتار سیستم را نمایش می‌دهند Python نمودارهای
- پایداری و عملکرد صحیح سیستم را تایید کردند (Unit Test) تست‌های واحد

8. دستاوردها | Lessons Learned

- Embedded FSM design and implementation
- Hardware debugging and optimization
- Version control and project management using Git
- Integration of simulation tools with firmware development
- Improved understanding of Python for embedded system analysis
- Embedded در طراحی و پیاده‌سازی FSM
- دیباگینگ و بهینه‌سازی نزدیک به سخت‌افزار
- مدیریت نسخه و پروژه با Git
- Firmware ادغام ابزارهای شبیه‌سازی با توسعه
- Embedded برای تحلیل سیستم‌های Python افزایش مهارت در استفاده از

9. منابع | References

- GitHub Repository: [<https://github.com/ELHAMsh1988/embedded-firmware-fsm->]
- Tutorials: Faradars (C++ & Python), MATLAB documentation
- Microcontroller Datasheets and Reference Manuals

- مخزن GitHub: [\[https://github.com/ELHAMsh1988\]](https://github.com/ELHAMsh1988)
- MATLAB ، مستندات (Faradars (C++ & Python): آموزش‌ها
- دیتاشیت و راهنمای میکروکنترلر

FSM Diagram / نمودار FSM



Simulation Output / سازی شبیه خروجی

```

Current State: IDLE
Transition: IDLE -> RUNNING
Current State: RUNNING
Transition: RUNNING -> IDLE
Current State: IDLE
  
```

State/ مرحله	(Inputs)/ ورودی‌ها	(Outputs) / خروجی‌ها 	Remark / توضیحات
INT	Power ON	LED OFF	سیستم آماده‌سازی اولیه
STATE_1	Sensor HIGH	LED GREEN	ماشین در حالت عملیاتی ۱
STATE_2	Sensor LOW	LED YELLOW	ماشین در حالت آماده باش
STATE_3	Error Detected	LED RED	خطا شناسایی شد، سیستم هشدار می‌دهد

RESET	Reset Button	LED OFF	بازگشت به حالت اولیه
-------	--------------	---------	----------------------