

# 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 مازولار طراحی شده:
  - جریان اصلی برنامه را کنترل می کند
  - را پیاده سازی می کند **FSM** منطق
- وضعیت ها و انتقال ها را بر اساس ورودی های مشخص مدیریت می کند
- 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

- با چند وضعیت FSM (Idle, Processing, Error, Complete) پیاده‌سازی
- مدیریت ورودی‌ها با Interrupt
- 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. Implementation | پیاده‌سازی

- 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 مازول‌های نرم‌افزار در
- با منطق ماشین حالت محدود پیاده‌سازی شد
- 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. Results | نتایج

- 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 تعریف شده با موقبیت 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 تأیید کرند (Unit Test) تست‌های واحد

## 8. Lessons Learned | دستاوردها

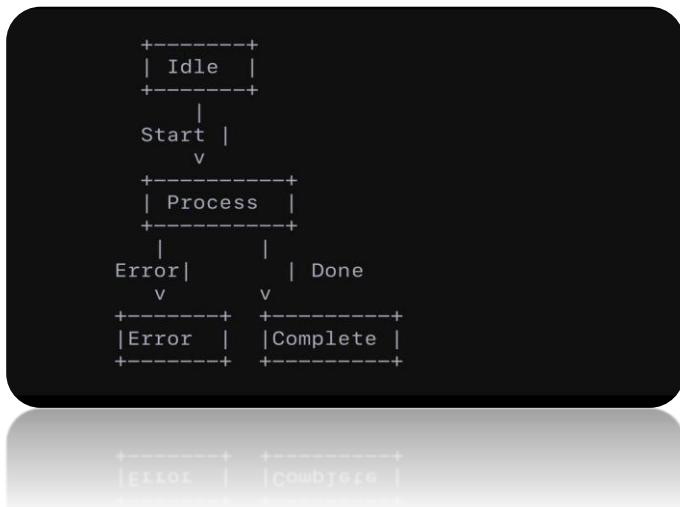
- Embedded FSM design and implementation
- Hardware debugging and optimization
- Version control and project management using Git
- Integration of simulation tools with firmware development
- Improved understanding of Python for embedded system analysis
- در FSM Embedded در طراحی و پیاده‌سازی
- دیباگینگ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

## 9. References | منابع

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

- GitHub مخزن: <https://github.com/ELHAMsh1988>
  - مستندات MATLAB ، آموزش‌ها: Faradars (C++ & Python)
  - 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
  
```

Current state: IDLE

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

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