CSE211/CSE322: Introduction to Embedded Systems

Fall 2025

Project Description

Door Locker Security System

Project Objectives

The objective of this project is to design and implement a Door Locker Security System using two TM4C123 (TivaC) microcontroller-based ECUs. The system integrates password-based access control, inter-ECU UART communication, motor control, alarm management, and persistent configuration storage. This project aims to give students hands-on experience in:

- Layered embedded software architecture (MCAL, HAL, Application)
- Inter-microcontroller communication using UART
- Persistent data storage using EEPROM
- Timer-based control (e.g., for door actuation, lockout timing)
- Security feature implementation (e.g., password handling, lockout)
- Systematic software testing and documentation
- Professional engineering practices (e.g., coding standards, resource measurement)

Students will work in teams of **10 members**, organized into sub-teams handling driver development, application logic, testing, documentation, and code quality.

Project Overview

The Door Locker Security System allows entry only to users who provide the correct password. The system is split into two microcontroller-based ECUs:

- HMI_ECU (Human-Machine Interface ECU):
  Responsible for interacting with the user via an LCD, keypad, potentiometer, and RGB LED.
- Control_ECU:
  Responsible for making decisions based on user input, controlling the door (via motor), managing EEPROM storage, and activating the alarm when necessary.

Key Features:

- Password-based access control
- EEPROM-based persistent storage
- UART-based communication between ECUs
- Motor control for door locking/unlocking
- Alarm system using a buzzer
- Timeout setting via a potentiometer
- Visual status indication using RGB LEDs

Hardware Requirements

HMI_ECU (User Side):

- 16x2 LCD for user messages and menus
- 4x4 Keypad for input
- Potentiometer (with ADC) for timeout configuration
- RGB LED for status indication
- UART interface for communication with Control_ECU
- SysTick/Timers for delays and debouncing

Control_ECU (Control Side):

- EEPROM for storing password and configuration
- DC Motor for door actuation (via PWM)
- Buzzer for alarm signaling
- UART interface for HMI_ECU communication
- GPTM timers for motor/buzzer control
- UART and timer-based interrupts

Functional Requirements

Step 1 – Initial Password Setup

- LCD displays: *"Enter Password"*
- User enters a 5-digit password (displayed as *)
- Password is confirmed and saved to EEPROM on Control_ECU
- If entries do not match, process is repeated

Step 2 – Main Menu

Displayed on LCD:

- + → Open Door
- - → Change Password
- * → Set Auto-Lock Timeout

Step 3 – Open Door

- User is prompted for password
- If correct:
  - Motor rotates to unlock
  - Waits for timeout (5–30 seconds)
  - Motor relocks the door
  - LCD shows appropriate messages
- If incorrect:
  - Up to 3 attempts allowed
  - On 3rd failure: buzzer sounds for (n) seconds, system enters lockout
  - After lockout, return to main menu

Step 4 – Change Password

- Prompt: *"Enter Old Password"*
- If correct: repeat initial password setup
- If incorrect: up to 3 attempts, then buzzer + lockout

Step 5 – Set Auto-Lock Timeout

- LCD displays: *"Adjust Timeout"*
- Potentiometer is used to select value (5–30 seconds), shown live on LCD
- User selects "Save"
- Prompt: *"Enter Password"*
- If correct: value saved to EEPROM and used in next door operation

Non-Functional Requirements

Code Quality

- Follow a coding standard (e.g., MISRA C or CERT C, etc.)
- At least 5 violations must be documented with before/after fixes

Resource Analysis

- Measure and report usage of ROM, RAM, and stack
- Explain methodology (e.g., map file, debugger, runtime tools)

Testing

- Testing team must develop independent test code
- Testing must include:
    - Unit tests for drivers
    - Integration testing
    - Full system functional tests
- Results must be auto-logged in pass/fail format

Self-Grading

- Each team completes a self-assessment matrix

Project Deliverables (Week 13)

1. **Fully Functional Prototype**

   Working Door Locker System with all features
2. **Source Code**

   Organized using software layers (MCAL, HAL, Application) and compliant with chosen coding standard
3. **Final Report**

   Must include:
   - System architecture and layered design
   - Testing strategy and results
   - Resource usage analysis
   - Code quality analysis with standard compliance
   - Completed self-assessment matrix
4. **Live Demonstration**

   A working demo of the system during the evaluation session

Grading Rubric (25 Marks)

Criteria	Marks	Details
1) Functional Requirements	16	
- Password Setup & Storage	3	Initial setup, confirmation, and EEPROM persistence
- Open Door Functionality	3	Motor control, timeout handling, and user feedback via LCD
- Wrong Password Handling	3	Up to 3 attempts, buzzer activation, and lockout mode & recovery
- Change Password	2	Old password required, new password confirmation and storage
- Auto-Lock Timeout Setting	2	Timeout via potentiometer, password to save & EEPROM storage
- LED Feedback	1	LED indicates system state
- UART Communication	2	Reliable communication between boards
2) Non-Functional Requirements	6	
- Code Quality & Standards	2	Coding standard followed, 5 violations documented and resolved
- Resource Analysis	2	ROM, RAM, and stack usage measured and methodology explained
- Testing Implementation	2	Includes unit, integration, and functional testing with result logging
3) Software Design & Report	3	
- Layered Architecture	1	Clear separation into MCAL, HAL, and Application layers
- Final Report	2	Well-structured and complete with required sections