Description:

You supposed to design and implement a simple ATM machine and handle simple flows of a transaction.

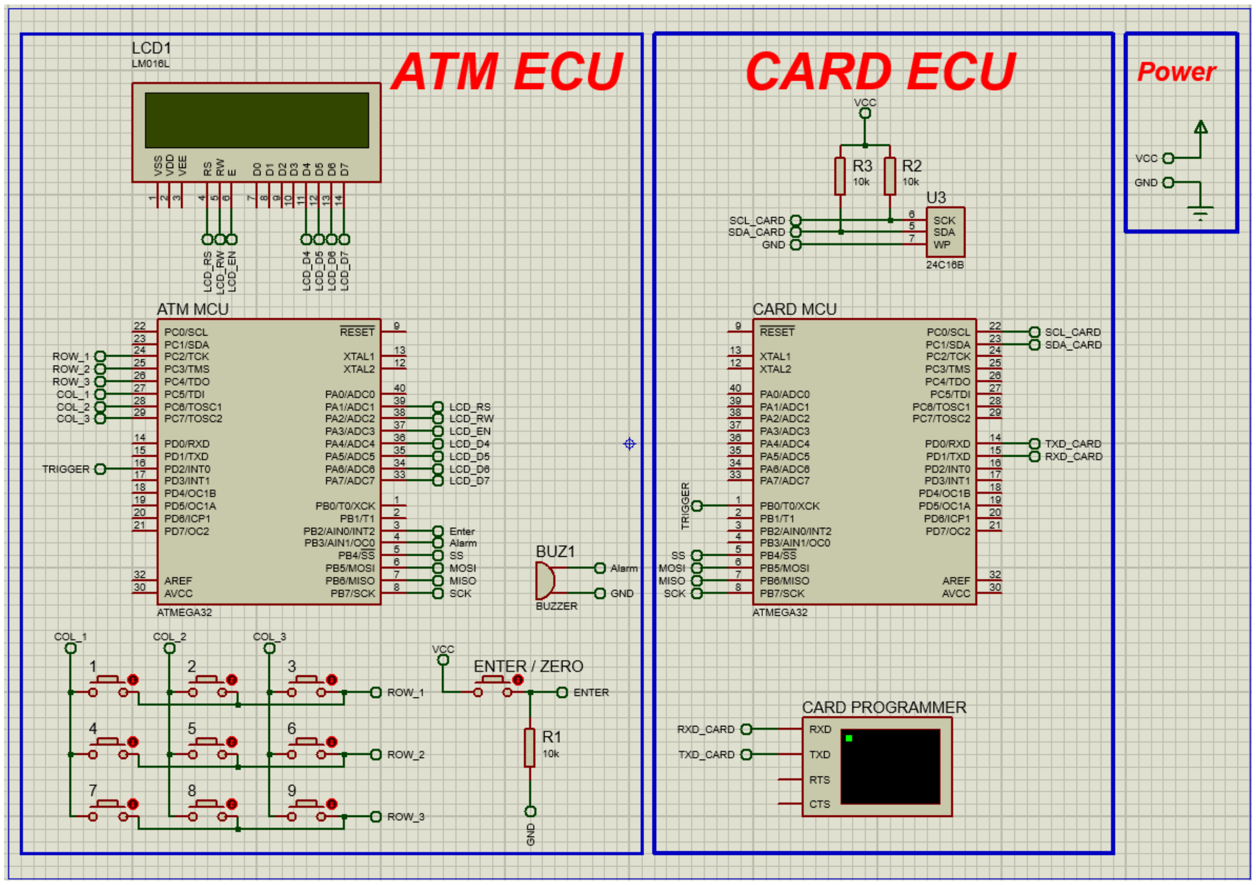

Figure 1: Simple ATM Machine

Detailed Requirements

1. **Create a backlog for the team**
    1. Create an excel sheet named **Team Backlog** that contains the below columns
        1. Task Name
        2. Assignee
        3. Task Status
        4. Expected time to finish
        5. Actual time to finish
2. **Read System Requirement Specifications**
    1. **Hardware Requirements**:
        1. ATM ECU

        1. ATM MCU
        2. 16 x 2 LCD
        3. 3 x 3 Keypad
        4. Buzzer
        5. Enter/Set Button
    2. CARD ECU
        1. CARD MCU
        2. EEPROM
        3. Serial Terminal (Use Putty on your PC)
2. **Software Requirements**:
    1. ATM MCU
        1. This MCU will handle transaction main flows
        2. After Reset
            1. Welcome message is displayed for 1s “Welcome to ATM”
            2. “Insert a Card” message is displayed in the first line
            3. No action can be taken further and all other input devices are blocked until a trigger signal came from the CARD ECU
        3. After a trigger signal is received from the CARD ECU
            1. “Enter Your PIN” message is displayed in the first line
            2. Waiting for the input from the keypad and type it in ‘****’ format in the second line
            3. PIN is only four numeric characters
            4. Pressing the Enter/Zero button for 2s will initiate a communication between the ATM ECU and the CARD ECU to validate if the PIN is correct or not
            5. If the PIN is not correct, repeat for further 2 trials, and then if it is still wrong, sound the alarm and lock every input in the ATM, this blocking can be revealed by hard reset.
            6. If the PIN is correct, then “Enter Amount” message is displayed in the first line and wait for the amount to be entered from the keypad and appeared in the second line
            7. Amount is a float string with max 4 integer digits and 2 decimal digits ”0000.00”
            8. You can enter ‘0’ when pressing Enter/Zero button for less than 2s
            9. After entering the amount to withdraw, several checks on the database are done to finalize the transaction
                1. Check if there is an account attached to this card
                2. Check if the card is blocked or not
                3. Check if the amount required exceeds the maximum daily limit or not
                4. Check for available amount

            10. If one of the checks failed, a declined message will appear accordingly
                1. “This is a fraud card” – if the card PAN is not found – Alarm will be initiated
                2. “This card is stolen” – if the card is blocked– Alarm will be initiated
                3. “Maximum limit is exceeded” – if the required amount exceeds the maximum allowed limit
                4. “Insufficient fund” – if the balance is lower than the required amount
            11. If all checks are passed then “Approved Transaction” message is displayed for 1s and the remaining balance is displayed for 1s “Remaining Balance: 0000.00”
            12. After the checks are done and messages are displayed, display “Ejecting Card” message for 1s
            13. Repeat from the after reset again
        4. Data base
            1. The data base will be hard coded array of structures for accounts that contains (PAN, Account State (blocked/running), and balance)
            2. The maximum allowed limit will be hardcoded “5000.00”
2. CARD MCU
    1. The CARD MCU has two modes of operations
        1. Programming Mode
            1. The CARD MCU will enter this mode after reset
            2. For the first time only the MCU will send the following messages to the terminal
                1. “Please Enter Card PAN:” and wait for the PAN
                2. “Please Enter New PIN:” and wait for the PIN
                3. “Please Confirm New PIN:” and wait for the PIN
                4. If PIN is matched, then change to user mode
                5. If PIN is not matched, not numeric, and exceeds 4 characters, then “Wrong PIN” message is displayed and repeat from step 2
            3. For any further after resets
                1. “Please press 1 for entering user mode and 2 for programming mode: ” message is sent

            to the terminal and wait for a valid response, only accepts 1 or 2
        4. PAN is 16 to 19 length numeric string
        5. PIN is 4 numeric digits
        6. All data taken will be stored in the EEPROM
    2. User Mode
        1. The CARD MCU will enter this mode
            1. After completing the programming mode
            2. Or after choosing 2 in any further after resets
        2. In this mode, the CARD ECU will send a trigger signal to the ATM ECU that will make the ATM initiate its flow

3. **Prepare your design**
    1. Please note that any functionality based on timers should be separated in a separate module, and all timers should be operating in **Normal mode**
    2. Create a PDF file with the name **Simple ATM Machine Design**
    3. The design document should contain the below fields
        1. Cover Page
        2. Table of content
        3. Project introduction
        4. High Level Design for both ECUs
            1. Layered architecture
            2. Modules Descriptions
            3. Drivers' documentation
        5. Low Level Design
            1. Provide the flowchart for each function in each module
4. **Preparing development environment**
    1. Create layer's folders
        1. Create a folder for each layer
        2. All folders should be in **upper case**
        3. Ex: **MCAL**, **HAL**, **APP**, … etc
    2. Create diver's folders and files
        1. Create a folder for each driver
            1. Each folder contains **only one .c** file and **at least one .h** file
            2. All files names should be in **lower case**
        2. All driver folders names should be in **lower case**
        3. Ex: **dio**, **timer**, **pwm**, … etc.
    3. Add header file guard
        1. All header files must include the header file guard
5. **Drivers implementation and code convention**
    1. All drivers provided in the design document should be implemented
    2. All drivers should be tested against different test cases

3. Function’s descriptions should be included
4. Don’t use magic numbers, use Macros or Enums instead
5. Follow a proper indentation in your code
6. Use a meaningful name for your variables
7. Follow the below naming for the functions
    1. MODULENAME_functionName
8. Follow this convention for naming variables
    1. typeIndicator_scopeIndicator_variableName
    2. typeIndicators (u8, u16, u32, i8, i16, st (struct), en (enum), arr (array), .. etc)
    3. scopeIndicators (g (global), gs (global static), a (argument))

6. **Implement and integrate the main application for both ECUs**
    1. Implement the main application that fulfil the system requirements for both ECUs
7. **Test your application**
    1. Create an excel sheet named **Test Protocol**
    2. The sheet should contain the below columns
        1. Test Case ID
        2. Test Case Description
        3. Test Case steps
        4. Expected Result
        5. Actual Result
        6. Pass/Fail
    3. Fill in the sheet with the test cases you will execute
    4. Execute the test cases on **simulator** (**Mandatory**)
    5. Execute the test cases on **hardware** (**Mandatory**)
    6. Test the following use cases and user stories
        1. **ATM ECU Test**
            1. Pressing all inputs before getting the trigger signal
            2. Entering Wrong PIN three times
            3. Entering invalid PIN – PIN length
            4. Entering correct PIN
        2. **CARD ECU Test**
            1. Enter PAN > 19 numeric characters
            2. Enter PAN < 16 numeric characters
            3. Enter PAN With alphabetic characters
            4. Enter Correct PAN
            5. Enter PIN > 4 numeric digits
            6. Enter PIN < 4 numeric digits
            7. Enter PIN alphabetic digits
            8. Enter different PINs in confirmation
            9. Enter correct PIN
            10. Enter correct PIN confirmation

3. **Approved User Story – Mention the balance before the transaction**
4. **Declined User Stories**
    1. Fraud card
    2. Stolen card
    3. Max limit exceeded
    4. Insufficient fund

Delivery

1. Deliver the Team Backlog sheet
2. Deliver the Design Document
3. Deliver all project files and folders including the .hex file
4. All code conventions must be followed
5. English Video recording presenting all of your work as a team
    1. The video should be 15 minutes maximum
    2. Each team member should present himself and discuss his role and what did he delivered through the backlog and what test strategy did he/she made to test his/her work
    3. Application testing should be presented by the team coordinator starting from the Test protocol sheet to simulator and/or the hardware
    4. Any limitation or failed test cases should be communicated in the video
    5. Complete approved flow from card programming to approving the transaction must be covered in the video
    6. All worst case scenarios mentioned must be covered in the video