Smart Lock System

Names:

Magy Samuel

Joseph Ashraf

Magy Hamdy

Karin Essam

Peter Nasrat

Marian Ashraf

Simulation:

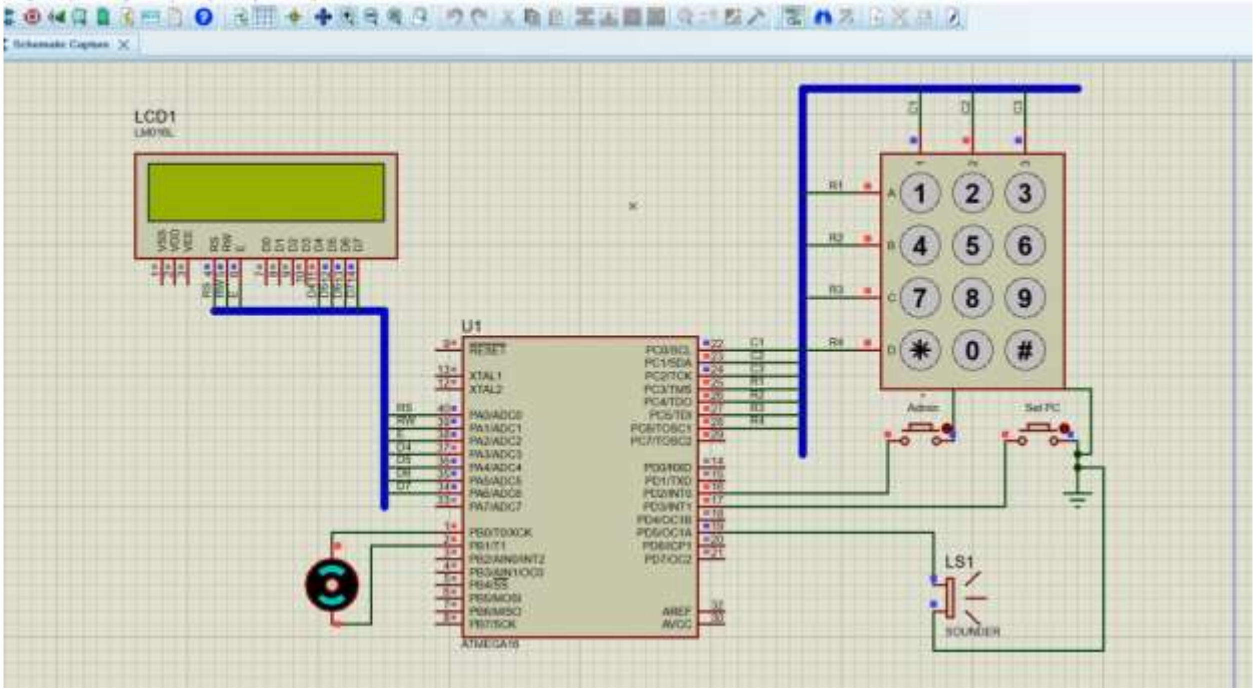


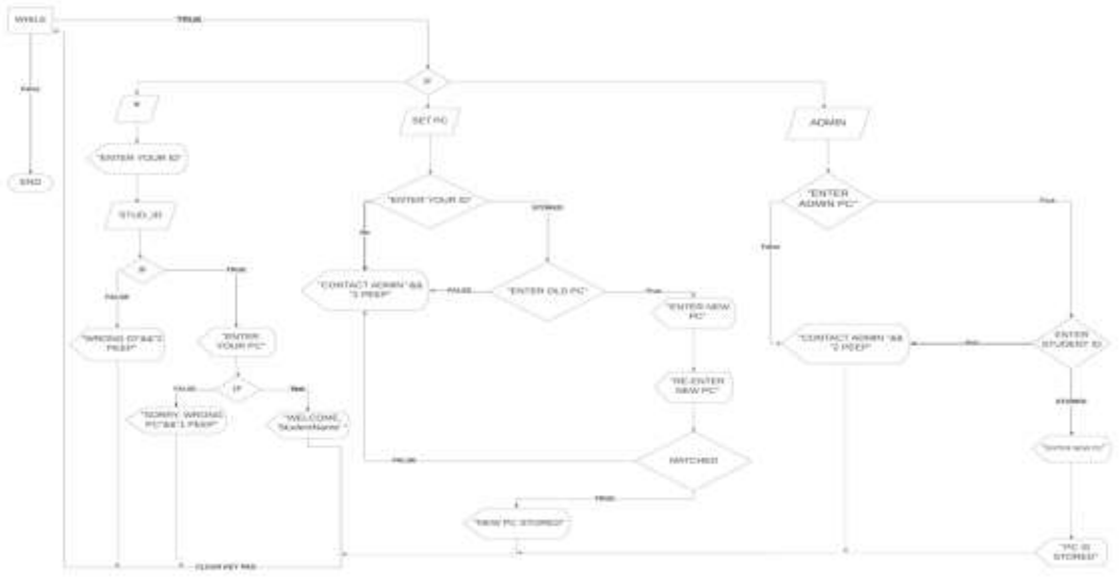
Table Connection:

Port Name	Input or Output	Hardware item
PC0	output	Keypad first column
PC1	output	Keypad second column
PC2	output	Keypad third column
PC3	Input	Keypad first row
PC4	Input	Keypad second row
PC5	Input	Keypad third row
PC6	Input	Keypad forth row
Port A	Output	LCD
PB0	Input - Output	Motor
PB1	Input - Output	Motor
PD5	Output	Sounder
PD2 (INT0)	Input	Admin Push-Pull button
PD3 (INT1)	Input	Set PC Push-Pull button

The needed Hardware items:

1) Atmega16 Microcontroller:
   Central processing unit that executes the code and controls the overall operation of the security system.
2) 16 * 2 LCD Display:
   Used for displaying messages and information to users.
3) Keypad (4*3):
   Input device for users to enter their ID, passcodes, and other information.
4) Motor for locking mechanism:
   A motor that controls the locking mechanism of the door. Connect this to the appropriate pins on the microcontroller for opening and closing the door.
5) Peep Alarm:
   A sounder or buzzer that produces a peeping sound when an error or unauthorized access is detected. Connect this to the microcontroller for controlling the alarm.
6) Admin Push-Pull Button:
   A button used by the admin to trigger admin-related functions.
7) Set PC Button:
   A button used by users to initiate the passcode change process.
8) EEPROM:
   Non-volatile memory for storing user information, IDs, passcodes, and other data.
9) Door Locking Mechanism:
   The physical locking mechanism controlled by the motor.
10) Pull-Up Resistors:
   Resistors used to enable pull-up resistance for input pins.

The System operation flow chart:



The priorities of the used interrupts seem reasonable based on the described functionalities and the criticality of the associated tasks.

Here's a breakdown of the reasons why the priorities are appropriately assigned:

1. **INT0 (Admin push-pull button):**
   - i) Priority Justification
     - (1) Admin functions, such as reprogramming passcodes, are likely to be less frequent but more critical in nature.
     - (2) Admin actions may involve sensitive operations, and granting high priority to this interrupt ensures that these operations are handled promptly.
   - ii) Reasoning:
     - (1) Admin-related tasks are essential for the overall security and management of the system.
     - (2) Admin functions are typically less frequent compared to student entry attempts or passcode changes.
2. **INT1 (Set PC button):**
   - i. Priority Justification:
     - (1) Changing passcodes is important, but it may not be as critical as admin functions.
     - (2) Passcode changes are likely to occur less frequently than entry attempts, and they may not have the same level of urgency.
   - ii) Reasoning:
     - (1) While changing a passcode is a significant action, it is a user-initiated process and is expected to be less frequent than admin-related actions.
     - (2) Lowering the priority of this interrupt aligns with the assumption that passcode changes are less time-sensitive.
3. **Keypad (Polled):**
   - i) Priority Justification:
     - (1) The keypad is regularly polled within the main loop, ensuring consistent monitoring of user input.
     - (2) User interactions, including entry attempts and passcode changes, heavily rely on efficient keypad handling.
   - ii) Reasoning:
     - (1) The keypad is central to user interactions and must be responsive at all times.
     - (2) Since it is not interrupt-driven but regularly polled, it is crucial to keep its processing efficient within the main loop.

Overall Assessment:

- The priorities align with the criticality and frequency of associated tasks.
- Admin functions, being less frequent but more critical, have a higher priority.
- Passcode changes, being less critical and less frequent, have a lower priority.
- The keypad, while not an interrupt, is efficiently handled within the main loop to ensure responsiveness to user input.

These priorities should remain as mentioned, assuming that the system's behavior and requirements align with the assumptions made during the priority assignment.

Missing Actions:

- Admin must be able to add new students.

Code:

LinkToCode