***FATIMA JINNAH WOMEN UNIVERSITY***

******

Computer Organization and  Assembly Language

Project Report

**Group Members:**

ANSA ABID (BCS-009)

HADIYABILA(BCS-017)

MOMINA AWAN(BCS-028)

UMMAMAH LATIF(BCS-032)

SUBMITTED TO : SIR MAJID SHAFIQUE

***Department Of Computer Science***

**Project Report: Home Automation System**

**Overview**

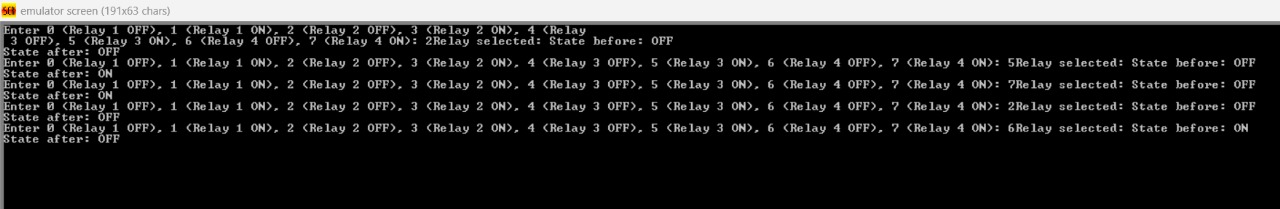
The project is a home automation system implemented in Assembly language for the 8086 microprocessor. The system allows users to control various home appliances via a console interface by inputting commands to toggle relays on and off. Each relay represents a different appliance such as lights, fans, heaters, and pumps. The project demonstrates basic input/output operations, conditional logic, and function usage in assembly language.

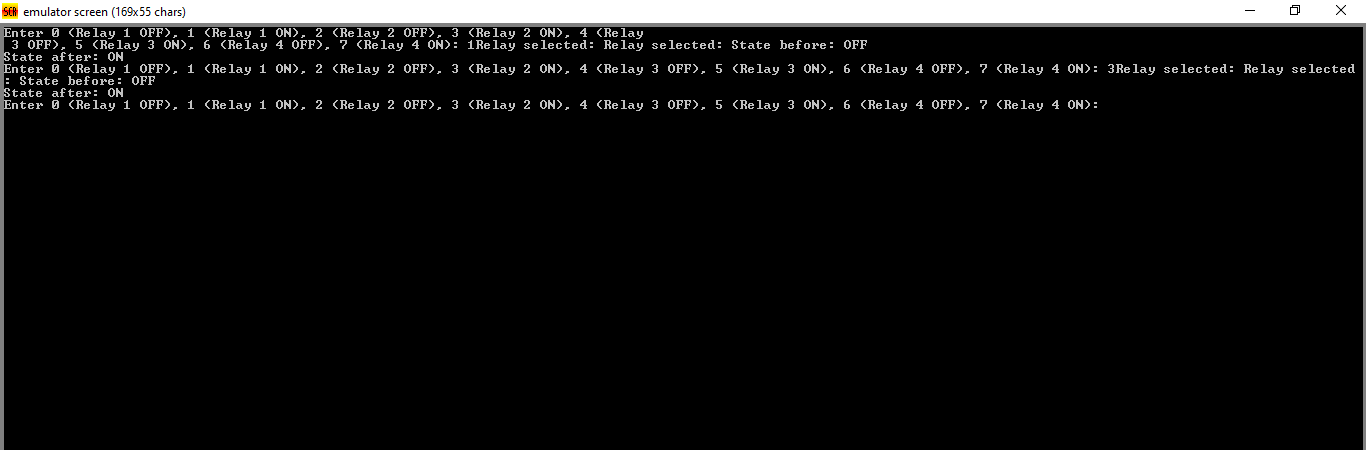
**Components and Functions**

1. **Relays and States**:
   * The project uses four byte variables to represent the state of four relays:
     + stateRelay1 (Light)
     + stateRelay2 (Fan)
     + stateRelay3 (Heater)
     + stateRelay4 (Pump)
   * Each relay can be in one of two states: ON (1) or OFF (0).
2. **Messages**:
   * Several string constants are defined for prompts and relay state messages:
     + promptMsg: Instructions for user input.
     + relayMsg: Message indicating which relay is selected.
     + beforeStateMsg: Message indicating the state before an operation.
     + afterStateMsg: Message indicating the state after an operation.
     + Relay-specific messages for each appliance (e.g., relay1Msg for Light).
     + Messages for ON and OFF states.
     + newline: Carriage return and line feed characters for new lines.
3. **Functions**:
   * PRINT\_STRING: Prints a string to the console.
   * READ\_INPUT: Reads a single character input from the user.
   * DISPLAY\_RELAY\_STATE: Displays the state of a relay.
   * DISPLAY\_UPDATED\_STATE: Updates and displays the new state of a relay.

**Main Program Logic**

1. **Initialization**:
   * The data segment is initialized.
   * All relays are assumed to start in the OFF state.
   * The port address 0x379 is initialized for relay control.
2. **User Interaction Loop**:
   * The program runs an infinite loop that continuously prompts the user for input.
   * Based on the user's input, the program toggles the corresponding relay's state using conditional statements (CMP and JE).
3. **Input Handling**:
   * The program reads a character from the user and executes the corresponding case:
     + '0': Turns Relay 1 (Light) OFF.
     + '1': Turns Relay 1 (Light) ON.
     + '2': Turns Relay 2 (Fan) OFF.
     + '3': Turns Relay 2 (Fan) ON.
     + '4': Turns Relay 3 (Heater) OFF.
     + '5': Turns Relay 3 (Heater) ON.
     + '6': Turns Relay 4 (Pump) OFF.
     + '7': Turns Relay 4 (Pump) ON.
     + Any other input loops back to prompt the user again.
4. **Function Execution**:
   * DISPLAY\_RELAY\_STATE is called to display the current state of the specified relay.
   * The relay state is updated using bitwise operations:
     + Turning off a relay: AND operation to clear the specific bit.
     + Turning on a relay: OR operation to set the specific bit.
   * DISPLAY\_UPDATED\_STATE is called to display the new state of the relay.





### Conclusion

This home automation system project effectively demonstrates the use of basic assembly language constructs such as loops, conditionals, procedures, and I/O operations. The project showcases how to interact with hardware using port I/O operations and manage relay states using bitwise operations. Future enhancements could involve more complex functionalities such as timing controls, integration with sensors, or communication with other device.

**INPUT:**

 If input is '0': Go to **Turn OFF Relay 1**

* If input is '1': Go to **Turn ON Relay 1**
* If input is '2': Go to **Turn OFF Relay 2**
* If input is '3': Go to **Turn ON Relay 2**
* If input is '4': Go to **Turn OFF Relay 3**
* If input is '5': Go to **Turn ON Relay 3**
* If input is '6': Go to **Turn OFF Relay 4**
* If input is '7': Go to **Turn ON Relay 4**
* If input is invalid: Go to **Display Prompt for User Input**

**OUTPUT:**

 **Turn OFF Relay 1:**

* Display Relay 1 State
* Clear bit 0 of port to turn OFF Relay 1
* Update Relay 1 State to OFF
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn ON Relay 1:**

* Display Relay 1 State
* Set bit 0 of port to turn ON Relay 1
* Update Relay 1 State to ON
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn OFF Relay 2:**

* Display Relay 2 State
* Clear bit 1 of port to turn OFF Relay 2
* Update Relay 2 State to OFF
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn ON Relay 2:**

* Display Relay 2 State
* Set bit 1 of port to turn ON Relay 2
* Update Relay 2 State to ON
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn OFF Relay 3:**

* Display Relay 3 State
* Clear bit 2 of port to turn OFF Relay 3
* Update Relay 3 State to OFF
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn ON Relay 3:**

* Display Relay 3 State
* Set bit 2 of port to turn ON Relay 3
* Update Relay 3 State to ON
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn OFF Relay 4:**

* Display Relay 4 State
* Clear bit 3 of port to turn OFF Relay 4
* Update Relay 4 State to OFF
* Display Updated State
* Go to **Display Prompt for User Input**

 **Turn ON Relay 4:**

* Display Relay 4 State
* Set bit 3 of port to turn ON Relay 4
* Update Relay 4 State to ON
* Display Updated State
* Go to **Display Prompt for User Input**