Project Based Learning - 1

Mini Project Title: PC Based Notice Board

Student Names and Roll Numbers:

1. Anjali Singh 36 1032231187
2. Akansha Jain 38 1032231210
3. Aaditi Salunkhe 40 1032231221
4. Shloka Shetty 47 1032231290

Department of Computer Engineering and Technology

School of Computer Science and Engineering

Panel: B

Submission Date: October 24, 2024

Abstract

This project implements a PC-based notice board using an 8051 microcontroller. The system allows users to input messages through a computer interface, which are then displayed on an LCD screen connected to the microcontroller. This digital notice board offers a modern, efficient alternative to traditional bulletin boards, enabling real-time updates and improved information dissemination. The project demonstrates the practical application of microcontroller programming and serial communication in creating user-friendly information systems.

Table of Contents

Introduction

Objective of the Project

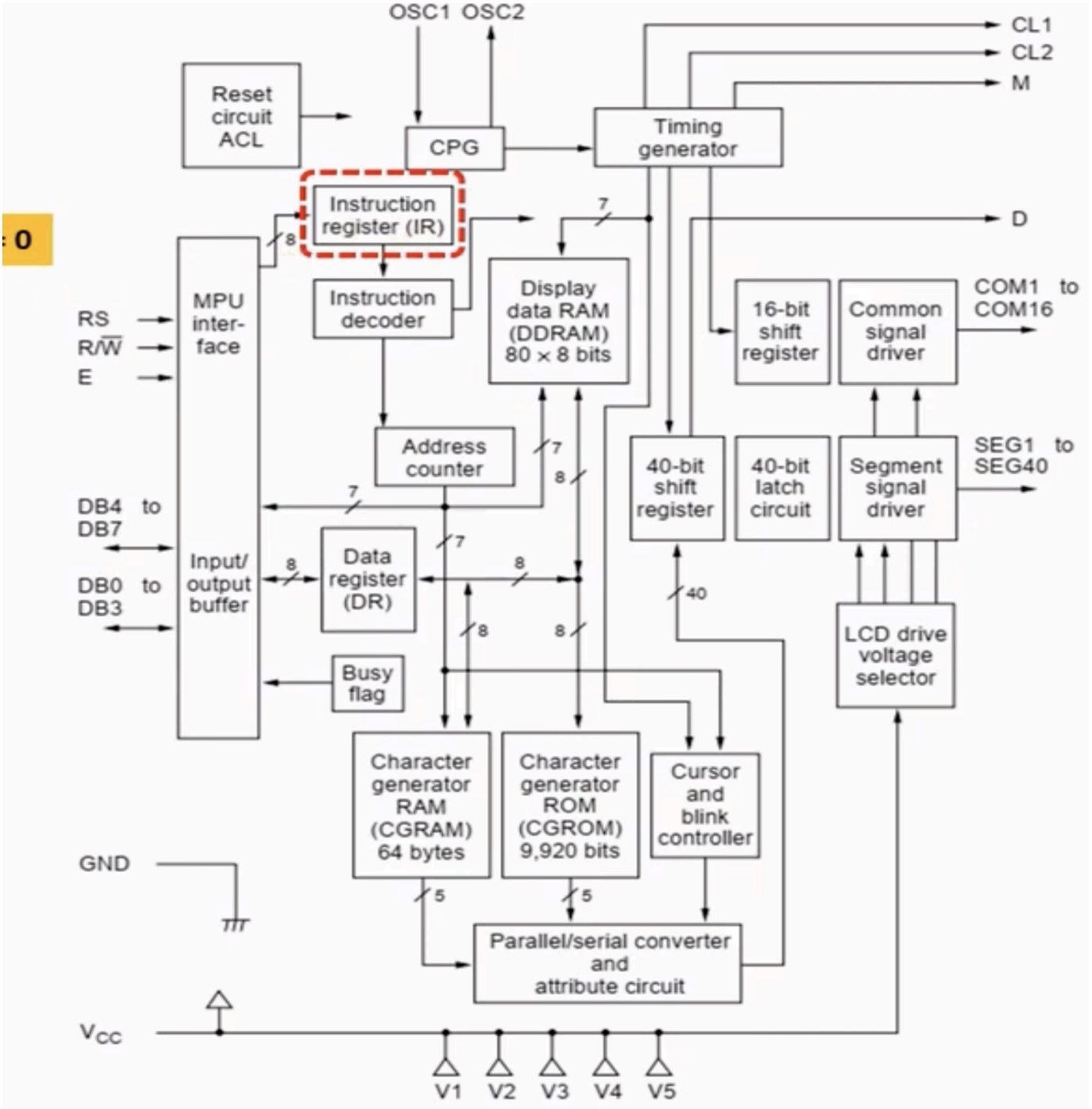
The primary objective of this project is to design and implement a PC-based notice board using an 8051 microcontroller. This system aims to modernize information sharing in various settings such as educational institutions, offices, or public spaces by replacing traditional physical notice boards with a digital, easily updatable alternative.

Project Motivation

In today's fast-paced digital world, the need for quick and efficient information dissemination is crucial. Traditional notice boards are often overlooked and require manual updates, which can be time-consuming and inefficient. By creating a PC-based notice board, we address these issues, providing a solution that allows for real-time updates, increased visibility, and improved information management. This project also serves as an excellent opportunity to apply our knowledge of microcontroller programming and interfacing in a practical, real-world scenario.

Implementation

Block Diagram (Architecture)



Explanation of Components

1. PC Interface: A simple GUI application running on a computer that allows users to input and send messages to the microcontroller.
2. 8051 Microcontroller: The central processing unit that receives data from the PC, processes it, and controls the LCD display.
3. MAX232 IC: Used for converting the PC's RS232 serial communication signals to TTL levels compatible with the 8051 microcontroller.
4. 16x2 LCD Display: Displays the messages received from the PC.
5. Power Supply: Provides the necessary voltage to run the microcontroller and other components.

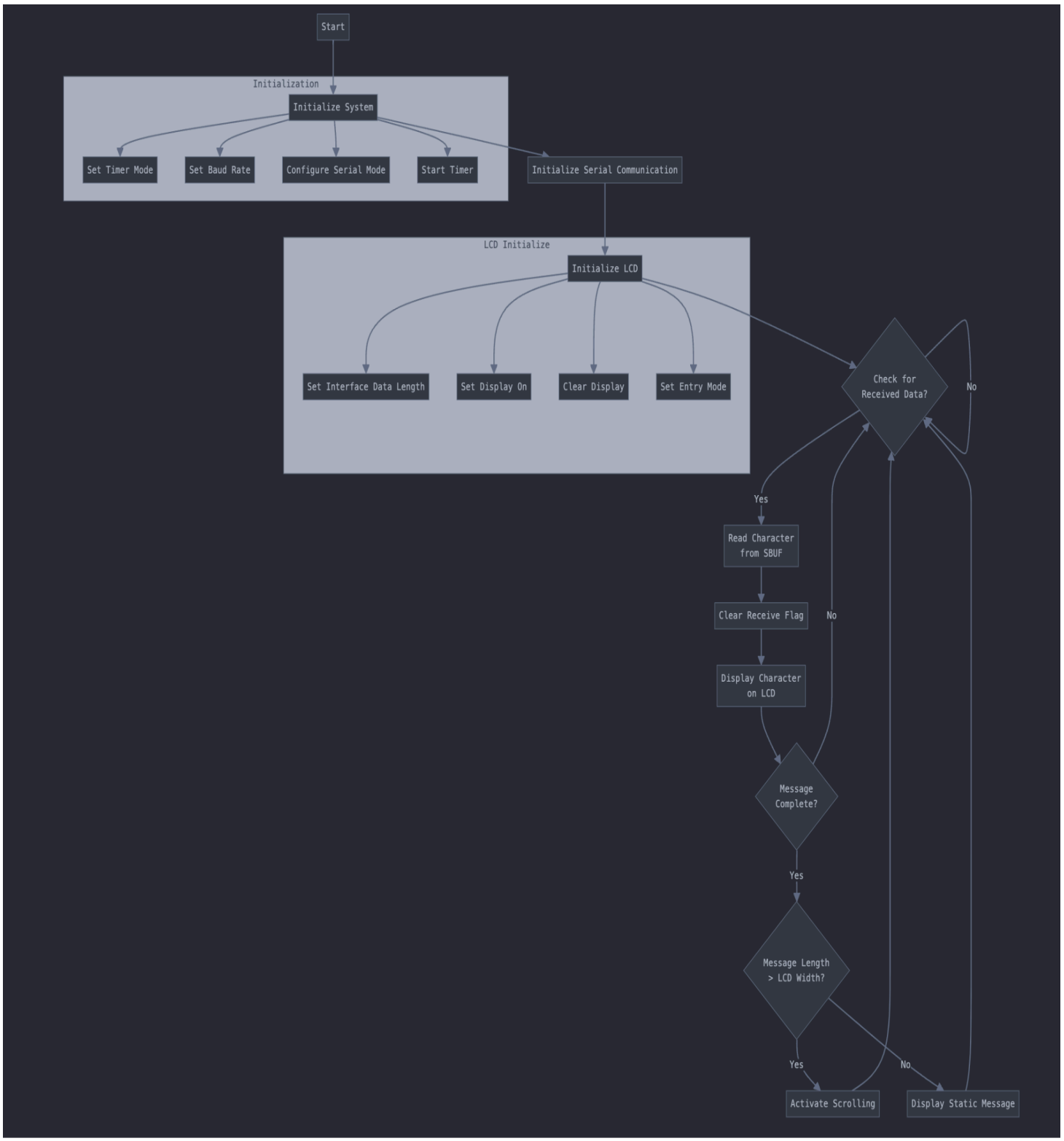
Hardware Specifications

- AT89S52 (8051 family) Microcontroller
- 16x2 LCD Display
- MAX232 IC
- 11.0592 MHz Crystal Oscillator
- 33pF Capacitors (2)
- 10µF Capacitors (4)
- 10KΩ Resistor
- 10KΩ Potentiometer
- Push Buttons (for additional functionality if needed)

Software Specifications

- Edsim51 for assembly programming

Flowchart


Start
Initialization
Initialize System
Set Timer Mode
Set Baud Rate
Configure Serial Mode
Start Timer
Initialize Serial Communication
LCD Initialize
Initialize LCD
Set Interface Data Length
Set Display On
Clear Display
Set Entry Mode
Check for Received Data?
No
Yes
Read Character from SBUF
Clear Receive Flag
No
Display Character on LCD
Message Complete?
Yes
Message Length > LCD Width?
Yes
No
Activate Scrolling
Display Static Message

Implemented Code

```
Org 0000h

RS                      Equ     P1.3
E                       Equ             P1.2
; R/W* is hardwired to 0V, therefore it is always in write mode
; ---------------------------------- Main -----------------------------------
                        Clr RS                  ; RS=0 - Instruction register is selected.
                                                ; Stores instruction codes, e.g., clear
display...
; Function set
                        Call FuncSet
; Display on/off control
                        Call DispCon
; Entry mode set (4-bit mode)
                        Call EntryMode
; Send data
                        SetB RS                 ; RS=1 - Data register is selected.
                                                ; Send data to data register to be displayed.

                        Mov DPTR,#LUT1
Back:           Clr A
                        Movc A,@A+DPTR
                        Jz Next
                        Call SendChar
                        Inc DPTR
                        Jmp Back

Next:           Call CursorPos  ;Put cursor onto the next line

                        SetB RS                 ; RS=1 - Data register is selected.
                                                ; Send data to data register to be displayed.

                        Mov DPTR,#LUT2
Again:          Clr A
                        Movc A,@A+DPTR
                        Jz EndHere
                        Call SendChar
                        Inc DPTR
                        Jmp Again

EndHere:        Jmp $
;---------------------------------- END -----------------------------------
```

```
;------------------------------ Subroutines -------------------------------
; ------------------------ Function set -------------------------------------
FuncSet:        Clr  P1.7                   ;
                Clr  P1.6                   ;
                SetB P1.5                   ; bit 5=1
                Clr  P1.4                   ; (DB4)DL=0 - puts LCD module into 4-bit mode

                Call Pulse

                Call Delay                  ; wait for BF to clear

                Call Pulse

                SetB P1.7                   ; P1.7=1 (N) - 2 lines
                Clr  P1.6
                Clr  P1.5
                Clr  P1.4

                Call Pulse

                Call Delay
                Ret
;-------------------------------------------------------------------------
;------------------------------ Display on/off control ---------------------
; The display is turned on, the cursor is turned on
DispCon:        Clr P1.7            ;
                Clr P1.6            ;
                Clr P1.5            ;
                Clr P1.4            ; high nibble set (0H - hex)

                Call Pulse

                SetB P1.7                   ;
                SetB P1.6                   ; Sets entire display ON
                SetB P1.5                   ; Cursor ON
                SetB P1.4                   ; Cursor blinking ON
                Call Pulse

                Call Delay                  ; wait for BF to clear
                Ret
;-------------------------------------------------------------------------
CursorPos:      Clr RS
                SetB P1.7                   ; Sets the DDRAM address
                SetB P1.6                   ; Set address. Address starts here - '1'
                Clr P1.5            ;
              '0'
                Clr P1.4            ;
              '0'
                                                ; high nibble
                Call Pulse
```

```
                    Clr P1.7        ;
                '0'
                    Clr P1.6        ;
                '0'
                    Clr P1.5        ;
                '0'
                    Clr P1.4        ;
                '0'
                                                ; low nibble
                                                ; Therefore address is 100 0000 or 40H

                    Call Pulse

                    Call Delay              ; wait for BF to clear
                    Ret
;---------------------------- Entry mode set (4-bit mode) ----------------------
;   Set to increment the address by one and cursor shifted to the right
EntryMode:  Clr P1.7        ; P1.7=0
                    Clr P1.6        ; P1.6=0
                    Clr P1.5        ; P1.5=0
                    Clr P1.4        ; P1.4=0

                    Call Pulse

                    Clr  P1.7               ; P1.7 = '0'
                    SetB P1.6               ; P1.6 = '1'
                    SetB P1.5               ; P1.5 = '1'
                    Clr  P1.4               ; P1.4 = '0'

                    Call Pulse

                    Call Delay              ; wait for BF to clear
                    Ret

;---------------------------------- Pulse ---------------------------------
Pulse:      SetB E          ; P1.2 is connected to 'E' pin of LCD module
                    Clr  E          ; negative edge on E
                    Ret
;--------------------------------------------------------------------------
;---------------------------------- SendChar ------------------------------
SendChar:   Mov C, ACC.7        ;
                    Mov P1.7, C                 ;
                    Mov C, ACC.6            ;
                    Mov P1.6, C                 ;
                    Mov C, ACC.5            ;
                    Mov P1.5, C                 ;
                    Mov C, ACC.4            ;
                    Mov P1.4, C                 ; high nibble
                    ;Jmp $
```

```
                        Call Pulse

                        Mov C, ACC.3        ;
                        Mov P1.7, C                 ;
                        Mov C, ACC.2        ;
                        Mov P1.6, C                 ;
                        Mov C, ACC.1        ;
                        Mov P1.5, C                 ;
                        Mov C, ACC.0        ;
                        Mov P1.4, C                 ; low nibble

                        Call Pulse

                        Call Delay                  ; wait for BF to clear

                        Mov R1,#55h
                        Ret

;------------------------------------ Delay ----------------------------------
Delay:          Mov R0, #50
                        Djnz R0, $
                        Ret
;-----------------------------------------------------------------------------
;------------------------------- End of subroutines ----------------------------
;---------------------------- Look-Up Table (LUT) -----------------------------
                        Org 0200h
LUT1:     DB 'N', 'O', 'T', 'I', 'C', 'E', ' ', 'B', 'O', 'A', 'R', 'D', 0
LUT2:     DB 'P', 'B', 'L', ' ', 'I', 'S', ' ', 'B', 'E','S','T', 0
;-----------------------------------------------------------------------------
Stop:           Jmp $

                        End
```

Conclusion and Result

Test Cases

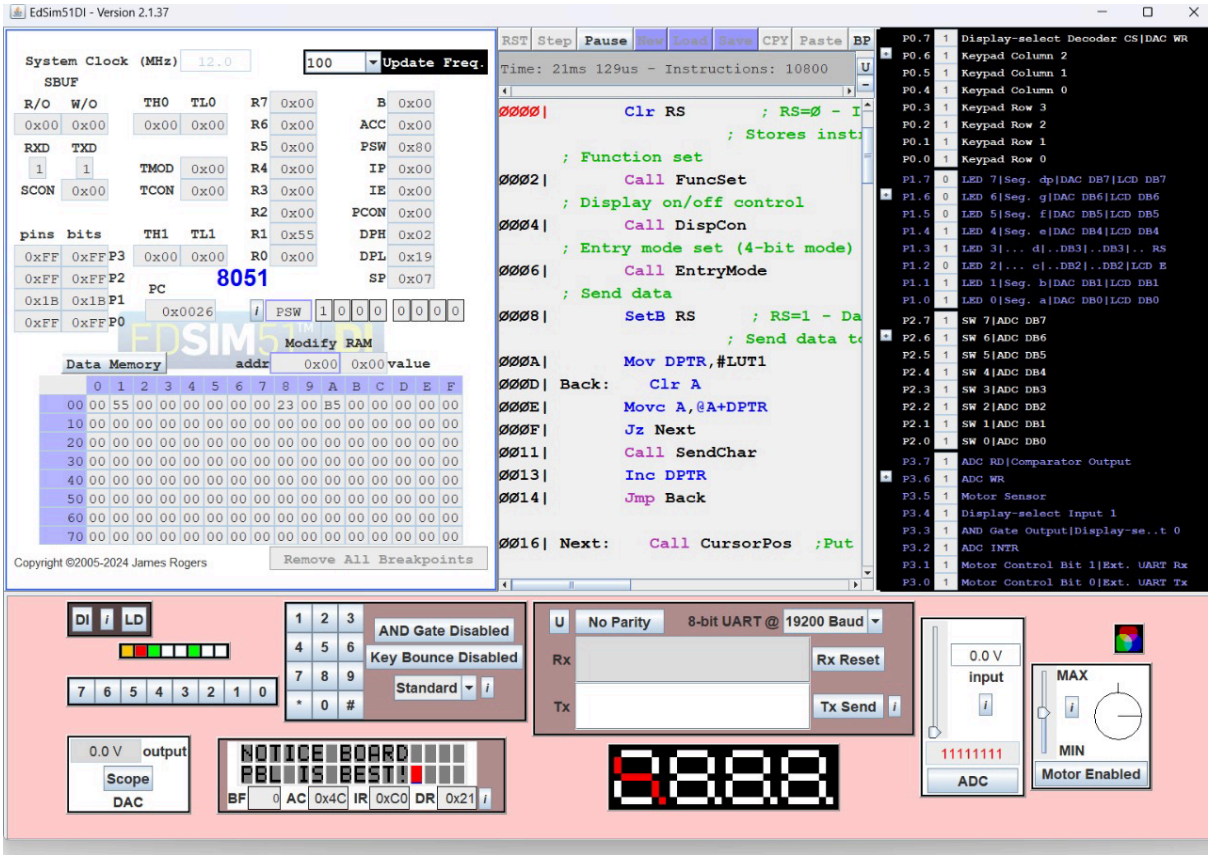
1. Single line message display
2. Multi-line message display
3. Special character handling
4. Message scrolling for long texts

Results

The PC-based notice board successfully receives and displays messages sent from the computer interface. All test cases were executed with the following outcomes:

1. Single line messages are displayed correctly and instantly.
2. Multi-line messages are handled by automatically scrolling or switching between lines.
3. Special characters are displayed without any issues.
4. Long messages scroll smoothly across the LCD screen.
5. The system properly handles and notifies users of invalid inputs.

Screenshots



Conclusion

The PC-based notice board project successfully demonstrates the practical application of 8051 microcontroller programming in creating a modern information display system. By integrating computer interfaces with microcontroller-driven hardware, we have created a versatile and efficient alternative to traditional notice boards.

This project has not only met its primary objectives but also provided valuable insights into serial communication, LCD interfacing, and real-time data processing. The system's ability to quickly update displayed information makes it suitable for various environments where timely information dissemination is crucial.

Future enhancements could include wireless communication capabilities, support for multiple display units, and integration with existing information management systems.

References

1. Mazidi, M. A., & Mazidi, J. G. (2006). The 8051 Microcontroller and Embedded Systems. Pearson Education India.
2. James Rogers (2023). Edsim51’s guide to 8051, core of the popular 51 series 8-bit microcontrollers.
3. Steiner, C. (2005). The 8051/8052 Microcontroller: Architecture, Assembly Language, and Hardware Interfacing. Universal Publishers.
4. Davies, J. H. (2008). MSP430 Microcontroller Basics. Elsevier.
5. "8051 Tutorial." Microcontrollers Lab, www.microcontrollerslab.com/8051-tutorial/. Accessed 11 Oct 2024.
6. "LCD Interfacing with 8051 Microcontroller." ElectronicWings, www.electronicwings.com/8051/lcd-interfacing-with-8051. Accessed 11 Oct 2024.
7. Links followed are listed below

https://edsim51.wordpress.com/wp-content/uploads/2024/08/hd44780.pdf

#32 EdSim51 LCD - Display Two Lines of Text

#28 EdSim51 LCD Module