
Embedded LCD System Documentation

Table of Contents

- Overview
- Features
- Getting Started
 - Prerequisites
 - Installation
 - Proteus Simulation
- Developers
- Main file
- Header file

Overview

This documentation provides details about an Embedded LCD System project developed by a team of developers. The system allows users to print values on an LCD using a keypad.

Features

- Print values on LCD using a keypad.

Getting Started

Prerequisites

Ensure you have the following tools and components:

- Proteus 8 Professional
- CodeVisionAVR Evaluation
- ATmega16 Microcontroller
- Other necessary components (LCD, Keypad)

Installation

1. Clone the repository:

```
1 git clone https://github.com/Hussein119/lcd-system.git
2 cd lcd-system
```

2. Open the project in CodeVisionAVR.

- Launch CodeVisionAVR and open the project file (`\Code\lcd sys Project 2.prj`).
- Customize project settings if necessary.

3. Simulate in Proteus.

- Open Proteus 8 Professional.
- Load the simulation file (`\Simulation\lcd sys Project 2.pdsprj`) and run the simulation.

4. Hardware Implementation.

- Connect the ATmega16 to the necessary components.
- Program the microcontroller using CodeVisionAVR.

Proteus Simulation

Hardware Components

1. ATmega16 Microcontroller
2. LCD Display
3. Keypad 4x3

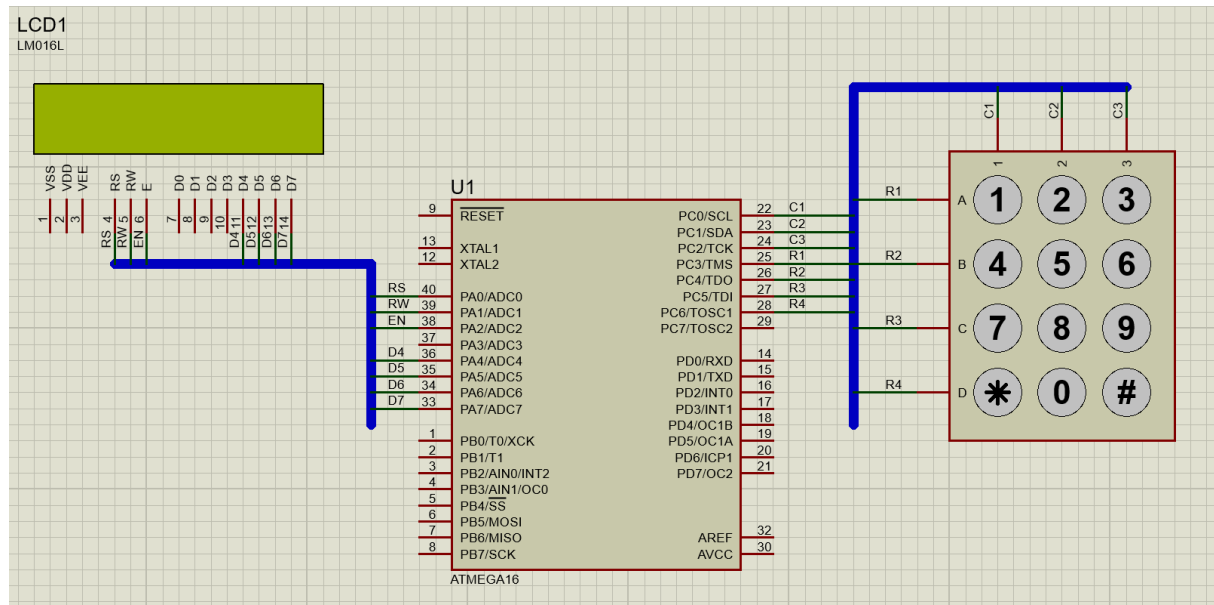


Figure 1: Hardware

Developers

- Islam Abdelhady Hassanein
- Ahmed Hesham Fathall Farag
- Elsherif Shapan Abdelmageed
- Hussein Abdelkader Hussein
- Enas Ragab AbdEllatif
- Mariam Tarek Saad

Main File

lcdSys.c

- lcdSys.c - Main file for the embedded lcd system project.
- This file serves as the main entry point for the lcd system project. It includes necessary header files

```

1  /*
2  * lcd sys Project 2.c
3  *
4  * Created: 12/21/2023 11:42:33 AM
5  * Author: Hos10

```

```

6  */
7
8  #include "SysInit.h"
9  #include <alcd.h>
10
11 void main(void)
12 {
13     // Initialize Hardware
14     initializeHardware();
15
16     while (1)
17     {
18         int i;
19         int input;
20
21         for (i = 0; i <= 32; i++)
22         {
23             input = keypad();
24
25             if (input == 10)
26                 lcd_putchar('*');
27             else if (input == 11)
28                 lcd_putchar('#');
29             else
30                 lcd_printf("%d", input);
31         }
32         lcd_clear();
33     }
34 }

```

Header File

SysInit.h

- lockSysInit.h - Header file containing initialization functions for the embedded lcd system.
- This file includes functions for initializing various hardware components such as the keypad, and LCD.

```

1  #include <mega16.h>
2  #include <alcd.h>
3
4  // Function prototypes
5  void initializeHardware();
6  void initializeKeypad();
7  char keypad();
8
9  // Function to initialize hardware components
10 void initializeHardware()

```

```

11 {
12     initializeKeypad();
13     lcd_init(16); // Initialize the LCD
14 }
15
16 // Function to initialize keypad
17 void initializeKeypad()
18 {
19     // Set keypad ports
20     DDRC = 0b00000111; // 1 unused pin, 4 rows (input), 3 columns (
        output)
21     PORTC = 0b11111000; // pull-up resistance
22 }
23
24 // Function: keypad
25 // Description: Reads the input from a 4x3 matrix keypad and returns
        the corresponding key value.
26 //           The keypad is connected to port C, and the function
        scans each row and column
27 //           combination to determine the pressed key.
28 // Returns: Character representing the pressed key.
29 char keypad()
30 {
31     while (1)
32     {
33         PORTC .0 = 0;
34         PORTC .1 = 1;
35         PORTC .2 = 1;
36
37         switch (PINC)
38         {
39             case 0b11110110:
40                 while (PINC .3 == 0)
41                     ;
42                 return 1;
43             case 0b11101110:
44                 while (PINC .4 == 0)
45                     ;
46                 return 4;
47             case 0b11011110:
48                 while (PINC .5 == 0)
49                     ;
50                 return 7;
51             case 0b10111110:
52                 while (PINC .6 == 0)
53                     ;
54                 return 10;
55         }
56
57         PORTC .0 = 1;
58         PORTC .1 = 0;

```

```
59     PORTC .2 = 1;
60
61     switch (PINC)
62     {
63     case 0b11110101:
64         while (PINC .3 == 0)
65             ;
66         return 2;
67     case 0b11101101:
68         while (PINC .4 == 0)
69             ;
70         return 5;
71     case 0b11011101:
72         while (PINC .5 == 0)
73             ;
74         return 8;
75     case 0b10111101:
76         while (PINC .6 == 0)
77             ;
78         return 0;
79     }
80
81     PORTC .0 = 1;
82     PORTC .1 = 1;
83     PORTC .2 = 0;
84
85     switch (PINC)
86     {
87     case 0b11110011:
88         while (PINC .3 == 0)
89             ;
90         return 3;
91     case 0b11101011:
92         while (PINC .4 == 0)
93             ;
94         return 6;
95     case 0b11011011:
96         while (PINC .5 == 0)
97             ;
98         return 9;
99     case 0b10111011:
100         while (PINC .6 == 0)
101             ;
102         return 11;
103     }
104 }
105 }
```