Task1:

#include <reg51.h>

#include <string.h>

// Function declarations

void delay(int);

void lcdinit(void);

void writecmd(int);

void writedata(char);

sbit E = P3^6; //E pin for LCD

sbit RS = P3^7; //RS pin for LCD

int main(void) {

//lcdinit(); // Initialize LCD

writecmd(0x0C);

writedata('F');

writedata('A');

//writecmd(0x01); // clear display

writedata('a');

writecmd(0xC0);

writecmd(0x38);

writedata('1');

while (1) {

}

}

void delay(int a) {

int i;

for (i = 0; i < a; i++); // Null statement

}

void writedata(char t) {

RS = 1; // This is data

P2 = t; // Data transfer

E = 1; // => E = 1

delay(100);

E = 0; // => E = 0

delay(100);

}

void writecmd(int z) {

RS = 0; // This is command

P2 = z; // Data transfer

E = 1; // => E = 1

delay(100);

E = 0; // => E = 0

delay(100);

}

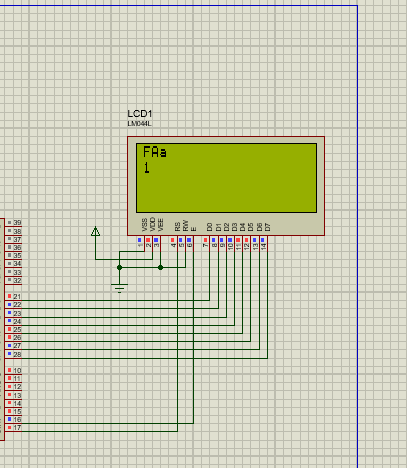
void lcdinit(void) {

delay(15000); // Wait for LCD to power up

writecmd(0x38); // Function set

writecmd(0x0C); // Display on, cursor off, blink off

}



Task2:

#include <reg51.h>

#include <string.h>

// Define Macros

#define Error 13 // Any value other than 0 to 9 is good here

// Function declarations

void cct\_init(void);

void delay(int);

void lcdinit(void);

void writecmd(int);

void writedata(char);

char READ\_SWITCHES(void);

char get\_key(void);

int get\_num(char);

void disp\_num(int);

// Pin description

sbit RowA = P1^0; //RowA

sbit RowB = P1^1; //RowB

sbit RowC = P1^2; //RowC

sbit RowD = P1^3; //RowD

sbit C1 = P1^4; //Column1

sbit C2 = P1^5; //Column2

sbit C3 = P1^6; //Column3

sbit C4 = P1^7; //Column4

sbit E = P3^6; //E pin for LCD

sbit RS = P3^7; //RS pin for LCD

int main(void) {

char key; // key char for keeping a record of pressed key

int num = 0; // Number entered by the user

cct\_init(); // Make input and output pins as required

lcdinit(); // Initialize LCD

while (1) {

key = get\_key(); // Get key from the user

num = get\_num(key); // Get the number from the key (if valid)

if (num != Error) { // If a valid number is entered

disp\_num(num); // Display the number on the LCD

}

/\* if (key == 'C') { // If C/On button is pressed

writecmd(0x01); // Clear display

delay(1000); // Delay for a second after clearing

} else {

num = get\_num(key); // Get the number from the key (if valid)

if (num != Error) { // If a valid number is entered

disp\_num(num); // Display the number on the LCD

}

}

\*/

}

}

void cct\_init(void) {

P0 = 0x00;

P1 = 0xF0; // Used for generating outputs and taking inputs from Keypad

P2 = 0x00; // Used as data port for LCD

P3 = 0x00;

}

void delay(int a) {

int i;

for (i = 0; i < a; i++);

}

void writedata(char t) {

RS = 1; // This is data

P2 = t; // Data transfer

E = 1; // => E = 1

delay(100); // Adjust delay as per LCD's requirements

E = 0; // => E = 0

delay(100); // Adjust delay as per LCD's requirements

}

void writecmd(int z) {

RS = 0; // This is command

P2 = z; // Data transfer

E = 1; // => E = 1

delay(100); // Adjust delay as per LCD's requirements

E = 0; // => E = 0

delay(100); // Adjust delay as per LCD's requirements

}

void lcdinit(void) {

delay(15000); // Wait for LCD to power up

writecmd(0x38); // Function set

writecmd(0x0C); // Display on, cursor off, blink off

writecmd(0x01); // Clear display

// writecmd(0x06); // Entry mode set: increment cursor, no shift

}

char READ\_SWITCHES(void) {

RowA = 0; RowB = 1; RowC = 1; RowD = 1; // Test Row A

if (C1 == 0) { delay(10000); while (C1 == 0); return '7'; }

if (C2 == 0) { delay(10000); while (C2 == 0); return '8'; }

if (C3 == 0) { delay(10000); while (C3 == 0); return '9'; }

RowA = 1; RowB = 0; RowC = 1; RowD = 1; // Test Row B

if (C1 == 0) { delay(10000); while (C1 == 0); return '4'; }

if (C2 == 0) { delay(10000); while (C2 == 0); return '5'; }

if (C3 == 0) { delay(10000); while (C3 == 0); return '6'; }

RowA = 1; RowB = 1; RowC = 0; RowD = 1; // Test Row C

if (C1 == 0) { delay(10000); while (C1 == 0); return '1'; }

if (C2 == 0) { delay(10000); while (C2 == 0); return '2'; }

if (C3 == 0) { delay(10000); while (C3 == 0); return '3'; }

RowA = 1; RowB = 1; RowC = 1; RowD = 0; // Test Row D

if (C1 == 0) { delay(10000); while (C1 == 0); return 'C'; } // Clear screen

return 'n'; // Means no key has been pressed or an invalid key

}

char get\_key(void) {

char key = 'n'; // Assume no key pressed

while (key == 'n') {

key = READ\_SWITCHES(); // Scan the keys until a key is pressed

}

return key; // Return the pressed key

}

int get\_num(char ch) {

switch (ch) {

case '1': return 1; break;

case '2': return 2; break;

case '3': return 3; break;

case '4': return 4; break;

case '5': return 5; break;

case '6': return 6; break;

case '7': return 7; break;

case '8': return 8; break;

case '9': return 9; break;

default: return Error; break; // Invalid input

}

}

void disp\_num(int num) {

writedata(num + '0'); // Display the number on the LCD

}

