Keypad integrated Kannada Character Display

PROJECT REPORT

Department of Computer Science & Engineering

M.S.Ramaiah Institute of Technology

Bangalore

VI semester ECS PROJECT

Prepared by ,

1. **Avinash S (1MS09CS126)**
2. **Nishanth T (1MS09CS131)**

Under the Guidance of
Veena G S

Keypad integrated kannada character display using 8051 microcontroller (AT89C52)

The Kannada character display system is widely needed in places across states where in the local population does not have suitable knowledge of English language. This character display system provides an efficient way to display information in a variety of languages as designed which helps in easy understanding, better interaction and reaching a wide range of people. When integrated with a keypad this system can be readily implemented on the local transportation means in order to cater to a wide variety of people who have limited knowledge of English language. This system when put in place can provide a means for effective communication without the commuter needing to ask directions from anyone around. It is build around 8051 microcontroller (AT89C52).

DESCRIPTION

This Keypad integrated Kannada display system has eight pins connected to the 16*2 LCD screen. The output is displayed on a LCD Screen as foresaid (interfaced with microcontroller), which shows the corresponding output as per the entered program. A running animation is also provided for the text which can be used to signify the respective things as per the user convenience. The connections of the LCD screen, keypad and microcontroller board is shown in the circuit diagram.

There are a total of twelve input key presses. Nine keys of port P0 of the keypad are corresponding to nine text's to be displayed , one for providing running text animation and the remaining two for user defined needs. On the output side a 16*2 LCD screen is used to display the user defined text. There is also a provision for providing a text animation for small duration.

When the system starts, the LCD screen doesn't display any output as defined. The microcontroller keeps scanning for any key presses. As soon as any one of the key is pressed, the microcontroller calls the suitable function defined as per the appropriate key press in the code embedded in the microcontroller. The 16*2 LCD screen displays the text defined in the function corresponding to the key pressed. Now even if any other input key is pressed, an appropriate function is called and suitably displayed in the LCD screen.

CIRCUIT DIAGRAM
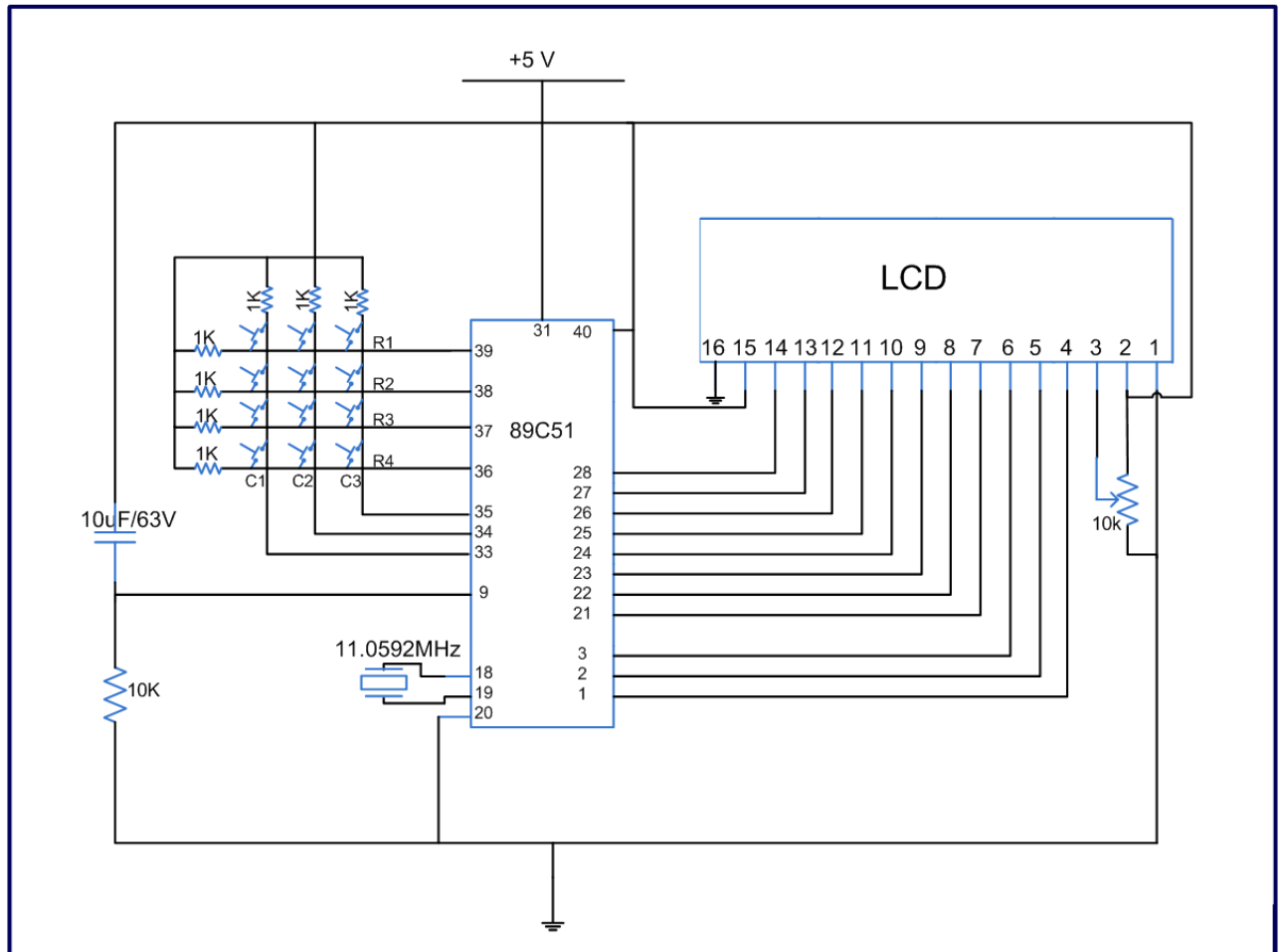
CODE

```c
#include<reg51.h>
#define port P1
#define dataport P2  // Dataport for lcd
#define key P0  // Port for keypad
#define sec 100
sbit rs = port^0;
sbit rw = port^1;
sbit en = port^2;

sbit col1=key^4;
sbit col2=key^5;
sbit col3=key^6;
sbit row1=key^0;
sbit row2=key^1;
sbit row3=key^2;
sbit row4=key^3;
void delay(unsigned int msec) //Time delay function
{
int i,j ;
for(i=0;i<msec;i++)
  for(j=0;j<1275;j++);
}

void lcd_cmd(unsigned char item) //Function to send command to LCD
{
dataport = item;
rs= 0;
rw=0;
en=1;
delay(1);
en=0;
return;
}

void lcd_data(unsigned char item) //Funtion to send data on LCD
{
dataport = item;
rs= 1;
rw=0;
en=1;
delay(1);
en=0;
return;
}

void lcd_data_string(unsigned char *str) // Function to send string on LCD
{
int i=0;
while(str[i]!='\0')
```

```
{
 lcd_data(str[i]);
 i++;
 delay(10);
}
return;
}

void lcd(unsigned char str[10])  // Funtion to Initialize LCD
{
lcd_cmd(0x38);
lcd_cmd(0x0e);
//delay(sec);
lcd_cmd(0x01);
//delay(sec);
lcd_cmd(0x82);
//delay(sec);
lcd_data_string(str);
}

        void ab1() // used to call the function when row1 and column1 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(24);
lcd_data(24);
lcd_data(8);
lcd_data(8);
lcd_data(24);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);  //Address of the location where the kannada character is to be displayed
lcd_data(0);  // Displaying the character created at address 0x64
delay(10);

lcd_cmd(72);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(14);
lcd_data(10);
lcd_data(14);
lcd_cmd(0x82);
lcd_data(1);
delay(10);
```

```c
lcd_cmd(80);
lcd_data(2);
lcd_data(2);
lcd_data(30);
lcd_data(20);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
delay(10);
}
 void ab2()// used to call the function when row1 and column2 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(16);
lcd_data(8);
lcd_data(4);
lcd_data(2);
lcd_data(1);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);  //Address of the location where the kannada character is to be displayed
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(2);
lcd_data(2);
lcd_data(30);
lcd_data(18);
lcd_data(18);
lcd_data(18);
lcd_data(18);
lcd_data(30);
lcd_cmd(0x82);
lcd_data(1);
delay(10);

lcd_cmd(80);
lcd_data(1);
lcd_data(1);
lcd_data(31);
lcd_data(17);
```

```c
lcd_data(19);
lcd_data(17);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
delay(10);
}
  void ab3()// used to call the function when row1 and column3 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada custom character is stored
lcd_data(1);
lcd_data(0);
lcd_data(15);
lcd_data(0);
lcd_data(0);
lcd_data(20);
lcd_data(20);
lcd_data(31);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(16);
lcd_data(16);
lcd_data(16);
lcd_data(23);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(29);
lcd_cmd(0x81);
lcd_data(1);
delay(10);

lcd_cmd(80);
lcd_data(4);
lcd_data(4);
lcd_data(28);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
```

```c
delay(10);
}
void ab4()// used to call the function when row2 and column1 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(31);
lcd_data(25);
lcd_data(1);
lcd_data(1);
lcd_data(29);
lcd_data(21);
lcd_data(21);
lcd_data(23);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(29);
lcd_data(21);
lcd_data(29);
lcd_data(17);
lcd_data(17);
lcd_data(17);
lcd_data(17);
lcd_data(31);
lcd_cmd(0x82);
lcd_data(1);
delay(10);

lcd_cmd(112);
lcd_data(1);
lcd_data(4);
lcd_data(4);
lcd_data(14);
lcd_data(10);
lcd_data(10);
lcd_data(14);
lcd_data(0);
lcd_cmd(0xC4);
lcd_data(6);
delay(10);
}
  void ab5()// used to call the function when row2 and column2 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
```

```
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(31);
lcd_data(21);
lcd_data(21);
lcd_data(1);
lcd_data(1);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(1);
lcd_data(31);
lcd_data(17);
lcd_data(17);
lcd_data(21);
lcd_data(31);
lcd_data(4);
lcd_data(7);
lcd_cmd(0x82);
lcd_data(1);
delay(10);
}
 void ab6()// used to call the function when row2 and column3 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(28);
lcd_data(20);
lcd_data(28);
lcd_data(19);
lcd_data(16);
lcd_data(16);
lcd_data(8);
lcd_data(7);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(28);
lcd_data(20);
lcd_data(20);
```

```c
lcd_data(28);
lcd_data(16);
lcd_data(16);
lcd_data(16);
lcd_data(16);
lcd_cmd(0x81);
lcd_data(1);
delay(10);

lcd_cmd(80);
lcd_data(4);
lcd_data(28);
lcd_data(20);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
delay(10);
}
 void ab7()// used to call the function when row3  and column1 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
//delay(sec);
lcd_cmd(0x01);
//delay(sec);
lcd_cmd(0x82);
//delay(sec);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(1);
lcd_data(31);
lcd_data(16);
lcd_data(8);
lcd_data(6);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(0);
lcd_data(17);
lcd_data(17);
lcd_data(10);
lcd_data(4);
lcd_data(10);
```

```
lcd_data(17);
lcd_data(31);
lcd_cmd(0x82);
lcd_data(1);
delay(10);

lcd_cmd(80);
lcd_data(0);
lcd_data(0);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(28);
lcd_cmd(0x83);
lcd_data(2);
delay(10);
}
 void ab8()// used to call the function when row3 and column2 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
//delay(sec);
lcd_cmd(0x01);
lcd_cmd(0x82);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(0);
lcd_data(16);
lcd_data(8);
lcd_data(4);
lcd_data(2);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(14);
lcd_data(10);
lcd_data(14);
lcd_cmd(0x82);
lcd_data(1);
delay(10);
```

```
lcd_cmd(80);
lcd_data(27);
lcd_data(17);
lcd_data(17);
lcd_data(27);
lcd_data(17);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
delay(10);

lcd_cmd(88);
lcd_data(0);
lcd_data(0);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(4);
lcd_data(28);
lcd_cmd(0x85);
lcd_data(3);
delay(10);
}
 void ab9()// used to call the function when row3 and column3 are pressed
{
 lcd_cmd(0x38);
lcd_cmd(0x0e);
//delay(sec);
lcd_cmd(0x01);
//delay(sec);
lcd_cmd(0x82);
//delay(sec);
lcd_cmd(64);  //Address where values of the first kannada character is stored
lcd_data(24);
lcd_data(24);
lcd_data(8);
lcd_data(24);
lcd_data(16);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x80);
lcd_data(0);
delay(10);

lcd_cmd(72);
lcd_data(0);
```

```
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(14);
lcd_data(10);
lcd_data(14);
lcd_cmd(0x82);
lcd_data(1);
delay(10);

lcd_cmd(80);
lcd_data(25);
lcd_data(25);
lcd_data(9);
lcd_data(29);
lcd_data(21);
lcd_data(21);
lcd_data(21);
lcd_data(31);
lcd_cmd(0x84);
lcd_data(2);
delay(10);

lcd_cmd(88);
lcd_data(4);
lcd_data(28);
lcd_data(4);
lcd_data(4);
lcd_data(29);
lcd_data(5);
lcd_data(5);
lcd_data(31);
lcd_cmd(0x86);
lcd_data(3);
delay(10);
}
ab10()// used to call the function when row4 and column2 are pressed
{
int i=0;
for(i=0;i<12;i++)
{
 lcd_cmd(0x1C);  //Shift the entire display to right
  delay(75);
}
}
void display(int a) //Display functon for LCD

{
switch(a)
{
```

```c
   case 1:ab1();
     break;
   case 2:ab2();
     break;
   case 3:ab3();
     break;
   case 4:ab4();
     break;
   case 5:ab5();
     break;
   case 6:ab6();
     break;
   case 7:ab7();
     break;
   case 8:ab8();
     break;
   case 9:ab9();
     break;
   case 0:ab10();
     break;
   case 11:lcd("*");
     break;
   case 12:lcd("#");
     break;
}
}

void check_col1() //Function for checking column one
{
row1=row2=row3=row4=1;
row1=0;
if(col1==0)
display(1);
row1=1;
row2=0;
if(col1==0)
display(4);
row2=1;
row3=0;
if(col1==0)
display(7);
row3=1;
row4=0;
if(col1==0)
display(11);
row4=1;
}

void check_col2() //Function for checking column two
{
row1=row2=row3=row4=1;
```

```c
row1=0;
if(col2==0)
display(2);
row1=1;
row2=0;
if(col2==0)
display(5);
row2=1;
row3=0;
if(col2==0)
display(8);
row3=1;
row4=0;
if(col2==0)
display(0);
row4=1;
}

void check_col3() //Function for checking column three
{
row1=row2=row3=row4=1;
row1=0;
if(col3==0)
display(3);
row1=1;
row2=0;
if(col3==0)
display(6);
row2=1;
row3=0;
if(col3==0)
display(9);
row3=1;
row4=0;
if(col3==0)
display(12);  //For #
row4=1;
}

void main()
{
col1=col2=col3=1;  //Input Port
while(1)
{
  row1=row2=row3=row4=0;
  if(col1==0)
  check_col1();
  else
   if(col2==0)
   check_col2();
   else
```

```
    if(col3==0)
    check_col3();
}
}
```

Components List

- 8051 Project board.

- 3*4 Keypad

- Double ended female cable(3 sets)

- 89c52 micro-controller chip(ATMEL)

- Bug sticks

- 16*2 LCD Display

- Power adapter (12 V)