Amrita Vishwa Vidyapeetham, Amritapuri, Kollam

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

A PROJECT REPORT

ON

"SIMPLE CALCULATOR"

SUBMITTED BY

Mr. Chetlapalli Dattasai Vamsikrishna (AM.EN.U4ECE18011)
Mr. Chokkapu Sandeep (AM.EN.U4ECE18012)
Mr. Yerra Bharat Sai Teja (AM.EN.U4ECE18058)
Miss. Garlapati Kavya (AM.EN.U4ECE18017)

UNDER THE GUIDANCE OF

PROF. SENTHIL MURUGAN

(Academic Year: 2018-2022)

Amrita Vishwa Vidyapeetham, Amritapuri, Kollam

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Certificate

This is to certify that project entitled

"SIMPLE CALCULATOR"

has been completed by
Mr. Chetlapalli Dattasai Vamsikrishna (AM.EN.U4ECE18011)
Mr. Chokkapu Sandeep (AM.EN.U4ECE18012)
Mr. Yerra Bharat Sai Teja (AM.EN.U4ECE18058)
Miss. Garlapati Kavya (AM.EN.U4ECE18017)

in partial fulfillment for the award of Bachelor of "Electronics and Communication Engineering" of Amrita Vishwa VidyaPeetam,Amritapuri, Kollam.

Prof. Senthil Murugan **Signature**

Place: Amritapuri, Kollam.

Date: 13/12/2020

ACKNOWLEDGEMENT

We like to share our sincere gratitude to all those who help us in completion of this project.It gives us great pleasure and satisfaction in presenting this mini project on **"SIMPLE CALCULATOR".** *During the work we faced many challanges due to lack of knowledge and experience but these people help us to get over from all the difficulties and in final compilation of our idea to a shaped sculpture.*

We would like to thank **Prof. Mr. Senthil Murugan** sir for his guidance, because of which our whole team was able to learn the minute aspects of a project work.

We would like to thank our project coordinators **Miss.Baby Mam** *and* **Miss. Chinmayi** mam *for their help and monitoring during the project. In the last we would like to thank the management of Amrita Vishwa Vidya Peetham, Amritapuri for providing us such an opportunity to learn from these experiences.*

All of our team is thankful to Mr.Senthil Murugan sir, Miss.Chinmayi mam, Miss.Baby mam of Department of **Electronics and Communication Engineering***, for this help and support towards this project and our team.*

Thank you all.

Mr. Chetlapalli Dattasai Vamsikrishna (AM.EN.U4ECE18011)
Mr. Chokkapu Sandeep (AM.EN.U4ECE18012)
Mr. Yerra Bharat Sai Teja (AM.EN.U4ECE18058)
Miss. Garlapati Kavya (AM.EN.U4ECE18017)

Contents

Chapter 1

ABSTRACT

Our Aim of this project is to design a **Simple Calculator** *using* **LPC2148** *interfacing with* **16X2 LCD** *and* **4X4 Matrix Keypad.**

Today, calculators are everywhere. Everything around us involves a math operation between numbers and their calculations, from buying a handful of candies to handling complex digits. Today technology has grown up such that performing operation on complex digits is even easy. Calculators have also become an integral part of life. Calculators can benefit or serve as crutches for society. They prove beneficial in speeding up calculations when paying bills and taking tests and so on. Calculator is a device helps us to perform calculations easy.

A calculator is a device that performs arithmetic operations on numbers. The simplest calculators can do only addition, subtraction, multiplication, and division. More sophisticated calculators can handle exponential operations, roots, logarithms, trigonometric functions, hyperbolic functions, complex calculations and so on.Each new generation of calculators builds on the previous one, with heightened speed and more advanced capabilities.

The Enhancement of the technology has brought many changes over past few decades. From a bulky Calculator design with several blocks to a calculator application on smart phone right there on the home screen.

The objective of this Project is to design a Simple Calculator which performs Arithmetic operations on given numbers through keypad and display the arithmetic result on LCD using LPC2148 micro controller.

Chapter 2

INTRODUCTION

A calculator is a small hand-held electronic device used to perform mathematical and logical operations. It often has a microprocessor or micro controller chip embedded in it that enables it to perform these functions. The microprocessor came about as a programmable replacement for control circuits and calculator chips in the 1970s. Up to this point, most control systems using digital logic were implemented using individual logic integrated circuits to create the design and as more fun required, the circuits became very bulky unreliable.

Many hundreds of these discreet logic chips were needed just to create a simple four function calculator. But with the invention of the microprocessor which has a complete processing unit inside a microchip, the reliability of digital logic circuits was greatly increased as a result of reduced size. Subsequently, there was need to include a permanent memory in the microprocessor which ultimately gave birth to the micro controller the processor component used in this design. The microprocessor that doesn't have a program or data memory, the micro controller has a program and data memory in addition to a central processing unit (CPU).

An embedded micro controller (also called a microcomputer and sometimes abbreviated µCor uC or MCU) contains a processor core, memory, and programmable input/output peripherals integrated circuit. An integrated circuit is a complete electronic circuit made semiconductor material (also called an encapsulated in plastic and used as one component.

The simple calculator realized in this design is capable of performing four arithmetic operations: addition (+), subtraction (-), multiplication (*) and division (/).

Chapter 3

LITERATURE REVIEW

Having a basic understanding of the development of technology over the course of history can help us to predict the potential growth of technology to come. Here, the professionals at Tech mart take you through the history of the calculator, and how we came to have the ubiquitous devices we use every day.

It is almost impossible for us to imagine mathematics without something as seemingly simple as a calculator. That being said, the calculators we know today were not invented until the 1970s, and the use of smartphones as calculators did not begin until at least the late nineties. But that doesn't mean that mathematical tools were not available before the 20th century—there were many different computing machines created long before digital calculators and smartphones.

The first tool created specifically for use in mathematical computations was the abacus, likely invented in Sumeria around 2500 B.C. The abacus was a table of successive columns with beads or stones representing a single unit, which could be used for addition or subtraction. Other cultures altered and refined the abacus; the Chinese, for example, put beads on wire within a bamboo frame to enhance ease of use. Unfortunately, the abacus was not very useful for multiplication or division, necessitating the invention of a new and more sophisticated device.

Fast forward 4,500 years to 1617, when Scottish mathematician John Napier published Rabdology, or "calculation with rods." In his writing, Napier described a device that came to be known as Napier's bones. The "bones" are thin rods, which are inscribed with multiplication tables, and the user determines their sum by changing the vertical alignment of the rods, and horizontally reading the multiplication tables. While these devices greatly assisted calculations, they were not true "calculators," merely assisting the person doing the mental calculation.

In 1642, the first true "calculator" was invented: one that performed calculations through a clockwork-type of mechanism. The Pascal calculator, invented by French inventor and mathematician Blaise Pascal, was lauded for attempting arithmetic calculations previously thought impossible. But unfortunately, they were difficult to produce and very few were ever made. The mechanical calculator then invented by Thomas de Colmar in the mid-nineteenth century, and subsequent others, were easier to produce, but extremely large and bulky–not at all the pocket calculators we know today.

Curt Herzstark invented the first handheld, mechanic calculator in 1945, from a design he had created in 1938. Shaped like a stout pepper grinder, the Curta calculators were produced in large quantities until 1970, when a company in Japan invented the first digital pocket calculator and the demand for the antiquated device faded. Companies in the United States, such as Texas Instruments, adapted the design of the Japanese device, and enhanced it by creating the graphing calculators we know today.

With the invention of the first smartphone in 1995, individuals began to replace expensive digital calculators with the multi use device. This required even the most sophisticated calculator designs to be upgraded in order to remain relevant in the market.

Calculators have not only greatly enhanced our ability to perform the regular computations that are involved in everyday life, but provided humans with the ability to understand mathematics on a greater scale than ever imagined. Calculations which were previously cumbersome and time-consuming can now be done in minutes or even seconds, all at the push of a few buttons. Without calculators, advanced math courses, such as Calculus, would require much longer class-times and reformatted lesson plans.

Chapter 4

THEORY

LPC2148 MICROCONTROLLER

ARM is one of the major options available for embedded system developer.LPC2148 is the widely used IC from ARM-7 family.ARM is a family of instruction set architectures based on a reduced instruction set computing (RISC) architecture.The ARM LPC2148 has two ports Port 0 and Port 1 respectively each port contains 32-bits for I/O operations. The Port 0 has 32-bit of I/O pins for individual directions and Port 1 has 32-bits of I/O pins for bidirectional purpose.

The function of a pin can select using a register PINSEL. Two bits in the PINSEL register is used to select the function of one port pin. Using PINSEL0 we can only select the function of port0 pin up to 15.for remaining pins we can choose PINSEL1PINSEL2. The IODIR register is used to configure the direction of the pin if write one in this register corresponding pin become output. Using a register named IOSET, we can out a value through corresponding port. Using this register we can out only high level (1).writing zero in using IOSET register make no change in port pins. IOCLR register is used for making port pins low by writing high(1).

16X2 LCD DISPLAY

16×2 alphanumeric LCD can display 16 characters in each of its row, and it has two rows. To interface an LCD we need to know about its internal registers and pin functions. An LCD has two important registers namely command register and Data register. Both are 8-bit registers. The command register is written with various commands by the user. These commands are responsible for the functioning of LCD.

The data register is loaded with the ASCII value of the Data (Character/Number, etc.) to be displayed. Both the registers are accessed through 8-bit Data Line (D0-D7). There is one selection line provided, namely R/S Pin, in LCD Module which selects between these two registers.
R/S=0– Command Register
R/S=1–Data Register

Apart from R/S, there are other two Pins, viz. R/W and EN. EN pin is to enable

(1) or to disable (0) the LCD. R/W is used to select the operation 'Read from LCD' or 'Write to LCD'. Logic 0 at R/W stands for Write operation and 1 for Read operation. Apart from these pins, the last two Pins (LED+ LED-) on the LCD are for Backlight. And pins 1,2,3 are for GND, +VCC and +VEE(contrast adjustment) respectively.

LCD PIN DESCRIPTIONS	
LCD pin name	LCD pin Description
Vss	Ground pin of the LCD module.
Vcc	Power to LCD module (+5V supply)
VEE	Contrast adjustment pin
RS	Register select pin RS=0 command register. RS=1 data register.
R/W	Read/Write modes R/W=1; Read mode R/W=1; Write mode
EN	This pin is meant for enabling the LCD module
DB0 to DB7	8 data pins
LED+	Anode of the back light LED
LED-	Cathode of the back light LED

The data and command are given through these pins. EN: enable The lcd controller take the data from data pins only get a high to low transition in enable pin.

R/W : Mode of operation is select using this pin. Giving low to this pin for write something on lcd .read from lcd by making this pin in high state.

RS: Register select pin.LCD has two registers namely data register and command register.

Writing commands codes for initializing the LCD. For display something on LCD first we have to initialize the LCD by giving some commands. Make RS pin low for giving commands. The LCD controller will take the values in data pins as commands when RS pin is in low state.

If RS is in high state the controller will take the values in data pins for displaying. There are various commands that need to be send to LCD so as to Initialize it, to make it work accordingly. These command are listed below.

Mostly used commands or instructions for LCD	
Instruction	Hex
Function Set: 8-bit mode, 1 Line, 5x7 Dots matrix for each character display	0x30
Function Set: 8-bit mode, 2 Line, 5x7 Dots matrix for each character display	0x38
Function Set: 4-bit mode, 1 Line, 5x7 Dots matrix for each character display	0x20
Function Set: 4-bit mode, 2 Line, 5x7 Dots matrix for each character display	0x28
Clear display screen	0x01
Return Home	0x02
Decrements cursor(Shift to left)	0x04
Increment cursor(Shift to Right)	0x06
Display off Cursor off	0x08
Display on Cursor on	0x0E
Display on Cursor off	0x0C
Display on Cursor blinking	0x0F
Shift entire display left	0x18
Shift entire display right	0x1C
Move cursor left by one character	0x10
Move cursor right by one character	0x14
Clear Display (also clear DDRAM content)	0x01
Force cursor position on first position of first	0x80
Force cursor position on first position of second row	0xC0

4X4 MATRIX KEYPAD:

Keypad is used as an input device to read the key pressed by the user and to process it. 4x4 keypad consists of 4 rows and 4 columns. Switches are placed between the rows and columns. A key press establishes a connection between corresponding row and column between which the switch is placed. In order to read the key press, we need to configure the rows as outputs and columns as inputs.Columns are read after applying signals to the rows in order to determine whether or not a key is pressed and if pressed, which key is pressed.

Keypad is organized as a matrix of switches in rows and column. Here uses a 4X4 matrix keypad and a 16x2 LCD for displaying the output of keypad.Every number is assigned two unique parameters, row and column number. Hence every time a key is pressed the number is identified by detecting the row and column number of the key pressed. The above process is very fast and even if the switch is pressed for a very small duration of time the controller can detect the key which is pressed. The controller displays the number corresponding to the row and column on the LCD.

working of keypad

Step-1:Set output port to logic 0 and input port to logic 1.

If all rows are set to logic 1 and no key is pressed, reading input port will result all input pins set at logic 1 as there is no path available for current to flow between Vcc and Ground. It is a function of micro-controller to read input port continuously to identify whether the key has been pressed or not.

Step-2: Whenever any key is pressed, one of the input pin will be grounded by pressed key. By reading input pins, we can identify the column in which key has been pressed.Now we have to identify the row in which key has been pressed.

Step-3: Ground first row and set rest of the rows. Read input port. If all are 1s, no switch in that row is pressed. Now ground second row and set all others rows at logic 1 and see whether any key has been pressed in that row or not. Continue same process until you find out in which row, key is pressed.

Once we know column and row in which key has been pressed, we can identify the pressed key easily.

Step-4: Ground all rows and read columns. If any of the columns has 0, keep doing this process unless all columns are 1.

Step-5: Once we have identified the pressed key, we can display it on LCD

Chapter 5

COMPONENTS

Software's Used:

- ☐ **KEIL SOFTWARE:**

  It is used for coding, compiling and debugging the logic in c-language.

- ☐ **PROTEUS SOFTWARE:**

  It is used for building the circuit and for simulating the hardware components along with logic interfacing from software.

Hardware Used:

- ☐ **LPC2148:**

  It is used as a micro controller for writing a program and used to connect to external peripherals and logic.

- ☐ **4x4 Matrix KEYPAD:**

  It is used to input the numbers to perform any operation.

- ☐ **16X2 LCD:**

  It is used to display or output the content provided.

Chapter 6

BLOCK DIAGRAM

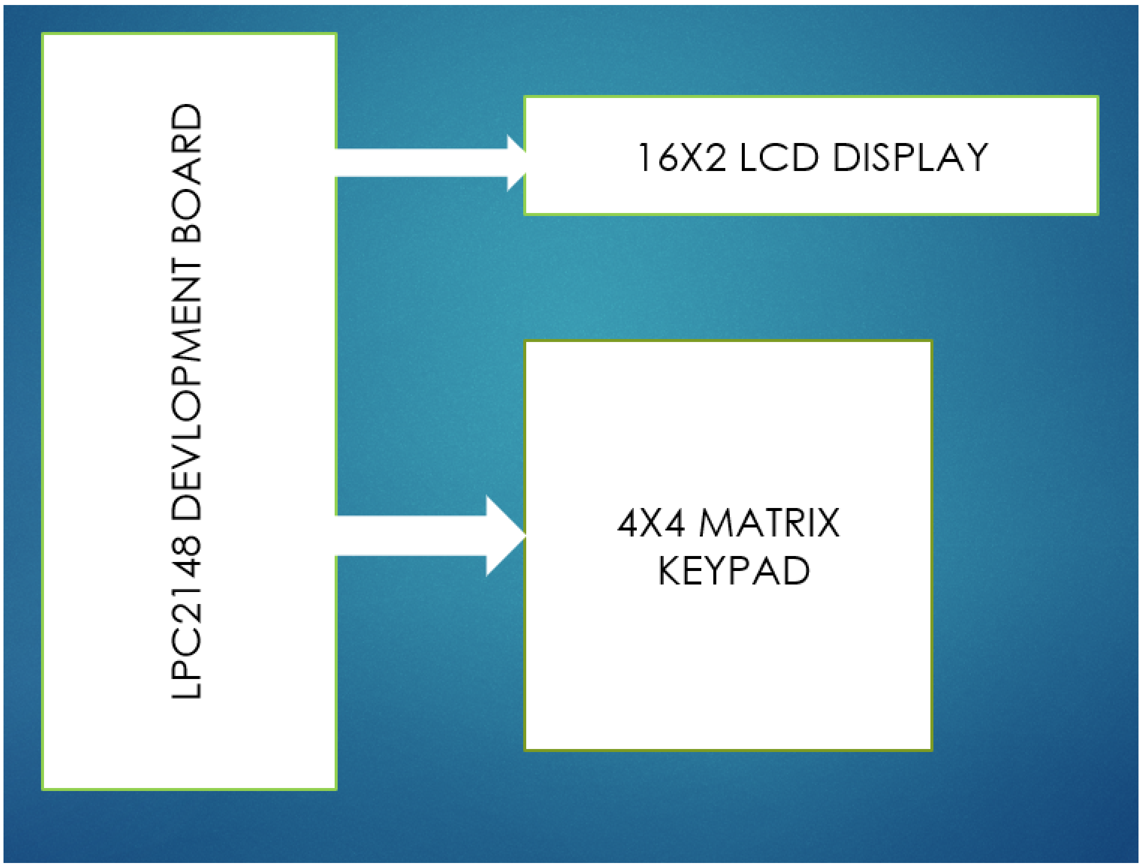

Figure 6.1: Block diagram of circuit

The above figure 6.1 describes overview of circuit where we can see what components are included in circuit and how circuit is going to be designed.

Chapter 7

CIRCUIT DIAGRAM

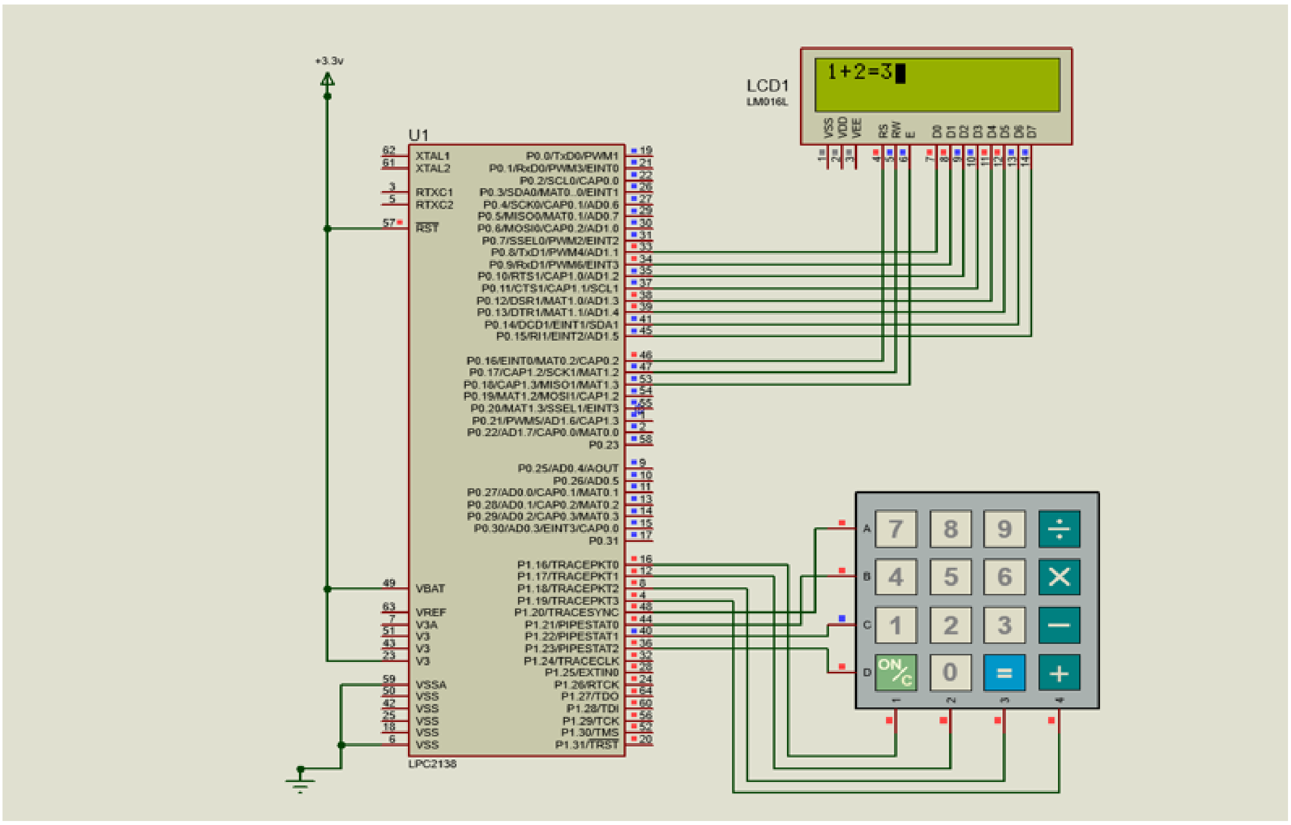

Figure 7.1: Circuit diagram

The above figure 7.1 describes circuit showing for **"SIMPLE CALCULATOR"**.The connections and circuit explanation is discussed in Implementation.

Chapter 8

PROCEDURE

STEPS:

- ☐ The LPC2148 is interfaced with Keypad and LCD display.
- ☐ The arithmetic operations such as addition, subtraction, multiplication, and division are performed on inputs given through the keypad.
- ☐ The logic has been coded in LPC2148.
- ☐ The output is displayed on LCD display.
- ☐ Thus, The input is given through the Keypad and the output is sensed at the display.
- ☐ Therefore, The arithmetic result of input numbers through keypad is displayed on LCD.

SOFTWARE INTERFACING:

- ☐ To simulate a simple arithmetic calculator using Proteus we are using LPC2148 as main micro controller in the Project
- ☐ Proteus software and the code in Keil is interfaced through hex file created in Keil with proteus components.

Chapter 9

RESULTS

The arithmetic result of input numbers through keypad is displayed on LCD.

COST OF COMPONENTS:

COMPONENTS COST	
Components	Cost
LPC2148	1150/-
16X2 LCD	70/-
4X4 MATRIX KEYPAD	80/-

Code referral link:
https://github.com/garlaptikavya/Simple-Caluculator.git

Chapter 10

FUTURE WORK

We can further improve the current simple calculator design to a perform the scientific calculations involving trigonometric functions, logical functions, polynomial functions, Complex logic's, and so on.

APPLICATIONS:

Simple Calculations are even plays a high role in daily works such as budget, interest calculations.... and these simple calculations may be part of a larger control systems like a petrol pump control mechanism, a water level sensor, or an inverter, and so on.

Chapter 11

REFERENCES

☐ https://www.electronicwings.com/arm7/lcd-16x2-interfacing-with-lpc2148-8-bit-mode

☐ https://www.electronicwings.com/arm7/4x4-keypad-interfacing-with-lpc2148

☐ https://embetronicx.com/tutorials/microcontrollers/lpc2148/keypad-interfacing-with-lpc2148/

☐ http://www.firmcodes.com/microcontrollers/arm/keypad-interfacing-with-arm7-lpc2148/

☐ https://kapawarsantoshmicrocontrollerprojects.wordpress.com/2016/08/14/calculator-design-using-lpc2148/

Chapter 12

APPENDIX

CODE:

#include ¡lpc21xx.h¿

#include ¡string.h¿
#define LCD (0xff¡¡8)

#define RS (1¡¡16)
define RW (1¡¡17)
define EN (1¡¡18)

define r1 (1¡¡16)
define r2 (1¡¡17)
define r3 (1¡¡18)
define r4 (1¡¡19)

define c1 (1¡¡20)

define c2 (1¡¡21)

define c3 (1¡¡22)

define c4 (1¡¡23)

define Error 13

void cct$_i nit(void)$;
$voiddelay(unsignedinttime)$;
$voidlcdinit(void)$;

$voidwriteline(char * str)$;

$voidReturnHome(void);$
$charget_key(void);$
$intget_num(char);$
$charget_func(char);$
$voidDispError(int);$
$voiddisp_num(int);$
$voidwritecmd(unsignedcharcommand);$
$voidwritedata(unsignedintdata);$
$voidkeypad_delay(void);$
$unsignedcharREAD_SWITCHES(void);$
$intmain(void)$
$charkey; intnum1 = 0; charfunc =' +'; intnum2 = 0;$
cct$_init();$
$lcdinit();$
$while(1)$

key = get$_key();$
$writecmd(0x01);$
$writedata(key);$
$num1 = get_num(key);$

if(num1!=Error)

key = get$_key();$
$writedata(key);$
$func = get_func(key);$

if(func!='e')

key = get$_key();$
$writedata(key);$
$num2 = get_num(key);$

if(num2!=Error)

key = get$_key();$
$writedata(key);$

if(key == '=')

switch(func)

case '+': disp$_num(num1 + num2); break;$
$case'-' : disp_num(num1 - num2); break;$
$case'*' : disp_num(num1 * num2); break;$
$case'/' : disp_num(num1/num2); break;$

$else$
$if(key ==' C')writecmd(0x01);elseDispError(0);$

void cct$_init(void)$
$PINSEL0 = 0x00000000; IODIR0 = 0Xffffffff; PINSEL1 = 0x00000000; IODIR1 = 0x00f$

void keypad$_delay(void)$
$unsignedintt1, t2; for(t1 = 0; t1 < 300; t1 + +)for(t2 = 0; t2 < 1275; t2 + +);$

unsigned char READ$_SWITCHES(void)$
$unsignedcharkey; IOCLR1| = (c1|c2|c3|c4|r1|r2|r3|r4); while(1)IOCLR1| = c1; IOSET1| = (c2|c3$
if((IOPIN1r1)==0)

key='7';
keypad$_delay();$
$returnkey;$

$elseif((IOPIN1r2) == 0)$
$key =' 8'; keypad_delay(); returnkey;$
$elseif((IOPIN1r3) == 0)$
$key =' 9'; keypad_delay(); returnkey;$
$elseif((IOPIN1r4) == 0)$
$key =' /'; keypad_delay(); returnkey;$

IOCLR1—=c2;
IOSET1—=(c1—c3—c4);

if((IOPIN1r1)==0)

key='4';
keypad$_delay()$;
$returnkey$;

$elseif((IOPIN1r2) == 0)$
$key =' 5'; keypad_delay(); returnkey;$
$elseif((IOPIN1r3) == 0)$
$key =' 6'; keypad_delay(); returnkey;$
$elseif((IOPIN1r4) == 0)$
$key =' *'; keypad_delay(); returnkey;$

IOCLR1—=c3;
IOSET1—=(c1—c2—c4);

if((IOPIN1r1)==0)

key='1';
keypad$_delay()$;
$returnkey$;

$elseif((IOPIN1r2) == 0)$
$key =' 2'; keypad_delay(); returnkey;elseif((IOPIN1r3) == 0)$
$key =' 3'; keypad_delay(); returnkey;elseif((IOPIN1r4) == 0)$
$key =' -'; keypad_delay(); returnkey;$
IOCLR1—=c4;
IOSET1—=(c1—c2—c3);

if((IOPIN1r1)==0)
key='C';
keypad$_delay()$;
$returnkey$;

else if((IOPIN1r2)==0)

key='0';
keypad$_delay()$;
$returnkey$;

$elseif((IOPIN1r3) == 0)$
$key ='='; keypad_delay(); returnkey;$

$elseif((IOPIN1r4) == 0)$
$key =' +'; keypad_delay(); returnkey;$

void writecmd(unsigned char command)

IO0CLR—=(RS—RW—EN—LCD);
IO0SET—=(command¡¡8);
IO0CLR—=RS;
IO0CLR—=RW;
IO0SET—=EN;
delay(2);
IO0CLR—=EN;
delay(3);

void writedata(unsigned int data)

IO0CLR—=(RS—RW—EN—LCD);
IO0SET—=(data¡¡8);
IO0SET—=RS;
IO0CLR—=RW;
IO0SET—=EN;
delay(2);
IO0CLR—=EN;
delay(3);

void lcdinit(void)

delay(5);
writecmd(0X38);
writecmd(0X0f);
writecmd(0X06);
writecmd(0X01);
delay(5);
writecmd(0X80);

void delay(unsigned int time)

unsigned int t1,t2;
for(t1=0;t1¡time;t1++)
for(t2=0;t2¡1275;t2++);

void writeline(char *str)

while(*str!=”)

writedata(*str);
str++;

char get$_key(void)$
$charkey =' n';$
while(key==’n’)
key = READ$_SWITCHES();$

return key;

int get$_num(charch)$
$switch(ch)case'0' : return0; break; case'1' : return1; break; case'2' : return2; break; case'3' : return3;$

void ReturnHome(void)

writecmd(0x02);
delay(1500);

char get$_func(charchf)$
$if(chf ==' C')writecmd(0x01); return'e';$
if(chf!=’+’ chf!=’-’ chf!=’*’ chf!=’/’)

DispError(1);
return ’e’;

return chf;

void DispError(int numb)

writecmd(0x01);
switch(numb)

case 0: writeline(”Wrong Input”); break;
case 1: writeline(”Wrong Function”); break;
default: writeline(”Wrong Input”); break;

void disp$_num(intnumb)$
$unsignedcharUnitDigit = 0; unsignedcharTenthDigit = 0;$
if(numb¡0)

numb = -1*numb;
writedata(’-’);

TenthDigit = (numb/10);

if(TenthDigit != 0)
writedata(TenthDigit+0x30);

UnitDigit = numb - TenthDigit*10;

writedata(UnitDigit+0x30);