**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỆN TỬ**

---------------o0o---------------

****

**BÁO CÁO**

**BÀI TẬP LỚN MÔN HỌC**

**LẬP TRÌNH HỆ THỐNG NHÚNG**

**ĐỀ TÀI**

**MÔ HÌNH MÁY TÍNH CẦM TAY**

**NHÓM: X LỚP: L02 HK202**

**GIẢNG VIÊN HƯỚNG DẪN: THẦY TRƯƠNG QUANG VINH**

SINH VIÊN THỰC HIỆN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ** | **TÊN** | **GHI CHÚ** |
| 1 | 1810098 | DƯƠNG VĂN | ĐẠI |  |
| 2 | 1810162 | DƯƠNG NHẬT | HUY |  |
| 3 | 1613398 | NGUYỄN CÔNG | THÔNG |  |

**TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2021**

***LỜI CẢM ƠN***

*Nhóm em xin gửi lời cám ơn tới thầy Trương Quang Vinh, giảng viên môn Lập trình hệ thống nhúng đã giúp đỡ nhóm trong quá trình hoàn thành sản phẩm.*

*Tp. Hồ Chí Minh, ngày 4 tháng 6 năm 2021.*

**Lớp L02. Nhóm 2**

**TÓM TẮT**

Bài báo cáo nhóm 2 chúng em xin trình bày về sản phẩm của nhóm là “Mô hình Máy tính cầm tay”. Trong thực tế xã hội hiện nay nhu cầu tính toán là rất cần thiết. Vì vậy nhóm đề xuất thiết bị máy tính cầm tay. Trong bài báo cáo này nhóm sẽ trình bày về thiết kế phần cứng, phần mềm và quá trình thực hiện sản phẩm của nhóm.

MỤC LỤC

[**1. GIỚI THIỆU 1**](#_Toc73202794)

[**1.1 Tổng quan 1**](#_Toc73202795)

[**1.2 Nhiệm vụ đề tài*.* 1**](#_Toc73202796)

[**1.3 Phân chia công việc trong nhóm 1**](#_Toc73202797)

[**2. LÝ THUYẾT 2**](#_Toc73202798)

[**3. ĐẶC TẢ HỆ THỐNG (SYSTEM REQUIREMENT) 6**](#_Toc73202799)

[**4. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 8**](#_Toc73202800)

[**4.1. System block diagram 8**](#_Toc73202801)

[**4.2. Block description 9**](#_Toc73202802)

[**5. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 10**](#_Toc73202803)

[**6. KẾT QUẢ THỰC HIỆN 24**](#_Toc73202804)

[**7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 25**](#_Toc73202805)

[**7.1 Kết luận 25**](#_Toc73202806)

[**7.2 Hướng phát triển 25**](#_Toc73202807)

[**8. TÀI LIỆU THAM KHẢO 26**](#_Toc73202808)

# GIỚI THIỆU

## Tổng quan

Trong thực tế hiện nay, nhu cầu thực hiện tính toán từ học sinh đến người buôn bán, làm việc là rất nhiều, yêu cầu sự nhanh chóng, và tiện lợi. Từ đó nhóm tìm hiểu và thực hiện sản phẩm máy tính cầm tay nhỏ gọn với nhiều chức năng tính toán như cộng, trừ, nhân, chia, lũy thừa với cả số nguyên và số thập phân.

## Nhiệm vụ đề tài*.*

Đề tài thực hiên thiết bị có khả năng xử lý và cho kết quả các phép tính nhập từ keypad hiển thị lên màn hình LCD. Yêu cầu về sự chính xác và tốc độ tính toán tức thời, thiết bị nhỏ gọn dễ dàng mang đi và công suất thấp tiết kiệm điện cho mục đích sử dụng lâu dài.

## Phân chia công việc trong nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Công việc | Ghi chú |
| 1 | Dương Văn Đại | Làm báo cáo | Hoàn thành công việc |
| 2 | Dương Nhật Huy | Lập trình phần mềm | Hoàn thành công việc |
| 3 | Nguyễn Công Thông | Lập trình phần cứng | Hoàn thành công việc |

Trong quá trình thực hiện đề tài, các thành viên cùng nhau thực hiện và hỗ trợ lẫn nhau, tất cả đều hoàn thành công việc đúng thời gian ngày 31/5/2021 để chuẩn bị báo cáo.

# LÝ THUYẾT

**2.1. Vi điều khiển Tiva TM4C123GXL:**

Kit phát triển dòng ARM Cortex ™ -M4  32 bit sử dụng chip TM4C123GXL của Texas Intruments. Kit Tiva Lauchapad là một trong những kit được sử dụng nhiều nhất hiện nay để tiếp cận vi điều khiển ARM, kit sử dụng vi điều khiển ARM cortex M4 TM4C123 từ Texas Intrument( TI ), có tích hợp sẵn mạch nạp, giao tiếp UART và Debugger trên một thiết kế nhỏ gọn , dễ sử dụng , ngoài ra kit có chuẩn chân cắm đực cái rất dễ kết nối và làm các shield ghép tầng.

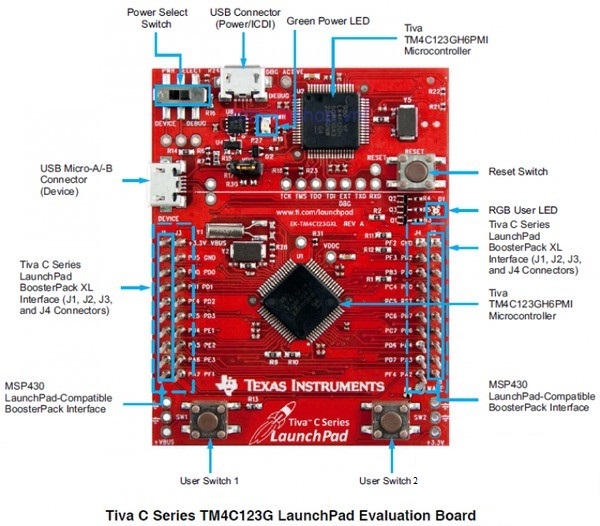
EK-TM4C123GXL bao gồm các nút người dùng có thể lập trình và đèn LED RGB cho các ứng dụng tùy chỉnh. Các tiêu đề có thể xếp chồng lên nhau cung cấp giao diện để kết nối các mô-đun plug-in BoosterPack ™ và giúp việc mở rộng chức năng của TM4C123G LaunchPad trở nên dễ dàng và đơn giản thông qua hệ sinh thái Texas Instruments BoosterPack. TivaWare SDK do TI phát triển cung cấp trình điều khiển cho tất cả các thiết bị ngoại vi và mã ví dụ cho hàng chục ứng dụng cho TM4C123G LaunchPad.

**Thông tin cơ bản Board Tiva TM4C123:**

ARM Cortex-M4F Based MCU TM4C123G LaunchPad (EK-TM4C123GXL) cung cấp các tính năng sau:

* Microcontroller: TM4C123GH6PM 64 chân 80MHz
* 256KB Flash, 32KB SRAM, 2KB EEPROM
* ICDI USB trên bo mạch (Giao diện gỡ lỗi trong mạch trên bo mạch)
* Cổng USB Micro AB
* Công tắc nguồn thiết bị / ICDI
* ADC 12-bit 2MSPS kép, PWM điều khiển chuyển động
* 8 UART, 6 I2C, 4 SPI
* 2 nút USER (SW2 được kết nối với chân WAKE)
* Nút Reset
* 3 đèn LED RGB (1 thiết bị ba màu)
* Các điểm kiểm tra đo lường hiện tại
* Tinh thể dao động chính 16MHz
* Tinh thể đồng hồ thời gian thực 32kHz
* Bộ điều chỉnh 3.3V
* Hỗ trợ nhiều IDE như:

- Keil  
- Sourcery CodeBench  
- IAR Tools  
- CodeComposer Studio



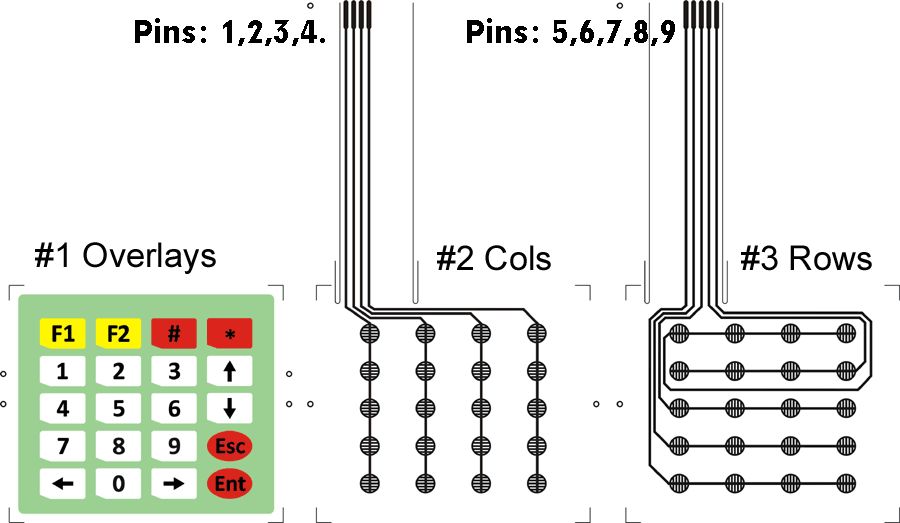
**2.2. Keypad 4x5:**

KeyPad là một thiết bị nhập chứa các nút bấm cho phép người dùng nhập các chữ số, chữ cái hoặc ký tự điều khiển. KeyPad không chứa tất cả bảng mã ASCII như keyboard vì thế nó thường được sử dụng trong các ứng dụng chuyên dụng và tương đối đơn giản, ở đó, số lượng nút nhấn thay đổi phụ thuộc vào ứng dụng.

Keypad 4×5 dùng làm bàn phím để nhập dữ liệu. Bàn phím có đầy đủ các phím chức năng F1, F1, ↑, ↓, ←, →, \*, #, Esc, Enter

**Thông số kĩ thuật:**

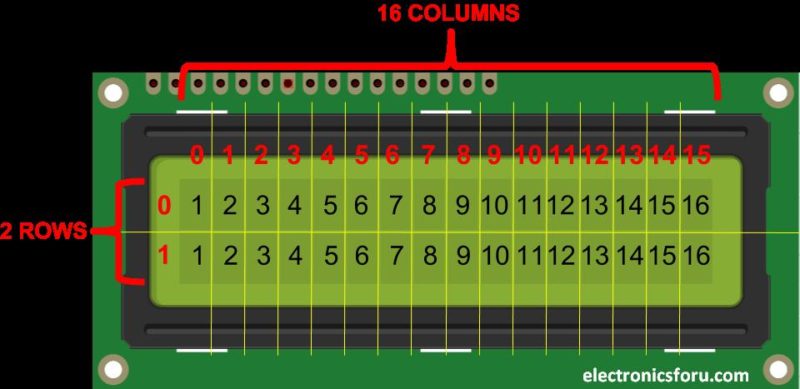
* Module bàn phím ma trận 4×5 loại phím mềm.
* Điện áp tối đa: 35V
* Dòng tối đa đi qua: 100mA
* Điện trở tiếp xúc: 0.5 – 10Ω
* Điện trở cách điện: 100MΩ
* Độ dài cáp: 150mm.
* Nhiệt độ hoạt động 0 – 70ºC.
* Đầu nối ra 9 chân.
* Kích thước bàn phím 85 x 73 mm
* Sơ đồ cấu tạo Keypad:

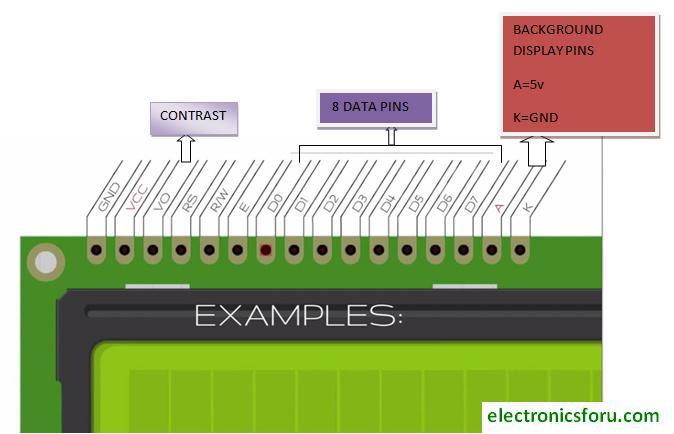


**Nguyên lý hoạt động**

Để làm việc với KeyPad 4×5, người lập trình thường sử dụng giải thuật “quét phím”. Giải thuật này yêu cầu VĐK liên tục đưa các tín hiệu đầu ra ở hàng (hoặc cột) và thu lại đầu vào ở cột (hoặc hàng), nếu phím được bấm, đầu phát tín hiệu sẽ được kết nối với đầu thu, từ đó xác định được phím đã bấm

**2.3. Màn hình LCD:**





|  |  |  |
| --- | --- | --- |
| **Stt** | **Ký hiệu** | **Mô tả** |
| 1 | Vss | Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển |
| 2 | VDD | Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển |
| 3 | VEE | Điều chỉnh độ tương phản của LCD. |
| 4 | RS | Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi.  + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read)  + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD. |
| 5 | R/W | Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc. |
| 6 | E | Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.  + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.  + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp. |
| 7 - 14 | DB0 - DB7 | Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :  + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.  + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7 |
| 15 | A | Nguồn dương cho đèn nền |
| 16 | K | GND cho đèn nền |

# ĐẶC TẢ HỆ THỐNG (SYSTEM REQUIREMENT)

**NAME:** Thiết bị máy tính cầm tay.

**PURPOSE:** Thiết bị có chức năng nhập và thực hiện các phép tính toán như cộng, trừ, nhân, chia, lũy thừa.

**INPUT/OUTPUT:**

* INPUT:
  + Phép tính nhập từ Keypad.
* OUTPUT:
  + Kết quả tính toán

**USE CASES:**

* Brief description: thực hiện tính toán và đưa ra kết quả.
* Basic description:
  + Khi phép tính đúng cú pháp, máy tính thực hiện và đưa kết quả ra màn hình
  + Khi phép tính sai cú pháp, máy tính hiển thị “Syntax error” ra màn hình, kết quả tràn số hiển thị “Math error”.
  + Máy tính tự động là tròn với các số thập phân dài hơn 15 ký tự ở phần thập phân
* Requirement: các thiết bị hoạt động chính xác và real-time.

**FUNCTION:** Cộng, trừ, nhân, chia, lũy thừa, số nguyên và số thập phân.

**PERFORMANCE:**

* Điện áp hoạt động: 5V – DC.
* Realtime delay: < 0.1s.
* Tuổi thọ: >2 năm.
* Dòng trung bình là 60mA, nếu sử dụng 2 pin 18650 3400mAh thì có thể dử dụng liên tục trong 56h.

**PHYSICAL SIZE, WEIGHT:**

* Size: 110mm x 165mm
* Weight: 100g

**INSTALLATION:**

* Thiết bị nhỏ gọn có thể cất giữ hay mang theo dễ dàng.
* Cấp nguồn điện cho thiết bị.
* Bảo quản khô ráo và tránh va đập để tránh bị hư hỏng.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

**4.1. System block diagram**

**LCD**

**DISPLAY**

**MCU**

**( Tiva TM4C123GXL )**

**Biến trở điều chỉnh độ sáng LCD**

**POWER**

**4.2. Block description**

**-** *Keypad* 4 x 5*:* Nhập số liệu và các phép tính

- *Biến trở:* Chức năng điều chỉnh độ sáng của màn hình LCD.

- *Power*: Cung cấp nguồn cho MCU, Keypad, LCD.

- *LCD Display*: Hiển thị các phép tính và kết quả.

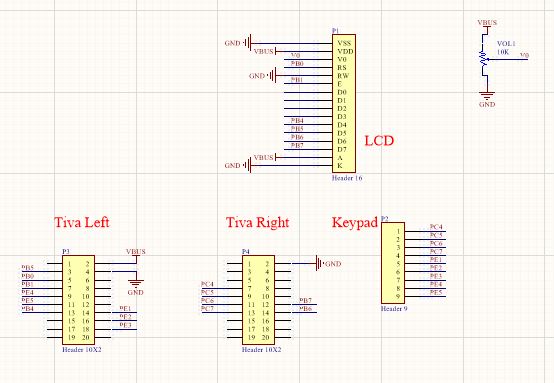
*- Vi xử lý Tiva TM4C123GXL*:

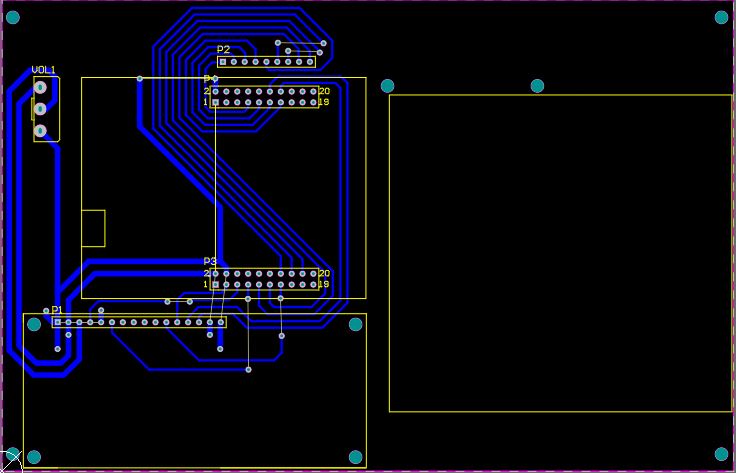
+ Xử lý đọc tín hiệu lệnh từ keypad.

+ Thực hiện các phép tính đọc từ Keypad.

+ Ghi ra LCD lệnh từ Keypad và kết quả phép tính (real – time).

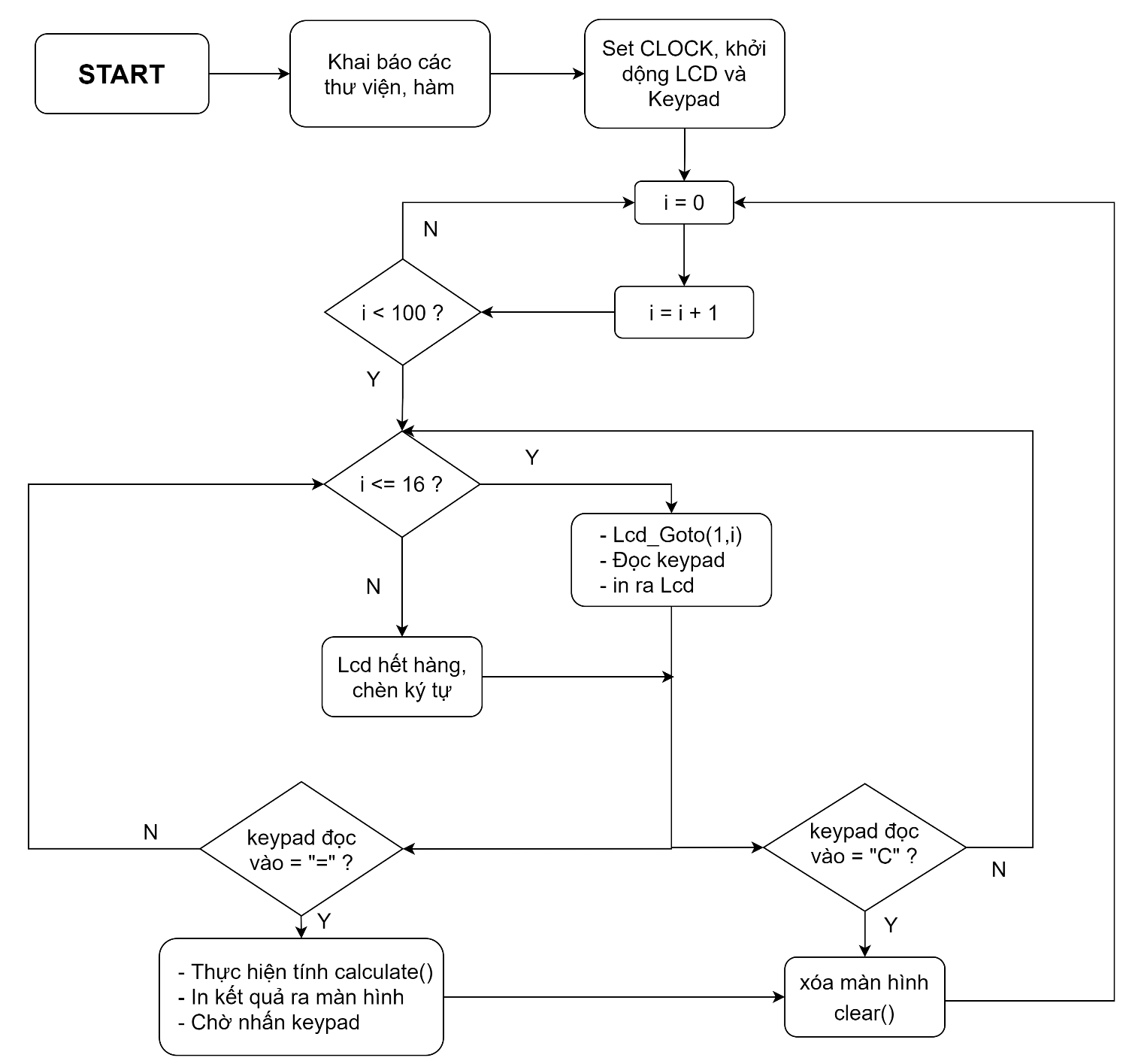
**Sơ đồ mạch điện:**





# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

**Sơ đồ giải thuật:**



**CODE:**

1. **KEYPAD:**

* **Khai báo Keypad:**

// Keypad.h

#ifndef KEYPAD\_KEYPAD\_H\_

#define KEYPAD\_KEYPAD\_H\_

#include "Delay.h"

#define padRows 5

#define padCols 4

char keypad\_getkey(void);

void keypad\_Init(void);

#endif /\* KEYPAD\_KEYPAD\_H\_ \*/

* **Hàm đọc Keypad**

---------------------

// [PE1 - PE5] -> [R1 - R5] Raws

// [PC4 - PC7] -> [C1 - C4] Cols

#include "Keypad.h"

#include "tm4c123gh6pm.h"

unsigned int ncols = 0;

unsigned int nrows = 0;

unsigned const char symbol[padRows][padCols] = {{ '+', '-', '\*', '/'},

{ '1', '2', '3', '^'},

{ '4', '5', '6', '('},

{ '7', '8', '9', ')'},

{ '.', '0', '=', 'C'}};

void keypad\_Init(void)

{

SYSCTL\_RCGCGPIO\_R |= 0x14; //enable clc for port C & E

while ((SYSCTL\_RCGCGPIO\_R&0x14)==0); //wait for clock to be enabled

GPIO\_PORTC\_CR\_R |= 0xF0; //allow changes to all the bits in port C

GPIO\_PORTE\_CR\_R |= 0x3E; //allow changes to all the bits in port E

GPIO\_PORTC\_DIR\_R |= 0xF0; //set directions cols are o/ps

GPIO\_PORTE\_DIR\_R |= 0x00; //set directions raws are i/ps

GPIO\_PORTE\_PDR\_R |= 0x3E; //pull down resistor on Raws

GPIO\_PORTC\_DEN\_R |= 0xF0; //digital enable pins in port C

GPIO\_PORTE\_DEN\_R |= 0x3E; //digital enable pins in port E

}

char keypad\_getkey(void)

{

while(1)

{

int i;

for(i = 0; i < 4; i++) //columns traverse

{

GPIO\_PORTC\_DATA\_R = (1U << i+4);

delay\_us(2);

int j;

for(j = 0; j < 5; j++) //raws traverse

{

if((GPIO\_PORTE\_DATA\_R & 0x3E) & (1U << j+1))

return symbol[4-j][i];

}

}

}

}

---------------------

1. **LCD:**

**Hàm khai báo LCD**

// LCD.h

#ifndef LCD\_LCD\_H\_

#define LCD\_LCD\_H\_

#define LCD\_PORT GPIO\_PORTB\_BASE

#define LCD\_PORT\_ENABLE SYSCTL\_PERIPH\_GPIOB

#define RS GPIO\_PIN\_0

#define E GPIO\_PIN\_1

#define D4 GPIO\_PIN\_4

#define D5 GPIO\_PIN\_5

#define D6 GPIO\_PIN\_6

#define D7 GPIO\_PIN\_7

void Lcd\_Temizle(void);

void Lcd\_Puts(char\*);

void Lcd\_Goto(char,char);

void Lcd\_init(void);

void Lcd\_Putch(unsigned char,int type);

#endif /\* LCD\_LCD\_H\_ \*/

**Hàm ghi LCD:**

---------------------

#include <stdbool.h>

#include <stdint.h>

#include <stdio.h>

#include <inttypes.h>

#include "inc/hw\_ints.h"

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "driverlib/sysctl.h"

#include "driverlib/gpio.h"

#include "LCD.h"

void Lcd\_init() {

SysCtlPeripheralEnable(LCD\_PORT\_ENABLE);

GPIOPinTypeGPIOOutput(LCD\_PORT, 0xFF);

SysCtlDelay(50000);

GPIOPinWrite(LCD\_PORT, RS, 0x00 );

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, 0x30 );

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, 0x30 );

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, 0x30 );

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, 0x20 );

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

Lcd\_Temizle();

Lcd\_Putch(0x28,0);//fonksiyon secimi

Lcd\_Putch(0x06,0);//kursörü saða kaydýr

Lcd\_Putch(0x80,0);//kursör konumu

Lcd\_Putch(0x0f,0);//kursörü aç

Lcd\_Temizle();

}

void Lcd\_Goto(char x, char y){

if(x==1)

Lcd\_Putch(0x80+((y-1)%16),0);

else

Lcd\_Putch(0xC0+((y-1)%16),0);

}

void Lcd\_Temizle(void){

Lcd\_Putch(0x01,0);

SysCtlDelay(10);

}

void Lcd\_Puts( char\* s){

while(\*s)

Lcd\_Putch(\*s++,1);

}

//komut ise 0 , karakter ise 1

void Lcd\_Putch(unsigned char c,int type) {

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, (c & 0xf0) );

if(type==1)GPIOPinWrite(LCD\_PORT, RS, 0x01);

else GPIOPinWrite(LCD\_PORT, RS, 0x00);

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCD\_PORT, D4 | D5 | D6 | D7, (c & 0x0f) << 4 );

if(type==1)GPIOPinWrite(LCD\_PORT, RS, 0x01);

else GPIOPinWrite(LCD\_PORT, RS, 0x00);

GPIOPinWrite(LCD\_PORT, E, 0x02);

SysCtlDelay(10);

GPIOPinWrite(LCD\_PORT, E, 0x00);

SysCtlDelay(50000);

}

---------------------

1. **Hàm tính toán:**

**Khai báo:**

**//** Tinh.h

#ifndef TINH\_TINH\_H\_

#define TINH\_TINH\_H\_

int error(char s[100]);

double xuly(char s[100]);

#endif /\* TINH\_TINH\_H\_ \*/

**Hàm tính:**

#include<stdio.h>

#include<math.h>

int ut(int i)

{

switch(i)

{

case '+': return 2;

case '-': return 2;

case '\*': return 3;

case '/': return 3;

case 'v': return 5;

case '^': return 4;

case '(': return 0;

case ')': return 1;

}

return 0;

}

double kq(int i, double a, double b)

{

switch(i)

{

case '+': return a+b;

case '-': return a-b;

case '\*': return a\*b;

case '/': return a/b;

case '^': return pow(a,b);

}

}

int error(char s[100])

{

int ng,i;

ng=strlen(s)-2;

if ((s[ng]<'0' || s[ng]>'9') && s[ng]!= ')') return 1;

ng=0;

for (i=0;i<strlen(s)-1;i++)

{

if (s[i]=='(') ng++;

if (s[i]==')') ng--;

if (ng < 0) return 1;

if (s[i] > '9' || s[i]<'0' && s[i] != ')')

if (s[i+1] =='\*' || s[i+1] == '/' || s[i+1] =='^'|| s[i+1]==')')

return 1;

if (s[i] <= '9' || s[i]>='0')

{

int dem=1;

while (s[i+dem] >= '0'&& s[i+dem] <='9') dem++;

if (s[i+dem] == '.')

{

if (dem>15) return 1;

i+=dem+1;

while (s[i+1] >= '0'&& s[i+1] <='9') i++;

if (s[i+1] == '.')return 1;

}

}

}

if (ng!=0) return 1;

else return 0;

}

double xuly(char s[100])

{

int sh[20],std=0,shd=0,dem,i;

double pn,tp,st[20];

if (s[0] == '-') st[++std]=0;

for (i=0;i<strlen(s);i++)

{

if (s[i]=='(')

{

if ((s[i-1] >= '0'&& s[i-1] <='9') || s[i-1] == ')') sh[++shd] ='\*';

if (s[i+1] == '+'|| s[i+1]=='-') st[++std]=0;

}

if (s[i] < '0'|| s[i] >'9')

{

if ((s[i]=='+' || s[i]=='-')

&&(s[i-1] =='\*' || s[i-1]=='/' ||s[i-1]=='^'))

{

dem=0;

if (s[i] == '-') dem++;

while (s[i+1]=='+' || s[i+1]=='-') if (s[++i]=='-') dem++;

if (dem%2==1) sh[++shd]='-';

else sh[++shd]='+';

st[++std]=0;

}

else

{

if (s[i] == '+' || s[i]=='-')

{

dem=0;

if (s[i] == '-') dem++;

while (s[i+1] == '+' || s[i+1]=='-')

if ( s[++i]=='-') dem++;

if (dem%2==1) s[i]='-';

else s[i]='+';

}

if ((shd==0 || ut(s[i]) >ut(sh[shd]) || s[i] =='(' )&& s[i] != ')') sh[++shd]=s[i];

else

{

while ((ut(s[i]) <= ut(sh[shd]))&& shd>0)

{

st[std-1]=kq(sh[shd],st[std-1],st[std]);

std--;

shd--;

}

if (s[i] == ')') shd--;

else sh[++shd]=s[i];

}

}

}

else

{

pn = s[i] - 48;

while (s[i+1] >= '0' && s[i+1] <='9') pn = pn\*10+s[++i]-48;

if (s[i+1]=='.')

{

i++;

tp=1;

while (s[i+1] >= '0'&& s[i+1] <='9')

{

tp = tp \* 10;

pn = pn + (s[++i]-48)/tp;

}

}

st[++std]=pn;

}

}

return st[1];

}

**Chương trình chính:**

**Hàm khai báo:**

// include.h

#ifndef INCLUDE\_H\_

#define INCLUDE\_H\_

#include <stdbool.h>

#include <stdint.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <stdio.h>

//thu vien driver API

#include "driverlib/gpio.h"

#include "driverlib/can.h"

#include "driverlib/eeprom.h"

#include "driverlib/i2c.h"

#include "driverlib/lcd.h"

#include "driverlib/mpu.h"

#include "driverlib/pwm.h"

#include "driverlib/sysctl.h"

#include "driverlib/pin\_map.h"

#include "driverlib/timer.h"

#include "driverlib/adc.h"

#include "driverlib/interrupt.h"

#include "driverlib/qei.h"

#include "driverlib/fpu.h"

#include "driverlib/uart.h"

#include "driverlib/rom.h"

//thu vien ho tro phan cung

#include "inc/hw\_ints.h"

#include "inc/hw\_gpio.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

//thu vien khai bao them (dua vao cac module minh tao ra)

//#include "GPIO/GPIO.h"

#endif /\* INCLUDE\_H\_ \*/

**Hàm thực hiện:**

**------**

#include "include.h"

#include "LCD/LCD.h"

#include "KEYPAD/Keypad.h"

#include "KEYPAD/Delay.h"

#include "Tinh/Tinh.h"

const double max=999999999999999,min=0.00000000000001;

char charc[100];

int i;

void clear()

{

int a;

Lcd\_Temizle();

for(a=0; a<100; a++)

{

charc[a]=0;

}

Lcd\_Goto(1,1);

i=0;

}

void print(double x)

{

int so,kytu=0;

double t=10;

char s[100];

if (x<0)

{

s[kytu++]='-';

x=-x;

}

while (t<=x) t\*=10;

do

{

t/=10;

so=(int)floor(x/t+min);

x= x -so\*t;

s[kytu++]=so+48;

}

while (t>1);

if ( x>0 && kytu<15 )

{

s[kytu++]='.';

while ( x>0 && kytu<16)

{

x \*=10;

so =(int) floor(x+min);

s[kytu++]=so+48;

x=x-so;

}

}

for (so = 0;so < kytu; so++) Lcd\_Putch(s[so],1);

}

void cclear(void)

{

delay\_ms(500);

keypad\_getkey();

clear();

}

void calculate(void)

{

Lcd\_Goto(2,1);

int kt = error(charc);

if (kt) Lcd\_Puts("Symtax Error");

else

{

double Kq = xuly(charc);

if (Kq>max) Lcd\_Puts("Math Error");

else print(Kq);

}

cclear();

}

void main(void)

{

SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_OSC\_MAIN | SYSCTL\_XTAL\_16MHZ);

Lcd\_init();

keypad\_Init();

while(1)

{

int j;

for(i = 1; i<100; i++)

{

if(i<=16)

{

Lcd\_Goto(1,i);

charc[i-1] = keypad\_getkey();

Lcd\_Putch(charc[i-1], 1);

if(charc[i-1] == '=')

{

calculate();

}

if(charc[i-1] == 'C')

{

clear();

}

delay\_ms(500);

}

else

{

Lcd\_Goto(1,16);

charc[i-1] = keypad\_getkey();

Lcd\_Goto(1,1);

Lcd\_Temizle();

for(j = i-15; j <= i; j++)

{

Lcd\_Putch(charc[j-1], 1);

}

if(charc[i-1] == '=')

{

calculate();

}

if(charc[i-1] == 'C')

{

clear();

}

delay\_ms(500);

}

}

}

}

**SƠ ĐỒ MẠCH:**

# KẾT QUẢ THỰC HIỆN



Kết quả thực hiện sản phẩm chạy gần như chính xác với yêu cầu ban đầu, thời gian phản hồi nhanh và độ chính xác cao.

**Đánh giá về kết quả làm việc nhóm: Các thành viên hoàn thành đúng thời gian và nhiệm vụ, sản phẩm đúng với dự tính và kết quả mô phỏng ban đầu**

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

1. Về kinh tế:
   * Thiết bị đơn giản, chi phí sản xuất có thấp và có thể tối giản hơn tùy vào nhu cầu sử dụng
   * Dễ dàng lắp đặt, vận hành
   * Công suất nhỏ, tiêu hao ít năng lượng
   * Thiết kế không quá khó nên mọi người đều có thể tự thiết kế sản phẩm cho mình
2. Về xã hội:
   * Nhu cầu tính toán luôn hiện hữu ở mọi nơi với tất cả mọi người nên một thiết bị đơn giản như thiết bị máy tính cầm tay là cần thiết

## Hướng phát triển

Ở đây nhóm chỉ thực hiện một vài phép tính cơ bản như cộng, trừ, nhân, chia, lũy thừa. Có thể kết nối thêm keypad để tăng thêm các chức năng tính toán khác nhiều hơn. Ngoài ra board Tiva TM4C123GXL còn nhiều chức năng mà nhóm chỉ mới sử dụng một phần, vì vậy lập trình viên cũng có thể xây dựng thêm nhiều chứng năng khác tùy theo nhu cầu của mình với đáp ứng của board.

# TÀI LIỆU THAM KHẢO

1. “TM4C123G\_LaunchPad\_Workshop\_Workbook.pdf”,

<http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/GSW-TM4C123G-LaunchPad/TM4C123G_LaunchPad_Workshop_Workbook.pdf>

1. “Giới thiệu cơ bản về LCD 16x2”, <http://thutemplate2115.blogspot.com/2015/08/d.html>
2. <https://www.datasheetarchive.com/KEYPAD+4x5-datasheet.html>
3. <https://github.com/>

---------------------------END---------------------------