ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN ĐIỆN TỬ

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

****

**BÀI TẬP LỚN VI XỬ LÝ**

**THIẾT KẾ MÁY TÍNH BỎ TÚI SỬ DỤNG**

**VI ĐIỀU KHIỂN 8051**

GVHD: Bùi Quốc Bảo

SVTH: Phạm Dũng Hoàng Duy - 1710808

Trần Thanh Dương - 1710911

TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2019

***LỜI CẢM ƠN***

Lời đầu tiên nhóm thực hiện đề tài xin gửi lời cảm ơn chân thành đến thầy hướng dẫn Bùi Quốc Bảo, người đã tận tình chỉ bảo, giúp đỡ về mặt chuyên môn và động viên nhóm trong suốt quá trình thực hiện đề tài.

Nhóm thực hiện đề tài cũng xin gửi lời cảm ơn đến các thầy cô, những người đã giảng dạy từ những buổi đầu giúp chúng em có đủ kiến thức để áp dụng thực hiện đề tài này.

Cảm ơn các bạn, những người đã góp ý, giúp đỡ và động viên nhóm hoàn thành đề tài.

Cuối cùng, nhóm thực hiện đề tài xin gửi lời biết ơn đến bố mẹ và những người trong gia đình đa luôn bên cạnh và ủng hộ chúng tôi trong cuộc sống.

*Tp. Hồ Chí Minh, ngày 20 tháng 05 năm 2019.*

Nhóm thực hiện đề tài

Phạm Dũng Hoàng Duy – Trần Thanh Dương

**TÓM TẮT**

Báo cáo này trình bày về toàn bộ quá trình của nhóm khi thực hiện đề tài “Thiết kế máy tính bỏ túi sử dụng vi điều khiển 8051” bao gồm:

* Phân chia công việc trong nhóm giữa các thành viên.
* Đưa ra thời gian thực hiện đề tài.
* Trình bày ý tưởng thiết kế phần cứng của sản phẩm. Đưa ra bảng số liệu các linh kiện được sử dụng để hoàn thiện sản phẩm.
* Trình bày giải thuật xử lý cũng như mã nguồn cho vi điều khiển.

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc9505946)

[1.1 Tổng quan 1](#_Toc9505947)

[1.2 Nhiệm vụ đề tài 1](#_Toc9505948)

[1.3 Phân chia công việc trong nhóm 1](#_Toc9505949)

[2. LÝ THUYẾT 2](#_Toc9505950)

[2.1 Giới thiệu về lịch sử vi điều khiển 8051 2](#_Toc9505951)

[2.2 Giới thiệu về kiến trúc vi điều khiển 8051 3](#_Toc9505952)

[2.3 Cơ cấu hiển thị LCD 1602 6](#_Toc9505953)

[3. YÊU CẦU HỆ THỐNG 7](#_Toc9505954)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 7](#_Toc9505955)

[4.1 Ghép nối với vi điều khiển 7](#_Toc9505956)

[4.2 Tổng quan sơ đồ nguyên lý 10](#_Toc9505957)

[5. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 10](#_Toc9505958)

[6. KẾT QUẢ THỰC HIỆN 20](#_Toc9505959)

[7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 21](#_Toc9505960)

[7.1 Kết luận 21](#_Toc9505961)

[7.2 Hướng phát triển 22](#_Toc9505962)

[8. TÀI LIỆU THAM KHẢO 22](#_Toc9505963)

[9. PHỤ LỤC 22](#_Toc9505964)

# GIỚI THIỆU

## Tổng quan

Máy tính bỏ túi là một công cụ học tập quen thuộc của mọi học sinh, sinh viên, đặc biệt là sinh viên kỹ thuật. Trong đời sống hàng ngày, nó là một vật cần thiết cho việc tính tiền trong việc buôn bán. Trong sản xuất, nó được dùng để tính toán số sản phẩm làm ra. Và còn rất nhiều ứng dụng khác nữa của máy tính bỏ túi, điều đó cho thấy sự phỏ biến và tầm quan trọng đặc biệt của nó trong thời đại hiện nay.

Được sự phân công và hướng dẫn tận tình của thầy **Bùi Quốc Bảo**. Nhóm đã hoàn thành xong đề tài “Thiết kế máy tính bỏ túi sử dụng vi điều khiển 8051”. Tuy nhiên, vì thời gian cũng như kiến thức có hạn nên sản phẩm của nhóm chỉ thực hiện được các phép tính đơn giản: cộng, trừ, nhân, chia, căn bậc hai, sin, cos.

## Nhiệm vụ đề tài

* Thiết kế máy tính bỏ túi sử dụng vi điều khiển 8051.
* Máy tính bỏ túi có các chức năng sau:
* Thực hiện được các phép tính cộng, trừ, nhân, chia, căn bậc hai, sin, cos.
* Có thể xử lí được 3 chữ số thập phân sau dấu phẩy (đối với các phép tính căn bậc hai, sin, cos).
* Bàn phím 4x5 bao gồm: 10 phím số (từ 0-9), 7 phím phép tính (cộng, trừ, nhân, chia, căn bậc hai, sin, cos), 1 phím xóa màn hình, 1 phím xóa ký tự, 1 phím bằng.
* Sử dụng LCD 1602 để hiển thị kết quả.

## Phân chia công việc trong nhóm

Sau khi nhận được đề tài, nhóm đã phân chia công việc cho các thành viên như sau:

|  |  |
| --- | --- |
| Công việc | Thực hiện |
| Thiết kế phần cứng | Phạm Dũng Hoàng Duy |
| Thiết kế mã nguồn chương trình | Trần Thanh Dương |

Mặc dù có sự phân chia công việc, nhưng các thành viên vẫn thường xuyên trao đổi, họp nhóm với nhau để hỗ trợ ý tưởng thực hiện cũng như giải quyết các vấn đề phát sinh trong quá trình thi công sản phẩm.

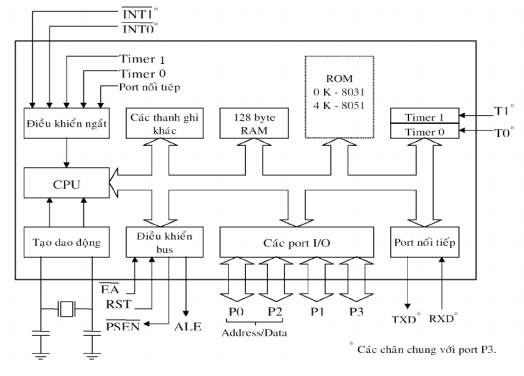
# LÝ THUYẾT

## Giới thiệu về lịch sử vi điều khiển 8051

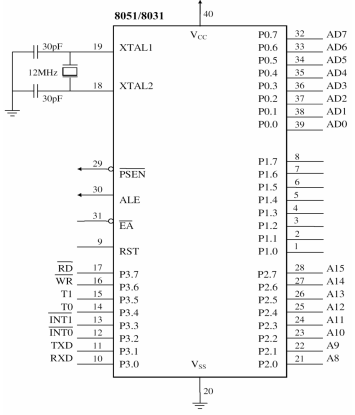
Vào năm 1981, hãng Intel đã giới thiệu một số bộ vi điều khiển được gọi là 8051. Bộ vi điều khiển này có 128 byte RAM, 4K byte dữ ROM trên chip, hai bộ định thời, một cổng nối tiếp và 4 cổng (đều rộng 8 bit) vào ra, tất cả đều được đặt trên một con chip. Lúc ấy, nó được coi là một “hệ thống trên chip”. 8051 là một bộ xử lí 8 bit, có nghĩa là CPU chỉ có thể làm việc với 8 bit dữ liệu tại một thời điểm. Dữ liệu lớn hơn 8 bit được chia thành các dữ liệu 8 bit để xử lý. 8051 có tất cả 4 cổng vào – ra I/O, mỗi cổng rộng 8 bit. Mặc dù 8051 có thể có một ROM trên chip cực đại là 64K byte, nhưng các nhà sản xuất lúc đó đã cho xuất xưởng chỉ với 4K byte ROM trên chip. Điều này sẽ được bàn chi tiết hơn sau này.

8051 đã trở nên phổ biến hơn sau khi Intel cho phép các nhà sản xuất khác sản xuất và bán bất kỳ dạng biến thể nào của 8051 với điều kiện họ phải để mã lại tương thích với 8051. Điều này dẫn đến sự ra đời của nhiều phiên bản 8051 với các tốc độ xử lí và dung lượng ROM trên chip khác nhau. Điều quan trong là có nhiều phiên bản khác nhau của 8051 nhưng tất cả chúng đều tương thích với 8051 lúc đầu về các lệnh. Điều này có nghĩa là khi ta lập trình cho một phiên bản 8051 nào đó thì nó có thể chạy với mọi phiên bản bất kỳ khác mà không phân biệt từ nhà sản xuất nào.

## Giới thiệu về kiến trúc vi điều khiển 8051



Hình 2.1 Sơ đồ khối của 8051



Hình 2.2 Sơ đồ chân của 8051

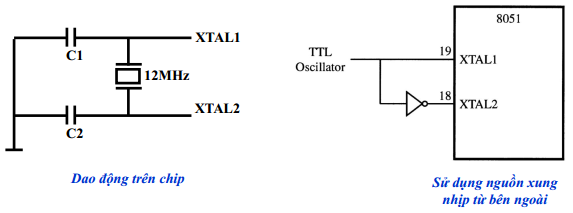
* **Sơ lược về các chân 8051:**
* Port 0 (P0.0 – P0.7): còn được ký hiệu là AD0 – AD7, có 2 chức năng:
* Xuất/Nhập: khi kit vi xử lý không sử dụng bộ nhớ ngoài.
* Data bus/Address bus (A0 – A7): khi kit vi xử lý sử dụng bộ nhớ ngoài.
* Port 1 (P1.0 – P1.7): chỉ có chức năng I/O giao tiếp với các thiết bị ngoài.
* Port 2 (P2.0 – P2.7): có 2 chức năng là xuất/nhập hoặc address bus phần cao.
* Port 3 (P3.0 – P3.7): xuất/nhập hoặc các chức năng đặc biệt khác.

Các chức năng đặc biệt của Port 3:

|  |  |  |
| --- | --- | --- |
| Bit | Tên | Chức năng |
| P3.0 | RxD | Ngõ thu nối tiếp |
| P3.1 | TxD | Ngõ phát nối tiếp |
| P3.2 | INT0 | Ngõ vào Interrupt 0 |
| P3.3 | INT1 | Ngõ vào Interrupt 1 |
| P3.4 | T0 | Ngõ vào Timer 0 |
| P3.5 | T1 | Ngõ vào Timer 1 |
| P3.6 | WR | Tín hiệu điều khiển ghi |
| P3.7 | RD | Tín hiệu điều khiển đọc |

(**Lưu ý**: Các port 1,2,3 đều có điện trở trong kéo lên, Port 0 thì chưa có)

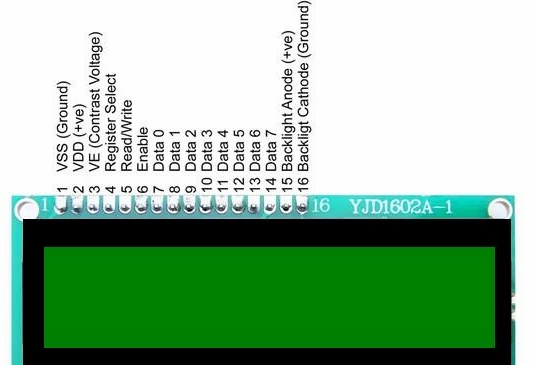
* Các ngõ vào bộ dao động (XTAL1 và XTAL2): kết nối với thạch anh theo sơ đồ như sau:



Hình 2.3 Hai cách kết nối chân XTAL với thạch anh

* Chân ALE (Address Latch Enable): được dùng để chốt tín hiệu địa chỉ. Thường được nối đến ngõ cho phép chốt của IC chốt (74373, 74374, 74573, 74574,…).
* Chân (Program Store Enable): điều khiển xuất ra của 8051, nó cho phép chọn bộ nhớ ngoài và được nối chung với chân OE (Output Enable) của EPROM ngoài để cho phép đọc các byte của chương trình.
* Chân (External Access): ngõ để lựa chọn truy xuất bộ nhớ trong hay ngoài (EA=0 để chọn bộ nhớ ngoài và EA=1 để chọn bộ nhớ trong).
* Chân RST (Reset): khi tín hiệu này ở mức cao (ít nhất trong 2 chu kỳ máy), các thang ghi bên trong 8051 được tải những giá trị thích hợp để khởi động lại hệ thống.
* Chân VCC: nối với nguồn +5V.
* Chân GND: nối đất (MASS, GND).
* Các đặc tính chủ yếu của 8051:
* Có 128 byte RAM nội, 4K ROM nội.
* Có khả năng quản lý 64K bộ nhớ chương trình và 64K bộ nhớ dữ liệu.
* Có 2 bộ Timer.
* Có khả năng thu/phát nối tiếp.
* Có khả năng interrupt.

## Cơ cấu hiển thị LCD 1602



Hình 2.4 LCD và thứ tự các chân

|  |  |  |
| --- | --- | --- |
| **Chân số** | **Tên** | **Chức năng** |
| 1 | VSS | Đất |
| 2 | VDD | Cực + của nguồn điện |
| 3 | VE | Tương phản |
| 4 | RS | Chọn thanh ghi (Register Select) |
| 5 | R/W | (Read/Write) |
| 6 | E | Cho phép (Enable) |
| 7 | D0 | Bit 0 của dữ liệu |
| 8 | D1 | Bit 1 của dữ liệu |
| 9 | D2 | Bit 2 của dữ liệu |
| 10 | D3 | Bit 3 của dữ liệu |
| 11 | D4 | Bit 4 của dữ liệu |
| 12 | D5 | Bit 5 của dữ liệu |
| 13 | D6 | Bit 6 của dữ liệu |
| 14 | D7 | Bit 7 của dữ liệu |
| 15 | A | Nguồn + cấp cho LED nền |
| 16 | K | Nguồn – cấp cho LED nền |

# YÊU CẦU HỆ THỐNG

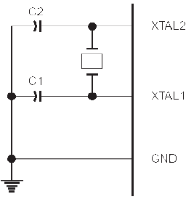
Hệ thống được thiết kế phải nhỏ gọn, không được quá cồng kềnh khó di chuyển. Các chức năng phải được đáp ứng đầy đủ như yêu cầu đặt ra ban đầu. Kết quả tính toán phải được hiển thị rõ ràng trên LCD.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## Ghép nối với vi điều khiển

* Mạch dao động thạch anh:

Thạch anh có nhiệm vụ tạo ra xung cho vi điều khiển làm việc (thường từ 8-24MHz đối với 8051), tần số dao động càng lớn thì tốc độ làm việc của vi điều khiển càng nhanh. Tuy nhiên, nếu vượt quá ngưỡng cho phép thì vi điều khiển sẽ bị đơ. Để biết chính xác giá trị thạch anh cho từng loại vi điều khiển, ta cần tra cứu kỹ trong Datasheet của nó. Ở đây ta sử dụng 2 tụ 33pF có nhiệm vụ ổn định xung nhip cho thạch anh, từ đó ổn định hoạt động cho vi điều khiển.



Hình 4.1 Mạch dao động thạch anh 12MHz

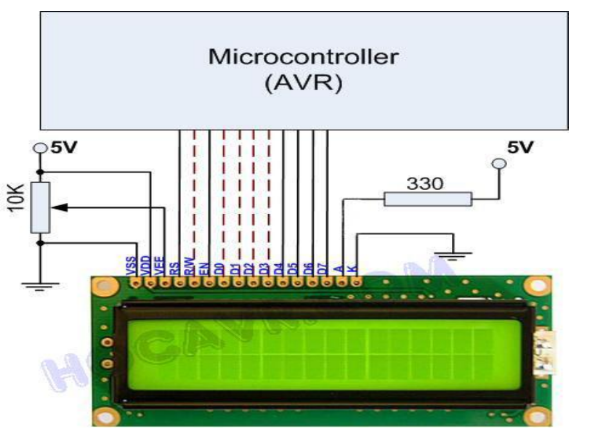
* Ma trận phím 5x4:

Như đã trình bày ở phần trên, ta cần sử dụng ma trận phím bấm 5x4 gồm 5 hàng (được đặt tên Row 1 – Row 5) và 4 cột (được đặt tên Col 1 – Col 4). Mỗi nút bấm có 2 chân. Chân thứ nhất của các nút trên cùng một hàng được nối với nhau, chân còn lại của các nút trên cùng một cột cũng được nối với nhau và tất cả được nối ra chân của vi điều khiển.

* Các chân từ Col 1 – Col 4 được nối đến P1.0 – P1.3.
* Các chân từu Row 1 – Row 4 được nối đến P1.4 – P1.7, riêng chân Row 5 được nối đến P3.0.
* Mỗi tọa độ được xác định bởi số hàng và cột sẽ được gán 1 chức năng như bảng sau:

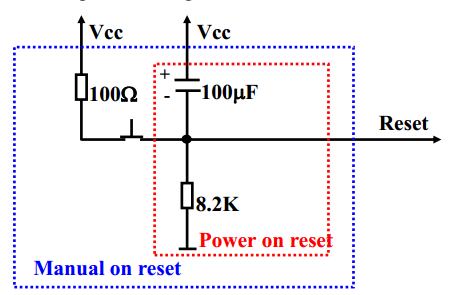
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Col 1 | Col 2 | Col 3 | Col 4 |
| Row 1 | 1 | 2 | 3 | 4 |
| Row 2 | 5 | 6 | 7 | 8 |
| Row 3 | 9 | 0 | C | DEL |
| Row 4 | + | - | x | / |
| Row 5 | Sin | Cos | Căn bậc 2 | = |

* Kết nối LCD 1602:



Hình 4.2 Sơ đồ ghép nối LCD 1602

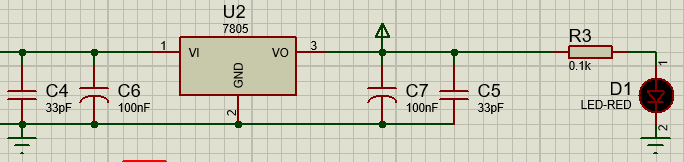
* Các chân D0 – D7 của LCD được nối đến P2.7 – P2.0.
* Chân VSS: được nối đất.
* Chân VDD: được nối lên nguồn 5V.
* Chân VEE: được nối đến biến trở 1K để chỉnh độ tương phản của LCD.
* Chân RS: được nối đến P3.1.
* Chân EN: được nối đến P3.2.
* Chân RW: được nối đất vì ta chỉ sử dụng chế độ ghi LCD.
* Chân A: được nối lên nguồn 5V.
* Chân K: được nối đất.
* Mạch RESET:



Hình 4.3 Sơ đồ mạch RESET

8051 được reset bằng cách giữ chân RST ở mức cao tối thiểu là 2 chu kỳ máy (lúc này VCC được nạp vào tụ) rồi sau đó chuyển về mức thấp (VCC từ chân RST được xả ra qua điện trở kéo xuống GND, điện trở này giúp ổn định mức 0 cho vi điều khiển, tránh việc bị reset liên tục).

* Mạch nguồn:



Hình 4.4 Sơ đồ mạch nguồn

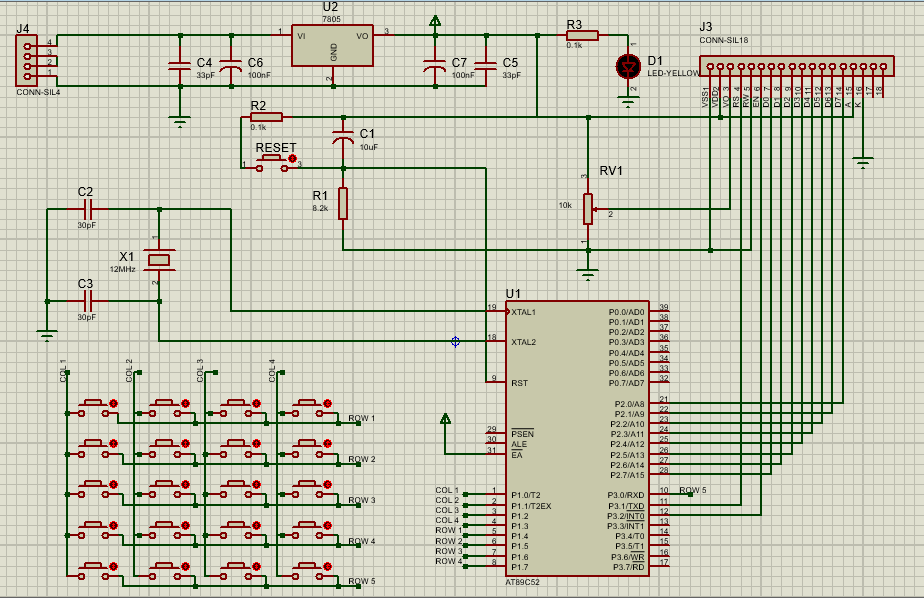
Ở đây ta dùng IC 7805 để ổn áp cho đầu ra 5V làm VCC cung cấp cho các linh kiện. Bên cạnh đó LED cũng được sử dụng để làm đèn nguồn.

* Vi điều khiển (Ở đây ta sử dụng AT89S52):

Ngoài các chân đã được mô tả cách kết nối như ở trên, ta cần kết nối một vài chân của vi điều khiển như sau:

* Chân VCC nối lên nguồn 5V.
* Chân GND nối đất.
* Chân EA nối nguồn vì ta sử dùng bộ nhớ nội để lưu chương trình.

## Tổng quan sơ đồ nguyên lý



Hình 4.5 Sơ đồ nguyên lý mạch thiết kế máy tính bỏ túi sử dụng vi điều khiển 8051

Trong sơ đồ trên, ta sử dụng JUMPER J3 16 chân để làm đế cắm chân cho LCD.

# 5. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

* Yêu cầu đặt ra cho phần mềm:
  + Xử lí được các phép tính cộng trừ nhân chia sin cos căn bậc 2 với số nguyên.
  + Hiển thị nhiều chữ số nhất có thể.
  + Cho phép nhập nhiều chữ số nhất có thể.
  + Xử lí được các lỗi đầu ra khi người dùng nhập biểu thức không phù hợp.
* Phân tích
  + Phân tích các yêu cầu để đưa ra phương pháp thực hiện chương trình:
    - Yêu cầu 1: xử lí được các phép tính cộng trừ nhân chia sin, cos, căn bậc hai. Ta chia yêu cầu ra làm hai phần là xử lí các phép tính có 2 số hạng ( cộng trừ nhân chia) và một số hạng ( sin cos căn ). Các hàm tính toán sẽ trả về kiểu dữ liệu là số nguyên do LCD chỉ hiện được số nguyên.
    - Yêu cầu 2: hiển thị nhiều chữ số nhất có thể: kết quả sẽ cố gắng thể hiện nhiều chữ số thập phân nhất có thể, nhưng do giới hạn về không gian chứa của ROM nội (nhóm không mở rộng bộ nhớ ngoài), cũng như giới hạn của kiểu dữ liệu nên kết quả trả về sẽ như sau:

|  |  |
| --- | --- |
| **Phép tính** | **Miền giá trị kết quả** |
| Cộng trừ | Số có dấu từ -1998 đến 1998 |
| Nhân | Số có dấu từ -9801 đến 9801 |
| Chia | Số thập phân lấy 2 chữ số, giá trị từ -99.00 đến 99.00 |
| Sin, cos | Số thập phân lấy 3 chữ số, giá trị từ -1.000 đến 1.000 |
| Căn bậc hai | Sô thập phân lấy 3 chữ số, giá trị từ -31.623 đến 31.623 ( ) |

* + - Yêu cầu 3: cho phép nhập nhiều chữ số nhất có thể. Cũng giống yêu cầu 2, do hạn chế về phần cứng nên phần mềm cũng giá trị số nhập vào, nhóm chỉ xử lí số nguyên.

|  |  |
| --- | --- |
| **Phép tính** | **Miền giá trị nhập vào** |
| Cộng trừ | Số nguyên trong khoảng từ -999 đến 999 |
| Nhân chia | Số nguyên trong khoảng từ -99 đến 99 |
| Sin, cos, căn bậc hai | Sô nguyên trong khoảng -999 đến 999, riêng phép tính căn sẽ báo lỗi với số âm |

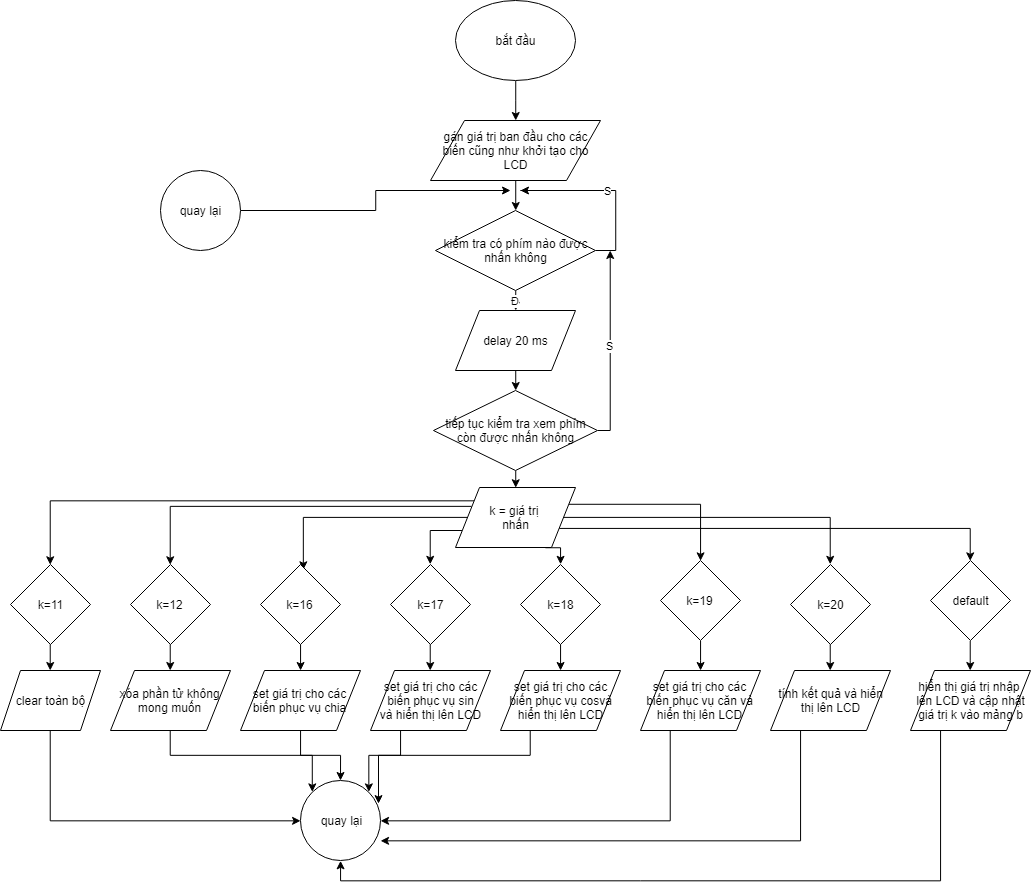
* + - Yêu cầu 4: xử lí các lỗi do người dùng nhập sai cách: các lỗi nhóm đưa ra có 2 lỗi:

+ Lỗi 1: sai cú pháp : xảy ra khi người dùng nhập sai cú pháp đúng của phép tính .

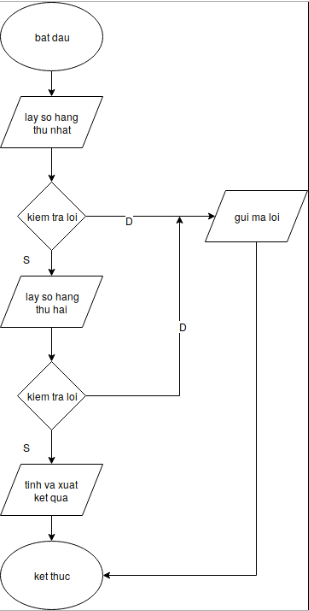
+ Lỗi 2: lỗi tính toán: xảy ra khi người dùng nhập số liệu vào quá miền cho phép hoặc nhập phép tính vô lí ( chia cho 0, căn số âm )

+ Ngoài ra nhóm còn có cảnh báo khi người dùng không nhập giá trị cho phép tính.

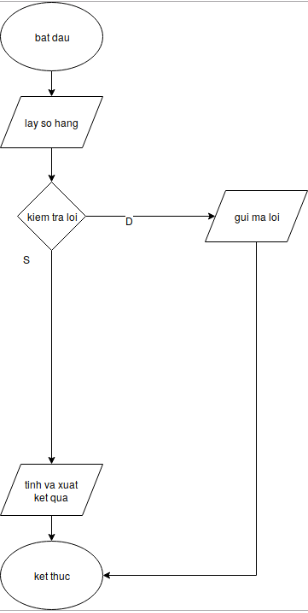
* + - Ngoài ra nhóm còn có các phím chức năng là C và DEL giúp xóa màn hình cũng như xóa kí tự không mong muốn.
* Các lưu đồ giải thuật sử dụng trong chương trình:



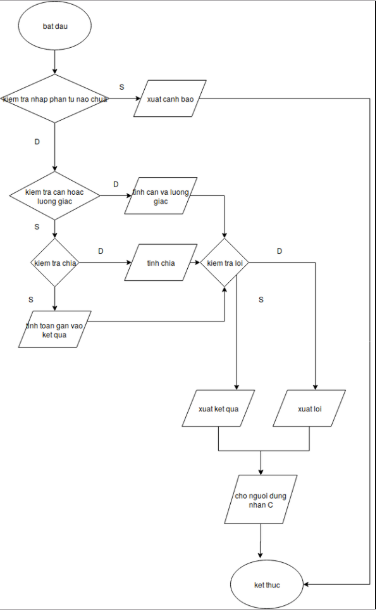
Hình 5.1 Lưu đồ giải thuật tổng quát



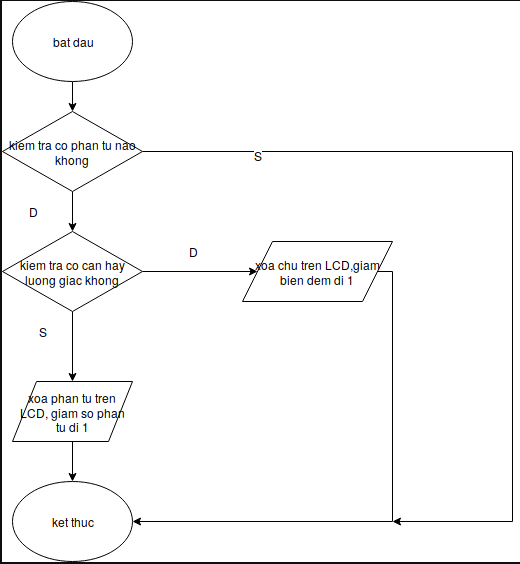
Hình 5.2 Lưu đồ giải thuật tính cộng, trừ, nhân, chia



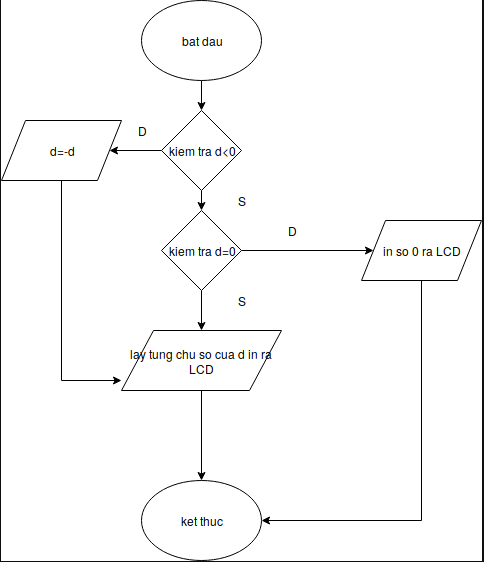
Hình 5.3 Lưu đồ giải thuật tính sin, cos, căn bậc hai



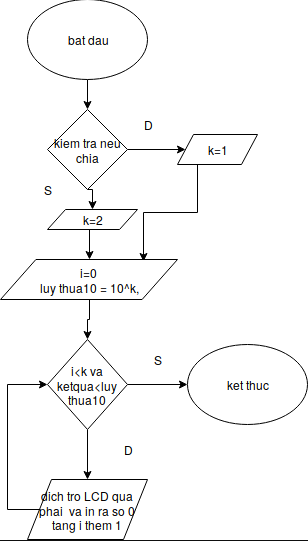
Hình 5.4 Lưu đồ giải thuật bấm phím “=’



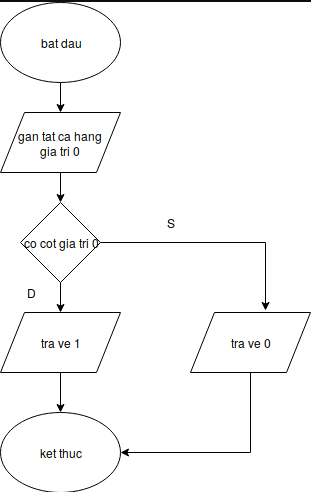
Hình 5.5 Lưu đồ giải thuật bấm phím C (Clear)



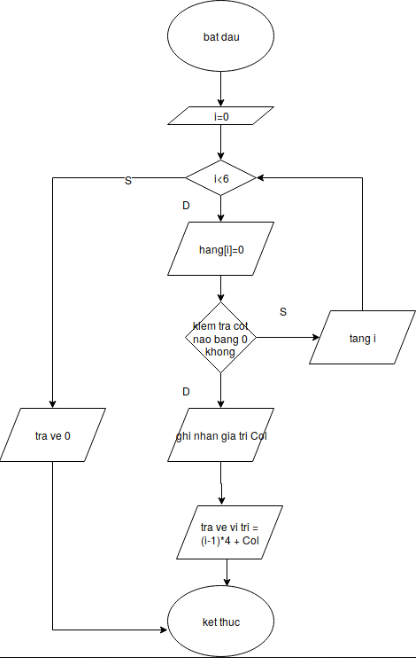
Hình 5.6 Lưu đồ giải thuật in số ra màn hình LCD



Hình 5.7 Lưu đồ giải thuật hiệu chỉnh phần thập phân



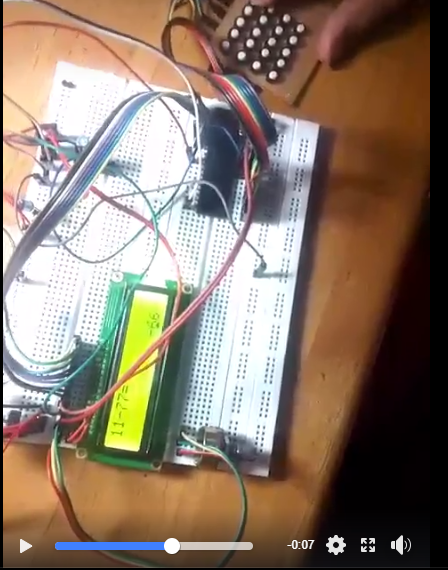
Hình 5.8 Lưu đồ giải thuật kiểm tra có phím nào được nhấn hay không



Hình 5.9 Lưu đồ giải thuật kiểm tra vị trí phím được nhấn

# 6. KẾT QUẢ THỰC HIỆN

* Trước khi tiến hành làm mạch in, nhóm đã cắm thử mạch trên breadboard để thử nghiệm:



Hình 6.1 Kết quả thử nghiệm khi cắm mạch trên Breadboard

* Sau khi thử nghiệm thành công trên breadboard, nhóm tiến hành làm mạch in:



Hình 6.2 Mạch máy tính bỏ túi đã hoàn thiện

* Phần mềm sử dụng để lập trình:
* Keil uVision5: viết chương trình cho vi xử lý (chương trình được viết bằng ngôn ngữ C).
* Proteus 8 Professional: vẽ sơ đồ nguyên lý và layout để thực hiện mô phỏng và hàn mạch in.
* Đánh giá về kết quả làm việc nhóm :

Các thành viên trong nhóm tích cực tham gia hoàn thành nhiệm vụ được giao, đồng thời góp ý bổ sung cho nhau để chất lượng sản phẩm được hoàn thiện hơn, đáp ứng được thời hạn hoàn thành thiết kế và mục tiêu ban đầu của đề tài đặt ra.

# 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 7.1 Kết luận

Sau khi thực hiện xong đề tài, các thành viên trong nhóm cũng đã có thêm cho mình nhiều kinh nghiệm về hàn mạch in, cắm thử breadboard để thử nghiệm mạch, được tiếp cận với các linh kiện điện tử trong thực tế.

Đánh giá ưu và khuyết điểm của sản phẩm:

* Ưu điểm:
* Mạch chạy tốt, đáp ứng được các yêu cầu ban đầu đặt ra.
* Trong quá trình thi công mạch, nhóm đã gặp phải vài khó khăn (cắm breadboard và hàn mạch in ban đầu không chạy) tuy nhiên các thành viên trong nhóm đã khắc phụ được.
* Khuyết điểm:
* Cần trau dồi thêm kỹ năng hàn mạch, sắp xếp linh kiện vẽ layout.
* Các linh kiện được sắp xếp trên mạch chưa thực sự tối ưu.

## 7.2 Hướng phát triển

Sản phẩm làm ra tuy đã đáp ứng được các yêu cầu đặt ra ban đầu song vẫn còn nhiều hạn chế. Trong tương lai nhóm sẽ nâng cấp cho sản phẩm thực hiện tính toán được với nhiều chữ số hơn và các phép tính phức tạp hơn.

Đề tài mang tính thực tiễn cao, đáp ứng được nhiều yêu cầu tính toán đặt ra trong cuộc sống, giá thành tạo ra sản phẩm tương đối thấp, có thể hoàn thiện thêm về chức năng và mẫu mã để thương mại hóa sản phẩm, đem lại lợi ích về kinh tế.

# 8. TÀI LIỆU THAM KHẢO

1. Lê Ngọc Bích – Phạm Quang Huy, “Ứng dụng vi xử lý và vi điều khiển”, Nhà xuất bản Bách Khoa Hà Nội.
2. Bùi Quốc Bảo, “Tài liệu thí nghiệm vi xử lý”, Bộ môn Điện tử Đại học Bách Khoa thành phố Hồ Chí Minh, 2018.
3. Hồ Trung Mỹ, “Giáo trình vi xử lý”.
4. Nguồn tham khảo trên Internet.

# 9. PHỤ LỤC

Mã nguồn của chương trình:

|  |
| --- |
| #include <string.h> // dinh kem thu vien chuoi  #include <at89x51.h> // dinh kem thu vien at89x51  #include <stdio.h>  #include <math.h>  #define pi 3.14159265  #define ma\_loi\_1 20000  #define ma\_loi\_2 20001  #define loi\_1 "Sai cu phap!"  #define loi\_2 "Loi tinh toan!"  #define khoangtrang " "  sfr LCDdata=0xA0; // dung P2 lam port du lieu  sbit RS=P3^1; // chan RS noi voi P3.1  sbit EN=P3^2; // chan EN noi voi P3.2    sbit c1=P1^0;  sbit c2=P1^1;  sbit c3=P1^2;  sbit c4=P1^3;    sbit h1=P1^4;  sbit h2=P1^5;  sbit h3=P1^6;  sbit h4=P1^7;  sbit h5=P3^0;  unsigned char LCD\_code[16]={'1','2','3','4','5','6','7','8','9','0',' ',' ','+','-','\*','/'};  // phat hien co nut nhan hay khong  unsigned char KeyPressed()  {  h1=h2=h3=h4=h5=0;  if((c1==0)||(c2==0)||(c3==0)||(c4==0))  return 1;  return 0;  }  // phat hien cot duoc nhan  unsigned char ScanCol()  {  unsigned char c=0;  if(c1==0) c=1;  else if(c2==0) c=2;  else if(c3==0) c=3;  else if(c4==0) c=4;  return c;  }  // phat hien hang duoc nhan  unsigned char ScanRow(int hang)  {  h1=h2=h3=h4=h5=1;  if(hang==1) h1=0;  else if(hang==2) h2=0;  else if(hang==3) h3=0;  else if(hang==4) h4=0;  else if(hang==5) h5=0;  return 0;  }  // tinh toan vi tri phim duoc nhan  unsigned char Key\_code()  {  unsigned char i,Col;  for(i=1;i<6;i++)  {  ScanRow(i);  Col=ScanCol();  if(Col>0) break;  }  return ((i-1)\*4 + Col);  }  void delayms(unsigned int time)  {  while(time--);  }  void out2LCD(unsigned char x,bit y)  {  unsigned char i=0;  unsigned char tam=0;  for(;i<8;i++)  {  tam>>=1;  tam=tam|(x&0x80);  x<<=1;  }  LCDdata=tam;// xuat gia tri ra Port du lieu  EN=1;  RS=y; // RS=0: che do LCD nhan lenh  delayms(1000);  EN=0;  }  void init\_LCD(void)  {  out2LCD(0x38,0);  out2LCD(0x0E,0);  out2LCD(0x06,0);  out2LCD(0x01,0);  }  void gotoxy(unsigned char x,bit y)  {  if(y==1)  {  out2LCD(192+x,0);  return;  }  out2LCD(128+x,0);  }  void LCD\_num(int d)  {  if(d<0)  {  LCD\_num(-d);  out2LCD(0x10,0);  out2LCD('-',1);  }else if(d>0){  while(d>0)  {  out2LCD(0x10,0);  out2LCD(d%10+48,1);  out2LCD(0x10,0);  d=d/10;  }  }else if(d==0)  out2LCD('0',1);  }  void LCD\_string(char \*str)  {  unsigned char i=0;  while(str[i]!=0)  {  out2LCD(str[i],1);  i++;  }  }  int cong\_tru\_nhan\_chia(unsigned char so\_phan\_tu,unsigned char b[10])  {  int sh1=0;  int sh2=0;  unsigned char dau=0;  unsigned char i=0;  bit soduong=1;  if(b[so\_phan\_tu-1]>10)  return ma\_loi\_1;  if(b[0]==14)  {  soduong=0;  i=1;  }  while((b[i]<11)&&(i<so\_phan\_tu))  {  sh1=sh1\*10+b[i]%10;  if(sh1>999)  return ma\_loi\_2;  i++;  }  sh1=sh1\*(2\*soduong-1);  if(i==so\_phan\_tu)  return sh1;  dau=b[i];  i++;  if(b[i]==14)  {  soduong=0;  i++;  }else  soduong=1;  while((i<so\_phan\_tu)&&(b[i]<=10))  {  sh2=10\*sh2+b[i]%10;  if(sh2>999)  return ma\_loi\_2;  i++;  }  if(i!=so\_phan\_tu)  return ma\_loi\_1;  sh2=sh2\*(2\*soduong-1);  switch(dau)  {  case 13:  return sh1+sh2;  case 14:  return sh1-sh2;  case 15:  if((abs(sh1)>99)||(abs(sh2)>99))  return ma\_loi\_2;  return sh1\*sh2;  case 16:  if((sh2==0)||(abs(sh1)>99)||(abs(sh2)>99))  return ma\_loi\_2;  return sh1\*100/sh2;  default:  return sh1;  }  }  int tinh\_luong\_giac\_va\_can(unsigned char so\_phan\_tu,unsigned char b[10],bit ktsin,bit can,unsigned char dem)  {  int sh=0;  unsigned char i=0;  bit soduong=1;  if(dem!=1)  return ma\_loi\_1;  if(b[0]==14)  {  if(can)  return ma\_loi\_2;  soduong=0;  i=1;  }  while((i<so\_phan\_tu)&&(b[i]<=10))  {  sh=sh\*10+b[i]%10;  if(sh>999)  return ma\_loi\_2;  i++;  }  if(i!=so\_phan\_tu)  return ma\_loi\_1;  sh=sh\*(2\*soduong-1);  if(can)  return (int) 1000\*sqrt(sh);  if(ktsin)  return (int) 1000\*sin(sh\*pi/180);  return (int) 1000\*cos(sh\*pi/180);  }  bit kiem\_tra\_loi(int ketqua)  {  if(ketqua==ma\_loi\_1)  {  gotoxy(1,1);  LCD\_string(loi\_1);  return 1;  }  if(ketqua==ma\_loi\_2)  {  gotoxy(1,1);  LCD\_string(loi\_2);  return 1;  }  return 0;  }  void hieu\_chinh\_thap\_phan(int ketqua,bit chia)  {  unsigned char k=2;  unsigned char i=0;  unsigned char luythua10=1;  if(chia)  k=1;  for(;i<k;i++)  luythua10=10\*luythua10;  for(i=0;i<=k;i++)  {  if(ketqua<luythua10)  {  out2LCD(0x10,0);  out2LCD('0',1);  out2LCD(0x10,0);  }  luythua10=luythua10/10;  }  }  // chuong trinh chinh  void main()  {  unsigned char b[10];  unsigned char i;  unsigned char so\_phan\_tu=0;  unsigned char k;  unsigned char dem=0;  int ketqua;  bit luonggiac=0;  bit sin;  bit can=0;  bit chia=0;  bit cho\_phep=1;  init\_LCD();  while(1)  {  if(KeyPressed())  delayms(20000);  if(KeyPressed())  {  k=Key\_code();  if(!cho\_phep)  {  while(k!=11)  k=Key\_code();  }  switch(k)  {  case 11:  init\_lCD();  for(i=0;i<10;i++)  b[i]=0;  so\_phan\_tu=0;  luonggiac=0;  can=0;  chia=0;  dem=0;  cho\_phep=1;  break;  case 12:  if(so\_phan\_tu==0)  {  if(dem!=0)  {  gotoxy((dem-1)\*4,0);  LCD\_string(khoangtrang);  gotoxy((dem-1)\*4,0);  dem--;  break;  }else  break;  }  out2LCD(0x10,0);  out2LCD(' ',1);  out2LCD(0x10,0);  so\_phan\_tu--;  break;  case 17:  dem++;  luonggiac=1;  sin=1;  LCD\_string("sin(");  break;  case 18:  dem++;  luonggiac=1;  sin=0;  LCD\_string("cos(");  break;  case 19:  dem++;  can=1;  LCD\_string("can(");  break;  case 20:  if(so\_phan\_tu==0)  {  gotoxy(1,1);  LCD\_string("Nhap so vao");  delayms(30000);  gotoxy(1,1);  LCD\_string(" ");  if(can||luonggiac)  gotoxy(4,0);  else  gotoxy(0,0);  break;  }  cho\_phep=0;  if(luonggiac||can)  {  LCD\_string(")=");  ketqua=tinh\_luong\_giac\_va\_can(so\_phan\_tu,b,sin,can,dem);  if(kiem\_tra\_loi(ketqua))  break;  if(ketqua/1000==0)  gotoxy(10,1);  else  gotoxy(11,1);  LCD\_num(ketqua/1000);  ketqua=ketqua%1000;  gotoxy(11,1);  out2LCD('.',1);  gotoxy(15,1);  LCD\_num(ketqua);  hieu\_chinh\_thap\_phan(ketqua,chia);  }else if(chia)  {  ketqua=cong\_tru\_nhan\_chia(so\_phan\_tu,b);  out2LCD('=',1);  if(kiem\_tra\_loi(ketqua))  break;  if(ketqua/100==0)  gotoxy(11,1);  else  gotoxy(12,1);  LCD\_num(ketqua/100);  ketqua=ketqua%100;  gotoxy(12,1);  out2LCD('.',1);  gotoxy(15,1);  LCD\_num(ketqua);  hieu\_chinh\_thap\_phan(ketqua,chia);  }else  {  ketqua=cong\_tru\_nhan\_chia(so\_phan\_tu,b);  out2LCD('=',1);  if(kiem\_tra\_loi(ketqua))  {  break;  }  gotoxy(15,1);  LCD\_num(ketqua);  }  break;  case 16:  chia=1;  default:  out2LCD(LCD\_code[k-1],1);  b[so\_phan\_tu]=k;  so\_phan\_tu++;  break;  }  }  }  } |