

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

HỌ VÀ TÊN SINH VIÊN

TRẦN LÊ MINH ĐĂNG

BÁO CÁO ĐỒ ÁN MÔN HỌC GIỮA KỲ

ĐÈN GIAO THÔNG SỬ DỤNG VI ĐIỀU KHIỂN 8051

GIẢNG VIÊN HƯỚNG DẪN

TS. Đỗ Trí Nhựt

THÀNH PHỐ HỒ CHÍ MINH, 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ
CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
ĐỘC LẬP – TỰ DO – HẠNH PHÚC

CHI TIẾT

TÊN ĐỒ ÁN TIẾNG VIỆT: ĐÈN GIAO THÔNG SỬ DỤNG VI ĐIỀU KHIỂN 8051	
TÊN ĐỒ ÁN TIẾNG ANH: TRAFFIC LIGHT USING AN 8051 MICROCONTROLLER	
Giảng viên hướng dẫn: TS. Đỗ Trí Nhựt, Khoa Kỹ thuật máy tính	
Thời gian thực hiện: Từ ngày: 13/2/2023 Đến ngày: 23/4/2023	
Sinh viên thực hiện: Trần Lê Minh Đăng	
Phương pháp thực hiện: Mô phỏng	
Ngôn ngữ lập trình: C	
Phần mềm mô phỏng: Proteus	
Chứng nhận của giảng viên (chữ ký và họ tên đầy đủ)	Tp. Hồ Chí Minh, 23/4/2023 Sinh viên (chữ ký và họ tên đầy đủ)

MỤC LỤC

PHẦN 1: GIỚI THIỆU ĐỀ TÀI	1
1.1 Giới thiệu	1
1.1.1 Chức năng của hệ thống	1
1.1.2 Hoạt động của hệ thống	1
1.2 Phân tích các khối chức năng của hệ thống	2
1.2.1 Khối xử lý trung tâm	2
1.2.2 Bàn phím	2
1.2.3 Nút nhấn	2
1.2.4 Khối nguồn	2
1.2.5 Khối hiển thị	3
1.3 Giới thiệu về vi điều khiển 8051	3
1.3.1 Sơ lược về chân của IC 89C51	3
1.4 Giới thiệu về bàn phím	4
1.5 Giới thiệu về đèn giao thông	4
1.6 Giới thiệu led 7 đoạn loại dương chung	5
PHẦN 2: THIẾT KẾ VÀ THI CÔNG MẠCH	6
2.1 Thiết kế	6
2.1.1 Khối điều khiển trung tâm	6
2.1.2 Nút nhấn	6
2.1.3 Bàn phím	6
2.1.4 Khối mạch điều khiển	7
2.2 Thi công mạch	7
PHẦN 3: LƯU ĐỒ THUẬT TOÁN VÀ CÁC CHƯƠNG TRÌNH	7
3.1 Lưu đồ thuật toán chương trình chính	7

3.2 Lưu đồ thuật toán của các chương trình con	8
3.2.1 Thuật toán quét led	8
3.2.2 Thuật toán quét phím.....	9
3.2.3 Thuật toán hiển thị thời gian đèn giao thông	10
3.2.4 Thuật toán điều chỉnh thời gian ban đầu bằng bàn phím	11
3.2.5 Thuật toán hiển thị thời gian mặc định.....	12
3.3 Chương trình	13
PHẦN 4: THỰC NGHIỆM.....	19
4.1 Môi trường thực nghiệm.....	19
4.2 Kết quả thu được.....	20
PHẦN 5: ĐÁNH GIÁ.....	20
5.1 Giới hạn số lần điều chỉnh thời gian.....	20
5.2 Thời gian đáp ứng	20
5.3 Một số hạn chế còn tồn tại.....	20
5.4 Kết luận	20
TÀI LIỆU THAM KHẢO	20

DANH MỤC HÌNH ẢNH

Hình 1 Các khối chức năng của hệ thống	2
Hình 2 Sơ đồ chân của vi điều khiển 8051	3
Hình 3 Hình ảnh bàn phím.....	4
Hình 4 Hình ảnh đèn giao thông.....	4
Hình 5 Hình ảnh led 7 đoạn 2 chữ số dương chung	5
Hình 6 Sơ đồ mạch thiết kế.....	6
Hình 7 Lưu đồ thuật toán chương trình chính	7
Hình 8 Lưu đồ thuật toán quét led	8
Hình 9 Lưu đồ thuật toán quét phím.....	9
Hình 10 Lưu đồ thuật toán hiển thị đèn giao thông	10
Hình 11 Lưu đồ thuật toán điều chỉnh thời gian bằng bàn phím	11
Hình 12 Lưu đồ thuật toán hiển thị thời gian mặc định.....	12

PHẦN 1: GIỚI THIỆU ĐỀ TÀI

1.1 Giới thiệu

Ngày nay cùng với sự phát triển của khoa học kỹ thuật, xã hội ngày càng văn minh và hiện đại, các đô thị ngày một đi lên. Nhu cầu về giao thông ngày càng trở nên cấp thiết, nhất là trong các khu vực thành thị. Do nhu cầu của đời sống con người, đặc biệt là nhu cầu đi lại, các loại phương tiện giao thông đã tăng một cách chóng mặt. Riêng tại Việt Nam số lượng xe máy trong những năm qua tăng một cách đột biến, mật độ xe lưu thông trên đường ngày một nhiều, trong khi đó hệ thống đường xá tại Việt Nam còn quá nhiều hạn chế nên thường gây ra các hiện tượng như kẹt xe, ách tắc giao thông, đặc biệt là tai nạn giao thông ngày càng phổ biến trở thành mối hiểm họa cho nhiều người.

Vì lý do đó các luật giao thông lần lượt ra đời và được đưa vào sử dụng phổ biến như hiện nay. Trong đó hệ thống đèn giao thông là công cụ điều khiển giao thông công cộng thực tế và hiệu quả có vai trò rất lớn trong việc đảm bảo an toàn và giảm thiểu tai nạn giao thông.

Với yêu cầu khi hệ thống chạy, các đèn xanh, vàng và đỏ sáng trong thời gian mặc định tương ứng là 30s, 2s và 20s. Ngoài ra còn có bàn phím cho phép điều chỉnh thời gian ban đầu cho các đèn.

1.1.1 Chức năng của hệ thống

- Hệ thống gồm 2 cột đèn giao thông, mỗi cột gồm 3 đèn: xanh, vàng, đỏ; 2 cột đèn sẽ hoạt động đồng bộ với nhau.
- Có hệ thống phím số gồm 12 phím có tác dụng chỉnh thời gian ban đầu cho các đèn

1.1.2 Hoạt động của hệ thống

- Hệ thống sẽ có 2 nút nhấn:
 - + 1 nút BUTTON 1 có tác dụng khởi động đèn ở chế độ mặc định
 - + 1 nút BUTTON 2 có tác dụng thiết lập lại trạng thái ban đầu của hệ thống
- 2 cột đèn hoạt động đồng bộ với nhau:

Giả sử ta nạp vào 3 giá trị Đỏ, Xanh, Vàng. Khi đó giá trị đèn ở mỗi cột là:

 - + Đỏ 1 = Đỏ
 - + Xanh 1 = Xanh
 - + Vàng 1 = Vàng

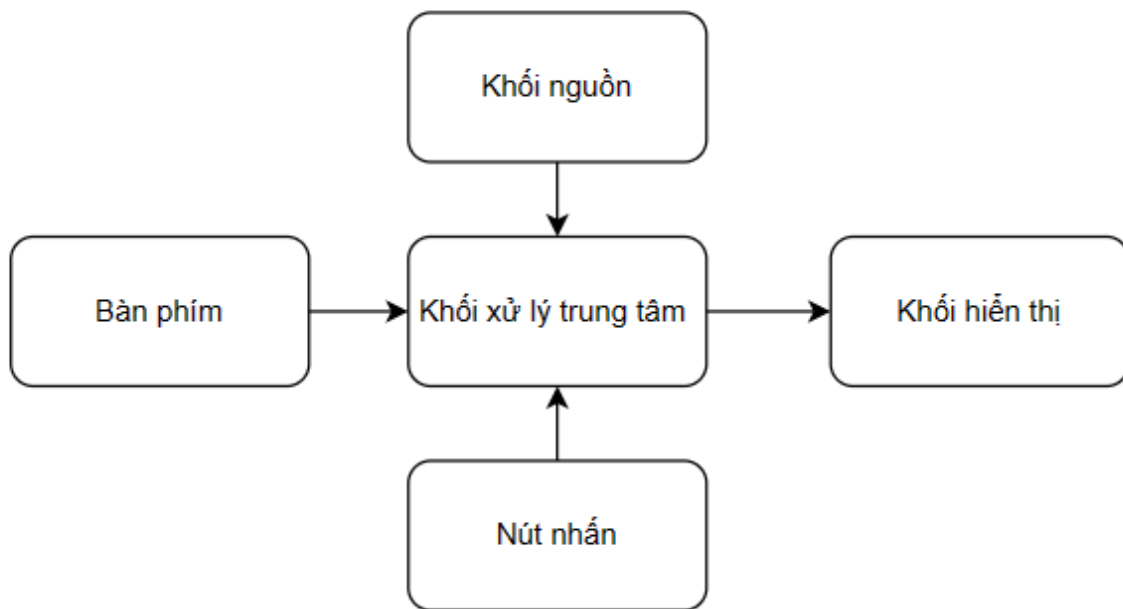
+ Đỏ 2 = Xanh + Vàng + 1

+ Xanh 2 = Đỏ - Vàng - 1

+ Vàng 2 = Vàng

- Ngoài ra khi vào giờ cao điểm lượng phương tiện giao thông nhiều, ta có thể điều chỉnh giá trị ban đầu của các đèn thông qua bàn phím. Vì 2 đèn hoạt động đồng bộ với nhau nên ta chỉ cần chỉnh thời gian các đèn ở cột 1.

1.2 Phân tích các khối chức năng của hệ thống



Hình 1 Các khối chức năng của hệ thống

1.2.1 Khối xử lý trung tâm

- Chức năng: Đóng vai trò đầu não của hệ thống, tiếp nhận tín hiệu từ bàn phím và nút nhấn. Từ đó đưa ra tín hiệu điều khiển đến khối mạch điều khiển.

1.2.2 Bàn phím

- Bàn phím gồm các nút. Chức năng là điều chỉnh tín hiệu ban đầu vào cho bộ xử lý trung tâm.

1.2.3 Nút nhấn

- Có 2 nút nhấn

+ 1 nút BUTTON 1 có tác dụng khởi động đèn ở chế độ mặc định

+ 1 nút BUTTON 2 có tác dụng thiết lập lại trạng thái ban đầu của hệ thống

1.2.4 Khối nguồn

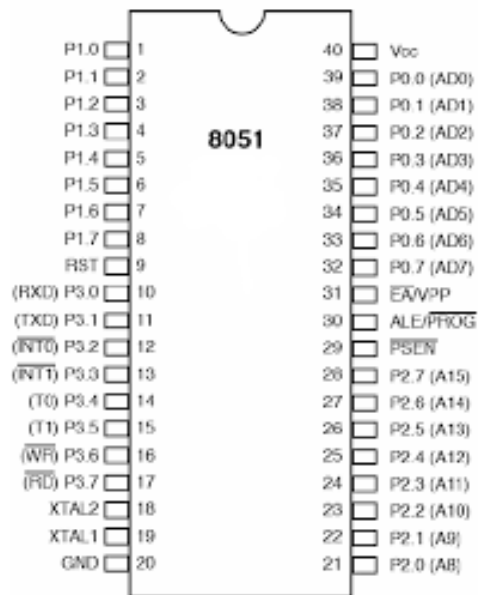
- Chức năng: cung cấp nguồn hoạt động cho hệ thống vi điều khiển.

1.2.5 Khối hiển thị

- Chức năng: hiển thị đèn giao thông.

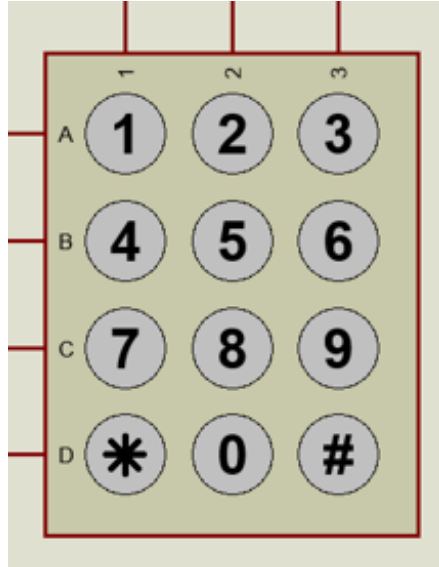
1.3 Giới thiệu về vi điều khiển 8051

1.3.1 Sơ lược về chân của IC 89C51



Hình 2 Sơ đồ chân của vi điều khiển 8051

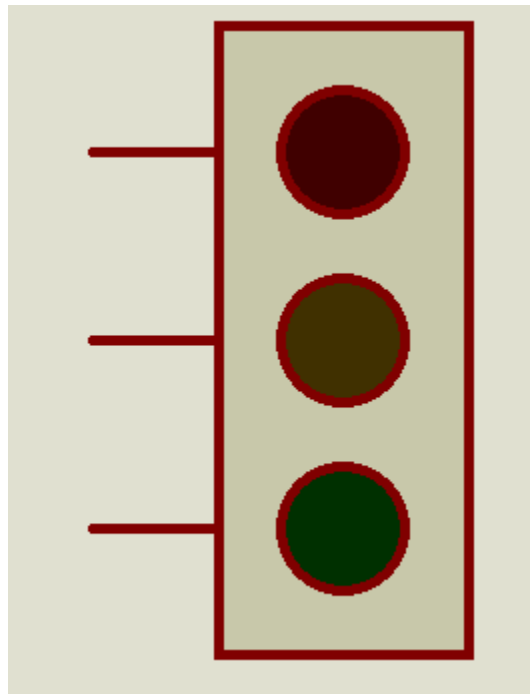
1.4 Giới thiệu về bàn phím



Hình 3 Hình ảnh bàn phím

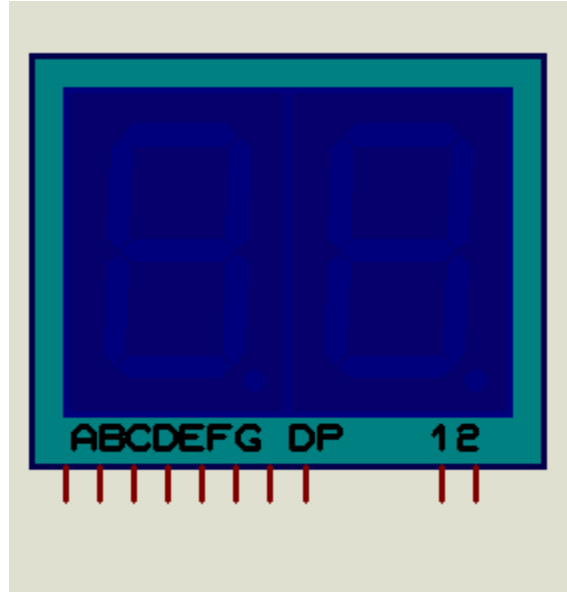
Bàn phím 4×3 có tổng cộng 12 nút ở dạng Ma trận. Một đầu nối nối 7 chân được cung cấp để kết nối nó với các mạch điều khiển microcontroller của bạn. Do bàn phím được mắc ở dạng ma trận nên phải dùng thuật toán quét phím để sử dụng.

1.5 Giới thiệu về đèn giao thông



Hình 4 Hình ảnh đèn giao thông

1.6 Giới thiệu led 7 đoạn loại dương chung

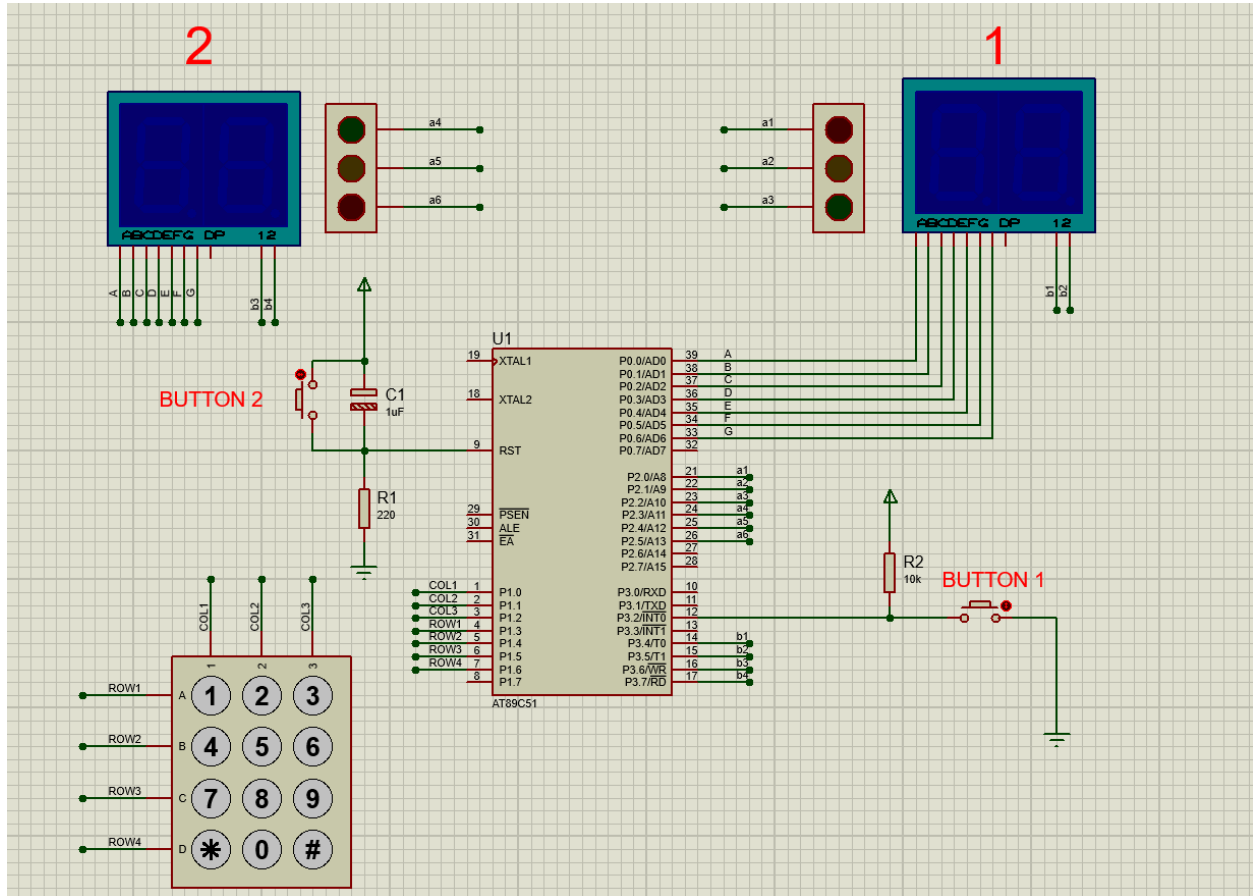


Hình 5 Hình ảnh led 7 đoạn 2 chữ số dương chung

- Cực dương của tất cả các LED được nối với nhau và các cực âm đứng riêng lẻ
- Gồm 11 chân
- + Chân A, B, C, D, E, F, G nối với LED hiển thị số 0, 1, 2,....
- + Chân DP (Decimal Point) nối với LED hiển thị dấu chấm thập phân
- + Chân 1, 2 để hiển thị chữ số đầu tiên, chữ số thứ hai

PHẦN 2: THIẾT KẾ VÀ THI CÔNG MẠCH

2.1 Thiết kế



Hình 6 Sơ đồ mạch thiết kế

2.1.1 Khối điều khiển trung tâm

-Sử dụng vi điều khiển 8051 làm nhiệm vụ điều khiển các khối chức năng khác.

2.1.2 Nút nhấn

- Có 2 nút nhấn

+ BUTTON1 có tác dụng khởi động đèn giao thông ở chế độ mặc định

+ BUTTON2 có tác dụng thiết lập lại trạng thái ban đầu của hệ thống

2.1.3 Bàn phím

Bàn phím là ma trận phím 4x3 gồm:

- 12 nút nhấn nối lại với nhau thành 4 hàng (ROW1 đến ROW4) và 3 cột (COL1 đến COL3)
- Sử dụng Port1 để kết nối ma trận phím với vi điều khiển. 3 bit thấp nối với 3 cột, 4 bit tiếp theo nối với các hàng.

2.1.4 Khối mạch điều khiển

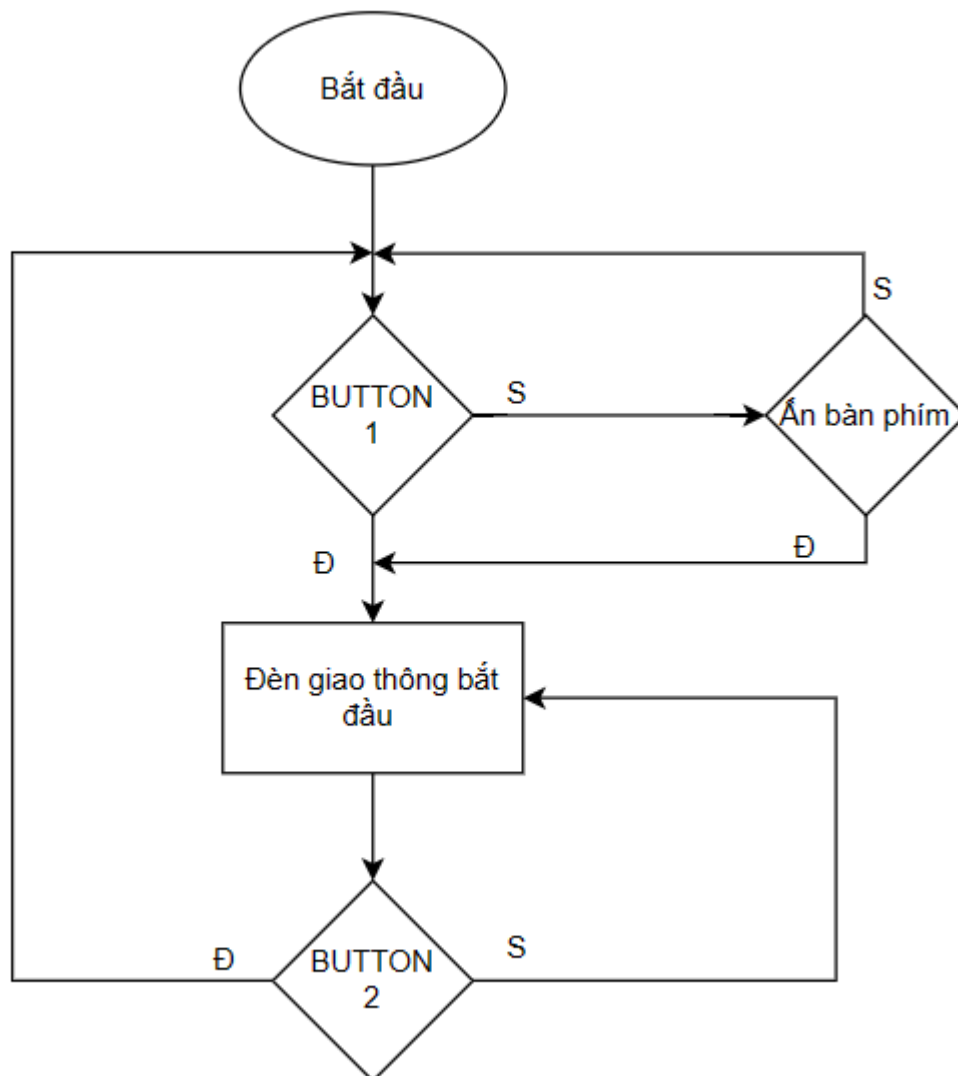
Gồm 2 LED 7 đoạn 2 chữ số loại dương chung có tác dụng hiển thị thời gian của đèn giao thông

2.2 Thi công mạch

Hệ thống được thực hiện bằng phần mềm mô phỏng Proteus

PHẦN 3: LƯU ĐỒ THUẬT TOÁN VÀ CÁC CHƯƠNG TRÌNH

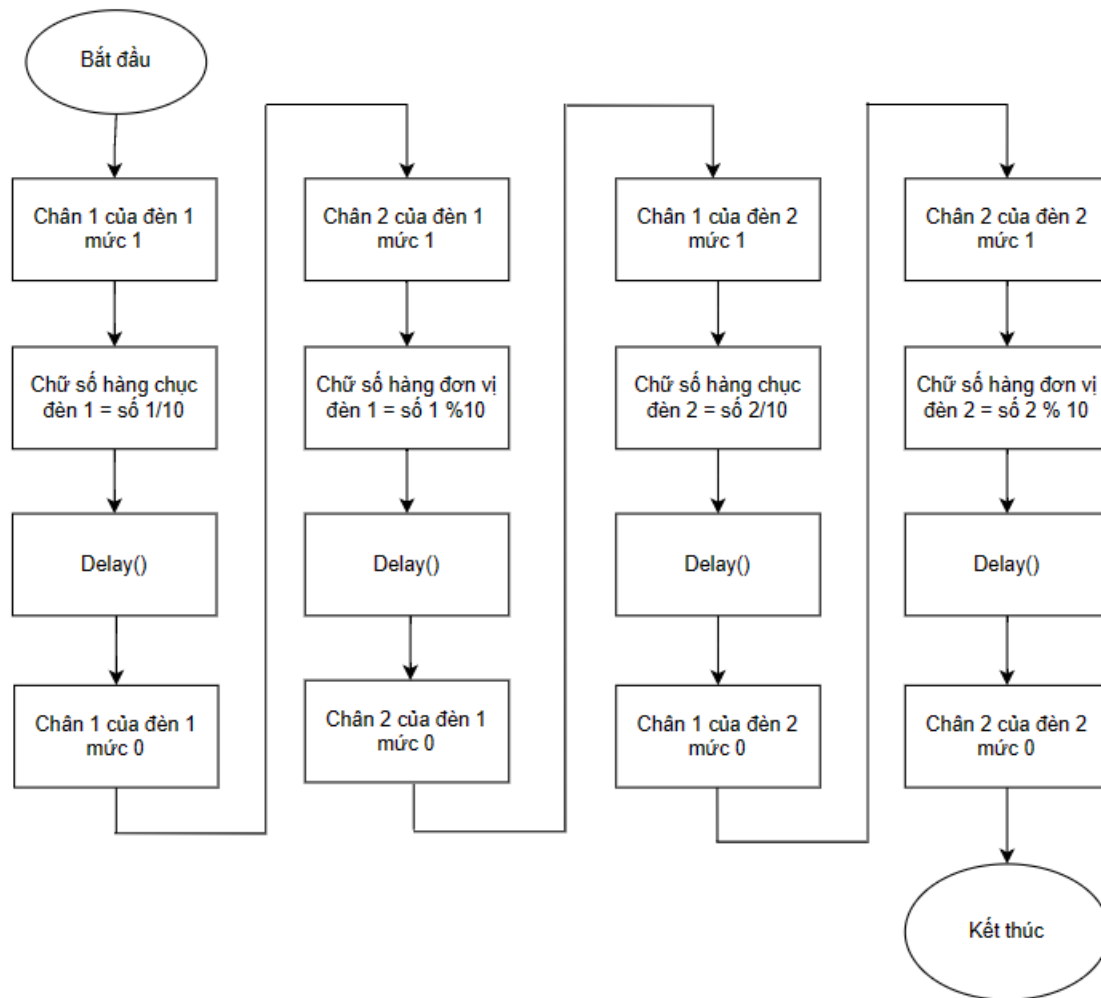
3.1 Lưu đồ thuật toán chương trình chính



Hình 7 Lưu đồ thuật toán chương trình chính

3.2 Lưu đồ thuật toán của các chương trình con

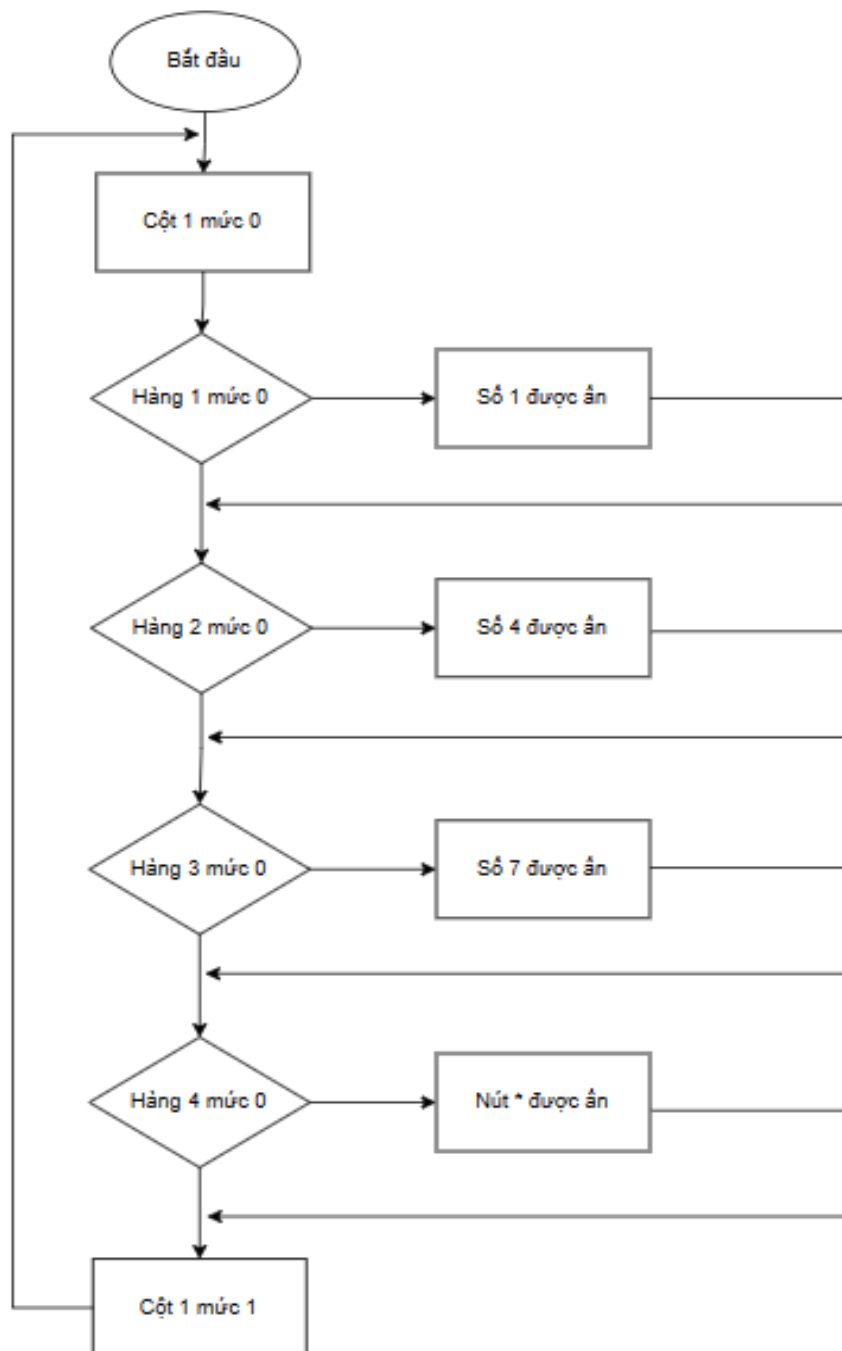
3.2.1 Thuật toán quét led



Hình 8 Lưu đồ thuật toán quét led

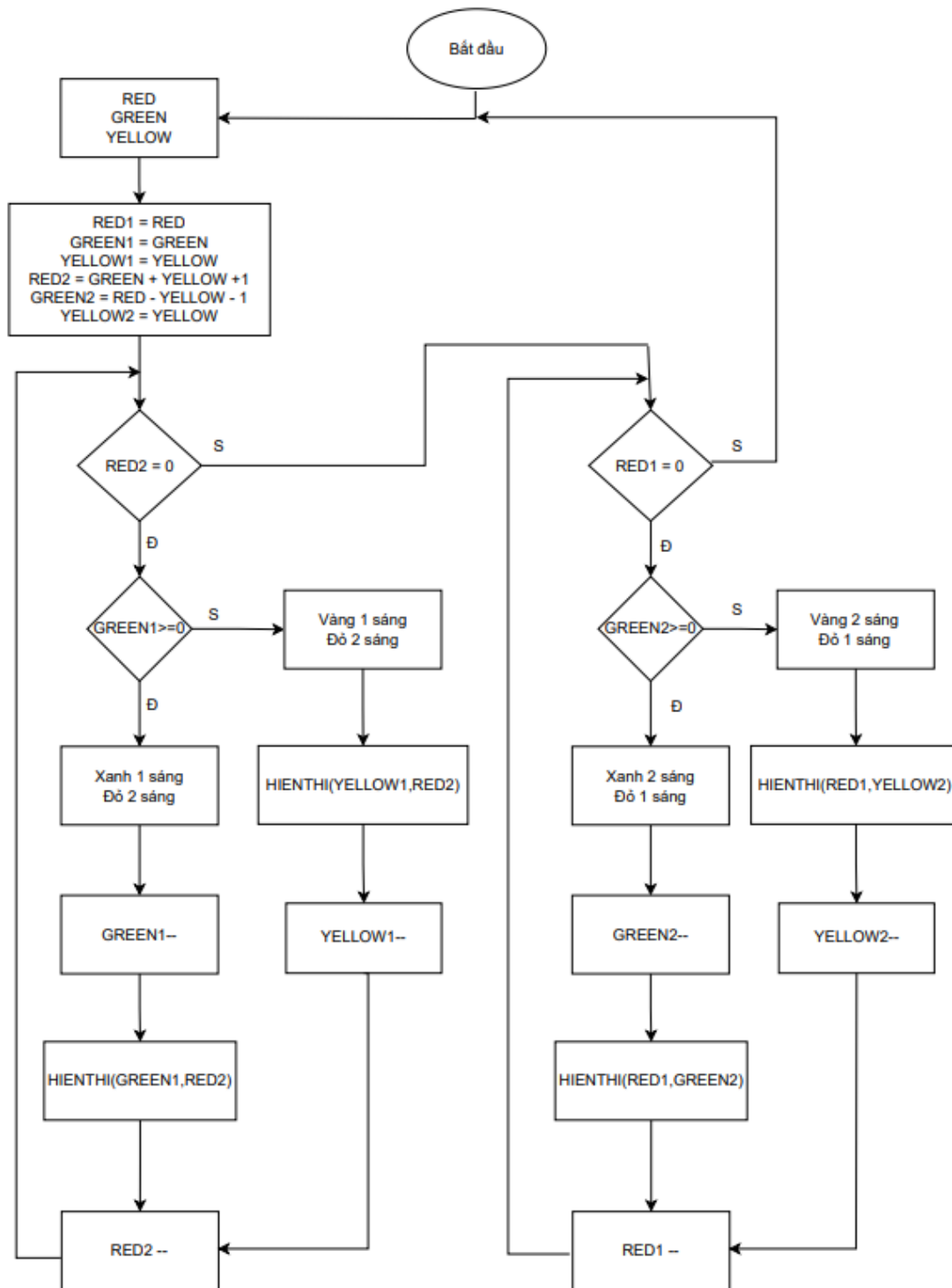
3.2.2 Thuật toán quét phím

Thuật toán này trình bày việc quét phím ở 1 cột, các cột còn lại thực hiện tương tự



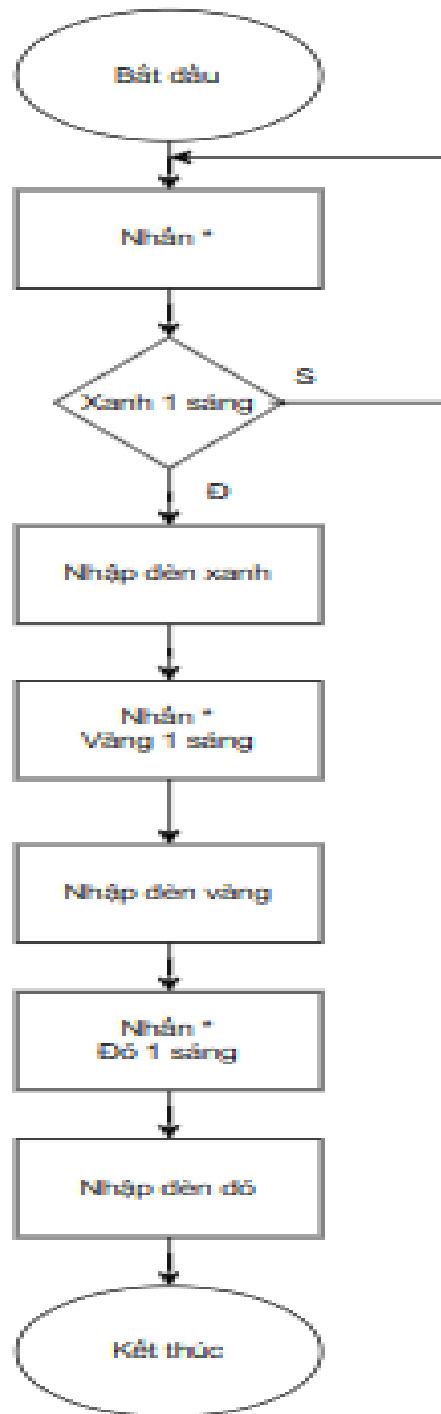
Hình 9 Lưu đồ thuật toán quét phím

3.2.3 Thuật toán hiển thị thời gian đèn giao thông



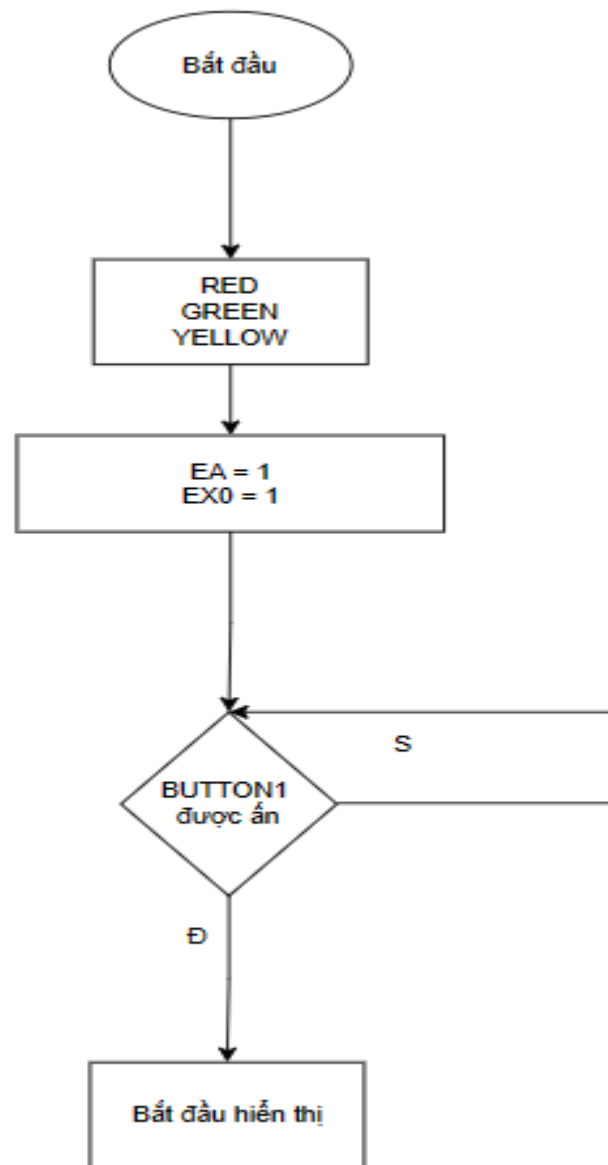
Hình 10 Lưu đồ thuật toán hiển thị đèn giao thông

3.2.4 Thuật toán điều chỉnh thời gian ban đầu bằng bàn phím



Hình 11 Lưu đồ thuật toán điều chỉnh thời gian bằng bàn phím

3.2.5 Thuật toán hiển thị thời gian mặc định



Hình 12 Lưu đồ thuật toán hiển thị thời gian mặc định

3.3 Chương trình

```
#include <REGX51.H>
unsigned char SO[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
sbit RED1 = P2 ^ 0;
sbit YELLOW1 = P2 ^ 1;
sbit GREEN1 = P2 ^ 2;
sbit RED2 = P2 ^ 5;
sbit YELLOW2 = P2 ^ 4;
sbit GREEN2 = P2 ^ 3;
sbit COL1 = P1 ^ 0;
sbit COL2 = P1 ^ 1;
sbit COL3 = P1 ^ 2;
sbit ROW1 = P1 ^ 3;
sbit ROW2 = P1 ^ 4;
sbit ROW3 = P1 ^ 5;
sbit ROW4 = P1 ^ 6;
int index = 0;
int count_red, count_green, count_yellow;
int count_red1, count_green1, count_yellow1;
int count_red2, count_green2, count_yellow2;
int i;
unsigned char key[9] = {-1, -1, -1, -1, -1, -1, -1, -1, -1}; //Chua gia tri nhap vao den giao thong
////////////////////////////////////
void delay(int time)
{
    while (time--);
}
void hienthi(int number, int number2)
{
    for (i = 0; i < 30; i++)
    {
        P3_4 = 1;
```

```

    P0 = SO[number / 10];
    delay(1000);
    P3_4 = 0;
    P3_5 = 1;
    P0 = SO[number % 10];
    delay(1000);
    P3_5 = 0;
    P3_6 = 1;
    P0 = SO[number2 / 10];
    delay(1000);
    P3_6 = 0;
    P3_7 = 1;
    P0 = SO[number2 % 10];
    delay(1000);
    P3_7 = 0;
}
}
void start()
{
    count_red1 = count_red;
    count_green1 = count_green;
    count_red2 = count_green + count_yellow + 1;
    count_green2 = count_red - count_yellow - 1;
    count_yellow1 = count_yellow2 = count_yellow;
    for (count_red2; count_red2 >= 0; count_red2--)
    {
        if (count_green1 >= 0)
        {
            RED1 = 0;
            YELLOW1 = 0;
            GREEN1 = 1;
            RED2 = 1;
            YELLOW2 = 0;
            GREEN2 = 0;

```

```

        hienthi(count_green1, count_red2);
        count_green1--;
    }
    else
    {

        RED1 = 0;
        YELLOW1 = 1;
        GREEN1 = 0;
        RED2 = 1;
        YELLOW2 = 0;
        GREEN2 = 0;
        hienthi(count_yellow1, count_red2);
        count_yellow1--;
    }
}
for (count_red1; count_red1 >= 0; count_red1--)
{
    if (count_green2 >= 0)
    {
        RED1 = 1;
        YELLOW1 = 0;
        GREEN1 = 0;
        RED2 = 0;
        YELLOW2 = 0;
        GREEN2 = 1;
        hienthi(count_red1, count_green2);
        count_green2--;
    }
    else
    {
        RED1 = 1;
        YELLOW1 = 0;
        GREEN1 = 0;

```

```

        RED2 = 0;
        YELLOW2 = 1;
        GREEN2 = 0;
        hienthi(count_red1, count_yellow2);
        count_yellow2--;
    }
}
}
void keyPress()
{
    COL1 = 0;
    if (ROW1 == 0)
    {
        key[index++] = 1;
        delay(50000);
    }
    if (ROW2 == 0)
    {
        key[index++] = 4;
        delay(50000);
    }
    if (ROW3 == 0)
    {
        key[index++] = 7;
        delay(50000);
    }
    if (ROW4 == 0)
    {
        key[index++] = -2;
        delay(50000);
    }
    COL1 = 1;
    COL2 = 0;
    if (ROW1 == 0)

```

```
{
    key[index++] = 2;
    delay(50000);
}
if (ROW2 == 0)
{
    key[index++] = 5;
    delay(50000);
}
if (ROW3 == 0)
{
    key[index++] = 8;
    delay(50000);
}
if (ROW4 == 0)
{
    key[index++] = 0;
    delay(50000);
}
COL2 = 1;
COL3 = 0;
if (ROW1 == 0)
{
    key[index++] = 3;
    delay(50000);
}
if (ROW2 == 0)
{
    key[index++] = 6;
    delay(50000);
}
if (ROW3 == 0)
{
    key[index++] = 9;
```

```

    delay(50000);
}
if (ROW4 == 0)
    delay(50000);
COL3 = 1;
}
void initTime()
{
    index = 0;
    while (index < 9)
    {
        keyPress();
        if(key[0]!=-2)
        {
            index = 0;
            continue;
        }
        if (index < 3 && key[0] == -2)
        {
            GREEN1 = 1;
            RED1 = 0;
            YELLOW1 = 0;
        }
        if (index >= 3 && index < 6 && key[3] == -2)
        {
            GREEN1 = 0;
            RED1 = 0;
            YELLOW1 = 1;
        }
        if (index >= 6 && index < 9 && key[6] == -2)
        {
            GREEN1 = 0;
            RED1 = 1;
            YELLOW1 = 0;
        }
    }
}

```

```

    }
}
}
////////////////////////////////////
///// Chương trình chính////////////////////////////////////
void main()
{
    P2 = 0x00;
    EA = 1;
    EX0 = 1; // Ngat ngoai 0
    initTime();
    count_green = key[1] * 10 + key[2];
    count_yellow = key[4] * 10 + key[5];
    count_red = key[7] * 10 + key[8];
    while (1)
        start();
}
////////////////////////////////////
//////////Ngat ngoai 0 //////////
void ngat() interrupt 0
{
    count_red = 10;
    count_green = 20;
    count_yellow = 3;
    while (1)
        start();
}

```

PHẦN 4: THỰC NGHIỆM

4.1 Môi trường thực nghiệm

Tiến hành điều chỉnh thời gian ban đầu bằng bàn phím hoặc nhấn nút để cho hiển thị ở thời gian mặc định

Dưới đây là testcase mẫu dùng cho việc điều chỉnh bằng bàn phím:

*50*03*20	*50*04*20	*50*05*10
*20*03*20	*20*04*20	*40*05*10
*10*03*20	*10*04*20	*30*05*10

4.2 Kết quả thu được

Đối với việc nhập bàn phím, chương trình mô phỏng được thử nghiệm qua các testcase khác nhau và đều cho kết quả đúng với thực tế.

PHẦN 5: ĐÁNH GIÁ

5.1 Giới hạn số lần điều chỉnh thời gian

Muốn điều chỉnh lại thời gian các lần tiếp theo cần phải ấn nút BUTTON 2 để thiết lập lại trạng thái ban đầu của hệ thống

5.2 Thời gian đáp ứng

Trên phần mềm mô phỏng thời gian đáp ứng nhanh, không có hiện tượng lỗi phần cứng.

5.3 Một số hạn chế còn tồn tại

Thời gian của đèn giao thông sẽ có nhiều sai số nếu áp dụng vào môi trường thực tế.

5.4 Kết luận

Vì đây là hệ thống được thực hiện trên phần mềm mô phỏng nên so với thực tế sẽ còn nhiều sự khác biệt.

TÀI LIỆU THAM KHẢO

- [1] Hồ Khánh Lâm, *Kỹ thuật Vi xử lý tập 1*, Nhà xuất bản Bưu điện
- [2] Tống Văn On, *Họ vi điều khiển 8051*, Nhà xuất bản Lao động – Xã hội