**Bài thí nghiệm số 5**

GVHD: Thầy Nguyễn Xuân Minh

Sinh viên: Cổ Chí Hào 1510888

Nguyễn Đình Hoàng Quý 41203052

1. Mục đích bài thí nghiệm

\_ Khảo sát cấu tạo, hoạt động của LED ma trận và LED 7 đoạn

\_ Tìm hiểu kỹ thuật quét LED ma trận và LED 7 đoạn

1. Thực hiện các yêu cầu:

\_ Nhóm em đã thực hiện các yêu cầu chính của buổi thí nghiệm gồm có các yêu cầu như sau:

* 1. Yêu cầu 1:

\_Xuất ra dữ liệu một hàng LED ma trận

\_Mức độ hoàn tất: Đã hoàn thành

Các bước thực hiện yêu cầu đã hoàn thành:

\_**Bước 1**: Kết nối mạch

\_**Bước 2**: Tạo dự án mới Tn05, tập tin nguồn Led.c

\_**Bước 3**: Định nghĩa, khởi động các port đã chọn và nút nhấn RA5

volatile unsigned char row \_\_at(0xF8A); //LATB

volatile unsigned char row\_io \_\_at(0xF93); //TRISB

volatile unsigned char gcol \_\_at(0xF8B); //LATC

volatile unsigned char gcol\_io \_\_at(0xF94); //TRISC

volatile unsigned char rcol \_\_at(0xF8C); //LATD

volatile unsigned char rcol\_io \_\_at(0xF95); //TRISD

#define XUAT 0

#define NHAP 1

#define NUTNHAN PORTAbits.RA5

#define NUTNHAN\_IO TRISAbits.RA5

void init()

{

ADCON1 = 0x0F; //Nut nhan RA5 nhap

NUTNHAN\_IO = NHAP; //Cac port led xuat

row\_io = XUAT;

gcol\_io = XUAT;

rcol\_io = XUAT;

}

\_**Bước 4**: Xuất dữ liệu cho ma trận LED sáng như yêu cầu của đề bằng cách định nghĩa G\_DATA và R\_DATA như sau:

#define G\_DATA 0x3C

#define R\_DATA 0xCC

void main(void)

{

init();

//row = 0;

gcol = G\_DATA;

rcol = R\_DATA;

row = 0b00000001;

}

\_**Bước 5**: Chạy chương trình và quan sát

\_**Bước 6**: Kiểm tra RA5 thực hiện xoay vòng chọn hàng từ 0 đến hàng 7

void main(void)

{

init();

//row = 0;

gcol = G\_DATA;

rcol = R\_DATA;

row = 0b00000001;

while(1)

{

if(NUTNHAN == 0)

{

row<<=1;

if(row==0) {row =1;}

while(NUTNHAN == 0);

}

}

}

* 1. Yêu cầu 1: Xuất hình ra LED ma trận

\_**Bước 8**: tạo ra module Led\_Matrix.c xuất khung hình. Đề xuất 1 khung hình có 8 hàng ra ma trận LED. Định nghĩa dữ liệu của 8 hàng được tổ chức theo bảng trong ROM thành 16byte liên tiếp ( 8 byte đầu là dữ liệu xanh, 8 byte sau là dữ liệu đỏ ).

#include<xc.h>

#pragma config OSC=HS, WDT = OFF, LVP = OFF

volatile unsigned char row \_\_at(0xF8A); //LATB

volatile unsigned char row\_io \_\_at(0xF93); //TRISB

volatile unsigned char gcol \_\_at(0xF8B); //LATC

volatile unsigned char gcol\_io \_\_at(0xF94); //TRISC

volatile unsigned char rcol \_\_at(0xF8C); //LATD

volatile unsigned char rcol\_io \_\_at(0xF95); //TRISD

#define XUAT 0

#define NHAP 1

#define NUTNHAN PORTAbits.RA5

#define NUTNHAN\_IO TRISAbits.RA5

#define COLMAX 8

#define ROWMAX 8

#define FRAME 2

#define RED 0

#define GREEN 1

void init()

{

row\_io = XUAT;

gcol\_io = XUAT;

rcol\_io = XUAT;

}

const unsigned char maurom[FRAME][ROWMAX]={

{0x01,0x02,0x3C,0x24,0x24,0x3C,0x40,0x80},

{0x80,0x40,0x3C,0x24,0x24,0x3C,0x02,0x01}

};

unsigned char rambuf[FRAME][ROWMAX], rowidx;

\_**Bước 9**: Thêm module Timer sử dụng Timer0 10ms. Trong timer\_process() thực hiện quy trình quét LED theo chỉ số rowidx

void timer\_process()

{

row = 0;

gcol=rambuf[GREEN][rowidx];

rcol=rambuf[RED][rowidx];

row=1<<rowidx++;

rowidx%=ROWMAX; }

\_**Bước 10**: Chạy chương trình chính:

void main()

{

init();

rom2ram();

timer0\_init();

while(1);

}

\_**Bước 11**: Để chỉnh lại thời gian quét để ma trận không bị chớp ta thay đổi

#define SODEM10MS -12500 thành -1250 trong module Timer.c

* 1. Yêu cầu 2: Hiển thị số ra LED 7 đoạn

\_Bước 12: Kết nối lại dây cáp

\_Bước 13: Tạo file mới Led7doan.c

\_Bước 14: Dữ liệu xuất ra 8 LED 7 đoạn gồm 8 byte (chứa số từ 0 đến 9)

\_Bước 15: Giải mã LED 7 đoạn:

* Sáng số 0: dữ liệu là 0b00111111 = 0x3F
* Sáng số 1: dữ liệu là 0b00000110 = 0x06
* Sáng số 2: dữ liệu là 0b01011011 = 0x5B
* Sáng số 3 : dữ liệu là 0b01001111= 0x4F
* Sáng số 4 : dữ liệu là 0b01100110= 0x66
* Sáng số 5 : dữ liệu là 0b01101101= 0x6D
* Sáng số 6 : dữ liệu là 0b01111101= 0x7D
* Sáng số 7 : dữ liệu là 0b00000111= 0x07
* Sáng số 8 : dữ liệu là 0b01111111= 0x7F
* Sáng số 9 : dữ liệu là 0b01101111= 0x6F
* Sáng chữ A : 0x77
* Sáng chứ B : 0x7C
* Sáng chữ C : 0x39
* Sáng chữ D : 0x5E
* Sáng chữ E : 0x79
* Sáng chữ F : 0x71

const unsigned char d17doan[MAX7S]={

0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,

0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71};

unsigned char dbuf[SELMAX],selidx;

unsigned char gm\_7doan(unsigned char so)

{

if (so>0x1F) return 0;

if (so<0x10) return d17[so];

return d17doan[so&0x0F] | 0x80;

}

void init()

{

TRISD = XUAT;

TRISB = XUAT;

idx=0;

}

\_Bước16 : Kết hợp timer0 10ms để hiện thị số trên cả 8 LED.

void timer\_process()

{

select = 0;

dcol = gm\_7doan(dbuf[selidx]);

select = 1<<selidx++;

selidx%=SELMAX;

}