**Bài thí nghiệm số 6**

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 ma trận phím

\_ Tìm hiểu kỹ thuật quét hàng / đọc cột, định vị phim nhấn

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:

\_ Lưu trong folder Part\_1

\_Kiểm tra phim trên từng hàng

\_Mức độ hoàn tất: Đã hoàn thành

Các bước thực hiện yêu cầu đã hoàn thành:

\_**Bước 1**: Tạo dự án mới tn06.c

\_**Bước 2**: Khởi động Port LED 8-bit, timer0, portB

#include<xc.h>

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

//#include"Timer.h"

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

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

\_**Bước 3**: Kết nối mạch

\_**Bước 4**: Kết nối hàng cột với PortB

* Cột COL0 – COL3: cổng nhập, nối với các bí RB0-3
* Hàng ROW0 – ROW3: cổng xuất, nối với RB4-7

\_**Bước 5**: Cấu hình PortB cho 4 bit cao là xuất, 4 bit thấp là nhập

#define XUAT 0

#define NHAP 1

#define X4N4 0x0F // 4 bit thap la nhap

#define ROW0 0b11101111

#define ROW1 0b11011111

#define ROW2 0b10111111

#define ROW3 0b01111111

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

volatile unsigned char keyin \_\_at(0xF81); //PORTB

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

void init()

{

ADCON1 = 0x0F; //Nhap digital

key\_io = X4N4; //B7-B4:hang xuat, B3-B0:cot nhap

led\_io = XUAT; //port LED xuat

led = 0;

}

\_**Bước 6**: Thực hiện kiểm tra phím trên 1 hàng. Làm cho hàng 3 xuống mức 0, các hàng khác lên mức 1. Sử dụng các dữ liệu ROW0 – ROW3 để xuất ra cổng chọn hàng (keyout).

void main(void)

{

init();

keyout = ROW3; // Xuat du lieu ra cong chon hang`

while(1) led = keyin;

}

\_**Bước 7:** Dịch, chạy, nhấn phím trên hàng 1 và quan sát port LED

\_**Bước 8:** Thực hiện tiếp kiểm tra trên các hàng còn lại.

* 1. Yêu cầu 2: Quét toàn bàn phím

\_ Lưu trong folder part\_2

\_ Tạo file Tn06\_timer.c để lưu mã nguồn.

\_**Bước 9**: Thêm vào module Timer0 tạo thời khoản 20ms. Trong timer\_process() gọi hàm quetphim() thực hiện kiểm tra phím theo các bước kế tiếp

#include<xc.h>

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

#include"Timer.h"

#define SODEM20MS 2

unsigned char dem20ms;

#define XUAT 0

#define NHAP 1

#define MAXIDX 4

#define X4N4 0x0F

#define ROW0 0b11101111

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

volatile unsigned char keyin \_\_at(0xF81); //PORTB

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

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

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

#define COL0 0b00001110

#define COL1 0b00001101

#define COL2 0b00001011

#define COL3 0b00000111

#define NOKEY 0xFF

void init()

{

ADCON1 = 0x0F;

key\_io = X4N4; //B7-B4:hang xuat, B3-B0:cot nhap

led\_io = XUAT; //port LED xuat

led = 0;

dem20ms = SODEM20MS;

}

\_**Bước 10**: Sử dụng một biến idx (giá trị = 0 ,1,2,3) để ghi nhớ vị trí hàng đang đc kiểm tra:

unsigned char keyreg;

unsigned char idx;

\_**Bước 11**: Tạo hàm quetphim()

void getkey()

{

unsigned char col;

col = keyin & 0x0F; // RB3-RB0

switch (col)

{

case COL0: keyreg =0; break;

case COL1: keyreg =1; break;

case COL2: keyreg =2; break;

case COL3: keyreg =3; break;

default: keyreg = NOKEY;

}

// doi ra so thu tu (keycode)

if(keyreg!=NOKEY) keyreg += (idx\*4);

}

void quetphim()

{

unsigned char dl\_quet;

keyout = 0xFF; // chon hang theo idx

dl\_quet = ROW0<<idx;

keyout = dl\_quet;

getkey(); // doc cot, phat hien phim

idx = (++idx)%MAXIDX;

if(keyreg!=NOKEY) xulyphim();

}

\_**Bước 12:** Tạo hàm xử lý phím xulyphim()

void xulyphim()

{

led = keyreg;

}

\_ Chương trình chỉnh của phần này được thực hiện như sau:

void main()

{

init();

timer0\_init();

while(1);

}

* 1. Bài tập:

1. Hiện mã phím hoặc một câu thông báo ra màn hình LCD khi nhấn phím

Tạo module tn06\_LCD để lưu mã nguồn và được lưu trong folder part\_3:

#include<xc.h>

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

#include<stdio.h>

#include"Timer.h"

#include"tn06\_LCD.h"

#define XUAT 0

#define NHAP 1

#define MAXIDX 4

#define X4N4 0x0F

#define ROW0 0b11101111

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

volatile unsigned char keyin \_\_at(0xF81); //PORTB

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

#define COL0 0b00001110

#define COL1 0b00001101

#define COL2 0b00001011

#define COL3 0b00000111

#define NOKEY 0xFF

unsigned char idx;

unsigned char keyreg;

/\*-----------------------------------------------------------\*/

void getkey()

{

unsigned char col;

col = keyin & 0x0F; // RB3-RB0

switch (col)

{

case COL0: keyreg =0; break;

case COL1: keyreg =1; break;

case COL2: keyreg =2; break;

case COL3: keyreg =3; break;

default: keyreg = NOKEY;

}

// doi ra so thu tu (keycode)

if(keyreg!=NOKEY) keyreg += (idx\*4);

}

void xulyphim()

{

printf("\f%d!=\n%e",keyreg);

}

void quetphim()

{

unsigned char dl\_quet;

keyout = 0xFF; // chon hang theo idx

dl\_quet = ROW0<<idx;

keyout = dl\_quet;

getkey(); // doc cot, phat hien phim

idx = (++idx)%MAXIDX;

if(keyreg!=NOKEY) xulyphim();

}

/\*-----------------------------------------------------------\*/

void init()

{

ADCON1 = 0x0F;

key\_io = X4N4; //B7-B4:hang xuat, B3-B0:cot nhap

}

void main()

{

init();

lcd\_init();

while(1){

quetphim();

}

}

1. Xử lý chống rung bằng phần mềm.