**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**----oOo----**

**Text, logo

Description automatically generated**

**VI ĐIỀU KHIỂN VÀ ỨNG DỤNG**

**BÀI BÁO CÁO 12**

**NHÓM 1**

Giảng viên: **Phạm Quang Trí**

Sinh viên:

Nguyễn Ngọc Huy\_21115981

Nguyễn Thị Như Huyền\_21126481

Hồ Bửu Sơn\_21119571

**TP.HCM 2023**

**Yêu cầu 1**: **Trong quyển báo cáo, dựa vào yêu cầu bài tập đã chọn, phải vẽ sơ đồ kết nối phần cứng của toàn bộ hệ thống sử dụng vi điều khiển STM32.**

* Sơ đồ kết nối phần cứng

A computer circuit diagram with text and numbers

Description automatically generated with medium confidence

**Yêu cầu 2: Trong quyển báo cáo, trình bày rõ ràng lưu đồ giải thuật và mã nguồn của chương trình điều khiển**

* Lưu đồ giải thuật

lcd\_send\_cmd

lcd\_write\_nibble

Gửi 4 bit của lệnh (4 bit cao)

Chọn chế độ lệnh hoặc dữ liệu cho LCD

Gửi 4 bit của lệnh (4 bit thấp)

Cho phép đọc thông tin từ LCD

Delay 2ms

Cho phép LCD trao đổi thông tin

Xuất 4 bit dữ liệu ra các chân kết nối D4-D7 của LCD

lcd\_send\_data

Gửi 4 bit của lệnh (4 bit cao)

Delay 1us

Gửi 4 bit của lệnh (4 bit thấp)

Không cho phép LCD trao đổi thông tin

Delay 2ms

Delay 1us

lcd\_display

Tăng con trỏ dữ liệu

lcd\_init

Delay 20ms

Đúng

Chưa hết chuỗi dữ liệu

Gửi dữ liệu hiển thị

Đặt chế độ LCD 8 bit dữ liệu

Sai

Delay 5ms

Đặt chế độ LCD 8 bit dữ liệu

Delay 1ms

lcd\_gotoxy

Đặt chế độ LCD 4 bit   
Hiển thị 2 dòng thông tin  
Kiểu ký tự 5x8

Mã tọa độ bằng 0

Biểu thị thông tin   
Tắt hiển thị con trỏ

Đúng

Hàng trên ?   
(row=0)

Mã tọa độ = 0x80+col

Xóa màn hình LCD

Sai

Hàng dưới?   
(row=1)

Đúng

Mã tọa độ = 0xC0+col

Chế độ tăng con trỏ tự động và không dịch chuyển màn hình hiển thị

Sai

Gửi lệnh điều khiển (mã tọa độ)

Gửi thông báo "OK" qua UART khi đã nhập đúng

uart\_send\_ok

Gửi thông báo lỗi qua UART khi kích thước đầu vào không đúng

missing\_char

sai\_ki\_tu\_dau

sai\_ki\_tu\_cuoi

Gửi thông báo lỗi qua UART khi kí tự cuối bị nhập sai

Gửi thông báo lỗi qua UART khi kí tự đầu bị nhập sai

USART1\_IRQHandler

Gán giá trị được nhận từ UART và lưu vào mảng 'get\_Frame' tại vị trí 'cnt'

Đọc từng byte dữ liệu từ cổng UART

Gán cờ ngắt 'flag' = 1

main

Khởi động LCD

Gửi dữ liệu thời gian và ngày, tháng, năm ban đầu (SMH và DMY) đến mô-đun RTC thông qua giao thức I2C.

Gửi hướng dẫn nhập lệnh qua cổng UART để hướng dẫn người dùng nhập

Gửi lệnh cấu hình cổng UART để nhận dữ liệu theo cách ngắt

C

Gửi một byte dữ liệu từ vi điều khiển đến thiết bị I2C với địa chỉ là DevAddress. Dữ liệu được lấy từ mảng 'pData'

Nhận dữ liệu từ thiết bị I2C với địa chỉ là DevAddress. Dữ liệu được lưu vào mảng 'data'

Định dạng dữ liệu thời gian và ngày, tháng, năm để hiển thị lên màn hình LCD.

Xóa màn hình LCD

Định vị tọa độ "0,4"

A

A

Hiển thị thời gian lên LCD

Định vị tọa độ "1,3"

Hiển thị ngày, tháng, năm lên LCD

Kiểm tra flag xem có lệnh được nhập qua UART?

Sai

C

Đúng

Kiểm tra xem kí tự cuối có bằng '\*'?

Sai

Đặt lại biến 'flag' và 'cnt' để sẵn sàng nhận lệnh mới

Thông báo lỗi về ký tự cuối không đúng

Đúng

Kiểm tra xem kí tự đầu có bằng '\*'?

Đặt lại biến 'flag' và 'cnt' để sẵn sàng nhận lệnh mới

Sai

Thông báo lỗi về ký tự đầu không đúng

Đúng

Kiểm tra xem lệnh được nhập có đủ kí tự?

Sai

Đặt lại biến 'flag' và 'cnt' để sẵn sàng nhận lệnh mới

C

Thông báo đã nhập sai kích thước

Đúng

B

B

Gửi 'OK' qua UART

Gán giá trị từ lệnh nhập vào mảng 'SMH\_Update' và 'DMY\_Update'.

Gửi dữ liệu mới nhập là thời gian và ngày, tháng, năm (SMH và DMY) đến mô-đun RTC thông qua giao thức I2C.

Đặt lại biến 'flag' và 'cnt' để sẵn sàng nhận lệnh mới

Delay 1s

C

* Mã nguồn chương trình

*- Chương trình ngắt*

void USART1\_IRQHandler(void)

{

HAL\_UART\_IRQHandler(&huart1);

get\_Frame[cnt] = frame;

HAL\_UART\_Receive\_IT(&huart1, &frame, 1);

cnt++;

flag = 1;

}

*- Chương trình chính*

#include "main.h"

#include "stdio.h"

#define NUM 8

I2C\_HandleTypeDef hi2c1;

UART\_HandleTypeDef huart1;

uint8\_t DevAddress = 0xD0;

uint8\_t pData[2] = {0x00, 0x00};

uint8\_t data[7];

uint8\_t SMH[4] = {0x00,0x21,0x11,0x23}; //giay,phut,gio

uint8\_t DMY[4] = {0x04,0x21,0x11,0x95}; //day,month,year

uint8\_t uart\_re[100];

uint8\_t frame = 0;

uint8\_t get\_Frame[100] = {0};//#second:min:hour-day,month,year\*

uint8\_t cnt;

int flag = 0;

uint8\_t SMH\_Update[4] = {0x00}; //giay,phut,gio

uint8\_t DMY\_Update[4] = {0x04}; //day,month,year

uint8\_t pData\_Update[2] = {0x00, 0x00};

uint8\_t data\_Update[7];

uint8\_t lcd\_String\_Top[16];

uint8\_t lcd\_String\_Bot[16];

void SystemClock\_Config(void);

static void MX\_GPIO\_Init(void);

static void MX\_I2C1\_Init(void);

static void MX\_USART1\_UART\_Init(void);

void missing\_char()

{

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)"Ban da nhap sai kich thuoc. Moi nhap lai\r\n", 44, 10);

}

void uart\_send\_ok()

{

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)"OK\r\n", 6, 10);

}

void sai\_ki\_tu\_cuoi()

{

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)"Ban da nhap sai ki tu cuoi. Moi nhap lai\r\n", 44, 10); //sai \*

}

void sai\_ki\_tu\_dau()

{

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)"Ban da nhap sai ki tu dau. Moi nhap lai\r\n", 43, 10); //sai \*

}

void lcd\_write\_nibble(uint8\_t rs, uint8\_t data){

HAL\_GPIO\_WritePin(RS\_GPIO\_Port, RS\_Pin,rs);

HAL\_GPIO\_WritePin(RW\_GPIO\_Port, RW\_Pin,0);

HAL\_GPIO\_WritePin(EN\_GPIO\_Port, EN\_Pin,1);

HAL\_GPIO\_WritePin(D4\_GPIO\_Port, D4\_Pin, (data>>0) & 0x01);

HAL\_GPIO\_WritePin(D5\_GPIO\_Port, D5\_Pin, (data>>1) & 0x01);

HAL\_GPIO\_WritePin(D6\_GPIO\_Port, D6\_Pin, (data>>2) & 0x01);

HAL\_GPIO\_WritePin(D7\_GPIO\_Port, D7\_Pin, (data>>3) & 0x01);

for(uint8\_t i=0; i<72; i++) asm("NOP");

HAL\_GPIO\_WritePin(EN\_GPIO\_Port, EN\_Pin,0);

for(uint8\_t i=0; i<72; i++) asm("NOP");

}

void lcd\_send\_cmd(uint8\_t cmd){

lcd\_write\_nibble(0, (cmd>>4)&0x0F);

lcd\_write\_nibble(0, cmd&0x0F);

HAL\_Delay(2);

}

void lcd\_send\_data(uint8\_t data){

lcd\_write\_nibble(1, (data>>4)&0x0F);

lcd\_write\_nibble(1, data&0x0F);

HAL\_Delay(2);

}

void lcd\_init(void){

HAL\_Delay(20);

lcd\_write\_nibble(0, 0x03); HAL\_Delay(5);

lcd\_write\_nibble(0, 0x03); HAL\_Delay(1);

lcd\_write\_nibble(0, 0x03); HAL\_Delay(1);

lcd\_write\_nibble(0, 0x02); HAL\_Delay(1);

lcd\_send\_cmd(0x28);

lcd\_send\_cmd(0x0C);

lcd\_send\_cmd(0x01);

lcd\_send\_cmd(0x06);

}

void lcd\_display(uint8\_t \*data, uint8\_t len)

{

for (uint8\_t j = 0; j < len; j++)

{

lcd\_send\_data(data[j]);

}

}

void lcd\_gotoxy(uint8\_t row, uint8\_t col){

uint8\_t coordinates = 0;

switch(row){

case 0:

coordinates = 0x80 | col;

break;

case 1:

coordinates = 0xC0 | col;

break;

}

lcd\_send\_cmd(coordinates);

}

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_I2C1\_Init();

MX\_USART1\_UART\_Init();

lcd\_init();

HAL\_I2C\_Master\_Transmit(&hi2c1, DevAddress, &SMH[0], 4, 10);

HAL\_I2C\_Master\_Transmit(&hi2c1, DevAddress, &DMY[0], 4, 10);

HAL\_UART\_Transmit(&huart1, (uint8\_t \*)"\t Hay Nhap theo mau duoi day \r\n \t\* second min hour day month year \*\n trong do \* nhap la 2A\n", 97,HAL\_MAX\_DELAY);

HAL\_UART\_Receive\_IT(&huart1, &frame, 1);

while (1)

{

HAL\_I2C\_Master\_Transmit(&hi2c1, DevAddress, &pData[0], 1, 10);

HAL\_I2C\_Master\_Receive(&hi2c1, DevAddress, &data[0], 7, 10);

sprintf((char\*)lcd\_String\_Top,"%x:%x:%x",data[0],data[1],data[2]);

sprintf((char\*)lcd\_String\_Bot,"%x/%x/20%x",data[4],data[5],data[6]);

lcd\_send\_cmd(0x01);

lcd\_gotoxy(0, 4);

lcd\_display(lcd\_String\_Top, 8);

lcd\_gotoxy(1, 3);

lcd\_display(lcd\_String\_Bot, 10);

if(flag == 1){

if (frame == 0x2a){

if(get\_Frame[0] == 0x2a){

if(cnt == NUM){

uart\_send\_ok();

for (int j = 1; j < 4;j++){

SMH\_Update[j] = get\_Frame[j];

}

for(int k = 1; k < 4; k++){

DMY\_Update[k] = get\_Frame[k +3];

}

HAL\_I2C\_Master\_Transmit(&hi2c1, DevAddress, &SMH\_Update[0], 4, 10);

HAL\_I2C\_Master\_Transmit(&hi2c1, DevAddress, &DMY\_Update[0], 4, 10);

flag = 0;

cnt = 0;

}

else{

flag = 0;

cnt = 0;

missing\_char();

}

}

else {

flag = 0;

cnt = 0;

sai\_ki\_tu\_dau();

}

}

else {

flag = 0;

cnt = 0;

sai\_ki\_tu\_cuoi();

}

}

HAL\_Delay(1000);

}

}

**Yêu cầu 3: Nạp chương trình vào kit thí nghiệm STM32 và thực hiện cho chạy thử trên phần cứng trong phòng thực hành. Quay video clip minh chứng kết quả thực hiện, tải lên Youtube duy nhất 1 video clip ( trong trường hợp minh chứng có nhiều video clip nhỏ thì sinh viên phải tự ghép lại thành 1 video clip tổng hợp ) và ghi liên kết vào báo cáo.**

Liên kết video: https://youtu.be/tV0wiy-TIYw