**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM**



**VI ĐIỀU KHIỂN**

**BÀI BÁO CÁO 5**

**NHÓM 4**

Giảng viên : **PHẠM QUANG TRÍ**

Sinh viên :

* Trần Công Hòa 20017691
* Đỗ Tuấn Duy 20061261
* Nguyễn Quốc Dương 20123251

**TP.HCM – 2022**

**Bài tập mức độ 3:**

1. **Sơ đồ nguyên lí kết nối phần cứng tối thiểu dùng STM32:**

Diagram, schematic

Description automatically generated

1. **Cách thiết lập chức năng trong phần mềm CubeMX:**

**Bước 1**: Khởi động phần mềm CubeMX, tạo New Project và chọn mã số STM32F103C8T6.

Graphical user interface, text, application

Description automatically generated

**Bước 2**: Chọn mạch nạp:

Graphical user interface, application

Description automatically generated

**Bước 3**: Chọn bộ giao động:

**Graphical user interface

Description automatically generated**

**Bước 4:** Chọn tần số giao động:

Graphical user interface

Description automatically generated with medium confidence

**Bước 5** : Cấu hình Uart1 chọn baundrate 57600, parity none và chọn ngắt

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

**Graphical user interface

Description automatically generated**

**Bước 6**: Cấu hình chân GPIO Output cho LCD

Graphical user interface, application

Description automatically generated

**Bước 7**: Đặt tên cho Project và chọn nơi lưu trữ, chọn phần mềm viết chương trình

Graphical user interface, text, application, email

Description automatically generated

1. **Lưu đồ giải thuật:**

**Chương trình ngắt:**

**Diagram

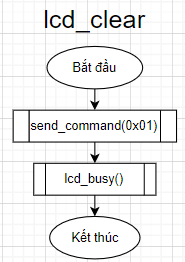
Description automatically generated**

**Chương trình con:**

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Chương trình chính:**

**A picture containing text, indoor, map

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedChart, diagram

Description automatically generatedDiagram

Description automatically generated**

1. **Mã nguồn chương trình:**

**Chương trình ngắt:**

void USART1\_IRQHandler(void)

{

/\* USER CODE BEGIN USART1\_IRQn 0 \*/

/\* USER CODE END USART1\_IRQn 0 \*/

HAL\_UART\_IRQHandler(&huart1);

/\* USER CODE BEGIN USART1\_IRQn 1 \*/

HAL\_UART\_Receive\_IT(&huart1, &data\_receive[0], 1);

data\_copy[ptu] = data\_receive[0];

ptu++;

flag=1;

/\* USER CODE END USART1\_IRQn 1 \*/

}

**Chương trình con:**

void write\_to\_lcd\_4bits(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(D7\_GPIO\_Port, D7\_Pin, ((data>>3)&0x01));

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

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

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

for(i = 0; i < 72; i++)

{

\_\_asm("NOP");

}

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

lcd\_busy();

}

void send\_command(uint8\_t data)

{

write\_to\_lcd\_4bits(0, (data>>4)&0x0f);

write\_to\_lcd\_4bits (0, (data>>0)&0x0f);

}

void display\_lcd(uint8\_t data)

{

write\_to\_lcd\_4bits(1, (data>>4)&0x0f);

write\_to\_lcd\_4bits (1, (data>>0)&0x0f);

}

void display\_text(char \*data)

{

uint8\_t j;

for(j=0; j<strlen(data); j++)

{

display\_lcd(data[j]);

}

}

void lcd\_clear()

{

send\_command(0x01);

lcd\_busy();

}

void nap\_tudien(void)

{

char i = 0;

send\_command(0x40); // che do nap tu dien moi

while(Font[i] != 0x99)

{

display\_lcd(Font[i]);

i++;

}

}

void nap\_tudien2(void)

{

char i = 0;

send\_command(0x40); // che do nap tu dien moi

while(Font2[i] != 0x99)

{

display\_lcd(Font2[i]);

i++;

}

}

void set\_cursor(uint8\_t row, uint8\_t col)

{

uint8\_t temp;

switch(row)

{

case 0:

temp = col|0x80;

break;

case 1:

temp = col|0xC0;

break;

default:

temp = col|0x80;

}

send\_command(temp);

}

void lcd\_config()

{

send\_command(0x28); //LCD 4bit, 2-line, 5x8 hieu 2

lcd\_busy();

send\_command(0x28); //LCD 4bit, 2-line, 5x8 hieu 8

lcd\_busy();

send\_command(0x0f);

lcd\_busy();

send\_command(0x06);

lcd\_busy();

send\_command(0x01);

lcd\_busy();

}

void lcd\_busy(void)

{

uint8\_t busy\_flag\_copy;

GPIO\_InitTypeDef GPIO\_InitStruct = {0};

GPIO\_InitStruct.Pin = GPIO\_PIN\_5;

GPIO\_InitStruct.Mode = GPIO\_MODE\_INPUT;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

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

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

do{

busy\_flag\_copy = 0;

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

for(int i = 0; i < 72; i++);

busy\_flag\_copy = HAL\_GPIO\_ReadPin(GPIOA, GPIO\_PIN\_5);

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

for(int i = 0; i < 72; i++);

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

for(int i = 0; i < 72; i++);

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

for(int i = 0; i < 72; i++);

}while(busy\_flag\_copy);

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

GPIO\_InitStruct.Pin = GPIO\_PIN\_5;

GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;

GPIO\_InitStruct.Pull = GPIO\_NOPULL;

GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;

HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

}

**Chương trình chính:**

#include "main.h"

/\* Private includes ----------------------------------------------------------\*/

/\* USER CODE BEGIN Includes \*/

#include "string.h"

/\* USER CODE END Includes \*/

/\* Private typedef -----------------------------------------------------------\*/

/\* USER CODE BEGIN PTD \*/

/\* USER CODE END PTD \*/

/\* Private define ------------------------------------------------------------\*/

/\* USER CODE BEGIN PD \*/

/\* USER CODE END PD \*/

/\* Private macro -------------------------------------------------------------\*/

/\* USER CODE BEGIN PM \*/

/\* USER CODE END PM \*/

/\* Private variables ---------------------------------------------------------\*/

UART\_HandleTypeDef huart1;

/\* USER CODE BEGIN PV \*/

uint8\_t OK[2] = {"OK"};

uint8\_t Er0[8] = {"ERROR 41"};

uint8\_t Er1[8] = {"ERROR 42"};

uint8\_t Er2[8] = {"ERROR 43"};

uint8\_t Er3[8] = {"ERROR 44"};

uint8\_t Er4[8] = {"ERROR 45"};

uint8\_t Er5[8] = {"ERROR 46"};

uint8\_t ptu = 0;

uint8\_t data\_receive[1];

uint8\_t data\_copy[16];

uint8\_t flag = 0;

uint8\_t check\_xor;

void lcd\_busy(void);

/\* USER CODE END PV \*/

…

int main(void)

{

/\* USER CODE BEGIN 1 \*/

/\* USER CODE END 1 \*/

/\* MCU Configuration--------------------------------------------------------\*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/

HAL\_Init();

/\* USER CODE BEGIN Init \*/

/\* USER CODE END Init \*/

/\* Configure the system clock \*/

SystemClock\_Config();

/\* USER CODE BEGIN SysInit \*/

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/

MX\_GPIO\_Init();

MX\_USART1\_UART\_Init();

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Receive\_IT(&huart1, &data\_receive[0], 1);

lcd\_config();

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

while (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

if(flag == 1)

{

if(data\_copy[1] <= 0x00 || data\_copy[1] > (0x10))

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &Er2[0], 8, 10);

}

else if(ptu != (data\_copy[1] + 5))

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &Er4[0], 8, 10);

}

else if(data\_copy[0]=='\*')

{

if (data\_copy[(data\_copy[1]+4)]=='#')

{

check\_xor = data\_copy[1];

for(int i = 2; i <= (data\_copy[1]+2); i++)

{

check\_xor = check\_xor^data\_copy[i];

}

if(data\_copy[2] == 0 || data\_copy[2] == 1)

{

if(check\_xor == data\_copy[6])

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &OK[0], 2, 10);

lcd\_clear();

set\_cursor (data\_copy[2], 0);

for (int i=3; i<(data\_copy[1]+3); i++)

{

display\_lcd(data\_copy[i]);

}

}

else

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &Er5[0], 8, 10);

}

}

else

{

HAL\_UART\_Transmit(&huart1, &Er3[0], 8, 10);

ptu=0;

}

}

else

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &Er1[0], 8, 10);

}

}

else

{

ptu=0;

HAL\_UART\_Transmit(&huart1, &Er0[0], 8, 10);

}

}

flag=0;

HAL\_Delay(50);

}

/\* USER CODE END 3 \*/

}

1. **Video minh chứng:**

Link youtube: <https://youtu.be/3pR3W4mQ0Hc>