**BỘ CÔNG THƯƠNG**

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



**VI ĐIỀU KHIỂN**

**BÀI BÁO CÁO 4**

**NHÓM 3**

Giảng viên : **PHẠM QUANG TRÍ**

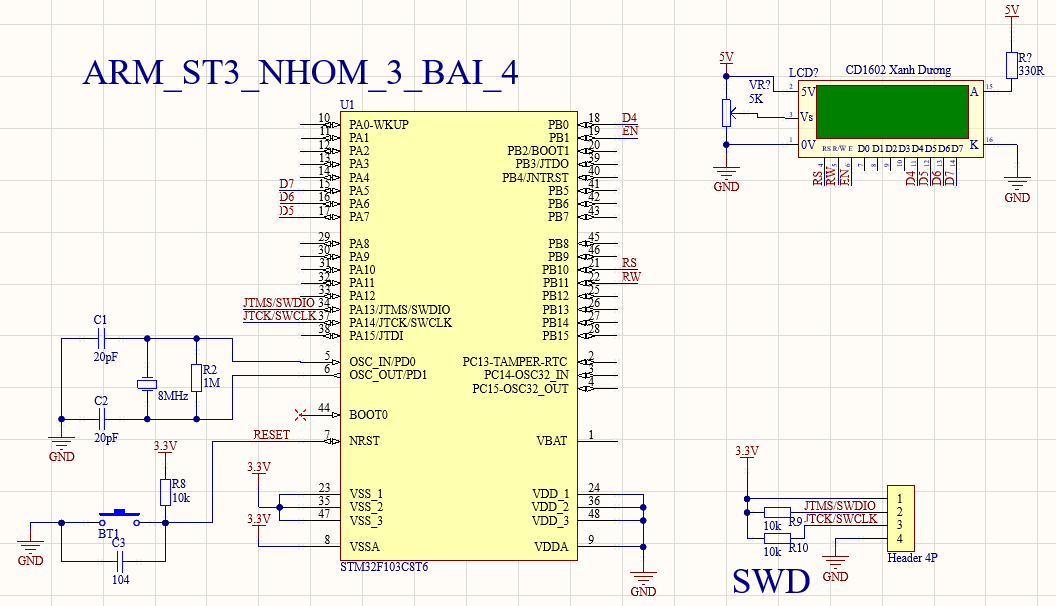
Sinh viên :

* 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:**



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

Description automatically generated

**Bước 3**: Chọn bộ mạch 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 chân GPIO Output

Graphical user interface

Description automatically generated

**Bước 6**: Đặt tên cho Project và chọn nơi lưu trữ, chọn phần mềm viết chương trình

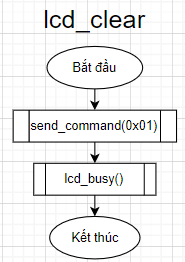
Graphical user interface, text, application, email

Description automatically generated

1. **Lưu đồ giải thuật:**

**Chương trình con:**

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Chương trình chính:**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

1. **Mã nguồn chương trình:**

**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 \*/

…

/\* USER CODE BEGIN PV \*/

uint16\_t i;

const unsigned char Font[] = {

0x00, 0x04, 0x0A, 0x0E, 0x11, 0x11, 0x0E, 0x00,//?

0x06, 0x0B, 0x0E, 0x01, 0x0F, 0x11, 0x0F, 0x00,//? huyen

0x00, 0x04, 0x02, 0x0E, 0x11, 0x11, 0x0E, 0x00,//?

0x05, 0x0A, 0x03, 0x0E, 0x11, 0x11, 0x0E, 0x00,//? nga

0x0C, 0x0A, 0x09, 0x1D, 0x09, 0x0A, 0x0C, 0x00,// ?

0x0C, 0x1A, 0x0E, 0x01, 0x0F, 0x11, 0x0F, 0x00,// ? sac

0x99};

const unsigned char Font2[] = {

0x05, 0x0E, 0x0A, 0x0E, 0x11, 0x1E, 0x0E, 0x00,//? nga

0x04, 0x0C, 0x0A, 0x0E, 0x11, 0x11, 0x0E, 0x00,//? sac

0x00, 0x00, 0x03, 0x12, 0x12, 0x12, 0x0C, 0x00,// uw

0x00, 0x00, 0x03, 0x0E, 0x11, 0x11, 0x0E, 0x00,// ow

0x99};

const unsigned char line1\_1[] ={0x54,0x72,0x01,0x6E,' ', 0x43, 0x00, 0x6E,0x67,' ',0x48,0x02,0x61};

const unsigned char line1\_2[] = {0x04,0x03,' ',0x54,0x75,0x05,0x6E,' ',0x44,0x75,0x79};

const unsigned char line1\_3[] = {0x4E,0x67,' ',0x51,0x75,0x01,0x63,

' ',0x44,0x02,0x03,0x6E,0x67};

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();

/\* USER CODE BEGIN 2 \*/

lcd\_config();

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

while (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

lcd\_clear();

nap\_tudien();

set\_cursor(0,2);

for(int i = 0 ; i < 13; i++)

{

display\_lcd(line1\_1[i]);

}

set\_cursor(1,4);

display\_text("20017691");

HAL\_Delay(5000);

lcd\_clear();

set\_cursor(0,2);

for(int i = 0 ; i < 11; i++)

{

display\_lcd(line1\_2[i]);

}

set\_cursor(1,4);

display\_text("20061261");

HAL\_Delay(5000);

lcd\_clear();

nap\_tudien2();

set\_cursor(0,1);

for(int i = 0 ; i < 13; i++)

{

display\_lcd(line1\_3[i]);

}

set\_cursor(1,4);

display\_text("20123251");

HAL\_Delay(5000);

}

/\* USER CODE END 3 \*/

}

1. **Video minh chứng:**

Link youtube: <https://www.youtube.com/watch?v=1a4OgqHPVIM>