

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

-----***-----



BÁO CÁO DỰ ÁN CUỐI KÌ

Môn học: Nhập môn hệ thống nhúng

ĐỀ TÀI

THIẾT KẾ VÀ THỰC HIỆN HỆ THỐNG LA BÀN SỐ

	Họ và tên (Full name)	Mã SV (ID)	Đóng góp (Contribution)
Thành viên 1 (Member 1)	Nguyễn Hoài Nam	21020040	Tìm hiểu datasheet, code LCD, MAG3110 và I2C.
Thành viên 2 (Member 2)	Hoàng Trung Kiên	21020921	Tìm hiểu, tra datasheet KL46Z, MAG3110, code nháy LED và khởi tạo các hàm xử lý ngắt.
Tên/Địa chỉ Repo trên Github	https://github.com/kienht23/compass-project		

Tóm tắt (Abstract - from 5 to 10 lines)
--

<p>Dự án này nhằm thiết kế và triển khai một hệ thống la bàn số bằng cách sử dụng bảng FRDM-KL46Z và cảm biến từ tính MAG3110. Hệ thống có khả năng xác định hướng và hiển thị thông tin trên màn hình LCD. Mục tiêu chính là vận dụng kiến thức về hệ thống nhúng để tạo ra một la bàn kỹ thuật số hoạt động với khả năng hiển thị hướng theo thời gian thực. Các chức năng quan trọng của hệ thống bao gồm chuyển đổi giữa các trạng thái hoạt động và không hoạt động thông qua nút bấm và phản hồi bằng đèn LED. Dự án này minh họa sự kết hợp hiệu quả giữa phần cứng và phần mềm để đạt được mục tiêu đề ra.</p>
--

Từ khóa (Keywords)

FRDM-KL46Z, MAG3110, LCD, LED, INTERRUPT
--

Document History

Version	Time	Revised by	Description
V0.1	29/05/2024	Nguyễn Hoài Nam Hoàng Trung Kiên	Original Version
V0.2	03/06/2024	Hoàng Trung Kiên	Modified the format of report
V0.3	04/06/2024	Nguyễn Hoài Nam	Final Version

Mục lục (Table of Contents)

Document History	4
Mục lục (Table of Contents)	5
1. Giới thiệu (Introduction)	6
2. Yêu cầu đối với thiết kế (Requirements)	6
2.1. <i>Yêu cầu đối với thiết kế</i>	<i>6</i>
2.2. <i>Đặc tả kỹ thuật (Specification)</i>	<i>7</i>
3. Thực hiện hệ thống (Implementation)	7
3.1. <i>Kiến trúc phần cứng (Hardware Architecture)</i>	<i>7</i>
3.1.1. <i>Khối xử lý trung tâm</i>	<i>8</i>
3.1.2. <i>Cảm biến đầu vào</i>	<i>10</i>
3.1.3. <i>Các LED</i>	<i>10</i>
3.1.4. <i>LCD</i>	<i>11</i>
3.2. <i>Lập trình phần mềm</i>	<i>11</i>
3.2.1. <i>Thiết lập Clock cho hệ thống</i>	<i>11</i>
3.2.2. <i>Khởi tạo các LED</i>	<i>12</i>
3.2.3. <i>Khởi tạo các Switch</i>	<i>12</i>
3.2.4. <i>Khởi tạo SysTick Timer</i>	<i>13</i>
3.2.5. <i>Thiết lập mức ưu tiên cho các ngắt</i>	<i>13</i>
3.2.6. <i>Chương trình điều khiển (Hàm main())</i>	<i>14</i>
4. Kiểm chứng (Validation)	16
5. Kết luận (Conclusion)	16
Danh sách hình (List of Figures)	18
Danh sách Bảng (List of Tables)	19
References	20

1. Giới thiệu (Introduction)

Mục tiêu của dự án: Vận dụng các kiến thức, kỹ năng đã được học để thiết kế và thực thi một hệ thống nhúng cho la bàn số. Hệ thống được thiết kế để có thể thực hiện bằng các mô-đun cảm biến từ trường và các ngoại vi có sẵn trên bo mạch FRDM-KL46z. Sinh viên cũng có thể sử dụng một bo mạch tương đương (dùng vi xử lý ARM) để thực hiện dự án.

Nội dung của dự án: Thiết kế một hệ thống nhúng có chức năng đọc cảm biến từ trường và hiển thị dữ liệu trên màn hình LCD như 1 chiếc la bàn số. Các công việc trong dự án là kết nối cảm biến từ trường với vi mạch để lấy giá trị, xử lý dữ liệu nhận được thành dữ liệu phương hướng để hiển thị trên màn hình LCD

2. Yêu cầu đối với thiết kế (Requirements)

2.1. Yêu cầu đối với thiết kế

Yêu cầu thiết kế:

- *Chức năng:* phát hiện phương hướng giống như một chiếc la bàn và hiển thị kết quả trên màn LCD.
- *Các đầu vào:*
 - Hệ thống sử dụng cảm biến từ trường MAG3110 để phát hiện phương hướng.
 - SW1 để chuyển đổi hệ thống giữa trạng thái hoạt động và trạng thái dừng. Khi hệ thống dừng hoạt động, nếu bấm SW1 bộ đếm chuyển sang trạng thái hoạt động, và ngược lại. Khi chuyển từ trạng thái dừng sang trạng thái hoạt động, hệ thống tiếp tục giám sát trạng thái của cảm biến và hiển thị trên LCD.
 - SW2 để xóa (reset) trạng thái hệ thống về trạng thái bình thường.
- *Các đầu ra:*
 - Hệ thống có 2 lối ra trạng thái: LED xanh nhấp nháy với tần số 1Hz khi hệ thống hoạt động; tắt khi hệ thống dừng hoạt động. LED đỏ nhấp ở tần số 2Hz khi hệ thống dừng hoạt động; tắt khi hệ thống ở trạng thái hoạt động.
 - LCD: Hiện thị hướng của la bàn
- Bộ đếm sử dụng một timer để xác định khoảng thời gian nhấp nháy các LED.

2.2. Đặc tả kỹ thuật (Specification)

- *Yêu Cầu Phần Cứng:*

- Vi điều khiển: FRDM-KL46z
- Cảm biến từ trường: MAG3110
- Màn hình LCD: Màn hình LCD tích hợp SLCD module.

- *Yêu Cầu Phần Mềm:*

- Ngôn ngữ lập trình: C
- Môi trường phát triển: Keil uVision.
- Giao tiếp: Giao tiếp I2C cho cảm biến với vi điều khiển

3. Thực hiện hệ thống (Implementation)

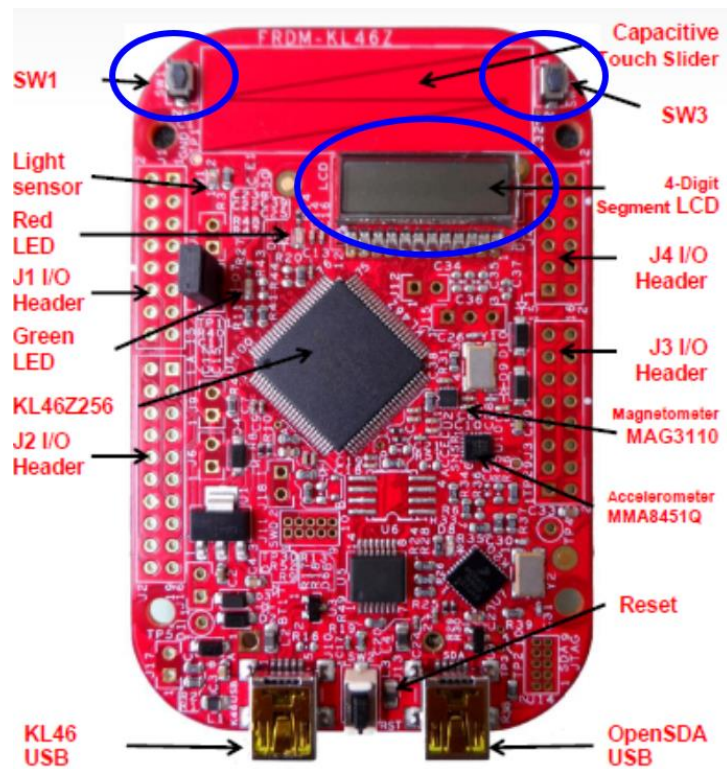
3.1. Kiến trúc phần cứng (Hardware Architecture)

Miêu tả chức năng của các ngoại vi có sẵn trên bo mạch FRDM-KL46Z được sử dụng để thực hiện hệ thống (**Bảng 1**);

Bảng 1: Mô tả các ngoại vi được sử dụng.

STT	Tên ngoại vi	Chức năng
1	SW1	Start/Stop bộ đếm
2	SW2	Reset
3	MAG3110	Trả về giá trị từ trường x, y, z
4	LCD	Hiện thị phương hướng
5	LED xanh	Nháy chu kì 1 Hz khi mạch hoạt động
6	LED đỏ	Nháy chu kì 2 Hz khi mạch dừng hoạt động

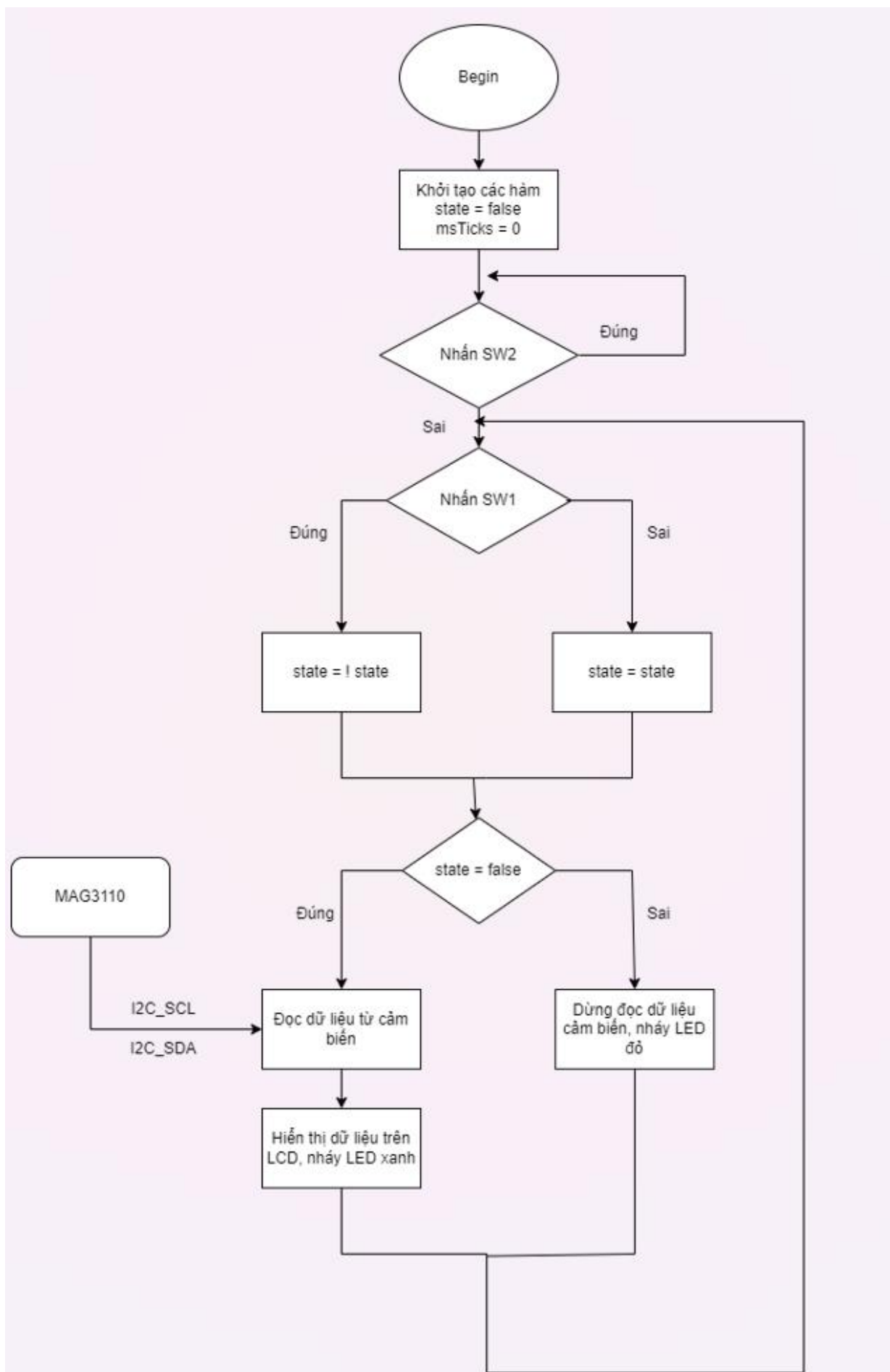
Các mô-đun phần cứng trên bo mạch FRDM-KL46Z được sử dụng để thực hiện hệ thống như sau:



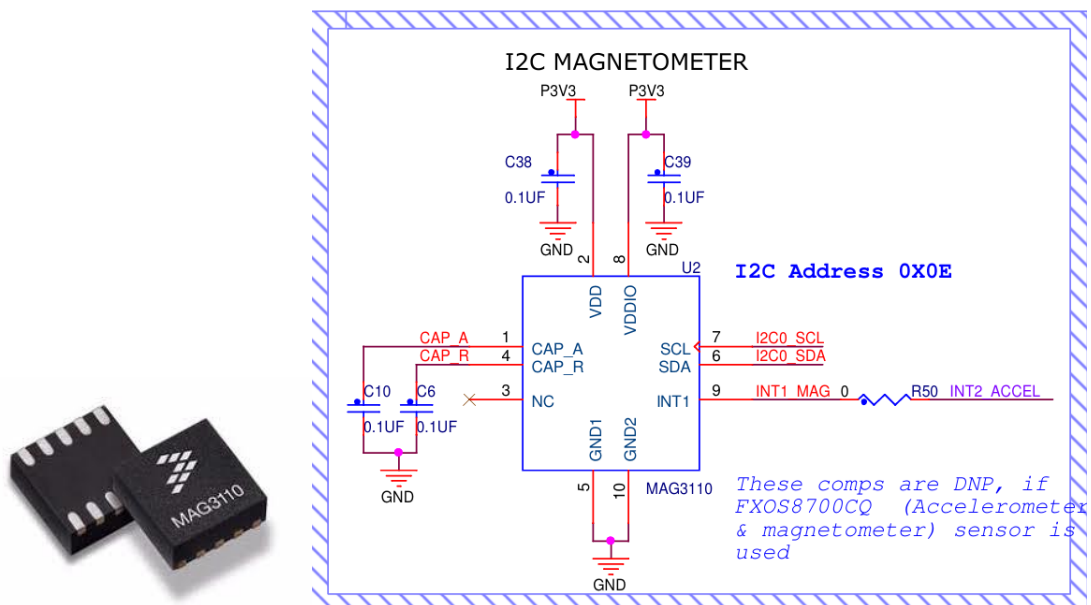
Hình 1. Bo mạch FRDM-KL46z và các mô đun phần cứng

3.1.1. Khối xử lý trung tâm

Lưu đồ thuật toán

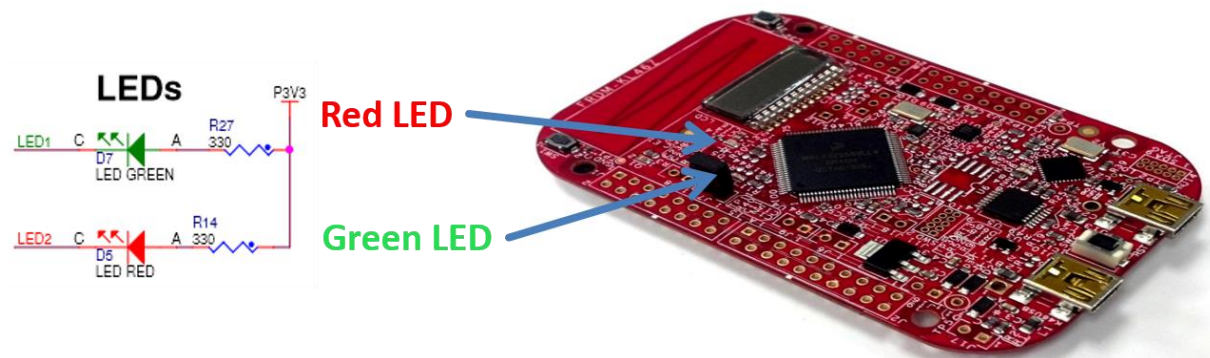


3.1.2. Cảm biến đầu vào



Hình 3: Cấu trúc của cảm biến đầu vào.

3.1.3. Các LED



Hình 4: Các LED trên bo mạch FRDM-KL46Z.

Bảng 2. Kết nối các LED trên bo mạch KL46Z

LED	KL46Z Pin
Red	PTE29
Green	PTD5

3.1.4. LCD

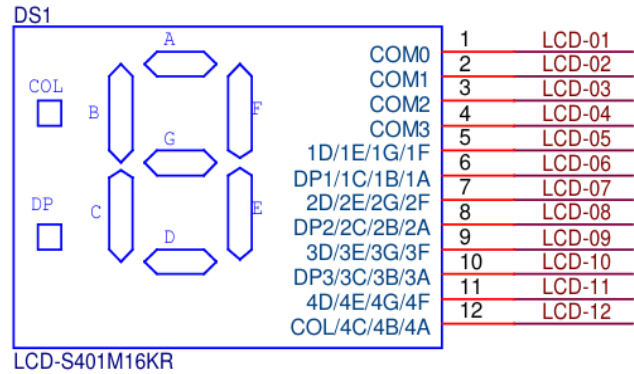
Table 4. sLCD connections

s401 pin	KL46 LCD Pin
1	LCD_P40 (COM0)
2	LCD_P52 (COM1)
3	LCD_P19 (COM2)
4	LCD_P18 (COM3)
5	LCD_P37
6	LCD_P17
7	LCD_P7
8	LCD_P8
9	LCD_P53
10	LCD_P38
11	LCD_P10
12	LCD_P11



Figure 6 s401 segments layout

7 Segment LCD



3.2. Lập trình phần mềm

3.2.1. Thiết lập Clock cho hệ thống

Mạch vi điều khiển được thiết lập với giá trị SystemCoreClock = 20971520 Hz.

Bật clock cho các cổng

- Điều khiển LCD:

```
SIM->SCGC5 |= SIM_SCGC5_PORTB_MASK | SIM_SCGC5_PORTC_MASK
              | SIM_SCGC5_PORTD_MASK | SIM_SCGC5_PORTE_MASK
              | SIM_SCGC5_SLCD_MASK;
//Enable Clock to ports B, C, D and E, and SegLCD Peripheral
```

- I2C giao tiếp giữa MAG3110 và chip:

```
SIM->SCGC5 |= SIM_SCGC5_PORTE_MASK; //enable clock port E
SIM->SCGC4 |= SIM_SCGC4_I2C0_MASK; //enable clock I2C0
```

- Điều khiển LED :

```
//LED Green PortD5
SIM->SCGC5 |= 1 << 12; //This enables the clock to PortD

//LED Red PortE29
SIM->SCGC5 |= 1 << 13; //This enables the clock to PortE
```

3.2.2. Khởi tạo các LED

```
//LED Green PortD5
SIM->SCGC5 |= 1 << 12; //This enables the clock to PortD
PORTD->PCR[5] |= 1 << 8; //Mux 001 | GPIO
PTD->PDDR |= 1 << 5; //This sets PTD5 as an output

//LED Red PortE29
SIM->SCGC5 |= 1 << 13; //This enables the clock to PortE
PORTE->PCR[29] |= 1 << 8; //Mux 001 | GPIO
PTE->PDDR |= 1 << 29; //This sets PTE29 as an output
```

3.2.3. Khởi tạo các Switch

a) Cấp xung clock cho cổng GPIO:

```
SIM->SCGC5 |= 1u << 11; //This enables the clock to PORTC

//SW1 PORTC3
PORTC->PCR[3] |= 1u << 8 | 1 << 1 | 1 << 0; //This sets GPIO and Enable pull-up resistor
PTC->PDDR &= ~(1u << 3); //This sets PTC3 as an input.
```

b) Thiết lập ngắt cho Switches:

```
void init_ITR_SW(void) {
    //Select falling edge interrupts for PORTC[3].
    PORTC->PCR[3] |= (0xA) << 16; //OR    PORTC->PCR[3] |= PORT_PCR_IRQC(0xA);

    NVIC_ClearPendingIRQ(31); // number for both PORTC and PORTD are 31
    NVIC_EnableIRQ(31);
}
```

```

//SW1 interrupt Handler
void PORTC_PORTD_IRQHandler(void) {

    //If SW1 not press => do nothing
    if ((PTC->PDIR & (1<<3)) != 0) {
        return;
    }else { //Toggle state System
        if (state == false) {
            PTE->PCOR |= 1 << 29;
            PTD->PSOR |= 1 << 5;
            state = true;
        }
        else {
            PTE->PSOR |= 1 << 29;
            PTD->PCOR |= 1 << 5;
            state = false;
        }
    }
}

```

3.2.4. Khởi tạo SysTick Timer

```

void Init_SysTick(){
    SysTick->CTRL |= SysTick_CTRL_CLKSOURCE_Msk
                  |SysTick_CTRL_TICKINT_Msk
                  |SysTick_CTRL_ENABLE_Msk;
    SysTick->LOAD = SystemCoreClock/1000;
    NVIC_SetPriority(-1,15);
}

```

3.2.5. Thiết lập mức ưu tiên cho các ngắt

Thiết lập mức ưu tiên cho SysTick:
 NVIC_SetPriority(-1,15);

3.2.6. Thiết lập hiển thị LCD

Thư viện lcd.h gồm có các hàm như:

- Hàm khởi tạo: SegLCD_Init(void)
- Hàm thiết lập hiển thị các số từ 0 đến 9: SegLCD_Set(uint8_t Value,uint8_t Digit)
- Hàm hiển thị số thập phân: SegLCD_DisplayDecimal(uint16_t Value)
- Hàm hiển thị lỗi: SegLCD_DisplayError(uint8_t ErrorNum)

3.2.7. Thiết lập giao tiếp I2C giữa cảm biến và vi điều khiển

Thư viện I2C.h gồm có các hàm như:

- Hàm khởi tạo xung Clock và các chế độ hoạt động: I2C_init(void)
- Hàm đọc giá trị từ thiết bị ngoại vi và trả về giá trị I2C_read(uint8_t register_address)

3.2.8. Thiết lập giá trị của cảm biến từ trường

Đưa về 3 giá trị data.x, data.y, data.z

```
mag3110 data;

uint8_t x1= I2C_read(0x01);
Delay(1);
uint8_t x2= I2C_read(0x02);
Delay(1);
uint8_t y1= I2C_read(0x03);
Delay(1);
uint8_t y2= I2C_read(0x04);
Delay(1);
uint8_t z1= I2C_read(0x05);
Delay(1);
uint8_t z2= I2C_read(0x06);
Delay(1);
int16_t raw_x_data = ((int16_t)(x1 << 8 | x2));
int16_t raw_y_data = ((int16_t)(y1 << 8 | y2));
int16_t raw_z_data = ((int16_t)(z1 << 8 | z2));

//Real data (Converted)
data.x = raw_x_data ;
data.y = raw_y_data ;
data.z = raw_z_data ;
```

3.2.9. Chương trình điều khiển (Hàm main())

```
#include "MKL46Z4.h"
```

```
#include "I2C.h"
```

```
#include "delay.h"
```

```
#include "mag.h"
```

```
#include "lcd.h"
```

```
#include<math.h>
```

```
#define true 1
```

```
#define false 0
```

```
int main() {
```

```

uint8_t volatile state = false;

Init_Systick();
I2C_init();
SegLCD_Init();
mag_Init();
init_LED();
init_SW();
init_ITR_SW();
Delay(1);
while(1){

    //SW2_RESET();
    mag3110 data = mag_read();
    float d = fabs((float)data.x);
    float theta = atan2(data.y-1000.0 , data.x-45); //tinh goc
    float theta_deg = theta * (180.0 / 3.14159) + 180.0 ;
    int thetaa = round(theta_deg);

    SegLCD_DisplayDecimal(thetaa);

    Delay(1000);

}
return 0;
}

```



Hình 5:..Kết quả

4. Kiểm chứng (Validation)

[Video](#)

5. Kết luận (Conclusion)

Dự án thiết kế và triển khai hệ thống la bàn số sử dụng vi mạch FRDM-KL46Z và cảm biến từ tính MAG3110 đã thành công trong việc hiện thực hóa một thiết bị định hướng kỹ thuật số. Hệ thống không chỉ xác định chính xác hướng mà còn hiển thị kết quả trực quan trên màn hình LCD, đáp ứng yêu cầu về khả năng hiển thị theo thời gian thực.

Việc tích hợp các thành phần phần cứng và phần mềm trong dự án này đã thể hiện sự ứng dụng hiệu quả của kiến thức về hệ thống nhúng. Các chức năng bổ sung như chuyển đổi giữa trạng thái hoạt động và không hoạt động thông qua các nút bấm, cùng với phản hồi bằng đèn LED, đã gia tăng tính linh hoạt và tiện dụng cho hệ thống.

Kết quả của dự án không chỉ minh chứng cho khả năng thực hiện của hệ thống la bàn số mà còn mở ra hướng phát triển cho các ứng dụng định hướng và đo lường khác. Những kinh

nghiệm và kiến thức thu được từ dự án này sẽ là nền tảng vững chắc cho các nghiên cứu và phát triển tương lai trong lĩnh vực hệ thống nhúng và cảm biến.

Tuy nhiên, trong quá trình thực hiện dự án, nhóm em đã gặp phải nhiều khó khăn trong việc tìm và chọn lọc tài liệu để tham khảo cho dự án, điều đó gây cản trở cho công việc hoàn thành dự án. Hơn nữa, việc kiểm tra hệ thống trong môi trường khác nhau đòi hỏi nhiều thời gian để đạt được sự ổn định và chính xác nhất định.

Danh sách hình (List of Figures)

Hình 1. Bo mạch FRDM-KL46z và các mô đun phần cứng 8

Hình 2: Cấu trúc của cảm biến đầu vào. 10

Hình 3: Các LED trên bo mạch FRDM-KL46Z. 10

Hình 4:..Kết quả 16

Danh sách Bảng (List of Tables)

Bảng 1: Mô tả các ngoại vi được sử dụng.....	7
Bảng 2. Kết nối các LED trên bo mạch KL46Z.....	10

References

- [1] KL46 Sub-Family Reference Manual, Rev. 3, July 2013. 2. Freescale Semiconductor, Inc.
- [2] NXP Semiconductors. FRDM-KL46Z User's Manual. Austin, NXP Semiconductors, 2013.
- [3] NXP Semiconductor. FRDM-KL46Z Schematic. 13, June 2013.
- [4] NXP Semiconductors. "MAG3110 Three-Axis, Digital Magnetometer Datasheet.
- [5] NXP Semiconductors. "MAG3110 Three-Axis Digital Magnetometer Reference Manual.