# **Báo Cáo Hệ Thống Phát Hiện Phương Hướng và Hiển Thị Kết Quả**

## **Mục lục (Table of Contents)**

1. Giới thiệu (Introduction)
2. Yêu cầu đối với thiết kế (Requirements)
   1. Yêu cầu đối với thiết kế
   2. Đặc tả kỹ thuật (Specification)
3. Thực hiện hệ thống (Implementation)
   1. Kiến trúc phần cứng (Hardware Architecture)
      1. Khối xử lý trung tâm
      2. Cảm biến đầu vào
      3. Các LED
      4. LCD
   2. Lập trình phần mềm
      1. Thiết lập Clock cho hệ thống
      2. Khởi tạo các LED
      3. Khởi tạo các Switch
      4. Khởi tạo Systick Timer
      5. Thiết lập mức ưu tiên cho các ngắt
      6. Chương trình điều khiển (Hàm main())
4. Kiểm chứng (Validation)
5. Kết luận (Conclusion)
6. Appendix A: Schematic
7. Appendix B: Code

## **1. Giới thiệu (Introduction)**

Hệ thống này được thiết kế để phát hiện phương hướng giống như một chiếc la bàn và hiển thị kết quả trên màn hình LCD. Hệ thống sử dụng cảm biến từ trường MAG3110 để đo từ trường và tính toán phương hướng. Kết quả được hiển thị trên màn hình LCD và trạng thái của hệ thống được hiển thị bằng các đèn LED.

## **2. Yêu cầu đối với thiết kế (Requirements)**

### **2.1. Yêu cầu đối với thiết kế**

* Sử dụng cảm biến từ trường MAG3110 để phát hiện phương hướng.
* Hiển thị kết quả trên màn hình LCD.
* Sử dụng đèn LED để hiển thị trạng thái hoạt động của hệ thống.
* Sử dụng nút nhấn để chuyển đổi trạng thái và đặt lại hệ thống.

### **2.2. Đặc tả kỹ thuật (Specification)**

* MAG3110 sẽ được cấu hình để đọc dữ liệu từ trường và tính toán phương hướng.
* Hệ thống có khả năng chuyển đổi trạng thái hoạt động và dừng thông qua nút nhấn.
* Khi chuyển đổi trạng thái, hệ thống sẽ tiếp tục hoặc dừng giám sát trạng thái của cảm biến và hiển thị kết quả trên màn hình LCD.
* Nút nhấn SW2 sẽ đặt lại hệ thống về trạng thái ban đầu.
* Đèn LED xanh và đỏ sẽ phản ánh trạng thái hoạt động của hệ thống dựa trên tần số nhấp nháy được thiết lập.

## **3. Thực hiện hệ thống (Implementation)**

### **3.1. Kiến trúc phần cứng (Hardware Architecture)**

#### **3.1.1. Khối xử lý trung tâm**

* **Vi điều khiển FRDM-KL46Z**: Được sử dụng để điều khiển toàn bộ hệ thống.

#### **3.1.2. Cảm biến đầu vào**

* **Cảm biến từ trường MAG3110**: Được sử dụng để phát hiện phương hướng.

#### **3.1.3. Các LED**

* **LED xanh**: Kết nối với GPIOB pin 5, nhấp nháy với tần số 1Hz khi hệ thống hoạt động.
* **LED đỏ**: Kết nối với GPIOE pin 29, nhấp nháy với tần số 2Hz khi hệ thống dừng.

#### **3.1.4. LCD**

* **Màn hình LCD**: Hiển thị kết quả phương hướng được tính toán từ cảm biến MAG3110.

### **3.2. Lập trình phần mềm**

#### **3.2.1. Thiết lập Clock cho hệ thống**

* Thiết lập clock cho các peripheral cần thiết như GPIO, I2C, Timer.

#### **3.2.2. Khởi tạo các LED**

* Cấu hình các chân GPIO cho LED xanh và LED đỏ.

#### **3.2.3. Khởi tạo các Switch**

* Cấu hình các chân GPIO cho nút nhấn SW1 và SW2, kích hoạt ngắt cho các chân này.

#### **3.2.4. Khởi tạo Systick Timer**

* Cấu hình Systick Timer để tạo ngắt định kỳ, điều khiển nhấp nháy của LED.

#### **3.2.5. Thiết lập mức ưu tiên cho các ngắt**

* Thiết lập mức ưu tiên cho các ngắt của nút nhấn và Timer.

#### **3.2.6. Chương trình điều khiển (Hàm main())**

c

Sao chép mã

#include "MKL46Z4.h"

#include "MAG3110.h"

#include "GPIO.h"

#include "LCD.h"

#include "Timer.h"

#include "global.h"

#include <stdbool.h>

#include <stdint.h>

volatile bool isSystemActive = false; // Khai báo biến toàn cục

int main(void) {

// Initialize components

GPIO\_Init();

Timer\_Init();

LCD\_Init();

MAG3110\_Init();

int16\_t x, y, z;

while (1) {

if (isSystemActive) {

MAG3110\_Read(&x, &y, &z);

Display\_Direction(x, y);

}

}

}

## **4. Kiểm chứng (Validation)**

* Kiểm tra hoạt động của hệ thống bằng cách thay đổi hướng của cảm biến và quan sát kết quả hiển thị trên màn hình LCD.
* Kiểm tra các nút nhấn để đảm bảo chúng chuyển đổi trạng thái và đặt lại hệ thống đúng cách.
* Kiểm tra tần số nhấp nháy của các LED để đảm bảo chúng hoạt động đúng như mong đợi.

## **5. Kết luận (Conclusion)**

Hệ thống phát hiện phương hướng và hiển thị kết quả trên màn hình LCD, cùng với việc sử dụng LED để phản ánh trạng thái hoạt động, là một ứng dụng hữu ích trong việc học tập và nghiên cứu về cảm biến từ trường và giao tiếp phần cứng.

## **Appendix A: Schematic**

## **Appendix B: Code**

**GLOBAL\_H**

#ifndef GLOBAL\_H\_

#define GLOBAL\_H\_

#include <stdbool.h>

#include <stdint.h>

// Biến toàn cục để lưu trạng thái hệ thống

extern volatile bool isSystemActive;

#endif /\* GLOBAL\_H\_ \*/

**GPIO.h**