|  |  |
| --- | --- |
| Học trực tuyến CNTT, học lập trình từ cơ bản đến nâng cao |  |

**IOT303 – Lập trình C nhúng nâng cao cho vi điều khiển**

**THUYẾT MINH ASSIGNMENT 2**

**GIAO TIẾP VỚI CẢM BIẾN NHIỆT ĐỘ - ĐỘ ẨM VÀ HIỂN THỊ CÁC THÔNG SỐ LÊN MÀN HÌNH LCD**

|  |  |
| --- | --- |
| Tên học viên:  Mã học viên:  Ngày báo cáo: | **Nguyễn Hoàng Dũng**  **FX19603**  **16/05/2024** |

**HÀ NỘI, 05/2024**

**MỤC LỤC**

[1. Danh sách thư viện 4](#_Toc166796957)

[2. Danh sách các biến toàn cục và các macro 4](#_Toc166796958)

[3. Danh sách các hàm 5](#_Toc166796959)

[4. Hàm main ( ) 6](#_Toc166796960)

[4.1. Mô tả 6](#_Toc166796961)

[4.2. Tham số truyền vào 7](#_Toc166796962)

[4.3. Giá trị trả về 7](#_Toc166796963)

[5. Hàm AppInitManager ( ) 7](#_Toc166796964)

[5.1. Mô tả 7](#_Toc166796965)

[5.2. Tham số truyền vào 7](#_Toc166796966)

[5.3. Giá trị trả về 7](#_Toc166796967)

[6. Hàm LCD\_Setup ( ) 7](#_Toc166796968)

[6.1. Mô tả 7](#_Toc166796969)

[6.2. Tham số truyền vào 7](#_Toc166796970)

[6.3. Giá trị trả về 7](#_Toc166796971)

[7. Hàm Sensor\_Init ( ) 7](#_Toc166796972)

[7.1. Mô tả 7](#_Toc166796973)

[7.2. Tham số truyền vào 7](#_Toc166796974)

[7.3. Giá trị trả về 7](#_Toc166796975)

[8. Hàm I2C\_Start ( ) 7](#_Toc166796976)

[8.1. Mô tả 7](#_Toc166796977)

[8.2. Tham số truyền vào 8](#_Toc166796978)

[8.3. Giá trị trả về 8](#_Toc166796979)

[9. Hàm I2C\_Address\_Direction ( ) 8](#_Toc166796980)

[9.1. Mô tả 8](#_Toc166796981)

[9.2. Tham số truyền vào 8](#_Toc166796982)

[9.3. Giá trị trả về 8](#_Toc166796983)

[10. Hàm I2C\_TransmitData ( ) 8](#_Toc166796984)

[10.1. Mô tả 8](#_Toc166796985)

[10.2. Tham số truyền vào 8](#_Toc166796986)

[10.3. Giá trị trả về 8](#_Toc166796987)

[11. Hàm I2C\_Receive\_ACK ( ) 8](#_Toc166796988)

[11.1. Mô tả 8](#_Toc166796989)

[11.2. Tham số truyền vào 8](#_Toc166796990)

[11.3. Giá trị trả về 8](#_Toc166796991)

[12. Hàm I2C\_Receive\_NACK ( ) 8](#_Toc166796992)

[12.1. Mô tả 8](#_Toc166796993)

[12.2. Tham số truyền vào 8](#_Toc166796994)

[12.3. Giá trị trả về 8](#_Toc166796995)

[13. Hàm I2C\_Stop ( ) 9](#_Toc166796996)

[13.1. Mô tả 9](#_Toc166796997)

[13.2. Tham số truyền vào 9](#_Toc166796998)

[13.3. Giá trị trả về 9](#_Toc166796999)

[14. Hàm TemHumSensor\_readRegister ( ) 9](#_Toc166797000)

[14.1. Mô tả 9](#_Toc166797001)

[14.2. Tham số truyền vào 9](#_Toc166797002)

[14.3. Giá trị trả về 9](#_Toc166797003)

[15. Hàm TemHumSensor\_getTemp ( ) 9](#_Toc166797004)

[15.1. Mô tả 9](#_Toc166797005)

[15.2. Tham số truyền vào 9](#_Toc166797006)

[15.3. Giá trị trả về 9](#_Toc166797007)

[16. Hàm TemHumSensor\_getHumi ( ) 9](#_Toc166797008)

[16.1: Mô tả 9](#_Toc166797009)

[16.2. Tham số truyền vào 9](#_Toc166797010)

[16.3. Giá trị trả về 9](#_Toc166797011)

[17. Hàm Update\_ValueSensor ( ) 10](#_Toc166797012)

[17.1. Mô tả 10](#_Toc166797013)

[17.2. Tham số truyền vào 10](#_Toc166797014)

[17.3. Giá trị trả về 10](#_Toc166797015)

[18. Hàm Update\_LCD ( ) 10](#_Toc166797016)

[18.1. Mô tả 10](#_Toc166797017)

[18.2. Tham số truyền vào 10](#_Toc166797018)

[18.3. Giá trị trả về 10](#_Toc166797019)

[19. Hàm printDataToLCD ( ) 10](#_Toc166797020)

[19.1. Mô tả 10](#_Toc166797021)

[19.2. Tham số truyền vào 10](#_Toc166797022)

[19.3. Giá trị trả về 10](#_Toc166797023)

[20. Hàm delay\_ms ( ) 10](#_Toc166797024)

[20.1. Mô tả 10](#_Toc166797025)

[20.2. Tham số truyền vào 10](#_Toc166797026)

[20.3. Giá trị trả về 10](#_Toc166797027)

[21. Luồng xử lý code. 11](#_Toc166797028)

[21.1. Lưu đồ thuật toán tổng quát 11](#_Toc166797029)

[21.2. Sơ đồ 1 12](#_Toc166797030)

[21.3. Sơ đồ 2 13](#_Toc166797031)

**NỘI DUNG THUYẾT MINH**

# Danh sách thư viện

|  |  |  |
| --- | --- | --- |
| **STT** | **Thư viện** | **Mô tả** |
|  | stdio.h | Thư viện hệ thống |
|  | stdint.h | Thư viện lưu trữ và làm việc với các dữ liệu ở cấp độ bit:   * uint8\_t: 8 bit * uint16\_t: 16 bit * uint32\_t: 32 bit |
|  | string.h | Thư viện cung cấp các hàm để xử lý chuỗi |
|  | timer.h | Thư viện cho phép quản lý các tác vụ theo thời gian |
|  | ucg.h | Thư viện cung cấp các hàm để in thông tin ra màn hình LCD |
|  | Ucglib.h | Thư viện cung cấp hàm để cấu hình các chân giao tiếp SPI của STM32 |
|  | stm32f401re\_rcc.h | Thư viện chứa các macro và hàm liên quan tới RCC:   * RCC\_AHB1PeriphClockCmd(): hàm bật / tắt cấp clock cho ngoại vi của bus AHB1 * RCC\_APB1PeriphClockCmd(): hàm bật / tắt cấp clock cho ngoại vi của bus APB1 * ... |
|  | stm32f401re\_gpio.h | Thư viện chứa các macro và hàm liên quan tới GPIO:   * GPIO\_Init(): cấu hình các chân GPIO * ... |
|  | stm32f401re\_i2c.h | Thư viện chứa các macro và hàm liên quan tới I2C:   * I2C\_Init(): Cấu hình chức năng cho I2C * I2C\_Cmd(): Cho phép I2C hoạt động * ... |
|  | kalman\_filter.h | Thư viện cung cấp các hàm để tạo bộ lọc |

# Danh sách các biến toàn cục và các macro

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến toàn cục và macro** | **Mô tả** |
|  | I2C\_SPEED | Tốc độ của I2C |
|  | I2C\_MASTER\_RCC | Địa chỉ RCC của I2C1 |
|  | I2C\_MASTER\_INSTANCE | Địa chỉ của I2C1 |
|  | I2C\_MASTER\_GPIO\_RCC | Địa chỉ RCC của GPIOB |
|  | I2C\_MASTER\_PORT | Địa chỉ của cổng GPIOB |
|  | SCL\_MASTER\_PIN | Địa chỉ của chân thứ 8 |
|  | SDA\_MASTER\_PIN | Địa chỉ của chân thứ 9 |
|  | SENSOR\_ADDR | Địa chỉ của cảm biến Si7020 |
|  | TEMP\_CMDCODE | Mã lệnh của nhiệt độ |
|  | HUMI\_CMDCODE | Mã lệnh của độ ẩm |
|  | TIME\_WAIT\_GET\_TEMP | Thời gian đợi để lấy giá trị nhiêt độ |
|  | TIME\_WAIT\_GET\_HUMI | Thời gian đợi để lấy giá trị độ ẩm |
|  | PERIOD\_UPDATE\_SENSOR | Chu kỳ cập nhật giá trị nhiệt độ, độ ẩm từ cảm biến |
|  | PERIOD\_UPDATE\_LCD | Chu kỳ cập nhật giá trị lên màn hình LCD |
|  | CHANGE\_VALUE\_TEMP | Giá trị nhiệt độ chênh lệch giữa 2 lần đo gần nhất |
|  | CHANGE\_VALUE\_HUMI | Giá trị độ ẩm chênh lệch giữa 2 lần đo gần nhất |

# Danh sách các hàm

|  |  |  |
| --- | --- | --- |
| **STT** | **Danh sách hàm** | **Mô tả** |
|  | int main () | Hàm chính của chương trình |
|  | static void AppInitManager (void); | Hàm khởi tạo các tài nguyên cần sử dụng |
|  | static void LCD\_Setup (void); | Hàm cài đặt các thông số cho LCD |
|  | static void Sensor\_Init (void); | Hàm giao tiếp với cảm biến (khởi tạo I2C1 ở chế độ master) |
|  | static void I2C\_Start (void); | Hàm truyền bit Start |
|  | static void I2C\_Address\_Direction (uint8\_t Address, uint8\_t Direction); | Hàm truyền bit địa chỉ |
|  | static void I2C\_TransmitData (uint8\_t data) | Hàm truyền dữ liệu của các thanh ghi trong cảm biến |
|  | static uint8\_t I2C\_Receive\_ACK (void); | Hàm nhận dữ liệu với ACK |
|  | static uint8\_t I2C\_Receive\_NACK (void) | Hàm nhận dữ liệu với NACK |
|  | static void I2C\_Stop (void) | Hàm truyền bit Stop |
|  | static void TemHumSensor\_readRegister (uint8\_t AddrSensor, uint8\_t pAddrReg, uint8\_t \*pDataRead, uint8\_t byDataLen, uint16\_t rDelay); | Hàm giao tiếp với các thanh ghi trong cảm biến để đọc giá trị |
|  | static double TemHumSensor\_getTemp (void) | Hàm xử lý dữ liệu nhiệt độ |
|  | static double TemHumSensor\_getHumi (void) | Hàm xử lý dữ liệu độ ẩm |
|  | static void Update\_ValueSensor (void) | Hàm cập nhật dữ liệu nhiệt độ, độ ẩm theo thời gian yêu cầu |
|  | static void Update\_LCD (void) | Hàm cập nhật thông tin lên màn hình LCD theo thời gian yêu cầu |
|  | static void printDataToLCD (void) | Hàm in dữ liệu ra màn hình LCD |
|  | static void delay\_ms (uint32\_t milisecond) | Hàm tạo thời gian delay |

# Hàm main ( )

## 4.1. Mô tả

Hàm main ( ) thực hiện việc gọi các hàm sau:

* AppInitManager ( )
* Trong hàm while (1) gọi các hàm:
* processTimerScheduler ( ): Xử lý các sự kiện theo thời gian đã cài đặt trước
* Update\_ValueSensor ( ): Cập nhật giá trị nhiệt độ, độ ẩm
* Update\_LCD ( ): Cập nhật thông tin lên màn hình LCD

## 4.2. Tham số truyền vào

Không có

## 4.3. Giá trị trả về

Không có

# Hàm AppInitManager ( )

## 5.1. Mô tả

Hàm AppInitManager ( ) thực hiện việc khởi tạo các tài nguyên cần sử dụng:

* SystemCoreClockUpdate ( ): Cấu hình clock của hệ thống là 84 MHz
* TimerInit ( ): Khởi tạo timer system tick để xử lý các sự kiện thời gian
* LCD\_Setup ( ): Cài đặt các thông số cho LCD
* Sensor\_Init ( ): Khởi tạo và cấu hình giao tiếp I2C
* KalmanFilterInit ( ): Khởi tạo bộ lọc Kalman

## 5.2. Tham số truyền vào

Không có

## 5.3. Giá trị trả về

Không có

# Hàm LCD\_Setup ( )

## 6.1. Mô tả

Hàm LCD\_Setup ( ) thực hiện việc cài đặt các thông số cho LCD

## 6.2. Tham số truyền vào

Không có

## 6.3. Giá trị trả về

Không có

# Hàm Sensor\_Init ( )

## 7.1. Mô tả

Hàm Sensor\_Init ( ) thực hiện việc giao tiếp với cảm biến (khởi tạo I2C1 ở chế độ master)

## 7.2. Tham số truyền vào

Không có

## 7.3. Giá trị trả về

Không có

# Hàm I2C\_Start ( )

## 8.1. Mô tả

Hàm I2C\_Start ( ) thực hiện việc truyền bit Start

## 8.2. Tham số truyền vào

Không có

## 8.3. Giá trị trả về

Không có

# Hàm I2C\_Address\_Direction ( )

## 9.1. Mô tả

Hàm I2C\_Address\_Direction ( ) thực hiện việc truyền bit địa chỉ

## 9.2. Tham số truyền vào

* Biến Address (kiểu dữ liệu uint8\_t): địa chỉ cần truyền
* Biến Direction (kiểu dữ liệu uint8\_t): truyền hay nhận dữ liệu

## 9.3. Giá trị trả về

Không có

# Hàm I2C\_TransmitData ( )

## 10.1. Mô tả

Hàm I2C\_TransmitData ( ) thực hiện việc truyền dữ liệu của các thanh ghi trong cảm biến

## 10.2. Tham số truyền vào

Biến data (kiểu dữ liệu uint8\_t): dữ liệu cần truyền

## 10.3. Giá trị trả về

Không có

# Hàm I2C\_Receive\_ACK ( )

## 11.1. Mô tả

Hàm I2C\_Receive\_ACK ( ) thực hiện việc nhận dữ liệu với ACK

## 11.2. Tham số truyền vào

Không có

## 11.3. Giá trị trả về

Biến data\_receive (kiểu dữ liệu uint8\_t)

# Hàm I2C\_Receive\_NACK ( )

## 12.1. Mô tả

Hàm I2C\_Receive\_NACK ( ) thực hiện việc nhận dữ liệu với NACK

## 12.2. Tham số truyền vào

Không có

## 12.3. Giá trị trả về

Biến data\_receive (kiểu dữ liệu uint8\_t)

# Hàm I2C\_Stop ( )

## 13.1. Mô tả

Hàm I2C\_Stop ( ) thực hiện việc truyền bit Stop

## 13.2. Tham số truyền vào

Không có

## 13.3. Giá trị trả về

Không có

# Hàm TemHumSensor\_readRegister ( )

## 14.1. Mô tả

Hàm TemHumSensor\_readRegister ( ) thực hiện việc giao tiếp với các thanh ghi trong cảm biến để đọc giá trị

## 14.2. Tham số truyền vào

* Biến AddrSensor (kiểu dữ liệu uint8\_t): Địa chỉ ngoại vi (cảm biến)
* Biến pAddrReg (kiểu dữ liệu uint8\_t): Địa chỉ của thanh ghi chứa dữ liệu nhiệt độ, độ ẩm (CmdCode)
* Con trỏ pDataRead (kiểu dữ liệu uint8\_t): Dữ liệu đọc được từ thanh ghi tương ứng
* Biến byDataLen (kiểu dữ liệu uint8\_t): Kích thước dữ liệu
* Biến rDelay (kiểu dữ liệu uint16\_t): Thời gian delay để đọc dữ liệu

## 14.3. Giá trị trả về

Không có

# Hàm TemHumSensor\_getTemp ( )

## 15.1. Mô tả

Hàm TemHumSensor\_getTemp ( ) thực hiện việc xử lý dữ liệu nhiệt độ

## 15.2. Tham số truyền vào

Không có

## 15.3. Giá trị trả về

Biến temperature (kiểu dữ liệu double)

# Hàm TemHumSensor\_getHumi ( )

## 16.1: Mô tả

Hàm TemHumSensor\_getHumi ( ) thực hiện việc xử lý dữ liệu độ ẩm

## 16.2. Tham số truyền vào

Không có

## 16.3. Giá trị trả về

Biến humidity (kiểu dữ liệu double)

# Hàm Update\_ValueSensor ( )

## 17.1. Mô tả

Hàm Update\_ValueSensor ( ) thực hiện việc cập nhật dữ liệu nhiệt độ, độ ẩm theo thời gian yêu cầu

## 17.2. Tham số truyền vào

Không có

## 17.3. Giá trị trả về

Không có

# Hàm Update\_LCD ( )

## 18.1. Mô tả

Hàm Update\_LCD ( ) thực hiện việc cập nhật dữ liệu lên màn hình LCD theo thời gian yêu cầu

## 18.2. Tham số truyền vào

Không có

## 18.3. Giá trị trả về

Không có

# Hàm printDataToLCD ( )

## 19.1. Mô tả

Hàm printDataToLCD ( ) thực hiện việc in dữ liệu ra màn hình LCD

## 19.2. Tham số truyền vào

Không có

## 19.3. Giá trị trả về

Không có

# Hàm delay\_ms ( )

## 20.1. Mô tả

Hàm delay\_ms ( ) thực hiện việc tạo thời gian delay

## 20.2. Tham số truyền vào

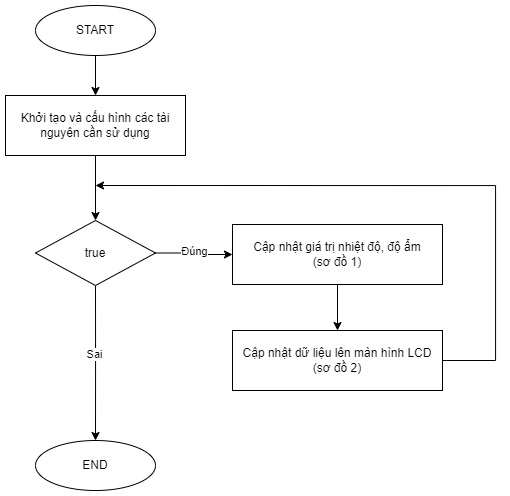
Biến milisecond (kiểu dữ liệu uint32\_t)

## 20.3. Giá trị trả về

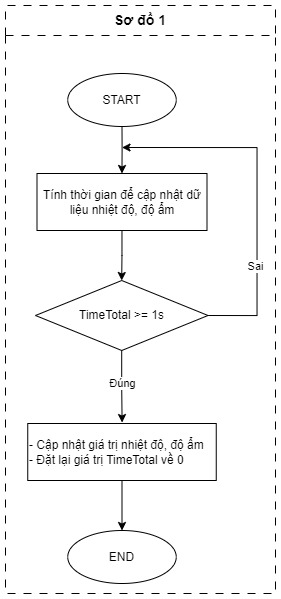
Không có

# Luồng xử lý code.

## 21.1. Lưu đồ thuật toán tổng quát



## 21.2. Sơ đồ 1



## 21.3. Sơ đồ 2

