|  |  |
| --- | --- |
| 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 3**

**GIAO TIẾP VI ĐIỀU KHIỂN STM32F4 VỚI PHẦN MỀM MÔ PHỎNG PC\_SIMULATOR\_KIT**

|  |  |
| --- | --- |
| Tên học viên:  Mã học viên:  Ngày báo cáo: | **Nguyễn Hoàng Dũng**  **FX19603**  **28/05/2024** |

**HÀ NỘI, 05/2024**

**MỤC LỤC**

[1. Danh sách thư viện 5](#_Toc168000924)

[2. Danh sách các biến toàn cục và các macro 6](#_Toc168000925)

[3. Danh sách các hàm 10](#_Toc168000926)

[4. Hàm main ( ) 11](#_Toc168000927)

[4.1. Mô tả 11](#_Toc168000928)

[4.2. Tham số truyền vào 12](#_Toc168000929)

[4.3. Giá trị trả về 12](#_Toc168000930)

[5. Hàm AppInitManager ( ) 12](#_Toc168000931)

[5.1. Mô tả 12](#_Toc168000932)

[5.2. Tham số truyền vào 12](#_Toc168000933)

[5.3. Giá trị trả về 12](#_Toc168000934)

[6. Hàm AppStateManager ( ) 12](#_Toc168000935)

[6.1. Mô tả 12](#_Toc168000936)

[6.2. Tham số truyền vào 12](#_Toc168000937)

[6.3. Giá trị trả về 13](#_Toc168000938)

[7. Hàm SetStateApp ( ) 13](#_Toc168000939)

[7.1. Mô tả 13](#_Toc168000940)

[7.2. Tham số truyền vào 13](#_Toc168000941)

[7.3. Giá trị trả về 13](#_Toc168000942)

[8. Hàm GetStateApp ( ) 13](#_Toc168000943)

[8.1. Mô tả 13](#_Toc168000944)

[8.2. Tham số truyền vào 13](#_Toc168000945)

[8.3. Giá trị trả về 13](#_Toc168000946)

[9. Hàm SerialCustom\_Init ( ) 13](#_Toc168000947)

[9.1. Mô tả 13](#_Toc168000948)

[9.2. Tham số truyền vào 13](#_Toc168000949)

[9.3. Giá trị trả về 13](#_Toc168000950)

[10. Hàm USART2\_Init ( ) 13](#_Toc168000951)

[10.1. Mô tả 13](#_Toc168000952)

[10.2. Tham số truyền vào 13](#_Toc168000953)

[10.3. Giá trị trả về 13](#_Toc168000954)

[11. Hàm USART2Modify\_IRQHandler ( ) 13](#_Toc168000955)

[11.1. Mô tả 13](#_Toc168000956)

[11.2. Tham số truyền vào 13](#_Toc168000957)

[11.3. Giá trị trả về 14](#_Toc168000958)

[12. Hàm LoadConfiguration ( ) 14](#_Toc168000959)

[12.1. Mô tả 14](#_Toc168000960)

[12.2. Tham số truyền vào 14](#_Toc168000961)

[12.3. Giá trị trả về 14](#_Toc168000962)

[13. Hàm DeviceStateMachine ( ) 14](#_Toc168000963)

[13.1. Mô tả 14](#_Toc168000964)

[13.2. Tham số truyền vào 14](#_Toc168000965)

[13.3. Giá trị trả về 14](#_Toc168000966)

[14. Hàm Clamp ( ) 15](#_Toc168000967)

[14.1. Mô tả 15](#_Toc168000968)

[14.2. Tham số truyền vào 15](#_Toc168000969)

[14.3. Giá trị trả về 15](#_Toc168000970)

[15. Hàm Increase\_LedLevel ( ) 15](#_Toc168000971)

[15.1. Mô tả 15](#_Toc168000972)

[15.2. Tham số truyền vào 15](#_Toc168000973)

[15.3. Giá trị trả về 15](#_Toc168000974)

[16. Hàm Decrease\_LedLevel ( ) 15](#_Toc168000975)

[16.1. Mô tả 15](#_Toc168000976)

[16.2. Tham số truyền vào 15](#_Toc168000977)

[16.3. Giá trị trả về 15](#_Toc168000978)

[17. Hàm MultiSensorScan ( ) 15](#_Toc168000979)

[17.1. Mô tả 15](#_Toc168000980)

[17.2. Tham số truyền vào 15](#_Toc168000981)

[17.3. Giá trị trả về 15](#_Toc168000982)

[18. Hàm Task\_MultiSensorScan ( ) 16](#_Toc168000983)

[18.1. Mô tả 16](#_Toc168000984)

[18.2. Tham số truyền vào 16](#_Toc168000985)

[18.3. Giá trị trả về 16](#_Toc168000986)

[19. Hàm Serial\_SendPacketCustom ( ) 16](#_Toc168000987)

[19.1. Mô tả 16](#_Toc168000988)

[19.2. Tham số truyền vào 16](#_Toc168000989)

[19.3. Giá trị trả về 16](#_Toc168000990)

[20. Hàm processSerialReceiverCustom ( ) 16](#_Toc168000991)

[20.1. Mô tả 16](#_Toc168000992)

[20.2. Tham số truyền vào 16](#_Toc168000993)

[20.3. Giá trị trả về 16](#_Toc168000994)

[21. Hàm PollRxBuff ( ) 16](#_Toc168000995)

[21.1. Mô tả 16](#_Toc168000996)

[21.2. Tham số truyền vào 16](#_Toc168000997)

[21.3. Giá trị trả về 16](#_Toc168000998)

[22. Hàm ButtonCmdSetState ( ) 17](#_Toc168000999)

[22.1. Mô tả 17](#_Toc168001000)

[22.2. Tham số truyền vào 17](#_Toc168001001)

[22.3. Giá trị trả về 17](#_Toc168001002)

[23. Hàm LedCmdSetState ( ) 17](#_Toc168001003)

[23.1. Mô tả 17](#_Toc168001004)

[23.2. Tham số truyền vào 17](#_Toc168001005)

[23.3. Giá trị trả về 17](#_Toc168001006)

[24. Hàm BuzzerCmdSetState ( ) 17](#_Toc168001007)

[24.1. Mô tả 17](#_Toc168001008)

[24.2. Tham số truyền vào 17](#_Toc168001009)

[24.3. Giá trị trả về 17](#_Toc168001010)

[25. Hàm LcdCmdSetState ( ) 17](#_Toc168001011)

[25.1. Mô tả 17](#_Toc168001012)

[25.2. Tham số truyền vào 17](#_Toc168001013)

[25.3. Giá trị trả về 18](#_Toc168001014)

[26. Luồng xử lý code. 18](#_Toc168001015)

[26.1. Sơ đồ thuật toán tổng quát 18](#_Toc168001016)

[26.2. Sơ đồ 1 19](#_Toc168001017)

[26.3. Sơ đồ 2 20](#_Toc168001018)

[26.4. Sơ đồ 3 21](#_Toc168001019)

[26.5. Sơ đồ 4 22](#_Toc168001020)

**NỘI DUNG THUYẾT MINH**

# Danh sách thư viện

|  |  |  |
| --- | --- | --- |
| **STT** | **Thư viện** | **Mô tả** |
|  | stdio.h | Thư viện có các hàm nhập xuất:  + sprintf: Gửi output đã được định dạng tới một chuỗi |
|  | 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 |
|  | system\_stm32f4xx.h | Thư viện chứa các hàm của hệ thống |
|  | timer.h | Thư viện cho phép quản lý các tác vụ theo thời gian |
|  | led.h | Thư viện cung cấp các hàm để sử dụng LED |
|  | buzzer.h | Thư viện cung cấp các hàm để sử dụng còi |
|  | melody.h | Thư viện cung cấp các hàm để chọn chế độ kêu của còi |
|  | lightsensor.h | Thư viện cung cấp các hàm để giao tiếp và lấy giá trị của cảm biến ánh sáng |
|  | temhumsensor.h | Thư viện cung cấp các hàm để giao tiếp và lấy giá trị của cảm biến nhiệt độ và độ ẩm |
|  | eventbutton.h | Thư viện cung cấp các hàm để sử dụng các sự kiện của nút nhấn |
|  | button.h | Thư viện cung cấp các hàm để sử dụng nút nhấn |
|  | ucg.h | Thư viện cung cấp các hàm để cài đặt thông số và in thông tin ra màn hình LCD |
|  | Ucglib.h | Thư viện cung cấp hàm để cấu hình các chân giao tiếp SPI của STM32 |
|  | buff.h | Thư viện cung cấp các hàm để sử dụng hàng đợi Queue |
|  | 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\_usart.h | Thư viện chứa các macro và hàm liên quan đến USART  + USART\_SendData(): hàm gửi dữ liệu  + USART\_GetFlagStatus(): hàm kiểm tra các cờ trạng thái  + ... |
|  | misc.h | Cho phép cấu hình ngắt và tạo ngắt của bộ quản lý ngắt NVIC |

# Danh sách các biến toàn cục và các macro

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến toàn cục và macro** | **Mô tả** |
|  | SIZE\_QUEUE\_DATA\_RX | Giá trị kích thước của bộ đệm dữ liệu (FIFO) |
|  | USART2\_GPIO\_PORT | Cổng GPIO của USART2 |
|  | USART2\_TX\_GPIO\_PIN | Chân GPIO của đường Tx |
|  | USART2\_RX\_GPIO\_PIN | Chân GPIO của đường Rx |
|  | USART2\_GPIO\_RCC | Clock của cổng GPIO có chức năng là USART2 |
|  | USART2\_RCC | Clock của USART2 |
|  | USART\_BAUDRATE | Tốc độ truyền của USART2 |
|  | RX\_BUFFER\_SIZE | Giá trị kích thước mảng lưu dữ liệu được lấy ra từ hàng đợi |
|  | FRAME\_SOF | Giá trị START của khung truyền |
|  | FRAME\_ACK | Giá trị ACK của khung truyền |
|  | FRAME\_NACK | Giá trị NACK của khung truyền |
|  | CXOR\_INIT\_VAL | Giá trị khởi tạo của Checkxor |
|  | CMD\_TYPE\_GET | Bản tin lấy thông tin, trạng thái của thiết bị từ máy tính (HOST) |
|  | CMD\_TYPE\_RES | Bản tin phản hồi thông tin, trạng thái của thiết bị lên máy tính (HOST) |
|  | CMD\_TYPE\_SET | Bản tin điều khiển thiết bị từ máy tính (HOST) |
|  | CMD\_OPT | Giá trị của bit Option trong khung truyền |
|  | CMD\_SEQUENCE | Giá trị của bit Sequence trong khung truyền |
|  | CMD\_ID\_LED | Giá trị ID của Led trong khung truyền |
|  | CMD\_ID\_BUZZER | Giá trị ID của còi trong khung truyền |
|  | CMD\_ID\_BUTTON | Giá trị ID của nút nhấn trong khung truyền |
|  | CMD\_ID\_TEMP\_SENSOR | Giá trị nhiệt độ trong khung truyền |
|  | CMD\_ID\_HUMI\_SENSOR | Giá trị độ ẩm trong khung truyền |
|  | CMD\_ID\_LIGHT\_SENSOR | Giá trị cường độ ánh sáng trong khung truyền |
|  | CMD\_ID\_LCD | Giá trị ID của LCD trong khung truyền |
|  | CMD\_ID | ID của bản tin |
|  | CMD\_TYPE | Loại bản tin |
|  | CMD\_DATA1 | Dữ liệu 1 |
|  | CMD\_DATA2 | Dữ liệu 2 |
|  | CMD\_DATA3 | Dữ liệu 3 |
|  | CMD\_DATA4 | Dữ liệu 4 |
|  | CMD\_DATA5 | Dữ liệu 5 |
|  | CMD\_DATA\_LED\_ID | Giá trị channel của Led trong khung truyền |
|  | CMD\_DATA\_LED\_COLOR | Giá trị màu Led trong khung truyền |
|  | CMD\_DATA\_LED\_NUMBLINK | Giá trị số lần nháy của Led trong khung truyền |
|  | CMD\_DATA\_LED\_INTERVAL | Giá trị khoảng thời gian nháy Led trong khung truyền |
|  | CMD\_DATA\_LED\_LASTSTATE | Giá trị trạng thái gần nhất của Led trong khung truyền |
|  | CMD\_DATA\_BUZZER\_STATE | Giá trị trạng thái của còi trong khung truyền |
|  | CMD\_DATA\_BUTTON\_EVENT | Giá trị sự kiện của nút nhấn trong khung truyền |
|  | CMD\_DATA\_BUTTON\_STATE | Giá trị trạng thái của nút nhấn trong khung truyền |
|  | CMD\_DATA\_LCD | Giá trị gửi về LCD trong khung truyền |
|  | CYCLE\_LED\_CHANGE | Chu kỳ thay đổi độ sáng LED |
|  | PERIOD\_SCAN\_MULTISENSOR | Chu kỳ quét của cảm biến |
|  | event\_api\_t | Kiểu dữ liệu enum cho các sự kiện của chương trình |
|  | \*event\_api\_p | Con trỏ kiểu dữ liệu enum trỏ tới các sự kiện của chương trình |
|  | state\_app\_t | Kiểu dữ liệu enum cho các trạng thái của chương trình |
|  | RX\_STATE | Trạng thái bộ nhận dữ liệu |
|  | USART\_STATE | Trạng thái của USART |
|  | g\_ucg | Biến để cài đặt các thông số màn hình LCD |
|  | g\_eCurrentState | Biến trạng thái hiện tại của chương trình |
|  | g\_ledRed | Biến thể hiện trạng thái đèn LED, khi nhấn nút B1 |
|  | g\_ledGreen | Biến thể hiện trạng thái đèn LED, khi nhấn nút B2 |
|  | g\_ledWhite | Biến thể hiện trạng thái đèn LED, khi nhấn nút B4 |
|  | g\_ledBlue | Biến thể hiện trạng thái đèn LED, khi nhấn nút B5 |
|  | g\_ledChange | Biến dùng để thay đổi trạng thái LED khi nhấn giữ nút B1/B5 |
|  | g\_idTimerStartIncrease | Biến lưu giá trị định danh ID của hàm TimerStart khi nhấn giữ nút B1 |
|  | g\_idTimerStartDecrease | Biến lưu giá trị định danh ID của hàm TimerStart khi nhấn giữ nút B5 |
|  | g\_idTimerDisplayLCD | Biến lưu giá trị định danh ID của hàm TimerStart khi gọi hàm MultiSensorScan |
|  | g\_idTimerSensorUpdate | Biến lưu giá trị định danh ID của hàm TimerStart khi gọi hàm Task\_multiSensorScan |
|  | g\_B3Count | Biến đếm số lần nhấn nút B3 khi thao tác trên phần mềm mô phỏng |
|  | g\_temperature | Biến lưu giá trị nhiệt độ được cập nhật từ cảm biến |
|  | g\_humidity | Biến lưu giá trị độ ẩm được cập nhật từ cảm biến |
|  | g\_light | Biến lưu giá trị cường độ ánh sáng được cập nhật từ cảm biến |
|  | g\_RxBufState | Biến lưu trạng thái quá trình nhận dữ liệu của bộ đệm |
|  | g\_IndexRxBuf | Biến lưu vị trí phần tử, của mảng lưu dữ liệu được lấy ra từ hàng đợi |
|  | g\_byCheckXorRxBuf | Biến lưu giá trị checkxor của khung truyền |
|  | g\_strRxBufData | Mảng lưu dữ liệu được truyền vào FIFO (dữ liệu đầu vào) |
|  | g\_serialQueueRx | Con trỏ để trỏ đến đối tượng FIFO |
|  | g\_strRxBuffer | Mảng lưu dữ liệu được lấy ra từ hàng đợi |
|  | g\_Data\_Receive | Biến lưu dữ liệu nhận được gần nhất của ngoại vi USART |
|  | g\_strTemp[30] | Mảng lưu giá trị nhiệt độ |
|  | g\_strHumi[30] | Mảng lưu giá trị độ ẩm |
|  | g\_strLight[30] | Mảng lưu giá trị cường độ ánh sáng |

# Danh sách các hàm

|  |  |  |
| --- | --- | --- |
| **STT** | **Danh sách hàm** | **Mô tả** |
|  | int main () | Hàm chính của chương trình |
|  | void AppInitManager (void); | Hàm khởi tạo các tài nguyên cần sử dụng |
|  | void AppStateManager (uint8\_t event); | Hàm xử lý các sự kiện của chương trình chính |
|  | void SetStateApp (state\_app\_t state); | Hàm SET trạng thái của chương trình |
|  | state\_app\_t GetStateApp (void); | Hàm lấy trạng thái hiện tại của chương trình |
|  | void SerialCustom\_Init (void); | Hàm khởi tạo bộ đệm FIFO và USART2 |
|  | void USART2\_Init (void); | Hàm cấu hình giao tiếp USART2 |
|  | void USART2Modify\_IRQHandler (void); | Hàm thực thi ngắt USART2 |
|  | void LoadConfiguration (void); | Hàm hiển thị thông tin lên LCD và PC\_Simulator\_KIT khi bắt đầu cắm nguồn cho thiết bị |
|  | void DeviceStateMachine (uint8\_t event); | Hàm điều khiển Led, Buzzer và hiển thị thông tin sau khi nhấn các nút trên bộ KIT. Đồng thời, gửi các bản tin lên phần mềm mô phỏng |
|  | uint8\_t Clamp (uint8\_t value, uint8\_t min , uint8\_t max); | Hàm tạo khoảng giá trị |
|  | void Increase\_LedLevel (void); | Hàm tăng độ sáng đèn Led |
|  | void Decrease\_LedLevel (void); | Hàm giảm độ sáng đèn Led |
|  | void MultiSensorScan (void); | Hàm cập nhật giá trị nhiệt độ, độ ẩm, cường độ ánh sáng |
|  | void Task\_multiSensorScan (void); | Hàm lấy giá trị nhiệt độ, độ ẩm, cường độ ánh sáng từ cảm biến và hiển thị lên LCD. Đồng thời, gửi các giá trị đó lên PC\_Simulator\_KIT |
|  | void Serial\_SendPacketCustom (uint8\_t byOption,uint8\_t byCmdId, uint8\_t byCmdType, uint8\_t \*pPayload, uint8\_t byLengthPayload); | Hàm gửi đoạn text lên PC\_Simulator\_KIT |
|  | void processSerialReceiverCustom (void); | Hàm kiểm tra và xử lý dữ liệu nhận về từ PC\_Simulator\_KIT |
|  | uint8\_t PollRxBuff (void); | Hàm xử lý các bản tin nhận được theo định dạng |
|  | void ButtonCmdSetState (uint8\_t button\_event, uint8\_t button\_state); | Hàm điều khiển nút nhấn khi nhận được bản tin từ PC\_Simulator\_KIT |
|  | void LedCmdSetState (uint8\_t led\_id, uint8\_t led\_color, uint8\_t led\_num\_blink, uint8\_t led\_interval, uint8\_t led\_last\_state); | Hàm điều khiển bật/tắt led RGB trên KIT khi nhận được bản tin từ PC\_Simulator\_KIT |
|  | void BuzzerCmdSetState (uint8\_t buzzer\_state); | Hàm điều khiển bật/tắt còi khi nhận được bản tin từ PC\_Simulator\_KIT |
|  | void LcdCmdSetState (char \*text); | Hàm hiển thị đoạn text lên màn hình LCD khi nhận được bản tin từ PC\_Simulator\_KIT |

# Hàm main ( )

## 4.1. Mô tả

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

* AppInitManager ()
* SetStateApp (): Khởi tạo chương trình chính về trạng cũ của thiết bị trước khi tắt nguồn
* EventSchedulerAdd (): Thêm các sự kiện khởi tạo khi hệ thống bắt đầu hoạt động vào Queue
* Trong hàm while(1) gọi các hàm:
* processTimerScheduler (): Xử lý các sự kiện theo thời gian đã cài đặt trước
* processEventScheduler (): Xử lý các sự kiện khi có sự kiện nhấn nút
* processSerialReceiverCustom (): Xử lý các bản tin nhận được theo định dạng từ phần mềm mô phỏng

## 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
* EventSchedulerInit (AppStateManager): Khởi tạo bộ đệm lưu trữ danh sách sự kiện của chương trình
* EventButton\_Init (): Cấu hình chân GPIO của các nút nhấn trên mạch
* BuzzerControl\_Init (): Cấu hình chân GPIO của còi trên mạch
* LedControl\_Init (): Cấu hình chân GPIO của các led RGB trên mạch
* TemHumSensor\_Init (): Cấu hình ngoại vi I2C để giao tiếp với cảm biến nhiệt độ - độ ẩm
* LightSensor\_Init (ADC\_READ\_MODE\_DMA): Cấu hình ngoại vi ADC hoạt động ở chế độ DMA để đọc dữ liệu của cảm biến ánh sáng
* SerialCustom\_Init (): Khởi tạo bộ đệm FIFO và USART
* Ucglib4WireSWSPI\_begin (&ucg, UCG\_FONT\_MODE\_SOLID): Cấu hình các chân giao tiếp SPI của STM32 với màn hình LCD
* ucg\_ClearScreen (&ucg): Xóa màn hình hiển thị LCD
* ucg\_SetFont (&ucg, ucg\_font\_ncenR10\_hf): Cài đặt font chữ kiểu ncenR10\_hr để hiển thị
* ucg\_SetColor (&ucg, 0, 255, 255, 255): SET màu trắng cho text hiển thị
* ucg\_SetColor (&ucg, 1, 0, 0, 0): SET màu đen cho nền hiển thị (backgroup)
* ucg\_SetRotate180 (&ucg): Xoay màn hình 1 góc 180 độ

## 5.2. Tham số truyền vào

Không có

## 5.3. Giá trị trả về

Không có

# Hàm AppStateManager ( )

## 6.1. Mô tả

Hàm AppStateManager ( ) thực hiện việc xử lý các sự kiện của chương trình chính

## 6.2. Tham số truyền vào

Biến event (kiểu dữ liệu uint8\_t): Sự kiện khi bấm nút trên bộ KIT

## 6.3. Giá trị trả về

Không có

# Hàm SetStateApp ( )

## 7.1. Mô tả

Hàm SetStateApp ( ) thực hiện việc đặt trạng thái của chương trình khi bật thiết bị

## 7.2. Tham số truyền vào

Biến state (kiểu dữ liệu enum: state\_app\_t): Trạng thái của chương trình

## 7.3. Giá trị trả về

Không có

# Hàm GetStateApp ( )

## 8.1. Mô tả

Hàm GetStateApp ( ) thực hiện việc lấy trạng thái hiện tại của chương trình

## 8.2. Tham số truyền vào

Không có

## 8.3. Giá trị trả về

Trả về giá trị g\_eCurrentState: Trạng thái hiện tại của chương trình

# Hàm SerialCustom\_Init ( )

## 9.1. Mô tả

Hàm SerialCustom\_Init ( ) thực hiện việc khởi tạo bộ đệm FIFO và USART2

## 9.2. Tham số truyền vào

Không có

## 9.3. Giá trị trả về

Không có

# Hàm USART2\_Init ( )

## 10.1. Mô tả

Hàm USART2\_Init ( ) thực hiện việc cấu hình giao tiếp USART2

## 10.2. Tham số truyền vào

Không có

## 10.3. Giá trị trả về

Không có

# Hàm USART2Modify\_IRQHandler ( )

## 11.1. Mô tả

Hàm USART2Modify\_IRQHandler ( ) thực hiện việc thực thi ngắt USART2

## 11.2. Tham số truyền vào

Không có

## 11.3. Giá trị trả về

Không có

# Hàm LoadConfiguration ( )

## 12.1. Mô tả

Hàm LoadConfiguration ( ) thực hiện việc hiển thị thông tin lên LCD và PC\_Simulator\_KIT khi bắt đầu cắm nguồn cho thiết bị

## 12.2. Tham số truyền vào

Không có

## 12.3. Giá trị trả về

Không có

# Hàm DeviceStateMachine ( )

## 13.1. Mô tả

Hàm DeviceStateMachine ( ) thực hiện việc điều khiển Led, Buzzer và hiển thị thông tin lên LCD sau khi nhấn các nút trên bộ KIT. Đồng thời, gửi bản tin cập nhật trạng thái màu led tương ứng, trạng thái còi lên PC\_Simulator\_KIT

* Nhấn nút B3 năm lần khi đó tất cả các Led trên Kit mở rộng sẽ nháy năm lần màu GREEN và hiển thị thông tin sau thiết bị lên màn hình LCD:

*Device: Board STM32 Nucleo*

*Code: STM32F401RE\_NUCLEO*

*Manufacturer: STMicroelectronics*

*Kit expansion: Lumi Smarthome*

*Project: Simulator touch switch*

Sau khi hiển thị các thông tin trên, thì hiển thị thông tin nhiệt độ, độ ẩm, cường độ ánh sáng được cập nhật từ cảm biến lên màn hình LCD

* Nhấn các nút B1, B2, B4, B5 một lần để điều khiển bật/tắt màu tương ứng RED, GREEN, WHITE, BLUE của tất cả các led RGB trên Kit mở rộng và còi sẽ kêu một bíp một lần. Đồng thời, gửi bản tin cập nhật trạng thái màu led tương ứng, trạng thái còi kêu một tiếng bíp lên phần mềm mô phỏng
* Nhấn giữ B1/B5 để điều khiển tăng/giảm cường độ sáng của led RGB:
* Nhấn giữ nút B1 với thời gian T lớn hơn một giây để điều khiển tăng độ sáng của led và nhả ra thì dừng điều khiển.
* Nhấn giữ nút B5 với thời gian T lớn hơn một giây để điều khiển giảm độ sáng của led và nhả ra thì dừng điều khiển.

## 13.2. Tham số truyền vào

Biến event (kiểu dữ liệu uint8\_t): Sự kiện khi bấm nút trên bộ KIT

## 13.3. Giá trị trả về

Không có

# Hàm Clamp ( )

## 14.1. Mô tả

Hàm Clamp ( ) thực hiện việc tạo khoảng giá trị của biến

## 14.2. Tham số truyền vào

* Biến value (kiểu dữ liệu uint8\_t): Giá trị hiện tại
* Biến min (kiểu dữ liệu uint8\_t): Giá trị thấp nhất
* Biến max (kiểu dữ liệu uint8\_t): Giá trị cao nhất

## 14.3. Giá trị trả về

Trả về giá trị value

# Hàm Increase\_LedLevel ( )

## 15.1. Mô tả

Hàm Increase\_LedLevel ( ) thực hiện việc tăng độ sáng đèn LED khi nhấn giữ nút B1

## 15.2. Tham số truyền vào

Không có

## 15.3. Giá trị trả về

Không có

# Hàm Decrease\_LedLevel ( )

## 16.1. Mô tả

Hàm Decrease\_LedLevel ( ) thực hiện việc giảm độ sáng đèn LED khi nhấn giữ nút B5

## 16.2. Tham số truyền vào

Không có

## 16.3. Giá trị trả về

Không có

# Hàm MultiSensorScan ( )

## 17.1. Mô tả

Hàm MultiSensorScan ( ) thực hiện việc cập nhật giá trị nhiệt độ, độ ẩm, cường độ ánh sáng

## 17.2. Tham số truyền vào

Không có

## 17.3. Giá trị trả về

Không có

# Hàm Task\_MultiSensorScan ( )

## 18.1. Mô tả

Hàm Task\_MultiSensorScan ( ) thực hiện việc lấy giá trị nhiệt độ, độ ẩm, cường độ ánh sáng từ cảm biến và hiển thị lên LCD. Đồng thời, gửi các giá trị đó lên PC\_Simulator\_KIT

## 18.2. Tham số truyền vào

Không có

## 18.3. Giá trị trả về

Không có

# Hàm Serial\_SendPacketCustom ( )

## 19.1. Mô tả

Hàm Serial\_SendPacketCustom ( ) thực hiện việc gửi đoạn text lên PC\_Simulator\_KIT

## 19.2. Tham số truyền vào

* Biến byOption (kiểu dữ liệu uint8\_t): Byte Option của khung truyền
* Biến byCmdId (kiểu dữ liệu uint8\_t): Byte CmdID của khung truyền
* Biến byCmdType (kiểu dữ liệu uint8\_t): Byte CmdType của khung truyền
* Con trỏ pPayload (kiểu dữ liệu uint8\_t): Byte Data của khung truyền
* Biến byLengthPayload (kiểu dữ liệu uint8\_t): Kích thước của Data

## 19.3. Giá trị trả về

Không có

# Hàm processSerialReceiverCustom ( )

## 20.1. Mô tả

Hàm processSerialReceiverCustom ( ) thực hiện việc kiểm tra và xử lý dữ liệu nhận về từ PC\_Simulator\_KIT

## 20.2. Tham số truyền vào

Không có

## 20.3. Giá trị trả về

Không có

# Hàm PollRxBuff ( )

## 21.1. Mô tả

Hàm PollRxBuff ( ) thực hiện việc xử lý các bản tin nhận được theo định dạng

## 21.2. Tham số truyền vào

Không có

## 21.3. Giá trị trả về

Biến byUartState (kiểu dữ liệu uint8\_t): Trạng thái của USART

# Hàm ButtonCmdSetState ( )

## 22.1. Mô tả

Hàm ButtonCmdSetState ( ) thực hiện việc điều khiển nút nhấn khi nhận được bản tin từ PC\_Simulator\_KIT

## 22.2. Tham số truyền vào

* Biến button\_event (kiểu dữ liệu uint8\_t): Sự kiện nút nhấn tương ứng trên KIT
* Biến button\_state (kiểu dữ liệu uint8\_t): Trạng thái của nút nhấn

## 22.3. Giá trị trả về

Không có

# Hàm LedCmdSetState ( )

## 23.1. Mô tả

Hàm LedCmdSetState ( ) thực hiện việc điều khiển bật/tắt led RGB trên KIT khi nhận được bản tin từ PC\_Simulator\_KIT

## 23.2. Tham số truyền vào

* Biến led\_id (kiểu dữ liệu uint8\_t): LED tương ứng trên KIT
* Biến led\_color (kiểu dữ liệu uint8\_t): Màu sắc tương ứng của LED
* Biến led\_num\_blink (kiểu dữ liệu uint8\_t): Số lần nháy LED
* Biến led\_interval (kiểu dữ liệu uint8\_t): Chu kỳ bật tắt của LED
* Biến led\_last\_state (kiểu dữ liệu uint8\_t): Trạng thái cuối của LED

## 23.3. Giá trị trả về

Không có

# Hàm BuzzerCmdSetState ( )

## 24.1. Mô tả

Hàm BuzzerCmdSetState ( ) thực hiện việc điều khiển bật/tắt còi khi nhận được bản tin từ PC\_Simulator\_KIT

## 24.2. Tham số truyền vào

Biến buzzer\_state (kiểu dữ liệu uint8\_t): Trạng thái của còi

## 24.3. Giá trị trả về

Không có

# Hàm LcdCmdSetState ( )

## 25.1. Mô tả

Hàm LcdCmdSetState ( ) thực hiện việc hiển thị đoạn text lên màn hình LCD khi nhận được bản tin từ PC\_Simulator\_KIT

## 25.2. Tham số truyền vào

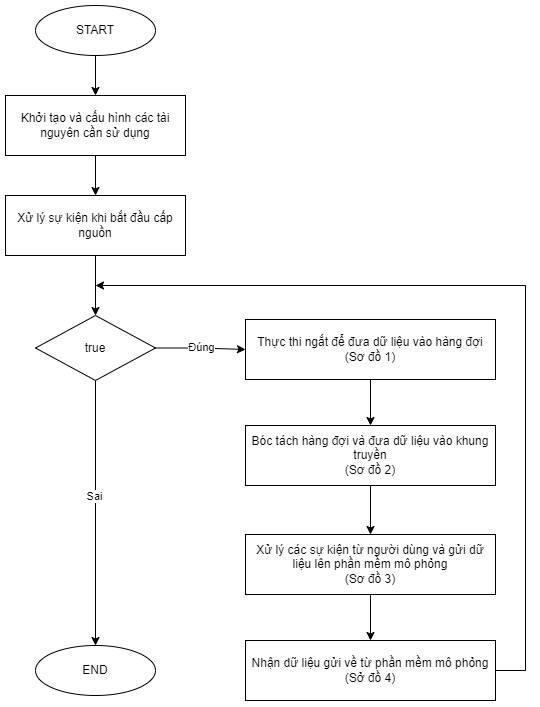
Con trỏ text (kiểu dữ liệu char): Đoạn text cần hiển thị

## 25.3. Giá trị trả về

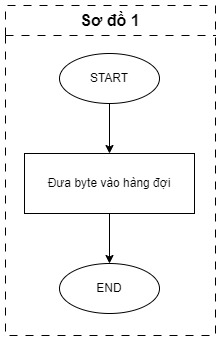
Không có

# Luồng xử lý code.

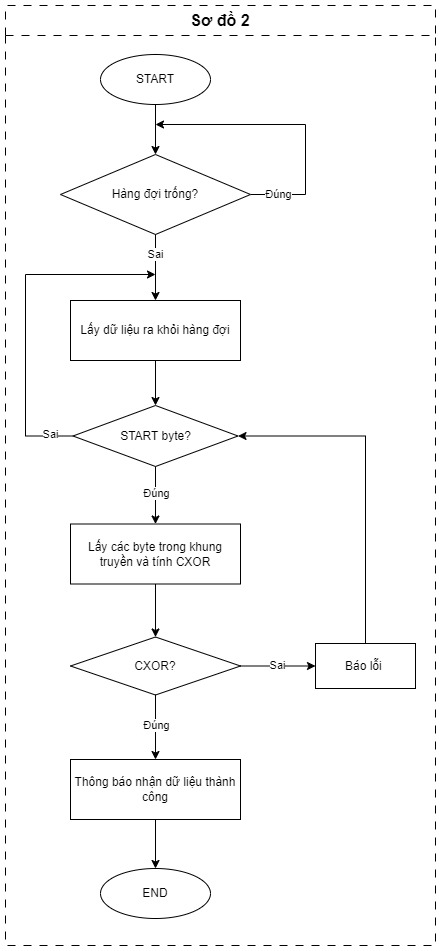
## 26.1. Sơ đồ thuật toán tổng quát



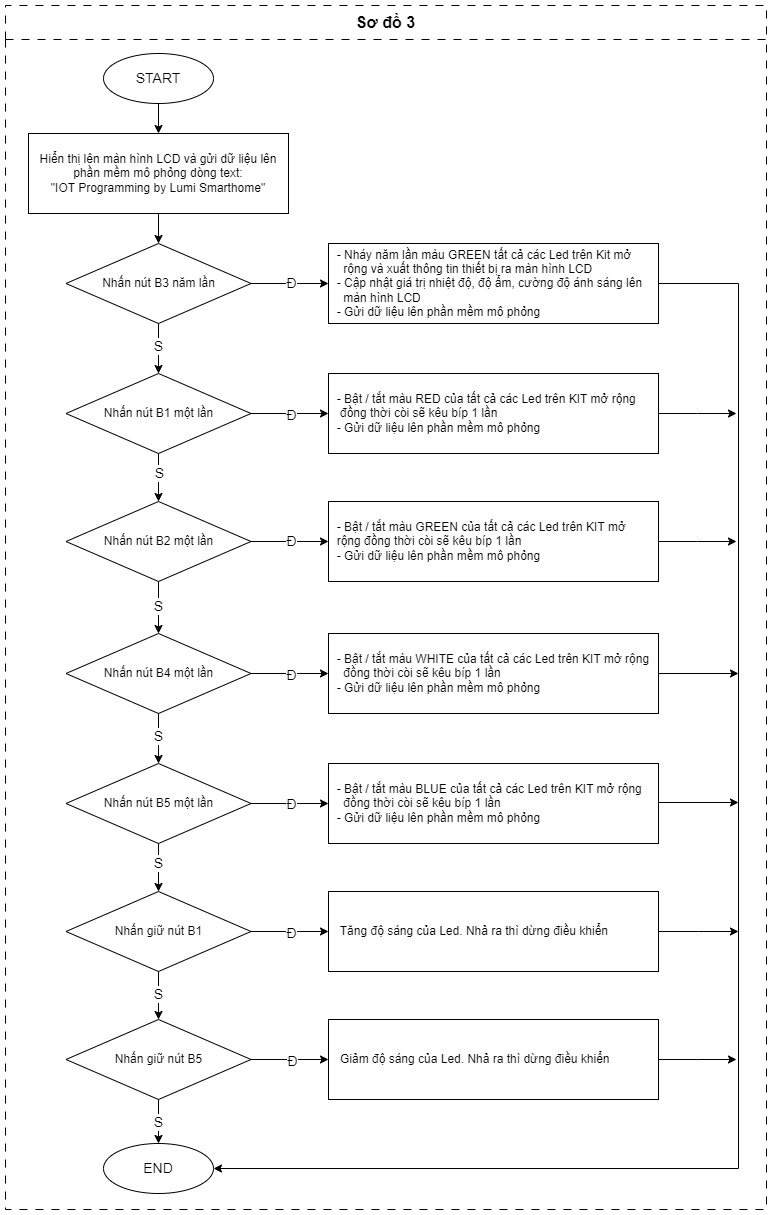
## 26.2. Sơ đồ 1



## 26.3. Sơ đồ 2



## 26.4. Sơ đồ 3



## 26.5. Sơ đồ 4

