



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:	Nguyễn Hoàng Dũng
Mã học viên:	FX19603
Ngày báo cáo:	28/05/2024

HÀ NỘI, 05/2024

MỤC LỤC

1. Danh sách thư viện	5
2. Danh sách các biến toàn cục và các macro	6
3. Danh sách các hàm	10
4. Hàm main ()	11
4.1. Mô tả	11
4.2. Tham số truyền vào.....	12
4.3. Giá trị trả về	12
5. Hàm AppInitManager ()	12
5.1. Mô tả	12
5.2. Tham số truyền vào.....	12
5.3. Giá trị trả về	12
6. Hàm AppStateManager ()	12
6.1. Mô tả	12
6.2. Tham số truyền vào.....	12
6.3. Giá trị trả về	13
7. Hàm SetStateApp ()	13
7.1. Mô tả	13
7.2. Tham số truyền vào.....	13
7.3. Giá trị trả về	13
8. Hàm GetStateApp ()	13
8.1. Mô tả	13
8.2. Tham số truyền vào.....	13
8.3. Giá trị trả về	13
9. Hàm SerialCustom_Init ()	13
9.1. Mô tả	13
9.2. Tham số truyền vào.....	13
9.3. Giá trị trả về	13
10. Hàm USART2_Init ()	13
10.1. Mô tả	13
10.2. Tham số truyền vào.....	13
10.3. Giá trị trả về	13
11. Hàm USART2Modify_IRQHandler ()	13
11.1. Mô tả	13
11.2. Tham số truyền vào.....	13
11.3. Giá trị trả về	14

12. Hàm LoadConfiguration ()	14
12.1. Mô tả	14
12.2. Tham số truyền vào.....	14
12.3. Giá trị trả về	14
13. Hàm DeviceStateMachine ().....	14
13.1. Mô tả	14
13.2. Tham số truyền vào.....	14
13.3. Giá trị trả về	14
14. Hàm Clamp ()	15
14.1. Mô tả	15
14.2. Tham số truyền vào.....	15
14.3. Giá trị trả về	15
15. Hàm Increase_LedLevel ()	15
15.1. Mô tả	15
15.2. Tham số truyền vào.....	15
15.3. Giá trị trả về	15
16. Hàm Decrease_LedLevel ()	15
16.1. Mô tả	15
16.2. Tham số truyền vào.....	15
16.3. Giá trị trả về	15
17. Hàm MultiSensorScan ()	15
17.1. Mô tả	15
17.2. Tham số truyền vào.....	15
17.3. Giá trị trả về	15
18. Hàm Task_MultiSensorScan ().....	16
18.1. Mô tả	16
18.2. Tham số truyền vào.....	16
18.3. Giá trị trả về	16
19. Hàm Serial_SendPacketCustom ()	16
19.1. Mô tả	16
19.2. Tham số truyền vào.....	16
19.3. Giá trị trả về	16
20. Hàm processSerialReceiverCustom ().....	16
20.1. Mô tả	16
20.2. Tham số truyền vào.....	16
20.3. Giá trị trả về	16

21. Hàm PollRxBuff ().....	16
21.1. Mô tả	16
21.2. Tham số truyền vào.....	16
21.3. Giá trị trả về	16
22. Hàm ButtonCmdSetState ()	17
22.1. Mô tả	17
22.2. Tham số truyền vào.....	17
22.3. Giá trị trả về	17
23. Hàm LedCmdSetState ().....	17
23.1. Mô tả	17
23.2. Tham số truyền vào.....	17
23.3. Giá trị trả về	17
24. Hàm BuzzerCmdSetState ().....	17
24.1. Mô tả	17
24.2. Tham số truyền vào.....	17
24.3. Giá trị trả về	17
25. Hàm LcdCmdSetState ().....	17
25.1. Mô tả	17
25.2. Tham số truyền vào.....	17
25.3. Giá trị trả về	18
26. Luồng xử lý code.	18
26.1. Sơ đồ thuật toán tổng quát	18
26.2. Sơ đồ 1	19
26.3. Sơ đồ 2	20
26.4. Sơ đồ 3	21
26.5. Sơ đồ 4	22

NỘI DUNG THUYẾT MINH

1. Danh sách thư viện

STT	Thư viện	Mô tả
1	stdio.h	Thư viện có các hàm nhập xuất: + sprintf: Gửi output đã được định dạng tới một chuỗi
2	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
3	string.h	Thư viện cung cấp các hàm để xử lý chuỗi
4	system_stm32f4xx.h	Thư viện chứa các hàm của hệ thống
5	timer.h	Thư viện cho phép quản lý các tác vụ theo thời gian
6	led.h	Thư viện cung cấp các hàm để sử dụng LED
7	buzzer.h	Thư viện cung cấp các hàm để sử dụng còi
8	melody.h	Thư viện cung cấp các hàm để chọn chế độ kêu của còi
9	lightsensor.h	Thư viện cung cấp các hàm để giao tiếp và lấy giá trị của cảm biến ánh sáng
10	temhumsensor.h	Thư viện cung cấp các hàm để giao tiếp và lấy giá trị của cảm biến nhiệt độ và độ ẩm
11	eventbutton.h	Thư viện cung cấp các hàm để sử dụng các sự kiện của nút nhấn
12	button.h	Thư viện cung cấp các hàm để sử dụng nút nhấn
13	ucg.h	Thư viện cung cấp các hàm để cài đặt thông số và in thông tin ra màn hình LCD
14	Ucglib.h	Thư viện cung cấp hàm để cấu hình các chân giao tiếp SPI của STM32
15	buff.h	Thư viện cung cấp các hàm để sử dụng hàng đợi Queue
16	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 + ...
17	stm32f401re_gpio.h	Thư viện chứa các macro và hàm liên quan tới GPIO: + GPIO_Init(): cấu hình các chân GPIO + ...
18	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 + ...
19	misc.h	Cho phép cấu hình ngắt và tạo ngắt của bộ quản lý ngắt NVIC

2. Danh sách các biến toàn cục và các macro

STT	Biến toàn cục và macro	Mô tả
1	SIZE_QUEUE_DATA_RX	Giá trị kích thước của bộ đệm dữ liệu (FIFO)
2	USART2_GPIO_PORT	Cổng GPIO của USART2
3	USART2_TX_GPIO_PIN	Chân GPIO của đường Tx
4	USART2_RX_GPIO_PIN	Chân GPIO của đường Rx
5	USART2_GPIO_RCC	Clock của cổng GPIO có chức năng là USART2
6	USART2_RCC	Clock của USART2
7	USART_BAUDRATE	Tốc độ truyền của USART2
8	RX_BUFFER_SIZE	Giá trị kích thước mảng lưu dữ liệu được lấy ra từ hàng đợi
9	FRAME_SOF	Giá trị START của khung truyền
10	FRAME_ACK	Giá trị ACK của khung truyền
11	FRAME_NACK	Giá trị NACK của khung truyền
12	CXOR_INIT_VAL	Giá trị khởi tạo của Checkxor

13	CMD_TYPE_GET	Bản tin lấy thông tin, trạng thái của thiết bị từ máy tính (HOST)
14	CMD_TYPE_RES	Bản tin phản hồi thông tin, trạng thái của thiết bị lên máy tính (HOST)
15	CMD_TYPE_SET	Bản tin điều khiển thiết bị từ máy tính (HOST)
16	CMD_OPT	Giá trị của bit Option trong khung truyền
17	CMD_SEQUENCE	Giá trị của bit Sequence trong khung truyền
18	CMD_ID_LED	Giá trị ID của Led trong khung truyền
19	CMD_ID_BUZZER	Giá trị ID của còi trong khung truyền
20	CMD_ID_BUTTON	Giá trị ID của nút nhấn trong khung truyền
21	CMD_ID_TEMP_SENSOR	Giá trị nhiệt độ trong khung truyền
22	CMD_ID_HUMI_SENSOR	Giá trị độ ẩm trong khung truyền
23	CMD_ID_LIGHT_SENSOR	Giá trị cường độ ánh sáng trong khung truyền
24	CMD_ID_LCD	Giá trị ID của LCD trong khung truyền
25	CMD_ID	ID của bản tin
26	CMD_TYPE	Loại bản tin
27	CMD_DATA1	Dữ liệu 1
28	CMD_DATA2	Dữ liệu 2
29	CMD_DATA3	Dữ liệu 3
30	CMD_DATA4	Dữ liệu 4
31	CMD_DATA5	Dữ liệu 5
32	CMD_DATA_LED_ID	Giá trị channel của Led trong khung truyền
33	CMD_DATA_LED_COLOR	Giá trị màu Led trong khung truyền

34	CMD_DATA_LED_NUMBLINK	Giá trị số lần nháy của Led trong khung truyền
35	CMD_DATA_LED_INTERVAL	Giá trị khoảng thời gian nháy Led trong khung truyền
36	CMD_DATA_LED_LASTSTATE	Giá trị trạng thái gần nhất của Led trong khung truyền
37	CMD_DATA_BUZZER_STATE	Giá trị trạng thái của còi trong khung truyền
38	CMD_DATA_BUTTON_EVENT	Giá trị sự kiện của nút nhấn trong khung truyền
39	CMD_DATA_BUTTON_STATE	Giá trị trạng thái của nút nhấn trong khung truyền
40	CMD_DATA_LCD	Giá trị gửi về LCD trong khung truyền
41	CYCLE_LED_CHANGE	Chu kỳ thay đổi độ sáng LED
42	PERIOD_SCAN_MULTISENSOR	Chu kỳ quét của cảm biến
43	event_api_t	Kiểu dữ liệu enum cho các sự kiện của chương trình
44	*event_api_p	Con trỏ kiểu dữ liệu enum trỏ tới các sự kiện của chương trình
45	state_app_t	Kiểu dữ liệu enum cho các trạng thái của chương trình
46	RX_STATE	Trạng thái bộ nhận dữ liệu
47	USART_STATE	Trạng thái của USART
48	g_ucg	Biến để cài đặt các thông số màn hình LCD
49	g_eCurrentState	Biến trạng thái hiện tại của chương trình
50	g_ledRed	Biến thể hiện trạng thái đèn LED, khi nhấn nút B1
51	g_ledGreen	Biến thể hiện trạng thái đèn LED, khi nhấn nút B2
52	g_ledWhite	Biến thể hiện trạng thái đèn LED, khi nhấn nút B4

53	g_ledBlue	Biến thể hiện trạng thái đèn LED, khi nhấn nút B5
54	g_ledChange	Biến dùng để thay đổi trạng thái LED khi nhấn giữ nút B1/B5
55	g_idTimerStartIncrease	Biến lưu giá trị định danh ID của hàm TimerStart khi nhấn giữ nút B1
56	g_idTimerStartDecrease	Biến lưu giá trị định danh ID của hàm TimerStart khi nhấn giữ nút B5
57	g_idTimerDisplayLCD	Biến lưu giá trị định danh ID của hàm TimerStart khi gọi hàm MultiSensorScan
58	g_idTimerSensorUpdate	Biến lưu giá trị định danh ID của hàm TimerStart khi gọi hàm Task_multiSensorScan
59	g_B3Count	Biến đếm số lần nhấn nút B3 khi thao tác trên phần mềm mô phỏng
60	g_temperature	Biến lưu giá trị nhiệt độ được cập nhật từ cảm biến
61	g_humidity	Biến lưu giá trị độ ẩm được cập nhật từ cảm biến
62	g_light	Biến lưu giá trị cường độ ánh sáng được cập nhật từ cảm biến
63	g_RxBufState	Biến lưu trạng thái quá trình nhận dữ liệu của bộ đệm
64	g_IndexRxBuf	Biến lưu vị trí phần tử, của mảng lưu dữ liệu được lấy ra từ hàng đợi
65	g_byCheckXorRxBuf	Biến lưu giá trị checkxor của khung truyền
66	g_strRxBufData	Mảng lưu dữ liệu được truyền vào FIFO (dữ liệu đầu vào)
67	g_serialQueueRx	Con trỏ để trỏ đến đối tượng FIFO
68	g_strRxBuffer	Mảng lưu dữ liệu được lấy ra từ hàng đợi
69	g_Data_Receive	Biến lưu dữ liệu nhận được gần nhất của ngoại vi USART
70	g_strTemp[30]	Mảng lưu giá trị nhiệt độ

71	g_strHumi[30]	Mảng lưu giá trị độ ẩm
72	g_strLight[30]	Mảng lưu giá trị cường độ ánh sáng

3. Danh sách các hàm

STT	Danh sách hàm	Mô tả
1	int main ()	Hàm chính của chương trình
2	void AppInitManager (void);	Hàm khởi tạo các tài nguyên cần sử dụng
3	void AppStateManager (uint8_t event);	Hàm xử lý các sự kiện của chương trình chính
4	void SetStateApp (state_app_t state);	Hàm SET trạng thái của chương trình
5	state_app_t GetStateApp (void);	Hàm lấy trạng thái hiện tại của chương trình
6	void SerialCustom_Init (void);	Hàm khởi tạo bộ đệm FIFO và USART2
7	void USART2_Init (void);	Hàm cấu hình giao tiếp USART2
8	void USART2Modify_IRQHandler (void);	Hàm thực thi ngắt USART2
9	void LoadConfiguration (void);	Hàm hiển thị thông tin lên LCD và PC_Simulator_KIT khi bắt đầu cắm nguồn cho thiết bị
10	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
11	uint8_t Clamp (uint8_t value, uint8_t min , uint8_t max);	Hàm tạo khoảng giá trị
12	void Increase_LedLevel (void);	Hàm tăng độ sáng đèn Led
13	void Decrease_LedLevel (void);	Hàm giảm độ sáng đèn Led
14	void MultiSensorScan (void);	Hàm cập nhật giá trị nhiệt độ, độ ẩm, cường độ ánh sáng

15	<code>void Task_multiSensorScan (void);</code>	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
16	<code>void Serial_SendPacketCustom (uint8_t byOption, uint8_t byCmdId, uint8_t byCmdType, uint8_t *pPayload, uint8_t byLengthPayload);</code>	Hàm gửi đoạn text lên PC_Simulator_KIT
17	<code>void processSerialReceiverCustom (void);</code>	Hàm kiểm tra và xử lý dữ liệu nhận về từ PC_Simulator_KIT
18	<code>uint8_t PollRxBuff (void);</code>	Hàm xử lý các bản tin nhận được theo định dạng
19	<code>void ButtonCmdSetState (uint8_t button_event, uint8_t button_state);</code>	Hàm điều khiển nút nhấn khi nhận được bản tin từ PC_Simulator_KIT
20	<code>void LedCmdSetState (uint8_t led_id, uint8_t led_color, uint8_t led_num_blink, uint8_t led_interval, uint8_t led_last_state);</code>	Hàm điều khiển bật/tắt led RGB trên KIT khi nhận được bản tin từ PC_Simulator_KIT
21	<code>void BuzzerCmdSetState (uint8_t buzzer_state);</code>	Hàm điều khiển bật/tắt còi khi nhận được bản tin từ PC_Simulator_KIT
22	<code>void LcdCmdSetState (char *text);</code>	Hàm hiển thị đoạn text lên màn hình LCD khi nhận được bản tin từ PC_Simulator_KIT

4. 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

5. 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

6. 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

7. 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

8. 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

9. 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

10. 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

11. 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

12. 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

13. 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

14. 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

15. 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

16. 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

17. 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

18. 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

19. 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

20. 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

21. 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

22. 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

23. 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

24. 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

25. 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

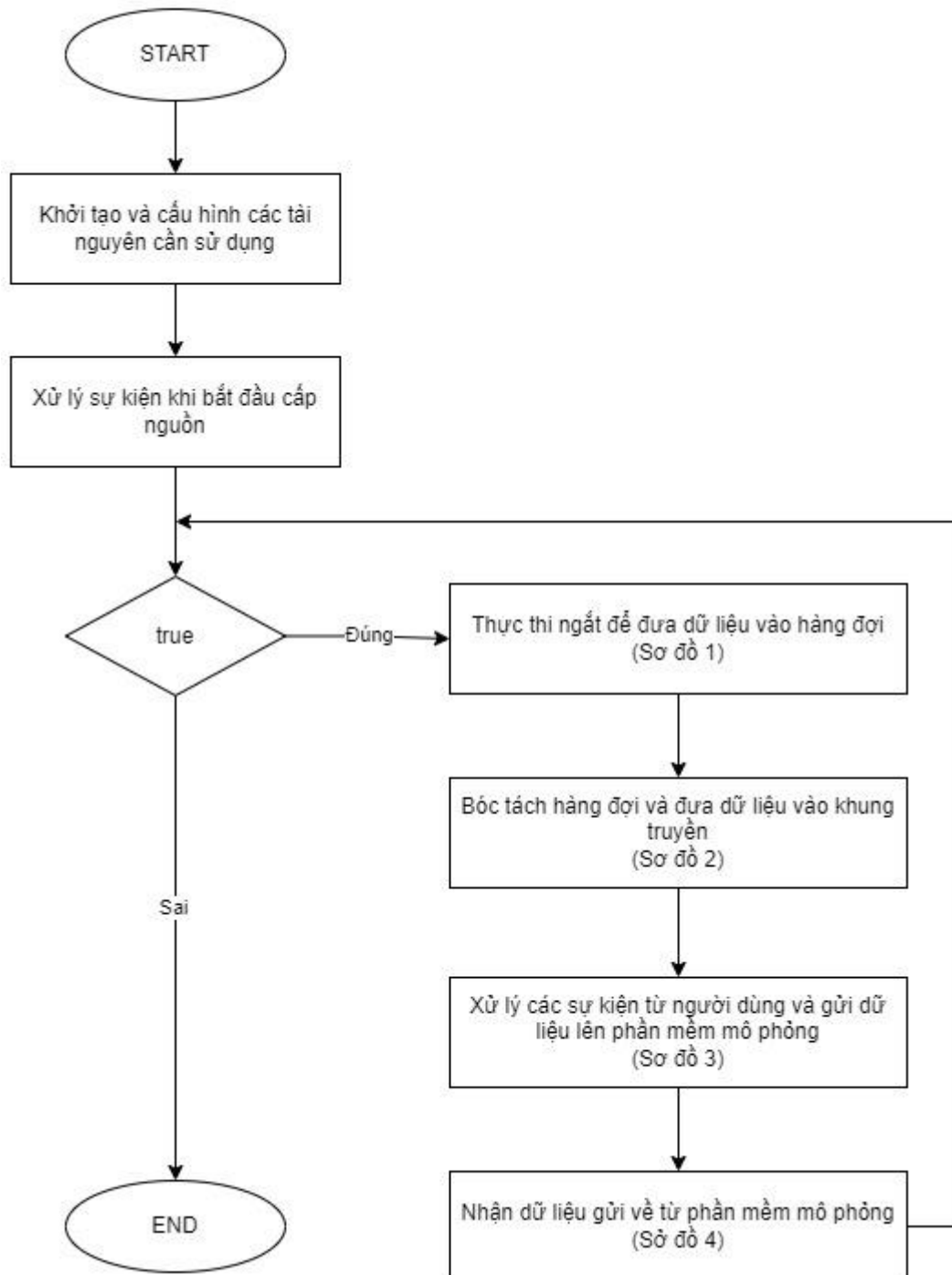
Con trỏ text (kiểu dữ liệu char): Đoạn text cần hiển thị

25.3. Giá trị trả về

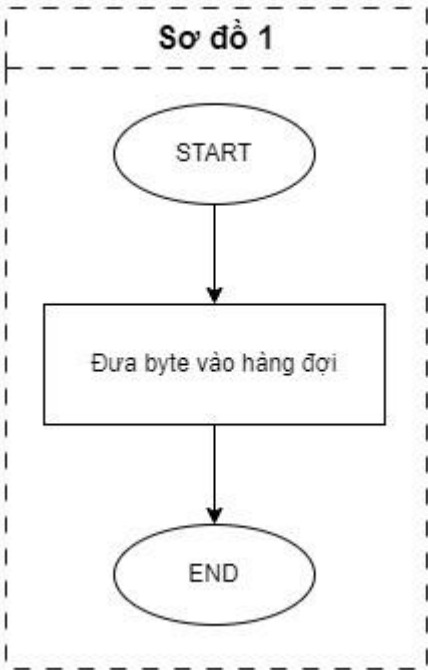
Không có

26. Luồng xử lý code.

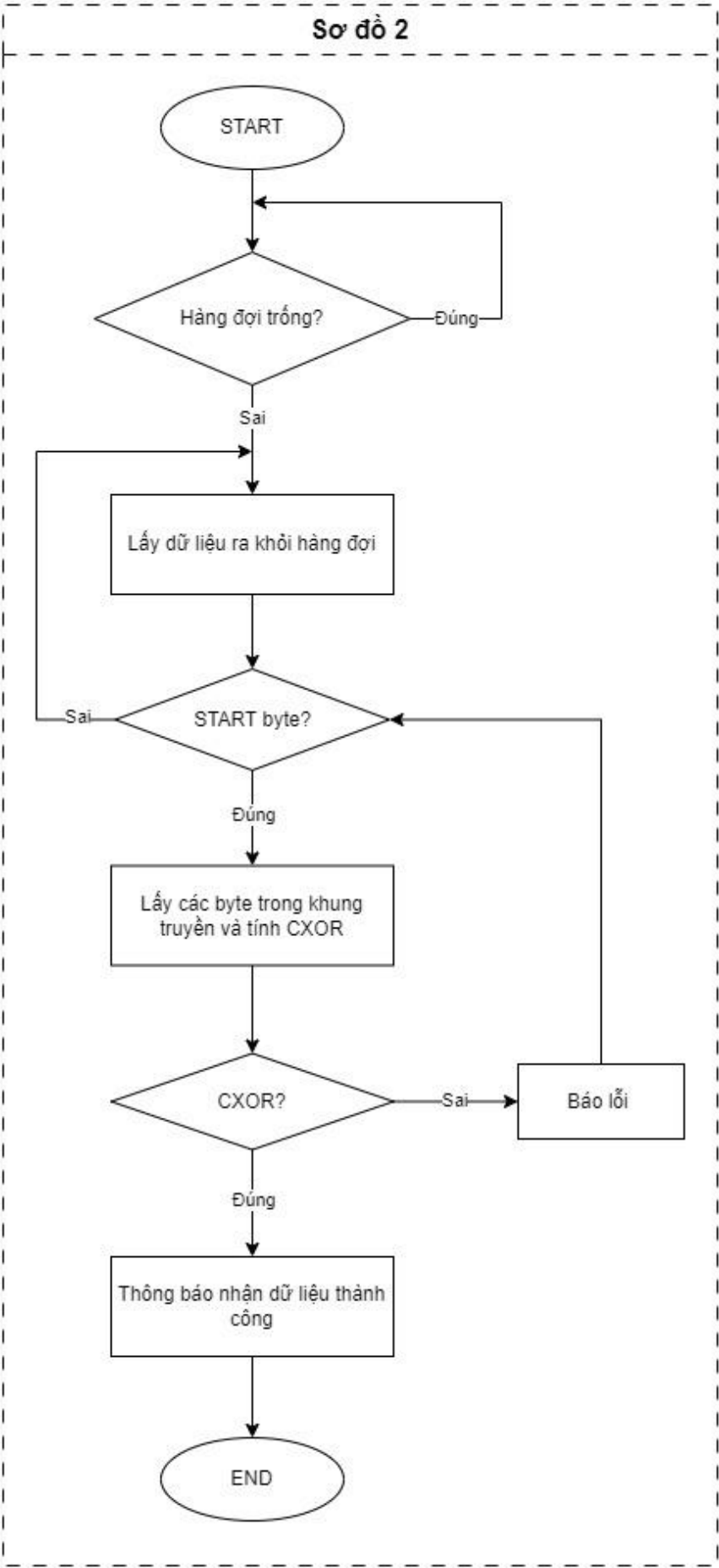
26.1. Sơ đồ thuật toán tổng quát



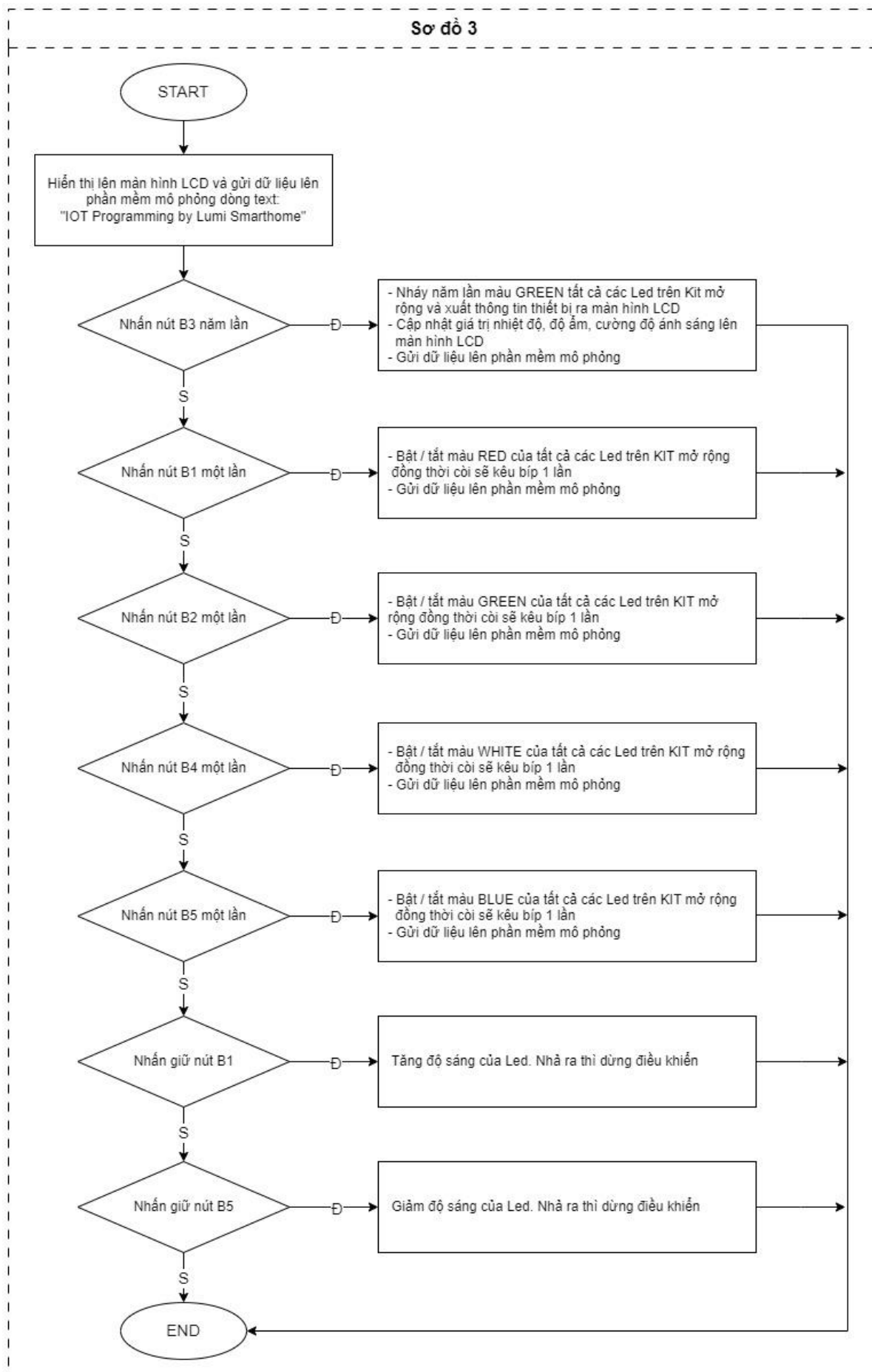
26.2. Sơ đồ 1



26.3. Sơ đồ 2



26.4. Sơ đồ 3



26.5. Sơ đồ 4

