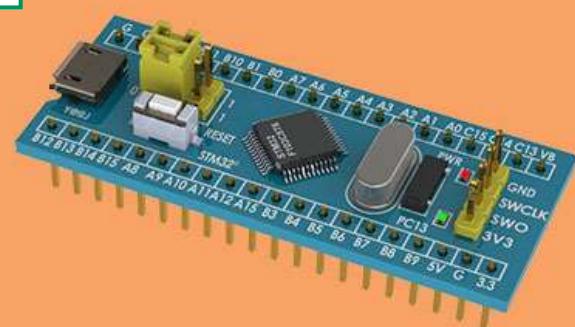


LẬP TRÌNH STM32

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231

POSTED ON 17/07/2020 BY KHUÊ NGUYỄN

17
Th7**Khuê Nguyễn Creator**

Bài 10: Giao tiếp I2C là gì, giao tiếp với module RTC DS3231

Giao thức I2C là một chuẩn truyền thông đồng bộ trong vi xử lý. Giao thức này được sử dụng rất nhiều trong thiết kế vì tính đơn giản và dễ dàng làm việc. Tuy nhiên bù lại tốc độ thường không cao và chỉ sử dụng trong mạch (on board).

Bài 10 trong serie **Học lập trình STM32 từ A tới Z**



Mục Lục

1. Giao thức I2C là gì
2. Kết nối vật lý của giao thức I2C
3. Cách truyền dữ liệu của giao thức I2C
 - 3.1. Start condition(Điều kiện bắt đầu)
 - 3.2. Bit Read/Write
 - 3.3. Bit ACK / NACK
 - 3.4. Khối dữ liệu
 - 3.5. Điều kiện kết thúc (Stop condition)
4. Giới thiệu chip thời gian thực DS3231
5. Cấu hình giao thức I2C trên STM32 CubeMX
6. Lập trình giao thức I2C
 - 6.1. Kết
 - 6.2. Related posts:

Giao thức I2C là gì

I2C viết tắt của Inter- Integrated Circuit là một phương thức giao tiếp được phát triển bởi hãng Philips Semiconductors. Dùng để truyền tín hiệu giữa vi xử lý và các IC trên các bus nối tiếp.

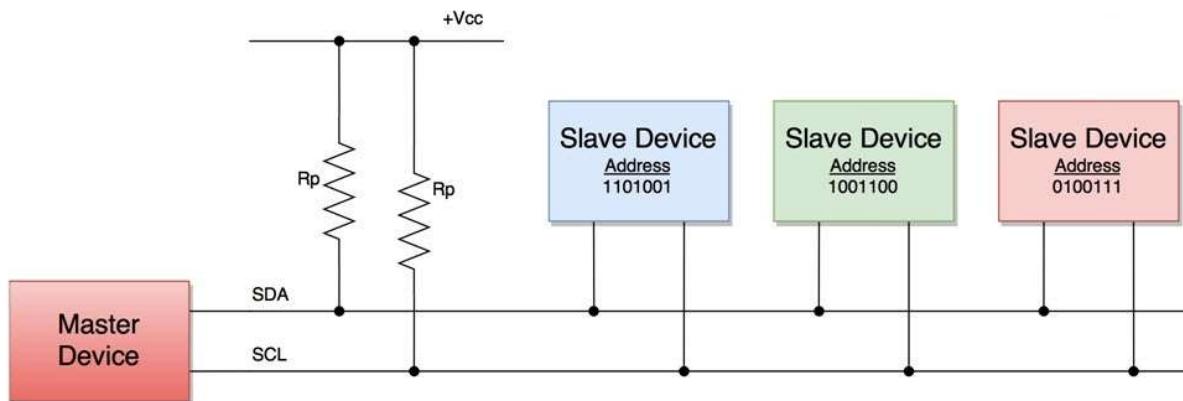
Đặc điểm:

- Tốc độ không cao
- Thường sử dụng onboard với đường truyền ngắn
- Nối được nhiều thiết bị trên cùng một bus
- Giao tiếp đồng bộ, sử dụng Clock từ master
- Sử dụng 7 bit hoặc 10 bit địa chỉ
- Chỉ sử dụng 2 chân tín hiệu SDA, SCL
- Có 2 tốc độ tiêu chuẩn là Standard mode (100 kb/s) và Low mode (10 kbit/s)

Kết nối vật lý của giao thức I2C

Bus I2C sử dụng 2 dây tín hiệu là SDA (Serial Data Line) và SCL (Serial Clock Line). Dữ liệu truyền trên SDA được đồng bộ với mỗi xung SCL. Đường SCL chỉ master mới có quyền điều khiển.

Tất cả các thiết bị đều dùng chung 2 đường tín hiệu này.



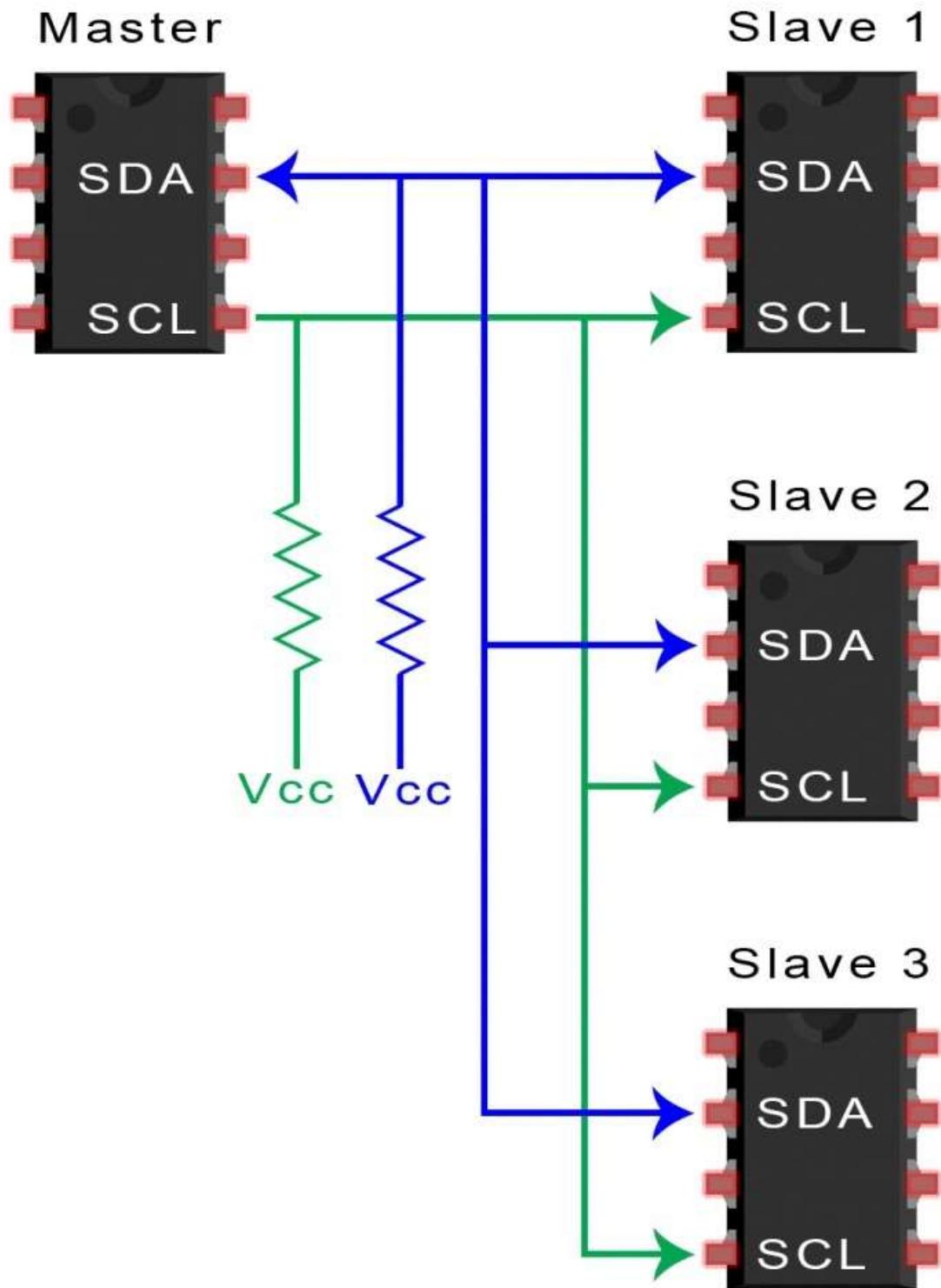
Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 58

Hai đường bus SDA và SCL hoạt động ở chế độ Open Drain hay cực máng hở. Nghĩa là tất cả các thiết bị trong mạng đều chỉ có thể lái 2 chân này về 0 chứ ko thể kéo lên 1. Để tránh việc xảy ra ngắn mạch khi thiết bị này kéo lên cao, thiết bị kia kéo xuống thấp.

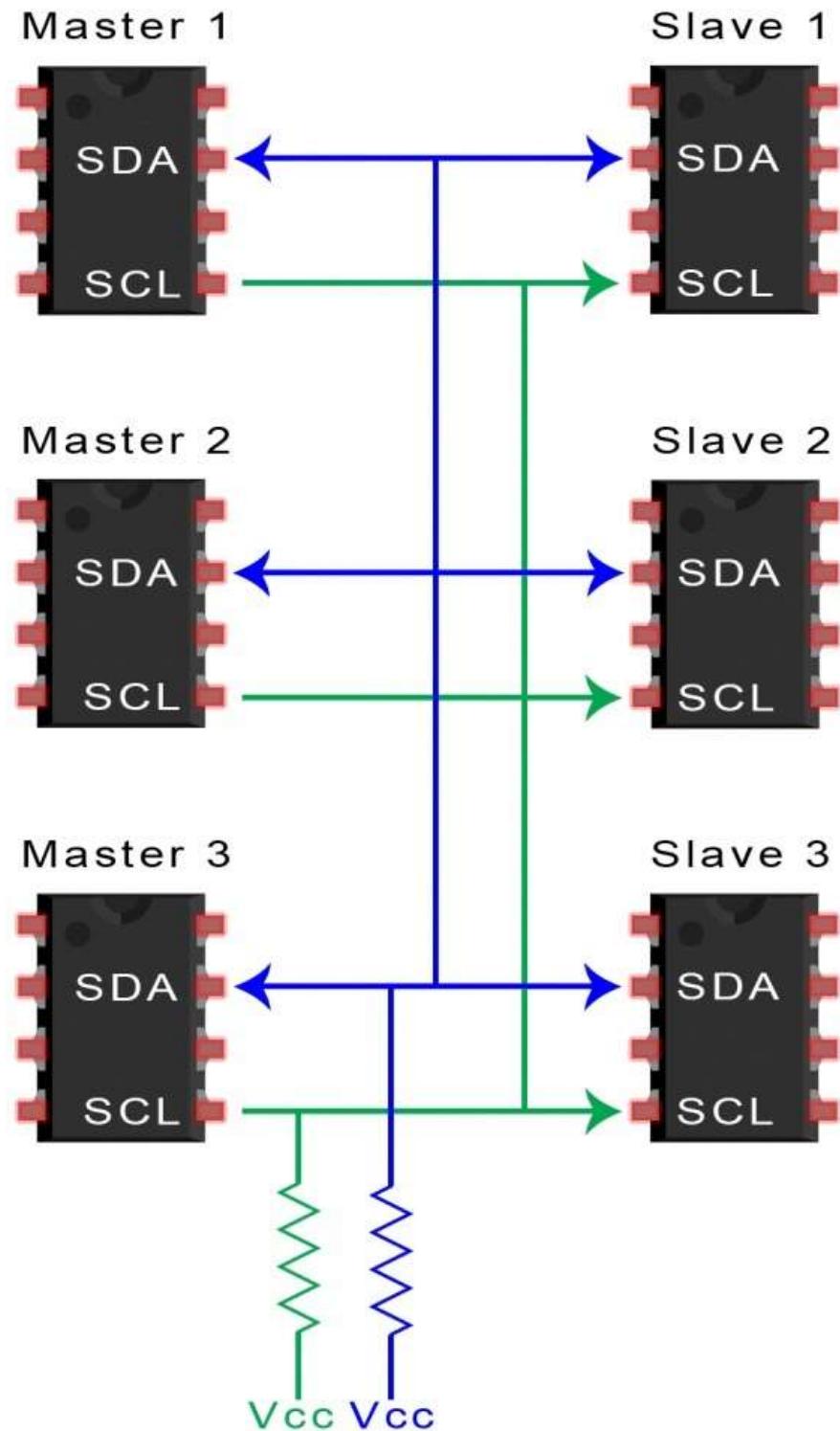
Để giữ mức logic là 1 ở trạng thái mặc định phải mắc thêm 2 điện trở treo lên Vcc (thường từ 1k – 4k7).

Mỗi Bus I2C sẽ có 3 chế độ chính:

- Một Master, nhiều Slave
- Nhiều master, nhiều Slave
- Một Master, một Slave



Một master nhiều Slave



Nhiều Master, nhiều Slave

Tại một thời điểm truyền nhận dữ liệu chỉ có một Master được hoạt động, điều khiển dây SCL và phát tín hiệu bắt đầu tới các Slave.

Tất cả các thiết bị đáp ứng sự điều hướng của Master gọi là Slave. Giữa các Slave với nhau, phân biệt bằng 7bit địa chỉ.

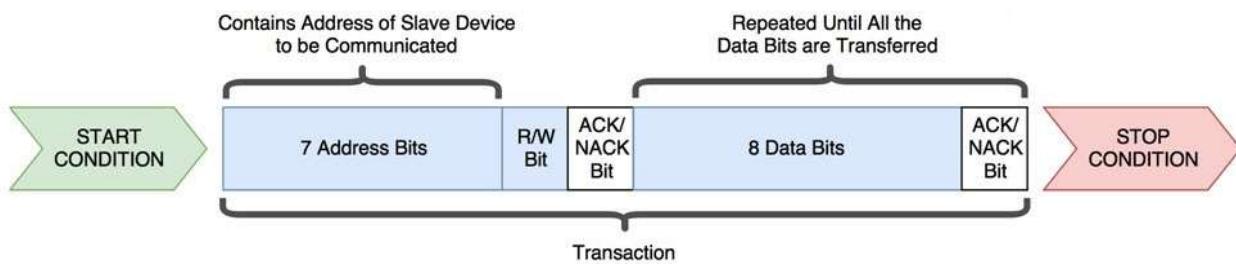
Cách truyền dữ liệu của giao thức I2C

Giao thức (phương thức giao tiếp) là cách các thiết bị đã thống nhất với nhau khi sử dụng một chuẩn nào đó để truyền và nhận tín hiệu.

Dữ liệu được truyền đi trên dây SDA được thực hiện như sau:

1. Master thực hiện điều kiện bắt đầu I2C (Start Condition)
2. Gửi địa chỉ 7 bit + 1bit Đọc/Ghi (R/W) để giao tiếp muốn đọc hoặc ghi dữ liệu tại Slave có địa chỉ trên
3. Nhận phải hồi từ Bus, nếu có một bit ACK (Kéo SDA xuống thấp) Master sẽ gửi dữ liệu
4. Nếu là đọc dữ liệu R/W bit = 1, chân SDA của master sẽ là input, đọc dữ liệu từ Slave gửi về. Nếu là ghi dữ liệu R/W = 0, chân SDA sẽ là output ghi dữ liệu vào Slave
5. Truyền điều khiển kết thúc (Stop Condition)

Mỗi lần giao tiếp có cấu trúc như sau:



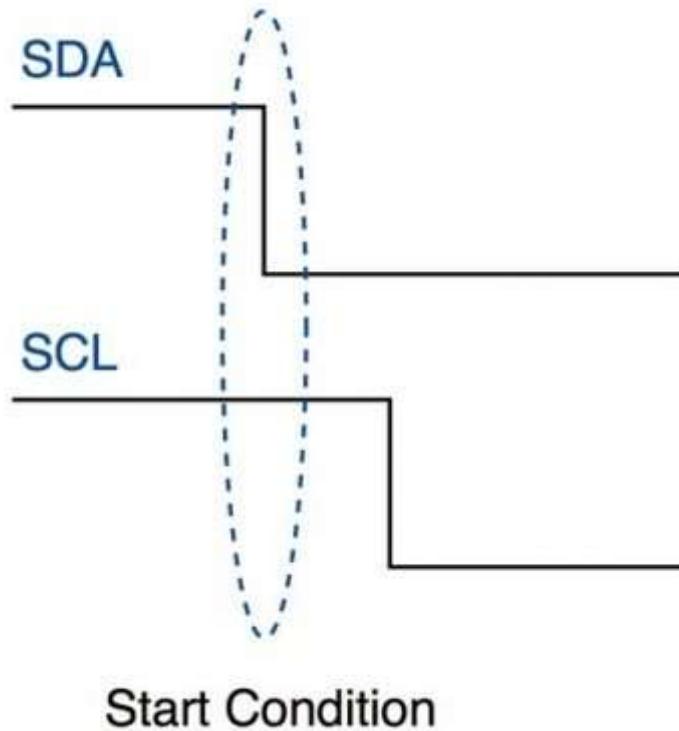
Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 59

Start condition(Điều kiện bắt đầu)

Bất cứ khi nào một thiết bị chủ / IC quyết định bắt đầu một giao dịch, nó sẽ chuyển mạch SDA từ mức điện áp cao xuống mức điện áp thấp trước khi đường

SCL chuyển từ cao xuống thấp.

Khi điều kiện bắt đầu được gửi bởi thiết bị Master, tất cả các thiết bị Slave đều hoạt động ngay cả khi chúng ở chế độ ngủ (sleep mode) và đợi bit địa chỉ.



Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 60

Bit Read/Write

Bit này xác định hướng truyền dữ liệu. Nếu thiết bị Master / IC cần gửi dữ liệu đến thiết bị Slave, bit này được thiết lập là '0'. Nếu IC Master cần nhận dữ liệu từ thiết bị Slave, bit này được thiết lập là '1'.

Bit ACK / NACK

ACK / NACK là viết tắt của Acknowledged/Not-Acknowledged. Nếu địa chỉ vật lý của bất kỳ thiết bị Slave nào trùng với địa chỉ được thiết bị Master phát, giá trị của bit này được set là '0' bởi thiết bị Slave. Ngược lại, nó vẫn ở mức logic '1' (mặc định).

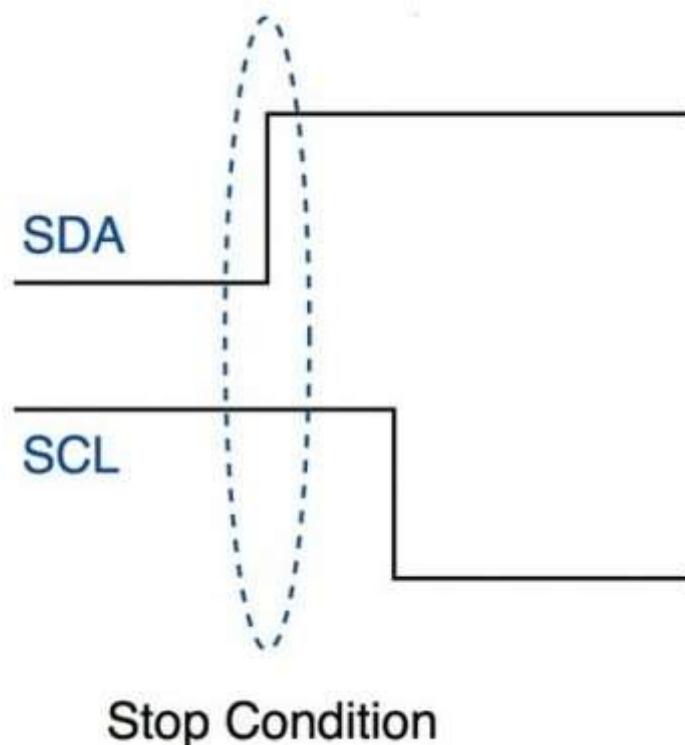
Khối dữ liệu

Nó bao gồm 8 bit và chúng được thiết lập bởi bên gửi, với các bit dữ liệu cần truyền tới bên nhận. Khối này được sau bởi một bit ACK / NACK và được set thành '0' bởi bên nhận nếu nó nhận thành công dữ liệu. Ngược lại, nó vẫn ở mức logic '1'.

Sự kết hợp của khối dữ liệu theo sau bởi bit ACK / NACK được lặp lại cho đến quá trình truyền dữ liệu được hoàn tất.

Điều kiện kết thúc (Stop condition)

Sau khi các khung dữ liệu cần thiết được truyền qua đường SDA, thiết bị Master chuyển đường SDA từ mức điện áp thấp sang mức điện áp cao trước khi đường SCL chuyển từ cao xuống thấp.



Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 61

Giới thiệu chip thời gian thực DS3231

DS3231 là chip thời gian thực, giao tiếp thông qua giao thức I2C. Làm việc tại dải điện áp từ 2.3 đến 5.5V, tích hợp sẵn thạch anh nội nên rất nhỏ gọn.

Có 2 chế độ hẹn giờ có thể Config từng giây tới ngày trong tháng.

Datasheet các bạn down tại đây:

<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

DS3231 có địa chỉ 7bit là 0x68. Cách đọc và truyền được mô tả như trong hình

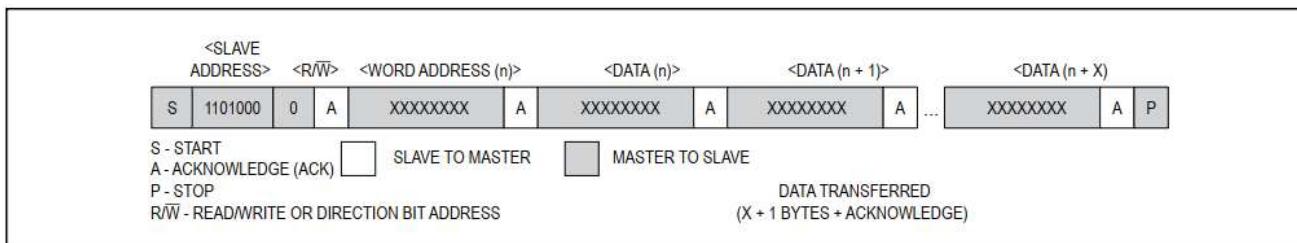


Figure 3. Data Write—Slave Receiver Mode

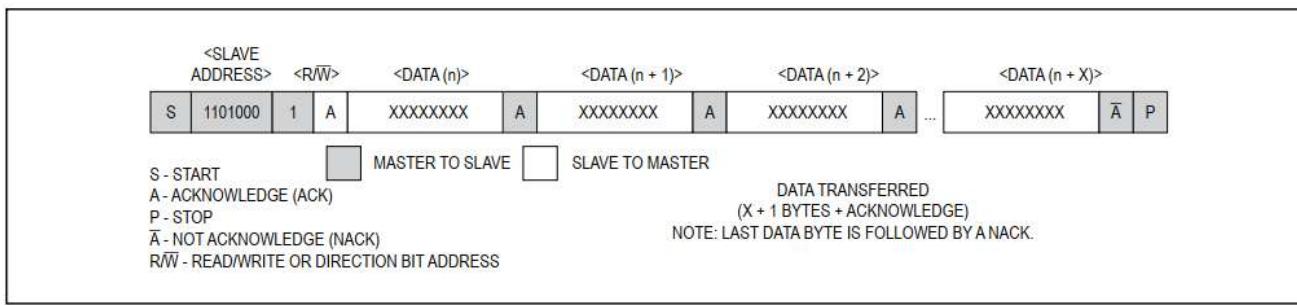


Figure 4. Data Read—Slave Transmitter Mode

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 62

Bảng sau mô tả địa chỉ lưu các giá trị ngày tháng năm đó là từ 0x00 tới 0x06.

Các byte từ 0x07 tới 0x0D lưu giá trị Hẹn giờ A1M và A2M

Các byte từ 0x0E tới 0x12 là các thanh ghi điều khiển DS3231

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds				Seconds				Seconds 00–59
01h	0	10 Minutes				Minutes				Minutes 00–59
02h	0	12/24	AM/PM 20 Hour	10 Hour	Hour				Hours 1–12 + AM/PM 00–23	00–23
03h	0	0			0	0	0	Day		
04h	0	0	10 Date			Date				Date 01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds				Seconds				Alarm 1 Seconds 00–59
08h	A1M2	10 Minutes				Minutes				Alarm 1 Minutes 00–59
09h	A1M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM 00–23
0Ah	A1M4	DY/DT			Day				Alarm 1 Day	1–7
0Bh	A2M2	10 Minutes				Minutes				Alarm 2 Minutes 00–59
0Ch	A2M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 2 Hours	1–12 + AM/PM 00–23
0Dh	A2M4	DY/DT			Day				Alarm 2 Day	1–7
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
0Fh	OSF	0	0	0	EN32KHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 63

Cấu hình giao thức I2C trên STM32 CubeMX

Mở phần mềm, chọn chip STM32F103C8 nhấn start project.

Trong Sys chọn Debug : Serial Wire. Nêu rõ trong **Bài 3**

Trong Tab Connectivity: Chọn giao thức I2C1

Mode: I2C

Parameter: để mặc định với Speed mode là standard, Clock speed 100khz.

Trong NVIC tick chọn bật ngắt cho I2C event.

I2C1 Mode and Configuration

Mode

I2C

Configuration

Reset Configuration

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

Master Features

I2C Speed Mode	Standard Mode
* I2C Clock Speed (Hz)	100000

Slave Features

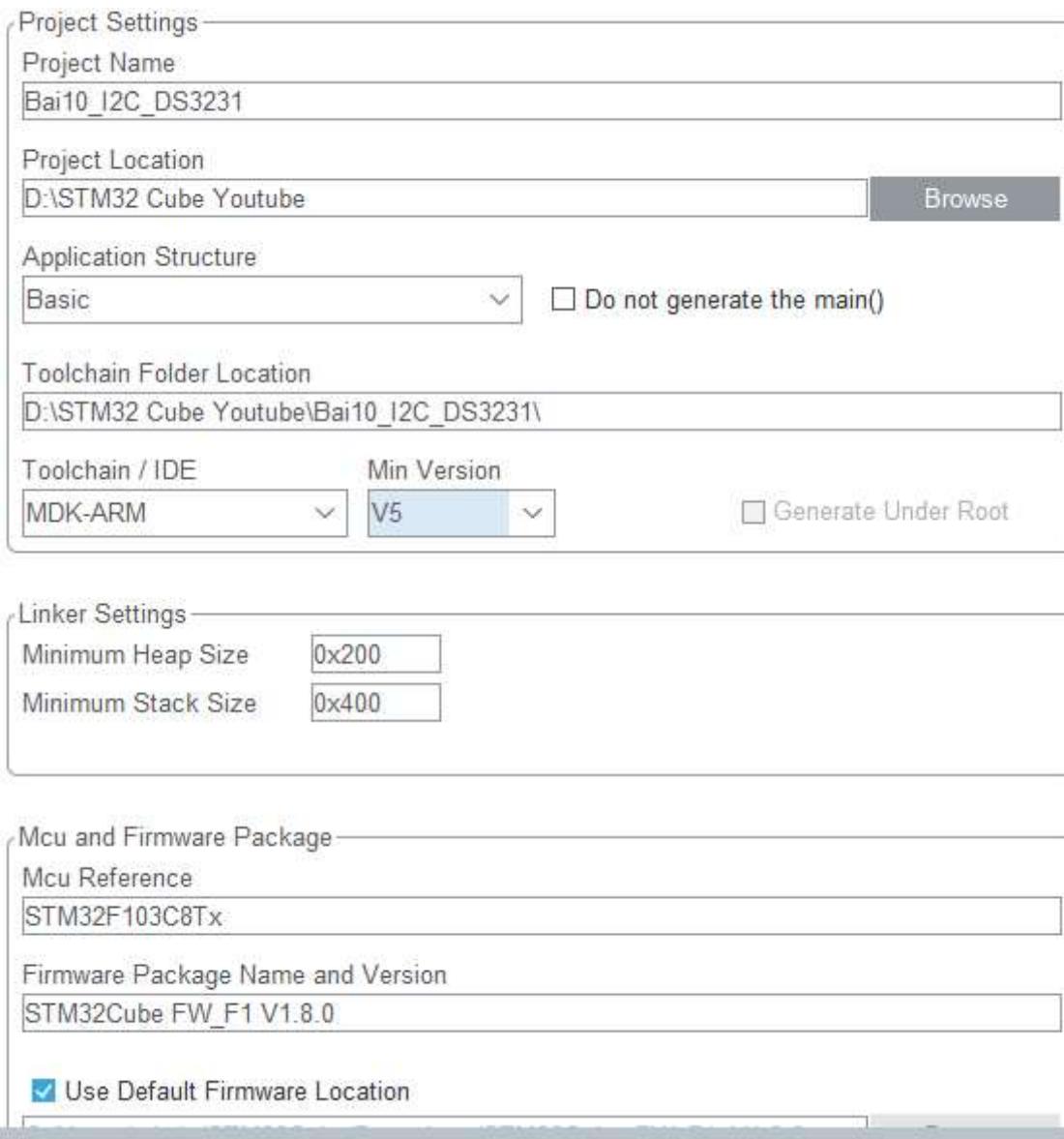
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 64

Config thêm PC13 là Led để kiểm tra trạng thái I2C

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 65

Đặt tên project rồi Gencode



Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 66

Lập trình giao thức I2C

Trong KeilC chúng ta cấu hình như sau.

Khai báo địa chỉ của DS3231 là 0x68<<1. Các bạn phải dịch trái 1 đơn vị, vì các hàm i2c của hal yêu cầu vậy, bit thứ 8 là R/W sẽ được thêm vào ngay sau giá trị địa chỉ đó.

Khai báo 2 nguyên mẫu hàm là BCD2DEC và DEC2BCD. Hai hàm này có tác dụng chuyển đổi dữ liệu kiểu BCD thành kiểu số thập phân(hay hexa, bin) và ngược lại. Vì dữ liệu đọc ghi vào Ds3231 là kiểu BCD

Hai mảng lưu dữ liệu truyền và nhận từ BCD gồm 7 byte

Một biến Status để đọc phản hồi của giao thức I2C

Định nghĩa biến dữ liệu kiểu Struct để lưu các giá trị thời gian, sau đó khởi tạo 2 biến TimeNow để đọc giá trị thời gian và Time Set để cài đặt thời gian lên DS3231

```

22 #include "main.h"
23
24 /* Private includes ----- */
25 /* USER CODE BEGIN Includes */
26 #define DS3231_ADDRESS 0x68<<1
27 uint8_t BCD2DEC(uint8_t data);
28 uint8_t DEC2BCD(uint8_t data);
29 uint8_t u8_revBuffer[7];
30 uint8_t u8_tranBuffer[7];
31 uint8_t status = 0;
32 typedef struct
33 {
34     uint8_t hours;
35     uint8_t min;
36     uint8_t sec;
37     uint8_t date;
38     uint8_t day;
39     uint8_t month;
40     uint8_t year;
41 }DS3231_typedef;
42
43 DS3231_typedef DS3231_TimeNow;
44 DS3231_typedef DS3231_TimeSet;
45
46 /* USER CODE END Includes */

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 67

Sau đó viết hàm thực thi chuyển đổi BCD2DEC và ngược lại

```

45
46 uint8_t BCD2DEC(uint8_t data)
47 {
48     return (data>>4)*10+ (data&0x0f);
49 }
50 uint8_t DEC2BCD(uint8_t data)
51 {
52     return (data/10)<<4|(data%10);
53 }

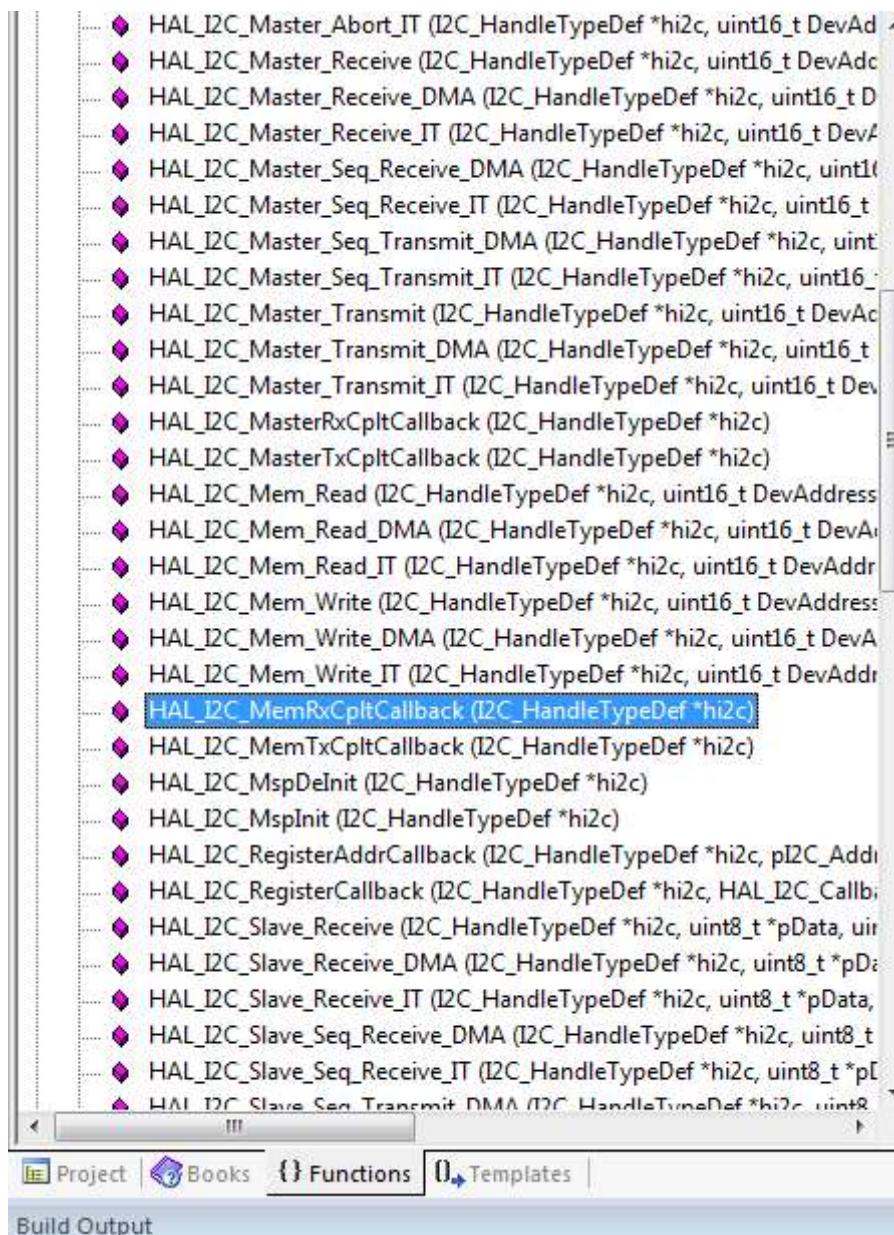
```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 68

Các bạn tìm hàm Call back của giao thức I2C bằng cách tìm từ stm32f1xx_it.c như các bài trước, hoặc sử dụng list functions như sau.

Mở tab Function ở phía work space project, mở file I2C và tìm tới hàm

HAL_I2C_MemRxCpltCallback(I2C_HandleTypeDef *hi2c)



Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 69

Copy phần thực thi của hàm đó và dán vào phần tiền xử lý trên main(). Sau đó thêm các bước sau.

Điều hướng với lệnh if(hi2c->Instance == I2C1) nếu I2C1 sảy ra ngắt

Lưu các giá trị thời gian vào biến DS3231_TimeNow, với byte đầu tiên là giây, phút, giờ Sử dụng hàm chuyển đổi BCD2DEC

```

86  /* Private user code -----*/
87  /* USER CODE BEGIN 0 */
88 void HAL_I2C_MemRxCpltCallback(I2C_HandleTypeDef *hi2c)
89 {
90     /* Prevent unused argument(s) compilation warning */
91     UNUSED(hi2c);
92     if(hi2c->Instance == I2C1)
93     {
94         DS3231_TimeNow.sec = BCD2DEC(u8_revBuffer[0]);
95         DS3231_TimeNow.min = BCD2DEC(u8_revBuffer[1]);
96         DS3231_TimeNow.hours = BCD2DEC(u8_revBuffer[2]);
97
98         DS3231_TimeNow.day = BCD2DEC(u8_revBuffer[3]);
99         DS3231_TimeNow.date = BCD2DEC(u8_revBuffer[4]);
100        DS3231_TimeNow.month = BCD2DEC(u8_revBuffer[5]);
101        DS3231_TimeNow.year = BCD2DEC(u8_revBuffer[6]);
102    }
103    /* NOTE : This function should not be modified, when the callback is needed,
104             the HAL_I2C_MemRxCpltCallback can be implemented in the user file
105             */
106}

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 70

Trong phần main(), đầu tiên chúng ta quét thử xem có thiết bị i2c nào kết nối không bằng cách.

Tắt Led PC13.

Cho hàm HAL_I2C_IsDeviceReady vào vòng lặp for, khi có kết nối, sẽ lưu địa chỉ của Slave đó vào Status và bật Led PC13

```

132
133     /* Initialize all configured peripherals */
134     MX_GPIO_Init();
135     MX_I2C1_Init();
136     /* USER CODE BEGIN 2 */
137     HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_SET);
138     for(uint8_t i=0; i<255; i++)
139     {
140         if(HAL_I2C_IsDeviceReady(&hi2c1,i,1,10) == HAL_OK)
141         {
142             status = i;
143             HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_RESET);
144         }
145     }
146

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 71

Trước While(1) chúng ta sẽ ghi giá trị thời gian cho DS3231, hiện tại là 14h42 ngày 17 tháng 7 năm 2020. Ta sẽ ghi như sau. Sau đó ghi vào I2C bằng hàm

HAL_I2C_Mem_Write_IT(&hi2c1,DS3231_ADDRESS,0x00,I2C_MEMADD_SIZE_8BIT,u8_tranBuffer,7);

Với 0x00 là địa chỉ bắt đầu,

I2C_MEMADD_SIZE_8BIT: kiểu dữ liệu bộ nhớ là 8bit

Gửi 7 byte: từ 0x00 → 0x06

```

139
140     /* Initialize all configured peripherals */
141     MX_GPIO_Init();
142     MX_I2C1_Init();
143     /* USER CODE BEGIN 2 */
144     u8_tranBuffer[0]= DEC2BCD(00);
145     u8_tranBuffer[1]= DEC2BCD(42);
146     u8_tranBuffer[2]= DEC2BCD(14);
147
148     u8_tranBuffer[3]= DEC2BCD(7); // ngày
149     u8_tranBuffer[4]= DEC2BCD(17);
150     u8_tranBuffer[5]= DEC2BCD(7);
151     u8_tranBuffer[6]= DEC2BCD(20);
152     HAL_I2C_Mem_Write_IT(&hi2c1,DS3231_ADDRESS,0x00,I2C_MEMADD_SIZE_8BIT,u8_tranBuffer,7);
153     /* USER CODE END 2 */
154
155     /* Infinite loop */

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 72

Trong While(1) ta sẽ đọc dữ liệu từ DS3231 mỗi 5s một lần, ta làm như sau

```

155     /* Infinite loop */
156     /* USER CODE BEGIN WHILE */
157     while (1)
158     {
159         /* USER CODE END WHILE */
160
161         /* USER CODE BEGIN 3 */
162         HAL_I2C_Mem_Read_IT(&hi2c1,DS3231_ADDRESS,0x00,I2C_MEMADD_SIZE_8BIT,u8_tranBuffer,7);
163         HAL_Delay(5000);
164     }
165     /* USER CODE END 3 */

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 73

Nhấn Build F7 và nạp chương trình. Kết nối DS3231 SDA với PB7, SCL với PB6, sau đó nhấn vào debug .

Trong debug Add 2 biến Time_now và Status vào Watch 1

The screenshot shows the Keil MDK-ARM IDE interface. On the left is the Disassembly window, which contains assembly code. On the right is the Project Explorer window, showing files like main.c, stm32f1xx_hal_i2c.c, and startup_stm32f103xb.s. The main editor window displays C code for a RTC module. A context menu is open over the line 'DS3231_TypeDef DS3231_TimeNow;'. The menu items include:

- Split Window horizontally
- Insert '#include file'
- Go to Headerfile "main.h"
- Go To Definition Of 'DS3231_TimeNow'
- Go To Reference To 'DS3231_TimeNow'
- Add 'DS3231_TimeNow' to... (highlighted)
- Insert Tracepoint at 'DS3231_TimeNow'...
- Enable/Disable Tracepoint
- Insert/Remove Bookmark Ctrl+F2
- Undo Ctrl+Z
- Redo Ctrl+Y
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Select All Ctrl+A
- Execution Profiling
- Outlining
- Advanced

The 'Add 'DS3231_TimeNow' to...' option is highlighted in yellow.

```

Disassembly
118:           status = 1;
0x08001B78 FBDF8104 LDR.W   r8,[pc,#260] ; @0
119:           HAL_GPIO_WritePi
120:           }
121:           }

main.c
58  uint8_t u8_revBuffer[7];
59  uint8_t u8_tranBuffer[7];
60  uint8_t status = 0;
61  typedef struct
62  {
63      uint8_t hours;
64      uint8_t min;
65      uint8_t sec;
66      uint8_t date;
67      uint8_t day;
68      uint8_t month;
69      uint8_t year;
70  }DS3231_TypeDef;
71  DS3231_TypeDef DS3231_TimeNow;
72  DS3231_TypeDef DS3231_TimeSet;
73  /* USER CODE END PFP */
74  /* USER CODE BEGIN PFP */

```

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 74

Ta thấy rằng Status hiển thị 0xD1 nghĩa là 11010001 chính là địa chỉ 0x68 <<1 + R/W bit, với R/W = 1 nghĩa là đọc dữ liệu

Thời gian sẽ hiển thị đúng với giá trị ta nạp vào và sẽ thay đổi mỗi 5S một lần.

Name	Value	Type
status	0xD1 'Ñ'	unsigned...
DS3231_TimeNow	0x200000...	struct < u...
hours	14	unsigned...
min	42 '*'	unsigned...
sec	9	unsigned...
date	17	unsigned...
day	7	unsigned...
month	7	unsigned...
year	20	unsigned...
<Enter expression>		

Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231 75

Kết

Giao thức I2C rất dễ sử dụng, vì các slave đều phản hồi lại master trước khi truyền, đó là các bit Ack Nack, vậy nên việc truyền sai địa chỉ, truyền sai dữ liệu rất dễ phát hiện.

Hi vọng bạn đã nắm rõ cách thức hoạt động của giao thức I2C, hãy chuyển tới bài tiếp theo để tiếp tục học nhé

4/5 - (4 bình chọn)

Related Posts:

1. [Bài 12: Lập trình STM32 với giao thức SPI](#)
2. [Bài 11: Lập trình STM32 với Giao thức UART](#)
3. [Bài 9: Lập trình STM32 ADC nhiều kênh với DMA](#)
4. [Bài 8: Lập trình STM32 đọc ADC một kênh](#)

5. Bài 3: Lập trình STM32 GPIO điều khiển Led và nút nhấn

6. Bài 2: Tổng quan về KIT STM32F103C8T6 Blue Pill



KHUÊ NGUYỄN

Chỉ là người đam mê điện tử và lập trình. Làm được gì thì viết cho anh em xem thôi. :D

5 THOUGHTS ON “BÀI 10: GIAO THỨC I2C, LẬP TRÌNH STM32 VỚI MODULE RTC DS3231”



Đào Duy Ngữ says:

Cho mình hỏi ở cái hàm HAL_I2C_MEM_READ_IT cái đoạn gửi địa chỉ 0x00 mà trong khi đó là dữ liệu bộ nhớ từ 0x00-0x06 thì chỉ cần đưa địa chỉ đầu tiên là 0x00 thôi à.

24/05/2021 AT 12:59 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Đúng rồi bạn, bạn chỉ cần xác định địa chỉ đầu tiên, số byte, và kiểu byte là nó tự động đọc từ đầu đến cuối

24/05/2021 AT 1:38 CHIỀU

TRẢ LỜI



Thanh Vũ says:

Ngay chỗ HAL_I2C_MEM_READ_IT while(1) ngay đoạn u8_tranBuffer phải là u8_revBuffer đúng không anh?

14/07/2021 AT 1:11 CHIỀU

TRẢ LỜI

**Khuê Nguyễn** says:

anh chưa hiểu ý em là sao

14/07/2021 AT 10:25 CHIỀU

TRẢ LỜI

**Thông** says:

Hình như đoạn đó anh phải đọc dữ liệu từ slave ra vi xử lí, nên anh phải lưu vào buffer là hàm u8_revBuffer , vì hàm callback của anh , anh dùng u8_revBuffer q

02/08/2021 AT 6:09 CHIỀU

TRẢ LỜI

Trả lời

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận *

Tên *

Email *

Trang web

PHẢN HỒI

Fanpage

Khuê Nguyễn Creator - Học Lập Trình Vi Điều Khiển
khoảng một tháng trước

Lý do thời gian gần đây mình không viết bài và làm thêm gì cả là đây 😊
Chính thức ra mắt sản phẩm định vị thông minh vTag.

Đây là một sản phẩm định vị đa năng với 3 công nghệ định vị WIFI, GPS, LBS kết hợp với sóng NB-IOT dành riêng cho các sản phẩm IOT.

Chỉ với 990.000đ chúng ta đã có thể có sản phẩm đẽ:

- Định vị trẻ em, con cái... [Xem thêm](#)

Bài viết khác

Lập trình 8051 - AT89S52

Khuê Nguyễn Creator

PROTEUS

Bài 1: Tổng quan về 8051 và chip AT89S51 - 52

Tổng quan về 8051

8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

Lập trình STM32 USB HID Host giao tiếp với chuột và bàn phím máy tính

Trong bài này chúng ta sẽ cùng học STM32 HID Host, biến STM32 giống như...

[ĐỌC THÊM](#)



Khuê Nguyễn Creator



Lộ trình học lập trình nhúng từ A tới Z

Lộ trình học lập trình nhúng từ A tới Z

Lập trình nhúng là một ngành có cơ hội nhưng cũng đòi hỏi nhiều kiến...

3 COMMENTS

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ

Lập trình STM32 SDIO đọc ghi dữ liệu vào thẻ nhớ SD card

Trong bài này chúng ta cùng học cách lập trình STM32 SDIO, một chuẩn giao...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32F407 DAC chuyển đổi số sang tương tự

Lập trình STM32 DAC tạo sóng hình Sin trên KIT STM32F407 Discovery

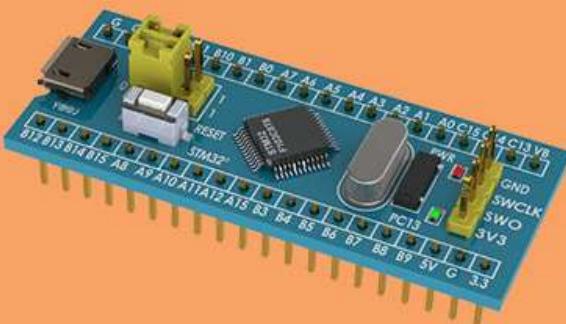
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Sử dụng hàm printf để in Log khi Debug trên STM32

Hướng dẫn sử dụng printf với STM32 Uart để in Log trên Keil C

Trong bài này chúng ta sẽ học cách retarget hàm printf của thư viện stdio...

3 COMMENTS

[ĐỌC THÊM](#)

ESP32 và Platform IO



Khuê Nguyễn Creator



Bài 9 WIFI: Lập trình ESP32 OTA nạp firmware trên Internet

Lập trình ESP32 FOTA nạp firmware qua mạng Internet với OTA Drive

Trong bài này chúng ta sẽ học cách sử dụng ESP32 FOTA (Firmware Over The...

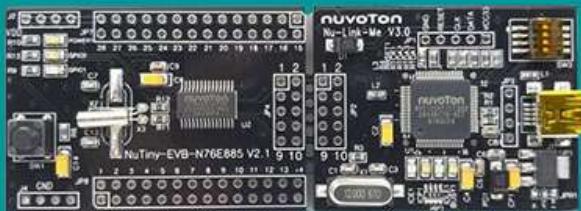
4 COMMENTS

[ĐỌC THÊM](#)

Lập trình Nuvoton



Khuê Nguyễn Creator



Cài đặt SDC Complier và Code:Blocks IDE

Hướng dẫn cài đặt SDCC và Code::Blocks lập trình Nuvoton

Ở bài này chúng ta sẽ cài đặt các công cụ cần thiết cho việc...

ĐỌC THÊM



Blog này làm ra để lưu trữ tất cả những kiến thức, những câu chuyện của mình. Đôi khi là những ý tưởng nhất thời, đôi khi là các dự án tự mình làm. Chia sẻ cho người khác

cũng là niềm vui của mình, kiến thức mỗi người là khác nhau, không hẳn quá cao siêu nhưng sẽ có lúc hữu dụng.

Liên Kết

Nhóm: Nghịen Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn