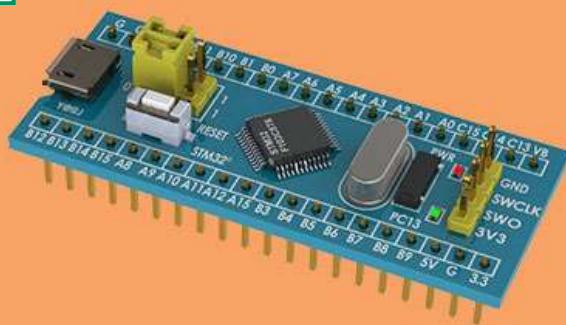


LẬP TRÌNH STM32

Bài 11: Lập trình STM32 với Giao thức UART

POSTED ON 18/07/2020 BY KHUÊ NGUYỄN

18
Th7**Khuê Nguyễn Creator**

Bài 11: UART là gì, nhận chuỗi dùng ngắt và DMA

Lập trình STM32 với Giao thức UART sẽ cung cấp cho các bạn kiến thức

- Giao thức UART là gì
- Cách cấu hình giao thức UART STM32 trên Cube MX
- Cách sử dụng giao thức UART trong lập trình

Bài 11 thuộc serie **Học lập trình STM32 từ A tới Z**

Mục Lục

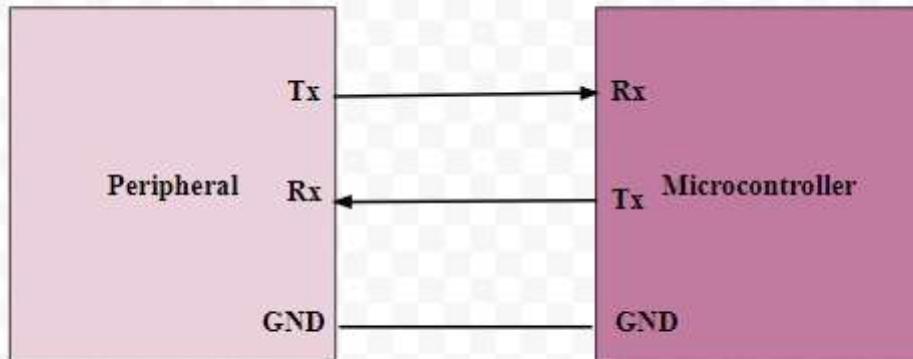
- 1. Giao thức UART là gì?
 - 1.1. Khái quát về UART
 - 1.2. Cách hoạt động của giao thức UART
- 2. Cấu hình giao thức UART STM32 trong Cube MX
- 3. Lập trình giao thức UART trên Keil C
 - 3.1. Related posts:

Giao thức UART là gì?

Khái quát về UART

UART tiếng anh là Universal Asynchronous Reciver/Transmister một chuẩn giao tiếp không đồng bộ cho MCU và các thiết bị ngoại vi.

Chuẩn UART là chuẩn giao tiếp điểm và điểm, nghĩa là trong mạng chỉ có hai thiết bị đóng vai trò là transmister hoặc reciver.



Bài 11: Lập trình STM32 với Giao thức UART 56

Cách hoạt động của giao thức UART

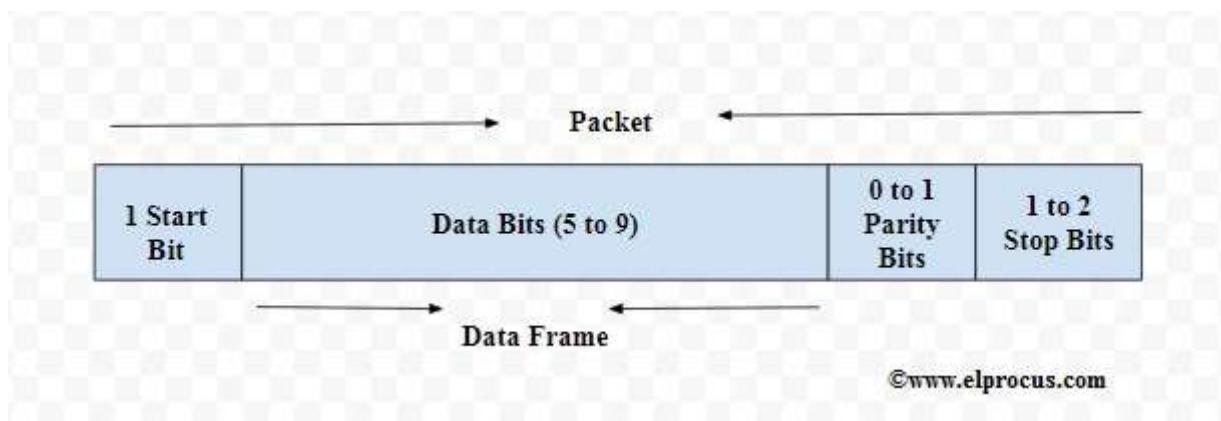
UART là giao thức truyền thông không đồng bộ, nghĩa là không có xung Clock, các thiết bị có thể hiểu được nhau nếu các Setting giống nhau

UART là truyền thông song công(Full duplex) nghĩa là tại một thời điểm có thể truyền và nhận đồng thời.

Trong đó quan trọng nhất là Baund rate (tốc độ Baund) là khoảng thời gian dành cho 1 bit được truyền. Phải được cài đặt giống nhau ở gửi và nhận.

Sau đó là định dạng gói tin.

Định dạng gói tin như sau:



Bài 11: Lập trình STM32 với Giao thức UART 57

Start – Bit

Start-bit còn được gọi là bit đồng bộ hóa được đặt trước dữ liệu thực tế. Nói chung, một đường truyền dữ liệu không hoạt động được điều khiển ở mức điện áp cao. Để bắt đầu truyền dữ liệu, truyền UART kéo đường dữ liệu từ mức điện áp cao (1) xuống mức điện áp thấp (0). UART thu được thông báo sự chuyển đổi này từ mức cao sang mức thấp qua đường dữ liệu cũng như bắt đầu hiểu dữ liệu thực. Nói chung, chỉ có một start-bit.

Stop – Bit

Bit dừng được đặt ở phần cuối của gói dữ liệu. Thông thường, bit này dài 2 bit nhưng thường chỉ sử dụng 1 bit. Để dừng sóng, UART giữ đường dữ liệu ở mức

điện áp cao.

Parity Bit

Bit chẵn lẻ cho phép người nhận đảm bảo liệu dữ liệu được thu thập có đúng hay không. Đây là một hệ thống kiểm tra lỗi cấp thấp & bit chẵn lẻ có sẵn trong hai phạm vi như Chẵn lẻ – chẵn lẻ cũng như Chẵn lẻ – lẻ. Trên thực tế, bit này không được sử dụng rộng rãi nên không bắt buộc.

Data frame

Các bit dữ liệu bao gồm dữ liệu thực được truyền từ người gửi đến người nhận. Độ dài khung dữ liệu có thể nằm trong khoảng 5 & 8. Nếu bit chẵn lẻ không được sử dụng thì chiều dài khung dữ liệu có thể dài 9 bit. Nói chung, LSB của dữ liệu được truyền trước tiên sau đó nó rất hữu ích cho việc truyền.

Cấu hình giao thức UART STM32 trong Cube MX

Ở bài này chúng ta sẽ thực hành truyền UART từ bộ UART1 lên máy tính, sử dụng Baundrate = 9600, các setup khác để mặc định.

Trong chế độ sử dụng ngắn ta sẽ kiểm tra chuỗi gửi đến, nếu thấy kí tự Enter thì gửi ngược lại

Trong chế độ DMA ta sẽ sử dụng truyền thẳng vào Buffer để biết cách hoạt động của ngắn và DMA

Các bạn phải chuẩn bị bộ chuyển đổi UART to USB như Bài 1 mình giới thiệu, và phần mềm Hercules Terminal.

Xem hướng dẫn tại: [Hướng dẫn sử dụng Hercules Terminal](#)

Đầu tiên các bạn mở CubeMX lên, Chọn chip STM32F103C8, trong SYS debug chọn Serial Wire. Mình đã nói rõ trong [Bài 3](#) nhé.

Trong Tab UART chọn UART1

Bài 11: Lập trình STM32 với Giao thức UART 58

Mode: Asynchronous (không đồng bộ)

Hard ware: Disable

NVIC Setting Tick vào bật ngắt UART

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
USART1 global interrupt	<input checked="" type="checkbox"/>	0	0

Bài 11: Lập trình STM32 với Giao thức UART 59

Đặt tên chọn tool chain rồi Gen code

Project Settings

Project Name: Bai11_UART_IT_DMA

Project Location: D:\STM32 Cube Youtube Browse

Application Structure: Basic ▼ Do not generate the main()

Toolchain Folder Location: D:\STM32 Cube Youtube\Bai11_UART_IT_DMA\

Toolchain / IDE: MDK-ARM ▼ Min Version: V5 ▼ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

Mcu Reference: STM32F103C8Tx

Firmware Package Name and Version: STM32Cube FW_F1 V1.8.0

Use Default Firmware Location
C:/Users/admin/STM32Cube/Repository/STM32Cube_FW_F1_V1.8.0 Browse

Bài 11: Lập trình STM32 với Giao thức UART 60

Lập trình giao thức UART trên Keil C

Mở project trong main.c cấu hình như sau.

Khởi tạo 2 Buffer để truyền nhận dữ liệu

1 biến nhận Rx data

1 biến lưu giá trị con trả ghi vào Rx Buffer

1 Cờ báo nhận thành công để gửi TX

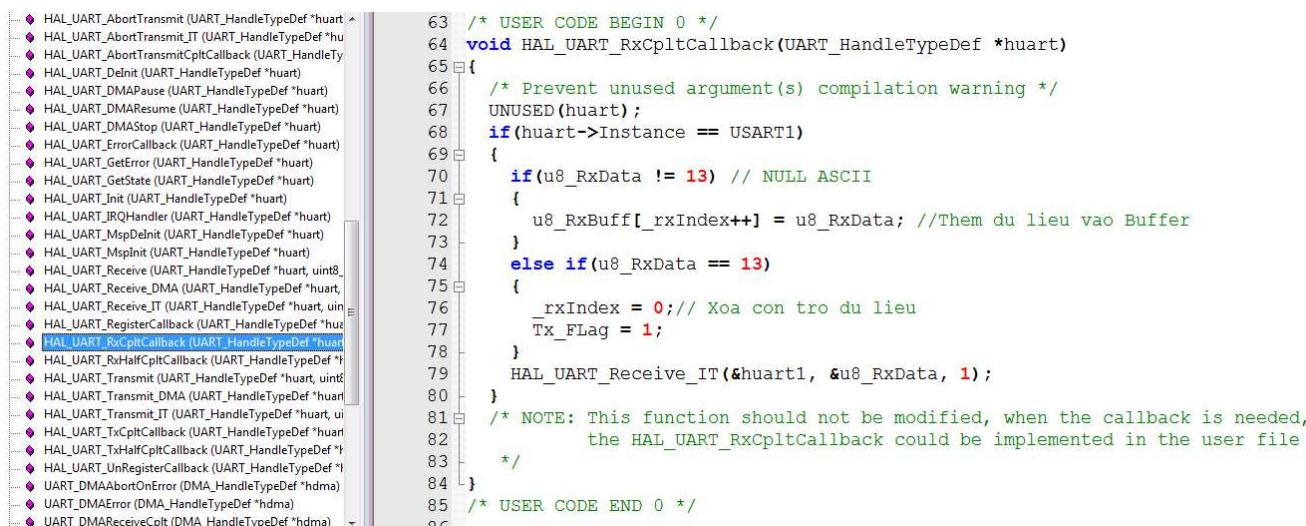
```

21 /* Includes --- */
22 #include "main.h"
23
24 /* Private includes ----- */
25 /* USER CODE BEGIN Includes */
26 uint8_t u8_RxBuff[20]; // buffer luu chuoi nhan duoc
27 uint8_t u8_RxData; // luu byte nhan duoc
28 uint8_t u8_TxBuff[20] = "Hello Anh em !!!!"; // buffer truyen di
29 uint8_t _rxIndex; // con tro cua rxbuff
30 uint16_t Tx_Flag = 0;
31 /* USER CODE END Includes */

```

Bài 11: Lập trình STM32 với Giao thức UART 61

Tìm hàm RxCpltCallback sau đó cấu hình như sau



```

63 /* USER CODE BEGIN 0 */
64 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
65 {
66     /* Prevent unused argument(s) compilation warning */
67     UNUSED(huart);
68     if(huart->Instance == USART1)
69     {
70         if(u8_RxData != 13) // NULL ASCII
71         {
72             u8_RxBuff[_rxIndex++] = u8_RxData; //Them du lieu vao Buffer
73         }
74         else if(u8_RxData == 13)
75         {
76             _rxIndex = 0;// Xoa con tro du lieu
77             Tx_Flag = 1;
78         }
79         HAL_UART_Receive_IT(&huart1, &u8_RxData, 1);
80     }
81     /* NOTE: This function should not be modified, when the callback is needed,
82            the HAL_UART_RxCpltCallback could be implemented in the user file
83     */
84 }
85 /* USER CODE END 0 */

```

Bài 11: Lập trình STM32 với Giao thức UART 62

Nếu u8_RxData nhận được là khác kí tự Enter (giá trị 13 trong bảng ASCII), ghi giá trị u8_RxData vào u8_RxBuf sau đó tăng giá trị _RxIndex

Nếu RxData là kí tự Enter, xóa biến _RxIndex về 0 và bật cờ Tx

Gọi hàm HAL_UART_Receive_IT(&huart1, &u8_RxData, 1); để tiếp tục nhận dữ liệu nếu có dữ liệu truyền qua.

Trước main chúng ta gửi UART dữ liệu ban đầu trong u8_TxBuff là Hello anh em!!!

Sau đó bật nhận dữ liệu bằng Ngắt

```

114     MX_GPIO_Init();
115     MX_USART1_UART_Init();
116     /* USER CODE BEGIN 2 */
117     HAL_UART_Transmit(&huart1, u8_TxBuff, sizeof(u8_TxBuff), 100);
118     HAL_UART_Receive_IT(&huart1, &u8_RxData, 1);
119     /* USER CODE END 2 */
120
121     /* Infinite loop */
122     /* USER CODE BEGIN WHILE */
123     while (1)
124     {
125         /* USER CODE END WHILE */
126
127         /* USER CODE BEGIN 3 */
128         if(Tx_Flag)
129         {
130             for(int i = 0; i< 20; i++) // copy du lieu tu Rx sang Tx
131             {
132                 u8_TxBuff[i] = u8_RxBuff[i];
133             }
134             HAL_UART_Transmit(&huart1, u8_TxBuff, sizeof(u8_TxBuff), 100);
135             Tx_Flag = 0;
136         }
137     }
138     /* USER CODE END 3 */

```

Bài 11: Lập trình STM32 với Giao thức UART 63

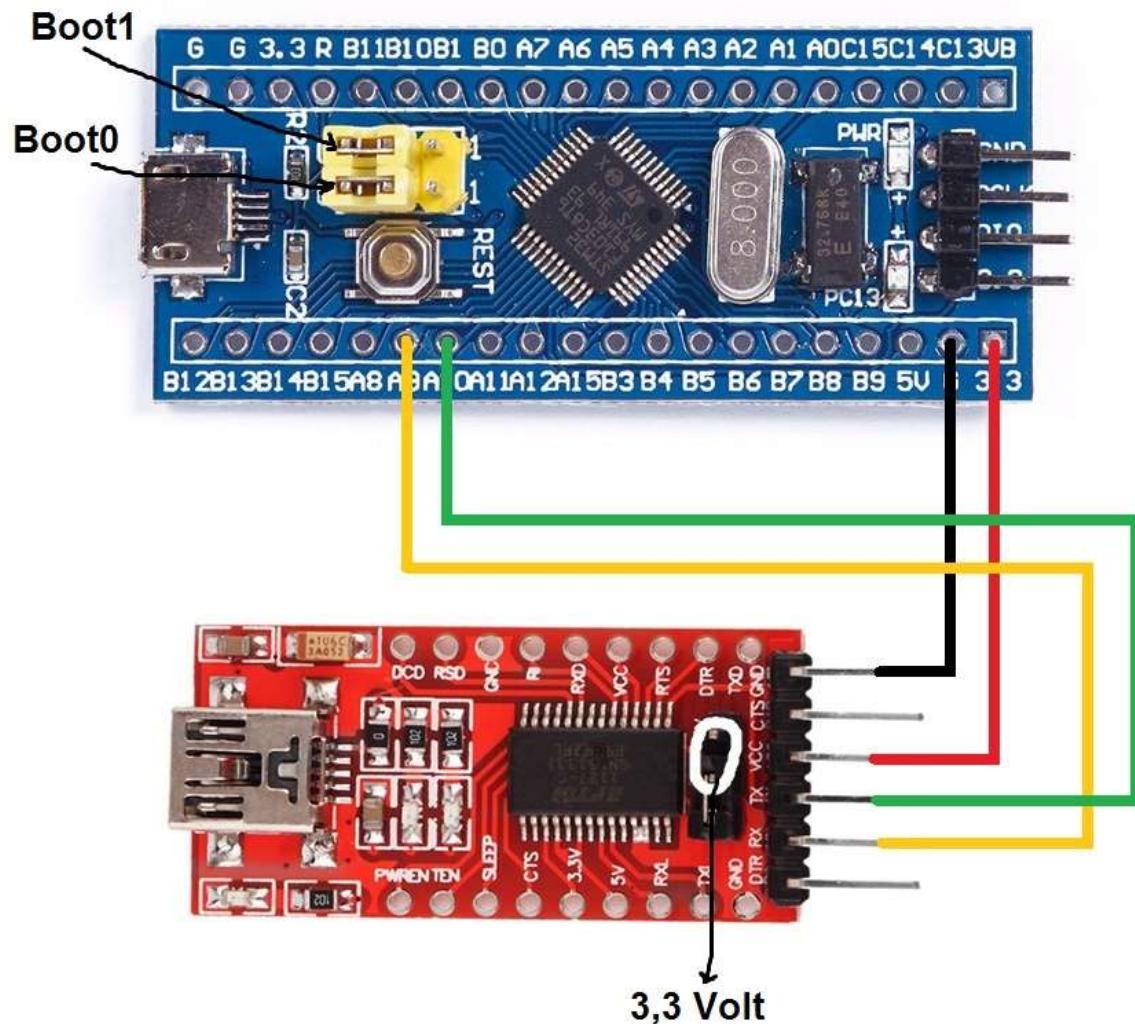
Trong while(1)

Kiểm tra cờ truyền UART nếu được bật lên.

Sao chép giá trị RxBuff cho TxBuff, sau đó truyền đi và xóa cờ TX

Nhấn F7 Build và nạp chương trình vào chip.

Kết nối STM32 vào UART to USB



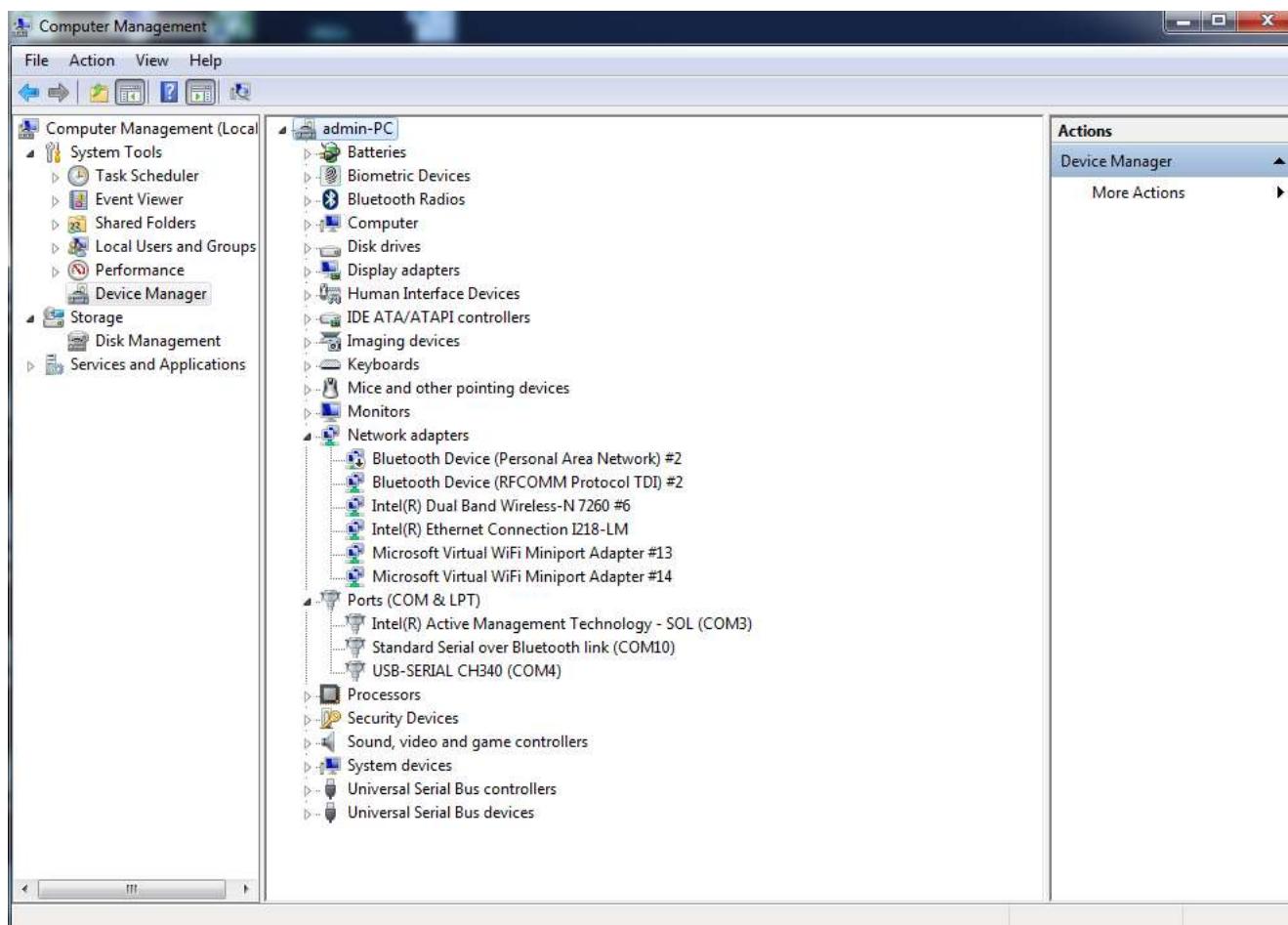
Bài 11: Lập trình STM32 với Giao thức UART 64

Sau đó bật Hercules terminal lên, kiểm tra cổng kết nối với UART to USB bằng cách nhấn chuột phải vào biểu tượng My computer -> manager



Bài 11: Lập trình STM32 với Giao thức UART 65

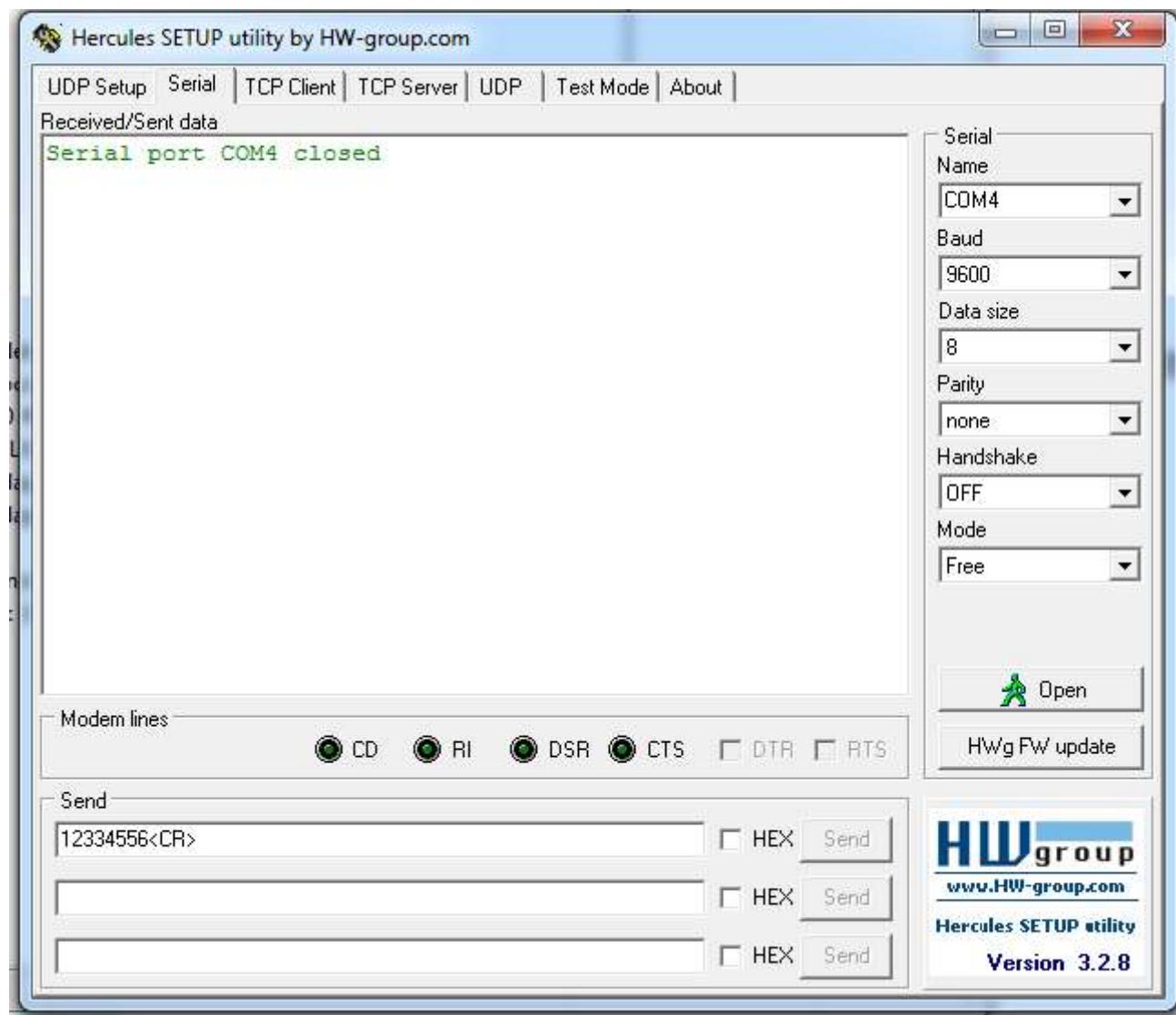
Mở tab Device -> Ports ta thấy rằng port đang sử dụng là COM4 với driver là CH340



Bài 11: Lập trình STM32 với Giao thức UART 66

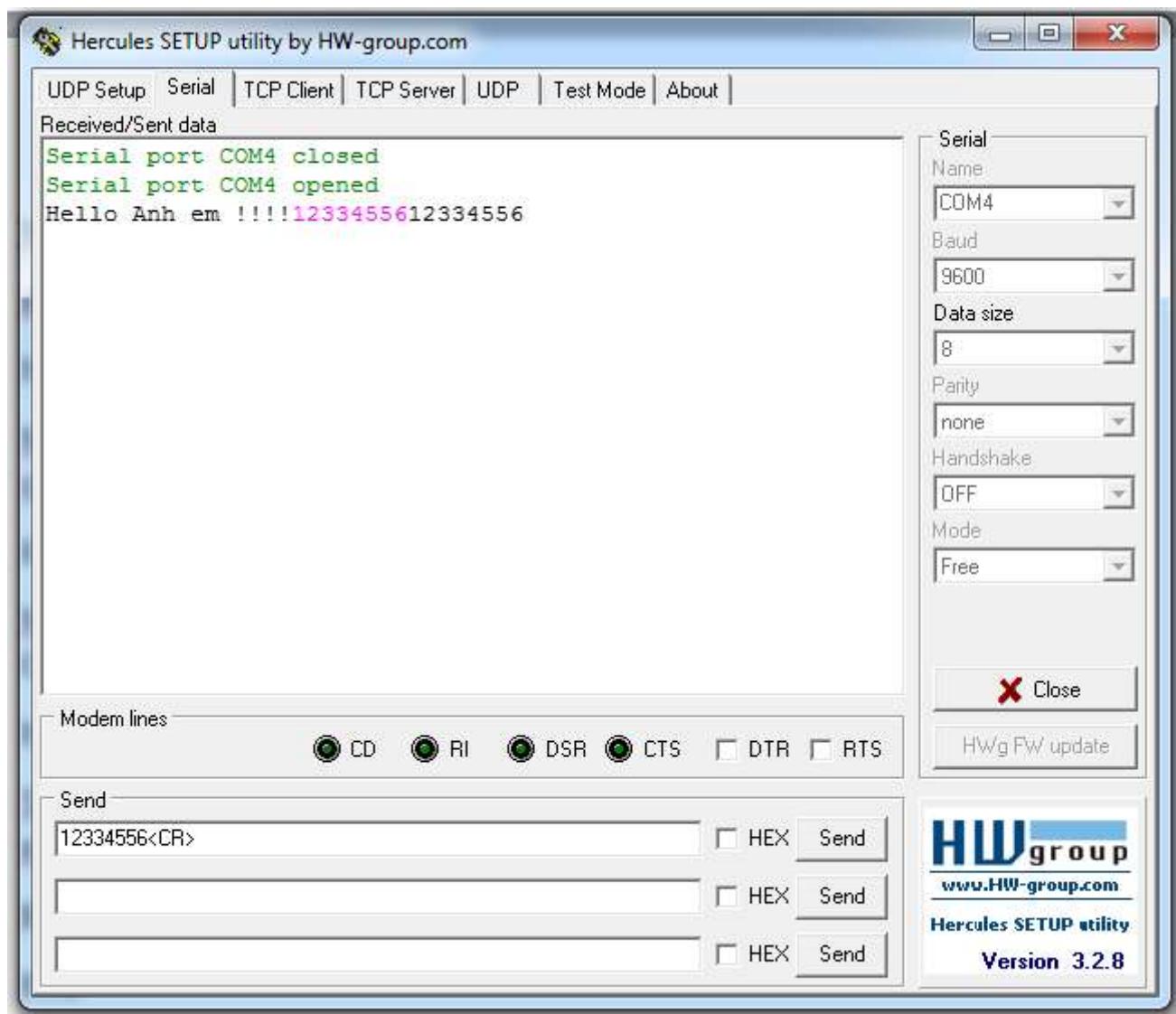
Trong Terminal, chuyển Tab Serial – Name chọn cổng com bạn đang dùng với mình là COM4

Baud set là 9600, còn lại để mặc định rồi ấn vào Open



Bài 11: Lập trình STM32 với Giao thức UART 67

Nhấn Reset trên mạch để mạch chạy từ đầu. Ta sẽ thấy mạch gửi kí tự đầu tiên. Nhập tin nhắn bạn cần gửi lên mạch và kết thúc là <CR> (kí tự Enter trong ASCII), mạch sẽ gửi lại đúng tin nhắn đó.



Bài 11: Lập trình STM32 với Giao thức UART 68

Truyền nhận dữ liệu bằng DMA

Để truyền nhận dữ liệu sử dụng DMA các bạn làm như sau

Trong CubeMX , Tab UART1 Config-> DMA Settings

Các bạn nhấn Add, DMA request chọn USART1_RX, Mode Circular (chỉ cần gọi 1 lần sẽ truyền liên tục từ ngoại vi về bộ nhớ), Data Width là Byte

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 5	Peripheral To Memory	Low

Add Delete

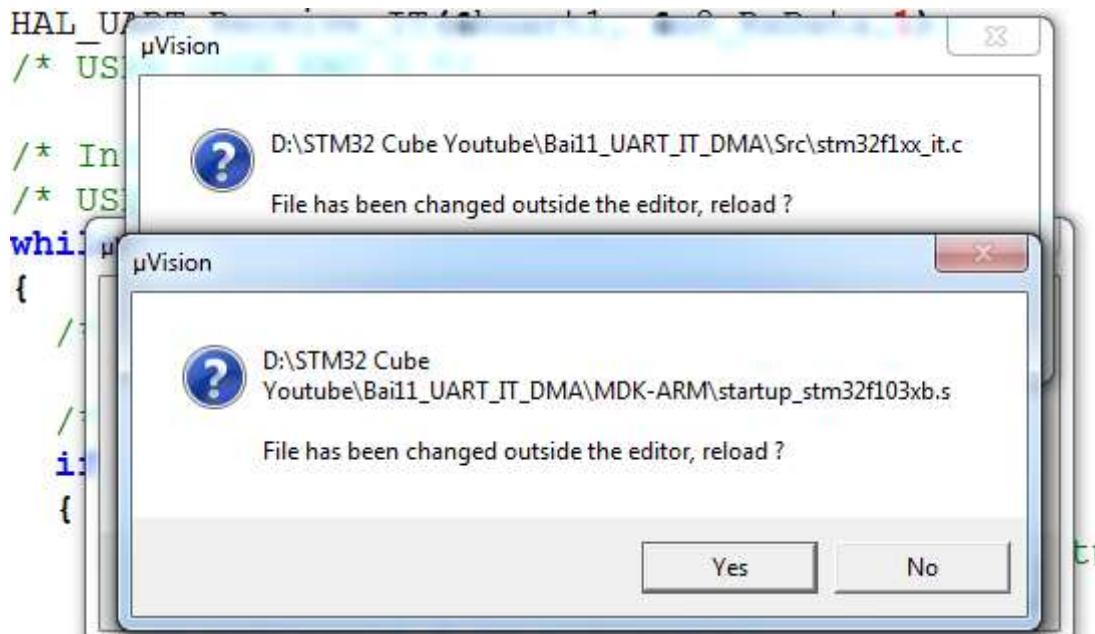
DMA Request Settings

	Peripheral	Memory
Mode	Circular	<input checked="" type="checkbox"/>
Increment Address	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Data Width	Byte	Byte

Bài 11: Lập trình STM32 với Giao thức UART 69

Nhấn Gen Code và Close

Trong KeilC sẽ báo Reload lại file, bạn nhấn ok



Bài 11: Lập trình STM32 với Giao thức UART 70

Trong hàm callback ta comment hết các hàm trước, viết lại hàm Transmit

```

66 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
67 {
68     /* Prevent unused argument(s) compilation warning */
69     UNUSED(huart);
70     if(huart->Instance == USART1)
71     {
72         if(u8_RxData != 13) // NULL ASCII
73         {
74             u8_RxBuff[_rxIndex++] = u8_RxData; //Them du lieu vao Buffer
75         }
76         else if(u8_RxData == 13)
77         {
78             _rxIndex = 0;// Xoa con tro du lieu
79             TX_FFlag = 1;
80         }
81         HAL_UART_Receive_IT(&huart1, &u8_RxData, 1);
82         HAL_UART_Transmit(&huart1, u8_TxBuff, sizeof(u8_TxBuff),100);
83     }
84     /* NOTE: This function should not be modified, when the callback is needed,
85            the HAL_UART_RxCpltCallback could be implemented in the user file
86     */
87 }
88 /* USER CODE END 0 */

```

Bài 11: Lập trình STM32 với Giao thức UART 71

Trong main() ta gọi hàm Recive_DMA thay cho Recive_IT, và comment toàn bộ While (1)

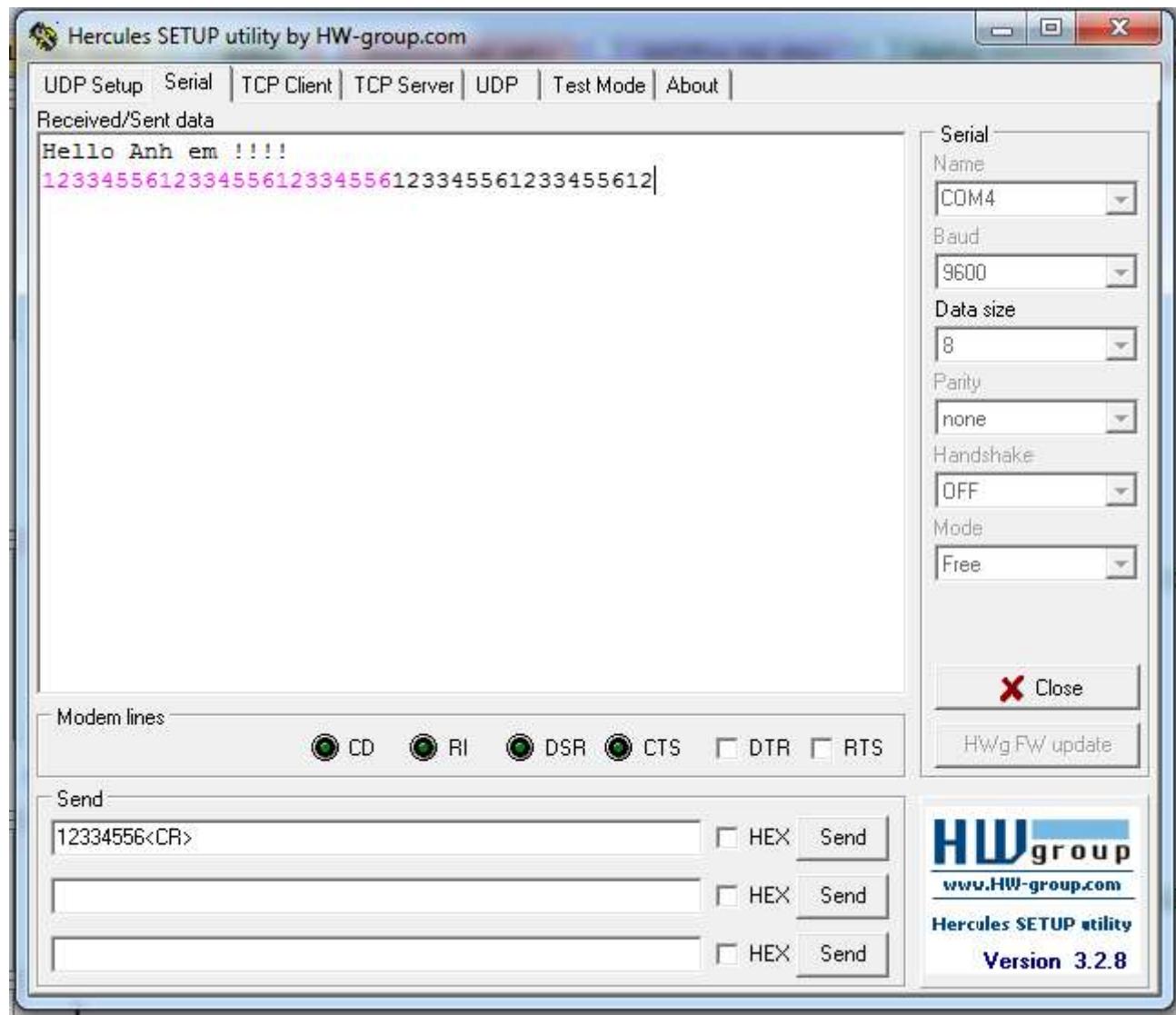
```

123
124     HAL_UART_Transmit(&huart1, u8_TxBuff, sizeof(u8_TxBuff),100);
125     HAL_UART_Receive_DMA(&huart1, u8_RxBuff, sizeof(u8_RxBuff));
126     /* USER CODE END 2 */
127
128     /* Infinite loop */
129     /* USER CODE BEGIN WHILE */
130     while (1)
131     {
132         /* USER CODE END WHILE */
133
134         /* USER CODE BEGIN 3 */
135         //     if(TX_FFlag)
136         //     {
137         //         for(int i = 0; i< 20; i++) // copy du lieu tu Rx sang Tx
138         //         {
139         //             u8_TxBuff[i] = u8_RxBuff[i];
140         //         }
141         //         HAL_UART_Transmit(&huart1, u8_TxBuff, sizeof(u8_TxBuff),100);
142         //         TX_FFlag = 0;
143         //     }
144     }
145     /* USER CODE END 3 */

```

Bài 11: Lập trình STM32 với Giao thức UART 72

Build và nạp chương trình và xem kết quả.



Bài 11: Lập trình STM32 với Giao thức UART 73

Ta thấy rằng, chỉ khi nhấn đủ 20 giá trị = Độ dài của Rx_Buffer thì chương trình mới nhảy vào ngắt. và truyền dữ liệu lại. Vì vậy truyền UART nên sử dụng các frame có độ dài cố định.

Kết luận

Giao thức UART được sử dụng rất nhiều trong các sản phẩm nhúng. Việc lập trình giao thức này cũng không quá khó khăn, chỉ cần bạn đọc kĩ phần hướng dẫn, chắc chắn sẽ làm được.

5/5 - (1 bình chọn)

Related Posts:

1. [Lập trình STM32 từ A tới Z](#)
2. [Bài 12: Lập trình STM32 với giao thức SPI](#)
3. [Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231](#)
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

23 THOUGHTS ON “BÀI 11: LẬP TRÌNH STM32 VỚI GIAO THỨC UART”



Nguyễn văn công says:

Mình muốn nhờ chút có cách nào đọc mã giao tiếp giữa 2 con stm32 nối với nhau theo chuẩn uart không, mình đã dùng một mạch USB TTL uart nối song song với dây nối giữa 2 mạch giao tiếp bằng máy tính mà mã truyền của nó không đúng.

29/03/2021 AT 7:54 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Bạn truyền giữa Stm32 và máy tính hay stm32 với stm32 nhỉ?

29/03/2021 AT 9:34 CHIỀU

TRẢ LỜI



Manh Duc says:

Em hỏi chút q, có cách nào để giao tiếp uart giữa stm32 với esp8266 không.
Vì e truyền từ stm32 sang esp8266 1 chuỗi mà n không nhận.

31/05/2021 AT 8:04 SÁNG

TRẢ LỜI



Khuê Nguyễn says:

Truyền bình thường mà e, ko nhận thì do e code bên esp rồi

01/06/2021 AT 12:06 SÁNG

TRẢ LỜI



Khoa Đỗ says:

Thế e muốn STM đọc ADC rồi UART qua ESP thì làm sao q?

21/06/2021 AT 9:12 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

E tách ra thành 2 phần. Phần 1 đọc ADC, phần 2 Gửi
UART, data gửi chính là giá trị ADC đọc được

22/06/2021 AT 8:36 SÁNG

TRẢ LỜI



thống says:

mình mới tìm hiểu nhưng đọc và tra tài liệu thì những
chân Rx Tx của esp những người hướng dẫn nó khác so
với sơ đồ chân mặc dù cùng loại (CH340 hay cp2102)

20/03/2022 AT 3:38 CHIỀU

TRẢ LỜI



Dũng nguyễn says:

Anh ơi cho em hỏi chút với q . Cái chỗ Kết nối STM32 vào UART to USB là
phải có cái mạch chuyển UART USB mới dùng được cái hercules terminal hả
anh.

22/06/2021 AT 8:15 SÁNG

TRẢ LỜI



Khuê Nguyễn says:

đúng rồi em, e mua và cài driver cho nó thường là ch340 và pl2303

22/06/2021 AT 8:39 SÁNG

TRẢ LỜI



quang says:

vậy để truyền dữ liệu thay đổi(ví dụ nhiệt độ, tốc độ) qua uart thì phải làm ntn ạ

11/07/2021 AT 12:57 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

truyền bình thường thôi em

01/08/2021 AT 2:21 CHIỀU

TRẢ LỜI



dong tran says:

Cho em hỏi: Em dùng DMA để nhận dữ liệu, nhưng khi mảng dữ liệu nhận, tràn. Thì nó không ghi đè lên, hay xóa đi các kí tự cũ đi, là sao vậy anh?

01/08/2021 AT 10:40 SÁNG

TRẢ LỜI



Khuê Nguyễn says:

em phải tự xóa mà

01/08/2021 AT 2:22 CHIỀU

TRẢ LỜI



Bao.N says:

anh ơi cho em hỏi uart giữa stm32 với module R305 có giống với cách anh hướng dẫn không anh

04/09/2021 AT 10:10 CHIỀU

TRẢ LỜI

Khuê Nguyễn says:



Giống nhau e nhé, vậy nên mới gọi là giao thức

04/09/2021 AT 10:13 CHIỀU

TRẢ LỜI



nguyễn tú says:

cho em hỏi. Truyền một gói tin từ máy tính xuống STM32 có định dạng như sau:

|X|Data|#(bao gồm 3 byte, trong đó "" là kí tự bắt đầu,"#" là kí tự kết thúc,"X"(X=1->16) là byte khai báo kích thước của data;" data" là các kí tự, có kích thước từ 1 đến 16byte.

khi nhận đúng định dạng gói tin: hiển thị nội dung gói tin lên LCD 16x2 đồng thời truyền trả về máy tính chữ "OK" thông qua UART. Khi nhận sai gói định dạng: không thay đổi nội dung LCD, đồng thời truyền trả về máy tính chữ "ERROR" thông qua UART

30/09/2021 AT 8:10 CHIỀU

TRẢ LỜI



Trường kelvin says:

anh ơi cho em hỏi làm thế nào để sau khoảng 1-3s thì data truyền sau sẽ ghi đè lên data truyền trước ???

10/10/2021 AT 2:56 CHIỀU

TRẢ LỜI



Tú Huỳnh says:

bài này anh làm ví dụ hay hơn những người khác lắm luôn.

05/12/2021 AT 9:23 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

cám ơn em

25/12/2021 AT 4:35 CHIỀU

TRẢ LỜI

huy says:

lập trình gửi dữ liệu từ Herculus qua protues và
hiển thị dữ liệu lên LCD 16x2 .. thì làm thêm code gì nữa anh ạ

07/12/2021 AT 7:37 CHIỀU

TRẢ LỜI

Khuê Nguyễn says:

Trc tiên là điều khiển dc LCD16x2 sau đó là truyền dữ liệu qua lcd
thay vì UART thôi e

25/12/2021 AT 4:36 CHIỀU

TRẢ LỜI

Ngọc says:

Anh ơi, nếu em dùng UART từ vi điều khiển qua RS 485 rồi mới vào máy tihs
thì cũng làm tương tự đúng không hả anh?

Em cảm ơn ạ!

31/12/2021 AT 6:30 SÁNG

TRẢ LỜI

Khuê Nguyễn says:

đúng rồi, về cơ bản là giống nhau

04/01/2022 AT 1:24 CHIỀU

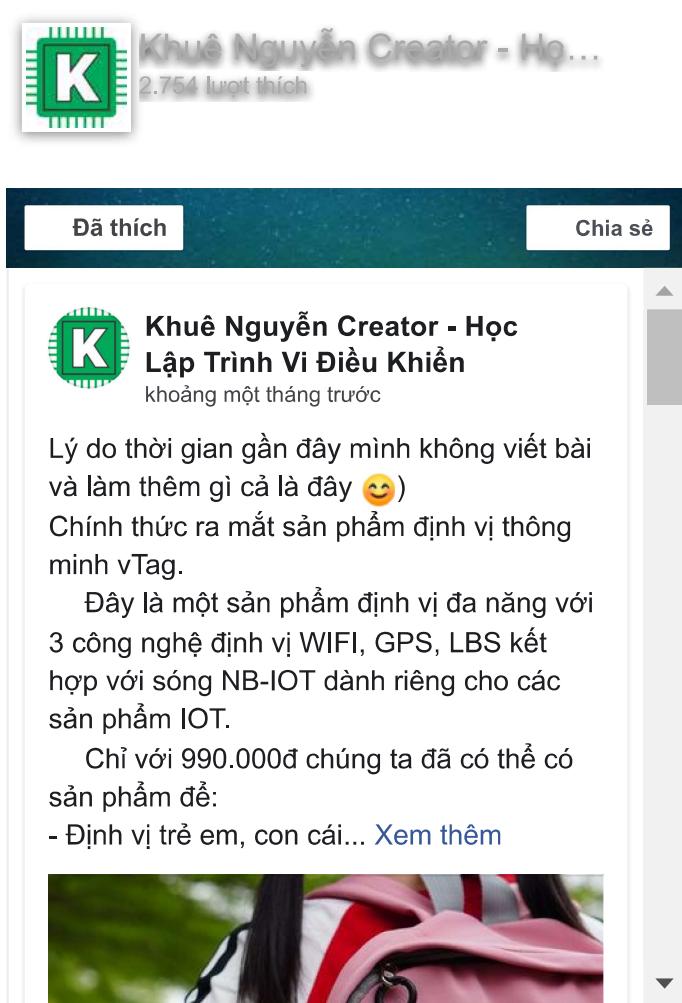
TRẢ LỜI

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
2.754 lượt thích

Đã thích Chia sẻ

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 để:

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



Bài viết khác

Lập trình 8051 - AT89S52



Khuê Nguyễn Creator



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 chung ta sẽ cùng học STM32 HID Host, biến STM32 giống như...

[ĐỌC THÊM](#)



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

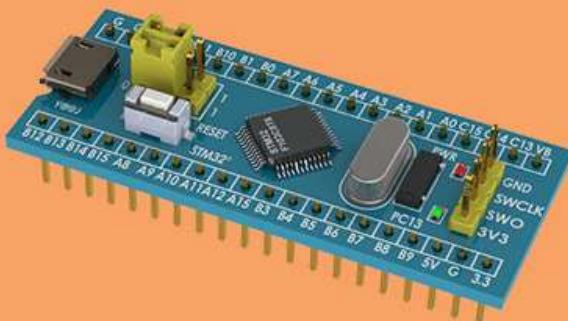
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[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: Nghiên Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn