**CHUYÊN ĐỀ THIẾT KẾ HỆ THỐNG NHÚNG 1**

**BÀI THỰC HÀNH 3: GIAO TIẾP CAN TRÊN VI ĐIỀU KHIỂN STM32F405**

1. **Mục tiêu**

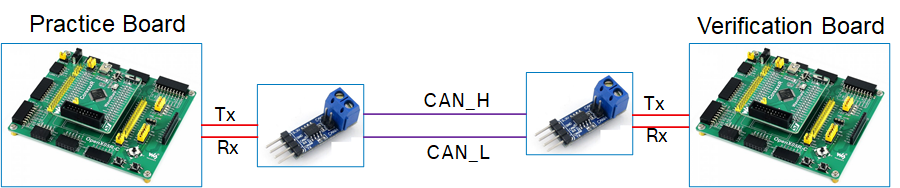
* Giúp sinh viên hiểu và có thể lập trình được chuẩn giao tiếp CAN trên vi điều khiển STM32F405.
* Giúp sinh viên biết được cách truyền và nhận dữ liệu, cách giao tiếp giữa các node trong chuẩn giao tiếp CAN ở tầng vật lý.
* Giúp sinh viên làm quen với việc kiểm tra độ tin cậy của dữ liệu nhận được và ứng dụng checksum vào việc truyền nhận dữ liệu.

1. **Chuẩn bị trước**

* Tìm hiểu lý thuyết về chuẩn giao tiếp CAN, đặc biệt là CAN high speed.

1. **Nội dung thực hành**
2. **Giới thiệu về chuẩn giao tiếp CAN**

CAN là một chuẩn giao tiếp có dây bán song công thực hiện theo kiểu truyền nhận tuần tự từng bit không đồng bộ. Mạng giao tiếp CAN có từ hai node trở lên và có mô hình truyền nhận như trong **Hình 1**. Sinh viên đọc thêm tài liệu BOSCH CAN specification 2.0 [1] để có thể hiểu thêm về chuẩn giao tiếp CAN.



**Hình 1 Tổng quan chuẩn giao tiếp CAN**

1. **Lập trình giao tiếp CAN trên KIT Open405R-C Package A, STM32F4 Development Board**

Sinh viên viết chương trình để thực hiện giao tiếp giữa KIT Open405R-C Package A, STM32F4 Development Board và máy tính. Trên bộ KIT đã cung cấp sẵn 2 thiết bị CAN transceiver để sinh viên có thể tạo thành mạng CAN 2 nodes. Để kiểm tra các chân giao tiếp và sơ đồ nguyên lý cũng như các tài liệu tham khảo về KIT thực hành, sinh viên tham khảo trang wiki của sản phẩm tại [2].

1. **Kiểm tra, theo dõi và nhận diện dữ liệu CAN trên đường truyền**

Sinh viên thực hiện kết nối mạch CAN transceiver với bộ dao động ký (Oscilloscope) để theo dõi tín hiệu được truyền trên đường truyền cũng như các giá trị dữ liệu sau khi ra khỏi CAN controller.

1. **Hướng dẫn thực hành**
2. **Lập trình giao tiếp CAN đơn giản trên KIT Open405R-C Package A, STM32F4 Development Board**

Sinh viên viết chương trình để thực hiện giao tiếp CAN trên KIT Open405R-C Package A, STM32F4 Development Board.

* Sinh viên tạo project mới như các bài thực hành trước.
* Tại phần Connectivity, sinh viên chọn CAN1 và tick vào Activated. Sinh viên chọn các chân tương ứng cho CAN1 theo như sơ đồ nguyên lý của nhà sản xuất để cấu hình các chức năng cho CAN1. Sau đó sinh viên cài đặt các thông số cho CAN1 theo các thông số yêu cầu sau:
  + CAN ID: 11-bit
  + Bit timing:

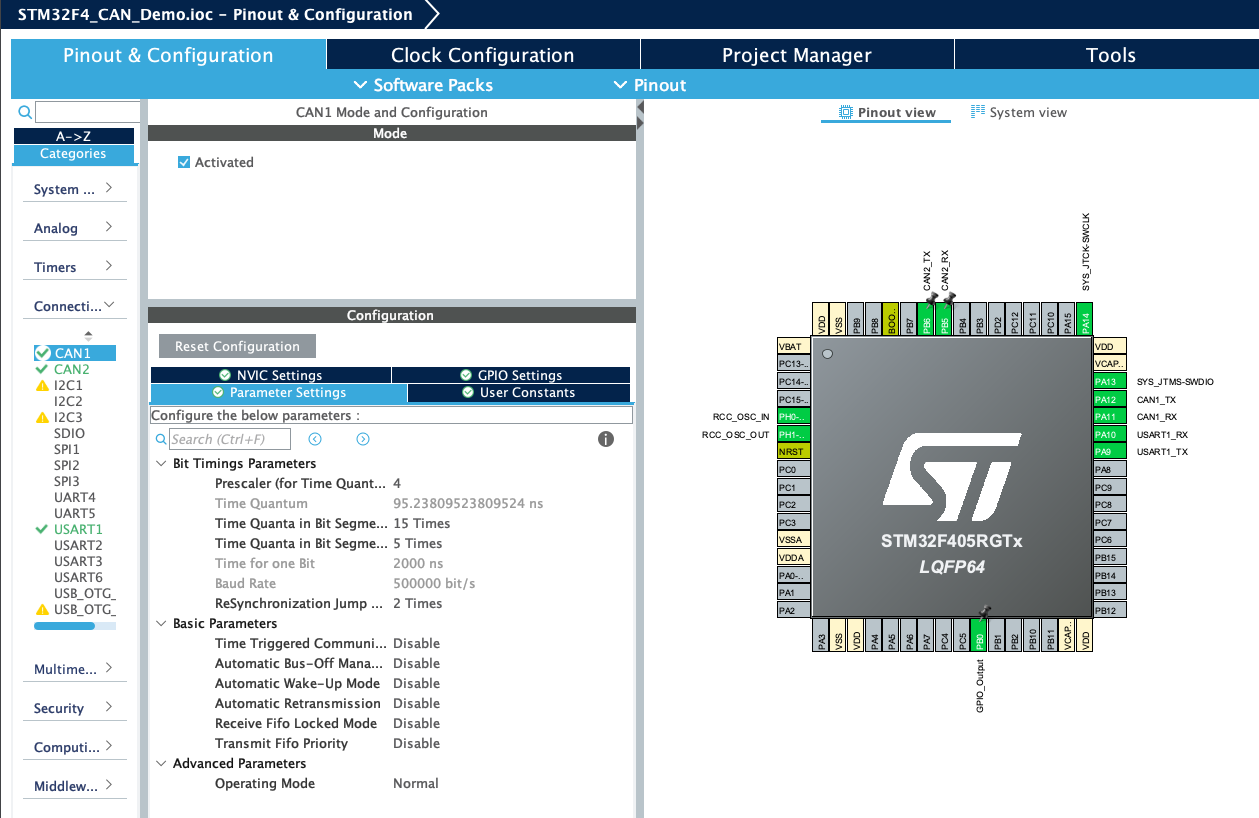
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Symbol | Minimum value | Nominal value. | Maximum value | Unit | Remarks |
| Bit timing | tBit | 1992 | 2000 | 2008 | ns | fHSCAN = 500kbps (± 0.4 %) |
| Tq quantity | NBT | 10 | 16 | 20 | Tq |  |
| Sampling position | tSP | 75 | - | 82 | % |  |
| Synchronisation jump width | SJW | 2 | 2 | 3 | Tq |  |
| Sampling amount | NSP | - | 1 | - |  |  |

* + Thứ tự các bit tín hiệu được truyền nhận theo định dạng Motorola (big endian format). **Hình 2** thể hiện một ví dụ về một đoạn dữ liệu có chứa gói tin gồm 10-bit tín hiệu theo định dạng của Motorola với start bit tại vị trí bit thứ 28.
  + Những bit không sử dụng trong gói tin phải được gán bằng 0. Những byte không sử dụng phải được gán bằng 0x00.



**Hình 2 Cài đặt cho gói tin 10-bit tín hiệu theo định dạng Motorola**

* Ngoài ra, để có thể nhận gói tin từ CAN1 thông qua interrupt thì sinh viên cấu hình cho phép interrupt từ chân CAN1 RX tương ứng trong phần NVIC setting.
* Sinh viên cài đặt các thông số cho CAN2 tương tự với CAN1.
* Sinh viên thực hiện gửi gói tin từ CAN1 (Node 1) tới CAN2 (Node 2) với các thông số sau: ID: 0x555, Data 8-byte: “Hi CE437” và kiểm tra các gói tin nhận được từ CAN2 (Node 2).
* Tương tự, sinh viên thực hiện gửi gói tin từ CAN2 (Node 2) tới CAN1 (Node 1) với các thông số sau: ID: 0x2AA, Data 8-byte: “Bye Bye!” và kiểm tra gói tin nhận được từ CAN1 (Node 1).
* Sinh viên có thể sử dụng cổng UART và các LEDs trên Kit để kiểm tra lại gói tin nhận được trên đường truyền CAN.



**Hình 3 Cài đặt các thông số cơ bản cho CAN1**

Diagram, table

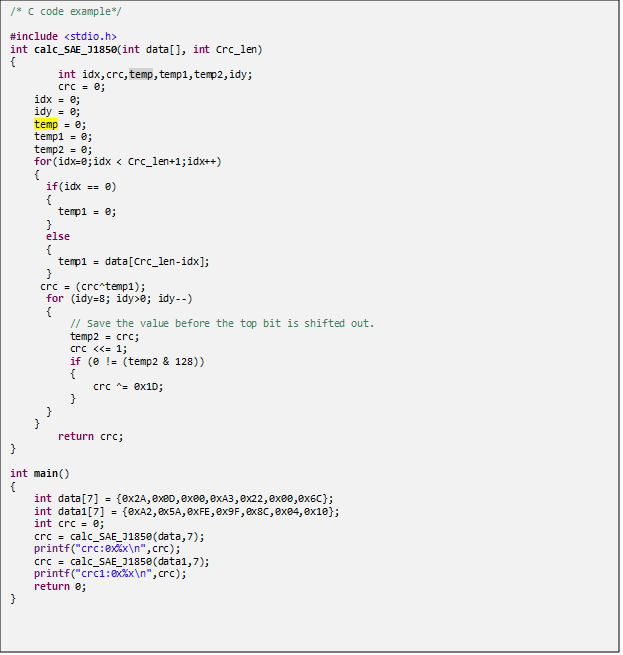
Description automatically generated

**Hình 4 Định dạng một gói tin CAN**

1. **Lập trình giao tiếp CAN theo yêu cầu**

Sinh viên lập trình giao tiếp giữa 2 module CAN1 và CAN2 trên Kit thực hành theo các yêu cầu sau:

* CAN1 là node 1:
  + TX\_ID: 0x012
  + RX\_ID: 0x0A2
* CAN2 là node 2:
  + TX\_ID: 0x0A2
  + RX\_ID: 0x012
* Node 2 gửi gói tin gồm: byte 0 đến byte 6 đều có giá trị: 0xAA và byte còn lại chứa giá trị đếm số thứ tự của gói tin được gửi. (0x00-0xFF) đến Node 1.
* Node 1 nhận gói tin từ Node 2 và tăng giá trị của byte 7 lên 1 đơn vị. Sau đó gửi lại gói tin cho Node 2 gồm: byte 0 đến byte 5 có giá trị: 0x55. Byte 6 là giá trị nhận được từ byte 7 của gói tin nhận được tăng lên 1. Byte 7 là checksum của các dữ liệu byte0-byte6.
* Check sum được tính theo chuẩn CRC 8 SAE J1850 như được thể hiện trong **Hình 5**.
* Chu kỳ gửi gói tin của CAN1 là 200ms và chu kỳ gửi gói tin của CAN2 là 500 ms.
* Sinh viên sử dụng cổng UART để hiển thị gói tin và các LEDs để hiển thị trạng thái kiểm tra checksum sau khi nhận được gói tin.



**Hình 5 Giải thuật tính checksum theo chuẩn CRC 8 SAE J1850**

1. **Kiểm tra, theo dõi và nhận diện dữ liệu CAN trên đường truyền**

Sinh viên kết nối chân TXD và RXD của một trong hai node CAN controller với bộ dao động ký và vẽ lại dạng sóng của dữ liệu được thể hiện trên bộ dao động ký như trên **Hình 7**.

**Hình 6** thể hiện dữ liệu thu được từ máy dao động ký gồm có 4 kênh: kênh TXD và kênh RXD của Node thu dữ liệu, kênh TXD và kênh RXD của Node truyền. Sinh viên cho biết các trường dữ liệu của node transmiter có giá trị lần lượt là bao nhiêu? (Giá trị ID, có bao nhiêu byte dữ liệu, giá trị của các byte dữ liệu)



**Hình 6 Dữ liệu CAN thu được từ máy dao động ký**

Graphical user interface

Description automatically generated

**Hình 7 Gói tin mẫu thu được từ máy dao động ký**

1. **Báo cáo thực hành**

Sinh viên viết báo cáo theo mẫu đã được cung cấp để trình bày lại các kết quả đã làm được trong bài thực hành. Giải thích các đoạn chương trình mà sinh viên đã hiện thực để giải quyết các yêu cầu của bài thực hành.

|  |  |
| --- | --- |
| [1] | "CAN Specification," Robert Bosch GmbH, Stuttgart, 1991. |
| [2] | [Online]. Available: https://www.waveshare.com/wiki/Open405R-C. [Accessed 2022]. |
| [3] | "Open405R-C Package A, STM32F4 Development Board," Waveshare Electronics, 2022. [Online]. Available: https://www.waveshare.com/open405r-c-package-a.htm. |

**Tài liệu tham khảo**