ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG

----------oOo----------

BÀI TẬP LỚN TRUYỀN SỐ LIỆU VÀ MẠNG

ĐỀ TÀI

CONTROL AREA NETWORK

GVHD: Nguyễn Khánh Lợi

Lớp: A03

Nguyễn Thắng Anh Quân	1813708
Nguyễn Anh Quốc	1813733
Nguyễn Tấn Sang	1813803
Nguyễn Hoàng Quân	1813705

TP.HCM, Ngày 28 tháng 12 năm 2020

MỤC LỤC

DANH SÁCH HÌNH ẢNH

CHƯƠNG 1. GIỚI THIỆU CHUNG

1.1. Khái niệm

CAN viết tắt của (Controller Area network) mạng điều khiển khu vực hay đơn giản gọi là mạng CAN, là một mạng công nghệ ghép nối tiếp, là giao thức giao tiếp có khung truyền dữ liệu lớn.

1.2. Lịch sử hình thành

- Vào những năm đầu của thập niên 1980, nền công nghiệp xe hơi đang ngày càng phát triển mạnh mẽ. Cùng với sự phát triển đó là sự gia tăng về nhu cầu tự động hóa các thiết bị điện tử (ECU: Electronic Control Unit) được sử dụng trong xe. Khi đó hầu hết các ECU được kết nối với nhau thông qua giao tiếp nối tiếp thông thường như RS232, UART,… Điều này dẫn đến việc kiểm soát hoạt động của các ECU và việc liên kết chúng với nhau trở nên rất phức tạp, kết cấu cồng kềnh và chi phí nối dây cũng không hề nhỏ.
- Ví dụ về việc kết nối các ECU khi CAN chưa ra đời:

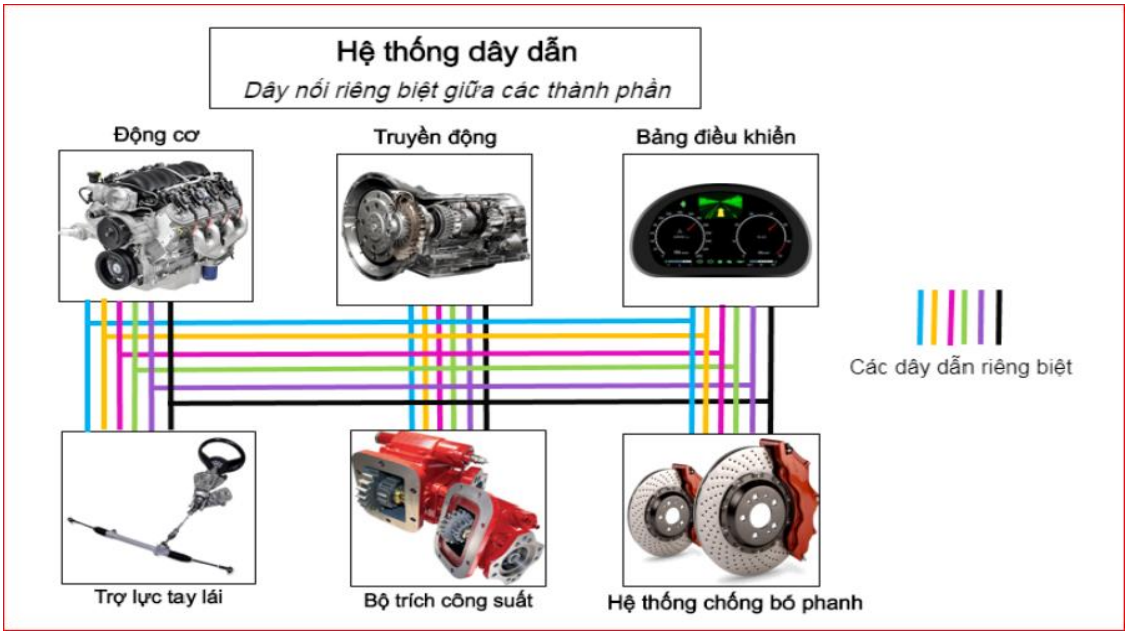


Figure 1.1. Sơ đồ kết nối các ECU trong ô tô trước khi giao thức CAN ra đời

- Để giải quyết vấn đề đó, Công ty Robert Bosch GmbH của Đức đã bắt đầu phát triển giao thức CAN từ nằm 1983 và được chính thức công bố vào năm 1986 ở Hội nghị của hiệp hội Kỹ sư ô tô (Society of Automotive Engineers – SAE) ở Detroit, Michigan, Mỹ.

1.3. Quá trình phát triển đến hiện tại

- Những chip điều khiển CAN đầu tiên được sản xuất bởi Intel và Philips, xuất

hiện trên thị trường vào năm 1987. Năm 1994, CAN chính thức trở thành một tiêu chuẩn quốc tế (ISO 11898) và trở thành một tiêu chuẩn bắt buộc được áp dụng rộng rãi trong ô tô ở châu Âu. Xe Mercedes-Benz W140 là chiếc xe đầu tiên được trang bị CAN.

- Cho tới nay, giao thức CAN vẫn luôn được sử dụng phổ biến cho các phương tiện giao thông, đặc biệt là xe hơi và được sử dụng rộng rãi hơn trong các hệ thống tự động hóa công nghiệp hay hàng không.

CHƯƠNG 2. CẤU TẠO CỦA CAN

2.1. Tổng quan

- CAN là một chuẩn ISO (ISO 11898) cho truyền thông tin tiếp. Giao thức được BOSCH xây dựng vào năm 1980 cho các ứng dụng tự động. Ngày nay CAN đã được sử dụng rộng rãi trong công nghiệp tự động. Chuẩn CAN bao gồm:

- Tầng vật lý.
- Tầng liên kết dữ liệu:
  - Vài loại thông điệp.
  - Các chuẩn phân xử cho truy cập bus.
  - Các phương pháp dò lỗi và giam lỗi.

- CAN là một mạng điều khiển vùng cho phép các thiết bị trong cùng Bus có thể giao tiếp nối tiếp với nhau chỉ thông qua 2 dây nối (CAN – High và CAN – Low). Các thiết bị trong cùng Bus được gọi là các Node (trong xe hơi ta có thể coi chúng là các ECU), chúng có thể lên tới vài chục Node trong phạm vi từ vài trăm mét đến vài kilomet mà vẫn đảm bảo được tốc độ truyền tín hiệu. điều đó tạo nên sự khác biệt của CAN so với các giao thức khác.

- Ví dụ về việc CAN Bus làm cho kết nối giữa các thiết bị trong xe hơi trở nên đơn giản hơn:

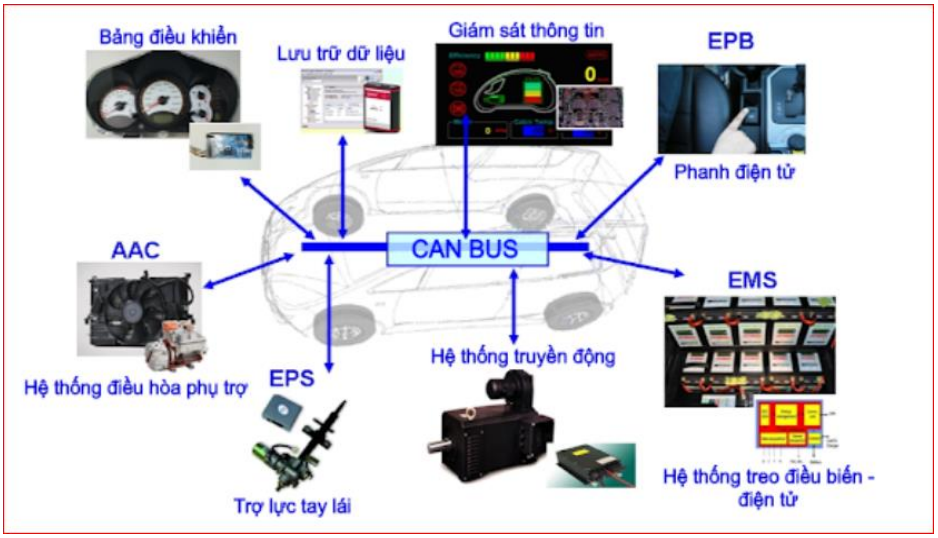


Figure 2.1. Sơ đồ kết nối các ECU trong ô tô khi giao thức CAN ra đời

2.2. Lớp vật lý

2.2.1. None-return-to-zero

Mỗi bit trong mạng CAN được mã hóa bằng phương pháp None-return-to-zero (NRZ method). Trong suốt quá trình của một bit, mức điện áp. của dây được giữ nguyên, có nghĩa trong suốt quá trình một bit được tạo, giá trị của nó giữ không đổi.

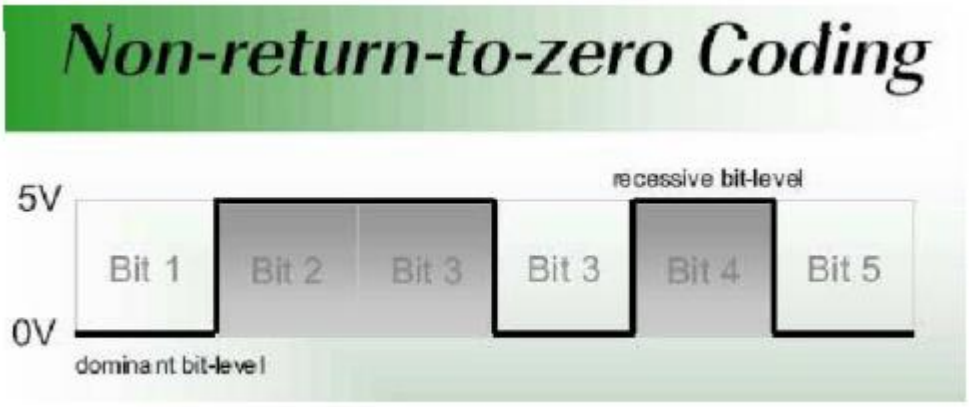


Figure 2.2. NRZ method

2.2.2. Bit Stuffing

Một trong những ưu điểm của cách mã hóa NRZ là mức của bit được giữ trong suốt quá trình của nó. Điều này tạo ra vấn đề về độ ổn định nếu một lượng lớn bit giống nhau nối tiếp. Kỹ thuật Bit Stuffing áp đặt tự động một bit có giá trị ngược lại khi nó phát hiện 5 bit liên tiếp trong khi chuyển.

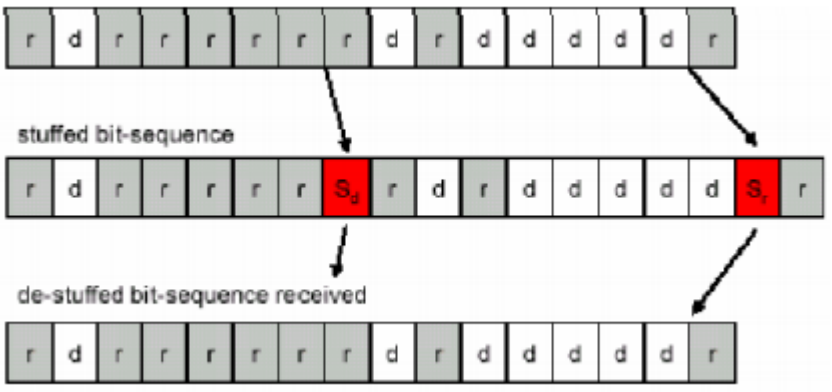


Figure 2.3. Kỹ thuật Bit Stuffing

2.2.3. Bit Timing

Ta định nghĩa thời gian đơn vị nhỏ nhất, là Time Quantum. Thời gian cơ bản này là một phân số của thời gian dao động của bus. Một bit khoảng 8 đến 25 quanta.

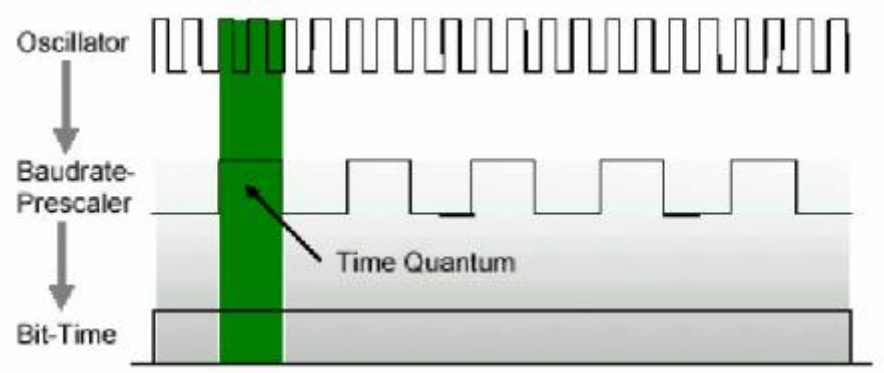


Figure 2.4. Giản đồ thời gian

2.2.4. Độ dài của một bus:

- Độ dài của một bus phụ thuộc vào những thông số sau:

- Độ trễ lan truyền trên đường dây của bus.
- Sự khác nhau của thời gian Time Quantum (định nghĩa ở trên), vì sự khác nhau của xung clock tại các nút.
- Biên độ tín hiệu thay đôi theo điện trở của cáp và tông trở vào của các nút.

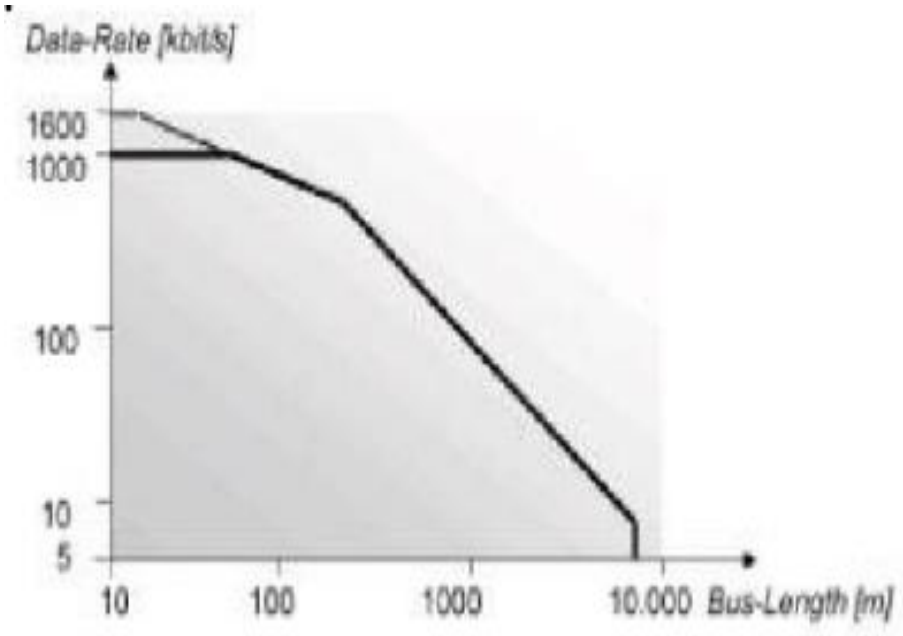


Figure 2.5. Tốc độ tỉ lệ nghịch với độ dài bus

Bit Rate	Bus Length	Nominal Bit-Time
1 Mbit/s	30 m	1 µs
800 kbit/s	50 m	1,25 µs
500 kbit/s	100 m	2 µs
250 kbit/s	250 m	4 µs
125 kbit/s	500 m	8 µs
62,5 kbit/s	1000 m	20 µs
20 kbit/s	2500 m	50 µs
10 kbit/s	5000 m	100 µs

Figure 2.6. Tốc độ bit- độ dài- bit time

- Cần chú ý rằng bất cứ module nào kết nối vào một bus CAN phải được hỗ trợ vớ tốc độ tối thiểu là 20kbit/s. Để sử dụng bus có độ dài hơn 200m, cần thiết phải sử dụng một Optocoupleur và để sử dụng bus dài hơn 1 km, phải cần một hệ thống kết nối trung gian như repeater hoặc bridge.

2.2.5. Trạng thái "Dominant" và "Recessive"

- Ở lớp vật lý, Bus CAN định nghĩa hai trạng thái là "Dominan" và "Recessive" tương ứng với hai trạng thái là 0 và 1. Trạng thái "Dominant" chiếm ưu thế so với trạng thái "Recessive". Bus chỉ ở trạng thái "Reccessive" khi không có node nào phát đi trạng thái "Dominant". Điều này tạo ra khả năng giải quyết chanh chấp khi nhiều hơn một Master cùng muốn chiếm quyền sử dụng bus. Bởi tính chất vật lý của bus, cần thiết phải phân biệt 2 dạng truyền:

- Truyền CAN low speed
- Truyền CAN high specd

Bảng sau tổng kết những tính chất cơ bản khác nhau giữa 2 dạng, đặc biệt là tốc độ:

Thông số	CAN low speed	CAN high speed
Tốc độ	125kb/s	125kb/s tới 1 Mb/s
Số nút trên bus	2 tới 20	2 tới 30

Trạng thái dominant	CAN H = 4V CAN L = 1V	CAN H = 3.25V CAN L = 1.5V
Trạng thái recessive	CAN H = 1.75V CAN L = 3.25V	CAN H = 2.5V CAN L = 2.5V
Tính chất của cáp	30pF giữa cáp và dây	2*120 Ohm
Mức điện áp cung cấp	5V	5V

- So sánh CAN low speed và CAN high speed:

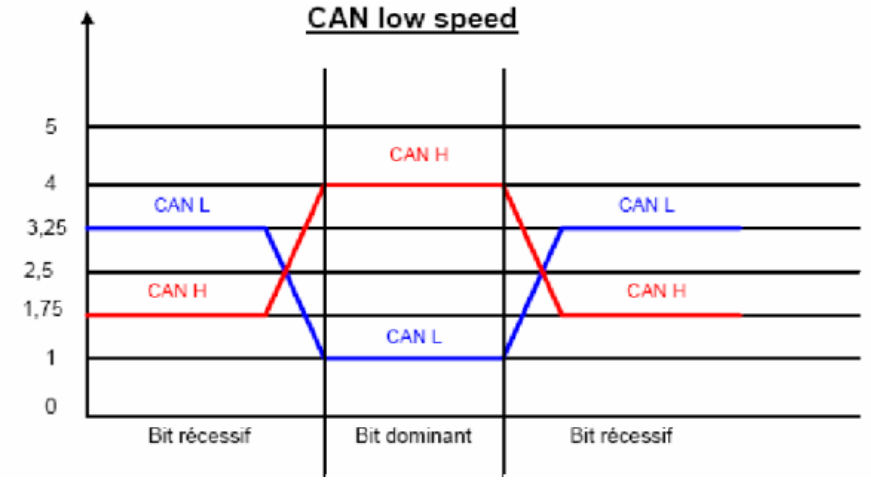


Figure 2.7. CAN low speed

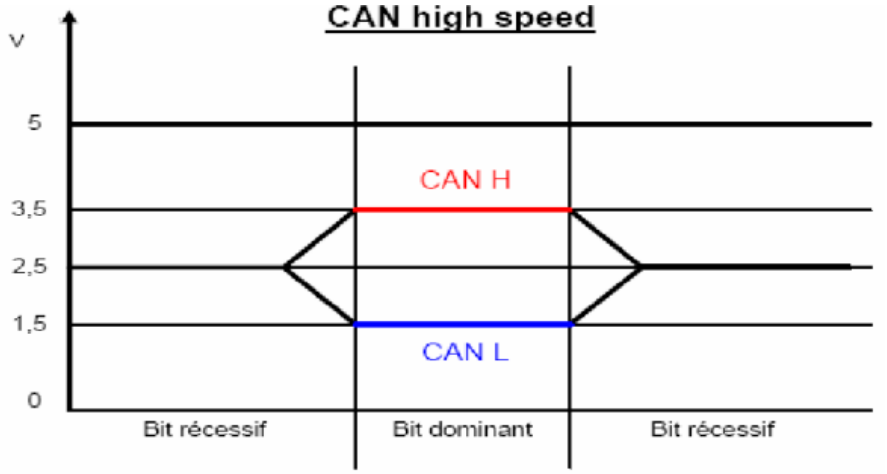


Figure 2.8. CAN high speed

- Vì tính chất vi sai trên đường truyền tín hiệu của bus CAN, sự miễn trừ tác động điện từ được bảo đảm vì 2 dây của bus đều bị tác động như nhau cùng một lúc bởi tín hiệu nhiễu.

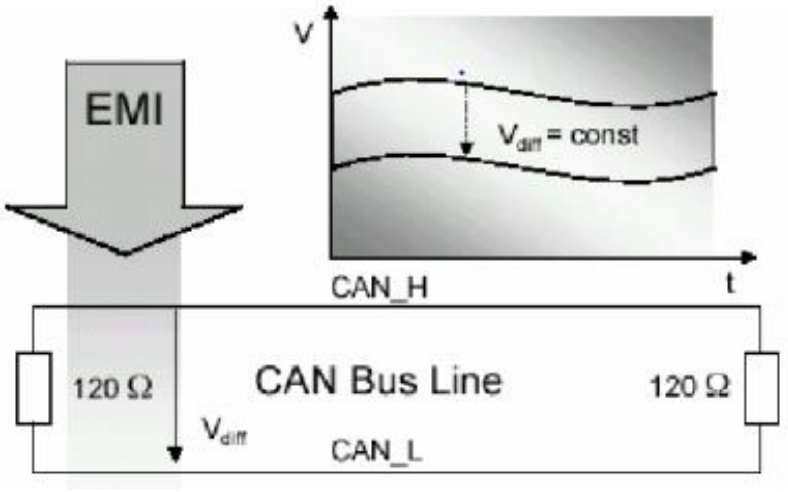


Figure 2.9. Sự kháng nhiễu với ảnh hưởng của điện từ

2.3. Cơ chế giao tiếp

Đặc trưng của CAN là phương pháp định địa chỉ và giao tiếp hướng đối tượng, trong khi hầu hết các hệ thống bus thường khác đều giao tiếp dựa vào địa chỉ các node. Mỗi thông tin trao đổi trong mạng được coi như một đối tượng, được gắn một mã số căn cước. Thông tin được gửi trên bus theo kiểu truyền thông báo với độ dài có thể khác nhau. Các thông báo không được gửi tới một địa chỉ nhất định mà bất cứ node nào cũng có thể nhận theo nhu cầu. Nội dung mỗi thông báo được các node phân biệt qua một mã căn cước (IDENTIFIER). Mã căn cước không nói lên địa chỉ đích của thông báo, mà chỉ biểu diễn ý nghĩa của dữ liệu trong thông báo. Vì thế, mỗi node trên mạng có thể tự quyết định tiếp nhận và xử lý thông báo hay không tiếp nhận thông báo qua phương thức lọc thông báo (message filtering). Cũng nhờ sử dụng phương thức lọc thông báo, nhiều node có thể đồng thời nhận cùng một thông báo và có các phản ứng khác nhau. Một node có thể yêu cầu một node khác gửi dữ liệu bằng cách gửi 1 khung REMOTE FRAME. Node có khả năng cung cấp nội dung thông tin đó sẽ gửi trả lại một khung dữ liệu DATA FRAME có cùng mã căn cước với khung yêu cầu. Bên cạnh tính năng đơn giản, cơ chế giao tiếp hướng đối tượng ở CAN còn mang lại tính linh hoạt và tính nhất quán dữ liệu của hệ thống. Một node CAN không cần biết thông tin cấu hình hệ thống (ví dụ địa chỉ node), nên việc bổ sung hay bỏ đi

một node trong mạng không đòi hỏi bất cứ một sự thay đổi nào về phần cứng hay phần mềm ở các node khác. Trong một mạng CAN , có thể chắc chắn rằng một thông báo hoặc được tất cả các node quan tâm tiếp nhận đồng thời, hoặc không được node nào tiếp nhận, tính nhất quán dữ liệu được đảm bảo qua các phương pháp gửi đồng loạt và xử lý lỗ.

2.4. Giải quyết tranh chấp trên bus

Phương thức giao tiếp của bus CAN là sự phát tán thông tin (broadcast): mỗi điểm kết nối vào mạng thu nhận fame truyền từ nút phát. Sau đó, mỗi nút sẽ quyết định việc xử lý message, có trả lời hay không, có phản hồi hay không... Cách thức này giống như sự phát thông tin về đường đi của một node phát thanh: khi nhận được thông tin về đường đi, người lái xe có thể thay đổi lộ trình của anh ta, dừng xe hay thay đổi tài xế hoặc chẳng làm gì cả… Giao thức CAN cho phép các nút khác nhau đưa dữ liệu cùng lúc và một quá trình nhanh chóng, ổn định của cơ chế arbitration sẽ xác định xem nút nào được phát đầu tiên. Để xử lý thời gian thực, dữ liệu phải được truyền nhanh. Điều này ảnh hưởng không chỉ đường truyền vật lý cho phép tới IMbits, mà còn đòi hỏi một sự cập phát nhanh bus trong trường hợp xung đội, khi mà rất nhiều nút muốn truyền đồng thời. Khi trao đổi dữ liệu trên bus, thứ tự sẽ được xác định dựa vào loại thông tin. Ví dụ, các giá trị hay biến đổi nhanh, như trạng thái của một cảm biến, hay phản hồi của một động cơ, phải được truyền liên tục với độ trễ thấp nhất, hơn là các giá trị khác như nhiệt độ của động cơ, các giá trị thay đổi ít. Trong mạng CAN, bộ phận ID của mỗi InesSage, là một phần tử gồm 11 bit (version 2.0A) xác định mức ưu tiên. Phần ưu tiên này nằm ở đầu mỗi message. Mức ưu tiên được xác định bởi 7 bit cho verdion 2.0A, tới 127 mức và mức 128 là 0000000 theo NMT (Network Management). Quy trình arbitration của bus dựa trên phân giải từng bit, theo những nút đang tranh chấp, phát đồng thời trên bus. Nút nào mức ưu tiên thấp hơn sẽ mất sự cạnh tranh với nút có mức ưu tiên cao.

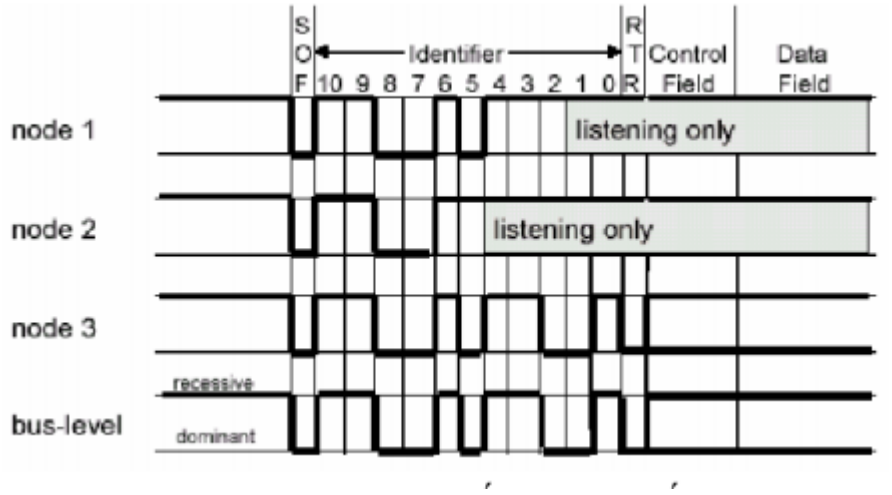


Figure 2.10. Giải quyết tranh chấp trên bus

2.5. Cấu trúc "CAN frame"

- Một khung truyền có dạng sau:

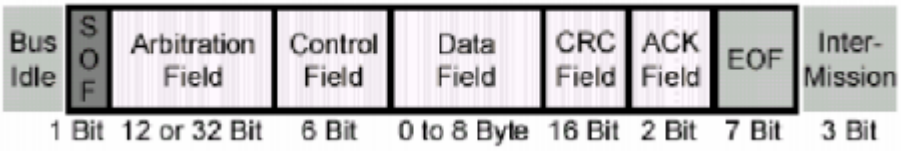


Figure 2.11. Khung truyền

- Chuẩn CAN định nghĩa bốn loại Frame: Data frame dùng khi node muốn truyền dữ liệu tới các node khác. Remote frame dùng để yêu cầu truyền data frame. Error frame và Overload frame dùng trong việc xử lý lỗi.

- **Data frame:** dùng để truyền đi một message. Có hai dạng: standard frame và extended frame.

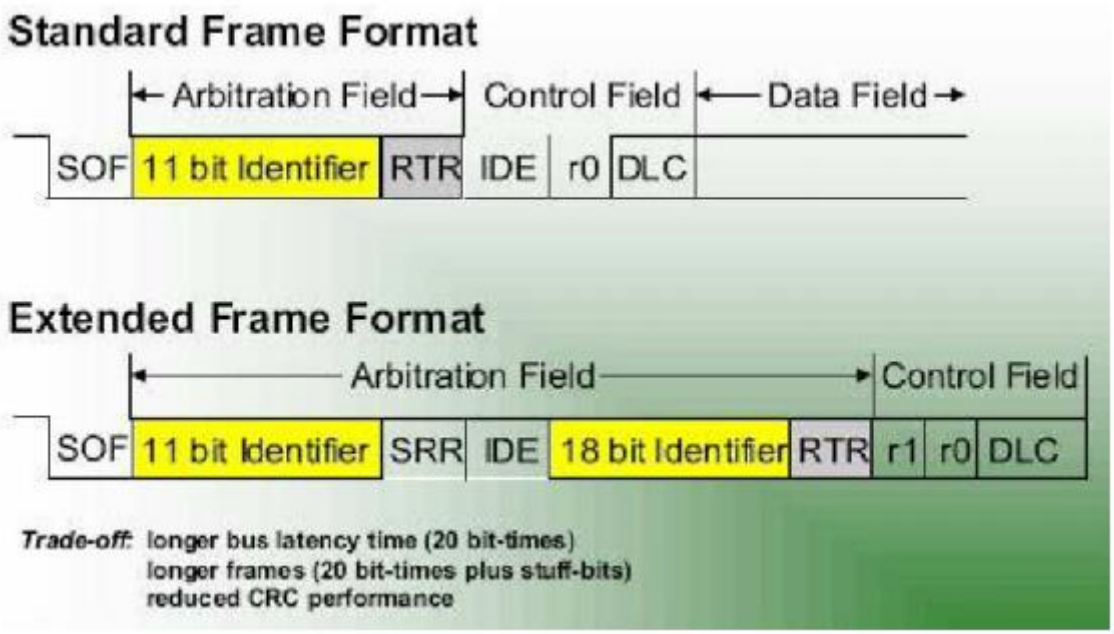


Figure 2.12. CAN Data frame

- **Standard frame**: bắt đầu bằng 1 bit start of frame (SOE) luôn ở trạng thái dominant, 11 bit ID tiếp theo, 1 bit Remote Transmit Request (RTR) để phân biệt remote frame và data frame nếu bằng dominant nghĩa là data frame, nếu bằng recessive nghĩa là remote frame. Tiếp đến là 1 bit Identifier Extension để phân biệt giữa Standard frame ("dominant") và Extended frame ("recessive"). Tiếp theo là 1 bit r_0 luôn ở trạng thái dominant. Tiếp đến là 3 bit Data Length Control cho biết số lượng byte data của frame. Tiếp đến là 0 đến 8 bytes data. Tiếp đến là 15 bit CRC và bit CRC delimiter. Tiếp đến là 1 bit Acknoledge và 1 bit delimiter, tiếp theo là 7 bit End of frame luôn ở trạng thái recessive. Cuối cùng là khoảng cách tối thiểu giữa hai frame truyền inter-frame space (IEFS).

Figure 2.13. CAN standard frame

- **Extended frame**: gần giống như standard frame và có 29 bit ID:

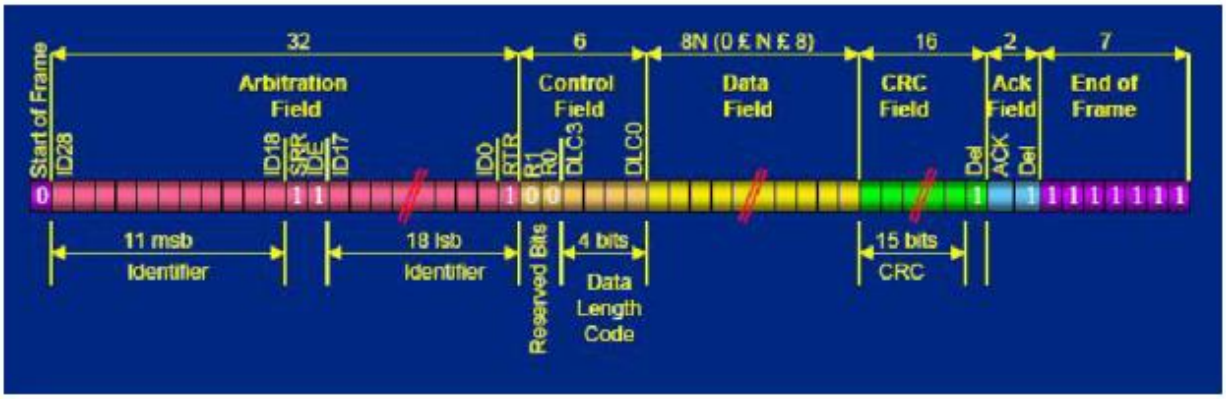


Figure 2.14. CAN extended frame

Chi tiết các phần khác nhau trong một khung truyền dữ liệu:

- **Start of frame:** Nắm phần đầu của một frame dữ liệu hay Remote frame, luôn ở trạng thái dominant. Một nút có thể bắt đầu truyền dữ liệu nếu bus rảnh. Sau đó tất cả các nút đều đồng bộ sau SOF của nút bắt đầu truyền.
- **CRC Field:** bao gồm một chuỗi gồm 15 bit và CRC Delimiter (là 1 bit recessive)

Một chuỗi CRC (Cyclic Redundancy Code) cho phép kiểm tra sự nguyên vẹn của dữ liệu truyền. Tất cả các nút nhận phải thực hiện quy trình kiểm tra này. Chỉ vùng SOF, vùng tranh chấp, vùng điều khiển và vùng dữ liệu được sử dụng để tính toán chuỗi CRC. Trên thực tế, độ dài cực đại của frame không vượt quá 2^{15} bit cho một chuỗi CRC 15 bit.

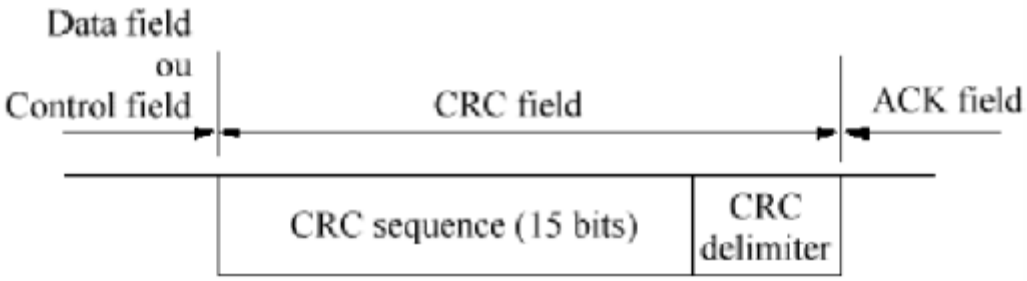


Figure 2.15. CRC field

- **ACK Field**: Gồm 2 bit: ACK slot và ACK Delimiter (là 1 bit recessive)
  - Một nút đang truyền sẽ gửi một bit recesive trong ACK slot.
  - Một nút nhận đúng message thông báo cho nút truyền sẽ gửi 1 bit dominant trong ACK slot.
- **Remote frame(khung yêu cầu dữ liệu):** dùng để yêu cầu truyền data frame

tới một nút khác. Gần giống data frame nhưng có DLC=0 và không có data field.

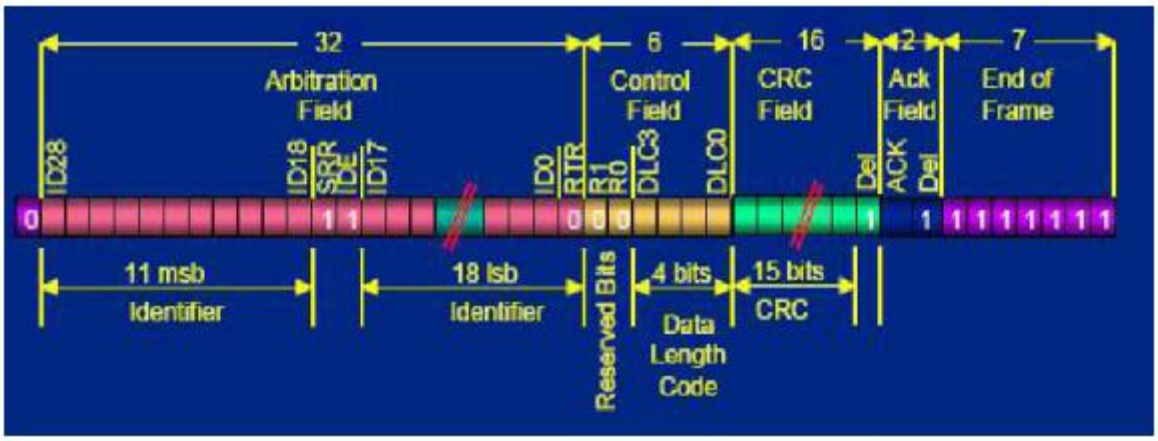


Figure 2.16. CAN remote frame

- **Erro frame:** được phát ra khi node phát hiện lỗi.

Frame lỗi bao gồm 2 phần:

- Cờ lỗi.
- Phần delimiter Overload frame: Dùng khi frame bị tràn bộ đệm, nhằm tạo một khoảng cách thời gian bổ sung giữa 2 khung dữ liệu hoặc yêu cầu dữ liệu trong trường hợp một node bị quá tải.

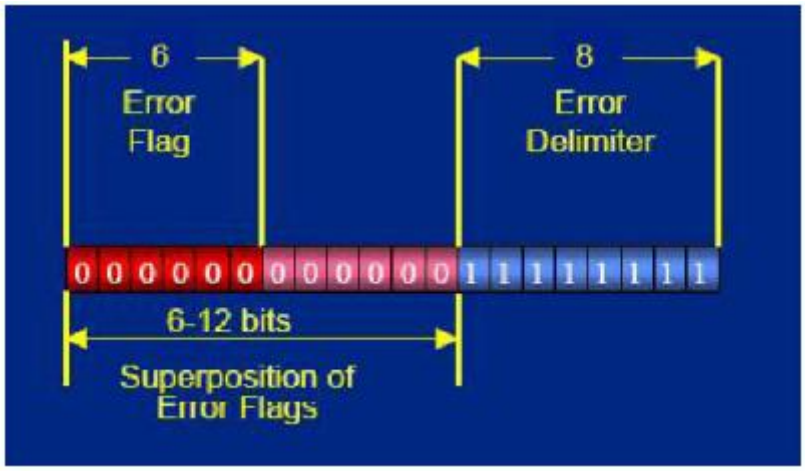


Figure 2.17. CAN error frame

2.6. Nominal Bit Time

- Nominal Bit Time là độ dài của một bit trên bus. Mỗi nút trên bus phải điều chỉnh nhịp cùng với Nominal Bit Time để có thể phát và nhận chính xác dữ liệu trên bus.

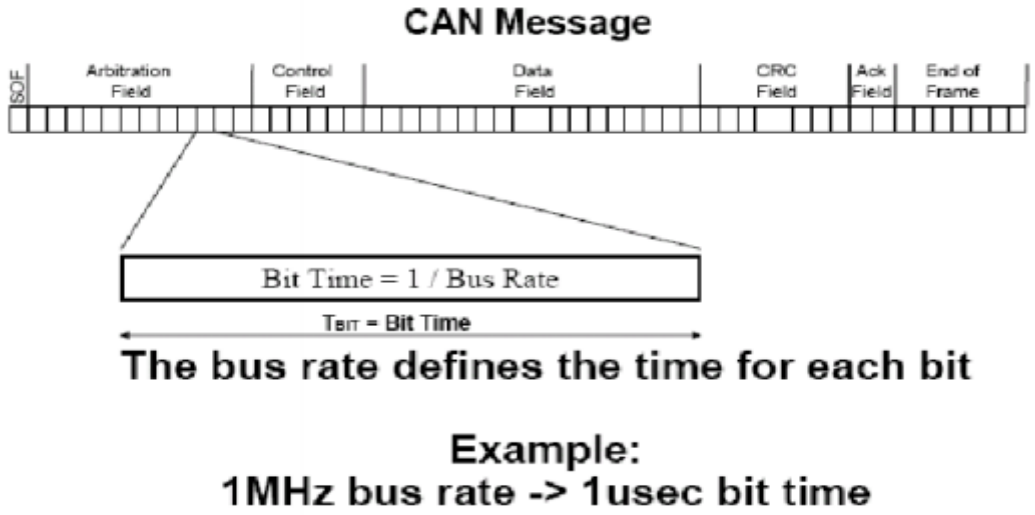


Figure 2.18. Baudrate định nghĩa thời gian cho 1 bit

- Chuẩn BOSCH mô tả thành phần của Nominal Bit Time, được chia ra thành nhiều đoạn (segmernt):

- Đoạn đồng bộ (SYNC_SEG)
- Đoạn lan truyền (PROG_SEG)
- Đoạn pha buffer I (PHASE_SEG1)
- Đoạn pha buffer 2 (PHASE_SEG2)

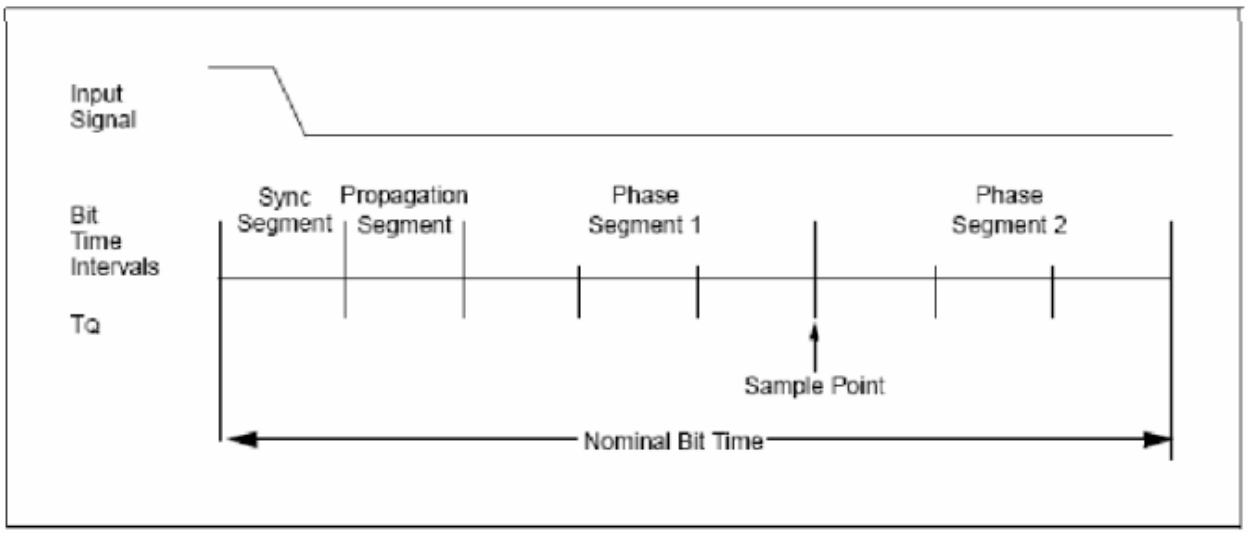


Figure 2.19. Mỗi bit được cấu tạo bởi 4 segments

- Nominal Bit Time, tính theo giây, là nghịch đảo của dung lượng trên bus:

$$Nominal\_Bit\_Time = \frac{1}{Nominal\_Bit\_Rate}.$$

2.7. Sự đồng bộ xung clock

Mỗi nút phải tạo một thời gian danh nghĩa Bit Time để có thể nhận và phát dữ liệu xuống bus với sự đồng bộ các nút khác. Thực tế, nếu Nominal Bit Time của mỗi nút không được đồng bộ với nhau, giá trị đọc từ bus tại thời điểm lấy mẫu có thể không là giá trị đúng với thời điểm mong muốn. Độ trễ này có thể làm ảnh hưởng trong nút nhận frame, khi mà có ít thời gian tính toán CRC và gởi 1 bit dominant trong ACK Slot để xác nhận răng frame đã đúng.

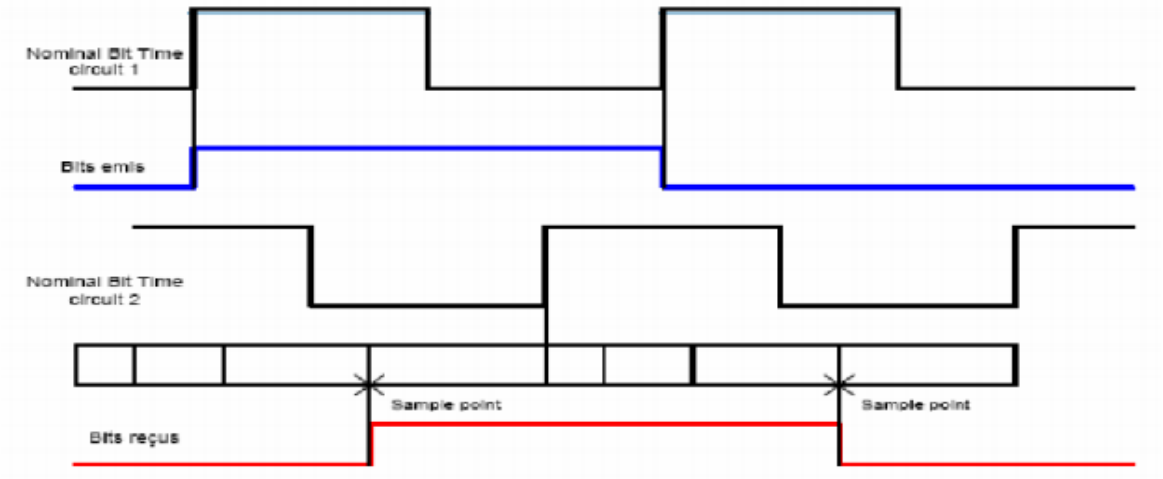


Figure 2.20. Vấn đề đồng bộ

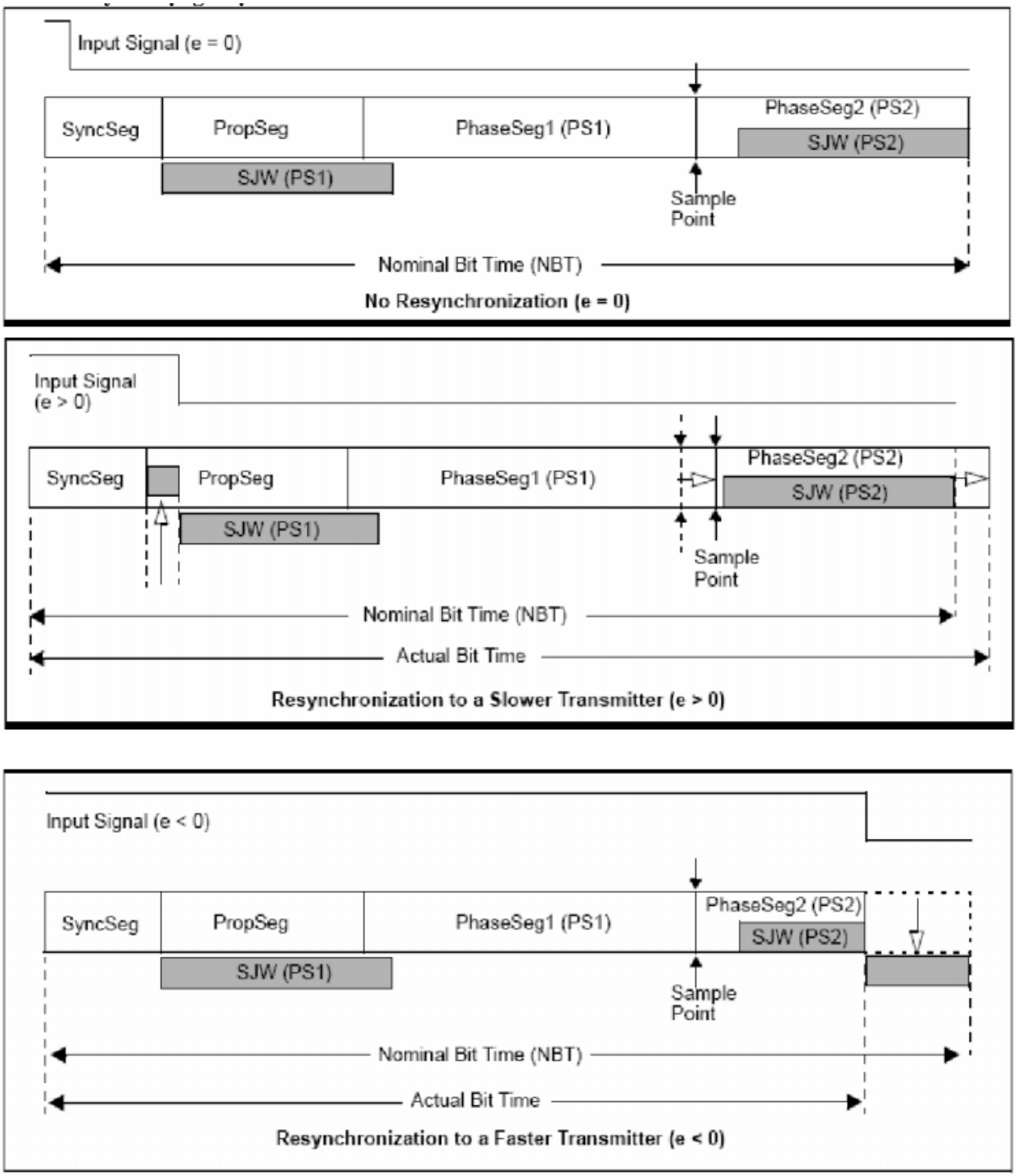


Figure 2.21. Lấy mẫu bit

2.7.1. Cơ chế đồng bộ Đồng bộ cứng (Hard Synchronization):

Chỉ xảy ra khi chuyển cạnh bit đầu tiên từ recessive thành dominant (logic ”1” thành “0”) khi bus rảnh, báo hiệu 1 Start of Frame (SOF). Đồng bộ cứng làm cho bộ đếm bit timing khởi động lại, gây nên một chuyển cạnh trong SyncSeg. Tại thời điểm này, mọi nút

nhận sẽ đồng bộ với nút phát. Đồng bộ cứng chỉ xảy ra một lần trong suốt một message. Và đồng bộ lại có thể không xảy ra trong cùng một bit (SOF) khi mà đồng bộ cứng đang xảy ra.

2.7.2. Cơ chế đồng bộ lại (Resynchronizafion):

Được thực hiện để bảo toàn sự đồng bộ đã thực hiện bởi đồng bộ cứng. Thiếu đồng bộ lại, nút nhận không thể có được sự đồng bộ vì sự lệch pha của các bộ dao động tại mỗi nút. Sự tính toán và mức độ đồng bộ lại được đưa ra từ giá trị sai số pha e và cũng phụ thuộc vào giá trị SJW:

- Nếu sai số pha e bằng 0 (c=0, chuyển cạnh trong Sync Seg), cơ chế đồng bộ lại cũng giống như đồng bộ cứng.
- Nếu sai số pha e dương và bé hơn giá trị tuyệt đối SJW, PHASE.SEG 1 sẽ kéo dài thêm 1 đoạn SJW
- Cuối cùng, nếu sai số pha e âm nhưng lớn hơn giá trị SJW về tuyệt đối, PHASE_SEG 2 sẽ ngănlại 1 đoạn SJW.

Lỗi pha	Tác động lên PHASE_SEG 1	Tác động lên PHASE_SEG 2
$0 < e < \text{SJW}$	Kéo dài thêm e	
$e < 0$ và $\lvert e \rvert < \text{SJW}$		Làm ngắn 1 đoạn e
$e > 0$ và $e > \text{SJW}$	Kéo dài thêm SJW	
$e < 0$ và $\lvert e \rvert > \text{SJW}$		Làm ngắn 1 đoạn SJW

Bảng tóm tắt kết quả cơ chế

2.8. Truyền nhận message

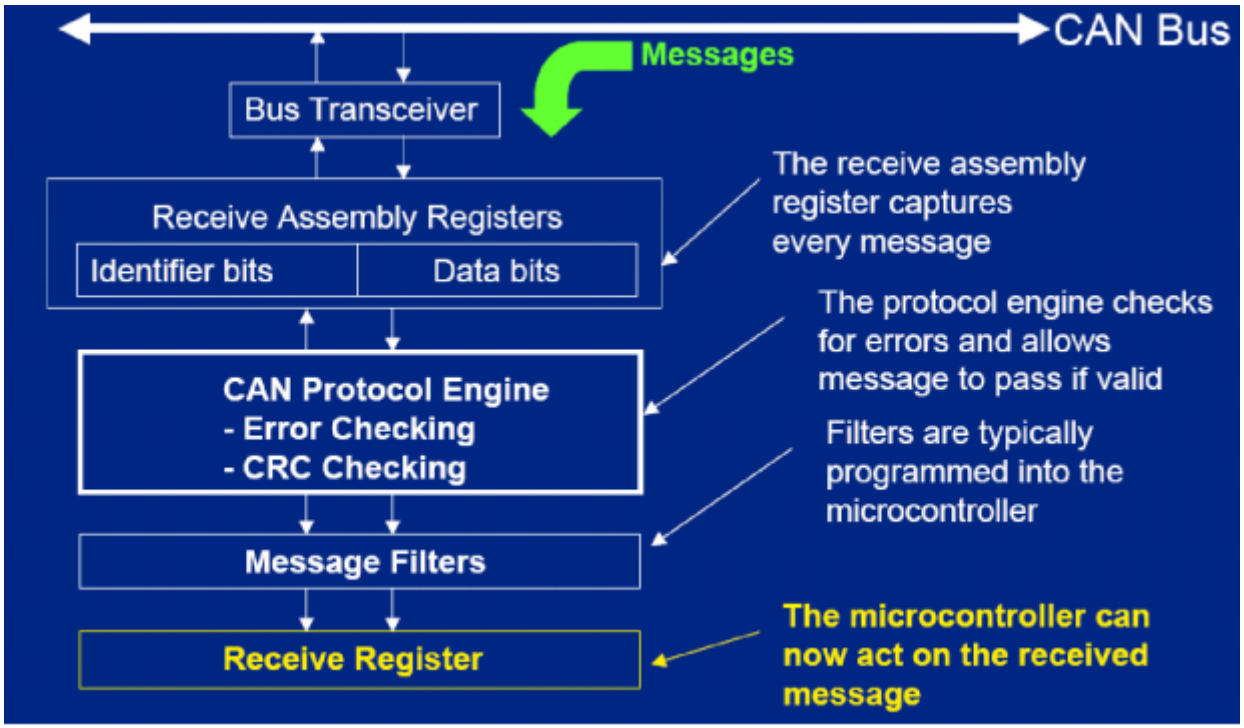


Figure 2.22. Sơ đồ khối bộ nhận CAN message

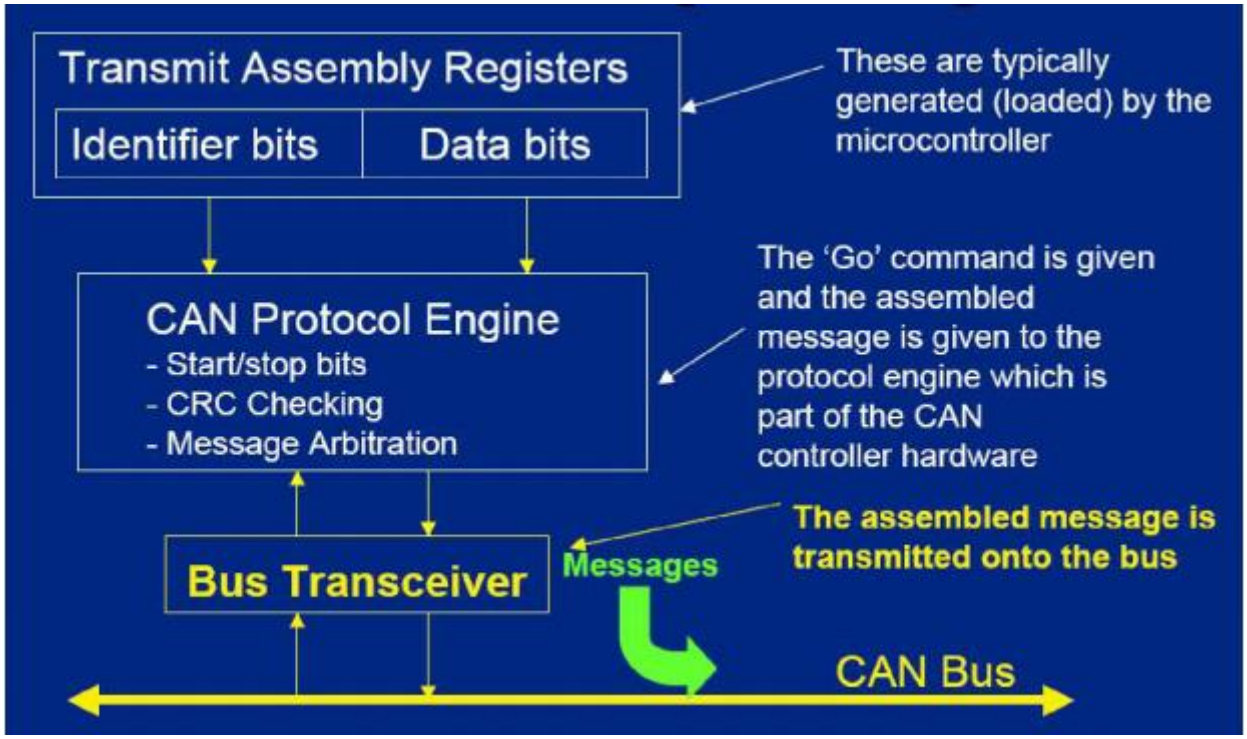


Figure 2.23. Sơ đồ khối độ truyền CAN message

2.9. Xử lý lỗi

- Khi truyền một frame trên bus, lỗi truyền có thể ảnh hưởng đến hoạt động của các nút trên bus. Lỗi có thể đến từ một nút, làm cho mạng không còn hoạt động chính xác. Vì vậy, nhiều cách phát hiện lỗi được sử dụng trong CAN
- Các loại lỗi:

- Bit Error: mỗi khi nút truyền gửi một bit xuống bus, nó kiểm tra xem mức điện áp trên bus có đúng với bit cần gửi hay không. Nếu không đúng, nó sẽ báo hiệu bằng một Bit Error.

Tuy nhiên, Bit Error sẽ không báo hiệu trong những trường hợp sau:

- Không có Bit Error nào được tác động khi một bit dominant được gửi trong vùng ID thay thế cho một bit recessive. Cũng như vậy, trong vùng ACK Slot, thay cho một bit recessive.
- Một nút phát gửi một cờ lỗi (bit recessive) và nhận bit dominant, ko cần phải báo hiệu Bit Error.

- Lỗi Stuffing (Stuff Error): Một lỗi Stuffing được phát hiện trong mỗi lần có 6 bit hay nhiều hơn liên tục trên một đường dây của Bus. Tuy nhiên, lỗi Stuffing sẽ không báo trong vùng ID, vùng điều khiển và vùng CRC. Cơ chế Bit Stuffing không áp dụng sau CRC. Trong mọi trường hợp, lỗi Bit-Stuffing sẽ không báo trong đoạn kết thúc của Frame hay trong vùng ACK.
- Lỗi Cyclic Redundancy (CRC Error): Nếu giá trị CRC tính toán bởi nút nhận không giống với giá trị gửi đi bởi nút phát sẽ có một lỗi CRC(CRC Error).
- Lỗi ACK Delimiter: Một lỗi ACK Delimiter được báo khi nút nhận không thấy một bit recessive trong vùng ACK Delimter hay trong vùng CRC Delimiter.
- Lỗi Slot ACK (ACK Error): Một lỗi Slot ACK được báo bởi nút phát khi nó không đọc thấy bit dominant trong vùng Slot ACK.

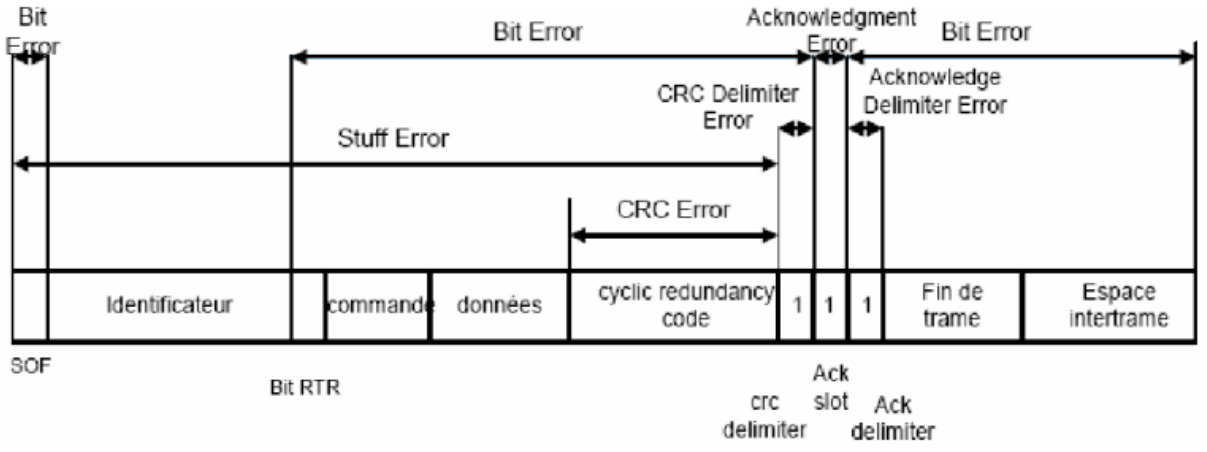


Figure 2.24. Các loại lỗi khác nhau

CHƯƠNG 3. MÔ PHỎNG BẰNG MCU

3.1. Chuẩn bị phần cứng

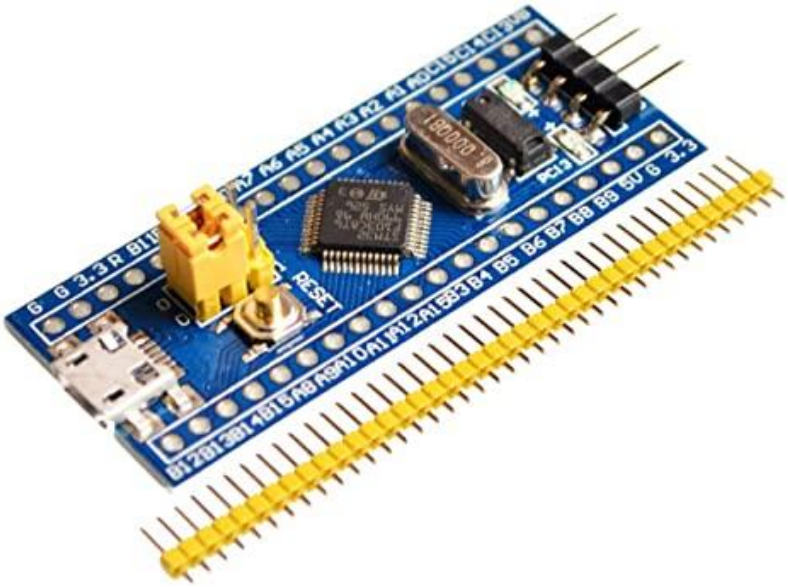


Figure 3.1. MCU STM32F103C8T6

- Bao gồm ngoại vi bxCAN (Basic Extended CAN):
- Hỗ trợ chuẩn CAN 2.0A và B
- Bit rate tối đa đạt 1Mb/s
- Hỗ trợ ngắt (transmit, FIFO, error and status)

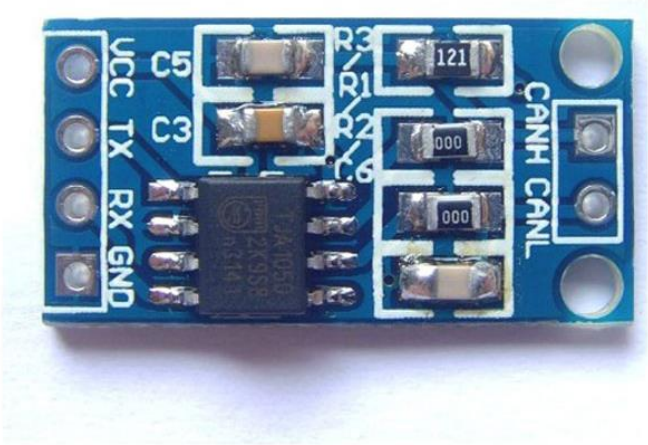


Figure 3.2. Transceiver TJA 1050

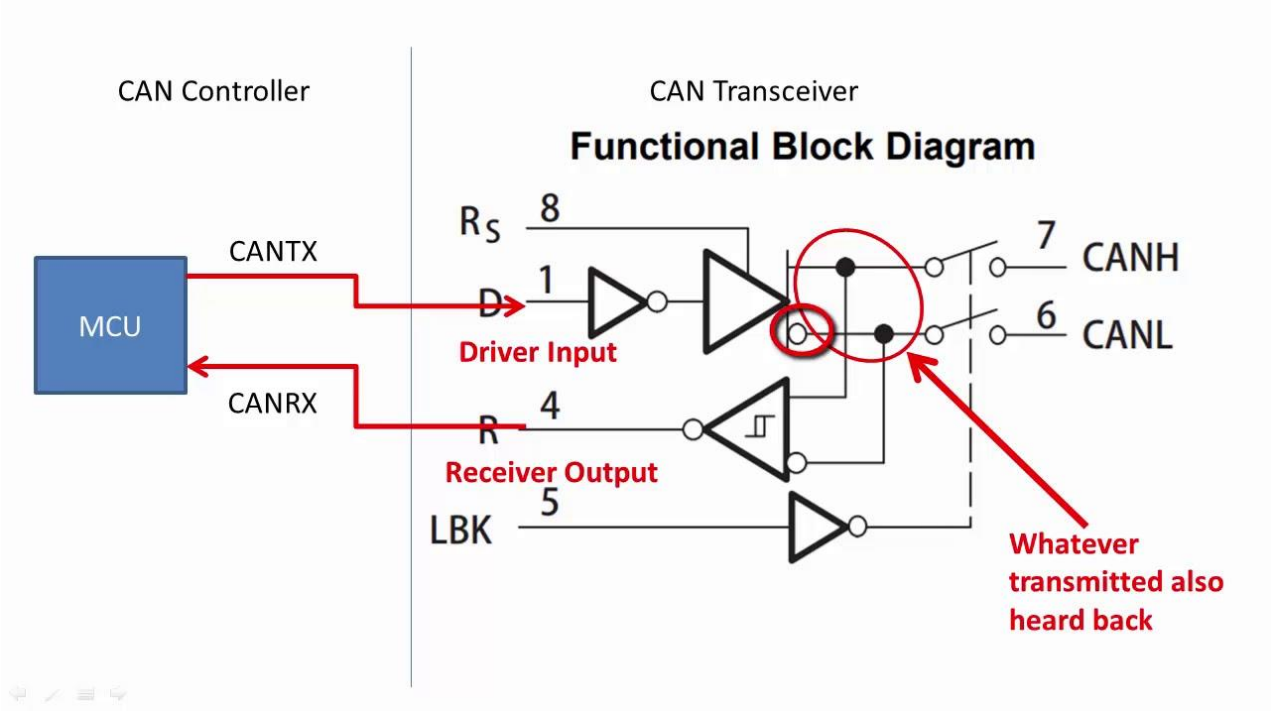


Figure 3.3. Sơ đồ khối của Transceiver

- Bộ điều khiển CAN sẽ được kết nối với bộ chuyển đổi CAN – Transceiver thông qua 2 đường CAN Tx và CAN Rx. Chức năng của bộ chuyển đổi này là để chuyển đổi từ tín hiệu số trên đường TX thành tín hiệu vi sai trên bus CAN (CAN_H và CAN_L) và chuyển đổi từ tín hiệu vi sai trên bus CAN thành tín hiệu số trên RX.

Figure 3.4. Saleae logic analyser

- Sử dụng để đo và hiển thị tín hiệu Digital lên máy tính qua cổng giao tiếp USB.
- Thông số kỹ thuật:

- 8 kênh nhận và xử lý tín hiệu số (Digital) khoảng điện áp từ 0~5.5VDC, tín hiệu

lớn hơn 1.5VDC là mức High, nhỏ hơn 1.5VDC là mức Low.

- Tần số lấy mẫu tối đa: 25Mhz.

3.2. Cài đặt các thông số

3.2.1. Bit timing:

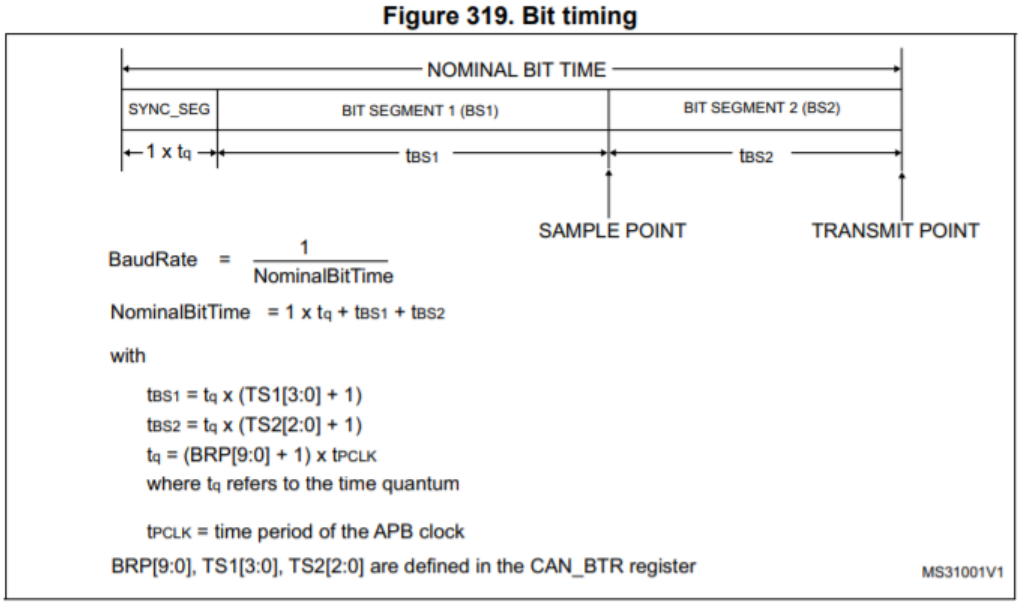
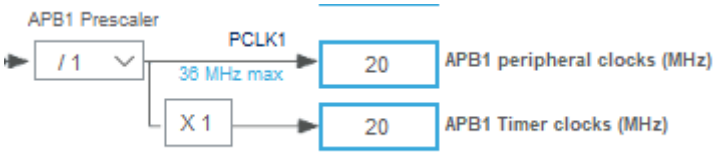


Figure 3.5. Bit timing

- Trong đó t_q là thời gian lượng tử hóa, có giá trị bằng (tần số APB1/prescaler)$^{-1}$

được setup.



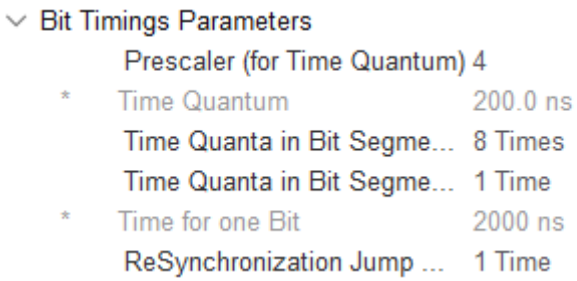


Figure 3.6. Tần số của ngoại vi cài đặt trên MCU

- Như vậy $t_q=(20*10^6 / 4)^{-1} = 200ns$

- Chọn tBS1 = 8*t_q và tBS2 = t_q ta sẽ có Bit time là t_q + tBS1 + tBS2 = 2000ns. Và bit sẽ được lấy mẫu tại 90% thời gian đó.

3.2.2. Xử lý truyền dữ liệu:

- Để truyền đi một mẫu tin, cần chọn một mailbox rỗng, cài đặt các thông số như mã căn cước (Id), chiều dài dữ liệu (DLC), chế độ của Identifier, có yêu cầu truyền lại không.

```
TxHeader.DLC = 5;
TxHeader.StdId = 0x65D;
TxHeader.IDE = CAN_ID_STD;
TxHeader.RTR = CAN_RTR_DATA;
TxHeader.TransmitGlobalTime = DISABLE;
```

Figure 3.7. Khởi tạo việc truyền dữ liệu

- Đưa data vào mailbox. Tích cực bit TXRQ để thực hiện yêu cầu truyền. Lúc đó mailbox sẽ vào trạng thái chờ, đến khi đạt được mức ưu tiên và CAN bus rảnh (idle) để phát.

- Sau khi phát xong phần cứng sẽ báo bằng các tích cực 2 bit RQCP và TXOK.

```
void CAN_Transmit(void)
{
    uint8_t TxData[5] = {'B', 'K', 'T', 'E', 'L'};
    HAL_CAN_AddTxMessage(&hcan, &TxHeader, TxData, &TxMailbox);
    while(HAL_CAN_IsTxMessagePending(&hcan, TxMailbox));
}
```

Figure 3.8. Tiến hành truyền dữ liệu

3.2.3. Kết quả:

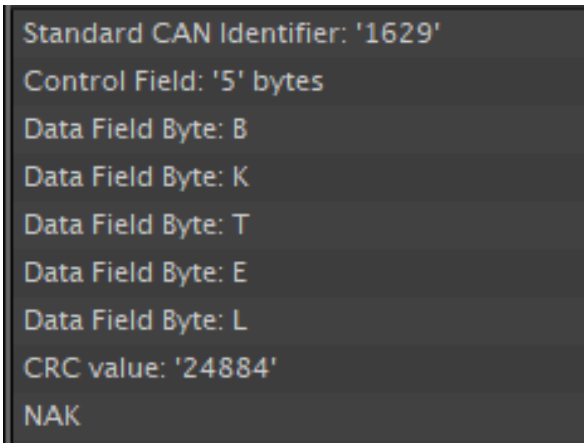


Figure 3.9. Phần mềm giải mã thành công gói dữ liệu truyền đi.

Figure 3.10. Tín hiệu trên dây CANH đo được trên oscilloscope

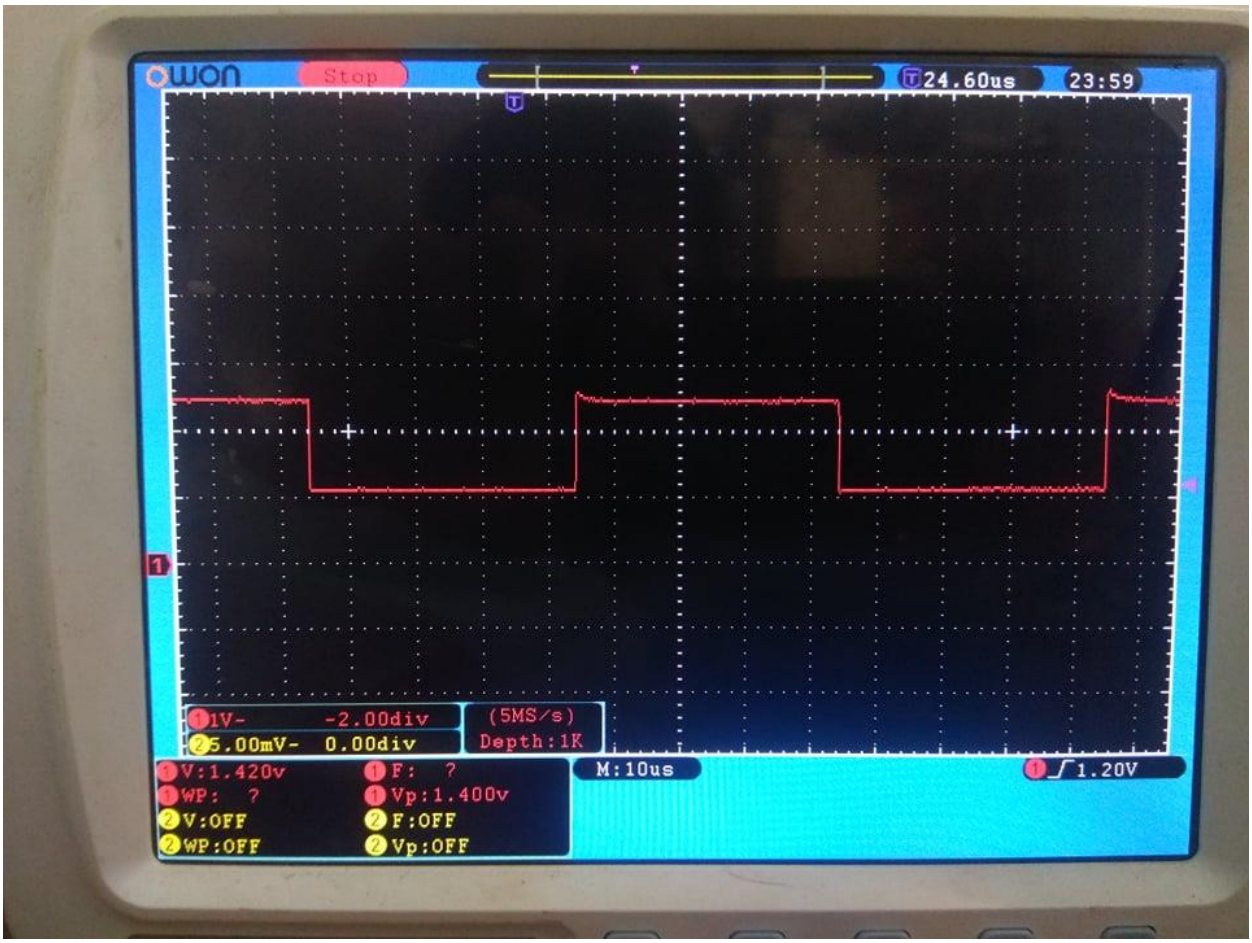


Figure 3.11. Tín hiệu trên dây CANL đo được trên oscilloscope

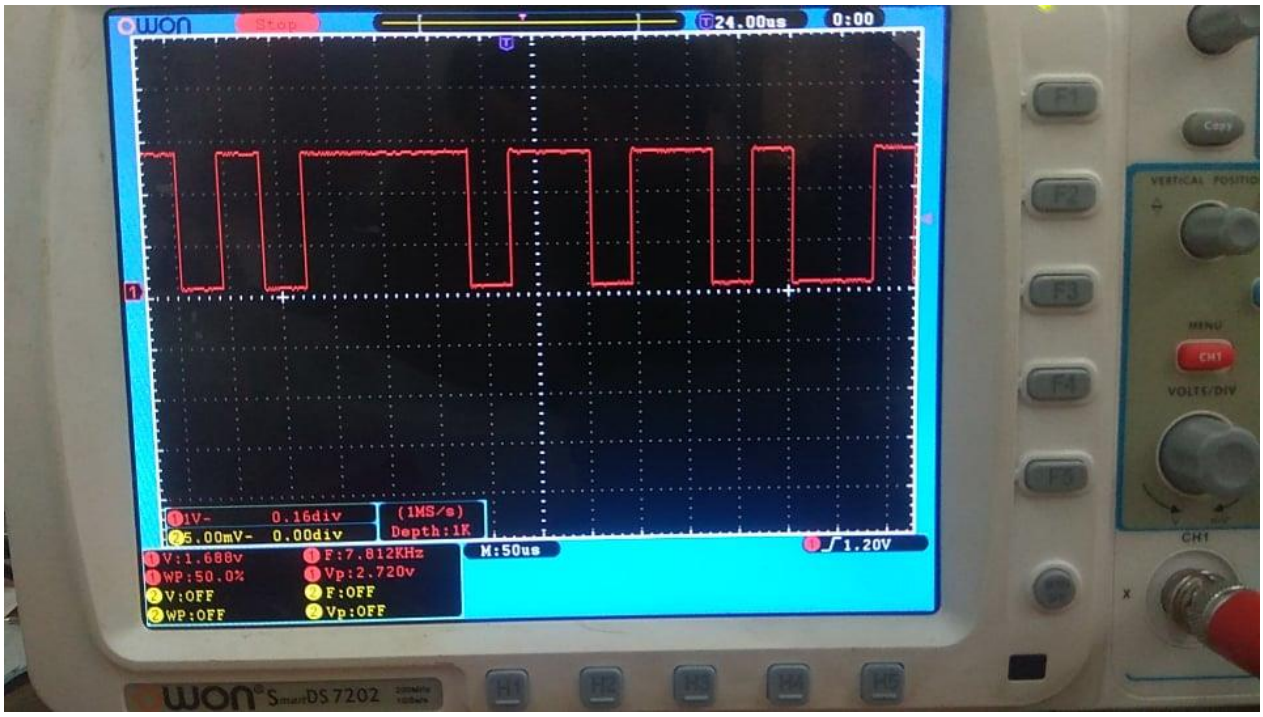


Figure 3.12. Tín hiệu vi sai

CHƯƠNG 4. ỨNG DỤNG THỰC TẾ

- CAN được ứng dụng trong rất nhiều lĩnh vực khác nhau như: Xe ô tô, tàu khách và tàu hàng, hệ thống điện tử hàng hải, điện tử máy bay, hàng không, tự động hóa trong nhà máy, điều khiển máy công nghiệp, tự động hóa tòa nhà, thang máy, thiết bị y tế... với tốc độ có thể lên tới 1Mbit/s.
- Ứng dụng mạng trong doanh nghiệp: Xây dựng mạng phụ để khởi động hệ thống và hỗ trợ tháo lắp thiết bị trên các bo PC lớn dùng để định tuyến.
- Ứng dụng trong bệnh viện: điều khiển các thiết bị trong phòng như bảng điện tử, máy nội soi, máy X-quang, camera, máy in...
- Ứng dụng trong tự động hóa nhà máy: điều khiển tự động hóa các thiết bị.
- Ứng dụng trong hệ thống thang máy: Các bo điều khiển thông tin với nhau qua bus CAN ở các tầng và bo mạch chính.
- Ứng dụng trong hệ thống cửa tự động: Ứng dụng trong cửa Nabco.

CHƯƠNG 5. ƯU ĐIỂM VÀ SỰ KHÁC BIỆT CỦA CAN SO VỚI CÁC KẾT NỐI KHÁC

- Đơn giản, chi phí thấp: Bus CAN chỉ có 2 dây giúp kết nối các module điều khiển với nhau dễ dàng hơn khi so sánh với cách làm truyền thống. Việc này giúp cho việc lắp đặt, sửa chữa, bảo trì hệ thống khi có sự cố một cách dễ dàng.
- Tạo ra một giao thức chung để nhiều nhà cung cấp khác nhau có thể phát triển các module điều khiển tương thích với nhau.
- Tính ưu tiên của thông điệp (Prioritization of messages): mỗi thông điệp được truyền từ một nút (node) hay node (station) trên bus CAN đều có mức ưu tiên. Khi nhiều thông điệp được truyền ra bus cùng một lúc thì thông điệp nào có mức ưu tiên cao nhất sẽ được truyền đi. Các thông điệp có mức ưu tiên thấp hơn sẽ được truyền khi các lệnh ưu tiên được thực hiện. Việc xác định mức ưu tiên của thông điệp dựa trên các quy định trong chuẩn ISO11898.
- Cấu hình linh hoạt: Cho phép thiết lập cấu hình bao gồm thời gian bit, thời gian đồng bộ, độ dài dữ liệu truyền, dữ liệu nhận...
- Nhận dữ liệu đa điểm với sự đồng bộ thời gian: một thông điệp có thể được nhận bởi nhiều node khác nhau trong bus cùng lúc. Tất cả các node trên bus đều có thể thấy thông điệp đang truyền trên bus, tùy vào cấu hình ở mỗi node mà node sẽ quyết định có chấp nhận thông điệp này hay không.
- Nhiều master (Multimaster): Có thể quản lí từ nhiều nguồn
- Phát hiện và báo lỗi: Mỗi thông điệp có kèm theo mã CRC (Cyclcic Redundancy Code) để thực hiện kiểm tra lỗi. Nếu lỗi xuất hiện, node nhận sẽ bỏ qua thông điệp lỗi và truyền khung báo lỗi (error frame) lên bus CAN. Mỗi node trong bus có một bộ đếm quản lý lỗi truyền nhận riêng để xác định trạng thái lỗi của chính nó. Nếu lỗi xuất hiện quá nhiều, một node có thể tự động ngắt khỏi bus. Ngoài ra còn một số dạng lỗi khác có thể được phát hiện với chuẩn CAN.
- Tự động truyền lại các thông điệp bị lỗi khi bus rảnh: Một thông điệp được truyền ra bus nếu bị lỗi thì sẽ không mất đi mà node truyền thông này sẽ giữ nó lại và tự động phát lại thông điệp này khi bus rảnh cho đến khi thành công. Điều này giúp đảm bảo tính toàn vẹn dữ liệu trong bus.

TÀI LIỆU THAM KHẢO

[1] Khái niệm cơ bản về giao tiếp CAN (bkaii.com.vn)

[2] CAN bus – Wikipedia tiếng Việt

[3] CAN Bit Time Calculation (can-wiki.info)

[4] Controller Area Network (CAN) Overview - NI

[5] CAN Bus Explained - A Simple Intro (2020) (csselectronics.com)

[6] STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced Arm®-based 32-bit MCUs - Reference manual

[7] Mastering STM32 by Carmine Noviello