Introduction to CAN

Summary

This document is an “introduction to CAN” for those who wish to study CAN from now. It describes the outline of CAN and the CAN protocol, explaining what CAN is, the features of CAN, and its position in standard specifications.

Notes on Use of this Document

This document is prepared as a guide to the outline of CAN and the CAN protocol proposed by BOSCH (Robert Bosch GmbH), and is only a reference material to be consulted when using CAN in application systems. It does not implicitly or otherwise provide any guarantee for the application systems that incorporate the CAN functions.

Contents

1. What is CAN?

CAN, which stands for Controller Area Network, is the serial communication protocol internationally standardized by ISO *1.

The automobile industry has hitherto witnessed the advent of various electronic control systems that have been developed in pursuit of safety, comfort, pollution prevention, and low cost. These control systems, however, presented a drawback in that since the communication data types, required reliability, etc. differed between each system, they were configured in multiple bus lines, resulting in increased wire harnesses. Therefore, the need arose for reducing the number of wire harnesses, transferring large amounts of data at high speed via multiple LANs, and so on. To meet the need, BOSCH, an electrical equipment manufacturer in Germany, developed CAN in 1986 as a communication protocol for automotives. Thereafter, CAN was standardized in ISO 11898 and ISO 11519, establishing itself as the standard protocol for in-vehicle networking in Europe now.

Today, CAN is widely accepted for its high performance and reliability, and is used in a broad range of fields from FA devices and ships to medical and industrial equipment.

Figure 1 schematically shows how in-vehicle networking will be conceived. In this conception, CAN and the other communication protocols developed concurrently made it possible for multiple LANs to exchange data efficiently via a gateway.

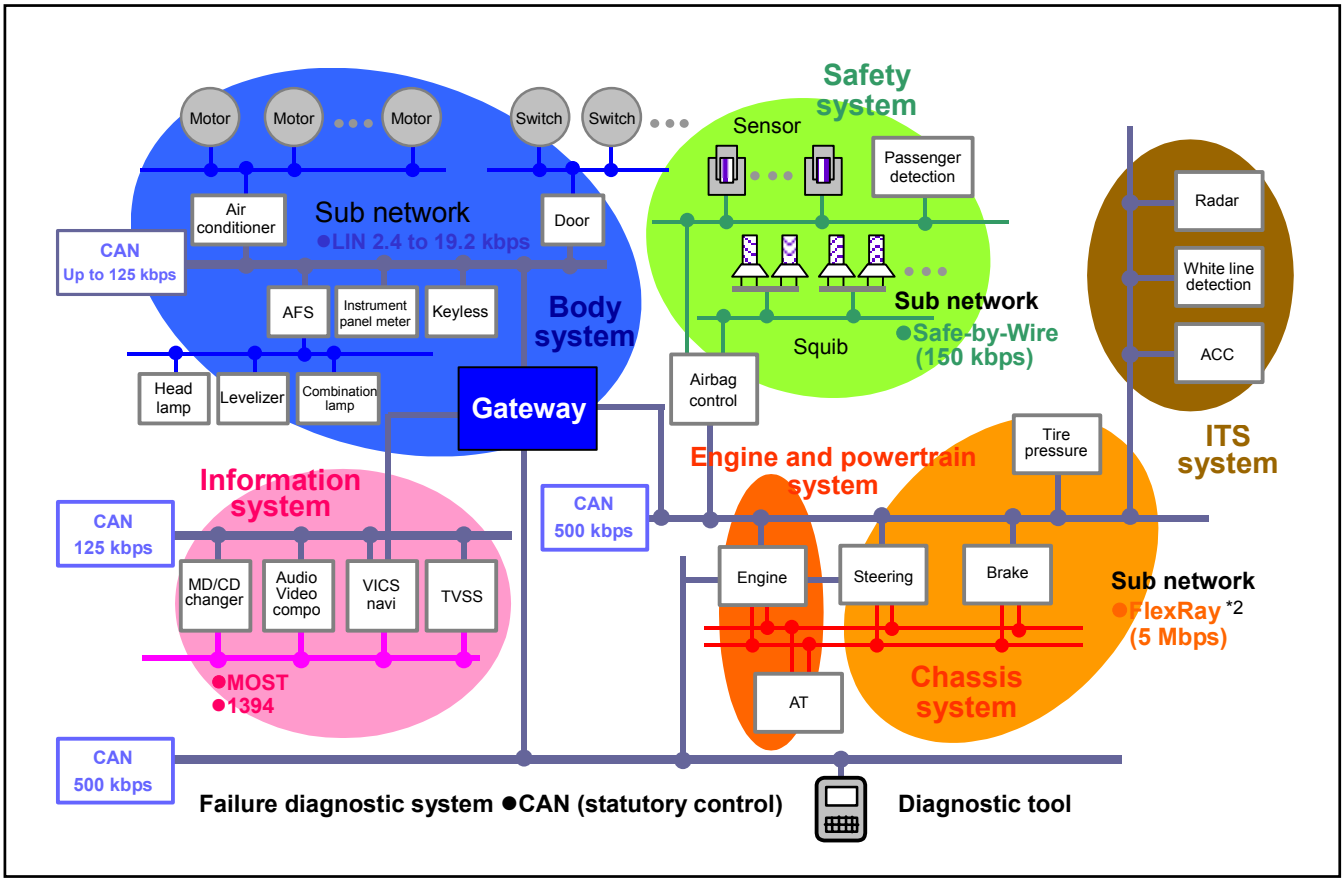


Figure 1. Conception of In-vehicle Networking

*1 : ISO stands for International Organization for Standardization.

*2 : FlexRay™ is a registered trademark of DaimlerChrysler AG.

1.1 Example Application of CAN

Figure 2 shows an example application of CAN.

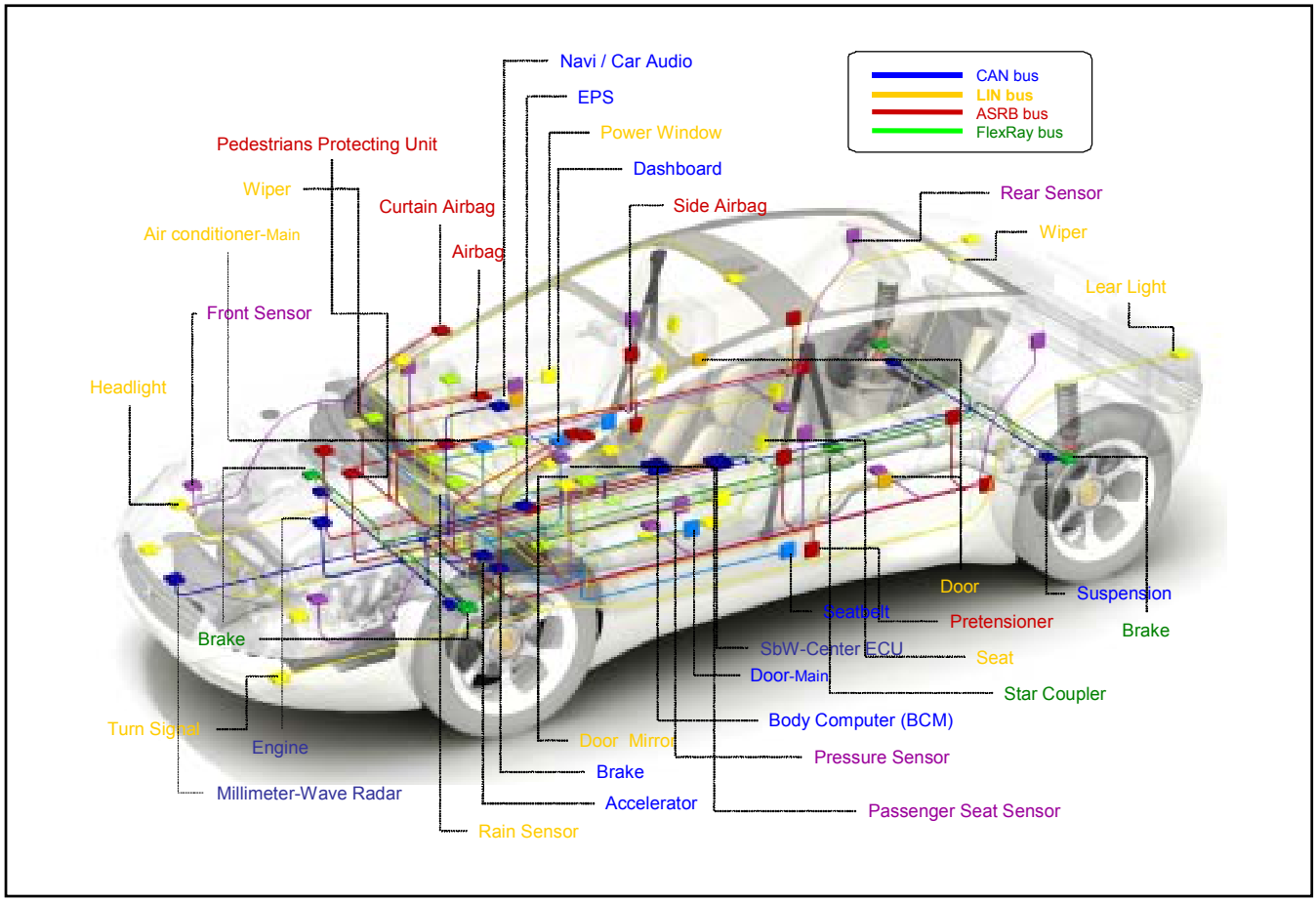


Figure 2. Example Application of CAN

1.2 Bus Topology

The CAN controller determines the level of a bus by potential difference in two wires that comprise the bus. There are two bus levels: dominant level and recessive level, either of which is assumed at any given time. The transmit unit can send a message to the receive unit by changing this bus level.
Figure 3 shows a typical CAN connection diagram.
All units connected to the bus can send a message. If two or more units start sending a message at the same time, the unit whose message has highest priority is granted the rights to send, and all other units perform a receive operation.

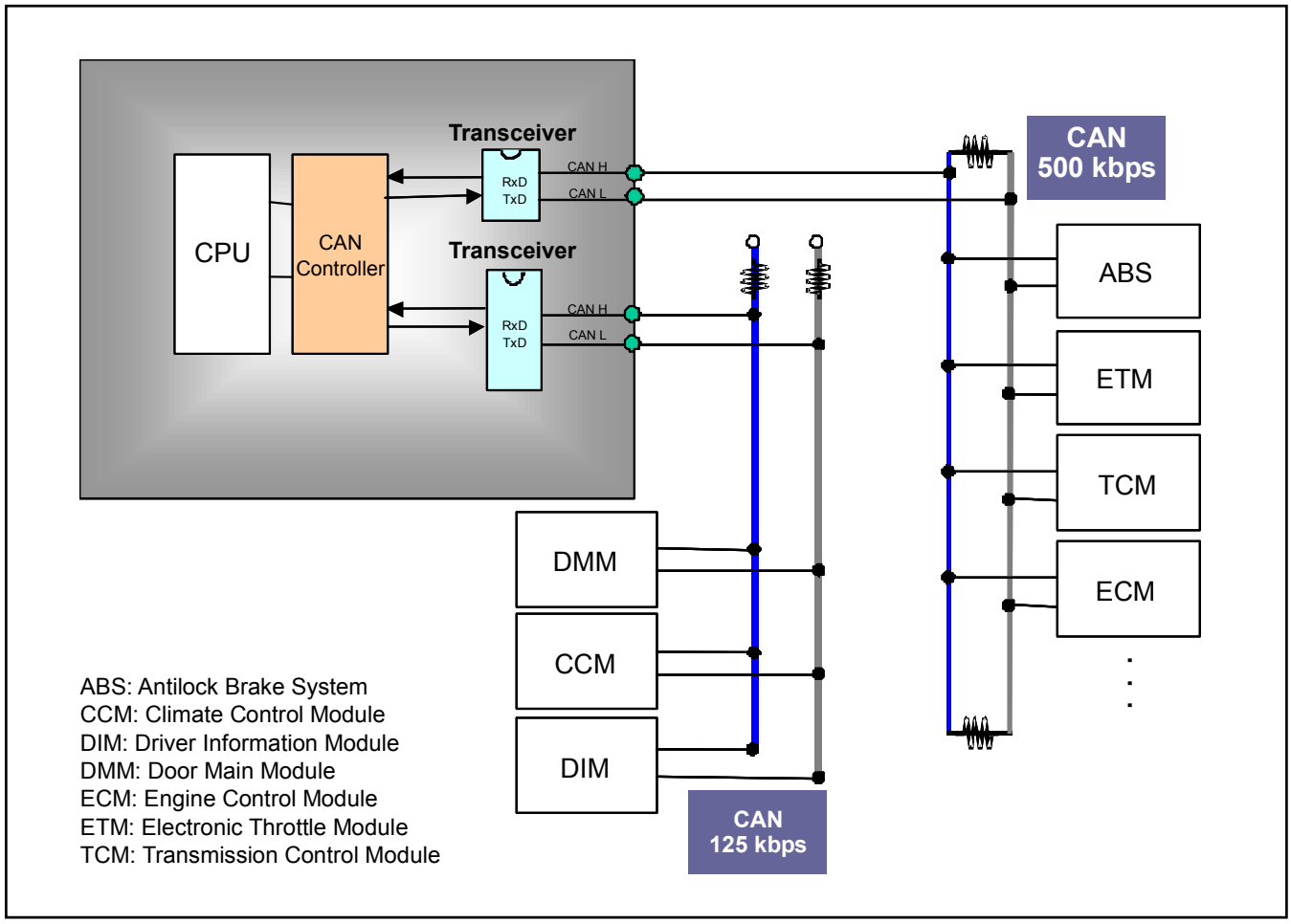


Figure 3. Typical CAN Connection Diagram

2. Features of CAN

The CAN protocol has the following features.

(1) Multimaster

When the bus is free, all of the units connected to it can start sending a message (multimasters).
The unit that first started sending a message to the bus is granted the right to send (CSMA/CR method*1).
If multiple units start sending a message at the same time, the unit that is sending a message whose ID has the highest priority is granted the right to send.

(2) Message transmission

In the CAN protocol, all messages are transmitted in predetermined format. When the bus is unoccupied, all units connected to the bus can start sending a new message. If two or more units start sending a message at the same time, their priority is resolved by an identifier (hereafter the ID). The ID does not indicate the destination to which a message is sent, but rather indicates the priority of messages in which order the bus is accessed. If two or more units start a message at the same time, contention for the bus is arbitrated according to the ID of each message by comparing the IDs bitwise. The unit that won the arbitration (i.e., the one that has the highest priority) can continue to send, while the units that lost in arbitration immediately stop sending and go to a receive operation.

(3) System flexibility

The units connected to the bus have no identifying information like an address. Therefore, when a unit is added to or removed from the bus, there is no need to change the software, hardware, or application layer of any other unit connected to the bus.

(4) Communication speed

Any communication speed can be set that suits the size of a network.
Within one network, all units must have the same communication speed. If any unit with a different communication speed is connected to the network, it will generate an error, hindering communication in the network. This does not apply to units in other networks, however.

(5) Remote data request

Data transmission from other units can be requested by sending a "remote frame" to those units.

(6) Error detection, error notification, and error recovery functions

All units can detect an error (error detection function).
The unit that has detected an error immediately notifies all other units of the error simultaneously (error notification function). If a unit detects an error while sending a message, it forcibly terminates message transmission and notifies all other units of the error. It then repeats retransmission until the message is transmitted normally (error recovery function).

(7) Error confinement

There are two types of errors occurring in the CAN: a temporary error where data on the bus temporarily becomes erratic due to noise from the outside or for other reasons, and a continual error where data on the bus becomes continually erratic due to a unit's internal failure, driver failure, or disconnections. The CAN has a function to discriminate between these types of errors. This function helps to lower the communication priority of an error-prone unit in order to prevent it from hindering communication of other normal units, and if a continual data error on the bus is occurring, separate the unit that is the cause of the error from the bus.

(8) Connection

The CAN bus permits multiple units to be connected at the same time. There are no logical limits to the number of connectable units. However, the number of units that can actually be connected to a bus is limited by the delay time and electrical load in the bus. A greater number of units can be connected by reducing the communication speed. Conversely, if the communication speed is increased, the number of connectable units decreases.

*1 : CSMA/CR stands for Carrier Sense Multiple Access with Collision Resolution.

3. Errors

3.1 Error States

A unit is always in one of three states.

(1) Error active state

The error active state is a state in which the unit can participate in communication on the bus normally.
If the unit in an error active state detects an error, it transmits an active-error flag.
For details, refer to Chapter 6, “CAN Protocol.”

(2) Error passive state

The error passive state is a state in which the unit tends to cause an error.
Although the unit in an error passive state can participate in communication on the bus, it cannot notify other units of an error positively during a receive operation in order not to hinder their communication.
Even when the unit in an error passive state has detected an error, if no errors are detected by other units in an error active state, it is assumed that no errors have occurred anywhere in the bus.
When the unit in an error passive state has detected an error, it transmits a passive-error flag.
Furthermore, the unit in an error passive state cannot start transmission immediately after it has finished sending. A suspend transmission period (comprised of 8 recessive bits) is inserted in an interframe space before the next transmission can start.
For details, refer to Chapter 6, “CAN Protocol.”

(3) Bus off state

The bus off state is a state in which the unit cannot participate in communication on the bus.
When in this state, the unit is disabled from all transmit/receive operations.

Each of these states are managed by the transmit error counter and receive error counter, and the relevant error state is identified by a combination of these counter values. The relationship between error states and counter values is shown in Table 1 and Figure 4.

Table 1. Error States and Counter Values

Error state of a unit	Transmit error counter (TEC)		Receive error counter (REC)
Error active state	0 – 127	AND	0 – 127
Error passive state	128 – 255	OR	128 – 255
Bus off state	Minimum 256		–

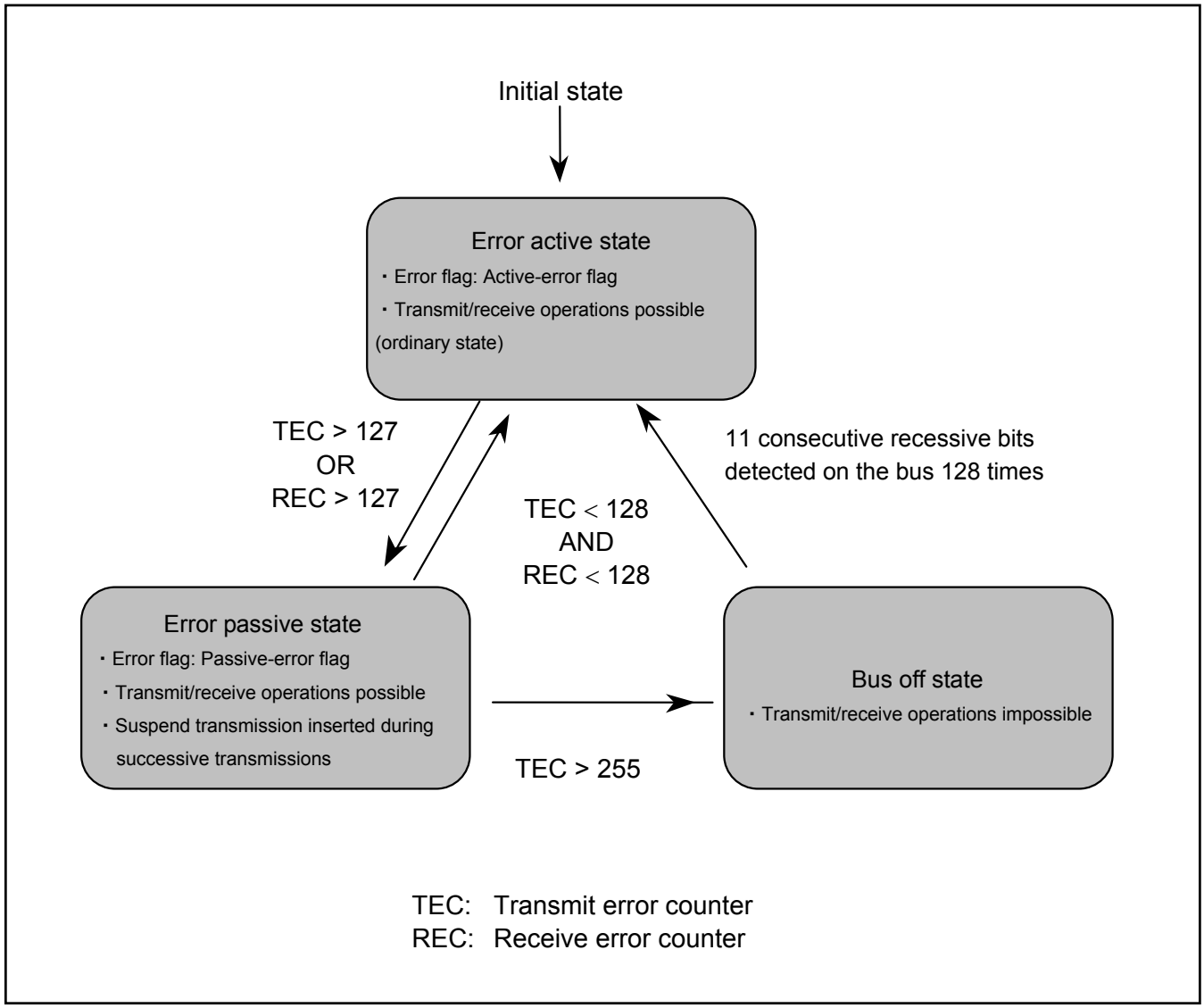


Figure 4. Error States of a Unit

3.2 Values of the Error Counters

The values of the transmit and receive error counters change according to prescribed conditions.
Table 2 summarizes the conditions under which the error counter values change.
It is possible that multiple conditions will overlap in one data transmit/receive operation.
The time at which the error counter counts up coincides with the first bit position of the error flag.

Table 2. Error Counter Value Change Conditions

	Transmit/receive error counter change conditions	Transmit error counter (TEC)	Receive error counter (REC)
1	When the receive unit has detected an error Except in the case where the receive unit detected a bit error while it was sending an active-error flag or overload flag.	–	+1
2	When the receive unit has detected a dominant level in the first bit that it received after sending an error flag.	–	+8
3	When the transmit unit has transmitted an error flag *1	+8	–
4	When the transmit unit has detected a bit error while sending an active-error flag or overload flag	+8	–
5	When the receive unit has detected a bit error while sending an active-error flag or overload flag	–	+8
6	When any unit has detected a dominant level in 14 consecutive bits from the beginning of an active-error or an overload flag, and each time the unit has detected a dominant level in 8 consecutive bits thereafter.	For a transmit unit +8	For a receive unit +8
7	When any unit has detected a dominant level in additional 8 consecutive bits after a passive-error flag, and each time the unit has detected a dominant level in 8 consecutive bits thereafter.	For a transmit unit +8	For a receive unit +8
8	When the transmit unit has transmitted a message normally (ACK returned and no errors detected until completion of EOF).	-1 ±0 when TEC = 0	–
9	When the receive unit has received a message normally (no errors detected until ACK slot and the unit was able to return ACK normally).	–	–1 when 1 ≤ REC ≤ 127 ±0 when REC = 0 When REC > 127, a value between 119 to 127 is set in REC
10	When the unit in a bus-off state has detected a recessive level in 11 consecutive bits 128 times.	Cleared to TEC = 0	Cleared to REC = 0

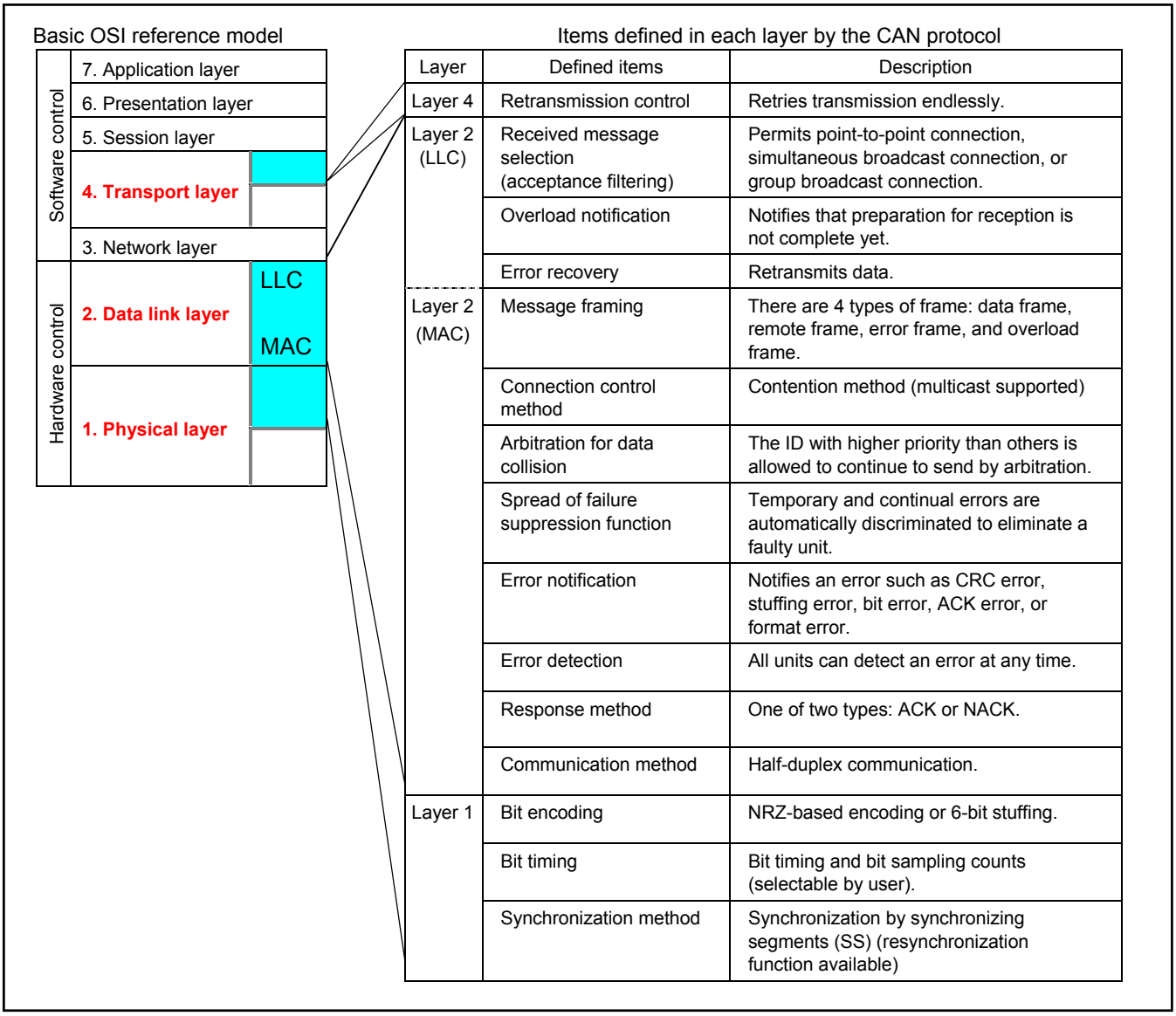
*1: The transmit error counter does not change in the following cases:

- When the transmit unit while in an error-passive state has detected an ACK error for reasons that ACK was not detected and has detected no dominant levels while sending a passive-error flag.
- When the transmit unit has encountered a stuffing error during arbitration (dominant level is detected although it transmitted a recessive level as bit stuffing).

4. Basic Concept of the CAN Protocol

The CAN protocol includes the transport, data link, and physical layers of the basic OSI*1 reference model. Figure 5 shows the defined items of each layer of the basic OSI reference model in the CAN protocol.
The data link layer is divided into MAC*2 and LLC*3 sub-layers, of which the MAC sub-layer constitutes the nucleus of the CAN protocol. The function of the data link layer is to put the signals received from the physical layer together into a meaningful message to provide a procedure for data transmission control such as transmission error control. More specifically, this includes assembling messages into a frame, arbitrating data collision, returning ACK, and detecting or notifying errors. These functions of the data link layer normally are executed in hardware by the CAN controller.
For the physical layer, the protocol defines the manner in which signals are actually transmitted and the bit timing, bit encoding, and synchronization procedures. However, this does not mean that the signal levels, communication speeds, sampling point values, driver and bus electrical characteristics, and connector forms are defined specifically by the CAN protocol. All of these must be selected for each system by the user.
In CAN specifications of BOSCH, there are no definitions with respect to the electrical characteristics, etc. of transceivers and buses. In the ISO standards for the CAN protocol, i.e., ISO11898 and ISO11519-2, however, the electrical characteristics, etc. of transceivers and buses are defined in each.

*1 : OSI stands for Open System Interconnection.
*2 : MAC stands for Medium Access Control
*3 : LLC stands for Logical Link Control



Items defined in each layer by the CAN protocol

Layer	Defined items	Description
Layer 4	Retransmission control	Retries transmission endlessly.
Layer 2 (LLC)	Received message selection (acceptance filtering)	Permits point-to-point connection, simultaneous broadcast connection, or group broadcast connection.
	Overload notification	Notifies that preparation for reception is not complete yet.
	Error recovery	Retransmits data.
Layer 2 (MAC)	Message framing	There are 4 types of frame: data frame, remote frame, error frame, and overload frame.
	Connection control method	Contention method (multicast supported)
	Arbitration for data collision	The ID with higher priority than others is allowed to continue to send by arbitration.
	Spread of failure suppression function	Temporary and continual errors are automatically discriminated to eliminate a faulty unit.
	Error notification	Notifies an error such as CRC error, stuffing error, bit error, ACK error, or format error.
	Error detection	All units can detect an error at any time.
	Response method	One of two types: ACK or NACK.
	Communication method	Half-duplex communication.
Layer 1	Bit encoding	NRZ-based encoding or 6-bit stuffing.
	Bit timing	Bit timing and bit sampling counts (selectable by user).
	Synchronization method	Synchronization by synchronizing segments (SS) (resynchronization function available)

Figure 5. Basic OSI Reference Model of ISO and the CAN Protocol

5. CAN Protocol and Standard Specifications

5.1 CAN Protocol Standardized by ISO

The CAN protocol has been standardized by ISO, so that there are several ISO standards for CAN such as ISO11898 and ISO11519-2. In the ISO11898 and ISO11519-2 standards, there are no differences in the definition of the data link layer, but differences exist for the physical layer.

(1) About ISO11898

ISO11898 is a standard for high-speed CAN communication.
Formerly, both the data link and the physical layers were stipulated in ISO11898. It was then separated into ISO11898-1 or the standard for only the data link layer and ISO11898-2 or the standard for only the physical layer. On that occasion, several specifications were added to ISO11898-1.

(2) About ISO11519

ISO11519 is a standard for low-speed CAN communication for communication speeds up to 125 kbps.

5.2 Differences between ISO11898 and ISO11519-2

Figure 6 shows the stipulated range of the CAN protocol and that of ISO11898-1 and ISO11898-2 in the CAN protocol. What is defined by the CAN protocol are the three sublayers of the physical layer: PLS*1 sublayer, PMA*2 sublayer, and MDI*3 sublayer. Of these, the defined contents of the PMA and MDI sublayers differ.

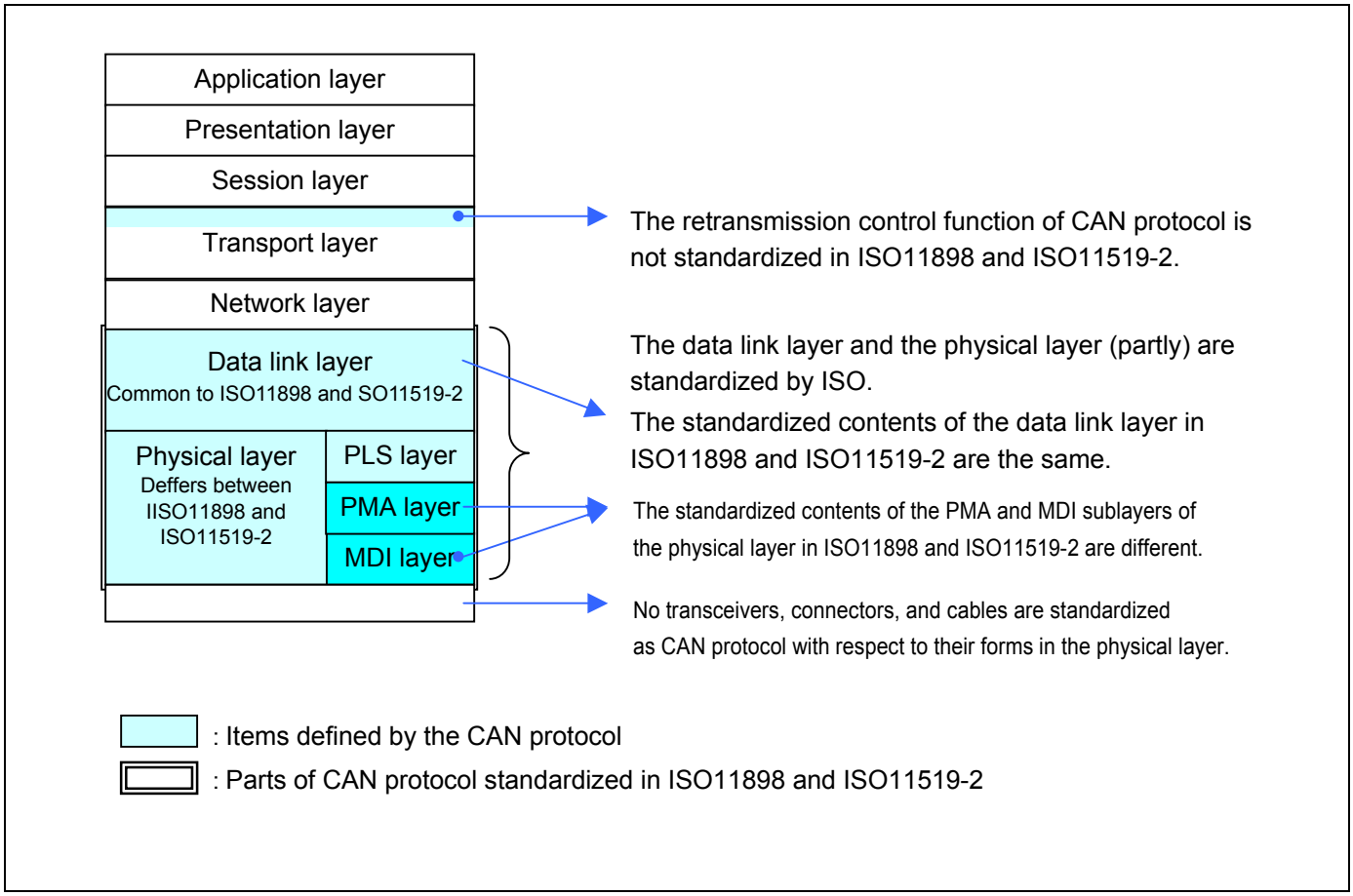


Figure 6. CAN Protocol Standard by ISO

*1 : PLS stands for Physical Signaling Sub-layer.
*2 : PMA stands for Physical Medium Attachment.
*3 : MDI stands for Medium Dependent Interface.

Table 3 lists the primary differences in the physical layer between ISO11898 and ISO11519-2. Figure 7 shows the relationship between the communication speed and bus length.
The communication speed and bus length need to be set according to the system by the user.

Table 3. Main Differences in the Physical Layer between ISO11898 and ISO11519-2

Physical layer	ISO 11898 (High speed)						ISO 11519-2 (Low speed)					
Communication speed	Up to 1 Mbps						Up to 125 kbps					
Maximum bus length	40 m/ 1 Mbps						1 km/ 40 kbps					
Number of connected units	Maximum 30						Maximum 20					
Bus topology*1	Recessive			Dominant			Recessive			Dominant		
	Min.	Nom.	Max.	Min.	Nom.	Max.	Min.	Nom.	Max.	Min.	Nom.	Max.
CAN_High (V)	2.00	2.50	3.00	2.75	3.50	4.50	1.60	1.75	1.90	3.85	4.00	5.00
CAN_Low (V)	2.00	2.50	3.00	0.50	1.50	2.25	3.10	3.25	3.40	0.00	1.00	1.15
Potential difference (H–L) (V)	-0.5	0	0.05	1.5	2.0	3.0	-0.3	-1.5	-	0.3	3.00	-
	Twisted pair wire (shielded/unshielded) Loop bus Impedance (Z): 120 Ω (Min. 85 Ω, Max. 130 Ω) Bus resistivity (r): 70 MΩ/m Bus delay time: 5 ns/m Terminating resistance: 120 Ω (Min. 85 Ω, Max. 130 Ω)						Twisted pair wire (shielded/unshielded) Open bus Impedance (Z): 120 Ω (Min. 85 Ω, Max. 130 Ω) Bus resistivity (r): 90 MΩ/m Bus delay time: 5 ns/m Terminating resistance: 2.20 Ω (Min. 2.09 Ω, Max. 2.31 Ω) CAN_L and GND capacitance: 30 pF/m CAN_H and GND capacitance: 30 pF/m CAN_L and GND capacitance: 30 pF/m					

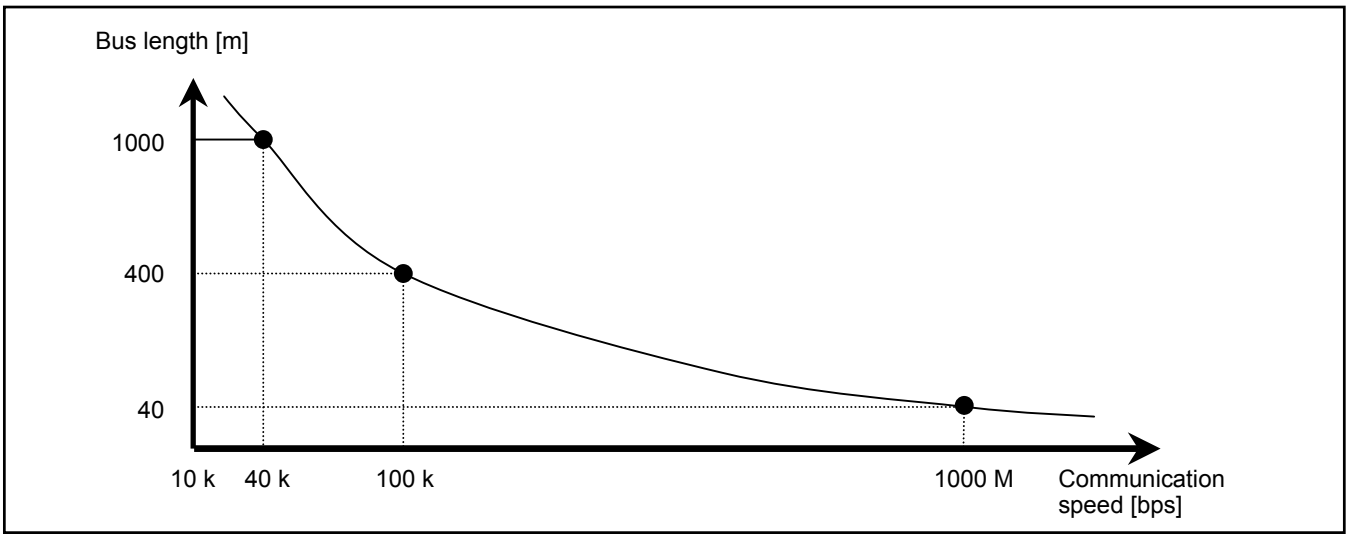


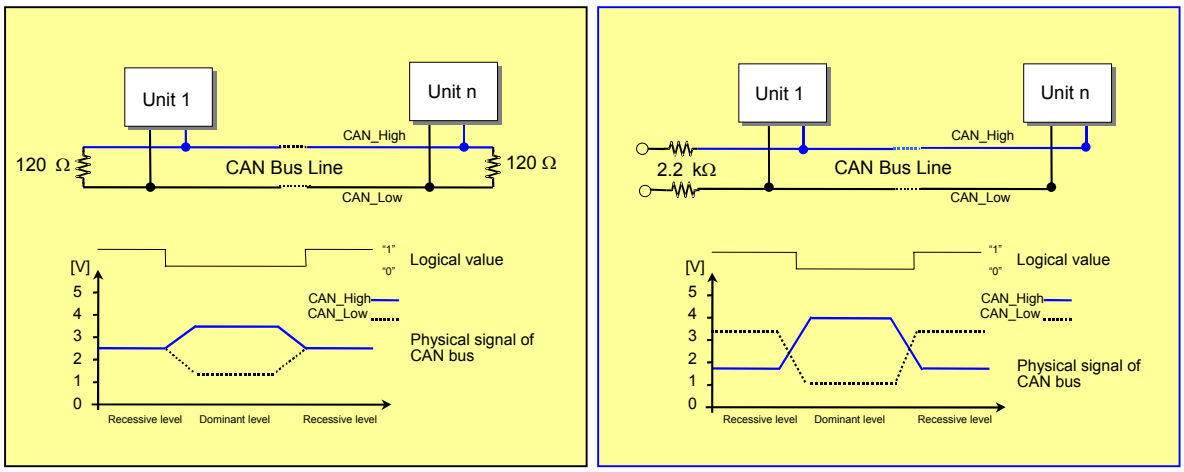
Figure 7. Communication Speed and Maximum Bus Length

*1 : Bus topology

The CAN bus normally consists of two wires (CAN_High and CAN_Low), and the CAN controller is connected to those two wires via a transceiver. The bus level is determined by a potential difference between the CAN_High and CAN_Low wires. There are two bus levels, dominant and recessive, and the bus assumes either level at any given point of time. For logically wire-AND'd buses, the dominant and the recessive levels are recognized as a logic 0 and logic 1, respectively. A transmit unit can send a message to receive units by changing these bus levels.

The terminating resistance and the dominant level and recessive level potential differences in the physical layer differ between ISO11898 and ISO11519-2, as shown in Table 4.

Figure 8 shows the features of the physical layer in ISO11898 and ISO11519-2. Note that ISO11898 and ISO11519-2 require a transceiver conforming to the respective standards. Table 4 lists the main transceiver ICs conforming to ISO11898 and ISO11519-2.



[For ISO11898 (125 kbps to 1 Mbps)] **[For ISO11519-2 (10 kbps to 125 kbps)]**

Figure 8. Features of the Physical Layer in ISO11898 and ISO11519-2

Table 4. ISO11898 and ISO11519-2 Compliant Driver ICs

Transceiver IC	ISO11898	ISO11519-2
	HA13721RPJE(RENESAS)	
	TJA1050T(Philips)	TJA1054T(Philips)
	TLE6250G(Infineon)	TLE6254-3G(Infineon)
	CF150C(BOSCH)	

5.3 CAN and Standard Specifications

In addition to ISO, the CAN protocol is standardized by industry organizations such as SAE*1, as well as by private institutions and corporations.
Table 5 lists CAN based standard specifications. Figure 9 shows the communication protocols for automotives classified by communication speed.

Table 5. CAN Protocol and Standard Specifications

Name	Baud rate	Specification	Application field
SAE J1939-11	250 k	Two-wire shielded twisted pair	Truck, bus
SAE J1939-12	250 k	Two-wire shielded twisted pair 12 V supply	Agricultural machine
SAE J2284	500 k	Two-wire twisted pair (non-shielded)	Automobile (high-speed: power train system)
SAE J2411	33.3 k, 83.3 k	One-wire	Automobile (low-speed: body system)
NMEA-2000*2	62.5 k, 125 k, 250 k, 500 k, 1 M	Two-wire shielded twisted pair Power supply	Ship
DeviceNet	125 k, 250 k, 500 k	Two-wire shielded twisted pair 24 V supply	Industrial equipment
CANopen	10 k, 20 k, 25 k, 50 k, 125 k 250 k, 500 k, 800 k, 1 M	Two-wire twisted pair Optional (shielded, power supply)	Industrial equipment
SDS*3	125 k, 250 k, 500 k, 1 M	Two-wire shielded twisted pair Optional (power supply)	Industrial equipment

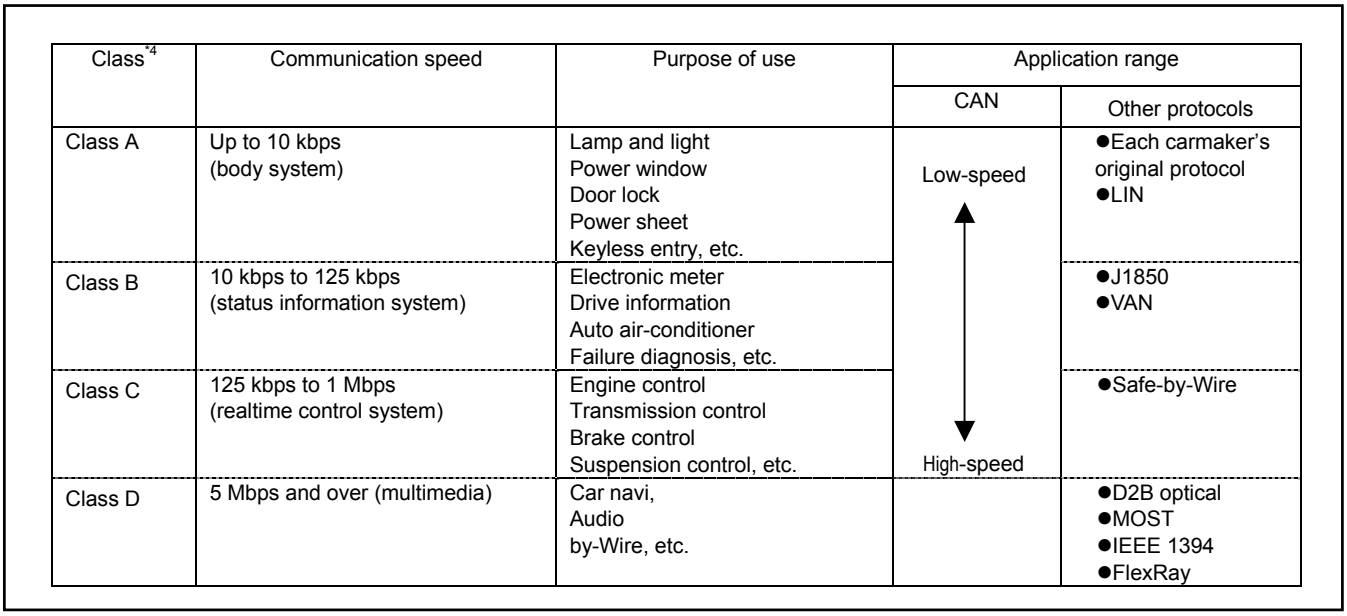


Figure 9. Classification of Communication Protocols

*1 : SAE stands for Society of Automotive Engineers
*2 : NMEA stands for National Marine Educators Association
*3 : SDS stands for Smart Distributed System
*4 : Class names correspond to classification by SAE.

6. CAN Protocol

6.1 Types of Frames

Communication is performed using the following five types of frames.

- Data frame
- Remote frame
- Error frame
- Overload frame
- Interframe space

Of these frames, the data and the remote frames need to be set by the user. The other frames are set by the hardware part of CAN.
The data and the remote frames come in two frame formats: standard and extended. The standard format has a 11-bit ID, and the extended format has a 29-bit ID.
Roles of each frame are listed in Table 6. The structure of each frame is shown in Figure 10 to Figure 14.

Table 6. Frame Types and Roles of Each Frame

Frame	Roles of frame	User settings
Data frame	This frame is used by the transmit unit to send a message to the receive unit.	Necessary
Remote frame	This frame is used by the receive unit to request transmission of a message that has the same ID from the transmit unit.	Necessary
Error frame	When an error is detected, this frame is used to notify other units of the detected error.	Unnecessary
Overload frame	This frame is used by the receive unit to notify that it has not been prepared to receive frames yet.	Unnecessary
Interframe space	This frame is used to separate a data or remote frame from a preceding frame.	Unnecessary

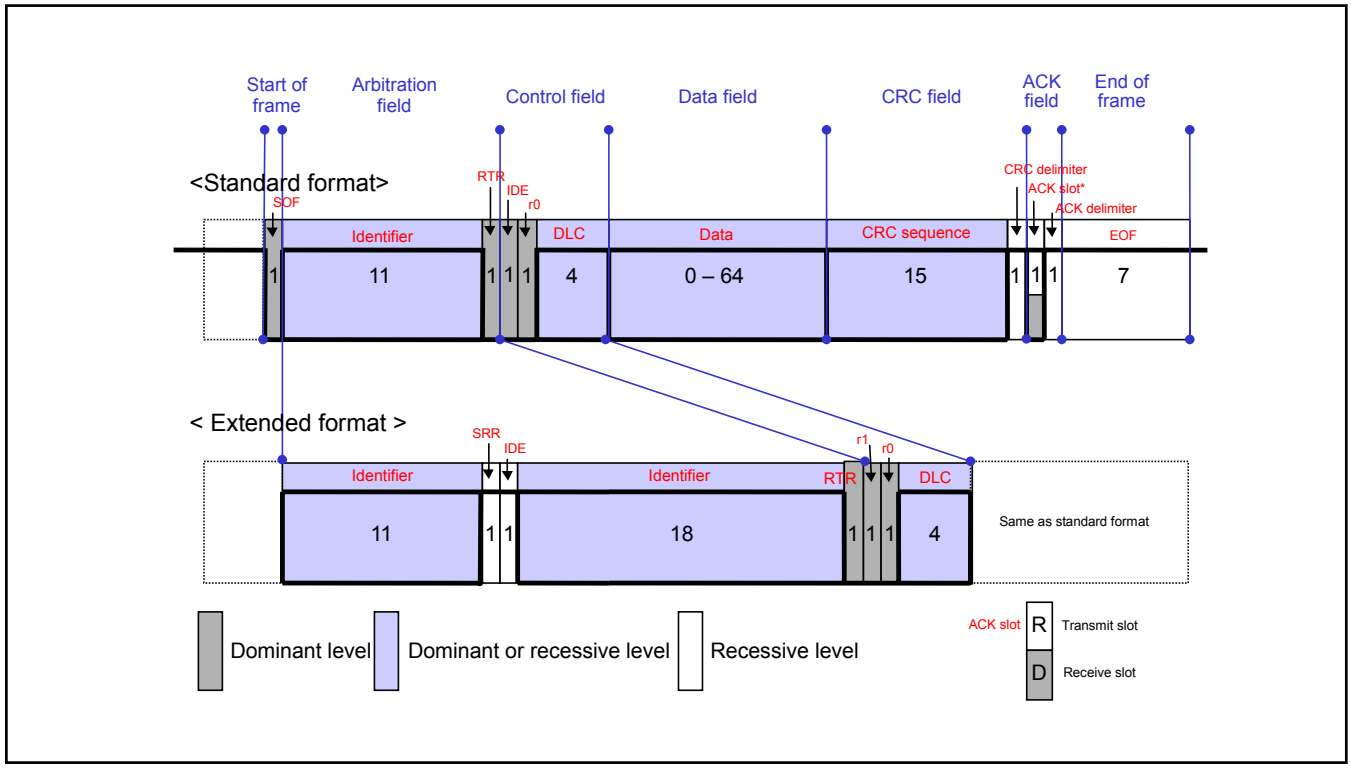


Figure 10. Structure of the Data Frame

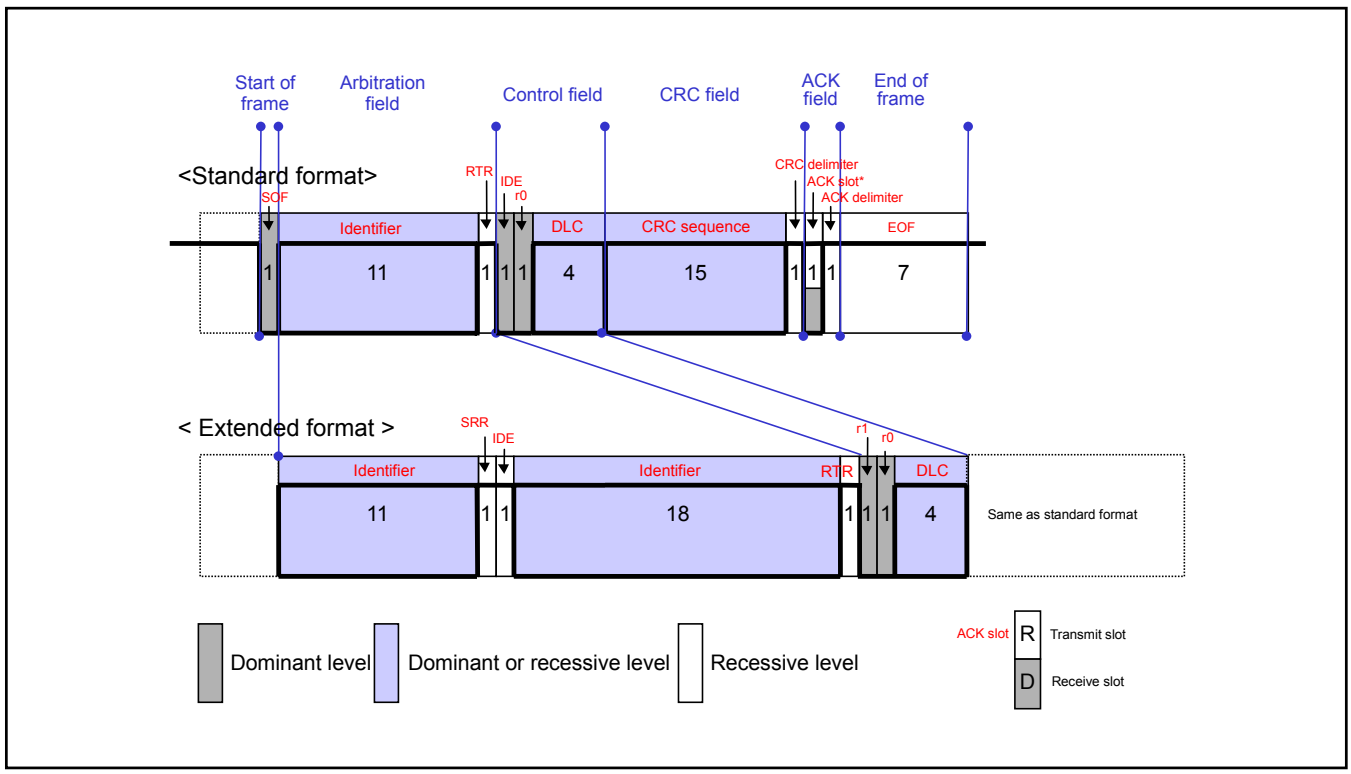


Figure 11. Structure of the Remote Frame

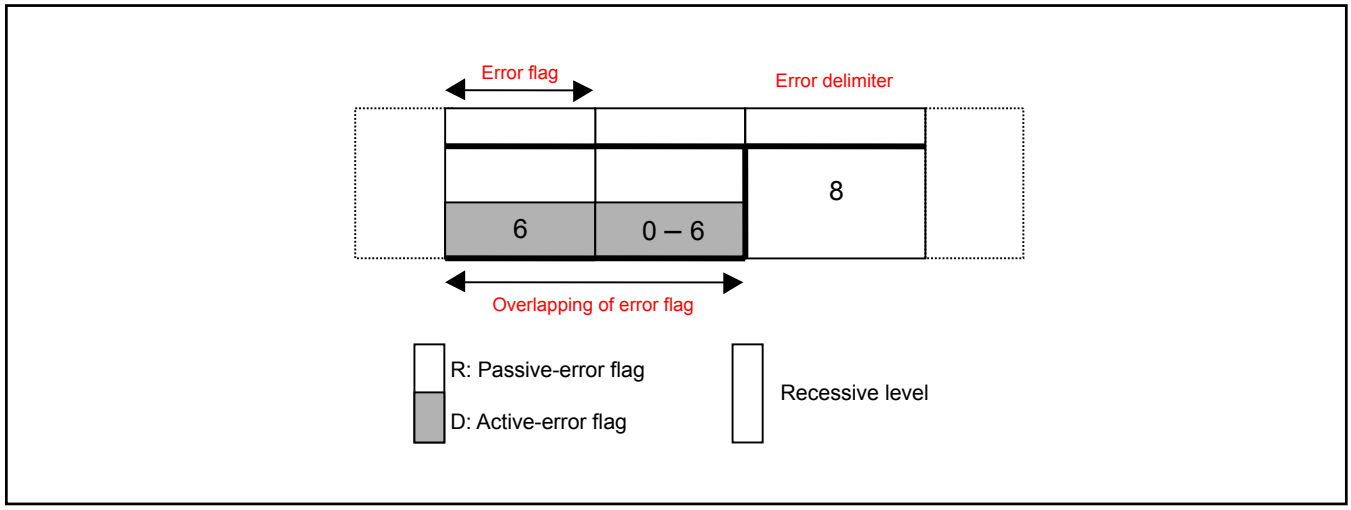


Figure 12. Structure of the Error Frame

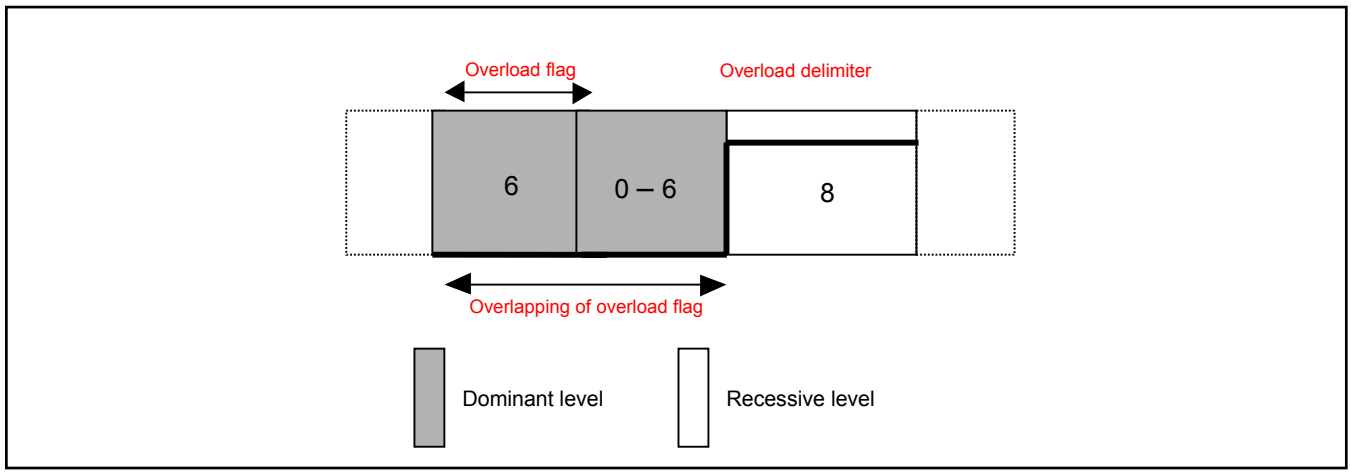


Figure 13. Structure of the Overload Frame

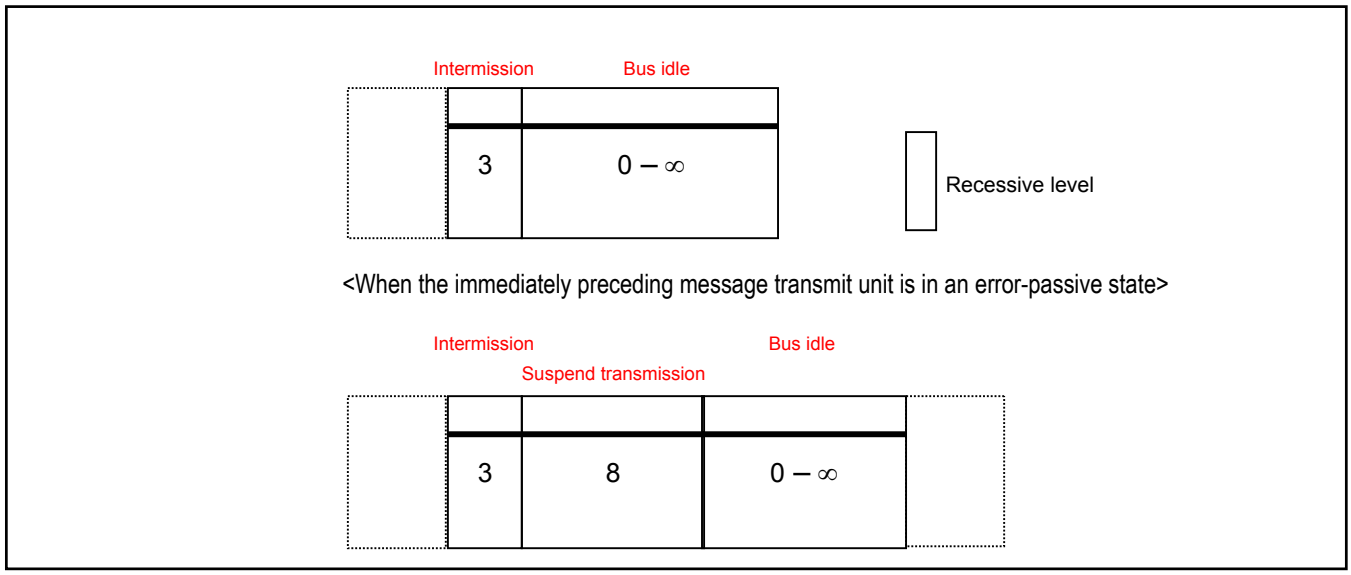


Figure 14. Structure of the Interframe Space

6.2 Data Frame

The data frame is used by the transmit unit to send a message to the receive unit, and is the most fundamental frame handled by the user.
The data frame consists of seven fields.
Figure 15 shows the structure of the data frame.

(1) Start of frame (SOF)
This field indicates the beginning of a data frame.

(2) Arbitration field
This field indicates the priority of a frame.

(3) Control field
This field indicates reserved bits and the number of data bytes.

(4) Data field
This is the content of data. Data in the range 0 to 8 bytes can be transmitted.

(5) CRC field
This field is used to check the frame for a transmission error.

(6) ACK field
This field indicates a signal for confirmation that the frame has been received normally.

(7) End of frame
This field indicates the end of a data frame.

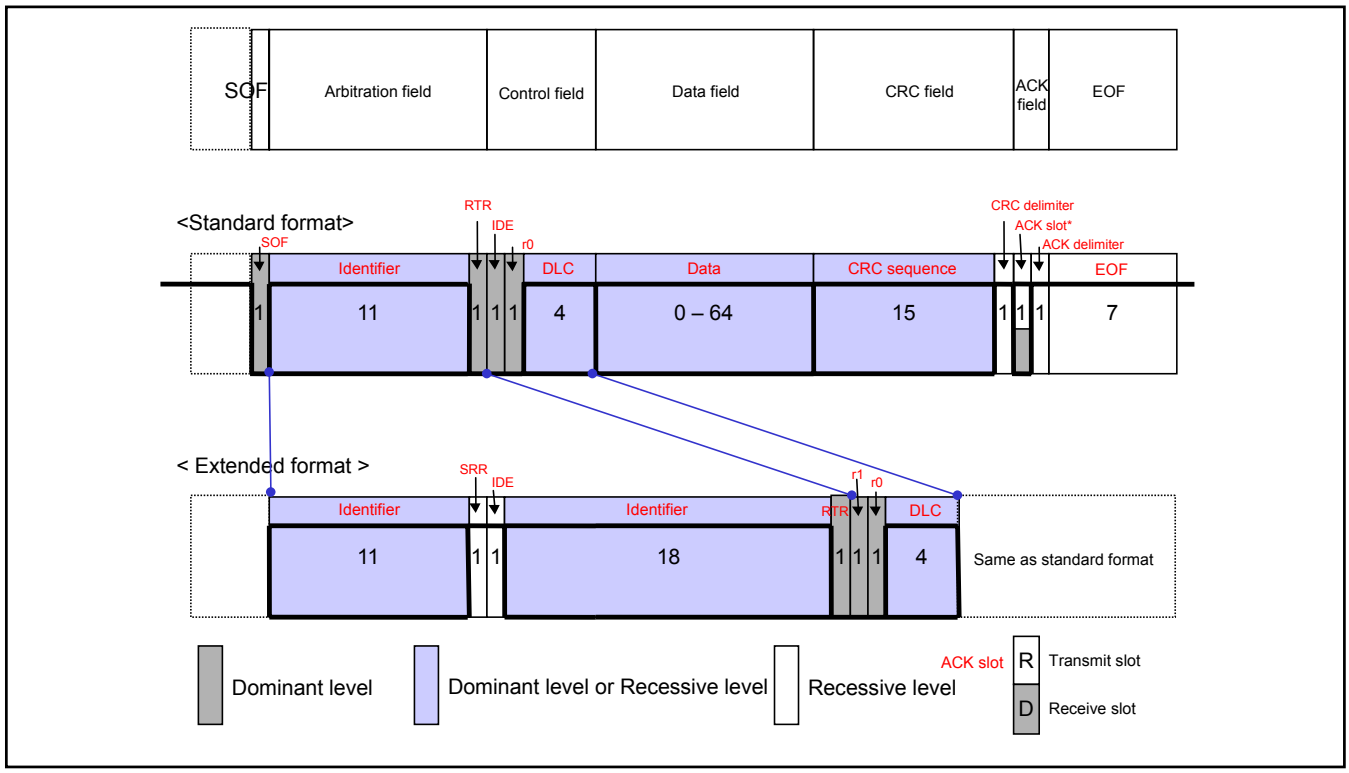


Figure 15. Structure of the Data Frame

(1) Start of frame (common to both standard and extended formats)

This field indicates the beginning of a frame. It consists of one dominant bit.

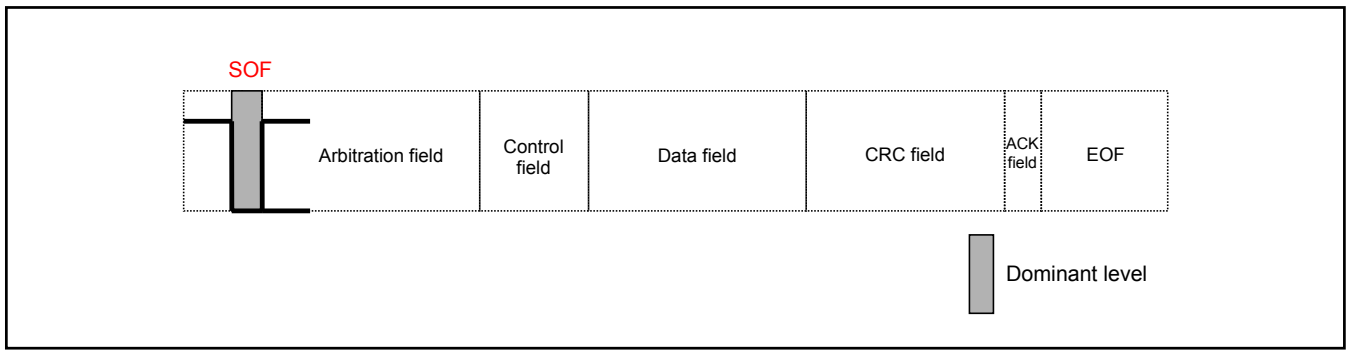


Figure 16. Data Frame (Start of Frame)

(2) Arbitration field

This field indicates the priority of data.

The structure of this field differs between the standard and extended formats.

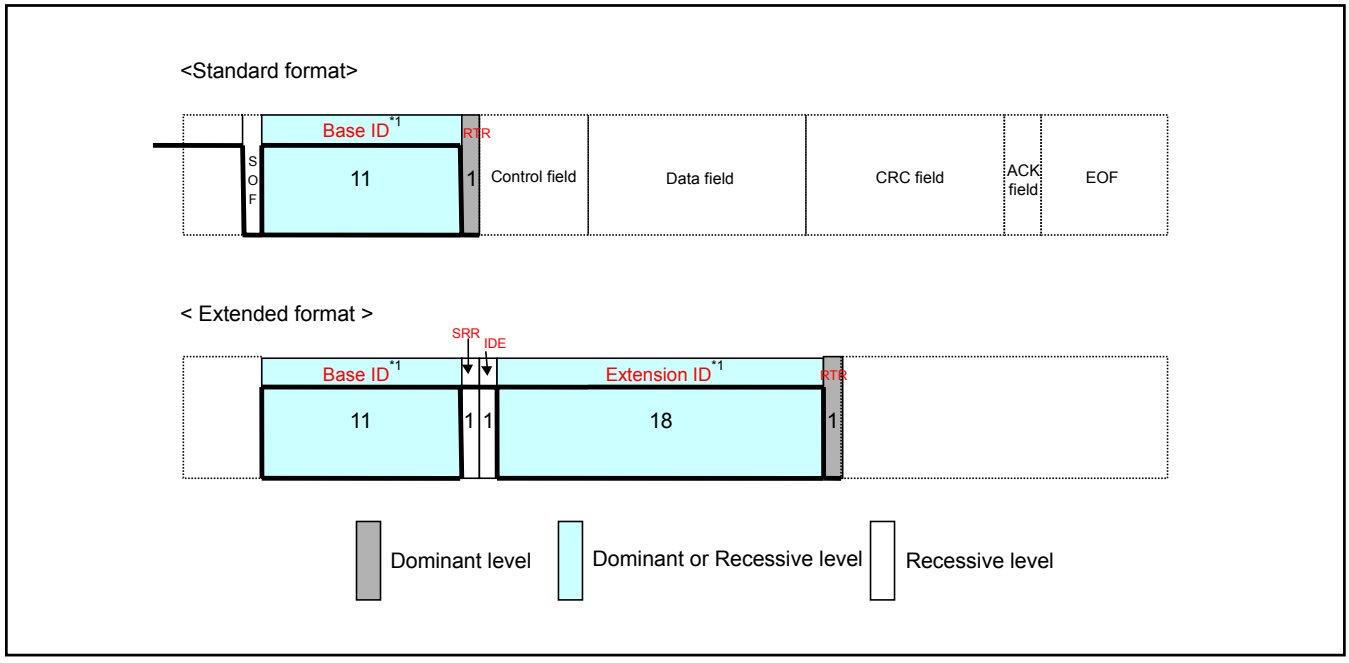


Figure 17. Data Frame (Arbitration Field)

*1 : ID

The standard format ID consists of 11 base ID bits (ID28–ID18). These ID bits are transmitted sequentially beginning with ID28. The 7 high-order bits cannot all be recessive. (IDs set to 1111111XXXX are prohibited.) Up to 2,032 discrete IDs can be set.

The extended format ID consists of 11 base ID bits (ID28–ID18) and 18 extended ID bits (ID17–ID0). The base ID is the same as in the standard format. The 7 high-order bits cannot all be recessive. (IDs set to 1111111XXXX are prohibited.) Up to $2,032 \times 2^{18}$ discrete IDs can be set. In no case can multiple units on the bus transmit data frames with the same ID at the same time.

(3) Control field

This 6-bit field indicates the number of data bytes in a message to be transmitted. The structure of this field differs between the standard and extended formats.

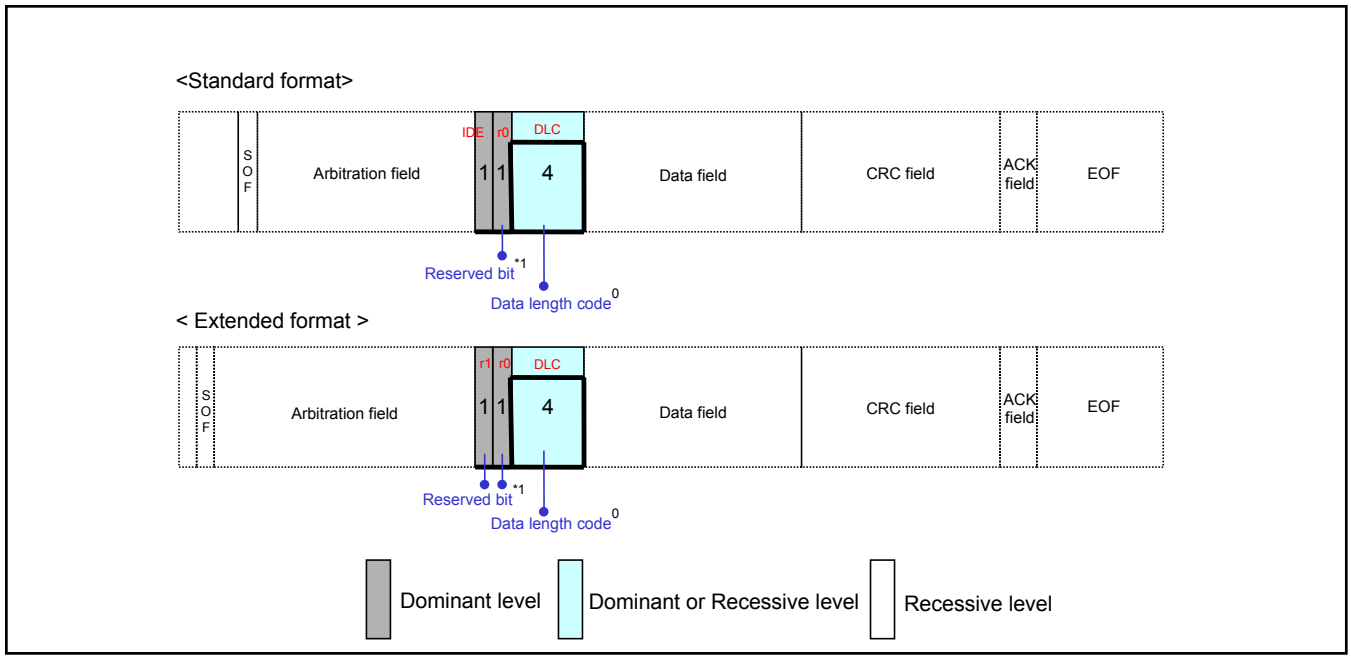


Figure 18. Data Frame (Control Field)

*1 : Reserved bits (r0, r1)
The reserved bits must all be transmitted with a dominant level. On the receiving side, however, they are received in any combination of dominant and recessive levels.

*2 : Data length code (DLC)
The data length code and the number of data bytes are shown in Table 7.

Table 7. Data Length Code and the Number of Data Bytes

Number of data bytes	Data length code			
	DLC3	DLC2	DLC1	DLC0
0	D	D	D	D
1	D	D	D	R
2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D or R	D or R	D or R

D: Dominant level, R: Recessive level

(4) Data field (common to both standard and extended formats)

This field indicates the content of data. Zero to 8 bytes of data set in the control field can be transmitted. The data is output beginning with the MSB side.

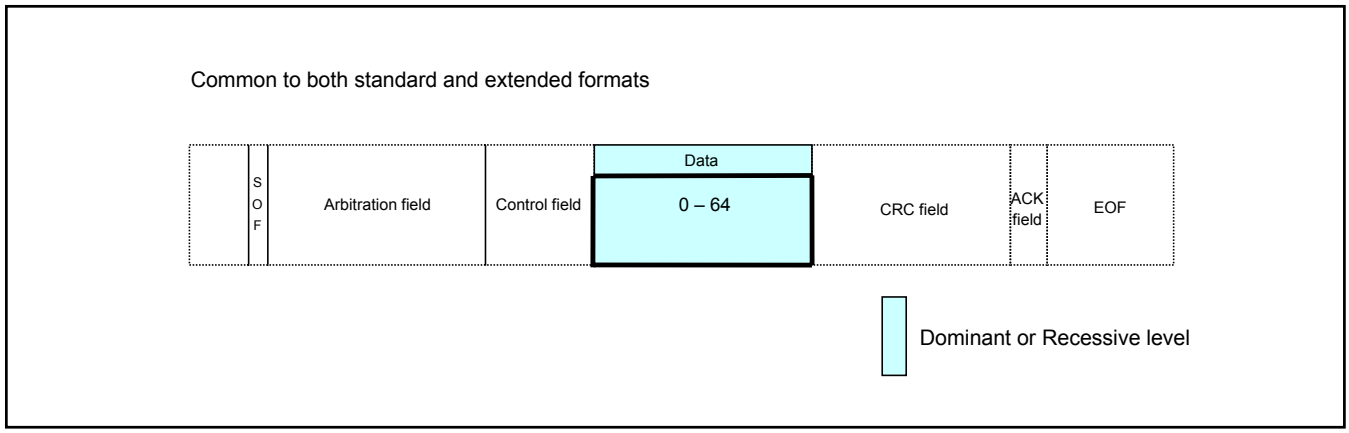


Figure 19. Data Frame (Data Field)

(5) CRC field (common to both standard and extended formats)

This field is used to check the frame for a transmission error. It consists of a 15-bit CRC sequence and a 1-bit CRC delimiter (separating bit).

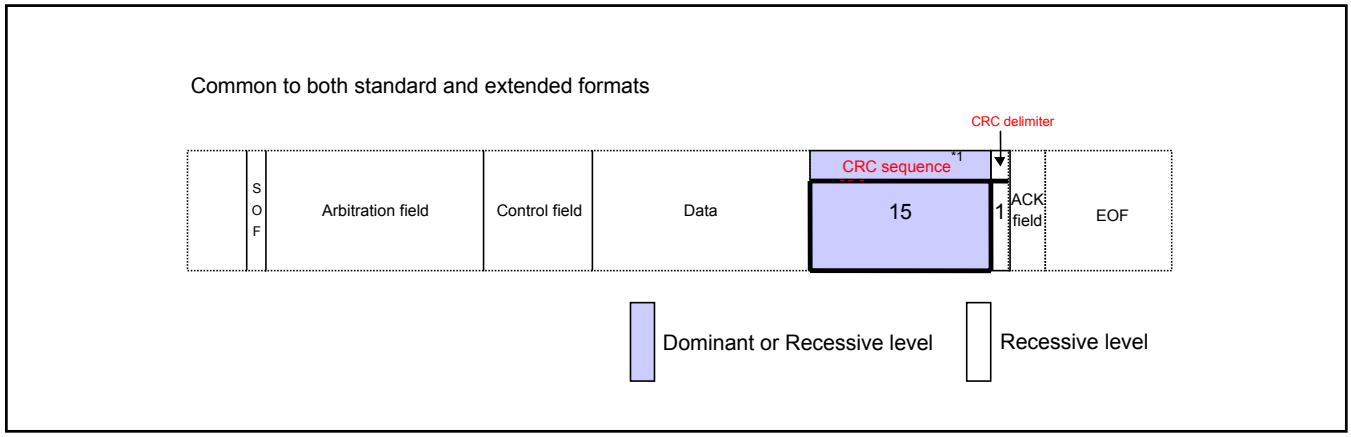


Figure 20. Data Frame (CRC Field)

*1 : CRC sequence

The CRC sequence consists of a CRC value generated by the polynomial P(X) shown below. The CRC calculation range includes the start of frame, arbitration field, control field, and data field. On the receive unit side too, a CRC value is calculated in the same range of fields. The calculated CRC and the received CRC sequence are compared, and if they do not match, an error is assumed.

$P(X) = X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$

(6) ACK field

This field indicates a signal for confirmation that the frame has been received normally. It consists of 2 bits, one for ACK slot and one for ACK delimiter.

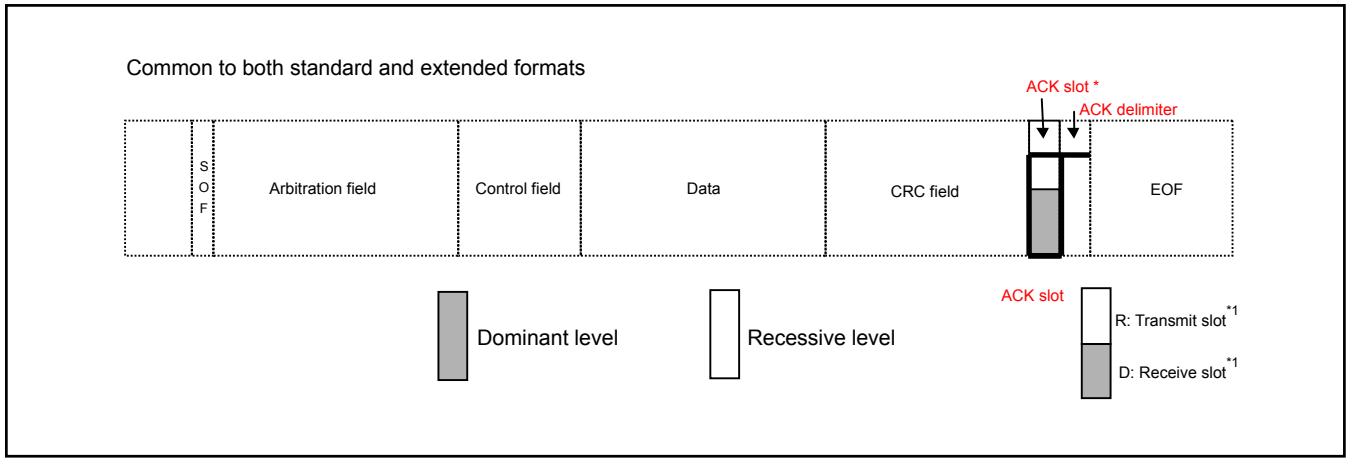


Figure 21. Data Frame (ACK Field)

*1 : ACK field of the transmit unit
The transmit unit sends the ACK slot and ACK delimiter in recessive bits.

*2 : ACK field of the receive unit
The receive unit that has received a correct message sends a dominant bit in the ACK slot of the received frame to notify the transmit unit that it has finished receiving normally. This is referred to as “sending ACK” or “returning ACK.”

Returning an ACK

Of all receive units that are in neither a bus-off state nor a sleep state, only the unit that received a correct message can send an ACK. (The transmit unit does not send an ACK.) No ACKs are returned if no units except the transmit unit exist on the bus that can receive a message. For communication to be established, there must be at least one unit that can receive a message, in addition to the transmit unit. In cases where two or more units exist on the bus that can receive a message, an ACK will be returned if any one of them has received a message normally.

(7) End of frame

This field indicates the end of a frame. It consists of 7 recessive bits.

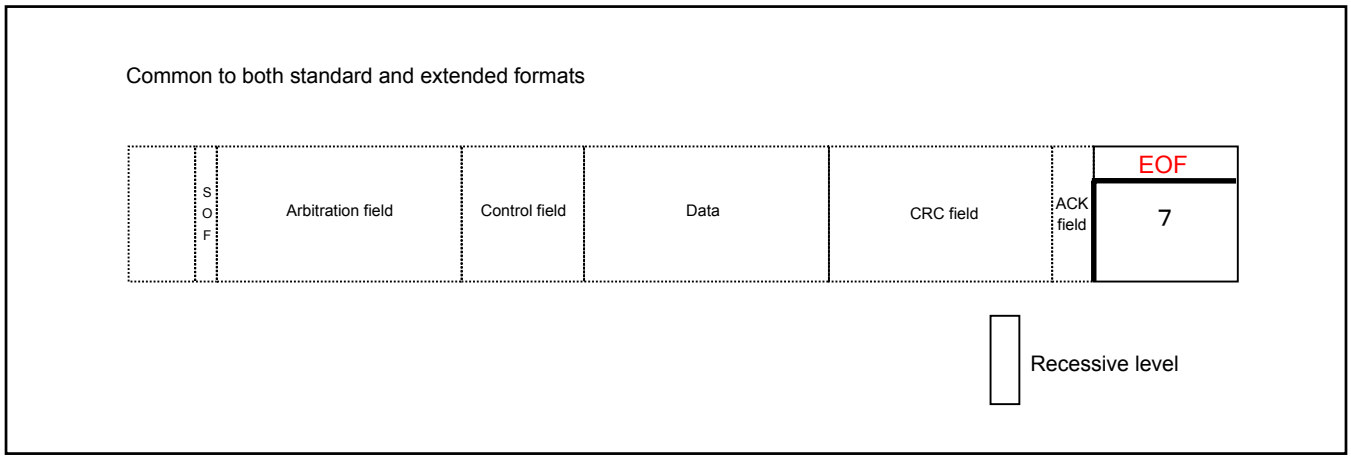


Figure 22. End of Frame (Data Frame)

6.3 Remote Frame

This frame is used by the receive unit to request transmission of a message from the transmit unit. The remote frame consists of six fields. The remote frame is the same as a data frame except that it does not have a data field.
Figure 23 shows the structure of the remote frame.

(1) Start of frame (SOF)
This field indicates the beginning of a frame.

(2) Arbitration field
This field indicates the priority of data. A data frame with the same ID can be requested.

(3) Control field
This field indicates reserved bits and the number of data bytes.

(4) CRC field
This field is used to check the frame for a transmission error.

(5) ACK field
This field indicates a signal for confirmation that the frame has been received normally.

(6) End of frame
This field indicates the end of a remote frame.

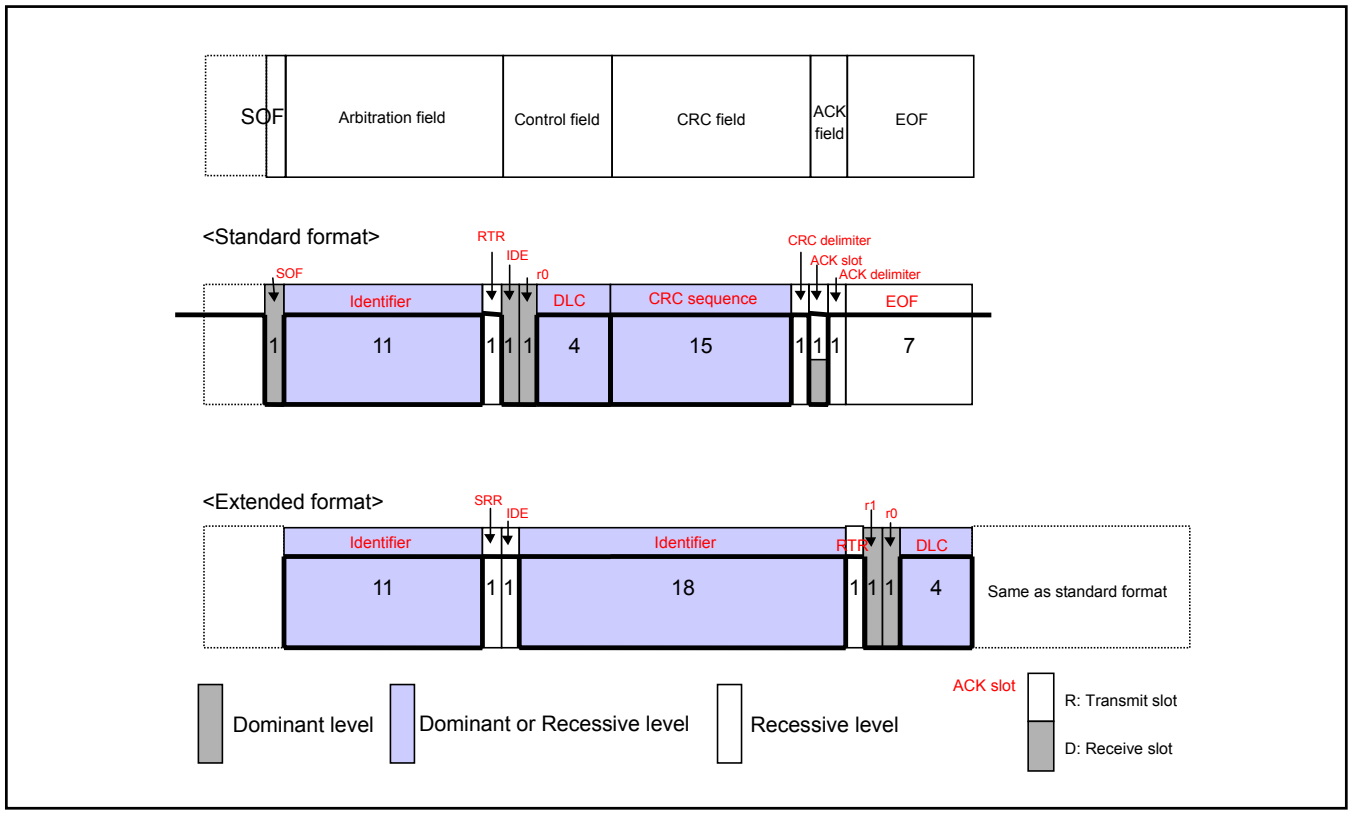


Figure 23. Structure of the Remote Frame

Remote Frame and Data Frame

- Differences between the data frame and remote frame
  - The remote frame differs from a data frame in that it does not have a data field, and that the RTR bit in its arbitration field is of a recessive level.
  - The data frame without a data field and the remote frame can be discriminated by the RTR bit.

- What does the data length code of the remote frame that has no data fields indicate?
  - The value of the data length code of the remote frame indicates the data length code of the corresponding data frame.

- For what is the data frame without a data field used?
  - For example, this data frame may be used by each unit to confirm or respond for connection periodically, or to place real information in the arbitration field itself.

6.4 Error Frame

This frame is used to notify an error that has occurred during transmission. The error frame consists of an error flag and an error delimiter. Error frames are transmitted by the hardware part of CAN.
Figure 24 shows the structure of the error frame.

(1) Error flag

There are two types of error flags: active-error flag and passive-error flag.

- Active-error flag: 6 dominant bits
- Passive-error flag: 6 recessive bits

(2) Error delimiter

The error delimiter consists of 8 recessive bits.

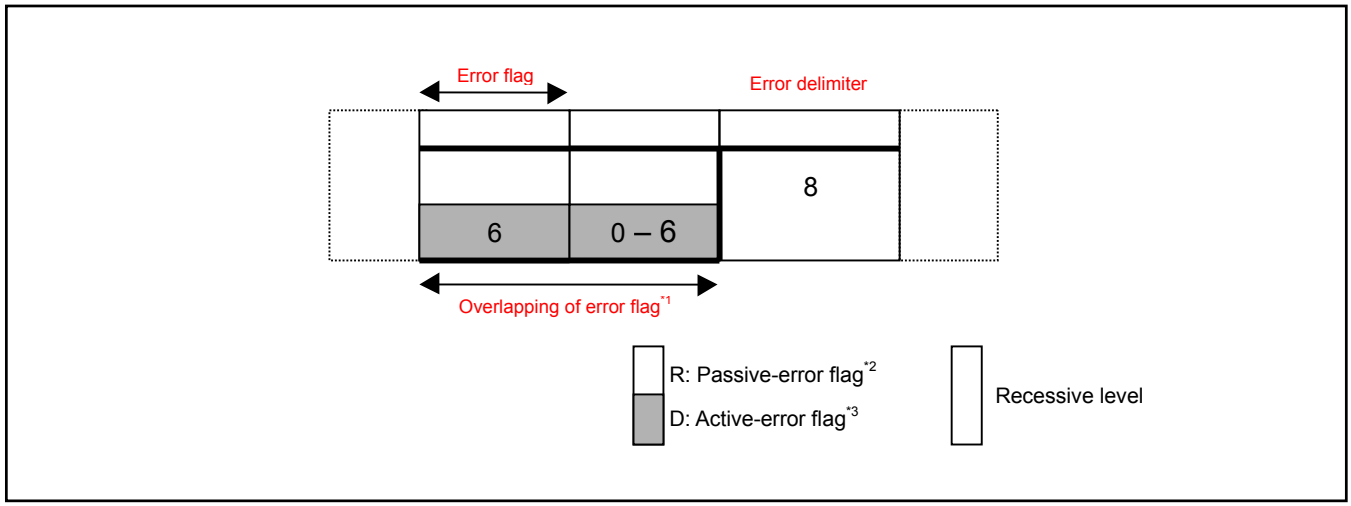


Figure 24. Error Frame

*1 : Error flag overlapping
Depending on the timing at which an error is detected by each unit connected to the bus, error flags may overlap one on top of another, up to 12 bits in total length.

*2 : Passive-error flag
This error flag is output by a unit in an error-passive state when it detected an error.

*3 : Active-error flag
This error flag is output by a unit in an error-active state that it detected an error.

6.5 Overload Frame

The overload frame is used by the receive unit to notify that it has not been prepared to receive frames yet. It consists of an overload flag and an overload delimiter.
Figure 25 shows the structure of the overload frame.

(1) Overload flag

It consists of 6 dominant bits.
The overload flag is structured the same way as the active-error flag of the error frame.

(2) Overload delimiter

It consists of 8 recessive bits.
The overload delimiter is structured the same way as the error delimiter of the error frame.

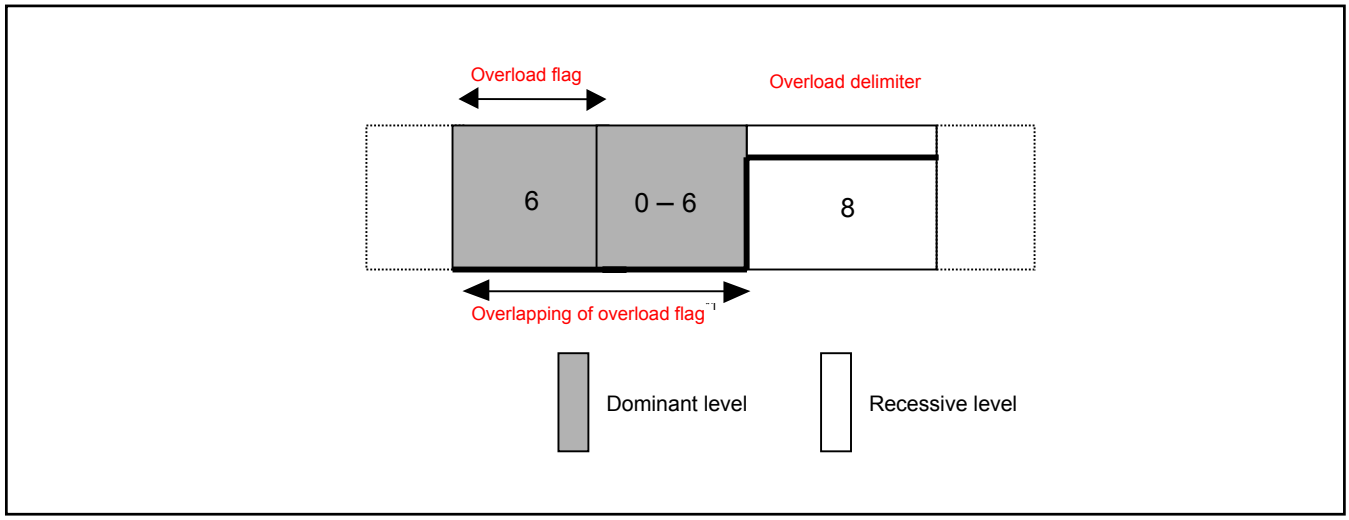


Figure 25. Structure of the Overload Frame

*1 Error flag overlapping

Depending on timing as for the error flag, overload flags may overlap one on top of another, up to 12 bits in total length.

6.6 Interframe Space

This frame is used to separate the data and the remote frames. The data and the remote frames, whichever frame (data, remote, error, or overload frame) may have been transmitted prior to them, are separated from the preceding frame by an inserted interframe space. However, no interframe spaces are inserted preceding overload and error frames.
Figure 26 shows the structure of the interframe space.

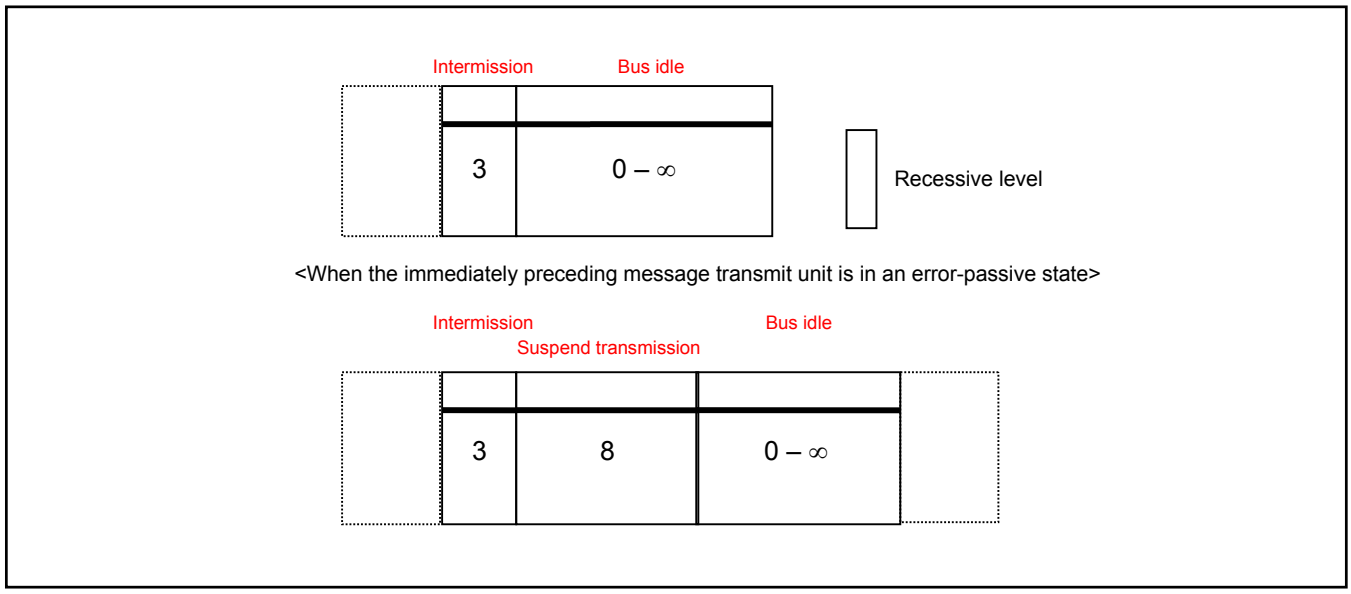


Figure 26. Structure of the Interframe Space

(1) Intermission

Consists of 3 recessive bits.
If a dominant level is detected during an intermission, an overload frame must be transmitted. However, if the third bit of an intermission is of a dominant level, the intermission is recognized as the SOF.

(2) Bus idle

Consists of a recessive level. There are no limitations on its length (it can be zero bits in length).
While in this state, the bus is thought to be free, and any transmit unit can start sending a message.

(3) Suspend transmission (transmission pause period)

Consists of 8 recessive bits.
This field is included in only an interframe space if the immediately preceding message transmit unit was in an error-passive state.

6.7 Priority Resolution by Arbitration

The unit that first output a message during a bus idle state is allowed to send. If multiple units started sending a message at the same time, they are arbitrated for contention by the arbitration field of each transmitted frame beginning with the first bit in it. The unit that output a “dominant level” successively in a greater number of bits than any other units is allowed to send. The units that lost in this arbitration go to a receive operation beginning with the next bit.
Figure 27 shows the mechanism of arbitration.

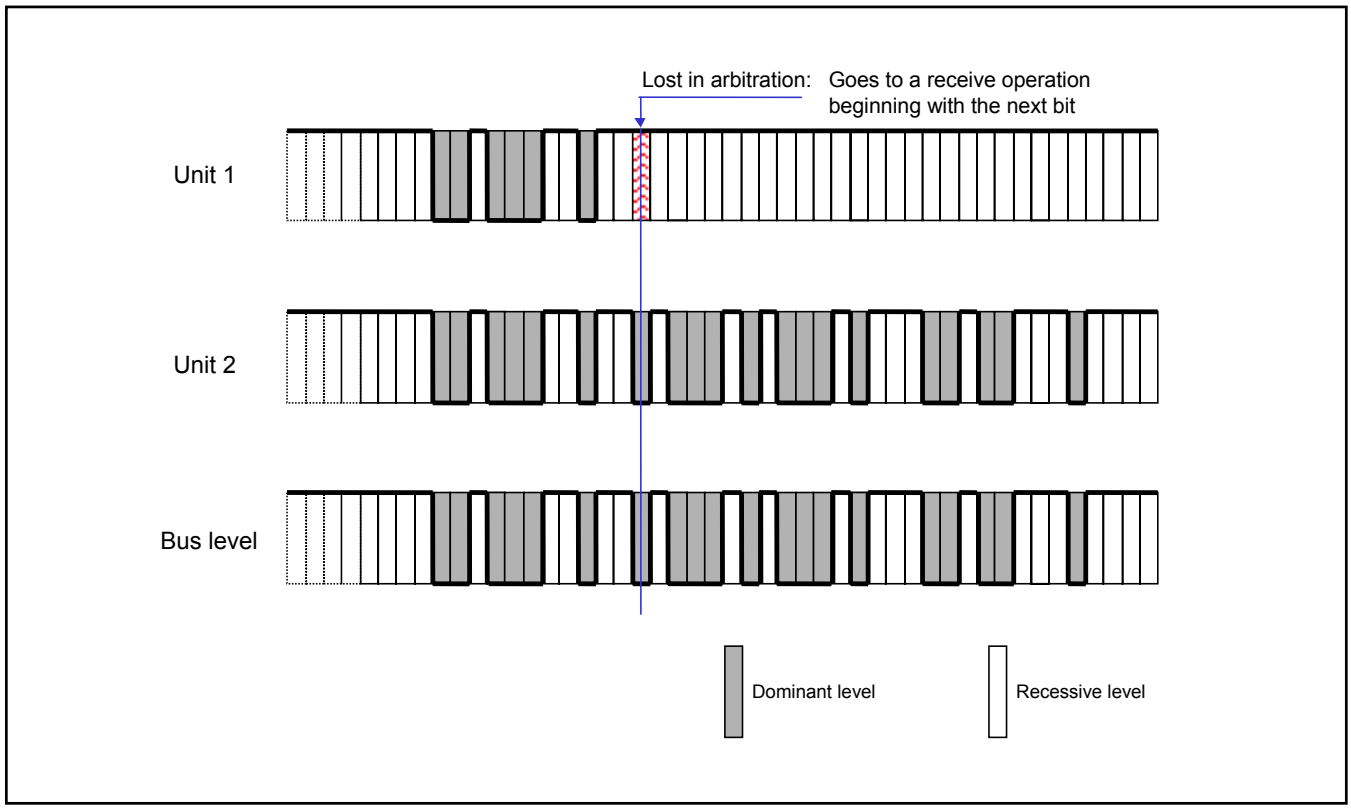


Figure 27. Arbitration

(1) Priority of data and remote frames

If data and remote frames with the same ID contend on the bus, the data frame whose last bit (RTR) in the arbitration field is of a dominant level has priority and is allowed for transmission.

Figure 28 shows the mechanism of arbitration between data and remote frames.

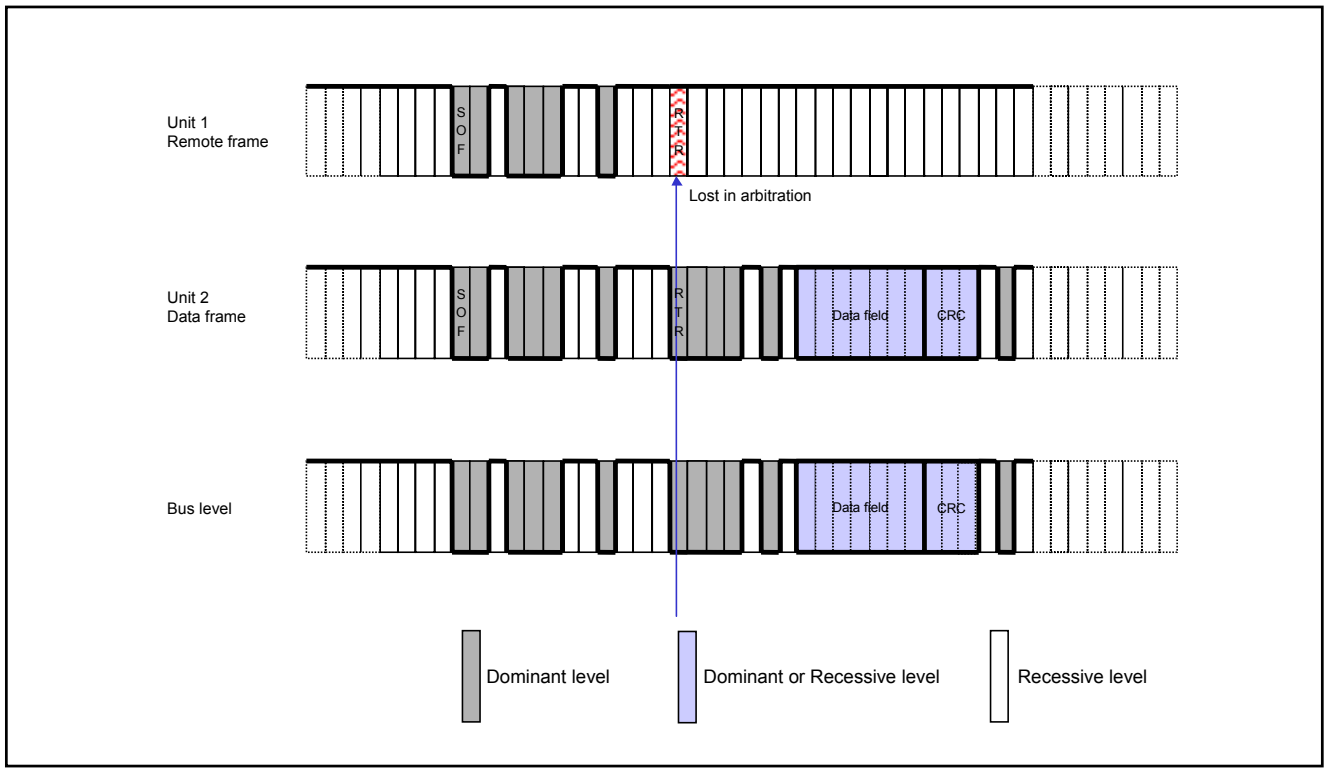


Figure 28. Arbitration between Data and Remote Frames

(2) Priority of standard and extended formats

If data or remote frames in standard and extended formats with the same base ID contend on the bus, the standard format whose RTR bit is of a dominant level has priority and is allowed for transmission. Figure 29 shows the mechanism of arbitration between the standard and extended formats.

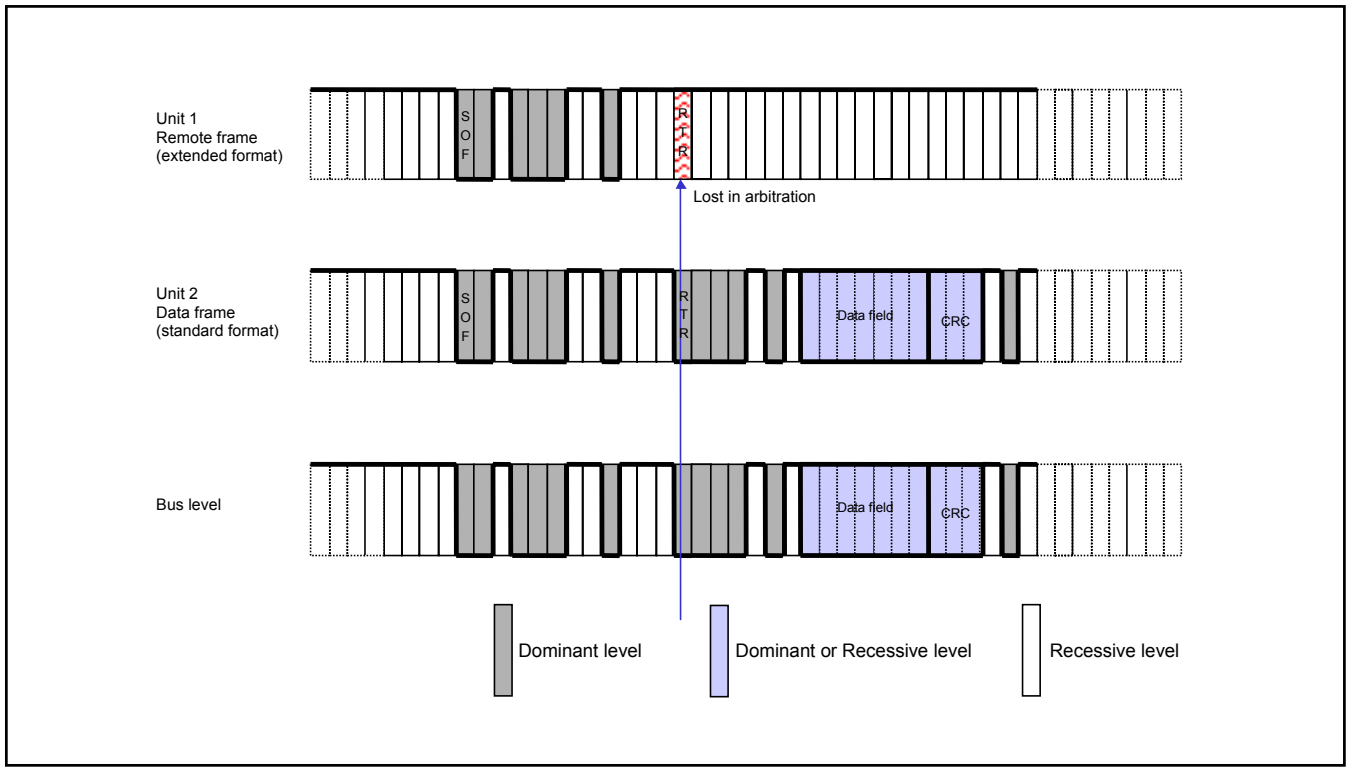


Figure 29. Arbitration between Standard and Extended Formats

6.8 Bit Stuffing

Bit stuffing refers to the function that periodically resynchronizes transmit/receive operations to prevent timing errors between receive nodes from accumulating. If 5 consecutive bits with the same level appear, one bit of inverted data is added.
Figure 30 shows the mechanism of bit stuffing.

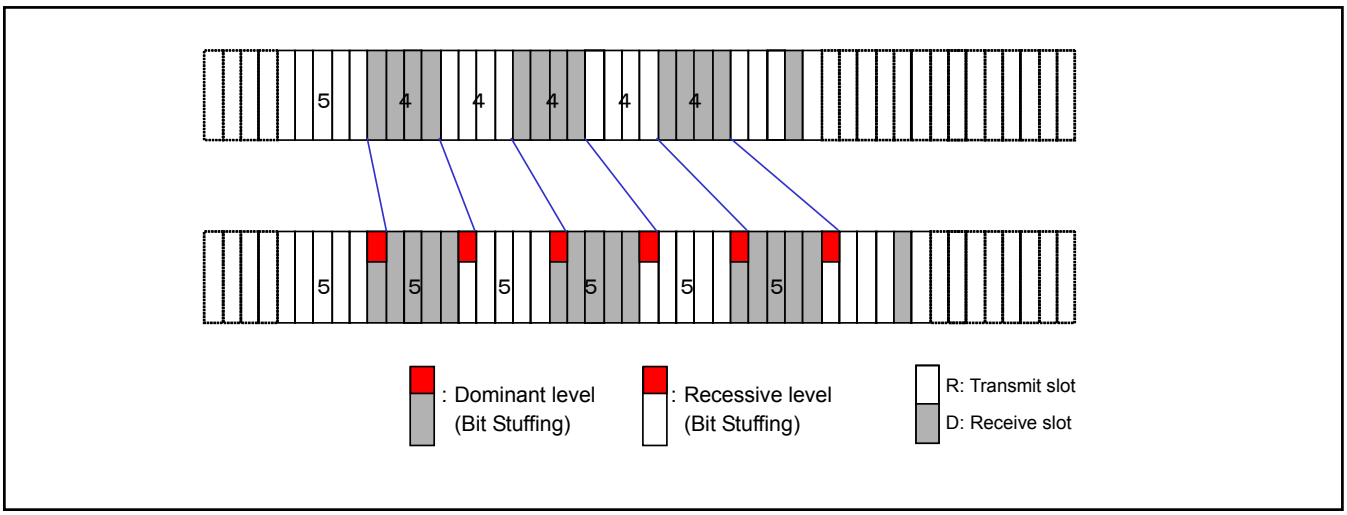


Figure 30. Bit Stuffing

(1) Operation of the transmit unit

If when sending a data or a remote frame, the same level occurs in 5 consecutive bits during the SOF-to-CRC sequence, an inverted level of the preceding 5 bits is inserted in the next bit (6th bit).

(2) Operation of the receive unit

If when receiving a data or a remote frame, the same level occurs in 5 consecutive bits during the SOF-to-CRC sequence, the next bit (6th bit) is deleted from the received frame. If the deleted 6th bit is at the same level as the preceding 5th bit, an error (stuffing error) is detected.

6.9 Types of Errors

There are five types of errors. Two or more errors may occur at the same time.

- Bit error
- Stuffing error
- CRC error
- Form error
- ACK error

Table 8 lists types and contents of errors, target frames in which errors are detected, and the units that detect those errors.

Table 8. Types of Errors

Type of error	Content of error	Target frame (field) in which errors detected	Unit by which errors detected
Bit error	This error is detected when the output level and the data level on bus do not match when they are compared (Level comparison: Dominant output stuffing bits are compared, whereas arbitration fields and ACK bits during transmission are not compared).	• Data frame (SOF to EOF) • Remote frame (SOF to EOF) • Error frame • Overload frame	Transmit unit Receive unit
Stuffing error	This error is detected when the same level of data is detected for 6 consecutive bits in any field that should have been bit-stuffed.	• Data frame (SOF to CRC sequence) • Remote frame (SOF to CRC sequence)	Transmit unit Receive unit
CRC error	This error is detected if the CRC calculated from the received message and the value of the received CRC sequence do not match.	• Data frame (CRC sequence) • Remote frame (CRC sequence)	Receive unit
Form error	This error is detected when an illegal format is detected in any fixed-format bit field.	• Data frame (CRC delimiter, ACK delimiter, EOF) • Remote frame (CRC delimiter, ACK delimiter, EOF) • Error delimiter • Overload delimiter	Transmit unit Receive unit
ACK error	This error is detected if the ACK slot of the transmit unit is found recessive (i.e., the error that is detected when ACK is not returned from the receive unit).	• Data frame (ACK slot) • Remote frame (ACK slot)	Transmit unit

Note 1:

- If while any unit is outputting a recessive level in its arbitration field a dominant level is detected, the unit is interpreted as having lost in arbitration and no bit errors are assumed.
- If while any unit is outputting a recessive level as stuffing bit in its arbitration field a dominant level is detected, no bit errors are detected and a stuffing error is assumed instead.
- If while a transmit unit is outputting a recessive level in its ACK slot a dominant level is detected, this is considered to be an ACK returned from another unit and no bit errors are assumed.
- If while any unit is outputting a passive-error flag a dominant level is detected, the unit waits until the same (dominant or recessive) level is detected in 6 consecutive bits according to error flag complete conditions and no bit errors are assumed.
- For receive units, even when the last bit of EOF (7th bit) is at a dominant level, no form errors are assumed (overload frame transmitted).
- Even when the last bit of error delimiter or overload delimiter (8th bit) is at a dominant level, no form errors are assumed (overload frame transmitted).

6.10 Output Timing of an Error Frame

The unit that detected an error condition outputs an error flag to notify other units of the error. The error flags output at this time are either an active-error flag or a passive-error flag depending on the unit's error status. The transmit unit resends the data or remote frame after it output an error frame.
Table 9 shows the timing with which an error flag is output.

Table 9. Error Flag Output Timing

Type of error	Output timing
Bit error Stuffing error Form error ACK error	The error flag is output beginning with the bit that immediately follows the one in which an error was detected.
CRC error	The error flag is output beginning with the bit next to the ACK delimiter.

6.11 Bit Timing

The number of bits per second that is transmitted without resynchronization by the transmit unit is referred to as the bit rate. One bit is comprised of the following four segments.

- Synchronization segment (SS)
- Propagation time segment (PTS)
- Phase buffer segment 1 (PBS1)
- Phase buffer segment 2 (PBS2)

These segments each are further comprised of minimum units known as Time Quantum, hereafter referred to as Tq.
One message bit is divided into 4 segments, each of which is further divided in units known as Tq. This configuration is referred to as bit timing.
Any desired bit timing can be set by adjusting the number of Tq's that comprise one message bit and the number of Tq's that comprise each segment in it. By setting this bit timing, it is possible to set a sampling point so that multiple units on the bus will sample messages with the same timing. The sampling point is a place at which the bus level is latched and the latched level is determined to be the bit value. This position is the end of PBS1.
Table 10 shows the roles of each segment and the number of Tq's. Figure 31 shows the segment structure of one bit.

Table 10. Types of Segments and Their Roles

<table>
<tr><th>Segment name</th><th>Roles of segment</th><th colspan="2">Number of Tq's</th></tr>
<tr><td>Synchronization segment (SS)</td><td>Multiple units connected to the bus are timed to be coincident during the interval of this segment as they send or receive a frame.
A recessive to dominant or a dominant to recessive transition edge is expected to exist in this segment.</td><td>1</td><td rowspan="4">8 – 25 Tq's</td></tr>
<tr><td>Propagation time segment (PTS)</td><td>This segment absorbs physical delays on network, which include an output delay of the transmit unit, a propagation delay of signal on bus, and an input delay of the receive unit.
The duration of this segment is twice the sum of each delay time.</td><td>1 – 8</td></tr>
<tr><td>Phase buffer segment 1 (PBS1)</td><td rowspan="2">This segment is used to correct for errors when signal edges do not occur within SS.
Since each unit is operating with their own clock, even a minute clock error in each unit will accumulate. This segment absorbs such an error. To absorb a clock error, add or subtract within the SJW setup range for PBS1 and PBS2. Larger the values of PBS1 and PBS2, the greater the tolerance, but the communication speed slows down.</td><td>1 – 8</td></tr>
<tr><td>Phase buffer segment 2 (PBS2)</td><td>PBS1 or IPT*1 whichever is larger*2</td></tr>
<tr><td>Resynchronization jump width (SJW)</td><td>Some units may get out of sync for reasons of a drift in clock frequency or a delay in transmission path. SJW is the maximum width in which this out-of-sync is corrected for.</td><td colspan="2">1 – 4 and ≦PBS1</td></tr>
</table>

*1: IPT stands for Information Processing Time.
This is the time equal to or less than 2 Tq's that is needed for the hardware to fix the bit level immediately after a sampling point.
*2: Since a sampling point exists at the end of PBS1, IPT and PBS2 overlap. When IPT = 2 Tq's, PBS2 cannot be chosen to be 1. Therefore, PBS2 must be 2 to 8 Tq's.

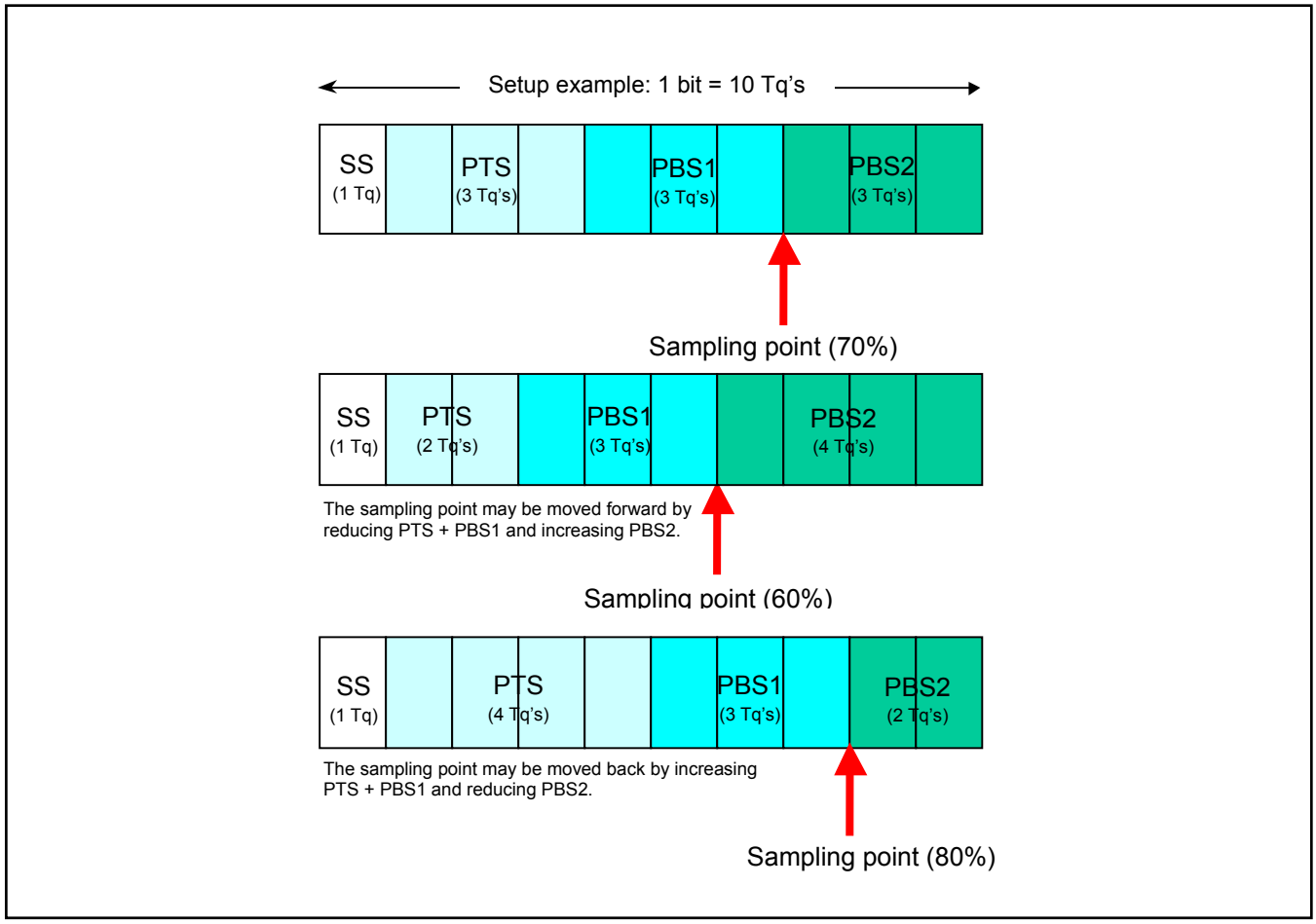


Figure 31. Composition of One Bit

6.12 How Synchronization is Achieved

The communication method adopted for the CAN protocol is NRZ (Non-Return to Zero). No synchronizing signals are attached at the beginning or end of each bit. The transmit unit starts sending frames synchronously with the bit timing. The receive unit synchronizes itself by a change of the bus level as it receives frames.
However, the transmit and the receive units may get out of sync with respect to the other because of a clock error on either side or a phase delay in transmission path (e.g., cable or driver). Therefore, the receive unit adjusts its operation timing by means of hardware synchronization or resynchronization as it receives frames.

6.13 Hardware Synchronization

This synchronization is performed when a receive unit has detected an SOF during a bus idle state. A point in time at which a signal edge from the recessive to the dominant level is detected is recognized as SS irrespective of the SJW value.
Figure 32 shows the mechanism of hardware synchronization.

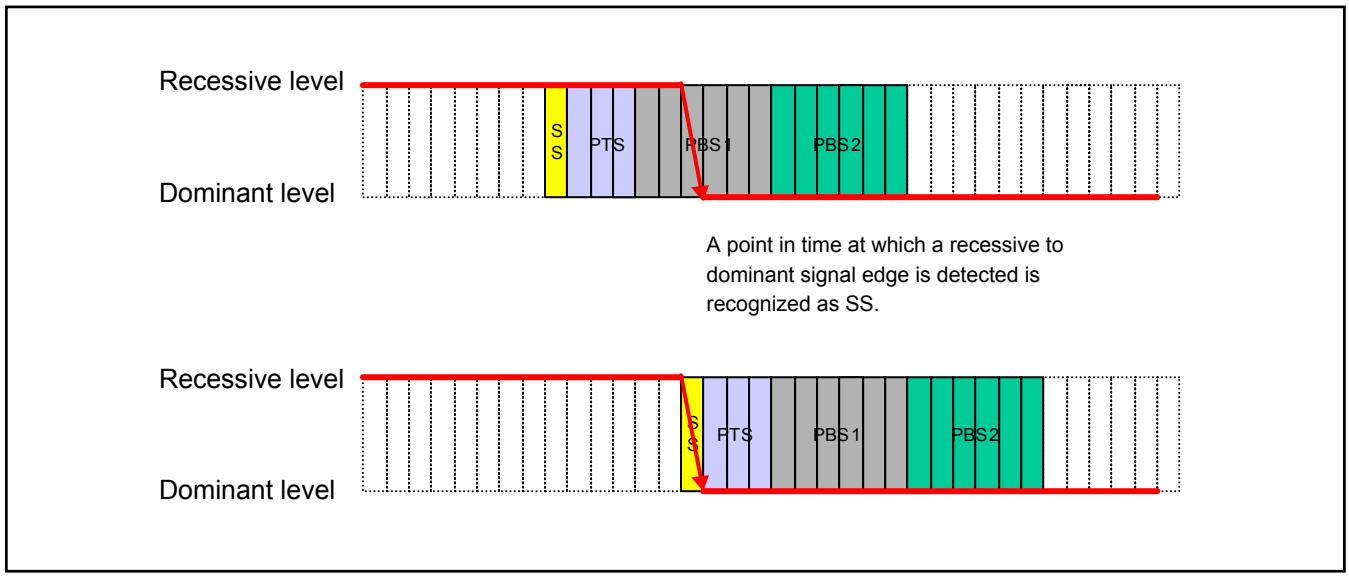


Figure 32. Hardware Synchronization

6.14 Resynchronization

This method of synchronization is performed when a change of levels on the bus is detected during a receive operation.
Each time a signal edge (level change on bus) is detected, transmit/receive units are resynchronized by adjusting the SJW value to expend PBS1 or contract PBS2 according to timing errors. However, if the timing error is greater than the SJW value, it can be corrected for only the SJW value.
Figure 33 shows the mechanism of resynchronization.

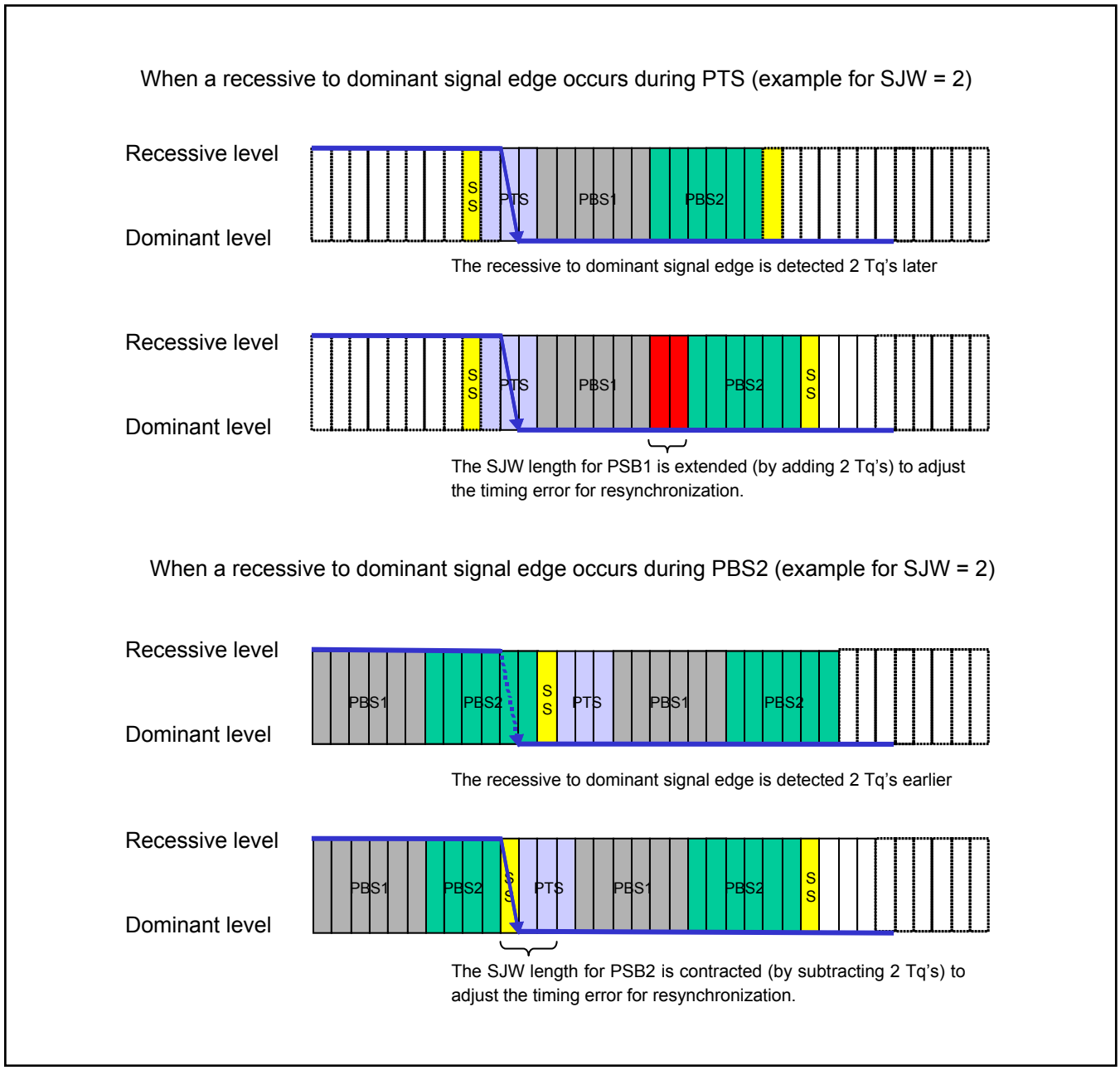


Figure 33. Resynchronization

6.15 Rules of Synchronizations Performed

Hardware synchronization and resynchronization are accomplished following the rules described below.

(1) Synchronization is performed only once in 1 bit (between two sampling points).

(2) Only when the bus level at the previous sampling point and the bus level immediately after a signal edge differ, the signal edge is used for synchronization.

(3) If a recessive to dominant signal edge occurs and the rules in (1) and (2) are satisfied, synchronization is performed.

(4) If a recessive to dominant signal edge occurs during an interframe space (except the first bit in intermission), hardware synchronization is always performed.

(5) If a signal edge from all other recessive levels to a dominant level occurs, resynchronization is performed.

(6) If the dominant level that a transmit unit itself output is observed with a delay, no resynchronization is performed.

Web site and Contact for Support

Renesas Technology Web site
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	2006.04.07	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.