ИНТЕРФЕЙСЫ

6. Стандарт CAN

CAN – общие сведения

Стандарт CAN (Control Area Network) был разработан фирмой Robert Bosch GmbH для использования в автомобильной электронике.

Основное достоинство CAN – высокая надёжность.

Скорость передачи – до 1 мегабит в секунду при длине 60 м.

Расстояние между узлами – до 1 км.

Текст стандарта разделён на части A и B, во многом повторяющиеся. В части B введён расширенный формат кадра.

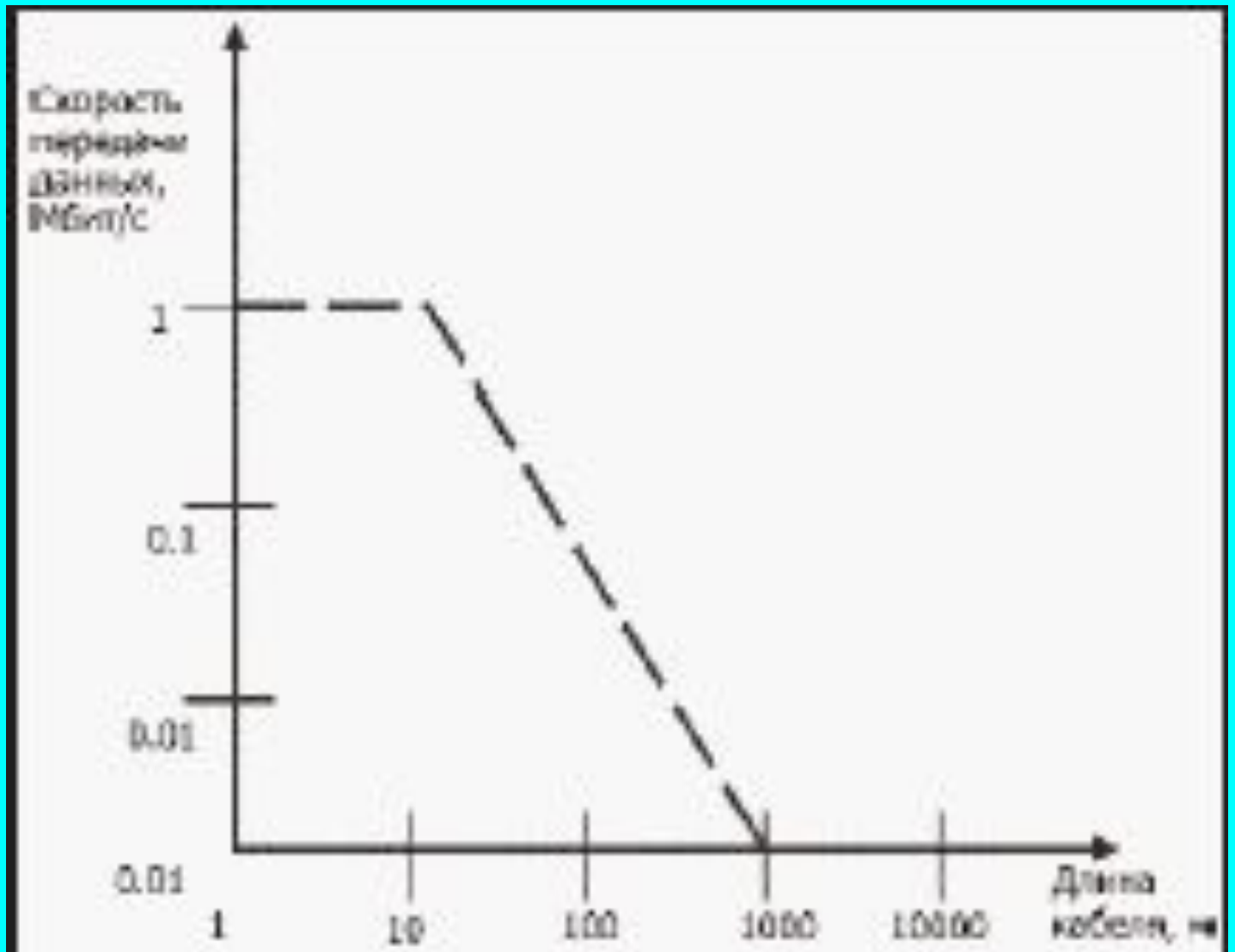


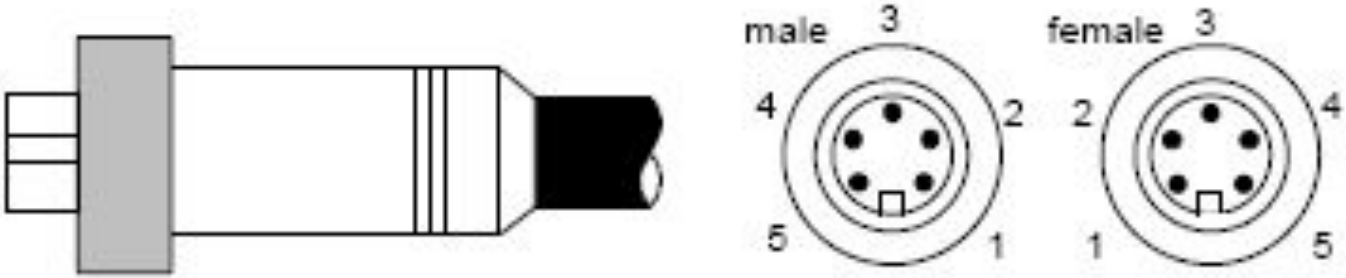
Рис. 6. Скорость передачи данных относительно длины кабеля для CAN-протокола

CAN - свойства

- каждое сообщение имеет определенный приоритет
- существуют гарантированные времена ожидания
- гибкость конфигурации
- групповой приём с временной синхронизацией
- система непротиворечивости данных
- multimaster
- обнаружение и сигнализация ошибок
- автоматическая ретрансляция разрушенных сообщений
- различие между временными ошибками и постоянными отказами узлов и автономное отключение дефектных узлов

Mini Style Connector

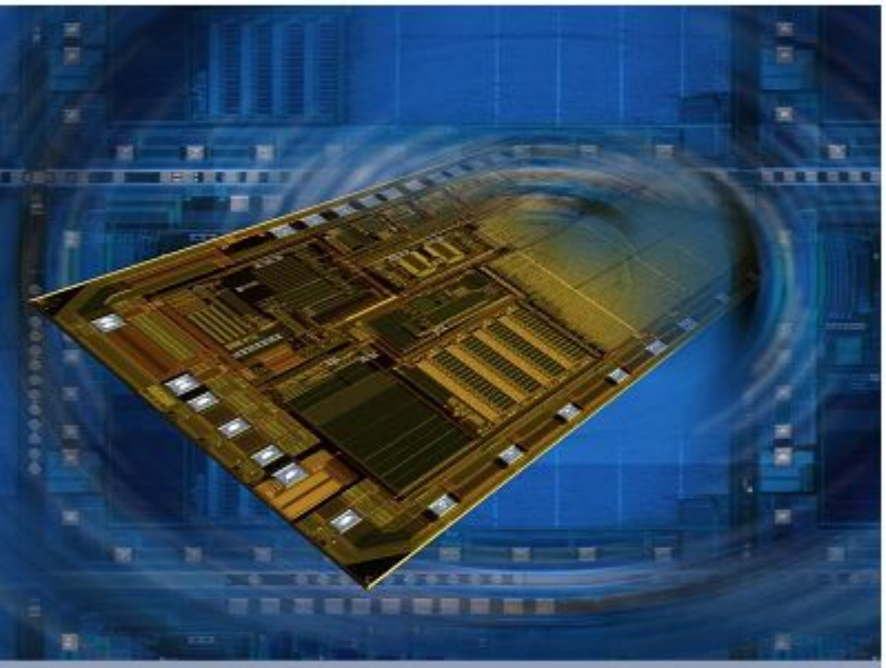
5-pin Mini Style Connector : ANSI/B93.55M-1981



If 5-pin Mini Style Connectors are used the following pinning applies:

Pin	Signal	Description
1	(CAN_SHLD)	Optional CAN Shield
2	(CAN_V+)	Optional CAN external positive supply (dedicated for supply of transceiver and optocouplers, if galvanic isolation of the bus node applies)
3	CAN_GND	Ground / 0V / V-
4	CAN_H	CAN_H bus line (dominant high)
5	CAN_L	CAN_L bus line (dominant low)

BOSCH



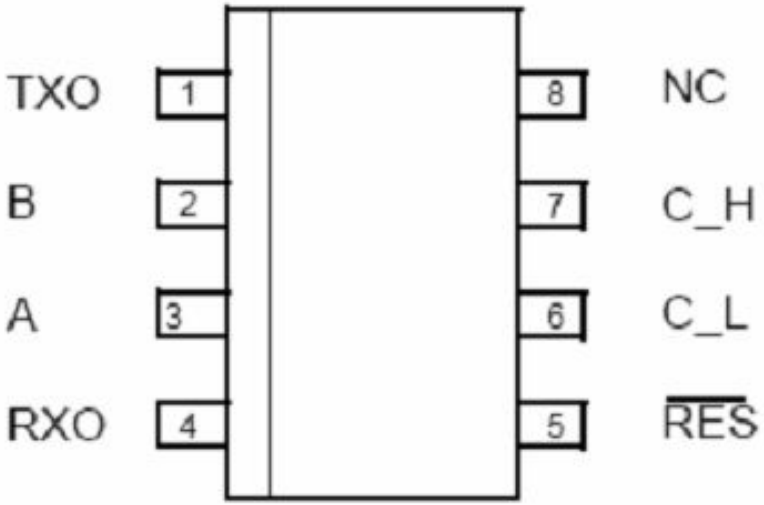
CAN Bus Transceiver

The CF160 is a bidirectional transceiver for signal conditioning and processing in connection with a CAN controller. Data rates of up to 1 MBaud are supported using either shielded or non-shielded pair of lines.

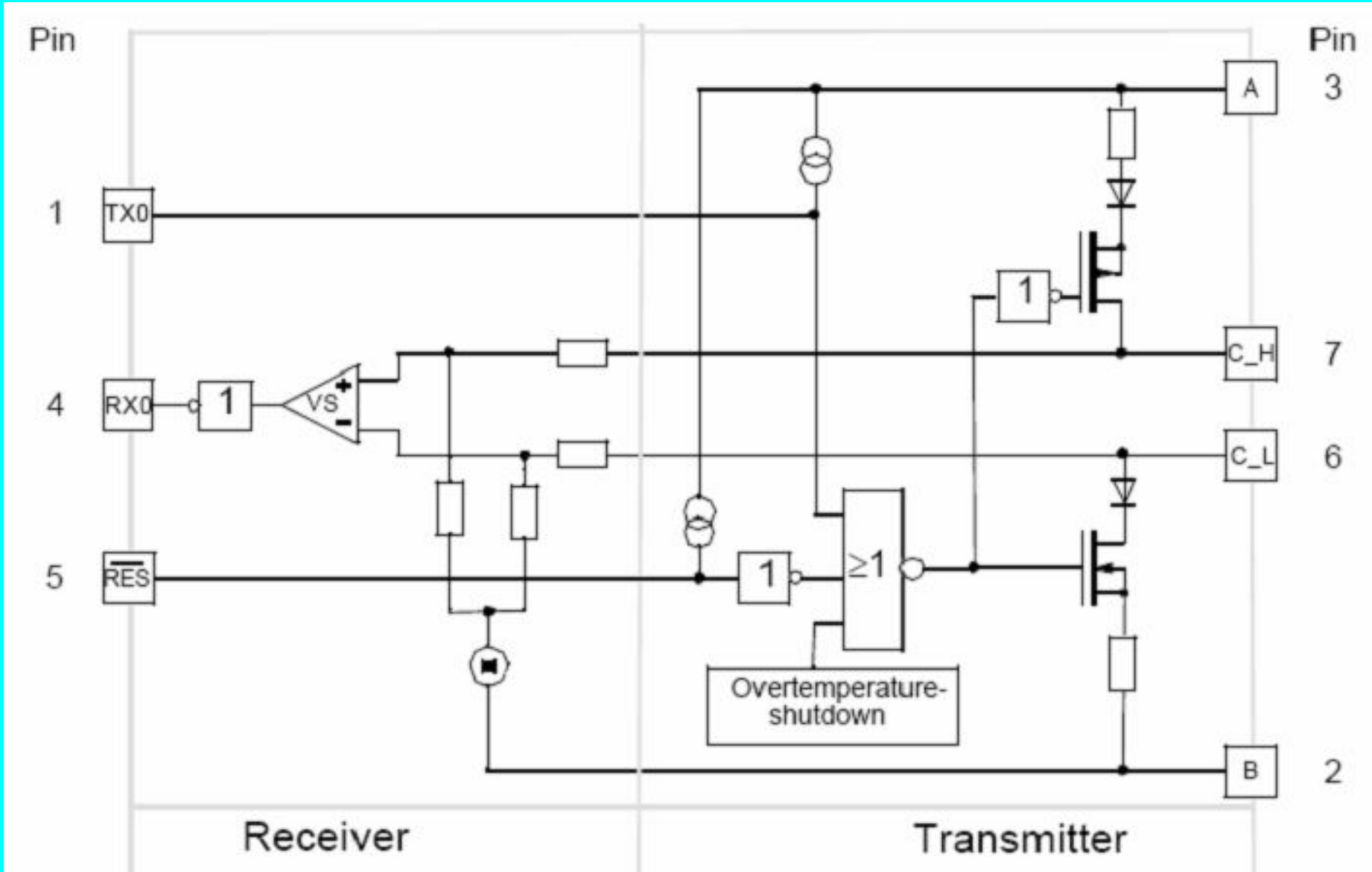
Customer benefits:

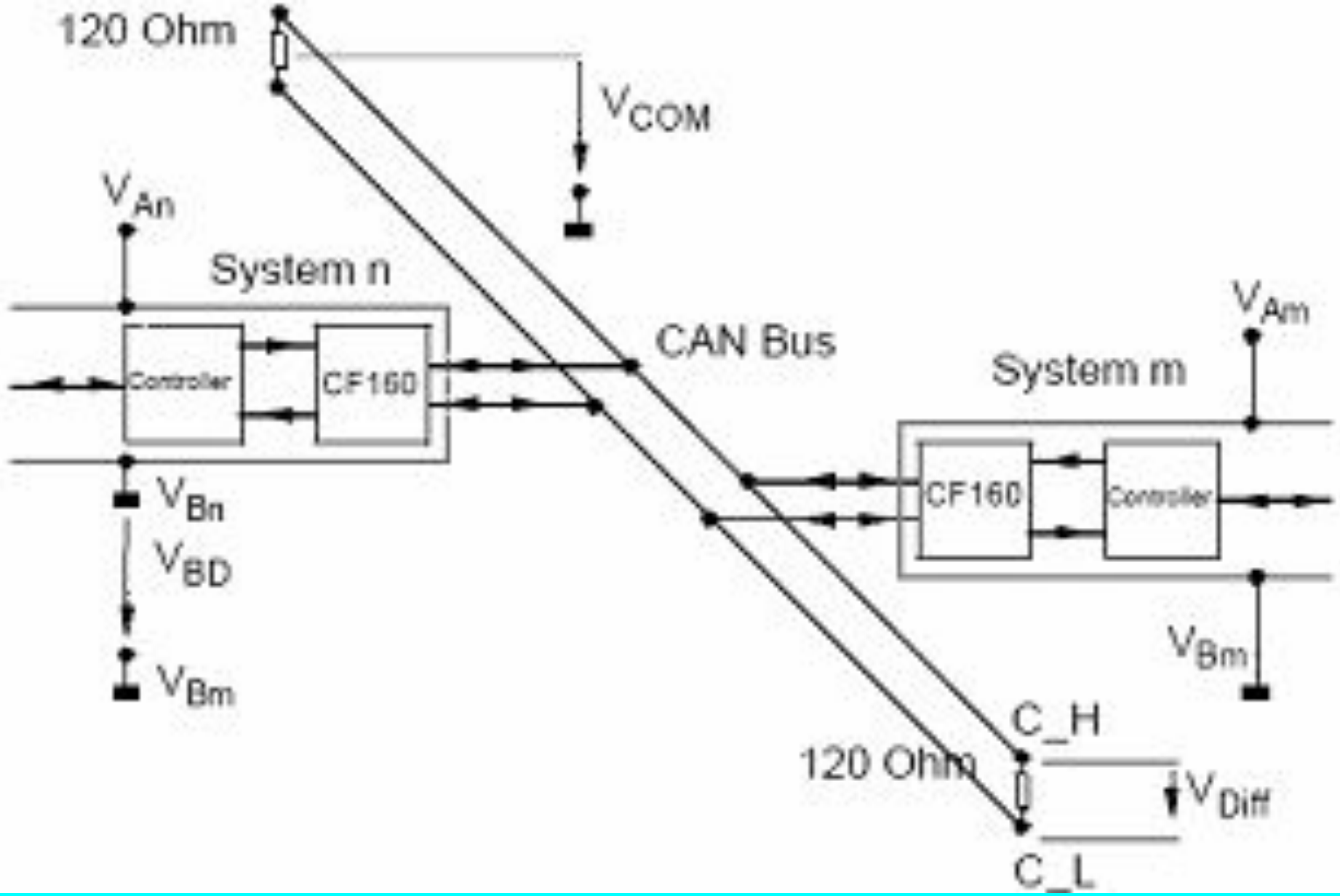
- Excellent system know-how
- Smart concepts for system safety
- Secured supply
- Long- term availability of manufacturing processes and products

PIN configuration



CAN BUS Transceiver

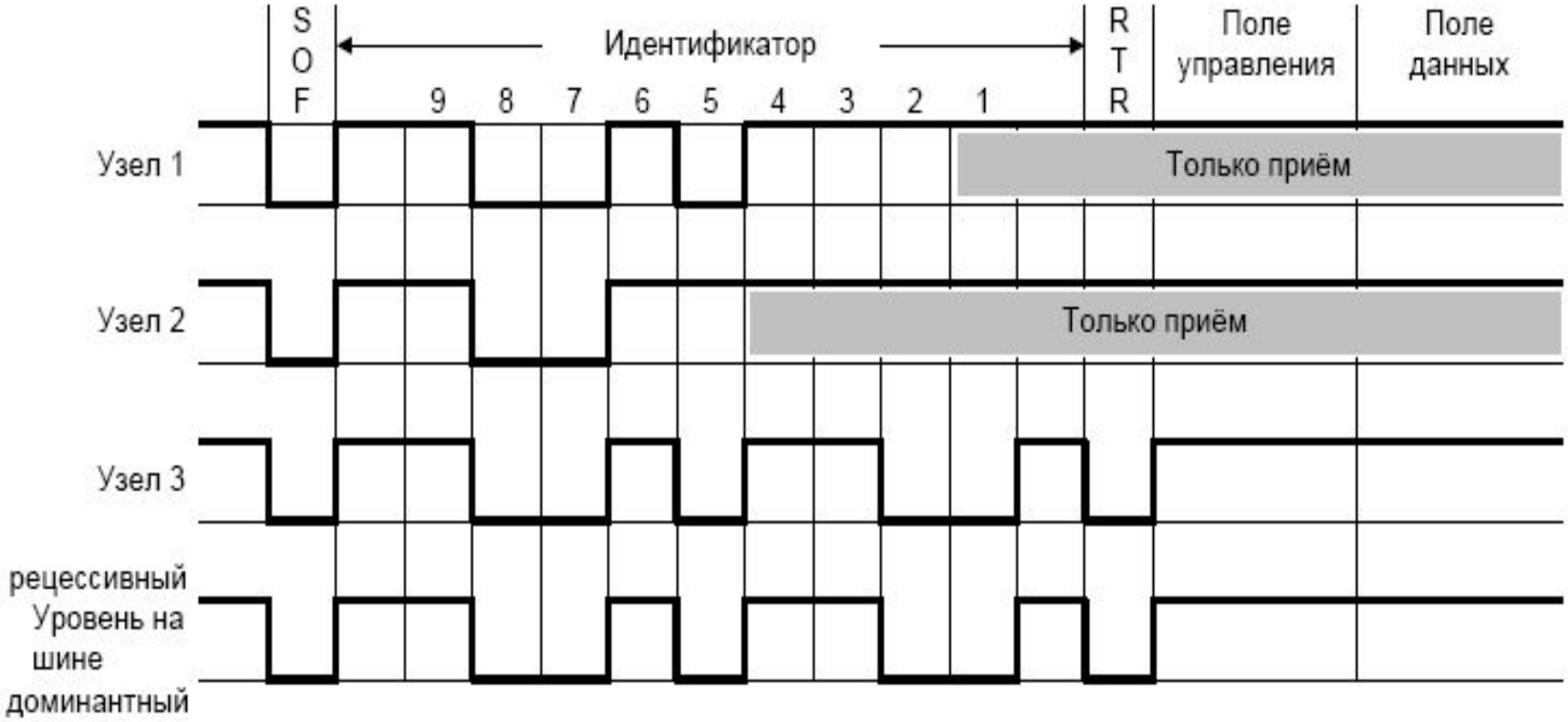



120 Ohm
V_{COM}
V_{An}
System n
Controller
CF160
CAN Bus
V_{Am}
System m
CF160
Controller
V_{Bn}
V_{BD}
V_{Bm}
V_{Bm}
C_H
120 Ohm
V_{Diff}
C_L

CAN – сведения об арбитраже

Когда шина свободна, любой узел может начать передачу сообщения. Если два или больше узла начинают передавать сообщения в одно и тоже время, конфликт при доступе к шине будет решен поразрядным арбитражем используя идентификатор и RTR - бит. Механизм арбитража гарантирует, что ни время, ни информация не будут потеряны. Если кадр данных и кадр удаленного запроса данных начинают передаваться в одно время, то кадр данных имеет более высокий приоритет, чем кадр удаленного запроса данных. В течение арбитража каждый передатчик сравнивает уровень переданного бита с уровнем, считываемым с шины. Если эти уровни одинаковы, узел может продолжать посылать данные дальше. Если был послан уровень лог. '1' (recessive), а с шины считан уровень лог. '0' (dominant), то узел теряет право дальнейшей передачи данных и должен прекратить посылку данных на шину.

Процедура арбитража



Узел 3 выиграл арбитраж и начинает передачу данных

CAN – определение передатчика и приёмника

Передатчик

Узел, передающий сообщение называется передатчиком этого сообщения. Узел является передатчиком до тех пор, пока он не потерял арбитраж.

Приёмник

Узел называется приёмником сообщения, если он не передатчик сообщения и шина занята.

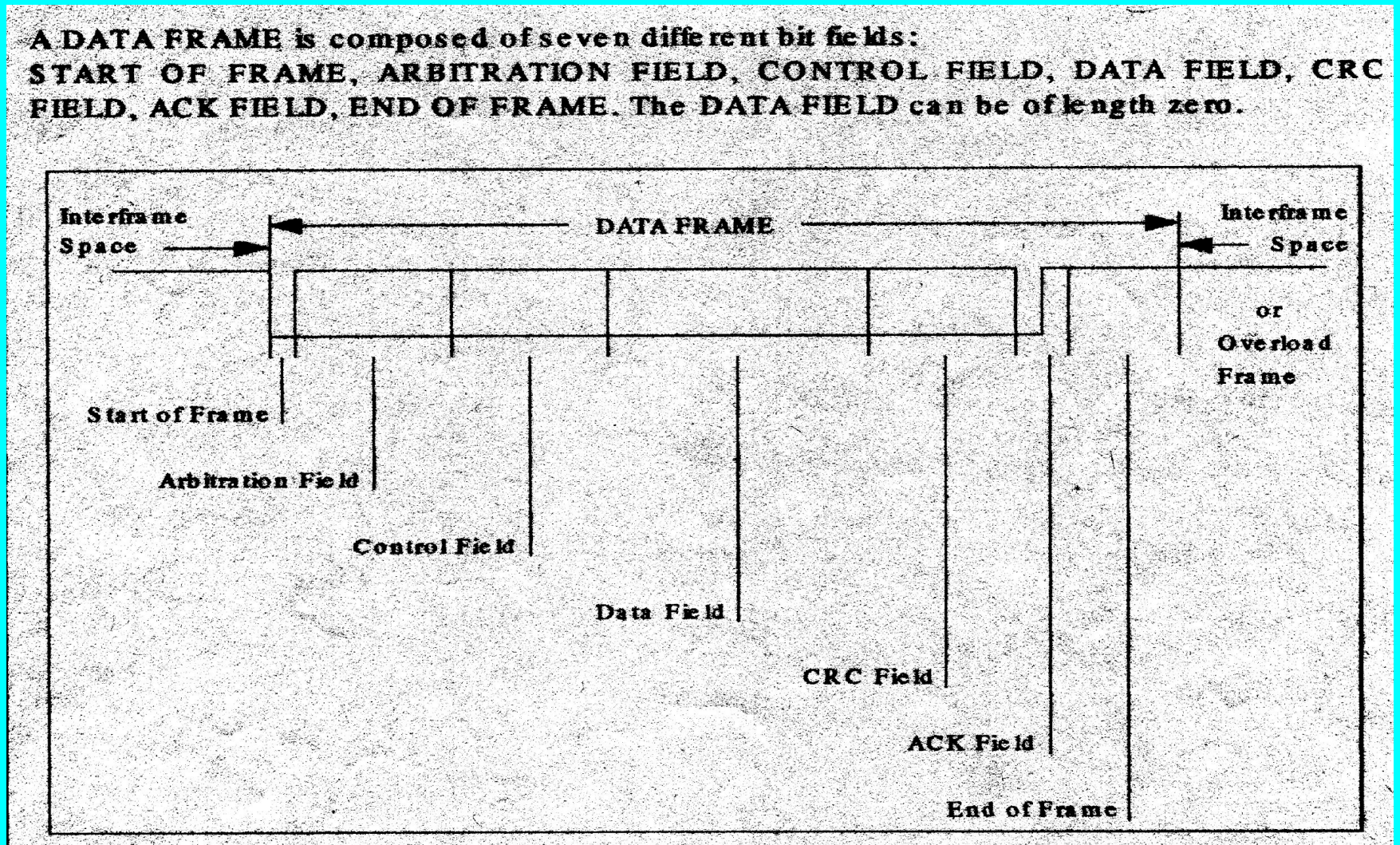
Кадры данных и запроса данных

Шина свободна	SOF	Поле арбитража	Поле управления	Поле данных	Поле CRC	Поле ASK	EOF	Промежуток между кадрами
	1 бит	16/32 бит	6 бит	0...8 байт	16 бит	2 бит	7 бит	3 бита

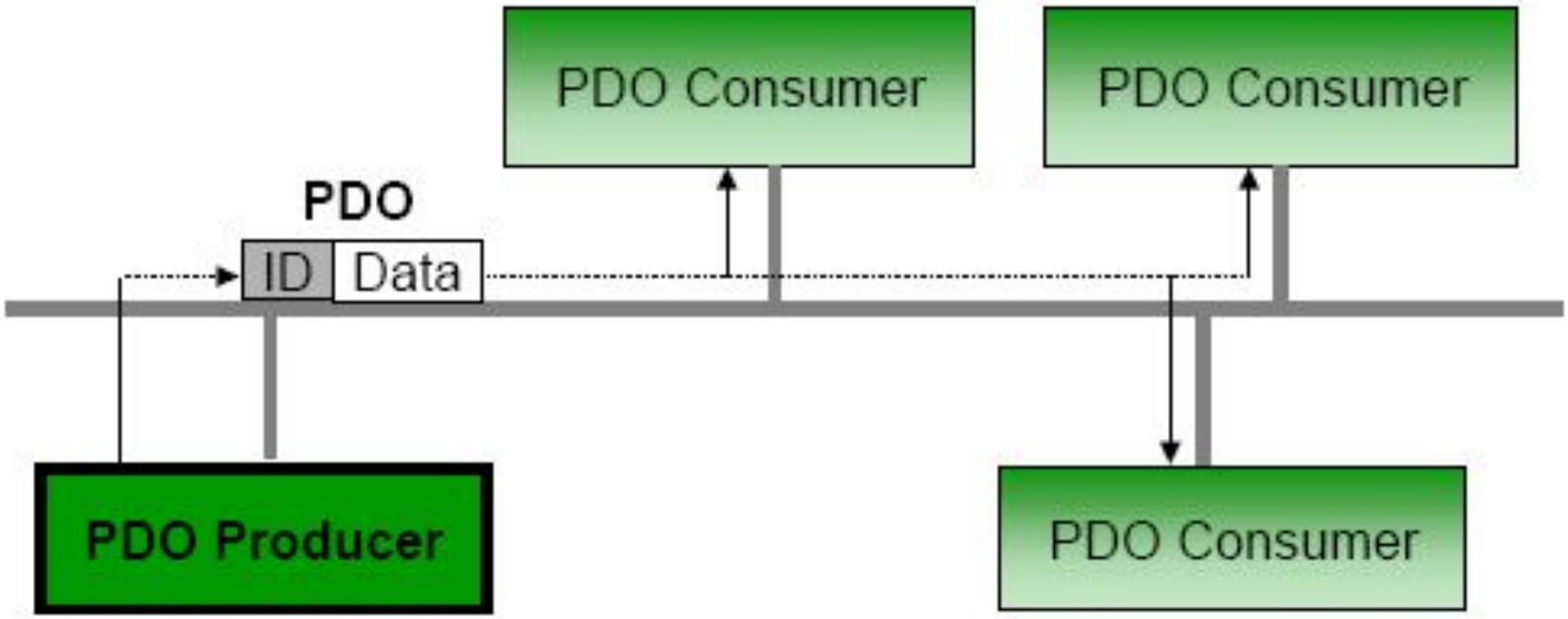
Шина свободна	SOF	Поле арбитража	Поле управления	Поле CRC	Поле ASK	EOF	Промежуток между кадрами
	1 бит	16/32 бит	6 бит	16 бит	2 бит	7 бит	3 бита

CAN – стандартный формат фрейма данных

Кадр данных состоит из 7 различных полей: "Начало кадра", "поле арбитража", "поле контроля", "поле данных", "поле CRC", "поле подтверждения", "конец кадра".

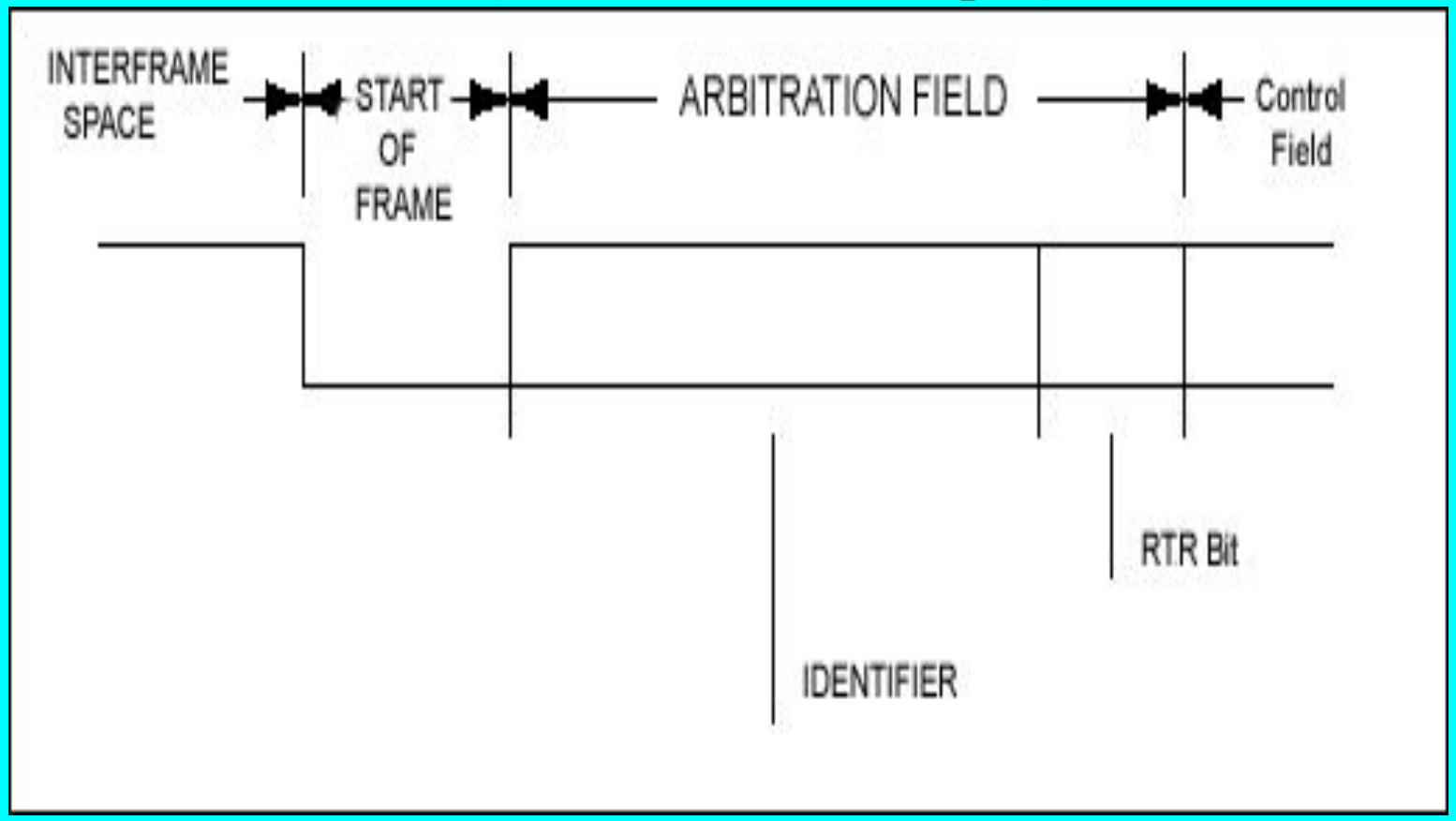


Process Data Object



CAN – старт фрейма и поле арбитража

(по части А стандарта)



<u>Идентификатор (Identifier)</u>

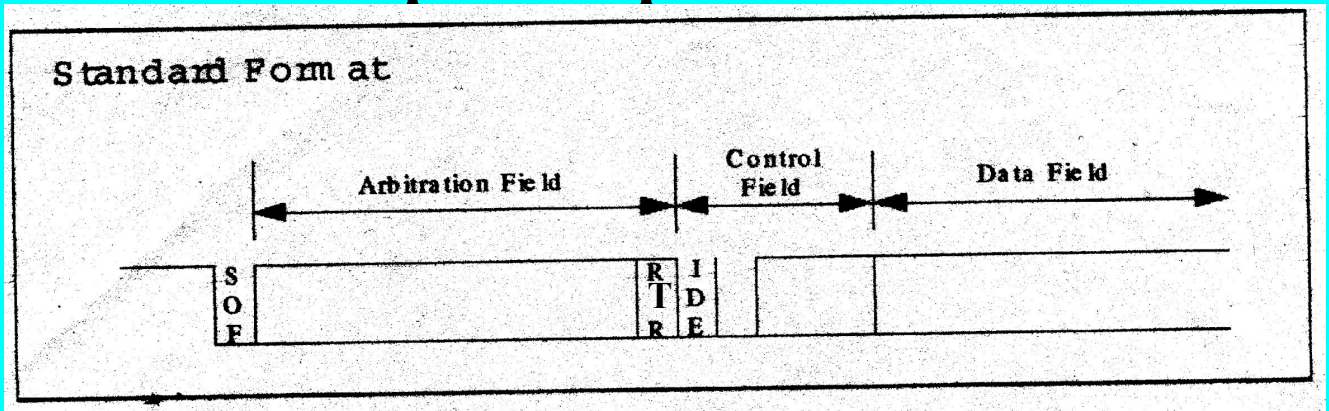
Имеет длину 11 бит. Эти биты должны быть переданы в порядке от ID28до ID18

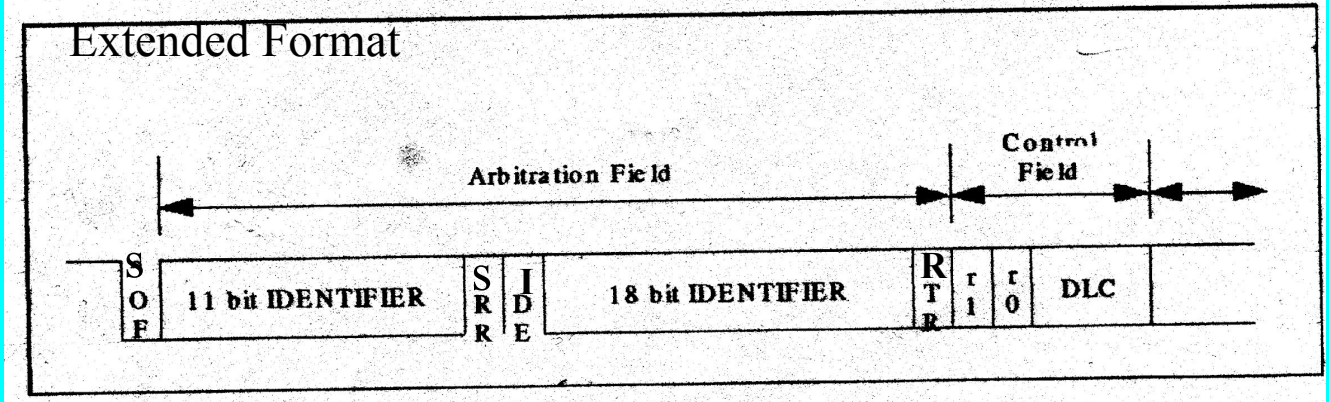
7 старших битов не должны быть все битами с лог '1'.

<u>RTR-бит (RTR - bit)</u>

В кадре данных RTR-бит - "0". Внутри кадра удаленного запроса данных - "1".

CAN – стандартный и расширенный форматы поля арбитража





CAN – различие полей арбитража в стандартном и расширенном форматах

ARBITRATION FIELD

The format of the ARBITRATION FIELD is different for Standard Format and Extended Format Frames.

- In Standard Format the ARBITRATION FIELD consists of the 11 bit IDENTIFIER and the RTR-BIT. The IDENTIFIER bits are denoted ID-28 ... ID-18.

- In Extended Format the ARBITRATION FIELD consists of the 29 bit IDENTIFIER, the SRR-Bit, the IDE-Bit, and the RTR-BIT. The IDENTIFIER bits are denoted ID-28 ... ID-0.

IDENTIFIER

IDENTIFIER - Standard Format

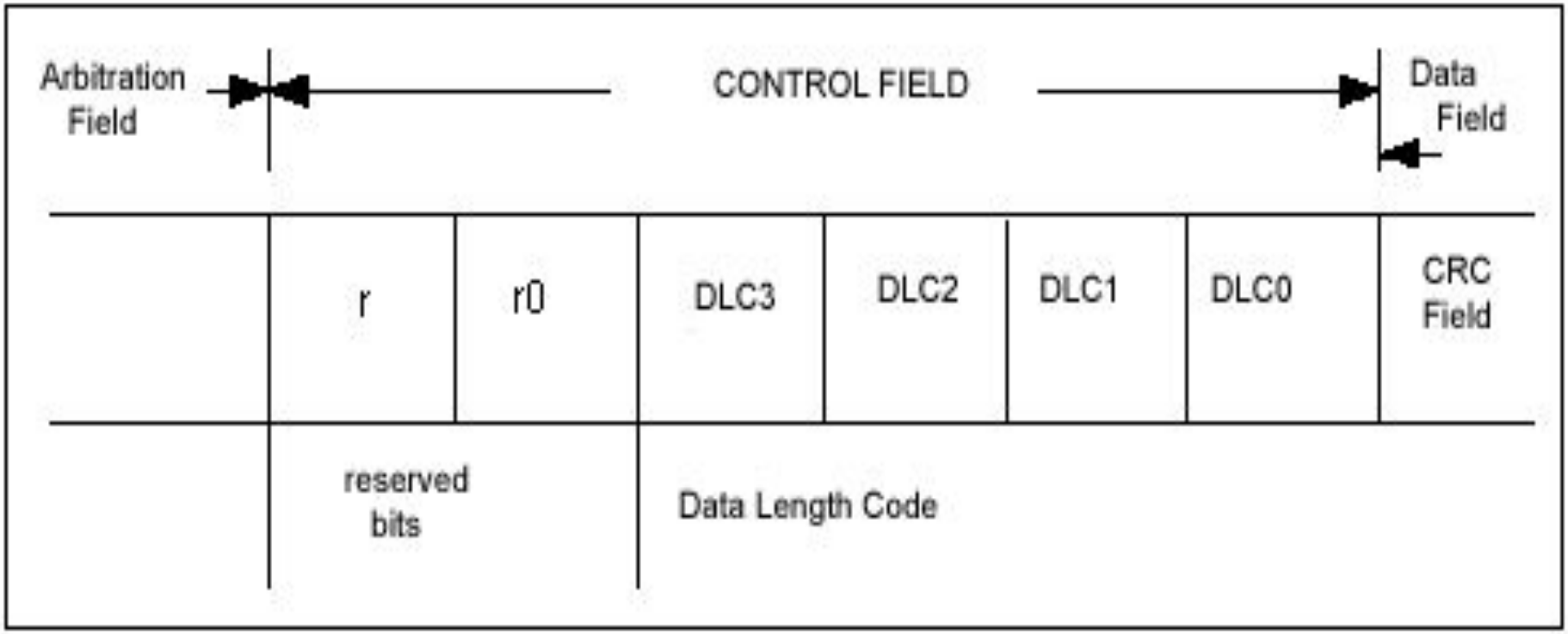
The IDENTIFIERÕ length is 11 bits and corresponds to the Base ID in Extended Format. These bits are transmitted in the order from ID-28 to ID-18. The least significant bit is ID-18. The 7 most significant bits (ID-28 - ID-22) must not be all ÕrecessiveÕ

IDENTIFIER - Extended Format

In contrast to the Standard Format the Extended Format consists of 29 bits. The format comprises two sections:

Base ID with 11 bits and the
Extended ID with 18 bits

CAN – формат управляющего поля



В расширенном формате на месте доминантного бита r1 посылается рецессивный бит IDE

CAN – кодировка длины поля данных

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	d	d

DATA FRAME: admissible numbers of data bytes: {0,1,....,7,8}.
Other values may not be used.

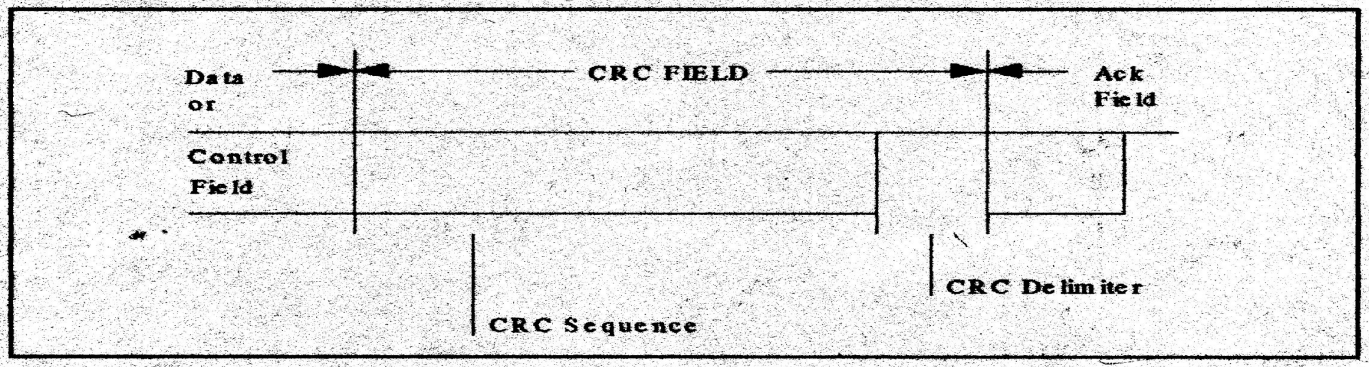
CAN – поле данных и поле CRC

DATA FIELD

The DATA FIELD consists of the data to be transferred within a DATA FRAME. It can contain from 0 to 8 bytes, which each contain 8 bits which are transferred MSB first.

CRC FIELD

contains the CRC SEQUENCE followed by a CRC DELIMITER.



CRC SEQUENCE

The frame check sequence is derived from a cyclic redundancy code best suited for frames with bit counts less than 127 bits (BCH Code).

In order to carry out the CRC calculation the polynomial to be divided is defined as the polynomial, the coefficients of which are given by the destuffed bit stream consisting of START OF FRAME, ARBITRATION FIELD, CONTROL FIELD, DATA FIELD (if present) and, for the 15 lowest coefficients, by 0. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial:

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1.$$

Для вычисления CRC полинома, полином, коэффициенты которого задаются потоком, состоящим из значений, бит полей: "начало кадра ", "поле арбитража", "поле контроля", "поле данных" (если имеется) (самые младшие 15 коэффициентов полинома =0), должен быть разделён полином следующего вида:

x^15+x^14+x^10+x^8+x^7+x^4+x^3+1

CAN – расчёт CRC, ограничитель CRC

The remainder of this polynomial division is the CRC SEQUENCE transmitted over the bus. In order to implement this function, a 15 bit shift register CRC_RG(14:0) can be used. If NXTBIT denotes the next bit of the bit stream, given by the destuffed bit sequence from START OF FRAME until the end of the DATA FIELD, the CRC SEQUENCE is calculated as follows:

```
CRC_RG =0;                                          // initialize shift register
REPEAT
        CRCNXT =NXTBIT EXOR CRC_RG(14);
        CRC_RG(14:1) =CRC_RG(13:0);                 // shift left by
        CRC_RG(0) =0;                               // 1 position

        IF CRCNXT THEN
                CRC_RG(14:0) =CRC_RG(14:0) EXOR (4599hex);
        ENDIF
UNTIL (CRC SEQUENCE starts or there is an ERROR condition)
```

After the transmission / reception of the last bit of the DATA FIELD, CRC_RG contains the CRC sequence.

CRC DELIMITER

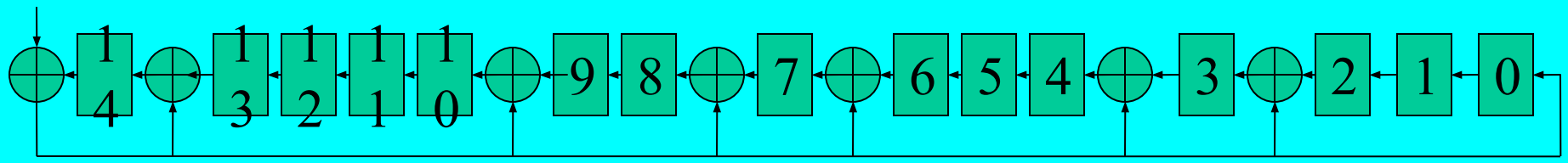
The CRC SEQUENCE is followed by the CRC DELIMITER which consists of a single ÕrecessiveÕbit.

Формирование CRC сдвигающим регистром

В стандарте CAN

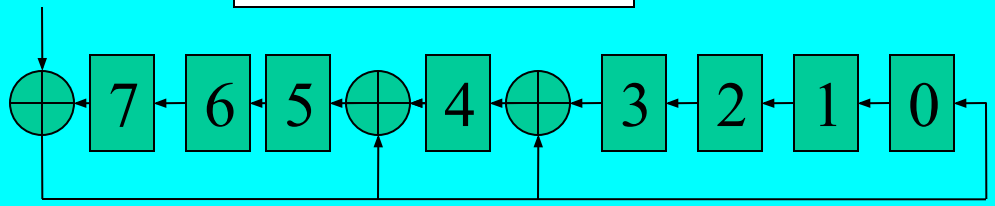
$x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1$

100 0101 1001 1001 B = 4599 H



В стандарте

$x^{8}+x^{5}+x^{4}+1$



V.41 CCITT

$x^{16}+x^{12}+x^{5}+1$

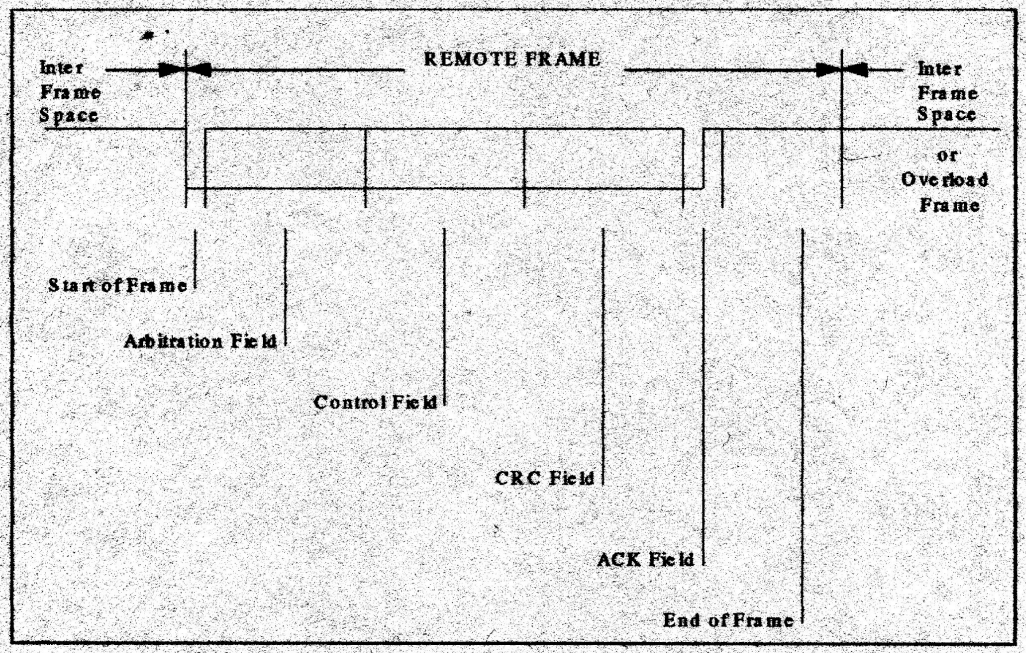
IEEE 802 и др.

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}\ x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

Кадр удаленного запроса

3.1.2 REMOTE FRAME

A station acting as a RECEIVER for certain data can initiate the transmission of the respective data by its source node by sending a REMOTE FRAME.
A REMOTE FRAME is composed of six different bit fields:
START OF FRAME, ARBITRATION FIELD, CONTROL FIELD, CRC FIELD, ACK FIELD, END OF FRAME.
Contrary to DATA FRAMEs, the RTR bit of REMOTE FRAMEs is ÕrecessiveÕ There is no DATA FIELD, independent of the values of the DATA LENGTH CODE which may be signed any value within the admissible range 0...8. The value is the DATA LENGTH CODE of the corresponding DATA FRAME.

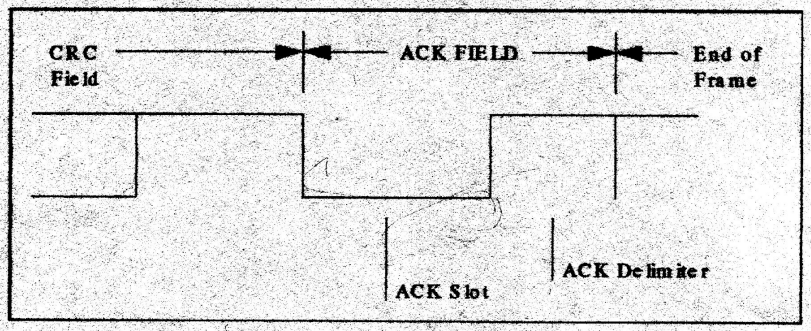


The polarity of the RTR bit indicates whether a transmitted frame is a DATA FRAME (RTR bit ÕdominantÕ) or a REMOTE FRAME (RTR bit ÕrecessiveÕ).

CAN – конец фрейма данных и удалённого запроса

ACK FIELD

The ACK FIELD is two bits long and contains the ACK SLOT and the ACK DELIMITER. In the ACK FIELD the transmitting station sends two ÕrecessiveÕbits.
A RECEIVER which has received a valid message correctly, reports this to the TRANSMITTER by sending a ÕdominantÕbit during the ACK SLOT (it sends ÕACKÕ).



ACK SLOT

All stations having received the matching CRC SEQUENCE report this within the ACK SLOT by superscribing the ÕrecessiveÕbit of the TRANSMITTER by a ÕdominantÕbit.

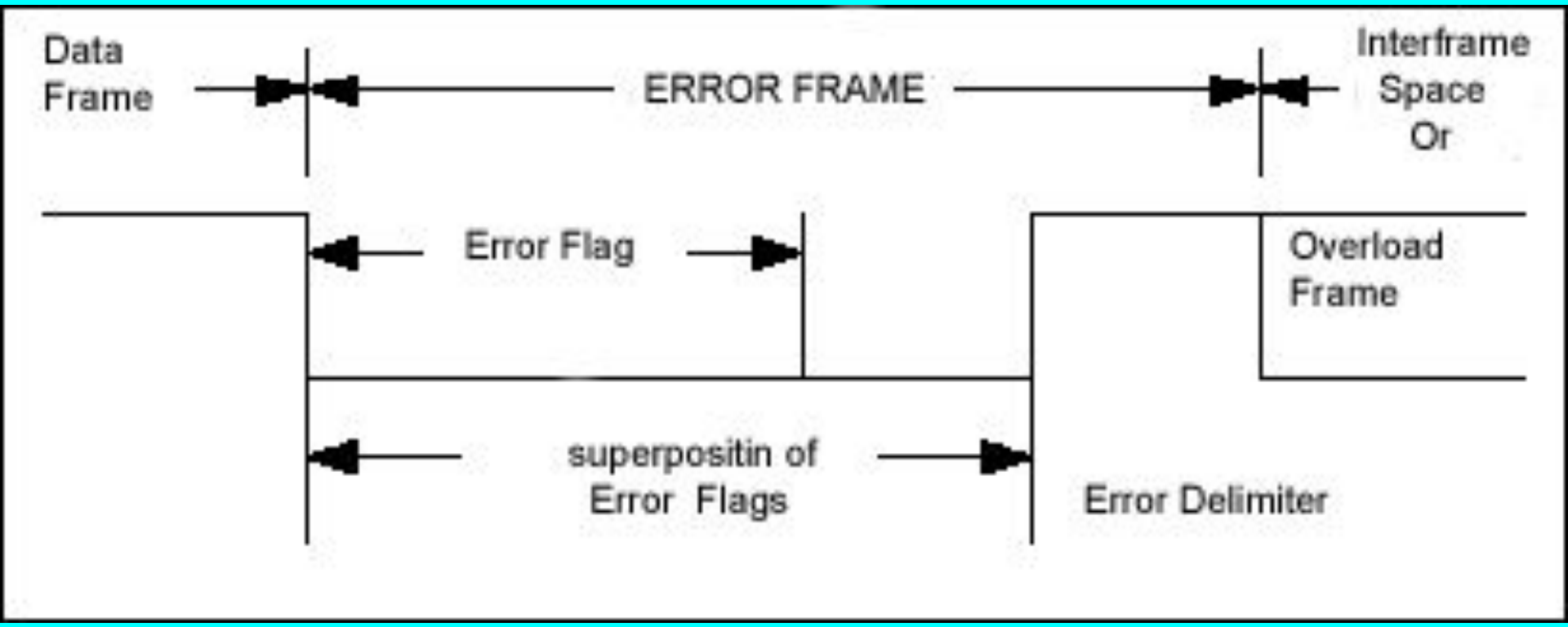
ACK DELIMITER

The ACK DELIMITER is the second bit of the ACK FIELD and has to be a ÕrecessiveÕ bit. As a consequence, the ACK SLOT is surrounded by two ÕrecessiveÕbits (CRC DELIMITER, ACK DELIMITER).

END OF FRAME

Each DATA FRAME and REMOTE FRAME is delimited by a flag sequence consisting of seven ÕrecessiveÕbits.

CAN – фрейм ошибки



Состоит из двух различных полей. Первое поле является суперпозицией флагов ошибки различных узлов, второе поле - поле разделителя ошибки(8 бит).

ограничитель ошибки состоит из **восьми рецессивных битов**.

CAN – пять видов ошибок

1.**разрядная ошибка**
Узел, который посылает что - либо на шину также контролирует шину. Разрядная ошибка может быть обнаружена во время передачи бита, если переданное значение отличается от значения, прочитанного с шины.
2. **ошибка заполнения**
Ошибка заполнения обнаруживается во время приема последовательности из шести бит одинакового разрядного уровня в поле сообщения, которое должно быть кодировано методом разрядного заполнения.
3. **ошибка CRC**
Последовательность CRC состоит из результата вычисленного передатчиком. Приёмники вычисляют CRC таким же образом, как и передатчик. Ошибка CRC обнаруживается при несовпадении расчетного результата CRC - последовательности в приёмнике и присланной CRC - последовательности передатчика.
4.**ошибка формата**
Ошибка формата обнаруживается, когда разрядное поле фиксированного формата содержит один или несколько лишних бит.
5. **ошибка подтверждения**
Ошибка подтверждения обнаруживается передатчиком всякий раз, когда нет контроля бита с лог. "0" в течение области подтверждения.

CAN –флаг ошибки

ERROR FLAG

There are 2 forms of an ERROR FLAG: an ACTIVE ERROR FLAG and a PASSIVE ERROR FLAG.

1. The ACTIVE ERROR FLAG consists of six consecutive ÕdominantÕbits.

2. The PASSIVE ERROR FLAG consists of six consecutive ÕrecessiveÕbits unless it is overwritten by ÕdominantÕbits from other nodes.

An Õerror activeÕstation detecting an error condition signals this by transmission of an ACTIVE ERROR FLAG. The ERROR FLAGÕ form violates the law of bit stuffing (see CODING) applied to all fields from START OF FRAME to CRC DELIMITER or destroys the fixed form ACK FIELD or END OF FRAME field. As a consequence, all other stations detect an error condition and on their part start transmission of an ERROR FLAG. So the sequence of ÕdominantÕbits which actually can be monitored on the bus results from a superposition of different ERROR FLAGs transmitted by individual stations. The total length of this sequence varies between a minimum of six and a maximum of twelve bits.

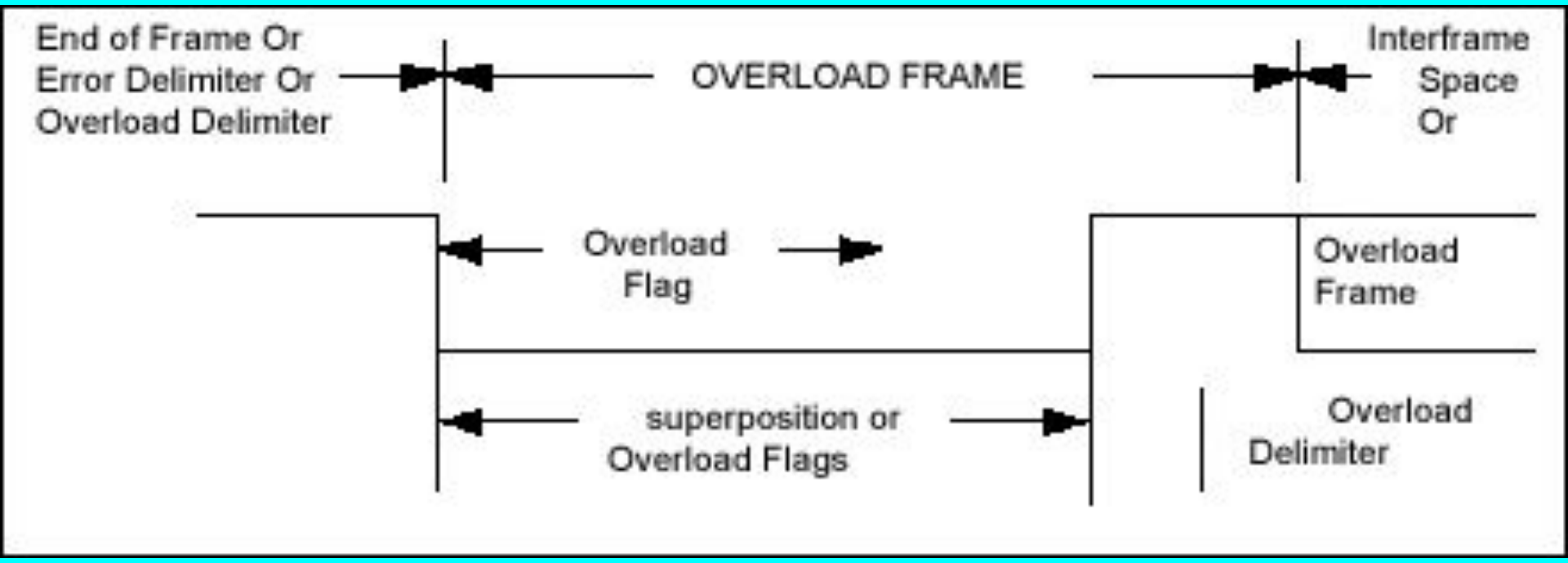
An Õerror passiveÕstation detecting an error condition tries to signal this by transmission of a PASSIVE ERROR FLAG. The Õerror passiveÕstation waits for six consecutive bits of equal polarity, beginning at the start of the PASSIVE ERROR FLAG. The PASSIVE ERROR FLAG is complete when these 6 equal bits have been detected.

ERROR DELIMITER

The ERROR DELIMITER consists of eight ÕrecessiveÕbits.

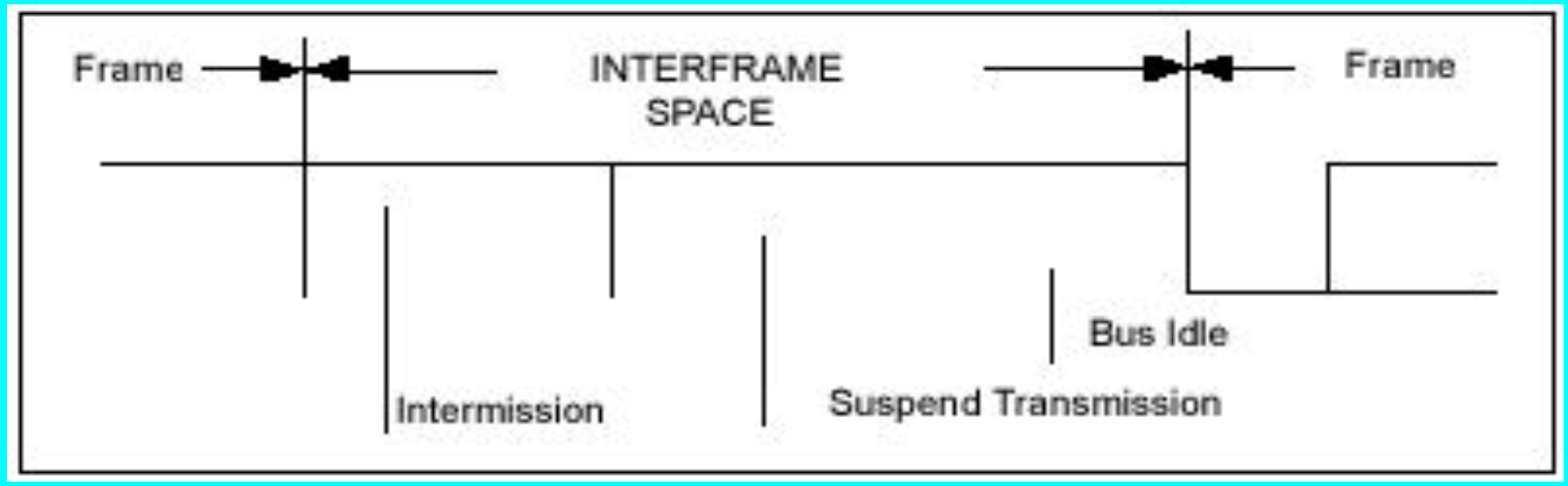
After transmission of an ERROR FLAG each station sends ÕrecessiveÕbits and monitors the bus until it detects a ÕrecessiveÕbit. Afterwards it starts transmitting seven more ÕrecessiveÕbits.

CAN – фрейм перегрузки



Имеются два вида перегрузки, которые оба приводят к передаче кадра перегрузки.
1. Внутреннее состояние приёмника, которое требует задержки следующего кадра данных или кадра удаленного запроса данных.
2. Обнаружение бита с лог. "0" в течение поля перерыва (см. межкадровое пространство).

CAN – межфреймовый промежуток



Поле перерыва (Intermission)

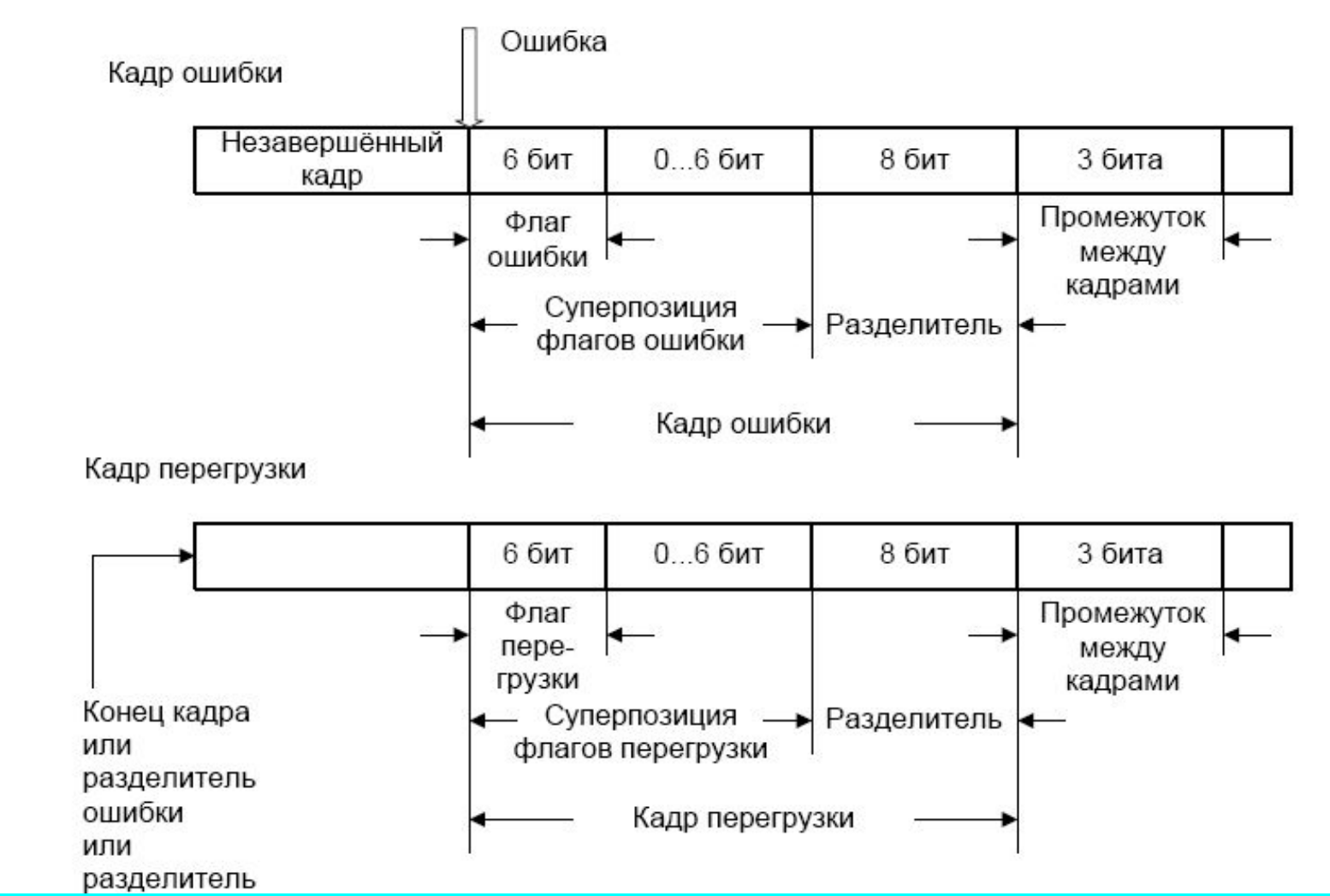
Состоит из 3 бит с лог. "1". В течение перерыва никакому узлу нельзя начинать передачу кадра данных или кадра удаленного запроса данных. Единственно возможное действие - это сигнализация состояния перегрузки.

Простой шины (Bus Idle)

Простой шины может иметь произвольную длину. Если шина опознана как свободная, любой узел, который имеет что - либо для передачи может начать передачу. Сообщение, которое было задержано для передачи другого сообщения, начинает передаваться в первом бите после поля перерыва.

Обнаружение бита с лог. "0" на шине в течение этого поля интерпретируется как поле "начало кадра".

Кадры ошибок и перегрузок



CAN – особенности спящего режима

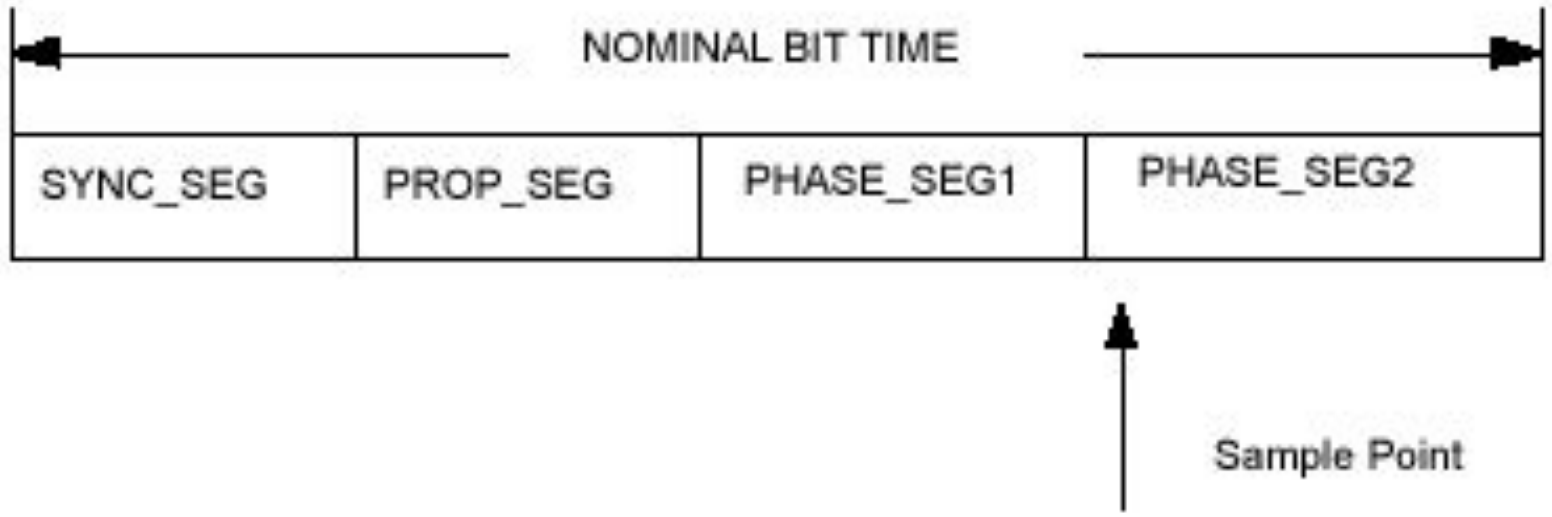
Чтобы уменьшить потребляемую мощность системы, узел CAN может быть переведен в режим "сна". Режим "сна" заканчивается при любом действии на шине или внутреннем состоянии системы. При пробуждении запускается внутренняя синхронизация, канальный уровень ждёт стабилизации генератора системы, а затем будет ожидать самосинхронизации к действиям на шине (синхронизация к действиям на шине заканчивается после принятия последовательности 11 битов с лог. "1"). Для пробуждения узла из режима покоя может использоваться некоторое сообщение пробуждения со специальным идентификатором.

Кодирование битового потока

Следующие поля: "начало кадра", "поле арбитража", "поле контроля", "поле данных" и "поле CRC" кодированы методом разрядного заполнения. Всякий раз, когда передатчик передает пять последовательных бит идентичной величины в битовом потоке, он автоматически вставляет дополняющий бит противоположного значения в фактически передаваемый битовый поток.

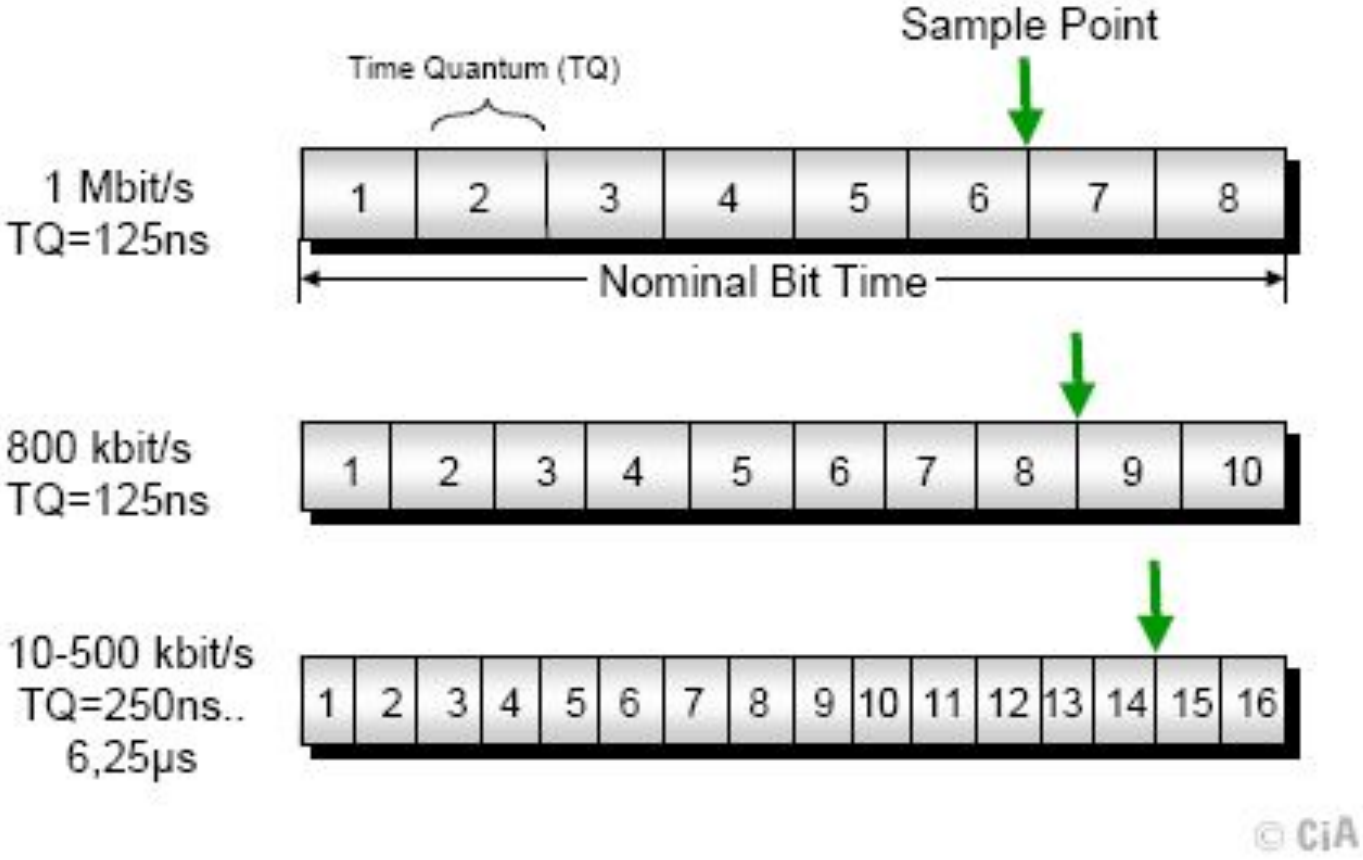
Оставшиеся битовые поля кадра данных или кадра удаленного запроса данных ("разделитель CRC", "поле подтверждения" и "конец кадра") имеют фиксированную форму и не кодируются. Кадр ошибки и кадр перегрузки также имеют фиксированную длину и не кодируются методом разрядного заполнения.

Требования к синхронизации



- **SYNC SEG** Эта часть времени передачи бита используется, чтобы синхронизировать различные узлы на шине. Ожидается, что фронт сигнала находится внутри этого сегмента.
- **PROP SEG** Эта часть времени передачи бита используется, чтобы компенсировать физическую задержку времён внутри сети.
- **TSEG1 и TSEG2.** Эти сегменты используются, чтобы компенсировать ошибки смещения фазы сигнала. Эти сегменты могут быть удлинены или укорочены пересинхронизацией.
- **Точка считывания (Sample point)** Точка считывания - точка времени, в которой уровень шины читается и интерпретируется, как величина соответствующего бита. Ее место - в конце TSEG1.

Bit-Timing Specification



At a bitrate of 1 Mbit/s the bit consists out of 8 time quanta, at 800 kbit/s out of 10 time quanta and from 500 kbit/s to 10 kbit/s out of 16 time quanta. CANopen uses single sampling mode only.

Bit rate Bus length [1]	Nominal bit time t_b	Number of time quanta per bit	Length of time quantum t_q	Location of sample point	Index
1 Mbit/s 25 m	1 µs	8	125 ns	6 t_q (750 ns)	0
800 kbit/s 50 m	1.25 µs	10	125 ns	8 t_q (1 µs)	1
500 kbit/s 100 m	2 µs	16	125 ns	14 t_q (1.75 µs)	2
250 kbit/s 250 m [2]	4 µs	16	250 ns	14 t_q (3.5 µs)	3
125 kbit/s 500 m [2]	8 µs	16	500 ns	14 t_q (7 µs)	4
50 kbit/s 1000 m [3]	20 µs	16	1.25 µs	14 t_q (17.5 µs)	5
20 kbit/s 2500 m [3]	50 µs	16	3.125 µs	14 t_q (43.75 µs)	6
10 kbit/s 5000 m [3]	100 µs	16	6.25 µs	14 t_q (87.5 µs)	7

Эффективность обнаружения ошибок

Механизмы обнаружения ошибок имеют следующие возможности:

- обнаружение всех глобальных ошибок
- обнаружение всех локальных ошибок передатчиков
- обнаружение до 5 случайно распределённых ошибок в сообщении
- обнаружение последовательной группы ошибок длиной до 15
- обнаружение любого числа нечетных ошибок в сообщении

Общая остаточная вероятность ошибки для необнаруженных, разрушенных сообщений, меньше чем:

4.7*10E-11