eBUS Specifications

Physical Layer – OSI 1
Data-Link Layer – OSI 2

V.1.3.1

03/2007

1 Table of Content

2 Introduction

The eBUS is a serial, LOW-active two-wire bus with multi-master properties that is usually operated in asynchronous mode. For certain applications, it is advantageous to utilise an synchronous access procedure. The synchronous access procedure is described as a supplement to this specification. It may be activated in context with management functions as described in layer 7 from the asynchronous access procedure under exclusion of a mixed process.

The eBUS serves mutual communications between heating components that may participate in the communication as master and also as slaves.

The eBUS is characterised by following key features:

- Pure two wire bus, supply and communications is accomplished over two wires
- Fixed Baud rate 2400 Baud +/- 1,2% Tolerance
- LOW-active bus with following levels:

Core Level	Transmitter	Receiver
HIGH = 1	> 15V DC	> 15V DC... maximal 24VDC
LOW = 0	≥ 8V DC and ≤ 10V DC, typically 9V DC	≥ 8 VDC and ≤ 12 V DC

With reference to the transmitting medium, the following limitations apply for transmitters of higher transmission levels – up to max. 12V DC:

- Multi-master capability: max. 25 masters and 228 slaves
- Communication Master ↔ Master, Master ↔ Slave and broadcast is possible
- Byte-oriented protocol with byte-oriented arbitration
- Realisation with standard-UART (8bit + start- and stop bit)
- Data protection through 8-bit CRC (generating polynomial: $x^8 + x^7 + x^4 + x^3 + x + 1$)
- Various bus supply options
- Supply of participants or bus couplers respectively by the bus (depending on bus supply)
- RAM memory requirements: 10 bytes for bus management + utility data buffer
- Variable utility data from 0 bytes to max 16 bytes
- 254 primary commands and 254 secondary commands (64516 different commands) are possible

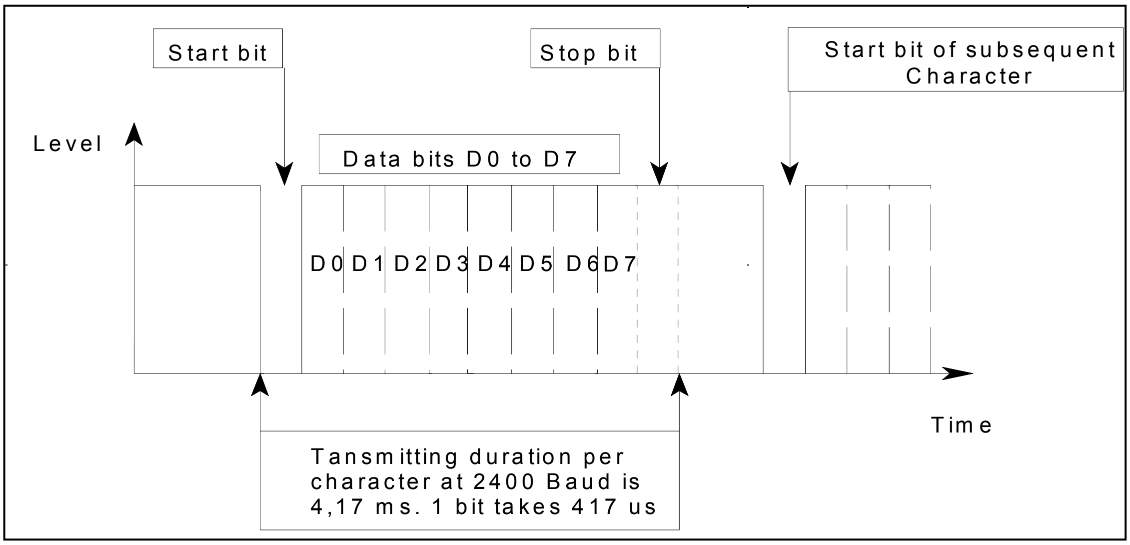
On following pages you will find a more detailed description of the eBUS. Listed is all necessary information, such as description of design of bus telegrams, arbitration, bus access etc. for the development of a network participant.

3 The Data Format

For serial data transmission, the 8-bit UART-mode is used.

This entails that the 8-bit word is furnished with a start bit at the beginning and a stop bit at the end. In case of serial transmission, the individual bits are sent in succession over the bus. Transmission always begins with the start-bit (LOW), then follow the eight data bits (bit0 to bit7) and, finally, the stop-bit (HIGH) is sent.

The following illustration depicts the 8-bit-UART mode.

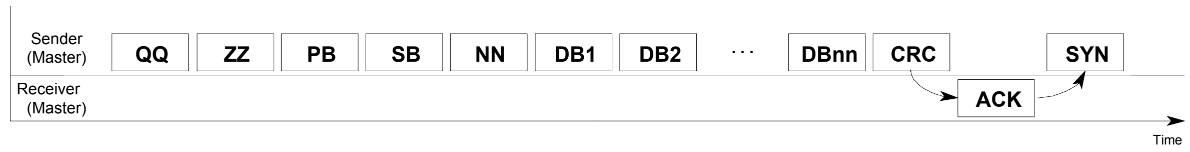


Architecture of a serial Character in 8-Bit-UART Mode

4 Telegrams

A telegram is the sequence of individual characters that a participant must use, when accessing the bus. In order for all participants to communicate with each other, the structure of these telegrams must be standardised.

4.1 The Master -Master Telegram



QQ	= Source address
ZZ	= Destination address (Master address)
PB	= Primary command
SB	= Secondary command
NN	= No. of following data bytes (NN=nn)
DB1	= 1st data byte
DB2	= 2nd data byte
	.....
Dbnn	= last data byte
CRC	= 8-Bit CRC as of symbol QQ
ACK	= Acknowledgement byte
SYN	= End of message synchronisation symbol

In this telegram, communication starts with the sender address (Master). Reason: sender address is used in arbitration procedure.

For the purpose of acknowledgement of a message, an acknowledgement byte has been appended. The acknowledgement byte is not transmitted by the sender, but rather from the addressed receiver. In other words, during transmission of the acknowledgement byte, the transmitter is in receiving mode.
The SYN-symbol at the end of a telegram causes enabling bus access for the next participant. This permits a high degree of bus utilisation.

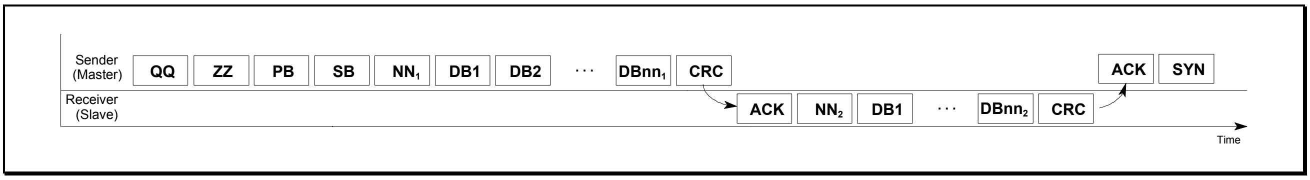
4.1.1 Master

A master may start communication with other bus participants. It has to meet certain criteria, i.e. must carry out a bus-arbitration in order to control bus allocation in case of simultaneous bus access action by several masters.

Furthermore, a master possesses a slave address. The slave address of a master results from the addition of a 5 to the master address. (Example: Slave-address of master 01h = 06h). Provided it supports them, it may be accessed also in slave mode.

SPECIAL CASE: The master with the address FFh recieves by addition of 5 to its master–address the slave-address 04h.

4.2 The Master-Slave Telegram

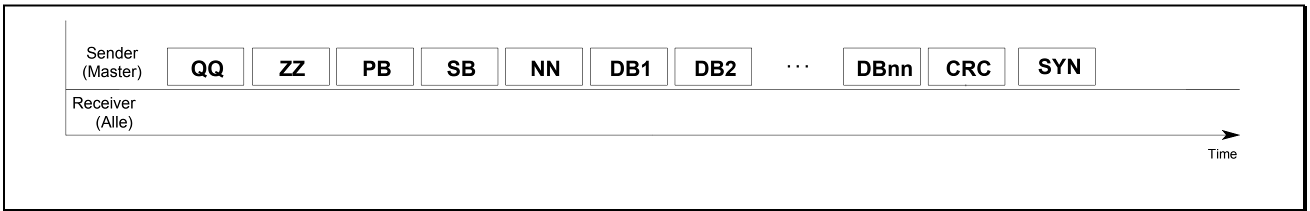


QQ	= Source address
ZZ	= Destination address (slave address)
PB	= Primary command
SB	= Secondary command
NN_1	= No. of following data bytes ($NN_1 = nn_1$)
DB1	= 1st data byte of master QQ
DB2	= 2^{nd} data byte of master QQ
.....	
$Dbnn_1$	= last data byte of master QQ
CRC	= 8-bit CRC as of symbol QQ
ACK	= Acknowledgment byte
NN_2	= No. of following data bytes ($NN_2 = nn_2$)
DB1	= 1^{st} data byte of slave ZZ
DB2	= 2^{nd} data byte of slave ZZ
	.....
$Dbnn_2$	= last data byte of slave ZZ
CRC	= 8-bit CRC as of symbol NN_2
ACK	= Acknowledgement byte

4.2.1 Slave

A slave cannot access the bus on its own; rather it has to be summoned by a master to respond. This is why its telegram structure is not identical to that of a master. In the event that a master addresses a slave, the bus is kept open by the master following successful acknowledgement. This enables the slave to send a message to the master. Since a slave only sends data, it has no need for address, command or SYN-symbol.

4.3 The Broadcast Telegram



QQ	= Source address
ZZ	= Destination address (broadcast address)
PB	= Primary command
SB	= Secondary command
NN	= No. of following data bytes
DB1	= 1^{st} data byte
DB2	= 2^{nd} data byte
	.....
Dbnn	= last data byte
CRC	= 8-bit CRC as of symbol QQ
ACK	= Acknowledgement byte
SYN	= End of message synchronisation symbol

A broadcast telegram is addressed to all participants and may not be acknowledged.

5 Telegram Components

5.1 The Synchronisation Symbol [SYN]

By means of the SYN-symbol, all transmission-authorised participants are advised to access the bus. The SYN-symbol has bit sequence 10101010 (AAH). This bit sequence is reserved for the SYN symbol and may not occur in any other symbol or character. Should this happen, e.g. in a data byte, value AAH has to be converted into two data bytes and expanded to sequence A9H + 01H. If only one data byte has the value A9H, again two data bytes need to be generated, i.e. A9h is expanded to A9h + 00h. When generating data, care must be given to keep data words A9H and AAH as rarely as possible from occurring. Otherwise, the data utilisation rate diminishes.

5.2 The Source Address [QQ]

The source address is the address of the master that is issuing a telegram. At the same time, it serves for arbitration of a bus conflict. A maximum of 25 source addresses are possible.

5.3 The Destination Address [ZZ]

The destination address is the address of the receiver. A maximum of 254 destination addresses are possible.

5.4 The primary Command [PB]

As of the primary command, the telegram becomes participant-specific. The primary command classifies the subsequent message. The primary command must be known by the receiver; otherwise he would not know what to do with the completely received telegram. Primary commands may be freely defined among bus participants. Likewise, general instructions can be defined which have a common meaning for all bus participants. A maximum of 254 primary commands are possible (AAh and A9h excepted).

5.5 The secondary Command [SB]

The secondary command further codes the primary command, making a total of 254 x 254 commands possible (AAh and A9h excepted).

5.6 Number of Data Bytes [NN]

The number of data bytes NN indicates how many data bytes (useful bytes) will yet follow. The following applies:

$$0 \leq NN \leq 16.$$

HINT: The Substitution of an useful byte A9 and AA respectively does not change the number [NN] of a telegram's useful bytes.

5.7 The 8-Bit CRC [CRC]

The 8-bit CRC serves to check a transmitted message for its correctness. When a sender transmits a telegram to another participant, the receiver calculates a CRC over the sequence of all received characters, i.e. over the expanded transmission sequence. At the end of a telegram, you may find the potentially expanded CRC-byte of the sender. If the transmitted CRC-byte matches the CRC-byte of the receiver, the message is regarded as being received correctly.

5.8 The Acknowledgement Byte [ACK]

The acknowledgement byte informs the sender, if the message has been correctly received. A positive ACK (reception correct) is defined as 00H. A negative ACK (reception incorrect) is defined as FFH.

6 The Bus Access

Normally, the eBUS is operated in asynchronous fashion, meaning that all devices have to have the ability of asynchronous bus access. For certain applications it is possible to apply an synchronous access procedure. This synchronous access procedure is described as a supplement of this specification. Under exclusion of a mixed operation, it may be activated from the asynchronous access procedure in the context of the management functions as described in OSI layer 7.

6.1 Function of the SYN Symbol

The SYN-symbol found at the end of a telegram plays a significant role in the context of bus access. The SYN-symbol tells other participants that the bus is free. If a participant has received such a SYN-symbol, it must take access to the bus immediately if it intends to send a telegram. In case several bus participants reach for bus access simultaneously, arbitration will decide which participant may gain access to the bus.

When the eBUS system is turned on, no participant will attempt to access the bus, because all participants are waiting for the SYN-symbol to appear. This first SYN-symbol is sent onto the bus by the SYN-symbol generation. Additionally, the SYN-symbol generator monitors the bus. Should no edge be registered within the SYN-repeat time, the SYN-symbol generator will send another SYN-symbol (AUTO-SYN).

The reason for this AUTO-SYN-symbol shall be explained by means of the following example:

6.1.1 Example for the Meaning of the automatic SYN-Symbol Generation

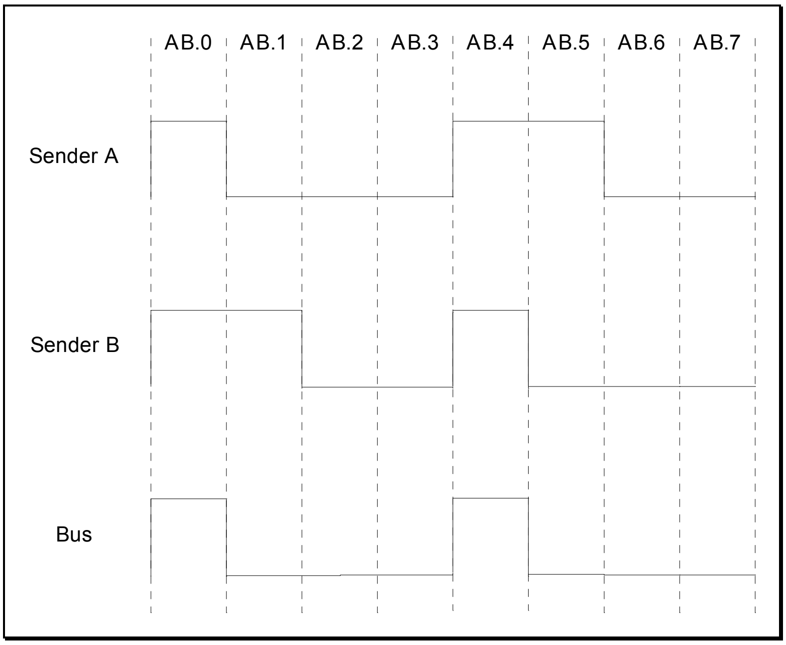
Participant 1 has sent its telegram onto the bus. By means of the SYN-symbol at the end of his telegram, the other participants are advised that the bus is now free. Should other transmission-authorised participants wish to access the bus, this access has to take place immediately following the SYN-symbol. If a transmission-authorised participant has no data in its send-buffer, it will not access the bus. Should data be loaded into the send-buffer shortly after the SYN-symbol, these can only be sent, after a SYN-symbol is received again.
Should this case occur with all participants, none would be able to send any data, because no participant will receive another SYN-symbol. At this point, the bus supply intervenes. After expiration of SYN-symbol repeat time, it sends the AUTO-SYN symbol. Only then, all authorised participants may access the bus again.

6.2 The Arbitration

If several participants access to the bus following the SYN-symbol, the arbitration procedure must prevent a bus conflict. For arbitration, the address of a participant serves as decision criterion. Allocation is made to that participant that has the lower value address (higher priority). As with the CAN- or I^2C bus, that participant will stay on the bus that finds its address again in its receiving buffer of the serial interface. The procedure that is used with the CAN- or I^2C bus cannot be applied here. This is due to the fact that it is bit-oriented. This means that each bit that is sent to the bus by a participant is being checked for correctness. With the interface of the UART, only each byte can be checked.
The problem that is associated with a byte-oriented procedure is described below.

6.2.1 Example of Bus Collision

The following diagram shows what happens, if sender A and B operate in a byte-oriented mode.



Both transmitters send address bits AB.0 to AB.7, one after another onto the bus. In their respective receiving buffers is found the word that was generated on the bus. In both cases, none of the senders recognises itself anymore. Both would have to withdraw from the bus and start a new attempt following the next SYN-symbol. Through this repeated access, the bus may become paralysed completely, because both participants will attempt gaining access to the bus following each SYN-symbol. A useful solution of the address collision is achieved by byte-oriented arbitration and the assignment of suitable participant addresses.

6.2.2 The Byte-oriented Arbitration Procedure

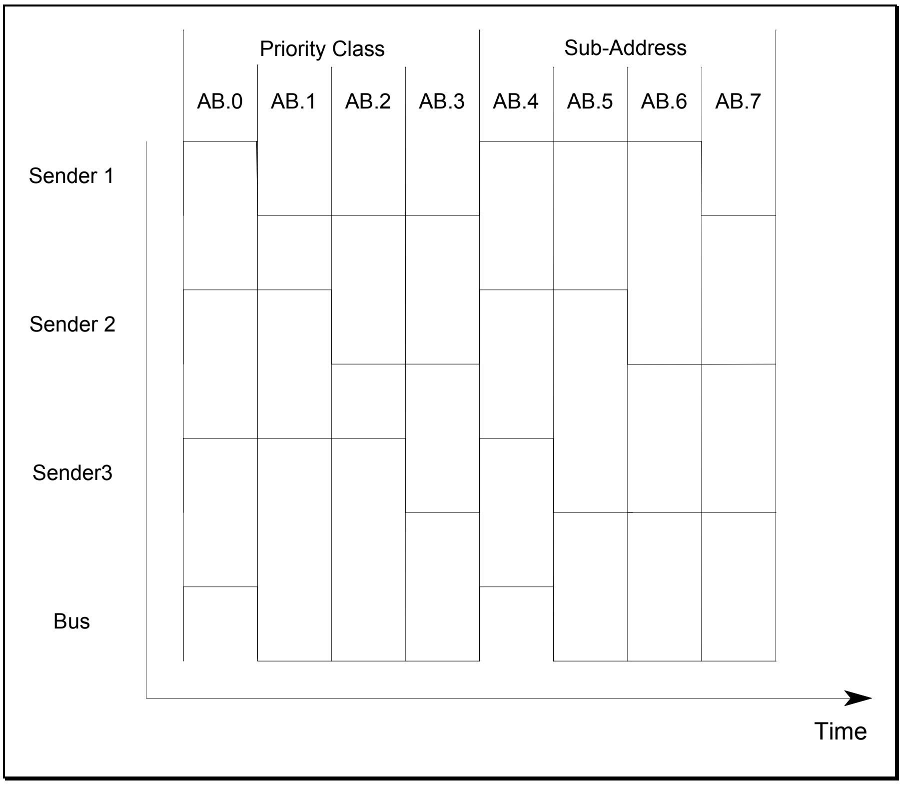
In a byte-oriented arbitration procedure the assignment of addresses is of utmost importance.

6.2.2.1 Priority Class and Sub Address

The address (the 8-bit word) is split in two 4-bit words. The address consists of a priority class (bit.0 to bit.3) and of a sub-address (bit.4 to bit.7). By this measure, it is possible to assign 25 master addresses.

	Address-Byte Sub-Address	Priority Class
Participant 1	0000	0000
Participant 2	0001	0000
Participant 3	0011	0000
Participant 4	0111	0000
Participant 5	1111	0000
Participant 6	0000	0001
Participant 7	0001	0001
.		
.		
.		
Participant 24	0111	1111
Participant 25	1111	1111

Also with these addresses, it may occur that none of the participants recognises itself, if several participants attempt to access the bus simultaneously. This is illustrated in the following diagram.



6.2.2.2 Collision Remedy

If several sending-authorised participants attempt to access the bus simultaneously and fail to recognise themselves, these participants withdraw from the bus. Subsequently, each of these participants checks if its address bit.0 to address bit.3 (priority class) matches with the received word. If this is the case, another access attempt will be started following the next SYN-symbol (AUTO-SYN).

When the SYN-symbol is sent, all participants that have made no prior attempt to access the bus will recognise that only one word has been sent to the bus between two SYN-symbols. Through word sequence SYN-symbol / Address-symbol / AUTO-SYN-symbol, these participants will be barred and may only attempt to access the bus following the next SYN-symbol. The members that have recognised their priority class before that, will access to the bus following the AUTO-SYN symbol.

Since all participants that acces to the bus in this second round of arbitration have the same priority class, one of them will recognise itself and stay on the bus. The others that do not recognise their priority class will withdraw.

6.3 The immediate Bus Access following the SYN-Symbol

As already indicated, bus access of all participants has to occur following the SYN-symbol, in order for the arbitration to proceed successfully.
Each participant needs a certain amount of time to decide if it wishes to send onto the bus following reception of a SYN-symbol. This access time is caused by the program flow that is initiated when a SYN-symbol has been detected. The participants usually have slightly different times t_Z in order to access the bus. These access times are defined by a minimal access time t_{Zmin} and a maximal access time t_{Zmax} following the start edge of the SYN-symbol, see chapter 10.9.

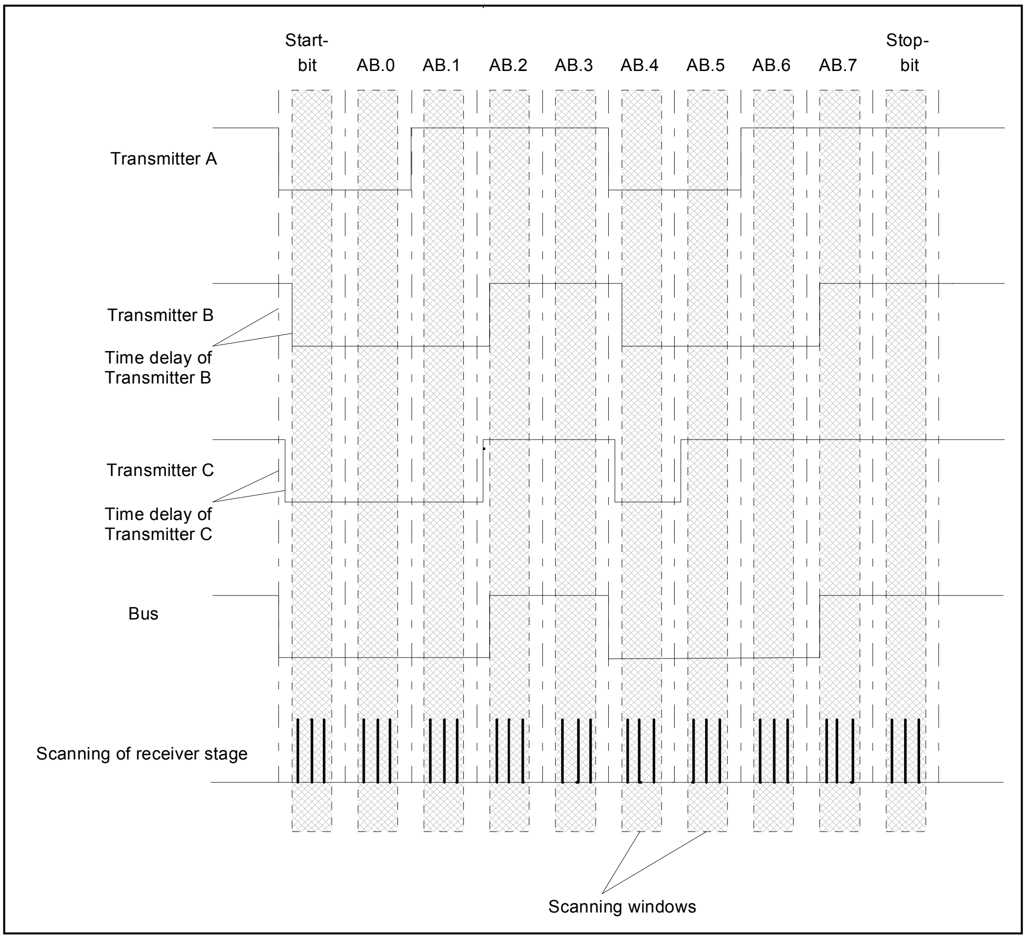
That is to say that the arbitration procedure allows for a maximal variation t_{Vmax} for these access times $t_{Zmax} - t_{Zmin}$. For two participants that are attempting bus access, that means that one of them may access the bus delayed by t_{Vmax} than the other. In case more than two participants attempt bus access following the SYN-symbol, the time interval between the first and the last participant may not exceed t_{Vmax}.

The maximal delay t_{Vmax} may be calculated as follows:

The receiver stage of the serial interface of the UART is divided in 16 segments by a bit. As one bit at 2400 Baud needs 416,67 µs, at 2400 Baud + 1,2%, i.e. only appr. 412µs and at 2400 – 1,2% however requires 421,72 µs, the duration needed per segment varies between 25,73 µs and 26,36 µs, depending on the internal clock speed:

$$t_{Segment,min} = t_{Bit,min} / 16 = 411,7\mu s / 16 = 25,73\mu s$$
$$t_{Segment,max} = t_{Bit,max} / 16 = 421,7\mu s / 16 = 26,36\mu s$$

Only in segments 7, 8 and 9 of each bit, is the level for scanning guaranteed, as is shown in the following illustration:

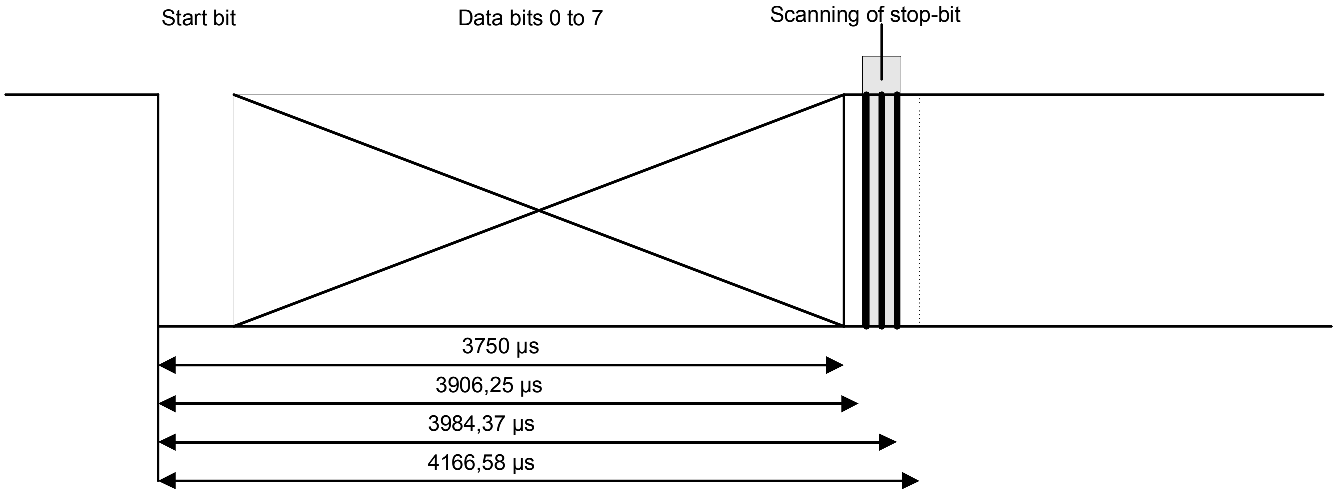


This design principle prevents initiating a reception sequence through interference. The receiving stage reacts to the negative edge of the start-bit. The lengths of the subsequent 10 bits (start bit + 8 data bits + stop bit) are determined by the internal clock of the device. The start edge could, however, also be caused by a negative interference impulse. Should this occur, the level following the interference impulse will again be on HIGH. Since a HIGH level will be registered on the bus during scanning, the receiving stage will cut off and wait for the next negative edge.

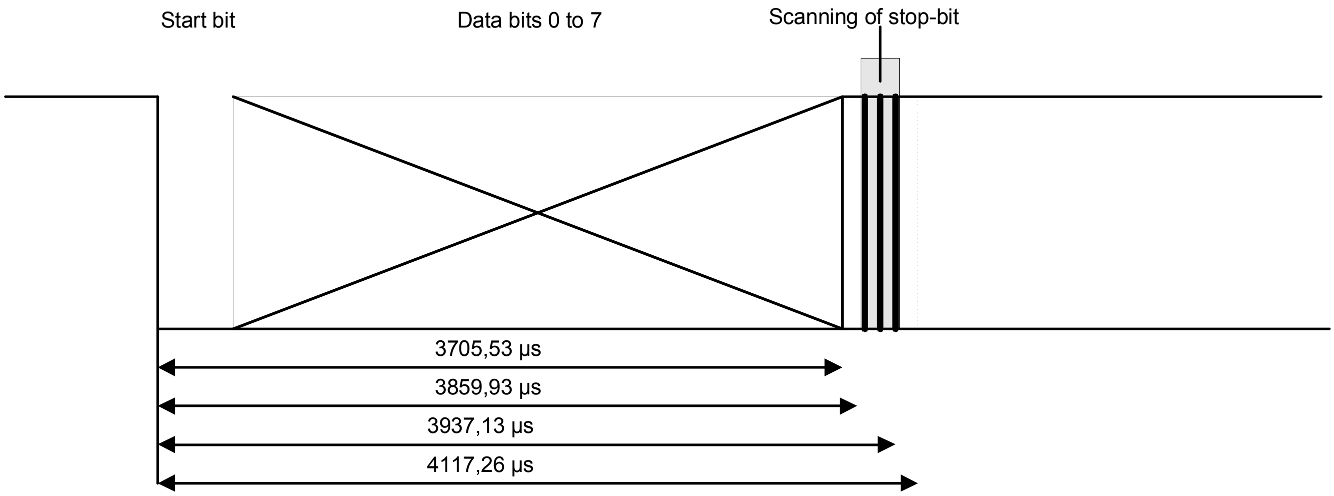
The negative edge is followed by 6 segments, during which the receiving stage does not scan. Then the start-bit is scanned, the receiving stage waits for the window of the 1st data bit and so on until the scan window of the stop-bit is reached. Therefore, scanning of the stop-bit occurs dependent on the device-internal clock at varying points in time.

In each case, the following illustration depicts marking of the first of the indicated points in time for the end of the 9th bit, the second time shows the begin of the scan window for the stop-bit, the third indicates the end of the scan window and the last indicates the total time to the end of the entire byte.

Time Values at 2400 Baud



Time Values at 2400 Baud + 1,2% =2428,8 Baud



If several transmitters are accessing the bus, one of them will be successful first. The maximal time delay of a transmitter depends on its own tolerance $-1{,}2 \leq x \leq 1{,}2$. In the worst case, the 1st transmitter sends immediately and with the positive tolerance of 1,2%. The time delay t_V of any device must not exceed

$$t_{Vmax} < (9+6/16)/(2400(1 + 0{,}012)) - 9/(2400\ (1 + x/100)) - T_{Edge\_LH}$$

The maximal edge rising time T_{Edge_LH} from last data bit to stop-bit is not negligible, compare chapter 10.3.1. In adjacent chart, some limit values are listed, that allow for a max. edge rising time of T_{Edge_LH} = 50 µs. The dependence of the maximal delay time of a device from its own tolerance x has a nearly linear progression within the tolerance range of $-1{,}2 \leq x \leq 1{,}2$.

Tolerance x%	t_{Vmax} <
1.2 %	104,40 µs
0.5 %	78,58 µs
0.0 %	59.93 µs
- 0.5 %	41.09 µs
- 1.2 %	14.38 µs

This maximal time delay guarantees that the stop-bits of all transmitters remain within the scan windows of all units.

6.4 The asynchronous Access Procedure

All bus users have to have command of the asynchronous access procedure.

Each master keeps a lock counter, which at first is initialised to value 0. Each master is transmission-authorised, if its lock counter contains the value 0. Following successful completion of an own bus access, each master sets its lock counter to its max.-value.

As long as a lock counter shows a value > 0, it will be decremented by value 1 with each on the bus detected SYN-symbol. Exception: if the detected SYN-symbol follows an arbitration that does not yield a clear winner.

6.4.1 Network Management

It is recommended to implement a network management according to the *eBUS Network Management Specifications* (available through the eBUS User Club). Implementation is, however, optional. Furthermore, the following definitions apply.
The organisation of the participants in normal operation and in exceptional situations is discussed in the following sub-chapters.

6.4.1.1 Initialisation

There are no directives: During net start-up, each node becomes bus-active on its own and does so independently from the other participants.

6.4.1.1.1 Lock Counter

Each master sets its lock_counter to maximum value.
The maximum value of the lock_counter has to be in the following quantity:

$\in$ {3,4, .,25}.

Through services of OSI-Layer 7, the maximum value of the respective lock counter may be altered.

Exceptions:

- The master with address FFH does not require a lock counter, respectively may set its lock counter max. value to zero.
- For diagnostic purposes, it is admissible to have lock counter_max $\in$ {0, 1, ..., 25}.

Hint: Experience has shown that every master should set its lock_counter_max value to the sum of masters participating the bus.

6.4.1.2 Normal Operation

There are no directives. Each node is bus-active independently of the others. The over all status of the net, i.e. which node exists within the net with what abilities, is not specified.
Users may interrogate the status, e.g. through commands of layer OSI 7.

6.4.1.3 Exceptions

6.4.1.3.1 Loss of a Node

No specific services are required, compare chapters 6.4.1.1 Initialising and 6.4.1.2 Normal Operation.

6.4.1.3.2 Adding a Node during Operation

No specific services are required, compare chapters 6.4.1.1 Initialising and 6.4.1.2 Normal Operation.

7 Message Transmission

7.1 Issuing a Master Telegram

The surviving master of an arbitration procedure sends first the destination address and then the remainder of the message. The last symbol is the CRC check sum. Should the message not be a broadcast message, the sender expects an acknowledgement from the receiver, (positive or negative acknowledgement).

7.2 CRC Test Sum

The CRC test sum is generated by the respective sender over the expanded byte transmission sequence, using the generator polynomial $X^8 + X^7 + X^4 + X^3 + X + 1$ and transmitted as the last byte of the message. If required, it may even be sent in expanded form.

7.3 Acknowledgement

If the receiver has received the message correctly, i.e. received CRC and calculated CRC match, it will send a positive acknowledgement (00H). Should, however, the received CRC not match the calculated CRC, the receiver will send a negative acknowledgement (FFH). Broadcast messages are not acknowledged.

7.4 Repetition of a Telegram in Case of failed Transmission

In case a master or slave receives a negative acknowledgement, they will repeat their telegram.
The telegram repetition must be generated before the AUTO-SYN facility initiates an AUTO-SYN symbol. To prevent blocking the bus by constant telegram repetitions, the repeat rate is set to the value 1, i.e. repetition is limited to one. In case a repeated message again does not get a positive acknowledgement, the respective telegram is rated as not sent.

7.5 Reception of the Reply

In case of master-master communication, there is no direct reply. Rather, a new master telegram follows as answer but does not have to immediately follow the next SYN-symbol. It is feasible to have one or more messages travel over the bus in between.

In case of a master-slave communication, the reply from the slave must follow the acknowledgement of the inquiry by the master within the AUTO-SYN time slot.

By way of the reply, the master again forms a CRC checksum that is compared with the received CRC from the slave. Repetition and acknowledgement procedures are as described above.

7.6 Release of the Bus

Following successful acknowledgement of all telegram segments, the master sends a SYN-symbol and thus releases the bus for the next arbitration. This eliminates the waiting time for the AUTO-SYN symbol, which is generated by the AUTO-SYN facility.

8 Data Utilisation Rates

8.1 Data Utilisation Rate as Master

It is calculated from the number of data bytes (the two bytes for primary and secondary command are added to the data bytes) and the number of total bytes in a telegram.

$$\text{Data Utilisation Rate} \ \frac{\text{No. of Datanbytes} + 2}{\text{No. of Bytes in Telegram}}$$

8.2 Examples:

NN = 10:

In a master-master telegram, 10 data bytes are transmitted (characters A9H and AAH do not occur). The total number of all bytes of such a telegram is thus 18.

$$\text{Data Utilis. Rate} = \frac{10 + 2}{18} = 0.6667 = 66{,}67\%$$

The simple rule applies, the larger the number of data bytes in a telegram, the larger is also the data utilisation rate of a telegram.

Data Utilisation Rate for Worst Case (NN=0):

Zero data bytes are transmitted. As in a master-master telegram only the primary and secondary command are contained, we find

$$\text{Data Utilis.Rate} = \frac{2}{8} = 0{,}25 = 25\%$$

Data Utilisation Rate in case that data bytes of bit sequence AAH or A9H are transmitted:

10 data bytes of bit sequence AAH or A9H shall be sent. This symbol must be expanded to A9H + 01H or A9H + 00H prior to transmission.
That yields 20 transmitted data bytes. However, they correspond to only NN=10 utilisation bytes.
The length of the telegram is thus increased to 28. The number of data bytes is 10 and results in:

$$\text{Data Utilis. Rate} = \frac{10 + 2}{28} = 0{,}4286 = 42{,}86\%$$

8.3 Data Utilisation Rate as Slave

Through the reduced number of bytes in a slave telegram, the data utilisation rate of the slave is higher than that of the master

$$\text{Data Utilis. Rate Slave} = \frac{\text{Anzahl Datenbytes}}{\text{Anzahl Datenbytes} + 3}$$

8.4 Example:

10 data bytes are transmitted:

$$\text{Data Utilis. Rate Slave} = \frac{10}{13} = 0{,}7692 = 76{,}92\%$$

9 The automatic SYN-Symbol Generation

The SYN-symbol at the end of a telegram informs all transmission-authorised bus participants that the bus is now free.

In order to enable bus access, a SYN-symbol must be sent!

9.1 The SYN-Symbol AUTO-SYN

The AUTO–SYN symbol is sent whenever the HIGH-level remains on the bus over the SYN repetition time incessantly. This may occur, if for example a participant ends his telegram with a SYN-symbol and none of the other participants is ready to transmit. In this case, without an automatically generated SYN-symbol, no further communications will occur on the bus. In order to enable participants to access the bus at a later date, 35 ms after uninterrupted HIGH level (exception: **SYN repeat time ≤ 35ms**), a SYN-symbol will be sent to the bus. In case none of the last registered symbols prior to begin of the HIGH level is a SYN-symbol, the SYN-symbol generator must wait at least 30 ms (exception**: SYN repeat time ≥ 30 ms**), before it may generate the next SYN-symbol. For SYN-symbol generation following a SYN-symbol or a character that follows a SYN-symbol, it is permissible to occur as early as after 5 ms (**SYN repeat time ≥ 5 ms**).

9.2 SYN-Symbol Generators

At least one SYN-symbol generator must guarantee the generation of automatic SYN-Symbols for the bus. SYN-symbol generators may be integrated in the bus supply or in masters. For a SYN-symbol generator, a new HIGH level phase begins with each rising edge. A HIGH level phase ends through a falling edge on the bus. As soon as a HIGH level phase ends, a SYN-symbol generator may not access the bus and must wait for a new HIGH level phase, before it may generate a SYN-symbol. In case of simultaneous attempts by several SYN-symbol generators, the time divergence may not exceed 50 µs.

The SYN-symbol generator may be realised with an astable circuit that is reset by the edges on the bus. Should the edges on the bus not occur , the astable circuit will generate an interrupt for the processor, upon which it will send a SYN-symbol to the bus.

9.2.1 Character-oriented SYN-Symbol Generators

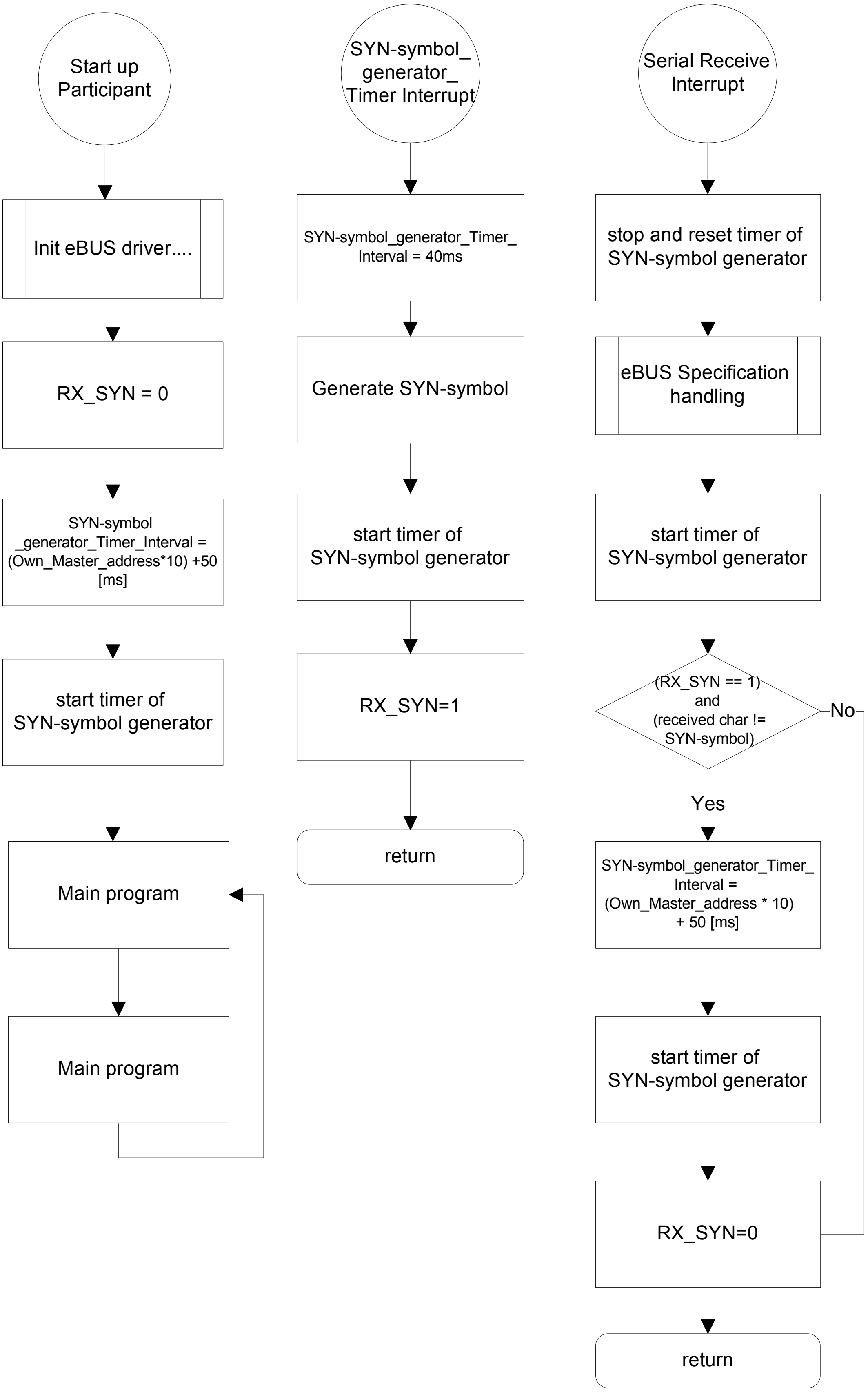
If the begin of the HIGH level phase within a byte cannot be certainly identified, **one** of the SYN-symbol generators on the bus may also be realised in character-oriented form.
A character-oriented SYN-symbol generator produces an AUTO-SYN character that is recognised on the bus no sooner than after 40 ms **(exception: SYN repeat time ≥ 40 ms)** and no later than 45 ms **(SYN repeat time ≤ 45 ms)** following the end of the last on the bus recognised symbol. The end of a character is the end of its stop-bit. The begin of a character is defined by the falling edge of the start-bit. Also in case of "min. one of the last recognised symbols on the bus is a SYN-symbol" a character-oriented SYN-symbol generator must wait at least 35 ms before generating a SYN-symbol!

9.2.2 More than one Character-Oriented SYN-Symbol Generator in a single eBUS System

After its startup as an participant on the bus has every Master an **unique** timer value (exceed the normal SYN repeat time). This **unique** timer value is used for the SYN-symbol generation. The timer is reset to its start value on every recognised symbol on the bus. If the timer of a participant has reached zero, this participant will generate a SYN-symbol, reset its timer value to the normal SYN repeat time (all other participants will keep their **unique** timer values).

Formula for the calculation of the **unique** timer value for each master:

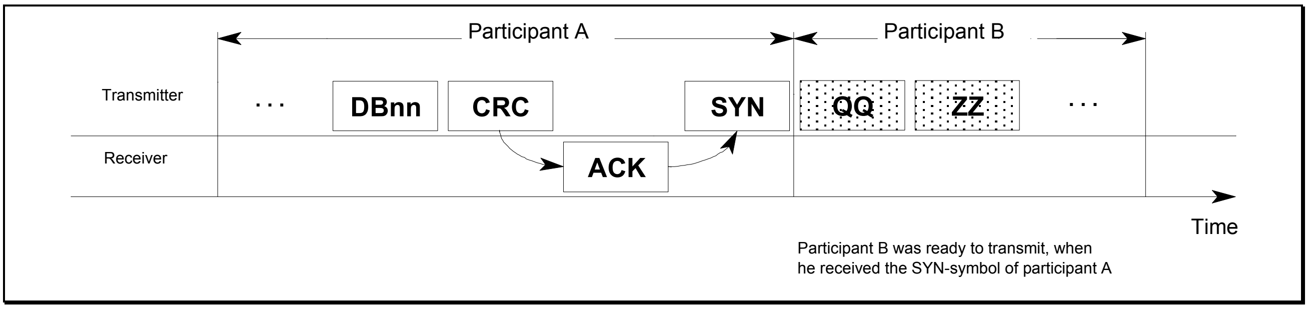
$$\mathbf{t_{unique} = (Master\ address * 10\ ms) + 50\ ms} \qquad \textit{[+ 5 ms tolerance]}$$



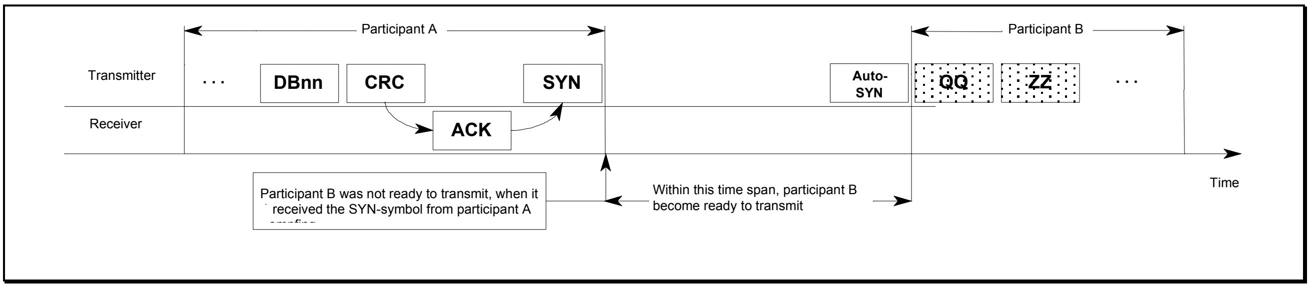
SYN-Symbol generation: Flow chart

9.3 Case Examples

9.3.1 Participant B was ready to transmit, when participant A sent its SYN-symbol:



9.3.2 Participant B was not ready to transmit, when participant A sent its SYN-symbol:



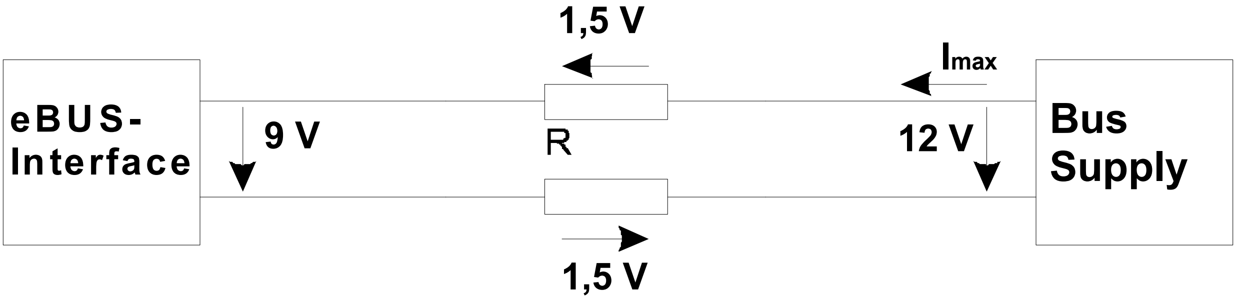
10 Electrical Realisation

10.1 Transmission Medium

The transmission medium is a twisted two-wire line of minimum 0.6 mm diameter.
Devices that transmit LOW levels with more then 10V, have a detrimental effect on the maximum length of the line. It applies, that the voltage drop over the entire length of the line shall not exceed 0.25 V. For the to and from length, this means that the voltage drop may not exceed 0.5 V.

10.1.1 Example for the Calculation of Line Length for the typical Transmission Voltage of 9V

With reference to supply current and line cross section, the following interdependence applies:



R_{line}=(U_{Lowmax} - U_{Lowmin}) / (I_{Supply} + I_{Excess})

R_{line}= 3V / I_{max}

l_{Line}= 0,5 * R_{Line} * A * χ χ = 56 m/(mm^2 * Ω) (Copper)

10.1.2 Examples for I_{max}= 100 mA and various Line Cross Sections:

A_1=0,28mm^2
l_{Line_A1}= 0,5 * R_{Line} * A * χ = 0,5 * 30Ω * 0,28mm^2 * 56 m/(mm^2 * Ω) = 235,2m

A_2=0,5mm^2
l_{Line_A2}= 0,5 * R_{Line} * A * χ = 0,5 * 30Ω * 0,5mm^2 * 56 m/(mm^2 * Ω) = 420m

A_3=1,5mm^2
l_{Line_A3}= 0,5 * R_{Line} * A * χ = 0,5 * 30Ω * 1,5mm^2 * 56 m/(mm^2 * Ω) = 1260m

10.2 Socket Connectors for eBUS

For eBUS connection, a variety of socket connectors may be used.

10.2.1 Examples

10.2.1.1 Rast 5 Connector with Coding 02-K59

These socket connectors are pole coded and may be used for eBUS devices that have an integrated bus supply.

Pole code: Pin 1 = eBUS +
Pin 2 = eBUS -

10.2.1.2 Phoenix Peb Strip 2-Pole Type MSTBA 2,5/2-G-5.08

Peg strip 2-pole type MSTB 2,5/2-G-5.08
Peg strip Wieland type SL
Used as matching plug-in terminals, use

- Phoenix type MSTB 2-pole and
- Weidmüller type BLA/ SLA 2-pole

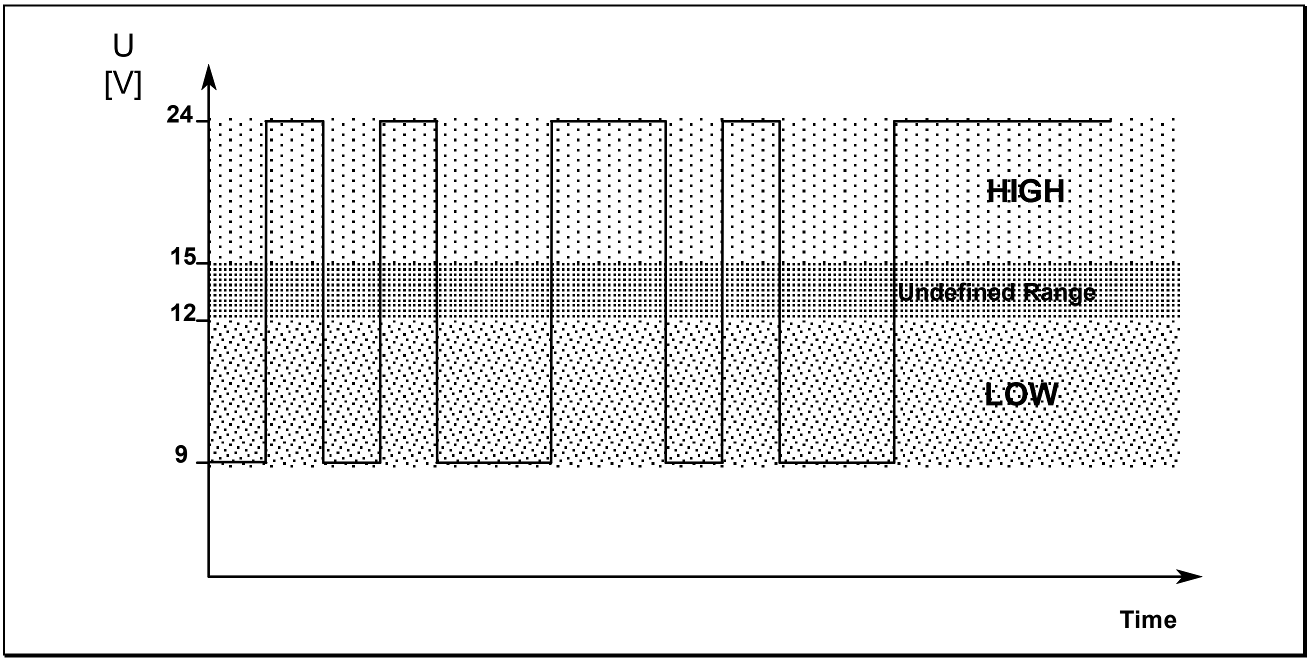
These socket connectors may be used for eBUS interfaces that are not polarised.

10.2.1.3 Western Modular 4-Pole

- As service socket on front of hardware units
- No polarisation
- The two pins on right and the two pins on left respectively are combined.

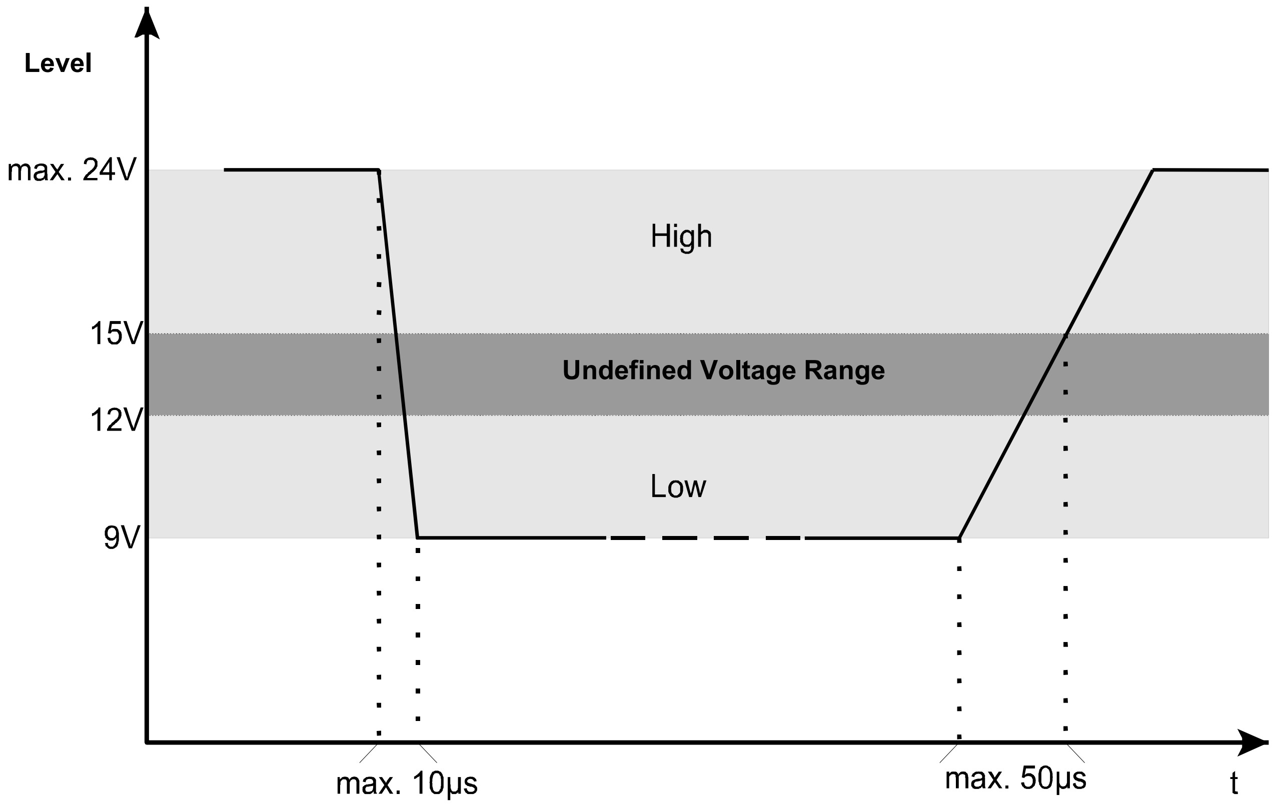
10.3 Bus Level

The bus level is defined in this way, to enable electric supply of individual bus participants over the bus. This is achieved through setting the LOW level 9V ≤ LOW ≤ 12V. In order to maintain a sufficient signal-to-noise-ratio, the following applies for the HIGH level: 15V < HIGH ≤ 24V, typically 20V.



10.3.1 Edge Slope on the eBUS

The edge slope on the eBUS depends on the amount of excess current. On the basis of current excess of 30 mA, the following diagram furnishes approximate indications.



Negative Edge t_{Edge_HL} of max. Pegel to 9V : max. 10µs, i.e. T_{Edge_HL} = 10 µs

Positive Edge t_{Edge_LH} of 9V to 15V : max. 50µs, i.e. T_{Edge_LH} = 50 µs

The negative edge chiefly depends on the degree of over-steer of the switching transistor in the eBUS Interface. The smaller the excess current on the eBUS, the faster is the negative edge.
The positive edge mainly depends on the capacitive load and the excess current on the eBUS.

At an excess current of 30mA and a slope rise time of max. 50µs (from 9V to 15V),
The following capacitive load ensues:

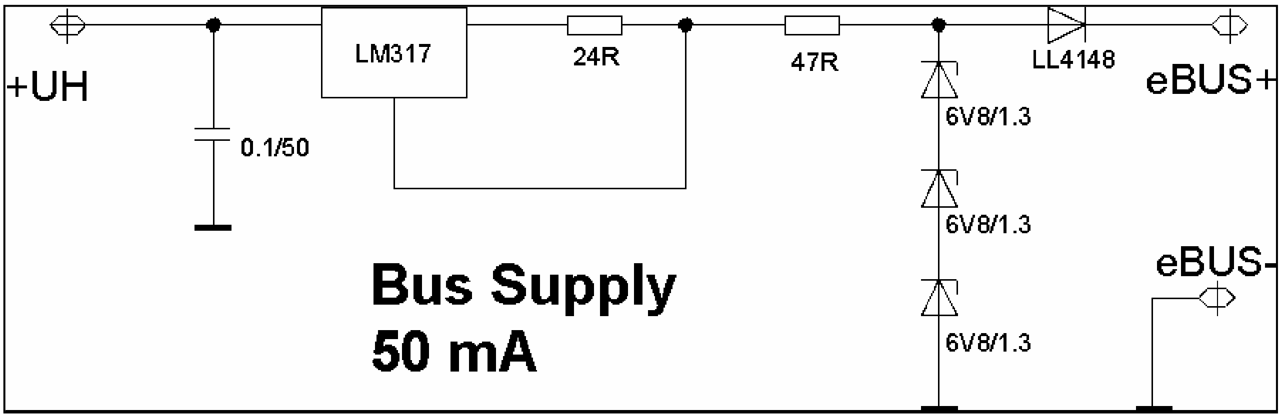
$C_{Load\ 30\ mA}$ = Excess Current * Rise Time / (High volt. – Low volt.) = 30 mA * 50µs / (15V - 9V) = 250 nF

Additional examples for 50 mA and 100 mA of excess current:

$C_{Load\ 50\ mA}$ = 50 mA * 50µs / (15V - 9V) = 416 nF

$C_{Load\ 100\ mA}$ = 100mA * 50µs / (15V - 9V) = 833 nF

10.4 Bus Supply



Examples for Switching of Bus Supply

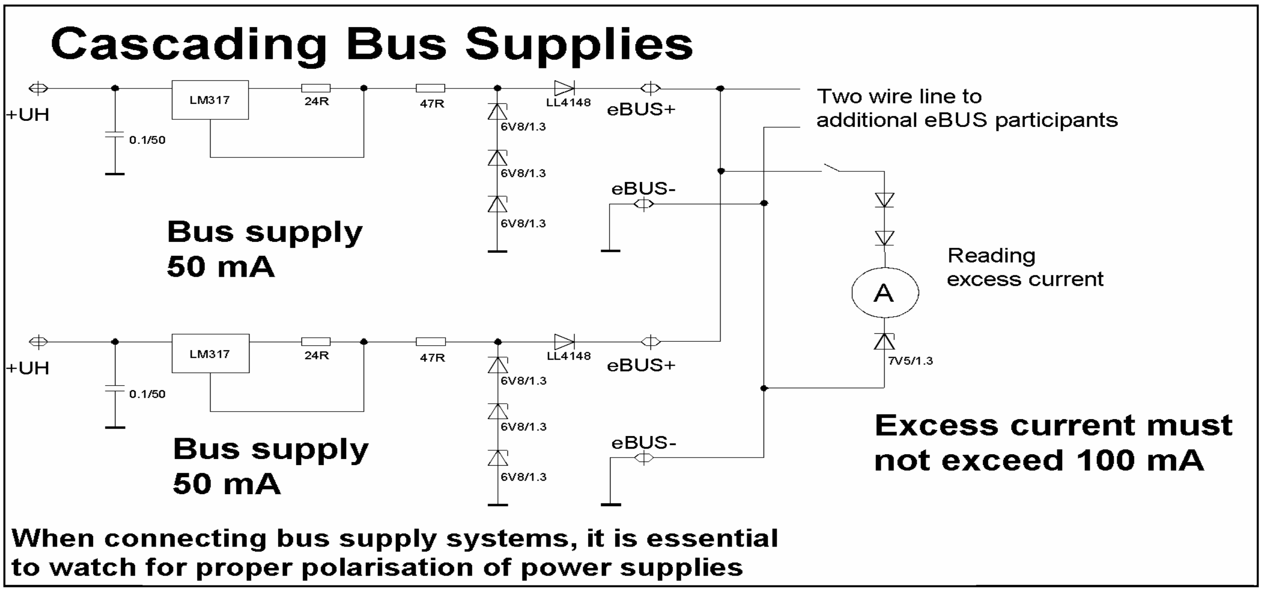
The bus supply represents a power source with a short circuit current level of 50 mA to maximal 100 mA. The source voltage is limited to 24V.
The bus connection must be electrically isolated from the mains voltage.
Participants with their own power supply must be electrically separated from the main supply, including ground wire (Protection Class II), or they must be separated from the bus by opto-coupler.
If desirable, several bus supply systems may be switched in parallel.
Rule of thumbs formula for dimensioning a bus supply:

Total Current = Stand-by Current + Excess Current

Stand-by current: Current draw of all connected participants at maximum voltage
Current excess: minimum 30 mA to **maximum 100 mA**.

10.5 Supply of individual Bus Participants over the eBUS

In order to achieve clean edges, bus participants must possess voltage stabilisers. The stabilised operating voltage of 5 Volts is achieved easily through a length stabilization with a transistor and a Z-Diode.
The length stabilization must be sized such as to obtain a minimal stand-by current. Of course, this depends on the maximum load current that is drawn by the subsequent circuit.
Using voltage stabiliser ICs is not suitable, as these have a tendency of causing oscillations on the bus.



Example of designing a cascaded bus supply

10.6 The eBUS Interface

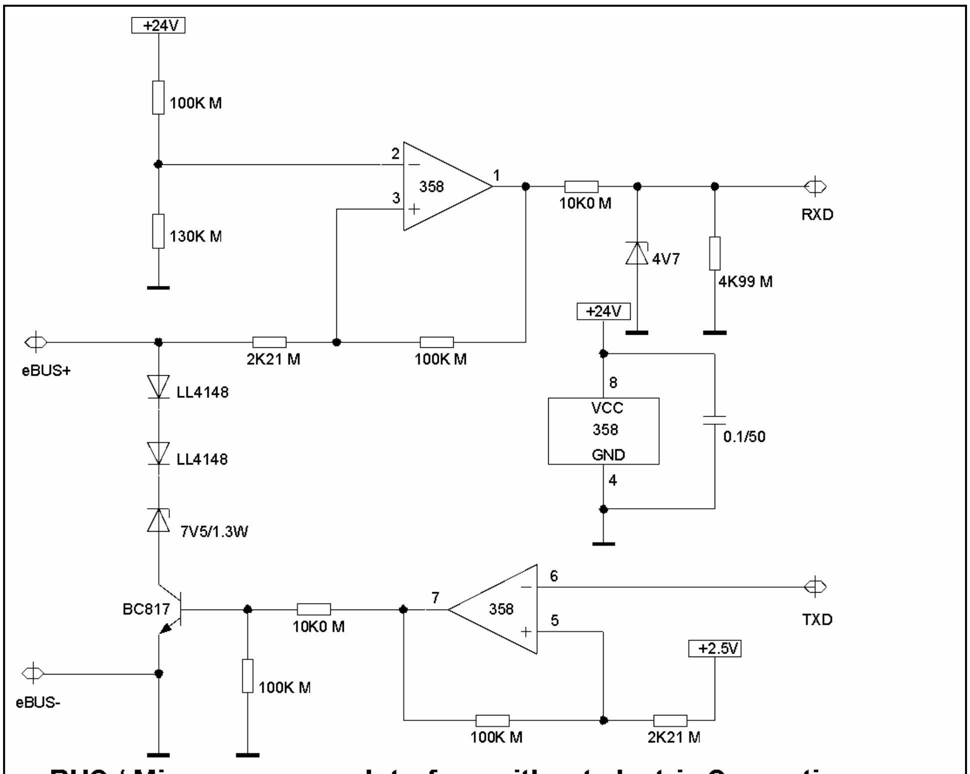
The eBUS interface represents the interface between the bus and a micro-processor/micro-controller.
In case of electric separation, two opto-couplers (for transmission and receiving) are required.
Participants that are supplied with energy over the bus have no need for any opto-coupler.

Receiving System of the Interface

The receiving system detects the level on the bus. In the range of 9V and 12V (LOW), the receiver opto-coupler is addressed (current flow through LED). In a range between 15 and max. 24V, the receiver opto-coupler is not addressed (no current flow through LED).

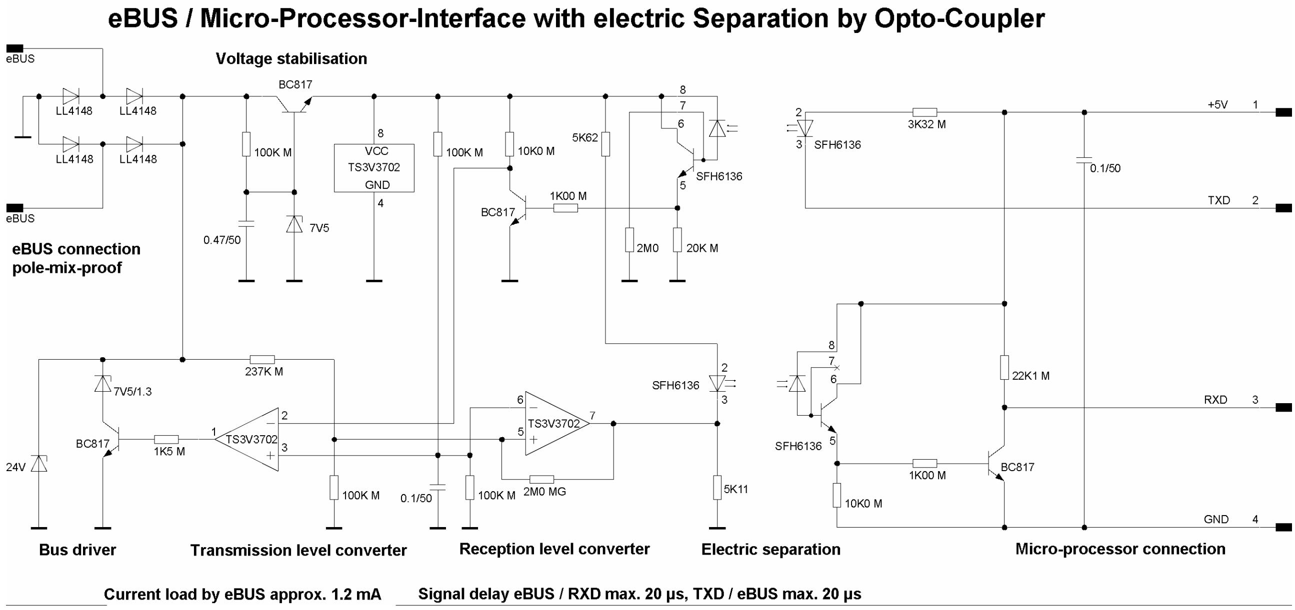
Transmitter System of the Interface

The connected micro-processor/micro-controller addresses the opto-coupler of the interface. By way of a comparator, a switching stage is addressed that will switch the bus level to approximately 9V.



eBUS / Micro-processor Interface without electric Separation

Electronic Circuit without electric Separation



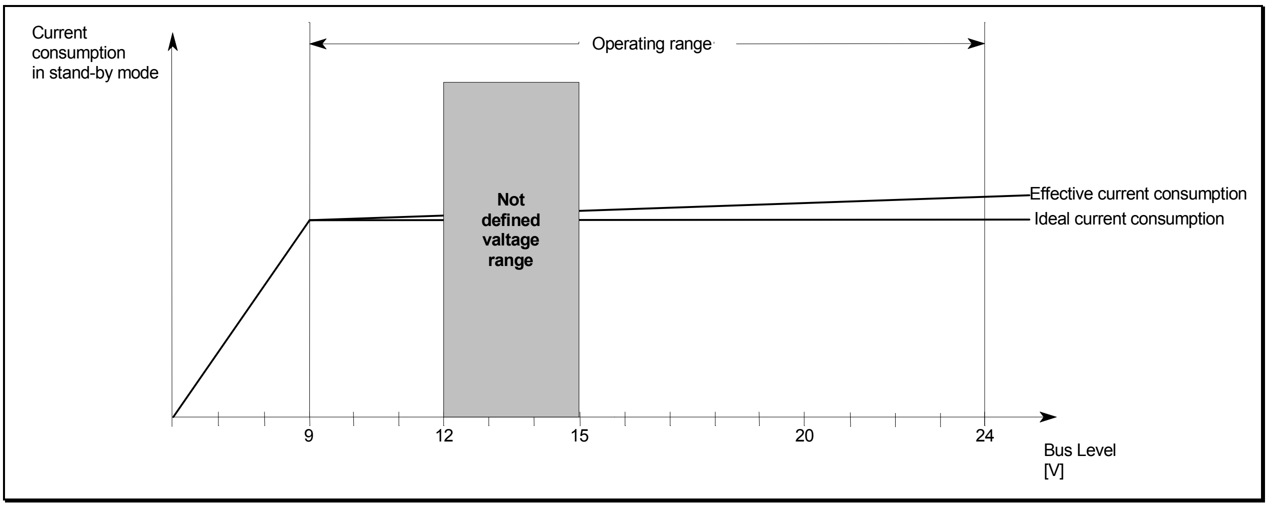
Electronic Circuit with electric Separation through Opto-Coupler

10.7 Current Draw of an eBUS Device

An eBUS device has a certain stand-by current consumption $I_{IDLE} \geq 0$. The current draw I_{IDLE} in the HIGH segment should resemble that in LOW segment, namely:

$$I_{IDLE}(HIGH) \approx I_{IDLE}(LOW)$$

A constant current consumption would, of course, be desirable, but the realisation represents a problem.



With a viable eBUS device, within an operating range of 9V to 24V may vary. If it varies excessively, it will limit the maximum number of participants on a bus:

$I_\Delta = \| I_{IDLE}(LOW) - I_{IDLE}(HIGH) \|$	Max. No. of Participants	Declaration of device as
$I_\Delta \leq 0.307$ mA	228	Class 0
$I_\Delta \leq 1$ mA	70	Class 1 or indication of I_Δ
$I_\Delta \leq 2$ mA	35	Class 2 or indication of I_Δ
2 mA < $I_\Delta \leq 5$ mA	$14 \leq N_{max} < 35$	Class 3 and indication of I_Δ

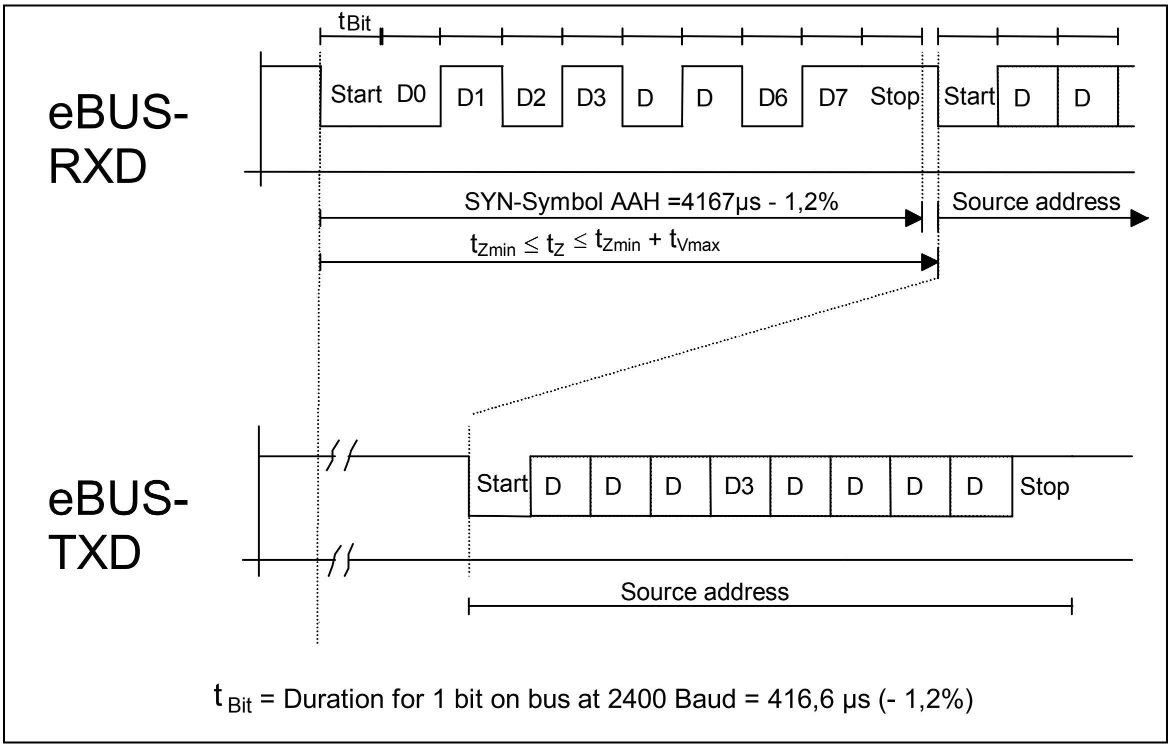
The max. number of participants N_{max} calculated as follows:

$$N_{max} = 70\ mA / I_{\Delta}$$

As a general rule, the sum of all variations in terms of current consumption must not exceed 70 mA.

10.8 Timing following SYN-Symbol

Bus access following a received synchronisation symbol and a transmission-ready participant



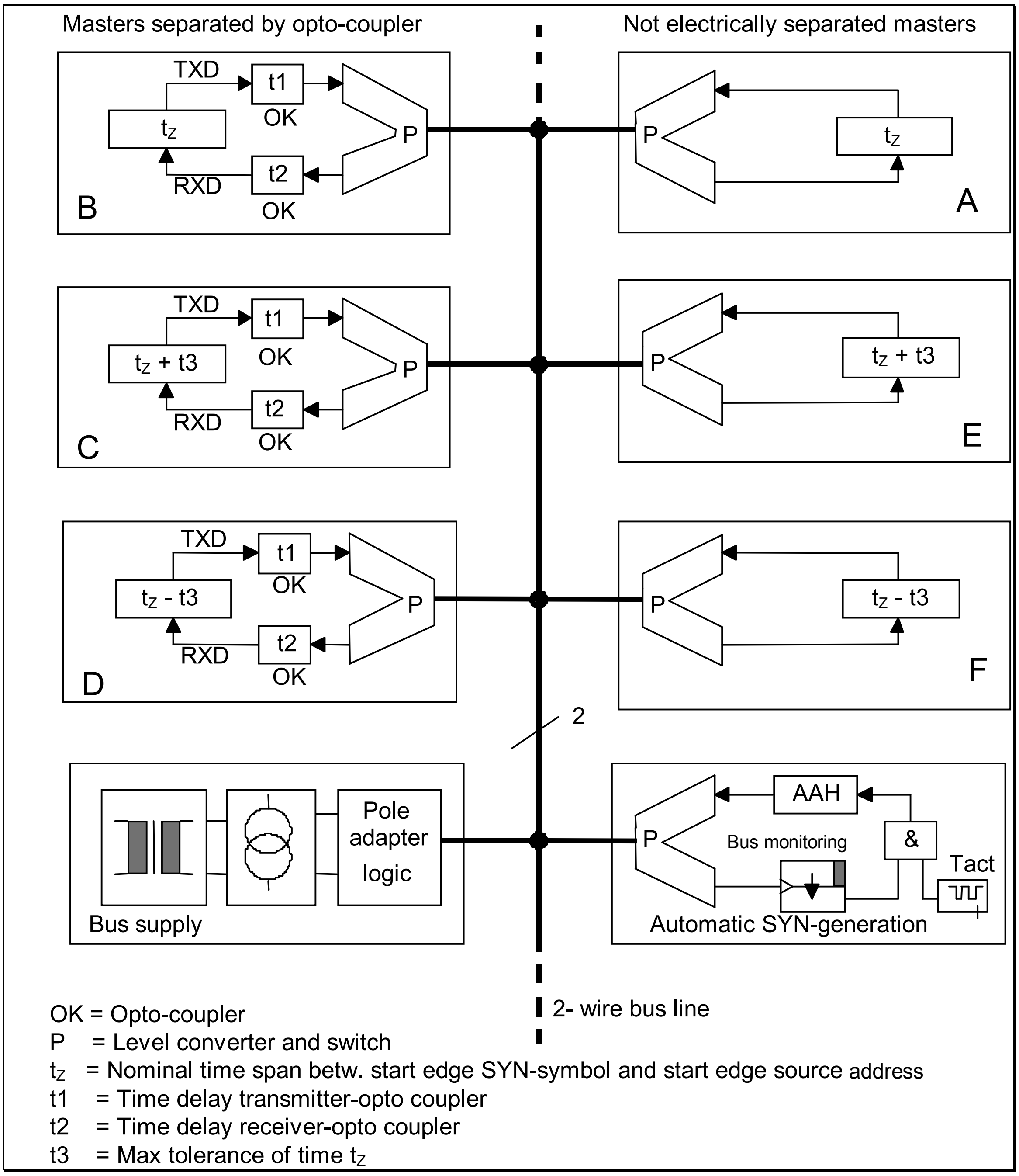
The indicated time spans refer all to the falling edge of the start bit on the bus. All masters on the bus must have their ISR (Interrupt Service Routine) properly designed. That is to say, in case of an access event to the bus, following the synchronisation-symbol, the start edge of the source address has to appear on the bus after expiration of the access time interval $\leq t_Z \leq t_{Zmax}$.

This synchronisation is required only following a SYN-symbol, in order for the arbitration procedure to operate properly.

The maximum delay time t_{Vmax} is defined in chapter 6.3, namely:

t_{Zmin} = 4300 µs as well as $t_{Zmax} = t_{Zmin} + t_{Vmax}$

10.9 Run Times on the BUS



eBUS-Master with and without electrical separation and with various time tolerance values

The individual bus participants must adhere to the conditions for access time t_Z on the bus. I.e. they must consider the run times in the level converter and in the opto-couplers in the transmission and receiver path, when designing their software.

11 Supplement

Under normal conditions, the eBUS is operated in asynchronous mode. That means that all devices must have command of the asynchronous bus access. For certain applications, it is feasible to employ a synchronous access procedure on this basis. This synchronous access procedure is described in this specifications supplement. It may be activated from the asynchronous access procedure and in context with management functions as described in OSI layer 7 under exclusion of a mixed operation.

11.1 The synchronous Access Procedure

Each synchronously sending participant must also have command of the asynchronous access procedure. The mixed operation with synchronously and asynchronously accessing participants is not specified.
Each eBUS-master has a priority class counter. Inside an eBUS-master the priority class counter is a variable, which changes its value in the range from 0 to 4. The priority class counter in an eBUS-master is only incremented, if two SYN-symbols in sequence are detected on the bus. If the priority class counter has the value 4, and the counter must be incremented, the new value is 0.
These are the advantages of the synchronous access procedure:

- Each eBUS-Master is capable to transmit one telegram in one eBUS-cycle. An eBUS-master can transmit a telegram in its priority class - it is not a must. Because of this property the eBUS gets a real-time-feature. The time behaviour of the eBUS is more transparent than in the asynchronous mode. A better composeability of eBUS-components from different manufactures can easier be achieved.
- Inside a priority class are no extended arbitrations with the help of ASYN´s. Due to this fact the transmit capacity in synchronous mode is greater than in the asynchronous mode.
- It is easy to control, if the participiants apply to the regulations of the synchronous access procedure. It is much more difficult to control, if the lock counter in the asynchronous mode is set to the maximum value.

A master is authorised to transmit only if its priority class is permitted on the bus. Each synchronously sending bus participant must be able to be reset to the initialising phase through a service of layer 7 or by manual means. This in case this device is permitted for operation also outside of purely synchronous configuration.

11.1.1 Network Management

Implementation of a network management according to *eBUS Network Management Specification* (available through the eBUS User Club) is recommended. Implementation of this concept is, however, optional. Furthermore, the following conventions apply.
The organisation of the participants is described in sub-points below, this for normal operation as well for exceptions.

11.1.2 Initialising

A node that would like to transmit according to synchronous access method must first determine, if this access procedure is actually adjusted throughout the net. For this, it will proceed through the following algorithm following its own activation:

A: The node does not know, if all other bus participants will send synchronously:
a) The node may be operated also outside of purely synchronous configuration:

1. Adjust own bus access for asynchronous operation.
2. Determine which devices are bus-active and if they master synchronous bus access, using services of OSI-layer 7.
3. In case **all** bus-active devices master synchronous bus access, the bus access procedure will be adjusted for synchronous access, using a service of OSI-layer 7.
4. In case access adjustment of the access mode was successful throughout the net: switch own bus access mode to synchronous.

b) The node may only be operated within purely synchronous configurations:
1. Deactivation of transmission access as master.

B: The node knows that all other bus participants may and will send synchronously:

1. Adjustment for synchronous bus access method.

Immediately following initialisation, all master addresses are authorised for transmission.

11.1.2.1 Normal Operation

In general, the transmission authority of a bus participant depends on the priority class of its master address. As a rule, the synchronous access procedure limits the simultaneous transmission authority always for masters of same priority class. In case of competing bus access attempts, this makes for immediate assertion by the respective highest-ranking master. Fairness is achieved in that cyclic availability of a slot for each priority class. A slot comprises up to 5 subsequent bus access procedures by masters of same priority class.

Within its slot, a master may transmit only one time.

The switching from one slot to the next is initiated by an AUTO-SYN symbol on the bus. Within one slot, all masters of the assigned, valid priority class are transmission-authorized. This asserts that a maximum of 5 masters will transmit.

After the fact that through the initialising phase the synchronous bus access mode is activated throughout the net, all masters are at first transmission-authorized. The first message that is sent to the bus following bus access initialisation marks the beginning of the first bus cycle and determines the priority class of the first slot. Following the subsequent SYN-symbol, all masters are no longer transmission-authorized, rather only those of identical priority class of the sender. The sender of the first message must wait for the next bus cycle, before it is allowed to transmit again.

In the subsequent slot, all masters of next lower priority class are transmission-authorized. Following a slot for priority class 0 follows a slot for class 1, then a slot for class 3, class 7 and, finally, a slot for class F. In order to close a bus cycle even if another class than class 0 has initiated the bus, after a slot for class F a slot for class 0 will follow.

A total of 5 slots will pass through one bus cycle, in order to serve all masters with authorisation to transmit to the bus. Only after all masters have received this transmission authorisation, a new bus cycle with identical slot priority class assignment will follow immediately after the just expired bus cycle.

To realise the synchronous bus access, each participant may manage the present priority class over a *Priority Class Counter*, that is incremented following each AUTO-SYN symbol and as closed counting chain may assume one of these five values (0, 1, 2, 3, 4).

11.1.2.2 Exceptions

11.1.2.2.1 Loss of a Node

There is no need for special services, see chapters 11.1.2 Initialisation and 11.1.2.1 Normal Operation.

11.1.2.2.2 Adding a Node during Operation

All bus participants will be reset to the initialisation phase and are obligated to re-initialise the synchronous access procedure according to chapter 11.1.2 Initialisation throughout the net.

12 List of Modifications

12.1 Modifications Version 1.2 → Version 1.2.1

Date	Page V1.2	Page V1.2.1	Description	Author
05/2000	6	6	Explanation Slave Address of a Master	Frank Fischer Maria Scheurer
	14/26	14/26	Additional Explanations to the Network Management	

Date	Page V1.2(eng)	Page V1.2(eng)	Description	Author
03/2001			translation errors	Maier (Lamberti) FF
03/2001			Additional Explanation to the Supplement	Maier (Lamberti) FF
07/2001			translation mistakes	M. Held (Dungs)

12.2 Änderungen Version 1.2.1 → Version 1.2.2

Date	Page V1.2.1(eng)	Page V1.2.2(eng)	Description	Author
12/2000	6	6	Explanation: A master's slave address –using the address FFh as example.	Sven-Uwe Landvoigt
	8	8	Hint : The number of useful bytes [NN] if A9 or AA respectively is substituted	
	13	13	Corrected the formula for the calculation of t_{Vmax}.as well as the table about limits.	

12.3 Modifications Version 1.2.2 → Version 1.3

Date	Page V1.2.2(eng)	Page V1.3(eng)	Description	Author
02/2001		14	Added a hint about the lock_counter_max value	Sven-Uwe Landvoigt
		17 ff	Expansion by chapters „More than one character-oriented SYN-Symbol Generator in a single eBUS system“ and „Flow chart of an example implementation for a software SYN-symbol generator“	

12.4 Modifications Version 1.3 → Version 1.3.1

Date	Page V1.3(eng)	Page V1.3.1(eng)	Description	Author
03/2007		footnote	Change of the document from eBUS User Club to eBUS Interest Group	Frank Hoffmann