Features

- **Fully Compliant to VAN Specification ISO/11519-3**
- **Handles All Specified Module Types**
- **Handles All Specified Message Types**
- **Handles Retransmission of Frames on Contention and Errors**
- **3 Separate Line Inputs With Automatic Diagnosis and Selection**
- **Normal or Pulsed (optical and radio mode) Coding**
- **VAN Transfer Rate: 1 Mbit/s maximum**
- **SPI/SCI Interface**
- **SPI Transfer Rate: 4 Mbit/s maximum**
  SCI Transfer Rate: 125 Kbit/s maximum
- **Idle and Sleep Modes**
- **128 bytes of General Purpose RAM**
- **14 Identifier Registers With All Bits Individually Maskable**
- **6-source Maskable Interrupt Including An Interrupt-on-reset To Detect Glitches on the Reset Pin**
- **Integrated crystal or Resonator Oscillator with Internal Baud Rate Generator and Buffered Clock Output**
- **Single +5V Power Supply**
- **0.5 µm CMOS Technology**
- **SO 16 Packaging**

Description

The TSS463B is a circuit which allows the transfer of all the status information needed in a car or truck over a single low-cost wire pair, thereby minimizing electrical wire usage.

It can be used to interconnect powerful functions and to control and interface car body electronics (lights, wipers, power window, etc.).

The TSS463B is fully compliant with the VAN ISO standard ISO/11519-3. This standard supports a wide range of applications such as low-cost remote controlled switches, typically used for lamp control, up to complex, highly autonomous, distributed systems which require fast and secure data transfers.

The TSS463B is a microprocessor interfaced line controller for mid to high complexity bus-masters and listeners like dashboard controllers, car stereo or mobile telephone CPUs.

The microprocessor interface consists of a 256 byte RAM and register area divided into 11 control registers, 14 channel register sets and 128 bytes of general purpose RAM, used as a message storage area, and a 6-source maskable interrupt.

The circuit operates in the RAM using DMA techniques, controlled by the channel and control registers. This allows virtually any microprocessor including SPI/SCI interface to be connected with ease with the TSS463B.

Messages are encoded in enhanced Manchester code, and an optional pulsed code for use with an optical or radio link, at a maximum bit rate of 1 Mbit/s. The TSS463B analyzes the messages received or transmitted according to 6 different criteria including some higher level checks.

In addition the bus interface has three separate inputs with automatic source diagnosis and selection, allowing for multibus listening or the automatic selection of the most reliable source at any time if several line receivers are connected to the same bus.

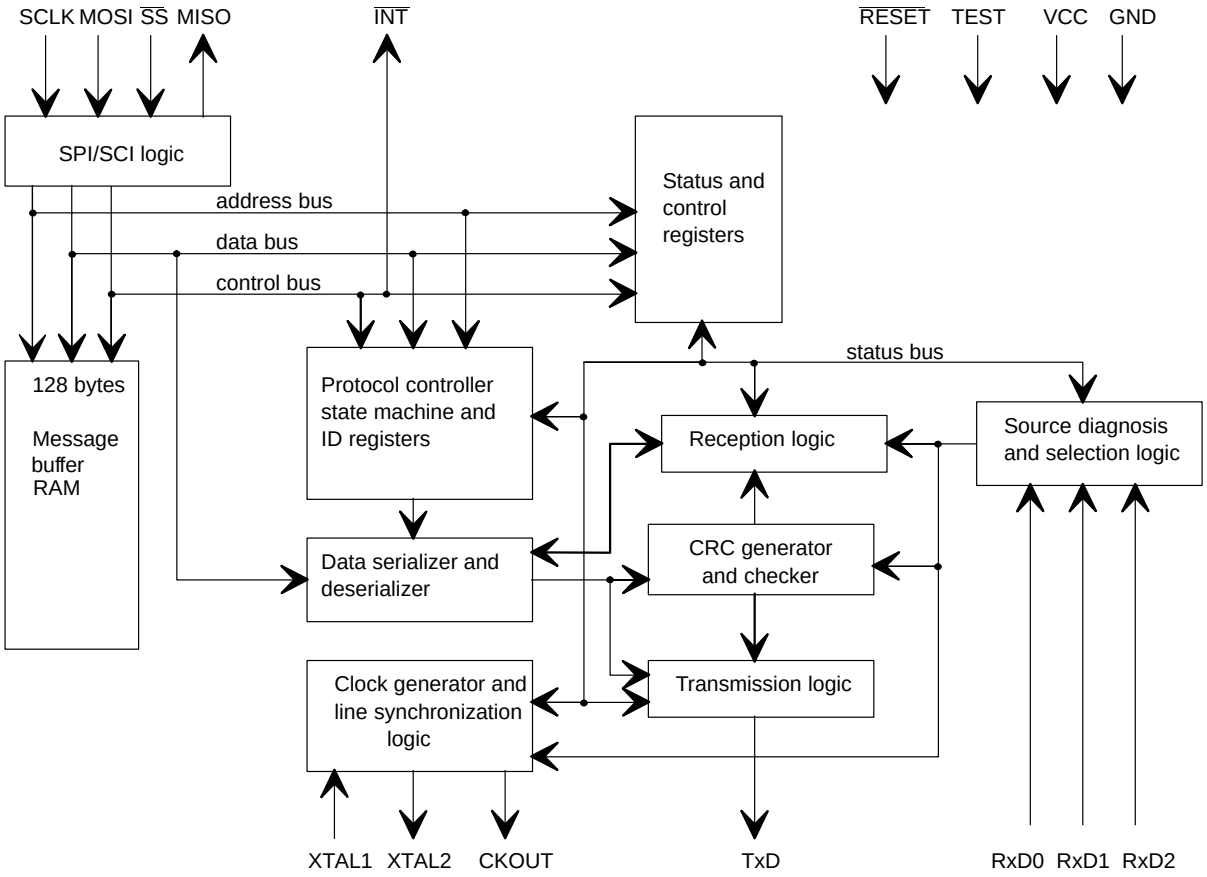
VAN Data Link Controller with Serial Interface

TSS463B

Rev. 4102C–AUTO–04/02

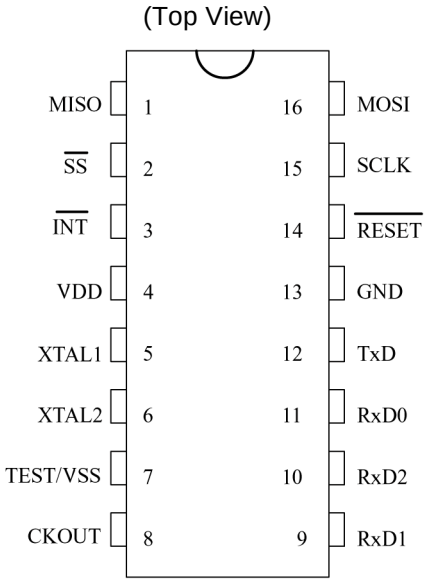
Block Diagram

Figure 1. Block Diagram



Pin Configuration

Figure 2. Pin Configuration



I/O Type	Pin Name	Pin No	Pin Function
O 3-state	MISO	1	SPI/SCI Data Output
I trigger CMOS	$\overline{SS}$	2	SPI/SCI Slave Select (active low)
Open drain	$\overline{INT}$	3	Interrupt (active low)
Power	VDD	4	+ 5 V power supply
I CMOS	XTAL1	5	Crystal oscillator or clock input pin from 1 to 16 Mhz
O	XTAL2	6	Crystal oscillator output pin
Ground	TEST/VSS	7	Test mode input
O	CKOUT	8	Buffered clock output

I/O Type	Pin Name	Pin No	Pin Function
I CMOS Pull down	RxD1	9	VAN bus input 1
I CMOS Pull down	RxD2	10	VAN bus input 2
I CMOS Pull down	RxD0	11	VAN bus input 0
O 3-state	TxD	12	VAN bus output
Ground	GND	13	
I trigger CMOS pullup	$\overline{RESET}$	14	Hardware Reset (active low)
I trigger CMOS	SCLK	15	SPI/SCI Clock Input
I trigger CMOS	MOSI	16	SPI/SCI Data Input

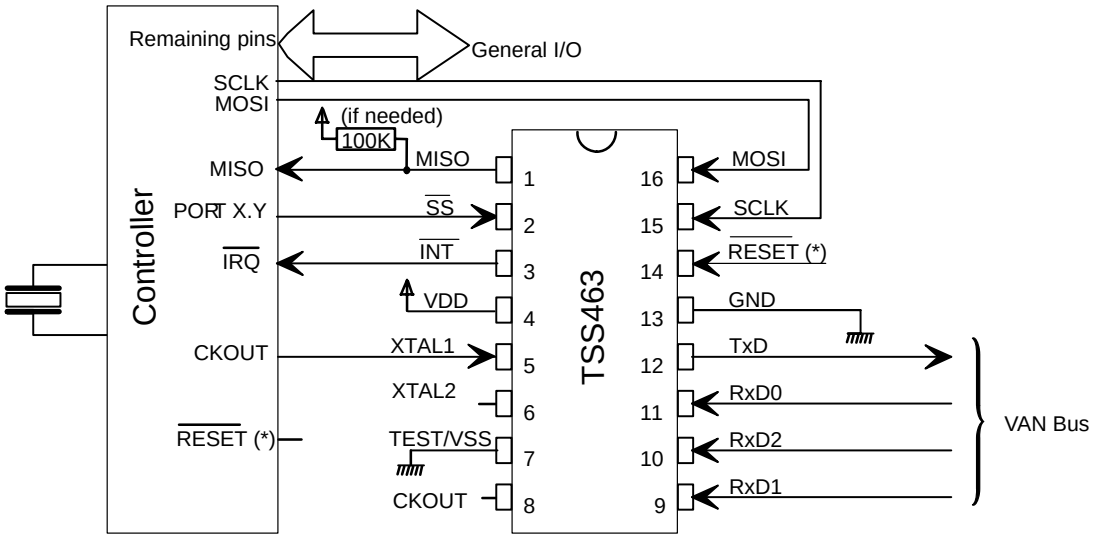
Application

The TSS463B is a microprocessor controlled line controller for the VAN bus. It can interface to virtually any microprocessor which includes SPI or SCI interface.

- On the first hand, the TSS463B provides one full Motorola compatible SPI interface.
- On the other hand, it includes one full compatible Intel UART (mode 0 only).
- And finally, one 9-bits SCI interface is also integrated.

The circuit also features a single interrupt pin. This pin can be treated as level sensitive, i.e. if there is a pending interrupt inside the circuit when another interrupt is reset the $\overline{INT}$ pin will emit a high pulse with the same pulse width as the internal write strobe (typically 20 ns).

Figure 3. Typical Application With Motorola SPI Mode



Note: (*)The TSS463B $\overline{RESET}$ pin can either be connected to GND through a 1 µF capacitor, or the µC $\overline{RESET}$ pin or unconnected (inactive with internal pull-up).

Leaving MISO output pin floating in high impedance mode slightly increases standby consumption. A 100KΩ pullup/pulldown resistor is recommended.

Microprocessor Interface

The processor controls the TSS463B by reading and writing the internal registers of the circuit. These registers appear to the processor as regular memory locations.

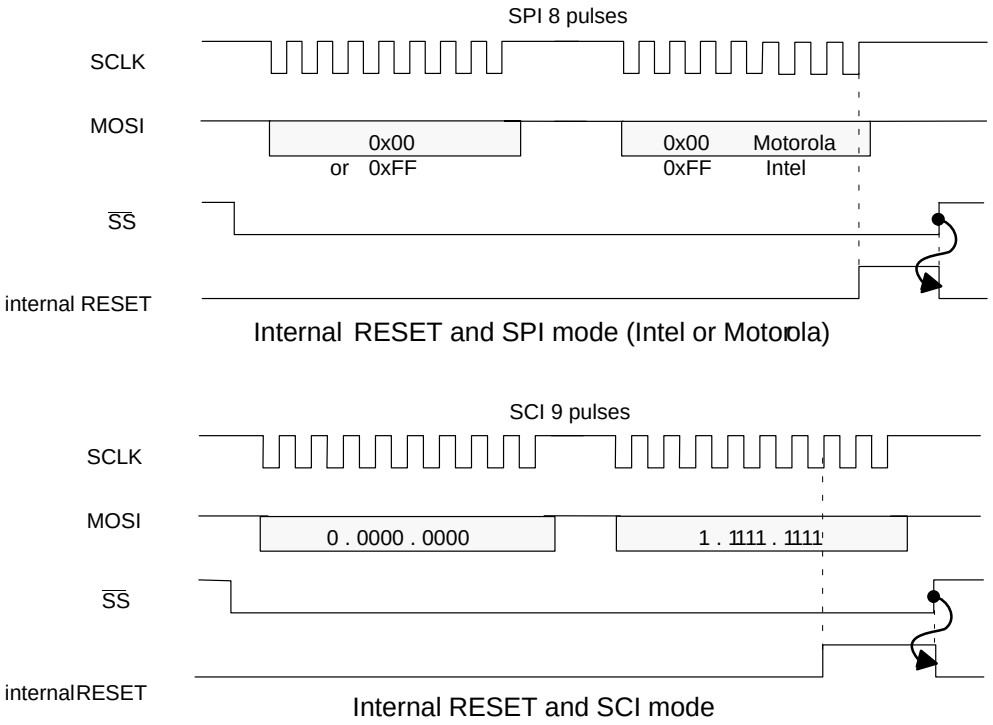
Interface Modes

The TSS463B must be connected with an SPI or SCI serial interface.The following section provides information switching from one mode to another.

Motorola SPI Mode

The first two bytes to be sent by the master (CPU) are called «Initialization Sequence»: This sequence provides a proper asynchronous RESET in the TSS463B and it selects the Motorola SPI, Intel SPI or the SCI serial mode. This initialization sequence is shown on figure 4: two 0x00 will cause an internal RESET and assert the Motorola SPI mode, two «0xFF» will provide an internal RESET and assert the Intel SPI mode and «9 bits to 0 followed by 0xFF or 0xFE» will generate an internal RESET and assert the 9-bits SCI mode.

Figure 4. Mode Configuration Byte

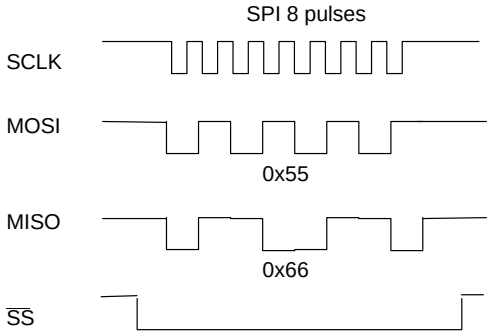


The Motorola Serial Peripheral Interface (SPI) is fully compatible with the SPI Motorola protocol. The interface is implemented for slave-mode only (the TSS463B can not generate SPI frames by itself).

The SPI interface allows the interconnection of several CPUs and peripherals on the same printed circuit board.

The SPI mode interface consists of 4 pins: separate wires are required for data and clock, so the clock is not included in the data stream as shown on figure 5. One pin is needed for the serial clock SCLK, two pins for data communication MOSI and MISO and one pin for Slave Select $\overline{SS}$.

Figure 5. SPI Data Stream



SCLK: Serial Clock

The master device provides the serial clock for the slave devices. Data is transferred synchronously with this clock in both directions. The master and the slave devices send/receive a data byte during an eight clock pulse sequence.

MOSI: Master Out Slave In

The MOSI pin is the master device data output (CPU) and the slave device data input (TSS463B). Data is transferred serially from the master to the slave on this line; most significant bit (MSB) first, least significant bit (LSB) last.

MISO: Master In Slave Out

The MISO pin is configured as the slave device data output (TSS463B) and as master device data input (CPU). When the slave device is not selected ($\overline{SS}$ = 1), this pin is in high impedance state.

$\overline{SS}$: Slave Select

The $\overline{SS}$ pin is the slave chip select. It is low active. A low state on the Slave Select input allows the TSS463B to accept data on the MOSI pin and send data on the MISO pin. The Slave Select signal must not toggle between each transmitted byte and so, should be left at a low level during the whole SPI frame. $\overline{SS}$ must be asserted to inactive high level at the end of the SPI frame.

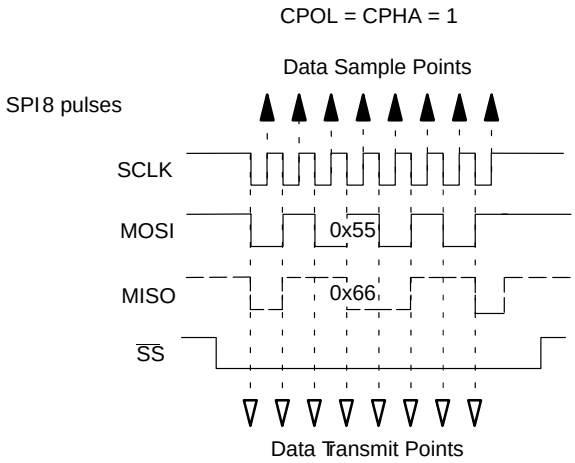
As mentioned earlier, if $\overline{SS}$ is not asserted, MISO pin is in a high impedance state and incoming data is not driven to the serial data register.

SPI Protocol

The general format of the data communication in the SPI frame between the TSS463B and the host is a bit-for-bit exchange on each SCLK clock pulse. Data is arranged in the TSS463B such that the significance of a bit is determined by its position from the start for output and from the end for input, most significant bit (MSB) is sent first. Bit exchanges in multiples of 8 bits are allowed.

The Idle Clock Polarity (CPOL) and the Clock Phase (CPHA) are not programmable: the CPOL and CPHA values to be programmed in the master (CPU) are CPOL=CPHA=1. This is available for all modes. Waveforms with transmit and sample points are shown on figure 6.

Figure 6. CPOL and CPHA in the TSS463B



At the beginning of a transmission over the serial interface, the first byte is the address of the TSS463B register to be accessed. The next byte transmitted is the control byte which determines the direction of the communication. The following bytes are data bytes (consecutive bytes are written in or read from Address, Address+1, Address+2, ..., Address+n with n = 0 to 28).

To make sure the TSS463B is not out of synchronization, the SPI interface will transmit data «0xAA» and «0x55» on the MISO pin during address and control byte time. This way, the master always ensures the TSS463B is well-synchronized. If the TSS463B is out of synchronization, the master can assert the $\overline{SS}$ pin inactive to resynchronize the SPI interface or can assert the $\overline{RESET}$ pin active or can send an initialization sequence. When the $\overline{SS}$ pin is inactive, the SCLK is allowed to toggle. This will have no effect on the TSS463B SPI module.

SPI Control Byte

The SPI control byte is transmitted by the master (CPU) to the TSS463B. It specifies whether it is a TSS463B Write or Read.

Table 1. SPI Control Byte

7	6	5	4	3	2	1	0
DIR	1	1	0	0	0	0	0

DIR: Serial Transfer Direction

Zero: Read Operation. The data bytes will be read by the master (CPU) from the TSS463B.

One: Write Operation. The data bytes will be written by the master (CPU) to the TSS463B.

In both cases, address auto-increment mechanism will take place when more than one data byte is read or written. This mechanism is inhibited when address value reaches 0xFF.

The seven following bits are reserved and must be equal to: 1100000.

When the master (CPU) conducts a write, it sends an address byte, a control byte and data bytes on its MOSI line. The slave device (TSS463B) will send, if well-synchronized, «0xAA» during the address byte and «0x55» during the control byte on its MISO line.

When the master (CPU) conducts a read , it sends an address byte, a control byte and dummy characters («0xFF» for instance) on its MOSI line. In the case of a VAN messages RAM read (VAN frame received), the first data byte sent back by the TSS463B on its MISO pin is the *data length* so the master knows how many dummy characters it must send to read the VAN frame properly. When the TSS463B responds back with data, it will not take care of the MOSI line.

The master must activate and desactivate $\overline{SS}$ between each data frame.

Synchronization bytes must be monitored carefully. For instance, if «0xAA» and «0x55» are not monitored correctly, then the previous transmission may be incorrect too.

A control byte containing «0x00» or «0xFF» is forbidden except during an « Initialization Sequence ».

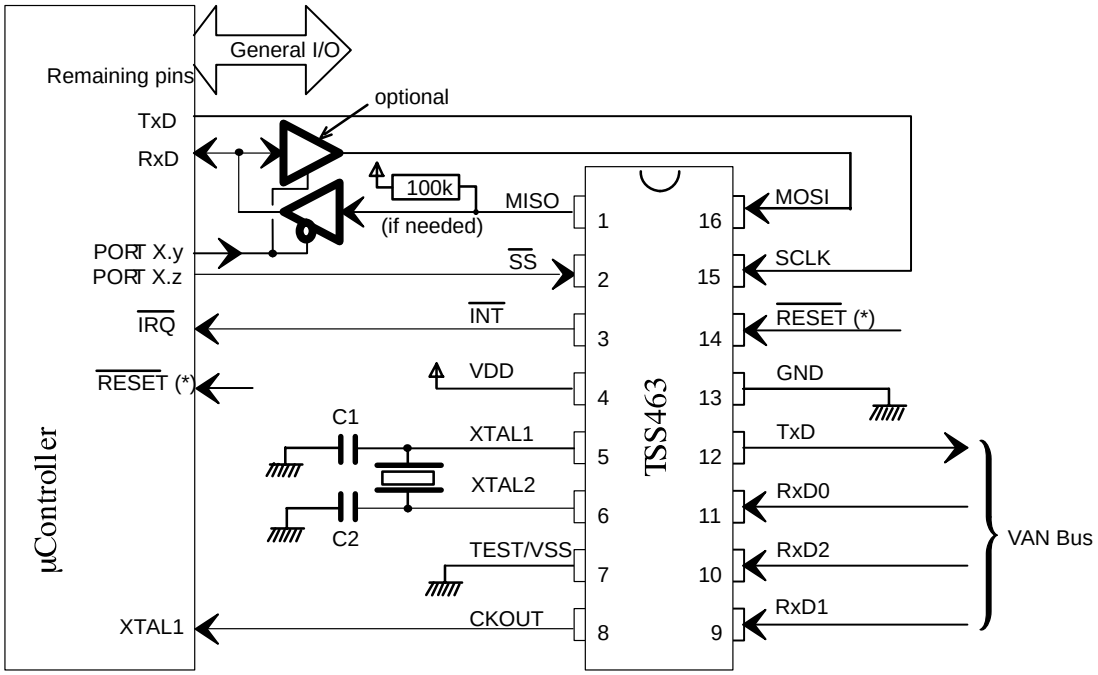
Intel SPI Mode

The Intel SPI mode is the second type of interface. As mentioned earlier, the TSS463B enters this mode if the Initialization Sequence contains (first two bytes received) «0xFF, 0xFF».

This mode is fully compatible to the Intel UART serial interface programmed in mode 0 only. That means it is the same as Motorola SPI mode (same CPOL and CPHA) but with inverted communication sense (LSB first and MSB last). The protocol is also the same.

However, from the master point of view (host microcontroller), the hardware is different. Figure 8 shows how to connect the TSS463B and Intel type microcontroller.

Figure 7. Typical Application With the 8051 UART in Mode 0.



Note: (*)The $\overline{RESET}$ pin can either be connected to GND through a 1 µF capacitor, or the µC $\overline{RESET}$ pin or unconnected (inactive with internal pull-up).

The master device provides the serial clock on the TxD pin and is still connected to SCLK pin of the slave device.

Then, the RxD replaces the MOSI and MISO pins and is a bidirectional pin. To achieve a correct communication, the user should add a little hardware to connect the master RxD pin to the MOSI-MISO slave pins.

Figure 8 proposes two 3-state buffers controlled by the master through a general purpose I/O pin.

It is obvious that, in this Intel SPI mode, the master cannot monitor the «0xAA and 0x55» synchronization bytes while sending the address and control bytes. It is the only exception of this mode compared with the Motorola SPI mode.

SCI Mode

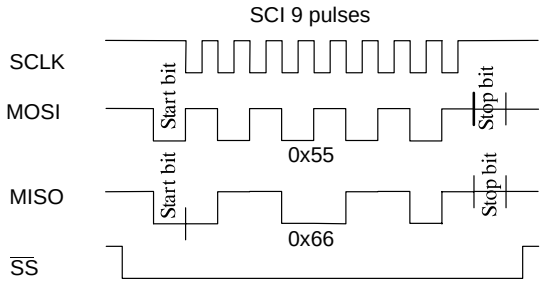
The SCI mode is the third type of interface. As mentioned earlier, the TSS463B enters this mode if the Initialization Sequence contains (first two bytes received) «0x00, 0xFF».

The SCI is compatible to a 9-bits SCI protocol. The interface is implemented for slave-mode only (the TSS463B can not generate SCI frames by itself).

The SCI interface allows an interconnection of several CPUs and peripherals on the same printed circuit board.

The SCI mode interface consists of 4 pins: separate wires are required for data and clock, so the clock is not included in the data stream as shown on figure 9. One pin is needed for the serial clock SCLK, two pins for data exchange MOSI and MISO and one pin for Slave Select $\overline{SS}$.

Figure 8. SCI Data Stream



SCLK: Serial Clock

The master device provides the serial clock for the slave devices. Data is transferred synchronously with this clock in both directions. The master and the slave devices exchange a data byte during a **nine** clock pulses sequence. However, the TSS463B will only monitor 8 bits on its MOSI line and send 9 bits on its MISO line.

MOSI: Master Out Slave In

The MOSI pin is the master device data output (CPU) and the slave device data input (TSS463B). Data is transferred serially from the master to the slave on this line; **least** significant bit (LSB) first, **most** significant bit (MSB) last. The TSS463B will only monitor 8 bits starting from the LSB to MSB-1.

MISO: Master In Slave Out

The MISO pin is configured as the slave device data output (TSS463B) and as master device data input (CPU). When the slave device is not selected ($\overline{SS}$ = 1), this pin is in high impedance state. The value of the MSB (9^{th} bit) sent on the MISO pin will always be «1» and should not be used by the master.

$\overline{SS}$: Slave Select

The $\overline{SS}$ pin is the slave chip select. It is low active. A low state on the Slave Select input allows the TSS463B to accept data on the MOSI pin and send data on the MISO pin. The Slave Select signal must not toggle between each transmitted byte and therefore,

should be left at a low level during the whole SCI frame. $\overline{SS}$ must be asserted to inactive high level at the end of the SCI frame.

As mentioned earlier, if SS is not asserted, MISO pin is in high impedance state and incoming data is not driven to the serial data register.

SCI Protocol

Same as the SPI protocol described earlier except for data arranging (LSB first and MSB last).

Only 8 bits are monitored by the TSS463B and master must monitor the 8 first bits too (9^{th} bit always equal to 1).

SCI Control Byte

Same as the SPI control byte.

Clocks and Speed Considerations

SCLK and XTAL Clocks

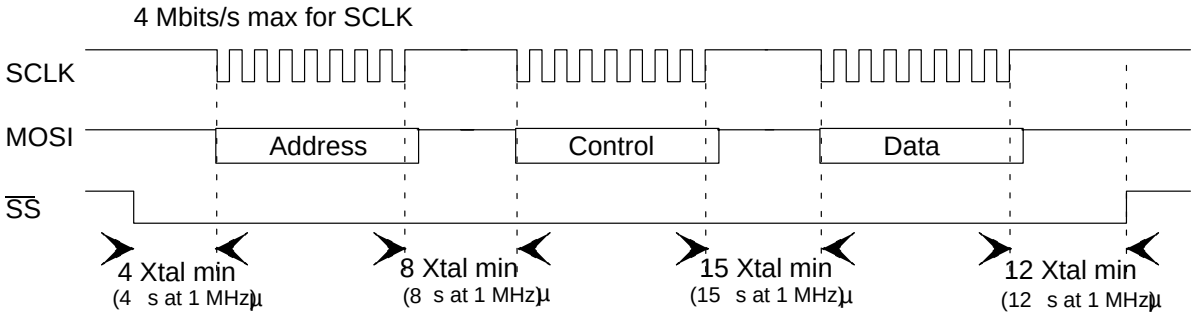
The SPI/SCI speed rate is given by the CPU producing SCLK. XTAL clock controls the speed rate on the VAN bus. The two clocks are asynchronous but a minimum SPI/SCI interframe spacing must be apply according to XTAL clock.

Intel and Motorola SPI Modes

Within an SPI byte, the maximum speed allowed on the MOSI line is 4 Mbits/s.

For example, when using a 1 Mhz oscillator (sufficient to provide 62.5 kTS/s on the VAN bus) the minimum inter-character delay is 12µs (12 oscillator periods). Speed considerations are detailed on Figure 9.

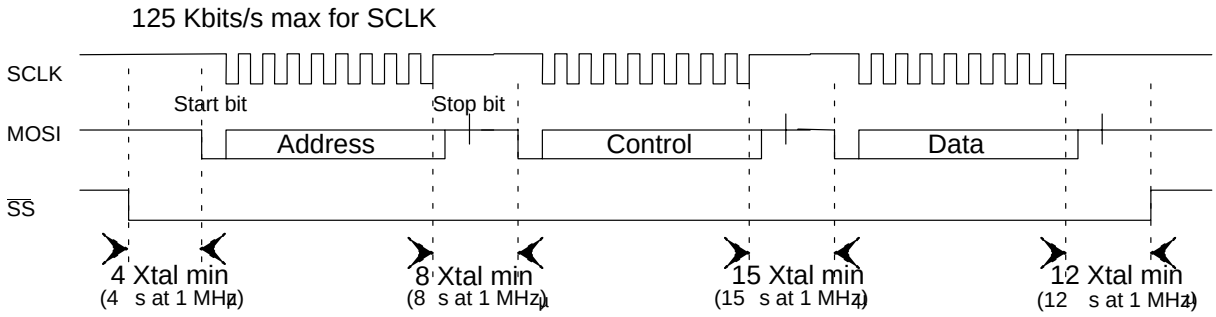
Figure 9. SPI Speed Considerations



SCI Mode

Within an SCI 9-bits data, the maximum speed allowed on the MOSI line is 125 Kbits/s. When using a 1 Mhz oscillator, the data transfer speed and the minimum delay time between SCI bytes are shown on Figure 10.

Figure 10. SCI Speed Considerations



Interrupts

If an event occurs in the TSS463B that needs the attention of the processor, this will be signalled on the active low, open drain interrupt request pin. The event that creates this request is controlled by the internal registers.

Every time the microprocessor accesses any of the interrupt registers (addresses 0x08 to 0x0B) the $\overline{INT}$ pin will be released momentarily. This enables the TSS463B to work with processors that have either edge or level sensitive interrupt inputs.

Reset

The reset is applied asynchronously or synchronously to the XTAL clock.

Asynchronous Reset

It can be done either by the $\overline{RESET}$ pin (hardware asynchronous reset) or by software (software asynchronous reset).

The $\overline{RESET}$ pin is a CMOS trigger input with a pull-up resistor (~ 70 KΩ). An external 1 µF capacitor to GND provides to $\overline{RESET}$ pin an efficient behavior.

The asynchronous software reset is made by the «Initialization Sequence» described in “Motorola SPI Mode” on page 5.

Two «0x00» bytes provide an asynchronous software reset and configure the TSS463B in the Motorola SPI mode while two «0xFF» bytes provide a reset and configure the component in the Intel SPI mode and «0x00 followed by 0xFF » provide a reset and configure the component in the SCI mode. The $\overline{SS}$ pin must be asserted as shown on figure 12. The SPI/SCI logic will monitor these two bytes and provide an internal reset pulse asserting the TSS463B in the right mode.

Synchronous Reset

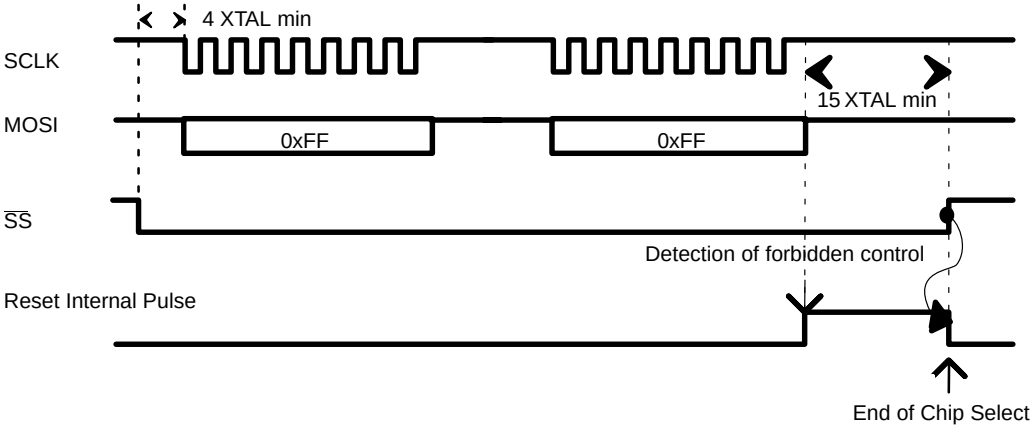
A synchronous reset (regarding XTAL clock) is also available on the TSS463B during current operation. It is made through the GRES command bit of the Command Register (address 0x03).

The two kinds of reset are ordered and filtered. Then the internal reset, always asserted asynchronously, enables the internal oscillator. Then it waits for 12 clock periods the oscillator stability.

The different blocks of the TSS463B need to be turned on synchronously. So the release of the internal reset is synchronous and a loose of clock can let the TSS463B in permanent reset after applying Reset.

It is important to note that even after a reset on the $\overline{RESET}$ pin, the user should wait for 12 clock periods before sending the «Initialization Sequence» in order to select the SPI or SCI mode (because the default mode after a hardware reset is the Motorola SPI mode).

Figure 11. Asynchronous Software Reset with UART Intel mode



Oscillator

An oscillator is integrated in the TSS463B, and consists of an inverting amplifier of which the input is XTAL1 and the output XTAL2.

A parallel resonance quartz crystal or ceramic resonator must be connected to these pins. As can be seen from Figure 8. , two capacitors have to be connected from the crystal pins to ground. The values of C2 depend on the frequency chosen and can be selected using the nomograph given in Figure 41.

If the oscillator is not used, then a clock signal must be fed to the circuit via the XTAL1 input.

Note, that this pin will behave as a CMOS level compatible Schmitt trigger input.

In this case the XTAL2 output should be left unconnected. The oscillator also features a buffered clock output pin CKOUT. The signal on this pin is directly buffered from the XTAL1 input, without inversion.

There is one more pin used for the oscillator. The TEST/VSS pin is in fact its ground, and unless this pin is firmly connected to ground, with decoupling capacitors, the oscillator will not operate correctly.

The test mode itself, i.e. when the TEST/VSS pin is held high, is only intended for factory use, and the functionality of this mode is not specified in any way.

The TEST/VSS pin is subject to change without notice, the only exception being for incoming inspection tests using the test program.

The clock signal is then fed to the clock generator that generates all the necessary timing signals for the operation of the circuit. The clock generator is controlled by a 4-bit code called the clock divider.

$$f(TSCLK) = \frac{f(XTAL1)}{n \times 16}$$

Table 2. Clock Divider

Clock Divider	Divided by	16 MHz*		12 MHz*		8 MHz	
		KTS/s	Kbits/s	KTS/s	Kbits/s	KTS/s	Kbits/s
0000	1	1000	800	750	600	500	400
0001	2	500	400	375	300	250	200
0010	4	250	200	187.50	150	125	100
0011	8	125	100	93.75	75	62.5	50
0100	16	62.50	50	46.875	37.5	31.25	25
0101	32	31.25	25	23.438	18.75	15.625	12.5
0110	64	15.625	12.5	11.718	9.375	7.813	6.25
0111	128	7.813	6.25	5.859	4.688	3.906	3.125
1000	1.5	666.667	533.333	500	400	333.333	266.666
1001	3	333.333	266.666	250	200	166.666	133.333
1010	6	166.666	133.333	125	100	83.333	66.666
1011	12	83.333	66.666	62.50	50	41.666	33.333
1100	24	41.666	33.333	31.25	25	20.833	16.666
1101	48	20.833	16.666	15.625	12.50	10.416	8.333
1110	96	10.416	8.333	7.813	6.25	5.208	4.166
1111	192	5.208	4.166	3.906	3.125	2.604	2.083

Note: *These frequencies are not tested.

Table 3. Clock Divider

Clock Divider	Divide by	6 MHz		4 MHz		1 MHz	
		KTS/s	Kbits/s	KTS/s	Kbits/s	KTS/s	Kbits/s
0000	1	375	300	250	200	62.50	50
0001	2	187.50	150	125	100	31.25	25
0010	4	93.75	75	62.50	50	15.625	12.5
0011	8	46.875+	37.5	31.25	25	7.813	6.25
0100	16	23.438	18.75	15.625	12.5	3.906	3.125
0101	32	11.718	9.375	7.813	6.25	1.953	1.562
0110	64	5.859	4.688	3.906	3.125	166.666	133.333
0111	128	500	400	1.953	1.562	83.333	66.666
1000	1.5	250	200	166.666	133.333	41.666	33.333
1001	3	125	100	83.333	66.666	20.833	16.666
1010	6	62.50	50	41.666	33.333	10.416	8.333

Table 3. Clock Divider (Continued)

Clock Divider	Divide by	6 MHz		4 MHz		1 MHz	
		KTS/s	Kbits/s	KTS/s	Kbits/s	KTS/s	Kbits/s
1011	12	31.25	25	20.833	16.666	5.208	4.166
1100	24	15.625	12.50	10.416	8.333	2.604	2.083
1101	48	7.813	6.25	5.208	4.166	1.302	1.042
1110	96	3.906	3.125	2.604	2.083	0.651	0.521
1111	192	1.953	1.5625	1.302	1.042	0.3255	0.2605

VAN Protocol

Line Interface

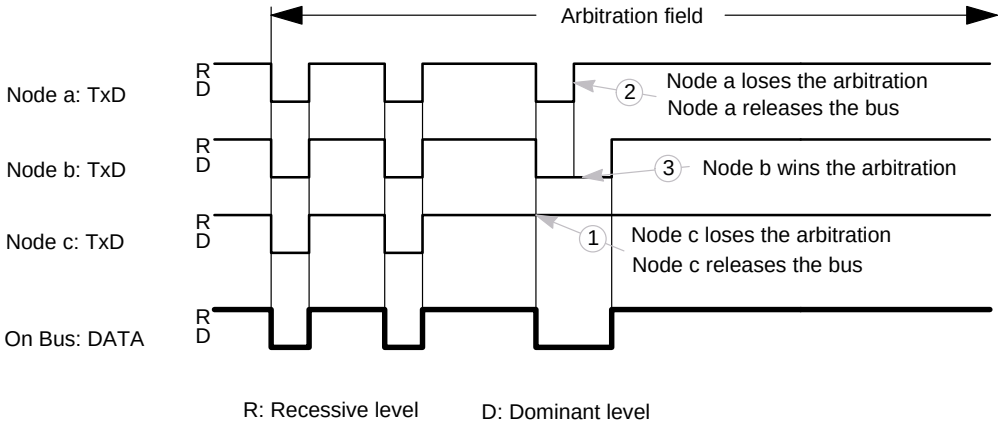
There are three line inputs and one line output available on the TSS463B. Which of the three inputs to use is either programmable by software or automatically selected by a diagnosis system.

The diagnosis system continuously monitors the data received through the three inputs, and compares it with each other and the selected bitrate. It then chooses the most reliable input according to the results.

The data on the line is encoded according to the VAN standard ISO/11519-3. This means that the TSS463B is using a two level signal having a recessive (1) and a dominant (0) state. Furthermore, due to the simple medium used, all data transmitted on the bus is also received simultaneously.

The VAN protocol is hence a CSMA/CD (Carrier Sense Multiple Access Collision Detection) protocol, allowing for continuous bitwise arbitration of the bus, and non-destructive (for the higher priority messa ge) collision detection.

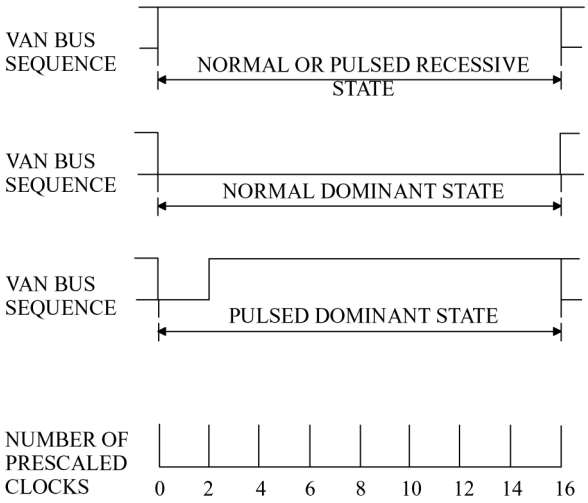
Figure 12. CSMA/CD Arbitration



In addition to the VAN specification there is also a pulsed coding of the dominant and recessive states. This mode is intended to be used with an optical or radio link. In this mode the dominant state for the transmitter is a low pulse, (2x prescaled clocks at the beginning of the bit) and the recessive state is just a high level. When receiving in this mode it is not the state of the signal itself which is decoded, but the edges. Also, reception is imposed on the RxD0 input, and the diagnosis system does not operate correctly.

In addition, in this mode there is an internal loopback in the circuit since optical transceivers are not able to receive the signal that they themselves transmit.

Figure 13. State Encoding



In Figure 14 the pulsed waveforms are shown. In Figure 17 through Figure 23 the low "timeslots" (i.e. blocks of 16 prescaled clocks) should be replaced by the dominant waveform showed in Figure 14 if the correct representations for pulsed coding are to be seen.

VAN Frame

Figure 14. Van Bus Frame

SOF	Identifier Field	Command				Data Field	Frame Check Sum	EOD	ACK	EOF
		EXT	RAK	R/W	RTR					

The VAN bus supports three different module (unit) types:

- First, the *Autonomous* module, which is a bus master. It can transmit Start Of Frame (SOF) sequences, it can initiate data transfers and can receive messages.
- Second, the *Synchronous access* module. It cannot transmit SOF sequences, but it can initiate data transfers and can receive messages.
- And finally, the *Slave* module, which can only transmit using an in-frame mechanism and can receive messages.

Figure 15. Hierarchical Access Methods

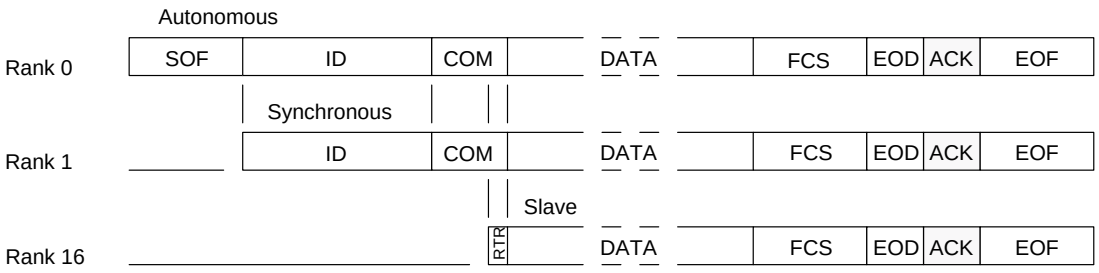
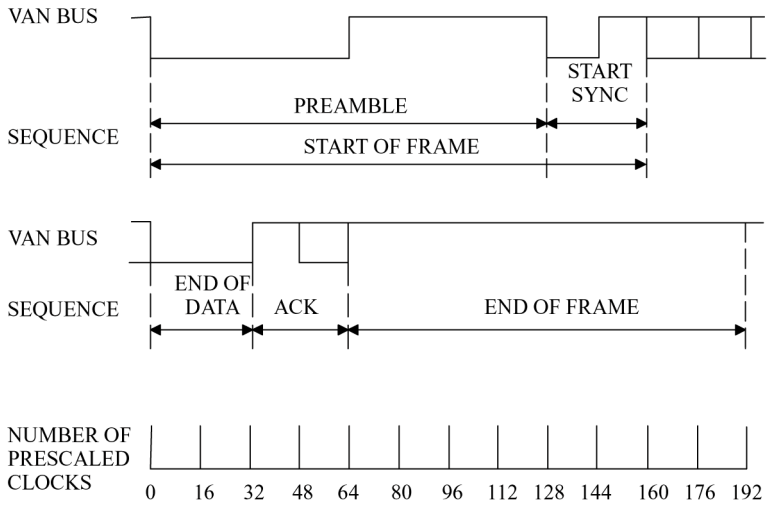


Figure 14 shows a normal VAN bus frame. It is initiated with a Start Of Frame (SOF) sequence shown in Figure 16. The SOF can only be transmitted by an autonomous module. During the preamble the TSS463B will synchronize its bit rate clock to the data received.

Figure 16. Framing Sequences



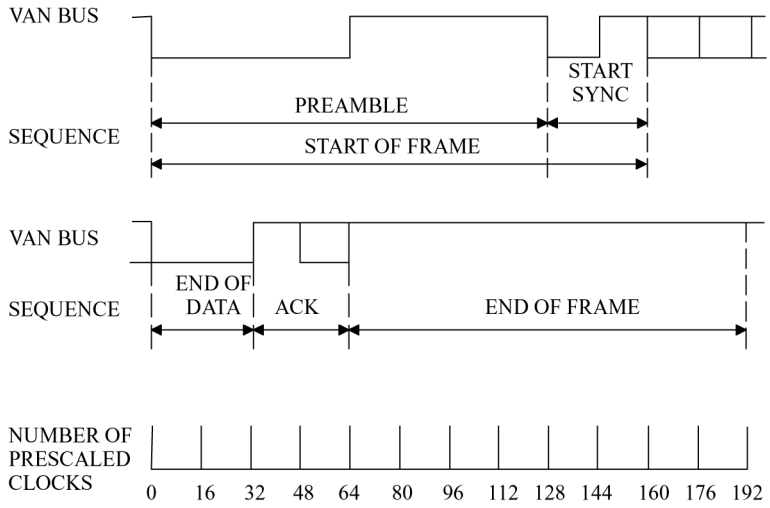
When the complete SOF sequence has been transmitted or received, the circuit will start the transmission or reception of the identifier field.

All data on the VAN bus, including the identifier and Frame Check Sum (FCS), are transmitted using enhanced Manchester code.

In enhanced Manchester code three NRZ bits are transmitted first followed by one Manchester bit, then three more NRZ bits followed by one Manchester bit and so on.

Since the high state is recessive and the low state is dominant, the bus arbitration can be done. If a module wants access to the bus, it must first listen to the bus during one full End Of Frame (EOF) and one full Inter Frame Spacing (IFS) period, to determine whether the bus is free or not (i.e. no dominant states received).

Figure 17. Data Encoding



The IFS is defined to be a minimum of 64 prescaled clock periods. The TSS463B, accepts an IFS of zero prescaled clocks for the reception only of a SOF sequence.

Once the bus has been determined as being free, the module must, if it is an autonomous module, emit an SOF sequence or, if it is a synchronous access module, wait until it detects a preamble sequence.

Will this point there can be several modules transmitting on the bus, and there is no possibility of knowing if this is the case or not. Therefore the first field in which arbitration can be performed is the identifier field. Since the logical zeroes on the bus are dominant, and all data is transmitted with the most significant bit (MSB) first, the first module to transmit a logical zero on the bus will be the prioritized module, i.e. the message that is tagged with the lowest identifier will have priority over the other messages.

It is, however, conceivable that two messages transmitted on the bus will have the same identifier. The TSS463B therefore continues the arbitration of the bus throughout the whole frame. Moreover, if the identifier in transmission has been programmed for reception as well, it transmits and receives messages simultaneously, right up until the Frame Check Sequence (FCS). Only then, if the TSS463B has transmitted the whole message, does it discard the message received. Arbitration loss in the FCS field is considered as a CRC error during transmission.

This feature is called full data field arbitration, and it enables the user to extend the identifier. For instance it can be used to transmit the emitting modules address in the first bytes of the data field, thus enabling the identifier to specify the contents of the frame and the data field to specify the source of the information.

The identifier field of the VAN bus frame is always 12 bits long, and it is always followed immediately by the 4-bit command field:

- The first bit of the command is the extension bit (EXT). This bit is defined by the user on transmission and is received and retained by the TSS463B. To conform with the standard it should be set to 1 (recessive) by the user, else the frame is ignored without any IT generation.
- The second bit is the request ACKnowledge bit (RAK). If this bit is a logical one, the receiving module must acknowledge the transfer with an in-frame acknowledgement in the ACK field. If it is set to logical zero, then the ACK field must contain an acknowledge absent sequence.
- Third we have the Read/Write bit (R/W). This bit indicates the direction of the data in a frame.
  - If set to zero it is a "write" message, i.e. data transmitted by one module to be received by another module.
  - If it is set to one it implies a "read" message, i.e. a request that another module should transmit data to be received by the one that requested the data (reply request message).
- Last in the command field is the Remote Transmission Request bit (RTR). This bit is a logical zero if the frame contains data and a logical one if the frame does not contain data. In order to conform with the standard a received frame included the combination R/W. RTR = 01 is ignored without any IT generation.

All the bits in the command field are automatically handled by the TSS463B, so the user need not to be concerned for the encoding and decoding of these. The command bits transmitted on the VAN bus are calculated from the current status of the active message.

After the command field comes the data field. This is just a sequence of bytes transmitted MSB first. In the VAN standard the maximum message length is set to 28 bytes, but the TSS463B handles messages up to 30 bytes.

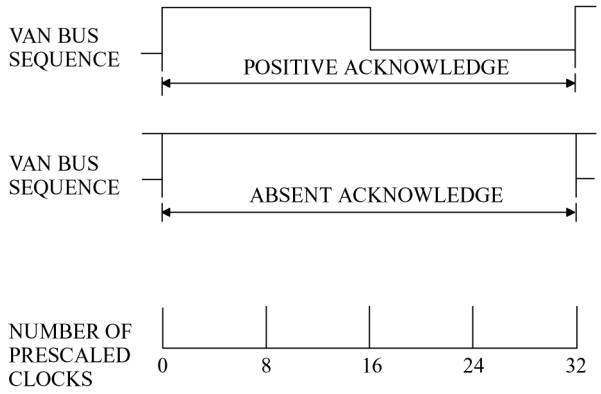
The next field is the FCS field. This field is a 15 bit CRC checksum defined by the following generator polynomial g(x) of order 15:

$g(x)= x^{15}+x^{11}+x^{10}+x^{9}+x^{8}+x^{7}+x^{4}+x^{3}+x^{2}+1$

The division is done with a rest initialized to 0x7FFF, and an inversion of the CRC bits is performed before transmission.

However, since the CRC is calculated automatically from the identifier, command and data fields by the TSS463B, it need not concern the user of the circuit. When the frame check sequence has been transmitted, the transmitting module must transmit an End Of Data (EOD) sequence, followed by the ACKnowledge field (ACK) and the End Of Frame sequence (EOF) to terminate the transfer.

Figure 18. Acknowledge Sequences



Frame Examples

The frames transmitted on the VAN bus are generated by several modules, each supplying different parts of the message. Figure 20. through Figure 23. show the four frame types specified in the VAN standard, and what module is generating the different fields.

- The most straightforward frame is the normal data frame in Figure 20. Like all other frames it is initiated with a SOF sequence. This sequence is generated by a bus master (not shown in figure).

During this frame there is basically only one module transmitting with the only exception being the acknowledgement, generated by the receiving module if requested in the RAK bit.

- The reply request frame with immediate reply in Figure 21. is the only frame in which a slave module can transmit data by filling it into the appropriate field.

The only difference for the frame on the bus is that the R/W bit has changed state compared to the normal frame.

This is a highly interactive frame where a bus master generates the SOF and the initiator generates the identifier, the three first bits of the command, and the acknowledge. The RTR bit, the data field, the frame check, the EOD and the EOF are all generated by the replying module.

- The reply request frame with deferred reply in Figure 22. is basically the same frame as the reply request frame with immediate reply, but since the requested module does not generate the RTR bit the requesting module will continue with the frame check, the EOD and the EOF.

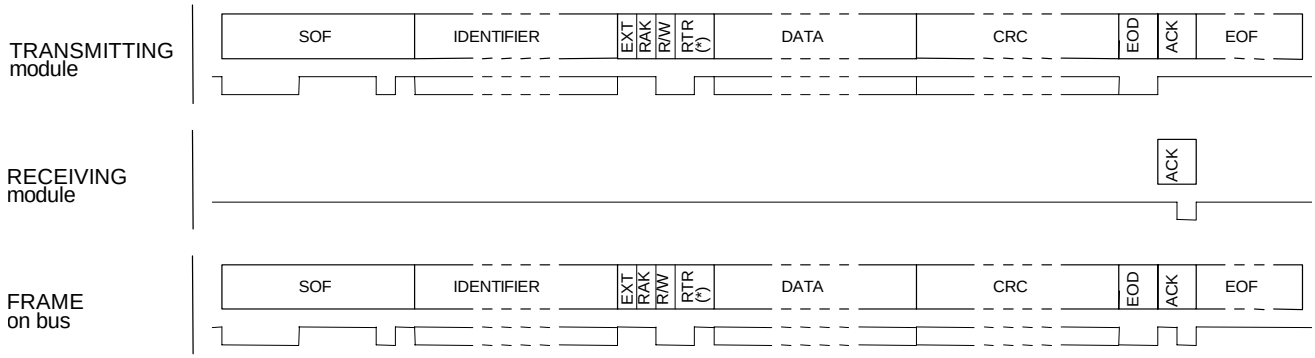
During this frame the requested module will only generate the acknowledge, and only if this was requested by the initiator through the RAK bit.

- Finally the deferred reply frame in Figure 23. which is sent when a module has prepared a reply for a reply request that has been received earlier.

This frame very closely mimics the normal data frame with the only exception being the R/W bit that has changed state.

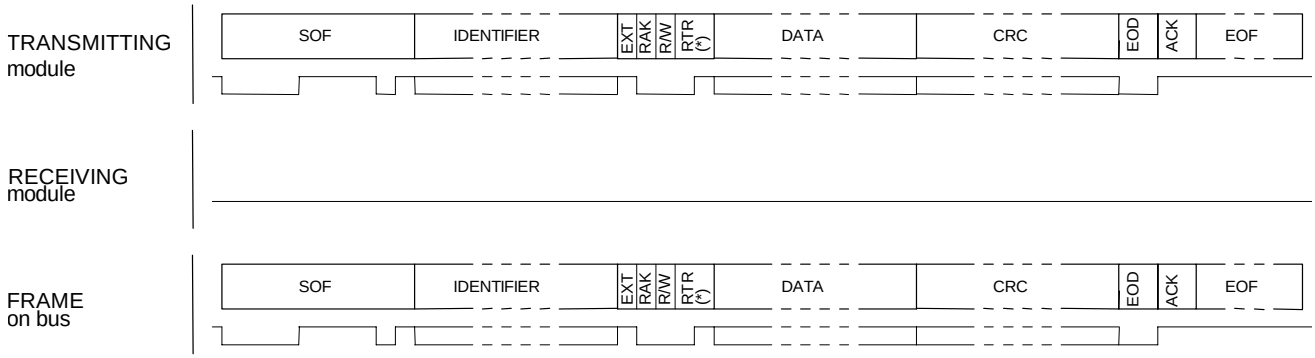
Figure 19. Normal Data Frame

With acknowledgment



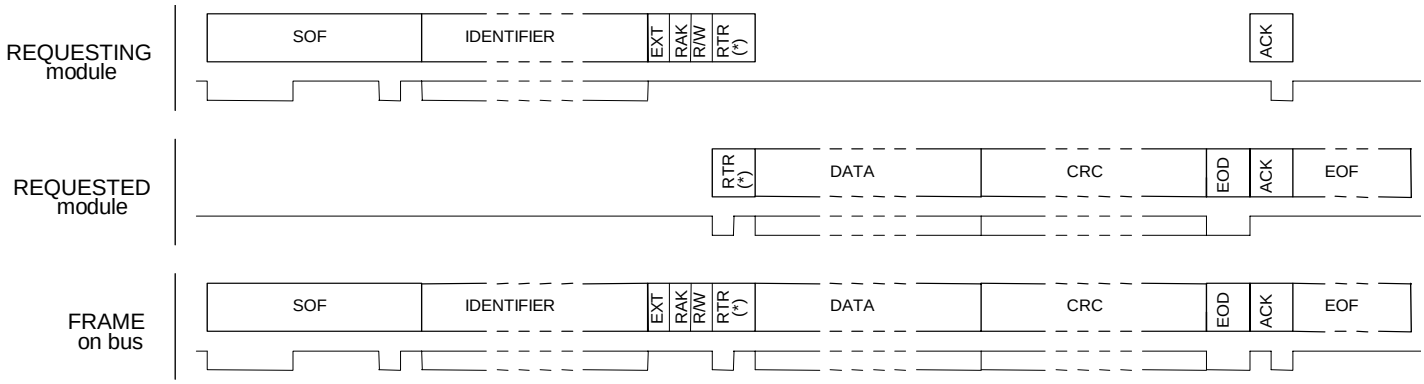
EXT : Recessive from Transmitter
RAK : Recessive for acknowledge from Transmitter
R/W : Dominant from Transmitter
RTR : Dominant from Transmitter- (*) Manchester bit
ACK : Positive from Receiver because RAK is Recessive

Without acknowledgment



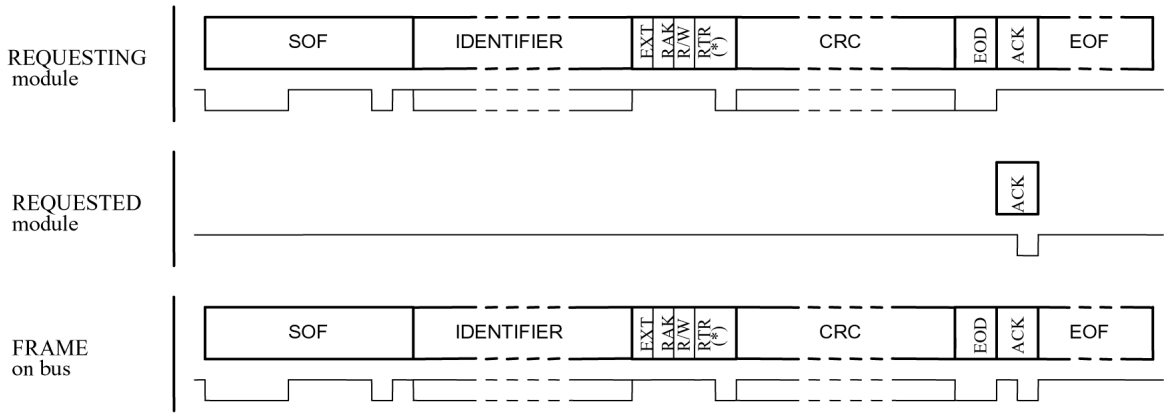
EXT : Recessive from Transmitter
RAK : Dominant for no acknowledge from Transmitter
R/W : Dominant from Transmitter
RTR : Dominant from Transmitter- (*) Manchester bit
ACK : Absent from Transmitter and from Receiver because RAK is Dominant

Figure 20. Reply Request Frame with Immediate Reply



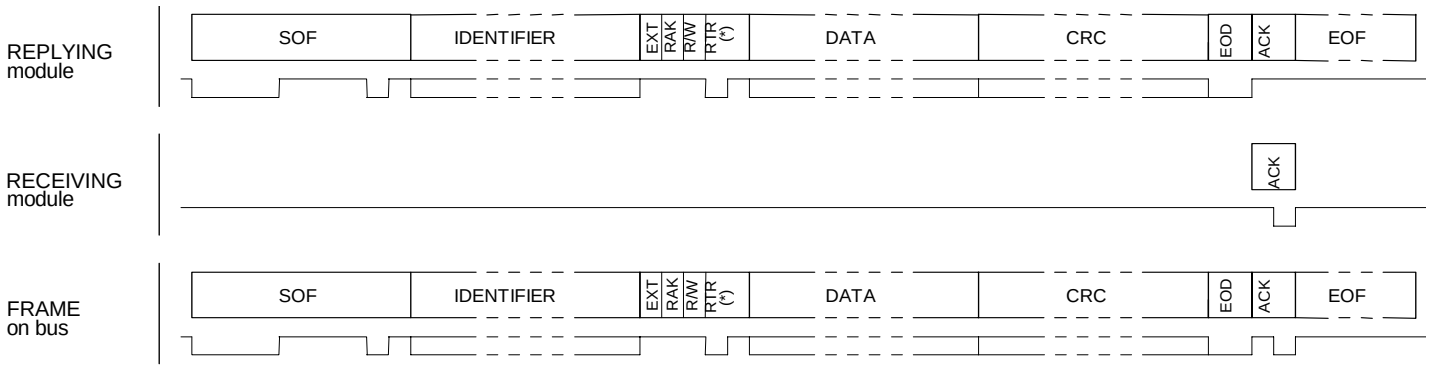
EXT : Recessive from Requestor
RAK : Recessive for acknowledge from Requestor
R/W : Recessive from Requestor
RTR : Recessive from Requestor and Dominant from Requestee - (*) Manchester bit
ACK : Absent from Requestee and Positive from Requestor because RAK is Recessive

Figure 21. Reply Request Frame with Deferred Reply



EXT : Recessive from Requestor
RAK : Recessive for acknowledge from Requestor
R/W : Recessive from Requestor
RTR : Recessive from Requestor – (*) Manchester bit
ACK : Absent from Requestor and Positive from Requestee because RAK is Recessive

Figure 22. Deferred Reply Frame



EXT : Recessive from Replyer
RAK : Recessive for acknowledge from Replyer
R/W : Recessive from Replyer
RTR : Dominant from Replyer - (*) Manchester bit
ACK : Absent from Replyer and Positive from Receiver because RAK is Recessive

Diagnosis System

The purpose of the diagnosis system is to detect any short or open circuits on either the DATA or DATA lines and to permit, if it is possible, to carry the communications on the non-defective line.

The diagnosis system is based on the assumption that three separate line receivers are connected to the VAN bus see Figure 3:

- One of the line receivers is connected in differential mode, sensing both DATA and DATA signals, and is connected to the RxD0 input.
- The other two line receivers are operating in single wire mode and are sensing only one of the two VAN bus signals:
- the line receiver sensing DATA is connected to RxD1,
- the line receiver sensing $\overline{\text{DATA}}$ is connected to RxD2.

The diagnosis system analyses and compares the data sent over both VAN lines. So, the diagnosis system executes a digital filtering and transition analyses. In order to perform its investigation, three internal signals are generated, RI (*Return to Idle)*, SDC (*Synchronous Diagnosis Clock)* and TIP (*Transmission In Progress*).

One of four operating modes can be chosen to manage the results of the diagnosis system.

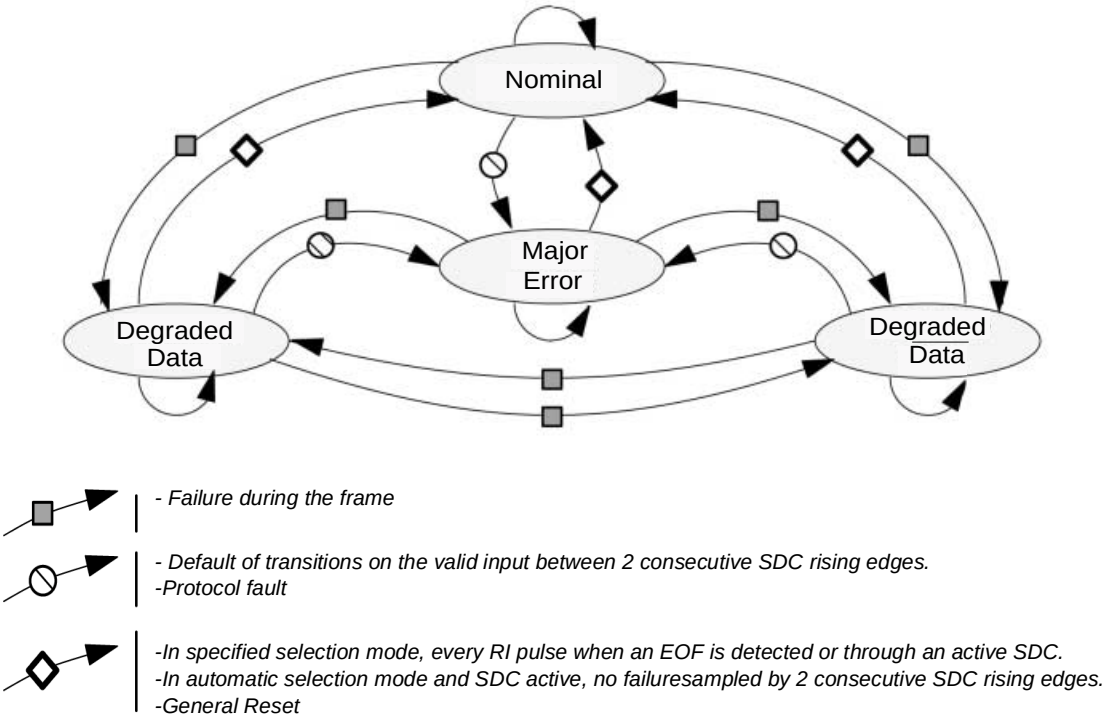
Diagnosis States

If the diagnosis system finds a failure on either of the VAN bus signals, it changes from nominal to degraded mode, and connects the line receiver not coupled to the failing signal to the reception logic.

When the diagnosis system finds that the failing signal is working again, it returns to nominal mode and re-connects the differential line receiver to the reception logic.

A major error occurs when both the VAN bus signals are failed.

Figure 23. Diagnosis States



Status bits give permanent information on the diagnosis performed, whatever the programmed operating mode. This is encoded over three bits: Sa, Sb and Sc.

- Sa and Sb bits indicate the four possible states of the VAN bus.

Table 4. Status bits: Sa & Sb

Sa	Sb		Communication
0	0	Mode	nominal
		Fault	no fault on VAN bus
		Status	differential communication on DATA and $\overline{DATA}$
0	1	Mode	degraded on DATA
		Fault	fault on $\overline{DATA}$
		Status	communication on DATA
1	0	Mode	degraded on $\overline{DATA}$
		Fault	fault on DATA
		Status	communication on $\overline{DATA}$
1	1	Mode	major error
		Fault	fault on DATA and $\overline{DATA}$
		Status	no communication on DATA and $\overline{DATA}$ (attempt to communicate alternatively on DATA then $\overline{DATA}$ every SDC period)

  – Sc: As soon as one of the three inputs (RXD2, RXD1, RXD0) differs from the others in the input comparison analysis performs by the diagnosis system, Sc is set.

The only way to reset this status bit are through the RI signal or a general reset.

Internal Operations

Digital Filtering

If several spurious pulses occur during one bit, the diagnosis for defective conductor may be corrupted. To avoid such errors, digital filters are implemented.
Filtering operation is based on sampling of the comparator output signals. A transition is taken into account only if it is observed over five samples ($1/16^{th}$ of timeslot).

Transition Analyses

These analyses are continuously done on the effective edges on comparators after digital filtering.

- **Asynchronous diagnosis**
  The asynchronous diagnosis is done by comparing the number of edges on DATA and $\overline{DATA}$.
  If four edges are detected on one input and no edges on the other during the same period, the second input is considered faulty and the diagnosis mode will change to one of the degraded modes.
- **Synchronous diagnosis**
  The synchronous diagnosis counts the number of edges on the data input connected to the reception logic during one SDC period.
  If there are less than four edges during one SDC period, the diagnosis mode will change to the major error mode.

- **Transmission diagnosis**
  The transmission compares RxD1 and RxD2 inputs (through the input comparators and the filters) with the data transmitted on TxD output.
  At a time when the transmission logic generates a dominant - recessive transition, the inputs can give different values. Taking into account the filtering delay, the bus line seen as dominant is assumed to be correct, the other one, recessive, is considered faulty. The diagnosis mode is changed to reflect that.
- **Protocol fault**
  The protocol fault is detected by counting the number of consecutive dominant timeslots.
  If eight consecutive timeslots are dominant, the diagnosis mode will change to the major error mode.

Generation of Internal Signals

RI Signal (Return to Idle)

This signal is used to return to nominal mode in the three specified selection modes (see “Diagnosis States” on page 23 and “Programming Modes” on page 26). The RI signal is disabled in automatic selection mode.

The RI signal is a pulse generated when an EOF is detected. Thus, at the end of each frame, the user, regarding the diagnosis status bit Sa, Sb & Sc, can make its own choice.

SDC Signal (Synchronous Diagnosis Clock)

This time base is used by diagnosis system in automatic selection mode (see section Section “Programming Modes”, page 26) when no event is recorded on the bus.

The SDC is generated either by a special SDC divider connected to the timeslot clock, either can be performed manually. The SDC clock period must be long compared to the timeslot duration.

A typical SDC period should be greater than the maximum frame length appearing on the VAN network.

TIP Signal (Transmission In Progress)

This signal must be enabled to allow the transmission diagnosis (see Section “Transition Analyses”, page 24).

The TIP turns on synchronously with the beginning of the transmission:

- for asynchronous bus access, the beginning of SOF,
- for synchronous bus access, the beginning of the identifier field,
- for a request of in frame reply, the RTR bit of the command field.

The TIP turns off synchronously with the end of the transmission:

- after EOF
- after a losing of arbitration or a code violation detection
- for a requestor of in frame reply, when the arbitration is lost on RTR the bit.

This signal is not generated when the transmission logic only sends an ACK.

Programming Modes

Four programming modes determine the way to use the three different inputs and the diagnosis system.

- 3 specified selection modes
- 1 automatic selection mode

Table 5. Programming modes

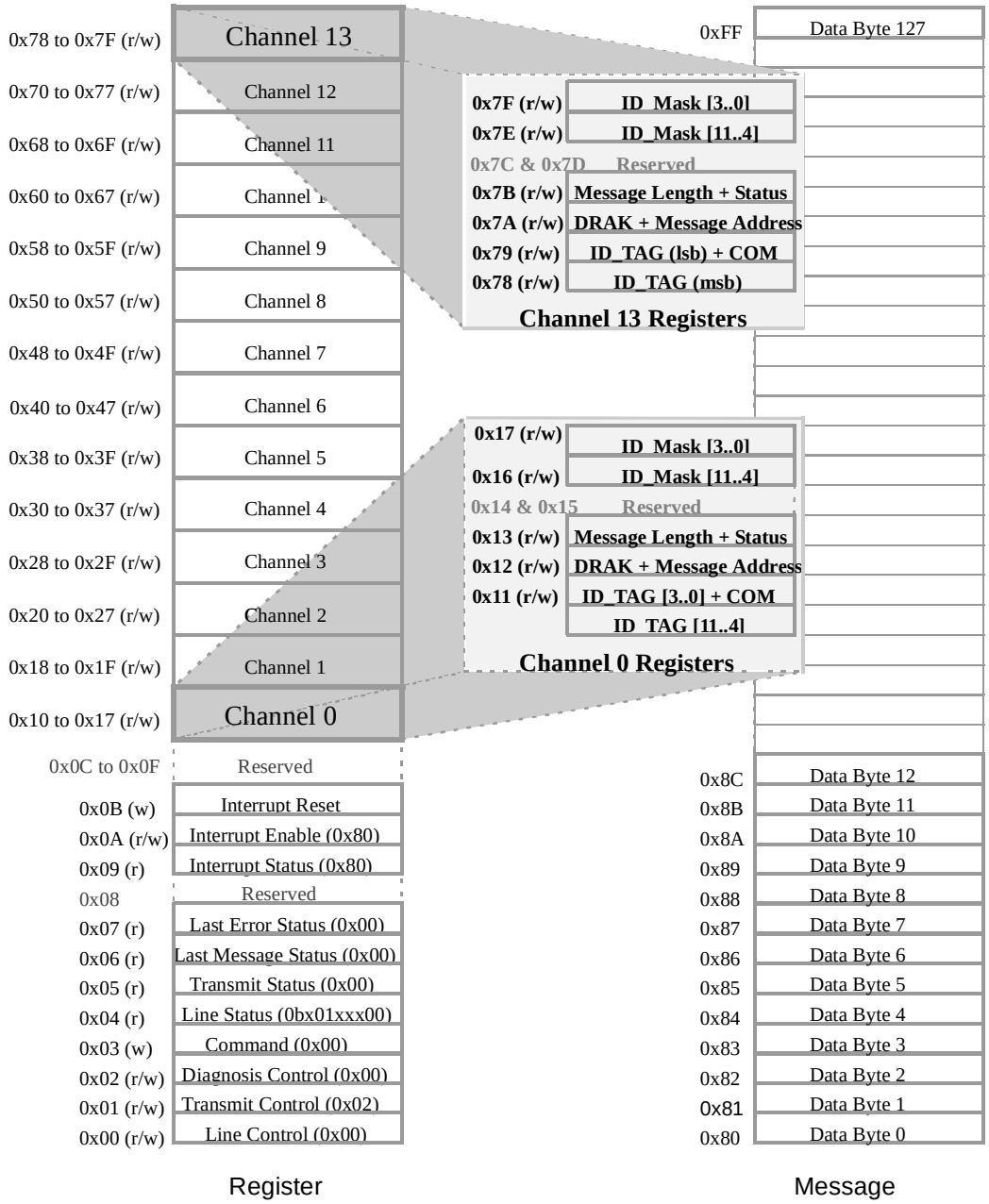
Ma	Mb	Operating Mode
0	0	Differential communication
0	1	Degraded communication on RxD2 ($\overline{\text{DATA}}$)
1	0	Degraded communication on RxD1 (DATA)
1	1	Automatic selection according the diagnosis status

Registers

The TSS463B memory map consists of three different areas, the Control & Status registers, the Channel registers and the Message data (or Mailbox).

Mapping

Figure 24. Memory Map



Notes:
1. All the non specified addresses between 0x00 and 0x7F are considered as absent.
2. (r) means read only register.
   (w) means write only register.
   (r/w) means read/write register.
3. Value after RESET is found after register name. If no value is given, the register is not initialized at RESET.

Control and Status Registers

Line Control Register (0x00)

7	6	5	4	3	2	1	0
CD3	CD2	CD1	CD0	PC	0	IVTX	IVRX

- Read/write register.
- Default value after reset: 0×00
- Reserved: Bit 2, this bit must not be set by the user; a 0 must always be written to this bit.

CD[3:0]:Clock Divider

They control the VAN Bus rate through a Baud Rate generator according to the formula below:

$$f(TSCLK) = \frac{f(XTAL1)}{n \times 16}$$

PC: Pulsed Code

One: The TSS463B will transmit and receive data using the pulsed coding mode (i.e optical or radio link mode). The use of this mode implies communication via the RxD0 input and the non-functionality of the diagnosis system.
Zero: (default at reset) The TSS463B will transmit and receive data using the Enhanced Manchester code. (RxD0, RxD1, RxD2 used).

IVTX: Invert TxD output

IVRX: Invert RxD inputs

The user can invert the logical levels used on either the TxD output or the RxD inputs in order to adapt to different line drivers and receivers.
One: A one on either of these bits will invert the respective signals.
Zero: (default at reset) The TSS463B will set TxD to recessive state in Idle mode and consider the bus free (recessive states on RxD inputs).

Transmit Control Register (0x01)

7	6	5	4	3	2	1	0
MR3	MR2	MR1	MR0	VER2	VER1	VER0	MT

- Read/Write register.
- Default value after reset: 0x02

MR[3:0]: Maximum Retries

These bits allow the user to control the amount of retries the circuit will perform if any errors occurred during transmission.

Table 6. Retries

MR [3:0]	Max. Nb of retries	Max. Nb of transmissions
0000	**0**	**1**
0001	1	2
0010	2	3
0011	3	4
0100	4	5
0101	5	6
0110	6	7
0111	7	8
1000	8	9
1001	9	10
1010	10	11
1011	11	12
1100	12	13
1101	13	14
1110	14	15
1111	15	16

Note: Bus contention is not regarded as an error and that an infinite number of transmission attempts will be performed if bus contention occurs continuously.

VER[2:0] = 001

DLC Version after reset.

These bits must not be set by user. 001 must always be written to these bits.

MT: Module type

The three different module types are supported (see “VAN Frame” on page 16):

One: The TSS463B is at once an autonomous module (Rank 0), a synchronous access module (Rank 1) or a slave module (Rank 16).
Zero: The TSS463B is at once an synchronous access module (Rank 1) or a slave module (Rank 16).

Diagnosis Control Register (0x02):

7	6	5	4	3	2	1	0
SDC3	**SDC2**	**SDC1**	**SDC0**	**Ma**	**Mb**	**ETIP**	**ESDC**

- Read/Write register
- Default value after reset: 0×00.

The diagnosis is discussed in greater detail in the section “Diagnosis System” on page 23 of this chapter.

- In its four high order bits the user can program the SDC rate SDC [3:0],
- In its two medium order bits the diagnosis system mode is controlled: M1, M0.
- In the two low order bits, the user controls if the SDC and TIP are to be generated automatically ETIP, ESDC.

SDC [3:0]: SDC divider

The input clock is the time slot clock.

Table 7. System Diagnosis Clock Divider

SDC DIVIDER SDC [3:0]	Divide by
0000	64
0001	128
0010	256
0011	512
0100	1024
0101	2048
0110	4096
0111	8192
1000	16384
1001	32768
1010	65536
1011	131072
1100	262144
1101	524288
1110	1048576
1111	2097152

SDC calculation: (see “SDC Signal (Synchronous Diagnosis Clock)” on page 26)

Notes: 1. For each module, determine the largest interframe spacing, **LIFS** (*).
2. For the whole network, get the maximum LIFS, **MAX-LIFS**.
3. SDC period > **MAX-LIFS.**

(*) IFS min. = 4 TS

Example: For VAN frame speed rate = 62,5 KTS/s (1 TS= 16 µs), SDC >100 ms

=> 100 ms / 16 µs = 6250, divider chosen: 8192, SDC [3:0] = 0111.

Ma, Mb: Operating Mode Command Bits

Table 8. Diagnosis System Command Bits

Ma	Mb	
0	0	Forces the Communication on RxD0 (differential)
0	1	Forces the Communication on RxD2 ($\overline{\text{DATA}}$)
1	0	Forces the Communication on RxD1 (DATA)
1	1	Automatic selection

ETIP: Enable Transmission In Progress

The Transmission In Progress (TIP), tells the diagnosis system to enable transmission diagnosis.

One: Enable TIP generation
Zero: Disable TIP generation.

ESDC: Enable System Diagnosis Clock

The Synchronous Diagnosis Clock (SDC), controls the cycle time of the synchronous diagnosis.

One: Enable SDC divider.
Zero: Disable SDC divider.

Command Register (0x03):

7	6	5	4	3	2	1	0
GRES	**SLEEP**	**IDLE**	**ACTI**	**REAR**	**0**	**0**	**MSDC**

- Write only register.
- Reserved: Bit 1, 2 these bit must not be set by the user; a zero must always be written to these bit.
- If the circuit is operating at low bitrates there might be a considerable delay between the writing of this register and the performing of the actual command (worst case 6 timeslots). The user is therefore recommended to verify, by reading the Line Status Register (0x04), that the commands have been performed.

GRES: General Reset

The Reset circuit command bit performs, if set, exactly as if the external reset pin was asserted. This command bit has its own auto-reset circuitry.

One: Reset active
Zero: Reset inactive

SLEEP: Sleep command

(see Section “Sleep Command”, page 52).If the user sets the Sleep bit, the circuit will enter sleep mode. When the circuit is in sleep mode, all non-user registers are setup to minimize power consumption. Read/write accesses to the TSS463B via the SPI/SCI interface are impossible, the oscillator is stopped.

To exit from this mode the user must apply either an hardware reset (external $\overline{\text{RESET}}$ pin) either an asynchronous software reset (via the SPI/SCI interface).

One: Sleep active
Zero: Sleep inactive

IDLE: Idle command

(see Section “Idle and Activate Commands”, page 52).If the user sets the Idle bit, the circuit will enter idle mode. In idle mode the oscillator will operate, but the TSS463B will not transmit or receive anything on the bus, and the TxD output will be in three state

One: Idle active
Zero: Idle inactive

ACTI: Activate command

(see Section “Idle and Activate Commands”, page 52). The Activate command will put the circuit in the active mode, i.e it will transmit and receive normally on the bus. When the circuit is in activate mode the TxD three-state output is enabled.

One: Activate active
Zero: Activate inactive

REAR: Re-Arbitrate command.

This command will, after the current attempt, reset the retry counter and re-arbitrate the messages to be transmitted in order to find the highest priority message to transmit.

One: Re-arbitrate active
Zero: Re-arbitrate inactive

MSDC: Manual System Diagnosis Clock.

Rather than using the SDC divider described in section , the user can use the manual SDC command to generate a SDC pulse for the diagnosis system.

This MSDC pulse should be high at least 2 time slot clock.

Line Status Register (0x04):

7	6	5	4	3	2	1	0
X	SPG	IDG	Sc	Sb	Sa	TXG	RXG

- Read only register
- Default value after reset: 0bx01xxx00.
- This register reports the operation mode of the TSS463B in the Sleep an Idle bits (Command Register located at address 0×03) as well as the diagnosis system status bits Sa to Sc discussed in the section “Diagnosis System” on page 23

SPG: Sleeping

IDG: Idling

Default mode at reset

Sa, Sb and Sc: Diagnosis system status bits

- Sa and Sb

Table 9. Diagnosis System Status Bits

Sb	Sa	COMMUNICATION INDICATION
0	0	Nominal mode, differential communication
0	1	Degraded over $\overline{\text{DATA}}$, fault on DATA
1	0	Degraded over DATA, fault on $\overline{\text{DATA}}$
1	1	Major error, fault on DATA and $\overline{\text{DATA}}$

- **Sc:** As soon as one of the three inputs (RxD2, RxD1, RxD0) differs from the others in the input comparison analysis performs by the diagnosis system, Sc is set.
  The only ways to reset this status bit are through the RI signal or a general reset.

TXG: Transmitting

If this status bit is active, it indicates that the TSS463B has chosen an identifier to transmit, and it will continue to make transmission attempt for this message until it succeeds or the retry count is exceeded.

RXG: Receiving

The receiving indicates that there is activity on the bus.
Note: For safe modification of active channel registers both bits should be inactive (except “abort” command).

Transmission Status Register (0x05)

7	6	5	4	3	2	1	0
NRT3	NRT2	NRT1	NRT0	IDT3	IDT2	IDT1	IDT0

- Read only register.
- Default value after reset: 0x00.
- The transmission Status register contains the number of retries made up-to-date, according to the Table 6., and the channel currently in transmission.

NRT [3:0]: Number of retries done in transmission.

IDT [3:0]: Channel number currently in transmission.

Last Message Status Register (0x06)

7	6	5	4	3	2	1	0
NRTR3	NRTR2	NRTR1	NRTR0	IDTR3	IDTR2	IDTR1	IDTR0

- Read only register.
- Default value after reset: 0x00.
- This register is basically the same as the transmission status register. It contains the last identifier number that was successfully transmitted, received or exceeded its retry count.
  If it was a successful transmission, the number of retries performed can be seen in this register as well.

NRTR [3:0]: Number of retries done successfully in transmission. In case of reception NRTR[3:0] is undefined.

IDTR [3:0]: Channel number that was successfully transmitted, received or exceeded its retry count.

Last Error Status Register (0x07)

7	6	5	4	3	2	1	0
X	BOC	BOV	X	FCSE	ACKE	CV	FV

- Read only register.
- Default value after reset: 0×00.
- The Last Error Status Register contains the error code for the last transmission or reception attempt. It is updated after each attempt, i.e. several error codes can be reported during one single transmission (with several retries).

BOC: Buffer occupied

- when one channel configured in “Reply request” mode has its “received” bit set when it attempts to transmit its request.
- BOC with the link capability between two channels sharing the same received buffer, is set when one channel has already set its “received” bit in its “Message length and status Channel register” and a receive is attempt on the other one.

One: BOC active
Zero: BOC inactive

BOV: Buffer overflow

BOV indicates that the buffer length setup in the Channel Status Register was shorter than the number of bytes received plus 1, and thus, some data was lost.

One: BOV active
Zero: BOV inactive

FCSE: Framing Check Sequence Error

FCSE indicates a mismatch between the FCS received and the FCS calculated

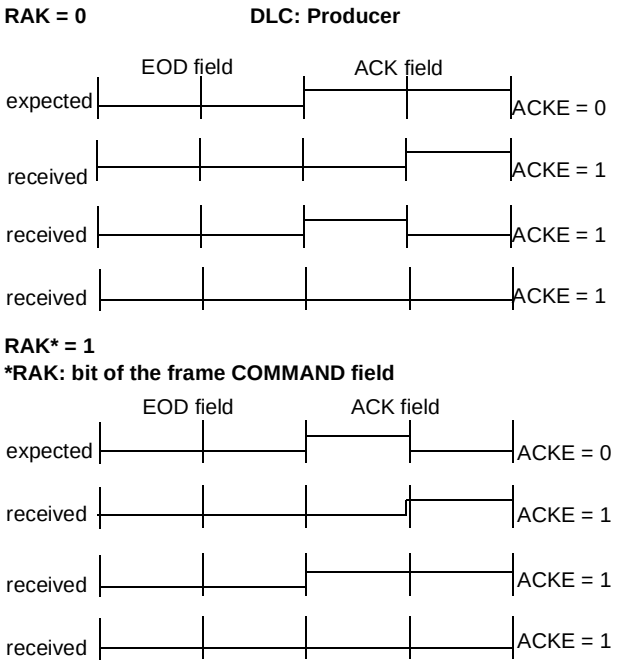
One: FCSE active
Zero: FCSE inactive

ACKE: Acknowledge Error

ACKE indicates a physical violation or collision on ACK field of the frame when the TSS463B is producer.

One: ACKE active
Zero: ACKE inactive

Figure 25. ACKE Status bit



CV: Code Violation

CV indicates:

- either a Manchester code violation (2 identical TS on Manchester bit), or a physical violation (transmitted bit “dominant”, received bit “recessive”), on fields ID, COM, DATA and CRC.
- either a physical violation or collision on field “preamble” and the “recessive” bit of the “Star Sync” field.

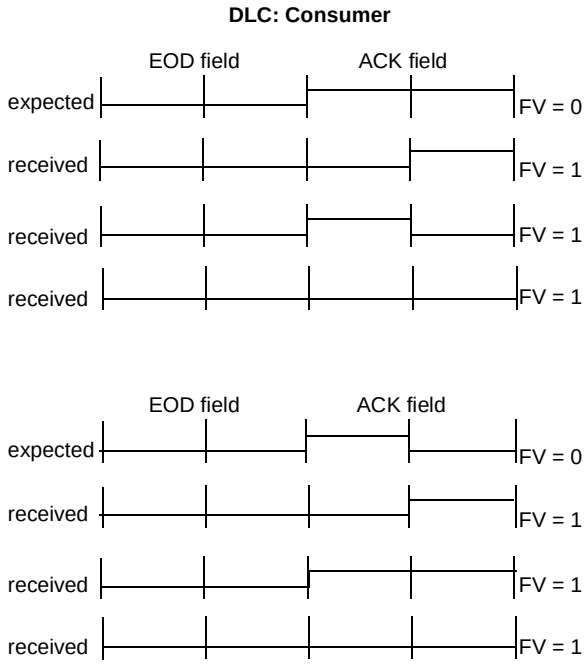
One: CV active
Zero: CV inactive

FV: Frame Violation

FV indicates a physical violation or collision on ACK field of the frame when the TSS463B is consumer.

One: FV active
Zero: FV inactive

Figure 26. FV Status bit



Interrupt Status Register (0x09)

7	6	5	4	3	2	1	0
RST	X	X	TE	TOK	RE	ROK	RNOK

- Read only register.
- Default value after reset: 1xx0 0000

RST: Reset interrupt

RE indicates that the circuit has detected a valid reset command via the $\overline{\text{RESET}}$ pin or the reset command bit GRES. This interrupt cannot be disabled, since its enable bit is set when a reset is detected.

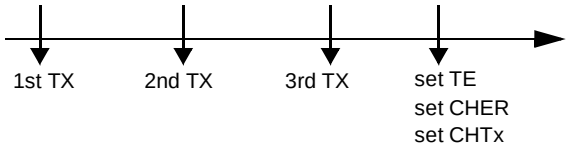
One: Status flag activated
Zero: No status flag.

TE: Transmit error status flag (or exceeded retry)

This flag is set only when the Max number of transmission (1+MR [3:0]) is reached with error of transmission.

One: Status flag activated
Zero: No status flag.

Figure 27. Exceeded retry with MR[3..0] = 3



TOK: Transmit OK status flag

One: Status flag activated
Zero: No status flag.

RE: Receive error status flag
One: Status flag activated
Zero: No status flag.

ROK: Receive "with RAK (RAK=1)" OK status flag
One: Status flag activated
Zero: No status flag.

RNOK: Receive "with no RAK (RAK=0)" OK status flag
One: Status flag activated
Zero: No status flag.

Interrupt Enable Register (0x0A)

7	6	5	4	3	2	1	0
1	X	X	TEE	TOKE	REE	ROKE	RNOKE

- Read/write register.
- Default value reset: 1xx0 0000

Note: On reset the Reset Interrupt Enable bit is set to 1 instead of 0, as is the general rule.

TEE: Transmit Error Enable.
One: IT enabled.
Zero: IT disabled.

TOKE: Transmission OK Enable.
One: IT enabled.
Zero: IT disabled.

REE: Reception Error Enable.
One: IT enabled.
Zero: IT disabled.

ROKE: Reception "with RAK" OK enable.
One: IT enabled.
Zero: IT disabled.

RNOKE: Reception "with no RAK" OK enable.
One: IT enabled.
Zero: IT disabled.

Interrupt Reset Register (0x0B):

7	6	5	4	3	2	1	0
RSTR	0	0	TER	TOKR	RER	ROKR	RNOKR

- Write only register.
- Reserved bit: 5 and 6. This bit must not be set by user; a zero must always be written to this bit.

RSTR: Reset Interrupt Reset.
One: Status flag reset.
Zero: Status flag unchanged.

TER: Transmit Error status flag Reset.
One: Status flag reset.
Zero: Status flag unchanged.

TOKR: Transmit OK status flag Reset.
One: Status flag reset.
Zero: Status flag unchanged.

RER: Receive Error status flag Reset.
One: Status flag reset.
Zero: Status flag unchanged.

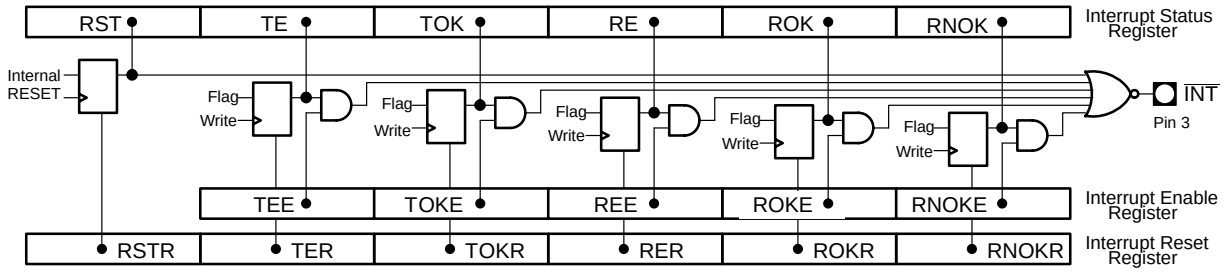
ROKR: Receive “with RAK” OK status flag Reset.

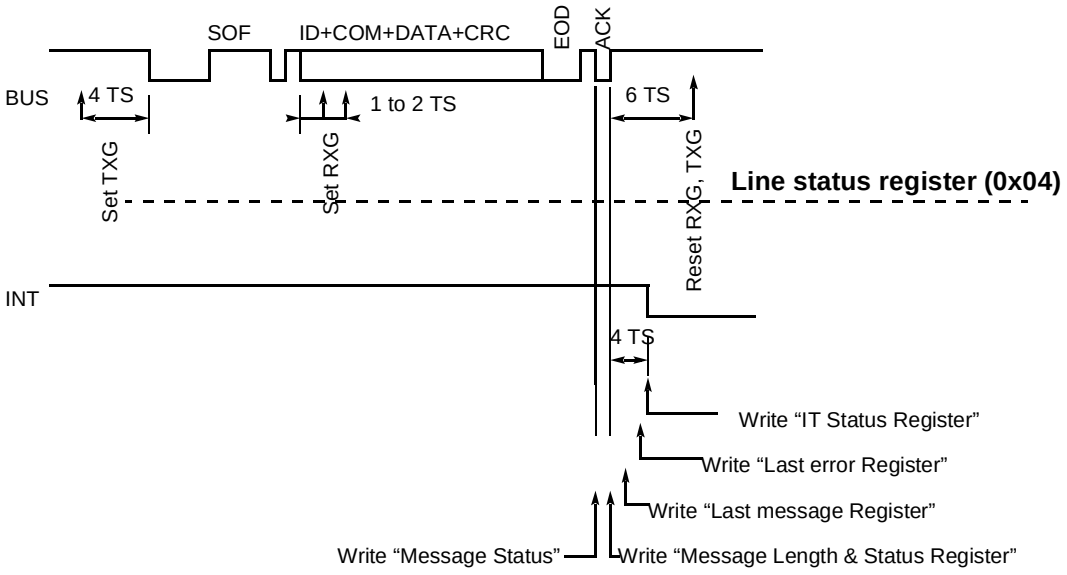
One: Status flag reset.
Zero: Status flag unchanged.

RNOKR: Receive “with no RAK” OK status flag Reset.

One: Status flag reset.
Zero: Status flag unchanged.

Figure 28. Update of the Status Register





Channel Registers

There is a total of 14 channel register sets, each occupying 8 bytes for addressing simplicity, integrated into the circuit. Each set contains two 2x8-bit registers for the identifier tag, identifier mask and command fields plus two 1x8-bit registers for DMA pointers and message status.
The base_address of each set is:

(0x10 + (0x08 * channel_number)).

When the TSS463B is reset either via the external $\overline{\text{RESET}}$ pin or the general reset command, the channel registers are not affected. That is, on power-up of the circuit, all the channel registers start with random values.
Due to this fact, the user should take care to initialize all the channel registers before exiting from idle mode. The easiest way to disable an channel register is to set the received and transmitted bits to 1 in the Message Length & Status Register.

Table 10. Channel Register Sets Map

Channel Number	From	To	Channel Number	From	To
6	0x40	0x47	13	0x78	0x7F
5	0x38	0x3F	12	0x70	0x77
4	0x30	0x37	11	0x68	0x6F
3	0x28	0x2F	10	0x60	0x67
2	0x20	0x27	9	0x58	0x5F
1	0x18	0x1F	8	0x50	0x57
0	0x10	0x17	7	0x48	0x4F

Table 11. Channel Register Set Structure

Reg. Name	Offset	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ID_MASK	0x07	ID_M [3:0]				x	x	x	x
ID_MASK	0x06	ID_M [11:4]							
(no register)	0x05	x	x	x	x	x	x	x	x
(no register)	0x04	x	x	x	x	x	x	x	x
MESS_L / STA	0x03	M_L [4:0]					CHER	CHTx	CHRx
MESS_PTR	0x02	DRACK	M_P [6:0]						
ID_TAG / CMD	0x01	ID_T [3: 0]				EXT	RAK	RNW	RTR
ID_TAG	0x00	ID_T [11:4]							

Identifier Tag and Command Registers

The identifier tag and command registers is located at the base_address and base_address + 1. It allows the user to specify the full 12-bit identifier field of the ISO standard and the 4-bit command.

7	6	5	4	3	2	1	0	
ID_T 3	ID_T 2	ID_T 1	ID_T 0	EXT	RAK	RNW	RTR	base_address + 0x01

7	6	5	4	3	2	1	0	
ID_T 11	ID_T 10	ID_T 9	ID_T 8	ID_T 7	ID_T 6	ID_T 5	ID_T 4	base_address + 0x00

- Read / Write registers.

ID_T [11:0]: Identifier Tag.

Upon a reception hit (i.e, a good comparison between the identifier received and an identifier specified, taking the comparison mask into account, as well as a status and command indicating a message to be received), the identifier tag bits value will be rewritten with the identifier bits actually received.

EXT, RAK, RNW & RTR:

(see Section “Message Types”, page 45).No comparison will be done on the command bits, excepted on EXT bit. The RAK, RNW and RTR bits will be written into the first byte of the Message upon a reception hit.
The RNW and RTR bits, as well as the status bits in the length and status register, must be in a valid position for reception or transmission. If not, the message corresponding to this identifier is considered as inactive or invalid.
The way of knowing if an acknowledge sequence was requested or not is to check the first byte of the Message.

Message Pointer Register

The message pointer register at address (base_address + 0x02) is 8 bits wide. It indicates where in the Message DATA RAM area the message buffer is located.

7	6	5	4	3	2	1	0	
DRAK	M_P 6	M_P 5	M_P 4	M_P 3	M_P 2	M_P 1	M_P 0	base_address + 0x02

- Read / Write register.

DRAK: Disable RAK (used in ‘spy mode’)

In reception: whatever is the RAK bit of the incoming valid frame, no ACK answer will be set. If the message was successfully received, an IT is set (ROK or RNOK).

In transmission: no action.
One: disable active, ‘spy mode’.
Zero: disable inactive, normal operation.

M_P [6:0]: Message pointer

Since the Message DATA RAM area base address is 0x80, the value in this register is the offset from that address. If the message buffer length value is illegal (i.e. zero), this register is redefined as being a link pointer, thus containing the channel number of the channel that contains the actual message pointer, message length and received status. However, the identifier, mask, error and transmitted status used will be that of the originally matched channel. In any case, if a link is intended, the three high bits of M_P [6:0] should be set to 0.
This allows several channels to use the same actual reception buffer in Message DATA RAM, thus diminishing the memory usage.

Note: Only 1 level of link is supported.

Message Length And Status Register

The message length and status register at address (base_address + 0x03) is also 8 bits wide. It indicates the length of reserved for the message in the Message DATA RAM area.

7	6	5	4	3	2	1	0	
M_L 4	M_L 3	M_L 2	M_L 1	M_L 0	CHER	CHTx	CHRx	base_address + 0x03

- Read / Write register.

M_L [4:0]: Message Length

The 5 high bits of this register allows the user to specify either the length of the message to be transmitted, or the maximum length of a message receivable in the pointed reception buffer.
Note: The first byte in this register does not contain data, but the length of the message received. This implies that the length value has to be equal to or greater than the maximum length of a message to be received in this buffer (or the length of a message to be transmitted) plus 1, thus allowing a maximum length of 30 bytes and a minimum length of 0 byte.
If the value of this field is “illegal” (i.e 0x00) then this message pointer is defined as being a link (see Message pointer & register and “Linked Channels” on page 53).

M_L [4:0] = 0x00	Linked channel
M_L [4:0] = 0x01	Frame with no DATA field (*)
M_L [4:0] = 0x02	Frame with 1 DATA byte
- - - - - - -	- -
M_L [4:0] = 0x1D	Frame with 28 DATA bytes
M_L [4:0] = 0x1E	Frame with 29 DATA bytes
M_L [4:0] = 0x1F	Frame with 30 DATA bytes
(*) Different of a reply request frame with no in-frame reply (deferred reply).	

CHER: Channel error status and abort command.

As status, this bit is set by the TSS463B when error occurs in transmission or on a received frame. The user must reset it.
To abort the transmission defined in the channel, this bit can bit set to 1 by the user (see Section “Activate, Idle and Sleep Modes”, page 52 and “Abort” on page 50)

CHTx: Channel transmitted and transmit enable command.

CHRx: Channel received and receive enable command.

The 2 low order bits of this register contains the message status. Together with the RNW and RTR bits of the command register (base_address + 0x01), they define the message type of this channel (see section “Message Types” on page 45). As a general rule (see section “Message Types” on page 45), the status bits are only set by the TSS463B, so the user must reset them to perform a transmission (CHTx) or/and a reception (CHRx). The received and transmitted bits are only set if the corresponding frame is without errors or if the retry count has been exceeded.

Identifier Mask Registers

The Identifier Mask registers (base_address + 0x06 and base_address + 0x07) allow bitwise masking of the comparison between the identifier received and the identifier specified.

7	6	5	4	3	2	1	0	
ID_M 3	ID_M 2	ID_M 1	ID_M 0	x	x	x	x	base_address + 0x07
7	6	5	4	3	2	1	0	
ID_M 11	ID_M 10	ID_M 9	ID_M 8	ID_M 7	ID_M 6	ID_M 5	ID_M 4	base_address + 0x06

- Read / Write registers.

ID_M [11:0]: Identifier Mask

A value of 1 indicates comparison enabled.
A value of 0 indicates comparison disabled.

Example:

- ID_M[11:0] = 0x0FF8
- Acceptance: ID's from 0x0FF8 up to 0x0FFF

Mailbox

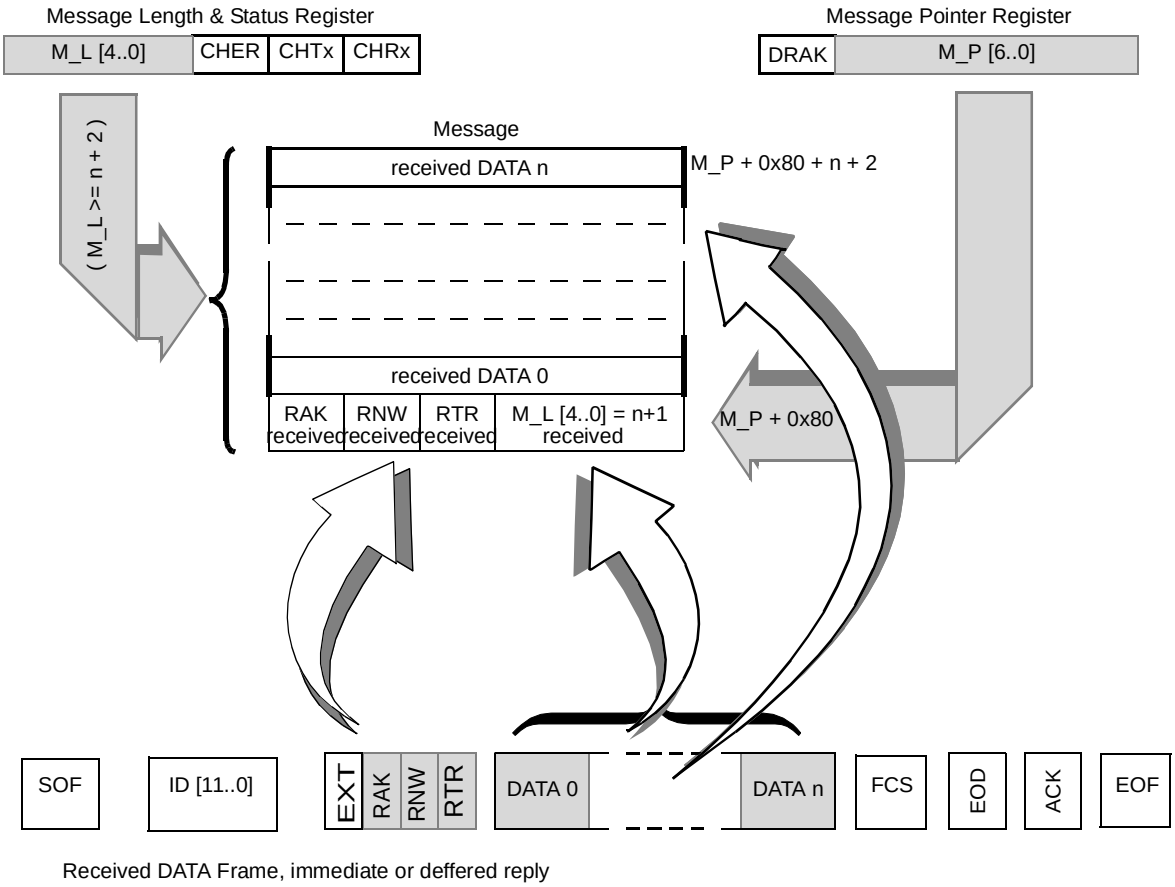
The mailbox contents all the messages received or to be transmitted. Each messages is link to a channel. The Mailbox RAM area has 128 bytes and is mapped from 0x80 to 0xFF (see Section “Mapping”, page 27).

The message (or message buffer) is composed of:

- 1 byte of message status (only used in receiving),
- n bytes of data. These data are the bytes of the DATA field of the frame with the same organization.

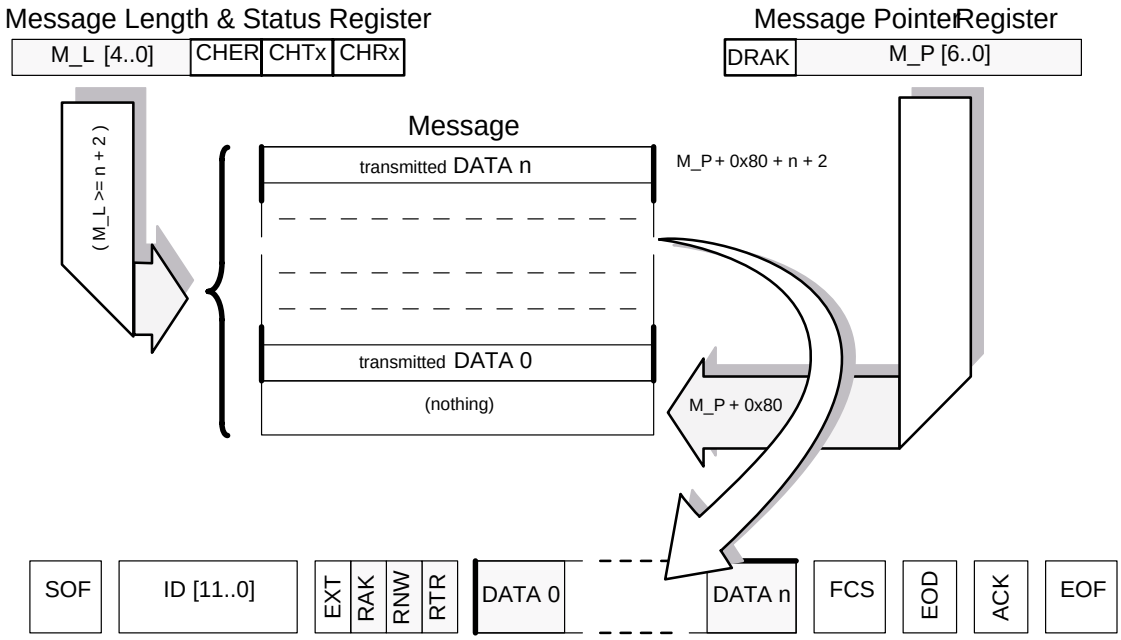
The message is pointed by the Message Pointer Register of the channel, the length of the message is given by the Message Length & Status Register of the channel (sections and). This area is a pure RAM, it contents a random value after reset.

Figure 29. Message Buffer Structure for Reception



Received DATA Frame, immediate or deffered reply

Figure 30. Message Buffer Structure for Transmission



Transmitted DATA Frame

Message Status (pointed by: Message Pointer Register)

7	6	5	4	3	2	1	0
RRAK	RRNW	RRTR	RM_L4	RM_L3	RM_L2	RM_L1	RM_L0

- (no significant value in case of message to be transmitted)

RRAK: Received RAK bit — This bit is the RAK bit coming from the COM field of the received frame.

RRNW: Received RNW bit — This bit is the RNW bit coming from the COM field of the received frame.

RRTR: Received RTR bit — This bit is the RTR bit coming from the COM field of the received frame.

RM_L[4:0]: Message length of the received frame — If the DATA field of the received frame included DATA0 to DATAn, RM_L[4:0] = n+1, even if the reserved length (Message Length & Status Register) is larger.

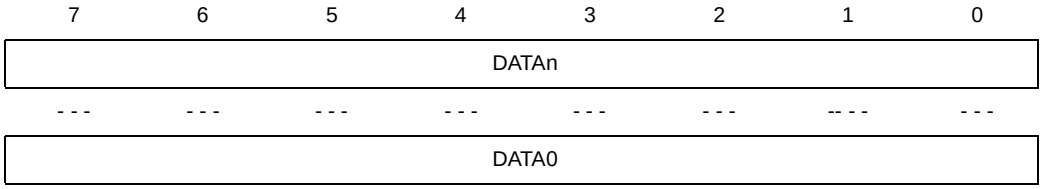
Figure 31. Message Status Updating

Frame Type	Node x	Commu-nication	Node A	Message Status on Node A after IT(*)			
Data Frame	I, P		C	RAK	RNW	RTR	length
Immediate Reply	I, C		P	RAK	RNW	RTR	previous value
Deferred Reply	I, C		P	RAK	RNW	RTR	previous value
Data Frame	C		I, P	previous values			
Immediate Reply	P		I, C	RAK	RNW	RTR	length
Deferred Reply	P		I, C	RAK	RNW	RTR	length

P: Producer I: Initiator C: Consumer

(*) After IT ROK or RNOK. In case of IT RE, the values can be erroneous.

Message Data (string pointed by: Message Pointer Register + 1)



DATA0 is the first received (or transmitted) byte, **DATAn** is the last one.

Note: 1. If the length reserved (in the message length & status register) for an incoming frame is 2 bytes greater or more, the TSS463B will write the 2 bytes of the CRC field in the message string just after DATAn.
Because the VAN frame does not content a message length, the only way for the component to know the length of the DATA field is either the message length register value, either the EOD field detection. When the reserved length is too large, at the moment when it detects the EOD, the TSS463B has already written the 2 bytes of the CRC field, considering these bytes as normal DATA.

2. The Mailbox RAM area is a circular buffer. The next location after 0xFF is 0x80.

Message Types

There are 5 basic message types defined in the TSS463B. Two of them (transmit and receive message types) correspond to the normal frame, and the rest correspond to the different versions of reply frames.

Transmit Message				
	RNW	RTR	CHTx	CHRx
Initial setup	0	0	0	Don't care
After transmission	0	0	1	Unchanged

To transmit a normal data frame on the VAN bus, the user must program an identifier as a Transmit Message. The TSS463B will then transmit this message on the bus until it has succeeded or the retry count is exceeded.

Receive Message				
	RNW	RTR	CHTx	CHRx
Initial setup	0	1	Don't care	0
After transmission	0	1	Unchanged	1

The opposite of the transmit message type is the Receive Message type. This message type will not generate any frames on the bus. Instead it will listen to the bus until a frame passes that matches its identifier, with the mask taken into account, and then receive the data in that frame.

The data received will be stored in the message buffer and the length of the message received is stored in the first byte of the message buffer.

The actual identifier received is stored in the identifier register itself. This identifier may differ from the identifier specified in the register due to the effect of the mask register.

Normally this should not interfere with the next identifier comparison since the bits that may differ are masked via the mask register.

Reply Request Message				
	RNW	RTR	CHTx	CHRx
Initial setup	1	1	0	0
After transmission (Waiting for reply)	1	1	1	0
After reception (of reply)	1	1	1	1

The Reply Request Message type is a demand to transmit on the VAN bus a reply request. When this message type is programmed, three things can happen.

In the first case no other modules on the bus responded with an in-frame reply, and in this case the TSS463B will set the message type to the after transmission state. When this message type is programmed, the TSS463B will listen on the bus for a deferred reply frame matching this identifier, without transmitting the reply request.

The second case is that another module on the bus replies with an in-frame reply. In this case the message type will pass immediately into the after reception state, without passing the after transmission state.

Reply Request Message without transmission				
	RNW	RTR	CHTx	CHRx
Initial setup	1	1	Don't care	0
After reception	1	1	Unchanged	1

In the third case the TSS463B has not yet started to transmit the reply request, when another module either requests a reply, and gets it, or transmits a deferred reply. Warning! This should be avoided as it may result in an illegal message type (Illegal reply Request).

Immediate Reply Message				
	RNW	RTR	CHTx	CHRx
Initial setup	1	0	0	0
After transmission	1	0	1	1

The immediate Reply Message will attempt to transmit an in-frame reply, using the data in the message buffer.

Deferred Reply Message				
	RNW	RTR	CHTx	CHRx
Initial setup	1	0	0	1
After reception (of reply request)	1	0	1	1

Above a Deferred Reply Message is shown. This message type will immediately transmit a deferred reply frame.

Reply Request Detection Message				
	RNW	RTR	CHTx	CHRx
Initial setup	1	0	1	0
After reception	1	0	1	1

Finally there is the Reply Request Detector Message type. Its purpose is to receive a reply request frame and notify the processor, without transmitting an in-frame reply.

Inactive Message				
	RNW	RTR	CHTx	CHRx
Recommended	Don't care	Don't care	1	1
After transmission	0	0	1	Don't care
After reception	0	1	Don't care	1
Illegal reply request	1	1	0	1

The table above shows all inactive messages types. The last combination will transmit a reply request, but will not receive the reply since its buffer is tagged as occupied.

Priority Among the Different Channels

The priority handling on the VAN bus itself is already explained in the Line interface section. The priorities for the messages in the TSS463B is however slightly different.

For instance it's possible that an identifier matches two or more of the identifiers programmed into the registers. In this case, it is the lowest identifier number that has priority. i.e. if both identifier 5 and 10 match the identifier received, it is the identifier 5 that will receive the message.

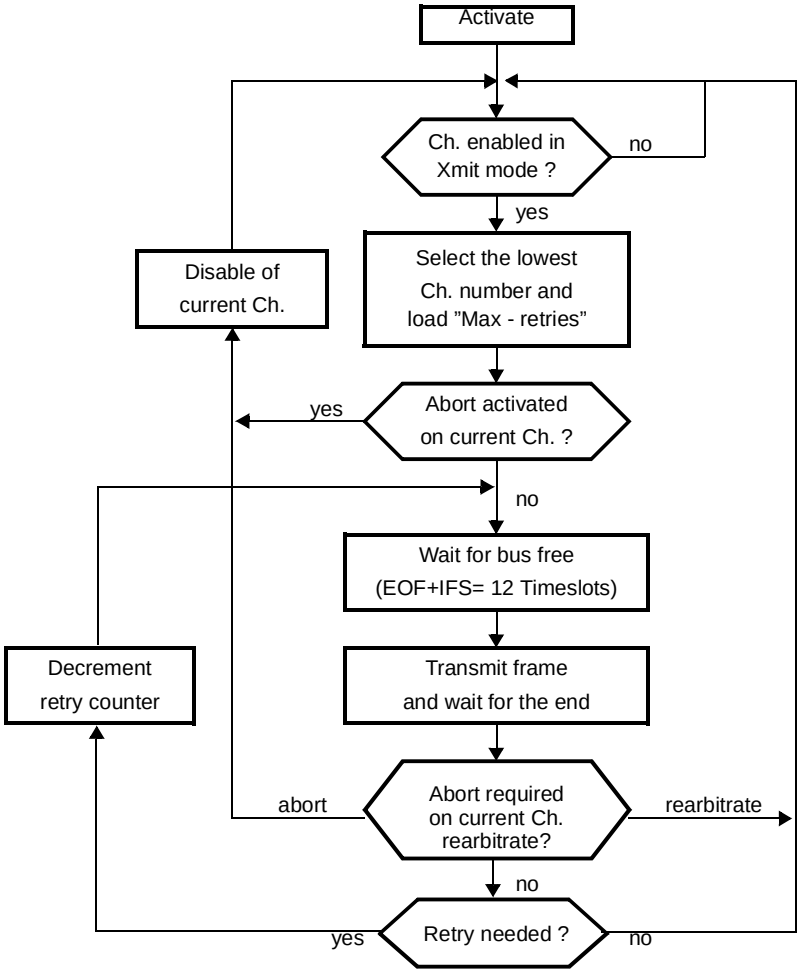
However, since the identifier 5 will become an inactive message when it has received the frame, the next time the same identifier is seen on the bus, the corresponding data will be received by identifier 10.

The same is valid for messages to be transmitted, i.e. if two or more messages are ready to be transmitted, it is the one with the lowest identifier number that will get priority.

Retries, Rearbitrate and Abort

Retries and rearbitrate commands are located, respectively, in the Transmit Control Register and in the Command Register. An abort command is located in each channel register set, in the Message Length & Status Register (base_address + 0x03). These three commands are available only when the TSS463B is producer.

Figure 32. Transmit Function



Retries

The purpose of the retries feature is to provide, for the user, the capability of retrying a transmit request in case of failure, when a node tries to reach another node, either on normal DATA frame or on REPLY REQUEST frame.

The maximum number of retries is programmable through MR[3:0] of the Transmit Control Register (0x01). When a channel is enable - bit CHTx= 0 of Message Length & Status Register, a 4-bit counter is loaded with MR[3:0]. At each attempt, this counter will be count-down. To 0, an IT TE is set in the Interrupt Status Register (0x09), and the transmission is stopped.

MR[3:0]=1 indicates 1 retry, hence 2 transmission attempts will be performed (see Table 4, “Status bits: Sa & Sb,” on page 24). The number of retries performed, as well as the current channel number associated, can be read in the Transmission Status Register (0x05).

The Last Error Status Register (0x07) informs about the trouble uncounted:

- Failure cases:
  - Code viol (CV error bit)
  - Acknowledge error (ACKE error bit)
  - CRC error (FCSE error bit)
- It should be noticed that contention is considered as normal CSMA/CD protocol and, therefore, is not taken into account in failure cases. So, an 'infinite' number of attempts can be performed if bus contention occurs continuously.

There is only one retries counter for all channels. When the user writes the Max_Retries value, all channels start their transmission with this parameter.

Rearbitrate

The purpose of rearbitrate feature is to postpone a channel already in transmission in order to authorize an higher priority (see Section "Priority Among the Different Channels", page 47) message to be transmit.

Typical example

- Max_retries = 1 (2 transmissions attempts).
- If Ch 8 is in a the retry loop and the user wants to transmit the Ch 5 without waiting the end of the loop, the user can use the rearbitrate command.
- The TSS463B will then wait until the end of the current transmission, reload the retries counter and enable the Ch 5 to transmit.
- At the end of this transmission Ch5, either when the attempt is successful or the exceeded retry count is reached, the retries counter is reloaded and the transmission is activated for the Ch 8 again.

Figure 33. Rearbritrate Example

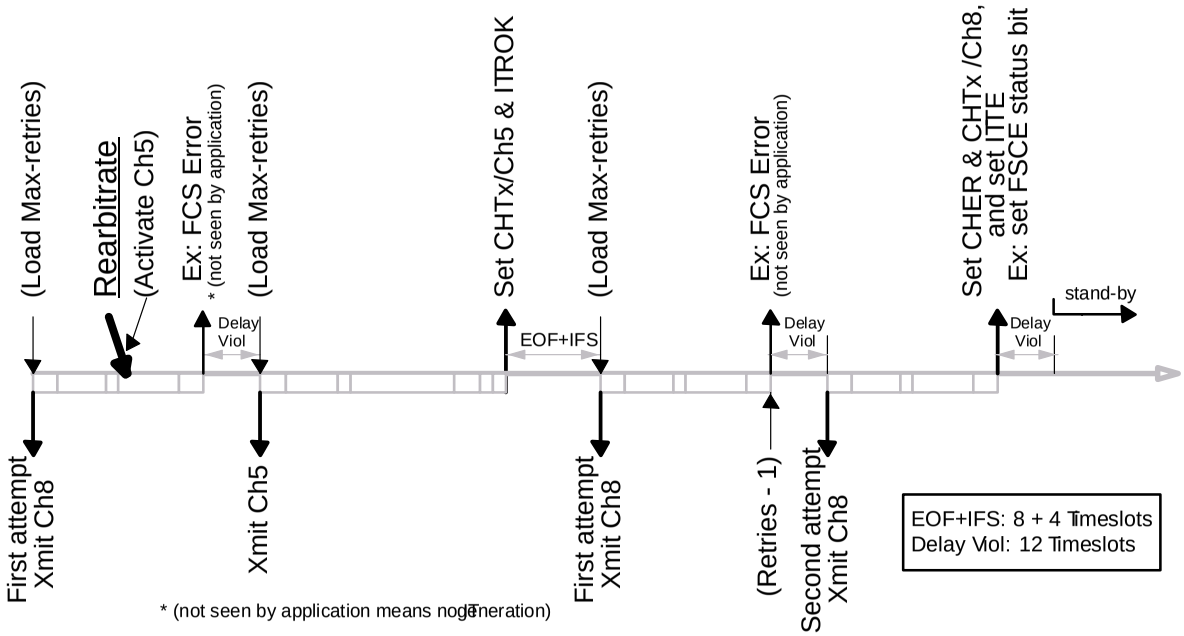
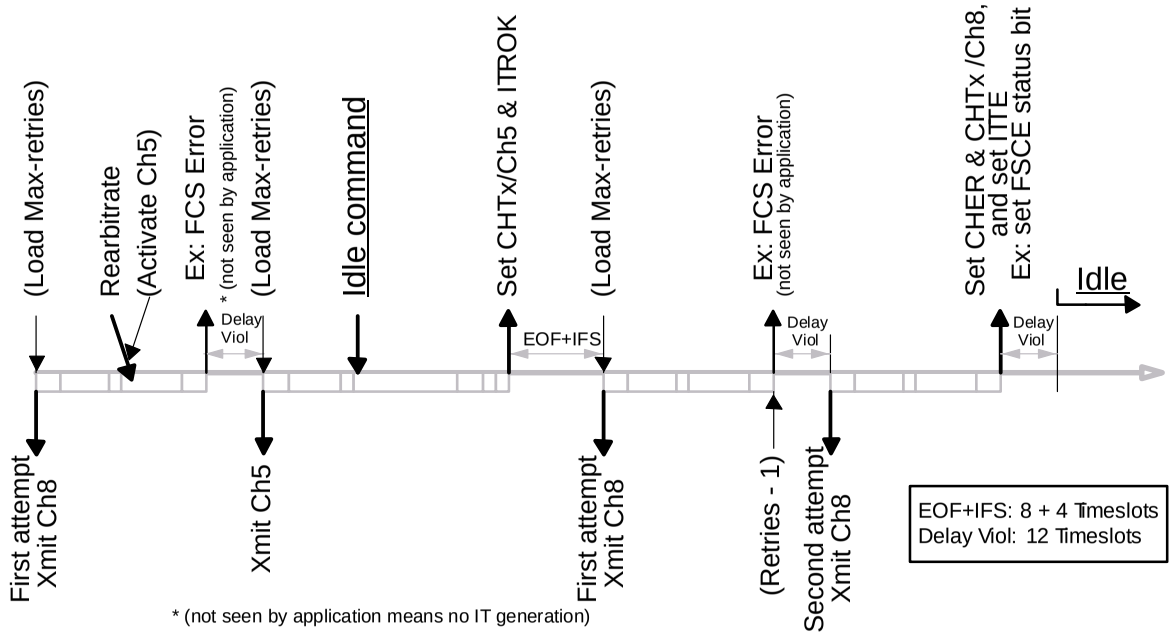
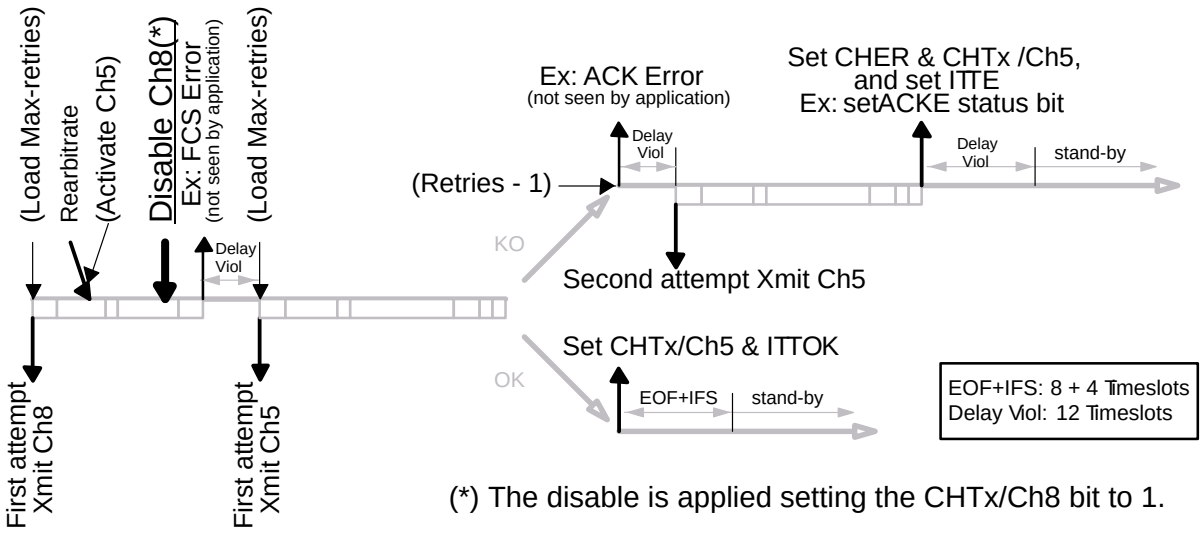


Figure 34. Idle and Rearbitrate Example



If the user sets the idle bit anywhere (after rearbitrate), the idle mode is entered only at the end of all the transmit attempts (for more information about idle command, see “Activate, Idle and Sleep Modes” on page 52.).

Figure 35. Disable Channel After Rearbitrate



(*) The disable is applied setting the CHTx/Ch8 bit to 1.

In this case, the TSS463B completes the current attempt (Ch8) and let the transmission go on the new channel (Ch5 if validated), otherwise it stops all attempts on the current channel.

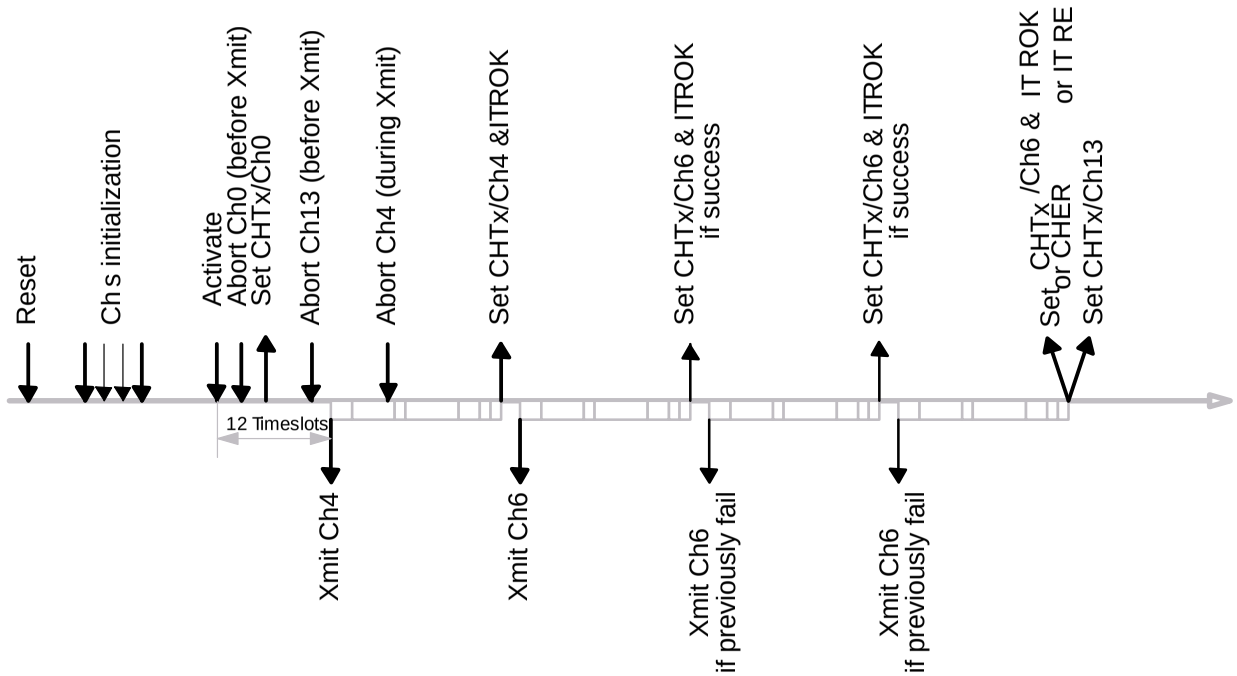
Abort

An abort command is dedicated to channels already enabled in transmission or in in-frame response. For example, this command can be used to break the retry procedure on one channel.

Abort channel is done by setting the Error bit (CHER) in the Message Length & Status Register (base_address + 0x02). This command is taken into account if the channel aborted is not transmitted. When this abort command is really done, the TSS463B set to 1 the Transmitted bit (CHTx) of the Message Length & Status Register.

The abort mechanism is integrated into the transmit function. This mainly means, abort, priority and retries live together in the transmit function.

Figure 36. Abort Example



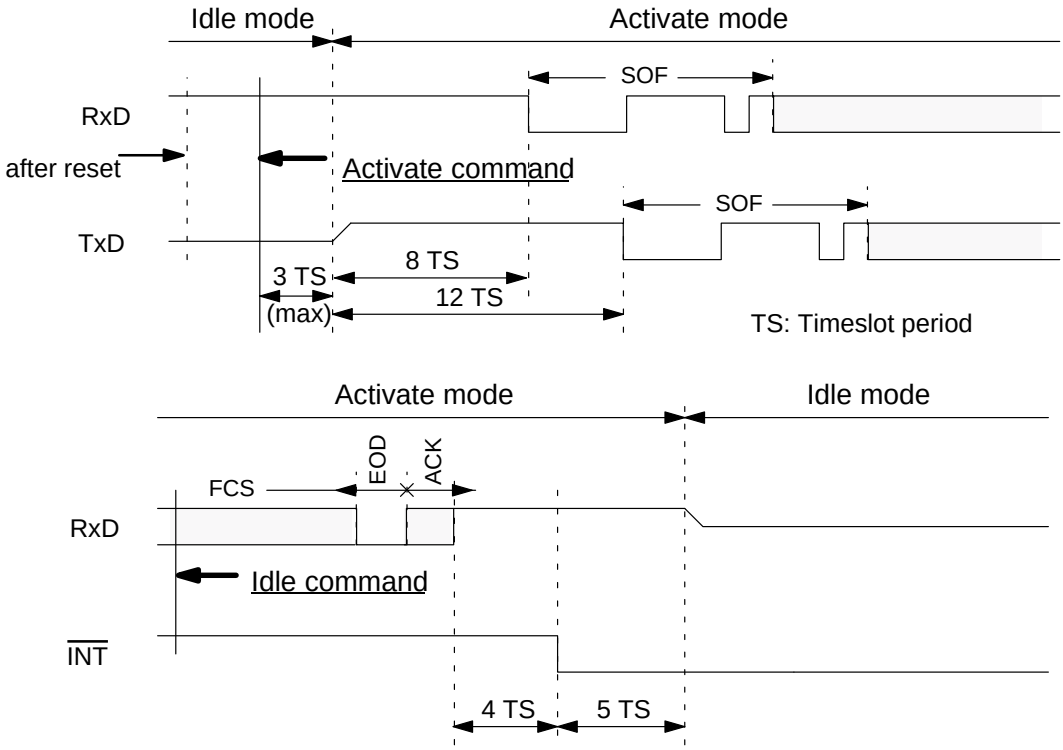
Activate, Idle and Sleep Modes

Sleep, idle and activate commands are located in the Command Register (0x03). These three commands are general commands for the TSS463B.

Idle and Activate Commands

After reset, the TSS463B starts in idle mode. In this mode, the oscillator operates (CKOUT pin active) but the circuit cannot transmit or receive anything on the VAN bus. The TxD output (pin 12) is in three state mode, a pull-up resistor must be provided externally or by the line driver to avoid floating state on the VAN bus. To activate the TSS463B, the user must set the activate bit (ACTI) and reset the idle bit (IDLE).

Figure 37. Idle and Activate Timings



In both cases, the idle state can be verified reading the Line Status register (0x04).

Sleep Command

If the user sets the sleep bit (SLEEP), the TSS463B enters in sleep mode, regardless of the values of activate and idle bits. It means that, all non-user registers are set-up to reduce the power consumption and the internal oscillator is immediately stopped. Then, accesses to all registers (and to the messages) via the SPI/SCI interface are impossible and CKOUT is not provided.

To exit from this mode the user must apply either an hardware reset (external reset pin) either an asynchronous software reset (via the SPI/SCI interface).

In an application (i.e. typical application figure 8) using the CKOUT feature (pin 8), if the TSS463B is put in sleep mode, the clock provided to the microcontroller is stopped. So, the system does not run and the only way to awake this application is an external reset.

Linked Channels

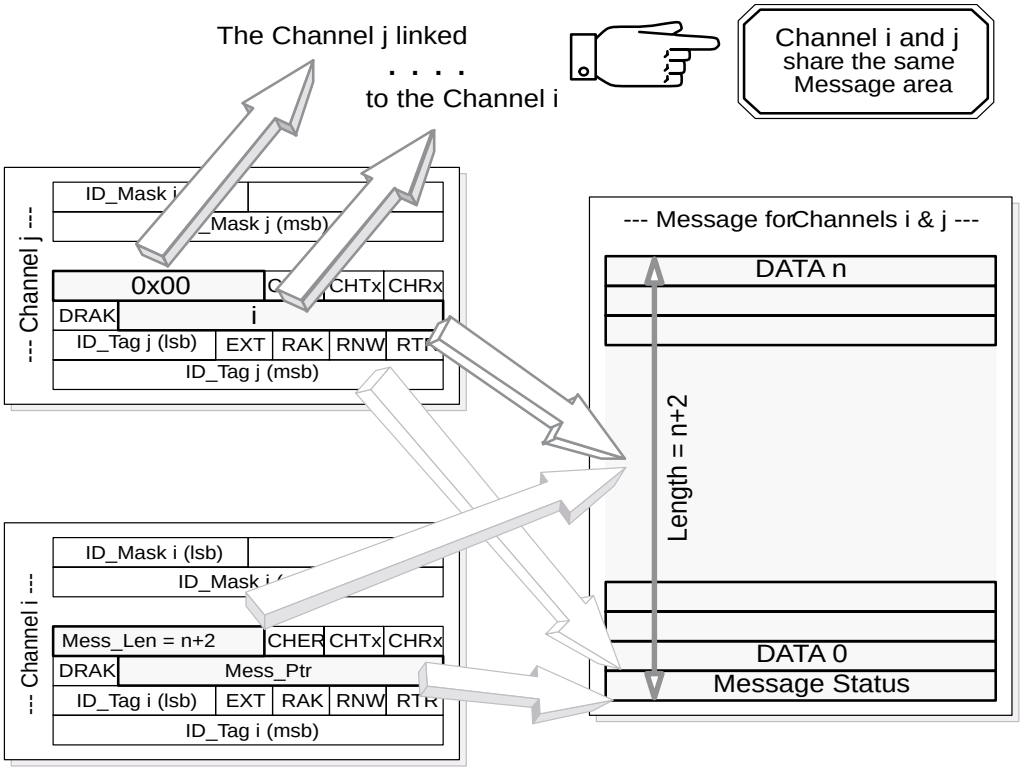
The linkage feature allows to channels to share the same Message area, the message pointer and the message length assumes this property:

- Zero value as message length (M_L [4:0] - *base_address + 0x03*) declares the channel linked to another channel.
- The number of this other channel is defined in the message pointer field (M_P [6:0] - *base_address + 0x02*).
- The pointer and the length values for the Message area are defined only once time, in the register set of this other Channel.

Only one level of linkage can be created. This means, (see Figure 38) a Channel k can be linked to the Channel i but not to Channel j, already defined as linked to Channel i.

All the others can be different between the two channels, for example the ID_Tag.

Figure 38. Linkage Mechanism



This Message area sharing permits either to optimize the allocation of the 128 bytes of DATA, either to perform some special communications between the different nodes of the network.

Absolute Maximum Ratings*

Ambient temperature under bias:	
Automotive	-40°C to 125°C
Storage Temperature	-65°C to 150°C
Voltage on V_{CC} to V_{SS}	-0.5 to +7.0 V
Voltage on any pin to V_{SS}	-0.5 V to V_{CC} +0.5 V

*NOTICE: Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions exceeding those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC Characteristics

Table 12. TA = -40°C to 125°C; VCC = 5 V + 10%; VSS = 0 V

Symbol	Parameter	Min.	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	$0.3 \cdot V_{CC \cdot min}$	V	
V_{IH}	Input High Voltage	$0.7 \cdot V_{CC\ max}$	V_{CC}+0.5	V	
V_{HY}	Hyteresis voltage of trigger CMOS inputs	0.4	-	V	see Figure 2
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 3.2 mA, Vcc min
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -3.2 mA, Vcc min
$\lvert I_L \rvert$	Input Leakage Current (SCLK, MOSI, SS)		5	µA	$0 < V_{IN} < V_{CC}$
$\lvert I_{OZ} \rvert$	Output Tristate Leakage Current (MISO)		5	µA	$0 < V_{IN} < V_{CC}$
R_{PU}, R_{PD}	Input pullup & pulldown resistors	70		kΩ	Note 4
C_{IO}	I/O Buffer Capacitance		10	pF	Not tested
I_{CCSB}	Power Supply Current Sleep mode		50	µA	(Note 1)
I_{CCOP}	Power Supply Current Idle or Active mode		9	mA	(Notes 2, 3)

Notes:
1. Notes: 1. Sleep Mode I_{CCSB} is measured according to with a V_{SS} Clock Signal.
2. Active mode I_{CCOP} is measured at: XTAL = 8 MHz clock, VAN speed rate = 125 KTS/s.
3. I_{CC} is a function of the Clock Frequency. Figure 39 displays a graph showing I_{CC} versus Clock frequency.
4. $\overline{\text{RESET}}$, RxD0, RxD1, RxD2 inputs.

Figure 39. I_{CC}

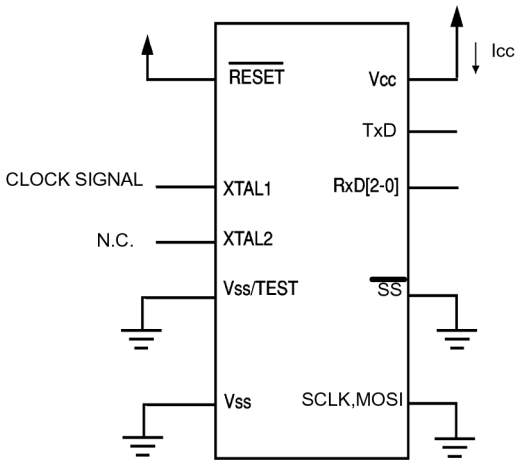
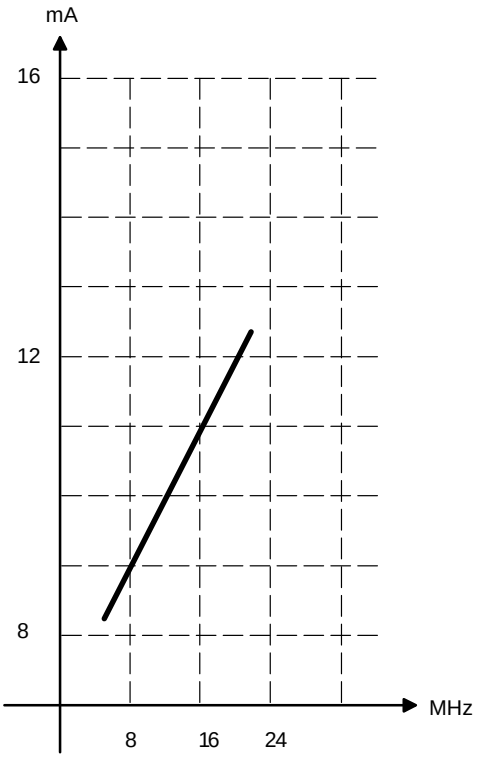


Figure 40. I_{CCOP} Versus Clock Frequency at 125 KTimeslot/s



AC Characteristics

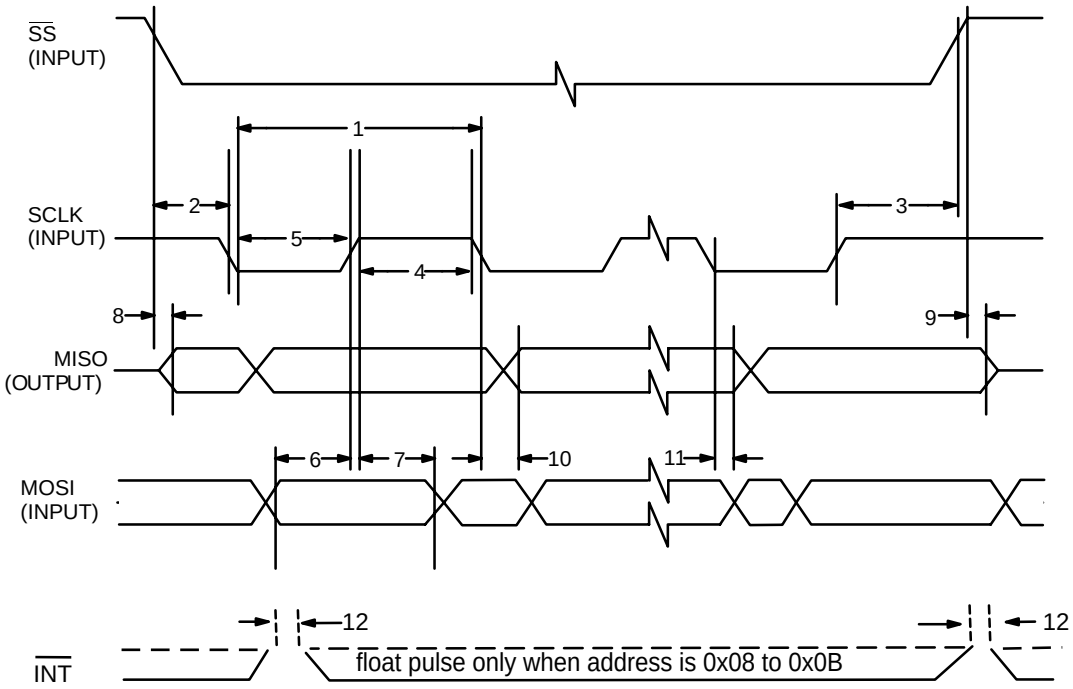
T_A = -40°C to 125°C; V_{CC} = 5V + 10%; V_{SS} = 0V

Microprocessor Inrterface

C_{LOAD} = 200pF on SPI/SCI lines

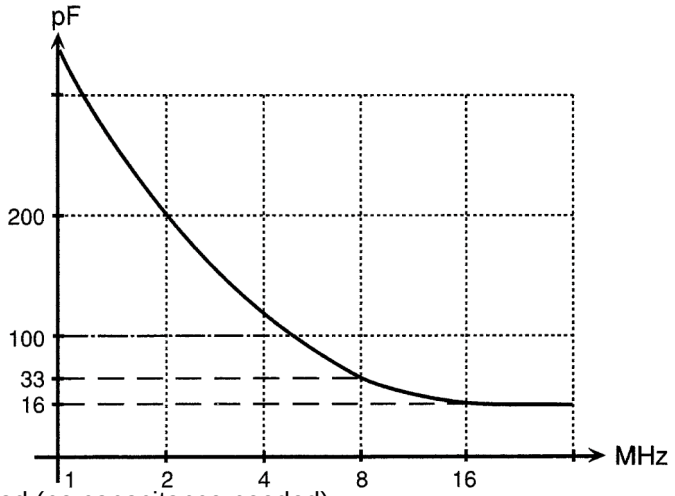
	Symbol	Characteristic	Min	Max	Unit
	f_{OP}	Operating Frequency SPI SCI	dc dc	4 125	MHZ KHZ
1	t_{CYC}	Cycle Time SPI SCI	250 8	- -	ns ms
2	t_{LEAD}	Enable Lead Time	4	-	XTAL Period
3	t_{LAG}	Enable Lead Time	12	-	XTAL Period
4	$t_{W(SCKH)}$	Clock (SCLK) High Time	100	-	ns
5	$t_{W(SCKL)}$	Clock (SCLK) Low Time	100	-	ns
6	$\mathbf{t_{SU}}$	Data Setup Time (Inputs)	40	-	ns
7	t_H	Data Hold Time (Inputs)	40	-	ns
8	t_A	Slave Access Time (Time to Data Active from High-Impedance State)	0	100	ns
9	t_{DIS}	Slave Disable Time (Hold Time to High-Impedance State)	-	200	ns
10	t_V	Data Valid (After Enable Edge)	-	60	ns
11	t_{HO}	Data Hold Time (Outputs After Enable Edge)	0	-	ns
12	t_{IZIL}(*)	$\overline{INT}$ Float Pulse Width	20		ns

Note: (*) Simulated Data



Oscillator Characteristics

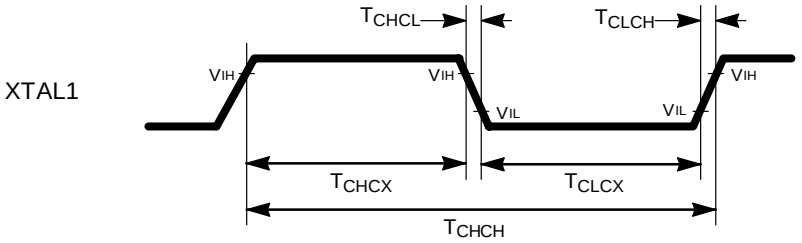
Figure 41. C2 Versus Frequency



C1 = Crystal load (no capacitance needed)

External Clock drive characteristics (XTAL1)

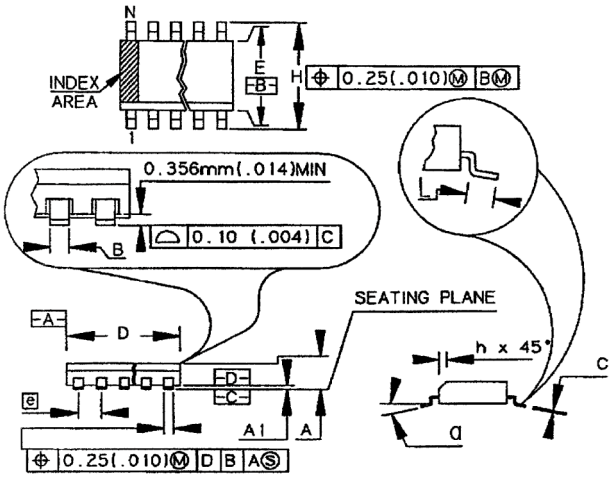
Symbol	Parameter	Min	Max	Unit
T_{CHCH}	Oscillator period	60		ns
T_{CHCX}	High Time	20		ns
T_{CLCX}	Low Time	20		ns
T_{CLCH}	Rise Time		20	ns
T_{CHCL}	Fall Time		20	ns



Packaging

SO 16

SO 16



SO	MM		INCH	
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
B	0.35	0.49	0.014	0.019
C	0.23	0.32	0.009	0.013
D	10.10	10.50	0.398	0.413
E	7.40	7.60	0.291	0.299
e	1.27	BSC	0.050	BSC
H	10.00	10.65	0.394	0.419
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
N	16		16	
a	0° 8°		0° 8°	

Ordering Information

Part Number	Supply Voltage	Temperature Range	Package	Packing
TSS463B-TERA	5V ±10%	-40°C - +125°C	SO 16	Stick
TSS463B-TERA	5V ±10%	-40°C - +125°C	SO 16	Tape & Reel

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

4102C–AUTO–04/02 /xM