National Semiconductor

PRELIMINARY

DS36001
SLIO1 Serial Link Input Output Device

General Description

The DS36001 SLIO1 is designed to conform to the ISO CAN protocol. The Controller Area Network (CAN) is a serial communication protocol that supports distributed real-time control and is specially designed to provide efficient data communication between automotive electronic subsystems.

The SLIO1 implements an 8-bit Serial Linked I/O port for a remote microcontroller with the link being provided by the CAN network. The device features capabilities of a microcontroller including, the ability to generate an interrupt for the master when one of it's I/O pins changes state.

The DS36001 is designed to allow the implementation of very low cost nodes. The port has been defined so as to cover the widest possible application range. To reduce the overall system cost, an on-board oscillator has been developed which requires no external components.

The use of the CAN bus and SLIO1 nodes represents a very cost effective way of increasing the I/O capability of a microcontroller and reducing the amount of wiring which is required to connect all peripherals to the microcontroller.

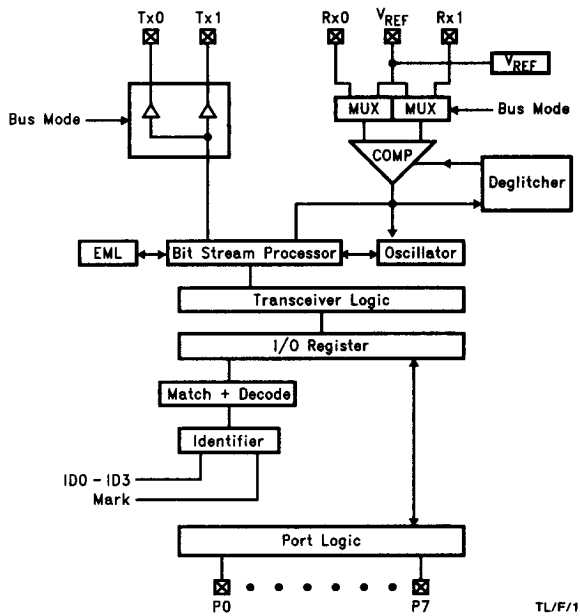
Features

- Supports CAN (Controller Area Network) specification 2.0 B.
- Provides variable port configuration
- Calibration of on-board oscillator is done internally (without external components)
- Operates from 20 Kbits/s to 125 Kbits/s
- On-chip error detection logic to provide automatic diagnostic
- Built-in reference voltage of 2.5V (one half of the power supply)
- Capable of operating in a single wire bus configuration. This feature guarantees safe operation even when one of the two wires is damaged
- Provides individual Enable/Disable for each port

Applications

- Automotive
  — Body electronics and instrumentation
- Industrial applications
  — Sensor/actuators interface
- Microprocessor-based System Designs
  — Extension of I/O capabilities of microprocessors

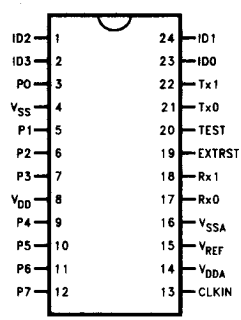
Logic Diagram



TL/F/11931–1

FIGURE 1

Connection Diagram



TL/F/11931–2

FIGURE 2. DS36001
Order Number DS36001TM
See NS Package Number M24B
Order Number DS36001TN
See NS Package Number N24D

Table of Contents

Absolute Maximum Ratings (Notes 1 and 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	6.5V
Input Voltage (Any Input)	V_{CC} + 0.5V to GND − 0.5V
D.C. Output Current for I/O Port Pins	±5 mA
D.C. Output Current for All Other Pins	±25 mA
Power Dissipation at 25°C	200 mW
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 4 sec.)	260°C

Recommended Operating Conditions

	Min	Max	Units
Supply Voltage V_{CC}	4.8	5.2V	V
Input Voltage (Any pin)	−0.3	V_{CC} + 0.5	V
Operating Temperature (free air)	−40°	+125°	C

DC Electrical Characteristics T_A = −40°C to +125°C, V_{CC} = 5V ± 4% (Notes 2 and 3)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
INPUT COMPARATOR (Rx0 and Rx1)						
V_{IH}	Minimum Input High Voltage		V_{CC} − 1.5			V
V_{IL}	Maximum Input Low Voltage				1.5	V
V_{diff} (Dom)	Differential Voltage (Dominant)	(Note 4)			−25	mV
V_{diff} (Rec)	Differential Voltage (Recessive)	(Note 4)	+25			mV
OUTPUT DRIVERS (Tx0 and Tx1)						
V_{OL}	Output Voltage Low	I_{OL} = 1.5 mA @ Tx0			0.1	V
V_{OH}	Output Voltage High	I_{OH} = −1.5 mA @ Tx1	V_{CC} − 0.1			V
CONTROL SIGNALS (EXTRST, TEST, CLKIN)						
V_{IL}	Input Low Voltage				1.5	V
V_{IH}	Input High Voltage		3.5			V
DIGITAL PARALLEL PORT (P0–P7), Ext Clk						
V_{OL}	Output Low Voltage	Sink Current = 4.0 mA			1.0	V
V_{OH}	Output High Voltage	Source Current = −4.0 mA	V_{CC} − 0.1			V
V_{IL}	Input Low Voltage				1.5	V
V_{IH}	Input High Voltage		3.5			V
I_{IH}	Input High Current	V_{IN} = V_{CC} or GND	−10		+10	μA
I_{IL}	Input Low Current	V_{IN} = V_{CC} or GND	−10		+10	μA
V_{CL}	Input Diode Clamp Voltage	I_{CLAMP} = −12 mA	−1.5			V
I_{CC}	Supply Current			26		mA
I_{SL}	Sleep Current				500	μA
V_{REF}	Reference Voltage	I_{OUT} ≥ −75 μA	(V_{CC}/2) − 0.12		(V_{CC}/2) + 0.12	V

Note 1: "Absolute Maximum Ratings" are those beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

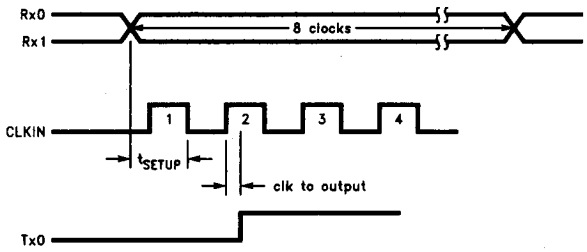
Note 2: All input and/or output pins shall not exceed V_{CC} plus 0.5V and shall not exceed the absolute maximum rating at anytime, including power-up and power-down.

Note 3: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified. All typical values are specified under these conditions: V_{CC} = 5V and T_A = 25°C, unless otherwise stated.

Note 4: V_{diff} = Rx0–Rx1.

AC Electrical Characteristics $T_A = -40°C$ to $+125°C$, $V_{CC} = 5V \pm 4\%$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_d Total	Total Delay of the Input Comparator and Output Driver	$1.5V < (V_{Rx0} + V_{Rx1}) < V_{CC} - 1.5V$			200	ns
t_{CLK}	Clock Period	CLKIN = Ext Clock	100		250	ns
CAN Propagation Time						
t_{total}	$t_{setup} + t_{CLK}$ to Output Tx0	(See Waveform in *Figure 3*)			200	ns



($t_{SETUP} + t_{CLK}$ to Tx0 < 200 ns)
viewed at Tx0 in TEST mode

TL/F/11931–3

FIGURE 3. Waveforms in TEST Mode

Pin Description (Note 5)

TABLE I. Pin Description

Pin Name	Number of Pins	Input/Output	Description
Clkin	1	I	External Clock
ExtRst	1	I	Reset and Decode Control
ID0–ID3	4	I	Identifiers
P0–P7	8	I/O	Port Receiver Input and Driver Output
Rx0–Rx1	2	I	Bus Driver Inputs
Tx0–Tx1	2	O	Bus Driver Outputs
TEST	1	I	Decode Control
V_{ref}	1	O	Reference Voltage
V_{DD}	1		Digital Power Supply
V_{DDA}	1		Analog Power Supply
V_{SS}	1		Digital Ground
V_{SSA}	1		Analog Ground

Note 5: The above listed functions of the pins are only valid for the normal mode of operation. The normal mode of operation is achieved by placing low on the TEST and ExtRst pins. (See information on the testing below.)

1.0 Introduction

The SLIO1 contains the following circuit blocks.

Bit Stream Processor (BSP)
Oscillator
Cyclic Redundancy Check Register (CRC Register)
Transmit/Receive Shift register (Tx/Rx Shift Register)
Match & Decode Logic
Identifier Logic
Port Logic
Error Management Logic (EML)

1.1 BIT STREAM PROCESSOR

The Bit Stream Processor (BSP) is a sequencer controlling the data stream between the Tx/Rx Shift Register, CRC Register, Identifier, Port Logic, and the bus line. The BSP also controls the Error Management Logic (EML) and the oscillator such that functions such as: reception, arbitration, transmission, and error signalling are performed according to the CAN protocol and the correct calibration of the oscillator's pre-scaler is maintained. Note that the automatic retransmission of messages which have been corrupted by noise or other external error conditions on the bus line is handled by the BSP.

1.2 OSCILLATOR

The clock is generated from the on-board oscillator which is calibrated using the calibration message received via CAN bus. These calibration messages are sent by those nodes which contain quartz controlled clocks. The circuit for calibrating the on-board oscillator is implemented in the DS36001; therefore, a frequency variation of 500% can be tolerated by the system. In order to maintain the clock synchronization by the SLIO1 nodes, a calibration message must be sent regularly. As a consequence of the internal clock generation, the bus speed range of such a SLIO1 node is limited between 20 kbits/s to 125 kbits/s.

1.3 REGISTERS AND COUNTERS

1.3.1 Cyclic Redundancy Check Register

This register generates the Cyclic Redundancy Check (CRC) code which is transmitted after the data bytes and checks the CRC code of incoming messages. This is done by dividing the data stream by the code generator polynomial.

1.3.2 Transmit/Receive Shift Register

This Tx/Rx Shift Register holds the destuffed bit stream from the bus line to allow the parallel access to the Identifier for the acceptance match test and, afterwards, the parallel transfer of the two data bytes to the port logic.

1.3.3 Identifier Register

During Reset, the programmabie four bits of the identifier are stored into this register as defined by the pull-up or pull-down resistors on the pins **ID0–ID3**. The other identifier bits are fixed and can oniy be changed by making a new rnask for the chip. The last bit of the identifier is generated by the BSP, depending on the direction of the message.

1.3.4 Error Counters

In each CAN module there are two error counters to perform a sophisticated error management. The Receive Error Count (REC) is 7-bits wide and switches the device to the error passive state if it overflows. The Transmit Error Count (TEC) is 8-bits wide. If it is greater then 127 the device is also switched to the error passive state. As soon as the TEC overflows the device is switched bus-off, i.e., it does not participate in any bus activity.

The following errors can be detected and lead to an increase by eight of either the Receive or Transmit Error Count in every module detecting.

- Bit Error — the transmitted bit is not the received one
- Stuff Rule — there is no stuff bit where it is supposed to be
- Frame check — a fixed frame bit does not have the specified value
- Bit CRC check — the calculated CRC does not match the received one
- ACK check — a transmitting node does not get any acknowledgment

1.4 PORT LOGIC

This block contains logic which enables the programming of the port functions. It interprets data received from the Tx/Rx Shift Register and loads data to be transmitted into this register.

1.5 ERROR MANAGEMENT LOGIC

The Error Management Logic (EML) is responsible for the fault confinement of the CAN device. All messages are received, checked and acknowledged by any node in the network. Even messages which are filtered by the acceptance filter are checked for errors. If any node detects an error it starts transmitting an error frame.

There are two error counters, one for the transmitted data and one for the received data, which are incremented as soon as an error occurs. If either counter goes beyond a specific value the node goes to an error state. A valid frame causes the error counters to decrease.

2.0 DS36001 Functional Description

2.1 BUS MODES

The comparator monitors levels of the Rx0 and Rx1 input pins. The output of the comparator is "1" if the voltage levels of the CAN bus lines are regarded as recessive and it is "0", if they are regarded as dominant.

There are three possibilities to generate the output signal of the comparator. In normal operation, the CAN bus is configured of two wires, and the Rx0 and Rx1 levels are compared against each other. If one of the two wires is damaged, the SLIO1 can still operate in a single wire CAN bus configuration. In this case, the level of one single wire is compared against the on-board generated reference voltage V_{REF}. Additionally, if only the Rx0 input is regarded the Tx1 output is turned off, to take into account the possibility of a short circuit between the two CAN bus lines. When enabled, the SLIO1 can exist in the following four bus configurations.

2.0 DS36001 Functional Description (Continued)

TABLE II. Comparator Input Configuration

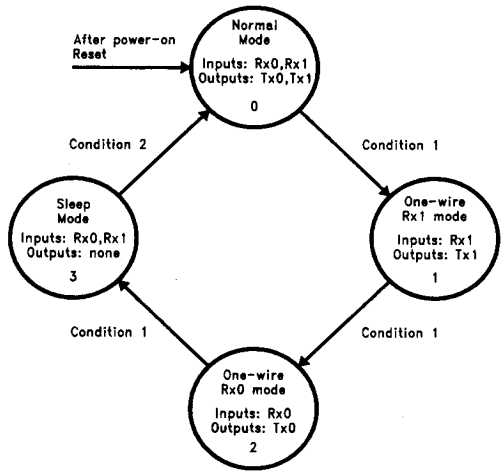
	Bus Mode	Mode Bits	Recessive	Dominant	Comment
0	Differential	00	Rx0 > Rx1	Rx0 < Rx1	Differential Communication
1	One-Wire Rx1	01	Rx1 < V_{REF}	Rx1 > V_{REF}	Communication on Tx1/Rx1
2	One-Wire Rx0	10	Rx0 > V_{REF}	Rx0 < V_{REF}	Communication on Tx0/Rx0
3	Sleep	11	Rx0 > V_{REF} and Rx1 < V_{REF}	*Rx0 < V_{REF} *Rx1 > V_{REF}	Low Current Mode

*Wake up condition.

When the external reset is performed, the normal mode is active and the SLIO1 waits for a suitable calibration message. If it does not receive such message within adequate time, it switches to the next mode in the numerical order. If it reaches sleep mode, all activities are stopped until the next dominant bit is monitored on the bus.

After the hardware reset, the SLIO1 will be always in the bus mode 0. There are three conditions to switch to the next bus mode, an overflow of the bit counter (8 Kbits since the last calibration message or since reset), or the Receive or Transmit Error Counters reach the error passive limit of 128. By switching to the next bus mode, the SLIO1 is internally reset and waits for the next calibration message before starting the CAN protocol bus off recovery sequence and going on bus again. The SLIO1 switches the bus modes from the bus mode 0 to mode 1 to mode 2. The SLIO1 can switch to the sleep mode only from the bus mode 2. The SLIO1 will leave this sleep mode upon detecting a dominant bit on either of the two bus lines.

The following state diagram shows the switch-over conditions for the possible four CAN-bus modes.



Condition 1: bit counter overflow (>8191) or error counter overflow (>255).
Condition 2: Dominant bit detected on Rx0 or Rx1

TL/F/11931–4

FIGURE 4. CAN-bus Modes and Switch-Over Conditions

TABLE III. CAN Bus Modes

	Bus Mode	Mode Bits	Reception			Transmission	
			Diff	Rx1	Rx0	Tx1	Tx0
0	Differential	00	x			x	x
1	One-wire Rx1	01		x		x	
2	One-wire Rx0	10			x		x
3	Sleep	11		x	x		

Diff: differential input voltage on Rx0 and Rx1 (recessive Rx0 > Rx1)
Rx1: input voltage on Rx1 compared to V_{REF} (recessive Rx1 < V_{REF})
Rx0: input voltage on Rx0 compared to V_{REF} (recessive Rx0 < V_{REF})

2.0 DS36001 Functional Description (Continued)

The Tx1 output is disabled in bus mode 2 to tolerate the short-circuit between the CAN bus wires CAN_H and CAN_L.

The Deglitcher is an active filter which is realized by inhibiting the comparator output for 8 clock cycles after performing a signal change at the comparator output. A glitch at the comparator input will simply be ignored because, in order for a glitch to cause a change in the signal level at the comparator output, it should last for at least 8 clock cycles. The deglitcher will increase the system reliability.

2.2 OSCILLATOR AND CALIBRATOR

The on-board oscillator is an RC type oscillator. This oscillator is calibrated to the exact frequency required by examining messages coming over the CAN bus. On power-up, a calibration must be sent on the bus to calibrate the oscillator. Once the oscillator is well calibrated to correctly receive messages, the calibration logic will then only calibrate itself to nodes which send a particular calibration message, labeled by a special Identifier. The calibration message must be sent periodically (typically 20 ms) or the device will stop responding. This calibration message may be sent only by the quartz controlled nodes. Only when the SLIO1 is correctly synchronized to a quartz node, will it allow itself to write a dominant level onto the bus. The Identifier of this calibration message is defined by hardware and can only be changed by modifying one mask of the chip.

If the SLIO1 does not receive a suitable message for calibration of its oscillator within 8 Kbit times after the last suitable message or after waking up, it will switch to the next bus mode. After trying both one wire bus modes without success it will enter the sleep mode. During this mode, the total power consumption is reduced to less than 500 μA. Any external bus activity (either bus wire in the dominant state) will cause the device to wake up, whereupon it will reinitialize itself and calibrate its oscillator. On wakeup the bus mode is reset to normal two wire differential operation (bus mode 0). Note that on switching bus modes or entering sleep mode, the device is effectively reset to its power-up state, resetting also the port registers. The SLIO1 broadcasts this reset by sending a special message after the successful calibration.

As distinct from other CAN nodes, the SLIO1 CAN node is not able to wake up by local events or to wake up other nodes, because the SLIO1 CAN node cannot start transmission if its oscillator is not calibrated. Therefore, to keep the network alive, a quartz node should send the calibration message regularly with a repeating period less (to take into account the possibility of bus errors) than the maximum distance of 8 kbit/Baud-Rate.

Note: The calibration of the oscillator requires at least three consecutive messages, the second and third of them error free, if the node was in sleep mode. Therefore, it is possible for the quartz controlled node to go error-passive before it gets an acknowledge to its wake-up message.

Requirements for messages which are used for calibration are:

The message must come from a quartz controlled node and have the Identifier: 000 1010 1010.

In the message, the first recessive to dominant transition after the Control Field must be followed by another recessive to dominant transition in a distance of exactly 32 bit times, including stuff bits.

One suitable message is (there are many others, using different data bytes)

Identifier = 000 1010 1010

DLC = 0010

Data (2 bytes Data Field) = 10101010 00000100

Bus bit stream = 0 000 1010 1010 0 00 0|010 10101010 0000|0100

000|01011100000|0 ("|" signifies a recessive stuff bit)

In this example, the first recessive to dominant transition after the Arbitration Field is in the first data byte, from the first to the second bit. The bit number 32, after the first bit of the first data byte is (there are three stuff bits in the Data and CRC Fields) the last but one bit of the CRC Field. This last but one bit is recessive ("|") and is followed by a dominant ("0") bit. The total length of this message (from the Start of the Frame to the end of Intermission) is 67 bits.

2.3 ERROR MANAGEMENT LOGIC

2.3.1 Error States

With respect to fault confinement a unit may be in one of the following three states:

- error active
  A node is "error active" once it detects an error but has not yet become "error passive".
- error passive
  A node is "error passive" when the Transmit Error Count equals or exceeds 128, or when the "Receive Error Count" equals or exceeds 128. An error condition letting a node become "error passive" causes the node to send an Passive Error Flag.
- bus off
  A node is "bus off" when the Transmit Error Count is greater than or equal to 256.

An "error active" unit can normally take part in bus communication and send an Active Error Flag when an error has been detected.

An "error passive" unit must not send an Active Error Flag. It takes part in bus communication, but when an error is detected only Passive Error Flag is sent. Also, after the transmission, an "error passive" unit will wait before initiating a further transmission.

A "bus off" unit is not allowed to have any influence on the bus, (e.g., output drivers switched off.)

Special error handling is performed at following situations:

A stuff error occurs during arbitration when a transmitted recessive stuff bit is received as a dominant bit. This does not lead to an incrementation of the TEC.

An ACK error occurs in an error passive device and no dominant bits are detected in the passive error flag. This does not lead to an incrementation of the TEC.

A valid reception or transmission leads to a decrementation of the error counters by one. *Figure 5* shows the connection of different error states according to the error counters.

2.0 DS36001 Functional Description (Continued)

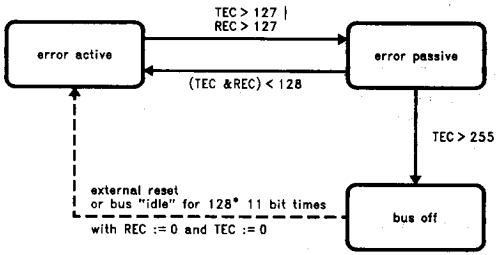


FIGURE 5. CAN Bus States

2.3.2 Rules and Exceptions

There are two error counters implemented in this device for the error detection.

Transmit Error Count

Receive Error Count

These counters are modified according to the following rules:

1. When a receiver detects an error, the Receive Error Count will be increased by 1, except when the detected error was a Bit error during the sending of an Active Error Flag.
2. When a receiver detects a "dominant" bit as the first bit after sending an error flag the Receive Error Count will be increased by 8.
3. When a transmitter sends an Error Flag the Transmit Error Count is increased by 8.

Exception 1:

If the transmitter is "error passive", and it detects an Acknowledgment Error because of not detecting a "dominant" ACK and does not detect a "dominant" bit while sending its Passive Error Flag.

Exception 2:

If the transmitter sends an Error Flag because a stuff error occurred during arbitration whereby the stuff bit is located before the RTR bit, and should have been "recessive" and has been sent as "recessive" but monitored as "dominant".

In both exceptions, the Transmit Error Count is not changed.

If a transmitter detects a bit error while sending an Active Error Flag or an Overload Flag the Transmit Error Count is increased by 8.

If a receiver detects a bit error while sending an Active Error Flag or an Overload Flag the Receive Error Count is increased by 8.

Any node tolerates up to 7 consecutive "dominant" bits after sending an Active Error Flag, Passive Error Flag or Overload Flag. After detecting the 14th consecutive "dominant" bit (in case of an Active Error Flag or an Overload Flag) or after detecting the 8th consecutive "dominant" bit following a Passive Error Flag, and after each sequence of additional eight consecutive "dominant" bits every transmitter increases its Transmit Error Count by 8 and every receiver increases its Receive Error Count by 8.

After the successful transmission of a message (getting ACK and no error until the End of Frame is finished) the Transmit Error Count is decreased by 1 unless it was already 0.

After the successful reception of a message (reception without error up to the ACK SLOT and the successful sending of the ACK bit), the Receive Error Count is decreased by 1, if it was between 1 and 127, if the Receive Error Count was 0, it stays 0, and if it was greater than 127, then it will be set to a value between 119 and 127.

An "error passive" node becomes "error active" again when both the Transmit Error Count and the Receive Error Count are less than or equal to 127.

A node which is "bus off" is permitted to become "error active" (no longer "bus off") when both its error counters are set to 0 after 128 occurrences of 11 consecutive "recessive" bits have been monitored on the bus.

Note: If during system start-up only 1 node is on-line, and if this node transmits some message, it will get no acknowledgment, detect an error and repeat the message. It can become "error passive" but not "bus off" due to this condition.

2.4 PORT FUNCTIONS

The port functions are controlled by various registers. Each writeable register may be written by sending a Data Frame with a two byte long Data Field where the lower part of the first byte is the Register Marker for the addressed register and the remaining byte is the information which will be written to the register. The first part of the first byte is reserved.

2.4.1 SLIO1 Status Information and Register Marker

The first byte of each message transmitted by a SLIO1, contains status information and the Register Marker to describe the contents of the following byte.

7	6	5	4	3	2	1	0
Rstd	EW	Bus Mode		0	Register Marker		

Four parts of the status information are available:

Rstd (Bit 7) — This bit is set in the first message after the SLIO1 has been reset.

Rstd = 1: SLIO1 has just entered the state where oscillator is calibrated.

Rstd = 0: Other data frame

EW (Bit 6) — This bit is set if the Receive Error Count or the Transmit Error Count has exceeded, at least temporarily, the Error Warning limit (32) since the last successful transmission of a message.

EW = 1: error warning limit (32) reached.

EW = 0: error warning limit not reached.

Bus Mode (Bits 5,4) — Valid values for the Bus Mode are [0, 1 and 2]

00: Mode 0 (two-wire mode)

01: Mode 1 (one-wire mode, Tx0 disconnected)

10: Mode 2 (one-wire mode, Tx1 disconnected)

Reserved (Bit 3) — Reserved and transmitted as "0".

2.0 DS36001 Functional Description (Continued)

Register Marker (Bits 2,1,0) — Register marker bits are used to select the I/O register (Register Markers are shown below).

Marker	Abbr.	Function
0	P	Input Data (Port) Register (read only)
1	PE	Positive Edge Register (write only)
2	NE	Negative Edge Register (write only)
3	OD	Output Data Register (write only)
4	DD	Data Direction Register (write only)
5–7		reserved

These registers can be cleared by activating the ExtRst pin, or when entering the sleep mode. On wake up, or after reset, after the oscillator is calibrated, the SLIO1 will transmit the contents of the Input Data Register (all port pins input), with the Rstd bit in the status byte set to "1". In this way the CPU is made aware that a reset has been executed by the SLIO1 CAN node.

2.4.2 Input Data Register

Register Marker = 0

This register is loaded with the actual digital value of the port pins P0..P7 when it is transmitted by the SLIO1. The content of this register is sent in response to a Remote Frame, or by a SLIO1 initiated transmission when an edge has been detected on a pin and has been enabled in the appropriate Edge Register. A high/low level on the pin is transmitted as a 1/0 data bit respectively. Note that after detecting an edge, the register will not actually be loaded into the Tx/Rx shift register until the Control Field in the CAN message has been sent. This effectively provides an input settling delay. Additionally, this register will automatically be sent by the SLIO1 after the chip has been reset by the ExtRst signal, upon waking up after sleep mode or after changing its Bus Mode, once it has successfully calibrated its oscillator.

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

2.4.3 Positive Edge Register

Register Marker = 1

This register is used to enable automatic transmission of the contents of the input data register in the event that the corresponding pin P0..P7 makes a positive transition. A logical one enables such a transmission on a rising edge of the corresponding pin; a logical zero disables it.

7	6	5	4	3	2	1	0
PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0

2.4.4 Negative Edge Register

Register Marker = 2

As for the positive edge register but for falling edges.

7	6	5	4	3	2	1	0
NE7	NE6	NE5	NE4	NE3	NE2	NE1	NE0

2.4.5 Output Data Register

Register Marker = 3

This register holds the logical value which is output to the port pins P0..P7 which are enabled as outputs by the corresponding bits in the Data Direction Register. A 1/0 bit in the Output Data Register corresponds to the high/low level respectively on the output pin. This register is written by sending a message with the Register Marker set to 3.

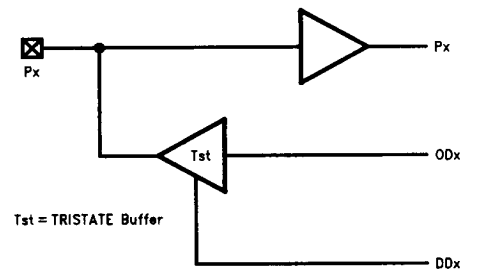
7	6	5	4	3	2	1	0
OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0

2.4.6 Data Direction Register

Register Marker = 4

This register controls which pins will be used for the output. A logical 1 means that the pin will be driven and used as an output. A logical 0 means that the pin will not be driven internally. This register is written by sending a message with the Register Marker set to 4.

7	6	5	4	3	2	1	0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

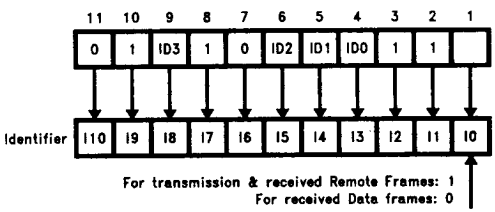


TL/F/11931–6

FIGURE 6. Data Direction Control

2.5 IDENTIFIER PROGRAMMING

During Reset, the port pins will not be driven. At this time four bits for the Identifier for the messages for this chip will be read in from the pins ID0–ID3. The Identifier can therefore be set for each SLIO1 node by using resistors to V_{CC} and V_{SS} connected to these port pins. Port pins which are not otherwise used for input and output may be tied to V_{CC} or V_{SS}. The value of the resistors is determined solely by the fact that they must be able to ensure that the pin is at the required valid logic level before the reset signal is deactivated.



TL/F/11931–7

FIGURE 7. Data and ID Formats

2.0 DS36001 Functional Description (Continued)

The SLIO1 uses the higher priority of the two adjacent 11 bit Identifiers for the Identifier of the messages to be received, and the lower priority Identifier for the transmission. (see Table IV).

Every device uses two adjacent addresses out of the 2032 possible in the standard CAN frame format. The higher prioritized ID (6 or E) is used to set up the device with a data frame. The lower prioritized ID (7 or F) is used for polling the SLIO1 via an RTR and is used from the SLIO1 to transmit its frame. This is required by the CAN protocol as only one transmitter for a specific data frame should exist.

The following table shows different address locations, which can be set up via the SLIO1 pins ID0 to ID3:

TABLE IV. Identifier Address Locations

ID3	ID2	ID1	ID0	Addresses (hex)
0	0	0	0	0x286 0x287
0	0	0	1	0x28E 0x28F
0	0	1	0	0x296 0x297
0	0	1	1	0x29E 0x29F
0	1	0	0	0x2A6 0x2A7
0	1	0	1	0x2AE 0x2AF
0	1	1	0	0x2B6 0x2B7
0	1	1	1	0x2BE 0x2BF
1	0	0	0	0x386 0x387
1	0	0	1	0x38E 0x38F
1	0	1	0	0x396 0x397
1	0	1	1	0x39E 0x39F
1	1	0	0	0x3A6 0x3A7
1	1	0	1	0x3AE 0x3AF
1	1	1	0	0x3B6 0x3B7
1	1	1	1	0x3BE 0x3BF

2.6 TRANSMISSION OF DATA FRAMES

A Data frame that is transmitted by the DS36001 device consists of two bytes. The first byte contains the SLIO1 status information and the register marker. The second byte contains the data from the I/O register. The Identifier will have a logical 1 for its least significant bit. This Identifier is also the Identifier which should be used by another node when sending a Remote Frame. Such a Remote Frame should always have its Data Length Code set to 2.

After successful transmission of a data frame, the SLIO1 delays the transmission of a possibly further pending message for a 3-bit time. This provides an opportunity for other CAN controllers having lower priority to transmit a message in case of a faulty contact at one of the edge-triggered port pins. In that case, the supererogatory bus load can be reduced by resetting the corresponding bit in the Positive or Negative Edge Register.

2.7 RECEPTION OF DATA FRAMES

Received data frames have the same format as the transmitted frames. Only the direction bit in the arbitration field is different. The status bits (Rstd, EW, Bus Mode, etc.) are ignored during reception. The Input Data Register will also be transmitted in response to the reception of a Remote Frame with the Transmit Identifier of the SLIO1. The device confirms each reception of a remote frame by transmitting a data frame containing the contents of the addressed I/O register.

2.8 ALTERNATE OPERATING MODES

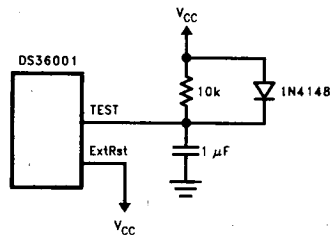
2.8.1 External Clock

In addition to the normal operating mode, the DS36001 device can also operate with the external clock or in the TEST (SCAN) mode. These modes are controlled by the input pins TEST and ExtRst.

TEST	ExtRst	Mode
0	0	Normal mode using the integrated oscillator
0	1	Hardware reset
1	1	Normal mode using an external clock connected to ClkIn
1	0	Production test mode, port pins are redefined (SCAN mode)

How to Use an External Clock

In order to use an external clock, the external clock should be connected to the ClkIn input pin. The ExtRst pin should be connected to high, and the TEST pin can be used for the power-up reset function as shown in *Figure 8*. The frequency of the external clock can be selected between 4 MHz and 10 MHz.

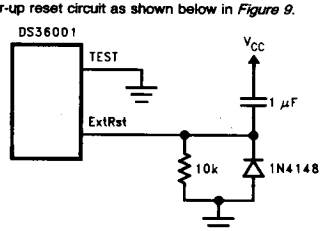


TL/F/11931–8

FIGURE 8. Power-up Reset Circuit for External Clock Mode

The advantage of using an external Quartz-controlled clock is that it will allow the bus bandwidth (frequency limit) to be increased on both ends. For example, with an external clock of 5 MHz, the lower limit of the baud rate can be decreased from 20 kBaud to 10 kBaud. Likewise, with an external clock running at 10 MHz, the upper limit of the baud rate can be increased from 125 kBaud to 250 kBaud.

Note: The power-up reset circuit can also be used for the internal operating mode with internal oscillator. In this case, the TEST pin should be connected to "GND" and the ExtRst pin can be connected to the power-up reset circuit as shown below in *Figure 9*.



TL/F/11931–9

FIGURE 9. Power-up Reset Circuit for Internal Clock Mode

2.0 DS36001 Functional Description

(Continued)

2.8.2 Production Test Mode

In this mode the function of the I/O port is redefined to allow a tester an access to the interior registers and signals; the clock of the integrated oscillator is replaced by an external clock which is connected to pin ClkIn. The information of the internal states, together with the signals at the Rx and Tx pins, provides sufficient data to ensure that the device behaves according to the CAN protocol and that the oscillator is calibrated to the bit stream.

The port pins are defined as shown below:

P0 Select second Function of P1, P3 and P4

P1 Scan-Path Output/Output of integrated Oscillator's Frequency

P2 Output of the calibrated Clock phI

P3 Scan-Path Input/Drive TxO Pin with the Signal of the Rx-Comparator

P4 Load PLA-Out in Scan Registers/Let Bit Counter run 10 Times Faster

P5 Scan-Path ck1

P6 Scan-Path ck2

P7 Error Warning

The Tx1 output pin is driven with the Tx Clock signal during this production test mode.

3.0 Protocol Overview

3.1 CAN DISCLAIMER

This overview describes some of the elements of the CAN protocol. For complete details, see the CAN Protocol Specification.

3.2 CAN FRAME FORMATS

There are two different types of frame used for data transmission and two types of frame used for control purposes in the CAN protocol.

3.2.1 Data Frame

Data frames consist of seven different bit fields:

Start of Frame (SOF)

Arbitration field

Control field (reserved bit, extended frame bit and DLC field)

Data field

CRC field

ACK field

End of Frame (EOF)

(DLC = Data Length Code, see explanation on "Control Field")

SOF	Arbitration Field Identifier + RTR	Control Field	Data Field	CRC Field	ACK Field	EOF
1 Bit	12 Bit	6 Bit	n*8 Bit	16 Bit	2 Bit	7 Bit

$n \in (0,8)$

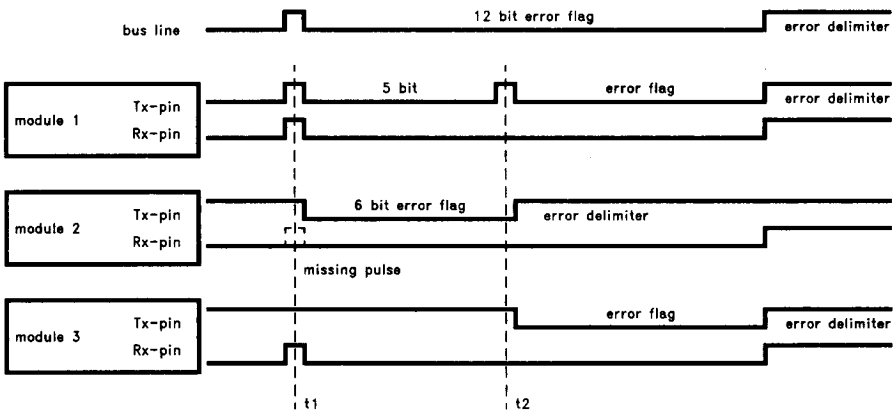
FIGURE 10. CAN Frame Format

3.2.2 Remote Frame

Remote frames are identical to Data frames except that they do not contain the data field. The DLC will contain the length code of the data requested.

3.2.3 Error Frame

The error Frame consists of two bit fields: the error flag and the error delimiter. The error flag field is built up from the various error flags of the different nodes. Therefore, its length may vary from a minimum of six bits up to a maximum of twelve bits depending on when a module is detecting the error. *Figure 11* shows how a local fault at one module (module 2) leads to a 12 bit error frame on the bus.

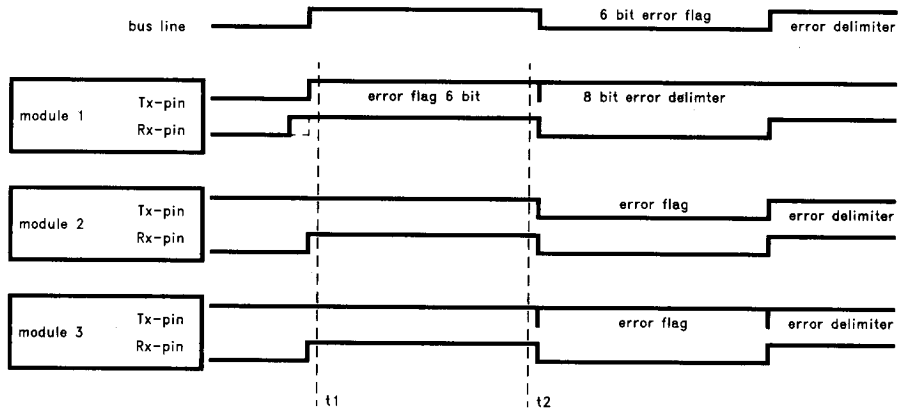


module 1 = error active transmitter detects bit error at t2
module 2 = error active receiver with a local fault at t1
module 3 = error active receiver detects stuff error at t2

TL/F/11931-10

FIGURE 11. Error Frame Length—Error Active Transmitter

3.0 Protocol Overview (Continued)



module 1 = error passive transmitter with local fault at t1
module 2 = error active receiver detects stuff error at t2
module 3 = error passive receiver detects stuff error at t2

TL/F/11931–11

FIGURE 12. Error Frame—Error Passive Transmitter

The bus level may either be dominant for an error-active node or recessive for an error-passive node. An error active node, detecting an error, starts transmitting an active error flag consisting of six dominant bits. This causes the destruction of the actual frame on the bus. The other nodes detect the error flag as either the rule of bit-stuffing or the value of a fixed bit field is destroyed. As a consequence all other nodes start transmission of their own error flag. This means, that the error sequence which can be monitored on the bus has a maximum length of twelve bits.

If an error passive node detects an error it transmits six recessive bits on the bus. This sequence does not destroy a message sent by another node and is not detected by other nodes. However if the node detecting an error was the transmitter of the frame the other modules will get an error condition by a violation of a fixed bit or the stuff rule. *Figure 12* shows how an error passive transmitter transmits a passive error frame and when the error is detected by the receivers.

After any module has transmitted its active or passive error flag it waits for the error delimiter which consists of eight recessive bits before continuing.

3.2.4 Overload Frame

Like an error frame, an overload frame consists of two bit fields: the overload flag and the overload delimiter. The bit fields have the same length as the error frame field: six bits for the overload flag and eight bits for the delimiter. The overload frame can only be sent after the end of frame (EOF) field and so destroys the fixed form of the intermission field. As a consequence all other nodes also detect an overload condition and start the transmission of an overload flag, too. After an overload flag has been transmitted the overload frame is closed by the overload delimiter.

3.3 DATA AND REMOTE FRAME FIELDS

3.3.1 Start of Frame (SOF)

The SOF indicates the beginning of data and remote frames. It consists of a single "dominant" bit. A node is only allowed to start transmission when the bus is idle. All nodes have to synchronize to the leading edge (first edge after the bus was idle) caused by the Start of Frame of the node which starts transmission first.

3.3.2 Arbitration Field

The arbitration field is composed of the identifier field and the RTR (Remote Transmission Request) bit. The value of the RTR bit is "dominant" in a data frame and "recessive" in remote frame.

3.3.3 Control Field

The control field consists of six bits. It starts with two bits reserved for future expansion followed by the four-bit Data Length Code. Receivers must accept all possible combinations of the two reserved bits. Until the function of these reserved bits is defined, the transmitter shall only send "0" bits. The first reserved bit is actually defined to indicate an extended frame with 29 Identifier bits if set to "1". The DS36001 will receive extended frames, and send ACK, however, no output will be changed.

The Data Length Code indicates the number of bytes in the data field. This Data Length Code consists of four bits. The data field can be of length zero. The admissible number of data bytes for a data frame ranges from 0 to 8. Other values than those specified in Table V may not be used.

3.0 Protocol Overview (Continued)

TABLE V. Coding of the Number of Data Bytes by the Data Length Code

Number of Data Bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0

3.3.4 Data Field

The Data field consists of the data to be transferred within a data frame. It can contain 0 to 8 bytes and each byte contains 8 bits. A remote frame has no data field. The SLIO1 can only have two bytes of data.

3.3.5 CRC Field

The CRC field consists of the CRC sequence followed by the CRC delimiter. The CRC sequence is derived by the transmitter from the modulo 2 division of the preceding bit fields, from the SOF to the end of the data field, excluding stuff-bits by the generator polynominal:

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

The remainder of this division is the CRC sequence transmitted over the bus. On the receiver side the module divides all bit fields until the CRC delimiter, excluding stuff- bits, and checks if the result is zero. This will then be interpreted as a valid CRC. After the CRC sequence a single recessive bit is transmitted as the CRC delimiter.

3.3.6 ACK Field

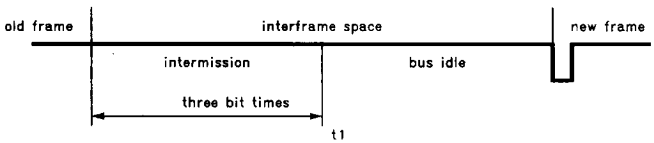
The ACK field is two bits long and contains the ACK slot and the ACK delimiter. The ACK slot is filled with a recessive bit by the transmitter. This bit is overwritten with a dominant bit by every receiver that has received a correct CRC sequence. The second bit of the ACK field is the acknowledge delimiter. It has a fixed form of recessive bits. As a consequence the acknowledge flag of a valid frame is surrounded by two recessive bits, the CRC delimiter and the ACK delimiter.

3.3.7 EOF Field

The End of Frame field closes a data and a remote frame. It consists of seven recessive bits.

3.3.8 Specification of the Inter Frame Space

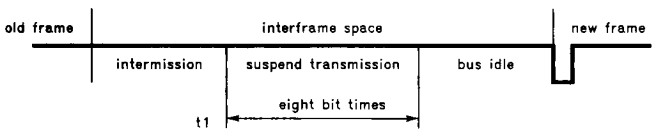
Data and remote frames are separated from every preceding frame (data, remote, error and overload frames) by the inter frame space, see *Figure 13* and *Figure 14* for details. Error and Overload frames are not separated by an inter frame space in front of them. They can be transmitted as soon as the condition occurs. The inter frame space consists of a minimum of three bit fields relating to the error state of the node.



t1 is the first possible start bit of a new frame

TL/F/11931–12

FIGURE 13. Inter Frame Space for Nodes Which are Not Error Passive or Have Been Receivers for the Last Frame



t1 except the error passive module which has transmitted the last frame any module can start transmission

TL/F/11931–13

FIGURE 14. Inter Frame Space for Nodes Which are Error Passive and Have Been Transmitters for the Last Frame

3.0 Protocol Overview (Continued)

These bit fields are coded as follows.

The intermission has the fixed form of three "recessive" bits. While this bit field is active no node is allowed to start a transmission of a data or a remote frame. The only action to be taken is signaling an overload condition. This means that also an error in this bit field would be interpreted as an overload condition. Suspend transmission has to be inserted by error-passive nodes that were transmitters for the last message. This bit field has the form of eight recessive bits. However, it may be overwritten by dominant start-bit from another non-error passive node which starts transmission. The bus idle field consists of recessive bits. Its length is not specified and depends on the bus load.

3.4 MULTI-MASTER PRIORITY BASED BUS ACCESS

The CAN protocol is a message based protocol that allows a total of 2032 different messages in the standard format and 512 million different messages in the extended frame format.

The CAN protocol allows several transmitting modules to start transmission at the same time as soon as they monitor the bus to be idle. During the start of transmission, every node monitors the bus line to determine if its message is overwritten by a message with a higher priority. As soon as a transmitting module detects another module with a higher priority accessing the bus, it stops transmitting its own frame and switches to receive mode. For illustration see *Figure 17*.

3.5 MULTICAST FRAME TRANSFER BY ACCEPTANCE FILTERING

Every CAN Frame is placed on the common bus. Each module receives every frame and filters out the frames which are not required for the module's task. For example if the dashboard sends a request to switch on the headlights, the CAN module responsible for the brake lights must not react to this message.

3.6 REMOTE DATA REQUEST

A CAN master module has the ability to set a specific bit called the Remote Transmission Request bit in a frame. This causes another module, either another master or a slave, to transmit a data frame.

3.7 SYSTEM FLEXIBILITY

Additional modules can be added to an existing network without a configuration change. These modules can either perform completely new functions requiring new data like an automatic window opener, or process existing data to perform a new function, such as oil-pressure measurement.

3.8 SYSTEM WIDE DATA CONSISTENCY

As the CAN network is message oriented, a message can be used like a variable which is automatically updated by the controlling processor. If any module cannot process information it can send an overload frame. However, the implementation of overload frame is optional.

If a data or remote frame was overwritten by either a higher-prioritized data frame or an error frame, the transmitting module will automatically retransmit it.

3.9 FRAME CODING

Remote and Data Frames are NRZ coded with bit-stuffing. In every bit field which holds computable information for the interface i.e., SOF, arbitration field, control field, data field (if present) and CRC field. Error and overload frames are NRZ coded without bit stuffing.

3.10 BIT STUFFING

After five consecutive bits of the same value, a stuff bit of the inverted value is inserted by the transmitter and deleted by the receiver (see *Figure 15*).

destuffed bit stream	100000x	011111x
stuffed bit stream	1000001x	0111110x
		$x \in \{0,1\}$

FIGURE 15. Bit-Stuffing

3.11 ORDER OF BIT TRANSMISSION

A frame is transmitted starting with the SOF, sequentially followed by the remaining bit fields. In every bit field the MSB is transmitted first. The transmission order from either data byte is not defined in the CAN specification. Here, it is assumed that the data bytes are transmitted in the same way as the bits are, i.e., most significant byte first.

SOF	Identifier	RTR	Control	Data	CRC
	ID10 ID0		MSB LSB	MSB LSB	MSB LSB

↑ first bit transmitted

FIGURE 16. Order of Bit Transmission

3.12 FRAME VALIDATION

According to the CAN 2.0 specification frames have a different validation point for the transmitter and receiver. A frame is valid for the transmitter of a message if there is no error until the first bit of End of Frame field. A frame is valid for a receiver, if there is no error until the last but one bit of the End of Frame.

3.13 FRAME ARBITRATION AND PRIORITY

Except for an error passive node which transmitted the last frame, all nodes are allowed to start transmission of a frame after the intermission, which can lead to two or more nodes starting transmission at the same time. To prevent a node from destroying another node's frame it monitors the bus during transmission of the identifier field and the RTR-bit. As soon as it detects a dominant bit while transmitting a recessive bit it releases the bus, immediately stops transmission and starts receiving the frame. This causes no data or remote frame to be destroyed by another. Therefore the highest priority message with the identifier 0x000 always gets the bus.

3.0 Protocol Overview (Continued)

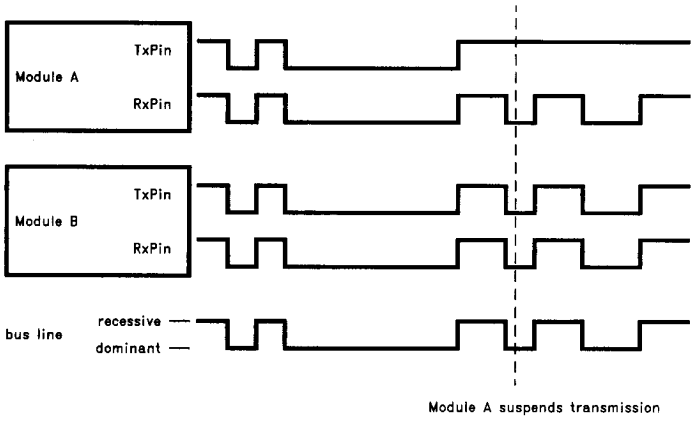


FIGURE 17. Message Arbitration

There are three more items that should be taken into consideration to avoid unrecoverable collision on the bus:

- Within one system each message must be assigned to a unique identifier. Consider two messages, e.g. oil-pressure and rpm, having the same identifier but in most cases different data. So both modules may start transmission of a frame at the same time and both win arbitration. This will always result in bit errors in either or both of the modules as one is transmitting a dominant data bit while the other is transmitting a recessive data bit.
- Data frames with a given identifier and a non-zero DLC should only be initiated by one node. Otherwise in the worst cases, two nodes count up to the bus-off state, due to bit errors, if they always start transmitting the same ID with different data.
- Every remote frame should have a system-wide DLC which is the DLC of the corresponding data frame. Otherwise two modules starting transmission of a remote frame at the same time will overwrite each others DLC which results in bit errors as described above.

3.14 ACCEPTANCE FILTERING

Every node performs acceptance filtering on the identifier of a data or a remote frame to filter out the messages which are not required by the node. So only the data of frames which match the acceptance filter is stored in the corresponding data buffers.

However every node which is not in the bus-off state and has received a correct CRC-sequence acknowledges the frame.

3.15 BIT TIMING DEFINITION

3.15.2 Nominal Bit Rate

The nominal bit rate is the number of bits per second transmitted in the absence of resynchronization by an ideal transmitter.

The bit rate will adjust itself to that of the rest of the network system within the range of 20 Kbaud to 125 Kbaud.

3.15.2 Nominal Bit Time

Nominal Bit Time = 1/Nominal Bit Rate

3.15.3 Segments of Bit Time

The nominal bit time can be thought of as being divided into four non-overlapping time segments:

- Synchronization Segment
- Propagation Segment
- Phase Buffer Segment #1
- Phase Buffer Segment #2

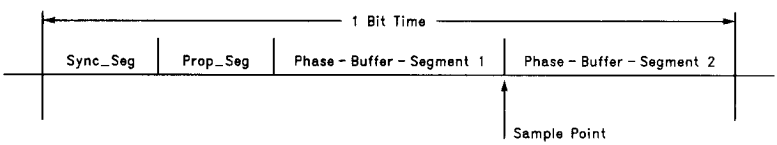


FIGURE 18. Segments of 1 Bit Time

3.0 Protocol Overview (Continued)

Synch Segment:

This part of the bit time is used to synchronize various modes on the bus. An edge is expected to lie within this segment.

Propagation Segment:

This part is used to compensate for the physical delay times within the network. This segment is twice the sum of the signal's propagation time on the bus line, the input comparator delay, and the output driver delay.

Phase Segments 1 and 2:

The Phase Segment Buffers are used to compensate for edge phase errors. These segments can be lengthened or shortened by synchronization.

The nominal bit time is internally adjusted to the bit time of the calibration message, provided that its bit time is within the specified range.

Sample Point:

The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. Its location is the end of Phase Segment.

Information Processing Time:

The information processing time is the time segment starting with the sample point reserved for the calculation of the subsequent bit level.

3.15.4 Time Quantum

The Time Quantum is a fixed unit of time derived from the oscillator period. There exists a programmable prescaler, with integral values, ranging at least from 1 to 32. Starting with the minimum Time Quantum, the Time Quantum can have a length of:

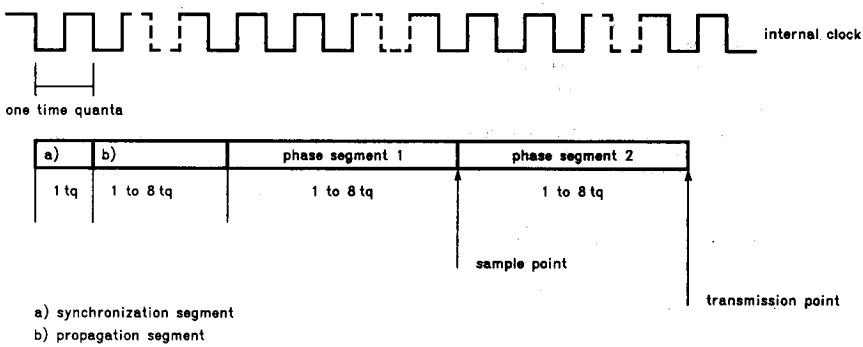
Time Quantum = m * minimum Time Quantum (m is the value of prescaler).

The length of each time segment is fixed for DS36001 and is given below.

Segment	Length in Time Quanta
Sync_Segment	1
Prop_Segment	1
Phase_Seg1	4
Phase_Seg2	4

3.15.5 Synchronization

Every receiver starts with a "hard synchronization" on the falling edge of the SOF bit. As stated before, one bit time consists of four time segments.



TL/F/11931–16

FIGURE 19. Bit Timing

3.0 Protocol Overview (Continued)

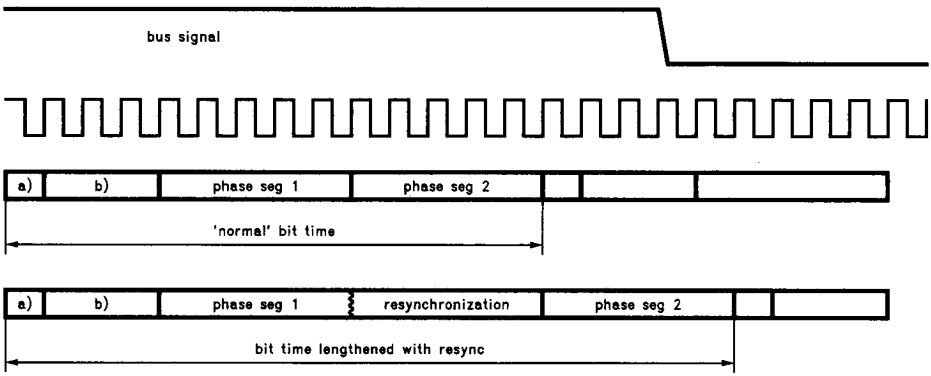
Either a rising or falling edge of the data signal should be in the synchronization segment. This segment has the fixed length of one time quanta. To compensate for the various delays within the network the propagation segment is used.

There are two types of synchronization supported:

"Hard synchronization" is done with the falling edge on the bus while the bus is idle, which is then interpreted as the SOF. It restarts the internal logic.

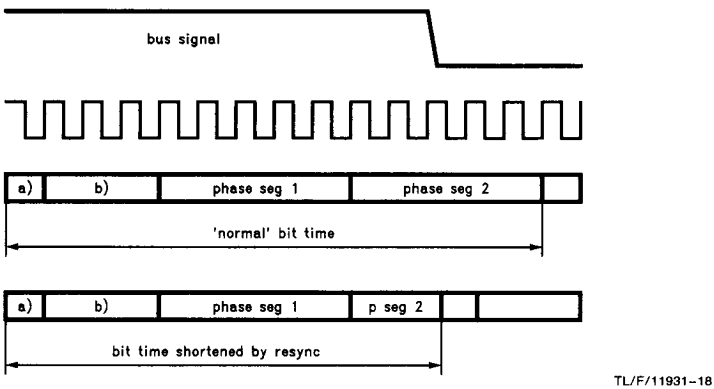
"Soft synchronization" is used to lengthen or shorten the bit time while a data or remote frame is received. Whenever a falling edge is detected in the propagation segment or in phase segment 1, the segment is lengthened by a specific value, the resynchronization jump width, see *Figure 20*.

If a falling edge lies in the phase segment 2 it is shortened by the resynchronization jump width. Only one resynchronization is allowed during one bit time. The sample point lies between the two phase segments and is the point where the received data is supposed to be valid. The transmission point lies at the end of phase segment 2 to start a new bit time with the synchronization segment.



TL/F/11931–17

FIGURE 20. Resynchronization 1



TL/F/11931–18

FIGURE 21. Resynchronization 2