SGS-THOMSON MICROELECTRONICS

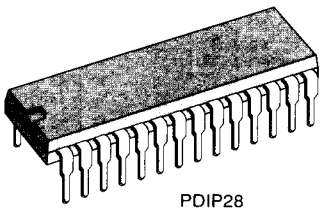
ST9560
ST9561

DATA LINK CONTROLLER FOR VEHICLE AREA NETWORK

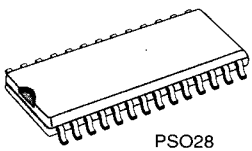
Developped with GIE PSA RENAULT

ADVANCE DATA

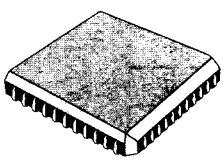
- Multimaster contention-based bus data link controller
- Fully compliant to VAN Specification ISO/TC22/SC3/WG1 Revision 4.0
- Programmable bit coding: MANCHESTER-E plus extensions to support MANCHESTER-L and pulse coding
- Supports all specified message types
- Data Field length: 0 to 30 bytes
- Programmable exchange speed up to 1.2 Mbits/s
- 16 maskable internal identifiers
- 160 bytes User RAM
- Bus access priority by message header buffer contents
- Error handling and line diagnosis
- VHDL model available
- Flexible microcontroller interface: Motorola, Intel, Z-bus, multiplexed and non-multiplexed
- Direct connection to ST9 microcontrollers
- 6 to 16 I/O lines in 44-pin version.
- 28 pin DIP and PSO (ST9560), 44 pin PLCC (ST9561) packages available



PDIP28



PSO28



PLCC44

(Ordering Information at the end of the Datasheet)

August 1992

Figure 1. ST9560 Pin Configuration

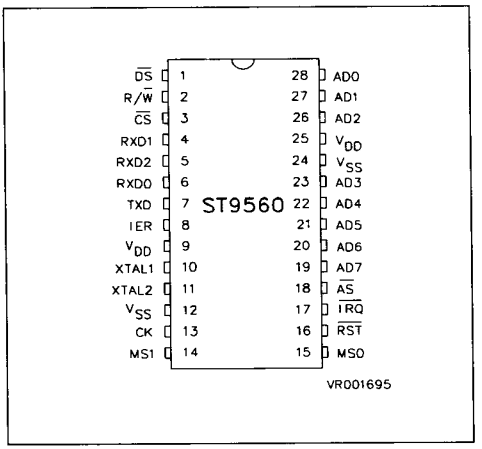


Figure 2. ST9561 Pin Configuration

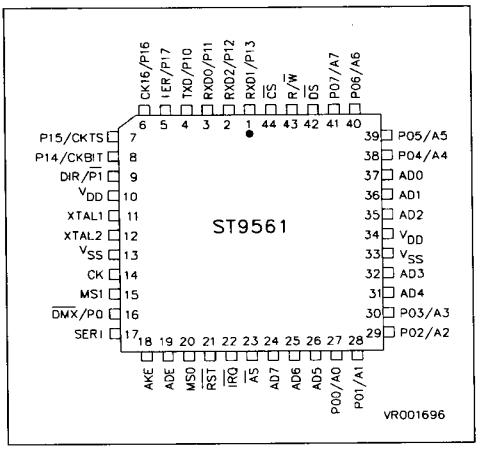
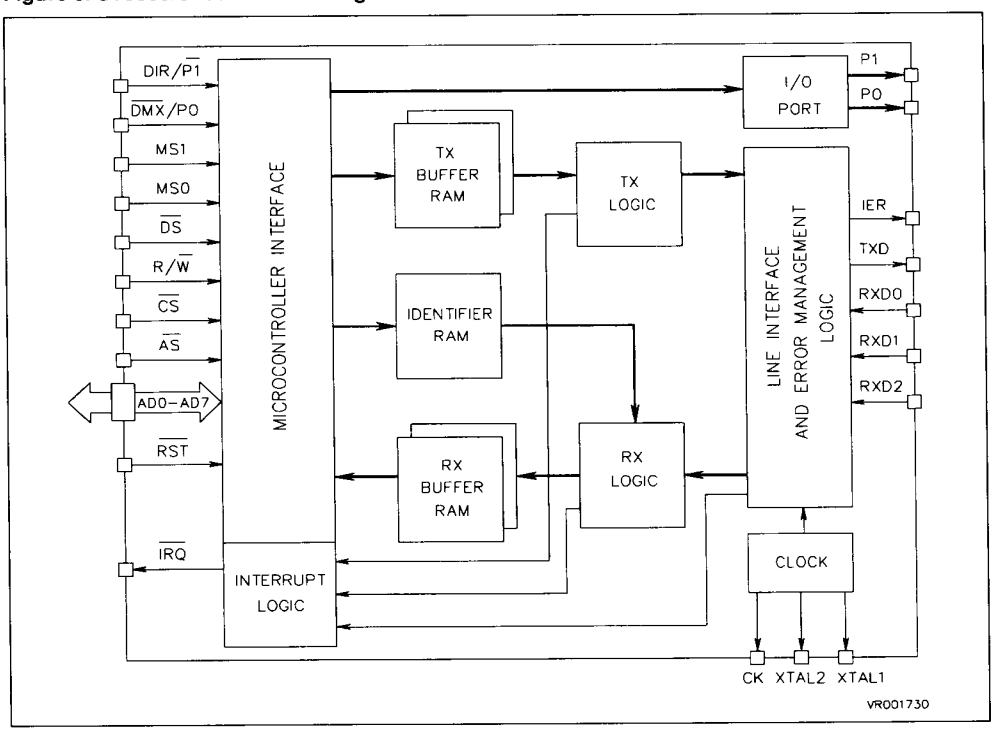


Figure 3. ST9560/ST9561 Block Diagram



GENERAL DESCRIPTION

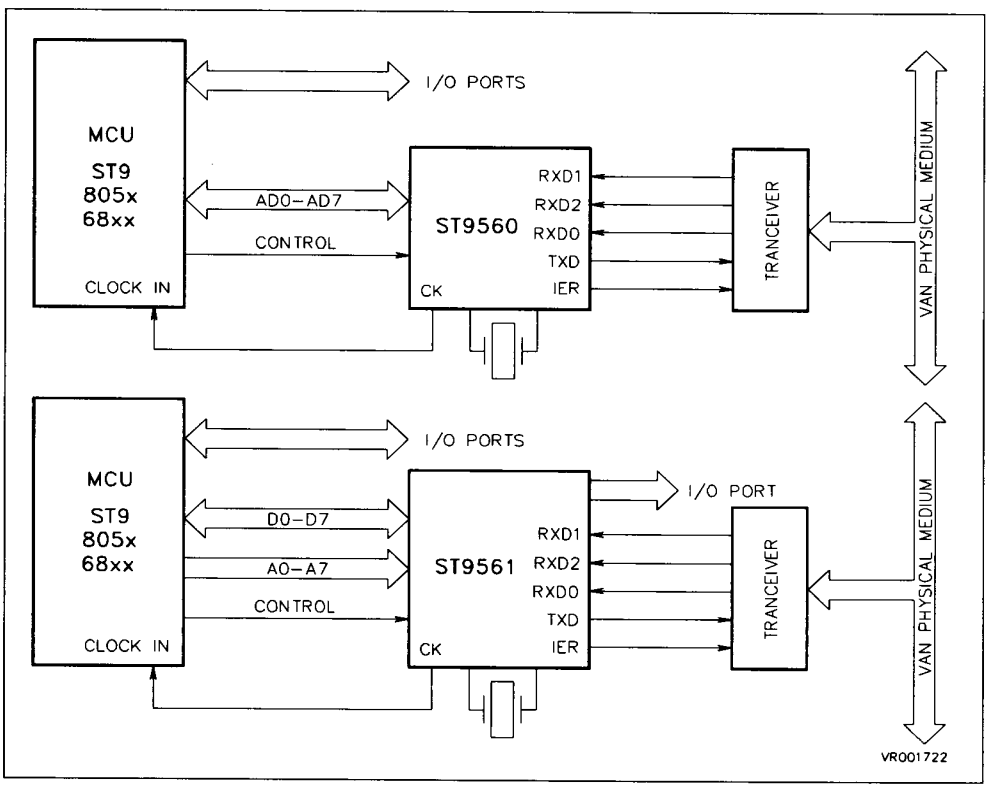
The ST9560 and ST9561 are integrated solutions to data-link control for the Vehicle Area Network (VAN) and are fully compliant to the standard ISO/TC22/SC3/WG1 Revision 4.0. Together with a microcontroller (MCU) and the appropriate line interface circuitry, the ST9560 and ST9561 provide complete functionality for the implementation of the VAN protocol in the DATA LINK LAYER and top level of the PHYSICAL layer of the OSI communications model.

The ST9560 manages the transmission and the reception of VAN messages and also error handling (CRC, code violation detection, message frame check) and diagnosis.

The ST9560 is the 28-pin stand alone version, in Plastic Dual in Line and the Plastic SO28 Wide package, whose MCU interface is tailored for microcontrollers such as ST9, ST10 or general purpose microcontrollers using a multiplexed 8-bit address/data bus. The ST9561 is the 44-pin PLCC version with a non-multiplexed bus supplementary capability and I/O port control.
As the devices are fully compatible in their functionality, from this point on all references to the ST9560 refer to both the ST9560 and ST9561 unless otherwise noted.

The host MCU communicates with the ST9560 through 4 32-byte static RAM buffers (one or two buffers for transmit data, 2 alternate use 32-byte receive buffers).

Figure 4. ST9560 and ST9561 VAN-DLC Application Example



PIN DESCRIPTION

V_{DD}. Main Power Supply Voltage (+5V ±10%)

V_{SS}. Digital Circuit Ground

XTAL1, XTAL2. These pins connect a parallel-resonant crystal (30MHz maximum), or an external source to the on-chip clock oscillator and buffer. XTAL1 is the input of the oscillator and internal clock generator or external clock input if used; XTAL2 is the output of the oscillator.

CK. Clock output (CK is XTAL1 buffered and is not affected by the $\overline{RST}$ input).

MS1, MS0. Microcontroller Bus Mode select inputs

MS1	MS0	MCU Bus Type
0	0	Motorola like
0	1	Intel like
1	0	Z-bus like (ST9)
1	1	Reserved

$\overline{CS}$. Chip Select input (active low)

R/$\overline{W}$. In Motorola and Z-bus modes, Read/Write input low = write from MCU). In Intel mode, Write Transfer synchronisation input (active low).

$\overline{DS}$. In Motorola and Z-bus modes, Data Strobe input. In Intel mode, Read Transfer synchronisation input (active low).

$\overline{AS}$. Address Strobe input (active low in Z-bus mode, active high in Motorola and Intel modes)

$\overline{RST}$. Reset input (active low)

$\overline{IRQ}$. Interrupt output (push-pull, active low)

AD0-AD7. Multiplexed Address/Data Bus.
Data bus in non-multiplexed mode of ST9561.

The ST9560 provides the following additional pins:

RxD2, RxD1, RxD0. VAN bus Serial inputs from receiver device.

TxD. Serial output.

IER. Reset output for the VAN bus driver device.

The following pins are available on the ST9561 only.

DIR/$\overline{P1}$. Port 1 configuration
When DIR/$\overline{P1}$ = '0', Port 1 is reconstructed.
When DIR/$\overline{P1}$ = '1', Port 1 takes the Alternate VAN interface functions.
This pin includes an internal pull up.

TXD/P10. Serial output

RXD0/P11. VAN bus serial input

RXD2/P12. VAN bus serial input

RXD1/P13. VAN bus serial input

CKBIT/P14. Sample clock bit of sampled data

CKTS/P15. Time slot Clock

CK16/P16. F_{base} Clock

IER/P17. Reset output for bus driver

$\overline{DMX}$/P0. Port 0 configuration
When $\overline{DMX}$/P0 = '0', Port 0 is used as address bus (non-multiplexed micro interface)
$\overline{DMX}$/P0 = '1': Port 1 is reconstructed
This pin includes an internal pull-up.

P00-P07. General Purpose 8-bit I/O port in standard mode, Address Bus I/O in non-multiplexed mode.

SERI. Serial output decoded VAN data.
(IDEN, COM, DATA)

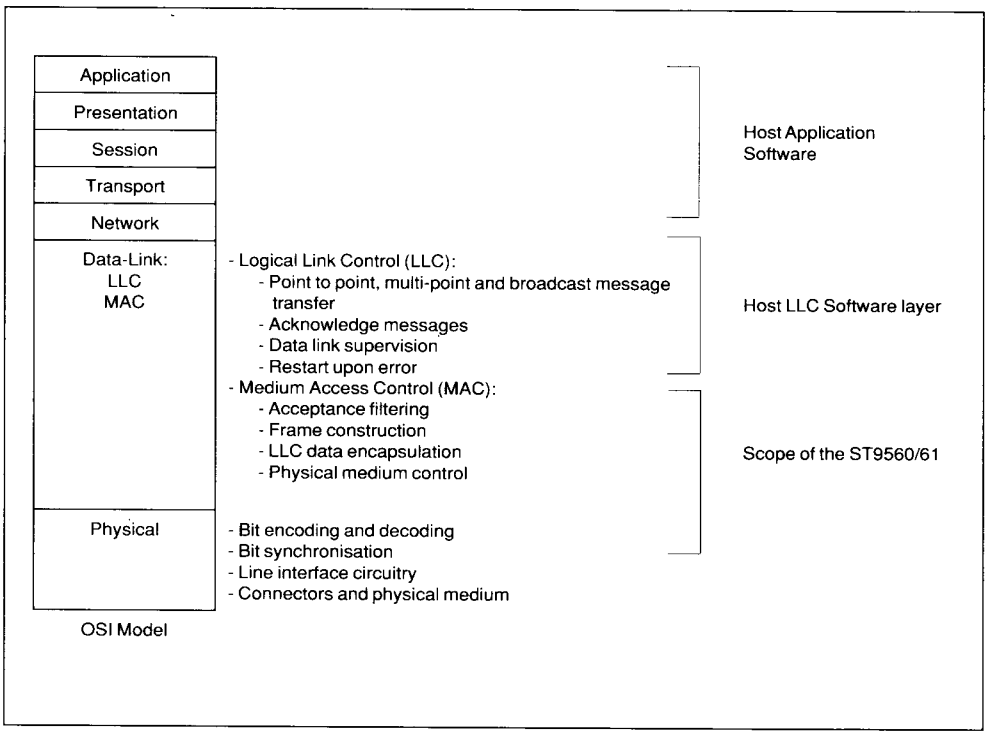
ADE. When bit MASK of register MRR is set and ADE = '1' , the frame passes the acceptance filter.

AKE. When bit MASK of register MRR is set and AKE = '1' , an acknowledge is sent.

O.S.I. MODEL AND ST9560/61

The Open System Interconnection model standardized by ISO is subdivided into 7 functionnal layers. The scope of the VAN standard, as submitted to ISO/TC22/SC3/WG1, is the Data Link and the Physical layers (see Figure 5). The Data Link layer is in turn subdivided into 2 sub-layers, the Medium Access Control (MAC) and the Logical Link Control (LLC). The ST9560/61 handles the MAC sub-layer and the upper part of the Physical layer (bit encoding/decoding). The LLC sub-layer and the upper layers of the OSI model are handled by the host processor software.

Figure 5. VAN description according to the OSI model and scope of the ST9560/61



FUNCTIONAL OVERVIEW

The ST9560

ST9560 is composed of nine major blocks:
(see block diagram)

- Memories (XBF0, XBF1, RBF0, RBF1, IDEN)
- Micro interface with interrupt logic
- Interface Management Logic (transmit and receive)
- Bit Stream Processor (transmit and receive) - Transceiver Control Logic
- Error Management Logic
- Bit Timing Logic and clock circuitry
- Diagnosis circuitry
- I/O ports.

Memories

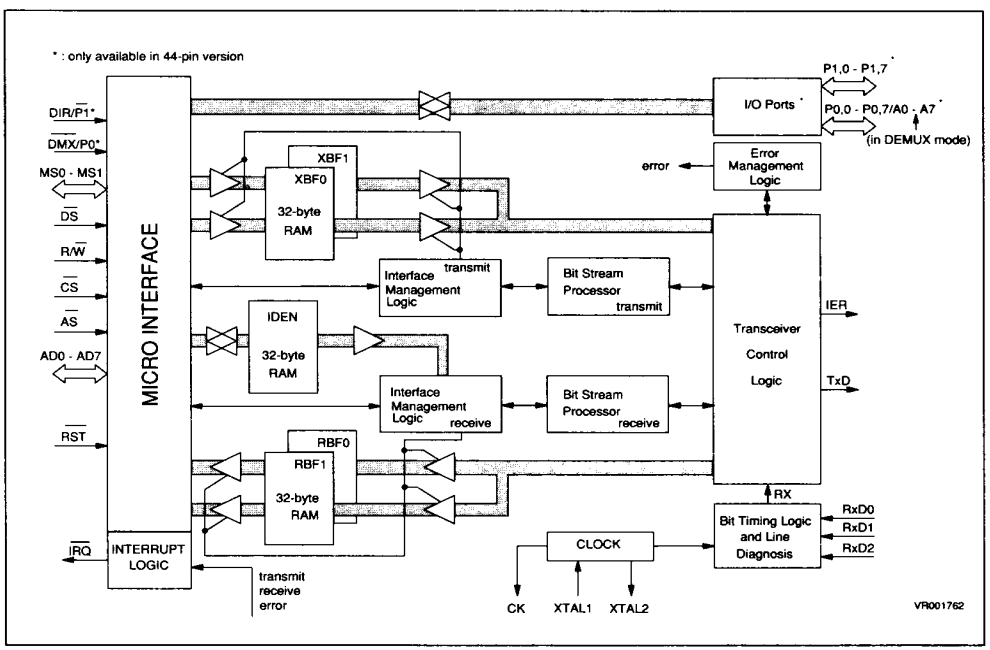
The host MCU stores the frames to transmit in a 32-byte static RAM used as a transmit buffer (XBF0 or XBF1 according to user choice). Received frames are loaded by the ST9560 in a double 32-byte static RAM (RBF0 and RBF1), transparently seen as only one buffer from the MCU interface (the user can select the first and the last transmit byte addresses, thus choosing the size of the data message).

In addition to DATA memories containing the frame buffers, a 32-byte RAM (IDEN) containing up to 16 identifiers is provided.

A set of 18 registers including status, command and identifier masks allows the host to manage the serial interface operation.

For the ST9561, another set of 6 registers allows the host the control of two 8-bit (maximum size) I/O ports.

Figure 6. Detailed Block Diagram



FUNCTIONAL OVERVIEW (Continued)

ST9560 Micro Interface

Two versions are provided:

- a 28-pin stand alone version (ST9560), whose micro interface is tailored for micro such as ST9, ST10 or general purpose microcontrollers using a multiplexed 8-bit address/data bus. The host communicates with ST9560 with 32-byte ram buffers (one or two buffers for transmit data, 2 alternate-use 32-byte receive buffers),
- a 44-pin version (ST9561) with non-multiplexed bus supplementary capability for host processors with non-multiplexed buses and I/O port control.

Direct interface to three popular MCU buses is configurable by two mode select lines, MS0 and MS1. These lines are to be hardwired to a logic one or zero in the application circuit to suit the bus configuration of the host.

Interface Management Logic

The interface management executes commands from the host and controls the data transfers on the serial bus.

Control, status and interrupt registers are used by the interface management.

Bit Stream Processor (BSP)

The bit stream processor controls the data stream between the interface (parallel data) and the serial bus.

A 32-byte identifier RAM contains up to 16 identifiers. When a message is received, the bit stream processor reads the identifiers loaded in RAM and performs the acceptance filter function. The identifier, command and data field are stored in RAM only if the message has an identifier which passes the acceptance filter.
The processor controls reception, arbitration, transmission and error signalling. The ST9560 cannot transmit in-frame responses.

Timing Logic

This block performs the synchronization functions according to the VAN specification. Each data bit is encoded and decoded according to its time slot representation. The rules of synchronization depend on the received signal.

Transceiver Control Logic

This block consists of bit coding/decoding according to the programmed type of encoding.

Error Management Logic

The different types of error (monitoring, CRC, code violation) are reported by the Bit Stream Processor and are passed to the interrupt management logic.

Diagnosis

This block monitors the states of the TXD and the three RxDi inputs (i = 0-2) and processes the sampled data in order to diagnose any potential VAN bus defects.
According to user choice (as under program control) degraded transmission modes can be chosen.

The three RxDi inputs provide different information on the VAN bus. RxD0 give the true differential logical information. RxD1 and RxD2 provide logical information from each of the two bus wires (using a threshold operation). The diagnosis logic performs the following functions.

- Digital filtering (on successive samples) in order to eliminate noise spikes.
- Transition analysis (synchronous, asynchronous protocol).
- Decision logic in order to select (automatically or not) a degraded mode (monofilar) or to detect that transmission is impossible and to go back to the normal differential mode using the TOP DIAG clock.

Control and status bits are provided to manage the operation of the diagnosis logic.

Clock Generator

The on-chip clock generator consists of an oscillator and clock divider register. The buffered output of the oscillator is passed through the CK pin. The optimum use of this signal is to drive the oscillator of the host MCU, reducing the need for a second oscillator circuit.

I/O Ports

The ST9561, 44-pin version, contains two additional standard 8-bit I/O ports.

Port 0 can be used as a standard I/O port when the multiplexed bus is used, allowing recreation of the host port used for the 9560 interface, or as the LSB byte address inputs A0-A7 if the non-multiplexed mode is selected.

Port 1 can be used as a standard 8-bit I/O port, or to output the VAN interface signals.

The ST9561 I/O ports are controlled via data registers (one per port) and configuration registers (two per port).

VAN OVERVIEW

In order to better understand the function of the ST9560, a short description of the relevant parts of the VAN protocol is provided here.

The VAN bus.

The VAN protocol implemented in the ST9560 is a superset of the standard ISO/TC22/SC3/WG1 version 4.0 and uses an access method which is multi-access based on non-destructive collision with bit-wise arbitration. All data is transferred inside frames.

Two consecutive frames are separated by an Inter-Frame Space. The ST9560 can send an in-frame request and process the in-frame response from another device, the ST9560 can not itself send an in-frame response.

VAN bus logical levels

The serial bus has two states: recessive and DOMINANT.

If two (or more) VAN modules send simultaneously, the resulting state of the bus is RECESSIVE only if all transmitters send recessive bits at the same time.

If one (or more) transmit dominant states, the resulting state of the bus will be DOMINANT.

Inside the ST9560 a dominant bit is represented by a logical '0' and a recessive bit by a logical '1'.

CODING / DECODING

Two types of encoding can be programmed by the host (Mode Control Register):

- MANCHESTER_L
- ENHANCED MANCHESTER (MANCHESTER-E)

Each of these two encoding modes can be combined with two time slot encoding modes:

- standard VAN (e.g. wire-based applications)
- pulsed (e.g. Optical based applications)

The base clock F_{base} is obtained from the local clock oscillator after user-programmable division.

The bit is encoded and decoded according to its time slot representation.

The Time Slot is the time duration of 16 F_{base} periods

TIME-SLOT in VAN STANDARD coding

In this type of coding a logical '0' is represented by a low state during the whole time-slot and a logical '1' by a high state during the whole time-slot.

TIME-SLOT in PULSE encoding:

In this type of coding, a logical '0' is represented by a low state during the first 1/8th of the time slot (2 periods of F_{base}).

A logical '1' is represented by a high state during the whole time slot.

Bit synchronisation

The synchronisation rule is based on the received signal edges and is the same for all the modules. During the sending of the preamble the synchronisation rule is disabled.

The 3 main points of time slot are:

- resynchronisation point (located at 2/16 of time slot duration in standard mode, at 4/16 of time slot duration in pulsed mode)
- sample point (located at 11/16 of time slot duration).
- transmit point.

Figure 7. Time slot reprsentation in VAN standard coding and pulse coding

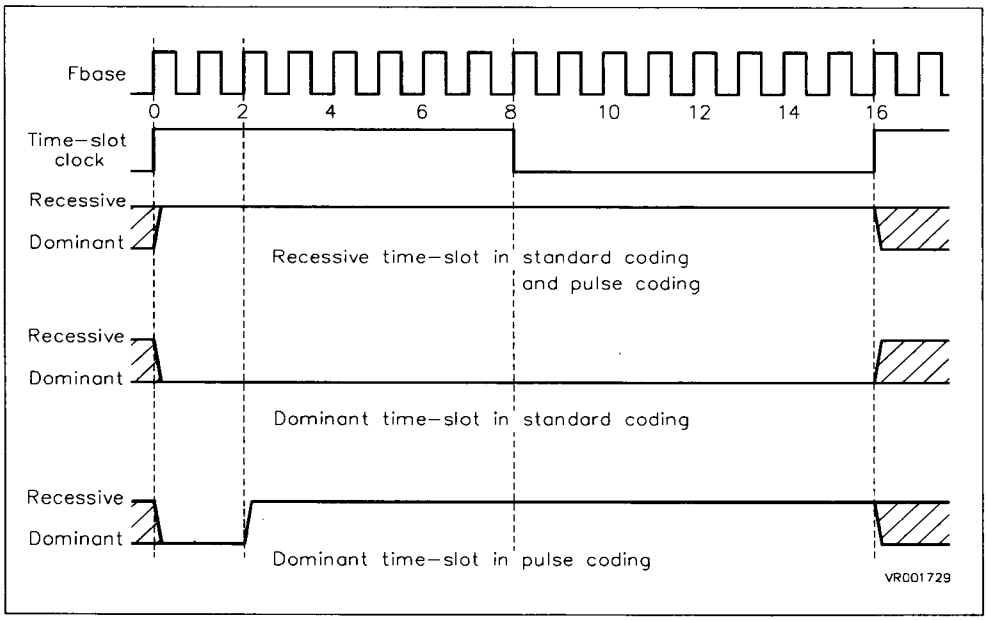
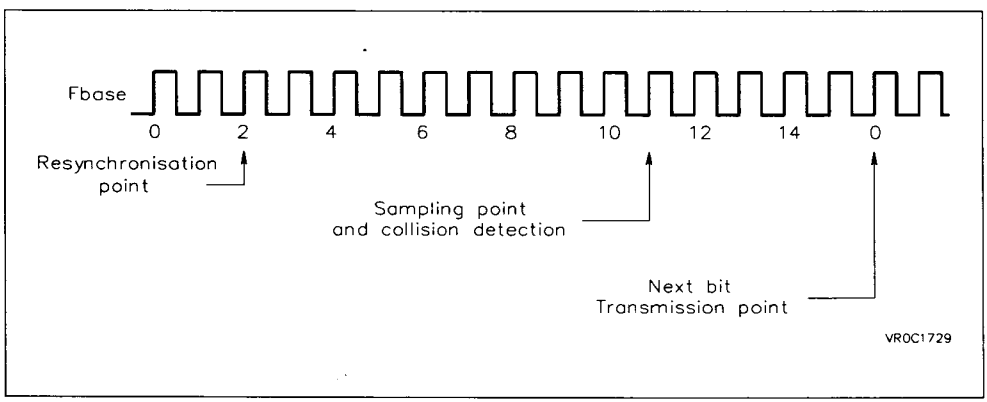


Figure 8. Time slot timing in VAN standard coding and pulse coding



VAN OVERVIEW (Continued)

Bit Representation

BIT REPRESENTATION in MANCHESTER_L:

- a logical '1' is coded as a '1' state during one time slot followed by a '0' state during one time slot.
- a logical '0' is coded as a '0' state during one time slot followed by a '1' state during one time slot.

As a consequence, 2 time slots are necessary to encode one bit.

BIT REPRESENTATION in MANCHESTER_E:

In this type of coding, bits are not coded individually but by nibble; 4 bits being encoded with 5 time slots.

- The three first bits are NRZ coded and the last bit of the nibble is MANCHESTER_L encoded.

As a consequence, a 1.25 time slot average is necessary to encode one bit.

Figure 9. Bit Representation in Manchester L

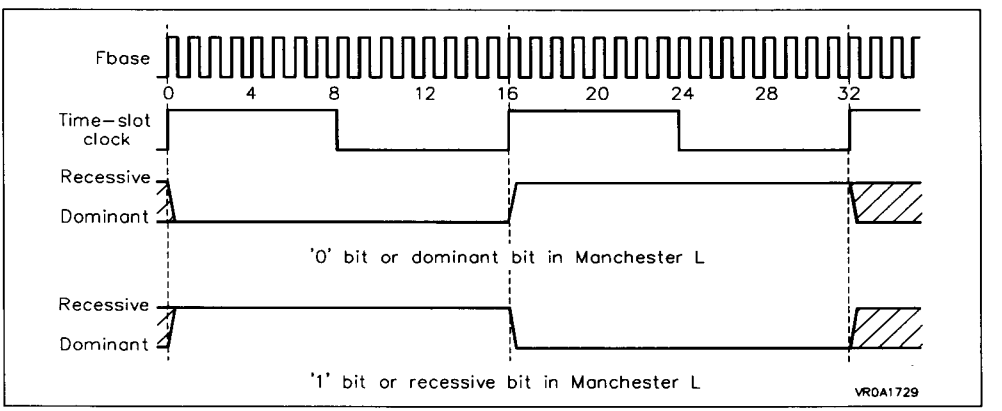
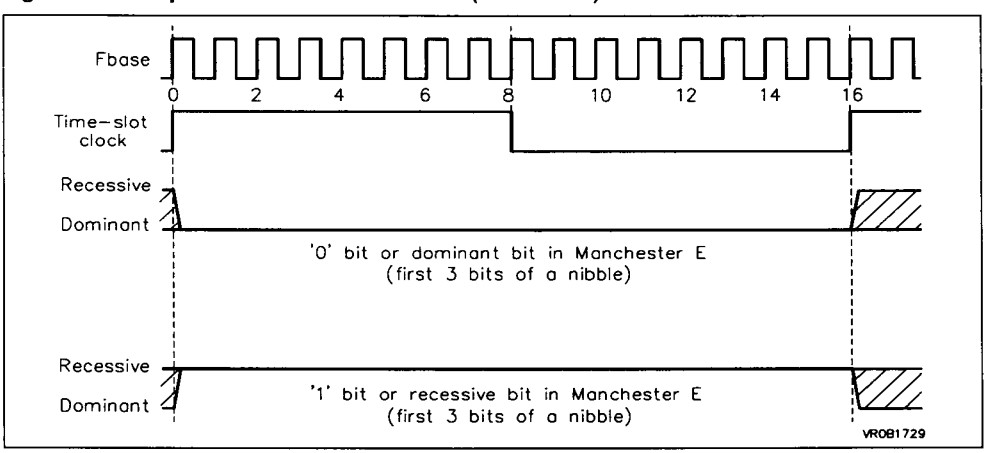


Figure 10. Bit Representation in Manchester E (NRZ Coded)



VAN OVERVIEW (Continued)

DESCRIPTION OF VAN FRAMES

Data Frames

Each frame is formed in a single format and is composed of following fields:

Start Of Message, Identifier, Command, Data, Frame Check Sequence, End Of Data, Acknowledge, End Of Message.

1) **SOF**: Start Of Frame

The Start Of Frame delimiter marks the beginning of the frame and allows synchronization for reception.

SOF is made of a preamble and a start bit. The preamble gives a common time reference for clock error confinement while the start bit initializes the frame.

SOF must be preceded by the IFS field (4 recessive time-slots).

The Preamble comprises of 4 dominant time slots followed by 4 recessive time slots.

The Start bit comprises of 1 dominant time slot followed by 1 recessive time slot.

2) **ID**: IDentifier: The identifier is a 12-bit field used to specify the frame's identification.

The identifier of a frame to be transmitted is provided by the host.

3) **COM**: COMmand:

This is a 4-bit field whose meaning is as follows:

Table 1. Command Field Description

	ML	ME	Meaning
EXT	1	1	Reserved for future extensions
RAK	1	1	acknowledge request
	0	0	acknowledge not requested
R/W	0	1	read request
	1	0	write request
RTR	0	0	data transmission
	1	1	remote transmission request

Note:
ML: MANCHESTER_L encoding
ME: MANCHESTER_Enhanced encoding

- EXTension: a reserved bit set to '1' (recessive state)
- RAK: Request AcKnowledge: this bit determines whether the sending module requests that a receiver module acknowledges successful reception of the frame.

RAK = '1': acknowledgement is requested

RAK = '0': acknowledgement is not requested

- R/W: this bit indicates whether the frame sent is a write or a read request
- RTR: Remote Transmit Request

If the bit is '0', the frame sent contains LLC-level data. If the bit is '1', the frame contains no data, and is interpreted as an in-frame request to send.

4) **DATA** field: the data field consists of a sequence having a whole number of bytes. The content of this field is provided by the LLC sublayer.

5) **FCS**: Frame Check Sequence

CRC (Cyclic Redundancy Check) is used in the receive and transmit procedures to detect any individual errors or packet errors.

The FCS field is a 15-bit field containing check bits.

The encoding is defined by the polynomial generator:

$$x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^4 + x^3 + x^2 + 1.$$

The CRC calculation includes the identification, command and data fields.

In transmit mode, the data bits are subjected to an encoding process, which is equivalent to division by the polynomial generator. The remainder obtained (ones complemented) is sent on the bus immediately after the data, in decreasing order of terms.

6) **EOD**: End Of Data

This field marks the end of the fields transmitted by the sender, i.e the following fields: identifier, command, LLC data, and FCS fields. It therefore determines the length of LLC data sent in the frame. This field is made of two consecutive dominant time slots.

7) **ACK** ACKnowledge

- positive acknowledge consists of one recessive time slot followed by a dominant time slot.
- absent acknowledge consists of two consecutive recessive time slots.

8) **EOF** End Of Frame marks the end of the frame.

It consists of eight consecutive recessive bits.

VAN OVERVIEW (Continued)

Figure 11. VAN frame structure

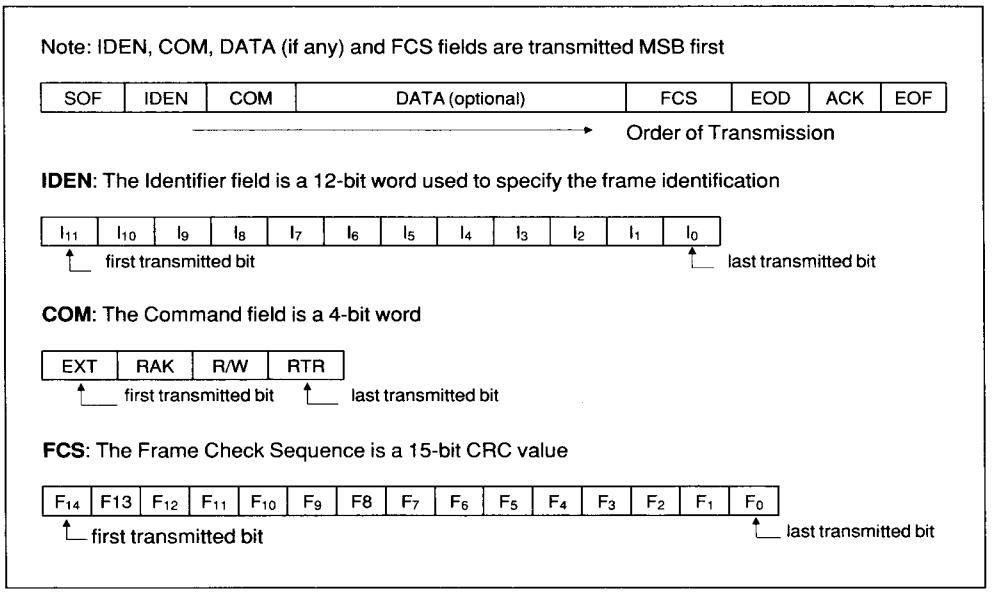
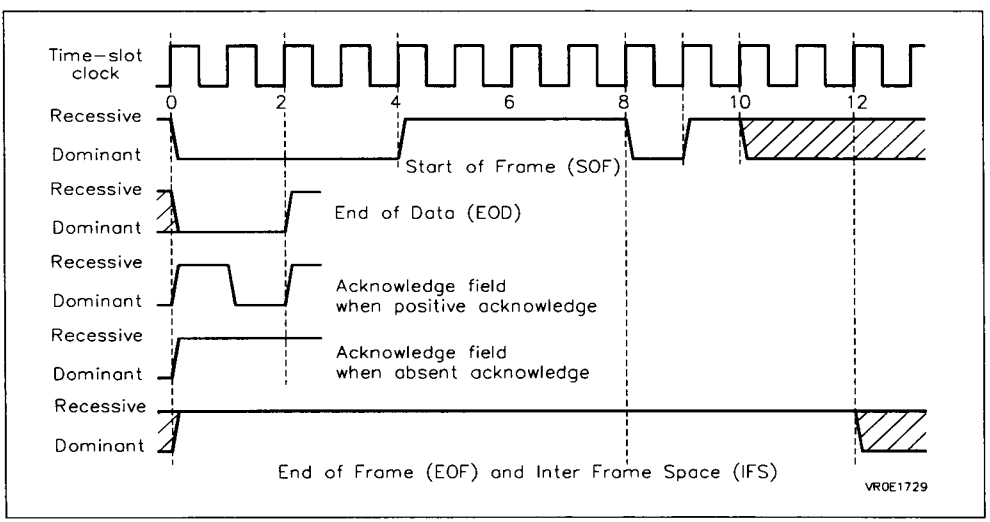


Figure 12. VAN Frame field signal coding



VAN OVERVIEW (Continued)

ARBITRATION

In the case where two or more ST9560s start transmission concurrently, the bus access conflict is solved by a bit-wise arbitration method. The transmit logic compares the bit level transmitted to the bit level monitored on the bus. The transmit logic stops transmission if a recessive bit was sent and a dominant bit was monitored. This method guarantees transmission of the message with the highest priority even if there is a collision without loss of time.

As a result the identifier not only indicates the destination of the message but also its priority.

In case of loss of arbitration, the ST9560 continues to monitor the bus and when the bus is free, it tries again to send the message.

The arbitration process described above covers the fields:

- IDEN, COM, DATA, FCS

The halting of transmission in the case of a loss of arbitration does not stop the reception process, allowing the in-frame response mechanism to operate.

Request/Response Frames

ST9560 is able to send request/response frames.

Such a frame is characterized by a data field of zero length, and by a command field for which the R/W and RTR bits have the following values:

R/W designates a read mode.
RTR is specified as remote request.

In the event of an in-frame response, the initiator sends the frame part including IDENTIFIER and COMMAND fields.

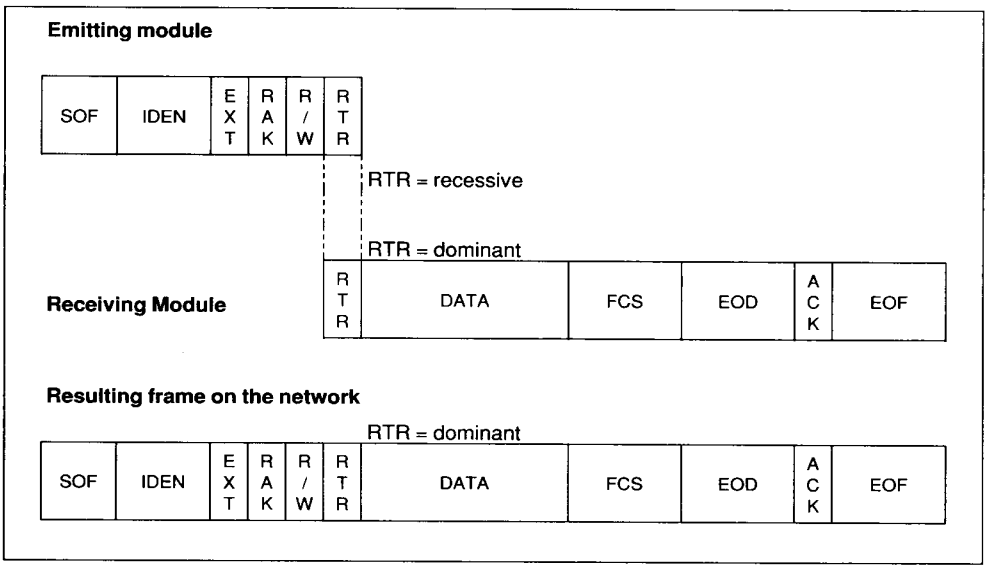
The response part of frame generated by the polled answering module, contains the following fields:

- a dominant RTR bit whose value is superimposed on the recessive bit of the request.
- a data field
- an FCS sequence

Superposition of the two parts of the frame (request and in-frame response) gives a frame on the bus which has a structure in compliance with the general frame structure.

The ST9560 cannot send in-frame responses.

Figure 13. In-frame response description



VAN OVERVIEW (Continued)

ERROR HANDLING

Error Detection

The ST9560 is equipped with several error mechanisms, implemented in hardware for efficiency

Note: Receive errors are reported only if the frame identifier is accepted.

Coherence of Reception Error

This error corresponds to the detection of an incoherence between the three receive inputs.

Code Violation Error

An error is detected on one of the bits of the binary or data fields of the frame (IDEN, COM, DAT, FCS).

This type of error occurs when the time between two transitions is less than 12/16 of time slot duration.

If such an error is detected during transmission, then bit TCVL of the Error Transmit Register is set.

If such an error is detected during reception of a message having passed the acceptance filter, bit RCVL of the Error Receive Register is set.

CRC Error

A Cyclic Redundancy Check is used in the transmit and receive procedures to detect any individual or packet errors in the messages transmitted.

The FCS is a 15-bit field, the remainder of the division by the generator polynomial:

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

The CRC calculation is made on the fields:

- IDEN, COM, DATA

In transmit mode the FCS field is sent onto the line immediately after the data bits.

The receiver computes the CRC again. If both CRC, transmitted and computed by the receiver are different, an error has occured.

If such an error is detected during transmission, bit TCRC of the Error Transmit Register is set.

If such an error is detected during reception of a message having passed the acceptance filter, bit TCRC of the Error Receive Register is set.

DIAGNOSIS

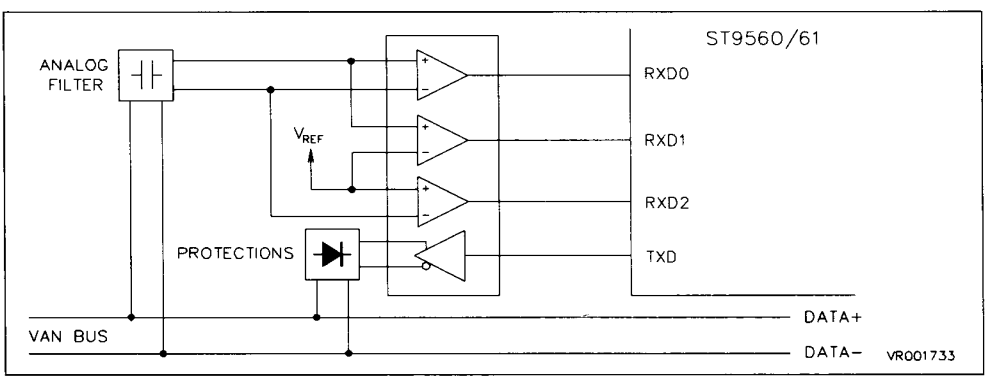
The three RxDi inputs provide three different information states on the bus. RxD0 gives the true differential logical information. RxD1 and RxD2 provide logical information from each of the two bus wires (with a threshold operation).

The diagnosis block logic performs the following functions:

- digital filtering (on 5 successive samples) in order to eliminate spikes
- transition analysis (synchronous, asynchronous, transmit, protocol)
- decision logic in order to select (automatically or not) a degraded mode (monofilar) or to detect that transmission is impossible and when to go back to the normal differential mode with the help of the TOP DIAG 'clock'.

Control and status bits are provided to manage the operation of the diagnosis logic.

Figure 14. Line-Interface Diagnosis Schematic



ST9560 MICRO INTERFACE

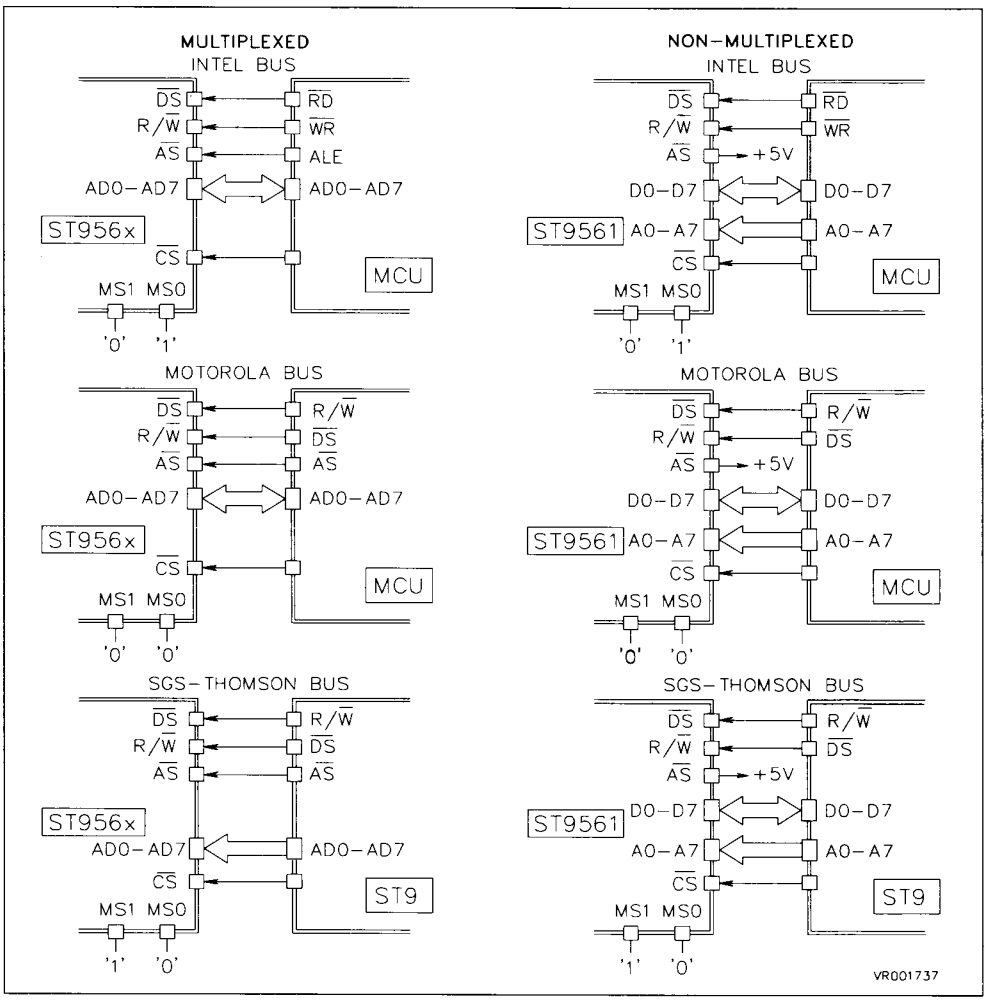
Two ST9560 versions are provided, both having the same VAN controller core and different micro interface.

A first version, a 28-pin one, is the stand alone version which is able to interface with different microprocessors such as ST9 and other general purpose microcontrollers such as Motorola 68HC11 or Intel 8051.

It uses an 8-bit address/data bus multiplexed and several control lines (ALE, $\overline{RD}$ or $\overline{DS}$, $\overline{WR}$ or R/$\overline{W}$, $\overline{CS}$) and system lines ($\overline{RST}$, $\overline{IRQ}$).

A second version, a 44-pin one, is also available offering a multiplexed/non-multiplexed bus selection. The remaining pins are used as standard I/O pins which can be configured as standard I/O's or alternate function outputs.

Figure 15. Micro Interface Schematics



FUNCTIONAL DESCRIPTION

Memory Mapping

The 160 bytes of on-chip RAM is divided into three areas:

- 2 x 32-byte receive buffers. Both of these buffers are located at the same logical addresses from address 32 (20h) to address 63 (3Fh).
- 2 x 32-byte transmit buffers. The buffers are located from address 64 (40h) to address 95 (5Fh) for transmit buffer 1 and from address 96 (60h) to 127 (7Fh) for transmit buffer 0. Only one buffer is used at any time and is user selectable. This allows the precreation of certain messages, perhaps error or warning frames, held in one transmit buffer until needed, the second buffer being used for normal transmit functions.
- 32-byte identifier buffer. This buffer is located at the same addresses as the receive buffers.

Command and status registers are located from address 0 to 19 (13h).
Locations 20 to 23 are not used (13h to 17h).

The locations 24 to 31 hold the I/O port registers of the ST9561 (14h to 1Fh). These addresses must not be used in the ST9560.

ST9560 Memory mapping

Function	Address (hex)
Transmit Buffer 0 XBF0	7F 60
Transmit Buffer 1 XBF1	5F 40
Identifiers/ Receive Buffer IBUF/RBUF0/1	3F 20
I/O Control Registers (ST9561 only)	1F 18
Reserved	17 14
Control/Status Registers	13 00

Control and Status Registers

18 registers, located at lowest addresses, both command and status, are provided for management of ST9560:

All unused address locations are reserved for future use.

Table 2. Registers

Register/Function		Address
General Purpose		
System Control Register	SCR	00h
Mode Control Register	MCR	01h
Physical Control Register	PCR	04h
Interrupt registers		
Enable Interrupt Register	EIR	05h
Status Interrupt Register	SIR	06h
Reset Interrupt Register	RIR	07h
Receive Registers		
Mode Received Register	MRR	08h
Status Received Register 1	SRR1	09h
Status Received Register 2	SRR2	0Ah
Error Received Register	ERR	0Bh
Mask Registers		
MSB of Mask 0	MSBM0	0Ch
LSB of Mask 0	LSBM0	0Dh
MSB of Mask 1	MSBM1	0Eh
LSB of Mask 1	LSBM1	0Fh
Transmit Registers		
Mode Transmit Register	MTR	10h
Command Transmit Register	CTR	11h
Status Transmit Register	STR	12h
Error Transmit Register	ETR	13h

FUNCTIONAL DECRIPTION (Continued)

SCR System Control Register
address 00h (read/write register)

7							0
ABORT	RESET	RES	RES	M3	M2	M1	M0

b7 = **ABORT**. This bit has no action when = '0', when set to '1' transmission is stopped after the end of the current transmission if transmitting a frame or immediately if not. This bit is identical in function to ABORT1 in CTR and is logically OR'ed with it.

b6 = **RESET**. Setting this level-sensitive bit to '1' generates an internal hardware reset of the ST9560, equivalent to the low state on the hardware $\overline{RST}$ pin.

All bits of control registers are set to '0', with the exception of the INIT and TI interrupts which must be cleared before starting initalisation and communication. The contents of the RAM buffers are undefined and should be initialised by the host MCU. P0C1, P1C1, P0D and P1D registers are set to "FFh".
This bit has no action when = '0'.

b5-b4 = *RESERVED*

b3-b0 = **M3,M2,M1,M0.** *Mode select* The bits in this field set the type of coding used for transmission and reception as shown in table 4.
The VAN standard coding is intended for wire transmission and the pulse coding is adapted to fibre-optic transmission.

NOTE: The reserved bits of read/write or write-only registers should be forced to 0. These bits are shown as *RESERVED* in the register descriptions.

MCR Mode Control Register
address 01h (read/write register)

7							0
PRESC	CD2	CD1	CD0	TDDIV2	TDDIV1	TDDIV0	TDIAG

b7 = **PRESC**. *Prescaler* When this bit is '0', the 1/3 prescaler is disabled, setting this bit to '1' enables the 1/3 prescaler (see Figure 21).

b6-b4 = **CD2, CD1, CD0**. *Clock Divider* These 3 bits set the division ratio of the oscillator clock divider to generate the F_{BASE} clock. They are coded as $\log_2$ (CD2-CD0) as shown in Table 3.

b3-b1 = **TDDIV2-TDDIV0**. *Top Diag Divider.* These 3 bits are written by the host to define the period of the TOP DIAG clock used in line diagnosis. The bit clock is divided in order to generate TOP DIAG clock.
The division rate depends of TDDIV2, TDDIV1, TDDIV0 value as shown in Table 5.

b0 = **TDIAG.** *Top Diag* When TOP DIAG is set to '1' by the host, a pulse is generated into the line diagnosis part. This pulse is used to generate an internal clock (FMS). In a period between 2 TOP DIAG pulses, every module able to transmit a frame should have transmitted a minimum of one frame. If no response is detected within this time, the network may have a major problem.

Table 3. Clock Divider Ratio (CDDIV)

CD2	CD1	CD0	Divider
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

FUNCTIONAL DECRIPTION (Continued)

Table 4. Line Interface Mode select

M3	M2	M1	M0	Type Of Coding	Typical Line Interface
0	0	0	0	E-MANCHESTER (standard)	Wire
0	0	0	1	L-MANCHESTER (standard)	Wire
0	0	1	0	E-MANCHESTER (pulse)	Fibre Optic
0	0	1	1	L-MANCHESTER (pulse)	Fibre Optic

Note: All other combinations are reserved for future types of communication.

Table 5. Top Diagnostic Clock divider

TDDIV2	TDDIV1	TDDIV0	TOP DIAG Clock (1)	Frequency Division
0	0	0	NO CLOCK(2)	-
1	0	0	100ms	8.192k
0	1	0	200ms	16.38k
1	1	0	400ms	32.77k
0	0	1	800ms	65.54k
1	0	1	1.6s	131.1k
0	1	1	3.2s	262.1k
1	1	1	6.4s	524.3k

Notes: 1. These periods are calculated for E-MANCHESTER coding with a 100 kT/s rate
2. Direct TOP DIAG Pulse

FUNCTIONAL DECRIPTION (Continued)

PCR Physical layer Control Register
address 04h (read/write register)

7							0
TPOL	RPOL	IER	RES	RES	RES	RX1	RX0

b7 = **TPOL**. *Transmit Polarity* In order to accomodate line interfaces which change the polarity of the transmission, this bit allows the automatic inversion of the data transmitted by the TxD pin.
When TPOL = '0' TxD has standard polarity (dominant = '0', recessive = '1'), when TPOL = '1' TxD has the inverse polarity (dominant = '1', recessive = '0').

b6 = **RPOL**. *Receive Polarity* This bit controls the polarity of the three RxDi inputs.
When RPOL = '0' the RxDi inputs have direct polarity, when RPOL = '1' they have inverse polarity.

b5 = **IER**. *Line Tranceiver Reset* This bit sets the logic state of the output pin IER. It may be used as the reset signal to the external line circuitry.

b4-b2 = *RESERVED*

b1-b0 = **RX1, RX0**. *Receiver Input Select*

RX1	RX0	Function
0	0	RxD0 selected as serial input
0	1	RxD1 selected as serial input
1	0	RxD2 selected as serial input
1	1	automatic selection of serial input

EIR Enable Interrupt Register
address 05h (read/write register)

7							0
GIE	RES	FTIE	RES	TIE	FRIE	OVIE	RIE

All bits are active high to enable the interrupt.

b7 = **GIE**. Global Interrupt Enable. All interrupt enable bits are OR'ed with GIE.

b6 = *RESERVED*

b5 = **FTIE**. Failed Transmit Interrupt Enable

b4 = *RESERVED*

b3 = **TBEIE**. Transmit Interrupt Enable

b2 = **FRIE**. Failed Receive Interrupt Enable

b1 = **OVIE**. Overrun Interrupt Enable

b0 = **RIE**. Receiver Interrupt Enable

SIR Status Interrupt Register
address 06h (read-only register)

7							0
IRQ	INIT	FTI	RES	TBEI	FRI	OVI	RDAI

All bits are active high. To clear a set interrupt status bit, it is necessary to write a '1' to the corresponding bit in the Reset Interrupt Register RIR.

b7 = **IRQ**. Interrupt (logical OR of the 6 interrupt sources)

b6 = **INIT**. *Initialisation Interrupt*. This bit is set to '1' on RESET (internal or external). It must be cleared by writing to the corresponding bit in RIR at the beginning of the host initialisation of the ST9560.

b5 = **FTI**. *Failed Transmit Interrupt*. An error in a frame being transmitted will set this bit. This error can be:

- discrepancy between RxD0, RxD1 and RxD2,
- CRC error,
- code violation

as indicated in Error Transmit Register ETR.

b4 = *RESERVED*

b3 = **TBEI**. *Transmit Buffer Empty Interrupt* After a frame has been successfully transmitted, this bit is set to indicate completion and the transmit buffer selected is available for another message.
This bit is set to '1' on RESET (internal or external). It must be cleared by writing to the corresponding bit in RIR at the beginning of the host initialisation of the ST9560.

b2 = **FRI**. *Failed Receive*. An error causing a failure to make a valid reception of a frame will set this bit. The specific errors are flagged in the Error Receive Register ERR.

b1 = **OVI**. *Overrun*. This flag is set when the 2 receive buffers are filled with accepted incoming frames and a third incoming frame has passed the acceptance filter before the host has released a receive buffer.

b0 = **RDAI**. *Receive Data Available*. This bit is set when an accepted incoming frame is available in a receive buffer. To release the buffer the host has to read the message contents and to clear RDAI by setting the RDAIC bit of the Reset Interrupt Register RIR to '1'.

FUNCTIONAL DECRIPTION (Continued)

RIR Reset Interrupt Register
address 07h (write-only register)

7							0
RES	INITC	FTIC	RES	TIC	FRIC	OVIC	RDAIC

Interrupts are cleared one F_{BASE} cycle time after the corresponding bit in this register is set to '1'.

b7 = *RESERVED*

b6 = **INITC**. *Clear Init Interrupt*

b5 = **FTIC**. *Clear Failed Transmit Interrupt*

b4 = *RESERVED*

b3 = **TIC**. *Clear Transmit Interrupt*

b2 = **FRIC**. *Clear Fail Receive Interrupt*

b1 = **OVIC**. *Clear Overrun Interrupt*

b0 = **RDAIC**. *Clear Receive Data Available Interrupt*

MMR Mode Receive Register
address 08h (read/write register)

7							0
BA	RES	MF	RES	RES	RES	RES	RES

b7 = **BA.** *Buffer Access* This bit controls the host access to the two parallel 32byte RAM areas holding the identifiers and the receive buffer between addresses 20h and 2Fh as shown in Table 6.

b6 = *RESERVED*

b5 = **MF**. *Mask Filter* If an external acceptance identifier filter is used with the additional signals ADE and AKE available with the ST9561, this bit disables the internal filtering.

b5 = '0' normal mode
b5 = '1' every identifier on the VAN bus is recognized by the ST9561 when ADE = '1' (frames with RAK = '1' correctly received are acknowledged)

Note: The ADE input is not externally available for the ST9560 and the ADE input is always pulled high, thus setting the MF bit of the ST9560 will cause all incoming frames to be accepted.

b4-b0 = *RESERVED*

SRR1 Status Receive Register 1
address 09h (read-only register)

7							0
RDA	RES	RES	RES	IN3	IN2	IN1	IN0

b7 = **RDA**. *Receive Data Available* When set to '1', this bit indicates a received frame is available in the receive buffer.

b6-b4 = *RESERVED*

b3-b0 = **IN3-IN0**. *Identifier Number* (b3 = MSB) Filtered identifier index number into the Identifier RAM. This value is used by the host MCU as the index to the identifier in the RAM which has matched (after masking) the identifier field of the incoming frame. The identifier number is based upon the user allocation of the 16 identifiers in the Identifier RAM.

SRR2 Status Receive Register 2
address 0Ah (read-only register)

7							0
RDA1	RES	RES	LRBP4	LRBP3	LRBP2	LRBP1	LRBP0

b7 = **RDA1** This bit has the same meaning as RDA in SRR1.

b6-b5 = *RESERVED*

b4-b0 = **LRBP4-LRBP0**. *Last Received Byte Pointer* (b4 = MSB) This field is programmed by the ST9560 to indicate the address of the last byte received in the receive buffer. This provides the message length.

Table 6. Buffer Access Bit

BA	Host Read 20h-2Fh	Host Write 20h-2Fh
0	Receive buffers	Identifiers
1	Identifiers	Identifiers

FUNCTIONAL DECRIPTION (Continued)

ERR Error Received Register
address 0Bh (read-only register)

7							0
RF	OVR	RCVL	RES	RCRC	RDS1	RDS0	RVP

All bits of this register are active high. They are reset when clearing the Failed Receive Interrupt (FRI of SIR) by writing a '1' to FRIC of RIR.

b7 = **RF**. *Receive Failure* This bit indicates (when set) that the last reception has failed, causing a Failed Receive interrupt request.

b6 = **OVR**. *Overrun* This flag is set when the 2 receive buffers are filled with accepted incoming frames and a third incoming frame has passed the acceptance filter before the host has released a receive buffer. This is physically the same bit as OVI in SIR.

b5 = **RCVL**. *Receive Code Violation* This bit indicates (when set) that a code violation has been detected during reception. A code violation can be an incorrect SOF pattern or an incorrect Manchester bit.

b4 = *RESERVED*

b3 = **RCRC**. *Receive CRC Error* This bit indicates (when set) that a CRC error has been detected during reception.

b2-b1 = **RDS1, RDS0** *Receive Data Status*

RDS2	RDS1	Function
0	0	RxD0 selected
0	1	RxD2 selected
1	0	RxD1 selected
1	1	Major Error, No communication

b0 = **RVP** *Receive Physical Violation* This bit indicates (when set) that an incoherence has been detected between RxD0, RxD1 and RxD2 during reception.

MSBM0 MSB Mask Register 0
address 0Ch (read/write register)

7							0
MSK0.11	MSK0.10	MSK0.9	MSK0.8	MSK0.7	MSK0.6	MSK0.5	MSK0.4

b7-b0 = **MSK0.11-MSK0.4** *MSB of MASK0*

LSBM0 LSB Mask Register 0
address 0Dh (read/write register)

7							0
MSK0.3	MSK0.2	MSK0.1	MSK0.0	RES	RES	RES	RES

b7-b4 = **MSK0.3-MSK0.0** *LS Bits of MASK 0*

b3-0 = *RESERVED*

Each bit set to '1' in Mask register 0 enables the comparison of the corresponding bit of the received message identifier to the corresponding bits in the identifiers in the identifier RAM when the ST9560 performs the internal message acceptance filtering function.
Mask 0 is selected by bits RCM1 = '0' and RCM0 = '1' for each identifier.

MSBM1 MSB Mask Register 1
address 0Eh (read/write register)

7							0
MSK1.11	MSK1.10	MSK1.9	MSK1.8	MSK1.7	MSK1.6	MSK1.5	MSK1.4

b7-b0 = **MSK1.11-MSK1.4**. *MSB of MASK 1*

LSBM1 LSB Mask REGISTER 1
address 0Fh (read/write register)

7							0
MSK1.3	MSK1.2	MSK1.1	MSK1.0	RES	RES	RES	RES

b7-b4 = **MSK1.3-MSK1.0**. *LS Bits of MASK 1*

b3-b0 = *RESERVED*

Each bit set to '1' in Mask register 1 enables the comparison of the corresponding bit of the received message identifier to the corresponding bits in the identifiers in the identifier RAM when the ST9560 performs the internal message acceptance filtering function.
Mask 1 is selected by bits RCM1 = '1' and RCM0 = '0' for each identifier.

FUNCTIONAL DECRIPTION (Continued)

MTR Mode Transmit Register
address 10h (read/write register)

7							0
RANK	RES	BUF	FTB4	FTB3	FTB2	FTB1	FTB0

b7 = **RANK**. *VAN Rank Select* When the ST9560 is configured in the with RANK = '0', it is a VAN autonomous module and can transmit (if the RTS bit is set) as soon as the bus is free.
When the ST9560 is configured with RANK = '1', it is a VAN synchronous access module and can transmit only if a Start Of Frame generated by another module is detected on the network.

b6 = *RESERVED*

b5 = **BUF**. *Transmit Buffer Select* When BUF = '0', Transmit buffer 0 (XBF0) is used
When BUF = '1', Transmit buffer 1 (XBF1) is used

b4-b0 = **FTBP4-FTBP0**. *First Transmit Byte Pointer* This field is programmed by the host MCU to select the first byte to transmit in the selected Transmit buffer. Once transmission commences, the data to send will be taken from the buffer at incrementing addresses until the address is equal to the Last Transmit Byte pointer in the CTR register.

CTR Control Transmit Register
address 11h (read/write register)

7							0
RTS	ABORT1	RES	LTB4	LTB3	LTB2	LTB1	LTB0

b7 = **RTS**. *Request To Send* When RTS is set by the host, the ST9560 commences the transmission of a frame according to the setting of MTR and CTR.

b6 = **ABORT1**. *Abort Transmission* When ABORT1 is set to '1' by the host, it stops transmission after the end of the current transmission if transmitting a frame, immediately if not. This is identical in function to the ABORT bit of SCR.

b5 = *RESERVED*

b4-b0 = **LTBP4-LTBP0**. *Last Transmit Byte Pointer* b4 is MSB. This field is programmed by the host MCU to select the last byte to transmit in the selected Transmit buffer. Once transmission commences, the data to send will be taken from the buffer at incrementing addresses, starting from the address programmed into FTBR, until the address is equal to the Last Transmit Byte pointer in the CTR register.

FUNCTIONAL DECRIPTION (Continued)

STR Status Transmit Register
address 12h (read-only register)

7							0
TBA	RES	SBUF	CTB4	CTB3	CTB2	CTB1	CTB0

b7 = **TBA** *Transmit Buffer Available*. The host MCU may write into the XBF selected by the BUF bit of MTR when TBA = '1'.
When TBA = '0' the host must not write to the selected XBF as it is being used by the transmit logic.

TBA is set to '1' when the buffer has been transmitted successfully (TI interrupt) and is reset to '0' when the host starts a transmission by setting the relevant parameters in CTR and MTR.

b6 = *RESERVED*

b5 = **SBUF** *Transmit Buffer Selected* The Status information is associated to XBF selected by the BUF bit of MTR.
When SBUF = '0' the status is associated to XBF0.
When SBUF = '1' the status is associated to XBF1.

b4-b0 = **CTBP4-CTBP0**. *Current Transmit Byte Pointer* b4 is MSB. This field is programmed by the ST9560 to indicate the address within the transmit buffer of the current byte being transmitted.

ETR Error Transmit Register
address 13h (read-only register)

7							0
FTA	RES	TCVL	RES	TCRC	TDS1	TDS0	TVP

All bits in this register are active high. They are reset when clearing the Failed Transmit Interrupt (FTI of SIR) by writing a '1' to FTIC of RIR.

b7 = **FTA**. *Failed Transmission Alert* This bit indicates (when set) that last transmission has failed, causing a Failed Transmit interrupt request.

b6 = *RESERVED*

b5 = **TCVL**. *Transmit Code Violation* This bit indicates (when set) that a code violation has occured during transmission of the XBF in use. A code violation can be an incorrect SOF pattern or an incorrect Manchester bit.

b4 = *RESERVED*

bit 3 = **TCRC**. *Transmit CRC Error* This bit indicates (when set) that a CRC error has occured during transmission of the XBF in use.

b2-b1 = **TDS1, TDS0** *Transmit Data Select*

TDS1	TDS0	Function
0	0	RxD0 selected
0	1	RxD2 selected
1	0	RxD1 selected
1	1	Major Error, No communication

b0 = **TVP** *Transmit Physical Violation Error* This bit indicates (when set) that an incoherence has been detected between RxD0, RxD1 and RxD2 during transmission of the XBF in use.

RESERVED LOCATIONS

Locations from address 14h to 17h are reserved for future use.

I/O PORTS

In the ST9561 two I/O ports, Port 0 and Port 1, are available. Both ports may be configured by hardware (pin $\overline{DMX}$/P0 for Port 0 and DIR/$\overline{P1}$ for Port 1) and by software using control registers associated to ports.

I/O Port control Registers

When used in I/O Register mode, the Data Registers of the ports are mapped as following:

P0 Data Register Port 0 (P0D) Address 1Bh

P1 Data Register Port1 (P1D) Address 1Fh

The Data Registers control the logic level of the ports when configured in output mode. They are set to FFh after reset.

Two control registers per Port (PxC0 and PxC1) are used to select the output configuration. They are mapped as follows:

P0C0 Control Register 0 Port 0 Address 18h

P0C1 Control Register 1 Port 1 Address 19h

P1C0 Control Register 0 Port 0 Address 1Ch

P1C1 Control Register 1 Port 1 Address 1Dh

When $\overline{DMX}$/P0 ='1' and DIR/$\overline{P1}$ = '0', Port 0 and Port 1 are General Purpose I/O Ports. Port 0 allows the functional recreation of the MCU I/O port lost in communication with the ST9561. Each bit of the port can be individually set according to the following table.

PxC0.i	PxC1.i	I/O Bit Configuration
0	0	Input to Data Register PxD
0	1	Reserved
1	0	Output from Data Register PxD
1	1	Output Alternate Function.

where i = 0-7 and x = 0 or 1

After reset, PxC0 and PxC1 are set to 00h. It is highly recommended to refresh all data bits in these registers when using these ports in order to prevent accidental reconfiguration of non-connected port pins as inputs

Port 0 Output Alternate Functions

The PORT0 output alternate functions are as follows:

P0.7 DIN
P0.6 STAA line diagnosis signal.
P0.5 STAB line diagnosis signal.
P0.4 STAC line diagnosis signal.
P0.3 ERROR bit. This bit is set to '1' if an error has been detected in the frame.
P0.2 CVF2, encoded current state of the ST9561 internal state machine. Decoding is shown in the next table:
P0.1 CVF1
P0.0 CVF0

CVF2	CVF1	CVF0	State
0	0	0	WAIT
0	0	1	IDLER
0	1	0	IDLE
0	1	1	TBD
1	0	0	ACTIVE
1	0	1	ACKNOWLEDGE
1	1	0	ERROR

Port 1 Output Alternate Functions

When Port 1 bits 0, 4, 5, 6 and 7 are configured as output, the output alternate functions are as follows:

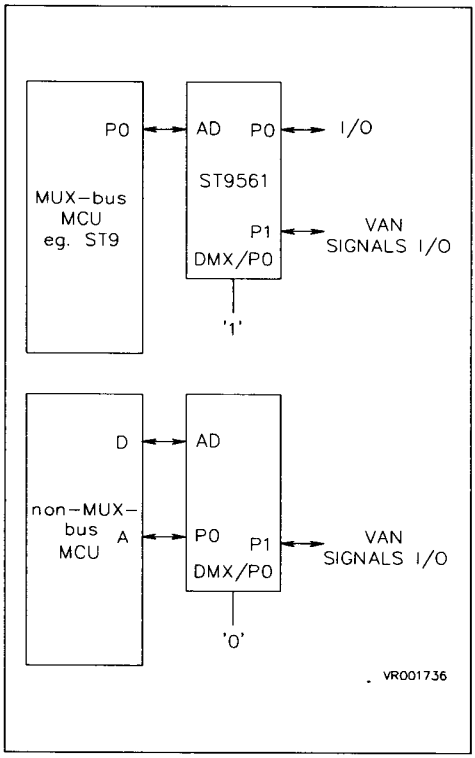
P1.0 TXD
P1.4 CKBIT
P1.5 CKTS
P1.6 CK16
P1.7 FTIP

FTIP is normally '1', it is set to '0' when the ST9561 is transmitting a frame, from the beginning of SOF to the end of EOD.

When bits 1, 2, 3 are configured as input, the input source is respectively RXD0, RXD2, RXD1.

I/O PORTS (Continued)

Figure 16. ST9561 Port Reconstruction



PORT FUNCTIONS

PORT 0

When $\overline{DMX}$/P0 ='0', Port 0 is the address input for the non-multiplexed bus interface.

7							0
P0.7	**P0.6**	**P0.5**	**P0.4**	**P0.3**	**P0.2**	**P0.1**	**P0.0**
A7	A6	A5	A4	A3	A2	A1	A0

PORT 1

When DIR/$\overline{P1}$ = '1', PORT1 is in "direct mode". The pin functions are as follows:

7							0
P1.7	**P1.6**	**P1.5**	**P1.4**	**P1.3**	**P1.2**	**P1.1**	**P1.0**
IER	CK16	CKTS	CKBIT	RXD1	RXD2	RXD0	TXD

P1.7 IER reset output for bus tranceiver
P1.6 CK16 F_{BASE} clock (1)
P1.5 CKTS TIME SLOT clock (1)
P1.4 CKBIT clock bit (1)
P1.3 RXD1 serial input 1
P1.2 RXD2 serial input 2
P1.1 RXD0 serial input 0
P1.0 TXD serial output

Note 1. DIR/$\overline{P1}$ is tied to "1" internally in the ST9560. Pins P1.6, P1.5 and P1.4 are not used and are set to inputs tied to V_{DD}.

RESERVED LOCATIONS

Locations from address 1Ah to 1Eh are reserved for future use.

IDENTIFIER BUFFER

Identifiers are stored in a 32-byte RAM.

An identifier is a 12-bit field, up to sixteen identifiers may be stored on-chip.

This RAM is mapped in the ST9560 space memory from address 32 to 63 (20h to 3Fh). Bits within the Identifier are maskable with the MASK registers. This feature is controlled by the bit "MF, Mask Filter" of the Mode Receive register. When the MF bit is '1', all identifiers are accepted.

Identifier Bit Allocation

7							0
ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4

7							0
ID3	ID2	ID1	ID0	RES	AE	RCM1	RCM0

The 8 MSB bits are stored at even addresses (b7 = MSB of these 7 bits).

The 4 remaining LS bits are stored at odd addresses, bits 7 to 4 (b7 = MSB).

Bits 3 to 0 are used as following:

b3 = *RESERVED*

bit 2 = Acknowledge enable (AE)

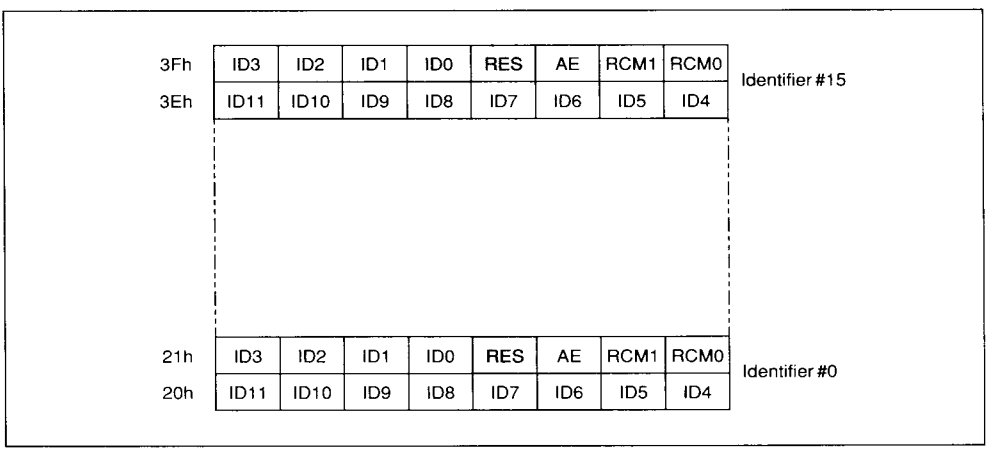
b2 = '0' acknowledge not to be generated.

b2 = '1' acknowledge generated if required.

b1-b0 = Receive control mode

RCM1	RCM0	Function
0	0	don't care identifier
0	1	mask # 1 filtered
1	0	mask # 2 filtered
1	1	no masking

Figure 17. Identifier mapping in the Identifier buffer



TRANSMIT AND RECEIVE BUFFERS

Transmit Buffer

The transmit buffer is the interface between the host CPU and the Bit Stream Processor and is able to store a whole message.

The buffer is written by the host CPU and is read by the BSP and is implemented as a 32-byte single port RAM with mutual exclusive access from the CPU and the BSP.

Two buffers are provided, XBF0 and XBF1, the user can choose either of the buffers to be written by the host.

Transmit is initiated by the host which writes in the CTR register (address 7):

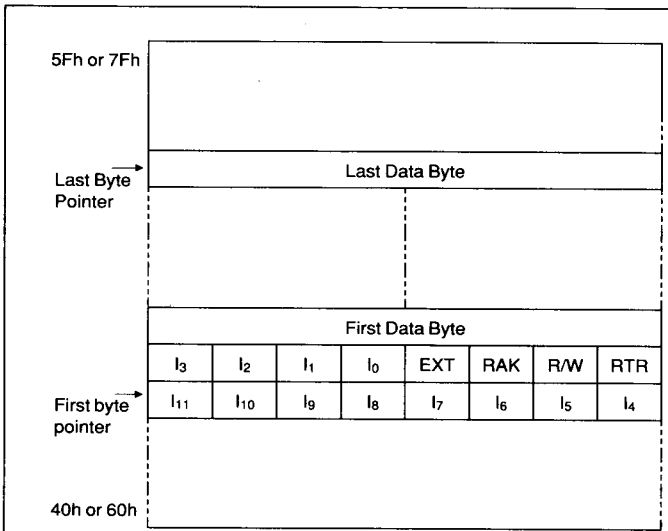
- request to send
- byte first and last address

Only one "request to send" is allowed at the same time.

An interrupt is generated at the end of transmit.

During frame transmission, the host cannot access the transmit buffer which is being used by the ST9560.

Figure 18. Transmit buffer structure



Notes:

- When there are no Data bytes (request frames), the last byte pointer is set to the 2nd Identifier command byte.
- The maximum frame length is 30 Data bytes.
- It is possible to store several frames in the same buffer.

TRANSMIT AND RECEIVE BUFFERS (Continued)

Receive Buffer

The receive buffer is the interface between the BSP and the host CPU which stores a message received from serial bus.

Once filled by the BSP and allocated to the CPU, the buffer is not available for another message. Therefore, unless the CPU releases the buffer, messages may be lost.

In order to reduce CPU requirements, two receive buffers are provided. While one buffer is allocated to the CPU, the BSP may write in the other.

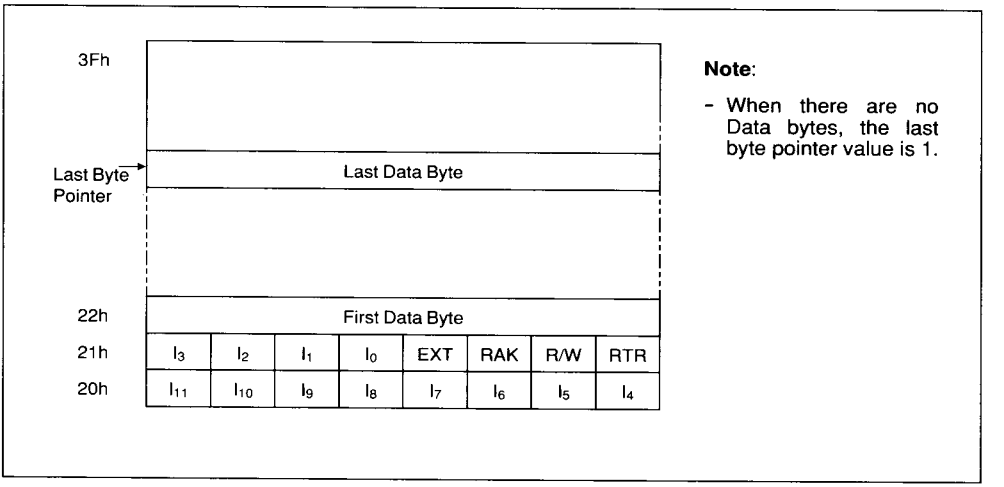
The BSP only writes into a receive buffer when the message being received passes one of the 16 acceptance filters loaded in the acceptance identifier RAM.

Both buffers are located at the same address from address 32 to address 63 (20h to 3Fh).

When a correct message is received, the Receive Status Registers 1 and 2 are updated and an interrupt is generated.

The receive buffers share the same address zone as the IDEN RAM, selection of the RAM area to be read is made under control of b7 of the MRR.

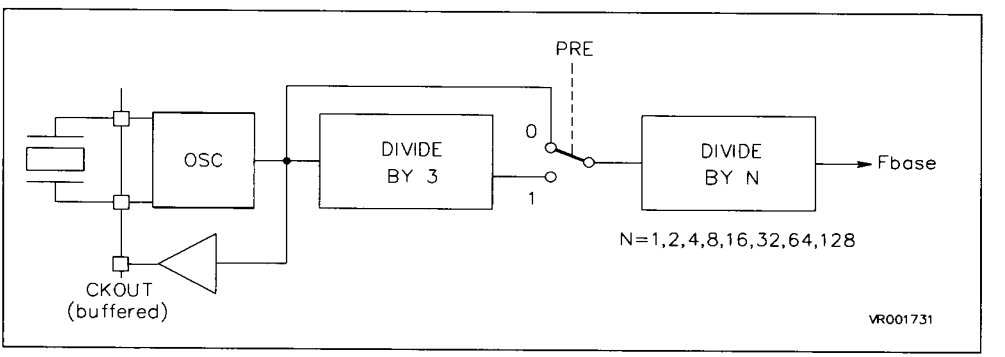
Figure 19. Receive Buffer Structure



Note:

- When there are no Data bytes, the last byte pointer value is 1.

Figure 20. F_{BASE} Clock Generator



TIMING AND SERIAL LINK RATE

ST9560 transmission rate

Time slot and bit frequencies are derived from elementary time clock called F_{BASE}.

F_{BASE} is obtained from the crystal oscillator frequency by the flow shown in figure 20, where N equals 1, 2, 4, 8, 6, 32, 64, 128 and is coded as $\log_2$ N in MCR.

According to the above flow, bit rate transmission is given by table 7, where:

- PRE designates the prescaler dividing ratio (1 or 3)
- DIV designates the coded programmed value N (F_{XTAL} or $F_{XTAL}/3$ is always divided by 2^N)
- F_{BASE} designates the elementary time frequency
- F_{TS} designates Time Slot frequency (kHz)
- M_L rate designates the rate in Manchester coding
- M_E rate designates the rate in Enhanced Manchester coding

ST9560 Oscillator

The oscillator generating the F_{BASE} timebase requires the use of an external crystal at the appropriate frequency for the transmission after division.

The circuit required is shown in the next figure:

Figure 21. ST9560 Oscillator

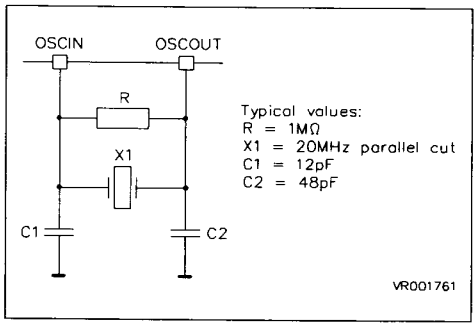


Table 7. Bit Rate Transmission (F_{XTAL} = 24MHz)

DIV	PRE	F_{BASE} (MHz)	F_{TS} (kHz)	M_L (kbits/s)	M_E (kbits/s)
1	1	24	1500	750	1200
1	3	8	500	250	400
2	1	12	750	375	600
2	3	4	250	125	200
4	1	6	375	187.5	300
4	3	2	125	62.5	100
8	1	3	187.5	93.75	150
8	3	1	62.5	31.25	37.5
16	1	1..5	93.75	46.875	75
16	3	0.500	31.25	15.625	25
32	1	0.75	46.87	23.438	37.5
32	3	0.25	15.625	7.813	12.5
64	1	0.375	23.438	11.718	18.75
64	3	0.125	7.813	3.906	6.25
128	1	0.1875	11.718	5.859	9.375
128	3	0.0625	3.906	1.953	3.125

INTERRUPT CONTROL

Interrupt Sources

The ST9560 can interrupt the micro in the five following cases, by asserting the $\overline{IRQ}$ pin low.

a) receive data available:

This occurs when an uncorrupted message, having passed the acceptance filter, is stored either in RBF0 or RBF1 buffer.

b) failed receive:

This occurs when an error is detected during reception of a message having passed the acceptance filter.

c) overrun:

A message passes the acceptance filter and neither RBF0 nor RBF1 buffers are available.

d) initialisation request:

The interrupt is generated by a low level on the $\overline{RST}$ pin or by setting bit 0 of the SCR register. No operation is allowed until this interrupt is cleared.

e) transmit data available:

The transmit buffer XBF0 or XBF1 (according to user choice) is available.

f) failed transmission:

An error has occurred during transmission of the message.

Interrupt Source Reset

Each interrupt source is cleared by setting to '1' the corresponding bit of the Reset Interrupt Register.

Enable Interrupt Register

An enable bit is associated to each interrupt source. Excepting the INIT interrupt, if the corresponding bit of an interrupt source is cleared, the source cannot interrupt the host MCU. In addition, a general interrupt enable is provided. If this bit is cleared, all interrupt sources are disabled, except the INIT interrupt.

Reset

After a reset of the ST9560 the TxD output is forced to high impedance. The ST9560 is able to transmit a level on TxD output when the INIT status interrupt is set to level 0 by the host.

ST9560 MAIN FUNCTIONS

Transmit

- acceptance of parallel data coming from the host (transmit buffer available)
- frame formatting by adding specific fields (Start Of Frame, CRC, End Of Data, End Of Frame)
- presentation to the physical layer of a serialized bit stream, starting reading from first transmit address of the transmit buffer, MSB first
- deferment of frame transmission so long as the physical medium is busy
- implementation of the arbitration mechanism
- termination of transmission upon collision detection
- in the event of loss of arbitration, scheduling a resend and continuing to receive the frame passing via the bus.
- computing the Frame Check Sequence (CRC) addition, adding it to the frame sent.
- management of the time-out to comply with the interframe spacing (IFS)
- signalling to the host that the transmit buffer is available.

Receive

- receiving a serialized bit stream from the serial bus, starting storing at address 32 (20h) of the receive buffer, MSB first.
- recompiling the complete structure of the frame (structure by field)
- checking the frame identifier and implementing the acceptance filter.
- computing the CRC of received bit stream and comparing the result with the CRC sent with the received message.
- checking the frame format
- signalling to the transmit part that an acknowledge must be sent
- removal of specific fields from the received frame
- transferring the useful parallel information (Identifier, Command and Data) to the host.

ST9560 MAIN FUNCTIONS (Continued)

Transmit Process

The ST9560 is capable of transmitting messages with variable length from 2 bytes (identifier and command fields) to lower than or equal to 30 bytes.

Transmit buffer XBF0 is used (or XBF1, according to user choice).

Host flow for transmission of a message

This flow consists in following steps:

1) check that the selected XBF buffer is available (bit 7 of Status Transmit register equals to '1')

2) write the IDENTIFIER, COMMAND and DATA fields in the TX buffer

3) write the byte count in the Command Transmit Register and set bit 7 (Request to send) of the Command Transmit Register to '1'.

4) reset interrupt source Transmit Data Available.

ST9560 flow for transmitting a message

1) check that 'Request to send' is set and TDBA (bit 3 of SIR) is cleared

2) check for 'bus idle', if not wait until the bus is free

3) send Start Of Frame

4) read the parallel data in the selected XBF buffer, serialise it and send it on the serial bus

5) monitor arbitration, if arbitration is lost, restart from step 2

6) check for errors

7) if transmission is not completed, repeat steps 4,5,6

8) send FCS field (CRC computed in parallel with step 4)

9) send EOD, ACK, EOF fields

10) set 'Transmit Buffer Available' in Status Transmit Register and assert Interrupt Transmit Request

Receive Process

As for transmit process, the ST9560 can receive frames with variable length.

Single mode is used to receive messages with length lower or equal to 32-byte.

Host flow

This flow consists in following steps:

1) check that RBF0 or RBF1 buffer is available (bit 7 of Status Receive Register equals to '1')

2) read the filtered identifier and the BYTE COUNT

3) read data from RBF0 or RBF1 buffer (BYTE COUNT is LOCATED in the Status Receive Register)

4) reset RDA and interrupt source

ST9560 flow

1) detects Start Of Frame

2) checks identifier

3) Data frame or request frame?

4) transfers IDEN, COM and DATA in the selected RBF0 or RBF1 buffer

5) checks errors

6) if not EOD, increments byte count and repeats steps 4,5

7) checks that the computed CRC result is equal to 4B15h and, if required and allowed, sends an acknowledge.

8) sets the byte count in Status Receive Register

ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$Tj = T_A + PD \times RthJA$

Where :

T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage	-0.3 to 7.0	V
T_A	Operating Temperature Range	-40 to +125	°C
T_{STG}	Storage Temperature	-55 to 150	°C
P_{DMAX}	Maximum Power Dissipation	-	W

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{DD}	Power Supply		4.5	5.0	5.5	V
V_{IL}	Input Low Level Voltage				$0.3V_{DD}$	V
V_{IH}	Input High Level Voltage		$0.7V_{DD}$			V
I_{IL} I_{IH}	Input Leakage Current	non-I/O pins	-5		5	µA
V_{OL}	Low Level Output Voltage	I_{OL} = 4.0mA			0.4	V
V_{OH}	High Level Output Voltage	I_{OH} = -4.0mA	2.4			V
PD	Power Dissipation					W