Vehicle Area Network Data Link Controller

Description

Cost optimization in car manufacturing is of extreme importance today. Solutions to this problem often implies the use of more advanced and intelligent electronic circuits.

The 29C461B is such a circuit, which allows the transfer of all the status information needed in a car or truck over a single low-cost wire pair, thereby minimizing the electrical wire usage.

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

The 29C461B is fully compliant with the ISO standard ISO/11519-3. This standard supports a wide range of applications such as low-cost remote controlled switches, typically used for lamp control, up to complex, highly autonomous, distributed systems like engine controls, which require fast and secure data transfers.

The 29C461B is a microprocessor interfaced line controller for mid to high complexity bus-masters and listeners like injection/ignition control calculators, dashboard controllers and car stereo or mobile telephone CPUs.

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

The circuit operates in the RAM using DMA techniques, controlled by the identifier and control registers. This allows virtually any microprocessor to interface with ease to the 29C461B, and to use the free RAM as a scratch pad.

Messages are encoded in enhanced Manchester code, and an optional pulsed code for use with an optical or radio link, at a maximum bit rate of 1 Mbit/s. The 29C461B analyzes the messages received or transmitted according to 7 different criterias including some higher level checks like message type mismatch and buffer available status.

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

Features

- Fully compliant to VAN specification ISO/TC22/SC3/WG1 revision 4,000.
- Handles all specified module types.
- Handles all specified message types.
- Handles retransmission of frames on contention and errors.
- 3 separate line inputs with automatic diagnosis and selection.
- 1 Mbit/s maximum transfer rate.
- Normal or pulsed (optical and radio mode) coding.
- INTEL, NEC, TI and MOTOROLA compatible 8-bit microprocessor interface.
- Multiplexed address and data bus.
- Idle and sleep modes.
- 128 bytes of general purpose RAM.
- DMA capabilities for message handling.
- 14 identifier registers with all bits individually maskable.
- 6-source maskable interrupt including an interrupt-on-reset to detect glitches on the reset pin.
- Integrated crystal or resonator oscillator with internal baud rate generator and buffered clock output.
- Single +5V power supply.
- 0.8 µm Cmos technology.
- So 24 packaging.

Figure 1. Block Diagram

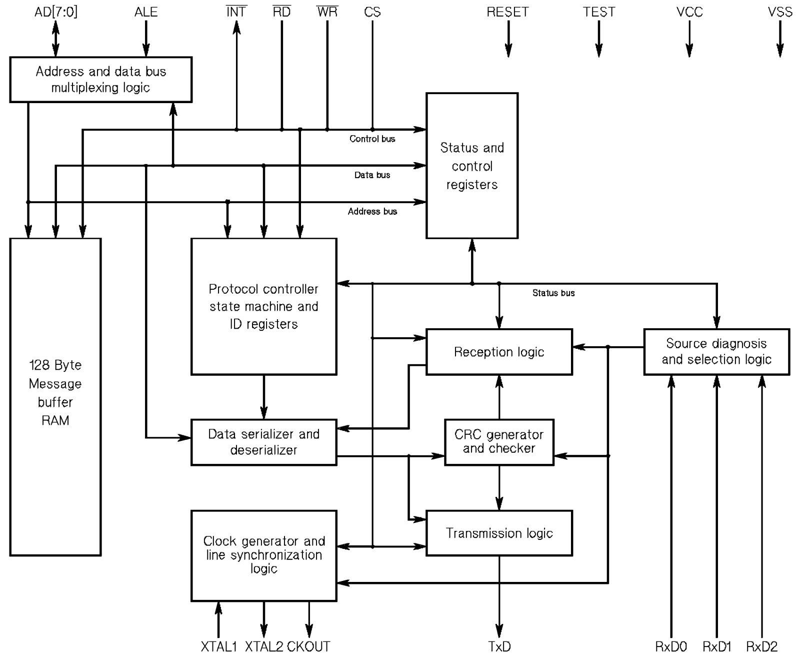
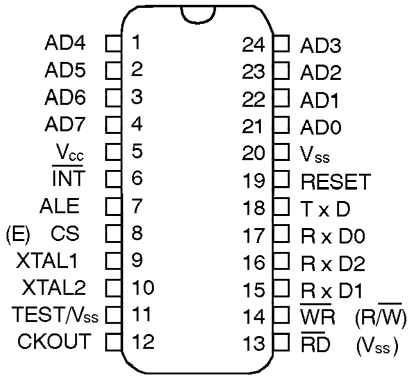

AD[7:0]
ALE
$\overline{\text{INT}}$
$\overline{\text{RD}}$
$\overline{\text{WR}}$
CS
RESET
TEST
VCC
VSS
Address and data bus multiplexing logic
Control bus
Data bus
Address bus
Status and control registers
128 Byte Message buffer RAM
Protocol controller state machine and ID registers
Status bus
Reception logic
Source diagnosis and selection logic
Data serializer and deserializer
CRC generator and checker
Transmission logic
Clock generator and line synchronization logic
XTAL1
XTAL2
CKOUT
TxD
RxD0
RxD1
RxD2

Figure 2. Pinout



24 pin SO

I/O Type	Pin Name	Pin No	Pin Function
I/O TTL	AD0	21	Multiplexed address and data bus. The address is latched on the falling edge of ALE.
	AD1	22	
	AD2	23	
	AD3	24	
	AD4	1	
	AD5	2	
	AD6	3	
	AD7	4	
I trigger TTL	ALE	7	Address Latch Enable
	$\overline{\text{RD}}$ (Vss)	13	Read command
	$\overline{\text{WR}}$ (R/W)	14	Write command
	CS (E)	8	Chip select
Open drain	$\overline{\text{INT}}$	6	Interrupt
I trigger CMOS pulldown	RESET	19	General reset
I CMOS Pull down	RxD0	17	VAN bus inputs
	RxD1	15	
	RxD2	16	
3–state	TxD	18	VAN bus output
I	XTAL1	9	Crystal oscillator or clock input pins
O	XTAL2	10	
O	CKOUT	12	Buffered clock output
Ground	TEST/Vss	11	Oscillator ground
Power	VCC	5	± 5 V power supply pins
Ground	vss	20	

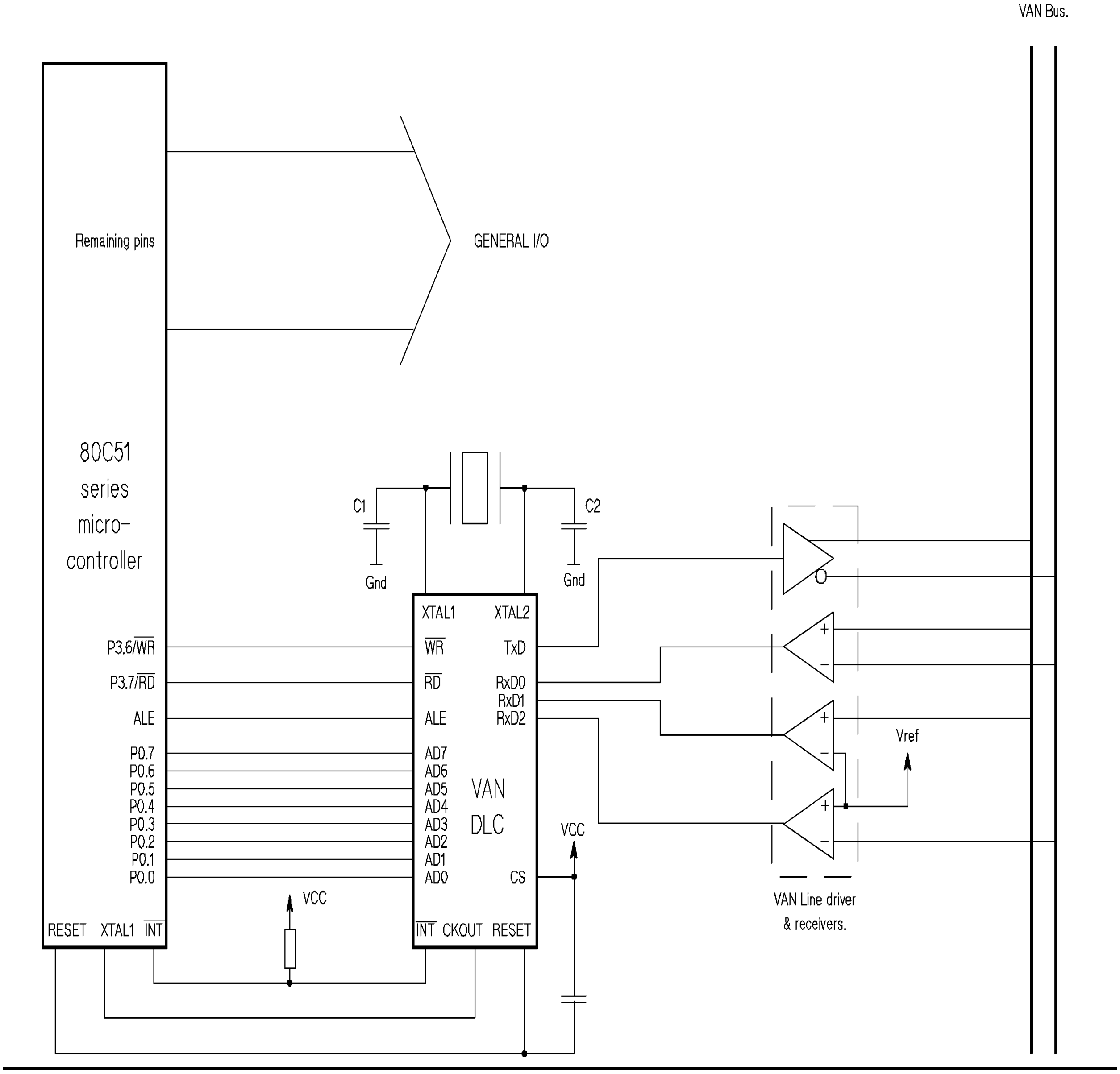
Operation

The 29C461B is a microprocessor controlled line controller for the VAN bus. It can interface to virtually any microprocessor, but the I/O signals of the circuit have been optimized for use with the MHS 80C51 series of microcontrollers.

This means that it features a multiplexed address and data bus, controlled by an address strobe pin -ALE -and separated read -$\overline{\text{RD}}$ -and write -$\overline{\text{WR}}$ -command pins. The address is latched on the falling edge of ALE.

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

Figure 3. Typical Application



1 – Pinout

The 29C461B is available in SO 24 package. Figure 2 shows the pinout. The names in parenthesis refer to the functionalities in Motorola mode.

Possibilities exist to supply the 29C461B in a PDIL 24 package.

2 – Microprocessor Interface

The processor controls the 29C461B by reading and writing the internal registers of the circuit. These registers appear to the processor as regular memory locations. The kind of interface (INTEL or MOTOROLA) is selected at reset time assuming WR signal is hold high and the first micro access is a write cycle.

2.1 – INTEL Mode

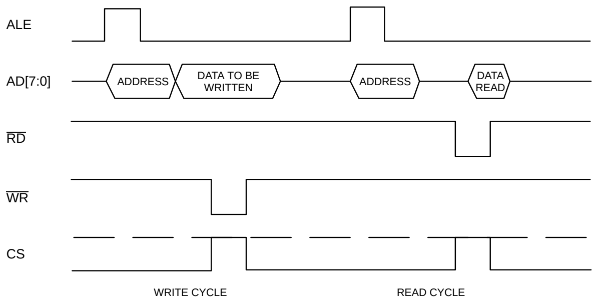
The Intel mode interface consists of 13 pins. 8 pins are the multiplexed address and data bus, and the rest are the address strobe, the read and write commands, the chip select and the interrupt request pins.

To access the memory locations in Intel mode, the processor must first assert a valid address on the multiplexed address and data bus and drive the address strobe pin high. When the required set-up time has passed the processor must drive the address strobe low, and keep the address valid for the required hold time.

The processor must then either assert the data to be written on the address and data bus, if a write is intended, or float the data bus for a read. The next step is to drive either the write or read command pins low, according to the function required, and at the same time drive the chip select pin high.

The 29C461B access cycle is then terminated by driving the chip select and command pins low.
Note, that the chip select pin may be driven high for the entire access cycle, and may also remain high during and after the termination of the cycle.

Figure 4. Read and Write Cycles.

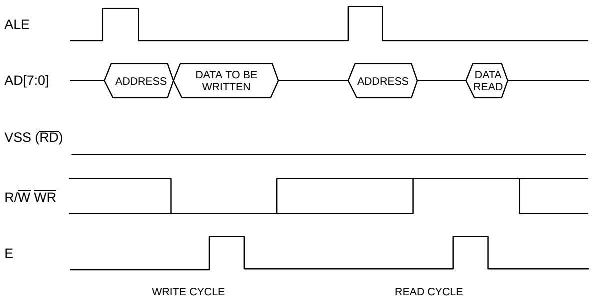


2.2 – MOTOROLA Mode

In Motorola mode the $\overline{\text{WR}}$ pin becomes the R/$\overline{\text{W}}$ command, the $\overline{\text{RD}}$ pin must be connected to ground and the CS pin becomes the E strobe. This means that there is no separate chip select input, i.e. if some external decoder is to be used, this decoder should not drive the E input high unless the processors E output is high as well.

Please refer to figure 5 for the Motorola read and write cycles. The main difference between Intel and Motorola mode is that the timing in Intel mode is referenced to the command signals ($\overline{\text{RD}}$ and $\overline{\text{WR}}$), but in Motorola mode the reference is the E signal.

Figure 5. Motorola Read and Write Cycles.


ALE
AD[7:0]
ADDRESS
DATA TO BE WRITTEN
ADDRESS
DATA READ
VSS ($\overline{\text{RD}}$)
R/$\overline{\text{W}}$ $\overline{\text{WR}}$
E
WRITE CYCLE
READ CYCLE

Interrupts

If an event occurs in the 29C461B, that needs the attention of the processor, this will be signalled on the active low, open drain interrupt request pin. Which events that create such a request is controlled by the internal registers.

Every time the microprocessor accesses any of the interrupt registers (addresses 0x08 to 0x0b) the $\overline{\text{INT}}$ pin will be released momentarily. This enables the 29C461B to work with processors that have either edge or level sensitive interrupt inputs.

3 – Oscillator

An oscillator is integrated in the 29C461B, and consists of an inverting amplifier of which the input is XTAL1 and the output XTAL2.

A parallel resonance quartz crystal or ceramic resonator must be connected to these pins. As can be seen from figure 3, two capacitors have to be connected from the crystal pins to ground. The values of C2 depend on the frequency chosen and can be selected using the nomograph given in figure 20.

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

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

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

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

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

Furthermore, it is subject to change without notice, the only exception beeing for incoming inspection tests using the MHS test program.

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

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

Table 1 shows the different values for the clock divider versus clock frequency and bit rates.

Table 1 : Clock Divider.

		32 MHz		**24 MHz**		**8 MHz**	
Clock Divider	**Divide by**	**Ktimeslot /s**	**Kbits /s**	**Ktimeslot /s**	**Kbits /s**	**Ktimeslot /s**	**Kbits /s**
0000	1	2000	1600	1500	1200	500	400
0001	2	1000	800	750	600	250	200
0010	4	500	400	375	300	125	100
0011	8	250	200	187.5	150	62.5	50
0100	16	125	100	93.75	75	31.25	25
0101	32	62.5	50	46.875	37.5	15.625	12.5
0110	64	31.25	25	23.438	18.75	7.813	6.25
0111	128	15.625	12.5	11.718	9.375	3.906	3.125
1000	1.5	1333	1067	1000	800	333.333	266.666
1001	3	666.667	533.333	500	400	166.666	133.333
1010	6	333.333	266.666	250	200	83.333	66.666
1011	12	166.666	133.333	125	100	41.666	33.333
1100	24	83.333	66.666	62.5	50	20.833	16.666
1101	48	41.666	33.333	31.25	25	10.416	8.333
1110	96	20.833	16.666	15.625	12.5	5.208	4.166
1111	192	10.416	8.333	7.813	6.125	2.604	2.083

4 – Line Interface

There are three line inputs and one line output available on the 29C461B. Which of the three inputs to use is either programmable by software or automatically selected by a diagnosis system.

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

The data on the line is encoded according to the VAN standard ISO/11519-3. This means that the 29C461B is using a two level signal having a recessive (1) and a dominant (0) state. Furthermore, due to the simple medium used, all data transmitted on the bus is also received simultaneously.

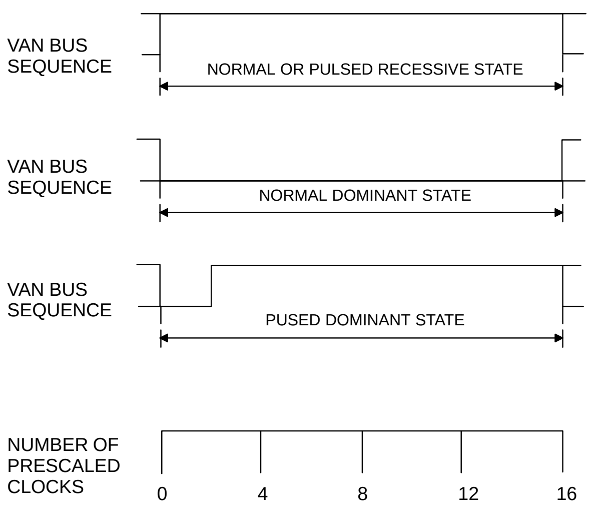
The VAN protocol is hence a CSMA/CD (carrier sense multiple access / collision detection) protocol, allowing for continuous bitwise arbitration of the bus, and non-destructive (for the higher priority message) collision detection.

In addition to the VAN specification there is also a pulsed coding of the dominant and recessive states. This mode is intended to be used with an optical or radio link. In this mode the dominant state for the transmitter is a low pulse, and the recessive state is just a high level.

When receiving in this mode it is not the state of the signal itself which is decoded, but the edges. Also, reception is imposed on the RxD0 input, and the diagnosis system does not operate correctly.

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

Figure 6. State Encoding.



In figure 6 the pulsed waveforms are shown. In figures 8 through 11 the low “timeslots” (i.e. blocks of 16 prescaled clocks) should be replaced by the dominant waveform showed in figure 6, if the correct representations for pulsed coding are to be seen.

The VAN bus supports three different module (unit) types. First, the autonomous module which is a bus master. It can emit SOF sequences, initiate data transfers and receive messages.

The synchronous access module, cannot emit any SOF sequences, but it can initiate data transfers and receive messages.

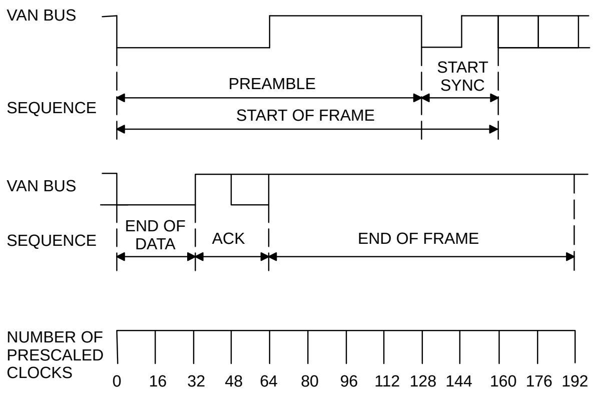
Finally the slave module, which can only transmit using a in-frame response mechanism, and receive messages.

Figure 7. Van Bus Frame.

SOF	IDENTIFIER FIELD	COMMAND				DATA FIELD	FRAME CHECK SUM	EOD	ACK	EOF
		EXT	RAK	R/W	RTR					

Figure 7 shows a normal VAN bus frame. It is initiated with a start of frame (SOF) sequence shown in figure 8. The SOF can only be transmitted by a autonomous module. During the preamble the 29C461B will synchronize its bit rate clock to the data received.

Figure 8. Framing Sequences.



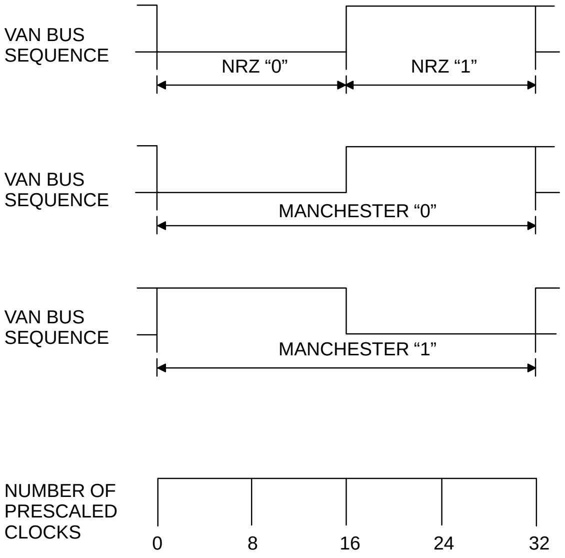
When the complete SOF sequence has been transmitted or received, the circuit will start the transmission or reception of the identifier field.

All data on the VAN bus, including the identifier and frame check sum, are transmitted using enhanced Manchester code.

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

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

Figure 9. Data Encoding.



The IFS is defined to be a minimum of 64 prescaled clocks periods. The 29C461B, accepts an IFS of zero prescaled clocks for the reception only of a SOF sequence.

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

Up till this point there can be several modules transmitting on the bus, and there is no possibility of knowing if this is the case or not. Therefore the first field in wich arbitration can be performed is the identifier field.

Since the logical zeroes on the bus are dominant, and all data is transmitted with the most significant bit (MSB) first, the first module to transmit a logical zero on the bus will be the prioritized module, i.e. the message that is tagged with the lowest identifier will have priority over the other messages.

It is, however, conceivable that two messages transmitted on the bus will have the same identifier. The 29C461B therefore continues the arbitration of the bus throughout the whole frame. More, if the identifier in transmission has been programmed for reception as well, it transmits and receives messages simultaneously, right up till the frame check sequence. Only then, if the 29C461B has transmitted the whole message, does it discard the message received. Arbitration loss in the Frame Check Sum (FCS) field is considered as a CRC error during transmission.

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

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

The first bit of the command is the extension bit (EXT). This bit is defined by the user on transmission and is received and retained by the 29C461B, but in the VAN recommendation its use is not yet specified. To conform with the standard it should be set to 1 (recessive) by the user.

The second bit is the request acknowledge bit (RAK). If this bit is a logical one, the receiving module must acknowledge the transfer with an in-frame acknowledgement in the ACK field. If it is set to logical zero, then the ACK field must contain an acknowledge absent sequence.

Third we have the read/write bit (R/W). This bit indicates the direction of the data in a frame. If set to zero it is a “write” message, i.e. data transmitted by one module to be received by another module.

If it is set to one it implies a “read” message, i.e. a request that another module should transmit data to be received by the one that requested the data (reply request message).

Last in the command field is the remote transmission request bit (RTR). This bit is a logical zero if the frame contains data and a logical one if the frame does not contain data.

All the bits in the command field are automatically handled by the 29C461B, so the user need not to be concerned for the encoding and decoding of these. The command bits transmitted on the VAN bus are calculated from the current status of the active message.

After the command field comes the data field. This is just a sequence of bytes transmitted MSB first. In the VAN standard the maximum message length is set to 28 bytes, but the 29C461B handles messages up to 30 bytes.

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

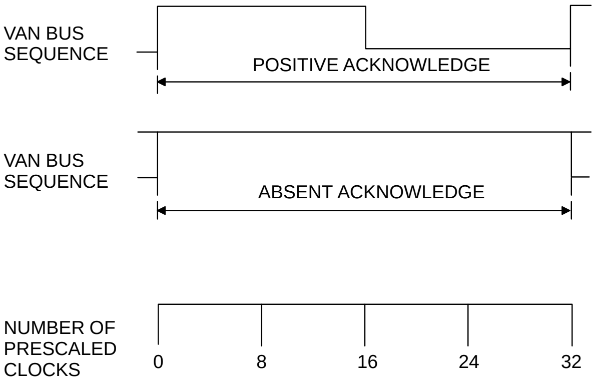
$$\mathbf{g(x) = x^{15}+x^{11}+x^{10}+x^{9}+x^{8}+x^{7}+x^{4}+x^{3}+x^{2}+1}$$

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

However, since the CRC is calculated automatically from the identifier, command and data fields by the 29C461B, it need not concern the user of the circuit.

When the frame check sequence has been transmitted, the transmitting module must transmit an end of data (EOD) sequence, followed by the acknowledge field (ACK) and the end of frame sequence (EOF) to terminate the transfer.

Figure 10. Acknowledge Sequences.



Frame Examples

The frames transmitted on the VAN bus are generated by several modules, each supplying different parts of the message. Figures 11 through 14 show the four frame types specified in the VAN standard, and what module is generating the different fields.

The most straightforward frame is the normal data frame in figure 11. Like all other frames it is initiated with a SOF sequence. This sequence is generated by a bus master (not shown in figure).

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

The reply request frame with immediate reply in figure 14 is the only frame in which a slave module can transmit data by filling it into the appropriate field.

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

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

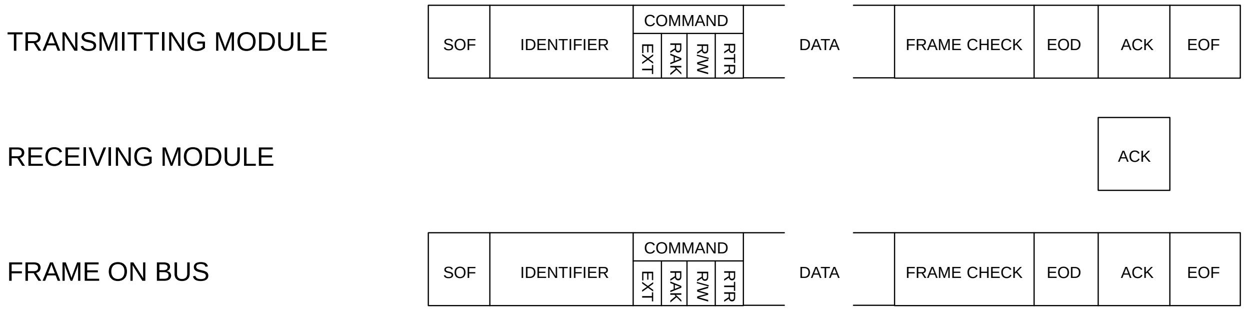
The reply request frame with deferred reply in figure 14 is basically the same frame as the reply request frame with immediate reply, but since the requested module does not generate the RTR bit the requesting module will continue with the frame check, the EOD and the EOF.

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

Finally the deferred reply frame in figure 15 which is sent when a module has prepared a reply for a reply request that has been received earlier.

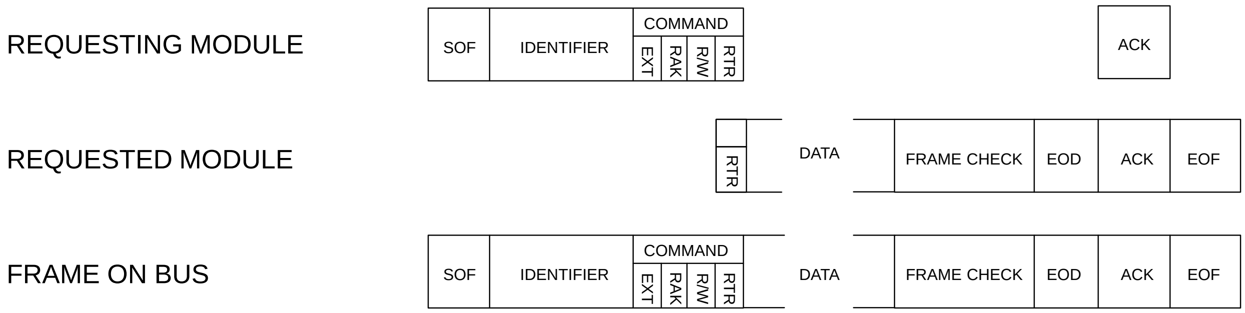
This frame very closely mimicks the normal data frame with the only exception beeing the R/W bit that has changed state.

Figure 11. Normal Data Frame.



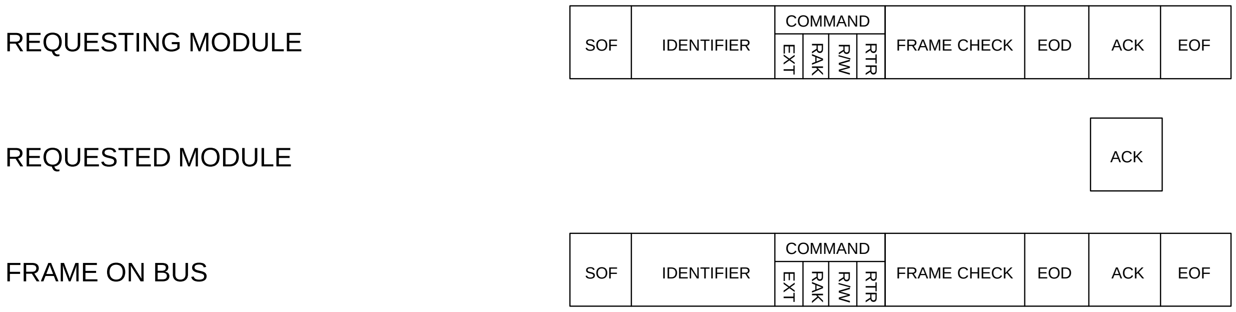
EXT recessive from transmitter
RAK recessive for acknowledge or dominant for no acknowledge from transmitter
R/W dominant from transmitter
RTR dominant from transmitter
ACK absent from transmitter and positive from receiver if RAK recessive

Figure 12. Reply Request Frame with Immediate Reply.



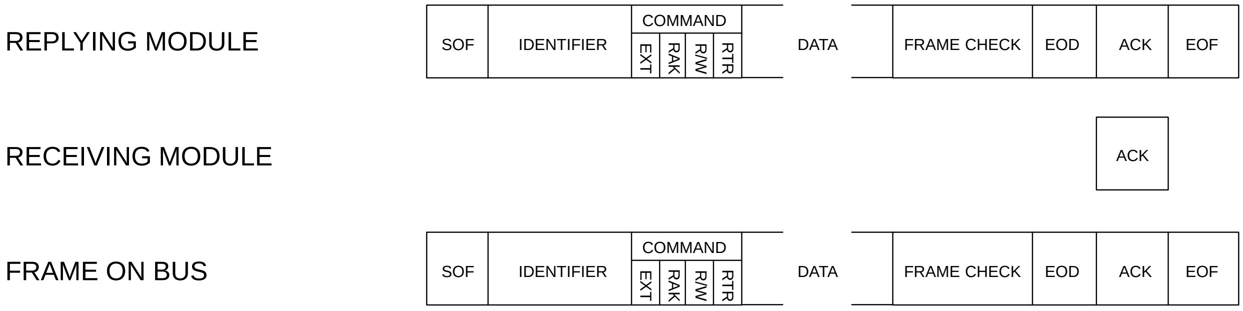
EXT recessive from requestor
RAK recessive for acknowledge or dominant for no acknowledge from requestor
R/W recessive from requestor
RTR recessive from requestor and dominant from requestee
ACK absent from requestee and positive from requestor if RAK recessive

Figure 13. Reply Request Frame with Deferred Reply.



EXT recessive from requestor
RAK recessive for acknowledge or dominant for no acknowledge from requestor
R/W recessive from requestor
RTR recessive from requestor
ACK absent from requestor and positive from requestee if RAK recessive

Figure 14. Deferred Reply Frame.



EXT recessive from replyer
RAK recessive for acknowledge or dominant for no acknowledge from replyer
R/W recessive from replyer
RTR dominant from replyer
ACK absent from replyer and positive from receiver if RAK recessive

5 – Diagnosis System

The diagnosis system is based on the assumption that three separate line receivers are connected to the VAN bus, as shown in the typical application in figure 3.

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

The objective is to detect if there are any short or open circuits on either the DATA or $\overline{\text{DATA}}$ lines.

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

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

If the user wishes to disable the diagnosis system, he only needs to select one of the fixed modes as listed in table 2 (section 5.7), and disregard the status bits in the line status register.

The system performs five different types of analysis on the VAN bus signals :

Input comparison.
Asynchronous diagnosis.
Synchronous diagnosis.
Transmission diagnosis.
Protocol anomaly.

In order to perform these analysis three signals are generated internally in the 29C461B.

The Return-to-Idle signal (RI), returns the diagnosis system to the nominal mode immediately.

The Synchronous-Diagnosis-Clock (SDC), controls the cycle time of the synchronous diagnosis. Typically there should be at least one message on the bus per SDC period.

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

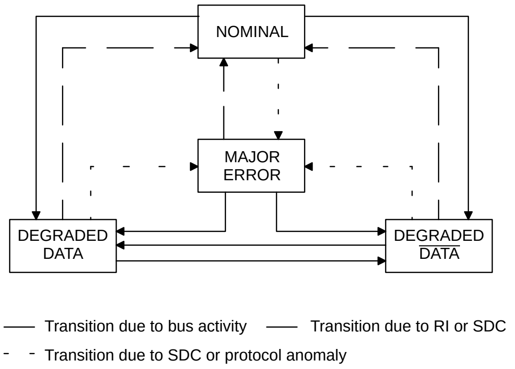
5.1 – Input Comparison

The input comparison is done once every "timeslot" (one timeslot is about 16 prescaled clock periods. The about is due to the synchronization of the timeslots to the data received).

If one of the three inputs differs from the others in this comparison, the high order bit of the status code (S2) is set.

The only ways of resetting this status bit are through the RI signal or a general reset. Every time RI goes high this status bit is reset.

Figure 15. Diagnosis System Status.


NOMINAL
MAJOR ERROR
DEGRADED DATA
DEGRADED $\overline{\text{DATA}}$
Transition due to bus activity
Transition due to RI or SDC
Transition due to SDC or protocol anomaly

5.2 – Asynchronous Diagnosis

The asynchronous diagnosis is done by comparing the number of edges on the DATA and the $\overline{\text{DATA}}$ inputs (RxD1 and RxD2).

If four edges are detected on one input and no edges on the other during the same timeperiod, the second input is considered faulty, and the status will change to one of the degraded modes.

5.3 – Synchronous Diagnosis

The synchronous diagnosis is counting the number of edges on the data input connected to the reception logic during one SDC period.

If there are less that four edges during one SDC period the status will change to the major error mode.

5.4 – Transmission Diagnosis

The transmission diagnosis compares the RxD1 and RxD2 inputs with the data transmitted on the TxD output.

If the TxD output is recessive during transmission and only one of the inputs are dominant the recessive input is considered faulty (due to a open circuit) and the status is changed to reflect that.

5.5 – Protocol Anomaly

The protocol anomaly is detected by counting the number of consecutive dominant “timeslots”.

If eight consecutive “timeslots” are dominant the status will change into the major error mode.

5.6 – Returing to Nominal Mode

There are two ways of returning to the nominal mode.

The first is if the diagnosis system detects no errors during two consecutive SDC periods. This functionality allows the system to return to nominal mode if some error is due to, for instance, a bad connector.

The second way of returning to nominal mode is through the RI signal (or a general reset).

5.7 – Automatically Selected Input

If the mode of the diagnostic system is set to automatic selection the input connected to the reception logic is decided by the operation mode.

In nominal mode the RxD0 (differential) input is connected to the reception logic, in degraded DATA mode the RxD2 and in degraded $\overline{\text{DATA}}$ the RxD1 inputs respectively.

In the major error mode the RxD1 and RxD2 input are connected alternatively to the reception logic. The input connected is changed every SDC period until a different diagnosis mode is entered.

Table 2 : Status and Mode Codes.

M1	M0		S1	S0	
0	0	Communication on R×D0	0	0	Nominal mode
1	0	Communication on R×D1	0	1	Degraded DATA mode
0	1	Communication on R×D2	1	0	Degraded $\overline{\text{DATA}}$ mode
1	1	Automatic selection	1	1	Major error mode

Note : Automatic selection is only recommended if SDC is active.
RI is disabled in automatic more and enabled in the fixed modes.

5.8 – RI, SDC and TIP Generation

The RI signal can be generated at the end of each frame. The automatic RI generator permits the user to make a diagnosis per frame, and is always enabled in the fixed modes, but always disabled in the automatic mode.

The SDC can be generated by a divider chain connected to the “timeslot” clock and controlled by the SDC divider. The general idea is to generate one SDC period per frame on the VAN bus. One SDC period can also be generated through the manual SDC command.

The TIP signal is generated by the 29C461B if the generate TIP bit is set, thus enabling transmission diagnosis.

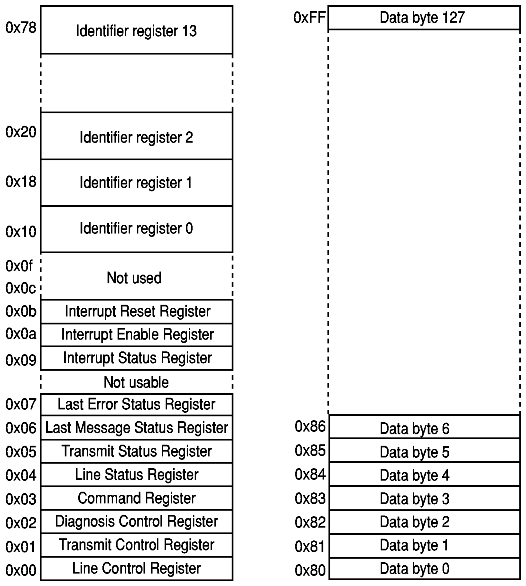
Table 3 : SDC Divider.

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

6 – Registers

The 29C461B memory map consists of three different areas, described below

Figure 16. Memory Map.


0x78 Identifier register 13
0x20 Identifier register 2
0x18 Identifier register 1
0x10 Identifier register 0
0x0f
0x0c Not used
0x0b Interrupt Reset Register
0x0a Interrupt Enable Register
0x09 Interrupt Status Register
Not usable
0x07 Last Error Status Register
0x06 Last Message Status Register
0x05 Transmit Status Register
0x04 Line Status Register
0x03 Command Register
0x02 Diagnosis Control Register
0x01 Transmit Control Register
0x00 Line Control Register
0xFF Data byte 127
0x86 Data byte 6
0x85 Data byte 5
0x84 Data byte 4
0x83 Data byte 3
0x82 Data byte 2
0x81 Data byte 1
0x80 Data byte 0

6.1 – Control Registers

There are currently 11 control registers defined in the 29C461B. If the circuit is reset, either via the external pin or the General Reset command bit, all control register bits are set to zero, with the only exceptions beeing the Reset interrupt status bit and the Reset interrupt enable bit.

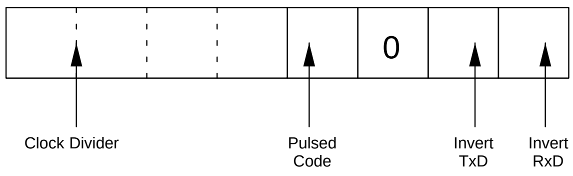
All unused bits are reserved and should not be set to anything else than the indicated “safe” value by the user, since their uses are subjected to change without notice.

The registers are divided into three groups. The control and command group, the status group and the interrupt group.

The control and command group has four registers :

6.1.1 – Line Control Register

Line Control Register



The Line Control register is located at address 0x00. It controls the VAN bus bitrate through the clock divider.

The values for the clock divider and its corresponding bitrates are discussed in detail in the oscillator section (Table 1).

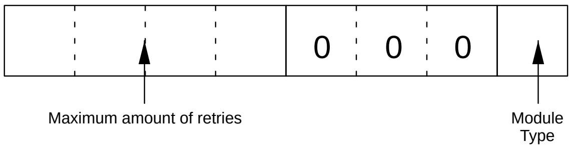
If the Pulsed Code bit is set the 29C461B will transmit and receive data using the pulsed coding mode (i.e. optical or radio link mode). The use of this mode implies communication via the RxD0 input and the non-functionality of the diagnosis system.

The Following bit is reserved for future expansion, and should always be set to zero (0).

In the two low-order bits the user can invert the logical levels used on either the TxD outputs or the RxD inputs, in order to adapt to different line drivers and receivers. A one in either of these bits will invert the respective signals.

6.1.2 – Transmit Control Register

Transmit Control Register



The Transmit Control Register at address 0x01 allows the user to control the amount of retries the circuit will perform if any errors occurred during transmission.

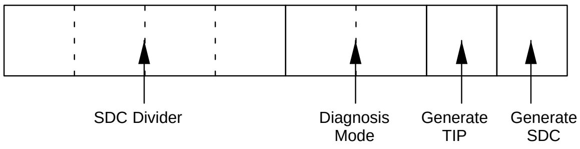
Note, that bus contention is not regarded as an error and that an infinite number of transmission attempts will be performed, if bus contention occurs continuously.

Also note, that a value of 0b0001 indicates 1 retry, hence 2 transmission attempts (not counting arbitration losses).

If the Module Type bit is set to 0 the circuit will not generate SOF sequences, and is hence a synchronous access or a slave module. If it is set to 1 the circuit will generate SOF sequences and is an autonomous module.

6.1.3 – Diagnosis Control Register

Diagnosis Control Register



At address 0x02 resides the Diagnosis Control Register.

In its four high order bits the user can program the SDC rate, in its two medium order bits the diagnosis system mode is controlled, and in the two low order bits the user controls if the SDC and TIP signals are to be generated automatically.

The diagnostic system is discussed in greater detail in paragraph 5 of this chapter (tables 2 and 3).

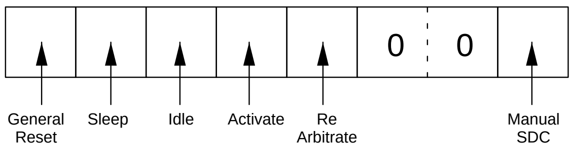
6.1.4 – Command Register

The Command Register is located at address 0x03. This register is a write only register.

If the circuit is operating at low bitrates there might be a considerable delay between the writing of this register and the performing of the actual command (worst case 6 bit periods).

The user is therefore recommended to verify by reading the Line Status register that the commands have been performed.

Command Register



The Reset circuit command bit performs, if set, exactly as if the external reset pin was asserted.

If the user sets the Sleep bit, the circuit will enter sleep mode. When the circuit is in sleep mode, all non-user registers are setup to minimize power consumption and the oscillator is stopped. To exit from this mode the user must set either the idle or activate commands.

Likewise, if the user sets the Idle bit, the circuit will enter idle mode. In idle mode the oscillator will operate, but the 29C461B will not transmit or receive anything on the bus.

Note, that the circuit will enter this mode when resetted.

The Activate command will put the circuit in the active mode, i.e. it will receive and transmit nornally on the bus. When the circuit is in active mode the TxD three-state output is enabled.

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

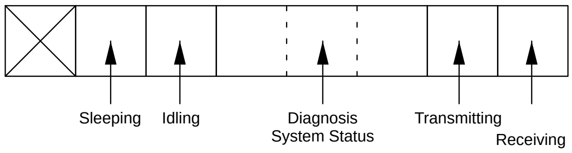
Finally the Manual SDC command that will, when set, generate a SDC pulse for the diagnosis system.

6.2 – Status Registers

The second group of register are the status registers. There are four status registers in the 29C461B.

6.2.1 – Line Status Registers

Line Status Register



The Line Status register is located at address 0x04, and reports the operation mode of the 29C461B in the Sleeping and Idling bits, as well as the diagnosis systems status bits S2 to S0.

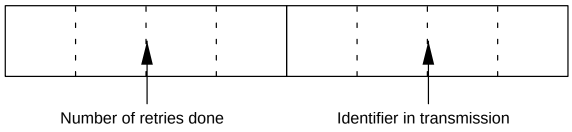
In the low order bits the user can see if the circuit is currently trying to transmit or receive on the bus. These two status bits can be observed to see if it is safe to change the setup of active ID registers.

If the transmitting status bit is active it indicates that the 29C461B has chosen an identifier to transmit, and it will continue to make transmission attempt for this message until it succeeds or the retry count is exceeded.

The receiving bit, however, indicates that there is activity on the bus. For safe modification of identifier registers both bits should be inactive.

6.2.2 – Transmission Status Registers

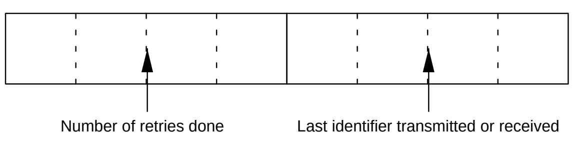
Transmission Status Register



The Transmission Status register at address 0x05 contains the identifier number currently in transmission, and the number of retries made up-to-date.

6.2.3 – Last Message Status Register

Last Message Status Register

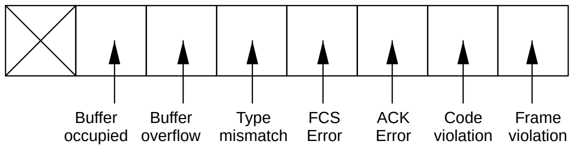


At address 0x06, the Last Message Status register is basically the same as the transmission status register. It contains the last identifier number that was successfully transmitted, received or exceeded its retry count.

If it was a successful transmission the number of retries performed can be seen in this register as well.

6.2.4 – Last Error Status Register

Last Error Status Register



Finally the Last Error Status register at address 0x07 that contains the error code for the last transmission or reception attempt. This register is updated after each attempt, i.e. several error codes can be reported during one single transmission (with several retries).

Buffer occupied indicates that the received bit of the identifier status register was set and a reception attempt was made.

Buffer overflow indicates that the buffer length setup in the identifier status register was shorter than the number of bytes received plus 1, and thus some data was lost.

Type mismatch means that an identifier received is specified in an identifier register as either a reply or a received message, but that the message received on the bus indicates the opposite type.

FCS error indicates a mismatch between the FCS received and the FCS calculated.

ACK error is set when a transmission attempt requested an acknowledge from the receiver but none was received, or vice versa.

Code violation means that a data bit on the bus was expected to be encoded as a Manchester bit, but it was not, and also than a transmitted dominant bit was received recessive.

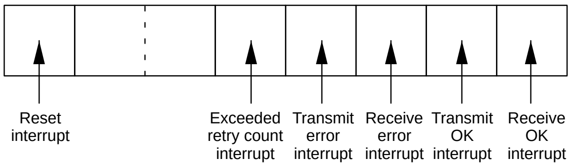
And finally Frame violation that indicates a bad framing sequence on the VAN bus. Bad SOF sequences detected on the bus does not, however, generate this type of error.

6.3 – Interrupt Registers /

The interrupt group which contains three registers:

6.3.1 – Interrupt Status Register

Interrupt Status Register



The Interrupt status register at address 0x09 that indicates which source generated the interrupt.

Reset interrupt indicates that the circuit detected a valid reset command via the RESET pin or the reset command bit. This interrupt cannot be disabled, since its enable bit is set when a reset is detected.

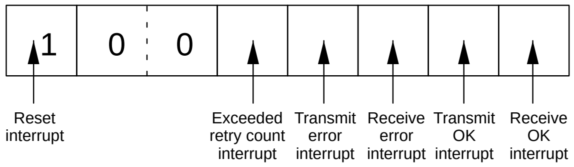
Exceeded retries interrupt is set when a transmission has been retried too many times.

The last four interrupts indicate successful and unsuccessful receptions and transmissions.

Note, that on reset the Reset interrupt status bit is set to 1 instead of 0, as is the general rule.

6.3.2 – Interrupt Enable Register

Interrupt Enable Register



The Interrupt enable register at address 0x0a, where the user indicates which conditions that should generate an interrupt request to the microprocessor.

The bit positions corresponds exactly to the bit positions in the Interrupt status register.

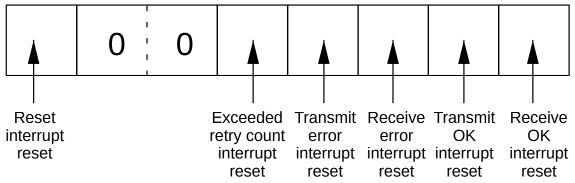
Note, that on reset the Reset interrupt enable bit is set to 1 instead of 0, as is the general rule.

6.3.3 – Interrupt Reset Register

And the Interrupt reset register at address 0x0b, written by the user to indicate which interrupts should be resetted. The bit positions in this register corresponds exactly to the bit positions in the Interrupt status and interrupt enable registers.

Note, that this register is a write-only register, and should not be read since it will generate random data.

Interrupt Reset Register


0
0
Reset interrupt reset
Exceeded retry count interrupt reset
Transmit error interrupt reset
Receive error interrupt reset
Transmit OK interrupt reset
Receive OK interrupt reset

6.4 – Identifier Registers

There is a total of 14 identifier registers, each occupying 8 bytes for addressing simplicity, integrated into the circuit. Each register contains two 16-bit registers for the identifier, mask and command fields plus two 8-bit registers for DMA pointers and message status.

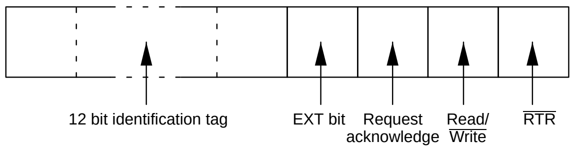
The base_address of each identifier register is (0x10 + (0x08 * identifier_number)).

When the 29C461B is reset either via the external reset pin or the general reset command, the identifier registers are not affected. That is, on power-up of the circuit, all the identifier registers start with random values.

Due to this fact the user should take great care to initialize all the identifier registers before exiting from idle mode. The easiest way to disable an identifier register is to set the received and transmitted bits to 1 in the Lenght of Status register.

6.4.1 – Identifier & Command Register

Identifier & Command Register



The Identifier and command word register is a 16-bit register located at the base_address. It allows the user to specify the full 12-bit identifier field of the ISO standard and the 4-bit command.

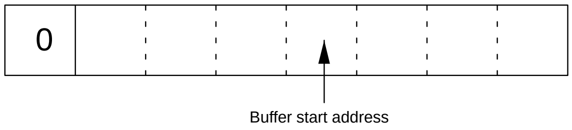
Note, that no identifier comparison will be done on the command bits, but that the EXT bit will be written into the register upon a reception hit.

Also note, that the R/$\overline{\text{W}}$ and RTR bits, as well as the status bits in the length and status register, must be in a valid position for reception or transmission. If not, the message corresponding to this identifier is considered as unactive or invalid.

Upon a reception hit (i.e. a good comparison between the identifier received and an identifier specified, taking the comparison mask into account, as well as a status and command indicating a message to be received) the 13 high bits of this register will be rewritten with the identifier and EXT bits actually received. There is no way of knowing if an acknowledge sequence was requested or not.

6.4.2 – Message Start Address Register

Message Start Address Register



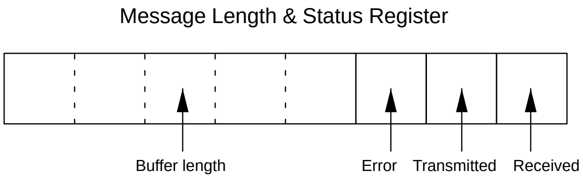
The Message start address register at address (base_address + 0x02) is 8 bits wide. It indicates where in the RAM area the message buffer is located.

Since the RAM area base address is 0x80, the value in this register is the offset from that address. If the message buffer length value is illegal (i.e. zero), this register is redefined as beeing a link pointer, thus containing the identifier number of the identifier that contains the actual buffer address, length and received status. However, the identifier, mask, error and transmitted status used will be that of the originally matched identifier. In any case, if a link is intended, the four high bits of this register should be set to 0.

This allows several identifier registers to use the same actual reception buffer in RAM, thus diminishing the memory usage.

Note, that only 1 level of link is supported.

6.4.3 – Message Length & Status Register



The Message buffer length and status register at address (base_address + 0x03) is also 8 bits wide.

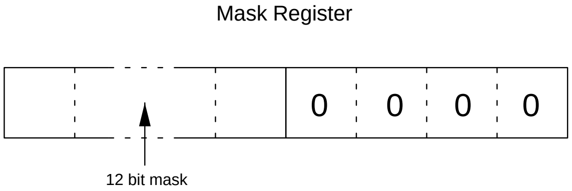
The 5 high bits of this register allows the user to specify either the length of the message to be transmitted, or the maximum length of a message receivable in the pointed reception buffer.

Note, that the first byte in the message buffer does not contain data, but the length of the message received. This implies that the length value has to be equal to or greater than the maximum length of a message to be received in this buffer (or the length of a message to be transmitted) plus 1, thus allowing a maximum length of 30 bytes.

If the value of this field is "illegal" (i.e. 0x00) then this message pointer is defined as beeing a link (see Message start address register).

The three low order bits of this register contains the message status, and together with the R/$\overline{W}$ and RTR bits of the command, they define the message type of this identifier. The status bits are only set by the 29C461B, so it's the user that must reset them. The received and transmitted bits are only set if the corresponding frame is without errors or the retry count has been exceeded.

6.4.4 – Mask Register



The 16-bit Mask register at adress (base_adress + 0x06) allows bitwise masking of the comparison between the identifier received and the identifier specified.

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

7 – Message Types

There are 5 basic message types defined in the 29C461B. Two of them (transmit and receive message types) correspond to the normal frame, and the rest correspond to the different versions of reply frames.

	Transmit Message			
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	0	0	0	Don’t care
After transmission	0	0	1	Unchanged

To transmit a normal data frame on the VAN bus, the user must program an identifier as a Transmit Message. The 29C461B will then transmit this message on the bus until it has succeeded or the retry count is exceeded.

	Receive Message			
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	0	1	Don’t care	0
After transmission	0	1	Unchanged	1

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

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

The actual identifier received is stored in the identifier register itself. This identifier may differ from the identifier specified in the register due to the effect of the mask register. Normally this should not interfere with the next identifier comparison since the bits that may differ are masked via the mask register.

	Reply Request Message			
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	1	1	0	0
After transmission (Waiting for reply)	1	1	1	0
After reception (of reply)	1	1	1	1

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

In the first case no other modules on the bus responded with an in-frame reply, and in this case the 29C461B will set the message type to the after transmission state. When this message type is programmed, the 29C461B will listen on the bus for a deferred reply frame matching this identifier, without transmitting the reply request.

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

Reply Request Message without transmission				
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	1	1	Don’t care	0
After reception	1	1	Unchanged	1

In the third case the 29C461B has not yet started to transmit the reply request, when another module either requests a reply, and gets it, or transmits a deferred reply. Warning! This should avoided as it may result in a illegal message type (Illegal Reply Request).

	Immediate Reply Message			
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	1	0	0	0
After transmission	1	0	1	1

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

	Deferred Reply Message			
	R/$\overline{W}$	RTR	Transmitted	Received
Initial setup	1	0	0	1
After Reception (of reply request)	1	0	1	1

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

Deferred Request Detection Message				
	$R/\overline{W}$	RTR	Transmitted	Received
Initial setup	1	0	1	0
After Reception	1	0	1	1

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

Inactive Message				
	$R/\overline{W}$	RTR	Transmitted	Received
Recommended	Don't Care	Don't Care	1	1
After Transmission	0	0	1	Don't care
After Reception	0	1	Don't care	1
Illegal reply request	1	1	0	1

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

8 – Priority

The priority handling on the VAN bus itself is already explained in the line interface section. The priorities for the messages in the 29C461B is however slightly different.

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

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

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

If for instance identifier 10 is beeing transmitted, and identifier 5 becomes ready for transmission, the 29C461B will continue to transmit identifier 10 until it succeeds or exceeds the maximum amount of retries allowed.

Should it be urgent that the message in identifier 5 be transmitted, the microprocessor can give the rearbitrate command to the 29C461B, and it will abort the transmission of identifier 10 after the current attempt, and check what messages are available for transmission.

Since identifier 5 has a higher priority than identifier 10, the 29C461B will start to transmit the former until it is successful or exceeds the retry count, and then continue with identifier 10. The retry count for identifier 10 will be lost though, and is reset to zero.

9 – Absolute Maximum Ratings*

Ambient temperature under bias :

A = Automotive	–40°C to 125°C
Storage Temperature	–65°C to 150°C
Voltage on V_{CC} to V_{SS}	–0.5 to +7.0 V
Voltage on any pin to V_{SS}	–0.5 V to V_{CC} + 0.5 V

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

10 – DC Characteristics

TA = –40°C to 125°C ; VCC = 5 V ± 10 % ; VSS = 0 V

Symbol	Parameter	Min	Max	Type	Test Conditions
V_{IL}	Input Low Voltage (except RESET and XTAL1)	–0.5	0.8	V	
V_{IH}	Input High Voltage (except RESET and XTAL1)	2.0	V_{CC}+0.5	V	
V_{IL1}	Input Low Voltage (RESET and XTAL1)	–0.5	0.3·V_{CC}	V	
V_{IH1}	Input High Voltage (RESET and XTAL1)	0.7·V_{CC}	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 3.2 mA, Vcc min
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –3.2 mA, Vcc min
I_L	Input Leakage Current		±5	µA	0 < V_{IN} < V_{CC}
R_{PD}	Input pulldown resistor	75		kΩ	Note 5
C_{IO}	I/O Buffer Capacitance		10	pF	Not tested
I_{CC}	Power Supply Current Sleep mode Idle or Active mode		 50 15	 µA mA	 (Note 1) (Notes 2,4)
I_{CCOSC}	Oscillator Power Supply Current 1MHz 5MHz 20MHz		 10 10 10	 mA mA mA	Not tested (Note 3)

Notes :
1. Sleep Mode I_{CC} is measured according to figure 17, with a V_{SS} Clock Signal.
2. Active mode I_{CC} is measured according to figure 17. 20 MHz clock, 250 KTS/s.
3. I_{CCOSC} is measured according to figure 18.
4. I_{CC} is a function of the Clock Frequency. In figure 19 is displayed a graph showing I_{CC} versus Clock frequency.
5. RESET, RXD0, RXD1, RXD2 inputs.

Figure 17. I_{CC}

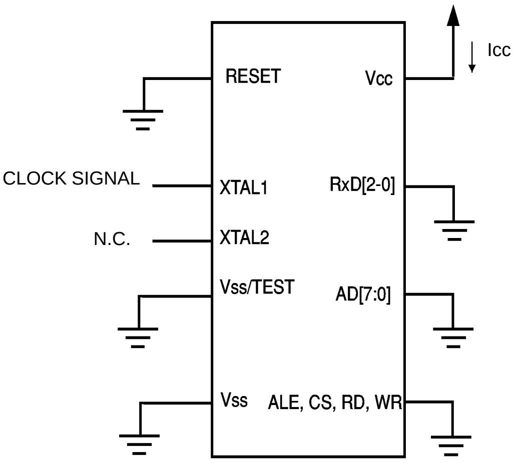


Figure 18. I_{CCOSC}

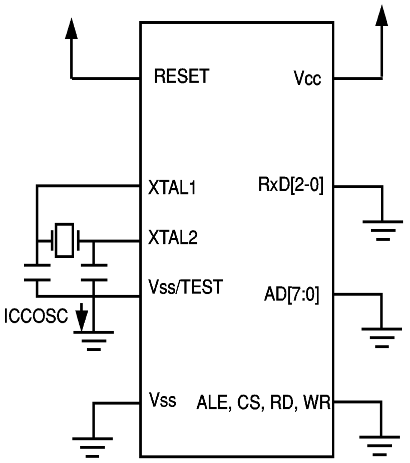
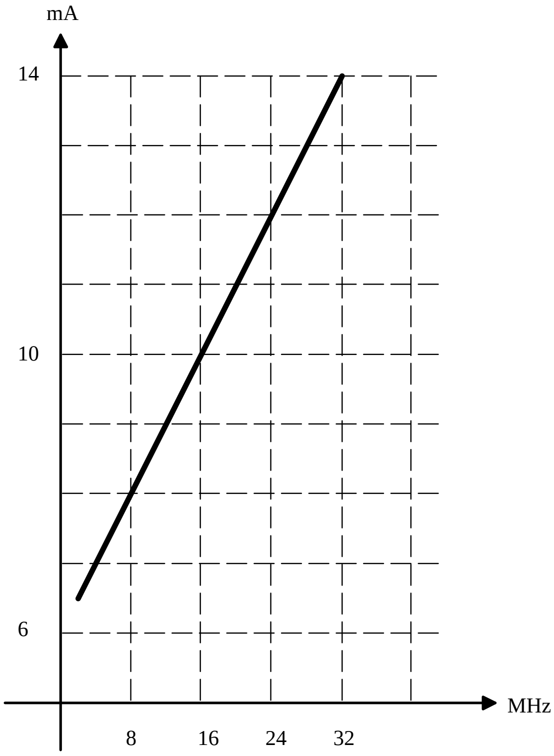


Figure 19. I_{CC} **Versus Clock Frequency at 250 KTimeslot/s**



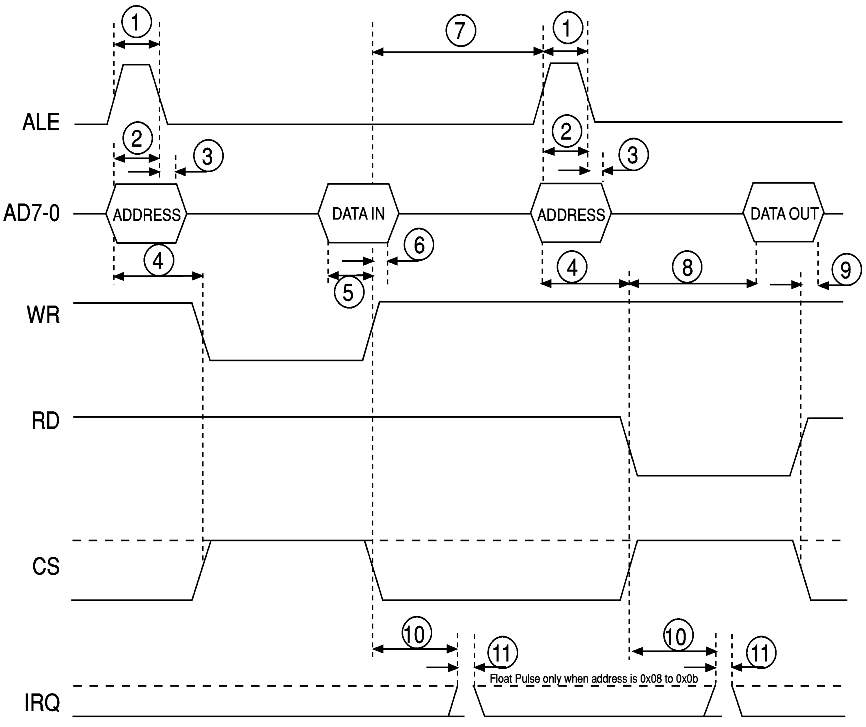
AC Characteristics

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

Microprocessor Interface

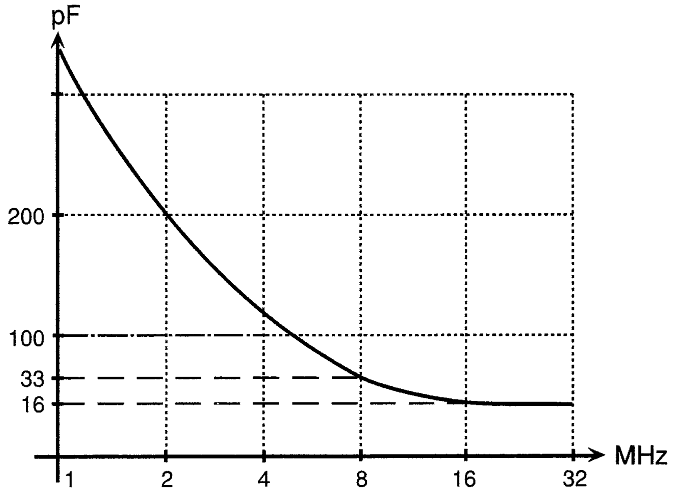
CLOAD = 30pF

	Symbol	Parameter	Min	Max	Unit
	T_{RESET}	RESET High Pulse Width (For Power–up Reset)	16		Clock Cycles
1	T_{LHLL}	ALE High Pulse Width	15		ns
2	T_{AVLL}	Address Valid to ALE Low Setup Time	10		ns
3	T_{LLAX}	ALE Low to Address Invalid Hold Time	10		ns
4	T_{AVWL}	Address Valid to Command Active Time	20		ns
5	T_{DVWH}	Data Valid to Write Inactive Setup Time	10		ns
6	T_{WHDX}	Write Inactive to Data Invalid Hold Time	10		ns
7	T_{WHLH}	Write Inactive to ALE High Recovery Time	20		ns
8	T_{RLDV}	Read Active to Data Valid Access Time		120	ns
9	T_{RHDZ}	Read Inactive to Data Float Time		20	ns
10	T_{WHRLIZ}	Write Inactive or Read Active to IRQ Float Time		120	ns
11	T_{IZIL}	IRQ Float Pulse Width	2	20	ns



Oscillator Characteristics

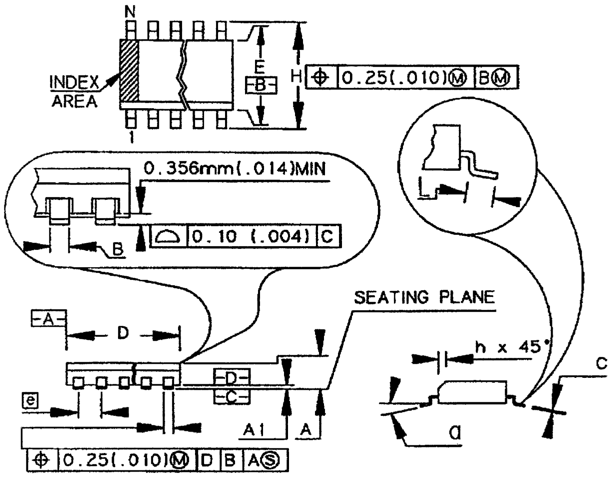
Figure 20. C2 Versus Frequency.



C1 = Crystal load (no capacitance needed)

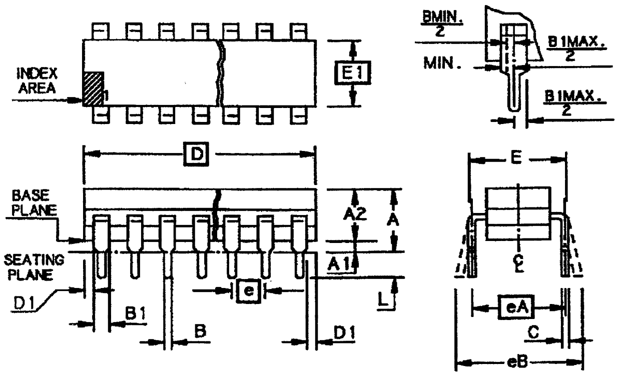
Packaging

SO 24



SO	MM		INCH	
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
B	0.35	0.49	0.014	0.019
C	0.23	0.32	0.009	0.013
D	15.20	15.60	0.599	0.614
E	7.40	7.60	0.291	0.299
e	1.27	BSC	0.050	BSC
H	10.00	10.65	0.394	0.419
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
N	24		24	
a	0°		0°	

PDIL 24



PDIL	MM		INCH	
A	–	5.33	–	0.210
A1	0.39	–	0.015	–
A2	2.92	4.95	0.115	0.195
B	0.36	–	0.014	–
B1	1.14	1.78	0.045	0.070
C	0.20	0.38	0.008	0.015
D	28.60	32.30	1.125	1.275
E	7.62	8.25	0.300	0.325
E1	6.10	7.11	0.240	0.280
e	2.54	BSC	0.100	BSC
eA	7.62	BSC	0.300	BSC
eB	–	10.92	–	0.430
L	2.92	4.06	0.115	0.160
D1	0.13	–	0.005	–

Ordering Information

A	**T** **P**	**29C461B**	**: R**
Temperature Range A : Automotive	Package Type T : SO P : PDIL	Part Number	Packing Blank : Tubes R : Tape and Reel

The information contained herein is subject to change without notice. No responsibility is assumed by MATRA MHS SA for using this publication and/or circuits described herein : nor for any possible infringements of patents or other rights of third parties which may result from its use.