ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

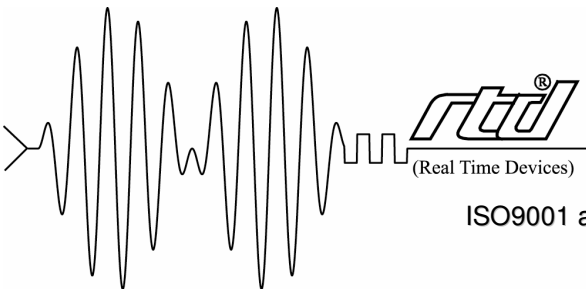
- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *RTD Embedded Technologies ECAN527HR-D* at our website: *Click HERE*

ECAN527DHR Dual Channel CAN Interface Board User's Manual



RTD Embedded Technologies, Inc.

(Real Time Devices)

"Accessing the Analog World"®

ISO9001 and AS9100 Certified

BDM-610020018
Rev. D

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ECAN527DHR DUAL CHANNEL CAN INTERFACE BOARD USER'S MANUAL



RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803-0906

Phone: +1-814-234-8087
FAX: +1-814-234-5218

E-mail
sales@rtd.com
techsupport@rtd.com

Web site
http://www.rtd.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Revision History

20/05/1998		Release 2.0
21/09/1998		Hardware release 1.1 changes: - No interrupt sharing - EEPROM pullup resistors - CE information
02/02/1999		Updated CE information Corrected diagram of interrupt jumpers
03/01/2001		Small changes in driver section
11/07/2001		Name of company changed, formatted pages
Rev. A		New Manual Naming Method
Rev. B		Added CAN reset function
Rev. C	05/04/2007	- Clarified references to QNX. - Minor grammar and formatting improvements.
Rev. D	01/20/2010	- Added individual reset description - Added IDAN configuration

Published by:

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803-0906

Copyright 1999-2010 by RTD Embedded Technologies, Inc.
All rights reserved

The RTD Logo is a registered trademark of RTD Embedded Technologies. cpuModule and utilityModule are trademarks of RTD Embedded Technologies. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 7

FEATURES 7
CAN-BUS CONTROLLER 7
PHYSICAL INTERFACE 7
MECHANICAL DESCRIPTION 8
CONNECTOR DESCRIPTION 8
WHAT COMES WITH YOUR BOARD 8
BOARD ACCESSORIES 8
USING THIS MANUAL 8

CHAPTER 2 BOARD SETTINGS 10

FACTORY-CONFIGURED JUMPER SETTINGS 10
BASE ADDRESS JUMPERS (FACTORY SETTING: D0000H) 11

CHAPTER 3 BOARD INSTALLATION 14

BOARD INSTALLATION 14
General installation guidelines: 14
Installation integrated with a PC/104 module stack 14
General purpose digital I/O connector 16
Galvanically isolated CAN bus connectors 17
Galvanically isolated CAN bus Termination Jumpers 17

CHAPTER 4 - HARDWARE DESCRIPTION 18

Galvanic isolation of the CAN-bus 19
Configuration EEPROM 20
DIGITAL I/O 20

CHAPTER 5 BOARD OPERATION AND PROGRAMMING 21

DEFINING THE MEMORY MAP 21
Global Reset 21
Individual Reset 21
INTERRUPTS 21
What is an interrupt? 21
Interrupt request lines 22
8259 Programmable Interrupt Controller 22
Interrupt Mask Register (IMR) 22
End-of-Interrupt (EOI) Command 22
What exactly happens when an interrupt occurs? 22
Using Interrupts in your Program 23
Writing an Interrupt Service Routine (ISR) 23
Saving the Startup Interrupt Mask Register (IMR) and interrupt vector 24
Common Interrupt mistakes 25
Example on Interrupt vector table setup in C-code: 25

CHAPTER 6 ECAN527DHR SPECIFICATIONS 27

HOST INTERFACE 27
CAN INTERFACES 27
DIGITAL I/O (NON ISOLATED) 27
CONNECTORS 27
ELECTRICAL 27

CHAPTER 7 IDAN-ECAN527DHRS CONFIGURATION 28

IDAN SYSTEM MODULE DIMENSIONS 28
IDAN SYSTEM MODULE PINOUT 29

CHAPTER 8 RETURN POLICY AND WARRANTY 30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RETURN POLICY 30
LIMITED WARRANTY 31

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LIST OF ILLUSTRATIONS AND TABLES

FIGURES

FIGURE 1: ECAN527DHR BOARD LAYOUT SHOWING JUMPER LOCATIONS 10
FIGURE 2: BASE ADDRESS JUMPERS ILLUSTRATING ADDRESS D8000 12
FIGURE 3: INTERRUPT JUMPERS 13
FIGURE 4: ECAN527DHR INTEGRATED IN A PC/104 RTD CPUMODULE STACK 15
FIGURE 5: 19" EUROCARD RACK INSTALLATION WITH AN INTEGRATED PC/104 DATAMODULE AND 15
FIGURE 6: ECAN527DHR BLOCK DIAGRAM 18
FIGURE 7 IDAN DIMENSIONS 28
FIGURE 8 IDAN CONFIGURATION 28

TABLES

TABLE 1: FACTORY CONFIGURED JUMPER SETTINGS 10
TABLE 2: BASE ADDRESS JUMPER SETTINGS ECAN527DHR 12
TABLE 3: DIGITAL I/O CONNECTOR J28 OF THE ECAN527DHR 16
TABLE 4: DIGITAL I/O CONNECTOR J49 OF THE ECAN527DHR 16
TABLE 5: PHYSICAL INTERFACE CONNECTOR J35 / J45 PIN-OUTS OF THE ECAN527DHR 17
TABLE 6: DIGITAL I/O CONFIGURATION OF PORT P2 OF THE 82527 CHIP NUMBER 1 20
TABLE 7: DIGITAL I/O CONFIGURATION OF PORT P2 OF THE 82527 CHIP NUMBER 2 20
TABLE 8 IDAN PINOUT 29

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 1 INTRODUCTION

This user's manual describes the operation of the ECAN527DHR CAN-bus Interface board.

Features

Some of the key features of the ECAN527DHR include:

- Two independent CAN-network controllers
- 1 Mb/s maximum data rate (fully programmable)
- Full CAN-functionality 2.0 B
- 2 Independent Intel 82527 CAN-bus controllers
- 256 bytes of configuration EEPROM
- Galvanically isolated physical interfaces
- 14 user accessible bit programmable bi-directional digital I/O
- 4 status LED's (2 for each channel)
- +5V only operation
- PC/104 compliant

The following paragraphs briefly describe the major features of the ECAN527DHR. A more detailed discussion in included in chapter 4 (Hardware description) and in Chapter 5 (Board operation and programming). The board setup is described in Chapter 2 (Board Settings). A full description of the Intel 82527 CAN-controllers is included in Chapter 5 (Board operation and programming).

CAN-bus controller

The ECAN527DHR CAN-bus interface is implemented using the Intel 82527 chips. This controller supports CAN Specification 2.0. This versatile chip supports standard and extended Data and Remote frames; a Programmable Global Message Identifier Mask; 15 message objects of 8-byte Data Length and a Programmable Bit Rate. This fully integrated chip supports all the functionality of the CAN-bus protocol.

Physical Interface

Industrial environments require galvanic isolation and bus filtering to provide reliable data communication and safety. The *galvanically isolated physical interface* is implemented using high-speed optocouplers and a DC/DC converter. To protect the input from radiated bus noise a specially balanced bus filter is used. The bus connectors conform to the ISO11898/2 specification. (For more information on CAN bus please visit the CAN in Automation Website at**: http://www.can-cia.de**.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Mechanical description

The ECAN527DHR is designed on a PC/104 form factor. An easy mechanical interface to both PC/104 and EUROCARD systems can be achieved. Stack your ECAN527DHR directly on a PC/104 compatible computer using the onboard mounting holes.

Connector description

There are two 10-pin interface header connectors on the ECAN527DHR to directly interface to the galvanically isolated CAN-networks. The general-purpose digital outputs and inputs are connected to the ECAN527DHR by using a 20-pin header connector.

What comes with your board

Your ECAN527DHR package contains the following items:

- ECAN527DHR CAN-bus interface module
- Companion CD with documentation, drivers, and example software

Note: Latest revisions of documentation and software can be downloaded from our website.

If any item is missing or damaged, please call RTD customer service department at the following number: +1-814-234-8087.

Board accessories

In addition to the items included in your ECAN527DHR delivery, several accessories are available. Contact your distributor for more information and for advice on selecting the most appropriate accessories to support your instrumentation system.

Using this manual

This manual is intended to help you install your new ECAN527DHR card and get it working quickly, whilst also providing enough detail about the board and its functions so that you can obtain maximum use of its features even in the most demanding applications. The scope of this manual does not cover CAN-bus network programming and system design.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When you need help

This manual and all the example programs will provide you with enough information to fully utilize all the features on this board. If you have any problems with installation or use of the board, contact our Technical Support Department (814) 234-8087. Alternatively, send Email to: **tech-support@rtd.com** When sending an Email request, please include the following information: Your company's name and address, your name, your telephone number, and a brief description of the problem.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 2 BOARD SETTINGS

The ECAN527DHR CAN bus interface board has jumper settings which can be changed to suit your application and host computer memory configuration. The factory settings are listed and shown in the diagram at the beginning of this chapter.

Factory-Configured Jumper Settings

Table 1 below illustrates the factory jumper setting for the ECAN527DHR. Figure1 shows the board layout of the ECAN527DHR and the locations of the jumpers. The following paragraphs explain how to change the factory jumper settings to suit your specific application.

Table 1: Factory configured jumper settings
(Please refer to figure 1 below for detailed locations)

JUMPER NAME	DESCRIPTION	NUMBER OF JUMPERS	FACTORY SETTING
ADDR	Base Addresses	10	D0000
IRQ'S	Host interrupts	10 + 10	5/11
J34/J52	Can-bus termination	1+1	Closed

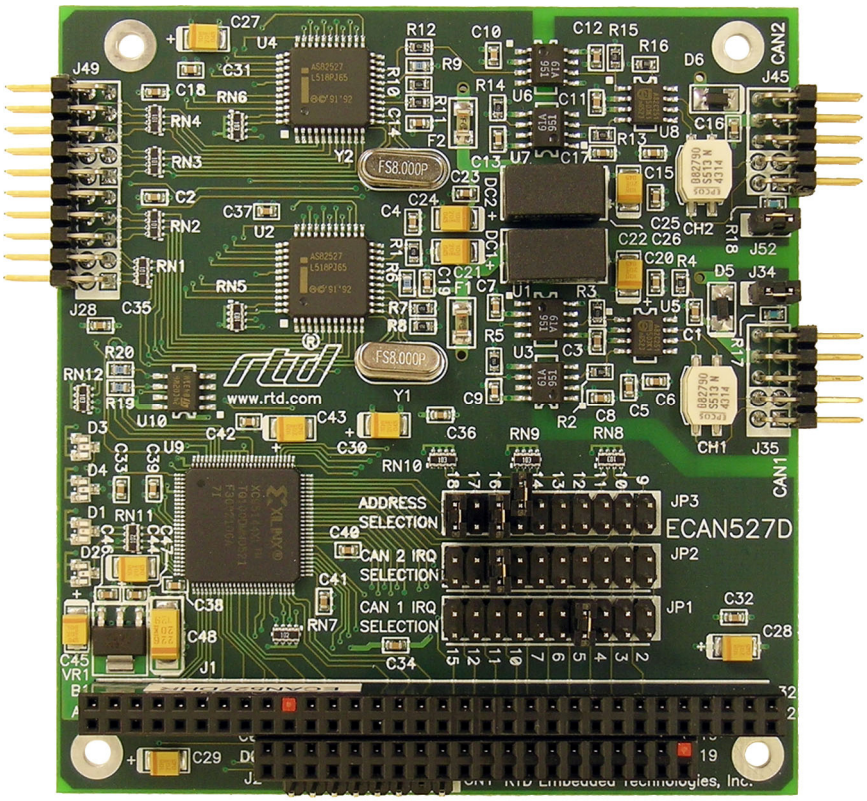


Figure 1: ECAN527DHR Board layout showing jumper locations

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Base address Jumpers (Factory setting: D0000h)

The ECAN527DHR is memory mapped into the low memory of your host XT/AT. The board occupies a consecutive memory window of 512 bytes starting from the base address. The first 256 bytes are for the first CAN controller and the second 256 bytes for the second CAN controller.

The most common cause of failure when you are first setting up your module is address contention: Some of your computers I/O space is already occupied by other devices and memory resident programs. When the ECAN527DHR attempts to use its own reserved memory addresses (which are being already used by another peripheral device) erratic performance may occur and the data read from the board may be corrupted.

To avoid this problem make sure you set up the base address by using the ten jumpers marked “ADDR”. This allows you to choose from a number of different addresses in your host computer’s memory map. Should the factory installed setting of D0000h be incompatible to your system configuration, you may change this setting to another using the options illustrated in Table 2 (overleaf). The table shows the jumper settings and their corresponding values in hexadecimal form. Ensure that you verify the correct location of the base address jumpers. When the jumper is removed it corresponds to a logical "0", connecting the jumper to a "1". When you set the base address of the module, record the setting inside the back cover of this manual

Note:	If you are using a memory manager such as EMM386, make sure that you exclude the memory section occupied by the ECAN527DHR. For example, this may be X= D0000-D00FF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BASE ADDRESS JUMPER SETTINGS ECAN527DHR

Base address Hex	Jumper Settings 18 17 16 15
80XXX	0 0 0 0
88XXX	0 0 0 1
90XXX	0 0 1 0
98XXX	0 0 1 1
A0XXX	0 1 0 0
A8XXX	0 1 0 1
B0XXX	0 1 1 0
B8XXX	0 1 1 1
C0XXX	1 0 0 0
C8XXX	1 0 0 1
D0XXX	1 0 1 0
D8XXX	1 0 1 1
E0XXX	1 1 0 0
E8XXX	1 1 0 1
F0XXX	1 1 1 0
F8XXX	1 1 1 1
0 = JUMPER OFF	**1 = JUMPER CLOSED**

Table 2: Base Address Jumper settings ECAN527DHR

Note: The above table only illustrates the settings for the high address bits A18-A15. If you for instance want to configure address D8E00 - D8EFF for your ECAN527DHR you must set the jumpers as: 1 0 1 _ 1 0 0 0 _ 1 1 1 0! Address line A19 is always decoded as "1".

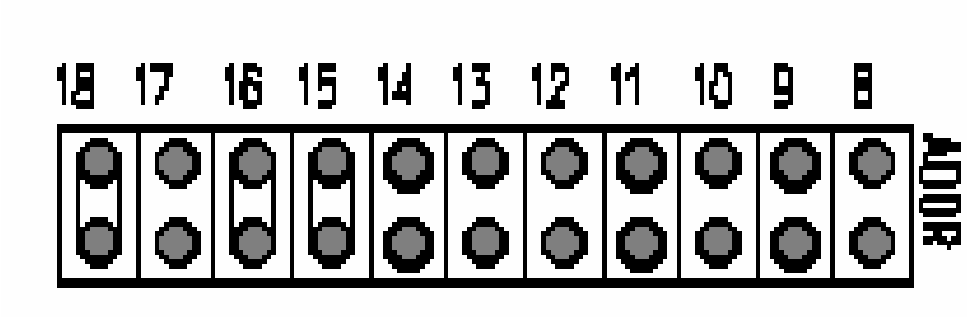


Figure 2: Base address jumpers illustrating address D8000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Interrupt Channels (Factory setting: IRQ5 / IRQ11)

The header connector, shown in Figure 3 below, lets you connect the onboard 82527 CAN controllers interrupt outputs to one of the interrupt channels available on the host XT/AT bus. If your board has no AT extension interrupts then IRQ 10-15 are not available. Both CAN-controllers must use different interrupts and therefore two headers are available. IRQ1 marks the Channel 1 interrupt and IRQ2 marks the interrupt for Channel 2.

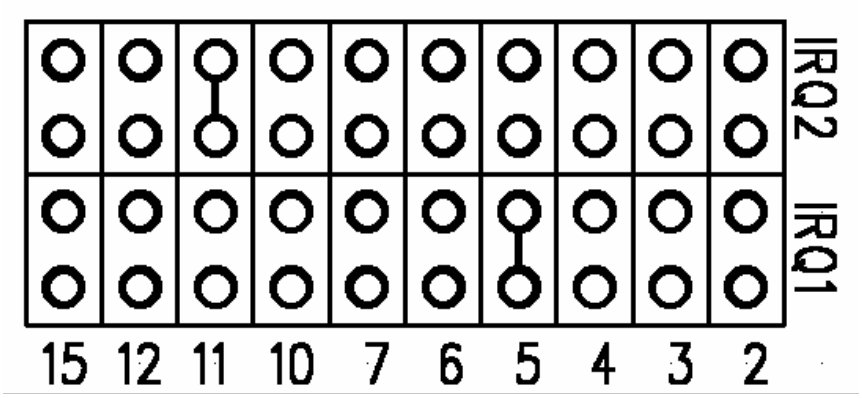


Figure 3: Interrupt jumpers

Note:	Current hardware revisions of the ECAN527D do not support interrupt sharing!

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3 BOARD INSTALLATION

The ECAN527DHR CAN-bus interface board is very easy to connect to your industrial distributed control system. Direct interface to PC/104 systems as well as EUROCARD boards is possible. This chapter gives step by step instructions on how to install the ECAN527DHR into your system.

After completing the installation it is recommended that you use the diagnostic and test software to fully verify that your board is working.

Board Installation

Keep your board in the antistatic bag until you are ready to install it to your system! When removing it from the bag, hold the board at the edges and do not touch the components or connectors. Please handle the board in an antistatic environment and use a **grounded** workbench for testing and handling of your hardware. Before installing the board in your computer, check the jumper settings. Chapter 2 reviews the factory settings and how to alter them. If any alterations are needed, please refer to the appropriate instructions in this chapter. Do however note that incompatible settings can result in unpredictable board operation and erratic response.

General installation guidelines:

- ***Turn OFF the power to your computer and all devices connected to the ECAN527DHR.***
- Touch the grounded metal housing of your computer to discharge any antistatic build-up and then remove the board from its antistatic bag.
- Hold the board by the edges and install it in an enclosure or place it on the table on an antistatic surface.
- Connect the board to the CAN field-bus using the two field-bus interface header connectors. Make sure that the orientation of the cables is correct and the same.

Installation integrated with a PC/104 module stack

- Secure the four PC/104 installation holes with standoffs.
- Connect the board to the CAN bus using the two field-bus interface header connectors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

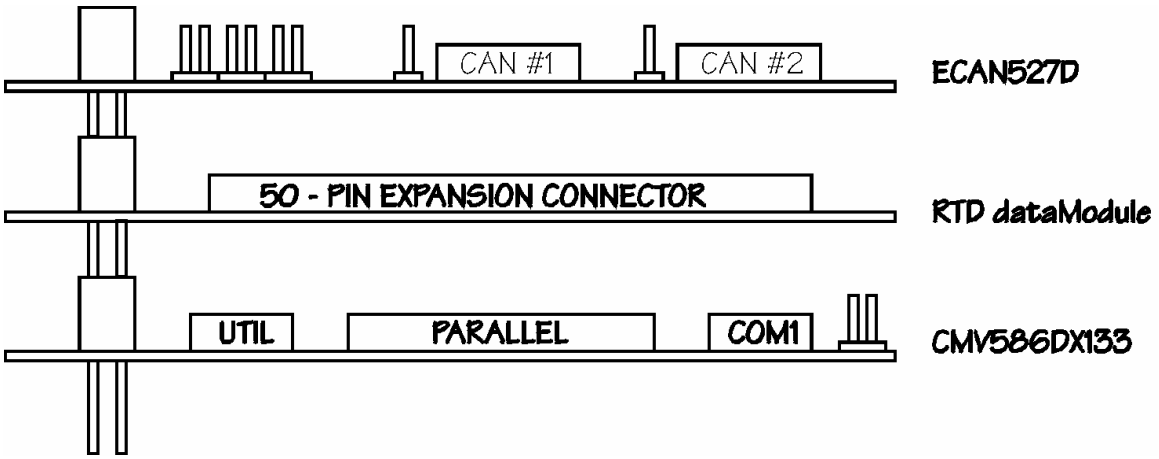


Figure 4: ECAN527DHR integrated in a PC/104 RTD cpuModule stack

3U rack or enclosure installation with a EUROCARD CPU containing an ECAN527DHR

The PC/104 system can be easily inserted into a 19" rack installation using the CPU as a "form factor adapter". Assemble your PC/104 data modules on an RTD single board EUROCARD computer and install the system in a 19" enclosure. Multiple ECAN527DHR boards can be easily connected to this system. See figure 5 below.

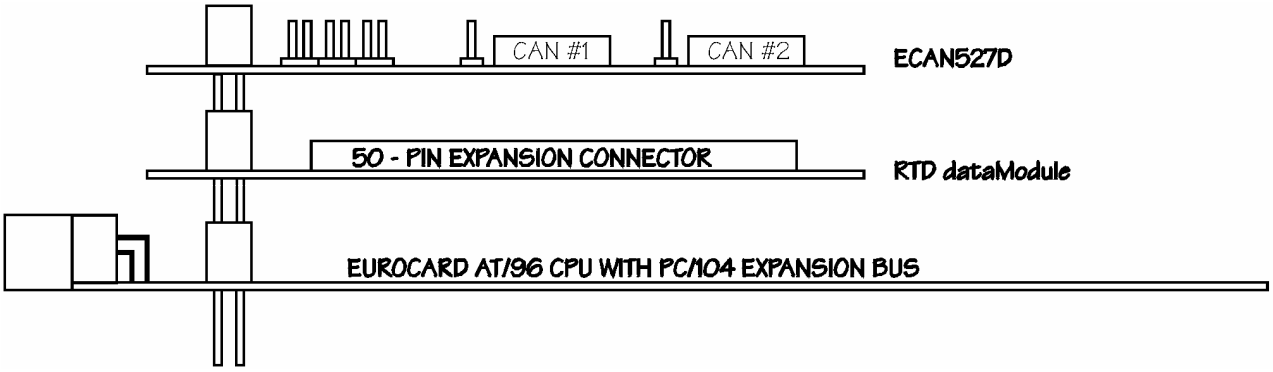


Figure 5: 19" Eurocard rack installation with an integrated PC/104 dataModule and EUROCARD cpuModule computer system

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

External I/O Connections

General purpose digital I/O connector

Table 3 below shows the connector layout of the ECAN527DHR digital I/O connectors J28 and J49. These connectors are located towards the left of the board. Refer to this diagram when making signal connections.

PIN number	Function (Channel 1)
1	P2.0 / LED1
2	P2.1 / LED2
3	P2.2 EEPROM Clock
4	P2.3 EEPROM Data
5	P2.4
6	P2.5
7	P2.6
8	P2.7
9	+5V
10	GND

Table 3: Digital I/O connector J28 of the ECAN527DHR

Note: I/O signals 3 and 4 are reserved for the use of the non-volatile EEPROM. In this case these pins are not available for user I/O

PIN number	Function (Channel 2)
1	P2.0 / LED1
2	P2.1 / LED2
3	P2.2
4	P2.3
5	P2.4
6	P2.5
7	P2.6
8	P2.7
9	+5V
10	GND

Table 4: Digital I/O connector J49 of the ECAN527DHR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Galvanically isolated CAN bus connectors

Table 5 below shows the identical CAN physical interface connectors. These connectors are to the right hand side of your board. The pin-outs conform to the ISO 11898 standard specification.

PIN number	Function
1	N.C.
2	GND_isolated
3	BUS_L
4	BUS_H
5	GND_isolated
6	N.C.
7	N.C.
8	+5V-isolated
9	GND-isolated
10	N.C.

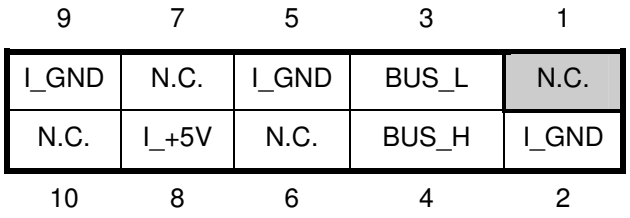


Table 5: Physical interface connector J35 / J45 pin-outs of the ECAN527DHR

Galvanically isolated CAN bus Termination Jumpers

The jumpers marked as J34 and J53 are CAN bus termination jumpers. Only two termination jumpers should be closed at the endpoints of the CAN bus. Failure to do so may degrade the performance of the bus or even cause permanent damage to the driver chips. The typical drive capability of the transceiver is 32 nodes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4 - HARDWARE DESCRIPTION

This chapter describes in detail the major features of the ECAN527DHR:

- The Intel 82527 CAN bus controller
- Galvanic isolation of the CAN bus
- The onboard configuration EEPROM
- Digital I/O

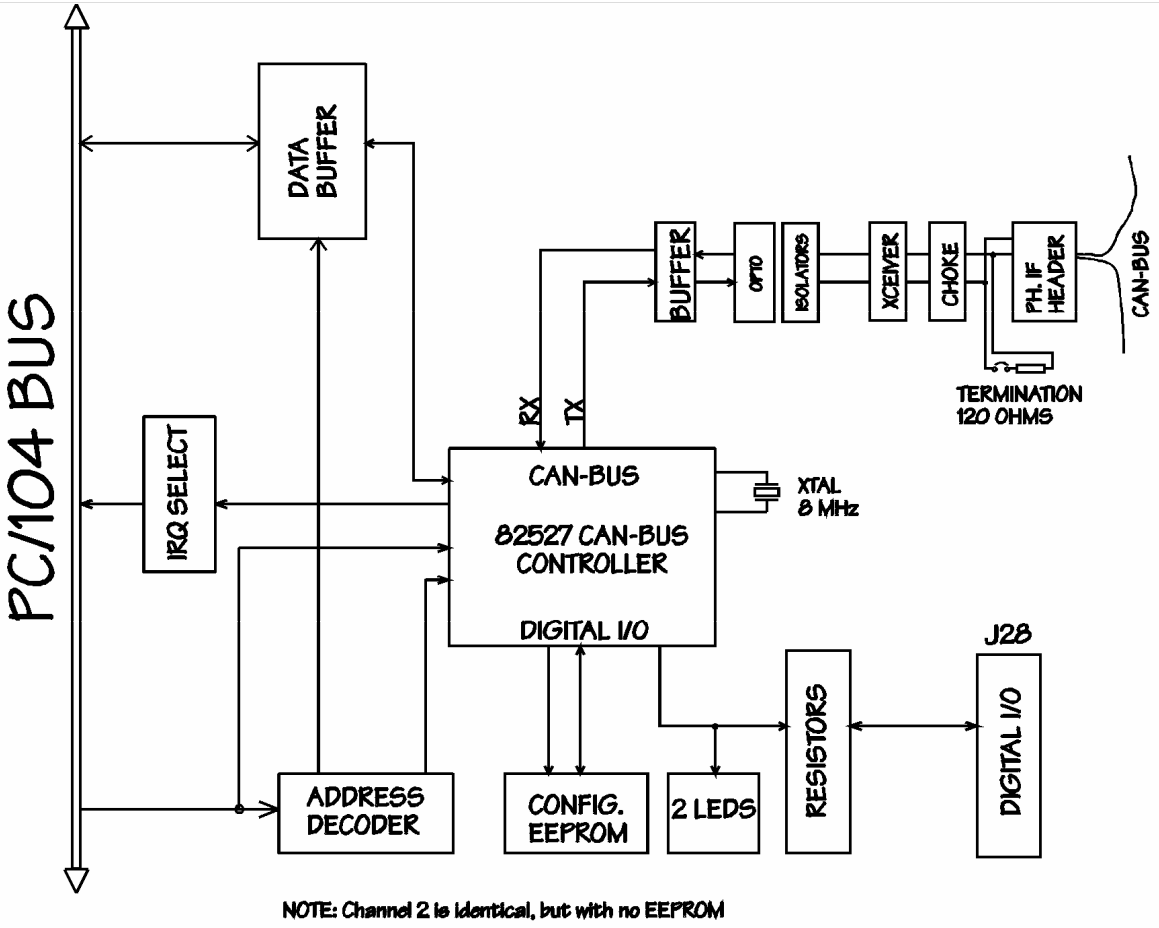


Figure 6: ECAN527DHR Block diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

82527 CAN bus controller

Reference note: (Intel publication CAN Architectural Overview, Automotive Products Data book)

The 82527 CAN controller consists of six functional blocks. The function of the CPU interface logic is to manage the communication to the host computer. The CAN controller interfaces to the CAN bus and implements the rules of the CAN protocol for the transmission and reception of messages. The RAM is the physical interface layer between the host CPU and the CAN bus. One eight-bit I/O port provides low speed I/O capabilities.

The 82527 RAM provides storage for 15 message objects of 8-byte length. Each message object has a unique identifier and can be configured to either transmit or to receive except for the last message object. The last message object is a receive-only buffer with a special mask design to allow selected groups of different message identifiers to be received.

Each message identifier contains control and status bits. A message object with a direction set for receive will send a remote frame by requesting a message transmission. A message set as transmit will be configured to automatically send a data frame whenever a remote frame with a matching identifier is received over the CAN bus. All message objects have separate transmit and receive interrupts and status bits, allowing the CPU full flexibility in detecting when a remote frame has been sent or received.

The 82527 chip also supports a global masking feature for acceptance filtering. This feature allows the user to globally mask, or "don't care", any identifier bits of the incoming message. This mask is programmable to allow the user to design application specific message identification strategies. There are separate global masks for standard and extended frames.

The incoming message first passes through the global mask and is then matched to the identifiers in the message objects 1-14. If there is no identifier match then the message passes through the local mask in message object 15. The local mask allows a large number of infrequent messages to be received by the 82527. Message object 15 is also buffered to allow the CPU time to service a message received.

Galvanic isolation of the CAN-bus

The galvanic isolation of the ECAN527DHR is implemented using the following:

- Optocouplers for data transmission
- DC/DC converter to supply power to the CAN bus and the physical interface circuitry.

The high-speed optocouplers are directly connected to the 82527. The optocouplers drive the CAN bus transceiver. A special balanced CAN bus choke is used not only to improve immunity to bus noise, but also to protect the bus transceiver. This choke also reduces the radiated emissions in the range of 30-200Mhz.A 1W DC/DC converter may be used to power other remote devices on the CAN bus. The output power of this converter is isolated up to 1.5 kV peak. 125mA fuses (green) are used to protect the DC/DC converters.

The jumpers marked as J34 and J53 are CAN bus termination jumpers. Only two termination jumpers should be closed at the endpoints of the CAN bus. Failure to do so may degrade the performance of the bus or even cause permanent damage to the driver chips. The maximum drive of the transceiver is 32 nodes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration EEPROM

An onboard configuration EEPROM may store up to 256 bytes of user data. This memory may be used to store CAN bus parameters or other identifier codes required by the system.

CAN controller 1. Data is read and written by clocking these lines to provide the serial interface required by the EEPROM device. See ECAN527DHR Specifications (Chapter 6) for more exact descriptions on EEPROM interfacing. The EEPROM used the I2C interface scheme with the device address set to 000.

Digital I/O

The tables 6 and 7 below show the functionality of the 82527 port P2 bits. Bits 0 and 1 have two functions. They drive the two onboard status-LED's and operate as user TTL-level I/O. The output impedance of this port is 100 Ohms. This series resistor on every I/O line provides protection against short circuits and transient voltages.

P2.0	**P2.1**	**P2.2**	**P2.3**	**P2.4**	**P2.5**	**P2.6**	**P2.7**
LED1	LED2	CLK	DATA	DIO3	DIO4	DIO5	DIO6

Table 6: Digital I/O configuration of port P2 of the 82527 chip number 1

P2.0	**P2.1**	**P2.2**	**P2.3**	**P2.4**	**P2.5**	**P2.6**	**P2.7**
LED1	LED2	DIO3	DIO4	DIO5	DIO6	DIO7	DIO8

Table 7: Digital I/O configuration of port P2 of the 82527 chip number 2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 5 BOARD OPERATION AND PROGRAMMING

This chapter shows you how to program and use your ECAN527DHR. It provides a complete detailed description of the memory map and a detailed discussion of programming operations to aid you in programming. The full functionality of the ECAN527DHR is described in the attached data sheet reprint from Intel on the 82527 CAN controller chip.

Defining the Memory Map

The memory map of the ECAN527DHR occupies 512 bytes of host PC low memory space. CAN controller number 1 occupies addresses BASE to BASE+FFh, while CAN controller number 2 occupies addresses BASE+100h to BASE+1FFh. This window is freely selectable by the user as described in *Chapter 2, Table 2.* After setting the base address you have access to the internal resources of the 82527 CAN controller chips. These resources are described in the next sections reprinted from the Intel 82527 manual.

Global Reset

An additional feature is that both CAN controller chips can be held in reset by writing a 0 to BASE + 4h or BASE + 104h. The reset will be released by writing a 0 to BASE + 5h or BASE + 105h. A write to either address will reset both CAN controllers.

Individual Reset

A write of 55h to BASE + 4h will hold CAN controller 1 only in reset. A write of 55h to BASE + 5h will release the reset on CAN controller 1 only.

A write of 55h to BASE + 104h will hold CAN controller 2 only in reset. A write of 55h to BASE + 105h will release the reset on CAN controller 2 only.

The reset must be asserted for 1 millisecond minimum. The resets can be used in any combination and sequence.

INTERRUPTS

What is an interrupt?

An interrupt is an event that causes the processor in your computer to temporarily halt its current process and execute another routine. Upon completion of the new routine, control is returned to the original routine at the point where its execution was interrupted.

Interrupts are a very flexible way of dealing with asynchronous events. Keyboard activity is a good example; your computer cannot predict when you might press a key and it would be a waste of processor time to do nothing whilst waiting for a keystroke to occur. Thus the interrupt scheme is

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

used and the processor proceeds with other tasks. When a keystroke finally occurs, the keyboard then 'interrupts' the processor so that it can get the keyboard data .It then places it into the memory, and then returns to what it was doing before the interrupt occurred. Other common devices that use interrupts are A/D boards, network boards, serial ports etc.

Your ECAN527DHR can interrupt the main processor when a message is received or transmitted if interrupts are enabled on the ECAN527DHR board. By using interrupts you can write powerful code to interface to your CAN network.

Interrupt request lines

To allow different peripheral devices to generate interrupts on the same computer, the PC AT bus has interrupt request channels (IRQs). A rising edge transition on one of these lines will be latched into the interrupt controller. The interrupt controller checks to see if the interrupts are to be acknowledged from that IRQ and, if another interrupt is being processed, it decides if the new request should supersede the one in progress or if it has to wait until the one in progress has been completed. The priority level of the interrupt is determined by the number of the IRQ as follows; IRQ0 has the highest priority whilst IRQ15 has the lowest. Many of the IRQs are used by standard system resources, IRQ0 is dedicated to the internal timer, IRQ1 is dedicated to the keyboard input, IRQ3 for the serial port COM2, and IRQ4 for the serial port COM1. Often interrupts 5,7,11,15 are often free.

8259 Programmable Interrupt Controller

The chip responsible for handling interrupt requests in a PC is the 8259 Interrupt Controller. To use interrupts you will need to know how to read and set the 8259's internal interrupt mask register (IMR) and how to send the end-of-interrupt (EOI) command to acknowledge the 8259 interrupt controller.

Interrupt Mask Register (IMR)

Each bit in the interrupt mask register (IMR) contains the mask status of the interrupt line. If a bit is set (equal to 1), then the corresponding IRQ is masked, and it will not generate an interrupt. If a bit is cleared (equal to 0), then the corresponding IRQ is not masked, and it can then generate an interrupt. The interrupt mask register is programmed through **port 21h**.

End-of-Interrupt (EOI) Command

After an interrupt service routine is complete, the 8259 Interrupt Controller must be acknowledged by writing the value 20h to port 20h.

What exactly happens when an interrupt occurs?

Understanding the sequence of events when an interrupt is triggered is necessary to correctly write interrupt handlers. When an interrupt request line is driven high by a peripheral device (such as the ECAN527DHR), the interrupt controller checks to see if interrupts are enabled for that IRQ.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

It then checks to see if other interrupts are active or requested and determines which interrupt has priority. The interrupt controller then interrupts the processor. The current code segment (CS), instruction pointer (IP), and flags are pushed onto the system stack, and a new set if CS and IP are loaded from the lowest 1024 bytes of memory.

This table is referred to as the interrupt vector table and each entry to this table is called an interrupt vector. Once the new CS and IP are loaded from the interrupt vector table, the processor starts to execute code from the new Code Segment (CS) and from the new Instruction Pointer (IP). When the interrupt routine is completed the old CS and IP are popped from the system stack and the program execution continues from the point where interruption occurred.

Using Interrupts in your Program

Adding interrupt support to your program is not as difficult as it may seem especially when programming under DOS. The following discussion will cover programming under DOS. Note that even the smallest mistake in your interrupt program may cause the computer to hang up and will only restart after a reboot. This can be frustrating and time-consuming.

Writing an Interrupt Service Routine (ISR)

The first step in adding interrupts to your software is to write an interrupt service routine (ISR). This is the routine that will be executed automatically each time an interrupt request occurs for the specified IRQ. An ISR is different from other sub-routines or procedures. First, on entrance the processor registers must be pushed onto the stack before anything else! Second, just before exiting the routine, you must clear the interrupt on the ECAN527DHR by writing to the 82527 CAN controller, and write the EOI command to the interrupt controller. Finally, when exiting the interrupt routine the processor registers must be popped from the system stack and you must execute the IRET assembly instruction. This instruction pops the CS, IP and processor flags from the system stack. These were pushed onto the stack when entering the ISR.

Most compilers allow you to identify a function as an interrupt type and will automatically add these instructions to your ISR with one exception: most compilers do not automatically add the EOI command to the function, you must do it yourself. Other than this and a few exceptions discussed below, you can write your ISR as any code routine. It can call other functions and procedures in your program and it can access global data. If you are writing your first ISR, we recommend you stick to the basics; just something that enables you to verify you have entered the ISR and executed it successfully. For example: set a flag in your ISR and in your main program check for the flag.

Note: If you choose to write your ISR in in-line Assembly, you must push and pop registers correctly and exit the routine with the IRET instruction instead of the RET instruction.

There are a few precautions you must consider when writing ISRs. The most important is, **do not use any DOS functions or functions that call DOS functions from an interrupt routine.** DOS is not re-entrant; that is, a DOS function cannot call itself. In typical programming, this will not happen because of the way DOS is written. But what about using interrupts? Consider then the following situation in your program: If DOS function X is being executed when an interrupt occurs and the interrupt routine makes a call to the same DOS function X, then function X is essentially being called while active. Such cases will cause the computer to crash. DOS does not support

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

such operations. The general rule is that do not call any functions that use the screen, read keyboard input or any file I/O routines. These should not be used in ISRs.

The same problem of reentrancy also exists for many floating-point emulators. This effectively means that you should also avoid floating point mathematical operations in your ISR.

Note that the problem of reentrancy exists, no matter what programming language you use. Even, if you are writing your ISR in Assembly language, DOS and many floating point emulators are not re-entrant. Of course there are ways to avoid this problem, such as those which activate when your ISR is called. Such solutions are, however, beyond the scope of this manual.

The second major concern when writing ISRs is to make them as short as possible in term of execution time. Spending long times in interrupt service routines may mean that other important interrupts are not serviced. Also, if you spend too long in your ISR, it may be called again before you have exited. This will lead to your computer hanging up and will require a reboot.

Your ISR should have the following structure:

- Push any processor registers used in your ISR. Most C compiler do this automatically
- Put the body of your routine here
- Read interrupt status register of the 82527 chip on your ECAN527DHR board
- Clear the interrupt bit by writing to the 82527 CAN controller
- Issue the EOI command to the 8259 by writing 20h to address 20h
- Pop all registers. Most C compilers do this automatically

The following C example shows what the shell of your ISR should be like:

```
/*---------------------------------------------------------------------------
| Function:        new_IRQ_handler
| Inputs:Nothing
| Returns:         Nothing - Sets the interrupt flag for the EVENT.
|---------------------------------------------------------------------------*/
void interrupt far new_IRQ_handler(void)
{

        IRQ_flag = 1;               // Indicate to main process interrupt has occurred
        {
        // Your program code should be here
        }
                                    // Read interrupt status registers
                                    // Clear the interrupt on ECAN527DHR
        outp(0x20, 0x20);           /* Acknowledge the interrupt controller. */
}
```

Saving the Startup Interrupt Mask Register (IMR) and interrupt vector

The next step after writing the ISR is to save the start-up state of the interrupt mask register (IMR) and the original interrupt vector you are using. The IMR is located in address **21h**. The interrupt vector you will be using is located in the interrupt vector table which is an array of pointers (addresses) and it is locate din the first 1024 bytes of the memory (Segment 0 offset 0). You can read

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

this value directly, but it is better practice to use DOS function 35h (get interrupt vector) to do this. Most C compilers have a special function available for doing this. The vectors for the hardware interrupts on the XT - bus are vectors 8-15, where IRQ0 uses vector 8 and IRQ7 uses vector 15. Thus if your ECAN527DHR is using IRQ5 it corresponds to vector number 13.

Before you install your ISR, temporarily mask out the IRQ you will be using. This prevents the IRQ from requesting an interrupt while you are installing and initializing your ISR. To mask the IRQ, read the current IMR at I/O port 21h, and set the bit that corresponds to IRQ. The IMR is arranged so that bit 0 is for IRQ0 and bit 7 is for IRQ7. See the paragraph entitled *Interrupt Mask Register (IMR*) earlier in this discussion for help in determining your IRQ's bit. After setting the bit, write the new value to I/O port 21h.

With the startup IMR saved and the interrupts temporarily disabled, you can assign the interrupt vector to point to your ISR. Again you can overwrite the appropriate entry in the vector table with a direct memory write, but this is not recommended. Instead use the DOS function 25h (Set Interrupt Vector) or, if your compiler provides it, the library routine for setting up interrupt vectors. Remember, that interrupt vector 8 corresponds to IRQ0, vector 9 for IRQ1 etc.

If you need to program the source of your interrupts, do that next. For example, if you are using transmitted or received messages as an interrupt source, program it to do that. Finally, clear the mask bit for your IRQ in the IMR. This will enable your IRQ.

Common Interrupt mistakes

Remember that hardware interrupts are from 8-15, XT IRQs are numbered 0-7Forgetting to clear the IRQ mask bit in the IMR Forgetting to send the EOI command after ISR code. Disables further interrupts.

Example on Interrupt vector table setup in C-code:

```
void far _interrupt new_IRQ1_handler(void );                /* ISR function prototype */
#define IRQ1_VECTOR 3                                       /* Name for IRQ */
void (interrupt far *old_IRQ1_dispatcher)
        (es,ds,di,si,bp,sp,bx,dx,cx,ax,ip,cs,flags);        /* Variable to store old IRQ_Vector */
void far _interrupt new_IRQ1_handler(void );

/*-------------------------------------------------------------------------
| Function:     init_irq_handlers
| Inputs:Nothing
| Returns:      Nothing
| Purpose:      Set the pointers in the interrupt table to point to
|               our funtions ie. setup for ISR's.
|-------------------------------------------------------------------------*/
void init_irq_handlers(void)
{
        _disable();
                old_IRQ1_handler = _dos_getvect(IRQ1_VECTOR + 8);
                _dos_setvect(IRQ1_VECTOR + 8, new_IRQ1_handler);
                Gi_old_mask = inp(0x21);
                outp(0x21,Gi_old_mask & ~(1 << IRQ1_VECTOR));
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
        _enable();
}

|/*---------------------------------------------------------------------
| Function:     restore do this before exiting program
| Inputs:Nothing
| Returns:      Nothing
| Purpose:      Restore interrupt vector table.
|---------------------------------------------------------------------*/
void restore(void)
{
        /* Restore the old vectors */

        _disable();

                _dos_setvect(IRQ1_VECTOR + 8, old_IRQ1_handler);
                outp(0x21,Gi_old_mask);

        _enable();

}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 6 ECAN527DHR SPECIFICATIONS

Host Interface

Memory mapped into low memory, occupies 512 bytes
Jumper-selectable base address
8-bit data bus, 16-bit AT bus connector
Jumper selectable interrupts XT and AT

CAN Interfaces

Galvanically isolated transceiver 1 Mb/s data-rate
Timing parameters and speed of bus programmable
Balanced CAN-bus Choke
Jumper selectable 120 Ohm termination resistors
Transceiver drive: 32 nodes
0.8W 5V isolated and fused output power for other field devices

Digital I/O (non isolated)

Number of lines	16
LED's	4
Series resistor	100 Ohms

Connectors

Galvanically isolated CAN bus	Two 10-pin headers
Non Isolated I/O's	10-pin header connectors
Host bus	XT/AT PC/104 bus

Electrical

Operating temperature range	-40 to +85 C
Supply voltage	+5V, +-8%
Power consumption	1.2W typical

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IDAN System Module Dimensions

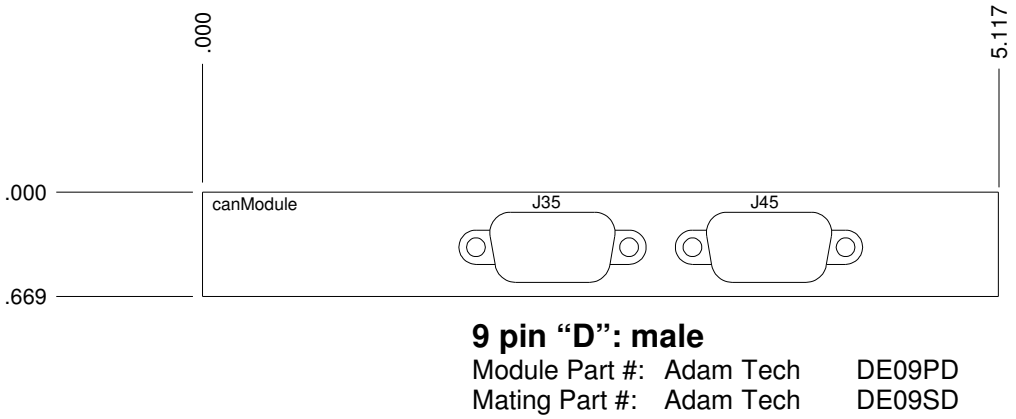


Figure 7 IDAN Dimensions

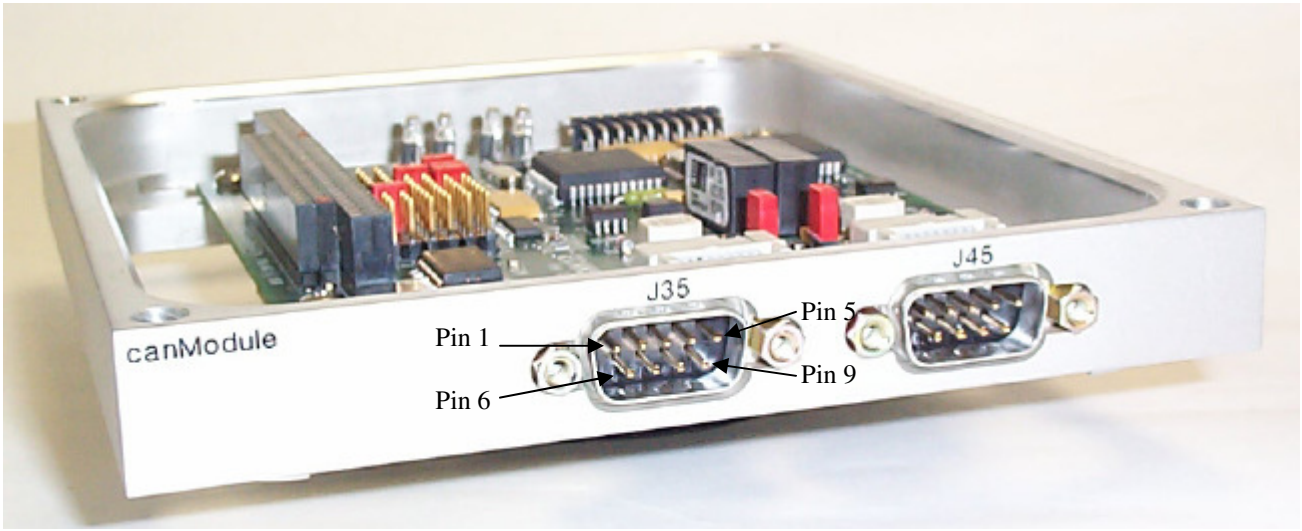


Figure 8 IDAN Configuration

Drawing not to scale.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IDAN System Module Pinout

Table 8 IDAN Pinout

J35, J45 - 9 Pin "D" Connector (male)		
IDAN Pin #	**Signal**	**Name**
1	RESERVED	RESERVED
2	BUS_L	Bus Low
3	GND	GND_isolated
4	RESERVED	RESERVED
5	GND	GND_isolated
6	GND	GND_isolated
7	BUS_H	Bus High
8	RESERVED	RESERVED
9	+5 VDC	+5V_isolated

Do not make connections to RESERVED pins.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 8 RETURN POLICY AND WARRANTY

Return Policy

If you wish to return a product to the factory for service, please follow this procedure:

Read the Limited Warranty to familiarize yourself with our warranty policy.

Contact the factory for a Return Merchandise Authorization (RMA) number.

Please have the following available:

- Complete board name
- Board serial number
- A detailed description of the board's behavior

List the name of a contact person, familiar with technical details of the problem or situation, **along with their phone and fax numbers, address, and e-mail address** (if available).

List your shipping address!!

Indicate the shipping method you would like used to return the product to you.
We will not ship by next-day service without your pre-approval.

Carefully package the product, using proper anti-static packaging.

Write the RMA number in large (1") letters on the outside of the package.

Return the package to:

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College PA 16803-0906
USA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Limited Warranty

RTD Embedded Technologies, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from RTD Embedded Technologies, INC. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD Embedded Technologies will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to RTD Embedded Technologies. All replaced parts and products become the property of RTD Embedded Technologies. Before returning any product for repair, customers are required to contact the factory for an RMA number.

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED AS A RESULT OF ACCIDENT, MISUSE, ABUSE (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by RTD Embedded Technologies, "acts of God" or other contingencies beyond the control of RTD Embedded Technologies), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN RTD Embedded Technologies. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES ARE EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND RTD Embedded Technologies EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MECHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE DURATION OF THIS WARRANTY. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, THE PURCHASER'S SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL RTD Embedded Technologies BE LIABLE TO THE PURCHASER OR ANY USER FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP