

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 *Excalibur Systems M4KCAN4* at our website: ***Click HERE***

M4KCAN

Test and Simulation Module
for the EXC-4000
Family of Carrier Boards

User's Manual

EXCALIBUR SYSTEMS

311 Meacham Ave ♦ Elmont ♦ NY 11003 ♦ tel. (516) 327-0000 ♦ fax (516) 327-4645
e:mail: excalibur@mil-1553.com website: www.mil-1553.com

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

Table of Contents

Figures

Figure 1-1 *M4KCAN* Module – Block Diagram 1-3
Figure 2-1 *M4KCAN* General Memory Map 2-3
Figure 3-1 *M4KCAN* Channel Memory Map 3-2
Figure 3-2 CAN Specification Bit Timing 3-13
Figure 3-3 *M4KCAN* Bit Timing Definitions 3-14
Figure 5-1 *M4KCAN* Module Layout 5-1
Figure 5-2 Terminal Stick Connector: Front View 5-3
Figure 5-3 Adapter Cable 26-pin Connector: Front View 5-5
Figure A-1 Standard CAN Message Format A-2
Figure A-2 Extended CAN Message Format A-2
Figure A-3 Error Frame Format A-3
Figure A-4 Overload Frame Format A-5
Figure A-5 Interframe Space Format A-5

Tables

Table 5-1 DIP Switch Settings Required to Select Termination Mode 5-2
Table 5-2 Bus DIP Switch 5-2
Table 5-3 Factory Default DIP Switch Settings 5-3
Table 5-4 CAN Connector Pin Assignment Configuration 5-4
Table 5-5 *M4KCAN* Adapter Cable Connections 5-5
Table A-1 CAN Standard and Extended Message Formats A-2

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

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

1 Introduction

Chapter 1 provides an overview of the *M4KCAN* avionics communication module. The following topics are covered:

1.1 Overview 1-1
1.1.1 *M4KCAN* Module Features 1-2
1.1.2 *M4KCAN* Module Block Diagram 1-3
1.2 Installation 1-4
1.2.1 Module Installation 1-4
1.3 VME/VXI Byte Swapping 1-5
1.4 Technical Support 1-5

1.1 Overview

The *M4KCAN* is an intelligent serial communications interface module for the multimode, multiprotocol Excalibur EXC-4000 family of carrier boards. The *M4KCAN* provides a complete solution for developing and testing CAN interfaces and performing system simulation of the CAN bus. It is a highly integrated module that performs serial communication according to the CAN protocol.

Each module of the *M4KCAN* contains up to 6 channels each containing 256 bytes of dual-port RAM for Data Blocks and control registers. All Data Blocks and control registers are memory mapped, and may be accessed in real time.

The CAN protocol uses a multi-master (contention based) bus configuration for the transfer of "communication objects" between nodes of the network. This multi-master bus is also referred to as CSMA/CR or Carrier Sense, Multiple Access, with Collision Resolution. The *M4KCAN* performs all serial communication functions such as transmission and reception of messages, message filtering, transmit search, and interrupt search with minimal interaction from the user.

M4KCAN also performs error handling without any burden on the user's computer, featuring several error detection mechanisms such as Cyclical Redundancy Check (CRC) and bit coding rules ("bit stuffing/destuffing"). The polynomial of the CRC has been optimized for control applications with short messages. If a message was corrupted by noise during transmission, it is not accepted at the receiving nodes. Current transmission status is monitored in the control segment of the appropriate communication object within the transmitting node, automatically initiating a repeated transmission in the case of errors. *M4KCAN* also has built-in mechanisms to locate error sources and to distinguish permanent hardware failures from occasional soft errors. Defective nodes are switched off the bus, implementing a fail-safe behavior (thus, hardware errors will not let defective nodes control the bus indefinitely).

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

The *M4KCAN* supports the standard and extended message frames in CAN Specification 2.0 part B. It has the capability to transmit, receive, and perform message filtering on extended message frames with a 29-bit Message Identifier. Due to the backwardly compatible nature of CAN Specification 2.0, the *M4KCAN* also fully supports the standard message frames with an 11-bit Message Identifier as in CAN Specification 2.0 part A.

The *M4KCAN* comes complete with a C-driver software library including source code, and 4000 mating connectors with terminal sticks and plastic hoods or an adapter cable.

The *M4KCAN*-E option is an extended temperature (-40° to +85°C) version of the module.

Note: The *M4KCAN* module can be mounted on the following Excalibur carrier boards:

- *EXC-4000PCI[e]*
- *EXC-4000cPCI*
- *EXC-2000PCI[e]*
- *EXC-4000VME*
- *EXC-4000VXI*
- *EXC-4000P104plus*
- *EXC-4500ccVPX*
- *EXC-664PCIe*

1.1.1 *M4KCAN* Module Features

Global Features

Up to 6 independent CAN channels
Realtime operations
Extended Temperature range available: – 40° to +85°C

Features per Channel

Supports CAN Specification 2.0B
- Standard Data and Remote Frames
- Extended Data and Remote Frames

Programmable Global Mask
- Standard Message Identifier
- Extended Message Identifier

15 Message Objects of 8-byte Data Length
- 14 TX/RX Buffers
- 1 RX Buffer with Programmable Mask

8-bit Asynchronous Non-Multiplexed user interface
Programmable Bit Rate (up to 1 Mbps)
Flexible Interrupt Structure
Flexible Status Interface

See Chapter 6: **Ordering Information**, for the exact part numbers.

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

1.1.2 *M4KCAN* Module Block Diagram

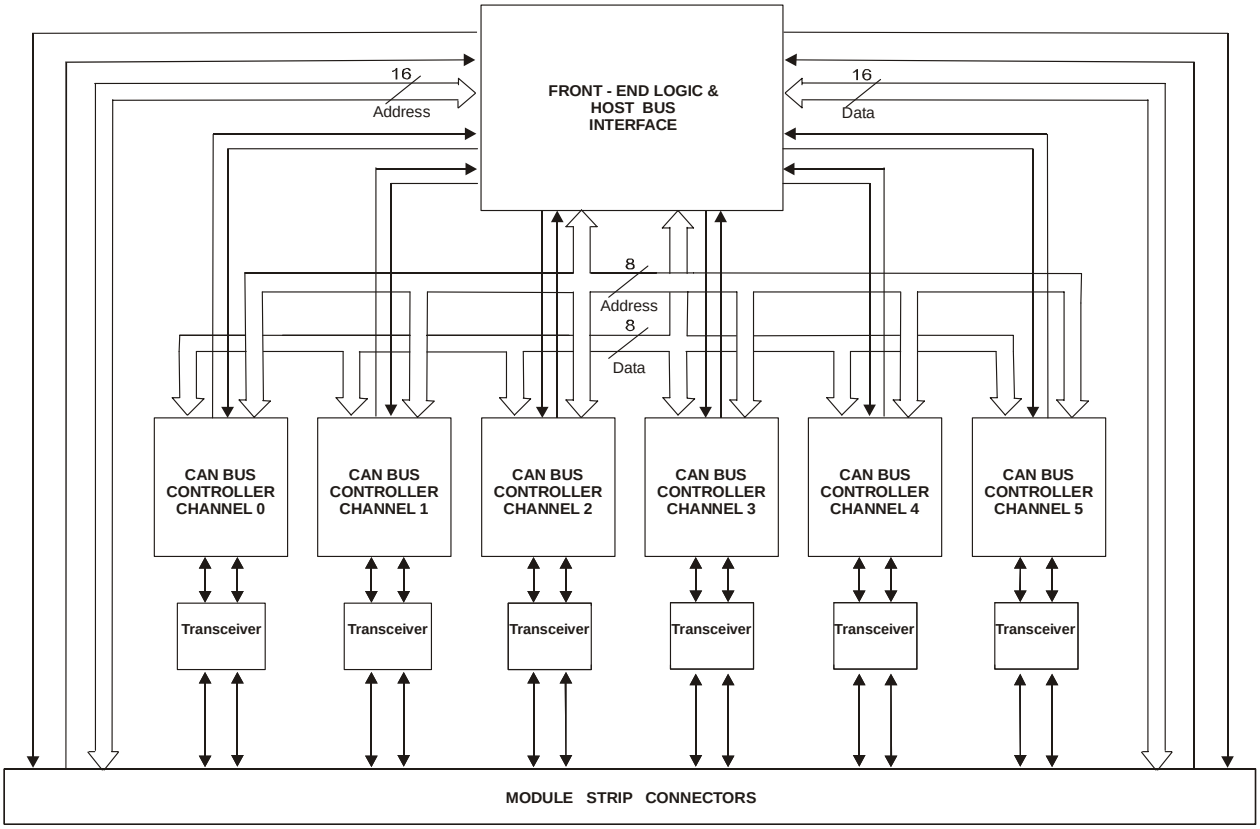


Figure 1-1 *M4KCAN* Module – Block Diagram

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

1.2 Installation

For hardware and software installation instructions, see the **readme.pdf** file on the root folder of the installation CD. When downloading new software from the Excalibur website, the **readme.pdf** file is contained in the zip file.

The *Excalibur Installation CD* you received with your package is the most recent release of the CD as of the date of shipping. Software and documentation updates can be found and downloaded from our website: www.mil-1553.com.

The standard software provided with Excalibur boards and modules is for Windows operating systems. For more details, see the **readme.pdf** file. Software for other operating systems may be available. Check on our website or write to excalibur@mil-1553.com.

1.2.1 Module Installation

Warning Wear a suitably grounded electrostatic discharge wrist strap whenever handling the Excalibur module.

1. If the module is supplied separately from the carrier board, *very carefully*, insert the module on the carrier EXC-4000 board. The pin #1 marker (marked with a white rectangle) on the module must be aligned with the white rectangles on the carrier board.
2. The *M4KCAN* allows for termination of the bus with 120 Ohm resistors via DIP switches. Use DIP switches SW1 and SW2 to set the coupling mode to the CAN bus(es). See **5.3 Module Bus Termination Select DIP Switches** on page 5-2.
3. With the computer power source switched **off**, insert the EXC-4000 carrier board with the *M4KCAN* module into a slot in the computer.
4. Attach the *M4KCAN* adapter cable to the carrier board I/O connector and to the bus. **This is very important**: The cables may be connected to or disconnected from the board while power to the computer is turned on but not while the board is transmitting over the bus.

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

1.3 VME/VXI Byte Swapping

When the module is installed on an Excalibur 4000 PCI family carrier board, all 8-bit registers and 8-bit memory locations can be found in the locations described in this manual.

When the module is installed on an *EXC-4000VME/VXI* carrier board, all 8-bit registers and 8-bit memory locations may be byte swapped. That is, during any individual 8-bit read or write operation to these registers, the even addresses may be swapped with the odd address (and vice versa) within a single 16-bit boundary (one word). This does not occur for 16-bit read/write operations.

For example:

PCI Boards		VME/VXI Boards	
Channel Register	00 H	Status Register	00 H
Status Register	01 H	Channel Register	01 H

1.4 Technical Support

Excalibur Systems is ready to assist you with any technical questions you may have. For technical support, see the Technical Support section of our website: www.mil-1553.com. You can also contact us by phone. To find the location nearest you, see the Contact section of our website.

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

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

2 *M4KCAN* Module

Chapter 2: **M4KCAN Module** describes the module operation.

2.1 *M4KCAN* Operation 2-1
2.2 *M4KCAN* General Memory Map 2-3
2.3 Global Registers 2-4
2.3.1 Interrupt Status Register 2-4
2.3.2 Interrupt Reset Register 2-4
2.3.3 FPGA Revision Register 2-4
2.4 Hardware Reset 2-4

2.1 *M4KCAN* Operation

In CAN bus communication, a transmitting CAN node broadcasts its message to all other nodes on the CAN bus. The data messages transmitted do not contain addresses of either the transmitting node, or of any intended receiving node. Instead, the content of the message is labeled by an identifier that is unique throughout the network. All other nodes on the bus receive the message and each performs an acceptance test on the identifier to determine if the message, and thus its content, is relevant to that particular node. If the message is relevant, it will be processed; otherwise it is ignored.

The Excalibur *M4KCAN* module consists of up to six channels, each of which acts as a node on a CAN bus. Every channel contains 15 Message Objects, numbered 1 – 15. A Message Object is a data buffer accompanied by instructions to the channel's CAN controller of what to do – transmit the data, or fill the buffer with data received over the network. There are four types of instructions, or messages, that a Message Object can give to its CAN controller.

In this scenario, one node transmits data on the bus, and another node sees that the identifier of the transmitted data matches its own identifier, and then receives that data.

- *Transmit Message:* The Message Object instructs the CAN controller to immediately send data over the CAN bus.
- *Receive Message:* The Message Object instructs the CAN controller that if it detects data being transmitted on the bus with a matching identifier, that data should be recorded in the data words for this Message Object.

In this scenario, often referred to as Remote, one node puts out a request on the bus asking for remote nodes to transmit data with a specific identifier. Another node sees that the identifier associated with the request matches its own identifier, and transmits the data as requested.

- *Request Update Message:* The Message Object instructs the CAN controller to transmit a request on the CAN bus, asking that data with a specific identifier be sent from a remote CAN controller.
- *Transmit Standby Message*: This Message Object instructs the CAN controller that if it detects a request for data on the bus, with a matching identifier, the data words in this Message Object should be transmitted over the CAN bus.

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

Message identifier matching is based on the actual message identifier specified in the transmit and receive message, as well as global message id masks, which are set up per channel. Message Objects are serviced on a priority basis. If multiple Message Objects are waiting to transmit, the CAN controller will first transmit the message from the lowest numbered Message Object.

Once the Message Object is set up by the user, the transmission procedure and eventual error handling is then done without any user involvement. All Message Objects have separate transmit and receive interrupts and status bits, allowing the user full flexibility in detecting when a data has been sent or received.

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

2.2 *M4KCAN* General Memory Map

The *M4KCAN* can have up to six channels. Each channel occupies 256 bytes of the modules 128 Kbytes memory space. The 256 bytes are shared between the Control Registers and the Data Blocks

Global	Reserved	01806 – 20000 H
	FPGA Rev	01804 H
	Interrupt Reset	01802 H
	Interrupt Status	01800 H
Channel 5	Reserved	01500 – 017FF H
	CAN Control Registers and Data Block	01400 – 014FF H
Channel 4	Reserved	01100 – 013FF H
	CAN Control Registers and Data Block	01000 – 010FF H
Channel 3	Reserved	00D00 – 00FFF H
	CAN Control Registers and Data Block	00C00 – 00CFF H
Channel 2	Reserved	00900 – 00BFF H
	CAN Control Registers and Data Block	00800 – 008FF H
Channel 1	Reserved	00500 – 007FF H
	CAN Control Registers and Data Block	00400 – 004FF H
Channel 0	Reserved	00100 – 003FF H
	CAN Control Registers and Data Block	00000 – 000FF H

Figure 2-1 *M4KCAN* General Memory Map

The functionality of the registers is identical in each channel.

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

2.3 Global Registers

2.3.1 Interrupt Status Register

Address: 01800 (H)
Length 16 bits

Read only The Interrupt Status register indicates which channels are currently interrupting.

Bit		Description
06 – 15	Reserved – set to 0	
05	1	Indicates that channel 5 is interrupting
04	1	Indicates that channel 4 is interrupting
03	1	Indicates that channel 3 is interrupting
02	1	Indicates that channel 2 is interrupting
01	1	Indicates that channel 1 is interrupting
00	1	Indicates that channel 0 is interrupting

Interrupt Status Register

2.3.2 Interrupt Reset Register

Address: 01802 (H)
Length 16 bits

Write only The Interrupt Reset register is used to reset the individual bits of the Interrupt Status register.

Bit		Description
06 – 15	Reserved – set to 0	
05	1	Resets channel 5 interrupt status bit
04	1	Resets channel 4 interrupt status bit
03	1	Resets channel 3 interrupt status bit
02	1	Resets channel 2 interrupt status bit
01	1	Resets channel 1 interrupt status bit
00	1	Resets channel 0 interrupt status bit

Interrupt Reset Register

2.3.3 FPGA Revision Register

Address: 01804 (H)
Length 16 bits

Read only The FPGA Revision register contains the FPGA revision of the board.

2.4 Hardware Reset

A hardware mode reset, which will reset all CAN channels on the module, is performed by writing to the 4000PCI Global registers.

See your carrier board's *User's Manual*.

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

3 Channel Operation

3.1 Channel Initialization 3-1
3.2 Channel Registers 3-2
3.2.1 Channel Register Reset Values 3-3
3.2.2 Control Register 3-4
3.2.3 Status Register 3-6
3.2.4 User Interface Register 3-8
3.2.5 Global Mask Standard Register 3-9
3.2.6 Global Mask Extended Register 3-10
3.2.7 Message 15 Mask Register 3-12
3.2.8 Bus Configuration Register 3-12
3.2.9 Bit Timing Registers 3-13
3.2.10 Interrupt Register 3-17
3.2.11 Channel Reset 3-19

3.1 Channel Initialization

After reset, the channel is in "channel initialization" state. (See **Control Register,** page 3-4, bit 01 **Init** bit). While **Init** is set, all message transfers to and from the *M4KCAN* are stopped and its outputs are in a recessive state. Use the initialization state to configure the *M4KCAN* RAM without risk of CAN bus receptions or transmissions.

For each CAN channel in the module, the following user programmable registers should be initialized to the following recommended values:

1. Set Control register (0x00) CCE bit (bit 06) to 1. This enables writing to the configuration registers at addresses 0x2F, 0x3F, 0x4F.
2. Set Bus Configuration register (0x2F) to 0x4A (or 0x6A for reverse polarity).
3. Set Bit Timing registers 0, 1 (0x3F, 0x4F) to obtain required bit rate, synchronization jump width, and sampling mode.
4. Set Control register (0x00) **CCE** bit back to 0, to protect against unintentional writes to configuration registers.
5. Set User Interface register (0x02) to 0x0
6. Reset **Init** bit (Control register, bit 01) to complete initialization. The *M4KCAN* synchronizes itself to the CAN bus by waiting for 11 consecutive recessive bits (called bus idle) before it will take part in bus activities.

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

3.2 Channel Registers

This section discusses the functional operation of the *M4KCAN* channel by describing the registers used to configure the channel.

For a description of Message Objects, see **Chapter 4: M4KCAN Message Objects**.

Control Register	00 H		
Status Register	01 H		
User Interface Register	02 H		
Reserved	03 – 05 H		
Global Mask – Standard	06 – 07 H		
Global Mask – Extended	08 – 0B H		
Message 15 Mask	0C – 0F H		
Message 1	10 – 1E H		
Reserved	1F H	Reserved	8F H
Message 2	20 – 2E H	Message 9	90 – 9E H
Bus Configuration Register	2F H	Reserved	9F H
Message 3	30 – 3E H	Message 10	A0 – AE H
Bit Timing Register 0	3F H	Reserved	AF H
Message 4	40 – 4E H	Message 11	B0 – BE H
Bit Timing Register 1	4F H	Reserved	BF H
Message 5	50 – 5E H	Message 12	C0 – CE H
Interrupt Register	5F H	Reserved	CF H
Message 6	60 – 6E H	Message 13	D0 – DE H
Reserved	6F H	Reserved	DF H
Message 7	70 – 7E H	Message 14	E0 – EE H
Reserved	7F H	Reserved	EF H
Message 8	80 – 8E H	Message 15	F0 – FE H
Reserved	8F H	Channel Reset	FF H

Figure 3-1 *M4KCAN* Channel Memory Map

Note: The addresses in this memory map are for Excalibur 4000 PCI family carrier boards. For *EXC-4000VME/VXI* carrier boards, the even and odd addresses may be swapped. For more information, see **1.3 VME/VXI Byte Swapping** on page 1-5.

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

3.2.1 Channel Register Reset Values

The registers of the *M4KCAN* have the following values after a reset. On power-up the registers defined as unchanged should be interpreted as undefined.

Register		Reset Value
Control Register	00 H	01 H
Status Register	01 H	Undefined
User Interface Register	02 H	61 H
Global Mask Short	06 – 07 H	Unchanged
Global Mask Long	08 – 0B H	Unchanged
Mask Last Message	0C – 0F H	Unchanged
Bus Configuration	2F H	00 H
Bit Timing Register 0	3F H	Unchanged
Bit Timing Register 1	4F H	Unchanged
Interrupt Register	5F H	00 H
Channel Reset	FF H	Undefined
Messages 1 – 15		Unchanged

***M4KCAN* Register Values After Hardware Reset**

Note: The error management counters and the busoff state are reset by a hardware reset. Also, the outputs of the *M4KCAN* are set to a recessive state.

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

3.2.2 Control Register **Address: 00 (H)**

Bit #	Bit Name	Description
07		This bit is always 0
06	**CCE**	**Change Configuration Enable** [read/write] 1 The user has write access to configuration registers 0 The user has no write access to configuration registers Configuration register addresses are 2F H, 3F H and 4F H. This bit should be set back to 0 to protect against unintentional re-writing of critical registers following the initialization sequence.
04 – 05		These bits are always 0
03	**EIE**	**Error Interrupt Enable** [read/write] 1 Enable Error interrupts. A change in the error status of the specific channel will cause an interrupt to be generated. 0 Disable Error interrupts. No error interrupt will be generated. Error interrupts are **BOff** and **Warn** in the Status register. Error Interrupt Enable is set by the user to allow the specific channel to interrupt user when an abnormal number of CAN bus errors have been detected. It is recommended to enable this interrupt during normal operation. The Interrupt Enable bit (**IE** bit 01) must be set as well. for interrupts to actually be generated. If the **IE** bit is not set the *M4KCAN* will not generate interrupts although the Interrupt register (5F H) will still be updated.
02	**SIE**	**Status Change Interrupt Enable** [read/write] 1 Enable Status Change interrupt. An interrupt will be generated when a CAN bus error is detected in the Status register or a transfer (reception or transmission) is successfully completed, independent of the interrupt enable bits in any Message Object. 0 Disable Status Change interrupt. No status interrupt will be generated. Status change interrupts are **Wake**, **RXOK**, **TXOK**, and **LEC 0-2** in the Status register. This bit is set by the user. **RXOK** occurs upon every successful message transmission on the CAN bus, regardless of whether the message is stored by the *M4KCAN*. This interrupt is useful for hardware development to detect bus errors caused by physical layer issues such as noise. The **LEC** bits are very helpful to indicate whether bit or form errors are occurring. In normal operation it is not advised to enable this interrupt for **LEC** errors since the CAN protocol was designed to handle these error conditions in hardware by error frames and the automatic retransmission of messages. When cumulative **LEC** errors result, the warning and busoff flags will be set and the Error Interrupt should be enabled to detect these conditions. In most applications, this bit should not be set. Since this interrupt will occur for every message, the user will be unnecessarily burdened. Instead, interrupts should be implemented on a Message Object basis so interrupts occur only for messages that are used by the user. If the Status Change Interrupts and Message Object receive/transmit interrupts are enabled, there will be two interrupts for each message successfully received by a Message Object. The Interrupt Enable bit (**IE** bit 01) must be set as well. for interrupts to actually be generated. If the **IE** bit is not set the *M4KCAN* will not generate interrupts although the Interrupt register (5F H) will still be updated.

Control Register

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

Bit #	Bit Name	Description
01	**IE**	**Interrupt Enable** [read/write] 1 Enable Global interrupts. This is used in conjunction with **EIE**, **SIE** or Message Object Tx/Rx interrupts. If those are enabled, this bit must be set as well in order to actually generate interrupts. 0 Disable Global interrupts. If **EIE**, **SIE** or Message Object Tx/Rx interrupts are enabled, the *M4KCAN* will not generate interrupts although the Interrupt register (5FH) will still be updated. If the interrupt contains some value other than zero when this bit is set to one, an interrupt will be generated. For example, no interrupt will be lost because of periodic setting or resetting of this bit.
00	**Init**	**Initialization** [read/write] 1 Enable Channel initialization 0 Disable Channel initialization Setting the Init bit puts the channel into Channel Initialization state, allowing the user to configure the *M4KCAN* RAM without the chip participating in any CAN bus transmissions. While Init equals one, all message transfers to and from the CAN bus are stopped, and the status of the CAN bus outputs are recessive. The Init bit can be set by the user or the *M4KCAN* itself: 1. User initialization will most often be used the first time after power up. (See **3.1 Channel Initialization,** page 3-1) Init should not be used in normal operation when the user is modifying transmit data; the user update bit (UserUpd) bit in each Message Object is used instead. 2. M4KCAN initialization is set by the M4KCAN when it has removed itself from the CAN bus after going busoff. Busoff recovery cannot be hastened by setting or resetting the **Init** bit. If the *M4KCAN* goes busoff, the *M4KCAN* will set the Init bit by itself, thereby stopping its bus activities. Once Init is cleared by the user, the *M4KCAN* will wait for 128 occurrences of bus idle before resuming normal operation. During the initialization sequence, each time 11 recessive bits are received, a bit 00 Error code is written to the Status register, enabling the user to readily check whether or not the CAN bus is stuck in dominant state. **Note: Init** set to 1 does not break a transmission or reception of a message in process, but will stop the *M4KCAN* from transmitting or receiving the next message. Also it does not change configuration register values or error counters. Following a hardware reset, this bit will be set to 1.

Control Register (cont.)

The default value of the Control Register after a hardware reset is 01 H.

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

3.2.3 Status Register — Address: 01 (H)

Bit #	Bit Name	Description
07	BOff	**Busoff Status** [read] 1 There is an abnormal rate of occurrences of errors on the CAN bus. This condition occurs when an error counter in the *M4KCAN* has reached the limit of 256. This results in the *M4KCAN* going busoff. During busoff, no messages can be received or transmitted. The only way to exit this state is by resetting the Init bit in the Control register (location 00 H). When this bit is reset, the busoff recovery sequence begins. The busoff recovery sequence resets the transmit and receive error counters. After the *M4KCAN* counts 128 packets of 11 consecutive recessive bits on the CAN bus, the busoff state is exited. 0 The *M4KCAN* is not busoff. The busoff Status bit is written by the *M4KCAN*.
06	Warn	**Warning Status** [read] 1 There is an abnormal rate of occurrences of errors on the CAN bus. This condition occurs when an error counter in the *M4KCAN* has reached the limit of 96. 0 There is no abnormal occurrence of errors The Warning Status bit is written by the *M4KCAN*. When this bit is set, an interrupt will occur if the **EIE** and **IE** bits of the Control Register (00 H) are set.
05	Wake	**Wake up Status** [read] 1 This bit is set when the *M4KCAN* had been previously set into Sleep mode by the user, and bus activity occurs. The Sleep bit or the Power down bit in the user Interface register is reset (location 02 H) by the *M4KCAN*. 0 The Wake Up interrupt is reset by reading the Status Register. Setting the **SLEEP** bit (bit 3, register 02 H) to a '1' will place the *M4KCAN* into SLEEP mode. While in SLEEP mode, the **WAKE** bit is '0'. The WAKE bit will become '1' when bus activity is detected or when the user writes the SLEEP bit to '0'. The WAKE bit will also be set to '1' after the *M4KCAN* comes out of Power Down mode. This bit is written by the *M4KCAN*.
04	RXOK	**Receive Message Successfully** [read/write] 1 Since this bit was last reset to zero by the user, a message has been successfully received. 0 Since this bit was last reset by the user, no message has been successfully received. This bit is never reset by the *M4KCAN*. A successfully received message may be any CAN bus transmission that is error-free, regardless of whether the *M4KCAN* has configured a Message Object to receive that particular Message Identifier. This bit may be cleared by the user. The *M4KCAN* will set this bit, but will not clear it.

Status Register

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

Bit #	Bit Name	Description
03	TXOK	**Transmit Message Successfully** [read/write] 1 Since this bit was last reset to zero by the user, a message has been successfully transmitted (error free and acknowledged by at least one other node). 0 Since this bit was last reset by the user, no message has been successfully transmitted. This bit may be cleared by the user. The *M4KCAN* will set this bit, but will not clear it.
00 – 02	LEC 0 – 2	**Last Error Code** [read/write] This field contains a code which indicates the type of the first error to occur in a frame on the CAN bus. If a message is without error the field will be cleared to 0. The code 7 is unused and may be written by the user to check for updates. 0 **No error** 1 Stuff error More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 2 **Form error** The fixed format part of a received frame has the wrong format. 3 **Acknowledgement error** (Ackerror) The message transmitted by this device was not acknowledged by another node. 4 **Bit 1 error** During the transmission of a message (with the exception of the Arbitration field), the *M4KCAN* wanted to send a recessive level (bit of logical value 1), but the monitored CAN bus value was dominant. 5 **Bit 0 error** During the transmission of a message (with the exception of the Arbitration field), the *M4KCAN* wanted to send a dominant level (bit of logical value 0), but the monitored CAN bus value was recessive. During busoff recovery, this status is set each time a recessive bit is received (indicating the CAN bus is not stuck dominant). 6 **CRC error** The CRC checksum was incorrect in the message received. The CRC received for an incoming message does not match with the CRC value calculated by this device for the received data. 7 Unused

Status Register (cont.)

Status Interrupts

The status change interrupt has a value of 1. The Status Register must be read if a status change interrupt occurs.

Note: Reading the status register will clear the status change interrupt (value = 1) from the Interrupt register (5F H), if a status change interrupt is pending. A status change interrupt will occur on every successful reception or transmission, regardless of the state of the **RXOK** and **TXOK** bits. Therefore, if **TXOK** is set and a subsequent transmission occurs, an interrupt will occur (if enabled) even though **TXOK** was previously equal to one.

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

There are two ways to implement receive and transmit interrupts. The difference between these two methods is one relies on the hardwired priority of the Message Objects and the other is suitable for polling. The first and preferred method uses the **TXIE** and **RXIE** bits in the message control register for each corresponding message object. Whenever a message is transmitted or received by this Message Object, the corresponding interrupt is serviced in accordance with its priority (if the **IE** bit of register 00 H is set). This method uses the hardwired priority scheme of the *M4KCAN* which requires minimal user intervention.

The second method sets the **SIE** bit of the Control Register to "1" which will force an interrupt whenever successful message transmissions or receptions occur. The **RXOK** and **TXOK** bits will be set when any of the Message Objects transmits or receives a message. A successfully received message may be any CAN bus transmission that is error-free, regardless of whether the *M4KCAN* has configured a Message Object to receive that particular Message Identifier. This method allows the user to more easily define the interrupt priority of each Message Object by polling the Message Objects following an **SIE** interrupt.

The default value of the Status Register after a hardware reset is undefined.

3.2.4 User Interface Register — Address: 02 (H)

Bit #	Bit Name	Description
07	**RstSt**	**Hardware Reset** Status [read] 1 The hardware reset of the *M4KCAN* is active. While reset is active, no access to the *M4KCAN* is possible. 0 Normal operation. The user must ensure this bit is zero before the first access to the *M4KCAN* after reset. This bit is written by the *M4KCAN*.
0 5 – 06		User must set to 0
04	**PwD**	**Power Down Mode Enable** [read/write] 1 Channel is in Power Down mode 0 Channel is **not** in Power Down mode This bit is written by the user
03	**Sleep**	**Sleep Mode Enable** [read/write] 1 Channel is in Sleep mode 0 Channel is **not** in Sleep mode This bit is written by the user
00 – 02		User must set to 0

User Interface Register

The default value of the User Interface Register *during* hardware reset is E1 H. The default value of the User Interface Register *after* hardware reset is 61 H.

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

Low Current Modes

Power Down and Sleep Modes are activated by the **PwD** and **Sleep** bits in the user Interface register (02 H) under the control of the programmer. This register is accessible during reset and normal operation. In both modes no access to the Message Objects is possible. Access to the user Interface register (02 H) is allowed.

The *M4KCAN* exits from Power Down by either a hardware reset or by resetting the **PwD** bit to 00. The user must read the hardware reset bit, **RstSt**, (bit 07, register 02 H) to ensure the *M4KCAN* has exited Power Down.

The *M4KCAN* enters Sleep Mode after the **Sleep** bit in the User Interface register (bit 03, register 02 H) is set. The **Sleep** current is 100 mA. Sleep mode is exited by resetting the **Sleep** bit or when there is activity on the CAN bus. The *M4KCAN* requires a minimum of 10 ms to come out of Sleep Mode after bus activity occurs.

Power Down and Sleep Mode should not be entered directly after **reset**. The user program must perform a minimum RAM configuration at any time (preferably during the initialization) prior to entering these modes. Programming the following registers satisfies the minimum configuration requirement.

Control Register (00H) - set **CCE** bit to '1'

Bus Configuration Register (2F H)

Bit Timing Register 1 (4F H)

Control Register (00 H) - **Init** bit reset to '0',

Writing these registers will activate *M4KCAN* circuitry required to ensure minimum power consumption.

3.2.5 Global Mask Standard Register — Address: 06 – 07 (H)

Register	Bit #	Bit Name	Description
06	**00 – 07**	**ID21 – ID28**	**Message Identifiers 21 – 28** [read/write]
07	**05 – 07**	**ID18 – ID20**	**Message Identifiers 18 – 20** [read/write]
	00 – 04		Reserved - read as 1

Global Mask Standard Register

The Global Mask - standard register applies only to messages using the standard CAN identifier, or for Message Objects with the **Xtd** bit set to '0'. This feature, also called message acceptance filtering, 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 develop an application specific masking strategy.

A '0' value means "don't care" or accept a '0' or '1' for that bit position. A '1' value means that the incoming bit value "must-match" identically to the corresponding bit in the Message Identifier.

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

When a remote frame is sent, a *M4KCAN* receiver node will use the Global Mask registers to determine whether the remote frame matches any of its Message Objects. If the *M4KCAN* is programmed to transmit a message in response to a remote frame Message Identifier, the *M4KCAN* will transmit a message with the Message Identifier of the *M4KCAN* Message Object. The result is the remote message and the responding *M4KCAN* transmit message may have different Message Identifiers because some *M4KCAN* global mask register bits are '0'.

The default value of the Global Mask - standard after a hardware reset is unchanged.

Note: See **3.2.6.1 Acceptance Filtering Implications,** page 3-10.

3.2.6 Global Mask Extended Register — Address: 08 – 0B (H)

Register	Bit #	Bit Name	Description
08	**00 – 07**	**ID21 – ID28**	**Message Identifiers 21 – 28** [read/write]
09	**00 – 07**	**ID13 – ID20**	**Message Identifiers 13 – 20** [read/write]
0A	**00 – 07**	**ID5 – ID12**	**Message Identifiers 5 – 12** [read/write]
0B	**03 – 07**	**ID0 – ID4**	**Message Identifiers 0 – 4** [read/write]
	00 – 02		Reserved – read as 000

Global Mask Extended Register

The Global Mask - extended register applies only to messages using the extended CAN identifier, or message objects with the **Xtd** bit set to '1'. 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 the develop an application specific masking strategy. A '0' value means "don't care" on that bit for acceptance filtering. A '1' value means that the *M4KCAN* will consider this bit for acceptance filtering.

When a remote frame is sent, an *M4KCAN* receiver node will use its Global Mask registers to determine whether the remote frame matches any of its Message Objects. If the *M4KCAN* is programmed to transmit a message in response to a remote frame Message Identifier, the *M4KCAN* will transmit a message with the Message Identifier of the *M4KCAN* Message Object. The result is the remote message and the responding *M4KCAN* transmit message may have different Message Identifiers because some *M4KCAN* Global Mask register bits are '0'.

The default value of the Global Mask - extended after a hardware reset is unchanged.

Note: See **3.2.6.1 Acceptance Filtering Implications**.

3.2.6.1 Acceptance Filtering Implications

The *M4KCAN* implements two acceptance masks which allow Message Objects to receive messages with a range of Message Identifiers (IDs) instead of just a single message ID. This provides the application the flexibility to receive a wide assortment of messages from the bus.

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

The *M4KCAN* observes all messages on the CAN bus and stores any message that matches a message's ID programmed into an "active" Message Object. It is possible to define which message ID bits must identically match those programmed in the Message Objects to store the message. Therefore, ID bits of incoming messages are either "must-match" or "don't care". By selecting bits to be "don't care", Message Objects will receive multiple message IDs.

Message Objects programmed to transmit are also affected by the Global Masks (standard and extended). The *M4KCAN* uses the Global Mask registers to identify which of its Message Objects transmitted a message. If two *M4KCAN* transmit Message Objects have message IDs that are non-distinct in all "must-match" bit locations, a successful transmission of the higher numbered message object will not be recognized by the *M4KCAN*. The lower numbered Message Object will be falsely identified as the transmit Message Object and its transmit request bit will be reset and its interrupt pending bit set. The actual transmit Message Object will re-transmit without end because its transmit request bit will not be reset.

This could result in a catastrophic condition since the higher numbered Message Object may dominate the CAN bus by resending its message without end.

To avoid this condition, applications should require all transmit Message Objects to use message IDs that are unique with respect to the "must-match" bits. If this is not possible, the application should disable lower numbered Message Objects with similar message IDs until the higher numbered Message Object has transmitted successfully.

Another configuration to avoid filtering issues is to dedicate messages 1 – 14 for transmit and use message 15 for receive. The message 15 mask will have no impact on messages 1 – 14. The Acceptance Masks also apply to remote messages. When the *M4KCAN* receives a remote message, and a transmit message ID matches after taking into account the global masks, the *M4KCAN* will respond by transmitting a data message with its programmed message ID.

Example: Effect of Masking on Message Identifiers

Transit Message Object ID	1 1 0 0 0 0 0 0 0 0 0
Mask (0 = don't care; 1 = must match)	0 0 0 0 0 0 0 0 0 1 1
Received remote Message Object ID	0 0 1 1 1 1 1 1 1 0 0
Resulting Message Object ID	0 0 1 1 1 1 1 1 1 0 0

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

3.2.7 Message 15 Mask Register

Address: 0C – 0F (H)

Register	Bit #	Bit Name	Description
0C	**00 – 07**	**ID21 – ID28**	**Message Identifiers 21 – 28** [read/write]
0D	**00 – 07**	**ID13 – ID20**	**Message Identifiers 13 – 20** [read/write]
0E	**00 – 07**	**ID5 – ID12**	**Message Identifiers 5 – 12** [read/write]
0F	**03 – 07**	**ID0 – ID4**	**Message Identifiers 0 – 4** [read/write]
	00 – 02		Reserved – read as 000

Message 15 Mask Register

The Message 15 Mask register is a programmable local mask. This feature allows the user to locally mask, or "don't care", any identifier bits of the incoming message for Message Object 15. Incoming messages are first checked for an acceptance match in Message Objects 1 – 14 before passing through to Message Object 15. Consequently, the Global Mask and the local mask apply to messages received in Message Object 15.

A '0' value means "don't care" or accept a '0' or '1' for that bit position. A '1' value means that the incoming bit value "must-match" identically to the corresponding bit in the Message Identifier.

Note: The Message 15 Mask is "ANDed" with the Global Mask. This means that any bit defined as "don't care" in the Global Mask will automatically be a "don't care" bit for message 15.

The default value of the Message 15 Mask register after a hardware reset is unchanged.

3.2.8 Bus Configuration Register

Address: 2F (H)

Bit #	Bit Name	Description
07		User must set to 0
06		User must set to 1
05	**Pol**	**Polarity** [read/write] 1 A logical one is interpreted as dominant and a logical zero is recessive on the input. 0 A logical one is interpreted as recessive and a logical zero is dominant on the input, (default after hardware reset).
04		User must set to 0
03		User must set to 1
02		User must set to 0
01		User must set to 1
00		User must set to 0

Bus Configuration Register

The default value of the Bus Configuration register after a hardware reset is 00 H.

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

3.2.9 Bit Timing Registers

3.2.9.1 Bit Timing Overview

Bit timing registers are used to define the CAN bus bit rate, up to 1 Mbps (1 MHz), the sample point within a bit time, and the mode of synchronization.

A CAN message consists of a series of bits that are transmitted in consecutive bit times. A bit time accounts for propagation delay of the bit, *M4KCAN* input and output delay, and synchronization tolerances. This section describes components of a bit time from the perspective of the CAN Specification and the *M4KCAN*.

CAN Specification Bit Timing Definitions

The nominal bit time is composed of four time segments: SYNC_SEG, PROP_SEG, PHASE_SEG1, and a PHASE_SEG2. These time segments are separate and non-overlapping as shown in Figure 3-2.

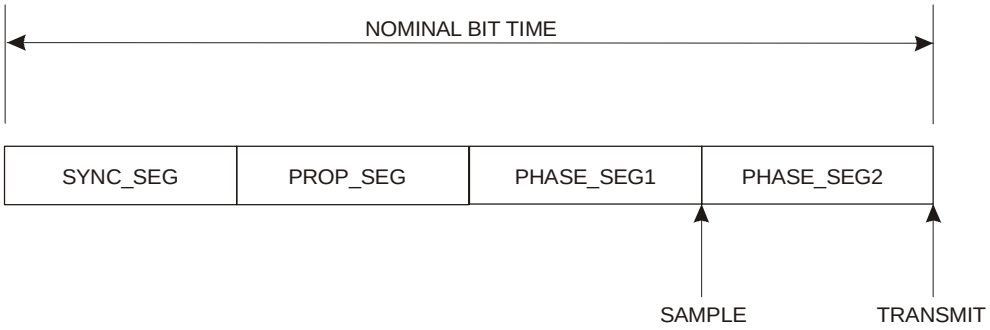


Figure 3-2 CAN Specification Bit Timing

SYNC_SEG Bit used to synchronize the various nodes on the bus. An edge is expected to lie within this segment.

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

Note: The factor of two accounts for error detection which requires nodes to monitor all bus transmissions.

PHASE_SEG1, PHASE_SEG2 These phases are used to compensate for edge phase errors. These segments can be lengthened or shortened by resynchronization.

SAMPLE POINT The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. Its location is at the end of PHASE_SEG1.

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

M4KCAN Bit Timing Definitions

The *M4KCAN* represents SYNC_SEG, PROP_SEG, PHASE_SEG1, and PHASE_SEG2 by dividing the bit time into three time segments: T_{SYNC_SEG}, T_{TSEG1} and T_{TSEG2}. The 'T' prefix indicates the segment is a length of time.

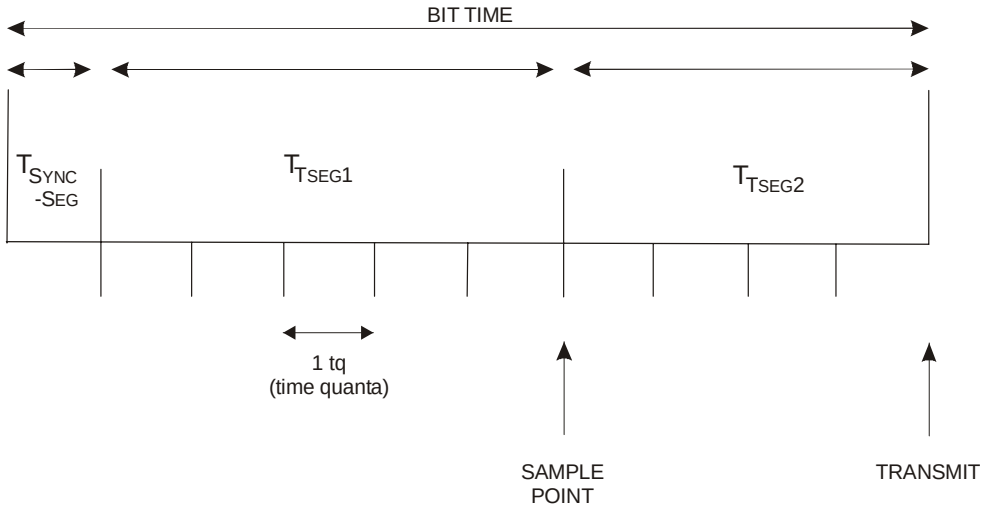


Figure 3-3 *M4KCAN* Bit Timing Definitions

Figure 3-3 represents a bit time from the perspective of the *M4KCAN*.

The length of the bit time is the sum of the segments:

T_{SYNC_SEG}, T_{TSEG1} and T_{TSEG2}

A bit time is subdivided into time quanta.

Terminology

tq — Time quantum. A fixed unit of time.T_{SYNC_SEG} Synchronizes the various nodes on the bus and an edge from the transmitter is expected to lie within this segment. T_{SYNC_SEG} is 1 time quantum.

T_{TSEG1} — The sum of PROP_SEG and the PHASE_SEG1 as defined by the CAN specification. TSEG1 is a value programmed into the *M4KCAN* CAN device to specify T_{TSEG1}

Note: In three sample mode, 2 time quanta must be added to T_{TSEG1}. This allows the *M4KCAN* to sample the bit two additional times prior to the sample at the end of T_{TSEG1}

T_{TSEG2} — This time segment is equivalent to PHASE_SEG2 as defined by the CAN specification. TSEG2 is a value programmed into the *M4KCAN* CAN device to specify T_{TSEG2}.

T_{PROP} — Two times the maximum of the sum of the delay of the physical bus delay, the output driver delay and the input comparator delay rounded up to the nearest multiple of **tq**.

T_{SJW} — The maximum number of time quanta a bit time may be shortened or lengthened by one resynchronization.

Sampling mode — The number of times the CAN bus is sampled per bit time.

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

Bit Timing Relationships

The following are relationships of the *M4KCAN* bit timing:

- bit time = $T_{SYNC_SEG} + T_{TSEG1} + T_{TSEG2}$
  (see Figure 3-3)
- tq (1 time quantum) = 8 MHz x baud rate prescale value
- $T_{SYNC_SEG} = 1$ tq

The following conditions must be met to maintain synchronization.

- $T_{TSEG} \geq 2$tq
  (minimum tolerance for resynchronization)
- $T_{TSEG2} \geq T_{SJW}$
  (If $T_{SJW} > T_{TSEG2}$, sampling may occur after the bittime)
- $T_{TSEG1} \geq 3$tq
  (minimum tolerance for resynchronization with 1tq propagation delay allowance)
- When one sample is used: $T_{TSEG1} \geq T_{SJW} + T_{PROP}$
- When three sample mode $T_{TSEG1} \geq T_{SJW} + T_{PROP} + 2$tq

3.2.9.2 Bit Timing Register 0 Address: 3F (H)

Bit #	Bit Name	Description
0 6 – 07	**SJW**	**(Re-)Synchronization Jump Width** [read/write] The valid programmed values are 0 – 3. The SJW defines the maximum number of time quanta a bit time may be shortened or lengthened by one resynchronization. The actual interpretation of this value by the hardware is to use one more than the programmed value.
00 – 05	**BRP**	**Baud Rate Prescaler** [read/write] The valid programmed values are 0 – 63. The baud rate prescaler programs the length of one time quantum as follows: tq = 8 MHz × (BRP + 1). The actual interpretation of this value by the hardware is to use one more than the programmed value.

Bit Timing Register 0

The default value of the Bit Time register 0 after a hardware reset is unchanged.

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

3.2.9.3 Bit Timing Register 1 Address: 4F (H)

Bit #	Bit Name	Description
07	SPL	**Sampling Mode** [read/write] 1 Three samples are used for determining the valid bit value using majority logic. The CAN bus is sampled three times per bit time. 0 One sample is used for determining the valid bit value. The CAN bus is sampled once per bit time. Sampling mode '0' may result in faster bit transmissions rates, while sampling mode '1' is more immune to noise spikes on the CAN bus.
04 – 06	TSEG2	**Time Segment 2** [read/write] Legal values are 1-7. TSEG2 is the time segment after the sample point. The actual interpretation of this value by the hardware is one more than the value programmed by the user.
00 – 03	TSEG1	**Time Segment 1** [read/write] Legal values are 2-15. TSEG1 is the time segment before the sample point. The actual interpretation of this value by the hardware is one more than the value programmed by the user. **Note:** In order to achieve correct operation according to the CAN protocol, the total bit length should be a minimum of 8tq with (TSEG1 + TSEG2 ≥ 5).

Bit Timing Register 1

The default value of the Bit Time register 1 after a hardware reset is unchanged.

Programming *M4KCAN* With Bit Time Values

T_{TSEG1} = (TSEG1 + 1) x **tq**, where TSEG1 can range from 2 – 15

T_{TSEG2} = (TSEG2 + 1) x **tq,** where TSEG2 can range from 1 – 7

T_{SJW} = (SJW + 1) x **tq** where **SJW** can range from 0 – 3

Baud rate prescale value = BRP +1, where **BRP** can range from 0 – 63

Sampling mode: if one sample is used **SPL** = 0; if three samples are used **SPL** = 1

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

Bit Timings Calculations Example

$$\text{CAN bus frequency} = \frac{8\ \text{MHz}}{(\text{baud rate prescale value}) \times (T_{sync_seg} + T_{Tseg1} + T_{Tseg2})} = \frac{8\ \text{MHz}}{(BRP+1) \times (1 + (TSEG+1) + (TSEG2+2))}$$

Example: Bit Time register 0 (Reg 3F H) = 4A H

With resulting bit timing parameters:
SJW = 1
BRP = 10

Bit Time register 1 (Reg 4F H) = 25 H

With resulting bit timing parameters:
SPL = 0
TSEG1 = 5
TSEG2 = 2

$$\text{CAN bus frequency} = \frac{8\ \text{MHz}}{[10+1] \times [1+(5+1)+(2+1)]} = 72{,}727\ \text{bits/second}$$

The *M4KCAN* timings calculations and their associated timing equation implicitly account for synchronization jump widths.

3.2.10 Interrupt Register

Address: 5F (H)

Bit #	Bit Name	Description
00 – 07	Intid	**Interrupt Identifier** [read only]

Interrupt Register

The Interrupt register is a read-only register. The value in this register indicates the source of the interrupt. When no interrupt is pending, this register has the value '0'.

If the **SIE** bit in the Control register (00 H) is set and the *M4KCAN* has updated the Status register, the Interrupt register value will be '1'. This indicates an interrupt is pending due to a change in the Status register.

If the **IntPnd** bit in Message Object 15 is set, the value of the Interrupt register is '2'. (Message Object 15 has the highest priority of all Message Objects.)

If the **IntPnd** bit in one of the Message Objects 0 – 14 is set, the value of the Interrupt register is "Message object +2". (For Message Object 0, the value of the Interrupt register is 2; for Message Object 1, the value is 3, etc.)

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

Interrupt	Register Value [hex]
none	0
Status Register	1
Message Object 15	2
Message Object 1	3
Message Object 2	4
Message Object 3	5
Message Object 4	6
Message Object 5	7
Message Object 6	8
Message Object 7	9
Message Object 8	A
Message Object 9	B
Message Object 10	C
Message Object 11	D
Message Object 12	E
Message Object 13	F
Message Object 14	10

Interrupt Register Value

For example, a message is received by Message Object 13 with the **IE** (Channel Control register) and **RXIE** (in the Control 0 register of Message Object 13,) bits set. The interrupt pin will be pulled low and the value 15 (0F H) will be placed in the interrupt register.

The status change interrupt has higher priority that interrupts from Message Objects. If the value of the Interrupt register is '1', indicating Status Change Interrupt, the Status register (at location 01 H) must be read to update the Interrupt register. Then the Interrupt Register is automatically set to '0' or to the lowest value corresponding to a message with **IntPnd** set.

The *M4KCAN* will respond to each status change event independently and will not bundle interrupt events in a single interrupt signal. However, if two status change events occur before the first is acknowledged by the user, the next event will not generate a separate interrupt output. Therefore, when servicing status change interrupts, the user code should check all useful status bits upon each status change interrupt.

After resetting the **IntPnd** bit in the Control 0 Register of individual Message Objects, the minimum delay of the *M4KCAN* resetting the interrupt pin and updating the Interrupt register (5F H) is 8.93 nS and a maximum of 41.67 nS (after the user write operation to this register is finished). When a status change interrupt occurs, reading the Status register (01 H) will reset the interrupt pin in a maximum of 176.25 nS. Clearing the **IntPnd** bit of the message object will deactivate the INT# pin.

The default value of the Interrupt register after a hardware reset is unchanged

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

3.2.11 Channel Reset Address: FF (H)

Channel Reset is used to synchronize accesses of individual channels on the *M4KCAN*. It will synchronize between the *M4KCAN* channel and the user when the user does not want to execute a hardware reset, which will reset all CAN channels on the module.The user must write a string of 16 'FF H' bytes to achieve synchronization.

Bit #	Description
00 – 07	**Channel Reset** write only

Channel Reset

The default value of Channel Reset after a hardware reset is undefined.

Note: A hardware reset, which will reset all CAN channels on the module, is performed by writing to the carrier board's Global registers.

See your carrier board's *User's Manual*.

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

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

4 *M4KCAN* Message Objects

4.1 Message Object Structure 4-1
4.2 Message Object Registers 4-2
4.2.1 Control 0 and Control 1 Registers 4-2
4.2.2 Arbitration 0, 1, 2, 3 Registers [read/write] 4-6
4.2.3 Message Configuration Register 4-7
4.2.4 Data Bytes 4-7
4.3 Special Treatment of Message Object 15 4-8

4.1 Message Object Structure

The Message Object is the means of communication between the user and the *M4KCAN*. Message Objects are configured to transmit or receive messages.

There are 15 Message Objects located at fixed addresses in the *M4KCAN*. Each Message Object starts at a base address that is a multiple of 16 bytes and uses 15 consecutive bytes. For example, Message Object 1 starts at address 10 H and ends at address 1E H. The remaining byte in the 16 byte field is used for other *M4KCAN* functions.

Message Object 15 is a receive-only Message Object that uses a local mask called the Message 15 Mask register. This mask allows a large number of infrequent messages to be received by the *M4KCAN*. In addition, Message Object 15 is buffered to allow the user more time to receive messages.

Base Address	+0	Control 0
	+1	Control 1
	+2	Arbitration 0
	+3	Arbitration 1
	+4	Arbitration 2
	+5	Arbitration 3
	+6	Message Configuration
	+7	Data 0
	+8	Data 1
	+9	Data 2
	+10	Data 3
	+11	Data 4
	+12	Data 5
	+13	Data 6
	+14	Data 7

Message Object Structure

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

4.2 Message Object Registers

4.2.1 Control 0 and Control 1 Registers

Base Address + 0
Base Address + 1

Writing and Reading Control Registers

The Message Object Control registers are configured with two bits per function to make testing and setting these bits as efficient as possible. This method allows write access to single bits of the byte, leaving the other bits unchanged, without need to perform a read/write/modify cycle.

Writing to Control Registers

The allowed operations are:

- Write 0x2 to the bit pair, effectively setting it to a value of 1. This is referred to as **Set**.
- Write 0x1 to the bit pair, effectively setting it to a value of 0. This is referred to as **Reset**.
- Write 0x3 to the bit pair, leaving the value as is. This is referred to as **Unchanged**

Writing 0x0 is not allowed; the resulting value is indeterminate

Reading from Control Registers.

- When the bit pair is read as 0x2, its value is effectively 1. This referred to as **Set**.
- When the bit pair is read as 0x1, its value is effectively 0. This referred to as **Reset**.

	MSB	LSB	Meaning
Write	0	1	Change value to 0: **Reset**
	1	0	Change value to 1: **Set**
	1	1	Value remains as is: **Unchanged**
Read	0	1	Value is 0: **Reset**
	1	0	Value is 1: **Set**

Control Register Bit Representation

Example: To set the **TxRqst** bit pair (bits 4 and 5) of the Control 1 register to an effective value of 1:

Write 0xEF H [binary: 11 10 11 11] to the register

This will set the TxRqst value to 1 and leave the other values as they are.

Each bit in the Control 0 and Control 1 registers may be set and reset by the user as required.

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

4.2.1.1 Control Register 0 Base Address + 0

Bit #	Bit Name	Description
06 – 07	MsgVal	**Message Valid** [read/write] The **MsgVal** bit is an individual halt bit for each Message Object. While this bit is reset the *M4KCAN* will not access this Message Object for any reason. 1 The Message Object is valid 0 The Message Object is invalid The Message Valid bit is set to indicate the Message Object is configured and is ready for communications transactions. This bit may be reset at any time if the message is no longer required, or if the identifier is being changed. If a Message Identifier is changed, the message object must be made invalid first, and it is not necessary to reset the chip following this modification. The user must reset the **MsgVal** bit of all unused messages during initialization of the *M4KCAN* before the **Init** bit of the Control Register (00 H) is reset. The contents of Message Objects may be reconfigured dynamically during operation and the **MsgVal** bit assists reconfiguration in many cases. This bit is written by the user. Two or more Message Objects may not have the same Message Identifier and also be valid at the same time.
04 – 05	TXIE	**Transmit Interrupt Enable** [read/write] 1 An interrupt will be generated after a successful transmission of a frame 0 No interrupt will be generated after a successful transmission of a frame The Transmit Interrupt Enable bit enables the *M4KCAN* to initiate an interrupt after the successful transmission by the corresponding Message Object. This bit is written by the user. **Note:** In order for **TXIE** to generate an interrupt, IE in the Control register must be set.
02 – 03	RXIE	**Receive Interrupt Enable** [read/write] 1 An interrupt will be generated after a successful reception of a frame 0 No interrupt will be generated after a successful reception of a frame This bit enables the *M4KCAN* to initiate an interrupt after the successful reception by the corresponding Message Object. This bit is written by the user. **Note:** In order for **RXIE** to generate an interrupt, **IE** in the Control register must be set.
00 – 01	IntPnd	**Interrupt Pending** [read/write] 1 This Message Object has generated an interrupt. 0 No interrupt was generated by this Message Object since the last time the user cleared this bit. This bit is set by the *M4KCAN* following a successful transmission or reception as controlled by the **RXIE** and **TXIE** bits. The user **must** clear this bit when servicing the interrupt.

The default value of Control 0 register after a hardware reset is unchanged.

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

4.2.1.2 Control Register 1 Base Address: + 1

Bit #	Bit Name	Description
06 – 07	RmtPnd	**Remote Frame Pending** [read/write] 1 The transmission of this Message Object has been requested by a remote node and is not yet done 0 There is no waiting remote request for the Message Object This bit is only used by Message Objects with direction = transmit. This bit is set by the *M4KCAN* after receiving a remote frame which matches its Message Identifier, taking into account the global mask register. The corresponding Message Object will respond by transmitting a message, if the **UserUpd** bit = zero. Following this transmission, the *M4KCAN* will clear the **RmtPnd** bit. In other words, when this bit is set it indicates a remote node has requested data and this request is still pending because the data has not yet been transmitted. **Note:** Setting **RmtPnd** will not cause a remote frame to be transmitted. The **TxRqst** bit is used to send a remote frame from a receive Message Object.
04 – 05	TxRqst	**Transmit Request** [read/write] 1 The transmission of this Message Object has been requested and has not been completed 0 This Message Object is not waiting to be transmitted This bit is set by the user to indicate the Message Object data should be transmitted. Conditions required to transmit a Data frame: **Init** bit = 0 **MsgVal** bit = 1 **Direction** = transmit **NewDat** bit = 1 **TxRqst** bit = 1 If direction = receive, (**Init** = 0 and **MsgVal** = 1) then a remote frame is sent to request a remote node to send the corresponding data. **TxRqst** is also set by the *M4KCAN* (at the same time as **RmtPnd** in Message Objects whose direction = transmit) when it receives a remote frame from another node requesting this data. This bit is cleared by the *M4KCAN* along with **RmtPnd** when the message has been successfully transmitted, if the **NewDat** bit has not been set. **Note:** Setting **TxRqst** will send a data frame for a transmit Message Object and a remote frame for a receive Message Object.
02 – 03	MsgLst	**Message Lost** [read/write] Message Object direction: ***Receive*** 1 The *M4KCAN* has stored a new message in this Message Object when **NewDat** was still set. 0 No message was lost since the last time this bit was reset by the user. This bit is used to signal that the *M4KCAN* stored a new message into this Message Object when the **NewDat** bit was still set. Therefore, this bit is set if the user did not process the contents of this Message Object since the last time the *M4KCAN* set the **NewDat** bit; this indicates the last message received by this Message Object overwrote the previous message which was not read and is lost.

Control Register 1

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

Bit #	Bit Name	Description
02 – 03	**UserUpd**	**User Updating** [read/write]
Message Object direction: ***Transmit***		1 This Message Object may not be transmitted 0 This Message Object may be transmitted, if direction = transmit The user sets this bit to indicate it is updating the data contents of the Message Object and the message should not be transmitted until this bit has been reset. The user indicates message updating has been completed by resetting this bit (it is not necessary to use the **MsgVal** bit to update the Message Object's data contents). The purpose of this bit is to prevent a remote frame from triggering a transmission of invalid data.
00 – 01	**New Dat**	**New Data** [read/write] This bit has different meanings for receive and transmit Message Objects. 1 The *M4KCAN* or user has written new data into the data section of this Message Object. 0 No new data has been written into the data section of this Message Object since the last time this bit was cleared by the user. For Message Objects with direction = receive, the *M4KCAN* sets this bit whenever new data has been written into the Message Object. **Note:** For Message Objects 1-14, when new data is written into Message Objects, the unused data bytes will be overwritten with non-specified values. The user should clear this bit before reading the received data and then check if the bit remained cleared when all bytes have been read. If the **NewDat** bit is set, the user should re-read the received data to prevent working with a combination of old and new data. For Message Object 15, new data is written into the shadow register. The foreground register is not over-written with new data. For Message Object 15 messages, the data should be read first, the **IntPnd** reset, and then the **NewDat** and **RmtPnd** bits are reset. Resetting the **NewDat** and **RmtPnd** bits before resetting the **IntPnd** bit will result in the interrupt line remaining active. For Message Objects with direction = transmit, the user must set this bit to indicate it has updated the message contents. This is done at the same time the user clears the **UserUpd** bit. This will ensure that if the message is actually being transmitted during the time the message was being updated by the user, the *M4KCAN* will not reset the **TxRqst** bit. In this way, the **TxRqst** bit is reset only after the actual data has been transferred.

Control Register 1 (cont.)

The default value of Control 1 register after a hardware reset is unchanged.

To program a transfer request, the Control 1 register of the Message Object should have the **TxRqst** and **NewDat** bits set to '1'. Therefore, this register may be written with the value 066 H to initiate a transmission.

A Message Object can be configured so that a remote frame will be received, an interrupt flag set, and no transmit sent in response. To do so:

- Set the **UserUpd** and **RXIE** bits in the Message Object Control register to '1'.
- Set the **Dir** bit in the Message Configuration register to Transmit ('1').

A remote frame will be received by this Message Object, the **IntPnd** bit will be set to '1' since it received a request to transmit. No transmit message will be sent. The data can now be updated and transmitted.

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

Message Object Priority

If multiple Message Objects are waiting to transmit, the *M4KCAN* will first transmit the message from the lowest numbered Message Object, regardless of Message Identifier Priority.

If two Message Objects are capable of receiving the same message (possibly due to message filtering strategies), the message will be received by the lowest numbered Message Object. For example, if all acceptance mask bits were set as 'don't care', Message Object 1 will receive all messages.

4.2.2 Arbitration 0, 1, 2, 3 Registers [read/write] Base Address + 2 – Base Address + 5

When the *M4KCAN* receives a message, the entire Message Identifier, the data length code (**DLC**) and the Direction bit are stored into the corresponding message object.

Arbitration 0 Register Base Address + 2	
Bit	**Bit Name**
07	ID28
06	ID27
05	ID26
04	ID25
03	ID24
02	ID23
01	ID22
00	ID21

Arbitration 1 Register Base Address + 3	
Bit	**Bit Name**
07	ID20
06	ID19
05	ID18
04	ID17
03	ID16
02	ID15
01	ID14
00	ID13

ID0 – ID28 Message Identifier

ID0 – ID28
is the identifier for an extended frame

ID18 – ID28
is the identifier for a standard frame

Arbitration 2 Register Base Address + 4	
Bit	**Bit Name**
07	ID12
06	ID11
05	ID10
04	ID9
03	ID8
02	ID7
01	ID6
00	ID5

Arbitration 3 Register Base Address + 5	
Bit	**Bit Name**
07	ID4
06	ID3
05	ID2
04	ID1
03	ID0
02	Reserved
01	Reserved
00	Reserved

Note: When the *M4KCAN* receives a message, the entire Message Identifier, the data length code [**DLC**] and the Direction bit are stored in the corresponding Message Object.

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

4.2.3 Message Configuration Register — Base Address + 6

Bit #	Bit Name	Description
04 – 07	**DLC**	**Data Length Code** [read/write] The valid programmed values are 0 – 8 bytes. The data length code of a Message Object is written with the value corresponding to the data length.
03	**Dir**	**Direction** [read/write] 1 Transmit: When **TXRqst** is set, the Message Object will be transmitted. 0 Receive: When **TXRqst** is set, a Remote frame will be transmitted. When a message is received with a matching identifier, the message will be stored in the Message Object.
02	**Xtd**	**Extended or Standard Identifier** [read/write] 1 This Message Object will use an Extended 29-bit Message Identifier. 0 This Message Object will use a Standard 11-bit Message Identifier. If an extended frame Message Identifier is used (arbitration bits 0 – 17) and the message configuration register **Xtd** bit is '0' to specify a standard frame, the *M4KCAN* will reset the extended bits in the arbitration registers to '0'. An extended receive Message Object (**Xtd** ='1') will not receive standard messages. If a Message Object receives a message from the bus, the entire Message Identifier will be stored in the Message Object. Therefore, if acceptance filtering (masking register) is used, the masked-off "don't care" bits will be rewritten corresponding to the message ID of the incoming message.
00 – 01		Reserved - user must set to 0

Message Configuration Register

The default value of the Message Configuration Register after a hardware reset is unchanged.

4.2.4 Data Bytes — Base Address + 7 – Base Address + 14

When the *M4KCAN* stores a message all 8 data bytes will be written into the Message Object.

The default value of the data bytes 0 – 7 after a hardware reset is unchanged. The values of unused data bytes are random and change during operation.

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

4.3 Special Treatment of Message Object 15

Message Object 15 is a receive-only Message Object with a programmable local mask called the Message 15 Mask register. Since this Message Object is a receive only Message Object, the **TXRqst** bit and the **TXIE** have been hardwired inactive and the **UserUpd** bit has no meaning.

The incoming messages for Message Object 15 will be written into a two-message alternating buffers to avoid the loss of a message if a second message is received before the user has read the first message. Once Message Object 15 is read, it is necessary to reset the **NewDat** and the **RmtPnd** bits to allow the user to read the shadow message buffer which will receive the next message or which may already contain a new message.

If two messages have been received by Message Object 15, the first will be accessible to the user. The alternate buffer will be overwritten if a subsequent (third) receive message is received. Once again, after reading message 15, the user program should reset the **IntPnd** bit followed by a reset of the **NewDat** and **RmtPnd** bits in the Control 1 register.

The **Xtd** bit in the Message Configuration register determines whether a standard or an extended frame will be received by this Message Object.

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

5 Mechanical And Electrical Specifications

5.1 *M4KCAN* Module Layout 5-1
5.2 LED Indicators 5-2
5.3 Module Bus Termination Select DIP Switches 5-2
5.3.1 Factory Default DIP Switch Settings 5-3
5.4 Connectors 5-3
5.4.1 Terminal Stick Connector 5-3
5.4.2 Connector Pin Assignments: CAN Configuration 5-4
5.4.3 *M4KCAN* Module Adapter Cable 5-5
5.5 Power Supply Requirements 5-6

5.1 *M4KCAN* Module Layout

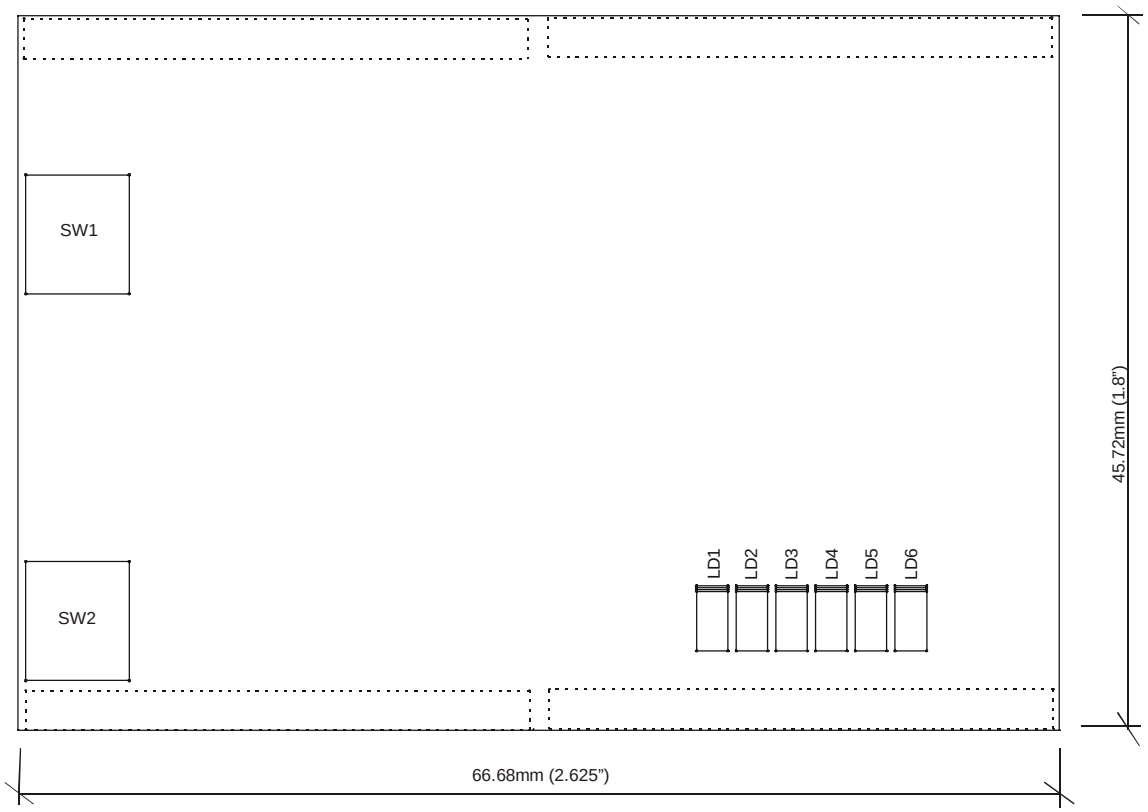


Figure 5-1 *M4KCAN* Module Layout

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

5.2 LED Indicators

The *M4KCAN* contains six LEDs. The LEDs indicate operational mode and bus activity.

LED	Indications
LD1	Channel 0 bus activity
LD2	Channel 1 bus activity
LD3	Channel 2 bus activity
LD4	Channel 3 bus activity
LD5	Channel 4 bus activity
LD6	Channel 5 bus activity

LED Indicators

5.3 Module Bus Termination Select DIP Switches

Each channel and associated bus of the *M4KCAN* can be terminated with a 120 Ohm resistor via a DIP switches SW1 and SW2.

Table 5-1 defines the DIP switch settings:

Termination Mode	Switch Position
Bus terminated with 120 Ohms	ON
No bus termination	OFF

Table 5-1 DIP Switch Settings Required to Select Termination Mode

Table 5-2 defines the DIP switch for each Bus:

Bus	DIP Switch	Switch #
Channel 0	SW2	4
Channel 1	SW2	3
Channel 2	SW2	2
Channel 3	SW2	1
Channel 4	SW1	4
Channel 5	SW1	3

Table 5-2 Bus DIP Switch

Example: To terminate the Channel 0 bus, switch on switch #4 of SW2.

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

5.3.1 Factory Default DIP Switch Settings

The factory default settings are:

DIP Switch	Switch Position	Coupling Mode
SW2 – 4 [Channel 0 Bus]	OFF	Not Terminated
SW2 – 3 [Channel 1 Bus]	OFF	Not Terminated
SW2 – 2 [Channel 2 Bus]	OFF	Not Terminated
SW2 – 1 [Channel 3 Bus]	OFF	Not Terminated
SW1 – 4 [Channel 4 Bus]	OFF	Not Terminated
SW1 –3 [Channel 5 Bus]	OFF	Not Terminated

Table 5-3 Factory Default DIP Switch Settings

5.4 Connectors

Description	Part Number (P/N)
All the I/O communication signals of the *M4KCAN* module arrive at the **Molex® LFH Female 96-pin connector** located on the EXC-4000 carrier board. Each of the modules on the carrier board is allocated a separate row on the I/O connector	51-26-0000
Molex® 24-pin Terminal Stick	51-25-1012
Four Terminal Sticks inserted into **Plug Connector Sub-assembly**, make up the mating connector for the connector board I/O connector	51-26-0012

Connectors

The *M4KCAN* module is delivered with a set of all the required connectors. Additional connectors are available directly from Molex®.

5.4.1 Terminal Stick Connector

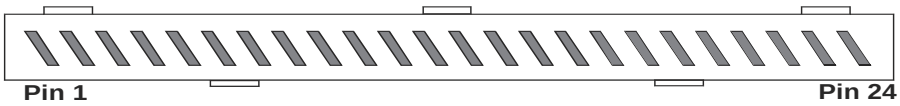


Figure 5-2 Terminal Stick Connector: Front View

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

5.4.2 Connector Pin Assignments: CAN Configuration

Carrier Board Terminal Stick Pin #	Signal Name	Description
1	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
2	CAN0H	Channel 0 connection Hi
3	CAN0L	Channel 0 connection Lo
4	CAN1H	Channel 1 connection Hi
5	CAN1L	Channel 1 connection Lo
6	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
7	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
8	CAN2H	Channel 2 connection Hi
9	CAN2L	Channel 2 connection Lo
10	CAN3H	Channel 3 connection Hi
11	CAN3L	Channel 3 connection Lo
12	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
13	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
14	CAN4H	Channel 4 connection Hi
15	CAN4L	Channel 4 connection Lo
16	CAN5H	Channel 5 connection Hi
17	CAN5L	Channel 5 connection Lo
18	SHIELD	Provided for CAN cables shield connection. This signal is connected to the case of the computer
19 - 20	Reserved	
21	GND	Provides ground reference
22 - 24	Reserved	

Table 5-4 CAN Connector Pin Assignment Configuration

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

5.4.3 *M4KCAN* Module Adapter Cable

Excalibur's standard *M4KCAN* adapter cable converts the Molex® terminal stick to a standard 26-pin high density D-type male connector with jack screws. The cable is 0.5 meter in length and a mating connector with hoods is provided. Table 5-4 provides *M4KCAN* Module Adapter Cable pinouts (the HDB 26-pin connector pinouts, and the associated module terminal stick pinouts).

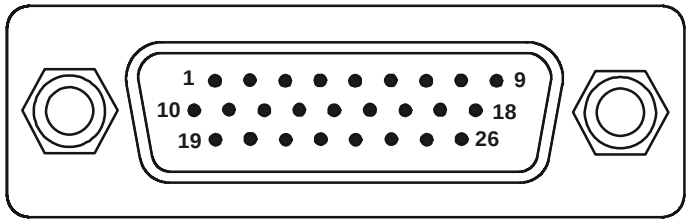


Figure 5-3 Adapter Cable 26-pin Connector: Front View

High density D-type male 26-pin connector pin #	Signal Name	24-pin Terminal Stick pin #
1	SHIELD	1
2	CH0H	2
3	CH0L	3
4	CH1H	4
5	CH1L	5
6	SHIELD	6
7	SHIELD	7
8	CH2H	8
9	CH2L	9
10	CH3H	10
11	CH3L	11
12	SHIELD	12
13	SHIELD	13
14	N/C	N/C
15	CH4H	14
16	CH4L	15
17	CH5H	16
18	CH5L	17
19	SHIELD	18
20	Reserved	19
21	Reserved	20
22	GND	21
23 - 26	N/C	22 - 24

Table 5-5 *M4KCAN* Adapter Cable Connections

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

5.5 Power Supply Requirements

The *M4KCAN* power requirements are:

- At standby: +5V @ 200mA
- Maximum additional power required per channel: +5V @ 70mA

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

6 Ordering Information

Chapter 6 explains how to indicate the options you want when ordering an *M4KCAN* module.

Add **-E** to the name of the module to indicate extended temperature/ruggedized options.

Part Number	Description
M4KCAN2	2 independent channels of CAN 2.0B protocol with standard and extended message frames and Message Identifiers
M4KCAN4	Same as above with 4 independent channels
M4KCAN6	Same as above with 6 independent channels
M4KCAN*[2/4/6]*–E	As above with extended temperature operation and ruggedized (-40°C to + 85°C)

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

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

Appendix A Transmission Characteristics

Appendix A: Transmission Characteristics includes:

A.1 Frame Types A-1
A.1.1 Data Frame A-1
A.1.2 Remote Frame A-3
A.1.3 Error Frame A-3
A.1.4 Overload Frame A-4
A.1.4.1 Interframe-space A-5
A.1.4.2 Coding Decoding A-5
A.1.4.3 Arbitration A-6
A.2 Error Detection and Confinement A-7
A.2.1 Bit Error A-7
A.2.2 Bit Stuffing Error A-8
A.2.3 CRC Error A-8
A.2.4 Form Error A-8
A.2.5 Error Detection Capabilities A-9
A.2.6 Error Confinement A-9
A.2.7 *M4KCAN* States with Respect to the Serial Bus A-10

A.1 Frame Types

The *M4KCAN* communications controller supports four frame types:

Data frame
Remote frame
Error frame
Overload frame

A.1.1 Data Frame

A Data frame is composed of seven fields:

Start bit
Arbitration field
Control (Identifier) field
Data field
CRC field
Acknowledge field
End of Frame

The minimum message lengths of standard and extended message formats are summarized for Data and Remote frames. The actual lengths of these messages may differ because ‘stuff’ bits are added to the message. Stuff bits assist synchronization by adding transitions to the message. A stuff bit is inserted in the bit stream after five consecutive-equal value bits are transmitted; the stuff bit is the opposite polarity of the five consecutive bits. All message fields are stuffed except the **CRC** delimiter, the **ACK** field and the **End of Frame**.

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

Table A-1 describes the Standard and Extended message formats for Data and Remote frames.

Message Field	Number of Bits: Standard Format	Number of Bits: Extended Format	Description
SOF	1	1	**Start of Frame bit** (dominant bit) Marks the beginning of a Data/Remote frame
Arbitration (ID)	11	11 18	**Arbitration** One or two fields which contain the Message Identifier bits. The standard format has one 11-bit field and the extended format has two fields, 11- and 18-bits wide.
RTR	1	1	**Remote Transmission Request bit** Bit is dominant for Data frames and recessive for Remote frames. This bit is in the arbitration field.
SRR	n/a	1	**Substitute Remote Request bit** Used in extended messages and is recessive. This bit is a substitute for the **RTR** bit in the standard format. This bit is in the arbitration field of the extended format.
IDE	1	1	**Identifier Extension bit** Dominant for standard format and recessive for extended format. This bit is in the arbitration field of the extended format and in the control field of the standard format.
Control Field **r0** **r1** **DLC**	1 N/A 4	1 1 4	**r0 and r1 reserved bits** Reserved bits r0 and r1 are sent as dominant bits. **DLC** The 4-bit Data Length Code [DLC] indicates the number of bytes in the data field.
Data Field	0 – 64	0 – 64	**Data Field** The data bytes are located in the Data frame (0 – 8 bytes). A remote frame contains 0 data bytes.
CRC Field	16	16	**Cyclical Redundancy Code** This field is composed of a 15-bit Cyclical Redundancy Code error code and a recessive CRC delimiter bit.
ACK Field	2	2	**Acknowledge** A dominant bit sent by nodes receiving the data/remote frame and is followed by a recessive ACK delimiter bit.
End of Frame	7	7	Seven recessive bits ending the frame.
	44 – 108	**64 – 128**	**Total bits in Message Field**
INT	3	3	Intermission is the three recessive bits which separate data and remote frames.

Table A-1 CAN Standard and Extended Message Formats

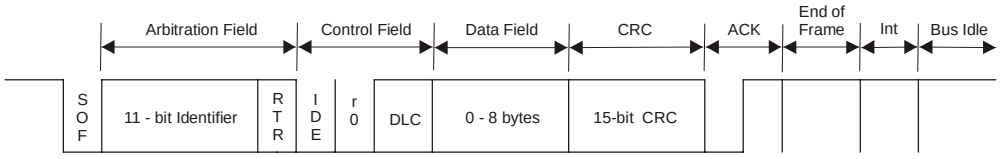


Figure A-1 Standard CAN Message Format

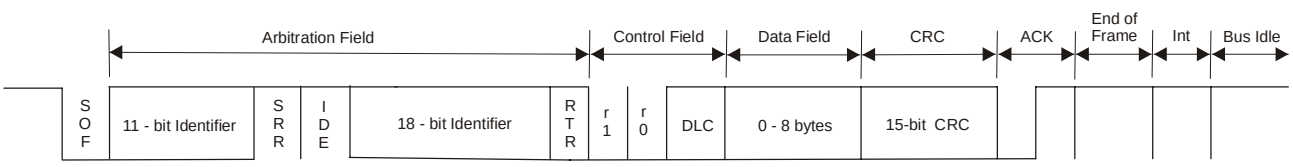


Figure A-2 Extended CAN Message Format

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

A.1.2 Remote Frame

A Remote frame is composed of seven fields:

Start bit
Arbitration field
Control (Identifier) field
Data field
CRC field
Acknowledge field
End of Frame

Contrary to the Data Frame, the **RTR**-bit of the Remote Frame is "recessive" and no data segment is transmitted independent of the **Data Length Code** set by the Descriptor of the corresponding Communication Object.

The **RTR**-bit allows Remote Transmission Requests from any node to the system. This provides the capability to request information in addition to the standard broadcast characteristics. It also supports powerful diagnostic capability by being able to determine if the primary transmitter (data source) of a specific parameter(s) is on the bus and functional.

A.1.3 Error Frame

The Error Frame contains a sequence of variable length dominant bits as a result of error flags being transmitted by different system-nodes. This is an important aspect of the *M4KCAN* communication protocol with regards to data consistency within a communication network. The Error Frame is followed by an Error Delimiter.

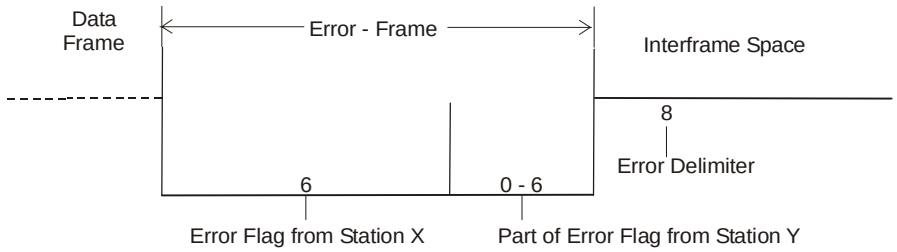


Figure A-3 Error Frame Format

Error flag consists of six consecutive dominant bits. Since this "violates" bit stuffing rules, it is used as an error indicator to the system. (See **Coding Decoding** on page A-5.)

An Error flag is transmitted if an *M4KCAN* operates as an error active node and has detected an error condition during or after a message transfer. If an Error flag is generated by a transmitter, or a receiver, all other nodes interpret the Error flag as a bit stuffing rule violation. As a consequence, they, in turn, transmit an Error flag. A variable sequence of dominant bits result from the superposition of the different Error flags transmitted by individual nodes. The total length of the Error flag sequence varies between six bits minimum to twelve bits maximum.

An error condition is signaled by the transmission of six recessive bits while in the error passive operation mode. This way an error passive node with a temporary local receiver problem will not destroy messages received correctly by other nodes. The recessive bits may be overwritten by an Error Flag generated by one or more error active system nodes, but the error passive *M4KCAN* waits for at least six bits

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

of equal polarity before entering into the next internal receive or transmit mode. (See **M4KCAN States with Respect to the Serial Bus** on page A-10.)

Note: The *M4KCAN* will not perform storage of a message (positive acceptance filtering) into the communication buffer, if reception of the message was followed by an Error flag on the serial bus.

The Error Delimiter consists of eight recessive bits generated by the *M4KCAN* after the end of an Error flag on the serial bus line. This is monitored by detection of a transition from the dominant to recessive bit level.

Detected errors during the transmission of a Data or Remote frame can be signaled within the transmission time of the respective frame. This procedure associates an Error flag to the corresponding frame, and initiates a retransmission of the frame. As the *M4KCAN* monitors, any deviation of its Error frame will start retransmitting an Error frame. If this occurs several times in a sequence the *M4KCAN* will become error passive.

A.1.4 Overload Frame

The Overload frame consists of two bit fields, the Overload flag and the Overload delimiter.

There are two cases of overload conditions which result in the transmission of an Overload flag:

a. Internal conditions of the receiver circuitry of a *M4KCAN* which requires a delay time before receiving the next frame (receiver not ready). The *M4KCAN* will not generate Overload frames when CAN bus transmission rates are 1 Mbit/sec or less.
b. Detection of a “dominant” bit during Interframe Space.

The Overload frame consists of six dominant bits that correspond to the Error flag and destroy the fixed form of the Interframe Space Field. As a consequence, all other nodes see the dominant bit during the Interframe Space time and interpret the recessive to dominant edge as a start of frame and transmit an Overload flag because of the overload condition.

The Overload delimiter consists of seven recessive bits generated by the *M4KCAN*.

After transmission of an Overload frame, each *M4KCAN* within the system monitors the bus line until a transition from a dominant to a recessive level occurs. This indicates to each *M4KCAN* the end of Overload frames and each node simultaneously starts the transmission of six more recessive bits.

Note: The earliest time an Overload frame can be transmitted is at the first bit time of the Interframe Space Field. This is contrary to the Error frame and allows the *M4KCAN* to differentiate between the Error frame and Overload frame.

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

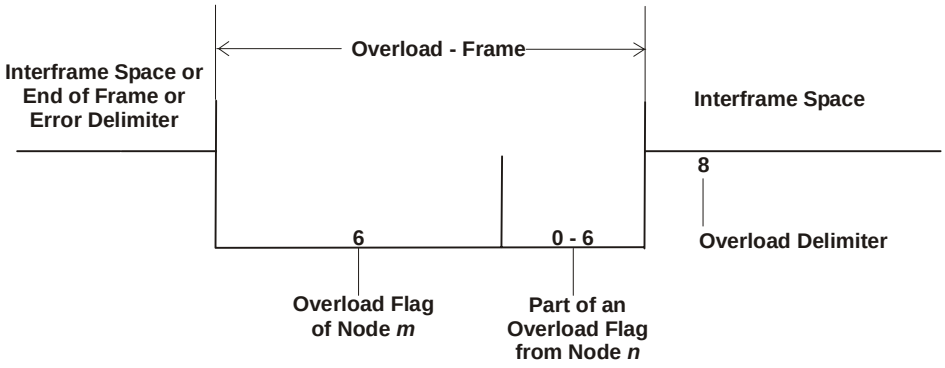


Figure A-4 Overload Frame Format

A.1.4.1 Interframe-space

Data frame and Remote frame are separated from preceding frames by an Interframe Space consisting of the Intermission bit field and a possible Bus Idle time. An Error frame is not preceded by an Interframe-space.

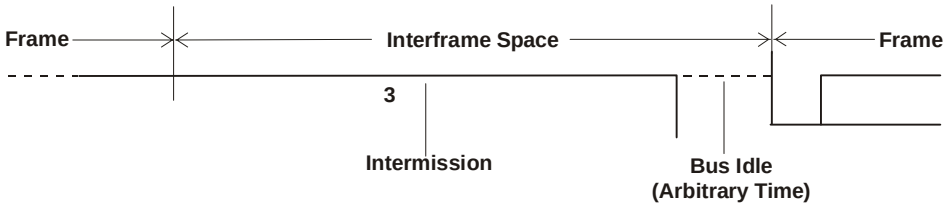


Figure A-5 Interframe Space Format

Intermission consists of three recessive bits. During Intermission time the *M4KCAN* will not start transmission of a frame. Intermission is a fixed time period for the *M4KCAN* to execute internal processes prior to the next receive or transmit task.

Data received within a Data frame will be stored in the communication buffer and the control bits are updated if no error condition has occurred through the last bit of the End of Frame field.

The Bus Idle time may be of arbitrary length. After the Interframe Space period, the *M4KCAN* looks for Bus Idle before initiating transmission, if requested by a USER. The detection of a dominant bit after Intermission or bus idle is interpreted by the *M4KCAN* as Start of Frame.

A.1.4.2 Coding Decoding

Coding

The frame segments (Start of Frame, Arbitration field, Control field, Data field and **CRC** sequence) are coded using bit stuffing. Whenever the transmit logic of the *M4KCAN* detects five consecutive bits of identical levels to be transmitted, the logic inserts a complement bit in the transmitted bit stream.

Bit stuffing is used to guarantee enough edges in the NRZ Bit Stream to maintain synchronization.

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

Decoding

Whenever the *M4KCAN* has received five identical consecutive bit levels in the received bit stream the logic automatically deletes the next bit from the data stream (destuffing). Some field formats do not use bit stuffing. In these cases the bit stuffing and destuffing logic is turned off.

A.1.4.3 Arbitration

In the case when two or more *M4KCAN*s start transmission concurrently, the bus access conflict is solved by a bitwise arbitration method during transmission of the arbitration field.

The transmit logic compares the bit level transmitted to the level monitored on the serial bus. Both the transmitter and receiver are on the bus at the same time. The transmit logic stops message transfer if a recessive bit was sent but a dominant bit was monitored. This method guarantees transmission of the message with the highest priority even if there is a collision during the Arbitration field of one or more Message Identifiers.

The *M4KCAN* protocol architecture requires each message used in the communication network to have a unique Identifier characterizing the type of data within the data field. Using this method, the Identifier assigns a name to the data frame and automatically implies the priority of the message.

As a result, the Identifier during bus access represents not only the message name but, more important, the priority of each specific message. Since the Most Significant Bit (MSB) of an Identifier is transmitted first, the Identifier with the smallest digital value has the highest priority for bus access.

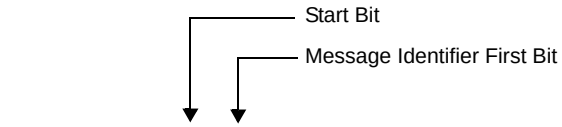
An Identifier should not be used for more than one specific message to ensure that two or more nodes never simultaneously start a transmission of a Data frame with the same message priority. Following this rule, bus access conflicts are resolved during the transmission of the Identifier.

One exception would be the simultaneous transmitter and receiver initiated frame transfer for the same message. If one *M4KCAN* generates a request for actual data of a certain type by transmitting a Remote frame and simultaneously, the *M4KCAN* responsible for this type of data starts the transmission, arbitration can not be solved by the Identifier itself and is not required.

To deal with this exception, the **RTR** bit is included in the Arbitration field. The **RTR** bit of the transmitter is always set dominant and, therefore, has a higher priority than the requesting *M4KCAN* (**RTR** bit recessive). This way the Remote frame request by the receiver gets an immediate response by the transmitter.

Example: Non-destructive prioritized bitwise arbitration

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



node 1:	0	0	0	1	0	1	1	0	0	0	1	Engine Speed	0B1 H
node 2:	0	1*	1	0	0	1	0	1	1	0	1	Coolant Temp.	32D H
node 3:	0	0	0	1*	1	0	1	1	0	1	0	Vehicle Speed	0DA H
Bus	0	0	0	1	0	1	1	0	0	0	1		0B1 H

Note: Node 1 wins the arbitration with the lowest identifier [0B1 H]

- Zero is the dominant bit on the bus

* Node detected the loss of arbitration and stopped transmitting but continues to receive the message

The CAN protocol architecture defines that each Communication Object used must have a unique Identifier characterizing the priority of the message. This allows bitwise arbitration of the bus if a conflict arises. The transmit logic compares the level monitored on the serial bus with that transmitted. The transmit logic immediately stops transmission if there is a conflict. This guarantees the data transfer of the Communication Object with the highest priority even if there is a collision.

The Error Detection mechanism is implemented in hardware for efficiency.

A.2 Error Detection and Confinement

A.2.1 Bit Error

During a transmit operation, the *M4KCAN* monitors the bus on a bit-by-bit basis. If the bit level monitored is different from the transmitted bit, a Bit error is signaled.

Exceptions: During **Arbitration**, a recessive bit can be overwritten by a dominant bit. In this case, the *M4KCAN* interprets a bit error as an arbitration loss.

During the **ACK-SLOT**, a transmitter may detect a falsified bit (recessive to dominant) meaning that at least one receiver has received the message correctly.

Note: Except during transmission of the Arbitration field and during the time window of the **ACK-SLOT**, all global and local errors at the transmitter are detected.

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

A.2.2 Bit Stuffing Error

As described earlier, the frame segments are coded by a method of bit stuffing.

There are two possibilities where bit stuffing errors may occur:

1. A disturbance generates more consecutive bits of equal level than allowed by the rule of bit stuffing. These errors are detected by all nodes.
2. A disturbance falsifies one or more of the five bits preceding the stuff bit. This error is not recognized by a receiver; however, if the error also appears at the transmitter, it will be detected as a Bit error (transmitter monitors bus as it transmits).

In any case, the error is detected by a receiver either by the bit stuffing mechanism (the stuff bit of the transmitter is not dropped but taken as an information bit) or by the **CRC** check.

A.2.3 CRC Error

To ensure the validity of a transmitted message, all receivers perform a **CRC** check. In addition to the information bits, the CRC includes control bits used for error detection.

Description of the CRC Code

The code used for the *M4KCAN* is a (shortened) BCH Code, extended by a parity check and the following attributes:

- 127 bits as maximum length of the code word
- length of the CRC sequence is 15 bits
- Hamming distance d = 6
  d = min A (x EXOR y) / x, y different code words
  A(x) = number of “recessive” bits in the code word x

$f(x) = (x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1)\ (x + 1)$

f(x) = 1100 0101 1001 1001

f(x) = C599 Hex

Burst errors are detected up to a length of 15 [degree of f(x)]. Multiple errors (number of disturbed bits at least d = 6) are not detected with a residual error probability of 3×10^{-5}.

A.2.4 Form Error

Form errors result from the violation of the fixed form of the following bit fields:

End of Frame

Interframe Space

ACK Delimiter

CRC Delimiter

During the transmission of these bit fields, an error condition is recognized if a “dominant” bit level is detected.

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

A.2.5 Error Detection Capabilities

Global errors, which occur at all fully functional nodes, are 100% detectable.

For local errors, e.g. errors which may appear at some nodes only, the shortened BCH Code extended by the parity check has the following error detection capabilities:

- Up to 5 single bit errors are detected 100% even if those errors are being distributed randomly within the code word.
- All single bit errors are detected if their total number within the code word is odd.
- The residual error probability of the **CRC** check is $2^{-15} = 3 \times 10^{-5}$. As an error may be detected by the **CRC** check, and/or by additional implemented error detection mechanism, the residual error probability is significantly less than 3×10^{-5}.

A.2.6 Error Confinement

Error Confinement is implemented on the *M4KCAN* as a self-checking mechanism for distinguishing "temporary errors" from "permanent failures". A permanent failure is noted when an average of one in eight messages is corrupted. This type of error condition can be caused by a defective connector, transmitter, receiver or a long lasting disturbance from outside the network. If the node continues to observe a failure over a period of time the node will be removed from the bus. A disconnected node is not placed again on the serial bus until the user has issued a software reset to the *M4KCAN*, and an *M4KCAN* internal delay time has elapsed.

The implementation of the error confinement consists of two counters (Receive Error Count and Transmit Error Count) and some control logic. These counters are modified according to a number of rules, which may be considered as the core of the error confinement. If a message is transmitted or received without an error the error-counter is decremented by a fixed number, if it is not already 0. The error counter is increased by a fixed number, if an error is detected on the serial bus. No access is provided to the error counters; however, two flags are provided (Error Status and Bus Status) as a summary of error events.

The count added to the error-counter depends on the type of the error detected. For instance, whenever a node detects and reports an error condition (Error flag), all system nodes will also detect an error condition due to that Error flag, even if the information up to that time was received error-free.

Note: 1. In case an error condition is not detected by all nodes at the exact same bit time, the node reporting the error first is more likely to be responsible for such an error condition compared to those nodes reacting to the error flag. Therefore, a node that is often responsible for error conditions, as mentioned above, is the origin of the error as a result of a defect.

In general, a defective node exchanges information for a short time so as to prevent it from loading the bus and slowing down other nodes on the bus.

An *M4KCAN* in the busoff state will neither transmit nor receive messages. In order to restart the *M4KCAN* it is necessary to reset the **Init** bit in the Control Register (00 H).

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

2. After this sequence, the CAN node will be active on the serial bus after 128 × 11 consecutive recessive bits.

A.2.7 *M4KCAN* States with Respect to the Serial Bus

The *M4KCAN* may be in one of three states:

Error Active

Error Passive

Busoff

Error-Active is the normal mode of operation, it is characterized by the *M4KCAN* transmitting an Error flag of 6 dominant bits in case a receive or a transmit error is detected.

An "Error Passive" *M4KCAN* does not send an Active Error flag (6 dominant bits). In this mode, the *M4KCAN* communicates on the bus but on detecting a receive or a transmit error, it sends a Passive Error frame (6 recessive bits). After transmitting a message, an "Error Passive"*M4KCAN* does not initiate another transmission immediately; instead, after the Interframe Space, it transmits 7 "recessive bits". If during this time period (7 recessive bits), another *M4KCAN* starts transmission on the bus, the "error passive" *M4KCAN* becomes the receiver.

A busoff *M4KCAN* does not transmit or receive any information; its output drivers are put in a float state. This state is indicated to the user by the **BOff** bit in the Status register (01 H).

For Error Confinement two error counters are implemented in the *M4KCAN* architecture:

- Transmit Error counter
- Receive Error counter

These counters are modified by the *M4KCAN* with no read or write access to the user. At the system start-up, there may be only one *M4KCAN* active on the bus. If this node transmits a message, it will not get an acknowledgement and hence detect an error; it will correspondingly repeat the message. Because of repeated transmissions, *M4KCAN* will become "Error Passive" but not busoff.

Similarly, an *M4KCAN* sending a wake-up message may become "Error Passive".

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

The information contained in this document is believed to be accurate. However, no responsibility is assumed by Excalibur Systems, Inc. for its use and no license or rights are granted by implication or otherwise in connection therewith. Specifications are subject to change without notice.

May 2011, Rev A-4

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