



## Artisan Technology Group is your source for quality new and certified-used/pre-owned equipment

- FAST SHIPPING AND DELIVERY
- TENS OF THOUSANDS OF IN-STOCK ITEMS
- EQUIPMENT DEMOS
- HUNDREDS OF MANUFACTURERS SUPPORTED
- LEASING/MONTHLY RENTALS
- ITAR CERTIFIED SECURE ASSET SOLUTIONS

### SERVICE CENTER REPAIRS

Experienced engineers and technicians on staff at our full-service, in-house repair center

### *InstraView*<sup>SM</sup> REMOTE INSPECTION

Remotely inspect equipment before purchasing with our interactive website at [www.instraview.com](http://www.instraview.com) ↗

### WE BUY USED EQUIPMENT

Sell your excess, underutilized, and idle used equipment. We also offer credit for buy-backs and trade-ins. [www.artisanng.com/WeBuyEquipment](http://www.artisanng.com/WeBuyEquipment) ↗

### LOOKING FOR MORE INFORMATION?

Visit us on the web at [www.artisanng.com](http://www.artisanng.com) ↗ for more information on price quotations, drivers, technical specifications, manuals, and documentation

**Contact us:** (888) 88-SOURCE | [sales@artisanng.com](mailto:sales@artisanng.com) | [www.artisanng.com](http://www.artisanng.com)

**MN-42916XM-001**  
**DD-42916XM-300**  
**DDC-429 VXI**  
**Software Interface**  
**(Word Serial Communication)**

The information provided in this User's Manual is believed to be accurate; however, no responsibility is assumed by Data Device Corporation for its use, and no license or rights are granted by implication or otherwise in connection therewith.

Specifications are subject to change without notice.



105 Wilbur Place, Bohemia, New York 11716-2482

**For Technical Support** – 1-800-DDC-5757 ext.7402 or 7384  
**Headquarters** – Tel: (516) 567-5600 ext. 7318  
Fax: (516) 567-7358  
**Northern New Jersey** – Tel: (908) 687-0041  
Fax: (908) 687-0470  
**West Coast** – Tel: (714) 895-9777  
Fax: (714) 895-4988  
**Europe** – Tel: +44-(0) 1635-811140  
Fax: +44-(0) 1635-32264  
**Asia/Pacific** – Tel: +81-(0) 3-3814-7688  
Fax: +81-(0) 3-3814-7689  
**World Wide Web** – <http://www.ddc-web.com>

All rights reserved. No part of this User's Guide may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Data Device



Rev B July 99

© 1996, 1999 Data Device Corp.

Corporation.

# Table of Contents

---

Introduction .....	1
1.1 License and Policies .....	1
1.1.1 Warranty.....	1
1.1.2 Update Policy .....	1
1.2 Conventions.....	1
1.2.1 Typographical Conventions.....	1
1.3 Where to Start .....	1
1.4 Firmware Highlights.....	2
Getting Started.....	3
2.1 The DDC-429 VXI Module .....	3
2.1.1 Host Commands.....	5
2.2 Initialization.....	6
2.3 Transmitting .....	7
2.3.1 Transmission Schedules.....	7
2.4 Receiving.....	8
2.5 Serial Port Control.....	9
2.6 Reading and Writing Discretes .....	10
2.7 Timers and Timestamps .....	10
2.8 Generating Interrupts.....	10
2.9 Error Handling.....	11
2.10 Utility Functions .....	11
Module Communication .....	12
Command Definitions .....	18
4.1 Configure Serial Port .....	18
4.2 Define Rx Channel.....	19
4.3 Define Rx Filter Words.....	21
4.4 Define Rx Interrupt Filter.....	21
4.5 Define Tx Channel .....	22
4.6 Define Tx Schedule Rates .....	24
4.7 Define Tx Schedule Table .....	24
4.8 Dequeue Errors .....	26
4.9 Error Message Request.....	26
4.10 Get Clock Time .....	27
4.11 Get Interrupt Source.....	27
4.13 Get Rx Filter.....	28
4.14 Get Schedule Rates.....	29
4.15 Halt Channel.....	30
4.16 Load Engineering Conversion Table.....	30
4.17 Modify Rx Channel.....	31
4.18 Modify Tx Channel .....	32
4.19 Read Input Discrete .....	33
4.20 Read Mailbox .....	33
4.21 Receive FIFO.....	34

4.22 Reset Card .....	36
4.23 Reset Clock .....	38
4.24 Serial Receive .....	39
4.25 Serial Transmit.....	39
4.26 Set Filter Enabling .....	40
4.27 Set Interrupt Conditions.....	40
4.28 Software Version.....	44
4.29 Start Channel.....	44
4.30 Transmit FIFO .....	45
4.31 Transmit Scheduled.....	46
4.32 Write Output Discrete.....	47
System Level Example.....	49
Customer Contact Report .....	51
Acronyms .....	52
Index.....	48

## List of Figures

---

Figure 2.1 ARINC 429 VXI Module Architecture .....	4
Figure 2.2 Transmission Schedule Table .....	7
Figure 2.3 Receiver Methods.....	9
Figure 2.4 Interrupt Vector.....	10
Figure 3.1 Device Talking Format.....	14
Figure 4.1 Transmission Schedule Table .....	25
Figure 4.2 Error Message Format .....	27
Figure 4.3 Arbitrary Block Program Data .....	38

## List of Tables

---

Table 1.1 Typographical Conventions .....	1
Table 2.1 Host Commands.....	5
Table 3.1 Error Messages .....	15

# Chapter 1

---

## Introduction

This document describes the DDC-429 VXI Software Interface for use with the DD-42916XM-300 VXI word serial Interface Modules. The following describes how to interface with the software that resides on the modules. For more information on the DDC-429 Modules, refer to your module user's manual.

### **1.1 License and Policies**

Please read this section first. It contains information about the product license, warranty and software updates.

#### **1.1.1 Warranty**

The warranty for this firmware is contained in your module user's manual.

#### **1.1.2 Update Policy**

We will update the product as new functions are developed or major bugs are fixed to the extent of the warranty period. We will send you further information on updates, as it becomes available. Updates after the warranty period expires may require a purchase at nominal charges.

### **1.2 Conventions**

Conventions used in this manual are specified in the following sections.

#### **1.2.1 Typographical Conventions**

**Table 1.1** shows the typographical conventions that are used in this manual.

**Table 1.1 Typographical Conventions**

<b>Type Style</b>	<b>Used For</b>
<b>Bold</b>	Important words or information, and table, figure and chapter references Function names
<b><i>Bold Italic</i></b>	DDC Part numbers

### **1.3 Where to Start**

The remaining chapters of this manual are organized as follows:

**Chapter 2** Contains an overview of how all of the functions relate to one another and what is needed to start programming the module

**Chapter 3** provides a detailed description of how to communicate with the module

**Chapter 4** Contains an alphabetical listing of all the setup functions on the module, how to use them, what parameters are passed to them, what errors can occur, and some examples

## **1.4 Firmware Highlights**

The DDC ARINC 429 VXI Module has firmware installed that allows a programmer access to the following module features in a simple, easy to understand format:

### **Transmitting**

- Queued transmission to any channel
- Scheduled label transmission to any channel
- programmable voltage levels

### **Receiving**

- “Mailbox” type receiving from any channel
- Queued receiving from any channel
- Filtering of received labels

### **Miscellaneous**

- Error handling functionality
- Programmable bit rate
- Programmable word size
- Programmable ARINC bus speed
- Programmable ARINC bit gap
- RS 232/RS 422 transmission/reception
- Programmable discrete input and output

# Chapter 2

---

## Getting Started

This chapter provides an overview of all the module's functions, how they relate to each other, what Initialization is necessary, and the order in which some functions must be used. It is recommended that you read this chapter in its entirety before trying to use any of the module's functions.

### **2.1 The DDC-429 VXI Module**

The basic DDC-429 VXI module architecture is a multi-processor design that provides reliable high-performance ARINC 429 reception and transmission. See **Figure 2.1** on the following page. There is a single "Board Processor" that transfers data from the host to and from the "Channel Processors". Each channel processor handles up to 2 configurable receive or transmit channels (high or low speed), and 2 low speed receive channels. The two configurable channels, when acting as transmitters, can output anywhere from 0 to 12 Volts. All channels can be configured for variable word sizes (2 – 32 bits), parity (EVEN/ODD/NONE), and variable frequency (.1 kHz – 120 kHz).

The DDC-429 VXI module communicates to the host via Standard Commands for Programmable Instruments (SCPI) word serial communication. The interrupts are used to indicate to the host that an important event warranting immediate attention has just occurred.

When transmitting, the host can request that the module send data sequentially from a First-In-First-Out (FIFO) queue, or can request that the module automatically send scheduled data and allow the host to update the data asynchronously.

When receiving, the host can get data sequentially from a FIFO, or get the most recent data for any label/Source-Destination Indicator (SDI) combination from a "mailbox" slot. The data placed in the receiver FIFO is timestamped by the module.

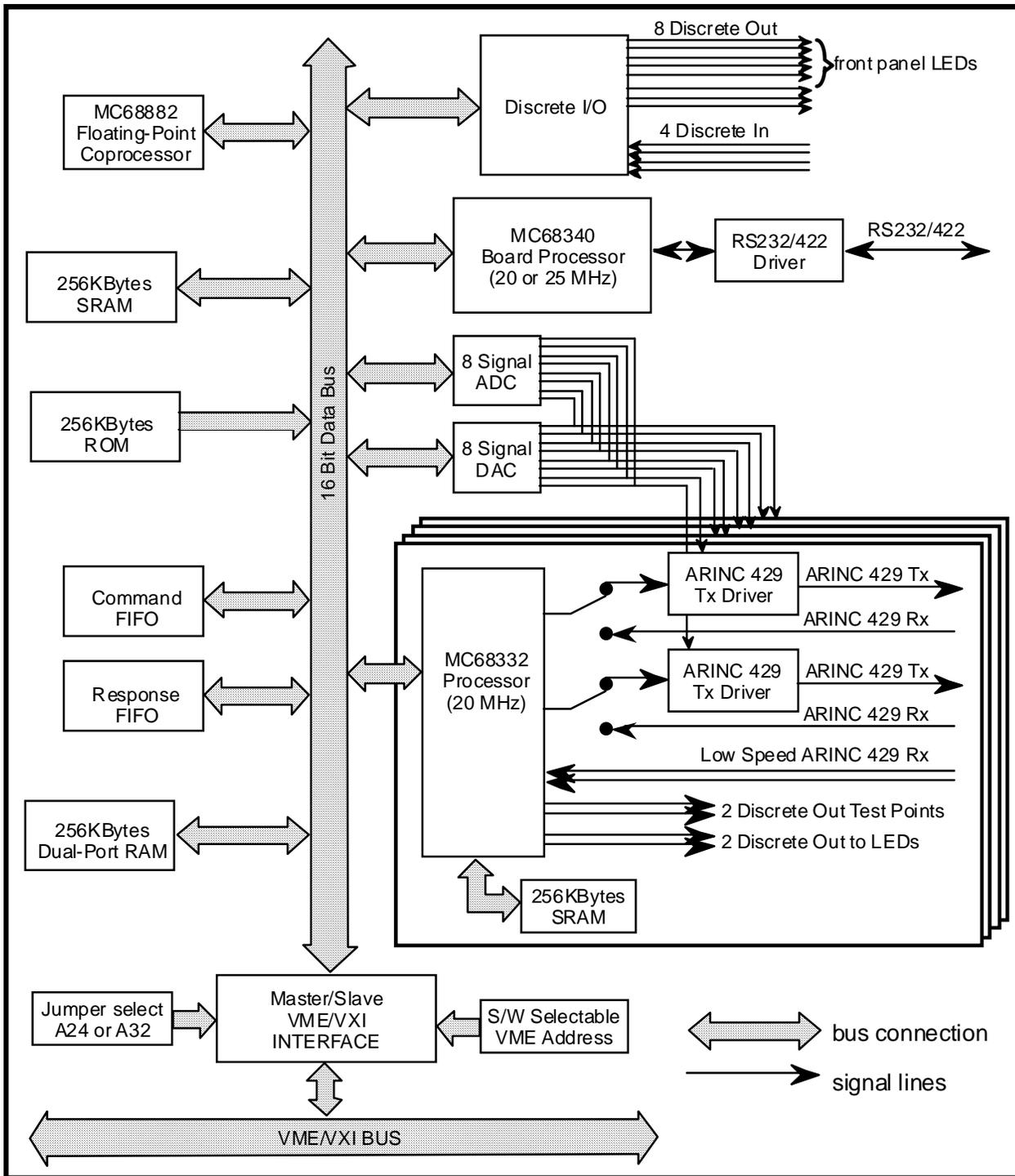


Figure 2.1 ARINC 429 VXI Module Architecture

The host can configure transmit and receive FIFOs stored in the module's memory to be any size desired.

The host can download an Engineering Unit Conversion Table to the module that enables the module to convert the raw ARINC 429 words to engineering units. This table determines the format for all ARINC 429 data. If no table has been defined, all words default to "raw" format.

There are a total of eight output discrettes. Three of these discrettes are host-controllable (D6, D7, and D8). The remaining five are used to control lights on the module's front panel (D5=FAIL Light, D1..D4=LED 0..3). All eight-output discrettes are available on the front panel connector.

For applications that use multiple modules, the timer in each channel processor used to timestamp received words, can be synchronized. Connecting all the modules through pin 1 on the 15-pin connector does this.

### 2.1.1 Host Commands

**Table 2.1** contains an alphabetical list of all commands available to the host.

**Table 2.1 Host Commands**

<b>Command</b>	<b>Description</b>
<b>Configure Serial Port</b>	Configures the serial port
<b>Define Rx Channel</b>	Configures a receive channel
<b>Define Rx Filter Words</b>	Defines the words to be placed in a receive FIFO
<b>Define Rx Interrupt Filter</b>	Defines the words that will generate receive interrupts
<b>Define Tx Channel</b>	Configures a transmit channel
<b>Define Tx Schedule Rates</b>	Defines a schedule that will continually be transmitted by the module
<b>Define Tx Schedule Table</b>	Loads the transmit schedule
<b>Dequeue Errors</b>	Retrieves the error counts
<b>Error message request</b>	Requests the next entry from the instrument's error/event queue
<b>Get Clock Time</b>	Reads the clock value used for the specified channel's time stamp
<b>Get Interrupt Source</b>	Determines the source of a module interrupt to the host
<b>Get Label Rates</b>	Sends a request to the card to return the scheduling rate for the specified SDI/label combination of the specified channel
<b>Get Rx Filter</b>	Returns the filter state of a particular SDI/label
<b>Get Schedule Rates</b>	Sends a request to the card to return the scheduling rates for all SDI/label combinations defined for the specified channel
<b>Halt Channel</b>	Stops a channel from operating

<b>Load Engineering Conversion Table</b>	Loads the engineering conversion table
<b>Modify Rx Channel</b>	Modifies the configuration for a receive channel
<b>Modify Tx Channel</b>	Modifies the configuration for a transmit channel
<b>Read Input Discretes</b>	Reads the current state of the input discretes
<b>Read Mailbox</b>	Returns data from the requested mailbox slots
<b>Receive FIFO</b>	Dequeues a specified number of elements from the Rx input queue
<b>Reset Card</b>	Resets the module to power-up state
<b>Reset Clock</b>	Resets the timestamp timers to zero
<b>Serial Receive</b>	Receives data from the serial port
<b>Serial Transmit</b>	Transmits data over the serial port
<b>Set Filter Enabling</b>	Enables or disables FIFO Rx filtering
<b>Set Interrupt Conditions</b>	Configures the module to generate interrupts to the host
<b>Software Version</b>	Retrieves the current module firmware version
<b>Start Channel</b>	Starts a channel operating
<b>Transmit FIFO</b>	Enqueues elements onto the Tx output queue
<b>Transmit Scheduled</b>	Enqueues elements onto the Schedule Table queue
<b>Write Output Discretes</b>	Writes the state of the host controllable output discretes

## 2.2 Initialization

When the module is first powered up, it automatically runs through its Power-On-Self-Test (POST) functions, and places the results in the output buffer and flags the host that return messages are waiting. This appears to the host as if a **Reset Card** command was initiated as “MASTER” (see **Section 4.16**).

In order for the module to function as desired, some initialization must be performed. The host must choose how each channel is to operate, whether a configurable channel will function as a transmitter, or receiver, the speed at which it will operate, and how the module will process the data for a channel. To configure a receiver channel, the **Define Rx Channel** command is used (see **Section 2.4**). To configure a transmitter channel, the **Define Tx Channel** command is used (see **Section 2.3**).

To start a channel operating the **Start Channel** command should be used, as the channels default to the “halted” state. (The **Halt Channel** command is used to turn the channel off again). The **Load Engineering Conversion Table** command allows the host to define word types and scale factors for each label/SDI combination for each separate channel.

## 2.3 Transmitting

Before trying to transmit on the DDC-429 VXI Module, the host should first configure the channel with the **Define Tx Channel** command, and start the channel operating with the **Start Channel** command. The **Define Tx Channel** command allows the host to select the channel's word sizes (2 – 32 bits), the size of the bit gap between transmitted words (1/2..32767), the parity (EVEN / ODD / NONE), the transmission voltage level (0 – 12 V), and the bit frequency (100 to 120,000 bits/second).

The firmware is capable of transmitting in two modes: FIFO and Scheduled. When FIFO mode is selected through the **Define Tx Channel** command, a First-In-First-Out queue is created, in which the host can place ARINC 429 data to be transmitted in the exact order in which it is placed in the queue (using the **Transmit FIFO** command). When Scheduled mode is selected through the **Define Tx Channel** command, a queue is created in which the host can place “new” ARINC 429 data (using the **Transmit Scheduled** command) that is transmitted when scheduled. The size of the FIFO and Schedule data queue are selectable and can be anywhere from 16 to 32,768 32-bit words.

If both FIFO and Scheduled modes are selected, the scheduled data always takes priority over the FIFO data. FIFO data words “fill-in” the gaps between the scheduled data.

After the channel has been defined once, the host can reconfigure the channel's word size, bit gap, parity, voltage level, and bit frequency using the **Modify Tx Channel** command.

### 2.3.1 Transmission Schedules

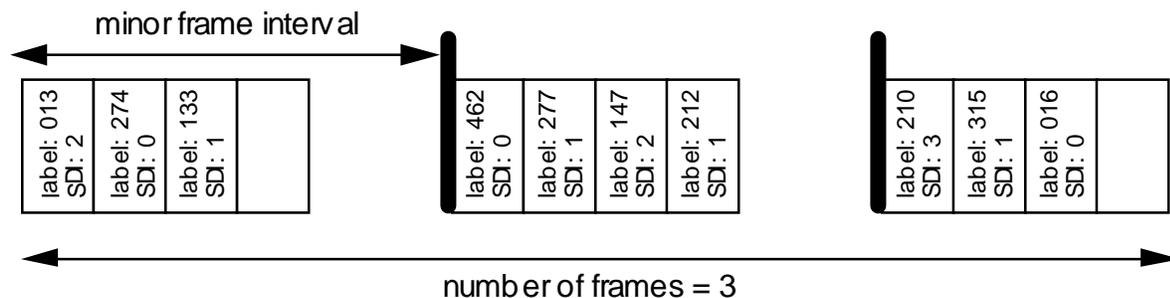
If scheduled data is to be transmitted, the user must select Scheduled mode through the **Define Tx Channel** command. The **Define Tx Schedule Table** or the **Define Tx Schedule Rates** command should then be used, before the **Start Channel** command.

If the **Define Tx Schedule Table** command is used, a schedule table determines when a label is to be transmitted.

The schedule table organizes a user-defined number of “minor frames” into a “major frame”. The module will transmit each minor frame in sequence until all the minor frames have been transmitted. This cycle of minor frames comprises one major frame. Once the major frame has been transmitted, the module will start over again with the first minor frame.

The user defines which ARINC 429 words (as defined by label/SDI combinations) are transmitted as part of each minor frame. The schedule table is configured when the user defines the minor frame interval, frame size, and a number of minor frames, which compose the major frame. The minor frame interval is the number of 100µs ticks in a minor frame. The frame size is defined as the maximum number of transmitted words in any one minor frame.

For example, if the number of minor frames is three, and the minor frame size is four, then after the three minor frames are transmitted once, they will repeat in sequence as shown in **Figure 2.2**.



**Figure 2.2 Transmission Schedule Table**

The label/SDI combinations are inserted into the table in the order received, via the **Define Tx Schedule Table** command. The data that is transmitted is not part of the **Define Tx Schedule Table** command. It is sent through the scheduled FIFO in Dual-Port RAM.

It is possible to define a schedule table that is too large to fit in the channel processor's memory. If this occurs, a MEM\_ERR is returned.

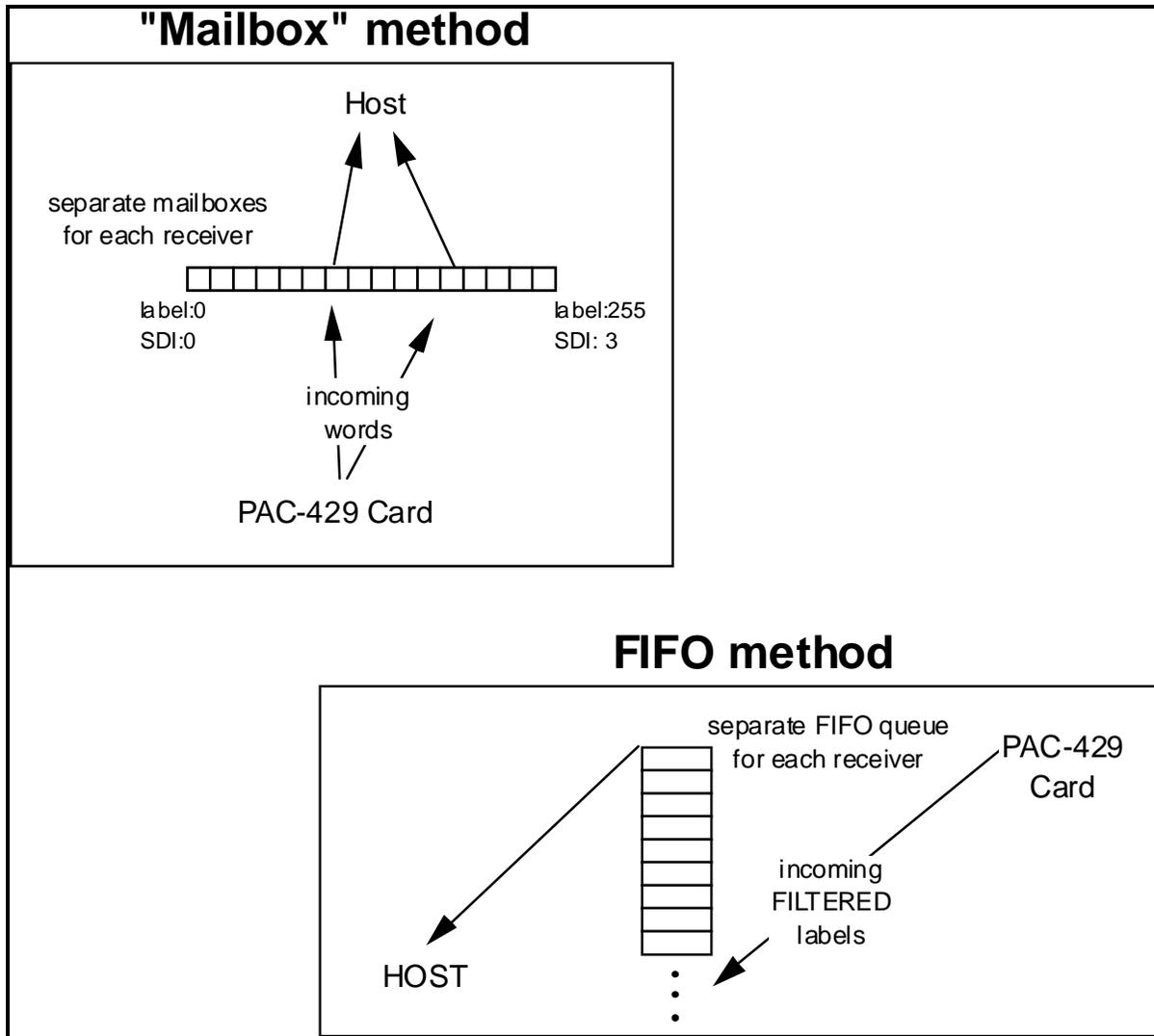
The **Set Interrupt Conditions** command allows the host to enable interrupts from the card at the start of each major frame.

If the **Define Tx Schedule Rates** command is used to determine when a label is to be transmitted, it is necessary to only send the labels and the rate at which they are to be transmitted. See the **Define Tx Schedule Rates** command for additional information.

## **2.4 Receiving**

Before trying to receive on the DDC-429 VXI Module, the host should configure the channel with the **Define Rx Channel** command, and start the channel operating with the **Start Channel** command. The **Define Rx Channel** command allows the host to select the channel's word sizes (2 – 32 bits), the size of the bit gap between received words (3..32767), the parity (EVEN / ODD / NONE), and the bit frequency (100 to 120,000 bits/second).

The firmware is capable of receiving in two modes: FIFO and Mailbox. See **Figure 2.3**. When FIFO mode is selected through the **Define Rx Channel** command, a First-In-First-Out queue is created in the module's memory, from which the host can read the received ARINC 429 data and timestamps in the exact order in which the words are received (using the **Receive FIFO** command). The timestamps have a 100  $\mu$ s resolution. When Mailbox mode is selected through the **Define Rx Channel** command, a "Mailbox" is created from which the host can read the most current data for a particular label/SDI combination (using the **Read Mailbox** command). The size of the FIFO is selectable and can be anywhere from 16 to 32,768 32-bit words. Both FIFO and Mailbox modes can use simultaneously.



**Figure 2.3 Receiver Methods**

When using the FIFO mode, the label/SDI combinations that are placed in the FIFO can be filtered using the **Define Rx Filter Words** command. This command allows the host to define a list of label/SDI combinations that will be placed in the FIFO when received. All Rx channels are automatically configured to receive all words until the first **Define Rx Filter Words** command is received. From that point on, the channel receives only the words defined by this command.

After the channel has been defined once, the host can reconfigure the channel's word size, bit gap, parity, and bit frequency using the **Modify Rx Channel** command.

## **2.5 Serial Port Control**

Serial port control is established using the **Configure Serial Port** command, which allows the host to configure the baud rate, data bits, start bits, stop bits, and parity. In addition, the host can select the size of the receive and transmit buffers. The **Serial Transmit** and **Serial Receive** commands allow the host to send and receive the serial data itself.

## 2.6 Reading and Writing Discretes

There are two basic commands for handling of the module's discrete inputs and outputs. The **Write Output Discrete** command is used for setting the value of all discrete outputs on the DDC-429 VXI Module, and the **Read Input Discrete** routine is used for reading the discrete inputs.

## 2.7 Timers and Timestamps

There is a timer on each channel processor that is used to timestamp the ARINC-429 words as they are received. All of these timers are initially set to zero when the module is first powered-up, or the **Reset Card** command is issued. The host using the **Reset Clock** command can reset any or all of these timers. Timestamps have 100  $\mu$ s resolution. Therefore, a timestamp of 10 would be  $100 \times 10 \mu\text{s} = 1000 \mu\text{s} = 1 \text{ ms}$ . Timestamps are 32 bits of unsigned data. Therefore, the timestamp "rollover" is approximately every 119 hours of operation.

Multiple modules can be synchronized using the **Reset Card** command (see **Section 4.16**).

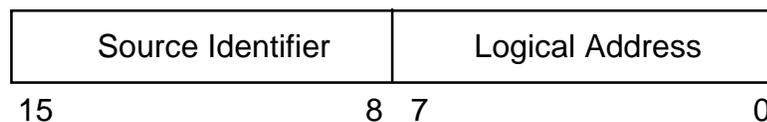
## 2.8 Generating Interrupts

The DDC-429 VME/VXI Module is capable of generating interrupts to the host. Interrupts can be generated when any of the following events occur:

- any ARINC 429 word is received in the Receive FIFO
- a particular label/SDI combination is received in the Receive FIFO
- a receive error occurs
- a Receive FIFO becomes half-full
- a Receive FIFO becomes not-empty
- a Transmit FIFO becomes empty
- serial data is received
- a response is placed in the Response FIFO

The **Set Interrupt Conditions** command allows the host to configure which of these conditions are enabled for each channel, and which VME Interrupt line (1..7) will be used by the module. The **Define Rx Interrupt Filter** command allows the host to select a subset of label/SDI combinations that will generate interrupts.

Once the module has been configured to generate interrupts, the module will place the interrupt vector shown in **Figure 2.4**, on the bus during the VME IACK cycle. The interrupt vector contains the logical address in the low order byte and the interrupt source identifier in the high order byte. The logical address is set using switches S5 and S4 on the module (see your X8 User Manual for details). This interrupt source identifier is the same number as the interrupt configuration used by the **Set Interrupt Conditions** command. It is possible for multiple interrupt causes to occur before the host acknowledges an interrupt. To ensure that the host is made aware of all interrupt causes, there is also a queue in Dual-Port RAM that allows the host to retrieve a sequential list of interrupt causes. The format of this queue is described in **Section 3.2.4**. The host's interrupt routine must perform the VME IACK cycle, which retrieves the interrupt vector and clears the interrupt.



**Figure 2.4** Interrupt Vector

## **2.9 Error Handling**

The module keeps track of how many errors of a particular kind have occurred for each channel. This includes the number of parity errors, and the number of short word errors. The **Dequeue Errors** command allows the host to retrieve the current error counts for all these error types. After the **Dequeue Errors** command is responded to, the error counts are all cleared to zero.

## **2.10 Utility Functions**

There is a single utility function that allows the host to determine the version number of the module's firmware. This command is called **Software Version**, and it returns three digits that determine the version "X,Y,Z".

# Chapter 3

## Module Communication

This section contains information regarding host/Module Communications.

The DDC-429 VXI Word Serial Module receives commands and sends back responses via SCPI word serial communications (Version 1993.0). For details about SCPI communication through the VXI backplane refer to SCPI 1993 Volume 1: Syntax and Style and IEEE Standard Codes, Formats, Protocols and Common Commands Std 488.2 and VMEbus Extensions for Instrumentation VXI bus Rev 1.4.

The scope of this manual is not to teach the reader word serial communication but to inform the reader device specific implementations.

All of these commands have a short and a long form. To send the short form only the bold part of the command needs to be sent. Some commands are suffixed with the channel number they will apply to which is represented by the **<ch#>** symbol. Only the commands suffixed by a “?” will automatically return a response.

Please note that some combinations of Basic and GPIB controllers may require a semicolon (;) command string terminator. HP Basic is an example.

Below is the DDC-429 SCPI Command tree which has been implemented to date:

### DDC- Module specific commands

#### Arinc429:

<b>:Channel&lt;ch#&gt;</b>	
<b>:DERrors?</b> <none>	<b>Dequeue Errors</b>
<b>:DIFW</b> <label>, <SDI>, <filter>	<b>Define Rx Interrupt Filter</b>
<b>:DRChannel</b> <com.>, <ws>, <gap>, <parity>, <speed>, <freq.>, <time stamp>, <fifo size>	<b>Define Rx Channel</b>
<b>:DRFWord</b> <SDI>, <label>, <filter>	<b>Define Rx Filter Words</b>
<b>:DTChannel</b> <com.>, <ws>, <gap>, <parity>, <Volt>, <speed>, <freq.>, <fifo size>, <sched.size>	<b>Define Tx Channel</b>
<b>:DTSRates</b> <list of{, <SDI>, <label>, <rates>	<b>Define Tx Schedule Rates</b>
<b>:DTSTable</b> <frame size>, <#frames>, <Interval>, <list of{, <frame#>, <SDI>, <label>}>	<b>Define Tx Schedule Table</b>
<b>:GCTime?</b> <none>	<b>Get Clock Time</b>
<b>GLRates?</b> <SDI>, <label>	<b>Get Label Rate</b>

:GRFilter? <SDI>, <label>	<b>Get Rx Filter</b>
:GSRates? <none>	<b>Get Schedule Rates</b>
:HALt <none>	<b>Halt Channel</b>
:LECTable <label>, <type>, <sig.digit>, <resolution>	<b>Load Engineering Conversion Table</b>
:MRChannel <ws>, <gap>, <parity>, <speed>, <freq.>	<b>Modify Rx Channel</b>
:MTChannel <ws>, <gap>, <parity>, <Volt>, <speed>, <freq.>	<b>Modify Tx Channel</b>
:RMailbox? <list of{, <SDI>, <label>}>	<b>Read Mailbox</b>
:SFEnabling <State>	<b>Set Filter Enabling</b>
:STart <none>	<b>Start Channel</b>
:CSPort<baud rate>, <#data bits>, <parity>, <receive buffer size>, <transmit buffer size>, <port type>	<b>Configure Serial Port</b>
:FIfo<ch#>	
:RECEive? <number to dequeue>	<b>Receive FIFO</b>
:TRANsmit <list of{, <SSM>, <SDI>, <label>, <data>}>	<b>Transmit FIFO</b>
:TScheduled <list If{<Valid>, <SSM>, <SDI>, <label>, <data>}>	<b>Transmit Scheduled</b>
:GISource? <none>	<b>Get Interrupt Source</b>
:RIDiscrete? <none>	<b>Read Input Discrete</b>
:RClock<processor 1>, <processor 2>, <processor 3>, <processor 4>	<b>Reset Clock</b>
:RESet <Mode>	<b>Reset Card</b>
:SIConditions <interrupt>, list of <conditions>	<b>Set Interrupt Conditions</b>
:SREceive? <none>	<b>Serial Receive</b>
:STRansmit <#chars>, list of <chars>	<b>Serial Transmit</b>
:VERsion? <none>	<b>Software Version</b>
:WODiscrete <D1>, <D2>, <D3>, <D4>	<b>Write Output Discrete</b>

SCPPI required commands:

:SYSTEM:

:ERRor? <none>	<b>Error message request</b>
----------------	------------------------------

At the time of this release the DDC-429 Module will respond to the following word serial commands as described by the VXI bus specification Rev. 1.4 Section E.1 pages 179-180. The implemented

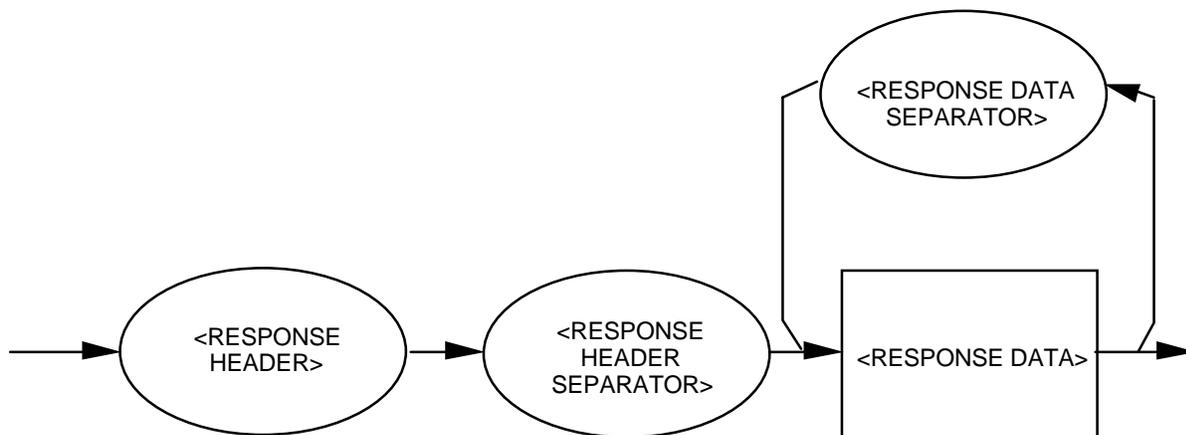
commands are **Byte Available**, **Byte Request**, **Read Protocol**, and **Begin Normal Operation**. Any other word serial commands will be ignored.

If an error occurs during the parsing or execution of the command then the VXI interrupt line number 7 will be asserted, informing the host that an error has occurred. It is the host's responsibility to handle this interrupt and to respond by sending back the SYST:ERR? command which will cause the DDC-429 Module to send back the error message generated. Several interrupts can be generated in a short period of time so the host may not have enough time to send a SYST:ERR? for each before the next interrupt occurs. Because of this, it is advised that the interrupt handler perform only an increment of the number of errors encountered, and that the actual SYST:ERR? messages are sent elsewhere in the code.

It will be necessary for the reader to refer to the IEEE 488.2 document for detailed explanations of how to implement the above command tree.

Parameters shall always be separated by commas. White space will always be ignored.

If a response is returned by a command it will be of the form:



**Figure 3.1 Device Talking Format**

Refer to IEEE 488.2 section 8 ,*Device Talking Formats*, for a complete description of the above terms.

Commands are sent to the DDC-429 Module using the VXI word serial command **Byte Available**. Commands are received using the VXI word serial command **Byte Request**. These two commands are described in the VXI bus System Specification Rev. 1.4 section E1. These messages are arbitrated using the Data In Ready (DIR) and Data Out Ready (DOR) bits in the response register. The **Byte Available** command may be sent only when the DOR bit is set and the **Byte Request** command may be sent only when the DIR bit is set. All messages are sent and received using the data low register in the DDC-429 A16 space. Reading and writing to this data low register is arbitrated using the read "ready and write ready" bits in the response register. Nothing may be written to the data low register until the write ready bit is set and nothing may be read from this register until the read ready bit is set.

Table 3.1 Error Messages

Error Message	Description	Value
No error	The command was successfully executed as commanded by the host	0
Bad Parameter	One or more of the parameters passed as part of the command was not a legal value	1
Channel in use	Error generated when host tries to re-define a channel using <b>Define Tx Channel</b> or <b>Define Rx Channel</b> when the channel has been previously defined	2
Channel low only	Error generated when the host tries to configure a channel for high speed when it is capable of low speed only	3
Channel not defined	Error generated when the host tries to use a channel that has not yet been defined	4
Channel not FIFO	Error generated when the host tries to setup filtering labels using the <b>Define Rx Filter Words</b> command when the channel was not configured as a FIFO receive channel using the <b>Define Rx Channel</b> command	5
Channel not receive	Error generated when the host tries to use a receive channel command (such as <b>Define Rx Filter Words</b> ) and the channel was configured as a transmitter using the <b>Define Tx Channel</b> command	6
Channel not scheduled	Error generated when the host tries to use the <b>Define Tx Schedule Table</b> command when the transmit channel was not configured for scheduled transmission using the <b>Define Tx Channel</b> command	7
Channel not transmit	Error generated when the host tries to use a transmit channel command (such as <b>Define Tx Schedule Table</b> ) and the channel was configured as a receiver using the <b>Define Rx Channel</b> command	8
Channel receive only	Error generated when the host tries to configure a channel as a transmitter when it is a receiver only channel	9
Checksum error	Error generated when a software load has been completed, but the checksum test does not pass	10
Memory error	Error generated when there is not enough RAM remaining to configure the FIFOs or Mailboxes requested by either the <b>Define Tx Channel</b> , <b>Define Rx Channel</b> , or <b>Configure Serial Port</b> command	11
Not implemented	Error generated when an unrecognized command is sent by the host	12
Channel processor error	This error indicates that there is a problem with the software on the module. If it is ever detected by the host, the user should contact DDC technical support	15
Channel processor overload	This error indicates that a channel processor has been overloaded. This occurs when a single channel processor is asked to handle more than one channel with either schedule transmission or time stamped reception. This error is returned by <b>Define Tx Channel</b> or <b>Define Rx Channel</b> commands	16
Time out error	This error indicates that a waiting period has elapsed before the event being waited for occurred	17

Serial Port not defined	The serial port has not yet been defined	18
Dequeue error	This error indicates that there is no data in the <b>Rx FIFO</b>	21
Mailbox error	This error indicates that an error occurred while retrieving the contents of a mailbox slot	22
Element not enqueued	This error will occur when a <b>Transmit FIFO</b> or <b>Transmit Scheduled</b> command fails to enqueue a word to be transmitted due to an internal queue overflow. This can be solved by either (a) making the queue sizes larger through the <b>Define Tx Channel</b> command, or by enqueueing data in smaller blocks.	23
Buffer Empty	Serial Rx Buffer is empty	24
No error	No errors are in error queue	0
Syntax error	This error indicates that a syntax error occurred while attempting to parse input string	-102
Command header error	Only a partial header was received	-110
Undefined header	This command is unrecognized	-113
Queue overflow	The error queue has overflowed	-350



# Chapter 4

## Command Definitions

This section contains detailed descriptions of all of the commands in the DDC-429 VXI module firmware. Each command description contains information about the command's functionality, the parameters passed, and the possible error messages that can be returned.

### 4.1 Configure Serial Port

The **Configure Serial Port** command configures the serial port on the module.

SCPI Command: **Arinc429:CSPort**

HOST	<u>Parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	baud rate	0=1200, 1=2400, 2=4800, 3=9600, 4=14400	<NRf>*
	# data bits	5, 6, 7, 8	<NRf>
	parity	NONE, EVEN, ODD, LOW, HIGH	<Character Program Data>
	receive buffer size	1..16 K	<NRf>
	transmit buffer size	1..16 K	<NRf>
	port type	RS422, RS232	<Character Program Data>

\*See Section 7.7.2 IEEE Standard Codes, Formats, Protocols and Common Commands Std 488.2.

EXAMPLE: A429:CSP 3, 7, ODD, 1024, 2048, RS232

This configures the serial port to have a baud rate of "9600". There will be "7" data bits and the parity will be odd. The receiver buffer size will be "1024" bytes and the transmit buffer size will be "2048" bytes. The port type will be RS232.

#### ERRORS

GENERATED: No error, Bad Parameter, Memory error, Channel processor error

#### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Serial Transmit**, **Serial Receive**

## 4.2 Define Rx Channel

The **Define Rx Channel** command allows the host to configure a receive channel. When the **Define Rx Channel** is complete, the channel is in the “halted” state, and must be started using the **Start Channel** command.

Normally the module swaps the order in which the 8 label bits are received on the bus to conform to the standard ARINC 429 bit ordering. However, if the word size is less than 32 bits, the module cannot determine how many bits are in the label, and therefore will not swap the label bits. In addition, the module will not calculate parity, and will not perform any engineering unit conversions or parity calculations.

**NOTE:** Only low-speed receivers can be set from 160 to 30,000 bits/sec.

SCPI Command: **Arinc429:CHannel<ch#>:DRChannel**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	FIFO/MAILBOX/BOTH	FIFO, MAILBOX, BOTH	<Character Program Data>
	word size	2..32	<NRf>
	inter-word gap (in # of half-bits)	6..32767 0=default value of 4 bits	<NRf>
	parity	NONE, EVEN, ODD	<Character Program Data>
	speed	LOW, HIGH, FREQ	<Character Program Data>
	frequency	number of bits/second (100 to 120,000 bits/second)	<NRf>
	timestamp in FIFO	ON, OFF	<Character Program Data>
	raw word in FIFO	ON, OFF	<Character Program Data>
	timestamp in mailbox	ON, OFF	<Character Program Data>
	raw word in mailbox	ON, OFF	<Character Program Data>
	FIFO queue size	16 words .. 32 K words	<NRf>

EXAMPLE: A429:CH0:DRC FIFO, 32, 0, ODD, HIGH, 0, ON, OFF, OFF, OFF, 1024

This will define channel 0 to be a receiver. This channel has a FIFO reception queue. The word size is “32” bits and the bit gap is the default 4 bits. The parity is odd and the speed is high. Since a set speed is

used the next parameter frequency is ignored. This receiver has the time stamp turned on, the raw data word turned off, timestamp in mailbox off, raw word in mailbox off and the FIFO queue size is "1024".

**ERRORS**

GENERATED: No error, Bad Parameter, Memory error, Channel in use, Channel low only

**RESPONSE**

PARAMETERS: NONE

SEE ALSO: **Modify Rx Channel, Define Rx Filter Words**

### 4.3 Define Rx Filter Words

The **Define Rx Filter Words** command allows the host to filter the words that are placed in the Receive FIFO, based on label/SDI combinations. The receive channels are automatically configured to receive all words until the first **Define Rx Filter Words** command is received. From that point on, the channel receives only those words defined by the command.

The host only needs to pass the label/SDI filters it wishes to change via any one **Define Rx Filter Words** command. Any previously set filters will not be affected.

SCPI Command:       **Arinc429:CHannel<ch#>:DRFWords**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	list of the following:		
	label	0..255	<NRf>
	SDI	0..3	<NRf>
	Filter	ON, OFF	<Character Program Data>

EXAMPLE:       A429:CH1:DRFW 0, 0, ON, 0, 1, OFF

This defines two receive filter words for channel 1. All incoming words with a SDI of “0” and a label of “0” will pass through the filter and be recorded. All incoming words with a SDI of “0” and a label of “1” will not pass through the filter.

ERRORS

GENERATED: No error, Bad Parameter, Channel not receive, Channel not FIFO

RESPONSE

PARAMETERS:       NONE

### 4.4 Define Rx Interrupt Filter

The **Set Interrupt Conditions** command is used to turn on the “word received” interrupt for a particular channel. Then the **Define Rx Interrupt Filter** command allows the host to select a subset of label/SDI combinations that will generate interrupts.

If the **Define Rx Interrupt Filter** command is not used, all words received and placed in the receive FIFO will generate an interrupt.

SCPI Command:       **Arinc429:Channel<ch#>:DIFW**

HOST	<u>parameter</u>	<u>valid values</u>	<u>size</u>
PARAMETERS:	list of the following:		
	label	0..255	<NRf>

SDI	0..3, 15=don't care	<NRf>
Filter	ON, OFF	<Character Program Data>

EXAMPLE: A429:CH10:DIFW 255, 0, ON

This command will cause the module to generate an interrupt only for label/SDI combination 255/0 on channel 10.

RESPONSE

STATUS: No error, Bad Parameter, Channel not receive,  
Channel not FIFO

RESPONSE

PARAMETERS: NONE

## 4.5 Define Tx Channel

The **Define Tx Channel** command allows the host to configure a transmit channel. When the **Define Tx Channel** is complete, the channel is in the "halted" state, and must be started using the **Start Channel** command.

Normally the module swaps the order in which the 8 label bits are transmitted on the bus to conform to the standard ARINC-429 bit ordering. However, if the word size is less than 32 bits, the module cannot determine how many bits are in the label, and therefore will not swap the label bits. In addition, the module will not perform any engineering unit conversions or parity calculations.

To transmit words longer than 32 bits, set the inter-word gap to 1 half-bit. This will cause the module to stream all the transmit words together, making them look to a receiving system like one long word.

SCPI Command: **Arinc429:CHannel<ch#>:DTChannel**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	FIFO/SCHED/BOTH	FIFO, SCHEDULED, BOTH	<Character Program Data>
	word size	2..32	<NRf>
	inter-word gap (in # of half-bits)	1..32767 0=default value of 4 bits	<NRf>
	parity	NONE, EVEN, ODD	<Character Program Data>
	voltage level	0=0 V to 255=12 V, -1=Default of 10 V	<NRf>
	speed	LOW, HIGH, FREQ	<Character Program Data>
	frequency	number of bits/second (100 to 120,000 bits/second)	<NRf>

FIFO queue size	16 words .. 32 K words	<NRf>
Scheduled queue size	16 words .. 32 K words	<NRf>

EXAMPLE: A429:CH2:DTC SCHEDULED, 24, 5, EVEN, -1, LOW, 0, 0, 1024

This defines channel 2 to be a transmitter. This transmitter only sends scheduled data. The word size is “24” and the bit gap is “5” bits. The parity is even and the voltage level is the default of 10 volts. The transmit speed is low speed, so the next parameter (frequency) is not used. Since this transmit channel is scheduled only the FIFO queue size parameter is not used. This transmitter has a scheduled queue size of “1024”.

#### ERRORS

GENERATED: No error, Bad Parameter, Memory error, Channel in use, Channel receive only, Channel processor overload

#### RESPONSE

PARAMETERS: NONE

SEE ALSO: Modify Tx Channel, Define Tx Schedule Table

## 4.6 Define Tx Schedule Rates

The **Define Tx Schedule Rates** command allows the user to define a schedule that will continually be transmitted by the module.

The user must list all the label/SDI combinations going out and the rate at which they will be transmitted. If the user wishes to distinguish only by label and not label/SDI, the SDI parameter must be set to 15. The rate is the number of 100 micro second ticks between transmissions.

The data that is transmitted is not part of this command. The scheduled data is placed in the scheduled data FIFO using the **Transmit Scheduled** command.

SCPI: **Arinc429:CHannel<ch>:DTSRates**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	SDI	0..3 (15=Don't care)	<NRf>
	Label	0..255	<NRf>
	Rate	0..4294967295	<NRf>

EXAMPLE: **A429:CH0:DTSR 1,65,1000,0,32,2000,15,20,500**

This defines a transmission schedule for channel 0 containing 3 labels. The first word has SDI 1, label 65 and its rate is 100ms. The second word has SDI 0, label 32, and its rate is 200ms. The third word has SDI 15 (which means that we will not differentiate between SDI's only labels), label 20, and its rate is 50ms.

### ERRORS

GENERATED: No error, Bad Parameter, Channel not scheduled, Channel not transmit

### RESPONSE

PARAMETERS: NONE

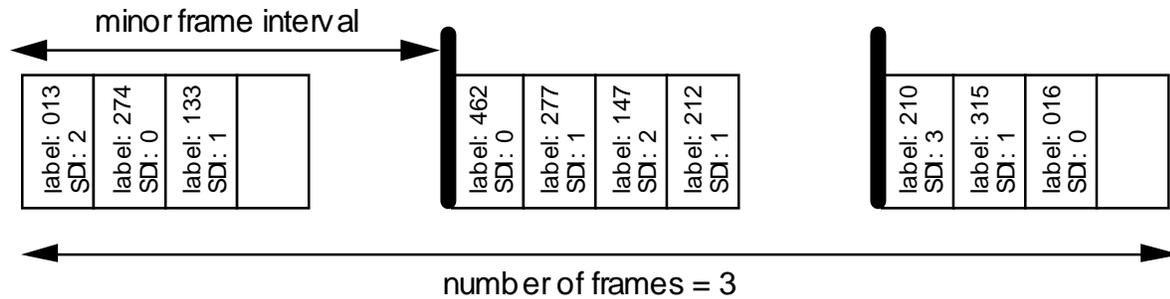
## 4.7 Define Tx Schedule Table

The **Define Tx Schedule Table** command allows the host to define a schedule that will continually be transmitted by the module.

The schedule table organizes a host-defined number of "minor frames" into a "major frame". The module will transmit each minor frame in sequence until all the minor frames have been transmitted. This cycle of minor frames comprises one major frame. Once the major frame has been transmitted, the module will start over again with the first minor frame.

The host defines which ARINC 429 words (as defined by label/SDI combinations) are transmitted as part of each minor frame. The schedule table is configured when the host defines the minor frame interval, frame size, and a number of minor frames, which compose the major frame. The minor frame interval is the number of 100µs ticks in a minor frame. The frame size is defined as the maximum number of transmitted words in any one minor frame.

For example, if the number of frames minor is three, and the minor frame size is four, then after the three minor frames are transmitted once, they will repeat in sequence as shown in **Figure 4.1**.



**Figure 4.1 Transmission Schedule Table**

The data for scheduled transmission is sent as part of the **Transmit Scheduled** command.

It is possible to define a schedule table that is too large to fit in the channel processor's memory. If this occurs, a MEM\_ERR is returned. The number of minor frames times the maximum minor frame size can not exceed 13,296.

The label/SDI combinations are placed in each minor frame in the order in which they are received via the **Define Tx Schedule Table** command.

The **Set Interrupt Conditions** command allows the host to enable interrupts from the card at the start of each major frame.

SCPI: **Arinc429:CHannel<ch#>:DTSTable**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	max minor frame size	1..4096	<NRf>
	# of minor frames	1..4096	<NRf>
	minor frame interval (# of 100µs ticks per minor frame)	1..32767	<NRf>
	list of the following:		
	frame #	1..4096	<NRf>
	label	0..255	<NRf>
	SDI	0..3	<NRf>

EXAMPLE: A429:CH2:DTST 10, 2, 1000, 1, 0, 0, 2, 0, 1

This defines a transmission schedule table for channel 2. The maximum frame size is "10" and the number of frames is "2". The frame interval is 100ms. Frame number "1" has the label/SDI combination of "0", "0" for the first slot. Frame number 2 has the label/SDI combination of "0", "1" for it's first slot.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not scheduled, Channel not transmit, Memory error

#### RESPONSE

PARAMETERS: NONE

## 4.8 Dequeue Errors

The **Dequeue Errors** command allows the host to determine how many errors have occurred since the last time the command was issued. Each time the **Dequeue Errors** command is sent, the module resets all the error counts to zero.

SCPI Command: **Arinc 429:CHannel<ch#>:DERrors**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

EXAMPLE: A429:CH3:DER?

This requests that the parity error count and short word error count for channel 3 be returned to the host.

### ERRORS

GENERATED: OK, Bad Parameter, Channel not defined

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	parity error count	0..32,767	<NRf>
	short word error count	0..32,767	<NRf>

EXAMPLE: A429:CH3:DER? 0,0

This response informs the host that no errors have occurred on channel 3 since the last dequeue errors request.

## 4.9 Error Message Request

The **Error Message Request** command returns next entry from the instrument's error/event queue. This command is called after an error has been reported via the assertion of the VME interrupt line number 7.

SCPI Command: **SYSTEM:ERRor?**

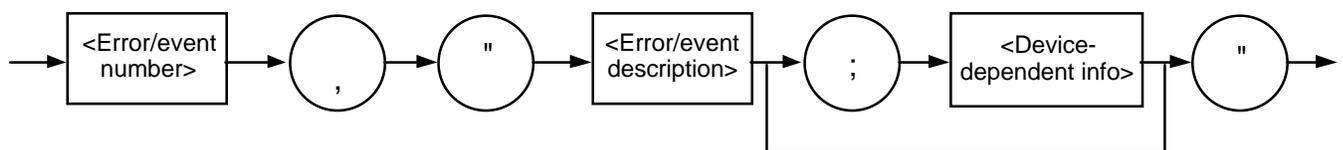
HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

EXAMPLE: SYST:ERR?

This example makes a request to the module to report the next error.

### RESPONSE

The DDC-429 Module shall respond to the SYST:ERR? query using the form described in **Figure 4.2** below:



### Figure 4.2 Error Message Format

The <error/event\_number> is a unique error/event descriptor. The error numbers returned by the DDC-429 Module are listed in this manual. The <error/event\_description> is a short description of the error/event, which is followed some of the time by optional information about the error.

The <Device-dependent info> informs the host which command created the error. For the **Transmit FIFO** and **Transmit Scheduled** commands the number of elements not enqueued by the command will also be included.

EXAMPLE: 1,"Bad Parameter;A429:F10:REC"

This response tells the host that the command "A429:F10:REC" had an incorrect parameter. The response number was "1". Refer to Table 3.1 for a list of the possible response messages/numbers and their meanings.

## 4.10 Get Clock Time

The **Get Clock Time** command allows the user to read the clock value used for the specified channel's time stamp. The value returned is the number of 100 micro second clock ticks that have occurred since the clock was started or the last **Reset Clock** command was given for this channel.

SCPI: Arinc429:CHannel<ch>:GCTime?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
------	------------------	---------------------	-------------

PARAMETERS: NONE

EXAMPLE: A429:CH4:GCT?

This command tells the card to return the clock value for channel 4.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel processor error

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
----------	------------------	---------------------	-------------

PARAMETERS: Clock 0..4294967295 <Nrf>

EXAMPLE: A429:CH4:GCT? 27364

This response informs the user that the clock on channel 4 reads 27364.

## 4.11 Get Interrupt Source

The **Get Interrupt Source** command will indicate the reason for an interrupt to the host. The interrupt source corresponds to the numbers configured through to the **Set Interrupt Conditions** command. For example, a 7 tells the host that the interrupt sent was for a word received on channel 7.

If three interrupts have been received since the last **Get Interrupt Source** command was sent then the next **Get Interrupt Source** command will return the source of the first interrupt. Subsequent **Get Interrupt Source** commands will retain the next interrupt source.

SCPI Command: Arinc429:GISource?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
------	------------------	---------------------	-------------

PARAMETERS: NONE

EXAMPLE: A429:GIS

This will cause the module to send back to the host the next pending interrupt source.

#### ERRORS

GENERATED: No error, Bad Parameter

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	Interrupt source	See the <b>Set Interrupt Conditions</b> command	<NRf>
	data was lost	1 = TRUE, 0 = FALSE	<NRf>

EXAMPLE: A429:GIS? 17, 0

This response tells the host that the source of the first pending interrupt was condition “17”. This tells the host that a receive error occurred for channel 1. The data lost flag is “0” which means that no interrupt sources have been lost.

SEE ALSO: **Set Interrupt Conditions** 4.12 Get Label Rate

The **Get Label Rate** command sends a request to the card to return the scheduling rate for the specified SDI/label combination for the specified channel.

SCPI: Arinc429:CHannel<ch>:GLRate?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	SDI 0..3 <NRf>		
	Label 0..255 <NRf>		

EXAMPLE: A429:CH3:GLR? 3,70

This command asks the card for the rate at which SDI 3, label 70 is being transmitted at for channel 3.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not transmit, Channel not scheduled

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	SDI 0..3 <NRf>		
	Label 0..255 <NRf>		
	Rate 0..4294967295 <NRf>		

EXAMPLE: A429:CH3 :GLR? 3,70,2500

This response informs the user that SDI 3, label 70 is being transmitted every 250ms.

## 4.13 Get Rx Filter

The **Get Rx Filter** command allows the user to determine the filter state of a particular label/SDI.

SCPI: Arinc429:CHannel<ch>:GRFilter?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
------	------------------	---------------------	-------------

PARAMETERS: SDI 0..3 <NRf>  
Label 0..255 <NRf>

EXAMPLE: A429:CH2:GRF? 3,45

This command asks the card for the filter state of label 45, SDI 3.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not receive, Channel not FIFO,

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
----------	------------------	---------------------	-------------

PARAMETERS: SDI 0..3 <NRf>  
Label 0..255 <NRf>  
State ON, OFF <CPD>

EXAMPLE: A429:CH2: GRF? 3,45,OFF

This response informs the user that the requested filter state for channels 2 is SDI 3, Label 45 is OFF. This means that this SDI/Label combination will not be placed into the Rx FIFO if received. This SDI/Label will be filtered out.

## 4.14 Get Schedule Rates

The **Get Schedule Rates** command sends a request to the card to return the scheduling rates for all SDI/label combinations defined for the specified channel.

SCPI: Arinc429:CHannel<ch>:GSRates?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
------	------------------	---------------------	-------------

PARAMETERS: NONE

EXAMPLE: A429:CH3:GSR?

This command asks the card for the SDI/label rates defined for channel 3.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not transmit, Channel not scheduled

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
----------	------------------	---------------------	-------------

PARAMETERS: SDI 0..3 (15=Don't care) <NRf>  
Label 0..255 <NRf>  
Rate 0..4294967295 <NRf>

EXAMPLE: A429:CH3 :GSR? 1,55,1000,2,56,2000

This response informs the user that there are currently two SDI/label rates defined for channel 3. SDI 1, label 55 is being transmitted every 100ms and SDI 2, label 56 is being transmitted every 200ms.

## 4.15 Halt Channel

The **Halt Channel** command will halt a channel after it has been started using the **Start Channel** command.

SCPI Command: **Arinc429:CHannel<ch#>:HAIt**

HOST parameter valid values Type

PARAMETERS: NONE

EXAMPLE: A429:CH4:HA

This command halts channel 4.

### ERRORS

GENERATED: No error, Bad Parameter, Channel not defined

### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Start Channel**

## 4.16 Load Engineering Conversion Table

The **Load Engineering Conversion Table** command allows the host to load a conversion table that the module can use to convert the raw ARINC 429 words to engineering units. The **Define Rx Channel** or **Define Tx Channel** command must be issued for a channel before the **Load Engineering Conversion Table** command can be used. After an engineering conversion table has been loaded, the data sent to the module is always the converted data in the format that is shown in **Table 4.1**.

The host can define any mix of word types, (i.e., some words can be BCD, some BNR, and some UNDEF). Similarly the host does not need to define conversions for all possible label/SDI combinations on a particular channel. The host needs only to define conversions for those words it plans to use. Before using the **Load Engineering Conversion Table** command, all label/SDI combinations are set to UNDEF.

Table 4.1 Engineering Units Conversion Table

Word Type	Input/Output Format	ARINC 429 Word Format (Attachment 6)
Alpha-Numeric	Three bytes of alpha numeric data	Alpha/Numeric (ISO Alphabet No. 5) Data - Intermediate Word Format
Binary	Standard Floating Point Format	Generalized BNR Word Format
Binary Coded Decimal (BCD)	Standard Floating Point Format	Generalized BCD Word Format
BCD Raw	ARINC 429 Raw Word (Hexadecimal representation)	Generalized BCD Word Format
Discrete	19 bits of extracted discrete data – (Bits 11 through 29 of the raw ARINC 429 word) (Hexadecimal representation)	Discrete Word Format
Latitude/Longitude	Standard Floating Point Format	BCD Encoding of Latitude and Longitude

Undefined (UNDEF)	ARINC 429 Raw Word (Hexadecimal representation)	None
-------------------	---	------

**NOTE:** If the host tries to transmit data that exceeds the maximum value that can be transmitted based on the resolution of an ARINC 429 word, the truncated value is transmitted, but no indication is given to the host that an overflow occurred.

**NOTE:** If the parity defined through **Define Rx Channel** or **Define Tx Channel** commands is “NONE”, the **Load Engineering Conversion Table** command cannot be used for that channel. This means all words on that channel must be “Undefined” so that the parity bit can be chosen by the host for each data word.

SCPI Command: **Arinc429:CHannel<ch#>:LECTable**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	list of the following:		
	label	0..255	<NRf>
	word type	UNDEF DISCRETE BCD BNR ALPHA BCD_RAW	<CPD>
	significant digits	1..20	<NRf>
	resolution	IEEE Standard Floating Point Format	<NRf>

EXAMPLE: A429:CH1:LECT 24, BCD, 4, .01

This tells the module that all ARINC words on channel 1 with a label of “24” (Decimal) are of “BCD” format, have “4” significant digits and that the resolution is “0.01”.

#### ERRORS

GENERATED: No error, Bad Parameter

#### RESPONSE

PARAMETERS: NONE

## 4.17 Modify Rx Channel

The **Modify Rx Channel** command allows the host to reconfigure a receive channel.

SCPI Command: **Arinc429:CHannel<ch#>:MRChannel**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	word size	2..32	<NRf>
	inter-word gap (in # of half-bits)	6..32767 0=default value of 4 bits	<NRf>

parity	NONE, EVEN, ODD	<Character Program Data>
speed	LOW, HIGH, FREQ	<Character Program Data>
frequency	number of bits/second (100 to 120,000 bits/second)	<NRf>

EXAMPLE: A429:CH2:MRC 32, 6, NONE, Low, 0

This will modify receiver channel 2. The word size is set for “32” bits and the bit gap is “6” bits. No parity is used and the speed is low. Since the low speed reception is used the last parameter is not used but must still be sent.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not receive

#### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Define Rx Channel**

## 4.18 Modify Tx Channel

The **Modify Tx Channel** command allows the host to reconfigure a transmit channel.

SCPI Command: **Arinc429:CHannel<ch#>:MTCchannel**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	word size	2..32	<NRf>
	inter-word gap (in # of half-bits)	1..32767 0=default value of 4 bits	<NRf>
	parity	NONE, EVEN, ODD	<Character Program Data>
	voltage level	0=0 V to 255=12 V, -1=Default of 10 V	<NRf>
	speed	LOW, HIGH, FREQ	<Character Program Data>
	frequency	number of bits/second (100 to 120,000 bits/second)	<NRf>

EXAMPLE: A429:CH3:MTC 32, 0, ODD, 180, FREQ, 12000

This modifies transmit channel 3. The word size is set to “32” and the bit gap is the default “4” bits. The parity is set to odd and the output voltage level is set to 8.47 volts. The frequency is set to “12,000 bits”/second.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not transmit

#### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Define Tx Channel**

## 4.19 Read Input Discrete

The **Read Input Discrete** command reads the values for all four of the input discretets.

SCPI Command: **Arinc429:RIDiscrete?**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
------	------------------	---------------------	-------------

PARAMETERS: NONE

EXAMPLE: A429:RID?

This will cause the module to return the contents of the input discrete word.

#### ERRORS

GENERATED: No error, Bad Parameter

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	discrete 1	0=OFF, 1=ON	<NRf>
	discrete 2	0=OFF, 1=ON	<NRf>
	discrete 3	0=OFF, 1=ON	<NRf>
	discrete 4	0=OFF, 1=ON	<NRf>

EXAMPLE: A429:RID? 0, 1, 1, 0

This response informs the host that input discrete “1” is off, that input discrete “2” and “3” are on, and that input discrete “4” is off.

SEE ALSO: **Write Output Discrete**

## 4.20 Read Mailbox

If configured through the **Define Rx Channel** command, there will be a segment of memory that contains a “slot” for each label/SDI combination for any receive channel. The host can read the “latest” data from a mailbox slot when it is convenient. The DDC-429 Module writes the data whenever it is received. The **Read Mailbox** command returns the contents of the specified mailbox.

Within any 32 bit raw (UNDEF) ARINC 429 word, the parity bit (bit #32) will contain the following depending on what parity the channel was configured for:

EVEN/ODD Parity error flag 0=OK, 1=ERROR  
 NONE Raw parity bit  
 SCPI Command: **Arinc429:CHannel<ch#>:RMailbox?**

HOST

PARAMETERS:	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
	SDI	0 3	<NRF>
	Label	0 .. 255	<NRF>

EXAMPLE: A429:CH2:RM? 2, 125

This requests the contents of the mailbox slot for SDI 2, label 125 for channel 2.

ERRORS

GENERATED: No error, Bad Parameter, Mailbox error

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	Data	varies depending upon which ARINC type the data is.	<STRING PROGRAM DATA>*
	Timestamp (If enabled in the <b>Define Rx Channel</b> command)	0..4,294,967,295	<Non-Decimal Numeric Program Data>
	Raw Data (If enabled in the <b>Define Rx Channel</b> command)	0..0xFFFFFFFF	<Non-Decimal Numeric Program Data>

\* Since the data type being enqueued onto the Tx queue varies according to the ARINC data type used, **Data** is sent as a string so that the corresponding format can be sent. If the ARINC data type is **Binary** then Data could be "34.5". If the ARINC data type is **Discrete** then Data could be "AF49". If the ARINC data type is **alphanumeric** then Data could be "XYZ". Refer to the **Load Engineering Conversion Table** command for further explanation as to what form the **Data** will take.

EXAMPLE: A429:RM2 "23.56",#H102310CA,#H6024D001

This response tells the user that the data in the mailbox slot is 23.56, that this channel is configured to receive timestamps in the mailbox and that the timestamp is 0x102310CA, and that this channel is configured to receive raw words in the mailbox and that the raw word received is 0x6024D001.

SEE ALSO: **Load Engineering Conversion Table, Transmit FIFO, Receive FIFO, Transmit Scheduled**

## 4.21 Receive FIFO

The **Receive FIFO** command removes the specified number of elements from the requested reception queue.

The timestamp element is optional. It is only part of the response if the timestamp parameter passed to the **Define Rx Channel** command was "ON". For details on the timestamp, see **Section 2.7**.

The SSM is only calculated by the module for binary, BCD, BCD Raw, and lat/long data words as defined by the **Load Engineering Conversion Table** command.

The data was lost flag indicates whether the module's internal receive FIFO overflowed and received data was lost. This generally occurs when the host does not dequeue data often enough.

Within any 32 bit raw (UNDEF) ARINC 429 word, the parity bit (bit #32) will contain the following depending on what parity the channel was configured for:

EVEN/ODD	Parity error flag 0=OK, 1=ERROR
NONE	Raw Parity bit

SCPI Command: **Arinc429:Fifo<ch#>:RECEive?**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	number of elements to dequeue	This can vary	<NRf>

EXAMPLE: A429:FI0:REC? 1

This makes a request to dequeue 1 element from the receive FIFO for channel 0.

#### ERRORS

GENERATED: No error, Bad Parameter, Channel not defined, Dequeue error

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	List of		
	data was lost	1=data lost 0=no data lost	<NRf>
	SSM	(varies)	<Non-Decimal Numeric Program Data>
	SDI	(0 .. 3)	<Non-Decimal Numeric Program Data>
	label	(0 .. 255)	<Non-Decimal Numeric Program Data>
	Data	varies depending upon ARINC data type and resolution used	<STRING PROGRAM DATA> *
	Time stamp (If enabled in the <b>Define Rx Channel</b> command)	0 .. 4,294,967,295	<Non-Decimal Numeric Program Data>
	Raw Data (If enabled in the <b>Define Rx Channel</b> command)	0 .. 0xFFFFFFFF	<Non-Decimal Numeric Program Data>

\* Since the data type being enqueued onto the Tx queue varies according to the ARINC data type used, **Data** is sent as a string so that the corresponding format can be sent. If the ARINC data type is **Binary** then Data could be “34.5”. If the ARINC data type is **Discrete** then Data could be “AF49”. If the ARINC data type is **alphanumeric** then Data could be “XYZ”. Refer to the **Load Engineering Conversion Table** command for further explanation as to what form the **Data** will take.

EXAMPLE: A429:FI0:REC? 0, #H00, #H02, #H20, “23.56”, #H102310CA

This response tells the host that no data was lost and that the element dequeued from channel 0 is:

```
SSM    = 0    (in Hex)
SDI    = 2    (in Hex)
Label  = 20   (in Hex)
Timestamp = 102310CA (in Hex) = 27,073.4 seconds
Data   = 23.56
```

SEE ALSO: **Transmit Scheduled, Load Engineering Conversion Table, Transmit FIFO**

## 4.22 Reset Card

The **Reset Card** command resets the DDC-429 VXI module to its power-up state. It automatically runs the Power-On-Self-Tests (POST), and returns a list of failures. The **Reset Card** command is implicit on power-up. The list of POST failures is placed in the output buffer each time the module is powered-up. The host must wait for this response in the output buffer before sending the module commands.

In order to synchronize the timestamps on multiple modules, each module must be connected via pin 1 on the 15-pin connector. When the modules are first powered up, they are not synchronized. The **Reset Card** command is used to designate which module generates the “master” time. Each module must be reset first as a “slave”. Then the **Reset Clock** command must be issued to each module to zero all the counters. Lastly, one of the modules is reset as a “master” which causes all the cards to start in a synchronized state. It makes no difference which module is designated as the master.

For single module applications, the timer synchronization parameter passed to **Reset Card** must be master.

SCPI Command:	<b>Arinc429:RESet</b>		
HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	timer synchronization	SLAVE, MASTER	<Character Program Data>

EXAMPLE: A429:RES MASTER

This will reset the module as the master.

ERRORS

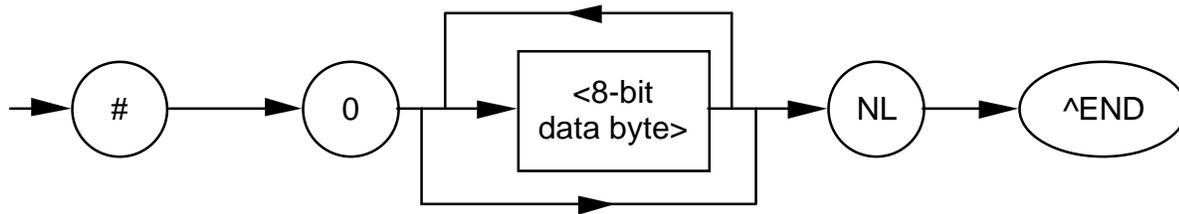
GENERATED: No error

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	possible list of failures		<Arbitrary Block Program Data>*

RAM	2
test failure	
channel processor 1 RAM failure	3
channel processor 2 RAM failure	4
channel processor 3 RAM failure	5
channel processor 4 RAM failure	6
signal output voltage failure for digital to analog converter 1	8
signal output voltage failure for digital to analog converter 2	9
signal output voltage failure for digital to analog converter 3	10
signal output voltage failure for digital to analog converter 4	11
signal output voltage failure for digital to analog converter 5	12
signal output voltage failure for digital to analog converter 6	13
signal output voltage failure for digital to analog converter 7	14
signal output voltage failure for digital to analog converter 8	15
math co-processor failure	7
channel processor 1 failure	16
channel processor 2 failure	17
channel processor 3 failure	18
channel processor 4 failure	19

**NOTE:** These 8-bit data byte blocks are in hexadecimal notation so that two digits represent each 8-bit byte.

\*Arbitrary Block Program Data has the form shown in **Figure 4.3**.



**Figure 4.3 Arbitrary Block Program Data**

EXAMPLE: A429:RES #0070813

This response tells the host that after reset the module start up diagnostics discovered a math co-processor problem, a signal output voltage problem for digital to analog converter 1, and a channel processor failure.

## **4.23 Reset Clock**

The **Reset Clock** command resets any or all of the timers that timestamp received data.

SCPI Command: Arinc**429:RC**lock

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	Reset Ch. Proc. 1	YES, NO	<Character Program Data>
	Reset Ch. Proc. 2	YES, NO	<Character Program Data>
	Reset Ch. Proc. 3	YES, NO	<Character Program Data>
	Reset Ch. Proc. 4	YES, NO	<Character Program Data>

EXAMPLE: A429:RC YES, YES, YES, NO

This will reset the clocks for channel processors 0-2, but not for channel processor 3.

ERRORS

GENERATED: No error, Bad Parameter

RESPONSE

PARAMETERS: NONE

## 4.24 Serial Receive

The **Serial Receive** command allows the host to receive character data from the serial port.

The module reception software waits until a STX (Start of Text = 0x02) is received, and stores each character until an ETX (End of Text = 0x03) is received. The STX and ETX are stored as part of the string.

If the module detects that data has been lost (i.e., the previous string had not been retrieved by the host when a new string was received) the module will set the data lost flag to YES.

SCPI Command:       **Arinc429:SR**receive?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

EXAMPLE:     A429:SR?

This will cause the module to send back to the host the next serial string in the serial receive buffer.

### ERRORS

GENERATED: No error, Bad Parameter, Buffer Empty

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	# Characters	1..2048	<NRf>
	Data lost	0=NO, 1=YES	<NRf>
	list of string data	0..0xFF	<Arbitrary Block Program Data>

EXAMPLE:     A429:SR 4, 0, #041424344

This response informs the host that a string of four characters has been received. This string is "ABCD".

SEE ALSO:     **Configure Serial Port, Serial Transmit**

## 4.25 Serial Transmit

The **Serial Transmit** command allows the host to transmit character data via the serial port.

The module then transmits the string exactly as it is sent in the command. If a STX and ETX is needed, the host must include them in the string.

SCPI Command:       **Arinc429:STR**ansmit

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	# Characters	1..2048	<NRf>
	list of string data	0..0xFF	<Arbitrary Block Program Data>

EXAMPLE: A429:ST 6, #0024142434403

This sends a six-character string out the serial port. This string starts with an STX and ends with an ETX character. The string is “ABCD” (Hex numbers: 41, 42, 43, 44).

ERRORS

GENERATED: No error, Bad Parameter, Serial Port not Defined

RESPONSE

PARAMETERS: NONE

SEE ALSO: **Configure Serial Port, Serial Receive**

## 4.26 Set Filter Enabling

The **Set Filter Enabling** command allows the card to use the filters that have been assigned to it with the **Define Rx Filter Words** command. If the filter is disabled, then all words will be received without filtering. If this command is called with the filter enabled but no filters have been assigned to the card with the **Define Rx Filter Words** command, then no words will be received until some SDI/label filters are enabled.

SCPI: Arinc429:CHannel<ch>:SFEnabling?

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	State	ON, OFF	<CPD>

EXAMPLE: A429:CH7:SFE OFF

This command tells the card to not use the filter states given to it for channel 7 but to receive all words without filtering.

ERRORS

GENERATED: No error, Bad Parameter, Channel not receive, Channel not FIFO,

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

## 4.27 Set Interrupt Conditions

The **Set Interrupt Conditions** command allows the host to determine under what conditions the module will generate an interrupt. The list of interrupt conditions that is passed as a parameter to this command must be the complete list, as this command resets the entire list each time is used. The **Get Interrupt Source** command allows the host to determine which of the conditions generated the interrupt.

The interrupt for received words is generated only after a word is placed in the Receive FIFO. So label/SDI combinations that are being filtered out of the Receive FIFO will never cause an interrupt. In addition, the **Define Rx Interrupt Filter** command will allow the host to select a subset of label/SDI combinations that will generate interrupts when a receive word is placed in the Receive FIFO.

SCPI Command: Arinc429:SIConditions

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	VXI Interrupt	1..6 7 = use previously set interrupt	<NRf>
	conditions to turn on	0 = word received on ch0 1 = word received on ch1 2 = word received on ch2 3 = word received on ch3 4 = word received on ch4 5 = word received on ch5 6 = word received on ch6 7 = word received on ch7 8 = word received on ch8 9 = word received on ch9 10 = word received on ch10 11 = word received on ch11 12 = word received on ch12 13 = word received on ch13 14 = word received on ch14 15 = word received on ch15 16 = Rx Error Occurred ch0 17 = Rx Error Occurred ch1 18 = Rx Error Occurred ch2 19 = Rx Error Occurred ch3 20 = Rx Error Occurred ch4 21 = Rx Error Occurred ch5 22 = Rx Error Occurred ch6 23 = Rx Error Occurred ch7 24 = Rx Error Occurred ch8 25 = Rx Error Occurred ch9 26 = Rx Error Occurred ch10 27 = Rx Error Occurred ch11 28 = Rx Error Occurred ch12 29 = Rx Error Occurred ch13 30 = Rx Error Occurred ch14 31 = Rx Error Occurred ch15 32 = Tx FIFO Empty ch 0	<NRf> for each

- 33 = Tx FIFO Empty ch 1
- 34 = Tx FIFO Empty ch 2
- 35 = Tx FIFO Empty ch 3
- 36 = Tx FIFO Empty ch 4
- 37 = Tx FIFO Empty ch 5
- 38 = Tx FIFO Empty ch 6
- 39 = Tx FIFO Empty ch 7
- 40 = Serial Data Received
- 41 = Major frame occurred ch0
- 42 = Major frame occurred ch1
- 43 = Major frame occurred ch2
- 44 = Major frame occurred ch3
- 45 = Major frame occurred ch4
- 46 = Major frame occurred ch5
- 47 = Major frame occurred ch6
  
- 48 = Major frame occurred ch7
- 49 = Rx FIFO half-full ch 0
- 50 = Rx FIFO half-full ch 1
- 51 = Rx FIFO half-full ch 2
- 52 = Rx FIFO half-full ch 3
- 53 = Rx FIFO half-full ch 4
- 54 = Rx FIFO half-full ch 5
- 55 = Rx FIFO half-full ch 6
- 56 = Rx FIFO half-full ch 7
- 57 = Rx FIFO half-full ch 8
- 58 = Rx FIFO half-full ch 9
- 59 = Rx FIFO half-full ch 10
- 60 = Rx FIFO half-full ch 11
- 61 = Rx FIFO half-full ch 12
- 62 = Rx FIFO half-full ch 13
- 63 = Rx FIFO half-full ch 14
- 64 = Rx FIFO half-full ch 15
- 65 = Rx FIFO not empty ch 0
- 66 = Rx FIFO not empty ch 1
- 67 = Rx FIFO not empty ch 2

68 = Rx FIFO not empty ch 3  
69 = Rx FIFO not empty ch 4  
70 = Rx FIFO not empty ch 5  
71 = Rx FIFO not empty ch 6  
72 = Rx FIFO not empty ch 7  
73 = Rx FIFO not empty ch 8  
74 = Rx FIFO not empty ch 9  
75 = Rx FIFO not empty ch 10  
76 = Rx FIFO not empty ch 11  
77 = Rx FIFO not empty ch 12  
78 = Rx FIFO not empty ch 13  
79 = Rx FIFO not empty ch 14  
80 = Rx FIFO not empty ch 15  
81 = Command Response now  
available

EXAMPLE: A429:SIC 7, 0, 6, 19, 40

This tells the module to use the interrupt line previously designated. The module will interrupt the host for the following conditions:

- Word received on channel 0
- Word received on channel 6
- Receive error occurred on channel 3
- Serial data received

ERRORS

GENERATED: No error, Bad Parameter

RESPONSE

PARAMETERS: NONE

SEE ALSO: **Get Interrupt Source**

## 4.28 Software Version

The **Software Version** command returns the version of the module's firmware to the host. The response parameters x, y, and z make up the version number "X.Y.Z".

SCPI Command: **Arinc429:VERsion?**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

EXAMPLE: A429:VER?

This will cause the module to return the version of software running on the module.

### ERRORS

GENERATED: No error, Bad Parameter

RESPONSE	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	X	0..255	<NRf>
	Y	0..255	<NRf>
	Z	0..255	<NRf>

EXAMPLE: A429:VER? 0, 2, 3

This response informs the host that version "0.2.3" is running on the module.

## 4.29 Start Channel

The **Start Channel** command starts a channel operating after it has been halted. Note that the channels default to the halted state after they have been configured.

SCPI Command: **Arinc429:CHannel<ch#>:STart**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	NONE		

EXAMPLE: A429:CH3:ST

This will start channel 3.

### ERRORS

GENERATED: No error, Bad Parameter, Channel not defined

### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Halt Channel**

## 4.30 Transmit FIFO

The **Transmit FIFO** command will enqueue a list of Tx elements into the transmission queue.

If the SSM = 0xFF, the SSM is calculated by the module from the data for binary, BCD, BCD Raw, and lat/long data words as defined by the **Load Engineering Conversion Table** command. For UNDEF words, the label, SDI, and SSM in the command are ignored. The 32 bits of ARINC 429 Data are transmitted exactly as the host specifies. However, the parity bit is still calculated by the module as defined by the **Define Tx Channel** command.

The “Element Not Enqueued” error indicates that the module’s internal FIFO’s overflowed preventing a word from being enqueued. Each unsuccessful enqueue will generate an “Element not enqueued” error. See **Section 3.0** for further details.

SCPI Command: **Arinc429:Fifo<ch#>:TRAnsmi**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	list of		
	SSM	varies depending upon ARINC data type used	<Nrf>
	SDI	0 .. 3	<Nrf>
	Label	0 .. 255	<Nrf>
	Data	varies depending upon ARINC data type used	<STRING PROGRAM DATA>*

\* Since the data type being enqueued onto the Tx queue varies according to the ARINC data type used, **Data** is sent as a string so that the corresponding format can be sent. If the ARINC data type is **Binary** then Data could be “34.5”. If the ARINC data type is **Discrete** then Data could be “AF49”. If the ARINC data type is **alphanumeric** then Data could be “XYZ”. Refer to the **Load Engineering Conversion Table** command for further explanation as to what form the **Data** will take.

EXAMPLE: A429:FI2:TRA 255, 1, 125, “17.93”, 0, 0, 0, “12345678”

For this example label 125 has been set up as BNR format and label 0 as undefined using the **Load Engineering Conversion Table** Command.

This example Enqueues two ARINC 429 words onto the transmit queue to be transmitted. The first word has an SSM of “255” which tells the module to calculate the SSM based upon the sign of the number given. The SDI and label for this word are “1” and “125” respectively. The data being transmitted is “17.93”. The second word being transmitted is of the type UNDEF therefore the SSM, and SDI are not used and can be anything. The label is “0” and the value being placed into the “32 “ bit ARINC 429 word is “12345678” (Hexadecimal format).

### ERRORS

GENERATED: No error, Bad Parameter, Channel not defined, Element Not Enqueued

### RESPONSE

PARAMETERS: NONE

SEE ALSO: **Define Tx Channel, Transmit Scheduled, Load Engineering Conversion Table**

## 4.31 Transmit Scheduled

The **Transmit Scheduled** command will update the data for scheduled transmission. This element will then be transmitted in accordance with the schedule defined in the **Define Tx Schedule Table** command.

If the SSM = 0xFF, the SSM is calculated by the module from the data for binary, BCD, BCD Raw, and lat/long data words as defined by the **Load Engineering Conversion Table** command.

For UNDEF words, the ARINC 429 word will be transmitted exactly as placed in the “Data” portion of the command. The label and SDI will only be used to determine in which schedule frames the word should be transmitted. However, the parity bit is still calculated by the module as defined by the **Define Tx Channel** command.

Scheduled data is configured through the Define Tx Schedule Table command. When the channel is first started, and the host has not yet placed data in the scheduled data FIFO, nothing will be transmitted, even though the schedule has been defined. As the host places data words in the scheduled data FIFO, those words will begin to transmit as scheduled data.

The “Element Not Enqueued” error indicates that the module’s internal FIFO’s overflowed preventing a word from being enqueued. Each unsuccessful enqueue will generate an “Element not enqueued” error. See **Section 4.7** for further details.

SCPI Command: **Arinc429:Fifo<ch#>:TScheduled**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	list of		
	Valid	0 = Valid, 1=Not Valid *	<Nrf>
	SSM	varies depending upon ARINC data type used	<Nrf>
	SDI	0 .. 3	<Nrf>
	Label	0 .. 255	<Nrf>
	Data	varies depending upon ARINC data type used	<STRING PROGRAM DATA>**

\* This flag is used to turn on and off this scheduled label. If a 1 is placed in this field then no data with this Label/SDI combination will be sent until this flag is cleared.

\*\* Since the data type being enqueued onto the Tx queue varies according to the ARINC data type used, **Data** is sent as a string so that the corresponding format can be sent. If the ARINC data type is **Binary** then Data could be “34.5”. If the ARINC data type is **Discrete** then Data could be “AF49”. If the ARINC data type is **alphanumeric** then Data could be “XYZ”. Refer to the **Load Engineering Conversion Table** command for further explanation as to what form the **Data** will take.

EXAMPLE: A429:FI7:TS 0, 255, 0, 3, “105.7”

For this example, label 3 has been set up as a BCD word having 4 significant digits and a resolution of 0.1.

This example enqueues one element into the transmission schedule queue. This element is “valid”, (i.e. to be transmitted). The SSM is “255” so the SSM will be calculated based on the data given. The SDI and label values are “0” and “3” respectively. The data is “105.7”.

ERRORS

GENERATED: No error, Bad Parameter, Channel not defined, Element Not Enqueued

RESPONSE

PARAMETERS: NONE

SEE ALSO: **Define Tx Channel, Transmit FIFO, Load Engineering Conversion Table , Receive FIFO**

## **4.32 Write Output Discrete**

The **Write Output Discrete** command sets the value for all three of the host controllable output discretets.

There are a total of eight output discretets. Three of these discretets are host-controllable (D6, D7, and D8). The remaining five are used to control lights on the module's front panel (D5=FAIL Light, D1..D4=LED 0..3). All eight-output discretets are available on the front panel connector.

SCPI Command: **Arinc429:WODiscrete**

HOST	<u>parameter</u>	<u>valid values</u>	<u>Type</u>
PARAMETERS:	discrete out 6	0=OFF, 1=ON	<NRf>
	discrete out 7	0=OFF, 1=ON	<NRf>
	discrete out 8	0=OFF, 1=ON	<NRf>

EXAMPLE: A429:WOD 1, 0, 1

This turns output discretets 6 and 8 on, and 7 off.

ERRORS

GENERATED: No error, Bad Parameter

RESPONSE

PARAMETERS: NONE

SEE ALSO: **Read Input Discrete**



# Chapter 5

## System Level Example

This example shows how to define and use a single transmitter and receiver channel on the DDC-429 module. For purposes of this example, it is assumed that transmitter channel 0 is connected to receiver channel 1 on the module's external connector.

The following table contains the commands and their respective responses that the host must communicate with the module in order to configure and run the channels correctly.

Command/Response Name	Command sent by host / Response sent by module	Explanation of command
COMMAND - Reset Card	A429:RES MASTER	This will reset the card.
RESPONSE - Reset Card	A429:RES #0	Card reset with no errors detected.
COMMAND - Define Tx Channel	A429:CH0:DTC BOTH,32,0,ODD,-1,HIGH,0,1024,1024	This sets up channel 0 to be a transmitter.
COMMAND - Define Rx Channel	A429:CH1:DRC FIFO,32,0,ODD,HIGH,0,ON,ON,OFF,OFF,1024	This sets up channel 1 to be a receiver.
COMMAND - Load Engineering Conversion Table	A429:CH0:LECT 24,BCD,4,.01,25,BNR,18,.1,26,BCD,5,.001	This defines label 24 for channel 0 to be a BCD word, label 25 to be a BNR word, and label 26 to be a BCD word.
COMMAND - Load Engineering Conversion Table	A429:CH1:LECT 24,BCD,4,.01,25,BNR,18,.1,26,BCD,5,.001	This repeats the same engineering unit conversions for channel 1.
COMMAND - Define Rx Filter Words	A429:CH1:DRFW 24,1,ON,25,2,ON,26,0,ON,27,1,ON	This specifies that channel 1 will receive only the four-label/SDI combinations: 24/1, 25/2, 26/1, and 27/1.
COMMAND - Set Interrupt Conditions	A429:SIC 4,1,32	This will cause the card to send an interrupt to the host via VME interrupt line 4 when a word is received on channel 1 and when the transmit FIFO is empty for channel 0.
COMMAND - Define Rx Interrupt Filter	A429:CH1:DIFW 24,1,ON	This specifies that channel 1 will only generate an interrupt when it receives label 24 / SDI 1. All other received words will not generate an interrupt.

COMMAND - Define Tx Schedule Table	A429:CH0:DTST 10,10,10000,1,24,1,5,25,2	This defines a transmit schedule table. Label 24 / SDI 1 will be transmitted every 10 seconds and five seconds later label 25 / SDI 2 will also be transmitted every 10 seconds.
COMMAND - Start Channel	A429:CH0:ST	Channel 0 is started. (Nothing is being transmitted yet because no transmit scheduled or FIFO data has been loaded).
COMMAND - Start Channel	A429:CH1:ST	Channel 1 is started.
COMMAND - Transmit FIFO	A429:FI0:TRA 255,0,26,"12.345",255,1,27, "12345678"	This will cause two words to be transmitted. The first word will be a BNR and the second a BCD as defined earlier.
COMMAND - Receive FIFO	A429:FI1:REC? 2	Two elements will be dequeued from the receive FIFO and sent back to the host.
RESPONSE - Receive FIFO	A429:FI1:REC? 0,#H06,#H00,#H1A, "12.345",#H002DC6C0,0, #H00,#H02,#H1B, "12345678",#H002DC6C4	The two requested ARINC words received are sent to the host. The label, SDI, SSM, time stamp, and data are all listed separately.
COMMAND - Transmit FIFO	A429:FI0:TRA 255,0,28,"18765432",255,1,29,"1234 5678"	Two UNDEF raw words will be transmitted.
COMMAND - Receive FIFO	A429:FI1:REC? 2	Request that two words from the receive FIFO be dequeued. (In this case the queue is empty because the words did not pass the receive filter).
VME interrupt line asserted		An interrupt will be sent to the host via VME interrupt line 7 telling the host that an error has occurred.
COMMAND - Error message request	SYST:ERR?	This command is requesting that the error message for the last error be sent to the host.
RESPONSE - Error message request	21,"Dequeue error:A429:FI1:REC?"	This indicates a dequeue was attempted on an empty queue.
COMMAND - Transmit Scheduled	A429:FI0:TS 0,255,1,24,"12.34",0,255,0,25,"- 2345.9"	Label 24 and 25 will be repeatedly transmitted every ten seconds. They will be spaced apart every five seconds.

# Appendix A

---

## Customer Contact Report

Date: \_\_\_\_\_

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Phone: \_\_\_\_\_

Fax: \_\_\_\_\_

e mail: \_\_\_\_\_

Product/Model Number: \_\_\_\_\_

Software Version: \_\_\_\_\_

Hardware Serial Number: \_\_\_\_\_

Briefly Describe the problem you are experiencing:

---

---

---

---

---

---

---

Please call Customer Support at (800) DDC-5757 between 8:30 - 5:00 Eastern Time, FAX this form to (516) 567-7358 or contact us through our website at: [www.ddc-web.com](http://www.ddc-web.com)

# Appendix B

---

## Acronyms

The following are some acronyms and terms often used in this manual:

A429	ARINC 429
ADC	Analog to Digital Converter
BCD	Binary Code Decimal
BNR	Binary
ch	Channel
CSP	Configure Serial Port
DAC	Digital to Analog Converter
DER	Dequeue Errors
DIR	Data In Ready
DOR	Data Out Ready
DRC	Define Rx Channel
DTC	Define Tx Channel
DTST	Define Tx Schedule Table
ERR	Error message request
EXT	Extension
FI	FIFO
FIFO	First-In First-Out
GIS	Get Interrupt Source
HA	Halt Channel
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
K	Kilo
KByte	Kilo-Byte
KHz	Kilo-Hertz
LECT	Load Engineering Conversion Table
LED	Light Emitting Diodes
MHz	Mega-Hertz
MRC	Modify Rx Channel
ms	Millisecond
MTC	Modify Tx Channel

POST	Power On Self Test
RAM	Random Access Memory
RC	Reset Clock
REC	Receive FIFO
RES	Reset Card
RFW	Define Rx Filter Words
RID	Read Input Discrete
RM	Read Mailbox
ROM	Read Only Memory
Rx	Receive
S/W	Software
ST	Start Channel
SCPI	Standard Commands for Programmable Instruments
SDI	Source/Destination Indicator
SIC	Set Interrupt Conditions
SR	Serial Receive
SRAM	Static Random Access Memory
SSM	Sign/Status Matrix
STR	Serial Transmit
STX	Start of Text
SYST	System
TRA	Transmit FIFO
TS	Transmit Scheduled
Tx	Transmit
UNDEF	Undefined
VER	Software Version
VXI	VMEbus Extensions for Instrumentation
WOD	Write Output Discrete
μs	Microsecond

# Index

<b>C</b>		<b>Serial Transmit</b> .....9, 34
Command Definitions .....	17	<b>Set Interrupt Conditions</b> ..... 7, 19, 24, 35
<b>Configure Serial Port</b> .....	9, 14, 17	<b>Software Version</b> .....
<b>D</b>		10, 38
<b>Define Rx Channel</b> 8, 14, 17, 26, 27, 28, 29,		<b>Start Channel</b> ..... 7, 8, 17, 20, 26, 38
30, 31		System Level Example .....
<b>Define Rx Filter Words</b> .....	9, 14, 19	43
Define Rx Interrupt Filter .....	19	<b>T</b>
<b>Define Tx Channel</b> .7, 14, 20, 26, 27, 38, 39		Timers and Timestamps .....
<b>Define Tx Schedule Table</b> ..... 7, 14, 20, 21		9
<b>Dequeue Errors</b> .....	10, 23	Transmission Schedules .....
<b>E</b>		7
Error .....	2	<b>Transmit FIFO</b> .....
Error Handling .....	10	23, 30, 31, 38, 40
<b>Error Message Request</b> .....	23	<b>Transmit Scheduled</b> ..... 21, 23, 30, 31, 39
<b>F</b>		<b>Transmitting</b> .....
Filtering .....	2	1, 7
<b>G</b>		<b>U</b>
Generating Interrupts.....	9	updates .....
<b>Get Interrupt Source</b> .....	24, 35	1
<b>H</b>		<b>W</b>
<b>Halt Channel</b> .....	26, 38	Warranty .....
<b>I</b>		1
Initialization.....	3, 6	<b>Write Output Discrete</b> .....
<b>L</b>		9, 29, 40
license .....	1	
<b>Load Engineering Conversion Table</b> 5, 6,		
11, 26, 27, 30, 31, 38, 39, 40, 43, 46		
<b>M</b>		
<b>Modify Rx Channel</b> .....	9, 18, 28	
<b>Modify Tx Channel</b> .....	7, 20, 28	
Module Communication .....	11	
<b>R</b>		
<b>Read Input Discrete</b> .....	29, 41	
<b>Read Input Discrete</b> .....	9	
<b>Read Mailbox</b> .....	29	
<b>Receive FIFO</b> .....	30, 40	
<b>Receiving</b> .....	2, 8	
<b>Reset Card</b> .....	9, 31	
<b>Reset Clock</b> .....	9, 31, 33	
<b>S</b>		
Serial Port Control .....	9	
<b>Serial Receive</b> .....	9, 33	



## Artisan Technology Group is your source for quality new and certified-used/pre-owned equipment

- FAST SHIPPING AND DELIVERY
- TENS OF THOUSANDS OF IN-STOCK ITEMS
- EQUIPMENT DEMOS
- HUNDREDS OF MANUFACTURERS SUPPORTED
- LEASING/MONTHLY RENTALS
- ITAR CERTIFIED SECURE ASSET SOLUTIONS

### SERVICE CENTER REPAIRS

Experienced engineers and technicians on staff at our full-service, in-house repair center

### *InstraView*<sup>SM</sup> REMOTE INSPECTION

Remotely inspect equipment before purchasing with our interactive website at [www.instraview.com](http://www.instraview.com) ↗

### WE BUY USED EQUIPMENT

Sell your excess, underutilized, and idle used equipment. We also offer credit for buy-backs and trade-ins. [www.artisanng.com/WeBuyEquipment](http://www.artisanng.com/WeBuyEquipment) ↗

### LOOKING FOR MORE INFORMATION?

Visit us on the web at [www.artisanng.com](http://www.artisanng.com) ↗ for more information on price quotations, drivers, technical specifications, manuals, and documentation

**Contact us:** (888) 88-SOURCE | [sales@artisanng.com](mailto:sales@artisanng.com) | [www.artisanng.com](http://www.artisanng.com)