

IP-ARINC429

OCTAL ARINC429 Industry Pack Module REFERENCE MANUAL

Revision 1.2
September 2009

ALPHI TECHNOLOGY CORP.
1898 E. Southern Ave
Tempe, AZ 85282 USA
Tel : (480) 838 - 2428

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual or from the use of information contain herein.

ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY's products to improve reliability, performance, function or design. ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

ALPHI TECHNOLOGY CORPORATION
All Rights Reserved

This document shall not be duplicated, nor its contents used for any purpose, unless express permission has been granted in advance.

TABLE OF CONTENTS

GENERAL DESCRIPTION	5
INTRODUCTION	5
FUNCTIONAL DESCRIPTION	6
INTERFACE TO THE IP CARRIER	7
IPBUS IDSPACE	7
IPBUS INTSPACE	7
IPBUS IOSPACE	7
BOARD CONTROL REGISTER	7
ENTX STATE 8	
IPBUS MEMSPACE	8
FIFO STATUS REGISTER	9
CONTROL REGISTER	10
RECEIVER 1, RECEIVER 2, TRANSMITTER FIFO USAGE	10
INTERVAL REGISTER	10
BURST COUNT 11	
CONTINUOUS TRANSMIT REGISTER	11
SOFTWARE DETAILS	12
LIBRARY FUNCTIONS	12
ARINC429_FINDNOOFBOARDS	12
ARINC429_OPEN	12
ARINC429_RESET	12
ARINC429_CLOSE	12
ARINC429_GETDRIVERVERSION	12
ARINC429_GETERRORMSG	13
ARINC429_CLEARTRXFIFO	13
ARINC429_ENABLEDISABLESELFTTEST	14
ARINC429_CONFIGURETX	14
ARINC429_COMPOSEARINCWORD	14
ARINC429_READTXSTATUS	15
ARINC429_READTXFIFOSTATUS	16
ARINC429_CONFIGURECONTINUOUSMODE	16
ARINC429_LOADCONTINUOUSMODEDATA	16
ARINC429_STARTCONTINUOUSMODE	16
ARINC429_CONFIGUREBURSTMODE	17
ARINC429_STARTBURSTMODE	17
ARINC429_STOPBURSTORCONTINUOUSMODE	17
ARINC429_CONFIGURERX	18
ARINC429_READRXFIFOSTATUS	18
ARINC429_READRXSTATUS	18
ARINC429_READARINCWORD	19
ARINC429_SETLABELS	19
ARINC429_READLABELMATCH	19

ARINC429_INTENABLEDISBLE	20
ARINC429_ENABLEDISABLETXINTERRUPT	20
ARINC429_READTXINTERRUPTSTATUS	21
ARINC429_ENABLEDISABLERXINTERRUPT	21
ARINC429_READRXINTERRUPTSTATUS	22
ARINC429_GETSEMID	22
ARINC429_READINTERRUPTDATA	22
STRUCTURES	23
TRANSMITTER OR RECEIVER CONFIGURATION	23
ARINC429 WORD	23
BUFFER	23
MISCELLANEOUS	23
<i>HARDWARE DETAILS</i>	24
JUMPERS	24
CONNECTORS	24
CONNECTOR DESCRIPTIONS	25
IP EXTERNAL I/O CONNECTOR	25

GENERAL DESCRIPTION

INTRODUCTION

The **IP_ARINC429** is a high density IP module (Industry Pack-Compatible) interface for Avionics protocols that provides up to eight pairs of fully independent ARINC 429 channels.

The logic on the **IP_ARINC429** is designed to meet the ARINC 429 specifications for loading level detection, timing and protocol.

ARINC429 (Aeronautical Radio Incorporated) is a specification, which defines how avionics equipment and system should communicate with each other. ARINC 429 employs a unidirectional data bus standard known as Mark 33 Digital Information Transfer System (DITS). Messages are transmitted and received at a bit rate of either 12.5 kbits/second or 100 kbits/second

FEATURES

- 8 independent Receive and Transmit channels
- Label matching for all receiver channels
- Two 256 kWords SRAMs buffers for storing data
- Software selected data rate of 12.5 kbits/second or 100 kbits/second with automatic slew rate adjustment
- Programmable word length selection, with the parity bit generated automatically
- Serial to parallel conversion
- Automatic word gap timer
- Burst and Continuous Mode for transmitting data at the rate of 100 msec to 1200 msec
- Highly compact credit-card sized Industry-Pack module
- 8/32 MHz IP clock.

A data flow block diagram of the **IP_ARINC429** is presented in Figure 1-1.

Figure 1.1: Data Flow Block Diagram

FUNCTIONAL DESCRIPTION

INTERFACE TO THE IP CARRIER

IPBUS IDSPACE

Up to 32 bytes of registered data provide information about the module. Although 16-bit access is used, only the lower byte contains the data. As provided in the IP-module specification, it contains manufacturer data, the type of module, revision, etc...

ID space address	Description	Value
\$00	ASCII "I"	\$49
\$02	ASCII "P"	\$50
\$04	ASCII "A"	\$41
\$06	ASCII "H"	\$48
\$08	Manufacturer identification	\$11
\$0A	Module type	\$33
\$0C	Revision module	\$0A
\$0E	Reserved	\$00
\$10	Driver ID, low byte	\$00
\$12	Driver ID, high byte	\$00
\$14	Number of bytes used	\$0A
\$16	CRC	\$00
\$18-\$3E	User space	\$00

Table 2-1 IDSEL0 SPACE byte content

IPBUS INTSPACE

During an interrupt acknowledge, the interrupt vector is sent to the processor.

IPBUS IOSPACE

The IOspace is used mostly for diagnostics registers.

IP-ARINC429 Address	Read/Write	Access	Register
0x00	R	8-bit	Board Control
0x02	R	8-bit	ENTX state (Read only)
0x04	R	8-bit	DBC enable state
0x06	R	8-bit	Slope state
0x08	R	8-bit	Transmit Ready
0x0a	R	8-bit	Receiver 1 Ready
0x0c	R	8-bit	Receiver 2 Ready
0x0e	R	8-bit	Version Number

Table 2.2: IO Registers

BOARD CONTROL REGISTER

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used		Not Used	Not Used	Not Used	IRQ1	IRQ0	Global Int Enable	IPclk Speed

IPclk Speed is 0 if the IP clock is at 32 MHz.

Global Interrupt Enable (GIE) is 1 when the interrupts are enabled.

IRQ0 shows the state of the interrupt requests on the lowest 4 blocks (Tx1 to TX4, Rx1 to Rx4, Rx9 to Rx12). A state of "0" means that the request is active.

IRQ1 shows the state of the interrupt requests on the highest 4 blocks (Tx5 to TX8, Rx5 to Rx8, Rx13 to Rx16). A state of "0" means that the request is active.

ENTX STATE

This register is only used for debug.

Each bit represents the state of the ENTX line going to the corresponding HOLT 3282, the state is 1 if the transmitter is enabled. The bit 0 corresponds to TX1 and so on, until the bit 7 corresponding to TX8.

These bits are set by enabling the transmitter in the control block in IP mem.

IPBUS MEMSPACE

The transmitters and receivers on the boards are split in 8 groups of 1 transmitter and 2 receivers. Some of the programming of the 2 receivers in a group is common and for instance they must have the same reception speed.

IP-ARINC429 Address	Register
0x000-0x07e	Transmit/Receive Group #1
0x080-0x0fe	Transmit/Receive Group #2
0x100-0x17e	Transmit/Receive Group #3
0x180-0x1fe	Transmit/Receive Group #4
0x200-0x27e	Transmit/Receive Group #5
0x280-0x2fe	Transmit/Receive Group #6
0x300-0x37e	Transmit/Receive Group #7
0x380-0x3fe	Transmit/Receive Group #8

Table 2.3: MEMSpace addresses

IP-ARINC429 Address	Read/Write	Access	Register
0x00	R	16-bit	FIFO Status Register
0x02	R/W	16-bit	Control Register
0x04	R	16-bit	Receiver 1 FIFO lower word
0x04	W	16-bit	Transmitter FIFO lower word
0x06	R	16-bit	Receiver 1 FIFO upper word
0x06	W	16-bit	Transmitter FIFO upper word
0x08	R	16-bit	Receiver 2 FIFO lower word
0x0a	R	16-bit	Receiver 2 FIFO upper word
0x0c	R/W	16-bit	Continuous Transmit lower word
0x0e	R/W	16-bit	Continuous Transmit upper word
0x10	R/W	16-bit	Interrupt Control Register
0x12	W	16-bit	Transmitter lower word
0x14	W	16-bit	Transmitter upper word
0x16	R	16-bit	Receiver1 FIFO pointer
0x18	R	16-bit	Receiver2 FIFO pointer

0x1a	R	16-bit	Transmitter FIFO pointer
0x1c	R/W	16-bit	Interval Register
0x1e	R/W	16-bit	Burst Count
0x20-0x3e	R/W	16-bit	Label programming Receiver 1
0x40-0x5e	R/W	16-bit	Label programming Receiver 2

Table 2.4: Group Registers

FIFO STATUS REGISTER

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Recv2 FIFO empty	Recv2 FIFO half full	Recv2 FIFO full	Recv1 FIFO empty	Recv1 FIFO half full	Recv1 FIFO full	Transmit FIFO empty	Transmit FIFO half full	Transmit FIFO full

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
Continuous Mode	Slope	DBCEN	Enable Transmission	Clear Transmit FIFO	Clear Receiver 1 FIFO	Clear Receiver 2 FIFO

- **Transmit FIFO Full:**
 - 0: Not full.
 - 1: Full.
- **Transmit FIFO Half Full:**
 - 0: less than half.
 - 1: more than half full.
- **Transmit FIFO Empty:**
 - 0: Not empty
 - 1: Empty
- **Receiver 1 FIFO Full:**
 - 0: Not full.
 - 1: Full.
- **Receiver 1 FIFO Half Full:**
 - 0: less than half.
 - 1: more than half full.
- **Receiver 1 FIFO Empty:**
 - 0: Not empty
 - 1: Empty
- **Receiver 2 FIFO Full:**
 - 0: Not full.
 - 1: Full.
- **Receiver 2 FIFO Half Full:**
 - 0: less than half.
 - 1: more than half full.
- **Receiver 2 FIFO Empty:**
 - 0: Not empty
 - 1: Empty
- **Continuous Mode:** This clock select is used by both receivers in the group so that they need to be programmed with the same speed.
 - 0: Disabled
 - 1: Enabled
- **Slope:** This bit selects the edge speed of the ARINC429 signal. The normal edge speed depends on the transmission rate.

- 0: 12.5 kbit/sec transmission rate
- 1: 100 kbit/sec transmission rate
- **Enable Transmission:**
 - 0: Disabled
 - 1: Enabled
- **Clear FIFO:** These 3 bits control the FIFO reset mechanism. To do the reset, put the bit to 0, then put it back to 1. The FIFO stays disabled as long as the bit stays at 0.
 - 0: FIFO reset and disabled
 - 1: normal functioning

CONTROL REGISTER

Bit 14	Bit 13	Bit 12	Bit 11-10	Bit 9	Bit 8-7	Bit 6	Bit 5	Bit 4
Receiver Data Clock Select	Transmit Data Clock Select	Transmitter Parity	Recv2 SDI value	Recv2 SDI enable	Recv2 SDI value	Recv2 SDI enable	Self-Test	Parity Enable

- **Receiver Data Clock Select:** This clock select is used by both receivers in the group so that they need to be programmed with the same speed.
 - 0 for a 100 kbit/sec transmission rate
 - 1 for a 12.5 kbit/sec transmission rate
- **Transmitter Data Clock Select:** This clock select is used by both receivers in the group so that they need to be programmed with the same speed.
 - 0 for a 100 kbit/sec transmission rate
 - 1 for a 12.5 kbit/sec transmission rate
- **Parity Enable:**
 - 0: Parity disabled
 - 1: Parity enabled
- **Transmitter Parity:**
 - 0: Odd parity (normal)
 - 1: Even parity
- **Self Test:** This clock select is used by both receivers in the group so that they need to be programmed with the same speed.
 - 0: Self Test enabled
 - 1: Normal (Self Test disabled)
- **Receiver 1 SDI, Receiver 2 SDI:** When this mode is enabled, only the words with the proper SDI pattern are received and copied to the FIFO.
 - 0: SDI test disabled
 - 1: SDI test enabled

RECEIVER 1, RECEIVER 2, TRANSMITTER FIFO USAGE

These registers indicates how much of the FIFO words are used. The number is between 0 (FIFO empty) and 2047 (FIFO full).

INTERVAL REGISTER

This register can be used to select a delay between the word transmission. The delay is indicated in milliseconds between the beginning of transmission of two consecutive words. A value of 0 means that there is no delay and the words are transmitted sequentially. The maximum delay between word is 2047 milliseconds.

BURST COUNT

When read, this register indicates how many words are still going to be sent in burst mode. When written, the maximum value is 63.

It can be overwritten even if a burst transfer is currently taking place. This can be used to prolongate the burst.

CONTINUOUS TRANSMIT REGISTER

This register contains the value sent during continuous or burst mode. Changing it changes immediately what is being sent, although it does not change whatever value is already in the FIFO.

INTERRUPT CONTROL REGISTER

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Recv2 FIFO empty	Recv2 FIFO half full	Recv2 FIFO full	Recv1 FIFO empty	Recv1 FIFO half full	Recv1 FIFO full	Transmit FIFO empty	Transmit FIFO half full	Transmit FIFO full

- **Transmit FIFO Full Interrupt:**
Interrupts when the FIFO is **not full**.
 - 0: Not enabled.
 - 1: Enabled.
- **Transmit FIFO Half Full Interrupt:**
Interrupts when the FIFO is **less than half-full**.
 - 0: Not enabled.
 - 1: Enabled.
- **Transmit FIFO Empty Interrupt:**
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 1 FIFO Full Interrupt:**
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 1 FIFO Half Full Interrupt:**
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 1 FIFO Empty Interrupt:**
Interrupts when the FIFO is **not empty**.
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 2 FIFO Full Interrupt:**
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 2 FIFO Half Full Interrupt:**
 - 0: Not enabled.
 - 1: Enabled.
- **Receiver 2 FIFO Empty Interrupt:**
Interrupts when the FIFO is **not empty**.
 - 0: Not enabled.
 - 1: Enabled.

SOFTWARE DETAILS

LIBRARY FUNCTIONS

ARINC429_FINDNOOFBOARDS

Arinc429Status Arinc429_FindNoOfBoards (UINT16 *nbrOfBoards)

This function detects the number of ARINC429 cards present in the chassis

Error Codes

ARINC429_INTERNAL_ERROR

ARINC429_OPEN

```
Arinc429Status Arinc429_Open (  
    UINT16 brdNbr,  
    ARINC429handle *ARINC429  
)
```

This function initializes the board control structure and returns a handle to it.

Error Codes

ARINC429_INTERNAL_ERROR

ARINC429_INVALID_BOARD_NUM

ARINC429_RESET

```
Arinc429Status ARINC429_Reset (  
    ARINC429handle ARINC429  
)
```

This function resets the board.

Error Codes

ARINC429_INVALID_HANDLE

ARINC429_CLOSE

```
Arinc429Status ARINC429_Close(  
    ARINC429handle ARINC429  
)
```

This function frees the resources for the specified board and releases the handle.

Error Codes

ARINC429_INVALID_HANDLE

ARINC429_GETDRIVERVERSION

```
void ARINC429_GetDriverVersion (  
    unsigned long *DriverVersion  
)
```

This function returns a version number for the driver at the address pointed to by the DriverVersion field.

ARINC429_GETERRORMSG

```
void ARINC429_GetErrorMsg (  
Arinc429Status errCode,  
char **ErrMsg  
)
```

This function returns a string corresponding to the error code.

Error Codes as input

```
ARINC429_INTERNAL_ERROR  
ARINC429_INVALID_BOARD_NUM  
ARINC429_BOARD_NOT_PRESENT  
ARINC429_INVALID_HANDLE  
ARINC429_INVALID_CHANNEL_NUM  
ARINC429_INVALID_SELF_TEST_ENABLE_VAL  
ARINC429_INVALID_VALUE  
ARINC429_INPUT_MODE  
ARINC429_OUTPUT_MODE  
ARINC429_INVALID_LOGIC_SEL  
ARINC429_INVALID_STROBE_MODE  
ARINC429_INVALID_GROUP  
ARINC429_INVALID_FREQUENCY  
ARINC429_INVALID_INPUT_MODE  
ARINC429_INVALID_MASK_VALUE  
ARINC429_INVALID_MODE  
ARINC429_INVALID_SELF_TEST_DATA  
ARINC429_SELF_TEST_DISABLED  
ARINC429_FAILED_SELF_TEST  
ARINC429_INVALID_TIME_BOUNCE_VAL  
ARINC429_INT_ALREADY_ENABLED  
ARINC429_INT_NOT_ENABLED  
ARINC429_INVALID_TRANRCVSTS  
ARINC429_INVALID_TRANSNUMBER  
ARINC429_INVALID_  
ARINC429_INVALID_RECVSTATUS  
ARINC429_INVALID_TRPAIR  
ARINC429_INVALID_TRANSSTATUS  
ARINC429_INVALID_LABELNUMBER  
ARINC429_INVALID_SDI  
ARINC429_INVALID_PARITY  
ARINC429_INVALID_DATARATE  
ARINC429_INVALID_DATALENGTH  
ARINC429_INVALID_SSM  
ARINC429_INVALID_EMTINTSTS  
ARINC429_INVALID_HFINTSTS  
ARINC429_INVALID_FINTSTS
```

ARINC429_CLEARTXRXFIFO

```
Arinc429Status ARINC429_ClearTxRxFIFO (  
ARINC429handle brd,  
UINT16 fifo,  
UINT16 channel,  
UINT16 ClearSts);
```

The function is used to clear either the receive or transmit FIFO. The fifo value is 0 for the receive FIFO and 1 for the transmit FIFO. The channel number is a value between 1 and 8. The ClearSts is 0 for "Clear", and 1 for "Unclear".

Error Codes

ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANRECVSTS
ARINC429_INVALID_TRANSNUMBER
ARINC429_INVALID_RECVNUMBER

ARINC429_ENABLEDISABLESELFTEST

```
Arinc429Status Arinc429_EnableDisableSelfTest(  
ARINC429handle brd,  
UINT16 channel,  
UINT16 SelfTestStatus  
)
```

The function is used to enable / disable a self-test signal.

The channel number is a value between 1 and 8. The SelfTestStatus specifies the connection of self-test signal from the transmitter X directly to receiver X.

0 – Connects the self test signal
1 – Disconnects the self test signal

.

Error Codes :

ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRPAIR
ARINC429_INVALID_SLFTST_STATUS

ARINC429_CONFIGURETX

```
Arinc429Status Arinc429_ConfigureTx (  
ARINC429handle brd,  
struct TxorRxConfiguration txConfig)
```

The function is used to configure the transmitter channels.

.

Error Codes

ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER
ARINC429_INVALID_TRANSSTATUS
ARINC429_INVALID_LABELNUMBER
ARINC429_INVALID_SDI
ARINC429_INVALID_PARITY
ARINC429_INVALID_DATARATE
ARINC429_INVALID_DATALENGTH

ARINC429_COMPOSEARINCWORD

```
Arinc429Status ARINC429_ComposeArincWord(  
    ARINC429handle brdHandle,  
    UINT16 channel,  
    UINT32 arincData  
)
```

Each ARINC word is a 32-bit value that contains five fields:

- Bit 32 is the parity bit, and is used to verify that the word was not damaged or garbled during transmission.
- Bits 30 to 31 is the Sign/Status Matrix, or SSM, and often indicates whether the data in the word is valid.

The SSM can also indicate the Sign (+/-) of the data or some information related to it like an orientation (North/South/East/West).

- Bits 11 to 29 contain the data. Bit-field, Binary Coded Decimal (BCD), and two's complement binary encoding (BNR) are common ARINC 429 data formats. Data formats can also be mixed.
- Bits 9 and 10 are Source/Destination Identifiers (SDI) and indicate for which receiver the data is intended or more frequently which subsystem transmitted the data.
- Bits 1 to 8 contain a label (label words), expressed in octal, identifying the data type.

Using a 25 bit there are only 2 fields:

- Bits 9 to 24 contain the data.
- Bits 1 to 8 contain a label (label words), expressed in octal, identifying the data type.

Parameters:

- brdHandle : the handle to the board.
- channel : transmitter channel number (1 to 8).
- arincData : Arinc data in 19-bit or 16-bit depending on word length selection.

Error Codes:

ARINC429_INVALID_TRANSNUMBER
ARINC429_INVALID_HANDLE

ARINC429_READTXSTATUS

```
Arinc429Status ARINC429_ReadTxStatus (
    ARINC429handle brdHandle,
    UINT16 channel,
    UINT16 *DataRdySts,
    UINT16 *EmptySts,
    UINT16 *HFullSts,
    UINT16 *FullSts
)
```

The function reads data ready status, empty status, half-full status and full status of a transmitter.

Parameters:

- brdHandle : Handle to the board.
- channel : transmitter channel number (1-8).
- DataRdySts : transmitter data ready.
 - 0 – The transmitter is ready to transmit
 - 1 – not ready
- EmptySts : Empty FIFO status.
 - 0 – The TX FIFO contains at least a word
 - 1 – The TX FIFO is empty
- HFullSts : Half-full interrupt status.
 - 0 – The TX FIFO is less than half full
 - 1 – The TX FIFO is more than half full
- FullSts : Full interrupt status.
 - 0 – The TX FIFO is less than full
 - 1 – The TX FIFO is full

Error Codes:
ARINC429_INVALID_TRANSNUMBER
ARINC429_INVALID_HANDLE

ARINC429_READTXFIFOSTATUS

```
Arinc429Status ARINC429_ReadTxFIFOStatus(  
    ARINC429handle brdHandle,  
    UINT16 channel,  
    UINT16 *wordNbr  
)
```

The function returns the number of ARINC words in the receive buffer. The channel number is a value between 1 and 8.

Error Codes :
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER

ARINC429_CONFIGURECONTINUOUSMODE

```
Arinc429Status Arinc429_ConfigureContinuousMode (  
ARINC429handle brd,  
UINT16 channel,  
UINT16 timeInterval  
)
```

The function allows specifying the time interval in the continuous mode of a transmitter channel. The channel number is a value between 1 and 8. timeInterval is expressed in millisecond and is a value between 100 and 1200.

Error Codes :
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER

ARINC429_LOADCONTINUOUSMODEDATA

```
Arinc429Status Arinc429_LoadContinuousModeData (  
ARINC429handle brd,  
UINT16 channel,  
UINT32 arincData  
)
```

The function specifies the data to be sent in the continuous mode of a transmitter channel. The channel number is a value between 1 and 8. arincData is a 19-bit or 16-bit data depending if the transmitter is in 25 or 32-bit mode.

Error Codes
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER

ARINC429_STARTCONTINUOUSMODE

```
Arinc429Status Arinc429_StartContinuousMode (  
ARINC429handle brd,
```


UINT16 channel
)

The function starts a continuous mode transmission on the selected channel. The channel number is a value between 1 and 8.

Error Codes
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSMISSION_NUMBER

ARINC429_CONFIGUREBURSTMODE

Arinc429Status Arinc429_ConfigureBurstMode (
ARINC429handle brd,
UINT16 channel,
UINT16 timeInterval,
UINT32 arincData
)

The function configures a burst mode transmission on the selected channel. The channel number is a value between 1 and 8. timeInterval is expressed in millisecond and is a value between 100 and 1200. arincData is a 19-bit or 16-bit data depending if the transmitter is in 25 or 32-bit mode.

Error Codes
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSMISSION_NUMBER

ARINC429_STARTBURSTMODE

Arinc429Status Arinc429_StartBurstMode (
ARINC429handle brd,
UINT16 channel,
UINT16 repeatCount
)

The function starts a continuous mode transmission on the selected channel. The channel number is a value between 1 and 8. repeatCount specifies the number of times the data will be sent. It is a number between 1 and 100.

Error Codes
ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSMISSION_NUMBER

ARINC429_STOPBURSTORCONTINUOUSMODE

Arinc429Status Arinc429_StopBurstorContinuousMode (
ARINC429handle brd,
UINT16 channel
)

The function stops the transmission on the selected channel. The channel number is a value between 1 and 8.

ARINC429_CONFIGURERX

```
Arinc429Status Arinc429_ConfigureRx (  
ARINC429handle brd,  
struct TxorRxConfiguration rxConfig)
```

The function is used to configure the receiver channels.

Error Codes

```
ARINC429_INVALID_HANDLE  
ARINC429_INVALID_RECVNUMBER  
ARINC429_INVALID_RECVSTATUS  
ARINC429_INVALID_LABELNUMBER  
ARINC429_INVALID_SDI  
ARINC429_INVALID_PARITY  
ARINC429_INVALID_DATARATE  
ARINC429_INVALID_DATALENGTH
```

ARINC429_READRXFIFOSTATUS

```
Arinc429Status Arinc429_ReadRxFIFOStatus (  
ARINC429handle brd,  
UINT16 channel,  
UINT32 *wordNbr  
)
```

The function returns the number of ARINC words in the receive buffer. The channel number is a value between 1 and 8.

Error Codes

```
ARINC429_INVALID_HANDLE  
ARINC429_INVALID_TRANNUMBER
```

ARINC429_READRXSTATUS

```
Arinc429Status ARINC429_ReadRxStatus (  
ARINC429handle brdHandle,  
UINT16 channel,  
UINT16 *DataRdySts,  
UINT16 *EmptySts,  
UINT16 *HFullSts,  
UINT16 *FullSts  
)
```

The function reads data ready status, empty status, half-full status and full status of a receiver.

Parameters:

brdHandle : Handle to the board.

channel : Receiver channel number (1-16).

DataRdySts : Receiver data ready.

0 – There is an ARINC word in the receiver

1 – no ARINC data is ready in the receiver

EmptySts : Empty FIFO status.

0 – The RX FIFO contains at least a word

1 – The RX FIFO is empty

HFullSts : Half-full interrupt status.

0 – The RX FIFO is less than half full
1 – The RX FIFO is more than half full
FullSts : Full interrupt status.
0 – The RX FIFO is less than full
1 – The RX FIFO is full

Error Codes:

ARINC429_INVALID_RECVNUMBER
ARINC429_INVALID_HANDLE

ARINC429_READARINCWORD

```
Arinc429Status Arinc429_ReadArincWord (  
ARINC429handle brd,  
UINT16 channel,  
struct sArincWrdFormat *rcvvArincData  
)
```

The function reads one ARINC words from the specified receiver. The channel number is a value between 1 and 8.

Error Codes

ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER

ARINC429_SETLABELS

```
Arinc429Status Arinc429_SetLabels (  
ARINC429handle brd,  
UINT16 channel,  
UINT16 label,  
UINT8 labelEnable  
)
```

The function specifies a label for the specified receiver. The channel number is a value between 1 and 8. The label is between 0 and 255. labelEnable is 1 to enable the label, and 0 to disable it.

Error Codes

ARINC429_INVALID_HANDLE
ARINC429_INVALID_TRANSNUMBER

ARINC429_READLABELMATCH

```
Arinc429Status ARINC429_ReadLabelMatch(  
ARINC429handle brd,  
UINT16 channel,  
UINT16 label,  
UINT16 *labelEnable  
)
```

The function is used to read status label match set by the receiver for the selected receiver channel.

Parameters:

brdHandle : Handle to the board.

channel : Receiver channel number (1-16).

label : Label number to check for the selected channel (0-255).

labelEnable : enable status of the selected channel.

- 0 – Label accepted for the selected channel
- 1 – Label is accepted for the selected channel

Error Codes:

ARINC429_INVALID_HANDLE
 ARINC429_INVALID_RECVNUMBER
 ARINC429_INVALID_LABELNUMBER

ARINC429_INTENABLEDISBLE

```
Arinc429Status ARINC429_IntEnableDisble(
    ARINC429handle brdHandle,
    UINT16 enable
```

);

Parameters

brdHandle : the handle to the board.

enable : whether the interrupt should be enabled or disabled.

- 0 – Disable the interrupt
- 1 – Enable the interrupt

Error Codes:

ARINC429_INVALID_HANDLE
 ARINC429_INT_ALREADY_ENABLED
 ARINC429_FAILED_INT_ENABLE

ARINC429_ENABLEDISABLETXINTERRUPT

```
Arinc429Status ARINC429_EnableDisableTxInterrupt(
    ARINC429handle brdHandle,
    UINT16 channel,
    UINT16 EISts,
    UINT16 HFISts,
    UINT16 FISts
```

)

The function enables / disables interrupt for a transmitter.

Parameters:

brdHandle : Handle to the board.

channel : Receiver channel number (1-16).

EISts : Empty interrupt status.

- 0 – Disables the interrupt generated from TX empty
- 1 – Enables the interrupt generated from TX empty

HFISts : Half-full interrupt status.

- 0 – Disables the interrupt generated from TX HFull
- 1 – Enables the interrupt generated from TX HFull

FISts : Full interrupt status.

- 0 – Disables the interrupt generated from TX Full
- 1 – Enables the interrupt generated from TX Full

Error Codes:

ARINC429_INVALID_HANDLE
 ARINC429_INVALID_TRANNUMBER
 ARINC429_INVALID_EMTINTSTS
 ARINC429_INVALID_HFINTSTS

ARINC429_INVALID_FINTSTS

ARINC429_READTXINTERRUPTSTATUS

```
Arinc429Status ARINC429_ReadTxInterruptStatus(  
    ARINC429handle brdHandle,  
    UINT16 channel,  
    UINT16 *EISts,  
    UINT16 *HFISts,  
    UINT16 *FISts  
)
```

The function reads the interrupt status of receivers.

Parameters:

brdHandle : Handle to the board.
channel : Transmitter channel number (1-8).
EISts : Empty interrupt status.
 0 – No interrupt request from TX empty
 1 – Interrupt request from TX empty
HFISts : Half-full interrupt status.
 0 – No interrupt request from TX HFull
 1 – Interrupt request from TX HFull
FISts : Full interrupt status.
 0 – No interrupt request from TX Full
 1 – Interrupt request from TX Full

Error Codes:

ARINC429_INVALID_TRANNUMBER
ARINC429_INVALID_HANDLE

ARINC429_ENABLEDISABLERXINTERRUPT

```
Arinc429Status ARINC429_EnableDisableRxInterrupt(  
    ARINC429handle brd,  
    UINT16 channel,  
    UINT16 EISts,  
    UINT16 HFISts,  
    UINT16 FISts  
)
```

The function enables / disables interrupt for a receiver.

Parameters:

brdHandle : Handle to the board.
channel : Receiver channel number (1-16).
EISts : Empty interrupt status.
 0 – Disables the interrupt generated from RX empty
 1 – Enables the interrupt generated from RX empty
HFISts : Half-full interrupt status.
 0 – Disables the interrupt generated from RX HFull
 1 – Enables the interrupt generated from RX HFull
FISts : Full interrupt status.
 0 – Disables the interrupt generated from RX Full
 1 – Enables the interrupt generated from RX Full

Error Codes:

ARINC429_INVALID_HANDLE

ARINC429_INVALID_RECVNUMBER
ARINC429_INVALID_EMTINTSTS
ARINC429_INVALID_HFINTSTS
ARINC429_INVALID_FINTSTS

ARINC429_READRXINTERRUPTSTATUS

```
Arinc429Status ARINC429_ReadRxInterruptStatus(  
    ARINC429handle brdHandle,  
    UINT16 channel,  
    UINT16 *EISts,  
    UINT16 *HFISts,  
    UINT16 *FISts  
)
```

The function reads the interrupt status of receivers.

Parameters:

brdHandle : Handle to the board.
channel : Receiver channel number (1-16).
EISts : Empty interrupt status.
 0 – No interrupt request from RX empty
 1 – Interrupt request from RX empty
HFISts : Half-full interrupt status.
 0 – No interrupt request from RX HFull
 1 – Interrupt request from RX HFull
FISts : Full interrupt status.
 0 – No interrupt request from RX Full
 1 – Interrupt request from RX Full

Error Codes:

ARINC429_INVALID_RECVNUMBER
ARINC429_INVALID_HANDLE

ARINC429_GETSEMID

```
Arinc429Status ARINC429_GetSemId(  
    ARINC429handle brdHandle,  
    void **SemId  
)
```

This function returns a pointer to the semaphore descriptor used to handle the interrupts. The interrupt processing thread is requesting the semaphore and blocking on getting it until the interrupt occurs.

Error Codes:

ARINC429_INVALID_HANDLE

ARINC429_READINTERRUPTDATA

```
Arinc429Status ARINC429_ReadInterruptData (  
    ARINC429handle brdHandle,  
    ARINC429_INT_BUFFER *intBufData,  
    UINT16 *dataLength  
)
```

This function returns information about the source of the interrupt(s).

Parameters:

brdHandle : Handle to the board.
intBufData: pointer to an array of ARINC429_INT_BUFFER. It must be at least with as many elements as the sum of the number of receivers plus the number of transmitters, since all of them might have generated an interrupt.
dataLength: number of records returned in intBufData.

Error Codes:

ARINC429_INTERNAL_ERROR
ARINC429_INVALID_HANDLE

STRUCTURES**TRANSMITTER OR RECEIVER CONFIGURATION**

```
struct TxorRxConfiguration
{
    UINT16 usChnNo;           /* Channel Number 1 to 8 */
    UINT16 usEnbDis;         /* Enables (1) or Disables (0) the selected transmitter */
    UINT16 usLabelNo;        /* Label Number: 0 to 255 */
    UINT16 usSDI;            /* Source / Destination Identification process: 0 to 2 */
    UINT16 usParity;         /* Selects odd (0) or even (1) Parity */
    UINT16 usdr;             /* Selects High / Low Speed Transmitter Data Rate */
    UINT16 usWdlen;          /* Word Length (25-bit (1) or 32-bit (0)) */
    UINT16 usSSM;            /* Sign/Status Matrix */
};
```

ARINC429 WORD

```
struct sArincWrdFormat
{
    UINT16 usLable;
    UINT16 usSDI;
    UINT16 usParity;
    UINT16 usSSM;
    UINT32 ulData;
};
```

BUFFER

```
typedef struct
{
    UINT16 usChn;
    UINT16 usTxorRx;
    UINT16 uses;
    UINT16 ushfs;
    UINT16 usfs;
}ARINC429_INT_BUFFER;
```

MISCELLANEOUS

```
typedef UINT16 Arinc429Status
```

HARDWARE DETAILS

JUMPERS

CONNECTORS

The connector placement is depicted below.

Figure 4.2: Connector Locations

CONNECTOR DESCRIPTIONS

IP EXTERNAL I/O CONNECTOR

The signals are routed as follows.

Pin	Connection	Pin	Connection
1	RX1AIN	26	GND
2	RX1BIN	27	TX5AOUT
3	RX9AIN	28	TX5BOUT
4	RX9BIN	29	TX6AOUT
5	RX2AIN	30	TX6BOUT
6	RX2BIN	31	TX7AOUT
7	RX10AIN	32	TX7BOUT
8	RX10BIN	33	TX8AOUT
9	RX3AIN	34	TX8BOUT
10	RX3BIN	35	RX5AIN
11	RX11AIN	36	RX5BIN
12	RX11BIN	37	RX13AIN
13	RX4AIN	38	RX13BIN
14	RX4BIN	39	RX6AIN
15	RX12AIN	40	RX6BIN
16	RX12BIN	41	RX14AIN
17	TX1AOUT	42	RX14BIN
18	TX1BOUT	43	RX7BIN
19	TX2AOUT	44	RX7BIN
20	TX2BOUT	45	RX15AIN
21	TX3AOUT	46	RX15BIN
22	TX3BOUT	47	RX8AIN
23	TX4AOUT	48	RX8BIN
24	TX4BOUT	49	RX16AIN
25	GND	50	RX16BIN

Table 4.1: IP external I/O connector