

ARINC 429 API

for Sital ARINC429 Transmitter & Receiver IP
Cores

Programmer and Reference Guide

Rev 1.4

September 2019

Current documentation version does not include Returned Error

Copyright (C) 2019 by Sital Technology Ltd.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Sital Technology Ltd.

Table of Contents

1	Introduction	4
1.1	Scope	4
1.2	Audience	4
1.3	Related Documentation.....	4
1.4	Support	4
2	Overview & Basic Workflow	5
3	Basic API Reference	6
3.1	sitalDevice_Initialize	6
3.2	sital_A429_GetCount.....	7
3.3	sital_A429_Channel_Open	8
3.4	sital_A429_Channel_Close	9
3.5	sital_A429_Channel_SetConfigRegister	10
3.6	sital_A429_Channel_GetConfigRegister	13
3.7	sital_A429_Channel_GetStatusRegister	14
3.8	sital_A429_Receive.....	15
3.9	sital_A429_Send	17
3.10	sital_A429_GetRxWordsPending.....	18
3.11	sital_A429_Channel_Reset	19
4	Label API Reference	20
4.1	Sital_A429_LabelArray_Reset.....	20
4.2	Sital_A429_LabelArray_Add	21
4.3	Sital_A429_LabelArray_Get.....	22
4.4	Sital_A429_LabelArray_Enable.....	23
4.5	Sital_A429_LabelArray_Disable.....	24
5	Service Functions.....	25
5.1	a429_Write	25
5.2	a429_Read	26
6	User Application – Code Example	27

1 Introduction

1.1 Scope

This document is the programmer and reference guide for the Sital ARINC429 API driving the Sital ARINC429_NTx_MRx IP Core.

1.2 Audience

This document assumes that the reader is familiar with the ARINC 429 protocol.

1.3 Related Documentation

Vivado_SDK_ArtyS7.PDF

1.4 Support

If you have any question or require further assistance, use any of the following methods to contact Sital customer support:

- By Email: support@sitaltech.com
- By Phone : +972-9-7633300

2 Overview & Basic Workflow

The ARINC429_NTx_MRx IP Core targeting ARINC429 protocol is implemented for synthesis on every FPGA device and family.

This document covers the standard API functions that handle the RINC429 activation, transmit and receive functionality.

Basic Workflow

- Read the number of cards (ARINC429_NTx_MRx IP Cores) and number of channels (cross every IP Core) detected.
- Initialize card/s (ARINC429_NTx_MRx IP Core/s)
- Reset all channels.
- Read each channel's information and open them.
- Configure each channel parameters (for example, High / Low speed, Parity, etc.).
- Transmit and receive in the relevant Tx and Rx channels.

NOTE – in ARINC429_NTx_MRx IP Core there are 16 ARINC channels: channels 0 - 7 are Tx channels, channels 8 – 15 are Rx channels.

Please describe what this pseudo code is

```

#define sitalMaximum_CHANNELS_ARINC429                                16U
///< Maximum number of devices in the system per ARINC429_NTx_MRx IP Core.
#define sitalMaximum_ARINC429_NTx_MRx_IP_CORES                       3U
///< Maximum number of devices in the system per ARINC429_NTx_MRx IP Core.
#define sitalMax_TX_CHANNELS_ARINC429                                8U
///< Maximum number of tx channels in the system per ARINC429_NTx_MRx IP Core.
#define sitalIndex_NULL_RX_CHANNEL_ARINC429                          8U
///< First index of rx channel in the system per ARINC429_NTx_MRx IP Core.
#define sitalMax_RX_CHANNELS_ARINC429                                8U
///< Maximum number of rx channels in the system per ARINC429_NTx_MRx IP Core.

```

3 Basic API Reference

3.1 `sitalDevice_Initialize`

```

S16Bit sitalDevice_Initialize (
    U16BIT swCard
    U16BIT wAccess
    U16BIT wMode
    U32BIT dwSizeOfAllocatedMemory
    U32BIT dwRegistersAddress
    U32BIT dwMemoryAddress
)
    
```

Parameters

<i>swCard</i>	(in) Unique Card /Device ID 0 - (sitalMaximum_A429_IP_CORES - 1)
<i>wAccess</i>	Type of device access (sitalAccess_*)
<i>wMode</i>	Operation mode (sitalMode_*)
<i>dwSizeOfAllocatedMemory</i>	Size of memory (unused)
<i>dwRegistersAddress</i>	Base address for registers (unused)
<i>dwMemoryAddress</i>	Base address for memory (unused)

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Initialize hardware & software resources.

NOTE – in order to properly initialize ARINC429_NTx_MRx IP Cores, use the following function parameters:
`sitalDevice_Initialize(0 /*<Card Number>*/, sitalAccess_CARD, sitalMode_ARINC429, 0, 0, 0);`

3.2 `sital_A429_GetCount`

```
S16Bit  sital_A429_GetCount    (
                                U16BIT*    swCards
                                U16BIT*    numOfChannels
                                )
```

Parameters

<i>swCards</i>	A pointer which returns the number of cards (ARINC429_NTx_MRx IP Cores) detected
<i>numOfChannels</i>	A pointer which returns the total number of channels detected across all cards (ARINC429_NTx_MRx IP Cores).

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function returns the number of detected cards and accumulated number of ARINC429 channels on all cards.

```
⋮
S16BIT result;
U16BIT cards, channels;

result = sital_A429_GetCount(&cards, &channels);
if (result < 0) {
    printf("\nsital_A429_GetCount Failed");
    getchar();
    return -1;
}
```

3.3 `sital_A429_Channel_Open`

```
S16Bit  sital_A429_Channel_Open (
                                U16BIT      swCard
                                U16BIT      channel
                                )
```

Parameters

<i>swCard</i>	Card number
<i>Channel</i>	Channel number

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function opens the specified channel and returns updated information on the specified channel.

IMPORTANT: This function must be used for each channel before communicating with it as it prepares the channel for Tx Rx communications operations and bringing it out from reset state.

```
:
for (int i = 0; i < cards; i++) {
    for (int j = 0; j < (channels / cards); j++) {
        result = sital_A429_Channel_Reset(i, j, 0);
        if (result < 0) {
            printf("\sital_A429_Channel_Reset Failed - card %i, channel %i, error %i", i, j, result);
            getchar();
            return -1;
        }
        result = sital_A429_Channel_Open(i, j);
        if (result < 0) {
            printf("\nsital_A429_Channel_Open Failed");
            getchar();
            return -1;
        }
    }
}
```


3.4 `sital_A429_Channel_Close`

```
S16Bit sital_A429_Channel_Close (
                                U16BIT      swCard
                                U16BIT      Channel
                                )
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function closes the specified channel.

IMPORTANT: Once a channel is closed it cannot perform Arinc429 operations until `sital_A429_Channel_Open(..)` is invoked.

```
⋮
S16BIT result;
U16BIT card, channel;

result = sital_A429_Channel_Close(card, channel);
```

3.5 `sital_A429_Channel_SetConfigRegister`

```
S16Bit
sital_A429_Channel_SetConfigRegister (
    U16BIT          swCard
    U16BIT          channel
    U32BIT          chanFlags
)
```

Parameters

<i>swCard</i>	Card number
<i>Channel</i>	Channel number
<i>chanFlags</i>	Channel flags by the Tx and Rx vectors specified below

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function sets the configuration for the specified channel. The channel flags are described below and can be used with the following constants, also defined in SitalAPI.h file.

```
// Following bits are same for TX and RX modes:
#define sital_A429_CFG_HIGH_RATE          0x0001 ///< Data rate: 100 KHz
#define sital_A429_CFG_LOW_RATE          0x0000 ///< Data rate: 12.5 KHz
#define sital_A429_CFG_MASK_RATE          0x0001
#define sital_A429_CFG_PARITY_NONE data. 0x0000 ///< Data parity: none. MSB can be used as
data.
#define sital_A429_CFG_PARITY_EVEN          0x0002 ///< Data parity: even
#define sital_A429_CFG_PARITY_ODD          0x0006 ///< Data parity: odd
#define sital_A429_CFG_MASK_PARITY          0x0006 // Mask for parity bits

#define sital_A429_NUMBER_OF_WORDS_MASK Bits 0-7 0x000000FF // Number of words in FIFO mask -
Bits 0-7
#define sital_A429_FIFO_FULL          0x00000100 // FIFO full
#define sital_A429_FIFO_EMPTY          0x00000200 // FIFO empty

#define sital_A429_TRANSMIT_ADDRESS          0x0
#define sital_A429_RECEIVE_ADDRESS          0x0

#define sital_A429_CONFIG_ADDRESS          0x4
#define sital_A429_STATUS_ADDRESS          0x8
```

Tx Configuration

--Each Tx IP gets 32 bytes (=8 dwords)

--Tx Mapping (in bytes):

IP number	Base Offset
0	0x00
1	0x20
2	0x40
3	0x60
4	0x80
5	0xA0
6	0xC0
7	0xE0

--

--For each IP 32 bytes Tx Mapping:

--

-- Offset Usage

- +0x0 Write Tx_Data(07 downto 00)
- +0x1 Write Tx_Data(15 downto 08)
- +0x2 Write Tx_Data(23 downto 16)
- +0x3 Write Tx_Data(31 downto 24)
- +0x4 Write Tx_Control(7 downto 0)

--

- +0x8 Read Tx_FIFO_NumOfWords(7 downto 0)
- +0x9 Read Tx_FIFO_Empty in bit 0
- +0x9 Read Tx_FIFO_Full in bit 1

--

Rx Configuration

--Each Rx IP gets 32 bytes (=8 dwords)

—

--Rx Mapping (in bytes):

IP number	Base Offset
0	0x200
1	0x220
2	0x240
3	0x260
4	0x280
5	0x2A0
6	0x2C0
7	0x2E0

--

--For each IP 32 bytes Rx Mapping:

--

-- Offset Usage

+0x0	Read Rx_Data(07 downto 00)
+0x1	Read Rx_Data(15 downto 08)
+0x2	Read Rx_Data(23 downto 16)
+0x3	Read Rx_Data(31 downto 24)
+0x4	Write Rx_Control(7 downto 0)
--	
+0x8	Read Rx_FIFO_NumOfWords(7 downto 0)
+0x9	Read Rx_FIFO_Empty in bit 0
+0x9	Read Rx_FIFO_Full in bit 1
--	

Example of function Usage:

U32BIT cfHigh = (sital_A429_CFG_HIGH_RATE | sital_A429_CFG_PARITY_NONE);

U32BIT cfLow = (sital_A429_CFG_LOW_RATE | sital_A429_CFG_PARITY_NONE);

result = sital_A429_Channel_SetConfigRegister(0, 0, cfHigh); //cfLow);

result += sital_A429_Channel_SetConfigRegister(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, cfHigh);
//cfLow);

```

if (result < 0) {
    printf("\nsital_A429_Channel_SetConfigRegister Failed");
    getchar();
    return -1;
}

```

3.6 `sital_A429_Channel_GetConfigRegister`

```

S16Bit
sital_A429_Channel_GetConfigRegister (
                                U16BIT          swCard
                                U16BIT          channel
                                U32BIT *       chanFlags
                                )

```

Parameters

<i>swCard</i>	Card number
<i>Channel</i>	Channel number
<i>chanFlags*</i>	Returns the Channel flags by the Tx and Rx vectors specified below

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function gets the configuration for the specified channel.
 The channel flags are described for `sital_A429_Channel_SetConfigRegister` function.

Example of function Usage:

```

U32BIT chanFlags;

result = sital_A429_Channel_GetConfigRegister(0, 0, &chanFlags);

if (result < 0) {
    printf("\nsital_A429_Channel_GetConfigRegister Failed");
    getchar();
    return -1;
}

```

3.7 `sital_A429_Channel_GetStatusRegister`

```

S16Bit
sital_A429_Channel_GetStatusRegister (
                                U16BIT           swCard
                                U16BIT           channel
                                U32BIT*        chanStats
)
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>chanStats*</i>	Returns the Channel status by the Tx and Rx vectors specified below

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function gets the status register for the specified channel. It can be used to determine the FIFO state and number of words in Tx/Rx FIFO.

The channel flags are described below and can be used with the following constants, also see description of `sital_A429_Channel_SetConfigRegister` function.

```

#define sital_A429_NUMBER_OF_WORDS_MASK      0x000000FF // Number of words in FIFO mask -
Bits 0-7
#define sital_A429_FIFO_FULL                 0x00000100 // FIFO full
#define sital_A429_FIFO_EMPTY               0x00000200 // FIFO empty
    
```

```

#define sital_A429_TRANSMIT_ADDRESS         0x0
#define sital_A429_RECEIVE_ADDRESS         0x0
    
```

```

#define sital_A429_CONFIG_ADDRESS          0x4
#define sital_A429_STATUS_ADDRESS          0x8
    
```

Example of function Usage:

```

S16BIT result = 0;
U32BIT sr = 0x99;
result = sital_A429_Channel_GetStatusRegister (swCard, channel, &sr);
    
```

3.8 `sital_A429_Receive`

```

S16Bit  sital_A429_Receive  (
                                U16BIT      swCard
                                U16BIT      channel
                                U32BIT      bufferSize
                                U32BIT*     Buffer
                                U32BIT*     numberOfReceivedWords
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>bufferSize</i>	Size of assigned buffer
<i>Buffer</i>	A pointer to the buffer in which the data returned
<i>numberOfReceivedWords</i>	Number of words actually received on the bus

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function gets the data received and number of words received on the bus, which are in a buffer.

TIP: This function can be coupled with `sital_A429_GetRxWordsPending(..)` to check if data is received and waiting in the FIFO.

Example of function usage:

U32BIT numberOfWords;

```
result = sital_A429_GetRxWordsPending(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, &numberOfWords);
```

```
if (result < 0) {  
    printf("\nsital_A429_GetRxWordsPending Failed");  
    getchar();  
    return -1;  
}
```

```
if(numberOfWords > 0)
```

```
{  
    result = sital_A429_Receive(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, numberOfWords, readBuff, &rcv);  
    if (result < 0) {  
        printf("\nsitalIndex_NULL_RX_CHANNEL_ARINC429 Failed");  
        getchar();  
        return -1;  
    }  
}
```


3.9 `sital_A429_Send`

```

S16Bit  sital_A429_Send      (
                                U16BIT      swCard
                                U16BIT      channel
                                U32BIT      bufferSize
                                U32BIT*     Buffer
                                U32BIT*     numberOfWrittenWords
                                )
  
```

Parameters

<i>swCard</i>	Card number
<i>Channel</i>	Channel number
<i>bufferSize</i>	Size of assigned buffer
<i>Buffer</i>	A pointer to the data buffer to send
<i>numberOfWrittenWords</i>	Number of words actually sent on the bus

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function transmits the data buffer on the bus and returns the number of words sent.

Example of function usage:

```

for (int idx = 0; idx < 50; idx++)
{
    buff[idx] = idx;
    readBuff[idx] = 0;
}
printf("Tx from 0\n");

result = sital_A429_Send(0, 0, 50, buff, &written);
if (result < 0) iterationsFailed++;
  
```

3.10 `sital_A429_GetRxWordsPending`

```

S16Bit
sital_A429_GetRxWordsPending (
    U16BIT           swCard
    U16BIT           Channel
    U32BIT*         numberOfWords
)
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>numberOfWords</i>	Number of pending words in the Rx FIFO

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function gets the number of words pending in the Rx FIFO.

Example of function usage:

```
U32BIT numberOfWords;
```

```
result = sital_A429_GetRxWordsPending(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, &numberOfWords);
```

```

if (result < 0) {
    printf("\\nsital_A429_GetRxWordsPending Failed");
    getchar();
    return -1;
}
    
```

```

if(numberOfWords > 0)
{
    result = sital_A429_Receive(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, numberOfWords, readBuff, &rcv);
    if (result < 0) {
        printf("\\nsitalIndex_NULL_RX_CHANNEL_ARINC429 Failed");
        getchar();
        return -1;
    }
}
    
```

3.11 `sital_A429_Channel_Reset`

```

S16Bit
sital_A429_Channel_Reset      (
                                U16BIT          swCard
                                U16BIT          channel
                                U32BIT          chanFlags
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>Channel</i>	Channel number
<i>chanFlags</i>	Not in use

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This functions resets the specified channel’s flags and closes the connection

TIP: For reusing the selected channel, call `sital_A429_Channel_Open(..)` again.

Example of function usage:

```

for (int i = 0; i < cards; i++) {
    for (int j = 0; j < (channels / cards); j++) {
        result = sital_A429_Channel_Reset(i, j, 0);
        if (result < 0) {
            printf("\nsital_A429_Channel_Reset Failed - card %i, channel %i, error %i", i, j, result);
            getchar();
            return -1;
        }
        result = sital_A429_Channel_Open(i, j);
        if (result < 0) {
            printf("\nsital_A429_Channel_Open Failed");
            getchar();
            return -1;
        }
    }
}
    
```

4 Label API Reference

4.1 Sital_A429_LabelArray_Reset

```

S16Bit
sital_A429_LabelArray_Reset    (
                                U16BIT    swCard
                                U16BIT    channel
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Clear out label array for Rx channel.

IMPORTANT: After calling this function **sital_A429_Receive** will receive nothing if **sital_A429_LabelArray_Enable** is called. In order to enable a label for receive use function **sital_A429_LabelArray_Add**.

4.2 Sital_A429_LabelArray_Add

```

S16Bit
sital_A429_LabelArray_Add      (
                                U16BIT      swCard
                                U16BIT      Channel
                                U8BIT      Label
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>label</i>	Label value

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Add the label to label array for Rx channel.

4.3 Sital_A429_LabelArray_Get

```

S16Bit
sital_A429_LabelArray_Get      (
                                U16BIT      swCard
                                U16BIT      channel
                                U8BIT *     labels
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>labels</i>	Label's values

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Get all the labels of label array for Rx channel.

IMPORTANT: Please make sure sufficient memory space is allocated for all the labels. Labels array should be not less than `sital_A429_MAX_LABELS_NUMBER` long.

```
#define sital_A429_MAX_LABELS_NUMBER    120U
```

4.4 Sital_A429_LabelArray_Enable

```

S16Bit
sital_A429_LabelArray_Enable    (
                                U16BIT    swCard
                                U16BIT    channel
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function enables the use of LabeArray. Once call, the **sital_A429_Receive** function will receive the only appropriate words – with matching labels, which are included in LabelArray.

TIP:This function can be coupled with **sital_A429_GetRxWordsPending(..)** to check if data is received and waiting in the FIFO.

4.5 Sital_A429_LabelArray_Disable

```

S16Bit
sital_A429_LabelArray_Disable (
    U16BIT      swCard
    U16BIT      Channel
)
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

This function disables the use of LabeArray . Once called, the **sital_A429_Receive** function will receive the all words from the bus, regardless of their label.

TIP: This function can be coupled with **sital_A429_GetRxWordsPending(..)** to check if data is received and waiting in the FIFO.

5 Service Functions

5.1 a429_Write

```

S16Bit  a429_Write          (
                                U16BIT          swCard
                                U16BIT          channel
                                U32BIT          address
                                U32BIT          len
                                U32BIT*        buffer
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>address</i>	Writing address
<i>len</i>	Size of buffer to write
<i>buffer</i>	Pointer to the writing buffer

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Mode: Ready & Runtime

This service function writes the data in buffer to the specified address.

5.2 a429_Read

```

S16Bit  a429_Write          (
                                U16BIT      swCard
                                U16BIT      channel
                                U32BIT      address
                                U32BIT      len
                                U32BIT*     buffer
                                )
    
```

Parameters

<i>swCard</i>	Card number
<i>channel</i>	Channel number
<i>address</i>	Writing address
<i>len</i>	Size of buffer to write
<i>buffer</i>	Pointer to the buffer read

Returns

sitalReturnCode_SUCCESS Function successfully completed	sitalReturnCode_SUCCESS Function successfully completed
Negative sitalReturnCode_* Error condition or function failed	Negative sitalReturnCode_* Error condition or function failed

Description

Mode: Ready & Runtime

This service function returns the data in buffer read from a specified address.

6 User Application – Code Example

The Following fragment demonstrates how to send and receive ARINC 429 buffers on ARINC Tx and Rx channels

```
unsigned long long iterations = 0;
//unsigned long long iterationsCompleted = 0;
unsigned long long iterationsfailed = 0;

while (1)
{
    for (int idx = 0; idx < 50; idx++)
    {
        buff[idx] = idx;
        readBuff[idx] = 0;
    }
    printf("Tx from 0\n");

    result = sital_A429_Send(0, 0, 50, buff, &written);
    if (result < 0) iterationsfailed++;
    usleep(800*1000);

    result =
    sital_A429_Receive(0, sitalIndex_NULL_RX_CHANNEL_ARINC429, 50, readBuff, &rcv);

    if (result < 0) iterationsfailed++;

    for (int idx = 0; idx < 50; idx++) {
        if (buff[idx] != readBuff[idx])
        {
            iterationsfailed++;
        }
    }

    iterations++;
    printf("\n\nIterations: %llu, Iterations Failed: %llu\n",
            iterations, iterationsfailed);
}
}
```



17 Atir Yeda St., Kfar-Saba, ISRAEL 44643

Email: info@sitaltech.com

Website: <http://www.sitaltech.com>

The information provided in this User's Guide is believed to be accurate; however, no responsibility is assumed by Sital Technology for its use, and no license or rights are granted by implication or otherwise in connection therewith. Specifications are subject to change without notice.

Please visit our Web site at <http://www.sitaltech.com> for the latest information.

© 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 by Sital Technology.