Texas Instruments

Series 2000 Reader System

TIRIS Bus Protocol

Reference Guide

TIRIS Technology by Texas Instruments™

March 2000

SCBU026

Series 2000 Reader System

TIRIS Bus Protocol

Reference Guide

TIRIS Technology by Texas Instruments™

Literature Number: SCBU026
March 2000

Texas Instruments

Contents

List of Figures

List of Tables

Edition One - March 2000

This is the first edition of this manual. It describes the TIRIS™ Bus Protocol that can be used with the Series 2000 readers, the Series 2000 control module, and the S2510 reader.

About This Guide

This manual describes the TIRIS Bus Protocol (TBP) that can be used for communication between the TIRIS S2000/S2510 reader and a host system. It describes the different command codes that can be used and the timing involved in the communication.

Conventions

WARNING

A warning is used where care must be taken, or a certain procedure must be followed in order to prevent injury or harm to your health.

CAUTION

This indicates information on conditions which must be met, or a procedure which must be followed, which if not heeded could cause permanent damage to the equipment or software.

Note: Indicates conditions which must be met, or procedures which must be followed, to ensure proper functioning of the equipment or software.

If You Need Assistance

Application Centers are located in Europe, North and South America, the Far East and Australia to provide direct support. For more information, please contact your nearest TIRIS Sales and Application Center. The contact addresses can be found on our home page: http://www.tiris.com

Trademarks

The TIRIS logo and the words TIRIS and Tag-it are trademarks or registered trademarks of Texas Instruments.

Introduction

This chapter will introduce you to the TIRIS Bus Protocol, what it is, where and how to use it. It goes on to describe the format of the messages and what is contained in each part of the message. The last part of the chapter describes how the flow of messages is controlled.

Topic **Page**

1.1 General

The TIRIS Bus Protocol (TBP) defines the format, content and meaning of messages exchanged between TIRIS reader stations and a host system. The protocol is designed to permit efficient communication between a single master and multiple slaves with a minimum of software complexity.

1.1.1 References

The CCITT Red Book, Volume VIII, Geneva, 1986

1.2 Protocol Description

The TIRIS Bus Protocol is a single-master, half-duplex, byte-count protocol suitable for communication between one master and several slaves. It may be used, for example: with a single slave unit on a RS232 line, with up to 31 slaves on a RS422/485 line, or for parallel links.

All data transfers are initiated by the host. Slave units only send data when requested to do so by the host. There cannot, therefore, be any bus contention. All data are transferred by messages. All messages, regardless of direction, have the same format.

The protocol employs a positive acknowledgment scheme whereby a slave indicates receipt of a message from the master.

1.3 Message Format

All messages are constructed according to the following format:

Byte	Contents
0	Start-Mark (SOH, 01_{hex})
1	Dest-Address
2	Source-Address
3	Message-Code
4	Data-Length
5	Data-Field(1)
. .	. .
N+4	Data-Field(N)
N+5	CRC-Field(1) (Most Significant Byte)
N+6	CRC-Field(2)
N+7	End-Mark (EOT, 04_{hex})

1.3.1 Start-Mark

The 'Start-Mark' signifies the beginning of a message. It is represented by the ASCII character SOH ('Start Of Header', 01_{hex}).

1.3.2 Destination Address

This byte indicates the destination of the message. The binary value corresponds to the unit ID of the destination or a special code for broadcast messages. Unit IDs range from 0x00 to $0xFE_{hex}$ (254 decimal). The maximum number of units possible in a specific application will be determined by physical limitations like, for example: 31 readers on a RS485 bus. A destination address of $0xFF_{hex}$ indicates a broadcast message.

The main purpose of broadcast messages, received by all slaves, are system maintenance functions. They may also be used to initiate time-synchronized operations.

1.3.3 Source-Address

This byte indicates the source of the message. The binary value corresponds to the unit ID of the sending unit.

1.3.4 Message-Code

The 'Message-Code' byte defines the meaning of the message. Depending on the direction of transfer, the code contained in this byte will be a 'Command-Message-Code' (master-to-slave) or a 'Response-Message-Code' (slave-to-master).

1.3.4.1 Command-Message-Code

The 'Command-Message-Code' indicates to the slave the action requested by the master. It has the following structure:

	Bit	Contents
MSB	7	Queued-Response Flag
	6	Command-Code (0..127)
	5	
	4	
	3	
	2	
	1	
LSB	0	

The 'Queued-Response-Flag' indicates that the destination unit should acknowledge and execute the command but that its response should be queued until requested by the master. This mechanism is utilized for certain commands which have a delayed response to avoid suspension of other line activity.

Queued-Response Flag (MSB)

0 = no queued response

1 = queued response

The command code indicates the operation requested by the master. For more details refer to Chapter 2, Command Codes.

1.3.4.2 Response-Message-Code

The 'Response-Message-Code' indicates the slave's acknowledgment of the command.

It has the following structure:

	Bit	Contents
MSB	7	Error-Flag
	6	Busy-Flag
	5	Data-Available-Flag
	4	Broadcast-Received-Flag
	3	Response-Code (0..15)
	2	
	1	
LSB	0	

All flags are set with '1' and reset with '0'.

The 'Error-Flag' indicates that the slave detected an error in the transmission or contents of the command. The error encountered is described by the 'Response-Code'.

The 'Busy-Flag' indicates that the slave is temporarily unable to accept commands. The master can retry the command at a later time.

The 'Data-Available-Flag' informs the master that there is at least one message in the slave's output queue. The master can access the queue by issuing the appropriate command.

The 'Broadcast-Received-Flag' signifies that a previously broadcast message has been received correctly. The flag is set only if a broadcast message contained no errors and could be actioned. The response to the broadcast command will be queued and can be accessed by the master using a 'Send-Next' command. The flag is reset once it has been transmitted in a response message.

The 'Response-Code' describes the error encountered in more details. Depending on the 'Error-Flag' the 'Response-Code' has different meanings:

Process-Response (Error-Flag=0)	Error-Response (Error-Flag=1)
0 = Command-Completed	0 = Transmission-Error
1 = Accepted (Queued)	1 = Command-Invalid
2 = Queue Empty	2 = Task Error
3 = Nothing to Resend	3 = Data field length - Error
	4 = Parameter - Error

Command-Completed	Indicates that a non-queued-response command has been received and executed correctly. Any relevant data will be contained in the data field.
Accepted (Queued)	Indicates that a queued-response command has been correctly received and is being executed. The response will not be transmitted directly, however, but will be placed on the queue until requested by the master issuing a 'Send-Next' command.
Queue Empty	Indicates that there is no message in the queue.
Nothing to Resend	Indicates that there is no information for resending available.

Transmission-Error	Indicates that the slave received a message containing its own destination ID but found an error in the data. The slave uses this code to signify that either the message failed the CRC check or the complete frame was not received within the expected time.
Command-Invalid	Indicates that although the complete message frame was received and the CRC check was correct, the command code is not valid for this slave. This could e.g. result from the use of a wrong slave destination ID.
Task-Error	Indicates that, while the transmission was successful, the command was rejected by the processing task.
Data field length - Error	Indicates that the command was not executed because the data field length was wrong.
Parameter Error	Indicates that the command was not executed due to an invalid parameter.

1.3.5 Data-Length

The 'Data-Length' byte indicates the length, in bytes, of the following data field. If no data field is required by the command or response, 'Data-Length' will be zero. Otherwise, the binary value (1 to 255 decimal) defines the number of data bytes which will follow.

1.3.6 Data-Field

The 'Data-Field' exists only if 'Data-Length' is non-zero. It consists of the number of bytes specified in 'Data-Length'. The content of the field depends on the message code and may include any byte value. There is no special end code for the data section.

1.3.7 CRC-Field

There are two methods of checking the validity of a TBP message: one is the (default) reverse CRC-CCITT described in Section 1.3.7.1, and the other a less secure LRC algorithm is described in Section 1.3.7.2. Which of these used is selected by the configuration part of the TIRIS Reader Manager (TRM) at start-up.

1.3.7.1 Reverse CRC-CCITT

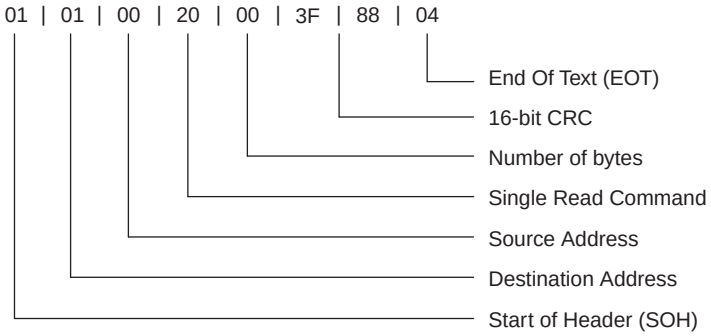
The CRC field is a two-byte value of the Cyclic Redundancy Check calculation for the preceding message. The CRC calculation is performed on the whole message excluding the 'SOH' character, the BCC field and the 'EOT' character.

The CCITT polynomial used is defined in "The CCITT Red Book, Volume VIII, International Telecommunications Union, Geneva, 1986" and is:

$$P(X)_{CCITT} = X^{16} + X^{12} + X^{5} + 1$$

The first byte of the CRC field contains the MSB (Most Significant Byte).

Single READ Command



Single READ Response (No Tag Read)

01	00	01	00	01	40	47	67	04

Single READ Response (Tag number 0 Read)

01	00	01	00	09	01	00	00	00	00	00	00	00	00	8D	A6	04

Single READ Response (Tag number 3 Read)

01	00	01	00	09	01	03	00	00	00	00	00	00	00	07	76	04

Single READ Response (Tag number 9 Read)

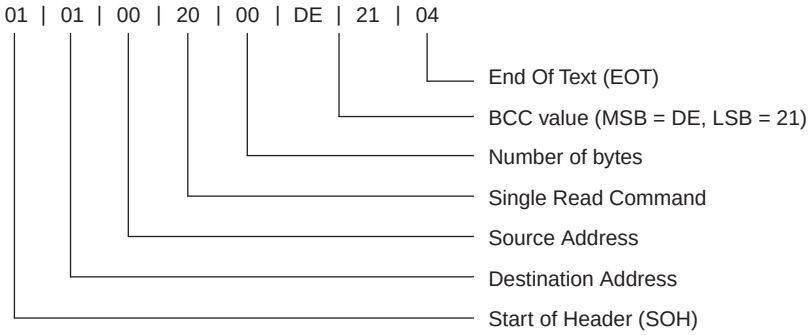
01	00	01	00	09	01	09	00	00	00	00	00	00	00	21	A5	04

1.3.7.2 LRC Alternative Block Check Algorithm for the TBP

The BCC field is a two-byte value containing the Longitudinal Redundancy Check (LRC) values for the preceding message. The LRC calculation is performed on the whole message excluding the 'SOH' character, the BCC field and the 'EOT' character.

Each character is XORed with the previous value and the result is stored in the LSByte, the 1’s Complement of the result is stored in the MSByte. If the MSB and the LSB are ANDed together the result is zero.

Single READ Command



Single READ Response (No Tag Read)

01	00	01	00	01	40	BF	40	04

Single READ Response (Tag number 0 Read)

01	00	01	00	09	01	00	00	00	00	00	00	00	00	F6	09	04

Single READ Response (Tag number 3 Read)

01	00	01	00	09	01	03	00	00	00	00	00	00	00	F5	0A	04

Single READ Response (Tag number 9 Read)

01	00	01	00	09	01	09	00	00	00	00	00	00	00	FF	00	04

1.3.8 End-Mark

The 'End-Mark' signifies the end of a message. It is represented by the ASCII character EOT ('End Of Transmission', 04_{hex}).

1.4 Flow-Control

Flow control is not possible with a two-wire, half-duplex circuit (for example: RS485). All units using this type of data transmission must be capable of receiving and buffering the maximum length message at the line operating speed. If a unit is unable to correctly handle the incoming data stream, it must wait until the message ends and then transmit an error response.

Under some special circumstances, it may be necessary for a slave to mask its communications interrupt while performing other processing. When the slave re-enables its monitoring of the line, it may incorrectly interpret the contents of the data field as the start of a new message frame. This situation should be caught by a failure of the CRC check or the expiration of the Inter-Byte time-out at the end of the partial frame. A slave should not transmit an error response for the first frame received after re-enabling line monitoring in order to avoid interfering with (possibly correct) reception by another slave.

1.5 Data Flow

1.5.1 Immediate Commands

Data flow is always controlled by the master unit. The sequence of 'Command-Response' operations for immediate commands is shown in Figure 1-1.

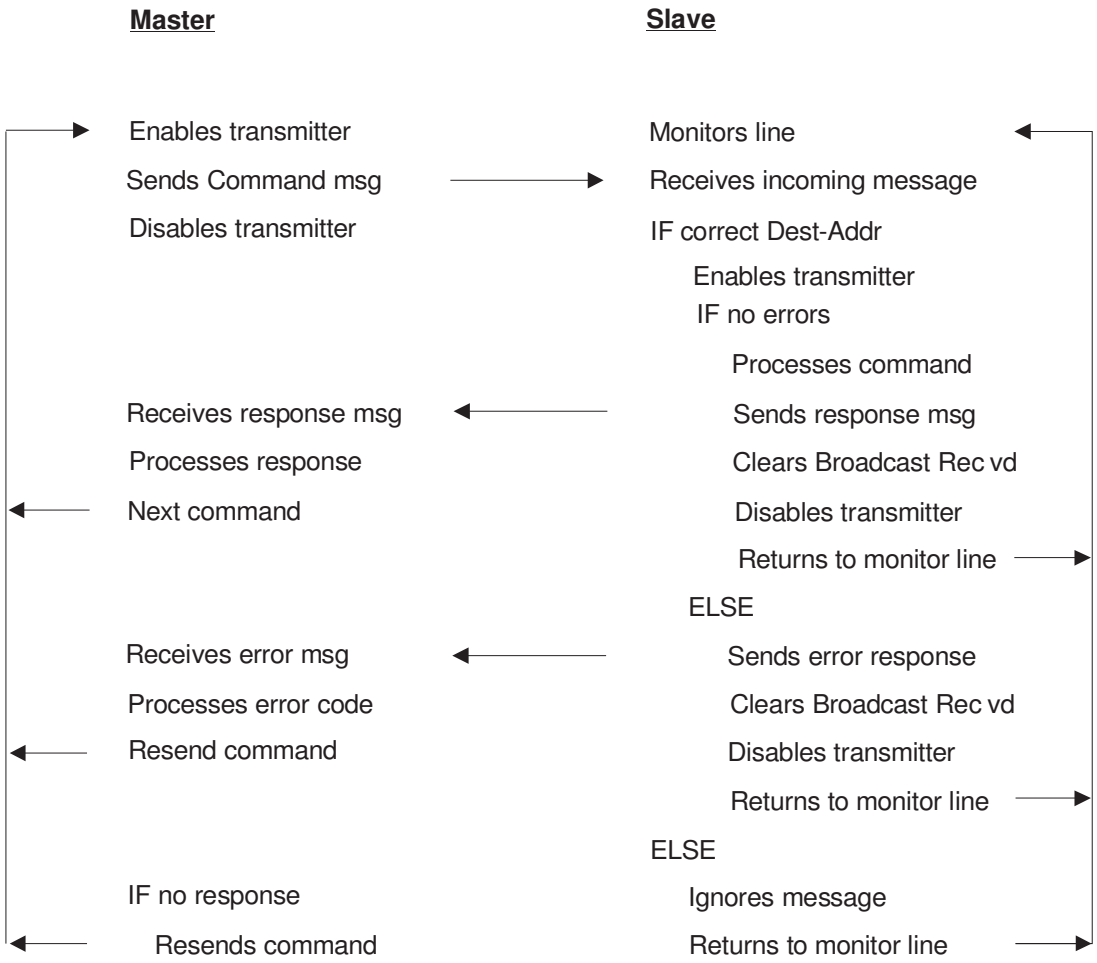


Figure 1-1. Immediate Commands

If the master requests a queued response, the slave's response is simply an acknowledgment of correct receipt of the message. The master must send a further command, at a later time, to retrieve the full response from the queue.

1.5.2 Broadcast Messages

The sequence of 'Command-Response' operations for broadcast commands is shown in Figure 1-2.

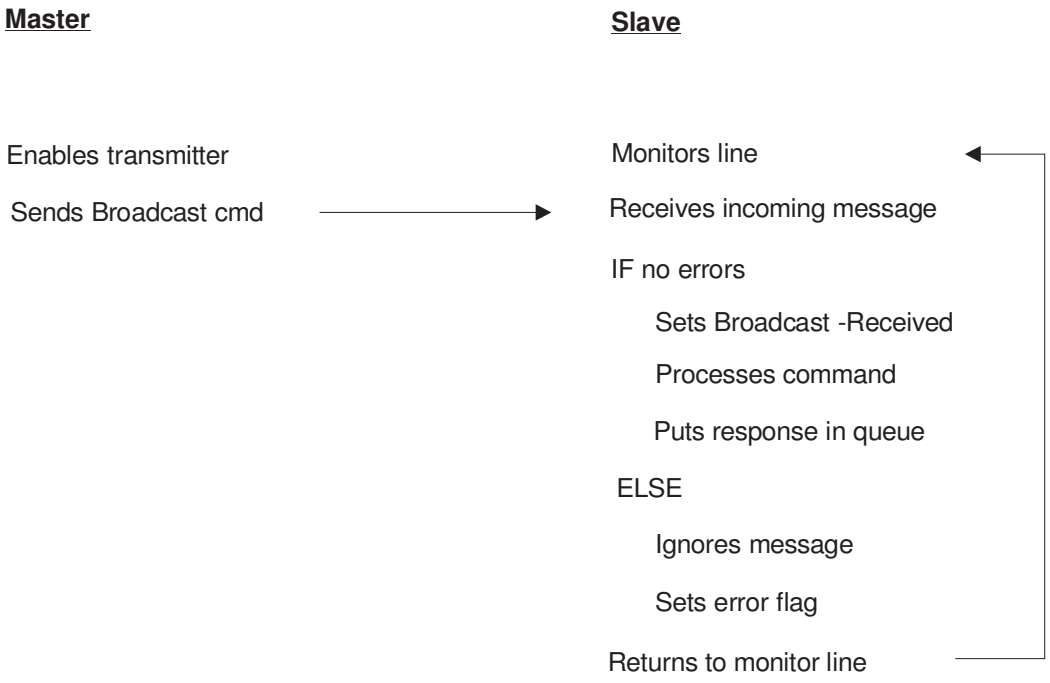


Figure 1-2. Broadcast Commands

As previously noted, broadcast messages are not acknowledged in the normal way. Verification of their receipt is performed by the master through the special status bit in the 'Response-Code' field of the slave's response to the next non-broadcast command.

As an example of this mechanism, assume that a master wishes to initiate an operation at all slaves simultaneously. It sends a broadcast command for the required operation (destination ID = $0xFF_{hex}$). The master waits for the expected operation time and then polls each individual slave with a 'Send-Next-Record' command. Each slave's response will contain the 'Broadcast-Received-Flag' which will verify correct reception or indicate that a retry may be required. Where the broadcast was received, the remainder of the response code and the data field will contain the response.

1.5.3 Queued Messages

1.5.3.1 Message Format and Response

The message format of a queued message is slightly different to a broadcast or immediate message. The Nth data field element of a command message contains a Sequence Number, which is given by the master unit in order to have a reference when the queued messages are requested from the slave units. The slave unit indicates that a queued-response command has been correctly received and is being executed by sending a response message to the master unit where the Accepted (Queued) flag is set in the response code.

The response will not be transmitted directly from the slave unit, but will be placed on the slave's queue until requested by the master issuing a *Send Next Record from Queue* command.

Queued Command Message		Response Message (Acknowledge)	
Byte	**Contents**	**Byte**	**Contents**
0	Start-Mark (SOH, 01_{hex})	0	Start-Mark (SOH, 01_{hex})
1	Dest-Address	1	Dest-Address
2	Source-Address	2	Source-Address
3	Command-Code	3	Response-Code
4	Data-Length	4	Data-Length = 0
5	Data-Field(1)	5	CRC-Field(1) (Most Significant Byte)
.	.	6	CRC-Field(2)
.	.		
.	.	7	End-Mark (EOT, 04_{hex})
.	.		
N+4	Data-Field(N) Sequence Number		
N+5	CRC-Field(1) (Most Significant Byte)		
N+6	CRC-Field(2)		
N+7	End-Mark (EOT, 04_{hex})		

The following example shows a queued response message resulting from a Send Next Record from Queue or Send Record N from Queue Command. The response message of a queued command sends back this Sequence Number in the Nth data field element and the corresponding command code in N-1 data field.

Send Next Rec. Command message		Queued Response message	
Byte	**Contents**	**Byte**	**Contents**
0	Start-Mark (SOH, 01_{hex})	0	Start-Mark (SOH, 01_{hex})
1	Dest-Address	1	Dest-Address
2	Source-Address	2	Source-Address
3	Command-Code	3	Response-Code
4	Data-Length	4	Data-Length
5	CRC-Field(1) (Most Significant Byte)	5	Data-Field(1)
6	CRC-Field(2)	.	.
		.	.
7	End-Mark (EOT, 04_{hex})	.	Data-Field (N-1) Command-Code
		.	
		N+4	Data-Field(N) Sequence Number
		N+5	CRC-Field(1) (Most Significant Byte)
		N+6	CRC-Field(2)
		N+7	End-Mark (EOT, 04_{hex})

If the slave unit receives a queued message, the Sequence number is automatically stripped off during receiving and checking of the command message and the Data Length is decremented.

In the same way, the Sequence Number and the Command-Code are added to the response message and the Data-Length is automatically incremented by 2.

When the master unit receives a queued message, the Command-Code and the Sequence Number must be stripped off and the Data-Length must be decremented by 2 while the message is being received and checked.

The Sequence Number gives the reference to the previous command and the Command Code gives the information to decode the response data. For example, if a Charge Only Read command is issued several times, the sequence counter should be incremented with each command, thus giving a reference to the read status when the responses are requested using the Send Record N from Queue or Send Next Record from Queue Command.

1.5.3.2 Queue Management

Whenever a reader is asked to queue a response, either by setting the command high bit, or by issuing a broadcast command, those responses are held in a 30 position circular buffer. There are two pointers associated with that circular buffer, they are the “Store Next Message” and the “Read Next Message” pointers.

After each host computer (master unit) Next Record Command (0x01), the Read Next Message pointer is moved to the next location. At the same time, a copy of the response is transferred to the “resend Buffer”, in case Resend Last Message (0x03) command is issued.

It is not necessary to delete responses as when the “Store Next Message” pointer is incremented, it overwrites previous responses.

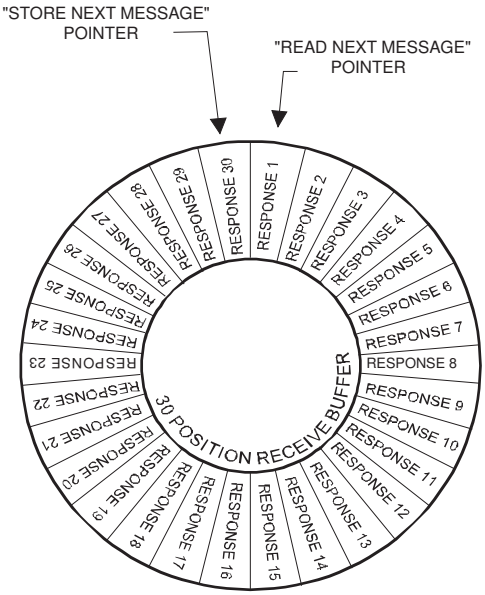


Figure 1-3. Response Buffer

As an example (see Figure 1-3), if 29 locations have been filled, and the “Read Next Message” pointer is at location 1; then the next response will be stored in location 30, the “Store Next Message” pointer moved to location 1 and the “Read Next Message” is incremented to location 2 because location 1 will be overwritten.

Command Codes

This chapter describes the four groups of command codes that can be used. It lists the command codes per group and gives you the description, response type, command data and returned data for each command code.

Topic **Page**

2.1 General

The TIRIS Bus Protocol supports four groups of commands (0 to 3):

Group 0	(Codes 0 to 31) are handled directly by the communications task - generally to control the queue or message traffic (2.2).
Group 1	(Codes 32 to 63) are passed to the RFID task and normally cause a read or write operation (2.3).
Group 2	(Codes 64 to 95) are passed to the control task - typically to set parameter registers or execute an auxiliary I/O function (2.4).
Group 3	(Codes 96 to 127) are reserved for user add-on tasks.

The Command Codes of Group 0 to 2 are reserved for tasks which are defined or will be defined in the coming software releases. Command Codes of Group 3 can be defined by the customer for application specific tasks.

In general, all commands can be executed as immediate, broadcast or queued tasks. The Response Type that is given for each command should give a recommendation as to which way the command should be executed.

2.2 Communications Task Command Codes

Table 2-1. Command Overview

Command Number		Description	Section
hex	dec		
0x00	0	Send Count of Records	Section 2.2.1
0x01	1	Send Next Record from Queue	Section 2.2.2
0x02	2	Send Record N from Queue	Section 2.2.3
0x03	3	Resend Last Record	Section 2.2.4
0x04	4	Clear Queue	Section 2.2.5
0x05	5	Reserved	
. .		. .	
0x1F	31	Reserved	

2.2.1 Send Count of Records

Command Code: $0x00_{hex}$ 0_{dec}

Description: This command returns the number of records, which are buffered in the queue.

Response Type: Immediate

Command Data: none

Returned Data: unsigned char queued_msg_counter

Data Field Index	Content	Description
0	queued_msg_counter	

2.2.2 Send Next Record from Queue

Command Code: $0x01_{hex}$ 1_{dec}

Description: This command returns the next message buffered in the queue.

Response Type: Immediate

Command Data: none

Returned Data: The Command Code in the response of a queued message can be used as a reference. Please refer to the RFID-Task Command Codes and Control-Task Command Codes for a detailed description of the returned message.

2.2.3 Send Record N from Queue

Command Code: $0x02_{hex}$ 2_{dec}

Description: This command returns the Nth message buffered in the queue.

Response Type: Immediate

Command Data: unsigned char record

Data Field Index	Content	Description
0	record	Nth buffered message

Returned Data: The Command Code in the response of a queued message can be used as a reference. Refer to the RFID or Control-Task Command Codes for detailed description of the returned message.

2.2.4 Resend Last Record

Command Code: $0x03_{hex}$ 3_{dec}

Description: This command resends the last sent message.

Response Type: Immediate

Command Data: none

Returned Data: The Command Code in the response of a queued message can be used as a reference. Refer to the RFID-Task Command Codes and Control-Task Command Codes for detailed description of the returned message.

2.2.5 Clear Queue

Command Code: 0x04 $_{hex}$ 4 $_{dec}$
Description: This command clears the messages buffered in the queue.
Response Type: Immediate
Command Data: none
Returned Data: none

2.3 RFID Task Command Codes

Table 2-2 provides a list of the available RFID commands and Table 2-3 gives an overview of the status codes which are returned as a result of the transponder read, write and lock commands.

Table 2-2. Overview of Read, Write and Lock Commands

Command Number		Description	Section
hex	dec		
0x20	32	Charge only read	Section 2.3.1
0x21	33	Read page N	Section 2.3.2
0x22	34	Read page N (80 bit)	Section 2.3.3
0x23	35	Selective Read	Section 2.3.4
0x24	36	Reserved	
0x25	37	Reserved	
0x26	38	Reserved	
0x27	39	Reserved	
0x28	40	Reserved	
0x29	41	Reserved	
0x2A	42	Reserved	
0x2B	43	Program 64 bit	Section 2.3.5
0x2C	44	Program page N	Section 2.3.6
0x2D	45	Program page N (80 bit)	Section 2.3.7
0x2E	46	Selective Program	Section 2.3.8
0x2F	47	Program 80 bit to R/W Transponder	Section 2.3.9
0x30	48	Reserved	
0x31	49	Reserved	
0x32	50	Lock page N	Section 2.3.10
0x33	51	Lock page N (SAMPT)	Section 2.3.11
0x34	52	Reserved	
. .		. .	
0x3F	63	Reserved	

Table 2-3. Status Codes for Read, Program, and Lock Commands

Status Description	Status Value	Comment
RO_TRP	0x00	Successful read of RO transponder
RW_TRP	0x01	Successful read of R/W transponder
MPTCOTRP_U	0x02	Successful read of page 1 (unlocked) of an MPT as a result of a charge-only read
MPTCOTRP_L	0x03	Successful read of page 1 (locked) of an MPT as a result of a charge-only read
MPTRP_U	0x04	Successful read of an unlocked page of an MPT as a result of a read command
MPTRP_L	0x05	Successful read of a locked page of an MPT as a result of a read command
MPTRP_80_U	0x06	Successful read of an unlocked page of an MPT as a result of a read command in 80 bit mode
MPTRP_80_L	0x07	Successful read of a locked page of an MPT as a result of a read command in 80 bit mode
RO_TRP_80	0x08	Successful read of a RO transponder in 80 bit mode
RW_TRP_80	0x09	Successful read of a R/W transponder in 80 bit mode
PROG_OK	0x30	Successful writing to a R/W transponder or MPT
LOCK_OK	0x31	Successful locking of an MPT
NO_READ	0x40	No transponder data received
INCOMPLETE	0x41	Transponder startbyte detected, but CRC check failed
MPTERR_CRC	0x42	MPT Frame CRC is O.K, but Data CRC check failed
MPTRERR_STATUS	0x43	Invalid status returned reading an MPT
MPTRERR_PAGE_U	0x44	Read unlocked page of an MPT, but it is different to the requested page
MPTRERR_PAGE_L	0x45	Read locked page of an MPT, but it is different to the requested page
MPTRERR_SPC_DATA	0x46	Special Data status returned reading an MPT
MPTPERR_STATUS	0x47	Invalid status returned writing an MPT
PERR_FALSE_ID	0x48	Different ID returned from writing a R/W transponder or MPT
MPTPERR_LOW_VOLT	0x49	Programming voltage is too low to write an MPT
MPTPERR_UNREL	0x4A	Writing to an MPT is not reliable
MPTPERR_LOCK	0x4B	Attempt to write a locked page of an MPT
MPTPERR_SPC_DATA	0x4C	Received Special Data status on writing an MPT
MPTPERR_PAGE_U	0x4D	Read unlocked page of an MPT as a result of writing an MPT, but it is different to the requested page
MPTPERR_PAGE_L	0x4E	Read locked page of an MPT as a result of writing an MPT, but it is different to the requested page
MPTLERR_STATUS	0x4F	Invalid status returned on locking an MPT page
MPTLERR_FS_DROP	0x50	Field strength dropped while locking an MPT page
MPTLERR_UNREL	0x51	Locking of an MPT page is not reliable
MPTLERR_PAGE_U	0x52	Read unlocked page of an MPT as a result of locking an MPT, but it is different to the requested page
MPTLERR_PAGE_L	0x53	Read locked page of an MPT as a result of locking an MPT, but it is different to the requested page

2.3.1 Charge Only Read

Command Code: 0x20 $_{hex}$ 32 $_{dec}$

Description: Perform a single charge only read.

Response Type: Immediate/Broadcast/Queued

Command Data: none

Returned Data: output = status,{trp_id_64bit}
unsigned char status
unsigned char trp_id_64bit[8]

Data Field Index	Content	Description
0	status	Read status
1-8	trp_id_64bit[8]	(LSB at Data Field Index 1)

Depending on the returned status code, the data field length of this command varies. Possible status codes with the corresponding data field length are:

Data Field Length	Status Code	Note
1	NO_READ INCOMPLETE MPTRERR_SPC_DATA MPTRERR_STATUS	See Table 2-3.
9	RO_TRP RW_TRP MPTCOTRP MPTCOTRP	Data fields 1 to 8 contain the read Transponder ID

2.3.2 Read Page N

Command Code: 0x21 $_{hex}$ 33 $_{dec}$

Description: Single read of specified MPT page.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char page_nr range: 0x01 .. 0x3F

Data Field Index	Content	Description
0	page_nr	MPT page number to be read

Returned Data: output = status,{trp_id_64bit, {page_nr}}
unsigned char status
unsigned char page_nr
unsigned char trp_id_64bit[8]

Data Field Index	Content	Description
0	status	Read status
1-8	trp_id_64bit	LSB at Data Field Index 1
9	page_nr	Read MPT page number

Depending on the returned status code, the data field length of this command varies. Possible status codes with the corresponding data field length are:

Data Field Length	Status Code	Note
1	NO_READ INCOMPLETE MPTRERR_SPC_DATA MPTRERR_STATUS	See Table 2-3.
9	RO_TRP RW_TRP	Data fields 1 to 8 contain the read Transponder ID.
10	MPTRP_U MPTRP_L MPTRERR_PAGE_U MPTRERR_PAGE_L	Data fields 1 to 8 contain the read Transponder ID. Data field element 9 contains the read page number.

2.3.3 Read Page N (80 bit)

Command Code: 0x22 $_{hex}$ 34 $_{dec}$

Description: Single read of specified MPT page with 80 bit. If page_nr is in the range of 0x01 to 0x3F, a general read of a Multipage Transponder is executed.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char page_nr range: 0x01 .. 0x3F

Data Field Index	Content	Description
0	page_nr	MPT page number to be read

Returned Data: output = status,{trp_id_80bit,{page_nr}}
unsigned char status
unsigned char trp_id_80bit[10]
unsigned char page_nr

Data Field Index	Content	Description
0	status	Read status
1-10	trp_id_80bit	LSB at Data Field Index 1
11	page_nr	Read MPT page number

Depending on the returned status code, the data field length of this command varies. Possible status codes with the corresponding data field length are:

Data Field Length	Status Code	Note
1	NO_READ INCOMPLETE MPTRERR_SPC_DATA MPTRERR_STATUS	See Table 2-3.
11	RO_TRP RW_TRP RO_TRP_80 RW_TRP_80	Data fields 1 to 10 contain the read Transponder ID.
12	MPTRP_U MPTRP_L MPTRERR_PAGE_U MPTRERR_PAGE_L MPTRP_80_U MPTRP_80_L	Data fields 1 to 10 contain the read Transponder ID. Data field element 11 contains the read page number.

2.3.4 Selective Read

Command Code: 0x23 $_{hex}$ 35 $_{dec}$

Description: Selective read of a specified page of a SAMPT.

Response Type: Immediate/Broadcast/Queued

Definitions
sampt_type:
0x00 = 8 bit SAMPT
0x01 = 16 bit SAMPT
0x02 = 24 bit SAMPT
0x03 = 32 bit SAMPT

Command Data:
unsigned char sampt_type range: 0x00 .. 0x03
unsigned char page_nr range: 0x01 .. 0x3F
unsigned char sampt_address 8 bit SAMPT
unsigned char sampt_address[2] 16 bit SAMPT
unsigned char sampt_address[3] 24 bit SAMPT
unsigned char sampt_address[4] 32 bit SAMPT

Data Field Index	Content	Description
0	sampt_type	Specifies SAMPT type. 8, 16, 24 or 32 bit SAMPT address length.
1	page_nr	MPT page number to be read
2 {2-3} {2-4} {2-5}	sampt_address	LSB at Data Field Index 2

Returned Data:
output = status, {trp_id_64bit, {page_nr}}
unsigned char status
unsigned char trp_id_64bit[8]
unsigned char page_nr

Data Field Index	Content	Description
0	status	Read status
1-8	trp_id_64bit	LSB at Data Field Index 1
9	page_nr	Read SAMPT page

Depending on the returned status code, the data field length of this command varies. Possible status codes with the corresponding data field length are:

Data Field Length	Status Code	Note
1	NO_READ INCOMPLETE MPTRERR_SPC_DATA MPTRERR_STATUS	See Table 2-3.
9	RO_TRP RW_TRP	Data fields 1 to 8 contain the read Transponder ID.
10	MPTRP_U MPTRP_L MPTRERR_PAGE_U MPTRERR_PAGE_L	Data fields 1 to 8 contain the read Transponder ID. Data field element 9 contains the read page number.

2.3.5 Program 64-Bit Transponder

Command Code: $0x2B_{hex}$ 43_{dec}

Description: Writes a 64-bit ID to a R/W transponder.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char trp_id_64bit[8]

Data Field Index	Content	Description
0-7	trp_id_64bit	LSB at Data Field Index 0

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Programming status

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	PROG_OK PERR_FALSE_ID NO_READ INCOMPLETE RO_TRP	See Table 2-3.

2.3.6 Program Page N

Command Code: 0x2C $_{hex}$ 44 $_{dec}$

Description: Write to the Nth page of a MPT.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char page_nr range: 0x01 .. 0x3F
unsigned char trp_id_64bit[8]

Data Field Index	Content	Description
0	page_nr	Write MPT page number
1-8	trp_id_64bit	LSB at Data Field Index 1

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Programming result

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	PROG_OK PERR_FALSE_ID NO_READ INCOMPLETE MPTPERR_STATUS MPTPERR_UNREL MPTPERR_PAGE_U MPTPERR_PAGE_L MPTPERR_SPC_DATA MPTPERR_LOW_VOLT MPTPERR_LOCK RO_TRP RW_TRP	See Table 2-3.

2.3.7 Program Page N (80 bit)

Command Code: 0x2D $_{hex}$ 45 $_{dec}$

Description: Write to the Nth page of a MPT.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char page_nr
unsigned char trp_id_80bit[10]

Data Field Index	Content	Description
0	page_nr	Write MPT page number
1-10	trp_id_80bit	LSB at Data Field Index 1

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Programming result

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	PROG_OK PERR_FALSE_ID NO_READ INCOMPLETE MPTPERR_STATUS MPTPERR_UNREL MPTPERR_PAGE_U MPTPERR_PAGE_L MPTPERR_SPC_DATA MPTPERR_LOW_VOLT MPTPERR_LOCK RO_TRP RW_TRP	See Table 2-3.

2.3.8 Selective Program

Command Code: 0x2E $_{hex}$ 46 $_{dec}$

Description: Selective program of a specified page of a SAMPT.

Response Type: Immediate/Broadcast/Queued

Definitions: sampt_type:
0x00 = 8 bit SAMPT
0x01 = 16 bit SAMPT
0x02 = 24 bit SAMPT
0x03 = 32 bit SAMPT

Command Data:
unsigned char sampt_type range: 0x00 .. 0x03
unsigned char page_nr range: 0x01 .. 0x3F
unsigned char sampt_address 8 bit SAMPT
unsigned char sampt_address[2] 16 bit SAMPT
unsigned char sampt_address[3] 24 bit SAMPT
unsigned char sampt_address[4] 32 bit SAMPT

Data Field Index	Content	Description
0	sampt_type	Specifies SAMPT type: 8, 16, 24 or 32 bit SAMPT address length
1	page_nr	Page number to write to
2 {2-3} {2-4} {2-5}	sampt_ address	8 bit SAMPT address 16 bit SAMPT address 24 bit SAMPT address 32 bit SAMPT address
3, 4, 5, or 6 to 12, 13, 14, 15	trp_id_64bit[8]	Transponder ID (LSB first)

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Programming result

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	PROG_OK PERR_FALSE_ID NO_READ INCOMPLETE MPTPERR_STATUS MPTPERR_UNREL MPTPERR_PAGE_U MPTPERR_PAGE_L MPTPERR_SPC_DATA MPTPERR_LOW_VOLT MPTPERR_LOCK RO_TRP RW_TRP	See Table 2-3.

2.3.9 Program 80-bit to R/W Transponder

Command Code: 0x2F $_{hex}$ 47 $_{dec}$

Description: Writes 80-bit data to a R/W transponder. Even if the programming was successful, it is not indicated with the green LED.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char trp_id_80bit[10]

Data Field Index	Content	Description
0-9	trp_id_ 80bit	LSB at Data Field Index 0

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Programming status

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	PROG_OK PERR_FALSE_ID NO_READ INCOMPLETE RO_TRP	See Table 2-3.

2.3.10 Lock Page N

Command Code: 0x32 $_{hex}$ 50 $_{dec}$

Description: Locks the given page of a MPT.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char page_nr range: 0x01 .. 0x3F

Data Field Index	Content	Description
0	page_nr	Page number to be locked

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Locking result

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	MPTLERR_FS_DROP MPTLERR_UNREL MPTLERR_STATUS MPTLERR_PAGE_U MPTLERR_PAGE_L RO_TRP RW_TRP	See Table 2-3.

2.3.11 Selective Lock Page N

Command Code: 0x33 $_{hex}$ 51 $_{dec}$

Description: Selective lock of a specified page of a SAMPT.

Response Type: Immediate/Broadcast/Queued

Definitions: sampt_type:
0x00 = 8 bit SAMPT
0x01 = 16 bit SAMPT
0x02 = 24 bit SAMPT
0x03 = 32 bit SAMPT

Command Data:
unsigned char sampt_type range: 0x00 .. 0x03
unsigned char page_nr range: 0x01 .. 0x3F
unsigned char sampt_address 8 bit SAMPT
unsigned char sampt_address[2] 16 bit SAMPT
unsigned char sampt_address[3] 24 bit SAMPT
unsigned char sampt_address[4] 32 bit SAMPT

Data Field Index	Content	Description
0	sampt_type	Specifies SAMPT type: 8, 16, 24 or 32 bit SAMPT address length
1	page_nr	Page to be locked
2 {2-3} {2-4} {2-5}	sampt_address	8 bit SAMPT address 16 bit SAMPT address 24 bit SAMPT address 32 bit SAMPT address

Returned Data: unsigned char status

Data Field Index	Content	Description
0	status	Locking result

The data field length of this command is always 1. Possible status codes are:

Data Field Length	Status Code	Note
1	MPTLERR_FS_DROP MPTLERR_UNREL MPTLERR_STATUS MPTLERR_PAGE_U MPTLERR_PAGE_L RO_TRP RW_TRP	See Table 2-3.

2.4 Control Task Command Codes

Table 2-4. Command Overview

Command Number		Description	Section
hex	dec		
0x40	64	Get Version	Section 2.4.1
0x41	65	Set RF parameter	Section 2.4.2
0x42	66	Get RF parameter	Section 2.4.3
0x43	67	Set Antenna	Section 2.4.4
0x44	68	Get Antenna	Section 2.4.5
0x45	69	Write EEPROM	Section 2.4.6
0x46	70	Read EEPROM	Section 2.4.7
0x47	71	Reserved	
0x48	72	Reserved	
0x49	73	Read byte from Input	Section 2.4.8
0x4A	74	Reserved	
0x4B	75	Reserved	
0x4C	76	Reserved	
0x4D	77	Write byte to output	Section 2.4.9
0x4E	78	Reserved	
0x4F	79	Reserved	
0x50	80	Reserved	
0x51	81	Reserved	
0x52	82	Reserved	
0x53	83	Reserved	
0x54	84	Reserved	
0x55	85	Get Count of Buffered ID Records	Section 2.4.10
0x56	86	Set Read Mode	Section 2.4.11
0x57	87	Delete ID Records	Section 2.4.12
0x58	88	Get Read Mode	Section 2.4.13
0x59	89	Reserved	
0x5A	90	Get ID Records from the ID Buffer	Section 2.4.14
0x5B	91	Put ID Records to the ID Buffer	Section 2.4.15
0x5C	92	Reserved	
0x5D	93	Read Block of Data from the Memory	Section 2.4.16
0x5E	94	Write Block of Data to the Memory	Section 2.4.17
0x5F	95	Reset Reader	Section 2.4.18

2.4.1 Get Version

Command Code: 0x40 $_{hex}$ 64 $_{dec}$

Description: Returns a string of characters which contain the Reader S/W version. The length of the string is returned in the Data-Field- Length byte of the response.
The string 'S2y00 - TBP 1.x' is returned (x = updates, y = reader version).

Response Type: Immediate

Command Data: none

Returned Data: unsigned char version_string[]

Data Field Index	Content	Description
0 .. strlen-1	version_string	

2.4.2 Set RF Parameter

Command Code: 0x41 $_{hex}$ 65 $_{dec}$

Description: This command sets the RF parameter charge period and duty cycle pause, temporarily, until the reader is either reset or powered-up again.

Response Type: Immediate/Broadcast

Command Data: unsigned char charge_period range: 0x0F .. 0xFF
unsigned int duty_cycle_pause range: 0x0000 .. 0x3FFF

Data Field Index	Content	Description
0	charge_period	Charge-up period in ms
1-2	duty_cycle_pause	Duty-cycle pause in ms (LSB at data field index 1)

Returned Data: none

2.4.3 Get RF Parameter

Command Code: 0x42 $_{hex}$ 66 $_{dec}$

Description: Get the current setting of the RF parameter charge period and duty cycle pause.

Response Type: Immediate/Broadcast

Command Data: none

Returned Data: unsigned char charge_period range: 0x0F .. 0xFF
unsigned int duty_cycle_pause range: 0x0000 .. 0x3FFF

Data Field Index	Content	Description
0	charge_period	charge up period in ms
1-2	duty_cycle_pause	duty cycle pause in ms (LSB at data field index 1)

2.4.4 Set Antenna

Command Code: 0x43 $_{hex}$ 67 $_{dec}$
This command is only valid for readers with the S2000 standard RFM (RI-RFM-104B).

Description: Select the receive antenna of the RFM. If the parameter is 0, receive antenna 0 is selected, if the parameter is 1, receive antenna 1 is selected. All other values between 2 and 255 are reserved and return 'Parameter - Error' in the response code and the Error flag is set.

Response Type: Immediate/Broadcast

Command Data: unsigned char antenna range: 0x00 .. 0x01

Data Field Index	Content	Description
0	antenna	

Returned Data: none

2.4.5 Get Antenna

Command Code: 0x44 $_{hex}$ 68 $_{dec}$
This command is only valid for readers with the S2000 standard RFM (RI-RFM-104B).

Description: Get the current setting of the receive antenna of the RFM.

Response Type: Immediate/Broadcast

Command Data: unsigned char antenna range: 0x00 .. 0x01

Data Field Index	Content	Description
0	antenna	

Returned Data: unsigned char antenna

Data Field Index	Content	Description
0	antenna	

2.4.6 Write EEPROM

Command Code: 0x45 $_{hex}$ 69 $_{dec}$

Description: Write one or more bytes to the reader EEPROM.

> **CAUTION**
>
> The EEPROM addresses 0x00 to 0x5F keep the configuration information of the reader. If these EEPROM locations are accidentally programmed, the RF-Tasks of the Reader may not work. The EEPROM addresses 0x60 to 0x7F can be used for general purposes without any effect on the reader.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char eeprom_startaddress range: 0x00 .. 0x7F
unsigned char eeprom_data[]

Data Field Index	Content	Description
0	eeprom_startaddress	EEPROM startaddress
1 ..	eeprom_data	EEPROM data

Returned Data: none

2.4.7 Read EEPROM

Command Code: 0x46 $_{hex}$ 70 $_{dec}$

Description: Read one or more bytes from the reader EEPROM.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char eeprom_startaddress range: 0x00 .. 0x7F
unsigned char num_bytes range: 0x01 .. 0x80
unsigned char eeprom_data[]

Data Field Index	Content	Description
0	eeprom_startaddress	EEPROM startaddress
1	num_byte	Number of bytes to read

Returned Data: unsigned char eeprom_data []

Data Field Index	Content	Description
0 ..	eeprom_data	Read eeprom data

2.4.8 Read Byte from Input

Command Code: 0x49 $_{hex}$ 73 $_{dec}$

Description: Read the status of one of the reader I/O ports which is specified by the port identifier (port_id).

Response Type: Immediate/Broadcast

Definitions: port_id:
0x00 = Open Collector Outputs
0x01 = I/O lines 0 to 7
0x02 = I/O lines 0 to 3
0x03 = I/O lines 4 to 7
0x04 = Input lines 0/1

Command Data: unsigned char port_id

Data Field Index	Content	Description
0	port_id	Port identifier

Returned Data: unsigned char port_value

Data Field Index	Content	Description
0	port_value	port_id: 0x00 = Open Collector Outputs port value range 0x00 .. 0x03 0x01 = I/O lines 0 to 7 port value range: 0x00 .. 0xFF 0x02 = I/O lines 0 to 3 port value range: 0x00 .. 0x0F 0x03 = I/O lines 4 to 7 port value range: 0x00 .. 0x0F 0x04 = Input lines 0/1 port value ange: 0x00 .. 0x03

2.4.9 Write Byte to Output

Command Code: 0x4D $_{hex}$ 77 $_{dec}$

Description: Write a byte to the output port of the Reader which is specified by the port identifier (port_id). The output value is the result of the logical operation of the current port setting and the parameter passed to the function.

Response Type: Immediate/Broadcast

Definitions: port_id:

0x00 = Open Collector Outputs port value range: 0x00 .. 0x03
0x01 = I/O lines 0 to 7 port value range: 0x00 .. 0xFF
0x02 = I/O lines 0 to 3 port value range: 0x00 .. 0x0F
0x03 = I/O lines 4 to 7 port value range: 0x00 .. 0x0F

logical operation:

0x00 = write port value direct to the output
0x01 = read status ANDed with port value
0x02 = read status ORed with port value
0x03 = read status XORed with port value
0x04 = invert current output setting

Command Data: unsigned char port_id range: 0x00 to 0x03
unsigned char port_value
unsigned char logical_operation range: 0x00 to 0x04

Data Field Index	Content	Description
0	port_id	Port identifier
1	port_value	
2	logical_operation	

Returned Data: none

2.4.10 Get Count of Buffered ID Records

Command Code: 0x55 $_{hex}$ 85 $_{dec}$

Description: This function returns the count of buffered ID Records.

Response Type: Immediate

Command Data: none

Returned Data: unsigned char id_counter_LSB
unsigned char id_counter_MSB
id_counter range: 0x0000 to 0x0400

Data Field Index	Content	Description
0	id_counter_LSB	
1	id_counter_MSB	

2.4.11 Set Read Mode Command

Command Code: 0x56 $_{hex}$ 86 $_{dec}$

Description: This function sets the Read Mode.

Response Type: Immediate/Broadcast/Queued

Command Data: unsigned char read_mode
0x00 = Idle Mode
0x01 = Gate Mode
0x02 .. 0xFF = Reserved

Data Field Index	Content	Description
0	read_mode	

Returned Data: none

2.4.11.1 Gate Mode Description

Gate Mode is a charge only function to read transponders continuously. It supports RO, R/W and MPT (page 1 only) transponders to overcome the disadvantage of having to trigger each read with a single TBP command.

General Settings:

ID record capacity:	1024
ID record size:	9 bytes (8 bytes transponder ID plus one byte transponder type)

2.4.12 Delete ID Records

Command Code: 0x57 $_{hex}$ 87 $_{dec}$

Description: This function deletes a given number of ID records from the ID memory. Two parameters are required to specify the start record and the number of records to be deleted. This firmware starts at zero regardless of the value it receives. For future releases it is planned to accept a start record above zero.
If a value for delete_n which is larger than the number of buffered records is passed, all ID records will be deleted.
The example below shows the ID buffer keeping 30 Identifications before and after the delete operation of 5 ID records (the IDX column shows the number in hex).

Buffer content

before the delete operation

Idx.	ID
1E	3030303030303030
1D	2929292929292929
1C	2828282828282828
1B	2727272727272727
1A	2626262626262626
19	2525252525252525
18	2424242424242424
17	2323232323232323
:	:
:	:
:	:
:	:
09	0909090909090909
08	0808080808080808
07	0707070707070707
06	0606060606060606
05	0505050505050505
04	0404040404040404
03	0303030303030303
02	0202020202020202
01	0101010101010101 ← start

after the delete operation

Idx.	ID
19	3030303030303030
18	2929292929292929
17	2828282828282828
16	2727272727272727
15	2626262626262626
14	2525252525252525
13	2424242424242424
12	2323232323232323
:	:
:	:
:	:
:	:
04	0909090909090909
03	0808080808080808
02	0707070707070707
01	0606060606060606 ← start

Response Type: Immediate/Broadcast

Command Data:

unsigned char	start_rec_LSB
unsigned char	start_rec_MSB
unsigned char	delete_n_LSB
unsigned char	delete_n_MSB

start_rec range: 0x0000 to 0x0400
delete_n range: 0x0001 to 0x0400

Data Field Index	Content	Description
0	start_rec_LSB	Specifies the start record
1	start_rec_MSB	
2	delete_n_LSB	Specifies the number of ID records to be deleted
3	delete_n_MSB	

Returned Data: none

2.4.13 Get Read Mode Command

Command Code: $0x58_{hex}$ 88_{dec}

Description: This function returns the currently selected Read Mode.

Response Type: Immediate

Command Data: none

Returned Data: unsigned char read_mode
0x00 = Idle Mode
0x01 = Gate Mode
0x02 .. 0xFF = Reserved

Data Field Index	Content	Description
0	read_mode	

2.4.14 Get ID Records from the ID Buffer

Command Code: 0x5A $_{hex}$ 90 $_{dec}$

Description: This function returns a given number of ID records stored in the ID Buffer. The command requires three parameters which specify the start record, the number of ID records and the length of an ID record.
If the total size of the ID records requested exceeds the maximum TBP data field length, the maximum number of ID records that fit into the message data field are returned. The actual number of ID records is always returned in data field element zero.
The message data field keeps up to 253 Bytes for ID records. For example: if the ID record length is 9, a maximum number of 28 ID records will fit into one returned TBP message.

Response Type: Immediate

Command Data:

```
unsigned char    start_rec_LSB
unsigned char    char start_rec_MSB
unsigned char    char n_records
unsigned char    char rec_len
                 start_rec range: 0x0000 to 0x0400
```

Data Field Index	Content	Description
0	start_rec_LSB	Specifies the start record LSB
1	start_rec_MSB	Specifies the start record MSB
2	n_records	Specifies the number of records to be returned
3	rec_len	Specifies the record length (should be set to 9)

Returned Data:

```
id_records =
{
unsigned char    trp_type     range: 0x00 to 0x03 (see Table 2-3)
unsigned char    trp_id_64bit[8]
}
unsigned char    rec_count
unsigned char    id_counter_MSB
                 id_records id_rec
```

Data Field Index	Content	Description
0	rec_count	Actual count of ID records
1 .. 9	id_rec 1	First ID record
10 .. 18	id_rec 2	Second ID record
. .	. .	
x .. x+rec_len-1	id_rec n	nth ID record

x = (n-1) x+rec_len+1

2.4.15 Put ID Records to the ID-Buffer

Command Code: 0x5B $_{hex}$ 91 $_{dec}$

Description: This function writes a given number of ID records to the ID buffer. The command requires three parameters which specify the start record, the count of ID records, the length of an ID record followed by the ID records to be written to the buffer. The data field keeps up to 250 Bytes for ID records. For example: if the ID record length is 9, a maximum number of 27 IDs will fit into one passed TBP message.

Response Type: Immediate

Command Data: id_records =

{

unsigned char	trp_type range: 0x00 to 0x03 (see Table 2-3)
unsigned char	trp_id_64bit[8]

}

unsigned char	start_rec_LSB
unsigned char	start_rec_MSB
unsigned char	n_records
unsigned char	rec_len
id_records	id_rec[n_records]
	start_rec range: 0x0000 to 0x0400

Data Field Index	Content	Description
0	start_rec_LSB	Specifies the start record LSB
1	start_rec_MSB	Specifies the start record MSB
2	n_records	Specifies the number of records to be written
3	rec_len	Specifies the record length (should be set to 9)
4 .. 12	id_rec 1	First id_record
10 .. 18	id_rec 2	Second id_record
. .	. .	
x .. x+rec_len-1	id_rec n	nth ID record

x = (n-1) x+rec_len+1

Returned Data: none

2.4.16 Read Block of Data from the Data Memory

Command Code: 0x5D $_{hex}$ 93 $_{dec}$

Description: This function returns a given number of bytes from the data memory (32 kByte of Reader memory). The command requires two parameters which specify the start address and the number of bytes to be read.

Response Type: Immediate

Command Data: unsigned char start_adr_LSB

unsigned char start_adr_MSB

unsigned char n_bytes

start_adr range: 0x0000 to 0x7FFF

Data Field Index	Content	Description
0	start_adr_LSB	Specifies the start address LSB
1	start_adr_MSB	Specifies the start address MSB
2	n_bytes	Specifies the number of bytes to be returned

Returned Data: unsigned char memory_data[n_bytes]

Data Field Index	Content	Description
0 .. (n_bytes-1)	memory_data	Returned memory content

2.4.17 Write Block of Data to the Data Memory

Command Code: 0x5E $_{hex}$ 94 $_{dec}$

Description: This function writes a given number of data records to the data memory (32 kByte of Reader memory).

Response Type: Immediate/Broadcast

Command Data: mem_record =

```
{
unsigned char     start_adr_lsb
unsigned char     start_adr_msb
unsigned char     block_len
unsigned char     data_block[block_len]
}
mem_record        mem_rec[]
                  start_adr range: 0x0000 to 0x0400
```

Data Field Index	Content	Description
0 ..	mem_rec 0	First data record
. .	. .	
.. x	mem_rec n	Last data record

Returned Data: none

2.4.18 Reset Reader

Command Code: 0x5F $_{hex}$ 95 $_{dec}$

Description: Reset the reader unit.

Response Type: Immediate/Broadcast

Command Data: none

Returned Data: none

Timing

This chapter describes the timing that you need to implement or be aware of when using the TIRIS Bus Protocol.

Topic **Page**

3.1 Basic Timing Units

In either master or slave processes, timing is fundamentally dependent on the time taken to transmit each message byte – the Inter-Byte time. This is dictated by the line transmission speed or baud-rate. While the TIRIS Bus Protocol can be used across a range of line speeds, it is assumed in the following sections that a speed of 38400 baud will be used. This speed is a compromise between high throughput and the cost of interface and processing components. We recommend that you do not use baud rates below 9600 baud.

At 38400 baud, each byte of data will require 260.42 μs to be transmitted. In practice, data throughput does not always match the theoretical rate so that an Inter-Byte time of 300 μs will be used to allow at least one additional bit-time for signal delays.

3.2 Timing Constraints

Particularly for response timing, the maximum expected times will depend on the position – in the protocol handler – of the timing check. It is assumed that the check will be made effectively in real-time – close to the UART – so that response times will refer to the time from last byte transmitted to first byte received. If the timing check is performed at a higher level in the protocol prior to an interrupt paced queue; for example, the timing will need to be adjusted to include message transmission time(s). As message length can vary between 8 and 263 bytes, such timing checks will need to calculate the transmission time by multiplying the message length by the Inter-Byte time (300 μs).

3.3 Master Timing

The master is responsible for ensuring that a response is received for each nonbroadcast message. If a response is not received within the expected time, the master will re-transmit the message up to three times before issuing a communications reset and retrying the transmission a further four times.

Continued failure will be reported back to the requesting task. The expected response time is a combination of the time required to reverse the line (Turnaround) and the time required for the slave to process the message and commence transmission of the response.

3.3.1 Master Turnaround Time

The actual, electrical turnaround time of the line and driver/receiver circuitry is insignificant compared to the processing time required to initiate the turnaround. The master should disable its transmitter at the completion of each message for which a response is expected. Before disabling the transmitter, the master should check that the last byte has been completely transmitted, or wait at least one Inter-Byte time. The time required for turnaround this way will be covered by the somewhat larger requirement for slave processing.

When turning around the line in the other direction - when the master transmits after receiving - the master can enable its transmitter immediately but should wait two Inter- Byte times (600 μs) before transmitting to allow the slave to prepare for reception.

3.3.2 Slave Processing Time

The maximum time that the master will wait for a slave's response must be sufficient to permit the slave to process the command. It should on the other hand not be so long that it degrades system throughput.

Processing time for any message consists of both the communications processing time and the actual command execution. To simplify matters, two groups of commands will be considered:

The first group contains all commands which can be executed by the slave without the need for reader activity. These are typically system or communications control functions requiring only software processing. This group also contains reader commands that specify a queued response. The initial response to those commands is simply an acknowledgment of correct receipt.

The second group contains reader commands for immediate execution where the slave will not respond until at least one reader cycle has been performed.

Group 1, or FAST, commands can all be expected to be processed in a standard time. It is considered that a maximum time allowance of 2.4 ms (8 Inter-Byte times) should be sufficient for any of these commands.

Group 2, or SLOW, commands can not easily be generalized. These commands are primarily paced by the number of reader cycles required. The reader cycles also will vary for read and write operations, and be affected by synchronization with other readers. It is suggested, therefore, that SLOW commands are used only when essential as they will seriously degrade the throughput of the system. In practice, the master will need to calculate an expected response time for this group on the basis of each type of command. It would probably be best implemented through the use of a table. It is suggested that the maximum reader cycle time + 3 ms should be an appropriate value.

3.3.3 Incomplete Message (Master)

Once the first byte of a response has been received, the master should wait for up to two Inter-Byte times (600 μs) for each byte within the message. Beyond this time, the message is considered incomplete and the master must request a resend of the response.

3.4 Slave Timing

The slave's responsibilities for timing are relatively simple. Apart from ensuring that responses are transmitted within the maximum times allowed by the master, only two critical time factors need to be considered as described below.

3.4.1 Slave Turnaround Time

The slave should only enable its transmitter when it is about to transmit a response and should disable it once the transmission is complete. To ensure that the master is ready for reception, the slave should ensure that at least two Inter-Byte times (600 μs) have elapsed since the last byte was received.

3.4.2 Incomplete Message (Slave)

In the same way as the master, once the first byte of a message has been received, the slave should wait for up to two Inter-Byte times (600 μs) for each byte within the message. Beyond this time, the message is considered incomplete and the slave must issue an error response to the master. Prior to transmitting the response, however, the slave should wait a further Inter-Byte time and check for line activity. If further data is received, the slave should ignore the previous incomplete message and continue to monitor the line for incoming messages.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265