TEXAS INSTRUMENTS

Tag-it™ Reader System Series 6000

Host Protocol

Reference Manual

11-04-21-001 July 1999

TIRIS Technology by Texas Instruments™

Edition Two - July 1999

This is the second edition of this manual for the **Tag-it Reader Host Protocol**.
It describes the firmware command set as implemented with release 2 and beyond (Hardware part number with suffix ‘-01’ or greater).

Texas Instruments (TI) reserves the right to make changes to its products or services or to discontinue any product or service at any time without notice. TI provides customer assistance in various technical areas, but does not have full access to data concerning the use and applications of customer’s products

Therefore, TI assumes no liability and is not responsible for customer applications or product or software design or performance relating to systems or applications incorporating TI products. In addition, TI assumes no liability and is not responsible for infringement of patents and/or any other intellectual or industrial property rights of third parties, which may result from assistance provided by TI.

TI products are not designed, intended, authorized or warranted to be suitable for life support applications or any other life critical applications which could involve potential risk of death, personal injury or severe property or environmental damage.

Preface

Read This First

About This Manual

This reference manual describes the Tag-it host to reader protocol (or Tag-it Host Protocol) and its layout. It is designed for use by TI customers who are engineers experienced with TIRIS and radio frequency identification (RFID) components, using this document for the purpose of integrating TIRIS reader products into customer application systems.

Conventions

The following pictograms and designations are used in the operating instructions:

⚠	**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 needed could cause permanent damage to the system.**
☞	**Note:** Indicates conditions which must be met, or procedures which must be followed, to ensure proper functioning.
ℹ	**Information:** Indicates conditions or procedures that should be followed to ensure proper functioning of the system.

If You Need Assistance

Application Centers are located in Europe, North and South America, the Far East and Australia to provide direct engineering 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.

Numerical Representations

Unless otherwise noted, numbers are represented as decimal.

Hexadecimal numbers are represented with the suffix $_{16}$, e.g. $A5F1_{16}$

Binary numbers are represented with the suffix $_2$, e.g. 1011_2

Byte representations: the least significant bit (lsb) is bit 0 and the most significant bit (msb) is bit 7.

Document Overview

Chapter 1: Introduction 7
1.1 Description 8
1.2 Command Structure 11
Chapter 2: Frame Request/Response Protocol 12
2.1 General 13
2.2 Request/Response Frame Block Definitions 14
2.2.1 Start Code 14
2.2.2 Data Length 14
2.2.3 Service Code 14
2.2.4 Control Flag 17
2.2.5 Status Flag 19
2.2.6 Message 21
2.2.7 BCC 21
Chapter 3: Message Structure 22
3.1 General 23
3.1.1 Transponder Protocol Pass-Through 24
3.1.2 Single Message 25
3.1.3 Compound Message 28
3.1.4 Ancillary Message 30
3.2 Flow Control 30
3.3 Data Format / Orientation 31
3.4 Host Protocol Timing Parameters 32
3.5 Error Detection 33
Chapter 4: Service and Error Codes 34
4.1 General 35
4.2 Service Code Listing 38
4.3 Error Code Listing 39
Chapter 5: List of Commands 40
5.1 Tag Version 41
5.2 Read Block 43
5.3 Write Block 45
5.4 Write and Lock Block 47
5.5 Lock Block 49
5.6 Read Multiblock 51
5.7 Write Multiblock 54
5.8 Lock Multiblock 56
5.9 Read Block SID 58
5.10 Read Multiblock SID 62
5.11 Repeat Last Request 67

5.12 Send Last Response 69
5.13 Stop Continuous 71
5.14 Start Synch 73
5.15 Reset Reader 74
5.16 Reader Version 76
5.17 Reader Diagnostic 78
5.18 Read Reader Setup 80
5.19 Start Flash Loader 82
5.20 Write SID Code 83
5.21 Factory Lock Block 85
5.22 Factory Programming Off 87

Appendix A: BCC Error Detection Methods 89

Appendix B: Reader Synchronization 90

Appendix C: Reader Setup Parameters 92

Appendix D: Firmware Programming 93

Appendix E: Glossary 94

Chapter 1

Introduction

Topic **Page**

1.1 Description 8
1.2 Command Structure 11

1.1 Description

The Tag-it™ system is one of Texas Instruments' TIRIS (Texas Instruments Registration and Identification Solutions) families of radio frequency identification (RFID) products. The Tag-it™ system consists of a reader, connected to a host (or supervisory) system, which interrogates Tag-it™ transponders when present in the reader's antenna field.

The Tag-it Host Protocol defines the communication between the host system and the Tag-it readers provided by Texas Instruments. A second protocol, the Tag-it Transponder Protocol, defines communication between the reader and the Tag-it transponders present in the reader's active field and is described in a separate reference manual.

Derived from industry-standard protocols, both protocols have been designed to be efficient and flexible, while still remaining simple to implement. A layered approach allows for implementation of compound commands for performance optimization.

The Host Protocol performs two main functions:

- Handling of data communication between host and reader.
- Carrying of requests for commands and responses to those commands from transponders.

Other services that the Host Protocol provides include:

- Management of the reader functionality, e.g. reader software version reporting
- Enhancement of basic transponder functionality by requesting execution of compound commands.

The Host Protocol provides the means of controlling the Tag-it™ reader through the host system, which may consist of a microcontroller, a personal computer, or a programmable logic controller.

The Tag-it™ reader supports an asynchronous serial port (ASP) communications interface.

To maximize data throughput, particularly for continuous operation, the Tag-it™ reader does not store large amounts of identification data on-board, and does not handle external inputs or outputs other than connection to the host.

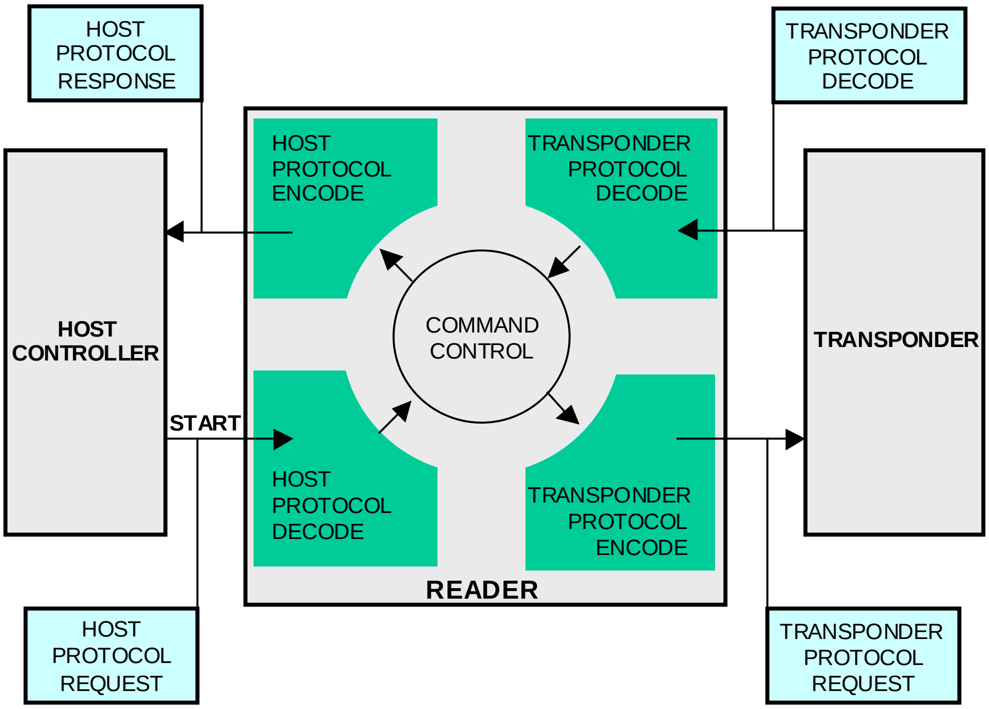
The Host Protocol is designed for point-to-point, half-duplex communications, with the host controller acting as the primary station and the reader as the secondary station.

Note:
If using a modem to communicate between the host system and the Tag-it™ reader, a full-duplex modem should be used to prevent loss of response frames.

The host computer initiates all communications using the Host Protocol. This protocol consists, in most cases, of request/response pairs where the host waits for the response before continuing. Therefore, the host is not presented with large sequences of open requests and complex sequence numbering. To reduce message traffic, messages can be run continuously, producing multiple responses for a single message. The host must be able to buffer and process these responses, interrupting the flow if necessary.

The Host Protocol is a binary, bytecount-oriented protocol. To allow flexibility in controlling the reader, variable length data is passed within a defined frame. The data length is always passed as a parameter within the message, and the possible structures of the data contained are known by the communicating parties.

The following diagram is a view of the data flow through the Tag-it reader. The host controller and transponder activities are external events which the Tag-it reader must react to. The Tag-it reader acts on the data and maintains control to ensure the highest possible data throughput.



1.2 Command Structure

Commands from the host system to the reader contain coded instructions and parameters. Responses contain status information and data.

The reader normally operates in a half-duplex mode. However, if the Auto-repeat mode is switched on (see Auto-repeat flag), the host system may receive responses at unsolicited times, e.g. when sending a request. This must be taken into account when programming the system software and designing the system architecture.

Note:
If using a modem to communicate between the host system and the reader, this should be full duplex to ensure no loss of response frames.

The receiving buffer on the host side should be large enough to accept several response frames while the host system is busy processing other data. It is recommended that a buffer able to store a minimum of 10 x 512-byte responses be provided.

All messages consist of data — the actual message — and a frame. The structure of the message is governed by the service code, which is contained in the header.

A message from the host to the reader is a *request*; conversely a message from reader to the host is a *response*. Requests can only be initiated by the host. A request can receive single or multiple responses, depending on the type of message and the execution mode employed.

.

Chapter 2

Frame Request/Response Protocol

Topic **Page**

2.1 General 13

2.2 Request/Response Frame Block Definitions 14

2.2.1 Start Code 14

2.2.2 Data Length 14

2.2.3 Service Code 14

2.2.4 Control Flag 17

2.2.5 Status Flag 19

2.2.6 Message 21

2.2.7 BCC 21

2.1 General

The data communication between the host system and the reader is performed within a frame structure.

The frame structure is essentially the same for requests (host-to-reader) and responses (reader-to-host), except that the flags have different meanings in the request frame (control flags) and the response frame (status flags).

Request Frame

Block	Start Code	Data Length	Service Code	Control Flag	Message (Data)	BCC
Byte Position	1	2, 3	4	5	6 to (N – 1) where length N ≤ 4091 bytes	N+6, N+7

Response Frame

Block	Start Code	Data Length	Service Code	Status Flag	Message (Data)	BCC
Byte Position	1	2, 3	4	5	6 to (N – 1) where length N ≤ 4091 bytes	N+6, N+7

2.2 Request/Response Frame Block Definitions

2.2.1 Start Code

A standard CSMA/CD Start frame delimiter (SFD) with the value = $D5_{16}$ is used to identify the start of a complete frame.

2.2.2 Data Length

Specifies the number of data bytes that will follow this block (including the BCC). The range is from 4 to 4095. The minimum of 4 bytes includes the Service code (1 byte), the Control flag (1 byte) or Status flag (1 byte), and the BCC (2 bytes).

2.2.3 Service Code

This code is discussed in detail in Chapter 4.

This code indicates one of the four types of commands supported.

01:	Pass-through service
02:	Single service
03:	Compound service
04:	Ancillary service

01 - Pass-Through Service

The reader only passes the information it receives from the host directly on to the transponder without further action, suppressing only stuffing if present. When the reader receives a response from the transponder, it sends it on to the host unchanged except for any bit stuffing (up to 7 bits) performed to ensure it comprises a complete number of bytes. If the reader does not receive a response within a specified period, it sends a response to the host with the error flag set and the corresponding error code. The Pass-through service is useful since it allows the host to have direct control over the transponder. It can also be used to handle new versions of transponders which are not yet fully supported.

Examples of Pass-through commands are:

Read block
Write block
Lock block

Note:
Pass-through commands are not normally used to communicate with the reader. Only single, compound, and ancillary service commands are explained in detail in this document.

This service code should only be used by expert users and transponder developers.

02 Single Service

On receipt of a Single service request the reader analyzes and verifies it, then formats and sends a single command to the transponder.

When the reader then receives a response from the tag, it analyzes and verifies it then formats and sends a unique response to the host.

Examples of Single service commands are:

Read block
Write block
Lock block

03 Compound Service

Compound messages instruct the reader to perform defined actions on transponders using the Transponder Protocol. These messages differ from the Single service transponder messages in that they instruct the reader to perform complex operations involving multiple commands before returning a response to the host.

Compound messages are also used to emulate Transponder Protocol commands which are not implemented by a particular transponder version. An example is a Compound message which emulates Read multiblock using a sequence of Read block commands. In this case, a compound message saves the host from having to initiate the Read block message for each block required.

On receipt of a Compound service request, the reader analyzes and verifies it, then formats and sends one or more commands to the transponder.

When the reader receives one or more responses, it analyses and verifies them, then formats and sends a unique response to the host. This response may be contained in several frames, depending on the size of the response (see More bit).

The Compound service is used to improve system performance. For example, it is much faster to send a single Read three blocks of memory command rather than to send three separate Read block commands.

The Compound service is also used to request the reader to perform complex operations such as simultaneous identification (SID) of the transponders present in the reader antenna’s field. It is an effective way of offloading the host system from detailed and complex actions.

Examples of Compound service are:

Read block SID
Read multiblock
Write multiblock

☞ **Note:**
Single commands can be emulated by the reader similarly to Compound commands if the reader knows that the target transponder cannot execute the requested command. For example: if the host requests a “Read multiblock” and the target transponder supports only “Read (single) block”, the reader turns this into a sequence of several Read block commands. This capability is authorized by the emulation flag in the Request control flags field.

04 Ancillary Service

On receipt of an Ancillary service request, the reader analyzes and verifies it, then executes the request locally. No commands are sent to the transponder.

The Ancillary service request can be used to download a new version of the reader software, or to set operating and communications parameters.

Examples of Ancillary service are:

Reader reset
Read reader version
Reader diagnostic

2.2.4 Control Flag

Control flags are used in a request frame. In a response frame, the status flags are used (see section 2.2.5).

For control flags, four of the bits in this block are used to specify instructions to the reader or information about the message. The bits are numbered 0 (lsb) to 7 (msb) and the bit functions are as follows.

Control Flag Bit Functions

Bit	Function
0	Reserved
1	More flag
2	Emulation flag
3	Auto repeat flag
4	BCC flag
5-7	Reserved

Bit 1 (More flag):

0 = No more data associated with this request (single message or end of a series)

1 = More data associated with this request is contained in following messages

This More bit control flag is similar to the More bit in the X.25 CCITT recommendation. If it is set, the reader should continue to expect more frames until the host system sends a frame with the More bit reset.

Bit 2 (Emulation flag):

0 = emulate off

1 = emulate on

The emulation bit tells the reader whether it is allowed to emulate a complex request by sending multiple single requests, so that it can substitute a complex command with a compound one.

Emulate on means that if the command cannot be performed by the current transponder version the reader will attempt to use a compound command to complete the request.

Example: The host system requests a Read multiblock, the reader determines (for example by using “Read version”) that the target does not support Read multiblock and also sees that the Emulate bit is set. This results in the reader sending multiple

Read (single) block commands to the transponder, assembling the responses into a single response and sending that response back to the host.

<u>Bit 3 (Auto repeat flag):</u>

0 = Auto Repeat off.

1 = Auto Repeat on.

The Auto repeat flag specifies whether or not the reader should automatically repeat the execution of the request.

With the Auto repeat bit on, the reader will:

a: Execute the request (send a request to the target transponder).

b: Receive the response and send it back to the host system.

c: Wait for a specified time (see Reader parameter settings) and resend the request.

d: Receive the response and send it back to the host system.

e: Carry on repeating steps c & d until it receives another instruction without the Auto repeat bit set, or until the Auto repeat mode is switched off.

<u>Bit 4 (BCC flag):</u>

0 = This message uses CRC-CCIT for BCC

1 = This message uses LRC for BCC

The BCC flag specifies what kind of Block check code is used to ensure this request against communication errors. The reader will use the same method in the response.

The CRC offers much better protection against data errors, but is more complex to implement. The LRC is simpler to implement and requires less processing time at the host, but offers a lower level of protection.

Wherever possible, use the CRC. However, some hosts (for example, PLCs) may not have the capability to support it, in which case the LRC is a useful and simple alternative.

2.2.5 Status Flag

Status flags are used in a response frame. In a request frame, the control flag is used (see section 2.2.4).

This flag describes the result of the request and provides information about the message. Five of the bits in this block are used; the bits being numbered 0 (lsb) to 7 (msb):

Status Flag Bit Functions:

Bit	Function
0	Exception flag
1	More flag
2	Emulation flag
3	Auto repeat flag
4	BCC flag
5-7	Reserved

Bit 0 (Exception flag):

0 = No exception occurred. Request completed OK.

1 = Exception occurred. Status code is contained in the first byte of message.

The Exception flag indicates whether the request was successfully routed to the reader.

Exception Status Codes

Error Code	Definition
00_{16}	Reserved
01_{16}	Request data corrupted, not executed
02_{16}	Application not supported
03_{16}	Data format error, request aborted
04_{16}	Continuous mode not available for this request
05_{16}	Reserved
06_{16}	Reserved
07_{16}	Reserved
08_{16}	Reserved
09_{16}	Reserved
$0A_{16}$	Reserved
$0B_{16}$	Reserved
$0C_{16}$	Reserved
$0D_{16}$	Reserved
$0E_{16}$	Reserved
$0F_{16}$	Undefined system error, request aborted

Bit 1 (More flag):

0 =	No more data associated with this request (single message or the end of a sequence).
1 =	More data associated with this request is contained in following frames.

The Data extension flag informs the reader whether more data follows. If set, the reader expects more frames until it receives a frame with the Data extension bit reset.

Bit 2 (Emulation flag):

0 =	Request does not emulate a complex command.
1 =	Request emulates a complex command.

The Emulation flag tells the reader whether the transponder completed the request using a compound command to emulate a complex command.

Bit 3 (Auto-repeat flag):

0 =	Request executed once.
1 =	Request executed continuously.

The Auto-repeat flag informs the reader whether the transponder performed the request once or continuously.

Bit 4 (BCC flag):

0 =	This message uses CRC for BCC.
1 =	This message uses LRC for BCC.

The BCC flag specifies what kind of Block check character method (CRC or LRC) is used.

2.2.6 Message

This part of the request carries the actual message for the transponder. The format depends on the type of command as specified in the Service code byte. See chapter 3 for definitions of message structure.

2.2.7 BCC

This block contains the CRC (or LRC) of all the preceding data, excluding the start block. The Control flag bit 4 indicates whether CRC (Bit=0) or LRC (Bit=1) is used.

The BCC is based on the message that it accompanies. If any other BCC value is read, it indicates that the information read is either corrupted or is not what was expected.

Chapter 3

Message Structure

Topic **Page**

3.1 General 23
3.1.1 Transponder Protocol Pass-Through........ 24
3.1.2 Single Message 25
3.1.3 Compound Message........ 28
3.1.4 Ancillary Message........ 30
3.2 Flow Control........ 30
3.3 Data Format / Orientation 31
3.4 Host Protocol Timing Parameters........ 32
3.5 Error Detection........ 33

3.1 General

The message, starting at byte position 6, is defined as data which is being sent or received. It is encapsulated by a frame consisting of header and BCC.

The message length may be up to 4091 bytes, the actual maximum length being defined specifically for each Service Code.

Only the host can send a request message, the actual message to the transponder. The format depends on the type of command, as specified by the Service Code.

The message from the transponder is called a response. A request may receive single or multiple responses, depending on the type of message and the execution mode used.

The service code signals what type of request is transmitted. (See chapter 4: Service and Error codes)

The request types are defined as:

Transponder protocol pass-through (01)
Single (02)
Compound (03)
Ancillary (04)

3.1.1 Transponder Protocol Pass-Through

Frame	Length	Transponder Protocol Message	Pad	Frame
5 bytes	1 byte	Up to 310 bits	1-7 bits	2 bytes

If the request is designed to pass through a Transponder Protocol message without reformatting, it is called a Transponder protocol pass-through. These are discussed in more detail in the Transponder protocol reference guide, but have the following structure:

Length (1 byte):

Determines the following data length in bytes, which includes the Transponder protocol message and pad.

Transponder Protocol Message:

Since the Transponder Protocol is a bit-oriented protocol, data can cross byte boundaries, with three possible message structures:

Single message
Compound message
Ancillary message

Pad (1-7 bits):

Depending on the Transponder protocol message length, a pad of 1 to 7 bits may be added to the end of the message data to ensure the message is oriented on a byte boundary.

3.1.2 Single Message

Frame	Command	Format	Synch	Message Data	Frame
5 bytes	1 byte	1 byte	1 byte	Up to 4088 bytes	2 bytes

This command type is a formatted Transponder Protocol message. The command code reflects that used by the Transponder Protocol and is the same for the request and response. The Format code contains information about the format and status of the message.

Format Code (Request)

Bit	Name	Value	Description
0 (lsb)	Reserved	-	-
1	Code extension (not currently implemented)	0 1	No code extension (default) Code extension used. First data byte defines extended command code.
2	Addressing	0 1	Not addressed Addressed. Address is contained in the first 4 bytes of data field
3	Format (currently only fixed format)	0 1	Fixed format (default) Variable format
4	Transmitter control	0 1	Turn transmitter off after executing this request Leave transmitter on after executing this request
5	Reserved	-	-
6	Reserved	-	
7 (msb)	Reserved	-	-

Format Code (Response)

Bit	Name	Value	Description
0 (lsb)	Error	0 1	No error. Request completed ok Error occurred. Status code is first byte of data field after any address information
1	Code extension (not currently implemented)	0 1	No code extension (default) Code extension used. First data byte defines extended command code.
2	Addressing	0 1	Not addressed Addressed. Address is contained in the first 4 bytes of data field
3	Format (currently only fixed format)	0 1	Fixed format (default) Variable format
4	Reserved	-	-
5	Reserved	-	-
6	Reserved	-	-
7 (msb)	Reserved	-	-

Messages can be in one of two modes; addressed or non-addressed, although not all messages support both modes. The address is obtained from the transponder by an individual Tag version message when only one transponder is within the field, or during the SID sequence when more than one transponder is within the field.

Addressing allows further operations to be carried out on a single transponder without influencing any other transponders nearby. If used, the 32-bit address is always sent as the first 4 bytes of the message data.

The response to an addressed request always includes the address as verification. Addressed and non-addressed request/response pairs may not be mixed. Thus only an addressed response to an addressed request is valid. Similarly, only a non-addressed response to a non-addressed request is valid.

If the command is addressed, the address is always the first part of the data field.

Synchronization Codes (Request)

Bit	Name	Value	Description
0– 2(lsb)	Synchronization type	000_2 001_2 010_2 011_2	No synchronization Master/Slave synchronization (Master) Master/Slave synchronization (Slave) Cascaded synchronization
3	‘No transponder’ response during continuous mode	0 1	All responses sent to host ‘No transponder’ responses not sent to host.
4	Alternate operation during continuous mode	0 1	All valid data responses sent to host Identical responses only sent once to host
5	Reserved	-	-
6	Reserved	-	
7 (msb)	Reserved	-	-

Synchronization Codes (Response)

Bit	Name	Value	Description
0– 2(lsb)	Synchronization type	000_2 001_2 010_2 011_2	No synchronization Master/slave synchronization (master) Master/slave synchronization (slave) Cascaded synchronization
3	‘No transponder’ response during continuous mode	0 1	All responses sent to host ‘No transponder’ responses not sent to host.
4	Alternate operation during continuous mode	0 1	All valid data responses sent to host Identical responses only sent once to host
5	Reserved	-	-
6	Reserved	-	
7 (msb)	Reserved	-	-

3.1.3 Compound Message

Frame	Command	Format	Synch	Message Data	Frame
5 bytes	1 byte	1 byte	1 byte	Up to 4088 bytes	2 bytes

This command type is used to carry out a predetermined sequence of Transponder Protocol messages, and/or other actions before returning the status to the host. It may also be used to emulate any Transponder Protocol message not supported by the transponder, but which is in the field of the reader. In the latter case, the command code reflects that used by the Transponder Protocol and is the same for the request and response. Otherwise, the command code is assigned from the most significant value downward to avoid possible conflict with future Transponder Protocol message emulation. The Format code contains information about the format and status of the message.

Compound Message Format Code (Request)

Bit	Name	Value	Description
0 (lsb)	Reserved	-	-
1	Reserved	-	-
2	Addressing	0 1	Not addressed Addressed. Address is contained in the first 4 bytes of data field
3	Format (currently only fixed format)	0 1	Fixed format (default) Variable format
4	Transmitter control	0 1	Turn transmitter off after executing this request Leave transmitter on after executing this request
5	Reserved	-	-
6	Reserved	-	-
7 (msb)	Reserved	-	-

Compound Message Format Code (Response)

Bit	Name	Value	Description
0 (lsb)	Error	0 1	No error. Request completed ok Error occurred. Status code is first byte of data field after any address information
1	Reserved	-	-
2	Addressing	0 1	Not addressed Addressed. Address is contained in the first 4 bytes of data field
3	Format (currently only fixed format)	0 1	Fixed format Variable format
4	Reserved	-	-
5	Reserved	-	-
6	Reserved	-	-
7 (msb)	Reserved	-	-

Synchronization Code (Request)

Bit	Name	Value	Description
0– 2(lsb)	Synchronization type	000_2 001_2 010_2 011_2	No synchronization Master/slave synchronization (master) Master/slave synchronization (slave) Cascaded synchronization
3	‘No transponder’ response during continuous mode	0 1	All responses sent to host ‘No transponder’ responses not sent to host.
4	Alternate operation during continuous mode	0 1	All valid data responses sent to host Identical responses only sent once to host
5	Reserved	-	-
6	Reserved	-	
7 (msb)	Reserved	-	-

Synchronization Code (Response)

Bit	Name	Value	Description
0– 2(lsb)	Synchronization type	000_2 001_2 010_2 011_2	No synchronization Master/slave synchronization (master) Master/slave synchronization (slave) Cascaded synchronization
3	‘No transponder’ response during continuous mode	0 1	All responses sent to host ‘No transponder’ responses not sent to host.
4	Alternate operation during continuous mode	0 1	All valid data responses sent to host Identical responses only sent once to host
5	Reserved	-	-
6	Reserved	-	
7 (msb)	Reserved	-	-

3.1.4 Ancillary Message

This command type defines messages used for reader service functions (not involving transponder communication), and also host to reader communications control functions. The most significant bit of the command code is used by the response to indicate the completion status.

Frame	Command	Message Data	Frame
5 bytes	1 byte	Up to 4090 bytes	2 bytes

The least significant 7 bits of the command code normally remain the same for request and response. Exceptions are Repeat_last_request and Send_last_response, which respond with the command code of the last request.

Bit	Name	Value	Description
0 (lsb) - 6	Command code	0 - $7F_{16}$	Request/response code
7 (msb)	Error flag	0 1	Request completed ok Error occurred processing request. First byte of data field details error status

3.2 Flow Control

The method of flow control used is a form of stop-and-wait automatic repeat request. This means that every request requires a response. The response contains an exception flag which, if set, indicates an error. The request may also be terminated by a timeout if no response is received. Since the host always acts as the primary station, it is responsible for controlling the data flow by reacting to responses or regulating the flow of responses from the reader.

The exception flag is returned as 0 if the request was correctly interpreted and action was taken by the reader.

The exception flag is returned as 1 if the request was corrupted, or otherwise not completely received by the reader. In this case a status code is returned in the data field explaining the reason for the exception.

If an exception occurs, the reader takes no action beyond returning a Nack to the host. The host is responsible for managing recovery from the exception, normally by re-sending the original request.

The host must be prepared to accept a response when issuing a request. In the case of continuously executing requests, the host may terminate these at any time if it is unable to process the data flow received.

3.3 Data Format / Orientation

A maximum data length of 4091 bytes is allowed in a single message defined by the 2 byte data length parameters sent with each message. Data lengths may be variable for each message, dependent on the parameters used and/or the amount of response data supplied. The data formats for each possible form are known by the host and the reader.

All bytes are transmitted least significant bit first via the ASP.

Frame Layout

Start frame delimiter	Data length	Service code	Control /status flag	Data (as in command definition, up to 4091 bytes)	BCC

Byte ordering within frame

MS Byte ………LS Byte		MS Byte …………. LS Byte	etc

<-- direction of transmission

Bit ordering for transmission

LSB ………….MSB	LSB ………….MSB	LSB …………MSB	etc

<-- direction of transmission

3.4 Host Protocol Timing Parameters

Note:
The list of timing parameters will be expanded as required.

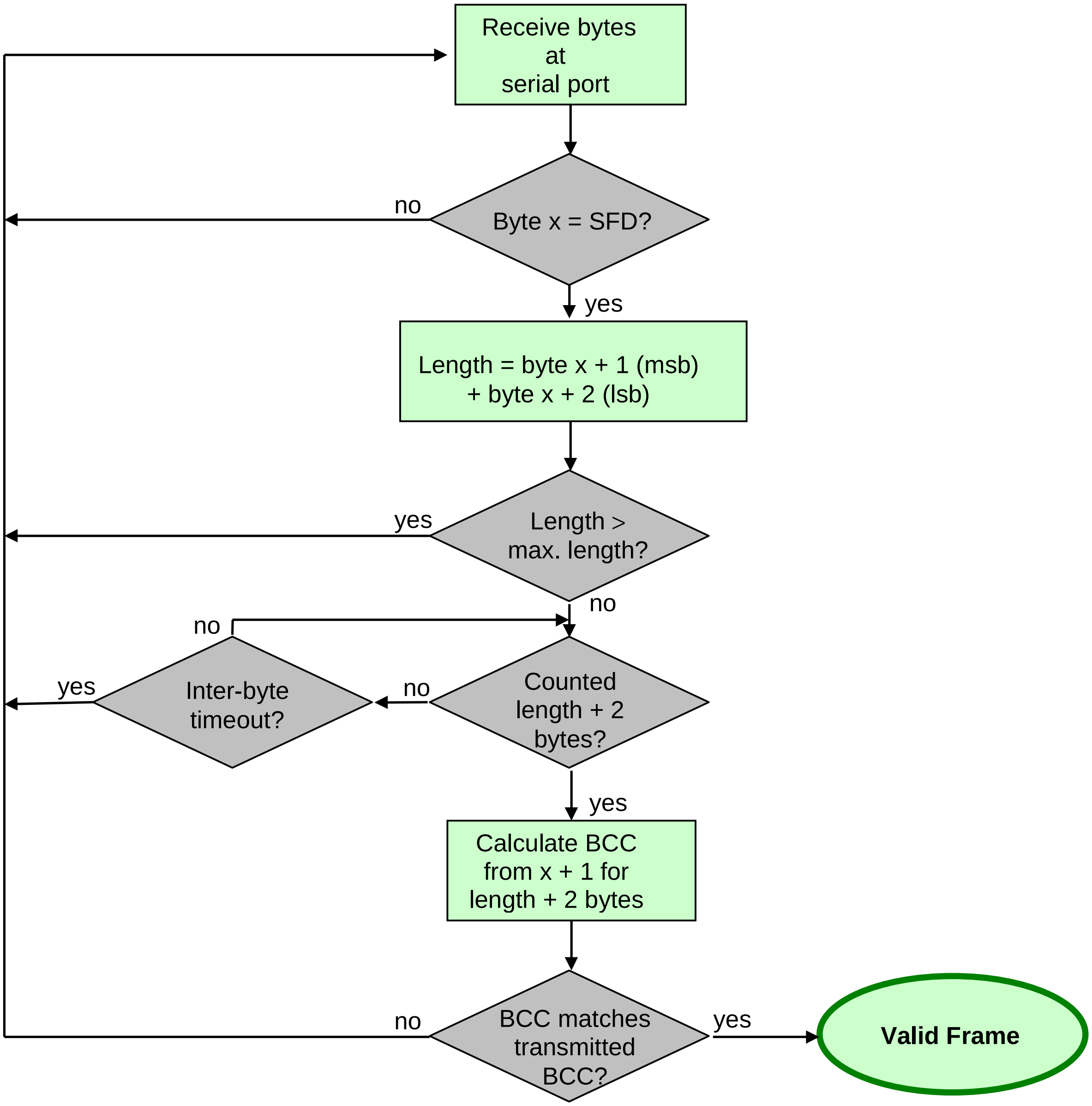
Timing Parameter	Description	Value
Inter-byte ASP	Error flagged when time between bytes received exceeds this value	baudrate dependent
Response time transponder message	Time before response is received at host (excluding continuous mode)	maximum 1s
Response time Compound message	Time before response is received at host (excluding continuous mode)	maximum 10s
Pause time continuous mode	Pause between executing Transponder Protocol commands when executing continuously	adjustable via setup

3.5 Error Detection

A Block check character (BCC) is used for error detection, employed as part of the message frame. The BCC is calculated on all bytes in the frame, excluding the Start frame delimiter (SFD), and is attached as the last two bytes of the message. As such, its purpose is error detection and delimiting of the end of the frame together with the data length supplied in the message

The default (and recommended) method of error detection is the Cyclic redundancy check (CRC). The CRC used is defined as follows:

FRAME VALIDATION


Receive bytes at serial port
Byte x = SFD?
no
yes
Length = byte x + 1 (msb) + byte x + 2 (lsb)
Length > max. length?
yes
no
Counted length + 2 bytes?
no
Inter-byte timeout?
yes
no
yes
Calculate BCC from x + 1 for length + 2 bytes
BCC matches transmitted BCC?
no
yes
Valid Frame

Chapter 4

Service and Error Codes

Topic **Page**

4.1 General 35
4.2 Service Code Listing 38
4.3 Error Code Listing 39

4.1 General

The Service Code indicates one of the four types of commands supported.

01:	Pass-through service
02:	Single service
03:	Compound service
04:	Ancillary service

01 - Pass-Through Service

The reader only passes the information it receives from the host directly on to the transponder without further action, suppressing only stuffing if present. When the reader receives a response from the transponder, it sends it on to the host unchanged except for any bit stuffing (up to 7 bits) performed to ensure it comprises a complete number of bytes. If the reader does not receive a response within a specified period, it sends a response to the host with the error flag set and the corresponding error code. The Pass-through service is useful since it allows the host to have direct control over the transponder. It can also be used to handle new versions of transponders which are not yet fully supported.

Examples of Pass-through commands are:

Read block
Write block
Lock block

☞ **Note:**
Pass-through commands are not normally used to communicate with the reader. Only Single, Compound, and Ancillary service commands are explained in detail in this document.

This service code should only be used by expert users and transponder developers.

02 Single Service

On receipt of a Single service request the reader analyzes and verifies it, then formats and sends a single command to the transponder.

When the reader receives a response, it analyzes and verifies it, then formats and sends a unique response to the host.

Examples of Single service commands are:

Read block
Write block
Lock block

03 Compound Service

Compound messages instruct the reader to perform defined actions on transponders using the Transponder Protocol. These messages differ from the single service transponder messages in that they instruct the reader to perform complex operations involving multiple commands before returning a response to the host.

Compound messages are also used to emulate Transponder Protocol commands which are not implemented by a particular transponder version. An example is a Compound message which emulates Read multiblock using a sequence of Read block commands. In this case, a compound message saves the host from having to initiate the Read block message for each block required.

On receipt of a Compound service request, the reader analyzes and verifies it, then formats and sends one or more commands to the transponder.

When the reader receives one or more responses, it analyses and verifies them, then formats and sends a unique response to the host. This response may be contained in several frames, depending on the size of the response (see More bit).

The Compound service is used to improve system performance. For example, it is much faster to send a single Read three blocks of memory command rather than to send three separate Read block commands.

The Compound service is also used to request the reader to perform complex operations such as simultaneous identification (SID) of the transponders present in the reader antenna's field. It is an effective way to offload the host system from detailed and complex actions.

Examples of Compound service are:

Read block SID
Read multiblock
Write multiblock

☞ **Note:**
Single commands can be emulated by the reader similar to Compound commands if the reader knows that the target transponder cannot execute the requested command. For example: if the host requests a "Read multiblock" and the target transponder supports only "Read (single) block", the reader turns this into a sequence of several Read block commands. This capability is authorized by the Emulation flag in the request control flags field.

04 Ancillary Service

On receipt of an Ancillary service request, the reader analyzes and verifies it, then executes the request locally. No commands are sent to the transponder.

The Ancillary service request can be used to download a new version of the reader software, or to set operating and communications parameters.

Examples of Ancillary service are:

Reader reset
Read reader version
Reader diagnostic

4.2 Service Code Listing

Note:
The Pass-through (01) service code is not listed here since its format is defined by the Transponder Protocol and is referenced in that reference guide.

Command	Service Code			Command Code
	Single	Compound	Ancillary	
Tag Version	02_{16}	-	-	03_{16}
Read Block	02_{16}	-	-	01_{16}
Read Multiblock	02_{16}	03_{16}	-	02_{16}
Read Block SID	-	03_{16}	-	FE_{16}
Read Multiblock SID	-	03_{16}	-	FD_{16}
Write Block	02_{16}	-	-	05_{16}
Write and Lock Block	02_{16}	-	-	07_{16}
Write Multiblock	02_{16}	03_{16}	-	06_{16}
Lock Block	02_{16}	-	-	08_{16}
Lock Multiblock	02_{16}	03_{16}	-	09_{16}
Repeat Last Request	-	-	04_{16}	01_{16}
Send Last Response	-	-	04_{16}	02_{16}
Stop Continuous	-	-	04_{16}	04_{16}
Start Synch	-	-	04_{16}	05_{16}
Reader Reset	-	-	04_{16}	10_{16}
Reader Version	-	-	04_{16}	11_{16}
Reader Diagnostic	-	-	04_{16}	12_{16}
Read Reader Setup	-	-	04_{16}	13_{16}
Start Flash Loader	-	-	04_{16}	16_{16}
Factory Lock Block	02_{16}	-	-	$3D_{16}$
Write SID Code	02_{16}	-	-	$3E_{16}$
Factory Programming Off	02_{16}	-	-	$3F_{16}$

Note:
The typical numbering schemes used in the transponder start at 0, to maximize the usable range of numbers. For example the 8 blocks in the transponder are numbered 0 to 7, but are referred to here as block number 1 to 8

Note:
Write SID Code, *Factory Lock Block* and *Factory Programming Off* are Factory Commands, not available for customer use.

4.3 Error Code Listing

If bit 0 of the Format code status flag equals 1, an error has occurred. For Ancillary service messages, the msb of the command code is set to 1 to indicate an error. The error code is the first byte of the message data following any address.

General Error Codes		Specific Error Codes for Single, Compound and Ancillary messages	
Error Code	**Definition**	**Error Code**	**Definition**
00_{16}	Reserved	10_{16}	Block not available
01_{16}	Command not supported by transponder	11_{16}	At least one block not available
02_{16}	Command not supported by reader	12_{16}	Block already locked
03_{16}	data format error, request aborted	13_{16}	At least one block already locked
04_{16}	Addressing mode not available for this request	14_{16}	Block not read
05_{16}	Reserved	15_{16}	At least one block not read
06_{16}	No transponder found	16_{16}	Block not successfully written
07_{16}	Reserved	17_{16}	At least one block not successfully written
08_{16}	Reserved	18_{16}	Block not successfully locked
09_{16}	Reserved	19_{16}	At least one block not successfully locked
$0A_{16}$	Reserved	$1A_{16}$	Reserved
$0B_{16}$	Reserved	$1B_{16}$	Reserved
Reserved	Reserved	$1C_{16}$	Reserved
$0D_{16}$	Reserved	$1D_{16}$	Reserved
$0E_{16}$	Reserved	$1E_{16}$	No previous request (reader reset)
$0F_{16}$	Undefined system error, request aborted	$1F_{16}$	function not allowed (disabled)
		FA_{16} *	format/range failure in uplink data
		FB_{16} *	uplink CRC error
		FC_{16} *	uplink Manchester decoding error
		FD_{16} *	uplink general timeout; no complete response found
		FE_{16} *	uplink EOF (end-of-frame) not found
		FF_{16} *	uplink SOF (start-of-frame) not found

Error codes 00_{16} - $0F_{16}$ are general error codes. Error codes 10_{16} - FF_{16} are specific error codes. The detailed message references describe the nature of the specific errors. These error messages are only available if *Detailed error messages* is turned on via the Ancillary message *Reader diagnostics.* Otherwise error message 06_{16}, *No transponder found* is returned in all cases.

Chapter 5

List of Commands

Topic **Page**

5.1 Tag Version 41
5.2 Read Block 43
5.3 Write Block 45
5.4 Write and Lock Block 47
5.5 Lock Block 49
5.6 Read Multiblock 51
5.7 Write Multiblock 54
5.8 Lock Multiblock 56
5.9 Read Block SID 58
5.10 Read Multiblock SID 62
5.11 Repeat Last Request 67
5.12 Send Last Response 69
5.13 Stop Continuous 71
5.14 Start Synch 73
5.15 Reset Reader 74
5.16 Reader Version 76
5.17 Reader Diagnostic 78
5.18 Read Reader Setup 80
5.19 Start Flash Loader 82
5.20 Write SID Code 83
5.21 Factory Lock Block 85
5.22 Factory Programming Off 87

5.1 Tag Version

Description:	Reads version information from a transponder. This information is permanently factory-programmed into the transponder
Command code:	03_{16}
Request:	Continuous Mode possible Addressed mode permitted Non-addressed mode permitted

Request Data Format

Name	bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
Address	4	Transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
Manufacturer code	1	Identifies tag manufacturer	Only least-significant 7 bits valid 0000001_2 = Texas Instruments 0000010_2 – 1111111_2 unassigned
Chip version	2	Chip version used in this transponder. Refer to Texas Instruments web site for current chip version number.	Only least-significant 9 bits valid Range 000000000_2 – 111111111_2
Block size	1	Size of each block for this transponder, coded in number of bytes, counting from value 0. For example, a value of 3 indicates 4 bytes, or 32 bits.	Only least-significant 5 bits valid Range 00000_2 – 11111_2
Number of blocks	1	Number of blocks in this transponder, counting from value 0. For example, a value of 7 indicates 8 blocks.	00_{16} – FF_{16}

Request Data

Example: *D5 00 07 02 00 03 00 00 DB A3$_{16}$*
Tag version (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Tag version	03$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
BCC	2	CRC—calculated on previous 7 bytes	DB A3$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 10 02 00 03 00 00 00 12 AB 81 01 00 05 03 07 E1 85$_{16}$*
Tag version data returned ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	16 bytes follow this block.	00 10$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Tag version	03$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Address	4		00 12 AB 81$_{16}$
Manufacturer code	1	Texas Instruments	01$_{16}$
Chip version	2	Version 5	00 05$_{16}$
Block size	1	3+1 bytes x 8 = 32 bits	03$_{16}$
Number of blocks	1	7+1 = 8 blocks	07$_{16}$
BCC	2	CRC –calculated on previous 16 bytes	E1 85$_{16}$

Note:
The address is always returned in response to an addressed or non-addressed request since it constitutes part of the transponder information.

5.2 Read Block

Description:	Reads the contents of a single block of data from a transponder.
Command code	01_{16}
Request:	Continuous mode possible Addressed mode permitted Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
Block number	1	Section of transponder data	00_{16} - 07_{16}

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
Block number	1	Section of transponder data	00_{16} - 07_{16}
Lock status	1	Irreversible locking of block	00_{16} = unlocked 01_{16} = user-locked 02_{16} = factory-locked (secured)
Block data	4	Block data	00 00 00 00_{16} – FF FF FF FF_{16}

Specific error codes: Block not available = 10_{16}

Request Data

Example: *D5 00 08 02 00 01 00 00 01 B0 B9$_{16}$*
Read block 2 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	8 bytes follow this block.	00 08$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Read block	01$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Block number	1	Block 2	01$_{16}$
BCC	2	CRC—calculated on previous 8 bytes	B0 B9$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 0D 02 00 01 00 00 01 00 12 34 56 78 78 E6$_{16}$*
Block 2 read ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	13 bytes follow this block.	00 0D$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Read block	01$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		12 34 56 78$_{16}$
BCC	2	CRC -calculated on previous 13 bytes	78 E6$_{16}$

5.3 Write Block

CAUTION:

Using this command in non-addressed mode may write the block to any transponder within the field.

Description: Writes a single block of data to a transponder

Command code: 05_{16}

Request: Continuous mode possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
Block number	1	Section of transponder data	00_{16} - 07_{16}
Block data	4	Block data	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Specific error codes : Block not available = 10_{16}
Block already locked = 12_{16}
Block not successfully put = 16_{16}

Request Data

Example: *D5 00 0C 02 00 05 00 00 02 12 AB CD 90 02 6B$_{16}$*
Write block 3 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	12 bytes follow this block.	00 0C$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Write block	05$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Block number	1	Block 3	02$_{16}$
Block data	4		12 AB CD 90$_{16}$
BCC	2	CRC—calculated on previous 12 bytes	02 6B$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 02 00 05 00 00 69 03$_{16}$*
Block 3 written ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Write block	05$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	69 03$_{16}$

5.4 Write and Lock Block

CAUTION:

Using this command in non-addressed mode may write the block to any transponder within the field and irreversibly lock it.

Description: Writes a single block of data to a transponder and locks it

Command code: 07_{16}

Request: Continuous mode not possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
Block number	1	Section of transponder data	00_{16} - 07_{16}
Block data	4	Block data	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Specific Error Codes**:** Block not available = 10_{16}
Block already locked = 12_{16}
Block not successfully put = 16_{16}

Request Data

Example: *D5 00 0C 02 00 07 00 00 04 12 34 56 78 AC 77$_{16}$*
Write block 3 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	12 bytes follow this block.	00 0C$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Write and lock block	07$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Block number	1	Block 3	02$_{16}$
Block data	4		12 34 56 78$_{16}$
BCC	2	CRC—calculated on previous 12 bytes	AC 77$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 02 00 07 00 00 07 63$_{16}$*
Block 3 written and locked ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Write and lock block	07$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	07 63$_{16}$

5.5 Lock Block

CAUTION:

Using this command in non-addressed mode may irreversibly lock this block for any transponder within the field.

Description: Locks a single block of data on a transponder.

Command code: 08_{16}

Request: Continuous mode not possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
Block number	1	Section of transponder data	00_{16} - 07_{16}

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Specific error codes: Block not available = 10_{16}
Block already locked = 12_{16}
Block not successfully put = 16_{16}

Request Data

Example: *D5 00 08 02 00 08 00 00 03 63 8C$_{16}$*
Lock block 4 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	8 bytes follow this block.	00 08$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Lock block	08$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Block number	1	Block 4	03$_{16}$
BCC	2	CRC—calculated on previous 8 bytes	63 8C$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 02 00 08 00 00 2B 52$_{16}$*
Block 4 locked ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Lock block	08$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	2B 52$_{16}$

5.6 Read Multiblock

Description: Reads the contents of more than one block of data from a transponder.

Command code: 02_{16}

Request: Continuous mode possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
First block number	1	Section of transponder data	00_{16} - 07_{16}
Number of blocks	1	Total number of blocks starting at first block above	01_{16} - 08_{16}

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
Number of blocks read X	1	This determines how many times the following 3 items are repeated	00_{16} - 07_{16}
Block number	1	Section of transponder data	00_{16} - 07_{16}
Lock status	1	Irreversible locking of block	00_{16} = unlocked 01_{16} = user-locked 02_{16} = factory-locked (secured)
Block data	4	Block data	00 00 00 00_{16} – FF FF FF FF_{16}
< error block number>	1	Present if not all of requested block could be read. Indicates first block at which an error occurred.	00_{16} - 07_{16}

Specific error codes: At least one block not available = 11_{16}
At least one block not read = 15_{16}

Request Data

Example: *D5 00 09 03 00 02 00 00 00 03 12 E6$_{16}$*
Read multiblock from 1 to 3 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	9 bytes follow this block.	00 09$_{16}$
Service code	1	Compound	03$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Read multiblock	02$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
First block number	1	Block 1	00$_{16}$
Number of blocks	1	3 blocks	03$_{16}$
BCC	2	CRC—calculated on previous 9 bytes	12 E6$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example *D5 00 1A 03 00 02 00 00 03 00 00 12 34 56 78 01 00 AA BB CC DD 02 01 55 55 55 55 17 33$_{16}$*
3 Blocks read ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	26 bytes follow this block.	00 1A$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Read multiblock	02$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Number of blocks read	1	3 blocks (following data)	03$_{16}$
Block number	1	Block 1	00$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		12 34 56 78$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		AA BB CC DD$_{16}$
Block number	1	Block 3	02$_{16}$
Lock status	1	User-locked	01$_{16}$
Block data	4		55 55 55 55$_{16}$
BCC	2	CRC –calculated on previous 26 bytes	17 33$_{16}$

5.7 Write Multiblock

CAUTION:

Using this command in non-addressed mode may write the blocks to any transponder within the field.

Description:	Writes more than one block of data to a transponder
Command code:	06_{16}
Request:	Continuous mode possible Addressed mode permitted Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
First block number	1	Section of transponder data	00_{16} - 07_{16}
Number of blocks	1	Total number of blocks starting at first block above	01_{16} - 08_{16}
Block data	4	Repeated 'Number of blocks' times	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
< error block number >	1	Present if not all of requested block could be written. Indicates first block at which an error occurred.	00_{16} - 07_{16}

Specific error codes**:**	At least one block not available 11_{16} At least one block already locked = 13_{16} At least one block not successfully written = 17_{16}

Request Data

Example: *D5 00 15 03 00 06 00 00 00 03 12 AB CD 90 33 44 55 66 12 34 56 78 D9 6D$_{16}$*
Write multiblock 1 to 3 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	21 bytes follow this block.	00 15$_{16}$
Service code	1	Compound	03$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Write multiblock	06$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
First block number	1	Block 1	00$_{16}$
Number of blocks	1	3 blocks	03$_{16}$
Block data	4	Block 1 data	12 AB CD 90$_{16}$
Block data	4	Block 2 data	33 44 55 66$_{16}$
Block data	4	Block 3 data	12 34 56 78$_{16}$
BCC	2	CRC—calculated on previous 21 bytes	D9 6D$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 03 00 06 00 00 9A 02$_{16}$*
Blocks 1 to 3 written ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Write multiblock	06$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	9A 02$_{16}$

5.8 Lock Multiblock

CAUTION:

Using this command in non-addressed mode may irreversibly lock these blocks for any transponder within the field.

Description: Locks more than one block of data on a transponder

Command code: 09_{16}

Request: Continuous mode not possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
First block number	1	Section of transponder data	00_{16} - 07_{16}
Number of blocks	1	Total number of blocks starting at first block above	01_{16} - 08_{16}

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	00 00 00 00_{16} – FF FF FF FF_{16}
< error block number >	1	Present if not all of requested block could be locked. Indicates first block at which an error occurred.	00_{16} - 07_{16}

Specific error codes: At least one block not available = 11_{16}
At least one block already locked = 13_{16}
At least one block not successfully locked = 19_{16}

Request Data

Example: *D5 00 09 03 00 09 00 00 00 03 FE 19$_{16}$*
Lock multiblock 1 to 3 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	9 bytes follow this block.	00 09$_{16}$
Service code	1	Compound	03$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Lock multiblock	08$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
First block number	1	Block 1	00$_{16}$
Number of blocks	1	3 blocks	03$_{16}$
BCC	2	CRC—calculated on previous 9 bytes	FE 19$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 03 00 09 00 00 B6 33$_{16}$*
Blocks 1 to 3 locked ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Lock multiblock	09$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	B6 33$_{16}$

5.9 Read Block SID

Description: Reads the contents of a single block of data from all transponders within reading range of the reader, using simultaneous identification.

A maximum of 100 transponders may be read on a single pass, whilst each response may contain data from up to 50 transponders. This means, if over 50 transponders are found, 2 responses will be returned, as indicated by the More flag, contained in the status flag.

Command code: FE_{16}

Request: Continuous mode possible
Addressed mode not permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
Block number	1	Section of transponder data	00_{16} - 07_{16}
Response options	1	Determines which data is returned in any successful response	Bit 0: 0 = no addresses 1 = addresses Bit 1: 0 = no version data 1 = version data Bit 2: 0 = respond with block data plus data selected by bits 0 and 1 1 = respond with addresses only Bits 3 – 7 reserved

Response Data Format

Name	# of bytes	Description	Permitted values/range
Response options	1	Determines which data is returned in any successful response	Bit 0: 0 = no addresses 1 = addresses Bit 1: 0 = no version data 1 = version data Bit 2: 0 = response with block data plus data selected by bits 0 and 1 1 = response with addresses only Bits 3 – 7 reserved
Number of transponders found X	1	This determines how many times the following 3 items are repeated	00_{16} - 32_{16}
Address	4	Transponder address Present if bit 0 or bit 2 in the response options is set to 1	00 00 00 00_{16} – FF FF FF FF_{16}
Manufacturer code	1	Identifies tag manufacturer Present if bit 1 in the response options is set to 1, and bit 2 is set to 0.	Only least-significant 7 bits valid 0000001_2 = Texas Instruments 0000010_2 – 1111111_2 unassigned
Chip version	2	Chip version used in this transponder. Refer to Texas Instruments web site for current chip version number. Present if bit 1 in the response options is set to 1, and bit 2 is set to 0.	Only least-significant 9 bits valid Range 000000000_2 – 111111111_2
Block size	1	Size of each block for this transponder, coded in number of bytes, counting from value 0. For example, a value of 3 indicates 4 bytes, or 32 bits. Present if bit 1 in the response options is set to 1, and bit 2 is set to 0.	Only least-significant 5 bits valid Range 00000_2 – 11111_2
Number of blocks	1	Number of blocks in this transponder, counting from value 0. For example, a value of 7 indicates 8 blocks. Present if bit 1 in the response options is set to 1, and bit 2 is set to 0.	00_{16} – FF_{16}
Block number	1	Section of transponder data Present if bit 2 in the response options is set to 0	00_{16} - 07_{16}

Lock status	1	Irreversible locking of block Present if bit 2 in the response options is set to 0	00_{16} = unlocked 01_{16} = user-locked 02_{16} = factory-locked (secured)
Block data	4	Block data Present if bit 2 in the response options is set to 0	00 00 00 00_{16} – FF FF FF FF_{16}

Specific error codes: Block not available = 10_{16}

Request Data

Example: *D5 00 09 03 00 FE 00 00 01 01 B5 E8*$_{16}$
Read Block 2 with SID

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	9 bytes follow this block.	00 09_{16}
Service code	1	Compound	03_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Read block SID	FE_{16}
Format code*	1	Non-addressed	00_{16}
Synch code*	1	No synchronization	00_{16}
Block number	1	Block 2	01_{16}
Response options	1	001_2 = respond with addresses, block data, but no version data.	01_{16}
BCC	2	CRC—calculated on previous 9 bytes	B5 $E8_{16}$?
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 27 03 00 FE 00 00 01 03 00 11 12 13 01 00 12 34 56 78 00 02 33 33 01 01 55 55 55 55 00 00 1A 12 01 02 AA BB AA BB D8 B9$_{16}$*
Block 2 read from 3 transponders

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	39 bytes follow this block.	00 27$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Read block SID	FE$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Response options	1	001$_2$ = respond with addresses, block data, but no version data.	01$_{16}$
Number of transponders read	1	3 transponders (following data)	03$_{16}$
Address	4	Transponder 1	00 11 12 13$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		12 34 56 78$_{16}$
Address	4	Transponder 2	00 02 33 33$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	User-locked	01$_{16}$
Block data	4		55 55 55 55$_{16}$
Address	4	Transponder 3	00 00 1A 12$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Factory-locked	02$_{16}$
Block data	4		AA BB AA BB$_{16}$
BCC	2	CRC –calculated on previous 39 bytes	D8 B9$_{16}$

5.10 Read Multiblock SID

Description: Reads the contents of more than one block of data from all transponders within reading range of the reader, using simultaneous identification.

A maximum of 100 transponders may be read on a single pass, whilst each response contains data from a single transponder. For example, if 20 transponders found and 8 blocks are read from each, then 20 responses will be returned, as indicated by the Data extension bit in the status flag.

Command code: FD_{16}

Request: Continuous mode possible
Addressed mode not permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
First block number	1	Section of transponder data	00_{16} - 07_{16}
Number of blocks	1	Total number of blocks starting at first block above	01_{16} - 08_{16}
Response options	1	Determines which data is returned in any successful response	Bit 0: 0 = no addresses 1 = addresses Bit 1: 0 = no version data 1 = version data Bits 2 – 7 reserved

Response Data Format

Name	# of bytes	Description	Permitted values/range
Response options	1	Determines which data is returned in any successful response	Bit 0: 0 = no addresses 1 = addresses Bit 1: 0 = no version data 1 = version data Bits 2 – 7 reserved
Address	4	Transponder address Present if bit 0 in the response options is set to 1	00 00 00 00_{16} – FF FF FF FF_{16}
Manufacturer code	1	Identifies tag manufacturer Present if bit 1 in the response options is set to 1.	Only least-significant 7 bits valid 0000001_2 = Texas Instruments 0000010_2 – 1111111_2 unassigned
Chip version	2	Chip version used in this transponder. Refer to Texas Instruments web site for current chip version number. Present if bit 1 in the response options is set to 1.	Only least-significant 9 bits valid Range 000000000_2 – 111111111_2
Block size	1	Size of each block for this transponder, coded in number of bytes, counting from value 0. For example, a value of 3 indicates 4 bytes, or 32 bits. Present if bit 1 in the response options is set to 1.	Only least-significant 5 bits valid Range 00000_2 – 11111_2
Number of blocks	1	Number of blocks in this transponder, counting from value 0. For example, a value of 7 indicates 8 blocks. Present if bit 1 in the response options is set to 1.	Only least-significant 5 bits valid 00_{16} – FF_{16}
Number of blocks read X	1	This determines how many times the following 3 items are repeated	01_{16} - 08_{16}
Block number	1	Section of transponder data	00_{16} - 07_{16}
Lock status	1	Irreversible locking of block	00_{16} = unlocked 01_{16} = user-locked 02_{16} = factory-locked (secured)
Block data	4	Block data	00 00 00 00_{16} – FF FF FF FF_{16}

Specific error codes: At least one block not available = 11_{16}

Request Data

Example: *D5 00 0A 03 00 FD 00 00 00 03 01 A3 A6$_{16}$*
Read blocks 1 to 3 with SID

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	10 bytes follow this block.	00 0A$_{16}$
Service code	1	Compound	03$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Read block SID	FE$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
First block number	1	Block 1	00$_{16}$
Number of blocks	1	3 blocks	03$_{16}$
Response options	1	001$_2$ = Respond with addresses, block data, but no version data.	01$_{16}$
BCC	2	CRC—calculated on previous 10 bytes	A3 A6$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data (First Response)

Example: *D5 00 1F 03 02 FD 00 00 01 00 11 12 13 03 00 01 12 34 56 78 01 02 9A BC DE F0 02 00 22 44 66 88 96 62$_{16}$*
Blocks 1 to 3 read from 2 transponders. First response.

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	31 bytes follow this block.	00 1F$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception, more responses follow for this request	02$_{16}$
Command code	1	Read block SID	FD$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Response options	1	001$_2$ = respond with addresses, block data, but no version data.	01$_{16}$
Address	4	Transponder 1	00 11 12 13$_{16}$
Number of blocks read	1	3 Blocks	03$_{16}$
Block number	1	Block 1	00$_{16}$
Lock status	1	User-locked	01$_{16}$
Block data	4		12 34 56 78$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Factory-locked	02$_{16}$
Block data	4		9A BC DE F0$_{16}$
Block number	1	Block 3	02$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		22 44 66 88$_{16}$
BCC	2	CRC –calculated on previous 31 bytes	96 62$_{16}$

Response Data (Second Response)

Example: *D5 00 1F 03 00 FD 00 00 01 00 00 22 33 03 00 00 55 55 55 55 01 00 44 44 44 44 02 00 CC CC CC CC D6 C3$_{16}$*
Blocks 1 to 3 read from 2 transponders. Second response.

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	31 bytes follow this block.	00 1F$_{16}$
Service code	1	Compound	03$_{16}$
Status flag	1	No exception, last response for this request	00$_{16}$
Command code	1	Read block SID	FD$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Response options	1	001$_2$ = Respond with addresses, block data, but no version data.	01$_{16}$
Address	4	Transponder 2	00 00 22 33$_{16}$
Number of blocks read	1	3 Blocks	03$_{16}$
Block number	1	Block 1	00$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		55 55 55 55$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		44 44 44 44$_{16}$
Block number	1	Block 3	02$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		CC CC CC CC$_{16}$
BCC	2	CRC –calculated on previous 31 bytes	D6 C3$_{16}$

5.11 Repeat Last Request

Description: The reader repeats the last request that it performed.

The last request received by the reader is executed again, using the same parameters. If this involves read and write operations with transponders, the complete request is executed again, and therefore the response is exactly as if the last request had been sent.

No data is associated with this message.

Command code: 01_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Specific error codes**:** No previous request (reader reset) = $1E_{16}$

Request Data

Example: *D5 00 05 04 00 01 9E 57_{16}*
Repeat last request (following Read block 2)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Repeat last request	01_{16}
BCC	2	CRC—calculated on previous 5 bytes	9E 57_{16}
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: D5 00 0D 02 00 01 00 00 01 00 12 34 56 78 78 E6$_{16}$
Last request repeated - Block 2 read ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	13 bytes follow this block.	00 0D$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Read block	01$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		12 34 56 78$_{16}$
BCC	2	CRC -calculated on previous 13 bytes	78 E6$_{16}$

5.12 Send Last Response

Description: The last response sent by the reader is sent again, without executing the request again

No data is associated with this message.

Command code: 02_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data

Example: *D5 00 05 04 00 02 AE 34_{16}*
Send last response (following Read block 2)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Repeat last request	01_{16}
BCC	2	CRC—calculated on previous 5 bytes	AE 34_{16}
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 0D 02 00 01 00 00 01 00 12 34 56 78 78 E6$_{16}$*
Last request repeated - Block 2 read ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	13 bytes follow this block.	00 0D$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Read block	01$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
Block number	1	Block 2	01$_{16}$
Lock status	1	Unlocked	00$_{16}$
Block data	4		12 34 56 78$_{16}$
BCC	2	CRC -calculated on previous 13 bytes	78 E6$_{16}$

5.13 Stop Continuous

Description: This message causes the reader to stop running in continuous mode, as determined by the format code of the last request. The reader completes any actions that have already been started, including responding with data to the host system. Finally the reader sends the Stop continuous response to the host system.

The same action can be performed implicitly by sending any other request, in which case the current request running continuously will be halted, and the new request executed. The state of the continuous mode will then depend on the flag settings of the new request.

The message has no effect if the reader is not running in continuous mode.

No data is associated with this message.

Command code: 04_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data

Example: *D5 00 05 04 00 04 CE F2$_{16}$*
Stop continuous

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	5 bytes follow this block.	00 05$_{16}$
Service code	1	Ancillary	04$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Stop continuous	04$_{16}$
BCC	2	CRC—calculated on previous 5 bytes	CE F2$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 05 04 00 04 CE F2$_{16}$*
Continuous mode stopped

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	5 bytes follow this block.	00 05$_{16}$
Service code	1	Ancillary	04$_{16}$
Status flag*	1	No exception	00$_{16}$
Command code	1	Stop continuous	04$_{16}$
BCC	2	CRC—calculated on previous 5 bytes	CE F2$_{16}$

5.14 Start Synch

Description: This message is used to trigger a sequence of synchronized operations between a group of coupled readers, by starting the reader designated as the master by previously issued requests where the Synch flag contained a setting other than 'No synchronization'

There is no data associated with this request. There is no associated response, since issuing this request causes the stored request with synchronization to be executed.

Command code: 05_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data

Example: *D5 00 05 04 00 05 DE D3*$_{16}$
Start synch

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Start Synch	05_{16}
BCC	2	CRC—calculated on previous 5 bytes	DE $D3_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

5.15 Reset Reader

Description:

This message causes the reader to perform a software reset. Any currently executing command will be completed, including any response to the host. Thereafter, the reader sends the Reader reset response to the host, and immediately afterwards, performs a software reset.

Any temporary settings or data stored in volatile memory will be lost after reset.

There is no data associated with this request.

Command code 10_{16}

Request:

Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data

Example: *D5 00 05 04 00 10 9C 47$_{16}$*
Reset reader

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Reset reader	10_{16}
BCC	2	CRC—calculated on previous 5 bytes	9C 47_{16}
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 05 04 00 10 9C 47$_{16}$*
Reader reset (immediately after response)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	5 bytes follow this block.	00 05$_{16}$
Service code	1	Ancillary	04$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Reset reader	10$_{16}$
BCC	2	CRC—calculated on previous 5 bytes	9C 47$_{16}$

5.16 Reader Version

Description: The reader responds by sending the firmware version running on this reader.

There is no data associated with this request.

Command code: 11_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Response Data Format

Name	# of bytes	Description	Permitted values/range
Firmware version	3		00.00.00 – 99.99.99
Firmware type	1	Denotes special versions	Bit 0 = 0: standard version Bit 0 = 1: engineering version

Request Data

Example: *D5 00 05 04 00 11 8C 66_{16}*
Reader version

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Reader version	11_{16}
BCC	2	CRC—calculated on previous 5 bytes	8C 66_{16}
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 09 04 00 11 02 00 00 00 3C 21$_{16}$*
Reader version 2.0.0

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	9 bytes follow this block.	00 09$_{16}$
Service code	1	Ancillary	04$_{16}$
Status flag*	1	No exception	00$_{16}$
Command code	1	Reader version	11$_{16}$
Firmware version	3	Version 2.00.00	02 00 00$_{16}$
Firmware type	1	Standard	00$_{16}$
BCC	2	CRC -calculated on previous 9 bytes	3C 21$_{16}$

5.17 Reader Diagnostic

Description: The reader performs diagnostic operations and settings associated with the reader.

Command code: 12_{16}

Request: Continuous mode possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data Format

Name	# of bytes	Description	Permitted values/range
Diagnostic operation	1	Operation to be performed	01_{16} = Selftest 02_{16} = Reserved 03_{16} = Transmitter on 04_{16} = Transmitter off 05_{16} = Detailed error messages 06_{16} = Normal error messages (default setting) 07_{16} - FF_{16} = Reserved

Response Data Format

Name	# of bytes	Description	Permitted values/range
Diagnostic status	1	Status after executing operation	01_{16} = Selftest failed FF_{16} = Selftest passed 02_{16} = Reserved 03_{16} = Transmitter turned on 04_{16} = Transmitter turned off 05_{16} = Detailed error messages on 06_{16} = Normal error messages on (default setting) 07_{16} – FE_{16} = Reserved

Request Data

Example: *D5 00 06 04 00 12 01 83 AB$_{16}$*
Reader diagnostic – selftest

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	6 bytes follow this block.	00 06$_{16}$
Service code	1	Ancillary	04$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Reader diagnostic	12$_{16}$
Diagnostic operation	1	Selftest	01$_{16}$
BCC	2	CRC—calculated on previous 6 bytes	83 AB$_{16}$

Response Data

Example: *D5 00 06 04 00 12 FF 8D 7A$_{16}$*
Reader selftest OK

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	6 bytes follow this block.	00 06$_{16}$
Service code	1	Ancillary	04$_{16}$
Status flag*	1	No exception	00$_{16}$
Command code	1	Reader diagnostic	12$_{16}$
Diagnostic status	3	Selftest OK	FF$_{16}$
BCC	2	CRC -calculated on previous 6 bytes	8D 7A$_{16}$

5.18 Read Reader Setup

Description: The reader sends the current setup and configuration data from the on-board flash memory.

There is no data associated with this request.

Command code: 13_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Response Data Format

Name	# of bytes	Description	Permitted values/range
Reader setup data	12	The reader setup data size can be dependent on the firmware version.	See table below for version 2.00.xx setup data.

Request Data

Example: *D5 00 05 04 00 13 AC* $\mathit{24}_{16}$
Read reader setup

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	5 bytes follow this block.	00 05_{16}
Service code	1	Ancillary	04_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Read reader setup	13_{16}
BCC	2	CRC—calculated on previous 5 bytes	AC 24_{16}
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 0D 04 00 13 01 04 01 01 00 00 0A 00 6B 28$_{16}$*
Reader setup data

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	13 bytes follow this block.	00 0D$_{16}$
Service code	1	Ancillary	04$_{16}$
Status flag*	1	No exception	00$_{16}$
Command code	1	Read reader setup	13$_{16}$
Setup data	1	Setup descriptor	01$_{16}$
Setup data	1	Baudrate	04$_{16}$
Setup data	1	Read retries	01$_{16}$
Setup data	1	Write retries	01$_{16}$
Setup data	1	Retry control	00$_{16}$
Setup data	2	Continuous delay	00 0A$_{16}$
Setup data	1	Compound error	00$_{16}$
BCC	2	CRC -calculated on previous 13 bytes	6B 28$_{16}$

5.19 Start Flash Loader

Description:

To perform flash re-programming operations, for firmware upgrade or setup data alteration, an on-board flash loader program must be started.

Once the flash loader is started, the reader unit no longer communicates via the Host Protocol, but uses and abbreviated protocol governed by the size of the loader. The flash loader runs until a hard or soft reset is performed on the reader. For this reason there is no response to this request.

For performing flash programming operations, it is recommended to use the Tag-it Navigator program, which is able to perform setup data alteration and firmware re-programming.

Command code: 16_{16}

Request: Continuous mode not possible
Addressed mode not applicable
Non-addressed mode not applicable

Request Data

Example: *D5 00 05 04 00 16 FC 81*$_{16}$
Start flash loader

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	5 bytes follow this block.	00 05$_{16}$
Service code	1	Ancillary	04$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Start flash loader	16$_{16}$
BCC	2	CRC—calculated on previous 5 bytes	FC 81$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

5.20 Write SID Code

Note:
This is a factory command and is only supported by TI internal Engineering versions of the reader software.

Description: Puts the SID code into the transponder, which is from then on used as the address. After locking (see Factory Lock Block), the SID code cannot be changed.

Command code 3E$_{16}$

Request: Continuous mode not possible
Addressed mode not permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
Address	4	Transponder address	00 00 00 00$_{16}$ – FF FF FF FF$_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
Address	4	Transponder address	00 00 00 00$_{16}$ – FF FF FF FF$_{16}$

Specific Error Codes**:** Block already locked = 12$_{16}$
Block not successfully put = 16$_{16}$
Function not allowed (disabled) = 1F$_{16}$

For all error codes the actual SID address is returned as data.

Request Data

Example: *D5 00 0B 02 00 3E 00 00 12 34 56 78 C9 3A*
Write SID Code

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	11 bytes follow this block.	$00\ 0B_{16}$
Service code	1	Single	02_{16}
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00_{16}
Command code	1	Write SID code	$3E_{16}$
Format code*	1	Non-addressed	00_{16}
Synch code*	1	No synchronization	00_{16}
Address	4	SID Address (Data)	$12\ 34\ 56\ 78_{16}$
BCC	2	CRC—calculated on previous 11 bytes	$C9\ 3A_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 0B 02 00 3E 00 00 12 34 56 78 C9 3A*$_{16}$
SID Code written ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	$D5_{16}$
Data length	2	11 bytes follow this block.	$00\ 0B_{16}$
Service code	1	Single	02_{16}
Status flag	1	No exception	00_{16}
Command code	1	Write SID Code	$3E_{16}$
Format code	1	Non-addressed, no error	00_{16}
Synch code	1	No synchronization	00_{16}
Address	4	SID Address (Data)	$12\ 34\ 56\ 78_{16}$
BCC	2	CRC –calculated on previous 11 bytes	$C9\ 3A_{16}$

5.21 Factory Lock Block

Note:
This is a factory command and is only supported by TI internal Engineering versions of the reader software.

CAUTION:

Using this command in non-addressed mode may irreversibly lock this block for any transponder within the field.

Description: Locks a single block of data on a transponder with the status “Factory Programmed”.

Command code: $3D_{16}$

Request: Continuous mode not possible
Addressed mode permitted
Non-addressed mode permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$
Block number	1	Section of transponder data	00_{16} - 07_{16}

Response Data Format

Name	# of bytes	Description	Permitted values/range
< Address >	4	Optional transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Specific error codes: Block not available = 10_{16}
Block already locked = 12_{16}
Block not successfully locked = 18_{16}
Function not allowed (disabled) = $1F_{16}$

Request Data

Example: *D5 00 08 02 00 3D 00 00 07 B3 A4$_{16}$*
Factory Lock Block 8 (non-addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	8 bytes follow this block.	00 08$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	Factory Lock Block	3D$_{16}$
Format code*	1	Non-addressed	00$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Block number	1	Block 8	07$_{16}$
BCC	2	CRC—calculated on previous 8 bytes	B3 A4$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 07 02 00 3D 00 00 05 07$_{16}$*
Block 8 factory locked ok

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	7 bytes follow this block.	00 07$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	Factory Lock Block	3D$_{16}$
Format code	1	Non-addressed, no error	00$_{16}$
Synch code	1	No synchronization	00$_{16}$
BCC	2	CRC –calculated on previous 7 bytes	05 07$_{16}$

5.22 Factory Programming Off

Note:
This is a factory command and is only supported by TI internal Engineering versions of the reader software.

Description: Clears the Factory Programming bit irreversible, disabling all factory programming functions.

Command code: $3F_{16}$

Request: Continuous mode not possible
Addressed mode not permitted
Non-addressed mode not permitted

Request Data Format

Name	# of bytes	Description	Permitted values/range
Address	4	Transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Response Data Format

Name	# of bytes	Description	Permitted values/range
Address	4	Transponder address	$00\ 00\ 00\ 00_{16}$ – $FF\ FF\ FF\ FF_{16}$

Specific Error Codes**:** Block not successfully locked = 18_{16}
Function not allowed (disabled) = $1F_{16}$

Request Data

Example: *D5 00 0B 02 00 3F 04 00 12 34 56 78 77 FA$_{16}$*
Factory Programming Off (Addressed)

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	11 bytes follow this block.	00 0B$_{16}$
Service code	1	Single	02$_{16}$
Control flag*	1	More bit = no more data associated with this request Emulation bit = off Auto-repeat bit = off BCC bit = CRC used.	00$_{16}$
Command code	1	FP Off	3F$_{16}$
Format code*	1	Addressed	04$_{16}$
Synch code*	1	No synchronization	00$_{16}$
Address	4	SID Address	12 34 56 78$_{16}$
BCC	2	CRC—calculated on previous 11 bytes	77 FA$_{16}$
* All reserved bits must be set to zero. Set remaining bits as the situation requires.			

Response Data

Example: *D5 00 0B 02 00 3F 04 00 12 34 56 78 77 FA$_{16}$*
Factory programming turned off

Name	# of bytes	Description	Value in example
Start	1	Start frame delimiter	D5$_{16}$
Data length	2	11 bytes follow this block.	00 0B$_{16}$
Service code	1	Single	02$_{16}$
Status flag	1	No exception	00$_{16}$
Command code	1	FP Off	3F$_{16}$
Format code	1	Addressed, no error	04$_{16}$
Synch code	1	No synchronization	00$_{16}$
Address	4	SID Address	12 34 56 78$_{16}$
BCC	2	CRC -calculated on previous 11 bytes	77 FA$_{16}$

Appendix A

BCC Error Detection Methods

The block check character (BCC) in the protocol frame is calculated on the data forming a message, and sent with the message for error detection purposes. The BCC may be a cyclic redundancy check (CRC) or a longitudinal redundancy Check (LRC).

It is recommended that the default method of error detection, cyclic redundancy check (CRC), be used as is defined as follows:

CRC Type	Length	Polynomial	Direction	Preset	Residue
CRC-CCITT	16 bits	$x^{16} + x^{12} + x^{5} + 1$	Forward	$FFFF_{16}$	$1D0F_{16}$

The CRC is calculated and the one's complement of this value is appended to the frame with the most significant bit (msb) first.

The CRC offers much better protection against data errors, but is more complex to implement. The LRC is simpler to implement and requires less processing time at the host, but offers a lower level of protection.

Wherever possible, use the CRC. However, some hosts (for example, PLCs) may not have the capability to support it, in which case the LRC is a useful and simple alternative.

An alternative error detection method, longitudinal redundancy check (LRC), may be used if the host controller is unable to support CRC calculation. However, the LRC provides a reduced level of error detection in comparison to the CRC.

The LRC is calculated by performing a cumulative Exclusive-OR operation on all the bytes in the message that generates an 8-bit LRC. This value is placed in the lowest significant byte (LS Byte) of the BCC, while the one's-complement of the same value is placed in the most significant byte (MS Byte).

To illustrate the possible BCC values, an example using the Reset Reader request is shown below:

SFD	BCC'd data	Transmitted BCC		Calculated BCC
D5	00 05 04 00 10	$9C\ 47_{16}$	CCITT-CRC	$63\ B8_{16}$
D5	00 05 04 10 10	$FE\ 01_{16}$	LRC	01_{16}

Appendix B

Reader Synchronization

This option will be implemented in future versions of the Tag-it reader. To enable more than one reader to be used in close proximity, the RF operations executed by the readers must be synchronized to prevent unwanted interference.

This may be achieved in three ways:

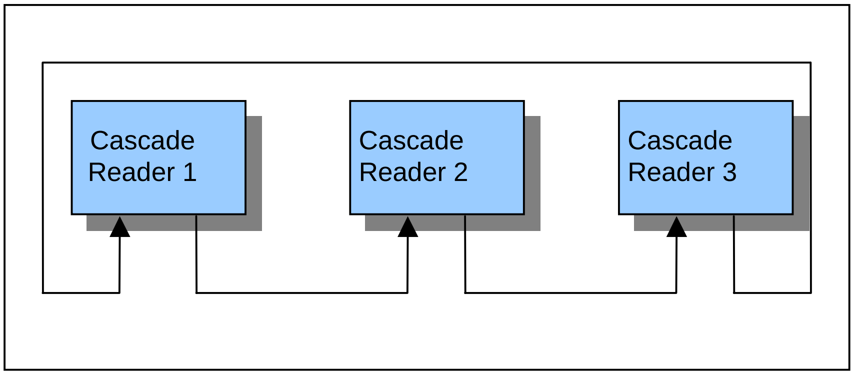
Host control
Master-slave control
Cascaded control

Host control means the readers are explicitly controlled with single commands from a host. This prevents more than one reader operating at any one time, assuring there is no interference between readers. Using a fast host, readers can be polled in a round-robin manner. However, the readers cannot be exercised at their full reading speed since continuous mode cannot be used on the reader.

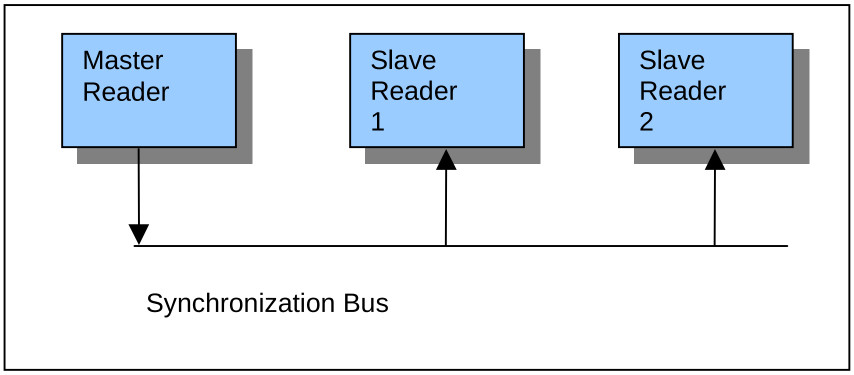
Master-slave control defines an architecture where the reader units are linked together via a separate synchronization line, with one reader acting as the master. The master issue triggers for pre-sent requests under control of a host, allowing all the readers to execute their RF request simultaneously.

In combination with Continuous mode, this allows maximum speed of operation. However: For complex operations, such as reading using SID, it is not possible to synchronize each reader fully. In this case Cascaded control should be used, where each reader passes control to the next after it has completed its task. This is effectively a round-robin control, but is much faster than single host control since it is controlled at a reader level.

Cascaded Reader Synchronization



Master-Slave Reader Synchronization



For both Master-slave and Cascaded control, the procedure to setup and run commands under the Host Protocol is similar; the required request is sent to each reader in the synchronization chain in turn, with the relevant synchronization method selected in the Synch code in the message, and, if required, Continuous mode selected to enable free-running of the reader units.

All reader units having received this request will wait for the synchronization trigger, which is sent by the host in the form of a Synch start request to the master reader, or the first cascaded reader. That reader then initiates the request, triggering further readers as defined by the selected synchronization mode.

The readers can be halted by sending a Stop continuous request to the master reader, or first cascaded reader.

Please refer to the reader hardware manual for details of the synchronization cable connections.

Appendix C

Reader Setup Parameters

Commonly used setup parameters are handled dynamically via the Service code and Message code flags, allowing these setting to be varied by the host controller. Less used settings may be programmed into flash memory using the on-board flash programmer. This uses a proprietary protocol for compactness, and is supported by the Tag-it Navigator.

Group	Parameter	Description	Bytes	Values	Default
-	Setup descriptor	defines setup parameter	1	1 - FF$_{16}$	-
ASP	Baudrate		1	0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200	4
Transponder protocol	Get retries	number of attempts after failed get	1	0 - 5	0
	Put retries	number of attempts after failed put	1	0 - 5	0
	Retry control	settings for tailoring retry actions	1	bit 0 - 0 = no retry on first block of multiblock bit 0 - 1 = retry on first block of multiblock	0
	Continuous delay	delay between executing request in continuous mode	2	0 - FFFF$_{16}$ ms	0
	Compound error	compound request error control	1	0 = terminate compound request at error 1 = always complete compound request	0

Appendix D

Firmware Programming

The reader firmware resides in flash memory on-board the reader DSP. Firmware upgrades may be performed using the Tag-it Navigator, which supports the protocols for executing memory control commands, loading firmware and setup data, and programming the flash memory. Firmware updates may be obtained from the Texas Instruments web site.

Appendix E

Glossary

ASP	Asynchronous serial port
BCC	Block check character. A character calculated on the data forming a message, and sent with the message for error detection purposes. The BCC may be a CRC or LRC.
Block	This is the smallest unit of user data that can be ‘read’ and ‘write’ using the standard messages. It may vary in length between different transponder versions.
CRC	Cyclic redundancy check
DSP	Digital signal processor
FCC	U.S. Federal Communications Commission
Host	The host responsible for controlling the reader. This may be a microcontroller, PC, PLC, or any other type of controller.
Host Protocol	Protocol for communication between the host controller and the reader.
LRC	Longitudinal redundancy check
LSB	Least significant bit.
LS Byte	Least significant byte
Message	A Host Protocol message. Can be a response or a request.
MSB	Most significant bit.
MS Byte	Most significant byte
Request	A Host Protocol message from the host to the reader
Response	A Host Protocol message from the reader to the host
SID	Simultaneous identification. A method of avoiding message collisions to enabling multiple transponders to communicate simultaneously with the reader.
Transponder Protocol	Defines the protocol used to communicate primarily between transponder and reader at a logical level. This includes protocol data structure, message definitions and sequences, error control. While the details of the RF protocol are covered in the system specification, the relevant facets such as data encoding and data rates are included in the Transponder Protocol definition.