Texas Instruments

Tag-it™ Transponder Protocol

Reference Manual

SCBU032 (11-04-21-002) March 2000

TIRIS Technology by Texas Instruments™

Edition Two - March 2000

This is the second edition of this manual.

Read This First

About This Manual

This manual describes the RF data protocol used for communication between Tag-it compliant reader systems and TI Tag-it transponders which are based on the 37110 chip family (Product codes RI-I∗∗-110∗-∗∗ where ∗ may be any character).

The manual contains the following chapters:

Chapter 1: Introduction and Basics

Chapter 2: Transponder Protocol Structure

Chapter 3: List of Commands

Purpose

Designed for Texas Instruments TIRIS development partners, this document is the Transponder Protocol reference manual for the Tag-it™ Reader-to-Transponder communications software protocol. In addition, the Host Protocol Reference Manual for Host-to-Reader communications is available on request or through the Internet Documentation Center at http://www.tiris.com .

Audience

This reference manual is designed for use by TI partners, primarily engineers who have experience with TIRIS and radio frequency identification (RFID) components.

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}

In byte representations, the least significant bit (LSB) is bit 0 and the most significant bit (MSB) is bit 7.

Explanation of Symbols and Warnings

The following pictograms and designations are used in this document:

WARNING:

A WARNING IS USED WHERE CARE MUST BE TAKEN OR A CERTAIN PROCEDURE MUST BE FOLLOWED, IN ORDER TO PREVENT INJURY OR HARM TO YOUR HEALTH.

CAUTION:

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

Note:

Indicates conditions which must be met, or procedures which must be followed, to ensure proper functioning.

Information:

Indicates information which makes usage of the equipment software easier.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Topic	Page

1.1 System Components

The Tag-it ™ system comprises a reader and associated transponders. The reader is controlled by a host system which may be a PC, a larger computer, or some other type of intelligent device, for example a ticket printer. This chapter describes the system environment, the system components and how the RF system works.

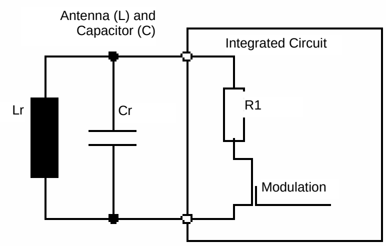
1.1.1 Tag-it Reader

A typical Tag-it reader consists of an electronic board and an antenna. The board can be either battery or mains powered and is connected to its host (Application Processor) through a serial connection or a local area network. It sends energy and commands to, and receives signals from, the transponder through the antenna (aerial).

1.1.2 Transponder

A Tag-it transponder comprises an antenna, a resonance capacitor and an integrated circuit. In this case, the transponder IC is powered by the electromagnetic field generated by the reader's antenna, this being generally described as a passive transponder.

Figure 1: Typical Tag-it Transponder



The IC is mounted onto the antenna by means of flip-chip technology or conventional wire bonding, figure 1 showing a typical transponder circuit. The resonance capacitor (Cr) may be either internal (included in the IC) or external, i.e. mounted on the transponder. The electrical parameters of the components, the antenna (including its inductance), the resonance capacitor and the layout of the antenna loops all have an effect on the transponder performance. To provide optimal performance, the Lr/Cr resonance circuit can be trimmed to the required target frequency using externally mounted components.

1.1.3 System Overview

As previously stated, an RF System consists of the transponder, the reader and its antenna. The operating frequency is 13.56 MHz. This frequency is the center frequency of the 13.56 MHz ISM frequency band with a defined bandwidth of ±7 kHz, this being used to supply the tag with operating power and to maintain the data communication between transponder and reader.

During a transaction, the transponder is permanently powered by the signal sent by the reader. This signal also provides the clock frequency for the transponder. It is amplitude modulated to transmit requests from the reader to the transponder.

In order to transmit responses to the reader, the transponder derives the operating energy and the required clock from the reader's power signal to generate its modulation frequencies.

The modulation depth is a system parameter and is a compromise between power management (storage capacitor on chip) and the target operating range of the (passive) transponders. It is influenced by the coupling factor between the reader's antenna and the transponder, and thus by the distance between them.

1.2 Transponder Protocol Overview

Note:

This is a generic document and does not relate to any specific Tag-it product. Some of the functions described here may not be implemented in all Tag-it products.

1.2.1 Description

The Tag-it transponder protocol is half-duplex, the fundamental operation being a transaction which consists of:

- a request sent by the reader to the transponder
- a response sent back by the transponder to the reader

Both the request and the response contain a Command Code which specifies the operation to be (or that was) performed by the transponder.

The transponder NEVER initiates a response without having been instructed to do so by the reader. The request must have been fully understood by the transponder before it can respond. The presence of the 13.56 MHz carrier frequency will power up the transponder but does not generate a spontaneous emission (response) of any kind by the transponder.

This mechanism is particularly useful when transponders of different technologies which may happen to use the same 13.56 MHz frequency are present in the reader field. Tag-it transponders will not start any emission without having been instructed to do so, allowing other transponders to be interrogated according to their own mechanism.

If a similar principle is used by the other technologies present, Tag-it transponders can be handled without external interference.

1.2.2 Principle of Operation

Each Tag-it transponder has a unique address which is factory-programmed and 32 bits long, thus allowing an address range of more than 4 billion individual addresses. The possibility cannot be excluded that at some future time, a previously used address is assigned to a new transponder. Considering the expected lifetime of transponders however, and based on statistical calculations, the probability that 2 transponders with the same address are present simultaneously in the same reader location is well below 10^{-10}.

If several transponders are expected to be present in the read area (this is application dependent), the first step is to inventory them. This is done by the Simultaneous Identification (SID) mechanism described in Section 3.4 which results in the reader storing in its memory the addresses of each transponder present within its range. At this point, the reader may pass them on to the application processor.

Once each transponder has been inventoried by its unique individual address, individual transactions can be performed in a manner similar to the use of a phone number or the addressing of a workstation on a LAN by its address.

A transaction is carried out with a single transponder, which is identified by its address. The transaction may read an area of its memory, write to another area or process a more advanced feature.

1.2.3 Addressed Operations

In addressed mode, at a given point in time, a reader dialogs with only one transponder.

1.2.4 Non-Addressed Operations

If an application is so organized that one and only one transponder can be present in the reader field, it is possible to use the non-addressed mode. In this case, any transponder receiving a non-addressed request will execute it and send a response.

It is key to a successful implementation of the non-addressed operation that only one transponder is within the reader's range. Otherwise, the following may occur:

- If the reader is trying to perform a read function (obtaining information from the transponder), 2 or more answers will be sent by transponders. These answers will collide, resulting in an unintelligible message at the reader's side.
- If the reader is performing a write function (writing information to a memory area of the transponder), all reachable transponders will perform this function, and for the reasons stated above, their multiple responses will collide, such that the reader will not receive a clear confirmation that the operation has been performed correctly. This can result in severe data corruption or make the transponder unusable if the Lock Command has been used.

1.2.5 Transponder Version

Tag-it products have been designed as a family of products based on a common technology. This therefore means that transponders equipped with different IC's, each IC having different features, may be present at the same time in a reader's field.

It is possible, for instance, to find transponders with 256 bits of memory organized in blocks of 32 bits and other transponders with 1024 bits of memory organized in blocks of 128 bits, and still others equipped with enhanced security features.

These different types of transponders could be used for the same application (the application has adopted the most up-to-date products, but is however still using products from a previous generation) or for different applications (each application has chosen the product with the most suitable features).

In order to know what the characteristics of each transponder IC are, and thus be able to correctly request execution of commands, each Tag-it IC is programmed during manufacturing with its version number, its manufacturer code and additional information about memory size and structure.

By requesting this information from the transponder using a command supported by all Tag-it IC's, the reader can determine the supported features and act according to the IC capabilities.

This guarantees that in a given application, upward compatibility with new products can be achieved, thus allowing each application to take advantage of any new features to improve the overall performance. Across applications, products with different features can co-exist and be handled by a common reader.

It is the responsibility of the reader and/or the application processor to know by reading the IC version number what the transponder's detailed characteristics are. As more Tag-it chip families will become available, these characteristics will be described on the tag datasheet(s).

1.3 Reader-to-Transponder Communication

1.3.1 Method

Table 1: Downlink Parameters (Reader-to-Transponder)

	Symbol	Note	Min	Nom	Max	Unit
Transmitter frequency	F_{TX}		13.557 -200ppm	13.560	13.563 +200ppm	MHz
Charge Up Time	T_{CHG}		300	-	-	µsec
Pulse Pause	T_{OFF}		7.5	-	11.1	µsec
Pulse Length, Code Violation	T_{CV}		39	50	75	µsec
Pulse Length, Data High-Bit	T_{DH}		139	150	175	µsec
Pulse Length, Data Low-Bit	T_{DL}		89	100	125	µsec
Programming Burst Length	T_{PROG}		10	-	-	msec
Modulation Depth ASK 100%	m		-	100	-	%
Down-Link Baud rate		Data Bits	5.71		11.2	kbps

Reader to transponder communication is achieved by amplitude modulation of the carrier frequency. This technique allows the reader to power the transponder while transmitting sequences of binary coded data.

Pulse width modulation determines whether the bit being transmitted is 0 or 1, this depending on the length of the pulse (how long the carrier frequency is present). The bits are separated by a fixed duration pause (no carrier).

High Bits (1) are coded by a power burst of 150 µs (see figure 2).

Low Bits (0) are coded by a power burst of 100 µs (see figure 3).

This technique allows an easy and simple implementation both at the reader and on the transponder's IC. The timing is as follows:

Table 2: Downlink Parameter Clock Cycles

Signal	No. of Cycles@13.56 MHz	Nominal Timing
Reference	1 cycle	74 ns
T_{OFF}	150 cycles	11.1 ≈ 11µs
T_{CV}	675 cycles	49.9 µs ≈ 50 µs
T_{DH}	2024 cycles	149.8 µs ≈ 150 µs
T_{DL}	1356 cycles	100.3 µs ≈ 100 µs

Figure 2: High Bit

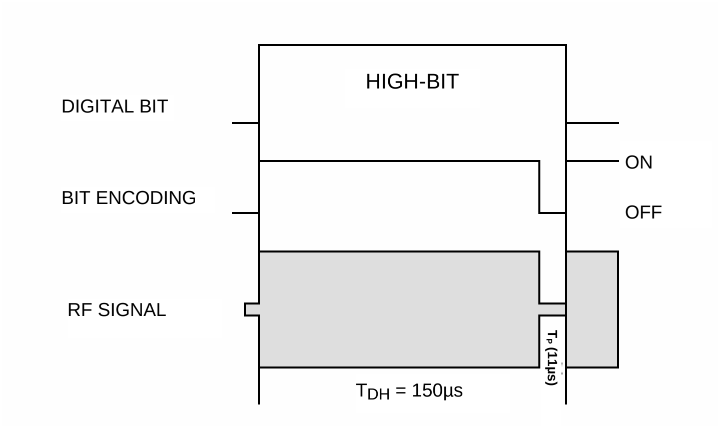
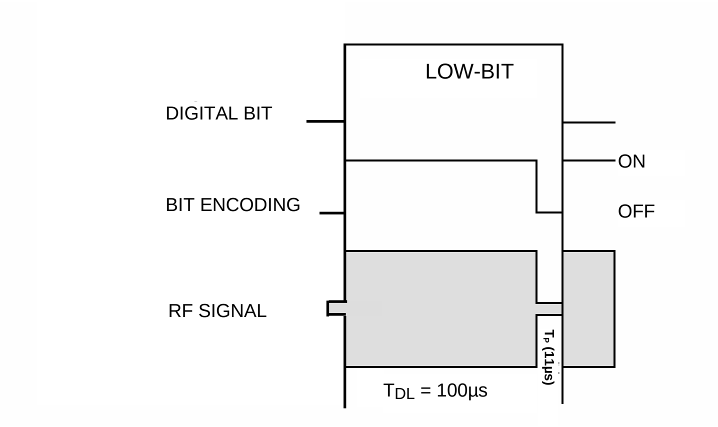


Figure 3: Low Bit

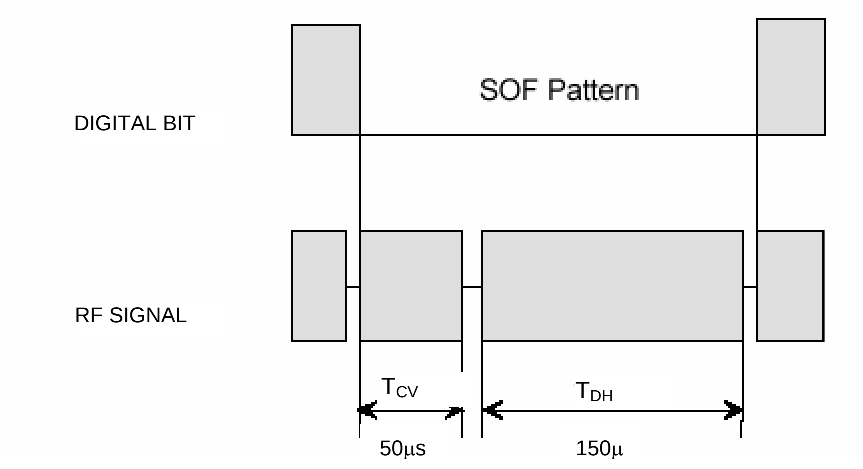


1.3.2 Framing

The framing is comprised of a Start Of Frame (SOF) and an End Of Frame (EOF).

The SOF consists of a 50 µs power pulse, followed by a standard duration pause, followed by a 150 µs power pulse as shown in figure 4.

Figure 4: Start of Frame Synchronization Pattern (Reader-to-Transponder)



The EOF consists of a 50 µs power pulse, followed by a standard duration pause, followed by a 100 µs power pulse as shown in figure 5.

Figure 5: End of Frame Synchronization Pattern (Reader-to-Transponder)

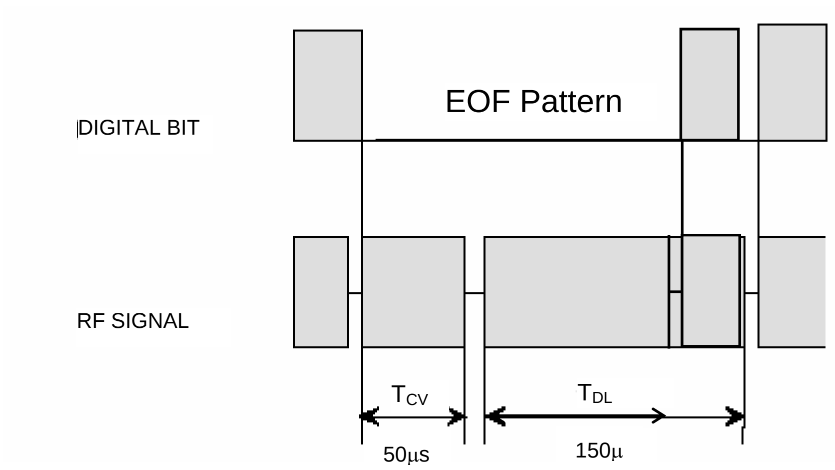
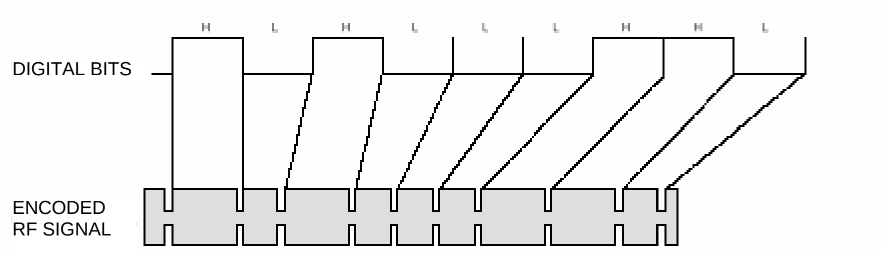


Figure 6: Example of the Transmission of a Sequence of Data Bits

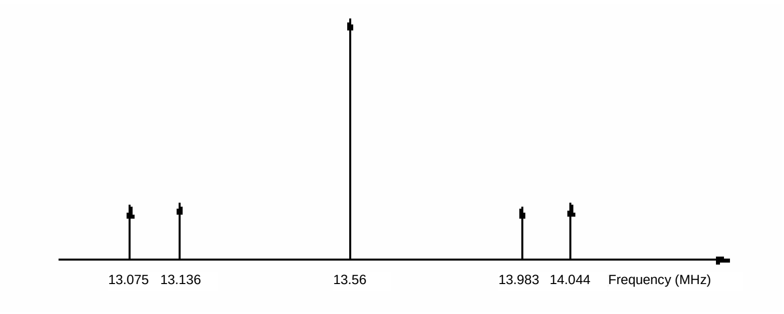


1.4 Transponder to Reader Communication

1.4.1 Parameters

Carrier Frequency:	13.56 MHz
Modulation:	Field load modulation
Encoding:	Manchester
Frequency modulation:	423.75 kHz (Manchester High level) 484.29 kHz (Manchester Low level)
Transmission rate:	26.7 kBaud
Bit length:	37.42 µs
Modulation depth:	Refer to specific transponder data sheet

Figure 7: Transponder-to-Reader Frequency Spectrum



1.4.2 Method

Data transmitted from the transponder to the reader is encoded using the Manchester encoding technique, this being a well known technique used in serial data transmission and local networks. The main advantage of this type of encoding is that the clock information is embedded into the encoded data signal.

A logical “1” data bit is represented by a LOW-to-HIGH transition as shown in figure 8 and a logical “0” data bit is represented by a HIGH-to-LOW transition as shown in figure 9.

The divider ratios and the bit lengths are:

T_{MH} = 8c* T_H = 8*2.36 µs = 18.88 µs

T_{ML} = 9c* T_H = 9*2.06 µs = 18.54 µs

T_{MB} = T_{ML} +T_{MH} = 18.88 µs+18.54 µs = 37.42 µs

Bit length T_{bi} = 37.42 µs ; (1/T_{bit}) = 26.7 kHz

A typical data stream is shown in figure 10.

Figure 8: Data Encoding HIGH Bit (Transponder-to-Reader)

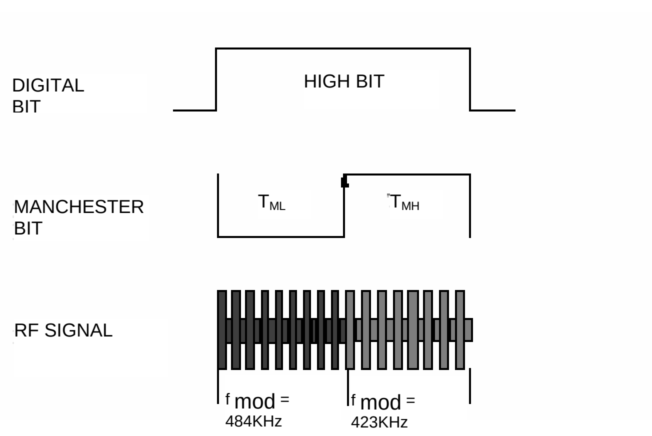


Figure 9: Data Encoding LOW Bit (Transponder-to-Reader)

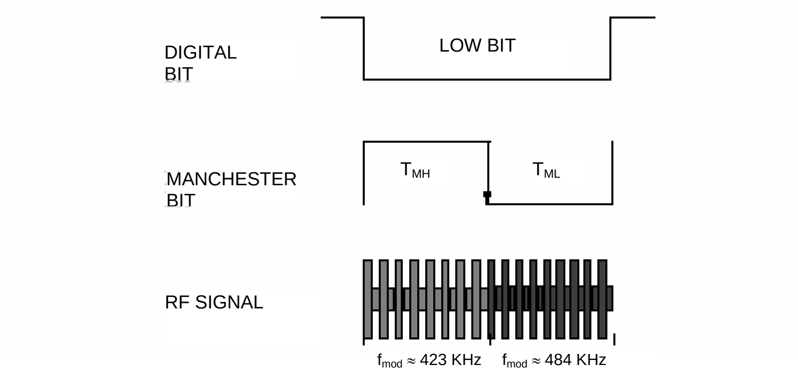
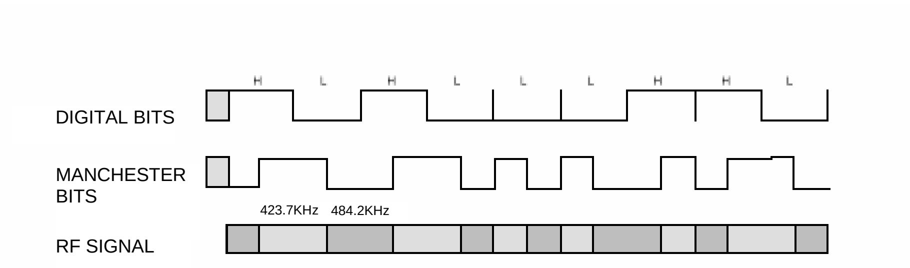


Figure 10: Data Encoding Example (Transponder-to-Reader)



1.4.3 Framing

The Transponder-to-Reader framing technique is based on the principle generally described as special pattern. This method is reliable, easy to implement in the transponder IC and easy to detect.

The Start Of Frame (SOF) comprises a sequence of:

2 high bits
A frequency burst of 1.5 bits length at a Manchester frequency of 484 kHz followed by a further 1.5 bits length at a Manchester frequency of 423 kHz (this is the special pattern sequence)
1 high bit

The End of Frame (EOF) comprises a sequence of:

1 low bit
A frequency burst of 1.5 bits length at a Manchester frequency of 423 kHz followed by a further 1.5 bits length at a Manchester frequency of 484 kHz (this is the special pattern sequence)
1 low bit

The Start and End Of Frame patterns are shown in figures 11 and 12.

Figure 11: Start of Frame Synchronization Pattern (Transponder-to-Reader)

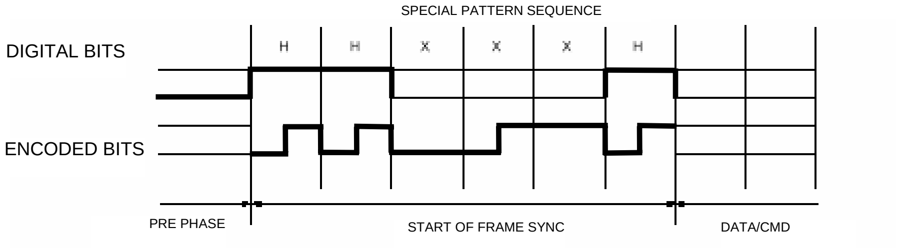
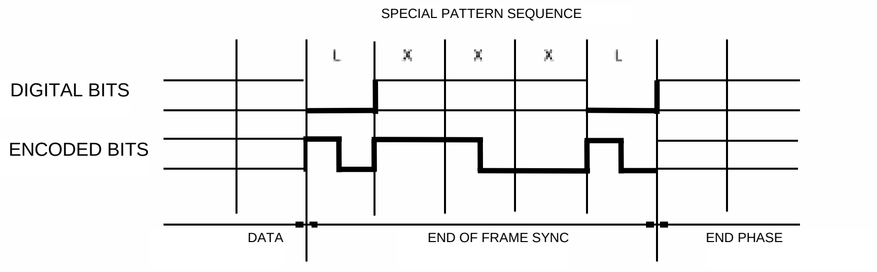


Figure 12: End of Frame Synchronization Pattern (Transponder-to-Reader)



1.5 Simultaneous Identification (SID)

The SID mechanism offers the capability to inventory in a very short time a large number of transponders by their unique address, provided they are within the reader operating range.

The SID mechanism is based on an algorithm handled by the reader:

- Each transponder has a unique address.
- Transponders are interrogated by a sequence of SID Requests, using a special addressing scheme.
- If two or more transponders answer to the same interrogation, the collision is detected and stored into reader's memory.
- If only one transponder answers, its address is registered into the reader's memory. It will not be interrogated again during the current SID cycle.
- The reader then executes the collision handling algorithm, which results in the interrogation of transponders not yet identified by a modified addressing scheme.
- The cycle is continued until no collision is detected, i.e. all transponders have been identified and their address stored in the reader's memory. The inventory of transponders is then complete.

The collision-handling algorithm theoretically permits a 100% probability that all the transponders that are present in the reader’s range are be detected. However, in real applications, mutual influence of transponders may affect the result.

If only one transponder is present, a single SID Request is needed.

If 2 or more transponders are present, the average number of SID Requests per transponder stabilizes at around 0.40. While the number of SID Requests per transponder varies from one set of transponders to another, simulation has shown that the maximum never exceeds 0.75, and may even decrease when the number of transponders increases. The minimum may be as low as 0.10, although this is of lesser importance even if it improves the average performance.

The fact that the maximum never exceeds 0.75 whatever the number of transponders is means that all transponders are guaranteed to inventory within a time proportional to the number of transponders, irrespective of how many transponders there are.

The SID mechanism is deterministic and linear. It is reliable and reproducible, as opposed to, for instance, other mechanisms based on physical parameter random deviations.

It allows the inventory of any number of transponders within a time proportional to the number of transponders present. Only physical influences, i.e. the coupling of resonant circuits, can limit the performance in practice.

Some other advantages are:

The transponders have a unique factory programmed address.

Transponders do not need to be powered between two SID Requests. Even if the reader keeps powered on, transponders moving in various orientations within the reader's antenna field may not receive enough power to keep temporary information in memory. Since this is not required however, the whole process being handled by the reader, the impact of such an occurrence is drastically decreased.

A Quiet Request may be used to ask specific transponders to ignore any subsequent SID Request. This may be useful if it is found that some collisions could not be detected because the signal strength of the closest transponders is so much higher than that of more distant transponders. In this case, once the closest transponders have been inventoried, they are asked to stay Quiet, and another SID sequence is performed, to which only transponders not yet inventoried (with a possibly weaker signal) will respond, but without the risk of having this weaker signal masked by a stronger signal from a closer transponder. Again, this is only a matter of method selection in the reader software and thus offers optimal flexibility in implementation.

The detailed mechanism is explained in section 3.4.

1.6 EEPROM Programming Burst

Current Tag-it ICs use an EEPROM memory technology. This requires that after all requests requiring some writing action in the transponder, for example a Put Block Request, the powering carrier is kept on for a given time (Tprog). The required value for this can be found in the IC datasheet, the order of magnitude being around 10 ms.

Following the programming burst, the carrier must be switched off for the defined Interbit Pause time, indicating the end of the programming cycle to the transponder, and initiating the response, which follows after the defined receiver recovery time.

1.7 Receiver Recovery Time

Once the reader has sent a complete request, its electronic circuits need time to switch to the receiving mode.

In order to allow the reader to perform this switch, the transponder IC implements a time delay (T_{REC}) before sending its response. If no answer has been received by the reader after this time, it indicates that the transponder has probably not received or correctly processed the request.

Note that for commands involving a programming burst (Put, Lock), responses may occur at two possible points:

- After the request plus receiver recovery time. This occurs if an error is generated, such as Block Already Locked. In this case the request can be aborted before the programming burst is carried out.
- A correct response or error response may otherwise occur after the programming burst, following the Carrier Off Interbit Pause plus receiver recovery time.

The receiver recovery time for all commands except SID Poll is T_{REC1}. SID POLL uses time T_{REC2} for SID slot 0 and T_{REC3} for slots 1 to 15.

Table 3: Uplink Parameters (Transponder-to-Reader)

	Symbol	Min	Nom	Max	Unit
Receiver Recovery Time All commands except SID Poll	T_{REC1}	311.0	312.0	313.0	µsec
Receiver Recovery Time SID Poll slot 0	T_{REC2}	310.5	311.0	312.0	µsec
Receiver Recovery Time SID Poll slots 1 – 15	T_{REC3}	315.0	315.5	316.0	µsec

Figure 13: Transaction Timing: Put Command

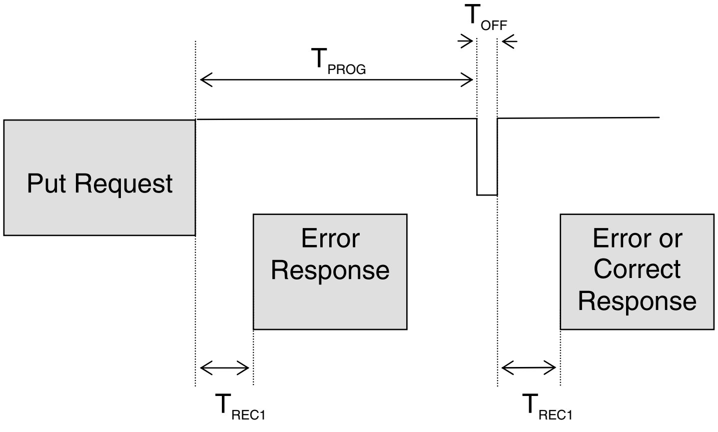
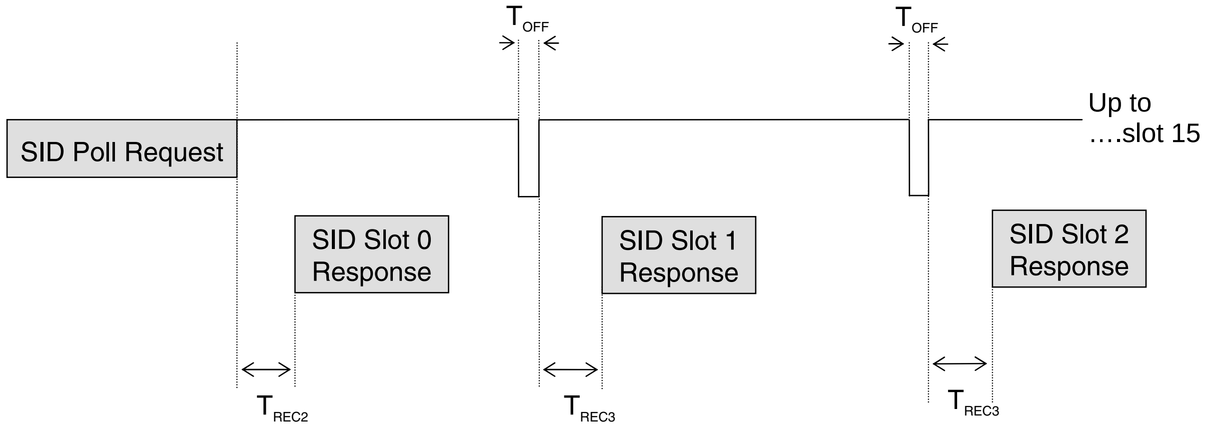


Figure 14: Transaction Timing: SID Poll Command



1.8 Memory Organization

The Tag-it IC user memory is typically organized in blocks (or pages) with each block individually addressable. All blocks in an IC have the same size which can be found in the relevant IC datasheet. The block size and the number of blocks can also be retrieved by the Get_Version Command.

In addition to the user memory, service memory and information memory are implemented, either in programmable form or in mask-programmed form.

Service memory is provided to store information about the memory, for instance, to store the locking status of the blocks.

Information memory is provided in order to store information about the IC, for instance the SID address and the IC version and some elements of its characteristics.

1.9 Block Locking

A block can only be locked once. This can be done when the IC is manufactured (factory-locked), or by the user at any time (user-locked).

A locked block (regardless of which method was used to lock it) cannot be modified by any subsequent command. It is permanently locked and the data contained in that block cannot be changed.

Two bits of service memory are reserved for each block to store information about its lock status.

1.9.1 User Block Locking

Any block can be locked once by the user. This is especially useful if it is desired to permanently program information about the object or the person the transponder is related to. This could be, for example, a serial number, a manufacturing date, or a driving license number. Temporary information such as a flight number or the next destination, which may need to be modified, should not be locked.

1.9.2 Factory Block Locking

It is only possible to factory lock a block during the manufacturing process, it cannot be done later. This can be useful to guarantee that specific transponders are for use only by a specific customer or application.

It is possible, for instance, to factory lock the first block after the primary user code (for example: an airline code) has been programmed. This guarantees that all transponders with the specific user code in the first block and having the same first block factory locked have been manufactured and delivered to the airline.

Factory locking is subject to specific conditions, therefore please contact a Texas Instruments sales office in case of inquiries.

CHAPTER 2

Transponder Protocol Structure

Topic **Page**

2.1 Command Structure

The reader issues a request to the transponder in the form of coded instructions and parameters. The transponder replies with a response block containing status information and data.

2.2 Frames Overview

Frame structures are used to format the data communication between the reader system and a transponder. The format for request and response are very similar, however the response frame has additional error status fields.

Table 4: Transponder Request Frame Structure

Request Frame	Bit Length	Bit Position
Request Code	2	0-1
Command Code	8	2-9
Format Type	1	10
Address Flag	1	11
Reserved	1	12
SID Address (optional)	32	13-44
Parameters	256 bits max	45-…
CRC	16	depends on previous fields

Note:

The request frame has a variable format depending on the address flag. If this field value is set to 0, then the SID address field will not exist.

Table 5: Transponder Response Frame Structure

Response Frame	Bit Length	Bit Position
Response Code	2	0-1
Command Code	8	2-9
Format Type	1	10
Address Flag	1	11
Reserved	1	12
Error Flag	1	13
SID Address (optional)	32	14-45
Error Code (optional)	8	46-53
Data	256 bits max	54
CRC	16	depends on previous fields

Note:

The response frame has a variable format depending on the address flag and the error flag. If these field values are set to 0, then their associated fields SID Address or Error Code will not exist.

Table 6: Transponder Request Protocol Frame

Transponder Request Protocol Frame			
Field Name	**Bits**	**Definition**	
Request Code	2	00_2 = This is a request from the Reader to the Transponder	
Command Code	8	The Reader indicates which command the Transponder should perform. For parameter and data details, refer to Transponder Commands.	
		01_{16}	Get_Block: Request that the Transponder send the data contained in the specified block.
		03_{16}	Get_Version: Request for the Transponder to send version of its IC.
		05_{16}	Put_Block: Request for the Transponder to program data into the specified block in its memory.
		07_{16}	Put_Block_Lock: This is a combination of the Put_Block and the Lock_Block Command.
		08_{16}	Lock_Block: Request for the Transponder to permanently user-lock the specified block.
		$0A_{16}$	SID_Poll: Request for all Transponders in the Reader's antenna field to send back their SID address and chip characteristics.
		$0B_{16}$	Quiet: Request for a specific Transponder, as specified by the Address field, to "stay quiet" until a request other than an SID_POLL is received.
Format Type	1	Indicates the format of the request. Only fixed format is supported at this time.	
		0_2	= value must be set to zero for fixed format.
Address Flag	1	Indicates whether or not the Request is addressed to a specific Transponder.	
		0_2	= Non-Addressed
		1_2	= Addressed
Reserved	1	0_2 = value must be set to zero	
SID Address	32	Address of the Transponder to which the Request is sent. This field should only be present if the Address Flag field is set to "1".	
Parameters	256 max	Request dependent. For the parameter and data details, refer to the Transponder Commands Descriptions.	
CRC	16	The calculated value of the CCITT CRC.	

Table 7: Transponder Response Protocol Frame

Transponder Response Protocol Frame			
Field Name	**Bits**	**Definition**	
Response Code	2	11_{16} = This is a Response from the Transponder to the Reader.	
Command Code	8	The Transponder responds with the Command Code it performed. Corresponding data is found in the parameter and data field. For the data layout, refer to Transponder Commands.	
		01_{16}	Get_Block: the Transponder responds with data from the specified block.
		03_{16}	Get_Version: the Transponder responds with the version information of its IC.
		05_{16}	Put_Block: the Transponder responds after it stored program data into the specified block in its memory.
		07_{16}	Put_Block_Lock: this is a combined Put_Block and Lock_Block Command.
		08_{16}	Lock_Block: the Transponder responds after it permanently sets the user-lock on the specified block
		$0A_{16}$	SID_Poll: the individual SIDs and IC characteristics from all the Transponders in the Reader's antenna field.
		$0B_{16}$	Quiet: the Transponder does not Respond until it receives an addressed Request.
Format Type	1	The Transponder indicates the format of the Request. Only fixed format is supported at this time.	
		0_2	= value must be set to zero for fixed format
Address Flag	1	The Transponder sets this flag to indicate whether the Address field will contain the address of the Transponder sending the Response.	
		0_2	= Non-Addressed
		1_2	= Addressed
Reserved	1	0_2	= Value must be set to zero
Error Flag	1	The Transponder indicates whether the Request was successful.	
		0_2	= Success. The Request was completed successfully.
		1_2	= An error occurred. The error code is present in the Error Code field.
SID address	32	Contains the address of the Transponder which is sending back the Response. This field will only be present if the Address Flag field is set to "1". This can be used to confirm the identity of the Transponder.	
Error Code	8	The Transponder returns the error code. This field will only be present if the Error Flag field is set to "1".	
Data	256 max	Request dependent. For details, refer to Transponder Commands.	
CRC	16	The calculated value of the CCITT CRC.	

CHAPTER 3

List of Commands

Topic **Page**

3.1 Command Code Reference Tables

Table 8: Transponder Protocol Command Codes

Transponder Command Code Summary		Addressed Mode	
	Command Code	Addressed mode possible	Non-Addressed mode possible
Get_Block	01_{16}	✓	✓
Get_Version	03_{16}	✓	✓
Put_Block	05_{16}	✓	✓
Put_Block_Lock	07_{16}	✓	✓
Lock_Block	08_{16}	✓	✓
SID_Poll	$0A_{16}$	✗	✓
Quiet	$0B_{16}$	✓	✓

Blocks are numbered starting at 0 through to 255, allowing a maximum of 256 blocks to be used. Each block may be locked, making the contents unchangeable. The Lock Flag shows the status for the block.

Table 9: Lock Flags

Lock Flag		
Flag Value	Lock Status	Description
00_2	Unlocked	Block contents may be changed
01_2	User-locked	Irreversibly locked by user. Cannot be changed
10_2	Factory-locked	Irreversibly locked in production. Cannot be changed
11_2	Reserved	-

Most commands can be executed in addressed or non-addressed mode. The exception is SID_Poll, which may only be non-addressed. It is not recommended to use the Quiet command in non-addressed mode, as this will turn off all transponders in the field.

The examples shown use a block size of 32 bits and the number of blocks is 8.

All data forming command requests and responses is transmitted in a defined bit order starting with the first bit of the Request Code or Response Code. Since the protocol is bit-oriented there are no defined MSB or LSB values.

3.2 Transponder Message Error Codes

Table 10: Transponder Message Error Codes

Transponder Message Error Code Reference (valid when Error Flag =1)	
Error Code	Definition
Error Codes 00_{16} - $0F_{16}$ are general error codes which can be sent with any response.	
00_{16}	Reserved
01_{16}	Command not supported
02_{16} - $0F_{16}$	Reserved
Error Codes 10_{16} – FF_{16} are specific error codes.	
10_{16}	The specified block is not available (doesn't exist).
12_{16}	The specified block is already user-locked and thus cannot be locked again.
16_{16}	The specified block was not successfully programmed. A common source of this error is that the transponder did not receive enough energy to program the non-volatile memory. This is likely to occur if the transponder is too far away from the reader's antenna.
18_{16}	The specified block was not successfully locked. A common source of this error is that the transponder did not receive enough energy to program the non-volatile memory. This is likely to occur if the transponder is too far away from the reader's antenna
$1F_{16}$	Command not allowed

In the case of one of the following error conditions, the transponder will not respond:

- CRC mismatch (frame error)
- Not enough bits, but CRC ok and command implemented
- Too many bits, but CRC ok and command implemented
- Unknown command, regardless of CRC status

Otherwise, if an error was detected within a valid frame, the transponder returns a command specific error code (> $0F_{16}$) or a general error code (< 10_{16}).

The error code is returned as the first byte of data in the response except that it is always preceded by the address if the command is executed in addressed mode.

3.3 Commands

3.3.1 Get_Block Request

Description: Request for the transponder to send the data contained in the specified block.

Example: 00_2 01_{16} 0_2 1_2 0_2 00 01 8B $1A_{16}$ 02_{16} $DA39_{16}$

Table 11: Get_Block Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	01_{16}	Get_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID address (optional)	32	00 01 8B $1A_{16}$	Transponder address is $00018B1A_{16}$
Parameters	8	02_{16}	Block Number 2
CRC	16	$DA39_{16}$	Calculated value of the CCITT CRC.

Required Parameters:

Block Number**:** enter the block number in the Parameters field

Note:

The number of blocks in the Transponder and their size can be obtained by the Get_Version Request, or if the chip version is known, by referring to the product datasheet.

There can be as many as 256 blocks in the chip’s memory. The actual number of blocks and the block size are dependent on the memory organization. The block size is determined by the chip version.

3.3.2 Get_Block Response

Description: Response from the transponder with the data contained in the requested block.

Example: 11_2 01_{16} 0_2 1_2 0_2 0_2 00 01 8B $1A_{16}$ 02_{16} 00_2 11 F2 7B 45_{16} $C0C1_{16}$

Table 12: Get Block Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A response from the transponder to the reader
Command Code	8	01_{16}	Get_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in addressed mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
SID address (optional)	32	00 01 8B $1A_{16}$	Transponder address is $00018B1A_{16}$
Data	8	02_{16}	Block Number 2
	2	00_2	Lock Status is Unlocked
	Block size	11 F2 7B 45_{16}	Data (32 bits) from Block Number 2
CRC	16	$C0C1_{16}$	Calculated value of the CCITT CRC.

Returned Data:

Block Number: Confirms the data block that is being returned

Lock Status: The current lock status of that block of data

Block Data: The data contained in the requested block. The size is the transponder chip block size.

Note:

The number of blocks in the Transponder and their size can be obtained by the Get_Version Request, or if you know the Chip Version, by referring to the product Datasheet.
There can be as many as 256 blocks in the chip memory. The actual number of blocks and the block size are dependent on the memory organization.

Specific Error Codes	
Definition	Code
Block not available	10_{16}

3.3.3 Get_Version Request

Description: Request for the transponder to send back version information.

Example: 00_2 03_{16} 0_2 1_2 0_2 00 01 8B $1A_{16}$ $FFE0_{16}$

Table 13: Get_Version Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	03_{16}	Get_Version Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID address (optional)	32	00 01 8B $1A_{16}$	Transponder address is $00018B1A_{16}$
CRC	16	$FFE0_{16}$	Calculated value of the CCITT CRC.

Required Parameters: There are no parameters for this request.

3.3.4 Get_Version Response

Description: Response from Transponder with SID and chip characteristics.

Example: 11_2 03_2 0_2 1_2 0_2 0_2 00 01 8B $1A_{16}$ 0000001_2 000000101_2 000_2 00011_2 07_{16} $88F4_{16}$

Table 14: Get_Version Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A Response from the Transponder to the Reader
Command Code	8	03_{16}	Get_Version Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
Data	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
	7	0000001_2	Chip Manufacturer Code (1 = Texas Instruments)
	9	000000101_2	Chip Version 5
	3	000_2	Reserved
	5	00011_2	Block Size (in bytes) minus one (3 + 1 = 4 bytes = 32 bits)
	8	07_{16}	Number of Blocks minus one (7 + 1 = 8 blocks)
CRC	16	$88F4_{16}$	Calculated value of the CCITT CRC.

Returned Data:

Transponder's SID address

Chip Manufacturer Code ('0' is excluded, allowing for a maximum of 127)

Chip Version: the Chip Version is specific to each Manufacturer, so the combined fields of Chip Manufacturer + Chip Version is a unique combination.

Block Size minus one: indicates the Transponder Chip's block size in bytes.

Number of Blocks minus one: indicates the number of data blocks on the chip.

If the request is addressed, the response parameter SID Address is not included in the response data, since it is already sent as the address.

Specific Error Codes	
Definition	Code
-	-

3.3.5 Put_Block Request

Description: Request for the transponder to store a block of data in a specified block.

This command must be followed by a Programming Burst as with all Put/Lock commands (see section 1.6)

Example: 00_2 05_{16} 0_2 1_2 0_2 00 01 8B $1A_{16}$ 03_{16} 07 00 AA AA_{16} $C4C7_{16}$

Table 15: Put_Block Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	05_{16}	Put_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID address (optional)	32	00 01 8B $1A_{16}$	Transponder address is $00018B1A_{16}$
Parameters	8	03_{16}	Block Number 3
	Block size	07 00 AA AA_{16}	Block Data (32 bits).
CRC	16	$C4C7_{16}$	Calculated value of the CCITT CRC.

Required Parameters:

Block Number: Address on the Transponder

Block Data: Data to be stored in the Transponder block

Note:

Block Size is dependent on the transponder chip version.

3.3.6 Put_Block Response

Description: Response from the Transponder after the Put_Block transaction.

Example: 11_2 05_{16} 0_2 1_2 0_2 0_2 00 01 8B $1A_{16}$ $0CEA_{16}$

Table 16: Put_Block Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A Response from the Transponder to the Reader
Command Code	8	05_{16}	Put_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
CRC	16	$0CEA_{16}$	Calculated value of the CCITT CRC.

Returned Data: None unless an error has occurred

Specific Error Codes	
Definition	Code
Block not available	10_{16}
Block already locked	12_{16}
Block not successfully put	16_{16}

3.3.7 Put_Block_Lock Request

Description: Request for the transponder to store a block of data in a specified block and to lock that block.

This command must be followed by a Programming Burst as with all Put/Lock commands (see section 1.6)

Example: 00_2 07_{16} 0_2 1_2 0_2 00 01 8B $1A_{16}$ 03_{16} 07 00 AA AA_{16} 5051_{16}

Table 17: Put_Block_Lock Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	07_{16}	Put_Block_Lock Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
Parameters	8	03_{16}	Block Number is 3
	Block size	07 00 AA AA_{16}	Block Data (32 bits)
CRC	16	5051_{16}	Calculated value of the CCITT CRC.

Required Parameters:

Block Number: Address on the Transponder

Block Data: Data to be stored in the Transponder block

Note:

Block Size is dependent on the Transponder Chip Version.

3.3.8 Put_Block_Lock Response

Description: Response from the transponder after the Put and Lock combined transaction.

Example: 11_2 07_{16} 0_2 1_2 0_2 0_2 00 01 8B $1A_{16}$ $045E_{16}$

Table 18: Put_Block_Lock Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A Response from the Transponder to the Reader
Command Code	8	07_{16}	Put_Block_Lock Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
CRC	16	$045E_{16}$	Calculated value of the CCITT CRC.

Returned Data: None unless an error has occurred

Specific Error Codes	
Definition	Code
Block not available	10_{16}
Block already locked	12_{16}
Block not successfully put	16_{16}
Block not successfully locked	18_{16}

Note:

Error Code 16_{16} can occur if either the block put *or* lock fails.

3.3.9 Lock_Block Request

Description: Request for the transponder to lock a specific block.

This command must be followed by a Programming Burst as with all Put/Lock commands (see section 1.6)

Example: 00_2 08_{16} 0_2 1_2 0_2 00 01 8B $1A_{16}$ 02_{16} $BD6B_{16}$

Table 19: Lock_Block Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	08_{16}	Lock_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
Parameters	8	02_{16}	Block Number 2
CRC	16	$BD6B_{16}$	Calculated value of the CCITT CRC.

Required Parameters:

Block Number**:** Enter the block number to be locked in the Parameters field

3.3.10 Lock_Block Response

Description: Response from the transponder after executing the Lock Request.

Example: 11_2 08_{16} 0_2 1_2 0_2 0_2 00 01 8B $1A_{16}$ $3B08_{16}$

Table 20: Lock_Block Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A Response from the Transponder to the Reader
Command Code	8	08_{16}	Lock_Block Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
CRC	16	$3B08_{16}$	Calculated value of the CCITT CRC.

Returned Data: None unless an error has occurred

Specific Error Codes	
Definition	Code
Block not available	10_{16}
Block already locked	12_{16}
Block not successfully locked	18_{16}

3.3.11 SID_Poll Request

Description: Request for all Transponders in the Reader's Antenna field to send back their SID address and chip characteristics.

Example: 00_2 $0A_{16}$ 0_2 0_2 0_2 1_2 0_2 00 01 00_2 01 01_2 $0CDE_{16}$

Table 21: SID_Poll Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	$0A_{16}$	SID_Poll Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	0_2	Operating in non-Addressed Mode
Reserved	1	0_2	Must be set to zero.
Parameters	1	1_2	Info Flag – instructs the transponder to send version data in the response.
	1	0_2	Reserved – must be set to 0
	6	00 01 00_2	Mask Length – determines the bit length of the following parameter
	Mask Length	01 01_2	Mask – initial matching condition for SID address comparison by transponder
CRC	16	$0CDE_{16}$	Calculated value of the CCITT CRC.

Required Parameters:

Info Flag: If set, instructs the transponder to send its version data in the response.

Mask Length: Specifies the length of the Mask parameter in bits.

Mask: Initial matching condition for SID address comparison by transponder variable length.

Note:

This command is only used in Non-Addressed mode.

3.3.12 SID_Poll Response

Description: Response from the Transponder with SID and version data.

Example: 11_2 $0A_{16}$ 0_2 0_2 0_2 0_2 $00\ 01\ 8B\ 1A_{16}$ 0000001_2 000000101_2 000_2 00011_2 07_{16} $F857_{16}$

Table 22: SID_Poll Response

Field Name	# of bits	Value in Example	Description
Response Code	2	11_2	A Response from the Transponder to the Reader
Command Code	8	$0A_{16}$	SID_Poll Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	0_2	Operating in non-Addressed Mode
Reserved	1	0_2	Must be set to zero.
Error Flag	1	0_2	No error was encountered.
Data	32	$00\ 01\ 8B\ 1A_{16}$	Transponder SID Address is $00018B1A_{16}$
	7	0000001_2	Chip Manufacturer Code (1 = Texas Instruments)
	9	000000101_2	Chip Version 5
	3	000_2	Reserved
	5	00011_2	Block Size (in bytes) minus one (3 + 1 = 4 bytes = 32 bits block size)
	8	07_{16}	Number of Blocks minus one
CRC	16	$F857_{16}$	Calculated value of the CCITT CRC.

Returned Data:

Transponder SID address

Chip Manufacturer Code: (0 is excluded, allowing for 127 maximum)

Chip Version: Chip Version is specific to each manufacturer, so the combined fields of Chip Manufacturer + Chip Version is a unique combination.

Block Size minus one: Indicates the Transponder Chip’s block size in bytes.

Number of Blocks minus one: Indicates the number of data blocks on the Chip.

Specific Error Codes		
Definition	Code	Description
Command not allowed	$1F_{16}$	Returned if request was addressed
Command not supported	01_{16}	Returned if reserved parameter set to 1_2 , or if Mask Length > 28

3.3.13 Quiet Request

Description: Request that the transponder does not respond to any subsequent SID_Poll commands.

Example: 00_2 $0B_6$ 0_2 1_2 0_2 00 01 8B $1A_{16}$ $EE88_{16}$

Table 23: Quiet Request

Field Name	# of bits	Value in Example	Description
Request Code	2	00_2	A request from Reader to Transponder
Command Code	8	$0B_{16}$	Quiet Command Code
Format Type	1	0_2	For fixed format
Address Flag	1	1_2	Operating in Addressed Mode
Reserved	1	0_2	Must be set to zero.
SID Address (optional)	32	00 01 8B $1A_{16}$	Transponder SID Address is $00018B1A_{16}$
CRC	16	$EE88_{16}$	Calculated value of the CCITT CRC.

Required Parameters:

Transponder SID address

3.3.14 Quiet Response

Returned Data: None – there is no response from the transponder.

Note:

Upon receipt of the Quiet Command, the transponder enters the Quiet mode by setting an internal flag. The Quiet mode flag will be reset only by power-off of the transponder, i.e. the reader transmitter is turned off or the transponder moves out of the powered field. Whilst the Quiet mode flag is set, the transponder will continue to respond to any request other than SID_Poll.

Although the Quiet Command supports both the addressed and non-addressed modes, it is expected to be used only in addressed mode. Using it in non-addressed mode will turn off all transponders.

3.4 SID Mechanism

The SID_Poll Request is different to all other requests, in that it is completed by a managed sequence of power bursts, interrupted by pauses.

The SID polling mechanism has some similarity with the concept designed into the ALOHA network by the University of Hawaii in 1969, in the version referred to as "discrete random packet broadcasting".

It differs however in many respects, the main ones being:

Transponders (stations in the Aloha network) transmit their answer only when they are polled.

The polling mechanism is selective.

The time slots during which transponders can answer are specified by the reader, by the mean of interruption (SID pause) in the emission of the carrier frequency.

Within a given time slot, transponders answer only if there is a match between their unique factory-programmed SID address and the SID_Poll Request, according to well defined rules based on the mask parameter specified by the reader.

As the time slots are "clocked" by the reader, the total duration of the SID_Poll Command execution is optimized to the number of transponders in the field. The SID_Poll Request is followed by a sequence of long power bursts, pauses and short power bursts.

Power-up Sequence: As described earlier, before transmitting the SID_Poll Request, the reader performs a power-up pulse to supply all the tags within its interrogation field with enough initial power. The duration of this pulse is specified in the transponder IC datasheet.

3.4.1 SID_Poll Request:

Read Short Phase (t_{rs}): When no tag response is received by the reader (e.g. start of frame or 10 consecutive bits), the system stops transmitting and generates the next pause.

Read Long Phase (t_{rl}): If the system detects within the Read Short Phase a tag signal and a valid Start of Frame pattern (SOF), it extends the power transmission up to a maximum time (t_{rl}) or until the End of Frame (EOF) pattern has been received.

SID Pause (t_p): The end of each of the SID sequences (power phases) will be marked by a pause. This pause will be used by the tags as the trigger to enable the execution of an action or a command (e.g. command to increase the transponder SID_Address, or latch the received data and compare).

At the beginning of a read cycle the reader transmits the Power-Up sequence followed by a SID Command. In order to send commands the reader modulates the RF signal to transmit the encoded Downlink data. Between two SID-pause phases the reader transmits the power signal continuously to supply the tag with energy and a system clock. The receiver is active during this time and samples the data input to receive the incoming data from the tag. The length of these SID pulses depends on whether a tag is in the field or not. If during a time frame of (for example) t_{rs}^* the receiver does not receive any valid data, the reader ceases the transmission to generate the next pause to increase the transponder address counter. If during the t_{rs} time frame the receiver detects the Start of Frame pattern and/or a number of valid consecutive bits the power transmission will be prolonged up to a maximum time frame (t_{rl}^*) or in case of real-time processing until the End of Frame pattern is detected.

The SID mechanism consists of 4 elements:

the SID_Poll Request, including its parameters, sent by the reader

the Request processing performed by the transponder

the Collision Detection performed by the reader

the Collision Management Algorithm performed by the reader

The Reader talks first. Any transponder talks only when so instructed.

3.4.2 SID_Poll Request Parameters

The SID_Poll Request sent by the reader to the transponders contains 2 parameters:

- mask length
- mask value

The mask length and value are set by the reader according to the algorithm described below.

The mask is used to selectively address the subset(s) of transponders having previously generated a collision.

The Request is followed by a series of 16 pauses (short interruptions of carrier), the time interval between 2 subsequent pauses varies as explained above. The 16 pauses allow time to analyze a 4-bit field. This value is believed to offer the best performance for the number of transponders which can be reasonably present in the reader field of interrogation (between 1 and 100).

3.4.3 SID_Poll Request processing by transponder

On receipt of a Request, each VICC stores the mask value and the mask length into the mask register. It also clears its sub-address counter.

The mask register and the sub-address counter are fed into a comparator with the transponder address.

Note:

As explained in the algorithm description below, in the first request of an inventory cycle, the mask length is null, thus the sub-address counter is compared to the 4 LSB of the transponder address.

The transponder then counts the pauses. Upon the detection of a pause, the VICC stops any current transmission of an ATQ, it increments it sub-address counter, it compares the bits contained in the sub-address counter and in the previously stored mask to the corresponding bits of its address (the “don’t care” bits of the address are ignored),

if a match is found, it sends back an ATQ (containing its address), otherwise (if no match is found) it stays quiet.

3.5 Collision Detection

Immediately after having generated a Pause, the reader listens for possible reception of responses sent by transponders present.

Several situations may occur:

No signal can be detected: This means that no transponder matching the current criteria (mask & sub-address) is present in the field. The reader then generates the next Pause. As this detection can be done in a very short time (approximately 1ms), this speeds up the whole process.

A signal is detected: The reader extends the carrier powering and receives the Response(s).

A collision is detected: This means that more than one transponder match the current criteria (mask & sub-address) and have sent back their response. The reader then generates the next Pause, without waiting for the full transmission of the responses. This Pause will stop transmission of any transponder, as described above. This also speeds up the process.

No collision is detected: The reader extends the carrier powering until reception of a complete frame, or until a time-out elapses.

Two cases may occur:

An OK response is received (validated CRC).

A frame is received with errors (bad CRC, etc.). This is handled by the algorithm as if a collision was detected (see above).

3.5.1 Collision Management Algorithm

Using the three mechanisms described above, the Collision Management Algorithm is executed by the VCD and manages detected collisions, so that all VICCs are selectively interrogated such that at some stage only one VICC will be selectively addressed and thus will be the only one to answer.

The algorithm is described below in pseudo-code and uses recursivity for the sake of efficiency and clarity of programming. Other implementations are possible.

```
Function push (mask, address); pushes on private stack

Function pop (mask, address); pops from private stack

Function pulse_next_pause; generates a power pulse

Function store(VICC_id); stores inventoried VICC; address (identifier)

Function poll_loop (sub_address_size as integer)

Pop (mask, address)

Mask = address & mask; generates new mask

; send the request

        send_Request(Request_cmd, mask length, mask value) for sub_address = 0
        to (2^sub_address_size - 1)

        pulse_next_pause

        if no_collision_is_detected then

; VICC is inventoried

        store (VICC_id)

        else

; remember a collision was detected

         push(mask,address)

         endif next sub_address

; if some collisions have been detected and not yet processed,

; the function calls itself recursively to process the last

; stored collision

        if stack_not_empty then poll_loop (sub_address_size)

        end poll_loop
```

```
main_cycle:
mask = null
address = null
push (mask, address)
poll_loop(sub_address_size)
end_main_cycle
```

3.6 CRC

The cyclic redundancy check (CRC) is calculated bit-wise on all the preceding bits of the command.

CRC-CCITT is used, since it offers a good level of security.

Table 24: CRC Definition

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

Note that the one's complement of the calculated CRC is generated after CRC calculation, and sent with the command. If the received CRC is included in the CRC calculation while checking at reception, this one's complement of the original CRC generates the residue $1D0F_{16.}$

An example of the calculated CRC for the Get_Block Request (block 2) is illustrated here. First, the complete request is shown in binary format:

$000000000100000000010110001111001011 1_2$

The last 16 bits form the attached CRC. Calculation of the CRC bit-wise from the value

$00000000010000000001 0_2$

generates a value of 3868_{16}.

This is one's complemented to give the value $C797_{16}$ which is attached to the request data for transmission.

Memory Structure of Tag-it Inlays

Figure 15: Memory Structure of Tag-it Inlays

(RI-I01-110A, RI-I01-110A)

BLOCK DATA		LOCK BITS (0 = unlocked, 1 = irreversibly locked)		Settings/Comments
		FACTORY LOCKED	USER LOCKED	
SID ADDRESS 32 bit		1		Unique factory-programmed number
	R/O Memory 32bit (Version Info)			Mask programmed. Contains info on manufacturer code, chip/tag version and memory architecture. Accessed via Get_Version or SID_Poll Commands.
	BLOCK 0 32bit	0	0	Read/Write application data
	BLOCK 1 32bit	0	0	Read/Write application data
	BLOCK 2 32bit	0	0	Read/Write application data
	BLOCK 3 32bit	0	0	Read/Write application data
	BLOCK 4 32bit	0	0	Read/Write application data
	BLOCK 5 32bit	0	0	Read/Write application data
	BLOCK 6 32bit	0	0	Read/Write application data
	BLOCK 7 32bit	0	0	Read/Write application data

APPENDIX B

Terms and Abbreviations

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 accessed with a ‘Get’ or ‘Put’ using the standard commands. It may vary in length between different transponder versions.
CRC	Cyclic redundancy check
DSP	Digital signal processor
FCC	U.S. Federal Communications Commission
FDX	Full-Duplex (In the context of transponders)
Get	Defines the action of getting data from the transponder. This may be equated to the existing ‘Read’ Command for LF TIRIS transponders.
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
LS Byte	Least significant byte
LSB	Least significant bit.
Message	A Host Protocol message. Can be a response or a request.
MS Byte	Most significant byte
MSB	Most significant bit.
Put	Defines the action of putting data into the transponder. This may be equated (in its simplest form) to the existing ‘Write’ Command for LF TIRIS transponders.
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.