Texas Instruments

Reader Series 4000

S4100 Multi-Function Reader Module RF-MGR-MNMN

Base Application Protocol Reference Guide

11-01-21-700 October 2003

First Edition - October 2003

This is the first edition of this manual. It describes the **TI Series 4000 Reader.**

It contains a description of the following reader module:

S4100 Multi-Function Reader Module P/N: **RF-MGR-MNMN-N0**

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.

The **TIRIS** logo, the words RFID Systems, **TIRIS** and are trademarks or registered trademarks of Texas Instruments Incorporated (TI).

Read This First

About This Manual

This reference guide for the Series 4000 Multi-Function (13.56 MHz & 134.2 KHz) Reader is designed for use by TI customers who are engineers experienced with RFID Systems and Radio Frequency Identification Devices (RFID).

Device Name	Boot Loader Firmware Version
RF-MGR-MNMN-N0	1.02

The Regulatory, safety and warranty notices that must be followed are provided in Chapter 2.

Conventions

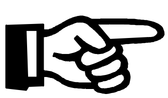
The following pictograms and designations are used in the operating instructions:

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.

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.tirfid.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

Topic **Page**

1.1 Introduction

The purpose of this chapter is to define the Application Layer Protocol and functions supported by the Multi-Function Reader (MFR). As a single application multiple frequency Radio Frequency IDentification reader, the MFR reader supports a single high-level Base Application that can be modified to be Host specific. The high level Application Layer module and various Entity Libraries models are compiled to create the Base Application. The Application Layer serves as the interface to these Entity Library modules and the functionality that they provide. This document uses protocol packet examples to demonstrate how to access the functions in the various libraries. The functionality that the MFR module supports is:

- 14443 layers 2,3,4 both type A & B
- 15693 layers 2 and 3
- Tag-it ™
- TI-RFID LF (DST, RO, RW)
- Serial I/O with TTL level I/F

Each packet has the same standard overhead bytes (Link Layer) to allow command validation and to insure the integrity of the packet data. This arrangement allows the packets to address the varying requirements of the data layer commands for each RF format.

Generic Packet Structure

The MFR reader is intended to operate as a slave to a system or device via a serial communications port. To support this relationship the MFR uses a standard protocol structure, the details of which are covered in the next section.

In general, this chapter details the protocol that gives the Host access to the functionality of the Base Application loaded into the reader. Examples will be provided of Request Packets to the MFR and response packets for the given requests. This document details how the Host can accesss the public functions that reside within the Base Application to control the LF and HF functionality.

How the Application Layer parses the Host protocols (packet structures) is detailed in the following chapters and sections. Each packet has the same standard overhead bytes to allow command validation and to insure the integrity of the packet data received. Additionally, this allows the packets to vary the length of the *<Data Layer>* to address the requirements of any given command, for any given RF format.

Programming Tools

The MFR's generic protocol packet structure and its implementation are independent of programming tools used, as any tools used to develop a Host application will depend on the individual project at hand. For compiling any developed application it is recommended that the easily available CodeVision AVR™ versions 1.23.7a or 1.23.8D compiler be used as the compiling tool.

1.2 MFR Protocol Commands

The figure below illustrates the generic packet structure for the request and response packets of the MFR. The Response Packet contains the *Response Status Byte* in the first Data field *<Data1>* while the data following the *Response Status Byte* depends on the Request Packet and how it is processed.

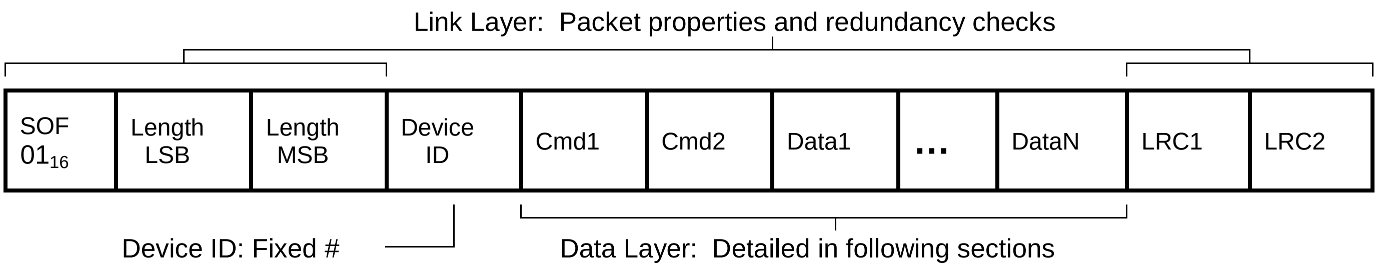


Figure 1-1: MFR Generic Packet Structure

The value of the *<Device ID>* field for the MFR terminal is always set to **03_{16}**. The Base Application will ignore any Request Packets that have the *<Device ID>* set to any value other than **03_{16}**.

The MFR terminal is loaded with a Base Application that can be compiled to support multiple Entity Module Libraries. The Base Application always contains an Application Layer. This Application Layer handles the protocol overhead for both the Request Packets and the Response Packets. Some of the Request Packets are directed to the Application Layer and others to a specific Entity Module. Each library module has a value assigned to it and these values are listed in the Entity Module Table. The *<Cmd 1>* field of the packet may be used to route the request to the proper library module. The Request Packet is always received and parsed by the Application Layer. The Application Layer is required to set up for and call the appropriate functions to achieve the results requested in the Request Packet. The Request Packet data is parsed and passed to the library module by calling the appropriate function from the selected library.

There are some commands that will be directed to the Application Layer. This does not mean that the Application cannot use the Entity Modules to complete the task. This document details the commands based on the Application Layer Commands (**01_{16} – 60_{16}**) and the Entity Module Commands (**61_{16}-FF_{16}**). The following sections detail the request commands currently supported by the Base Application for the MFR terminal.

The Application Layer of the MFR terminal parses the packets over the USART 0 port as detailed in the Receive Packets flow chart. The USART0 port generates an interrupt when data passes over the port. The Application Layer firstly checks for the *<SOF>*, then the *<Length Bytes>* and uses that length to determine how many data characters to wait for. The Application Layer will wait till the entire packet is received and validated before it attempts to see if the *<Cmd 2>* value is valid. The packet is declared Invalid if the inter-character timeout expires or the *BCC* digits received do not match the calculated values. The maximum packet length is 128 bytes. The command validation process begins after a successful Request Packet has been received over the USART 0 Port. This process is detailed in the Process Packet flow chart.

The *<Length LSB>* and *<Length MSB>* bytes are calculated on the complete string and include the SOF and LRC bytes. (See examples)

The Block Check Character(s) *<BCC Bytes>* validate the data within the packet. The first BCC byte is the 8-bit LRC calculated on the *<SOF>* characters and all of the packet bytes, up to the BCC field. The second BCC byte can be validated as follows. *<BCC 1>* + *<BCC 2>* = FF_{16}. This can also be considered *<BCC 1>* = *<LRC>* and *<BCC 2>* = *~<LRC>*.

1.3 Application Layer Command Overview (01-60$_{16}$)

The Application Layer parses all Request Packets over the USART Port and verifies that packets have a valid <*SOF*> character*, Length Bytes*, and valid *BCC Bytes*. The MFR reader will ignore any invalid packets.

As soon as the *BCC* digits have been validated, the MFR reader tests to see if the <*Cmd 2*> field is a valid Command. The Application Layer Command values can range from **01**$_{16}$ to **60**$_{16}$ but currently only a few of those values are in use. The MFR ignores Request Packets that fall within this range but are not currently supported. The currently supported values are detailed in this document. The Request Packets with a <*Cmd 2*> value from **61**$_{16}$ to **FF**$_{16}$ are addressed in the Entity Module Command section of this document.

The MFR terminal assumes that Request Packets are directed to the Application Layer and tests the <*Cmd 2*> value first to see if it is less than **61**$_{16}$and if there is a function associated with that particular value. The <*Cmd 1*> field is used to direct the Application Layer to the proper function within a particular Entity Library. The <*Cmd 1*> value may not be checked for most Application Layer functions and will only be validated in functions where it is applicable. The <*Cmd 1*> field is verified for all Request Packets with <*Cmd 2*> fields over **60**$_{16}$.

Functions like the Pass-Through, Find Token, and Transmitter ON/OFF requests are actually a hybrid between the Application Layer and the Entity Module commands. The <*Cmd 2*> value for these functions fall in the range of the Application Layer because they require every Entity Library to support this functionality. The <*Cmd 1*> value selects the Module to direct the request to. Every library must provide a function to support this functionality. The Get Version Request requires all libraries to support a function that will provide the current version, but the <*Cmd 1*> value will never be checked for this command. The Get Version command is always directed to the Application Layer and is not a true hybrid.

The <*Cmd 1*> value **01**$_{16}$ is reserved for the Application Layer commands. The <*Cmd 2*> field is always checked first to confirm it is a valid command, before the <*Cmd 1*> field is validated.

Information:

The <*Cmd 1*> field is only validated in functions that access multiple libraries. The Application Layer will ignore all Request Packets with invalid <*Cmd 2*> values. No Response Packet will be returned when there is an invalid Request Packet. A packet with a <*Device ID*> other than **03**$_{16}$ is considered invalid.

1.4 Base Application Command Structure

Although the Application Layer functions in a similar manner to previous TI RFID applications, the user should not confuse the MFR with other RFID devices. The MFR is a separate device from other RF devices and has separate sets of commands and tables. It is possible to have the functions associated with a particular <*Cmd 2*> value change as a MFR Base Application is modified to support a specific customers' needs. The following sub sections of this document details the current <*Cmd 2*> values for the Base Application Layer, which are listed in Figure 2-2.

Request Command	Function
20_{16}	**Enter Download Mode**
40_{16}	**Get Version**
41_{16}	**Find Token**
42_{16}	**Token Priority**
43_{16}	**Set Driver**
45_{16}	**Pass-Through**
46_{16}	**Set Baud Rate**
48_{16}	**Transmitter ON**
49_{16}	**Transmitter OFF**

Figure 1-2: Generic Base Application Command Definitions

Version Request (40_{16})

The Version Request can be used to retrieve the version numbers of the various terminal firmware components within any Base Application. This command may be helpful to identify the RF formats supported and the library versions in the Base Application of any MFR reader shipped to a customer.

The Application Layer queries each Entity Module compiled in the Base Application and places that information in the Response Packet that is returned to the Host application. There may be multiple Entity firmware versions within any Base Application. In some configurations a MFR terminal may contain a specific Entity Module that would not be included in other configurations. The Version Request Flow Chart will provide a visual representation of the function flow.

Information:

Additional libraries may be included as this product expands. The flow chart is just an example of the flow and is not intended to detail every Entity Module.

There will be several firmware versions returned in the data region of the response packet. Each library version is two bytes. The number of versions returned is determined by the number of libraries compiled in that particular Base Application and may vary between Base Applications. An Entity ID byte is returned before each two-byte version.

Information:

The Get Version Request is always directed to the Application Layer regardless of the contents of the *<Cmd 1>* field. The Application Layer returns the versions of each Entity Library but cannot be directed to get the version of a specific Entity Library.

The following is an example of a Get Version Request with multiple libraries compiled into the Base Application

Request Packet: (**01 08 00 03 01 40 4B B4**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	40	Version Request
BCC	4B B4	LRC and ~LRC

Response packet: (**01 24 00 03 01 40 00 01 03 01 11 04 01 02 01 01 03 01 01 04 02 01 05 02 01 06 03 01 07 01 01 08 02 01 78 87**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	24 00	Packet Length 36 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	40	Version Request
Status Byte	00	ERROR_NONE
Entity ID	01	Application Layer Firmware
Application Layer	03 01	Version 1.0.3
Entity ID	11	Boot loader Firmware
Boot loader	04 01	Version 1.0.4
Entity ID	02	14443-A Firmware
14443-A	01 01	Version 1.0.1
Entity ID	03	14443-B Firmware
14443-B	01 01	Version 1.0.1
Entity ID	04	15693 Firmware
15693	02 01	Version 1.0.2
Entity ID	05	Tag-it TM Firmware
Tag-it	02 01	Version 1.0.2
Entity ID	06	LF DST Firmware
LF DST	01 01	Version 1.0.3
Entity ID	07	14443 4 Firmware
14443 4	01 01	Version 1.0.1
Entity ID	08	Apollo Firmware
Apollo	02 01	Version 1.0.2
BCC	78 87	LRC and ~LRC

Find Token Request (41_{16})

The Host application can send the MFR terminal a Request Packet to check if a token is present. This packet contains a loop count parameter that sets the maximum number of times the MFR terminal will search for each RF format in the Priority Table. This parameter allows each Host to have control over how many times the terminal will sample for each RF format before determining that no token is present. The function exits and returns a response when it gets the first valid read.

This function gives the Host a great deal of flexibility. It is possible to search for a variety of transponders or a single type of token. If multiple RF formats have been selected, the function returns the first valid token that is read. The terminal will only complete the maximum number of loops specified if it does not detect a valid token. The terminal does not return every type of token that may be in the field, just the first one it reads.

Information:

The Priority Table selects the order in which the terminal looks for different RF tokens when the <*Cmd 1*> field is set to the Application Layer **01**$_{16}$. The reader ignores the Priority Table when the <*Cmd 1*> field points directly to a specific RF Entity ID. 14443-4 and Apollo are invalid RF Entity IDs for this function - these libraries help support the 14443 RF formats, but are invalid <*Cmd 1*> values.

After a successful read the MFR Terminal responds with **ERROR_NONE** in the Response Status Byte field followed by token's RF Technology Type and the requested token data. The RF polls stop once a valid token is found or the maximum number of loops have occurred. If a valid token is not found within the loop count selected, the MFR terminal responds with an **ERROR_TOKEN_NOT_PRESENT** for the Response Status byte.

Current RF Token types and their default order are defined in the Default RF Priority Table. This table also details the discretionary data that is returned in the Response Packet following the Token Type.

The <*Cmd 1*> field determines if the Token Priority Table is to be used or whether the terminal is to just search for a specific RF format. The terminal only uses the RF Priority Table when the <*Cmd 1*> field is pointing to the Application Layer. The MFR terminal defaults to searching for transponders in the default order when the RF Priority Table is empty. It is possible to set the Priority table, and bypass it when desired by setting the <*Cmd 1*> field to a value other than the Application Layer. The RF formats and the Set Token Priority Request Packet can modify the order in which they are checked. The MFR terminal can select from 1 to 5 RF formats to check during each loop through the Priority Table.

The Find Token Request changes from an Application Layer Command to an Entity Module Command when the <*Cmd 1*> field is set to a specific library value, rather than the Application Layer. The following examples demonstrate the various conditions. The Find Token Request Flow Chart provides a visual representation of code flow. The Loop Count Timings Table gives some sample timings for various loop counts.

Find Token Example: Loop 10 times through Default Priority Table, Present LF DST token:
Request Packet: (**01 09 00 03 01 41 0A 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	41	Find Token Request
Timeout	0A	10 loops maximum
BCC	41 BE	LRC and ~LRC

Response packet: (**01 0E 00 03 01 41 00 06 06 FA 04 00 B2 4D**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0E 00	Packet Length 14 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	41	Find Token Request
Status Byte	00	ERROR_NONE
Entity ID	06	LF DST Library
MID	06	Merchant ID
Serial #	FA 04 00	Tag # 1274
BCC	B2 4D	LRC and ~LRC

Example: Loop 10 times through Default Priority Table, Present 15693 token:
Request Packet: (**01 09 00 03 01 41 0A 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	41	Find Token Request
Timeout	0A	10 loops maximum
BCC	41 BE	LRC and ~LRC

Response Packet: (**01 14 00 03 01 41 00 04 00 00 FE B3 81 06 00 00 07 E0 7F 80**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	14 00	Packet Length 20 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	41	Find Token Request
Status Byte	00	ERROR_NONE
Entity ID	04	15693 Library
Inventory Flag	00	Inventory Flag
DSFID	00	Data Storage Format ID
UID	FE B3 81 06 00 00 07 E0	Unique ID LSB first
BCC	7F 80	LRC and ~LRC

Example: Loop 10 times through 14443-B Priority Table, Present 14443-B token:
Request Packet: (**01 09 00 03 03 41 0A 43 BC**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	41	Find Token Request
Timeout	0A	10 loops maximum
BCC	43 BC	LRC and ~LRC

Response Packet: (**01 0F 00 03 03 41 00 03 01 DF 35 45 83 61 9E)**

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0F 00	Packet Length 15 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	41	Find Token Request
Status Byte	00	ERROR_NONE
Entity ID	03	14443-B Library
Inventory Flag	01	CID
DSFID	DF 35 45 83	PUPI
BCC	61 9E	LRC and ~LRC

Set Token Priority (42_{16})

MFR terminals can support a variety of RF standards. The Find Token Request utilizes the RF token priority list when the <*Cmd 1*> field is set to the Application Layer **01**$_{16}$. This Priority Table sets the order in which the Find Token command will search for and validate tokens. The Find Token command will stop as soon as it has detected the first valid token in the field. The order of priority in which the transponders are checked could determine the token type returned, in the event that there are multiple types of transponders present. The Find Token Request command does not return all types of transponders present, just the first one it validates.

It is possible that there will be applications where the MFR terminal should only support certain types of transponders. This Token Priority Table can select which RF formats to support, as well as the order to check for them. The contents and order of this table may change from Base Application to Base Application. It is also possible to reconfigure the Priority Table within any given Base Application.

Information:

The Entity Library for any given RF format must be compiled into the Base Application before it can be supported in the RF Priority Table.

Current RF Token format IDs and their default order are defined in the Default RF Priority Table. This table also details the discretionary data that will be returned in the Response Packet for the Find Token Request.

The MFR terminal defaults as shown in the RF Priority table. The Set Token Priority Request command can be used to change the order and or eliminate some RF formats.

Each byte of the data field of the Request Packet is verified to make certain it is a valid RF Format. The RF format value must range between the minimum RF value and the maximum RF value as assigned in the table. The MFR terminal will reset the Priority table to default if any of the priority values are invalid. The Status Byte will be set to ERROR_INVALID_RF_FORMAT when an invalid RF format has been sent in the Request Packet.

Information:

The 14443-4 Module and TI Apollo Module Libraries are not listed as valid RF formats. Those libraries are used to support the 14443-A and 14443-B RF formats but are not stand-alone RF formats that can be used in the Priority Table. The Find Token command will never return a Response Packet that indicates it has detected a token of type 14443-4 or TI Apollo token.

It is possible to select from 1 to 5 RF formats to be supported within the MFR module. The transponders will be ignored when they are presented to a terminal that does not have that RF Format set in the RF Priority Table.

The user cannot specify the same RF technology twice in the same priority table. The terminal will return ERROR_INVALID_RF_FORMAT as the status byte and reset Priority Table when the priority table is invalid. Currently there are a maximum of 5 fields for the Priority Table. This does not mean all five fields must be used. If the user wishes to force the terminal to return to the default RF Priority Table, this can be achieved by sending a Set RF Priority request with no RF values set in the data field.

Set Token Priority Example: Force the RF Priority to the default order. There is NO RF format in the data field. Request Packet: (**01 08 00 03 01 42 49 B6**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	42	Set RF Priority Request
BCC	49 B6	LRC and ~LRC

Response packet: (**01 09 00 03 01 42 00 48 B7**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	42	Set RF Priority Request
Status Byte	00	ERROR_NONE
BCC	48 B7	LRC and ~LRC

Example: Set the RF Priority Table order. This will firstly check for 14443-B transponders then LF transponders and then finally check for 15693 transponders.

Request Packet: (**01 0B 00 03 01 42 03 06 05 4A B5**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0B 00	Packet Length 11bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	42	Set RF Priority Request
Priority #1	03	14443-B token
Priority #2	06	LF DST/RO/RW token
Priority #3	05	15693 token
BCC	4A B5	LRC and ~LRC

Response Packet: (**01 09 00 03 01 42 00 48 B7**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	42	Set RF Priority Request
Status Byte	00	ERROR_NONE
BCC	48 B7	LRC and ~LRC

Set Driver Request (43_{16})

The MFR terminal does not have any LED's or buzzer attached, but may be attached to a board with LED's and/or a buzzer fitted. The MFR terminal must support the drivers for these devices to allow the Host control even though they are not physically mounted on the MFR itself.

Each device has a certain bit reserved in the control byte. The Driver Allocation Table details the driver and bit association. When the bit is '1' the device is selected and when the bit is '0' the device is NOT selected. The first byte of the Request Packet will select the device/devices to control.

The Request Packet also has a second parameter to control the action of the device/devices selected by the first parameter. There are currently three action options that are supported. The action options are:

01$_{16}$ - Turn Device/Devices **ON**
02$_{16}$ - Turn Device/Devices **OFF**
03$_{16}$ - Toggle Device/Devices **ON/OFF**

The Toggle option provides a way to change the device state from the current state. This allows the Host application to loop and call this function to bring attention to the device selected. If an LED is off, it will turn it ON and if it is ON, it will turn it OFF. This command is an Application Layer Command and does not validate the *<Cmd 1>* value.

Set Driver Example: Turn ON the two LED's of the application or the attached motherboard.
Request Packet: (**01 0A 00 03 01 43 06 01 4D B2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	0A 00	Packet Length 10 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	43	Set Driver Request
Driver/Drivers	06	Select Drivers 2 and 3
Action	01	Turn the Drivers ON
BCC	4D B2	LRC and ~LRC

Response packet: (**01 09 00 03 01 43 00 49 B6**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	43	Set Driver Request
Status Byte	00	ERROR_NONE
BCC	49 B6	LRC and ~LRC

Enter Download Mode (20_{16})

The MFR terminal will normally be executing the Base Application firmware that resides in the program area of the FLASH but may be required to replace this Base Application, without using a JTAG device. The MFR Base application can be replaced though, through the Boot loader which resides in a separate region of FLASH memory. The terminal will execute this Boot loader firmware and enter the Download Mode, when it has received an Enter Download Request Packet.

The terminal enters the Download Mode when it jumps to and executes the Boot loader Application. A timeout period begins with execution of the Boot loader Application, with the terminal waiting for a valid Download Request Packet, for up to a maximum of 15 seconds. The terminal will return to the Target Application if no valid Download Request has been received prior to the expiration of the timeout period. The receipt of any Download Mode Request Packet resets the timeout period.

When executing the Download Application, the terminal will accept any valid Download Request, but the normal MFR Request Packets will be ignored.

The Download Mode Status Byte is set to **01**$_{16}$ when it enters the Download Mode, and is reset to **00**$_{16}$ when it returns to the Application Mode. The Download Status Byte is stored in EEPROM.

Information:

- The Application Layer does not send the Response Packet - the Boot loader Application returns the Response Packet.
- The Boot loader is not part of the Base Application. They are separate applications with separate Device ID values and different protocols.
- The Boot loader is identified with <*Entity ID*> **11**$_{16}$ but it should not be confused with a library within the Base Application. This is just a way to identify the Boot loader Application.

Enter Download Mode Example: Force the reader to enter the Download Mode.

Request Packet: (**01 08 00 03 01 20 2B D4**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	20	Enter Download Mode Request
BCC	2B D4	LRC and ~LRC

Response packet: (**01 09 00 02 FF 20 00 D5 2A**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Bootloader
Command 1	FF	N/A
Command 2	20	Enter Download Mode
Status Byte	00	ERROR_NONE
BCC	D5 2A	LRC and ~LRC

Pass-Through Request (45_{16})

The Pass-Though Request Packet provides a way for a Host to have direct communications with an Entity Module library. The Pass-Through Request Packet uses the <*Cmd 1*> field to specify which Entity to direct the data to and then uses the <*Data Layer*> to provide data that the library can parse and use. This function provides a way to isolate and test the library functions without having to make any modifications to the Application Layer.

This is an Application layer command, which directs the request to the proper Entity Library by validating the <*Cmd 1*> field. Each library may be required to perform unique functionality to complete this request. This is considered an Application Layer Command because each library will be required to have a Pass-Through function that can be called when the <*Cmd 2*> value is set to $\mathbf{45}_{16}$ and the <*Cmd 1*> field selects that specific library.

Information:

The Pass-Through Request is an Application Layer function that does not allow the <*Cmd 1*> field to be directed to the Application Layer $\mathbf{01}_{16}$. This function is used to allow the Host control of the RF formats and does not apply to the Application Layer.

The data is stripped out of the <*Data Layer*> of the Request Packet and passed directly to the selected Entity Library. The Library channels the data to the RF interface according to its specific RF protocol and returns the appropriate response information back to the Application Layer. The Application Layer builds a Response Packet based on the information returned by the Entity Library and sends it to the Host.

The Pass-Through command is designed to allow the Host some control of the RF format without having to add additional protocol overhead to the Application Layer and compiling a new Base Application. The <*Data Layer*> of the Request Packet can change the operation of the selected RF format without having to change out the Base Application. That data that is passed and the functionality it controls is very specific to each Entity Library. It is not the intention of this document to outline the full capability of the Pass-Through commands for each and every library. Refer to the documentation for the selected RF format for in depth information on how each library handles the Pass-Through Request.

14443-A Pass-Through Request

This function provides a way to isolate and test the 14443-A library functions without having to make any modifications to the Application Layer. The ISO 14443A library module allows direct access to the RF interface via the pass-through request. When the ISO 14443A module receives a pass-through request, the module configures the HF ASIC to an ISO 14443A – 2 specific RF scheme.

The Pass-Through packet's data contents are:

- Packet type byte (1 byte),
- Payload length byte (1 byte) and
- Payload data bytes (depends on Payload length and Packet type).

Packet Type: Three types of packets are described in ISO14443A specification.
0 for ISO14443A-3 short frame,
1 for ISO14443A-3 bit oriented anti-collision frame and
2 for ISO14443A-3 standard frame.

Payload Length: The interpretation of the contents of this field depends on the packet type. For packet types 0 and 1 it specifies the number of bits to be sent. For packet type 2 it specifies the number of bytes to be sent.

Payload data bytes: The number of bytes in this field should be sufficient to hold the #bits/#bytes specified in the Payload Length field.

The payload data in the pass-through request is not interpreted and it has meaning only to the token that may receive it. Therefore the process that initiates the request must know the structure and data content of the transponder protocol. Some of these transponder protocol details are specified in ISO 14443A layer 3 and layer 4 specification documents. Application and security layer commands may also be sent to the token via the pass-through request as long as the calling process understands and follows the states of the token being addressed.

The response data field contains (not including the Response Status Byte):

- # bytes field: Indicates how many complete bytes of data were received.
- # bits field: Indicates the number of extra bits received (see following example).
- Raw data as received from the token except for the parity bits which are first verified and then removed from the received data before being stored into the receive buffer.

The <*Cmd 1*> field is set to **02**$_{16}$ for the ISO 14443A library. The payload data portion of the Request Packet is passed to the function *Pass_Through_14443A()*. This gives the Host full control over the HF ASIC and what it sends. The data must strictly follow the ISO14443A layer 3 and 4 protocol.

Information:

The library only supports a PCD to PICC and PICC to PCD at a data rate of 106 kbps. Interleaving Pass-through requests in the MFR reader is not generally recommended. Sending a non-pass-through request may set internal data structures that influence how subsequent non-pass-through request parameters are handled.

14443-B Pass-Through Request

This function provides a way to isolate and test the 14443-B library functions without having to make any modifications to the Application Layer.

ISO14443B library module/entity allows direct access to the RF interface via the pass-through request. When the ISO14443B module receives a pass-through request, the module configures the HF ASIC to an ISO14443B – 2 specific RF scheme. The data in the pass-through request is not interpreted and it has meaning only to the token that may receive it. Therefore the process that initiated the request must know the structure and data content of the transponder protocol. Some of the transponder protocol details are specified in ISO14443B layer 3 and layer 4 specification documents. Application and security layer commands may also be sent to the token via the pass-through request as long as the calling process understands and follows the states of the token at hand.

S (WTXM) packet is an S-block according to the ISO14443-4 document. The token sends an S (WTXM) packet in response to an I-block packet if the token needs more time than is allowed by the Wait Time Index. The reader has to acknowledge with an S (WTXM) packet downlink, so that the token can proceed to perform the task requested. The MFR reader ISO14443B library can handle the S (WTXM) request from the token automatically if the pass-through packet requests it.

The <*Cmd 1*> field is set to **03**$_{16}$ for the ISO14443B library. The first data byte of the data field specifies to the reader whether the reader should handle any S (WTXM) Request Packets from the token. If the value set in this byte is **00**$_{16}$this feature is turned off and if **01**$_{16}$ the feature is on. This is useful for layer 4 packets or security and application layer packets, which use ISO14443, layer 4. The data portion of the Request Packet is passed to the function Pass_Through_14443B(). This gives the Host full control over the HF ASIC and what it sends. The data must strictly follow the ISO14443B layer 3 and 4 protocol.

Information:

The library only supports a PCD to PICC data rate of 106 kbps. The Host must make sure that the data rate specified in the ATTRIB pass-through command is set to 106 kbps for the PCD to PICC transmissions. Interleaving Pass-through requests and non-pass-through requests is not generally recommended. Sending a non-pass-through request sets internal structures that influence the parameters for subsequent non-pass-through requests.

15693 Pass-Through Request

The ISO 15693 library module allows direct access to the RF interface via the pass-through request. When the ISO 15693 module receives a pass-through request, the module configures the HF ASIC to the ISO 15693 specific RF scheme, according to the "Set Parameters" request (Refer to the ISO 15693 Protocol Reference Guide for details). The data in the pass-through request is not interpreted and it has meaning only to the token(s) that may receive it. Therefore the process that initiated the request must know the structure and data content of the transponder protocol. These transponder protocol details are specified in the ISO 15693 standards. Application layer commands may also be sent to the token via the pass-through request as long as the calling process understands and follows the states of the token at hand.

The <*Cmd 1*> field is set to $\mathbf{04}_{16}$ for the ISO 15693 library. The data portion of the Request Packet is passed to the function Pass_Through_15693(). This gives the Host full control over the HF ASIC and what it sends. The data must follow the ISO 15693 standards.

The details for the ISO 15693 library's pass-through functionality are described in detail in the ISO 15693 Protocol Reference Guide.

Tag-it™ Pass-Through Request

The Tag-it™ library module allows direct access to the RF interface via the pass-through request. When the Tag-it™ module receives a pass-through request, the module configures the HF ASIC to the Tag-it™ specific RF scheme. The data in the pass-through request is not interpreted and it has meaning only to the token(s) that may receive it. Therefore the process that initiated the request must know the structure and data content of the Tag-it™ transponder protocol. The document describing the Tag-it™ transponder protocol is under the reference section on the RFID home page (http://www.ti.com/tiris).

The <*Cmd 1*> field is set to $\mathbf{05}_{16}$ for the Tag-it™ library. The data portion of the Request Packet is passed to the function Pass_Through_TagIt(). This gives the Host full control over the HF ASIC and what it sends.

The details for the Tag-it™ library's pass-through functionality are described in detail in Tag-it™ Reference Guide.

LF Pass-Through Request

The LF DST library module allows direct access to the LF interface via the pass-through request. When the LF DST module receives a pass-through request, the module configures the LF ASIC to the specific RF scheme. The data in the pass-through request is not interpreted and it has meaning only to the token(s) that may receive it. Therefore the process that initiated the request must know the structure and data content of the LF transponder protocol.

The <*Cmd 1*> field is set to $\mathbf{06}_{16}$ for the Tag-it™ library.. The <*Data Layer*> portion of the Request Packet is parsed and passed to the LF DST Entity Library to allow the Host full control over the LF ASIC.

The <*Data Layer*> portion must be passed in the following format:

[PB1][PB2][T_OFF_H][T_ON_H][T_OFF_L][T_ON_L][DATA]

The first two bytes *(PB1, PB2*) represent Power Burst durations in milliseconds.
The following 4 parameters (*T-OFF_H, T_ON_H, T_OFF_L, T_ON_L*) are integer values for the ON and OFF times in milliseconds.

The remaining bytes in the <*Data Layer*> are the modulated data that is sent between power bursts. The Write Address is always the first byte of modulated data. The following diagram shows how these parameters work with a Challenge DST Request.

The details for the LF DST library's pass-through functionality are described in detail in the LF DST users guide document.

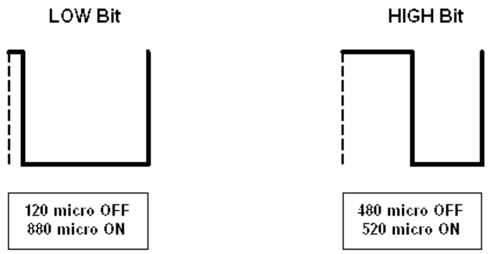


Figure 1-3: Example bit durations

Information:

The LF Pass-Through command can be used to communicate with Digital Signature Transponder (DST), Read Only (RO), and Read Write (RW) transponders. The Response Packet contents will indicate the type of token that responds.

Set USART 0 Baud Rate (46_{16})

The MFR reader can use the USART 0 Port command to modify the baud rate, with which the reader communicates with the Host. Some of the applicable protocol packets are defined within this document. Please note that the MFR reader will default to communication at the 9600-baud 8N1 level.

The <*Cmd 1*> field does not apply when the <*Cmd 2*> field is set to **46_{16}** because this is always directed to the Application Layer. The first byte of the <*Data Layer*> determines the new baud rate for the protocol. The supported baud rate values are as follows:

0 – 9600 baud
1 – 19200 bps
2 – 57600 bps
3 – 115200 bps
4 – 38400 bps

When executed this function returns a Response Packet at the current baud rate and then converts to the requested baud rate as soon as the last byte of the Response Packet has been transmitted out the USART 0 port. The Response Byte is set to ***ERROR_NONE*** when a valid baud rate is specified and ***ERROR_INVALID_BAUD*** when an invalid value is used.

Information:

The reader will default to 9600 baud after every power cycle or reboot. The program that the Host is using must be set to the same baud rate as the MFR reader when the Request Packet is transmitted, then change to the new baud rate after receiving the Response packet.

Set Protocol for 9600 Baud

To set the protocol to 9600 Baud:

Request Packet: (**01 09 00 03 01 46 00 4C B3**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud Request
Baud Rate	00	Set USART 0 to 9600 Baud
BCC	4C B3	LRC and ~LRC

Response Packet: (**01 09 00 03 01 46 00 4C B3**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud
Status Byte	00	ERROR_NONE
BCC	4C B3	LRC and ~LRC

Set Protocol for 19200 Baud

Request Packet: (**01 09 00 03 01 46 01 4D B2**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud Request
Baud Rate	01	Set USART 0 to 19200 Baud
BCC	4D B2	LRC and ~LRC

Response Packet: (**01 09 00 03 01 46 00 4C B3**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud
Status Byte	00	ERROR_NONE
BCC	4C B3	LRC and ~LRC

Set Protocol for 57600 Baud

Request Packet: (**01 09 00 03 01 46 02 4E B1**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud Request
Baud Rate	02	Set USART 0 to 57600 Baud
BCC	4E B1	LRC and ~LRC

Response Packet: (**01 09 00 03 01 46 00 4C B3**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	01	Application Layer
Command 2	46	Set USART 0 Baud
Status Byte	00	ERROR_NONE
BCC	4C B3	LRC and ~LRC

Transmitter ON Request (48_{16})

The Transmitter ON Request is used to turn the transmitter ON as required for any specific RF Format. For this command the Request Packet uses the *<Cmd 1>* field to select the intended Entity Library. The application layer firmware redirects the request to the intended Entity Library. The Entity Library then sets up the appropriate hardware and software configuration needed to implement its specific details and then turns the transmitter ON.

This function has a *<Cmd 2>* value indicating that it is an Application Layer Command. The *<Cmd 1>* field is validated and used to direct the request to the correct Entity Library. Each library may be required to perform unique functionality to complete this request. This is considered an Application Layer Command because each library will be required to have a Transmitter ON function that can be called when the *<Cmd 2>* value is set to **48**$_{16}$ and the *<Cmd 1>* field selects that specific library.

14443-A Transmitter ON Request

The ISO14443A Transmitter ON request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up the correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the ISO14443A protocol related register settings in the HF ASIC.
The *<Cmd 1>* field is set to **02**$_{16}$ for 14443-A Transmitter ON.

Request Packet: (**01 08 00 03 02 48 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	48	Transmitter On Request
BCC	40 BF	LRC and ~LRC

Response Packet: (**01 09 00 03 02 48 00 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	48	Transmitter On Request
Status Byte	00	ERROR_NONE
BCC	41 BE	LRC and ~LRC

14443-B Transmitter ON Request

The ISO14443B Transmitter ON request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the ISO14443-B protocol register settings in the HF ASIC. The <*Cmd 1*> field will be set to **03**$_{16}$ for 14443-B Transmitter ON. Request Packet: (**01 08 00 03 03 48 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	48	Transmitter On Request
BCC	41 BE	LRC and ~LRC

Response Packet: (**01 09 00 03 03 48 00 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	48	Transmitter On Request
Status Byte	00	ERROR_NONE
BCC	40 BF	LRC and ~LRC

15693 Transmitter ON Request

The ISO15693 Transmitter ON request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the ISO15693 protocol related register settings in the HF ASIC. The <*Cmd 1*> field is set to **04**$_{16}$ for 15693 Transmitter ON.

Request Packet: (**01 08 00 03 04 48 46 B9**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	04	Library Layer – ISO15693
Command 2	48	Transmitter On Request
BCC	46 B9	LRC and ~LRC

Response Packet: (**01 09 00 03 04 48 00 47 B8**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	04	Library Layer – ISO15693
Command 2	48	Transmitter On Request
Status Byte	00	ERROR_NONE
BCC	47 B8	LRC and ~LRC

Tag-it ™ Transmitter ON Request

The Tag-it™ Transmitter ON request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the HF ASIC to set up the Tag-it™ protocol register settings in the HF ASIC. The <*Cmd 1*> field is set to $\mathbf{05}_{16}$ for Tag-it™ Transmitter ON.

Request Packet: (**01 08 00 03 05 48 47 B8**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	05	Library Layer – Tag-it™
Command 2	48	Transmitter On Request
BCC	47 B8	LRC and ~LRC

Response Packet: (**01 09 00 03 05 48 00 46 B9**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	05	Library Layer – Tag-it™
Command 2	48	Transmitter On Request
Status Byte	00	ERROR_NONE
BCC	46 B9	LRC and ~LRC

LF Transmitter ON Request

The LF Transmitter ON request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the LF ASIC to set up the LF protocol related register settings in the LF ASIC. The <*Cmd 1*> field is set to $\mathbf{06}_{16}$ for LF Transmitter ON.

Request Packet: (**01 08 00 03 06 48 44 BB**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	06	Library Layer – LF
Command 2	48	Transmitter On Request
BCC	44 BB	LRC and ~LRC

Response Packet: (**01 09 00 03 06 48 00 45 BA**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	06	Library Layer - LF
Command 2	48	Transmitter On Request
Status Byte	00	ERROR_NONE
BCC	45 BA	LRC and ~LRC

Transmitter OFF Request (49_{16})

Transmitter OFF Request is used to turn the transmitter OFF as required for any specific RF Format.

The Request Packet uses the <*Cmd 1*> field to select the intended Entity Library. The application layer firmware redirects the request to the intended Entity Library. The Entity Library then turns the transmitter OFF

This function has a <*Cmd 2*> value that indicates it is an Application Layer Command. The <*Cmd 1*> field is validated and used to direct the request to the correct Entity Library. Each library may be required to perform unique functionality to complete this request. This is considered an Application Layer Command because each library will be required to have a Transmitter OFF function that can be called when the <*Cmd 2*> value is set to **49**$_{16}$ and the <*Cmd 1*> field selects that specific library.

14443-A Transmitter OFF Request

The ISO14443A entity turns the transmitter off, sets up internal status flags and returns a response. The Request Packet consists of no data bytes. The <*Cmd 1*> field indicates the Entity Library intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet: (**01 08 00 03 02 49 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	49	Transmitter Off Request
BCC	41 BE	LRC and ~LRC

Response Packet: (**01 09 00 03 02 49 00 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	02	Library Layer – ISO14443A
Command 2	49	Transmitter Off Request
Status Byte	00	ERROR_NONE
BCC	40 BF	LRC and ~LRC

14443-B Transmitter OFF Request

The ISO14443B entity turns the transmitter OFF, sets up internal status flags and returns a response. The Request Packet consists of no data bytes. The *<Cmd 1>* field indicates the Entity Library intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet: (**01 08 00 03 03 49 40 BF**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	49	Transmitter Off Request
BCC	40 BF	LRC and ~LRC

Response Packet: (**01 09 00 03 03 49 00 41 BE**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	03	Library Layer – ISO14443B
Command 2	49	Transmitter Off Request
Status Byte	00	ERROR_NONE
BCC	41 BE	LRC and ~LRC

15693 Transmitter OFF Request

The ISO 15693 entity turns the transmitter OFF, sets up some internal status flags and returns a response. The Request Packet consists of no data bytes. The *<Cmd 1>* field indicates the Entity Library intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet: (**01 08 00 03 04 49 47 B8**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	04	Library Layer – ISO15693
Command 2	49	Transmitter Off Request
BCC	47 B8	LRC and ~LRC

Response Packet: (**01 09 00 03 04 49 00 46 B9**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	04	Library Layer – ISO15693
Command 2	49	Transmitter Off Request
Status Byte	00	ERROR_NONE
BCC	46 B9	LRC and ~LRC

Tag-it™ Transmitter OFF Request

The Tag-it™ entity turns the transmitter OFF, sets up internal status flags and returns a response. The Request Packet consists of no data bytes. The *<Cmd 1>* field indicates the Entity Library intended to turn off the transmitter. The response status byte indicates the success/error status.

Request Packet: (**01 08 00 03 05 49 46 B9**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	05	Library Layer – Tag-it™
Command 2	49	Transmitter Off Request
BCC	46 B9	LRC and ~LRC

Response Packet: (**01 09 00 03 05 49 00 47 B8**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	05	Library Layer – Tag-it™
Command 2	49	Transmitter Off Request
Status Byte	00	ERROR_NONE
BCC	47 B8	LRC and ~LRC

LF Transmitter OFF Request

The LF Transmitter OFF request firstly configures the hardware ports to their proper data directions (input/output), sets up pull-up resistors for input ports and sets up correct logic levels at the output ports. The transmitter itself is turned on and then the entity communicates with the LF ASIC to set up the LF protocol related register settings in the LF ASIC. The *<Cmd 1>* field is set to $\mathbf{06}_{16}$ for LF Transmitter OFF.

Request Packet: (**01 08 00 03 06 49 45 BA**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	08 00	Packet Length 8 bytes
Device ID	03	Terminal is MFR
Command 1	06	Library Layer – LF
Command 2	49	Transmitter OFF Request
BCC	45 BA	LRC and ~LRC

Response Packet: (**01 09 00 03 06 49 00 44 BB**)

Field	Contents	Summary
SOF	01	Start of Frame
Packet Length	09 00	Packet Length 9 bytes
Device ID	03	Terminal is MFR
Command 1	06	Library Layer - LF
Command 2	49	Transmitter OFF Request
Status Byte	00	ERROR_NONE
BCC	44 BB	LRC and ~LRC

1.5 Entity Module Commands ($61\text{-}FF_{16}$)

The MFR terminal gives the Host the ability to pass data directly to the Entity Modules. The Application Layer simply parses the Request Packet and passes the information directly to the library selected in the *<Cmd 1>* field. These are referred to as Entity Module Commands within this document.

The Application Layer verifies that the Entity Library selected in the <Cmd 1> field supports the functionality specified in the *<Cmd 2>* field. The Application Layer ignores invalid <Cmd 2> values and never passes the data to the Entity Library.

The information in the Data fields is RF Format specific. The Application Layer does not verify the data - just that it is the right length and correct BCC. The selected Entity Module is responsible for validation of the data in the *<Data Layer>* of the Request Packet.

The packets with a *<Cmd 2>* field less than $\mathbf{60}_{16}$ are classified as Application Layer commands even though some are hybrids that pass the data directly to the Entity Libraries. Each Entity Library has the potential to have functionality for *<Cmd 2>* values $\mathbf{61}_{16}$ to $\mathbf{FF}_{16}$. This does not mean that each Entity Library must have a command for each of these values, but each library has the potential to support up to 158 public functions without any modification to the Application Layer protocol design.

.
The Request Packet *<Cmd 1>* field selects the Entity Library that is addressed and the Application Layer validates if that particular library supports the *<Cmd 2>* value. This requires a small modification to the Application Layer each time any Entity Library adds or removes any *<Cmd 2>* value from those currently supported. The Application Layer doesn't validate the *<Data Layer>* for the selected *<Cmd 2>* value, just that the *<Cmd 2>* value is supported by that library. This means it is possible to replace the functionality performed with a specific *<Cmd 2>* value of a specific Entity Library without having to make any modification to the Application Layer.

Refer to the Entity Library Protocol documentation for the specific Entity Library to find out the specific protocol information for that Entity Library. Each library is responsible for validating the *<Data Layer>* information that is passed and what that data is used for.

1.6 Response Codes and Status Bytes

Response Status Byte Codes

The response packets contain a response status byte. This byte is always the first byte of the data area of the response packet. The data following the response status byte depends on the Request Packet sent, the device, the terminal type, and the response status byte returned. The response status byte may be the only byte in the data area of the response packet. The terminal does not return a response packet, if the Request Packet is invalid or has been sent to a device/terminal type that does not support the Request Packet. The following table provides a list of error commands. **Note:** Not all error commands are applicable to each Request Packet.

Error Codes	Error
00	ERROR_NONE
01	ERROR_TOKEN_NOT_PRESENT
05	ERROR_INVALID_RF_FORMAT
0E	ERROR_DEVICE_ID_INVALID
10	ERROR_ILLEGAL_ACTION
11	ERROR_WRONG_DOWNLOAD_STATE
12	ERROR_WRITE_FAILED
13	ERROR_INVALID_ADDRESS

14	ERROR_INVALID_BAUD
15	ERROR_INVALID_CHECK_DIGITS
16	ERROR_NO_TIMER_AVAILABLE
17	ERROR_INVALID_ENTITY_ID
18	ERROR_DATA_TRUNCATED
19	ERROR_NO_DATA_READ
1A	ERROR_INVALID_START_BYTE
1B	ERROR_INVALID_CRC
1C	ERROR_CMD_REPLY_MISMATCH
20	ERROR_14443_A_DATA_INCORRECT
21	ERROR_14443_B_DATA_INCORRECT
22	ERROR_14443_B_TOKEN_NOT_FOUND
23	ERROR_HF_ASIC_RECEIVE_TIMEOUT
24	ERROR_HF_ASIC_ABORTED
25	ERROR_HF_ASIC_ATQB_ERR1
26	ERROR_HF_ASIC_ATQB_PROT_TYPE
27	ERROR_HF_ASIC_INVALID_CID
28	ERROR_HF_ASIC_INVALID_NAD
29	ERROR_HF_ASIC_CID_LOW_POWER
2A	ERROR_14443_B_HLTB_ERROR
2B	ERROR_14443_B_INVALID_BLK_TYPE
2C	ERROR_HF_ASIC_NOT_IBLOCK
2D	ERROR_HF_ASIC_NOT_RBLOCK
2E	ERROR_14443_B_SDESELECT_ERROR
2F	ERROR_14443_DATA_INCORRECT
30	ERROR_MANY_CID_NO_SUPRT_TRANSPONDERS
31	ERROR_COLISN_BPSK_AND_OR_CID
32	ERROR_COLISN_BPSK_DECODE
33	ERROR_14443_B_ABORTED
34	ERROR_TOKEN_BUFFER_FULL
35	ERROR_14443_A_UPLINK_PARITY
36	ERROR_14443_A_ATS
37	ERROR_14443_A_PPS
38	ERROR_14443_A_CASCADE_LEVEL
39	ERROR_14443_A_SAK_CRC
40	ERROR_BPSK_NO_ERROR
41	ERROR_BPSK_BAD_FRAME_WAIT
42	ERROR_BPSK_BAD_VAL_BAUD
43	ERROR_BPSK_CANCELLED
44	ERROR_BPSK_SUBCARRIER
45	ERROR_BPSK_TR0_TIMEOUT
46	ERROR_BPSK_RCV_OVERFLOW
47	ERROR_BPSK_NO_SOF
48	ERROR_BPSK_NO_EOF
49	ERROR_BPSK_TR1_TIMEOUT
4A	ERROR_BPSK_CRC_ERROR
4B	ERROR_BPSK_FRAME_ERROR
4C	ERROR_MODULE_ABORTED
4D	ERROR_PARAMETER_ERROR
57	ERROR_COLLISION_DETECT
60	ERROR_APOLLO_LIFE_CYCLE
61	ERROR_APOLLO_DATA_INCORRECT

Figure 1-4: Response Status Byte Error Codes

Entity Library Module Table

The following table lists the entity Library modules available in the MFR terminal. This table can used to identify the Software Versions, where to direct packet information and to identify where response data originated.

Entity Module ID	Description of the Entity Modules MFR
01	Application Layer
02	14443-A Module
03	14443-B Module
04	15693 Module
05	Tag-it ™ Module
06	LF DST Module
07	14443-4 Module
08	TI Apollo Module
11	Boot Code Module

Figure 1-5: Entity Library Module Table

Download Mode Table

The following table lists the download codes associated with each download state available on the MFR terminal.

Download State #	Brief description
0	Not in Download Mode
1	Application has forced to Download Mode
2	Authentication Failed
3	Download Required
4	Memory Write

Figure 1-6: Download Mode Table

Default RF Priority Table

The contents of this table indicate the current RF Priority default values. Each RF format returns a different token data when the Find Token Request has been issued. The far right column details the data that is returned for each RF format.

RF Technology Type	Data Field Value	RF Specific Data
14443 Type A	**02**	CID, Cascade Level, UID
14443 Type B	**03**	CID, PUPI
15693	**04**	Inventory Flag, DSFID, UID
Tag-it	**05**	SID
LF DST	**06**	MID, Serial #

Figure 1-7: Default RF Priority Table

Information:

The RF Technology Type matches the Data Field Value assigned in the Entity Module Table.

Driver Allocation Table

The Control Byte bits used in the Set Driver Request packet are detailed in the table below. These signify drive points for specific devices such as LED's, signal or sound drivers, etc.

Device	Bit # for Device
Device #1	**1**
Device #2	**2**
Device #3	**3**

Figure 1-8: Driver Allocation Table

Loop Count Timing Table

This table gives the approximate time interval that the terminal waits for a read based on the Loop Count.

Loop Count	Time in seconds
5	2.5
10	5
50	25

Figure 1-9: Loop Count Timing Table

Information:

Loop Count Timing values are calculated as the difference from when the Request Packet is sent and the Response Packet is received. The actual read time will be less due to the overhead, but this will give a rough estimate of the read duration with all formats turned on. The LF read function takes the longest time because it must send a power burst and modulated data and then wait for a response. The HF function loop time is negligible as it just looks for a signal. A Loop Count of **00** will loop until an actual token is detected.

1.7 MFR Packet Flow Charts

USART 0 Receiving Packets Flow Chart

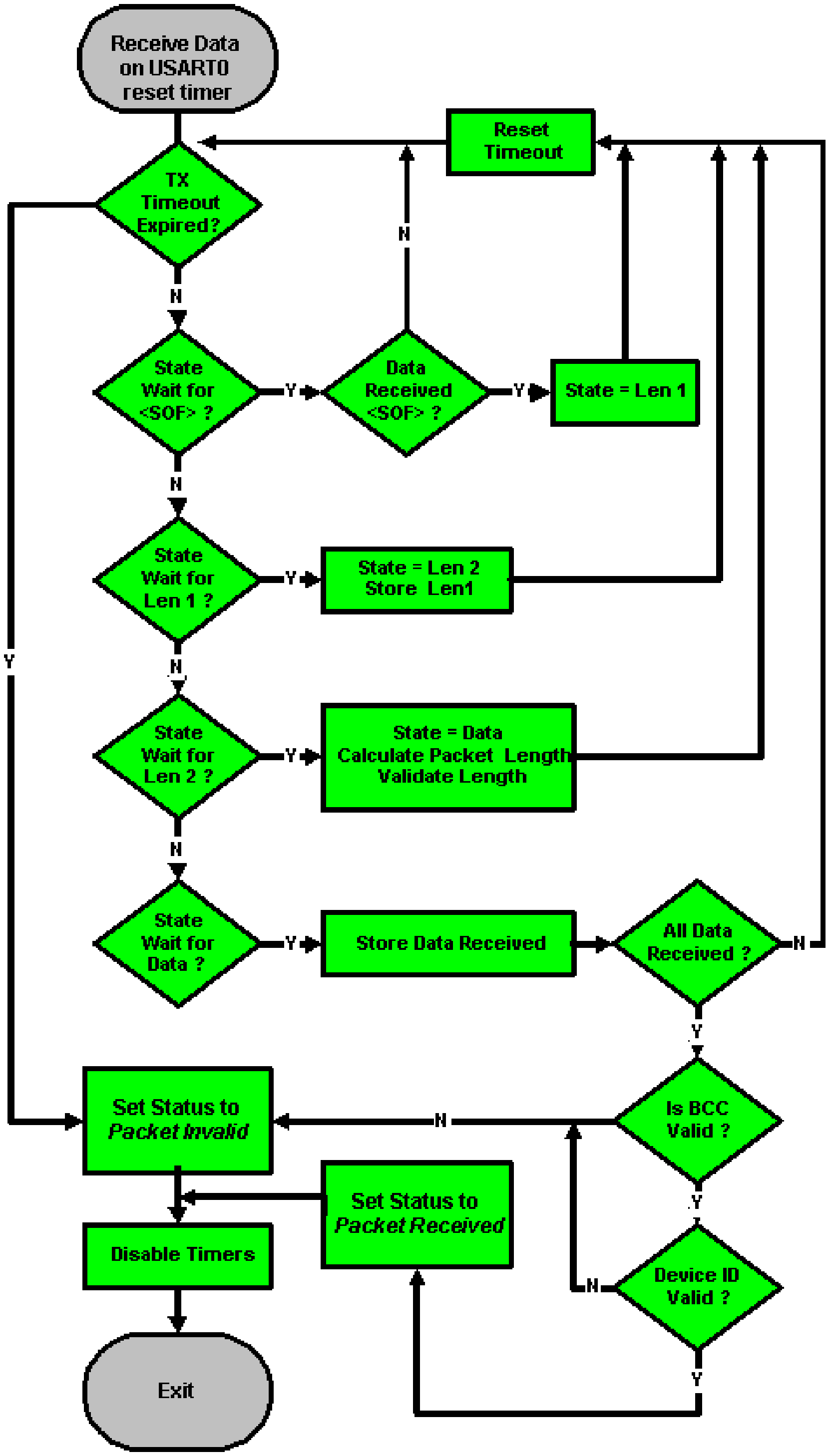


Figure 1-10: USART 0 Receiving Packets Flow Chart

Packet Parsing/Validation Flow Chart

Figure 2-13 is just an example of how the packet parsing flow works, and is not intended to cover every command.

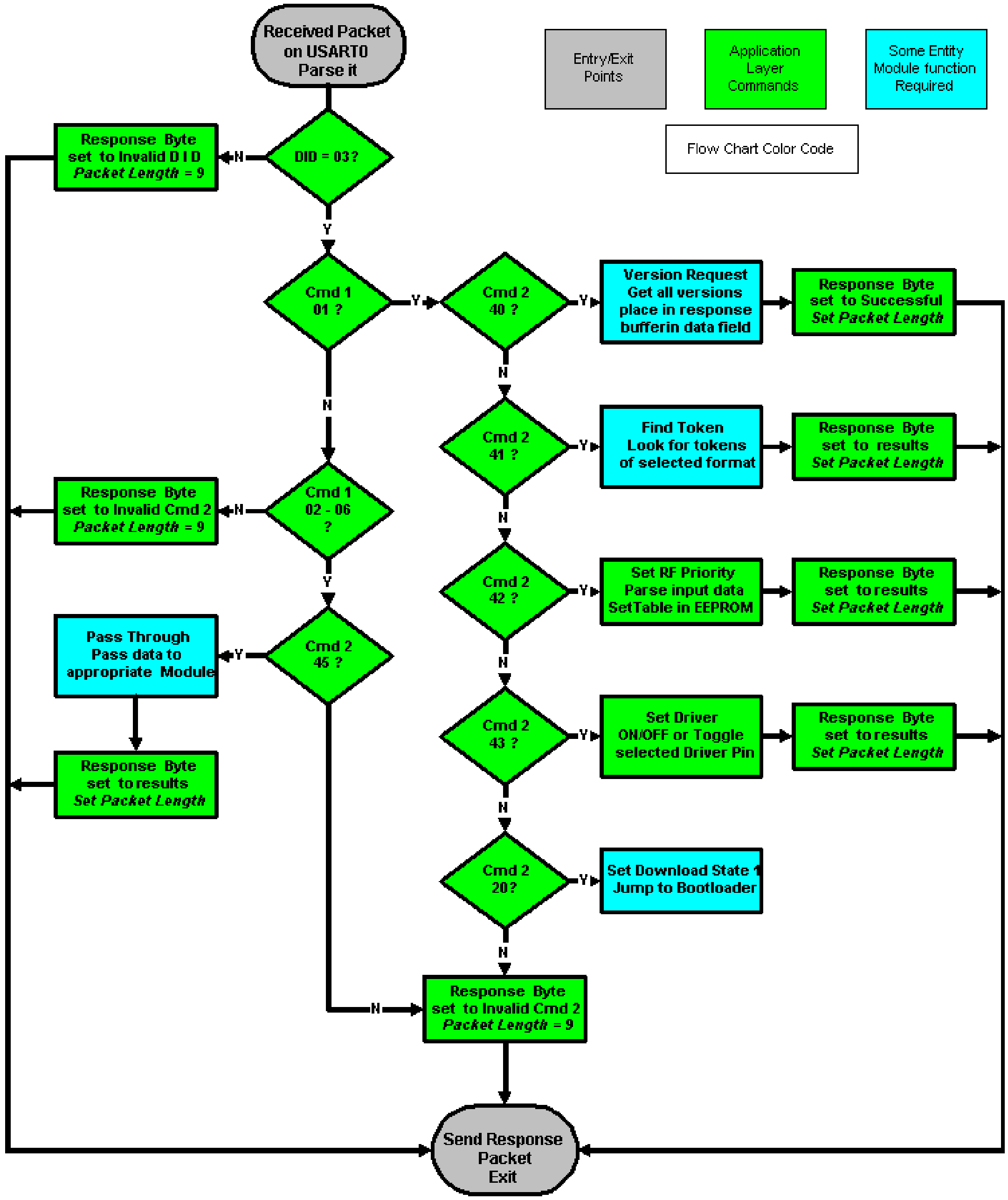


Figure 1-11: Packet Parsing/Validation Flow Chart

Version Request Flow Chart

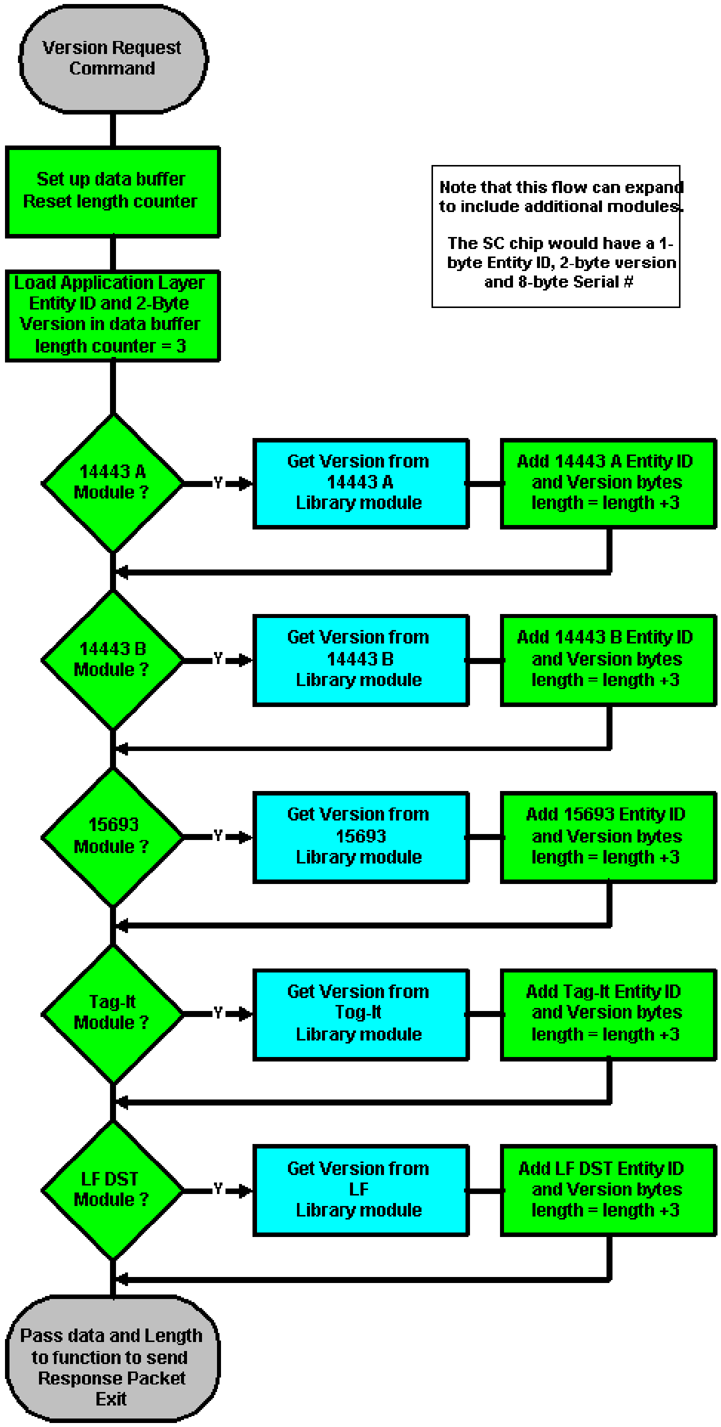


Figure 1-12: Version Request Flow Chart

Find Token Request Flow Chart

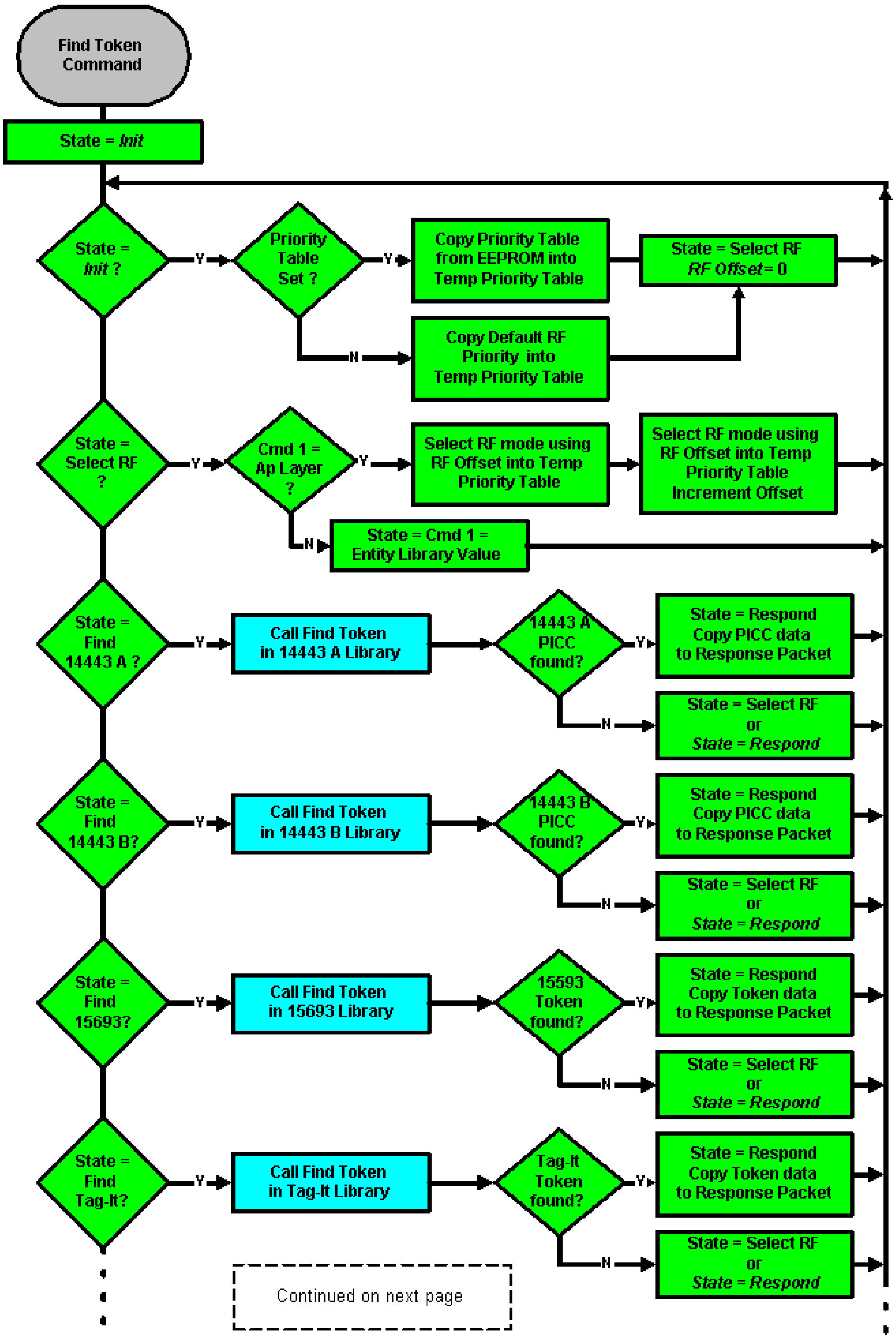


Figure 1-13: Find Token Request Flow Chart

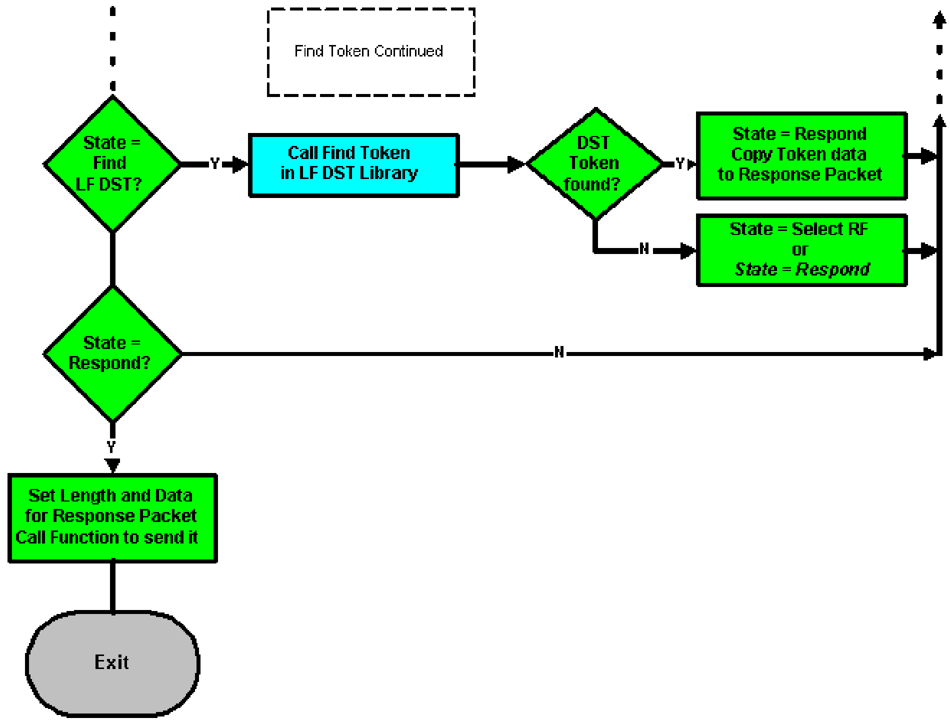


Figure 1-14: Find Token Request Flow Chart (Continued)

Set RF Format Priority Flow Chart

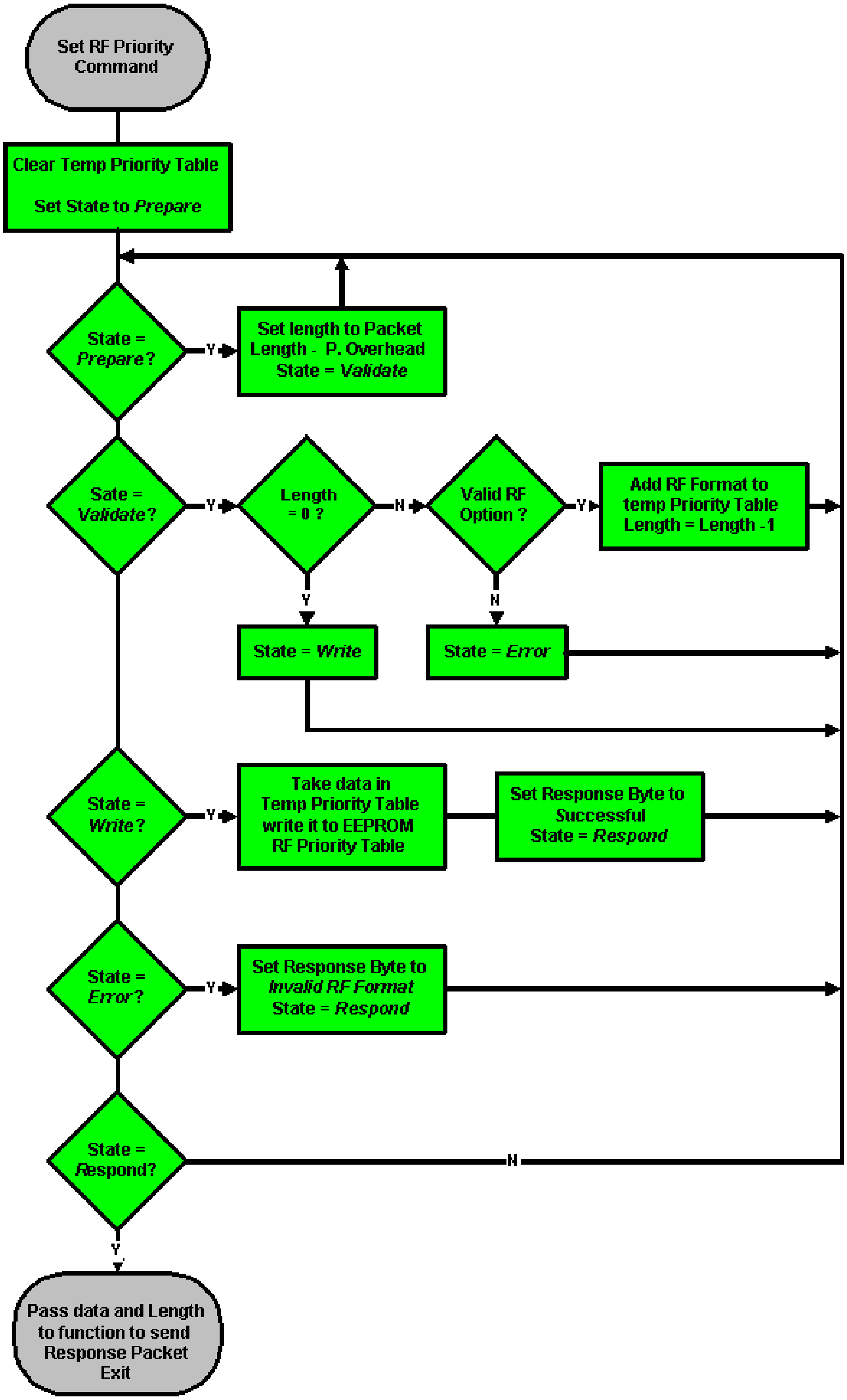


Figure 1-15: Set RF Format Priority Flow Chart

Pass-Through Request Flow Chart

The following flow chart does not include all of the Entity Libraries, just an indication of how it works.

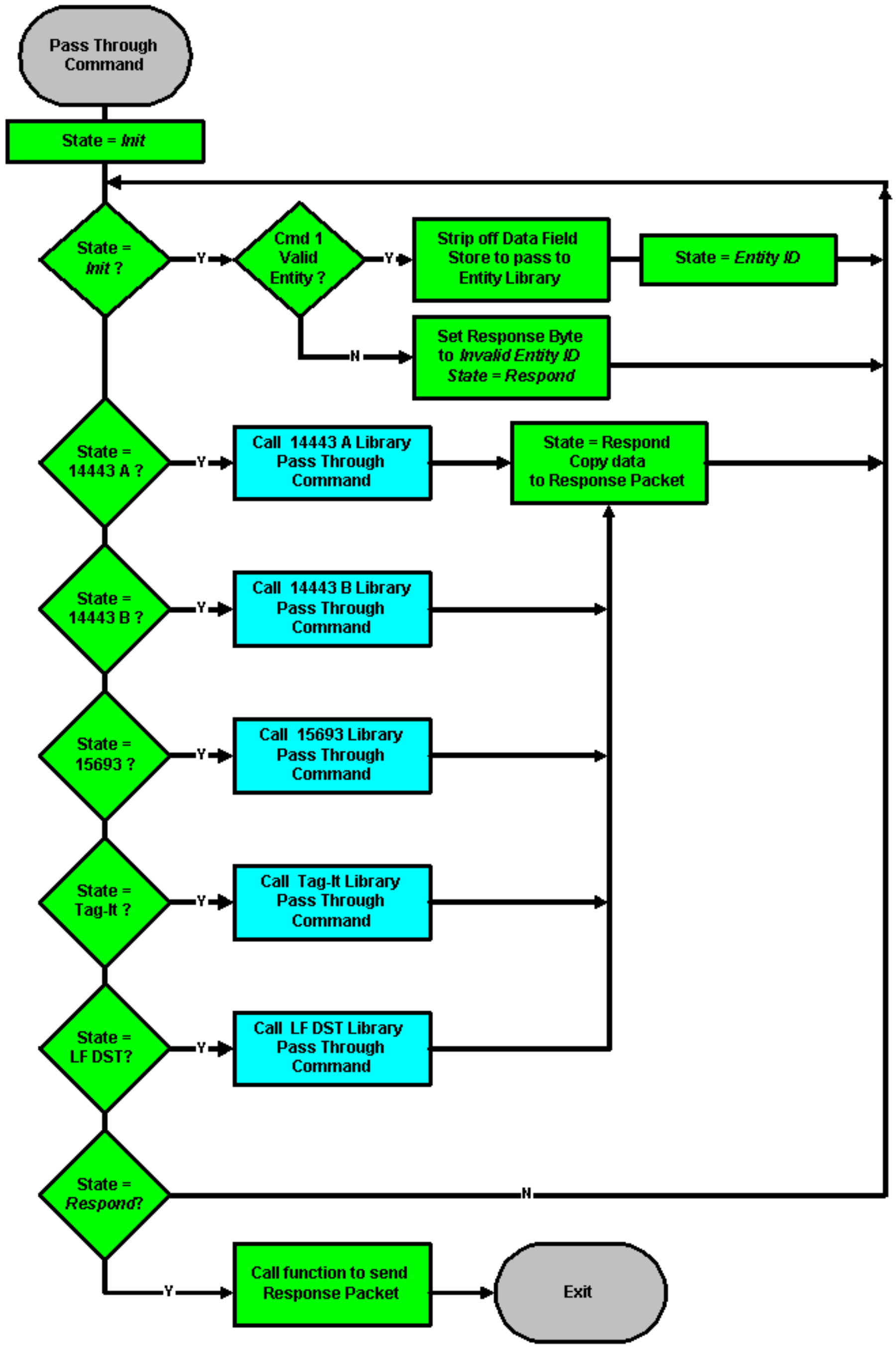


Figure 1-16: Pass-Through Request Flow Chart

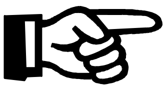
Chapter 2

Regulatory and Warranty Notices

Topic **Page**

2.1 FCC Conformity

The Series 4000 Multi-Function Reader is an intentional radiator. The transmitter portion operates at 13.56 MHz and is subject to FCC Part 15, Subpart C, “Intentional Radiator,” paragraph 15.225 (13.553-13.567MHz). Radiated emissions from the device are subject to the limits in Section 15.209 of the Rules outside of the 13.56 +/- 0.007 MHz band.



Note:

Any device or system incorporating the Series 4000 reader, in full or in part, needs to obtain FCC certification as part of the system within which this reader unit resides. A system containing this product may be operated only under an experimental license or final approval issued by the relevant approval authority. Before any such device or system can be marketed, an equipment authorization must be obtained form the relevant approval authority.

2.2 ETSI Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to comply with European Standard EN300330. It is the responsibility of each system integrator to have their complete system tested and to obtain approvals as required from the local authorities before operating or selling this system.

2.3 CE Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to have a CE Declaration of Conformity stating that it meets European EMC directive 99/5/EC. This must be issued by the system integrator or user of such a system prior to marketing or operating it in the European community.

2.4 Warranty and Liability

The "General Conditions of Sale and Delivery" of Texas Instruments Incorporated or a TI subsidiary apply. Warranty and liability claims for defect products, injuries to persons and property damages are void if they are the result of one or more of the following causes:

- Improper use of the reader module.
- Unauthorized assembly, operation and maintenance of the reader module.
- Operation of the reader modules with defective and/or non-functioning safety and protective equipment.
- Failure to observe the instructions during transport, storage, assembly, operation, maintenance and setting up of the reader modules.
- Unauthorized changes to the reader modules.
- Insufficient monitoring of the reader modules' operation or environmental conditions.
- Improperly conducted repairs.
- Catastrophes caused by foreign bodies and acts of God.