IT2410 Tag Programmer
User Guide

TransCore
3410 Midcourt Road, Suite 102
Carrollton, Texas 75006

January 2012

P/N 411360-007

For further information, contact:

TransCore
3410 Midcourt Road, Suite 102
Carrollton, Texas 75006 USA

Phone: (214) 461-4031
Fax: (214) 461-6478

Technical Support
Web: transcore.com/rfidsupport
Phone: (505) 856-8007

For comments or questions about this document, e-mail tech.pubs@transcore.com.

WARNING TO USERS IN THE UNITED STATES

FEDERAL COMMUNICATIONS COMMISSION (FCC) RADIO FREQUENCY INTERFERENCE STATEMENT
47 CFR §15.105(a)

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the Federal Communications Commission (FCC) rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency (RF) energy and may cause harmful interference to radio communications if not installed and used in accordance with the instruction manual. Operating this equipment in a residential area is likely to cause harmful interference, in which case, depending on the laws in effect, the users may be required to correct the interference at their own expense.

NO UNAUTHORIZED MODIFICATIONS
47 CFR §15.21

CAUTION: This equipment may not be modified, altered, or changed in any way without permission from TransCore, Inc. Unauthorized modification may void the equipment authorization from the FCC and will void the TransCore warranty.

USE OF SHIELDED CABLES IS REQUIRED
47 CFR §15.27(a)

NOTE: Shielded cables must be used with this equipment to comply with FCC regulations.

NOTE: A license issued by the FCC is required to operate this RF identification device in the United States. Contact TransCore for additional information concerning licensing requirements for specific devices.

TransCore, Inc.
USA

Contents

Contents

4 IT2410 Tag Programmer Setup

A IT2410 Tag Programmer Specifications

B Interface and Power Supply Information

C Frequently Requested Tag Programming Information

D Application Program Interface

List of Figures

List of Tables

1

Before You Begin

Chapter 1

Before You Begin

This chapter describes this guide's purpose and intended audience. It provides a list of topics covered in each section, a list of related documents, and the symbols and typographical conventions used.

Purpose

This *IT2410 Tag Programmer User Guide* contains installation and operating instructions for the hardware used to perform IT2410 Tag Programmer programming functions with TransCore IT2200-series tags. Figure 1-1 illustrates the features of the IT2410 Tag Programmer.

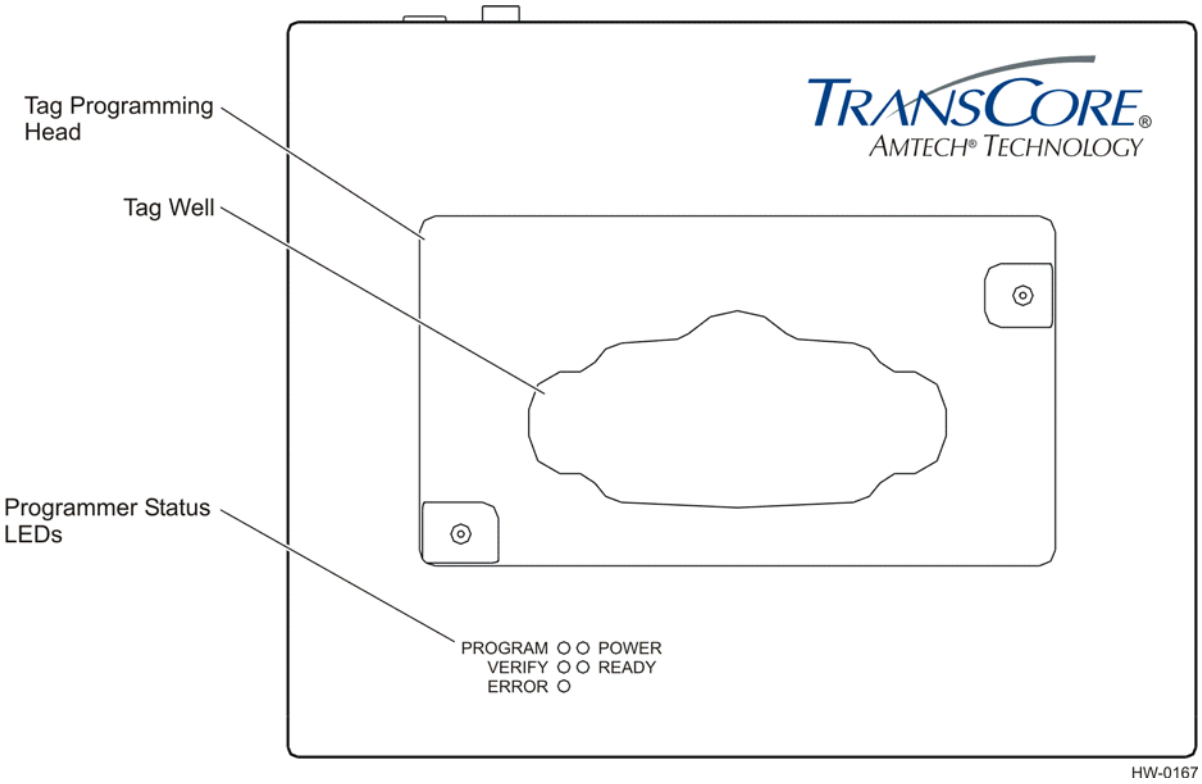

Figure 1-1 IT2410 Tag Programmer (Top View)

Intended Audience

The intended audience for this user guide includes all personnel who need to accomplish the following tasks:

- Connect the tag programmer to a host computer for the purpose of programming tags, operate a computer, and install electronic equipment by connecting components with cables and power supply cords.
- Develop host software.
- Understand basic tag programming.

Plan and Organize

TransCore IT2000-series tags that are programmed with the IT2410 Tag Programmer use sophisticated memory organization. TransCore recommends that you plan and organize the necessary steps for installing the tag programmer (refer to Chapter 4, "IT2410 Tag Programmer Setup," for instructions) and for programming the tags (refer to customer-supplied programming software instructions).

Guide Topics

This user guide contains the following chapters and appendixes.

Chapter 1 – Before You Begin	Describes the purpose, intended audience, guide topics, and document conventions.
Chapter 2 – IT2410 Tag Programmer Overview	Provides an overview of tag programmer functions and features.
Chapter 3 – IT2410 Tag Programmer Hardware	Describes the hardware components and requirements.
Chapter 4 – IT2410 Tag Programmer Setup	Provides information on setting up the tag programmer for first use.
Appendix A – IT2410 Tag Programmer Specifications	Provides the programmer's physical and environmental specifications.
Appendix B – Interface and Power Supply Information	Describes the interface configuration to a host computer. Also shows the power supply pin-outs.
Appendix C – Frequently Requested Tag Programming Information	Provides information regarding IT2200-series tag programming.

Appendix D – Application Programming Interface	Describes application program interface information on tag programming methodology.The protocol information that is presented provides reference information relevant to developing host software.

Typographical Conventions

The following typographical conventions are used in this guide (Table 1-1).

Table 1-1 Typographical Conventions

Convention	Indication
	This procedure might cause harm to the equipment and/or the user.
	Alerts user to concerns about a procedure.
`Code`	Code, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text appear in courier typeface.
Dialog Box Title	Title of a dialog box as it appears on screen.
Function	Start with the characters G4, mixed case with no underscores, and include parentheses after the name, as in G4FunctionName().
Menu Item	Appears on a menu.
Note	Auxiliary information that further clarifies the current discussion. These important points require the user's attention. The paragraph is in italics and the word Note is om boldface.
NUL	Zero-value ASCII character or a zero-value byte.
NULL	Zero-value pointers are null-terminated strings that refer to strings of printable ASCII characters with a zero-value byte placed in memory directly after the last printable character of the string.

2

IT2410 Tag Programmer Overview

Chapter 2

IT2410 Tag Programmer Overview

This chapter presents an overview of the IT2410 Tag Programmer.

Purpose of Tag Programmer

The tag programmer is a multi-functional product used to program TransCore IT2200-series tags. Programmer functions include frame programming, fixed-frame locking, and data frame interrogation.

Communications

The tag programmer connects to a personal computer (PC) serial port that complies with the RS–232 communications interface standard. TransCore offers a PC-to-programmer RS–232 serial cable in an optional tag programmer accessory kit. An Ethernet interface connection has been provided as a future communications feature but is not implemented in this version.

Programming Head

The programming head on the top of the tag programmer provides a mechanical interface to the tag. The programming head includes a drop-in tag well that is compatible with some IT2200-series tags. By placing the tag in the tag well the tag is correctly positioned to the programmer's internal antenna. Figure 2-1 shows the IT2410 Tag Programmer head.

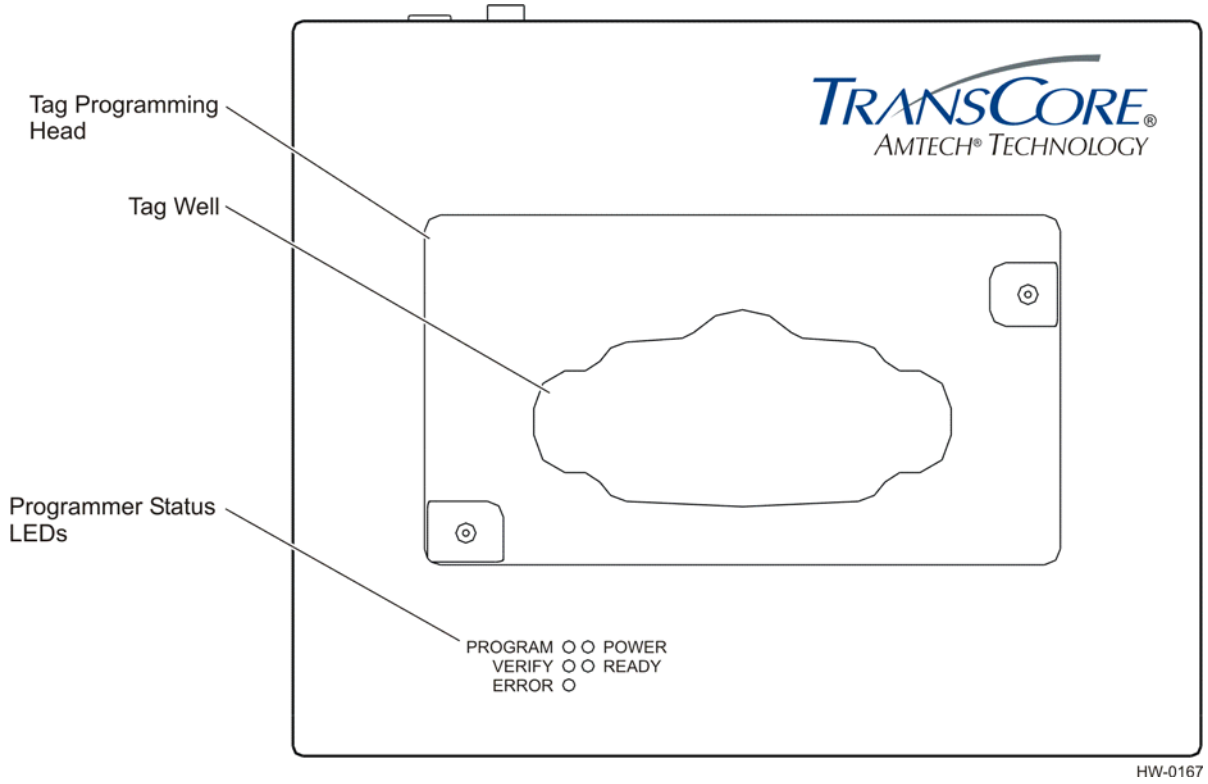

Figure 2-1 Tag Programming Head (Top View)

Indicator LEDs

The tag programmer uses light-emitting diodes (LEDs) to indicate the status of its operations. These LED indicators identify operation, functionality, and errors. Figure 2-2 illustrates the external status indicators found on the programmer.

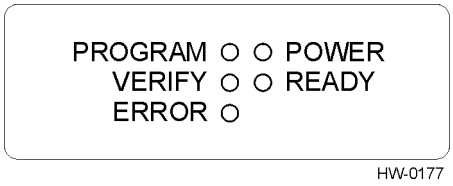

Figure 2-2 IT2410 LED Status Lights

Table 2-1 lists the indicator lights and describes their meanings.

Table 2-1 Indicator LEDs, Descriptions, and Actions

INDICATOR LED	DESCRIPTION	ACTION
PROGRAM (GREEN)	The tag is being programmed with user-specified data.	Status only — no action required by user
VERIFY (GREEN)	The tag programmer has read valid data from the specified tag frame.	Status only — no action required by user
ERROR (RED)	The tag programmer has detected an error in the programming or verifying process or during other operations.	Check system configuration, cabling, and power supply to system. Retry programming sequence after system check. If system still not working, contact TransCore for support.[a]
POWER (GREEN)	Power is being supplied to the tag programmer.	Status only — no action required by user
READY (GREEN)	The tag programmer is ready to accept commands from the PC.	Status only — no action required by user

a. Refer to page iii for the TransCore Technical Support telephone and fax numbers.

Standard Formats

The IT2410 Tag Programmer can code tag data using formats specified by wireless communications standards, such as the California Title 21 Regulation for AVI Compatibility.

Anti-static Wrist Strap

The IT2410 Tag Programmer has a connection for an anti-static wrist strap. You must wear a wrist strap when programming tags. The wrist strap prevents electrostatic discharge (ESD) damage to tag circuits. The wrist strap banana plug (Figure 2-3) inserts into the socket in the front of the tag programmer (Figure 2-4). TransCore offers an anti-static wrist strap in an accessory kit.

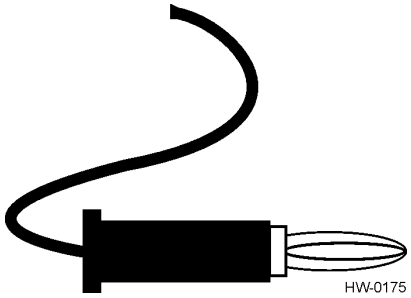

Figure 2-3 Banana Plug on ESD Wrist Strap

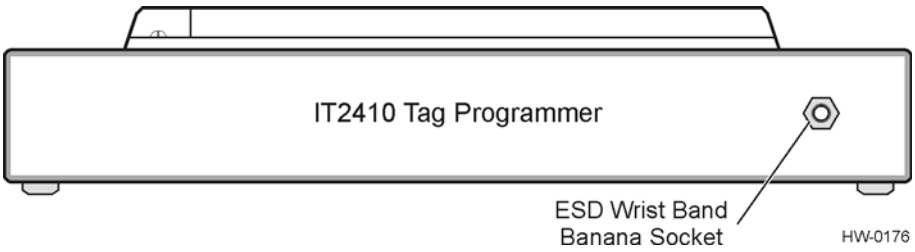

Figure 2-4 ESD Wrist Band Socket Location

Power

The IT2410 Tag Programmer requires 12V DC to operate. TransCore offers a 12V DC power supply in an accessory kit. Refer to Appendix B to this guide for a pin-out diagram of the power connector.

The tag programmer has an installed 5-amp (A), 250V Fast Blo fuse for excessive current protection.

Caution

If needed, replace the fuse with a 5A, 250V Fast Blo fuse only.

3

IT2410 Tag Programmer Hardware

Chapter 3

IT2410 Tag Programmer Hardware

This chapter describes the tag programmer's hardware components and requirements.

Components

The IT2410 Tag Programmer is a stand-alone component. TransCore offers an accessory kit that includes a power supply, interconnecting cables, and anti-static wrist strap. The hardware is designed to operate with a personal computer (PC). This section describes the hardware requirements for proper operation of the IT2410 Tag Programmer, and includes requirements for electrostatic discharge (ESD) protection.

Hardware Requirements

The customer-supplied PC requires the following minimum specifications:

- PC with Intel Pentium (or equivalent microprocessor) 90 MHz or higher
- Microsoft Windows 2000, 98/95, or NT 4.0 with Service Pack 3 or higher
- 32 MB RAM
- CD-ROM drive
- Available serial port
- VGA or high-resolution monitor
- One 3.5 in. floppy drive, one CD-ROM drive, and a hard drive
- Mouse or other Windows pointing device

ESD Protection Requirements

If you do not take proper precautions, ESD damage can occur to the programmer and tags during programming. The IT2410 Tag Programmer has an external banana plug socket that provides a point of attachment for an ESD wrist strap. You can obtain a wrist strap by purchasing an accessory kit from TransCore.

Static discharge may cause significant damage that can adversely affect a tag's operating performance. The following are typical symptoms of static discharge:

- Inability to program the tag
- Greatly reduced operating range
- Tag operating failure

Static is generated by friction and can often build to damaging levels. The following actions can cause static:

- Shoes moving across a carpeted or plastic floor
- Hot air blowing into a room from a hot-air duct
- Rubbing tags together
- Sliding tags across a table top
- Friction created while wearing certain types of clothing

ESD Workstation Design

Well-designed workstations use a system of multiple protection elements. At a minimum, this protection should include anti-static bench mats and wrist straps to discharge static safely away from the equipment.

As an additional precautionary measure to prevent static damage, keep tags in their original packaging, on an anti-static mat, or within an ESD-inhibiting container until you are ready to program the tags.

Using an ESD Wrist Strap

Slip on the anti-static wrist strap and insert the banana plug into the ESD banana socket on the front of the tag programmer (Figure 2-3 on page 2-6).

Caution

Always put on an ESD wrist strap before beginning programming operations.

4

IT2410 Tag Programmer Setup

Chapter 4

IT2410 Tag Programmer Setup

This chapter explains how to set up the tag programmer for first use.

Equipment List

The IT2410 Tag Programmer includes the tag programmer unit. The tag programmer includes a 5-amp (A), 250V Fast Blo fuse for excessive current protection.

Caution
If needed, replace the fuse with a 5A, 250V Fast Blo fuse only.

In addition to a personal computer (PC), the user must have the following components available:

- External 12V DC power module
- Electrostatic discharge (ESD) anti-static wrist strap
- PC-to-programmer RS–232 serial cable with DB-9 connectors on both ends

TransCore offers an accessory kit that includes the 12V DC power supply, wrist strap, and interface cable.

Figure 4-1 shows the tag programmer and peripheral components.

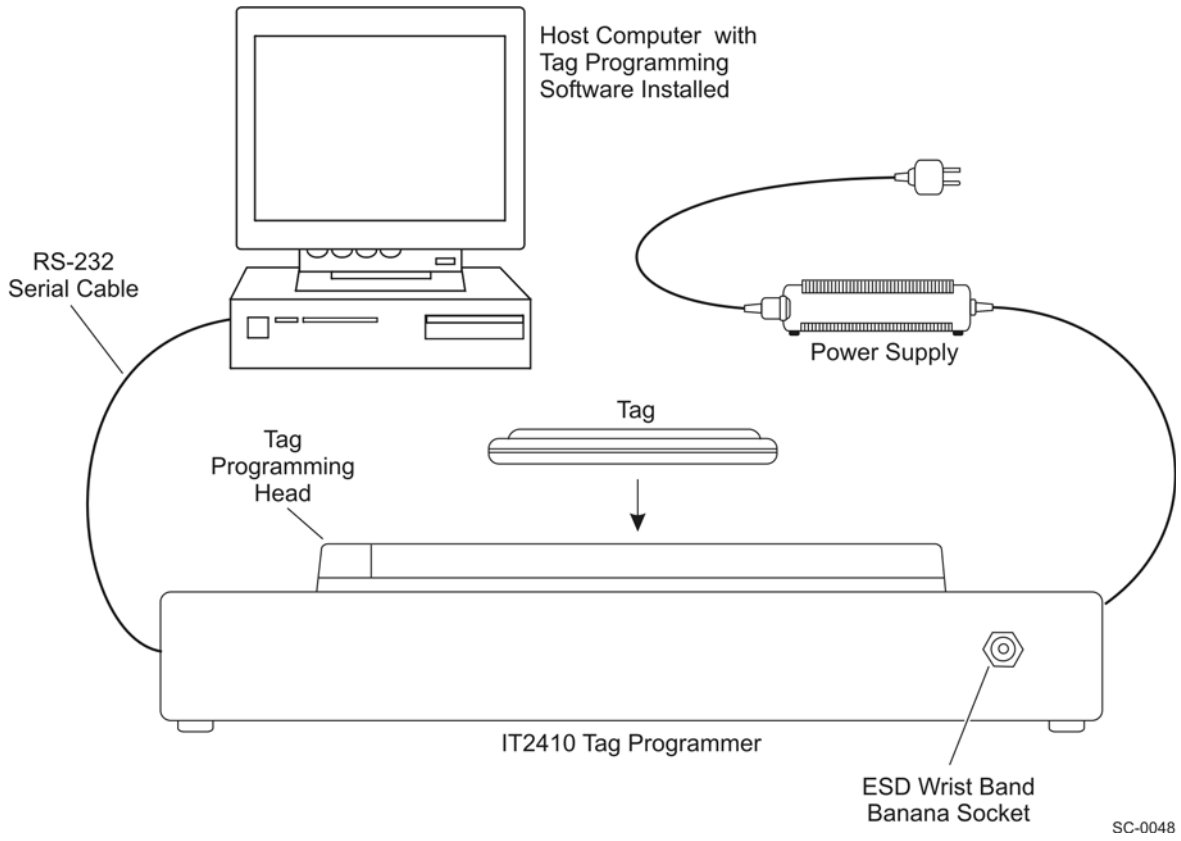

Figure 4-1 IT2410 Tag Programmer System (Front View)

Note: *To maintain FCC Part 15 Class A compliance, operate the tag programmer in a horizontal position.*

Connecting the IT2410 Tag Programmer to a PC

Use the following procedure to connect the equipment:

Caution

User must wear an ESD wrist strap when programming tags. Failure to do so can result in ESD damage to the tag.

1. Place the tag programmer on a flat, stable surface.
2. Connect the 12V DC power supply to the power socket on the tag programmer's back panel.
3. Connect a PC-to-programmer RS–232 serial cable to the COM1 port on the PC. Connect the other cable end to the RS–232 connector on the tag programmer's back panel.

Note: *The Ethernet connection located on the back panel is not implemented in this version of the IT2410 Tag Programmer.*

4. Connect the ESD wrist band banana plug to the socket on the tag programmer's front panel.

Figure 4-2 shows the location of the power and communications interface connections on the tag programmer's back panel.

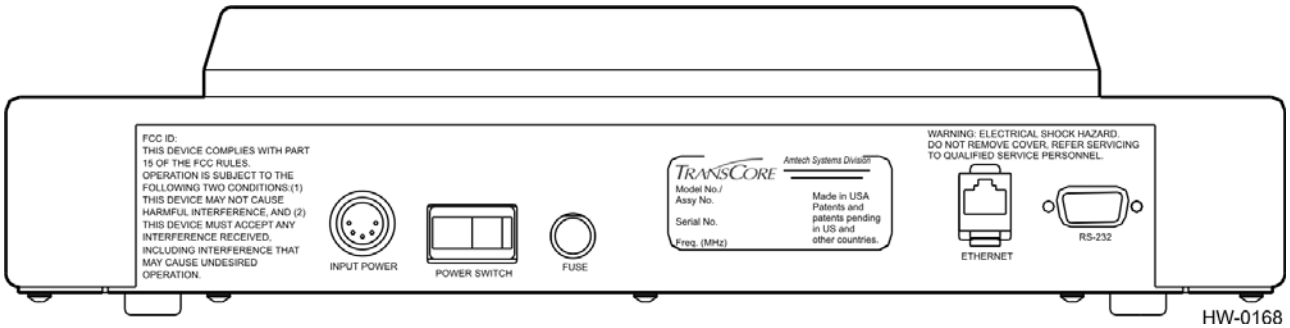

Figure 4-2 Programmer's Power and Communications Interface Connections (Back View)

Powering Up

When all the equipment is connected as described in the preceding section, apply power to the system components as follows:

1. Switch on the PC and verify its proper operation. Ensure that the time and date settings are correct. Incorrect time and date settings can prevent the tag programmer from powering up correctly.

Note: *Do not place a tag in the programmer head at this time. The indicator light-emitting diodes (LEDs) operate differently when a tag is already in place.*

2. Switch on the tag programmer's power switch located on the programmer's rear panel. On the top of the programmer, the green POWER LED illuminates.

Note: *The operating system will initiate a self-test upon power up or after power interruption.*

When the tag programmer's internal power-on self-test is successfully completed, the user is notified as follows:

- The green READY indicator light illuminates.
- After approximately 10 seconds, the tag programmer sounds three short tones in rapid succession.

The tag programmer incorporates an audible indicator that is controlled by software. The indicator's sound is pleasing to the ear (2200 Hz) and can be heard easily while operating near the unit under normal conditions.

The IT2410 Tag Programmer is now ready to start programming tags.

If the tag programmer fails to power up, the READY LED goes out, and the ERROR LED illuminates and the alarm beeps once for one second. Check time and date settings on PC and restart powering up sequence.

Programming Cycle

When the PC initiates the programming cycle, the PROGRAM LED illuminates and the alarm beeps for an approximate duration of 0.25 seconds, once a second for a 10-second period, or until the programming operation has completed.

The operator has an approximate 10-second time-out period in which to place an operating IT2200-series tag on the programming head fixture. Once the tag is in the fixture and the programmer detects the valid RF data, the programmer executes the specific write command from the host software.

If the programming is successful, the PROGRAM LED goes out and the alarm sounds three times for approximately 0.25 seconds each.

If the programming cycle is unsuccessful or if a tag is not detected within the 10-second period, the ERROR LED illuminates and the alarm sounds once for approximately 1 second.

Verifying Cycle

When the PC initiates the verify cycle, the VERIFY LED illuminates and the alarm beeps each second for an approximate 10-second period or until the verify operation is complete.

The operator has a 10-second time-out period in which to place an operating IT2200-series tag on the programming head. Once the tag is placed in the fixture and the programmer can detect the valid RF data, the programmer executes the specific read command from the host software.

If the verification is successful, the VERIFY LED goes out and the alarm sounds three short beeps.

If the verifying cycle is unsuccessful or if a tag is not detected within the 10-second period, the ERROR LED illuminates and the alarm beeps for 1 second.

The alarm audibly indicates the status of programmer operations. Table 4-1 lists the various programmer operations and the corresponding alarms that sound as an audible operation verification.

Table 4-1 Programmer Operations and Corresponding Alarms

Programmer Operation	Audible Alarm	Beep Duration
Successful Self Test	3 beeps	0.25 s
Failed Self Test	1 beep	3.0 s
Programming Cycle	1 beep per second for 10 s	0.25 s each

Table 4-1 Programmer Operations and Corresponding Alarms (continued)

Programmer Operation	Audible Alarm	Beep Duration
Successful Programming Cycle	3 beeps	0.25 s each
Unsuccessful Programming Cycle	1 beep	3.0 s
VERIFY Cycle	1 beep per second for 10 s	0.25 s each
Successful VERIFY Cycle	3 beeps	0.25 s each
Unsuccessful VERIFY Cycle	1 beep	3.0 s

A

IT2410 Tag Programmer Specifications

Appendix A

IT2410 Tag Programmer Specifications

This appendix lists the IT2410 Tag Programmer physical and environmental specifications.

Table A-1 lists the IT2410 Tag Programmer parameters and specifications.

Table A-1 IT2410 Tag Programmer Physical and Environmental Specifications

Parameter	Specification
Size (W x H x D)	14.3 x 3.2 x 11.5 in (36.3 x 8.1 x 29.2 cm)
Weight	6.8 lb (3.1 kg)
Input Voltage	12V DC
Power Consumption	36 W
Excessive Current Protection	5-amp, 250V Fast Blo fuse
Operating Temperature	32°F to +122°F (0°C to +50°C)
Storage Temperature	-4°F to +185°F (-20°C to +85°C)
Humidity	95% noncondensing @77°F to 131°F (25°C to 55°C)
Enclosure	Dustproof
Operational Vibration	1.04 G_{rms}, 5-500 Hz, power spectral density-uniform 0.0022 G^2/Hz, 1 hour per axis
Shock	4 $G_{zero\text{-}to\text{-}peak}$ by 11-ms half-sine duration in all 3 axes
PC-to-Programmer Cable	RS–232 (data terminal equipment)
RF Power	Programming head: 1.0 mW
FCC Classification	Part 15, site license not required

Environmental

The programmer is designed to operate in typical office environment conditions. The programmer performs to the specifications listed in Table A-1 and is ready to program with a warm-up time of not more than three minutes.

Mean Time Between Failures

The programmer has a minimum mean time between failures of 20,000 hours.

B

Interface and Power Supply Information

Appendix B

Interface and Power Supply Information

The tag programmer's PC interface complies with the RS–232 standard for data communications equipment and uses the protocol settings listed in Table B-1.

Table B-1 PC Interface Protocol Settings

Protocol	Setting
Data Rate	19,200 bps (factory default) or 38,400 bps
Data Bits	8
Parity	None
Stop Bits	1
Software Flow Control	None
Hardware Flow Control	None

Figure B-1 shows the pin-out locations for the power plug.

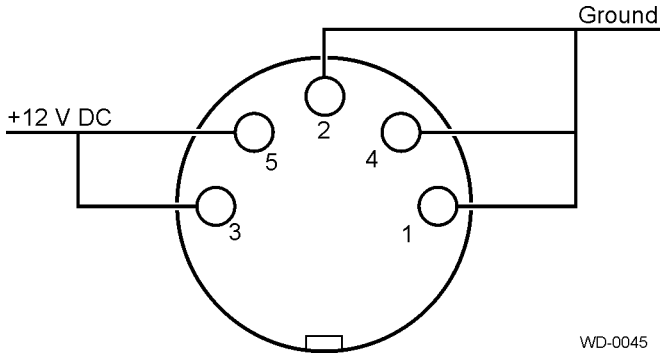

Figure B-1 Pin-out Diagram for Power Plug

C

Frequently Requested Tag Programming Information

Appendix C

Frequently Requested Tag Programming Information

This appendix explains frequently requested information regarding IT2200-series tag programming information.

Identification, Serial Number, and the Global Password

TransCore documentation and/or host software references a tag identification number (ID) that corresponds to the Title 21 ID. The Title 21 ID is a unique identifier used by radio frequency identification (RFID) systems in California. This is found in the first 8 hexadecimal characters (32 bits) of the tag's general page 1. In all IT2200-series tags, this ID is either the same as the serial number from page 0, or other application specifications.

The tag serial number (S/N) is the first 8 hexadecimal characters (32 bits) in general page 0. The global password is 64 bits long and the default is generated by repeating the S/N twice. Once this global password is known, it can be used to change read and write passwords on any of the general pages of data in the tag, change which frames are designated nonvolatile, and lock pages of data so that they cannot be changed unless the page is unlocked first using the global password.

These two IDs are in addition to any customer-specific IDs. Most of the customers that use the IT2200-series tags employ one of the other general pages of data available in the tag to store their own unique ID, which usually corresponds to a patron account number.

The global password in the IT2200-series tag acts like a master key and can be used to perform crucial tag operations. The password is meant to be controlled by the owner agency. It has a default setting that is created from the tag's unique S/N, but can be changed by the owner agency, if desired. Table C-1 shows the tag page layout.

Table C-1 Tag Page Layout

Page Number (HEX)	Field Name	Area	Length (bytes)
*0000H	Configuration page (reserved for factory use)	General	16
*0001H	General page 1/diagnostics	General	16
0002H	General page 2	General	16
0003H	General page 3	General	16

Table C-1 Tag Page Layout (continued)

Page Number (HEX)	Field Name	Area	Length (bytes)
0004H	General page 4	General	16
0005H	General page 5	General	16
0006H	General page 6	General	16
0007H	General page 7	General	16
0008H	General page 8	General	16
0009H	General page 9	General	16
000AH	General page A	General	16
000BH	General page B	General	16
000CH	General page C	General	16
000DH	General page D	General	16
000EH	General page E	General	16
000FH	General page F	General	16
*0000H	User password for page 0000H	User	4
*0001H	Read user password for page 0001H	User	4
*0002H	Read user password for page 0002H	User	4
*0003H	Read user password for page 0003H	User	4
*0004H	Read user password for page 0004H	User	4
*0005H	Read user password for page 0005H	User	4
*0006H	Read user password for page 0006H	User	4
*0007H	Read user password for page 0007H	User	4
*0008H	Read user password for page 0008H	User	4
*0009H	Read user password for page 0009H	User	4
*000AH	Read user password for page 000AH	User	4
*000BH	Read user password for page 000BH	User	4
*000CH	Read user password for page 000CH	User	4
*000DH	Read user password for page 000DH	User	4
*000EH	Read user password for page 000EH	User	4
*000FH	Read user password for page 000FH	User	4
*0014H	Write user password for page 0000H	User	4

Table C-1 Tag Page Layout (continued)

Page Number (HEX)	Field Name	Area	Length (bytes)
*0015H	Write user password for page 0001H	User	4
*0016H	Write user password for page 0002H	User	4
*0017H	Write user password for page 0003H	User	4
*0018H	Write user password for page 0004H	User	4
*0019H	Write user password for page 0005H	User	4
*001AH	Write user password for page 0006H	User	4
*001BH	Write user password for page 0007H	User	4
*001CH	Write user password for page 0008H	User	4
*001DH	Write user password for page 0009H	User	4
*001EH	Write user password for page 000AH	User	4
*001FH	Write user password for page 000BH	User	4
*0020H	Write user password for page 000CH	User	4
*0021H	Write user password for page 000DH	User	4
*0022H	Write user password for page 000EH	User	4
*0023H	Write user password for page 000FH	User	4
*0000H	Page lock bits	Owner	2
*0001H	Reserved for read password lock	Owner	2
*0002H	Reserved for write password lock	Owner	2
*0003H	Page NVRAM bits	Owner	2
*0004H	Global password	Owner	8

Note: ** These pages are write only.*

Changing the Global Password

As with personal IDs (PIN), which are used with automatic teller machines, the first time that a tag's global password is changed, it is from the default setting. You must know the tag's current global password before you can change it. To change it, you must execute a write request to owner page 4. The host software that is being used to communicate with the reader should prompt you for the current global password as well as the new password. Once you change the global password, safeguard it to prevent unauthorized access to the tag.

Reading and Setting Nonvolatile Status for General Pages

Note: *You should make any critical information in the tag nonvolatile, or the information will be lost when the tag batteries run down or are changed.*

To find out which general pages of data have been set to nonvolatile status, read owner page 3 by using the global password.

Perform a read request of owner page 3 and enter the global password when you are prompted for it. The read response should return 4 hexadecimal (hex) characters (16 bits) of data, each bit corresponding to a general page of data. If the data bit is a logic 1 for the general frame in question, then that frame is nonvolatile, and a copy is saved in electrically erasable programmable read-only memory (EEPROM). For example:

Returned data:	D400
Hex to binary conversion:	1101 0100 0000 0000
	MSB LSB

The most significant bit (MSB) corresponds to general page 0, and the least significant bit (LSB) corresponds to general page 15, so for this example pages 0,1,3, and 5 are nonvolatile.

To change the status of the nonvolatile bits, perform a write request to owner area page 3 using the global password. Depending on the host software used, the write data is entered in hexadecimal form. For example, to set only frames 3, 4, and 12 nonvolatile, the write data is entered as 1808 hex, which translates to the following binary data: 0001 1000 0000 1000.

Note: *To preserve factory configuration data and built-in-self-test diagnostic information, general pages 0 and 1 are saved in nonvolatile memory despite the bit settings in owner page 3.*

Locking Frames (Page Locks)

The page lock feature is used to designate general pages as read only. Once a page is locked, the global password must be used to unlock it before a successful write request can be processed to change the data in that frame. The page lock bits are stored in owner area page 0.

To read the page lock bits, execute a read request of owner page 0 and enter the global password when the host software prompts you for it. The read response should return 4 hex characters (16 bits) of data, in which each bit corresponds to a general page of data. If the data bit is a logic 1 for the general frame in question, then that frame is locked and is read only. For example:

Returned data:	c160
Hex to binary conversion:	1100 0001 0110 0000
	MSB LSB

The MSB corresponds to general page 0, and the LSB corresponds to general page 15. In this example pages 0,1,7,9, and 10 are locked.

To change the status of the nonvolatile bits, use the global password to perform a write request to owner area page 0. Depending on the host software used, the write data is entered in hexadecimal form. So for example, to lock frames 4 and 7, the write data is entered as 0900 hex. This translates to binary data as follows: 0000 1001 0000 0000.

Auto Beep on Acknowledge and Read Sequences

A TransCore IT2235 Tag features an auto beep function. This lets the tag emit various beep sequences after receiving either a Title 21 acknowledge (ACK) command or a Title 21 read request. A byte, called the mode bits, programmed into general page 0 controls this function. The mode bits are detailed in Table C-2.

Table C-2 Mode Bits

Field Name	Bit No.
EEPROM configuration mode — factory use only	112
Auto beep A/V code bit 4	113
Auto beep A/V code bit 3	114
Auto beep A/V code bit 2	115
Auto beep A/V code bit 1	116
Auto beep A/V code bit 0	117
Auto beep On ACK	118
Auto beep On read	119

The EEPROM configuration mode bit is for factory use only. When setting the auto beep function, always make this bit a logic 0.

All of the other bits in this byte apply to the auto beep function. The auto beep audio/visual (A/V) bits 4 through 0 can be set to emit the desired A/V sequence based on the information presented in Table C-3.

Table C-3 A/V Options Bits Decoding

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Audio Sequence
0	0	0	0	0	None
0	0	0	0	1	HI-HI-HI-HI
0	0	0	1	0	LO-LO-LO-LO
0	0	0	1	1	LO-HI
0	0	1	1	1	Long LO
0	1	0	1	1	HI-LO-HI-LO
0	1	1	1	1	LO-LO-LO
1	0	0	1	1	HI-HI-HI
1	0	1	1	1	HI-HI
1	1	0	1	1	Long HI
1	1	1	0	1	LO
1	1	1	1	0	HI
1	1	1	1	1	HI

The auto beep on ACK bit, when set to 1, enables the automatic initiation of the encoded A/V sequence determined in auto beep A/V code bits 0 to 4 after a Title 21 ACK is processed.

The auto beep on read bit, when set to 1, enables the automatic initiation of the encoded A/V sequence determined in auto beep A/V code bits 0 to 4 after a Title 21 read request has been processed, and the tag enters an idle state.

For example, if you want the tag to beep three times with a HI tone when it receives a Title 21 read, program the byte to a value of 01001101 or a hex 4D. Program general page 0 with this byte, and the following sequence must occur:

1. Read the page lock bits (owner page 0), note their values for later restoration.
2. Write to owner page 0 to change the lock bit for general page 0 to a value of 0.
3. Write to general page 0, changing only the mode bits field to the value desired (hex 4D in the example on page C-7).
4. Write to owner page 0 to change the lock bit for general page 0 to a value of 1.

Using the Custom Message Feature

The IT2235 Tag can store a custom message of up to 16 ASCII characters, and display the message on the liquid crystal display (LCD) when the user presses IT2235 Tag pushbutton. To use this capability, program the custom message into one of the general data pages (pages 2 to 15), and then send a general acknowledge (GENACK) command with a pointer indicating where the custom message is stored in tag memory.

The GENACK command has a field called the LCD message page pointer that consists of 1 byte (8 bits). If the custom message, for example, is stored in general page 5, this page pointer in the GENACK command is set to a value of 00000101 binary, or 05 hex.

The LCD controller interprets the data as ASCII data. Therefore, if a GENACK command is sent that points to a custom message page and the page has normal hex data, it still will be interpreted as ASCII and the tag will display an unintelligible message on its LCD.

To discontinue using the custom message, send the tag another GENACK command with 0s in the LCD message page pointer field.

D

Application Program Interface

Appendix D

Application Program Interface

This appendix describes the application program interface (API) for the IT2410 Tag Programmer.

Interface Definitions

Communications between the tag programmer and host computer occurs via a serial RS–232 interface. “Communications Protocols” on page D-3 describes the serial protocol. Software loaded onto the host computer or lane controller and that is used to program tags is referred to as host software in this appendix. This platform performs data transfer, commands, and responses in a binary-coded format similar to the industry standard architecture (ISA) data structure, but with error correcting protocol (ECP) additions to maintain error-free communications.

RS–232 Communications Conduit

The RS–232 conduit is configured with the defined communications parameters listed in Table D-1.

Table D-1 Communications Parameters

Parameter	Value
Data Rate	19,200 bps (factory default) or 38,400 bps
Data Bits	8
Parity	None
Stop Bits	1
Software Flow Control	None
Hardware Flow Control	None

Communications Protocols

TransCore reader systems use a programmer-to-host software protocol. The protocol information that is presented in this section provides reference information relevant to developing host software.

Each message is framed with start-of-message (som) and end-of-message (eom) characters so that the host software can detect the beginning and end of each message.

General Software Information

All tag programmer commands are preceded by a <som> ampersand character (&) followed by an <eom> percent character (%). All data after the <eom> character is ignored until the next <som> is detected.

Any & character that occurs in the message between the <som> and <eom> is converted to the backslash and *at* character (\@) sequence. Any % character is converted to the \? character sequence. All \ characters are converted to the \\ sequence. All <som> and <eom> character conversions are performed after the cyclic redundancy check (CRC) has been performed on the transmit data and before the CRC is performed on the receive data.

Programmer-to-Host Software Communications Protocol

The programmer and host software interface use the following communications protocol for command responses and asynchronous responses.

Binary-coded integer messages are contained in a protocol defined by

```
<som> <seq #> <len> <resp> [<data>] <crc> <eom>
```

where

<som> - Start of message is used to find the start of the next message following a message time-out or following the end of the previous message. All data is ignored until the <som> is detected. The start of message is defined as the ASCII character &.

<seq #> - Combination of <seq #> and <len> makes 16 bits or 2 bytes of data. The sequence number is 6 bits and the length is 10 bits. The sequence number is used for the error detection scheme. The value for sequence number may represent any number between 0 and 63 with the following restrictions:

- The sequence number for command responses matches the sequence number of the last received command.
- The sequence numbers for asynchronous messages are 0,2,4,6,...,62 and is incremented after each new asynchronous message.

The sequence number for the acknowledge (ACK)/not acknowledge (NACK) message matches the sequence number of the last received message.

<len> - Length is a group of 10 binary bits that specifies the number of bytes in the response and data fields. The value of length is 1 for ACK/NACK responses.

<resp> - Response is two bytes long to indicate the status of the command received by the programmer from the host software. Responses are described in Table D-2.

Table D-2 Command Responses and Definitions

Response	Meaning
0000H	Command complete
0001H	Command in progress
0002H	Command data invalid
0003H	Command invalid
0004H	Command aborted
3XXXH	Tag data
3000H	No-Tag Data Status Code
3001H	IT2200 Tag Data Status Code
3002H	Reserved
8XXXH	Diagnostic data
8000H	Power-Up Diagnostic Report Status Code
8001H	Background Diagnostic Report Status Code
8002H	Download Active
AXXXH	Diagnostics Statistical Data

[<data>] - Data field may be of length from 0 to 1021 bytes and is associated with each specific response. See “Command List” on page D-7, for a complete listing of tag commands.

<crc> - Cyclic redundancy check, or CRC, is 2 bytes or 16 binary bits long, denoting a CRC-XMODEM result generated on each message byte exclusive of the <som> and <eom>. The polynomial for the CRC calculation is $X^{16}+X^{12}+X^5+1$ with a feedback of 1021H for a XMODEM type CRC.

<eom> - End of message aids in finding the end of the message. All data after the <eom> is ignored until the next <som> is detected. The <eom> is defined as the ASCII character %.

Serial Data Acknowledge

The programmer and host software interface use the data acknowledge protocol as noted here. Binary-coded integer messages contained in a protocol are defined by

<som> <seq #> <len> <resp> <crc> <eom>

where

<resp> - response is 1 byte long to indicate the status of the data response received by the programmer from the host software. Table D-3 lists the responses.

Table D-3 Message Responses and Definitions

Response	Meaning
DDH	Data acknowledge (ACK)
EEH	Data invalid (NACK)

Serial Error Detection and Recovery for the Programmer

The IT2410 Tag Programmer method of error detection and recovery consists of a combination of CRC, sequence numbers, a simple structured protocol with message delimiters, a message length, and message time-outs. By using this combination of error detection and structured communications protocol, a high-level of confidence is obtained without the overhead associated with standard ECP.

These methods are described in the following sections.

Serial Message Failures

Serial data transmission failures can occur in one the three following forms:

- Modified byte
- Extra byte
- Lost byte

Modified bytes are detected by the CRC check. Extra bytes are detected by the CRC and are compensated for by the <som> and <eom>. Lost bytes require that time-outs be implemented to prevent lockups caused by anticipation of additional bytes being received as specified by the <len>. The time-outs implemented are described as follows:

- Any received <som> ... <eom> should take less than 500 milliseconds, which assumes 1500 characters at 19.2 k baud and long commands are followed by short responses.
- The asynchronous data <som> ... ACK ... <eom> should take less than 1.0 second.
- A command <som> ... response ... <eom> may take considerably longer because of command processing time.

If a message CRC or time-out error occurs during the command/response message sequence, the error reporting will occur as follows:

- For a CRC failure that occurs during the command message, the programmer reports the error to the host software using the NACK message.
- For a CRC failure or time-out failure that occurs during the response message, the host software reports the error to the programmer by retransmitting the command message.

If a message CRC or time-out error occurs during an asynchronous response/ACK message sequence, the error reporting will occur as follows:

- For a CRC failure that occurs during the asynchronous response message, the host software should report the error to the programmer using the NACK message.
- For a CRC failure or time-out failure that occurs during the ACK message, the programmer reports the error to the host software by retransmitting the asynchronous response message.

Serial Sequence Numbers

The host software maintains control of the sequence numbers for commands and their associated responses.

- A command is not complete until a response with a proper sequence number is received from the programmer.
- A response is not complete until the host software changes the command sequence number from the previous command.

If the command sequence number does not change, the response is retransmitted by the programmer.

Sequence numbers for asynchronous response communications sequences are maintained by the programmer.

- An asynchronous response transmission sequence is not complete until an ACK is received from the host software with a proper sequence number.
- An ACK is not complete until the programmer changes the data sequence number.

If the data sequence number does not change, the ACK is retransmitted by the host software.

To prevent asynchronous messages from being interpreted as command responses when host software and programmer sequence numbers pass each other, the host software sequence numbers should be odd (1,3,5,7,...,63) and programmer sequence numbers should be even (0,2,4,6,...,62).

Serial Asynchronous Responses

Tag data or diagnostics data will generate asynchronous responses, that is, responses that have no associated commands. If the host software begins sending a command during this time, the host software ignores the asynchronous response. The programmer automatically retransmits the asynchronous response following the command execution.

Command List

The command list (Table D-4) describes the command requests that are used between the programmer and the host software. Within the list, the commands are further divided according to their functionality, for example, communications and real-time clock.

Table D-4 Tag Programmer Commands

Command Type	Command
Configuring the Programmer	
Identify	0480H
Restore Programmer Configuration	057CH
Reset Programmer	057DH
Save Programmer Configuration	057EH
Get Programmer Security Characters	06ABH
Set Programmer Security Characters	056BH (key required)
Get Programmer Serial Number	06ACH
Set Programmer Serial Number	056CH (key required)
Get Programmer Control	06ADH
Set Programmer Control	056DH
Set Programmer Password	057BH
Communications Parameters for the Programmer	
Get Communications Baud Rate	06A6H
Set Communications Baud Rate	0566H
Working with Asynchronous Commands	
Get Asynchronous Transaction Response Count	06A1H
Accessing the Real-time Clock	
Get Time and Date	06A7H
Set Time and Date	0567H
Reading from and Writing to Tags	
R/W Tag Command Request	3000H
R/W Virtual Tag Command Request	3040H
Reserved/Spare[a]	
Spare	05601H
Spare	0562H
Spare	0564H-0565H
Spare	0572H-0579H
Reserved	057FH
Spare	06A2H
Spare	06A4H-06A5H
Spare	06B2H-06BEH
Reserved	06BFH

a. These commands have not been delegated a command type and are not listed in this section.

Note: *Command 3000H (host software tag command) has an extended command set that parallels the tag-reader command set for the interface between the reader and tag on the radio frequency (RF) interface.*

Identify

The Identify command requests the identification number (ID) and version numbers of the programmer application firmware, the RF module firmware, and the programmer hardware version.

Data associated with the Identify command for the programmer is shown in Table D-5.

Table D-5 Identify Command Code

Byte	Bit 7
0-7	Vendor
8-9	Hardware version
10-29	Boot software ID
30-49	Application software ID
50-69	Serial number on programmer
70-82	RF software ID
83-89	Reserved field on programmer

Vendor — The first group of eight ASCII characters provides the host software with a vendor name, *AMTECH*, followed by two spaces. The symbol ^ indicates a space or an ASCII character 20H in this document.

Hardware version — The next two bytes provide a version of the hardware. The general format of the vendor and hardware version is illustrated here:

Description	ASCII Characters
Vendor	AMTECH^^
Hardware version	XX

Boot and application software IDs — The remaining four groups provide 20-character version IDs of the boot and application software for the reader, and alternately serve as a method to read the programmer serial number. In the reader each software ID is composed of a 9-character Amtech internal part number and an 11-character software version. The general format for each of the four software IDs is illustrated here:

Description	ASCII Characters
Software ID	
Part number	XXXXX-XX^
Version	VER^X.XX^X^

Programmer serial number (S/N) — For the programmer, the first four bytes (50-53) of the software field return the programmer S/N.

Restore Programmer Configuration

The Restore Programmer Configuration command returns the programmer to its original factory default settings and requests the programmer to reconfigure its current internal settings (e.g., RF frequency, RF communications settings) to that which is saved in nonvolatile memory (NVM) or to default factory settings.

The data associated with the Restore Programmer Configuration is shown in Table D-6. This data defines the source (factory defaults or NVM) of the configuration parameters of the programmer. The nonvolatile random-access memory (NVRAM) parameters are the last saved using the Set Programmer Configuration command. Upon command, the programmer changes the programmer configuration to the values from the selected source immediately following the transmission of the command response (Table D-7).

Table D-6 Programmer Configuration Data

Bits 7 - 1	Bit 0
Spare	Programmer configuration control

Table D-7 Values for Programmer Configuration Bit

Value	Definition
0	Factory defaults
1	NVRAM

All the factory defaults for the programmer and the configuration parameters restored from NVRAM or from factory defaults are shown in Table D-8.

Table D-8 Factory Default Configuration Parameters

Parameter	Programmer Factory Defaults
RF frequency	914 MHz
RF attenuation level	0 (fixed attenuation)[a]
Communications baud rate	19,200 bps (factory default)
Check tag configuration	N/A
Tag response append data	N/A
Tag command mode	N/A
Custom auto tag command table	N/A
Custom auto user sequence table	N/A
Programmer control	Tag command time-out 10 s, buzzer enabled

a. *The programmer is designed so that the RF attenuation is fixed in the firmware to a setting of 0 dB, however, there is a fixed attenuation in the RF module to allow limited functional range of approximately 0.5 feet (0.15 m) for the built-in antenna.*

Reset Programmer

The Reset Programmer command causes all the programmer parameters to initialize to a power-up state (based on the NVM settings) and forces the internal logic card power-on diagnostics to be initiated (Table D-9).

Table D-9 Reset Data

Byte	Bit 7
0	Reset control (most significant byte)
1	Reset control (least significant byte)

A command complete response is associated with this command, followed by a 8000H asynchronous response type data code with the results of the power-on diagnostics test. All buffered tag responses will be deleted after execution of this command. Execution time is approximately 10 seconds.

Reset Control — This field contains a unique value to prevent a bit error from inadvertently causing a soft reset of the programmer. This field must contain the hexadecimal value A5A5 to invoke a reset of the programmer.

Save Programmer Configuration

The Save Programmer Configuration command causes the current configuration of the programmer to be saved to NVM. No data is associated with the Save Programmer

Configuration command. See the “Restore Programmer Configuration” on page D-10 for a listing of the parameters that are saved and the factory default values for each.

Get/Set Programmer Security Characters

Unique security characters are programmed into each customer tag programmer at the time of manufacture and are assigned and tracked by TransCore. Each security character set is linked to a programmer serial number, also assigned and managed by TransCore. The security codes are programmed at the factory prior to shipment.

The Get Programmer Security Characters command requests the programmer’s assigned security characters. Security characters are bit patterns in predetermined positions in the American Trucking Association page. The Set Programmer Security Characters command sets the programmer security characters.

The data associated with the Get/Set Programmer Security Characters command is shown in Table D-10.

Table D-10 Security Characters Data

Byte	Bits 7 - 0
0	Security character 0
1	Security character 1

Security characters — This field contains a two-byte quantity indicating that the security characters are assigned to the programmer. These security characters may be used to aid in controlling fraudulent use of any reader system. The hardware programming key is required to perform the set programmer security characters command successfully.

Valid security characters and their associated ASCII (hex) representations are listed in Table D-11.

Table D-11 Valid Security Characters

Security Character	ASCII (Hex)	Security Character	ASCII (Hex)
(space)	20	:	3A
!	21	;	3B
"	22	<	3C
#	23	=	3D
$	24	>	3E
%	25	?	3F
&	26	@	40
'	27	[	5B
(	28	\	5C
)	29	]	5D
+	2B	^	5E
,	2C	_	5F

The location of these security bits within the 128-bit page, based on automatic vehicle ID standards, including AAR and ISO, consists of two distinct 6-bit groups comprising the 106 through 111 and 112 through 117 bit positions. These 6-bit groups correspond to the 18th and 19th character locations based on the 6-bit automatic vehicle ID standards.

Security characters stored in the programmer's NVM can be set to one of two combinations. At least one of the security character positions must be programmed to customer-specific security characters from the security character set for customer programmers (see Table D-12).

Table D-12 Programmer Security Characters and Values

Programmer Security 1	Programmer Security 2	Description
Security character	Security character	Valid customer combination
Security character	Normal character	Valid customer combination
Normal character	Security character	Valid customer combination
Normal character	Normal character	Invalid combination
Space	Space	Reserved

Get/Set Programmer Serial Number

The Get Programmer Serial Number command requests the factory-programmed serial number. The Set Programmer Serial Number command sets the programmer serial number.

Programmer serial numbers are individually programmed into each programmer at the time of manufacture. Serial numbers are related to a set of security characters that have been assigned to the programmer. Serial numbers are identified in the unique released document for each customer programmer.

The data associated with the Get/Set Programmer Serial Number commands is shown in Table D-13.

Table D-13 Serial Number Data

Byte	Bit 7
0	Serial number (MSB)
1	Serial number
2	Serial number
3	Serial number (LSB)

Serial numbers are programmed by the host software command, which can only be accomplished, as with security characters, using the master key.

Serial number byte — This field contains a four-byte quantity. It indicates the serial number that is to be assigned to the programmer at the factory. The hardware programming key is required to perform the Set Programmer Serial Number command successfully.

Get/Set Programmer Control

The Get Programmer Control command requests the tag command time-out and programmer audio buzzer settings. The Set Programmer Control command enables or disables the programmer control of the tag time-out and audio buzzer.

The tag command time-out instructs the programmer to continue sending a tag command (read, write) over the RF link for a specified period of time. These values are stored in the programmer's NVM. The data associated with the Get/Set Programmer Control commands is shown in Table D-14.

Table D-14 Programmer Status Data

Byte	Bits 7 - 1	Bit 0
0	Tag command time-out	
1	Spare	Buzzer disable

Tag command time-out — This byte indicates the time-out period for tag commands in increments of 100 milliseconds. This field is a binary-coded integer with values in the range of 0 to 255. A value of 0 represents a time-out of 100 milliseconds. A value of 255 represents a time-out of 25.6 seconds. This tag command time-out determines the length of time that the programmer continues to send tag commands over the RF link after having received the tag command request from the host software. The factory default value for the tag command time-out is 10 seconds.

Buzzer disable — This bit is used to control the status of the audible indicator for the programmer. When set to 1, the buzzer functions for the programmer are disabled. The factory default for the buzzer disable is 0, or buzzer enabled.

Set Programmer Password

The Set Programmer Password command sets and verifies the user password. The user password can only be changed by resending the Set Programmer Password command within two seconds of password verification.

Note: *Any host software should prompt the user to reenter, or verify, the new password prior to sending a change to the programmer.*

These commands are applicable to the programmer only. This command is not linked with a get command, which maintains the integrity of the password.

The programmer maintains the current user password in NVM that is supplied by the host software. The programmer only responds to a limited command set until the password supplied by the host software matches the one stored in NVM. The user password is used to initiate a programming session. The data associated to the set programmer password command is provided in Table D-15.

Table D-15 Set Programmer Password Command Data

Byte	Bits 7 - 0
0	Password byte 0
1	Password byte 1
2	Password byte 2
3	Password byte 3
4	Password byte 4
5	Password byte 5

Password byte — This subfield contains a 6-byte quantity. It provides the password that is to be assigned to the programmer. All programmer passwords are set to a factory default of PROGMR at the time of manufacture. This nonvolatile parameter is capable of being changed by the user of the host software. Once changed, the new password is activated.

A backdoor password is available from the factory for customers who lose or forget their password. The backdoor password is only valid for a limited period of time. Perform an update of the programmer password after use of this backdoor password so that the current password is known. Table D-16 lists the backdoor password status data.

Table D-16 Password Status Data

Bits 7 - 0
Password status

A new programmer password may be set when you issue a Set Programmer Password command immediately following (within two seconds) the initial Set Programmer Password command request. The initial command request may contain either the valid programmer password or the backdoor password.

The response associated with the Set Programmer Password Command is shown in Table D-17. Only limited programmer commands are available until the user password is verified. Once the backdoor password has been verified, only limited programmer commands are available until the user password is changed.

Table D-17 Set Programmer Password Command Responses

Value	Description
0	Invalid password
1	Backdoor password verified
2	User password verified
3	User password changed

Programmer Command Restrictions

This section describes the interaction of commands and functions.

During manufacturing of the programmer, the security characters and serial number are set. The security characters and serial number are written to the programmer and can only be changed at the factory.

Each time the programmer is powered on it performs a self-test and awaits an acknowledgement from the host software. Only the select commands are valid until the Set Time/Date command is issued. Further restrictions apply until the programmer password is verified using the Set Programmer Password command.

The Identify, Reset Programmer, and Get/Set Time/Date commands are always available regardless of password status. The Set Programmer Password command is only available after the time and date have been set and validated. The Set Serial Number and Set Security Characters commands are available only to the factory. Limited tag commands are available until the programmer password has been verified. Virtually all commands are available once the programmer password has been verified.

Figure D-1 illustrates the precedence of these functions. The time/date is set in permanent programmer memory. Once set and saved using Save Programmer Configuration command, the time/date no longer need to be set.

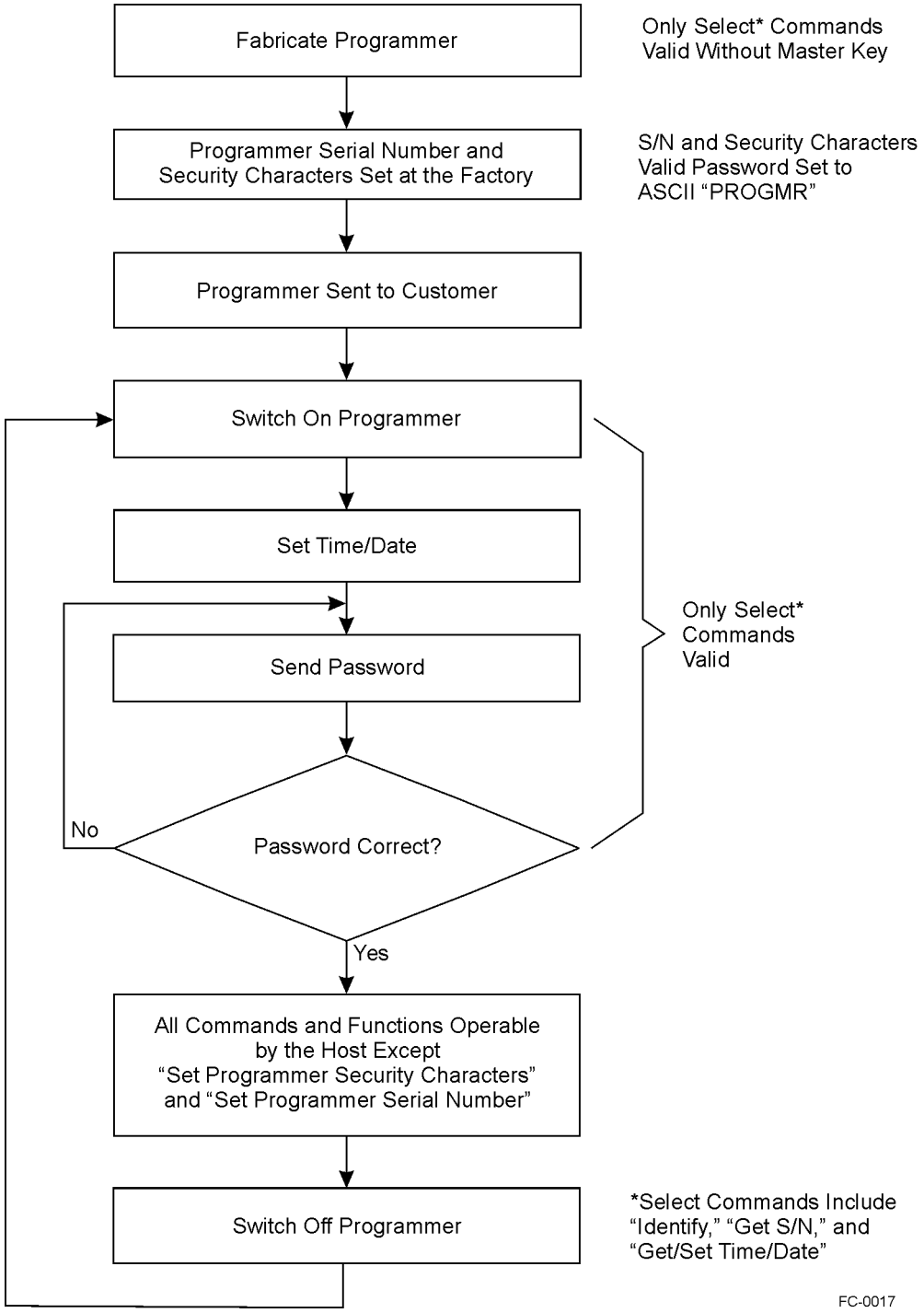

Figure D-1 Programmer Command Hierarchy

Get/Set Communications Baud Rate

The Get Communications Baud Rate command requests programmer's communications baud rate. The Set Communications Baud Rate command sets the programmer RS–232 communications baud rate.

Upon command, the programmer changes its baud rate to the selected value immediately following the transmission of the command response. It is recommended that the baud rate only be changed on the host computer after the programmer has received the acknowledge so that communications are maintained.

Note: *You might want to have a command in the host software to let the user change the baud rate when the programmer has been saved to a different baud rate than that of the host software default.*

The data associated with the Get/Set Communications Baud Rate is shown in Table D-18.

Table D-18 Communications Baud Rate Data

Bits 7 - 0
Communications baud rate

Value	Definition
0CH	19,200 bps (factory default)
0DH	38,400 bps

Get/Set Time/Date

The Get Time/Date command requests the current time and date that is set on the programmer's clock. The Set Time/Date command sets the programmer's real-time clock to the time and date specified in the request data.

The data associated with the Get/Set Time/Date defines the format of the time and date (Table D-19). You can modify and read this data at any time. This field is used with the tag read parameters to append time and date to any response to the host software.

Table D-19 Time/Date Data

Byte	Bit 7
0	Hours
1	Minutes
2	Seconds
3	Hundredths of seconds
4	Month
5	Day
6	Year

The range specifications for the individual time/date fields are shown in Table D-20

Table D-20 Time/Date Field Range Specifications

Description	Range
Hours	0-23
Minutes	0-59
Seconds	0-59
Hundredths	0-99
Month	1-12
Day	1-31
Year	0-99

Note: *The fields listed in Table D-20 are binary integers, not binary-coded data.*

Note: *Only the two least significant digits of the year are contained in the date. It is the responsibility of the application to use the year such that the year 2000 is processed correctly. The year 2000 is represented by a year value of zero.*

Asynchronous Responses

Asynchronous responses to tag commands, programmer modes, errors, and failure conditions are sent from the programmer to the host software.

Power on/Boot/Diagnostics Status Reports

The programmer status is reported asynchronously, once at powerup and when any change has occurred during background self-test processing.

If a fault message occurs, a bit representing the fault is set in the diagnostic report data. The status codes associated with the diagnostic reports are as follows:

- The command code at power-up or after the issue of a reset command for the diagnostic report status code is 8000H.
- The command code for background diagnostic report status code is 8001H.

The data associated with the power-on and background status reports for the programmer is specified in Table D-21.

Table D-21 Programmer Status Report Data

Byte	Bits 7 - 0
0	Programmer status

Programmer status — This field contains the subfields associated with the programmer as shown in Table D-22.

Table D-22 Subfields of Programmer Status

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Boot flash test	Application flash test	Reserved	RFI RAM test	Processor RAM test	Check tag test	Spare	Spare

Boot flash test — This bit indicates a failure of the programmer boot sector of the flash memory logic module when set to 1. Tests are automatically and continually run internally by the programmer.

Application flash test — This bit indicates a failure of the programmer logic module application sector flash memory when set to 1. Tests are automatically and continually run internally by the programmer.

Reserved — Reserved for future use.

RFI RAM test — This bit indicates a failure of the programmer RFI RAM logic module when set to 1. Tests are automatically and continually run internally by the programmer.

Processor RAM test — This bit indicates a failure of the programmer processor RAM logic module when set to 1. Tests are automatically and continually run internally by the programmer.

Check tag test — This bit indicates a check tag failure when set to 1.

Valid Response Tag Data

Valid response tag data is reported asynchronously when a tag is detected. The status codes associated with the tag data reports are shown in Table D-23.

Table D-23 Tag Data Report Status Codes

Code	Description
3000H	No-tag data status
3001H	IT2200 R/W tag data status
3002H	Reserved

When a tag is detected, tag data is sent to the host software. If it passes the CRC check, the tag data is assumed to not be corrupted.