THE FLASH PROGRAMMER MODEL TFP3

User's Guide
March 2015

maxim integrated™

©2015 Maxim Integrated Products, Inc.
All rights reserved.

No part of this documentation may be reproduced nor distributed in any form or by any means, graphic, electronic, or mechanical, including but not limited to photocopying, scanning, recording, taping, e-mailing, or storing in information storage and retrieval systems without the written permission of Maxim Integrated Products, Inc. (hereafter, “Maxim”). Products that are referenced in this document such as Microsoft Windows® may be trademarks and/or registered trademarks of their respective owners. Maxim makes no claim to these trademarks. While every precaution has been taken in the preparation of this document, individually, as a series, in whole, or in part, Maxim, the publisher, and the author assume no responsibility for errors or omissions, including any damages resulting from the express or implied application of information contained in this document or from the use of products, services, or programs that may accompany it. In no event shall Maxim, publishers, authors, or editors of this guide be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Rev. 2.2, March 2015

CONTENTS

1 Introduction 5

1.1 TFP3 Overview 5
1.2 TFP3 Features 7
1.3 Safety and ESD Notes 7
1.4 TFP3 Kit Contents 7
1.5 System Requirements 8

2 Getting Started 9

2.1 Connections to the Host and Setup 9
2.2 TFP3 Connection to the Host 10
2.3 Using the TFP3 in Host Mode 11
2.3.1 Procedure to Communicate with the TFP3 Device via TFP3.EXE 11
2.3.2 Procedure to Communicate with the TFP3 Device via TFP3GUI.EXE 12
2.4 TFP3 Secure Model Initialization 13
2.5 TFP3 Non-Secure Model Initialization 13

3 Features 14

3.1 Programming the TFP3 Internal Flash Memory 14
3.1.1 Steps for Programming the TFP3 Internal Flash Memory 14
3.2 Stand-Alone/Pushbutton Programming 15
3.2.1 Steps for Performing Pushbutton Programming 15
3.3 ICE Port 15
3.4 ATE Programming 17
3.4.1 Steps for ATE Programming 19
3.4.2 Flash Sizes of Energy Metering Microcontrollers 20
3.4.3 ATE Programming Sequence 21
3.5 TFP3 Diagnosis Information Access 22
3.5.1 Programming Counts 22
3.5.2 Total Pass Counts 22
3.5.3 Total Fail Counts 22
3.6 Secured Dumping of Target Code to Host 22
3.7 TFP3 GUI Operation 23
3.8 Generation and Loading of a Package File for TFP3 Secure Models 28
3.9 TFP3 Parameter Preservation 30
3.10 TFP3 Firmware Upgrade Using In-Application Programming 31
3.11 Loading of DUT Hex File to the TFP3 33

4 Status Indications 34

4.1 LED and Buzzer Indication 34

5 Supported Commands 35

5.1 Commands Common to Secure and Non-Secure Models of the TFP3 37
5.2 Commands Supported Only by the Secure Model of the TFP3 39
5.3 Commands Supported Only by the Non-Secure Model of the TFP3 41

6 Hardware Specifications 42
7 Ordering Information 43
8 Glossary of Terms and Abbreviations 44

TABLES

Table 1: TFP3 Versions 5
Table 2: ICE Connector to JTAG Cable (2x5) for TFP3Q-MAXQ30 and TFP3L-MAXQ30 16
Table 3: ICE Connector to ICE Cable (7x1) 16
Table 4: ATE Connector Pin Descriptions 19
Table 5: Maximum Flash Sizes of Different Energy Metering Microcontrollers 20
Table 6: Parameter Data Alignment 31
Table 7: Status LED and Buzzer Indications 34
Table 8: TFP3 Supported Commands 35
Table 9: Commands Common to Both Secure and Non-Secure Models 37
Table 10: Commands Supported Only by the Secure Model 39
Table 11: Commands Supported Only by the Non-Secure Model 41
Table 12: Ordering Numbers 43

FIGURES

Figure 1: Secure TFP3 8051 Device Model Number TFP3-8051 6
Figure 2: Secure TFP3 MAXQ30 Device Model Number TFP3Q-MAXQ30 6
Figure 3: Non-Secure TFP3 MAXQ30 Device Model Number TFP3L-MAXQ30 6
Figure 4: TFP3 Typical Connection Diagram 9
Figure 5: Device Manager 10
Figure 6: COM Port Enumeration 10
Figure 7: Navigating the Path to the TFP3.EXE 11
Figure 8: Executing TFP3 Secure Model Commands Using the TFP3.EXE 11
Figure 9: Executing TFP3 Non-Secure Model Commands Using the TFP3.EXE 12
Figure 10: ATE Connector Pin Locations 18
Figure 11: ATE Programming Sequence Flowchart 21
Figure 12: Flowchart for Execution of Load Package File Command on Target Side 29
Figure 13: TFP3 Memory Partition Layout 32
Figure 14: TFP3 IAP Programming Sequence and Code Execution 33

1 Introduction

The TFP3 (The Flash Programmer 3) is a multipurpose programmer from Maxim Integrated used to perform flash utility operations on Maxim energy metering SOCs. The available versions of the TFP3 are listed in *Table 1*.

Table 1: TFP3 Versions

ID	Function	Ordering Part Number
TFP3-8051	Flash Programmer for 8051-based Metering Devices. With Security (1)	80515-FPBM-TFP3
TFP3L-MAXQ30	Flash Programmer for ZON™ Family Metering Devices (MAXQ®-based). No Security	MAXQ30-FPBM-TFP3L#
TFP3Q-MAXQ30	Flash Programmer for ZON™ Family Metering Devices (MAXQ-based). With Security (1)	MAXQ30-FPBM-TFP3Q#

(1) With the security feature, the programming content is stored, programmed, and verified using AES 128-bit security.

This user's guide documents how to connect, set up, initialize, and perform flash utility operations and firmware upgrades using the TFP3. It also provides a detailed explanation of supported commands and insight into the various features of the TFP3 models with and without security features.

1.1 TFP3 Overview

The TFP3 is used to perform flash utility operations on Maxim Integrated's energy metering SoCs. The TFP3 has three operating modes:

- stand-alone
- host
- ATE

In the Stand-Alone mode, the TFP3 is used to program and verify the DUT. By pressing the pushbutton switch, the DUT is programmed with and verified against the preloaded flash programming file.

The Host mode is used to send and receive commands to/from the TFP3 using a GUI or CLI application, running on a Windows PC, via USB interface. In addition to performing flash utility operations, Host mode is also used to get diagnostic information and upgrade the programming file on the TFP3.

ATE mode is used for factory automation using the TFP3. Using this mode, the TFP3 can program and verify without using any host applications. The external ATE hardware, connected to the TFP3, drives the ATE signals to perform a program and verify operation of the DUT firmware.

TFP3 has internal flash memory, which can store a maximum 512KB file size of DUT firmware. The secure models of TFP3 also provide a high level of security to the DUT firmware by supporting features like AES key matching and data encryption/decryption using AES hardware. A USB interface is used to power up the TFP3 device and for communication with the host. The TFP3 can power the DUT at 3.3VDC with up to 300mA.

MAXQ is a registered trademark and ZON is a trademark of Maxim Integrated Products, Inc.

There are two variants of TFP3 devices available in the market:

- The TFP3 secure model is used for performing flash utility operations securely via a hardware AES encryption in ECB mode and decryption mechanism to protect the DUT's IP and also to restrict access to the device to authorized users.
- The TFP3 non-secure model does not use any security mechanism and allows the user much easier access to download and flash utility operations.



Figure 1: Secure TFP3 8051 Device Model Number TFP3-8051

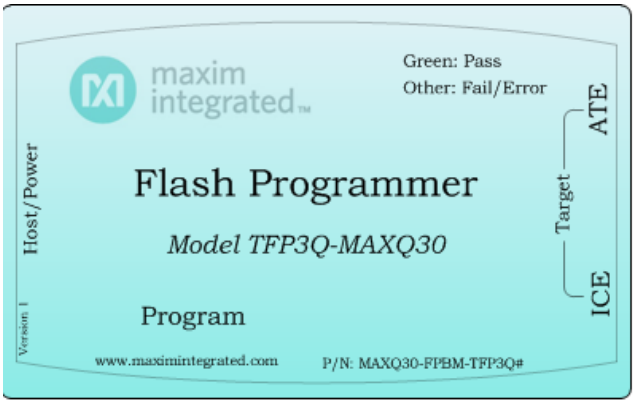


Figure 2: Secure TFP3 MAXQ30 Device Model Number TFP3Q-MAXQ30



Figure 3: Non-Secure TFP3 MAXQ30 Device Model Number TFP3L-MAXQ30

1.2 TFP3 Features

The TFP3 is a multipurpose flash programming device with the features listed below. Each feature is explained in detail in the following chapters.

- Programming of the TFP3 internal flash memory
- Pushbutton programming of the DUT via CC51 or JTAG Interface
- Host application-based programming of DUT via CC51 or JTAG Interface
- ATE-based programming of DUT via CC51 or JTAG Interface
- TFP3 diagnostic info access
- Secure dumping of the DUT code image to the host
- TFP3 GUI operation
- Generation and loading of the package file
- TFP3 parameter preservation
- Downloading of the DUT hex file to the TFP3
- TFP3 firmware upgrade using in-application programming

1.3 Safety and ESD Notes

Standard ESD handling precautions should be employed whenever handling electronic equipment. The TFP3 Flash Programmer utilizes ESD protection devices on its cable interfaces. Potential equipment damage and/or malfunction are possible if work-surface grounding procedures are not incorporated.



TFP3 ESD protection devices do not protect the target's hardware (DUT). Also, if the TFP3 device is connected to a target while the target is being powered using AC mains, then the ICE port and the ATE port are now high voltage. Correct handling procedures and proper work area grounding minimizes damage to all equipment!

1.4 TFP3 Kit Contents

- TFP3 Device
- FC-10 cable for CC51 or JTAG connection, 20-25cm in length
- USB A-to-B cable to connect the TFP3 to the host (PC) or to a USB 5V DC adapter (not provided with the kit)
- CD-ROM containing:
  - TFP3 Windows installation executable (TFP3 XX-YY.exe) with Microsoft .NET 4.0 packaged for TFP3 GUI and CLI application software
  - TFP3 Host USB CDC driver
  - TFP3 User's Guide
  - TFP3 Quick Start Guide

1.5 System Requirements

The TFP3 operating in host mode requires a Windows PC with the following features:

- 1 GHz processor and 1 GB RAM
- Minimum 1024 x 768 video display resolution
- Available USB port
- Microsoft Windows® 7 or Windows XP

  Windows is a registered trademark of Microsoft Corp.

2 Getting Started

This section provides details on connections, setup, and initialization procedures of the TFP3 device.

Note: In the following sections, software-related items are identified by bolding. **Text in bold** refers to items directly from the EV Kit software. **Text in bold and underlined** refers to items from the Microsoft Windows operating system.

2.1 Connections to the Host and Setup

The TFP3 connects to a host PC with a USB A-to-B cable, and the cable provides both data and power (no external supplies are needed). *Figure 4* shows the basic connection diagrams for the TFP3 for programming ZON (MAXQ30-based) and 71M65xx (8051-based) parts. It is possible to operate the TFP3 in both stand-alone (only target and TFP3) and PC-connected configurations. For stand-alone configurations, the TFP3 has to be powered via the USB connector.

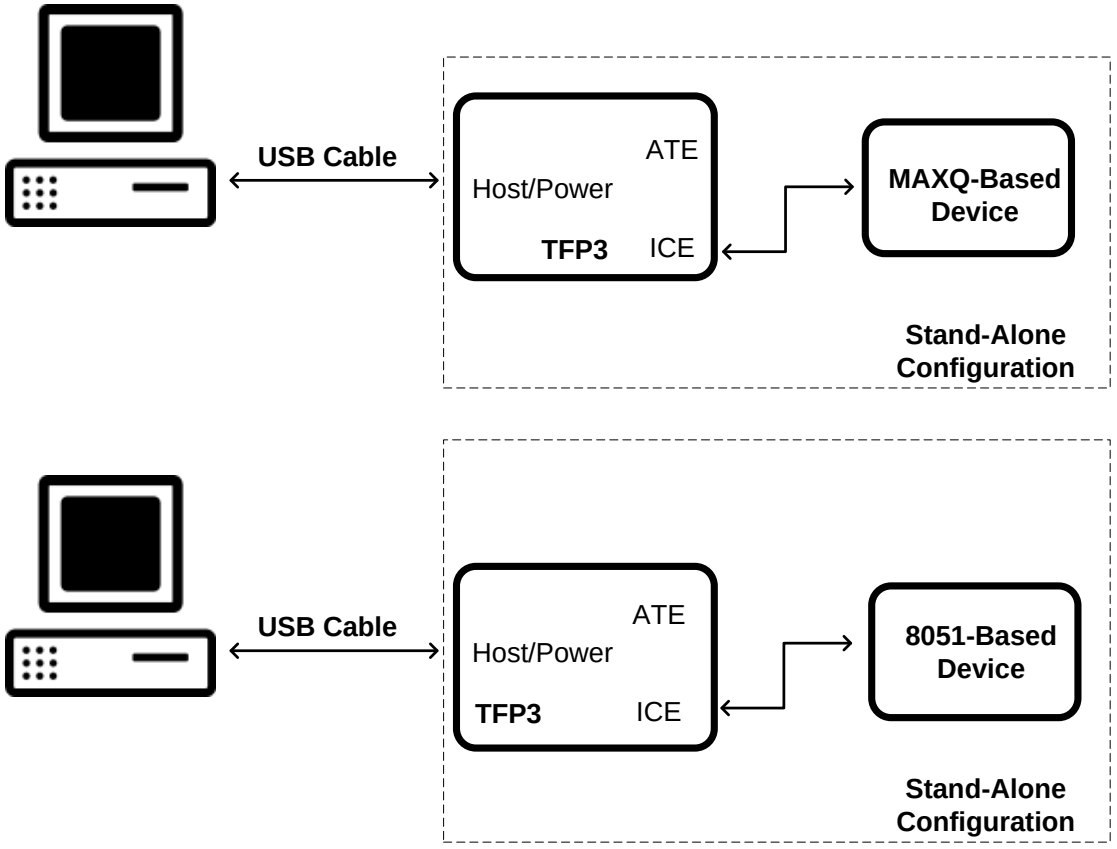


Figure 4: TFP3 Typical Connection Diagram

2.2 TFP3 Connection to the Host

The following steps are required to connect the TFP3 to the host successfully. Follow these steps before starting any communication with the TFP3 from the host.

1. Connect USB A-to-B cable between the host and the TFP3 device.
2. When the TFP3 is connected to the host, the status LED should be red first and turn to green continuously for the TFP3-8051 model. For other models of the TFP3, the LED is continuously green when connected to the host.
3. When the notification window with the text, "Installing device driver software," appears on the PC screen, go to **Start** > **Control Panel** > **Device Manager** > **Other Devices.** Right-click on "**MAXQ CDC-ACM Demo**," and update the driver software by pointing to the "**Oem54.inf**" file on the CD-ROM included in the TFP3 Programmer's Kit.

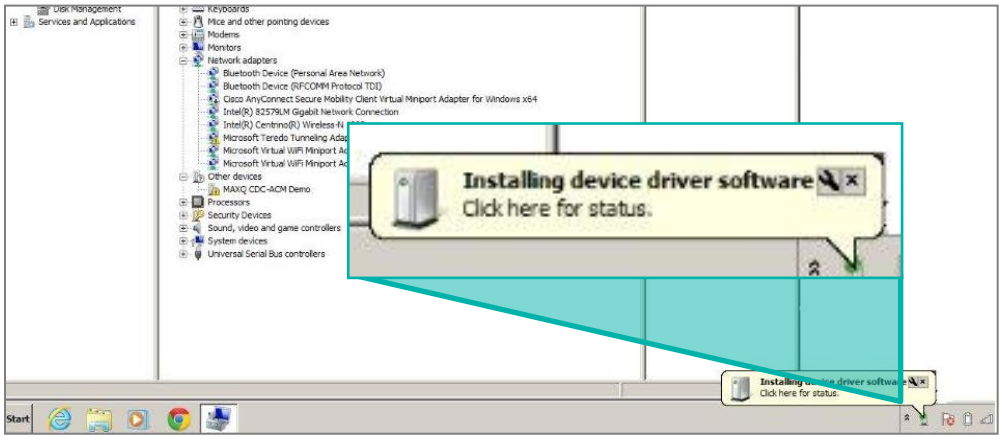


Figure 5: Device Manager

4. When the TFP3 Host driver software installation is successful, the TFP3 device is enumerated as a COM port, as shown in *Figure 6*.

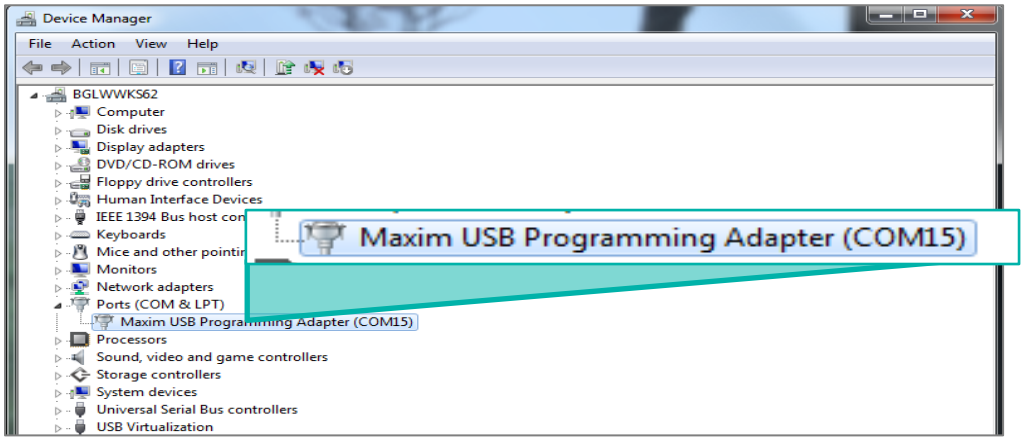


Figure 6: COM Port Enumeration

2.3 Using the TFP3 in Host Mode

The Host mode is used to get diagnostic information and upgrade the programming file on the TFP3. Once the TFP3 device is connected, powered on, and enumerated as a COM port, the user can start sending commands and receiving responses from the TFP3 in host mode using a command-line interface utility (TFP3.exe) or GUI application.

2.3.1 Procedure to Communicate with the TFP3 Device via TFP3.EXE

1. Open the Windows command prompt on the PC (**Start** > **Run** > **cmd**).
2. Enter the command **cd xxxx**, where "xxxx" is the path to where the TFP3.exe file is saved.
3. Execute the required commands using the TFP3.exe console.

The steps in the figures below detail the commands to get the version of the TFP3, set model type, set flash size of the DUT, and master erase the flash of the DUT.

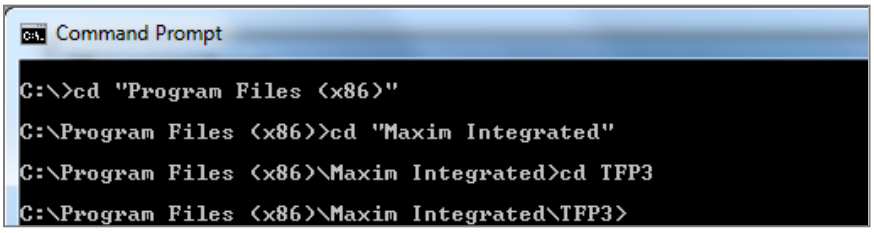


Figure 7: Navigating the Path to the TFP3.EXE

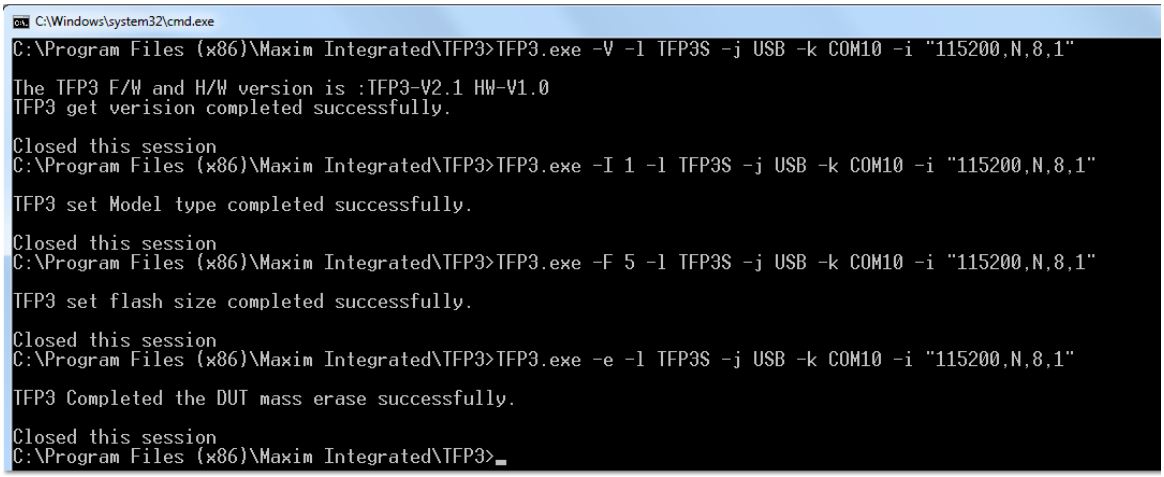


Figure 8: Executing TFP3 Secure Model Commands Using the TFP3.EXE

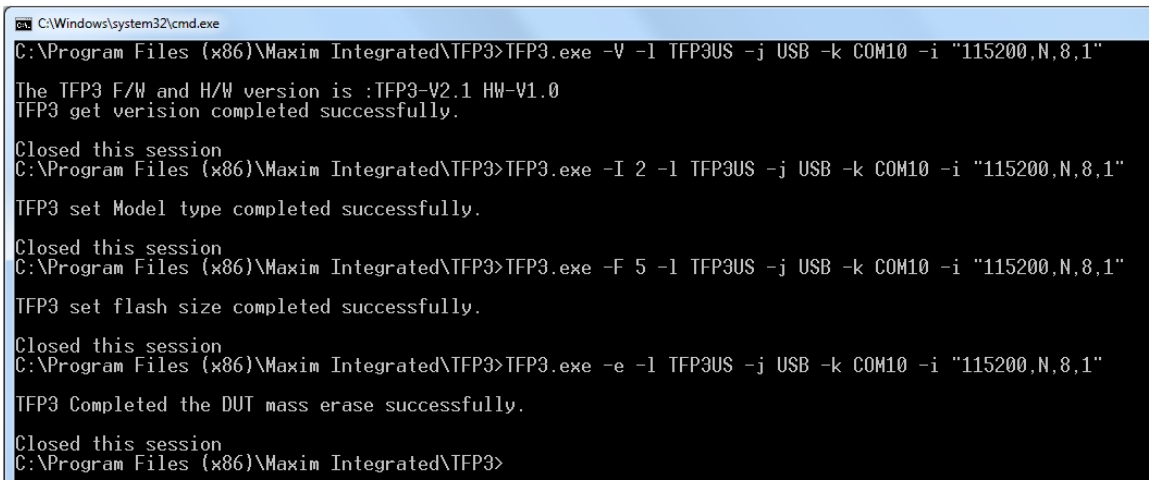


Figure 9: Executing TFP3 Non-Secure Model Commands Using the TFP3.EXE

For details of the commands supported by the TFP3, see ***Chapter 5: Supported Commands*** or the sample commands document provided in the TFP3 GUI installation folder for examples of usage with the host application.



TFP3 Host Mode communication should not be performed until the TFP3 enumerates as a COM port.

2.3.2 Procedure to Communicate with the TFP3 Device via TFP3GUI.EXE

1. Open the TFP3 GUI on the PC by clicking on **Start** > **All Programs** > **Maxim Integrated** > **TFP3** > **TFP3GUI.EXE**.
2. Select the Model type through which the DUT communication is achieved from **Select Model Type** drop-down menu.
3. Select the DUT flash size connected to the TFP3 from **Select DUT Flash Size** drop-down menu.
4. The TFP3 GUI will auto-detect the TFP3 device and connect to it.

   To manually configure and connect to the TFP3:

   a. On the **Options** menu, select **Configure serial port**, and choose the correct serial port number.

   b. On the **Target** menu, select **Connect to TFP3 device** to connect with the TFP3 device from the Host.
5. Send the required commands to the TFP3 device by selecting from the four command groups listed under the **File** menu.

For more details on using the TFP3 GUI, click on the application's **Help** menu, and select **Contents and Index**.

2.4 TFP3 Secure Model Initialization

The TFP3 needs to be initialized with the following commands before performing any other operations. This sequence is necessary to ensure that the DUT firmware is securely updated or verified.

1. Execute the **Modify AES keys** command (**K**) to modify the AES 128-bit keys in the TFP3.
2. Execute the **Select Model Type** command (**I**) to identify which type of TFP3 is used for DUT communication.
3. Execute the **Set Flash Size** command (**F**) to identify which type of target is connected to TFP3.
4. *(Optional)* Execute the **Set DUT Serial Number Count** command (**K**) to set the unique serial number of the DUT connected to TFP3. This command is required to send to the TFP3 device only if the parameter preservation feature is required by the user.
5. Execute the **Set TFP3 JTAG Clock** command (**J**) to set the clock by which the DUT communicates with the TFP3. Supported by the TFP3Q-MAXQ30 Model.

2.5 TFP3 Non-Secure Model Initialization

The TFP3 non-secure model needs to be initialized with the following commands before performing any other operations. This sequence is necessary to ensure that the DUT firmware is updated or verified properly.

1. Execute the **Select Model type** command (**I**) to identify which type of TFP3 is used for DUT communication.
2. Execute the **Set Flash Size** command (**F**) to identify which type of target is connected to the TFP3.
3. Execute the **Set TFP3 JTAG Clock** command (**J**) to set the clock by which the DUT communicates with the TFP3. Supported by the TFP3L-MAXQ30 Model.
4. Execute the **Set TFP3 Programming Count** command (**m**) to set the maximum number of times the TFP3 can perform the program and verify operation.

3 Features

3.1 Programming the TFP3 Internal Flash Memory

The TFP3 Programmer allows programming devices with a flash size up to 512KB. The DUT firmware is downloaded to the TFP3 using the USB interface and host application software. On the TFP3 secure model, the DUT firmware is AES ECB encrypted before storing it in the TFP3 internal memory. This ensures that the DUT firmware is protected from unauthorized access.



Setting the wrong model type and flash size values in the TFP3 will result in incorrect data being programmed in the DUT.

3.1.1 Steps for Programming the TFP3 Internal Flash Memory

1. Execute **Set Model Type** command (**I**) to set the model type of the TFP3 connected to the DUT.
2. Execute **Set Flash Size** command (**F**) to set the flash size of the connected DUT.
3. Execute **Generate Package File** command (**g**) to create a package file for the TFP3 Secure model.
4. Execute **Load Package File** command (**H**) to send it to the TFP3 device (TFP3 secure model).

   For the TFP3 non-secure model, execute the **Load Hex File** command (**H**) to send the DUT hex file to the TFP3 device.

For details of the commands supported by the TFP3, see *Chapter 5: Supported Commands* or the sample commands document provided in the TFP3 GUI installation folder for examples of commands that can be used with the host application.

3.2 Stand-Alone/Pushbutton Programming

This mode is used to perform only the DUT program and verify operation. This mode does not use host application software. A pushbutton switch on the TFP3 front panel is used to start this operation. The TFP3 indicates the USB enumeration failed by toggling the status LED between green and red and sounding a buzzer five times (300ms apart).

3.2.1 Steps for Performing Pushbutton Programming

1. Connect the DUT to the TFP3 using an FC-10 Flat interface cable.
2. Connect the TFP3 to the DC adapter using a USB A-to-B cable. This step is required to power on the TFP3.
3. The status LED will toggle from red to green and the buzzer will sound five times (300ms apart) to indicate that the TFP3 USB enumeration failed.
4. Press the red pushbutton switch for 300ms, and then release it to start the program and verify operation of the DUT firmware.

The status LED will toggle from red to green until the program and verify operation is complete.

If the TFP3 completes the program and verify operation successfully, the status LED turns green to indicate the programming and verification of the DUT flash memory is successful.

If the TFP3 does not complete the program and verify operation successfully, the status LED remains red and the buzzer will sound 3 times to indicate the failure.

3.3 ICE Port

The ICE port is used for DUT programming. The standard CC51 (TFP3-8051) and JTAG interface (TFP3L-MAXQ30, TFP3Q-MAXQ30) define the ICE port signals function TFP3.

Note: The cable port connected to the TFP3 device has pin 1 on the top left as shown above. Also, there is a notch on this connector, so it can only plug into the TFP3 device in one direction.

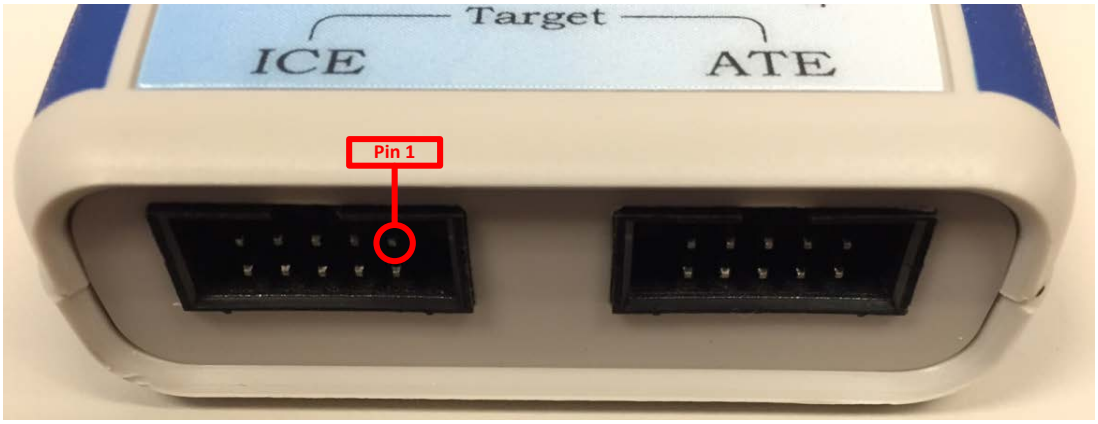


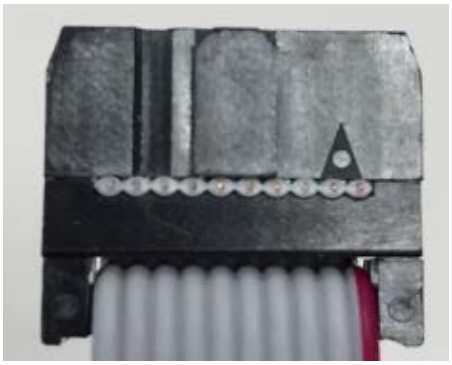
Table 2 and *Table* 3 describe the pin assignment on the cable port connected to the DUT for the various TFP3 models.

Note: The side of the connector where Pin 1 is located is marked with a triangle.

Table 2: ICE Connector to JTAG Cable (2x5) for TFP3Q-MAXQ30 and TFP3L-MAXQ30

Pin Number	Signal	Function
1	ICE_E	Enables the programming interface when high
2	GND	Ground, return
3	E_RST	Emulator reset
4	VCC	Supply Power (+3.3VDC at 300mA, max)
5	E_TCLK	Emulator clock
6	TMUX	Optional signal. Not required for programming.
7	N/C	Not connected
8	N/C	Not Connected. (+5VDC for TFP3L-MAXQ30)
9	E_RXTX	Emulator data (RX and TX), bidirectional
10	GND	Ground, return

Note: If pin 4 in *Table 2* is used to power the target, then a 1000µF/10V capacitor should be placed in between VCC and GND of the DUT board in order to prevent the TFP3 device from resetting due to the inrush current.



To ZON Device (2x5 cable)

Table 3: ICE Connector to ICE Cable (7x1)

Pin Number	Signal	Function
1	TMUX	Optional Signal
2	ICE_E	Enables the programming interface when high
3	GND – RETURN	Ground, return
4	E_RST	Emulator reset
5	E_TCLK	Emulator clock
6	E_RXTX	Emulator data (RX and TX), bidirectional
7	VCC	Supply power (+3.3VDC)

Note: Even though the TFP3 can supply power to the target, many target boards (e.g. 71M653x, 71M652x) assign V3P3D to the pin corresponding to pin 7 in *Table 3*. This means that the target boards still need to be powered with a separate DC supply. If pin 7 is used to power the target, then a 1000µF/10V capacitor should be placed in between VCC and GND of the target board in order to prevent the TFP3 device from resetting due to the inrush current.



To 8051 Device (7x1 cable)

3.4 ATE Programming

This mode is used to automate the programming or verification of DUT firmware using the TFP3 and external ATE hardware. ATE hardware drives the ATE signals as per TFP3 requirements to perform the program and verification operation. Other operations are not supported in this mode. This section describes the ATE programming setup and software algorithm to be implemented by the ATE. *Figure 10* and *Table 4* detail the ATE interface signals and descriptions.

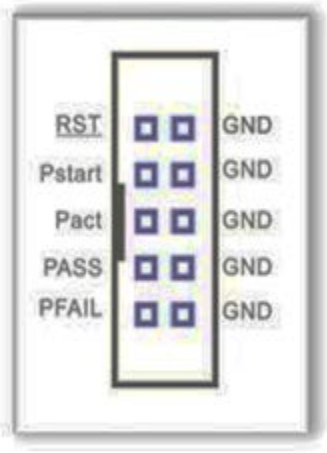


Figure 10: ATE Connector Pin Locations



Table 4: ATE Connector Pin Descriptions

ATE Connector Pin	Pin Function	TFP3 I/O	Description
1	ATE Reset	Input	Active Low Reset to TFP3 Hardware
2	GND	–	Ground
3	ATE Program Start	Input	Active High Signal Starts Target Flash Memory Programming. A minimum of 500ms pulse is required.
4	GND	–	Ground
5	ATE Program Active	Output	Active High Signal Indicates Target Flash Memory Programming in Progress. De-assert ATE program start upon ATE Program Active going high.
6	GND	–	Ground
7	ATE Pass	Output	ATE Pass Active High Signal Indicates Target Flash Memory Programmed and verified Target Microcontroller to be Correct. ATE Pass goes low when ATE Program Start reasserted high.
8	GND	–	Ground
9	ATE Fail	Output	ATE Fail Active High Signal Indicates Target Flash Memory Failed Programming and Verification of Target Microcontroller. ATE Fail goes low when ATE Program Start is reasserted high.
10	GND	–	Ground

3.4.1 Steps for ATE Programming

1. Connect external ATE hardware to the TFP3 ATE connector using a custom cable.
2. Power ON the external ATE hardware.
3. Connect the TFP3 to the DC adapter using a USB A-to-B cable. This step is required to power on the TFP3 and DUT.
4. The status LED will toggle from red to green and the buzzer will sound five times (300ms apart) to indicate that the TFP3 USB enumeration failed.

The TFP3 is now ready to perform ATE communication. The user can drive the ATE signals using any external ATE equipment in a factory production environment.

3.4.2 Flash Sizes of Energy Metering Microcontrollers

Table 5: Maximum Flash Sizes of Different Energy Metering Microcontrollers

Microcontroller	Max Flash Size (KB)
71M6513/6513H	64
71M6521BE, 71M6521DE, 71M6521FE	8, 16, 32
71M6531, 71M6531E	128, 256
71M6533D/6533DH, 71M6533F/6533FH	128, 256
71M6534	128
71M6534H	256
71M6545, 71M6545H	32, 64
71M6542F/H	64
71M6543G/GH	128
71M6542F	64
71M6541D, 71M6541F	32, 64
71M6532D, 71M6532F	128, 256
71M6534, 71M6534H	128, 256
MAX71313L, MAX71314L	64, 128
MAX71314, MAX71315	128, 256
MAX71334L, MAX71335L	128, 256
MAX71335S, MAX71336S	256, 512

3.4.3 ATE Programming Sequence

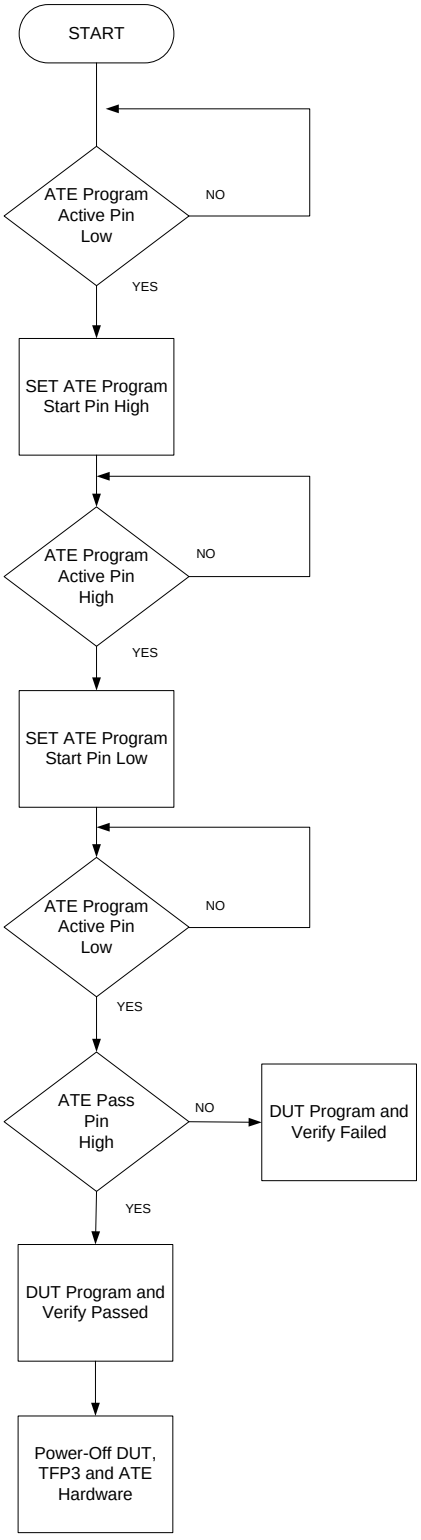


Figure 11: ATE Programming Sequence Flowchart

3.5 TFP3 Diagnosis Information Access

This feature is used to configure the TFP3 to restrict program and verify operations to a predefined number. This configuration can be set or reset by user commands. The TFP3 maintains the following parameters to keep the count of program and verify success/failed operations.

TFP3 provides the following diagnostic information:

1. Programming count
2. Total Pass count
3. Total Fail count

The default value of the Programming count is 1000000 (1 million). The Total Pass and Total Fail count default values are 0x00.

3.5.1 Programming Counts

The Programming count is a 4-byte value stored in the TFP3 EEPROM. The user has the flexibility to set this count value anywhere from 0 to 1000000 (1 million) on the TFP3 device. Once set, this count gets decremented by a value of 1 for every successful program and verify operation.

The Programming count value in the TFP3 decrements down to value 0. Once it reaches zero, the TFP3 device will not allow the user to perform any program and verify operation.

3.5.2 Total Pass Counts

The Total Pass count is a 4-byte value stored in the TFP3 EEPROM. The default value of this program count is set to 0x00. Whenever a program and verify operation is executed successfully by the TFP3 device, the Total Pass count value is incremented by 1.

3.5.3 Total Fail Counts

Total Fail count is a 4-byte value stored in the TFP3 EEPROM. The default value of this program count is set to 0x00. Whenever a program and verify operation is executed unsuccessfully by the TFP3 device, the Total Fail count value is incremented by 1.

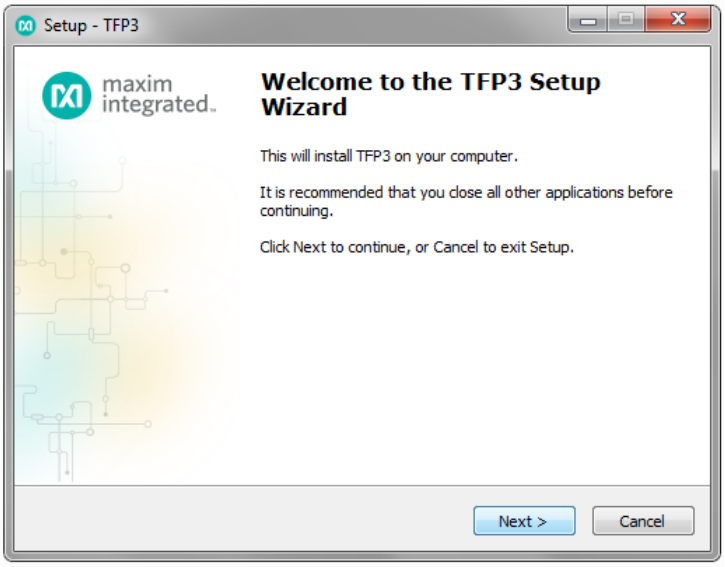
3.6 Secured Dumping of Target Code to Host

This feature is only available in TFP3 Secure model, which is used to dump DUT firmware securely onto the host. This is achieved through the **Dump DUT Code** command (**D**). The TFP3 device sends the encrypted DUT code to the host in response to the command. The host runs the software AES 128-bit algorithm and decrypts it using the AES key provided during command execution, and then stores the data in a hex file format.

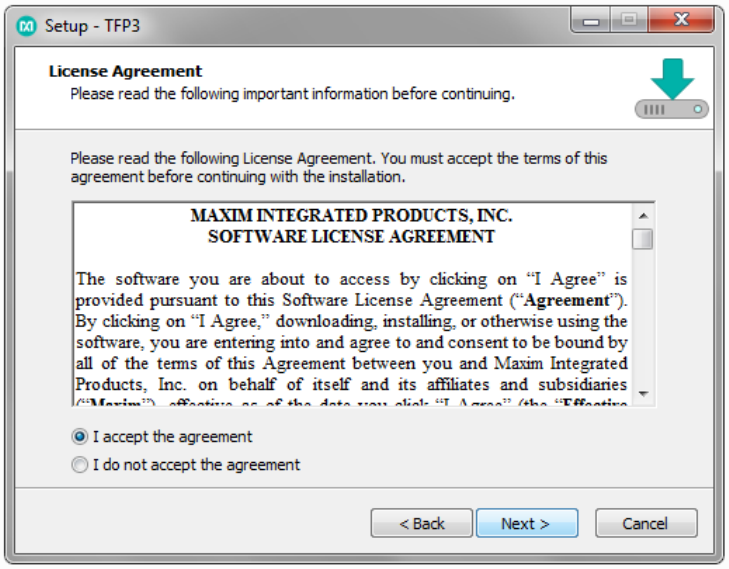
3.7 TFP3 GUI Operation

After the TFP3 is connected, powered on, and enumerated as a COM port, the user can start sending commands and receive responses from the TFP3 in host mode using the TFP3 GUI application. Listed below are the steps to install and start communicating with the TFP3 device using the TFP3 GUI application.

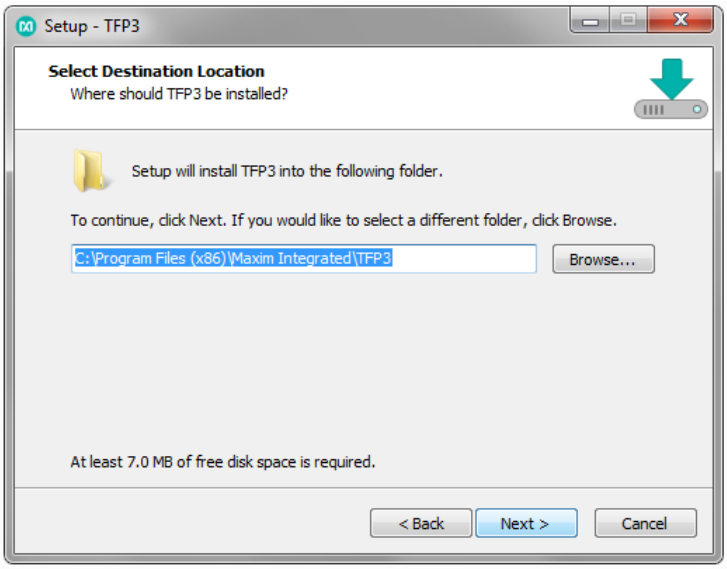
1. Run the TFP3 installation executable, **TFP3 XX-YY.EXE**, to start the installation process. (XX is the major version, and YY is the minor version.)



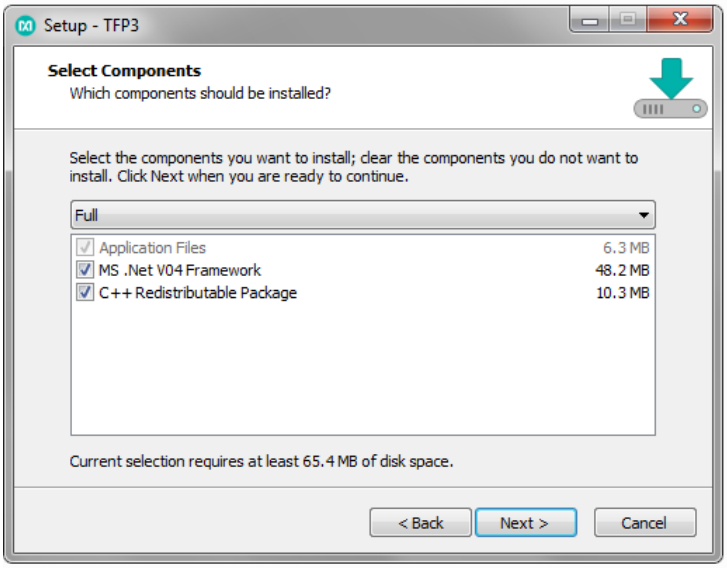
2. Accept the license agreement of TFP3 GUI to proceed with the installation process.



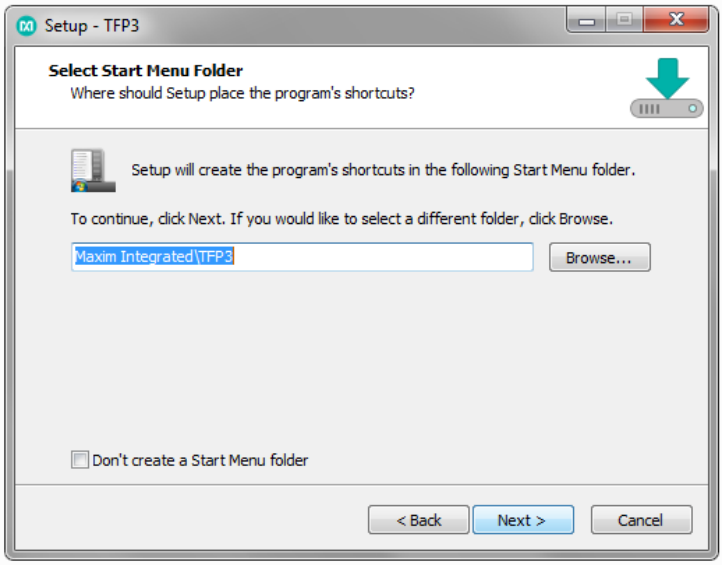
3. Provide the path to the directory in which to install the TFP3 GUI.



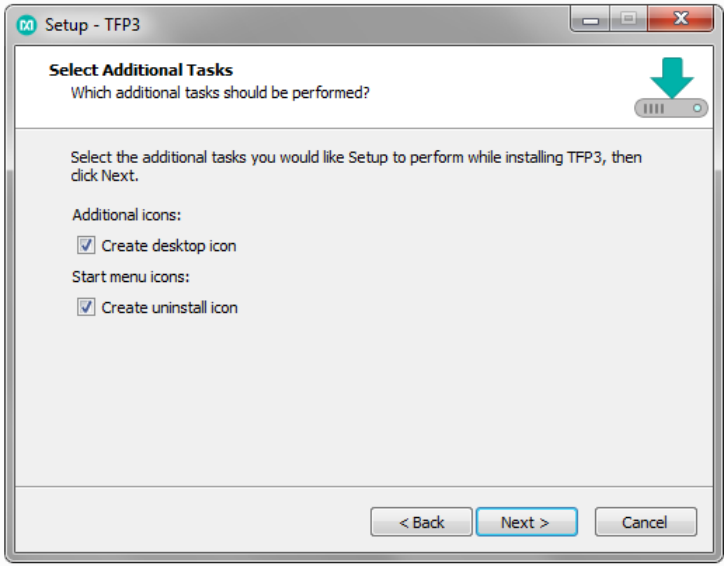
4. Select the components to install.



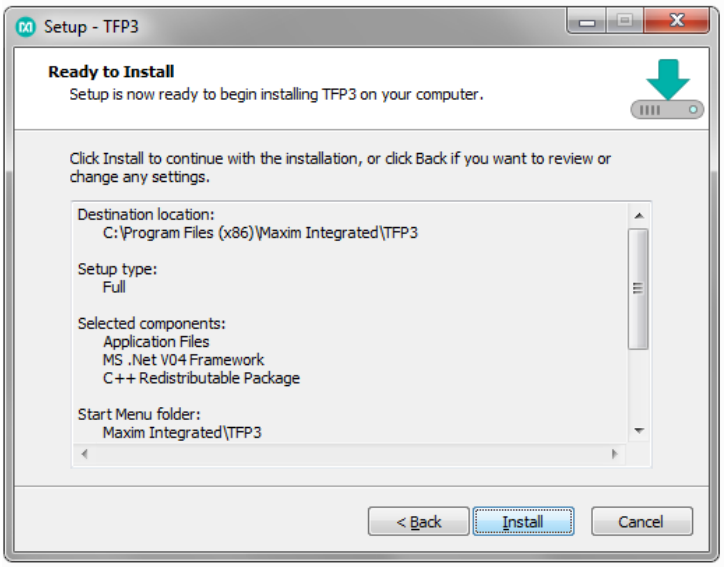
5. Provide the directory path for the Start Menu folder.



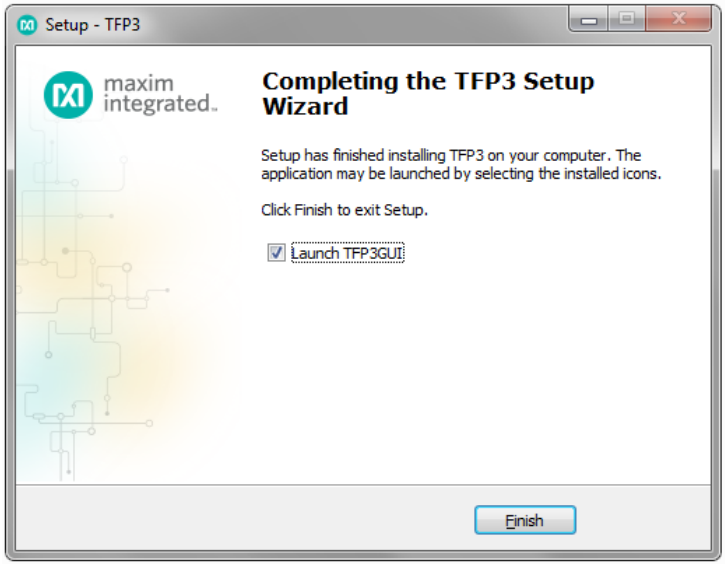
6. Select additional options.



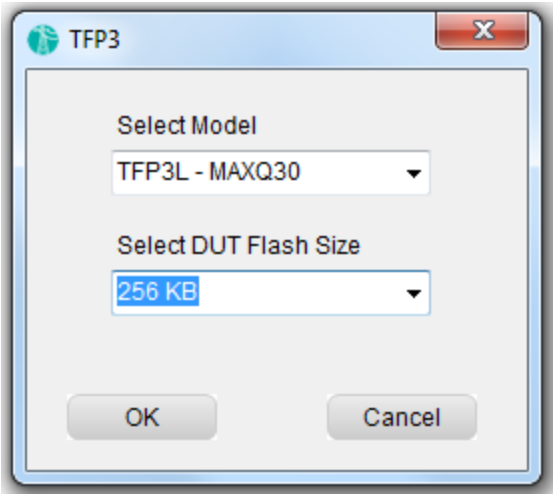
7. Click **Install** to start the installation process.



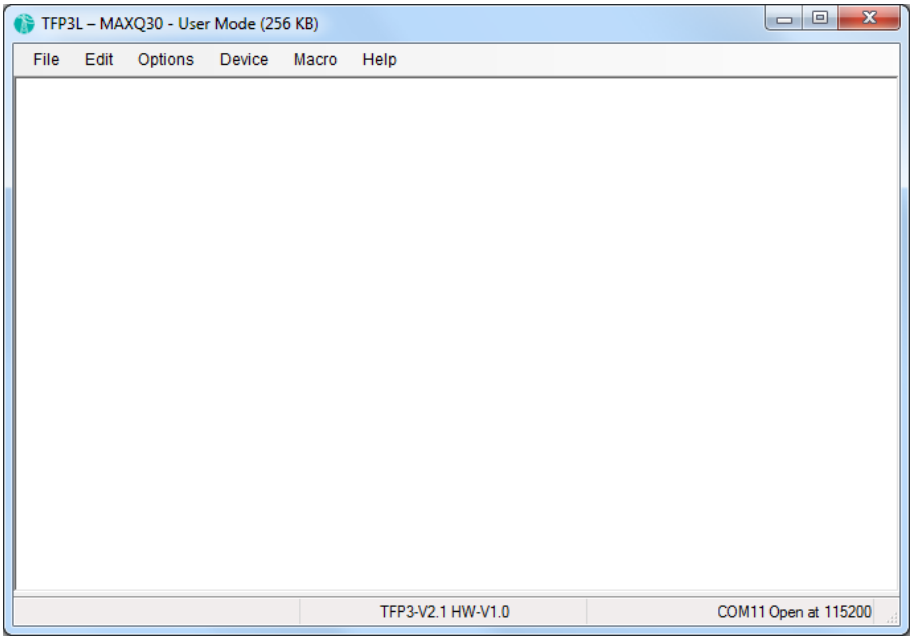
8. After the installation is complete, click **Finish** to launch the TFP3 GUI application.



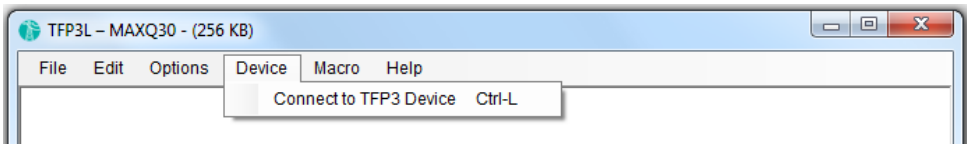
9. In the TPF3 GUI application, select the TFP3 Model type and the flash size of the DUT connected to the TFP3 before starting communication with the DUT.



10. The TFP3 GUI will auto-detect the COM port of the TFP3 device and open a session.



If auto-detection fails, manually connect to the TFP3 device by clicking on the **Device** menu, and selecting **Connect to TFP3 Device**.



For more information about using the TFP3 GUI and how to send commands to TFP3, go to the Help documentation in the **Help** menu of the application.

Review "Getting Started Help Documentation" found under the Help menu before performing any operations with the TFP3 hardware.

3.8 Generation and Loading of a Package File for TFP3 Secure Models

Generation and loading of a package file is only supported on the secure models of the TFP3. The package file is a structure of data containing the key constant, maximum programming count, hex file name to be stored on the TFP3 device and the hex file data contents. The data in a package file is AES 128-bit encrypted, with each data packet forming a 16-byte record. Generate this package file by executing the **Generate Package File** command (**g**). This command requires the user to input the AES 128-bit key, the maximum programming count value, the package file name and the hex file to load onto the TFP3 device.

The TFP3 host application, when the **Load Package File** command (**H**) is executed, loads this package file data onto the TFP3 device. The loading of a package file must pass two levels of checking, detailed below, before the DUT hex code is loaded onto the TFP3 device.

1. **Key constant flag comparison**

   The **Load Package File** command first sends 16 bytes of AES 128-bit key encrypted constant data to the TFP3 device. Upon receiving this data, the TFP3 device will decrypt it with its AES 128-bit key and compare the data received to a Maxim constant string. This key constant flag matching makes sure that the TFP3 device is accessed only by an authorized person who knows the key that is stored on the TFP3 device. If the comparison is successful, the loading of a package file moves to the second level of checking.

2. **Hex file name comparison**

   The **Load Package File** command sends 16 bytes of AES 128-bit key encrypted hex file name to the TFP3 device. Upon receiving this data, the TFP3 device will decrypt it with its AES 128-bit key and compare it with the hex file names stored in its EEPROM. If the comparison fails, then the TFP3 stores the hex file name on the TFP3 EEPROM and increments the Hex File Name counter by 1. Then the TFP3 receives the DUT hex file data from the host and stores it in the TFP3 flash memory. This feature eliminates the problem of loading the same hex file to the TFP3 multiple times. The Hex File Name counter will reset to 0 after 20 unique hex files are loaded.

Figure 12 shows the flowchart and the sequence of operation on the TFP3 device side for executing the **Load Package File** command (**H**).

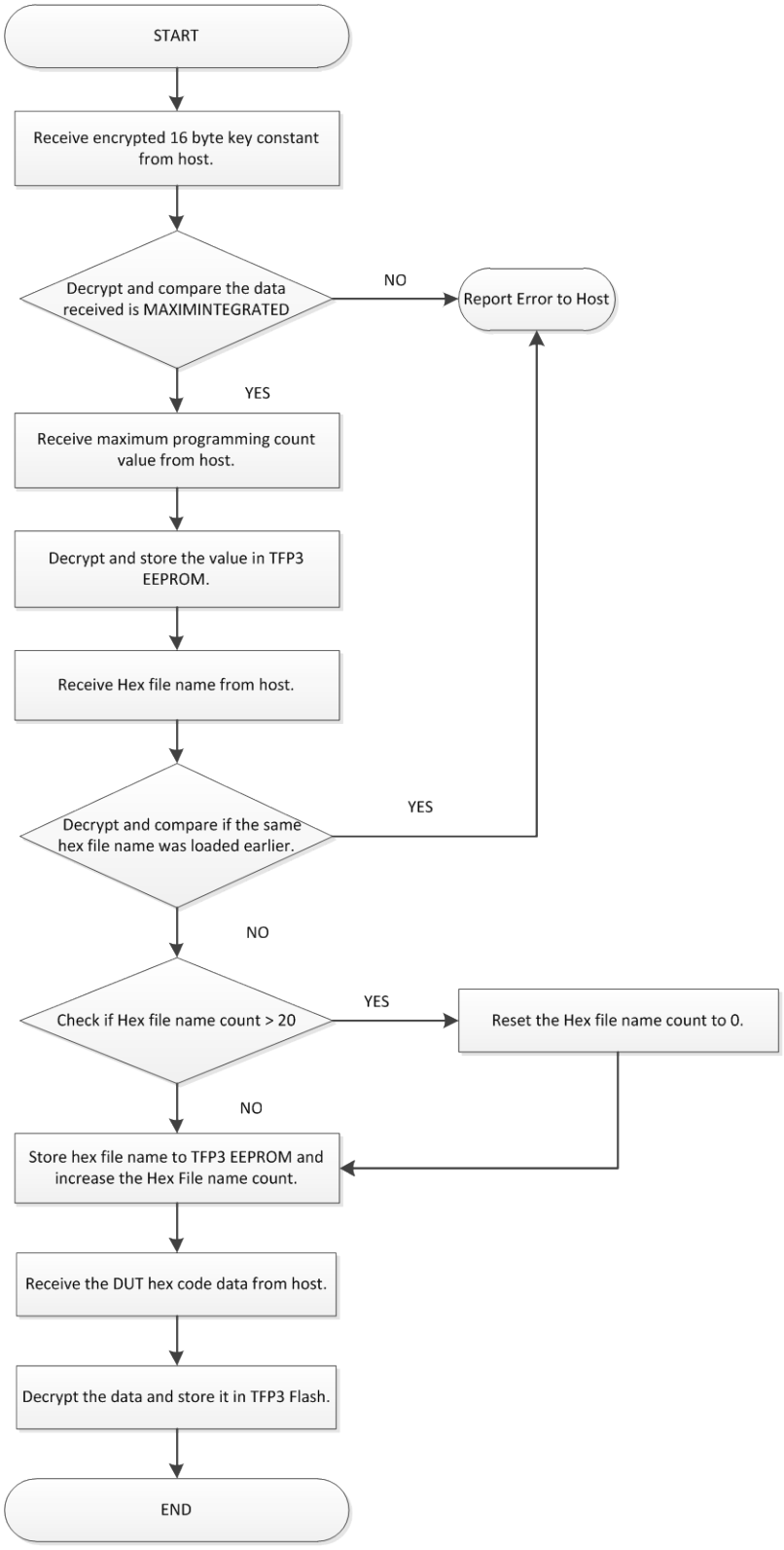


Figure 12: Flowchart for Execution of Load Package File Command on Target Side

3.9 TFP3 Parameter Preservation

The TFP3 firmware typically performs a bulk erase prior to programming the TFP3 device's flash memory. Additionally, the TFP3 programs the entire address space of the flash memory. Optionally, the TFP3 can reserve 16 bytes of flash memory for parameter data. This feature is only supported on the TFP3 secure model.

Use the **Enable or Disable Parameter Preservation** command (**E**) to toggle whether the 16 bytes of the parameter data is preserved or erased/overwritten during the program and verify operation.

After setting the parameter preservation mode, programming can be initiated using the **Program and Verify Operation** command (**P**) or by pressing the front panel Program button in stand-alone mode.

When parameter preservation mode is enabled, the TFP3 programming operation performs a bulk erase and reprogramming of the DUT flash memory. TFP3 then reads 16 bytes from the DUT parameter preservation address location that was requested by the user and checks whether the memory area has all 0xFFs present. If new code (non-0xFF data) is encountered, the TFP3 device continues with the comparison operation and sends a warning message to the host. The host then displays "Parameters preservation failed" after completing the program and verify operation.

The TFP3 device forms a 16-byte parameter structure that contains the unique serial number count, temperature, and RTC data.

1. **Unique serial number**

   This parameter is a unique 4-byte number whose default value can be set by the user. The default value of the unique serial number set by the TFP3 is 0. This count is incremented for every successful programming of the DUT. This count is also stored in TFP3 EEPROM.

2. **Temperature data for DS7505 and DUT**

   The size of this parameter is 4 bytes, split into 2 bytes of DS7505 temperature data and 2 bytes of DUT temperature data, consecutively. The TFP3 reads the temperature of the DS7505 and DUT, and then stores the hex values. Refer to the datasheets of the DUT and the DS7505 for the formula to convert this value back to an actual temperature reading.

3. **RTC data for DS3231 and DUT**

   The size of this parameter is 2 bytes. The TFP3 records the deviation of RTC offset in PPM. The duration of the 1-second pulse from the 71M654x DUT and DS3231 is counted from the TMUX2OUT pin and 1S pin output, respectively. Each pulse's duration is measured by a 48MHz clock. Because the DS3231 1S output is accurate, it serves as a reference to measure the 71M654x RTC 1S output. The offset of both of these clocks is used to calculate the PPM error. Customers can use the PPM error value to adjust the DUT's crystal at the same temperature when the program and verify operation is performed.

   The PPM error value recorded by the TFP3 device is a two's complement value, with the fifteenth bit representing the negative value.

Note: The DUT RTC and Temperature data is valid only if the connected DUT is 71M654x-based. Otherwise, the data read will be all 0xFFs.

Table 6 details the parameter preservation data alignment. The Parameter Address value is entered by the user.

Table 6: Parameter Data Alignment

Address in Flash	Item	Bytes
Parameter Address - 1	User code area	—
Parameter Address	Unique serial number	4
Parameter Address + 4	DS7505 temperature data	2
Parameter Address + 6	DUT temperature data	2
Parameter Address + 8	PPM of DS3231 and DUT RTC	2
Parameter Address + 10	Reserved bytes of 0xFF	6
Parameter Address + 16	User code area	—

3.10 TFP3 Firmware Upgrade Using In-Application Programming

TFP3 supports the self-update of its firmware via USB interface. The firmware of the TFP3 can be upgraded by executing the **In-Field Program Update** command (**U**). The user specifies the latest hex file, and then the TFP3 host application parses the hex file and sends the data to the TFP3 device. The TFP3 code described here implements a technique for upgrading the application residing in the TFP3 flash memory via the USB interface. To achieve this, the flash must be divided into three sections: the boot manager area, the application area, and the upgrade image area. The image area should have memory greater than or equal to the application area.

Boot Manager Area

This area is pre-loaded with the TFP3 boot code during TFP3 production. This code handles the following operations:

1. Signature checking or validation
2. Determining which area of code to jump to and execute
3. Uploading the application area code when the TFP3 is a fresh part

Application Area

This area is loaded with the TFP3 application code, which handles the following operations:

1. Receiving commands and performing TFP3 flash utility operations on the DUT
2. Copying of new TFP3 application code sent from the host to the image area, when an **In-Field Program Update** command (**U**) is executed by the user

Image Area

This area is will hold the latest application code to be programmed into the TFP3 application area. This area also has signature bytes stored at the address (0x1FFF8). The boot code and the application code use this signature area to update it with the valid signatures.

Figure 13 shows the TFP3 flash memory partition layout.

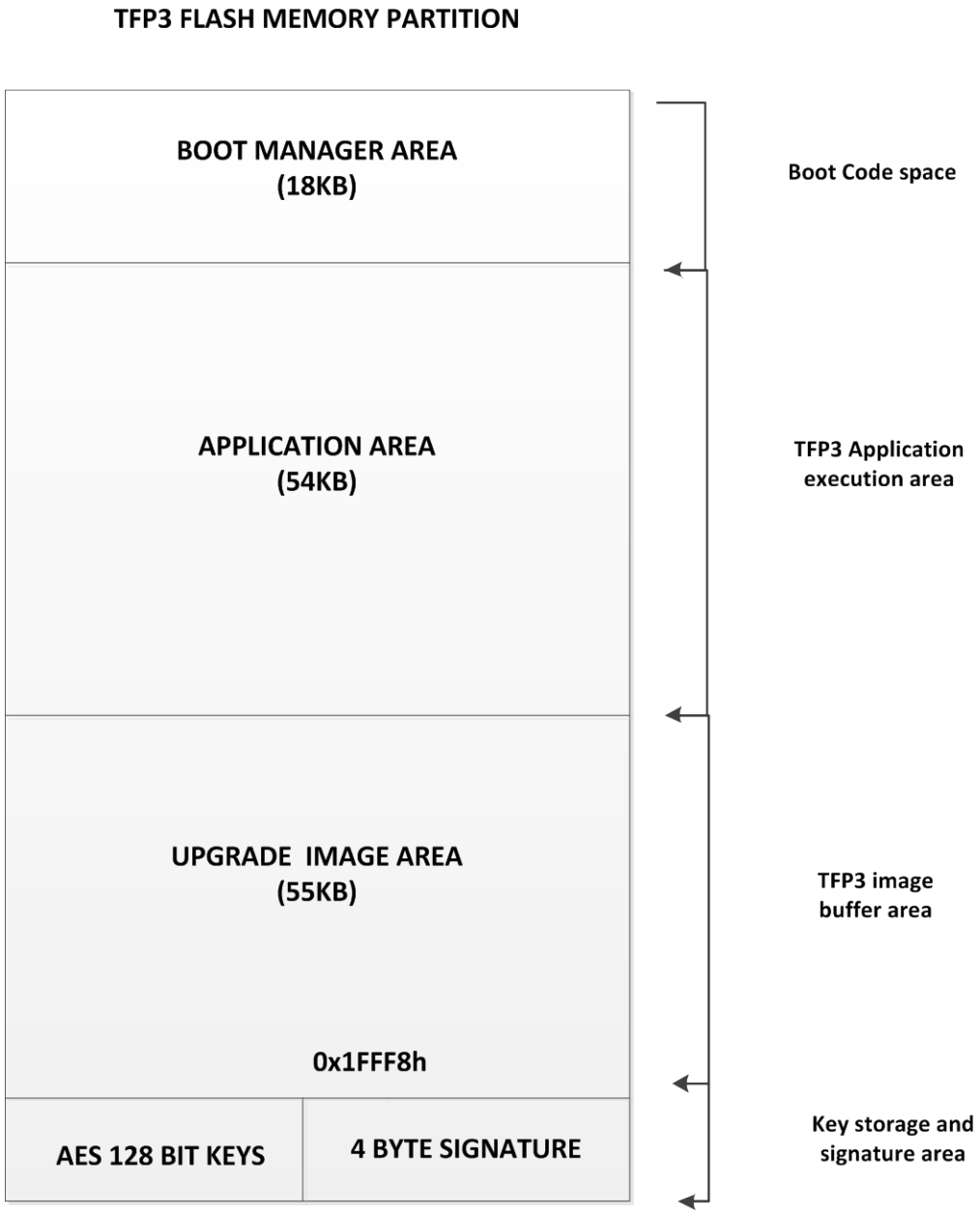


Figure 13: TFP3 Memory Partition Layout

There are three types of signatures implemented in the TFP3 device:

1. Default signature of 4 bytes (0xFFFFFFFF). This signifies that there is no application code present in the application code area or the image area.
2. Application signature of 4 bytes (0x12345678). This signifies that there is a valid application code present in the application area.
3. Upgrade signature of 4 bytes (0xAA55AA55). This signifies that there is a new, valid application code in the image area to be copied into the application code area.

Figure 14 shows the TFP3 IAP and application code execution.

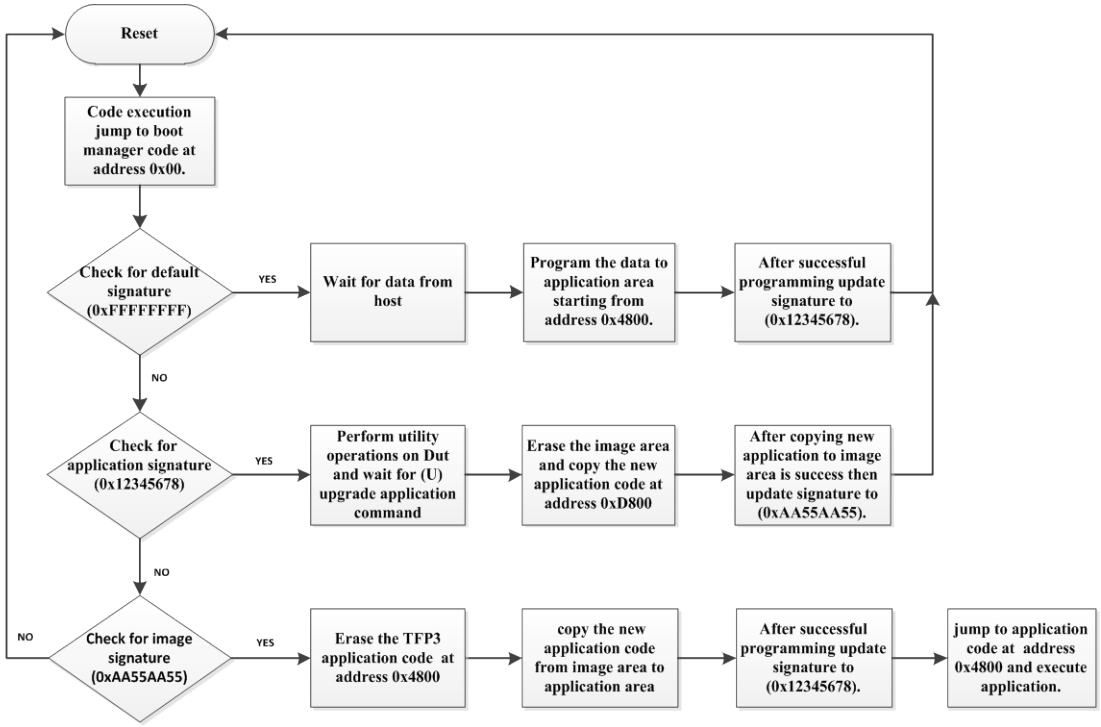


Figure 14: TFP3 IAP Programming Sequence and Code Execution

The upgrade image area is free to have the secondary application code (i.e. new code) loaded into it, or it can be left blank waiting for another upgrade image. At any time, the application code can erase the upgrade image area and write new data into it. This data can be marked as an upgrade image or secondary application code.

3.11 Loading of DUT Hex File to the TFP3

The TFP3 non-secure model supports downloading of the DUT hex file directly to the TFP3 without performing any secure operations. The user can simply select the hex file to be loaded and then run the **Program and Verify Operation** command (**P**) to download the hex file onto the target.

4 Status Indications

The TFP3 provides indication of the status of operations with LED and buzzer signals.

4.1 LED and Buzzer Indication

Table 7: Status LED and Buzzer Indications

TFP3 Operation	Status LED Operation
TFP3 Power-On	Red LED ON first and turns ON Green LED continuously for the TFP3-8051 model. Turns ON Green LED continuously for TFP3Q-MAXQ30 and TFP3L-MAXQ30 models.
TFP3 Programming	Green LED and Red LED toggles alternatively.
TFP3 Program and Verify Success	Green LED ON continuously.
TFP3 Program and Verify Fail	Red LED ON continuously. Buzzer will toggle between ON and OFF at rate of 100 milliseconds each for 3 times.
TFP3 USB Enumeration Fail	Green LED and Red LED along with buzzer toggles alternatively for 5 times at rate of 300 milliseconds and then Green LED glows continuously.
TFP3 Max Programming Count Reached	Green LED and Red LED toggles alternatively for 3 times at rate of 2 seconds and then glows Red LED continuously.
TFP3 Parameter Preservation Fail	Turn Red LED continuously ON.

5 Supported Commands

The command line interface commands supported by the TFP3 secure and non-secure models are listed below.

Table 8: TFP3 Supported Commands

SNO	Command	Description	Supported by TFP3 8051	Supported by TFP3Q MAXQ30	Supported by TFP3L MAXQ30
1	?	Displays the Help Menu and commands usage.	YES	YES	YES
2	V	Displays TFP3 host application Version and Firmware Version information.	YES	YES	YES
3	Z	This command displays in which mode the TFP3 device is operating either IAP or Application mode.	YES	YES	YES
4	e	This command only Erases the flash memory of target connected.	YES	YES	YES
5	T	Displays the Mode in which the DUT is (secure or non-secure mode).	YES	YES	NO
6	P	This command Erases, Programs and Verifies the DUT flash memory.	YES	YES	YES
7	G	This command gives user the diagnosis information like total count of Programming cycles executed, PASS and FAIL counts of TFP3.	YES	YES	YES
8	R	This command resets the PASS and FAIL counts to 0x00.	YES	YES	YES
9	F	This command is used to set the flash size of DUT connected.	YES	YES	YES
10	K	This command is used to Modify/Set the encryption and decryption key in TFP3. This will be used for encrypting and decryption of DUT flash memory contents.	YES	YES	NO
11	D	This command gets the DUT firmware stored in the TFP3's internal flash memory to the host and saves as hex file.	YES	YES	NO
12	C	This command compares the DUT firmware contents stored in TFP3's internal flash memory to the connected DUT's flash memory contents.	YES	YES	YES

SNO	Command	Description	Supported by TFP3 8051	Supported by TFP3Q MAXQ30	Supported by TFP3L MAXQ30
13	Q	This Factory reset command is used to reset the diagnosis information of TFP3 and also the security keys to its default values in secure model. This command also erases the target flash memory code and also the TFP3 internal memory contents.	YES	YES	YES
14	E	This command is used to enable or disable the parameter preservation while TFP3 programming and verify operation is in progress.	YES	YES	NO
15	B	This command is used to read the software version of the boot code present on the TFP3 device.	YES	YES	YES
16	U	This command is used to upgrade the image area with the latest application code received from host for performing the IAP over USB interface.	YES	YES	YES
17	g	This command is used to generate a package file in the host.	YES	YES	NO
18	H	This command is used to load the package file data to the TFP3 device.	YES	YES	NO
19	M	This command sets the serial number count of DUT.	YES	YES	NO
20	m	This command gets the serial number of DUT.	YES	YES	NO
21	I	This command sets the Model Type of TFP3 to communicate with DUT.	YES	YES	YES
22	T	This command gets the Model Type of TPP3 connected to DUT	YES	YES	YES
23	J	This command sets the JTAG clock in TFP3	NO	YES	YES
24	h	This command gets the JTAG clock from TFP3	NO	YES	TES

5.1 Commands Common to Secure and Non-Secure Models of the TFP3

Table 9: Commands Common to Both Secure and Non-Secure Models

Command	Description	Usage with TFP3 Console Application
?	Displays the Help menu and supported commands usage.	`TFP3.exe - ?`
V	Displays the TFP3 Host application version and TFP3 Firmware Version in format “TFP3-Vxx.yy SW-Vxx.yy”, where "xx" is the major version and "yy" is the minor version.	`TFP3.exe -V -l TFP3 -j USB -k com15 -i "115200,N,8,1"`
Z	This command displays in which mode the TFP3 device is operating, either IAP or Application mode.	`TFP3.exe -Z -l TFP3 -j USB -k COM15 -i "115200,N,8,1"`
e	This command bulk erases the DUT’s flash memory.	`TFP3.exe -e -l TFP3 -j USB -k COM15 -i "115200,N,8,1"`
P	This command is used to Erase, Load and Verify the DUT’s flash memory. The DUT firmware to be programmed is read from the TFP3’s internal flash memory and programmed in to the targets.	`TFP3.exe -P -l TFP3 -j USB -k com15 -i "115200,N,8,1"`
G	This command gives the TFP3 device’s diagnosis information to the user. This command gives three diagnosis counts: **1.** TFP3 Pass counts **2.** TFP3 Fail counts **3.** TFP3 total programming counts Default values of the pass and fail counts are 0x00. For every successful program and verify of the DUT, the Pass count is incremented by 1. For every unsuccessful program and verify of the DUT, the Fail count is incremented by 1. For every successful program and verify of the DUT, the total programming count is decremented by 1.	`TFP3.exe -G -l TFP3 -j USB -k COM15 -i "9600,N,8,1"` **Note:** When the total programming count reaches to zero, the DUT’s and TFP3’s flash is erased. User needs to set a new maximum programming count. Until a new maximum programming count is set by the user, the TFP3 will not perform the Program and Verify Operation. User has to send the Factory Reset command, load AES keys, generate and load a new package file to set a new programming count value.
R	This command resets the following two diagnosis counts to its default values : **1.** TFP3 Pass count **2.** TFP3 Fail count	`TFP3.exe -R -l TFP3 -j USB -k COM15 -i "9600,N,8,1"`

Command	Description	Usage with TFP3 Console Application
F	This command is used to set which type of DUT is connected to the TFP3 for performing the flash utility operations. This command has to be executed once every time a DUT of new flash size is connected. This command needs to be executed first before performing any operations with the TFP3. **Note:** Setting a wrong flash size value in the TFP3 will cause malfunctioning of the TFP3 device.	`TFP3.exe -F x -l TFP3 -j USB -k COM15 -i "9600,N,8,1"` Where "x" is the flash size of the DUT connected. 1 - Connected DUT's flash size is 16KB 2 - Connected DUT's flash size is 32KB 3 - Connected DUT's flash size is 64KB 4 - Connected DUT's flash size is 128KB 5 - Connected DUT's flash size is 256KB 6 - Connected DUT's flash size is 512KB
C	This command is used to compare the DUT's flash memory contents with the TFP3's internal flash memory contents.	`TFP3.exe -C -l TFP3 -j USB -k com15 -i "115200,N,8,1"`
Q	This command resets the TFP3 settings to default. The following are set to default values in the TFP3: **1.** Diagnosis information **2.** AES keys **3.** DUT and TFP3 flash memory	`TFP3.exe -Q -l TFP3 -j USB -k COM15 -i "115200,N,8,1"`
B	Displays the TFP3 IAP or boot code version in format "TFP3-Vxx.yy", where "xx" is the major version and "yy" is the minor version.	`TFP3.exe -B -l TFP3 -j USB -k COM15 -i "115200,N,8,1"`
U	This command is used to upgrade the image area with the latest application code received from the host for performing the IAP (In-Application programming).	`TFP3.exe -U -n xxxx.hex -l TFP3 -j USB -k COM15 -i "115200,N,8,1"` Where "xxxx.hex" is the latest TFP3 application code to be loaded in to image area of TFP3. **Note:** After copying the new firmware to the image area, perform a TFP3 device reset to start IAP (in-application programming) or TFP3 firmware upgrade.
I	This command sets the TFP3 Model type connected to the DUT. **Note: Setting an incorrect Model Type value in the TFP3 will cause the TFP3 device to malfunction.**	`TFP3.exe -I x –l TFP3 -j USB -k com15 -i "115200,N,8,1"` **Note:** The "x" the range is 0-2. 0 - TFP3 - 8051 model (secure) 1 - TFP3Q - MAXQ30 model (secure) 2 - TFP3L - MAXQ30 model (non-secure)

Command	Description	Usage with TFP3 Console Application
t	This command gets the configuration of model type.	`TFP3.exe -T -l TFP3 -j USB -k com15 -i "115200,N,8,1"`
J	This command sets the clock prescaler value of JTAG. This command is applicable when the Model Type TFP3Q -MAXQ30 or TFP3L - MAXQ30 is selected.	`TFP3.exe -J xxxx -l TFP3 -j USB -k com15 -i "115200,N,8,1"` **Note:** The value prescaler is 16 bits represented in “xxxx” decimal format.

Note: Use Boot loader type (-l option) as TFP3S for the secure model of the TFP3 and TFP3US for the non-secure model of TFP3 in the command above.

5.2 Commands Supported Only by the Secure Model of the TFP3

Table 10: Commands Supported Only by the Secure Model

Command	Description	Usage with TFP3 Console Application
K	This command is used to modify the AES keys of the TFP3 device for AES 128-bit key encryption and decryption. This is a security feature command that has to be executed to modify/store the AES keys in the TFP3. This command should be performed during initialization of the TFP3. These keys are used to encrypt or decrypt the DUT program code.	`TFP3.exe -K -r xxxx.pas -a zzzz.pas -l TFP3S -j USB -k COM15 -i "115200,N,8,1"` Where "xxxx.pas" is the new AES 128-bit key. "zzzz.pas” is the old AES 128-bit key data file with .pas extension. The AES key data size is 32 ASCII characters ranging from 0 to F. The default AES keys of the TFP3 device is all FFs.
D	This command is used to dump the DUT’s firmware onto the host and store the output as a hex file. The data read from the TFP3 device is encrypted and must be decrypted with 128-bit AES keys before creating a file in the host.	`TFP3.exe -D 200 -r Aes_Keys.pas -O "xx\yy\zz" -l TFP3S -j USB -k COM15 -i "115200,N,8,1"` Where " xx\yy\zz " is path where the DUT hex code read from the TFP3 device to be stored in the host. This command requires AES 128-bit key as one of the command arguments to decrypt the encrypted data received from the TFP3 device.

Command	Description	Usage with TFP3 Console Application
E	The Enable or disable parameter preservation command helps to preserve the parameter data in the DUT while the program and verify operation is in progress. The following data can be preserved by sending this command: **1.** DUT serial number count, a unique value set by the user that increments for every successful program and verify operation. **2.** Temperature sensor data for calibration. **3.** RTC data of DUT and DS3231 in PPM for calibration.	`TFP3.exe -E 1 -y 10000 -w 0 -l TFP3S -j USB -k COM15 -i "115200,N,8,1"` The value of 'E' is as follows: 1 - Enable the parameter preservation 0 - Disable the parameter preservation Where "w" is the DUT series connected type. 1 - DUT connected type is 71M654X series 0 - DUT connected type is not 71M654X series Where "y" is the address in DUT. The parameter preservation data will be stored in the provided address of DUT. It is the user's responsibility to make sure that the DUT has empty space (data of FFs) before writing. **Note:** The parameter preservation is possible only with the Maxim Integrated 71M654X series of microcontrollers.
g	This command is used to generate a package file in the host.	`TFP3.exe -g -S 100 -n aabbcc.hex -r ddeeff.pas -O xx\yy\zz TFP3_testing -W xyz.txt -l TFP3S -j USB -k COM15 -i "115200,N,8,1"` Where the value for S (maximum programming count) is in the range 1 - 1000000. Where "aabbcc.hex" is the DUT hex file to be loaded to the TFP3 device. Where "ddeeff.pas" is the AES 128-bit keys file which is used to encrypt and data and generate a package file. Where "xx\yy\zz" is the path to store the generated package file in host. Where "xyz.txt" is the package file name to be generated.

Command	Description	Usage with TFP3 Console Application
H	This command is used to load the package file data onto the TFP3 device. The package file contains the following data: **1.** Key constant **2.** Hex file name to be stored **3.** Maximum programming count value **4.** Hex file data to be stored. All the data in the package file is encrypted by AES 128-bit key and sent to the TFP3 device. The TFP3 device will decrypt the data and use it for loading the DUT hex data.	`TFP3.exe -H -z xxx.txt -l TFP3S -j USB -k COM15 -i "115200,N,8,1"` Where "xxxx.txt" is the name of the package file to be loaded on to the FP3 device.
m	This command retrieves the serial number of the DUT.	`TFP3.exe -m -l TFP3S -j USB -k COM15 -i "115200,N,8,1"`
T	This command retrieves the DUT mode as either secure or non-secure.	`TFP3.exe -T -l TFP3S -j USB -k COM15 -i "115200,N,8,1"`

5.3 Commands Supported Only by the Non-Secure Model of the TFP3

Table 11: Commands Supported Only by the Non-Secure Model

Command	Description	Usage with TFP3 Console Application
m	This command sets the maximum programming count of the TFP3. This is the number of times that the TFP3 will program a DUT hex file into the target.	`TFP3.exe -m xxx -l TFP3US -j USB -k COM15 -i "115200,N,8,1"` Where “xxx” is the programming count value in the range 0-1000000. Default value is 1000000 (1 million) and starts decrementing after every successful programming of a hex file onto the DUT.
H	This command is used to load the DUT Hex file onto the TFP3 device.	`TFP3.exe -H -z xxx.hex -l TFP3US -j USB -k COM15 -i "115200,N,8,1"` Where "xxx.hex" is the name of the DUT hex file to be loaded onto the TFP3 device.

6 Hardware Specifications

Environmental	
Operating Temperature	+10° to +50°C
Storage Temperature	-40°C to +85 C
Power Supply	
Supply Voltage	+5V DC ±10% regulated
Supply Current	50mA max
Cable Plug	USB AM to BM
ATE Connector	
Input Voltage Low	0.0V to +0.5V
Input Voltage High	+2.0V to +3.3V
Output Voltage Low	+0.45V max at 8mA
Output Voltage High	+2.4V min at 2mA
ICE Connector	
Input Voltage Low	0.0V to +0.5V
Input Voltage High	+2.0V to +3.3V
Output Voltage Low	+0.45V max at 8mA
Output Voltage High	+2.4V min at 2mA
TCLK Frequency(CC51)	10MHz to 48MHz
Output (Power DUT)	
Output Voltage	+3.3V DC ±10%
Output Current	300mA max
Dimensions of Box	
Length	4.6 inches (116.84 mm)
Width	3.1 inches (78.74 mm)
Height	1.5 inches (38.1 mm)
LxWxH	116.84 x 78.74 x 38.1 mm

7 Ordering Information

The following table lists the order numbers used to identify the TFP3 models.

Table 12: Ordering Numbers

Model Number	Ordering Number	Description
Flash Programmer Model TFP3-8051	80515-FPBM-TFP3	8051 target support Secure Model TFP3
Flash Programmer Model TFP3Q-MAXQ30	MAXQ30-FPBM-TFP3Q	MAXQ30 target support Secure Model TFP3
Flash Programmer Model TFP3L-MAXQ30	MAXQ30-FPBM-TFP3L	MAXQ30 target support Non-secured Model TFP3

8 Glossary of Terms and Abbreviations

Advanced Encryption Standard (128 bits)_	AES-128
ATE	Automated Test Equipment
CLI	Command-Line Interface
DUT (Device Under Test)	Target Microcontroller
ECB	Electronic Code Book
Flash Utility Operations	Program, Verify, Dump, and Erase
GUI	Graphical User Interface
Host	PC running Microsoft Windows®
Host Application	PC Console Application
IAP	In-Application Programming
JTAG	Joint Test Action Group
PC	Personal Computer
Signum Flash Interface Protocol	CC51
TAP	Test Access Port
TFP3	The Flash Programmer
TFP3L	The Flash programmer - MAXQ Lite
TFP3Q	The Flash Programmer - MAXQ
USB	Universal Synchronous Bus

REVISION HISTORY

Revision, Date	Change	Page(s)
2.2, Mar. 2015	Added note on powering the target board under Table 2 and Table 3	**16, 17**
2.1, Jan. 2015	Revised all formatting using Customer-Facing Documents template	**All**
1.1, Nov. 2014	Added TFP3L version	
1.0, June 2014	Initial release	