MAX32560 SECUREROM PACKAGE INSTALLER AND QUICK START

UG90H02

August 2017

maxim integrated™

Rev. B, August 2017

CONTENTS

REV. B, AUGUST 2017 2

1 Purpose

- The MAX32560 SecureROM package is a set of tools, examples and documents enabling user to securely program, load and run applications on the MAX32560 secure microcontroller .
- This package enables users to set a MAX32560 in phase 3, program the CRK on the MAX32560, move the MAX32560 to phase 4, program an application and runs it on the MAX32560.
- This package requires installation of 3rd parties tools.
- This is run on a PC, connected to the MAX32560.
- This package demonstrates:
  - how to set up the MAX32560 life-cycle,
  - how to program the customer CRK, using a test CRK,
  - how to program a demo application in the internal flash, to be run in the internal RAM.
  - a glossary is available at the end of this document

2 Release Notes

- This package version supports MAX32560 release A1: this is the default target
- This package version is validated on Windows Seven 64-bit and not on any 32-bit Windows OS

3 Tested configuration

- On PC side, this package has been tested on:
  - Windows Seven 64-bit, using Cygwin 64-bit 1.7,
- On embedded side, this package has been tested on
  - MAX32560 A2
  - Evkit 2.0

4 3rd parties' packages

- The Maxim Integrated delivered package contains documentation, examples and tools, from Maxim Integrated.
- 3rd party tools to be installed are:
    - For Windows users only:
        - Cygwin
            - a collection of tools which provide a Linux look and feel environment for Windows
            - needed for tools execution
            - download and install guide

5 Installation

1. copy the SecureROM package in `/cygdrive/c/securerom` (i.e. c:/securerom)
2. cygwin installation
    a. use the Cygwin_install.pdf document
3. set up package application
    a. Goto secureROM package home directory, e.g. securerom
    b. Select the targeted MAX32560 version, A1, by using the option --soc=A1 (default is A1)
    c. once this is done, the CLARA_SCRIPTS_PATH variable can be used for locating the package directory.

6 Use

6.1 program the CRK

1. program the test CRK and move MAX32560 life cycle phase from 3 to 4 (to be done once per chip)
   a. Note: the SCP packets have already been generated (and stored in write_maximtestcrk_packets) for this test CRK, this is why programming can be done immediately. These packets are an example of what Maxim Integrated generates from the CRK public key supplied to Maxim Integrated.
   b. the connection between the Host and the MAX32560 is a serial connection, 115200 8N1,
   c. connect the EvKit UART0 connector to the targeted COM port,
   d. run writecrk.sh with the chosen COM port (in our example, COM1 for Windows, /dev/ttyS0 for Linux) and prod_p3_write_crk as parameters

```
$ cd <securerom_path>/scripts
$ bash ./writecrk.sh COM1 ./write_maximtestcrk_packets
Ready to execute <securerom_path>/script/write_maximtestcrk_packets
Power cycle the MAX32560 system then press [Enter] IMMEDIATELY!
Please wait...
Open file: packet.list
Open serial port: COM1 (timeout: 2s)
Start SCP session (use -v for details)
Trying to Connect.
\
Connected !
[==================                                                  ]25%
ROM Version : 01020001
Phase : 03
JTAG : Active
Rework : Not Available
USN : a367081024050af0010430ca63
[=====================================================================] 100%
Disconnecting. . .
Disconnected !
SCP session OK SUCCESS.
```

2. Note1: the MAX32560 is now in phase 4, with the test CRK programmed in its OTP. Any SCP command will now require a signature with this key to be approved.

6.2 program the application

1. program test application in the MAX32560 internal flash,
   a. signed application
      i. the application signing operation is automatically performed within the MML SDK environment, in the Eclipse IDE
      ii. In the scripts/sla folder, there is a binary example of such signed application, freertos_demo.sbin
      iii. This .sbin has been signed using the test CRK private key.
   b. Generate the packets for the application programming

```
$ cd <securerom_path>/scripts
$ bash ./build_application.sh ./sla/freertos_demo.sbin buildapp ../keys/maximtestcrk.key
ROMVERSION=01020001
SBL/SCP packets builder v3.7.14 (build 1) (c)Maxim Integrated 2006-2014
--warning: this tool does not handle keys in a PCI PTS compliant way --
WARNING: <.\session_build.ini> not found
<securerom_path>/scripts/build_application.sh/scp.log> created
Generated SCP packets in: build_application.sh
Use sendscp.sh <serial_port_spec> build_application.sh to run the SCP script generated by
this utility.
SUCCESS.
```

i. buildapp directory has been created and packets have been generated and stored in

c. program the generated packets

i. connect the EvKit to the targeted serial port (COM1 is for Windows, to be replaced by /dev/ttyS0 for Linux)

ii. use sendscp.sh script to run the SCP session generated by this script

```
$ cd <securerom_path>/scripts
$ bash ./sendscp.sh COM1 buildapp
Ready to execute <securerom_path>/scripts/buildapp
Power cycle the MAX32560 system then press [Enter]!
Please wait...
Open file: packet.list
Open serial port: COM1 (timeout: 2s)
Start SCP session (use -v for details)
Trying to Connect.
/
Connected !
[======                                                                ]  8%
      ROM Version : 01020001
      Phase : 04
      JTAG : Active
      Rework : Not Available
      USN : 0500abcdef01000102abcdf6ae
[========================================================================] 100%
Disconnecting. . .
Disconnected !
SCP session OK SUCCESS.
```

6.3 Execution

1. start the application by resetting the board
   a. the EvKit FreeRTOS demo starts.

7 Notes

- If the application does not start, it is maybe because the MAX32560 is not configured in 144-pin package
- The setup.sh configures the package for a specific MAX32560 version. For another configuration, just run setup.sh with the accurate version specified with the --soc parameter.
- It is possible to change the SCP ports (UART and USB) timings, using a script and session build tool
  - for no UART timing: write-timeout 0 0000
  - for no USB timing: write-timeout U 0000
  - for no VBUS timing: write-timeout V 0000
  - for USB timing set up at 5s (i.e. 5000ms = 0x1388): write-timeout U 8813
  - the typical use is:

```
$ cd scripts

$ mkdir session_timeout

$ ../bin/session_build.exe session_mode=SCP_ANGELA_ECDSA verbose=yes ←-

output_file=session_timeout/session_timeout pp=ECDSA script_file= ←-

script_timeout.txt ecdsa_file=../keys/crk_ecdsa_angela_test.key

$ cd session_timeout

$ ls -1 *.packet >packet.list

$ cd ..

$ bash ./sendscp.sh COM1 session_timeoutMAX32560 SecureROM package installer
and quick start guide
```

8 Glossary

CRK	Customer Root Key, the customer ECDSA 256-bit public key, to be written in the OTP, used for signatures verifications
OTP	One-Time Programming Memory, an embedded memory used for configuration data storage
phase	The MAX32560 life cycle is made of several phases: phase 3 allows to program the CRK, phase 4 allows to program and run applications
SCP	Secure Communication Protocol: the secure protocol over UART and USB used for firmware, keys updates, OTP configuration, applets execution
SLA	Second-Level Application: the final application launched by the SecureROM
USN	Unique Serial Number, a value unique per chip.

9 Tools purpose

- For more details, see UG90H03 Secure ROM code User Guide
- session_build: builds the SCP packets from a script (typically the signed CRK or a signed final application)
- serial_sender: sends the SCP packets to the chip

REVISION HISTORY

Revision, Date	Change	Page(s)
2.0.0, August 2017	Revised instructions	**4-8**
1.0.0, January 2017	Initial version	**1-6**