Instruction

Manufacture Z-Wave 700 product in volume

Document No.:	INS14285
Version:	6
Description:	-
Written By:	JFR;SGANGULY;SCBROWNI
Date:	2019-09-16
Reviewed By:	OPP;JBU;LTHOMSEN;PSH;SCBROWNI;BBR;JOPEDERSEN;ABXAVIER
Restrictions:	Public

Approved by:

Date	CET	Initials	Name	Justification
2019-09-16	10:52:58	NTJ	Niels Johansen	

REVISION RECORD				
Doc. Rev	**Date**	**By**	**Pages Affected**	**Brief Description of Changes**
1	20190130	JFR	All	Initial draft
2	20190207	JFR	2	Clarified Lock Bit Page setting to protect IP and security material.
3	20190711	SGANGULY JFR	2	Split manufacturing flow for end devices and gateways. Also added details about bootloader and signing keys etc.
4	20190712	SCBROWNI	All	Minor typos
5	20190903	JFR	2.1 & 2.2	RAILTest replacing ApplicationTestPoll used in 500 Series
6	20190911	SCBROWNI	2.1 & 2.2	Edited and reviewed the changed sections.

Table of Contents

1 INTRODUCTION

This document describes the manufacturing test flow for Z-Wave 700 SoC-based products.

2 MANUFACTURING FLOW

The manufacturing flow differs for end devices (ZGM130S, etc.) and gateways using EFR32ZG14 because the Serial API Bridge Controller application is made by Silicon Labs to EFR32ZG14. Therefore, the bootloader on EFR32ZG14 must be flashed together with the Silicon Labs public signing key and encryption key to support firmware update of a new gbl file from Silicon Labs. For end devices the public signing key and encryption key will be manage by the vendor in question. In addition, it is also mandatory to adjust/calibrate Xtal crystal on EFR32ZG14-based products.

2.1 End Devices

The manufacturing production test flow for end devices (ZGM130S, etc.) must incorporate the following steps:

- Product-specific testing such as I/O, etc. Refer to 'Using RAIL Test' under SDK documentation section in the Simplicity Studio distribution.
- For RF testing, etc., use RAILTest. Refer to [7] regarding RF testing. The 500 Series ApplicationTestPoll function is not available in 700.
- Download OTA bootloader to SoC target via Serial Wire Debug (SWD) interface.
- Write your own public signing key and encryption key to SoC target via Serial Wire Debug (SWD) interface. There is a readme.txt file in the Z-Wave SDK release that tells how to generate your own keys and write them to device Lock Bits Page.
  - The path to the readme file in the Z-Wave SDK release is as follows:

    "Your ZWAVE Installation Directory\BootLoader\sample-keys\"
- Download the application firmware to the SoC target via the Serial Wire Debug (SWD) interface. Do not set the Lock Bit in this step.
- The application in SoC signals when the security materials, etc., are in place in the Lock Bit Page via manufacturing token TOKEN_MFG_ZW_INITIALIZED. The following steps are performed in the SoC at the application startup:
  - If public/private keypair and QR code are already present in the Lock Bit page (Check manufacturing token TOKEN_MFG_ZW_INITIALIZED), jump to the last step continuing normal operation. Refer to [1] for details about manufacturing tokens.
  - Calculate public/private key based on Curve25519.
  - Construct the QR code using public key, product type, and product ID (latter two from application) as described in [2].
  - Calculate SHA-1 checksum as per [2] and incorporate it in the QR code.
  - Write the QR code to Lock Bit Page as manufacturing token TOKEN_MFG_ZW_QR_CODE.
  - Write private/public keypair to Lock Bit Page as manufacturing tokens TOKEN_MFG_ZW_PRK and TOKEN_MFG_ZW_PUK.
  - Write completion of Lock Bit Page initialization as manufacturing token TOKEN_MFG_ZW_INITIALIZED. This token can be used to sync completion of data to Lock Bits Page in a production system.
  - Continue normal startup.
- Readout QR code from SoC.

- Set Lock Bit Page [6] to protect IP and security material against untrusted entities.
- QR code labeling of product. Refer to [5] for details.

The QR code format enables customization of the QR code with extra TLVs (e.g., MaxInclusionRequestInterval, proprietary serial number, etc.) instead of using the internally generated one. The manufacturing line programmer must then read out the public key, etc., and compose the wanted QR code and print it to a label. The new QR code can also be stored in, e.g., the User Data Page.

Set the following registers in the Lock Bit Page [6] as a minimum to protect IP and security material:

DLW = Disable the debug port by clearing the four LSBs
ULW = Ignore
MLW = Optional (disable mass erase through MSC)
ALW = Optional (disallow a mass erase operation)
CLW1 = Ignore
CLW2 = Ignore
PLW[0…121] = Ignore

2.2 Gateways

The manufacturing production test flow for gateways using EFR32ZG14 must incorporate the following steps:

- Product-specific testing such as I/O, etc. Refer to 'Using RAIL Test' under the SDK documentation section in the Simplicity Studio distribution.
- Optional application for EFR32ZG14-based products performs crystal adjustment. Refer to [4] for details.
- For RF testing, etc., use RAILTest. Refer to [7] regarding RF testing. The 500 Series ApplicationTestPoll function is not available in 700.
- Download OTW bootloader to the SoC target via Serial Wire Debug (SWD) interface.
- Write the public signing key and encryption key to the SoC target via the Serial Wire Debug (SWD) interface. These keys are necessary for upgrading the firmware in the field. Following simplicity commander commands will be used for writing keys into the device's Lock Bits Page.
    - commander flash --tokengroup znet --tokenfile zg14_encrypt.key --tokenfile zg14_sign.key-tokens.txt -d EFR32ZG14
    - The key files are locked in the Z-Wave release following path in your SDK installation

      "Your ZWAVE Installation Directory\BootLoader\ZG14-keys\"
- Download the application firmware to the SoC target via the Serial Wire Debug (SWD) interface. Do not set the Lock Bit in this step.
- The application in the SoC signals when security materials, etc., are in place in the Lock Bit Page via manufacturing token TOKEN_MFG_ZW_INITIALIZED. The following steps are performed in the SoC at the application startup:
    - If the public/private keypair and QR code are already present in the Lock Bit page (Check manufacturing token TOKEN_MFG_ZW_INITIALIZED), jump to the last step continuing normal operation. Refer to [1] for details about manufacturing tokens.
    - Calculate the public/private key based on Curve25519.
    - Construct the QR code using public key, product type, and product ID (latter two from application) as described in [2].
    - Calculate SHA-1 checksum as per [2] and incorporate it in the QR code.
    - Write the QR code to Lock Bit Page as manufacturing token TOKEN_MFG_ZW_QR_CODE.
    - Write private/public keypair to the Lock Bit Page as manufacturing tokens TOKEN_MFG_ZW_PRK and TOKEN_MFG_ZW_PUK.

  - Write completion of Lock Bit Page initialization as manufacturing token TOKEN_MFG_ZW_INITIALIZED. This token can be used to sync completion of data to the Lock Bits Page in a production system.
  - Continue normal startup.
- Readout QR code from SoC.
- Set Lock Bit Page [6] to protect IP and security material against untrusted entities.
- QR code labeling of product. Refer to [5] for details.

The QR code format enables customization of the QR code with extra TLVs (e.g., MaxInclusionRequestInterval, proprietary serial number, etc.) instead of using the internally generated one. The manufacturing line programmer must then read out the public key, etc., and compose the wanted QR code and print it to a label. The new QR code can also be stored in, e.g., the User Data Page.

Set the following registers in the Lock Bit Page [6] as a minimum to protect IP and security material:

DLW = Disable the debug port by clearing the four LSBs
ULW = Ignore
MLW = Optional (disable mass erase through MSC)
ALW = Optional (disallow a mass erase operation)
CLW1 = Ignore
CLW2 = Ignore
PLW[0…121] = Ignore

REFERENCES

[1] Silicon Labs, SDS14306, Software Design Specification, Z-Wave 700 Lock Bits and User Data Page Contents.

[2] Silicon Labs, INS13975, Instruction, SmartStart Production Control.

[3] Silicon Labs, SDS13937, Software Design Specification, Node Provisioning QR Code Format.

[4] Silicon Labs, INS14498, Instruction, Mandatory crystal adjustment for EFR32ZG14 based products.

[5] Z-Wave Alliance, Z-Wave Security 2 (S2) and SmartStart Product Labeling Requirements.

[6] Silicon Labs, EFR32xG1 Wireless Gecko Reference Manual. Rev. 1.1.

[7] Silicon Labs, INS14283, Instruction, Bring-up/test HW development.