Application Note

Smart Start Labeling Guide

Document No.:	APL14131
Version:	3
Description:	-
Written By:	ABR;JFR;NOBRIOT;BBR
Date:	2018-03-05
Reviewed By:	JFR;TMORTENSEN;ABR
Restrictions:	Approved Partners Only

Approved by:

Date	CET	Initials	Name	Justification
2018-03-05	14:41:01	NTJ	Niels Thybo Johansen	

SILICON LABS

CONFIDENTIAL

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1	20171007	ABR	ALL	First revision
2	20171013	JFR	Front page	Changed title
3	20180305	BBR	All	Added Silicon Labs template

Table of Contents

1 ABBREVIATIONS

Abbreviation	Explanation

2 INTRODUCTION

The Smart Start framework enables a new level of service provider support and user friendliness. Various pieces of information are required to achieve this.

This application node presents the steps required to prepare, design and produce a Z-Wave end device with Smart Start support.

2.1 Audience and prerequisites

Z-Wave application designers and production personnel.

3 PREPARATION – HARDWARE

The following hardware components are required

- ZDP03A Z-Wave Programming Board [*)]
- ZDB5101 Z-Wave Development Board mounted on the ZDP03A
- UZB Bridge Controller USB Stick
- Development PC running Windows 7, 8 or 10.
- (Optional) Printer for generation of QR code labels
- (Optional) QR code scanner device that acts as a USB keyboard

[*)] The ZDP03A is intended for development and prototype production only. For large-scale production environments, more stable connectors and module fixtures are needed. Silicon Labs recommends that Equinox Programming systems are used for Z-Wave chip programming in production.

4 PREPARATION – SOFTWARE

The following software components are required in the development PC

- Microsoft Windows 7, 8 or 10
- Keil Development Environment
- Z-Wave SDK 6.81.00:
  - SwitchOnOff sample application project
  - PC Programmer
  - PC Controller
  - PC Zniffer
- MicroSoft Excel 2003, 2010 or newer
- MicroSoft PowerPoint 2003, 2010 or newer

For building and testing Smart Start end devices, the following SDK components must be installed and properly configured for the corresponding hardware peripherals.

- SwitchOnOff sample application
- Z-Wave PC Programmer configured for ZDP03A USB device
- Z-Wave PC Controller configured for UZB Bridge Controller USB device
- QrCodeEncoder.xlsm QR code encoding utility
- QrCodeDecoder.xlsm QR code encoding utility

5 PREPARATION – APPLICATION

The SwitchOnOff sample application advertises various information via the NIF, Z-Wave Plus Information and Manufacturer Specific Information frames. The following information must be prepared for generation of QR codes:

Information	**Related Z-Wave commands**	**SwitchOnOff values**
Requested S2 Keys	S2	0x81 (S0 & S2 Unauthenticated bits)
Z-Wave Plus Device Type (Generic + Specific Device Class)	NIF	0x1001 (SWITCH_BINARY:POWER_SWITCH_BINARY)
Installer Icon Type	Z-Wave Plus Info	0x7000 (ICON_TYPE_GENERIC_ON_OFF_POWER_SWITCH)
Manufacturer ID	Manufacturer Specific Info	0x0000 (MFG_ID_SIGMA_DESIGNS)
Product Type	Manufacturer Specific Info	0x0003 (PRODUCT_TYPE_ID_ZWAVE_PLUS)
Product ID	Manufacturer Specific Info	0x0002 (PRODUCT_ID_SwitchOnOff)
Application Version	Version	0x0401 (ZAF_VERSION_MAJOR & ZAF_VERSION_MINOR)

6 BUILDING THE FIRMWARE IMAGE

There is no need to build the application ex– unless the application source code or property data (device type, icon type, manufacturer ID, etc.) has been changed.

A pre-built binary image for the SwitchOnOff application can be found in the “Bin” folder found in the “ProductPlus” folder of the SDK.
Choose the _BOOTLOADER variant intended for in-system programming.

7 PROGRAMMING AND SMART START PREPARATION - PROTOTYPING

For prototyping, the PC Programmer can be used to program individual Z-Wave boards via the ZDP03A programming board. For production, refer to chapter 8.

When building Smart Start products with the PC Programmer, the "Generate S2 keypair" and "Add S2 keypair" flags must be checked before clicking the <Calibrate, Program and Verify> button. Refer to [6].

After programming, be sure to make a sub-folder in the "Build" folder named "DSK_xxxxx-xxxxx-xxxxx-..", where the "xxxxx-xxxxx-xxxxx-…" is substituted with the DSK string advertised by the PC Programmer and copy the generated binary firmware image into this folder. This image contains unique S2 keys that match the actual DSK.

Certified Smart Start products must carry a label with a QR code and either a PIN or a complete DSK. It is however possible to demonstrate Smart Start inclusion with the PC Controller by simply pasting the DSK string "xxxxx-xxxxx-xxxxx-…" into the PC Controller; refer to chapter 9.

Copy the QrCodeEncoder.xlsm QR code encoding utility to the same "DSK_xxxxx-xxxxx-…." Folder and open the Excel sheet. Do not touch any fields marked with gray shading.
If prompted, click the <Enable Content> link in the top of the page.

- Update the Version field (F5) to 1 to make the QR indicate that it represents a Smart Start node

- Update the RequestedKeys field (F7) to a two-digit hex value, e.g. 0x81. The value must be derived from the application source code file config_app.h by evaluating the define REQUESTED_SECURITY_KEYS.
  Note that the application framework uses its own internal encoding of security class identifiers. The RequestedKeys field (F7) must be encoded by OR'ing the official S2 mask bit values ([7], table "*Requested Keys*"):
  - 0x01: S2 Unauthenticated
  - 0x02: S2 Authenticated
  - 0x04: S2 Access Control
  - 0x80: S0

- Update the DSK field (F8) to a 40- character string containing the decimal digits of the DSK without any delimiters. Be sure to enter a single quote character (" ' ")before the string to let Excel know that this is not a number but a string, e.g. '5152535455414243444531323334352122232425.
  The value must be copied from the DSK in the title of this particular firmware image folder and removing the delimiters.

- Update the TlvValue.ZwDeviceType field (F11) to a four-digit hex value, e.g. 0x1001. The value must be derived from the application source code file config_app.h by evaluating the defines

GENERIC_TYPE and SPECIFIC_TYPE.

- Update the TlvValue.ZwInstallerIconType field (F12) to a four-digit hex value, e.g. 0x7000. The value must be derived from the application source code file config_app.h by evaluating the define APP_ICON_TYPE.

- Update the TlvValue.ZwManufacturerId field (F15) to a four-digit hex value, e.g. 0x0000. The value must be derived from the application source code file config_app.h by evaluating the define APP_MANUFACTURER_ID.

- Update the TlvValue.ZwProductType field (F16) to a four-digit hex value, e.g. 0x0003. The value must be derived from the application source code file config_app.h by evaluating the define APP_PRODUCT_TYPE_ID.

- Update the TlvValue.ZwProductId field (F17) to a four-digit hex value, e.g. 0x0002. The value must be derived from the application source code file config_app.h by evaluating the define APP_PRODUCT_ID.

- Update the TlvValue.ZwApplicationVersion field (F18) to a four-digit hex value, e.g. 0x0401. The value must be derived from the application source code file ApplicationUtilities\zaf_version.h by evaluating the defines ZAF_VERSION_MAJOR and ZAF_VERSION_MINOR.

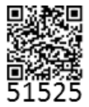
- The UUID field is optional but it must be included if a UUID value is stored in the chip NVM.
  - If a UUID is to be included in the QR code, the following applies:
    - Set the TLV-6:TlvTypeCritical field (F22) to 6
    - Set the TLV-6:TlvValue.UUIDPresFormat field (F24) to one of the following to select the format that a UI must use to present the UUID:
      - 0: 32 hex digits, no delimiters
      - 1: 16 ASCII chars, no delimiters
      - 2: "sn:" followed by 32 hex digits, no delimiters.
      - 3: "sn:" followed by 16 ASCII chars, no delimiters
      - 4: "UUID:" followed by 32 hex digits, no delimiters.
      - 5: "UUID:" followed by 16 ASCII chars, no delimiters
    - Set the TLV-6:TlvValue.UUIDData field (F25) to a 40-character string containing the decimal digits of the 16 byte UUID without any delimiters.
      Be sure to enter a single quote character (“ ’ ”)before the string to let Excel know that this is not a number but a string,
      e.g. '2122232425414243444511121314153132333435
  - If no UUID is to be included, fields F22, F24, F25 must be cleared.

The resulting QR code string is available in the “Complete QR Code String” field (A28).

If a physical QR code label is needed, do the following:

- Press the <Show QR Code> to launch a pop-up browser window with a QR code that contains the Complete QR Code String. (An online connection is required for this).

- Right-click the QR Code object and select “copy”.
- Paste the QR Code into a new PowerPoint page
- Adjust the QR code size to 10mm x 10mm
- Add a text box for the PIN code below the QR code.

  Insert the first 5 digits of the DSK with Calibri font, size 11, e.g. 

  (See also “Sample QR Code label.pptx” in the folder of this document)
- Print out the label and attach it to the product

8 PROGRAMMING AND SMART START PREPARATION - PRODUCTION

In a production setup, generation of S2 keys, DSK and QR codes is handled by the Equinox ISP-PRO programming system.

When application design is completed, a firmware image hex file
(e.g. SwitchOnOff_slave_enhanced_232_OTA_ZW050x_US_TI_WITH_BOOTLOADER.hex) and a production control file (e.g. SwitchOnOff_ZwSmartStart.csv) are handed over to the production facility.

The firmware image is built using the Z-Wave software toolchain (mk.bat).

To generate the production control file, do the following:

Copy the QrCodeEncoder.xlsm QR code encoding utility to the folder where the hex file resides
(e.g. ProductPlus\SwitchOnOff\Build) and open the Excel sheet.
Do not touch any fields marked with gray shading.
If prompted, click the <Enable Content> link in the top of the page.

Populate the fields of the Excel sheet according to the instructions found in chapter 7. DSK and UUID values are not needed and will be ignored by the encoding utility as these values are generated for each unique device programmed by the Equinox ISP-PRO programming system.

Each time a field is updated, the encoding utility updates the file “ZwSmartStart.csv” found in the same folder as the encoding utility. The file may be renamed before distributed together with the firmware image file

The contents of a sample production control file are shown below.

```
DESCRIPTION,This file contains the production control string for a Z-Wave Smart Start product
RELEASEDATE,20171009-1533
```

```
AUTHOR,abr
QRCODE,9001%CHECKSUM%129%DSK%00100409728672022000000000030000201025
LABELCONTROL,LABEL_QR_PIN.txt
LABELCONTROL,LABEL_QR_DSK.txt
MANUFACTURERNAME,Silicon Labs
PRODUCTNAME,SwitchOnOff
PRODUCTVERSION,4.1
```

The format of the production control file is defined in [3].

9 PREPARING PC CONTROLLER FOR SMART START INCLUSION

The PC Controller [6] currently has no support for scanning QR codes, neither support for extracting the DSK from a Complete QR Code String.

To prepare a Smart Start inclusion, copy the DSK string ("xxxxx-xxxxx-xxxxx-..") to the PC Controller "DSK" field in the "Smart Start" tab.

10 PERFORMING THE SMART START INCLUSION

Disconnect and re-connect the USB and DC power connections to the ZDP03A programmer with the ZDB5101 board mounted.

If power cycled in a stand-alone fashion, the ZDB5101 board handles the power-on reset event correctly. Resetting a ZDP03A programmer with a ZDB5101 board mounted does not trigger a power-on reset event in the ZDB5101.

It can be seen in the Zniffer that the sample application starts transmitting Network-Wide Inclusion requests with command types "Smart Start Inclusion Prime" (0x27) and "Smart Start Inclusion Request" (0x28).

The controller responds by initiating the actual inclusion.

Once S2 bootstrapping is completed, the controller shows the assigned NodeID next to the DSK in the controller's node list.

11 POWER CYCLING AN ALREADY INCLUDED NODE

Disconnect and re-connect the USB and DC power connections to the ZDP03A programmer with the ZDB5101 board mounted.

It can be seen in the Zniffer that the sample application transmits a Network-Wide Inclusion request with command type “INIF” (0x26).

The PC Controller currently does not respond to the INIF but future versions will warn the user that this node, while its DSK is recognized, cannot be added to the controller’s network before the node has been reset.

REFERENCES

[1] QrCodeEncoder.xlsm

[2] QrCodeDecoder.xlsm

[3] INS13975 Production Control File format

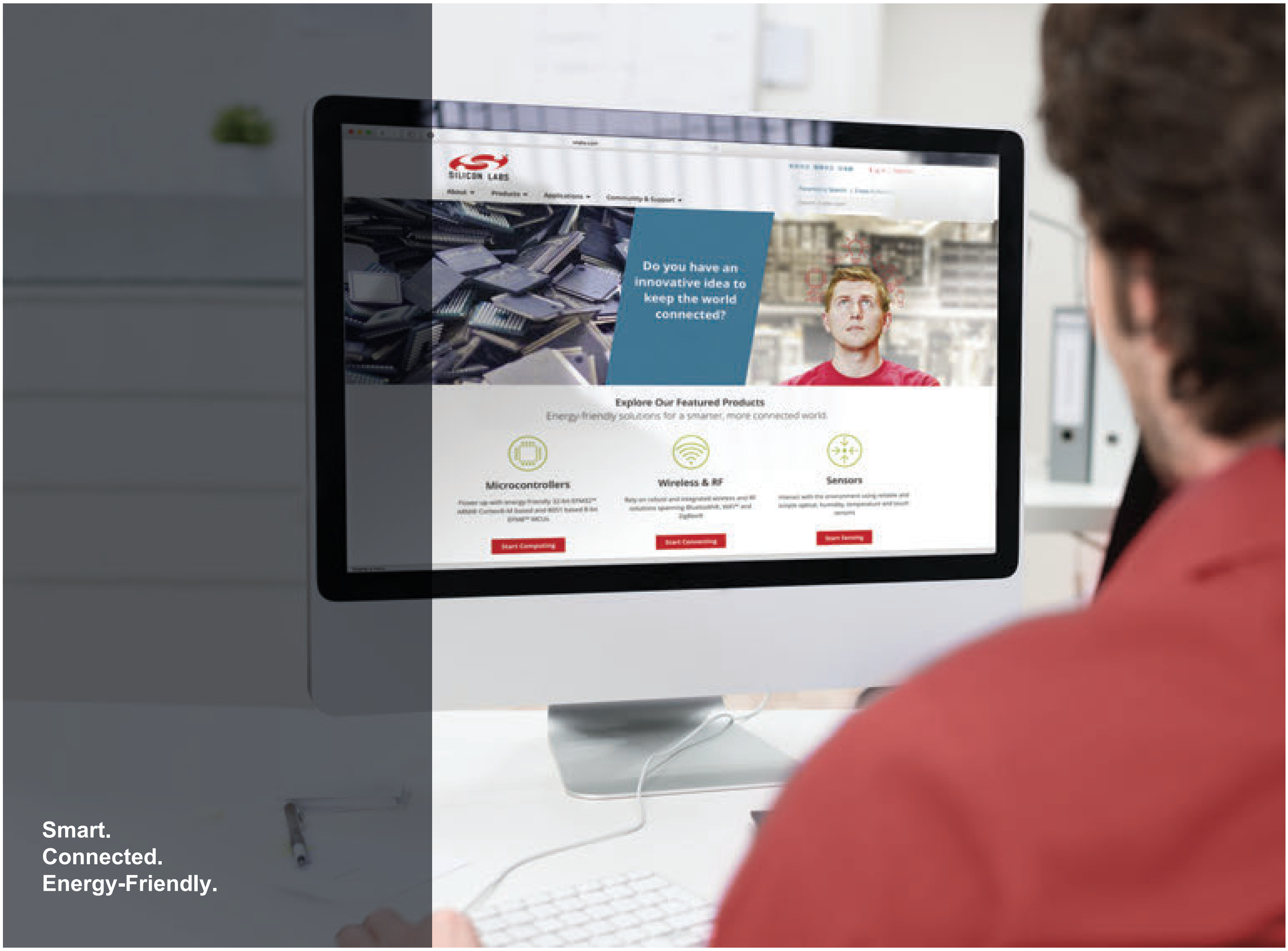
[4] SDS13937 Node Provisioning QR Code Format

[5] SDS13944 Provisioning Information Type Registry

[6] INS10679 Z-Wave Programmer User Guide

[7] INS13114 Z-Wave PC Controller User Guide

[8] SDS13783 Z-Wave Transport-Encapsulation Command Class Specification


SILICON LABS
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world.
Microcontrollers
Wireless & RF
Sensors
Smart.
Connected.
Energy-Friendly.

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com