User Manual

DA14585 Getting Started Guide with the IoT Multi Sensor Development Kit

UM-B-102

The focus of this User Manual is to easily introduce the IoT Multi sensor Kit. This is a reference design which include IoT Sensors Reference Application. This reference design integrates a number of sensors and provides to the user a sensor fusion experience using the IoT Sensors Android/iOS Application.

Contents

Contents 2
Figures 3
Tables 3
1 Abstract 4
2 Terms and Definitions 4
3 Introduction 5
3.1 How long should it take? 5
3.2 Block Diagram 5
3.3 Key Features 6
4 Kit Content 6
5 System Requirements 7
6 Hardware Description 7
6.1 PCB and Board Layout: 7
6.2 Sensors Overview 8
6.2.1 Environmental Sensor 8
6.2.2 Motion Sensor: Accelerometer/Gyroscope 9
6.2.3 Audio Sensor: Microphone 9
6.2.4 Electronic Compass (Magnetometer) 10
6.2.5 Barometric Pressure Sensor 10
6.2.6 Optical Sensor: Ambient Light and IR Proximity 10
6.3 Buttons and LEDs 10
6.4 NOR Flash Memory 11
6.5 Power Supply 11
7 Software Downloading and Programming 14
7.1 Drivers and Tools 14
7.1.1 SmartSnippets™ Installation 14
7.1.2 KEIL Installation 16
7.2 Development Software Source 16
8 Run the pre-loaded Demo 18
9 Build Your First IoT Application 21
9.1.1 How to Start Development 21
9.2 Flash Programming in MSK Applications 22
9.3 Beacon Reference Applications 26
9.3.1 Building and Running the Example 26
10 Appendices 30
10.1 Appendix A: Memory Map 30
10.2 Appendix B: Enclosure 30
10.3 Appendix C: mkimage script steps 31
11 Revision History 33

Figures

Figure 1: DA14585 IoT Multi Sensor Kit 4
Figure 2: DA14585 IoT MSK Block Diagram 5
Figure 3: PCBA of a DA14585 IoT MSK: Top View 7
Figure 4: PCBA of DA14585 IoT MSK: Bottom View 8
Figure 5: Battery Connection 12
Figure 6: Connection between DA14585 IoT MSK and the CIB 12
Figure 7: Communication Interface Board Layout 13
Figure 8: SmartSnippets Studio Install Link 14
Figure 9: Automatically Install J-Link 15
Figure 10: Automatically Install J-Link 15
Figure 11: VCOM on the Debugger 16
Figure 12: IoT MSK Software SDK Directory Structure 17
Figure 13: Quick Start with IoT MSK: Flow Diagram 18
Figure 14: Cloud setting 19
Figure 15: Applications: Historical Data 19
Figure 16: IoT Sensors Reference Application State Machine 20
Figure 17: IoT Sensors Reference Application State Machine 21
Figure 18: mkimage Scripts and Application Location 23
Figure 19: Available mkimage Scripts 23
Figure 20: Flash Programming with SmartSnippets Studio 24
Figure 21: SPI Flash Programmer Bootable option 24
Figure 22: Smart Tag Application 25
Figure 23: About Flash Programming Procedure 25
Figure 24: Locating the Beacon 28
Figure 25: Connecting with the DA14585 Eddystone Beacon 29
Figure 26: EDDYSTONE-TLM Data 29
Figure 22: Analyzing a Flash Memory Image 30
Figure 23: DA14585 IoT MSK Enclosure: Bottom/Top View 30
Figure 18: Convert .hex to .bin 31
Figure 19: Create a Single Image from a Binary 31
Figure 20: Create Another Single Image from a Binary 32
Figure 21: Create a Multi-Image from the Two Single Images 32

Tables

Table 1: DA14585 IoT MSK Top Main Devices 7
Table 2: DA14585 IoT MSK Bottom Main Devices 8
Table 3: GPIOs for Buttons and LEDs 10
Table 4: GPIOs for the Flash Memory 11
Table 5: Communication Interface Board (CIB) 13
Table 6: Beacon Reference Applications 26
Table 7: Revision History 33

1 Abstract

Dialog Semiconductor has created the DA14585 IoT Multi Sensor Development Kit (MSK) to help IoT device designers and engineers accelerate building their development platforms for designing IoT applications and solutions. This guide is intended to be an easy introduction to the IoT MSK. No experience of programming microcontrollers or the cloud is required. It helps with the setup of the hardware development environment, installing the required software, and downloading and running an example application on the MSK.

Figure 1: DA14585 IoT Multi Sensor Kit

2 Terms and Definitions

BLE	Bluetooth Low Energy
COM	Communication Port
e-CO2	Equivalent CO2
GATT	Generic Attribute Profile
GPIO	General Purpose Input/Output
HW	Hardware
AQI	Air Quality Indoor
iOS	iPhone OS
IoT	Internet of Things
IR	Infrared
JTAG	Joint Test Action Group (test interface)
LED	Light Emitting Diode
MSK	Multi Sensor Development Kit
OS	Operating System
OTP	One Time Programmable
PC	Personal Computer
PCB	Printed Circuit Board
PDM	Pulse Density Modulation
SDK	Software Development Kit
SOC	System On Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SUOTA	Software Update over the Air
SW	Software
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VOC	Volatile Oxide Compound

3 Introduction

The IoT MSK is based on DA14585 (a SmartBond™ Bluetooth Low Energy SoC) and a number of motion and environmental sensors. The DA14585 SoC is an optimized version of DA14580, offering a reduced boot time and supporting up to eight connections. It has a fully integrated radio transceiver and baseband processor for Bluetooth Low Energy. It can be used as a standalone application processor or as a data pump in hosted systems.

The IoT MSK board embeds a 2-Mbit QSPI Flash memory that can be used to store the downloaded software images over the SUOTA (Software Update Over The Air) profile and can also serve as storage for the second bootloader. Programming the IoT MSK is easy. There are two user manuals focusing on Software and Hardware details respectively.

Users can build a rich cloud application with the data from this IoT MSK in just a few steps. In addition, the cloud applications provided by Dialog can be used to monitor the data from the IoT MSK sensors and program IFTTT events.

NOTE
The key aspects of the hardware/software of the IoT MSK are explained in detail in this user manual. A Quick Start Guide is also available.

3.1 How long should it take?

This tutorial requires **30-40 minutes** to complete. For more information the user may consult the accompanied documentation these are mentioned as **for further reading**.

3.2 Block Diagram

Figure 2: DA14585 IoT MSK Block Diagram

3.3 Key Features

- Highly integrated DA14585 Bluetooth® Smart SoC from Dialog Semiconductor
- Standalone module
- Low cost due to printed antenna
- Low cost PCB
- Combined sensors
  - Audio
    - Microphone with single-bit PDM output
  - Gas and Environmental Sensor
    - Temperature
    - Humidity
    - Pressure
    - Air quality (b-VOC and AQI)
  - Motion Sensor
    - Combined accelerometer/gyroscope sensor unit
  - Magneto Sensor
  - Optical Sensor
  - Ambient Light Sensor and Infrared proximity
- Access to processor via JTAG and UART from the enclosure
- Programmable RF power up to +9.3 dBm
- Three LED indicators
- General purpose push button
- Expansion slots
- Powered by two low cost AAA alkaline batteries

4 Kit Content

This section describes the required hardware and software to start using the IoT MSK.

The IoT MSK can be ordered via various distributors with Digikey or Mouser.

Inside the IoT MSK package you will find:

- IoT Multi Sensor Kit board
- Programming interface board
- Mini USB cable
- Quick Start Guide

Download the development SW source. The Dialog IoT Sensors Mobile Application can be downloaded from App Store or Google Play. The schematics, PCB, Gerber files, Alegro files, and bill of materials (BOM) are available for download through this link.

5 System Requirements

The IoT MSK is programmed with a preloaded demo. To run it, you need a central device (smartphone or tablet) with 4.3 (minimum) or 8.0 (minimum) operating systems and BLE technology 4.0 (minimum).

To start developing IoT applications, you need Windows™ Operating System, (ver. 7 or higher) and KEIL development environment.

6 Hardware Description

This section gives an overview of the design architecture of the IoT MSK. **For further reading** more details are provided in UM-B-095.

6.1 PCB and Board Layout:

The top view layout of the IoT MSK is shown in **Figure 3**. The IoT MSK Enclosure Bottom/Top view is given in **Appendix B: Enclosure.**

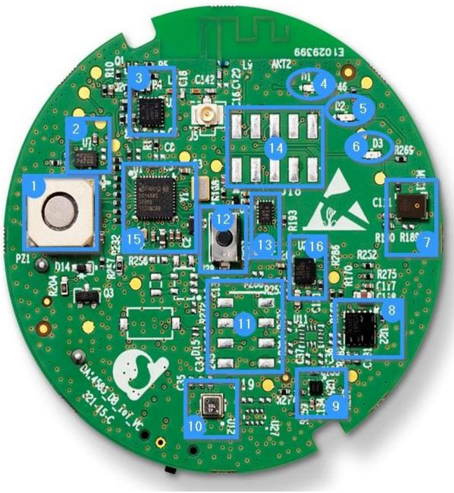

Figure 3: PCBA of a DA14585 IoT MSK: Top View

Table 1: DA14585 IoT MSK Top Main Devices

Reference	Device Name	Description
1	CSS-I4B20-SMT	Magnetic buzzer transducer from CUI INC.
2	U7: MX25R2035FZUIL0	Serial NOR Flash memory
3	U2: SKY66111-11	Power Amplifier from Skyworks
4	D1: Yellow LED	LED

5	D2: Red LED	LED
6	D3: Green Led	LED
7	SPK0838HT4H-B	Digital microphone with a single-bit PDM output from Knowles
8	U22: VCNL4010	Ambient light and IR proximity sensor from Vishay
9	U25: AK09915C	Magnetic sensor from Asahi Kasei
10	U12: BME680	Environmental and Gas Sensor from Bosch
11	J19: 8 Expansion slot	Connection slots to connect additional peripheral and sensors modules
12	User Push Button	General purpose push button
13	U21: FXL6408UMX	I²C controlled GPO expander from Fairchild
14	J18: 10 Expansion slot	Connection slots to connect additional peripheral and sensors modules
15	U1: DA14585	Bluetooth Smart SOC
16	U24: ICM42605	Accelerometer/Gyroscope Sensor from TDK Invensense

The bottom view of the PCBA of DA14585 IoT MSK is shown in **Figure 4**.

Figure 4: PCBA of DA14585 IoT MSK: Bottom View

Table 2: DA14585 IoT MSK Bottom Main Devices

Reference	Device Name	Description
1	Port	Debugging Connector
2	Battery	2x AAA Battery Holder
3	Switch	Power ON/OFF switch

6.2 Sensors Overview

The IoT MSK includes an accelerometer/gyro sensor, digital microphone, Gas (CO2) sensor, and an infrared proximity combined with an ambient light sensor in a single package. These sensors which can be accessed from the IoT MSK over the I²C, SPI, and PDM (Audio) interfaces.

6.2.1 Environmental Sensor

The DA14585 IoT MSK employs the **BME680** from Bosch Sensortec to detect environmental changes such as temperature, humidity, atmospheric pressure, and e-CO2. This highly compacted sensor is suitable for monitoring indoor air quality and can detect air contamination from paint, furniture, garbage, and others, using volatile oxide compound (VOC) levels. From the VOC readings, two air quality parameters can be displayed using smart algorithms: the indoor air quality index (IAQ) and the e-CO2. This sensor is connected to DA14585 via an I^2C interface.

6.2.2 Motion Sensor: Accelerometer/Gyroscope

The DA14585 IoT MSK employs the **ICM42605 motion sensor** from TDK Invensense that combines a 3-axis gyroscope and a 3-axis accelerometer with the following features:

- user-programmable interrupts
- wake-on-motion interrupt for low power operation of applications processor

The ICM42605 module is connected to DA14585 via an SPI interface which supports speeds up to 24 MHz.

NOTE
Note: In full operation mode with the accelerometer and gyroscope enabled, the current consumption is typically 0.72 mA. This drops to 11 μA in sleep mode.

NOTE
Note: For additional flexibility, the DA14585 IoT MSK is equipped with an additional PCB footprint of an alternative accelerometer/gyroscope sensor: BMI160. The ICM42605 should be unsoldered before using BMI160.

6.2.3 Audio Sensor: Microphone

The **SPK0838HT4H-B** from Knowles is a miniature, high-performance, low-power, and top-port silicon digital microphone with a single-bit PDM output. Due to its high power consumption in sleep mode, it is supplied via a dedicated GPO from the GPIO expander.

Warning
The microphone is not supported by the software reference applications provided with the DA14585 IoT MSK.

6.2.4 Electronic Compass (Magnetometer)

The DA14585 IoT MSK employs an electronic compass (magnetometer) sensor from Asahi Kasei, the **AK09915C**. It incorporates:

- a magnetic sensor for detecting terrestrial magnetism in the X-axis, Y-axis, and Z-axis
- a sensor driving circuit
- a signal amplifier chain
- an arithmetic circuit for processing signals from each sensor off-loading the main processing unit
- self-test function

The magnetic sensor is connected to the DA14585 via an SPI interface.

6.2.5 Barometric Pressure Sensor

The DA14585 IoT MSK employs a high-accuracy, low-power, and waterproof barometric pressure sensor from TDK InvenSense, **ICP10100**, for atmospheric pressure detection. This barometric pressure sensor is connected to DA14585 via an I^{2}C interface.

Warning
This sensor is not mounted on this reference design and is not supported by the software reference applications provided with the DA14585 IoT MSK. Users wanting to use this sensor need to do the soldering themselves.

6.2.6 Optical Sensor: Ambient Light and IR Proximity

The DA14585 IoT MSK has an on-board ambient light and IR proximity sensor from Vishay, **VCNL4010**. This sensor is fully integrated as the IR LED emitter is included in the package. It is connected to DA14585 via the I^{2}C interface.

Potential applications include:

- display contrast/brightness control
- proximity switch for consumer electronics, display, and devices
- dimming control

6.3 Buttons and LEDs

The IoT MSK is equipped with a general purpose user push button and three LED indicators. GPIOS for Buttons and LEDs shows the GPIO (DA14585 GPIO and the GPIO expander) pin assignment.

Table 3: GPIOs for Buttons and LEDs

GPIO	Function
GPIO Expander GPO 0	Yellow LED
GPIO Expander GPO 2	Red LED
GPIO Expander GPO 4	Green LED
P1_3	Push button

Note
When the kit is powered, the Yellow LED will blink for 60 seconds. During this time, the device will advertise. After 60 seconds, the device enters sleep mode and will only start advertising again when movement is detected

Note
The user push button is active-low and is de-bounced by an RC filter with a time constant of about 2 ms.

6.4 NOR Flash Memory

The DA14585 IoT MSK uses an external Serial NOR Flash memory to mirror its contents to RAM and execute the content. The Flash memory type is **MX25R2035FZUIL0**.

- 2-Mbit QSPI Flash memory, operated in single I/O mode.
- Operating voltage: 1.65 V to 3.6 V for read, erase, and program operations.
- 8USON package.

Table 4: GPIOs for the Flash Memory

GPIO	Function
P0_0 (SPI_CLK)	SPI clock
P0_3 (SPI_CS)	SPI chip select
P0_6 (SPI_MOSI)	SPI_MOSI
P0_5 (SPI_MISO)	SPI_MISO

Note
A pull-up resistor has been added in series with the chip select (CS) pin. This allows the CS pin to follow the voltage applied to the VCC pin during power-up and power-down which keeps the device not selected.

6.5 Power Supply

By default, IoT MSK is powered by two **AAA batteries** in the battery holder (BT1) which supply a 3 V voltage as shown in **Figure 5**. Another option is to use a **JTAG** supply. The two-position ON/OFF switch (SW2) is used to select between these options.

Figure 5: Battery Connection

If the IoT MSK is powered using JTAG:

1. Plug in the USB cable to the micro-USB header on the communication interface board (CIB). The other side of the USB cable can be connected to a PC.
2. Set the MSK side switch to OFF to cut off the battery and turn the CIB switch SW2 to ON to provide power to the MSK.
3. Connect an IDC-10 cable to the 1.27 mm pitch header (10) on the CIB. Connect the other end of the IDC-10 cable to the debugging port on the bottom of the DA14585 IoT MSK.

The connection between the CIB and DA14585 IoT MSK is shown in **Figure 6**.

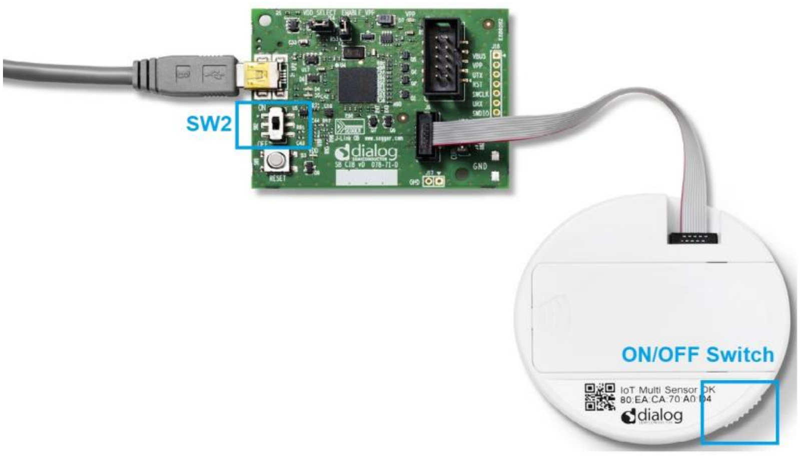

Figure 6: Connection between DA14585 IoT MSK and the CIB

The CIB implements USB-to-JTAG and USB-to-UART functions as shown in **Figure 7**.

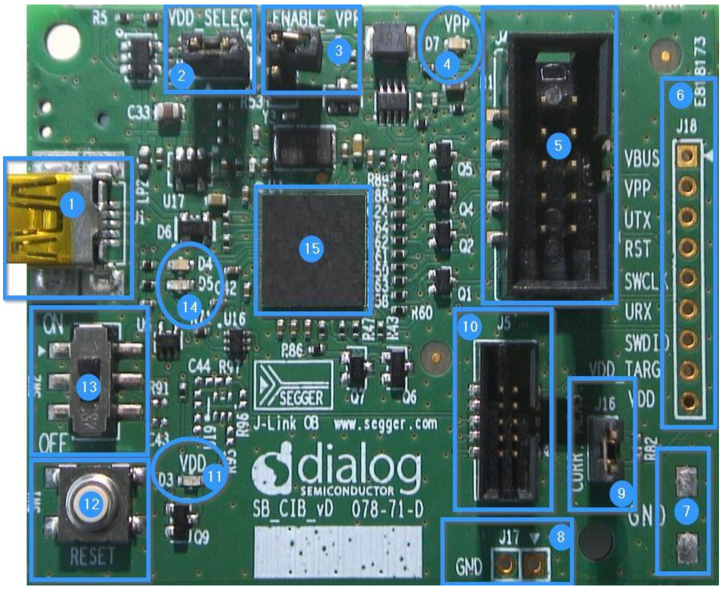

Figure 7: Communication Interface Board Layout

Table 5: Communication Interface Board (CIB)

Reference	Description
1	Mini USB Connector
2	VDD select, 1.8 V or 3 V
3	VPP enable (6.8 V)
4	VPP LED indicator
5	Target board connection header (2.54 mm pitch)
6	Output signals
7, 8	GND support points
9	Current measurement point
10	Target board connection header (1.27 mm pitch)
11	VDD LED indicator
12	Reset button
13	VDD ON/OFF switch
14	MCU LED indicator
15	MCU with Segger license

Note
For further reading more information about the CIB, refer to Communication Interface Board User Manual, UM-B-065

7 Software Downloading and Programming

This section contains reference information about the software required for downloading and programming.

7.1 Drivers and Tools

7.1.1 SmartSnippets™ Installation

This section describes the installation of SmartSnippets Studio. **For further reading** the installation procedure is described in detail in UM-B-057 SmartSnippets Studio User Manual.

A summary of the steps is given here.

1- Download the latest version of SmartSnippets™ Studio from Software and tools, as shown in **Figure 8**

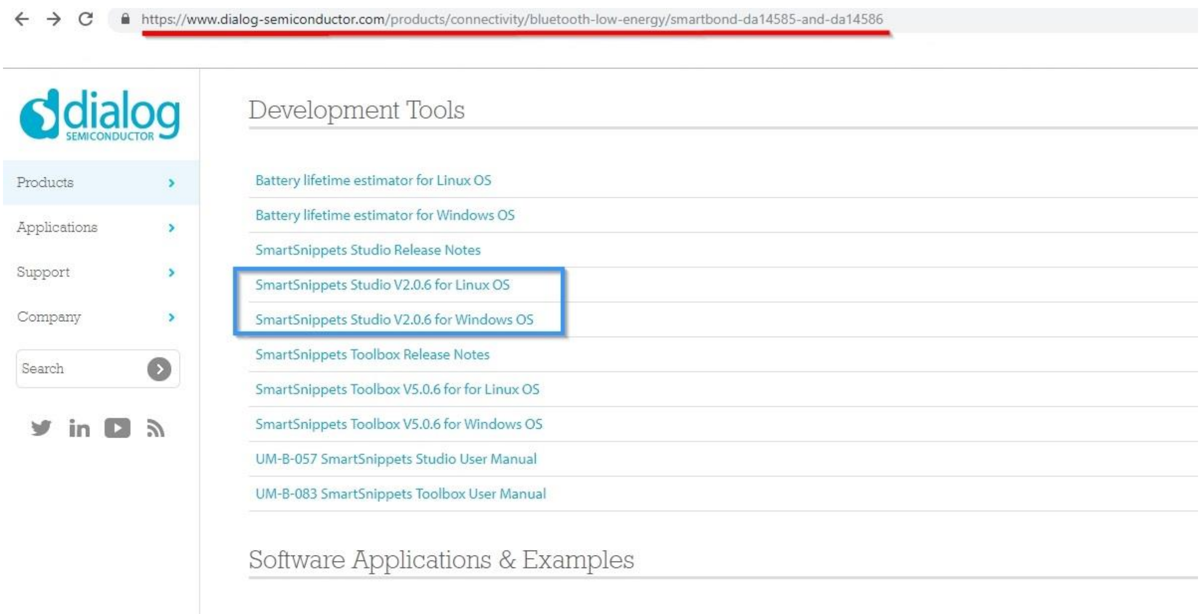

Figure 8: SmartSnippets Studio Install Link

Note
• The SmartSnippets™ version should be 2.0.6 and above if you wishes to use Eclipse/GCC. • Registration is required in order to download the SmartSnippets™.

2- Run the SmartSnippets™ Studio installer (.msi). Several of the required tools are automatically installed, others need to be manually downloaded and installed.

3- Select to install the latest version of SEGGER J-Link GDB server and click **Next**.

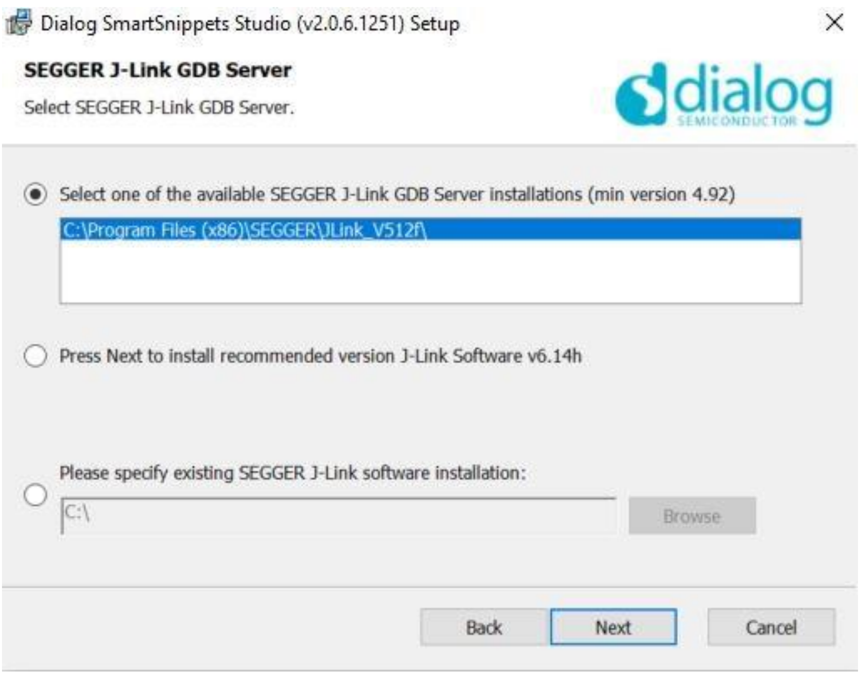

Figure 9: Automatically Install J-Link

4- Select the destination folder for the SmartSnippets™ Studio and click **Next**

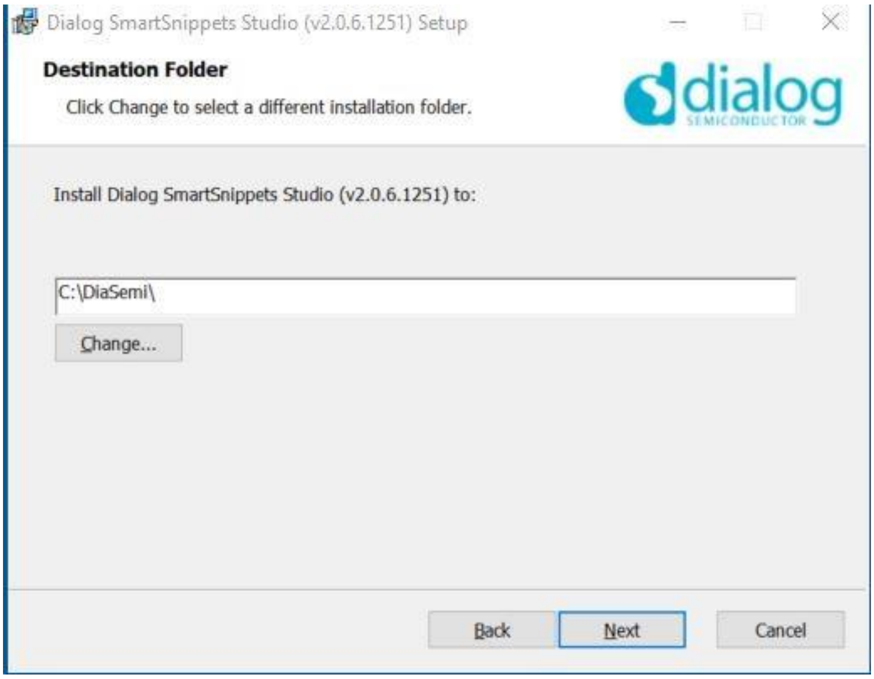

Figure 10: Automatically Install J-Link

The SmartSnippets™ Studio is Now installed.

Note
The communication interface board (CIB) includes an MCU embedded **J-Link debugger**, see **Figure 7**. This ensures the USB to JTAG function by loading the software from Segger to the internal ROM. This debugger provides also a virtual COM port to the PC/laptop as shown in **Figure 11.**

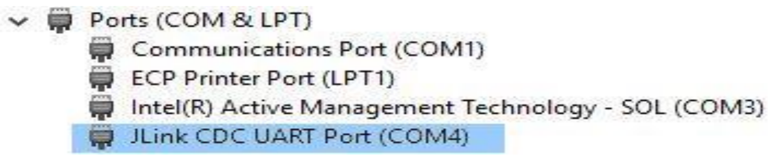

Figure 11: VCOM on the Debugger

7.1.2 KEIL Installation

KEIL µVision IDE must be downloaded and installed separately, registration is required.

1. Download and install the Keil tools from https://www.keil.com/demo/eval/arm.htm.

Note
• The Keil development tools can be run as a Free/Lite version without a product license. (**File > License Management**). The Free/Lite version offers 32KB code limitation and may be used for SmartTag and Beacon applications. Because of this limitation the Keil environment can't be used for the IoT Sensors application, if the user does not own µVision Keil License he may choose and activate an evaluation License that offers full functionality for limited time. • The recommended µVision version is v5.23.0.0.

Note
Note: Need help? Please contact Dialog BLE Software Forum.

7.2 Development Software Source

The directory structure of the IoT MSK Software SDK is shown in **Figure 12**. The IoT MSK application software runs on multi-sensor development kit design. The software firmware package contains five reference applications:

- The IoT sensors application
- The proximity tag application (SmartTag)
- Three beacon applications

The DA14585 IoT Multi Sensor Development SW Source can be downloaded from the customer support web page.

Note
For further reading additional information about the software architecture, see Development Kit Developer's Guide UM-B-101. This document describes the architecture and implementation details of DA14585 IoT MSK reference design and the supporting applications.

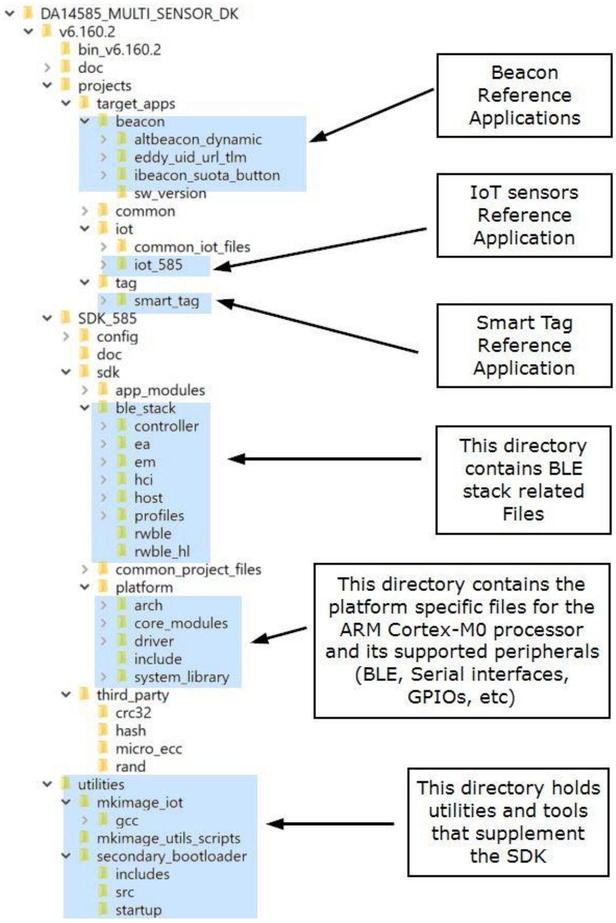

Figure 12: IoT MSK Software SDK Directory Structure

8 Run the pre-loaded Demo

This section explains how the user can build, program, **Figure 13** shows how to run the pre-loaded demo. These steps are already described in the Quick Start which is included in IoT MSK box.

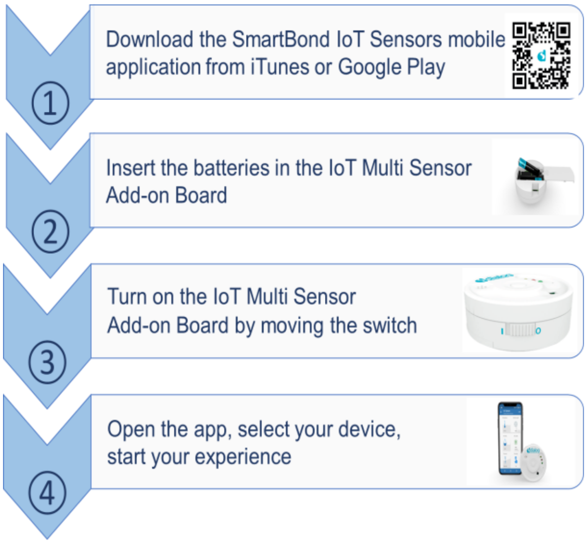

Figure 13: Quick Start with IoT MSK: Flow Diagram

As an option, user may enable the cloud functionality **Figure 14** (Internet availability is required). The IoT MSK Cloud feature allows data that are collected by the sensors on the IoT MSK to be uploaded to the cloud. The data on the cloud can be used in several ways:

- View historical data for a range of time. Example: View temperature logged over the 3 last days. Refer to **Figure 15.**
- Set triggers to receive E-mails when sensor data meet a condition. Example: If brightness is below 15 lumens send me an E-mail
- Control the IoT MSK Led when cloud conditions are met. Example: If temperature in Las Vegas is less than 20 Celsius turn on the Led.
- Play 3D Game. The user can use the IoT MSK as a joystick and play a 3D Game online.
- Setup IFTTT scenarios what can be triggered by the IOT sensor data or button press.
- Use Amazon Alexa to control the IOT device Led.

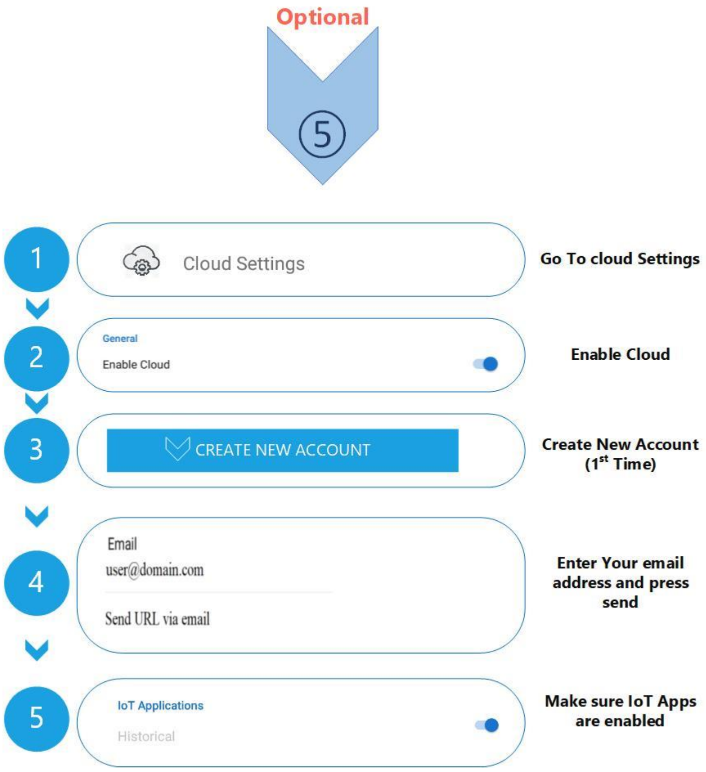

Figure 14: Cloud setting

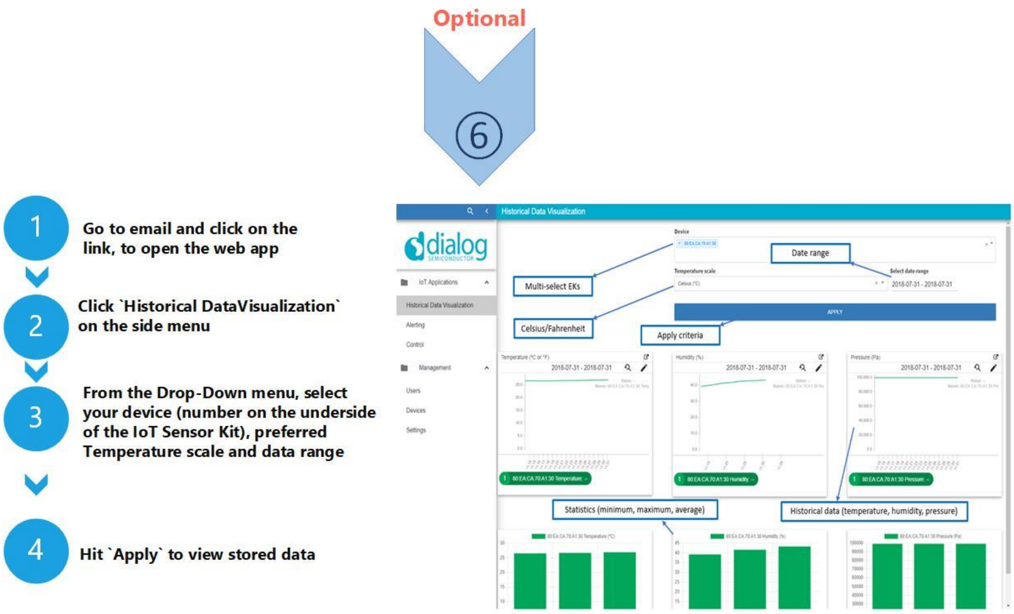

Figure 15: Applications: Historical Data

Upon successful connection to the IoT MSK application, you can navigate from the side menu to the following items:

- Environmental sensors
- IMU sensors
- Sensor Fusion 3D
- Cloud specific settings
- Configuration settings
- Information
- Disclaimer
- Magnetometer status

The IoT MSK advertises for 60 seconds (advertise timeout) before it goes to sleep. During this time the yellow LED will blink. The IoT MSK advertises for another 60 seconds when it detects movement and is paired again with the app. Error! Reference source not found. shows the application state machine.

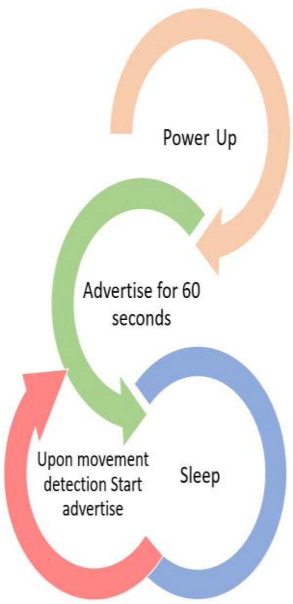

Figure 16: IoT Sensors Reference Application State Machine

See Also: For Further reading
+ For more details about advanced software features, see Section 5: Development Kit Developer's Guide UM-B-101. + For more details about the DA14585 advertising concept, see the Advertising Concepts tutorial.

9 Build Your First IoT Application

The user can build, program, and run a simple reference application on the IoT MSK. Make sure that you have all required tools installed as described in **Section 7**. The IoT MSK includes reference applications provided with preconfigured KEIL projects.

9.1.1 How to Start Development

After downloading the IoT MSK Software, the IoT Sensors Reference Application can be found in the `target_apps` directory.

1. Open the folder containing the IoT software files. This is the folder where you extracted the zip file `DA14585_IOTP_v6.160.x.yy.zip`
2. To open the project in Keil,

   in `<IoT_MSK_root_directory>/projects/target_apps/iot/iot_585/Keil_5`, double-click `iot585.uvproj`.

3. Refer to **Figure 17**: IoT Sensors Reference Application State Machine To build the KEIL project.

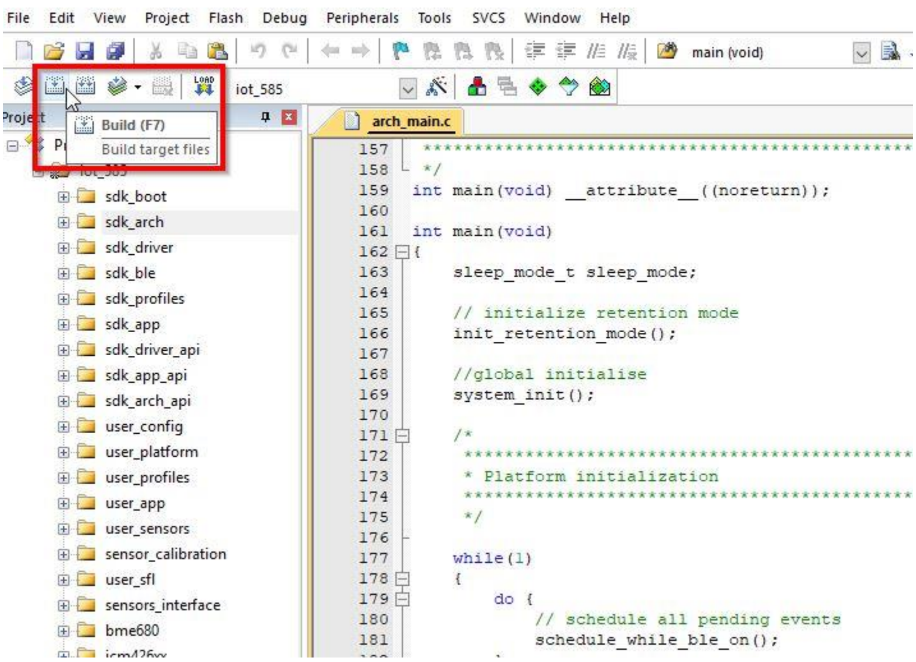

Figure 17: IoT Sensors Reference Application State Machine

Note
+**For further reading**, a group of compilation switches control the application's behavior. The most important switches are listed in **Table 5**: Configuration Parameters in Development Kit Developer's Guide UM-B-101. + The application is placed/executed in RAM. **For further reading** more details about RAM management, see How to change the RAM size tutorial. + For programming into Flash, see the next section.

Warning
+ You need a KEIL µVision license because the code size of most of the IoT reference applications exceeds 32 Kbytes. (Other provided projects can be compiled using the free version of Keil). **For further reading** See Section 8: Memory Footprint of Development Kit Developer's Guide UM-B-101. + The DA14585 IoT Sensors Reference Application can be compiled using the ARM GCC compiler. Dialog Semiconductor provides an example Eclipse project showing how to accomplish this. **For further reading** AN-B-064 describes the required steps to download and compile this project.

9.2 Flash Programming in MSK Applications

Before continuing, ensure that you have set up the hardware as shown in **Figure 6**.

This section describes the **mkimage** tool in detail. This is a tool to create an image to burn the SPI Flash (or any other non-volatile memory) according to the memory map already specified by the dual image bootloader. The mkimage tool is only needed to create the first full image when programming a single software application from the PC using a wired UART or SWD interface. In other cases, we can use **SUOTA** from a mobile phone to load any application (Eddystone UID-URL, for example) in external SPI Flash memory. **For further reading** refer to Tutorial 6.

The programmed devices come with the secondary bootloader already burned in the OTP memory. **For further reading** refer to the UM-B-012.

The SmartSnippets™ tool is used to program the bootloader in OTP and to program the product header and the dual images in SPI Flash. As stated in **Section 7,** we recommended installing the SmartSnippets™ tools to complete the software programming. This section provides more details and shows how to burn images in SPI Flash memory.

See Also: For Further reading
The secondary bootloader defines the memory map of the Flash consisting of two images and a program header, see **Appendix A: Memory Map.**

Note
The MSK reference application software implements the SUOTA receiver role. To support SUOTA, a dual image bootloader must be programmed in the OTP memory of the DA14585 and the corresponding non-volatile memory map must be applied in SPI Flash memory during the production phase.

These images are created by mkimage scripts or the mkimage application. **Figure 18** shows the location under the IoT MSK SDK.

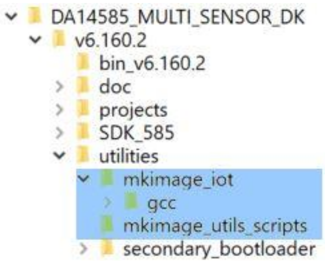

Figure 18: mkimage Scripts and Application Location

The available mkimage scripts are located (or should be placed) in :

`.../utilities/mkimage_utils_scripts` and are shown in **Figure 19**.

```
make_image_beacon.bat altbeacon_dynamic
make_image_iot.bat iot585
make_image_tag.bat smart_tag_585
make_all_images.bat
```

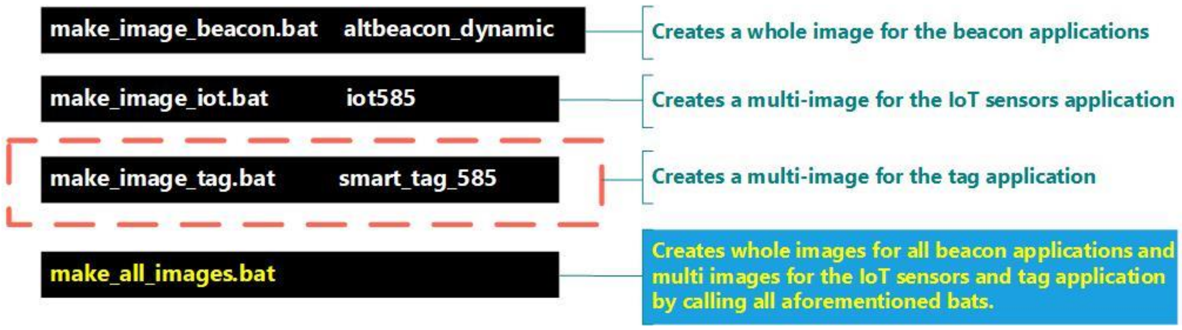

Figure 19: Available mkimage Scripts

Here after we will use the **Smart Tag application** as the compile example.

1- First you run the `make_image_tag.bat` script, you need to copy two files, in the same location as the mkimages scripts, as shown in **Figure 20.**

- The generated `.hex` file by you keil project : `smart_tag_585.hex`
- The SW Tag version `tag_sw_version.h` file found in :

  `projects/target_apps/tag/smart_tag/src/config`

```
make_image_tag.bat smart_tag_585
```

2- Then With SmartSnippets Studio, you can burn the generated `.bin` file (multi-image for tag application that we took as an example)

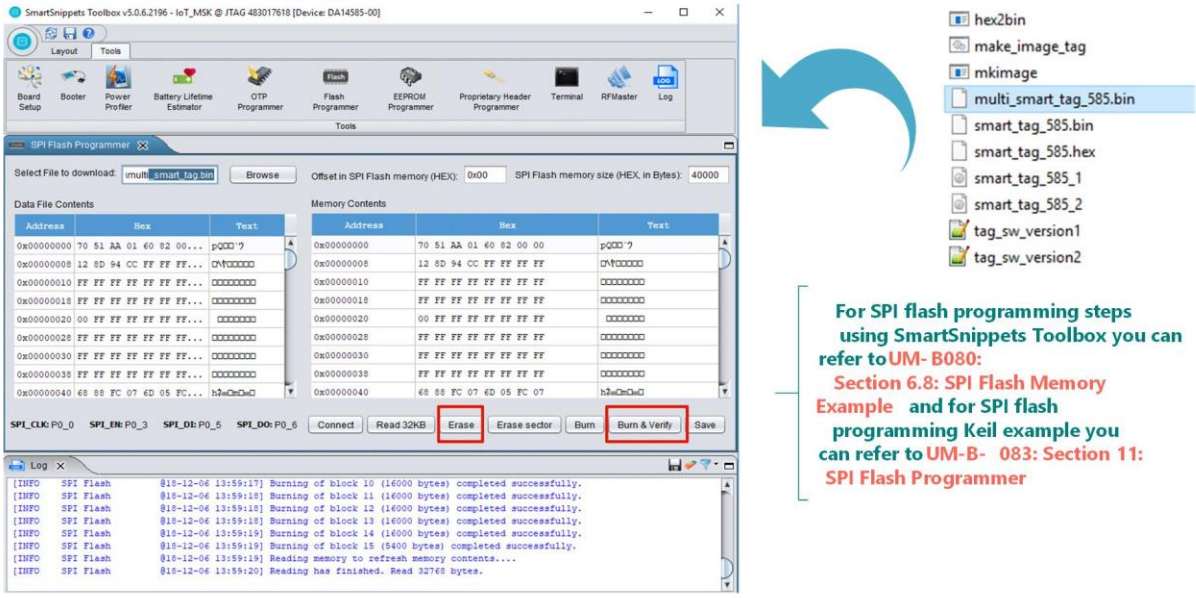

Figure 20: Flash Programming with SmartSnippets Studio

Note
• Erase: Erases the entire SPI Flash Memory • Burn & Verify: Adds a verification step after the burn process. After burning data to SPI Flash memory, it is verified that the contents of the memory are the same with the contents of the file that has been burned.

Warning
When trying to burn the .bin at SPI Flash Memory, You are presented with the option to make it bootable. You MUST NOT select the bootable option, because special header is added before the data and the data is written starting at the selected offset. Please refer to **Figure 21**

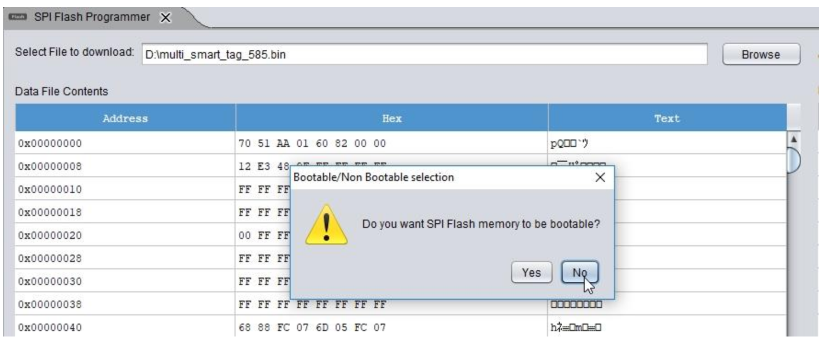

Figure 21: SPI Flash Programmer Bootable option

3- With The image is now burnt in flash and by pressing the Reset button on the CIB board, it will start working with the programmed application. Now you can see that the green Led blinks for Smart Tag Application. After 4 Minutes the Smart Tag stops advertising and enters continuous Deep Sleep mode. To restart advertising, you should Press Reset on the CIB board.

Figure 22: Smart Tag Application

Note
The make_image_tag.bat is executed in 4 steps as shown in **Figure 23** details is given in **Appendix C: mkimage script steps.**

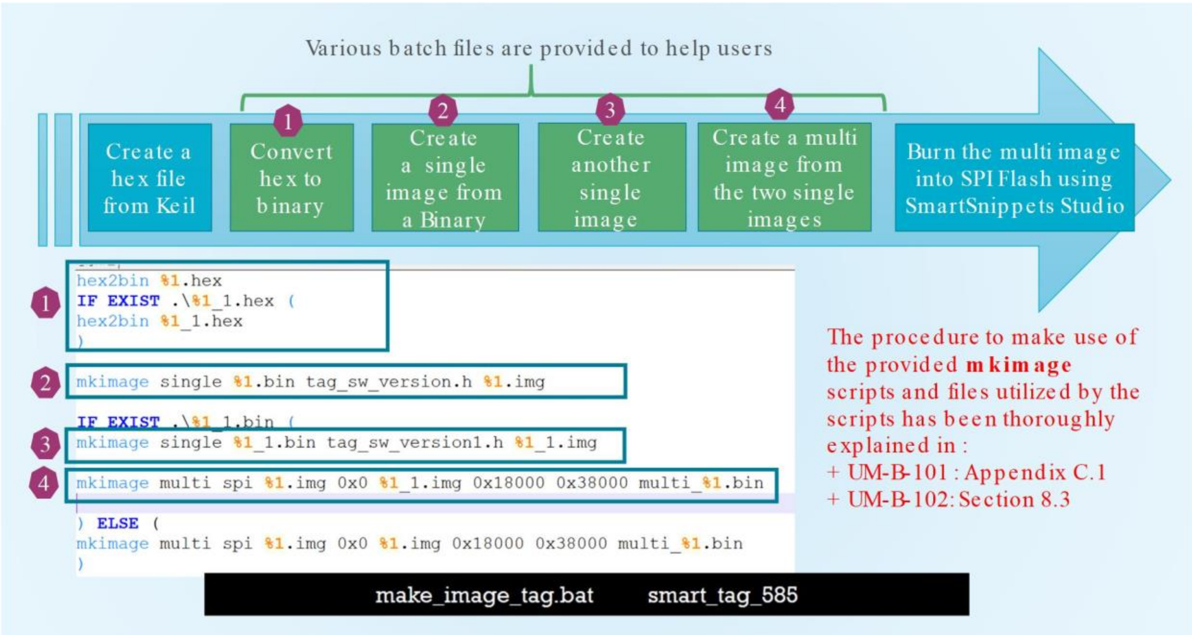

Figure 23: About Flash Programming Procedure

As shown in **Figure 19** , the mkimage tool has five different modes to create images.

- **single**: creates an `.img` file from the `.bin` file of the Keil project. This image contains the software version and the software version date.
- **multi**: creates a `.bin` file from the `.bin` file of the Keil project. This `.bin` file contains two images created by the **mkimage single** mode and a **product header** at the end of the file.
- **whole_img**: This mode is used to create a complete `.bin` file. This contains two alternative `.img` files, created by mkimage single mode, that are needed when using the **SUOTA** functionality, the `config_struct.cfg` file and the product header.
- **multi_no_suota**: This mode is used to create a whole `.img` file containing the `.bin` file of the Keil project preceded by the `config_struct.cfg` file. The image can be created for either an SPI Flash memory or an EEPROM Flash memory. The generated image will not include SUOTA functionality.
- **cfg**: This mode is used to create a `.cfg` file containing a device configuration struct preceded by its header. The device configuration struct header also contains a **4-byte CRC** which is calculated from the fields of the configuration struct. The application also checks a software version file and includes the version in the header of the corresponding field.

Note
The **whole_img** mode is only for beacons. It is similar to the **multi** mode with the addition of the beacon config struct.

9.3 Beacon Reference Applications

As shown in **Figure 12**, there are three different projects that demonstrate how connectable and non-connectable beacons can be used for various applications. These beacon examples use all the different beacon types and features supported by Dialog Semiconductor as shown in **Table 6**.

Table 6: Beacon Reference Applications

Beacon Application	Type
altbeacon_dynamic	Non-connectable
eddy_uid_url_tlm	Connectable
ibeacon_suota_button	Non-connectable

This section gives an overview of the **UID-URL Beacon** reference application design.

Section 6 of Development Kit Developer's Guide UM-B-101 describes what beacons are, what they can be used for, and how they are implemented within the BLE software stack of the DA14585 IoT MSK.

9.3.1 Building and Running the Example

To get started with Eddystone UID-URL example you need:

- The IoT MSK
- An Android/iOS mobile application. For an Android device you can use Locate Beacon.

After downloading the IoT MSK software, the Eddystone UID-URL Beacon Reference Application can be found in the `target_apps` directory.

1. Open the folder containing the IoT software files. This is the folder where you extracted the zip file.
2. To open the project in Keil, in

   `<IoT_MSK_root_directory>/projects/target_apps/beacon/eddy_uid_url_tlm/Keil_5`, double-click `eddy_uid_url_tlm.uvproj`.
3. In `user_config.h`, enable the `USE_EDDYSTONE_URL` flag and disable the `USE_EDDYSTONE_UID` flag as shown in enable the `USE_EDDYSTONE_URL` flag.

```
// Choose which Eddystone Mode to advertise
//#define USE_EDDYSTONE_UID
#define USE_EDDYSTONE_URL
```

The Eddystone UID-URL frame broadcasts a URL using a compressed encoding format. Once parsed and decompressed, the URL is directly usable by the client.

```
/// Default beacon configuration struct
struct user_beacon_config_tag user_default_beacon_config = {
   .uuid = { 0x58, 0x5C, 0xDE, 0x93, 0x1B, 0x01, 0x42, 0xCC, 0x9A, 0x13, /
/10-byte Namespace
           0x25, 0x00, 0x9B, 0xED, 0xC6, 0x5E },   //6-byte Instance
   .major_ALT_val1 = 0x0300,   //Major Value
   .minor_ALT_val2 = 0x0200,   //Minor Value
   .company_id = DIALOG_COMP_ID,   //Beacon company ID
   .adv_int = BEACON_ADVERTISING_INTERVAL,   //Advertising interval
   .power = 0xC5,         //Tx Power
   .beacon_type = EDDYSTONE_UID,
   .url_prefix = HTTPWWW,
   .url = { 0x0E, 'd', 'i', 'a', 's', 'e', 'm', 'i', DOTCOM },
   .TLM_version = 0x00,
   .TLM_used = 0x01
};
```

Note
Note: The `.url` field of the Beacon Configuration Struct contains the URL, preceded by its length incremented by 7 (<url_string>+7) and followed by the URL postfix (.com). The URL prefix (HTTPWWW) is stored in the previous field of the struct `.url_prefix`.

4. Build (you need a KEIL µVision license), download, and execute your project. For the hardware settings, see **Figure 6**.

The advertising string contains a encoded URL with a length ranging from 1 to 17 bytes. The Eddystone UID-URL Beacon reference application advertises an `EDDYSTONE-TLM` advertising string every

<defined number> EDDYSTONE-URL advertisements and then returns to advertising Eddystone-URL strings. The Eddystone UID-TLM packet contains information about the battery voltage and the temperature of the device, as well as how long the device has been powered on and the amount of advertising events it has executed. When connected to a central, the device provides four different GATT services: DISS and BASS which are official BLE GATT services, and two Dialog proprietary GATT services, env_data_ntf and device_config. The device_config and env_data_ntf services are described in Section 7.10 and 7.11 in Development Kit Developer's Guide UM-B-101.

Note
The default broadcasted link is www.diasemi.com which can be modified. The Eddystone protocol provides 17 bytes for the URL packet. If the URL is too long you can use a URL shortener.

The Locate Beacon application searches for available beacons and lists them:

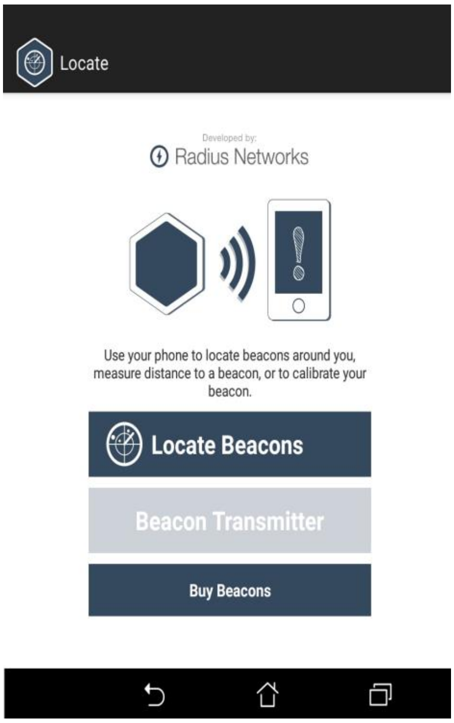

Figure 24: Locating the Beacon

Figure 25: Connecting with the DA14585 Eddystone Beacon

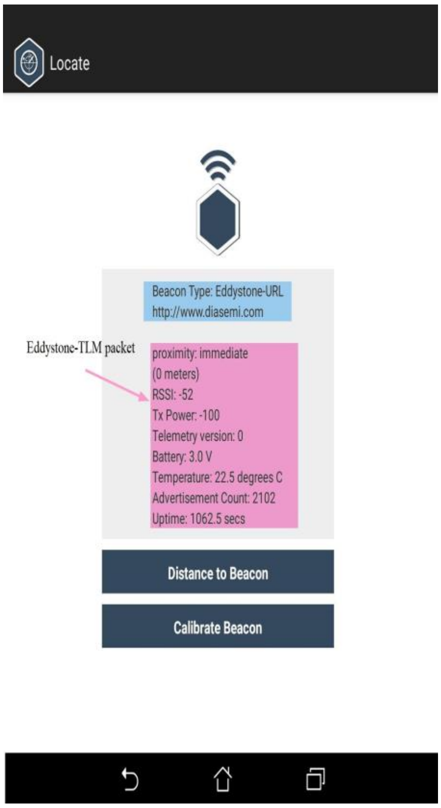

Figure 26: EDDYSTONE-TLM Data

For Further reading:

- About google eddystone.
- About Eddystone-url.

10 Appendices

10.1 Appendix A: Memory Map

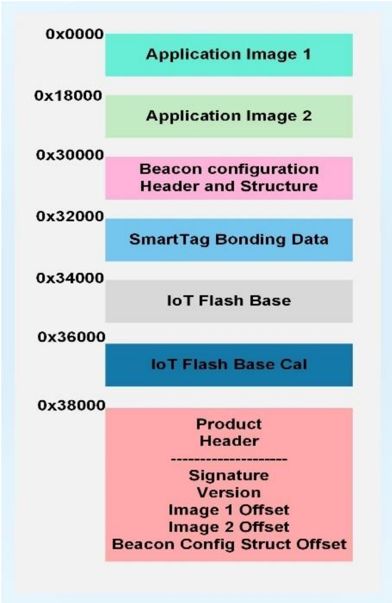

Figure 27: Analyzing a Flash Memory Image

10.2 Appendix B: Enclosure

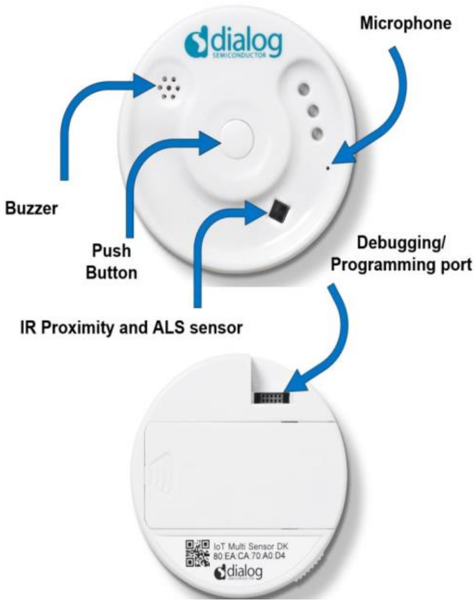

Figure 28: DA14585 IoT MSK Enclosure: Bottom/Top View

10.3 Appendix C: mkimage script steps

The steps in the script are:

1. Convert your `.hex` to `.bin`.

Figure 29: Convert .hex to .bin

2. Create a single image from a binary.

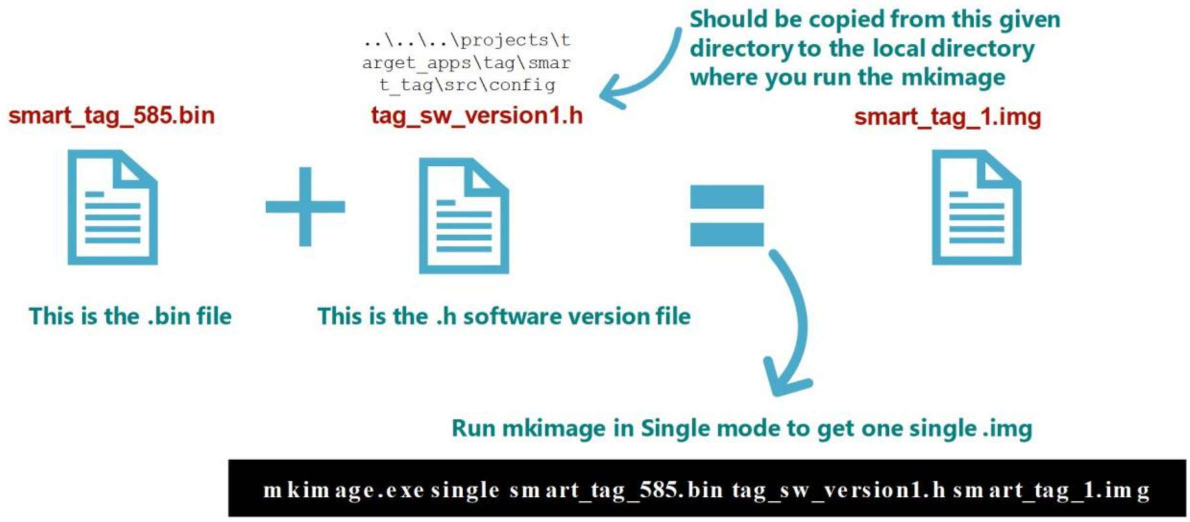

Figure 30: Create a Single Image from a Binary

3. Create another single image from a binary.

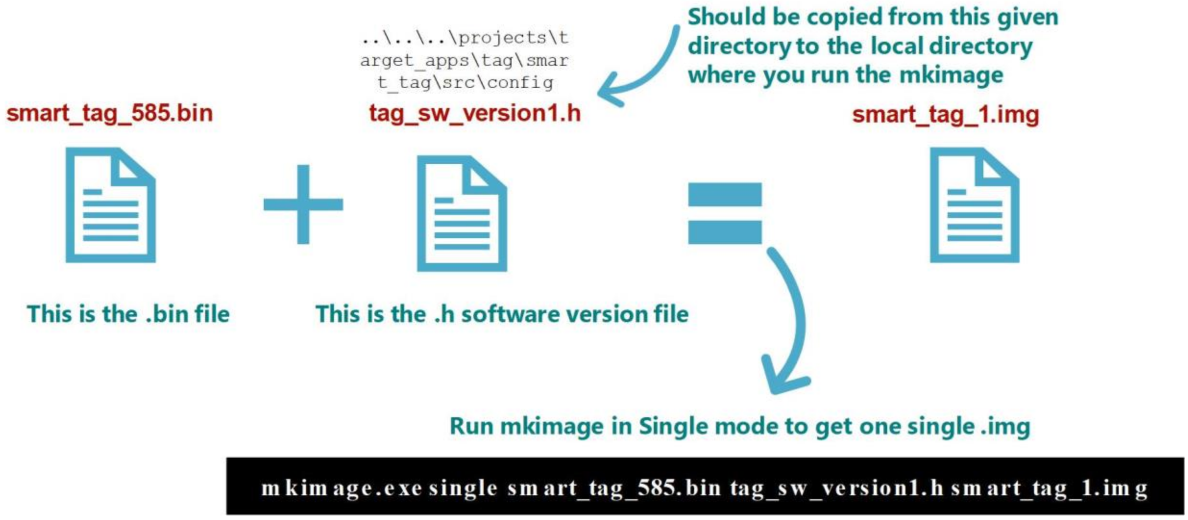

Figure 31: Create Another Single Image from a Binary

4. Create a multi-image from the single two images.

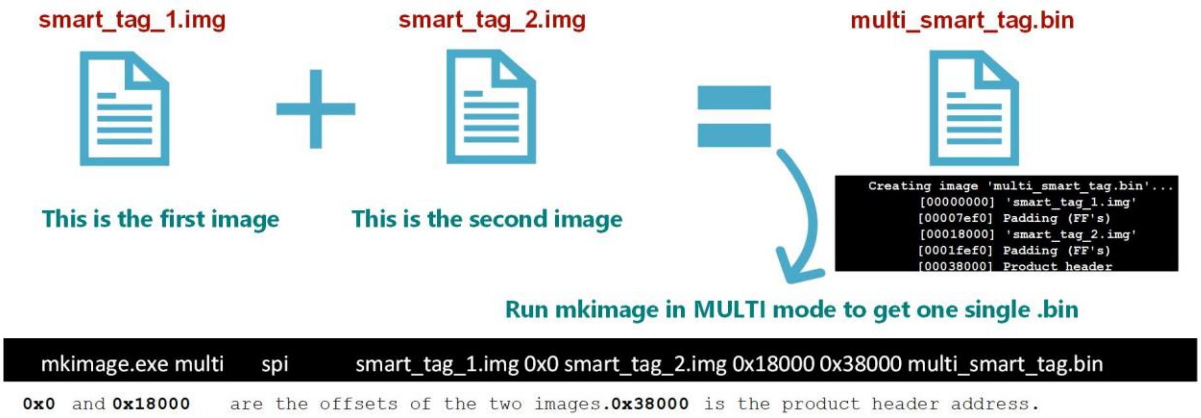

Figure 32: Create a Multi-Image from the Two Single Images

11 Revision History

Table 7: Revision History

Revision	Date	Description
1.0	10-Dec-2018	Initial public release version.
1.1	15-Feb-2019	- Add new section: **Run the pre-loaded Demo** - Add IoT cloud setting **Figure 14** and **Figure 15** - Add new **Appendix C: mkimage script steps** - Text revision

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

Disclaimer

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5769 5100

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)
Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)
Dialog Semiconductor China
Phone: +86 21 5424 9058