

User Manual

DA14585 IoT Multi Sensor Development Kit

UM-B-100

Abstract

IoT Multi Sensor Development Kit (MSK) based on DA14585 supports 15 Degrees of Freedom and includes five reference designs: IoT Sensors, Smart Tag, and three different types of beacons. The corresponding apps run on iOS/Android devices and the cloud services offers great flexibility to customers in product design. This document describes the architecture and the implementation details of DA14585 IoT MSK reference design and the supporting applications.

Contents

Abstract	1
Contents	2
Figures	4
Tables	5
1 Terms and Definitions	6
2 References	6
3 Introduction	7
4 IoT Sensors Reference Application	8
4.1 Description	8
4.2 IoT Sensors App in Android and iOS	8
4.2.1 Installing from Apple AppStore	8
4.2.2 Installing from Google PlayStore	9
4.3 Screens of Dialog IoT Sensors App	9
4.3.1 Scan Screen	9
4.3.2 Side Menu.....	10
4.3.2.1 Environmental Sensors	11
4.3.2.2 IMU Sensors	12
4.3.2.3 Sensor Fusion 3D Screen	13
4.3.3 Basic Settings Screen.....	14
4.3.3.1 Calibrate Proximity Sensor	15
4.3.4 Magnetometer Calibration Settings Screen.....	16
4.3.4.1 Magnetometer Calibration File	17
4.3.5 Magnetometer Auto-Calibration Procedure	17
4.3.5.1 Prerequisites for Magnetometer Calibration	18
4.3.5.2 Procedure of Magnetometer Calibration	18
4.3.5.3 Magnetometer Sensor and Calibration State Reporting.....	18
4.3.6 SmartFusion Coefficients Screen	19
4.3.7 Fast Accelerometer Calibration Screen.....	20
4.4 Air Quality Index.....	20
5 Smart Tag Reference Application	22
5.1 Introduction	22
5.2 SmartTags App in Android and iOS.....	22
5.2.1 Overview	22
5.2.2 Installation.....	23
5.2.3 Device List	23
5.2.4 Device Details.....	24
6 Beacon Reference Applications	28
6.1 Introduction	28
6.2 What is a Beacon?	28
6.3 Beacon Example	28
6.4 Finding Advertising Devices.....	30
7 IoT Cloud Applications	31

DA14585 IoT Multi Sensor Development Kit

7.1	Introduction	31
7.2	User and Device Management	31
7.2.1	Users.....	31
7.2.2	Devices	33
7.3	Sensor Data	34
7.3.1	Live Data.....	34
7.3.2	Historical Data	34
7.3.2.1	Cloud Application.....	34
7.3.2.2	Mobile Application.....	35
7.3.2.3	Rest API.....	36
7.4	Alerting	36
7.4.1	Cloud Application.....	36
7.4.2	Mobile Application.....	37
7.5	IoT Control Application.....	38
7.5.1	Cloud Application.....	38
7.5.2	Mobile Application.....	38
7.6	IoT Asset Tracking Application	39
7.6.1	Cloud Application.....	39
7.6.1.1	Asset Tracking.....	39
7.6.1.2	Asset Management.....	41
7.6.1.3	Gateway Management	42
7.6.1.4	Location Management	42
7.6.1.5	Supervisor Management	43
7.6.2	Mobile and Raspberry Pi Application.....	44
7.6.2.1	Mobile	44
7.6.2.2	Raspberry Pi.....	45
7.6.2.3	Quick Start-up Guide	46
7.7	3D Game	47
7.7.1	Cloud Application.....	47
7.8	Amazon Alexa.....	48
7.8.1	Skill Activation.....	48
7.8.2	How to Use	49
7.9	IFTTT.....	50
7.9.1	Create an IFTTT Applet.....	50
7.9.2	IFTTT Web Hook Key	55
7.9.3	Web App - IFTTT Settings.....	57
7.9.4	Mobile App: IFTTT Settings.....	57
Appendix A Troubleshooting		59
A.1	Magnetometer or Sensor Fusion Have Inconsistent Behaviors.....	59
A.2	Device Does Not Connect after Switching to a Different Firmware	59
A.3	Proximity Sensor is always On	59
A.4	Air Quality Index is not Working.....	59
A.5	Magnetometer is not Working	59
Appendix B Raspberry Pi for Data Gateway.....		60
B.1	Data Gateway Application.....	60

B.2 Configuration File for Asset Tracking and Data Gateway Application 60

Revision History 62

Figures

Figure 1: Internet of Things (IoT)..... 7

Figure 2: Installing Dialog IoT Sensors App from Apple AppStore 8

Figure 3: Installing Dialog IoT Sensors App from Google PlayStore 9

Figure 4: Scan screen of Dialog IoT Sensors App 9

Figure 5: Side Menu 10

Figure 6: Environmental Sensor Screen of Dialog IoT Sensors App 11

Figure 7: IMU Sensor Screen of Dialog IoT Sensors App..... 12

Figure 8: 3D Screen of Dialog IoT Sensors App 13

Figure 9: Basic settings of Dialog IoT Sensors App..... 14

Figure 10: Proximity Sensor Calibration Prompt 15

Figure 11: Calibration Settings of Dialog IoT Sensors App 16

Figure 12: Sensor Fusion Settings of Dialog IoT Sensors App..... 19

Figure 13: Accelerometer Calibration Settings of Dialog IoT Sensors App 20

Figure 14: SmartTags APP Icon..... 23

Figure 15: Installation of Dialog SmartTags App..... 23

Figure 16: Device List..... 24

Figure 17: Device Details 24

Figure 18: Distance Alert Window 25

Figure 19: How to Access the Side Menu 26

Figure 20: Device Information 26

Figure 21: Disclaimer..... 27

Figure 22: Seek & Find Screens 27

Figure 23: Beacon Protocol Logos 28

Figure 24: Bluetooth Low Energy Beacon..... 28

Figure 25: Description of the Exhibit on a Smartphone..... 29

Figure 26: Locate App Screenshot..... 30

Figure 27: Cloud Settings 32

Figure 28: Create New Account or Pairing with an Existing Account 32

Figure 29: User Management in Cloud Application..... 33

Figure 30: Device Management 33

Figure 31: Live Data 34

Figure 32: Historical Data: Cloud application 35

Figure 33: Historical Data: Mobile Application..... 35

Figure 34: Alerting: Cloud Application 37

Figure 35: Alerting: Mobile Application..... 37

Figure 36: Control: Cloud Application..... 38

Figure 37: Control: Mobile Application 39

Figure 38: Asset Tracking Table 40

Figure 39: Asset Tracker 40

Figure 40: Asset Tracker - Settings..... 41

Figure 41: Asset Tracker - Gateway settings 41

Figure 42: Asset Management 42

Figure 43: Gateway Management 42

Figure 44: Location Management..... 43

Figure 45: Supervisor Management 43

Figure 46: Dialog's IoT Sensors App, Asset Tracking Tab..... 44

Figure 47: Configuration File of Raspberry Pi for Asset Tracking 46

Figure 48: Configuration File of Raspberry Pi for Data Gateway 46

Figure 49: 3D Game IoT Application 47

Figure 50: 3D Demo 47

Figure 51: 3D Ball-Board Control Game 48

DA14585 IoT Multi Sensor Development Kit

Figure 52: Dialog Evaluation Kit Skill for Alexa 49

Figure 53: Alexa: Skill Activation Succeeded Screen 49

Figure 54: IFTTT Signup Page 50

Figure 55: New Applet 50

Figure 56: Make an IFTTT Applet 51

Figure 57: Make an IFTTT Applet Step 1 51

Figure 58: Make an IFTTT Applet Step 2 52

Figure 59: Make an IFTTT Applet, Complete Step 2 52

Figure 60: Make an IFTTT Applet, Create That 53

Figure 61: Make an IFTTT Applet, Step 3 53

Figure 62: Make an IFTTT Applet, Step 4 54

Figure 63: Make an IFTTT Applet Step 5 54

Figure 64: Make an IFTTT applet, Finish 55

Figure 65: IFTTT Web Hook Key, Step 1 55

Figure 66: IFTTT Web Hook Key Step 2 56

Figure 67: IFTTT Web Hook Key Step 3 56

Figure 68: Web App - IFTTT Settings 57

Figure 69: IFTTT Settings - Mobile App 58

Figure 70: Email When "Temperature" Occurred 58

Figure 71: Online Configuration Form 61

Tables

Table 1: Air Quality Index 21

Table 2: Example of Advertising Data from a Museum Beacon 29

1 Terms and Definitions

AQI	Air Quality Index
BD	Bluetooth Device
BLE	Bluetooth Low Energy
IMU	Inertial Measurement Unit
IoT	Internet of Things
MSK	Multi Sensor Development Kit
RAM	Random Access Memory
Report	Notification of Sensor Data and Control
RSSI	Received Signal Strength Indicator
SF	Sensor Fusion
SFL	Sensor Fusion Library
SI	Sensors Interface
SUOTA	Software Update over the Air
UART	Universal Asynchronous Receiver/Transmitter
VOC	Volatile Organic Compound

2 References

- [1] UM-B-080, DA14585/586 SDK 6 Software Developer's Guide, User Manual, Dialog Semiconductor
- [2] UM-B-079, DA14585/586 SDK 6 Software Platform Reference, User Manual, Dialog Semiconductor
- [3] RW-BLE-GAP-IS, GAP Interface Specification, Riviera Waves
- [4] UM-B-048, DA14585/DA14586 Getting Started Guide with the Basic Development Kit V2.0, User Manual, Dialog Semiconductor
- [5] DA14585, Datasheet v3.2, Dialog Semiconductor
- [6] UM-B-089, DA14585 Range Extender Reference Application, User Manual, Dialog Semiconductor
- [7] <https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers>
- [8] UM-B-065, Bluetooth® Smart Communication Interface Board, User Manual, Dialog Semiconductor
- [9] <https://github.com/google/eddystone>
- [10] <https://github.com/google/eddystone/tree/master/eddystone-uid>
- [11] <https://github.com/google/eddystone/tree/master/eddystone-url>
- [12] <https://github.com/google/eddystone/blob/master/eddystone-tlm/tlm-plain.md>
- [13] <https://github.com/AltBeacon/spec>
- [14] UM-B-095, DA14585 IoT Multi Sensor Development Kit Hardware Design, User Manual, Dialog Semiconductor
- [15] <https://www.bluetooth.com/specifications/gatt>
- [16] AN-B-001, DA14580/581/583 Booting from serial interfaces, Application Note, Dialog Semiconductor
- [17] BST-BME680-AN008, BSEC Integration Guide, Bosch Sensortec
- [18] UM-B-101, DA14585 IoT Multi Sensor Development Kit Developer's Guide, User Manual, Dialog Semiconductor

3 Introduction

With yearly shipments of more than 10 billion microcontrollers that all can exchange information locally or through the Internet, a huge variety of so called "intelligent devices" are enabled. These devices include motion sensors, pool pumps, gas/electric meters, streetlamps, and many more.

All these devices can be accessed over the Internet; thanks to the rapid increase in infrastructure coverage and Internet access. This evolution is often called the Internet of Things (IoT). Other names include Internet of Everything (IoE), Web of Things, Embedded Web, and Industry 4.0. The goal is to establish an Internet connection for the small "things" you carry with you or use in a factory, hospital, city, or at home (Figure 1). IoT industrial experts predict that there are going to be more than 50 billion connected devices within the next 10 years.

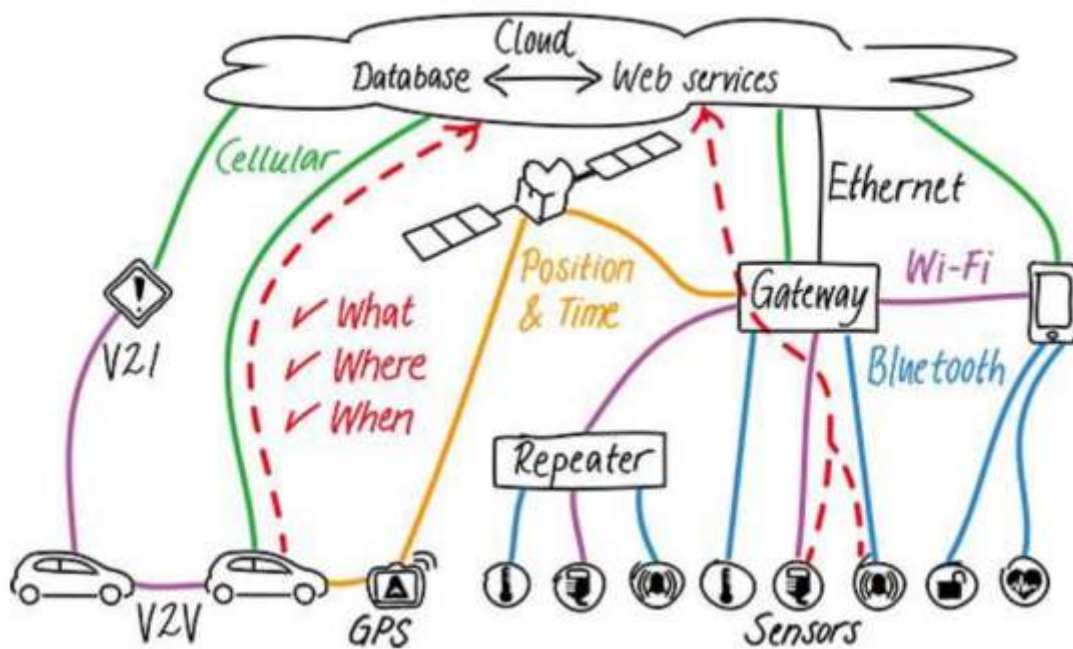


Figure 1: Internet of Things (IoT)

Dialog's IoT Multi Sensor development kit (MSK) is a compact development kit with 15 degrees of freedom, accompanied by five software reference applications and incorporating versatile sensor management with cloud applications. The IoT MSK is based on DA14585 Bluetooth Low Energy SoC and several motion and environmental sensors. The five software reference applications can be used by Dialog's apps running in iOS or Android devices. Users can build a rich cloud application with the data from this IoT MSK in just a few configuration steps. In addition, the cloud applications provided by Dialog can be used to monitor the data from the IoT MSK sensors and program IFTT events. This IoT MSK offers users high flexibility in product design to cope with the fast changing IoT market.

4 IoT Sensors Reference Application

4.1 Description

The IoT Sensors reference application is a mobile application developed by Dialog Semiconductor for both iOS and Android. It is designed to work with the DA14585 IoT MSK and offers a variety of functions that use most capabilities of the IoT MSK. Among others, users can graphically monitor the readings from environmental sensors, inertial measurement unit (IMU), or SmartFusion and configure the application according to the scenario's needs at the same time.

This section describes how to download and install Dialog's IoT Sensor App on a mobile device, and provides detailed information on the App interfaces and how to configure the App.

4.2 IoT Sensors App in Android and iOS

4.2.1 Installing from Apple AppStore

Dialog IoT Sensors app for iOS devices is available from Apple AppStore.



Figure 2: Installing Dialog IoT Sensors App from Apple AppStore

DA14585 IoT Multi Sensor Development Kit

4.2.2 Installing from Google PlayStore

Dialog IoT Sensors app for Android devices is available from Google PlayStore.



Figure 3: Installing Dialog IoT Sensors App from Google PlayStore

4.3 Screens of Dialog IoT Sensors App

4.3.1 Scan Screen

Once the DA14585 IoT MSK receives power, users will see the advertising LED blinking, indicating that the device is currently visible to client applications. After starting the app, the first screen users see is the "scan screen" (Figure 4).

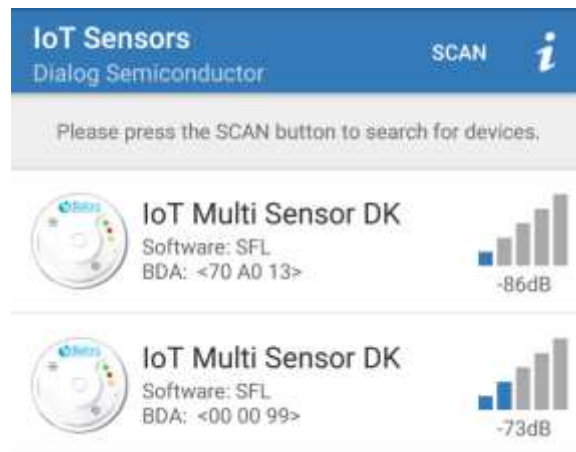


Figure 4: Scan screen of Dialog IoT Sensors App

Only the IoT MSK are listed on this page. The App hides BLE devices that do not have the advertise values related to this DK.

DA14585 IoT Multi Sensor Development Kit

Two fields are listed on the scan screen: the device type and the last three bytes of the Bluetooth Device (BD) address. Tap on a device in the list to connect to it. Note that a listed device might not be active (and the LED does not blink) when the advertise timeout has expired. To update the scan list, touch the screen to scroll downwards (iOS) or press the **Scan** button (Android).

4.3.2 Side Menu

From the side menu (Figure 5) users can navigate to the rest of the screens that present:

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

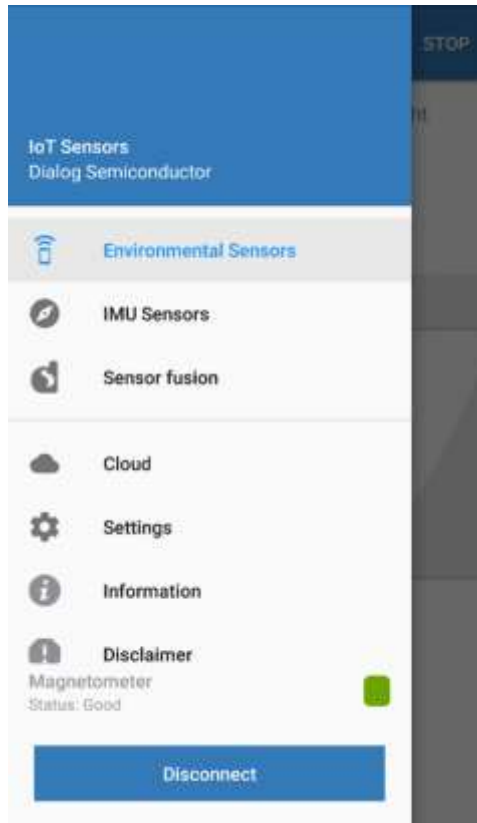


Figure 5: Side Menu

Also, a **Disconnect** button is available to disconnect the IoT MSK device. This action will route the application to the scan screen.

DA14585 IoT Multi Sensor Development Kit

4.3.2.1 Environmental Sensors

Upon successful connection, the environmental sensors screen (Figure 6) appears. This screen presents the following values:

- Temperature
- Humidity
- Pressure
- Ambient light
- Air Quality: GOOD/AVERAGE/LITTLE BAD/BAD/VERY BAD/GET OUT. The accuracy of the displayed Air quality status is also displayed (Note 1). The air quality is a feature that is not calculated by the default firmware shipped with the device due to code space constraints.
- Proximity status: ON if an object is close to the surface of the sensor/OFF if there is no object (Note 2).
- Button status

In this screen the sensors have received automatically the START command and the sensor enter active state as described in 4.3.5 in Ref. [18]. Users can stop the sensors at any time by pressing the **STOP** button on the top-right of the screen.



Figure 6: Environmental Sensor Screen of Dialog IoT Sensors App

Note 1 The Air quality algorithm is self-calibrated, but it needs to take some values of poor air quality to reach high accuracy of values quickly. This can be achieved by enclosing the IoT Multi Sensors DK in a box and blow through a hole in it. CO₂ is a big percentage of human breath.

Note 2 The Proximity sensor status is affected by the top cover of the plastic enclosure of the IoT MSK. If the MSK is in the enclosure, the values of the proximity hysteresis must be carefully selected manually or by the procedure described in Section 4.3.3.1, otherwise the Proximity sensor status will be always ON.

4.3.2.2 IMU Sensors

Figure 7 presents raw values from the following sensors:

- Accelerometer
- Gyroscope
- Magnetometer: it is not enabled by default. User should enable it from the Basic Settings (see Section 4.3.3)

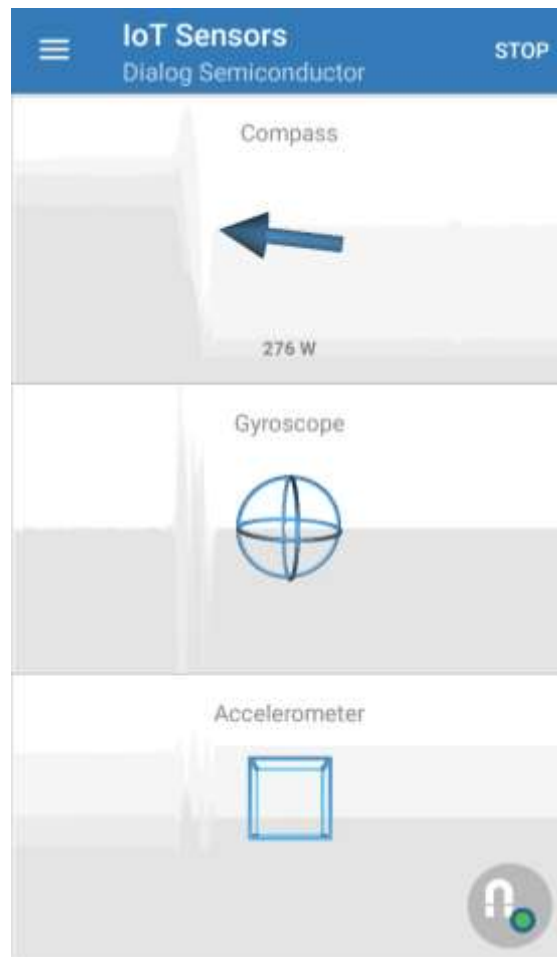


Figure 7: IMU Sensor Screen of Dialog IoT Sensors App

DA14585 IoT Multi Sensor Development Kit

4.3.2.3 Sensor Fusion 3D Screen

Users can navigate to the 3D screen from the side menu (SFL project only). Figure 8 is a 3D illustration of the position of the DA14585 IoT MSK. Notice that when we move the MSK, the 3D drawing immediately adjusts itself to the new position.



Figure 8: 3D Screen of Dialog IoT Sensors App

DA14585 IoT Multi Sensor Development Kit

4.3.3 Basic Settings Screen

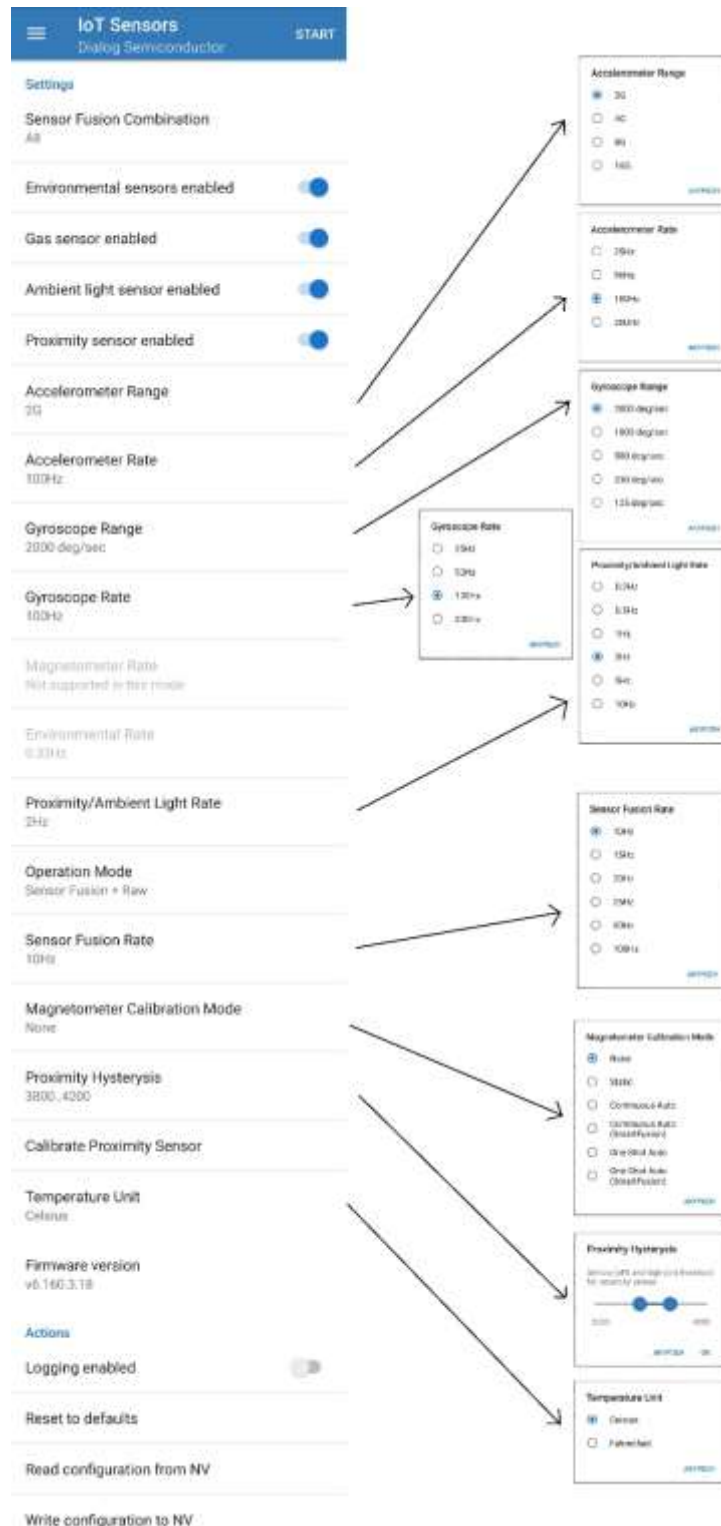


Figure 9: Basic settings of Dialog IoT Sensors App

In the basic settings screen (Figure 9), users can enable/disable available sensors and execute Sensor Fusion Library (SFL). Users can set the basic configuration parameters of the sensors, like range and sampling rate of accelerometer and gyro, sampling rate of Sensor Fusion, magneto,

DA14585 IoT Multi Sensor Development Kit

environmental and Proximity/Ambient Light sensors, and hysteresis of proximity sensor state. Calibrating the proximity sensor is available via the basic settings screen while users can also enable/disable the magneto calibration and select the desired method or mode.

Finally, the selected settings can be stored in nonvolatile memory or the configuration values can be reset to defaults.

IMPORTANT NOTES
<ul style="list-style-type: none"> Do not change the settings while the IoT MSK is in running state. First press the STOP button A modified setting will be applied after the next START command. The new setting will not be retained upon disconnection, unless the Write configuration to NV button has been pressed All modified settings only exist in RAM until the Write configuration to NV button is pressed. Press the Read Configuration from NV button to restore the previous settings. To restore the default settings, press the Reset to defaults button

4.3.3.1 Calibrate Proximity Sensor

To automatically find the proximity hysteresis limits, launch the proximity sensor calibration. Upon pressing the **Calibrate Proximity Sensor** button, users will first be presented with the screen in Figure 10. Pressing **START** generates the message **Calibrating, please wait...** and when the sensor is calibrated, another message **Proximity Sensor Calibrated** is present on the screen.

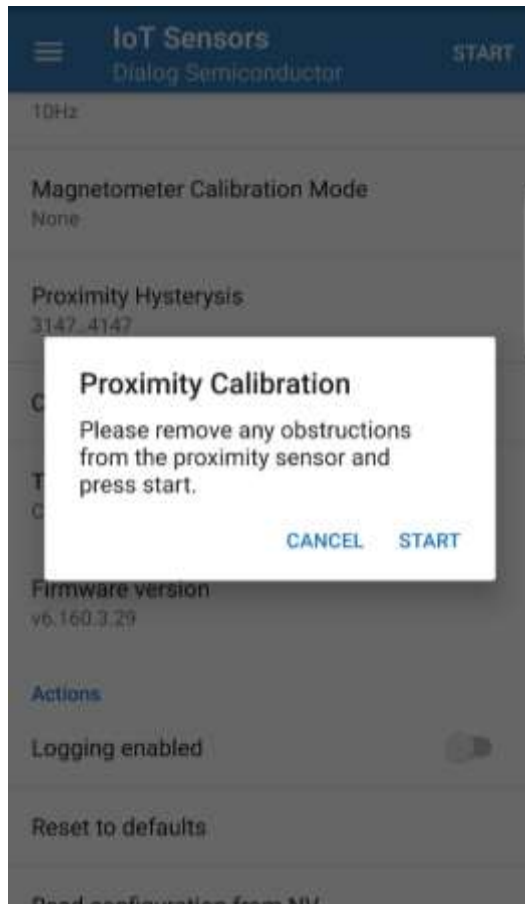


Figure 10: Proximity Sensor Calibration Prompt

DA14585 IoT Multi Sensor Development Kit

4.3.4 Magnetometer Calibration Settings Screen



Figure 11: Calibration Settings of Dialog IoT Sensors App

The calibration page (Figure 11) contains the controls and parameters for various calibration modes. Only the controls and parameters for the currently enabled calibration mode can be modified and those that apply to other modes are greyed out.

The controls and parameters are:

- **Apply:** Enables/Disables application of calibration coefficients on raw sensor data
- **Matrix apply:** Enables/Disables application of calibration matrix on raw sensor data. If disabled, only offset vector is applied. This control only applies to basic auto-calibration
- **Update:** Enables/Disables updating of calibration coefficients
- **Matrix update:** Enables/Disables updating of calibration matrix. If disabled, only offset vector is updated. This control only applies to basic auto-calibration
- **Initialize from Static Coeffs:** If enabled, the calibration coefficients algorithms are restored from the previous instantiation of the auto-calibration rather than being reset to default values
- **Reference Magnitude:** The expected magnitude of the magnetometer vector
- **Magnitude Range:** The expected range of magnitude of the magnetometer vector
- **Magnitude Alpha:** Coefficient for the magnetometer vector magnitude filtering
- **Magnitude Gradient Threshold:** Threshold applied to rate of change of magnetometer vector magnitude for detecting settling
- **Offset Mu:** The rate of convergence for the offset coefficients
- **Matrix Mu:** The rate of convergence for the matrix coefficients
- **Error Alpha:** Coefficient for the overall error filtering
- **Error Threshold:** Threshold applied to the filtered overall error for detecting convergence

NOTE
The fixed-point parameters are expressed in their raw integer form. For a more detailed description of the behavior of these controls and parameters, see <i>section 4.10</i> in Ref. [18].

4.3.4.1 Magnetometer Calibration File

It is possible to send the calibration coefficients for the currently enabled calibration mode from the application to the device. The coefficients must be loaded from file using the associated button. The coefficients can then be stored in Flash memory for subsequent initialization of the calibration algorithm. Similarly, the current set of calibration coefficients can also be saved to file.

The format of the file uses an INI format of the form:

```
[magnetometer calibration]
sensor_type      = 2
q_format         = 14
offset_vector    = 0, 0, 0
matrix           = 16384, 0, 0, 0, 16384, 0, 0, 0, 16384
```

All values are integers except for the matrix coefficients which are fixed point values (with precision specified by `q_format`) expressed in raw integer form. The `sensor_type` should always be set to two (representing magnetometer).

The `offset_vector` coefficients are listed in [x, y, z] order.

The matrix coefficients are listed in [row, column] order: [0, 0], [0, 1], [0, 2], [1, 0], [1, 1], [1, 2], [2, 0], [2, 1], [2, 2].

4.3.5 Magnetometer Auto-Calibration Procedure

When operating in auto-calibration mode, it is necessary to perform a calibration procedure to provide the calibration algorithm with enough raw sensor data for it to accurately determine the hard and soft iron magnetometer distortions that exist and calculate the calibration coefficients required to compensate for them. Until this calibration procedure has been completed, the magnetometer will not be properly calibrated and drift artefacts will be observable in the 3D orientation visualization.

DA14585 IoT Multi Sensor Development Kit

This calibration procedure involves rotating the board through a representative set of orientations so that a complete sphere of magnetometer vector positions has been captured. In order to produce sufficient data for successful calibration, it is recommended to rotate the board randomly through all orientations as quickly as possible for about 10 to 20 seconds.

The calibration procedure should be performed in a uniform and static magnetic environment. When the magnetic environment changes, the calibration procedure must be repeated in order to allow the algorithm to adapt.

When operating in an environment with much variability in the strength and direction of the magnetic field, it is recommended that the auto-calibration procedure is first performed in a controlled location. Auto-calibration should then be disabled by setting the **Update** control setting to OFF to prevent further modification of the calibration coefficients. This will prevent the constantly changing magnetic environment from upsetting the calibration algorithm and introducing drift artefacts in the 3D visualization.

4.3.5.1 Prerequisites for Magnetometer Calibration

- The device should be away from any sources of magnetic interference. Electric or electronic devices located near the IoT sensor might influence the outcome. Moreover, ferromagnetic materials used in office desks or chairs should be kept away from the device while calibrating
- The device should be kept within a single location while calibrating it. The calibration algorithms are designed to compensate within a stable and uniform magnetic environment. When sudden changes in the magnetic environment occur, it will take some time for the calibration algorithm to adapt
- The magnetization level of the battery is of importance, but the calibration algorithm will fail even without attached battery, when the other conditions are not met. The temperature of the battery is also important and should be kept stable during the calibration process
- Make sure the IoT MSK batteries are not depleted

4.3.5.2 Procedure of Magnetometer Calibration

1. Connect to the device using the provided iOS or Android IoT Sensors App.
2. Stop the device and go to the basic configuration page.
3. Select **Reset to Defaults**.
4. Select **Continuous and Auto Calibration Mode: Basic or SmartFusion** in **Magnetometer Calibration Mode**.
5. Press **Write configuration to NV** in order to preserve the settings.
6. Press **START** and rotate the device with random orientations with one to two rotations per second for 15 to 20 seconds. When rotation is stopped, the 3D image will stabilize, and the magnetometer value will be in effect in the sensor fusion algorithm.
7. If the calibration fails, restart the procedure after checking the battery and changing the calibration environment (sources of magnetic interference).
8. Every time **STOP** is pressed, the calibration coefficients are saved.

4.3.5.3 Magnetometer Sensor and Calibration State Reporting

To provide users some visibility of the state of the magnetometer sensor and the applied calibration, various indicators have been provided in the app's user interface.

DA14585 IoT Multi Sensor Development Kit

In the compass pane on the sensor screen, two icons can be seen:

- The icon indicates whether calibration has been applied to the magnetometer data. If struck out, it means that the magnetometer data is uncalibrated, either because calibration is disabled or the "apply" flag is not set
- The icon indicates whether the sensor data is valid. If visible, it means that the magnetometer is either starting up, saturated, or the calibration routine has detected some other issues with the magnetometer data

Overlaid on all screens is an icon, indicating the state of the calibration routine. The various states include:

- Cross: Calibration is disabled
- Question Mark: The calibration routine is initialized and is not yet applied to the magnetometer data
- Red Circle: The quality of the applied calibration is poor
- Yellow Circle: The quality of the applied calibration is OK but not optimal
- Green Circle: The quality of the applied calibration is good

4.3.6 SmartFusion Coefficients Screen

The SFL coefficients screen (Figure 12) contains parameters for the sensor fusion algorithm. These parameters are:

- **Beta A:** Accelerometer scaling factor
- **Beta M:** Magnetometer scaling factor

NOTE

These are fixed point parameters that are expressed in their raw integer form. For a more detailed description of the behavior of these parameters, see *section 4.11* in [18].

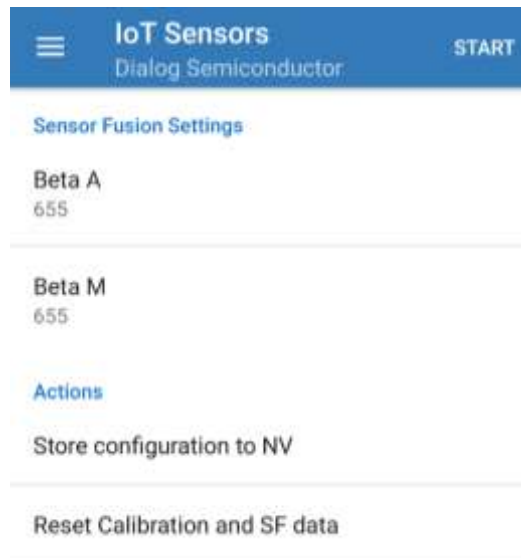


Figure 12: Sensor Fusion Settings of Dialog IoT Sensors App

4.3.7 Fast Accelerometer Calibration Screen



Figure 13: Accelerometer Calibration Settings of Dialog IoT Sensors App

Accelerometer offsets around 0 G may be observed when placing the device on a flat horizontal surface. In order to compensate for these offsets, Dialog's IoT Sensors app includes a fast calibration method. This method is simple but less accurate, but it is fast and can be performed easily. Note that the calibration should be repeated when the range of the accelerometer changes, since the offsets stored are static and not scaled on range change.

The fast calibration method includes the following steps:

1. Place the device on a stable flat surface with the LEDs facing upwards.
2. Connect the device to the IoT Sensor application.
3. Press **STOP**.
4. Navigate to the Fast Accelerometer Page.
5. Press **Start Calibration**.
6. The calibration procedure will be concluded within a few seconds and the **Calibrated** message will appear. The calibration offsets are now saved in Flash memory and the sensor board has returned to its idle status.

NOTE
The accelerometer calibration offset will be reset when the device is returned to defaults.

4.4 Air Quality Index

The BME680 environmental sensor contains a volatile organic compound (VOC) detector that may be used in combination with Bosch BSEC Library Ref. [17] to calculate the Air Quality Index (AQI) when `IAQ_ENABLED` is defined. The AQI can have values between 0 and 500. The value of AQI is inversely proportional to the air quality (Table 1). `IAQ_ENABLED` is not defined by the default code configuration due to code space constraints. If users would like to create an image that includes the

DA14585 IoT Multi Sensor Development Kit

BSEC Library that occupies approximately 23 KB of code space, remove other features. For more information see *section 4.2.4* of Ref. [18].

The BSEC Library combines the VOC values and the BME 680 environmental sensors to calculate an accurate AQI index after a self-calibration procedure. The calibration may take up to a few days since the VOC sensor needs to be exposed to a high variance of air conditions, but the procedure may be accelerated artificially by placing the IoT MSK inside a box with low air quality, which can be a high concentration of breathed air.

The library state may be periodically saved when the option `IAQ_RESTORE_STATUS_ENABLE` is enabled at the expense of code size. When this option is enabled during rebooting, a new valid AQI will be available within minutes, otherwise the auto-calibration procedure should be repeated.

Table 1: Air Quality Index

Air Quality Index	Air Quality
0-50	Excellent
51-100	Good
101-150	Moderate
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

5 Smart Tag Reference Application

5.1 Introduction

Smart Tag, Dialog's Bluetooth® low energy proximity tag reference application, provides an ideal starting point to develop a proximity tag application with the shortest time-to-market and lowest development cost and effort. The design comes with a complete software solution for the full proximity application and profile source codes. Dialog also provides fully-featured Android and iOS applications to manage the proximity tag's settings, trigger alerts, check the battery status, and play a fun **Seek & Find** game, all in source code.

The Smart Tag reference application functions in the role of a Bluetooth Low Energy Proximity Reporter defined in the Proximity Profile listed in the Bluetooth specification Ref. [15]. The Proximity profile defines the behavior of a Bluetooth device when it moves away from a peer device, and it covers the use case where a connection loss causes an immediate alert. This alert notifies the user that the devices have become separated.

The Smart Tag application is designed to run on Dialog's DA14585 MSK HW reference design.

As a starting point, developers are suggested to get familiar with the DA14585 datasheet Ref. [5], the software developer's guide of the DA14585 SDK Ref. [1], and the software platform reference of the DA14585 SDK Ref. [2].

5.2 SmartTags App in Android and iOS

5.2.1 Overview

The Smart Tag proximity reference application comes with a proximity monitoring demo app called SmartTags for mobile devices using iOS and Android. Dialog's **SmartTags** app can discover, connect, and control devices with a Smart Tag within the Bluetooth RF range of the iOS and Android devices.

The features of SmartTags include:

- Proximity reporter devices discovery
- Connection to discovered devices
- RSSI polling of connected devices
- Immediate alert
- Distance alert based on measured RSSI value
- Link Loss alert
- Battery level monitoring
- Disconnection
- Seek & Find game based on the current RSSI value

DA14585 IoT Multi Sensor Development Kit



Figure 14: SmartTags APP Icon

5.2.2 Installation

Dialog's SmartTags app is available in both Apple App Store and Google Play Store. Search for "Dialog SmartTags" and click the install icon (Figure 15). The application will then be downloaded and installed without further user interaction.

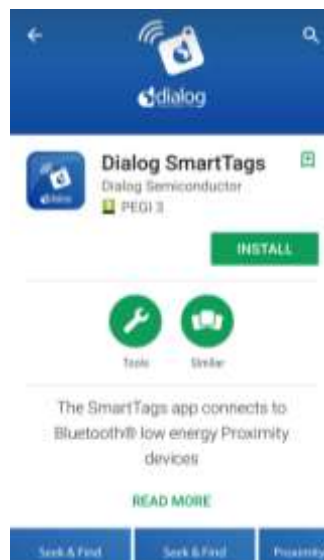


Figure 15: Installation of Dialog SmartTags App

5.2.3 Device List

Tap on the **Dialog SmartTags** icon to start the application. A list of all devices supporting the Proximity Reporter profile, including Smart Tag, is shown on the screen. A device with a smart tag gets displayed automatically when it gets within the Bluetooth RF range of the iOS/Android device. Tab on a device icon in the list to connect to it. If a device is not paired (*section 5.4.6* in Ref. [18]),

DA14585 IoT Multi Sensor Development Kit

users will be asked by a request to pair the device. If a device is already paired, it will be marked as **Connected** in the list.



Figure 16: Device List

5.2.4 Device Details

The following controls (Figure 17) are provided for monitoring and controlling of the device:

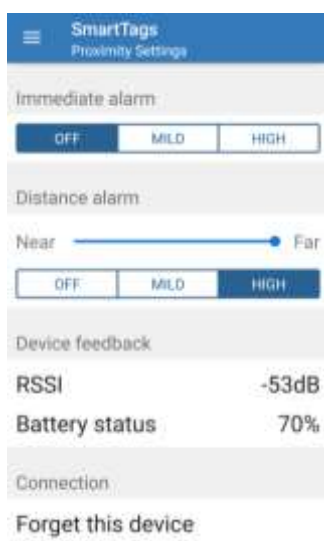


Figure 17: Device Details

- Immediate alarm:
 - Click on the **MILD** or **HIGH** button of the **Immediate alarm** section to trigger an immediate alert. The alert can be terminated by clicking on the **OFF** button
- Distance Alarm
 - Distance and link loss alerts are configured by the **Distance alarm** section controls. The SmartTags application polls the RSSI of the connected devices to estimate the distance between the iOS/Android system and the Smart Tag device
 - The **Near/Far** slider controls the RSSI threshold for triggering a distance alert. When the measured RSSI level of a connected device moves below the configured threshold or a device is disconnected, the SmartTags app will trigger an alert on both Smart Tag and iOS or Android devices. In case of an alert, the screen displays a message (Figure 18). Click **OK** to terminate the alert on either device

DA14585 IoT Multi Sensor Development Kit

- The alert level depends on the selected button (**OFF**, **MILD**, and **HIGH**). When the **OFF** button is selected, no alert will be triggered

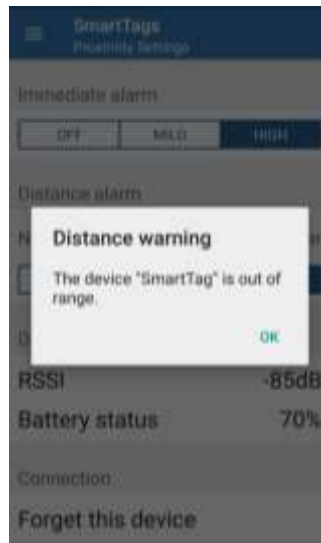


Figure 18: Distance Alert Window

- Device Feedback
 - The **Device feedback** section provides status information for the device connected to the SmartTags proximity reporter, showing the current RSSI value and the battery level of the device with a Smart Tag
- Connection
 - Users can disconnect a device from Dialog's SmartTags app by clicking on **Forget this Device**. Note that this control does not erase the bonding data. It just disables auto connection of the specific IoT MSK device
- Side Menu

Click the top left menu icon and the side menu appears (Figure 19). From the side menu users have access to the proximity settings, the **Seek & Find** game, the information of the device (Figure 20), and the disclaimer statement (Figure 21)

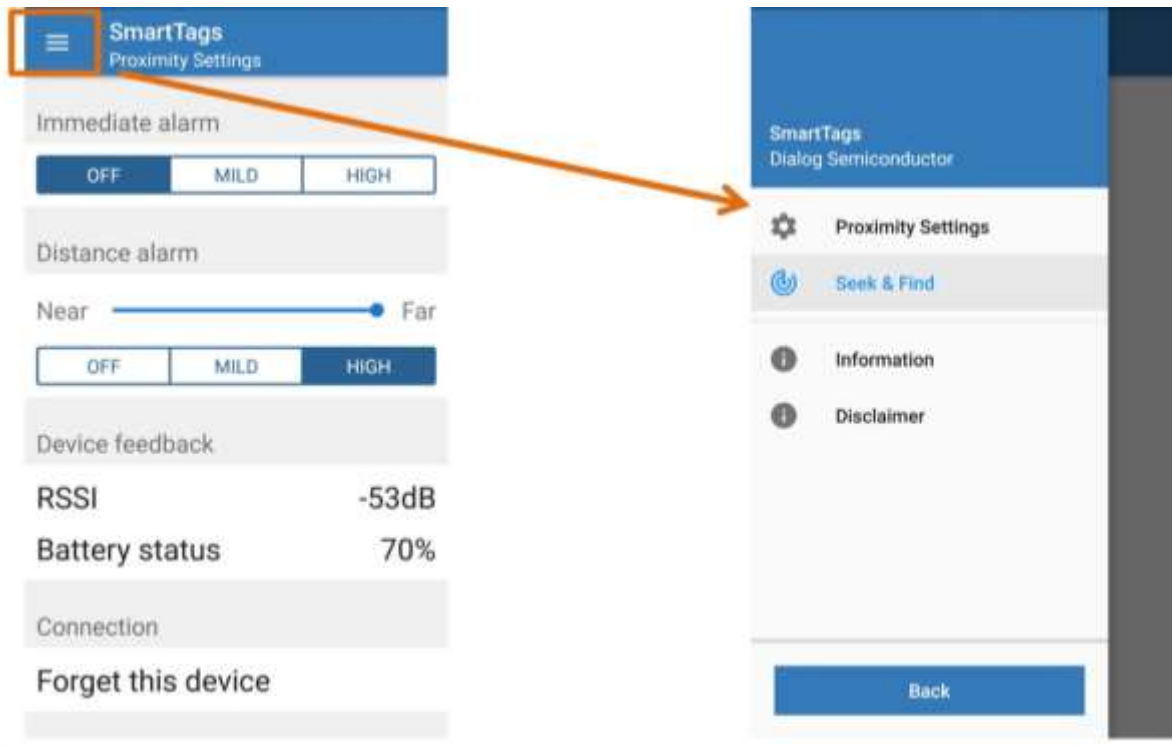


Figure 19: How to Access the Side Menu

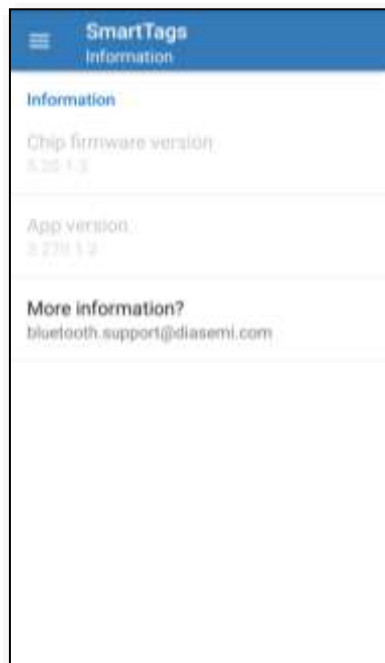


Figure 20: Device Information

DA14585 IoT Multi Sensor Development Kit



Figure 21: Disclaimer

- Seek & Find Game
 - After users click the **Seek & Find** icon (Figure 19), Dialog's SmartTags app will start a Warmer/Colder game with the selected device
 - The application changes states and displays an image according to the RSSI-based estimate of the distance of a device with a Smart Tag. The app changes to a warmer state when it approaches a device with a Smart Tag, while it changes to a colder state when it becomes further from a device with a Smart Tag. The various **Seek & Find** screens are shown in Figure 22

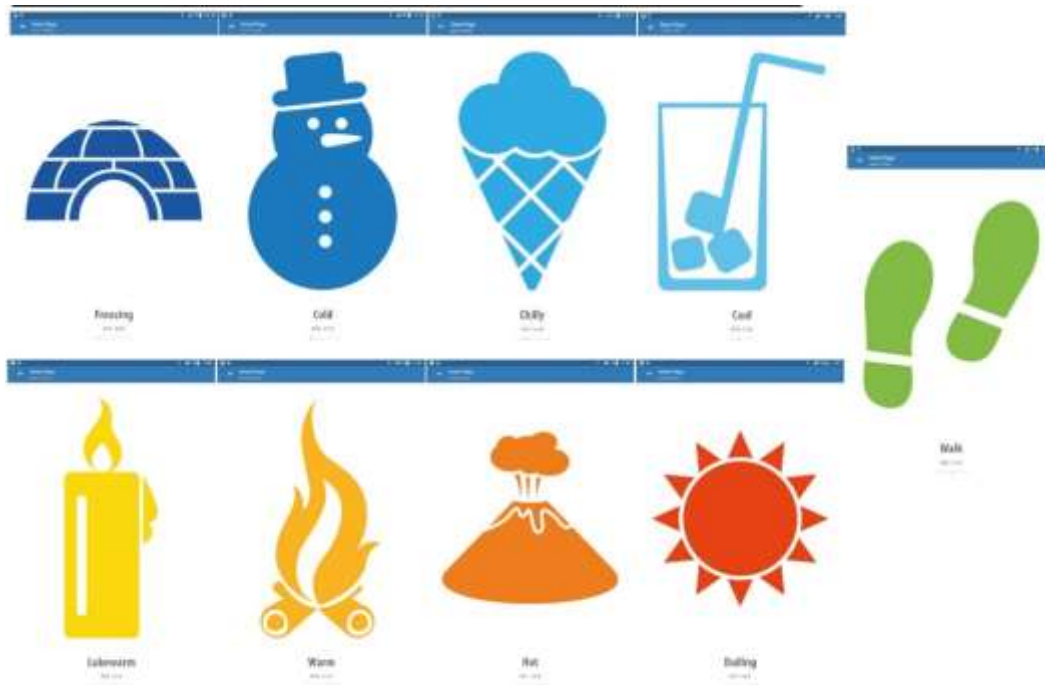


Figure 22: Seek & Find Screens

6 Beacon Reference Applications

6.1 Introduction

This section describes the Bluetooth® Low Energy Beacon reference application based on DA14585. It serves as a user guide to customize the beacon for any desired purpose.

Dialog Semiconductor is a member of the iBeacon™ program. Contact your local sales office to learn more about the possibilities we can offer relating to this standard. Dialog's Beacon reference application also supports the AltBeacon protocol as well as the Eddystone™ protocol, with supported modes being the Eddystone-UID, the Eddystone-URL, and the Eddystone-TLM (unencrypted).



Figure 23: Beacon Protocol Logos

The Beacon reference software can be downloaded from the [Dialog support portal](#) and run on the DA14585 IoT MSK reference hardware. It also runs on Dialog's DA14585/14586 Development kit (Expert/Pro/Basic).

6.2 What is a Beacon?

Beacons are battery powered devices that advertise a particular Bluetooth low energy payload with identifying information. In short, it is a device that just says (Figure 24):



Figure 24: Bluetooth Low Energy Beacon

6.3 Beacon Example

Table 2 shows a beacon configuration that is chosen for an international museum application, where each exhibit has its own unique beacon. The specific advertising data transmitted by each beacon is picked up by mobile phones near the exhibit. The mobile phone will then direct users to additional information regarding the exhibit.

In this museum example, the organization uses a Universal Unique Identifier (UUID). The museum location is indicated by the Major Field and the exhibit number within the museum by the Minor field.

DA14585 IoT Multi Sensor Development Kit

Table 2: Example of Advertising Data from a Museum Beacon

Museum Location		London	Paris	Amsterdam
UUID		1234A567-3854-ABED-8FAC-56E783159AE2		
Major		10	20	30
Minor	Exhibit #1	1	1	1
	Exhibit #2	2	2	2
	Exhibit #3	3	3	3

When the beacon sends "20" as a major value and "1" as a minor value, the application on the smartphone/tablet guides visitors to additional information on exhibit #1 from the Paris museum. This additional information might come from the smartphone application or from the Internet. As a result, visitors receive only the specific information of interest when standing close to the exhibit (Figure 25).



Figure 25: Description of the Exhibit on a Smartphone

The smartphone application can also provide a distance indication to the beacon using the RSSI value.

There are endless other applications using Beacon apart from the museum application, for example, in retail, advertising, and sports events.

DA14585 IoT Multi Sensor Development Kit

6.4 Finding Advertising Devices

A smart device running beacon scanning software will pick up the advertising data and present the received values on the screen. Several beacon monitoring applications for iOS and Android operating systems can be used for the evaluation of the beacon examples of the DA14585 IoT MSK reference application. In this document, **Locate** app, an Android application, has been used for the test purpose (Figure 26).



Figure 26: Locate App Screenshot

In Figure 26, two different beacons are advertising. The first is an Eddystone-UID and the second is an iBeacon. For the iBeacon, the application also parses the Major and Minor values.

7 IoT Cloud Applications

7.1 Introduction

This section contains all essential information for end users to make full use of the mobile-side and cloud-side functionalities provided by the five IoT reference applications offered together with the DA14585 IoT MSK. This section describes the five available IoT cloud applications (historical data, alerting, control, asset tracking, and 3-D game), and the accompanying Alexa voice service (applicable to the "control" and "asset tracking" cloud applications) as well as step-by-step procedures for their access and use.

The applications and the mobile apps cloud integration make use of the Yodiwo Cloud Platform, though it is presented through a new Dialog-branded domain called *.dialog-cloud.com.

To enable the cloud component, visit the new Cloud section in DA14585 IoT MSK's Android and iOS mobile apps. There, a link is automatically established among the mobile apps, the new cloud layer, and the Yodiwo Cloud Platform. This link is unique and securely tied to each user.

Multiple apps (and by extension, phones) can be linked to the same user account, using a simple transient-token-based mechanism.

Upon account creation, users also gain access to their own web application in dialog-cloud.com. Using the dedicated mobile apps and the web application, users can:

- View their collected data
- Setup and manage rules that will automatically be relayed to their IoT MSK via their linked mobile apps
- Use their linked mobile apps as tracking stations and assign asset names to their IoT MSK(s) in order to leverage a full-featured asset tracking solution
- Use Amazon Alexa to control their IoT MSK(s) or track their assets using voice commands from their mobile apps
- Administer their accounts
- Manage their IoT MSKs

All data and actions performed are completely personal to users and their IoT MSK(s) through the issue of a universally unique API Key. For convenience, users may choose to have this API Key and the web application's URI sent to an email of their choice.

7.2 User and Device Management

7.2.1 Users

By using the mobile application, users can create their personal accounts. To create a new user account, open the **IoT Sensors** mobile app in Android/iOS and go to the **Cloud Settings** page under the **Settings** menu (Figure 27). By selecting the **Enable Cloud**, a dialog pop-up window appears (Figure 28), which allows to either create a new account or to add the mobile device to an existing one. Click **CREATE NEW ACCOUNT** so that a new user account can be created, and the cloud link is generated. To access the generated cloud link via a specified email account, enter a valid e-mail address and click **SEND**. The cloud link will direct you to the dedicated cloud application of your account.

DA14585 IoT Multi Sensor Development Kit

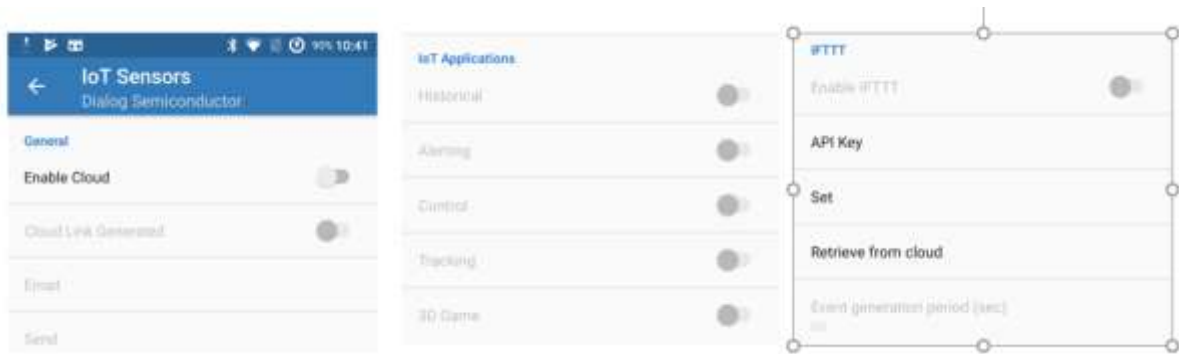


Figure 27: Cloud Settings

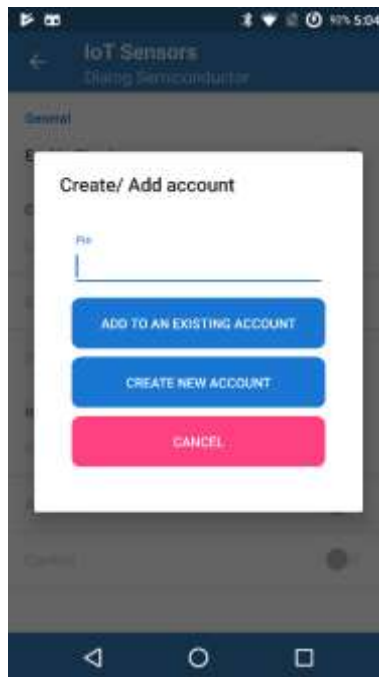


Figure 28: Create New Account or Pairing with an Existing Account

Multiple mobile apps can be paired to the same account following the procedure below:

1. Open the cloud application and go to the **Users** page under the **Management** menu (Figure 29). Press **GENERATE TOKEN** to generate a token and press **APPLY** to update the Yodiwo service.

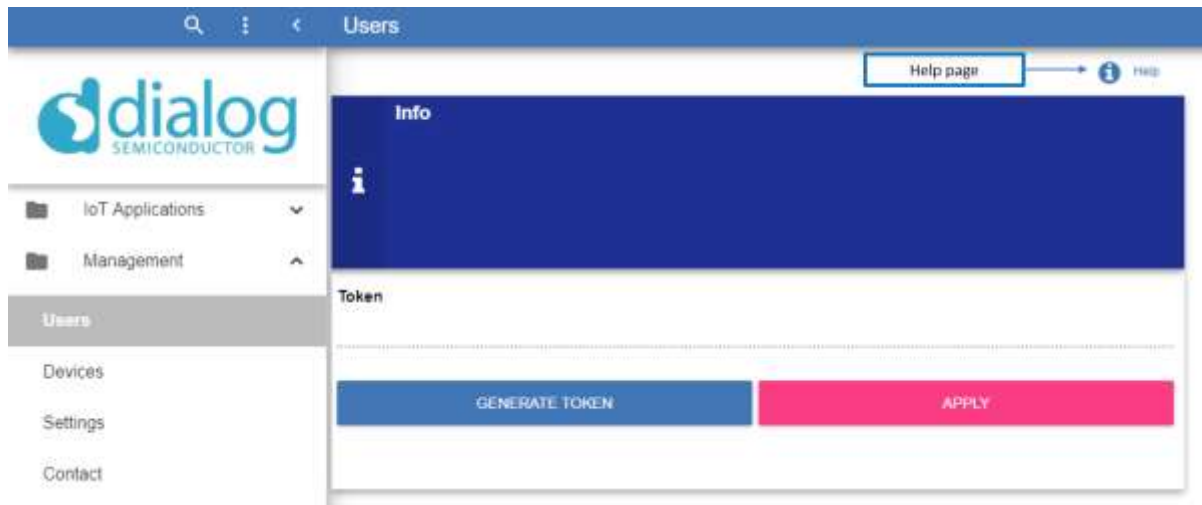


Figure 29: User Management in Cloud Application

2. In the IoT Sensors mobile app in Android/iOS, go to **Settings > Cloud Settings > Enable Cloud**, and in the pop-up window (Figure 28) type in the token generated at the cloud application and press **ADD TO AN EXISTING ACCOUNT** to pair this mobile app with an existing account.

7.2.2 Devices

Using the cloud application, users can manage their devices. Open the **Devices** tab under the **Management** menu, and all the devices are shown at the Devices table. Users can edit/delete any of their devices. Figure 30 shows that users can edit the name and description fields and update them by clicking on them.

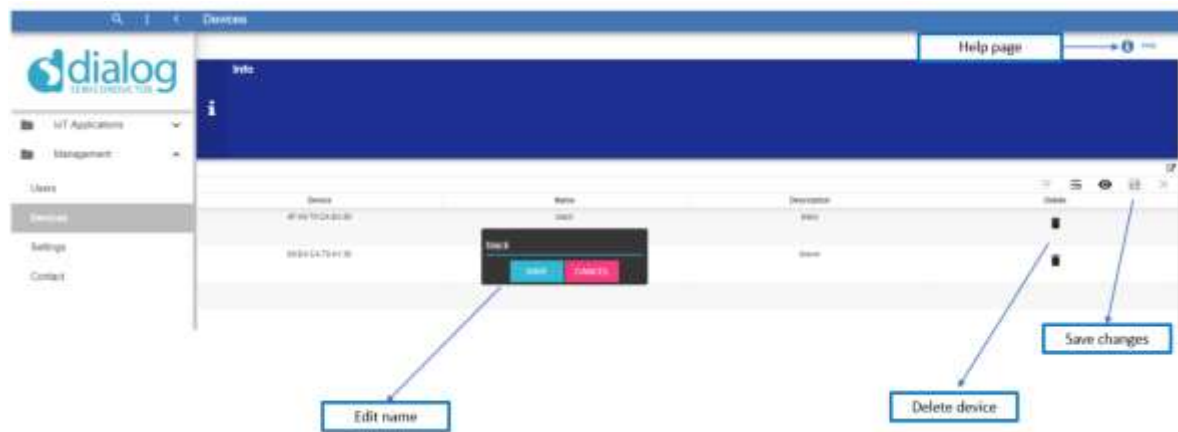


Figure 30: Device Management

7.3 Sensor Data

7.3.1 Live Data

Through the 'Live Data' application, users can view real time data received from their devices. To view the live data of your devices, open the cloud application and go to the **Live** page under **Sensor Data** in the **IoT Applications** folder. Now, you can set your filtering criteria and data from the selected device will be displayed once you press the **APPLY** button (Figure 31).



Figure 31: Live Data

7.3.2 Historical Data

Through the "Historical Data" application, users can view the available data received by the devices, filtered by date range, sensor type (for example, temperature, humidity, and pressure), and device.

7.3.2.1 Cloud Application

To view the received data of your devices (Figure 32), open the cloud application and go to the **Historical Data Visualization** page under the **IoT Applications** folder. Now, you can set your filtering criteria and the available data will be displayed by pressing the **APPLY** button.



Figure 32: Historical Data: Cloud application

7.3.2.2 Mobile Application

Users can also view the historical data using the mobile application (Figure 33). To access your historical data, select the **Historical Data** tab from the navigation bar, first from the left. Then, set your filtering criteria and press the **APPLY** button to display the available data.



Figure 33: Historical Data: Mobile Application

DA14585 IoT Multi Sensor Development Kit
7.3.2.3 Rest API

In order to retrieve the historical data of a specific sensor and date range, users should make the following `http GET` request:

https://service.dialog-cloud.com/dialog/edge/iotapps/historical/{SENSOR}?StartDate={START_DATE}&EndDate={END_DATE}&EKID={EKID}&UserId={USERID}, where:

- SENSOR could be one of the following:
 - g_temperature
 - g_pressure
 - g_humidity
- START_DATE and END_DATE parameters are in the format of "dd-mm-yyyy"
- EKID is the ID of the IoT device from which users would like to retrieve the historical data
USERID is the ID of the user

Below there is an example of the response to the previous request:

```
{
  "Timestamps": [
    "2018-10-31T08:44:41",
    "2018-10-31T08:46:11",
    "2018-10-31T08:47:41"
  ],
  "Values": [
    24.16,
    24.39,
    24.64
  ]
}
```

7.4 Alerting

Using the IoT Alerting application, users can configure a set of rules, which are triggered when the specified condition based on the values from the IoT MSK's sensors is met. Once a rule is triggered, an e-mail will be sent to the address specified by the user in the respective rule. The IoT Alerting application supports the following sensors in DA14585 IoT MSK:

- Temperature
- Humidity
- Pressure
- Air Quality
- Ambient Light

7.4.1 Cloud Application

To add/update an alerting rule (Figure 34), open the cloud application and go to the **Alerting** page under the **IoT Applications** folder. There a form for adding/updating rules is presented. To add a rule, first fill in the fields of the form and then press the **ADD/UPDATE** button to save the rule. Users can also enable/disable, delete, and/or edit an already added rule using the table at the bottom of the page.

A maximum of 10 alerting rules per IoT MSK can be added.

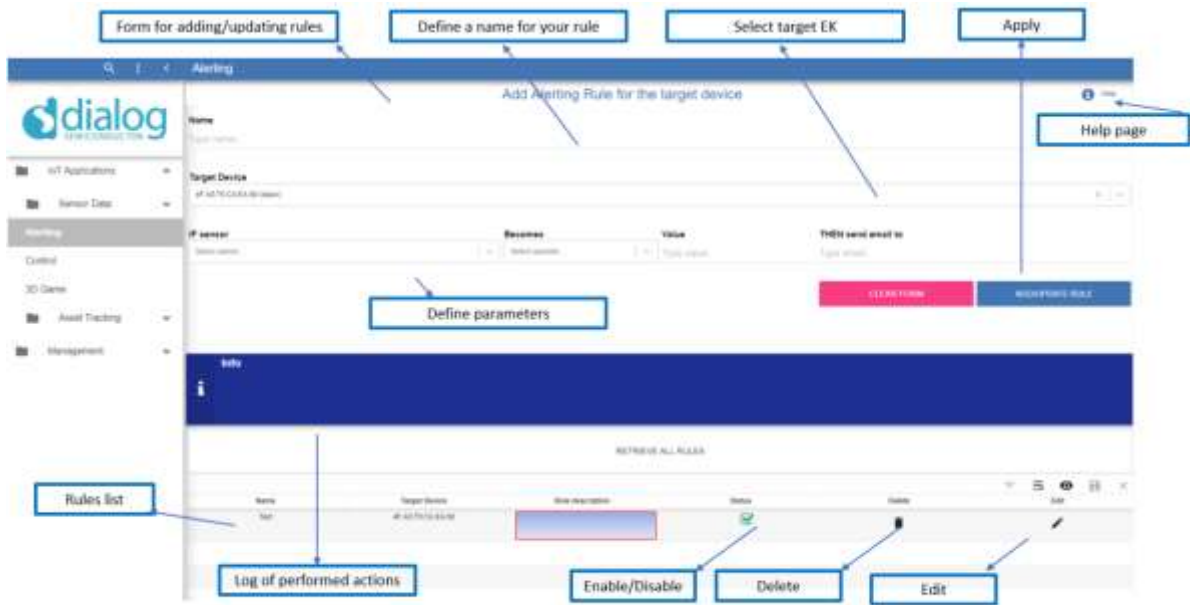


Figure 34: Alerting: Cloud Application

7.4.2 Mobile Application

An alerting rule can also be added from the mobile application (Figure 35). To add a new rule, select the **Alerting** tab from the navigation bar (second from the left), fill in the appropriate fields of the form, and press the **APPLY RULE** button. Users can also use the **SYNC** button to see the number of active rules.

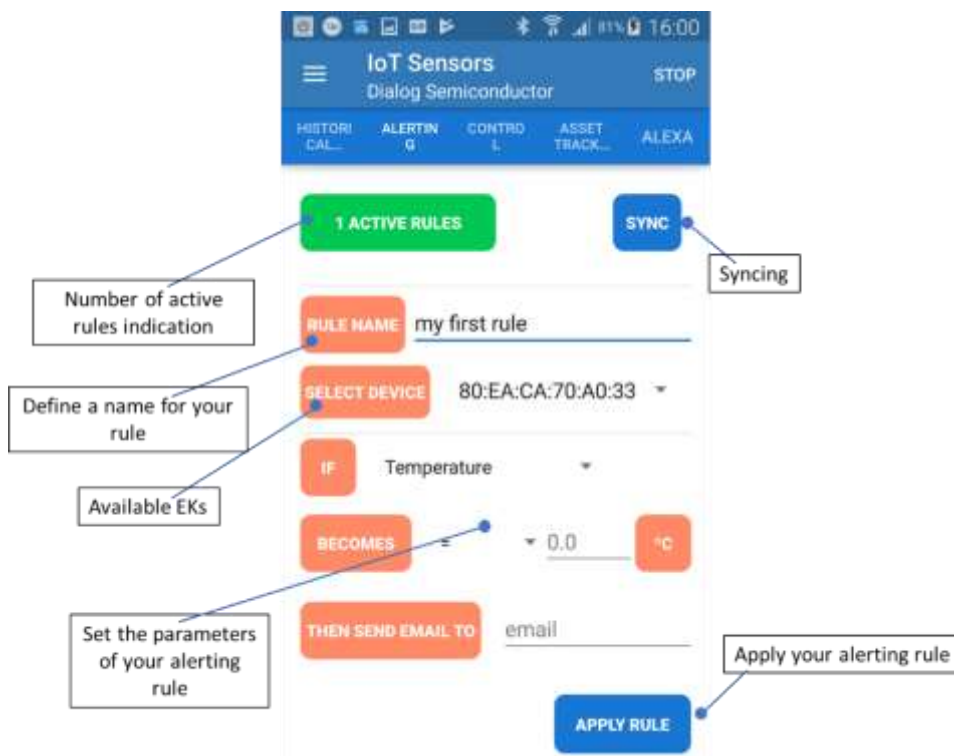


Figure 35: Alerting: Mobile Application

7.5 IoT Control Application

Using the IoT Control Application, users can configure a set of cloud-triggered rules. Two types of cloud triggers are supported:

- Weather
- Foreign Exchange Rates

Once a rule is triggered, the target MSK will be driven with the specified actions. The supported action types are:

- Switch LED on
- Switch LED off

7.5.1 Cloud Application

To use the "control" in Cloud application (Figure 36), open the web app and go to the **Control** page under the **IoT Applications** folder. To add a new rule; first fill in the fields of the form and then press the **ADD/UPDATE** button to save the rule. Users can also enable/disable, delete, and/or edit an already added rule using the table at the bottom of the page.

- For the cloud trigger "Weather", only the value "Temperature" is considered valid in the **IF** field
- For the cloud trigger Forex, the **IF** field should be filled in using one of the major forex symbols
 - EURUSD, USDJPY, GBPUSD, USDCHF, EURGBP
 - EURJPY, EURCHF, AUDUSD, USDCAD, NZDUSD
 – The **In** field is ignored for the FOREX rules

A maximum of 10 control rules per DA14585 IoT MSK can be added.



Figure 36: Control: Cloud Application

7.5.2 Mobile Application

A control rule can also be added from the mobile application (Figure 37). To add a new rule, select the **Control** tab from the navigation bar (third from the left), fill in the appropriate fields of the form,

DA14585 IoT Multi Sensor Development Kit

and press the **APPLY RULE** button. Users can also use the **SYNC** button to see the number of active rules.



Figure 37: Control: Mobile Application

7.6 IoT Asset Tracking Application

The IoT Asset Tracking application is a full-fledged tracking solution for users to manage and track their personal belongings/assets, which can be their wallets, backpacks, or anything they would like to track. These personal belongings can be linked to the DA14585 IoT MSK which is the device actually being tracked. The role of tracking stations, hereby referred to as "gateways", is carried out by either the mobile devices or Raspberry Pi(s) linked to the user's account.

IoT MSK(s) can be assigned to assets through the mobile application. Once the mobile App is linked to a user, the mobile devices are included in the user's available gateways automatically. Using the corresponding pages in the cloud application, users can:

- Manage their gateways: assign locations, names, and descriptions
- Manage their assets: assign supervisors for receiving lost alerts, names, and descriptions
- Manage supervisors
- Manage locations
- Monitor asset locations and status

The aforementioned concepts and procedures for configuring and using the asset tracking application are thoroughly described in the following sections.

7.6.1 Cloud Application

7.6.1.1 Asset Tracking

To use the Asset Tracking in Cloud application, open the web app and go to the **Asset Tracking** page under the **IoT Applications** folder. From **Tracking** (Figure 38) and **Asset Tracker** (Figure 39) pages, users can monitor the latest state of the assets.



Figure 38: Asset Tracking Table

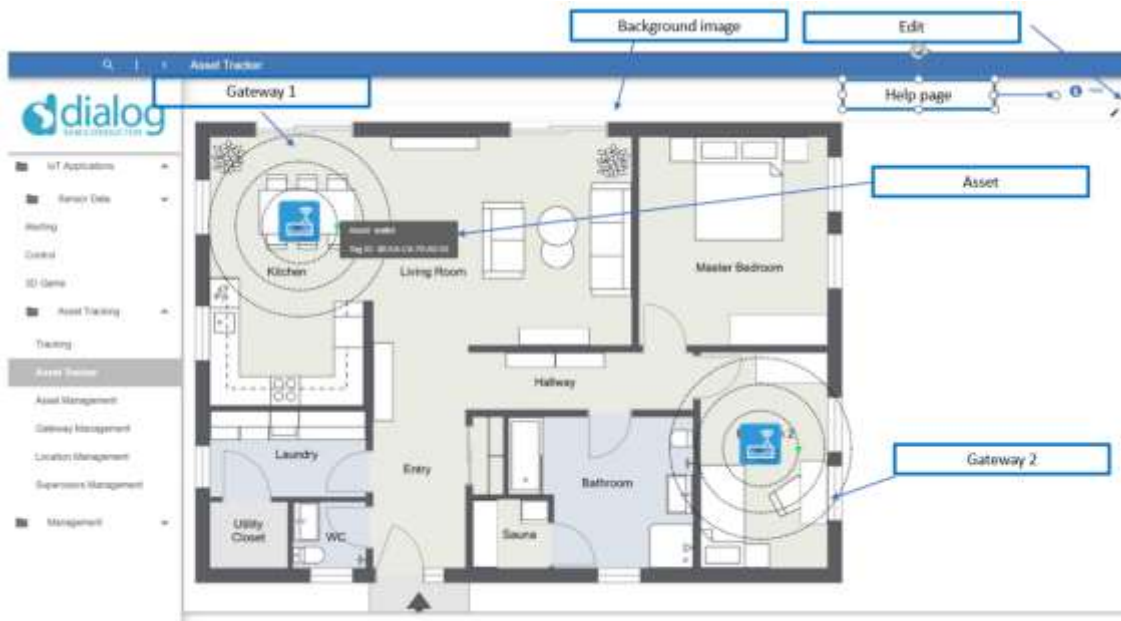


Figure 39: Asset Tracker

To change the settings of the Asset Tracker, press the **Edit** button. From the shown **Settings** page, users can change the background image and the gateways layout settings (Figure 39 and Figure 40). By pressing the Gateways layout **Edit** button, users can place the gateways on the top of the background image and enable/disable the available quarters for the assets. Each gateway has three concentric cycles corresponding to different levels of proximity: "Immediate", "Near", and "Far". Assets are represented as dots on a gateway’s circumference. Lost assets are represented by red dots and the rest by green dots.

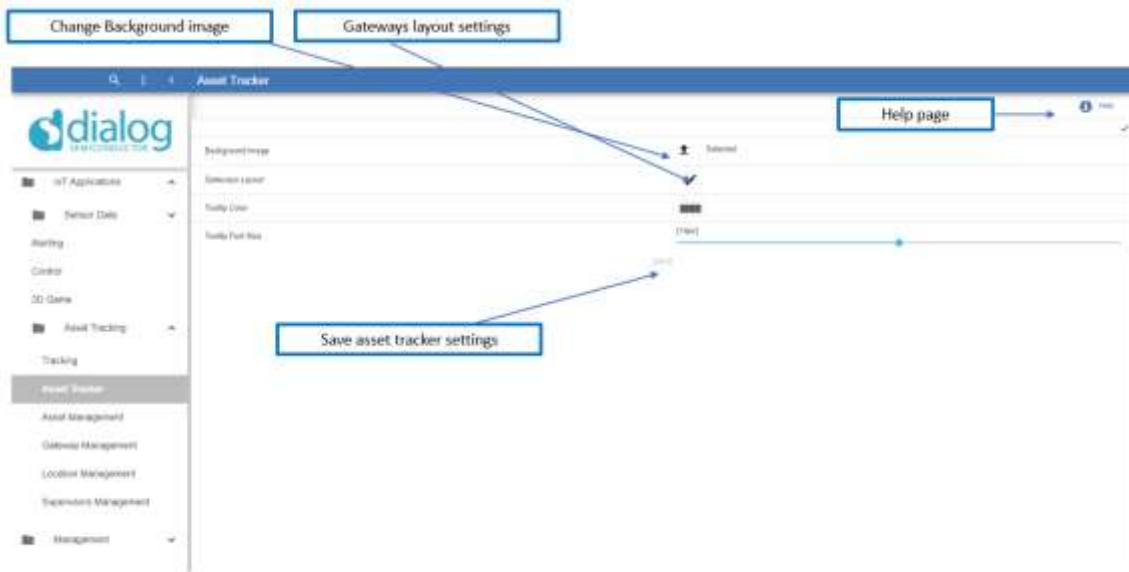


Figure 40: Asset Tracker - Settings



Figure 41: Asset Tracker - Gateway settings

7.6.1.2 Asset Management

Using the cloud application, users can manage their assets. Open the **Asset Management** tab under the **Asset Tracking** menu, and all the assets are shown at the assets table. Users can update the name, description, and supervisor of the selected asset by pressing **Edit** and then **UPDATE** button. See [Figure 42](#).



Figure 42: Asset Management

7.6.1.3 Gateway Management

Users can manage their gateways from **Gateway Management** tab under the **Asset Tracking** menu. Users can edit the name and the location of any of the gateways by pressing **Edit** and **UPDATE** button (Figure 43).



Figure 43: Gateway Management

7.6.1.4 Location Management

From the **Location Management** tab (Figure 44) users can add, update, and delete any of the locations assigned to the gateways. On the creation of a user account, a default location mentioned as Unknown location is created.

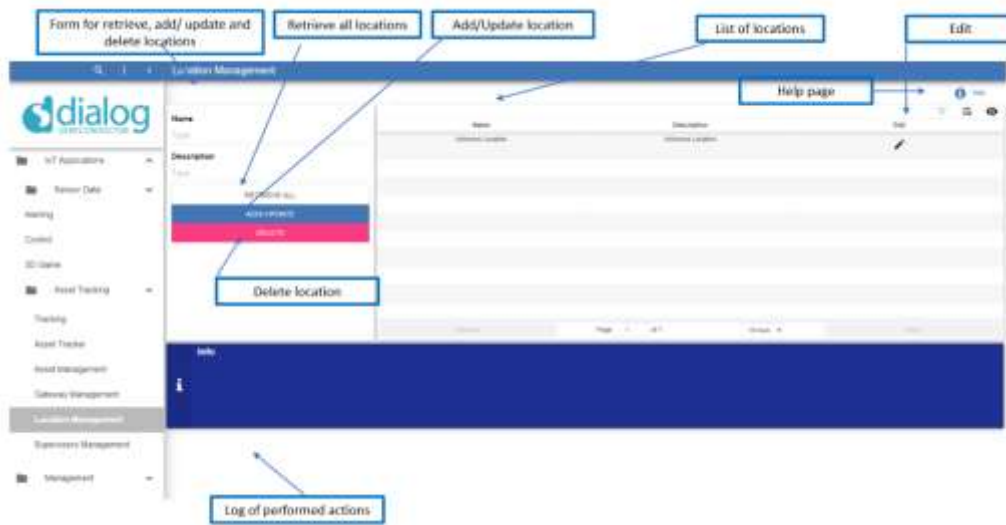


Figure 44: Location Management

7.6.1.5 Supervisor Management

Users can manage the supervisors of assets from **Supervisor Management** tab under the **Asset Tracking** menu. To add a new supervisor, fill in the name and email address and press the **ADD/UPDATE** button (Figure 45). Users can also edit or delete an existing supervisor.

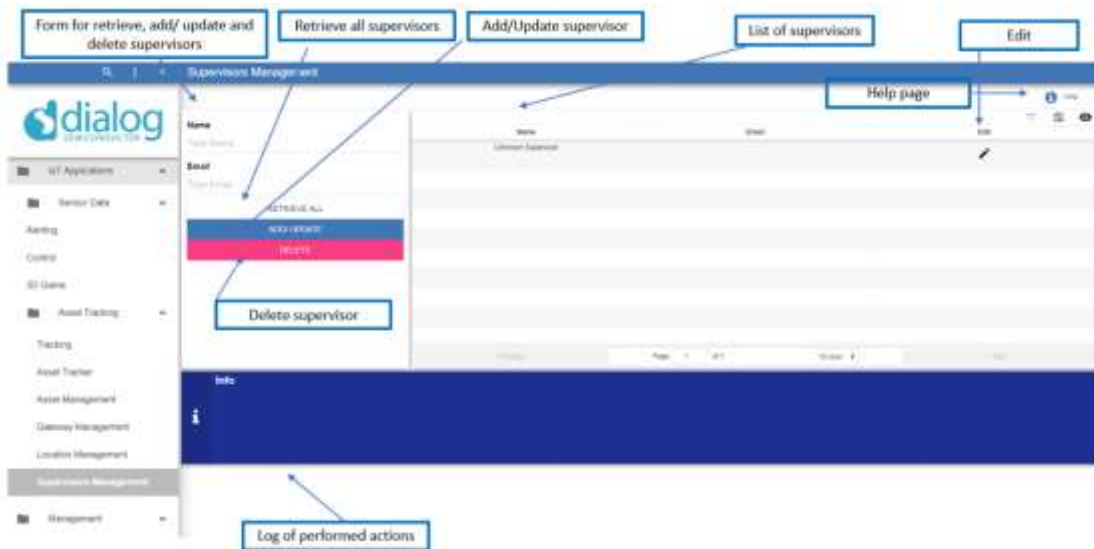


Figure 45: Supervisor Management

7.6.2 Mobile and Raspberry Pi Application

7.6.2.1 Mobile

To use a mobile device as a gateway of the Asset Tracking application, users should enable the cloud from cloud settings. Once a user account is created, a new gateway of the current mobile application is added to the available gateways of the user and it could be updated from the cloud application (go to Section 7.6.1.3).

An asset can be added only from the mobile application (Figure 46). To add a new asset, follow the steps below:

1. Enable **Asset Tracking** on the cloud settings of the **Dialog IoT Sensors App**
2. Select the **Asset Tracking** tab from the navigation bar of Cloud menu

Bring your mobile device near the desired IoT device, make sure that no other IoT devices are nearby, and then press the **SCAN** button. The **Device Id** field will be automatically filled with the BD address of the closest IoT device

3. Press the **SELECT** button to approve that this is your desired device.
4. Fill in a friendly name for the asset and then press the **APPLY** button.
5. A new asset is added to the available assets of the user and could be updated from the cloud application (go to Section 7.6.1.2).

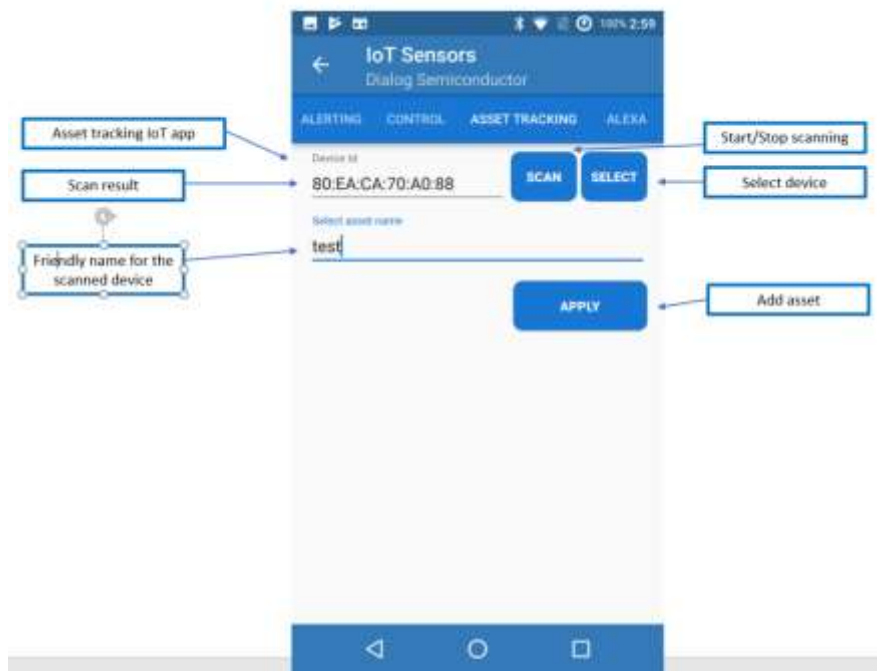


Figure 46: Dialog's IoT Sensors App, Asset Tracking Tab

DA14585 IoT Multi Sensor Development Kit

7.6.2.2 Raspberry Pi

A Raspberry Pi can be used as a gateway for the **Asset Tracking** application and for the **Data Gateway** functionality. When the Raspberry Pi is configured as an asset tracking gateway, it reports data from Dialog's IoT MSK Beacons to the cloud app, which then determines the position of Dialog's IoT MSK Beacons in an area. When the Raspberry Pi is configured as a Data Gateway, it connects to a Dialog's IoT MSK device and report all data to the cloud application. To configure the Raspberry Pi as a gateway, users should use the "DEKRaspberryGateway Python3 project". The project has been tested with the latest Raspbian 9 operating system.

There are two ways to set up the Raspberry Pi as a gateway:

- Download the "Raspberry Pi Gateway image" of the gateway from Dialog's customer support site and install it by following the official Raspberry Pi documentation:
<https://www.raspberrypi.org/documentation/installation/installing-images/>
- Or download the "Raspberry Pi Gateway Script Package" (DEKRaspberryGateway) from Dialog's customer support site. Transfer the `dekraspberrygateway_production_releaseb_vX.Y.zip` to the Raspberry Pi and unzip it. Edit `initial_setup.sh` script and change your time zone in line 9 of the script, for example, `sudo timedatectl set-timezone Europe/Athens`. Then execute the `initial_setup.sh` from a terminal: `$ sudo bash initial_setup.sh`

This will automatically install all required Python3 package dependencies to your system. If you choose to perform the installation manually, the required dependencies are listed in `requirements.txt` file, which can be found within the project directory and include:

- `libbluetooth3`
- `Python 3.x`.
- `enum34==1.1.6`
- `paho-mqtt==1.3.1`
- `PyBluez==0.22`
- `requests==2.18.4`
- `urllib3==1.22`
- `bluepy`

In the `/boot` partition of the produced image, the following files are present:

- `wpa_supplicant.conf`
- `wpa_supplicant.conf.sample`
- `dek_config.json`
- `dek_config.json.sample`

The `.sample` files have the same content as the `.conf` ones. End users need to edit `wpa_supplicant.conf` and/or `dek_config.json` to provide the necessary information for Wi-Fi and/or DEK gateway, respectively. After rebooting, these files will be moved to appropriate locations and no longer exist in `/boot`. This procedure can be repeated if necessary

The `.sample` files are provided to make it easier for end users to repeat the process without having to know the correct format for each file. Users only need to copy the `.sample` file to the same location without the `.sample` suffix in the filename and edit the file (for example, `dek_config.json.sample -> dek_config.json`). Each file can be configured independently. It is not required to configure both Wi-Fi and DEK gateway at the same time

The `dek_config.json` file includes user and gateway information. The configuration file of the project handles both **Asset Tracking** and **Data Gateway** applications

Users who want to use a Raspberry Pi as a gateway for "Asset Tracking Application" can update the configuration file by modifying it directly. The Role "asset_tracking" is pre-filled so only the `UserId` attribute needs to be updated with a desired user ID (Figure 47) to register the Raspberry

DA14585 IoT Multi Sensor Development Kit

Pi as a gateway in the specific account. The fields **BD address** and **Sensors** are only used in **Data Gateway** application, so they are left without a value. For more information see [Appendix B](#)

NOTE
The AppId attribute should not be set, because it is an auto-generated field.
Use quotes (") in the fields of the dek_config.json file.

The application will start automatically after reboot. In case you choose to run the application manually, run the following commands:

```
cd DEKRaspberryGateway
sudo python3 DEKClient.py
```

The gateway will be automatically added in the list of available gateways of a user after the user starts up the application.

```
1  {
2      "AppId": null,
3      "BdAddress": null ,
4      "Role": "asset_tracking",
5      "Sensors": "",
6      "UserId": "9cc821f4-26a0-2345-a19a-89a18265d6a4"
7  }
```

Figure 47: Configuration File of Raspberry Pi for Asset Tracking

```
1  {
2      "AppId": null,
3      "BdAddress": "80:EA:CA:67:A7:88",
4      "Role": "data_gateway",
5      "Sensors": "0",
6      "UserId": "9cc821f4-26a0-2345-a19a-89a18265d6a4"
7  }
```

Figure 48: Configuration File of Raspberry Pi for Data Gateway

7.6.2.3 Quick Start-up Guide

To use the IoT Asset tracking feature, users should follow the steps below:

1. Open the Dialog's IoT Sensors mobile app and connect it to an IoT device.
2. Enable cloud and then enable the Asset Tracking app from cloud settings under the Settings menu of the mobile app. By enabling the cloud, the mobile device is added as a user's gateway. Now the user can edit the gateway from the cloud application and assign a location that describes the current location of the gateway. To add a location, go to Section 7.6.1.4. To edit the gateway, go to Section 7.6.1.3.
3. Select the **Asset Tracking** tab from **Cloud** menu of the mobile app and add a new asset as described in Section 7.6.2.1.
4. Open the cloud application and assign a supervisor to the asset (see Section 7.6.1.2). By default, the "Unknown Supervisor" is assigned to a new asset. Users can create a new supervisor following the Section 7.6.1.5. A supervisor of an asset will receive notification emails if the asset is lost.
5. Set up the Asset Tracker widget by following Section 7.6.1.1.

After this procedure, users can monitor their assets from Asset Tracker or Tracking page of the cloud application.

7.7 3D Game

Using the IoT 3D Game application, users can use their IoT MSK's fusion sensor to control a three-dimensional ball-board control game in the cloud.

7.7.1 Cloud Application

To use the "3D Game" in the Cloud application (Figure 49), open the web app and go to the **3D Game** page under the **IoT Applications** folder. To start playing the game, select the IoT MSK device, press the **APPLY** button and then press the **Start 3D Demo** or the **Start 3D Game** button. In order to change the light intensity of the game, users can press the button on the IoT MSK device.



Figure 49: 3D Game IoT Application



Figure 50: 3D Demo

DA14585 IoT Multi Sensor Development Kit



Figure 51: 3D Ball-Board Control Game

7.8 Amazon Alexa

Alexa is Amazon's voice control system. She listens to users' requests and carries them out, at least simple ones, like dimming lights or playing music tracks.

IMPORTANT NOTE

To use Amazon Alexa, users have to install Amazon Alexa mobile application.

7.8.1 Skill Activation

To activate the Alexa voice service, enable the **Dialog Evaluation Kit Skill** public skill using one of the two available options:

- Via a web browser
- Via Amazon's mobile application **Amazon Alexa**

Follow the instructions below to activate the public skill.

- Via browser
 - a. Go to this link: <https://www.amazon.com/b?ie=UTF8&node=13727921011>
 - b. Login with your amazon account.
 - c. Search for "Dialog Evaluation Kit Skill".
 - d. Select and enable the skill (Figure 52).
 - e. Give your consent for the application to have access to your profile email.

DA14585 IoT Multi Sensor Development Kit



Figure 52: Dialog Evaluation Kit Skill for Alexa

- Via mobile application
 - a. Follow the instructions provided by the "Amazon Alexa" application.
 - b. Login with your amazon account.
 - c. Go to the **Skills** menu.
 - d. Search for "Dialog Evaluation Kit Skill".
 - e. Select and enable the skill.
 - f. Give your consent for the application to have access to your profile email.

Once the public skill is successfully activated, [Figure 53](#) is displayed.



Figure 53: Alexa: Skill Activation Succeeded Screen

7.8.2 How to Use

The Alexa functionality can only be used through the Amazon Alexa mobile application (Alexa tab in the IoT Sensors mobile application is not operational).

The currently available functional requests through the Amazon Alexa mobile application are:

- Alexa, ask dialog cloud to switch light on
- Alexa, ask dialog cloud to switch light off
- Alexa, ask dialog cloud where is/are my {asset's friendly name}
- Alexa, ask dialog cloud to get {environmental/proximity/air quality/brightness} readings from {friendly name of connected EK device}

7.9 IFTTT

7.9.1 Create an IFTTT Applet

1. Go to <https://ifttt.com/> and sign up for account (Figure 54).



Figure 54: IFTTT Signup Page

2. After logging in to IFTTT, click on **My Applets** at the top menu, and then click on the **New Applet** button (Figure 55).

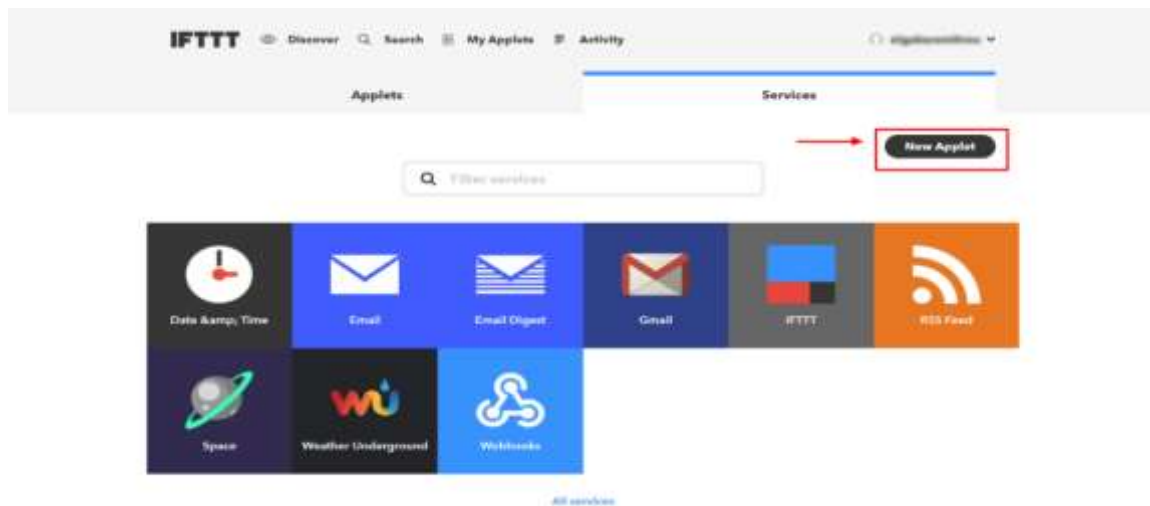


Figure 55: New Applet

3. On the **New Applet** screen, make an applet as shown in Figure 56.

DA14585 IoT Multi Sensor Development Kit

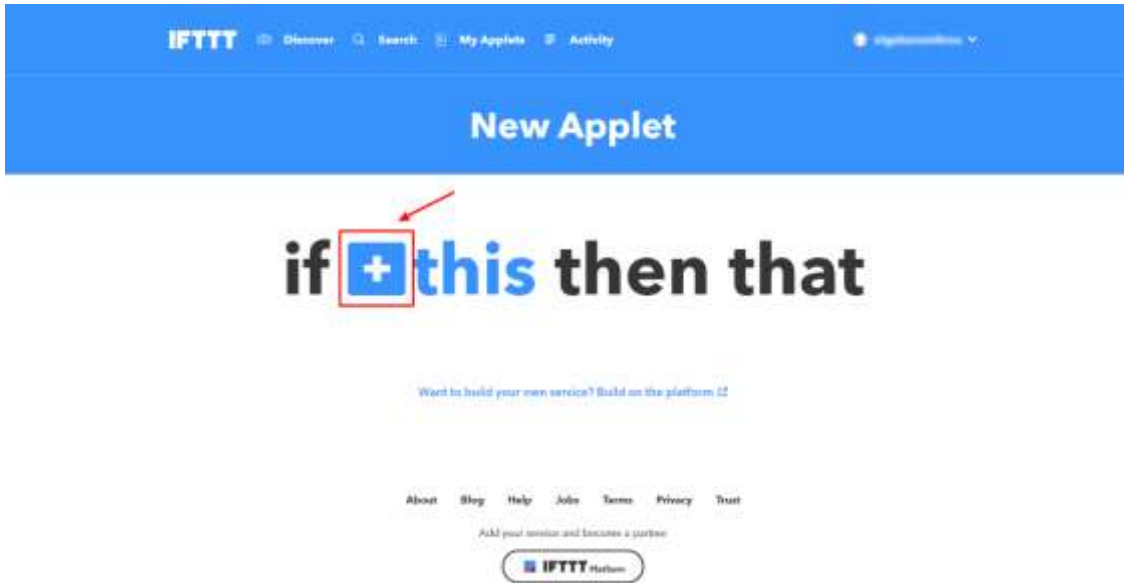


Figure 56: Make an IFTTT Applet

4. Type in "Webhooks" in the search field and choose the Webhooks service (Figure 57).

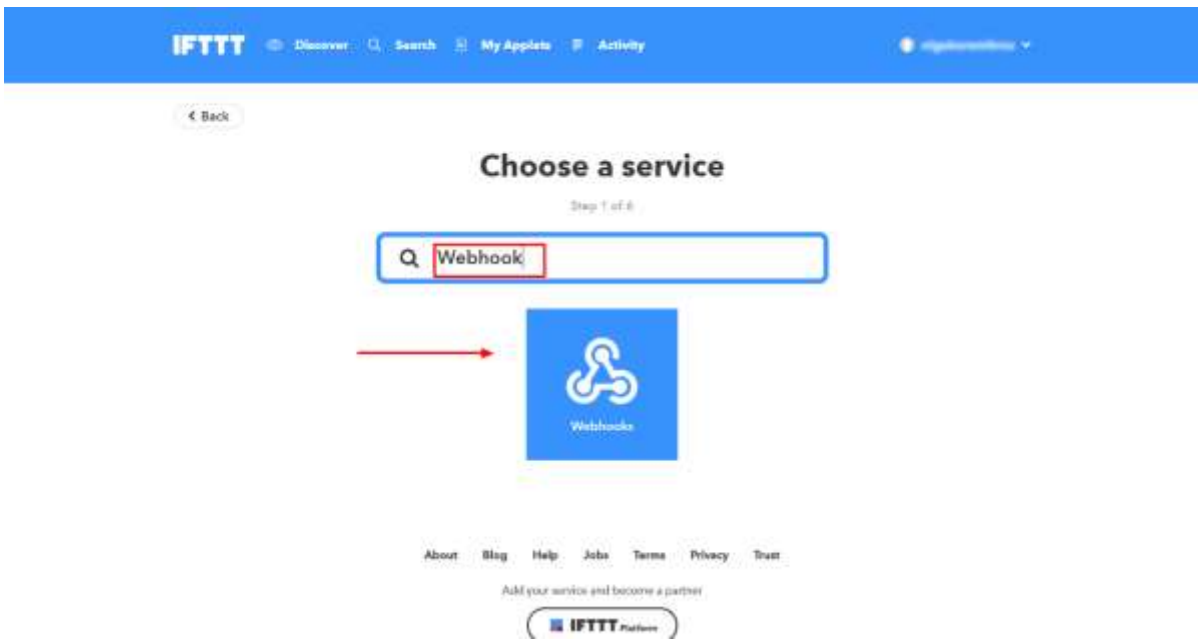


Figure 57: Make an IFTTT Applet Step 1

5. On the **Choose Trigger** screen, click on **Receive a web request** (Figure 58).

DA14585 IoT Multi Sensor Development Kit

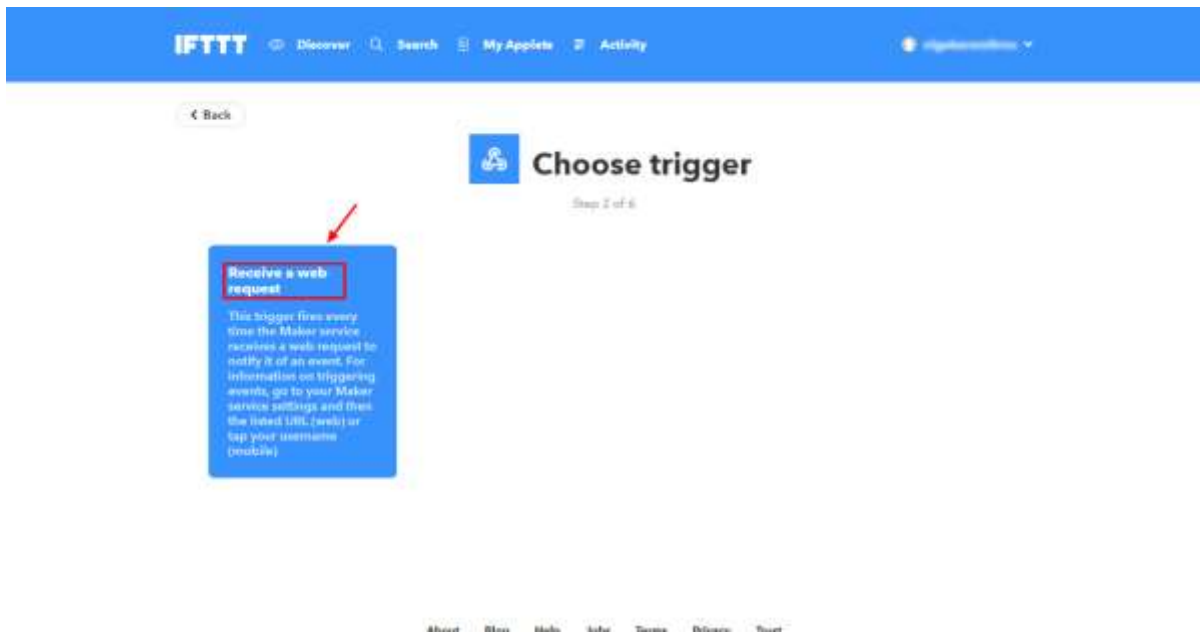


Figure 58: Make an IFTTT Applet Step 2

- On the **Complete trigger fields** screen, type in one of the supported event names and click the **Create trigger** button (Figure 59). Dialog's IoT Sensors app sends events at four different routes as it supports the following event names: temperature, humidity, pressure, and button.

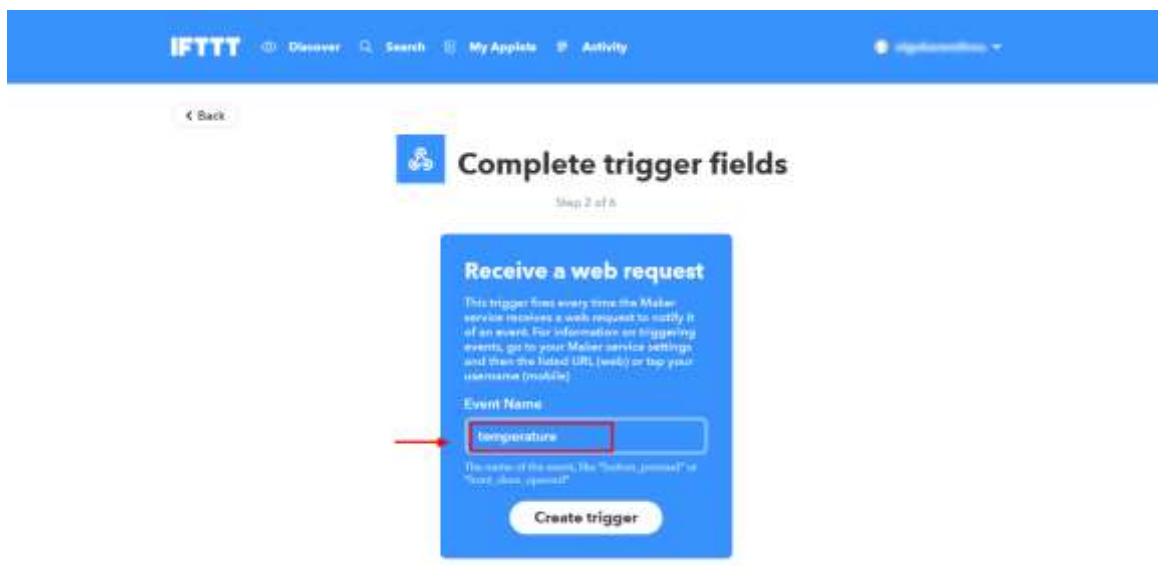


Figure 59: Make an IFTTT Applet, Complete Step 2

- Follow the instruction on Figure 60.

DA14585 IoT Multi Sensor Development Kit



Figure 60: Make an IFTTT Applet, Create That

- 8. On the **Choose action service** screen, type in "email" in the search field and click the **Email icon**, as shown in Figure 61.

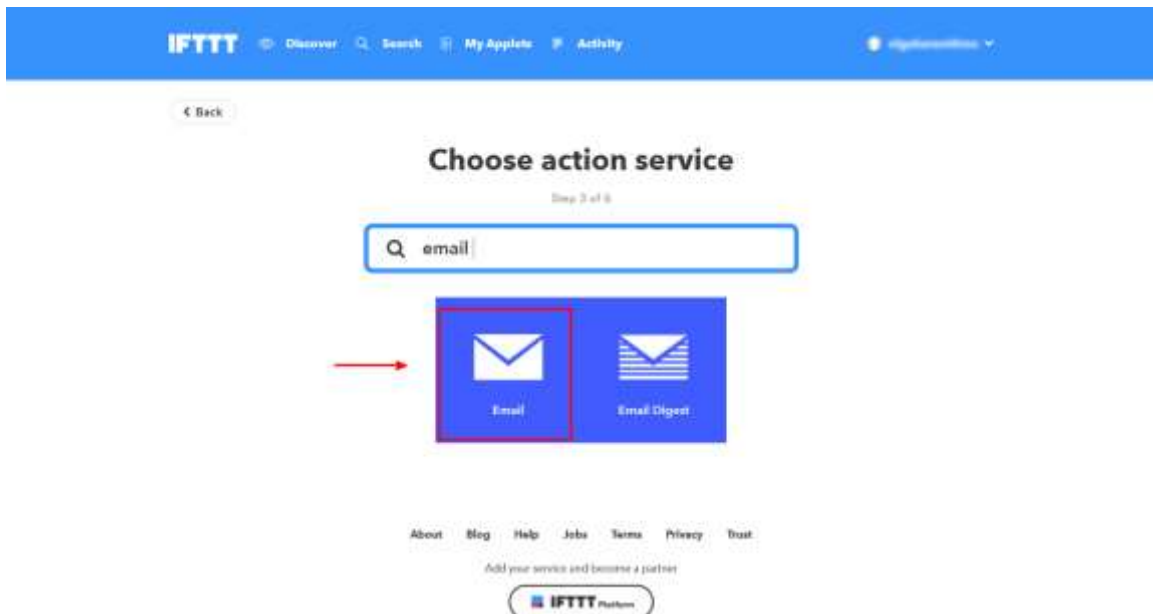


Figure 61: Make an IFTTT Applet, Step 3

- 9. On the **Choose Action** screen, click on **Send me an email** action (Figure 62).

DA14585 IoT Multi Sensor Development Kit



Figure 62: Make an IFTTT Applet, Step 4

10. On the **Complete action fields** screen, click the **Create action** button (Figure 63).

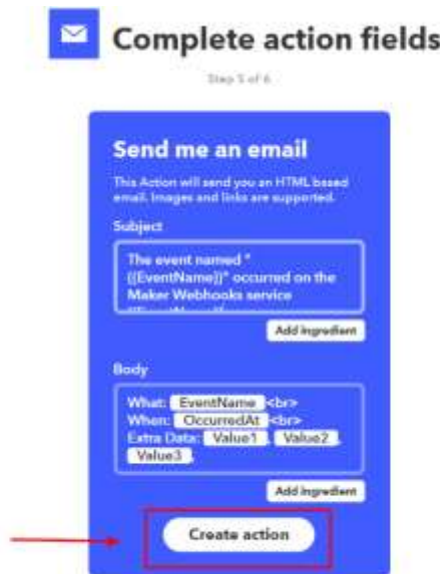


Figure 63: Make an IFTTT Applet Step 5

11. The IFTTT applet has been created. Enable the switch to receive notifications and click **Finish** button on the **Review and finish** screen (Figure 64). To receive events for all the supported event names (temperature, humidity, pressure, and button), repeat the above procedures and add as many applets as the supported events names.

DA14585 IoT Multi Sensor Development Kit



Figure 64: Make an IFTTT applet, Finish

7.9.2 IFTTT Web Hook Key

1. To find your key, go to **My Applets**, click the **Services** tab, and then click **Webhooks** (Figure 65).

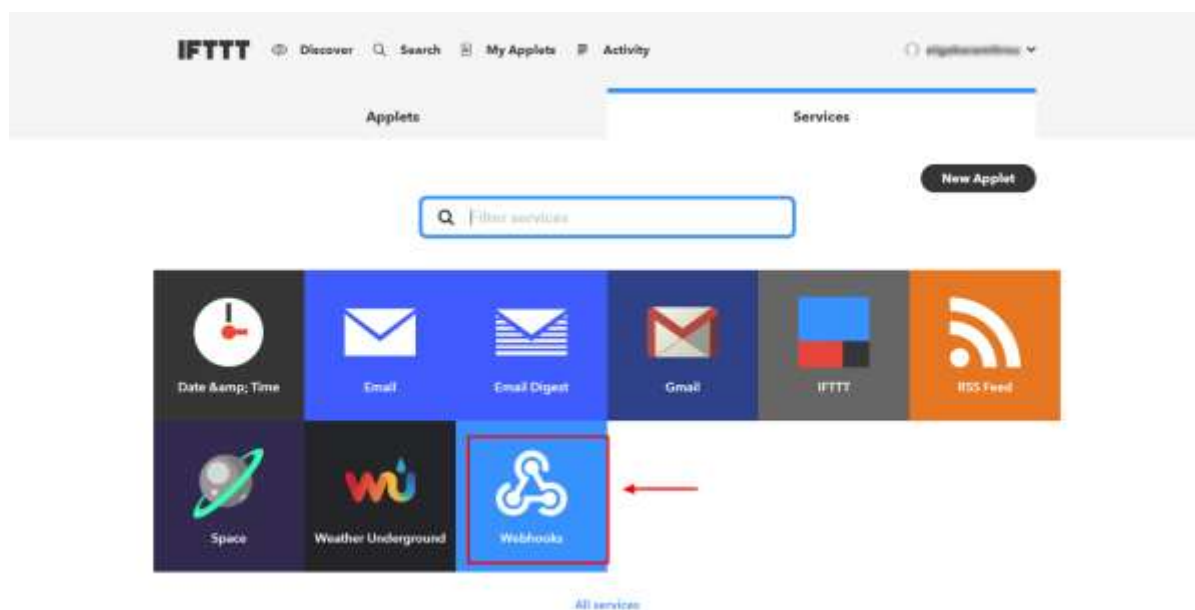


Figure 65: IFTTT Web Hook Key, Step 1

2. In the **Webhooks** screen click **Documentation** (Figure 66).

DA14585 IoT Multi Sensor Development Kit

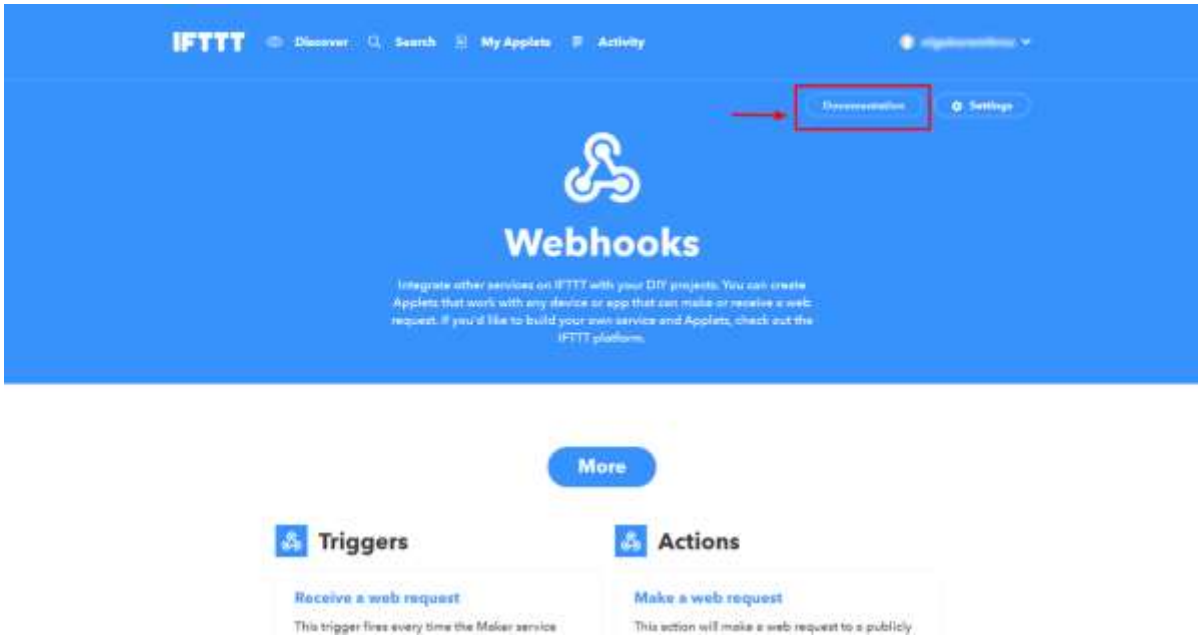


Figure 66: IFTTT Web Hook Key Step 2

3. The key will open in a new screen, as shown in Figure 67.



Figure 67: IFTTT Web Hook Key Step 3

7.9.3 Web App - IFTTT Settings

Once users have created a new webhook applet and retrieve the corresponding web hook key, they can update the IFTTT settings in the following step (Figure 68):

1. Open the web app.
2. Go to settings.
3. Add the key in the **IFTTT Web Hook Key** field and click the **ADD/UPDATE** button.



Figure 68: Web App - IFTTT Settings

7.9.4 Mobile App: IFTTT Settings

To enable IFTTT and start sending events using the created applet, follow the following steps (Figure 69):

1. Go to **Settings** screen in Dialog's IoT Sensors via the side menu.
2. Click on **Cloud settings**.
3. Insert the web hook API key or retrieve it from cloud. To add a web hook API key from the web application, see Section 7.9.2.
4. Enable IFTTT switch in **General** and sensors from which you would like to receive events in **Sensor Selection**.

By default, events will be received every 60 seconds for environmental sensors (temperature, humidity, and pressure). To change the event period, update the event period from IFTTT settings as shown in Figure 69.

Figure 70 shows the template of the email that users will receive once they have enabled the IFTTT and have created the "temperature" web hook event.

DA14585 IoT Multi Sensor Development Kit

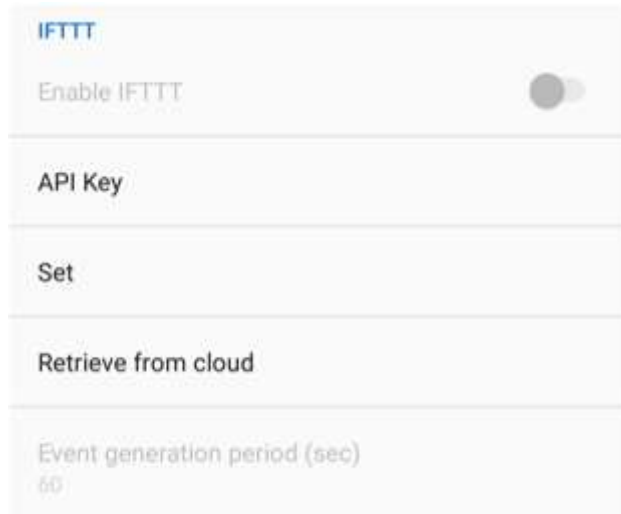


Figure 69: IFTTT Settings - Mobile App



Figure 70: Email When "Temperature" Occurred

Appendix A Troubleshooting

A.1 Magnetometer or Sensor Fusion Have Inconsistent Behaviors

Inconsistent behaviors of magnetometer or Sensor Fusion usually means that the magnetometer is not calibrated, or the magnetometer is calibrated but the environmental magnetic field has changed.

See Sections [4.3.4](#) and [4.3.5](#) for how to calibrate the magnetometer.

A.2 Device Does Not Connect after Switching to a Different Firmware

When switching between different references applications supported by DA14585 IoT MSK by directly burning one in Flash using the CIB or via SUOTA, users may experience connectivity or functionality issues with the respective Dialog's central application. For example, when customers switch to IoT Sensors after using Smart Tags, the IoT Sensors central application might not be able to establish connection.

This issue is caused by the Android/iOS BLE stack and a possible solution is to un-pair the device and/or restart Bluetooth in the Android/iOS system Settings.

A.3 Proximity Sensor is always On

The proximity sensor can detect infrared returns from its IR emitter at angles up to 75° from its perpendicular axis, meaning that the "field of view" is approximately 150°. The plastic enclosure reflects some of the IR light back to the detector and the default thresholds of the proximity detector (Section [4.3.3](#)) have been carefully chosen to avoid this problem.

However, due to PCB and enclosure tolerances, the default values might not work as expected. Users can solve this issue by increasing the thresholds of the proximity detector (Section [4.3.3](#)).

A.4 Air Quality Index is not Working

The AQI is not compiled within the default firmware shipped with the device due to code constraints. Users need to build a version that includes the Bosch BSEC library to make the AQI work. For more information see *section 4.2.4* of Ref. [\[18\]](#).

A.5 Magnetometer is not Working

The default setting of the magnetometer is off. It should be enabled from the basic settings menu (Section [4.3.3](#)). If it is used for the first time it should be calibrated. See Sections [4.3.4](#) and [4.3.5](#) for how to calibrate the magnetometer.

Appendix B Raspberry Pi for Data Gateway

B.1 Data Gateway Application

Raspberry Pi can also be used as a gateway for the Data Gateway application. In this application, the Raspberry Pi connects to a Dialog's IoT MSK device, receives sensor data from it, and uploads them to Dialog's Cloud. Users can choose data from which sensor to be uploaded to Dialog's Cloud Service. The uploaded sensor data can be viewed through the "Live data" application (see Section 7.3.1) or the "Historical Data" application (see Section 7.3.2). Sensor Fusion application can also use Raspberry Pi as a data gateway.

B.2 Configuration File for Asset Tracking and Data Gateway Application

As mentioned in Section 7.6.2.2, the configuration file handles both "Asset Tracking" and "Data Gateway" applications. In this appendix an additional way of updating the configuration file is presented (online form).

Users can update a configuration file in two ways:

- By modifying the configuration file directly:
 - For the role "asset_tracking", go to Section 7.6.2.2
 - For the role "data_gateway", users must add "data_gateway" to the role field and fill the "UserId", "BD Address", and "Sensors" fields (Figure 48)

"Sensors" attribute is only used in Data Gateway and defines which sensors will be uploaded to Dialog's Cloud Service. Every sensor has a unique number id. Users have to add the number IDs of the sensors to be uploaded to Cloud to the "Sensors" field. The number ID of each sensor is:

 - Temperature: 1
 - Humidity: 2
 - Pressure: 3
 - Light sensor :5
 - Proximity:6
 - Fusion:8
 - Button: 9
 - All the above sensors: 0

For example, if a user chooses to upload the data from the temperature and the humidity sensors, the "Sensors" field should be filled with "12"

"BD Address" attribute includes the address of the Dialog's IoT MSK device used for the "Data Gateway" application
- By submitting an online form: By clicking the address **raspberrypi_address:8080** from a browser, users can fill and submit the form in Figure 71:

DA14585 IoT Multi Sensor Development Kit

DEK CONFIGURATION

BD ADDRESS:

USER ID:

ROLE:
 Asset Tracking Data Gateway

CHOOSE SENSORS FOR DATA GATEWAY:

<input type="checkbox"/> Temperature	<input type="checkbox"/> Humidity
<input type="checkbox"/> Pressure	<input type="checkbox"/> Ambient light
<input type="checkbox"/> Proximity	<input type="checkbox"/> Fusion
<input type="checkbox"/> Button	<input type="checkbox"/> All the above

© 2018 Copyright Dialog Semiconductor

Figure 71: Online Configuration Form

By clicking the submit button, the configuration file (`config.json`) will be updated with the new values. After submitting the form, the Raspberry Pi reboots and starts running with the new settings. Next time users click the address of the form, the form fields will be prefilled with the current values of the configuration file.

Revision History

Revision	Date	Description
1.3	16-Feb-2022	Updated Sections 7.8 and 7.8.2 . Updated logo, disclaimer, copyright.
1.2	01-Aug-2019	Updates for Raspberry Pi Gateway Updated the legal disclaimer in the back page
Change details: For changes: <ul style="list-style-type: none"> ● Section 7.6.2.2 <ul style="list-style-type: none"> ○ Updated description how to set up Raspberry Pi Gateway ● Appendix B <ul style="list-style-type: none"> ○ Configuration files of Raspberry Pi for Data Gateway functionality 		
1.1	05-Feb-2019	Updates for 6.160.4
Change details: For changes: <ul style="list-style-type: none"> ● Section 4.4 <ul style="list-style-type: none"> ○ Remarks about the inclusion of AQI Bosch BSEC Library ● Section 7 <ul style="list-style-type: none"> ○ Various additions and improvements for the Cloud UI and Alexa Voice Control ● Added new troubleshooting appendixes A.4 and A.5 		
1.0	03-Aug-2018	Initial version.

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.