AN1030: WPA2/WPA Enterprise

This document describes how to connect to a network using WPA2/WPA Enterprise security features in the Wizard Gecko WGM110 Wi-Fi Module.

KEY POINTS

- Setting up
  - PEAP-MSCHAPv2
  - EAP-TLS
- Testing WPA2/WPA Enterprise functionality on WGM110 using BGTool
- Connecting to an Access Point using
  - PEAP-MSCHAPv2
  - EAP-TLS
- Troubleshooting advice

1. Introduction

This application note introduces the usage of WPA2/WPA Enterprise features in the WGM110. WPA2/WPA Enterprise is an extension of Wi-Fi Protected Access, requiring an authentication server e.g. RADIUS, designed for corporate networks requiring extra security which the normal pre-shared passkey networks cannot provide. The security setting requires an Extensible Authentication Protocol (EAP) type to be used for authentication.

The WGM110 offers two EAP types, PEAP-MSCHAPv2 and EAP-TLS. WPA2/WPA Enterprise security is only supported in client/station mode with the WGM110.

2. PEAP-MSCHAPv2

MSCHAPv2 is an insecure authentication method on its own but PEAP creates a secure tunnel between the devices using TLS. The WGM110 supports PEAP v0 and v1. Client authentication using username and password is mandatory in PEAP-MSCHAPv2. To verify the identity of the authentication server it uses a CA (certificate authority) certificate, also commonly referred to as root certificate. Verifying the identity of the client using a certificate is optional in the authentication server.

Note: For more information on how to load certificates into WGM110, please refer to *AN974: TLS and SMTP*.

To setup the WGM110 client to connect to a network using PEAP-MSCHAPv2, you must configure the MSCHAP username, password and the outer identity.

Before connecting to a WPA2/WPA Enterprise enabled network access point, the configurations need to be activated in the WGM110. Using the BGAPI EAP commands, like in the example below, the username, password and other EAP required settings are activated.

Once these configurations have been entered successfully to the stack, a network connection can be established with the connect SSID or connect BSSID commands. Please note that the debug output is optional.

```
# SSID
ssid_len = 9
ssid(0:ssid_len) = "test_ssid"

# MSCHAPv2 username
username_len = 4
username(0:username_len) = "test"

#MSCHAPv password
password_len = 8
password(0:password_len) = "testtest"

# Outer identity
identity_len = 9
identity(0:identity_len) = "anonymous"

# Endpoint for debugging via UART0
output_ep = 0

# Activate MSCHAPv2 username
call sme_set_eap_type_username (sme_eap_type_mschapv2, username_len, username(0:username_len))(cmd_result)

if cmd_result != 0
  # Debug output
  call endpoint_send(output_ep, 31, "EAP: invalid MSCHAPv2 username\r\n")
end if

# Activate MSCHAPv2 password
call sme_set_eap_type_password(sme_eap_type_mschapv2, password_len, password(0:password_len))(cmd_result)

if cmd_result != 0
  # Debug output
  call endpoint_send(output_ep, 31, "EAP: invalid MSCHAPv2 password\r\n")
end if

# Activate EAP configuration to PEAP-MSCHAPv2
call sme_set_eap_configuration(sme_eap_type_peap, sme_eap_type_mschapv2, identity_len,
identity(0:identity_len))(cmd_result)

if cmd_result != 0
  # Debug output
  call endpoint_send(output_ep, 32, "EAP: invalid EAP configuration\r\n")
end if

# Connect to the network.
# This call will trigger either sme_connected() event if the attempt succeeds or
# sme_connect_failed() event if it fails.
call sme_connect_ssid(ssid_len, ssid(0:ssid_len))(cmd_result, cmd_interface, cmd_bssid)

if cmd_result != 0
  # Debug output
```

```
  call endpoint_send(output_ep, 34, "EAP: Wi-Fi connect command failed\r\n")
end if
```

The WGM110 then checks the internal certificate store for a suitable certificate to validate the identity of the authentication server, but this can also be set explicitly with the command `sme_set_eap_type_ca_certificate()`.

3. EAP-TLS

EAP-TLS also uses a CA certificate to verify the identity of the authentication server. Unlike PEAP-MSCHAPv2 it is mandatory for the authentication server to verify the identity of the client, which requires a user certificate and a private key. Username and password are not used in EAP-TLS so the commands **sme_set_eap_type_password()** and **sme_set_eap_type_username()** should not be used.

The user certificate can be stored in flash or RAM but the corresponding private key can only be stored in RAM due to security reasons and therefore it can only be initialized using API commands. The private key should be stored in a secure external storage and sent to the module either by directly calling the required BGAPI commands from an external host or as plain data in which case the commands must be called through BGScript. The corresponding user certificate fingerprint must be provided when loading the private key, as well as the password used to encrypt the private key data. If the private key is sent unencrypted then the password length should be 0.

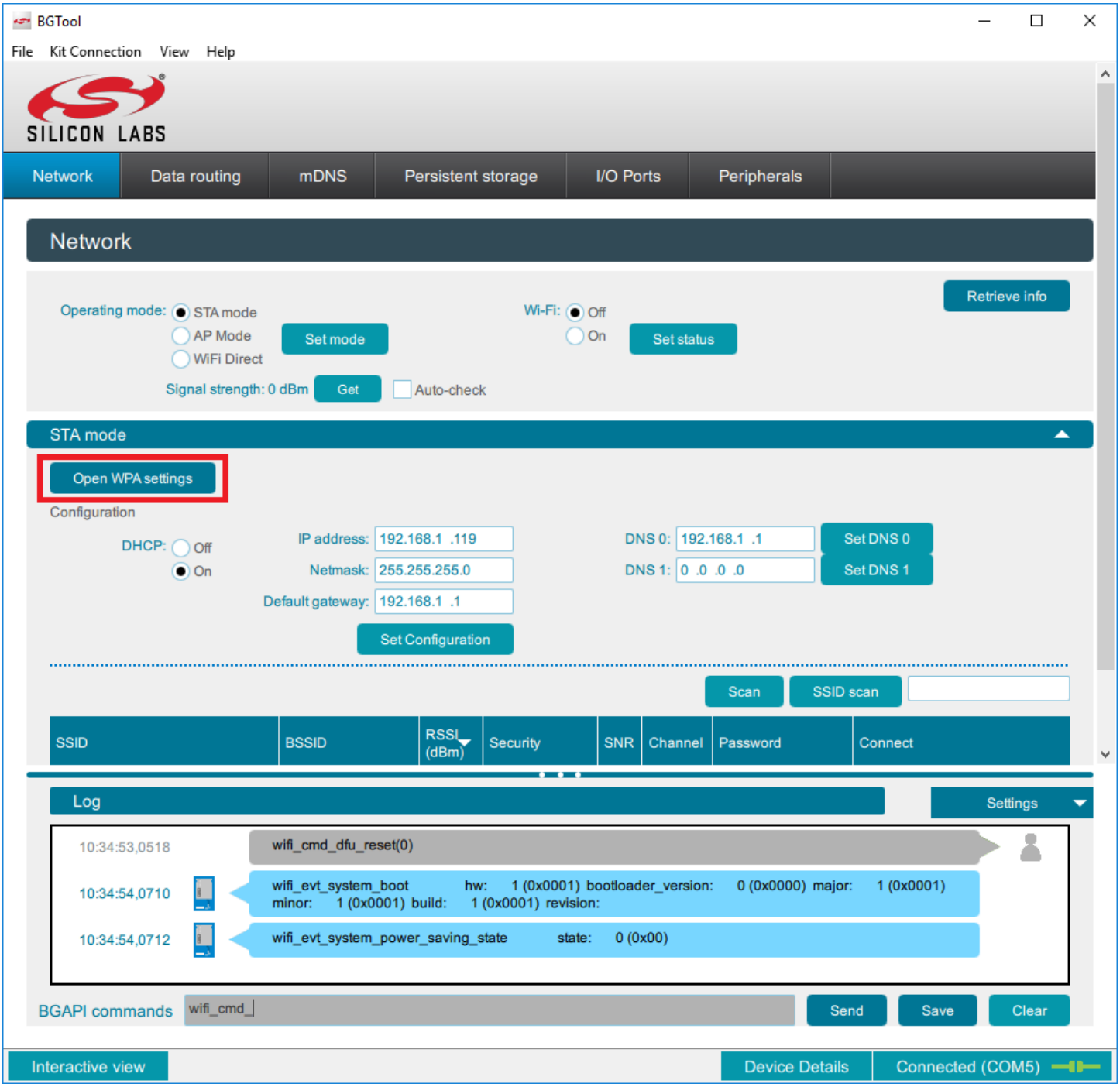
Once both the certificate and private key have been initialized then the certificate must be set as a user certificate. That will select both the certificate and private key, which have already been associated with each other (by providing the associated certificate fingerprint when loading the private key).

When the certificate has been set as user certificate and the EAP settings activated the network connection can be established with the connect SSID or connect BSSID commands as shown in the example below.

```
# SSID
ssid_len = 9
ssid(0:ssid_len) = "test_ssid"

# Outer identity
identity_len = 9
identity(0:identity_len) = "anonymous"

# Endpoint for debugging via UART0
output_ep = 0

# The following three commands are used to add a private key to the certificate store.
The fingerprint given in the command parameters must be the associated user certificate.
call x509_add_private_key(size, fingerprint_len, fingerprint_data)

# The next command must be called multiple times until all the private key data has been added.
call x509_add_private_key_data(data_len, data_data)

# Finally when all the data has been added the following command must be called
call x509_add_private_key_finish(password_len, password)

# Now we need to set the certificate as a user certificate using its fingerprint which will
automatically select its associated private key.
call sme_set_eap_type_user_certificate(sme_eap_type_tls, fingerprint_len, fingerprint_data)

# Activate EAP configuration to EAP-TLS
call sme_set_eap_configuration(sme_eap_type_tls, sme_eap_type_none, identity_len,
identity(0:identity_len))(cmd_result)

if cmd_result != 0
  # Debug output
  call endpoint_send(output_ep, 32, "EAP: invalid EAP configuration\r\n")
end if

# Connect to the network.
This call will trigger either sme_connected() event if the attempt succeeds or
sme_connect_failed() event if it fails.

call sme_connect_ssid(ssid_len, ssid(0:ssid_len))(cmd_result, cmd_interface, cmd_bssid)

if cmd_result != 0
  # Debug output
  call endpoint_send(output_ep, 34, "EAP: Wi-Fi connect command failed\r\n")
end if
```

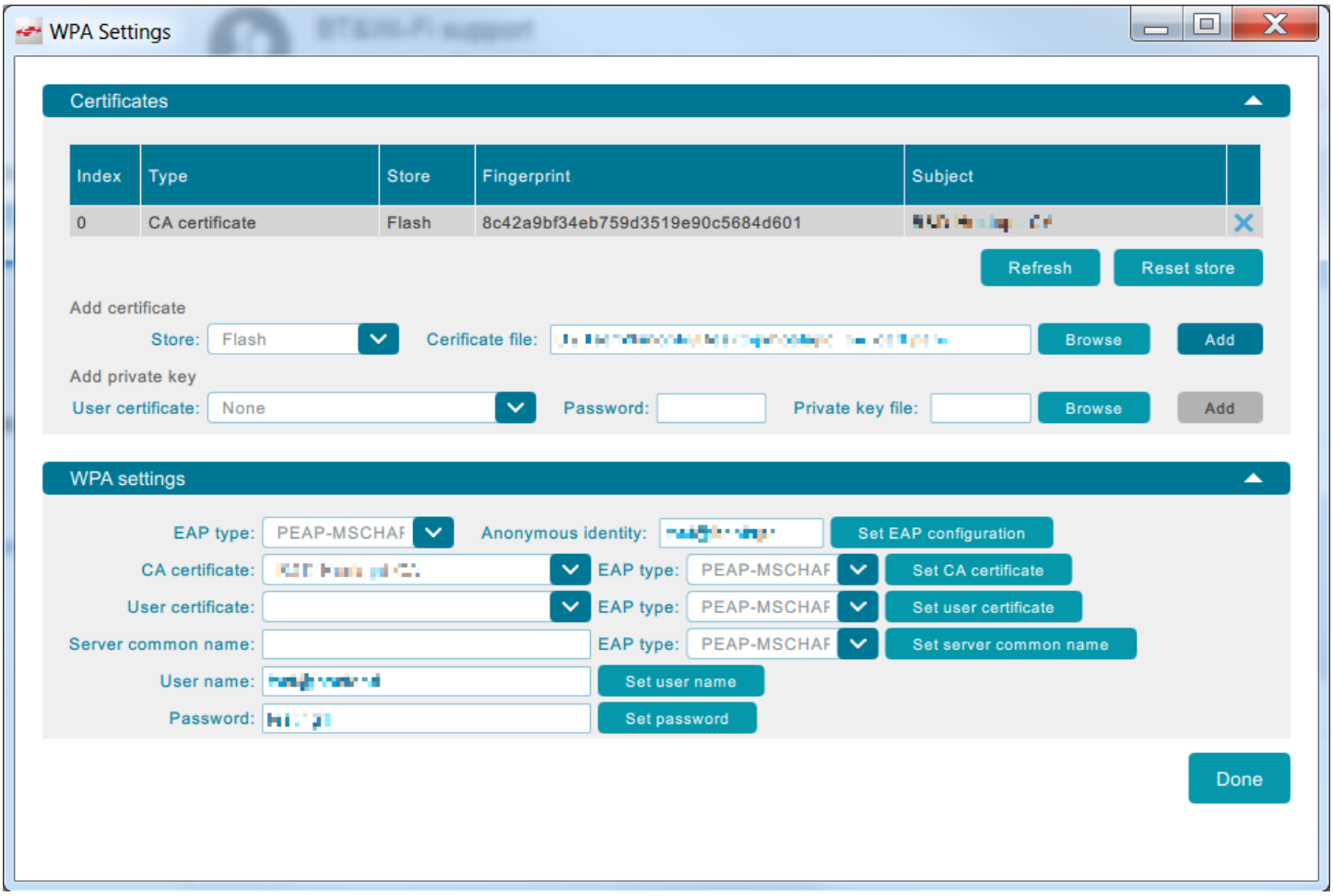
4. Using BGTool

BGTool can be used to test WPA2/WPA Enterprise functionality on WGM110. The WPA settings can be accessed by clicking on [**Open WPA settings**] button in the *"STA mode"* section as depicted below.



4.1 Connecting to an AP using PEAP-MSCHAPv2

To connect to a network using PEAP-MSCHAPv2 requires a CA certificate, which can be loaded through the WPA Settings window by browsing for the certificate file and adding it to the certificate store through the [**Add**] button. One the certificate is loaded, it will be listed in the certificate store as depicted in the image below.



The *"EAP type"* must be selected as PEAP-MSCHAP and the anonymous identity written to the *"Anonymous Identity"* text box. Then launch the iAPI command `sme_set_eap_configuration()` by pressing the [**Set EAP configuration**] button.

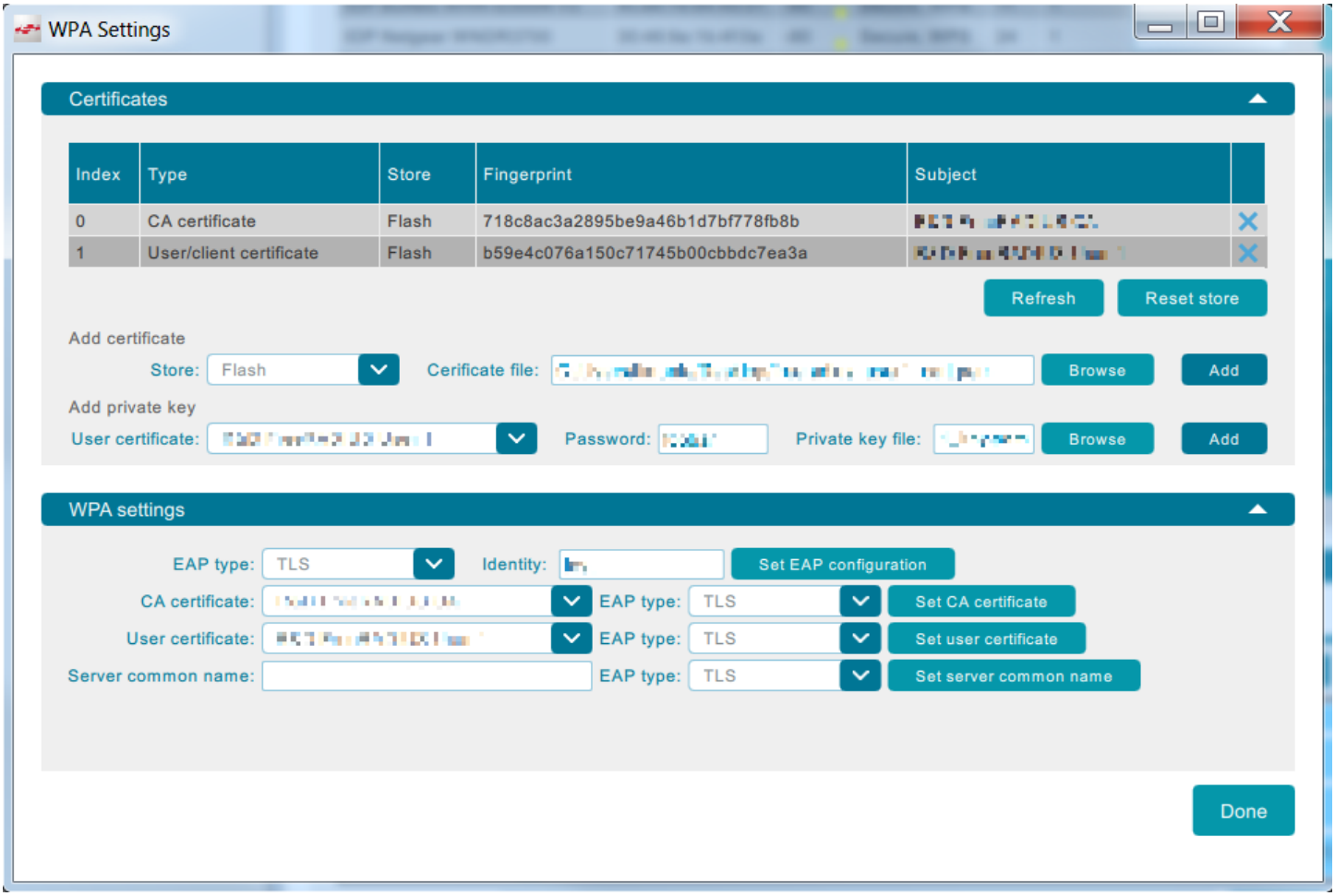
The CA certificate can be explicitly set to be the one that was loaded or the firmware can automatically look for the correct one as explained earlier in this document. To set the CA certificate it must be selected from the "*CA certificate*" drop-down list, "*EAP type*" must be set to PEAP_MSCHAP, and then you must click [**Set CA certificate**]. The user certificate is optional, and for this example the authentication server that is being used does not ask for client authentication.

Finally, the username and password must be written in to the "*User name*" and "*Password*" text input fields, and the buttons [**Set user name**] and [**Set password**] must be pressed.

Once the WPA settings are configured, the module is ready to connect to the network. This can be done back in the BGTool main window by scanning and selecting the network to which you wish to connect.

4.2 Connecting to an AP using EAP-TLS

To connect to a network using EAP-TLS requires a CA and user certificates. Additionally, the user private key must be also loaded, and it will be stored in RAM by default. To load the private key, the associated user certificate must be selected from the "User certificate" drop-down list, and the password and private key file must be given after. Clicking [**Add**] will run the private key loading command set.



The *"EAP type"* must be selected as TLS and the anonymous identity written to the *"Identity"* text input field. Then [**Set EAP configuration**] must be pressed.

The CA certificate can be explicitly set to be the one that was loaded or the firmware can automatically look for the correct one as explained earlier in this document. To set the CA certificate, it must be selected from the *"CA certificate"* drop-down list, *"EAP type"* must be set to **TLS**, and then press [**Set CA certificate**].

The user certificate is mandatory in EAP-TLS. The correct user certificate must be selected from the *"User certificate"* drop-down list, *"EAP type"* must be set to **EAP-TLS**, and then press [**Set user certificate**].

Once the WPA settings are configured, the module is ready to connect to the network. This can be done back in the BGTool main window by scanning and selecting the network to which you wish to connect.

4.3 Troubleshooting

If the connection is not successful, these are the most common errors that might occur and hints on what the root cause could be:

- 0x018B (ap_note_in_scanlist): The SSID of the AP is misspelled, length doesn't match the string size or the scan has been limited to a channel which is not being used by the AP.
- 0x081D (authentication failure): The WPA/WPA certificates have not been properly loaded or are missing, private key is missing (in case of EAP-TLS), EAP configuration is not set properly.

5. Revision History

5.1 Revision 0.2

September 15th, 2021

Updated BGTool graphic.

5.2 Revision 0.1

August 3rd, 2016

Initial release.

Smart. Connected. Energy-Friendly.

IoT Portfolio
www.silabs.com/products

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

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

www.silabs.com