Introduction

This document details the release notes for ATSAMB11 BluSDK Smart v6.2.
# Table of Contents

**Introduction**                                                                                     ...................................................... 1

1. BluSDK Smart Software Architecture Overview                                                        ...................................................... 4

2. Supported Hardware Platforms and IDEs                                                            ...................................................... 5

3. BluSDK Smart Release Contents                                                                   ...................................................... 6

4. BluSDK Smart Firmware Contents                                                                  ...................................................... 8

5. BluSDK Smart Release Features                                                                   ...................................................... 11

6. Release Revision History                                                                        ...................................................... 13
   6.1. BluSDK Smart v6.2                                                                           ...................................................... 13
   6.2. BluSDK Smart v6.1                                                                           ...................................................... 13
   6.3. BluSDK Smart v5.2                                                                           ...................................................... 14
   6.4. BluSDK Smart v5.1                                                                           ...................................................... 14
   6.5. BluSDK Smart v5.0                                                                           ...................................................... 15
   6.6. BluSDK Smart v4.1                                                                           ...................................................... 16
   6.7. BluSDK Smart v4.0                                                                           ...................................................... 16
   6.8. BluSDK Smart v3.0                                                                           ...................................................... 16
   6.9. BluSDK Smart v2.1                                                                           ...................................................... 17
   6.10. BluSDK Smart v2.0                                                                          ...................................................... 17

7. Known Issues                                                                                      ................................................................ 18

8. Limitations                                                                                      ................................................................ 20
   8.1. BluSDK Smart Software                                                                       ...................................................... 20
   8.2. Mobile Applications                                                                         ...................................................... 21

9. Additional Information/Pre-requisites                                                            ...................................................... 22
   9.1. How to Install ATSAMB11 Device Pack using Device Pack Manager                              ...................................................... 23
   9.2. How to Install SAMB11 Device Pack with atpack File                                         ...................................................... 25
   9.3. Measurement of Power Consumption                                                           ...................................................... 27

10. Hardware Resources used by BluSDK-SMART                                                          ................................................................ 31
   10.1. Timer                                                                                      ...................................................... 31

11. Appendix 1 – BluSDK SMART Change Log                                                              ................................................................ 32
   11.1. BluSDK Smart Module Changes                                                                 ................................................................ 32
   11.2. BLE Library API – New Features                                                             ................................................................ 33
   11.3. Difference between BLE APIs – BluSDK Smart 6.2 versus BluSDK Smart 5.2                    ................................................................ 34
   11.4. Recommended Code Implementation for Handling Platform Event                               ................................................................ 35

The Microchip Web Site                                                                           ................................................................ 39
1. **BluSDK Smart Software Architecture Overview**

The following diagram illustrates the various layers in the BluSDK Smart Architecture for implementing various applications.
2. Supported Hardware Platforms and IDEs

The following table provides the supported hardware platforms and IDEs of BluSDK Smart.

<table>
<thead>
<tr>
<th>Device</th>
<th>Development Kit</th>
<th>Supported IDEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSAMB11G18A</td>
<td>ATSAMB11 Xplained Pro</td>
<td>Atmel Studio v7.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keil uVision v5.14.0.0 and Above</td>
</tr>
<tr>
<td>ATSAMB11ZR</td>
<td>ATSAMB11 ZR Xplained Pro</td>
<td>Atmel Studio v7.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keil uVision v5.14.0.0 and Above</td>
</tr>
</tbody>
</table>
3. **BluSDK Smart Release Contents**

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Layer API Documentation</td>
<td></td>
</tr>
<tr>
<td>Driver API Documentation</td>
<td></td>
</tr>
<tr>
<td>Software Development Guide</td>
<td></td>
</tr>
<tr>
<td>Interrupts and ULP Guide</td>
<td></td>
</tr>
<tr>
<td>BLE Example Application Getting Started Guide: This includes Getting started guides for following example application.</td>
<td></td>
</tr>
<tr>
<td>1. Proximity Monitor Profile Example (pxp_monitor)</td>
<td></td>
</tr>
<tr>
<td>2. Proximity Reporter Profile Example (pxp_reporter)</td>
<td></td>
</tr>
<tr>
<td>3. Apple Notification Centre Service Example (ANCS)</td>
<td></td>
</tr>
<tr>
<td>4. Battery Service Example (battery_info)</td>
<td></td>
</tr>
<tr>
<td>5. Observer Example (ble_observer)</td>
<td></td>
</tr>
<tr>
<td>6. Device Information Service Example (device_info)</td>
<td></td>
</tr>
<tr>
<td>7. Health Thermometer Profile Example (htpt_temperature_sensor)</td>
<td></td>
</tr>
<tr>
<td>8. Time Information Profile Example (time_information)</td>
<td></td>
</tr>
<tr>
<td>9. Scan Parameter Example (scan_parameter)</td>
<td></td>
</tr>
<tr>
<td>10. Simple Broadcaster Example (simple_broadcaster)</td>
<td></td>
</tr>
<tr>
<td>11. Simple BLE Connection Example (simple_btlc1000_connection)</td>
<td></td>
</tr>
<tr>
<td>12. HID Keyboard Example (hid_device_keyboard)</td>
<td></td>
</tr>
<tr>
<td>13. HID Mouse Example (hid_mouse_keyboard)</td>
<td></td>
</tr>
<tr>
<td>14. Alert Notification Profile Example (alert_notification)</td>
<td></td>
</tr>
<tr>
<td>15. Blood Pressure Profile Example (blood_pressure)</td>
<td></td>
</tr>
<tr>
<td>16. Custom Profile Example (custom_serial_chat)</td>
<td></td>
</tr>
<tr>
<td>17. Find me Profile Example (find_me)</td>
<td></td>
</tr>
<tr>
<td>18. Heart Rate Profile Example (hr_sensor)</td>
<td></td>
</tr>
<tr>
<td>19. Phone Alert Status Profile Example (phone_alert_status)</td>
<td></td>
</tr>
<tr>
<td>20. Multi Role Multi Connect Example (multi_role_multi_connect)</td>
<td></td>
</tr>
<tr>
<td>21. Direct Test Mode Example</td>
<td></td>
</tr>
<tr>
<td>22. AT Command Example</td>
<td></td>
</tr>
<tr>
<td>23. OTAU Example</td>
<td></td>
</tr>
<tr>
<td>24. L2CAP Peripheral Example</td>
<td></td>
</tr>
<tr>
<td>25. L2CAP Central Example</td>
<td></td>
</tr>
<tr>
<td>26. Eddystone Beacon Example</td>
<td></td>
</tr>
<tr>
<td>27. Alt-Beacon Example</td>
<td></td>
</tr>
<tr>
<td><strong>SDK</strong></td>
<td>Example BLE Apps, BLE Profiles, BLE Services are integrated in ASF (vsix) and in Keil Project.</td>
</tr>
<tr>
<td>1. ASF (vsix)</td>
<td></td>
</tr>
<tr>
<td>Folder Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1.1.</td>
<td>SAMB11 Device Pack must be updated to version 2.3.190.</td>
</tr>
<tr>
<td>1.2.</td>
<td>ASF and DFP installation.</td>
</tr>
<tr>
<td>1.2.1.</td>
<td>If DFP version is not 2.3.190, refer following sections to update the DFP version to 2.3.190.</td>
</tr>
<tr>
<td>1.2.1.</td>
<td>– How to install SAMB11 device pack using Device Pack Manager or</td>
</tr>
<tr>
<td>1.2.1.</td>
<td>– How to install SABM11 device pack with atpack file</td>
</tr>
<tr>
<td>1.2.2.</td>
<td>use Atmel Studio Extension Manager (Tools -&gt; Extension Manager) or visit Atmel Gallery to update ASF in Atmel Studio 7.0.</td>
</tr>
</tbody>
</table>

2. Keil Project

2.1. No installation is required.

### Mobile Application

Contains the following mobile applications:


### PC Application

Wireless Composer **[BLE Performance Analyzer]**.  
OTAU_Image_Creator (Command line tool)
4. BluSDK Smart Firmware Contents

BluSDK Smart firmware builds and runs on a MCU host Table 2-1. The example profiles application userguide provides, how to find the example project using the ASF Example Project Wizard, build, Flash and execute the software.

Table 4-1. BluSDK Smart Firmware Contents

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apps</td>
<td>BLE Example Application Profiles and Services. Contains the following Example Applications:</td>
</tr>
<tr>
<td></td>
<td>1. Proximity Monitor Profile Example (pxp_monitor)</td>
</tr>
<tr>
<td></td>
<td>2. Proximity Reporter Profile Example (pxp_reporter)</td>
</tr>
<tr>
<td></td>
<td>3. Apple Notification Centre Service Example (ANCS)</td>
</tr>
<tr>
<td></td>
<td>4. Battery Service Example (battery_info)</td>
</tr>
<tr>
<td></td>
<td>5. Observer Example (ble_observer)</td>
</tr>
<tr>
<td></td>
<td>6. Device Information Service Example (device_info)</td>
</tr>
<tr>
<td></td>
<td>7. Health Thermometer Profile Example (htpt_temperature_sensor)</td>
</tr>
<tr>
<td></td>
<td>8. Time Information Profile Example (time_information)</td>
</tr>
<tr>
<td></td>
<td>9. Scan Parameter Example (scan_parameter)</td>
</tr>
<tr>
<td></td>
<td>10. Simple Broadcaster Example (simple_broadcaster)</td>
</tr>
<tr>
<td></td>
<td>11. Simple BLE Connection Example (simple_btlc1000_connection)</td>
</tr>
<tr>
<td></td>
<td>12. HID Keyboard Example (hid_device_keyboard)</td>
</tr>
<tr>
<td></td>
<td>13. HID Mouse Example (hid_mouse_keyboard)</td>
</tr>
<tr>
<td></td>
<td>14. Alert Notification Profile Example (alert_notification)</td>
</tr>
<tr>
<td></td>
<td>15. Blood Pressure Profile Example (blood_pressure)</td>
</tr>
<tr>
<td></td>
<td>16. Custom Profile Example (custom_serial_chat)</td>
</tr>
<tr>
<td></td>
<td>17. Find me Profile Example (find_me)</td>
</tr>
<tr>
<td></td>
<td>18. Heart Rate Profile Example (hr_sensor)</td>
</tr>
<tr>
<td></td>
<td>19. Phone Alert Status Profile Example (phone_alert_status)</td>
</tr>
<tr>
<td></td>
<td>20. BLE Startup Template (startup_template)</td>
</tr>
<tr>
<td></td>
<td>21. Multi Role Peripheral Multi Connect Example (multi_role_peripheral_multiconnect)</td>
</tr>
<tr>
<td></td>
<td>22. Direct Test Mode Connect Example (dtm_app)</td>
</tr>
<tr>
<td></td>
<td>23. AT Command Example (at_cmd_app)</td>
</tr>
<tr>
<td></td>
<td>24. Battery and Device Information Service with OTAU Example (bas_otau_app)</td>
</tr>
<tr>
<td></td>
<td>25. L2CAP Peripheral Example (l2cap_peripheral)</td>
</tr>
<tr>
<td></td>
<td>26. L2CAP Central Example (l2cap_central)</td>
</tr>
<tr>
<td></td>
<td>27. Eddystone Beacon Example (eddystone_beacon)</td>
</tr>
<tr>
<td></td>
<td>28. Alt-Beacon Example (altbeacon)</td>
</tr>
<tr>
<td>ble_profiles</td>
<td>Contains the following BLE Profiles:</td>
</tr>
<tr>
<td>Folder Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1. Alert Notification</td>
<td></td>
</tr>
<tr>
<td>2. Apple Alert Notification Service (ANCS)</td>
<td></td>
</tr>
<tr>
<td>3. Proximity Monitor</td>
<td></td>
</tr>
<tr>
<td>4. Proximity Reporter</td>
<td></td>
</tr>
<tr>
<td>5. Time Information</td>
<td></td>
</tr>
<tr>
<td>6. HID over GATT profile (HOGP)</td>
<td></td>
</tr>
<tr>
<td>7. Phone Alert Status</td>
<td></td>
</tr>
<tr>
<td>8. Blood Pressure</td>
<td></td>
</tr>
<tr>
<td>9. Heart Rate</td>
<td></td>
</tr>
<tr>
<td>10. Find me</td>
<td></td>
</tr>
<tr>
<td>11. Custom Serial Chat</td>
<td></td>
</tr>
<tr>
<td>12. OTAU Profile (Target)</td>
<td></td>
</tr>
</tbody>
</table>

Contains the following BLE Services:

1. Alert Notification Service
2. BLE Manager-Common BLE Event Frame Work
3. Apple Alert Notification Service (ANCS)
4. Battery Service (BAS)
5. Current Time Service (CTS)
6. Device Information Service (DIS)
7. Immediate Alert Service (IAS)
8. Link Loss Service (LLS)
9. Scan Parameter Service (SPS)
10. Tx Power Service (TXPS)
11. HID Device (HOGP Service)
12. Blood Pressure
13. Custom Serial Chat Service
14. Heart Rate Service
15. Next DST Service
16. Phone Alert Service
17. Reference Time Service
18. OTAU Service (Target)

Include
Top level includes for the delivered libraries and Platform API's.

Lib
Adaptor Library.

services
Application level services and Platform level - console, serial driver, timer.

Src (Atmel Studio Only)
Platform abstraction implementation for BLE Adapter Library.

utils
Utility functions used by BluSDK Smart components (for example: DBG_LOG, DBG_LOG_DEV etc.).
<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patch (Keil Only)</td>
<td>BLE Stack Firmware.</td>
</tr>
<tr>
<td>tools (Keil Only)</td>
<td>Flash Download Tools.</td>
</tr>
</tbody>
</table>
5. **BluSDK Smart Release Features**

BluSDK Smart v6.2 release supports the following profiles and services.

1. **Profiles**
   - Alert Notification
   - Apple Alert Notification Center Service (ANCS)
   - Blood Pressure
   - Custom Serial Chat
   - Find Me Target
   - Heart Rate
   - HOGP (HID Over GATT Profile)
   - Health Thermometer
   - Phone Alert Status
   - Scan Parameters
   - Time Information
   - Proximity
   - OTAU Profile (Target)

2. **Services Support**
   - Alert Notification Service
   - BLE Manager-Common BLE Event framework
   - Apple Alert Notification Service (ANCS)
   - Battery Service (BAS)
   - Current Time Service (CTS)
   - Device Information Service (DIS)
   - Immediate Alert Service (IAS)
   - Link Loss Service (LLS)
   - Scan Parameter Service (SPS)
   - Tx Power Service (TXPS)
   - HID Device (HOGP Service)
   - Blood Pressure
   - Custom Serial Chat Service
   - Heart Rate Service
   - Next DST Service
   - Phone Alert Service
   - Reference Time Service
   - OTAU Service (Target)
   - Health Thermometer Service

3. **Mobile Applications**
   - Microchip Smart Connect Mobile Application for Android
     - Health Thermometer
     - Proximity Monitor
     - Blood Pressure
Example applications are provided for the profile/services mentioned above and are available for ATSAMB11 MCU platform.
6. **Release Revision History**

6.1 **BluSDK Smart v6.2**

6.1.1 **Major New Features of BluSDK Smart Software**

1. Added generic PDS feature into BluSDK Smart v6.2.
2. Added PDS service support to BLE Manager to store, retrieve, list, delete, search the Devices bonding information in NVM. The bonding information will persist on the memory on various scenarios such as, Power-On-Reset, Watchdog Reset, device reset, etc.
3. Added PDS support into Battery Information example application.

6.1.2 **Major Improvement and Bug Fixes**

1. Added compile time option to configure the BLE_MAX_DEVICE_CONNECTION range (1 to 8) from Pre-processor symbol, when BLE_MAX_DEVICE_CONNECTION defined it overrides the default definition (ble_manager.h) BLE_MAX_DEVICE_CONNECTION value (8).
2. Added compile time option to disable the console logs by defining DEBUG_LOG_DISABLED Pre-processor symbol.
3. Added TDM mode ADC conversion example into the ATSAMB11 ADC quick start.
4. Fixed the ATSAMB11 TDM mode ADC conversion results are stored in next channel ADC output register. The workaround has been added for TDM mode ADC conversion output, where the ADC output values will be retrieved properly.
7. Fixed broken Doxygen link in BluSDK Smart.

6.2 **BluSDK Smart v6.1**

6.2.1 **Major New Features of BluSDK Smart Software**

1. SAMB11ZR/XR module support.
2. Eddy-stone beacon and alt-beacon.
3. L2CAP throughput Central/Peripheral applications are added.
4. Security cores – Added support to AES and SHA security cores.
5. AON GPIOs Pull up – Enable pull up on AON_GPIO_1/2.
6. Platform – Platform code cleanup (for Atmel Studio project only).
7. Tools – OTA image creator and Flash image is updated.

6.2.2 **Major Improvement and Bug fixes**

1. Supports the update of On-the-fly advertisement data.
2. BLE Manager event callback structure added for GATT, GAP, L2CAP and Custom events.
3. Added custom and platform events for BluSDK Smart examples.
4. Fixed value length of Indication/Notification is from uint8_t to uint16_t.
5. Fixed recovery issue observed when device is disconnected during and ‘in and out’ of range situation.


7. FW version: Fixed broken FW version API.

8. Added new API to get RF settings version. Current version is 1.0.

9. Fixed connection parameter update issue observed in some iOS and Android devices.

10. Re-initialization of variables not needed anymore (Issue fix for Keil projects only).

11. Modified Multi-Role Multi-Connect application to support for 8 links (1 master and 7 slave roles for DUT). Currently, number of links recommended is 4.

12. Prolonged IRQ handler execution will lead to unexpected reset with message BOUT. For more details, refer to 11.4 Recommended Code Implementation for Handling Platform Event.

6.3 BluSDK Smart v5.2

6.3.1 Major New Features of BluSDK Smart Software

1. Add BLE basic OTAU (Over The Air Upgrade) support by using Battery and Device Information Service. The following are the top level features of OTAU:
   - Fail-Safe upgrade mechanism
   - Image integrity check using 32-bit CRC
   - Updated process can resume back from left over during power interruptions and any other interrupts
   - App profile and OTAU profile can run in parallel.
   - Pause/Resume of the upgrade supported at both OTAU Manager and OTAU Target.
   - Forced upgrade – OTAU Manager can initiate to download same version of software as and when required. Pause and Resume is not allowed during this operation
   - Easy to include OTAU profile in any BLE application

2. GATT level user configurable authentication.

3. Mobile Application – Enhanced Microchip Smart Connect (Version 3.0) for both Android and iOS to support OTAU.


5. Tools – support for creation of factory and OTAU image by using ‘OTAUImageCreator’ tool.

6.3.2 Major Improvements and Bug Fixes

Startup template update to provide simple template.

6.4 BluSDK Smart v5.1

6.4.1 Major New Features of BluSDK Smart

1. Add Direct Test Mode and AT CMD samples.

2. ULP examples (HID Keyboard, Beacon and Battery Service examples) with AON_GPIO and AON_Timer Support.

3. Atmel Adapter Library & FW_Patch (v 2.9.1)
   - AON (Always ON) timer support.
   - AON GPIO support.
– GPIOs latch to maintain GPIO status in ULP mode.
– Fixed WDT ISR handler.
– SambFlashTool updated.
– Updated the download .py script to remove the dependency on the hex2bin.exe.

4. Platform
– Add support for AON_GPIO_0, AON_GPIO_1 and AON_GPIO_2 interrupt callback.
– Add two new APIs to register and unregister callbacks for any of the three wakeup pins.
  • wakeup_int_register_callback
  • wakeup_int_unregister_callback
  • platform_chip_reset API updated.

5. Tools - Support for Atmel Wireless Composer/Performance Analyzer using Direct Test Mode example.

6.4.2 Major Improvements and Bug Fixes
1. Fix BLECORE-795 issue (WDT reset is not behalf as expected).
2. Fix BLECORE-863 issue (Power optimization settings as recommended by systems team).
3. Fix BLECORE-865 issue (Support for AON_GPIO callback).
4. Fix BLECORE-874 issue (Save and restore of pull enables).
5. Fix BLECORE-876 issue (Disabling the output enables for GPIO 14 and GPIO 15).
6. Fix BLUSMART-73 issue (Disconnect test with HTPT).
7. Fix BLUSMART-74 issue (Reconnect test with HTPT).
8. Fix BLUSMART-77 issue (System blocked with “send_plf_int_msg_ind” API).
9. Fix BLUSMART-78 issue (Advertising is stopped after first interrupt of AON timer).
10. Fix BLUSMART-88 issue (2 MHz GPIO clock generating).
11. Fix BLUSMART-106 issue (AON timer stops suddenly after 10 to 40 minutes).
12. Fix BLUSMART-109 issue (The timeout of at_ble_event_get does not work).
13. Fix BLUSMART-110 issue (The platform_chip_reset api does not work.).
14. Fix BLUSMART-121 issue (AON timer working after adding debug message).

6.5 BluSDK Smart v5.0

6.5.1 Major New Features of Blu SDK Smart
1. Support Timeout functionality for at_ble_event_get API.
2. API Updates
   – at_ble_event_get() can support timeout value.
   – At_ble_chip_reset() API added.

6.5.2 Major Improvement and Bug Fixes
1. Fix vbat and vtemp calibration update issue.
2. Fix channel 39 advertising issue.
3. Fix disconnection with error 0x08.
4. Fix disconnect with error 0x3E.
5. Fix instability with some boards.
6. Fix system reset issue.
7. Fix timeout issue of at_ble_event_get.

6.6 BluSDK Smart v4.1

6.6.1 Major New BLUSDk Smart Features
1. Support for Features like Multi-Role/Multi-Connect to maintain both central and peripheral role simultaneously and multiple connections for various applications.
2. Mobile Application – Enhanced Microchip SmartConnect (version 2.0) tablet application for iOS and Android.
3. Atmel Adapter Library and Firmware (v2.5):
   – Optimized ULP (Ultra Low Power) mode.
   – Vbat/Temp calibration support.
   – TX-FM calibration on cold-boot.
4. Added three new APIs:
   – at_ble_calib_config().
   – at_ble_calib_get_temp().
   – at_ble_calib_get_voltage().
5. Fix missing Advertising packet on Channel 37.

6.7 BluSDK Smart v4.0

6.7.1 Major New Features
1. Support for profiles like, Blood Pressure, Heart Rate Monitor, Phone Alert Status, Time Information, Custom Serial Chat, Find Me, and Alert Notification on ATSAMB11 Platform.

6.8 BluSDK Smart v3.0

6.8.1 Major New Features
1. Updated Profile and Example application.
   – Device information sample application.
   – Battery Service sample application.
   – Tx power Service sample application.
   – Device information sample application.
   – Battery Service sample application.
   – Scan Parameter Service sample application.
   – Proximity Monitor sample application.
   – Proximity Reporter sample application.
   – Simple Broadcaster sample application.
– Simple Observer sample application.

2. Included Test applications.
   – Atmel Smart BLU.
   – Atmel Beacon Rader.
   – Health Thermometer.


4. ATSAMB11 Carrier Board and ATSAMB11 Xplained (A5) Board support.

6.9 BluSDK Smart v2.1

6.9.1 Major New Features

1. Updated profile and example application:
   – HID keyboard.
   – HID Mouse.

2. Included Test applications:
   – Atmel Smart BLU.
   – Atmel Beacon Rader.
   – Health Thermometer.


4. SAMB11 Carrier Board and SAMB11 Xplained (A5) Board support.

6.10 BluSDK Smart v2.0

6.10.1 Major New Features

1. Profile and Example application:
   – ANCS.
   – TIP.
   – HTPT.
   – iBeacon.

2. Included Test applications:
   – Atmel Smart BLU.
   – Atmel Beacon Rader.
   – Health Thermometer.


4. ATSAMB11 Carrier Board support.
## Known Issues

The following table provides the details on BluSDK Smart software.

**Table 7-1. BluSDK Smart Software**

<table>
<thead>
<tr>
<th>JIRA ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUSMART-60</td>
<td>Link layer connection failed to establish connection and returns 3E error.</td>
</tr>
<tr>
<td>BLUSMART-66</td>
<td>Unregistered interrupt can be added to registered interrupts list and vice-versa.</td>
</tr>
<tr>
<td>BLUSMART-80</td>
<td>iOS Application / TIP / Go back from Bluetooth menu.</td>
</tr>
<tr>
<td>BLUSMART-106</td>
<td>AON timer stops suddenly after 10 to 40 minutes.</td>
</tr>
<tr>
<td>BLUSMART-107</td>
<td>ATSAMB11 halt while testing scan.</td>
</tr>
<tr>
<td>BLUSMART-118</td>
<td>Source codes for 'Image creator' is different with Tools team stash.</td>
</tr>
<tr>
<td>BLUSMART-128</td>
<td>RX/TX current with Pre-release DTM tool measured is much different with datasheet.</td>
</tr>
<tr>
<td>BLUSMART-151</td>
<td>[OTAU Keil] ~16% higher power consumption during OTAU.</td>
</tr>
<tr>
<td>BLUSMART-152</td>
<td>[OTAU Keil] Decouple AON timer initialization from timer_init().</td>
</tr>
<tr>
<td>BLUSMART-189</td>
<td>SAMB11/MR – IOP issue with Huawei phone: Not observing any data after pairing is done and with HID mouse profile there is NO mouse pointer movement on the phone side.</td>
</tr>
<tr>
<td>BLUSMART-195</td>
<td>SAMB11 MR/ZR – Inspite of app screen lock write successfull message is seen.</td>
</tr>
<tr>
<td>BLUSMART-197</td>
<td>SAMB11 ZR Module – In Phone alert status, device re-connection is not happening in out of range scenario.</td>
</tr>
<tr>
<td>BLUSMART-203</td>
<td>SAMB11-MR / ZR Module – Connection issue observed in HTP profile.</td>
</tr>
<tr>
<td>BLUSMART-206</td>
<td>SAMB11 MR/ZR – When the cache is not cleared and when app generic profile data is clicked, iOS Atmel App crashes.</td>
</tr>
<tr>
<td>BLUSMART-212</td>
<td>SAMB11 MR/ZR – Multirole_multiconnect profile is not working as expected.</td>
</tr>
<tr>
<td>BLUSMART-214</td>
<td>SAMB11/GCC/MR – Scan Parameter Profile: &quot;Device disconnected Reason:0x3d Handle=0x0 &quot; issue observed once during testing.</td>
</tr>
</tbody>
</table>
### Known Issues

<table>
<thead>
<tr>
<th>JIRA ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUSMART-220</td>
<td>SAMB11 MR and ZR – In ALT Beacon profile, webpage not opening when the mobile device comes in close proximity to a given beacon.</td>
</tr>
<tr>
<td>BLUSDK-1078</td>
<td>Pairing information's are not stored in the NVM. During power-cycle host MCU lost its peer device bonding information, since there is no PDS implementation on Host side.</td>
</tr>
<tr>
<td>BLECORE-945</td>
<td>Application should be able to control confirmation for Indication.</td>
</tr>
<tr>
<td>BLUSDK-1071</td>
<td>Connection parameter update does not work properly</td>
</tr>
<tr>
<td>BLUSMART-144</td>
<td>DTM API – SAMB11 does not work.</td>
</tr>
<tr>
<td>WLNMC2000-38</td>
<td>SAMB11 platform stack is not aligned by 8.</td>
</tr>
<tr>
<td>WSGA-2082</td>
<td>SAMB11 storing user data in RFU 8 KB (0x3E000 to 0x3FFFF) region. The first 4 KB from address 0x3E000 to 0x3EFFF is used by OTA. Therefore, this 4 KB cannot be used for user data storage. The address from 0x3F000 to 0x3FFFF is available for user data storage.</td>
</tr>
</tbody>
</table>

The following table provides the details on mobile applications.

**Table 7-2. Mobile Applications**

<table>
<thead>
<tr>
<th>JIRA ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEMAPPSS-6</td>
<td>RSSI value is fluctuating more for Android Proximity Reporter Application.</td>
</tr>
<tr>
<td>BLEMAPPSS-9</td>
<td>When Bluetooth is turned off during scan, the scan progress bar shows that scanning is in progress.</td>
</tr>
</tbody>
</table>
8. Limitations

This section describes about BluSDK Smart Software and Microchip Smart Connect Mobile Applications.

8.1 BluSDK Smart Software

1. Adapter Layer and Platform:
   1.1. ULP (Ultra Low Power) – the use of the ULP has some restrictions
       • Cannot use the ULP mode with any application that uses timers except AON timer.
       • After wake up from ULP mode, ATSAMB11 peripherals must be reinitialized (for example— UART, SPI).
   1.2. Single Thread mode only.
   1.3. No OS features available for User app.

2. Profiles:
   2.1. Common
       • ULP is enabled by default in the iBeacon and Battery example application.
       • ULP is enabled optionally by using NENABLE_ULP or ENABLE_ULP symbol in the HID keyboard example application.
       • For other applications it is disabled explicitly. The rationale to disable ULP is because most of the examples rely on user input via user button (SW0 button) on the ATSAMB11 Xplained Pro, serial console that are not mapped as wake-up source on the development kit. This is applicable to ATSAMB11 MR only.
       Note: For ATSAMB11 ZR SW0 button is mapped to AON_GPIO_0 which can be used as a wake up source.
       • ULP can be enabled by connecting an external wake up source. The AO_GPIO_0 pin is mapped to the EXT1 header (pin 9) for ATSAMB11-MR XPRO. For more details, refer to the ATSAMB11 Xplained Pro User Guide and ATSAMB11 BluSDK SMART Interrupts and ULP Architecture and Usage User’s Guide.
       • AON timer is applied to only Battery Info example application.
   2.2. HOGP – HID
       • HID keyboard device example application and HID mouse device example application are compatible only with Android based mobile devices (Android 4.4/KitKat and higher versions).
       • HID keyboard device example application supports only “Just Works” pairing.
   2.3. Multi-Role/Multi-Connect
       • Both Scan and Advertisement cannot be performed simultaneously.
   2.4. PTS Compliance:
       • PTS compliance has not yet been achieved for the profile examples.
       • ANCS and TIP are Apple iOS specific implementations.
   2.5. OTA Upgrade
       • Image security is not supported currently.
       • Individual section upgrade is not tested.
   2.6. PDS
       • BLE Manager in BluSDK supports to store only 8-devices bonding information in to PDS. But for items it is not limited for storing other items in PDS. The PDS example
implementation available for Battery Info Application but it is not limited and can be added to other Example Applications.

3. IDE
   3.1. IAR IDE is not supported.

8.2 Mobile Applications

1. Microchip SmartConnect Android Application
   1.1. Common
       • Android Framework limitations:
           – When wrong passkey is entered, still pairing succeeds.
   1.2. Phone Alert: Display state is not supported.
   1.3. Custom Serial Chat:
       • Android-4.4.4 (KitKat) – Can support only upto 20 characters.
       • Android-5.x and 6.x– (Lollipop and Marshmallow) – Can support less than 150 characters. If more than 150 characters are entered, then complete data is not sent.
       • iOS – Can support upto 150 characters.
   1.4. When the device is moved, the beacon icon updates are as not fast as the movement pace. However there is an improvement in beacon icon update, compared to older versions of Android app.
9. **Additional Information/Pre-requisites**

1. **Pre-requisites to Test ULP mode**
   
   1.1. AON_GPIOs have to be connected to GND for ULP (Ultra Low Power) mode for full function. Leaving AON_GPIOs floating can produce undefined behavior.
   
   - **AON_GPIO_0**:
     - For ATSAMB11 MR— Pin 9 of EXT1 must be connected with GND
     - For ATSAMB11 ZR— Pin 9 of EXT1 must be floating
   
   - **AON_GPIO_1** — Pin 9 of EXT3 must be floating (do not connect to anything)
   
   - **AON_GPIO_2** — Pin 6 of EXT1 must be floating (do not connect to anything)

   ![ATSAMB11 MR-XPRO Board Settings](image)

2. To test ANCS and Time Profile, an iPhone supporting iOS version above 7.0 is required.

3. **General information with reference to BLE support on mobile devices are as follows:**

   3.1. **iOS:**
   
   - iPhone 4S and above supports the Bluetooth 4.0, and higher. It is recommended to use a device with iOS version 8.0 or above for best performance and stability.

   3.2. **Android:**
   
   - Preliminary support for Bluetooth 4.0 is added in Android Jellybean (4.2 onwards).
   - For best performance and stability, it is recommended to use a device with Android version above 5.x and 6.x (that is, preferably Marshmallow or Lollipop).
Due to a wide-spectrum of Android devices with many variants (customized Android devices, various hardware/connectivity chipset configurations), the performance of BLE Application are not the same on all devices. Android phone that includes a Bluetooth chipset supporting version 4.0 or higher is a pre-requisite.

Due to evolving nature of BLE APIs in Android, the performance/stability variations are observed, when testing with different Android release flavors (Jellybean/4.3, KitKat/4.4 and Lollipop/5.1).

HID over GATT profile was introduced in Android from version 4.4 (KitKat). A mobile device supporting Android 4.4 and higher is a pre-requisite for running the HID keyboard device and HID mouse device example applications.

Disconnection triggered by the application running in an Android takes a long time (12 to 15 seconds) to complete.

When the peripheral device is reset, then the manual unpair is required in Android phone to reconnect.

9.1 How to Install ATSAMB11 Device Pack using Device Pack Manager

1. Go to Tools > Device Pack Manager.

   Figure 9-2. Atmel Studio Start Page

2. Click Check for Updates, select new pack of ATSAMB11, and then “Install selected packs”.
3. Click **Install**.

4. Click **Close** and restart Atmel Studio.
9.2 How to Install SAMB11 Device Pack with atpack File

1. Go to Tools > Device Pack Manager.

2. In Device Pack Manager window, go to Install > Browse pack file.
3. Select appropriate .atpack file and click Open.

4. Click Install.
9.3 Measurement of Power Consumption

The following are the board settings of the ATSAMB11 MR and ATSAMB11 ZR.

1. The board settings of the ATSAMB11 MR are explained in the following steps.
   1.1. The first jumper on the right (below SW0 push button) must be on “By-Pass” position – (1).
   1.2. The next one must be on “Measure” position – (2).
   1.3. AON_GPIO_0 – Pin 9 of EXT1 must be connected with GND – (3).
       **Note:** In case of ATSAMB11 MR, aon_gpio0_config_default function defined in platform_drv.c file and called in platform_driver_init function must be commented out to allow ATSAMB11 to go to ULP mode. The aon_gpio0_config_default function must be invoked when AON_GPIO_0 – Pin 9 of EXT1 is in floating state, to avoid BOUT! issue in some scenarios.
   1.4. AON_GPIO_1 – Pin 9 of EXT3 must be floating (do not connect to anything).
   1.5. AON_GPIO_2 – Pin 6 of EXT1 must be floating (do not connect to anything).
2. The board settings of the ATSAMB11 ZR are explained in the following steps:
   2.1. The first jumper on the right (below SW0 push button) must be on “By-Pass” position – (1).
   2.2. The next one must be on “Measure” position – (2).
   2.3. AON_GPIO_0 – Pin 9 of EXT1 must be floating (do not connect to anything) – (3).
   \textbf{Note:} In ATSAMB11 ZR, SW0 is connected AON GPIO0 and the same pin functionality shared with EXT1-PIN 9. It is necessary to leave EXT1-PIN 9 unconnected to allow the device to enter Sleep mode at any time.
   2.4. AON_GPIO_1 – Pin 9 of EXT3 must be floating (do not connect to anything) – (4).
   2.5. AON_GPIO_2 – Pin 6 of EXT1 must be floating (do not connect to anything) – (5).
   2.6. Create iBeacon sample of ATSAMB11.
3. Build the project and run.
4. Go to \texttt{Tools > Data Visualizer}. 
5. Click **Connect** – (1).
7. Click **Start** – (3).

Figure 9-12. DGI Control Panel Window

8. The “Power Analysis” window is displayed – (4).
Figure 9-13. Power Analysis Window
10. Hardware Resources used by BluSDK-SMART

10.1 Timer

The following table provides the details of the timer used by BluSDK Smart.

<table>
<thead>
<tr>
<th>Board</th>
<th>ARM Timer</th>
<th>Dual Timer (Timer 1)</th>
<th>AON Sleep Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMB11-ZR</td>
<td>BLE Stack</td>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>SAMB11-MR</td>
<td>BLE Stack</td>
<td>Application</td>
<td>Application</td>
</tr>
</tbody>
</table>
11. Appendix 1 – BluSDK SMART Change Log

This section provide details of the new features introduced in the API and a change comparison with the previous version of BluSDK-SMART API.

11.1 BluSDK Smart Module Changes

The following table provides the details on “New features in the APIs”.

<p>| Table 11-1. New Features in the API - New Signature versus Modified Signature |
|-----------------------------|-----------------------------|-------------------------------|-------------------------------|</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>SL.No</th>
<th>Description</th>
<th>Old Signature</th>
<th>New Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE Manager</td>
<td>1</td>
<td>Enum representing device link layer state</td>
<td>blePeripheral_state</td>
<td>ble_device_ll_state</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Handler for Profile/Service/application to register the BLE events callbacks</td>
<td>bool ble_mgr_events_callback_handler(ble_mgr_event_cb_t event_cb_type, ble_mgr_event_t event_type, const ble_event_callback_t *ble_event_handler);</td>
<td>bool ble_mgr_events_callback_handler(ble_mgr_event_cb_t event_cb_type, ble_mgr_event_t event_type, const void *ble_event_handler);</td>
</tr>
<tr>
<td>Services\ UART (GCC)</td>
<td>3</td>
<td>Handler for getting character from console</td>
<td>int getchar_b11Timeout(unsigned int sec);</td>
<td>int getchar_timeout(unsigned int sec);</td>
</tr>
</tbody>
</table>

The following table provide the details on newly added Type (Enum and Structure) and API.

<p>| Table 11-2. Data Structures |
|-----------------------------|-----------------------------|-------------------------------|-------------------------------|</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>API</th>
<th>API Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE Manager</td>
<td>Client Characteristic Configuration Descriptor States</td>
<td>Enum</td>
<td>ble_gatt_cccd_status_t</td>
</tr>
<tr>
<td>BLE GAP Event Callback</td>
<td>Structure</td>
<td></td>
<td>ble_gap_event_cb_t</td>
</tr>
<tr>
<td>BLE GATT Client Event Callback</td>
<td>Structure</td>
<td></td>
<td>ble_gatt_client_event_cb_t</td>
</tr>
<tr>
<td>BLE GATT Server Event Callback</td>
<td>Structure</td>
<td></td>
<td>ble_gatt_server_event_cb_t</td>
</tr>
</tbody>
</table>
Table 11-3. New Features in the API

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>API</th>
<th>API Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE L2CAP Event Callback</td>
<td>Structure</td>
<td>ble_l2cap_event_cb_t</td>
<td></td>
</tr>
<tr>
<td>BLE HTPT Event Callback</td>
<td>Structure</td>
<td>ble_htpt_event_cb_t</td>
<td></td>
</tr>
<tr>
<td>BLE DTM Event Callback</td>
<td>Structure</td>
<td>ble_dtm_event_cb_t</td>
<td></td>
</tr>
<tr>
<td>BLE Custom Event Callback</td>
<td>Structure</td>
<td>ble_custom_event_cb_t</td>
<td></td>
</tr>
<tr>
<td>User Custom Event</td>
<td>Structure</td>
<td>user_custom_event_t</td>
<td></td>
</tr>
<tr>
<td>Platform ISR Event</td>
<td>Structure</td>
<td>platform_isr_event_t</td>
<td></td>
</tr>
<tr>
<td>Handler for checking the disconnected device role is peripheral or not</td>
<td>API</td>
<td>ble_check_disconnected_isPeripheral</td>
<td></td>
</tr>
<tr>
<td>Handler for getting the BluSDK SMART version</td>
<td>API</td>
<td>ble_sdk_version</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>Samb11 module version</td>
<td>Enum</td>
<td>samb11_module_version_tag</td>
</tr>
<tr>
<td>Initialize the samb11 module type</td>
<td>API</td>
<td>platform_set_module_type</td>
<td></td>
</tr>
<tr>
<td>Retrieves FW version</td>
<td>API</td>
<td>platform_get_fw_version</td>
<td></td>
</tr>
<tr>
<td>Retrieves RF version</td>
<td>API</td>
<td>platform_get_rf_version</td>
<td></td>
</tr>
<tr>
<td>Services \ dualtimer (GCC)</td>
<td>start the timer with time units in ms</td>
<td>API</td>
<td>hw_timer_start_ms</td>
</tr>
<tr>
<td>Services \ gpio \ button.h (GCC)</td>
<td>Get the button status</td>
<td>API</td>
<td>button_0_input_level</td>
</tr>
</tbody>
</table>

11.2 BLE Library API – New Features

This section provides details of the new features introduced in the API.

Table 11-3. New Features in the API

<table>
<thead>
<tr>
<th>Category</th>
<th>New features</th>
<th>Description</th>
<th>API/Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eddystone beacon</td>
<td>Add the following APIs to support Eddystone beacon:</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
### Category: New features

- `at_ble_adv_data_update_on_the_fly`
- `at_ble_get_adv_cnt`
- `at_ble_rst_adv_cnt`
- `at_ble_en_reg_adv_cmp_event`
- `AT_BLE_ADV_CMP_EVENT`

### Category: Security Cores

Added support to AES and SHA security cores using following API.

- `AES128_CTR_Enc`
- `AES128_CTR_Dec`
- `AES128_CBC_MAC`
- `AES128_CCM_Dec`
- `AES128_CCM_Enc`
- `AES128_CBC_Enc`
- `AES128_CBC_Dec`
- `AES128_encrypt`
- `SHA256_Hash`

### 11.3 Difference between BLE APIs – BluSDK Smart 6.2 versus BluSDK Smart 5.2

The table provides the difference between the old and the new signatures.

**Table 11-4. Difference between Old Signature and New Signature**

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>Old Signature</th>
<th>New Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP</td>
<td>Advertisement</td>
<td><code>at_ble_set_adv_channel_Map</code></td>
<td><code>at_ble_adv_channel_map_set</code></td>
</tr>
<tr>
<td>GAP</td>
<td>Channel map setting API</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2018 Microchip Technology Inc.
### Table 11-5. Signature Difference

<table>
<thead>
<tr>
<th>Category</th>
<th>Features</th>
<th>Type / Status</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Initialization</td>
<td>Init Configuration</td>
<td>Structure / Modified</td>
<td>at_ble_init_config_t</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Getting RF Version</td>
<td>API / New</td>
<td>at_ble_rf_version_get</td>
</tr>
<tr>
<td>GAP</td>
<td>Advertisement Channel Map</td>
<td>Enum / Modified</td>
<td>at_ble_adv_channel_map_t</td>
</tr>
<tr>
<td></td>
<td>Advertising Complete Event</td>
<td>Structure / New</td>
<td>at_ble_adv_cmp_evt_t</td>
</tr>
<tr>
<td></td>
<td>Disconnection Event</td>
<td>Structure / Modified</td>
<td>at_ble_disconnecte_d_t</td>
</tr>
<tr>
<td></td>
<td>Advertisement Update on the fly</td>
<td>Structure / New</td>
<td>at_ble_adv_data_update_on_the_fly_t</td>
</tr>
<tr>
<td></td>
<td>Getting advertisement Channel</td>
<td>API / New</td>
<td>at_ble_adv_channel_map_get</td>
</tr>
<tr>
<td>GATT</td>
<td>Descriptor Attribute information</td>
<td>Structure / Modified</td>
<td>at_ble_generic_att_desc_t</td>
</tr>
<tr>
<td></td>
<td>Attribute Value</td>
<td>Structure / Modified</td>
<td>at_ble_char_val_t</td>
</tr>
<tr>
<td></td>
<td>Notification Event Received</td>
<td>Structure / Modified</td>
<td>at_ble_notification_received_t</td>
</tr>
<tr>
<td></td>
<td>Indication Event Received</td>
<td>Structure / Modified</td>
<td>at_ble_indication_recieved_t</td>
</tr>
</tbody>
</table>

### 11.4 Recommended Code Implementation for Handling Platform Event

**Issue:** Prolonged IRQ handler execution can change the execution behavior.

In the Atmel Studio implementation, the user defined interrupt callback is executed right after the interrupt handler and in the same interrupt context. Any processing (based on time length of processing) inside the user callback leads to a prolonged IRQ handler execution that can change the execution behavior.

**Note:** The bold highlighted code indicates the codes to be added. Solution below describes how to avoid the prolonged IRQ handler issue for Dual timer of ATSAMB11 as an example.

**Solution**

1. Any time taking interrupt processing sequence must be executed in user context to avoid prolonged execution of IRQ handler in interrupt context. In order to achieve this User defined Interrupt callback should be modified as follows:

```c
static void timeevent_cb(void)
{
    // Remove the Platform event posting code as this results in prolonged IRQ handler execution
    if(((int*wakeup_event_pending) >> 24) & 0xFF) == 0)
    {
        *wakeup_event_pending &= ~(0xFF << 24); // Currently Bit 25 to 31 are not used.
        So clearing them. If using any of the bits then the corresponding bit should not be cleared
```
*wakeup_event_pending |= (0x1 << 24); // Setting the bit 24 indicates that further processing associated with this interrupt (which generated the user callback) needs to be done in user context
}

Note:
timeevent cb is the interrupt callback registered for the timer interrupt. The following code shall be registered during application initialization.
dualtimer_register_callback(DUALTIMER_TIMER1, timeevent_cb)

Bit numbers 0, 8 and 16 of wakeup_event_pending flag is reserved for AON_GPIO_0, AON_GPIO_1, AON_GPIO_2. In the above case we use bit number 24.
The pointer wakeup_event_pending is initialized in platform_driver_init (platform_drv.c)

2. Add a new enum value (TIMER_WAKEUP_SOURCE) for the interrupt source which has to be handled in user context.
enum port_wakeup_source is defined in platform.h.

enum port_wakeup_source {
    /** External Wakeup source AON_GPIO_0 */
    PORT_WAKEUP_SOURCE_AON_GPIO_0 = 0,
    /** External Wakeup source AON_GPIO_1 */
    PORT_WAKEUP_SOURCE_AON_GPIO_1 = 1,
    /** External Wakeup source AON_GPIO_2 */
    PORT_WAKEUP_SOURCE_AON_GPIO_2 = 2,
    /** Wakeup source Timer */
    TIMER_WAKEUP_SOURCE,
    PORT_WAKEUP_SOURCE_MAX_VAL
};

3. wakeup_active_event_callback: This is the common callback routine that is called by the firmware from user context when wakeup_event_pending flag is set. Check for the specific interrupt that triggered this callback. Add the following code for timer.
void wakeup_active_event_callback(uint32_t wakeup_source)
{
    portint_callback_t callback;
    uint32_t *pu32WakeSource = (uint32_t *)wakeup_source;
    if(((pu32WakeSource & 0xFF) == 1)
    {
        if(wakeup_source_callback[0] != NULL) {
            callback = wakeup_source_callback[0];
            callback();
        }
        *pu32WakeSource &= ~(0xFF);
    }
    if(((pu32WakeSource >> 8) & 0xFF) == 1)
    {
        if(wakeup_source_callback[1] != NULL) {
            callback = wakeup_source_callback[1];
            callback();
        }
        *pu32WakeSource &= ~(0xFF << 8);
    }
    if(((pu32WakeSource >> 16) & 0xFF) == 1)
    {
        if(wakeup_source_callback[2] != NULL) {
            callback = wakeup_source_callback[2];
            callback();
        }
        *pu32WakeSource &= ~(0xFF << 16);
    }
    if(((pu32WakeSource >> 24) & 0xFF) == 1)
    {
        if(wakeup_source_callback[3] != NULL) {
Note:

wakeup_active_event_callback has been already registered as follows in platform_driver_init (platform_drv.c):

*pu32WakeSource &= ~(0xFF << 24);
}

If the interrupt source is timer (TIMER_WAKEUP_SOURCE) then the callback registered using wakeup_int_register_callback is called from wakeup_active_event_callback.

4. In function wakeup_int_register_callback (platform_drv.c): Add a condition for timer for registering the timer interrupt specific callback.

enum port_status_code wakeup_int_register_callback(enum port_wakeup_source wakeup_source, portint_callback_t fp)
{
    enum port_status_code status = PORT_STATUS_OK;
    if((fp != NULL) && ((wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_0) ||
     (wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_1) ||
     (wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_2) ||
     (wakeup_source == TIMER_WAKEUP_SOURCE /*Used for Dual timer Callback*/))) {
        wakeup_source_callback[wakeup_source] = fp;
    } else {
        status = PORT_STATUS_ERR_INVALID_ARG;
    }
    return status;
}

5. Register the callback to execute the user context processing of the timer interrupt as follows:

wakeup_int_register_callback(TIMER_WAKEUP_SOURCE, timekeeping_cb);

This should be done during application initialization.

6. Further processing associated with the timer interrupt shall be done in a callback executed in the User context. This callback should be defined as follows:

static void timekeeping_cb(void)
{
    send_plf_int_msg_ind(RAM_ISR_TABLE_TIMER0_INDEX, TIMER_EXPIRED_CALLBACK_TYPE_DETECT, NULL, 0);
}

7. In function wakeup_int_unregister_callback (platform_drv.c): Add a condition for timer for unregistering the callback.

enum port_status_code wakeup_int_unregister_callback(enum port_wakeup_source wakeup_source)
{
    enum port_status_code status = PORT_STATUS_OK;
    if((wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_0) ||
     (wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_1) ||
     (wakeup_source == PORT_WAKEUP_SOURCE_AON_GPIO_2) ||
     (wakeup_source == TIMER_WAKEUP_SOURCE)) {
        wakeup_source_callback[wakeup_source] = 0;
    } else {
        status = PORT_STATUS_ERR_INVALID_ARG;
    }
}
8. In `platform_driver_init (platform_drv.c)` unregister the callback for timer interrupt as follows to ensure callback is not a uninitialized value:

```c
wakeup_int_unregister_callback(TIMER_WAKEUP_SOURCE);
```

**Issue:** Firmware usually waits for BLE event to occur before providing control to user application. In this case user application cannot execute functionality. If we have a timer interrupt every 100 msec and BLE events are less frequent like every 1 sec, then the user context execution of interrupt (`wakeup_active_event_callback` described above) will be called every 1 sec instead of 100 ms.

**Solution:** Post the semaphore `gstrFwSem` that is used for controlling the FW task. This semaphore needs to be posted from the Interrupt callback (interrupt context).

```c
static void timeevent_cb(void)
{
    if(((*wakeup_event_pending >> 24) & 0xFF) == 0)
    {
        *wakeup_event_pending &= ~(0xFF << 24);
        *wakeup_event_pending |= (0x1 << 24);
    }
    platform_set_wakeup();
}
```

In `platform_drv.c`, add the following function definition:

```c
void platform_set_wakeup(void)
{
    os_sem_up(gstrFwSem);
}
```

**Note:** `os_sem_up` and `gstrFwSem` are initialized in `platform_driver_init (platform_drv.c)`.

**Note:** All the above changes are applicable for gcc (ASF) applications. For Keil application, changes in `platform_drv.c` are not needed as the changes are already part of driver library (`driver_lib.lib`). The new value for "enum port_wakeup_source" is added as "WAKEUP_SOURCE_TIMER" in port.h for Keil package. Hence Step 2 above is not needed for Keil.
The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.


Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, maXSylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntellIMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.
Quality Management System Certified by DNV

ISO/TS 16949
Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Office</td>
<td>Australia - Sydney</td>
<td>Tel: 61-2-9868-6733</td>
<td></td>
</tr>
<tr>
<td>2355 West Chandler Blvd.</td>
<td>China - Beijing</td>
<td>Tel: 86-10-8569-7000</td>
<td></td>
</tr>
<tr>
<td>Chandler, AZ 85224-6199</td>
<td>China - Chengdu</td>
<td>Tel: 86-28-8665-5511</td>
<td></td>
</tr>
<tr>
<td>Tel: 480-792-7200</td>
<td>China - Chongqing</td>
<td>Tel: 86-23-8980-9588</td>
<td></td>
</tr>
<tr>
<td>Fax: 480-792-7277</td>
<td>China - Dongguan</td>
<td>Tel: 86-769-8702-9880</td>
<td></td>
</tr>
<tr>
<td>Technical Support:</td>
<td>China - Guangzhou</td>
<td>Tel: 86-20-8755-8029</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.microchip.com/support">http://www.microchip.com/support</a></td>
<td>China - Hangzhou</td>
<td>Tel: 86-571-8792-8115</td>
<td></td>
</tr>
<tr>
<td>Web Address:</td>
<td>China - Hong Kong SAR</td>
<td>Tel: 852-2943-5100</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.microchip.com">www.microchip.com</a></td>
<td>China - Nanjing</td>
<td>Tel: 86-25-8473-2460</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Qingdao</td>
<td>Tel: 86-632-8502-7355</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shanghai</td>
<td>Tel: 86-21-3326-8000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenyang</td>
<td>Tel: 86-24-2334-2829</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenzhen</td>
<td>Tel: 86-755-8864-2200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Suzhou</td>
<td>Tel: 86-186-6233-1526</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Wuhan</td>
<td>Tel: 86-27-5980-5300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xian</td>
<td>Tel: 86-29-8833-7252</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xiamen</td>
<td>Tel: 86-592-2388138</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Zhuhai</td>
<td>Tel: 86-756-3210040</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - Bangalore</td>
<td>Tel: 91-80-3090-4444</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - New Delhi</td>
<td>Tel: 91-11-4160-8631</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - Pune</td>
<td>Tel: 91-20-4121-0141</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan - Osaka</td>
<td>Tel: 81-6-6152-7160</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan - Tokyo</td>
<td>Tel: 81-3-6880-3770</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Daegu</td>
<td>Tel: 82-53-744-4301</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Seoul</td>
<td>Tel: 82-2-554-7200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Kuala Lumpur</td>
<td>Tel: 60-3-7561-7906</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Penang</td>
<td>Tel: 60-4-227-8870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philippines - Manila</td>
<td>Tel: 63-2-634-9065</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Singapore</td>
<td>Tel: 65-6334-8870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Hsin Chu</td>
<td>Tel: 886-3-577-8366</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Kaohsiung</td>
<td>Tel: 886-7-213-7830</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Taipei</td>
<td>Tel: 886-2-2508-8600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thailand - Bangkok</td>
<td>Tel: 66-2-694-1351</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vietnam - Ho Chi Minh</td>
<td>Tel: 90-24-2548-2100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Austria - Wels</td>
<td>Tel: 43-7242-2244-39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Belgium - Brussels</td>
<td>Tel: 32-2-2541-390</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Beijing</td>
<td>Tel: 86-10-8569-7000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Chongqing</td>
<td>Tel: 86-23-8980-9588</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Dongguan</td>
<td>Tel: 86-769-8702-9880</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Guangzhou</td>
<td>Tel: 86-20-8755-8029</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Hangzhou</td>
<td>Tel: 86-571-8792-8115</td>
<td></td>
</tr>
<tr>
<td></td>
<td>France - Paris</td>
<td>Tel: 33-1-69-53-63-20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Germany - Munich</td>
<td>Tel: 49-9984-2144-04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - Bangalore</td>
<td>Tel: 91-80-3090-4444</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - New Delhi</td>
<td>Tel: 91-11-4160-8631</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - Pune</td>
<td>Tel: 91-20-4121-0141</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan - Osaka</td>
<td>Tel: 81-6-6152-7160</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan - Tokyo</td>
<td>Tel: 81-3-6880-3770</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Daegu</td>
<td>Tel: 82-53-744-4301</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Seoul</td>
<td>Tel: 82-2-554-7200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Kuala Lumpur</td>
<td>Tel: 60-3-7561-7906</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Penang</td>
<td>Tel: 60-4-227-8870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philippines - Manila</td>
<td>Tel: 63-2-634-9065</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Singapore</td>
<td>Tel: 65-6334-8870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Hsin Chu</td>
<td>Tel: 886-3-577-8366</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Kaohsiung</td>
<td>Tel: 886-7-213-7830</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Taipei</td>
<td>Tel: 886-2-2508-8600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thailand - Bangkok</td>
<td>Tel: 66-2-694-1351</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vietnam - Ho Chi Minh</td>
<td>Tel: 90-24-2548-2100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>