

# BLE Alert Notification Profile

1.0

## Features

- BLE Alert Notification Service (ANS) GATT Client in GAP Peripheral role
- Support of “Email”, “Missed call” and “SMS/MMS” category IDs
- DeepSleep low power mode
- Workflow status reporting through UART
- LED status and incoming call indication

## General Description

This example project demonstrates the Alert Notification Client operation of the BLE PSoC Creator Component. The Alert Notification Client utilizes the BLE Alert Notification Profile with one instance of Alert Notification Service to receive information about “Email”, “Missed call” and “SMS/MMS” alerts from Alert Notification Server. Device remains in Sleep mode between BLE connection intervals.

## Development Kit Configuration

Default CY8CKIT-042 BLE Pioneer Kit configuration.

## Project Configuration

### BLE Alert Notification Profile Example project

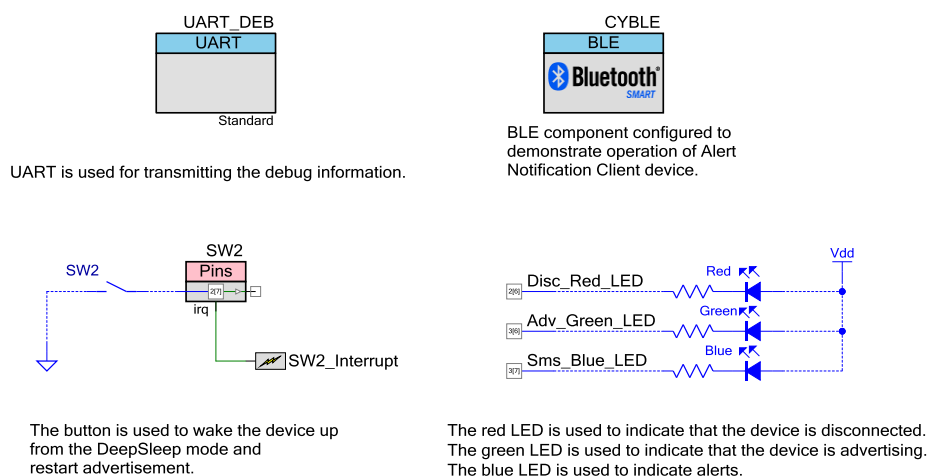


Figure 1. Top design schematic

The BLE (CYBLE) component is configured as Alert Notification Client in the GAP Peripheral role.

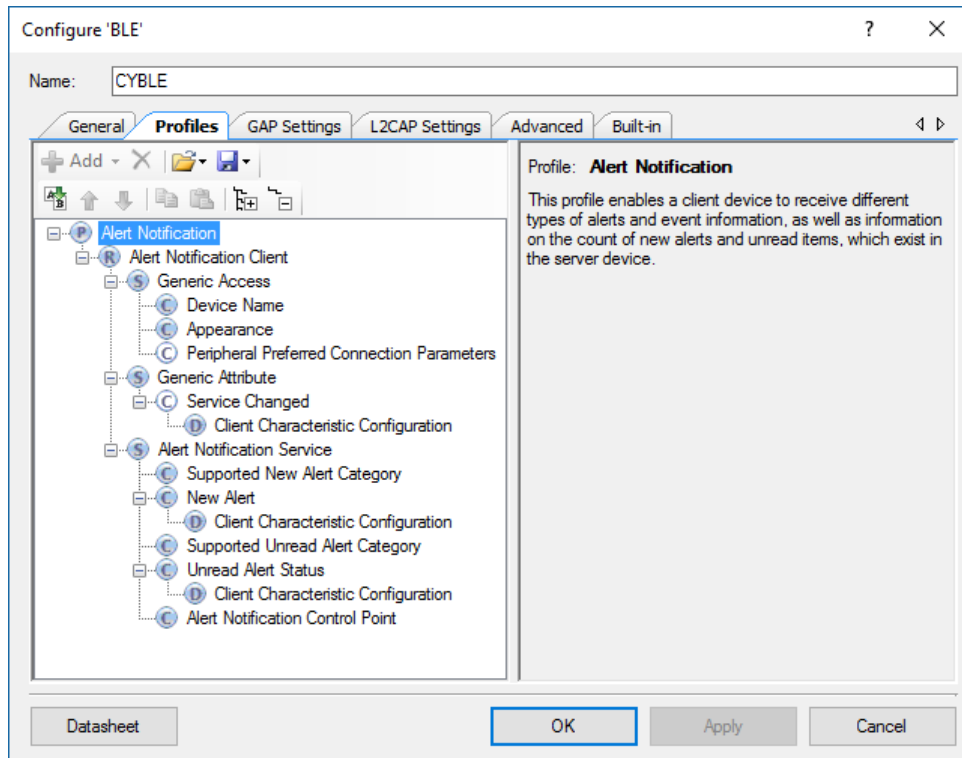


Figure 2. GATT Settings

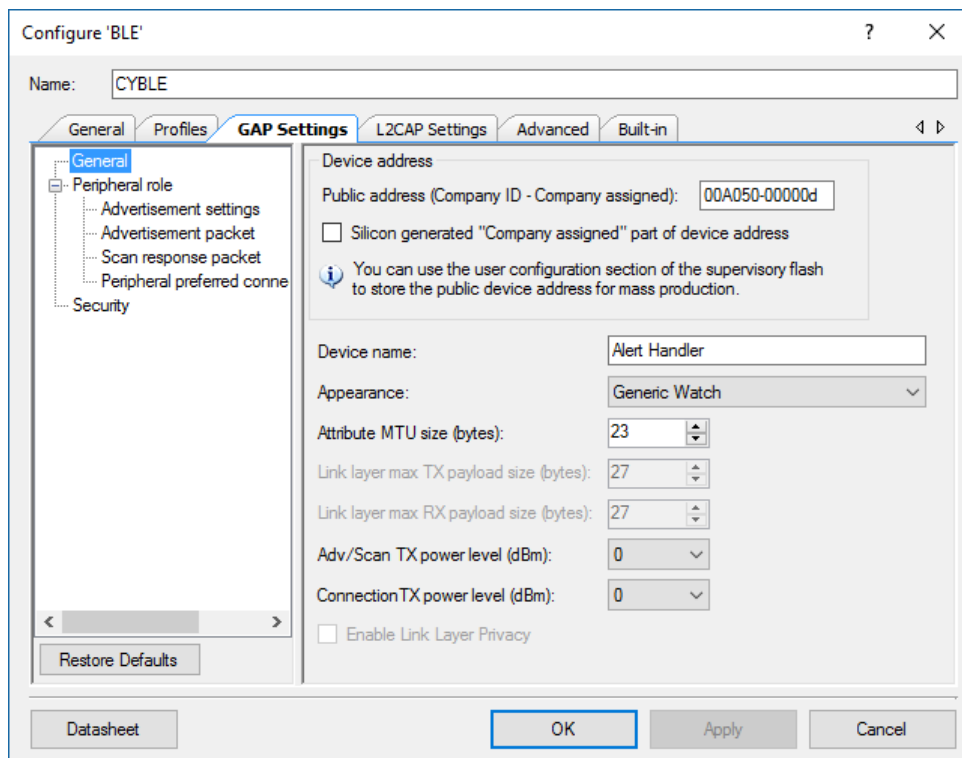


Figure 3. GAP Settings

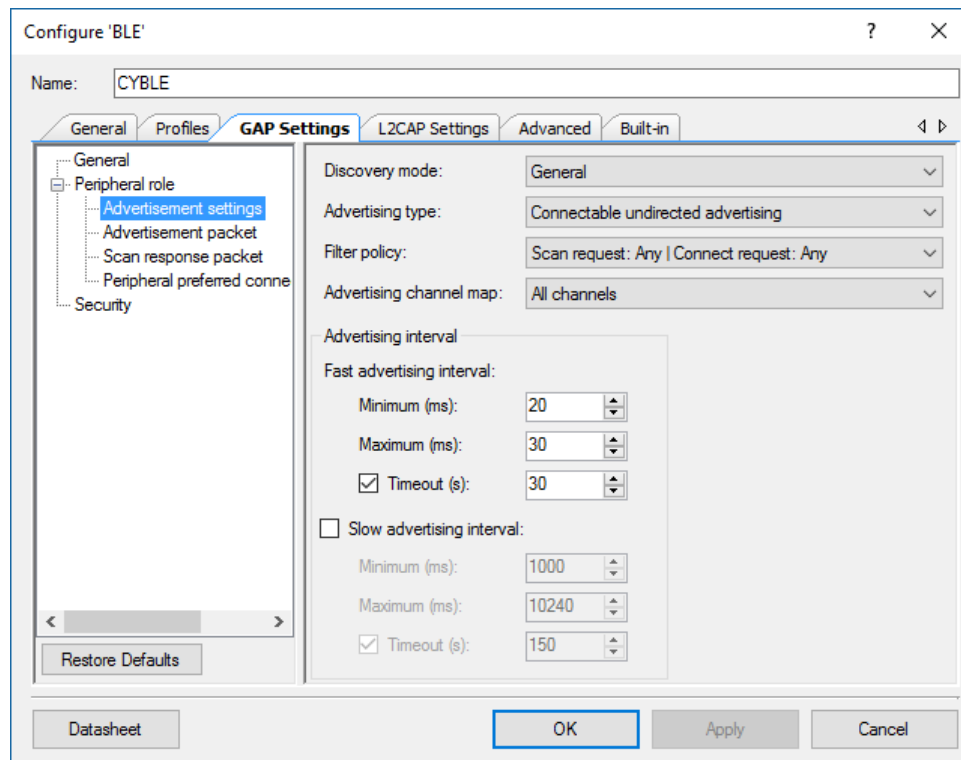


Figure 4. GAP Settings -&gt; Advertisement settings

Configure 'BLE'

Name: CYBLE

General Profiles **GAP Settings** L2CAP Settings Advanced Built-in

General

- Peripheral role
  - Advertisement settings
  - Advertisement packet
  - Scan response packet
  - Peripheral preferred connection parameters
- Security

Advertisement data settings:

Name	Value
<input checked="" type="checkbox"/> Flags	
<input checked="" type="checkbox"/> General discoverable mode	
<input checked="" type="checkbox"/> BR/EDR not supported	
<input checked="" type="checkbox"/> Local Name	
Local name	Complete
<input type="checkbox"/> TX Power Level	
<input type="checkbox"/> Slave Connection Interval Range	
<input type="checkbox"/> Service UUID	
<input checked="" type="checkbox"/> Service Solicitation	
<input checked="" type="checkbox"/> Alert Notification Service	
<input type="checkbox"/> Service Data	
<input type="checkbox"/> Service Manager TK Value	
<input type="checkbox"/> Appearance	
<input type="checkbox"/> Public Target Address	
<input type="checkbox"/> Random Target Address	
<input checked="" type="checkbox"/> Advertising Interval	
Data	25 ms
<input type="checkbox"/> LE Bluetooth Device Address	
<input type="checkbox"/> LE Role	
<input type="checkbox"/> URI	
<input type="checkbox"/> Manufacturer Specific Data	

Advertisement packet:

Description	Value	Index
AD Data 1: <<Flags>>		
Length	0x02	[0]
<<Flags>>	0x01	[1]
BR/EDR not supported   General discoverable mode	0x06	[2]
AD Data 2: <<Local Name>>		
Length	0x0E	[3]
<<Local Name>>	0x09	[4]
'A'	0x41	[5]
'I'	0x6C	[6]
'e'	0x65	[7]
'r'	0x72	[8]
't'	0x74	[9]
''	0x20	[10]
'H'	0x48	[11]
'a'	0x61	[12]
'n'	0x6E	[13]
'd'	0x64	[14]
'l'	0x6C	[15]
'e'	0x65	[16]
'r'	0x72	[17]
AD Data 3: <<16-bit Service UUIDs>>		
Length	0x03	[18]
<<16-bit Service UUIDs>>	0x14	[19]
Service: Alert Notification Service		
[0]	0x11	[20]
[1]	0x18	[21]
AD Data 4: <<Advertising Interval>>		
Length	0x03	[22]
<<Advertising Interval>>	0x1A	[23]
Value: 25 ms		
[0]	0x28	[24]
[1]	0x00	[25]

Restore Defaults

Datasheet

OK Apply Cancel

Figure 5. GAP Settings -&gt; Advertisement packet

Configure 'BLE' ? X

Name: CYBLE

General Profiles **GAP Settings** L2CAP Settings Advanced Built-in

General  
Peripheral role  
Advertisement settings  
Advertisement packet  
Scan response packet  
Peripheral preferred connection  
Security

Scan response data settings:

Name	Value
<input checked="" type="checkbox"/> Local Name	
Local name	Complete
<input type="checkbox"/> TX Power Level	
<input type="checkbox"/> Slave Connection Interval Range	
<input type="checkbox"/> Service UUID	
<input type="checkbox"/> Service Solicitation	
<input type="checkbox"/> Service Data	
<input type="checkbox"/> Service Manager TK Value	
<input type="checkbox"/> Appearance	
<input type="checkbox"/> Public Target Address	
<input type="checkbox"/> Random Target Address	
<input type="checkbox"/> Advertising Interval	
<input type="checkbox"/> LE Bluetooth Device Address	
<input type="checkbox"/> LE Role	
<input type="checkbox"/> URI	
<input type="checkbox"/> Manufacturer Specific Data	

Scan response packet:

Description	Value	Index
AD Data 1: <<Local Name>>		
Length	0x0E	[0]
<<Local Name>>	0x09	[1]
'A'	0x41	[2]
'I'	0x6C	[3]
'e'	0x65	[4]
'r'	0x72	[5]
'i'	0x74	[6]
''	0x20	[7]
'H'	0x48	[8]
'a'	0x61	[9]
'n'	0x6E	[10]
'd'	0x64	[11]
'l'	0x6C	[12]
'e'	0x65	[13]
'r'	0x72	[14]

Restore Defaults

Datasheet

OK Apply Cancel

Figure 6. GAP Settings -&gt; Scan response packet

Configure 'BLE' ? X

Name: CYBLE

General Profiles **GAP Settings** L2CAP Settings Advanced Built-in

General  
Peripheral role  
Advertisement settings  
Advertisement packet  
Scan response packet  
Peripheral preferred connection  
**Security**

Security mode: Mode 1

Security level: Authenticated pairing with encryption

Strict pairing: No

I/O capabilities: Display

Keypress notifications: No

Bonding requirement: Bonding

Encryption key size (bytes): 16

Restore Defaults

Datasheet

OK Apply Cancel

Figure 7. GAP Settings -&gt; Security

## Project Description

The project demonstrates the core functionality of the BLE component configured as an Alert Notification Client in the GAP Peripheral role. For operation the example project requires the Alert Notification Server configured in the GAP Central role. The example project requires CY8CKIT-042-BLE Pioneer Kit with PSoC 4100-BL, PSoC 4200-BL or PSoC 4000-BL devices.

For operation the example project uses two callback functions - AppCallBack() and AnsServiceAppEventHandler(). One callback function (AppCallBack()) is required for receiving the generic events from the BLE Stack and the second (AnsServiceAppEventHandler()) is required for receiving the events from the Alert Notification Service.

To start the example project operation, build it and program into development kit. Right after the startup, the BLE, UART, and ISR components are initialized. After the initialization the BLE component begins its operation that can be seen on the RGB LED which starts blinking with a green color. This indicates that the device has started advertising and it is available for the connection with a central device. For advertisement the example project uses the packet structure as shown in **Figure 5**. After 30 seconds, if no central device has connected to the Alert Notification Client, the Client stops advertisement and red LED is turned on indicating the disconnection state. To connect to the Alert Notification Client device, send a connection request to the device when the device is advertising. Note that for operation the example project requires bonding procedure which is described below.

While connected to the Server and between connection intervals, the device is put into the Sleep mode.

The example project uses UART component for displaying debug information as also for entering commands through the Terminal emulator app. Freeware, such as HyperTerminal, Bray's Terminal, Putty etc., is available on the web and can be used with this example. Commands are the procedures which user can perform. The list of the commands is shown below:

Command	Description
'd'	Send disconnect request to peer device.
'c'	Turn off LED of New Alert Category that was notified previously. Alert category is specified by the following commands: 'e' - "Email", 'm' - "Missed call", 's' - "SMS/MMS".
'b'	Initiate bonding procedure with peer device.
'e'	Send request with immediate notification command for New Alert Characteristic with Category ID set to "Email".
'm'	Send request with immediate notification command for New Alert Characteristic with Category ID set to "Missed Call".
's'	Send request with immediate notification command for New Alert Characteristic with Category ID set to "SMS/MMS".
'r'	Read Supported New Alert Category Characteristic. This command required for Client to configure local supported categories setting prior sending notification request to the Server.
't'	Send "Enable New Alert Notification" command to Alert Notification Control Point Characteristic. Category ID is set to "All categories".

'o'	Send "Disable New Alert Notification" command to Alert Notification Control Point Characteristic. Category ID is set to "All categories".
'0'	Enable notifications for New Alert Characteristic.
'1'	Disable notifications for New Alert Characteristic.

The above list is prompted to Terminal emulator when 'i' is entered in the app.

## Bonding

As the example project uses authentication and encryption, it requires a bonding procedure to establish a protected connection. The device will initiate the bonding procedure automatically after it is connected to the Server. The client side terminal emulator will display the passkey to be entered in the Server.

```
CYBLE_EVT_GAP_PASSKEY_DISPLAY_REQUEST. Passkey is: 025121.
Please enter the passkey on your Server device.
```

Note that the passkey prompted by the Terminal emulator will be different. The passkey above is shown only as an example.

After entering 6 digit passkey on the Server device the following message should be displayed on the Terminal emulator:

```
CYBLE_EVT_GAP_ENCRYPT_CHANGE: 1

CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT

CYBLE_EVT_GAP_AUTH_COMPLETE: security: 0x2, bonding: 0x1, ekeySize:
0x10, authErr 0x0
Bonding complete.
```

This message indicates a secure connection between two devices was established.

## Alert Notifications

After a Server is successfully connected to the Alert Notification Client and after successful bonding, the Client is ready for alert notification. Make sure that peer device Server (Operation system or Application) supports ANS - "Peer device supports ANS" message on Terminal emulator will confirm it. Use '0' command to enable notifications for the New Alert Characteristic. After that, send 't' command to enable notifications for all categories in the Alert Notification Control Point Characteristic. Next, use 's', 'm' or 'e' commands to request Server to send immediate notification for "SMS/MMS", "Email" or "Missed call" category. As the result, a message like following will be displayed on Terminal emulator:

```
New Alert Characteristic notification received.
Notified value is following:
Category ID - SMS/MMS.
Number of alerts: 5.
```

Additionally, the RGB LED on the CY8CKIT-042-BLE Pioneer Kit will be indicating like following:

Green LED is turned on if alert notification for “Email” category is received;  
Red LED is turned on if alert notification for “Missed call” category is received;  
Blue LED is turned on if alert notification for “SMS/MMS” category is received.

## Expected Results

Green LED is blinking when device is advertising and also turned on if the “Email” alert is received.

Red LED is turned on when device is in disconnected state and also if the “Missed call” alert is received.

Blue LED is blinking or turned on when device is alerting and also if the “SMS/MMS” alert is received.

### Example of Terminal emulator output:

```
Bluetooth On, Start advertisement with addr:00a0500000d.  
Device is entered Limited Discovery mode.
```

```
Advertisement is enabled
```

```
CYBLE_EVT_GATT_CONNECT_IND: 0
```

```
CYBLE_EVT_DEVICE_CONNECTED: 4  
Start Discovery
```

```
Discovery complete.  
Discovered services:  
Service with UUID 0x1800 has handle range from 0x1 to 0x5  
Service with UUID 0x1801 has handle range from 0x6 to 0x9  
Service with UUID 0x1811 has handle range from 0xa to 0x16  
Peer device supports ANS
```

```
CYBLE_EVT_GAP_AUTH_REQ: security=0x2, bonding=0x1, ekeySize=0x10,  
err=0x0
```

```
CYBLE_EVT_GAP_PASSKEY_DISPLAY_REQUEST. Passkey is: 044949.  
Please enter the passkey on your Server device.
```

```
CYBLE_EVT_GAP_ENCRYPT_CHANGE: 1
```

```
CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT
```

```
CYBLE_EVT_GAP_AUTH_COMPLETE: security: 0x2, bonding: 0x1, ekeySize:  
0x10, authErr 0x0  
Bonding complete.
```



CyBle\_AnscSetCharacteristicDescriptor routine Success

ANS Characteristic's Descriptor was written successfully.

CyBle\_AnscSetCharacteristicValue(CYBLE\_ANS\_ALERT\_NTF\_CONTROL\_POINT)  
routine Success

Alert Notification Control Point Characteristic was written  
successfully.

CyBle\_AnscSetCharacteristicValue(CYBLE\_ANS\_ALERT\_NTF\_CONTROL\_POINT)  
routine Success

New Alert Characteristic notification received.

Notified value is following:

Category ID - Email.

Number of alerts: 1.

© Cypress Semiconductor Corporation, 2009-2016. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.