

BLE Continuous Glucose Monitoring Sensor

1.0

Features

- BLE Continuous Glucose Monitoring and Bond Management services in GATT Server role
- Low Power mode
- Workflow status reporting through UART
- LED status indication

General Description

This example project demonstrates the BLE Continuous Glucose Monitoring Sensor application workflow. The application uses the BLE Continuous Glucose Monitoring Profile to report CGM Measurement records to the Client by the Continuous Glucose Monitoring Service and to manage bonding by the Bond Management Service. Also, the application uses the Device Information Service to assert the Device Name, etc. The PSoC 4 BLE and PSoC BLE devices are supported.

Development Kit Configuration

Default CY8CKIT-042 BLE Pioneer Kit configuration.

Project Configuration

BLE Continuous Glucose Monitoring Sensor Example Project

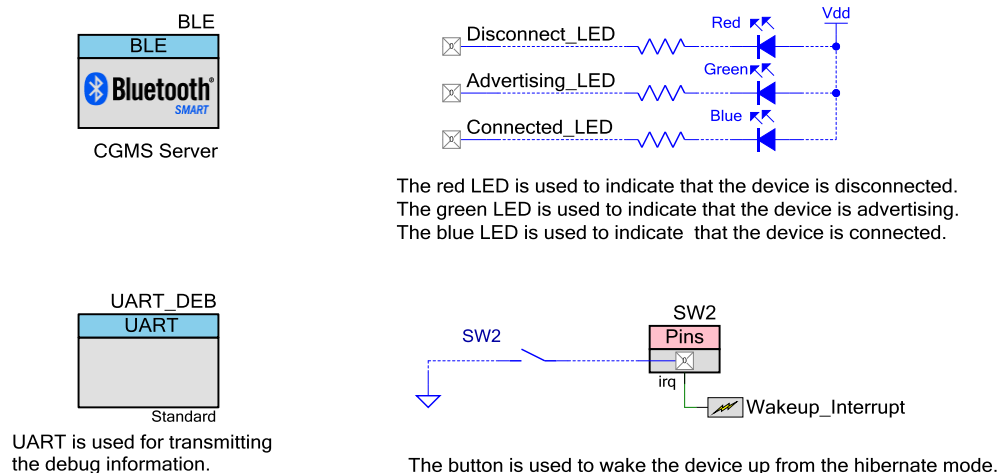


Figure 1. Top Design Schematic

The BLE component is configured as Continuous Glucose Monitoring Server in the GAP Peripheral role. Also, Bond Management and Device Information Services are included.

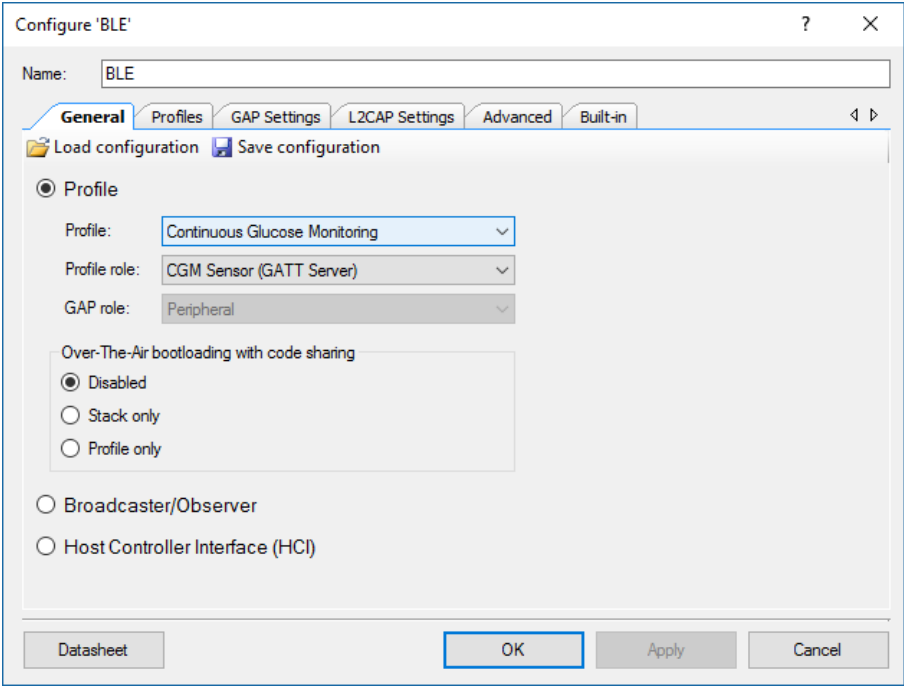


Figure 2. General Settings

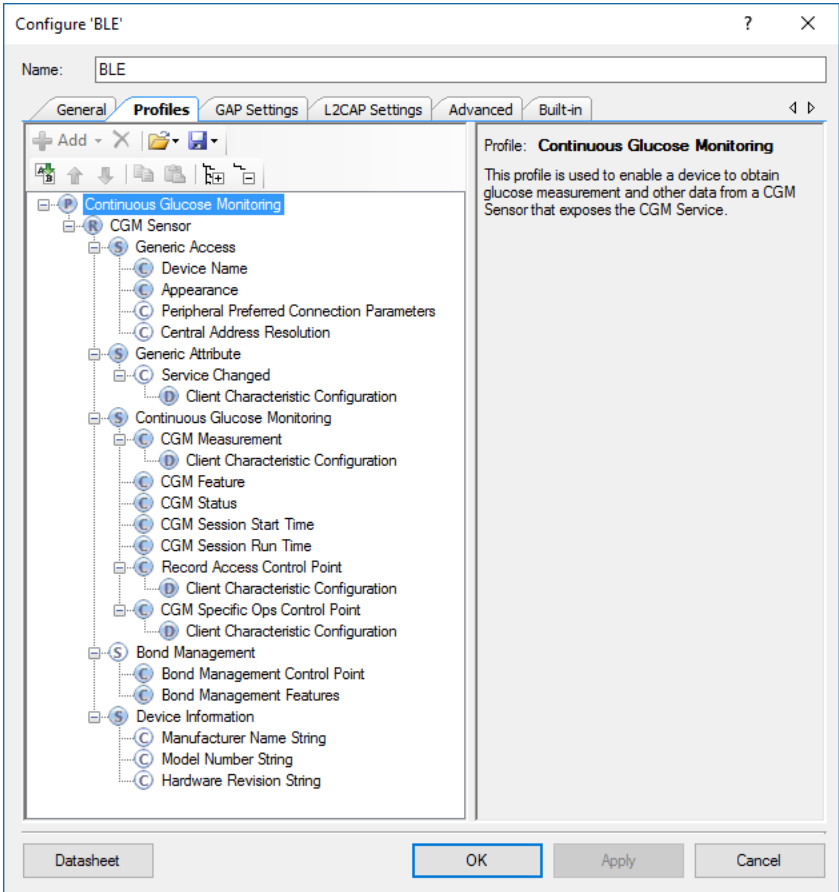


Figure 3. GATT Settings

Configure 'BLE'

Name: BLE

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

General

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

Device address

Public address (Company ID - Company assigned): 00A050-00001a

☐ Silicon generated "Company assigned" part of device address

You can use the user configuration section of the supervisory flash to store the public device address for mass production.

Device name: Glucose Monitor Sensor

Appearance: Generic Glucose Meter

Attribute MTU size (bytes): 23

Link layer max TX payload size (bytes): 27

Link layer max RX payload size (bytes): 27

Adv/Scan TX power level (dBm): 0

Connection TX power level (dBm): 0

☐ Enable Link Layer Privacy

Restore Defaults

Datasheet OK Apply Cancel

Figure 4. GAP Settings

Configure 'BLE'

Name: BLE

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

General

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

Discovery mode: General

Advertising type: Connectable undirected advertising

Filter policy: Scan request: Any | Connect request: Any

Advertising channel map: All channels

Advertising interval

Fast advertising interval:

Minimum (ms): 20

Maximum (ms): 30

☒ Timeout (s): 30

☒ Slow advertising interval:

Minimum (ms): 1000

Maximum (ms): 2500

☒ Timeout (s): 150

Restore Defaults

Datasheet OK Apply Cancel

Figure 5. GAP Settings -> Advertisement Settings

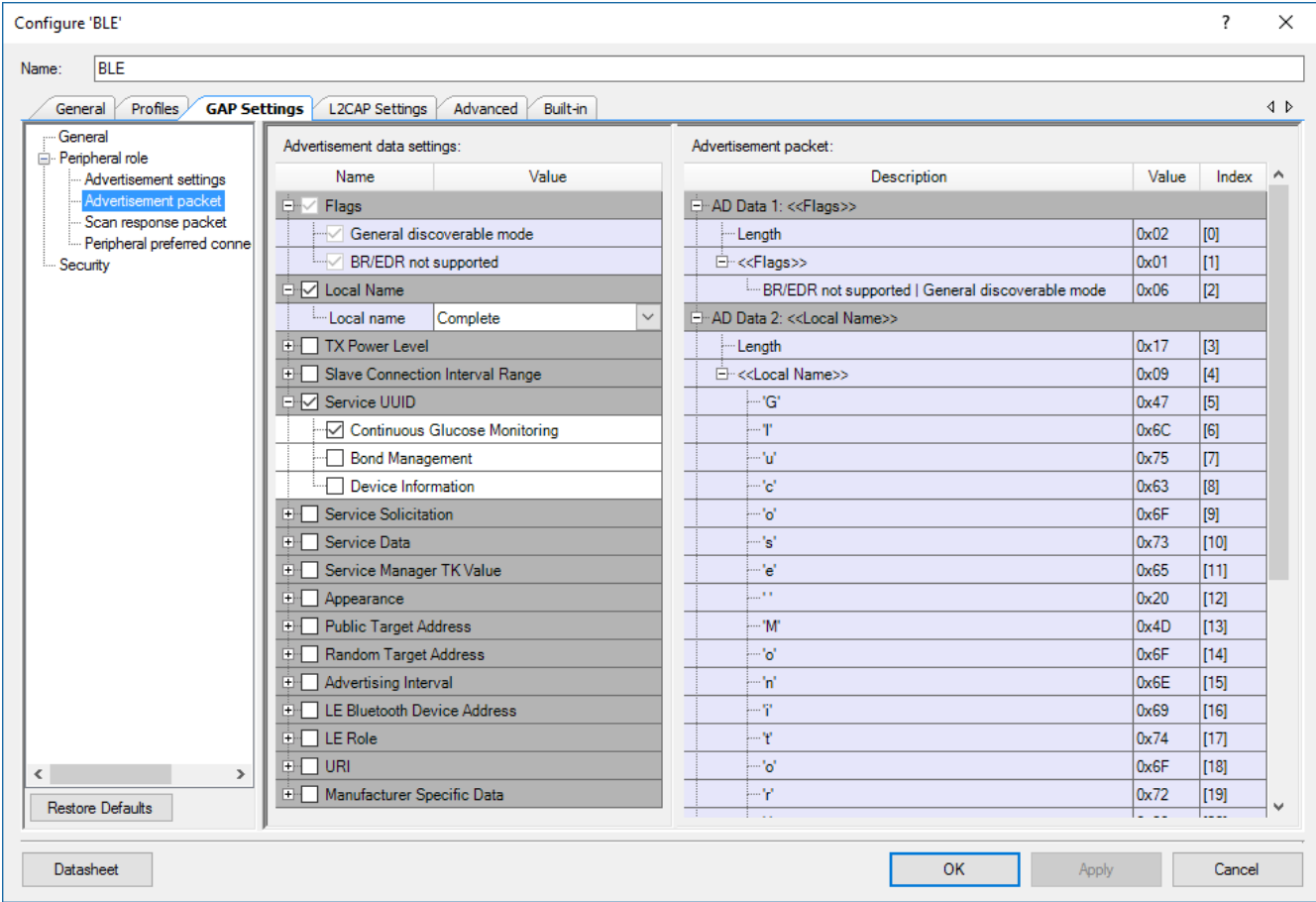


Figure 6. GAP Settings -> Advertisement Packet

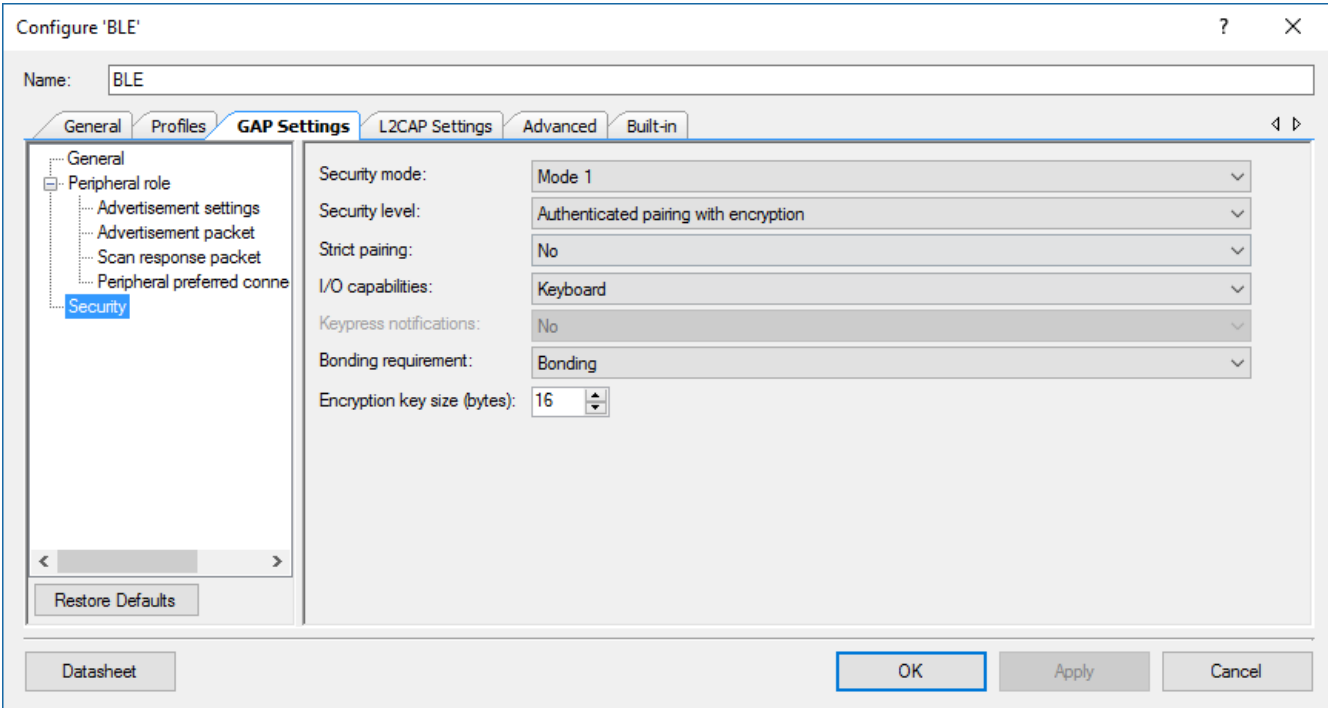


Figure 7. GAP Settings -> Security

The pin assignment is the next:

- The UART RX is connected to P1[4].
- The UART TX is connected to P1[5].
- A mechanical button is connected to P2[7].
- The red LED is connected to P2[6].
- The green LED is connected to P3[6].
- The blue LED is connected to P3[7].

Project Description

The project demonstrates the core functionality of the BLE component configured as a Continuous Glucose Monitoring Server.

Right after a startup the device performs BLE component initialization. In this project three callback functions are required for BLE operation. One callback function (AppCallBack()) is required to receive generic events from BLE Stack and service-specific callbacks CgmsCallBack() and BmsCallBack() for Continuous Glucose Monitoring and Bond Management service-specific events accordingly. The CYBLE_EVT_STACK_ON event indicates a successful initialization of BLE Stack. After this event is received, the component starts advertising with the packet structure as configured in the BLE component customizer (see **Figure 6**). The BLE component stops advertising as soon as a 180-second advertising period expires.

The Continuous Glucose Monitoring Sensor device can be connected to any BLE-compatible device (4.0 or later), configured as the GAP Central role and GATT Client which supports Continuous Glucose Monitoring Profile. Also, the Device Information Service may be optionally used.

To connect to the Continuous Glucose Monitoring Sensor device, send a connection request to the device while the device is advertising. The green LED is blinking while the device is advertising. The blue LED is turned on when the device is in connected state. The red LED is turned on when the device is in a disconnected state to indicate that no Client is connected to the device.

The Continuous Glucose Monitoring Sensor device requires authentication, the IO capability is “keyboard” (see **Figure 7**) i.e. the device requires entering the passkey that is indicated by the Client device through the UART interface (by any software, e.g. like HyperTerminal). When the Client is paired with the Continuous Glucose Monitoring Sensor, at first, the CGM Measurement characteristic notification and the Record Access Control Point (RACP) characteristic indication should be enabled. Then the RACP characteristic can be written to assert any RACP request (for details, see the Continuous Glucose Monitoring Profile and Service specifications adopted by Bluetooth SIG). When the RACP request is asserted, the Client should wait for any CGM Measurement characteristic notifications and the Record Access Control Point (RACP) characteristic indication (depending on the asserted request), or write the **Abort Operation** command into the RACP characteristic. Before writing any CGM Specific Ops Control Point (SOCPP) command, the SOCPP indication should be enabled. The WDT is used for time simulations and LED blinking.

While connected to the Client and between connection intervals, the device is put into Sleep Mode.

Expected Results

The project sends the Continuous Glucose Monitoring Service characteristic's notifications/indications to the Central Client device which shows them to the user. Also, the device performs Bond Management service operations requested by the Client. LEDs are blinking as described in the *Project Description* section. And the project sends log messages through the UART.

The project is intended to work in pair with any BLE-compatible device (e.g. phone, tablet) with appropriate software (e.g. Android, iOS with an installed application that supports Continuous Glucose Monitoring Profile.)

Also, the Continuous Glucose Monitoring Sensor can be used together with [CySmart app for Windows](#). It is required to match the security settings between Continuous Glucose Monitor and CySmart Client and perform pairing (bonding) before any writing (enabling notifications etc.) into the Server's GATT database. For further instructions on how to use the CySmart application, refer to [CySmart User Guide](#).

To use the CySmart Windows application as a Continuous Glucose Monitoring Service client:

- Connect the CySmart BLE dongle to a USB port on the PC.
- Launch CySmart app and select the connected dongle in the dialog window.
- Reset the development kit to start advertising by pressing the **SW1** button.
- Click the **Start Scan** button to discover available devices.
- Select the **Glucose Monitor Sensor** in the list of available devices and connect to it.
- Click **Pair**, enter a 6-digit passkey through the terminal, then **Discover All Attributes**, then **Read All Characteristics**, and finally **Enable All Notifications** in CySmart app.

- To reproduce the Continuous Glucose Monitoring Service functionality, for example, select the Record Access Control Point (RACP) characteristic value and write the command “01 01” which means “Report All Glucose Measurement Records” (all these commands are described in detail in [CGMS Specification](#)):

The screenshot shows the CySmart 1.1 application window. The main panel displays a list of attributes for a 'Glucose Monitor Sensor [00:A0:50:00:00:1A]'. The attributes are organized into categories like 'Primary Service Declaration', 'Characteristic Declaration', and 'Client Characteristic Configuration'. The 'Record Access Control Point' characteristic is highlighted, showing its handle as 0x001B and its value as 28:18:00:52:2A.

On the right, the 'Attribute Details' panel shows the selected attribute's details. The 'Value' field is set to '01:01', which is circled in red. Below this, the 'Write Value' button is also circled in red.

At the bottom, the 'Log' panel shows a series of events, including 'Event Data: [01]', 'Misc Event Notification' received, 'Event Code: 0x002C', 'Event Data: [0A:2D:F4:65:E3:BD:7B:49:1E:B4:C0:95:95:13:46:73]', 'Command Complete' event received, 'Status: BLE_STATUS_OK', 'Info: gpm.M1115: Pairing with Glucose Monitor Sensor [00:A0:50:00:00:1A] completed successfully.', and 'Get Whitelist' request sent.

- Observe the Server send three CGM Measurement characteristic notifications with the simulated data and the RACP indication "06 00 01 01" which means "The < Report All Glucose Measurement Records > command is performed successfully":

The screenshot shows the CySmart 1.1 application window. The main pane displays a list of attributes for a 'Glucose Monitor Sensor [00:A0:50:00:00:1A]'. The 'Primary Service Declaration: Continuous Glucose Monitoring' is expanded, showing several characteristics. The 'Characteristic Declaration: CGM Measurement' is selected, and its value is highlighted as '09:E0:32:B0:03:00:02:04:01'. The 'Characteristic Declaration: Record Access Control Point' is also selected, and its value is highlighted as '06:00:01:01'. The 'Log' pane at the bottom shows a series of events, including 'Write Characteristic Value' request sent, 'Command Status' event received, 'Command Complete' event received, and three 'Characteristic Value Notification' events received. The values for these notifications are highlighted in red: '00:E3:32:B0:01:00:02:04:01:00:00:00:00', '09:E0:32:B0:03:00:02:04:01:00:00:00', and '06:00:01:01'.

Handle	UUID	UUID Description	Value	Properties
Primary Service Declaration: Continuous Glucose Monitoring				
0x000E	0x2800	Primary Service Declaration	1F:18 (Continuous Glucose Monitoring)	
Characteristic Declaration: CGM Measurement				
0x000F	0x2803	Characteristic Declaration	10:10:00:A7:2A	
0x0010	0x2AA7	CGM Measurement	09:E0:32:B0:03:00:02:04:01	0x10
0x0011	0x2902	Client Characteristic Configuration	01:00	
Characteristic Declaration: CGM Feature				
0x0012	0x2803	Characteristic Declaration	02:13:00:A8:2A	
0x0013	0x2AA8	CGM Feature		0x02
Characteristic Declaration: CGM Status				
0x0014	0x2803	Characteristic Declaration	02:15:00:A9:2A	
0x0015	0x2AA9	CGM Status		0x02
Characteristic Declaration: CGM Session Start Time				
0x0016	0x2803	Characteristic Declaration	0A:17:00:AA:2A	
0x0017	0x2AAA	CGM Session Start Time		0x0A
Characteristic Declaration: CGM Session Run Time				
0x0018	0x2803	Characteristic Declaration	02:19:00:AB:2A	
0x0019	0x2AAB	CGM Session Run Time		0x02
Characteristic Declaration: Record Access Control Point				
0x001A	0x2803	Characteristic Declaration	28:1B:00:52:2A	
0x001B	0x2A52	Record Access Control Point	06:00:01:01	0x28
0x001C	0x2902	Client Characteristic Configuration	02:00	

Log

```

[19:06:16:943] : 'Write Characteristic Value' request sent
[19:06:16:943] : Attribute Handle: 0x001B
[19:06:16:943] : Value: [01:01]
[19:06:16:948] : 'Command Status' event received
[19:06:16:948] : Status: BLE_STATUS_OK
[19:06:16:963] : 'Command Complete' event received
[19:06:16:963] : Status: BLE_STATUS_OK
[19:06:16:976] : 'Characteristic Value Notification' event received
[19:06:16:976] : Attribute Handle: 0x0010
[19:06:16:976] : Value: [00:E3:32:B0:01:00:02:04:01:00:00:00:00]
[19:06:16:977] : 'Characteristic Value Notification' event received
[19:06:16:977] : Attribute Handle: 0x0010
[19:06:16:977] : Value: [09:E0:32:B0:03:00:02:04:01:00:00:00]
[19:06:16:977] : 'Characteristic Value Notification' event received
[19:06:16:977] : Attribute Handle: 0x0010
[19:06:16:977] : Value: [06:00:01:01]
[19:06:16:984] : 'Characteristic Value Indication' event received
[19:06:16:984] : Attribute Handle: 0x001B
[19:06:16:984] : Value: [06:00:01:01]
  
```

The correspondent example UART log is shown below:

```

Continuous Glucose Monitoring Sensor Example Project
BLE Stack Version: 3.1.0.179
CYBLE_EVT_STACK_ON
Start Advertisement with addr: 00a05000001a
CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP
state: advertising
CYBLE_EVT_GATT_CONNECT_IND: attId 0, bdHandle 4
  
```



```
CYBLE_EVT_GAP_DEVICE_CONNECTED: 4
CYBLE_EVT_GATTS_XCNHG_MTU_REQ
other event: 0x33
CYBLE_EVT_GAP_PASSKEY_ENTRY_REQUEST
Enter 6 digit passkey:021953 Passkey is sent
CYBLE_EVT_GAP_ENCRYPT_CHANGE: 1
CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT
CYBLE_EVT_GAP_AUTH_COMPLETE: security:2, bonding:0,
ekeySize:10, authErr 0
CYBLE_EVT_GATTS_INDICATION_ENABLED
RACP Indication is Enabled
RACP Indication is Enabled
Glucose Notification is Enabled
RACP characteristic is written: 01 01
Opcode: Report stored records
Operator: All records
Cgmt Ntf: 0d e3 32 b0 01 00 02 04 01 00 00 00 00
Cgmt Ntf: 0b e2 32 b0 02 00 02 04 01 00 00
Cgmt Ntf: 09 e0 32 b0 03 00 02 04 01
RACP Ind: 06 00 01 01
RACP Indication is Confirmed
```

The details about the CGMS characteristic data structures are in the [CGMS Specification](#).

© 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.