

# BLE Environmental Sensing Profile

1.0

## Features

- BLE ES Service Sensor (GATT Server) role operation
- Low Power mode
- LED status indication

## General Description

This example project demonstrates the Environmental Sensing Profile operation of the BLE PSoC Creator Component. The Environmental Sensor utilizes the Environmental Sensing Profile with one instance of Environmental Sensing and Device Information Services to simulate wind speed measuring. The Environmental Sensor operates with other devices that implement the Environmental Collector Profile. The device switches to the DeepSleep mode between BLE connection intervals. The BLE component supports the PSoC 4 BLE and PSoC BLE family devices.

## Development Kit Configuration

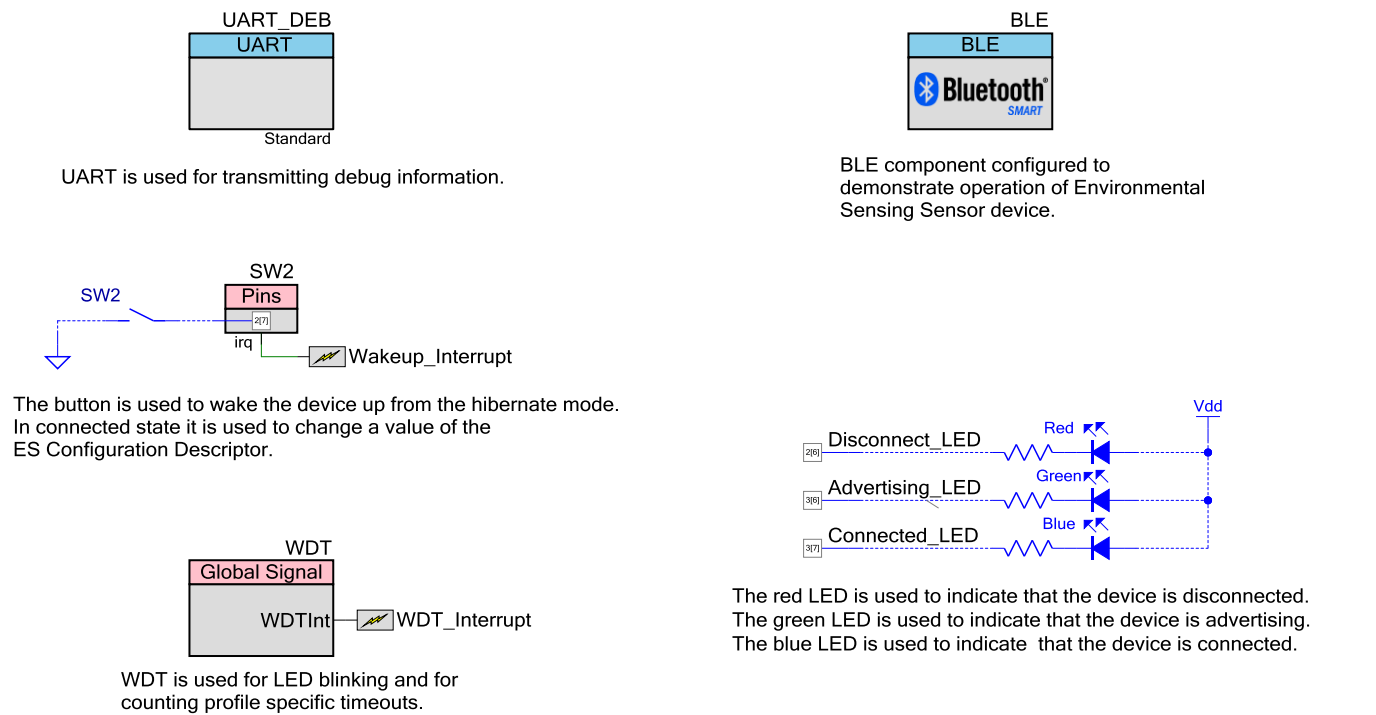
- Default CY8CKIT\_042 BLE Pioneer Kit configuration

## Project Configuration

The top design schematic is shown in **Figure 1**.

Figure 1. Top Design Schematic

BLE Environmental Sensing Profile Example Project



The BLE component is configured as Environmental Sensor in the GAP Peripheral role with the settings shown in the figures below.

Figure 2. GATT Settings

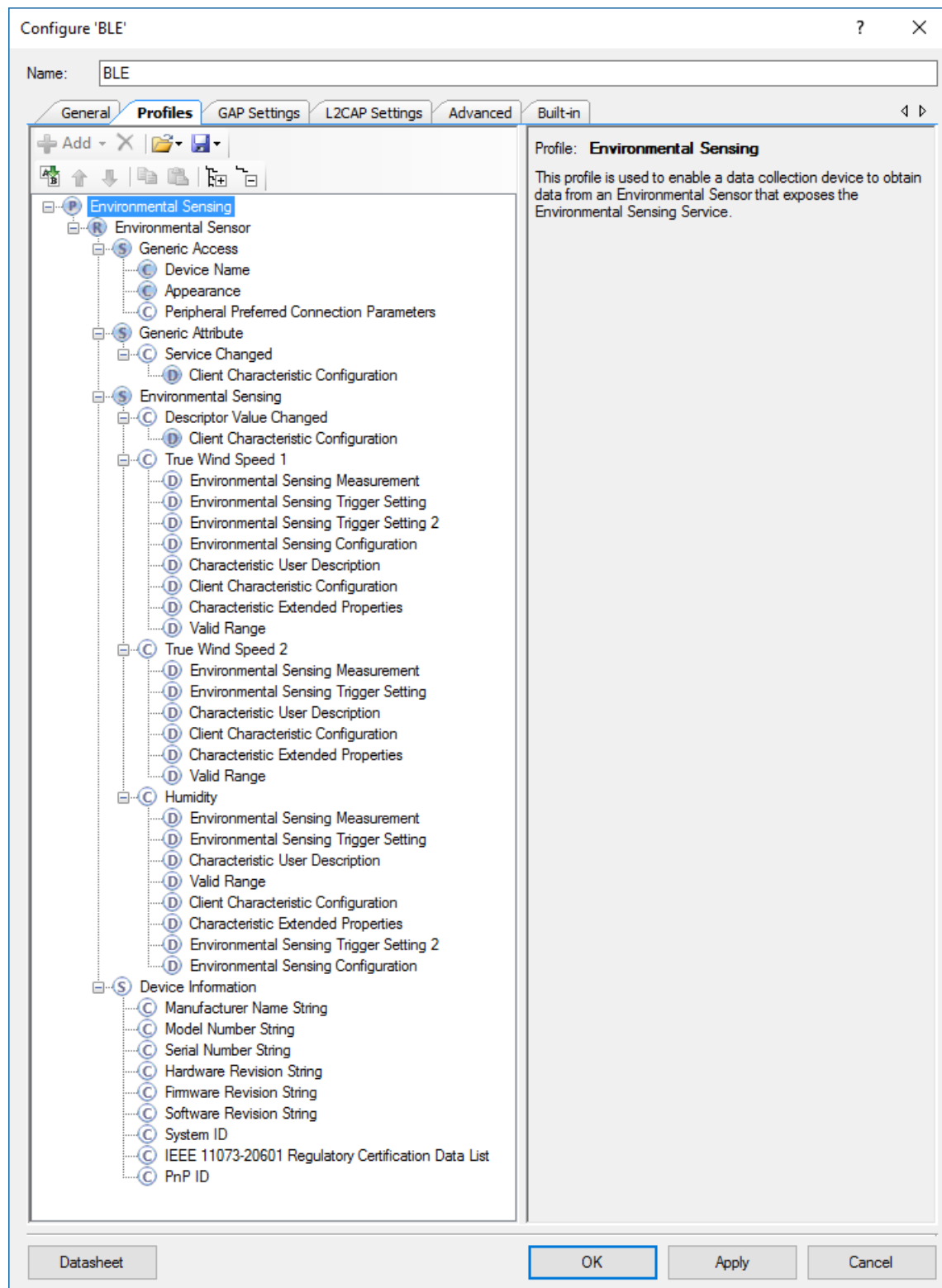


Figure 3. GAP Settings

Configure 'BLE'

Name: BLE

GeneralProfilesGAP SettingsL2CAP SettingsAdvancedBuilt-in

General

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

Device address

Public address (Company ID - Company assigned): 00A050-000019

☐ 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: Environmental Sensor

Appearance: Unknown

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

OKApplyCancel

Figure 4. GAP Settings -> Advertisement Settings

Configure 'BLE'

Name: BLE

GeneralProfilesGAP SettingsL2CAP SettingsAdvancedBuilt-in

General

- Peripheral role
  - Advertisement settings
  - Advertisement packet
  - Scan response packet
  - Peripheral preferred conn
- 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): 10240

☒ Timeout (s): 150

Restore Defaults

OKApplyCancel

Page 4 of 12



CYPRESS

Embedded in Tomorrow

Figure 5. GAP Settings -&gt; Advertisement Packet

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

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 type="checkbox"/> Service Solicitation	
<input checked="" type="checkbox"/> Service Data	
<input checked="" type="checkbox"/> Environmental Sensing	
Data: 00:00	
<input type="checkbox"/> Device Information	
<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	

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	0x15	[3]
<<Local Name>>	0x09	[4]
'E'	0x45	[5]
'n'	0x6E	[6]
'v'	0x76	[7]
'i'	0x69	[8]
'r'	0x72	[9]
'o'	0x6F	[10]
'n'	0x6E	[11]
'm'	0x6D	[12]
'e'	0x65	[13]
'n'	0x6E	[14]
'y'	0x74	[15]
'a'	0x61	[16]
'l'	0x6C	[17]
' '	0x20	[18]
'S'	0x53	[19]

Datasheet

OK Apply Cancel

Figure 6. Security 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

Security mode: Mode 1

Security level: Unauthenticated pairing with encryption

Strict pairing: No

I/O capabilities: No Input No Output

Keypress notifications: No

Bonding requirement: Bonding

Encryption key size (bytes): 16

Datasheet

OK Apply Cancel

## Project Description

The project demonstrates the core functionality of the BLE component configured as an Environmental Sensor.

Right after a startup the device performs initialization of the BLE component. In this project several callback functions are used for the BLE operation. One callback function (AppCallback()) is required for receiving generic events from the BLE, EssCallback() is required for receiving events from the Environmental Service. The CYBLE\_EVT\_STACK\_ON event indicates successful initialization of the BLE Stack. After this event is received, the component starts fast advertising with the packet structure as configured in the BLE component customizer (see **Figure 4**). Once the 30-second advertising period expires, the component switches to the slow advertisement parameters. On an advertisement event timed out, the device goes to Low Power mode (Hibernate mode) and waits for a SW2 button press to wake up the device again.

You can connect to the Environmental Sensor device with a BLE 4.0 or BLE 4.1 compatible device configured in the GAP Central role and capable of discovering the Environmental Sensing Service. To connect to an Environmental Sensor device, send a connection request to the device when the device is advertising. The green LED is blinking while the device is advertising. If the Client is connected to the Environmental Sensor, the blue LED is turned on.

While connected to the Client and between connection intervals, the device is put into DeepSleep mode.

### Wind Speed and humidity measuring simulation

The example project simulates data measured from two wind speed sensors (True Wind Speed Speed#1 and True Wind Speed Speed#2) and humidity sensor. Each of the sensors is configured to provide new measurement with respect to its Update Interval set in the ES Measurement Descriptor. Update Interval for True Wind Speed Speed#1 and #2 is configured for 15 and 5 seconds respectively. Update Interval for Humidity is configured for 10 seconds. Each of the sensors also has its Measurement Period which is 60 seconds for True Wind Speed Speed#1, 90 seconds for True Wind Speed Speed#2 and 40 seconds for Humidity. For more info on Measurement Period and Update Interval refer to the [Environmental Sensing Service specification](#). The device is configured to send notifications to the remote Client based on the trigger conditions captured in ES Trigger Settings Descriptors. Maximum number of ES Trigger Settings Descriptors is 3. Depending on the configuration in the ES Configuration Descriptor, the conditions in ES Trigger Settings Descriptors can be ORed or ANDed. The User Characteristic Descriptor is used for assigning the human-readable name of the Characteristic. The ES Trigger Settings, ES Configuration and User Characteristic Descriptor Descriptors are writable and can be set by the remote Client.

The example project allows configuring the simulation via the Customizer's GUI. The Measuring Period and Update Interval can be set in the ES Measurement Descriptor. The notification conditions can be configured in ES Trigger Settings Descriptor. Each of the characteristics can be assigned with default name through the User Characteristic Descriptor. The Valid Range Descriptor can be used to define the allowed ranges for the characteristic. In current example project the ranges are set to the maximum possible values.

The first wind sensor simulates an increase in the wind speed by 1.2 m/s each 15 seconds until it reaches the maximum of 80 m/s. Then the wind speed falls down to the minimum of 10

m/s, and then again it is increased by 1.2 m/s each 15 seconds. The second wind sensor simulates an increase in the wind speed by 0.2 m/s each 5 seconds until, it reaches the maximum of ~90 m/s. But notification of this parameter is every 20 seconds, because ES Trigger Settings Descriptor is set to **“No less than the specified time between transmissions”** in this example project and specified time is set to 20 seconds. The humidity sensor simulates an increase in the humidity by 1.40% each 10 seconds until it reaches the maximum of 99.00%. Then the humidity falls down to the minimum of 2%, and then again it is increased by 1.40% each 10 seconds. But value between 20.00% and 40.00% not notified, because ES Trigger Settings Descriptor is set to **“While less than the specified value”** (specified value is set to 20.00), ES Trigger Settings Descriptor 2 is set to **“While greater than the specified value”** (specified value is set to 40.00) and ES Configuration is set to **“Boolean OR”** for humidity in this example project. After that the speed is not updated any more holding the maximum wind speed.

## Using UART for debugging

In the example project UART component is used for printing various debug information. The printing of debug information is enabled by default but it can be disabled by changing the constant `DEBUG_UART_ENABLED` (common.h) from “YES” to “NO”.

A HyperTerminal program is required in the PC to receive debugging information. If you don't have a HyperTerminal program installed, download and install any serial port communication program. Freeware such as HyperTerminal, Bray's Terminal, Putty etc. is available on the web.

1. Connect the PC and kit with a USB cable.
2. Open the device manager program in your PC, find the COM port in which the kit is connected, and note the port number.
3. Open the HyperTerminal program and select the COM port in which the kit is connected.
4. Configure Baud rate, Parity, Stop bits and Flow control information in the HyperTerminal configuration window. By default, settings are following: Baud rate – 115200, Parity – None, Stop bits – 1 and Flow control – XON/XOFF. These settings have to match the configuration of the PSoC Creator UART component in the project
5. Start communicating with the device as explained in the project description.

## Expected Results

You can use CySmart app on a [Windows PC](#), [Android](#) or [iOS](#) BLE-compatible device as a Client for connection to the Environmental Sensor.

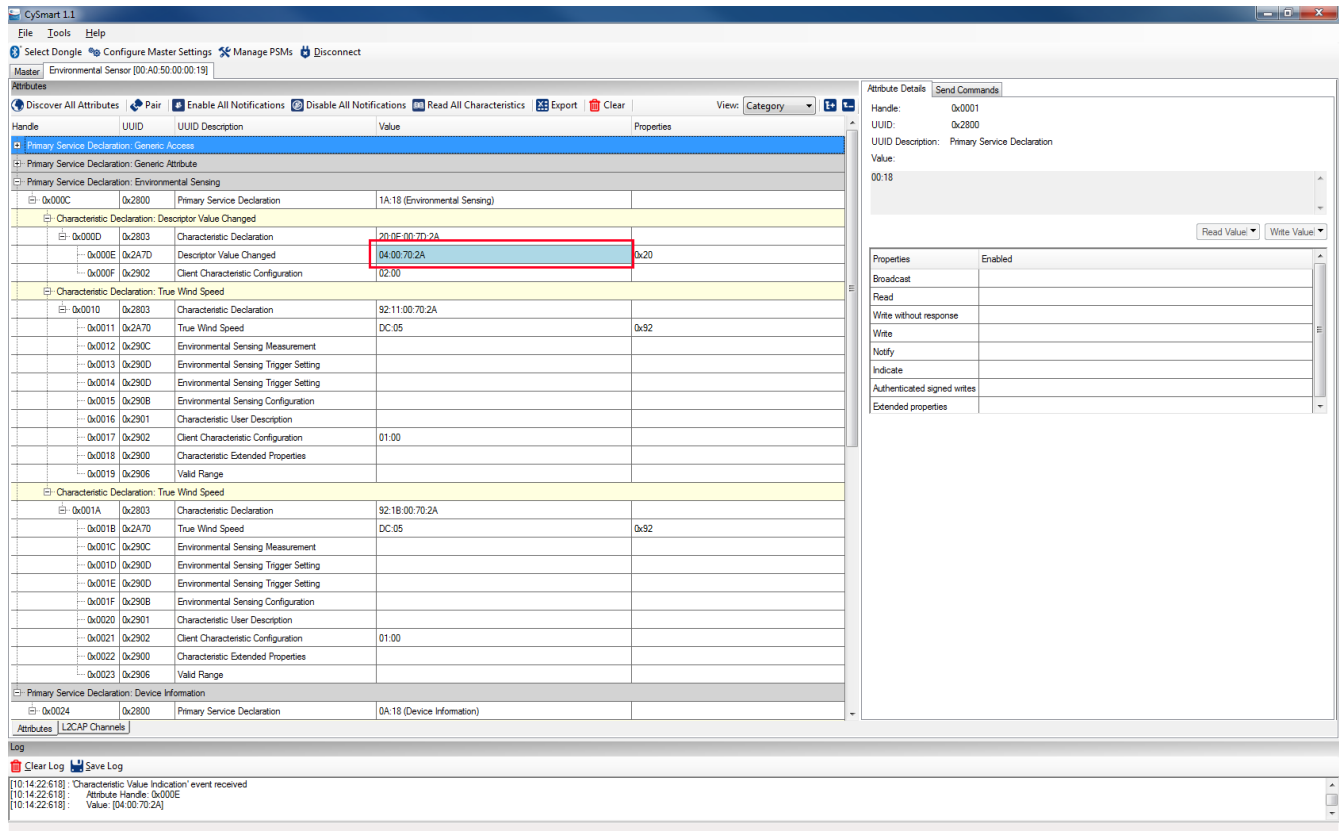
To use the CySmart Windows application as an Environmental Sensing Client:

- Connect the CySmart BLE dongle to a USB port on the PC.
- Launch the 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 **Environmental** in the list of available devices and connect to it.
- Click **Pair**, then **Discover All Attributes**, then **Read All Characteristics**, and finally **Enable All Notifications** in the CySmart app.
- Press the SW2 button on the kit and observe the indication about ES Configuration Descriptor change:

Figure 7. CySmart Windows app: Descriptor Value Changed Characteristic Indication



- Wait for at least 90 seconds (simulation of the measurement period) and observe that notifications for both of the True Wind Speed and Humidity Characteristics are received:



Figure 8. CySmart Windows app: True Wind Speed and Humidity Characteristics Notification

The screenshot displays the CySmart 1.1 application window. The main area shows a list of characteristics under the 'Attributes' tab. The 'Value' column for the 'True Wind Speed' characteristic (UUID 0x2A70) is highlighted with a red box, showing the value '80:11'. Similarly, the 'Value' column for the 'Humidity' characteristic (UUID 0x2A6F) is highlighted with a red box, showing the value '58:07'. The right panel shows the 'Attribute Details' for the selected characteristic, and the bottom panel shows a log of attribute changes.

Handle	UUID	UUID Description	Value	Properties
<b>Characteristic Declaration: True Wind Speed</b>				
0x0010	0x2803	Characteristic Declaration	92:11:00:70:2A	
0x0011	0x2A70	True Wind Speed	80:11	0x92
0x0012	0x290C	Environmental Sensing Measurement		
0x0013	0x290D	Environmental Sensing Trigger Setting		
0x0014	0x290D	Environmental Sensing Trigger Setting		
0x0015	0x290B	Environmental Sensing Configuration		
0x0016	0x2901	Characteristic User Description		
0x0017	0x2902	Client Characteristic Configuration	01:00	
0x0018	0x2900	Characteristic Extended Properties		
0x0019	0x2906	Valid Range		
<b>Characteristic Declaration: True Wind Speed</b>				
0x001A	0x2803	Characteristic Declaration	92:1B:00:70:2A	
0x001B	0x2A70	True Wind Speed	28:23	0x92
0x001C	0x290C	Environmental Sensing Measurement		
0x001D	0x290D	Environmental Sensing Trigger Setting		
0x001E	0x2901	Characteristic User Description		
0x001F	0x2902	Client Characteristic Configuration	01:00	
0x0020	0x2900	Characteristic Extended Properties		
0x0021	0x2906	Valid Range		
<b>Characteristic Declaration: Humidity</b>				
0x0022	0x2803	Characteristic Declaration	92:23:00:6F:2A	
0x0023	0x2A6F	Humidity	58:07	0x92
0x0024	0x290C	Environmental Sensing Measurement		
0x0025	0x290D	Environmental Sensing Trigger Setting		

The right panel shows the 'Attribute Details' for the selected characteristic. The 'Handle' is 0x000C, the 'UUID' is 0x2800, and the 'UUID Description' is 'Primary Service Declaration'. The 'Value' is 1A:18. The 'Read Value' and 'Write Value' buttons are visible.

The bottom panel shows the 'Log' with the following entries:

```

[18:25:33.962] : Attribute Handle: 0x0023
[18:25:33.962] : Value: [58:07]

```

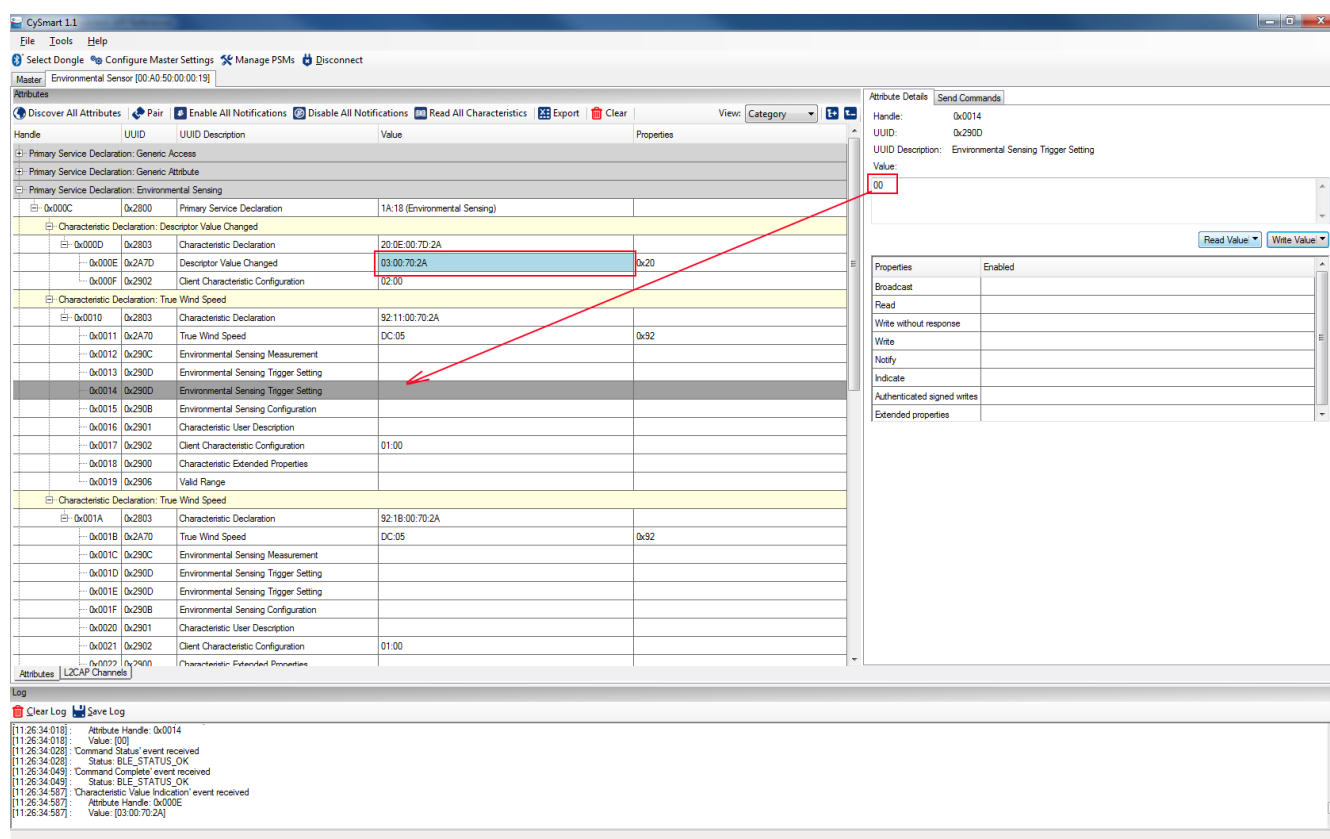
- Write a human readable name for a Characteristic to **Characteristic User Description Descriptor (UUID 0x2901)**, press the **Write Value** button the press **Read Value** and observe the Descriptor Value Changed Characteristic indication with regard to that. Before writing the descriptor value should be converted to the ASCII numbers first. On the **Figure 9** the name "Aerometer#1" - 41:65:72:6F:6D:65:74:65:72:23:31 (ASCII) is used.

Figure 9. CySmart Windows app: Writing to Characteristic User Description Descriptor

The screenshot displays the CySmart 1.1 application interface. The main window shows a list of attributes under the 'Attributes' tab. The 'Characteristic Declaration: Descriptor Value Changed' section is expanded, showing a table of descriptors. The 'Characteristic Declaration: True Wind Speed' section is also expanded. The 'Characteristic User Description' section is highlighted, showing a table with columns for Handle, UUID, and Value. The 'Value' column for the 'Characteristic User Description' entry is highlighted with a red box, showing the value '11:00:70:2A'. The 'Properties' tab on the right shows the 'Characteristic User Description' properties, including 'Handle: 0x0016', 'UUID: 0x2901', and 'Value: 41 65 72 6F 6D 65 74 65 72 23 31'. The 'Write Value' button is visible.

- Write the **0x00** value to the second **ES Trigger Settings Descriptor (UUID 0x290D)**, press **Write Value** button and observe the Descriptor Value Changed Characteristic indication with regard to that:

Figure 10. CySmart Windows app: Writing to ES Trigger Descriptor



The details about the Environmental Sensing Service characteristic data structures are in the [ESS Specification](#).

If you have problems with usage of the CySmart app, please, refer to [CySmart User Guide](#).

© 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.