PSoC 4 BLE – Pressure Sensor

Objective

This example demonstrates capability of PSoC® 4 BLE to communicate with an I2C digital sensor and a BLE Central device.

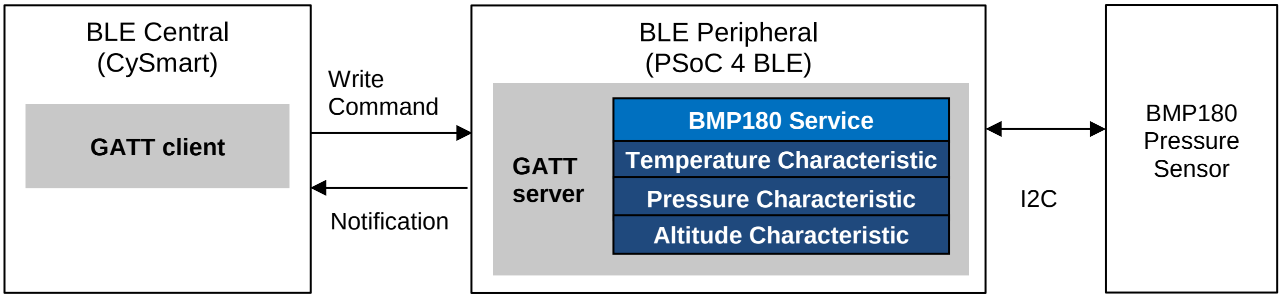
Overview

This example uses the BLE Pioneer Kit to communicate with a BMP180 Pressure Sensor and a BLE Central device. The data communicated over BLE is the Temperature, Barometric Pressure and Altitude values acquired from the sensor. This project uses an I2C component to acquire the data from the sensor and a BLE component to report it to the Central device. The sensor data is sent to the Central device as Notifications using a Custom Service. Additionally, the sensor data can also be monitored graphically in real time using Micrium® µC/Probe™ application.

The BLE profile in this example consists of a BLE custom service, named **BMP180**. The **BMP180** service consists of three custom characteristics, namely **Temperature**, **Pressure** and **Altitude**. These characteristics are used for sending 4-byte sensor data as notification to the connected GATT client device whenever new data is available.

For detailed information on the BMP180 Pressure Sensor and its operation, please refer to the BMP180 datasheet.

Figure 1: PSoC 4 BLE Pressure Sensor Application



Requirements

Design Tool: PSoC Creator 4.2, CySmart 1.2 SP1 and µC/Probe 3.5.15 (optional)

Programming Language: C (GCC 5.4.1 – included with PSoC Creator)

Associated Devices: All PSoC 4 BLE devices

Required Hardware: CY8CKIT-042-BLE-A Bluetooth® Low Energy (BLE) Pioneer Kit and BMP180 Pressure Sensor

Hardware Setup

The BLE Pioneer Kit has all the necessary hardware required for this lab, except for BMP180 Pressure Sensor which is not included in the kit and can be purchased from Adafruit®.

In this setup, the sensor is connected to outer row of the J2 Arduino™ compatible header on the BLE Pioneer Kit. The sensor communicates and sources the power using the PSoC 4 BLE pins available in the J2 header. The User Button which is used for waking up the PSoC 4 BLE device/to start the advertisement and the red LED which indicates advertising state are hardwired on the BLE Pioneer Kit.

Note that, this example project can also be used with CY8CKIT-042-BLE by changing the target device to 'CY8C4247LQI-BL483' from the Device Selector window in PSoC Creator. Refer to the PSoC Creator Help document for details on how to change the target device for a project.

Figure 2: Block Diagram

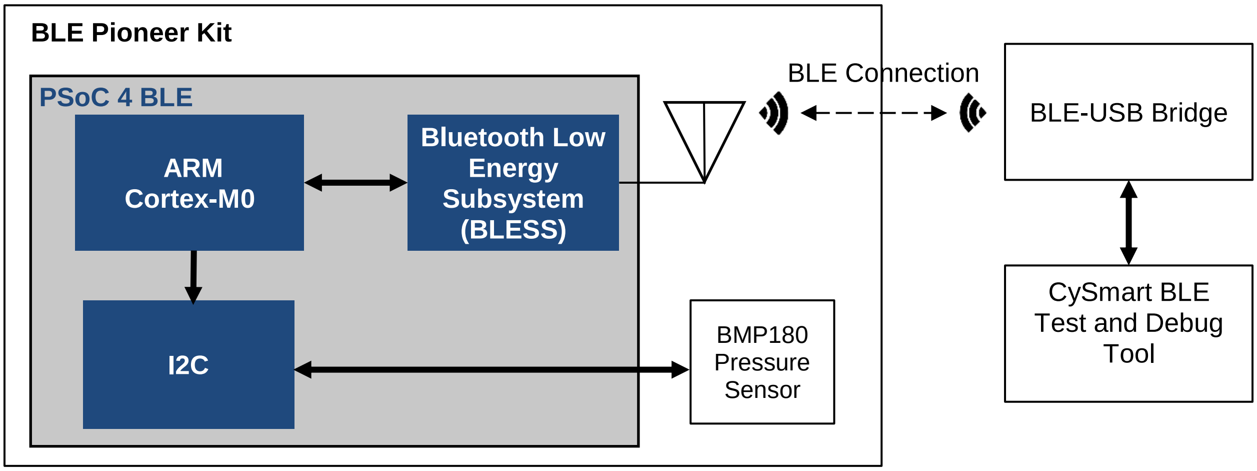
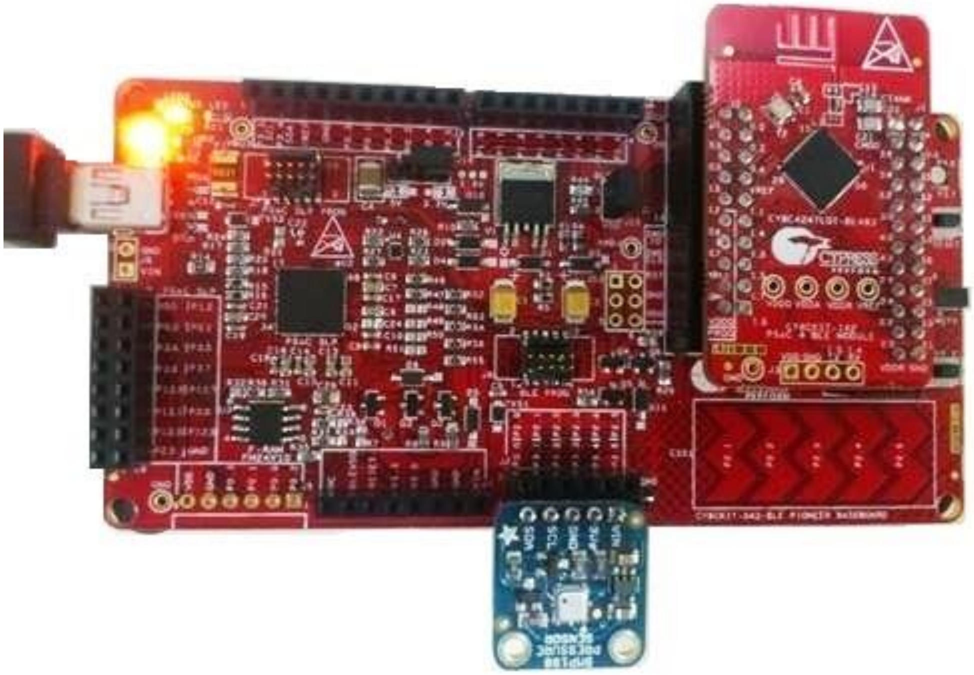
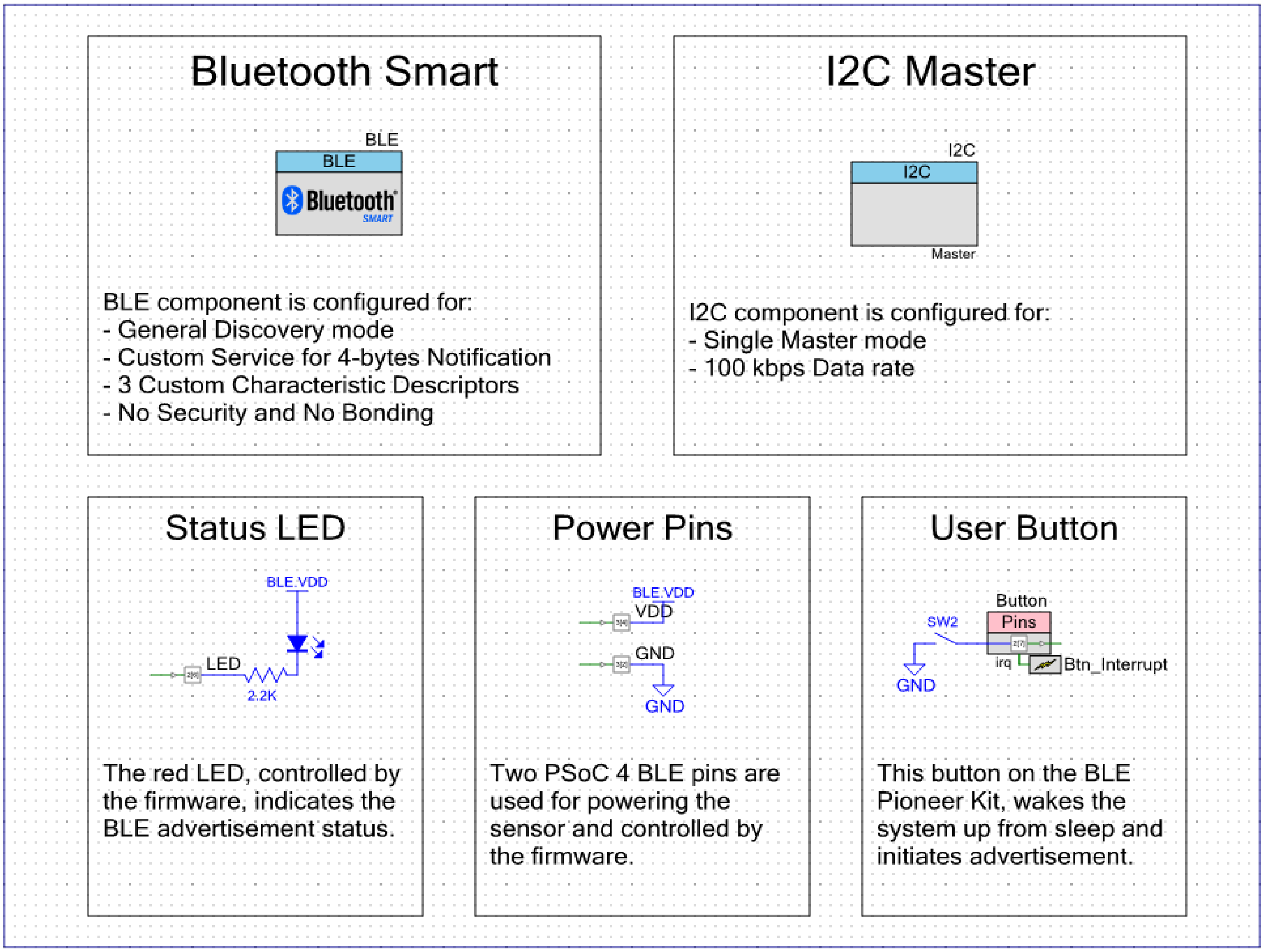


Figure 3: Kit Connection



PSoC Creator Schematic

Figure 4: PSoC Creator Schematic



BLE component is configured for:
- General Discovery mode
- Custom Service for 4-bytes Notification
- 3 Custom Characteristic Descriptors
- No Security and No Bonding

I2C component is configured for:
- Single Master mode
- 100 kbps Data rate

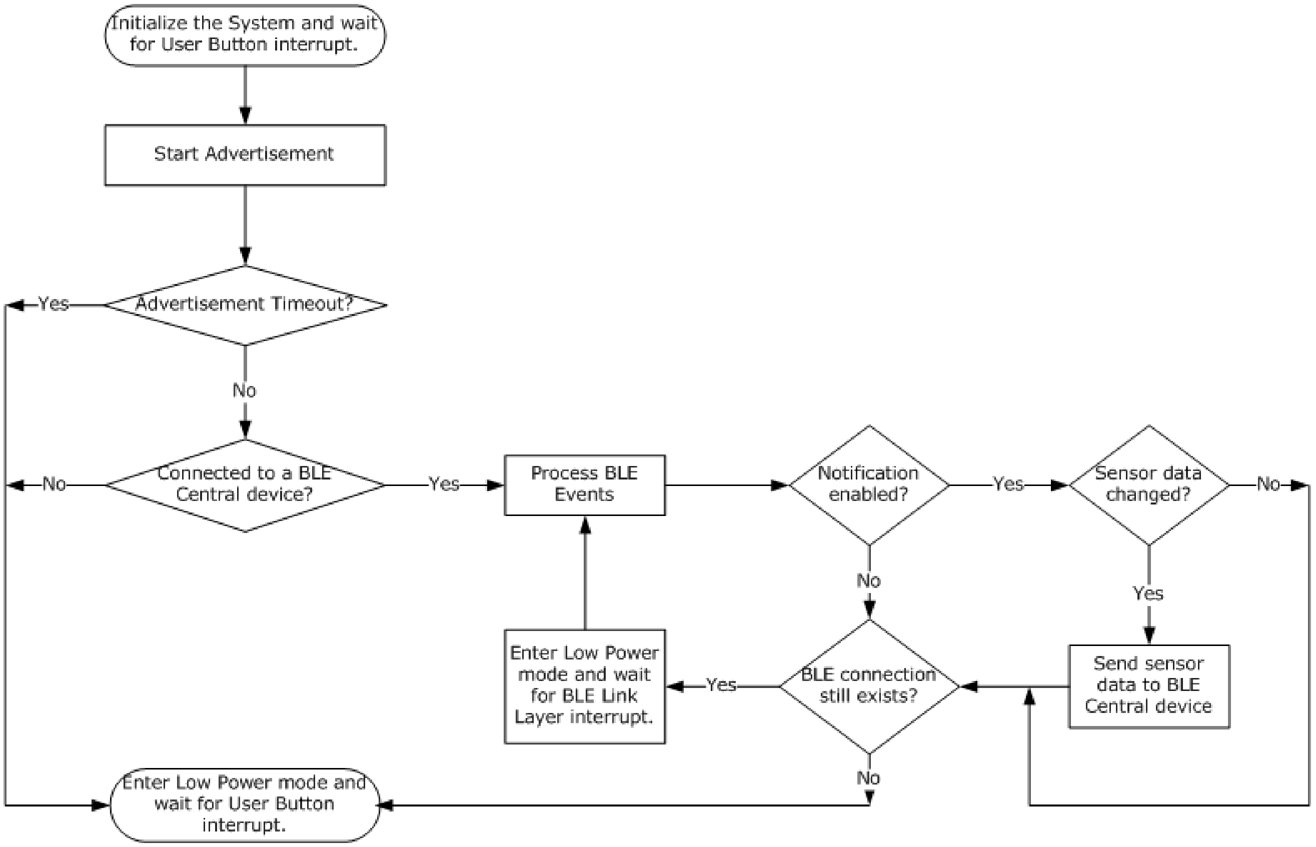
The red LED, controlled by the firmware, indicates the BLE advertisement status.

Two PSoC 4 BLE pins are used for powering the sensor and controlled by the firmware.

This button on the BLE Pioneer Kit, wakes the system up from sleep and initiates advertisement.

Firmware Flow

Figure 5: Firmware Flow


Initialize the System and wait for User Button interrupt.
Start Advertisement
Advertisement Timeout?
Yes
No
Connected to a BLE Central device?
No
Yes
Process BLE Events
Notification enabled?
Yes
No
Sensor data changed?
No
Yes
Send sensor data to BLE Central device
BLE connection still exists?
Yes
No
Enter Low Power mode and wait for BLE Link Layer interrupt.
Enter Low Power mode and wait for User Button interrupt.

This example project contains the following source files:

- **main.c**: This file contains the main function, which is the entry point and execution of the firmware application. It also contains function definitions for initializing the system and interrupts, reading the sensor data, processing the notification, handling the low power mode operation and handling the status LED.

  The low power mode function is continuously called in the main loop and is responsible for pushing the BLE subsystem (BLESS) as well as the CPU to Deep Sleep mode as much as possible. The wakeup source is either the BLE subsystem Link Layer internal timer or the interrupt from the User Button press. This allows for low power mode implementation.

- **BLEApplication.c**: This file contains function definitions related to BLE communication and operation. It contains the event callback function definition that is registered with the BLE component startup and used by the component to send various BLE events from the BLE stack to the application layer for processing. It contains functions to send the notifications to the GATT client device and for updating the BLE Connection Parameters, which is important for low power mode operation.

- **I2CMaster.c**: This file contains function definitions related to BLE communication and sensor operation. It contains functions for reading the calibration data, uncompensated temperature and pressure values from the sensor. It contains algorithm to calculate the true temperature, barometric pressure and the altitude value.

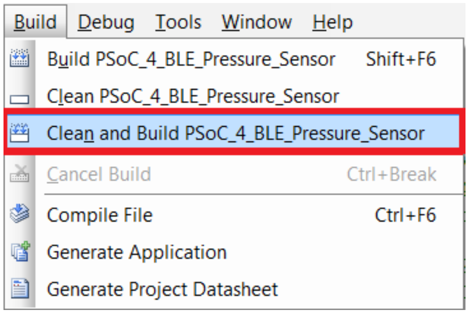
Build and Program

This section shows how to build the project and program the PSoC 4 BLE device. If you are using a development kit with a built-in programmer (BLE Pioneer Kit, for example), connect the BLE Pioneer Baseboard to your computer using the USB Standard-A to Mini-B cable. For other kits, refer to the kit user guide.

If you are developing on your own hardware, you need a hardware debugger, for example, a Cypress CY8CKIT-002 MiniProg3.

1. Launch PSoC Creator and open the *PSoC_4_BLE_Pressure_Sensor* project.
2. On PSoC Creator, select **Build** > **Clean and Build PSoC_4_BLE_Pressure_Sensor**, as shown in Figure 6.

Figure 6: Build Project

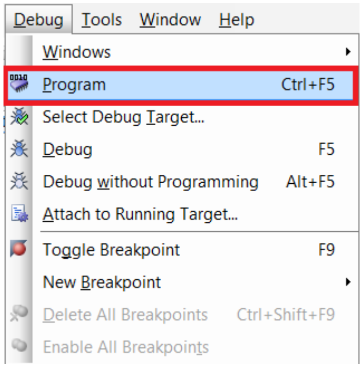

Build Debug Tools Window Help
Build PSoC_4_BLE_Pressure_Sensor Shift+F6
Clean PSoC_4_BLE_Pressure_Sensor
Clean and Build PSoC_4_BLE_Pressure_Sensor
Cancel Build Ctrl+Break
Compile File Ctrl+F6
Generate Application
Generate Project Datasheet

3. On a successful build, the total Flash and SRAM usage is reported in the Output window.
4. Select **Debug** > **Select Debug Target**. In the **Select Debug Target** window, click **Port Acquire**, and then click **Connect**. Click **OK** to close the dialog box.

   If you are using your own hardware, make sure the **Port Setting** configuration under **Select Debug Target** dialog box for your programming hardware is configured as per your setup.

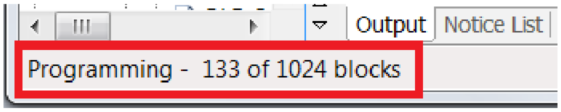
5. Select **Debug** > **Program** to program the device with the project, as shown in Figure 7.

Figure 7: Programming the Device



You can view the programming status on the PSoC Creator status bar (bottom-left corner of the PSoC Creator window), as shown in Figure 8.

Figure 8: Programming Status



Programming - 133 of 1024 blocks

Testing

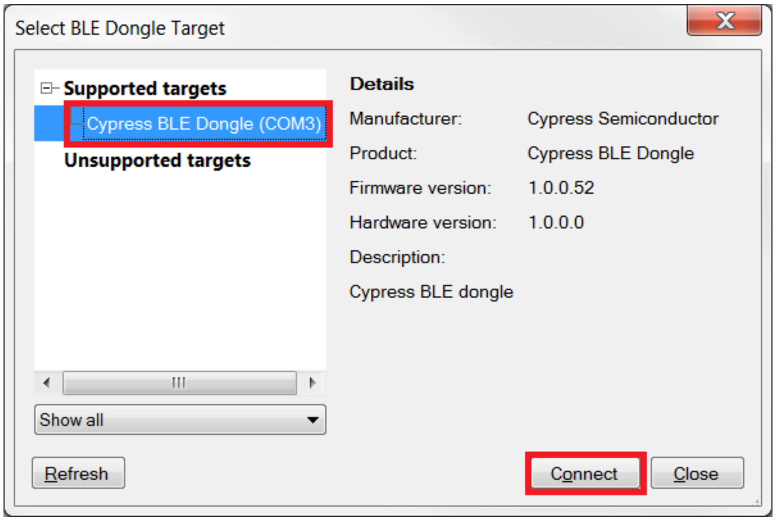
After programming the device successfully by following the steps in Build and Program, press the User Button (**SW2**) on the BLE Pioneer Kit to start the advertisement. Advertisement is indicated by a blinking red LED on the BLE Pioneer Kit.

Note: This example has an advertisement timeout of 30 seconds after which it returns to the low power mode. Press **SW2** again to restart the advertisement.

Testing with the CySmart BLE Test and Debug Utility for Windows PC:

1. Plug the BLE-USB Bridge (included with the BLE Pioneer Kit) in your computer's USB port. Make sure the sensor is connected to BLE Pioneer Baseboard as described in Hardware Setup.
2. On your computer, launch **CySmart 1.0**. It is located in the **All Programs -> Cypress -> CySmart 1.0** folder in the Windows Start menu. The tool opens up and asks you to **Select BLE Dongle Target**. Select the **Cypress BLE Dongle (COMxx)** and click **Connect**, as shown in Figure 9.

Figure 9: Select BLE Dongle Target



Details

Manufacturer: Cypress Semiconductor

Product: Cypress BLE Dongle

Firmware version: 1.0.0.52

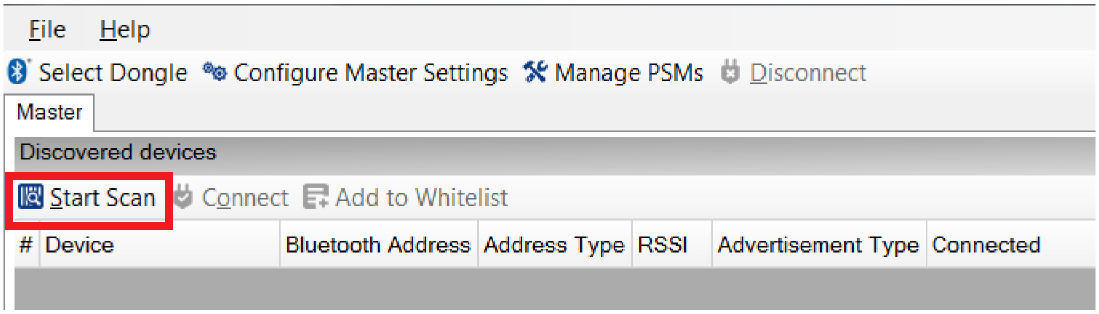
Hardware version: 1.0.0.0

Description:

Cypress BLE dongle

3. When the BLE-USB Bridge is connected, click on **Start Scan** to find your BLE device, as shown in Figure 10.

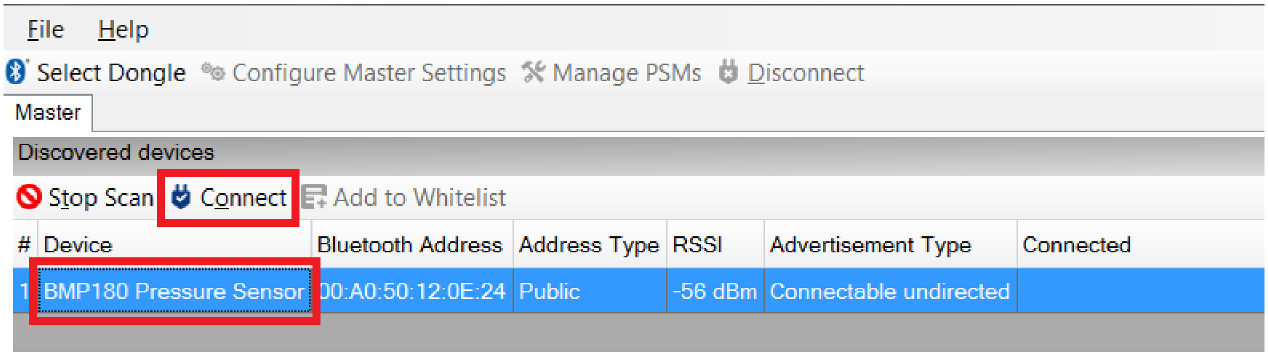
Figure 10: Finding a BLE Device



The tool lists all the nearby devices in the Discovered devices section.

4. Click on **BMP180 Pressure Sensor** and then click **Connect** to connect to the device, as shown in Figure 11.

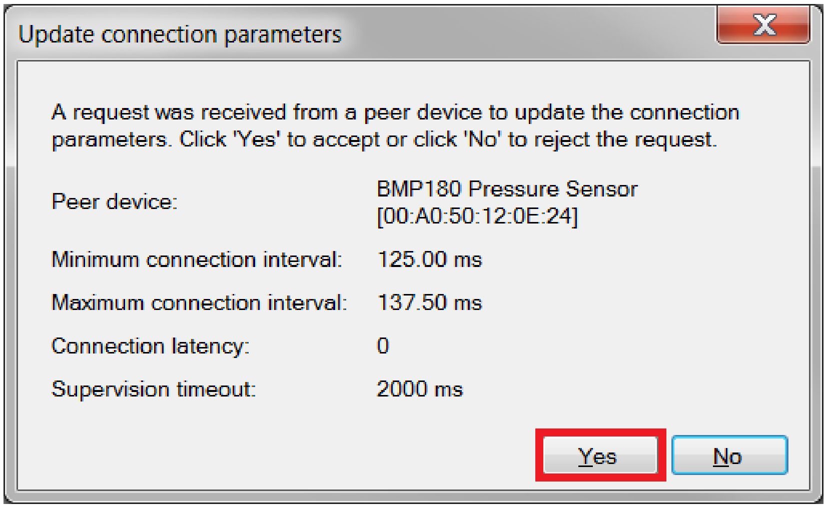
Figure 11: Connect to Peripheral



#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected
1	BMP180 Pressure Sensor	00:A0:50:12:0E:24	Public	-56 dBm	Connectable undirected	

5. When connected, the tool will display a message for the **Update connection parameters**. Select **Yes**, as shown in Figure 12.

Figure 12: Update Connection Parameter

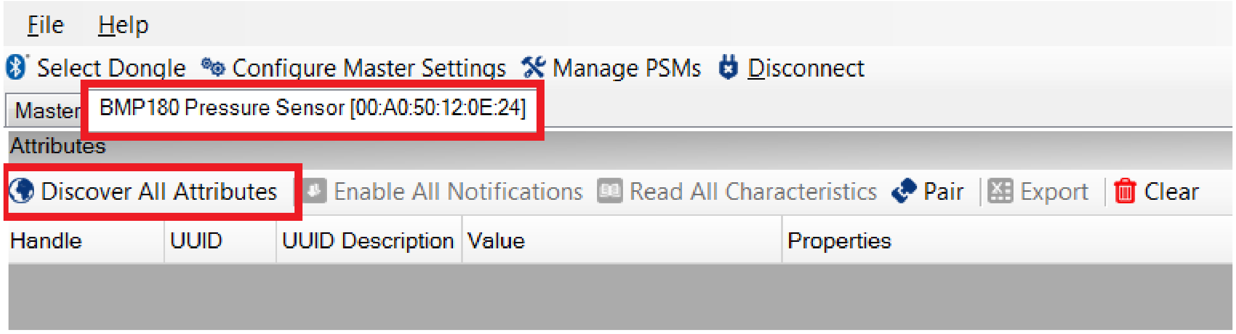


A request was received from a peer device to update the connection parameters. Click 'Yes' to accept or click 'No' to reject the request.

Peer device:	BMP180 Pressure Sensor [00:A0:50:12:0E:24]
Minimum connection interval:	125.00 ms
Maximum connection interval:	137.50 ms
Connection latency:	0
Supervision timeout:	2000 ms

6. The tool will now open a separate tab for the device. Click **Discover All Attributes** to list all the Attributes in the device, with their respective UUIDs and descriptions, as shown in Figure 13.

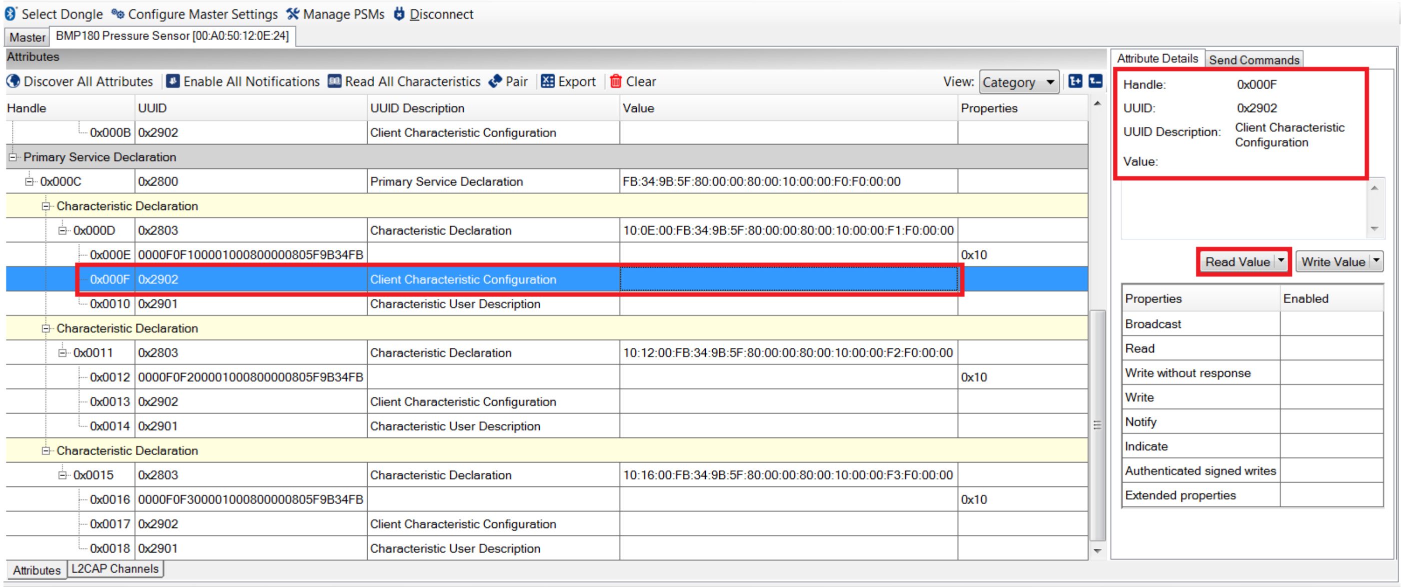
Figure 13: Discovering Attributes of a Connected BLE Device



Handle	UUID	UUID Description	Value	Properties

7. Locate the **Client Characteristic Configuration** Attribute for the **Temperature** characteristic (Handle 0x000F). Click **Read Value** to read the existing Client Characteristic Configuration Descriptor (CCCD) value, as shown in Figure 14.

Figure 14: Read Attribute Value



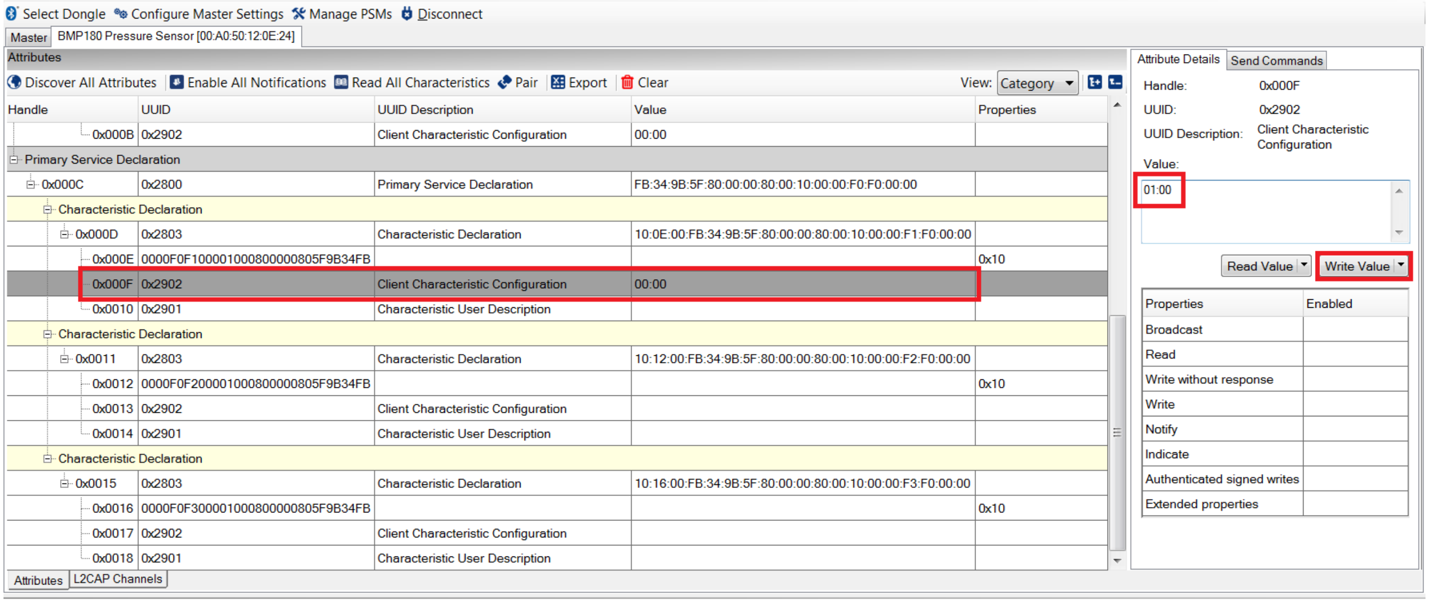
Handle	UUID	UUID Description	Value	Properties
0x000B	0x2902	Client Characteristic Configuration		
Primary Service Declaration				
0x000C	0x2800	Primary Service Declaration	FB:34:9B:5F:80:00:00:80:00:10:00:00:F0:F0:00:00	
Characteristic Declaration				
0x000D	0x2803	Characteristic Declaration	10:0E:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F1:F0:00:00	
0x000E	0000F0F100001000800000805F9B34FB			0x10
0x000F	0x2902	Client Characteristic Configuration		
0x0010	0x2901	Characteristic User Description		
Characteristic Declaration				
0x0011	0x2803	Characteristic Declaration	10:12:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F2:F0:00:00	
0x0012	0000F0F200001000800000805F9B34FB			0x10
0x0013	0x2902	Client Characteristic Configuration		
0x0014	0x2901	Characteristic User Description		
Characteristic Declaration				
0x0015	0x2803	Characteristic Declaration	10:16:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F3:F0:00:00	
0x0016	0000F0F300001000800000805F9B34FB			0x10
0x0017	0x2902	Client Characteristic Configuration		
0x0018	0x2901	Characteristic User Description		

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

8. Change the **Value** to **01:00** under **Attribute Details** tab and click **Write Value**, as shown in Figure 15. This enables the notifications for the **Temperature** characteristic.

Figure 15: Write Attribute Value



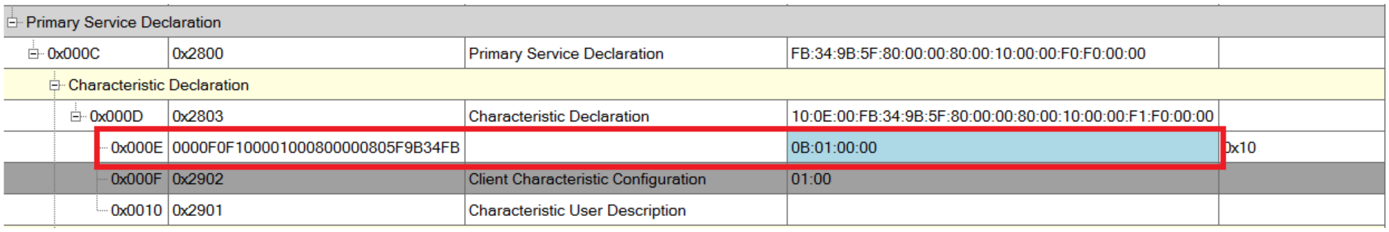
Handle	UUID	UUID Description	Value	Properties
0x000B	0x2902	Client Characteristic Configuration	00:00	
Primary Service Declaration				
0x000C	0x2800	Primary Service Declaration	FB:34:9B:5F:80:00:00:80:00:10:00:00:F0:F0:00:00	
Characteristic Declaration				
0x000D	0x2803	Characteristic Declaration	10:0E:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F1:F0:00:00	
0x000E	0000F0F100001000800000805F9B34FB			0x10
0x000F	0x2902	Client Characteristic Configuration	00:00	
0x0010	0x2901	Characteristic User Description		
Characteristic Declaration				
0x0011	0x2803	Characteristic Declaration	10:12:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F2:F0:00:00	
0x0012	0000F0F200001000800000805F9B34FB			0x10
0x0013	0x2902	Client Characteristic Configuration		
0x0014	0x2901	Characteristic User Description		
Characteristic Declaration				
0x0015	0x2803	Characteristic Declaration	10:16:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F3:F0:00:00	
0x0016	0000F0F300001000800000805F9B34FB			0x10
0x0017	0x2902	Client Characteristic Configuration		
0x0018	0x2901	Characteristic User Description		

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:
01:00

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

9. When the **Temperature** characteristic is enabled, the BLE Central device starts receiving the temperate data acquired from the sensor. The received data is in steps of 0.1°C and is displayed in the Value field for Handle 0x000E, as shown in Figure 16.

Figure 16: Notification Received



Primary Service Declaration				
0x000C	0x2800	Primary Service Declaration	FB:34:9B:5F:80:00:00:80:00:10:00:00:F0:F0:00:00	
Characteristic Declaration				
0x000D	0x2803	Characteristic Declaration	10:0E:00:FB:34:9B:5F:80:00:00:80:00:10:00:00:F1:F0:00:00	
0x000E	0000F0F100001000800000805F9B34FB		0B:01:00:00	0x10
0x000F	0x2902	Client Characteristic Configuration	01:00	
0x0010	0x2901	Characteristic User Description		

10. Locate the **Client Characteristic Configuration** Attribute for the **Pressure** characteristic (Handle 0x0013) and change the **Value** to **01:00** under **Attribute Details** tab and click **Write Value**, to enable the notifications for the **Pressure** characteristic.

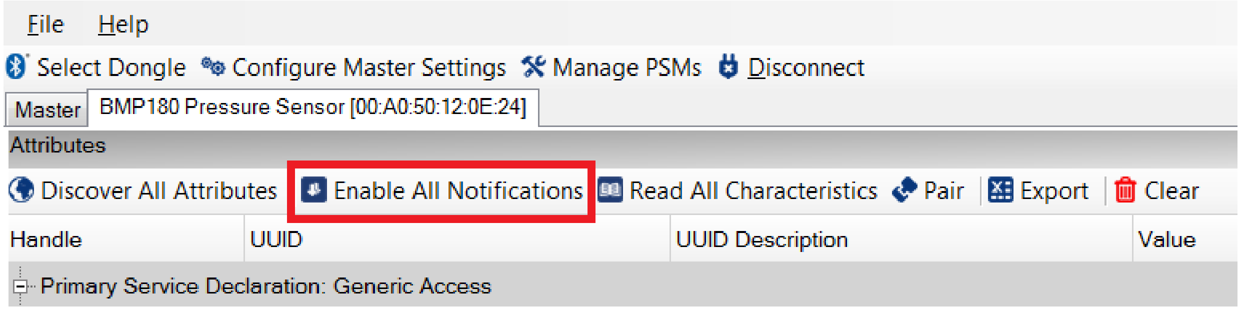
11. When the **Pressure** characteristic is enabled, the BLE Central device starts receiving the barometric pressure data acquired from the sensor. The received data is in steps of 0.01hPa and is displayed in the Value field for Handle 0x0012.

12. Locate the **Client Characteristic Configuration** Attribute for the **Altitude** characteristic (Handle 0x0017) and change the **Value** to **01:00** under **Attribute Details** tab and click **Write Value**, to enable the notifications for the **Altitude** characteristic.

13. When the **Altitude** characteristic is enabled, the BLE Central device starts receiving the altitude data acquired from the sensor. The received data is in steps of 1m and is displayed in the Value field for Handle 0x0016.

➢ Alternatively, all service notifications can be enabled by clicking on **Enable All Notifications**, as shown in Figure 17.

Figure 17: Enable All Notifications



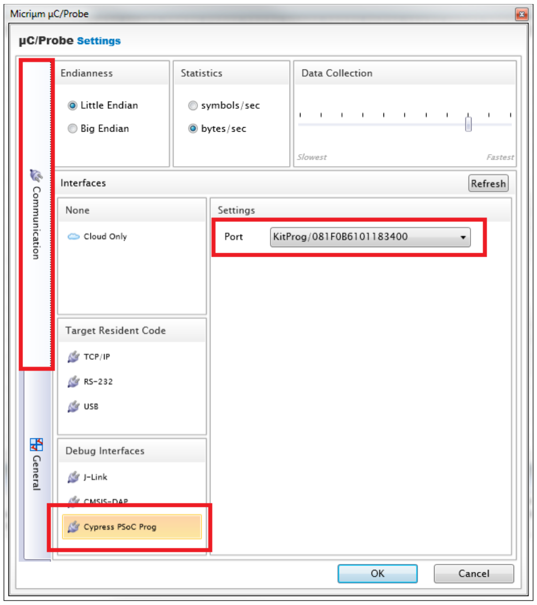
Testing with the µC/Probe application:

Once you are connected to a BLE Central device and enabled notifications as described in **Testing with the CySmart BLE Test and Debug Utility for Windows PC** section,

1. On your computer, launch **Micrium uC-Probe**. It is located in the **All Programs -> Micrium -> uC-Probe** folder in the Windows Start menu.
2. Click the **Settings** button from the toolbar and select the **Communication** tab in the µC/Probe Settings window.

3. Click on **Cypress PSoC Prog** under Debug Interface and select the **KitProg/<Serial_Number>** from the drop-down, as shown in Figure 18. Click **OK** to close the window.

Figure 18: µC/Probe Settings



Micriµm µC/Probe

µC/Probe Settings

Endianness: ● Little Endian ○ Big Endian

Statistics: ○ symbols/sec ● bytes/sec

Data Collection: Slowest – Fastest

Interfaces

None: Cloud Only

Settings: Port KitProg/081F0B6101183400

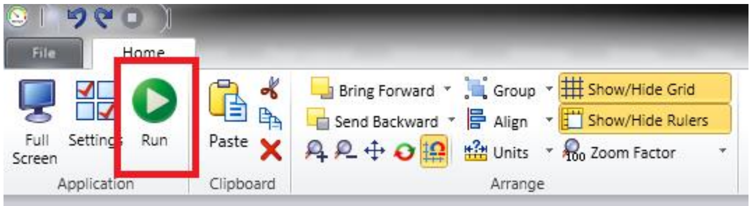
Target Resident Code: TCP/IP, RS-232, USB

Debug Interfaces: J-Link, CMSIS-DAP, Cypress PSoC Prog

4. Select **File > Open** menu, load the *PSoC_4_BLE_Pressure_Sensor.wspx* file included in the example project directory and click **Open** as shown in Figure.

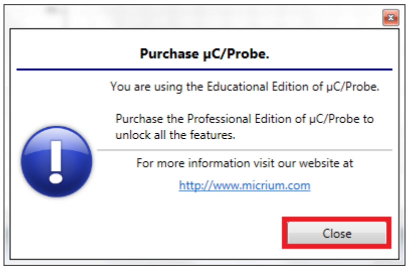
5. Click the **Run** button from the toolbar, as shown in Figure 19.

Figure 19: Run Button in µC/Probe



Note that if you are using the Educational Edition of the µC/Probe, a dialog appears reminding you to purchase the Professional Edition. Click **Close** on the dialog to continue, as shown in Figure 20.

Figure 20: Purchase µC/Probe



Purchase µC/Probe.

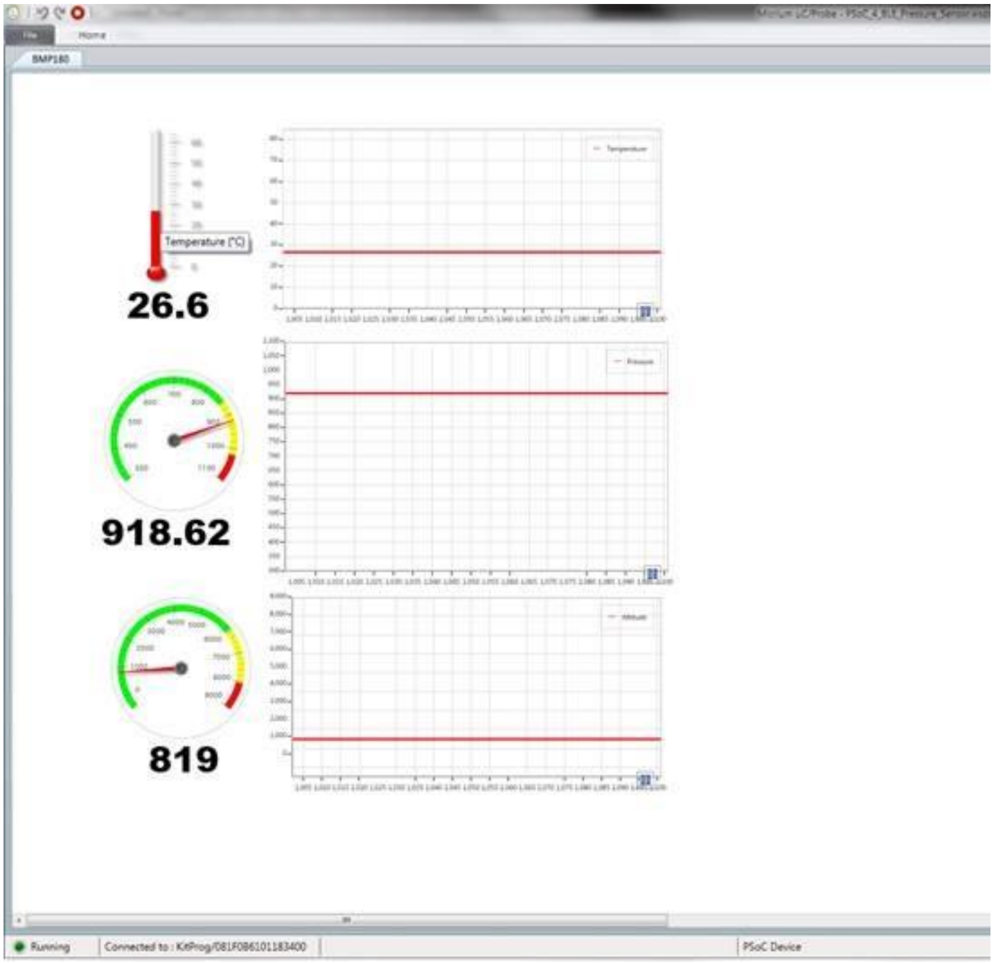
You are using the Educational Edition of µC/Probe.

Purchase the Professional Edition of µC/Probe to unlock all the features.

For more information visit our website at http://www.micrium.com

6. The µC/Probe application shows all the graphical elements configured in the workspace and each graphical element shows the corresponding data acquired from the sensor, as shown in Figure 21.

Figure 21: µC/Probe GUI Elements



Related Documents

Table 1 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Table 1: Related Documents

Document	Title	Comment
AN91267	Getting Started with PSoC 4 BLE	Provides an introduction to PSoC 4 BLE device that integrates a Bluetooth Low Energy radio system along with programmable analog and digital resources.
AN91445	Antenna Design Guide	Provides guidelines on how to design an antenna for BLE applications.