PSoC4 BLE – UART-BLE

Objective

This example demonstrates how to use PSoC4 BLE device as a UART-BLE bridge.

Overview:

This code example uses a custom BLE profile to demonstrate the UART-BLE bridge functionality. In order to emulate both peer devices as UART COM ports, this example essentially consists of two projects – Central and Peripheral. When appropriately interfaced to a PC, each of the peer devices can be used as a COM port using a terminal application (like TeraTerm). The data sent through the terminal application at one end, appears at the other end, as it does over wired UART communication. .

Requirements:

Programming Language	**:** C (GCC 4.8.4)
Associated Parts	**:** All PSoC4 BLE parts
Required software	**:** PSoC Creator 4.0, Terminal application (like TeraTerm)
Required hardware	**:** CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit
Optional hardware	**:** UART transceiver with flow control (if flow control is needed)

Hardware Setup:

Since BLE Pioneer Kit enumerates as a USB-UART, it can be directly be used for interfacing the module to a UART terminal application. However, if flow control or a baud rate of greater than 115200 is needed, an external UART transceiver can be used by appropriately connecting UART pins.

Building and Programming the device:

This section shows how to build the project and program PSoC4 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. Open the project ***"UART_to_BLE_central"*** using PSoC Creator.
2. Open the "***TopDesign.cysch***" file from the workspace explorer on left hand side of the screen and double click on the UART component to open the UART configuration window.
3. Configure the UART baud rate as per your application requirements. The default configuration is as shown in figure.

   Note: It should be noted that the USB UART on pioneer kit supports a max. baud rate of 115200bps.

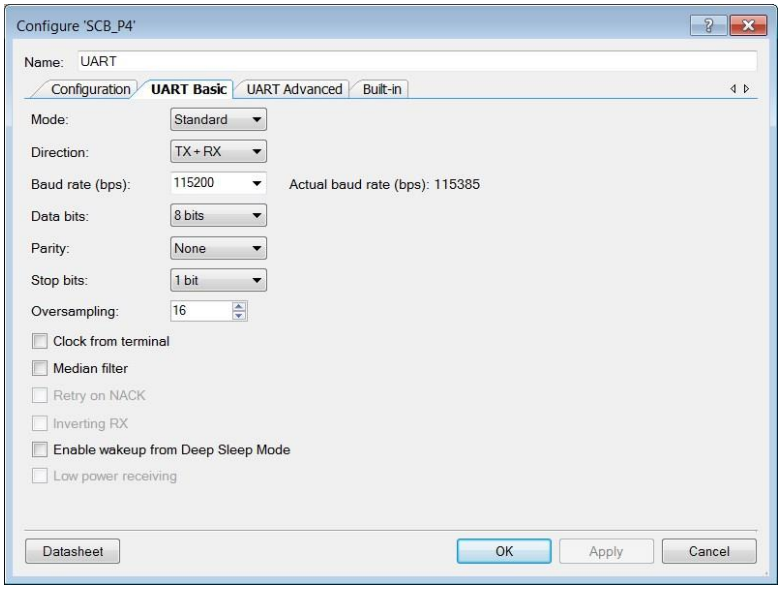


Figure 1: Configuration window of UART component

4. UART flow control can be enabled or disabled by checking/upchucking the RTS/CTS boxes in the “UART Advanced” tab of the configuration window. By default, both RTS and CTS are enabled.

   Note: If flow control is enabled, it should be made sure that the hardware being used supports flow control. The USB UART on pioneer kit does NOT support Flow control.

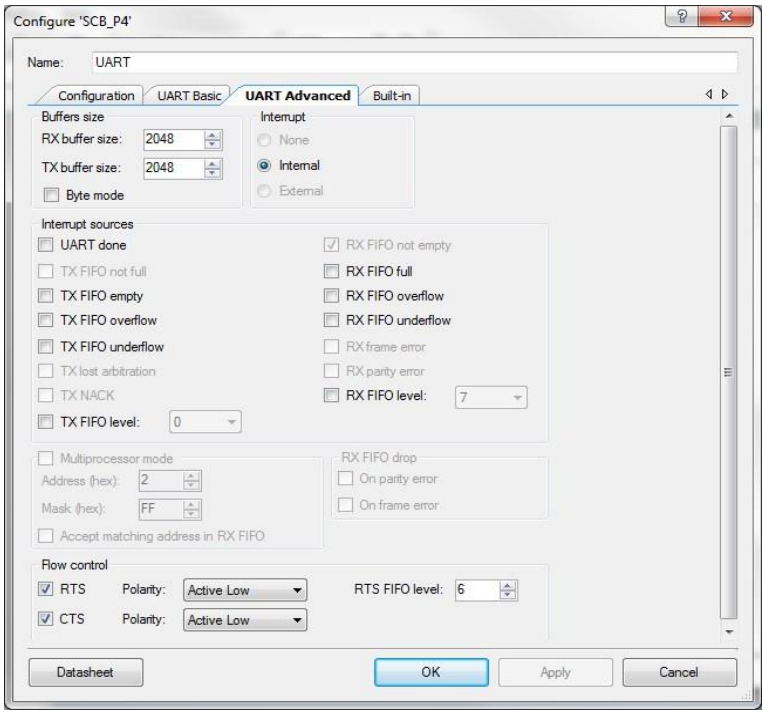


Figure 2: Advanced tab in configuration window of UART component

5. If RTS/CTS are enabled, make sure to uncomment the macro "***FLOW_CONTROL***" in ***main.h***. Other macros i.e. "***PRINT_MESSAGE_LOG***" and "***LOW_POWER***" as shown in figure can also be commented/uncommented based on the application requirements.

```
/***************************************
*   Conditional compilation parameters
***************************************/
//    #define     FLOW_CONTROL
#define     PRINT_MESSAGE_LOG
//    #define     LOW_POWER_MODE
```

Figure 3: Configuration window of UART component

6. If a custom hardware is being used, make sure to change the pin mapping in the corresponding "***.cydwr***" file of your project. The default mapping is for an external UART transceiver interfaced to Pioneer kit.

Alias	Name	Port	Pin	Lock
	\UART:rx\	P1[4] OA3:vminus, TCPWM2:line_out, SCB0:uart_rx, SCB0:i2c_sda, SCB0:spi_mosi	32	☑
	\UART:tx\	P1[5] OA3:vplus, TCPWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi_miso	33	☑
	Conn_LED	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart_cts, SRSS:ext_clk_lf	54	☑
	Scan_LED	P2[6] OA0:vplus_alt	43	☑

Figure 4: Pin mapping in *.cydwr* file

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

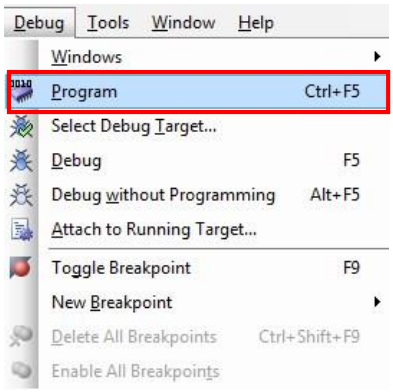


Figure 5: Programming device

8. The status bar (lower-left corner of the window) displays the programming status, as shown in figure.

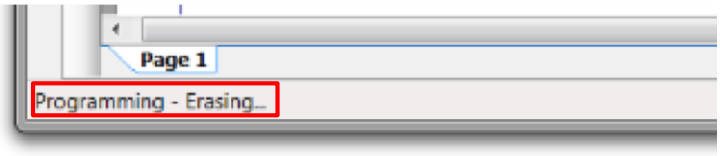


Figure 6: Programming status

9. Repeat steps (1) to (8) for the project ***"UART_to_BLE_peripheral"*** for the **BLE-USB Bridge.**

Testing and Expected Results:

1. After programming, plug the BLE Pioneer Kit with PSoC 4 BLE module to your computer's USB port. As soon as the device gets powered the RED LED should start blinking (for central role), indicating BLE SCANNING

2. Open a PC application (like TeraTerm) and configure the desired COM port (in terms of baud rate, flow control, etc.) as per configuration used in the project.

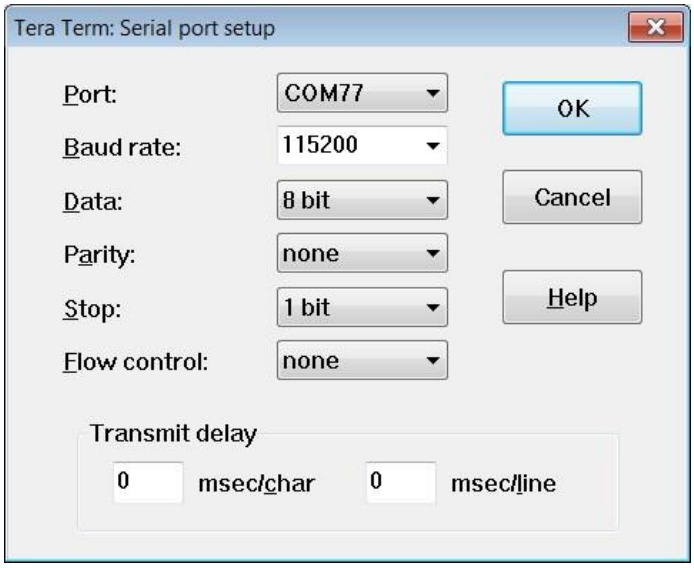


Figure 7: COM port configuration

3. If the macro "***PRINT_MESSAGE_LOG***" is uncommented in "***main.h***" of the PSoC Creator project, the configuration information should appear as shown in figure.

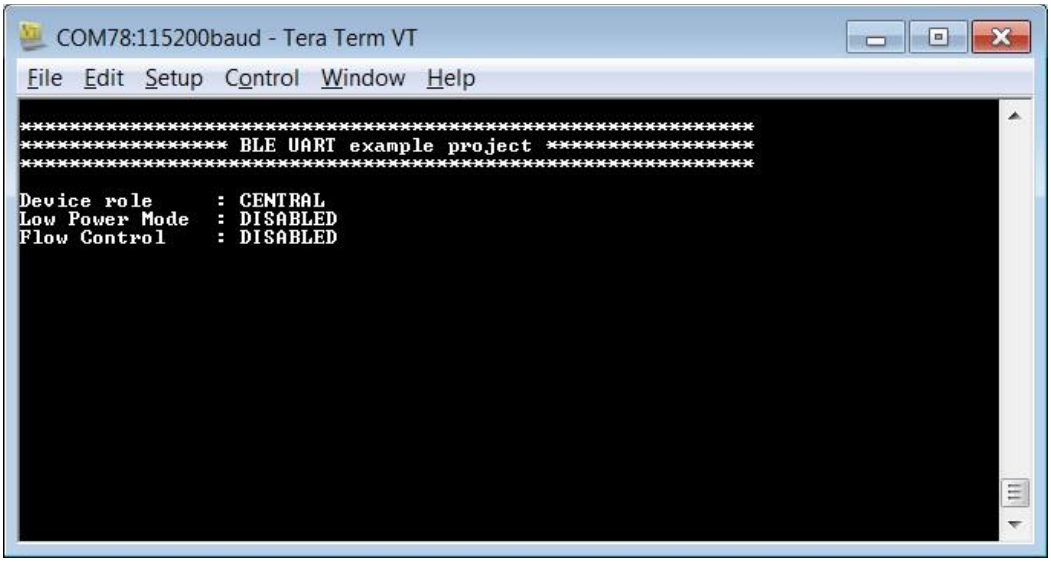


Figure 8: Project configuration information with *PRINT_MESSAGE_LOG* macro

Otherwise a blank screen should appear.

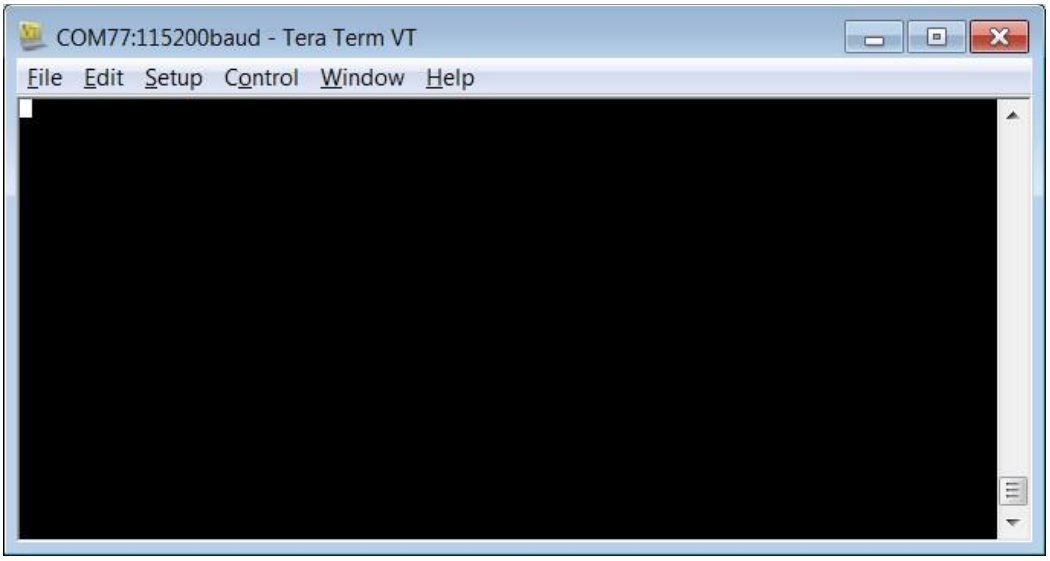


Figure 9: Project configuration information without *PRINT_MESSAGE_LOG* macro

4. Repeat steps 1 to 3 for BLE-USB Bridge.

5. If both the devices are in range of each other, they should automatically connect and BLUE LED on the Central should start blinking once the connection is established. The application corresponding to both should display messages, as shown in the following figures:

Figure 10: Message log for *UART_to_BLE_central* project

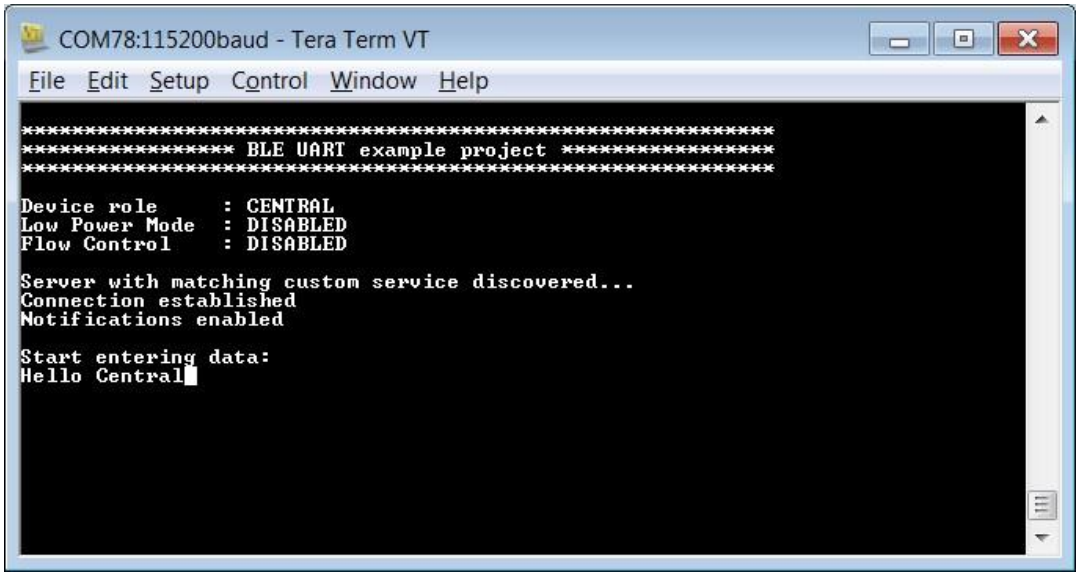
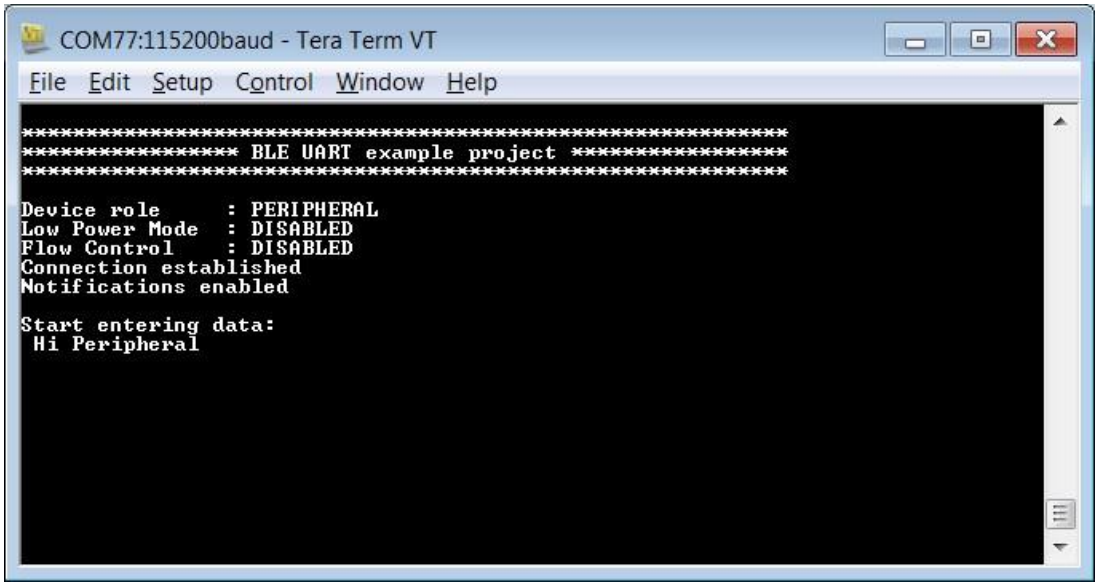


Figure 11: Message log for *UART_to_BLE_peripheral* project



6. The data entered on either side should appear on the other side.

Measuring the throughput:

1. For measuring the throughput, go to **File** > **Send file…**, browse for a .txt file and select open.

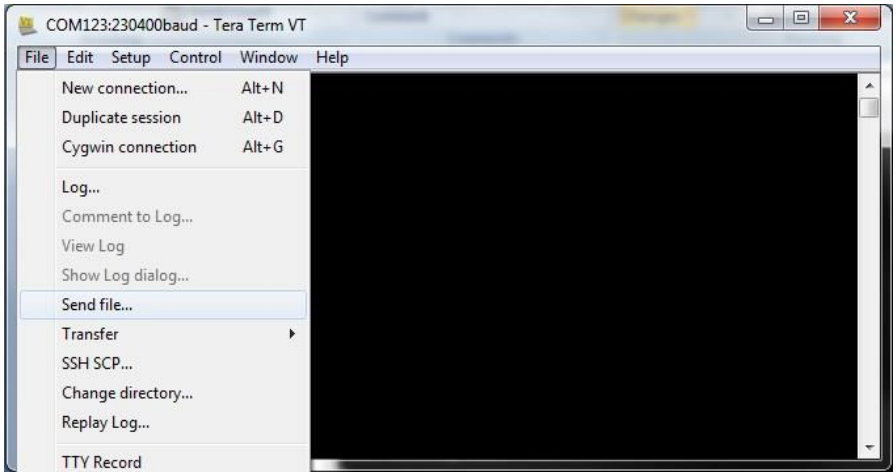


Figure 12: 'Send file…' option in TeraTerm

2. While the file is being sent, a dialog box (similar to below figure) should appear which displays the throughput.

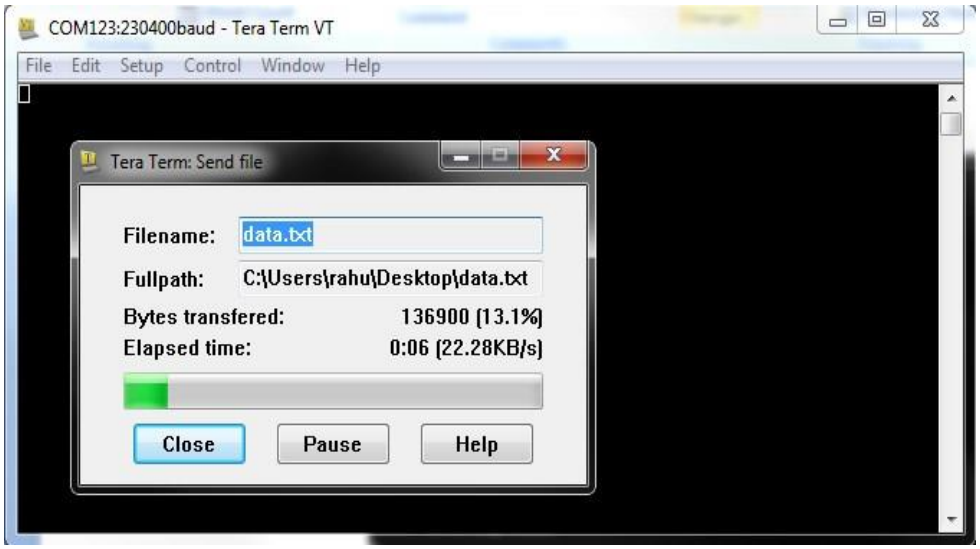


Figure 13: Throughput display in TeraTerm

3. If data integrity is to be verified, the TeraTerm log on the receiver end can be copied to a .txt file, which can be compared to the source .txt file using a compare tool (like Notepad++, etc.).

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 PSoC4 BLE	Provides an introduction to PSoC4 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.