



MICROCHIP

**RN4870/71 PICtail™/PICtail
Plus Daughter Board
User's Guide**

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
= ISO/TS 16949 =**

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KEELOQ, KEELOQ logo, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2016, Microchip Technology Incorporated, All Rights Reserved.
ISBN: 978-1-5224-1246-5

Object of Declaration: RN4870/71 PICtail™/PICtail Plus Daughter Board

EU Declaration of Conformity

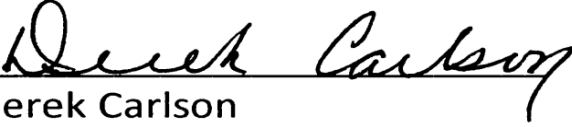
This declaration of conformity is issued by the manufacturer.

The development/evaluation tool is designed to be used for research and development in a laboratory environment. This development/evaluation tool is not a Finished Appliance, nor is it intended for incorporation into Finished Appliances that are made commercially available as single functional units to end users under EU EMC Directive 2004/108/EC and as supported by the European Commission's Guide for the EMC Directive 2004/108/EC (8th February 2010).

This development/evaluation tool complies with EU RoHS2 Directive 2011/65/EU.

This development/evaluation tool, when incorporating wireless and radio-telecom functionality, is in compliance with the essential requirement and other relevant provisions of the R&TTE Directive 1999/5/EC and the FCC rules as stated in the declaration of conformity provided in the module datasheet and the module product page available at www.microchip.com. For information regarding the exclusive, limited warranties applicable to Microchip products, please see Microchip's standard terms and conditions of sale, which are printed on our sales documentation and available at www.microchip.com.

Signed for and on behalf of Microchip Technology Inc. at Chandler, Arizona, USA.


Derek Carlson

VP Development Tools

11-NOV-16
Date

RN4870/71 PICtail™/PICtail Plus Daughter Board User's Guide

NOTES:



RN4870/71 PICtail™/PICtail PLUS DAUGHTER BOARD USER'S GUIDE

Table of Contents

Preface	7
Chapter 1. Overview	
1.1 Introduction	13
1.2 RN4870/71 PICtail™/PICtail Plus Daughter Board Description	13
1.3 Features	14
Chapter 2. Interface Description	
2.1 Introduction	15
2.2 RN4870 PICtail/PICtail Plus Daughter Board	15
2.3 RN4871 PICtail/PICtail Plus Daughter Board	17
2.4 Sensor Board	18
Chapter 3. Quick Start Guide	
3.1 Overview	19
3.2 Connecting the RN4870/71 PICtail/PICtail Plus Daughter Board to a Host PC	19
3.3 Changing Settings Using ASCII Commands	21
3.4 Connecting to the RN4870/71 Using SmartDiscover App	22
3.5 Creating Custom GATT Services	24
3.6 Accessing GATT Service Using UART Commands and SmartDiscover App	25
3.7 Reading Sensor Board Peripheral IO Ports Using UART Commands	27
Chapter 4. RN4870 Sensor Board	
4.1 Overview	29
4.2 Configuring the RN4870 Module Settings	29
4.3 Sensor Board GATT Service	30
4.4 Transferring Sensor Data into GATT Service Using Scripting	32
4.5 BLESensorApp Smartphone App	34
4.6 Provision Utility	36
Chapter 5. PIC Configuration Library	
5.1 Overview	39
5.2 Using BM7x Configuration Library with PICtail Board	39
Appendix A. Updating PICtail Firmware	
A.1 Overview	43
Appendix B. Schematics and BOM	
B.1 Introduction	45
B.2 RN4870 PICtail Schematic	46
B.3 RN4870 PICtail BOM	47

RN4870/71 PICtail™/PICtail Plus Daughter Board User's Guide

B.4 RN4871 PICtail Schematic	49
B.5 RN4871 PICtail BOM	50
B.6 RN4870 Sensor Board Schematic	52
B.7 RN4870 Sensor Board BOM	52
Appendix C. Bluetooth Low Energy Primer	
C.1 GAP Roles: Peripheral and Central	53
C.2 GATT Service: Client and Server	53
Appendix D. Sensor Board Configuration Command Text	
D.1 Overview	55
Worldwide Sales and Service	57

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our website (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is “DSXXXXXXXXA”, where “XXXXXXXX” is the document number and “A” is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the RN4870/71 PICtail™/PICtail Plus Daughter board. Items discussed in this chapter include:

- [Document Layout](#)
- [Conventions Used in this Guide](#)
- [Recommended Reading](#)
- [The Microchip Website](#)
- [Development Systems Customer Change Notification Service](#)
- [Customer Support](#)
- [Document Revision History](#)

DOCUMENT LAYOUT

This document describes how to use the RN4870/71 PICtail™/PICtail Plus Daughter board as a development tool to emulate and debug firmware on a target board, as well as how to program devices. The document is organized as follows:

- **Chapter 1. “Overview”** – This chapter introduces the RN4870/71 PICtail™/PICtail Plus Daughter board and provides an overview of its various features.
- **Chapter 2. “Interface Description”** – This chapter illustrates and describes the various components and the interfaces available on the PICtail/PICtail Plus Daughter board and the Sensor board.
- **Chapter 3. “Quick Start Guide”** – This chapter describes how to connect the RN4870/71 PICtail/PICtail Plus Daughter board to the Host PC (via USB and connecting and communicating with a smartphone using the SmartDiscover app), how to create and access GATT services, and how to read the Sensor Board Peripheral IO ports.
- **Chapter 4. “RN4870 Sensor Board”** – This chapter describes the procedure to set up the RN4870 to use the various sensors on the Sensor board and also

explains how to connect and communicate with the BLE Sensor app on the iOS smartphone.

- **Chapter 5. “PIC Configuration Library”** – This chapter provides information on how to use the Configuration Library to configure the RN4870/71 module using a host MCU over UART.
- **Appendix A. “Updating PICtail Firmware”** – This appendix shows the steps to update the firmware on the RN4870/71 PICtail/PICtail Plus Daughter board.
- **Appendix B. “Schematics and BOM”** – This appendix shows the schematics and BOM for the RN4870/71 PICtail/PICtail Plus Daughter board and the Sensor board.
- **Appendix C. “Bluetooth Low Energy Primer”** – This Appendix provides a brief summary of GAP roles and GATT services
- **Appendix D. “Sensor Board Configuration Command Text”** – This appendix provides an example of a script that can be used to control and communicate with the Sensor board.

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	<i>MPLAB[®] IDE User's Guide</i>
	Emphasized text	...is the <i>only</i> compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	<u><i>File>Save</i></u>
Bold characters	A dialog button	Click OK
	A tab	Click the Power tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	#define START
	Filenames	autoexec.bat
	File paths	c:\mcc18\h
	Keywords	_asm, _endasm, static
	Command-line options	-Opa+, -Opa-
	Bit values	0, 1
	Constants	0xFF, 'A'
Italic Courier New	A variable argument	<i>file.o</i> , where <i>file</i> can be any valid filename
Square brackets []	Optional arguments	mcc18 [options] <i>file</i> [options]
Curly brackets and pipe character: { }	Choice of mutually exclusive arguments; an OR selection	errorlevel {0 1}
Ellipses...	Replaces repeated text	var_name [, var_name...]
	Represents code supplied by user	void main (void) { ... }

RECOMMENDED READING

This user's guide describes how to use the RN4870/71 PICtail™/PICtail Plus Daughter Board. Other useful documents are listed below. The following Microchip documents are recommended as supplemental reference resources.

RN4870/71 Bluetooth® 4.2 Low Energy Module Data Sheet (DS50002489)

This document provides the technical specifications for the RN4870/71 module and is available for download from the Microchip website (www.microchip.com).

RN4870/71 Bluetooth® Low Energy Module User's Guide (DS50002466)

This document provides information for configuring the RN4870/71 module including a command reference, scripting features and connection examples.

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events; and listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com, click on **Customer Change Notification** and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools
- **Emulators** – The latest information on the Microchip MPLAB® REAL ICE™ In-Circuit Emulator
- **In-Circuit Debuggers** – The latest information on the Microchip In-Circuit Debugger, MPLAB ICD 3
- **MPLAB X IDE** – The latest information on Microchip MPLAB X IDE, the Windows® Integrated Development Environment for development systems tools
- **Programmers** – The latest information on Microchip programmers including the PICkit™ 3 development programmer

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at:

<http://www.microchip.com/support>.

DOCUMENT REVISION HISTORY

Revision A (December 2016)

This is the initial release of this document.

NOTES:

Chapter 1. Overview

1.1 INTRODUCTION

This document describes the hardware and software requirements for the RN4870/71 PICtail™/PICtail Plus board, part numbers: RN-4870-SNSR and RN-4871-PICtail.

The RN4870/71 PICtail™/PICtail Plus Daughter Board enables the designer to evaluate and demonstrate the capabilities of the RN4870/71 Bluetooth® 4.2 RF module.

The RN4870/71 PICtail/PICtail Plus Daughter board include the following key features:

- Integrated configuration and programming interface
- LED and push button bank for prototype user inputs
- MCP2200 USB to UART bridge for plug-and-play connectivity to host

An additional daughter board is included in the RN-4870-SNSR to demonstrate peripheral access using Bluetooth Low Energy (BLE).

In addition to RN4870/71 PICtail/PICtail Plus Daughter board hardware, several smartphone applications are provided to demonstrate Bluetooth data connections to the on-board RN4870/71 module.

The demonstration apps are available on Apple AppStore® and Google Play™ Store and include:

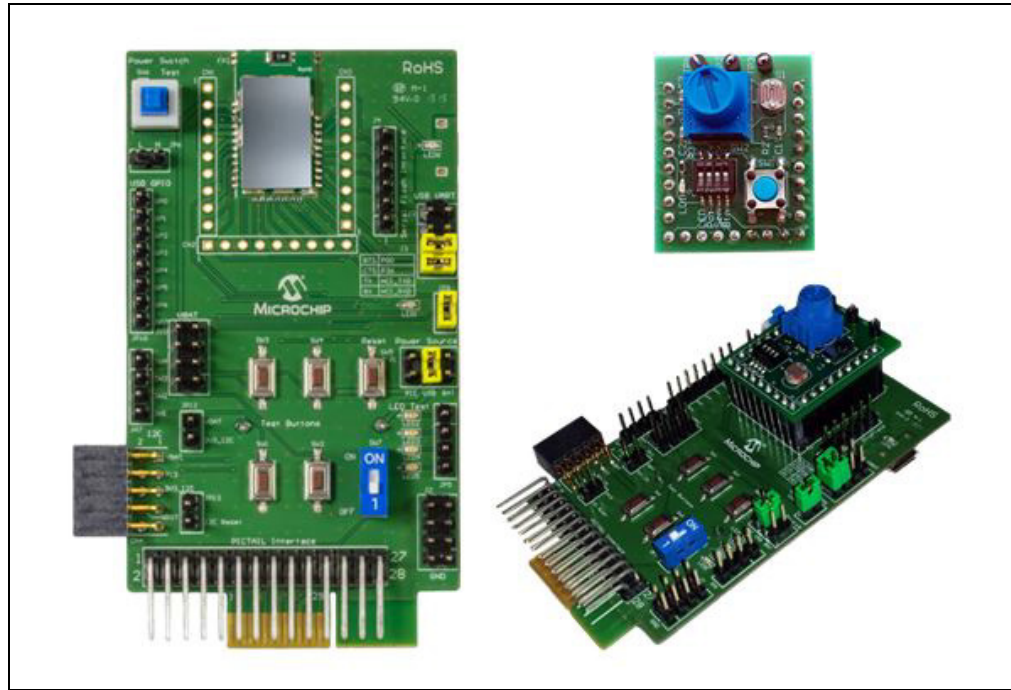
- SmartDiscover
- SmartData
- BLESensorApp

For more information on the RN487x commands and specifications, refer to the “*Bluetooth® 4.2 Low Energy Module Data Sheet*” (DS50002489) and the “*Bluetooth® Low Energy Module User’s Guide*” (DS50002466) available for download from the Microchip product web page at www.microchip.com/RN4870.

1.2 RN4870/71 PICtail™/PICtail PLUS DAUGHTER BOARD DESCRIPTION

The RN4870/71 PICtail/PICtail Plus Daughter board provides rapid prototyping and developing for Bluetooth data applications for Bluetooth Low Energy. It can be powered via USB host or through the Microchip PICtail Plus interface. The RN4870/71 PICtail/PICtail Plus Daughter board uses the RN4870/71 module, a fully certified Bluetooth 4.2 Low Energy module. The RN4870/71 PICtail/PICtail Plus Daughter board provides a USB-UART converter allowing flexible interface to a host PC, a PC terminal utility and smartphone apps to drive both BLE serial data connections and custom BLE services. The RN4870/71 PICtail/PICtail Plus Daughter board also provides Microchip PICtail and PICtail Plus interfaces to be able to interface with the Microchip PIC® micro-controllers using standard Microchip development tools.

FIGURE 1-1: RN4870/71 PICtail™/PICtail PLUS DAUGHTER BOARD AND SENSOR BOARD



1.3 FEATURES

The RN4870/71 PICtail/PICtail Plus Daughter board has the following features:

- Fully certified Bluetooth Low Energy 4.2 RF module
- +2 dBm maximum TX power
- On-Board Dual In-Line Package (DIP) switch block to set operating modes
- PICtail Plus and PICtail interfaces to fully access RN4870 pins using the external PIC MCU
- Embedded MCP2200 USB-UART converter to enable Application mode and programming interface to update firmware and configuration settings

Chapter 2. Interface Description

2.1 INTRODUCTION

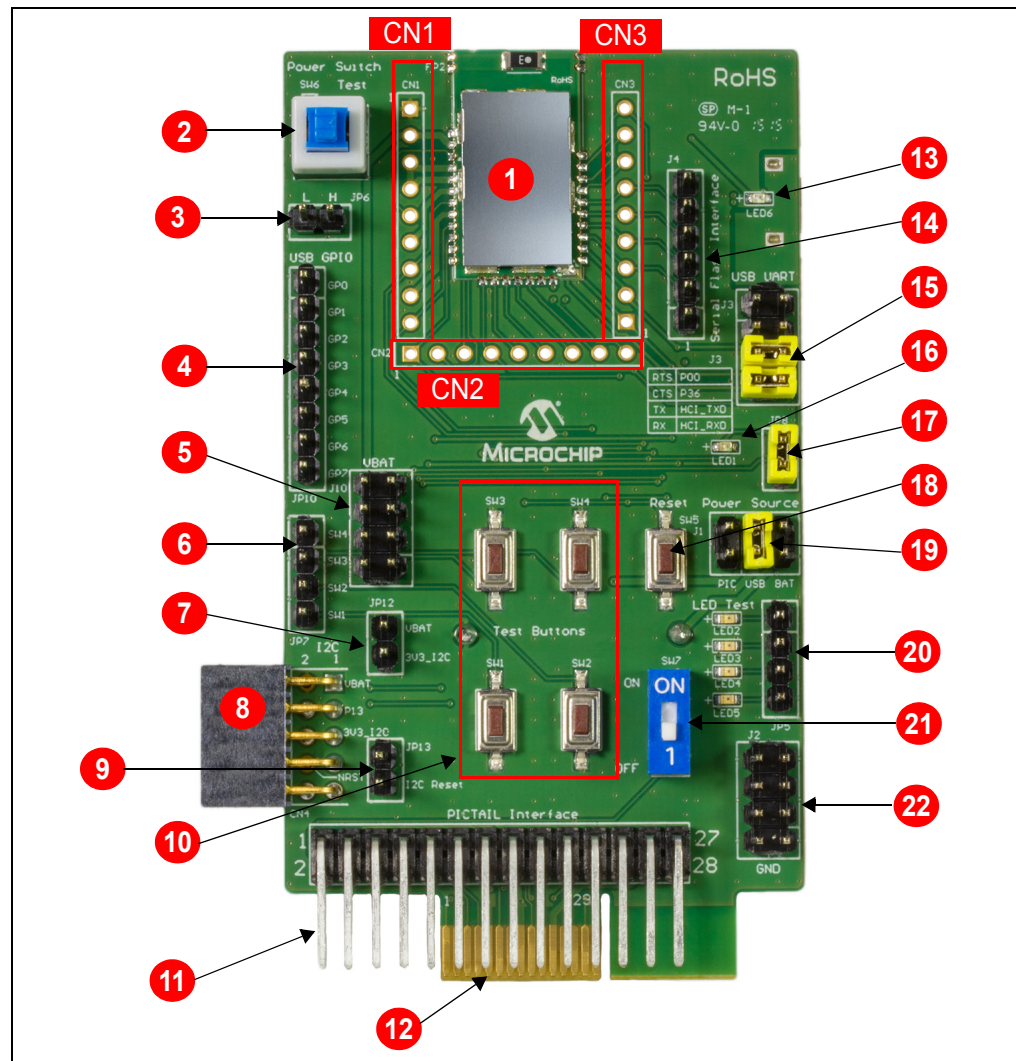
This chapter contains the board and interface descriptions of the following boards:

- [RN4870 PICtail/PICtail Plus Daughter Board](#)
- [RN4871 PICtail/PICtail Plus Daughter Board](#)
- [Sensor Board](#)

2.2 RN4870 PICtail/PICtail PLUS DAUGHTER BOARD

Figure 2-1 shows the interfaces of the RN4870 PICtail/PICtail Plus Daughter board. The PICtail board can be used in Standalone mode powered from either an external USB host, coin-cell battery, or from the PIC® Explorer Development board.

FIGURE 2-1: RN4870



2.2.1 Interface Description

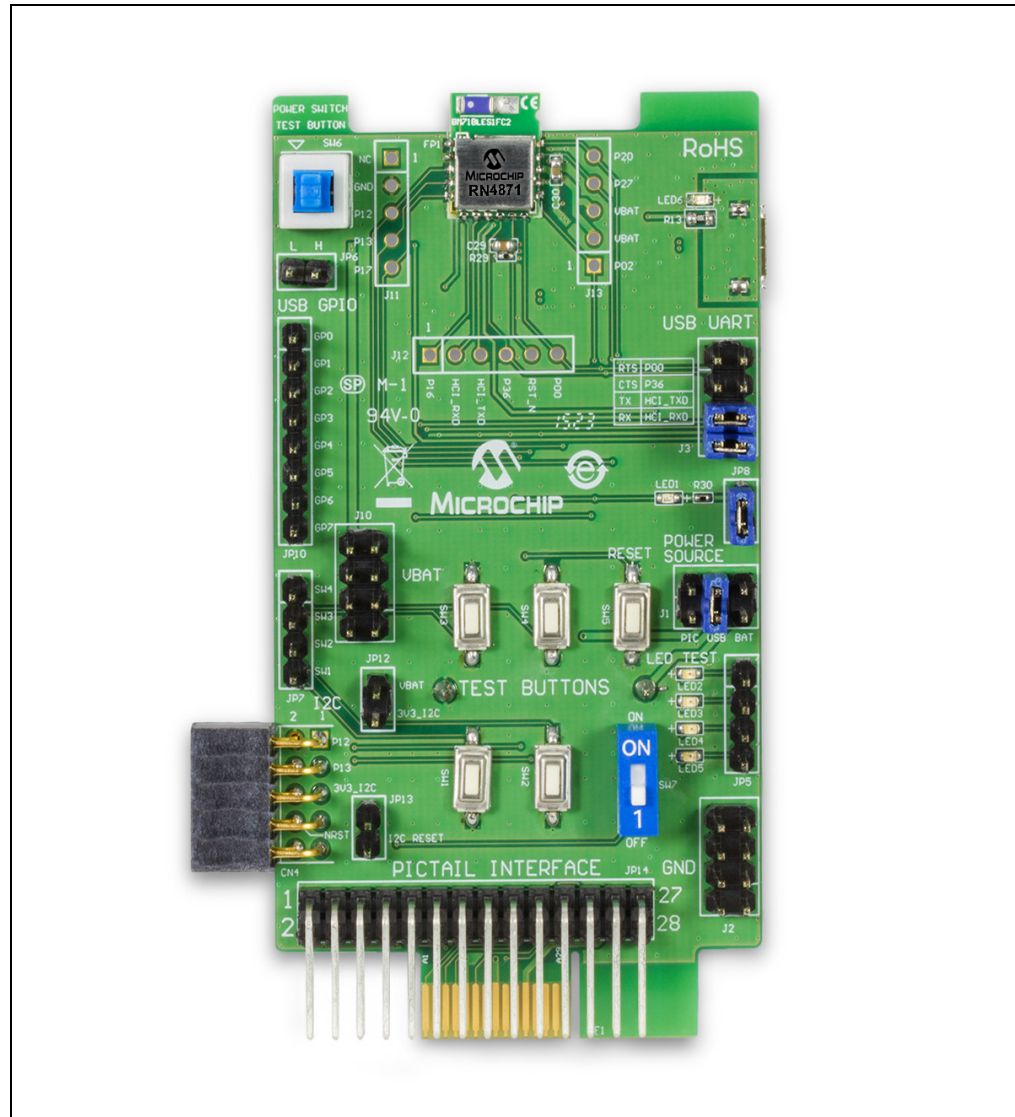
1. RN4870 module
 2. Power switch to wake-up the module from deep sleep (SW6)
 3. Power test point (JP6)
 4. Interface to MCP2200 USB/UART bridge (JP10)
 5. VBAT supply and test point (J10)
 6. Test button interface to SW1-SW4 (JP7)
 7. Power to I²C bus (JP12); I²C features depend on firmware revision
 8. Connector to I²C bus on RN4870 (CN4); I²C features depend on firmware revision
 9. I²C reset (JP13); I²C features depend on firmware revision
 10. SW1-SW4 test buttons
 11. PICtail Interface (2x14 right-angle header pin) for PIC18 Explorer board
 12. PICtail Plus Interface (edge connector) for PIC16 Explorer board
 13. External power from USB host (LED6)
 14. SPI Serial Flash Interface (J4)
 15. UART connections (J3); RTS/CTS jumpers must be closed when hardware flow control is enabled on module
 16. Status indicator LED (LED1); for blink rate descriptions, refer to the “*Bluetooth® 4.2 Low Energy Module Data Sheet*” (DS50002489)
 17. LED1 power jumper; disconnects LED from the RN4870 module if an external MCU is monitoring this pin
 18. Hardware Reset line to RN4870 (SW5)
 19. Power options for PICtail (J1); PIC Explorer board, USB host, Battery (coin-cell CR2032) on the back of the board
 20. Connections to test LEDs 1-4 (JP5). The LEDs can be connected to PIO lines or to an external MCU
 21. Operating mode switch (SW7); 1: Application mode runtime, ON: Programming or Configuration mode to update firmware or configuration settings
 22. Ground Test Connector (J2)
- CN1, CN2 and CN3 are header pins to the RN4870 module pins. The headers are used either to connect the Sensor board to the PICtail board or as test points.

2.3 RN4871 PICtail/PICtail PLUS DAUGHTER BOARD

The RN4871 PICtail uses the RN4871 BLE module as shown in [Figure 2-2](#). Due to its smaller footprint, the RN4871 has less IO pins than the full size RN4870 module. For the specific pinouts of the RN4870 and the RN4871 modules, refer to the “*Bluetooth® 4.2 Low Energy Module Data Sheet*” (DS50002489).

Although the module sizes are different, the command interface is very similar. The “*Bluetooth® Low Energy Module User’s Guide*” (DS50002466) describes the difference between the RN4870 and the RN4871 command pins.

FIGURE 2-2: RN4871 PICtail™/PICtail PLUS DAUGHTER BOARD

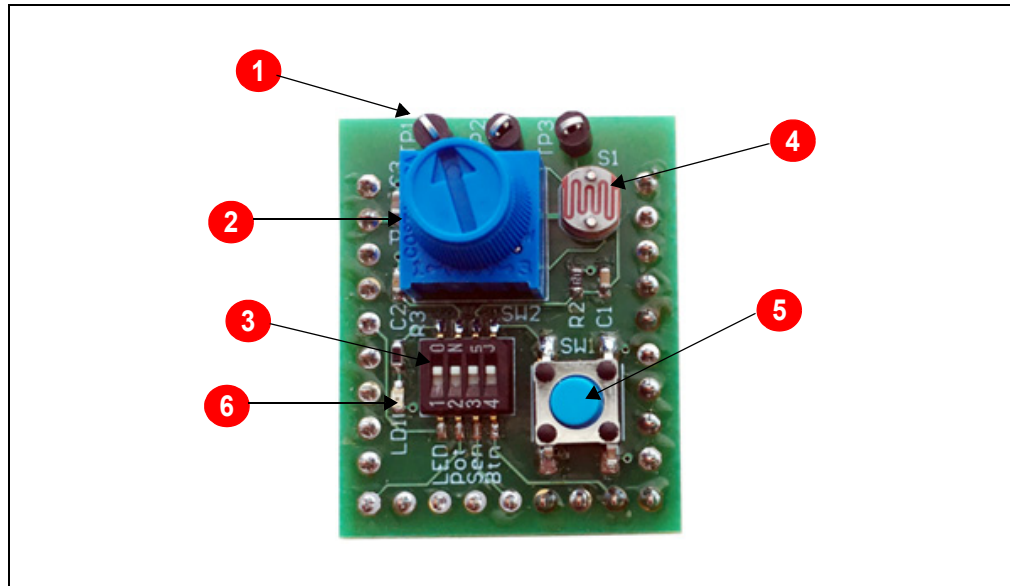


2.4 SENSOR BOARD

The RN4870/71 PICtail/PICtail Plus Daughter board is provided with the Sensor board to demonstrate the peripheral IO capabilities. The Sensor board interfaces are illustrated in [Figure 2-3](#).

The Sensor board is an accessory designed for the RN4870 PICtail/PICtail Plus Daughter board to demonstrate digital and analog IO capabilities over Bluetooth Low Energy connections. In order to use the Sensor board with the RN4870, it must be configured as described in [Chapter 4. "RN4870 Sensor Board"](#).

FIGURE 2-3: RN4871 PICtail SENSOR BOARD



2.4.1 Interface Descriptions

1. Test Point for ADC channels
2. Variable Resistor to drive ADC0 input
3. DIP Switch block to connect or disconnect sensors from the RN4870 module
4. Light Sensor to drive ADC1
5. Push Button for PIO2
6. LED connected to pin P22

Chapter 3. Quick Start Guide

3.1 OVERVIEW

The simplest method to access the RN4870/71 is to connect it to a PC host that supports USB CDC virtual COM (serial) ports. Simple ASCII commands can be sent to the RN4870/71 module by using a terminal emulator application.

To interact directly with the RN4870/71 module, the following softwares are needed:

- PC Host supporting USB CDC virtual serial port
 - The RN4870/71 PICtail uses MCP2200 USB/UART bridge. The drivers can be downloaded from www.microchip.com/MCP2200.
- Terminal Emulator Application
 - TeraTerm or CoolTerm is recommended
- Microchip SmartDiscover App for iOS® or Android
 - Available on AppStore® (for iOS) or Google Play™ Store (Android)
- RN-4870-SNSR Support Package (RN4870-Sensor-Board-Support.zip)
 - This is a collection of files and utilities to assist in configuring the Sensor board. The files can be downloaded from the www.microchip.com/RN4870 product page.

3.2 CONNECTING THE RN4870/71 PICtail/PICtail PLUS DAUGHTER BOARD TO A HOST PC

To connect the RN4870/71 PICtail/PICtail Plus Daughter board to a host PC, perform the following steps:

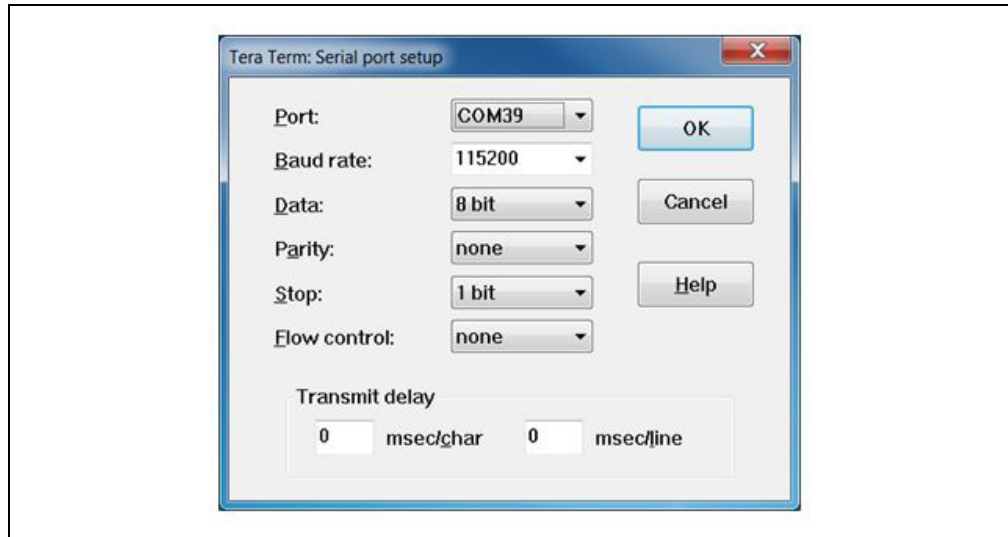
1. Ensure that the RN4870/71 PICtail/PICtail Plus Daughter board is configured as follows:
 - a) Middle jumper on J1 is closed to select USB power option
 - b) Operating mode switch (SW7) is in position 1 to enable Application runtime mode
 - c) TX and RX jumpers (J3) are in closed position

<p>Note: If hardware flow control is enabled then RTS and CTS jumpers must be installed.</p>

2. Connect the RN4870/71 PICtail/PICtail Plus Daughter board to the host PC using micro USB cable. Perform the following steps:
 - a) Verify that the virtual COM port is enumerated on the host PC. If the COM port does *not* enumerate, the MCP2200 drivers may be missing from the host PC. The drivers can be downloaded from the www.microchip.com/MCP2200 web page.
 - b) Ensure that LED2 (blue) is in ON position indicating USB power
 - c) Press **Reset** button (SW5) and verify that LED1 is flashing slowly

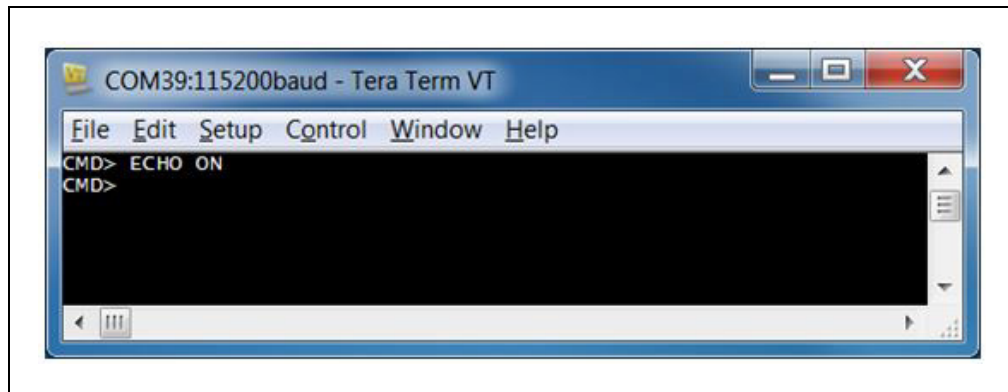
3. Start the Terminal Emulator software. In this example, TeraTerm is used. Configure the serial port settings using the enumerated COM port as shown in [Figure 3-1](#).

FIGURE 3-1: COM PORT SETTINGS FOR TERMINAL EMULATOR



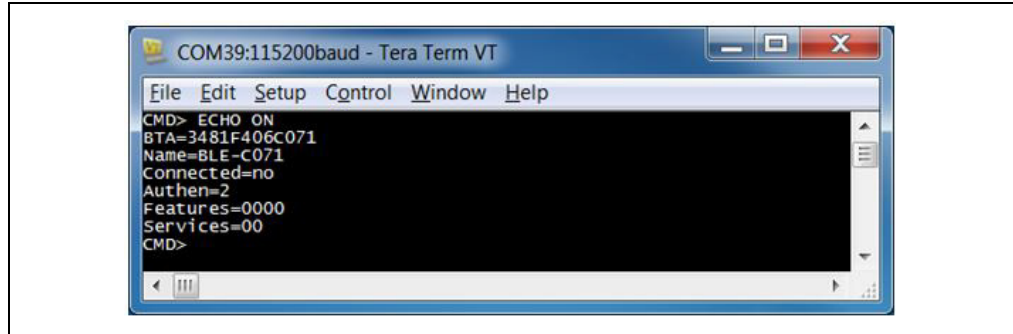
4. Enter Command mode by sending the command escape sequence \$\$\$\$. Pressing the dollar sign (\$) three times puts the RN4870/71 into Command mode and CMD> prompt is displayed. When interacting directly with the RN4870/71 using terminal emulator, enable local echo feature on the RN4870/71. Enter the + character on the command prompt. The ECHO ON response is displayed as shown in [Figure 3-2](#). Any character sent to the RN4870/71 is echoed back to the sender to improve interactivity.

FIGURE 3-2: COMMAND PROMPT AND ECHO ON RESPONSE



5. To display the basic configuration settings, press the letter **D** followed by <Enter> key (\r). The configuration settings can be modified using ASCII commands. Refer to the “RN4870/71 Bluetooth® Low Energy Module User’s Guide” (DS50002466) for a detailed explanation of the commands.

FIGURE 3-3: RESULTS OF “D” TO DISPLAY BASIC CONFIGURATION



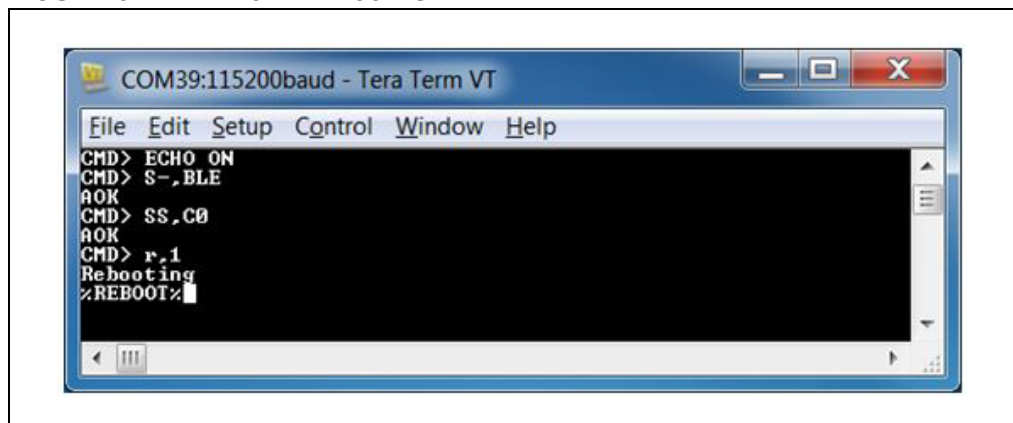
3.3 CHANGING SETTINGS USING ASCII COMMANDS

The RN4870/71 PICtail is shipped with a default configuration *not* including GATT services as noted by the result of the previous command **D** (display basic configuration). The services value is set to `Services=00`. The RN4870 module can be discovered using Bluetooth SmartDiscover App. The SmartDiscover App is available for iOS® devices on Apple AppStore. Check the Google Play Store for Android version.

To enable built-in services and connect to the RN4870/71 module, perform the following steps:

1. Connect the RN4870/71 PICtail to the host PC USB port:
 - a) Using Terminal Emulator, open the COM port to the RN4870/71 module
 - b) Type `$$$` to enter Command mode
 - c) Enter `+` to turn on ECHO. Refer to [Figure 3-2](#)
2. To change the default name and enable GATT services (Device Information Service and Transparent UART), send the following commands:
 - a) `S-,BLE` to create a unique serialized name based on Bluetooth (BT) address of the device
 - b) `SS,C0` to enable Device Information Profile and Transparent UART services
 - c) `R,1` to reboot the module and enable the configuration commands to take effect. Refer to [Figure 3-4](#).

FIGURE 3-4: s- AND ss COMMAND EXAMPLE

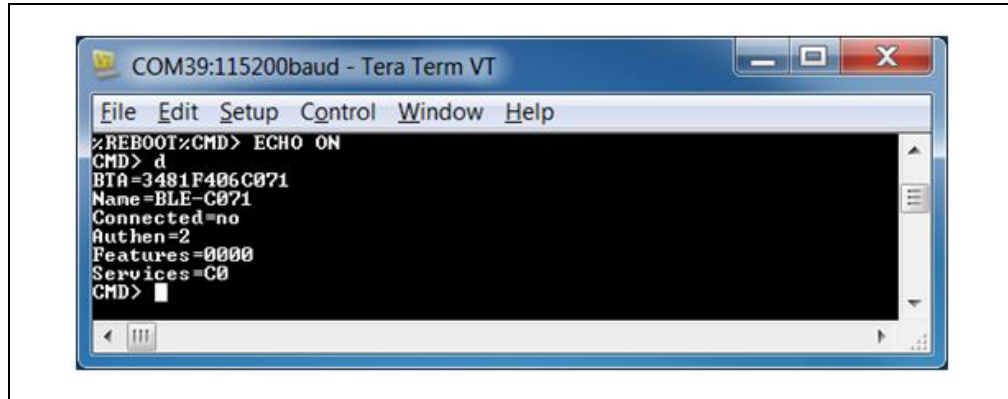


3. After any Set commands issued, the RN4870/71 module needs a device reboot (R, 1) for the changes to take effect.

To verify the configuration settings, perform the following steps:

- a) Type \$\$\$ to enter Command mode
- b) Enter + to turn on local echo
- c) Issue command D to display the settings
- d) Note the name of the device BLE-b1b0 where b1b0 are the low order bytes of the BT address; in this example, the Bluetooth is set to BLE-C071
- e) Verify that the services field is set to C0; refer to [Figure 3-5](#)

FIGURE 3-5: CONFIRM NEW SETTINGS IN COMMAND MODE



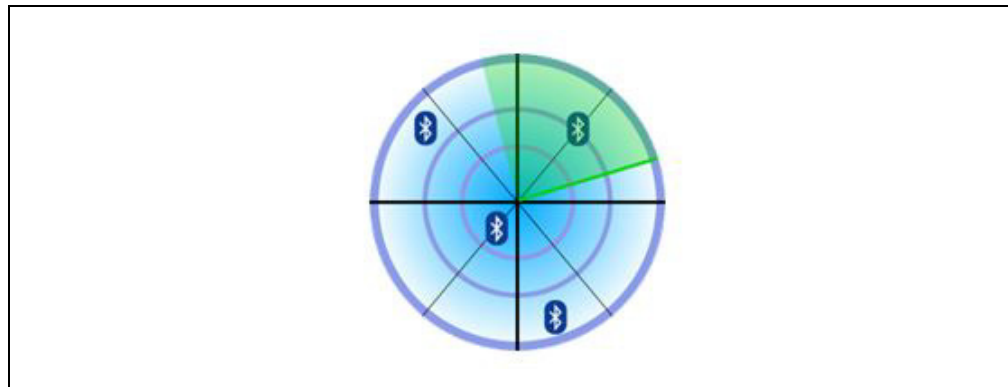
3.4 CONNECTING TO THE RN4870/71 USING SMARTDISCOVER APP

The RN4870/71 features can be demonstrated by using the SmartDiscover App.

To connect the RN4870/71 using SmartDiscover App, perform the following steps:

1. Configure the RN4870/71 PICtail according to the steps in [Section 3.3 "Changing Settings Using ASCII Commands"](#). Connect it to the host PC and type \$\$\$ to enter Command mode.
2. Download and install the SmartDiscover App on your device. See [Figure 3-6](#) for example of SmartDiscover App icon.

FIGURE 3-6: SMART DISCOVER ICON



3. Launch the SmartDiscover App. Refer to [Figure 3-7](#) for examples on how to demonstrate the SmartDiscover App on an iOS device.

FIGURE 3-7: SMART DISCOVER (iOS)



4. Verify that the RN4870/71 device name, BLE-b1b0, is displayed. In this example, the device name is BLE-C071 as shown in [Figure 3-7](#).
5. Click the RN4870/71 device from the list to initiate a connection.
6. Ensure that the connection is established. The GATT service view is illustrated in [Figure 3-8](#). The connection is also verified by the %CONNECT, 1, <MAC> status message returned by the RN4870/71 UART. The <MAC> is the address of the remote BT device that initiated the connection. The response in the terminal emulator is shown in [Figure 3-9](#).

FIGURE 3-8: GATT SERVICE VIEW WHEN CONNECTED

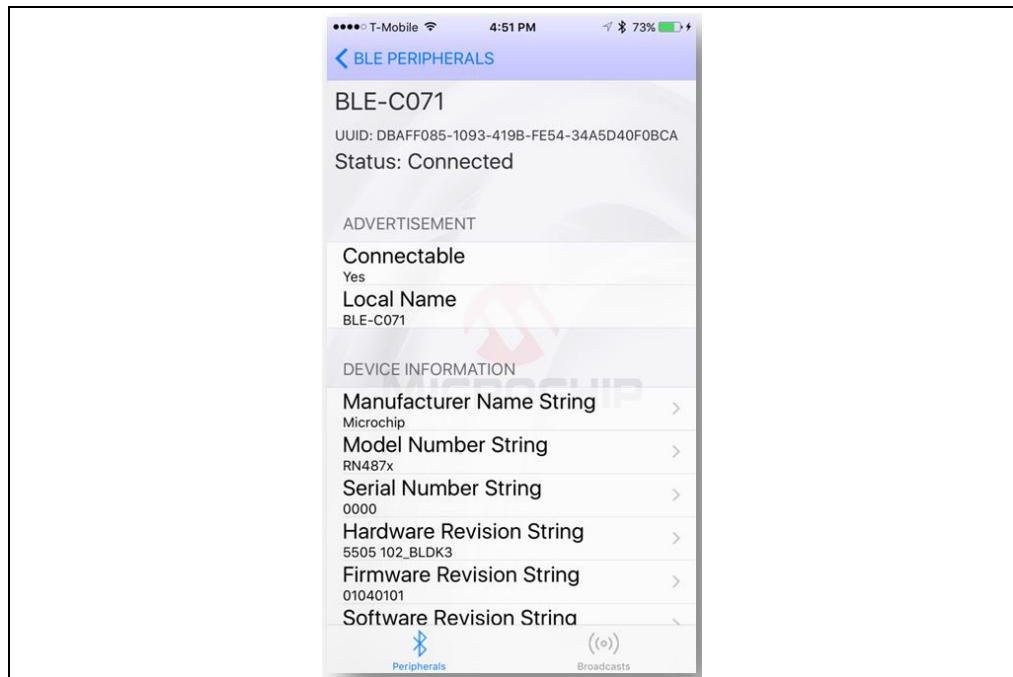
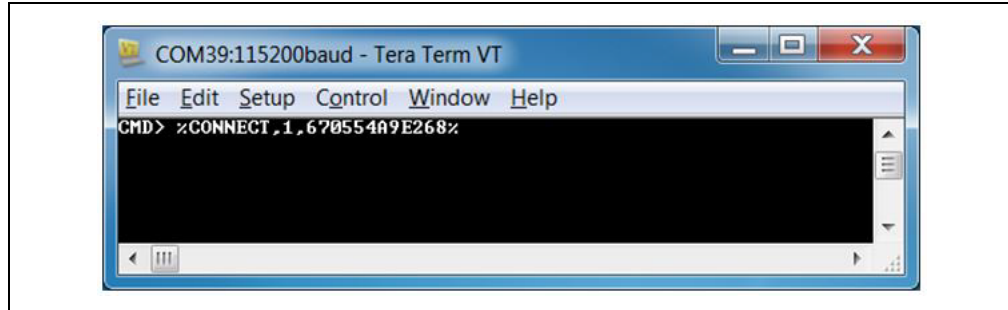


FIGURE 3-9: CONNECTED STATUS MESSAGE



3.5 CREATING CUSTOM GATT SERVICES

The RN4870/71 module supports custom (or private) GATT services. Public GATT services are defined by specifications published by the Bluetooth Special Interest Group (SIG). Private GATT services are defined by the user to host information stored in the GATT characteristics. For more information on GATT services, refer to [Appendix C. "Bluetooth Low Energy Primer"](#).

To create a private GATT service, enter the configuration commands listed in Step 3. The commands can be entered as shown in this user guide, or copied from configuration files and pasted into the terminal emulator. The cut and paste method saves time and minimizes keyboard entry errors. The configuration text files are included in the RN4870 Sensor board Support package that can be downloaded from the product web page at www.microchip.com/RN4870.

1. Connect the RN4870 PICtail to the host PC USB port
 - a) Using Terminal Emulator, open the COM port to the RN4870/71 module
 - b) Type \$\$\$ to enter Command mode
 - c) Enter + to turn on ECHO
2. Set Factory Default values
 - a) Enter SF, 1
 - b) Verify that the module reboots after the command is entered
3. Create the private GATT service with three characteristics by entering the following commands:

```
PS, 4D6963726F636869702D524E34383730
PC, BF3FBD80063F11E59E690002A5D5C501, 02, 02
PC, BF3FBD80063F11E59E690002A5D5C502, 02, 02
PC, BF3FBD80063F11E59E690002A5D5C503, 18, 04
```

These commands can be entered manually by typing each line followed by the <Enter> key from the file 3.4-GATT-service.txt, or each line copied and pasted one by one into the terminal emulator.

Command PS creates the GATT service, identified by the 16-byte value private UUID: 4D6963726F636869702D524E34383730. This command must be called before command PC.

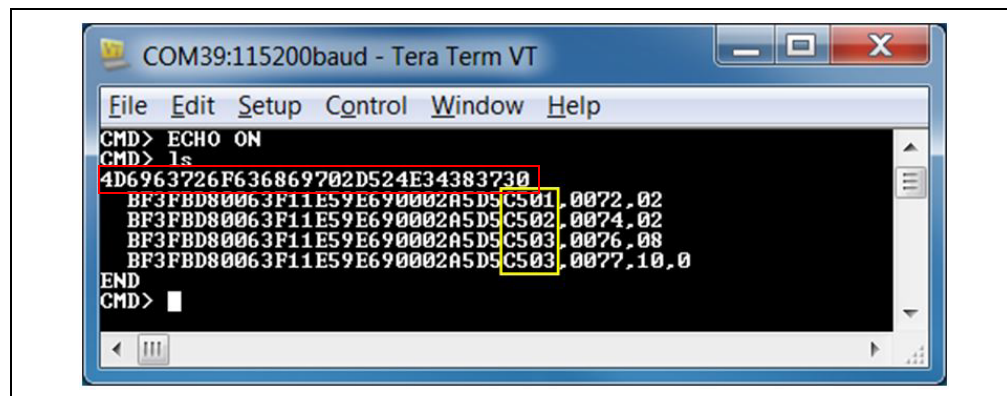
Command PC sets the private characteristics. Each characteristic is identified by the following UUIDs:

```
BF3FBD80063F11E59E690002A5D5C501,
BF3FBD80063F11E59E690002A5D5C502,
BF3FBD80063F11E59E690002A5D5C503.
```


Command PC expects three parameters. The first parameter is the UUID, the second parameter is the characteristic property (refer to Table C-1 in Appendix C. “Bluetooth Low Energy Primer”), and the third parameter is the size of the data value of the characteristic.

4. Reboot the module using command R, 1 to ensure that the new GATT service takes effect
5. Verify that the GATT service is correctly configured.
6. Enter Command mode (\$\$\$) after rebooting the module.
7. Issue the LS command to list the GATT server services and their characteristics. The response is illustrated in Figure 3-10.

FIGURE 3-10: PRIVATE SERVICES DISPLAY FROM LS COMMAND



3.6 ACCESSING GATT SERVICE USING UART COMMANDS AND SMARTDISCOVER APP

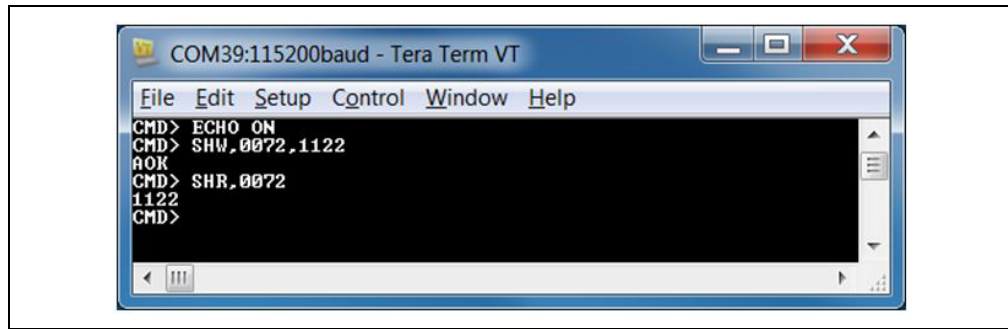
The result of the command LS is shown in Figure 3-10. A custom GATT service (UUID: 4D69637226F636869702D524E34383730) with three characteristics identified by low-order bytes C501, C502 and C503 from the 128-bit UUID. A 16-bit handle is assigned to each characteristic. Handles are used to reference and identify 128-bit characteristic UUIDs more efficiently in the GATT service.

Note that there are two references for C503 characteristic. In the first case, in reference 0076, the property value 08 has the write property enabled (refer to Table C-1 in Appendix C. “Bluetooth Low Energy Primer”). Likewise, reference 0077 has the notification property 10 enabled. This means that to write a value to characteristic C503, reference 0076 is used. To enable Client notifications on this characteristic, reference 0077 is used.

The following examples show how to read and write GATT characteristic values by using UART commands:

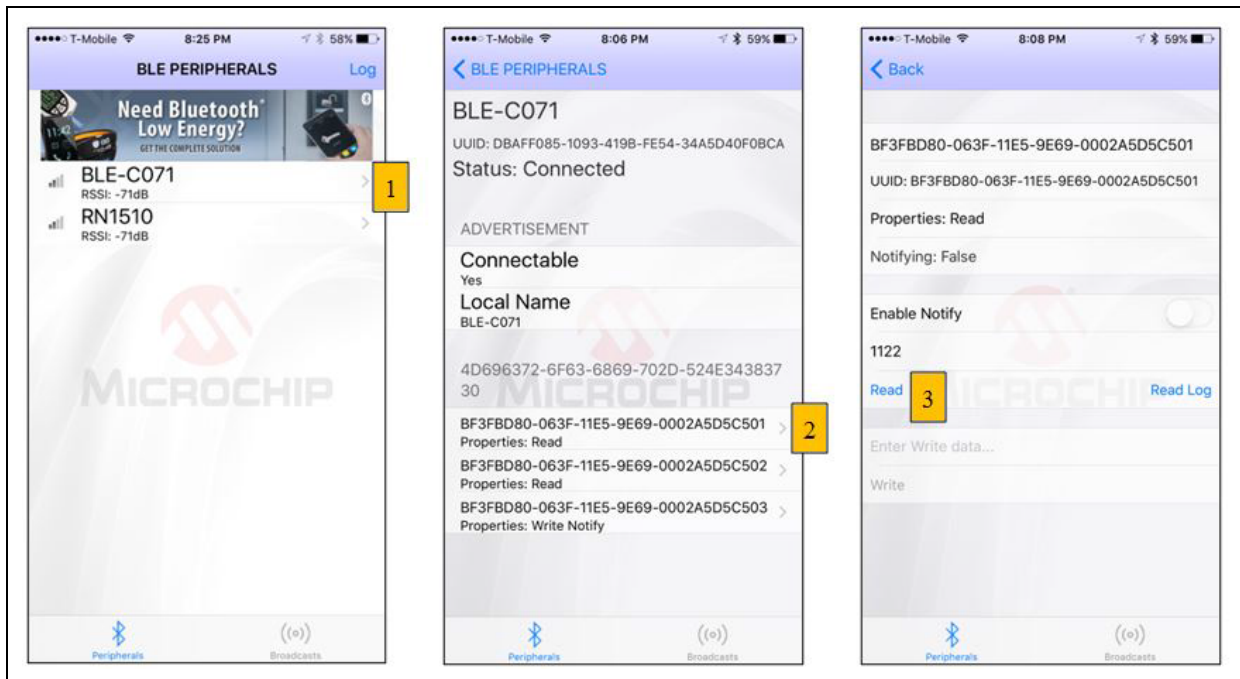
1. To write a value to the GATT Server characteristic C501, use the Server Handle Write (SHW) and the Server Handle Read (SHR) commands with reference 0072 as first parameter, following by hex-byte values as shown in Figure 3-11.

FIGURE 3-11: WRITING AND READING GATT VALUE BY HANDLE REFERENCE



- It is also possible to access the GATT server over a Bluetooth Low Energy connection using SmartDiscover App. Launch the SmartDiscover App and connect to the RN4870/71 PICtail/PICtail Plus Daughter board configured with the private GATT as shown in [Section 3.5 “Creating Custom GATT Services”](#). In this example, the device “BLE-C071” is used. Follow the steps in [Figure 3-12](#) to read the value of the GATT characteristic C501. In steps 1 and 2, tap on the name of your device (BLE-C071 >), on the UUID (BF3FBD80063F11E59E690002A5D5C501 and on **Read** in step 3. The characteristic value is read from the RN4870/71 into the Smart Discover App.

FIGURE 3-12: USING SMARTDISCOVER TO READ GATT CHARACTERISTIC VALUE



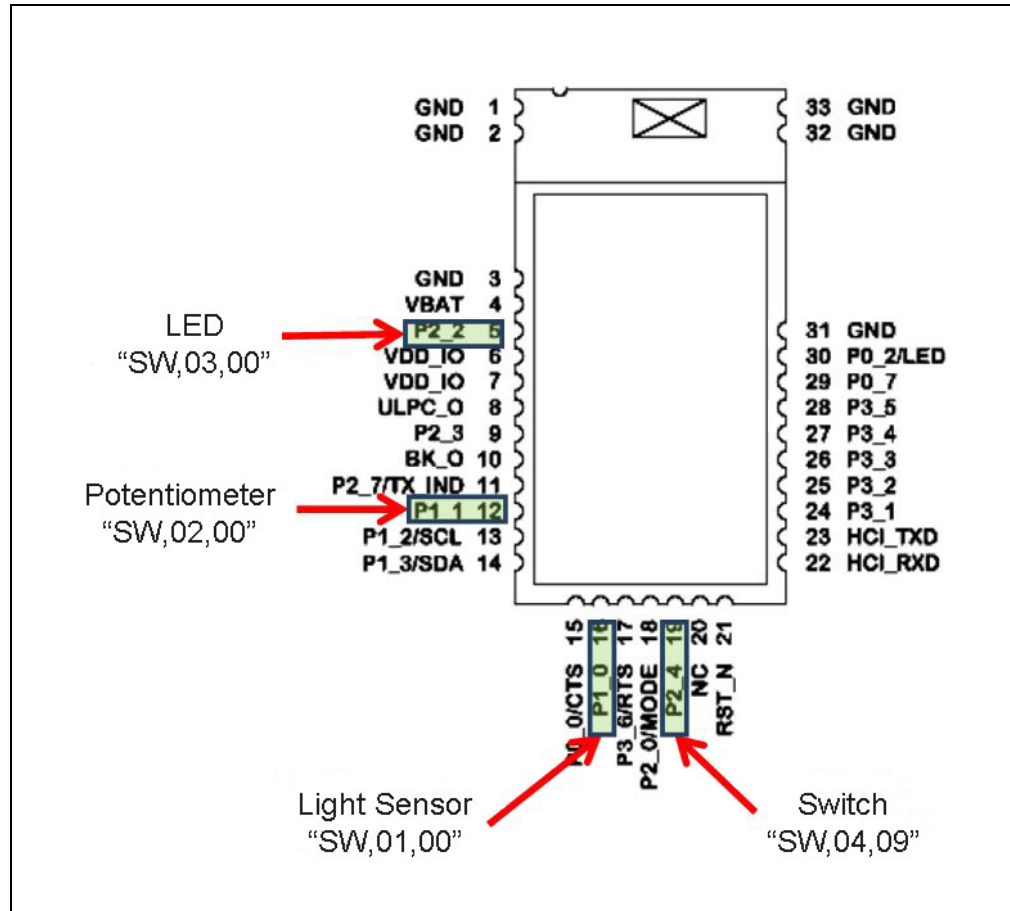
3.7 READING SENSOR BOARD PERIPHERAL IO PORTS USING UART COMMANDS

The RN4870/71 module has digital and analog peripheral inputs and outputs. For a detailed description of the IO interfaces, refer to the “*RN4870/71 Bluetooth® 4.2 Low Energy Module Data Sheet*” (DS50002489), and for the configuration details, refer to *Section 2.4.24* and *Section 2.6.5* through *Section 2.6.8* of the *RN4870/71 Bluetooth® Low Energy Module User’s Guide*” (DS50002466). Peripheral IO can be accessed through direct UART commands or over Bluetooth Low Energy connection by associating a GATT characteristic with a peripheral input or output.

In this section, UART commands are used to access peripheral IO on the Sensor board connected to the RN4870 PICtail as illustrated in [Figure 3-13](#).

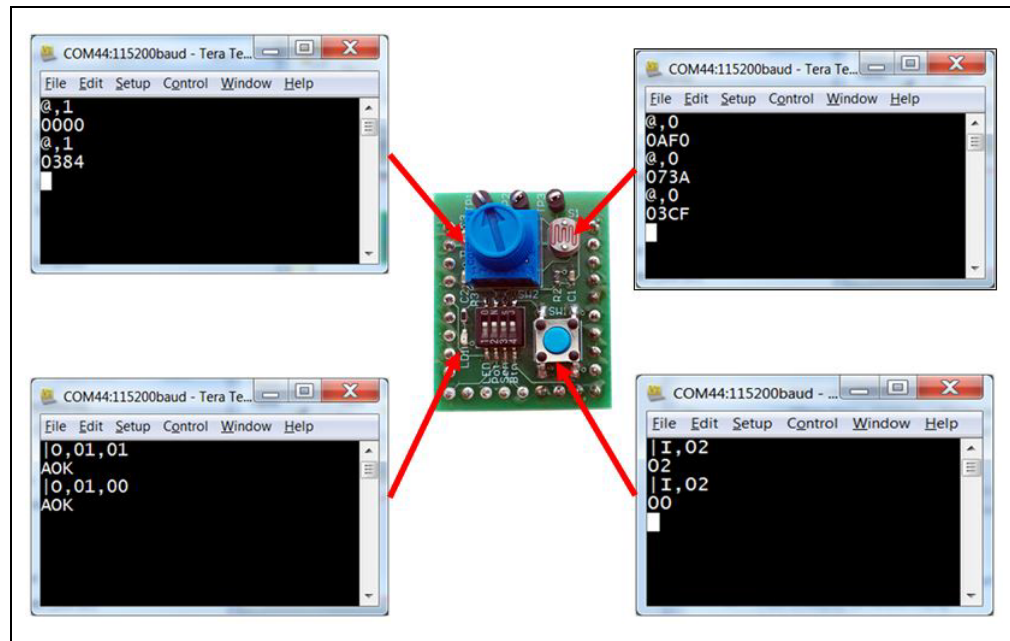
1. Connect the Sensor board to the RN4870 PICtail as shown in [Figure 1-1](#).
2. Set Factory Default to default Peripheral IO functions:
 - Using Terminal Emulator, open the COM port to the RN4870 module
 - Type \$\$\$ to enter Command mode
 - Enter SF, 1 and verify that the module reboots after the command is entered.
3. Type the following sequence of commands to set the RN4870 IO pin into the configuration as illustrated in [Figure 3-13](#):
 - Type \$\$\$ to enter Command mode
 - Enter + to enable local echo
 - Enter SW, 01, 00 for ADC input port 01 used by the light sensor
 - Enter SW, 02, 00 for ADC input port 02 used by the potentiometer
 - Enter SW, 03, 00 to configure digital IO port connected to the LED
 - Enter SW, 04, 09 to configure digital IO trigger connected to the push button
 - Enter SF, 1 to reboot the module for the settings to take effect

FIGURE 3-13: RN4870 PINS USED BY SENSOR BOARD INTERFACE



- After the IO ports are configured and the module is rebooted, perform the commands shown in Figure 3-14 to read and write peripheral IO.

FIGURE 3-14: COMMANDS TO READ AND WRITE SENSOR BOARD IO



Chapter 4. RN4870 Sensor Board

4.1 OVERVIEW

The RN4870 must be configured to enable the Sensor Board peripherals, and to enable the BLESensor App to communicate with the RN4870. The following is the configuration procedure:

1. Configure Peripheral IO port to sensors using command `SW`.
2. Define a GATT service (Sensor board) to hold the values from peripherals for a read access from a (GATT) client.
3. Similarly, when a GATT client writes a value to a GATT characteristic in a peripheral, there must be a method for the peripheral to write the value destined for characteristic into the peripheral port.
4. A GATT client on a Central (GAP) device is used to access the GATT characteristics in the peripheral which is an RN4870 with Sensor board. The GATT client for the Sensor board is a smartphone app named BLESensorApp. It is possible to use another RN4870 as the GATT client to access the RN4870 Sensor Board peripherals.

The critical function of associating GATT characteristic with a peripheral is performed by the RN4870 scripting capability.

The RN4870 can be configured manually by following the instructions in *Section 4.2* through *Section 4.5*. Alternatively, the RN4870 can also be configured by using the PC Utility (RN4870_SensorBoard_Cfg_Cmd.exe) as described in [Section 4.6 “Provision Utility”](#).

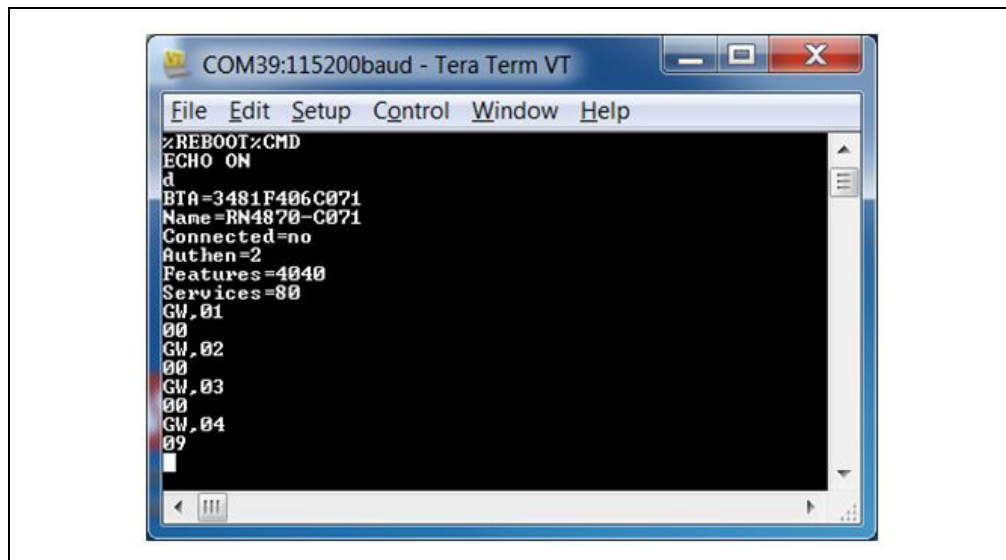
4.2 CONFIGURING THE RN4870 MODULE SETTINGS

The Sensor board can be configured by entering the following commands manually. Alternatively, the configuration commands are located in [Appendix D. “Sensor Board Configuration Command Text”](#). Each command can be copied to the clipboard via <CTRL+C> and sent to the RN4870 via TeraTerm using the <ALT+R> (Paste) command.

1. Connect the RN4870 PICtail to the host PC USB port
 - a) Using Terminal Emulator, open the COM port to the RN4870 module
 - b) Type `$$$` to enter Command mode
 - c) Enter `+` to turn on local echo
 - d) Enter `WP` to stop any running script
 - e) Enter `WC` to clear any previous script
 - f) Enter `PZ` to clear any previous user defined GATT service
 - g) Enter `SF, 1` to set the module into factory default and verify that the module reboots after the command is entered.
 - h) Type `$$$` to re-enter Command mode
 - i) Enter `+` to turn on local echo

2. Configure Module Features
 - a) Enter `S-`, RN4870 to serialize Bluetooth name
 - b) Enter `SS`, 80 to enable Device Information Profile
 - c) Enter `SR`, 4040 to enable scripting start on PWR_ON event and disable command prompt.
 - d) Enter `SW`, 01, 00 for ADC input port 01 used by the light sensor
 - e) Enter `SW`, 02, 00 for ADC input port 02 used by the potentiometer
 - f) Enter `SW`, 03, 00 to configure digital IO port connected to the LED
 - g) Enter `SW`, 04, 09 to configure digital IO trigger connected to the **Push** button
 - h) Enter `R`, 1 to reboot the module
 - i) Type `$$$` to re-enter Command mode
 - j) Enter `+` to turn on local echo
 - k) Enter command `D` to display settings as show in [Figure 4-1](#). Command `GW` is used to return to the current IO configuration.

FIGURE 4-1: RN4870 MODULE SETTINGS CONFIGURATION



4.3 SENSOR BOARD GATT SERVICE

The data from the RN4870 peripheral IO are stored in the characteristics of a custom GATT service accessed by the Sensor Board BLE Client App. To create the Sensor Board GATT Service, enter the following commands while in Command mode:

1. `PS,AD11CF40063F11E59E690002A5D5C51B` to create a 128-bit UUID to identify the GATT Service.
2. `PC,BF3FBD80063F11E59E690002A5D5C501,10,02` to declare a two-byte characteristic with notify property enabled. This characteristic stores the value of the ADC channel used for light sensor.
3. `PC,BF3FBD80063F11E59E690002A5D5C502,10,02` to declare a two-byte characteristic with notify property enabled. This characteristic stores the value of ADC channel used for potentiometer.
4. `PC,BF3FBD80063F11E59E690002A5D5C503,18,14` to declare a 20-byte value used to manage both the **Push** button state (virtual LED in App) and the LED blink rate on the Sensor board. Both notify and write properties are enabled for this characteristic.

5. PC,BF3FBD80063F11E59E690002A5D5C504,10,02 to declare a two-byte characteristic with notify property enabled. This characteristic stores the value of the ADC channel used for the RN4870 internal temperature sensor.
6. PC,BF3FBD80063F11E59E690002A5D5C505,10,02 to declare a two-byte characteristic with notify property enabled. This characteristic stores the value of the ADC channel used for the RN4870 internal battery voltage sensor.
7. Command R,1 to reboot the RN4870 module
8. Command \$\$\$ to enter Command mode
9. Command + to enable local echo
10. Command LS (list services) to display Sensor Board GATT Service as shown in Figure 4-2.

FIGURE 4-2: SENSOR BOARD GATT SERVICE DECLARED IN RN4870

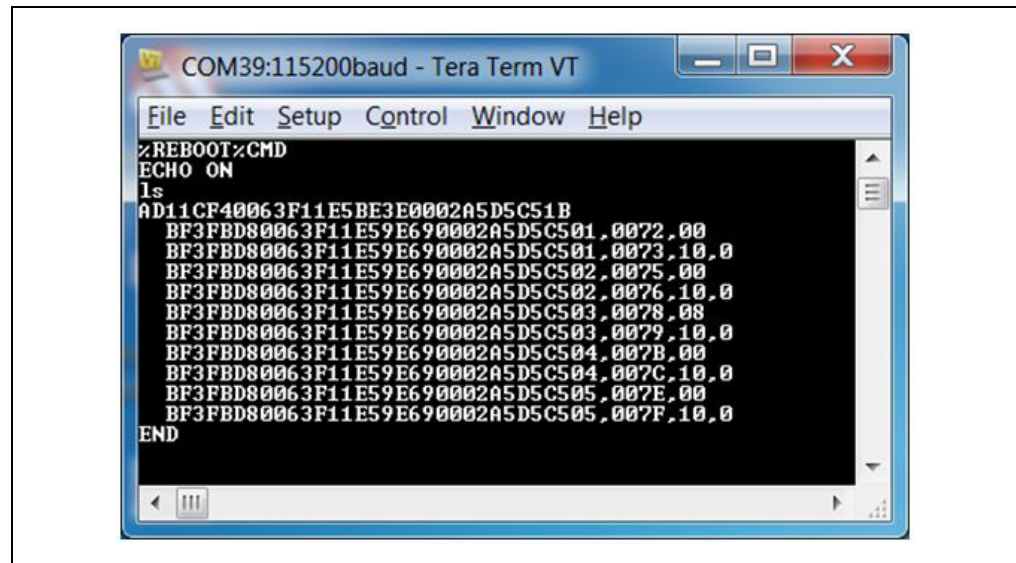


Figure 4-2 illustrates that a custom GATT service has been created in the RN4870 module. A 16-bit handle is assigned to each characteristic where each characteristic has a value in the handle and a property handle. The characteristic value and the properties are accessed by a short and more efficient 16-bit handle instead of the 128-bit UUID value. For example, two handles 0072 and 0073 are associated with the GATT characteristic BF3FBD80063F11E59E690002A5D5C501. A GATT client, such as a smartphone App, uses 0072 to read/write characteristic values, while 0073 is used to enable/disable the notify property.

The next step is to transfer the sensor values from the peripheral IO into the GATT characteristics. This is done with the RN4870 scripting feature as described in [Section 4.4 “Transferring Sensor Data into GATT Service Using Scripting”](#).

4.4 TRANSFERRING SENSOR DATA INTO GATT SERVICE USING SCRIPTING

For more information about the Sensor Board GATT service, refer to [Section 4.3 “Sensor Board GATT Service”](#). To populate the characteristics with data from the Sensor Board peripherals, and conversely transfer values received from the GAP client into peripheral IO channels, the RN4870/71 scripting feature is used. For more details on the scripting capabilities, refer to the “*Bluetooth® Low Energy Module User's Guide*” (DS50002466).

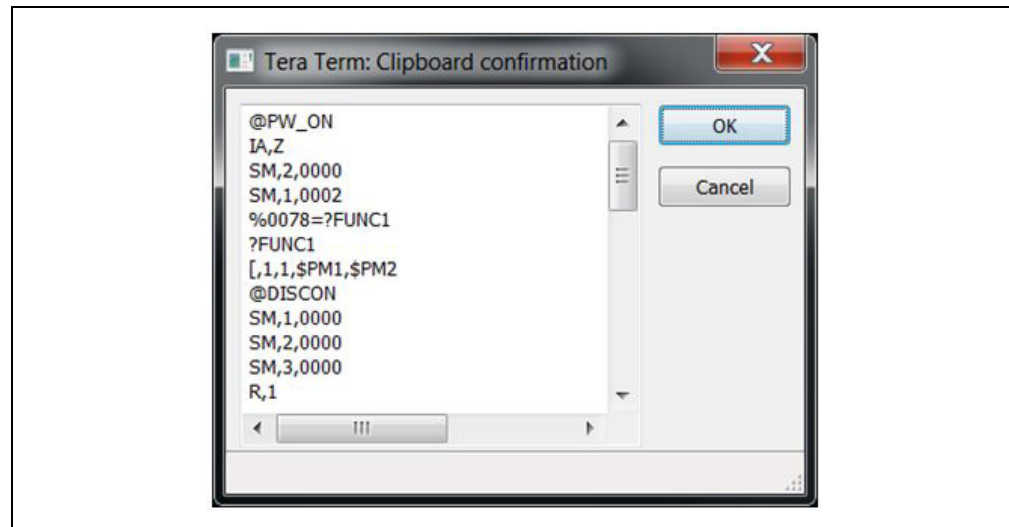
The script that drives the peripheral values to and from the GATT service is written into the Non-Volatile Memory (NVM) of the RN4870/71 via the UART.

First, put the module into Script Entry mode. The script is a series of text commands separated by carriage returns. The actual script is displayed in [Section D.1.3 “Sensor Board Script”](#). It can be copied to the clipboard and pasted in UART using TeraTerm. Similarly, the PC Utility can load it into RN4870.

To load the Sensor board script manually, perform the following steps:

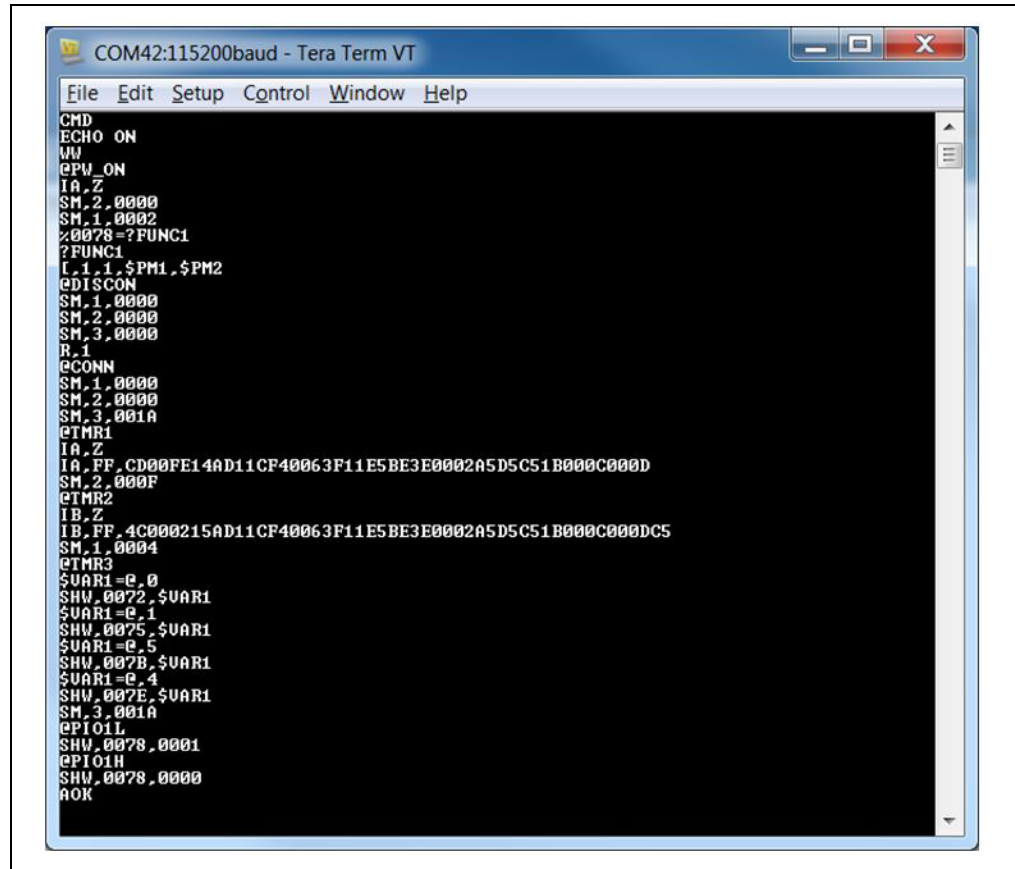
1. Connect the RN4870 PICtail to the host PC USB port
 - a) Using Terminal Emulator, open the COM port to the RN4870 module
 - b) Type \$\$\$ to enter Command mode
 - c) Enter + to turn on ECHO.
2. Enter Script Entry mode
 - a) Enter command ww to put the RN4870 module into Script Entry mode. This command does *not* return an AOK response.
 - b) Copy the script text from [Section D.1.3 “Sensor Board Script”](#) into the clipboard using <Ctrl+C>.
 - c) Paste the script from TeraTerm using the <Alt+R> (Paste) command. A dialog window showing the script text from the clipboard is displayed as shown in [Figure 4-3](#).

FIGURE 4-3: SENSOR BOARD SCRIPT DISPLAYED FROM CLIPBOARD



- d) Press **OK** to load the script into the RN4870.
- e) After the script is pasted into the RN4870, exit Script Entry mode by pressing the <ESC> key. An AOK response is sent to the terminal. The script entry is shown in [Figure 4-4](#).

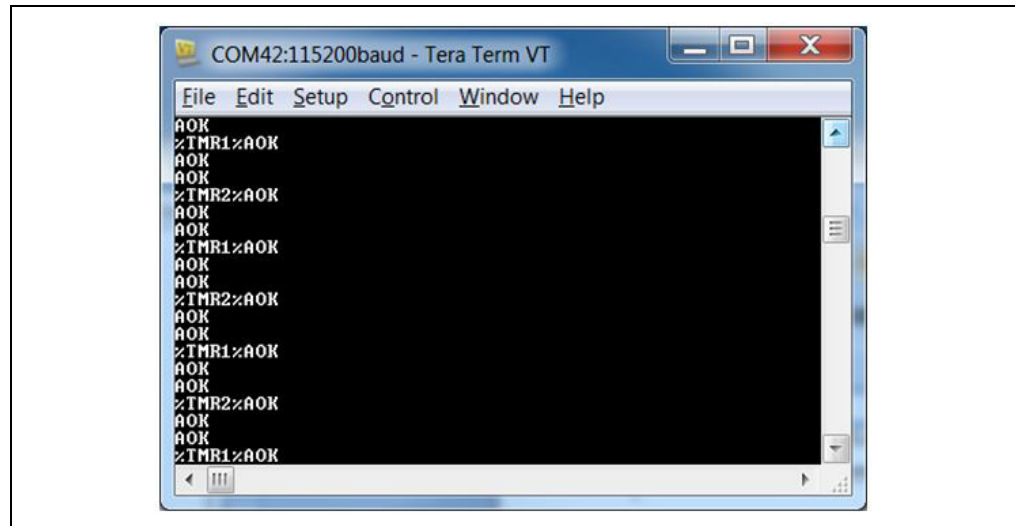
FIGURE 4-4: COMPLETED SENSOR BOARD SCRIPT



```
COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help
CMD
ECHO ON
MW
@PW_ON
IA,Z
SM,2,0000
SM,1,0002
%0078=?FUNC1
?FUNC1
L,1,1,$PM1,$PM2
@DISCON
SM,1,0000
SM,2,0000
SM,3,0000
R,1
@CONN
SM,1,0000
SM,2,0000
SM,3,001A
@TMR1
IA,Z
IA,FF,CD00FE14AD11CF4063F11E5BE3E0002A5D5C51B000C000D
SM,2,000F
@TMR2
IB,Z
IB,FF,4C000215AD11CF4063F11E5BE3E0002A5D5C51B000C000DC5
SM,1,0004
@TMR3
$UAR1=e,0
$HW,0072,$UAR1
$UAR1=e,1
$HW,0075,$UAR1
$UAR1=e,5
$HW,007B,$UAR1
$UAR1=e,4
$HW,007E,$UAR1
SM,3,001A
@PIO1L
$HW,0078,0001
@PIO1H
$HW,0078,0000
AOK
```

- f) Enter command `R, 1` to reboot the RN4870 module. After the reboot, the Sensor board is ready to use. The script event handlers can be seen on the TeraTerm display as shown in [Figure 4-5](#).

FIGURE 4-5: SENSOR SCRIPT RUNNING



```
COM42:115200baud - Tera Term VT
File Edit Setup Control Window Help
AOK
%TMR1%AOK
AOK
AOK
%TMR2%AOK
AOK
AOK
%TMR1%AOK
AOK
AOK
%TMR2%AOK
AOK
AOK
%TMR1%AOK
AOK
AOK
%TMR2%AOK
AOK
AOK
%TMR1%AOK
```

4.5 BLESENSORAPP SMARTPHONE APP

The RN4870 Sensor board is a BLE Peripheral that advertises a GATT service to BLE Central devices. In this demonstration, the BLE Central device is a smartphone application named BLESensorApp. It is available for iOS and Android platforms via iTunes AppSTORE, and Google Play Store, respectively. Search for the “*Microchip BLESensorApp*”, and download the App matching the icon in [Figure 4-5](#).

FIGURE 4-6: BLESENSORAPP ICON



The BLESensorApp requires the Sensor board to be configured as described in [Section 4.2-4.4](#).

1. Ensure that the Sensor board is attached to the RN4870 PICtail as shown in [Figure 1-1](#).
2. Verify the jumpers and switches are positioned according to [Figure 2-1](#) (SW7 must be in position 1: Application mode).
3. Power the PICtail via USB
 - a) Connect the RN4870 to power up.
 - b) Verify that LED1 blinks blue with long interval.
4. Launch the BLESensorApp from the smartphone. In this example, the iOS version is used as shown in [Figure 4-7](#).
5. The BLESensorApp attempts to connect to the first RN4870 Sensor board it discovers. Once connected, the main user interface of the BLESensorApp is displayed.

FIGURE 4-7: BLE DATA FLOW FROM SENSOR BOARD PERIPHERAL IO TO BLESENSORAPP

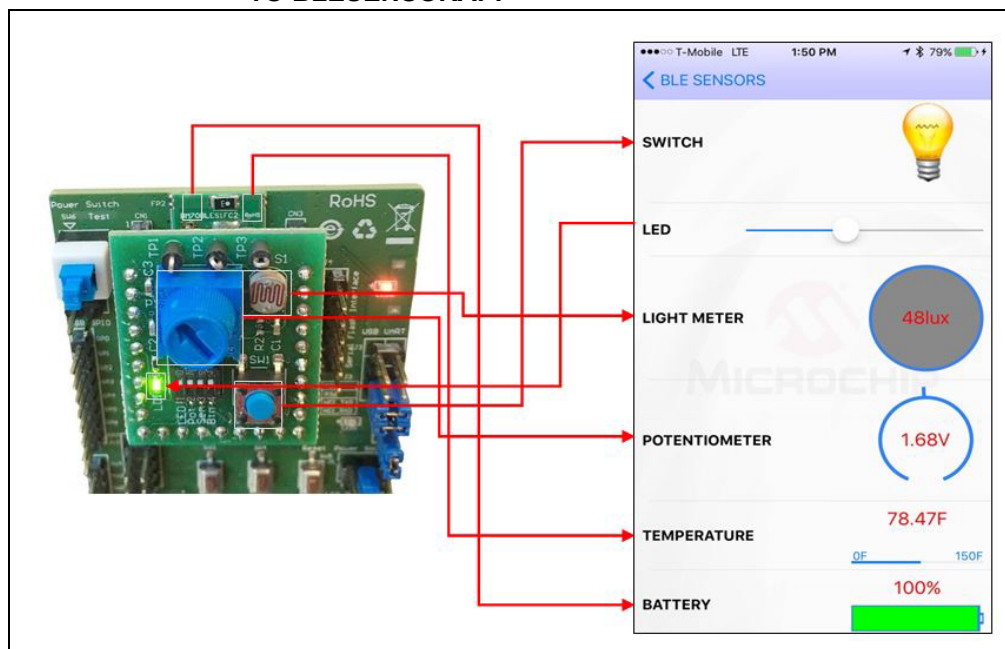


Figure 4-7 illustrates the data flow from the RN4870 Sensor Peripheral IO to the smart-phone app. The data streaming from the Sensor board to the App is performed by GATT Notifications. The script polls the sensor values, such as Switch, Light Meter, and writes the values into the appropriate GATT Characteristic. If the values change, a GATT Notification is sent to the BLESensorApp where the App updates the Graphical User Interface (GUI).

BLE data can also be transferred from the BLESensorApp to the RN4870 module, and later processed by the script to write a received value into the Peripheral output channel. In the Sensor board demo, slider control is used on the App to select a blink rate value. The value is written to the GATT characteristic in the RN4870. The sensor script monitors the Characteristic for incoming data then extracts the value which is subsequently written to the Pulse-With Modulation (PWM) output peripheral to control LED blink rate.

For more information on the BLE GATT data transfers, refer to [Appendix C. "Bluetooth Low Energy Primer"](#).

4.6 PROVISION UTILITY

The RN4870/71 Provision Utility can be used to provision/configure the RN4870/71 Bluetooth Low Energy (BLE) module from a Windows Host PC using a wired connection over Universal Asynchronous Receiver/Transmitter (UART) interface. The RN4870/71 Provision Utility is used to program the RN4870/71 PICtail with the configuration and script to work with the Sensor board and the BLESensorAPP.

The utility comes with a few prebuilt provision options that can be selected using the Provision File drop-down menu. Alternatively, a custom configuration XML file can be imported into the utility to provision the RN4870/71 module.

Detailed instructions for provisioning the RN4870/71 module and creating custom configuration XML file are provided in the RN4870/71 Provision Utility Help files. Examples of the Provision XML file are in the Examples folder.

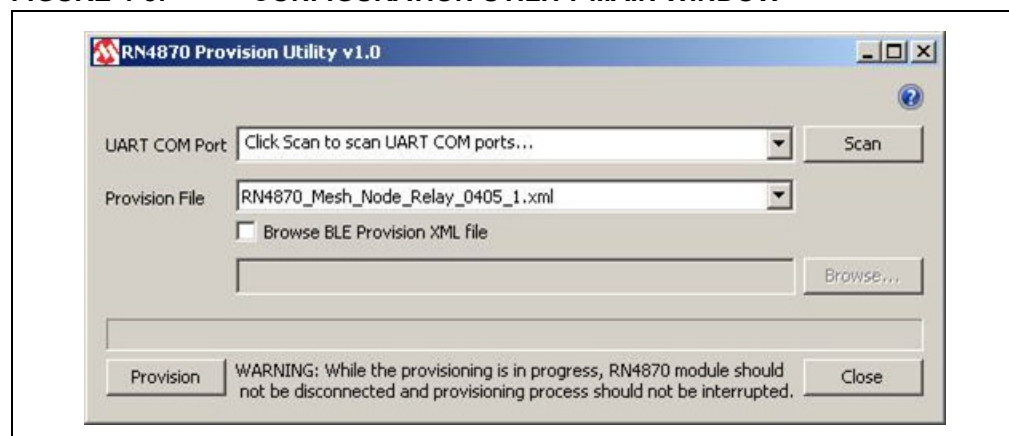
Download the RN4870/71 Provision Utility from www.microchip.com/RN4870. Extract the contents into a separate folder on the host PC. The contents of the extracted folder include the RN4870/71 Provision Utility executable, the RN4870/71 Provision Utility Help files and the example configuration XML files. This utility requires the MCP2200 USB/UART CDC drivers. The driver is also available from the web page. Install the driver if needed.

Note: While the provisioning is in progress, the RN4870/71 must *not* be disconnected and the process is uninterrupted. If the provisioning process is interrupted, the RN4870/71 module may*not* be configured as desired.

To configure an RN4870 PICtail with a Sensor board plugged in, perform the following steps:

1. Ensure that the Sensor board is attached to the RN4870 PICtail as shown in [Figure 1-1](#).
2. Verify the jumpers and switches are positioned according to [Figure 2-1](#) (SW7 must in be in position 1: Application mode)
3. Connect the RN4870 to the host PC using the micro USB cable and verify that LED1 blinks blue with long interval.
4. Wait for the RN4870 board to successfully enumerate as a Serial Port device.
5. After the RN4870 board enumerates as a Serial Port device, open the Device Manager on the Windows PC and note down the COM port number assigned to the RN4870 board under 'Ports'.
6. Run the RN4870 Provision Utility (RN4870ProvisionUtility.exe) on the Windows PC. Refer to [Figure 4-8](#).

FIGURE 4-8: CONFIGURATION UTILITY MAIN WINDOW

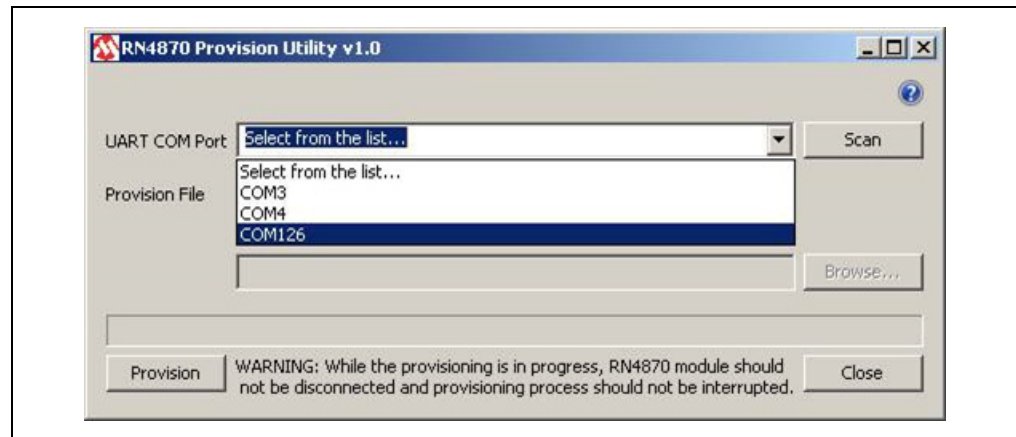


7. Ensure that the RN4870 board is still connected to the Windows PC and successfully enumerated. Click **Scan** button to scan for all the UART COM ports available on the Windows PC.

Note: Ensure that the COM port on which the RN4870 board is enumerated is *not* opened by another application like a Serial Terminal application.

8. Click the drop-down menu of the UART COM Port and select the correct COM port that is assigned to the RN4870 board to be provisioned. Refer to [Figure 4-9](#).

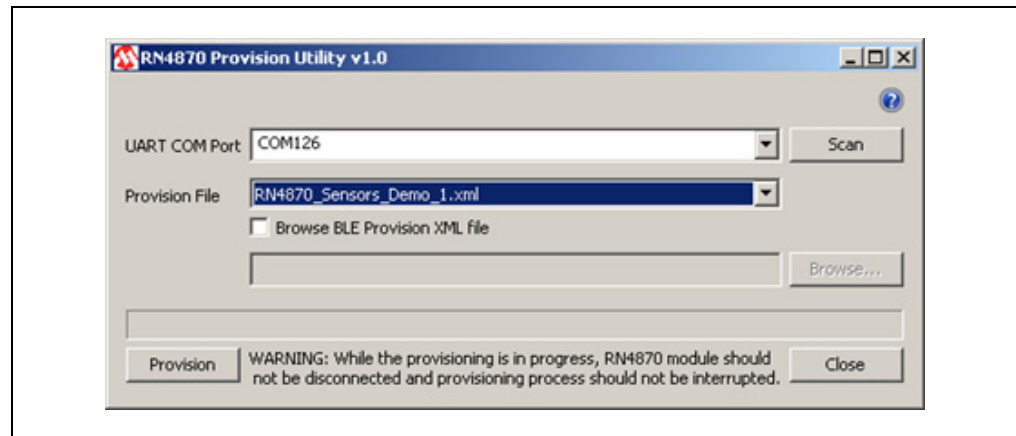
FIGURE 4-9: CONFIGURATION UTILITY: SELECT COM PORT



A set of prebuilt provision options for the RN4870 module are provided. Choose the `RN4870_Sensors_Demo_1.xml` XML file from the Provision File drop-down menu as shown in [Figure 4-10](#).

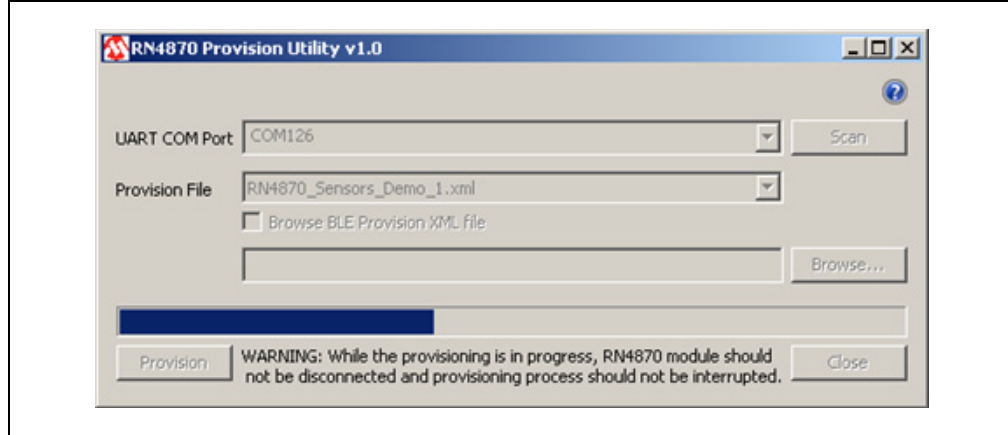
Note: Use only a valid BLE Provision XML (.xml) file.

FIGURE 4-10: CONFIGURATION UTILITY: SELECT XML FILE



9. With the RN4870 board still connected, the COM port and the `RN4870_Sensors_Demo_1.xml` XML file selected, click **Provision** button to start the RN4870 provisioning process. The progress bar shows the progress of the provisioning process. Refer to [Figure 4-11](#).

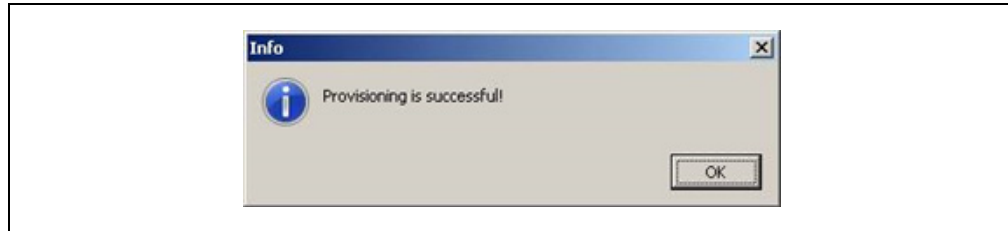
FIGURE 4-11: CONFIGURATION UTILITY: CONFIGURING MODULE



10. After the provisioning process is completed, an information dialog window pops up to inform about the successful provisioning of the RN4870 module. Refer to [Figure 4-12](#).

Note: After the provisioning is successful, the RN4870 automatically reboots and retains the provisioned configuration.

FIGURE 4-12: CONFIGURATION UTILITY: PROVISIONING SUCCESSFUL



Chapter 5. PIC Configuration Library

5.1 OVERVIEW

The BM7x Configuration Library provides a set of functions for the RN4870/71 module to create command packets to perform the following functions:

- Update the Bluetooth parameter table
- Perform pairing procedure
- Configure a subset of parameters in Application mode
- Set the module into different operating modes

The RN4870/71 module uses the UART interface for configuration and data transfer. The RN4870/71 Configuration and Events are defined as Command and Response protocol packets. A Command packet is sent to the RN4870/71 over UART to update a parameter. A Response packet is received from the RN4870/71 over UART for the command issued. RN4870/71 also sends out Event packets over UART when a defined event occurs. The PICtail Plus and PICtail interface on the RN4870/71 PICtail board are used to configure the RN4870/71 module over the UART using the BM7x Configuration Library with an external PIC MCU and to send and receive raw data over UART. The PICtail interface can also be used to access the UART and the GPIOs on the RN4870/71 module.

5.2 USING BM7X CONFIGURATION LIBRARY WITH PICtail BOARD

The RN4870/71 PICtail plugs into the Explorer 16 Development board with a PIC32/PIC24 Plug-In-Module (PIM) or into the PIC18 Explorer board with a PIC18 PIM. The BM7x Configuration Library demo is an example application that can be programmed in to the PIC32/PIC24/PIC18 PIMs to configure and control the RN4870/71 module on the PICtail board.

5.2.1 Download/Install

1. Download the BM7x Configuration Library installer from www.microchip.com/RN4870
2. Install the Library on a Windows PC.
3. Refer to the `getting_started.htm` file for more information on the MPLAB®X workspace.

5.2.2 Programming/Debugging

1. Plug the PIC Plug-In-Module (PIM) into the relevant Explorer Development board based on the hardware combination selected for evaluation.

Note: If using PIC32MX795F512L PIM, ensure that on jumpers J1 and J2 on the PIM, pins 2 and 3 are connected using a jumper to select the CAN configuration and all jumpers on J9 and J10 on the PIM are *not* connected.

2. Plug the RN4870/71 PICtail Plus board into the Explorer Development board with the RN4870/71 module facing towards the PIC PIM as shown in [Figure 5-1](#) or [Figure 5-2](#) based on hardware combination used.

3. Optionally, a debug UART port can be connected to a PC terminal emulator program.

Note: On the Explorer 16 Development board, the debug UART is available on the DB9 UART serial connector P1. On the PIC18 Explorer board, the debug UART needs to be manually tapped from pins RG1/TX2 and RG2/RX2 on the J5 header using external wiring.

4. Provide power through the 9V power input port available on the Explorer Development board.

FIGURE 5-1: EXPLORER 16 DEVELOPMENT BOARD WITH RN4870 PICtail PLUS

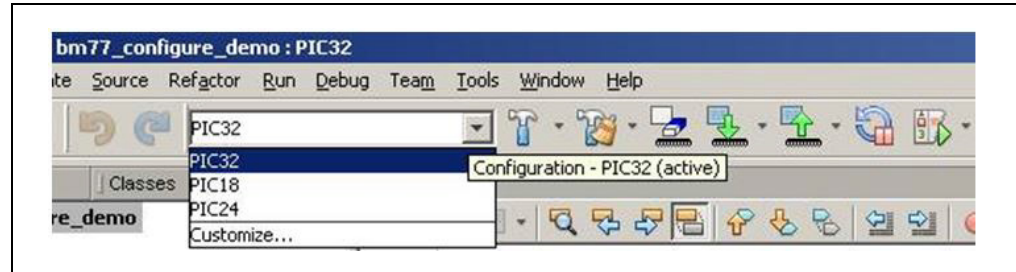


FIGURE 5-2: EXPLORER 16 DEVELOPMENT BOARD WITH RN4870 PICtail PLUS



5. Program the PIC32 with the `bm7x_configure_demo_xc32.hex` or the PIC24 with the `bm7x_configure_demo_xc16.hex` or the PIC18 with the `bm7x_configure_demo_xc8.hex` file in the `precompiled_hex` sub-directory.
6. Alternatively, open the `bm7x_configure_demo.X` MPLAB X workspace using the MPLAB X IDE to compile and program; or enter Debug mode and select either the PIC32, the PIC24 or the PIC18 workspace configuration based on the hardware combination as shown in [Figure 5-3](#).

FIGURE 5-3: SELECT WORKSPACE CONFIGURATION



7. Change the `#define BMXX_DEVICE xyz` in `main.c` to `BM70_BLUETOOTH_DEVICE` to work with RN4870.

5.2.3 Connecting/Running Demo

After setting up the hardware combination and programming the PIC, perform the following the steps to run the demo:

1. Observe the RN4870/71 configuration by inserting break points into the workspace or by optionally using the debug UART port.
2. Download and install the `BtChat_V1.0.3.apk` file on the Android device from the BM7x PICTail Plus web page.
3. Open the BtChat application installed by the `BtChat_V1.0.3.apk` on the Android device.
4. Scan for the Bluetooth devices on the Android app.
5. Select the BM7x device listed in the scan list and click to pair and connect.
6. The application uses Just Works pairing by default. If Passkey Entry or Passkey Yes/No Confirm mode is selected, provide the passkey or confirmation using the debug UART port. For the RN4870 module's supported pairing modes, refer to the "*Bluetooth® Low Energy Module User's Guide*" (DS50002466).
7. Once connected, the data can be transmitted from the RN4870/71 through the debug UART port to the BtChat app over Bluetooth link.
8. Enable Show Rx Text option in the BtChat app setting. The BtChat app can resend data to the RN4870/71 that is later received through the debug UART.

NOTES:

Appendix A. Updating PICtail Firmware

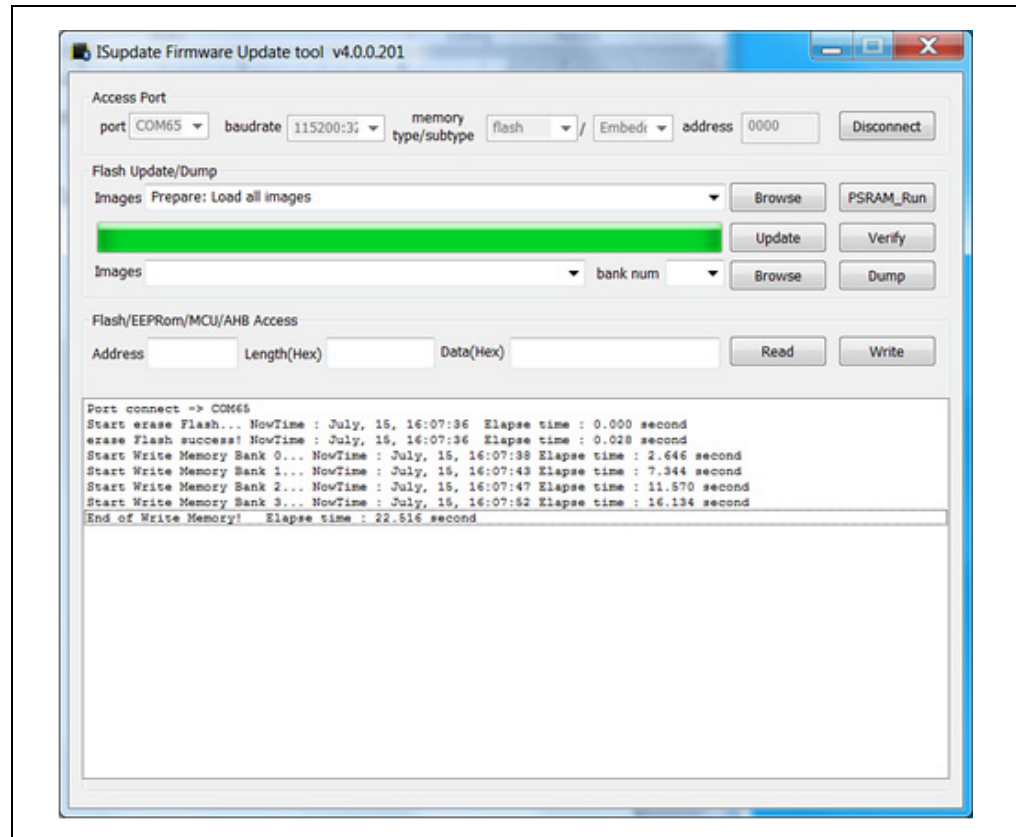
A.1 OVERVIEW

Firmware for the RN4870/71 PICtail can be updated using a PC Tool `isupdate.exe` over the USB port. The latest RN4870/71 firmware images and the `isupdate.exe` tool are available from the product web page at www.microchip.com/RN4870.

To update the firmware on the RN4870 PICtail, perform the following steps:

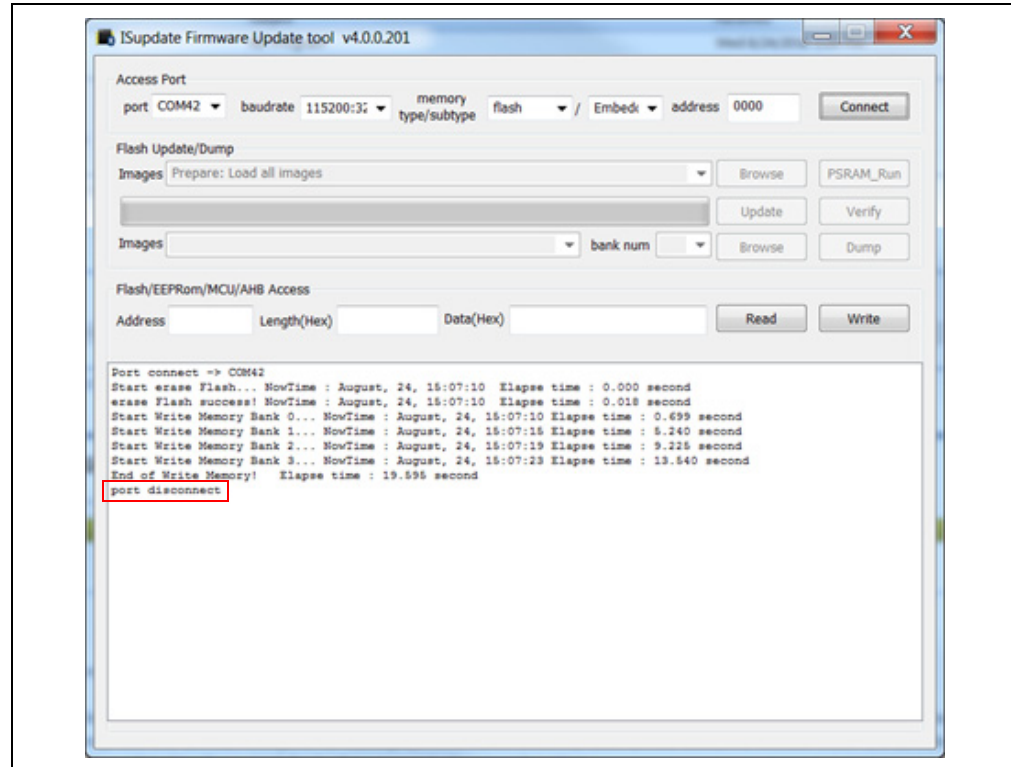
1. Download the firmware zip file from the product web page and extract the contents. The zip file contains the `isupdate.exe` utility and a folder including the firmware images.
2. Connect the PICtail to the host PC using the micro USB cable.
3. Verify that SW7 is set to ON position.
4. Press SW5 **Reset** button and verify that LED1 is showing a non-flashing blue that indicates the RN487x module is in programming mode.
5. Launch the `isupdate.exe` application as shown in:
 - Select the COM port used by the PICtail board
 - Verify that other settings (baud, memory type, and address) are set as shown in [Figure A-1](#).

FIGURE A-1: ISUPDATE.EXE FIRMWARE UPDATE UTILITY



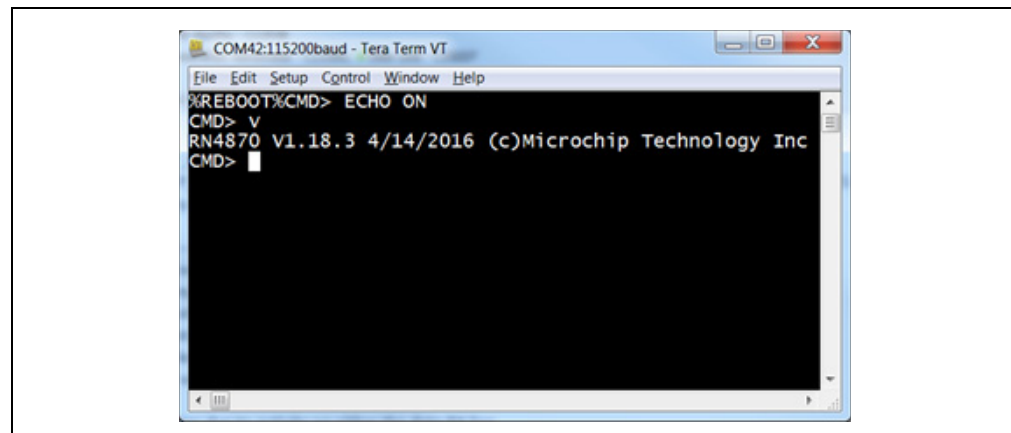
6. Click the **Connect** button and verify the “Port connect -> COMxx” is displayed in the text box.
7. Verify firmware update is successfully completed. The “End of Write Memory” message is displayed.
8. Click the **Disconnect** button to close the COM port. The “port disconnect” message is displayed as indicated in [Figure A-2](#).

FIGURE A-2: DISCONNECT ISUPDATE.COM FROM MODULE



9. Set SW7 to the '1' position for Application mode.
10. Open TeraTerm and connect to the module.
11. Press SW5 to reboot the module. Ensure that the %REBOOT% message is displayed.
12. Enter Command mode by sending \$\$\$ escape sequence.
13. Enter the v command and verify firmware version as illustrated in [Figure A-3](#).

FIGURE A-3: VERIFY FIRMWARE VERSION





Appendix B. Schematics and BOM

B.1 INTRODUCTION

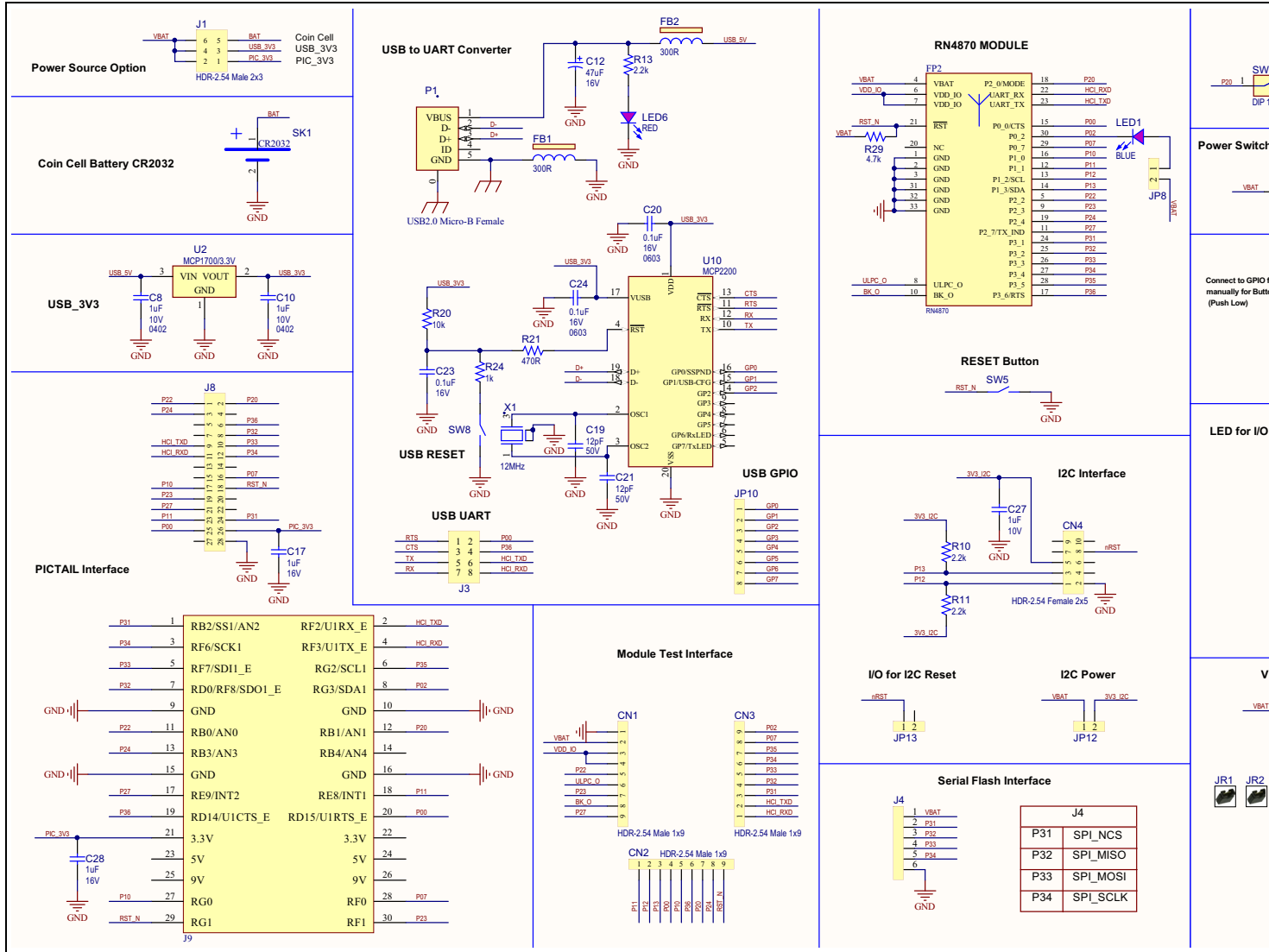
This appendix provides the schematics and the Bill of Materials (BOM) for the RN487x PICtail™/PICtail Plus Daughter board:

- [RN4870 PICtail Schematic](#)
- [RN4870 PICtail BOM](#)
- [RN4871 PICtail Schematic](#)
- [RN4871 PICtail BOM](#)
- [RN4870 Sensor Board Schematic](#)
- [RN4870 Sensor Board BOM](#)

B.2 RN4870 PICtail SCHEMATIC

Figure B-1 shows the schematic for the RN4870 PICtail™/PICtail Plus Daughter board.

FIGURE B-1: RN4870 PICtail SCHEMATIC



B.3 RN4870 PICtail BOM

TABLE B-1: RN4870 PICtail BOM

Oty	Reference	Description	Manufacturer	Manufacturer Part Number
3	C8, C10, C27	CAP CER 1 μ F 10V 10% X5R SMD 0402	Murata Electronics North America	GRM155R61A105KE15D
1	C12	CAP TANT 47 μ F 16V 10% 0.11R SMD C	AVX Corporation	TPSC476K016R0110
2	C17, C28	CAP CER 1 μ F 16V 10% X7R SMD 0603	TDK	C1608X7R1C105K
2	C19, C21	CAP CER 12pF 50V 1% NP0 SMD 0402	Murata Electronics North America	GRM1555C1H120FA01D
3	C20, C23, C24	CAP CER 0.1 μ F 16V 10% X7R SMD 0603	AVX	0603YC104KAT2A
3	CN1, CN2, CN3	CON HDR-2.54 Male 1x9 Gold 5.84MH TH VERT	Samtec	TSW-109-07-G-S
1	CN4	CON HDR-2.54 Female 2x5 GOLD TH R/A	Samtec	SSQ-105-02-G-D-RA
2	FB1, FB2	FERRITE 300R@100 MHz 2A SMD 0805	Laird-Signal Integrity Products	MI0805L301R-10
1	J1	CON HDR-2.54 Male 2x3 Gold 5.84MH TH VERT	Samtec	TSW-103-08-L-D
3	J2, J3, J10	CON HDR-2.54 Male 2x4 Gold 5.84MH TH VERT	Samtec	TSW-104-08-L-D
1	J4	CON HDR-2.54 Male 1x6 Gold 5.84MH TH VERT	FCI	68001-106HLF
1	J8	CON HDR-2.54 Male 2x14 Gold 5.84MH TH R/A	Sullins	PBC14DBDN
2	JP5, JP7	CON HDR-2.54 Male 1x4 Gold 5.84MH TH VERT	Samtec	TSW-104-07-G-S
4	JP6, JP8, JP12, JP13	CON HDR-2.54 Male 1x2 Gold 5.84MH TH VERT	FCI	77311-118-02LF
1	JP10	CON HDR-2.54 Male 1x8 Gold 5.84MH TH	FCI	68001-108HLF
1	LED1	DIO LED BLUE 2.8V 20 mA 15mcd Clear SMD 0603	Lite-On	LTST-C193TBKT-5A
4	LED2, LED3, LED4, LED5	DIO LED YELLOW 2.1V 20 mA 6 mcd Clear SMD 0603	Lite-On	LTST-C190YKT
1	LED6	DIO LED RED 1.8V 40 mA 10 mcd Clear SMD 0603	Lite-On	LTST-C190KRKT
1	P1	CON USB2.0 Micro-B Female SMD R/A	FCI	10118193-0001LF
4	R2, R26, R27, R28	RES TKF 330R 1% 1/16W SMD 0402	KOA Speer	RK73H1ETTP3300F
2	R8, R25	RES TKF 4.7k 5% 1/10W SMD 0603	Panasonic	ERJ-3GEYJ472V
3	R10, R11, R13	RES TKF 2.2k 1% 1/10W SMD 0402	Panasonic	ERJ-2RKF2201X
1	R20	RES TKF 10k 5% 1/8W SMD 0805	Panasonic	ERJ-6GEYJ103V
1	R21	RES TKF 470R 5% 1/8W SMD 0805	Panasonic	ERJ-6GEYJ471V

RN4870/71 PICtail™/PICtail Plus Daughter Board User's Guide

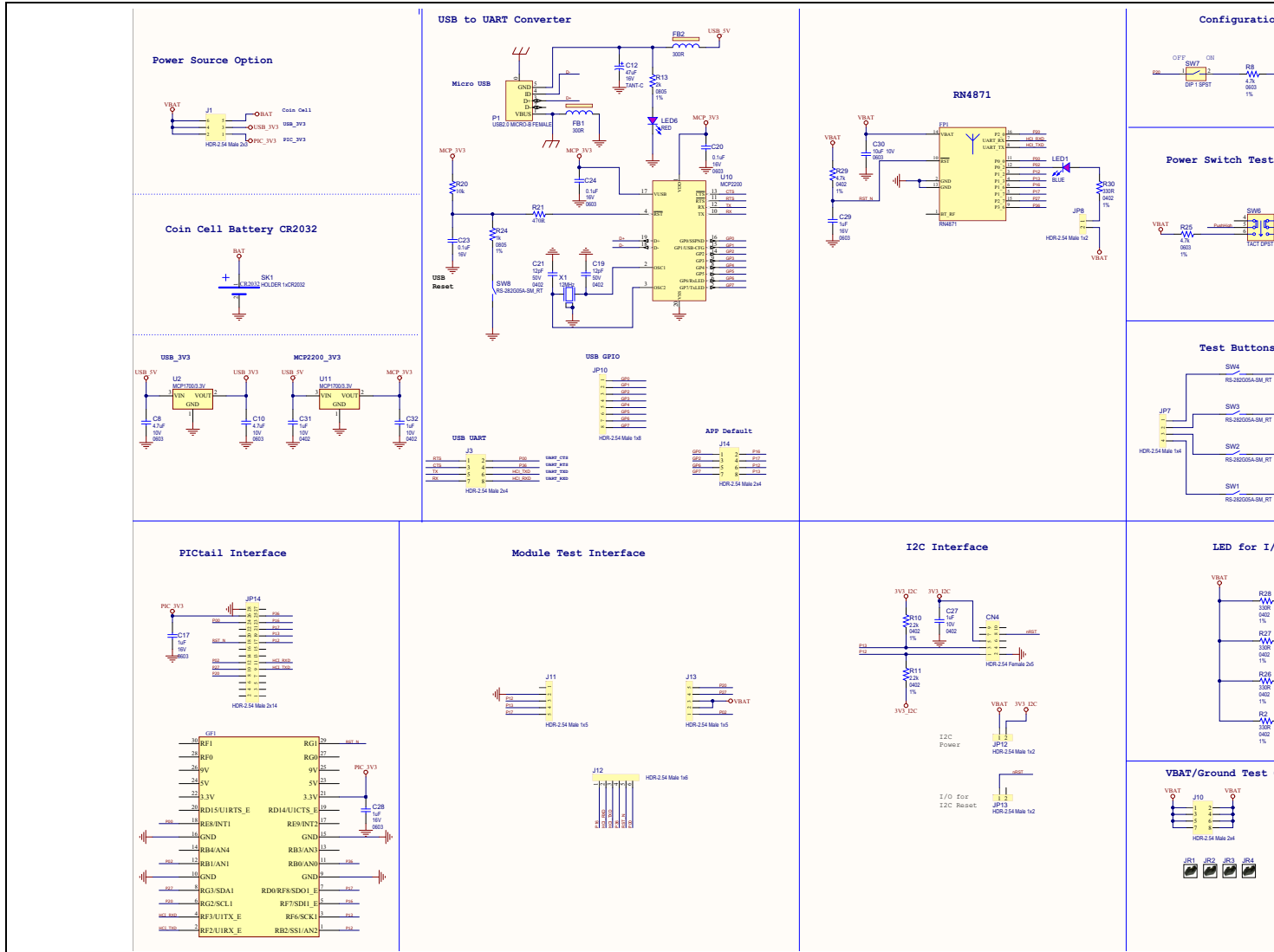
TABLE B-1: RN4870 PICtail BOM

Oty	Reference	Description	Manufacturer	Manufacturer Part Number
1	R24	RES TKF 1k 5% 1/8W SMD 0805	Panasonic	ERJ-6GEYJ102V
1	R29	RES TKF 4.7K 1% 1/16W 0402	KOA Speer	RK73H1ETTP4701F
1	SK1	BATT HOLDER COIN 1 TH CR2450	Keystone Electronics	1053
6	SW1, SW2, SW3, SW4, SW5, SW8	SWITCH TACT SPST 12V 50mA RS-282G05A3-SM RT	C&K Components	RS-282G05A3-SM RT
1	SW6	SWITCH TACTILE PUSH ON-OFF 6 PIN	ROKI ELECTRONICS CO., LTD.	PS07-22L-PF
1	SW7	SWITCH DIP 1 SPST 24V 25 mA 418117270901 SMD	Würth Electronics Inc	4,18117E+11
1	X1	CRYSTAL 12MHz 8 pF SMD NX3225SA	NDK	NX3225SA-12.000000MHZ
1	FP2	MCHP RF BLUETOOTH RN4870-V/RM118 MODULE-33	Microchip Technology Inc.	RN4870-V/RM118
1	U2	MCHP ANALOG LDO 3.3V MCP1700T-3302E/TT SOT-23-3	Microchip Technology Inc.	MCP1700T-3302E/TT
1	U10	MCHP INTERFACE USB UART MCP2200-I/SS SSOP-20	Microchip Technology Inc.	MCP2200-I/SS-ND

B.4 RN4871 PICtail SCHEMATIC

Figure B-2 shows the schematic for the RN4871 PICtail™/PICtail Plus Daughter board.

FIGURE B-2: RN4871 PICtail SCHEMATIC



B.5 RN4871 PICtail BOM

TABLE B-2: RN4871 PICTAIL BOM

Oty	Reference	Description	Manufacturer	Manufacturer Part Number
2	C8, C10	CAP CER 4.7 μ F 10V 10% X5R SMD 0603	KEMET	C0603C475K8PACTU
1	C12	CAP TANT 47 μ F 16V 10% 0.11R SMD C	AVX Corporation	TPSC476K016R0110
3	C17, C28, C29	CAP CER 1 μ F 16V 10% X7R SMD 0603	TDK	C1608X7R1C105K
2	C19, C21	CAP CER 12pF 50V 1% NP0 SMD 0402	Murata Electronics North America	GRM1555C1H120FA01D
3	C20, C23, C24	CAP CER 0.1 μ F 16V 10% X7R SMD 0603	AVX	0603YC104KAT2A
3	C27, C31, C32	CAP CER 1 μ F 10V 10% X5R SMD 0402	Murata Electronics North America	GRM155R61A105KE15D
1	C30	CAP CER 10 μ F 10V 20% X5R SMD 0603	Panasonic	ECJ-1VB1A106M
1	CN4	CON HDR-2.54 Female 2x5 GOLD TH R/A	Samtec	SSQ-105-02-G-D-RA
2	FB1, FB2	FERRITE 300R@100 MHz 2A SMD 0805	Laird-Signal Integrity Products	MI0805L301R-10
1	J1	CON HDR-2.54 Male 2x3 Gold 5.84MH TH VERT	Samtec	TSW-103-08-L-D
4	J2, J3, J10, J14	CON HDR-2.54 Male 2x4 Gold 5.84MH TH VERT	Samtec	TSW-104-08-L-D
2	J11, J13	CON HDR-2.54 Male 1x5 Tin 5.84MH TH VERT	Samtec	TSW-105-07-T-S
1	J12	CON HDR-2.54 Male 1x6 Gold 5.84MH SMD VERT	Samtec	TSM-106-01-L-SV
2	JP5, JP7	CON HDR-2.54 Male 1x4 Gold 5.84MH TH VERT	Samtec	TSW-104-07-G-S
4	JP6, JP8, JP12, JP13	CON HDR-2.54 Male 1x2 Gold 5.84MH TH VERT	FCI	77311-118-02LF
1	JP10	CON HDR-2.54 Male 1x8 Gold 5.84MH TH	FCI	68001-108HLF
1	JP14	CON HDR-2.54 Male 2x14 Gold 5.84MH TH R/A	Sullins	PBC14DBDN
1	LED1	DIO LED BLUE 2.8V 20 mA 15 mcd Clear SMD 0603	Lite-On	LTST-C193TBKT-5A
4	LED2, LED3, LED4, LED5	DIO LED YELLOW 2.1V 20 mA 6 mcd Clear SMD 0603	Lite-On	LTST-C190YKT
1	LED6	DIO LED RED 1.8V 40 mA 10 mcd Clear SMD 0603	Lite-On	LTST-C190KRKT
1	P1	CON USB2.0 MICRO-B FEMALE TH/SMD R/A	FCI	10118194-0001LF
5	R2, R26, R27, R28, R30	RES TKF 330R 1% 1/10W SMD 0402	KOA Speer	RK73H1ETTP3300F
2	R8, R25	RES TKF 4.7k 1% 1/16W SMD 0603	SPC Technology	MC0603WGF4701T5E-TC
2	R10, R11	RES TKF 2.2k 1% 1/10W SMD 0402	Panasonic	ERJ-2RKF2201X

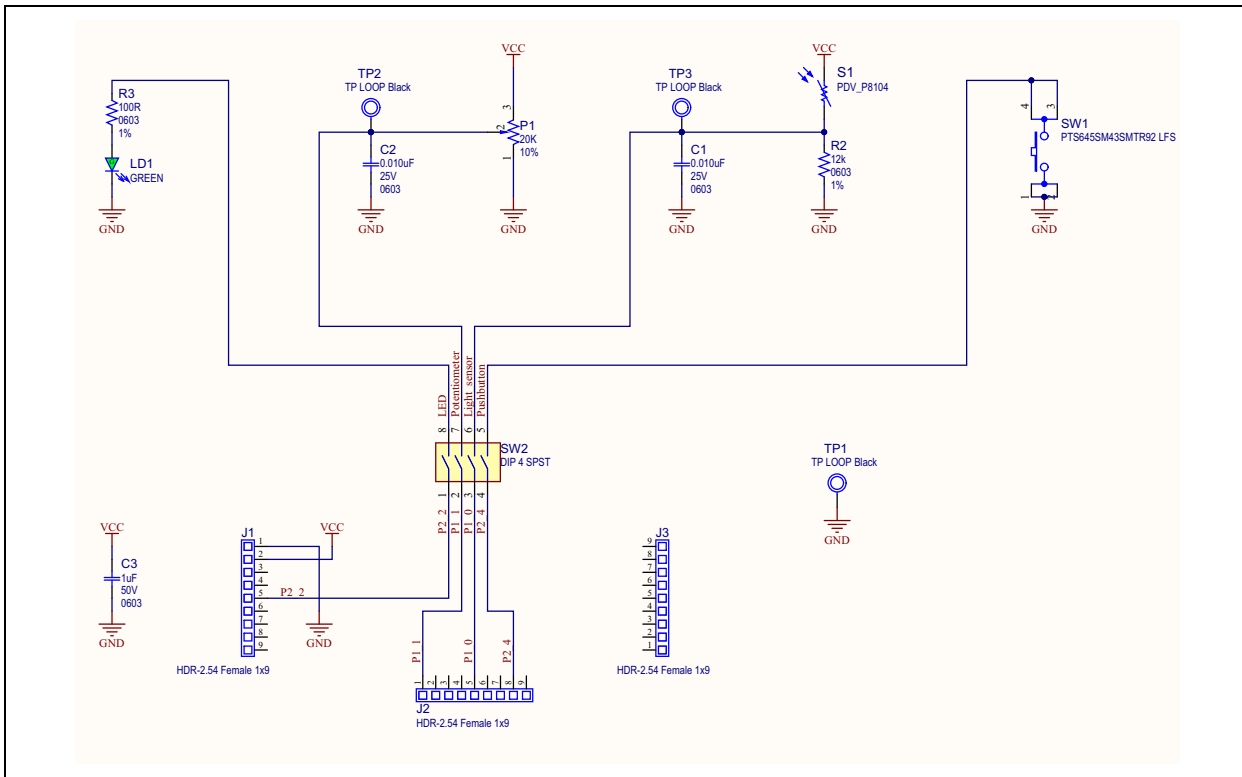
TABLE B-2: RN4871 PICTAIL BOM

Oty	Reference	Description	Manufacturer	Manufacturer Part Number
1	R13	RES TKF 2k 1% 1/8W SMD 0805	Panasonic	ERJ-6ENF2001V
1	R20	RES TKF 10k 1% 1/8W SMD 0805	Panasonic	ERJ-6ENF1002V
1	R21	RES TKF 470R 5% 1/8W SMD 0805	Panasonic	ERJ-6GEYJ471V
1	R24	RES TKF 1k 1% 1/8W SMD 0805	Vishay	CRCW08051K00FKEA
1	R29	RES TKF 4.7K 1% 1/10W 0402	KOA Speer	RK73H1ETTP4701F
1	SK1	BATT HOLDER TH CR2032	MPD	BH32T-C
6	SW1, SW2, SW3, SW4, SW5, SW8	SWITCH TACT SPST 12V 50 mA RS-282G05A3-SM RT	C&K Components	RS-282G05A3-SM RT
1	SW6	SWITCH TACTILE PUSH ON-OFF 6 PIN	ROKI ELECTRONICS CO., LTD.	PS07-22L-PF
1	SW7	SWITCH DIP 1 SPST 24V 25 mA 418117270901 SMD	Würth Electronics Inc	418117270901
1	X1	CRYSTAL 12 MHz 8 pF SMD NX3225SA	NDK	NX3225SA-12.000000MHZ
1	FP2	MCHP RF BLUETOOTH RN4871 Microchip Bluetooth module	Microchip Technology Inc.	RN4871-V/RM118
1	U2, U11	MCHP ANALOG LDO 3.3V MCP1700T-3302E/TT SOT-23-3	Microchip Technology Inc.	MCP1700T-3302E/TT
1	U10	MCHP INTERFACE USB UART MCP2200-I/SS SSOP-20	Microchip Technology Inc.	MCP2200-I/SS-ND

B.6 RN4870 SENSOR BOARD SCHEMATIC

Figure B-3 shows the schematic for the RN4870 Sensor board.

FIGURE B-3: RN4870 SENSOR BOARD SCHEMATIC



B.7 RN4870 SENSOR BOARD BOM

TABLE B-3: RN4870 SENSOR BOARD BOM

Oty	Reference	Description	Manufacturer	Manufacturer Part Number
2	C1, C2	CAP CER 0.010 μ F 25V 10% X7R SMD 0603	Yageo	CC0603KRX7R8BB103
1	C3	CAP CER 1 μ F 50V 10% X5R SMD 0603	Taiyo Yuden	LMK107BJ105KA-T
3	J1, J2, J3	CON HDR 2.54 Female 1x9 Gold 7MH TH VERT	Preci-Dip	801-87-009-10-001101
1	LD1	DIO LED GREEN 2V 30 mA 35 mcd Clear SMD 0603	Lite-On Inc.	LTST-C191KGKT
1	P1	RES Variable CC 20K 10% 1/2W TH 3386P1-xxxT Knob	Bourns Inc.	3386P-1-203TLF
1	R2	RES TKF 12k 1% 1/10W SMD 0603	Yageo	RC0603FR-0712KL
1	R3	RES TKF 100R 1% 1/10W SMD 0603	Panasonic	ERJ-3EKF1000V
1	S1	SENSOR PHOTOCCELL 27-60KOHM	Advanced Photonix Inc.	PDV-P8104
1	SW1	SWITCH TACT SPST 12V 50 mA PTS645SM43SMTR92 LFS SMD	C&K Components	PTS645SM43SMTR92 LFS
1	SW2	SWITCH DIP 4-POS SLIDE SMD 6V	Copal Electronics Inc.	CHS-04TB
3	TP1, TP2, TP3	MISC, TEST POINT MULTI PURPOSE MINI BLACK	Keystone	5001

Appendix C. Bluetooth Low Energy Primer

C.1 GAP ROLES: PERIPHERAL AND CENTRAL

When two Bluetooth Low Energy (BLE) devices want to be connected, one device must be in Central role and the other in Peripheral role. The Peripheral device advertises to show its connectable status, while Central device scans service advertisements, and if needed, initiates a connection to the Peripheral device. Once connected, either end of the connection can choose to bond. Once bonded, all security related keys are saved and security process are waived when reconnecting. Bonded peripheral device can only perform direct advertise, therefore, unable to connect to a device other than its bonded peer.

C.2 GATT SERVICE: CLIENT AND SERVER

Similar to Bluetooth Classic, BLE uses the concept of profiles to ensure interoperability between different devices. As illustrated in [Figure C-1](#), BLE profiles are collection of services. All BLE services are built on top of Generic Attribute Profile (GATT) where GATT defines accessibility of attributes called characteristics. The main functionality of the BLE profiles, therefore, is built around the characteristics. For those devices that maintain the value of characteristics in a service, such device is the *Server* of the service. On the other hand, devices that acquire data from their peer are called *Client*.

Each service and its characteristics can be identified by their Universally Unique Identifier (UUID). The UUID can be a short form (16-bit) or long form (128-bit). As specified by Bluetooth Core Specifications, all Bluetooth SIG adopted (public) services and characteristics have short UUID (16-bit), while user defined private UUIDs are in long form (128-bit). For the details of Bluetooth SIG adopted services and characteristics, refer to <https://developer.bluetooth.org/gatt/profiles/Pages/ProfilesHome.aspx>.

The accessibility of each characteristic is defined by a 8-bit characteristic property in bitmap format as shown in [Table C-1](#).

TABLE C-1: CHARACTERISTIC PROPERTIES

Property	Bitmap Value	Description
Extended Property ⁽¹⁾	0x80	Additional property available
Authenticated Write ⁽¹⁾	0x40	Write characteristic with authentication from GATT client to GATT server
Indicate	0x20	Indicate value of characteristic with acknowledgment from GATT server to GATT client
Notify	0x10	Notify value of characteristic without acknowledgment from GATT server to GATT client
Write	0x08	Write value of characteristic with acknowledgment from GATT client to GATT server
Write without response	0x04	Write value of characteristic without acknowledgment from GATT client to GATT server
Read	0x02	Read value of characteristic. Value is sent from GATT server to GATT client
Broadcast ⁽¹⁾	0x01	Broadcast value of characteristic

Note 1: These features are supported in future firmware releases

As shown in Figure C-1, the GATT client can access the characteristics in the GATT server in the Peripheral device. When connected, the GATT client reads the GATT server service and characteristic UUIDs. The characteristic values can be accessed by the GATT client using Write, Read, Indication and Notifications.

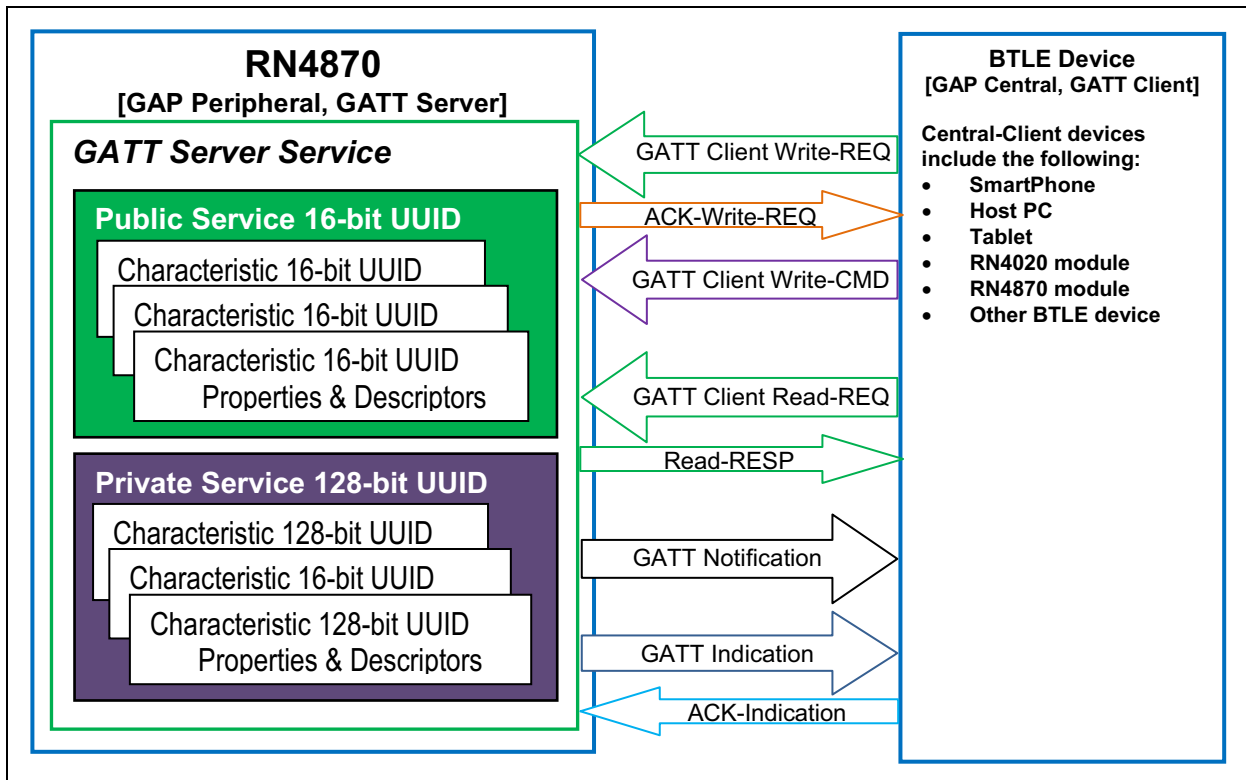
Write-REQ enables the GATT client to update characteristic values on the Peripheral's GATT server. The write requests can be performed using RN4870 CHW and CUW commands. Refer to Section 3.6 "Accessing GATT Service Using UART Commands and SmartDiscover App" for more information on GATT characteristic access commands.

A Write-CMD message performs an unacknowledged write from a GATT client to the GATT server. This is allowed for Transparent UART Write procedure on the RN4870 when SR, 0100 command is used.

A GATT client sends Read-REQ read a characteristic value on the Peripheral's GATT server. The read requests can be performed using RN4870 CHR and CUR commands.

Notifications and Indications are unsolicited updates sent from the GATT server to the GATT client. The GATT client must enable the Notification and Indication on a characteristic to receive the updates. On the RN487x module, this is done by using the CUW or CHW command to write non-zero value to the Notification Characteristic. When RN487x in Client mode receives a notification, the %WC, hhhh, dddddd% message is returned on UART in Command mode.

FIGURE C-1: GATT SERVICE IN



Appendix D. Sensor Board Configuration Command Text

D.1 OVERVIEW

This appendix provides an example for configuration commands to use all the sensors in the Sensor board in a GATT service.

- [RN4870 Module Settings](#)
- [RN4870 Sensor Board GATT Service](#)
- [Sensor Board Script](#)

D.1.1 RN4870 Module Settings

```
WP
WC
PZ
S-,RN4870
SR,4040
SS,80
SW,01,00
SW,02,00
SW,03,00
SW,04,09
R,1
```

D.1.2 RN4870 Sensor Board GATT Service

```
PZ
PS,AD11CF40063F11E59E690002A5D5C51B
PC,BF3FBD80063F11E59E690002A5D5C501,10,02
PC,BF3FBD80063F11E59E690002A5D5C502,10,02
PC,BF3FBD80063F11E59E690002A5D5C503,18,14
PC,BF3FBD80063F11E59E690002A5D5C504,10,02
PC,BF3FBD80063F11E59E690002A5D5C505,10,02
R,1
```

D.1.3 Sensor Board Script

To configure the Sensor board, enter the following sequence of commands:

```
@PW_ON
IA,Z
SM,2,0000
SM,1,0002
%0078=?FUNC1
?FUNC1
[,1,1,$PM1,$PM2
@DISCON
SM,1,0000
SM,2,0000
SM,3,0000
R,1
@CONN
SM,1,0000
SM,2,0000
SM,3,001A
@TMR1
IA,Z
IA,FF,CD00FE14AD11CF40063F11E5BE3E0002A5D5C51B000C000D
SM,2,000F
@TMR2
IB,Z
IB,FF,4C000215AD11CF40063F11E5BE3E0002A5D5C51B000C000DC5
SM,1,0004
@TMR3
$VAR1=@,0
SHW,0072,$VAR1
$VAR1=@,1
SHW,0075,$VAR1
$VAR1=@,5
SHW,007B,$VAR1
$VAR1=@,4
SHW,007E,$VAR1
SM,3,001A
@PIO1L
SHW,0078,0001
@PIO1H
SHW,0078,0000
```


X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Bluetooth Development Tools - 802.15.1 category](#):

Click to view products by [Microchip manufacturer](#):

Other Similar products are found below :

[DA14580PRODTLKT 1628](#) [SP14808ST MBH7BLZ02-EF-KIT](#) [FWM7BLZ20-EB-KIT](#) [SP14801-DUT](#) [SKY66111-21EK1](#) [SECO-RSL10-TAG-GEVB](#) [ENW89857AXKF 3026](#) [MIKROE-2471](#) [MOD-NRF8001](#) [BLE-IOT-GEVB 450-0184](#) [EKSHCNZXZ](#) [EVAL_PAN1026](#)
[EVAL_PAN1720](#) [EVAL_PAN1740](#) [2267](#) [2479](#) [2487](#) [2633](#) [STEVAL-IDB005V1D](#) [STEVAL-IDB001V1](#) [MIKROE-2545](#) [SIPKITSLF001](#)
[2995](#) [STEVAL-IDB007V1M](#) [2829](#) [DFR0267](#) [DFR0296](#) [DFR0492](#) [TEL0073](#) [BM-70-CDB](#) [WSM-BL241-ADA-008DK](#) [STEVAL-BTDP1](#)
[ACD52832](#) [TEL0095](#) [ISP1507-AX-TB](#) [RN-4871-PICTAIL](#) [DA14695-00HQDEVKT-P](#) [DA14695-00HQDEVKT-U](#) [EVK-NINA-B112](#)
[EBSHJNZXZ](#) [EKSGJNZWY](#) [EKSHJNZXZ](#) [BMD-200-EVAL-S](#) [ACN BREAKOUT BOARD](#) [ACN SKETCH](#) [2269](#)