



RN-131/171-PICTAIL-UM

RN-131-PICTAIL & RN-171-PICTAIL Evaluation Boards

© 2012 Roving Networks. All rights reserved.
RN-131/171-PICTAIL-UM Version 1.0 9/7/2012

USER MANUAL

OVERVIEW

The RN-131 and RN-171 WiFly radio modules are complete, standalone wireless LAN access devices. Each module contains a TCP/IP stack and related applications. After the module is configured, the radio can access the Wi-Fi network automatically, and transmit and receive data over a UART.

General Description

The RN-131-PICTAIL and RN-171-PICTAIL evaluation boards work seamlessly with the Microchip Technology PIC18 Explorer Development Board. The evaluation boards plug into the PICTail™ modular connector on the PIC18 Explorer Board, and add wireless functionality to PIC18-based microcontroller designs.

The evaluation boards are preloaded with firmware to simplify integration and minimize application development. In the simplest configuration, the hardware only requires four connections (PWR, TX, RX, and GND) to create a wireless data connection. The microcontroller can communicate with the boards via the UART and SPI interfaces, and can drive LEDs, wake the boards, and reset them. The evaluation boards are updated and controlled with a simple ASCII command language. Once the boards are set up, they can scan to find an access point, associate, authenticate, and connect over any Wi-Fi network.

The PIC18 Explorer Board (DM183032) is a development platform equipped with a 28-pin PICTail connector and PIM that can accommodate various versions of the 8-bit PIC16 and PIC18 microcontroller families. The PIC18 Explorer Board's 8-bit microcontroller communicates with the RN-131-PICTAIL/RN-171-PICTAIL evaluation board via the PICTail connector.

This document describes how to use the RN-131-PICTAIL/RN-171-PICTAIL evaluation board connected to a PIC18 Explorer Board to associate with a Wi-Fi network and communicate with other devices within a LAN. Communication with other LAN devices is performed using telnet.

RN-131-PICTAIL/RN-171-PICTAIL Module Features

- FCC/CE/IC certified 2.4-GHz IEEE 802.11b/g transceiver
- Plugs into the PICTail Plus connector on the PIC18 Explorer and Explorer 16 Development Boards
- Adds wireless capability to designs targeting the PIC18 Explorer and Explorer 16 boards
- Configurable transmit power: 0 to 10 dBm (RN-171-PICTAIL)
- PCB trace antenna (RN-171-PICTAIL) and on-board ceramic chip antenna (RN-131-PICTAIL)
- Ultra-low power
 - 4-uA sleep, 38-mA Rx, 120-mA Tx at 0 dBm (RN-171-PICTAIL)
 - 4 uA sleep, 40 mA Rx, 210 mA Tx (RN-131-PICTAIL)
- High throughput: 921 Kbps TX, 500 Kbps RX data rate with TCP/IP and WPA2 over UART, up to 2 Mbps over SPI slave
- UART interface with hardware flow control
- SPI interface with slave interrupt
- Real-time clock for wakeup and time stamping
- 3 status LEDs (2 configurable)
- Powered by PIC18 Explorer and Explorer 16 boards
- Supports ad hoc and infrastructure networks
- Wi-Fi Alliance certified for WPA2-PSK
- Complete on-board TCP/IP networking stack
- Environmentally friendly: RoHS compliant

Demo Applications

The example applications provided with this document include an MPLAB X project and 2 pre-compiled **.hex** files.

- The MPLAB X project files provide a sample 8-bit application framework that you can customize to configure and control the RN-131/RN-171 from an 8-bit Microchip microcontroller.
- You use the pre-compiled sample application **ConfigureInCmdMode.hex** to configure the RN-131/RN-171 module.
- **SampleTelnetApp.hex** accepts specific telnet commands from a remote device and turns on/off the LEDs on either the RN-131/RN-171 module or the PIC18 Explorer Development Board.

The following sections describe these applications in detail.

GETTING STARTED

This section describes how to setup up the PIC18 Explorer Board and to run the sample demo applications.

Hardware Requirements

To run the demo applications, you need the following hardware:

- RN-131-PICTAIL or RN-171-PICTAIL evaluation board
 - PIC18 Explorer Development Board equipped with a PIC18F87J11 PIM (MA180020)
 - Microchip programmer (e.g., ICD3 or Real Ice)
 - 2 jumper wires to configure jumpers J4 and J13 on the PIC18 Explorer Board
- NOTE:** You cannot use the default jumper settings for J4 and J13. Instructions for configuring the jumpers are provided in “Demo Setup” on page 4.
- 802.11b/g-compliant Wi-Fi access point
 - Serial cable or USB-to-serial converter cable

Software Requirements

You need the following software tools/applications to run the demo applications:

- *Terminal emulator application such as TeraTerm or CoolTerm*—You will use the terminal emulator to send configuration commands to the module over a UART interface. The emulator also displays information transmitted from the module.
- *MPLAB X version 1.2 or higher*—You will use this application with the programmer to load the application into the PIC, and to modify and customize the application.
- *XC8 v1.10 compiler*—You will use the compiler to build a customized application.

Sample Application Code

You can obtain the sample application code described in this user manual at www.microchip.com/wireless.

Demo Applications

This section demonstrates how to use the demo applications to accomplish the following tasks:

- Configure the module to join a known, user-specified access point automatically.
- Communicate with the module via a telnet session to turn on/off LEDs using a command set that the application running on the PIC recognizes.

Set Up Hardware

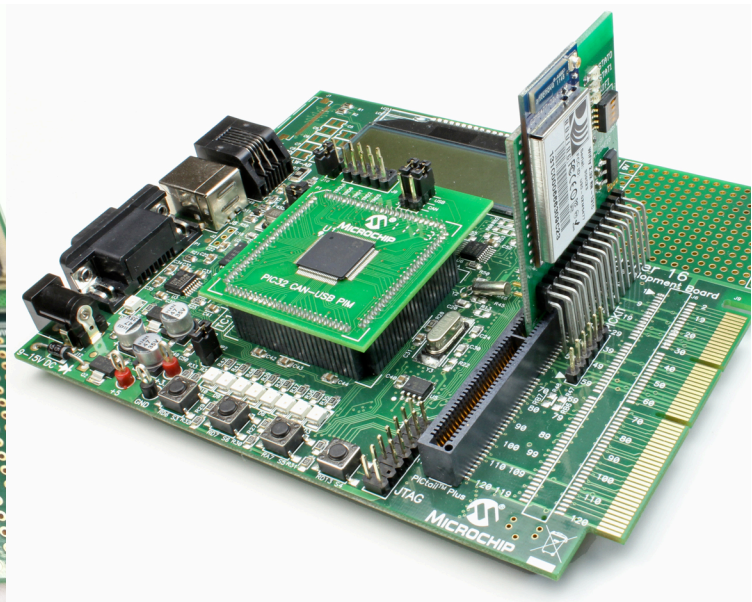
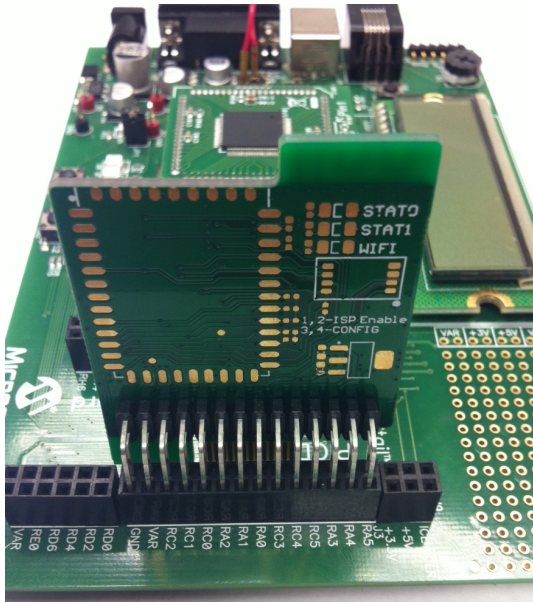
Perform the following steps to set up the hardware and prepare it for configuration:

1. Plug the RN-131-PICTAIL/RN-171-PICTAIL evaluation board into the PIC18 Explorer Board's PICTail connector. See Figure 1.

Figure 1. Modules Connected to PIC18 Explorer Board

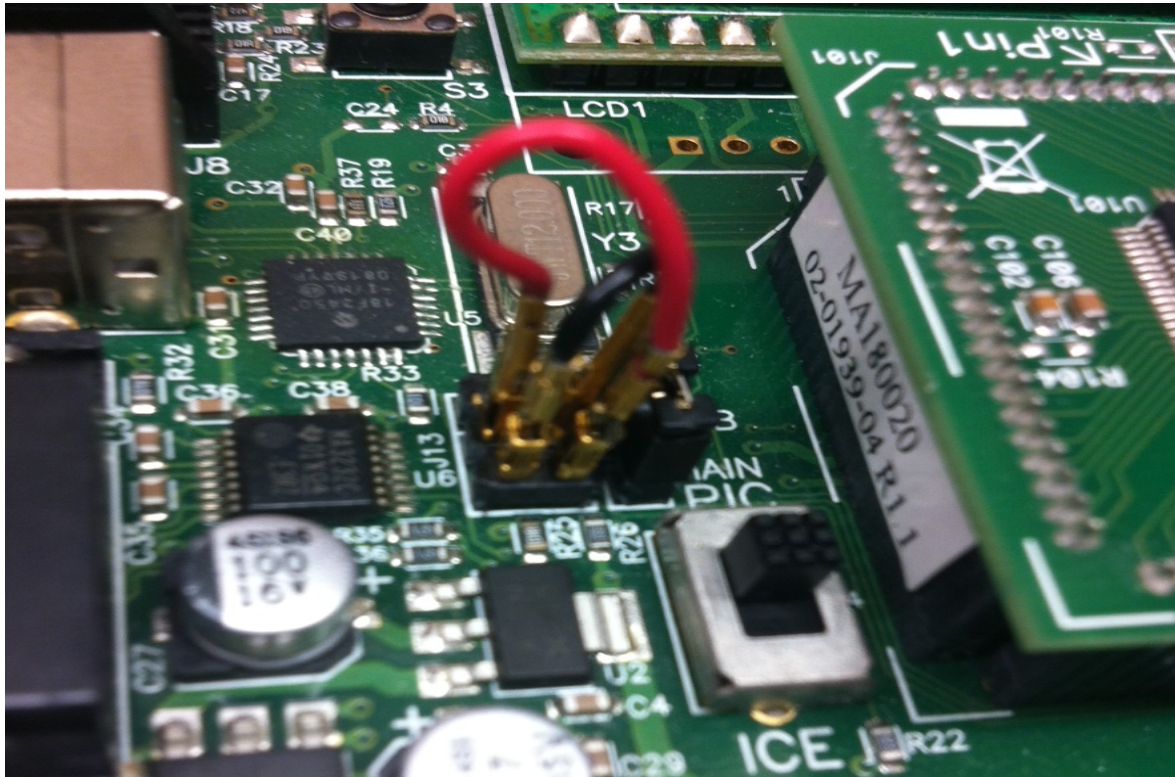
RN-171-PICTAIL Connected to the PIC18 Board

RN-131-PICTAIL Connected to the PIC18 Board



2. Configure the PIC18 Explorer Board's J4 and J13 jumpers as shown in Figure 2. This configuration is for command mode operation.

Figure 2. Jumper Selection for Command Mode Operation



3. Connect a serial cable from the PIC18 Explorer Board's DE9 connector to the PC. If you are using a laptop that does not have a serial connector, use a USB-to-serial converter cable to connect the serial cable to the laptop's USB port.
4. Connect the programmer to the PIC18 Explorer Board and apply power.

Program the PIC18F87J11 & Configure the RN-131/RN-171

In this step you configure the RN-131/RN-171 modules so that they automatically search for, and join, a preferred access point. First, you use the following steps to program the PIC18F87J11 with software that allows the RN-131/RN-171 to be configured for its current operating environment. These steps are specific to the PIC18 Explorer Board because of the way in which the UARTs are connected on the board.

1. Using the MPLAB X IDE and the programmer, load the **ConfigureInCmdMode.hex** file into the PIC18F87J11. This application performs two important tasks:
 - It configures the PIC18F87J11 I/O pins appropriately.
 - It allows the UART signal to traverse from the Explorer 18 board's DE9 connector to the RN-131-PICTAIL/RN-171-PICTAIL, temporarily bypassing the PIC (this step is specific to the PIC18 Explorer Board).
2. Open your terminal emulator to the COM port of the PIC18 Explorer Board/RN-131-PICTAIL/RN-171-PICTAIL. The serial port settings are 9600 baud, 8-bits data, no parity, 1-stop bit, and no flow control.

- Type the following commands in the console of your terminal emulator:

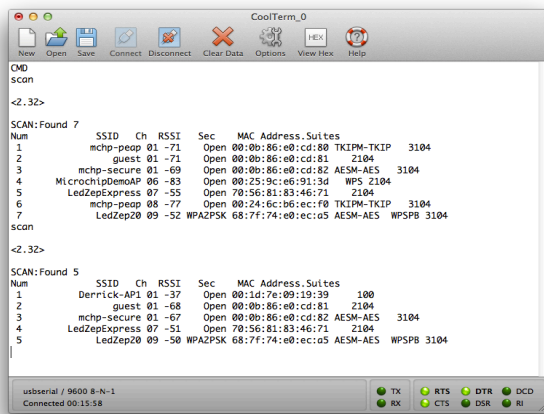
\$\$\$ This command places the RN-131/RN-171 module in command mode.

scan <cr> The device scans for networks and produces a list of available access points (see Figure 3 for an example). The access point to which you wish to connect should be listed. If it is not, repeat the **scan** command.

join # XX <cr> Associate with the access point, where XX is the access point's number interest as shown in the scan report.

leave <cr> This command asks the device to leave the network.

Figure 3. Access Point List



- Store the parameters from step 3 into the RN-131/RN-171 module's non-volatile memory so that they can be used in the next application you run on the PIC18F87J11. Type the following commands in the console of your terminal emulator:

set wlan ssid <string> <cr> Set the network's SSID where <string> is the SSID (e.g., **set wlan ssid RovingNET**).

set wlan pass <string> <cr> Set the passphrase to use when joining where <string> is the passphrase (e.g., **set wlan pass duckmauifries**).

save <cr> Save the settings to persistent storage; reused when joining.

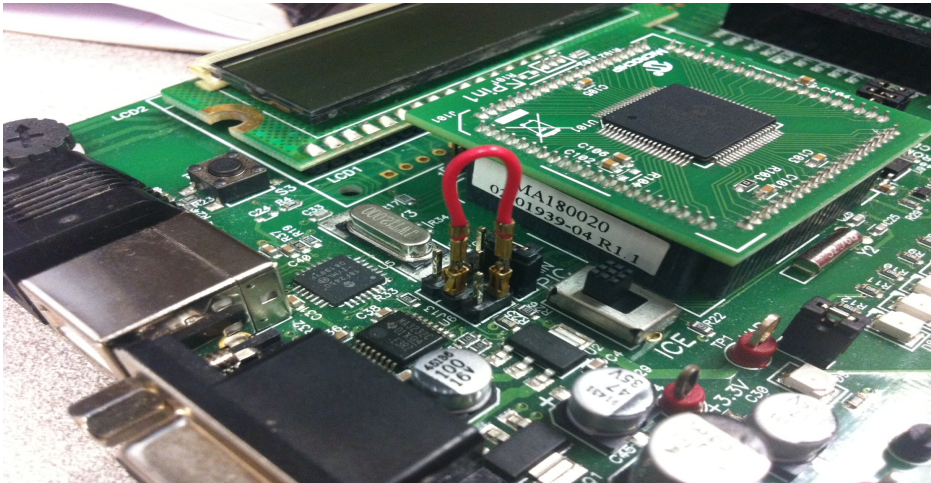
reboot <cr> Reboot the module so that the settings take effect.

Data Mode Operation

In the following steps you load an application into the PIC18F87J11 that allows the RN-131/RN-171 module to operate in data mode and accept a telnet connection request. Additionally, the application lets you control the module's LEDs by sending specific PIC commands.

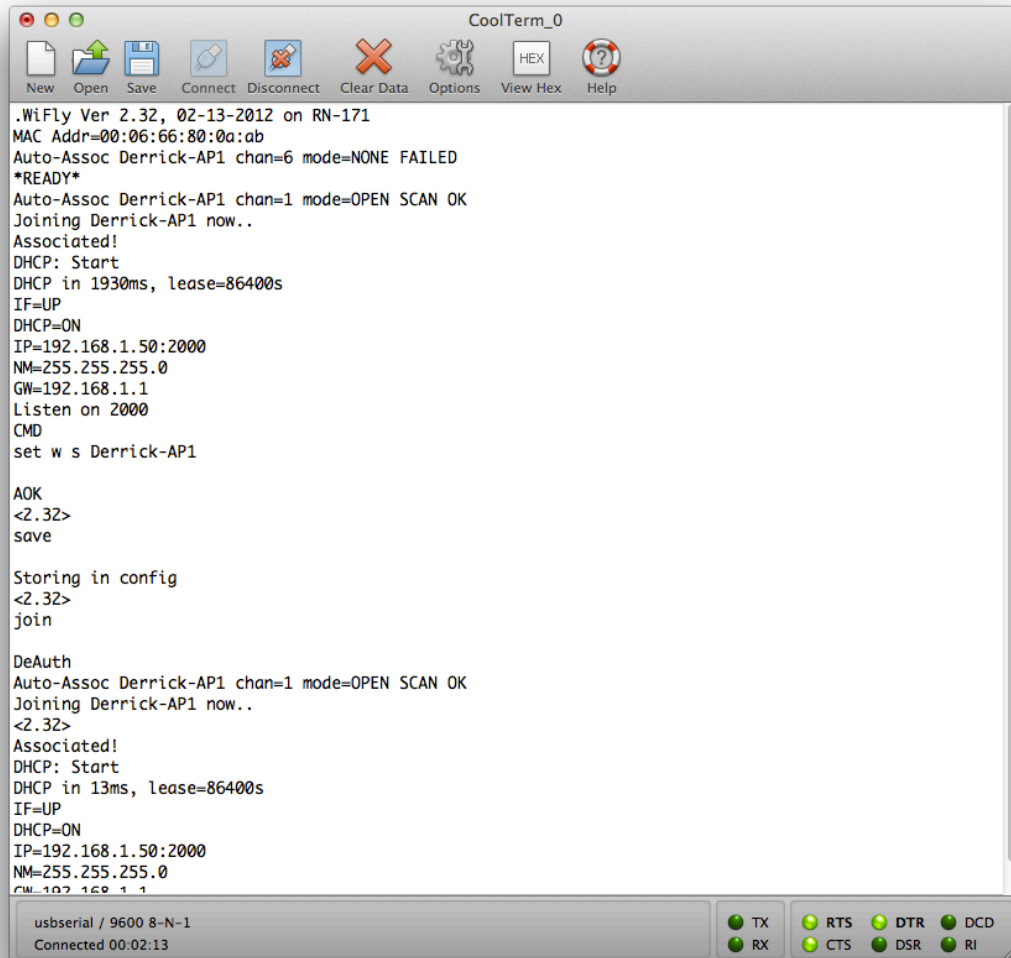
- Configure the jumpers J4 and J13 as shown in Figure 4. This setting allows the PIC's UART to communicate with RN-131/RN-171 module's UART via the PICTail connector, and also allows the module's transmitted data to echo back to the terminal emulator's console. (This setting is specific to the PIC18 Explorer Board.)

Figure 4. Data Mode Operation Jumper Selection



2. Use the MPLAB X IDE and the programmer to load the **SampleTelnetApp.hex** demo application into the PIC18F87J11.
3. Press the MCLR reset button on the Explorer 18 board and observe the output transaction on the terminal emulator console. A normal sequence is similar to that shown in Figure 5, i.e., the device is associated with the designated access point, and the yellow LED on the RN-131-PICTAIL/RN-171-PICTAIL is illuminated.

Figure 5. RN-131/RN-171 Association Start-Up Sequence



4. In the console, note the IP address that the access point assigns to the RN-131/RN-171 module. The IP is unique to each wireless environment's DHCP server. You will use this IP address to open a telnet session with the device.

NOTE: The application source code includes a file called **ConfigApp.h**. Two parameters in this file establish which SSID and passphrase is used if the default access point cannot be found:

```
#define NETWORK_SSID "xxxstringxxx"
#define NETWORK_PASS "yystringyyy"
```

You can change these parameters to match your local network environment. Then, rebuild and re-load the updated application. The application then uses these settings to search for the access point to join, in case the default device parameters stored in the device cannot be found or if found, they do not allow association.

The RN-131/RN-171 module is now associated to an access point and is operating in data mode. Use the following steps to open a telnet session with the device and send it commands:

1. Associate your computer's Wi-Fi connection with the same access point as the RN-131/RN-171 module.
2. On your computer, open a command line session and type the command:

telnet <address> **2000**, where <address> is the IP address of the RN-131/RN-171 module.

For example: **telnet 192.168.1.50 2000**

2000 is the port number on which the RN-131/RN-171 module listens for telnet session commands.

3. The telnet console should display the echoed response ***HELLO***.
4. Anything you type in the telnet console, followed by a <cr>, is echoed on the RN-131/RN-171 module's console. Type some random strings and observe the echoing.
5. The PIC18F87J11 connected to the RN-131/RN-171 module can recognize and act on the following commands:

MOD_TON_GRN	Turn on module's green LED.
MOD_TOFF_GRN	Turn off module's green LED.
EXP_TON_LED1	Turn on Explorer Board's D8 LED.
EXP_TOFF_LED1	Turn off Explorer Board's D8 LED.
EXP_TON_LED2	Turn on Explorer Board's D7 LED.
EXP_TOFF_LED1	Turn off Explorer Board's D7 LED.

Type one of these commands in the telnet console, either by itself or embedded in a longer string, followed by a <cr>. Observe that you can turn on or off the appropriate LEDs on either the evaluation board or the PIC18 Explorer Board.

You have completed the application demonstration. The next section describes how the software is designed for PIC18 microcontroller.

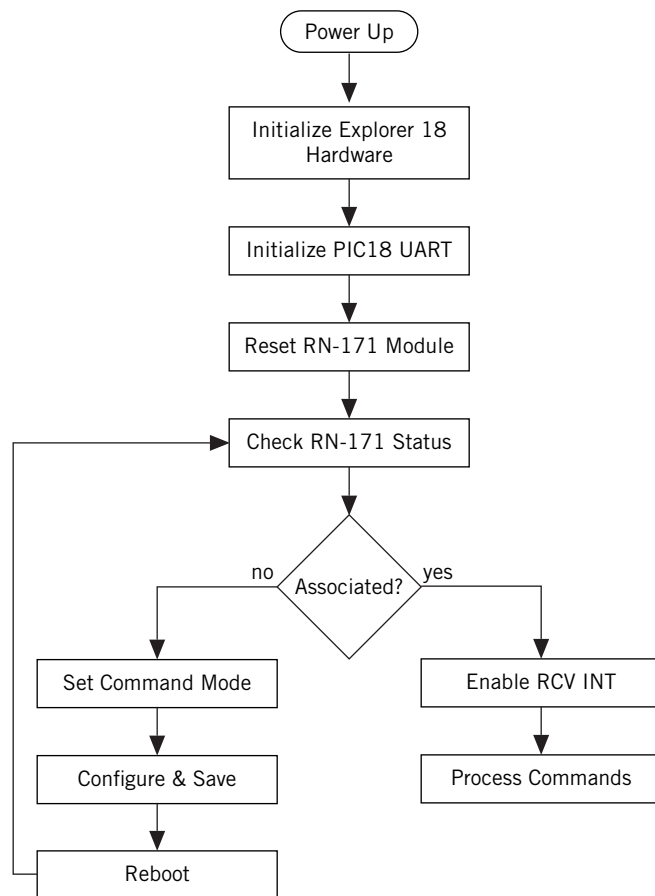
SAMPLE APPLICATION DESIGN

This chapter provides a high-level overview of how the sample demo application was designed, and shows some of the APIs you can use to communicate with RN-171 module.

Demo Application Flow Diagram

Figure 6 shows the demo application flow diagram.

Figure 6. Flow Diagram



Initialize the PIC18 Explorer Board Hardware

During this processing stage, the **void BoardInit(void)** API function performs the following tasks:

- Chooses the external 10-MHz crystal as the clock source.
- Configures the ports/pins as digital and selects their direction to match the functional requirements of the PICTail connector.
- Turns off the LEDs.
- Clears the interrupt flags.
- Disables the peripheral interrupts.

Initialize the PIC's UART

The **void ConsoleInit(void)** API function initializes the PIC's UART1, which communicates with the RN-131/RN-171 module. It sets the UART's baud rate to 9600 baud, 8 bits data, 1 stop bit, no parity, and no flow control. These settings are the default for the RN-131/RN-171 module. To change the baud rate, modify the **#define BAUD_RATE 9600** macro in the **console.h** header file, and re-initialize the UART.

The **ConsoleInit()** function enables both transmit and receive UART capabilities, but their respective interrupts are disabled until the application is prepared to receive them. Later, receive interrupts are enabled. Transmit interrupts remain disabled.

Reset the RN-171 Module

The PICTail connector has an active-low reset pin (RB1) that is used to reset the RN-131/RN-171 module after the PIC18 Explorer Board and UART are properly initialized. The **void rn_reset_wifi(void)** API function is used for this reset operation. It transitions the reset pin from high to low, holds the pin low for 10 ms, and then returns it high. This reset action triggers the module to begin its joining procedure.

Check Module's Status

Immediately after resetting the module, the PIC's UART receiver interrupt is enabled. This process prepares the PIC to receive the information that is transmitted from the RN-171's UART as it goes through its start up procedure.

The PIC18 parses the RN-171 output and searches for the module's IP address if it was successful in associating with an access point. If a valid IP address is received, the assumption is made that the device is safely on a network and is in data mode. In this case, the PIC18 waits to receive commands from the RN-131/RN-171 module in an endless loop. The PIC18 acts on each valid command as it receives it.

Command Mode

If the PIC18 application detects that the RN-171 module was not successful in its start-up attempt to associate with an access point, the PIC's application attempts to instruct the module to associate with a specific access point. The PIC18 application sends the following commands to the module in sequence:

\$\$\$	Put the module into command mode.
set wlan ssid <string>	Set the network's SSID where <string> is the SSID.
set wlan pass <string>	Set the passphrase to use when joining where <string> is the passphrase.
save	Save the settings to persistent storage; reused when joining.
reboot	Reboot the module so that the settings take effect.

When successful, the module reboots and associates with the specified access point. The following section describes the APIs used to carry out these operations.

Primary Module Communications API Functions

The API function that tells the device to go into command mode is:

```
err_t module_send_cmd(const char *response, const char *fmt, ...)
```

This function accepts 2 or more parameters (i.e., a variable length parameter list). The first parameter, **response**, is the reply expected from the RN-171 module after it has processed the command. The command is carried in the **fmt** parameter. For example, if the function is called as follows:

```
retValue = err_t module_send_cmd("CMD", "$$$");
```

The command is **\$\$\$** and the expected return is **CMD**. For a more complex instantiation such as:

```
retValue = module_send_cmd(CMD_AOK, "set wlan ssid %s", NETWORK_SSID)
```

The variable parameter list is essential to properly resolve the **%s** into the value of the **NETWORK_SSID** macro.

The next API demonstrates how you can combine individual commands inside a single function to carry out a more complex operation, such as joining a network. The following API function forces the device into command mode, saves the SSID and passphrase, and reboots the module.

```
err_t module_network_connect(void);
```

Internally, it sends a total of 5 commands.:

```
module_send_cmd("CMD", "$$$")  
module_send_cmd(CMD_AOK, "set w s %s", NETWORK_SSID)  
module_send_cmd(CMD_AOK, "set w p %s", NETWORK_PASS)  
module_send_cmd(NULL, "save")  
module_send_cmd(NULL, "join")
```

This function represents a typical API that sends a sequence of commands to the module to carry out a complex task. Most applications follow a similar pattern.

RESOURCES & RELATED DOCUMENTS

For more information, refer to the following sources, which are available on the Support page on the Roving Networks website at <http://www.rovingnetworks.com/support.php>:

- *RN-171 Data Sheet*
- *RN-131 Data Sheet*
- *Advanced User Manual*
- *WiFly Training Presentation*
- Drivers, tools, and utilities

NOTES

Roving Networks, Inc.
102 Cooper Court
Los Gatos, CA 95032
+1 (408) 395-5300
www.rovingnetworks.com

Copyright © 2012 Roving Networks. All rights reserved. Roving Networks is a registered trademark of Roving Networks. Apple Inc., iPhone, iPad, iTunes, Made for iPhone are registered trademarks of Apple Computer.

Roving Networks reserves the right to make corrections, modifications, and other changes to its products, documentation and services at any time. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Roving Networks assumes no liability for applications assistance or customer's product design. Customers are responsible for their products and applications which use Roving Networks components. To minimize customer product risks, customers should provide adequate design and operating safeguards.

Roving Networks products are not authorized for use in safety-critical applications (such as life support) where a failure of the Roving Networks product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use.

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- Конкурентоспособные цены и скидки постоянным клиентам.
- Специальные условия для постоянных клиентов.
- Подбор аналогов.
- Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.
- Приемлемые сроки поставки, возможна ускоренная поставка.
- Доставку товара в любую точку России и стран СНГ.
- Комплексную поставку.
- Работу по проектам и поставку образцов.
- Формирование склада под заказчика.
- Сертификаты соответствия на поставляемую продукцию (по желанию клиента).
- Тестирование поставляемой продукции.
- Поставку компонентов, требующих военную и космическую приемку.
- Входной контроль качества.
- Наличие сертификата ISO.

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- Регистрацию проекта у производителя компонентов.
- Техническую поддержку проекта.
- Защиту от снятия компонента с производства.
- Оценку стоимости проекта по компонентам.
- Изготовление тестовой платы монтаж и пусконаладочные работы.



Тел: +7 (812) 336 43 04 (многоканальный)

Email: org@lifeelectronics.ru

www.lifeelectronics.ru