

This Application Note describes how to create a wireless UART link using two sensor boards from a Jennic evaluation kit based on the JN5148, JN5139 or JN5121 wireless microcontroller. The software required to implement the system was developed using the 802.15.4 Stack API and Integrated Peripherals API, both provided free-of-charge by Jennic.

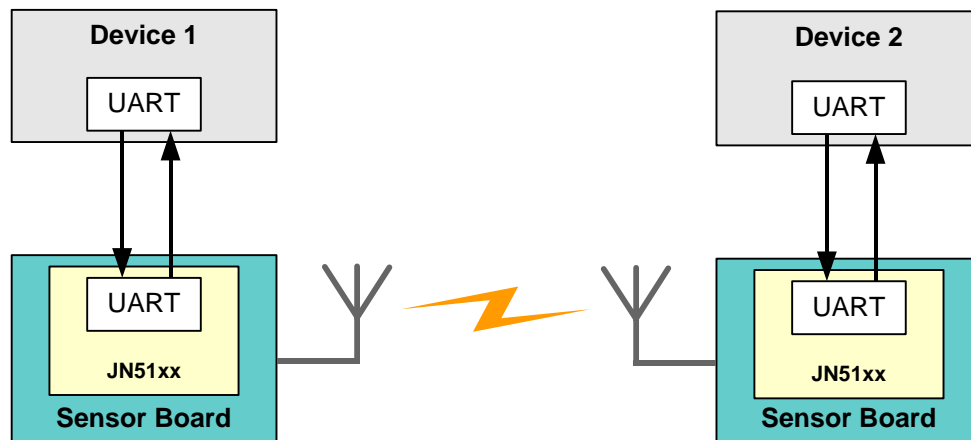
The benefits of using this solution include:

- Replacement of serial cables with wireless connectivity
- Quick addition of low-cost wireless connectivity to products using a UART link for communications

## Application Overview

This application allows two devices (such as PCs) containing UARTs to communicate via a wireless (radio) link. Each device connects to a sensor board from the Jennic evaluation kit – the device's UART connects to the on-chip UART on the sensor board. The two sensor boards then provide the radio communication link.

This is illustrated in the figure below.



**Figure 1: Wireless UART Link**

This configuration has been tested by connecting each of the sensor boards to a PC using the programming cable supplied with the evaluation kit.

**Note:** This example application implements basic communication, which may be lossy. For a lossless version, refer to the Application Note *802.15.4 Wireless UART with Flow Control* (JN-AN-1069).

## System Architecture

When the JN51xx device is used in an application such as this, it is assumed that a permanent power source will be available at both ends of the wireless link. This means the on-chip radio can always be active, eliminating the need to synchronise the transmission/reception of data. The link is designed to operate at up to 19200 baud.

One of the sensor boards must act as a PAN Co-ordinator. The PAN Co-ordinator is responsible for starting the network and allocating an address to the other sensor board, which acts as an End Device.

## Software Design

Apart from the network management performed by the PAN Co-ordinator, the software contained within each sensor board is identical. Data received via the radio is output to the connected device using the on-chip UART, and data received by the on-chip UART from the device is transmitted over the radio. This process is repeated every 10 ms.

The board LEDs are used as follows:

- LED2 on each board indicates the network status and is illuminated once the network has been successfully created, i.e. once the End Device has associated with the PAN Co-ordinator.
- LED1 flashes approximately once per second to indicate that the software is running.

The source files required to build the application are described below:

### **wuart\_c.c**

This is an application for the PAN Co-ordinator. It performs the following tasks:

- Selects the quietest channel
- Starts the network
- Allows an End Device to associate with the Co-ordinator
- Transfers data from the radio to the on-chip UART and vice-versa:
  - Data arriving from the connected device via the on-chip UART is buffered and transmitted via the radio every 10 ms
  - Data received via the radio is buffered and transferred to the connected device via the on-chip UART as soon as it is available

### **wuart\_e.c**

This is an application for the End Device. It performs the following tasks:

- Scans the radio channels, looking for the PAN Co-ordinator
- Once it has found the Co-ordinator, it associates with it

Data transfer between radio and the on-chip UART is identical to that described above.

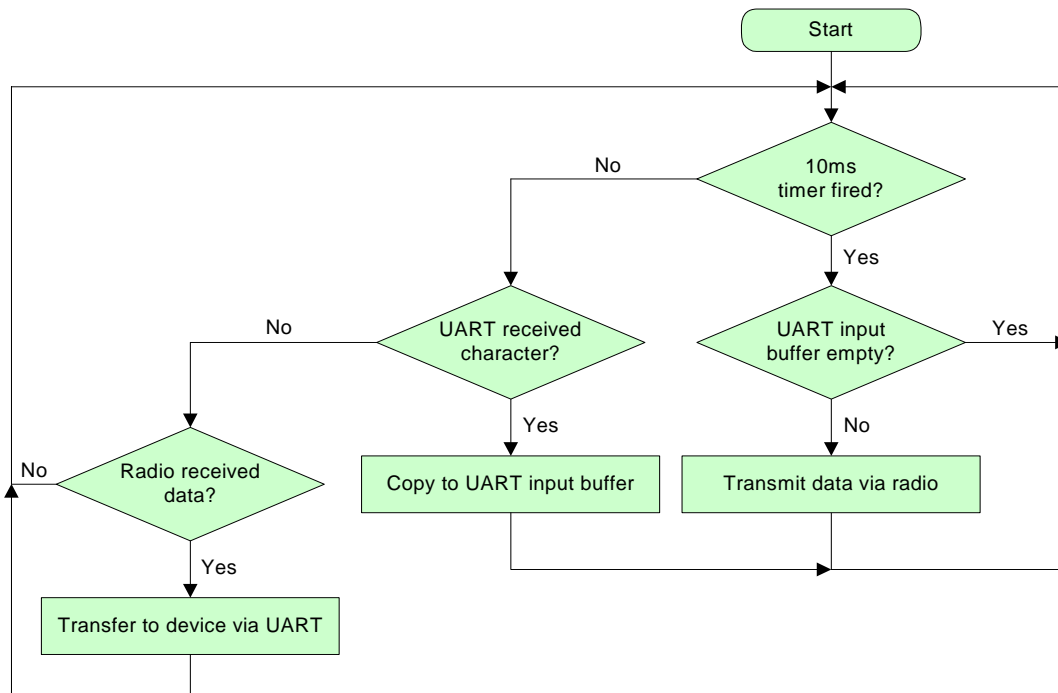
**uart.c, serialq.c, serial.c**

These files are common to both applications, and are used to support serial communications using the on-chip UART.

**config.h**

This file contains configuration information that is common to both applications, e.g. baud rate, PAN ID etc.

The following flowchart illustrates the transmit/receive processes for both the Co-ordinator and End Device. Note that the flowchart refers only to the on-chip UART – the UART input buffer accepts data coming from the connected device.



**Figure 2: Software Flowchart**

## UART Selection

The JN51xx device contains two UARTs (UART0 and UART1). This Application Note uses UART0 by default, but UART1 can be used simply by making the following change in the file **config.h**:

The line

```
#define UART                                E_AHI_UART_0
```

should be changed to

```
#define UART                                E_AHI_UART_1
```

## Compatibility

The software provided with this Application Note has been tested with the following Jennic kits and SDK versions:

Product Type	Part Number	Version	Supported Chips	Supported Protocols
Evaluation Kit	JN5121-EK000	-	JN5121	802.15.4
	JN5121-EK010	-	JN5121	802.15.4, ZigBee
	JN5139-EK000	-	JN5139	802.15.4, JenNet
	JN5139-EK010	-	JN5139	802.15.4, JenNet, ZigBee
	JN5148-EK010	-	JN5148	802.15.4, ZigBee PRO
SDK Libraries	JN-SW-4030	v1.5	JN5139, JN5121	802.15.4, JenNet, ZigBee
	JN-SW-4040	v1.1	JN5148	802.15.4, ZigBee PRO
SDK Toolchain	JN-SW-4031	v1.1	JN5139, JN5121	-
	JN-SW-4041	v1.0	JN5148	-

## Building and Loading the Application

The software provided with this Application Note can be built for any Jennic wireless controller: JN5148, JN5139 or JN5121. However, the available build methods differ between these chip types:

- JN5148 applications can be built using the Eclipse IDE or makefiles
- JN5139/JN5121 applications can be built using the Code::Blocks IDE or makefiles

Note that different makefiles are provided for JN5148 and JN5139/JN5121.

In order to build the supplied software, the application's folder must be placed in the **Application** folder of the Jennic SDK installation:

- For JN5148: **<JENNIC\_SDK\_ROOT>\Application**
- For JN5139/JN5121: **<JENNIC\_SDK\_ROOT>\cygwin\jennic\SDK\Application**

where **<JENNIC\_SDK\_ROOT>** is the path into which the Jennic SDK was installed (by default, this is **C:\Jennic**). The **Application** directory is automatically created when you install the Jennic SDK.

Build the application as described in the appropriate section below, depending on whether you intend to use Eclipse, Code::Blocks or makefiles.


### Using Eclipse (JN5148 Only)

To build the application and load it into JN5148 boards, follow the instructions below:

1. Ensure that the project directory is located in

**<JENNIC\_SDK\_ROOT>\Application**

where **<JENNIC\_SDK\_ROOT>** is the path into which the SDK was installed.

2. Start the Eclipse platform and import the relevant project files (**.project** and **.cproject**) as follows:
  - a) In Eclipse, follow the menu path **File>Import** to display the **Import** dialogue box.
  - b) In the dialogue box, expand **General**, select **Existing Projects into Workspace** and click **Next**.
  - c) Enable **Select root directory**, browse to the Jennic **Application** directory and click **OK**.
  - d) In the **Projects** box, select the project to be imported and click **Finish**.
3. Build an application. To do this, ensure that the project is highlighted in the left panel of Eclipse and use the drop-down list associated with the hammer icon  in the Eclipse toolbar to select the relevant build configuration – once selected, the application will automatically build. Repeat this to build the other application.  
 The binary files will be created in the relevant **Build** directories, the resulting filenames indicating both the chip type (**JN5148**) and networking stack (**154** for 802.15.4) for which they were built.
4. Load the resulting binary files (Release or Software Debug) into the boards. You can do this using the Jennic JN51xx Flash Programmer, which can be launched from within Eclipse or used directly (and is described in the *JN51xx Flash Programmer User Guide (JN-UG-3007)*).

## Using Code::Blocks (JN5139/JN5121 Only)

To build each application and load it into a JN5139/JN5121 board, follow the instructions below:

1. Ensure that the project directory is located in  
**<JENNIC\_SDK\_ROOT>\cygwin\jennic\SDK\Application**  
 where **<JENNIC\_SDK\_ROOT>** is the path into which the Jennic SDK was installed.
2. Open the appropriate Code::Blocks project file (**.cbp** file in the **CodeBlocksProject** directory) and build.  
 The project files are named according to both the chip type (**JN5139** or **JN5121**) and networking stack (**154** for 802.15.4) for which the binaries are to be built.  
 The binary file will be created in the **5139\_Build** or **5121\_Build** directory, the resulting filename matching that of the project file used to create it.
3. Load the resulting binary file into the board. You can do this using the Jennic JN51xx Flash Programmer, which can be launched from within Code::Blocks or used directly (and is described in the *JN51xx Flash Programmer User Guide (JN-UG-3007)*).



**Caution:** If problems occur when using your current Code::Blocks version, download the latest version along with the latest version of this Application Note from the Support area of the Jennic web site ([www.jennic.com/support](http://www.jennic.com/support)).

## Using Makefiles

The application for each node type (Co-ordinator, End Device) has its own **Build** directory, which contains the makefiles for the application.

Different makefiles are provided for JN5148 and JN5139/JN5121 – a JN5148 makefile is simply called **Makefile** while a JN5139/JN5121 makefile is called **Makefile\_JN5139.mk**.

To build each application and load it into a JN51xx board, follow the instructions below:

1. Ensure that the project directory is located in

**<JENNIC\_SDK\_ROOT>\Application** for JN5148

**<JENNIC\_SDK\_ROOT>\cygwin\jennic\SDK\Application** for JN5139/JN5121

where **<JENNIC\_SDK\_ROOT>** is the path into which the Jennic SDK was installed.

2. Navigate to the **Build** directory for the application to be built and follow the instructions below for your chip type:

**For JN5148:**

At the command prompt, enter:

```
make clean all
```

Note that for the JN5148, you can alternatively enter the above command from the top level of the project directory, which will build the binaries for both applications.

**For JN5139:**

At the command prompt, enter:

```
make -f Makefile_JN5139.mk clean all
```

**For JN5121:**

At the command prompt, enter:

```
make -f Makefile_JN5139.mk JENNIC_CHIP=JN5121 clean all
```

In all the above cases, the binary file will be created in the relevant **Build** directory, the resulting filename indicating both the chip type (**JN5148**, **JN5139** or **JN5121**) and networking stack (**154** for 802.15.4) for which the application was built.

3. Load the resulting binary file into the board. To do this, use the Jennic JN51xx Flash Programmer, described in the *JN51xx Flash Programmer User Guide (JN-UG-3007)*.

## Demonstration

The functionality of this application software can be demonstrated by forming a wireless connection between the serial ports of two PCs. This is illustrated in the figure below.

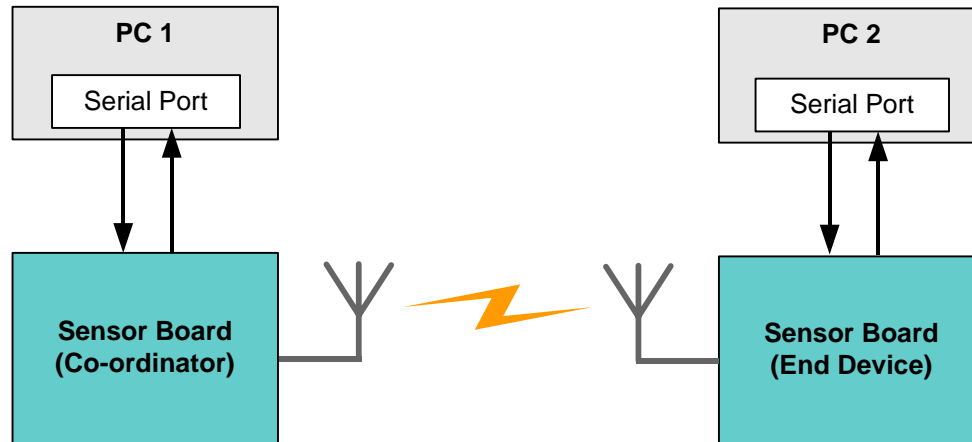


Figure 3: Wireless Serial Link

To perform this demonstration, follow the steps below:

1. Connect each board to the serial port of a PC, using the serial cable and programming adaptor provided with the evaluation kit.
2. Power-on both of the boards
3. Test the wireless serial link by transferring data using a terminal emulator such as HyperTerminal (configure the serial connection as follows: 19200 bps, 8 data bits, no parity, 1 stop bit, no handshaking).



**Caution:** Some terminal emulation programs, including HyperTerminal, hold RTS asserted when handshaking is set to 'none'. This forces the Jennic module into 'programming mode' in which it checks the PGM pin at power-up and finds it held LOW. If LED1 does not start flashing at power-up, you should disconnect the serial cable and power cycle the board. Alternatively, ensure that HyperTerminal is started **after** powering up the evaluation kit boards.

## Revision History

Version	Notes
0.1	Initial version.
1.0	Released to website.
1.1	Updated document style.
1.2	Updated to include ROM and library based source code.
1.3	Updated document template and corrected minor bugs in ROM MAC version of source code.
1.4	Added paragraph on Hyperterminal usage
1.5	Title of Application Note changed (from Wireless UART). Co-ordinator now selects best channel (based on energy scan). Improvements made to UART configuration, now easy to change baud rate and UART (0 or 1). Note : Above changes only available in ROM MAC version of source code.
1.6	Improvements made to UART selection code, and added text to Application Note regarding this feature. Note: Change only available in ROM MAC version of source code.
1.7	Corrected rounding error in baud rate calculation. Removed v2 version of source code.
1.8	Section added on building and downloading application
1.9	Updated to support JN5121 and JN513x devices
2.0	Updated for R1 of JN5139 device
2.1	Updated for JN5139 and new SDK
3.0	Support for JN5148 added



## Important Notice

Jennic reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and services at any time, and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders, and should verify that such information is current and complete. All products are sold subject to Jennic's terms and conditions of sale, supplied at the time of order acknowledgment. Information relating to device applications, and the like, is intended as suggestion only and may be superseded by updates. It is the customer's responsibility to ensure that their application meets their own specifications. Jennic makes no representation and gives no warranty relating to advice, support or customer product design.

Jennic assumes no responsibility or liability for the use of any of its products, conveys no license or title under any patent, copyright or mask work rights to these products, and makes no representations or warranties that these products are free from patent, copyright or mask work infringement, unless otherwise specified.

Jennic products are not intended for use in life support systems/appliances or any systems where product malfunction can reasonably be expected to result in personal injury, death, severe property damage or environmental damage. Jennic customers using or selling Jennic products for use in such applications do so at their own risk and agree to fully indemnify Jennic for any damages resulting from such use.

All trademarks are the property of their respective owners.

**Jennic Ltd**  
Furnival Street  
Sheffield  
S1 4QT  
United Kingdom

Tel: +44 (0)114 281 2655  
Fax: +44 (0)114 281 2951  
E-mail: [info@jennic.com](mailto:info@jennic.com)

For the contact details of your local Jennic office or distributor, refer to the Jennic web site:

**www.Jennic.com**  
TECHNOLOGY FOR A CHANGING WORLD