

Application Note: JN-AN-1132

JN51xx Customer Module Evaluation Tool

This Application Note describes how to use the JN51xx Customer Module Evaluation Tool to place the NXP JN5148, JN5142 or JN5139 microcontroller into a number of test modes that enable measurement of module performance.

Application Overview

The software supplied with this Application Note allows the performance of customer modules based on the NXP JN5148, JN5142 or JN5139 device to be evaluated in various ways including received signal strength, packet error rate (PER), crystal oscillator frequency and current consumption, among others. Some tests require the use of two boards (carrying JN51xx-based modules).

The tool is provided in two parts:

- **Customer Module Evaluation Tool:** This is the main part of the tool which provides all functionality except the PER testing (see below)
- **Module PER Test Tool:** This part performs PER testing on a JN51xx-based module.

Only one of the above tools can be used at a time. The required application must be loaded into the Flash memory of the module under test.

Compatibility

The software provided with this Application Note has been tested with the following kits and SDK (Software Developer's Kit) versions:

Product Type	Part Number	Version	Supported Chips
Evaluation Kit	JN5139-EK000	-	JN5139
	JN5139-EK010	-	JN5139
	JN5142-EK020	-	JN5142
	JN5148-EK010	-	JN5148
SDK Libraries	JN-SW-4030	v1.5	JN5139, JN5121
	JN-SW-4040	v1.5	JN5148, JN5142
	JN-SW-4054	v1.0	JN5148-J01
	JN-SW-4055	v1.0	JN5142-J01
SDK Toolchain	JN-SW-4031	v1.1	JN5139, JN5121
	JN-SW-4041	v1.1	JN5148, JN5142

Building and Downloading the Application

The JN51xx Customer Module Evaluation Tool is provided in this Application Note as two applications: **AN1132_CustomerModuleEvalTool** and **AN1132_ModulePerTest**.

These applications can be built for the JN5148, JN5142 or JN5139 microcontroller. However, the available build methods differ between these chip types:

- JN5148 and JN5142 applications can be built using the Eclipse IDE or makefiles
- JN5139 applications can be built using the Code::Blocks IDE or makefiles

Note that different makefiles are provided for JN514x and JN5139 devices.

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

- For JN5148 and JN5142: **<JN51xx_SDK_ROOT>\Application**
- For JN5139: **<JN51xx_SDK_ROOT>\cygwin\jennic\SDK\Application**

where **<JN51xx_SDK_ROOT>** is the path into which the SDK was installed (by default, this is **C:\Jennic**). The **Application** directory is automatically created when you install the SDK.

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



Note: To build for JN5148-001 device, install version (1v40) of libJPT_JN5148.a and JPT.h included in this application note

Copy libJPT_JN5148.a to C:\Jennic\Chip\JN5148\Library

Copy JPT.h to C:\Jennic\Components\Utilities\Include

Using Eclipse (JN5148 and JN5142)

To build one of the applications and load it into a JN514x-based module, follow the instructions below (only one application can be present in Flash memory at a time):

1. Ensure that the project directory is located in

<JN514x_SDK_ROOT>\Application

where **<JN514x_SDK_ROOT>** is the path into which the JN514x 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 **Application** directory and click **OK**.
- d) In the **Projects** box, select the project to be imported and click **Finish**.



Note: If building for the JN5142 device, follow the Eclipse menu path **Project>Properties>C/C++ Build** and modify the build command to `make JENNIC_CHIP=JN5142 clean all`. Alternatively, edit the JN514x makefile to set the chip type as `JENNIC_CHIP=JN5142`.




Note: For **JN5142-J01** device, specify `JENNIC_CHIP=JN5142J01`



Note: For **JN5148-J01** device, specify `JENNIC_CHIP=JN5148J01`



Note: For **JN5148-Z01** device, specify `JENNIC_CHIP=JN5148Z01`

3. Build the 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.

The binary file will be created in the **Build** directory, the resulting filename indicating the chip type (**JN5148** or **JN5142**) for which the application was built.

4. Load the resulting binary file into the module. You can do this using the 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 Only)

To build one of the applications and load it into a JN5139-based module, follow the instructions below (only one application can be present in Flash memory at a time):

1. Ensure that the project directory is located in

`<JN5139_SDK_ROOT>\cygwin\jennic\SDK\Application`

where `<JN5139_SDK_ROOT>` is the path into which the JN5139 SDK was installed.

2. Open the Code::Blocks project file (**.cbp** files in the **CodeBlocksProject** directory) and build.

The project filename includes the chip type (**JN5139**) for which the binary is to be built.

The binary file will be created in the **5139_Build** directory, the resulting filename matching that of the project file used to create it.

3. Load the resulting binary file into the module. You can do this using the 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)*).

Using Makefiles

The **Build** directory contains the makefiles for the applications. A makefile covers both applications.

Different makefiles are provided for JN514x and JN5139 – the JN514x makefile is simply called **Makefile** while the JN5139 makefile is called **Makefile_JN5139.mk**.

To build the applications and load one of them into a JN51xx-based module, follow the instructions below (only one application can be present in Flash memory at a time):

1. Ensure that the project directory is located in

<JN51xx_SDK_ROOT>\Application for JN5148 and JN5142

<JN51xx_SDK_ROOT>\cygwin\jennic\SDK\Application for JN5139

where **<JN51xx_SDK_ROOT>** is the path into which the SDK was installed.

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

For JN5148 and JN5142:

The instruction below assumes that the type of JN514x chip has been specified in the makefile using either `JENNIC_CHIP=JN5148` or `JENNIC_CHIP=JN5142`, as appropriate.

At the command prompt, enter:

```
make clean all
```

Alternatively, the JN514x chip type can be specified in the make command as:

```
make JENNIC_CHIP=JN5148 clean all
```

or

```
make JENNIC_CHIP=JN5142 clean all
```

For JN5139:

At the command prompt, enter:

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

In all the above cases, the binary files will be created in the **Build** directory, the resulting filenames indicating the chip type (**JN5148**, **JN5142** or **JN5139**) for which the applications were built.

3. Load one of the resulting binary files into the module. To do this, use the JN51xx Flash Programmer, described in the *JN51xx Flash Programmer User Guide (JN-UG-3007)*.

Setting Up and Running the Tool

You will access the tool running on the JN51xx-based module from a PC. Set up a connection between a PC and the board (carrying the JN51xx-based module) as follows:

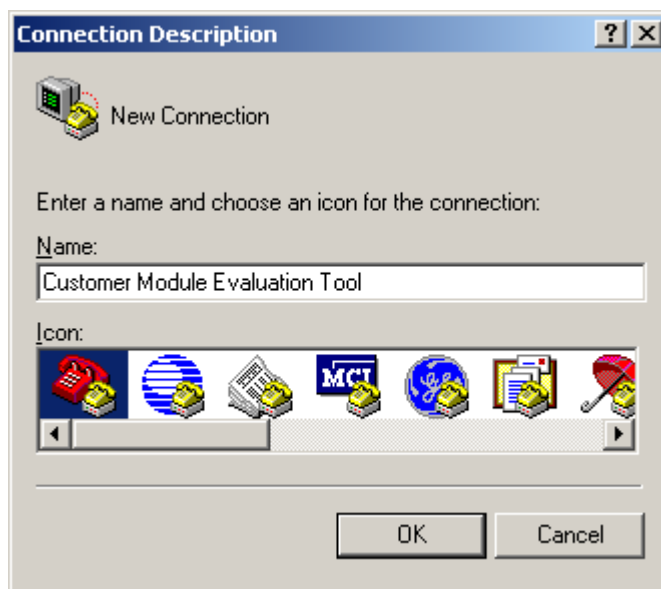
1. If not already done, load the desired application into a JN51xx-based module using the JN51xx Flash Programmer, described in the *JN51xx Flash Programmer User Guide (JN-UG-3007)*. The possible applications are:
 - **AN1132_CustomerModuleEvalTool_JN51xx.bin**
 - **AN1132_ModulePerTest_JN51xx.bin**

This step involves connecting a PC to the carrier board using a USB-to-serial cable. Once application loading has completed, leave this cable connected.

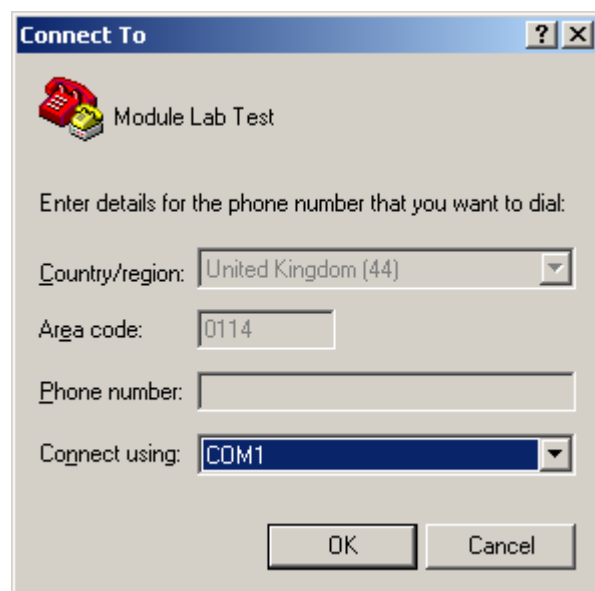
2. Close the JN51xx Flash Programmer.
3. Reset the target board.
4. Open a connection to the board using a terminal emulator at 38400 baud, no parity, 8 data bits, 1 stop bit and no handshaking.

Operational instructions are provided below for those using Microsoft HyperTerminal to interface with the tool.

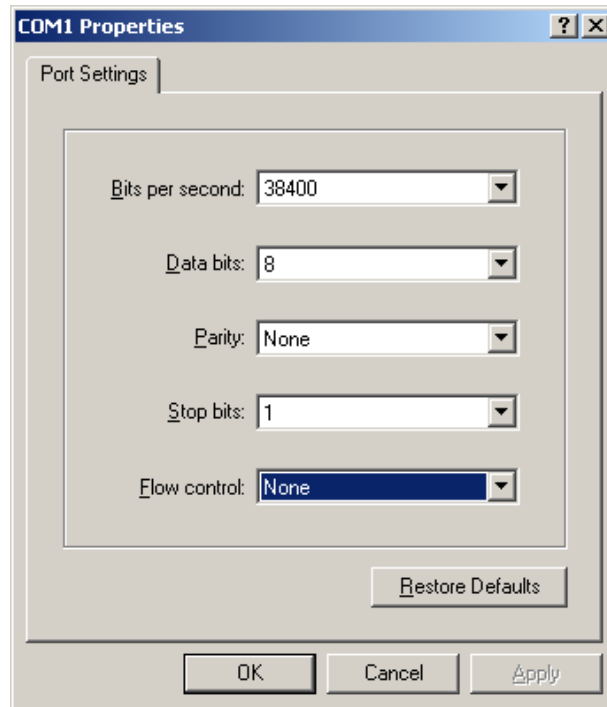
1. Open HyperTerminal: **Accessories > Communications > HyperTerminal**.
2. Access the **New Connection** dialogue box: **File > New Connection**.
Type a name for the connection, then click **OK**.



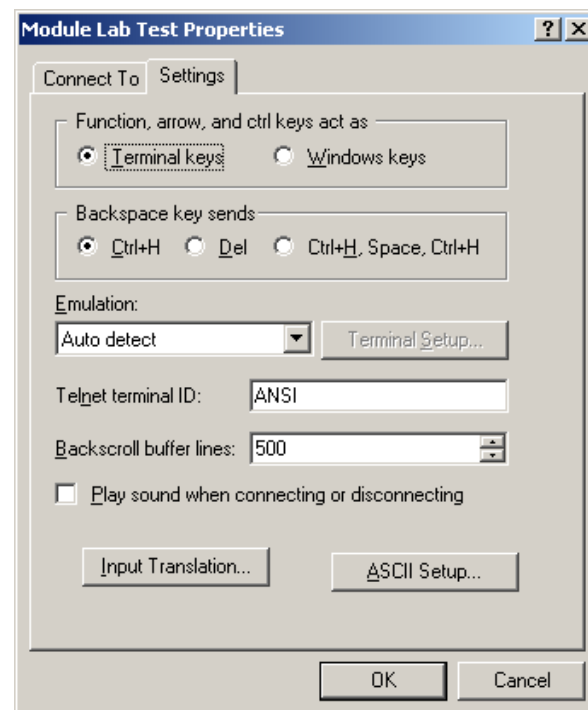
3. Choose the serial communications port that the board is connected to and then click **OK**



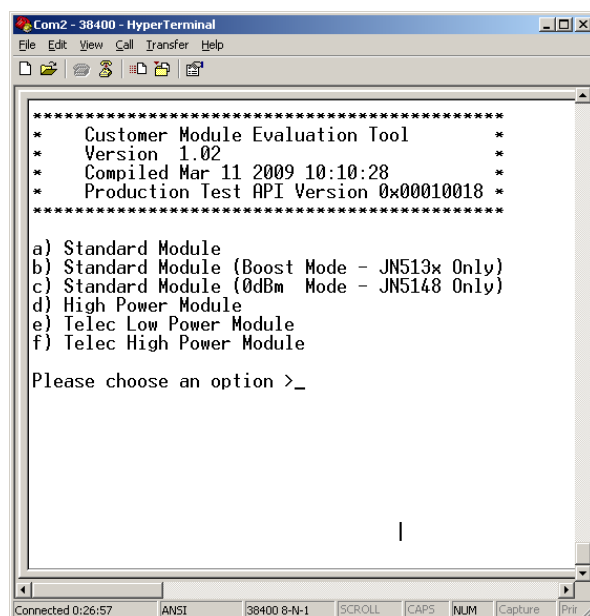
- Set the port settings to 38400 bits per second, 8 data bits, no parity, 1 stop bit and no flow control, then click **OK**.



- Access the **Properties** dialogue box: **File > Properties**. Click on the **Settings** tab (shown below), ensure the box labelled **Emulation** is set to "Auto detect" and then click **OK**.



6. Press any key, and the screen shown below will be generated by the board.



```
*****
* Customer Module Evaluation Tool *
* Version 1.02 *
* Compiled Mar 11 2009 10:10:28 *
* Production Test API Version 0x00010018 *
*****
a) Standard Module
b) Standard Module (Boost Mode - JN513x Only)
c) Standard Module (0dBm Mode - JN5148 Only)
d) High Power Module
e) Telec Low Power Module
f) Telec High Power Module

Please choose an option >_
```

7. In the above screen:
- Select the appropriate module type from those listed.
 - You will now be prompted to specify the CPU clock frequency – select 16 MHz or 32 MHz, as appropriate.
 - If you are using the main part of the tool, you will now be prompted to specify one of the following test modes:
 - TX Power Test (CW)
 - TX Power Test (Modulated)
 - Receive Test
 - Oscillator Frequency Test
 - Current Measurement Test
 - RF Power Measurement
 - Trigger Packet Test
 - Receive Packets Test
 - Transmit Packets Test
 - Connectionless Packet Error Rate Test
 - CCA Test

If you are using the PER tester, only one test mode is available.

The above test modes are described below.

Transmit Power Test (CW Mode)

The CW mode Transmit Power Test puts the device into a transmit state, generating a continuous wave signal. The duty cycle, power output level and operating frequency can be selected by the user.

Transmit Power Test (Modulated Mode)

The modulated Transmit Power Test mode puts the device into a transmit state, generating a signal modulated by a pseudo-random bit stream (PRBS). The duty cycle, power output level and operating frequency can be selected by the user.

Receive Test

The Receive Test mode puts the device in receive mode on a channel selected by the user, and generates a display showing current received signal strength and peak signal strength in dBm.

Oscillator Frequency Test

The Oscillator Frequency Test allows accurate measurements of the crystal oscillator frequency to be taken. When selected, the device is put into a test mode in which the 16-MHz microprocessor clock signal is output as a square wave on digital I/O pin DIO10.

Current Measurement Tests

This option offers a choice of the different modes that affect the device's electrical current consumption. Choosing one of these options allows current measurements to be taken in the corresponding mode.

The modes available are as follows:

- Microcontroller in deep sleep mode without memory retention
- Microcontroller in sleep mode without memory retention
- Microcontroller in sleep mode with memory retention
- Microcontroller in doze mode
- Microcontroller running
- Microcontroller running with protocol domain powered
- Microcontroller running with protocol domain powered and radio transmitting
- Microcontroller running with protocol domain powered and radio receiving

If carrying out sleep current measurements using a JN5148/JN5139 evaluation kit sensor board (DR1048), board current can be minimised by removing R9 to open circuit D9. Leakage current can also be minimised by setting the DIO control lines to the sensors HIGH and the DIO lines to D1 and D2 HIGH.



Note: For the JN5148 and JN5139 devices, the sleep tests with/without memory held use the function **vAHI_FlashPowerDown()** to put the ST Flash device into a low-current mode before the JN5148/JN5139 device enters sleep mode. This may require modification if other Flash devices are used and may require a different function to be called.



Caution: For the sleep mode current tests, depending on which serial terminal is being used, it may be necessary to disconnect the serial dongle and reset the board after the test is complete. This is due to some terminal emulators asserting the RTS signal, which is used by the JN51xx Flash Programmer to signal programming mode, causing the device to jump into programming mode upon start-up.

RF Power Measurement

When the RF Power Measurement option is selected, the JN514x/JN5139 device can be used as a simple RF power meter. This function continuously scans across channels and takes a measurement of the received energy on each channel. After a scan is complete, the highest recorded signal strength is displayed in dBm to an accuracy of about ± 3 dBm.

Trigger Packet Test

The Trigger Packet Test is designed to test the sensitivity of the receive path. In this mode, the device generates a trigger signal which should be connected to an arbitrary waveform generator configured to generate an ideal packet each time it is triggered. The number of trigger pulses generated, and hence the number of packets transmitted from the signal generator, can be configured by the user.

The device counts the number of packets received, and displays the number of packets expected, the number of packets actually received, the number of chip errors detected, and the throughput of the system (as a percentage).

Receive Packets Test

The Receive Packets Test puts the device into a mode that acts as a simple packet sniffer. In this test, the device is put into promiscuous mode, and will display the following information about each packet received:

- Timestamp, in microseconds, showing elapsed time since receiving the previous packet
- Packet type (e.g. Beacon, Data, Acknowledge, Mac Command, Reserved)
- Source address (short or extended)
- Destination address (short or extended)
- Packet checksum (good or bad)

Transmit Packets Test

The Transmit Packets Test transmits a packet at one of two rates – once every second (slow) or approximately once every 20 ms (fast). The packets are of type **data**, and are sent with the source extended address set to the MAC address of the module in use (displayed during the test), and a destination extended address of zero.

The packets have a 4-byte payload that is a 32-bit incrementing number equal to the “packets sent” value displayed during the test.

Connectionless Packet Error Rate Test

The Connectionless Packet Error Rate Test allows the measurement of packet error rate (PER) to be made using two modules running the Customer Module Evaluation Tool. One device must be configured as a slave, and the other device as a master that controls the test and displays the results. The test can be performed either with or without acknowledgments, on any selected channel. If using acknowledgments, the number of retries can be configured.

Packet Error Rate Test

The Packet Error Rate (PER) Test is designed to show real world performance of communications between two boards containing JN514x or JN5139 devices, and as such, this test requires two separate boards, each fitted with the same device type and each programmed with the **AN1132_ModulePerTest_JN51xx.bin** file.

Note: This test is not available on the JN5142 (as it requires simultaneous use of two UARTs).

The first board acts as the PER test master (the test controller) and the second as the PER test slave (the device under test).

Make the following connections between the two boards:

Master	Slave
DIO19 ←	→ DIO20
DIO20 ←	→ DIO19
GND ←	→ GND

These connections link UART 2 on the two boards to each other. This is required for test control purposes.

To start the test:

1. Connect the serial dongle to the slave board.
2. Establish a connection with HyperTerminal and choose “Run as PER test slave” from the Packet Error Rate Test menu.
3. Now disconnect the serial dongle from the slave board and connect it to the master.
4. Establish a connection with HyperTerminal and choose “Run as PER test master” from the Packet Error Rate Test menu to begin the test.

Once the test has started, packets are transmitted from the master to the slave. The slave board keeps a count of the headers and whole packets received - if a whole packet is received successfully, the slave returns an acknowledge packet to the master. The master keeps a count

of the acknowledgements received. 500 packets are sent on each channel and the results are displayed as percentages of successful packets (100% corresponds to all packets received).

Clear Channel Assessment Test

The Clear Channel Assessment (CCA) test is used to obtain the result of a clear channel assessment. The different CCA modes and the threshold for CCA may each be selected.

CCA_MODE	Mode number
ENERGY_DETECT	1
CARRIER_DETECT	2
CARRIER_OR_ENERGY_DETECT	3
CARRIER_AND_ENERGY_DETECT	4

Revision History

Version	Notes
1.0	First release
2.0	Support for JN5139 device added
2.1	Internal version
2.2	Minor modifications made
2.3	PER test application split into separate target CCA test added Support for JN5142 device added
2.4	Support for JN5148-J01 and JN5142-J01 devices added
2.8	Binaries updated

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.

NXP Laboratories UK Ltd
(Formerly 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

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

www.nxp.com/jennic