Freescale Semiconductor
Application Note

AN2952
Rev. 0, 7/2005

Romeo2Remote — Evaluation Software for Windows

by: John Logan
East Kilbride, Scotland

Introduction

Romeo2Remote is a Windows program that communicates with the Romeo2 embedded monitor described in application note AN2818, *Romeo2 Monitor for the MC68HC908AP64 MCU*. It provides a simple way to configure a Romeo2 RF module to allow testing and measurement of its parameters. Romeo2Remote allows the user to control Romeo2 using on-screen controls, or using script files.

Getting Started

To use Romeo2Remote, you will need the following:

- 1 x MC68HC908AP64 demo board (part no. DEMO908AP64)
- 1 x Romeo RF module (part no. MC33591MOD315 or MC33591MOD434 (others may be available))
- 1 x PC running Windows 98, Windows 2000, or Windows XP

freescale™ semiconductor

The MC68HC908AP64 demo board and Romeo RF module must be configured as described in application note AN2818, and must be programmed with the embedded monitor program supplied with application note AN2818.

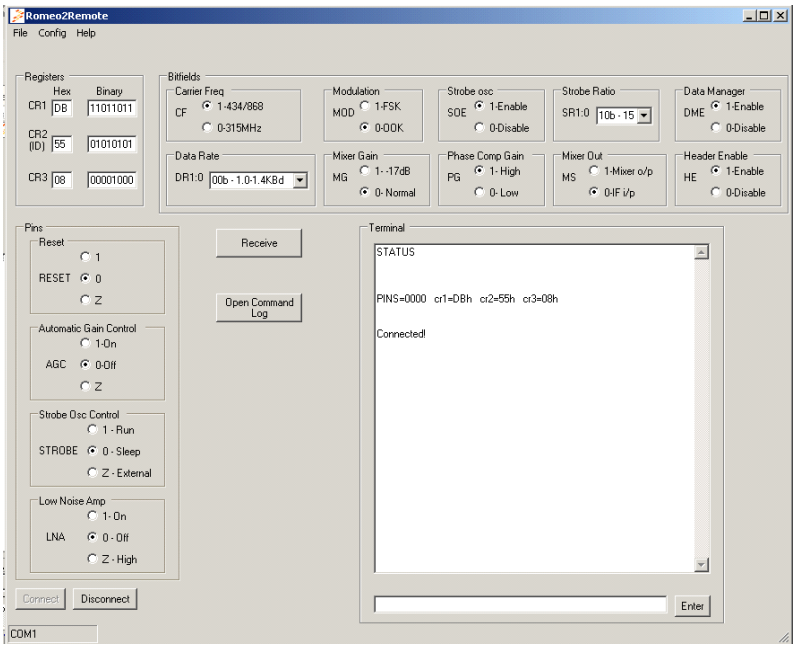

Figure 1. Romeo2Remote Screenshot

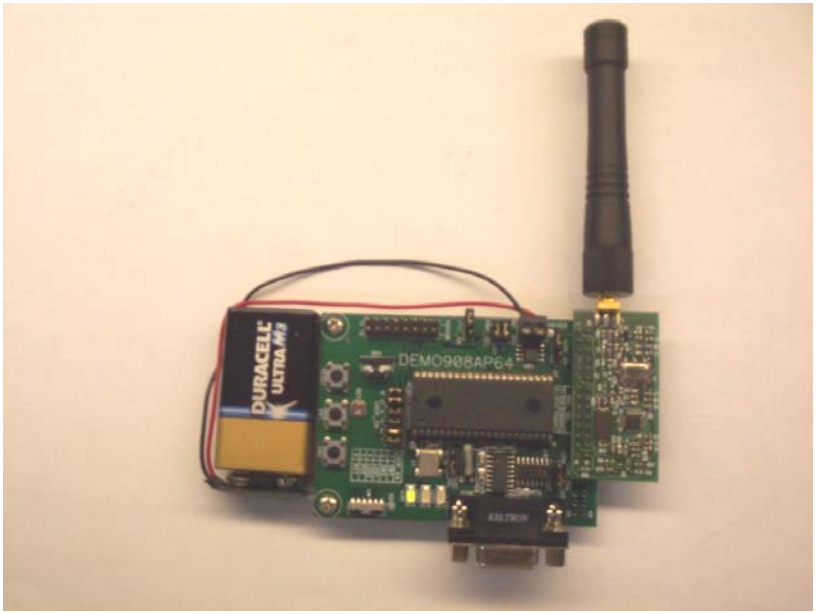

Figure 2. Hardware Setup

Installation

To install Romeo2Remote:

1. Download file AN2952SW.zip from www.freescale.com.
2. Unzip the contents of AN2952SW.zip.
3. Double-click on the Setup.exe file to begin the installation process.
4. Follow the on-screen instructions to install the Romeo2Remote package.

NOTE

The Microsoft .Net framework must be installed for Romeo2Remote to work. This is installed by default on newer versions of Windows XP. Older versions of Windows (98, 2000, and XP) do not have .Net installed by default. The Romeo2Remote installer will check to see if .Net is installed, and will provide a link to a download site if it is not installed.

Startup

The MC68HC908AP64 demo board and Romeo RF module must be configured as described in application note AN2818, and must be programmed with the embedded monitor program supplied with application note AN2818. Figure 2 shows hardware correctly configured.

1. Connect the MC68HC908AP64 demo board to a PC COM port using a serial cable.
2. Start Romeo2Remote from the Windows Start menu using Start->Romeo2Remote. You should now see the window shown in Figure 1.
3. Select the correct COM port using the Config->ComPort... menu option.
4. Click OK. Romeo2Remote will remember the last used COM port.
5. Click the Connect button.

Romeo2Remote should now connect to the MCU board and initialize the Romeo2 RF module. You will see a series of commands and replies in Romeo2Remote's Terminal window.

If communication was successful, you will see “Connected!” in the terminal window at the end of initialization.

If communication was not successful, press the reset button on the MCU board, then click the Connect button on Romeo2Remote again to retry. Typical reasons for communications failure are:

- COM port being used by another application. Shut down any other applications using the COM port, or change COM ports.
- MCU board not programmed with the Romeo2 embedded monitor. Ensure that the MC68HC908AP64 demo board and Romeo RF module are configured as described in application note AN2818.

Controls and Commands

Connect, Disconnect

These buttons are used to connect and disconnect the MCU board to and from the chosen COM port. The selected COM port is shown in the status bar below the buttons.

Registers and Bitfields

The Registers section allows the user to set the values of Romeo2's three internal registers. The values can be set in binary or hexadecimal. Alternatively, the user can set individual bitfields within each register using the radio buttons and pulldown lists in the Bitfields section.

I/O Controls — Reset, AGC, Strobe, LNA

Romeo2's four control pins can be controlled directly by ticking the appropriate boxes on screen. Each pin can be set to logic high, logic low, or high-impedance (“Z”).

The high-impedance setting allows external test equipment such as signal generators to be connected to Romeo2's pins without causing conflict with the MCU's I/O pins.

Terminal

The Terminal area displays the communications between the MCU board and Romeo2Remote. Communications use the commands defined in application note AN2818.

The Terminal area can also be used to send commands to the MCU board; type any Romeo2 monitor command (except the BAUD command) into the text box below the main terminal display and press return (or click the Enter button on-screen). Figure 3 and Figure 4 show an example of sending the Help command, which displays all available Romeo2 embedded monitor commands.

NOTE

Using the BAUD command will change the baud rate used by the MC68HC908AP64 demo board's serial port. This will cause communications with Romeo2Remote to fail.

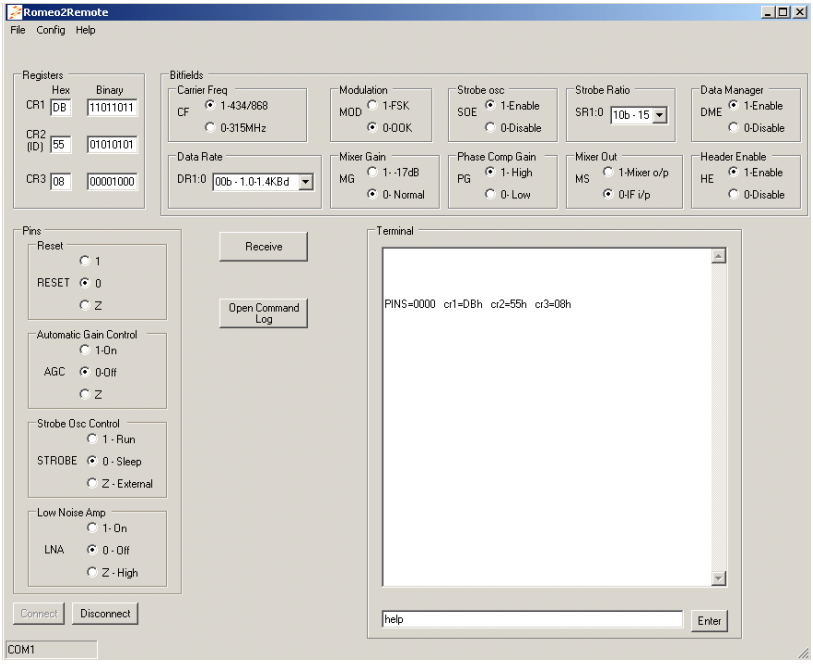

Figure 3. Before Help Command

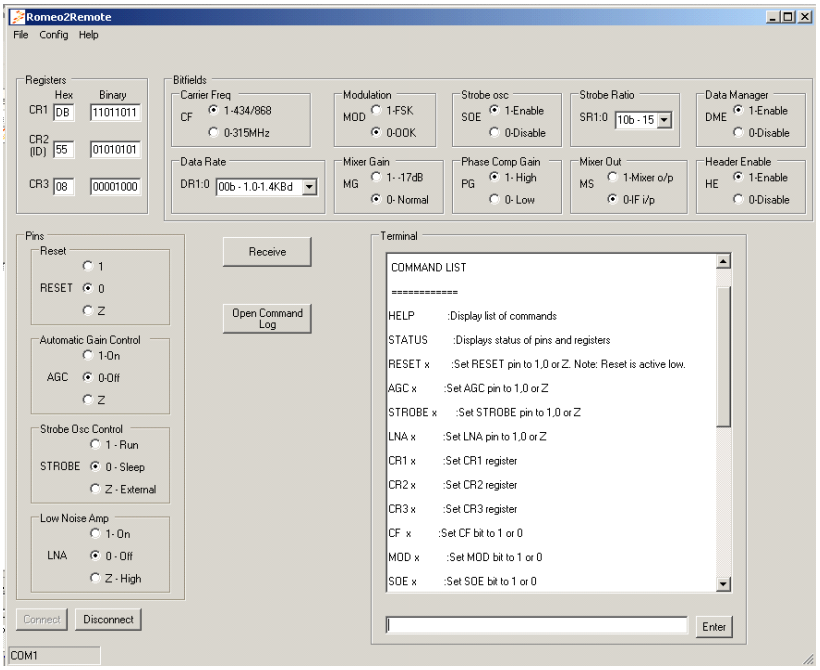

Figure 4. After Help Command

Receive Control

The Receive button allows the user to configure Romeo2 to receive messages. The received data is displayed in the Terminal window.

NOTE

To make Romeo2 receive data, its register contents and I/O pins must be correctly configured to match incoming data. Romeo2Remote is supplied with a selection of example script files, which configure the part to receive different types of message frame.

Open Command Log

This button allows the user to store all commands executed in a script file. The script file can then be loaded at any time to repeat the commands. Command logging can be halted at any time. A more detailed discussion of script files is given below.

Saving the Screen Setup

The current screen configuration can be saved to a script file using the File->Save option. This creates a script file containing a list of commands, which can be reloaded later.

Setting the COM Port

The Config->ComPort... menu item allows the user to set the COM port used to communicate with the MCU board.

NOTE

Romeo2Remote must be disconnected from the MCU board (click the Disconnect button) before this option becomes active.

Script Files

A script file is a text file containing a list of Romeo2 embedded monitor commands. Romeo2Remote can load these files and execute the commands. Figure 5 shows an example of a script file that configures Romeo2 to receive an OOK modulated message.

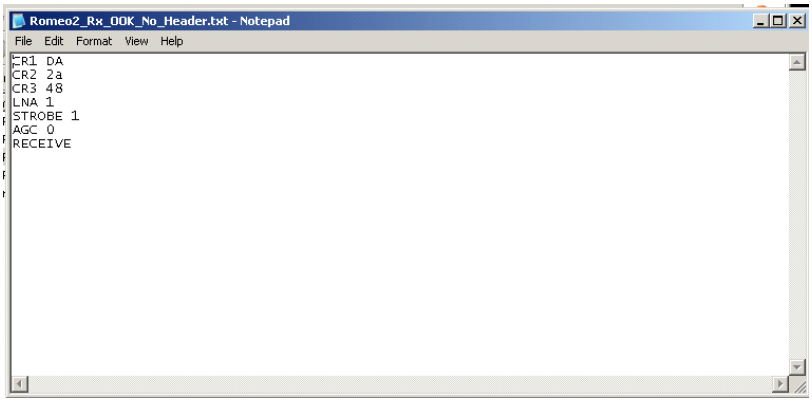

Figure 5. Example Script File

A Romeo2 embedded monitor command can be used in a script file. See application note AN2818 for a full list of commands.

Script files can be executed using the File->Open->Script option.

Script files have a .txt file extension by default. Any text editor (for example, Notepad) can be used to create script files. Romeo2Remote's Open Command Log, and Save options can also be used to generate script files.

NOTE

Using the BAUD command in a script will change the baud rate used by the MC68HC908AP64 demo board's serial port. This will cause communications with Romeo2Remote to fail. Do not use the BAUD command.

Example Script Files

A set of example script files is supplied with Romeo2Remote in file AN2952SW.zip. These script files match the example setups shown in application note AN2818.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN2952
Rev. 0, 7/2005

freescale™
semiconductor