

# An RF Data Link Using Tango3 and Romeo2 RF Modules

by: John Logan  
8/16-bit Division  
East Kilbride

---

## Introduction

This document describes the setup and use of a simple RF link using the MC33493 RF transmitter (Tango3) and the MC33591/2/3/4 family of RF receiver ICs (Romeo2) using Tango3 and Romeo2 RF modules, MC9S08RG60 and MC68HC908AP64 Demo Boards, and the Tango/Romeo software drivers described in Application Note AN2707.

The demo allows the user to set up a simple RF link between a Tango3/MC9S08RG60 transmitter and a Romeo2/MC68HC908AP64 receiver. The transmitter sends a message to the receiver when the user presses a button. The receiver flashes an LED to indicate correct reception of the message. The link can be configured for different frequency bands (315 MHz, 434 MHz, 868 MHz, 915 MHz).

Full source code is supplied to allow the user can modify the program or use it as the basis of other projects.

The reader should be familiar with the Tango3 and Romeo2 device data sheets.

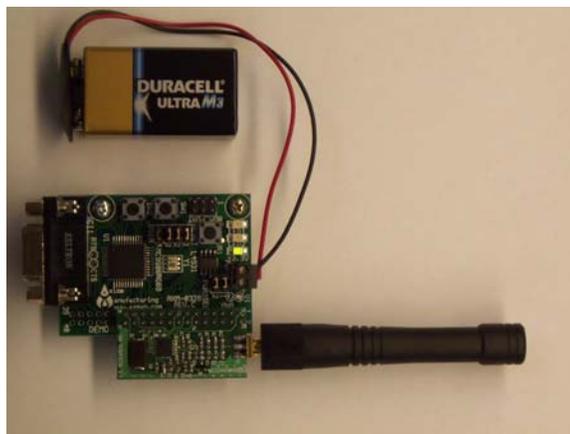


Figure 1. MC9S08RG60 and Tango3 RF Module



Figure 2. MC68HC908AP64 and Romeo2 RF Module

---

## Requirements

To set up the link, you must have the following hardware and software.

- MC9S08RG60 Demo Board (part no. DEMO9S08RG60)
- Tango RF module (part no MC33493MOD315 or MC33493MOD434 — others may be available)
- MC68HC908AP64 Demo Board (part no. DEMO908AP64)
- Romeo2 RF module (part no MC33591MOD315 or MC33591MOD434 — others may be available)

- MetroWerks CodeWarrior version 3.0 or later
- 'AN2783SW.zip' which contains the software files

## Programming Demo into the MC9S08RG60 MCU Using MC9S08RG60 Serial Monitor

The Tango3 demo program must be programmed into the FLASH memory of the MC9S08RG60 Demo Board before it can be used. There are several programming tools available for the HC(S)08 MCU families that can be used for this purpose. This application note describes how to program the MC9S08RG60 using a free evaluation copy of the CodeWarrior development environment and the MC9S08RG60 Demo Board's built-in serial monitor.

To program the MC9S08RG60:

1. Install power select (PWR\_SEL) jumpers 1 and 2 on the MC9S08RG60 Demo Board. Both jumpers must be installed.
2. Install all USER jumpers.
3. Connect the serial port connector on the MC9S08RG60 Demo Board to a PC comm. port using a 9-pin straight-through serial cable.
4. Connect 9V power supply or battery to power connector on MC9S08RG60 Demo Board.
5. Install CodeWarrior.

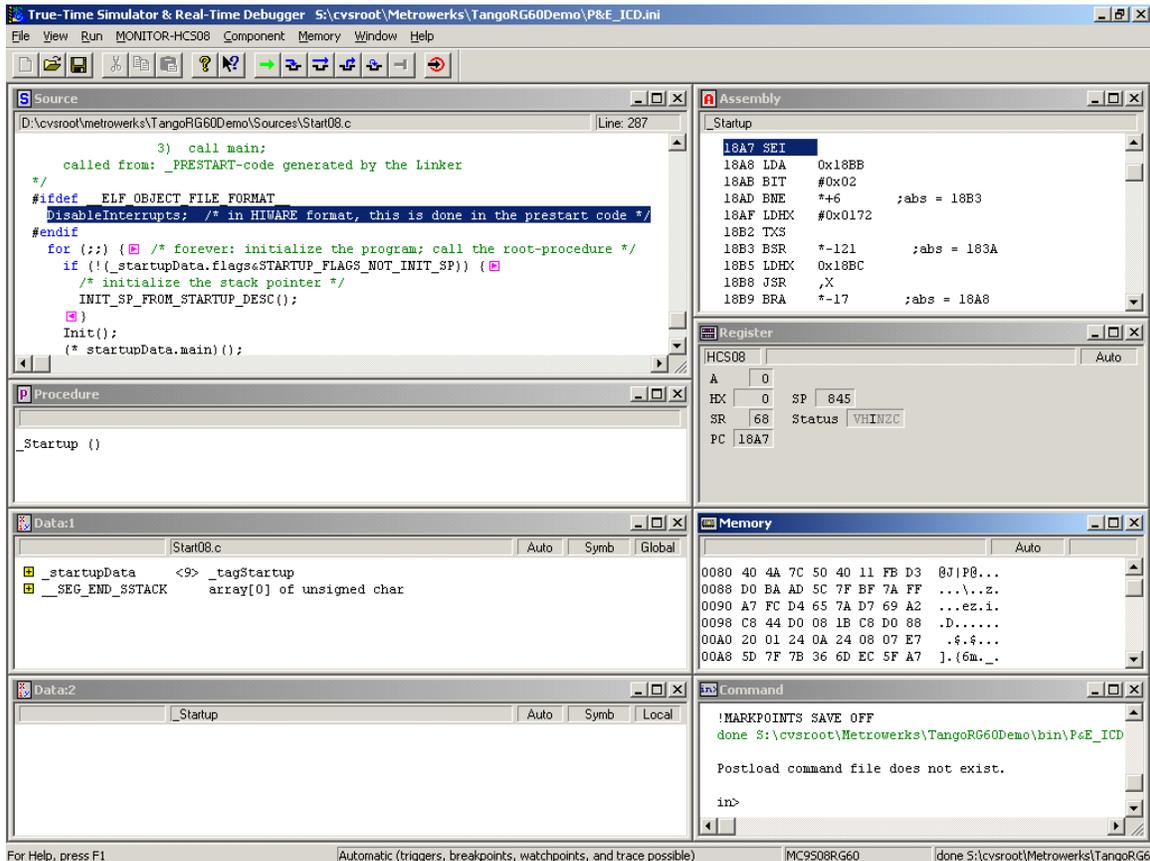
### NOTE

*You must have a copy of the CodeWarrior development tool for HC(S)08 installed. A copy of CodeWarrior is supplied with the MC9S08RG60 Demo Board. Follow the instructions supplied with the MC9S08RG60 Demo Board.*

6. Unzip file Tango3RG60Demo.zip, which is contained within AN2783SW.zip. This unzips a CodeWarrior project containing the programming file for the monitor.
7. Start CodeWarrior (Start menu->MetroWerks->CodeWarrior->CodeWarrior IDE).
8. Select File->Open and open the file Tango3RG60Demo.mcp. This opens a CodeWarrior project.
9. Press and hold the reset and SW1 switches on the MC9S08RG60 Demo Board. Release the reset switch while continuing to hold SW1. Then release SW1. This puts the demo board into 'serial monitor mode', ready to receive data from CodeWarrior.
10. In CodeWarrior, click on file 'main.c' in the Target window. Then press key F5 or select Project->Debug from the menu bar. This launches a debugger, which communicates with the MC9S08RG60 Demo Board and attempts to burn the demo program into FLASH memory on the MCU.
11. If the MCU's FLASH memory is blank, the debugger will program the Tango3 monitor into memory. You should see the screenshot shown in [Figure 3](#), when programming is completed. Go to step 14.
12. If the MCU's FLASH memory is not blank, the debugger will report an error. Click OK in any error windows that appear, then select MONITOR-HCS08->Erase Flash from the menu bar.
13. Now select MONITOR-HCS08->Load from the menu bar. Select file 'Tango3RG60Demo.abs' from the project/bin directory.

## Programming Demo into the MC9S08RG60 MCU Using MC9S08RG60 Serial Monitor

14. The debugger will now program the monitor into the MCU's FLASH memory. You should see screenshot shown in [Figure 3](#), when programming is completed.
15. The monitor program has now been programmed into the MC9S08RG60 Demo Board. CodeWarrior is no longer required. Shut down all CodeWarrior windows and exit the program.



**Figure 3. Screenshot of Debugger after Programming**

## Programming Demo into the MC68HC908AP64 MCU

If the Romeo2 demo program has already been programmed into the MC68HC908AP64 Demo Board, the user can begin to use the monitor (see section 'Using The Software'). Otherwise, the user must follow the programming procedure described below.

1. Install power select (PWR\_SEL) jumpers 1 and 2 on the MC68HC908AP64 Demo Board.
2. Install VST\_EN jumper.
3. Install MON\_EN jumper.
4. Select setting DEBUG on COM\_SEL header.
5. Install jumper across pins 1 and 2 of header OSC\_SEL.

6. Install all USER\_EN jumpers.
7. Connect the serial port connector on the MC68HC908AP64 Demo Board to a PC comm. port using a 9-pin straight-through serial cable.
8. Connect a 9V power supply or battery to the power connector on the MC68HC908AP64 board.
9. Install CodeWarrior.

#### NOTE

*You must have a copy of the CodeWarrior development tool for HC(S)08 installed. A copy of CodeWarrior is supplied with the MC68HC908AP64 Demo Board. Please follow the instructions supplied with the demo board to install CodeWarrior.*

10. Unzip file Romeo2AP64Demo.zip (contained within AN2783SW.zip). This unzips a CodeWarrior project containing the programming file for the monitor.
11. Start CodeWarrior (Start menu->MetroWerks->CodeWarrior->CodeWarrior IDE).
12. Select File->Open and open the file Romeo2AP64Demo.mcp. This opens a CodeWarrior project.
13. In CodeWarrior, click on file 'main.c' in the Target window. Then press key F5 or select Project->Debug from menu. This launches the debugger, which communicates with the MC68HC908AP64 Demo Board and attempts to burn the Romeo2 demo program into FLASH memory on the MCU. The "Attempting to contact target and pass security..." window should appear. Please make sure the following options are configured correctly.
  - a. Target Hardware Type: Class 3
  - b. Serial Port: 1 (Depends on the PC COM Port)
  - c. Baud: 9600 Baud
  - d. Target MCU Security bytes: Check the "IGNORE security failure and enter monitor mode" check box.
14. Click the "Contact target with these settings..." button and follow the instructions on the screen. When the "Erase and Program Flash?" window appears, click the "Yes" button.
15. Follow the on-screen instructions to cycle the MCU power. The "CPROG08SZ Programmer" window should close after the MCU FLASH is programmed.
16. The demo program has now been programmed into the FLASH memory of the MC68HC908AP64. Shut all CodeWarrior windows and exit the program.

---

## USING THE DEMO

### Tango3 Transmitter Hardware Setup

The Tango3 RF module and the MC9S08RG60 Demo Board form an RF transmitter. The hardware should be set up as follows.

1. Connect the Tango3 RF Module to connector J1 on the Mc9S08RG60 Demo Board. Pin 1 on each board must be aligned.
2. Connect an antenna to the Tango3 RF module.

**NOTE**

*Tango3 RF modules are available in a range of frequencies; each is supplied with an appropriate antenna.*

3. Install power select (PWR\_SEL) jumpers 1 and 2 on the MC9S08RG60 Demo Board. Both jumpers must be installed.
4. Connect a 9V power supply or battery to the power connector on the MC9S08RG60 Demo Board.

See [Figure 1](#) for a reference setup.

**Romeo2 Receiver Hardware Setup**

The Romeo2 RF module and MC68HC908AP64 Demo Board form an RF receiver. The hardware should be set up as follows.

1. Connect the Romeo2 RF Module to connector J1 on the MC68HC908AP64 Demo Board. Pin 1 on each board should be aligned.
2. Connect an antenna to the Romeo2 RF module.

**NOTE**

*Romeo2 RF modules are available in a range of frequencies; each is supplied with appropriate antenna.*

3. Install power select (PWR\_SEL) jumpers 1 and 2 on the MC68HC908AP64 Demo Board. Both jumpers must be installed.
4. Remove VST\_EN jumper.
5. Remove MON\_EN jumper.
6. Install jumper across pins 1 and 2 of header OSC\_SEL.
7. Install jumper on COM setting of COM\_SEL header.
8. Install all USER\_EN jumpers.
9. Connect 9V power supply or battery to power connector on MC68HC908AP64 board.

Refer to [Figure 2](#) for a reference setup.

**Configuring the RF Link Frequency Band**

The Tango3 and Romeo2 RF modules are available for various frequency bands (315 MHz, 434 MHz, 868 MHz, 915 MHz). Each module is hard-wired to one frequency band.

**NOTE**

*Please ensure that the Tango3 and Romeo2 modules used are both for the same frequency band.*

For the Tango3/MC9S08RG60 transmitter setup, no steps are required to configure the frequency band.

The Romeo2/MC68HC908AP64 receiver setup requires an additional configuration step, to set up correctly Romeo2's internal registers. (Romeo2 has a register bit to define carrier frequency.)

When the MC68HC908AP64 board powers up, it will flash LED1 to show the user the current frequency configuration. The number of flashes determines the selected frequency band (see [Table 1](#)).

**Table 1. Frequency Band Selection**

Number of Flashes	Frequency Band
1	315 MHz
2	434 MHz
3	868 MHz
4	915 MHz

The default value is 1 (315 MHz).

To change frequency setup to another band:

1. Press and hold the reset button.
2. While holding the reset button, hold down S1.
3. Now release the reset button while holding S1.
4. Let LED flash for number of times required. Release the S1 key during the last flash.

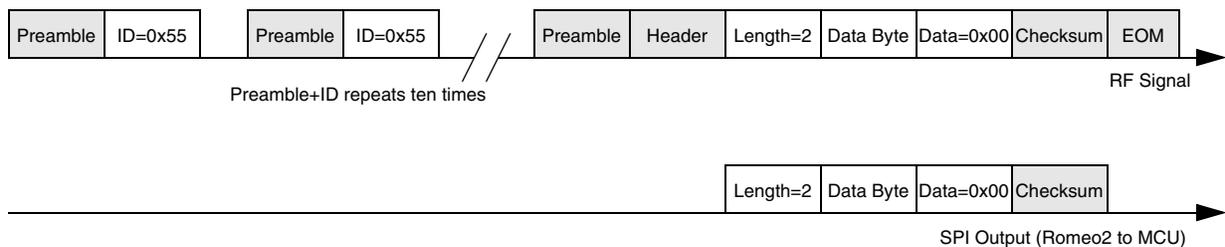
For example, if you wish to choose the 868 MHz band, S1 should flash three times. During the 3rd flash, release S1.

The frequency band value is held in FLASH memory on the MC68HC908AP64, so it will be kept even after removing power from the board.

S1 will flash to indicate the chosen frequency band, each time the board is reset.

### RF Link Description

Pressing switches S1 or S2 on the MC9S08RG60 Demo Board will cause Tango3 to transmit a message. The message uses the 'Header' format described in Application Note AN2707. The message structure is shown in [Figure 4](#).



**Figure 4. Message Frame**

## USING THE DEMO

The message consists of a stream of ten Preamble and ID fields followed by transmission of Preamble, a Header field, a Length byte (fixed value of 2), two data bytes, a checksum byte, and an EOM field.

The ID byte is fixed to 0x55.

The first data byte of the message will increment when S1 pressed, decrement when S2 pressed.

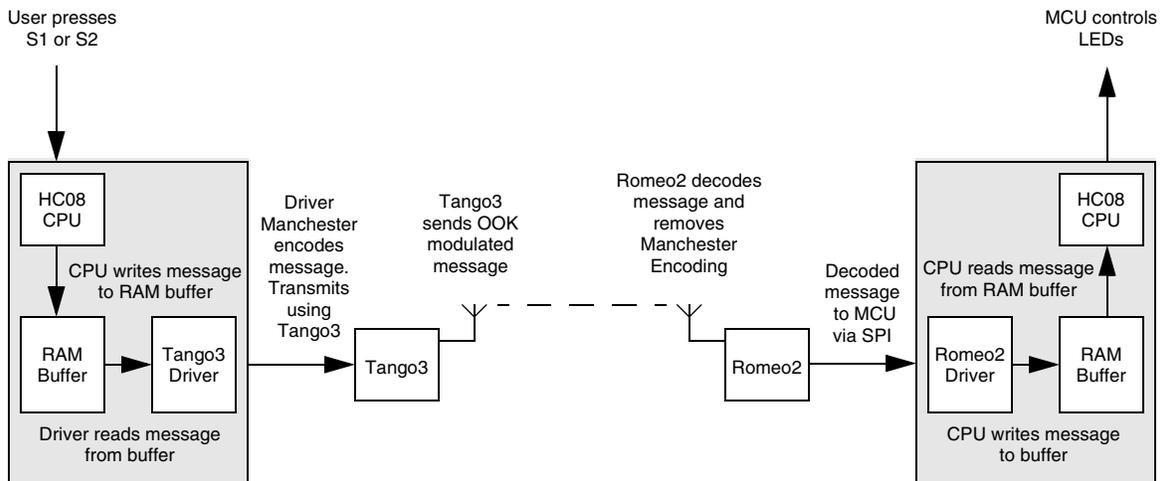
The second data byte is always 0x00.

LED1 on the MC9S08RG60 Demo Board is lit when S1 is pressed; LED2 is lit when S2 is pressed.

Romeo2 receives the message, strips the Preamble, ID, Header and EOM fields and passes the rest to the MC68HC908AP64 via its SPI interface. (See [Figure 4.](#))

The Romeo software driver then checks the message for errors. If there is no error, the message is passed to the MCU (note, the checksum byte is removed by the driver). The MCU then compares the first data byte in the message with the first data byte received in the previous message. If the values are different, it toggles an LED on port pin PTF0. If the values match, the LED is not toggled.

[Figure 5](#) shows the relationship between Tango3, Romeo3, the MCUs and the software drivers.



**Figure 5. Block Diagram of RF Link**

## Modifying the demo

The Tango3 demo program uses a simple structure. It uses the software drivers described in AN2707. The Romeo2 demo is a little more complex; it uses Freescale's Standard Software Driver for HC908 0.5  $\mu\text{m}$  Split Gate FLASH to allow storage of the carrier frequency setting in FLASH memory (these drivers and their documentation can be downloaded from [www.freescale.com](http://www.freescale.com)).

Full source code is supplied for the demo. It is liberally commented and should be easy to understand.

---

## References

1. MC33493 RF Transmitter IC (Tango3) data sheet
2. MC33591 RF Receiver (Romeo2) data sheet

---

## Trademarks

- CodeWarrior<sup>®</sup> is a registered trademark of MetroWerks, Inc., a wholly owned subsidiary of Motorola, Inc.
- MetroWerks<sup>®</sup> and the MetroWerks logo are registered trademarks of MetroWerks, Inc., a wholly owned subsidiary of Motorola, Inc.

## **How to Reach Us:**

### **USA/Europe/Locations not listed:**

Freescale Semiconductor Literature Distribution  
P.O. Box 5405, Denver, Colorado 80217  
1-800-521-6274 or 480-768-2130

### **Japan:**

Freescale Semiconductor Japan Ltd.  
SPS, Technical Information Center  
3-20-1, Minami-Azabu  
Minato-ku  
Tokyo 106-8573, Japan  
81-3-3440-3569

### **Asia/Pacific:**

Freescale Semiconductor H.K. Ltd.  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T. Hong Kong  
852-26668334

### **Learn More:**

For more information about Freescale Semiconductor products, please visit <http://www.freescale.com>

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.  
© Freescale Semiconductor, Inc. 2004.