Freescale Semiconductor
Application Note

Document Number: AN2962
Rev. 1, 05/2007

Echo Monitor for the MC9S08RG60 MCU

by: John Logan
East Kilbride, Scotland

1 Introduction

This document describes how to use the Echo Monitor program to evaluate the performance of the MC33696 RF receiver IC (code name ‘Echo’). It shows the hardware setup required, the steps to set up the monitor, lists all available commands and gives some examples of usage.

The reader should be familiar with the Echo device data sheet and with the RF data formats described in the data sheet.

2 Description

The Echo Monitor is a software program that runs on Freescale’s MC9S08RG60 Demo Board. The MC9S08RG60 Demo Board is connected to an Echo RF module and also to a PC via a serial port. The PC runs a terminal emulation program, for example, Hyperterminal. Figure 1 is a screen shot of

Contents

Hyperterminal showing the monitor program. Figure 2 shows the hardware setup required to use the monitor program.

You can control the pins on the Echo RF module by typing simple commands into the PC. You can also configure any of Echo's internal registers and enable reception of messages.

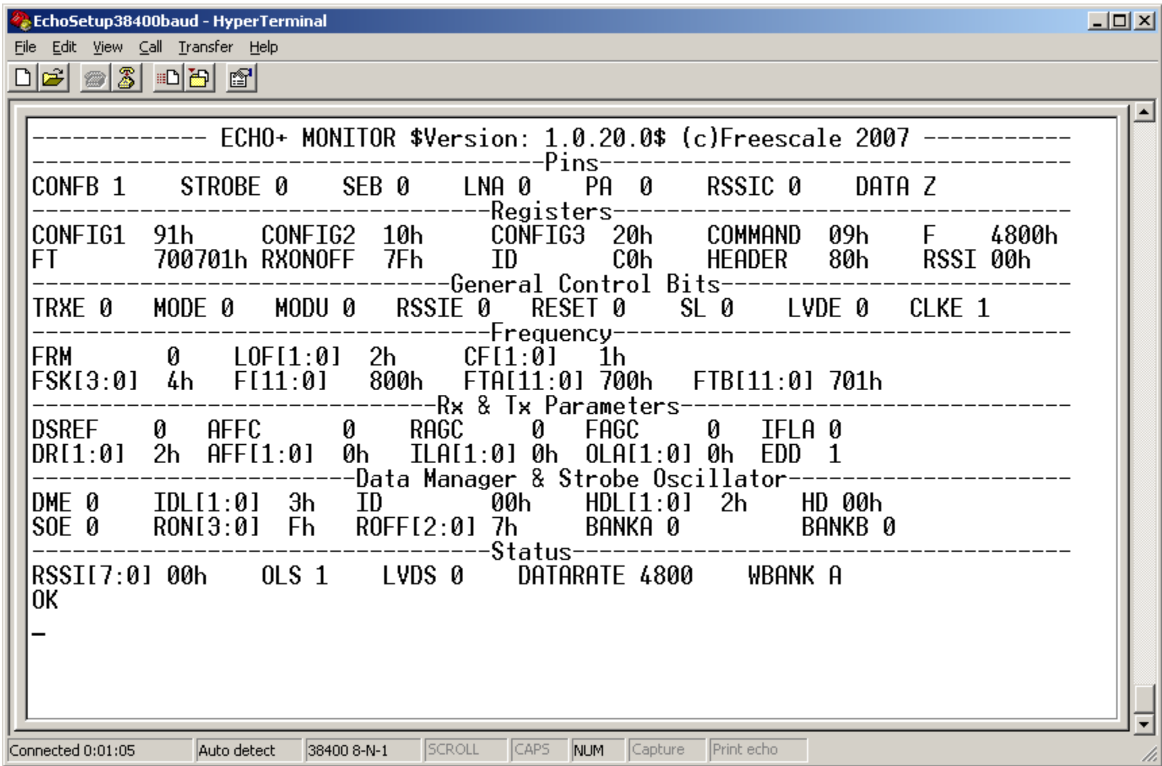

Figure 1. Echo Monitor Communicating with Hyperterminal

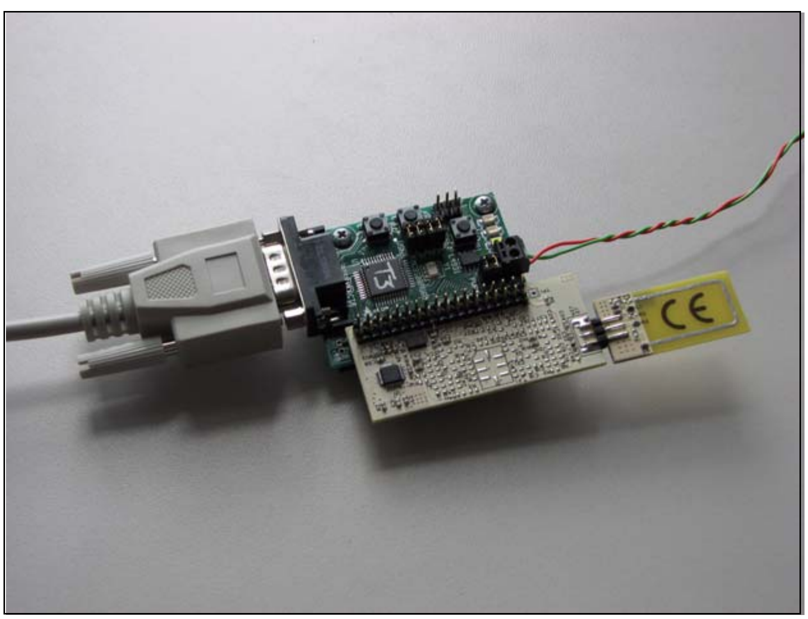

Figure 2. MC9S08RG60 Demo Board and Echo RF Module

3 System Configuration

To use the software, the following hardware and software are required:

- MC9S08RG60 Demo Board (part no. DEMO9S08RG60)
- Echo RF module (part no TBD — beta release)
- CodeWarrior™ version 3.0 or later
- AN2962SW.ZIP, which contains the software files
- A PC with Hyperterminal, or any other terminal program.

Full source code for the Echo monitor is available so the monitor can be extended or tailored if required.

3.1 Programming the Echo Monitor into the MC9S08RG60 MCU Using a MC9S08RG60 Serial Monitor

The Echo monitor program must be programmed into the Flash memory of the MC9S08RG60 demo board before use. 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 Echo monitor into the MC9S08RG60 MCU, perform the following steps:

1. Install CodeWarrior. 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. Please follow the instructions supplied with the MC9S08RG60 demo board.

   NOTE

   You must register CodeWarrior after installation. You will be sent a license file by email after registering. This file must be installed correctly. Please refer to the installation documentation supplied with the CodeWarrior CD

2. Unzip file EchoRG60MonProg.zip, which is contained in AN2952SW.zip. This unzips a CodeWarrior project containing the programming file for the monitor.

   NOTE

   This project does not contain any source code. If you wish to read or modify the Echo monitor source code, you should unzip file EchoRG60MonSource.zip. To modify this code, you require a full license for CodeWarrior which can be purchased from www.metrowerks.com or your local Freescale or Metrowerks representative.

3. Install power select (PWR_SEL) jumpers 1 and 2 on the MC9S08RG60 evaluation board. Both jumpers must be installed.
4. Install all USER jumpers.
5. Connect the serial port connector on the MC9S08RG60 Demo Board to a PC communication port using a 9-pin straight-through serial cable.
6. Connect a 9 V power supply or battery to the power connector on MC9S08RG60 demo board

7. Start CodeWarrior (Start menu->Metrowerks->CodeWarrior->CodeWarrior IDE).
8. Select File -> Open, navigate to the directory where 'EchoRG60MonProg.zip' was unzipped, then open the file 'EchoMonitor.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 the file 'EchoMonitor' 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 Echo monitor program into Flash memory on the MCU.
11. If the MCU's Flash memory is blank, the Echo monitor will be programmed into memory. You should see screen shot shown in Figure 3 when programming completes. Skip to step 15.
12. If the MCU's Flash memory is not blank, the debugger will report an error. Click OK in any error window that appears, then select 'MONITOR-HCS08->Erase Flash' from the menu bar.
13. Now select 'MONITOR-HCS08->Load' from the menu bar. Select file 'EchoMonitor.abs' from the project directory.
14. The debugger will program the monitor into the MCU's Flash memory. You should see the screen shot shown in Figure 3 when programming completes.
15. The monitor program has now been programmed into the MCU demo board. CodeWarrior is no longer required. Shut down all CodeWarrior windows and exit the program.

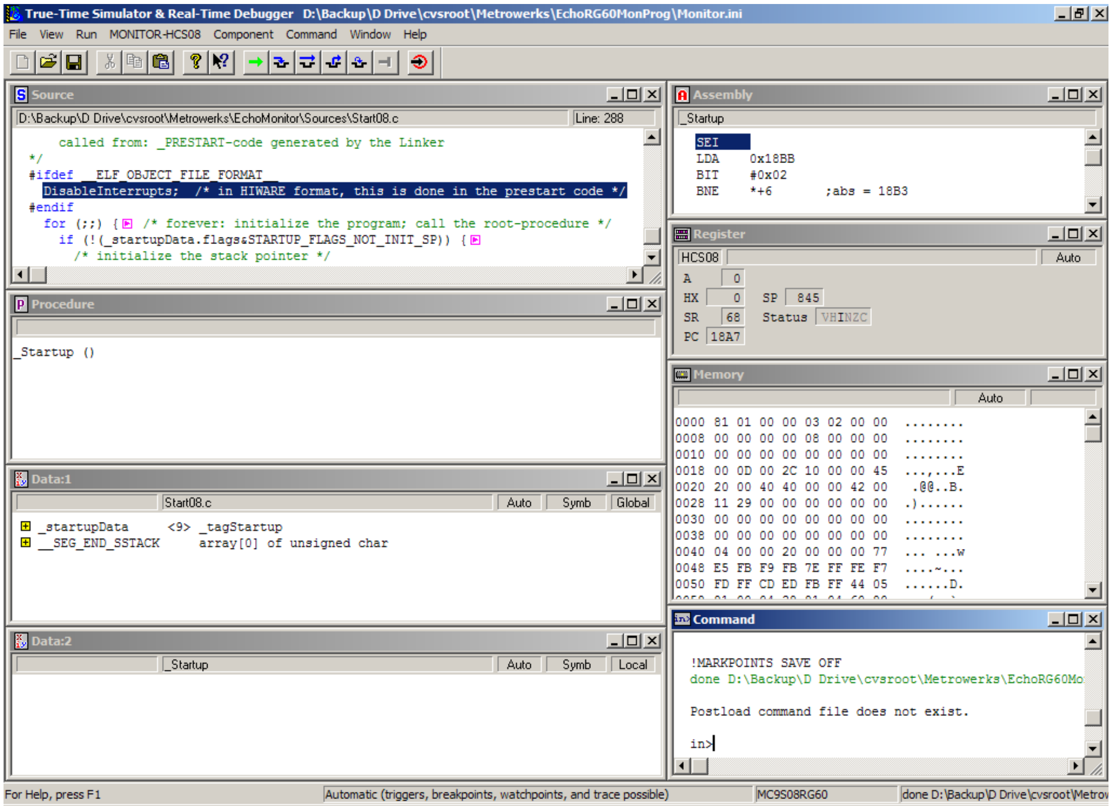

Figure 3. CodeWarrior Debugger After Successful Programming

NOTE

The MC9S08RG60 serial monitor is co-resident in the MC9S08RG60's Flash memory after this operation, so the board can be reprogrammed at a later stage.

4 Using the Echo Monitor

To use the software, connect the MC9S08RG60 demo board to a PC communication port using a straight-through 9-pin serial cable. The MCU board communicates with a terminal emulation program on the PC. Hyperterminal is a common terminal program that is supplied with the Windows operating system; it should be configured for 38400 baud, 8 data bits, no parity, 1 stop bit, no flow control. See Figure 4.

A setup file (EchoSetup38400baud.ht) for Hyperterminal is supplied in the software package for this application note (AN2962SW.ZIP).

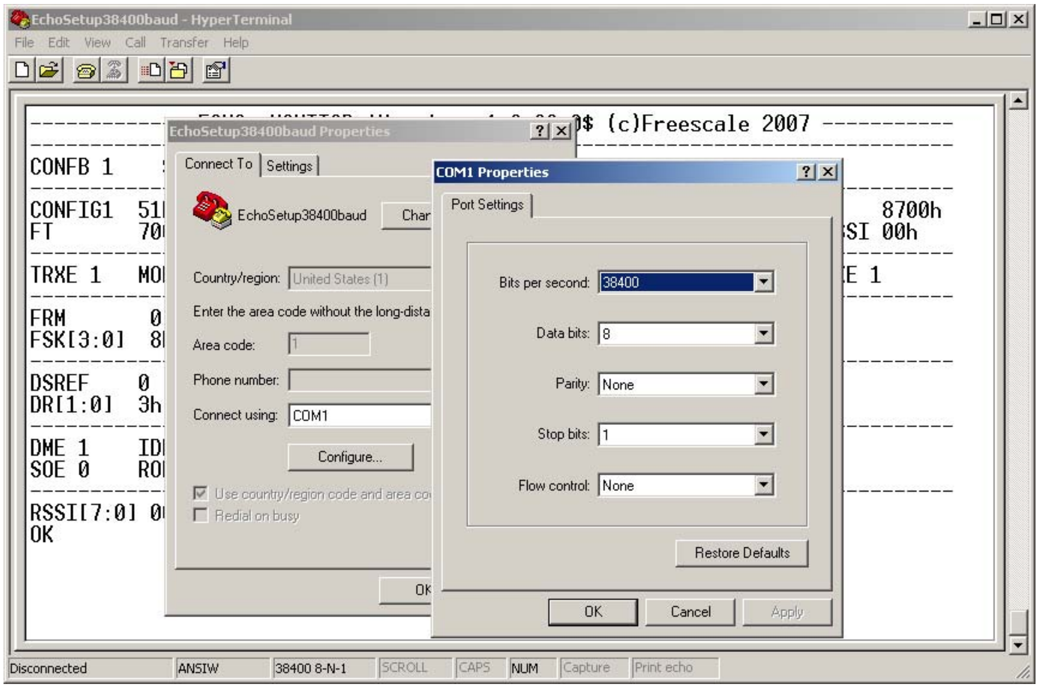

Figure 4. Terminal Program Configuration

To start the monitor:

1. Connect the Echo RF module to connector J1 on the MC9S08RG60 evaluation board. Pin 1 on each board should be aligned.
2. Connect an antenna to the Echo RF module.

NOTE

Echo RF modules are available in a range of frequencies.

3. Install power select (PWR_SEL) jumpers 1 and 2 on the MC9S08RG60 evaluation board. Both jumpers must be installed.
4. Connect the serial port connector on the MC9S08RG60 Demo Board to a PC communication port using a 9-pin straight-through serial cable.
5. Connect a 9 V power supply or battery to the power connector on the MC9S08RG60 demo board. See Figure 2 for a reference setup.
6. Configure the terminal program for 38400 baud, 8 data bits, no parity, 1 stop bit, no flow control (if using Hyperterminal, use ‘EchoSetup38400baud.ht’ setup file).
7. Connect the MC9S08RG60 demo board to the PC communication port using a 9-pin D connector.
8. Start the terminal program (in Hyperterminal, click the ‘Connect’ icon on the tool bar at the top of the screen).
9. Connect power to the MC9S08RG60 demo board.

You should now see the screen shown in Figure 1. If not, press the Reset switch on the MC9S08RG60 demo board.

This screen shows the state of various pins that connect between the Echo RF module and the MC9S08RG60 demo board. It also shows the currently selected data rate for RF transmission.

You can type commands in the terminal program window. Type HELP <return> for a complete list of commands, as shown in Figure 5.

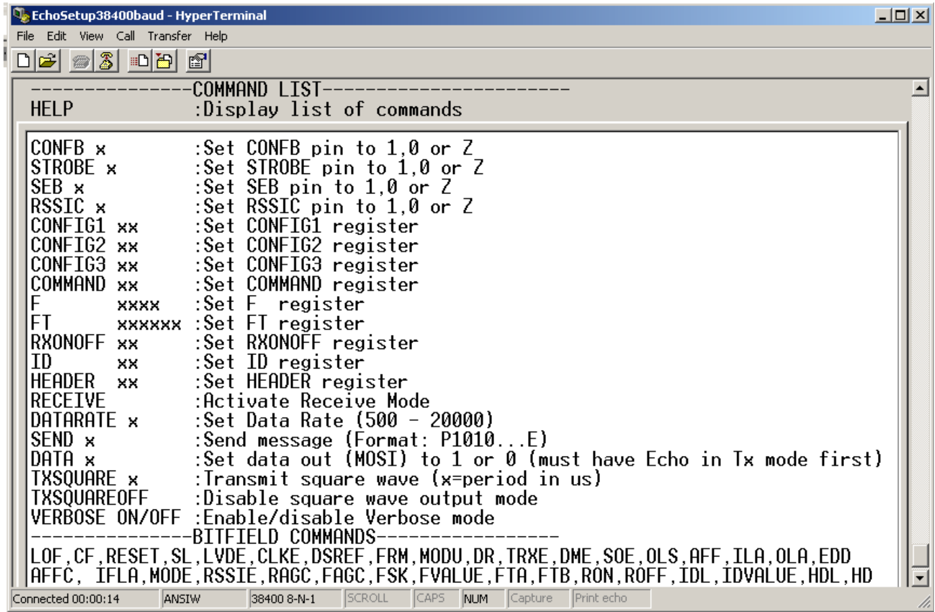

Figure 5. List of Commands

4.1 Command List

The following commands are supported by the Echo monitor.

HELP

Help displays a list of all available commands with a short description of each command.

HELP <return>

CONFB

CONFB allows you to configure the CONFB pin connection to Echo. The pin can be set to an output at logic 1 or logic 0, or set to an input (high impedance state)

CONFB 1 <return>Set CONFB pin to logic 1

CONFB 0 <return>Set CONFB pin to logic 0

CONFB Z <return>Set CONFB pin to input (high impedance state)

STROBE

STROBE allows you to configure the STROBE oscillator pin connection to Echo. The pin can be set to an output at logic 1 or logic 0, or set to an input (high impedance state)

STROBE 1 <return>Set STROBE pin to logic 1

STROBE 0 <return>Set STROBE pin to logic 0

STROBE Z <return>Set STROBE pin to input (high impedance state)

SEB

SEB allows you to configure the Serial Interface Enable pin connection to Echo. The pin can be set to an output at logic 1 or logic 0, or set to an input (high impedance state)

SEB 1 <return> Set SEB pin to logic 1

SEB 0 <return>Set SEB pin to logic 0

SEB Z <return>Set SEB pin to input (high impedance state)

RSSIC

RSSIC allows you to configure the Received Strength Signal Indicator Control pin connection to Echo. The pin can be set to an output at logic 1 or logic 0, or set to an input (high impedance state)

RSSIC 1 <return> Set RSSIC pin to logic 1

RSSIC 0 <return>Set RSSIC pin to logic 0

RSSIC Z <return>Set RSSIC pin to input (high impedance state)

CONFIG1

CONFIG1 allows you to set the value of the CONFIG1 register on Echo. The register can have any value in the range 0x00–0xFF.

CONFIG1 value <return>

CONFIG2

CONFIG2 allows you to set the value of the CONFIG2 register on Echo. The register can have any value in the range 0x00–0xFF.

CONFIG2 value <return>

CONFIG3

CONFIG3 allows you to set the value of the CONFIG3 register on Echo. The register can have any value in the range 0x00–0x3F.

CONFIG3 value <return>

COMMAND

COMMAND allows you to set the value of the COMMAND register on Echo. All control bits in the COMMAND register can be set or cleared.

COMMAND value <return>

F

F allows you to set the value of the frequency register F on Echo. F is a 16-bit register. The register can have any value in the range 0x0000–0xFFFF.

F value <return>

FT

FT allows you to set the value of the frequency register FT on Echo. FT is a 24-bit register. The register can have any value in the range 0x000000–0xFFFFFF.

FT value <return>

RXONOFF

RXONOFF allows you to set the value of the RXONOFF register on Echo. The register can have any value in the range 0x00–0x7F.

ID

ID allows you to set the value of the ID register on Echo. The register can have any value in the range 0x00–0xFF.

ID value <return>

HEADER

HEADER allows you to set the value of the HEADER register on Echo. The register can have any value in the range 0x00–0xFF.

HEADER value <return>

RECEIVE

RECEIVE configures the system to display received RF messages. The CONFB pin on the Echo RF module is set to logic 1, the SEB pin is set to logic 0. If Echo's internal registers have been configured to put the device in receive mode with its data manager enabled, and it receives RF data using any of the data formats listed in the Echo data sheet, these messages will be passed to the MC9S08RG60 via Echo's SPI interface. The MC9S08RG60 will pass these messages to the terminal program via its serial port. Examples of the use of the RECEIVE command are given in the following sections. Pressing the return key stops the reception of messages and puts the CONFB and SEB pins back to the levels they had before the RECEIVE command was executed.

DATARATE

DATARATE allows you to set the data rate for RF transmissions. This can be set to any value in the range 500–20000 bps (bits per second).

DATARATE value <return>Set data rate to 'value'

SEND

SEND allows you to transmit an RF data packet from Echo.

NOTE

Echo's internal registers must have been configured to put the device into transmit mode before using the SEND command.

Examples of the use of the SEND command are given in the following sections.

Data packets can be constructed from the following fields:

P — Preamble. The preamble is a fixed format field that allows Romeo to detect the start of a message. See the Romeo2 data sheet for details.

1 = Sends a logic 1, Manchester encoded

0 = Sends a logic 0, Manchester encoded

H — Sends a logic 1 for one bit time (no Manchester encoding)

L — Sends a logic 0 for one bit time (no Manchester encoding)

E — Sends an EOM (end of message). This is a fixed format field that indicates the end of a message.

These fields can be sent in any combination up to a maximum of 128 fields.

The P, 1, 0, and E fields follow the format defined in the Romeo2 data sheet.

Examples:

P1111111101010101E — Sends single message

P1111111101010101EP1111111101010101E — Sends two back-to-back messages

The H and L fields allow you to send messages without Manchester encoding, or mixed with Manchester encoded messages.

DATA

DATA allows you to configure the level of Echo's MOSI pin when Echo is configured for transmit mode (Echo uses MOSI as a data input pin in transmit mode). The pin can be set to an output at logic 1 or logic 0, or set to an input (high impedance state).

DATA 1 <return> Set DATA pin to logic 1

DATA 0 <return>Set DATA pin to logic 0

DATA Z <return>Set DATA pin to input (high impedance state)

NOTE

DATA has no effect on the MOSI pin if Echo's internal registers are not configured for transmit mode. The CONFB and SEB pins must also be configured for transmit mode (CONFB = 1, SEB = 0) before using DATA.

TXSQUARE

TXSQUARE sends a continuous square wave to Echo's DATA pin. Note that this pin's function is shared with the MOSI pin for Echo's SPI interface. If Echo is configured for Transmit mode, Echo will output the square wave on its RFOUT pin. You can end transmission by pressing <return>.

TXSQUARE x <return>Transmit a square wave to MOSI with a period of x microseconds (μs)

NOTE

TXSQUARE has no effect on MOSI pin if Echo's internal registers are not configured for Transmit mode. The CONFB and SEB pins must also be configured for transmit mode (CONFB = 1, SEB = 0) before using TXSQUARE.

TXSQUAREOFF

DATAOUTOFF halts transmission of square waves on the MOSI pin. This command is useful in script files. When typing commands directly into the monitor, it is not necessary to use this command (see TXSQUARE command definition above).

LOF

LOF allows you to set the value of the Local Oscillator Frequency bits in register CONFIG1. The bits can be set to any value in the range 0x00–0x03.

LOF value <return>

CF

CF allows you to set the value of the Carrier Frequency bits in register CONFIG1. The bits can be set to any value in the range 0x00–0x03.

CF value <return>

RESET

RESET allows you to write the value of the RESET bit in register CONFIG1. Writing 0 to this bit has no effect, writing a 1 will reset all of Echo's registers and counters.

RESET 1 <return> Reset Echo

RESET 0 <return>No effect

SL

SL allows you to set the value of the Switch Level bit in register CONFIG1. The bit can be set to logic 1 or logic 0.

SL 1 <return> Set SL bit to logic 1

SL 0 <return>Set SL bit to logic 0

LVDE

LVDE allows you to set the value of the Low Voltage Detect Enable bit in register CONFIG1. The bit can be set to logic 1 or logic 0.

LVDE 1 <return> Set LVDE bit to logic 1

LVDE 0 <return>Set LVDE bit to logic 0

CLKE

CLKE allows you to set the value of the Clock Enable bit in register CONFIG1. The bit can be set to logic 1 or logic 0.

CLKE 1 <return> Set CLKE bit to logic 1

CLKE 0 <return>Set CLKE bit to logic 0

DSREF

DSREF allows you to set the value of the Dataslicer Reference bit in register CONFIG2. The bit can be set to logic 1 or logic 0.

DSREF 1 <return> Set DSREF bit to logic 1

DSREF 0 <return>Set DSREF bit to logic 0

FRM

FRM allows you to set the value of the Frequency Reference Manager bit in register CONFIG2. The bit can be set to logic 1 or logic 0.

FRM 1 <return> Set FRM bit to logic 1

FRM 0 <return>Set FRM bit to logic 0

MODU

MODU allows you to set the value of the Modulation bit in register CONFIG2. The bit can be set to logic 1 or logic 0

MODU 1 <return> Set MODU bit to logic 1

MODU 0 <return>Set MODU bit to logic 0

DR

DR allows you to set the value of the Data Rate bits in register CONFIG2. The bits can be set to any value in the range 0x00–0x03.

DR value <return>

TRXE

TRXE allows you to set the value of the Transceiver Enable bit in register CONFIG2. The bit can be set to logic 1 or logic 0

TRXE 1 <return> Set TRXE bit to logic 1

TRXE 0 <return>Set TRXE bit to logic 0

DME

DME allows you to set the value of the data manager Enable (DME) bit in register CONFIG2. The bit can be set to logic 1 or logic 0

DME 1 <return> Set DME bit to logic 1

DME 0 <return>Set DME bit to logic 0

SOE

SOE allows you to set the value of the Strobe Oscillator Enable (SOE) bit in register CONFIG2. The bit can be set to logic 1 or logic 0

SOE 1 <return> Set SOE bit to logic 1

SOE 0 <return>Set SOE bit to logic 0

ILA

ILA allows you to set the value of the Input Level Attenuation bits in register CONFIG3. The bits can be set to any value in the range 0x00–0x03.

ILA value <return>

OLA

OLA allows you to set the value of the Output Level Attenuation bits in register CONFIG3. The bits can be set to any value in the range 0x00–0x03.

OLA value <return>

AFFC

AFFC allows you to set the value of the Average Filter Frequency Control bit in register COMMAND. The bit can be set to logic 1 or logic 0.

AFFC 1 <return> Set AFFC bit to logic 1

AFFC 0 <return>Set AFFC bit to logic 0

IFLA

IFLA allows you to set the value of the IF Level Attenuation bit in register COMMAND. The bit can be set to logic 1 or logic 0.

IFLA 1 <return> Set IFLA bit to logic 1

IFLA 0 <return>Set IFLA bit to logic 0

MODE

MODE allows you to set the value of the MODE bit in register COMMAND. The bit can be set to logic 1 or logic 0.

MODE 1 <return> Set MODE bit to logic 1

MODE 0 <return>Set MODE bit to logic 0

RSSIE

RSSIE allows you to set the value of the RSSI Enable bit in register COMMAND. The bit can be set to logic 1 or logic 0.

RSSIE 1 <return> Set RSSIE bit to logic 1

RSSIE 0 <return>Set RSSIE bit to logic 0

RAGC

RAGC allows you to write the value of the Reset AGC bit in register COMMAND. The bit can be set to logic 1 or logic 0.

RAGC 1 <return> Set RAGC bit to logic 1

RAGC 0 <return>Set RAGC bit to logic 0

FAGC

FAGC allows you to write the value of the Freeze AGC bit in register COMMAND. The bit can be set to logic 1 or logic 0.

FAGC 1 <return> Set FAGC bit to logic 1

FAGC 0 <return>Set FAGC bit to logic 0

FSK

FSK allows you to set the value of the FSK bits in register F. The bits can be set to any value in the range 0x00–0x0F.

FSK value <return>

FVALUE

FVALUE allows you to set the value of the F bits in register F. The bits can be set to any value in the range 0x000–0xFFF.

FVALUE value <return>

FTA

FTA allows you to set the value of the FTA bits in register FT. The bits can be set to any value in the range 0x000–0xFFF.

FTA value <return>

FTB

FTB allows you to set the value of the FTB bits in register FT. The bits can be set to any value in the range 0x000–0xFFF.

FTB value <return>

RON

RON allows you to set the value of the RON bits in register RXONOFF. The bits can be set to any value in the range 0x0–0xF.

RON value <return>

ROFF

ROFF allows you to set the value of the ROFF bits in register RXONOFF. The bits can be set to any value in the range 0x0–0x7.

ROFF value <return>

IDL

IDL allows you to set the value of the ID Length bits in register ID. The bits can be set to any value in the range 0x0–0x3.

IDL value <return>

IDVALUE

IDVALUE allows you to set the value of the ID bits in register ID. The bits can be set to any value in the range 0x0–0x3F.

IDVALUE value <return>

HDL

HDL allows you to set the value of the Header length bits in register HEADER. The bits can be set to any value in the range 0x0–0x3.

HDL value <return>

HD

HD allows you to set the value of the Header bits in register HEADER. The bits can be set to any value in the range 0x0–0x3F.

HD value <return>

AFF

AFF allows you to set the value of the Average Filter Frequency bits in register CONFIG3. The bits can be set to any value in the range 0x0–0x3.

AFF value <return>

AFFC

AFFC allows you to write the value of the Reset Average Filter Frequency select bit in register COMMAND. The bit can be set to logic 1 or logic 0.

AFFC 1 <return> Bits AFF1:0 set average filter cutoff frequency

AFFC 0 <return>Bits DR1:0 set average filter cutoff frequency

EDD

EDD allows you to write the value of the Envelope Detector Decay bit in register COMMAND. The bit can be set to logic 1 or logic 0.

EDD 1 <return> Fast decay

EDD 0 <return>Slow decay

WBANK

Defines working bank, all bit commands apply to working bank. Feedback screen shows working bank bits and registers.

WBANK A <return>All bit commands apply to Bank A

Feedback screen shows Bank A bits and registers

WBANK B <return>All bit commands apply to Bank B

Feedback screen shows Bank B bits and registers

COPY

Copies content of all registers to an opposite bank,

COPY A <return>Copy Bank A content to Bank B

COPY B <return>Copy Bank B content to Bank A

BANKA

BANKA and BANKB bits allow you to set the bank switching function of Echo. See datasheet for details. BANKA and BANKB bits can be set to logic 1 or logic 0

BANKA 1 <return>Set BANKA bit to logic 1

BANKA 0 <return>Set BANKA bit to logic 0

BANKB

BANKA and BANKB bits allow you to set the bank switching function of Echo. See datasheet for details. BANKA and BANKB bits can be set to logic 1 or logic 0

BANKB 1 <return>Set BANKB bit to logic 1

BANKB 0 <return>Set BANKB bit to logic 0

VERBOSE

The VERBOSE command determines the amount of information displayed by the Echo monitor. By default, Verbose is set to On, resulting in the detailed display shown in Figure 1. Setting Verbose to Off reduces the amount of information shown. See Figure 6.

VERBOSE ON <return> Full screen display

VERBOSE OFF <return>Reduced screen display

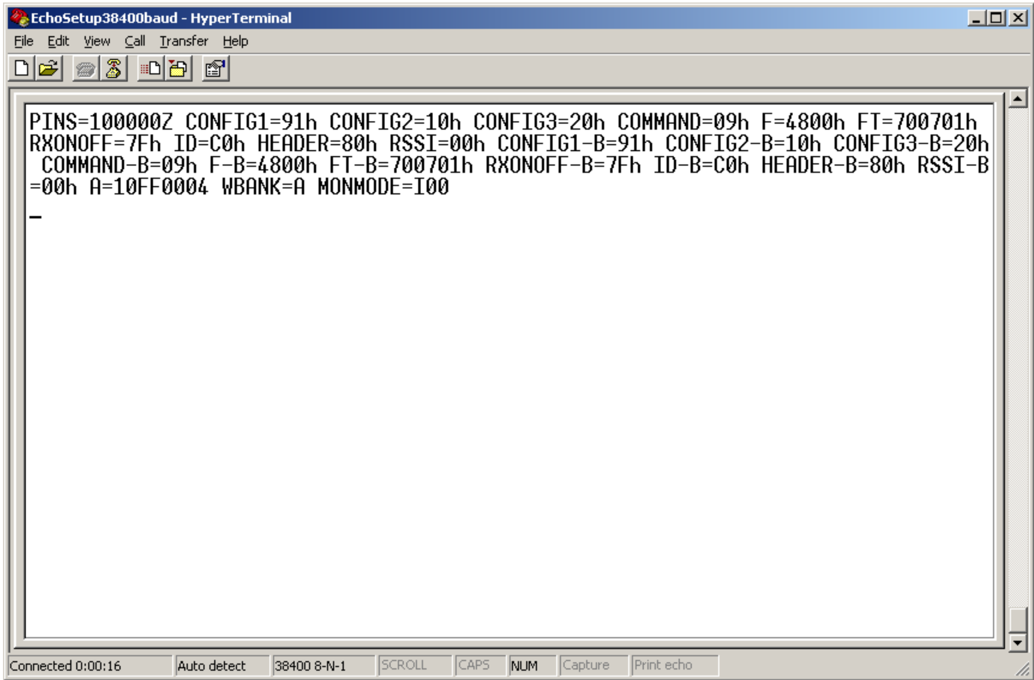

Figure 6. Echo Monitor Display with Verbose Off

4.2 Echo Message Formats

The Echo monitor allows you to send and receive messages using the data format defined in the Echo data sheet, when Echo's data manager is enabled. Figure 7 shows a basic messages following this format.

NOTE

The monitor can also be used to construct message with different formats. See the SEND command description above.

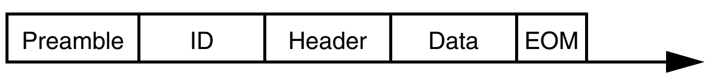

Figure 7. Basic Message Formats

The fields in the message are described below:

- **Preamble** — The preamble is a fixed format field that allows Echo to detect the start of a message. The Echo monitor can transmit Preambles with the correct format using the 'P' field
- **ID** — Each Echo device can be assigned an ID number in its ID register. It will receive messages with this particular ID only. This allows each Echo device in an RF network to have a unique ID. The Echo monitor can construct and send any ID byte using the '1' and '0' fields.
- **Header** — The header field is a fixed format field. It notifies Echo that message data is next. The field's value is defined in Echo's HEADER register. The Echo monitor can construct and send headers using the '1' and '0' fields
- **Data** — Data can be any length and consists of Manchester encoded 1's and 0's. The Echo monitor can construct any data pattern required.

- **EOM** — End of Message. This is a fixed format field that indicates the end of a message. The Echo monitor can transmit EOMs with the correct format using the 'E' field.

The ID, header and data fields are constructed from Manchester encoded 1's and 0's. A Manchester encoded bit is represented by a sequence of two opposite logic levels. A 0 bit of data is encoded as sequence '01', a 1 bit of data is encoded as sequence '10'. Figure 8 shows the data sequence '11001' using Manchester encoding.

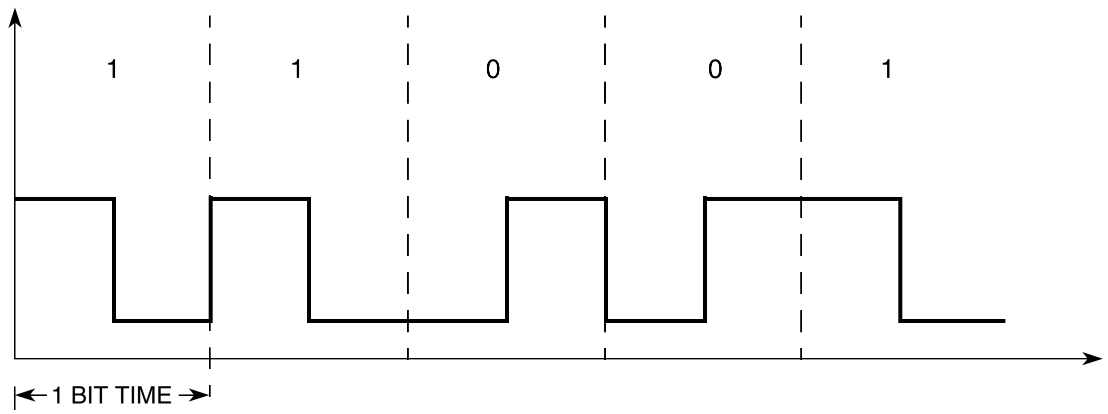

Figure 8. Manchester Encoding

5 Command Examples

This sections gives examples of how to use the monitor commands to perform some simple tasks.

5.1 Transmitting an OOK Message Frame

Here is an example of transmitting an OOK modulated message. The message ID is 0x07. The message Header is 0x6. The message data is 0xF0F0 (binary 1111000011110000). The data rate is 4800 bps. The carrier frequency is 434 MHz.

1. Configure Echo's registers using the following sequence of commands:

   DATARATE 4800

   CONFIG1 50

   CONFIG2 CC

   CONFIG3 00

   COMMAND 39

   F 07E5

   FT 700000

   ID C7

   HEADER 86

2. Send the message frame using the following command:

   SEND P000111011011110000111100000E

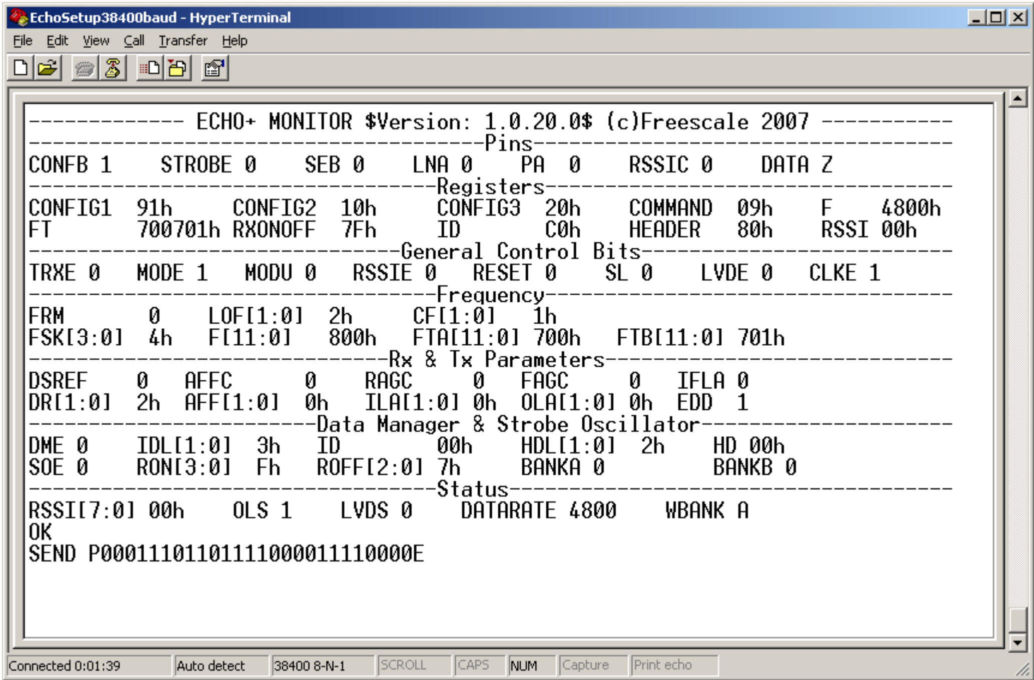

Figure 9. Transmitting an OOK Message

3. To transmit the message again, press the return key. You can send a different message by repeating step 3 using a different bit pattern (configuring the registers again is not necessary.)

5.2 Receiving an OOK Message Frame

Here is an example of receiving an OOK modulated message. The message ID is 0x7. The message Header is 0x6. The data rate is 4800 bps. The carrier frequency is 434 MHz.

1. Configure Echo's registers using the following sequence of commands:

   STROBE 1

   DATARATE 4800

   CONFIG1 50

   CONFIG2 CF

   CONFIG3 00

   COMMAND 19

   F 07E5

   FT 700000

   ID C7

   HEADER 86

2. Enable Echo to receive message frames using the following command:

   RECEIVE

NOTE

When receiving messages where the number of data bits is an exact multiple of eight, Echo may add an extra byte to the end of the message.

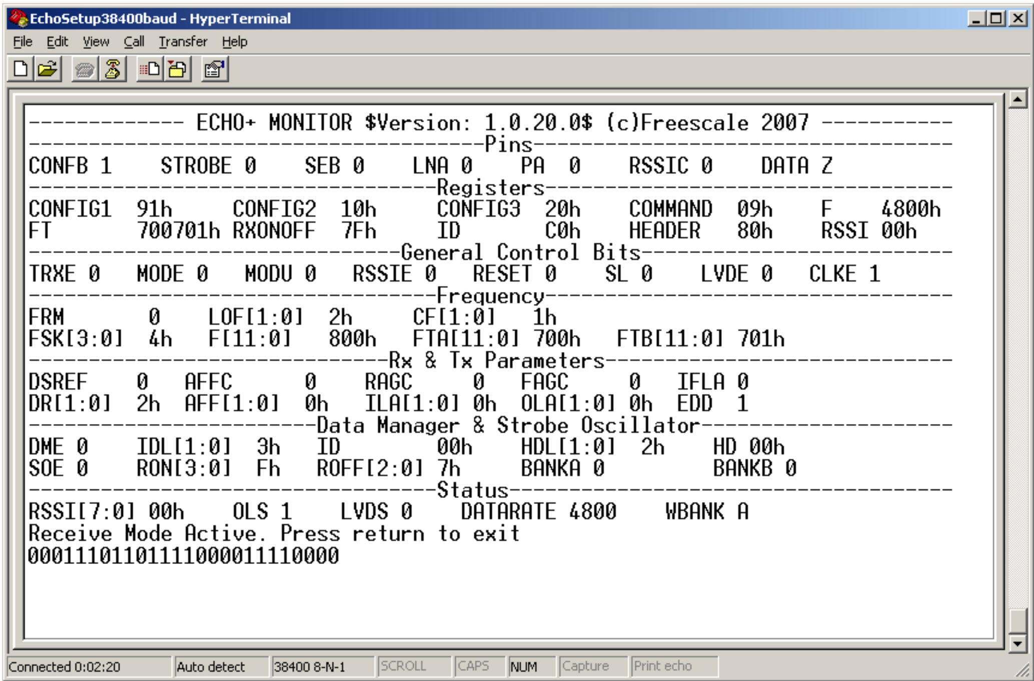

Figure 10. Receiving an OOK Message

5.3 Transmitting an FSK Message Frame

Here is an example of transmitting an FSK modulated message. The message ID is 0x07. The message Header is 0x6. The message data is 0xF0F0 (binary 1111000011110000). The data rate is 4800 bps. The carrier frequency is 434 MHz.

1. Configure Echo's registers using the following sequence of commands:

   DATARATE 4800

   CONFIG1 50

   CONFIG2 EF

   CONFIG3 00

   COMMAND 39

   F 07E5

   FT 6F170F

   RXONOFF 7F

   ID C7

   HEADER 86

2. Send the message frame using the following command:

   SEND P0001110110111100001111 0000E

3. To transmit the message again, press the return key. You can send different message by repeating step 3 using a different bit pattern (configuring the registers again is not necessary).

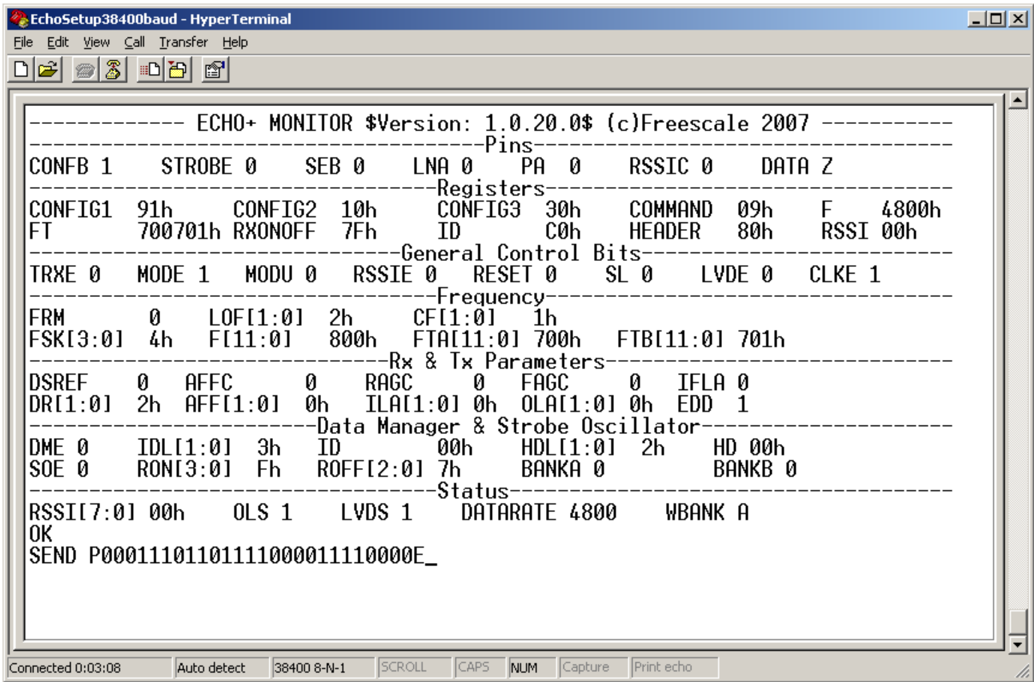

Figure 11. Transmitting an FSK Message

5.4 Receiving an FSK Message Frame

Here is an example of receiving an FSK modulated message. The message ID is 0x7. The message Header is 0x6. The data rate is 4800 bps. The carrier frequency is 434 MHz.

1. Configure Echo’s registers using the following sequence of commands:

   STROBE 1

   CONFIG1 50

   CONFIG2 EF

   CONFIG3 00

   COMMAND 19

   F 07E5

   FT 6F170F

   RXONOFF 7F

   ID C7

   HEADER 86

2. Enable Echo to receive message frames using the following command:

   RECEIVE

NOTE

When receiving messages where the number of data bits is an exact multiple of eight, Echo may add an extra byte to the end of the message.

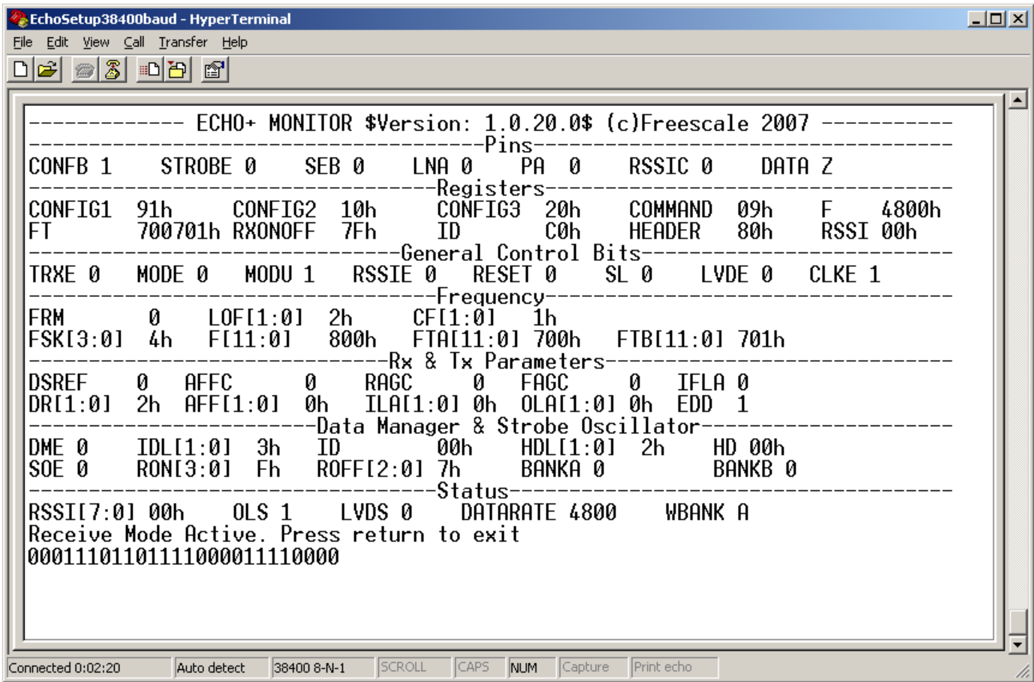

Figure 12. Receiving an FSK Message

5.5 Transmitting a Continuous Signal

Here is an example of transmitting a continuous logic 1 signal. This is useful for making transmit power measurements on the RF output of Echo. The carrier frequency is 434 MHz. The modulation type is OOK.

1. Configure Echo’s registers using the following sequence of commands:

   CONFIG1 50

   CONFIG2 CC

   CONFIG3 00

   COMMAND 39

   F 07E5

   FT 700000

   CONFB 1

   SEB 0

   DATA 1

   To halt, clear the MODE bit to exit transmit mode

5.6 Transmitting a Square Wave

Here is an example of transmitting a square wave with a period of 100 µs. The carrier frequency is 434 MHz. The modulation type is FSK.

1. Configure Echo's registers using the following sequence of commands:

   CONFIG1 50

   CONFIG2 EC

   CONFIG3 00

   COMMAND 39

   F 07E5

   FT 6f070f

   CONFB 1

   SEB 0

   TXSQUARE 100

   To halt, press return.

6 Using Hyperterminal Script Files

Hyperterminal allows you to send and receive text files via the serial port. This means you can put lists of commands in a text file and load them into Hyperterminal. This avoids the need to type long strings of commands for common tests. Figure 13 shows how to find Hypterminal's Send text command. The Echo monitor is supplied with some example text files to try. Before attempting to use them, please read the following paragraph.

Hyperterminal will read commands from text files at a very fast rate, which may be too fast for the Echo monitor to process. Therefore, you must configure Hyperterminal to add some delay between commands. This is done in the ASCII Setup dialog box, found by accessing File menu -> Properties, then clicking the Settings Tab, then clicking the ASCII Setup Button. See Figure 14.

The Echo monitor is supplied with a Hyperterminal setup file, EchoSetup38400baud.ht. This file sets the ASCII Setup transfer speeds for correct operation.

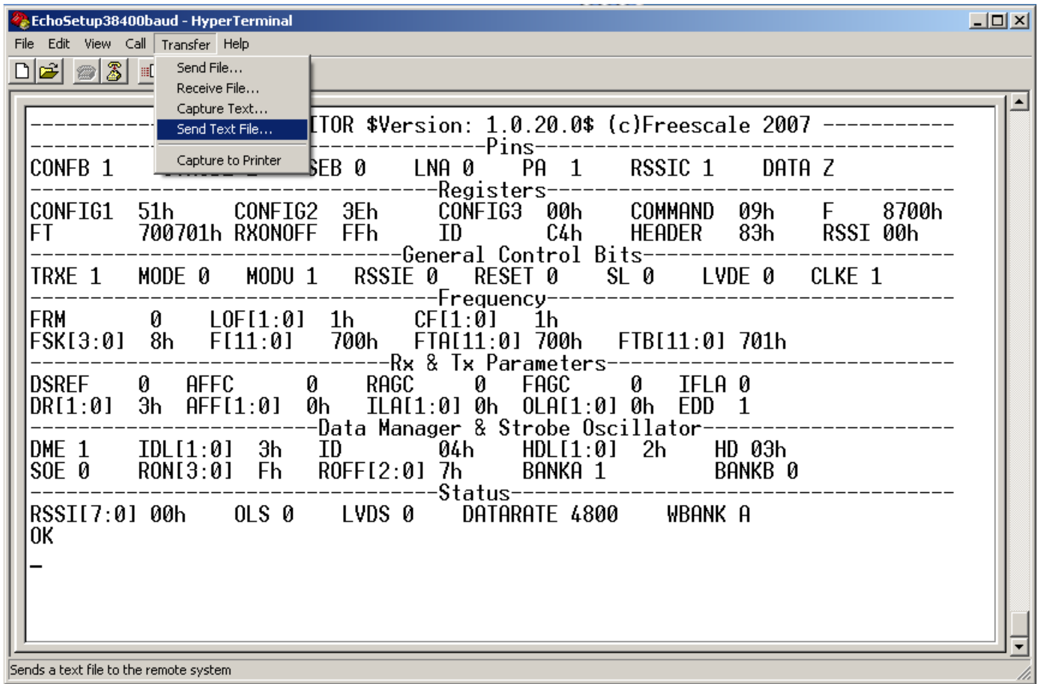

Figure 13. Hyperterminal's Send Text File Command.

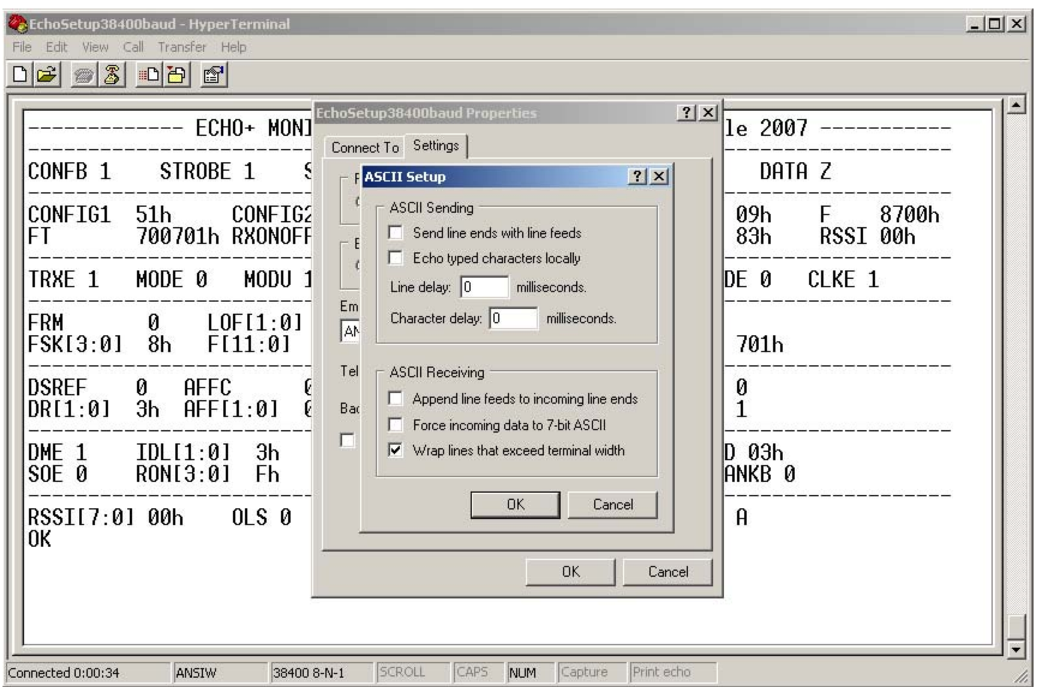

Figure 14. Configuring ASCII Send Delays

7 Echo Monitor Source Code

The Echo monitor is written in the C programming language. Full source code is supplied in file EchoRG60MonSource.zip.

8 Modifying Echo Monitor

The Echo monitor program has a simple structure. It has been written to allow easy porting to other MCUs. The code is liberally commented and should be easy to understand.

Decoding of commands typed into the PC terminal program is done in the SCIRx() routine. This routine is called each time the MCU receives a character from the PC keyboard. It stores characters in a buffer until a carriage return is detected. It then decodes the buffer and performs the required function for each message.

When configuring the internal registers on Echo, the monitor uses the MCU's SPI peripheral. It also uses the SPI when the RECEIVE command is used to receive data from Echo.

The monitor uses the following MCU resources:

- SPI1
- SCI1
- Timer1
- I/O pins
- Flash
- RAM

The code could be ported to another MCU by changing the register definitions, and some other definitions at the start of the code.

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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

Document Number: AN2962
Rev. 1
05/2007