AVR IAR C-SPY®

ROM-Monitor Supplement

COPYRIGHT NOTICE

DISCLAIMER

TRADEMARKS

First edition: February 2001

Part number: CSAVRR-1

WELCOME

Welcome to the AVR IAR C-SPY® ROM-Monitor Supplement, which describes how to use the IAR C-SPY ROM-monitor for the Smart Card Development Kit.

This guide should be read in conjunction with the *AVR IAR Embedded Workbench™ User Guide*, which describes how to use the IAR C-SPY simulator for the AVR.

ABOUT THIS GUIDE

This guide contains the following information:

Introduction describes how to install and run the ROM-monitor.

Controlling user applications contains information about breakpoints and how to resolve problems with the ROM-monitor.

C-SPY configuration describes how to set options from the AVR IAR Embedded Workbench and from the command line.

Advanced topics contains information about transparent commands, the linker file, and the memory layout file.

Diagnostic messages lists and describes the warning and error messages that the AVR IAR C-SPY ROM-monitor may produce.

THE OTHER GUIDES

The other guides describing the AVR IAR development tools are as follows:

AVR IAR Embedded Workbench™ User Guide

This guide provides information about running the IAR Systems tools from the IAR Embedded Workbench interface, and contains complete reference information about the IAR Embedded Workbench commands and dialog boxes.

The guide also describes how to use C-SPY for the AVR microcontroller, and provides reference information about the features of C-SPY.

AVR IAR Compiler Reference Guide

This guide provides programming information about the AVR IAR C Compiler. It includes reference information about the C library functions and language extensions, and provides information about support for the target-specific options such as memory models.

You should refer to this guide for information about the C language when writing and debugging C source programs.

This guide also includes a list of the C compiler diagnostic messages.

AVR IAR Assembler, IAR XLINK Linker™, and IAR XLIB Librarian™ Reference Guide

This guide provides reference information about the AVR IAR Assembler and IAR XLINK Linker for use with the IAR Embedded Workbench.

The assembler programming sections include details of the assembler source format, and reference information about the assembler operators, directives, and mnemonics.

The IAR XLINK Linker programming reference sections provide information about the IAR XLINK Linker commands and output formats.

The IAR XLIB Librarian programming sections provide information about the IAR XLIB Librarian commands.

Finally, the guide includes a list of diagnostic messages for each of these tools.

ASSUMPTIONS AND CONVENTIONS

ASSUMPTIONS

This guide assumes that you already have a working knowledge of the following:

◆ The AVR IAR Assembler language.

◆ The AVR microcontroller.

◆ Your Smart Card Development Board, including its memory layout.

◆ The procedures for using menus, windows, and dialog boxes in a windows environment.

CONVENTIONS

This user guide uses the following typographical conventions:

Style	*Used for*
`computer`	Text that you type in, or that appears on the screen.

Style	*Used for*
`parameter`	A label representing the actual value you should type as part of a command.
`[option]`	An optional part of a command.
`{a | b | c}`	Alternatives in a command.
bold	Names of menus, menu commands, buttons, and dialog boxes that appear on the screen.
reference	A cross-reference to another part of this guide, or to another guide.
🔧	Identifies instructions specific to the versions of the IAR development tools for the IAR Embedded Workbench.
🖥	Identifies instructions specific to the command line versions of IAR development tools.

CONTENTS

INTRODUCTION

This supplement describes the features of the AVR IAR C-SPY ROM-monitor version. It should be read in conjunction with the *AVR IAR Embedded Workbench™ User Guide*. Please note that a few additional features may have been added to the software after the *AVR IAR Embedded Workbench™ User Guide* was printed.

This supplement assumes that you already have some working knowledge of the Smart Card Development Board (SCDB) that you are using.

GETTING STARTED

RUNNING THE DEMO PROGRAM

To run the supplied demo program proceed as follows:

1. Connect the computer to the SCDB, using one of the supported communication channels.
2. Connect the board's power supply and turn it on.
3. Start C-SPY and open the file `tutor\csavr-ccr\demo.d90` for downloading.
4. Set the processor option `-v3`.
5. Click **Step** to run to the first instruction.

Note: If you are not starting C-SPY from the IAR Embedded Workbench, or if you start C-SPY without any project opened, you set the C-SPY processor option and board-specific parameters by typing the command line options in the **Session Options** dialog box. For more information, see *Setting command line options*, page 7.

COMPILING, LINKING, AND RUNNING C-SPY FROM THE IAR EMBEDDED WORKBENCH

In the `tutor\csavr-ccr` directory there is a program called `demo.c` and `demo_two.c` which will serve as our demo source. To compile and run this program from the Embedded Workbench proceed as follows:

1. Create a new project by choosing **New...** from the **File** menu.
2. Name the project `testproj`, for example.

3 Choose **Files...** from the **Project** menu and add the files `demo.c` and `demo_two.c` to the project. Check that the files are in the project tree in the `testproj.prj` window.

4 Double-click `demo.c` in the Project window to open the file in the editor for viewing.

5 Select the **Debug** node of the project tree and choose **Options...** from the **Project** menu.

6 Choose **General** in the **Category** list, set the processor option `-v3` and select the small memory model.

7 Choose **XLINK** in the **Category** list, select the **Include** page and set the path to the extended linker command file for your board, `lnkCC.xcl` that is found in the `tutor\csavr-ccr` directory. Then select the **Output** page and select the **Debug info with terminal I/O** option.

8 Choose **C-SPY** in the **Category** list, and select the `ROM-monitor CC` driver.

9 Set the port and baud rate for C-SPY; for additional information, see *Communication setup*, page 9.

10 Click **OK** to leave the **Options** dialog box.

11 Compile and link the program by clicking on the **Make** button or selecting **Make** from the **Project** menu.

12 Now start C-SPY by clicking on the **C-SPY** button or selecting **Debugger** from the **Project** menu.

13 Choose **Step** from the **C-SPY Execute** menu.

14 Click on the **Stop** button to stop execution.

DIFFERENCES BETWEEN C-SPY VERSIONS

The following table summarizes the key differences between the ROM-Monitor and simulator versions of C-SPY:

C-SPY ROM-Monitor	*C-SPY simulator*
Only OP-fetch breakpoints	OP-fetch and data breakpoints
Execution in real time	Not real time
Real interrupts	Simulated interrupts

C-SPY ROM-Monitor	*C-SPY simulator*
No cycle counter	Cycle counter
Real peripherals	Code coverage, profiling
Memory access according to the device	

CONTROLLING USER APPLICATIONS

This chapter describes the procedures for controlling user applications using the ROM-Monitor.

THE STOP FEATURE

When you press the **Stop** button, C-SPY will stop executing and the execution on the target device will also stop. To continue execution, press **Step** or **Go**.

DEBUGGING IN REAL TIME

This section describes restrictions to programs in a real-time environment.

To ensure that the program is executed in real time set **Realtime ON** in the **Control** menu. When real-time checking is enabled C-SPY does not execute the following commands in the Execute menu that require the CPU to be halted: **Step**, **Step Into**, **Autostep**, and **Go Out** (and their toolbar button equivalents).

If the function stack, **Calls**, is enabled, and the Terminal I/O window is open, the following warnings will be displayed:

```
Warning[20]: Calls disabled in realtime
```

The function stack and terminal I/O will automatically be turned off until the **Realtime** command is set back to **OFF**.

Real-time checking is off by default.

CPU HALT

When C-SPY is executed in non-real-time mode (default) the ROM-Monitor CPU will often be temporarily halted for information exchange with C-SPY. This is, however, not seen by the user.

Note: This affects the interrupt timing, and interrupts that occur during the halt periods could be lost. The execution speed will also be reduced.

Only the **Go** and **Goto Cursor** commands with the function stack disabled (**Calls OFF**) execute the program without halting the CPU, with full interrupt support.

RESOLVING ROM-MONITOR PROBLEMS

This section includes suggestions for resolving problems when debugging with C-SPY in conjunction with a ROM-Monitor.

VERIFYING THE DOWNLOAD

Use the **Target consistency check** options on the **ROM-Monitor** page of the C-SPY options: **Verify boundaries** (`-c1`) or **Verify all** (`-c2`) to verify that the SCDB memory is writable and mapped in a consistent way. The **Verify all** option verifies every byte after loading, and so is considerably slower but more thorough. For more information see *Target-dependent settings*, page 11.

CHECKING FOR PROBLEMS

Reload the application with the **Suppress load** (`-n`) and **Verify all** (`-c2`) options set on the **ROM-Monitor** page of the C-SPY options. This will verify whether the memory contents have changed, or whether the program is self-modifying. For more information see *Target-dependent settings*, page 11.

For problems concerning the operation of the SCDB, refer to the documentation supplied with it, or contact your hardware distributor.

POSSIBLE PROBLEMS

The following list gives details of the most common problems:

Monitor works, but application will not run
The application is probably linked to some illegal code area (like the interrupt table). Do not use the original `.xcl` file delivered with the compiler without changing the start addresses of `CODE` and `DATA` segments.

No contact with the monitor
Make sure that the communication channels are working properly.

If the serial cable is damaged or of the wrong type, C-SPY might use the wrong port or the wrong speed on the computer when trying to communicate with the ROM-monitor. For more information, see *Communication setup*, page 9.

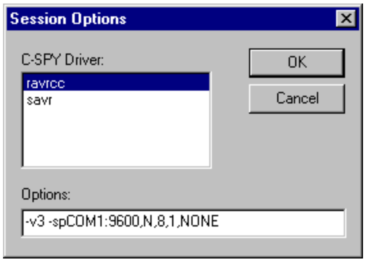
C-SPY CONFIGURATION

This section describes the additional options in the Embedded Workbench, and the command line options used by the ROM-Monitor version of C-SPY.

SETTING COMMAND LINE OPTIONS

The command line options can be given to C-SPY at startup, when opening a debug file for downloading, or by the macro `__processorOption` in the `execUserInit` macro; see the chapters *C-SPY macros* and *C-SPY command line options* in the *AVR IAR Embedded Workbench™ User Guide,* or the on-line help in C-SPY.

When opening a debug file for downloading the following dialog box is displayed to allow you to select the right C-SPY driver if you have more than one driver installed on your computer. The name of the ROM-Monitor driver is **ravrcc**.



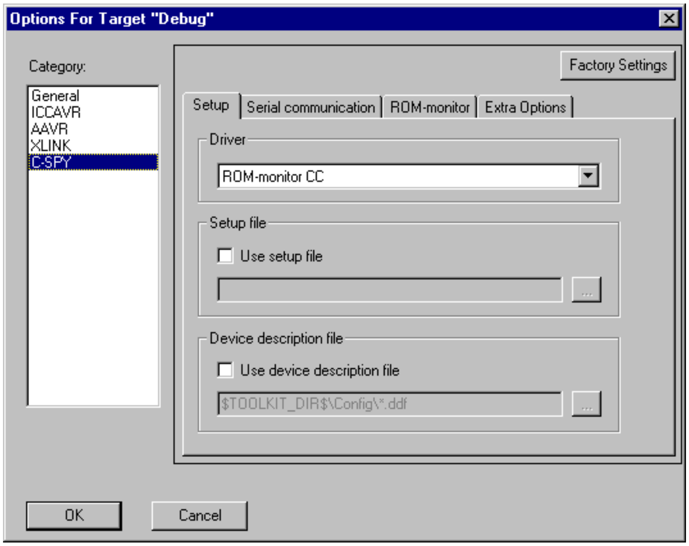
C-SPY Driver: ravrcc (selected), savr

Options: -v3 -spCOM1:9600,N,8,1,NONE

In the **Options** field you can edit the command line options; see *Target-dependent settings*, page 11.

SETTING C-SPY OPTIONS FROM THE EMBEDDED WORKBENCH

To set up the C-SPY options for a certain project in the Embedded Workbench, select the top node in the project tree, choose **Options...** from the **Project** menu, and select **C-SPY** in the **Category** list.



Driver: ROM-monitor CC

Setup file

☐ Use setup file

Device description file

☐ Use device description file

$TOOLKIT_DIR$\Config*.ddf

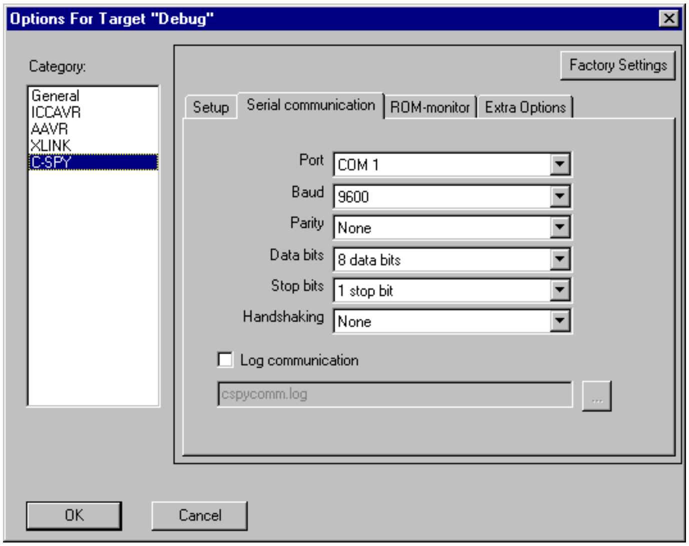
Select the appropriate driver from the **Driver** list box. In this case select **ROM-monitor CC**.

For a description of the **Setup file** and **Device description file** options, see the chapter *C-SPY options* in the *AVR IAR Embedded Workbench™ User Guide*.

COMMUNICATION SETUP

SETTING UP THE SERIAL PORT FROM THE EMBEDDED WORKBENCH

In the **Serial communication** page, the serial port to be used is set up.



Port: COM 1
Baud: 9600
Parity: None
Data bits: 8 data bits
Stop bits: 1 stop bit
Handshaking: None
☐ Log communication
cspycomm.log

C-SPY tries connecting with the selected baud rate when making the first contact with the ROM-monitor board.

For trouble-shooting purposes, there is a possibility to log all characters sent between C-SPY and the ROM-monitor to a file. If you check the **Log communication** dialog box the file `cspycomm.log` will be used in the current working directory.

SETTING UP THE SERIAL PORT FROM THE COMMAND LINE

Syntax

```
-spPORT[:BAUD[:PARITY[,BITS[,STOP[,HANDSHAKING]]]]]
```

Parameters	*Description*
`PORT`	One of the supported ports: `COM1`, `COM2`, `COM3`, `COM4`
`BAUD`	One of following speeds: `300`, `600`, `1200`, `2400`, `4800`, `9600`, `19200`, `38400`, `57600`, `115200` (default `9600`)

Parameters	*Description*
`PARITY`	Only `N` (None) is allowed.
`BITS`	Only 8 is allowed.
`STOP`	1 or 2 stop bits (default 1).
`HANDSHAKING`	`NONE` or `RTSCTS` (default `NONE`).

If this option has not been given, C-SPY will try using the COM1 port at 9600 baud. Of course the EVB has to support the requested baud rate.

Example
To use COM1 at 38400 baud, add the following to the C-SPY command line:

```
-spCOM1:38400,N,8,1,NONE
```

SETTING UP THE SERIAL PORT LOG FILE FROM THE COMMAND LINE

Syntax	*Description*
`-sl LOGFILE`	Logs the communication between C-SPY and the ROM-monitor to the specified log file.

OPTIMIZING DOWNLOADS

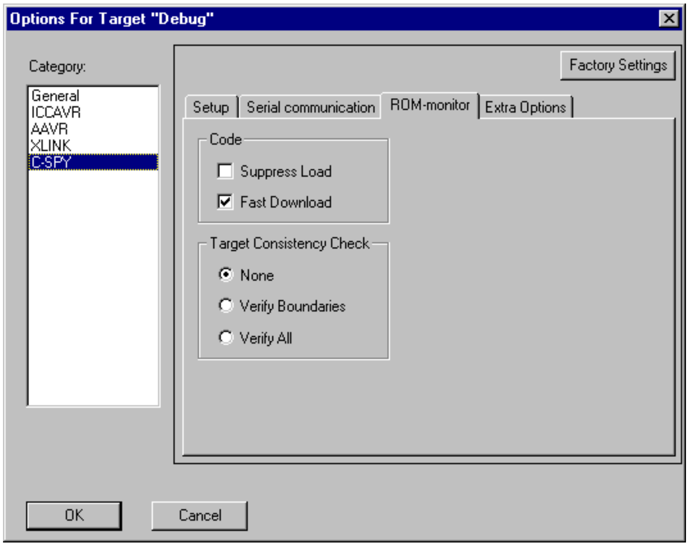
C-SPY uses a special fast download mode that runs with very little protocol overhead. The ROM-monitor must be fast enough to handle the incoming stream using full error checking on the memory or it will fail. If fast download fails, a warning message is given by C-SPY. The download will then restart using a slower (but safer) communication protocol.

If fast download fails constantly, there are a couple of things you can do:

- Lower the transmission speed (if possible).
- Use RTS/CTS handshaking between C-SPY and the ROM-monitor.
- Disable fast downloads. In the Embedded Workbench, choose **Options...** from the **Project** menu, then uncheck **Fast download** on the **ROM monitor** page. On the command line use the `-rz` option.

TARGET-DEPENDENT SETTINGS

The **ROM-Monitor** page contains all the ROM-monitor-specific options. For each option the corresponding command line option is given in parentheses.



Options For Target "Debug"

Category:
General
ICCAVR
AAVR
XLINK
C-SPY

Code
☐ Suppress Load
☑ Fast Download

Target Consistency Check
● None
○ Verify Boundaries
○ Verify All

SUPPRESS LOAD (-n)

Disables the downloading of code which can be time-consuming, but creates C-SPY tables internally. This command is useful if you need to exit C-SPY for a while and continue without loading code. The implicit RESET performed by C-SPY at startup is not disabled though.

FAST DOWNLOAD (-rz)

This flag enables fast downloading of user code. This option is checked by default, which means that fast downloading is enabled. If you uncheck it, downloading will take more time since the error-free protocol is used. However, this should only be necessary if the ROM-monitor is not fast enough to process the data stream or, for example, if the communication cable is insufficiently shielded.

For additional information, see *Optimizing downloads*, page 10.

VERIFY BOUNDARIES (-c1)

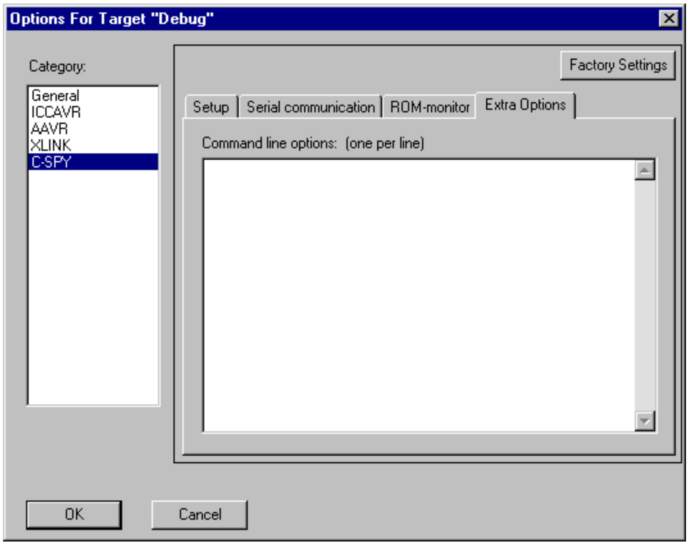
Verifies that the memory on the SCDB is writable and mapped in a consistent way. A warning message will be generated if there are any problems during download.

VERIFY ALL (-c2)

Verifies download. Similar to the **Verify boundaries** (`-c1`) option, but checks every byte after loading to verify that the hardware (SCDB) is OK.

EXTRA COMMAND LINE OPTIONS

Use the **Extra options** page to set additional command line options that are not available as Embedded Workbench options.



Enter one command line option per line.

ADVANCED TOPICS

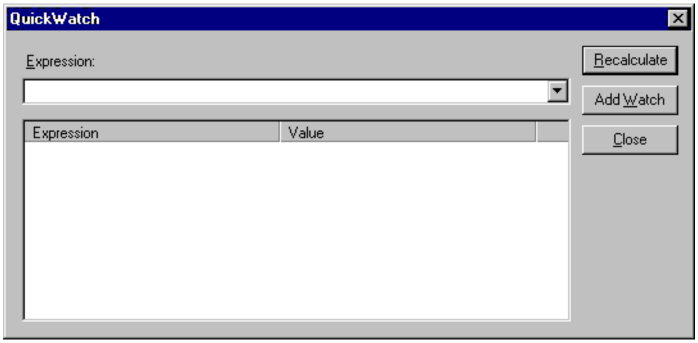
This chapter describes more advanced use of the ROM-monitor, such as using transparent commands and modifying the memory layout file.

USING TRANSPARENT COMMANDS

Transparent commands can be sent directly to the ROM-monitor using the predefined macro `__transparent` (*`commandstring`*). In this way, you can communicate directly with the ROM-monitor without having to go through C-SPY.

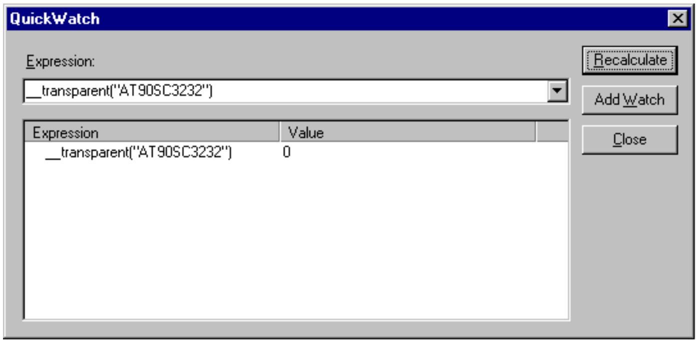
The following short description illustrates how to execute a transparent command:

1 Go to the **QuickWatch** dialog box by selecting **Quich Watch...** from the **Control** menu.



2 Send your transparent command to the ROM-monitor by using the predefined macro `__transparent`(*`commandstring`*). When pressing **Recalculate**, the macro will send the string argument *`commandstring`* as a transparent command to the ROM-monitor.

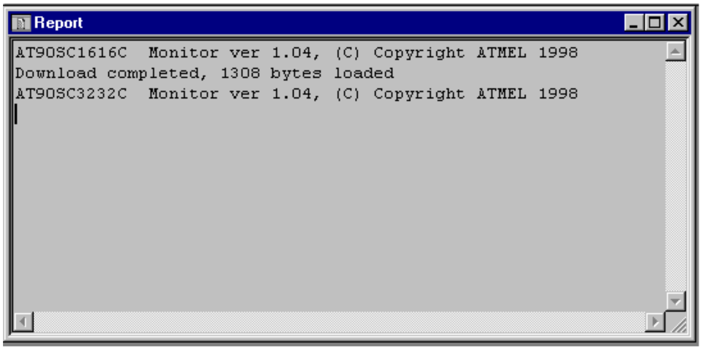
The following is the transparent command to change the mode of the board:



Expression: __transparent("AT90SC3232")

Expression	Value
__transparent("AT90SC3232")	0

3 The response will be displayed in the Report window:



```
AT90SC1616C  Monitor ver 1.04, (C) Copyright ATMEL 1998
Download completed, 1308 bytes loaded
AT90SC3232C  Monitor ver 1.04, (C) Copyright ATMEL 1998
```

DIAGNOSTIC MESSAGES

WARNING MESSAGES

The following table lists the warning messages of the IAR C-SPY ROM-monitor.

The monitor program can also generate warning and error messages and send them to the C-SPY Report window. Refer to your ROM-monitor documentation for information about messages from the monitor program.

40 `Fast download failed -- trying normal download`

The ROM-monitor is not quick enough to handle a constant stream of data and C-SPY switched to normal (slower) writes. You can disable fast download using the `-rz` switch on the command line. Hardware handshake or lowering the speed can also help. You should do something to get rid of this warning as downloading is slower than it should be.

40 `Breakpoint set failed at address` *`XXXX`*

If the given address is correct, check your hardware. Your application must reside in RAM. Otherwise C-SPY will not be able to step at assembly level. C-SPY also inserts breakpoints when you run with calls on.

ERROR MESSAGES

The following table lists the error messages of the IAR C-SPY ROM-monitor:

130 `PC at illegal instruction`

Tried to step an illegal instruction.

FATAL ERROR MESSAGES

The following table lists the fatal error messages of the IAR C-SPY ROM-monitor:

60 `Protocol version mismatch`

You are using a new C-SPY with an old ROM-monitor or vice versa.

60 `Illegal target ROM-monitor`

Use the correct C-SPY with the ROM-monitor.

60 `Unable to get extended error`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `Unable to read ROM-monitor memory`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `Unable to get protocol version`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `Unable to get Monitor status`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `Unable to get Sign on message`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `C-SPY address size not supported by Monitor`

You are running C-SPY in banked mode but the ROM-monitor does not support this.

60 `Failed to adjust Monitor address size`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.

60 `Unable to get address mask`

A protocol error occurred. Reset the ROM-monitor and restart C-SPY.