MMCEVB2103 Evaluation Board (EVB2103) User's Manual

CAUTION: ESD Protection

M•CORE development systems include open-construction printed circuit boards that contain static-sensitive components. These boards are subject to damage from electrostatic discharge (ESD). To prevent such damage, you must use static-safe work surfaces and grounding straps, as defined in ANSI/EOS/ESD S6.1 and ANSI/EOS/ESD S4.1. All handling of these boards must be in accordance with ANSI/EAI 625.

Contents

Section 1 Introduction

Section 2 Configuration

Section 3 Operation

Section 4 Connector Information

Figures

Tables

Section 1 Introduction

This user's manual explains connection, configuration, and operation information for the MMCEVB2103 Evaluation Board (EVB2103), a development tool of Motorola's M•CORE™ family. The EVB2103 lets you develop code to be embedded in an MMC2103 microcontroller unit.

A standalone tool, the EVB2103 uses an RS232 connection to your computer. This connection lets you use Motorola's M•CORE System Development Software (SysDS) or the GNU source-level debugger. The SysDS consists of a loader, the Picobug monitor, and a built-in selftest. The EVB2103 also has a OnCE™ connector, enabling you to use a debugging application that requires one.

Optionally, you may use the EVB2103 with a different emulator product, such as the Motorola Embedded Background Debug Interface (EBDI).

Motorola's SysDS loader lets you download your code into the EVB2103's SRAM (for execution) or FLASH memory (for execution or for storage in non-volatile memory).

1.1 EVB2103 Features

The EVB2103 features:

- MMC2103 resident MCU (a development version, in a 160-pin PQFP package).
- 2 megabytes FLASH memory.
- 0.5 megabytes FSRAM (fast static RAM), configurable for 8-, 16-, or 32-bit operations.
- Address decode logic.
- Power supply that converts line power to 12-volt input power.
- Two RS232 serial communication ports.
- A CAN communications port.
- A J1850 communications port.
- OnCE connector.
- Four user-accessible LEDs.
- Three DIP switches: two for system configuration, one for firmware selection.
- Two user prototyping (breadboard) areas.
- Motorola's System Development Software (SysDS).
- GNU source-level debugger (from the Free Software Foundation).
- Eyelets for connections to MMC2103 peripherals.

1.2 System and User Requirements

You need an IBM PC or compatible computer, running the Windows 95 or WindowsNT (version 4.0) operating system. The computer requires a Pentium (or equivalent) microprocessor, 16 megabytes of RAM, 50 megabytes of free hard-disk space, an SVGA color monitor, and an RS232 serial-communications port. To use the Picobug debug monitor, you also need Hyperterminal or a comparable terminal-emulation program.

To get the most from your EVB2103, you should be an experienced C or M•CORE assembly programmer.

Your EVB2103 requires 12-volt input power, at 0.5 amperes. The power supply that comes with your EVB2103 provides this voltage from line power.

1.3 EVB2103 Layout

Figure 1-1 shows the layout of the EVB2103. Jumper header W1 specifies the chip select for memory-mapped I/O (MMIO) device. Jumper header W2 specifies the FLASH chip select. Jumper header W3 specifies the FSRAM chip select.

Jumper header W6 selects either of two automotive-electronic communication protocols: CAN or J1850. Jumper header W7 activates or deactivates the RS232 communications driver of the resident MCU.

Connector J6 is the OnCE connector. Connector J52 is a connector for outputs in the J1850 automotive-electronics protocol. Connector J56 is the connector for 12-volt input power. Connector J58 is the connector for communications in the CAN automotive-electronics protocol. Connectors J60 and J61 are the RS232 serial connectors: ports 2 and 1, respectively.

Switch S1 is the reset switch. Switch S2 specifies the firmware module to be run upon reset. Switches S3 and S4 configure several aspects of resident-MCU operation.

Yellow LED DS1 confirms VPP programming voltage. LEDs DS2 through DS5 are general-purpose status indicators. Green LEDs DS6 and DS7 confirm 3.3-volt power and 5-volt power, respectively.

The EVB2103 has two prototyping areas. The smaller, squarish area is at the left edge of the board. The 3.3-volt and ground connections are the partial columns of holes on the left and right of this area. The 5-volt connections are the partial row of holes on the bottom of this area.

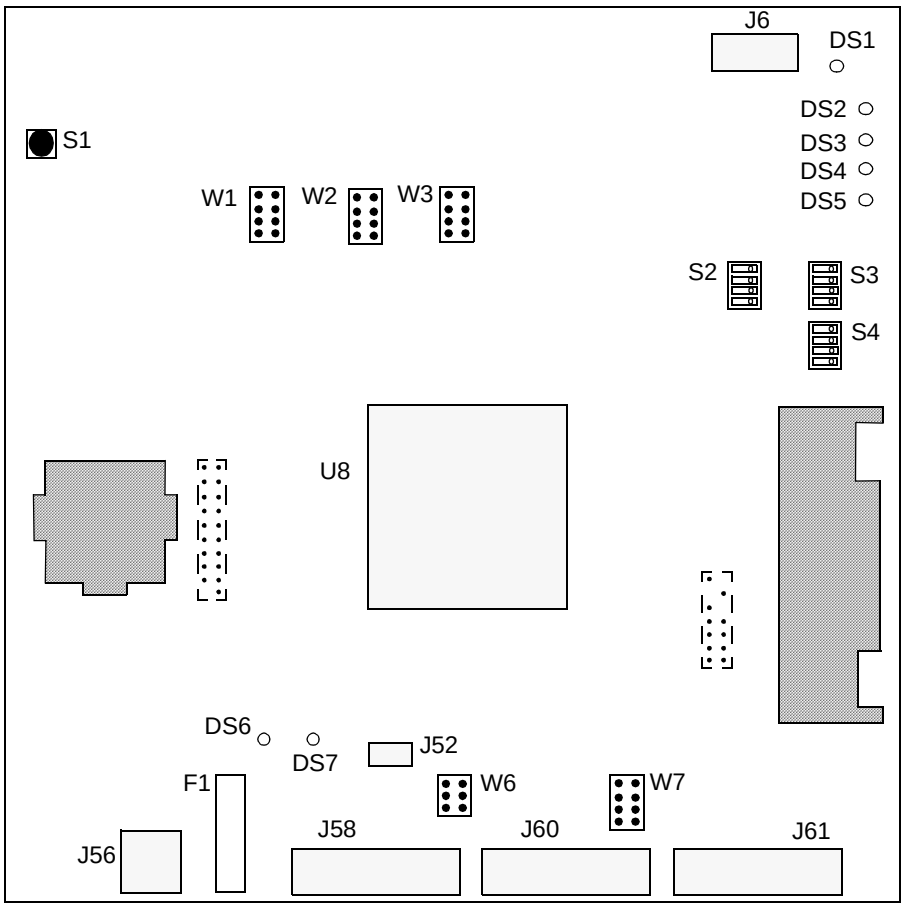


Figure 1-1 MMCEVB2103 Evaluation Board

The larger, rectangular prototyping area is at the right edge of the EVB2103. The 3.3-volt and 5-volt connections are the top and bottom rows of holes; the ground connections are the partial column of holes at the right.

Note the two groups of eyelets between each prototyping area and the adjacent MAPI connector. (In **Figure 1-1**, a dashed rectangle encloses each eyelet group.) The eyelets are convenient points for connection to peripheral signals likely to be used in prototyping.

Location F1 is for the EVB2103 fuse.

The resident MCU, at location U8, is an MMC2103 device, in a 160-pin PQFP package.

Table 1-1 lists EVB2103 specifications.

Table 1-1 MMCEVB2103 Evaluation Board Specifications

Characteristic	Specifications
MCU extension I/O port	HCMOS compatible
Operating temperature	0° to 40° C
Storage temperature	-40° to +85° C
Relative humidity	0 to 90% (non-condensing)
Reference clock oscillator	8 megahertz.
External clock	8 to 32 megahertz, set by system software, via the PLL function of the resident MCU.
Power requirements	12 volts dc, at 0.5 amperes, provided from a separate power source
Dimensions	6.9 x 8.2 inches (175 x 208 mm)

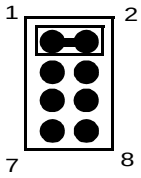
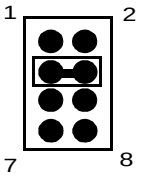
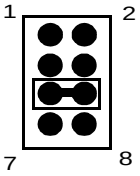
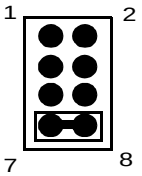
Section 2 Configuration

This chapter explains how to configure your EVB2103, and how to hook it up to your computer system.

2.1 Configuring Board Components

Table 2-1 is a summary of configuration settings; subsections 2.1.1 through 2.1.7 give additional information.

Table 2-1 Component Configuration Settings

Component	Position	Effect
MMIO Chip Select Header, W1, FLASH Chip Select Header, W2, or FSRAM Chip Select Header, W3		Configures chip select 1. **Factory setting for header W2.**
		Configures chip select 2. **Factory setting for header W3.**
		Configures chip select 3. **Factory setting for header W1.**
		Configures chip select 4.

Configuration

Table 2-1 Component Configuration Settings

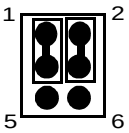
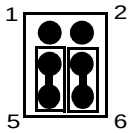
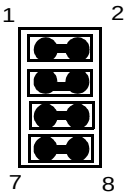
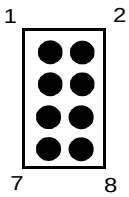
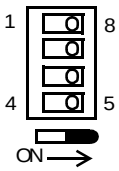
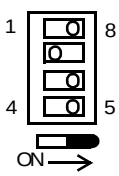
Component	Position	Effect
Protocol Header, W6		Configures the J1850 automotive-electronics protocol. **Factory setting.**
		Configures the CAN automotive-electronics protocol.
RS232 Header, W7		Activates RS232 I/O ports 2 and 1 (connectors J60 and J61). **Factory setting.**
		Deactivates RS232 ports 2 and 1. (Appropriate for MAPI-ring communication via a platform board.).
Reset Switch, S1		Push to reset all board components.
Firmware Select Switch, S2		Runs built-in selftest, which can detect board-level system problems.
		Runs Picobug monitor for software development. **Factory setting.**

Table 2-1 Component Configuration Settings

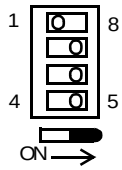
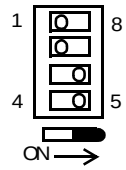
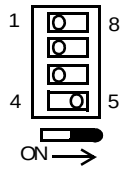
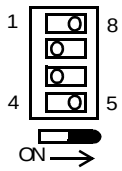
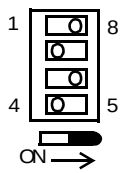
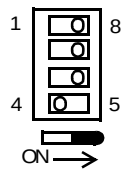
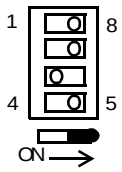
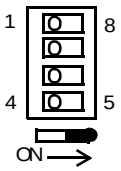
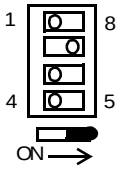
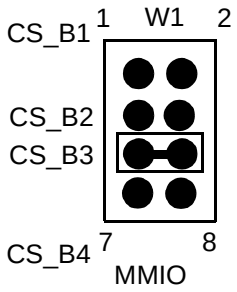
Component	Position	Effect
Firmware Select Switch, S2 (continued)		Runs programmer firmware module.
		Runs MetroTRK.
		Runs user code starting at address 0x8002_0000.
Configuration Switch, S3		Specifies 25pf output drive strength, external boot with 32-bit port size, and RS232 communication. (One of many plausible configurations.) **Factory setting.**
		Specifies 25pf output drive strength, internal boot with 32-bit port size (provided S4 configures master mode), and RS232 standby (tri-state) mode. (If S4 configures emulation mode, S3 configures external boot with 32-bit port size.) (Another of many plausible configurations.)
		Specifies 25pf output drive strength, internal boot with 32-bit port size (provided S4 configures master mode), and RS232 standby (tri-state) mode. (If S4 configures emulation mode, S3 configures external boot with 16-bit port size.) (Another of many plausible configurations.)
		Specifies 25pf output drive strength, external boot with 16-bit port size, and RS232 communication. (Another of many plausible configurations.)

Table 2-1 Component Configuration Settings

Component	Position	Effect
Configuration Switch, S4		Enables internal FLASH and specifies master mode. (One of severable possible configurations.)
		Enables internal FLASH and specifies emulation mode. (Another of 2 possible configurations.) **Factory setting.**

2.1.1 Setting the MMIO Chip Select Header (W1)

Jumper header W1 specifies the chip select for the memory-mapped I/O (MMIO) subsystem. The diagram below shows the factory configuration: the jumper between W1 pins 5 and 6 specifies chip select 3.

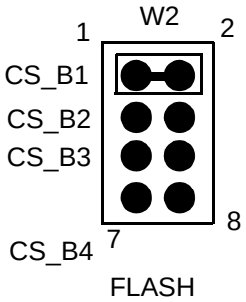


Alternatively, you can select chip select 1, 2, or 4 for MMIO by repositioning the W1 jumper, respectively, to pins 1 and 2, 3 and 4, or 7 and 8.

NOTES: *Each chip-select header, W1 through W3, must specify a different chip select. Do not install multiple jumpers in any of these headers.*

2.1.2 Setting the FLASH Chip Select Header (W2)

Jumper header W2 specifies the FLASH chip select. The diagram below shows the factory configuration: The jumper between W2 pins 1 and 2 specifies chip select 1 for FLASH memory.

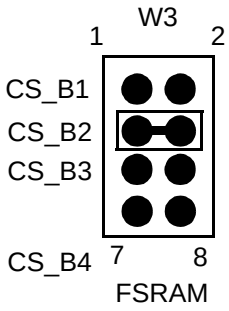


Alternatively, you can select chip select 2, 3, or 4 for FLASH memory by repositioning the W2 jumper, respectively, to pins 3 and 4, 5 and 6, or 7 and 8.

NOTES: *Each chip-select header, W1 through W3, must specify a different chip select. Do not install multiple jumpers in any of these headers.*

2.1.3 Setting the FSRAM Chip Select Header (W3)

Jumper header W3 specifies the FSRAM chip select. The diagram below shows the factory configuration: the jumper between W3 pins 3 and 4 specifies chip select 2 for FSRAM.

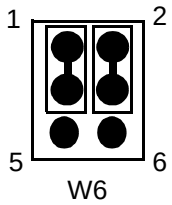


Alternatively, you can select chip select 1, 3, or 4 for FSRAM by repositioning the W3 jumper, respectively, to pins 1 and 2, 5 and 6, or 7 and 8.

NOTES: *Each chip-select header, W1 through W3, must specify a different chip select. Do not install multiple jumpers in any of these headers.*

2.1.4 Setting the Automotive Interface Header (W6)

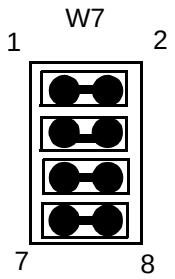
Jumper header W6 specifies either of two automotive-electronics interfaces. The diagram below shows the factory configuration: the jumpers between pins 1 and 3, and between pins 2 and 4, specify the J1850 interface, activating J1850 communications connector J52.



Alternatively, you may specify the CAN interface, activating CAN communications connector J58, by repositioning the W6 jumpers between pins 3 and 5, and between pins 4 and 6.

2.1.5 Setting the RS232 Header (W7)

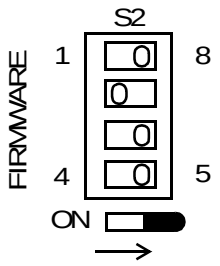
Jumper header W7 activates or deactivates the RS232 I/O ports (connectors J60 and J61). The diagram below shows the factory configuration: the four jumpers activate the two ports.



NOTE: *Leave the W7 jumpers installed for all normal use of your EVB2103. (Deactivation of RS232 ports pertains to possible uses of an MMCCMB2103 controller and memory board, which uses the same printed circuit board as your EVB2103.)*

2.1.6 Setting the Firmware Select Switch (S2)

Switch S2 specifies the firmware to be run on the processor upon a reset. The diagram below shows the factory configuration: subswitches 1and 3 ON, and subswitch 2 OFF. This specifies the Picobug monitor. (The position of subswitch 4 does not matter.)



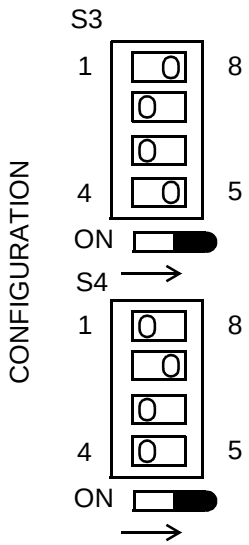
To specify a different firmware module, reset the S2 subswitches per **Table 2-2**.

Table 2-2 S2 Subswitch Settings

Firmware Module	Subswitch 1 (Pins 1, 8)	Subswitch 2 (Pins 2, 7)	Subswitch 3 (Pins 3, 6)
Built-in selftest	ON	ON	ON
Picobug monitor	ON	OFF	ON
Programmer	OFF	ON	ON
MetroTRK	OFF	OFF	ON
User code	OFF	OFF	OFF

2.1.7 Setting the Configuration Switches (S3, S4)

Switches S3 and S4 configure several operational aspects of the resident MCU. The diagram below shows the factory configuration. S3 subswitches 1 and 4 ON, and subswitches 2 and 3 OFF, configure 25pf output drive strength, external boot with a 32-bit port size, and RS232 communication enabled. S4 subswitch 1 OFF and subswitch 2 ON enable internal FLASH and configure emulation mode.



For a different configuration, reset the S3 and S4 subswitches per **Table 2-3**.

Table 2-3 S3, S4 Subswitch Settings

Subswitch	ON Configures	OFF Configures
S3-1	25pf output drive strength (recommended).	Full output drive strength.
S3-2	External boot, 16-bit port size.	External boot, 32-bit port size.
S3-3	Internal boot, 32-bit port size: overrides S3-2.	External boot per S3-2.
S3-4	RS232 communication.	RS232 standby (tri-state) mode.
S4-1	Disables internal FLASH.	Enables internal FLASH.
S4-2	Emulation mode.	Master mode.

2.2 Making Computer System Connections

When you have configured your EVB2103, you are ready to connect it to your computer system:

1. Make sure that power is disconnected.
2. If you will use RS232 communication directly with your host computer, connect an RS232 cable between EVB2103 connector J60 or J61 and the appropriate serial port of your computer.
3. If you will use a OnCE-compatible emulator with your EVB2103, connect an appropriate 14-lead ribbon cable between EVB2103 connector J6 and your emulator. Then use an appropriate cable to connect your emulator to your host computer.
4. Optional: You may use either of two automotive-electronics interfaces for communication with an emulator. (Jumper header W6 selects the interface)
   - For CAN-protocol communication, connect an appropriate cable between EVB2103 connector J58 and the corresponding connector of your emulator.
   - For J1850-protocol communication, connect an appropriate cable between EVB2103 connector J52 and the corresponding connector of your emulator.
5. Connect your 12-volt power supply to line power and to EVB2103 connector J56. Turn on power: green LEDs DS6 and DS7 light to confirm that the EVB2103 is powered and converting input voltage.

   Should the LEDs *not* light, you may need to replace the fuse at location F1, next to power connector J56. (Use a BUS GMA-1.5A fuse, or compatible.)
6. This completes system connections: you are ready to perform a selftest, per the instructions of subsection 2.3, below. You are ready to begin debugging or other development activities, per the instructions of Chapter 3.

2.3 Performing the EVB2103 Selftest

Once you have configured your EVB2103, you can perform a selftest of its components.

1. Make sure that EVB2103 power is turned off or disconnected. Green power LEDs DS6 and DS7 should be out.
2. Set switch S2 for the built-in selftest: all subswitches ON.
3. Set switch S3 subswitches 1 and 4 ON; set S3 subswitches 2 and 3 OFF.
4. Turn on power. Green LEDs DS6 and DS7 come on to confirm power, and the EVB2103 begins its selftest.

5. LEDs DS2 through DS5 light and go out during the test, according to the sequence of **Table 2-4**.

Table 2-4 EVB2103 Selftest LED Sequence

DS2	DS3	DS4	DS5	Test Action
ON	OFF	ON	OFF	8-bit write to memory.
OFF	ON	OFF	ON	8-bit read from memory.[1]
ON	OFF	ON	OFF	16-bit write to memory.
OFF	ON	OFF	ON	16-bit read from memory.[1]
ON	OFF	ON	OFF	32-bit write to memory.
OFF	ON	OFF	ON	32-bit read from memory.[1]

NOTES:
1. Should all four LEDs stay lit at this point, the EVB2103 has failed the SRAM test, aborting the rest of the selftest. Contact Motorola customer support for assistance.

6. Then individual LEDs light several times in the sequence, DS5, DS4, DS3, and DS2.
7. When all four LEDs go out, the EVB2103 has passed the selftest. (If any LEDs stay lit, the EVB2103 has failed the selftest: contact Motorola customer support for assistance.)
8. Disconnect power.
9. Configure switch S2 for your next development activity before restoring power to the EVB2103.

2.4 Memory Map

Table 2-5 is the default memory map for your EVB2103. **Table 2-6** lists FLASH sector boundaries: the shaded sectors contain system software.

Table 2-5 EVB2103 Default Memory Map

Address Range	Sub Range	Memory Resource	Related Signal
0x8000_0000 0x801F_FFFF		EVB FLASH (2 megabytes)	CS1
	0x8000_0000 0x8001_FFFF	System Software (128 kilobytes Sectors 0 — 3)	
	0x8002_0000 0x801F_FFFF	User Code (1920 kilobytes Sectors 4 — 18	
0x8020_0000 0x806F_FFFF		User address space (5 megabytes)	
0x8070_0000 0x807F_FFFF		MMIO (1 megabyte)	CS3
	0x8070_0000 0x8070_7002	unused	
	0x8070_7003	MMIO read byte (reads S2 subswitch settings) or MMIO write byte (controls LEDs and internal FLASH programming voltage)	
	0x8070_7004 0x807F_FFFF	unused	
0x8080_0000 0x8087_FFFF		EVB SRAM (0.5 megabyte)	CS2
	0x8080_0000 0x8080_FFFF	System Software (64 kilobytes)	
	0x8081_0000 0x8087_FFFF	User code (448 kilobytes)	
0x8088_0000 0x80BF_FFFF		unused	
0x80C0_0000 0x80FF_FFFF		User address space (4 megabytes)	CS4

Table 2-6 FLASH Sector Boundaries

Sector (Block)	Range
0	0x8000_0000 — 0x8000_7FFF
1	0x8000_8000 — 0x8000_BFFF
2	0x8000_C000 — 0x8000_FFFF
3	0x8001_0000 — 0x8001_FFFF
4	0x8002_0000 — 0x8003_FFFF
5	0x8004_0000 — 0x8005_FFFF
6	0x8006_0000 — 0x8007_FFFF
7	0x8008_0000 — 0x8009_FFFF
8	0x800A_0000 — 0x800B_FFFF
9	0x800C_0000 — 0x800D_FFFF
10	0x800E_0000 — 0x800F_FFFF
11	0x8010_0000 — 0x8011_FFFF
12	0x8012_0000 — 0x8013_FFFF
13	0x8014_0000 — 0x8015_FFFF
14	0x8016_0000 — 0x8017_FFFF
15	0x8018_0000 — 0x8019_FFFF
16	0x801A_0000 — 0x801B_FFFF
17	0x801C_0000 — 0x801D_FFFF
18	0x801E_0000 — 0x801F_FFFF

2.5 Memory Mapped I/O Operation

Address 0x8070_7003 is the MMIO write byte or read byte:

- Your code can write values to this address to activate or deactivate programming voltage, and to turn LEDs DS2 through DS5 on or off. **Figure 2-1** depicts this write-byte functionality.

0x8070_7003							
D7	D6	D5	D4	D3	D2	D1	D0
not used	not used	VPP	not used	LED DS2	LED DS3	LED DS4	LED DS5

Figure 2-1 MMIO Write Byte

  – A 0 written to bit D0, D1, D2, or D3 turns ON the corresponding LED. A 0 written to bit D5 enables the programming voltage.

  – A 1 written to written to bit D0, D1, D2, or D3 turns OFF the corresponding LED. A 1 written to bit D5 disables the programming voltage.

- Your code can read values from this address, to check the settings of S2 subswitches. **Figure 2-2** depicts this read-byte functionality.

0x8070_7003				
D7 — D4	D3	D2	D1	D0
not used	S2 — 4	S2 — 3	S2 — 2	S2 — 1

Figure 2-2 MMIO Read Byte

  – If your code reads a 0 from bit D0, D1, D2, or D3, the corresponding S2 subswitch is ON.
  – If your code reads a 1 from bit D0, D1, D2, or D3, the corresponding S2 subswitch is OFF.

2.6 Using the Prototyping and Eyelet Areas

The EVB2103 prototyping areas let you add your own components to the board. Merely insert the component's feet through holes in the board; solder the feet in place to hold the component in position. Run appropriate leads from the new component to board power and ground locations.

Note that separate columns or rows of each prototyping area are +3.3-volt sources, +5-volt sources, and ground connections.

The two eyelet areas offer convenient connection to several key signals. **Figure 2-3** represents the larger eyelet area, identifying the signal for each eyelet.

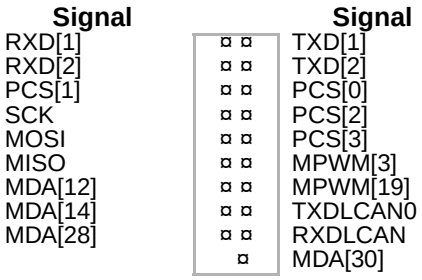


Figure 2-3 Large Eyelet Area Signals

Figure 2-4 represents the smaller eyelet area, identifying the signal for each eyelet.

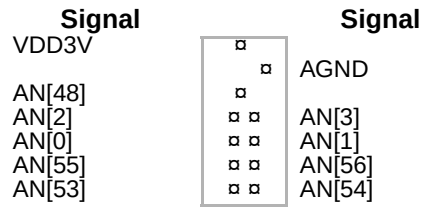


Figure 2-4 Small Eyelet Area Signals

Section 3 Operation

This chapter explains how to begin using debugging tools available for your MMCEVB2103 Evaluation Board, as well as how to use Motorola's SysDS Loader.

3.1 Debugging Embedded Code

With your EVB2103, you may use the Picobug monitor as standalone software. Optionally, you may use the GNU source-level debugger with the Picobug monitor. Other firms may produce still additional software to run, test, and modify the code you develop for embedding in an MMC2103 MCU.

3.1.1 Using the Picobug Monitor

The Picobug monitor comes burned into the external FLASH memory devices of your EVB2103**.** Before you start the Picobug monitor, make sure that you have an RS232 connection between EVB2103 connector J60 or J61 and a serial port of your computer.

To start the monitor, for use as a standalone debugger:

1. Make sure that power is *not* applied to your EVB2103.
2. Activate Hyperterminal or a comparable terminal-emulation program. (If you use a different terminal-emulation program, you must make corresponding changes in the commands and menu selections of these instructions, and in the instructions of paragraph 3.1.2.)
3. From the File menu, select Properties. This opens a properties dialog box.
4. Click on the Configure button of the dialog box. This opens a configuration dialog box.
5. Use the configuration dialog box to set these communications properties: 19,200 baud, 8 data bits, no parity, 1 stop bit, and no flow control. Also specify the correct communications port (for example, COM1). Click the OK button of the dialog box.
6. Set switch S2 for the Picobug monitor: subswitches 1, 3, and 4 ON; subswitch 2 OFF.
7. Apply power to the EVB2103 and press the enter key. The Picobug monitor starts automatically, displaying the command prompt: `picobug>`.

To use the Picobug monitor, merely enter commands at the prompt. **Table 3-1** explains these commands. To see a list of these commands on your computer screen, enter a question mark or the extra command `he` at the command prompt.

Table 3-1 Picobug Commands

Command	Explanation
br [*address*]	Breakpoint: • With optional *address* value, sets a new breakpoint at that address. • Without any *address* value, lists all current breakpoints.
g [*address*]	Go: • With optional *address* value, starts code execution from that address. • Without any *address* value, starts code execution from the current program-counter value. In either case, execution stops when it arrives at a breakpoint.
gr	Go to Return: Executes code from the current program-counter value to the return address of the calling routine. (Should execution arrive at a breakpoint before encountering the return address, execution stops at the breakpoint.)
gt *address*	Go to Address: Executes code from the current program-counter value to the specified *address* value. (Should execution arrive at a breakpoint before encountering the specified address, execution stops at the breakpoint.)
he	Help Displays available commands, identical to the ? command.
lo [*address*]	Download: • With optional *address* value, downloads a binary image to that address in SRAM. • Without any *address* value, downloads to SRAM an S-record text file.
md [*address1* [*address2*]] [;*size*]	Memory Display: • With optional *address1* and *address2* values, displays memory contents between the addresses. • With optional *address1* value, displays contents of 16 memory bytes. • With no address value, defaults to the last address viewed. • The optional *size* value specifies the format: **b** (bytes, the default), **h** (half words), **w** (words), or **i** (instructions).
mds [*address*]	Memory Display 256: • With optional *address* value, displays contents of 256 memory bytes, starting at that address. • With no address value, displays contents of 256 memory bytes, starting from the last address viewed.
mm [*address* [*value*]] [;*size*]	Modify Memory: • With optional *address* and *value* parameter values, assigns that value to the *address* location. • With optional *address* value but no *value* parameter value, prompts for a value for the *address* location, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • With no optional *address* value, prompts for a value for the last address viewed, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • The optional *size* value, specifies the format: **b** (bytes, the default), **h** (half words), **w** (words), or **i** (instructions).
nobr [*address*]	No Breakpoint: • With optional *address* value, removes the breakpoint from that address. • Without any *address* value, removes all the breakpoints.

Table 3-1 Picobug Commands (Continued)

Command	Explanation
reset	Reset: Resets the CPU and peripherals.
rd [*name*]	Register Display: • With optional *name* value, displays the value of that CPU register. • Without any *name* value, displays the values of all CPU registers.
rm *name value*	Register Modify: Assigns the *value* parameter value to the *name* CPU register.
t	Trace (Step): Single steps one instruction; identical to the s command.
s	Step (Trace): Single steps one instruction; identical to the t command.
?	Help Displays available commands, identical to the he command.

3.1.2 Picobug Sample Session

1. This sample session begins with the Picobug prompt:

```
picobug>
```

2. To see the contents of all registers, enter the Register Display (rd) command without any name value:

```
picobug>     rd
```

   The system responds with a display such as this:

```
     pc 32313033         epc 8000401c         fpc 00000000
    psr 80000000        epsr 80000000        fpsr 00000000
ss0-ss4 bad0beef   80010000  00000000  00000000  00000000         vbr 80805c00
  r0-r7 bad0beef   00000000  80010000  00efff34  000000ff   d4037308  74099403  73097409
 r8-r15 80800024   80010040  0000000f  80707003  0000002d   00000012  0000002d  800000b4
```

3. To see the contents of a specific register, such as the epc register, enter the Register Display (rd) command *with* the name value:

```
picobug>  rd epc
```

   The system responds with a display such as this:

```
epc:      8000401c
```

4. To see the contents of a specific memory location, enter the Memory Display (md) command with the location address. An optional size value (in this case w, for word) may be part of the command:

```
picobug>  md    0x8080B090   ;w
```

   The system responds with a display such as this:

```
8080B090:   FFFFF92C
```

5. To see the contents of a memory range, enter the Memory Display (md) command with the beginning and ending addresses. An optional size value (in this case b, for byte) may be part of the command:

```
picobug>  md    0x8080b090   0x8080b0a0   ;b
```

   The system responds with a display such as this:

```
8080B090: FF  FF  F9  2C  FF  FF  F9  2B  FF  FF  F9  2A  FF  FF  F9  29 ...,...+...*...
02000010: FF
```

6. To download into SRAM a program executable, in S-record format, enter the Download (lo) command without any address value:

```
picobug>  lo
```

   The system waits for you to send the program executable file. To do so, open the Transfer menu and select Send Text File. This opens a file-select dialog box. Use this dialog box to specify the appropriate S-record file, then click on the Open button. As soon as the download is complete (this may take several minutes), the Picobug prompt reappears:

```
picobug>
```

7. To see the new contents of registers, enter the Register Display (rd) command again, without any name value:

```
picobug>  rd
```

   The system responds with an updated display, which shows that the pc register value reflects the start of the program just downloaded:

```
    pc 8080b0b0         epc 00000000         fpc 00000000
   psr 80000000        epsr 00000000        fpsr 00000000
ss0-ss4 bad0beef   80010000  00000000  00000000  00000000         vbr 80805c00
  r0-r7 bad0beef   00000000  80010000  00efff34  000000ff   d4037308  74099403  73097409
 r8-r15 80800024   80010040  0000000f  00000000  00000000   00000000  00000000  800000b4
```

8. To set a breakpoint at address 0x8080B0AA, enter this address as part of the Breakpoint (br) command:

```
picobug>  br    0x8080b0aa
```

   The Picobug prompt reappears, confirming that the system set the breakpoint:

```
picobug>
```

9. To see the list of breakpoints, enter the Breakpoint (br) command *without* any address value:

```
picobug>  br
```

   The system responds with the addresses of breakpoints, in this case only the breakpoint set in step 8:

```
8080B0AA
```

10. To start program execution, enter the Go (g) command:

```
picobug>  g
```

    In this instance, the breakpoint set during step 8 stops code execution. The system responds with this new display of register values:

```
   At breakpoint!!
      pc 8080b0aa         epc 8080b0aa         fpc 00000000
     psr 80000100        epsr 80000100        fpsr 00000000
 ss0-ss4 bad0beef   80010000  00000000  00000000  00000000         vbr 80805c00
   r0-r7 8080e460   00000000  80010000  00efff34  00000000   8080d458  8080d457  8080d460
  r8-r15 8080e460   80010040  0000000f  00000000  00000000   00000000  00000000  8080b0aa
8080B0AA:    00CF       rts
```

11. To remove all breakpoints, enter the No Breakpoint (nobr) command, without any address value:

```
picobug>  nobr
```

    The Picobug prompt reappears, confirming that the system has removed the breakpoints:

```
picobug>
```

12. To see the list of breakpoints again, once more enter the Breakpoint (br) command without any address value:

```
picobug>  br
```

    As there are no longer any breakpoints, the system responds with the Picobug prompt:

```
picobug>
```

13. To continue with this example session, enter another appropriate command. For example, to resume program execution, enter the Go (g) command.

14. To end your Picobug session, remove power from the EVB and close the terminal-emulation program.

3.1.3 Using the GNU Source-Level Debugger

The GNU source-level debugger is on the CD-ROM that comes with your EVB2103. This GNU software works with the Picobug monitor to provide source-level debugging for your code.

The EVB2103 software release guide gives the instructions for loading the GNU software, and for making any connections different from standalone Picobug connections.

3.2 Using the SysDS Loader

The Motorola SysDS Loader lets you program code into FLASH memory, upload FLASH contents to a PC file, verify that FLASH contents match those of a download file, display memory contents, erase FLASH memory, erase a sector of FLASH memory, or blank check a sector of FLASH memory.

NOTE: *For the first action of an SysDS Loader session (downloading, verifying, displaying, erasing, or blank checking), the software may download algorithm file* `programmer2103.rec` *before carrying out the action.*

(If the software cannot find the algorithm file, an appropriate error message identifies the file. Click on the message's OK button to bring up a file-select dialog box, then use this dialog box to specify the location of the algorithm file. If necessary, recopy the file from the transmittal CD-ROM. Click on the OK button to resume your SysDS Loader action.)

Follow these steps to use the SysDS Loader:

1. If you have not already installed the SysDS Loader onto your computer hard disk, do so. The EVB2103 product release guide includes installation instructions.
2. If Hyperterminal is running in support of the Picobug monitor, stop Hyperterminal. (The SysDS Loader needs the same computer serial port that Hyperterminal uses.)
3. Set switch S2 for the Programmer: subswitch 1 OFF, all other subswitches ON.
4. Press switch S1 to reset the EVB2103.

5. Start the SysDS Loader. The main screen (**Figure 3-1**) appears.

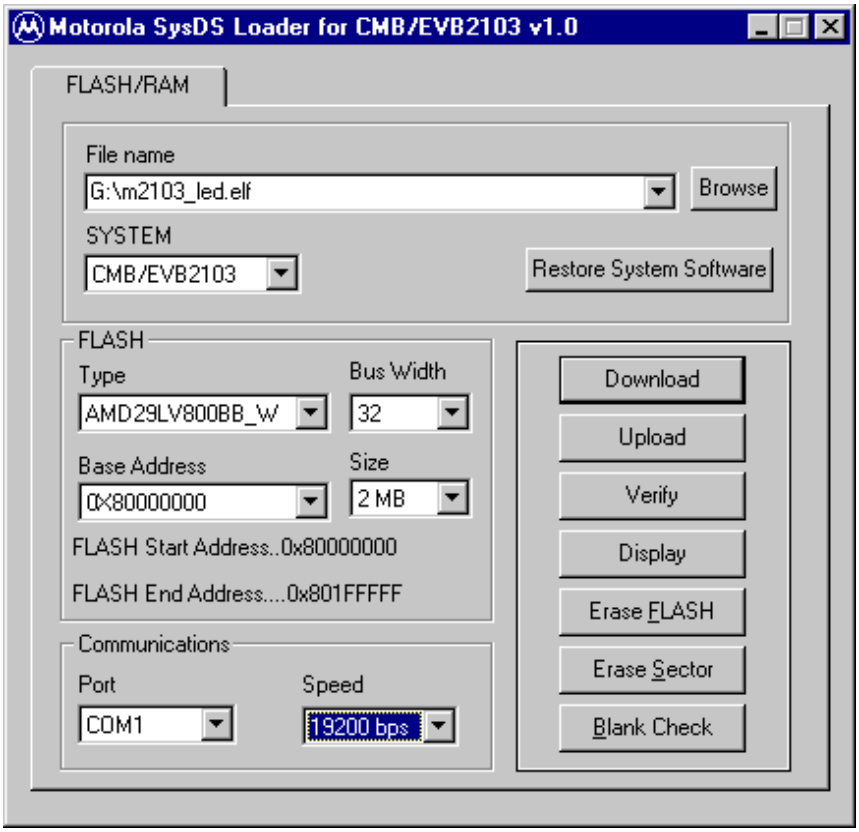


Figure 3-1 SysDS Loader Main Screen

6. Go to the File name field.
   - If you know the full pathname of the file to be programmed, enter the pathname in this field.
   - If you do not know the full pathname of the file to be programmed, click on the Browse button. This brings up a standard file-select dialog box: select the file and click on the OK button. This returns you to the main screen, entering the pathname in the File name field.
   - (If your only action for this Loader session will be uploading FLASH contents, you may leave the File name field blank.)

NOTE: *The Restore System Software button of the main screen updates FLASH sectors 0 through 3 with the software in your hard-drive directory \Motorola\Loader\MMC2103. Should a factory update replace the software in this directory, set switch S3, subswitch 3, for external boot, then click on the Restore System Software button.*

7. Use the FLASH area to configure the FLASH type, bus width, and size.

   The value in the Base Address field is automatic. (However, you may select the optional value <CUSTOM>, which brings up the Custom Address dialog box. Enter an appropriate address, then click on the dialog box OK button to return to the main screen.)

8. In the Communications area, use the Port field to specify the PC serial port, and use the Speed field to specify the communications rate. (The default rate is 19200 baud.)

9. To program FLASH memory, click on the Download button. As the software downloads the file you specified, a progress message appears in a Status dialog box. A Download successful message appears at the end of downloading: you are ready to use the code in FLASH memory.

   The error message Unable to Validate Flash configuration indicates some problem with the programming. A likely such problem is that the chip select base address does not correspond to the configured chip select. Correct the problem, then click again on the Download button.

10. To upload FLASH memory contents to a file in your PC, click on the Upload button. This brings up the Upload To File dialog box, **Figure 3-2**:

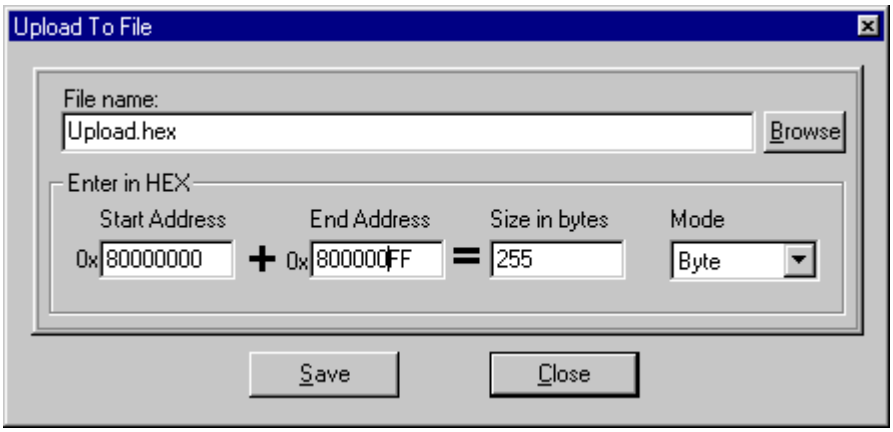


Figure 3-2 Upload To File Dialog Box

- Enter the name of the destination file. Optionally, click on the Browse button, to select a file via a standard file-select dialog box.
- The Start Address field indicates the start of EVB2103 FLASH memory or RAM. The default address value corresponds to the value of the SYSTEM field of the main screen, but you may enter a different address, if appropriate.
- Enter the appropriate value in the End Address field. (The system automatically determines the value for the Size in Bytes field.)
- The Size in Bytes field value corresponds to the value of the Size field of the main screen. (If appropriate, you may enter a different value.)

- The default Mode field value is Byte.
- When the Upload To File dialog box shows appropriate values, click on the Save button. A progress message appears during uploading.

11. To verify that the contents of Flash memory match the selected download file, click on the Verify button. A progress message appears as verification begins. A Verify successful message appears at the end of verification.
    - If verification fails, an error message specifies the location that did not have the expected contents.
    - To recover from a verification failure, try downloading Flash again, to replace the selected download file.
12. To view the contents of Flash memory, click on the Display button. This brings up the Display Flash/Ram display (**Figure 3-3**).

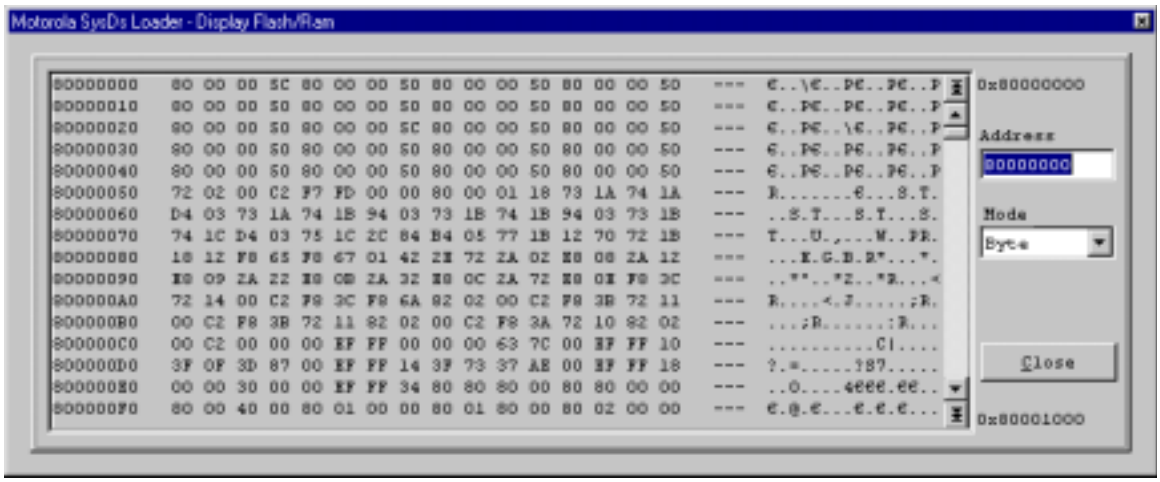


Figure 3-3 Display Flash/Ram Display

- The Address field shows the first address of the value display. One way to change the display is to enter a different address in this field.
- Another way to change the value display is to use the scroll bars.
- Use the Mode field to specify byte, half-word, or word values in the display.
- When you are done viewing the display, click on the Close button to return to the main screen.

13. To erase FLASH memory, click on the Erase FLASH button. The SysDS Loader erases all contents of the FLASH memory except for the sectors that contain system software.

14. To erase a sector of FLASH memory, click on the Erase Sector button. This brings up the Flash Sector Number dialog box. Enter the number of the sector to be erased (4 or greater), then click on the OK button.

    You may not erase FLASH sectors 0 through 3, which contain system software.

15. To verify that a FLASH sector is blank, click on the Blank Check button. This brings up a dialog box that asks for a sector number. Enter the number of the sector to be blank checked, then click on the OK button.

    A message tells you the results of the blank check. (If the sector is not blank, you can erase the sector or try a different sector.)

16. To end your SysDS Loader session, merely close the main screen.

Section 4 Connector Information

This chapter consists of pin assignments and signal descriptions for EVB2103 connectors.

4.1 OnCE Connector (J6)

Connector J6, a 2-by-7-pin connector, conveys data and control signals to and from the OnCE control block. **Figure 4-1** and **Table 4-1** give the pin assignments and signal descriptions for this connector.

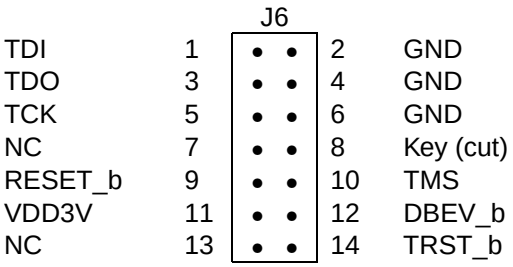


Figure 4-1 OnCE Connector J6 Pin Assignments

Table 4-1 OnCE Connector J6 Signal Descriptions

Pin	Mnemonic	Signal
1	TDI	TEST DATA INPUT – Data and command input line to the OnCE controller.
2, 4, 6	GND	GROUND
3	TDO	TEST DATA OUTPUT – Serial data output line from the OnCE controller.
5	TCK	TEST CLOCK – Serial clock input line to the OnCE control block.
7, 13	NC	No connection
8	---	Cut to be connector key.
9	RESET_b	RESET IN – Active-low input line to the OnCE controller, signalling a reset.
10	TMS	TEST MODE SELECT – Input signal that tells the OnCE control block to advance one mode state (of the cycle of mode states).
11	VDD3V	+3.3-volt power.
12	DBEV_b	DEBUG EVENT – Active-low debug-mode control line for the OnCE controller. An input signal from an external command controller makes the OnCE controller immediately enter debug mode. An output signal acknowledges debug-mode-entry to the external command controller.
14	TRST_b	TEST RESET – Active-low input line for an external reset signal to the OnCE controller.

4.2 Communication Connectors (J52, J58, J60, J61)

Connector J52, a simple, two-pin connector, is for communication in the J1850 automotive-electronics protocol:

- Pin 1: Ground
- Pin 2: the J1850 OUT signal.

Connector J58 is for communication in the CAN automotive-electronics protocols. The diagram below shows the pin numbering of this connector. **Table 4-2** lists the J58 CAN pin assignments.

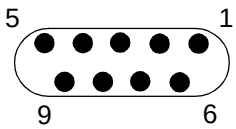


Table 4-2 CAN Connector J58 Pin Assignments

Pin	Signal
1, 4, 8, 9	No connection.
2	CANL — Low-level CAN value for I/O.
3, 6	GROUND
5	Reserved
7	CANH — High-level CAN value for I/O.

Connectors J60 and J61 are for pseudo-RS232 communication (the EVB2103 processor does not support full RS232 communication). These connectors have the same pin numbering as connector J58. **Table 4-3** lists the pin assignments and signal directions for these connectors.

Table 4-3 Communication Connector J60, J61 Pin Assignments

Pin	Signal	Signal Direction
1, 4, 6 — 9	No connection	—
2	TXD — Transmitted Data	Out
3	RXD — Received Data	In
5	GROUND	—
9	RI — Ring Indicator (Held inactive on the EVB2103.)	Out

4.3 MAPI Connector Sites (P1/J1, P2/J2, P3/J3, P4/J4)

The printed circuit board of your EVB2103 is identical to that of the MMCCMB2103 Controller and Memory Board (CMB2103). Although the EVB2103 does not have the CMB2103's modular, all-purpose interface (MAPI) connectors, the printed circuit board has sites for these connectors. MAPI connector sites P1 through P4 ring the U8 resident MCU. All the MAPI signals are available at these sites, as well as at sites J1 through J4, on the bottom of the board. **Figure 4-2** through **Figure 4-5**, and **Table 4-4** through **Table 4-7**, give the pin assignments and signal descriptions for these connector sites.

Connector Information

P1/J1

PTJ1[100]	100	99	VDD3V
PTJ1[98]	98	97	CS_b[4]
PTJ1[96]	96	95	PTJ1[95]
PTJ1[94]	94	93	PTJ1[93]
GND	92	91	PTJ1[91]
VDD5V	90	89	GND
PTJ1[88]	88	87	PTJ1[87]
PTJ1[86]	86	85	PTJ1[85]
PTJ1[84]	84	83	PTJ1[83]
PTJ1[82]	82	81	PTJ1[81]
PTJ1[80]	80	79	PTJ1[79]
PTJ1[78]	78	77	PTJ1[77]
PTJ1[76]	76	75	PTJ1[75]
GND	74	73	DVSP_b[0]
PTJ1[72]	72	71	GND
PTJ1[70]	70	69	PTJ1[69]
PTJ1[68]	68	67	PTJ1[67]
PTJ1[66]	66	65	PTJ1[65]
PTJ1[64]	64	63	PTJ1[63]
IRQ_b[0]	62	61	PTJ1[61]
VDD3V	60	59	VDD3V
GND	58	57	TXD[1]
PTJ1[56]	56	55	RXD[1]
PTJ1[54]	54	53	TXD[2]
GND	52	51	RXD[2]
PTJ1[50]	50	49	PCS[0]
SCK	48	47	PCS[1]
GND	46	45	PCS[2]
MOSI	44	43	PCS[3]
MISO	42	41	GND
GND	40	39	PTJ1[39]
PTJ1[38]	38	37	MPWM[3]
PTJ1[36]	36	35	MPWM[19]
IDVDD	34	33	MDA[12]
TXDLCAN0	32	31	MDA[14]
RXDLCAN	30	29	MDA[28]
GND	28	27	MDA[30]
GND1	26	25	GND1
PTJ1[24]	24	23	PTJ1[23]
PTJ1[22]	22	21	PTJ1[21]
PTJ1[20]	20	19	PTJ1[19]
PTJ1[18]	18	17	PTJ1[17]
PTJ1[16]	16	15	PTJ1[15]
PTJ1[14]	14	13	PTJ1[13]
GND1	12	11	PTJ1[11]
GND2	10	9	PTJ1[9]
PTJ1[8]	8	7	PTJ1[7]
PTJ1[6]	6	5	PTJ1[5]
PTJ1[4]	4	3	PTJ1[3]
GND2	2	1	PTJ1[1]

Figure 4-2 MAPI Connector Site P1/J1 Pin Assignments

Table 4-4 MAPI Connector Site P1/J1 Signal Descriptions

Pin	Mnemonic	Signal
100, 98, 96 — 93, 91, 88 — 75, 72, 70 — 63, 61, 56, 54, 50, 39, 38, 36, 24 — 13, 11, 9 — 3, 1	PTJ1[x]	Pass through.
99, 60, 59,	VDD3V	+3.3-volt power
97	CS_b[4]	CHIP SELECT (line 4) – Active-low output line that provides a chip select to external devices.
92, 89, 74, 71, 58, 52, 46, 41, 40, 28	GND	GROUND
90	VDD5V	+5-volt power.
73	DVSP_b[0]	DEVELOPMENT SPACE (line 0) — Active-low signal indicating that the current memory cycle is addressing on-board devices.
62	IRQ_b[0]	INTERRUPT REQUEST — Active-low output indicating that the processor core is servicing an internal interrupt.
58, 52, 40, 34	MID[0] — MID[3]	MID (identification code) lines 0—3 — Signals that identify the host processor board.
57, 53	TXD[1], TXD[2]	TRANSMIT SERIAL DATA (lines 1, 2) — Output signals.
55, 51	RXD[1], RXD[2]	RECEIVE SERIAL DATA (lines 1, 2) — Input signals.
49, 47, 45, 43	PCS[0] — PCS[3]	PERIPHERAL CHIP SELECTS (lines 0—3) — Chip-select lines for peripheral devices.
48	SCK	SERIAL CLOCK — If SPI is enabled, the serial clock signal. If SPI is disabled, a general-purpose port E I/O signal.
44	MOSI	MASTER OUT/SLAVE IN — If SPI is enabled, the data master-out/slave-in signal. If SPI is disabled, a general-purpose port E I/O signal.
42	MISO	MASTER IN/SLAVE OUT — If SPI is enabled, the data master-in/slave-out signal. If SPI is disabled, a general-purpose port E I/O signal.
37, 35	MPWM[3], MPWM[19]	PULSE WIDTH MODULATOR (lines 3, 19) — Outputs.
34	IDVDD	IDENTIFICATION POWER — Special 3-volt power signal, from an external source, for the identification code (MID) signals.
33, 31, 29, 27	MDA[12], MDA[14], MDA[28], MDA[30]	MIOS DOUBLE ACTION (lines 12, 14, 28, 30) — Signals for the MIOS timer block.
32	TXDLCAN0	TRANSMIT CAN DATA (line 0) — CAN-protocol output signal.
30	RXDLCAN	RECEIVE CAN DATA — CAN-protocol input signal.
26, 25, 12	GND1	GROUND — Connection to the GROUND 1 plane.
10, 2	GND2	GROUND — Connection to the GROUND 2 plane.

		P2/J2		
PTJ2[100]	100	• •	99	AGND
AN[52]	98	• •	97	AGND
MAPIVREFL	96	• •	95	AGND
MAPIVREFH	94	• •	93	AGND
PTJ2[92]	92	• •	91	AGND
PTJ2[90]	90	• •	89	AGND
PTJ2[88]	88	• •	87	AGND
AGND	86	• •	85	AGND
GND	84	• •	83	GND
VDD3V	82	• •	81	PTJ2[81]
PTJ2[80]	80	• •	79	VDD5V
PTJ2[78]	78	• •	77	PTJ2[77]
PTJ2[76]	76	• •	75	PTJ2[75]
PTJ2[74]	74	• •	73	PTJ2[73]
PTJ2[72]	72	• •	71	PTJ2[71]
PTJ2[70]	70	• •	69	PTJ2[69]
PTJ2[68]	68	• •	67	PTJ2[67]
PTJ2[66]	66	• •	65	PTJ2[65]
GND	64	• •	63	GND
VDD3V	62	• •	61	PTJ2[61]
PTJ2[60]	60	• •	59	PTJ2[59]
PTJ2[58]	58	• •	57	PTJ2[57]
PTJ2[56]	56	• •	55	PTJ2[55]
PTJ2[54]	54	• •	53	PTJ2[53]
PTJ2[52]	52	• •	51	PTJ2[51]
PTJ2[50]	50	• •	49	PTJ2[49]
PTJ2[48]	48	• •	47	PTJ2[47]
GND	46	• •	45	VDD5V
VDD3V	44	• •	43	GND
PTJ2[42]	42	• •	41	PTJ2[41]
PTJ2[40]	40	• •	39	PTJ2[39]
PTJ2[38]	38	• •	37	PTJ2[37]
PTJ2[36]	36	• •	35	PTJ2[35]
PTJ2[34]	34	• •	33	PTJ2[33]
PTJ2[32]	32	• •	31	SDCPS
PTJ2[30]	30	• •	29	VDD5V
PTJ2[28]	28	• •	27	PTJ2[27]
PTJ2[26]	26	• •	25	PTJ2[25]
PTJ2[24]	24	• •	23	PTJ2[23]
PTJ2[22]	22	• •	21	PTJ2[21]
VDD3V	20	• •	19	PTJ2[19]
GND	18	• •	17	GND
GND2	16	• •	15	GND2
PTJ2[14]	14	• •	13	PTJ2[13]
PTJ2[12]	12	• •	11	PTJ2[11]
PTJ2[10]	10	• •	9	PTJ2[9]
PTJ2[8]	8	• •	7	PTJ2[7]
PTJ2[6]	6	• •	5	PTJ2[5]
PTJ2[4]	4	• •	3	PTJ2[3]
PTJ2[2]	2	• •	1	GND2

Figure 4-3 MAPI Connector Site P2/J2 Pin Assignments

Table 4-5 MAPI Connector Site P2/J2 Signal Descriptions

Pin	Mnemonic	Signal
100, 92, 90, 88, 81, 80, 78 — 65, 61 — 47, 42—32, 30, 28 — 21, 19, 14 — 2	PTJ2[x]	Pass through.
99, 97, 95, 93, 91, 89, 87 — 85	AGND	ANALOG GROUND — Analog ground connection for the analog-digital converter.
98	AN[52]	ANALOG IN (line 52) — Analog input channel for the analog-digital converter.
96	MAPIVREFL	MAPI VOLTAGE REFERENCE LOW — Low reference for voltage supplied via the MAPI ring.
94	MAPOIVREFH	MAPI VOLTAGE REFERENCE HIGH — High reference for voltage supplied via the MAPI ring.
84, 83, 64, 63, 46, 43, 18, 17	GND	GROUND.
82, 62, 44, 20	VDD3V	+3.3-volt power
79, 45, 29	VDD5V	+5-volt power.
31	SDCPS	SHUT DOWN CMB POWER SUPPLY — Input signal. If low, disables the CMB on-board power supply.
16, 15, 1	GND2	GROUND — Connection to the GROUND 2 plane.

P3/J3

VDD3V	100	99	VDD3V
PTJ3[98]	98	97	GND
PTJ3[96]	96	95	GND
PTJ3[94]	94	93	EXTAL
PTJ3[92]	92	91	GND
PTJ3[90]	90	89	PTJ3[89]
TSIZ[1]	88	87	TRST_b
TSIZ[0]	86	85	TCLK
DBEV_b	84	83	TMS
TDI	82	81	GND
TDO	80	79	RSTOUT_b
PTJ3[78]	78	77	RESET_b
IDVDD	76	75	SHS_b
VDD5V	74	73	PSTAT[3]
PTJ3[72]	72	71	PSTAT[2]
PTJ3[70]	70	69	PSTAT[1]
GND	68	67	PSTAT[0]
PTJ3[66]	66	65	GND
PTJ3[64]	64	63	GND
PTJ3[62]	62	61	GND
VDD3V	60	59	VDD3V
PTJ3[58]	58	57	PTJ3[57]
PTJ3[56]	56	55	GND
PTJ3[54]	54	53	PTJ3[53]
PTJ3[52]	52	51	PTJ3[51]
PTJ3[50]	50	49	GND
PTJ3[48]	48	47	PTJ3[47]
PTJ3[46]	46	45	PTJ3[45]
PTJ3[44]	44	43	GND
PTJ3[42]	42	41	PTJ3[41]
PTJ3[40]	40	39	PTJ3[39]
PTJ3[38]	38	37	IDVDD
PTJ3[36]	36	35	PTJ3[35]
PTJ3[34]	34	33	PTJ3[33]
PTJ3[32]	32	31	GND
PTJ3[30]	30	29	PTJ3[29]
PTJ3[28]	28	27	PTJ3[27]
PTJ3[26]	26	25	GND
AGND	24	23	AGND
PTJ3[22]	22	21	AGND
PTJ3[20]	20	19	AGND
AN[48]	18	17	AGND
AN[3]	16	15	AGND
AN[2]	14	13	AGND
AN[1]	12	11	AGND
AN[0]	10	9	AGND
AN[56]	8	7	AGND
AN[55]	6	5	AGND
AN[54]	4	3	AGND
AN[53]	2	1	AGND

Figure 4-4 MAPI Connector Site P3/J3 Pin Assignments

Table 4-6 MAPI Connector Site P3/J3 Signal Descriptions

Pin	Mnemonic	Signal
100, 99, 60, 59,	VDD3V	+3.3-volt power
98, 96, 94, 92, 90, 89, 78, 72, 70, 66, 64, 62, 58 — 56, 54 — 50, 48 — 44, 42 — 38, 36 — 32, 30 — 26, 22, 20	PTJ3[x]	Pass Through.
97, 95, 91, 81, 68, 65, 63, 61, 55, 49, 43, 31, 25	GND	GROUND
93	EXTAL	EXTERNAL CLOCK — Off-board clock signal.
88, 86	TSIZ[1], TSIZ[0]	TRANSFER SIZE (lines 1, 0) — Signals that indicate the size of an external bus transfer.
87	TRST_b	TEST RESET – Active-low input signal to the Schmitt trigger, asynchronously initializing the test controller.
85	TCLK	TEST CLOCK – Input signal that synchronizes the JTAG test logic. The TCK pin has an internal pullup resistor.
84	DBEV_b	DEBUG EVENT – Open-drain, active-low debug signal. If an input signal from an external command controller, causes the processor to enter debug mode. If an output signal, acknowledges that the MCU is in debug mode.
83	TMS	TEST MODE SELECT – Input signal that sequences the JTAG test controller's state machine, sampled on the rising edge of the TCK signal. The TMS pin has an internal pullup resistor.
82	TDI	TEST DATA INPUT – Serial input signal for JTAG test instructions and data, sampled on the rising edge of the TCK signal. The TDI pin has an internal pullup resistor.
80	TDO	TEST DATA OUTPUT – Serial output signal for JTAG test instructions and data. Tri-stateable and actively driven in the Shift-IR and Shift-DR controller states, this signal changes on the falling edge of the TCK signal.
79	RSTOUT_b	RESET OUT – Active-low output signal, controlled by the processor, that resets external components. Activation of any internal reset sources asserts this line.
77	RESET_b	RESET IN – Active-low input signal that starts a system reset: a reset of the processor and most peripherals. This signal does not affect the debug module (which the system provides via the TRST* line).
76, 37	IDVDD	IDENTIFICATION POWER — Special 3-volt power signal, from an external source, for the identification code (MID) signals.
75	SHS	SHOW CYCLE STROBE — Output signal indicating that address and data are valid.
74	VDD5V	+5-volt power.
73, 71, 69, 67	PSTAT[3] — PSTAT[0]	PROCESSOR STATUS (lines 3—0) — Output signals that provide external status indications for the resident MCU.
37	IDVDD	MID (identification code) lines 9—4 — Signals that identify the host processor board.

Table 4-6 MAPI Connector Site P3/J3 Signal Descriptions (Continued)

Pin	Mnemonic	Signal
24, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1	AGND	ANALOG GROUND — Analog ground connection for the analog-digital converter.
18, 16, 14, 12, 10, 8, 6, 4, 2	AN[48], AN[3] — AN[0], AN[56] — AN[53]	ANALOG IN (lines 48, 3—0, 55—53) — Analog input channels for the analog-digital converter.

P4/J4

VDD5V	100	99	VDD3V	
CSE[1]	98	97	GND	
GND	96	95	CLKOUT	
CSE[0]	94	93	GND	
PTJ4[92]	92	91	CS_b[3]	
PTJ4[90]	90	89	CS_b[2]	
OE_b	88	87	CS_b[1]	
EB_b[0]	86	85	PTJ4[85]	
EB_b[1]	84	83	GND	
EB_b[3]	82	81	RW_b	
EB_b[2]	80	79	SHS_b	
TEA_b	78	77	TA_b	
GND	76	75	GND	
PTJ4[74]	74	73	PTJ4[73]	
PTJ4[72]	72	71	PTJ4[71]	
PTJ4[70]	70	69	PTJ4[69]	
PTJ4[68]	68	67	PTJ4[67]	
ADDR[22]	66	65	PTJ4[65]	
ADDR[20]	64	63	ADDR[21]	
ADDR[18}	62	61	ADDR[19]	
ADDR[16}	60	59	ADDR[17]	
GND	58	57	GND	
ADDR[14]	56	55	ADDR[15]	
ADDR[12]	54	53	ADDR[13]	
ADDR[10]	52	51	ADDR[11]	
ADDR[8]	50	49	ADDR[9]	
ADDR[6]	48	47	ADDR[7]	
ADDR[4}	46	45	ADDR[5]	
ADDR[2]	44	43	ADDR[3]	
PTJ4[42]	42	41	ADDR[1]	
GND	40	39	GND	
DATA[30]	38	37	DATA[31]	
DATA[28]	36	35	DATA[29]	
DATA[26]	34	33	DATA[27]	
DATA[24]	32	31	DATA[25]	
DATA[22]	30	29	DATA[23]	
GND	28	27	GND	
DATA[20]	26	25	DATA[21]	
DATA[18]	24	23	DATA[19]	
DATA[16]	22	21	DATA[17]	
DATA[14]	20	19	DATA[15]	
DATA[12]	18	17	DATA[13]	
GND	16	15	GND	
DATA[10]	14	13	DATA[11]	
DATA[8]	12	11	DATA[9]	
DATA[6]	10	9	DATA[7]	
DATA[4]	8	7	DATA[5]	
DATA[2]	6	5	DATA[3]	
DATA[0]	4	3	DATA[1]	
VDD3V	2	1	VDD3V	

Figure 4-5 MAPI Connector Site P4/J4 Pin Assignments

Table 4-7 MAPI Connector Site P4/J4 Signal Descriptions

Pin	Mnemonic	Signal
100	VDD5V	+5-volt power.
99, 2, 1	VDD3V	+3.3-volt power.
98, 94	CSE1, CSE0	EMULATION CHIP SELECTS (lines 1, 0) — Emulation-mode output chip-select signals.
97, 96, 93, 83, 76, 75, 58, 57, 40, 39, 28, 27, 16, 15	GND	GROUND
95	CLKOUT	CLOCK OUTPUT — An external, low-frequency clock source from the processor.
92, 90, 85, 74 — 67, 65, 42	PTJ4[x]	Pass Through
91, 89, 87	CS_b[3] — CS_b[1]	CHIP SELECTS (lines 3—1) — Active-low output lines that provide chip selects to external devices.
88	OE_b	OUTPUT ENABLE — Active-low output that indicates that a bus access is a read access; enables slave devices to drive the data bus.
86, 84, 82, 80	EB_b[0] — EB_b[3] (not in exact order)	ENABLE BYTES 0, 1, 3, 2 — Active-low outputs active during an operation to corresponding data bits (D31-D24 for enable byte 0, D23-D16 for enable byte 1, D15-D8 for enable byte 2, D7-D0 for enable byte 3).
81	RW_b	READ/WRITE ENABLE — Active-low signal indicating that the current bus access is a write access. Otherwise, the current bus access is a read access.
79	SHS_b	SHOW CYCLE STROBE — Output signal indicating that address and data are valid.
78	TEA_b	TRANSFER ERROR ACKNOWLEDGE — Active-low input that indicating that a bus transfer error has occurred.
77	TA_b	TRANSFER ACKNOWLEDGE — Active-low output indicating completion of a data transfer, for either a read or a write cycle.
66, 64 — 59, 56—43, 41	ADDR[22] — ADDR[0] (not in exact order)	ADDRESS BUS (lines 22—0) — Output lines for addressing external devices. These lines change state only during external-memory accesses.
38—29, 26—17, 14—3	DATA[31] — DATA[0] (not in exact order)	DATA BUS (lines 31–0) — Bi-directional data lines for accessing external memory. A hardware reset or no external-bus activity hods these lines in their previous logic state.

Index

U

Revision History

Revision Number	Date	Author	Summary of Changes
Original	July 2000	MTC DDOC	Original document.

Revision History

This manual is a product of the Motorola M•CORE Technology Center Design Documentation team. Technical writing, illustration, and production editing performed with Adobe Framemaker running on multiple platforms. Printed by Ken Cook, Inc. in Milwaukee, Wisconsin.