HC05P9EMUM/D

APRIL 1994

HC05P9
EMULATOR MODULE
USER'S MANUAL

PRELIMINARY

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

CONTENTS

FIGURES

TABLES

CHAPTER 1
INTRODUCTION

This user's manual explains connection, configuration, and operation information specific to the HC05P9EM Emulator Module (P9EM). The P9EM makes possible emulation and debugging of target systems based on these microcontroller units (MCUs):

MC68HC05P1 MC68HC05P4 MC68HC(7)05P6

MC68HC05P7 MC68HC(7)05P9

The P9EM can be part of two development systems. This chapter describes those systems and explains the layout of the P9EM.

The factory installs an MC68HC705P9 resident MCU in the P9 EM, then ships the P9EM with an extra MC68HC05P6 MCU. Chapters 2 and 3 explain when it is appropriate to use this substitute resident MCU.

1.1 DEVELOPMENT SYSTEMS

Your P9EM can be part of two Motorola development tools: the HC05 Motorola Modular Development System (MMDS05) or the M68HC05EVS Evaluation System (HC05EVS).

1.1.1 Motorola Modular Development System (MMDS05)

The MMDS05 is an emulator system that provides a bus state analyzer and real-time memory windows. The unit's integrated development environment includes an editor, an assembler, user interface, and source-level debug. A complete MMDS05 consists of:

- **a station module** — the metal MMDS05 enclosure, containing the control board and the internal power supply. The station module has a connector for serial communication with the host computer, as well as connectors (pods) for the logic clip cable assemblies.
- **an emulator module (EM)** — such as the P9EM: a separately purchased printed circuit board that enables system functionality for a specific set of MCUs. The EM fits into the station module through a removable panel in the enclosure top. The EM has a connector for the target cable.

- **two logic clip cable assemblies** — twisted-pair cables that connect the station module to your target system, a test fixture, a clock, or any other circuitry useful for evaluation or analysis. One end of each cable assembly has a molded connector, which fits into station-module pod A or pod B. Leads at the other end of each cable terminate in female probe tips. Ball clips come with the cable assemblies.
- **a 9-lead RS-232 serial cable** — the cable that connects the station module to the host computer RS-232 port.
- **a 9- to 25-pin adapter** — a molded assembly that lets you connect the 9-lead cable to a 25-pin serial port.
- **system software** — software, on 3-1/2 inch, high-density diskettes.
- **MMDS05 documentation** — an MMDS05 operations manual (MMDS05OM/D) and the appropriate EM user's manual.

As mentioned, your P9EM gives the MMDS05 the ability to emulate several MC68HC05P-series MCUs. By substituting a different EM, you can enable your MMDS05 to emulate an MCU of a different series. (Your Motorola representative can explain all the EMs available.)

Chapter 2 explains how to configure and use your P9EM as part of an MMDS05 system. For information about MMDS05 software or the station module, see the MMDS05 operations manual. For layout and configuration information pertaining to a different EM, see the corresponding EM user's manual.

1.1.2 M68HC05EVS Evaluation System (HC05EVS)

An HC05EVS is an economical, two-board tool for designing, debugging, and evaluating target systems based on an MC68HC05 MCU. A complete HC05EVS consists of:

- **a platform board (PFB)** — the bottom board, which supports the emulator module. The platform board has connectors for power and for the terminal or host computer.
- **an emulator module (EM)** — such as the P9EM: a separately purchased printed circuit board that enables system functionality for a specific set of MCUs. The EM fits onto the PFB. The EM has connectors for a target cable and for a cable to a logic analyzer.
- **an RS-232 serial cable** — the cable that connects the PFB to the host computer RS-232 port.
- **system software** — software, on 5-1/4 inch diskettes.
- **HC05EVS documentation** — an HC05EVS operations manual (HC05EVSOM/D) and the appropriate EM user's manual.

As mentioned, your HC05P9 EVS emulates several MC68HC05P-series MCUs. By substituting a different EM, you can enable your HC05EVS to emulate an MCU of a different series. (Your Motorola representative can explain all the EMs available.)

Chapter 3 explains how to configure and use your P9EM as part of an HC05EVS system. For information about HC05EVS software or the platform board, see the HC05EVS operations manual. For layout and configuration information pertaining to a different EM, see the corresponding EM user's manual.

1.2 EM LAYOUT

Figure 1-1 shows the layout of the P9EM. Jumper header J1 lets you select the clock-signal source. Jumper header J2 controls the path of a target reset .

Connector P4 is the target connector: a separately purchased, flexible cable runs from this connector to the MCU socket of your target system. If you install the P9EM in the MMDS05 station module, the target cable passes through the slit in the station module enclosure. Connector P1 connects to a logic analyzer. (Connector P1 has more significance for an EVS system, as an MMDS05 system includes a bus analyzer.)

Expansion header connectors P2 and P3 are on the bottom of the board. These connectors connect together the EM and the control board (for an MMDS05) or the EM and the platform board (for an EVS).

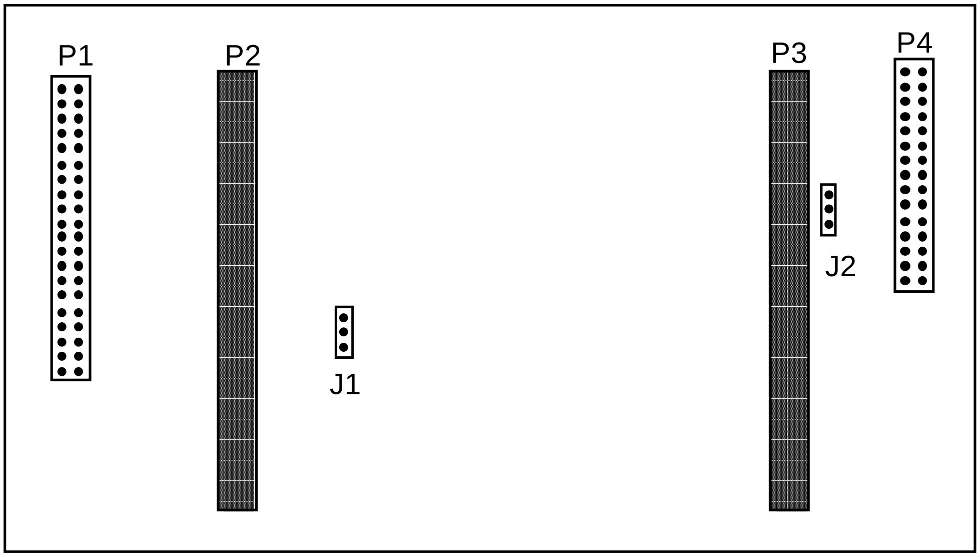
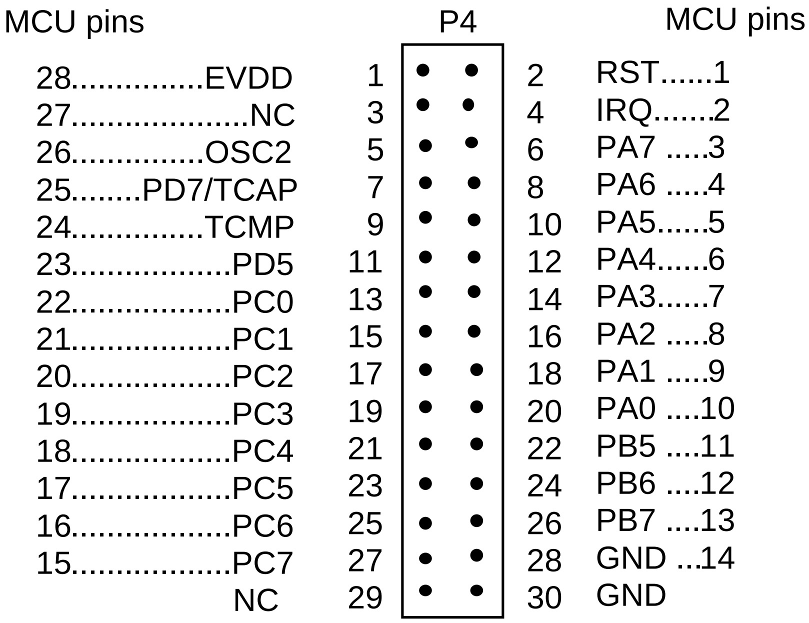


Figure 1-1. HC05P9 Emulator Module

1.3 TARGET CONNECTOR AND CABLE INFORMATION

Figure 1-2 shows the pin assignments for connector P4, the target connector. The italic numbers correlate the P4 pins with the pins of your MCU. Table 1-1 lists the signal descriptions for this connector.



MCU pins	P4		MCU pins
28..............EVDD	1	2	RST......1
27....................NC	3	4	IRQ.......2
26..............OSC2	5	6	PA7 3
25........PD7/TCAP	7	8	PA6 4
24..............TCMP	9	10	PA5......5
23..................PD5	11	12	PA4......6
22..................PC0	13	14	PA3......7
21..................PC1	15	16	PA2 8
20..................PC2	17	18	PA1 9
19..................PC3	19	20	PA0 10
18..................PC4	21	22	PB5 11
17..................PC5	23	24	PB6 12
16..................PC6	25	26	PB7 13
15..................PC7	27	28	GND ...14
NC	29	30	GND

Figure 1-2. Connector P4 Pin Assignments

Table 1-1. Connector P4 Signal Descriptions

Pin	Mnemonic	Signal
1	EVDD	EXTERNAL VOLTAGE DETECT - VDD input signal from the target system, for detection of target voltage.
2	RST	TARGET RESET - Active-low input signal from the target system that starts a system reset.
3, 29	-----	No connection
4	IRQ	TARGET INTERRUPT REQUEST - Active-low input signal from the target that asynchronously applies an MCU interrupt.
5	OSC2	OSCILLATOR 2 - Output clock signal at twice the internal bus frequency.
6, 8, 10, 12, 14, 16, 18, 20	PA7 — PA0	PORT A (bits 7—0) - General-purpose I/O lines controlled by software via data direction and data registers.
7	PD7 / TCAP	PORT D (bit 7) - General-purpose input-only line. TIMER CAPTURE - Input signal used by the input capture feature of the MCU programmable timer system.
9	TCMP	TIMER COMPARE - Output signal used by the output compare feature of the MCU programmable timer system.
11	PD5	PORT D (bit 5) - General-purpose I/O line controlled by software via data direction and data registers.
13, 15, 17, 19, 21, 23, 25, 27	PC0 — PC7	PORT C (bits 0—7) - General-purpose I/O lines controlled by software via data direction and data registers.
22, 24, 26	PB5 — PB7	PORT B (bits 5—7) - General-purpose I/O lines controlled by software via data direction and data registers.
28, 30	GND	GROUND

Connect your P9EM to your target system via a separately-purchased, target cable and head assembly. Figure 1-3 is a generic representation of such an assembly, which consists of a flexible ribbon cable and a target head that fits into the MCU socket of the target system.

- The Motorola assembly consists of **cable M68CBL05A** and **target head M68TA05P9P28**.

- To make your own assembly, use 28-lead ribbon cable between a DIP connector (3M part no. 3728-4000 or equivalent) and a wiremount socket (3M part no. 3419-6030 or equivalent).

- The **M68DIP28SOIC** adapter accommodates surface-mount packages.

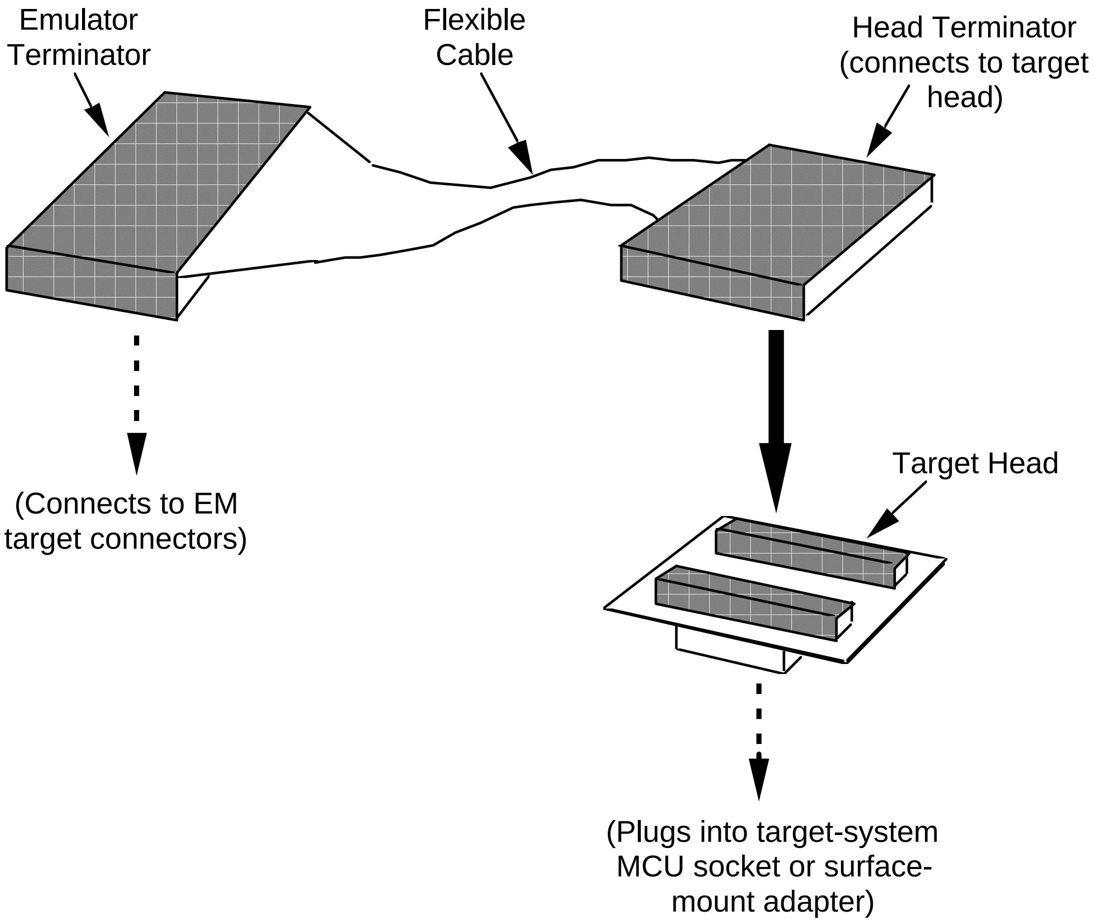

Emulator Terminator
Flexible Cable
Head Terminator (connects to target head)
(Connects to EM target connectors)
Target Head
(Plugs into target-system MCU socket or surface-mount adapter)

Figure 1-3. Target Cable and Head Assembly

As Figure 1-3 shows, connect the small cable terminator to the target head. Connect the large cable terminator to the P9EM target connector (J4). Plug the target head into the MCU socket of your target system. (For an SOIC package, solder the surface-mount adapter to the target system board, then plug the target head onto the adapter.)

If you install the P9EM in the MMDS05 station module, run the target cable through the slit in the station module enclosure.

As connector P1 has more importance to an HC05EVS than to an MMDS05, pin assignments for connector P1 are in Chapter 3.

To configure your P9EM for use as part of an MMDS05, follow the instructions of Chapter 2. To configure your P9EM for use as part of an HC05EVS, follow the instructions of Chapter 3.

CHAPTER 2

MMDS05 CONFIGURATION AND OPERATION

This chapter explains how to configure and use your P9EM as part of an MMDS05. For other parts of system installation or configuration, see the MMDS05 operations manual (MMDS05OM/D). (If you will use your P9EM as part of an HC05EVS system, go to Chapter 3 for corresponding information.)

Paragraph 2.1 explains how to set P9EM jumper headers. Paragraph 2.2 explains considerations pertaining to the MCU you emulate. Paragraph 2.3 explains remaining system installation.

Note that you can configure a P9EM already installed in the MMDS05 station module enclosure. To do so, switch off station-module power, remove the panel, then follow the guidance of this chapter.

CAUTION

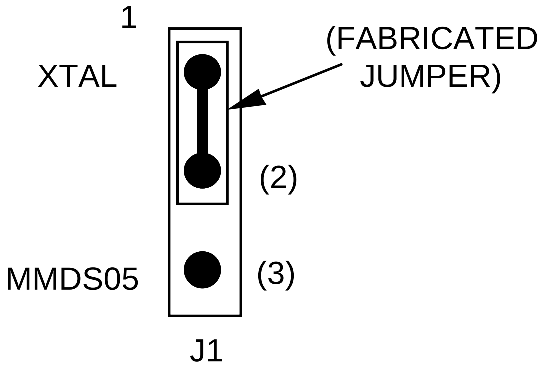
> Be sure to switch off power if you reconfigure an installed EM. Reconfiguring EM jumper headers with power left on can damage MMDS05 circuits.

2.1 SETTING P9EM JUMPER HEADERS

Your P9EM has two jumper headers, J1 and J2. Paragraphs 2.1.1. and 2.1.2 explain how to configure these headers.

2.1.1 Clock Source Select Header (J1)

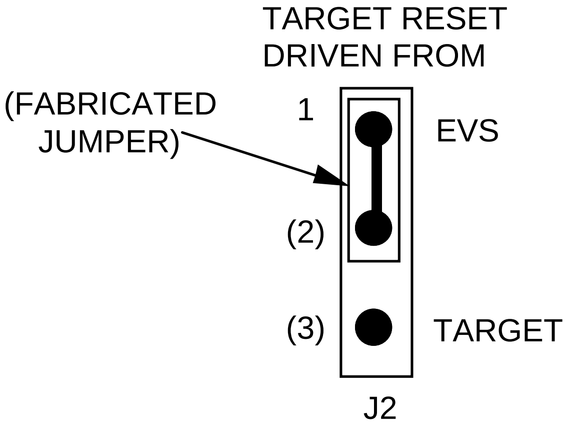
Jumper header J1 determines the source of the clock signal. The diagram below shows the factory configuration: the fabricated jumper between pins 1 and 2 selects the P9EM canned oscillator clock source.



Alternatively, you may use a clock source originating from the control board. To do so, reposition the J1 jumper between pins 2 and 3, then use the MMDS05 OSC command to select a frequency.

2.1.2 Reset Select Header (J2)

Jumper header J2 controls the path of a target reset. The diagram below shows the factory configuration: the fabricated jumper between pins 1 and 2 enables your software to control resets. This is the only correct J2 configuration for a P9EM that is part of an MMDS05.



NOTE

The J2 pins 2-and-3 configuration is not correct for a P9EM that is part of an MMDS05 system. Such a configuration would interfere with correct operation of the RESETIN and RESETOUT commands.

2.2 THE EMULATED MCU

Your MMDS05 system uses a specific personality file for the MCU type being emulated. For example, to emulate an MC68HC05P9 or MC68HC705P9 MCU, the system uses personality file 003FEVxx.MEM. Table 2-1 lists all the personality files that pertain to the P9EM. Note that personality file names follow the pattern **`00ZZZVxx.MEM`**, where **`ZZZ`** is the EM identifier or MCU name, and **`xx`** is the version of the file.

Table 2-1. MCUs and Personality Files

Emulated MCU	Personality file	Resident MCU	Comments
MC68HC(7)05P9	003FEVxx.MEM	MC68HC705P9	See paragraphs 2.2.1 and 2.2.2.
MC68HC05P1	000P1Vxx.MEM	MC68HC705P9	See paragraphs 2.2.1 and 2.2.3.
MC68HC05P4	000P4Vxx.MEM	MC68HC705P6	See paragraphs 2.2.1 and 2.2.4.
MC68HC(7)05P6	000P6Vxx.MEM	MC68HC705P6	See paragraphs 2.2.1 and 2.2.5.
MC68HC05P7	000P7Vxx.MEM	MC68HC705P9	See paragraphs 2.2.1 and 2.2.6.

MMDS05 software loads the personality file upon powerup; the default personality file is 003FEVxx.MEM. To replace this file with a different one, use the LOADMEM command. (For more information about the LOADMEM command, see the explanation in the MMDS05 operations manual.)

Alternatively, you can insert the line LOADMEM 00ZZZVxx.MEM in the STARTUP.05 script file. This way, the system automatically loads the specified personality file as part of system startup. (For more information about the STARTUP.05 script file, see the SCRIPT command explanation in the MMDS05 operations manual.)

NOTE

Part of a normal reset is resetting the port data direction registers (DDRs). But during emulation of any MCU listed above, a COP reset does *not* include resetting the port DDRs. This is because a COP reset is an internal event. The emulator rebuilds the ports to be external to the MCU; the emulator is not notified of a COP reset.

2.2.1 Resident MCU Substitution

The socket at P9EM location U5 contains the resident MCU. The factory ships the P9EM with an MC68HC705P9 resident MCU, which is appropriate for emulating MC68HC05P1, MC68HC05P7, MC68HC05P9, or MC68HC705P9 MCUs.

The factory also ships a substitute MC68HC705P6 resident MCU with the P9EM. This MCU is appropriate for emulating MC68HC05P4, MC68HC05P6, or MC68HC705P6 MCUs. Paragraph 2.2.4 explains information unique to MC68HC05P4 emulation; paragraph 2.2.5 explains information unique to MC68HC05P6 or MC68HC705P6 emulation.

Table 2-1 shows the resident MCU appropriate for your emulation. Also note paragraphs 2.2.2 through 2.2.6, below, for the information specific to the MCU you emulate.

Before you install a substitute MCU, make sure that it is properly programmed, per paragraphs 2.2.2 through 2.2.6. When you are ready to install the substitute MCU, turn off MMDS05 power and make sure that you are grounded. Then remove the MCU from P9EM location U5, insert the new resident MCU, and restore power.

2.2.2 P9 Emulation

To emulate either an MC68HC05P9 or MC68HC705P9, use the MC68HC705P9 for the resident MCU. Note that bit values of address $900 control three mask options:

- Bit 2 determines the first bit for transmissions via the serial I/O port:
  - If the value of this bit is 0, the development system sends and receives the most significant bit (MSB — bit 7) first.
  - If the value of this bit is 1, the development system sends and receives the least significant bit (LSB — bit 0) first.
- Bit 1 determines interrupt request (IRQ) sensitivity:
  - If the value of this bit is 0, the IRQ pin is edge sensitive.
  - If the value of this bit is 1, the IRQ pin is edge and level sensitive.
- Bit 0 enables or disables the computer operating properly (COP) watchdog timer:
  - The value 0 for this bit disables the COP.
  - The value 1 for this bit enables the COP.

The MC68HC705P9 MCU associates two internal mask option registers (MORs) with address $900: the MOR EPROM register, and the MOR latch register. The MOR EPROM register stores the defined reset values for these bits; your code cannot change values in the MOR EPROM register. The values in the MOR latch register control system operation; your code *can* change these bit values in the MOR latch register during execution. During a reset, the MCU loads the contents of the MOR EPROM register into the MOR latch register, restoring the reset state for each bit. (Changing the reset values stored in the MOR EPROM register requires an MCU programmer.)

Bit 2 of the EPROM PROG register ($1C) is the latch bit, or LATCH, which controls address $900 reads and writes. If LATCH = 0 (the default value), reads pertain to the MOR EPROM register and writes pertain to the MOR latch register. If LATCH = 1, reads pertain to the MOR latch register (writes are undefined).

In other words, the default LATCH value lets you change values of the MOR latch register. But to see those changed values, you must first set LATCH. Subsequently, to make other changes to the MOR latch register, you must clear LATCH.

2.2.3 P1 Emulation

To emulate an MC68HC05P1, use the MC68HC705P9 for the resident MCU. Remember that the MC68HC05P1 does not include all the MC68HC705P9 features. For accurate MC68HC05P1 emulation, do not use any of the additional MC68HC705P9 features.

For example, the MC68HC705P9 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P1 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

For more information about the MC68HC05P1 and MC68HC705P9 features, see the corresponding technical data books.

2.2.4 P4 Emulation

To emulate an MC68HC05P4, use the MC68HC705P6 substitute for the resident MCU. Remember that the MC68HC05P4 does not include all the MC68HC705P6 features. For accurate MC68HC05P4 emulation, do not use any of the additional MC68HC705P6 features.

For example, the MC68HC705P6 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P4 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

To emulate any mask options, you must program the MOR correctly before you insert the MC68HC705P6 MCU at P9EM location U5. Such programming requires a dedicated programmer.

To enable the COP, write #4 to location $1F. If the COP bit of the MOR is set and you pull the IRQ signal low, you should periodically reset the COP. Otherwise, the system will generate a COP reset, even if you have not enabled the COP.

For more information about the MC68HC05P4 and MC68HC705P6 features, see the corresponding technical data books.

2.2.5 P6 Emulation

To emulate either an MC68HC05P6 or MC68HC705P6, use the MC68HC705P6 substitute for the resident MCU. Note that MC68HC705P6 mask options are different from those of other HC05P-series MCUs. You must program the MOR correctly for the options you want before you insert the MC68HC705P6 MCU at P9EM location U5. Such programming requires a dedicated programmer.

To enable the COP, write #4 to location $1F. If the COP bit of the MOR is set and you pull the IRQ signal low, you should periodically reset the COP. Otherwise, the system will generate a COP reset, even if you have not enabled the COP.

For more information about the MC68HC705P6 MOR, see the MC68HC705P6 technical data book.

2.2.6 P7 Emulation

To emulate an MC68HC05P7, use the MC68HC705P9 for the resident MCU. Remember that the MC68HC05P7 does not include all the MC68HC705P9 features. For accurate MC68HC05P7 emulation, do not use any of the additional MC68HC705P9 features.

For example, the MC68HC705P9 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P7 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

For more information about the MC68HC05P7 and MC68HC705P9 features, see the corresponding technical data books.

2.3 REMAINING SYSTEM INSTALLATION

When you have configured headers J1 and J2, and made sure that the resident MCU is appropriate, you are through with P9EM configuration:

- To install the P9EM in an MMDS05 station module, remove the panel from the station-module top. Fit together the P9EM and the control board: P9EM connectors P2 and P3 connect to control-board connectors P11 and P12, respectively. Snap the corners of the EM onto the plastic standoffs. Connect the target cable, if appropriate. Replace the panel.
- If your P9EM already is installed in the station module, reconnect the target cable (if necessary). Replace the panel.

At this point, you are ready to make remaining cable connections, as necessary, and restore power. For instructions, consult the MMDS05 operations manual (MMDS05OM/D).

CHAPTER 3

HC05EVS CONFIGURATION AND OPERATION

This chapter explains how to configure your P9EM for use as part of an HC05EVS. For other parts of system installation or configuration, see the HC05EVS operations manual (HC05EVSOM/D). (If you will use your P9EM as part of an MMDS05 system, go to Chapter 2 for configuration information.)

Paragraph 3.1 explains how to set P9EM jumper headers. Paragraph 3.2 explains considerations pertaining to the MCU you emulate. Paragraph 3.3 lists limitations on using the P9EM in an HC05EVS. Paragraph 3.4 explains pin assignments and signal descriptions for the logic analyzer connector (P1). Paragraph 3.5 explains remaining system configuration.

Note that you can configure a P9EM already installed on the HC05EVS platform board. To do so, disconnect platform-board power, then follow the guidance of this chapter.

CAUTION

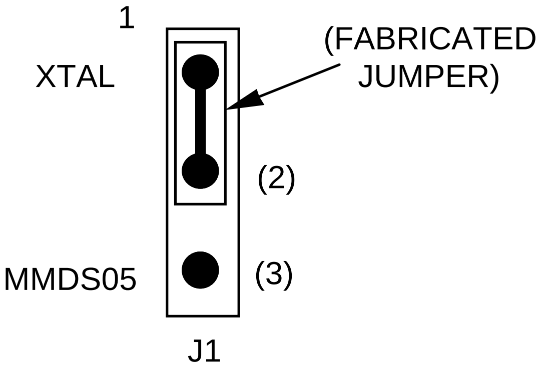
> Be sure to disconnect power if you reconfigure an installed EM. Reconfiguring EM jumper headers with power left on can damage HC05EVS circuits.

3.1 SETTING P9EM JUMPER HEADERS

Your P9EM has two jumper headers, J1 and J2. Paragraphs 3.1.1. and 3.1.2 explain how to configure these headers.

3.1.1 Clock Source Select Header (J1)

Jumper header J1 determines the source of the clock signal. The diagram below shows the factory configuration: the fabricated jumper between pins 1 and 2 selects the P9EM canned oscillator clock source. This is the only correct configuration for a P9EM that is part of an HC05EVS.

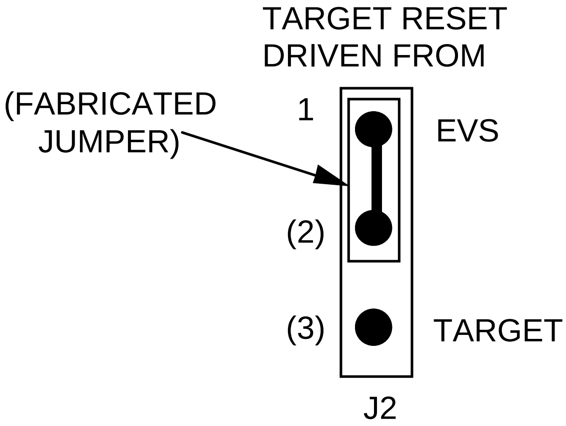


NOTE

The J1 pins 2-and-3 configuration is not correct for a P9EM that is part of an HC05EVS system.

3.1.2 Reset Select Header (J2)

Jumper header J2 controls the path of a target reset. The diagram below shows the factory configuration: the fabricated jumper between pins 1 and 2 enables your P9EVS to reset the target when you press the user or master reset switch.



Alternatively, you may enable the target system to reset the P9EVS. To do so:

- Reposition the P9EM J2 jumper between pins 2 and 3, and
- Insert a fabricated jumper into platform-board header J3.

3.2 RESIDENT MCU SUBSTITUTION

The socket at P9EM location U5 contains the resident MCU. The factory ships the P9EM with an MC68HC705P9 resident MCU, which is appropriate for emulating MC68HC05P1, MC68HC05P7, MC68HC05P9, or MC68HC705P9 MCUs.

The factory also ships a substitute MC68HC705P6 resident MCU with the P9EM. This MCU is appropriate for emulating MC68HC05P4, MC68HC05P6, or MC68HC705P6 MCUs. Paragraph 3.2.3 explains information unique to MC68HC05P4 emulation; paragraph 3.2.4 explains information unique to MC68HC05P6 or MC68HC705P6 emulation.

Table 3-1 shows the resident MCU appropriate for your emulation. Also note paragraphs 3.2.1 through 3.2.5, below, for the information specific to the MCU you emulate.

Table 3-1. Resident MCUs

Emulated MCU	Resident MCU	Comments
MC68HC(7)05P9	MC68HC705P9	See paragraph 3.2.1.
MC68HC05P1	MC68HC705P9	See paragraph 3.2.2.
MC68HC05P4	MC68HC705P6	See paragraph 3.2.3.
MC68HC(7)05P6	MC68HC705P6	See paragraph 3.2.4.
MC68HC05P7	MC68HC705P9	See paragraph 3.2.5.

Before you install a substitute MCU, make sure that it is properly programmed, per paragraphs 3.2.1 through 3.2.5. When you are ready to install the substitute MCU, turn off HC05EVS power and make sure that you are grounded. Then remove the MCU from P9EM location U5, insert the new resident MCU, and restore power.

NOTE

Part of a normal reset is resetting the port data direction registers (DDRs). But during emulation of any MCU listed above, a COP reset does not include resetting the port DDRs. This is because a COP reset is an internal event. The emulator rebuilds the ports to be external to the MCU; the emulator is not notified of a COP reset.

3.2.1 P9 Emulation

To emulate either an MC68HC05P9 or MC68HC705P9, use the MC68HC705P9 for the resident MCU. Note that bit values of address $900 control three mask options:

- Bit 2 determines the first bit for transmissions via the serial I/O port:
  - If the value of this bit is 0, the development system sends and receives the most significant bit (MSB — bit 7) first.
  - If the value of this bit is 1, the development system sends and receives the least significant bit (LSB — bit 0) first.
- Bit 1 determines interrupt request (IRQ) sensitivity:
  - If the value of this bit is 0, the IRQ pin is edge sensitive.
  - If the value of this bit is 1, the IRQ pin is edge and level sensitive.
- Bit 0 enables or disables the computer operating properly (COP) watchdog timer:
  - The value 0 for this bit disables the COP.
  - The value 1 for this bit enables the COP.

The MC68HC705P9 MCU associates two internal mask option registers (MORs) with address $900: the MOR EPROM register, and the MOR latch register. The MOR EPROM register stores the defined reset values for these bits; your code cannot change values in the MOR EPROM register. The values in the MOR latch register control system operation; your code *can* change these bit values in the MOR latch register during execution. During a reset, the MCU loads the contents of the MOR EPROM register into the MOR latch register, restoring the reset state for each bit. (Changing the reset values stored in the MOR EPROM register requires an MCU programmer.)

Bit 2 of the EPROM PROG register ($1C) is the latch bit, or LATCH, which controls address $900 reads and writes. If LATCH = 0 (the default value), reads pertain to the MOR EPROM register and writes pertain to the MOR latch register. If LATCH = 1, reads pertain to the MOR latch register (writes are undefined).

In other words, the default LATCH value lets you change values of the MOR latch register. But to see those changed values, you must first set LATCH. Subsequently, to make other changes to the MOR latch register, you must clear LATCH.

3.2.2 P1 Emulation

To emulate an MC68HC05P1, use the MC68HC705P9 for the resident MCU. Remember that the MC68HC05P1 does not include all the MC68HC705P9 features. For accurate MC68HC05P1 emulation, do not use any of the additional MC68HC705P9 features.

For example, the MC68HC705P9 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P1 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

For more information about the MC68HC05P1 and MC68HC705P9 features, see the corresponding technical data books.

3.2.3 P4 Emulation

To emulate an MC68HC05P4, use the MC68HC705P6 substitute for the resident MCU. Remember that the MC68HC05P4 does not include all the MC68HC705P6 features. For accurate MC68HC05P4 emulation, do not use any of the additional MC68HC705P6 features.

For example, the MC68HC705P6 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P4 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

To emulate any mask options, you must program the MOR correctly before you insert the MC68HC705P6 MCU at P9EM location U5. Such programming requires a dedicated programmer.

To enable the COP, write #4 to location $1F. If the COP bit of the MOR is set and you pull the IRQ signal low, you should periodically reset the COP. Otherwise, the system will generate a COP reset, even if you have not enabled the COP.

For more information about the MC68HC05P4 and MC68HC705P6 features, see the corresponding technical data books.

3.2.4 P6 Emulation

To emulate either an MC68HC05P6 or MC68HC705P6, use the MC68HC705P6 substitute for the resident MCU. Note that MC68HC705P6 mask options are different from those of other HC05P-series MCUs. You must program the MOR correctly for the options you want before you insert the MC68HC705P6 MCU at P9EM location U5. Such programming requires a dedicated programmer.

To enable the COP, write #4 to location $1F. If the COP bit of the MOR is set and you pull the IRQ signal low, you should periodically reset the COP. Otherwise, the system will generate a COP reset, even if you have not enabled the COP.

For more information about the MC68HC705P6 MOR, see the MC68HC705P6 technical data book.

3.2.5 P7 Emulation

To emulate an MC68HC05P7, use the MC68HC705P9 for the resident MCU. Remember that the MC68HC05P7 does not include all the MC68HC705P9 features. For accurate MC68HC05P7 emulation, do not use any of the additional MC68HC705P9 features.

For example, the MC68HC705P9 has an analog-to-digital (A/D) converter; the A/D control register is at address $001E. Before beginning P7 emulation, you should do a reset; this turns off the A/D converter. During emulation, avoid writing to address $001E, as this could turn on the A/D converter.

For more information about the MC68HC05P7 and MC68HC705P9 features, see the corresponding technical data books.

3.3 EVS LIMITATIONS

This paragraph explains the limitations that apply to using your P9EM in an HC05EVS. (These limitations apply regardless of the resident MCU.)

Limitation 1. CLI/RTI. You may not trace a clear interrupt mask (CLI) or return from interrupt (RTI) instruction with an interrupt enabled and pending, due to MCU interrupt handling. Attempting such a trace causes an interrupt in the monitor map; this forces a software reset of the HC05EVS. User breakpoints remain in the user map as SWI instructions; you must remove such SWI instructions.

Limitation 2. Branch. Do not trace a conditional branch instruction (such as BRCLR) that branches to itself. As the monitor places an SWI instruction on the object of the branch, the system never would execute the instruction. However, it would appear to you that the instruction had executed. You may enter a G command while the PC points to this type of instruction as long as the instruction is not a breakpoint address.

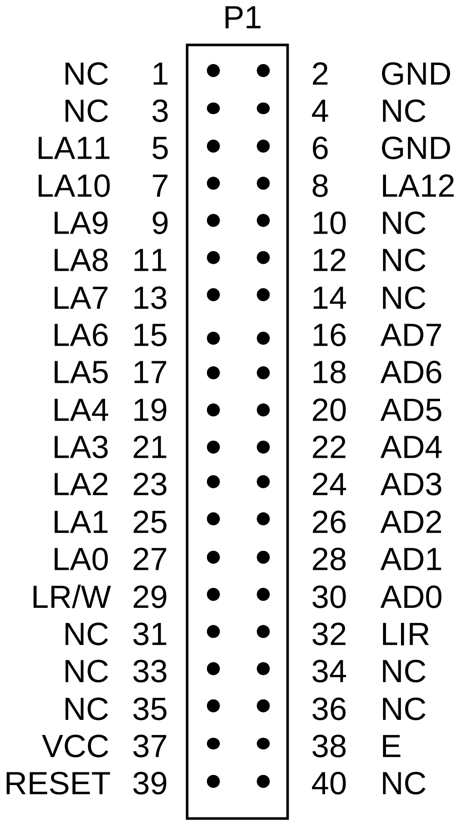
Limitation 3. IRQ/SWI. Whenever possible, avoid mixing interrupt requests (IRQs) and user software interrupts (SWIs). This prevents a possible IRQ-SWI timing problem: a concurrent hardware interrupt and SWI could cause an HC05EVS failure that could stop program execution. To recover from such a failure (which occurs very infrequently, statistically), press the master reset switch (SW3).

Limitation 4. Memory Map. You must be aware of the memory map of the MCU being simulated; be sure to use only valid ROM locations. The HC05EVS does not provide protection to limit user programs to the exact amount of MCU ROM available. (For information about memory maps, consult the technical data book for the MCU.)

Limitation 5. COP. The computer operating properly (COP) update register serves as an EPROM location for the resident MCU. Accordingly, the platform-board write-protect jumper header must enable write protection. Otherwise, any writes to the COP update register change the value stored in user pseudo ROM.

3.4 LOGIC ANALYZER CONNECTOR P1

Figure 3-1 shows the pin assignments for logic analyzer connector P1. Table 3-2 lists signal descriptions for this connector.



NC	1	2	GND
NC	3	4	NC
LA11	5	6	GND
LA10	7	8	LA12
LA9	9	10	NC
LA8	11	12	NC
LA7	13	14	NC
LA6	15	16	AD7
LA5	17	18	AD6
LA4	19	20	AD5
LA3	21	22	AD4
LA2	23	24	AD3
LA1	25	26	AD2
LA0	27	28	AD1
LR/W	29	30	AD0
NC	31	32	LIR
NC	33	34	NC
NC	35	36	NC
VCC	37	38	E
RESET	39	40	NC

Figure 3-1. Connector P1 Pin Assignments

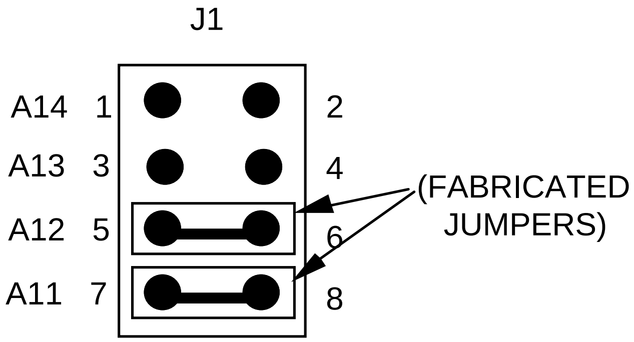
Table 3-2. Logic Analyzer Connector P1 Signal Descriptions

Pin	Mnemonic	Signal
1, 3, 4, 10, 12, 14, 31, 33 — 36, 40	-----	No connection
2, 6	GND	GROUND
5, 7, 9, 11. 13. 15. 17, 19, 21, 23, 25, 27	LA11 — LA0	LATCHED ADDRESSES (bits 11—0) - MCU latched output address bus.
8	LA12	LATCHED ADDRESSES (bit 12) - MCU latched output address bus.
16, 18, 20, 22, 24, 26, 28, 30	AD7 — AD0	DATA BUS (bits 7—0) - MCU multiplexed I/O data bus.
29	LR/W	LATCHED READ/WRITE - Active-high output signal that indicates the direction of data transferred on the bus.
32	LIR	LOAD INSTRUCTION REGISTER - Open-drain, active-low output signal that indicates an instruction is starting.
37	VCC	+5 VDC POWER - Input voltage (+5 Vdc @ 1.0 A) used by EVS logic circuits.
38	E	EXTERNAL CLOCK - Internally generated output clock signal used as a timing reference. The frequency of E clock is 1/2 the input frequency of the signal on the OSC2 pin.
39	RESET	RESET - Active-low bidirectional signal for starting an EVS reset.

3.5 REMAINING SYSTEM INSTALLATION

When you have configured headers J1 and J2, and made sure that the resident MCU is appropriate, you are through with P9EM configuration.

Find jumper header J1 of the HC05EVS platform board. Make sure that two fabricated jumpers are installed, per the diagram below.



To install the P9EM on an HC05EVS platform board, fit together EM connectors P2 and P3 (on the bottom of the board) and platform-board connectors P3 and P4, respectively. Snap the corners of the EM onto the plastic standoffs.

At this point, you are ready to make remaining cable connections, as necessary, and restore power. For instructions, consult the HC05EVS operations manual (HC05EVSOM/D).

INDEX