MOTOROLA

TECHNICAL UPDATE DEVELOPMENT TOOLS

M68HC05P8EVS

Technical Update contains updates to documented information appearing in other Motorola technical documents as well as new information not covered elsewhere.

We are confident that your Motorola product will satisfy your design needs. This Technical Update and the accompanying manuals and reference documentation are designed to be helpful, informative, and easy to use.

Should your application generate a question or a problem not covered in the current documentation, please call your local Motorola distributor or sales office. Technical experts at these locations are eager to help you make the best use of your Motorola product. As appropriate, these experts will coordinate with their counterparts in the factory to answer your questions or solve your problems. To obtain the latest document, call your local Motorola sales office.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters can and do vary in different applications. All operating parameters, including “Typicals” must be validated for each customer application by customer's technical experts. Motorola does not 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 applications intended to support or sustain life, or for any 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. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

TABLE OF CONTENTS

TECHNICAL UPDATE

DEVELOPMENT TOOLS

MC68HC05P8EVS

EVS Revision History

Date	Revision	Description
4/27/95	1.00	Original release. Includes trackers HC705P8EVS.001, HC705P8EVS.002, HC705P8EVS.003, HC705P8EVS.004, and HC705P8EVS.005.

Which EVS Platform and EM Boards Allow Programming of J2s

Tracker Number: HC705P8EVS.001 Revision: 1.00

Reference Document: MC68HC05P8EVS/D1, page 3-7

Platform boards with EVSbug05 Version 1.2 and later versions no longer have the capability to program HC705J2 parts.

The evalutation module boards also have undergone a change. Revision B and earlier EM boards have a programming socket, whereas revision C and later do not have programming sockets.

CHCK, PROG, COPY, and VERF Commands

Reference Document: MC68HC05P8EVS/D1, page 3-33

Tracker Number: HC05P8EVS.002 Revision: 1.00

Customers sometimes assume that the programming voltage doesn’t need to be applied because no EPROM programming exists during a COPY command. However, this assumption is wrong. All checking, programming, copying, and verifying of the EPROM device requires that the appropriate programming voltage be applied to the EPROM device.

These are bootloader commands, so the part must be in bootloader mode to execute the functions. The bootloader mode begins when a rising edge occurs on the $\overline{\text{RESET}}$ pin while the $\overline{\text{IRQ}}/V_{PP}$ pin is at V_{PP}.

HC705J2 Programming Example

Reference Document: M68HC05P8EVS/D1, page 3-32

Tracker Number: HC05P8EVS.003 Revision: 1.00

Listed below are the steps to program an HC705J2 on the HC05P8EVS emulator. A simple, square wave routine will be programmed into the HC705J2. The program outputs a 7.5 µsec, 60 percent duty cycle, square wave on the HC705J2's port A. The program is:

```
        ORG     $700            ;start of the J2's user EPROM array

        LDA     #$FF
        STA     $04             ;make port A all output

LOOP    LDA     #$00
        STA     $00             ;store '0' out to port A
        LDA     #$FF
        STA     $00             ;store '1' out to port A
        BRA     LOOP            ;branch back to make an infinite loop

        ORG     $0FFE           ;define the RESET vector
        DW      $0700
```

For this example, the P&E's IASM05 assembler and EVM05 debugger are used. It is assumed the reader has sufficient knowledge of the assembler and the debugger to perform the functions.

NOTE: Unless erasing the JC705J2, the window must be covered with an opaque covering.

1. If your HC705J2 is erased, skip down to No. 16.
2. Make sure the EVS has power. The V_{PP} pin should have 16.5 volts connected to it.
3. Execute the EVM05 software.
4. Make sure switch SW1 (top board) is OFF.

5. Press the MASTER RESET switch SW3 (bottom board).
6. Insert the HC705J2 into programming socket Z1 (top board).
7. Set switch SW1 (top board) to ON.
8. Enter the CHCK command at the debugger prompt.
9. Press 2 to instruct the EVS to check the HC705J2 memory map.
10. EVM05 will display the message BLANK CHECKING.
11. If the part is blank, the prompt character > will appear.
12. If the part is not blank, EVM05 will display PART NOT BLANK.
13. Set switch SW1 (top board) to OFF.
14. Remove the HC705J2 from the Z1 socket.
15. Blank check is now complete. If your part is not blank, erase it with an EPROM eraser.
16. Assemble the square wave program and download it to the EVS.
17. Make sure switch SW1 (top board) is OFF.
18. Press the MASTER RESET switch SW3 (bottom board).
19. Insert the HC705J2 into programming socket Z1 (top board).
20. Set switch SW1 (top board) to ON.
21. Enter the PROG command at the debugger prompt. This will program all of the EPROM cells. If you want to program only one byte, enter PROG F00 F00. For this example, the command will program the MOR byte at location $F00. Refer to page 3-22 of the M68HC05P8EVS manual for more information on the PROG command.
22. After entering the PROG command, enter in No. 2 for the HC705J2 memory map. The message "blank checking" appears as the EPROM cells are checked for erasure. Next, address values appear as the EPROM cells are being programmed. After this, the verifying message is displayed to show that each cell is being verified. Once the programming sequence is over, the prompt (>) is given to indicate that the PROG command is complete.
23. Set the programmer switch SW1 (top board) to OFF.
24. Remove the HC705J2 from the Z1 socket.

To see the results of the square wave routine that was programmed, test the output of port A by constructing a simple circuit with the HC705J2 on a protoboard.

HC705J2 as an HC05J1 Programming Example

Reference Document: M68HC05P8EVS/D1, page 3-32

Tracker Number: HC05P8EVS.004 Revision: 1.00

Listed below are the steps to program an HC705J2 as an HC05J1 on the HC05P8EVS emulator. A simple, square wave routine will be programmed into the HC705J2. The program outputs a 7.5 µsec, 60 percent duty cycle, square wave on the HC705J2's port A. The routine is the same as the one in tracker No. 3 for the HC05P8EVS. The code is originated for the HC05J1 address map. Refer to Figure 9-1 on page 9-2 of the HC705J2 Technical Data book, document order number MC68HC705J2/D, for the HC05J1 memory map. The program is:

```
        ORG     $300            ;start of the J1's user memory array

        LDA     #$FF
        STA     $04             ;make port A all output

LOOP    LDA     #$00
        STA     $00             ;store '0' out to port A
        LDA     #$FF
        STA     $00             ;store '1' out to port A
        BRA     LOOP            ;branch back to make an infinite loop

        ORG     $700
        DB      $04             ;set the J1 emulation bit
                                ; in the MOR byte of the J1
                                ; memory map

        ORG     $07FE           ;define the RESET vector
        DW      $0300
```

For this example, the P&E’s IASM05 assembler and the EVM05 debugger are used. It is assumed the reader has sufficient assembler and debugger knowledge to perform the functions.

NOTE: Unless erasing the HC705J2, the window must be covered with an opaque covering.

1. If the HC705J2 is erased, skip to No.16.
2. Make sure the EVS has power. The V_{PP} pin should have 16.5 volts connected to it.
3. Execute the EVM05 software.
4. Make sure switch SW1 (top board) is OFF.

5. Press the MASTER RESET switch SW3 (bottom board).
6. Insert the HC705J2 into programming socket Z1 (top board).
7. Set switch SW1 (top board) to ON.
8. Enter the CHCK command at the debugger prompt.
9. Press 2 to instruct the EVS to check the HC705J2 memory map.
10. EVM05 will display the message BLANK CHECKING.
11. If the part is blank, the prompt character > will appear.
12. If the part is not blank, EVM05 will display PART NOT BLANK.
13. Set switch SW1 (top board) to OFF.
14. Remove the HC705J2 from the Z1 socket.
15. Blank check is now completed. If the part is not blank, erase it with an EPROM eraser.
16. Assemble the square wave program and download it to the EVS.
17. Make sure switch SW1 (top board) is OFF.
18. Press the MASTER RESET switch SW3 (bottom board).
19. Insert the HC705J2 into programming socket Z1 (top board).
20. Set switch SW1 (top board) to ON.
21. Enter the PROG command at the debugger prompt. This will program all the EPROM cells. If you only want to program one byte, enter PROG 700 700. For this example, the command will program the MOR byte at location $700. Refer to page 3-22 of the M68HC05P8EVS manual for more information on the PROG command.
22. After entering the PROG command, enter in No.1 for the HC705J1 memory map. The message "blank checking" appears as the EPROM cells are checked for erasure. Next, address values appear as the EPROM cells are being programmed. After this, the verifying message is displayed to indicate that each cell is being verified. Once the programming sequence is over, the prompt (>) indicates that the PROG command is over.
23. Set the programmer switch SW1 (top board) to OFF.
24. Remove the HC705J1 from the Z1 socket.

To see the results of the square wave routine that was programmed, test the output of port A by constructing a simple circuit with the HC705J1 on a protoboard.

How to Program an HC705J2 as an HC05J1

Reference Document: M68HC05P8EVS/D1, page 3-32

Tracker Number: HC05P8EVS.005 Revision: 1.00

The MC68HC705J2 has the ability to emulate the MC68HC05J1. This mode is enabled by programming the J1 bit (bit 2) of the mask option register (MOR) to a logic one. In this mode, the memory map of the MC68HC705J2 is identical to that of the MC68HC05J1 with two exceptions: the MOR is present at $0700 and $07F1-$07F7 is EPROM instead of unused. The map is detailed in Figure 9-1 on page 9-2 of Technical Data book, document order number MC68HC705J2/D.

Because programming the J1 bit changes the memory map and moves the MOR, several questions arise concerning where to place code and where to program the MOR. For example:

- Where do I put my code?
- Do I need to program the MOR?
- Where do I program the MOR?
- How do I program the MOR?
- What do I need to do special on the programmer?

These questions will be answered for each of the three programming devices now available from Motorola: M68HC705J2PGMR, M68HC05P8EVS, and M68CDS8HC05.

M68HC705J2PGMR

To program an MC68HC705J2 as an MC68HC05J1 on the PGMR board, use the MC68HC05J1 emulation mode memory map on page 9-2 of the data book.

The J1 bit of the MOR must be programmed. If the MOR is not programmed, the part will power up in normal J2 mode when it is placed in the target application, and the code will be in the wrong place in the memory map.

The programming must be executed at address $0700, as shown in the emulation memory map.

Because the MOR is an EPROM cell instead of a RAM register, it must be programmed using a define byte similar to that shown below. This may be placed in the user code and programmed at the same time as the user code.

```
ORG   $0700
 DB    %00000100;program J1 bit only
```

A load/store operation will function on the EVS during emulation, but not on a programmed part.

SW1 on S3 of the PGMR must be turned on (closed). This connects PB5 to V_{DD} as required on page 9-1 of the databook. Pulling this pin low will tell the bootloader program inside the MCU to enter J1 emulation memory mode, even though the J1 bit of MOR is not yet programmed.

M68HC05P8EVS

To program an MC68HC705J2 as an MC68HC05J1 on the PGMR board, use the MC68HC05J1 emulation mode memory map on page 9-2 of MC68HC705J2/D.

The J1 bit of MOR must be programmed. If MOR is not programmed, the part will power up in normal J2 mode when it is placed in the target application, and the code will be in the wrong place in the memory map.

Programming must be executed at address $0700, as shown in the emulation memory map.

Because the MOR is an EPROM cell instead of a RAM register, it must be programmed using a define **???** byte similar to that shown below. This may be placed in the user code and programmed at the same time as the user code.

```
     ORG   $0700
      DB    #04;%00000100 - program J1 bit only
```

A load/store operation will function on the EVS during emulation, but will not function on a programmed part.

After the PROG command is issued, enter the code for the J1 MCU device type. The EVS will pull PB5 low, as required on page 9-1 of the databook. Pulling this pin low will tell the bootloader program inside the MCU to enter J1 emulation memory mode even though the J1 bit of MOR is not yet programmed.

M68CDS8HC05

The CDS8 is not able to program the address space of the MC68HC05J1. For this reason, the CDS8 cannot program the J2 as a J1.