

MC68HC908AZ60 FLASH/EEPROM Programmer

User's Guide

TABLE OF CONTENTS

1. PREFACE	3
2. CHECKLIST AND REQUIREMENTS	3
3. INSTALLATION AND USE	4
3.1 PROGRAMMER CHECK AND CONNECTION TO PC.....	5
3.2 PROGRAMMER FIRMWARE UPDATE.....	5
4 INTERFACE TYPES AND OPERATIONAL MODES	5
4.1 IN-CIRCUIT PROGRAMMING.....	6
4.2 ON-BOARD PROGRAMMING	6
4.3 MON08 MODE	6
4.4 AUTO SS MODE	6
4.5 ENTER SS MODE	7
5. WORKING WITH TARGET MCU	7
5.1 CUSTOMARY DEVICES.....	7
5.1.1 In-Circuit EEPROM reading example	7
5.1.2 In-Circuit EEPROM programming example	8
5.1.3 On-Board EEPROM reading example	8
5.1.4 On-Board EEPROM programming example	9
5.1.5 On-Board FLASH reading, programming	10
5.1.5.1 Unsecure device using Auto SS mode	10
5.1.5.2 Unsecure device using Enter SS mode	10
5.1.5.3 Unsecure device using FLASH Erase in Enter SS mode.....	11
5.2 ROM DEVICES.....	11
5.3 MC68HC908AZ60 4J74Y MASKSET	12
5.4 MC68HC08AZ60 1J35D MASKSET	12
5.5 MC68HC08AZ32A 1L52H MASKSET.....	12
5.6 MC68HC05H12 0H57A MASKSET	12
5.6.1 Working with EEPROM Options Register	13
6. EEPROM STRUCTURE	14
6.1 EEPROM BLOCK 1.....	14
6.2 EEPROM BLOCK 2.....	14
7. FLASH STRUCTURE.....	15
8. FILE OPERATIONS	16
8.1 LOAD FILE INTO BUFFER	16
8.2 SAVE FILE FROM BUFFER	16
9. ERRORS AND TROUBLESHOOTING	17
10. WARRANTY STATEMENT AND DISCLAIMER.....	18
11. APPENDIX.....	19

1. PREFACE

This manual will guide you through the installation and operation of the ETL MC68HC908AZ60 Programmer, referenced hereafter as the **HC08-Programmer**.

The **HC08-Programmer** has been designed for Reading, Programming of FLASH, ROM, EEPROM contents of the next Motorola Microcontroller Unit (MCU):

- | | |
|--------------------------|-------------------------|
| ✓ MC68HC908AZ60 (2J74Y) | ✓ MC68HC05H12 (0H57A) |
| ✓ MC68HC908AZ60 (4J74Y) | ✓ MC68HC08AB16A (0L72A) |
| ✓ XC68HC908AZ60 (8H62A)* | |
| ✓ MC68HC908AZ60A (3K85K) | |
| ✓ MC68HC908AS60 (3J74Y) | |
| ✓ MC68HC908AS60A (1L87J) | |
| ✓ MC68HC908AB32 (3K56G) | |
| ✓ MC68HC08AZ60 (1J35D) | |
| ✓ MC68HC08AZ32 (0J66D) | |
| ✓ MC68HC08AZ32 (1H56A) | |
| ✓ MC68HC08AS20 (0H94K) | |
| ✓ MC68HC08AZ32A (1L52H) | |



Note: Most number of devices can be programmed in two operating modes In-Circuit and On-Board. For full information about **HC08-Programmer** features see Appendix Table 7.



Note: XC68HC908AZ60 device memory allocation and monitor ROM similar to MC68HC908AS60 (3J74Y).



Note: Devices that not mentioned above in list can't be guaranteed of correct reading, programming by **HC08-Programmer**.



Note: For In-Circuit programming MON08 Rev1 adapter must be used (see Appendix).



2. CHECKLIST AND REQUIREMENTS

The following describes what items are supplied with the **HC08-Programmer** and the system requirements if used by a PC.

- ✓ **HC08-Programmer** – *supplied*
- ✓ Two QFP64 Adaptors – *supplied*
- ✓ One QFP100 Adaptor – *supplied*
- ✓ One PLCC52 Adaptor – *supplied*
- ✓ One MON08 Rev1 Adaptor – *supplied*
- ✓ Cable -A DB25 “straight-thru” cable - *supplied*
- ✓ **HC08-Programmer** PC software on CD-ROM – *Optional Extra*
- Desktop PC and a free Parallel Port (LPT1, 2)
- Memory - Minimum 32 Mbytes
- Display - Color VGA display recommended
- Preinstalled DWG Viewer 3.0 software*
- Power supply 12 Volt/500 mA linear power supply source
- OS -MS-Windows (Win98, Win2000)



Note: Free DWG Viewer 3.0 software can be downloaded from <http://www.etlweb.com/dwgsoft/dwgviewer.exe>

3. INSTALLATION AND USE

The **HC08-Programmer** includes two two-color LED's and two mode jumper (Figure 1). Color LED's indicates programmer state and external power supply voltage (Table 1, 2).

GREEN	External Power supply OK.
RED	External Power supply voltage or voltage on IRQ pin (2 pin Motorola device socket) lower than 9.5 Volt.
ORANGE	Voltage +5V supplied to Motorola device socket 22, 55 pins.
DARK	Voltage on HC08-Programmer lowers than 6 Volt.

Table 1. LED1 color meaning

GREEN	HC08-Programmer is ready.
GREEN Flashing	HC08-Programmer is busy.
RED	Error occurred while operation.

Table 2. LED2 color meaning

Jumper-1 (JP1) is intended for **HC08-Programmer** mode selection (see Table 3). JP2 and JP3 reserved for future use.



JP1 	Normal operation. Motorola device EEPROM/FLASH Reading/Programming.
	Service mode. HC08-Programmer firmware update.

Table 3. **HC08-Programmer** operation modes



Note: When Service mode selected LED1 and LED2 will be **RED**.

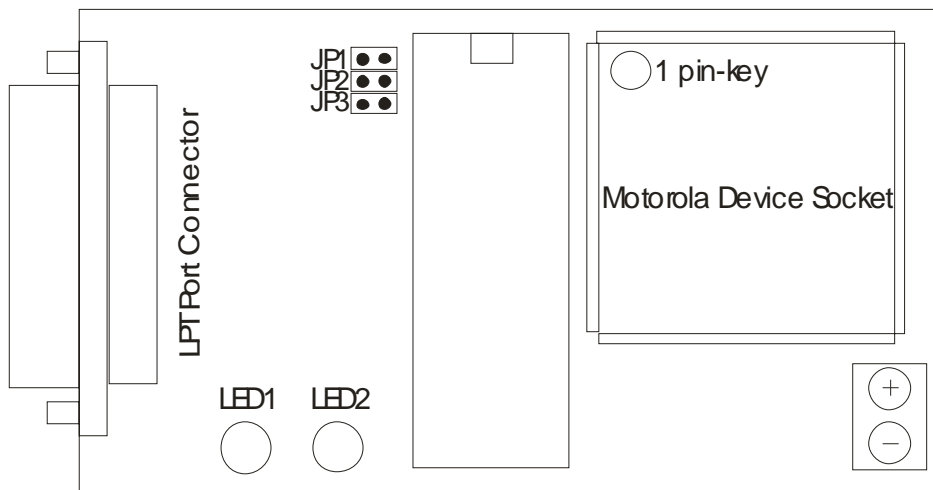


Figure 1. The **HC08-Programmer** board layout

3.1 PROGRAMMER CHECK AND CONNECTION TO PC

- Connect the power supply source to **HC08-Programmer** (an external 12 V DC power supply source is required).
- Attach a LPT cable to the 25-pin connector on the programmer and to a LPT port on the PC.
- Insert jumper JP1 (see Figure 1).
- Remove any adaptor from Motorola device socket.
- Turn On power supply source and make sure that LED1 appear green. LED1 meaning condition see Table 1.
- Start **HC08-Programmer** software.
- After few seconds you should see on display message: “MC68HC908AZ60 Programmer Ver-X.X detected”.
- In case when you should see message: “MC68HC908AZ60 Programmer Ver-X.X not found”, change LPT port base address and speed settings. To do that select “Tools” menu item and than select “LPT Port Options”. Set new LPT port options. After pressing “OK” button the new setting will be applied and software reattempts connection with programmer.
- Now **HC08-Programmer** ready to operate.



Note: Update **HC08-Programmer** firmware if required (see Section 3.2).

3.2 PROGRAMMER FIRMWARE UPDATE

This section describes how to update firmware (ATMEGA16(3)) of **HC08-Programmer**.

- Remove JP1 (see Table 3).
- Turn On power supply; make sure that LED1 and LED2 appear red.
- Start **HC08-Programmer** software.
- Compare firmware version “MC68HC908AZ60 Programmer **Ver-X.X** detected” and version specified in “Help>About” menu item.
- If versions not coincide, follow next steps to update firmware.
- Select menu item “Tools>Firmware Update”.
- After message box “Firmware Update” appeared, press “OK” button.
- After firmware update completed, turn Off power supply, close **HC08-Programmer** software.
- Insert JP1 (see Table 3).
- Turn On power supply. Make sure that LED1 appear green.
- Start **HC08-Programmer** software on PC. New firmware version “MC68HC908AZ60 Programmer **Ver-X.X** detected” will appear.

4 INTERFACE TYPES AND OPERATIONAL MODES

This section describes two interface type and three operational modes of **HC08-Programmer**. In-Circuit Programming and On-Board Programming interfaces are available in **HC08-Programmer**. MON08, Security Sequence Auto Search and Enter Security Sequence modes are available.

4.1 IN-CIRCUIT PROGRAMMING

Generally, In-Circuit programming interface useful for “fast” access to target MCU EEPROM resources. In this mode [HC08-Programmer](#) automatically detects target MCU bus speed. Ceramic resonator connected to target MCU must be in range from 1.6 MHz to 16 MHz. Also pins PTC0, PTC1, PTC3 must be correct configured to achieve required MCU bus speed (see Appendix Figure3). If the ceramic resonator frequency low than 1.6 MHz, use clock overriding from [HC08-Programmer](#). On the CNT2 connector MON08 Rev1 adapter, OSC1 pin support external 8MHz clock frequency to target MCU. Connect this pin to target MCU OSC1 pin. For circuit stability any components on MCU OSC1, OSC2 pins such as capacitors, resistors and crystal resonator must be disconnected.



Note: For In-Circuit programming MON08 Rev1 adapter must be used (see Appendix).



Note: Devices MC68HC908AZ60 (4J74Y), MC68HC08AZ60 (1J35D) has modified monitor ROM: On-board programming available only!



Note: Device MC68HC08AZ32A (1L52H) has extended security possibilities, so knowledge of Security Sequence required.

4.2 ON-BOARD PROGRAMMING

On board programming interface type is basic for [HC08-Programmer](#). In this interface type all programmer features are available. For example, FLASH reading, programming possible only with this interface type. Target MCU must be mounted to the corresponding adaptor QFP64, QFP100 or PLCC52 supplied with [HC08-Programmer](#). Regarding of operation mode external 800 KHz or 8MHz frequency supplied to MCU. Also, for Security Sequence passing specific power control realized.



Note: When target adaptor with MCU mounted check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!

4.3 MON08 MODE

This mode uses standard Motorola 08 Family MCU interface (see Appendix Figure 3). To enter this mode, defined conditions must be established. “High” voltage (10V) on IRQ pin, “High” level on PTC0 pin and “Low” levels on PTC1, PTC3. The PTA0 pin use for communication with host PC. Because of slow speed communication via PTA0 pin, [HC08-Programmer](#) realizes EEPROM reading, programming only. If In-Circuit programming interface selected, before passing Security Sequence [HC08-Programmer](#) automatically detects target MCU communication speed. After speed founded, standard MON08 protocol is performed. If On-Board programming interface selected, communication speed fixed according device type, and MON08 protocol is performed immediately. Also if MCU don’t have additional SCI interface pins (TxD, RxD) or these pins are damaged, MON08 mode is only one way to establish connection with MCU.

4.4 AUTO SS MODE

This mode is main difference from other programmers for 08 Motorola family devices. In this mode [HC08-Programmer](#) automatically search the Security Sequence and passing it. Than, if required, Security Sequence automatically stored in “SSLIST.txt” file. New, found SS than can be used in future in Enter SS mode (see Section 4.5). This mode is useful for reading FLASH contents

of MCU. Also on some devices such as MC68HC08AZ32A (1L52H) even EEPROM contents couldn't be read without knowledge of Security Sequence.

Because of using additional SCI interface pins (TxD, RxD) reading, writing of MCU performed on high speed.



Note: Security Sequence Auto search based on analyzing of electrical current noise on Motorola device power pins. So any equipment with high level of electromagnetic radiation must be switched off.



Note: Auto SS mode implemented for educational purposes only.

4.5 ENTER SS MODE

This mode implemented for “full” working with all FLASH, ROM, EEPROM areas of MCU. Only entering of right Security Sequence will allow user “full” access to MCU resources. Security Sequence located from \$FFF6 to \$FFFD address in memory space of MCU. Every time when host PC trying to establish connection with MCU correct Security Sequence must be entered. For work facilitation with Security Sequences [HC08-Programmer](#) automatically analyze all operations with \$FFF6-\$FFFD areas and if required, store new Security Sequence in “SSList.txt” file. This file can be found in the same directory as [HC08-Programmer](#) software. To load SS from file press “LoadSS” button, select required Security Sequence and press “OK” button. Also, User can see is MCU secured or unsecured in [HC08-Programmer](#) log window.

Because of using additional SCI interface pins (TxD, RxD) reading, writing of MCU performed on high speed.

5. WORKING WITH TARGET MCU

This section contains overall information about Motorola MCUs supported by [HC08-Programmer](#). Also, some device specific behavior described.

When [HC08-Programmer](#) successfully installed (see Section 3) devices (see Table 7) can be read, program and verify in corresponding mode.

5.1 CUSTOMARY DEVICES

This section describes basic rules working with next devices:

- ✓ MC68HC908AZ60 (2J74Y)
- ✓ MC68HC908AZ60A (3K85K)
- ✓ MC68HC908AS60 (3J74Y)
- ✓ MC68HC908AB32 (3K56G)

All devices can be read, program and verify in all [HC08-Programmer](#) modes (see Section 4).

5.1.1 In-Circuit EEPROM reading example

- Mount MON08 Rev1 adaptor into [HC08-Programmer](#) socket.
- Connect required pins to board with target MCU (see Appendix Figure 3).
- Apply power to [HC08-Programmer](#). If LED1 red (see Table 1) disconnect any components from IRQ pin on MCU.
- Check external +12 volt supplied to MON08 adaptor. The light green of LED (D4) shown that +12volts is present.
- Select corresponding device in [HC08-Programmer](#) software (“Device” button).
- Select “ICP” and “MON08” mode.

- Select “Read Sequence” panel in [HC08-Programmer](#) software.
- Press “EEPROM” button.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became to orange light. That means that +5 voltage regulator switched on.
- Also LED (D4) on MON08 Rev1 adaptor must become orange. That means that +5 V voltage regulator OK and logic levels on PTC0, PTC1, PTC3 pins OK.
- When LED2 on [HC08-Programmer](#) became permanent green, reading completed. Now target MCU powered off.
- After read sequence successfully completed it is necessarily to save memory dump to file (see Section 8.2).
- If some errors appeared during reading process refer to Section 9.



Note: For In-Circuit programming MON08 Rev1 adaptor must be used (see Appendix).

5.1.2 In-Circuit EEPROM programming example

- Mount MON08 Rev1 adaptor into [HC08-Programmer](#) socket.
- Connect required pins to board with target MCU (see Appendix Figure 3).
- Apply power to [HC08-Programmer](#). If LED1 appear to red color (see Table 1) disconnect any components from IRQ pin on MCU.
- Check external +12 volt supplied to MON08 adaptor. The light green of LED (D4) shown that +12volts is present.
- Select corresponding device in [HC08-Programmer](#) software (“Device” Button).
- Select “ICP” and “MON08” mode.
- Load EEPROM data from file (see Section 8.1) or enter data to Hex Editor.
- Select “Program Sequence” panel in [HC08-Programmer](#) software.
- Press “EEPROM” button.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became to orange light. That means that +5 voltage regulator switched on.
- Also LED (D4) on MON08 Rev1 adaptor must become to orange light. That means that +5 V voltage regulator OK and logic levels on PTC0, PTC1, PTC3 pins OK.
- When LED2 on [HC08-Programmer](#) became permanent green, programming completed. Now target MCU powered off.
- If some errors appeared during reading process refer to Section 9.



Note: For In-Circuit programming MON08 Rev1 adaptor must be used (see Appendix).



Note: Extra care must be taken when programming the EEPROM nonvolatile registers (EENVR1, EENVR2, EE1DIV, EE2DIV). See Section 6 for details.



Note: Strongly recommended before EEPROM programming, for a first time, read EEPROM contents and save it to file.

5.1.3 On-Board EEPROM reading example

- Mount target MCU on corresponding QFP64, QFP100 or PLCC52 adaptor.*
- Insert adaptor to Motorola Device Socket on [HC08-Programmer](#) in according to 1 pin-key (see Figure 1).

- Apply power to [HC08-Programmer](#).
- Select device in [HC08-Programmer](#) software (“Device” Button).
- Select “OBP” and “MON08” mode.
- Select “Read Sequence” panel in [HC08-Programmer](#) software.
- Press “EEPROM” button.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became orange light. That means that +5 voltage regulator switched on.
- When LED2 on [HC08-Programmer](#) became permanent green light, reading completed.
- When read sequence successfully completed it is necessarily to save memory dump to file (see Section 8.2).
- If some errors appeared during reading process refer to Section 9.



Note: When target MCU mounted on adaptor check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!



Note: During removing the adaptor with target MCU [HC08-Programmer](#) must be powered off to avoid damaging of programmer and target MCU!

5.1.4 On-Board EEPROM programming example

- Mount target MCU on corresponding QFP64, QFP100 or PLCC52 adaptor.*
- Insert adaptor to Motorola Device Socket on [HC08-Programmer](#) in according to 1 pin-key (see Figure 1).
- Apply power to [HC08-Programmer](#).
- Select device in [HC08-Programmer](#) software (“Device” Button).
- Select “OBP” and “MON08” mode.
- Load EEPROM data from file (see Section 8.1) or enter data to Hex Editor.
- Select “Program Sequence” panel in [HC08-Programmer](#) software.
- Press “EEPROM” button.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became orange light. That means that +5 voltage regulator switched on.
- When LED2 on [HC08-Programmer](#) became permanent green light, programming completed.
- If some errors appeared during reading process refer to Section 8.



Note: When target MCU mounted on adaptor check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!



Note: During removing the adaptor with target MCU [HC08-Programmer](#) must be powered off to avoid damaging of programmer and target MCU!



Note: Extra care must be taken when programming the EEPROM nonvolatile registers (EENVR1, EENVR2, EE1DIV, EE2DIV). See Section 6 for details.



Note: Strongly recommended before EEPROM programming, for a first time, read EEPROM contents and save it to file.

5.1.5 On-Board FLASH reading, programming

To perform this operations device need to be unsecured. Device can be unsecured only at On-Board programming interface. User can unsecure device in three ways:

5.1.5.1 Unsecure device using Auto SS mode

- Mount target MCU on corresponding QFP64, QFP100 or PLCC52 adaptor.*
- Insert adaptor to Motorola Device Socket on [HC08-Programmer](#) in according to 1 pin-key (see Figure 1).
- Apply power to [HC08-Programmer](#).
- Select device in [HC08-Programmer](#) software (“Device” Button).
- Select “OBP” and “ASS” mode.
- Select “Read Sequence” panel in [HC08-Programmer](#) software.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became orange light. That means that +5 voltage regulator switched on.
- Now Security Sequence search operation in progress. Progress bar at the right bottom corner indicates completion percent. Usually Auto SS procedure keeps from 5-35 minutes. If progress bar indicator doesn’t change value for a long time, procedure must be stopped, and reattempted again.
- After procedure successfully completed new Security Sequence will be add to “SSList.txt” file and can be used in Enter SS mode.
- Now FLAH can be read, program in the same manner as EEPROM. See previous sections.



Note: When target MCU mounted on adaptor check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!



Note: During removing the adaptor with target MCU [HC08-Programmer](#) must be powered off to avoid damaging of programmer and target MCU!

5.1.5.2 Unsecure device using Enter SS mode

- Mount target MCU on corresponding QFP64, QFP100 or PLCC52 adaptor.*
- Insert adaptor to Motorola Device Socket on [HC08-Programmer](#) in according to 1 pin-key (see Figure 1)
- Apply power to [HC08-Programmer](#).
- Select device in [HC08-Programmer](#) software (“Device” Button).
- Select “OBP” and “ESS” mode.
- Enter correct Security Sequence or load SS with help of “Load SS” button.
- Select “Read Sequence” panel in [HC08-Programmer](#) software.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became orange light. That means that +5 voltage regulator switched on.
- When LED2 on [HC08-Programmer](#) became permanent green light, operation completed.
- If message “Device Unsecured!” appeared in log Window than FLASH can be read, program in the same manner as EEPROM, see previous sections.

- If “Device Secured!” appeared in log Window than FLASH reading, programming will return unexpected results.



Note: When target MCU mounted on adaptor check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!



Note: During removing the adaptor with target MCU [HC08-Programmer](#) must be powered off to avoid damaging of programmer and target MCU!



Note: For “new”, erased devices eight Security Sequence bytes equal to FLASH erase state (see Table 5).

5.1.5.3 Unsecure device using FLASH Erase in Enter SS mode

WARNING: ALL FLASH CONTENTS WILL BE LOST!

- Mount target MCU on corresponding QFP64, QFP100 or PLCC52 adaptor.*
- Insert adaptor to Motorola Device Socket on [HC08-Programmer](#) in according to 1 pin-key (see Figure 1).
- Apply power to [HC08-Programmer](#).
- Select device in [HC08-Programmer](#) software (“Device” Button).
- Select “OBP” and “ESS” mode.
- Enter correct eight values of Security Sequence. These values equal to FLASH Erase State (see Table 5). For Example, for MC68HC908AZ60 this values equal to \$00.
- Select “Program Sequence” panel in [HC08-Programmer](#) software.
- Push “Erase FLASH” button.
- Press “Start” button.
- After several seconds LED1 on [HC08-Programmer](#) became orange light. That means that +5 voltage regulator switched on.
- Now FLASH area erased.
- Turn Off and than turn On power on [HC08-Programmer](#). This operation will reset previous security state of the device.
- Select “Read Sequence” panel in [HC08-Programmer](#) software.
- Press “Start” button.
- When LED2 on [HC08-Programmer](#) became permanent green light, operation completed.
- Message “Device Unsecured!” will appear in log window.
- Now FLASH can be read, program in the same manner as EEPROM. See previous sections.



Note: After target MCU mounted on adaptor check contacts careful to avoid short circuit. Otherwise [HC08-Programmer](#) voltage regulator can be damaged!



Note: During removing the adaptor with target MCU [HC08-Programmer](#) must be powered off to avoid damaging of programmer and target MCU!

5.2 ROM DEVICES

This section describes basic rules working with next devices:

- ✓ MC68HC08AZ32 (0J66D)
- ✓ MC68HC08AZ32 (1H56A)
- ✓ MC68HC08AS20 (0H94K)
- ✓ MC68HC08AB16A (0L72A)

Program memory of these devices formed while manufacturing of MCU by Motorola. So any attempts to reprogram ROM section will be failed. Only EEPROM reading, programming operation is available for this type of devices. Working with EEPROM area not differs from work with customary devices (see Section 5.1).

5.3 MC68HC908AZ60 4J74Y MASKSET

This section describes the algorithm to work with MC68HC908AZ60 device, 4J74Y maskset.

The device has non standard security scheme. As against to 08 family devices, security byte placed in EEPROM1 area in address \$0800. If value of this byte equal \$D3, the device permanently secured, without possibilities to unlocking. To break through this security only On-Board programming interface and Auto SS mode can be used. After [HC08-Programmer](#) successfully finish unlocking procedure value of security EEPROM cell can be changed to \$FF. After next power on reset device became unsecured.

Working with EEPROM, FLASH areas not differs from work with customary devices (see Section 5.1).

5.4 MC68HC08AZ60 1J35D MASKSET

This section describes the algorithm to work with MC68HC08AZ60 device, 1J35D maskset.

Device has non standard security scheme. To break through this security only On-Board programming interface and Auto SS mode can be used. After [HC08-Programmer](#) successfully finished User can work with EEPROM without any restrictions. Once device powered off new Auto SS procedure must be completed.

Working with EEPROM area not differs from work with customary devices (see Section 5.1).

5.5 MC68HC08AZ32A 1L52H MASKSET

This section describes the algorithm to work with MC68HC08AZ32 device, 1L52H maskset.

In this device EEPROM and ROM locked for reading, programming until device secured. Because the device ROM based it can be unsecured only if right Security Sequence entered. To find Security Sequence bytes On-Board programming and Auto SS mode must be used for first time. Than after right SS found, EEPROM can be read, program as On-Board as well as In-Circuit. See Section 5.1.5.1 for details.

5.6 MC68HC05H12 0H57A MASKSET

This section describes the algorithm to work with MC68HC05H12 device, 0H57A maskset.

EEPROM reading, programming operation is available for this device. Also, In-Circuit programming interface is used for MC68HC05H12 device.

Because of device don't have MON08 interface, emulation of programming interface implemented. Conformity between MON08 pins and MC68HC05H12 pins drawn in Figure 2. On the figure MON08 pins denoted by red color. [HC08-Programmer](#) has no possibilities of bus speed auto detection for the device. Ceramic resonator connected to device must be **4 MHz** or **4.19 MHz**.



Note: Do not use PLCC52 adaptor for this device. Only MON08 Rev1 adaptor can be used for In-Circuit programming.

EEPROM located form \$0400 to \$04FF.

EEPROM Options Register (EEOPR) at \$0400 address.

5.6.1 Working with EEPROM Options Register

Device has EEPROM Options Register (EEOPR) which contain EEPRT bit. This bit locks EEPROM area from \$0480 to \$04FF from erasing, programming when cleared.

To unlock area:

- Read EEPROM (On Read Sequence panel push “Read EEPROM” button than “Start” button).
- Save EEPROM contents to file (see Section 8.2).
- Set EEPROM cell \$0400 to \$FF in Hex Editor.
- Complete write to EEPROM (On “Program Sequence” panel push “Program EEPROM” button than “Start” button).
- On next power On EEPROM area from \$0480 to \$04FF will be unlocked.

To lock area:

- Load or enter desired data to EEPROM Area (see Section 8.1).
- Complete write to EEPROM (On “Program Sequence” panel push “Program EEPROM” button than “Start” button)
- Set EEPROM cell \$0400 to \$FD in Hex Editor.
- Complete write to EEPROM.
- On next power On EEPROM area from \$0480 to \$04FF will be locked for programming.

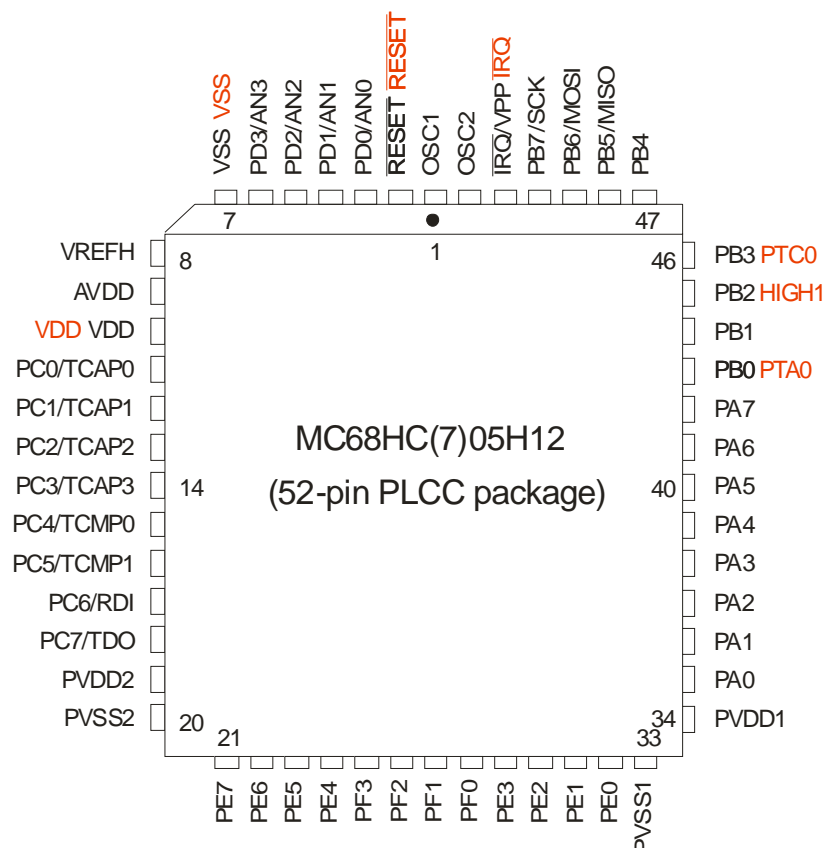


Figure 2. MC68HC05H12 pin assignments

6. EEPROM STRUCTURE

All Motorola devices 08 family has same EEPROM structure shortly described below.

6.1 EEPROM BLOCK 1

EEPROM1 - 512 bytes located from \$0800 to \$09FF.

EENVR1 - Nonvolatile Register (Block Protect Options and EEPROM1, 2 Security)*

EE1DIVH - Nonvolatile Divider Register High* (Not present on some devices see Table 4)

EE1DIVL - Nonvolatile Divider Register Low* (Not present on some devices see Table 4)

Extra care must be taken when programming EEPRTCT bit (4 bit) in EENVR1.

Once the EEPRTCT bit is programmed to 0 for the first time:

- Programming and erasing of secured locations \$08F0 to \$08FF is permanently disabled.
- Programming and erasing of EE1NVR is permanently disabled.
- Bulk and Block Erase operations are disabled for the unprotected locations \$0800-\$08EF, \$0900-\$09FF.

Extra care must be taken when programming EEDIVSECD bit (7 bit) in EE1DIVH.

Once EEDIVSECD in the EE1DIVH is programmed to 0 and after a system reset, the E1DIV security feature is permanently enabled because the EEDIVSECD bit in the EE1DIVH is always loaded with 0 thereafter. Once this security feature is armed, erase and program moderate disabled for EE1DIVH and EE1DIVL. Modifications to the EE1DIVH and EE1DIVL registers are also disabled.

6.2 EEPROM BLOCK 2

This EEPROM area not present on some devices (see Table 5).

EEPROM2 - 512 bytes located from \$0600 to \$07FF.

EENVR2 - Nonvolatile Register (Block Protect Options and EEPROM1, 2 Security)*

EE2DIVH - Nonvolatile Divider Register High* (Not present on some devices see Table 4)

EE2DIVL - Nonvolatile Divider Register Low* (Not present on some devices see Table 4)

Extra care must be taken when programming EEPRTCT bit (4 bit) in EENVR2.

Once the EEPRTCT bit is programmed to 0 for the first time:

- Programming and erasing of secured locations \$06F0 to \$06FF is permanently disabled.
- Programming and erasing of EE2NVR is permanently disabled.
- Bulk and Block Erase operations are disabled for the unprotected locations \$0600-\$06EF, \$0700-\$07FF.

Extra care must be taken when programming EEDIVSECD bit (7 bit) in EE2DIVH.

Once EEDIVSECD in the EE2DIVH is programmed to 0 and after a system reset, the E1DIV security feature is permanently enabled because the EEDIVSECD bit in the EE2DIVH is always loaded with 0 thereafter. Once this security feature is armed, erase and program moderate disabled for EE2DIVH and EE2DIVL. Modifications to the EE2DIVH and EE2DIVL registers are also disabled.



Note: Full description of registers bits assignment can be downloaded from <http://www.etlweb.com/> in “Documents” section.

Device	EEPROM1	EEPROM2	EENVR1	EENVR2	EE1DIVH	EE1DIVL	EE2DIVH	EE2DIVL
MC68HC908AZ60 (2J74Y)	✓	✓	\$FE1C	\$FE18				
MC68HC908AZ60 (4J74Y)	✓	✓	\$FE1C	\$FE18				
XC68HC908AZ60 (8H62A)	✓	✓	\$FE1C	\$FE18				
MC68HC908AZ60A(3K85K)	✓	✓	\$FE1C	\$FE7C	\$FE10	\$FE11	\$FF70	\$FF71
MC68HC908AS60 (3J74Y)	✓	✓	\$FE1C	\$FE18				
MC68HC908AS60A (1L87J)	✓	✓	\$FE1C	\$FE7C	\$FE10	\$FE11	\$FF70	\$FF71
MC68HC908AB32 (3K56G)	✓		\$FE1C		\$FE10	\$FE11		
MC68HC08AZ60 (1J35D)	✓	✓	\$FE1C	\$FE18				
MC68HC08AZ32 (0J66D)	✓		\$FE1C					
MC68HC08AZ32 (1H56A)	✓		\$FE1C					
MC68HC08AS20 (0H94K)	✓		\$FE1C					
MC68HC08AZ32A (1L52H)	✓		\$FE1C		\$FE10	\$FE11		
MC68HC08AB16A (0L72A)	✓		\$FE1C					

Table 4. 08 family EEPROM structure

7. FLASH STRUCTURE

This section shortly describes the structure of the embedded FLASH memory. FLASH area physically divided into separate blocks. These blocks can be protected from erase, write operations with help of FLASH Block Protect Registers:

FLASH-1 Block Protect Register (FL1BPR)

FLASH-2 Block Protect Register (FL2BPR) – Not present on some MCU (see Table 5).

There are no restrictions of programming of any FLASH area.



Note: Full description of FLASH memory can be downloaded from <http://www.etlweb.com/> in “Documents” section.

Device	FLASH	ROM	FL1BPR	FL2BPR	ERASE STATE
MC68HC908AZ60 (2J74Y)	✓		\$FF80	\$FF81	\$00
MC68HC908AZ60 (4J74Y)	✓		\$FF80	\$FF81	\$00
XC68HC908AZ60 (8H62A)	✓		\$FF80	\$FF81	\$00
MC68HC908AZ60A(3K85K)	✓		\$FF80	\$FF81	\$FF
MC68HC908AS60 (3J74Y)	✓		\$FF80	\$FF81	\$00
MC68HC908AS60 (1L87J)	✓		\$FF80	\$FF81	\$FF
MC68HC908AB32 (3K56G)	✓		\$FF7E		\$FF
MC68HC08AZ60 (1J35D)		✓			
MC68HC08AZ32 (0J66D)		✓			
MC68HC08AZ32 (1H56A)		✓			
MC68HC08AS20 (0H94K)		✓			
MC68HC08AZ32A (1L52H)		✓			
MC68HC08AB16A (0L72A)		✓			

Table 5. 08 family FLASH/ROM structure

8. FILE OPERATIONS

This section describes basic rules of working with files. Memory dump from Hex Editor (Buffer) can be load/save from/to hard disk. Also short descriptions such as project name, MCU type and memory cell assignment can be done for future fast remind. [HC08-Programmer](#) accepts tree types of file formats:

- ✓ BIN (Binary format)
- ✓ S-Record (Motorola S19, S20 Records format)
- ✓ EEf (Extended ETL Format)

8.1 LOAD FILE INTO BUFFER

- Select “File>Open” menu item.
- Press “Browse” button.
- In File Open Dialog window, select file witch need to be open.
- Press “Open” button.
- Than opposite “Auto Format Detected:” text, select correct file format. Note that software try automatically detects file format, but unknown records in file will fail this detection. “Load Entire file” check box must be checked if automatically loading procedures required.
- Than press “OK” button.

Sometimes load data from file to specific buffer allocations required. For Example if required load buffer from \$0400 address from binary file beginning from \$0000 address follow next steps:

- Select “File>Open” menu item.
- Press “Browse” button.
- In File Open Dialog window, select file which need to be open.
- Press “Open” button.
- Than, opposite “Auto Format Detected:” select Binary format.
- Uncheck “Load Entire File” check box.
- In field “Offset Value to Place Data to Buffer:” enter 0x0400.
- Than press “OK” button.
- Now data placed to Hex Editor Buffer from the beginning of 0x0400 address.

If more complicated operations with files required, for example load Hex Editor Buffer from many files “Lowest Address From File To Load”, “Highest Address From File To Load” and “Clear Buffer Before Loading File” options are available.

8.2 SAVE FILE FROM BUFFER

- Select “File>Save” menu item.
- Press “Browse” button.
- Select directory in which file will be saved.
- Type file name, for example “test1”
- Press “Save” button.
- Than select format in which file will be saved*.
- Press “OK” button.



Note: Use EEf Format for future “Load File Into Buffer” automatically processing. Also, only in EEf Format Project Description, Device Name and Memory Cells attributes can be saved.

9. ERRORS AND TROUBLESHOOTING

This section describes most recently encountered problems, errors and fixing solutions.

- **Problem:** LED1 (see Figure 1) Dark.
- **Causes:** This problem can accrue when external power supply connected to HC08-Programmer broken or connected in wrong polarity.
- **Solutions:** Check voltage on [HC08-Programmer](#) power clamps. It must be 12 V +/- 1V.

- **Problem:** LED1 (see Figure 1) Red.
- **Causes:** This problem can accrue when external power supply voltage or voltage on IRQ pin (2 pin Motorola device socket) lower than 9.5 Volt.
- **Solutions:** Remove any adaptors from Motorola device socket. Check voltage on [HC08-Programmer](#) power clamps. It must be 12 V +/- 1V.

- **Problem:** LED (D4) on MON08 Rev1 adaptor Dark.
- **Causes:** This problem can accrue when external power supply connected to MON08 Rev1 adaptor broken or connected in wrong polarity.
- **Solutions:** Check voltage on MON08 Rev1 adaptor power clamps. It must be 12 V +/- 1V.

- **Problem:** LED (D4) on MON08 Rev1 adaptor still Green when LED1 on [HC08-Programmer](#) became Orange.
- **Causes:** This problem can accrue when logical levels on PTC0, PTC1, PTC3 pins not correspond to MON08 monitor requirements. Also if Voltage Regulator placed on MON08 Rev1 adaptor damaged.
- **Solutions:** Disconnect PTC0, PTC1, PTC3 pins from target board. If LED1 on MON08 Rev1 appears Orange on attempting to Link Device, connect pin by pin to MCU to find out which pin need to be released from external components. If after relishing PTC0, PTC1, PTC3 pins LED1 not Orange, possible Voltage Regulator placed on MON08 Rev1 adaptor or Voltage Regulator on [HC08-Programmer](#) are broken.

- **Error Message:** "Reset line must be in 'High' state"
- **Causes:** This message can appear when MCU Reset pin connected to ground thru external components or no oscillation on OSC1, OSC2 pins.
- **Solutions:** Check MCU reset pin resistance relatively to VSS pin, and if it too low checks passive, active components connected. Or check integrity of ceramic resonator and passive components connected to OSC1, OSC2 pins.

- **Error Message:** "PTA0 line must be in 'High' state"
- **Causes:** This message can appear when MCU PTA0 pin connected to ground thru external components.
- **Solutions:** Cut any component from pin.

- **Error Message:** "No Echo from MCU"
- **Causes:** This message can appear when target MCU not connected; logical levels on PTC0, PTC1, PTC3 pins and voltage on IRQ pin not correspond to MON08 monitor requirements.
- **Solutions:** Check MON08 monitor requirements according Appendix Figure 3.

- **Error Message:** "Wrong Echo from MCU"

- **Causes:** This message can appear when wrong Device selected. Communication speed not corresponds to this device.
- **Solutions:** Select correct device type; take attention on MCU maskset.

- **Error Message:** "IRQ line must be in 'High' state"
- **Causes:** This message can appear when MCU IRQ pin connected to ground thru external components.
- **Solutions:** Cut any component from IRQ pin.

- **Error Message:** "Communication Error"
- **Causes:** This message can appear when Communication between **HC08-Programmer** and PC broken.
- **Solutions:** Try to decrease communication speed, or select another ready pin in "Tools>LPT Port Options" dialog window. Also this message can appear when LPT port cable has poor contact with DB-25 connectors.

- **Error Message:** "Monitor firmware Check Sum Error"
- **Causes:** This message can appear after monitor firmware loaded into target MCU with errors.
- **Solutions:** Try to reconnect to target MCU. If this error not disappeared, possible target MCU has damaged RAM.

- **Error Message:** "NO response from MCU, possible RAM, RxD or TxD pins damaged"
- **Causes:** This message can appear after monitor firmware loaded into target MCU and "waiting for echo time out error" encoded. Also additional pins (RxD, TxD) used for On-Board programming disconnected or damaged.
- **Solutions:** Trace contact form RxD, TxD pins to pins on adaptor. Try to reconnect with target MCU. If this error not disappeared, possible target MCU has damaged RAM.

- **Error Message:** "NO response from MCU, possible RAM damaged"
- **Causes:** This message appeared at the same conditions as described above with the exception of only PTA0 pin used for echo reply.
- **Solutions:** Trace contact form PTA0 pin to pins on adaptor. Try to reconnect to target MCU. If this error not disappeared, possible target MCU has damaged RAM.



Note: HC08-Programmer has Log Window which can be stored to file. To perform this operation right clicks on Log Window. Than click on "Save to HC08Programmer.log file" menu item. Now this file can be found in the same directory as HC08-Programmer software. Log File can be send by E-mail to techsupport@etlweb.com for non described problem solution.

10. WARRANTY STATEMENT AND DISCLAIMER

ETL warrants that Product delivered shall conform to applicable. Report any defects for a 45 days period, from the applicable data on invoice.

All ETL's product are intended for lawful service, repair or replacement of various electronic equipment with the laws of the country in which the product is being sold or used.

11. APPENDIX

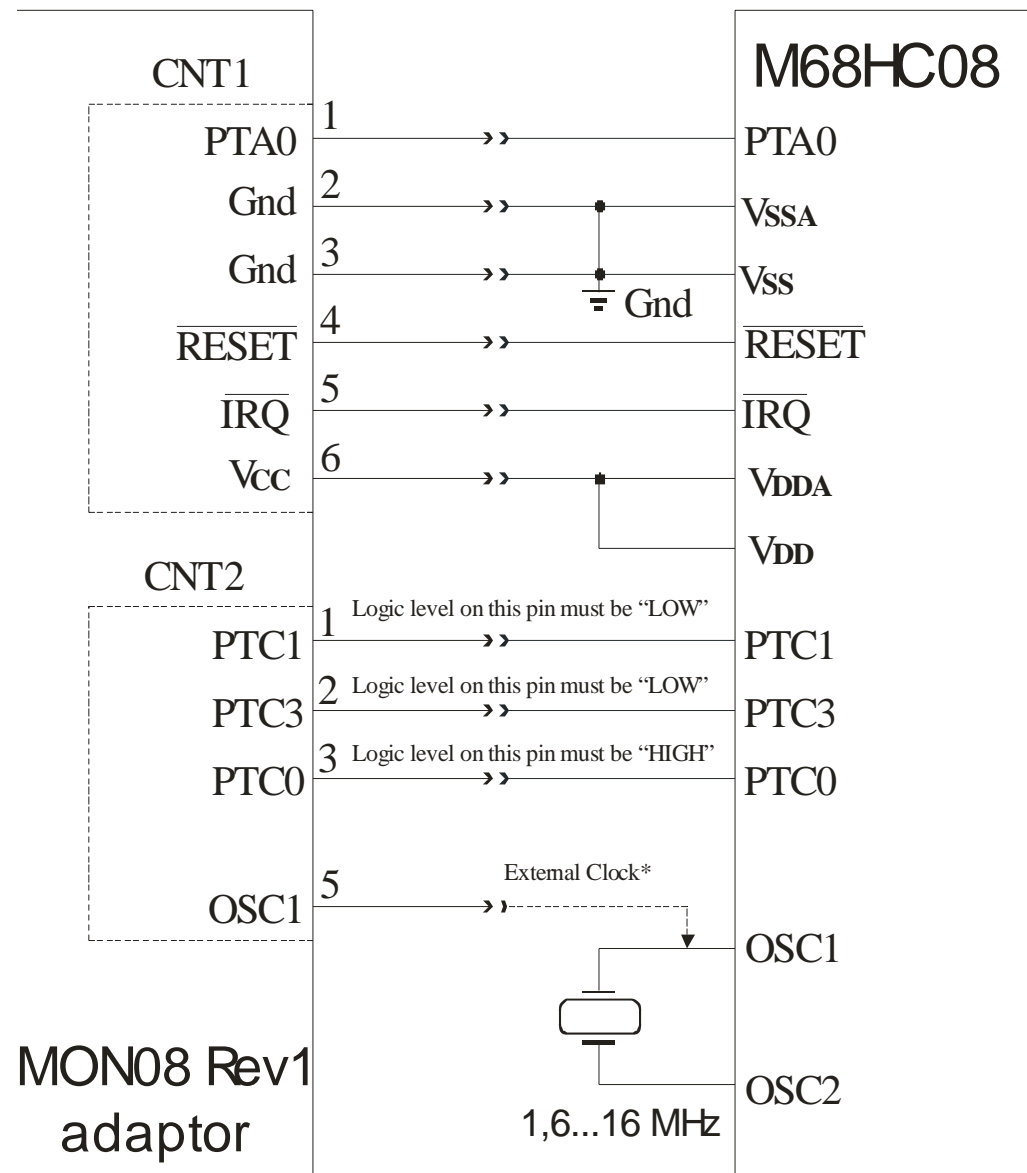


Figure 3. MON08 Circuit



Note: If crystal resonator frequency smaller than 1.6 MHz use External Clock from MON08 Rev1 adaptor. Any component on OSC1, OSC2 such as capacitors, resistors and crystal resonator must be disconnected.

	RST	IRQ	PTA0	PTC0	PTC1	PTC3	Vss	Vssa	Vdd	Vdda	OSC1	OSC2
QFP64	3	2	26	60	61	63	21	56	22	55	59	58
QFP100	3	2	44	96	97	99	30	92	31	91	95	94
PLCC52	10	9	27	4	5	7	25	52	26	51	3	2

Table 6. Pin assignments for various packages

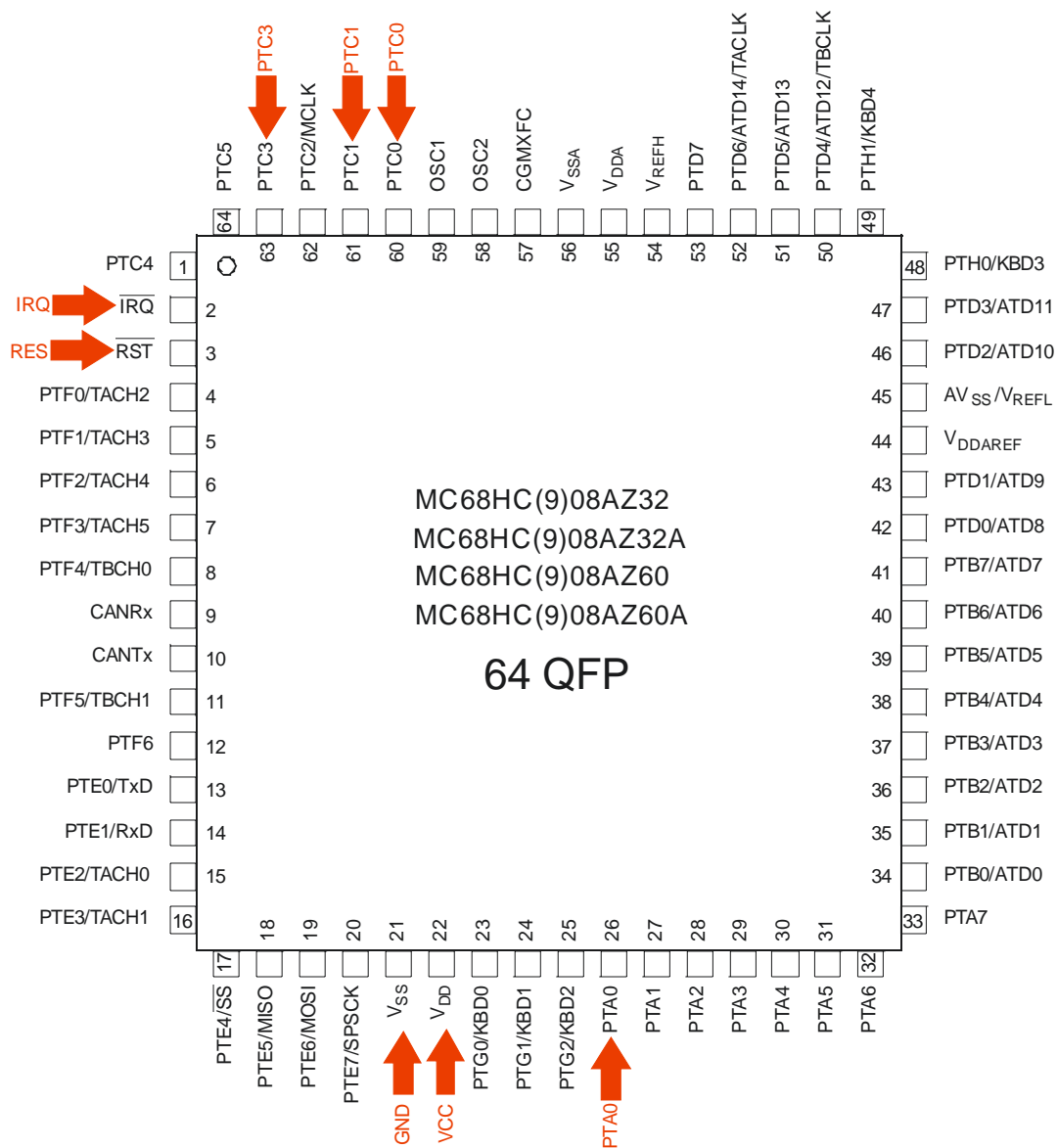


Figure 4. In-Circuit wiring diagram for 68HC(9)08AZxx Series MCU, QFP 64 Package

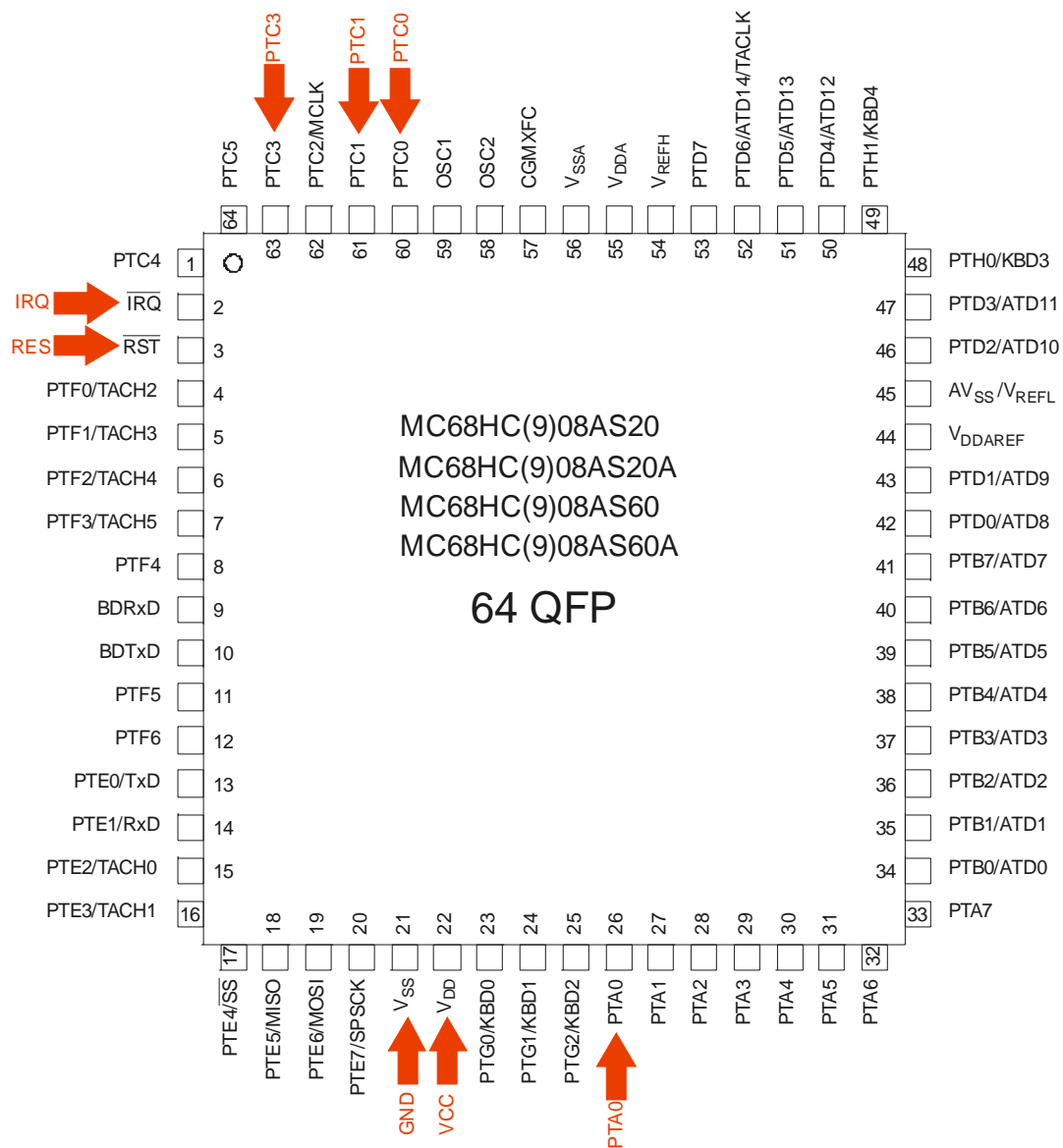


Figure 5. In-Circuit wiring diagram for 68HC(9)08ASxx Series MCU, QFP 64 Package

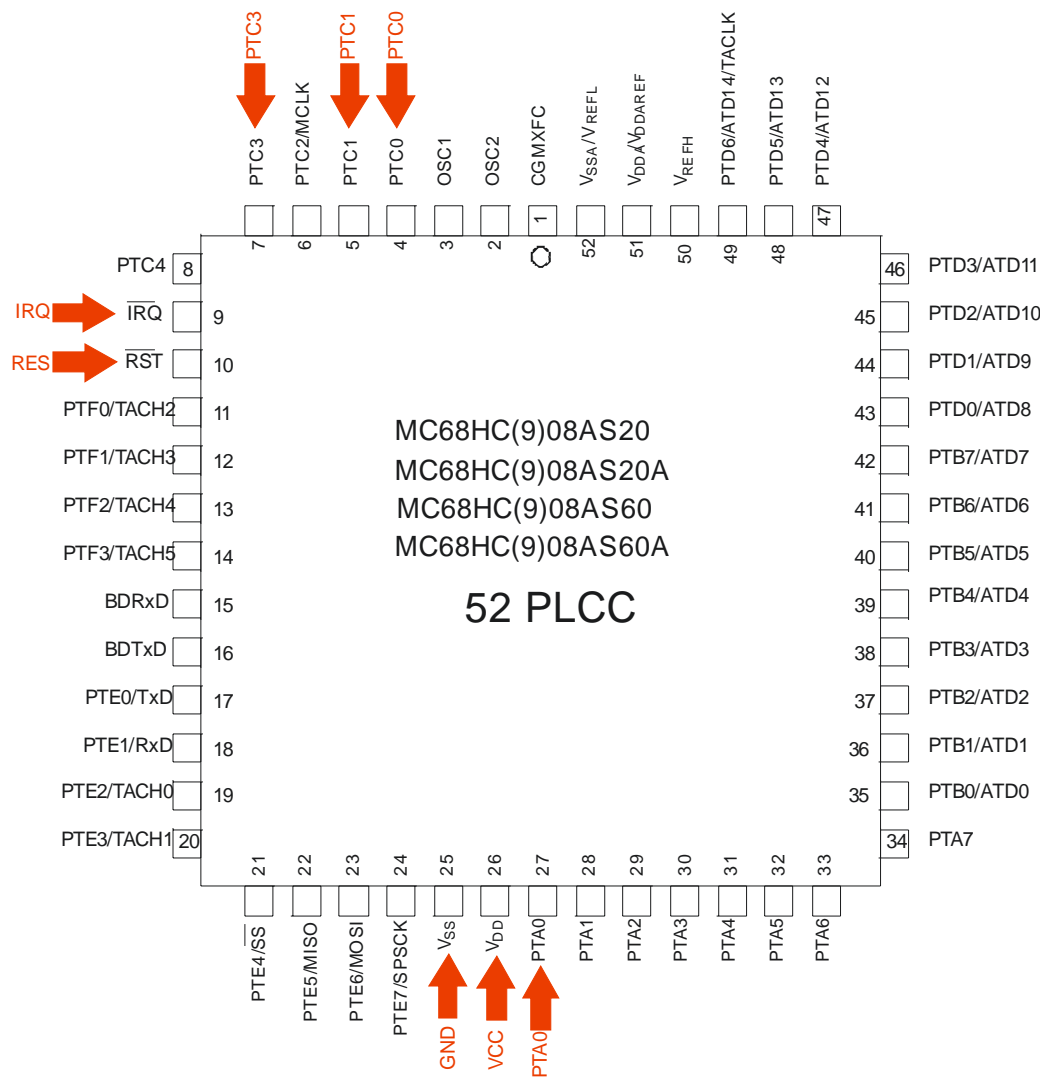


Figure 6. In-Circuit wiring diagram for 68HC(9)08ASxx Series MCU, PLCC 52 Package

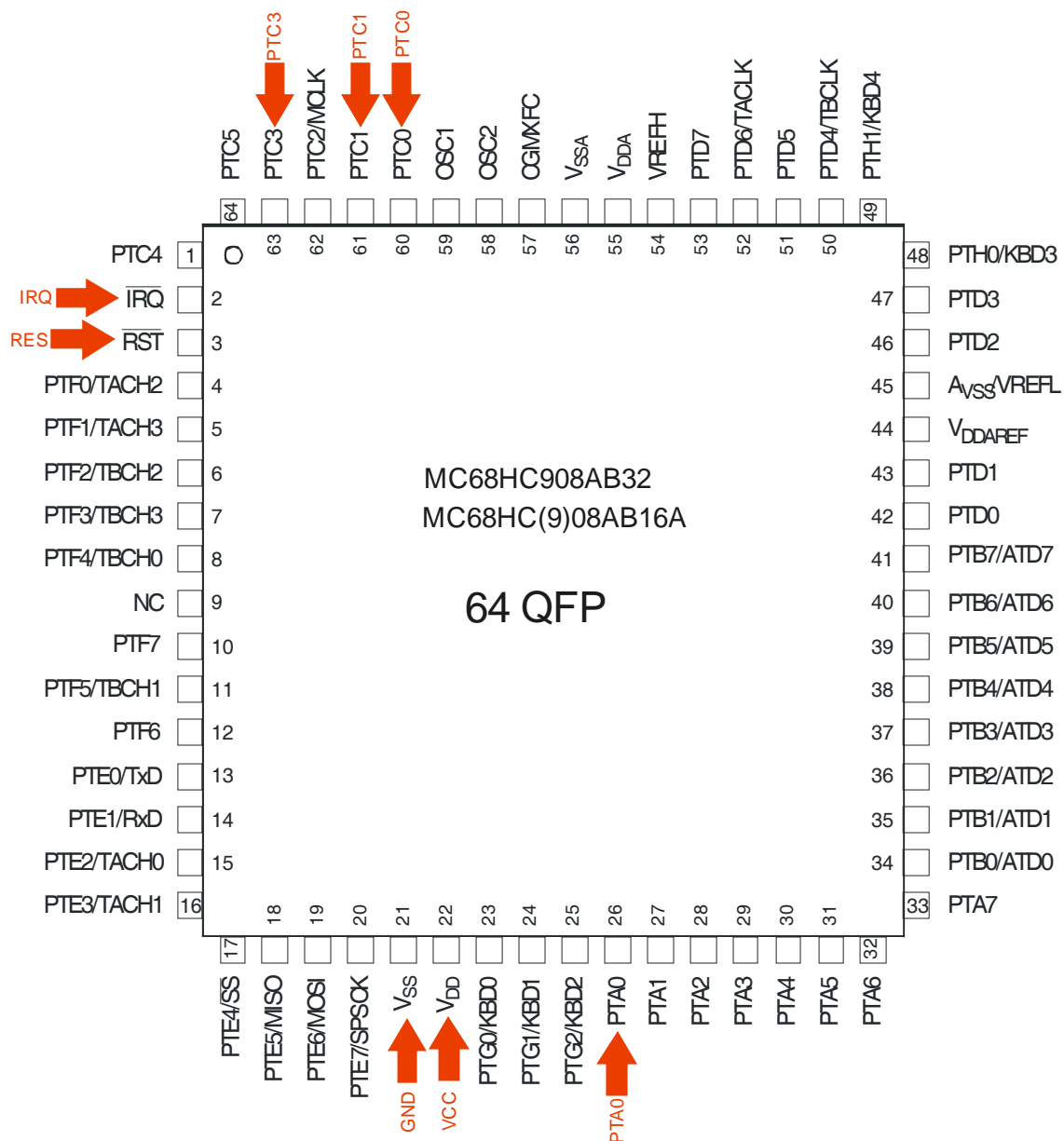


Figure 7. In-Circuit wiring diagram for 68HC(9)08ABxx Series MCU, QFP 64 Package

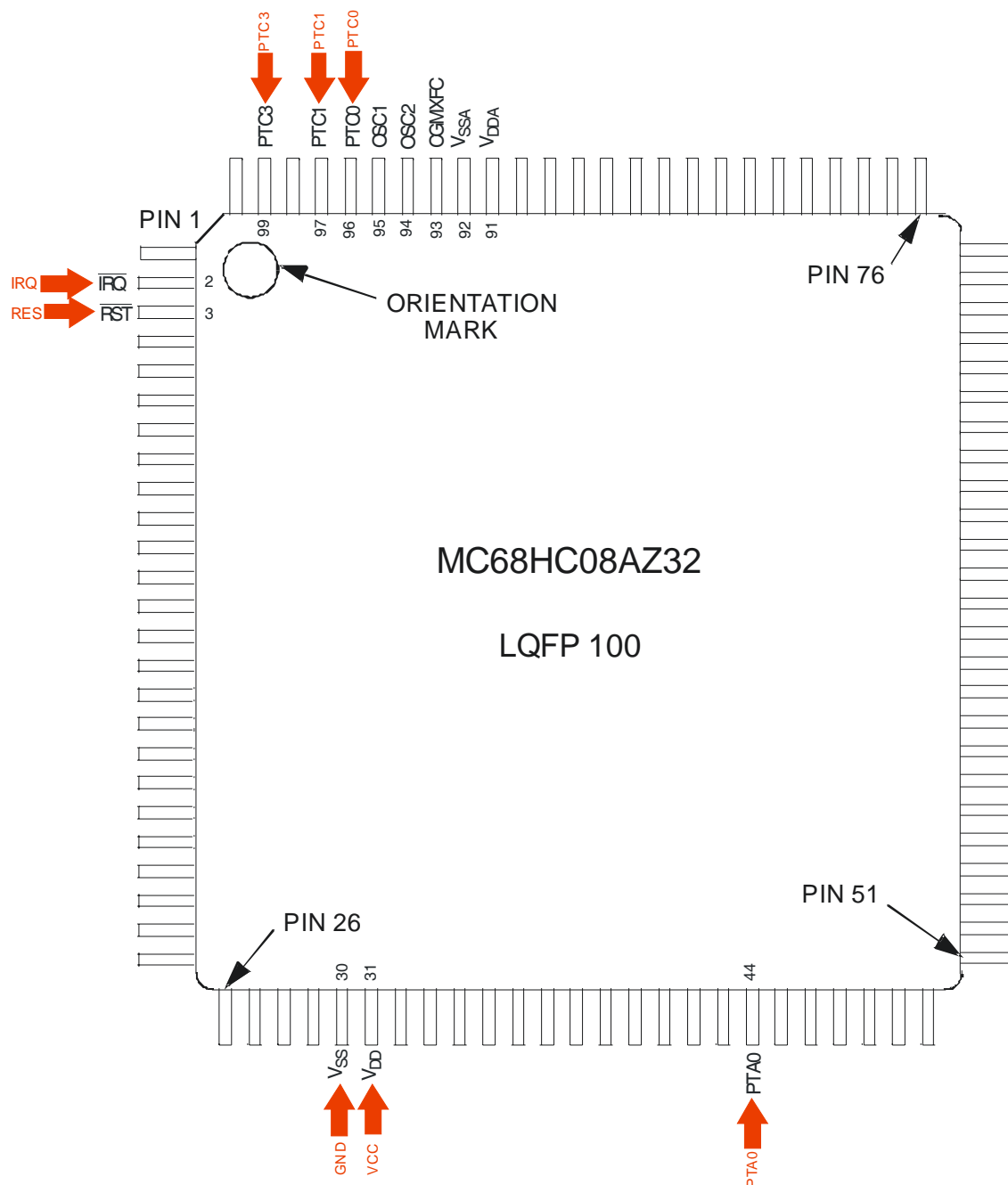


Figure 8. In-Circuit wiring diagram for 68HC08AZ32 MCU, LQFP 100 Package

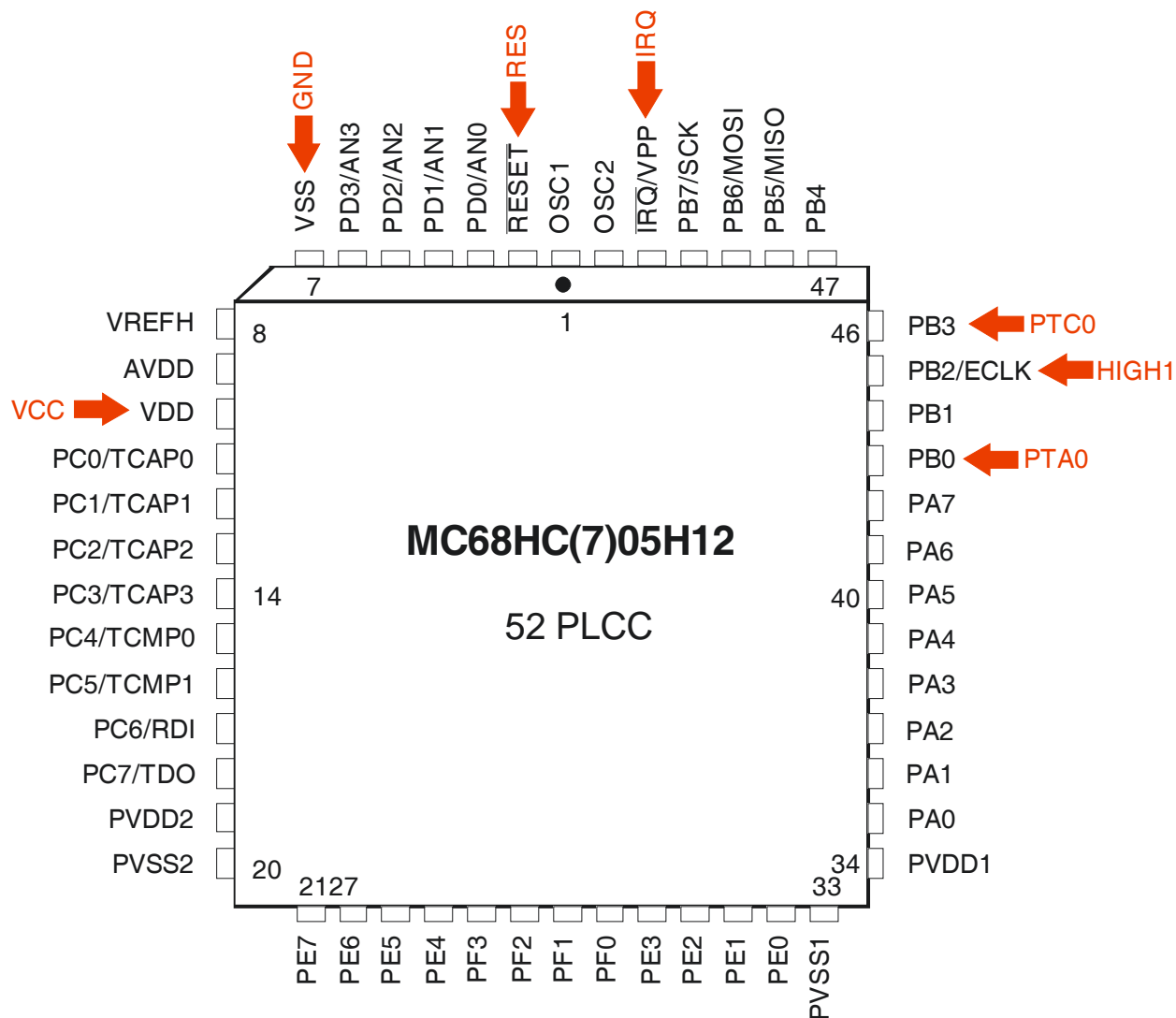


Figure 9. In-Circuit wiring diagram for 68HC(7)05H12 MCU, PLCC 52 Package

Device	EEPROM		FLASH/ROM		On-Board Programming	In-Circuit Programming	MON08 Mode	Auto SS Mode	Enter SS Mode	Feature
	Reading	Programming	Reading	Programming						
MC68HC908AZ60 (2J74Y)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MC68HC908AZ60 (4J74Y)	✓	✓	✓	✓	✓			✓		Non standard security for 08 family in EEPROM area.
XC68HC908AZ60 (8H62A)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MC68HC908AZ60A (3K85K)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MC68HC908AS60 (3J74Y)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MC68HC908AS60A (1L87J)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
MC68HC908AB32 (3K56G)	✓	✓	✓	✓	✓	✓	✓	✓	✓	Extended Security. Additional eight bytes in FLASH.
MC68HC08AZ60 (1J35D)	✓	✓			✓			✓		Extended security.
MC68HC08AZ32 (0J66D)	✓	✓			✓	✓	✓			Security Sequence not implemented.
MC68HC08AZ32 (1H56A)	✓	✓			✓	✓	✓			
MC68HC08AS20 (0H94K)	✓	✓			✓	✓	✓			
MC68HC08AZ32A (1L52H)	✓	✓	✓		✓	✓		✓	✓	
MC68HC08AB16A (0L72A)	✓	✓			✓		✓			
MC68HC05H12 (0H57A)	✓	✓				✓	✓			MON08 mode emulation.

Table 7. Complete supported device list and features

 **Note:** XC68HC908AZ60 device memory allocation and monitor ROM similar to MC68HC908AS60 (3J74Y).

