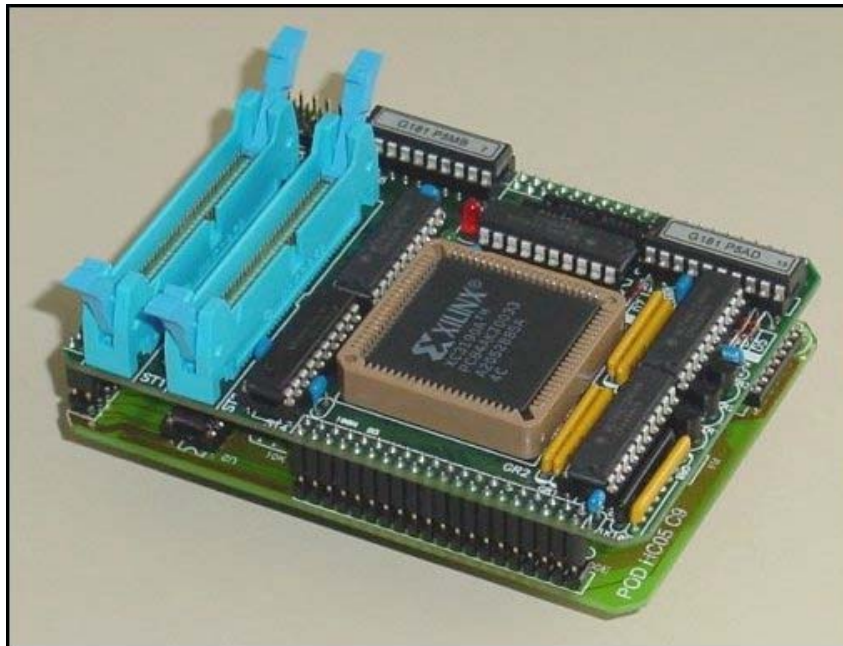

POD Hardware Reference

Motorola 68HC05 C9 POD rev. C

Ordering code	IC81081
---------------	---------



Thank you for purchasing this product from iSYSTEM. This product has been carefully crafted to satisfy your needs. Should any questions arise, do not hesitate to contact your local distributor or iSYSTEM directly. Our technical support personnel will be happy to answer all your technical support questions.

All information, including contact information, is available on our web site www.isystem.com. Feel free also to explore our alternative products.

iSystem constantly yields for development and therefore certain pictures in this documentation may vary slightly from the actual product you received. The differences should be minor, but should you find more serious inconsistencies of the product with the documentation, please contact your local distributor for more information.

This document and all documents accompanying it are copyrighted by iSYSTEM and all rights are reserved. Duplication of these documents is allowed for personal use. For every other case a written consent from iSYSTEM is required.

Copyright © 2002 iSYSTEM, GmbH.
All rights reserved.
All trademarks are property of their respective owners.

POD Hardware Reference

In-Circuit Emulation PODs

The following elements of interest are located on all In-Circuit emulation PODs:

- emulation CPU - acts on behalf of target's CPU. On some PODs you must use the same CPU on the POD as it is used on the target (see your POD reference page). In such cases, remove the CPU from the POD and insert the CPU that you use in the target system, in its place.
- red LED (D3) - lit when CPU is running
- green LED (D4) - lit when Emulator is ready for emulation
- a connector, mostly marked ST3 - contains signal lines, some of which are hardware configuration lines (such as bank select signals), others you can use for signal generation (pattern generator outputs).

Here are some common signals found on the signal connector, commonly marked as ST3:

- GND Ground
- BPE External breakpoint input. Active high.
- RESO/RO Reset output. Connect to target to reset peripherals.
- TRES/TR Target reset input.
- AUXn AUX signal inputs (same as inputs on Emulator/trace)

Note: On PODs that support synchronization between two or more Emulators (currently only the HC(S)12 Family, see the Synchronization section in the Hardware User's Guide for more information) AUX0 and AUX1 are cut short with Run/Stop synchronization line, and AUX2, AUX3 with RESET synchronization line. You should use these pins to connect to other PODs or target CPUs.

- PAT0-2 Pattern generator output on 16-bit POD
- OC4-6 Pattern generator output on 8-bit POD

Note: The signal connector can also have other markings, like P1, U1, etc. Please refer to the POD-specific documentation for the signal connector name and signals present.

For every POD the following information is given:

- Ordering code. If there are different speed versions of a POD the ordering code is modified by appending the speed in MHz (IC81020-16 for the 16 MHz 8031 POD)
- information on available speed versions and required Emulator access time
- POD size and position of PIN1 on the target adapter relative to bottom left corner.

The memory range specifies the range of addresses that a POD can address. If this specification is omitted the default 1MB is assumed.

Note: The In-Circuit Emulator can emulate a processor or a microcontroller. Beside the CPU, additional logic is integrated on the POD. The amount of additional logic depends on the emulated CPU and the type of emulation. A buffer on a data bus is always used (minimal logic) and when rebuilding ports on the POD, maximum logic is used. As soon as a POD is inserted in the target instead of the CPU, electrical and timing characteristics are changed. Different electrical and timing characteristics of used elements on the POD and prolonged lines from the target to the CPU on the POD contribute to different POD characteristics. Consequently, signal cross-talks and reflections occur, capacitance changes, etc.

Beside that, pull-up and pull-down resistors are added to some signals. Pull-up/pull-down resistors are required to define the inactive state of signals like reset and interrupt inputs, while the POD is not connected to the target. Because of this, the POD can operate as standalone without the target.

Final Target Application Test

After the application is being more or less debugged and final application test is performed, it is recommended to remove all breakpoints and to close all debug windows (memory, SFR, watch...) to eliminate any possible influence of the emulator on the CPU execution. There were cases where the target application has been behaving differently with the target CPU inserted or the POD connected. If the debugger is configured to update some debug windows in real-time, the user may not be aware of that the CPU execution may be slightly disturbed. However, when the monitor access type is configured to update debug windows while the CPU is running, the CPU execution is disturbed significantly, depending on the necessary number of memory accesses to update opened debug windows.

There are cases when internal peripheral device requires read access of the particular register during the device configuration. The user has had SFR window opened and the necessary read access was actually performed by the debugger and not by the application as it would be correct. Therefore, the application was working fine with the emulator, but a standalone application didn't work correctly, as the peripheral device was not configured properly.

For Better Understanding of the Hardware Reference: PIN 1 locations

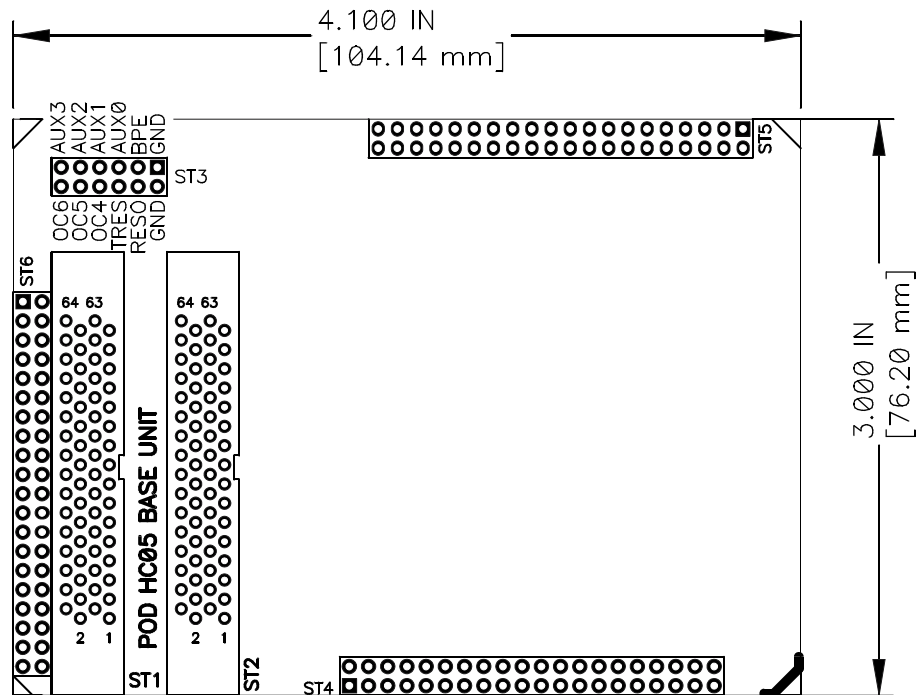
There are several references to pin 1 in the manual and many jumper settings, CPU and pinout orientations rely on the correct location of pin 1. If sometimes the location of pin 1 is not clear, check the markings on the POD. If there are no markings, check the PCB board of the POD. Pin 1 is always marked on the PCB with a square pin (the other pins are round). The pin 1 location is also visible on the board in the hardware reference, if not any other way it can be identified by searching for the square pin.

POD Hardware Reference

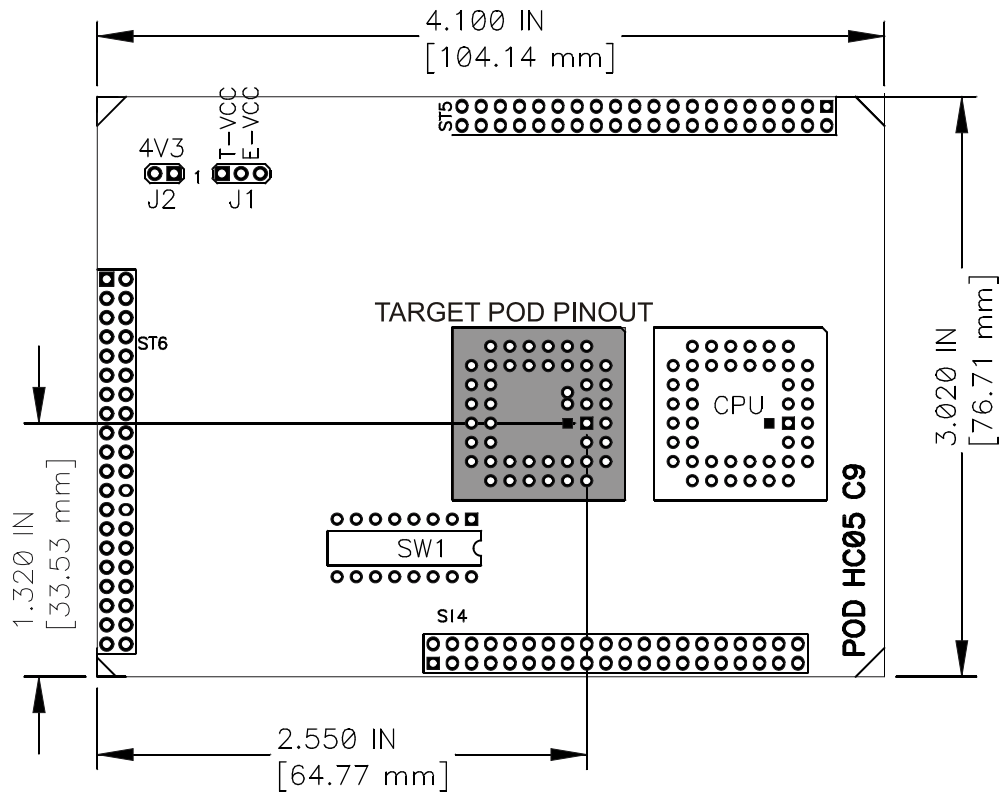
Motorola 68HC05 C9 POD rev. C

Ordering code	IC81081
POD Speed (MHz)	4
Emulator Speed (ns)	90
Exchange CPU	YES

Before connecting the PODs, make sure you have read the technical notes on the Motorola 68HC05 Family in the Hardware User's Guide.



Top Board Layout



Bottom Board Layout

Emulated CPU	CPU on POD
68HC05 C4	68HC05 C8
68HC05 C8	68HC05 C8
68HC05 C8A	68HC05 C8
68HC705 C8	68HC05 C8
68HC705 C8A	68HC05 C8
68HC05 C9	68HC05 C9
68HC05 C9A	68HC05 C9
68HC705 C9	68HC05 C9
68HC05 C12	68HC05 C9

Voltage Selection

Two jumpers on the bottom board of the POD named J1 and J2 determine the operational voltage and its source.

J1 setting	Vcc Source
1-2	Target Source Vcc (T-VCC)
2-3 *	Emulator Source Vcc (E-VCC)

Jumper J1 settings (- factory default)*

J2 setting	Emulator Source Vcc Voltage
Set	4.3 V
Removed	3.5 V

Jumper J14 settings (- factory default)*

Common Emulation Notes

Internal RAM, Internal EEPROM

If the CPU provides a capability to write to the internal RAM or EEPROM via memory window (no specific programming sequence required), the download file can be loaded to the internal RAM or EEPROM using the 'Target Download' option. The debugger downloads the code to the internal memory after reset via the CPU. If the CPU requires some registers to be configured before the CPU is able to write in the EEPROM/RAM area, the user must configure the necessary registers respectively, using the initialization dialog. Any sequence, added in the initialization dialog, is executed immediately after reset, before the download is performed.

Note that debugging is limited while executing the program in the internal EEPROM or RAM. While the CPU accesses internal memory resources, the in-circuit emulator (ICE) loses the control over the CPU since the external bus is not active. Therefore, breakpoints cannot be set and the user's program cannot be stopped or stepped when executing in the internal EEPROM/RAM. Additionally, debug windows cannot be updated as well.

Normally, in the target application the CPU executes the program in the internal or external ROM. Using the ICE, ROM memory is overlaid by the emulation memory and consequently the program can be debugged without restrictions. But, sometimes, there is a need to execute some short routines in the CPU internal memory. Using the ICE, the user can run such routine, but he cannot debug it.

Wait Mode, Stop Mode

Both modes are supported. When the CPU is in one of the mode, it generates (random) dummy cycles, the debugger loses the control over it and HALTED status is displayed in winIDEATM. The CPU can be brought out of the stop mode by the external IRQ and external RESET. Any interrupt or reset will cause the CPU to exit the wait mode.

Checksum

When performing any kind of checksum in the emulated (code) area, note that all breakpoints must be removed before, otherwise the results are distorted. Note that the emulator forces "breakpoint" instruction on the data bus when executing the code at the address where breakpoint is set.

Clock

Clock source can be either used internal from the emulator or external from the target. It is recommended to use the internal clock when possible. When using the clock from the target, it may happen that the emulator cannot initialize any more.

It is dissuaded to use a crystal in the target as a clock source during the emulation. It is recommended that the oscillator is used instead. Normally, a crystal and two capacitors are connected to the CPU's clock inputs in the target application as stated in the CPU datasheets. A length of clock paths is critical and must be taken into consideration when designing the target. During the emulation, the distance between the crystal in the target and the CPU (on the POD) is furthermore increased, therefore the impedance may change in a manner that the crystal doesn't oscillate anymore. In such case, a standalone crystal circuit, oscillating already without the CPU must be built or oscillator used

The Signal Connector

A signal connector is present on the POD, named ST3.

Description	Signal	Pin	Pin	Signal	Description
Ground	GND	1	2	GND	Ground
External Breakpoint	BPE	3	4	RESO	Reset Output
Auxilliary Input	AUX0	5	6	TRES	Target Reset
Auxilliary Input	AUX1	7	8	OC4	Pattern Generator Output
Auxilliary Input	AUX2	9	10	OC5	Pattern Generator Output
Auxilliary Input	AUX3	11	12	OC6	Pattern Generator Output

ST3 signal connector pinout

Target POD Pinout

This PODs support the standard DIP40 pinout.

Notes: