
POD Hardware Reference

MC9S12DP512 ActivePOD II Rev. D

Ordering code	IC30409
---------------	---------



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.

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 © 2006 iSYSTEM, GmbH.
All rights reserved.
All trademarks are property of their respective owners.

POD Hardware Reference

Active POD

Active PODs

Active PODs can be used connected to iC3000(HS) or iC4000 Emulator through the iBUS Active Emulator/Trace interface.

The following elements of interest are located on all Active 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 - lit when CPU is running
- green LED - lit when Emulator is ready for emulation

For every POD following information is given:

- ordering code;
- information on available speed versions and required Emulator access time;
- 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.

PIN1 location on every component is also marked on the circuit board with a square block (PIN 1 pin is soldered a square block, other pins have a round pin).

Trace

The ActivePOD has integrated Trace capabilities that can be utilised if the interface iCard with trace capabilities is used. See the Trace section in the Hardware User's Guide for general information on Trace and POD information for specific information on trace signals (AUX, TrigOut, etc.).

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.

POD Hardware Reference

ActivePOD Emulator Unit II

Ordering code	IC30250
Board dimensions (mm)	98x82



This Emulator Unit is used on some ActivePODs and is used with an ActiveEmulator II iCard.

The following elements of interest are located on the Emulator Unit:

- Green LED LD1 - Emulation - lit when Emulator is ready for emulation
- Red LED LD2 – Running – lit when CPU is running
- LD3 – CPU Power LED – lit when the CPU is powered (either when the POD is not connected to a target or is connected to the target and the target power supply is on)
- LD4 – POD Power LED – lit when the Emulator is powered

The Emulator unit has a Target connector with the following signals:

19	17	15	13	11	9	7	5	3	1
AUX1	AUX3	AUX5	AUX7	AUX9	AUX11	AUX13	AUX15	TRIGOUT	GND
AUX0	AUX2	AUX4	AUX6	AUX8	AUX10	AUX12	AUX14	EXTBP	GND
20	18	16	14	12	10	8	6	4	2

- AUX0 – 15 – available AUX inputs (3V LVTTL, connected to FPGA)

Note: The number of AUX inputs depends of the ActivePOD.

- TrigOut – Trigger or Qualifier output (depending on the setting in the Trigger dialog), active low, when trigger condition occurred (3V LVTTL, connected to FPGA)

The Trigger output is approx. 160µs wide.

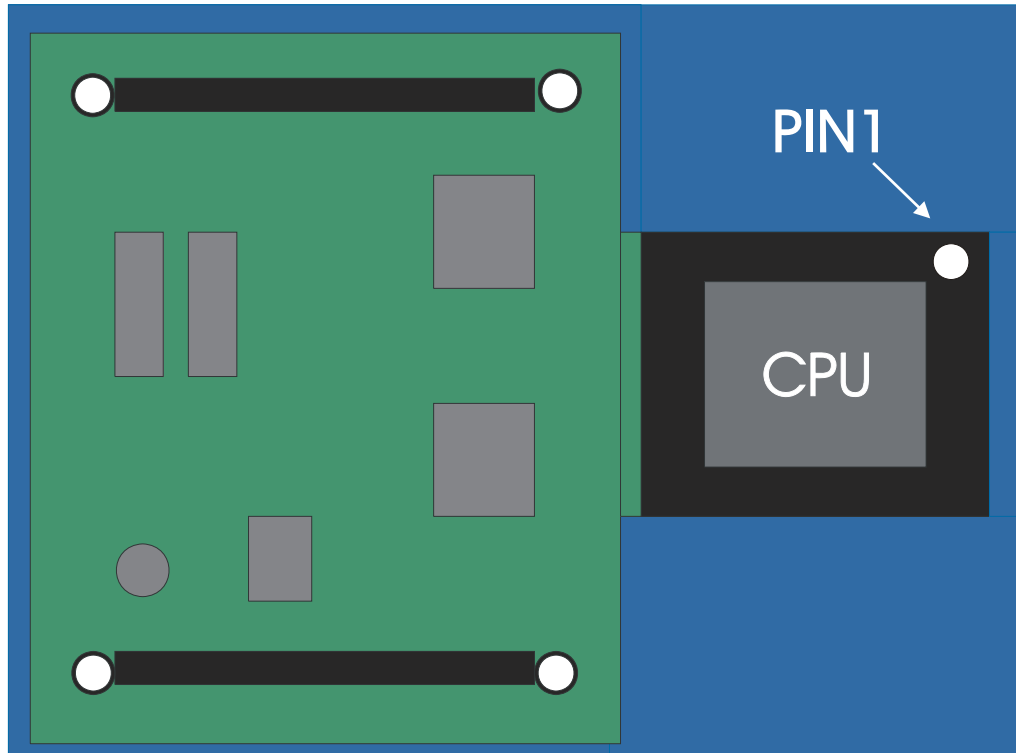
POD Hardware Reference

MC9S12DP512 ActivePOD II Rev. D

Ordering code	IC30409
POD ECLK Speed (MHz)	25
Exchange CPU	YES
Trace Depth	128k
Board dimensions (mm)	99x123
Pin 1 position (mm)	71x118

This POD is connected to the ActivePOD Emulator Unit II.

Before connecting the PODs, make sure you have read the technical notes on Motorola 68HC12 Family in the Hardware User's Guide.



ActivePOD



Jumper locations

Emulated CPUs	
MC9S12A64	MC9S12A128
MC9S12A256	MC9S12B64
MC9S12B128	MC9S12B256
MC9S12C32*	MC9S12C48*
MC9S12C96*	MC9S12C128*
MC9S12D32	MC9S12D64
MC9S12DB32	MC9S12DB128
MC9S12DG128	MC9S12DG256
MC9S12DJ64	MC9S12DJ128
MC9S12DJ256	MC9S12DP256
MC9S12DP512	MC9S12DT128
MC9S12DT256	MC9S12DT512
MC9S12GC16*	MC9S12GC32*
MC9S12GC64*	MC9S12GC128*
MC9S12KG128	MC9S12KT256
MC9S12Q32*	MC9S12Q48*
MC9S12Q64*	MC9S12Q96*
MC9S12Q128*	

Note (*): The MC9S12C128 CPU must be inserted to emulate these devices

Note: The list of emulated CPUs is very dynamic and is only valid at the time of creation of this document. For the latest list of supported devices, please check the iSYSTEM web page.

This ActivePOD is connected to the ActivePOD Emulation Unit II and this unit is connected to the Emulator with an iCARD ActiveEmulator II interface. Do not use the ActiveEmulator I iCard.

Clock Setting

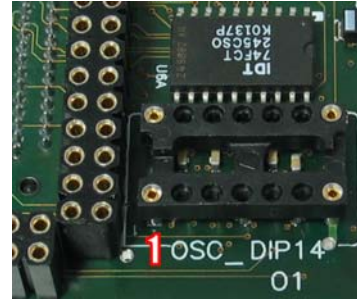
Selecting in winIDEA™, the POD can use either internal or external clock from the target.

When internal clock is selected, the user can choose the clock from the emulator (marked as PLL) or a clock generated by the inserted oscillator (Q1) on the bottom side of the POD (see picture below). A jumper J1 selects internal clock source being used. An operating frequency is set in the winIDEA™, when using a clock from the emulator. When inserted oscillator is used, it must have 5V voltage levels even though the CPU requires 2.5V levels. 5V oscillator output is divided down to necessary 2.5 V by the POD.

J1 setting	Internal clock source
1-2 (*)	PLL (emulator)
2-3	Q1 Oscillator

Jumper J1 settings (- factory default)*

Right: Q1 Pin 1 Location

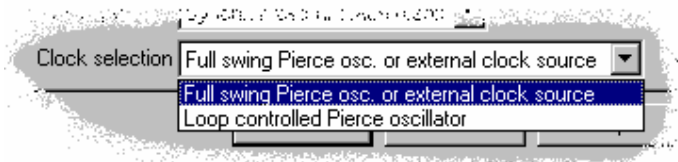


Note that Pin 1 on the J1 jumper is marked with a square.

When external clock is selected, the POD uses a clock from the target. Motorola has defined very precise PCB and connection guidelines when using a crystal as a clock source. Since these guidelines cannot be considered on the POD completely, it's not recommended to use a crystal in the target as a clock source. The distance between the crystal in the target and the CPU on the POD is critical and thus the crystal may not oscillate at all. Therefore, when external clock is being used, it is recommended to use an oscillator in the target to assure initialization and operation of the system. The oscillator used in the target must be 2.5V type, as the CPU requires 2.5V clock.

The CPU has a pin, with which the type of oscillator used is selected during reset. The selection depends of the hardware implementation of the oscillator circuit. This is done by clock selection dialog where either

- Full swing Pierce oscillator or external clock source or
- Loop controlled Pierce oscillator



can be selected. For more information on both types of oscillators, please consult the CPU manual.

Be sure that 'Full swing Pierce oscillator or external clock source' is selected when the clock is set to internal (Emulator) clock.

Vcc Setting

This POD normally operates on 5V. Only the 5V power supply is provided from the emulator. However, internal CPU core operates at 2.5V. This voltage can be generated internally by the CPU, using internal regulator, or external voltage can be used. Which voltage is being used, depends on the state of the CPU's VREGEN pin.

When the POD is not connected to the target, the CPU core uses internally generated voltage.

When the POD is connected to the target, the CPU senses the VREGEN (Voltage Regulator Enabled) pin to determine whether the internal voltage regulator is active and used or the power supply from the target is used. When using external power supply (VCCPLL, VCC1, VCC2), Motorola's guidelines must be considered.

In order to operate the POD at 3.3V, a CPU that also supports 3.3V (for example the MC9S12C128) must be inserted into the POD and the target power supply (which must provide 3.3V) must be selected in winIDEA.

Peripheral Register Location

By default, the peripheral registers (SFRs) and program code can overlap in the memory area between 0x0000 and 0x7FFF. For compatibility purposes, it is also possible to enable the mode in which SFR registers and the code are not able to overlap. This is set using the J14 jumper, located on the bottom of the POD.

J14 setting	Operation
1-2	SFR area set with J12 – no code overlap
2-3 (*)	SFR and code can overlap

Jumper J14 settings (- factory default)*

If J14 is set to 1-2, by default the peripheral registers (SFRs) are restricted to the memory area within 0x0000-0x3FFF and the user's code may reside in the area 0x4000-0x7FFF. The SFRs can also be relocated above 0x4000. For this purpose a J12 jumper is implemented on the bottom side of the POD.

J12 setting	SFR Area
Set (*)	0x0000-0x3FFF
Removed	0x0000-0x7FFF

Jumper J12 settings (- factory default)*

If the jumper is removed, the code cannot be located in the 0x4000-0x7FFF block. Consequentially, make sure the 'ROMHM - FLASH only in second half of memory map' option in the 'Hardware/Emulation Options/CPU Setup/Memory' tab is checked. If J14 is set to 2-3, the J12 setting is ignored.

Note that SFRs cannot be relocated above 0x8000.

AUX inputs

This ActivePOD supports 4 AUX inputs. The connector is available on the Emulation unit. Note that on the Emulation Unit II, AUX0 to AUX3 are active only. Any digital signal connected to the AUX input can be recorded by the analyzer, operating in the trace mode. Trigger on it can be set as well.

Target connection detection

By default, the emulation system auto-detects whether the POD is connected to the target or not and supplies the power to the sensitive POD logic only while the target is switched on. Consequentially, optimum conditions are met since the POD is powered synchronously to the target. In worst case, wrong manual switch on/off order can damage the hardware. Debug download is not possible while the POD is connected to target being switched off.

The auto-detection can be turned off using the J4 jumper.

J4 setting	Auto-detection
Set (*)	Auto-detection enabled
Removed	Auto-detection disabled

Jumper J4 settings (- factory default)*

If the auto-detection is disabled, the emulator doesn't auto detect whether the target is powered or not and supplies the power to the POD logic when the emulation system is switched on. The debug download is possible even while the target is switched off.

Emulating MC9S12C/GC/Qxxx

To emulate these devices, a MC9S12C128 CPU in the QFP112 package must be inserted into the POD. The special CPU required for this can be ordered from iSystem. Please contact your local sales representative for more information.

MC9S12C derivatives have minor differences comparing to other MC9S12 derivatives. When emulating them, pay attention:

- jumper J14 on POD must be set to position 1-2
- SFRs area must be below 0x4000 or 0x8000 depending on J12
- C128 (96,64,32) must be selected in winIDEA

Known CPU design flaw (MC9S12C128, mask 0L09S)

Internal RAM is disabled after reset. To enable it, INITRM register must be written prior any write to the internal RAM.

COP troubleshooting

When using COP, 'Reset from Target enabled' option in winIDEA 'CPU Setup/Options' dialog must be checked. When COP reset is generated, the CPU drives reset line. Any "non emulator" reset is target reset.

Synchronization of Two or More Emulators

When multiple emulators should operate synchronously, synchronization connector available on the side of the emulator must be used. All the emulators that should operate synchronously must have connected together (separately) SR (SYNC-RESET) pins, SS (SYNC-STOP) pins and GND (ground) pins.

2	4	6
GND	S-STOP	S-RESET
GND	S-STOP	S-RESET
1	3	5

Synchronization connector

Two lines for each signal are present to allow easier connection of multiple Emulators.

For more information, please refer to the "Synchronization of Two or More Emulators" section of the Hardware User's Guide.

Electrical Differences and Rebuilt Ports

In general, when emulating the single chip mode, some ports have to be rebuilt on the POD because original ports are used for emulation – typically ports used as address and data bus in extended mode. Special devices, so called port replacement units, provided already by the CPU vendor or other standard integrated circuits are used to rebuilt "lost" ports. Rebuilt ports are logically 100% compatible with original CPU's ports, but electrical characteristics may differ. If special device (port replacement unit) is available, electrical characteristics don't differ much and usually the user doesn't have to pay attention. The differences may become relevant when standard integrated circuits are used and operating close to electrical limits, e.g. when input voltage level is close to specified maximum voltage for low input level ("0") or specified minimum voltage for high input level ("1").

When emulating the single chip mode, original ports A, B, E and K are used for the emulation and rebuilt by standard integrated circuits on the POD therefore electrical characteristics are changed.

Using the emulator, the port registers (Port Registers and Port Data Direction Registers) belonging to the rebuilt ports must be mapped to the target when emulating the single-chip mode.

Whenever operating close to electrical limits and having problems with rebuilt ports please check pull-up and pull-down resistors. They shouldn't be too strong, neither too weak. Check the voltage level. Try to withdraw from voltage limits.

Using COP

The CPU has an internal watchdog, that must be refreshed periodically, or the CPU resets. The COP can be disabled in STOP (while the CPU is stopped by BDM), which is necessary for debugging. Since the register, which turns off COP is a write-once register, the whole register must be written.

Using COP with Star12 PODs

Use COP is a global option with, which the COP usage is selected. If this option is enabled, the option to insert COPCTL is available. If COP is enabled, the software presets the COPCTL register (address 0x003C) with the value, entered into the COPCTL field. It makes sure automatically that RSBCK (the bit, that disables COP while the user program is stopped) is always active.

Target (Watchdog) Reset and Assume Target Reset

If COP is enabled, the 'Target RESET Enabled' function must be enabled. At COP reset the CPU forces a reset, that the emulator only recognizes as target reset.

Both HC12 and Star12 CPUs have three sorts of resets with each having its own individual vector. These are:

- External reset - power on. Vector 0xFFFFE-0xFFFF
- Clock monitor fail reset. Vector 0xFFFFC-0xFFFFD
- COP failure reset. Vector 0xFFFFA-0xFFFFB

In most cases all vectors point to the same location, but this does not have to be so.

To test the situation when the program starts after reset with another vector, the software has three 'Assume Target RESET' options:

- Other (default) (Vector 0xFFFFE-0xFFFF)
- Clock monitor fail (Vector 0xFFFFC-0xFFFFD)
- COP failure (Vector 0xFFFFA-0xFFFFB)

Depending on the selected option, the CPU starts at every target reset on the location, to, which the appropriate vector points.

At every emulator reset (through the software) the CPU always starts on the location, to, which the 0xFFFFE-0xFFFF vector is pointing.

Emulation Mode

The Emulation Mode in which the POD operates can be set in the software. Two operating modes are available and there are limitations in both:

- Emulation Expanded Wide Mode

It is impossible to write to IRQE bit in the INTCR register (CPU flaw)

- Normal Expanded Wide Mode

PEAR register is write any time and thus writable by the user's application. The application misbehaves when ECLK is turned off by the application since the emulator requires ECLK for its operation. The ECLK signal that goes from the emulator to the target is rebuilt on the POD and set according to the settings in the 'CPU Setup/Advanced' dialog (PEAR value). The POD emulates single chip mode (to the target) according to the CPU specification despite that the CPU on the operates in the emulation/expanded mode.

When emulation mode is used, the register is write once register and first write is executed by the emulator, so the application cannot modify it later any more. The user needs to enter the PEAR value (that his application uses) in the 'CPU Setup' dialog.

It is recommended to always use the Emulation Expanded Wide Mode, except when the IRQE bit in the INTCR register is used.

Emulation notes

The CPU on the POD operates in the expanded mode, but it behaves to the target as operating in the single chip mode. Since the CPU on the POD operates in the expanded mode while emulating the single chip mode, some registers, like PEAR and MODE, contain different values than expected by the user. In addition, they cannot be written by the user. PEAR register value used by the application must be written in the debugger's 'Advanced' dialog. PEAR register cannot be written by the application itself.

Internal FLASH

The ROMON bit in the MISC register must not be set to 1. By doing so the internal FLASH would be enabled. The internal FLASH is overlaid with emulator's memory thus internal FLASH must be disabled.

General HC12 Emulation Notes

Clock

There are two major issues the user must pay attention to when using external clock:

- Also when using external clock, the user must specify target clock frequency ($2 * ECLK$) in the 'Hardware/Emulation Options/Vcc/Clock' tab, like in the case when using internal clock. It is required by the debugger to be able to synchronize with on-chip BDM firmware which operates at CPU's system clock frequency.
- It is not recommended 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.

COP

COP is fully supported, unless DP256 mask 0K36N is used. It is not advisable to use this mask of the CPU, since also other problems can arise. If the DP256 with the mask 0K36N must be used, select the 'DP256 Rev I' CPU in winIDEA. In this case, COP is not supported due to CPU specifics. If this mask is used, internal COP must be disabled when using the emulator, respectively while debugging.

When experiencing some problems with the emulator when using COP, please check, whether you have correct/valid interrupt vector table. If some other interrupt vector is defined by mistake as a watchdog interrupt vector, it may look like the COP resets the CPU when the interrupt occurs.

STOP Instruction

STOP instruction is completely supported by the emulator. After the STOP instruction is being executed, the CPU is stopped and the debugger displays HALTED status. Note that the debug windows cannot be updated while HALTED status is displayed. When the CPU is awoken either by interrupt or target reset, the emulation/execution proceeds normally.

Internal CPU Flash

Note that internal FLASH is disabled during the emulation and cannot be used in any way.

Internal RAM, Internal EEPROM

HCS12 derivatives don't allow writing to the internal EEPROM area via single memory access. They require an EEPROM programming algorithm, which is also implemented in winIDEA. The user can download to the EEPROM and modify its content in the memory window.

Considerations:

- 1) After reset, the EEPROM area is on some CPUs (e.g. MC9S12DP256) covered with the I/O area, the priority of which is higher than the priority of EEPROM and thus the EEPROM or portion of it may not be available.
- 2) The CPU has an EEPROM Protection Register (EEPROT). By this register, a certain EEPROM block can be protected. If a certain block is protected, the emulator and the user's program cannot change its content.
- 3) The CPU has an ECLKDIV register with which the EEPROM clock is set. The register contains division factor set according to the CPU specifications (EEPROM clock frequency of 150-200kHz is required). The emulator does not write to this register.

To download a file to the internal EEPROM:

- 1) 'Target Download' must be used;
- 2) The ECLKDIV register must be set using the initialization sequence;
- 3) If the whole EEPROM is required, the EEPROM area must be relocated with the initialization sequence (by writing to the INITEE register). This is necessary because on some CPUs the EEPROM area is covered with the I/O area. Don't forget to set a proper offset for the download file when relocating EEPROM memory area.
- 4) The EEPROM writes must not be disabled.

The run, stop and single step debug commands can be used in the internal RAM or EEPROM. Single step can be performed either in the disassembly or source window. Breakpoints can be set as well.

PLL

The Active PODs support PLL use.

After the PLL is initialized, below equation defines new CPU system clock frequency:

1) after reset PLL is inactive

$Extal = 2 * ECLK$

2) PLL enabled

$ECLK = Extal * (SYNR + 1) / (REFDIV + 1)$

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.

Slow Mode

It is not recommended to debug the application running in slow mode, due to a CPU flaw.

'Clear on read' register bits

Be careful when the CPU has register bits that are cleared on read access. Do note that when such register (memory location) is accessed either by memory/watch window or SFR window, the flags are cleared and the application may behave different when using the emulator or the target CPU. It is recommended not to display such register or the associated memory location in the memory/watch window during final test. Otherwise, it may happen that the target application doesn't work due to the bug in the code even though it works with the emulator.

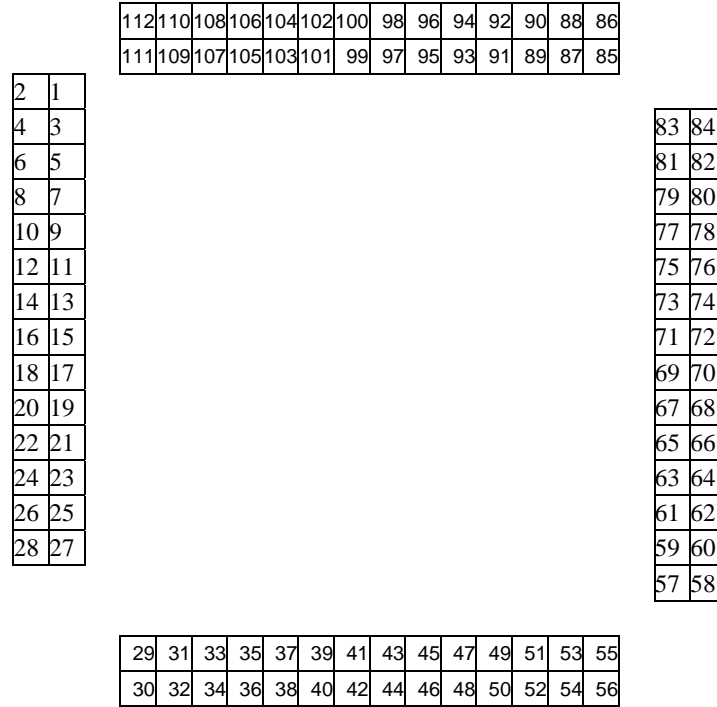
For instance, the user makes a mistake and doesn't clear the flag in the application. Using the emulator, the application works correctly since the user uses SFR window which clears the flag when the window is updated.

Target Adapters

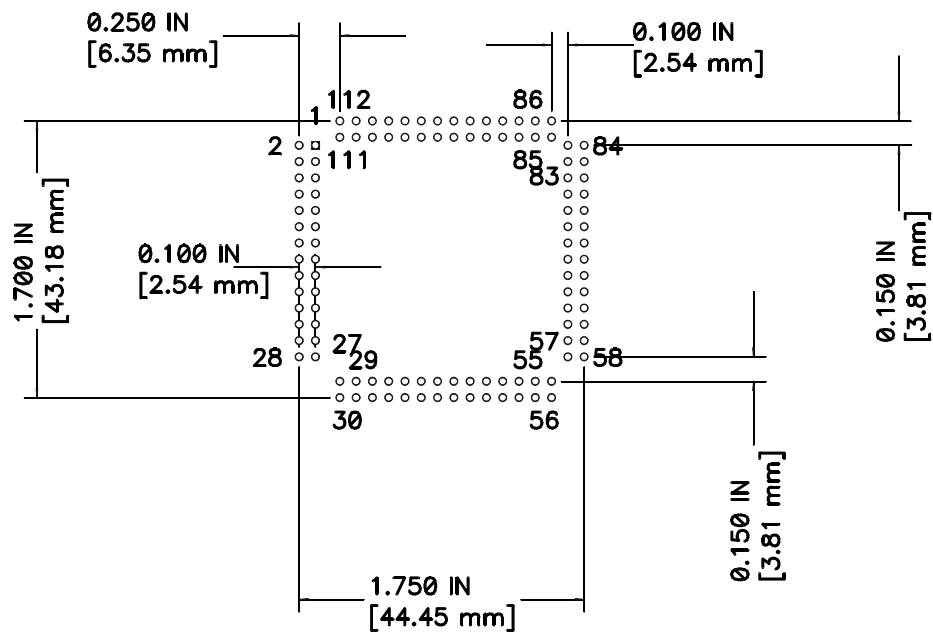
iSYSTEM offers various adapter solutions for this POD. Please refer to the adapter documentation for more details.

POD Target Layout

The POD target layout is T_QFP112.



T_QFP112 – Top POD view



T_QFP112 – Dimension

Notes:

Notes:

Notes: