|  |  |  |
| --- | --- | --- |
| Debug Port Specification - ARCHIVE http://i.msdn.microsoft.com/dynimg/IC435252.gif  Hardware Requirements for Debugging under the Microsoft Windows Operating Systems on a PC without a Legacy Serial Port  **Patent Notice. Microsoft provides you certain patent rights for implementations of this specification under the terms of Microsoft’s Community Promise, available at <http://www.microsoft.com/openspecifications/en/us/programs/community-promise/default.aspx>.**  Version 1.0b  **Updated: March 12, 2014**  **On This Page**  [http://i.msdn.microsoft.com/dynimg/IC79147.gifDebug Port Background](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Debug_Port_Background)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifGeneral Requirements and Approach](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#General_Requirements_and_Approach)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifLong-term Approach](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Long_term_Approach)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifShort-term Approach](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Short_term_Approach)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifSoftware Implementation Requirements](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Software_Implementation_Requirements)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifPhysical Connection Requirements](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Physical_Connection_Requirements)  [http://i.msdn.microsoft.com/dynimg/IC79147.gifDebug Port Table](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#Debug_Port_Table)   |  | | --- | | By using or providing feedback on these materials, you agree to the license agreement.  **DEBUG PORT TABLE SPECIFICATION LICENSE**  **IMPORTANT—READ CAREFULLY: This Microsoft License Agreement ("Agreement") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the version of the Microsoft specification identified above which you are about to download ("Specification"). BY DOWNLOADING, COPYING OR OTHERWISE USING THE SPECIFICATION, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD, COPY, OR USE THE SPECIFICATION.**  The Specification is owned by Microsoft or its suppliers and is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. **THE SPECIFICATION IS LICENSED, NOT SOLD**.  1. **GRANT OF LICENSE**.  (a) Provided that you comply with all terms and conditions of this Agreement, Microsoft grants to you the following non-exclusive, worldwide, royalty-free, perpetual, non-transferable, non-sublicensable, limited license under any copyrights or patents that cover the table values described in the Specification and that are owned or licensable by Microsoft without payment of consideration to third parties,  (i) to reproduce copies of the Specification for your and your contractor’s internal use for the sole purpose of (1) modifying your firmware and/or BIOS for computing devices ("Firmware") so that it writes to memory the appropriate table vales in the Specification or (2) modifying your software so that it may read from memory the appropriate table values (the "Purpose"),  (ii) to implement the table values in your firmware and/or BIOS,  (iii) to license to third parties directly and indirectly the table values as part of your Firmware (and any related documentation),  The foregoing license is granted only to the extent necessary to accomplish the Purpose and to license and/or distribute your Firmware containing the table values to third parties. The foregoing license shall not extend to any features of your Firmware that (i) are not required to comply with the Specification or (ii) to which there was a practicable alternative to infringing a patent.  (b) Microsoft reserves all other rights it may have in the Specification, its implementation and any intellectual property therein. The furnishing of this document does not give you or any other entity any license to any other Microsoft patents, trademarks, copyrights or other intellectual property rights. Microsoft does not grant to you or any other entity any implied licenses or rights whatsoever under this Agreement. Specifically, this Agreement does not grant any express or implied licenses or rights to any enabling technologies that may be necessary to fully utilize the table or the table values described in the Specification.    2. **ADDITIONAL LIMITATIONS AND OBLIGATIONS.**  (a) You must implement the table in its entirety (e.g., all fields) and without modification (e.g., byte length, offset, and permissible values as described in the Specification).  (b) Your license rights to the Specification are conditioned upon you not creating, modify, or distributing your Firmware in a way that such creation, modification, or distribution may (a) create, or purport to create, obligations for Microsoft with respect to the Specification (or intellectual property therein) or (b) grant, or purport to grant, to any third party any rights or immunities to Microsoft’s intellectual property or proprietary rights in the Specification.  (c) The foregoing license is applicable only to the version of the Specification which you are about to download. It does not apply to any additional versions of or extensions to the Specification.  (d) Without prejudice to any other rights, Microsoft may terminate this Agreement if you fail to comply with the terms and conditions of this Agreement. In such event you must destroy all copies of the Specification and must not further distribute the table values.  3. **INTELLECTUAL PROPERTY RIGHTS**. All ownership, title and intellectual property rights in and to the Specification, and any copies you are permitted to make herein, are owned by Microsoft or its suppliers  4. **DISCLAIMER OF WARRANTIES. To the maximum extent permitted by applicable law, Microsoft and its suppliers provide the Specification (and all intellectual property therein) and any (if any) support services related to the Specification ("Support Services") *AS IS AND WITH ALL FAULTS*, and hereby disclaim all warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties or conditions of merchantability, of fitness for a particular purpose, of lack of viruses, of accuracy or completeness of responses, of results, and of lack of negligence or lack of workmanlike effort, all with regard to the Specification, any intellectual property therein and the provision of or failure to provide Support Services. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT, WITH REGARD TO THE SPECIFICATION AND ANY INTELLECTUAL PROPERTY THEREIN. THE ENTIRE RISK AS TO THE QUALITY OF OR ARISING OUT OF USE OR PERFORMANCE OF THE SPECIFICATION, ANY INTELLECTUAL PROPERTY THEREIN, AND SUPPORT SERVICES, IF ANY, REMAINS WITH YOU.**  5. **EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES. To the maximum extent permitted by applicable law, in no event shall Microsoft or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use of or inability to use the SPECIFICATION, ANY INTELLECTUAL PROPERTY THEREIN, the provision of or failure to provide Support Services, or otherwise under or in connection with any provision of this AGREEMENT, even in the event of the fault, tort (including negligence), strict liability, breach of contract or breach of warranty of Microsoft or any supplier, and even if Microsoft or any supplier has been advised of the possibility of such damages.**  6. **LIMITATION OF LIABILITY AND REMEDIES. Notwithstanding any damages that you might incur for any reason whatsoever (including, without limitation, all damages referenced above and all direct or general damages), the entire liability of Microsoft and any of its suppliers under any provision of this Agreement and your exclusive remedy for all of the foregoing shall be limited to the greater of the amount actually paid by you for the Specification or U.S.$5.00. The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.**  7. **APPLICABLE LAW**. If you acquired this Specification in the United States, this Agreement is governed by the laws of the State of Washington. In respect of any dispute which may arise hereunder, you consent to the jurisdiction of the state and federal courts sitting in King County, Washington.  8. **ENTIRE AGREEMENT. This Agreement is the entire agreement between you and Microsoft relating to the Specification supersedes all prior or contemporaneous oral or written communications, proposals and representations with respect to the Specification or any other subject matter covered by this Agreement. To the extent the terms of any Microsoft policies or programs for Support Services conflict with the terms of this Agreement, the terms of this Agreement shall control.** |   To submit comments on this specification, please send e-mail to [debugprt@microsoft.com](mailto:debugprt@microsoft.com). |  |

### Debug Port Background

Legacy COM ports have become a prime source of confusion and frustration for PC users. Over the years, a huge array of devices has appeared that attach to the COM port. Each device uses the COM port and its various signals in proprietary ways that often have difficult timing requirements. This leads to interoperability problems that are confusing to the user.

Further, the COM port "bus" does not have a fully featured Plug and Play standard--making it difficult to detect when a device has been attached, removed, or swapped with another. This also leads to confusing scenarios that are difficult for the user solve.

For example, a user plugs a modem into the COM port. After making a call, that user unplugs the modem and plugs in a personal digital assistant. However, the operating system and modem application do not have any way of knowing that the modem is no longer attached, and therefore prevent the PDA synchronization software from running.

For all of these reasons, legacy-free machines do not include COM ports.

However, operating system vendors, device driver writers, and others still need some mechanism for running a kernel debugger. In order to prevent a continuation of COM port problems, it is undesirable to include a COM port in machines. This specification describes the options available to OEMs to provide the required debug port. These options reflect the assertion that implementation for a debug port must be cheap, easy to manufacture, and standard. Note that these options pertain to the Windows Millennium Edition (Windows Me) and Windows 2000 operating systems, and follow-on products.

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### General Requirements and Approach

Such a device has the following high-level requirements:

* Simple design: cheap, easy to manufacture.
* Simple standard software interface.
* Uses a standard connector and pinout--to eliminate logistical problems of proprietary cables and so that any computer can debug any other computer. In some cases, this connector may use existing standards or may be a new standard.
* Software interface requires only a very small overhead to run it. (This rules out SMI emulations.)
* Portable: the solution can be used on x86-based, Intel Itanium-based, and future platforms.
* Relatively high speed: 57600 baud or better.

In addition, for use in the field and for debugging crashes after the fact, it is highly desirable that a debug cable can be connected and used on the fly without requiring a reboot.

To meet the needs of upcoming product schedules for OEMs, both a long-term and short-term approach must be taken. Any register interface required (called the Debug Port register interface in this document) is discovered via the Debug Port Table in ACPI. The differences between the two approaches are in the external interface and in which component the debug port is implemented.

Of the solutions described in this document, the following are available for use as of January 2000 with legacy-free-capable Windows operating systems:

* IEEE 1394 port (support under Windows 2000 yet to be announced).
* Super I/O based, using rerouted, non-legacy addresses.
* Low-pin count (LPC) based with a debugging dongle.

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### Long-term Approach

Over the long term, the approach is to support debugging using either an IEEE 1394 or USB port.

**IEEE 1394 Port**   
Machines equipped with one or more standard OHCI-compliant IEEE 1394 controllers can utilize the standard IEEE 1394 ports for connecting a host machine with one or more targets (up to 62 on a single bus). The host controller can be on the motherboard or attached via PCI (including CardBus). Standard IEEE 1394 cables can be used for this purpose.

This approach is supported in the Windows Millennium Edition operating system and will be supported in future versions of the Windows 2000 operating system. Plans for support under the original version of Windows 2000 have not been announced.

**USB**   
Work is currently underway within the USB organization to develop a USB debug solution utilizing the USB 2.0 host controller specification. This approach will require a special debug dongle between the host and the target machine.

Microsoft is committed to working on this solution. Support for this solution in particular releases of the Windows operating system will be announced after hardware is available for testing.

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### Short-term Approach

There are two different short-term approaches if the IEEE 1394 solution is not chosen. These options are provided so that they can be implemented as quickly as possible with existing components. There are two options available to manufacturers: Super I/O-based and LPC-based. Note that both of these options result in a DF9 connector replacement for the legacy COM port.

**Super I/O-based**   
In machines where the Super I/O is still on the motherboard (but disabled via the BIOS) and the chip allows it, a COM port can be enabled and rerouted to non-legacy addresses (that is, addresses other than 03F8-03FF, 03E8-03EF, 02F8-02FF, and 02E8-02EF). The standard motherboard header is stuffed with the standard connector common on many motherboards today. This header is compatible with existing ribbon cable header-to-DB9 adapters. The external port is not stuffed.

The serial port connector used for the debug solution on a legacy-free PC must not be accessible to the end user. This can be ensured either by hiding the serial port connector or by providing a connection point for an internal expansion serial cable.

Although this is a quick and efficient solution for most desktops and servers, it is not well-suited for mobile platforms due to limited real estate to place the header, and the platform continues to carry the burden of the serial interface circuitry.

The Windows Me and Windows 2000 operating systems provide support for this solution. Future versions of Windows operating systems will also likely support this solution, although the expectation is that the Super I/O will disappear over time.

**LPC-based**   
The low-pin count (LPC) bus can provide a connection for a debugging dongle. The basic design encompasses using a Super I/O and UART on the dongle with a DB9 port out to provide similar debugging capabilities to the serial port. Both [Intel](http://developer.intel.com/technology/easeofuse/spec.htm)http://i.msdn.microsoft.com/dynimg/IC444318.gifand [AMD](http://www.amd.com/)http://i.msdn.microsoft.com/dynimg/IC444318.gif provide such solutions to the industry.

The Windows Me and Windows 2000 operating systems provide support for this solution. Future versions of Windows operating systems will also likely support this solution, although the expectation is that this is a short-term solution due to the availability of the IEEE 1394 solution and, ultimately, a USB solution.

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### Software Implementation Requirements

* Debug Port I/O range must be fixed. The location of these registers is described using the ACPI FADT table. (See "Debug Port Table" section later in this specification.)
* Debug Port and I/O range must be reported as system resources in the ACPI BIOS.
* Debug Port must also be claimed by the PNP0C02 devnode, so that its resources are not allocated to anything else and detection does not try to find it.

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### Physical Connection Requirements

* The Debug Port connector must not be easily accessible to the end user, so that the end user does not accidentally connect any devices to this port. It should, however, be relatively easy to access by field support personnel (for example, behind a removable panel).
* The Debug Port connector must be dedicated for use for system debugging (or as a management port).

[Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top) [Top of page](http://msdn.microsoft.com/en-us/library/windows/hardware/hh134821.aspx#top)

### Debug Port Table

If the Debug Port Table is not present, the operating system should revert to existing debugging devices. Systems indicate the address of the short-term debug port (this table is irrelevant for IEEE 1394 or USB debugging) by way of the Debug Port Table. This table is located in system memory with other ACPI tables. It must be referenced in the ACPI RSDT table.

The presence of the Debug Port Table indicates that the system includes a debug port. The contents contain information about the configuration of the debug port.

**Note:** This table has been submitted to the ACPI specification working group for inclusion in the next release of the ACPI specification. The table definition is reprinted here for reference.

**Debug Port Table Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Header |  |  |  |
| Signature | 4 | 0 | DBGP. Signature for the Debug Port Table. |
| Length | 4 | 4 | Length, in bytes, of the entire Debug Port Table. |
| Revision | 1 | 8 | 1 |
| Checksum | 1 | 9 | Entire table must sum to zero. |
| OEMID | 6 | 10 | OEM ID |
| OEM Table ID | 8 | 16 | For the Debug Port Description Table, the table ID is the manufacturer model ID. |
| OEM Revision | 4 | 24 | OEM revision of Debug Port Description Table for supplied OEM Table ID. |
| Creator ID | 4 | 28 | Vendor ID of utility that created the table. For the DSDT, RSDT, SSDT, and PSDT tables, this is the ID for the ASL Compiler. |
| Creator Revision | 4 | 32 | Revision of utility that created the table. For the DSDT, RSDT, SSDT, and PSDT tables, this is the revision for the ASL Compiler. |
| InterfaceType | 1 | 36 | Indicates the type of the register interface: 0 = full 16550 interface 1 = 16550 subset interface compatible with Microsoft Debug Port Specification 2-255 = reserved. |
| Reserved | 3 | 37 | Must be 0. |
| BASE\_ADDRESS | 12 | 40 | The base address of the Debug Port register set described using the Generic Register Address Structure |

**History**   
v.0.1 - 4/30/99 - First draft compiled from introductory meeting.  
v.0.2 - 5/5/99 - Notes/changes compiled from e-mail review.  
v.0.8 5/13/99 - Advanced version to indicate were closing. Removed CardBus example option; added requirement for 8 ports using UART registers.  
v0.9 - 5/15/99 - Added UART register detailed requirements. Updated signal requirements (3 lines; compatibility); clarifications.  
v0.91 - 5/15/99 - Minor edits.  
v0.92 6/16/99 - Added discussion, general requirements, and clarifications.  
v0.93 6/24/99 - Added debug port table.  
v0.94 7/15/99 - Added long/short approaches; debug table changes; removed questions; integrated exclusions in other sections; added more detail.  
V0.95 7/23/99 - Minor edits.

v1.00RC1 1/28/2000 - Incorporate new information for supported solutions.

v1.00RC2 2/2/2000 - Added requirement that user cannot access serial solution.

v1.00 2/9/2000 - Posted final.