

IEEE Standard for Boot (Initialization Configuration) Firmware: Bus Supplement for IEEE 1496 (SBus)

Sponsor
**Bus Architecture Standards Committee
of the
IEEE Computer Society**

Approved September 22, 1994
IEEE Standards Board

Abstract: Firmware is the read-only-memory (ROM)-based software that controls a computer between the time it is turned on and the time the primary operating-system takes control of the machine. Firmware's responsibilities include testing and initializing the hardware, determining the hardware configuration, loading (or booting) the operating systems, and providing interactive debugging facilities in case of faulty hardware or software. The core requirements and practices specified by IEEE Std 1275-1994 must be supplemented by system-specific requirements to form a complete specification for the firmware for a particular system. This standard establishes such additional requirements pertaining to the bus architecture defined by IEEE Std 1496-1993, IEEE Standard for a Chip and Module Interconnect Bus: SBus.

Keywords: boot, configuration, debug, FCode, firmware, Forth, initialization, plug-in device, ROM

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1994 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1994. Printed in the United States of America.

ISBN 1-55937-463-2

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE standards documents may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Introduction

[This introduction is not a part of IEEE Std 1275.2-1994, IEEE Standard for Boot (Initialization Configuration) Firmware: Bus Supplement for IEEE1496 (SBus).]

Firmware is the ROM-based software that controls a computer between the time it is turned on and the time the primary operating system takes control of the machine. Firmware's responsibilities include testing and initializing the hardware, determining the hardware configuration, loading (or booting) the operating system, and providing interactive debugging facilities in case of faulty hardware or software.

Open Firmware is the firmware architecture defined by IEEE Std 1275-1994, IEEE Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices. That standard is bus-independent, instruction-set-independent, and system-independent.

The core requirements and practices specified by IEEE Std 1275-1994 must be supplemented by system-specific requirements to form a complete specification for the firmware for a particular system. This standard establishes such additional requirements pertaining to the bus architecture defined by IEEE Std 1496-1993, IEEE Standard for a Chip and Module Interconnect Bus: SBus.

The following individuals were members of the P1275 Working Group at the time this document was produced:

William M. (Mitch) Bradley, *Chair*
David M. Kahn, *Vice Chair*
John Rible (*Draft Editor*)
Luan D. Nguyen, *Secretary*

Ron Hochsprung
Thanos Mentzelopoulos

David L. Paktor
Yongjae Rim
Paul Thomas

Mike Tuciarone
Martin Walsh

The following individuals contributed to this document:

Paul Fischer

Shawn Morrissey
Mike Williams

Ilan Rabinowitz

The following persons were on the balloting committee:

Ray S. Alderman
Keith D. Anthony
Pierre Joannes Aulagnier
Christos Bezirtzoglou
William M. Bradley
John C. Brightwell
Charles Brill
Andy Cheese
C. H. Chen
Robert Crowder
Samuel Duncan
Sourav K. Dutta
Robert Elliott
Paul Fischer
Wayne Fischer
Gordon Force
Julio Gonzalez Sanz
David Kahn

Thomas M. Kurihara
Conrad Laurvick
Gerald E. Laws
Bruce D. Lightner
Boon Lum Lim
Rollins Linser
Tim P. Marsland
Susan Mason
S. Fenton McDonald
Thanos Mentzelopoulos
Klaus Dieter Mueller
Luan Duy Nguyen
Ken Nicholas
Michael Orlovsky
Granville Ott
David L. Paktor
L. M. Patnaik

Mira Pauker
John Rible
Gary S. Robinson
Frederick E. Sauer
Gary Sloane
Robert Snively
Richard Spratt
Dongzhuang Su
Nobuaki Sugiura
Paul Thomas
Michael G. Thompson
Marc Tremblay
Robert Tripi
Michael Tuciarone
Martin Walsh
David Weaver
H. Michael Wenzel
Michael G. Williams

When the IEEE Standards Board approved this standard on September 22, 1994, it had the following membership:

Wallace S. Read, *Chair*
Donald C. Loughry, *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
Bruce B. Barrow
José A. Berrios de la Paz
Clyde R. Camp
James Costantino
Stephen L. Diamond
Donald C. Fleckenstein
Jay Forster*
Ramiro Garcia

Donald N. Heirman
Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
E. G. “Al” Kiener
Ivor N. Knight

Joseph L. Koepfinger*
D. N. “Jim” Logothetis
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall

Richard B. Engelman

Robert E. Hebner
David E. Soffrin

Paula M. Kelty
IEEE Standards Project Editor

CLAUSE	PAGE
1. Overview	1
2. References	1
3. Definitions of terms	1
4. Bus characteristics.....	2
4.1 Physical address formats and representations	2
4.2 Bus-specific configuration variables.....	2
4.3 Format of a probe list	2
4.4 Interrupt-specification format	2
4.5 FCode interpretation semantics.....	3
5. Bus nodes	3
5.1 Properties	3
5.2 Methods.....	6
6. Child nodes	7
6.1 Properties	7
6.2 Methods.....	8
7. User interface extensions	8
7.1 Bus-specific interpretations of standard commands	8
7.2 Bus-specific FCodes	8
7.3 Bus-specific user interface commands.....	8