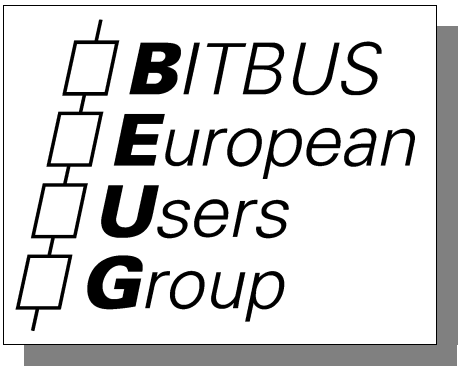
BITBUS
IEEE 1118 BROADCAST SERVICE IMPLEMENTATION

A BEUG Recommendation



Description: *Implementation of broadcast/multicast as further specification to the IEEE-1118 standard »microcontroller system serial control bus«*

Date: *May 4, 1995*

Last Change: *July 21, 1998, common format for all recommendations, PDF conversion. VG*

Status: *Accepted by BEUG on May 4, 1995*

Authors: *Frank J. Furrer, Volker Goller, Niels Trapp*

Contact: *e-mail: wg1@bitbus.org*

Contents

1. General

Broadcast (= send a message from the BITBUS-master to all the slaves in parallel, no answer allowed) and Multicast (= send a message from the BITBUS-master to a specified group of slaves in parallel, no answer allowed) are important functions for certain applications requiring simultaneous actions or synchronization. Unfortunately, the IEEE-1118 standard is not clear in details about the implementation of the broadcast- and multicast-functions.

This BEUG recommendation provides additional implementation information, thus interpreting the IEEE-1118 standard. This recommendation is limited to Broadcast- and Multicast-implementations with messages to user tasks and the extension. System management services are not treated in this recommendation, but will be considered later.

Broadcast- and Multicast services are implemented as connectionless services and the datagram type service is used. No establishment of a connection prior to communication is required and no answer from the receiving station is expected, nor allowed. RAC-services are described in the IEEE-1118 as Generic Bus Services (GBS): these services are connection-oriented by default. Therefore different mechanisms are used for the transportation and access to Broadcast- and Multicast services.

2. Data Link Layer

2.1. Data-Link Layer PDU

An IEEE-1118 Unnumbered Information Frame ("UI") is used to send Broadcast- and Multicast messages. The data link layer (DLL) protocol data unit (PDU) has the following structure ([1], 5.2.1.1):

```
PFS FLAG Addr CF ------------- INFO --------------- FCS FLAG
```

Addr: 255d is used as a Broadcast-address, 241d .. 254d are used as Multicast-addresses. (Note: in the IEEE-1118 [1] the boundary between logical addresses and the first multicast address is called a "fence" and the fence is a configuration parameter in the system, i.e.241d must not necessarily be the first multicast address)

CF: 03H ([2], p.81) = unnumbered Information frame with P/F = 0, i.e. no response allowed.

INFO: contains Application Layer (ALL) PDU.

2.2. Application Layer PDU

The INFO-field is structured as follows (=Application Layer ALL PDU. [1], 5.2.1.1):

LEN RES CMD LAYER ENTITY ---------- DATA ----------

LEN:	Length of messages (Number of data bytes + 5). Note: LEN = 7 Bit + Odd Parity ([1], 5.2.1.1.1), LEN < 128
RES:	reserved, set to 00H by sending station
CMD:	C0H ([1], 5.2.1.1.3, i.e. not System Management Protocol)
Layer:	07H ([1], 5.2.1.1.4, i.e. directed to Application Layer)
Entity:	FFH ([1], 5.2.1.1.5, i.e. User defined Application Layer Entity "nn", i.e. directed into the Broadcast/Multicast Receive Queue)
DATA:	contains the data bytes (8 bit transparent, no restrictions).

3. Client/Server-Model

For the broadcast-/multicast messaging a client/server model is used: the BITBUS master (= source of broadcast-/multicast message) acts as a server and sends out information to a group of clients (= BITBUS slaves) when it is available. The clients either wait or regularly check for the availability of the broadcast-/multicast-information. The implementation is best done using separate queues for sending and receiving broadcast-/multicast messages, thus clearly separating the handling of broadcast-/multicast messages from the GBS-services.

The "CMD"-byte (= C0H, see [1], 5.2.1.1.3, i.e. not System Management Protocol) and the "Layer"-byte (= 07H, see [1], 5.2.1.1.4) in the received PDU directs the received message to the application layer entity, i.e. into the receive queue available to user tasks or the extension in the slave. System management PDU's in later implementations will not be written into the receive queue, but will be directly processed by the system management firmware.

All received broadcast-/multicast-messages are written into the receive queue where they may or may not be read by any task or the extension in the slave. The size of the queue is limited to a maximum number of messages (e.g. 8 messages), therefore a mechanism avoiding blocking the system after a receive queue overflow must be implemented, e.g. ignore incoming messages as soon as the receive queue is full.

It is recommended to discard incoming messages directed to broad- or multicast addresses which are not served by a client task, i.e. if no "Initiate Accept Boradcast-/Multicast-Message"-Call for such an address has been made.

4. i8044-Behaviour

Multicast and Broadcast services are not implemented in the i8044, therefore these services are not available. The question arises, how i8044 nodes in a network using Multicast and Broadcast behave. Unfortunately, the i8044 firmware does not check the P/F-bit and responds with a "frame reject" message when an unumbered frame as used in Multicast and Broadcast is addressed to its own logical address or to the broadcast address FFH.

In any network containing i8044 therefore the address range 241d .. 254d shall not be used as logical addresses by any station. This should not cause any conflicts, because the addresses 251d ... 255d were already excluded (i.e. reserved by INTEL) in the original specification ([3], 4.2.2).

5. Communication between Protocol Microcontroller and an Extension

An extension is coupled to its own protocol microcontroller via e.g. shared-RAM or FIFO. The applications software in the extension can communicate with tasks in its own microcontroller using standard message formats. Unfortunately, the fictive address "FFH" (= 255d) has often been used to direct messages between extension and protocol microcontroller. When using the broadcast service, this older convention now causes a conflict in the BITBUS-master because a message with address = FFH must now be sent as a broadcast message to the network (and is not any more directed to a task in its own protocol microcontroller).

The following use of addresses results as the only logical and feasible convention:

Address for communication between extension and own protocol microcontroller: 00H (in master and in slaves)

Address for intertask-communication in the protocol microcontroller: 00H (in master and in slaves)

6. Application Layer Services

For the broadcast-/multicast messaging four new application layer services are provided:

a) Multicast Send Request (BITBUS master only): This service transmits a message addressed to a broadcast or multicast address.
b) Initiate Accept Broadcast-/Multicast Message (BITBUS slave only): This call informs the operating system which task is waiting for which broadcast- or multicast-messages. This call is used by the operating system to build up its distribution table for the received broadcast-/multicast messages. Note that this call must be issued only once for the broadcast- and for each multicast-address!
c) WAIT for Broadcast-/Multicast-Message (BITBUS slave only): The task is suspended until a message with the corresponding broadcast- and/or multicast address is received. Note that this service watches for alle addresses defined by the "Initiate Accept Broadcast-/Multicast Message" for the task. The task becomes immediately active when a message is received (if no higher priority task is running). Note that only the task which issued the Initiate Accept Broadcast-/Multicast Message-Call is allowed to use this call!
d) CANCEL Accept Broadcast-/Multicast Message (BITBUS slave only): This call cancels a previous "Initiate Accept Broadcast-/Multicast Message"-Call, i.e. the task will not longer be connected to this Broadcast-/Multicast Message-address. Note that this call is automatically executed by the operating system whenever a task is deleted.

The four services should be implemented as additional system calls in the real-time operating system used in the BITBUS protocol firmware.

7. References

[1]IEEE Std 1118-1990: IEEE Standard Microcontroller System Serial Bus, August 5, 1991.
[2]Frank J. Furrer (Ed.): BITBUS - Grundlagen und Praxis. Hüthig Buch-Verlag, GmbH, Heidelberg, 1994. ISBN 3-7785-2250-7.
[3]INTEL Corporation: The BITBUS Interconnect Serial Control Bus Specification.