ALMA08001Nx0004A

(ALMA Computing Memo 6)

ALMA-SW-0006

ALMA Monitor and Control Bus

Standard Interface Requirements

Larry D'Addario, Mick Brooks

2001-02-20

Revision History

Date	Revision	Description	Contributors
2001-01-17	1.0	First public release	Larry D'Addario, Mick Brooks
2001-02-20	A	Revise following review	Review committee

Table of Contents:

1 Introduction

All devices on the ALMA Monitor/Control Bus (AMB) must conform to the bus interface specification given in ALMA08001Nx0001 [1]. This specification is based on CAN [2], but it adds a few additional rules, principally that there can be only one Bus Master device and all others must act as slaves, responding only when polled. In general, any device that fully conforms to this specification may be attached to the bus. However, it is highly desirable for the project to provide an AMB Standard Interface (AMBSI) to be placed between the bus and the specialized circuitry of each slave device, and to expect that one of these standard interfaces will be incorporated into nearly all slave devices. There are two main reasons for this:

1. Standardization of the interfaces allows us to maintain better control of bus operation, ensuring that the ALMA master-slave protocol is followed correctly and ensuring isolation among devices on the bus. It also makes it easier to implement changes or enhancements to the bus protocol in the future, if necessary.

2. For specialized devices developed within the ALMA project (which will be most of the devices on the bus), incorporation of a standard interface by each device designer avoids duplication of effort. The interface can be designed once and re-used in many places, rather than being designed independently in various labs. Design errors are discovered and corrected once, not re-discovered multiple times.

The second objective can be accomplished only if the standard interface is designed to satisfy the monitor and control needs of most devices. Otherwise, the device designer must spend a comparable amount of effort building an interface to the interface. Therefore, the features and support functions provided by the AMBSIs at the device side must be carefully considered. These are discussed in the present document.

Any proposed AMBSI design should be shown to meet these requirements, and if it does not do so completely then the exceptions should be fully explained.

2 Implications of AMB Specification

2.1 Data Rates

Each slave node on the AMB is required to handle the full operating rate of the CAN bus, or 1Mbps.

In addition, the maximum transaction rate (monitor and control messages) is 2000 per second.

2.2 Serial Number

As required by [1], the AMBSI should use a device from the Dallas Semiconductor Silicon Serial Number range in order to provide a unique 64-bit serial number.

2.3 Bus Node Addressing

As explained in [1], each interface on a single bus must have a unique 11-bit node address. Since the AMBSI will be mass produced, some means of setting the node address of each unit must be provided. It is unlikely that all addresses allowed by the 11-bit number will be needed, so it is possible to provide a smaller number of settable bits. Some of these should be in non-volatile memory on the board, such as DIP switches. Electronic NV memory may be used, but then a simple means of setting the address bits under field conditions (away from the laboratory) must be provided.

At least four bits of the address must be determined by off-board wiring. This is to allow the AMBSI to be embedded in a module of which several copies are used on one bus (e.g., at an antenna); the modules are identical but must have different addresses according to where they are installed in the system. This is accomplished by wiring through the module's back plane connector. The largest number of identical modules on one bus in ALMA is currently 8; the specified 4 bits allow up to 16 such modules.

3 Required Logical Features, Device Side

3.1 Device Connection Styles

To accommodate a wide range of device complexity, the AMBSI should provide several styles of sending digital command data and retrieving monitor data to/from the device. It is believed that the following three styles are sufficient to satisfy the needs of all devices in the ALMA telescope.

3.1.1 Latched output bits for control signals; sampled input bits for monitor data

In this connection style, control data bits are latched upon receipt of a control message and each such bit drives a dedicated connection to the device. Similarly, monitor data bits from the device are wired to other dedicated connections; these are sampled by the AMBSI upon receipt of appropriate monitor request messages.

This style supports simple devices which require a relatively small number of control and/or monitor bits. It requires little or no logic outside the interface. The number of bits supported in each direction depends on the number of separate connections (pins) that the AMBSI provides on its device side; this is a design parameter. The mapping between each device connection and a specific data bit within a CAN bus message having a specific ID depends on firmware running in the AMBSI.

3.1.2 Parallel, addressed, read/write interface to device (memory bus style)

In this style, the device connection consists of a set of bi-directional data lines, a set of AMBSI-to-device address lines, and several handshaking lines. A control message on the bus results in the AMBSI setting the address lines and data lines, telling the device that data is available, and accepting an acknowledgment. A monitor request results in setting the address lines, telling the device that data is requested, accepting a data ready signal and reading the data lines.

This style supports devices in which data to/from the bus is mapped to a RAM in the device; or devices which must support many more command and/or monitor bits than can be handled by a practical number of separate connections to the AMBSI. The device must provide decoding of the address and handshaking signals, and appropriate routing of the data.

The address range and data word width that can be supported is limited by the number of separate connections provided, and a design compromise may be necessary. The bus specification allows 18 b of relative address and up to 64 b of data per message, so an 82-pin connection (plus handshaking signals) would be required to support this fully. Much less than this is believed to be adequate for nearly all devices; the actual number of pins provided by the AMBSI is a design parameter.

3.1.3 Bi-directional serial interface

Here data is transferred to and from the device as a synchronous serial stream. The AMBSI supplies a clock signal at a specified rate (which is a design parameter) whenever data is to be transferred. Upon receipt of a control message, the relative address and variable-length data are sent to the device serially on one data line; an acknowledge (or error code) bit sequence is then sent to the AMBSI on the same or another line. Upon receipt of a monitor request message, the relative address is sent to the device and (after a specified delay) the monitor data is sent to the AMBSI. (The exact protocol may differ somewhat from this description, and is a matter for detailed design; it is controlled by firmware in the AMBSI.)

This style is primarily intended to support devices containing a microprocessor that can support the serial protocol. This allows the device microprocessor to handle all device-specific functions and leave all bus-related functions to the AMBSI, thus ensuring isolation from other devices on the bus. It might also be possible to support the serial protocol with fixed logic (e.g., in an FPGA or CPLD) rather than with a microprocessor. This would reduce the number of wires between the AMBSI and the device compared with using the parallel interface (section 3.1.2 above), which might be convenient for some devices.

3.2 Analog voltage monitoring

Many, but not all, devices in ALMA are expected to require routine monitoring of signal voltages at relatively low resolution (~0.1% precision). Support for several separate channels of analog monitoring within the AMBSI is therefore desirable.

Since not all devices need this feature, its implementation should not be allowed to determine the size, cost or power consumption of the AMBSI. It turns out that many microcontrollers that are candidates for internal use in the AMBSI have an on-chip ADC and analog multiplexer, allowing analog monitoring to be provided while incurring negligible penalty if it is not used. Nevertheless, if it is impractical to include analog monitoring in the AMBSI without large size or cost penalties, then a separate add-on "analog module" should be provided. It would connect to7 the device side of the AMBSI and would provide the ADC, MUX, and control logic needed to digitize several analog signals from the device. The analog module should also be a standard item, to be used in each device that needs it, without its having to be separately designed for each case.

3.3 Device reset signal

On a dedicated connection from AMBSI to device, a brief pulse is produced whenever a control message is received on a specific CAN ID.

For those devices where it is appropriate, this signal should be connected so as to force the device into a predetermined, fixed state when the pulse occurs. Among other things, it can be used to perform a hardware reset of a microprocessor.

3.4 Device-specific features

Whereas all known practical implementations of AMB interfaces involve the use of a microcontroller, it is possible to include code in that controller to perform device-specific functions. The AMBSI would then be "customized" for each such device by loading the additional code. (A non-customized version, containing only the "standard" code and no device-specific code, would still provide the previously described features.) The functionality of each physical connection to the device can then be modified by the specialized firmware.

Implementation of this feature must involve some restrictions. First, the device specific code operates in addition to, not in place of, the standard code. The standard code handles all bus transactions, and the device specific code must not interfere with this; coding rules will be issued to ensure this. A tightly controlled software interface to the bus control code will be provided. These rules may significantly limit the functionality that a device designer can place in the AMBSI; complex devices may need a separate microcontroller to achieve sufficient functionality. Second, the AMBSI hardware must provide a means to load new firmware into non-volatile memory after the hardware is manufactured. This allows the hardware to be mass produced and customized afterwards.

4 Desired Physical Features

4.1 Small size

The AMBSI is required to be as small as possible, and must fit easily within a single-wide Australia Telescope style module while allowing plenty of room for other circuitry. (AT modules have inside height of 278 mm and depth of 410 mm and several widths.) It should be built as a single printed circuit board. Other alternatives, such as fabrication as a small daughtercard, and provision for device designers to directly lay the circuit out on their own boards may also be investigated.

4.2 Low power consumption

To facilitate its use in tightly shielded modules, power consumption and the resulting heat dissipation should be minimized. A goal of less than 1.0 W is suggested. It is preferable if it is powered by a single D.C. voltage from among the standard "backplane" voltages adopted by the project [3].

4.3 Low RFI

Precautions should be taken to minimize conducted and radiated emissions, although no specific targets are currently established. RFI limits are expected to be imposed at the device level, so the device designer must provide any isolation or shielding that may be required. To facilitate this, the emission characteristics of the AMBSI should be well established by testing.

4.4 Low cost

A target marginal cost of <$200 per unit in batches of 100 units is suggested. Much lower cost is desirable.

5 Design Tradeoffs

The required and desired features are to some extent conflicting, necessitating some tradeoffs. Considerations are discussed here.

It is possible that more than one version of the AMBSI will be produced, with each providing a subset of the required features but none providing all of them. Of course, the number of different versions should be minimized. Ideally there should be only one version of the hardware to be manufactured, with variations in features determined only by differences in firmware, but two or perhaps three different hardware versions could be considered if necessary. If variations in firmware are used to produced different versions, then it is highly desirable for these to be loadable after the hardware is manufactured.

The size, power consumption, and cost are all determined mainly by the number of chips and the number of I/O connections (pins) used in the implementation. There are many considerations in the choices of chips, and these will not be discussed here; generally, the simplest selection that supports the desired functionality is best. The number of pins is a parameter that is subject to more straightforward control. On the device side, as few as 4 signals might be provided if only the serial interface is implemented; and about 100 signals might be needed to support all features fully. The worst-case pin count is dominated by the parallel, addressed style (item 2 above), where the full address and data widths need not be supported; the maximum-possible and minimum-useful signal counts for this mode are:

	Maximum possible	Minimum useful
Address	18	8
Data	64	8
Handshake	3	3

A reasonable compromise within this range should be selected. It is estimated that an AMBSI providing a reasonable level of support for all features will require 30 to 50 device-side pins.

6 References

[1] ALMA08001Nx0001, "ALMA Monitor and Control Bus Specification." Draft dated 2000-10-25. (Formerly Computing Memo No. 7.)

[2] "CAN specification version 2.0." Phillips Semiconductors, 1991.

[3] ALMA09001Nx0004, "Power Distribution For Electronics." Draft specification, 1999-Dec.
http://www.tuc.nrao.edu/~ldaddari/powerDistributionSpec.txt