ALMA Computing Memos No. 4

ALMA M&C Frequently Asked Questions

Mick Brooks, Brian Glendenning
2000-February-29

1. Why CAN and not, say, Ethernet?

The requirements for ALMA M&C, spelled out in [2], include certain levels of determinism that Ethernet cannot guarantee. The use of a master/slave protocol will guarantee a maximum latency for an M&C transaction when used with either CAN or Ethernet. Ethernet has been used with deterministic higher layer protocols such as token rings, but these are declining in use. CAN, however, is naturally more deterministic than Ethernet since it avoids collisions rather than detecting them and backing off for a random amount of time.

CAN is specifically designed for and used in noisy industrial environments. This is the type of environment likely to exist in an antenna where M&C cable runs may be exposed to the environment, or travel near axis motors and compressor systems. It has specific error handling mechanisms that allow operation in conditions of high bit error rates and also guarantees that malfunctioning nodes will shut down and not flood the bus with erroneous packets.

CAN is available in single chip and integrated micro-controller silicon implementations. This allows nodes to be very simple and able to concentrate on local I/O operations rather than communications. Ethernet is almost always used in conjunction with TCP/IP and the CPU load and complexity required by these intermediate protocol layers will be in addition to whatever application level protocol we impose.

As an example of the possible size of the slave nodes for CAN, one engineering group at the NRAO is using an 8 bit PIC 16C74 micro-controller, with only 4K words of program memory and 192 bytes of RAM. This node fully implements the slave node protocol, performs analog to digital conversions and drives a local LCD display.

It is intended that general-purpose networking be used between high-level computers, for example between the central control building and the antennas.

2. Why are we developing special hardware? (The CERN page to give an example had a commercial National card for CAN, working also with LabView).

The hardware subsystems being developed at NRAO for the prototype antennas will all be designed and built in house. The receivers, cryogenics, local oscillators, signal processing and fibre optic transmission systems cannot be purchased off the shelf. Thus at some level, if we wish to monitor and control this equipment, special interfaces will have to be designed and built.

Standard parallel or serial interfaces could be designed into the hardware and connect to a local computer. There simply is not enough space for a dedicated computer system for each of these antenna subsystems. In addition, there is no need for such local processing as the majority of these M&C interfaces merely collect and digitize analog values. The PIC based slave node implementation is fully written in C, for example, but uses no operating system.

All of the serial and parallel connections could be run into a single computer that concentrates the M&C data and makes it available to the rest of the computing system. Some of the subsystems are tens of metres away from the likely location of such a computer. Something other than parallel lines would be necessary. Also, a large number of wires would need to be run, some through cable wraps or on external surfaces of the antenna structure. It was for these reasons that field buses were developed, and CAN is NRAO's choice of field bus for reasons outlined in [2] and [3].

All computers with operating systems will use off-the-shelf CAN boards. National Instruments makes a PCI card for use with LabView; SBS Modular I/O makes a PMC module for use with VME CPUs. Many others are available.

Most NRAO engineering groups wish to include the M&C interface circuitry directly onto their own boards. This saves space, and allows individual designers to properly shield the RF emissions of such interface circuits. These reasons also make developing a small CAN interface circuit desirable.

We could buy commercial CAN nodes. These nodes are usually small boards with a micro-controller and CAN circuitry. There are no devices I have found which satisfy all of our diverse requirements. Given that we need to design our own circuits anyway so we can put them directly on boards, having our own daughtercard is a trivial extension and cuts down on the number of node types we need to support.

NRAO currently has implementations for a Siemens micro-controller with integrated CAN (the C167) and a MicroChip PIC processor with a standalone CAN controller (the Intel 82527).

3. Why don't we specify the same CAN card for Linux and LabView?

The M&C interface specification is a description of how devices connect to the computer system. For the bus master any CAN interface board with software access should suffice. Our work has been with LabView and we have chosen a suitable board with good support. We of course have no objection to someone writing a Linux driver to this board or any other CAN board.

4. Why do we have VxWorks as a CAN master device? Why not Linux?

Properly, the M&C interface describes the protocol and the slave interface but not the implementation of the master. We are proposing both non real-time (LabView) and real-time (VxWorks) masters. While we believe that the choice of VxWorks is appropriate, especially since device characteristics are not yet fully determined, the choice of master is not determined by the device interface or the protocol.

5. However on the CAN node side I could imagine that we might want to allow both VxWorks (e.g. ACU) and Linux/RTLinux (at least to keep a door open).

A node may in principle be implemented in Linux or an RTLinux variant. The project may of course choose to impose standards about the number of supported operating systems. Similarly, rack and other space at the antennas may be very precious. In general, we suspect that a device complex enough to require a real OS is probably too large to be a CAN node. For example, we anticipate the NRAO optical telescope will be controlled entirely by Ethernet.

The ACU is an exception, but as it will be provided by the vendor and will be placed outside the receiver cabin, it is not subject to the previous comments.

6. What is the availability of software drivers?

NRAO has the following commercial drivers:

- National Instruments NICAN. This allows LabView VIs and C/C++ code to access the National Instruments PCI cards on Windows 95/98/NT. CAN bus master code to implement the protocol outlined in [4] has been developed for LabView.

- SBS GreenSpring Modular I/O PMC VxWorks Driver. This code works under VxWorks to access the PMC CAN board. Master code is under development.

NRAO has developed the following slave node implementations:

- Siemens 80C167. Developed using Keil C/C++ compiler and using integrated CAN controller. Demonstrated on Phytec kitCON-167 and miniMODUL-167 boards. Hardware based on these boards is in final layout at time of writing.

- PIC 16C74. Developed using HighTec C compiler and using a standalone Intel 82527 CAN controller interfaced using an SPI port. Demonstrated on a prototype laser control board developed at NRAO Socorro.

All of the above hardware and software has been tested on a CAN network running the protocol defined in the ALMA M&C Interface Specifications.

In addition the following products are available:

- Real Time Devices USA, Inc. (http://www.rtdusa.com/) has PC/104 CAN modules with twisted pair and fiber optic interfaces. Windows and DOS driver support is available from the manufacturer. A Linux driver is under development at NRAO, Tucson.
- Omnitech Robotics, Inc. (http://www.omnitech.com) make an ISA board available in both standard PC and PC/104 form factors. Driver routines written in C are available to support these boards, and would need to be adapted to specific operating systems.
- Zanthic Technologies, Inc (http://www.zanthic.com) also make ISA boards in multiple form factors with DOS driver support.
- Port (http://www.port.de) make an ISA CAN board with ANSI-C driver support and a CANOpen implementation. They also have a Linux driver for their board.
- Compcontrol, Inc. (http://www.compcontrol.com) make a dual CAN channel VME board but do not appear to have drivers.
- Wind River offer PC/104 and ISA CAN boards built by esd, with VxWorks, Windows 95, 98 and NT drivers. They also have CANOpen support for these boards.
- The Dearborn Group (http://www.dgtech.com/products/can.html) make protocol adapters for CAN and have Windows based drivers available.
- I+ME (http://www.ime-actia.com) has CAN cards for ISA, PCI and PC/104 formats. They have driver support mainly for DOS and Windows. In addition they have CAL and CANOpen protocol software.
- Softec (http://www.softec.de) make a VxWorks device driver for Intel 82527 and Philips 82C200 CAN controllers.
- ISK Automation (http://www.isk-automation.de) sells an ISA CAN board with DOS drivers and higher layer support.
- TEWS Datentechnik (http://www.tews-datentechnik.com) make PMC and IP CAN modules with VxWorks, OS9 and pSOS driver support. In fact the TEWS TP816 PMC CAN module and software is resold in the US by SBS Greenspring as their own product.

These products have not been tested with the application level protocol.

7. Why do we have a to develop a special higher level protocol, and not use a commercial product?

A higher level protocol is necessary with CAN to, at the very least, define the byte order of data. The protocol outlined in [4] also defines a slave node identification process and a method for allocating CAN address ranges to slave nodes. Several commercial protocols do essentially the same thing. Some of these are:

- DeviceNet
- Honeywell SDS
- CAN Application Layer (CAL)
- CANopen
- CAN Kingdom

The first two are very proprietary systems requiring you to buy software and hardware from the vendor. They also operate at 500 kbps rather than 1Mbps. For these reasons they were considered unsuitable.

The remaining three protocols are all fairly similar. While they offer peer-to-peer communication during operation, a master is designated for the configuration phase. This configuration phase is quite elaborate and requires the master to transmit information to each node detailing the CAN IDs that that node will use for transmission and reception of data. This means that on a reset or power outage, each node except the configuration master "forgets" its identity. Processing requirements for these protocols are far greater than for CAN alone.

It is not apparent which of these protocols will gain significant market acceptance, although CAL is a good bet. For these reasons, it was deemed premature and unsuitable to adopt one of these protocols and instead to develop a simple protocol of our own.

The majority of the CAN protocol is implemented in silicon. It is intended to use a standard controller such as the Intel 82527 (integrated within the Siemens 80C167). Only 1000 lines of fairly portable C are required to implement the thin protocol layer on top. This application layer is simple, small and quite portable; it is implemented directly on top of the only true (ISO) standard for CAN that exists.

Further work should investigate the suitability of using both CAL and our protocol within the same network. Also validation testing of drivers and hardware for the ALMA protocol needs to be performed.

References

[1] ALMA Software Memo #1, Monitor and Control Points for the MMA, F. Stauffer, 1999-May-11

[2] ALMA Software Memo #5, ALMA Monitor and Control Bus Requirements, M. Brooks, 1999-June-02

[3] ALMA Software Memo #6, ALMA Monitor and Control System, M. Brooks, 1999-June-07

[4] ALMA Software Memo #7, ALMA Monitor and Control Bus Draft Interface Specification, M. Brooks, 1999-December-09