IRAM-COMP-003

Revision: 0
1/13/2005

Contact Author

Institut de RadioAstronomie Millimétrique

PdB CAN Specification

Owner Alain Perrigouard (perrigou@iram.fr)

Keywords:

Approved by:	Date:	Signature:
A.Perrigouard	June 2004	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS

Content

1 Introduction

This document describes the CAN protocol used at IRAM to control and monitor the new generation of receivers for Plateau de Bure. In order to update the receiver cabin and to get rid of all VME chassis which were also needed for other hardware like the subreflector, the 22GHz receiver, etc..., these instruments are now interfaced to the CAN bus, following the same protocol.
This work takes advantage of our experience gained with the Alma project and the memo named ALMA Monitor and Control Bus written by M. Brooks and L. D'Addario is a good reference document.

2 Physical layer

This layer uses the version 2.0 B of the CAN protocol which supports the extended format.
As a consequence the CAN message identifiers are chosen to be 29 bit long.
The CAN bus is operated at 1Mbips.
There are one master node and one or several slave nodes on the CAN bus.
All the bus transactions are initialized by the master.

2.1 CAN-BUS Connector:

IRAM-made CAN nodes will use the standard SUB-D 9 male connector to connect to the Bus. To make bus extension easy, each CAN node will be equipped with 2 similar connectors, internally connected pin-to-pin.
The CAN nodes will be powered through the CAN connectors, which supply 18 to 24 Volt DC on 2 pins (reserved in the CAN standard). Each node will accept any polarity of the CAN power voltage and will have its own voltage regulator, able to supply the necessary voltage(s) from the CAN power. This regulator will be a compact switching module, isolated from the main ground, and from the earth. Precautions must be taken to avoid EMI/RFI interferences, and the 24V input of each CAN node must be filtered with a convenient device (PI filter with self and capacitors).
The CAN Ground is routed on one pin of the connector. Each CAN node is connected to this ground, which is the ONLY reference for the CAN-BUS signals.
The connector housing as well as the cable shield are connected to the local earth (chassis ground) on each CAN node.

Here is the pinout of the CAN connector:

PIN 1	PIN 2	PIN 3	PIN 4	PIN 5	PIN 6	PIN 7	PIN 8	PIN 9
Cable Drain	CAN-L	CAN GND	free	+24V POWER	free	CAN-H	free	0V POWER

2.2 CAN-BUS cable:

The CAN-BUS cable must be able to carry the CAN signals (CAN-L and CAN-H) and the power. The 2 signal wires should be a 120 Ohm twisted pair (or equivalent). The gauge of the 2 wires used for power depends on the current requirements (around 2 Amps), but should not be less than 0.68 mm2.
The CAN-BUS cable must be shielded, and must have a drain wire, connected to the cable shield. This drain connects to pin 1 of each connector. The connector housing must be connected to the cable shield, and metallic shells should be preferently used.
The total length of the cable must not exceed 40 meters when running at 1 Mbit/sec.
The last node on the CAN-BUS must have a 120-Ohm terminator plugged in its unused CAN connector.

3 Communication protocol

The remote frame transmission request (RTR) is not used by the bus master to gather data and to make the difference between control and monitoring request messages.
A monitoring request message issued from the master has no data (its Data Length is set to zero). The response from the slave node has the same message identifier with one or more data bytes.
A control message emitted by the master has at least one byte of data which can eventually be one dummy byte in the case that the message identifier brings all the information. The slave node acknowledges with a message which has the same CAN message identifier but no data.

3.1 CAN Message Identifier

This 29 bit field provides the node address and the relative CAN address (rca).
The node address, number equal or greater than 0 which identifies each slave node is deduced from the first 11 bits of the CAN message identifier.
The case, all the first 11 bits of the CAN Message ID set to 0, is reserved for the general purpose messages broadcasted on the bus by the master node. The CAN Message ID 0 (all 29 bits set to 0) is the first broadcast message and it used to initialize the bus.
In consequence, for a control or monitoring message addressed to a slave node, the node address is equal to the value of the first 11 bit field minus one.
For the control or monitoring messages, the relative CAN addresses (rca) are the last 18 bits of the CAN Message ID. This gives a range of 2^18 messages per slave address.

CAN Message Identifier usage:

Usage	CAN Message Identifier (29bits) in hexa
Broadcast messages issued from the master	00 00 00 00 to 00 03 FF FF
Control and monitoring messages	00 04 00 00 to 1F BF FF FF

Node address and rca ranges for the control and monitoring messages:

Node address	Node Identification CAN Message ID in hexa	Node C&M messages CAN Message ID range in hexa
0	00 04 00 00	00 04 00 01 to 00 07 FF FF
1	00 08 00 00	00 08 00 01 to 00 0B FF FF
2030	1F BC 00 00	1F BC 00 00 to 1F BF FF FF

As mentioned in the reference document, there are only 2031 possible slave nodes from address 0 to 2030 because the CAN specification mandates that the most significant 7 bits of the CAN Message ID must never be all ones.
The Node identification message is a special message, corresponding to the first CAN Message ID in the node Control and Monitoring (C&M) message range which is detailed in a following section.

3.2 Bus Identification

The master can broadcast a bus initialization message (CAN Message ID 0) to identify all the slave nodes present on the CAN bus. At the reception of the broadcasted request message, each slave node writes on the bus a message with its Node Identification as CAN Message ID and with a unique serial number as data. The serial number is an 8 byte vector.
Collisions are naturally solved by the low layers of the CAN protocol.
During this initialization process, Alma foresees to check any node address duplication. A slave node may detect a duplicate node address if two nodes begin transmitting to the same Node Address slot (same Can Message ID) at the same time, but with different serial numbers. In this case the slave node with the higher Serial Number will detect a transmission error during the data section transfer and will cease to transmit at this node address.

3.3 Control Messages

The control messages are issued by the master node and have at least one byte of data.

The acknowledgement bit informs only that at least one slave module has detected the message but does not indicate that the specified slave node in the CAN Message ID has received the control message.
The slave node concerned by the control message should reply with an acknowledgement message which should be a message with the same CAN Message ID but with no data (Data Length 0).
For each slave node the control message specification describes the control message written to the CAN bus by the master node.

3.4 Monitoring Messages

The monitoring request messages are issued by the master node and have no data.
The concerned slave node should answer with a monitoring message which has the same CAN Message ID but with the exact number of bytes expected for data.
For each slave node the monitoring message specification describes the monitoring message written to CAN bus by the slave node as response to a monitoring request message.

3.5 Byte and bit notations

Byte numbering corresponds to the transfer order on the CAN bus. Numbers go from 0 to the maximum of 7. When a block of bytes corresponds to a 16bit or 32bit variable, the byte with the lower number is the high significant byte.
In a Byte, the bits are numbered from 0 to 7. Bit [0] is the less significant and bit [7] is the most significant. When several bytes are concatenated for defining a 16 bit or a 32 bit variable, bit number can go beyond 7 and highest number corresponds to position in the most significant byte i.e. byte with the smaller number. For instance if Byte [2,3] represents a 16 bit variable bit [15] is equivalent to Byte [2] bit [7] and bit [7] is the most significant bit of Byte [3].