



IRAM-COMP-006

Revision: 1  
5 Dec 2005

Contact Author

## Institut de RadioAstronomie Millimétrique

# CanI2C:

## CAN to I2C Bridge

Owner Francis Morel (morel@iram.fr)

**Keywords:**

Approved by:

A.Perrigouard

Date:

December 2005

Signature:

## *Change Record*

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS
1	July 8 2004	Francis Morel		

## Content

<b>1</b>	<b>DESCRIPTION OF THE “CAN2I2C” INTERFACE .....</b>	<b>3</b>
1.1	CAN2IC Controller functionalities: .....	3
<b>2</b>	<b>HARDWARE: .....</b>	<b>3</b>
2.1	CAN2I2C Controller description: .....	3
2.1.1	Controller enclosure: .....	3
2.1.2	Controller front-panel: .....	3
2.1.3	Controller schematics .....	4
<b>3</b>	<b>SOFTWARE: .....</b>	<b>6</b>
3.1	Brief description of the CAN protocol used on the Plateau de Bure: .....	6
3.2	CAN2I2C Controller Background task: .....	6
3.3	CAN commands used by the CAN2I2C Controller: .....	6
3.4	CAN messages used for Receivers control: .....	6
1.		

## 1 DESCRIPTION OF THE “CAN2I2C” INTERFACE

This interface was designed to allow control of the I2C-interfaced modules (Junctions' polarization, HEMT Bias) of the receivers through the CAN Bus.

### 1.1 CAN2I2C Controller functionalities:

The CAN2I2C acts as a bridge between the CAN-Bus and the I2C-Bus. The CAN interface guarantees a complete compatibility with the CAN protocol. A Philips microcontroller is in charge of the I2C protocol, still respecting strictly the I2C norm.

## 2 HARDWARE:

### 2.1 CAN2I2C Controller description:

The board is built around a DIP-164 module developed by Systec. This module is a 40-pin DIP component, it includes Microprocessor, 32k RAM, 32k Flash-EPROM, RS-232 interface, Full-CAN interface. The CAN is not opto-isolated. The I2C is interfaced to this board through a 8-bit I2C microcontroller PCA9564. The CAN2I2C requires less than 200 mA under 5 Volt.

#### 2.1.1 Controller enclosure:

The controller is enclosed in a small metal box, which may be plugged on a DIN rail. The box is powered (5V, 200 mA).

#### 2.1.2 Controller front-panel:

3 LEDS monitor:

-**Power** (green)

-**CAN Message** (yellow)

-**CAN Error** (red).

2 DB-9 male connectors are used for connection to the CAN bus. The 2 connectors are tied pin-to-pin.

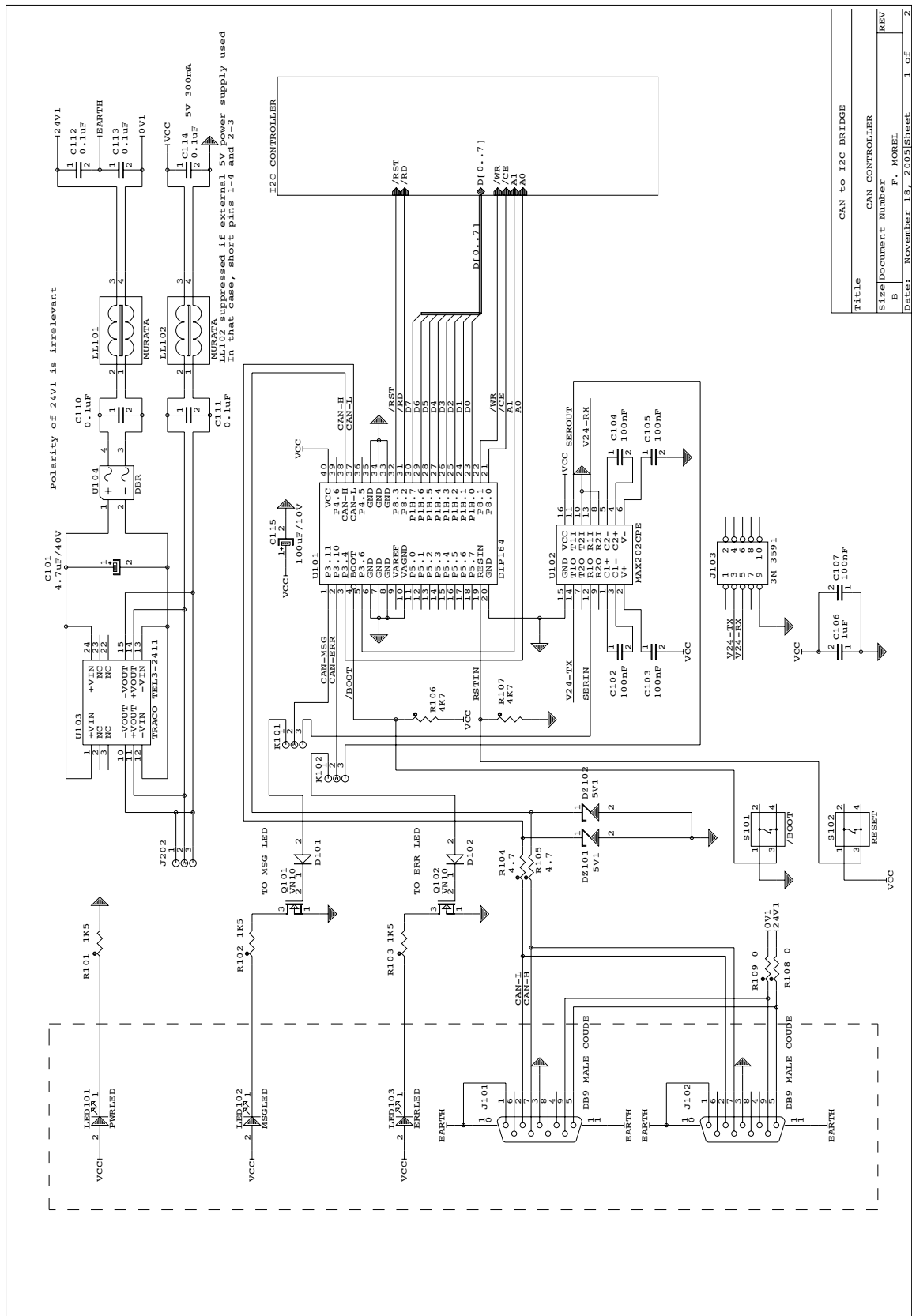
Connections:

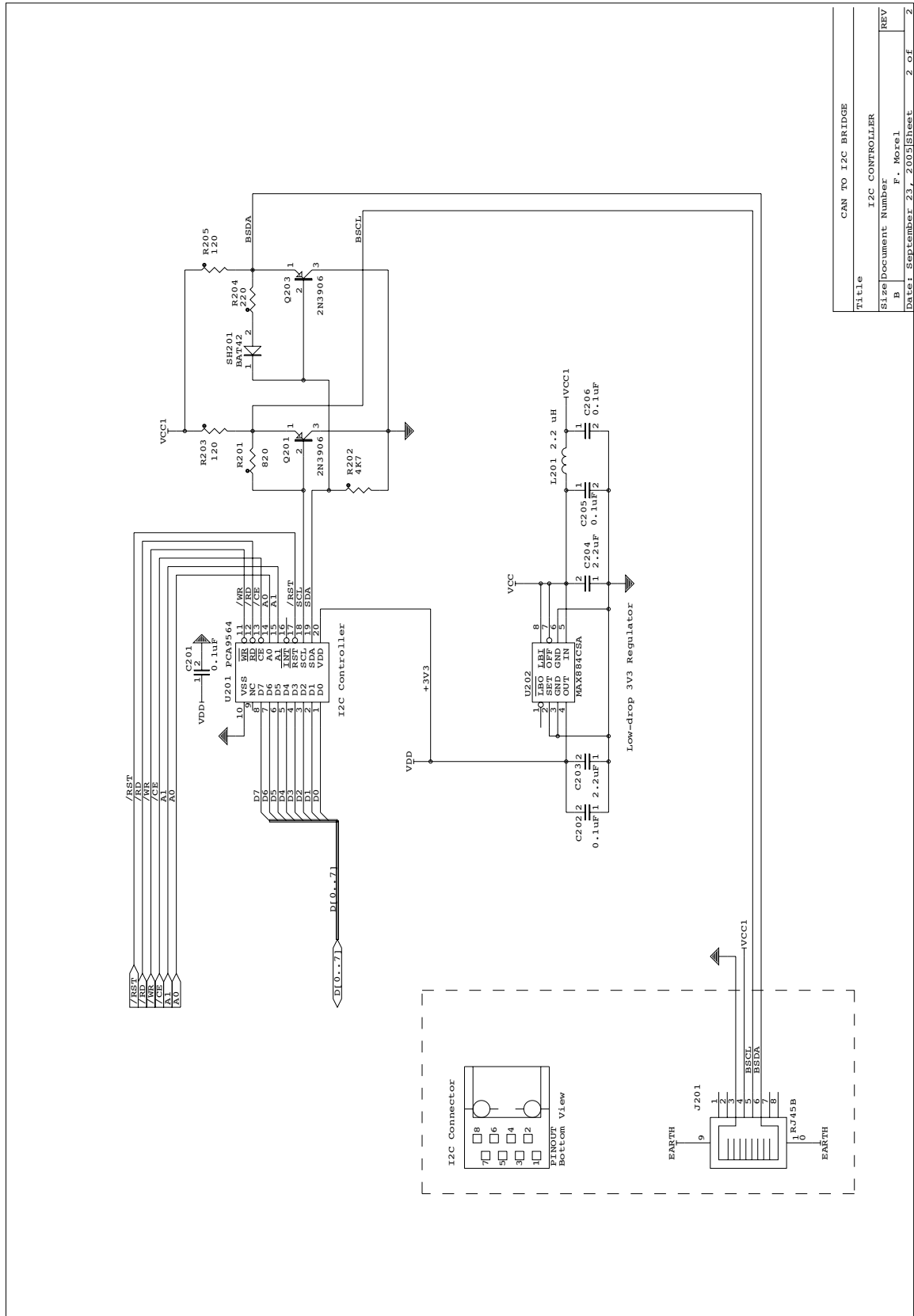
**2: CAN Low**

**7: CAN High**

**3: CAN Gnd**

2.1.3 Controller schematics





Title	CAN TO I2C BRIDGE
Size	I2C CONTROLLER
Document Number	B
Author	F. Morel
Date	September 23, 2005
Sheet	2 of 2

### 3 SOFTWARE:

#### 3.1 Brief description of the CAN protocol used on the Plateau de Bure:

**N.B:** A detailed description of the CAN protocol used on Plateau de Bure is available as document /netapp1/computer/doc/can/canPdBNG/canPdBNG.pdf. Alain Perrigouard wrote it.

Each CAN message includes a header. Inside this header, receiving nodes, to decide whether they are concerned with the current message, use 2 fields: The CAN ID (unique message identifier on 29 bits), and the DLC (Data Length Count) which declares the number of data bytes of the message. If both these parameters match the values expected by a node, it will accept the message.

Each CAN controller has a unique NODE ID, and uses it to filter the incoming CAN messages.

The CAN2I2C Controller accepts 3 kinds of message: Broadcast messages, Control messages and Monitoring messages.

Broadcast messages contain no data. Upon receipt of a Broadcast message, the CAN Controller replies with a message using its own NODE ID as CAN ID.

Control messages must contain at least one byte of data, eventually dummy data if the command is completely defined by the identifier. When receiving a control message, the CAN2I2C Controller will reply with an acknowledge message containing NO data, and having the same CAN-ID as the previously received message.

Monitoring messages contain NO data (DLC = 0). When receiving a monitoring message, the CAN2I2C Controller replies with a message containing a strictly defined number of data bytes, still using the CAN-ID of the received message.

#### 3.2 CAN2I2C Controller Background task:

The controller acts a **slave** device: It receives CAN messages from the CAN master PC, does the requested access to the I2C bus, and replies with CAN acknowledge/data messages. The incoming CAN messages have higher priority than the background task, triggering the CAN interrupt of the C164. A buffer allows storing up to 16 CAN messages.

#### 3.3 CAN commands used by the CAN2I2C Controller:

CAN command	CAN ID	DATA Size	Description
Get I2C status	NODE_ID + 0x1FC	2	Byte[0]: I2C 9564 status Normal value is 0xF8 Byte[1]: transaction report Bit[2]: CAN Error Bit[1]: I2C Write Error Bit[0]: I2C Read Error
Set Serial Number	NODE_ID + 0x1FD	8	Set board serial number Byte[0-1]: Key Byte[2-7]: Serial number
Set NODE_ID	NODE_ID + 0x1FE	8	Set board new CAN ID Byte[0-3]: Key Byte[4-7]: new CAN ID
Reset	NODE_ID + 0x1FF	1	Resets the board

#### 3.4 CAN messages used for Receivers control:

See document "CANPdBNG"