IRAM-COMP-010

Revision: 1
5 Dec 2005

Contact Author

Institut de RadioAstronomie Millimétrique

CanIO:

CAN Digital I/O Board

Owner Francis Morel (morel@iram.fr)

Keywords:

Approved by:	Date:	Signature:
A.Perrigouard	December 2005	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS
1	Jan 11 2004	Francis Morel		

Contents

1.

1 DESCRIPTION OF THE "CANIO" INTERFACE

This interface allows driving 16 optically isolated outputs and reading 16 optically isolated inputs under control of the CAN Bus. The CAN controller of the CANIO board is a C164 Microprocessor with built-in Full-CAN interface.

1.1 CANIO Controller functionalities:

The CANIO board uses optically isolated MOSFET switches as output drivers. Each output is isolated from the ground and from other outputs. The outputs are passive (they do NOT supply any voltage) and are able to drive a 2-Ampere current, withstanding a 48V AC or DC voltage. The polarity of the voltage source connected to the outputs is thus irrelevant. Each output has an ON resistance less than 0.2 Ohm.
The CANIO board uses optocouplers as input isolators. Each input is isolated from the ground and from other inputs. They accept voltages from 5 Volt to 24 Volt, with any polarity. The input current is 2mA under 5V, and 10 mA under 24V.
Inputs are readable as a 16-bit word, using the appropriate CAN command, and outputs are writable in the same way.

2 HARDWARE:

2.1 CANIO 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, CAN interface. The CAN is not opto-isolated. The I/O are interfaced to this board through a 8-bit I/O bus and 6 control signals. 2 push buttons on the board are used for **"Reset"** and **"Boot",** and a HE-10 10 pos connector is available for **RS-232** connection. These are reserved functions, and none of them is accessible when the controller board is inserted into the chassis. See schematics for more details.

2.2 CANIO chassis:

The CANIO is enclosed in a small 3U shielded cassette, which may be inserted in a standard 3U crate (no rear connector necessary) or fixed on a table. The CANIO is powered through the CAN connectors. The 2 CAN connectors are connected pin-to-pin. Pins 5 and 9 are reserved (according to CAN standard) for power supply, and accept voltages from 18V to 36V, with any polarity. Normally, pin 5 is reserved for +24V and pin 9 for 24V return, but inverting the polarity has no incidence. A DC-to-DC switching converter generates the internal voltages; its 24V input is protected with a Pi filter, using a common-mode self to avoid radio interferences.

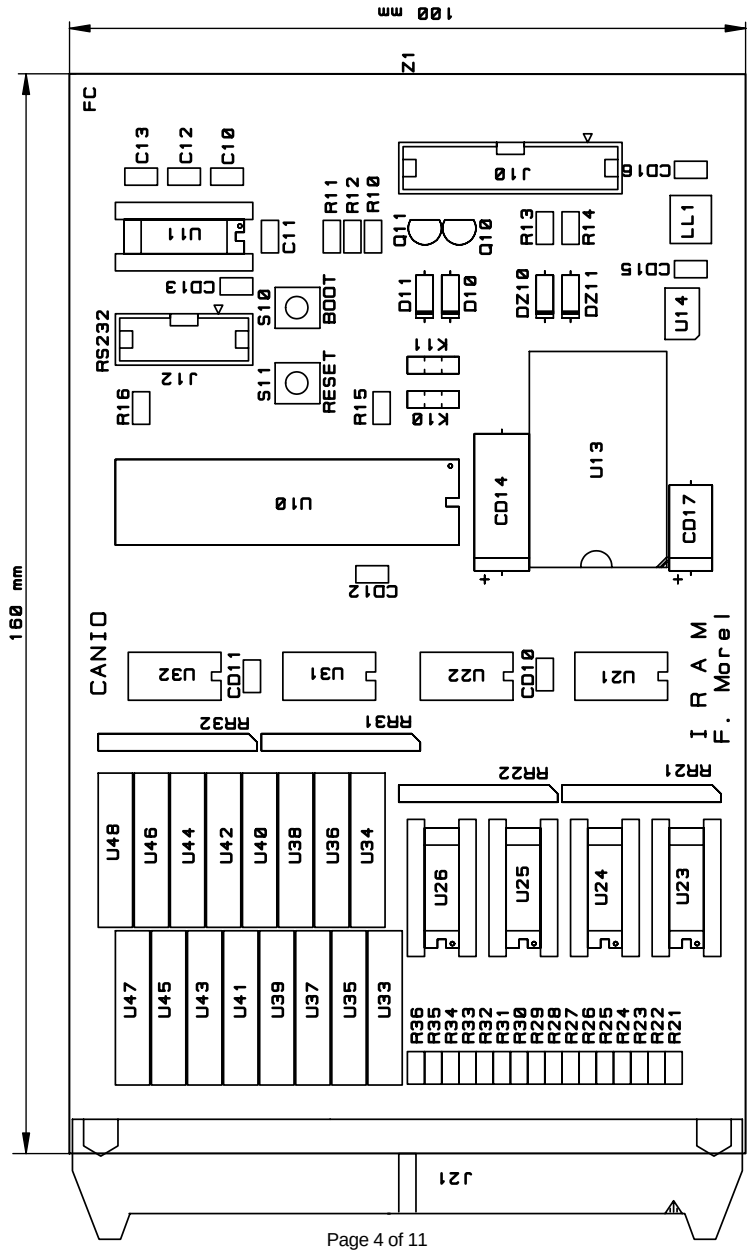
2.3 CANIO front-panel:

A small board (CANIOX) carries the 2 CAN connectors and 3 display LEDs. It connects to the main CANIO board through a 20-pos flat cable. The CANIOX board is fixed behind the front panel with the 4 female screws of the 2 SUB-D 9-pin CAN connectors.
3 LEDS monitor :
-Power (green)
-CAN Message(yellow)
-CAN Error (red**).**

2 DB-9 male connectors used for the CAN bus connections, tied pin-to-pin.
Connections:
1: Chassis ground, Protective Earth, Cable Drain (shield).
2: CAN Low
3: CAN GND (not connected to pin 1)
4: unused
5: +24V Power
6: unused
7: CAN High
8: unused
9: 0V Power

It is important that a shielded cable with drain wire be used for Can-Bus.

2.4 CANIO layout:



2.5 Inputs/Outputs mapping and connexions:

Conventions:
An input is said to be “ON” when current is flowing in the input optocoupler, and the input voltage in that case is (5V <= Vin <= 24V).
An output is said to be “ON” when the optical switch acts like a closed contact. The resistance (that may be measured with a simple Ohmmeter) in that case between the mating output pins is < 0.2 Ohm.

All Input/Output bits are available on a 64-pos HE10 connector located on the front-panel. Pins 1 to 32 are assigned to inputs, and pins 33 to 64 to outputs. The 64-cond flat cable may thus be swapped into two 32-pos cables.

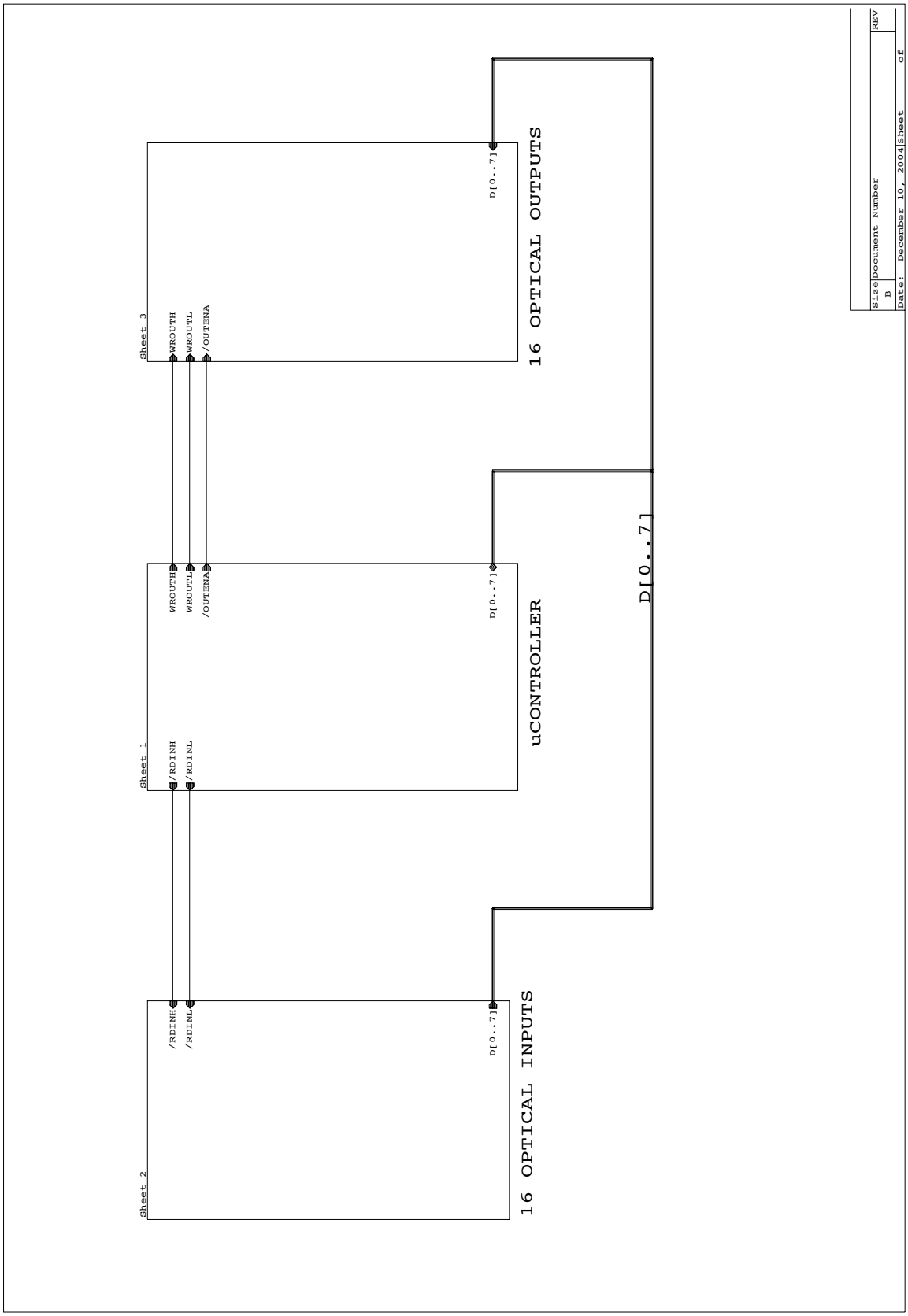
Inputs (Status register):

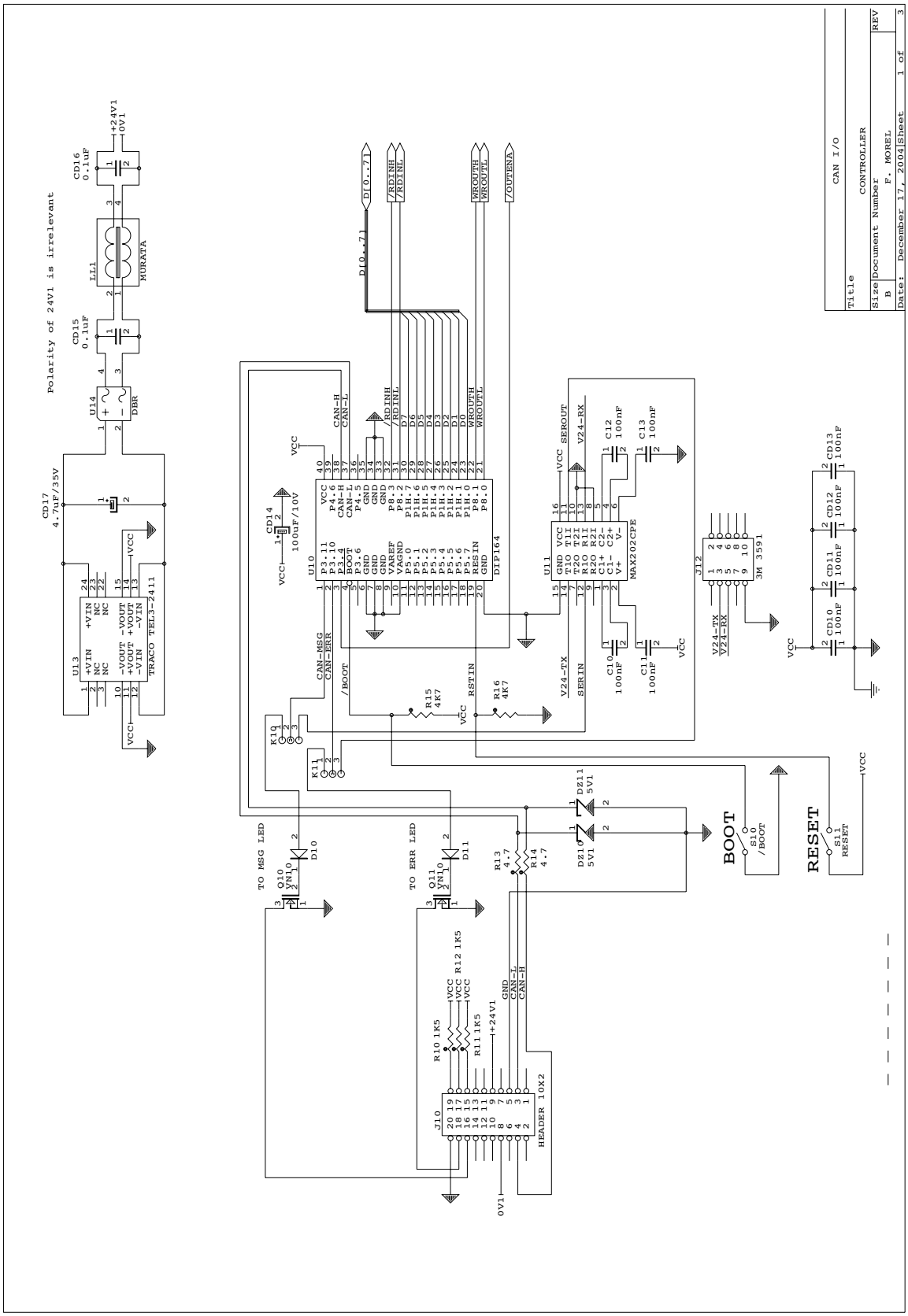
CAN Byte [0] bit [0]	Input [15]	J21 32-31	Bit = 1 means Input “ON”
CAN Byte [0] bit [1]	Input [14]	J21 30-29	
CAN Byte [0] bit [2]	Input [13]	J21 28-27	
CAN Byte [0] bit [3]	Input [12]	J21 26-25	
CAN Byte [0] bit [4]	Input [11]	J21 24-23	
CAN Byte [0] bit [5]	Input [10]	J21 22-21	
CAN Byte [0] bit [6]	Input [09]	J21 20-19	
CAN Byte [0] bit [7]	Input [08]	J21 18-17	
---	---	---	
CAN Byte [1] bit [0]	Input [07]	J21 16-15	
CAN Byte [1] bit [1]	Input [06]	J21 14-13	
CAN Byte [1] bit [2]	Input [05]	J21 12-11	
CAN Byte [1] bit [3]	Input [04]	J21 10-09	
CAN Byte [1] bit [4]	Input [03]	J21 08-07	
CAN Byte [1] bit [5]	Input [02]	J21 06-05	
CAN Byte [1] bit [6]	Input [01]	J21 04-03	
CAN Byte [1] bit [7]	Input [00]	J21 02-01	

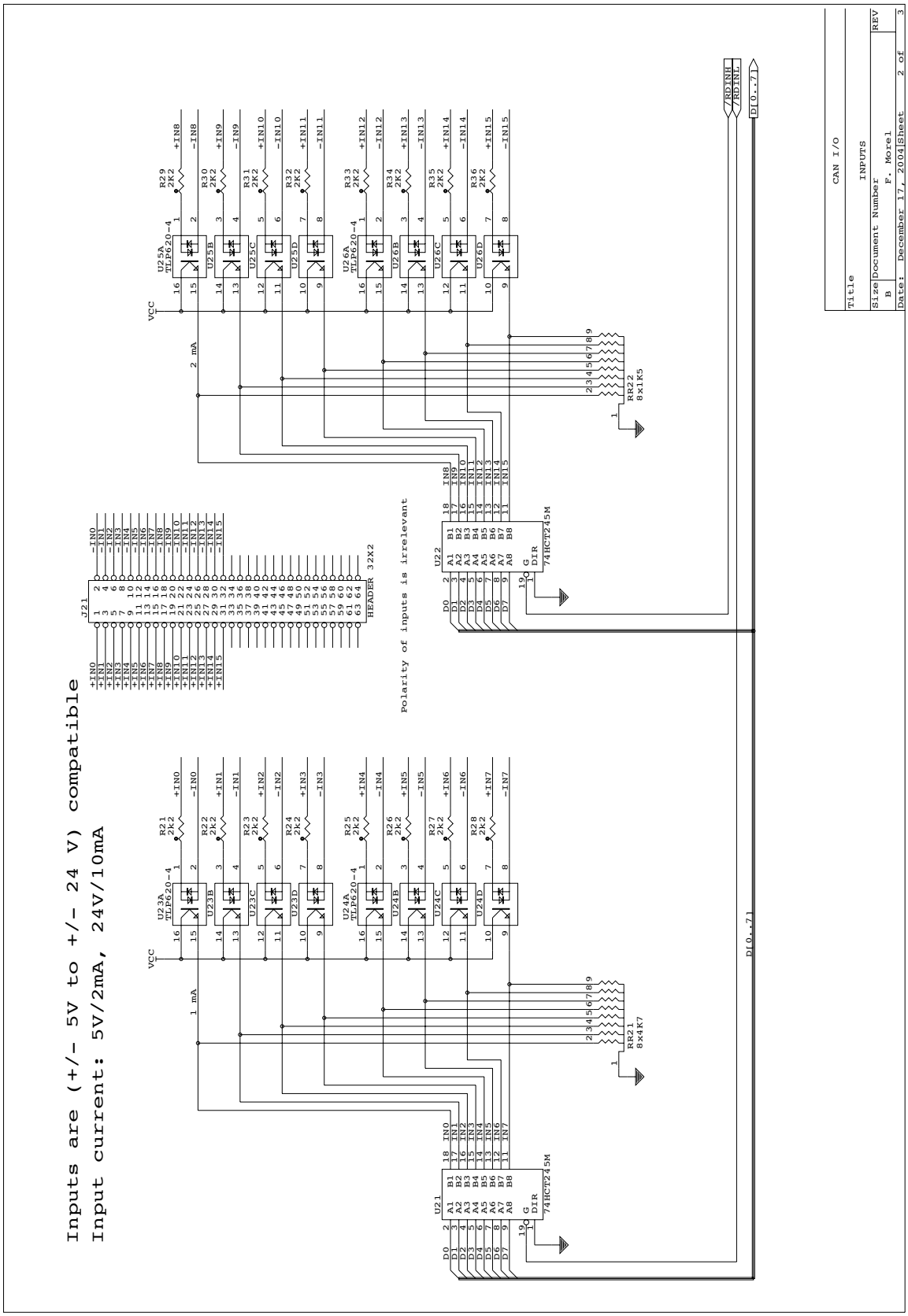
Outputs (Command Register):

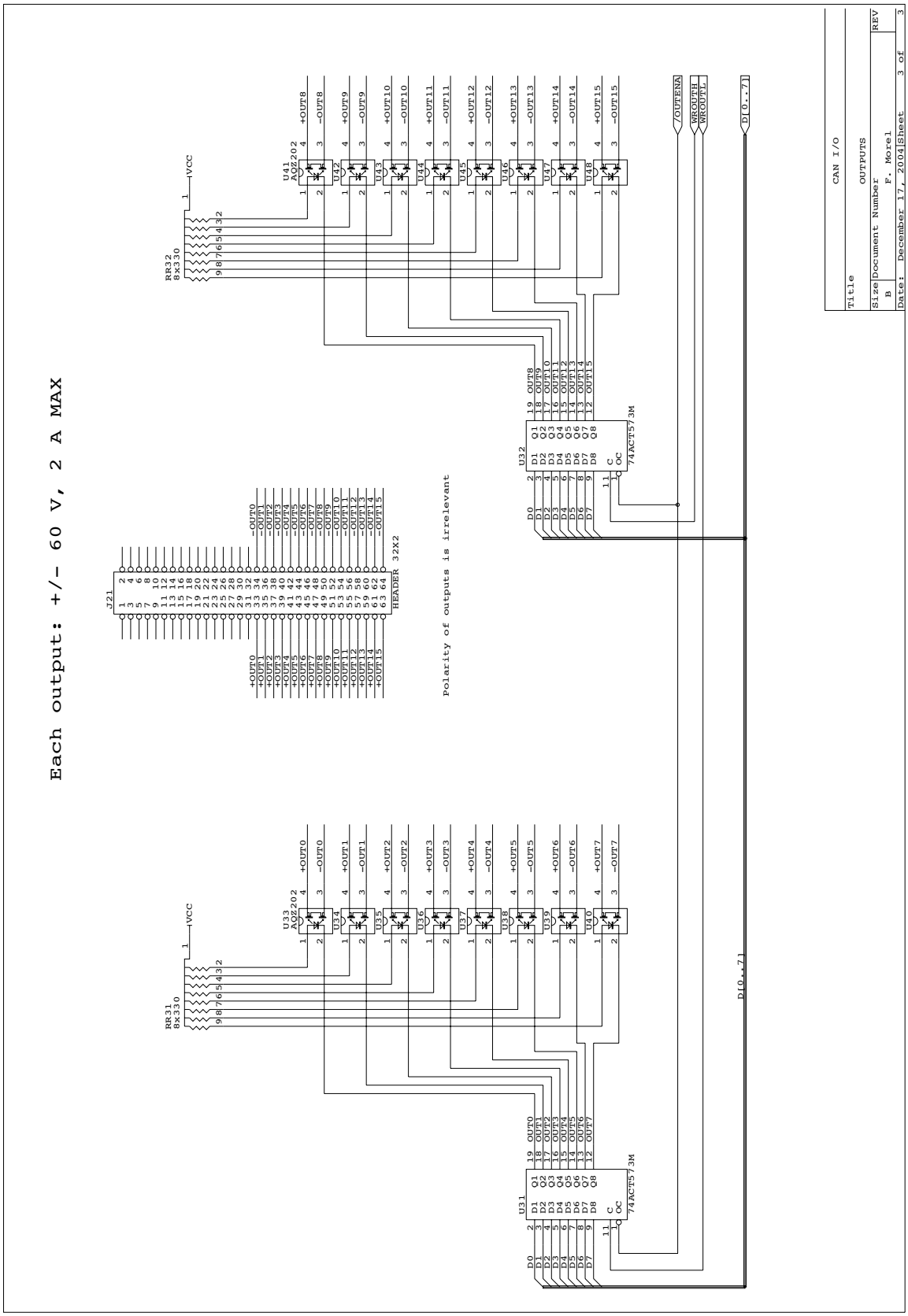
CAN Byte [0] bit [0]	Output [15]	J21 64-63	Bit = 1 sets Output “ON”
CAN Byte [0] bit [1]	Output [14]	J21 62-61	
CAN Byte [0] bit [2]	Output [13]	J21 60-59	
CAN Byte [0] bit [3]	Output [12]	J21 58-57	
CAN Byte [0] bit [4]	Output [11]	J21 56-55	
CAN Byte [0] bit [5]	Output [10]	J21 54-53	
CAN Byte [0] bit [6]	Output [09]	J21 52-51	
CAN Byte [0] bit [7]	Output [08]	J21 50-49	
---	---	---	
CAN Byte [1] bit [0]	Output [07]	J21 48-47	
CAN Byte [1] bit [1]	Output [06]	J21 46-45	
CAN Byte [1] bit [2]	Output [05]	J21 44-43	
CAN Byte [1] bit [3]	Output [04]	J21 42-41	
CAN Byte [1] bit [4]	Output [03]	J21 40-39	
CAN Byte [1] bit [5]	Output [02]	J21 38-37	
CAN Byte [1] bit [6]	Output [01]	J21 36-35	
CAN Byte [1] bit [7]	Output [00]	J21 34-33	

2.6 CANIO Schematics




CAN I/O
CONTROLLER
F. MOREL
December 17, 2004
Polarity of 24V1 is irrelevant
TRACO TEL3-2411
MURATA
DBR
D1P164
MAX202CPE
JM 3591
HEADER 10X2
TO MSG LED
TO ERR LED
BOOT
RESET
/BOOT
RSTIN
SERIN
SEROUT
V24-TX
V24-RX
CAN-MSG
CAN-ERR
CAN-H
CAN-L
/RDINH
/RDINL
WROUTH
WROUTL
/OUTENA
D[0..7]


Inputs are (+/- 5V to +/- 24 V) compatible
Input current: 5V/2mA, 24V/10mA
Polarity of inputs is irrelevant
CAN I/O
INPUTS
HEADER 32X2
74HCT245M
RR21 8x4K7
RR22 8x1K5
2 mA
1 mA
VCC
DI0..7
/RDINH
/RDINL


Each output: +/- 60 V, 2 A MAX
CAN I/O
OUTPUTS
Polarity of outputs is irrelevant
HEADER 32X2
74ACT573M
/OUTENA
WROUTH
WROUTL
D[0..7]
D[0..7]
VCC

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 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 CAN2VME 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 CAN2VME 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 CAN2VME Controller replies with a message containing a strictly defined number of data bytes, still using the CAN-ID of the received message.

3.2 CANIO Controller Background task:

The controller normally acts a **slave** device: It receives CAN control/monitoring messages from the CAN master PC, does the requested job, and replies with 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.
The actual inputs' value may be read, and the outputs' value may be written at any time.
The last outputs' value written through CAN is available in a register, which may be reread if necessary.

3.3 CAN commands used by the CANIO board:

CAN command	CAN ID	DATA Size	Description
Get digital inputs	NODE_ID + 0x100	3	Read 16 isolated inputs Byte[0]: inputs [15-8] Byte[1]: inputs [7-0] Byte [2]: Bit 2: CAN Error
Set digital outputs	NODE_ID + 0x110	2	Set 16 isolated outputs Byte[0]: outputs [15-8] Byte[1]: outputs[7-0]
Get digital outputs	NODE_ID + 0x120	3	Reread outputs Byte[0]: outputs [15-8] Byte[1]: outputs[8-0] Byte[2]: Bit 2: CAN 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 commands used for receiver control:

See document “CANPdBNG”