IRAM-COMP-009

Revision: 1
29 JUN 2004

Contact Author

Institut de RadioAstronomie Millimétrique

CanAna: CAN DAC/ADC 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	Apr 26 2005	Francis Morel		

Content

DESCRIPTION OF THE "CANANA" BOARD:

This board was designed to allow analog voltage generation and readout through the CAN Bus.

1.1 CANANA functionalities:

The CANANA outputs 16 analog 14-bit voltages (0-10Volt) and inputs 16 analog 16-bit voltages (0-10 Volt).

HARDWARE:

1.2 CANANA 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.
A 14-bit SPI DAC (AD5390) is in charge of the 16 analog outputs. All outputs share a common ground reference.
A 16-bit SPI ADC (AD977) is used for reading the 16 differential inputs. Each input has its own ground reference.

1.2.1 Controller enclosure:

The controller is enclosed in a small metal box, which may be plugged on a DIN rail. The box is powered (24V, 100 mA) through the CAN connector.

1.2.2 Controller front-panel:

3 LEDS monitor:
-Power (green)
-CAN Message (yellow)
-CAN Error (red).

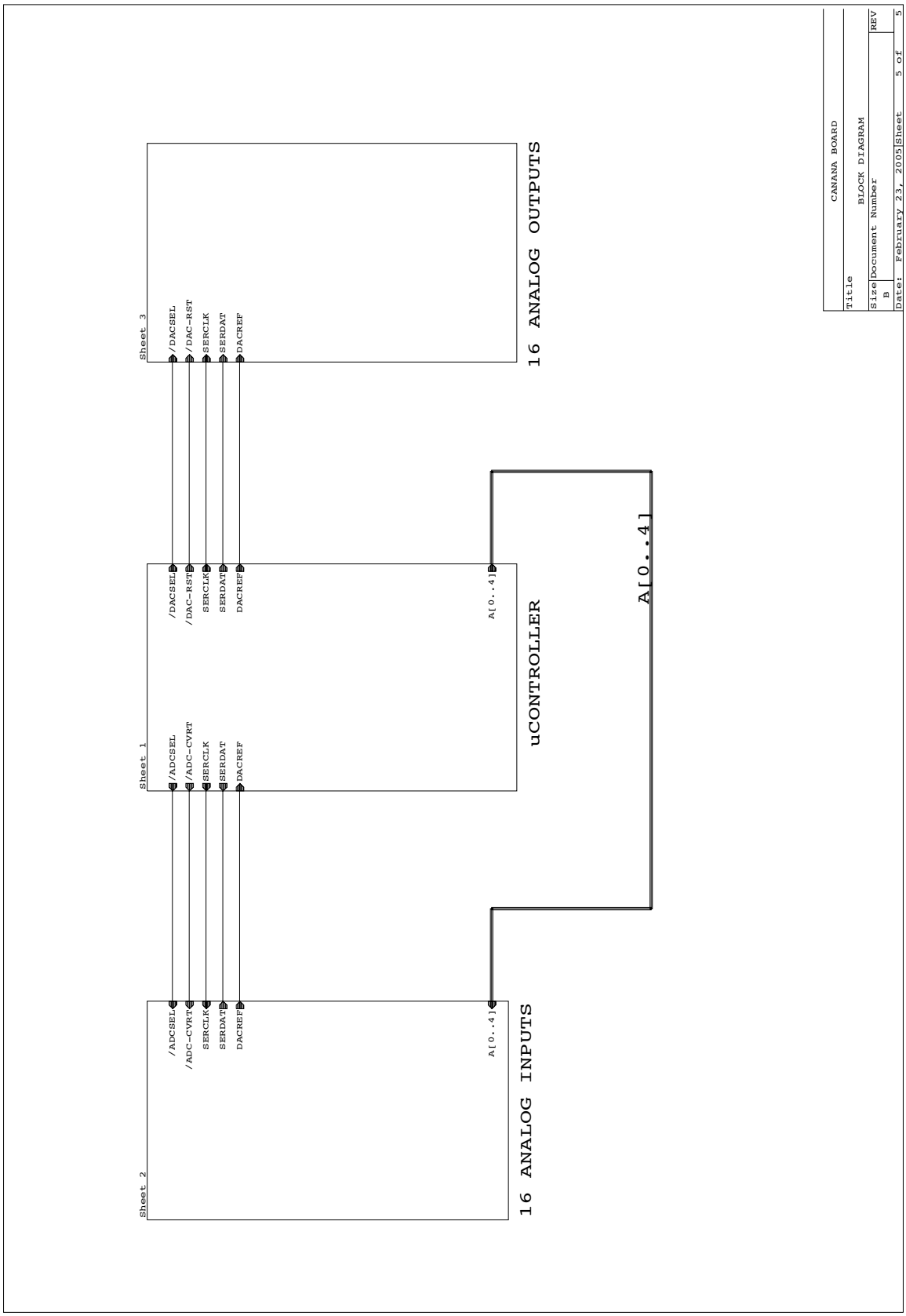
2 DB-9 male connectors are used for connection to the CAN bus. The 2 connectors are tied pin-to-pin.
Connections:
2: CAN Low
7: CAN High
3: CAN Gnd
5: +24 Volt
9: 0 Volt

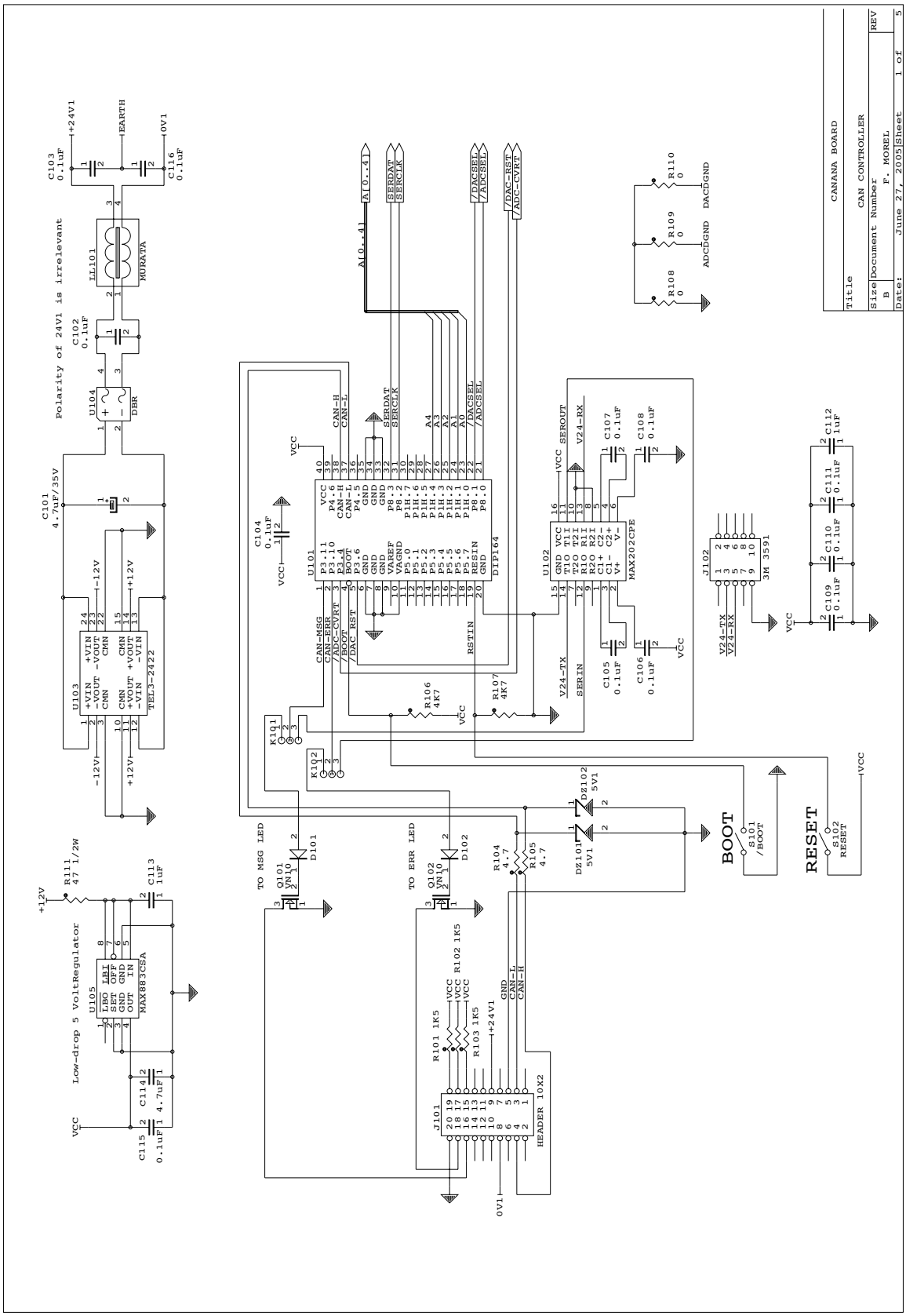
2 high-density SMC 50-pin connectors are used for analog I/O.
Connections:

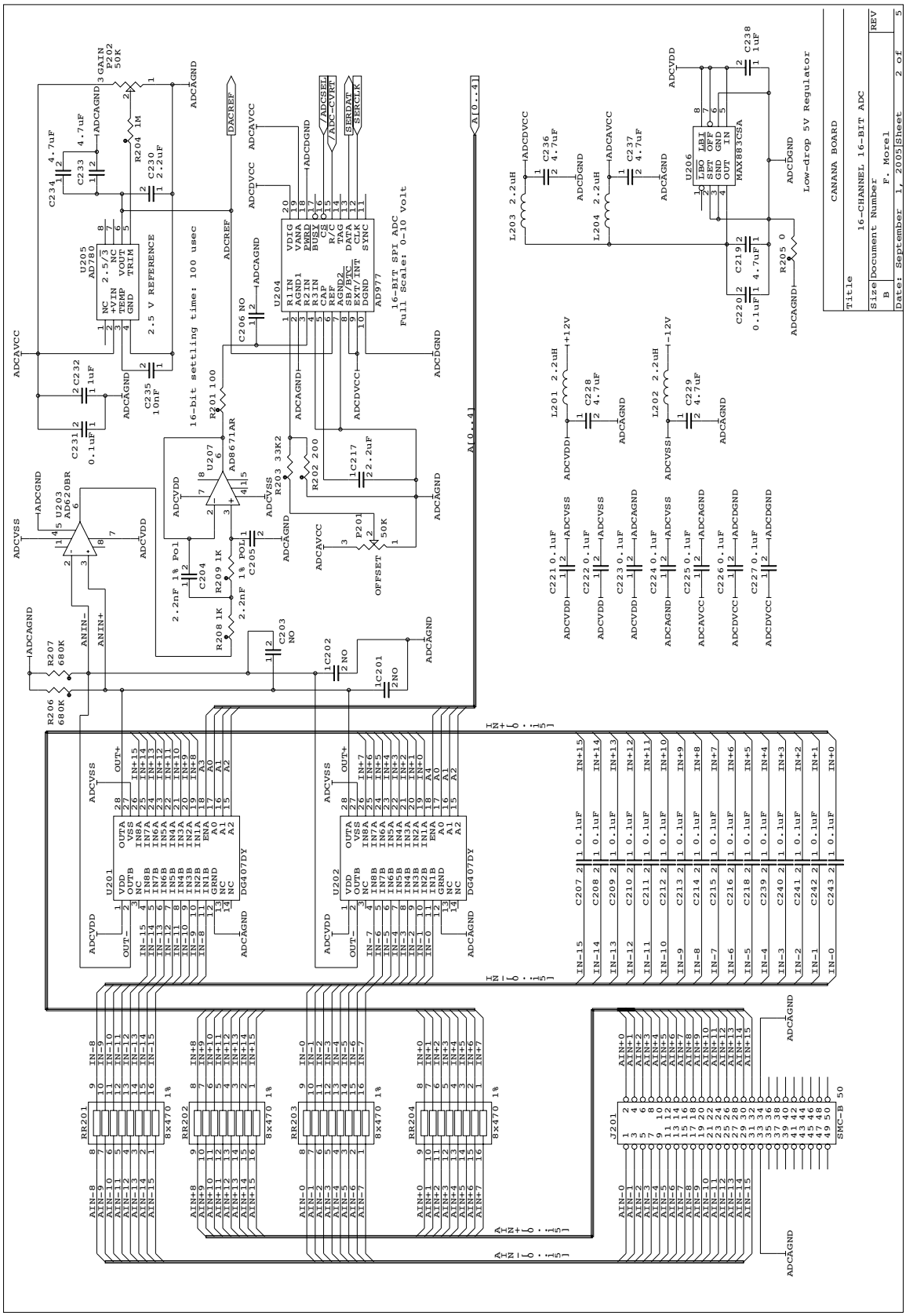
Pin number	ADC input connector	DAC output connector
1	IN0-	GND
2	IN0+	OUT0
3	IN1-	GND

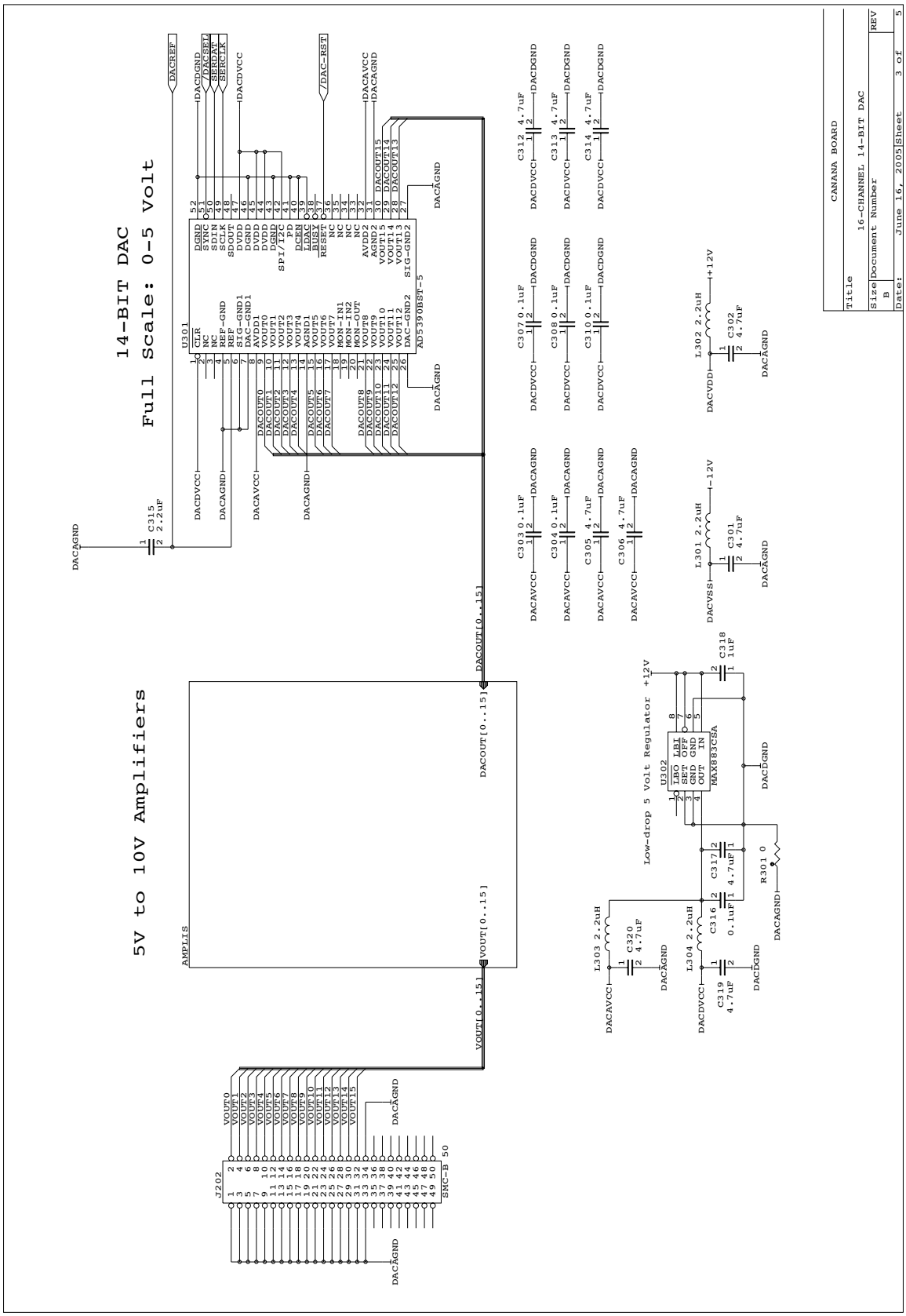
4	IN1+	OUT1
5	IN2-	GND
6	IN2+	OUT2
7	IN3-	GND
8	IN3+	OUT3
9	IN4-	GND
10	IN4+	OUT4
11	IN5-	GND
12	IN5+	OUT5
13	IN6-	GND
14	IN6+	OUT6
15	IN7-	GND
16	IN7+	OUT7
17	IN8-	GND
18	IN8+	OUT8
19	IN9-	GND
20	IN9+	OUT9
21	IN10-	GND
22	IN10+	OUT10
23	IN11-	GND
24	IN11+	OUT11
25	IN12-	GND
26	IN12+	OUT12
27	IN13-	GND
28	IN13+	OUT13
29	IN14-	GND
30	IN14+	OUT14
31	IN15-	GND
32	IN15+	OUT15
33	GND	GND
34	GND	GND
35-50	UNUSED	UNUSED

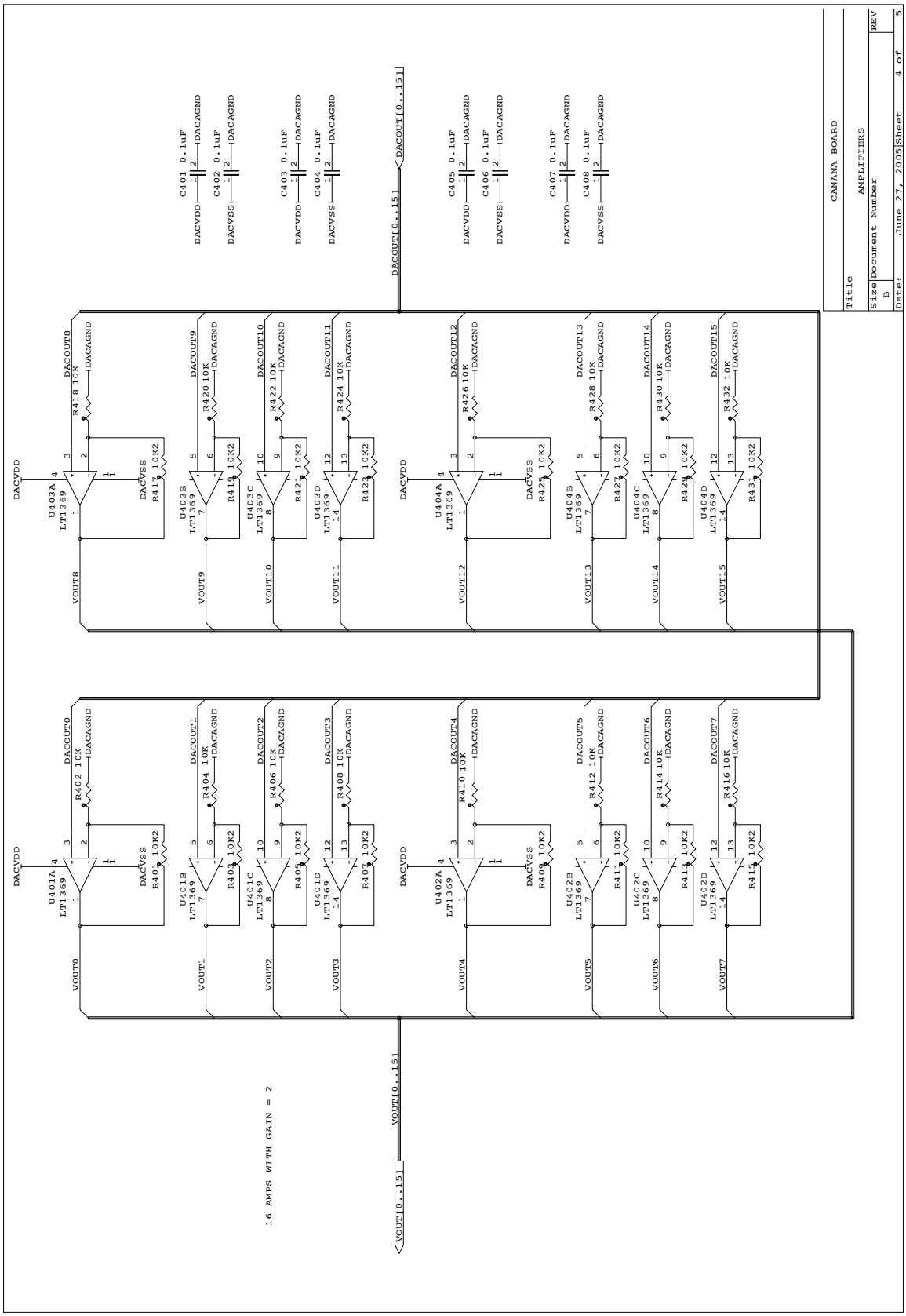
1.2.3 Controller schematics


CANANA BOARD
Title
BLOCK DIAGRAM
Size Document Number REV
B
Date: February 23, 2005 Sheet 5 of 5
16 ANALOG OUTPUTS
Sheet 3
/DACSEL
/DAC-RST
SERCLK
SERDAT
DACREF
uCONTROLLER
Sheet 1
/ADCSEL
/ADC-CVRT
SERCLK
SERDAT
DACREF
A[0..4]
A[0..4]
16 ANALOG INPUTS
Sheet 2
/ADCSEL
/ADC-CVRT
SERCLK
SERDAT
DACREF
A[0..4]


CANANA BOARD
CAN CONTROLLER
F. MOREL
June 27, 2005
Polarity of 24V1 is irrelevant
Low-drop 5 VoltRegulator
MAX883CSA
TEL3-2422
MURATA
DBR
MAX202CPE
DIP164
HEADER 10X2
TO MSG LED
TO ERR LED
BOOT
RESET
CAN-H
CAN-L
SERDAT
SERCLK
/DACSEL
/ADCSEL
/DAC-RST
/ADC-CVRT
V24-TX
V24-RX
SERIN
SEROUT
RSTIN
CAN-MSG
CAN-ERR
/ADC-CVRT
/DAC RST
/BOOT
VAREF
VAGND
ADCDGND
DACDGND
+24V1
0V1
EARTH
VCC
+12V
-12V


CANANA BOARD
16-CHANNEL 16-BIT ADC
Low-drop 5V Regulator
16-bit settling time: 100 usec
2.5 V REFERENCE
16-BIT SPI ADC
Full Scale: 0-10 Volt
AD977
AD8671AR
AD620BR
MAX883CSA
DG407DY
SMC-B 50
8x470 1%
F. Morel
September 1, 2005
2 of 5


14-BIT DAC
Full Scale: 0-5 Volt
5V to 10V Amplifiers
CANANA BOARD
16-CHANNEL 14-BIT DAC
U301
AD5390BST-5
U302
MAX883CSA
Low-drop 5 Volt Regulator
AMPLIS
J202
SMC-B 50
DACAGND
DACDGND
DACVCC
DACREF
DACVOUT[0..15]
VOUT[0..15]
+12V
-12V


CANANA BOARD
AMPLIFIERS
Size Document Number REV
B
Date: June 27, 2005 Sheet 4 of 5
16 AMPS WITH GAIN = 2
VOUT[0..15]
DACOUT[0..15]
C401 0.1uF
C402 0.1uF
C403 0.1uF
C404 0.1uF
C405 0.1uF
C406 0.1uF
C407 0.1uF
C408 0.1uF
DACVDD
DACVSS
DACAGND

1.2.4 Adjustments:

The ADC is iram-adjusted for offset and gain. This initial adjustment is made with 2 potentiometers, which should not be modified. An automatic software adjustment is available; the offset and gain correction parameters for each channel are stored in the EEPROM of the controller.
A similar DAC software adjustment procedure can then be run, using both DAC and ADC, connected pin-to-pin through the front-panel connectors with a turn-around cable.
These commands are **protected** and can be performed only with the right software key. They are normally used once, and should not be necessary later. They require anyway special equipment, a precision voltage generator and a precision voltmeter.
The calculated corrections stored in the EEPROM are immediately applied to the DAC outputs and to the ADC inputs. They will be recalled and applied upon each reset or startup. Sending CAN message [CAN-ID + 0x190] allows temporary (until next reset) disabling these corrections.

1.2.4.1 ADC offset adjustment:

This adjustment is to be made first, prior to ADC Gain adjustment.
Connect all ADC inputs to a 10.0 milliVolt source.
Send to the board the CAN message [CAN-ID+0x190], with DLC = 1 (dummy data).
Send to the board the CAN message [CAN-ID+0x1A0], with DLC = 4 and bytes[0-3] containing the software key.
Reset the board.

1.2.4.2 ADC gain adjustment:

This adjustment should be made after ADC Offset adjustment only.
Connect all ADC inputs to a 9.9900 Volt source.
Send to the board the CAN message [CAN-ID + 0x1B0], with DLC = 4 and bytes[0-3] containing the software key.
Reset the board.

1.2.4.3 DAC offset and gain Adjustment:

This adjustment should be made after the ADC adjustments only.
Connect the special pin-to-pin cable, thus connecting each DAC output [x] to the mating ADC input[x].
Send then the CAN message [CAN-ID+0x1D0], with DLC=4 and bytes[0-3] containing the software key, to the board. This will perform automatically the offset and gain calibration of the DAC, using the ADC as a reference. The calibration may last about one second.
After calibration, all 16 DAC output voltages should be 5.000 Volt +/- 1.22 mV. This helps for checkout.
Reset the board.

1.2.4.4 ADC and DAC correction parameters:

The calculated corrections are **stored** in a non-volatile EEPROM, and can be read:
Gain/offset corrections are readable for each channel [0-15] sending following CAN messages:
ADC correction parameters readout: [CAN-ID + 0x1C0 + Channel Number].
DAC correction parameters readout: [CAN-ID + 0x1E0 + Channel Number].

The reply message is 7 bytes long and has following structure:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
Gain[31-24]	Gain[23-16]	Gain[15-8]	Gain[7-0]	Offset[15-8]	Offset[7-0]	CAN Error Report

Offset correction format: 16-bit signed (-32768 to +32767), LSB = 1.
Offset correction parameter should always be in the range +/- 15 (between 0xFFF1 and 0x000F).

Gain correction format: 32-bit unsigned, **fix** format (0 to 65536), LSB = 1/ 65536.
Value 0x00 00 00 00 = 0.00000 Decimal
Value 0x00 00 FF F9 = 0.99989 Decimal
Value 0x00 01 00 00 = 1.00000 Decimal
Value 0x00 01 00 06 = 1.00009 Decimal
Value 0xFF FF FF FF = 65536 Decimal

Gain correction parameter should always be close to 1 (between 0x00 00 F0 00 and 0x00 01 11 11).
All these corrections are automatically fetched from the EEPROM upon startup or reset. They will be applied by default, and will be suspended (until reset) if the command [CAN-ID + 0x190] is sent to the board.

1.2.5 Performances:

DAC/ADC offset and gain errors (including the analog chain errors) are within +/- 2 LSB after calibration. For the ADC, this calibration is made by averaging 16 readouts of the same input.
The ADC and the DAC use the same external high precision 3 ppm/C reference (AD780).
The ADC has a 16-bit resolution; the LSB value is thus 153 microVolt. The conversion time is 32 microseconds when reading repeatedly the same channel. It will increase to 110 microseconds when reading randomly different channels. The CAN frames (2, one for command and one for reply) introduce a delay of the same order.
The ADC has a guaranteed linearity of +/- 2 LSB and drift of +/-2 ppm/C (200 microVolt for 10C). The input amplifiers introduce a 3 microVolt/C error. The total worst case error is in the range (+/- 3 LSB), which means that **the maximum ADC readout error will be +/- 500 microVolt in a 10C temperature range.**
The DAC is a 14-bit device, with a 610 microVolt LSB. The settling time of the outputs is around 100 microseconds, for a 10 Volt step. A complete output setting will thus take 230 microseconds (one CAN frame plus one conversion). The DAC has a guaranteed relative accuracy of +/- 3 LSB, and neglictible drift. **The maximum DAC output voltage error will be +/- 2 mV.**
Here is a table of the DAC outputs and the ADC inputs voltages. For that measurement, DACs and ADCs are connected in a back-to-back configuration. These voltages are compared to the actual value, read with a precision H-P voltmeter:

Value written to DAC	Actual (measured) value	Value read from ADC
0 mV	1.0 mV	1.0 mV
9.7 mV	10.0 mV	10.1 mV
39.0 mV	39.2 mV	39.2 mV
78.1 mV	78.6 mV	78.6 mV
0.3125 V	0.3131 V	0.3131 V
0.6250 V	0.6252 V	0.6251 V
1.2500 V	1.2504 V	1.2503 V
2.5000 V	2.5011 V	2.5011 V
5.0000 V	5.0013 V	5.0011 V
9.9902 V	9.9922 V	9.9919 V
10.0000 V	10.0000 V	10.0000 V

The full scale is adjusted at 10 Volt for DAC and ADC.

N.B:
The ADC input differential impedance is > 1Mohm, its common mode voltage is +/- 10 Volt. This implies that the source impedance of the measured voltage must be < 15 Ohm to guarantee a valid LSB readout.
The DAC outputs share the same ground. They should always be loaded with R > 10 kOhm, in order to avoid cable-induced offset errors or overloading the output amplifiers.

SOFTWARE:

1.3 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 command message.

1.4 CANANA Background task:

The controller acts a **slave** device: It receives CAN messages from the CAN master PC, does the requested job, 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.

1.5 CAN commands used by the CANANA board:

1.5.1 Summary of Control and Monitor points:

"i" [0-15] is the channel number.

Function	CAN ID	Data Size	Description
Get Analog Input[i]	NODE_ID+0x100+i	3	Get analog input [i]
Set Analog Output[i]	NODE_ID+0x110+i	2	Set Analog output [i]
Get Analog Output[i]	NODE_ID+0x120+i	3	Get Analog output [i]
Set corrections OFF	NODE_ID+0x190	1	Disable ADC/ADC corrections
Set ADC Offset auto-calibration	NODE_ID+0x1A0	4	Set ADC Offset calibration start
Set ADC Gain auto-calibration	NODE_ID+0x1B0	4	Start ADC Gain calibration
Get ADC Channel[i] correction parameters	NODE_ID+0x1C0+i	8	Get ADC Channel[i] gain and offset calculated corrections
Set DAC auto-calibration	NODE_ID+0x1D0	4	Start DAC Offset and Gain calibration
Get DAC Channel[i]correction parameters	NODE_ID+0x1E0+i	7	Get DAC Channel[i] gain and offset calculated corrections
Set Serial Number	NODE_ID+0x1FD	8	Set 64-bit board Serial Number
Set NODE_ID	NODE_ID+0x1FE	8	Set the new board NODE_ID

Set reset	NODE_ID+0x1FF	1	Reset the board

1.5.2 Detailed Control points:

Function	CAN ID	Data Size	Description
Set Analog Output[i]	NODE_ID+0x110+i	2	Set 14-bit Analog output [i] requested value. MAX value: 0x3FFF 0x3FFF == 10.000 Volt Byte[0,1]: data unsigned value
Set correction parameters OFF	NODE_ID+0x190	1	Disable ADC/ADC corrections, applies until next Reset. All Offset corrections = 0 All Gain corrections = 1.0
Set ADC Offset auto-calibration	NODE_ID+0x1A0	4	Start ADC Offset calibration. Adjusts automatically all 16 channels. All ADC inputs must first be connected to a 10.0 mVolt reference voltage. Bytes[0-1]: Security key1. Bytes[2-3]: all zeros
Set ADC Gain auto-calibration	NODE_ID+0x1B0	4	Starts ADC Gain calibration. Adjusts automatically all 16 channels. All ADC inputs must first be connected to a 9.9900 Volt reference voltage. Bytes[0-1]: Security key1. Bytes[2-3]: all zeros
Set DAC auto-calibration	NODE_ID+0x1D0	4	Start DAC Offset and Gain calibration. Each DAC output channel must first be connected to the same ADC input channel. Bytes[0-1]: Security key1. Bytes[2-3]: all zeros
Set Serial Number	NODE_ID+0x1FD	8	Set 64-bit board Serial Number Byte[0-1]: Security key1. Byte[2-7]: New Serial Number.
Set NODE_ID	NODE_ID+0x1FE	8	Set the new board NODE_ID Byte[0-3]: Security key2. Byte[4-7]: New NODE_ID.
Set reset	NODE_ID+0x1FF	1	Reset the board, similar to shutdown/restart.

1.5.3 Detailed monitor points:

Function	CAN ID	Data Size	Description
Get Analog Input[i]	NODE_ID+0x100+i	3	Get analog input [i] actual value: Max value: 0xFFFF 0xFFFF == 10.000 V Byte[0-1]: data unsigned value. Byte[2]: transaction report: Bit[2]: CAN Error.
Get Analog Output[i]	NODE_ID+0x120+i	3	Get Analog output[i] requested value:

			Max value: 0x3FFF 0x3FFF == 10.000 V Byte[0-1]: data unsigned value. Byte[2]: transaction report: Bit[2]: CAN Error.
Get ADC Channel[i] correction parameters	NODE_ID+0x1C0+i	7	Get ADC Channel[i] gain and offset calculated corrections. Bytes[0-3]: Gain correction. Unsigned, LSB = 1/65536. Bytes[4-5]: Offset correction. Signed. Byte[6]: Transaction report. Bit[2]: CAN Error.
Get DAC Channel[i]correction parameters	NODE_ID+0x1E0+i	7	Get DAC Channel[i] gain and offset calculated corrections. Bytes[0-3]: Gain correction. Unsigned, LSB = 1/65536. Bytes[[4-5]: Offset correction. Signed. Byte[6]: Transaction report. Bit[2]: CAN Error.

1.6 CAN messages used for Receiver control:

See document “CANpdBNG”