IRAM-COMP-004

Revision: 3
10/27/2008

Contact Author

Institut de RadioAstronomie Millimétrique

PdBNG CAN Interface

Owner Alain Perrigouard (perrigou@iram.fr)

Keywords: CAN, PdBNG receivers

Approved by:	Date:	Signature:
A.Perrigouard	May 10,2005	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS
3	October 2008	A.Perrigouard	Section 16	

Content

1 Introduction

The CAN bus in use for monitor and control the Plateau de Bure instruments consists of the CAN 2.0B variant and a non-standard higher level protocol defined in the document IRAM-COMP-003 “PdB CAN Specification”.
Here after there is summary of the monitor and control points with their CAN Ids, data sizes and descriptions.

2 Bias Junction

Originally the bus I2C is in use for monitoring and controlling the Bias junction. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

2.1 Summary of Control and Monitor Points

Name	CAN ID	Data Size	Description
SET_REFERENCE_JUNCTION1V	04 04 01 10	2	Set junction 1V reference
SET_REFERENCE_JUNCTION1H	04 04 01 14	2	Set junction 1H reference
SET_REFERENCE_JUNCTION3V	04 04 01 18	2	Set junction 3V reference
SET_REFERENCE_JUNCTION3H	04 04 01 1C	2	Set junction 3H reference
SET_REFERENCE_REGISTER_B1_B3	04 04 01 12	1	Set reference register
GET_ACTUAL_CURRENT_JUNCTION1V	04 04 01 11	3	Get junction 1V actual current
GET_ACTUAL_CURRENT_JUNCTION1H	04 04 01 19	3	Get junction 1H actual current
GET_ACTUAL_CURRENT_JUNCTION3V	04 04 01 15	3	Get junction 3V actual current
GET_ACTUAL_CURRENT_JUNCTION3H	04 04 01 1D	3	Get junction 3Hactual current
GET_ACTUAL_VOLTAGE_JUNCTION1V	04 04 01 13	3	Get junction 1V actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION1H	04 04 01 1B	3	Get junction 1H actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION3V	04 04 01 17	3	Get junction 3V actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION3H	04 04 01 1F	3	Get junction 3H actual voltage
GET_REFERENCE_JUNCTION1V	04 04 01 11	3	Get junction 1V reference
GET_REFERENCE_JUNCTION1H	04 04 01 19	3	Get junction 1H reference
GET_REFERENCE_JUNCTION3V	04 04 01 15	3	Get junction 3V reference
GET_REFERENCE_JUNCTION3H	04 04 01 1D	3	Get junction 3H reference
GET_REFERENCE_REGISTER_B1_B3	04 04 01 13	2	Get reference register
SET_REFERENCE_JUNCTION2V	04 04 01 20	2	Set junction 2V reference
SET_REFERENCE_JUNCTION2H	04 04 01 24	2	Set junction 2H reference
SET_REFERENCE_JUNCTION4V	04 04 01 28	2	Set junction 4V reference
SET_REFERENCE_JUNCTION4H	04 04 01 2C	2	Set junction 4H reference
SET_REFERENCE_REGISTER_B2_B4	04 04 01 22	1	Set reference register
GET_ACTUAL_CURRENT_JUNCTION2V	04 04 01 21	3	Get junction 2V actual current

GET_ACTUAL_CURRENT_JUNCTION2H	04 04 01 29	3	Get junction 2H actual current
GET_ACTUAL_CURRENT_JUNCTION4V	04 04 01 25	3	Get junction 4V actual current
GET_ACTUAL_CURRENT_JUNCTION4H	04 04 01 2D	3	Get junction 4H actual current
GET_ACTUAL_VOLTAGE_JUNCTION2V	04 04 01 23	3	Get junction 2V actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION2H	04 04 01 2B	3	Get junction 2H actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION4V	04 04 01 27	3	Get junction 4V actual voltage
GET_ACTUAL_VOLTAGE_JUNCTION4H	04 04 01 2F	3	Get junction 4H actual voltage
GET_REFERENCE_JUNCTION2V	04 04 01 21	3	Get junction 2V reference
GET_REFERENCE_JUNCTION2H	04 04 01 29	3	Get junction 2H reference
GET_REFERENCE_JUNCTION4V	04 04 01 25	3	Get junction 4V reference
GET_REFERENCE_JUNCTION4H	04 04 01 2D	3	Get junction 4H reference
GET_REFERENCE_REGISTER_B2_B4	04 04 01 23	2	Get reference register

The detailed descriptions of the CAN messages for the junctions 2V, 2H, 4V and 4H, and for the reference register B2_B4 are exactly the same as for the junctions 1V, 1H, 4V and 3H, and for the reference register B1_B3. Replace 1V by 2V, 1H by 2H, 3V by 4V, 3H by 4H, B1_B3 by B2_B4.

Convenience control and monitor points:

GET_STATUS_REGISTER_B1_B3	04 04 02 00	2	Get the status register for junctions 1V, 1H, 3V and 3H
GET_STATUS_REGISTER_B2_B4	04 04 02 02	2	Get the status register for junctions 2V, 2V, 4V and 4H
SET_JUNCTION1V_REFERENCE	04 04 02 10	2	Set junction 1V reference
GET_JUNCTION1V_REFERENCE	04 04 02 11	3	Get junction 1V reference
GET_JUNCTION1V_ACTUAL_VOLTAGE	04 04 02 12	3	Get junction 1V actual voltage
GET_JUNCTION1V_ACTUAL_CURRENT	04 04 02 13	3	Get junction 1V actual current
SET_JUNCTION1H_REFERENCE	04 04 01 18	2	Set junction 1H reference
GET_JUNCTION1H_REFERENCE	04 04 01 19	3	Get junction 1H reference
GET_JUNCTION1H_ACTUAL_VOLTAGE	04 04 01 1A	3	Get junction 1H actual voltage
GET_JUNCTION1H_ACTUAL_CURRENT	04 04 01 1B	3	Get junction 1H actual current
SET_JUNCTION2V_REFERENCE	04 04 02 20	2	Set junction 2V reference
GET_JUNCTION2V_REFERENCE	04 04 02 21	3	Get junction 2V reference
GET_JUNCTION2V_ACTUAL_VOLTAGE	04 04 02 22	3	Get junction 2V actual voltage
GET_JUNCTION2V_ACTUAL_CURRENT	04 04 02 23	3	Get junction 2V actual current
SET_JUNCTION2H_REFERENCE	04 04 02 28	2	Set junction 2H reference
GET_JUNCTION2H_REFERENCE	04 04 02 29	3	Get junction 2H reference
GET_JUNCTION2H_ACTUAL_VOLTAGE	04 04 02 2A	3	Get junction 2H actual voltage
GET_JUNCTION2H_ACTUAL_CURRENT	04 04 02 2B	3	Get junction 2H actual current
SET_JUNCTION3V_REFERENCE	04 04 02 30	2	Set junction 3V reference
GET_JUNCTION3V_REFERENCE	04 04 02 31	3	Get junction 3V reference
GET_JUNCTION3V_ACTUAL_VOLTAGE	04 04 02 32	3	Get junction 3V actual voltage
GET_JUNCTION3V_ACTUAL_CURRENT	04 04 02 33	3	Get junction 3V actual current
SET_JUNCTION3H_REFERENCE	04 04 02 38	2	Set junction 3H reference
GET_JUNCTION3H_REFERENCE	04 04 02 39	3	Get junction 3H reference
GET_JUNCTION3H_ACTUAL_VOLTAGE	04 04 02 3A	3	Get junction 3H actual voltage
GET_JUNCTION3H_ACTUAL_CURRENT	04 04 02 3C	3	Get junction 3H actual current
SET_JUNCTION4V_REFERENCE	04 04 02 40	2	Set junction 4V reference

GET_JUNCTION4V_REFERENCE	04 04 02 41	3	Get junction 4V reference
GET_JUNCTION4V_ACTUAL_VOLTAGE	04 04 02 42	3	Get junction 4V actual voltage
GET_JUNCTION4V_ACTUAL_CURRENT	04 04 02 43	3	Get junction 4V actual current
SET_JUNCTION4H_REFERENCE	04 04 02 48	2	Set junction 4H reference
GET_JUNCTION4H_REFERENCE	04 04 02 49	3	Get junction 4H reference
GET_JUNCTION4H_ACTUAL_VOLTAGE	04 04 02 4A	3	Get junction 4H actual voltage
GET_JUNCTION4H_ACTUAL_CURRENT	04 04 02 4B	3	Get junction 4H actual current

Receiver motors:

Name	CAN ID	Data Size	Description
SET_MIXER_BACKSHORT_V_BAND1_a	04 10 01 xx	1 or 2	See description below
SET_MIXER_BACKSHORT_H_BAND1_a	04 14 01 xx	1 or 2	See description below
SET_MIXER_BACKSHORT_V_BAND2_a	04 18 01 xx	1 or 2	See description below
SET_MIXER_BACKSHORT_H_BAND2_a	04 1C 01 xx	1 or 2	See description below
SET_MIXER_BACKSHORT_V_BAND3_a	04 20 01 xx	1 or 2	See description below
SET_MIXER_BACKSHORT_H_BAND3_a	04 24 01 xx	1 or 2	See description below
GET_MIXER_BACKSHORT_V_BAND1_b	04 10 01 yy	3 or 4	See description below
GET_MIXER_BACKSHORT_H_BAND1_b	04 14 01 yy	3 or 4	See description below
GET_MIXER_BACKSHORT_V_BAND2_b	04 18 01 yy	3 or 4	See description below
GET_MIXER_BACKSHORT_H_BAND2_b	04 1C 01 yy	3 or 4	See description below
GET_MIXER_BACKSHORT_V_BAND3_b	04 20 01 yy	3 or 4	See description below
GET_MIXER_BACKSHORT_H_BAND3_b	04 24 01 yy	3 or 4	See description below

See paragraph 7.4 for a definition of the control and monitor terms a and b and for the CAN ID offsets xx and yy.

2.2 Control Points in Detail

Name	SET_REFERENCE_JUNCTION1V
CAN ID	04 04 01 10
Description	Set junction 1V reference, voltage or current (see SET_REFERENCE_REGISTER)
Data	2 bytes, unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current)

Name	SET_REFERENCE_JUNCTION1H
CAN ID	04 04 01 18
Description	Set junction 1H reference, voltage or current (see SET_REFERENCE_REGISTER)
Data	2 bytes, unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current)

Name	SET_REFERENCE_JUNCTION3V

CAN ID	04 04 01 14
Description	Set junction 3V reference voltage or current (see SET_REFERENCE_REGISTER)
Data	2 bytes, unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current)

Name	SET_REFERENCE_JUNCTION3H
CAN ID	04 04 01 1C
Description	Set junction 3H reference voltage or current (see SET_REFERENCE_REGISTER)
Data	2 bytes, unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current)

Name	SET_REFERENCE_REGISTER_B1_B3
CAN ID	04 04 01 12
Description	Reference register. For each junction the register indicates the type of the reference, voltage (bit set to 0) or current (bit set to 1) Set as well the protection of the junctions. Important: When bit 7, readReferenceRegister flag, is set to 1, the other bits are not written into the reference register. Bit 7 should be equal to 0 in order to change the other bits of the reference register. Bit 7 is set to 1 in order to be able to read the reference currents or voltages.
Data	1 byte: Bit [0]: not used Bit [1]: 0 = Junction 1V reference voltage 1 = Junction 1H reference current Bit [2]: 0 = Junction 1H reference voltage 1 = Junction 1H reference current Bit [3]: 0 = Junction 3V reference voltage 1 = Junction 3V reference current Bit [4]: 0 = Junction 3H reference voltage 1 = Junction 3H reference current Bit [5]: 0 = junctions protected 1 = junctions non protected Bit [6]: 1 ADC calibration sequence Bit [7]: readReferenceRegister flag

Name	SET_REFERENCE_REGISTER_B2_B4
CAN ID	04 04 01 22
Description	See SET_REFERENCE_REGISTER_B1_B3

And the convenience control points:

Name	SET_JUNCTION1V_REFERENCE
CAN ID	04 04 02 10
Description	Set junction 1V reference voltage or current (see SET_REFERENCE_REGISTER). Identical to 04 04 01 10.

Data	2 bytes, unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current)

The detailed descriptions of the CAN messages for the junctions 1H, 2V, 2H, 3V, 3H, 4V and 4H are similar.
The following table gives the pair of CAN ID's which are identical and which set the same references:

SET_JUNCTION1V_REFERENCE	04 04 02 10	04 04 01 10	SET_REFERENCE_JUNTION1V
SET_JUNCTION1H_REFERENCE	04 04 02 18	04 04 01 18	SET_REFERENCE_JUNTION1H
SET_JUNCTION2V_REFERENCE	04 04 02 20	04 04 01 20	SET_REFERENCE_JUNTION2V
SET_JUNCTION2H_REFERENCE	04 04 02 28	04 04 01 28	SET_REFERENCE_JUNTION2H
SET_JUNCTION3V_REFERENCE	04 04 02 30	04 04 01 14	SET_REFERENCE_JUNTION3V
SET_JUNCTION3H_REFERENCE	04 04 02 38	04 04 01 1C	SET_REFERENCE_JUNTION3H
SET_JUNCTION4V_REFERENCE	04 04 02 40	04 04 01 24	SET_REFERENCE_JUNTION4V
SET_JUNCTION4H_REFERENCE	04 04 02 48	04 04 01 2C	SET_REFERENCE_JUNTION4H

2.3 Monitor Points in Detail

Name	GET_ACTUAL_CURRENT_JUNCTION1V
CAN ID	04 04 01 11
Description	Get junction 1V actual current if readReferenceRegister flag has been set to 0
Data	3 bytes Bytes [0,1]: Reading, 2's complement signed value = *data* To compute the actual current (in Ampere) Ijunc = (*data* * 5.0 / 0x4000) – Vjunc * (*Rpar* + *Rtc*) / (*Rpar* * *Rtc*) *Rtc* = 25000 *Rpar* = 100 **Warning for junction in band3, Rpar = 10000** Vjunc is the actual voltage of the junction, in Volt (see GET_ACTUAL_VOLTAGE_JUNCTION1V) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_ACTUAL_CURRENT_JUNCTION1H
CAN ID	04 04 01 19
Description	Get junction 1H actual current if readReferenceRegister flag has been set to 0
Data	See GET_ACTUAL_CURRENT_JUNCTION_1V

Name	GET_ACTUAL_CURRENT_JUNCTION3V
CAN ID	04 04 01 15
Description	Get junction 3V actual current if readReferenceRegister flag has been set to 0
Data	See GET_ACTUAL_CURRENT_JUNCTION_1V **but** *Rpar* = 10000

Name	GET_ACTUAL_CURRENT_JUNCTION3H
CAN ID	04 04 01 1D
Description	Get junction 3H actual current if readReferenceRegister flag has been set to 0
Data	See GET_ACTUAL_CURRENT_JUNCTION_1V **but** *Rpar* = 10000

Name	GET_ACTUAL_VOLTAGE_JUNCTION1V
CAN ID	04 04 01 13
Description	Get junction 1V actual voltage if readReferenceRegister flag has been set to 0
Data	3 bytes Bytes [0,1], 2's complement signed value = *data* To compute the actual voltage (in Volt) Vjunc = (*data** 5.0) / (*GAIN* * 0x4000) with *GAIN* = 500 Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

The detailed description for GET_ACTUAL_VOLTAGE_1H, 2V, 2H, 3V, 3H are similar.

Name	GET_REFERENCE_JUNCTION1V
CAN ID	04 04 01 11
Description	Get junction 1V reference, voltage or current (see SET_REFERENCE_REGISTER), and if readReferenceRegister flag has been set to 1
Data	3 bytes Bytes [0,1], unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_REFERENCE_JUNCTION1H
CAN ID	04 04 01 19
Description	Get junction 1H reference, voltage or current (see SET_REFERENCE_REGISTER), and if readReferenceRegister flag has been set to 1
Data	3 bytes Bytes [0,1], unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_REFERENCE_JUNCTION3V
CAN ID	04 04 01 15
Description	Get junction 3V reference, voltage or current (see SET_REFERENCE_REGISTER), and if readReferenceRegister flag has been set to 1
Data	3 bytes Bytes [0,1], unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_REFERENCE_JUNCTION3H
CAN ID	04 04 01 1D
Description	Get junction 3H reference, voltage or current (see SET_REFERENCE_REGISTER), and if readReferenceRegister flag has been set to 1
Data	3 bytes Bytes [0,1], unsigned value 0x8000=10mV (reference voltage) 0x8000=100uA (reference current) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_REFERENCE_REGISTER_B1_B3
CAN ID	04 04 01 13
Description	Get reference register if readReferenceRegister flag has been set to 1. For each junction the register indicates the type of the reference, voltage (bit set to 0) or current (bit set to 1)
Data	2 bytes Byte [0]: Bit [0]: meaningless Bit [1]: 0 = Junction 1V reference voltage 1 = Junction 1V reference current Bit [2]: 0 = Junction 1H reference voltage 1 = Junction 1H reference current Bit [3]: 0 = Junction 3V reference voltage 1 = Junction 3V reference current Bit [4]: 0 = Junction 3H reference voltage 1 = Junction 3H reference current Bit [5]: 0 = junctions protected 1 = junctions non protected Bit [6]: meaningless Bit [7]: meaningless Byte [1]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

And the convenience monitor points:

Name	GET_JUNCTION1V_REFERENCE
CAN ID	04 04 02 11
Description	Get junction 1V reference, voltage or current (see SET_REFERENCE_REGISTER)
Data	3 bytes Bytes [0,1], unsigned value 0x0000= 0mV (reference voltage) 0x8000=10mV (reference voltage) or 0x0000= 0uA (reference current) 0x8000=100uA (reference current) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error No prerequisite.

Name	GET_JUNCTION1V_ACTUAL_VOLTAGE
CAN ID	04 04 02 12
Description	Get junction 1V actual voltage
Data	3 bytes Bytes [0,1], 2's complement signed value. 0x4000=10mV Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error No prerequisite.

Name	GET_JUNCTION1V_ACTUAL_CURRENT
CAN ID	04 04 02 13
Description	Get junction 1V actual current
Data	3 bytes Bytes [0,1]: Reading, 2's complement signed value. 0x4000=100uA Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error No prerequisite.

The detailed descriptions of the CAN messages for the junctions 1H, 2V, 2H, 3V, 3H, 4V and 4H are similar.

Name	GET_STATUS_REGISTER_B1_B3
CAN ID	04 04 02 00
Description	Get reference register. For each junction the register indicates the type of the reference, voltage (bit set to 0) or current (bit set to 1)
Data	2 bytes Byte [0]: Bit [0]: meaningless Bit [1]: 0 = Junction 1V reference voltage 1 = Junction 1V reference current Bit [2]: 0 = Junction 1H reference voltage 1 = Junction 1H reference current Bit [3]: 0 = Junction 3V reference voltage 1 = Junction 3V reference current Bit [4]: 0 = Junction 3H reference voltage 1 = Junction 3H reference current Bit [5]: 0 = junctions protected 1 = junctions non protected Bit [6]: meaningless Bit [7]: meaningless Byte [1]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error No prerequisite.

Name	GET_STATUS_REGISTER_B2_B4
CAN ID	04 04 02 02

Description	Get reference register. For each junction the register indicates the type of the reference, voltage (bit set to 0) or current (bit set to 1)
Data	2 bytes Byte [0]: Bit [0]: meaningless Bit [1]: 0 = Junction 2V reference voltage 1 = Junction 2V reference current Bit [2]: 0 = Junction 2H reference voltage 1 = Junction 2H reference current Bit [3]: 0 = Junction 4V reference voltage 1 = Junction 4V reference current Bit [4]: 0 = Junction 4H reference voltage 1 = Junction 4H reference current Bit [5]: 0 = junctions protected 1 = junctions non protected Bit [6]: meaningless Bit [7]: meaningless Byte [1]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error No prerequisite.

3 Bias HEMT

Originally the bus I2C is in use for monitoring and controlling the Bias HEMT. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

3.1 Summary of Control and Monitor Points

Name	CAN ID	Data Size	Description
SET_CONTROL_REGISTER	04 04 01 50	1	Set control register
SET_CHANNEL_PCF8574A	04 04 01 70	1	Select channel PCF8574A
GET_CONVERTED_DATA	04 04 01 51	3	Get converted data
GET_CHANNEL_PCF8574A	04 04 01 71	2	Get channel PCF8574A

Convenience monitor points:

Name	CAN ID	Data Size	Description
GET_HEMT_1V_STAGE0	04 04 02 90	7	Get HEMT 1V stage0 values
GET_HEMT_1V_STAGE1	04 04 02 91	7	Get HEMT 1V stage1 values
GET_HEMT_1V_STAGE2	04 04 02 92	7	Get HEMT 1V stage2 values
GET_HEMT_1H_STAGE0	04 04 02 93	7	Get HEMT 1H stage0 values

GET_HEMT_1H_STAGE1	04 04 02 94	7	Get HEMT 1H stage1 values
GET_HEMT_1H_STAGE2	04 04 02 95	7	Get HEMT 1H stage2 values
GET_HEMT_2V_STAGE0	04 04 02 96	7	Get HEMT 2V stage0 values
GET_HEMT_2V_STAGE1	04 04 02 97	7	Get HEMT 2V stage1 values
GET_HEMT_2V_STAGE2	04 04 02 98	7	Get HEMT 2V stage2 values
GET_HEMT_2H_STAGE0	04 04 02 99	7	Get HEMT 2H stage0 values
GET_HEMT_2H_STAGE1	04 04 02 9A	7	Get HEMT 2H stage1 values
GET_HEMT_2H_STAGE2	04 04 02 9B	7	Get HEMT 2H stage2 values
GET_HEMT_3V_STAGE0	04 04 02 9C	7	Get all HEMT 3V VDMs
GET_HEMT_3V_STAGE1	04 04 02 9D	7	Get all HEMT 3V IDMs
GET_HEMT_3V_STAGE2	04 04 02 9E	7	Get all HEMT 3V VGMs
GET_HEMT_3H_STAGE0	04 04 02 9F	7	Get all HEMT 3H VDMs
GET_HEMT_3H_STAGE1	04 04 02 A0	7	Get all HEMT 3H IDMs
GET_HEMT_3H_STAGE2	04 04 02 A1	7	Get all HEMT 3H VGMs
GET_HEMT_4V_STAGE0	04 04 02 A2	7	Get HEMT 4V stage0 values
GET_HEMT_4V_STAGE1	04 04 02 A3	7	Get HEMT 4V stage1 values
GET_HEMT_4V_STAGE2	04 04 02 A4	7	Get HEMT 4V stage2 values
GET_HEMT_4H_STAGE0	04 04 02 A5	7	Get HEMT 4H stage0 values
GET_HEMT_4H_STAGE1	04 04 02 A6	7	Get HEMT 4H stage1 values
GET_HEMT_4H_STAGE2	04 04 02 A6	7	Get HEMT 4H stage2 values

3.2 Control Points in Detail

Name	SET_CONTROL_REGISTER
CAN ID	04 04 01 50
Description	Set control register
Data	1 byte 0x82: Standby 0x8C: Start Vdm read conversion 0x9C: Start Idm read conversion 0xA4: Start Vgm read conversion

Name	SET_CHANNEL_PCF8574A
CAN ID	04 04 01 70
Description	Select channel PCF8574A
Data	1 byte Bit [7]: 0 if unit4 selected Bit [6]: 0 if unit3 selected. Bit[7-6] always equal to 11. Bit [5]: 0 if unit2 (2nd HEMT bias box) selected, otherwise 1. Bit [4]: 0 if unit1 (1st HEMT bias box) selected, otherwise 1. Only one unit is selected at the time. Bits [3-0]: not[(Amplifier number) * 3 + (Stage number)] Amplifier number from 0 to 3 Stage number from 0 to 2

3.3 Monitor Points in Detail

Name	GET_CONVERTED_DATA
CAN ID	04 04 01 51
Description	Get converted data (12 bit ADC)
Data	3 bytes Bytes [0,1], 12-bits signed value in bits[15-4] 0x0400 = 5V (Vdm) 0x0400 = 10mA (Idm) 0x0400 = 2.5V (Vgm) Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_CHANNEL_PCF8574A
CAN ID	04 04 01 71
Description	Get channel PCF8574A
Data	2 bytes Byte [0]: Bit [7]: 0 if unit4 selected Bit [6]: 0 if unit3 selected. Bit[7-6] always equal to 11. Bit [5]: 0 if unit2 (2nd HEMT bias box) selected, otherwise 1. Bit [4]: 0 if unit1 (1st HEMT bias box) selected, otherwise 1. Only one unit is selected at the time. Bits [3-0]:not[(Amplifier number) * 3 + (Stage number)] Amplifier number from 0 to 3 Stage number from 0 to 2 Byte [1]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Relation between the HEM Bias box units, the amplifier numbers, the receiver bands and the polarizations:

HEM bias box unit	Amplifier number	Receiver band	Polarization
1	0	1	V
	1	1	H
	2	3	V
	3	3	H
2	0	2	V
	1	2	H
	2	4	V
	3	4	H
3	Not used		
4	Not used		

Convenience monitor points:

Name	CAN ID	Data Size	Description
GET_HEMT_ 1V_STAGE0	04 04 02 90	7	Get all HEMT 1V VDMs Bytes[0,1]: Stage 0 VDM, 12-bits signed value

			in bits[15-4] 0x400 == 5 Volt Bytes[2,3]: Stage 0 IDM, 12-bits signed value in bits[15-4] 0x400 == 10 mA Bytes[4,5]: Stage 0 VGM, 12-bits signed value in bits[15-4] 0x400 == 2.5 Volt Byte 6: I2C transaction report Bit [2]=CAN error Bit [1]=I2C write error Bit [0]=I2C read error

The detailed descriptions of the CAN messages for the channels 1H, 2V, 2H, 3V, 3H, 4V and 4H are similar.

4 Cryostat Temperature

Originally the bus I2C is in use for monitoring and controlling the Cryostat Temperature. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

4.1 Summary of control and Monitor Points

Name	CAN ID	Data Size	Description
SET_CRYO_CONTROL_REGISTER	04 04 01 82	2	Set control register
SET_CRYO_MAX6633_REGISTER	04 04 01 90	1	Set MAX6633 register
GET_ CRYO_TEMPERATURE	04 04 01 81	8	Get converted temperature data
GET_ CRYO_STATUS_REGISTER	04 04 01 83	3	Get status register
GET_CRYO_MAX6633_TEMPERATURE	04 04 01 91	3	Get MAX6633 temperature

4.2 Control Points in Detail

Name	SET_ CRYO_CONTROL_REGISTER	
CAN ID	04 04 01 82	
Description	Set the order register and in particular indicate the number of channel to convert	
Data:	2 bytes Bytes [0,1] = bits [15-0] Bits [14-9] = Command Bits [8-0] = Parameter	
	Command:	Parameter
	0x00 to 0x07: Standby	Memory read start address. 0 to 511
	0x08: Set 1st channel number	Channel number between 0 and 7

	0x09: Set write memory pointer	Memory write conversion address. 0 to 511
	0x10: Set last channel number	Channel number between 0 and 7
	0x11: Set number of samples per channel	Number-1 of samples/channel. 0 to 255
	0x20: Request 1st channel number	Memory read start address. 0 to 511
	0x21: Request memory pointer	Memory read start address. 0 to 511
	0x22: Request last channel number	Memory read start address. 0 to 511
	0x23: Request number of samples/channel	Memory read start address. 0 to 511
	0x28 to 0x2F: Conversion start	Memory read start address. 0 to 511
	0x38 to 0x3F: Soft reset	Memory read start address. 0 to 511

Default value at power on:

1st channel number	0
Memory write conversion address	0
Last channel number	7
Number of samples per channel -1	0
Memory read start address	0

Those values are the standard values for the operations at Plateau de Bure.

Operation:
When a conversion is started, the requested number of samples/channel of the given 1st channel are stored at the addresses starting from the value named "Memory write conversion address". The conversions are stored in 2 bytes words at consecutive addresses. This conversion continues with the next channel up to the last channel and then stops. Each conversion takes 67.114 milliseconds to complete.
The "Memory read start address" is the memory starting address for reading the converted temperatures through the field bus. Although it is set independently of the "Memory write conversion address" it seems reasonable to set both to the same value for normal operations.

Name	SET_CRYO_MAX6633_REGISTER
CAN ID	04 04 01 90
Description	Set the MAXIM 6633 configuration register
Data	1 byte = 0x00: enabled. Default value at power on = 0x01: disabled

4.3 Monitor Points in Detail

Name	GET_ CRYO_TEMPERATURE
CAN ID	04 04 01 81
Description	Get 4 channel values. After the execution the "Memory read start address" is incremented by 8 mod 256.
Data	8 bytes Bytes [0,1]: Bits [15]: 1 = Invalid data, 0 = OK Bits [14-12]: Channel number, 0 to 7 Bits [11-00]: Channel unsigned value from 0x00 to 0xFFF Bytes [2,3]: Bits [15]: 1 = Invalid data, 0 = OK

	Bits [14-12]: Channel number, 0 to 7 Bits [11-00]: Channel unsigned value from 0x00 to 0xFFF Bytes [4,5]: Bits [15]: 1 = Invalid data, 0 = OK Bits [14-12]: Channel number, 0 to 7 Bits [11-00]: Channel unsigned value from 0x00 to 0xFFF Bytes [6,7]: Bits [15]: 1 = Invalid data, 0 = OK Bits [14-12]: Channel number, 0 to 7 Bits [11-00]: Channel unsigned value from 0x00 to 0xFFF

Name	GET_ CRYO_STATUS_REGISTER
CAN ID	04 04 01 83
Description	Get the status register which depends on the last message SET_ CRYO_CONTROL_REGISTER
Data	3 bytes Bytes [0,1] = bits [15-0] Bits [14-9] = Status or last requested command Bits [8-0] = Parameter

Status or last requested command:	Parameter:
0x00 to 0x07: Standby	Current memory read address. 0 to 511
0x08: Set 1st channel number	Current memory read address. 0 to 511
0x09: Set write memory pointer	Current memory read address. 0 to 511
0x10: Set last channel number	Current memory read address. 0 to 511
0x11: Set number of samples per channel	Current memory read address. 0 to 511
0x20: 1st channel number	Channel number between 0 and 7
0x21: Write memory pointer	Memory write conversion address. 0 to 511
0x22: Last channel number	Channel number between 0 and 7
0x23: Number of samples/channel	Number-1 of samples/channel. 0 to 255
0x28 to 0x2F: Conversion start	Current memory read address. 0 to 511
0x38 to 0x3F: Soft reset	Current memory read address. 0 to 511

Byte[2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_CRYO_MAX6633_TEMPERATURE
CAN ID	04 04 01 91
Description	Get the MAXIM 6633 temperature
Data	3 bytes Bytes [0,1] = bits [15-0] Bits [15-3] = temperature. Bit [3], lsb = .0625deg Celcius. Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Connection between temperatures and receiver band

Channel number	Name	Sensor Type	Sensor	Temp. Range (K)
0	Band 1 Temp.	Carbon resistor	R32	1.5 to 100
1	Band 2 Temp.	Carbon resistor	R33	1.5 to 100
2	Band 3 Temp.	Carbon resistor	R41	1.5 to 100
3	Band 4 Temp.	Carbon resistor		1.5 to 100
4	Cryogenerator 77K	Platinum resistor	PT100	50 to 300
5	Cryog. 15K	Carbon resistor	R30	1.5 to 100
6	Cryog. 4K	Carbon resistor	R39	1.5 to 100
7	Cold load	Carbon resistor	T84	1.5 to 100

5 Hot Load Temperature

Originally the bus I2C is in use for monitoring and controlling the Hot Load Temperature. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

5.1 Summary of control and Monitor Points

Name	CAN ID	Data Size	Description
SET_HOT_LOAD_DS620_REGISTER	04 04 01 92	1	Set DS620 register
GET_HOT_LOAD_DS620_TEMPERATURE	04 04 01 93	3	Get the hot load temperature

Convenience monitor point:

GET_HOT_LOAD_TEMPERATURE	04 04 02 B0	3	Get the hot load temperature

5.2 Control Points in Detail

Name	SET_HOT_LOAD_DS620_REGISTER
CAN ID	04 04 01 92
Description	Set the DS620 configuration register
Data	1 byte = 0xAA to be able to read the hot load temperature. This value is incremented by each DS620 reading. As a consequence, this register has to be set to 0xAA each time the hot load temperature is monitored.

5.3 Monitor Points in Detail

Name	GET_HOT_LOAD_DS620_TEMPERATURE
CAN ID	04 04 01 93
Description	Get the hot load temperature (as far the configuration register is set to 0xAA).
Data	3 bytes Bytes [0,1] = bits [15-0] Bit [0], lsb = 1/128 deg Celcius. Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

Name	GET_HOT_LOAD_TEMPERATURE
CAN ID	04 04 02 B0
Description	Get the hot load temperature. It is a compound function which set automatically the configuration register for reading the hot load temperature.
Data	3 bytes Bytes [0,1] = bits [15-0] Bit [0], lsb = 1/128 deg Celcius. Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error

6 Coil currents

Originally the bus I2C is in use for monitoring and controlling the coil currents. Yves Bortolotti has developed the coil currents module. Get from him the applicable documentation. The module controls 4 current channels with the help of 4 DACs. 8 ADCs are needed to monitor the coil currents and the induced voltages.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

6.1 Summary of control and Monitor Points

Name	CAN ID	Data Size	Description
SET_COIL_CONTROL_REGISTER	04 04 01 42	1	Set control coil register
SET_COIL_DACS	04 04 01 40	8	Set control coil DACs
SET_COIL_MAX6633_REGISTER	04 04 01 88	1	Set MAX633 register
GET_COIL_CONTROL_REGISTER	04 04 01 43	8	Get control coil register
GET_COIL_DAC_ADC	04 04 01 41	8	Get DAC or DAC values
GET_COIL_MAX6633_TEMPERATURE	04 04 01 89	3	Get MAX633 temperature

Convenience control and monitor points:

SET_COIL_REF_CHANNELS	04 04 02 80	8	Set the 4 reference channels
GET_COIL REF_CHANNELS	04 04 02 81	8	Get the 4 reference channels

GET_COIL_ACTUAL _CHANNELS_01	04 04 02 82	8	Get the actual values of the channels 0 and 1
GET_COIL_ACTUAL_CHANNELS_23	04 04 02 83	8	Get the actual values of the channels 2 and 3

6.2 Control Points in Detail

Name	SET_COIL_CONTROL_REGISTER
CAN ID	04 04 01 42
Description	Set the operation mode and define the channels to read or write.
Data	1 byte Bits [7-4] = 0x1: Soft reset of the interface Bits [7-4] = 0x2: Power the interface down Bits [7-4] = 0x3: Standby – disable the ADC conversions Bits [7-4] = 0x4: Set the pointer needed for writing and reading DAC and DAC channels. The pointer value is set with bits [3-0]. Only pointer values of 0, 4 and 8 are meaningful for the receivers PdBNG.

Name	SET_COIL_DACS
CAN ID	04 04 01 40
Description	Set the 4 DAC channels.
Data	8 bytes Bytes [0,1]: Channel 0 reference value Bytes [2,3]: Channel 1 reference value Bytes [4,5]: Channel 2 reference value Bytes [6,7]: Channel 3 reference value Definition of a channel reference value: 2 bytes = bits [15-0] Bits [15-2]: DAC reference value. Signed 2's complement number between -2^{13} and $2^{13}-1$. 2^{13}=100mA Bit [1]: Not used Bit [0]: 1 = output enabled, 0 = disabled After reset (soft reset or after power on, the channel reference values are equal to 0x00 – For each channel bits [15-0]=0x00 Before setting the coil DACs, the pointer of the coil control register must be set with a SET_ COIL_CONTROL_REGISTER can message and its data byte set to 0x48.

Name	SET_COIL_MAX6633_REGISTER
CAN ID	04 04 01 88
Description	Set the MAXIM 6633 configuration register
Data	1 byte = 0x00: enabled. Default value at power on = 0x01: disabled

Convenience control point:

Name	SET_COIL_REF_CHANNELS
CAN ID	04 04 02 81
Description	Set the 4 reference channels.
Data	8 bytes Bytes [0,1]: Channel 0 reference value Bytes [2,3]: Channel 1 reference value Bytes [4,5]: Channel 2 reference value Bytes [6,7]: Channel 3 reference value Definition of a channel reference value: 2 bytes = bits [15-0] Bits [15-2]: DAC reference value. Signed 2's complement number between -2^{13} and $2^{13}-1$. 2^{13}=100mA Bit [1]: Not used Bit [0]: 1 = output enabled, 0 = disabled After reset (soft reset or after power on, the channel reference values are equal to 0x00 – For each channel bits [15-0]=0x00 No prerequisite.

6.3 Monitor Points in Detail

Name	GET_COIL_DAC_ADC
CAN ID	04 04 01 43
Description	Get 4 ADC values or read 4 DAC reference values
Data	8 bytes Depending on the pointer of the coil control register set with the last CAN message SET_ COIL_CONTROL_REGISTER, the 8 bytes may represent different channels. For SET_ COIL_CONTROL_REGISTER data = 0x40: Bytes [0,1]: Bits [15-2]: ADC value of channel 0, current, signed 2's complement value between -2^{13} and $2^{13}-1$. 2^{13}=100mA. Bit [1]: 1 = Channel 0 thermal limit. Bit [0]: 1 = Channel 0 current limit. Bytes [2,3]: Bits [15-2]: ADC value of channel 0, voltage, signed 2's complement value between -2^{13} and $2^{13}-1$. 2^{13}=2.5V. Bit [1]: 1 = Channel 0 thermal limit. Bit [0]: 1 = Channel 0 current limit. Bytes [4,5]: Bits [15-2]: ADC value of channel 1, current, signed 2's complement value between -2^{13} and $2^{13}-1$. 2^{13}=100mA. Bit [1]: 1 = Channel 1 thermal limit.

	Bit [0]: 1 = Channel 1 current limit. Bytes [6,7]: Bits [15-2]: ADC value of channel 1, voltage, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 1 thermal limit. Bit [0]: 1 = Channel 1 current limit. 0x44: Bytes [0,1]: Bits [15-2]: ADC value of channel 2, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 2 thermal limit. Bit [0]: 1 = Channel 2 current limit. Bytes [2,3]: Bits [15-2]: ADC value of channel 2, voltage, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 2 thermal limit. Bit [0]: 1 = Channel 2 current limit. Bytes [4,5]: Bits [15-2]: ADC value of channel 3, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 3 thermal limit. Bit [0]: 1 = Channel 3 current limit. Bytes [6,7]: Bits [15-2]: ADC value of channel 3, voltage, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 3 thermal limit. Bit [0]: 1 = Channel 3 current limit. 0x48: Bytes [0,1]: Bits [15-2]: DAC reference value of channel 0, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [2,3]: Bits [15-2]: DAC reference value of channel 1, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [4,5]: Bits [15-2]: DAC reference value of channel 2, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [6,7]: Bits [15-2]: DAC reference value of channel 2, current, signed 2’s complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled.

Name	GET_COIL_MAX6633_TEMPERATURE
CAN ID	04 04 01 89
Description	Get the MAXIM 6633 temperature
Data	3 bytes Bytes [0,1] = bits [15-0] Bits [15-3] = temperature. Bit [3], lsb = .0625deg Celcius. Byte [2]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error.

Convenience monitor points:

Name	GET_COIL_REF_CHANNELS
CAN ID	04 04 02 81
Description	Get the 4 reference channels
Data	8 bytes Bytes [0,1]: Bits [15-2]: DAC reference value of channel 0, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [2,3]: Bits [15-2]: DAC reference value of channel 1, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [4,5]: Bits [15-2]: DAC reference value of channel 2, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. Bytes [6,7]: Bits [15-2]: DAC reference value of channel 2, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: Not used. Bit [0]: 1 = output enabled, 0 = output disabled. No prerequisite.

Name	GET_COIL_ACTUAL_CHANNELS_01
CAN ID	04 04 02 82
Description	Get the actual values of the channels 0 and 1
Data	8 bytes Bytes [0,1]: Bits [15-2]: ADC value of channel 0, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 0 thermal limit. Bit [0]: 1 = Channel 0 current limit. Bytes [2,3]:

	Bits [15-2]: ADC value of channel 0, voltage, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 0 thermal limit. Bit [0]: 1 = Channel 0 current limit. Bytes [4,5]: Bits [15-2]: ADC value of channel 1, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 1 thermal limit. Bit [0]: 1 = Channel 1 current limit. Bytes [6,7]: Bits [15-2]: ADC value of channel 1, voltage, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 1 thermal limit. Bit [0]: 1 = Channel 1 current limit. No prerequisite.

Name	GET_COIL_ACTUAL_CHANNELS_23
CAN ID	04 04 02 83
Description	Get the actual values of the channels 2 and 3
Data	Bytes [0,1]: Bits [15-2]: ADC value of channel 2, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 2 thermal limit. Bit [0]: 1 = Channel 2 current limit. Bytes [2,3]: Bits [15-2]: ADC value of channel 2, voltage, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 2 thermal limit. Bit [0]: 1 = Channel 2 current limit. Bytes [4,5]: Bits [15-2]: ADC value of channel 3, current, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=100mA. Bit [1]: 1 = Channel 3 thermal limit. Bit [0]: 1 = Channel 3 current limit. Bytes [6,7]: Bits [15-2]: ADC value of channel 3, voltage, signed 2's complement value between -2^{13} and 2^{13}-1. 2^{13}=2.5V. Bit [1]: 1 = Channel 3 thermal limit. Bit [0]: 1 = Channel 3 current limit. No prerequisite.

7 Vacuum

Originally the bus I2C is in use for monitoring and controlling the Vacuum. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

7.1 Summary of control and Monitor Points

Name	CAN ID	Data Size	Description
SET_ VACUUM_CONTROL_REGISTER	04 04 01 52	1	Set control register
GET_VACUUM_DATA	04 04 01 53	4	Get vacuum data

7.2 Control Points in Detail

Name	SET_VACUUM_CONTROL_REGISTER
CAN ID	04 04 01 52
Description	Set control register
Data	1 byte 0xAB: Full power down 0xAA: Standby 0xA8: On 0x88: Start vacuum read conversion

The initialization sequences is:

- go in StandBy mode for 0.5 seconds
- power on
- begin conversions

The sensor must not be always on, otherwise it will be damaged.

7.3 Monitor Points in Detail

Name	GET_VACUUM_DATA
CAN ID	04 04 01 53
Description	Get vacuum data
Data	4 bytes Bytes [0,1] = bits [15-0] Bits [15-4]: Vacuum analog voltage (12bit ADC), unsigned value, 0x800 = 5V Byte [2]: Bit [7]: Gauge status, 1=On Bit [6]: Degas status, 1=On Bit [5]: Gauge power, 1=On Bit [4]: Gauge, 1=On

	Byte [3]: Error report. Bit [2]=CAN error, bit [1]=I2C write error, bit [0]=I2C read error.

Pression (Torr) == 10^(Volt –10)

(1 Torr == 1.333224 E+02 Pascal)

8 LO

There are 3 LO boxes for the band 1 (100GHz), band 2 (150GHz) and band 3 (230GHz)

8.1 Summary of Control and Monitor Points:

Name	CAN ID	Data size	Description
SET_LO_BAND1_COMMAND	01 00 01 10	2	LO band 1 command register
GET_LO_BAND1_COMMAND	01 00 01 20	3	LO band 1 command register
SET_LO_BAND1_HARM_MIXER_BIAS	01 04 01 10	2	LO band 1 harmonic mixer bias
SET_LO_BAND1_LOOP_GAIN	01 04 01 11	2	LO band 1 loop gain
SET_LO_BAND1_GUNN_BIAS	01 04 01 12	2	LO band 1 gunn bias
GET_LO_BAND1_HARM_MIXER_BIAS	01 04 01 20	2	LO band 1 harmonic mixer bias
GET_LO_BAND1_LOOP_GAIN	01 04 01 21	2	LO band 1 loop gain
GET_LO_BAND1_GUNN_BIAS	01 04 01 22	2	LO band 1 gunn bias
SET_LO_BAND2_COMMAND	02 00 01 10	2	LO band 2 command register
GET_LO_BAND2_COMMAND	02 00 01 20	2	LO band 2 command register
SET_LO_BAND2_HARM_MIXER_BIAS	02 04 01 10	2	LO band 2 harmonic mixer bias
SET_LO_BAND2_LOOP_GAIN	02 04 01 11	2	LO band 2 loop gain
SET_LO_BAND2_GUNN_BIAS	02 04 01 12	2	LO band 2 gunn bias
GET_LO_BAND2_HARM_MIXER_BIAS	02 04 01 20	2	LO band 2 harmonic mixer bias
GET_LO_BAND2_LOOP_GAIN	02 04 01 21	2	LO band 2 loop gain
GET_LO_BAND2_GUNN_BIAS	02 04 01 22	2	LO band 2 gunn bias
SET_LO_BAND3_COMMAND	03 00 01 10	2	LO band 3 command register
GET_LO_BAND3_COMMAND	03 00 01 20	2	LO band 3 command register
SET_LO_BAND3_HARM_MIXER_BIAS	03 04 01 10	2	LO band 3 harmonic mixer bias
SET_LO_BAND3_LOOP_GAIN	03 04 01 11	2	LO band 3 loop gain
SET_LO_BAND3_GUNN_BIAS	03 04 01 12	2	LO band 3 gunn bias
GET_LO_BAND3_HARM_MIXER_BIAS	03 04 01 20	2	LO band 3 harmonic

			mixer bias
GET_LO_BAND3_LOOP_GAIN	03 04 01 21	2	LO band 3 loop gain
GET_LO_BAND3_GUNN_BIAS	03 04 01 22	2	LO band 3 gunn bias
GET_LO_BAND1_STATUS	01 00 01 00	3	LO band 1 status register
GET_LO_BAND1_OFFSET_VOLTAGE	01 04 01 00	3	LO band 1 offset voltage
GET_LO_BAND1_PLL_IF_LEVEL	01 04 01 01	3	LO band 1 PLL OF level
GET_LO_BAND1_HARM_MIXER_CURRENT	01 04 01 02	3	LO band 1 harmonic mixer current
GET_LO_BAND2_STATUS	02 00 01 00	3	LO band 2 status register
GET_LO_BAND2_OFFSET_VOLTAGE	02 04 01 00	3	LO band 2 offset voltage
GET_LO_BAND2_PLL_IF_LEVEL	02 04 01 01	3	LO band 2 PLL OF level
GET_LO_BAND2_HARM_MIXER_CURRENT	02 04 01 02	3	LO band 2 harmonic mixer current
GET_LO_BAND3_STATUS	03 00 01 00	3	LO band 3 status register
GET_LO_BAND3_OFFSET_VOLTAGE	03 04 01 00	3	LO band 3 offset voltage
GET_LO_BAND3_PLL_IF_LEVEL	03 04 01 01	3	LO band 3 PLL OF level
GET_LO_BAND3_HARM_MIXER_CURRENT	03 04 01 02	3	LO band 3 harmonic mixer current

Receiver motors:

Name	CAN ID	Data Size	Description
SET_LO_FREQ_BAND1_a	01 10 01 xx	1 or 2	See description below
SET_LO_POWER_GUNN_BAND1_a	01 14 01 xx	1 or 2	See description below
SET_LO_HARM_MIXER_POWER_BAND1_a	01 18 01 xx	1 or 2	See description below
SET_LO_POWER1_BAND1_a	01 1C 01 xx	1 or 2	See description below
SET_LO_POWER2_BAND1_a	01 20 01 xx	1 or 2	See description below
SET_LO_FREQ_BAND2_a	02 10 01 xx	1 or 2	See description below
SET_LO_POWER_GUNN_BAND2_a	02 14 01 xx	1 or 2	See description below
SET_LO_HARM_MIXER_POWER_BAND2_a	02 18 01 xx	1 or 2	See description below
SET_LO_POWER1_BAND2_a	02 1C 01 xx	1 or 2	See description below
SET_LO_POWER2_BAND2_a	02 20 01 xx	1 or 2	See description below
SET_LO_FREQ_BAND3_a	03 10 01 xx	1 or 2	See description below
SET_LO_POWER_GUNN_BAND3_a	03 14 01 xx	1 or 2	See description below
SET_LO_HARM_MIXER_POWER_BAND3_a	03 18 01 xx	1 or 2	See description below
SET_LO_POWER1_BAND3_a	03 1C 01 xx	1 or 2	See description below
SET_LO_POWER2_BAND3_a	03 20 01 xx	1 or 2	See description below
GET_LO_FREQ_BAND1_b	01 10 01 xx	1 or 2	See description below
GET_LO_POWER_GUNN_BAND1_b	01 14 01 yy	3 or 4	See description below
GET_LO_HARM_MIXER_POWER_BAND1_b	01 18 01 yy	3 or 4	See description below
GET_LO_POWER1_BAND1_b	01 1C 01 yy	3 or 4	See description below
GET_LO_POWER2_BAND1_b	01 20 01 yy	3 or 4	See description below
GET_LO_FREQ_BAND2_b	02 10 01 yy	3 or 4	See description below
GET_LO_POWER_GUNN_BAND2_b	02 14 01 yy	3 or 4	See description below
GET_LO_HARM_MIXER_POWER_BAND2_b	02 18 01 yy	3 or 4	See description below
GET_LO_POWER1_BAND2_b	02 1C 01 yy	3 or 4	See description below
GET_LO_POWER2_BAND2_b	02 20 01 yy	3 or 4	See description below

GET_LO_FREQ_BAND3_b	03 10 01 yy	3 or 4	See description below
GET_LO_POWER_GUNN_BAND3_b	03 14 01 yy	3 or 4	See description below
GET_LO_HARM_MIXER_POWER_BAND3_b	03 18 01 yy	3 or 4	See description below
GET_LO_POWER1_BAND3_b	03 1C 01 yy	3 or 4	See description below
GET_LO_POWER2_BAND3_b	03 20 01 yy	3 or 4	See description below

8.2 Control Points in Detail:

z = 1 for band 1 (100GHz), z= 2 for band 2 (150GHz) or z = 3 for band 3 (230GHz).

Name	SET_LO_BANDx_COMMAND
CAN ID	0z 00 01 10
Description	Set LO command register
Data	2 bytes Byte [0]: unused Byte [1] Bit [7-4]: unused. Bit [3]: sweep: 1:On, 0:off. Bit [2]: loop: 1:Closed, 0:Open. Bit [1]: deltaF: 1:+, 0:-. Bit [0]: gunn: 1:On, 0:Off.

Name	SET_LO_BANDx_HARM_MIXER_BIAS
CAN ID	0z 04 01 10
Description	Set LO band x harmonic mixer bias
Data	2 bytes 14 bits DAC Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V

Name	SET_LO_BANDx_LOOP_GAIN
CAN ID	0z 04 01 11
Description	Set LO band x loop gain
Data	2 bytes 14 bits DAC Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V

Name	SET_LO_BANDx_GUNN_BIAS
CAN ID	0z 04 01 12
Description	Set LO band x gunn bias
Data	2 bytes 14 bits DAC Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V

8.3 Monitor Points in Detail:

z = 1 for band 1 (100GHz), z= 2 for band 2 (150GHz) or z = 3 for band 3 (230GHz).

Name	GET_LO_BANDx_STATUS
CAN ID	0z 00 01 00
Description	Get LO band x status register
Data	3 bytes Byte [0]: unused Byte [1]: Bit [7-4]: unused. Bit [3]: sweep: 1:On, 0:off. Bit [2]: loop: 1:Closed, 0:Open. Bit [1]: deltaF: 1:+, 0:-. Bit [0]: gunn: 1:On, 0:Off. Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_COMMAND
CAN ID	0z 00 01 20
Description	Get LO band x status register
Data	3 bytes Byte [0]: unused Byte [1]: Bit [7-4]: unused. Bit [3]: sweep: 1:On, 0:off. Bit [2]: loop: 1:Closed, 0:Open. Bit [1]: deltaF: 1:+, 0:-. Bit [0]: gunn: 1:On, 0:Off. Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_OFFSET_VOLTAGE
CAN ID	0z 04 01 00
Description	Get LO band x offset voltage
Data	3 bytes 16 bits ADC Byte [0,1] = 0: 0V Byte [0,1] = 0xFFFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_PLL_IF_LEVEL
CAN ID	0z 04 01 01
Description	Get LO band x PLL IF level
Data	3 bytes

	16 bits ADC Byte[0,1] = 0: 0V Byte[0,1] = 0xFFFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_HARM_MIXER_CURRENT
CAN ID	0z 04 01 02
Description	Get LO band x harmonic mixer current
Data	3 bytes 16 bits ADC Byte [0,1] = 0: 0V Byte [0,1] = 0xFFFF : 19.9997mA Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_HARM_MIXER_BIAS
CAN ID	0z 04 01 20
Description	Get LO band x harmonic mixer bias request
Data	3 bytes Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_LOOP_GAIN
CAN ID	0z 04 01 21
Description	Get LO band x loop gain request
Data	3 bytes Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LO_BANDx_GUNN_BIAS
CAN ID	0z 04 01 22
Description	Get LO band x gunn bias request
Data	3 bytes Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

8.4 Receiver Motors – Control and monitor points:

For details see document IRAM-COMP-008 “Receiver Motor control using CAN-Bus” written by Francis Morel.
a refers to a control function and xx is its CAN ID offset.
b refers to a monitoring function and yy is its CAN ID offset.

Control points:

a	RPOS
xx	0x01
Description	Sets the Motor Requested Position
Data	2 bytes Data bytes[0..1] = 12-bit unsigned value of Actual Position. byte[0] = Position MSByte, byte[1] = Position LSByte.

a	STOP
xx	0x03
Description	Stops the Motor.
Data	1 dummy byte

a	RESET
xx	0xFF
Description	Resets the Motor. This command has highest priority, and executes inside the CAN interrupt routine.
Data	1 dummy byte

Monitor points:

b	APOS
yy	0x00
Description	Reads the Motor Actual Position
Data	3 bytes Data bytes[0..1] = 12-bit unsigned value of Actual Position. byte[0] = Position MSByte, byte[1] = Position LSByte. Transaction report in byte[2]:: Bit[0] = CAN Warning.

b	STS
yy	0x02
Description	Reads the Motor Status
Data	4 bytes Data byte[0] = 1-bit Status Code *0x20: Board reset* *0x10: Board stopped*

	0x8: Requested Position error (bytes[1..2] = Requested Position) *0x4: Position Aborted (bytes[1..2] = Actual Position)* *0x2: Position Reached (bytes[1..2] = Requested Position)* *0x1: Running (bytes[1..2] = Actual Position)* Data byte[1..2] = (Requested OR Actual) Position. byte[1] = Position MSByte, byte[2] = Position LSByte. Transaction report in byte[3]:: Bit[0] = CAN Warning.

9 LO Reference Switching and Warm IF

9.1 Summary of Control and Monitor Points:

Name	CAN ID	Data size	Description
SET_LOREF_WARMIF_COMMAND	04 40 01 10	2	LO reference switching and warm IF command register
GET_LOREF_WARMIF_COMMAND	04 40 01 20	2	LO reference switching and warm IF command register
GET_IF_LEVEL_H_POLAR	04 44 01 00	3	IF level horizontal polarization
GET_IF_LEVEL_V_POLAR	04 44 01 01	3	IF level vertical polarization

9.2 Control Points in Detail

Name	SET_LOREF_WARMIF_COMMAND
CAN ID	04 40 01 10
Description	Set LO reference switching and warm IF command register
Data	2 bytes Byte [0,1] Bit [15]: IF attenuator 1 dB polar vertical: 1:Off, 0:On. Bit [14]: IF attenuator 2 dB polar vertical: 1:Off, 0:On. Bit [13]: IF attenuator 4 dB polar vertical: 1:Off, 0:On. Bit [12]: IF attenuator 8 dB polar vertical: 1:Off, 0:On. Bit [11]: IF attenuator 16 dB polar vertical: 1:Off, 0:On. Bit [9-10]: IF channel selection polar vertical: 0: band1, 1:band2, 2:band3, 3:band4 Bit [8]: IF attenuator 1 dB polar horizontal: 1:Off, 0:On. Bit [7]: IF attenuator 2 dB polar horizontal: 1:Off, 0:On. Bit [6]: IF attenuator 4 dB polar horizontal: 1:Off, 0:On.

	Bit [5]: IF attenuator 8 dB polar horizontal: 1:Off, 0:On. Bit [4]: IF attenuator 16 dB polar horizontal: 1:Off, 0:On. Bit [2-3]: IF channel selection polar horizontal: 0: band1, 1:band2, 2:band3, 3:band4 Bit [1]: Reference 2 switching (100MHz): 1:Crossed, 0:Direct. Bit [0]: Reference 1 switching (1.8GHz): 1:Crossed, 0:Direct.

9.3 Monitor Points in Detail

Name	GET_IF_LEVEL_H_POLAR
CAN ID	04 44 01 00
Description	Get IF level horizontal polarization
Data	3 bytes 16 bits ADC Byte [0,1] = 0: 0V Byte [0,1] = 0xFFFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_IF_LEVEL_V_POLAR
CAN ID	04 44 01 01
Description	Get IF level vertical polarization
Data	3 bytes 16 bits ADC Byte[0,1] = 0: 0V Byte[0,1] = 0xFFFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_LOREF_WARMIF_COMMAND
CAN ID	04 40 01 20
Description	Get LO reference switching and warm IF command register
Data	2 bytes Byte [0,1] Bit [15]: IF attenuator 16.0 DB polar vertical: 1:Off, 0:On. Bit [14]: IF attenuator 8.0 DB polar vertical: 1:Off, 0:On. Bit [13]: IF attenuator 4.0 DB polar vertical: 1:Off, 0:On. Bit [12]: IF attenuator 2.0 DB polar vertical: 1:Off, 0:On. Bit [11]: IF attenuator 1.0 DB polar vertical: 1:Off, 0:On. Bit [9-10]: IF channel selection polar vertical: 0: band1, 1:band2, 2:band3, 3:band4 Bit [8]: IF attenuator 16.0 DB polar horizontal: 1:Off, 0:On. Bit [7]: IF attenuator 8.0 DB polar horizontal: 1:Off, 0:On. Bit [6]: IF attenuator 4.0 DB polar horizontal: 1:Off, 0:On. Bit [5]: IF attenuator 2.0 DB polar horizontal: 1:Off, 0:On. Bit [4]: IF attenuator 1.0 DB polar horizontal: 1:Off, 0:On. Bit [2-3]: IF channel selection polar horizontal:

	0: band1, 1:band2, 2:band3, 3:band4 Bit [1]: Reference 2 switching (100MHz): 1:Crossed, 0:Direct. Bit [0]: Reference 1 switching (1.8GHz): 1:Crossed, 0:Direct.

10 Multiplier Unit

Doubler for band 2.

10.1 Summary of Control and Monitor Points

Name	CAN ID	Data size	Description
SET_BIAS_DOUBLER_VALUE	04 60 01 10	2	Doubler bias request
GET_BIAS_DOUBLER_VALUE	04 60 01 20	2	Doubler bias request
GET_BIAS_DOUBLER_VOLTAGE	04 60 01 00	3	Doubler bias voltage
GET_BIAS_DOUBLER_CURRENT	04 60 01 01	3	Doubler bias current

Receiver motors:

Name	CAN ID	Data Size	Description
SET_MULTIPLIER_BACKSHORT_INPUT_a	04 70 01 xx	1 or 2	See description below
GET_MULTIPLIER_BACKSHORT_OUTPUT_b	04 74 01 yy	3 or 4	See description below

See paragraph 7.4 for a definition of the control and monitor terms a and b and for the CAN ID offsets xx and yy.

10.2 Control Points in Detail

Name	SET_BIAS_DOUBLER_VALUE
CAN ID	04 60 01 10
Description	Set doubler bias voltage request
Data	2 bytes 14 bits DAC Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V

10.3 Monitor Points in Detail

Name	GET_BIAS_DOUBLER_VOLTAGE
CAN ID	04 60 01 00
Description	Get doubler bias voltage

Data	3 bytes 16 bits ADC Byte [0,1] = 0: 0V Byte [0,1] = 0xFFFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_BIAS_DOUBLER_CURRENT
CAN ID	04 60 01 01
Description	Get doubler bias current
Data	3 bytes 16 bits ADC Byte [0,1] = 0: 0V Byte [0,1] = 0xFFFF : 9.9998mA Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_BIAS_DOUBLER_VALUE
CAN ID	04 60 01 01
Description	Get doubler bias voltage request
Data	3 bytes 14 bits DAC Byte [0,1] = 0: 0V Byte [0,1] = 0x3FFF : 9.9998V Byte [2]: Transaction report Bit [2]: CAN error

11 Power Supply operations

Originally the bus I2C is in use for monitoring and controlling some power supplies. Yves Bortolotti has developed this interface. Get from him the applicable documentation.
The I2C bus is connected to a CAN controller used as a bridge. The CAN control and monitor points derivate directly from the I2C functions.

11.1 Summary of Control and Monitor Points

Name	CAN ID	Data size	Description
SET_POWER_SUPPLY_COMMAND	04 04 01 48	1	Switch power supplies on/off
GET_POWER_SUPPLY_STATUS	04 04 01 49	1	Read power supplies commands and status

11.2 Control Points in Detail

Name	SET_POWER_SUPPLY_COMMAND
CAN ID	04 04 01 48
Description	Switch on/off the power supplies which are under I2C control
Data	1 byte : Bits[7-4] = Must be equal to 0xF Bit[3] :1 = On, 0= Off. Command coil current and cryostat temperature module power supply Bit[2] :1 = On, 0= Off. Command bias HEMT module power supply Bit[1] :1 = On, 0= Off. Command bias junctions (5-8) module power supply Bit[0] :1 = On, 0= Off. Command bias junctions (1-4) module power supply

11.3 Monitor Points in Detail

Name	GET_POWER_SUPPLY_STATUS
CAN ID	04 04 01 49
Description	Get commands and status of the power supplies which are under I2C control
Data	1 byte : Bit[7] :0 = On, 1= Off. Coil current and cryostat temperature module power supply status Bit[6] :0 = On, 1= Off. Bias HEMT module power supply status Bit[5] :0 = On, 1= Off. Bias junctions (5-8) module power supply status Bit[4] :0 = On, 1= Off. Bias junctions (1-4) module power supply status Bit[3] :1 = On, 0= Off. Coil current and cryostat temperature power supply command Bit[2] :1 = On, 0= Off. Bias HEMT module power supply command Bit[1] :1 = On, 0= Off. Bias junctions (5-8) module power supply command Bit[0] :1 = On, 0= Off. Bias junctions (1-4) module power supply command

At power on, the power supplies are requested to be on and, after, as a consequence, before any SET_POWER_SUPPLY_COMMAND CAN message, the message GET_POWER_SUPPLY_STATUS returns a byte with bit[7-4] equal to the status of the 4 power supplies and bit[3-0]=0xF.

12 LO1Ref

12.1 Summary of the Control and Monitor Points

Name	CAN ID	Data size	Description
SET_LO2	04 80 01 10	2	Sets the LO1Ref commands
GET_LO2	04 80 01 00	3	Gets the LO1Ref status

12.2 Control Points in Detail

Name	SET_LO1REF
CAN ID	04 80 01 10
Description	Set the tune requests for the 2 LO1Ref.
Data	2 bytes: Bytes [0]: unused. Bytes [1]: Bit[7-2]: unused. Bit[1]: tune LO1Ref A Bit[0]: tune LO1Ref B

12.3 Monitor Points in Detail

Name	GET_LO1REF
CAN ID	04 80 01 00
Description	Gets the status of the 2 LO2.
Data	3 bytes Byte [0]: unused. Byte [1]: Bit[7] = LO1Ref B high limit. Bit[6] = LO1Ref B low limit. Bit[5] = LO1Ref B brake. Bit[4] = LO1Ref B tuneOK. 0 => error, 1=> OK. Bit[3] = LO1Ref A high limit. Bit[2] = LO1Ref A low limit. Bit[1] = LO1Ref A brake. Bit[0] = LO1Ref A tuneOK. 0 => error, 1=> OK. Byte [2]: Transaction report Bit [2]: CAN error

13 CAN2VME

See document IRAM-COMP-005 “Plateau de Bure: CAN control of the 22 GHz Receiver and of the Antenna Subreflector” for a full description. Francis Morel has written it

The controller CAN2VME is in charge of the VME boards “22G” and “Subref”, and will forward them the commands listed below in chapters 7 and 8.
3 messages are specific to the CAN2VME controller:

13.1 Summary of the Control Points

Name	CAN ID	Data size	Description

SET_CAN2VME_SN	00 08 03 FD	8	Sets the CAN2VME 64-bit Serial Number
SET_CAN2VME_ID	00 08 03 FE	8	Sets the CAN2VME 32-bit NODE_ID
SET_CAN2VME_RESET	00 08 03 FF	1	Resets the CAN2VME controller and the VME Bus

13.2 Control Points in Detail

Name	SET_CAN2VME_SN
CAN ID	00 08 03 FD
Description	Overwrites the Controller Serial Number. Possible only with a 16-bit security key. The Serial Number is stored in a EEPROM and cannot be overwritten when loading a new firmware version.
Data	8 bytes: Bytes [0,1]: 16-bit security key must match 16 MSbits of Serial Number Bytes [2..7]: 48 LSbits of new Serial Number

Name	SET_CAN2VME_ID
CAN_ID	00 08 03 FE
Description	Sets the new NODE_ID of the Controller. Possible only with a 32-bit key. The NODE_ID is stored in a EEPROM and cannot be overwritten when loading a new firmware version.
Data	8 bytes: Bytes [0..3]: 32-bit security key Bytes [4..7]: New NODE_ID

Name	SET_CAN2VME_RESET
CAN ID	00 08 03 FF
Description	Resets the CAN Controller and VME bus. This command executes inside the CAN interrupt routine, and has highest priority. ***Exception***: This control message expects NO acknowledge message.
Data	1 byte (dummy byte to fulfill control message format requirements)

14 CAN22G

See document IRAM-COMP-005 “Plateau de Bure: CAN control of the 22 GHz Receiver and of the Antenna Subreflector” for a full description. Francis Morel has written it

14.1 Summary of Control and Monitor Points

Name	CAN ID	Data Size	Description
SET_R22_CMR	00 08 03 20	1	Command Register write
GET_R22_CNTR0	00 08 03 00	5	Counter 0 readout

GET_R22_CNTR1	00 08 03 04	5	Counter 1 readout
GET_R22_CNTR2	00 08 03 08	5	Counter 2 readout
GET_R22_PELTIER_T	00 08 03 0C	5	Peltier temp readout
GET_R22_LOAD_T	00 08 03 10	5	Load temp readout
GET_R22_2MHZ	00 08 03 14	5	2 MHz Ref readout
GET_R22_CNTR3	00 08 03 18	5	Counter 3 readout
GET_R22_STATUS	00 08 03 1E	3	Status register readout

14.2 Control Points in Detail

Name	SET_R22_CMR
CAN ID	00 08 03 20
Description	Writes the 22G Receiver Command register
Data	1 byte: Byte [0]: Bit [7..4]: unused Bit [3]: CMD_IT_ENA, enables the VME interrupt Bit [2]: CMD_NOISE_ON, turns the noise diode ON Bit [1]: CME_LOAD_ON, moves the load in front of the receiver. Bit [0]: CMD_PWR, unused but functional

14.3 Monitor Points in Detail

Name	GET_R22_CNTR0
CAN ID	00 08 03 00
Description	Reads contents of VME board “22G” Counter 0
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of counter 0. Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_CNTR1
CAN ID	00 08 03 04
Description	Reads contents of VME board “22G” Counter 1
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of counter 1 Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_CNTR2
CAN ID	00 08 03 08
Description	Reads contents of VME board “22G” Counter 2
Data	5 bytes

	Bytes [0..3] = 32-bit unsigned value of counter 2 Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_PELTIER_T
CAN ID	00 08 03 0C
Description	Reads Peltier cooler temperature of the 22G Receiver
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of Peltier cooler temperature Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_LOAD_T
CAN ID	00 08 03 10
Description	Reads Load temperature of the 22G Receiver
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of Load temperature Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_2MHZ
CAN ID	00 08 03 14
Description	Reads the 2 MHz Reference of the 22G Receiver
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of the 2 MHz frequency Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_CNTR3
CAN ID	00 08 03 18
Description	Reads contents of VME board "22G" Counter 3
Data	5 bytes Bytes [0..3] = 32-bit unsigned value of counter 3 Byte [4]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_R22_STATUS

CAN ID	00 08 03 1E
Description	Reads the 22G Receiver Status
Data	3 bytes: Bytes [0,1] = 16-bit unsigned value of Status. Byte [0]: Bit [7]: ERR, set if any error occurs. Bit [6..3]: unused Bit [2]: CAN error Bit [1: VME bus time-out Bit [0]: VME bus stuck Byte [1]: Bit [7,6]: unused Bit [5]: 22G receiver ALARM Bit [4]: UNL, the 22G VME board is unlocked, and no longer synchronized with the "TU01" pulse. Bit [3]: IT_ENA, the VME interrupt is enabled Bit [2]: NOISE_ON, the noise diode has been requested to turn ON (This bit is NOT read from the receiver) Bit [1]: LOAD_ON, the reference load is in front of the receiver Bit [0]: unused Byte [2]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

14.4 Time Event Message

Name	INT_R22_EVENT
CAN ID	00 08 03 FC
Description	Sent by the CAN2VME Controller, it reports that the VME board generated an interrupt. This interrupt occurs normally once per second, triggered by the "TU01" pulse received from the GPS time-base distribution.
Data	1 byte data Data = 0, OK Data = 1, Error: The VME 22G board has lost synchro with the "TU01". Data = 2, Error: The VME 22G board did not acknowledge the interrupt.

15 CANSubref

See document IRAM-COMP-005 "Plateau de Bure: CAN control of the 22 GHz receiver and of the Antenna Subreflector" for a full description. Francis Morel has written it.

15.1 Summary of Control and Monitor points

Name	CAN ID	Data Size	Description
SET_SUBREF_COMMAND	00 08 02 20	2	Command register
SET_SUBREF_MOTOR1_RPOS	00 08 02 24	2	Motor 1 reference position
SET_SUBREF_MOTOR2_RPOS	00 08 02 28	2	Motor 2 reference position
SET_SUBREF_MOTOR3_RPOS	00 08 02 2C	2	Motor 3 reference position
SET_SUBREF_MOTOR4_RPOS	00 08 02 30	2	Motor 4 reference position
SET_SUBREF_MOTOR5_RPOS	00 08 02 34	2	Motor 5 reference position
SET_SUBREF_MODE	00 08 02 50	1	Set remote/local mode
SET_SUBREF_REMOTE_INIT	00 08 02 52	1	Force remote init
GET_SUBREF_STATUS	00 08 02 00	3	Status register
GET_SUBREF_COMMAND	00 08 02 21	3	Command register
GET_SUBREF_MOTOR1_APOS	0 008 02 04	3	Motor 1 actual position readout
GET_SUBREF_MOTOR2_APOS	00 08 02 08	3	Motor 2 actual position readout
GET_SUBREF_MOTOR3_APOS	00 08 02 0C	3	Motor 3 actual position readout
GET_SUBREF_MOTOR4_APOS	00 08 02 10	3	Motor 4 actual position readout
GET_SUBREF_MOTOR5_APOS	00 08 02 14	3	Motor 5 actual position readout
GET_SUBREF_MOTOR1_RPOS	00 08 02 25	3	Get motor #1 reference position
GET_SUBREF_MOTOR2_RPOS	00 08 02 29	3	Get motor #2 reference position
GET_SUBREF_MOTOR3_RPOS	00 08 02 2D	3	Get motor #3 reference position
GET_SUBREF_MOTOR4_RPOS	00 08 02 31	3	Get motor #4 reference position
GET_SUBREF_MOTOR5_RPOS	00 08 02 35	3	Get motor #5 reference position
GET_SUBREF_MODE	00 08 02 51	2	Get mode (remote or local)

15.2 Control Points in Detail

Name	SET_SUBREF_COMMAND
CAN ID	00 08 02 20
Description	Set command register
Data	2 bytes Byte [0]: Bit [7]: Test Bit [6]: NVR5, motor 5 negative velocity request. Bit [5]: PVR5, motor 5 positive velocity request. NVR5=PVR5=1: Stop motor 5. NVR5=PVR5=0: Request to move to motor 5 reference position. Bit [4]: ENA5, enable motor 5 init and then enable position request. Bit [3]: NVR4, motor 4 negative velocity request.

	Bit [2]: PVR4, motor 4 positive velocity request. NVR4=PVR4=1: Stop motor 4. NVR4=PVR4=0: Request to move to motor 4 reference position. Bit [1]: ENA4, enable motor 4 init and then enable position request. Bit [0]: NVR3, motor 3 negative velocity request. Byte [1] Bit [7]: PVR3, motor 3 positive velocity request. NVR3=PVR3=1: Stop motor 3. NVR3=PVR3=0: Request to move to motor 3 reference position. Bit [6]: ENA3, enable motor 3 init and then enable position request. Bit [5]: NVR2, motor 2 negative velocity request. Bit [4]: PVR2, motor 2 positive velocity request. NVR2=PVR2=1: Stop motor 2. NVR2=PVR2=0: Request to move to motor 2 reference position. Bit [3]: ENA2, enable motor 2 init and then enable position request. Bit [2]: NVR1, motor 1 negative velocity request. Bit [1]: PVR1, motor 1 positive velocity request. NVR1=PVR1=1: Stop motor 1. NVR1=PVR1=0: Request to move to motor 1 reference position. Bit [0]: ENA1, enable motor 1 init and then enable position request.

Name	SET_SUBREF_MOTOR1_RPOS
CAN ID	00 08 02 24
Description	Set motor1 reference position
Data	2 bytes Bytes [0,1]: Writing, 2's complement signed value.

The descriptions for SET_SUBREF_MOTORx_RPOS, when x=[1..5] are identical.

Name	SET_SUBREF_MOTOR3
CAN ID	00 08 02 2C
Description	Set motor3 reference position
Data	2 bytes Bytes [0,1]: Writing, 2's complement signed value.

Name	SET_SUBREF_MOTOR4
CAN ID	00 08 02 30
Description	Set motor4 reference position
Data	2 bytes Bytes [0,1]: Writing, 2's complement signed value.

Name	SET_SUBREF_MOTOR5
CAN ID	00 08 02 34
Description	Set motor5 reference position
Data	2 bytes

	Bytes [0,1]: Writing, 2's complement signed value.

Name	SET_SUBREF_MODE
CAN ID	00 08 02 50
Description	Set remote/local mode
Data	1 bytes Bytes [0]: 0 => REMOTE mode. ; 1 => LOCAL mode

Name	SET_SUBREF_REMOTE_INIT
CAN ID	00 08 02 52
Description	Force remote init: It emulates an init request from the computer in the footer antenna
Data	1 bytes Bytes [0]: any value fits.

15.3 Monitor Points in Detail

Name	GET_SUBREF_STATUS
CAN ID	00 08 02 00
Description	Get status register
Data	3 bytes Byte [0]: Bit [7]: Test Bit [6]: RUN5, 1: motor 5 is running. Bit [5]: IDONE5, 1: motor 5 init done. Bit [4]: SW5, init switch status. 1: end of motor 5 stroke, 0: most of its stroke. Bit [3]: RUN4, 1: motor 4 is running. Bit [2]: IDONE4, 1: motor 4 init done. Bit [1]: SW4, init switch status. 1: end of motor 5 stroke, 0: most of its stroke. Bit [0]: RUN3, 1: motor 3 is running. Byte [1]: Bit [7]: IDONE3, 1: motor 3 init done. Bit [6]: SW3, init switch status. 1: end of motor 5 stroke, 0: most of its stroke. Bit [5]: RUN2, 1: motor 2 is running. Bit [4]: IDONE2, 1: motor 2 init done. Bit [3]: SW2, init switch status. 1: end of motor 5 stroke, 0: most of its stroke. Bit [2]: RUN1, 1: motor 1 is running. Bit [1]: IDONE1, 1: motor 1 init done. Bit [0]: SW1, init switch status. 1: end of motor 5 stroke, 0: most of its stroke. IDONE* gets equal to 1 when ENA*=1 (see control register) and when SW* switches from 0 to 1. Byte [2]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_SUBREF_COMMAND
CAN ID	00 08 02 21
Description	Get command register
Data	3 bytes Byte [0]: Bit [7]: Test Bit [6]: NVR5, motor 5 negative velocity request. Bit [5]: PVR5, motor 5 positive velocity request. NVR5=PVR5=1: Stop motor 5. NVR5=PVR5=0: Request to move to motor 5 reference position. Bit [4]: ENA5, enable motor 5 init and then enable position request. Bit [3]: NVR4, motor 4 negative velocity request. Bit [2]: PVR4, motor 4 positive velocity request. NVR4=PVR4=1: Stop motor 4. NVR4=PVR4=0: Request to move to motor 4 reference position. Bit [1]: ENA4, enable motor 4 init and then enable position request. Bit [0]: NVR3, motor 3 negative velocity request. Byte [1] Bit [7]: PVR3, motor 3 positive velocity request. NVR3=PVR3=1: Stop motor 3. NVR3=PVR3=0: Request to move to motor 3 reference position. Bit [6]: ENA3, enable motor 3 init and then enable position request. Bit [5]: NVR2, motor 2 negative velocity request. Bit [4]: PVR2, motor 2 positive velocity request. NVR2=PVR2=1: Stop motor 2. NVR2=PVR2=0: Request to move to motor 2 reference position. Bit [3]: ENA2, enable motor 2 init and then enable position request. Bit [2]: NVR1, motor 1 negative velocity request. Bit [1]: PVR1, motor 1 positive velocity request. NVR1=PVR1=1: Stop motor 1. NVR1=PVR1=0: Request to move to motor 1 reference position. Bit [0]: ENA1, enable motor 1 init and then enable position request. Byte [2]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

Name	GET_SUBREF_MOTOR1_APOS
CAN ID	00 08 02 04
Description	Get motor 1 actual position
Data	3 bytes Bytes [0,1]: Reading, 2's complement signed value. Equal to 0 at start time. Byte [2]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out

	Bit [0]: VME Bus stuck

The descriptions for GET_SUBREF_MOTORx_APOS, when x = [1..5] are identical.

Name	GET_SUBREF_MOTOR1_RPOS
CAN ID	00 08 02 25
Description	Get motor1 reference position
Data	2 bytes Bytes [0,1]: Writing, 2's complement signed value. Byte [1]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

The descriptions for GET_SUBREF_MOTORx_RPOS, when x = [1..5] are identical.

Name	GET_SUBREF_MODE
CAN ID	00 08 02 51
Description	Get the current mode
Data	2 bytes Byte [0]: 0 => REMOTE mode ; 1 => LOCAL mode. Byte [1]: Transaction report Bit [2]: CAN error Bit [1]: VME time-out Bit [0]: VME Bus stuck

16 Deicing and Central Hub

16.1 Summary of Control and Monitor Points

Name	CAN ID	Data Size	Description
SET_DEICE_CH_COMMAND	00 0C 01 10	2	Deicing and central hub command register
SET_DEICE_CH_SN	00 0C 01 FD	8	Deicing and Central Hub board Serial Number
SET_DEICE_CH_NODE_ID	00 0C 01 FE	8	Deicing and Central Hub board CAN Node ID
SET_DEICE_CH_RESET	00 0C 01 FF	1	Reset the Deicing and Central Hub control board
GET_DEICE_CH_STATUS	00 0C 01 00	3	Deicing and central hub status register
GET_DEICE_CH_COMMAND	00 0C 01 20	3	Deicing and central hub command register

16.2 Control Points in Detail

Name	SET_DEICE_CH_COMMAND
CAN ID	00 0C 01 10
Description	Set deicing and central hub command register
Data	2 bytes Byte [0]: unused Byte [1] Bit [7]: cmdDeiceOn. (0 => stop deceing, 1 => start deiceing) deice tej parabola, 3sectors by default (do not toggle too fast this bit, but hold it 1 second at minimum, because the underlying deice automate is slow. Bit [6]: cmdOpenCh. (0 => inactive, 1=> open the central hub) Bit [5]: cmdCloseCh (0 => inactive, 1=> close the central hub) Bit [2]: cmdDeiceReset. 1: Reset theTSX17 program. Deicereset should be set to 1 for 1 second and then reset to 0. Bit [1]: cmdPwtg Power from track + generator for a 12-sectors deicing sequence (0=>inactive, 1=> 12-sectors deicing mode) Bit [0]: cmdPw4s (0 => 3-sectors deicing mode, 1=> 6-sectors deicing mode Note: When the deicing mode is changed, the deiceing must be stopped and restarted. And the cmdDeiceOn = 0 must be hold for 1 seconds

Name	SET_DEICE_CH_SN
CAN ID	00 0C 01 FD
Description	Overwrites the Controller Serial Number. Possible only with a 16-bit security key. The Serial Number is stored in a EEPROM and cannot be overwritten when loading a new firmware version.
Data	8 bytes: Bytes [0,1]: 16-bit security key must match 16 MSbits of Serial Number Bytes [2..7]: 48 LSbits of new Serial Number

Name	SET_DEICE_CH_NODE_ID
CAN_ID	00 0C 01 FE
Description	Sets the new NODE_ID of the Controller. Possible only with a 32-bit key. The NODE_ID is stored in a EEPROM and cannot be overwritten when loading a new firmware version.
Data	8 bytes: Bytes [0..3]: 32-bit security key Bytes [4..7]: New NODE_ID

Name	SET_DEICE_CH_RESET
CAN ID	00 0C 01 FF
Description	Resets the CAN Controller. This command executes inside the CAN interrupt routine, and has highest priority. All outputs are reset to "OFF" (open). ***Exception***: This control message expects NO acknowledge message.
Data	1 byte (dummy byte to fulfill control message format requirements)

16.3 Monitor Points in Detail

Name	GET_DEICE_CH_STATUS
CAN ID	00 0C 01 00
Description	Get deicing and central hub status register
Data	3 bytes Byte [0]: unused Byte [1]: Bit [15]: stsDeiceActive, (0=> off, 1=> deicing sequence is running) Bit [7]: stsForkCabOk, Fork Cabinet Circuit Breakers. (1 => no problem, 0=> faulty, one or more sectors are off.) Bit [6]: stsRemoteCh, reflects the “remote/local” key position on the front panel. (0=> Local, 1=> Remote) Bit [5]: stsOpenCh (1=> the central hub is completely opened) Bit [4]: stsFlt230Cab 230V Power for the receiver (0=> Error, 1=> OK) Bit [3]: stsPbUpsCab, UPS for the CTI rack. (0=> no problem, 1=> error) Bit [2]: stsCloseCh (1=> the central hub is completely closed) Bit [1]: stsRemoteDeice, reflects the “remote/local” key position on the front door. (0=> Local, 1=> Remote) Bit [0]: stsDeicePowerOk, (0 => alarm, 1=> Power OK for Deicing) Byte [2]: Transaction report Bit [2]: CAN error

Name	GET_DEICE_CH_COMMAND
CAN ID	00 0C 01 20
Description	Reread deicing and central hub command register
Data	3 bytes Byte [0]: unused Byte [1] Bit [7]: unused. Bit [6]: unused. Bit [5]: deicereset. 1: Reset theTSX17 program. Deicereset should be set to 1 for 1 second and then reset to 0. Bit [4]: closech, 1: Close the central hub door. Bit [3]: opench, 1: Open the central hub door. Bit [2]: pwtg, 0/1: Power from track/track and gene for a 12 sectors deicing sequence. Bit [1]: pw4s, 0/1: 2/4 sectors deicing sequence. Bit [0]: deiceon, 1: Start deicing sequence. Byte [2]: Transaction report Bit [2]: CAN error

17 Band 4 Local Oscillator

The Band 4 OL control requires 2 boards:

-The Lodio board is in charge of digital I/O. It uses Base-ID 0x05 00 00 00.
It sets the YIG frequency, outputting a parallel 12-bit word (0x000 == 15 GHz, 0xFFF == 21 GHz).
It also sets the polarization voltages ot the AMC and the Amplifiers through a serial link (SPI) to digital potentiometers. See section "**Conversion laws…**".
It reads a parallel 8-bit "digital input", and writes a parallel 8-bit "digital outputs".
All values (YIG, VD1, VD2, VG1, VG2, VDB, MD, VDE, VGE, digital outputs, clup) can be saved into a non-volatile memory (EEProm) as default values, applied upon startup, using CAN ID 050001B0 (1 byte dummy data).

-The Loana board is in charge of analog I/O. It uses CAN-ID 0x05 04 00 00 The analog outputs (Loana DAC function) are not used in the initial design.

17.1 Summary of control and monitor points:

-N.B:
-The parameter "clup", if not equal to zero, defines the duration of the "Clear Unlock" pulse sent to the PLL. This pulse is generated each time bit[0] of digital output byte is set, using CAN ID 05000170. Bit[0] is then reset by hardware and will be reread as zero.
The pulse duration is: [15 + (1.6 * clup)] microseconds.
-If parameter "clup" = 0, bit "Clear Unlock" is set and reset under software control, as any other output bit.

i[0-15] is the DAC channel number (used for control)
j[0-31] is the ADC channel number (used for monitoring)

Function	**CAN ID**	**Data Size**	**Description**
Set YIG frequency	05 00 01 00	2	Set YIG Oscillator frequency
Get YIG frequency	05 00 01 10	3	Get YIG Oscillator frequency
Set Ampli VD1,VD2,VG1,VG2	05 00 01 20	4	Set Ampli1 Polarisation voltages
Get Ampli VD1,VD2,VG1,VG2	05 00 01 30	5	Get Ampli2 Polarisation voltages
Set AMC VDB, MD, VDE, VGE	05 00 01 40	4	Set AMC Polarisation voltages
Get AMC VDB, MD, VDE, VGE	05 00 01 50	5	Get AMC Polarisation voltages
Get digital inputs	05 00 01 60	2	Get 8 digital inputs
Set digital outputs	05 00 01 70	1	Set 8 digital outputs
Get digital outputs	05 00 01 80	2	Get 8 digital outputs
Set clup value	05 00 01 90	2	Set Clear Unlock Pulse duration
Get clup value	05 00 01 A0	3	Get Clear Unlock Pulse duration
Save default values	05 00 01 B0	1	Store current values in EEPROM
Set reset	05 00 01 FF	1	Reset the Lodio board
Get analog input[j]	05 04 01 00 + j	3	Get analog input [0-31]
Set analog output[i]	05 04 01 20 + i	2	Set Analog output [0-15]
Get analog output[i]	05 04 01 30 + i	3	Get Analog output [0-15]
Set reset	05 04 01 FF	1	Reset the Loana board

17.2 Control points in detail:

"i" ranges from 0 to 15

Function	**CAN ID**	**Data**	**Description**

		Size	
Set YIG frequency	05 00 01 00	2	Byte[0]: freq[11-8] Byte [1]:freq[7-0] 0x000 == 15 GHz 0xFFF == 21 GHz LSB = around 1.465 MHz
Set Ampli VD1,VG1,VD2,VG2	05 00 01 20	4	Byte[0]: VD1[7-0] Byte[1]: VG1[7-0] Byte[2]: VD2[7-0] Byte[3]: VG2[7-0]
Set AMC VDB, MD, VDE, VGE	05 00 01 40	4	Byte[0]: VDB[7-0] Byte[1]: MD[7-0] Byte[2]: VDE[7-0] Byte[3]: VGE[7-0]
Set digital outputs	05 00 01 70	1	Byte[0]: Output bits[7-0] Bits[7-4]: available, undefined Bit3: PLL POL Bit2: PLL BWSEL Bit1: PLL ZERO Bit0: PLL CLR ULOCK
Set “clup” duration	05 00 01 90	2	Byte[0-1]: clup[15-0] Sets the duration of the pulse generated upon reception of a command “set digital outputs” with bit[0] = 1 (PLL CLR ULOCK). Unit is 1.6 usec.
Set default values	05 00 01 B0	1	Data is dummy. Stores in EEPROM current requested values of: -YIG frequency -Ampli (VD1,VG1,VD2,VG2) -AMC (VDB,MD,VDE,VGE) -Digital outputs These values will be applied upon startup or reset.
Set reset digital control	05 00 01 FF	1	Reset the LODIO board, similar to shutdown/restart. Default values are applied.
Set analog output[i]	05 04 01 20 + i	2	Set 14-bit Analog output [0-15] requested value. MIN/MAX value: 0xE000 == -10.000 Volt 0x1FFF == +10.000 Volt Byte[0,1]: data signed value
Set reset analog control	05 04 01 FF	1	Reset the LOANA board, similar to shutdown/restart.

17.3 Monitor points in detail:

"i" ranges from 0 to 15
"j" ranges from 0 to 31

Function	CAN ID	Data Size	Description
Get YIG requested frequency	05 00 01 10	3	Byte[0]: freq[11-8] Byte [1]: freq[7-0] Byte[2]: Bit2:CAN Error
Get Ampli VD1,VG1,VD2,VG2 requested values	05 00 01 30	5	Byte[0]: VD1[7-0] Byte[1]: VG1[7-0] Byte[2]: VD2[7-0] Byte[3]: VG2[7-0] Byte[4] Bit[2]: CAN Error
Get AMC VDB, MD, VDE, VGE requested values	05 00 01 50	5	Byte[0]: VDB[7-0] Byte[1]: MD[7-0] Byte[2]: VDE[7-0] Byte[3]: VGE[7-0] Byte[4]: Bit2:CAN Error
Get digital inputs	05 00 01 60	2	Byte[0]: Input bits[7-0] Bits[7-4]: available, undefined Bit3: unused Bit2: PLL LOCK Bit1: PLL LULOCK Bit0: PLL REF/IF Byte[1]: Bit2: CAN Error
Get digital outputs	05 00 01 80	2	Byte[0]: Output bits[7-0] Byte[1]: Bit2: CAN Error
Get "clup" duration	05 00 01 A0	3	Byte[0-1]: clup[15-0] Unit=1.6 usec Byte[2]: Bit2: CAN Error
Get Analog Input[j] actual value	05 04 01 00 + j	3	Get analog input [0-31] actual value: Min/Max value: 0x8000 == -10.000V 0x7FFF == +10.000 V Byte[0-1]: data signed value. Byte[2]: transaction report: Bit[2]: CAN Error.
Get Analog Output[i] requested value	05 04 01 30 + i	3	Get Analog output [0-15] requested value: Min/Max value: 0xE000 == -10.000V

			0x1FFF == +10.000 V Byte[0-1]: data signed value. Byte[2]: transaction report: Bit[2]: CAN Error.

17.3.1 ADC Channel Number:

Channel number (j)	**Analog input name**
0	AMPLI VD2
1	AMPLI ID2
2	AMPLI –3V
3	AMPLI VG2
4	AMPLI ID1
5	AMPLI VD1
6	AMPLI VG1
7	AMPLI +5V
8	PLL COR-Voltage
9	PLL TEMP
10	PLL IF
11	PLL REF
12	AMC VG(B)
13	AMC VD(B)
14	AMC VG(E)
15	AMC VG(A)
16	AMC VD(A)
17	AMC ID(B)
18	AMC ID(A)
19	AMC ID(F)
20	AMC VD(E)
21	AMC ID(E)
22	AMC +5V
23	AMC -3V
24	AMC M(D)
25	Power +6V
26	Power (+15V / 2) (read as +7.5V)
27	Power (-15V / 2) (read as -7.5V)
28	FREE
29	FREE
30	FREE
31	FREE

17.3.2 DAC Channel Number:

Channel number (i)	Analog output name
0	FREE
1	FREE
2	FREE
3	FREE
4	UNUSED
5	UNUSED
6	UNUSED
7	UNUSED
8	UNUSED
9	UNUSED
10	UNUSED
11	UNUSED
12	UNUSED
13	UNUSED
14	UNUSED
15	UNUSED

17.3.3 Conversion laws for Ampli and AMC requested voltages:

Each binary requested value drives a digital potentiometer, whose actual output voltage depends on the value written into the device, and on the supply voltage. The relation between requested value and output voltage is not linear, because of the influence of the load connected to the potentiometer output, as with any "standard" potentiometer.
Here are the conversion laws computed by Francois Mattiocco, using a 3rd order polynomial approximation:

BIN-VGE = 45.59 – (0.32*VGE) + (2.48^{E}-5*VGE^2) + (1.37^{E}-7*VGE^3), with VGE in milliVolt.
BIN-VDE = 0.025 + (0.1024*VDE), with VDE in milliVolt.
BIN-MD = 155.33512 – (29.21939*MD) – (1.21405*MD^2) + (0.14287*MD^3), with MD in Volt.
BIN-VDB = 0.041 + (VDB*0.051), with VDB in milliVolt.
BIN-VD1 = 0.0123 + (0.102*VD1), with VD1 in milliVolt.
BIN-VD2 = 0.0123 + (0.102*VD2), with VD2 in milliVolt.
BIN-VG1 = 45.45 – (0.32397*VG1) + (2.66E-5*VG1^2) + (1.40E-7*VG1^3), with VG1 in milliVolt.
BIN-VG2 = 45.45 – (0.32397*VG2) + (2.66E-5*VG2^2) + (1.40E-7*VG2^3), with VG2 in milliVolt.