IRAM-COMP-053

Revision: 1
11/7/2008

Contact Author

Institut de RadioAstronomie Millimétrique

PVNG CAN Interface

Owner Alain Perrigouard (perrigou@iram.fr)

Keywords: CAN, PVNG receivers

Approved by:	Date:	Signature:
A.Perrigouard	Sept 30, 2008	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS

Content

1 Introduction

The CAN bus in use for monitor and control the Pico Veleta 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

Band 1:

Name	CAN ID	Data Size	Description
SET_REFERENCE_B1_PV_J1	00 08 01 10	2	Set Band 1 Polarity V Junction 1 reference
SET_REFERENCE_B1_PV_J2	00 08 01 14	2	Set Band 1 Polarity V Junction 2 reference
SET_REFERENCE_B1_PH_J1	00 08 01 18	2	Set Band 1 Polarity H Junction 1 reference
SET_REFERENCE_B1_PH_J2	00 08 01 1C	2	Set Band 1 Polarity H Junction 2 reference
SET_JUNC_REF_REG_B1	00 08 01 12	1	Set Band 1 reference register
GET_ACTUAL_CURRENT_B1_PV_J1	00 08 01 11	3	Get Band 1 Polarity V Junction 1 actual current
GET_ACTUAL_CURRENT_B1_PV_J2	00 08 01 15	3	Get Band 1 Polarity V Junction 2 actual current
GET_ACTUAL_CURRENT_B1_PH_J1	00 08 01 19	3	Get Band 1 Polarity H Junction 1 actual current
GET_ACTUAL_CURRENT_B1_PH_J2	00 08 01 1D	3	Get Band 1 Polarity H Junction 2 actual current
GET_ACTUAL_VOLTAGE_B1_PV_J1	00 08 01 13	3	Get Band 1 Polarity V Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B1_PV_J2	00 08 01 17	3	Get Band 1 Polarity V Junction 2 actual voltage
GET_ACTUAL_VOLTAGE_B1_PH_J1	00 08 01 1B	3	Get Band 1 Polarity H Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B1_PH_J2	00 08 01 1F	3	Get Band 1 Polarity H Junction 2 actual voltage
GET_REFERENCE_B1_PV_J1	00 08 01 11	3	Get Band 1 Polarity V Junction 1 reference
GET_REFERENCE_B1_PV_J2	00 08 01 15	3	Get Band 1 Polarity V Junction 2 reference
GET_REFERENCE_B1_PH_J1	00 08 01 19	3	Get Band 1 Polarity H Junction 1 reference
GET_REFERENCE_B1_PH_J2	00 08 01 1D	3	Get Band 1 Polarity H Junction 2 reference
GET_JUNC_REF_REG_B1	00 08 01 13	2	Get Band 1 reference register

Band 2:

SET_REFERENCE_B2_PV_J1	00 08 01 20	2	Set Band 2 Polarity V Junction 1 reference
SET_REFERENCE_B2_PV_J2	00 08 01 24	2	Set Band 2 Polarity V Junction 2 reference
SET_REFERENCE_B2_PH_J1	00 08 01 28	2	Set Band 2 Polarity H Junction 1 reference
SET_REFERENCE_B2_PH_J2	00 08 01 2C	2	Set Band 2 Polarity H Junction 2 reference
SET_JUNC_REF_REG_B2	00 08 01 22	1	Set Band 2 reference register
GET_ACTUAL_CURRENT_B2_PV_J1	00 08 01 21	3	Get Band 2 Polarity V Junction 1 actual current
GET_ACTUAL_CURRENT_B2_PV_J2	00 08 01 25	3	Get Band 2 Polarity V Junction 2 actual current
GET_ACTUAL_CURRENT_B2_PH_J1	00 08 01 29	3	Get Band 2 Polarity H Junction 1 actual current
GET_ACTUAL_CURRENT_B2_PH_J2	00 08 01 2D	3	Get Band 2 Polarity H Junction 2 actual current
GET_ACTUAL_VOLTAGE_B2_PV_J1	00 08 01 23	3	Get Band 2 Polarity V Junction 1 actual voltage

GET_ACTUAL_VOLTAGE_B2_PV_J2	00 08 01 27	3	Get Band 2 Polarity V Junction 2 actual voltage
GET_ACTUAL_VOLTAGE_B2_PH_J1	00 08 01 2B	3	Get Band 2 Polarity H Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B2_PH_J2	00 08 01 2F	3	Get Band 2 Polarity H Junction 2 actual voltage
GET_REFERENCE_B2_PV_J1	00 08 01 21	3	Get Band 2 Polarity V Junction 1 reference
GET_REFERENCE_B2_PV_J2	00 08 01 25	3	Get Band 2 Polarity V Junction 2 reference
GET_REFERENCE_B2_PH_J1	00 08 01 29	3	Get Band 2 Polarity H Junction 1 reference
GET_REFERENCE_B2_PH_J2	00 08 01 2D	3	Get Band 2 Polarity H Junction 2 reference
GET_JUNC_REF_REG_B2	00 08 01 23	2	Get Band 2 reference register

Band 3:

SET_REFERENCE_B3_PV_J1	00 08 01 30	2	Set Band 3 Polarity V Junction 1 reference
SET_REFERENCE_B3_PV_J2	00 08 01 34	2	Set Band 3 Polarity V Junction 2 reference
SET_REFERENCE_B3_PH_J1	00 08 01 38	2	Set Band 3 Polarity H Junction 1 reference
SET_REFERENCE_B3_PH_J2	00 08 01 3C	2	Set Band 3 Polarity H Junction 2 reference
SET_JUNC_REF_REG_B3	00 08 01 32	1	Set Band 3 reference register
GET_ACTUAL_CURRENT_B3_PV_J1	00 08 01 31	3	Get Band 3 Polarity V Junction 1 actual current
GET_ACTUAL_CURRENT_B3_PV_J2	00 08 01 35	3	Get Band 3 Polarity V Junction 2 actual current
GET_ACTUAL_CURRENT_B3_PH_J1	00 08 01 39	3	Get Band 3 Polarity H Junction 1 actual current
GET_ACTUAL_CURRENT_B3_PH_J2	00 08 01 3D	3	Get Band 3 Polarity H Junction 2 actual current
GET_ACTUAL_VOLTAGE_B3_PV_J1	00 08 01 33	3	Get Band 3 Polarity V Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B3_PV_J2	00 08 01 37	3	Get Band 3 Polarity V Junction 2 actual voltage
GET_ACTUAL_VOLTAGE_B3_PH_J1	00 08 01 3B	3	Get Band 3 Polarity H Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B3_PH_J2	00 08 01 3F	3	Get Band 3 Polarity H Junction 2 actual voltage
GET_REFERENCE_B3_PV_J1	00 08 01 31	3	Get Band 3 Polarity V Junction 1 reference
GET_REFERENCE_B3_PV_J2	00 08 01 35	3	Get Band 3 Polarity V Junction 2 reference
GET_REFERENCE_B3_PH_J1	00 08 01 39	3	Get Band 3 Polarity H Junction 1 reference
GET_REFERENCE_B3_PH_J2	00 08 01 3D	3	Get Band 3 Polarity H Junction 2 reference
GET_JUNC_REF_REG_B3	00 08 01 33	2	Get Band 3 reference register

Band 4:

SET_REFERENCE_B4_PV_J1	00 08 01 E0	2	Set Band 4 Polarity V Junction 1 reference
SET_REFERENCE_B4_PV_J2	00 08 01 E4	2	Set Band 4 Polarity V

			Junction 2 reference
SET_REFERENCE_B4_PH_J1	00 08 01 E8	2	Set Band 4 Polarity H Junction 1 reference
SET_REFERENCE_B4_PH_J2	00 08 01 EC	2	Set Band 4 Polarity H Junction 2 reference
SET_JUNC_REF_REG_B4	00 08 01 E2	1	Set Band 4 reference register
GET_ACTUAL_CURRENT_B4_PV_J1	00 08 01 E1	3	Get Band 4 Polarity V Junction 1 actual current
GET_ACTUAL_CURRENT_B4_PV_J2	00 08 01 E5	3	Get Band 4 Polarity V Junction 2 actual current
GET_ACTUAL_CURRENT_B4_PH_J1	00 08 01 E9	3	Get Band 4 Polarity H Junction 1 actual current
GET_ACTUAL_CURRENT_B4_PH_J2	00 08 01 ED	3	Get Band 4 Polarity H Junction 2 actual current
GET_ACTUAL_VOLTAGE_B4_PV_J1	00 08 01 E3	3	Get Band 4 Polarity V Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B4_PV_J2	00 08 01 E7	3	Get Band 4 Polarity V Junction 2 actual voltage
GET_ACTUAL_VOLTAGE_B4_PH_J1	00 08 01 EB	3	Get Band 4 Polarity H Junction 1 actual voltage
GET_ACTUAL_VOLTAGE_B4_PH_J2	00 08 01 EF	3	Get Band 4 Polarity H Junction 2 actual voltage
GET_REFERENCE_B4_PV_J1	00 08 01 E1	3	Get Band 4 Polarity V Junction 1 reference
GET_REFERENCE_B4_PV_J2	00 08 01 E5	3	Get Band 4 Polarity V Junction 2 reference
GET_REFERENCE_B4_PH_J1	00 08 01 E9	3	Get Band 4 Polarity H Junction 1 reference
GET_REFERENCE_B4_PH_J2	00 08 01 ED	3	Get Band 4 Polarity H Junction 2 reference
GET_JUNC_REF_REG_B4	00 08 01 E3	2	Get Band 4 reference register

Convenience control and monitor points :
Only those convenience commands are used by the control software.

SET_JUNC_REF_REG_B1	00 08 01 12	1	Set Band 1 reference register
GET_JUNC_STATUS_REG_B2	00 08 02 02	2	Get the status register of Band 2 Junctions PV_J1, PV_J2, PH_J1,PH_J2.
GET_JUNC_STATUS_REG_B3	00 08 02 03	2	Get the status register of Band 3 Junctions PV_J1, PV_J2, PH_J1,PH_J2.
GET_JUNC_STATUS_REG_B4	00 08 02 04	2	Get the status register of Band 4 Junctions PV_J1, PV_J2, PH_J1,PH_J2.
SET_B1_PV_J1_REFERENCE	00 08 02 10	2	Set Band 1 Polar V Junction 1 reference
GET_B1_PV_J1_REFERENCE	00 08 02 11	3	Get Band 1 Polar V Junction 1 reference
GET_B1_PV_J1_ACTUAL_VOLTAGE	00 08 02 12	3	Get Band 1 Polar V Junction 1 actual voltage
GET_B1_PV_J1_ACTUAL_CURRENT	00 08 02 13	3	Get Band 1 Polar V Junction 1 actual current
SET_B1_PV_J2_REFERENCE	00 08 02 14	2	Set Band 1 Polar V Junction 2 reference
GET_B1_PV_J2_REFERENCE	00 08 02 15	3	Get Band 1 Polar V Junction 2 reference
GET_B1_PV_J2_ACTUAL_VOLTAGE	00 08 02 16	3	Get Band 1 Polar V

			Junction 2 actual voltage
GET_B1_PV_J2_ACTUAL_CURRENT	00 08 02 17	3	Get Band 1 Polar V Junction 2 actual current
SET_B1_PH_J1_REFERENCE	00 08 02 18	2	Set Band 1 Polar H Junction 1 reference
GET_B1_PH_J1_REFERENCE	00 08 02 19	3	Get Band 1 Polar H Junction 1 reference
GET_B1_PH_J1_ACTUAL_VOLTAGE	00 08 02 1A	3	Get Band 1 Polar H Junction 1 actual voltage
GET_B1_PH_J1_ACTUAL_CURRENT	00 08 02 1B	3	Get Band 1 Polar H Junction 1 actual current
SET_B1_PH_J2_REFERENCE	00 08 02 1C	2	Set Band 1 Polar H Junction 2 reference
GET_B1_PH_J2_REFERENCE	00 08 02 1D	3	Get Band 1 Polar H Junction 2 reference
GET_B1_PH_J2_ACTUAL_VOLTAGE	00 08 02 1E	3	Get Band 1 Polar H Junction 2 actual voltage
GET_B1_PH_J2_ACTUAL_CURRENT	00 08 02 1F	3	Get Band 1 Polar H Junction 2 actual current
SET_JUNC_REF_REG_B2	00 08 01 22	1	Set Band 2 reference register
GET_B2_PV_J1_REFERENCE	00 08 02 21	3	Get Band 2 Polar V Junction 1 reference
GET_B2_PV_J1_ACTUAL_VOLTAGE	00 08 02 22	3	Get Band 2 Polar V Junction 1 actual voltage
GET_B2_PV_J1_ACTUAL_CURRENT	00 08 02 23	3	Get Band 2 Polar V Junction 1 actual current
SET_B2_PV_J2_REFERENCE	00 08 02 24	2	Set Band 2 Polar V Junction 2 reference
GET_B2_PV_J2_REFERENCE	00 08 02 25	3	Get Band 2 Polar V Junction 2 reference
GET_B2_PV_J2_ACTUAL_VOLTAGE	00 08 02 26	3	Get Band 2 Polar V Junction 2 actual voltage
GET_B2_PV_J2_ACTUAL_CURRENT	00 08 02 27	3	Get Band 2 Polar V Junction 2 actual current
SET_B2_PH_J1_REFERENCE	00 08 02 28	2	Set Band 2 Polar H Junction 1 reference
GET_B2_PH_J1_REFERENCE	00 08 02 29	3	Get Band 2 Polar H Junction 1 reference
GET_B2_PH_J1	00 08 02 2A	3	Get Band 2 Polar H Junction 1 actual voltage
GET_B2_PH_J1_ACTUAL_CURRENT	00 08 02 2B	3	Get Band 2 Polar H Junction 1 actual current
SET_B2_PH_J2_REFERENCE	00 08 02 2C	2	Set Band 2 Polar H Junction 2 reference
GET_B2_PH_J2_REFERENCE	00 08 02 2D	3	Get Band 2 Polar H Junction 2 reference
GET_B2_PH_J2_ACTUAL_VOLTAGE	00 08 02 2E	3	Get Band 2 Polar H Junction 2 actual voltage
GET_B2_PH_J2_ACTUAL_CURRENT	00 08 02 2F	3	Get Band 2 Polar H Junction 2 actual current
SET_JUNC_REF_REG_B3	00 08 01 32	1	Set Band 3 reference register
GET_B3_PV_J1_REFERENCE	00 08 02 31	3	Get Band 3 Polar V Junction 1 reference
GET_B3_PV_J1_ACTUAL_VOLTAGE	00 08 02 32	3	Get Band 3 Polar V Junction 1 actual voltage
GET_B3_PV_J1_ACTUAL_CURRENT	00 08 02 33	3	Get Band 3 Polar V Junction 1 actual current
SET_B3_PV_J2_REFERENCE	00 08 02 34	2	Set Band 3 Polar V

			Junction 2 reference
GET_B3_PV_J2_REFERENCE	00 08 02 35	3	Get Band 3 Polar V Junction 2 reference
GET_B3_PV_J2_ACTUAL_VOLTAGE	00 08 02 36	3	Get Band 3 Polar V Junction 2 actual voltage
GET_B3_PV_J2_ACTUAL_CURRENT	00 08 02 37	3	Get Band 3 Polar V Junction 2 actual current
SET_B3_PH_J1_REFERENCE	00 08 02 38	2	Set Band 3 Polar H Junction 1 reference
GET_B3_PH_J1_REFERENCE	00 08 02 39	3	Get Band 3 Polar H Junction 1 reference
GET_B3_PH_J1_ACTUAL_VOLTAGE	00 08 02 3A	3	Get Band 3 Polar H Junction 1 actual voltage
GET_B3_PH_J1_ACTUAL_CURRENT	00 08 02 3B	3	Get Band 3 Polar H Junction 1 actual current
SET_B3_PH_J2_REFERENCE	00 08 02 3C	2	Set Band 3 Polar H Junction 2 reference
GET_B3_PH_J2_REFERENCE	00 08 02 3D	3	Get Band 3 Polar H Junction 2 reference
GET_B3_PH_J2_ACTUAL_VOLTAGE	00 08 02 3E	3	Get Band 3 Polar H Junction 2 actual voltage
GET_B3_PH_J2_ACTUAL_CURRENT	00 08 02 3F	3	Get Band 3 Polar H Junction 2 actual current
SET_JUNC_REF_REG_B4	00 08 01 E2	1	Set Band 4 reference register
GET_B4_PV_J1_REFERENCE	00 08 02 41	3	Get Band 4 Polar V Junction 1 reference
GET_B4_PV_J1_ACTUAL_VOLTAGE	00 08 02 42	3	Get Band 4 Polar V Junction 1 actual voltage
GET_B4_PV_J1_ACTUAL_CURRENT	00 08 02 43	3	Get Band 4 Polar V Junction 1 actual current
SET_B4_PV_J2_REFERENCE	00 08 02 44	2	Set Band 4 Polar V Junction 2 reference
GET_B4_PV_J2_REFERENCE	00 08 02 45	3	Get Band 4 Polar V Junction 2 reference
GET_B4_PV_J2_ACTUAL_VOLTAGE	00 08 02 46	3	Get Band 4 Polar V Junction 2 actual voltage
GET_B4_PV_J2_ACTUAL_CURRENT	00 08 02 47	3	Get Band 4 Polar V Junction 2 actual current
SET_B4_PH_J1_REFERENCE	00 08 02 48	2	Set Band 4 Polar H Junction 1 reference
GET_B4_PH_J1_REFERENCE	00 08 02 49	3	Get Band 4 Polar H Junction 1 reference
GET_B4_PH_J1_ACTUAL_VOLTAGE	00 08 02 4A	3	Get Band 4 Polar H Junction 1 actual voltage
GET_B4_PH_J1_ACTUAL_CURRENT	00 08 02 4B	3	Get Band 4 Polar H Junction 1 actual current
SET_B4_PH_J2_REFERENCE	00 08 02 4C	2	Set Band 4 Polar H Junction 2 reference
GET_B4_PH_J2_REFERENCE	00 08 02 4D	3	Get Band 4 Polar H Junction 2 reference
GET_B4_PH_J2_ACTUAL_VOLTAGE	00 08 02 4E	3	Get Band 4 Polar H Junction 2 actual voltage
GET_B4_PH_J2_ACTUAL_CURRENT	00 08 02 4F	3	Get Band 4 Polar H Junction 2 actual current

Receiver motors:
Motorized mixer backshorts are used for Band 2 and band 3 only.

Name	CAN ID	Data Size	Description
SET_MIXER_BACKSHORT_B2_PV	02 24 01 01	2	See description below
SET_MIXER_BACKSHORT_B2_PH	02 28 01 01	2	See description below
SET_MIXER_BACKSHORT_B3_PV	03 24 01 01	2	See description below
SET_MIXER_BACKSHORT_B3_PH	03 28 01 01	2	See description below
GET_MIXER_BACKSHORT_B2_PV	02 24 01 00	3	See description below
GET_MIXER_BACKSHORT_B2_PH	02 28 01 00	3	See description below
GET_MIXER_BACKSHORT_B3_PV	03 24 01 00	3	See description below
GET_MIXER_BACKSHORT_B3_PH	03 28 01 00	3	See description below

See Chapter 8.4 for more details on motor driving.

2.2 Control Points in Detail

Name	SET_REFERENCE_Bz_PV_J1 (z = 1,2,3,0xE for bands 1,2,3,4)
CAN ID	00 08 01 z0
Description	Set Band z Polar V Junction 1 reference, voltage or current (see SET_JUNC_REF_REG)
Data	2 bytes, signed value FS=20mV (reference voltage) FS=400uA (reference current)

Name	SET_REFERENCE_Bz_PV_J2 (z = 1,2,3,0xE for bands 1,2,3,4)
CAN ID	00 08 01 z4
Description	Set Band z Polar V Junction 2 reference, voltage or current (see SET_JUNC_REF_REG)
Data	2 bytes, signed value FS=20mV (reference voltage) FS=400uA (reference current)

Name	SET_REFERENCE_Bz_PH_J1 (z = 1,2,3,0xE for bands 1,2,3,4)
CAN ID	00 08 01 z8
Description	Set Band z Polar H Junction 1 reference, voltage or current (see SET_JUNC_REF_REG)
Data	2 bytes, signed value FS=20mV (reference voltage) FS=400uA (reference current)

Name	SET_REFERENCE_Bz_PH_J2 (z = 1,2,3,0xE for bands 1,2,3,4)
CAN ID	00 08 01 zC
Description	Set Band z Polar H Junction 2 reference, voltage or current (see SET_JUNC_REF_REG)
Data	2 bytes, signed value FS=20mV (reference voltage) FS=400uA (reference current)

Name	SET_JUNC_REF_REG_Bz (z = 1,2,3,0xE for bands 1,2,3,4)
CAN ID	00 08 01 z2
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 Bz_PV_J1 reference voltage 1 = Junction Bz_PV_J1 reference current bit[2]: 0 = Junction Bz_PV_J2 reference voltage 1 = Junction Bz_PV_J2 reference current bit[3]: 0 = Junction Bz_PH_J1 reference voltage 1 = Junction Bz_PH_J1 reference current bit[4]: 0 = Junction Bz_PH_J2 reference voltage 1 = Junction Bz_PH_J2 reference current bit[5]: 0 = junctions protected 1 = junctions non protected bit[6]: 1 ADC calibration sequence bit[7]: readReferenceRegister flag

And the convenience control points:

Name	SET_Bz_PV_J1 _REFERENCE (z=(1,2,3,4) for band(1,2,3,4)
CAN ID	00 08 02 z0
Description	Set Band z, Polar V, Junction 1 reference voltage or current (see SET_JUNC_REF_REG).
Data	2 bytes, signed value FS=20mV (reference voltage) FS=400uA (reference current)

Name	SET_Bz_PV_J2 _REFERENCE (z=(1,2,3,4) for band(1,2,3,4)
CAN ID	00 08 02 z4
Description	Set Band z, Polar V, Junction 2 reference voltage or current (see SET_JUNC_REF_REG).
Data	2 bytes, signed value see SET_Bz_PV_J1 _REFERENCE

Name	SET_Bz_PH_J1_REFERENCE (z=(1,2,3,4) for band(1,2,3,4)
CAN ID	00 08 02 z8
Description	Set Band z, Polar H, Junction 1 reference voltage or current (see SET_JUNC_REF_REG).
Data	2 bytes, signed value SET_Bz_PV_J1 _REFERENCE

Name	SET_Bz_PH_J2_REFERENCE (z=(1,2,3,4) for band(1,2,3,4)
CAN ID	00 08 02 zC
Description	Set Band z, Polar H, Junction 2 reference voltage or current (see SET_JUNC_REF_REG).
Data	2 bytes, signed value SET_Bz_PV_J1 _REFERENCE

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

SET_B1_PV_J1_REFERENCE	00 08 02 10	00 08 01 10	SET_REFERENCE_B1_PV_J1
SET_B1_PV_J2_REFERENCE	00 08 02 14	00 08 01 14	SET_REFERENCE_B1_PV_J2
SET_B1_PH_J1_REFERENCE	00 08 02 18	00 08 01 18	SET_REFERENCE_B1_PH_J1
SET_B1_PH_J2_REFERENCE	00 08 02 1C	00 08 01 1C	SET_REFERENCE_B1_PH_J2
SET_B2_PV_J1_REFERENCE	00 08 02 20	00 08 01 20	SET_REFERENCE_B2_PV_J1
SET_B2_PV_J2_REFERENCE	00 08 02 24	00 08 01 24	SET_REFERENCE_B2_PV_J2
SET_B2_PH_J1_REFERENCE	00 08 02 28	00 08 01 28	SET_REFERENCE_B2_PH_J1
SET_B2_PH_J2_REFERENCE	00 08 02 2C	00 08 01 2C	SET_REFERENCE_B2_PH_J2

SET_B3_PV_J1_REFERENCE	00 08 02 30	00 08 01 30	SET_REFERENCE_B3_PV_J1
SET_B3_PV_J2_REFERENCE	00 08 02 34	00 08 01 34	SET_REFERENCE_B3_PV_J2
SET_B3_PH_J1_REFERENCE	00 08 02 38	00 08 01 38	SET_REFERENCE_B3_PH_J1
SET_B3_PH_J2_REFERENCE	00 08 02 3C	00 08 01 3C	SET_REFERENCE_B3_PH_J2
SET_B4_PV_J1_REFERENCE	00 08 02 40	00 08 01 E0	SET_REFERENCE_B4_PV_J1
SET_B4_PV_J2_REFERENCE	00 08 02 44	00 08 01 E4	SET_REFERENCE_B4_PV_J2
SET_B4_PH_J1_REFERENCE	00 08 02 48	00 08 01 E8	SET_REFERENCE_B4_PH_J1
SET_B4_PH_J2_REFERENCE	00 08 02 4C	00 08 01 EC	SET_REFERENCE_B4_PH_J2

2.3 Monitor Points in Detail

Name	GET _ACTUAL_CURRENT_Bz_PV_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z1
Description	Get junction 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) Byte[2]: Error report bit[2]=CAN error bit[1]=I2C write error bit[0]=I2C read error

Name	GET _ACTUAL_CURRENT_Bz_PV_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z5
Description	Get junction actual current if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_CURRENT_Bz_PV_J1

Name	GET _ACTUAL_CURRENT_Bz_PH_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z9
Description	Get junction actual current if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_CURRENT_Bz_PV_J1 **but** *Rpar* = 10000

Name	GET _ACTUAL_CURRENT_Bz_PH_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 zD
Description	Get junction actual current if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_CURRENT_Bz_PV_J1 **but** *Rpar* = 10000

Name	GET _ACTUAL_VOLTAGE_Bz_PV_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z3
Description	Get junction 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

Name	GET _ACTUAL_VOLTAGE_Bz_PV_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z7
Description	Get junction actual voltage if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_VOLTAGE_Bz_PV_J1

Name	GET _ACTUAL_VOLTAGE_Bz_PH_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 zB
Description	Get junction actual voltage if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_VOLTAGE_Bz_PV_J1

Name	GET _ACTUAL_VOLTAGE_Bz_PH_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 zF
Description	Get junction actual voltage if readReferenceRegister flag has been set to 0
Data	See GET _ACTUAL_VOLTAGE_Bz_PV_J1

Name	GET_REFERENCE _Bz_PV_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z1
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG), and if readReferenceRegister flag has been set to 1
Data	3 bytes Bytes[0,1], signed value FS=20mV (reference voltage) FS=400uA (reference current) Byte[2]: Error report. bit[2]=CAN error bit[1]=I2C write error bit[0]=I2C read error

Name	GET_REFERENCE _Bz_PV_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z5
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG), and if readReferenceRegister flag has been set to 1
Data	See GET_REFERENCE _Bz_PV_J1

Name	GET_REFERENCE _Bz_PH_J1 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z9
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG), and if readReferenceRegister flag has been set to 1
Data	See GET_REFERENCE _Bz_PV_J1

Name	GET_REFERENCE _Bz_PH_J2 (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 zD
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG), and if readReferenceRegister flag has been set to 1
Data	See GET_REFERENCE _Bz_PV_J1

Name	GET_JUNC_REF_REG_BANDz (z=(1,2,3,0xE) for band (1,2,3,4))
CAN ID	00 08 01 z3
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]: not used bit[1]: 0 = Junction Bz_PV_J1 reference voltage 1 = Junction Bz_PV_J1 reference current

	bit[2]: 0 = Junction Bz_PV_J2 reference voltage 1 = Junction Bz_PV_J2 reference current bit[3]: 0 = Junction Bz_PH_J1 reference voltage 1 = Junction Bz_PH_J1 reference current bit[4]: 0 = Junction Bz_PH_J2 reference voltage 1 = Junction Bz_PH_J2 reference current bit[5]: 0 = junctions protected 1 = junctions non protected bit[6]: not used bit[7]: not used 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_Bz_PV_J1_REFERENCE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z1
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG)
Data	3 bytes Bytes[0,1], signed value FS=20mV (reference voltage) FS=400uA (reference current) Byte[2]: Error report. bit[2]=CAN error bit[1]=I2C write error bit[0]=I2C read error No prerequisite.

Name	GET_Bz_PV_J2_REFERENCE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z5
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG)
Data	3 bytes see GET_Bz_PV_J1_REFERENCE

Name	GET_Bz_PH_J1_REFERENCE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z9
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG)
Data	3 bytes see GET_Bz_PV_J1_REFERENCE

Name	GET_Bz_PH_J2_REFERENCE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 zD
Description	Get junction reference, voltage or current (see SET_JUNC_REF_REG)
Data	3 bytes see GET_Bz_PV_J1_REFERENCE

Name	GET_Bz_PV_J1_ACTUAL_VOLTAGE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z2
Description	Get junction actual voltage
Data	3 bytes Bytes[0,1], signed value. FS=20mV Byte[2]: Error report. bit[2]=CAN error bit[1]=I2C write error

	bit[0]=I2C read error No prerequisite.

Name	GET_Bz_PV_J2_ACTUAL_VOLTAGE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z6
Description	Get junction actual voltage
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_VOLTAGE

Name	GET_Bz_PH_J1_ACTUAL_VOLTAGE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 zA
Description	Get junction actual voltage
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_VOLTAGE

Name	GET_Bz_PH_J2_ACTUAL_VOLTAGE (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 zE
Description	Get junction actual voltage
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_VOLTAGE

Name	GET_Bz_PV_J1_ACTUAL_CURRENT (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z3
Description	Get junction actual current
Data	3 bytes Bytes[0,1], signed value. FS=400uA Byte[2]: Error report. bit[2]=CAN error bit[1]=I2C write error bit[0]=I2C read error No prerequisite.

Name	GET_Bz_PV_J2_ACTUAL_CURRENT (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 z7
Description	Get junction actual current
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_CURRENT.

Name	GET_Bz_PH_J1_ACTUAL_CURRENT (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 zB
Description	Get junction actual current
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_CURRENT.

Name	GET_Bz_PH_J2_ACTUAL_CURRENT (z=(1,2,3,4) for band (1,2,3,4))
CAN ID	00 08 02 zF
Description	Get junction actual current
Data	3 bytes see GET_Bz_PV_J1_ACTUAL_CURRENT.

Name	GET_JUNC_STATUS_REG_B1
CAN ID	00 08 02 01
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 V1 reference voltage 1 = Junction V1 reference current bit[2]: 0 = Junction V2 reference voltage 1 = Junction V2 reference current bit[3]: 0 = Junction H1 reference voltage 1 = Junction H1 reference current bit[4]: 0 = Junction H2 reference voltage 1 = Junction H2 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_JUNC_STATUS_REG_B2
CAN ID	00 08 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 see GET_JUNC_STATUS_REG_B1

Name	GET_JUNC_STATUS_REG_B3
CAN ID	00 08 02 03
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 see GET_JUNC_STATUS_REG_B1

Name	GET_JUNC_STATUS_REG_B4
CAN ID	00 08 02 04
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 see GET_JUNC_STATUS_REG_B1

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_HEMT_CONTROL_REGISTER	00 08 01 50	1	Set control register

SET_HEMT_CHANNEL_PCF8574A	00 08 01 70	1	Select channel PCF8574A
GET_HEMT_CONVERTED_DATA	00 08 01 51	3	Get converted data
GET_HEMT_CHANNEL_PCF8574A	00 08 01 71	2	Get channel PCF8574A

Convenience monitor points:

Name	CAN ID	Data Size	Description
GET_HEMT_1V_STAGE0	00 08 02 90	7	Get HEMT 1V stage0 values
GET_HEMT_1V_STAGE1	00 08 02 91	7	Get HEMT 1V stage1 values
GET_HEMT_1V_STAGE2	00 08 02 92	7	Get HEMT 1V stage2 values
GET_HEMT_1H_STAGE0	00 08 02 93	7	Get HEMT 1H stage0 values
GET_HEMT_1H_STAGE1	00 08 02 94	7	Get HEMT 1H stage1 values
GET_HEMT_1H_STAGE2	00 08 02 95	7	Get HEMT 1H stage2 values
GET_HEMT_2V_STAGE0	00 08 02 96	7	Get HEMT 2V stage0 values
GET_HEMT_2V_STAGE1	00 08 02 97	7	Get HEMT 2V stage1 values
GET_HEMT_2V_STAGE2	00 08 02 98	7	Get HEMT 2V stage2 values
GET_HEMT_2H_STAGE0	00 08 02 99	7	Get HEMT 2H stage0 values
GET_HEMT_2H_STAGE1	00 08 02 9A	7	Get HEMT 2H stage1 values
GET_HEMT_2H_STAGE2	00 08 02 9B	7	Get HEMT 2H stage2 values
GET_HEMT_3V_STAGE0	00 08 02 9C	7	Get all HEMT 3V VDMs
GET_HEMT_3V_STAGE1	00 08 02 9D	7	Get all HEMT 3V IDMs
GET_HEMT_3V_STAGE2	00 08 02 9E	7	Get all HEMT 3V VGMs
GET_HEMT_3H_STAGE0	00 08 02 9F	7	Get all HEMT 3H VDMs
GET_HEMT_3H_STAGE1	00 08 02 A0	7	Get all HEMT 3H IDMs
GET_HEMT_3H_STAGE2	00 08 02 A1	7	Get all HEMT 3H VGMs
GET_HEMT_4V_STAGE0	00 08 02 A2	7	Get HEMT 4V stage0 values
GET_HEMT_4V_STAGE1	00 08 02 A3	7	Get HEMT 4V stage1 values
GET_HEMT_4V_STAGE2	00 08 02 A4	7	Get HEMT 4V stage2 values
GET_HEMT_4H_STAGE0	00 08 02 A5	7	Get HEMT 4H stage0 values
GET_HEMT_4H_STAGE1	00 08 02 A6	7	Get HEMT 4H stage1 values
GET_HEMT_4H_STAGE2	00 08 02 A7	7	Get HEMT 4H stage2 values

3.2 Control Points in Detail

Name	SET_HEMT_CONTROL_REGISTER
CAN ID	00 08 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_HEMT_CHANNEL_PCF8574A
CAN ID	00 08 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_HEMT_CONVERTED_DATA
CAN ID	00 08 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_HEMT_CHANNEL_PCF8574A
CAN ID	00 08 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	00 08 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	00 08 01 82	2	Set control register
SET_CRYO_MAX6633_REGISTER	00 08 01 90	1	Set MAX6633 register
GET_ CRYO_TEMPERATURE	00 08 01 81	8	Get converted temperature data
GET_ CRYO_STATUS_REGISTER	00 08 01 83	3	Get status register
GET_CRYO_MAX6633_TEMPERATURE	00 08 01 91	3	Get MAX6633 temperature

4.2 Control Points in Detail

Name	SET_ CRYO_CONTROL_REGISTER	
CAN ID	00 08 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	00 08 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	00 08 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	00 08 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 repor
bit[2]=CAN error
bit[1]=I2C write error
bit[0]=I2C read error

Name	GET_CRYO_MAX6633_TEMPERATURE
CAN ID	00 08 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 Celsius. 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_LOAD1_DS620_REGISTER	00 08 01 92	1	Set DS620-1 register
GET_HOT_LOAD1_DS620_TEMPERATURE	00 08 01 93	3	Get hot load 1 temperature
SET_HOT_LOAD2_DS620_REGISTER	00 08 01 94	1	Set DS620-2 register
GET_HOT_LOAD2_DS620_TEMPERATURE	00 08 01 95	3	Get hot load 2 temperature

Convenience monitor point:

GET_HOT_LOAD1_TEMPERATURE	00 08 02 B0	3	Get hot load 1 temperature
GET_HOT_LOAD2_TEMPERATURE	00 08 02 B2	3	Get hot load 2 temperature

5.2 Control Points in Detail

Name	SET_HOT_LOAD1_DS620_REGISTER
CAN ID	00 08 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.

Name	SET_HOT_LOAD2_DS620_REGISTER
CAN ID	00 08 01 94
Description	Set the DS620 configuration register
Data	1 byte see SET_HOT_LOAD1_DS620_REGISTER

5.3 Monitor Points in Detail

Name	GET_HOT_LOAD1_DS620_TEMPERATURE
CAN ID	00 08 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_LOAD2_DS620_TEMPERATURE
CAN ID	00 08 01 95
Description	Get the hot load temperature (as far the configuration register is set to 0xAA).
Data	3 bytes see GET_HOT_LOAD1_DS620_TEMPERATURE

Name	GET_HOT_LOAD1_TEMPERATURE
CAN ID	00 08 02 B0
Description	Get the hot load temperature. It is a compound function which sets 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

Name	GET_HOT_LOAD2_TEMPERATURE
CAN ID	00 08 02 B2
Description	Get the hot load temperature. It is a compound function which sets automatically the configuration register for reading the hot load temperature.
Data	3 bytes see GET_HOT_LOAD1_TEMPERATURE

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	00 08 01 42	1	Set control coil register
SET_COIL_DACS	00 08 01 40	8	Set control coil DACs
SET_COIL_MAX6633_REGISTER	00 08 01 88	1	Set MAX633 register
GET_COIL_CONTROL_REGISTER	00 08 01 43	8	Get control coil register
GET_COIL_DAC_ADC	00 08 01 41	8	Get DAC or DAC values
GET_COIL_MAX6633_TEMPERATURE	00 08 01 89	3	Get MAX633 temperature

Convenience control and monitor points:

SET_COIL_REF_CHANNELS	00 08 02 80	8	Set the 4 reference channels
GET_COIL REF_CHANNELS	00 08 02 81	8	Get the 4 reference channels
GET_COIL_ACTUAL _CHANNELS_01	00 08 02 82	8	Get the actual values of the channels 0 and 1
GET_COIL_ACTUAL_CHANNELS_23	00 08 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	00 08 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 PVNG.

Name	SET_COIL_DACS
CAN ID	00 08 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_MAX6622_REGISTER
CAN ID	00 08 01 88
Description	Set the MAXIM 6622 configuration register
Data	1 byte = 0x00: enabled. Default value at power on = 0x01: disabled

Convenience control point:

Name	SET_COIL_REF_CHANNELS
CAN ID	00 08 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	00 08 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	00 08 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	00 08 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	00 08 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	00 08 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	00 08 01 52	1	Set control register
GET_VACUUM_DATA	00 08 01 53	4	Get vacuum data

7.2 Control Points in Detail

Name	SET_ VACUUM_CONTROL_REGISTER
CAN ID	00 08 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	00 08 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 4 LO boxes: one for band 1 (100GHz)
one for band 2 (150GHz
one for band 3 (230GHz)
one for band4 (320 GHz)

8.1 Summary of Control and Monitor Points:

z=1 for band 1 (100GHz), z=2 for band 2 (150GHz), z=3 for band 3 (230GHz),z=4 for band4 (320 GHz)

Name	CAN ID	Data size	Description
SET_BANDz_LO_COMMAND	0z 00 01 10	2	LO band 1 command register
GET_BANDz_LO_COMMAND	0z 00 01 20	3	LO band 1 command register
SET_BANDz_LO_HARM_MIXER_BIAS	0z 04 01 10	2	LO band 1 harmonic mixer bias
SET_BANDz_LO_LOOP_GAIN	0z 04 01 11	2	LO band 1 loop gain
SET_BANDz_LO_GUNN_BIAS	0z 04 01 12	2	LO band 1 gunn bias
GET_BANDz_LO_HARM_MIXER_BIAS	0z 04 01 20	2	LO band 1 harmonic mixer bias
GET_BANDz_LO_LOOP_GAIN	0z 04 01 21	2	LO band 1 loop gain
GET_BANDz_LO_GUNN_BIAS	0z 04 01 22	2	LO band 1 gunn bias
GET_BANDz_LO_STATUS	0z 00 01 00	3	LO band 1 status register
GET_BANDz_LO_OFFSET_VOLTAGE	0z 04 01 00	3	LO band 1 offset voltage
GET_BANDz_LO_PLL_IF_LEVEL	0z 04 01 01	3	LO band 1 PLL OF level
GET_BANDz_LO_HARM_MIXER_CURRENT	0z 04 01 02	3	LO band 1 harmonic mixer current

Receiver motors:

z=1 for band 1 (100GHz), z=2 for band 2 (150GHz), z=3 for band 3 (230GHz),z=4 for band4 (320 GHz)

Name	CAN ID	Data Size	Description
SET_ BANDz _LO_FREQ	0z 10 01 01	2	See description below
SET_BANDz _LO_POWER_GUNN	0z 14 01 01	2	See description below
SET_BANDz _LO_HARM_MIXER_POWER	0z 18 01 01	2	See description below
SET_BANDz _LO_POWER1	0z 1C 01 01	2	See description below
SET_BANDz _LO_POWER2	0z 20 01 01	2	See description below
GET_BANDz_LO_FREQ	0z 10 01 00	3	See description below
GET_BANDz _LO_POWER_GUNN	0z 14 01 00	3	See description below
GET_BANDz _LO_HARM_MIXER_POWER	0z 18 01 00	3	See description below
GET_BANDz _LO_POWER1	0z 1C 01 00	3	See description below
GET_BANDz _LO_POWER2	0z 20 01 00	3	See description below

8.2 Control Points in Detail:

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

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

Name	SET_BANDz_LO_GUNN_BIAS
CAN ID	0z 04 01 12
Description	Set LO 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), z=3 for band 3 (230GHz),z=4 for band4 (320 GHz)

Name	GET_BANDz_LO_STATUS
CAN ID	0z 00 01 00
Description	Get LO 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_BANDz_LO_COMMAND
CAN ID	0z 00 01 20
Description	Get LO command register (reread last written command)
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_BANDz_LO_OFFSET_VOLTAGE
CAN ID	0z 04 01 00
Description	Get LO 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_BANDz_LO_PLL_IF_LEVEL
CAN ID	0z 04 01 01
Description	Get LO 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_BANDz_LO_HARM_MIXER_CURRENT
CAN ID	0z 04 01 02
Description	Get LO 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_BANDz_LO_HARM_MIXER_BIAS
CAN ID	0z 04 01 20
Description	Get LO 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_BANDz_LO_LOOP_GAIN
CAN ID	0z 04 01 21
Description	Get LO 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_BANDz_LO_GUNN_BIAS
CAN ID	0z 04 01 22
Description	Get LO 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.

Control points:

SET_BANDz_LO_FREQ (xx=10) SET_BANDz_LO_POWER_GUNN(xx=14) SET_BANDz_LO_HARM_MIXER_POWER(xx=18) SET_BANDz_LO_POWER1(xx=1C) SET_BANDz_LO_POWER2(xx=20)	SET RPOS
xx =10,14,18,1C,20	0z xx 01 01
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.

STOP_BANDz_MOTOR_xx	STOP
xx=10,14,18,1C,20	0z xx 01 03
Description	Stops the Motor.
Data	1 dummy byte

RESET_BANDz_MOTOR_xx	RESET
xx=10,14,18,1C,20	0z xx 01 FF
Description	Resets the Motor. The motor will not move. This command has highest priority, and executes inside the CAN interrupt routine.
Data	1 dummy byte

Monitor points:

GET_BANDz_LO_FREQ (xx=10) GET_BANDz_LO_POWER_GUNN(xx=14) GET_BANDz_LO_HARM_MIXER_POWER(xx=18) GET_BANDz_LO_POWER1(xx=1C) GET_BANDz_LO_POWER2(xx=20)	GET APOS
xx=10,14,18,1C,20	0z xx 01 01
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. Byte[2]:Transaction report

	bit[0] = CAN Warning.

GET_BANDz_MOTORxx_STATUS	STS
xx=10,14,18,1C,20	0z xx 01 02
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 Warm IF

9.1 Summary of Control and Monitor Points:

Band 1

Name	CAN ID	Data size	Description
SET_B1_ATTENUATOR_COMMAND	01 08 01 10	2	Set attenuator command register
GET_B1_ATTENUATOR_COMMAND	01 08 01 20	3	Get attenuator command register
GET_B1_V_USB_IFLEVEL	13 04 01 00	3	IF level vertical polarization Upper side band
GET_B1_V_LSB_IFLEVEL	13 04 01 01	3	IF level vertical polarization Lower side band
GET_B1_H_USB_IFLEVEL	13 04 01 02	3	IF level horizontal polarization USB
GET_B1_H_LSB_IFLEVEL	13 04 01 03	3	IF level horizontal polarization LSB

Band 2

Name	CAN ID	Data size	Description
SET_B2_ATTENUATOR_COMMAND	02 08 01 10	2	Set attenuator command register
GET_B2_ATTENUATOR_COMMAND	02 08 01 20	3	Get attenuator command register
GET_B2_V_IFLEVEL	14 04 01 00	3	IF level vertical polarization
GET_B2_VX_IFLEVEL	14 04 01 01	3	unused
GET_B2_H_IFLEVEL	14 04 01 02	3	IF level horizontal polarization
GET_B2_HX_IFLEVEL	14 04 01 03	3	unused

Band 3

Name	CAN ID	Data size	Description
SET_B3_ATTENUATOR_COMMAND	03 08 01 10	2	Set attenuator command register
GET_B3_ATTENUATOR_COMMAND	03 08 01 20	3	Get attenuator command register
GET_B3_V_IFLEVEL	13 04 01 04	3	IF level vertical polarization
GET_B3_VX_IFLEVEL	13 04 01 05	3	unused
GET_B3_H_IFLEVEL	13 04 01 06	3	IF level horizontal polarization
GET_B3_HX_IFLEVEL	13 04 01 07	3	unused

Band 4

Name	CAN ID	Data size	Description
SET_B4_ATTENUATOR_COMMAND	04 08 01 10	2	Set attenuator command register
GET_B4_ATTENUATOR_COMMAND	04 08 01 20	3	Get attenuator command register
GET_B4_V_USB_IFLEVEL	14 04 01 04	3	IF level vertical polarization Upper side band
GET_B4_V_LSB_IFLEVEL	14 04 01 05	3	IF level vertical polarization Lower side band
GET_B4_H_USB_IFLEVEL	14 04 01 06	3	IF level horizontal polarization USB
GET_B4_H_LSB_IFLEVEL	14 04 01 07	3	IF level horizontal polarization LSB

9.2 Control Points in Detail

Name	SET_B1_ATTENUATOR_COMMAND
CAN ID	01 08 01 10
Description	Set Band1 attenuator command register
Data	2 bytes Byte[0,1] Bit[15]: B1_H_LSB 8 dB attenuator (1=OFF, 0=ON) Bit[14]: B1_H_LSB 4 dB attenuator (1=OFF, 0=ON) Bit[13]: B1_H_LSB 2 dB attenuator (1=OFF, 0=ON) Bit[12]: B1_H_LSB 1 dB attenuator (1=OFF, 0=ON) Bit[11]: B1_H_USB 8 dB attenuator (1=OFF, 0=ON) Bit[10]: B1_H_USB 4 dB attenuator (1=OFF, 0=ON) Bit[9]: B1_H_USB 2 dB attenuator (1=OFF, 0=ON) Bit[8] : B1_H_USB 1 dB attenuator (1=OFF, 0=ON) Bit[7]: B1_V_LSB 8 dB attenuator (1=OFF, 0=ON) Bit[6]: B1_V_LSB 4 dB attenuator (1=OFF, 0=ON) Bit[5]: B1_V_LSB 2 dB attenuator (1=OFF, 0=ON) Bit[4]: B1_V_LSB 1 dB attenuator (1=OFF, 0=ON) Bit[3] : B1_V_USB 8 dB attenuator (1=OFF, 0=ON) Bit[2]: B1_V_USB 4 dB attenuator (1=OFF, 0=ON) Bit[1]: B1_V_USB 2 dB attenuator (1=OFF, 0=ON) Bit[0]: B1_V_USB 1 dB attenuator (1=OFF, 0=ON)

Name	SET_B2_ATTENUATOR_COMMAND
CAN ID	02 08 01 10
Description	Set Band2 attenuator command register
Data	2 bytes Byte[0,1] Bit[15]: B2_Not used 8 dB attenuator (1=OFF, 0=ON) Bit[14]: B2_Not used 4 dB attenuator (1=OFF, 0=ON) Bit[13]: B2_Not used 2 dB attenuator (1=OFF, 0=ON)

	Bit[12]: B2_Not used 1 dB attenuator (1=OFF, 0=ON) Bit[11]: B2_H 8 dB attenuator (1=OFF, 0=ON) Bit[10]: B2_H 4 dB attenuator (1=OFF, 0=ON) Bit[9]: B2_H 2 dB attenuator (1=OFF, 0=ON) Bit[8] : B2_H 1 dB attenuator (1=OFF, 0=ON) Bit[7]: B2_Not used 8 dB attenuator (1=OFF, 0=ON) Bit[6]: B2_Not used 4 dB attenuator (1=OFF, 0=ON) Bit[5]: B2_Not used 2 dB attenuator (1=OFF, 0=ON) Bit[4]: B2_Not used 1 dB attenuator (1=OFF, 0=ON) Bit[3] : B2_V 8 dB attenuator (1=OFF, 0=ON) Bit[2]: B2_V 4 dB attenuator (1=OFF, 0=ON) Bit[1]: B2_V 2 dB attenuator (1=OFF, 0=ON) Bit[0]: B2_V 1 dB attenuator (1=OFF, 0=ON)

Name	SET_B3_ATTENUATOR_COMMAND
CAN ID	03 08 01 10
Description	Set Band3 attenuator command register
Data	See SET_B2_ATTENUATOR_COMMAND and replace “B2” with “B3”.

Name	SET_B4_ATTENUATOR_COMMAND
CAN ID	04 08 01 10
Description	Set Band4 attenuator command register
Data	See SET_B1_ATTENUATOR_COMMAND and replace “B1” with “B4”.

9.3 Monitor Points in Detail

Name	GET_B1_V_USB_IFLEVEL
CAN ID	13 04 01 00
Description	Get Band1 Vertical Polar Upper Side Band IF level
Data	3 bytes 16 bits ADC Byte[0,1] = 0: 0V Byte[0,1] = 0xFFFF : 9.9998V Readout Full Scale = 1 V = 0x0CCC Byte[2]: Transaction report bit[2]: CAN error

Name	GET_B1_V_LSB_IFLEVEL
CAN ID	13 04 01 01
Description	Get Band1 Vertical Polar Lower Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B1_H_USB_IFLEVEL
CAN ID	13 04 01 02
Description	Get Band1 Horizontal Polar Upper Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B1_H_LSB_IFLEVEL
CAN ID	13 04 01 03
Description	Get Band1 Horizontal Polar Lower Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B2_V_IFLEVEL
CAN ID	14 04 01 00
Description	Get Band2 Vertical Polar IF level
Data	3 bytes

	16 bits ADC Byte[0,1] = 0: 0V Byte[0,1] = 0xFFFF : 9.9998V Readout Full Scale = 1 V = 0x0CCC Byte[2]: Transaction report bit[2]: CAN error

Name	GET_B2_VX_IFLEVEL
CAN ID	14 04 01 01
Description	Not used
Data	

Name	GET_B2_H_IFLEVEL
CAN ID	14 04 01 02
Description	Get Band2 Horizontal Polar IF level
Data	See GET_B2_V_IFLEVEL

Name	GET_B2_HX_IFLEVEL
CAN ID	14 04 01 03
Description	Not used
Data	

Name	GET_B3_V_IFLEVEL
CAN ID	13 04 01 04
Description	Get Band3 Vertical Polar IF level
Data	See GET_B2_V_IFLEVEL

Name	GET_B3_VX_IFLEVEL
CAN ID	13 04 01 05
Description	Not used
Data	

Name	GET_B3_H_IFLEVEL
CAN ID	13 04 01 06
Description	Get Band3 Horizontal Polar IF level
Data	See GET_B2_V_IFLEVEL

Name	GET_B3_HX_IFLEVEL
CAN ID	13 04 01 07
Description	Not used
Data	

Name	GET_B4_V_USB_IFLEVEL
CAN ID	14 04 01 04
Description	Get Band4 Vertical Polar Upper Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B4_V_LSB_IFLEVEL
CAN ID	14 04 01 05
Description	Get Band4 Vertical Polar Lower Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B4_H_USB_IFLEVEL
CAN ID	14 04 01 06
Description	Get Band4 Horizontal Polar Upper Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B4_H_LSB_IFLEVEL
CAN ID	14 04 01 07
Description	Get Band4 Horizontal Polar Lower Side Band IF level
Data	See GET_B1_V_USB_IFLEVEL

Name	GET_B1_ATTENUATOR_COMMAND
CAN ID	01 08 01 20
Description	Get B1 attenuator command register
	3 bytes Byte[0,1] Bit[15]: B1_H_LSB 8 dB attenuator (1=OFF, 0=ON) Bit[14]: B1_H_LSB 4 dB attenuator (1=OFF, 0=ON) Bit[13]: B1_H_LSB 2 dB attenuator (1=OFF, 0=ON) Bit[12]: B1_H_LSB 1 dB attenuator (1=OFF, 0=ON) Bit[11]: B1_H_USB 8 dB attenuator (1=OFF, 0=ON) Bit[10]: B1_H_USB 4 dB attenuator (1=OFF, 0=ON) Bit[9]: B1_H_USB 2 dB attenuator (1=OFF, 0=ON) Bit[8] : B1_H_USB 1 dB attenuator (1=OFF, 0=ON) Bit[7]: B1_V_LSB 8 dB attenuator (1=OFF, 0=ON) Bit[6]: B1_V_LSB 4 dB attenuator (1=OFF, 0=ON) Bit[5]: B1_V_LSB 2 dB attenuator (1=OFF, 0=ON) Bit[4]: B1_V_LSB 1 dB attenuator (1=OFF, 0=ON) Bit[3] : B1_V_USB 8 dB attenuator (1=OFF, 0=ON) Bit[2]: B1_V_USB 4 dB attenuator (1=OFF, 0=ON) Bit[1]: B1_V_USB 2 dB attenuator (1=OFF, 0=ON) Bit[0]: B1_V_USB 1 dB attenuator (1=OFF, 0=ON) Byte[2]: Transaction report Bit[2]: CAN Error

Name	GET_B2_ATTENUATOR_COMMAND
CAN ID	02 08 01 20
Description	Get B2 attenuator command register
Data	3 bytes Byte[0,1] Bit[15]: B2_Not used 8 dB attenuator (1=OFF, 0=ON) Bit[14]: B2_Not used 4 dB attenuator (1=OFF, 0=ON) Bit[13]: B2_Not used 2 dB attenuator (1=OFF, 0=ON) Bit[12]: B2_Not used 1 dB attenuator (1=OFF, 0=ON) Bit[11]: B2_H 8 dB attenuator (1=OFF, 0=ON) Bit[10]: B2_H 4 dB attenuator (1=OFF, 0=ON) Bit[9]: B2_H 2 dB attenuator (1=OFF, 0=ON) Bit[8] : B2_H 1 dB attenuator (1=OFF, 0=ON) Bit[7]: B2_Not used 8 dB attenuator (1=OFF, 0=ON) Bit[6]: B2_Not used 4 dB attenuator (1=OFF, 0=ON) Bit[5]: B2_Not used 2 dB attenuator (1=OFF, 0=ON) Bit[4]: B2_Not used 1 dB attenuator (1=OFF, 0=ON) Bit[3] : B2_V 8 dB attenuator (1=OFF, 0=ON) Bit[2]: B2_V 4 dB attenuator (1=OFF, 0=ON) Bit[1]: B2_V 2 dB attenuator (1=OFF, 0=ON) Bit[0]: B2_V 1 dB attenuator (1=OFF, 0=ON) Byte[2]: transaction report Bit[2]: CAN Error

Name	GET_B3_ATTENUATOR_COMMAND
CAN ID	03 08 01 20

Description	Get B3 attenuator command register
Data	See GET_B2_ATTENUATOR_COMMAND and replace “B3” with “B2”

Name	GET_B4_ATTENUATOR_COMMAND
CAN ID	04 08 01 20
Description	Get B4 attenuator command register
	See GET_B1_ATTENUATOR_COMMAND and replace “B1” with “B4”

10 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.

10.1 Summary of Control and Monitor Points

Name	CAN ID	Data size	Description
SET_POWER_SUPPLY1_COMMAND	00 08 01 48	1	Switch power supplies on/off
GET_POWER_SUPPLY1_STATUS	00 08 01 49	1	Read power supplies commands and status
SET_POWER_SUPPLY2_COMMAND	00 08 01 4A	1	Switch power supplies on/off
GET_POWER_SUPPLY2_STATUS	00 08 01 4B	1	Read power supplies commands and status

10.2 Control Points in Detail

Name	SET_POWER_SUPPLY1_COMMAND
CAN ID	00 08 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

Name	SET_POWER_SUPPLY2_COMMAND
CAN ID	00 08 01 4A
Description	Switch on/off the power supplies which are under I2C control
Data	See SET_POWER_SUPPLY1_COMMAND

10.3 Monitor Points in Detail

Name	GET_POWER_SUPPLY1_STATUS
CAN ID	00 08 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

Name	GET_POWER_SUPPLY2_STATUS
CAN ID	00 08 01 4B
Description	Get commands and status of the power supplies which are under I2C control
Data	See GET_POWER_SUPPLY1_STATUS

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.

11 I2C Debug

For debugging purposes**:**

- The I2C Controller status can be read.
- 2 special I2C commands are implemented. They will allow reading or writing up to 6 bytes at arbitrary I2C address.

11.1 Summary of Control and Monitor Points

Name	CAN ID	Data size	Description
GET_I2C_CONTROLLER_STATUS	00 08 01 FC	1	Get I2C Controller status
DEBUG_I2C_WRITE	00 08 02 C0	8	Write x bytes at I2C address y
DEBUG_I2C_READ	00 08 02 C1	2 or 8	Read x bytes from I2C address y

11.2 Control Points in Detail

Name	DEBUG_I2C_WRITE
CAN ID	00 08 02 C0
Description	Writes a list of "x" datas at specified I2C address "y"
Data	8 Bytes: Byte[0] = I2C address to be accessed (y) Byte[1] = Number of data bytes to be written at I2C address (x) Byte[2-7] = Data bytes to be written, unused (excess) bytes will be ignored.

11.3 Monitor Points in Detail

Name	GET_I2C_CONTROLLER_STATUS
CAN ID	00 08 01 FC
Description	Get the Status byte of the I2C Controller, as sampled after last transaction.

Data	1 Byte: Read PCA9564 documentation for more information.

Name	DEBUG_I2C_READ
CAN ID	00 08 02 C1
Description	Master requests a list of "x" datas at specified I2C address "y"
Data	2 Bytes: Byte[0] = I2C address to be accessed (y) Byte[1] = Number of data Bytes to be read (x)

Name	DEBUG_I2C_READ
CAN ID	00 08 02 C1
Description	Slave replies a list of "x" data Bytes at specified I2C address "y"
Data	8 Bytes: Byte[0] = I2C accessed address (y) Byte[1] = Number of read data Bytes (x) Byte[2-7] = Read data bytes, unused bytes will be zeroed.

CAUTION: The Message "DEBUG_I2C_READ" does not respect the protocol defined for IRAM-CAN (see PdB CAN Specification, written by Alain Perrigouard) and should thus be used for debugging ONLY.

12 Band 4 Local Oscillator

The Band 4 LO control requires 3 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. Saving is done using CAN ID 050001B0 (sending one dummy byte as 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.

-The LOFP1 board is a passive interface board. All connections, including power supplies, pass through this board and are filtered or optocoupled for EMI/RFI compatibility.

12.1 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**	**Description**

		Size	
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

12.2 Control points in detail:

"i" ranges from 0 to 15

Function	CAN ID	Data Size	Description
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 Freq=(15 + Data*1.465E-3) GHz
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 bit[3]: PLL POL bit[2]: PLL BWSEL bit[1]: PLL ZERO bit[0]: 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.

12.3 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]: bit[2]: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]: bit[2]:CAN Error
Get digital inputs	05 00 01 60	2	Byte[0]: Input bits[7-0] bits[7-4]: available, undefined bit[3]: unused bit[2]: PLL LOCK bit[1]: PLL LULOCK bit[0]: PLL REF/IF Byte[1]: bit[2]: CAN Error
Get digital outputs	05 00 01 80	2	Byte[0]: Output bits[7-0] Byte[1]: bit[2]: CAN Error
Get "clup" duration	05 00 01 A0	3	Byte[0-1]: clup[15-0] Unit=1.6 usec Byte[2]: bit[2]: CAN Error
Get Analog Input[j] actual value	05 04 01 00 + j	3	Get analog input [0-31] actual value (ADC): 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 (DAC): Min/Max value: 0xE000 == -10.000V 0x1FFF == +10.000 V Byte[0-1]: data signed value. Byte[2]: transaction report: bit[2]: CAN Error.

12.3.1 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

12.3.2 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

12.3.3 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.

13 Calibration motors

3 stepper motors are used for calibration . Motors type is Phytron ZSH57-3. For more info, see:
motor-zsh57-3.pdf
Each motor is controlled by a CAN controller (BCD130-CAN developed by MIDDEX-Germany), which includes the CAN interface, the motor controller and the power driver.
These 3 controllers are connected to the main Control PC through a dedicated CAN Bus, using the 2nd port of the Tews TPMC 816-10, reserved for that special use.
The native protocol used by these controllers is CAN-Open. For more info, see:
bcd130can-en.pdf