Manual 30DV50CAN / 30DV300CAN

This manual describes the CAN interface on the 30DV50CAN and 30DV300CAN amplifiers. It is an addendum to the user manuals of the 30DV50 and 30DV300.

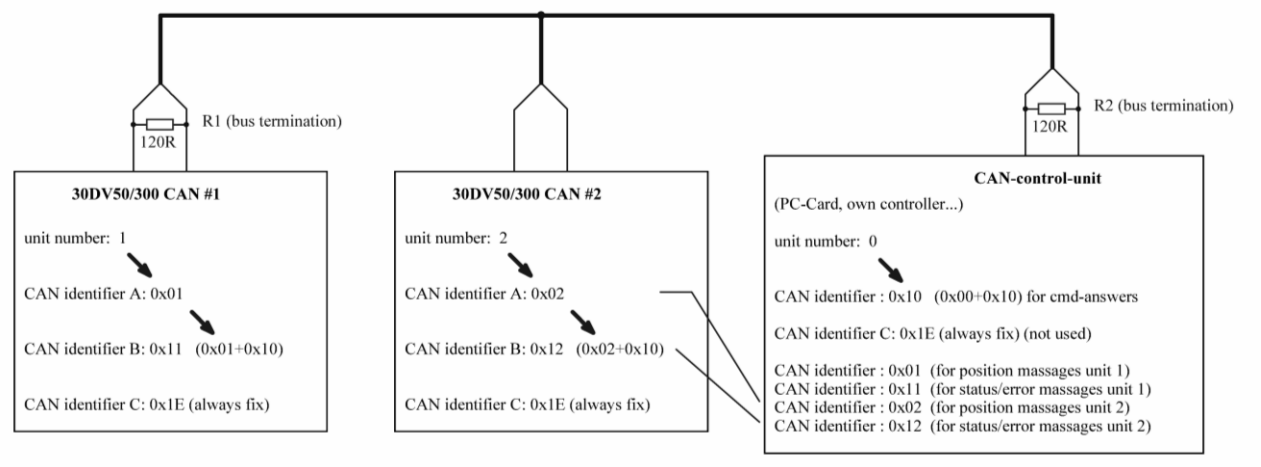
Introduction

For CAN bus details, please study external sources (e.g. https://en.wikipedia.org/wiki/CAN_bus).

piezosystem jena uses a proprietary CAN bus protocol. It was intended for controlling our EVD50/300 modules by the EDS module in the d-Drive system. The 30DV50/300CAN is based on these EVD modules. That means, that the inner-device-communication is now open for external control. Attention: The user has access to all internal commands, such as EEPROM write commands. Incorrect use can change the system data (e.g. calibration values). In such a case, the system will need a recalibration by piezosystem jena.

This manual is intended to give a brief overview over the most necessary commands for setup and control of the CAN-system. A full control and configuration (such as control- and filter parameters) should be realized by the RS232 interface.

Block sheet: CAN bus system



Sketch1: Basic CAN system and units with their receiving identifiers.

Used CAN Identifiers (11-bit Base frame format)

To separate different units, unique CAN addresses are used. From these addresses, the used CAN-Identifiers are derived. Each unit (30DV50/300CAN) uses three identifiers for receiving:

- identifier A for receiving set commands
- identifier B for receiving general commands
- identifier C (always 0x1e) for synchronous operations (e.g. reset) receiving

The (receiving) identifiers depend on the unit number (which can be read or be written by the RS232 command “canadr,unit number<CR>”).

The 30DV50/300CAN sends answers with the following identifiers:

- position messages with ID= 30DV50CAN unit number
- error / state messages with ID= 30DV50CAN unit number + 0x10
- answers to commands with ID= control unit number + 0x10

For the example shown in sketch1, the **CAN-control-unit** must receive messages with the following identifiers:

ID=0x01	position message from unit #1
ID=0x11	state / error message from unit #1
ID=0x02	position message from unit #2
ID=0x12	state / error message from unit #2
ID=0x10	answers to commands of unit number #0 (control unit)

The **CAN-control-unit** sends with the following identifiers:

ID=0x01	set position to unit #1
ID=0x11	all other commands to unit #1
ID=0x02	set position to unit #2
ID=0x12	all other commands to unit #2
ID=0x1E	synchronous operations (e.g. reset, do not use)

Note: while testing the examples, be sure the right unit-numbers are used. The CAN-Identifiers depend on this.

Examples

Example for setting position in unit number 1:

ID= 0x01	(set position receiving slot on unit #1)	
DLC=5	(5 data bytes used)	
D0=0x00	(sender ID, not used);	
D1..D4	position as single precision float, IEEE754 format, D1=LSB … D3=MSB	
	Value range:	OL: -20.0 ... 130.0 [V]
		CL: 0… xxx.x [µm / mrad]

response: --

Example for setting position in unit number 2:

ID= 0x02	(set position receiving slot on unit #2)	
DLC=5	(5 data bytes used)	
D0=0x00	(sender ID, not used);	
D1..D4	position as single precision float, IEEE754 format, D1=LSB … D3=MSB	
	Value range:	OL: -20.0 ... 130.0 [V]
		CL: 0… xxx.x [µm / mrad]

response: --

Example for setting CL mode in unit number 1:

ID= 0x11	(command receiving slot on unit #1)
DLC=3	(3 data bytes used)
D0=0x00	(sender ID, not used)
D1=0x0A	(command for OL/CL)
D2=0x01	(CL=on) (Value range: OL: 0 / CL: 1)

response: When change the ol/cl state, an automatic status message will be sent with ID=unit number + 0x10 = 0x11

Example for read out the actual actuator position on unit 2:

ID= 0x12	(command receiving slot on unit #2)
DLC=2	(3 data bytes used)
D0=0x00	(sender ID, not used)
D1=0x27	(position request command)

response:

The unit #2 sends the position with CAN-ID=unit number=2 (in position-frame-format)

ID= 0x02	(= unit number)
DLC=5	(5 data bytes used)
D0=0x02	(= unit number)
D1=LSB	(e.g 0x64)
D2=mid	(e.g 0x03)
D3=mid	(e.g 0x5C)
D4=MSB	(e.g 0x42)

(D1..D4 represents a single precision float with the actual position, e.g. 55.00331157… [V/µm/mrad])

Device Booting

After booting the device (30DV50/300CAN), each unit sends 6 messages:

- status word
- error word
- read_loop_controller (3 times, not to be used)
- status word

Afterwards the boot sequence is finished and normal operations can started.

Message frame format

11-Bit-Identifier, (base frame format) (CAN 2.0A)

Depending on the slots and message content, different frame formats are used:

position commands (Slot A):

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
	unit number (receiver)	5	sender ID (not analyzed)	position (CL) or voltage (OL) value as single				not used	not used	not used
				LSB	…	…	MSB			
example for set position to 50.0	0x01	5	0x00	0x00	0x00	0x48	0x42			

other commands:

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
version1*	unit number (receiver) + 0x10	5	sender ID	CMD number	Byte 0	Byte 1	Byte 2	Byte3	Param0**	Param1**
version2					Word 0		Word 1			
version3					Long / dword					
version4					Float					

*the used version depends on the CMD number

**in some cases D6 and D7 are used for additional parameters

Command list

(extraction, only user relevant commands)

position command (slot A = unit ID):

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
set position	unit number (receiver)	5	sender ID (not analyzed)	position (CL) or voltage (OL) value as single				not used	not used	not used
				LSB	…	…	MSB			
answer:	-	-	-	-	-	-	-	-	-	-

other commands (slot B = unit ID+0x10)

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
read status word	unit number (30DV50) + 0x10	2	sender ID	**0x08**	-	-	-	-	-	-
answer:	unit number (30DV50) + 0x10	4	unit number (30DV50)	**0x08**	device state		-	-	-	-

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
OL/CL set	unit number (30DV50) + 0x10	3	sender ID	**0x0A**	0/1	-	-	-	-	-
answer:	unit number (30DV50) + 0x10	4	unit number (30DV50)	**0x08**	device state		-	-	-	-

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
get position or voltage value	unit number (30DV50) + 0x10	2	sender ID	**0x27**	-	-	-	-	-	-
answer:	unit number (30DV50)	5	unit number (30DV50)	**0x27**	position or voltage value as single float				-	-

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
read error word	unit number (30DV50) + 0x10	2	sender ID	**0x7C**	-	-	-	-	-	-
answer:	unit number (30DV50) + 0x10	4	unit number (30DV50)	**0x7C**	error state		-	-	-	-

	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
read DSP version	unit number (30DV50) + 0x10	2	sender ID	**0x22**	-	-	-	-	-	-
answer:	sender-ID + 0x10	6	unit number (30DV50)	**0x22**	DSP version as (unsigned) long				-	-

Unit numbers

The used CAN-IDs are based on the unit numbers.
It can be read/written by the RS232 command “canadr”.
Following numbers are possible:
0...0x0D
0x0F
0x20…0x2F

Hardware:

Connector: D-Sub-9 male,

pin 2: CAN-Low (CAN−)
pin 3: GND (Ground)
pin 7: CAN-High (CAN+)

pic: view of the back panel with interfaces: Power (Vin), CAN and RS232

CAN baud rate: 1M baud

CAN bus terminator: The bus terminator is not installed on the devices, it is important to install this on CAN-Bus ends - the recommended value is 120 Ω.