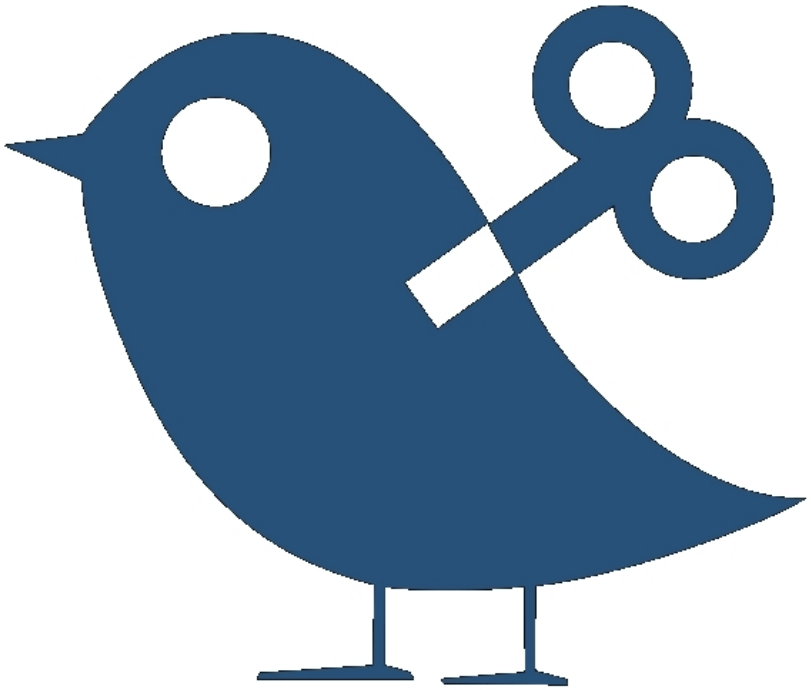
Servo actuator SD-01/02B
CANaerospace protocol description



Edition 04/27/2022

Contents

1 Introduction

CANaerospace is an extremely lightweight protocol/data format definition which was designed for the highly reliable communication of microcomputer-based systems in airborne applications via CAN (Controller Area Network). The purpose of this definition is to create a standard for applications requiring efficient data flow monitoring and easy time-frame synchronization within redundant systems. The definition is kept widely open to allow implementation of user-defined message types and protocols. CANaerospace can be used with CAN 2.0A and 2.0B (11-bit and 29-bit identifiers) and any bus data rate.

2 Communication layers

Enabling both ATM and PTP communication for CAN requires the introduction of independent network layers to isolate the different types of communication. This is realized for CANaerospace by forming CAN identifier groups as shown in Figure 1. The resulting structure creates Logical Communication Channels (LCCs) and assigns a specific communication type (ATM, PTP) to each of the LCCs. User-defined LCCs provide the necessary freedom for designers and allow the implementation of CANaerospace according to the needs of specific applications.

Channel Acronym	Communication Type	Description	CAN Identifier Range	Message Priority
EED	ATM	Emergency Event Data Channel	0 - 127	Highest
NSH	PTP	High Priority Node Service Data Channel	128 - 199	
UDH	ATM/PTP	High Priority User-Defined Data Channel	200 - 299	
NOD	ATM	Normal Operation Data Channel	300 - 1799	
UDL	ATM/PTP	Low Priority User-Defined Data Channel	1800 - 1899	
DSD	ATM/PTP	Debug Service Data Channel	1900 - 1999	
NSL	PTP	Low Priority Node Service Data Channel	2000 - 2031	Lowest

Figure 1 Logical Communication Channels for **CANaerospace**

As a side effect, the CAN identifier groups in Figure 1 affect the priority of the message transmission in case of bus arbitration. The communication channels are therefore arranged according to their relative importance:

- **Emergency Event Data Channel (EED):** This communication channel is used for messages which require immediate action (i.e. system degradation or reconfiguration) and have to be transmitted with very high priority. Emergency Event Data uses ATM communication exclusively.

- **High/Low Priority Node Service Data Channel (NSH/NSL):** These communication channels are used for client/server interactions using PTP communication. The corresponding services may be of the connection-oriented as well as the connectionless type. NSH/NSL may also be used to support test and maintenance functions.
- **Normal Operation Data Channel (NOD):** This communication channel is used for the transmission of the data which is generated during normal system operation and described in the CANaerospace identifier assignment list. These messages may be transmitted periodically or aperiodically as well as synchronously or asynchronously. All messages which cannot be assigned to other communication channels shall use this channel.
- **High/Low Priority User-Defined Data Channel (UDH/UDL):** This channel is dedicated to communication which cannot, due to their specific characteristics, be assigned other channels without violating the CANaerospace specification. As long as the defined identifier range is used, the message content and the communication type (ATM, PTP) for these channels may be specified by the system designer. To ensure interoperability it is highly recommended that the use of these channels is minimized.
- **Debug Service Data Channel (DSD):** This channel is dedicated to messages which are used temporarily for development and test purposes only and are not transmitted during normal operation. As long as the defined identifier range is used, the message content and the communication type (ATM, PTP) for these channels may be specified by the system designer.

3 Data types

For data representation, the most commonly used basic data types are defined. Additionally, combined data types (i.e. two, three and four 16-bit and 8-bit data types in one CAN message) and aggregate data types (64-bit double float) are supported. Other data types can be added to the type list as required. The type number in the range of 0-255 is used for data type specification as described in section 4.

Data Type	Range	Bits	Explanation	Type
NODATA	n.a.	0	«No data» type	0 (0x00)
ERROR	n.a.	32	Emergency event data type	1 (0x01)
FLOAT	1-bit sign 23-bit fraction 8-bit exponent	32	Single precision foating-point value according to IEEE-754-1985	2 (0x02)
LONG	-2147483647 to 2147483648	32	2's complement integer	3 (0x03)
ULONG	0 to 4294967295	32	unsigned integer	4 (0x04)
BLONG	n.a.	32	Each bit defines a discrete state. 32 bits are coded into four CAN data bytes	5 (0x05)
SHORT	-32768 to +32767	16	2's complement short integer	6 (0x06)
USHORT	0 to 65535	16	unsigned short integer	7 (0x07)
BSHORT	n.a.	16	Each bit defines a discrete state. 16 bits are coded into two CAN data bytes	8 (0x08)
CHAR	-128 to +127	8	2's complement char integer	9 (0x09)
UCHAR	0 to 255	8	unsigned char integer	10 (0x0A)
BCHAR	n.a.	8	Each bit defines a discrete state. 8 bits are coded into a single CAN data byte	11 (0x0B)
SHORT2	-32768 to +32767	2 x 16	2 x 2's complement short integer	12 (0x0C)
USHORT2	0 to 65535	2 x 16	2 x unsigned short integer	13 (0x0D)
BSHORT2	n.a.	2 x 16	2 x discrete short	14 (0x0E)

Data Type	Range	Bits	Explanation	Type
CHAR4	-128 to +127	4 x 8	4 x 2's complement char integer	15 (0x0F)
UCHAR4	0 to 255	4 x 8	4 x unsigned char integer	16 (0x10)
BCHAR4	n.a.	4 x 8	4 x discrete char	17 (0x11)
CHAR2	-128 to +127	2 x 8	2 x 2's complement char integer	18 (0x12)
UCHAR2	0 to 255	2 x 8	2 x unsigned char integer	19 (0x13)
BCHAR2	n.a.	2 x 8	2 x discrete char	20 (0x14)
MEMID	0 to 4294967295	32	Memory ID for upload/download	21 (0x15)
CHKSUM	0 to 4294967295	32	Checksum for upload/download	22 (0x16)
ACHAR	0 to 255	8	ASCII character	23 (0x17)
ACHAR2	0 to 255	2 x 8	2 x ASCII character	24 (0x17)
ACHAR4	0 to 255	4 x 8	4 xASCII character	25 (0x17)
CHAR3	-128 to +127	3 x 8	3 x 2's comple-ment char integer	26 (0x17)
UCHAR3	0 to 255	3 x 8	3 x unsigned char integer	27 (0x17)
BCHAR3	n.a.	3 x 8	3 x discrete char	28 (0x17)
ACHAR3	0 to 255	3 x 8	4 xASCII character	29 (0x17)
DOUBLEH	1-bit sign 52-bit fraction 11-bit exponent	32	Most significant 32 bits of double precision floating-point value according to IEEE-754-1985	30 (0x17)
DOUBLEL	1-bit sign 52-bit fraction 11-bit exponent	32	Least significant 32 bits of double precision floating-point value according to IEEE-754-1985	31 (0x17)
RESVD	n.a.	xx	Reserved for future use	32-99 (0x20-0x63)
UDEF	n.a.	xx	User-defined data types	100-255 (0x64-0xFF)

4 Message structure

The coding of the data into the CAN message bytes is according to the “Big Endian” definition as used by Motorola 68K, SPARC, PowerPC, MIPS and other major processor architectures. All CAN messages consist of 4 header bytes for identification and between 1 and 4 data bytes for the actual data.

The general message format uses a 4-byte message header for node identification, data type, message code and service code (for normal operation data (NOD), the service code field is user-defined). This allows identification of each message by any receiving unit without the need for additional information. Every message type uses the same layout for the CAN data bytes 0-3, while the number and the data type used for CAN data bytes 4-7 is user-defined:

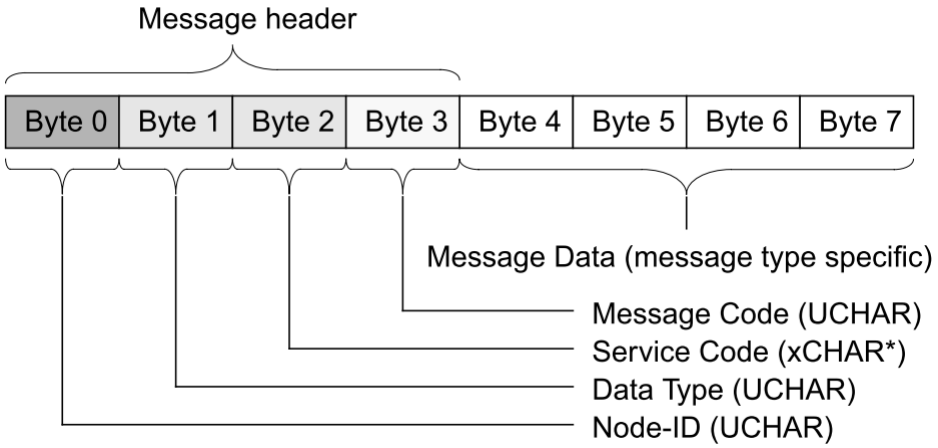


Figure 2 CANaerospace Self-Identifying Message Format

- **Node-ID**: For ATM communication (EED, NOD), the Node Identifier specifies the transmitting node. For PTP communication (NSH, NSL) it specifies the addressed node (client, server). For PTP communication, Node_ID "0" is used to address all stations in the network (multicast).
- **Data Type**: The Data Type specifies how the payload of the message shall be interpreted with respect to its data type (i.e. floating-point data or number of bytes in case of integer data). The corresponding data type code is taken from the CANaerospace data type list which also allows user-defined data type definitions.
- **Service Code**: For Normal Operation Data (NOD) the Service Code delivers information about the integrity of the parameter transmitted with the message. This may be the result of a continuous sensor built-in test, the current validity flag of a navigation signal or other parameter specific information. In case of PTP communication the Service Code specifies the service for the corresponding client/server interaction.
- **Message Code**: For Normal Operation Data (NOD) the Message Code is incremented by one for each message with a particular CAN identifier by the transmitting node. After reaching the value of 255, the Message Code rolls over to zero. This allows receiving stations to determine missing or delayed messages and to react accordingly. Concerning PTP communication (NSH, NSL) the Message Code is used in conjunction with the Service Code to specify the service for the corresponding client/server interaction in more detail.

5 Description of identifiers

Identifiers are taken by default. They can be changed in the USS (UAVOS Servomotor Studio).

5.1 Commands

5.1.1 Position

CAN identifier	Parameter Name	Data Type	Units	Notes
1300 (0x514)	Position	FLOAT	Deg.	-180° ÷ 180°

Example: move to position 20 deg.

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x01	0x02	0x00	0x00	0x41	0xA0	0x00	0x00

5.1.2 Velocity

CAN identifier	Parameter Name	Data Type	Units	Notes
1301 (0x515)	Velocity	FLOAT	Deg./sec	

Example: rotate at -10 deg/sec.

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x01	0x02	0x00	0x00	0xC1	0x20	0x00	0x00

5.1.3 Control

CAN identifier	Parameter Name	Data Type	Units	Notes
1302 (0x516)	Control	FLOAT	Deg. or Deg./sec	-1.0 ÷ 1.0°

Example: if «**Control Mode»** is **Velocity**. Rotate at a velocity of 0.5 of the MaxVelocity.

Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x01	0x02	0x00	0x00	0x3F	0x00	0x00	0x00

5.2 Telemetry Data

5.2.1 Actual Position

CAN identifier	Parameter Name	Data Type	Units	Notes
1350 (0x546)	Actual Position	FLOAT	Deg.	-180 ÷ 180

5.2.2 Actual Control

CAN identifier	Parameter Name	Data Type	Units	Notes
1351 (0x547)	Actual Control	FLOAT	-	-1.0 ÷ 1.0

5.2.3 Actual Velocity

CAN identifier	Parameter Name	Data Type	Units	Notes
1352 (0x548)	Actual Velocity	FLOAT	Deg./sec	-

5.2.4 Actual Current

CAN identifier	Parameter Name	Data Type	Units	Notes
1353 (0x549)	Actual Current	FLOAT	A	-

5.2.5 Actual Voltage

CAN identifier	Parameter Name	Data Type	Units	Notes
1354 (0x54A)	Actual Voltage	FLOAT	V	-

5.2.6 Actual Motor Temperature

CAN identifier	Parameter Name	Data Type	Units	Notes
1355 (0x54B)	Actual Motor Temperature	FLOAT	°C	-

5.2.7 Actual Pcb Temperature

CAN identifier	Parameter Name	Data Type	Units	Notes
1356 (0x54C)	Actual Pcb Temperature	FLOAT	°C	-

5.2.8 Actual Humidity

CAN identifier	Parameter Name	Data Type	Units	Notes
1357 (0x54D)	Actual Humidity	FLOAT	%	0 ÷ 100