MICHAEL STOCK
FLIGHT SYSTEMS

CAN Aerospace

NASA
AGATE data bus

Interface specification for airborne CAN applications
V 1.7

Revision history

Ver.	Who	When	What
1.0	M. Stock	10.10.98	Initial Version
1.05	M. Stock	5.11.98	Identifier lists modified
1.1	M. Stock	10.01.99	MIL connector definitions added
1.2	M. Stock	20.01.99	Navigation identifiers added
1.3	M. Stock	14.04.99	Engine/Fuel system parameter definitions changed, multiple data type support added
1.4	M. Stock	12.07.99	NAV section rearranged, new data types added
1.5	M. Stock	02.02.00	Connector definitions updated, some data normalized
1.6	M. Stock	13.03.01	Redundancy support and time triggered bus scheduling added
1.7	M. Stock	12.01.06	Identifier distribution list expanded, several node services added, clarifications

Copyright statement

CANaerospace is an interface standard open to everyone. No copyrights are reserved and no licenses are necessary for its implementation, use or distribution. Consequently, the author explicitely refuses any responsibility arising from the use of this standard in any application. This document in its current version is available from:

www.canaerospace.com

Comments to this standard are welcome:

Stock Flight Systems
Schützenweg 8a
82335 Berg/Farchach
Germany
phone.: +49-8151-9607-0
fax: +49-8151-9607-30
Email: info@stockflightsystems.com

Front cover photograph: NASA test pilot Rogers Smith flying the X-29 research aircraft over Mojave Desert, California

Table of Contents

Table of Contents

Section **Page**

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 an efficient data flow monitoring and easy time-frame synchronisation 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 Message/data types and identifier assignment

2.1 Message types

The data format definition specifies 6 basic message types, which are used for different network services. Each message type has an associated CAN-ID range defining the message priority. Generally, the identifier distribution within the specified ranges is at the user’s discretion. A standard CANaerospace identifier distribution addressing commonly used data objects and devices in aerospace applications is made in section 5, however. The use of this standard distribution scheme is highly encouraged for interoperability reasons.

Message Type	CAN-ID Range	Explanation
Emergency Event Data (EED)	0 - 127 ($000 - $07F)	Transmitted asynchronously whenever a situation requiring immediate action occurs.
High Priority Node Service Data (NSH)	128 - 199 ($080 - $0C7)	Transmitted asynchronously or cyclic with defined transmission intervals for operational commands (36 channels)
High Priority-User-Defined Data (UDH)	200 - 299 ($0C8 - $12B)	Message/data format and transmission intervals entirely user-defined
Normal Operation Data (NOD)	300 - 1799 ($12C - $707)	Transmitted asynchronously or cyclic with defined transmission intervals for operational and status data.
Low Priority-User-Defined Data (UDL)	1800 - 1899 ($708 - $76B)	Message/data format and transmission intervals entirely user-defined

Message Type	CAN-ID Range	Explanation
Debug Service Data (DSD)	1900 - 1999 ($76C - $7CF)	Transmitted asynchronously or cyclic for debug communication & software download actions.
Low Priority Node Service Data (NSL)	2000 - 2031 $7D0 - $7EF	Transmitted asynchronously or cyclic for test & maintenance actions (16 channels).

2.2 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 3.1.

Data Type	Range	Bits	Explanation	Type #
NODATA	n.a.	0	“No data” type	0 ($00)
ERROR	n.a.	32	Emergency event data type	1 ($01)
FLOAT	1-bit sign 23-bit fraction 8-bit exponent	32	Single precision floating-point value according to IEEE-754-1985	2 ($02)
LONG	-2147483647 to +2147483648	32	2’s complement integer	3 ($03)
ULONG	0 to 4294967295	32	unsigned integer	4 ($04)
BLONG	n.a.	32	Each bit defines a discrete state. 32 bits are coded into four CAN data bytes	5 ($05)
SHORT	-32768 to +32767	16	2’s complement short integer	6 ($06)
USHORT	0 to 65535	16	unsigned short integer	7 ($07)

Data Type	Range	Bits	Explanation	Type #
BSHORT	n.a.	16	Each bit defines a discrete state. 16 bits are coded into two CAN data bytes	8 ($08)
CHAR	-128 to +127	8	2’s complement char integer	9 ($09)
UCHAR	0 to 255	8	unsigned char integer	10 ($0A)
BCHAR	n.a.	8	Each bit defines a discrete state. 8 bits are coded into a single CAN data byte	11 ($0B)
SHORT2	-32768 to +32767	2 x 16	2 x 2’s complement short integer	12 ($0C)
USHORT2	0 to 65535	2 x 16	2 x unsigned short integer	13 ($0D)
BSHORT2	n.a.	2 x 16	2 x discrete short	14 ($0E)
CHAR4	-128 to +127	4 x 8	4 x 2’s complement char integer	15 ($0F)
UCHAR4	0 to 255	4 x 8	4 x unsigned char integer	16 ($10)
BCHAR4	n.a.	4 x 8	4 x discrete char	17 ($11)
CHAR2	-128 to +127	2 x 8	2 x 2’s complement char integer	18 ($12)
UCHAR2	0 to 255	2 x 8	2 x unsigned char integer	19 ($13)
BCHAR2	n.a.	2 x 8	2 x discrete char	20 ($14)
MEMID	0 to 4294967295	32	Memory ID for upload/download	21 ($15)
CHKSUM	0 to 4294967295	32	Checksum for upload/download	22 ($16)
ACHAR	0 to 255	8	ASCII character	23 ($17)

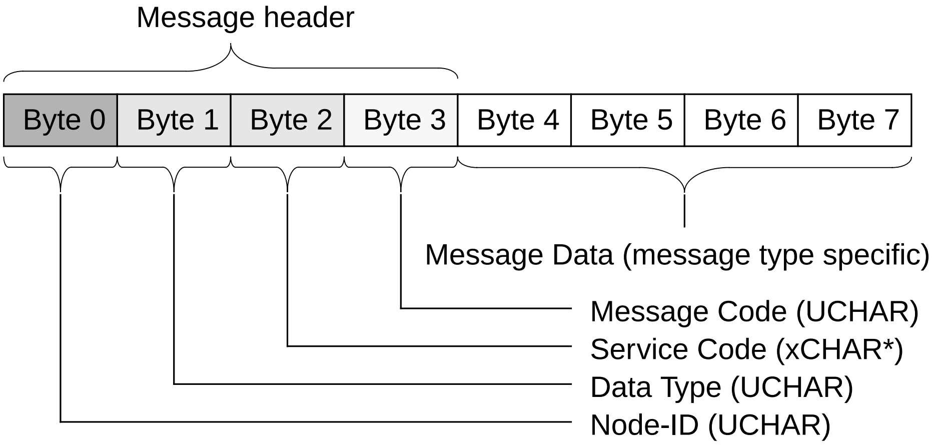
Data Type	Range	Bits	Explanation	Type #
ACHAR2	0 to 255	2 x 8	2 x ASCII character	24 ($18)
ACHAR4	0 to 255	4 x 8	4 xASCII character	25 ($19)
CHAR3	-128 to +127	3 x 8	3 x 2's complement char integer	26 ($1A)
UCHAR3	0 to 255	3 x 8	3 x unsigned char integer	27 ($1B)
BCHAR3	n.a.	3 x 8	3 x discrete char	28 ($1C)
ACHAR3	0 to 255	3 x 8	4 xASCII character	29 ($1D)
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 ($1E)
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 ($1F)
RESVD	n.a.	xx	Reserved for future use	32-99 ($20-$63)
UDEF	n.a.	xx	User-defined data types	100-255 ($64-$FF)

3 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.

3.1 General message format

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:



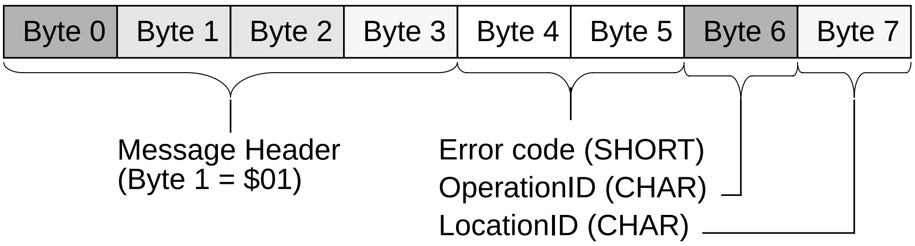
*: xCHAR may be CHAR, ACHAR, BCHAR or UCHAR

The header data fields have the following meaning:

- The node-ID is in the range of 0-255 with node-ID “0” being the broadcast-ID referring to “all nodes”. Note that for emergency event data (EED) and normal operation data (NOD) messages, the node-ID identifies the transmitting station, while for node service data (NSH/NSL) messages the node-ID identifies the addressed station.
- The data type specifies the coding of the data transported with the corresponding message. The number is taken from the data type list (see section 2.2).
- For normal operation data (NOD) messages, the message code is incremented by one for each message and may be used to monitor the sequence of incoming messages. The message code then rolls over to zero after passing 255. This feature allows any node in the network to determine the age of a signal and the proper sequence for monitoring purposes. For node service data (NSL/NSH) messages, however, the message code is used for extended specification of the service.
- For normal operation data (NOD) messages, the service code consists of 8 bits which may be used as required by the specific data (should be set to zero if unused). For node service data (NSL/NSH) messages, the service code contains the node service code for the current operation.

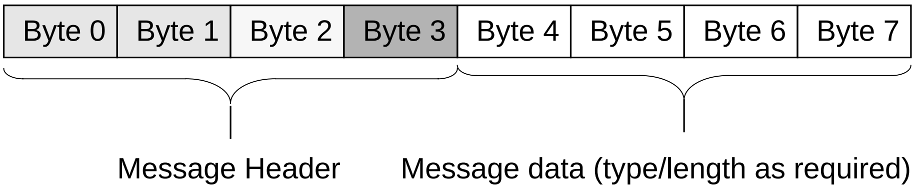
3.2 Emergency Event Data (EED) message format

Emergency Event Data (EED) is transmitted asynchronously by the affected unit whenever an error situation occurs. The corresponding data contains information about the location within the unit at which the error ocurred, the offending operation and the error code:



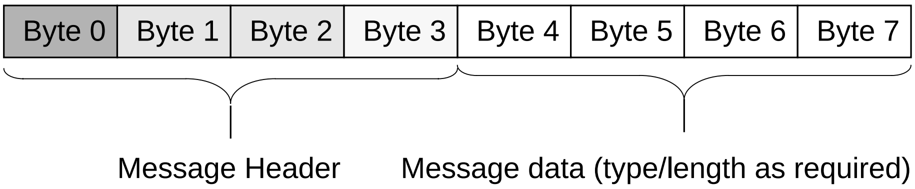
3.3 Normal Operation Data (NOD) message format

Normal Operation Data (NOD) is transmitted during normal operation, either cyclic or asynchronously. The data type (and therefore the message byte count) is taken from the data type list:



3.4 Node Service Data (NSH/NSL) message format

Node Service Data (NSH/NSL) is data associated to the node service protocol as specified in section 4. The message format is similar to NOD. Node service data, however, is transmitted on specific identifiers only:



3.5 Debug Service Data (DSD) message format

The Debug Service Data message format is entirely user-defined because of the specific requirements resulting from the various host/target communication protocols. Aside from using the specified identifier range, no restrictions apply. To maximize flexibility, the message layout and the data types must not follow any of the CANaerospace definitions. It is encouraged, however, to use the proposed standard.

3.6 User-Defined Data (UDL/UDH) message format

User-Defined Data message formats may be created for specific purposes. Aside from using the specified identifier range, no restrictions apply. To maximize flexibility, the message layout and the data types must not follow any of the CANaerospace definitions. It is encouraged, however, to use the proposed standard.

4 Node service protocol

In parallel to the data transfer during normal operation (Emergency Event Data, Normal Operation Data), the node service protocol provides a connection-oriented communication using a handshake mechanism. This protocol has been implemented to support command/ response type connections between two nodes for specific operations, i.e. for data download or client/server actions. Note that node service requests requiring action but no response are possible as well. Requests of this type may be sent to a specific or all nodes (broadcast).

The node service protocol may be run either in high priority or low priority mode, selected by identifier. For the high priority mode, 36 node service communication channels are available, while the low priority mode offers 16 communication channels. Each communication channel consists of one CAN identifier for the node service request and the immediately following one for the node service response. The identifier assignment for the high priority node service channels is as follows:

Node Service Channel	Node Service Request ID	Node Service Response ID
0	128 ($080)	129 ($081)
1	130 ($082)	131 ($083)
2	132 ($084)	133 ($085)
....	....	......
....	....	......
33	194 ($0C2)	195 ($0C3)
34	196 ($0C4)	197 ($0C5)
35	198 ($0C6)	199 ($0C7)

This is the identifier assignment for the low priority node service channels:

Node Service Channel	Node Service Request ID	Node Service Response ID
100	2000 ($7D0)	2001 ($7D1)
101	2002 ($7D2)	2003 ($7D3)

Node Service Channel	Node Service Request ID	Node Service Response ID
102	2004 ($7D4)	2005 ($7D5)
....	....	......
115	2030($7EE)	2031 ($7EF)

A node service is initiated by a node service request message, transmitted on the corresponding identifier. All nodes attached to the network are obliged to continously monitor these identifiers and check if received messages contain the own personal node-ID. If a match is detected, the corresponding node has to react by performing the required action and transmitting a node service response message on the corresponding identifier within 100ms (if this was required by the request type). The node service response must again contain the personal node-ID of the addressed node. Any node in the network is allowed to initiate node services. It is recommended, however, that each node in the network initiating node service requests uses a dedicated node service channel to avoid potential hand-shaking conflicts. The channel on which a particular node service is run may be defined by the user. If only one service channel is used, node services must be run on channel 0 by default. Some frequently used types of node services are already specified below, other services may be added as required.

Each CANaerospace unit must support at least the Identification Service (IDS) on Node Service Channel 0. This makes sure that a CANaerospace network can be scanned for attached units to determine their status, header type and identifier assignment. Note that within a CANaerospace network, other header types than the standard CANaerospace header and several identifier assignment schemes (including entirely user-defined ones) are supported. Whenever possible, it is strongly recommended to use the proposed standard header and identifier assignment, however.

Node Service	Service Code	Response Required	Action
IDS	0	Yes	Identification service. Requests a “sign-of-life” response together with configuration information from the addressed node.
NSS	1	No	Node synchronisation service, used to trigger a specific node or to perform a network wide time synchronisation.

Node Service	Service Code	Response Required	Action
DDS	2	Yes	Data download service. Sends a block of data to another node.
DUS	3	Yes	Data upload service. Receives a block of data from another node.
SCS	4	Yes	Simulation Control Service. Allows to change the behaviour of the addressed node by controlling internal simulation software functions.
TIS	5	Yes	Transmission Interval Service. Sets the transmission rate of a specific CAN message transmitted by the addressed node.
FPS	6	Yes	FLASH Programming Service. Triggers a node-internal routine to store configuration data permantently into non-volatile memory.
STS	7	No	State Transmission Service. Causes the addressed node to transmit all its CAN messages once.
FSS	8	Yes	Filter Setting Service. Used to modify the limit frequencies of node-internal highpass, lowpass or bandpass filters.
TCS	9	Yes	Test Control Service. Triggers internal test functions of the addressed node.
BSS	10	No/Yes (result dependant)	CAN Baudrate Setting Service. Sets the CAN baudrate ot the addressed node.
NIS	11	Yes	Node-ID Setting Service. Sets the Node-ID of the addressed node.
MIS	12	Yes	Module Information Service. Obtains information about installed modules within the addressed node.

Node Service	Service Code	Response Required	Action
MCS	13	Yes	Module Configuration Service. Configures installed modules within the addressed node.
CSS	14	Yes	CAN-ID setting service. Sets the CAN-ID of a specific message the addressed node transmits.
DSS	15	Yes	CAN-ID Distribution Setting Service. Sets the identifier distribution code transmitted as Byte 2 of the response to an IDS request.
XXS	16-99		Reserved for future use.
	100-255		User-defined services.

4.1 Identification Service (IDS)

The identification service is a client/server type service. It is used to obtain a “sign-of-life” indication from the addressed node and check if its identifier distribution and message header format is compliant with the network that it is attached to. The addressed node returns a 4 byte status information which contains information about hardware/software revision of the system, together with the identifier distribution and message header format the node is programmed for. Consequently, the data type of the response message is UCHAR4.

The term “identifier distribution” refers to the assignment of CAN identifiers to physical data objects. This distribution may either be as described in section 5 (CANaerospace standard) or of a different type:

Distribution ID (Byte 2 of IDS response)	Identifier Distribution
0	CANaerospace standard
1-99	Reserved for future use
100-255	User-defined distribution schemes

Likewise, a CANaerospace network may choose to deviate from the standard CANaerospace header format as described in section 3.1. If this is the case, byte 3 of the message must inform an interrogating node about this fact by returning a non-zero (user-defined) value in byte 3 of the IDS response. Note that in this case the IDS service must still be supported using the CANaerospace standard header format:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	NODATA	UCHAR4
2	Service Code	0	0
3	Message Code	<0>	<as in request>
4-7	Message Data	n.a.	Byte 0: Hardware Revision Byte 1: Software Revision Byte 2: Identifier Distribution (0 = CANaerospace standard distribution) Byte 3: Header Type (0 = CANaerospace standard header)

4.2 Node Synchronisation Service (NSS)

The node synchronisation service is a connectionless service (no service response required) used to perform time synchronisation of all nodes attached to the network. Therefore, the node-ID is set to 0. The time stamp may be used to submit a 32 bit value for clock settings:

Message Data Byte	Data Field Description	Service Request
0	Node-ID	0
1	Data type	ULONG
2	Service Code	1
3	Message Code	0
4-7	Message Data	<time stamp>

4.3 Data Download Service (DDS)

The data download is a connection-oriented service and is used to send a block of data to another node. The size of the data block may be in the range of 1-1020 bytes, specified by the message number field of the header. To initiate the service, the requesting station sends a “start download request message” to the addressed node, specifying a memory destination identifier and the number of messages which will be transmitted (the data type may change during the download to allow structured data to be sent). It then waits for the response. If the service response is received within 100ms and the message data is XON, the requesting station may transmit the specified number of data messages:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	MEMID	LONG
2	Service Code	2	2
3	Message Code	<0-255>	<as in request>
4-7	Message Data	<memory destination identifier>	-2 = INVALID -1 = ABORT 0 = XOFF 1 = XON

The addressed node now accepts data until the final message number has been reached. To control download speed, the addressed node may send a service response at any time during the download process, specifying the current message number and XOFF or XON. By specifying ABORT or INVALID, the download is cancelled immediately without further action. The transmitting station has to react correspondingly by stopping or resuming data transmission.

After the last message has been received, the addressed node transmits a service response with a checksum calculated from summing up all received data. This allows the requesting node to determine if all data has been received properly:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data type	<any>	CHKSUM
2	Service Code	2	2
3	Message Code	<last number>	<last number>
4-7	Message Data	<download data>	<checksum>

4.4 Data Upload Service (DUS)

The data upload is a connection-oriented service and is used to receive a block of data from another node. The size of the data block may be in the range of 1-1020 bytes, specified by the message number field of the header. To initiate the service, the requesting station sends a “start upload request message” to the addressed node, specifying the source memory identifier and the number of messages which is expected to be received. It then waits up to 100ms for a service response. After having transmitted the service response, the addressed station waits 10ms and then transmits the requested number of data messages:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	MEMID	LONG
2	Service Code	3	3
3	Message Code	<0-255>	<as in request>
4-7	Message Data	<source memory identifier>	-1 = ABORT 0 = OK

The requesting node now accepts data at the maximum transmission speed until the final message number has been reached. After the last data message, the addressed node transmits a service response with a checksum calculated from summing up all transmitted data. This allows the requesting node to determine if all data has been received properly. Additionally, the requesting node continously checks the proper message number sequence during the process to detect failures:

Message Data Byte	Data Field Description	Service Response
0	Node-ID	<node-ID>
1	Data Type	CHKSUM
2	Service Code	3
3	Message Code	<last number>
4-7	Message Data	<checksum>

4.5 Simulation Control Service (SCS)

The simulation control service is a connection-oriented service and is used to change the behaviour of the addressed node by controlling internal simulation software functions. Those functions are usually implemented to support the use of the equipment in a simulated environment where not all external interfaces are available and vary widely between systems. Consequently, the format of this service is mostly user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	<user-defined>	<user-defined>
2	Service Code	4	4
3	Message Code	<user-defined>	<user-defined>
4-7	Message Data	<user-defined>	<user-defined>

4.6 Transmission Interval Service (TIS)

The transmission interval service is a connection-oriented service and is used to set the transmission rate of a specific CAN message transmitted by the addressed node. This is accomplished by specifying the corresponding CAN identifier and the desired transmission rate for this data in milliseconds. The allowed range for the CAN identifier and the transmission rate is user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	SHORT2	NODATA
2	Service Code	5	5
3	Message Code	0	0 = OK -6 = CAN identifier or transmission rate out of range
4-5	Message Data	<CAN identifier>	n.a.
6-7	Message Data	<transmission rate in ms>	n.a.

4.7 FLASH Programming Service (FPS)

The FLASH programming service is a connection-oriented service used to store configuration data (i.e. settings made through BSS, NIS, CSS or similar services) into internal non-volatile memory. It usually triggers a node-internal software programming routine. A user-defined security code (transmitted as the message code) makes sure that this service can only be issued by “authorized” nodes in the network:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	NODATA	NODATA
2	Service Code	6	6
3	Message Code	<security code>	0 = OK -3 = invalid security code
4-7	Message Data	n.a.	n.a.

4.8 State Transmission Service (STS)

The state transmission service is a connectionless service which causes the addressed node to transmit all its CAN messages once. This service is used by other nodes in the network which need to to obtain current data that is normally transmitted upon state change only:

Message Data Byte	Data Field Description	Service Request
0	Node-ID	<node-ID>
1	Data Type	NODATA
2	Service Code	7
3	Message Code	0
4-7	Message Data	n.a.

4.9 Filter Setting Service (FSS)

The filter setting service is a connection-oriented service used to set the limit frequency for node-internal highpass, lowpass or bandpass filter functions:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	FLOAT	NODATA
2	Service Code	8	8
3	Message Code	<filter number>	0 = OK -6 = limit frequency or filter number out of range
4-7	Message Data	<limit frequency>	n.a.

4.10 Test Control Service (TCS)

The test control is a connection-oriented service and is used to control internal test functions of the addressed node. Such functions are usually implemented to support in-service testing of the equipment and vary widely between systems. Consequently, the format of this service is mostly user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	<user-defined>	<user-defined>
2	Service Code	9	9
3	Message Code	<user-defined>	<user-defined>
4-7	Message Data	<user-defined>	<user-defined>

4.11 Baudrate Setting Service (BSS)

The baudrate setting service is a (normally) connectionless service that modifies the CAN baudrate of the addressed node. The baudrate change becomes effective immediately and therefore no response to this service is transmitted in case of success. In case of failure, however, the node maintains its original baudrate and returns an error code:

Message Data Byte	Data Field Description	Service Request	Service Response (not transmitted when request was successful)
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	SHORT	NODATA
2	Service Code	10	10
3	Message Code	0	-1 = baudrate code unknown
4-5	Message Data	<baudrate code> 0 = 1 MBit/s 1 = 500 kBit/s 2 = 250 kBit/s 3 = 125 kBit/s	n.a.

4.12 Node-ID Setting Service (NIS)

The Node-ID setting service is a connection-oriented service used to set the ID of the addressed node. The new ID becomes effective immediately and is used by the affected node a.) to identify itself in the Node-ID field for all normal operation data (NOD) messages and b.) to determine if it is addressed by received node service requests:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	NODATA	NODATA
2	Service Code	11	11
3	Message Code	<new node-ID> 1 <= ID <= 199	0 = OK -6 = node-ID out of range
4-7	Message Data	n.a.	n.a.

4.13 Module Information Service (MIS)

The module information service is a connection-oriented service that returns information about modules installed in the addressed node. The nature of these “modules” varies widely between systems. Consequently, the format of this service is mostly user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	<user-defined>	<user-defined>
2	Service Code	12	12
3	Message Code	<user-defined>	<user-defined>
4-7	Message Data	<user-defined>	<user-defined>

4.14 Module Configuration Service (MCS)

The module configuration service is a connection-oriented service that configures modules installed in the addressed node. The nature of these “modules” varies widely between systems. Consequently, the format of this service is mostly user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	<user-defined>	<user-defined>
2	Service Code	13	13
3	Message Code	<user-defined>	<user-defined>
4-7	Message Data	<user-defined>	<user-defined>

4.15 CAN Identifier Setting Service (CSS)

The CAN identifier setting service is a connection-oriented service used to set the CAN identifier of a specific CAN message transmitted by the addressed node. This is accomplished by specifying the message through a unique “message number” together with the desired CAN identifier for this message. The allowed range for the message number and the CAN identifier is user-defined:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	SHORT2	NODATA
2	Service Code	14	14
3	Message Code	0	0 = OK -6 = message number or CAN identifier out of range
4-5	Message Data	<message number>	n.a.
6-7	Message Data	<CAN identifier>	n.a.

4.16 CANaerospace Identifier Distribution Setting Service (DSS)

The CANaerospace identifier distribution setting service is a connection-oriented service used to set the identifier distribution ID of the addressed node. The new distribution ID becomes effective immediately and will be returned by the affected node in byte 2 of the response to an identification service (IDS) request:

Message Data Byte	Data Field Description	Service Request	Service Response
0	Node-ID	<node-ID>	<node-ID>
1	Data Type	NODATA	NODATA
2	Service Code	15	15
3	Message Code	<new distribution ID> 0<= ID <= 240	0 = OK -6 = distribution ID out of range
4-7	Message Data	n.a.	n.a.

5 CANaerospace default identifier distribution

In addition to basic message formats and the node service protocol, CANaerospace defines an “identifier distribution” (also referred to as a “profile”) as a means to effectively support interoperability between nodes. Together with the self-identifying message format and the time-triggered bus scheduling, this concept makes sure that nodes designed by different vendors may be integrated into an already existing CANaerospace network quickly and with minimum effort.

By default, an identifier distribution scheme addressing the most commonly used data for aerospace applications has been specified. For this purpose, the available identifiers for normal operation data have been grouped for the various aircraft systems, thereby reserving the identifier range 300-1499. The identifiers from 1500-1799 are unassigned and may be used for other data at the user’s discretion.

Note that other identifier distribution schemes besides this default one may be developed and will eventually be added in future releases of the CANaerospace specification. Likewise, user-defined identifier distribution schemes, derived from the default, or with entirely different content are supported by CANaerospace. It is therefore extremely important that every node provides information about the identifier distribution it uses in response to an identification (IDS) node service request. This allows any node in the network to interrogate the entire range of node-IDs (1 to 255) and determine if its own identifier distribution scheme is compatible with the one used in the network.

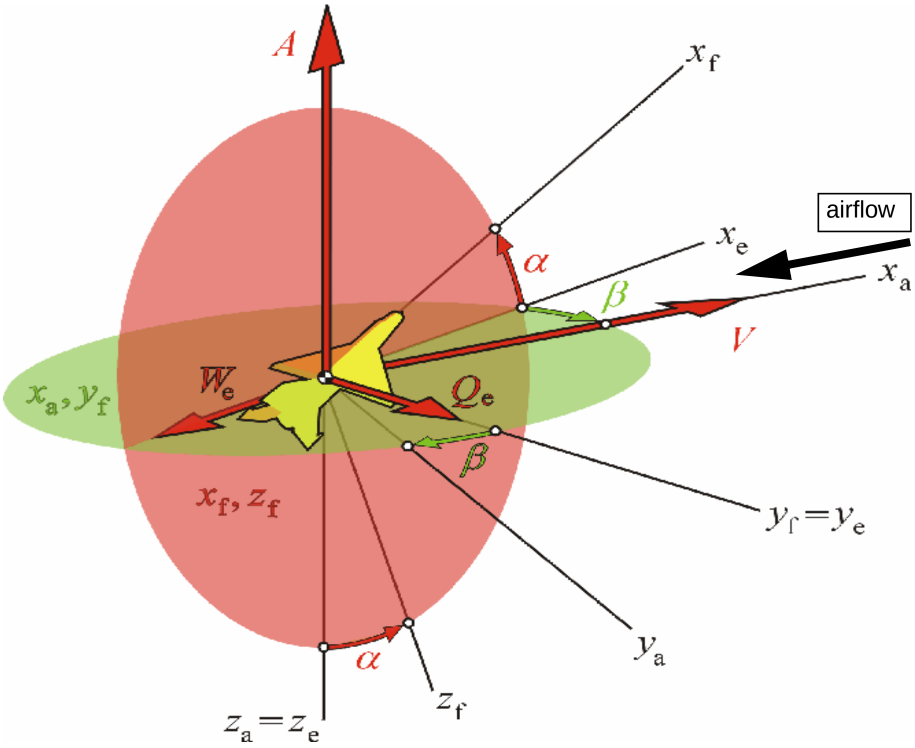
Parameters may support several data types if indicated in the “suggested data types” field. If this is the case, the user may select the appropriate type with respect to system requirements, processor performance, etc. If analogue parameters are transmitted as SHORT2 using this identifier distribution, the first SHORT variable contains the current value, while the second SHORT variable contains the maximum value of this parameter (to support parameter scaling for the receiving nodes) as described in the following example:

$$\text{SensorValue} = \text{AnalogValue}/\underbrace{(32767/\text{AnalogScale})}_{\text{AnalogFactor}}$$

Example:
EGT range: 0-1500K
AnalogScale = 1500 ($05DC)
AnalogFactor = 32767/1500 = 21.84
AnalogValue = 16384 ($4000)
SensorValue = 16384/21.84 = 750K

$7C	$0C	$00	$4A	$40	$00	$05	$DC
Message Header				AnalogValue (variable)		AnalogScale (constant)	

The data units are SI units according to ISO 1000 with some exceptions (hPa, deg/s). All data referring to aircraft axis systems are based on ISO 1151-1 and ISO 1151-2 (equivalent to LN9300) as shown in the picture below:



Symbol	Definition
x_f, y_f, z_f	Body axis system
x_a, y_a, z_a	Airpath axis system
x_e, y_e, z_e	Intermediate axis system
V	Velocity vector
A	Lift vector
Q_e	Transverse force vector
W_e	Drag vector
α	Angle of attack
β	Angle of sideslip

These axis systems define the signs used for flight state data. The signs used for the cockpit flight controls are not specified in ISO 1151 and therefore defined according to international conventions.

5.1 Flight state/air data

CAN identifier	Flight state parameter name	Suggested data types	Units	Notes
300 ($12C)	Body longitudinal acceleration	FLOAT SHORT2	g	forward: + aft: -
301 ($12D)	Body lateral acceleration	FLOAT SHORT2	g	right: + left: -
302 ($12E)	Body normal acceleration	FLOAT SHORT2	g	up: + down: -
303 ($12F)	Body pitch rate	FLOAT SHORT2	deg/s	nose up: + nose down: -
304 ($130)	Body roll rate	FLOAT SHORT2	deg/s	roll right: + roll left: -
305 ($131)	Body yaw rate	FLOAT SHORT2	deg/s	yaw right: + yaw left: -
306 ($132)	Rudder position	FLOAT SHORT2	deg.	clockwise deflection: + counterclock-wise deflection: -
307 ($133)	Stabilizer position	FLOAT SHORT2	deg.	deflection downward: + deflection upward: -
308 ($134)	Elevator position	FLOAT SHORT2	deg.	deflection downward: + deflection upward: -
309 ($135)	Left aileron position	FLOAT SHORT2	deg.	deflection downward: + deflection upward: -
310 ($136)	Right aileron position	FLOAT SHORT2	deg.	deflection downward: + deflection upward: -
311 ($137)	Body pitch angle	FLOAT SHORT2	deg.	nose up: + nose down: -
312 ($138)	Body roll angle	FLOAT SHORT2	deg.	roll right: + roll left: -

CAN identifier	Flight state parameter name	Suggested data types	Units	Notes
313 ($139)	Body sideslip	FLOAT SHORT2	deg.	yaw right: + yaw left: -
314 ($13A)	Altitude rate	FLOAT SHORT2	m/s	
315 ($13B)	Indicated airspeed	FLOAT SHORT2	m/s	
316 ($13C)	True airspeed	FLOAT SHORT2	m/s	
317 ($13D)	Calibrated airspeed	FLOAT SHORT2	m/s	
318 ($13E)	Mach number	FLOAT SHORT2	Mach	
319 ($13F)	Baro Correction	FLOAT SHORT2	hPa	
320 ($140)	Baro corrected altitude	FLOAT SHORT2	m	
321 ($141)	Heading angle	FLOAT SHORT2	deg	+/- 180^o
322 ($142)	Standard altitude	FLOAT SHORT2	m	
323 ($143)	Total air temperature	FLOAT SHORT2	K	
324 ($144)	Static air temperature	FLOAT SHORT2	K	
325 ($145)	Differential pressure	FLOAT SHORT2	hPa	
326 ($146)	Static pressure	FLOAT SHORT2	hPa	
327 ($147)	Heading rate	FLOAT SHORT2	deg/s	
328 ($148)	Port side angle-of-attack (AOA)	FLOAT SHORT2	deg	nose up: + nose down: -
329 ($149)	Starbord side angle-of-attack (AOA)	FLOAT SHORT2	deg	nose up: + nose down: -

CAN identifier	Flight state parameter name	Suggested data types	Units	Notes
330 ($14A)	Density altitude	FLOAT SHORT2	m	
331 ($14B)	Turn coordination rate	FLOAT SHORT2	deg/s	
332 ($14C)	True altitude	FLOAT SHORT2	m	
333 ($14D)	Wind speed	FLOAT SHORT2	m/s	
334 ($14E)	Wind direction	FLOAT SHORT2	deg	+/- 180^o
335 ($14F)	Outside air temperature (OAT)	FLOAT SHORT2	K	
336 ($150)	Body normal velocity	FLOAT SHORT2	m/s	
337 ($151)	Body longitudinal velocity	FLOAT SHORT2	m/s	
338 ($152)	Body lateral velocity	FLOAT SHORT2	m/s	
339 ($153)	Total pressure	FLOAT SHORT2	hPa	

5.2 Flight controls data

The flight controls section also contains aircraft engine controls. This data is divided into sections A and B to support dual redundant engine control systems (ECS). All engine data is available with two different CAN identifiers so that it may be transmitted on the same bus without interference.

In case this feature is not used, ECS channel A may also be interpreted as “starboard engines” and ECS channel B as “port engines” so that in total, up to eight engines per aircraft may be supported.

CAN identifier	Flight controls parameter name	Suggested data types	Units	Notes
400 ($190)	Pitch control position	FLOAT SHORT2	Norm -1/+1	aft: + forward: -

CAN identifier	Flight controls parameter name	Suggested data types	Units	Notes
401 ($191)	Roll control position	FLOAT SHORT2	Norm -1/+1	right: + left: -
402 ($192)	Lateral stick trim position command	FLOAT SHORT2	Norm -1/+1	right: + left: -
403 ($193)	Yaw control position	FLOAT SHORT2	Norm -1/+1	left pedal: - right pedal: +
404 ($194)	Collective control position	FLOAT SHORT2	Norm -1/+1	down: - up: +
405 ($195)	Longitudinal stick trim position command	FLOAT SHORT2	Norm -1/+1	aft: + forward: -
406 ($196)	Directional pedals trim position command	FLOAT SHORT2	Norm -1/+1	left pedal: - right pedal: +
407 ($197)	Collective lever trim position command	FLOAT SHORT2	Norm -1/+1	down: - up: +
408 ($198)	Cyclic control stick switches	BLONG BSHORT		
409 ($199)	Lateral trim speed	FLOAT SHORT2	Norm -1/+1	
410 ($19A)	Longitudinal trim speed	FLOAT SHORT2	Norm -1/+1	
411 ($19B)	Pedal trim speed	FLOAT SHORT2	Norm -1/+1	
412 ($19C)	Collective trim speed	FLOAT SHORT2	Norm -1/+1	
413 ($19D)	Nose wheel steering handle position	FLOAT SHORT2	Norm -1/+1	right: + left: -
414 - 417 ($19E - $1A1)	Engine #n throttle lever position (1 <= n <= 4) ECS channel A	FLOAT SHORT2	Norm -1/+1	
418 - 421 ($1A2 - $1A5)	Engine #n condition lever position (1 <= n <= 4) ECS channel A	FLOAT SHORT2	Norm -1/+1	

CAN identifier	Flight controls parameter name	Suggested data types	Units	Notes
422 - 425 ($1A6 - $1A9)	Engine #n throttle lever position (1 <= n <= 4) ECS channel B	FLOAT SHORT2	Norm -1/+1	
426 - 429 ($1AA - $1AD)	Engine #n condition lever position (1 <= n <= 4) ECS channel B	FLOAT SHORT2	Norm -1/+1	
430 ($1AE)	Flaps lever position	FLOAT SHORT2	Norm -1/+1	
431 ($1AF)	Slats lever position	FLOAT SHORT2	Norm -1/+1	
432 ($1B0)	Park brake lever position	FLOAT SHORT2	Norm -1/+1	
433 ($1B1)	Speedbrake lever position	FLOAT SHORT2	Norm -1/+1	
434 ($1B2)	Throttle max lever position	FLOAT SHORT2	Norm -1/+1	
435 ($1B3)	Pilot left brake pedal position	FLOAT SHORT2	Norm -1/+1	
436 ($1B4)	Pilot right brake pedal position	FLOAT SHORT2	Norm -1/+1	
437 ($1B5)	Copilot left brake pedal position	FLOAT SHORT2	Norm -1/+1	
438 ($1B6)	Copilot right brake pedal position	FLOAT SHORT2	Norm -1/+1	
439 ($1B7)	Trim system switches	BLONG BSHORT		
440 ($1B8)	Trim system lights	BLONG BSHORT		
441 ($1B9)	Collective control stick switches	BLONG BSHORT		helicopters only
442 ($1BA)	Stick shaker stall warning device	BLONG BSHORT		

5.3 Aircraft engine/ fuel supply system data

Aircraft engine data is divided into sections A and B to support dual redundant engine control systems (ECS). All engine data is available with two different CAN identifiers so that it may be transmitted on the same bus without interference.

In case this feature is not used, ECS channel A may also be interpreted as “starboard engines” and ECS channel B as “port engines” so that in total, up to eight engines per aircraft can be supported.

CAN identifier	Engine parameter name	Suggested data types	Units	Notes
500 - 503 ($1F4 - $1F7)	Engine #n N1 (1 <= n <= 4) ECS channel A	FLOAT SHORT2	1/min	N1 for jet engines, crankshaft RPM for piston engines
504 - 507 ($1F8 - $1FB)	Engine #n N2 (1 <= n <= 4) ECS channel A	FLOAT SHORT2	1/min	N2 for jet engines, propeller RPM for piston engines
508 - 511 ($1FC - $1FF)	Engine #n torque (1 <= n <= 4) ECS channel A	FLOAT SHORT2	Norm -1/+1	
512 - 515 ($200 - $203	Engine #n turbine inlet temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	TIT
516 - 519 ($204 - $207)	Engine #n interturbine temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	ITT
520 - 523 ($208 - $20B)	Engine #n turbine outlet temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	TOT
524 - 527 ($20C - $20F)	Engine #n fuel flow rate (1 <= n <= 4) ECS channel A	FLOAT SHORT2	l/h	
528 - 531 ($210 - $213)	Engine #n manifold pressure (1 <= n <= 4) ECS channel A	FLOAT SHORT2	hPa	piston engines only

CAN identifier	Engine parameter name	Suggested data types	Units	Notes
532 - 535 ($214 - $217)	Engine #n oil pressure (1 <= n <= 4) ECS channel A	FLOAT SHORT2	hPa	
536 - 539 ($218 - $21B)	Engine #n oil temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	
540 - 543 ($21C - $21F)	Engine #n cylinder head temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	piston engines only
544 - 547 ($220 - $223)	Engine #n oil quantity (1 <= n <= 4) ECS channel A	FLOAT SHORT2	l	
548 - 551 ($224 - $227)	Engine #n cooland temperature (1 <= n <= 4) ECS channel A	FLOAT SHORT2	K	
552 - 555 ($228 - $22B)	Engine #n power rating (1 <= n <= 4) ECS channel A	FLOAT SHORT2	Norm -1/+1	
556 - 559 ($22C - $22F)	Engine #n Status 1 (1 <= n <= 4) ECS channel A	BSHORT BLONG		bit encoding user defined
560 - 563 ($230 - $233)	Engine #n Status 2 (1 <= n <= 4) ECS channel A	BSHORT BLONG		bit encoding user defined
564 - 567 ($234 - $237)	Engine #n N1 (1 <= n <= 4) ECS channel B	FLOAT SHORT2	1/min	N1 for jet engines, crankshaft RPM for piston engines
568 - 571 ($238 - $23B	Engine #n N2 (1 <= n <= 4) ECS channel B	FLOAT SHORT2	1/min	N2 for jet engines, propeller RPM for piston engines

CAN identifier	Engine parameter name	Suggested data types	Units	Notes
572 - 575 ($23C - $23F)	Engine #n torque (1 <= n <= 4) ECS channel B	FLOAT SHORT2	Norm -1/+1	
576 - 579 ($240 - $243)	Engine #n turbine inlet temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	TIT
580 - 583 ($244 - $247)	Engine #n interturbine temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	ITT
584 - 587 ($248 - $24B)	Engine #n turbine outlet temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	TOT for jet engines, exhaust gas temperature for piston engines
588 - 591 ($24C - $24F)	Engine #n fuel flow rate (1 <= n <= 4) ECS channel B	FLOAT SHORT2	l/h	
592 - 595 ($250 - $253)	Engine #n manifold pressure (1 <= n <= 4) ECS channel B	FLOAT SHORT2	hPa	piston engines only
596 - 599 ($254 - $257)	Engine #n oil pressure (1 <= n <= 4) ECS channel B	FLOAT SHORT2	hPa	
600 - 603 ($258 - $25B)	Engine #n oil temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	
604 - 607 ($25C - $25F)	Engine #n cylinder head temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	piston engines only
608 - 611 ($260 - $263)	Engine #n oil quantity (1 <= n <= 4) ECS channel B	FLOAT SHORT2	l	

CAN identifier	Engine parameter name	Suggested data types	Units	Notes
612 - 615 ($264 - $267)	Engine #n coolant temperature (1 <= n <= 4) ECS channel B	FLOAT SHORT2	K	
616 - 619 ($268 - $26B)	Engine #n power rating (1 <= n <= 4) ECS channel B	FLOAT SHORT2	Norm -1/+1	
620 - 623 ($26C - $26F)	Engine #n Status 1 (1 <= n <= 4) ECS channel B	BSHORT BLONG		bit encoding user defined
624 - 627 ($270 - $273)	Engine #n Status 2 (1 <= n <= 4) ECS channel B	BSHORT BLONG		bit encoding user defined
628 - 659 ($274 - $293)	Reserved for future use			
660 - 667 ($294 - $29B)	Fuel pump #n flow rate (1 <= n <= 8)	FLOAT SHORT2	l/h	
668 - 675 ($29C - $2A3)	Fuel tank #n quantity (1 <= n <= 8)	FLOAT SHORT2	kg	
676 - 683 ($2A4 - $2AB)	Fuel tank #n temperature (1 <= n <= 8)	FLOAT SHORT2	K	
684 - 691 ($2AC - $2B3)	Fuel system #n pressure (1 <= n <= 8)	FLOAT SHORT2	hPa	

5.4 Power transmission system data

CAN identifier	Transmission system parameter name	Suggested data types	Units	Notes
700 - 703 ($2BC - $2C0)	Rotor #n RPM (1 <= n <= 4)	FLOAT SHORT2	1/min	helicopters only
704 - 711 ($2BD - $2C7)	Gearbox #n speed (1 <= n <= 8)	FLOAT SHORT2	1/min	
712 - 719 ($2BC - $2CF)	Gearbox #n oil pressure (1 <= n <= 8)	FLOAT SHORT2	hPa	
720 - 727 ($2D0 - $2D7)	Gearbox #n oil temperature (1 <= n <= 8)	FLOAT SHORT2	K	
728 - 735 ($2D8 - $2DF)	Gearbox #n oil quantity (1 <= n <= 8)	FLOAT SHORT2	l	

5.5 Hydraulic system data

CAN identifier	Hydraulic system parameter name	Suggested data types	Units	Notes
800 - 807 ($320 - $327)	Hydraulic system #n pressure (1 <= n <= 8)	FLOAT SHORT2	hPa	
808 - 815 ($328 - $32F)	Hydraulic system #n fluid temperature (1 <= n <= 8)	FLOAT SHORT2	K	
816 - 823 ($330 - $337)	Hydraulic system #n fluid quantity (1 <= n <= 8)	FLOAT SHORT2	l	

5.6 Electric system data

CAN identifier	Electric system parameter name	Suggested data types	Units	Notes
900 - 909 ($384 - $38D)	AC system #n voltage (1 <= n <= 10)	FLOAT SHORT2	volt	
910 - 919 ($38E - $397)	AC system #n current (1 <= n <= 10)	FLOAT SHORT2	ampere	
920 - 929 ($398 - $3A1)	DC system #n voltage (1 <= n <= 10)	FLOAT SHORT2	volt	
930 - 939 ($3A2 - $3AB)	DC system #n current (1 <= n <= 10)	FLOAT SHORT2	ampere	
940 - 949 ($3AC - $3B5)	Prop #n iceguard DC current (1 <= n <= 10)	FLOAT SHORT2	ampere	

5.7 Navigation system data

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1000 ($3E8)	Active nav system waypoint latitude	FLOAT SHORT2	deg	service code field contains waypoint #
1001 ($3E9)	Active nav system waypoint longitude	FLOAT SHORT	deg	service code field contains waypoint #
1002 ($3EA)	Active nav system waypoint height above ellipsoid	FLOAT SHORT2	m	service code field contains waypoint #
1003 ($3EB)	Active nav system waypoint altitude	FLOAT SHORT2	m	service code field contains waypoint #

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1004 ($3EC)	Active nav system ground speed (GS)	FLOAT SHORT2	m/s	service code field contains waypoint #
1005 ($3ED)	Active nav system true track (TT)	FLOAT SHORT2	deg	service code field contains waypoint #
1006 ($3EE)	Active nav system magnetic track (MT)	FLOAT SHORT2	deg	service code field contains waypoint #
1007 ($3EF)	Active nav system cross track error (XTK)	FLOAT SHORT2	m	service code field contains waypoint #
1008 ($3F0)	Active nav system track error angle (TKE)	FLOAT SHORT2	deg	service code field contains waypoint #
1009 ($3F1)	Active nav system time-to-go	SHORT	min	service code field contains waypoint #
1010 ($3F2)	Active nav system estimated time of arrival (ETA)	SHORT	min	service code field contains waypoint #
1011 ($3F3)	Active nav system estimated enroute time (ETE)	SHORT	min	service code field contains waypoint #
1012 ($3F4)	NAV waypoint identifier (char 0-3)	ACHAR4		service code field contains waypoint #
1013 ($3F5)	NAV waypoint identifier (char 4-7)	ACHAR4		service code field contains waypoint #
1014 ($3F6)	NAV waypoint identifier (char 8-11)	ACHAR4		service code field contains waypoint #
1015 ($3F7)	NAV waypoint identifier (char 12-15)	ACHAR4		service code field contains waypoint #

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1016 ($3F8)	NAV waypoint type identifier	LONG SHORT		service code field contains waypoint #
1017 ($3F9)	NAV waypoint latitude	FLOAT SHORT2	deg	service code field contains waypoint #
1018 ($3FA)	NAV waypoint longitude	FLOAT SHORT2	deg	service code field contains waypoint #
1019 ($3FB)	NAV waypoint minimum altitude	FLOAT SHORT2	m	service code field contains waypoint #
1020 ($3FC)	NAV waypoint minimum flight level	FLOAT SHORT2	m	service code field contains waypoint #
1021 ($3FD)	NAV waypoint minimum radar height	FLOAT SHORT2	m	service code field contains waypoint #
1022 ($3FE)	NAV waypoint minimum height above ellipsoid	FLOAT SHORT2	m	service code field contains waypoint #
1023 ($3FF)	NAV waypoint maximum altitude	FLOAT SHORT2	m	service code field contains waypoint #
1024 ($400)	NAV waypoint maximum flight level	FLOAT SHORT2	m	service code field contains waypoint #
1025 ($401)	NAV waypoint maximum radar height	FLOAT SHORT2	m	service code field contains waypoint #
1026 ($402)	NAV waypoint maximum height above ellipsoid	FLOAT SHORT2	m	service code field contains waypoint #
1027 ($403)	NAV waypoint planned altitude	FLOAT SHORT2	m	service code field contains waypoint #

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1028 ($404)	NAV waypoint planned flight level	FLOAT SHORT2	m	service code field contains waypoint #
1029 ($405)	NAV waypoint planned radar height	FLOAT SHORT2	m	service code field contains waypoint #
1030 ($406)	NAV waypoint planned height above ellipsoid	FLOAT SHORT2	m	service code field contains waypoint #
1031 ($407)	Distance to NAV waypoint	FLOAT SHORT2	m	service code field contains waypoint #
1032 ($408)	Time-to-go to NAV waypoint	SHORT	min	service code field contains waypoint #
1033 ($409)	NAV waypoint estimated time of arrival (ETA)	SHORT	min	service code field contains waypoint #
1034 ($40A)	NAV waypoint estimated enroute time (ETE)	SHORT	min	service code field contains waypoint #
1035 ($40B)	NAV waypoint status information	BLONG BSHORT		service code field contains waypoint #
1036 ($40C)	GPS aircraft latitude	DOUBLEL/ DOUBLEH FLOAT SHORT2	deg	
1037 ($40D)	GPS aircraft longitude	DOUBLEL/ DOUBLEH FLOAT SHORT	deg	
1038 ($40E)	GPS aircraft height above ellipsoid	FLOAT SHORT2	m	
1039 ($40F)	GPS ground speed (GS)	FLOAT SHORT2	m/s	
1040 ($410)	GPS true track (TT)	FLOAT SHORT2	deg	

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1041 ($411)	GPS magnetic track (MT)	FLOAT SHORT2	deg	
1042 ($412)	GPS cross track error (XTK)	FLOAT SHORT2	m	
1043 ($413)	GPS track error angle (TKE)	FLOAT SHORT2	deg	
1044 ($414)	GPS glideslope deviation	FLOAT SHORT2	m	
1045 ($415)	GPS predicted RAIM	ULONG USHORT		
1046 ($416)	GPS vertical figure of merit	FLOAT SHORT2	m	
1047 ($417)	GPS horizontal figure of merit	FLOAT SHORT2	m	
1048 ($418)	GPS mode of operation	SHORT		
1049 ($419)	INS aircraft latitude	FLOAT SHORT2	deg	
1050 ($41A)	INS aircraft longitude	FLOAT SHORT	deg	
1051 ($41B)	INS aircraft height above ellipsoid	FLOAT SHORT2	m	
1052 ($41C)	INS aircraft ground speed (GS)	FLOAT SHORT2	m/s	
1053 ($41D)	INS aircraft true track (TT)	FLOAT SHORT2	deg	
1054 ($41E)	INS aircraft magnetic track (MT)	FLOAT SHORT2	deg	
1055 ($41F)	INS aircraft cross track error (XTK)	FLOAT SHORT2	m	
1056 ($420)	INS aircraft track error angle (TKE)	FLOAT SHORT2	deg	
1057 ($421)	INS vertical figure of merit	FLOAT SHORT2	m	

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1058 ($422)	INS horizontal figure of merit	FLOAT SHORT2	m	
1059 ($423)	Auxiliary nav system aircraft latitude	FLOAT SHORT2	deg	
1060 ($424)	Auxiliary nav system aircraft longitude	FLOAT SHORT	deg	
1061 ($425)	Auxiliary nav system aircraft height above ellipsoid	FLOAT SHORT2	m	
1062 ($426)	Auxiliary nav system aircraft ground speed (GS)	FLOAT SHORT2	m/s	
1063 ($427)	Auxiliary nav system aircraft true track (TT)	FLOAT SHORT2	deg	
1064 ($428)	Auxiliary nav system aircraft magnetic track (MT)	FLOAT SHORT2	deg	
1065 ($429)	Auxiliary nav system aircraft cross track error (XTK)	FLOAT SHORT2	m	
1066 ($42A)	Auxiliary nav system aircraft track error angle (TKE)	FLOAT SHORT2	deg	
1067 ($42B)	Auxiliary nav system vertical figu-re of merit	FLOAT SHORT2	m	
1068 ($42C)	Auxiliary nav system horizontal figure of merit	FLOAT SHORT2	m	
1069 ($42D)	Magnetic heading (MH)	FLOAT SHORT2	deg	
1070 ($42E)	Radio Height	FLOAT SHORT2	m	

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1071 - 1074 ($42F - $432)	DME #n distance (1 < n <= 4)	FLOAT SHORT2	m	
1075 - 1078 ($433 - $436)	DME #n time-to-go (1 < n <= 4)	SHORT	min	
1079 - 1082 ($437 - $43A)	DME #n ground speed (1 < n <= 4)	FLOAT SHORT2	m/s	
1083 - 1086 ($43B - $43E)	ADF #n bearing (1 < n <= 4)	FLOAT SHORT2	deg	
1087 - 1090 ($43F - $442)	ILS #n localize deviation (1 < n <= 4)	FLOAT SHORT2	deg	
1091 - 1094 ($443 - $446)	ILS #n glideslope deviation (1 < n <= 4)	FLOAT SHORT2	deg	
1095 - 1096 ($446 - $447)	Flight director #n pitch deviation (1 < n <= 2)	FLOAT SHORT2	deg	
1097 - 1098 ($448 - $449)	Flight director #n roll deviation (1 < n <= 2)	FLOAT SHORT2	deg	
1099 ($44A)	Decision height	FLOAT SHORT2	m	
1100 - 1103 ($44B - $44F)	VHF #n COM frequency (0 < n < 4)	FLOAT ACHAR4	MHz	

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1104 - 1107 ($450 - $453)	VOR/ILS #n frequency (1 < n <= 4)	FLOAT ACHAR4	MHz	
1108 - 1111 ($454 - $457)	ADF #n frequency (1 < n <= 4)	FLOAT ACHAR4	KHz	
1112 - 1115 ($458 - $45B)	DME #n channel (1 < n <= 4)	FLOAT ACHAR4		
1116 - 1119 ($45C - $45F)	Transponder #n code (1 < n <= 4)	FLOAT ACHAR4		
1120 ($460)	Desired track angle	FLOAT SHORT2	deg	
1121 ($461)	Magnetic variation	FLOAT SHORT2	deg	
1122 ($462)	Selected glidepath angle	FLOAT SHORT2	deg	
1123 ($463)	Selected runway heading	FLOAT SHORT2	deg	
1124 ($464)	Computed vertical velocity	FLOAT SHORT2	m/s	
1125 ($465)	Selected course	FLOAT SHORT2	deg	
1126 - 1129 ($466 - $469)	VOR #n radial (1 < n <= 4)	FLOAT SHORT2	deg	
1130 ($46A)	True east velocity	FLOAT SHORT2	m/s	
1131 ($46B)	True north velocity	FLOAT SHORT2	m/s	

CAN identifier	Navigation system parameter name	Suggested data types	Units	Notes
1132 ($46C)	True up velocity	FLOAT SHORT2	m/s	
1133 ($46D)	true heading	FLOAT SHORT2	deg	+/- 180^{o}

5.8 Landing gear system data

CAN identifier	Landing gear system parameter name	Suggested data types	Units	Notes
1175 ($497)	Gear lever switches	BLONG BSHORT		
1176 ($498)	Gear lever lights/ WOW solenoid	BLONG BSHORT		
1177 - 1180 ($499 - $49C)	Landing gear #n tire pressure (1 < n <= 4)	FLOAT SHORT2	hPa	service code field contains tire number
1181 - 1184 ($49D - $4A0)	Landing gear #n brake pad thickness (1 < n <= 4)	FLOAT SHORT2	mm	service code field contains brake pad number

5.9 Miscellaneous data

CAN identifier	Miscellaneous parameter name	Suggested data types	Units	Notes
1200 ($4B0)	UTC	CHAR4		format: 13h43min22s 13 43 22 00
1201 ($4B1)	Cabin pressure	FLOAT SHORT2	hPa	
1202 ($4B2)	Cabin altitude	FLOAT SHORT2	m	
1203 ($4B3)	Cabin temperature	FLOAT SHORT2	K	

CAN identifier	Miscellaneous parameter name	Suggested data types	Units	Notes
1204 ($4B4)	Longitudinal center of gravity	FLOAT SHORT2	% MAC	
1205 ($4B5)	Lateral center of gravity	FLOAT SHORT2	% MAC	
1206 ($4B6)	Date	CHAR4		format: 12. June 1987 12 06 19 87

5.10 Reserved data

CAN identifier	Parameter name	Suggested data types	Units	Notes
1300 - 1499 ($514 - $577)	Reserved for future use			

6 Time-Triggered Bus Scheduling

This section describes a sample avionics system for a general aviation aircraft and the resulting CANaerospace bus implementation.

The purpose of this example is to serve as a guideline for the evaluation of system requirements, CANaerospace bus load and transmission rates based on the basic systems installed in modern technology general aviation aircraft.

6.1 Baseline system

The baseline system reflects the avionic system as installed in a IFR-equipped single engine general aviation aircraft using flat panel primary flight and navigation displays. This architecture was chosen as it can potentially support “highway in the sky” (HITS) technology. Even though it was kept rather simple, however, to better serve its purpose as explanatory example:

CANaerospace Node-ID	System Description
1	Attitude/heading reference system (AHRS)
2	Air data computer (ADC)
3	VHF communication transceiver #1
4	VHF communication transceiver #2
5	NAV/ILS/Marker receiver #1
6	NAV/ILS/Marker receiver #2
7	ATC transponder
8	ADF receiver
9	GPS receiver
10	Distance measuring equipment (DME)
11	Engine monitoring system (EMS)
12	Electrical trim system
13	Electric system

To determine the CANaerospace bus schedule, it will be assumed that the maximum transfer rate of parameters is 80Hz (12.5ms). We could transmit data at a much higher rate but this would make no sense unless there is equipment installed which can make use of this.

6.2 The transmission slot concept

The concept of the time-triggered bus scheduling uses a “minor time frame” (12.5ms in our case) and takes advantage of the fact that not all messages in a given system have to be transmitted at this interval. Specifying multiples of the minor time frame transmission interval and associated “transmission slots” allow a substantially larger number of parameters to be transmitted on a single bus:

Transmission interval	Parameters/ Transmission Slot	Number of Transmission Slots (equalling 100% bus load)	Transmission Slot Identification
12.5ms (80Hz)	1	100	A0 - A99
25ms (40Hz)	2	200	B0[0] - B99[1]
50ms (20Hz)	4	400	C0[0] - C99[3]
100ms (10Hz)	8	800	D0[0] - D99[7]
200ms (50Hz)	16	1600	E0[0] - E99[15]
400ms (2.5Hz)	32	3200	F0[0] - F99[31]
1000ms (1.0Hz)	80	8000	G0[0] - G99[79]

With this transmission slot concept, either 100 parameters transmitted each 12.5ms or 8000 parameters transmitted once a second would generate 100% bus load. More likely, however, a combination of parameters in the various transmission slot groups from this table (A-G) will be used. For our baseline system, we identified the following data and assigned them the transmission slot groups A,D and G:

Tx Slot	Parameter Name	Unit	Trans-mission Interval	CAN-ID	Data Type
A0	Body longitudinal acceleration	g	12.5ms	300 ($12C)	FLOAT
A1	Body lateral acceleration	g	12.5ms	301 ($12D)	FLOAT

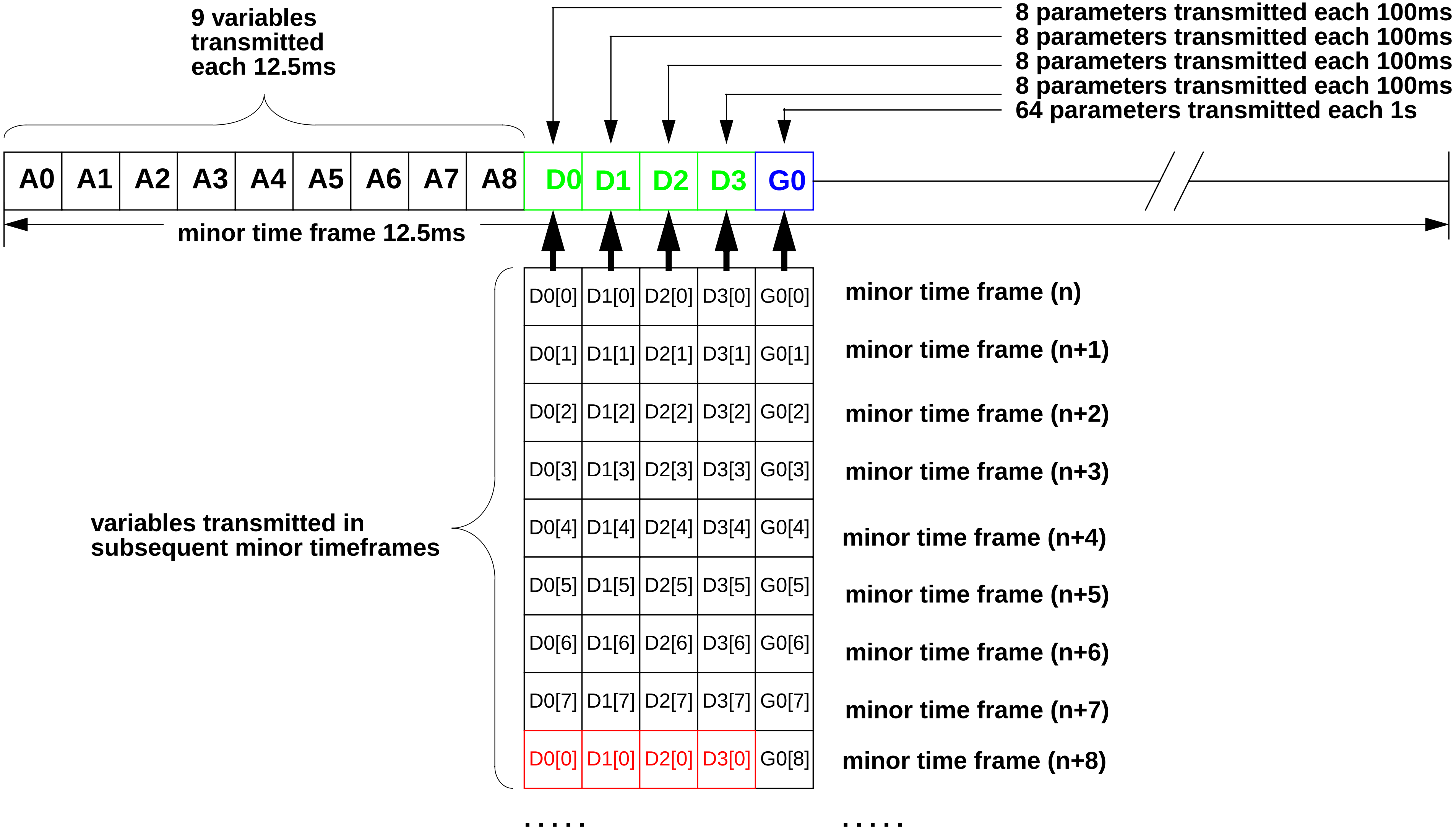
Tx Slot	Parameter Name	Unit	Trans-mission Interval	CAN-ID	Data Type
A2	Body normal acceleration	g	12.5ms	302 ($12E)	FLOAT
A3	Body pitch rate	deg/s	12.5ms	303 ($12F)	FLOAT
A4	Body roll rate	deg/s	12.5ms	304 ($130)	FLOAT
A5	Body yaw rate	deg/s	12.5ms	305 ($131)	FLOAT
A6	Body pitch angle	deg	12.5ms	311 ($137)	FLOAT
A7	Body roll angle	deg	12.5ms	312 ($138)	FLOAT
A8	Heading angle	deg	12.5ms	321 ($141)	FLOAT
D0[0]	Altitude rate	m/s	100ms	314 ($13A)	FLOAT
D0[1]	True airspeed	m/s	100ms	316 ($13C)	FLOAT
D0[2]	Computed (calibrated) airspeed	m/s	100ms	317 ($13D)	FLOAT
D0[3]	Baro correction	hPa	100ms	319 ($13F)	FLOAT
D0[4]	Baro corrected altitude	m	100ms	320 ($140)	FLOAT
D0[5]	Standard altitude	m	100ms	322 ($142)	FLOAT
D0[6]	Lateral stick trim position command	%	100ms	402 ($192)	FLOAT
D0[7]	Longitudinal stick trim position command	%	100ms	405 ($195)	FLOAT
D1[0]	Engine RPM	1/min	100ms	500 ($1F4)	FLOAT
D1[1]	Propeller RPM	1/min	100ms	504 ($1F8)	FLOAT
D1[2]	Engine exhaust gas temperature (EGT)	K	100ms	520 ($208)	FLOAT
D1[3]	Engine fuel flow rate	l/h	100ms	524 ($20C)	FLOAT

Tx Slot	Parameter Name	Unit	Trans-mission Interval	CAN-ID	Data Type
D1[4]	Engine manifold pressure	hPa	100ms	528 ($210)	FLOAT
D1[5]	Engine oil pressure	hPa	100ms	532 ($214)	FLOAT
D1[6]	Engine oil temperature	K	100ms	536 ($218)	FLOAT
D1[7]	Engine cylinder head temperature (CHT)	K	100ms	540 ($21C)	FLOAT
D2[0]	Fuel tank #1 quantity	kg	100ms	668 ($29C)	FLOAT
D2[1]	Fuel tank #2 quantity	kg	100ms	669 ($29D)	FLOAT
D2[2]	Fuel system pressure	hPa	100ms	684 ($2AC)	FLOAT
D2[3]	DC voltage	V	100ms	920 ($398)	FLOAT
D2[4]	DC current	A	100ms	930 ($3A2)	FLOAT
D2[5]	GPS height above ellipsoid	m	100ms	1030 ($40E)	FLOAT
D2[6]	GPS aircraft latitude	deg	100ms	1036 ($40c)	FLOAT
D2[7]	GPS aircraft longitude	deg	100ms	1037 ($40D)	FLOAT
D3[0]	GPS ground speed	m/s	100ms	1039 ($40F)	FLOAT
D3[1]	GPS true track	deg	100ms	1040 ($410)	FLOAT
D3[2]	DME distance	m	100ms	1071 ($42F)	FLOAT
D3[3]	DME time-to-station	min	100ms	1075 ($433)	FLOAT
D3[4]	DME ground speed	m/s	100ms	1079 ($437)	FLOAT
D3[5]	ILS #1 localizer deviation	deg	100ms	1087 ($43F)	FLOAT
D3[6]	ILS #2 localizer deviation	deg	100ms	1088 ($440)	FLOAT

Tx Slot	Parameter Name	Unit	Trans-mission Interval	CAN-ID	Data Type
D3[7]	ILS #1 glideslope deviation	deg	100ms	1091 ($443)	FLOAT
D4[0]	ILS #2 glideslope deviation	deg	100ms	1092 ($444)	FLOAT
D4[1]	VOR #1 radial	deg	100ms	1126 ($466)	FLOAT
D4[2]	VOR #2 radial	deg	100ms	1127 ($467)	FLOAT
D4[3]	ADF #1 relative bearing	deg	100ms	1083 ($43B)	FLOAT
G0[0]	Static air temperature	K	1s	324 ($144)	FLOAT
G0[1]	Trim system switches		1s event	439 ($1B7)	BSHORT
G0[2]	Trim system lights		1s event	440 ($1B8)	BSHORT
G0[3]	Engine status		1s event	556 ($22C)	BSHORT
G0[4]	VHF COM #1 frequency	MHz	1s event	1100 ($44B)	FLOAT
G0[5]	VHF COM #2 frequency	MHz	1s event	1101 ($44C)	FLOAT
G0[6]	Transponder #1 code	BCD	1s event	1116 ($45C)	FLOAT
G0[7]	ADF #1 frequency	kHz	1s event	1108 ($454)	FLOAT
G0[8]	VOR/ILS #1 frequency	MHz	1s event	1104 ($450)	FLOAT
G0[9]	VOR/ILS #2 frequency	MHz	1s event	1105 ($451)	FLOAT

A transmission interval of “1s/event” means that the respective parameter is transmitted once upon every state change and additionally once a second if the state is unchanged.

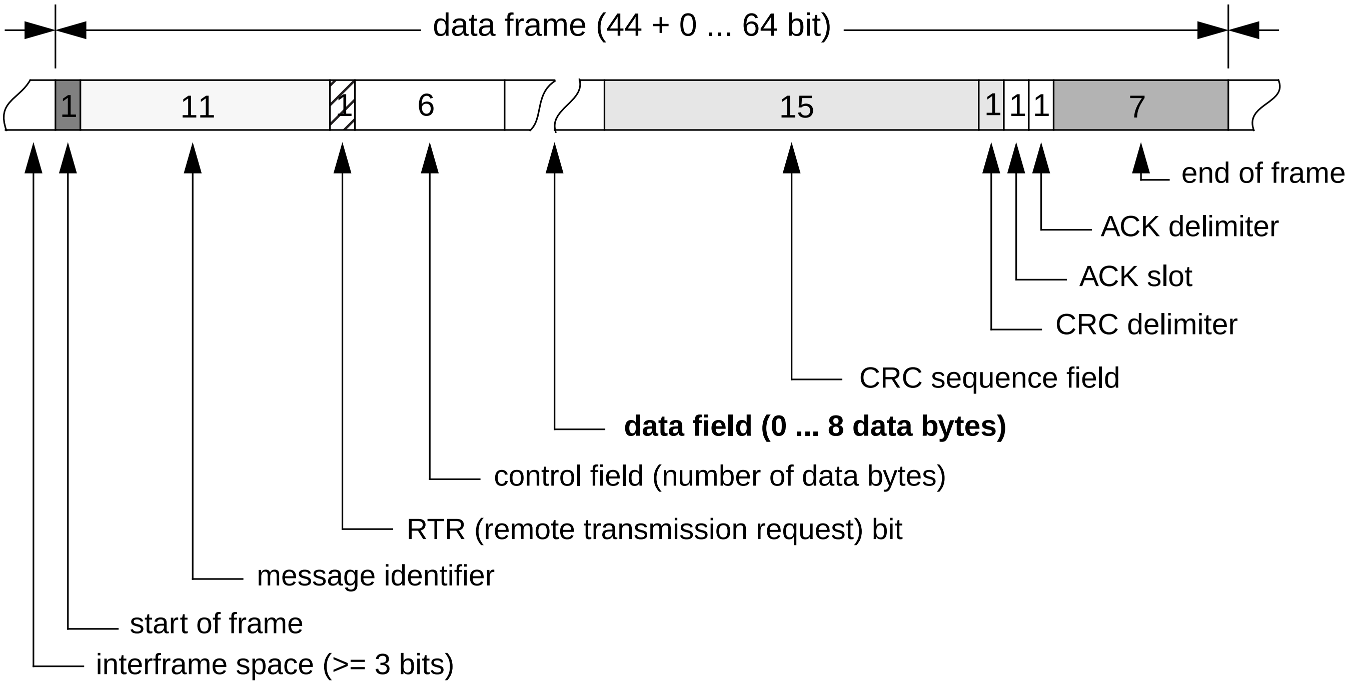
Analyzing the “Tx Slot” fields, we find out that our baseline system requires 9 parameters to be transmitted each 12.5ms, 32 parameters to be transmitted each 100ms and 10 parameters to be transmitted once a second. This results in the following transmission slot allocation:


9 variables transmitted each 12.5ms
8 parameters transmitted each 100ms
8 parameters transmitted each 100ms
8 parameters transmitted each 100ms
8 parameters transmitted each 100ms
64 parameters transmitted each 1s
A0
A1
A2
A3
A4
A5
A6
A7
A8
D0
D1
D2
D3
G0
minor time frame 12.5ms
variables transmitted in subsequent minor timeframes
.....
.....

D0[0]	D1[0]	D2[0]	D3[0]	G0[0]	minor time frame (n)
D0[1]	D1[1]	D2[1]	D3[1]	G0[1]	minor time frame (n+1)
D0[2]	D1[2]	D2[2]	D3[2]	G0[2]	minor time frame (n+2)
D0[3]	D1[3]	D2[3]	D3[3]	G0[3]	minor time frame (n+3)
D0[4]	D1[4]	D2[4]	D3[4]	G0[4]	minor time frame (n+4)
D0[5]	D1[5]	D2[5]	D3[5]	G0[5]	minor time frame (n+5)
D0[6]	D1[6]	D2[6]	D3[6]	G0[6]	minor time frame (n+6)
D0[7]	D1[7]	D2[7]	D3[7]	G0[7]	minor time frame (n+7)
D0[0]	D1[0]	D2[0]	D3[0]	G0[8]	minor time frame (n+8)

The CAN bus data frame (11-bit identifier) has the following outline:

6.3 Bus load computation


data frame (44 + 0 ... 64 bit)
1
11
1
6
15
1
1
1
7
end of frame
ACK delimiter
ACK slot
CRC delimiter
CRC sequence field
data field (0 ... 8 data bytes)
control field (number of data bytes)
RTR (remote transmission request) bit
message identifier
start of frame
interframe space (>= 3 bits)

Most CANaerospace messages use all 8 bytes of the data field which results in a message length of 44bits + 64bits = 108bits. To compute the maximum bus capacity, we have to add the interframe space (3bits) and a number of stuff bits (a maximum of 18 additional bits, we assume an average of 14) which gives a message length of 108bits + 3bits + 14bits = 125bits. Assuming the maximum data transfer rate of 1Mbit/s, a CANaerospace message takes 125µs to transmit. Hence, the CANaerospace bus capacity is 8.000 messages/second.

Defining a minor time frame of 12.5ms (80Hz) results in 100 parameters which can be transmitted during this interval. This number can be considered 100% bus load:

CANaerospace message time:	125µs
Selected minor time frame:	12.5ms
100% bus load:	12.5ms/125µs = 100 messages

For 29-bit identifier CAN messages (CAN 2.0B), the message time is 145µs, which results in 16% less bus capacity than for 11-bit identifier CAN messages. If both 11-bit and 29-bit messages are used at the same time, the calculation should be done for each identifier type seperately and combined afterwards to assemble the resulting bus schedule data.

Our baseline system uses the following parameter/transmission interval matrix:

Transmission interval	Parameters	Required Transmission Slots	Parameters/ Transmission Slot
12.5ms	9	9	1
100ms	36	5 (4.5)	8 (0.1s/12.5*10^{-3}s)
1s	10	1 (0.125)	80 (1s/12.5*10^{-3}s)
total	**55**	**15 (13.625)**	

As the baseline system uses 13.625 out of 100 available transmission slots, the corresponding CANaerospace bus load is:

13.625%

Keeping in mind that asynchronous event data or node service data might require additional bus capacity, we should leave some margin for this (around 20%). Therefore, our system should not continously exceed 80% bus load. Even though, this analysis shows that CANaerospace offers adequate growth capability for general aviation aircraft avionics.

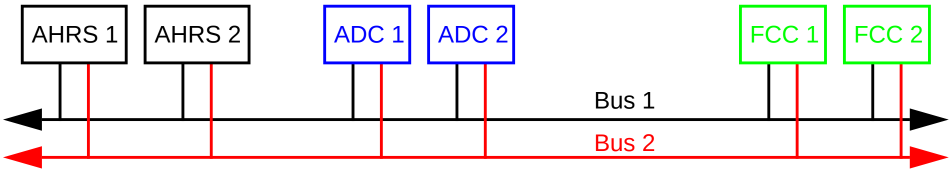
7 System redundancy support

The probability of an undetected data corruption in a CAN network is around 1 * 10^{-13} per message transmission. Assuming 100% bus load (around 8.000 messages per second), this will result in a probability of 2.9 * 10^{-6} undetected failures per flight hour, making CANaerospace a candidate for mission and flight critical systems.

While this figure is better than for any other bus system available today, it shows that a *single* CANaerospace bus (like all other buses) will most likely not be adequate for flight critical systems, especially for those requiring fail-operational behaviour. For those applications, system redundancy is inevitable to demonstrate a required level of functional safety.

7.1 Redundant message identifier assignment

A system architecture as used by many modern integrated avionics and electronic flight control systems is shown below. In this architecture, two redundant units of the same type communicate via an equal number of communication channels. Proper design provided, this system will prevent a single failure to cause a complete loss of function:



Using the standard identifier distribution, each CANaerospace data bus parameter has assigned a single, unique identifier (i.e. 304 for body roll rate). This means that only one unit would be allowed to transmit a particular parameter on the bus. A redundant system architecture as described is supported by CANaerospace, however, if the 29-bit identifiers are used. In this case, a “redundancy level offset” in multiples of 65536 ($10000) is added to each identifier so that the same parameter can be transmitted by several units using multiple unique identifiers:

Redundancy Channel #	Redundancy Level Offset	Example: Body Roll Rate ID (decimal)	Example: Body Roll Rate ID (hexadecimal)
0	0	304	$130
1	65536	65840	$10130
2	131072	131376	$20130
3	196608	196912	$30130
n	65536 * n	65536 * n + 304	$[10*n]130

7.2 System redundancy and the CANaerospace header

Unlike other buses like ARINC429, ARINC629 or MIL-STD-1553B, CANaerospace is a dynamic network with a bus schedule that varies within certain limits. Certification in flight safety critical applications, however, requires to demonstrate the proper function of the data transmission under all conditions. Monitoring CANaerospace messages during normal operation and processing the header information delivers the required information for certification. Additionally, the header information improves flexibility and supports dynamic network reconfiguration. Power down/up situations are handled gracefully, units may be added to the network without software changes. Taking advantage of the header information, CANaerospace bus analyzers and simulators can be inserted even into a running network and will immediately have all information about network structure, units and data. This ensures fast and cost-effective maintenance:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Message Header

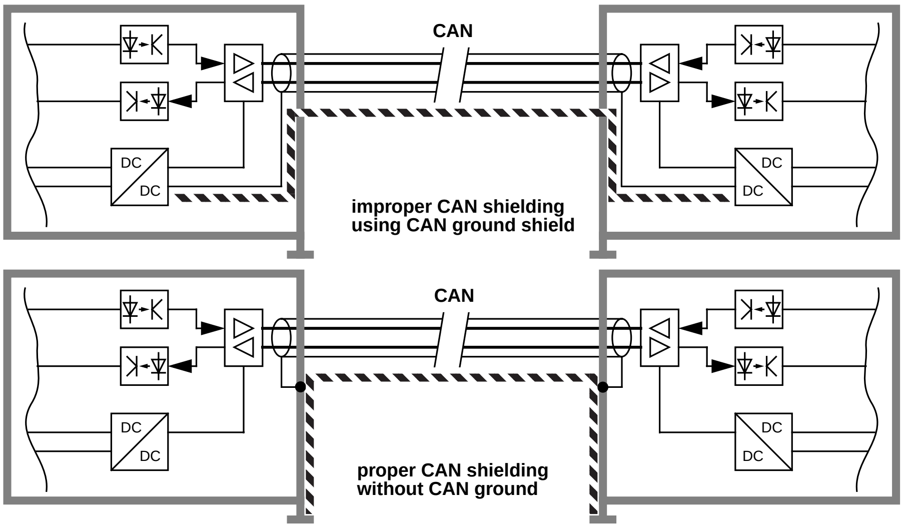
- Node-ID (Byte 0): Some system architectures employ backup units which become active if the main unit fails. The Node-ID allows to immediately identify this situation and react accordingly (i.e. mode change within redundancy management).
- Data Type (Byte 1): CANaerospace supports multiple data ty-

pes for every message. Backup units (or units from different vendors) may use different data types while performing identical functions. Specifying the data type with each message allows automatic system configuration, even during runtime.

- Service Code (Byte 2): For Normal Operation Data, this byte should continously reflect the status of the data (or the transmitting unit) to support data integrity monitoring within receiving units. With this information, the validity of data is known at any given time.
- Message Code (Byte 3): Message numbering allows to detect if messages are missing and if the transmitting unit is operating properly. Also, it can be used to compare the "age" of messages from redundant sources.

8 Physical connector definition

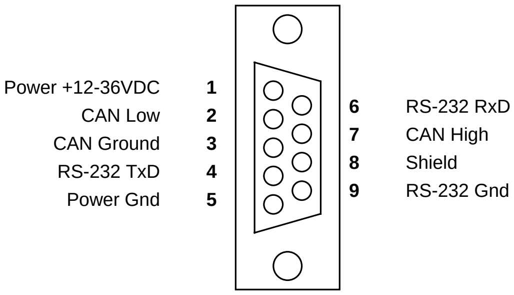
For CANaerospace, a physical connection suitable for airborne connector types has been defined (connectors according to CiA DS102 are also supported). Note that unlike most other definitions for CAN connections, CANaerospace connectors allows to supply +12-36VDC power to the units via the CAN connector (+12-36VDC, Power Ground). The RS-232 connection present on some of the connector types is optional and may be used for maintenance or debug interfaces. Note also that the use of CAN Ground is supported by the connector pinout but strongly discouraged due to potential EMC problems in airborne applications as shown below:



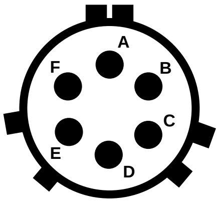
Strongly encouraged is the use of optically isolated CAN interfaces for all units in the network. For the wiring, AWG 22 aerospace standard shielded twisted pair (STP) or shielded twisted quadruple (STQ) should be used.

The pinout of the CANaerospace connectors is as follows:

MIL-24308/8 connector (similar to CiA DS102)

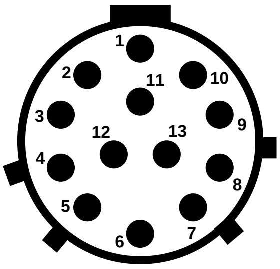


MIL-C-26482 connectors MS3470L1006PN (wall mount receptacle) and MS3476L1006SN (mating straight plug)



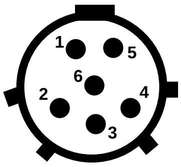
Pin A	Power Gnd
Pin B	+12-36VDC
Pin C	Shield
Pin D	CAN High
Pin E	CAN Low
Pin F	CAN Gnd

MIL-C-38999 connectors D38999/20FB35PN (wall mount receptacle) and D38999/26FB35SN (mating straight plug)



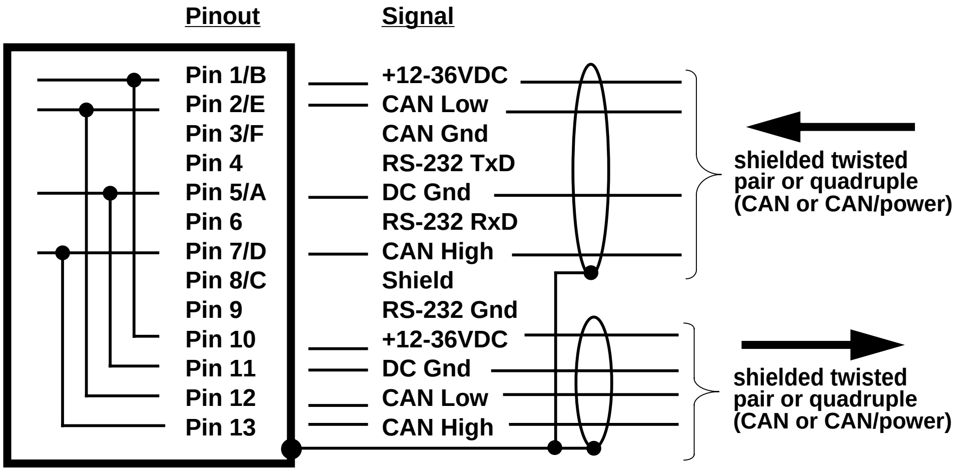
Pin 1	+12-36VDC	Pin 6	RS-232 RxD
Pin 2	CAN Low	Pin 7	CAN High
Pin 3	CAN Gnd	Pin 8	Shield
Pin 4	RS-232 TxD	Pin 9	RS-232 Gnd
Pin 5	DC Gnd	Pin 10-13	unused

MIL-C-38999 connector D38999/20FA35PN (wall mount receptacle) and D38999/26FA35SN (mating straight plug)



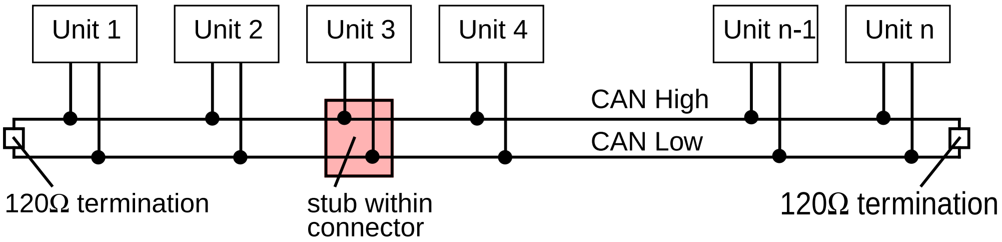
Pin 1	+12-36VDC	Pin 4	CAN High
Pin 2	CAN Low	Pin 5	DC Gnd
Pin 3	CAN Gnd	Pin 6	Shield

A sample interconnection of multiple CANaerospace systems using D38999/20FB35PN wall mount receptacles and D38999/26FB35SN straight plugs is shown here:



Note: The RS-232 lines defined for this connector type are optional and may be used for device programming, configuration, etc. They have no relationship to CAN or CANaerospace.

CANaerospace systems requiring high reliability or life-insertion capabilities should be connected as shown below. The preferred CAN bus topology is a shielded, twisted pair single line, terminated at both ends. Units are connected via simple stubs within the connector. Using this method, removing a unit from the bus (or reattaching it) will not adversely affect the others: The bus is not opened by unplugging the connectors:


Unit 1
Unit 2
Unit 3
Unit 4
Unit n-1
Unit n
CAN High
CAN Low
120Ω termination
stub within connector
120Ω termination