Bachelor Thesis

Philipp Iohannes Gmelin

Signal Transmission between Electrical Components in Microlight Aircraft

Fakultät Technik und Informatik

Department Fahrzeugtechnik und Flugzeugbau

Faculty of Engineering and Computer Science

Department of Automotive and Aeronautical Engineering

Philipp Iohannes Gmelin

Signal Transmission between Electrical Components in Microlight Aircraft

Bachelor thesis submitted as part of the bachelor examination

Degree program: Aeronautical Engineering
Department of Automotive and Aeronautical Engineering
Faculty of Engineering and Computer Science
Hamburg University of Applied Sciences

First examiner: Prof. Dr.-Ing. Dieter Scholz, MSME
Second examiner: Dipl.-Ing. Jan Kaminski

Submitted: 2022-02-24

DOI:
https://doi.org/10.5281/zenodo.15618552

URN:
https://nbn-resolving.org/urn:nbn:de:gbv:18302-aero2022-02-24.011
Associated URLs:
https://nbn-resolving.org/html/urn:nbn:de:gbv:18302-aero2022-02-24.011

Any further request may be directed to:
Prof. Dr.-Ing. Dieter Scholz, MSME
E-Mail see: http://www.ProfScholz.de

This work is part of:
Digital Library - Projects & Theses - Prof. Dr. Scholz
http://library.ProfScholz.de

Published by
Aircraft Design and Systems Group (AERO)
Department of Automotive and Aeronautical Engineering
Hamburg University of Applied Science

This report is deposited and archived:

- Deutsche Nationalbiliothek (https://www.dnb.de)
- ZENODO (https://zenodo.org/communities/digitallibrary)
- Internet Archive (https://archive.org)
  Item: https://archive.org/details/TextGmelinBachelor.pdf

Abstract

Purpose – This paper explains the basic characteristics of signal transmission in communication systems. Transmission channels such as copper and aluminum conductors, radio transmission and optical transmission are explained. Major aviation protocols are presented in their functioning. The signal processing and transmission are shown on the basis of the fuel level indicator. Decision guidance for the selection of sustainable avionics is provided.
Methodology – Transmission channels are compared in terms of bandwidth, cost and interference factors. Signal transmission in the application case of the fuel level indicator is carried out by measuring the sensor resistance and the further signal processing with a microcontroller.
Findings – For the individual application, a suitable selection of microlight communication systems is presented. Currently, the RS-232 standard (point-to-point connection) is still used to connect electrical devices from different manufacturers. For larger series production of microlight aircraft, a BUS set up with CANaerospace is an attractive option. Cloud networked systems are expected to be the future of avionics systems. Wireless communication protocols, such as ZigBee are a sustainable solution due to their modular adaptability to different requirements.
Practical Implications – Knowledge of signal transmission can help engineers design better systems for microlight aircraft, affecting safety, cost and weight.
Originality – After research, communication systems and avionics systems for microlight aircraft have not been covered in any scientific work. The listing of aviation protocols to this extent with their functionality has also not been published in open access. Own concepts for future avionics systems, such as a digital compass pointing to true north, are developed (Mag2True). Furthermore, the software for the fuel level indicator was developed from scratch as part of the thesis.

Keywords Ultralight aircraft, Airplanes, Avionics, Electric equipment, Signal processing, Wireless, Communication systems, Computer network protocols, Electric wiring, Arduino (Programmable controller), Microlight

Kurzreferat

Zweck – In dieser Arbeit werden die grundlegenden Eigenschaften von Signalübertragungen der Kommunikationssysteme erklärt. Übertragungsmedien wie Kupfer- und Aluminiumleiter, Funkübertragung und optische Übertragung werden erläutert. Wichtige Luftfahrt-Protokolle werden in ihrer Funktionsweise vorgestellt. Die konkrete Signalverarbeitung wird anhand der Tankanzeige für das Ultraleichtflugzeug Flywhale gezeigt. Eine Entscheidungshilfe für die Auswahl von zukunftsfähiger Avionik bei Ultraleichtflugzeug wird entwickelt.
Methodik – Die Übertragungsmedien werden unter den Aspekten der Bandbreite, Kosten und Störfaktoren verglichen. Die Signalübertragung der Kraftstoffmengenanzeige erfolgt über eine Messung des Sensorwiderstandes und weiterer Signalverarbeitung mit einem Mikrocontroller.
Ergebnisse – Der RS-232-Standard (Punkt-zu-Punkt-Verbindung) wird derzeit verwendet, um elektrische Geräte verschiedener Hersteller zu verbinden. Für eine größere Serienproduktion von Ultraleichtflugzeugen ist ein BUS-Aufbau mit CANaerospace eine attraktive Option. Für die Zukunft der Avioniksysteme ist ein Cloud-vernetztes System zu erwarten. Drahtlose Kommunikationsprotokolle wie ZigBee sind aufgrund ihrer modularen Anpassbarkeit an verschiedene Anforderungen eine zukunftsfähige Lösung.
Bedeutung in der Praxis – Das Wissen über die Signalübertragung hilft Ingenieuren bessere Systeme für Flugzeuge zu entwerfen, was sich auf Sicherheit, Kosten und Gewicht auswirkt.
Originalität – Kommunikations- und Avioniksysteme für Ultraleichtflugzeuge wurden bisher in keiner wissenschaftlichen Arbeit behandelt. Auch die Auflistung der Luftfahrt-Protokolle in ihrer Funktionsweise wurde in dem Umfang noch nicht unter freiem Zugang veröffentlicht. Eigene Konzepte für zukünftige Avioniksysteme, wie ein digitaler Kompass, der auf den wahren Norden zeigt, werden entworfen (Mag2True). Darüber hinaus wurde im Rahmen der Arbeit die Software für die Tankanzeige von Grund auf entwickelt.

Stichworte Flugzeug, Avionik, Nachrichtenübertragungstechnik, Computerunterstütze Kommunikation, Luftfahrt, Protokoll, Kabelverlegung, Arduino-Plattform, Ultraleichtflugzeug

DEPARTMENT OF AUTOMOTIVE AND AERONAUTICAL ENGINEERING

Signal Transmission between Electrical Components in Microlight Aircraft

Task for a *Bachelor Thesis*

Background

New microlight aircraft are becoming more complex with more technology and electronics onboard to make flying safer and easier while improving the user experience. One reason for this is the increase of the allowed maximum take-off weight in several countries since 2019. This enabled microlight aircraft manufacturers around the world to build more complex aircraft with larger weight margins. Uniplanes GmbH was therefore able to apply for a new certification for the amphibious Flywhale aircraft with a maximum take-off weight of 650 kg, which offers more comfort and range. Customers of personal aircraft are beginning to expect user interfaces they know from consumer electronics and cars. The electrical equipment exceeds the legal requirements. For example, the instruments of modern microlight aircraft would enable IFR, which is not permitted for this class. For all electrical components on board an aircraft to work together, they must be able to communicate. This communication requires a transmission channel between the devices and the ability to interpret their signals.

Task

The task of this thesis is to show the steps of signal transmission between electrical components in microlight aircraft. The distinction to larger aircraft classes shall be shown. Furthermore, the avionics of the Flywhale aircraft must be presented as an example for microlight aircraft. The structure of this work is carried out in the following steps:

- Explain the fundamentals of communication between electrical components.
- Discuss different transmission channels and their ability to transmit data.
- Examine the suitability of different protocols for microlight aircraft.
- Introduce the Flywhale aircraft from Uniplanes GmbH, focussing on its avionics.
- Develop a fuel level indicator system that incorporates signal processing and transmission. The component should be suitable for usage in a microlight aircraft.

The report has to be written in English based on German or international standards on report writing.

Table of Contents

List of Figures

List of Tables

List of Symbols

A	Area
Ag	Silver
Al	Aluminium
Au	Gold
C	Capacitor
C	Coal
c	Speed of light
Cu	Copper
E_0	Electric field
F	Frequency
f	Frequency
Fe	Iron
H_0	Magnetic field
L_P	Reflection coefficient, (Power ratio)
L_U	Reflection coefficient (Voltage ratio)
m	Mass
P	Power
Pb	Lead
R	Resistance
r	Radius
U	Voltage
W	Tungsten
X_C	Capacitive reactance
Z_0	Free space wave impedance

Greek Symbols

λ	Wavelenngth
µ	Micro
Ω	Ohm

List of Abbreviations

AAM	Advanced Air Mobility
ACK	Acknowledges
ADAHRS	Air Data and Heading Reference System
ADF	Automatic Direction Finder
ADN	Aircraft Data Network
ADS-B	Automatic Dependent Surveillance-Broadcast
AEEC	Airlines Electronic Engineering Committee
AFDX	Avionics Full-Duplex Switched Ethernet
AHRS	Attitude and Heading Reference Systems
AHRTAG	Aviation Heading Reference Transition Action Group
AI	Artificial Intelligence
ALT	Altimeter
AOA	Ange Of Attack
APP	Application
APU	Auxiliary Power Unit
ARINC	Aeronautical Radio Incorporated
ARNS	Aeronautical Radio Navigation Service
ARP	Address Resolution Protocol
AS	Aerospace Standard
ASCII	American Standard Code for Information Interchange
ASN	Abstract Syntax Notation
ATA	Air Transport Association of America
ATC	Air Traffic Control
ATM	Anyone-To-Many
BAG	Bandwidth Allocation Gap
BC	BUS Controller
BM	BUS Monitor
BRS	Ballistic Recovery System
BUS	Binary Unit System
CAD	Computer-Aided Design
CAN	Controller Area Network
CANH	CAN High
CANL	CAN Low
CD+AMP	Collision Detection and Arbitration on Message Priority
CFRP	Carbon Fiber Reinforced Polymers
CHAR	Character
CiA	CAN in Automation
COM	Communication
CRC	Cyclic Redundancy Check

CSMA	Carrier-Sense, Multiple Access
CW	Continuous Wave
CWDM	Coarse Wavelength-Division Multiplexer
DAC	Digital Analog Converter
DAeC	Deutscher Aero Club e.V.
DC	Direct Current
DEMUX	De-Multiplexer
DITS	Digital Information Transfer System
DLC	Data Length Code
DME	Distance Measuring Equipment
DNS	Domain Name System
DSN	Deep Space Network
EASA	European Aviation Safety Agency
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECU	Engine Control Unit
EFIS	Electronic Flight Instrument System
EHF	Extra High Frequency
EIA/TIA	Electronic Industry Association and the Telecommunication Industry Association
EMP	ElectroMagnetic Pulse
EMU	Engine Management Unit
EOF	End-Of-Frame
ESC	Electronic Speed Control
FCAS	Future Combat Air System
FMS	Flight Management System
FPV	First-Person View
FW	Flywhale
GmbH	Gesellschaft mit beschränkter Haftung | company with limited liability
GND	Gemeinsame Normdatei | common standards file
GND	Ground (negative)
GPS	Global Positioning System
HF	High Frequency
HTTP	Hypertext Transfer Protocol
ICAO	International Civil Aviation Organization
ICMP	Internet Control Message Protocol
ID	Identifier
IDE	Identifier Extension
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IFR	Instrument Flight Rules
IFS	Interframe Space
IMA	Integrated Modular Avionics
IP	Internet Protocol

IRJES	International Refereed Journal of Engineering and Science
IRS	Inertial Reference System
ITU-R	International Telecommunication Union Radiocommunication Sector
LASER	Light Amplification by Stimulated Emission of Radiation
LCSH	Library of Congress Subject Headings
LED	Light Emitting Diode
LF	Low Frequency
LLC	Logical Link Control
LRU	Line-Replaceable Unit
LTF-UL	Luftüchtigkeitsforderungen an ein aerodynamisch gesteuertes Ultraleichtflugzeug | Airworthiness requirements for an aerodynamically controlled microlight aircraft
MAC	Medium Access Control
MDCC	Multi-Domain Combat Cloud
MFD	Multi-Function Display
MIL	Military
MPEG	Moving Picture Experts Group
MSL	Mean Sea Level
MUX	Multiplexer
NASA	National Aeronautics and Space Administration
NFS	Network File System
OODA	Observe, Orient, Decide and Act
OSI	Open Systems Interconnection
PC	Personal Computer
PFD	Power Flux Density
PhD	Doctor of Philosophy
PNG	Portable Network Graphics
PPB	Part Per Billion
PPM	Part Per Million
PPP	Point-to-Point Protocol
PTP	Point-To-Point
PWM	Pulse-Width Modulation
QTR	Quality Test Report
RC	Radio Control
RPC	Remote Procedure Call
RPM	Revolutions Per Minute
RS	Recommended Standard
RT	Remote Terminal
RTR	Remote Transmission Request
RX	Receiver
S	Seconds
SARPs	Standards and Recommended Practices
SATS	Small Aviation Transportation System

SHF	Super High Frequency
SMTP	Simple Mail Transfer Protocol
SOF	Start Of Frame
SQL	Structured Query Language
SRR	Substitute Remote Request
SSH	Secure Shell
SSL	Secure Sockets Layer
STD	Standard
SUAV	Small Unmanned Aerial Vehicle
SV	SkyView
TCAS	Traffic Alert and Collision Avoidance System
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
TTBS	Time-Triggered BUS Scheduling
TTCAN	Time-Triggered communication via CAN
TTE	Time-Triggered Ethernet
TV	Television
TX	Transmitter
UART	Universal Asynchronous Receiver / Transmitter
UAV	Unmanned Aerial Vehicle
UCAS-D	Unmanned Combat Air System Demonstration
UCHAR	Unsigned Char
UDP	User Datagram Protocol
UHF	Ultra High Frequency
V	Volt
VAR	Variation (see list of definitions)
VCC	Voltage Common Collector (positive)
VL	Virtual Link
VoIP	Voice over IP
VFR	Visual Flight Rules
VHF	Very High Frequency
VLF	Very Low Frequency
VMC	Vehicle Management Computer
VOR	VHF Omnidirectional Radio Range
WAIC	Wireless Avionics Intra-Communication
WDM	Wavelength-Division Multiplexing
XLM	Extensible Markup Language

List of Definitions

Avionics
Electrical equipment installed on board an aerospace vehicle for control, navigation and use.

Cloud
A computer network on the internet where accessible data and programs can be stored.

Fly-by-wire
Fly-by-wire means that the conventional manual flight control (mostly rods, pulleys and cables) is replaced with an electrical system controlled by computers.

Mag2True
Since air carriers, air navigation service providers and avionics manufacturers spend a lot of money each year on magnetic variation management, the entire aviation should in the future uniformly use the true north pole as a reference. AHRTAG is working with ICAO to implement this effort called Mag2True.

Simplex, Half-Duplex and Full-Duplex Mode
Simplex mode is the transmission of data from A to B but not from B to A. Half-duplex mode is the transmission of data from A to B and from B to A but not simultaneously (usually over one line). Full-Duplex mode is the transmission of data from A to B and from B to A at any time (usually over two lines, but it is also possible over one line). (see figure 0.1)

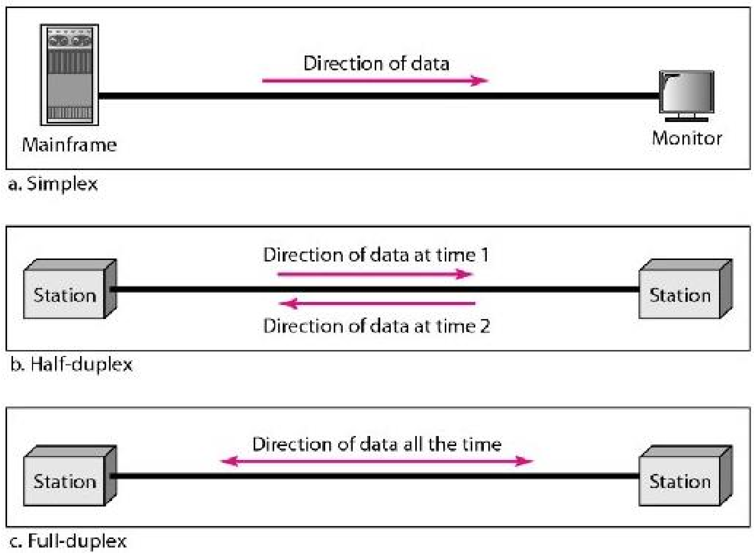


Figure 0.1 Simplex, half-duplex and full-duplex data flow (SlideToDoc 2013, page 19)

VAR

In aeronautical navigation, variation is the error resulting from the shift from magnetic north to geographic north. The variation changes at different locations in the world and must be taken into account when converting true courses, true headings, or true winds to magnetic direction. When variation is east, it is subtracted from the true heading. When variation is west, it is added to the true heading (see figure 0.2). (McBride 1976)

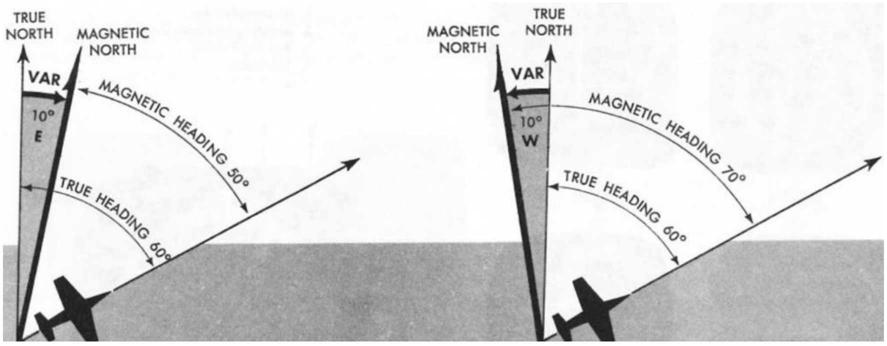


Figure 0.2 Variation in aeronautical navigation (McBride 1976, page 1-13)

1 Introduction

1.1 Motivation

In today's aviation, an increasing number of aircraft are equipped with digital onboard instruments, which increases the electrical complexity. Data communication between different devices is becoming more important. The development of own wiring concepts as well as the adaptation or the construction of avionics modules requires a basic knowledge of communication systems, which is built up in this work. In practice, this means that when installing purchased avionics equipment, no prefabricated wiring harnesses from manufacturers are needed, instead, a wiring harness with connectors can be built from scratch with a better understanding. With the understanding of data port inputs and outputs of devices, own devices can be successfully implemented into an overall system. When an aircraft is prototyped, the question arises which avionics system with which communication system should be implemented in the aircraft. A large microlight company with more resources will use a different avionics system than a prototype builder who can only rely on prefabricated overall systems. In this thesis, avionic systems and communication protocols from aviation are presented with regard to the use in microlight aircraft and future avionic systems.

1.2 Title Terminology

“Signal Transmission between Electrical Components in Microlight Aircraft"

The Cambridge Dictionary defines the words of the title as:

Signal
A series of energy (...) that carry a sound, picture, or other information.

Transmission
The process of passing something from one person or place to another.

Between
Connecting two or more places, things, or people.

Electrical
Using electricity for power, involved in the production or movement of electricity, or related in some way to electricity.

Components
One of the parts of a system, process, or machine.

Microlight

An extremely light and small aircraft with a very small engine, designed to carry only one or two people.

Aircraft

Any vehicle, with or without an engine, that can fly, such as a plane or helicopter.

1.3 Objectives

The aim of this thesis is to describe and analyze different data transmission systems in microlight aircraft. New electrical components can be integrated into an aircraft with a deeper understanding of existing data transmission modes and protocols. Furthermore, this work will help to create a new design for an avionic concept for a microlight aircraft. Along the way, a basic knowledge of communication with different transmission channels is established, a basic knowledge of how protocols work, and things to consider when implementing a communication system in an aircraft. The thesis brings us a step closer to the microlight avionic of tomorrow.

1.4 Previous Research

In the field of communication systems, there is a lot of good literature and electrical descriptions. The whole world is networked with a constantly growing majority of people having access to the World Wide Web from everywhere. Intercontinental connections include fiber optic cables laid across the ocean. Data packets and TV are being sent via satellites to receiving stations. Because of the large market, there is a lot of money involved, which is why large companies also invest research money. In aviation, the issue of electrical data transmission is much smaller. Descriptions of different protocols exist, but there is very little public work that describes the different protocols in a neutral way for a particular application in aviation. The microlight sector is still a small market, but due to modernization and increased safety, electrical understanding in this area is becoming more important to use compatible components. There are some companies that manufacture avionics components for microlight aircraft, but they try to sell a whole range of products and therefore use proprietary networks that are hardly able to connect electrical components from other manufacturers. The interfaces for communication between devices of different manufacturers have the RS-232 standard.

1.5 Literature Review

A good fundamental source for the topic of signal transmission is Carlson 2010. He describes functions and physics in a mathematically based way. Information on optical data transmission is nicely explained from the TU Dresden presentation (Hara 2015) and optical carrier manufacturer pages (Fosko 2010). The information on radio transmission can be found in the Lecture Notes of the Karlsruhe Institute of Technology (Younis 2018). Due to the large scope, from consumer tech to highly specialized technology, there is a large variety of sources used, ranging from newspapers, online articles, publications from various companies to papers from various universities. After the protocol layer overview according to ISO/OSI, individual protocols such as CAN by Corrigan 2016, CANaerospace by Stock 2005 are presented. Further information on CAN based protocols is provided by Stock 2009, a document by Klüster 2012 and the bachelor thesis by Horst 2018. Information on each protocol was gathered from various sources. ZigBee information comes from documents of the silicon labs (Silicon 2021) and the investigation of ZigBee in a flight environment comes from the PhD thesis of Hope 2011. Description of avionics come from the respective manufacturer datasheet. The understanding of PWM and the calculation of a low-pass filter comes from Mike Cook's 2021. The wiring harness of the fuel level indicator was created with the program QelectroTech 2019. Information about Future Combat Air System (FCAS) comes from Airbus 2022 and Koch 2019. The new non-military avionics system Anthem from Honeywell is well described on its own website and by Plane 2011.

The literature provided in this thesis is archived and mostly publicly accessible via the reference list. It is recommended to read the given sources in order to get a deeper insight into the topic.

1.6 Structure

The content of this thesis is structured as follows:

Chapter 2 gives a basic description of what a communication system is, why it is needed in an aircraft and where it is found. Using Mag2True as an example, it is estimated that in the future an increasing amount of avionics will be digital, making a good communication system between electrical components even more important.

Chapter 3 discusses the different transmission channels (transmission medium) and their bandwidth transmission strengths. The Chapter presents optical data transmission, radio data transmission and data transmission with metal cables. The advantages and disadvantages of the different transmission channels as well as the direct price comparison of copper cables to aluminum cables are described. At the end of the Chapter, the current wiring of the communication system of large aircraft is presented, which is very complex and involves a high installation effort.

Chapter 4 explains in general what protocols are, with examples and network topologies. Then the CAN protocol is explained in more detail with the specification for use in aircraft as CANaerospace. Electrophysical basics of data transmission are shown with protocol examples. In subsections, the Chapter goes into the description and function of many airborne protocols such as ARINC 429, 629, 664, 708, 812, 825, 826, and MIL-STD-1553, MIL-STD-1773 as well as IEEE 1384 (FireWire) and ASCB. Other protocols such as RS-232, RC transmission protocols and ZigBee, Z-Wave and EnOcean are also described at the end. The Chapter is rounded off by a decision and selection part that describes the importance of various protocols for microlight aircraft.

Chapter 5 introduces the company Uniplanes GmbH and the microlight aircraft Flywhale. Avionic components installed in various Flywhale aircraft are presented in terms of cabling and connection. Also pictures and descriptions from the wiring of microlight aircraft in a practical environment are included in the Chapter. Then a fuel level indicator is designed to meet the requirements of a Flywhale aircraft. This part of the Chapter explains the function of a PWM output signal which is converted to a smoothed DC signal by a low-pass filter with simple electrical components. Microcontroller programming, wiring and installation of the fuel level indicator into the Flywhale aircraft completes the Chapter.

Chapter 6 describes a possible future of the communication system in microlight aircraft on the basis of the most up-to-date avionics, military development decisions and wireless data communication presented in previous Chapters. The future vision of the Flywhale aircraft as DS-2C, marketed by Dornier Seawings, is also presented.

Chapter 7 represents a final summary and conclusion of the thesis.

2 Communication System

Today's aircraft have a variety of electrical components that must communicate with each other and provide information to the pilot via LED's, displays or audio signals. Even the smallest civil aircraft are required to have a radio on board to ensure air traffic is safe at all times. This allows communication between the pilot in the air and a person on the ground. The larger and more complex the aircraft, the more devices are on board that must have a communication system to be controlled.

Since the beginning of aviation, navigation has been done with the help of a magnetic compass. Until today, due to an international agreement, we still fly with the course reference of the geomagnetic north pole, so that all pilots follow the same rules. Practically, almost no one navigates purely by an analog magnetic compass anymore, even though it is still taught in VFR flying.

Magnetic compasses do not point to true north, they point to magnetic north which is about 500 nm (926 km) from true north. The difference is about 9.5 deg inclination to the rotational axis of the earth. The error is not static, the magnetic north drifts slowly and unpredictably. Systems that use the magnetic compass must be constantly updated. Especially for long (intercontinental) flights the local angle difference is added which changes during the flight. A modern Flight Management System (FMS) can automatically compensate for many of these variables, but only if updated regularly. The alternative is GPS-based navigation, which is combined with the Inertial Reference System (IRS) in larger aircraft. Even in the event of GPS signal failures or local disturbances, the autonomously operating IRS can continue to operate as a dead-reckoning navigation system until the GPS signal is restored. Smaller aircraft that do not have an IRS system would have to have a magnetic compass as a backup in case of GPS failures. For several years, the international Aviation Heading Reference Transition Action Group (AHRTAG), which coordinates with the International Civil Aviation Organization (ICAO), has been demanding the transition from magnetic north to true north (Mag2True) by March 2030. Since the geomagnetic north pole is located in the northern territory of Canada, Canada has been requiring that all northern flight operations use the True North heading reference for a long time. Thus, Canada already has a long history of practical experience in this area and plays a major role in the AHRTAG. In Figure 2.1 *Strength of global magnetic variation* is a world map of magnetic declination. The green line presents the locations with a VAR of 0°. Next to it, the VAR increases rapidly to 10° and more. (Learmount 2021)

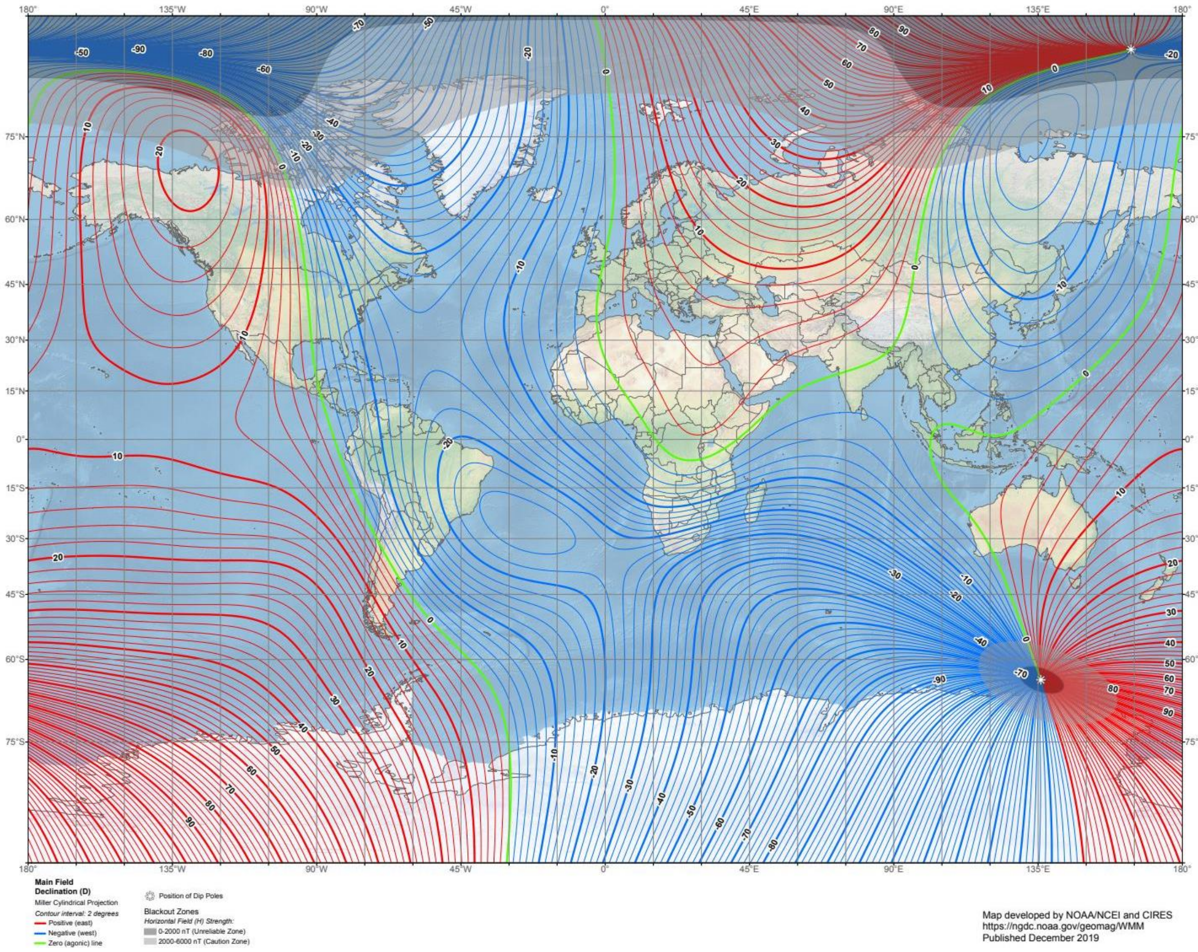


Figure 2.1 Strength of global magnetic variation 2020 (NOAA 2019)

With adaptations such as Mag2True and other modernization in the aircraft, it is even more important that good signal transmission and processing takes place. Multiple GPS sensors and magnetic compass sensors provide data that is constantly evaluated. In the event of a sensor error, the system should be able to decide to stop using the sensor. In smaller aircraft, the compass system could use magnetic and GPS sensors to constantly calculate the VAR and continue to work with the last determined VAR in the event of a GPS disturbance. In this way, a high-quality true north direction could be displayed with simple components, so there would be no problem for Mag2True in aviation.

The Air Transport Association of America (ATA) defines a standard for the technical documentation of air traffic systems with which different areas can be presented in chapters. The ATA 23 *Communication* chapter includes all components which allow crew members to communicate with each other, with the passengers, other aircraft and the ground stations including voice, data, continuous wave (CW) communicating components, intercom and voice recorder (Scholz 2002, *Avionic Systems: Communication*). Today's aircraft also have a variety of antennas for radio communication, radio navigation and radio surveillance systems in the aircraft (see Figure 2.2).

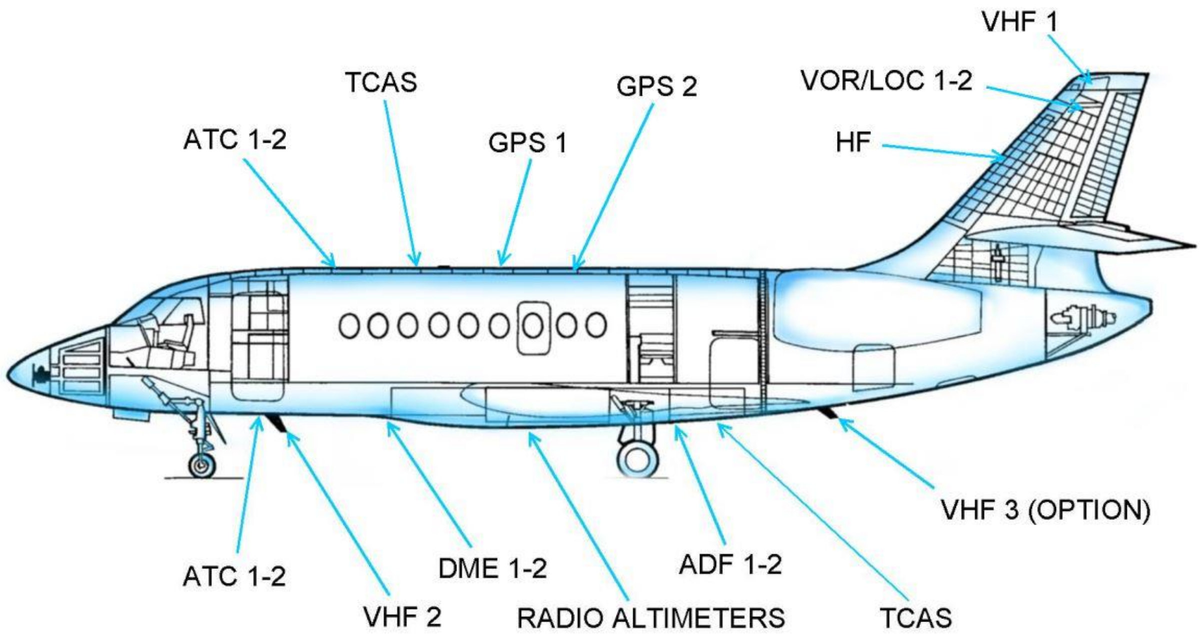


Figure 2.2 Antennas on a Dassault Falcon 2000 EX (Dassault 2000, ATA 23)

Communication systems are also used in many other chapters of the ATA, but at the system level. For example, the generator, APU and batteries must be regulated (ATA 24 - *Electrical Power*) or the entire aircraft is controlled by fly-by-wire, in which various control systems for the aileron, rudder, elevator, leading-edge slats, flaps and airbrakes must be operated (ATA 27 - *Flight Controls*). Also in small private aircraft, more and more electrical components are used to simplify flying. Customers of private aircraft expect increasingly more user interfaces (such as large screens with touch) which they know from consumer electronics and cars. To make flying comfortable and safer, more and more microlight aircraft have electric trim and digital displays connected to external or internal sensors. So how do all these components communicate with each other?

Communication is the transmission of information from a source to a destination. The type of information is very different. The information can be spoken (audio), it can be visual images or switches that are on or off. In order for the information to get from the source to the destination, it is first converted into a signal that can go through the communication system and then end up back in its original form, like the spoken word into a microphone that is converted into an electrical signal and then converted back again to create sound in the speaker. Figure 2.3 shows the general information transfer from a source like a sensor whose signal is converted into an input signal for the communication system. Depending on the amount of information (bandwidth) and the distance that has to be covered, the communication system can vary greatly in the technology and medium used for transmission. Generally, in a communication system, there is a **transmitter** that processes the input signal to produce a transmitted signal that is matched to the characteristics of the **transmission channel**. The transmitted signal is usually modulated and can also be coded. The **receiver** processes the received signal into an output signal which is as close as possible to the input signal (see Figure 2.4). (Carlson 2010, page 2-4)

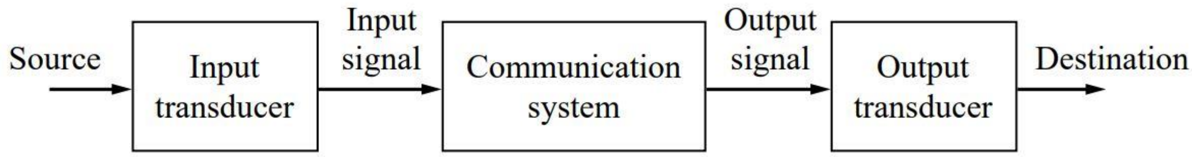


Figure 2.3 Communication system with input and output transducers (Carlson 2010, page 3)

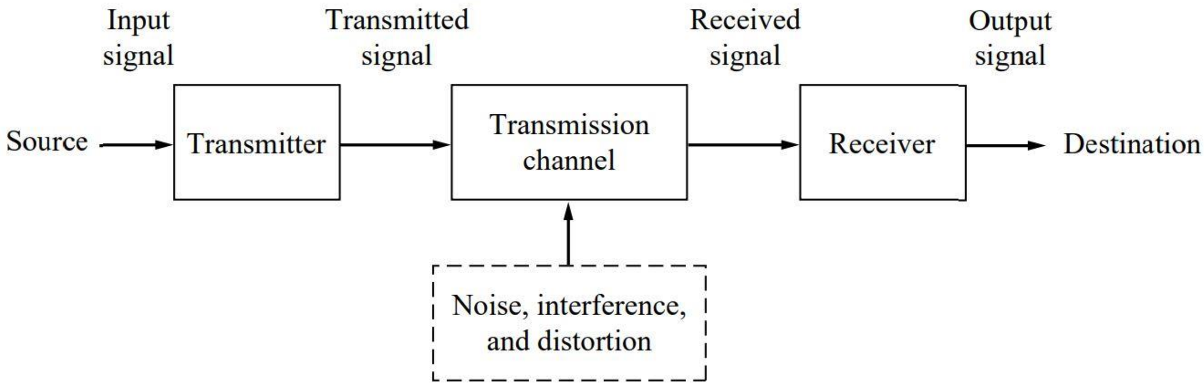


Figure 2.4 Elements of a communication system (Carlson 2010, page 4)

There are many different sources of information that can be classified as **analog** or **digital** for the best possible fidelity. Analog messages are the acoustic pressure on a microphone, the resistance tank level sensor, the angular position of an aircraft gyro or a brightness sensor for light intensity. Since the information is in a time-varying waveform, an analog communication system can transmit the waveform with the best fidelity. A digital message is an ordered sequence of symbols. The letters in text or the listing of hourly temperature readings should be able to be transmitted by a digital communication system with the best possible fidelity because this information contains discrete symbols. (Carlson 2010, page 3)

3 Transmission Channel

As already mentioned, the transmitter processes the input signal and generates a transmit signal that is adapted to the characteristics of the transmission channel. The transmission channel is a medium that bridges the distance between source and destination. It can be wires carrying an electric current, a radio wave transmitted through air, water or vacuum, or a laser beam transmitted through the air or another optical waveguide such as glass or plastic. Each channel causes a transmission loss (attenuation), so that the signal power decreases with increasing distance. (Carlson 2010, page 4)

The question of which medium and material should be used for the transmission channel depends on many factors. The most important factors are at what distance how much data will be transmitted. If large data is to be transmitted, a large bandwidth is required, which means that a high frequency should be used to transmit a lot of data in a short time. In addition, the decision of a transmission channel is usually based on cost, effort and weight. For the transmission of power and simple information, mostly copper cables are used, because they have good conductivity and good formability characteristics. Slowly, aluminum cables are also being used, which offer different characteristics (more in Chapter 3.4, *Copper and Aluminium Cables*). For data connections with high bandwidths, optical fiber is a very good choice and also offers advantages in terms of electromagnetic interference fields (more in Chapter 3.1, *Optical Transmission*). Radio technology is also important in aircraft for communication between aircraft or from the aircraft to the ground. An Internet data connection also partly exists, but locally in the aircraft as a transmission channel between the components, the technology is almost non-existent. In microlight aircraft, some onboard devices can transfer their data via Bluetooth or WLAN to mobile devices such as cell phones or tablets, allowing settings to be made or the mobile device to be used as a display. It would be an interesting attempt to design a microlight aircraft in which many sensors and electrical components communicate with each other via wireless technology (more in Chapter 4.8, *ZigBee, Z-Wave and EnOcean*). However, most components would need to be powered by batteries or most likely cables to the nearest power source. The wiring harness could be greatly simplified, as the cables would only need to be pulled to the nearest power bus and not to a specific electrical component (as would be the case with a wired communication). However, a radio network must be established that does not interfere with other devices and has the required communication properties (more on *Radio Waves* in Chapter 3.2).

3.1 Optical Transmission

Optical data transmission is usually chosen for very high data link speeds or high bandwidths. Further advantages to conventional cable-bound data transmission are the quite low attenuation for higher distances as well as complete electrical isolation of transmitter and receiver. There is also no electromagnetic interaction with the environment, which makes it also resistant to an ElectroMagnetic Pulse (EMP). This can be important for military purposes or for rockets in space missions as EMPs can be generated by hostile states, terrorists or naturally by the sun (Smith 2014). Optical fiber cables are usually much thinner and lighter. Coupling and connection are technically very demanding, as well as the integration of switches. Cable curve radiuses have to be planned larger. Since optical transmission is usually used for data transmission over longer distances, attenuation is an important issue. The better the optical transmission, the less or no optical amplifiers have to be built into the system. For optical transmission in a fiber, attenuation is the energy loss of the light beam caused by scattering and absorption. In addition, attenuation depends on the wavelength of the light beam. For high-purity fused silica, there are three useful optical regions at 850 nm, 1300 nm, and 1550 nm (see Figure 3.1). (Hara 2015)

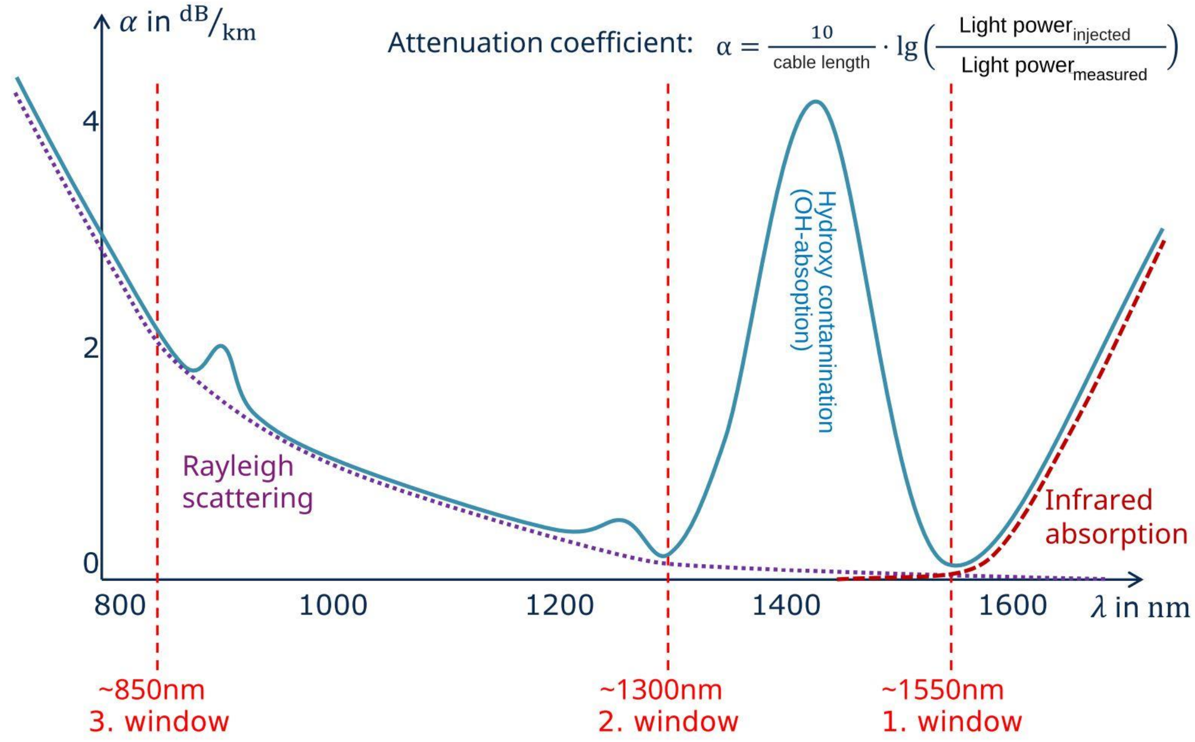


Figure 3.1 Attenuation in high purity quartz glass and 3 optically usable windows (Hara 2015)

The 1st window at a wavelength of 1550 nm is the absolute attenuation minimum from Rayleigh scattering and infrared absorption but the fabrication and implementation costs are expensive. The 2nd window is a local attenuation minimum at a wavelength of 1300 nm and the 3rd window is the most easily usable window at 850 nm since transmitting and receiving units are cheaper and easily manufacturable LASER LEDs can be used (visible red region).

The conversion from wavelength λ to frequency f is done with the formula

$$f = \frac{c}{\lambda} \tag{3.1}$$

$$\text{with } c = 299\,792\,458\ \frac{m}{s}$$

so that 850 nm is about 352.7 THz.

$$f = \frac{299\,792\,458\ \frac{m}{s}}{850 \cdot 10^{-9}\ m} = 3{,}5270 \cdot 10^{14}\ Hz$$

With this high frequency, a large bandwidth is possible, with which large amounts of data can be transmitted. The practical application is in the range of gigabits per second per fiber (without WDM). In the overview Table 3.2 *typicall valuas of transmission loss*, it is possible to see below the three useful optical windows with the corresponding loss in dB/km.

The material properties of the optical fiber are very important to keep the attenuation low. Smallest amounts of impurity ions, like Fe2+, Cu2+, Cr3+ and OH- ions, lead to losses. The purer the quartz glass, the narrower and lower are the curves shown in Figure 3.1, such as the Hydroxy contamination curve at about 1450 nm, caused by the OH ion (water dissolved in the glass), and the infrared absorption starting at about 1600 nm, caused by the SiO2 in the glass. The three useful optical windows become wider (wider wavelength spectrum) with lower attenuation due to material purity. Table 3.1 shows how strongly the contamination of various ion components causes attenuation in the quartz glass. (Fosko 2010)

Table 3.1 Optical attenuation in quartz glass due to contamination (Fosko 2010)

Impurity Ion	Loss due to 1 ppm[a] of impurity [dB/km]	Absorption Peak Wavelength [nm]
Fe^{2+}	0.68	1100
Fe^{2+}	0.15	400
Cu^{2+}	1.1	850
Cr^{2+}	1.6	625
V^{4+}	2.7	725
OH^{-}	1.0	950
OH^{-}	2.0	1240
OH^{-}	4.0	1380

[a] 1 ppm equals 0.000 1 %

Furthermore, it can be seen that, for example, 1 part per million (ppm) of Fe2+ leads to an additional loss of 0.68 dB/km at 1100 nm and 0.15 dB/km at 400 nm. It is now possible to produce ultra-pure quartz glass with impurities of less than 1 part per billion (ppb ≙ 0.000 000 1 %). The result is extremely low-loss fibers with a wide low-loss window, enabling the use of wavelength-division multiplexing (WDM) (see Figure 3.2). (Fosko 2010)

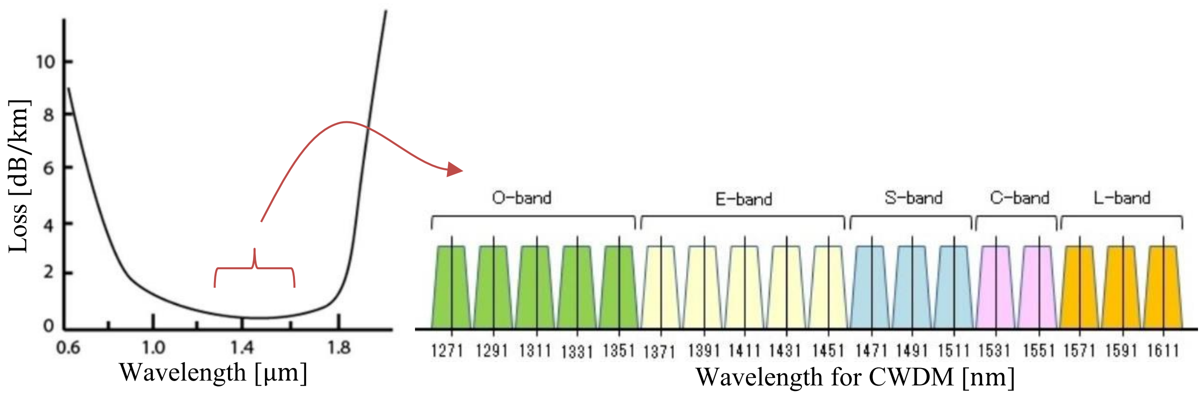


Figure 3.2 Low loss silica glass fibers for WDM, left (Fosko 2010) and wavelength channels, right (Fiberlaps 2021)

Figure 3.2 shows a division into 18 wavelengths with 20 nm spacing. This was approved by the International Telecommunication Union in December 2003 as coarse wavelength-division multiplexing (CWDM) (ITU-T G.694.2). Figure 3.3 shows the process of WDM, where several individual signals at specific wavelengths are sent into a wavelength multiplexer (MUX) on the transmitter side. The information is then mixed and passed through the optical fiber, and then translated by a wavelength demultiplexer (DEMUX) back into the individual signals of the specified wavelengths. (Fiberlaps 2021)

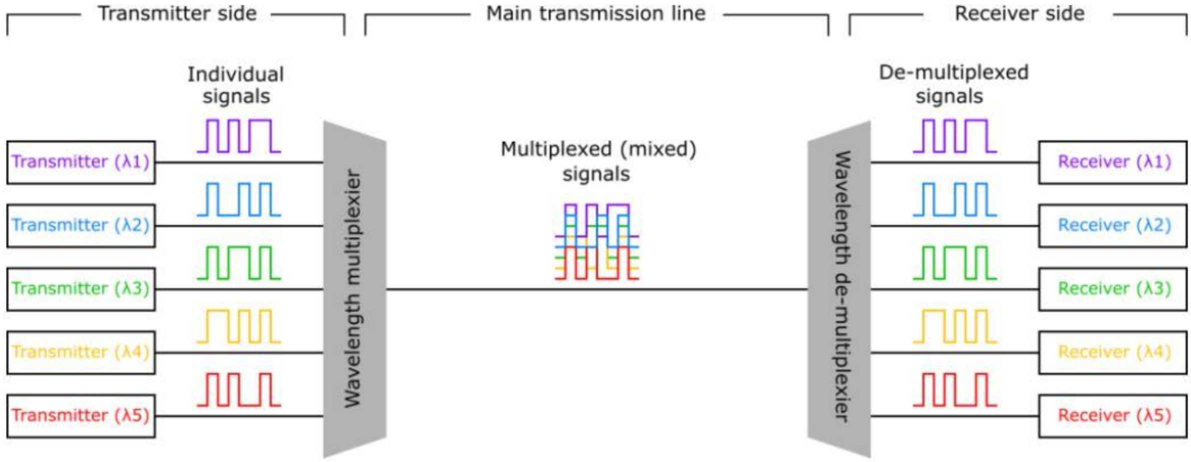


Figure 3.3 Schematic of WDM transmission system (Fiberlabs 2021)

In most applications, cheaper MUX and DEMUX are used, which use 4 wavelengths from 1531 to 1591 nm or 8 wavelengths from 1471 nm to 1611 nm. With the WDM method, even more information can be transmitted with only one optical fiber. The need to send as much information as possible over as few fibers as possible is higher the longer the distance to be bridged. Complicated, more expensive WDM methods that can send many wavelengths over only one fiber are therefore more likely to be used for distances greater than 100 km. (Fiberlaps 2021)

For distances of only a few meters, such as in microlight aircraft, but where a lot of data is to be transmitted from one device to another, the attenuation of the optical fiber is not decisive. Therefore, it is possible to stay in the more affordable range of the 3rd window at 850 nm or even choose an optical cable made of plastic.

3.2 Radio Waves

As mentioned at the end of Chapter 3 *Transmission Channel*, it would be interesting to design a microlight aircraft where many sensors and electrical components communicate with each other via wireless technology. Most of the sensors only need to transmit very simple information like the information of the angle of attack, air pressure at the leading edge of the wing, a temperature outside and inside, the information of a switch for the current position of the landing gear and the landing gear doors or the resistance of a tank sensor for the fuel level indicator. All devices and sensors that only need to transmit such simple information with very small bandwidth could be transmitted using ZigBee or Z-Wave protocols which are used in the Smarthome area. These protocols form a transmission mesh network that complements itself and does not interfere with each other (more information in Chapter 4.9, *ZigBee, Z-Wave and Enocean*). But how does wireless communication actually work?

Radio waves as used for radio and satellite TV are a time-varying electric and magnetic field that can propagate through space from one point to another. Information is converted by a transmitter into an electromagnetic wave that propagates äthrough space. This means that the electromagnetic wave is conducted through a cable to an antenna, which transmits the wave into free space. If the conductivity of a medium changes, so does the self-impedance, which in free space results in a constant value of about 377 Ω from the ratio of the electric field E_0 and the magnetic field H_0. (Younis 2018, Chapter 1.2)

$$Z_0 = \frac{E_0}{H_0} \approx 377\ \Omega \tag{3.2}$$

The task of the antenna structure is to produce a current distribution in such a way that a free space wave is generated. In this wave, the electric and magnetic fields reproduce each other. In other words, the electric field initiates the magnetic field and is therefore the source of the magnetic field. Since the two fields affect each other, there is a reactive near field in the very close range of the antenna where a direct influence takes place. In the radiating far field, the electric current density is zero (see Figure 3.4). (Younis 2018, Chapter 1.2)

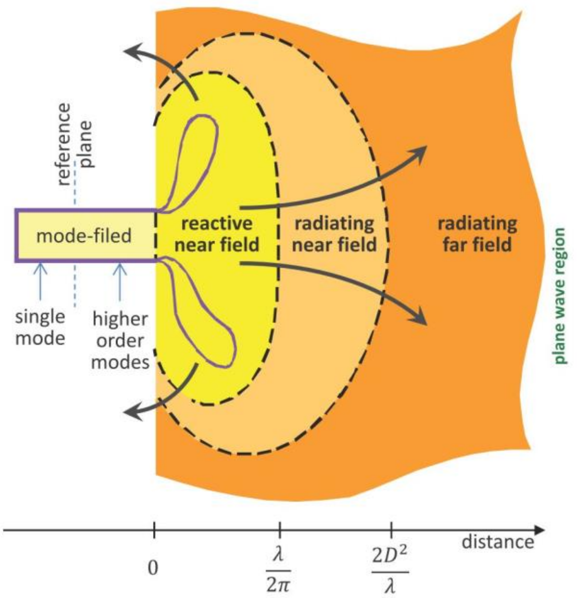


Figure 3.4 Field regions of an antenna (Younis 2018, Chapter 1.3.1)

The antenna architecture can be very different depending on the frequency to be used and the application. There are wire antennas, which are mostly used in the lower frequency range (MHz), aperture antennas, microstrip/patch antennas, parabolic (reflector) antennas, lens antennas, and reflectarrays (see Figure 3.5). A reflectarray is fed by an antenna and usually consists of many small array elements of various sizes, variable-length phased delay lines, variable angle rotations, and ferroelectric material. In this way, after adjusting the reflectarray (flat surface), the same reflector effect can be achieved as with a parabolic reflector. (Younis 2018, Chapter 1.3)

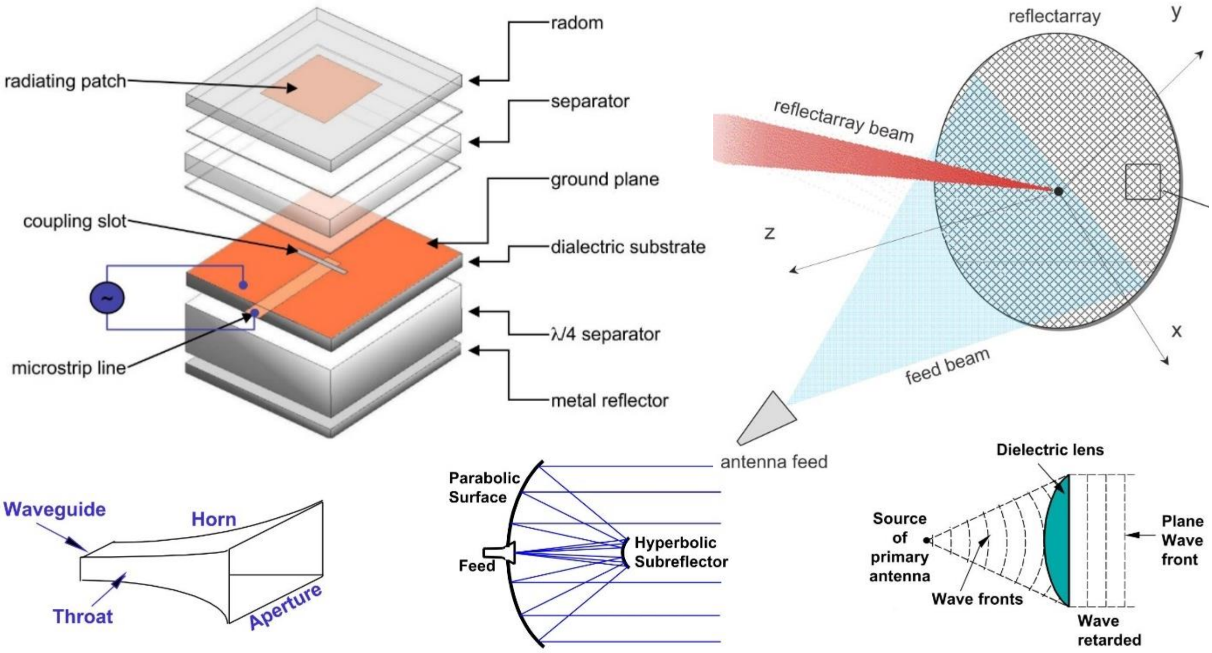


Figure 3.5 Microstrip-, aperture-, parabolic (reflector)-, lens- and reflectarrays (with feed) antennas (Younis 2018, Chapter 1.3)

Antennas have a certain directional effect in which they radiate amplified. This means that they radiate with different strengths in different directions. A theoretical antenna that radiates evenly in all directions is called an isotropic radiator. Mathematical calculations can be simplified and illustrated in this way. The transmission path between transmitter and receiver is very unpredictable for radio channels due to topographic reflection, diffraction and scattering. In reality, locations for large antennas are determined by measurements, statistics and tests. Only in very few cases is a permanent line-of-sight between transmitter and receiver available, where the transmission path is predictable. To measure the quality of an open field transmission channel, the attenuation is specified, which results from the 20 times logarithmic ratio of input voltage to output voltage (L_U) or the 10 times logarithmic ratio of input power to output power (L_P). (Younis 2018)

$$L_U = 20 \cdot \log\left(\frac{U_t}{U_r}\right) \text{ dB} \tag{3.3}$$

$$L_P = 10 \cdot \log\left(\frac{P_t}{P_r}\right) \text{ dB} \tag{3.4}$$

When designing a radio transmission channel, attenuation must be taken into account as well. This is referred to as propagation and free-space attenuation, which means that the field strength is reduced by absorption and reflection. The radio wave is also reflected and absorbed by the topography of the environment and foreign objects such as clouds, rain or snow in the air. In addition, the energy of the radio waves decreases with increasing distance. This is the same principle as the rule from acoustics that states that when the distance doubles, the sound energy decreases by 6 dB. Thereby, the radio wave power is measured in the power flux density W/m².

3.3 Transmission Loss

All types of transmission channels have a transmission loss that should not be neglected. When powering the devices, a minimum cable diameter (usually specified by the manufacturer) must be observed in order to keep losses but also heating low. The attenuation in metallic cables depends mainly on the conductivity, the cable length and the frequency. Due to capacitive coupling and inductance in the cable, high frequencies are attenuated more than low frequencies. Furthermore, cables can be attenuated by electromagnetic interaction from surrounding conductors or devices (crosstalk attenuation). (Carlson 2010)

Optical losses can be reduced by the purity of the optical conductor, which is more problematic at long distances. In aircraft, the loss in optical cables is negligible due to the short distances. Table 3.2 *Typical values of transmission loss* show the typical values of signal loss for each transmission medium. When the frequency is increased, for example for a higher data rate, it can be seen that the loss increases rapidly, so it makes sense to select the next better transmission medium if the loss becomes too high.

Table 3.2 Typical values of transmission loss (Carlson 2010, page 118)

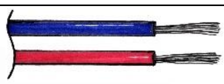
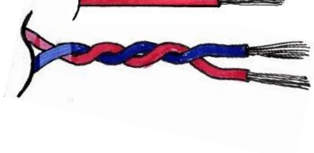
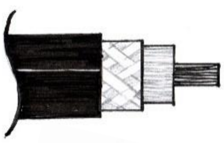
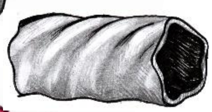
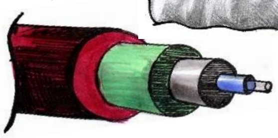
Transmission Medium	Frequency	Loss [dB/km]
Open-wire pair (0.3 cm diameter) 	1 kHz	0.05
Twisted-wire pair (16 gauge) 	10 kHz	2
	100 kHz	3
	300 kHz	6
Coaxial cable (1 cm diameter) 	100 kHz	1
	1 MHz	2
	3 MHz	4
Coaxial cable (15 cm diameter)	100 MHz	1.5
Rectangular waveguide (5 x 2.5 cm) 	10 GHz	5
Helical waveguide (5 cm diameter) 	100 GHz	1.5
Fiber-optic cable 	$3.6 \cdot 10^{14}$ Hz	2.5
	$2.4 \cdot 10^{14}$ Hz	0.5
	$1.8 \cdot 10^{14}$ Hz	0.2

Figure 3.6 shows the application areas of the different wavelengths with the corresponding transmission channel and the representative application. The frequencies can also be read directly with the wavelengths.

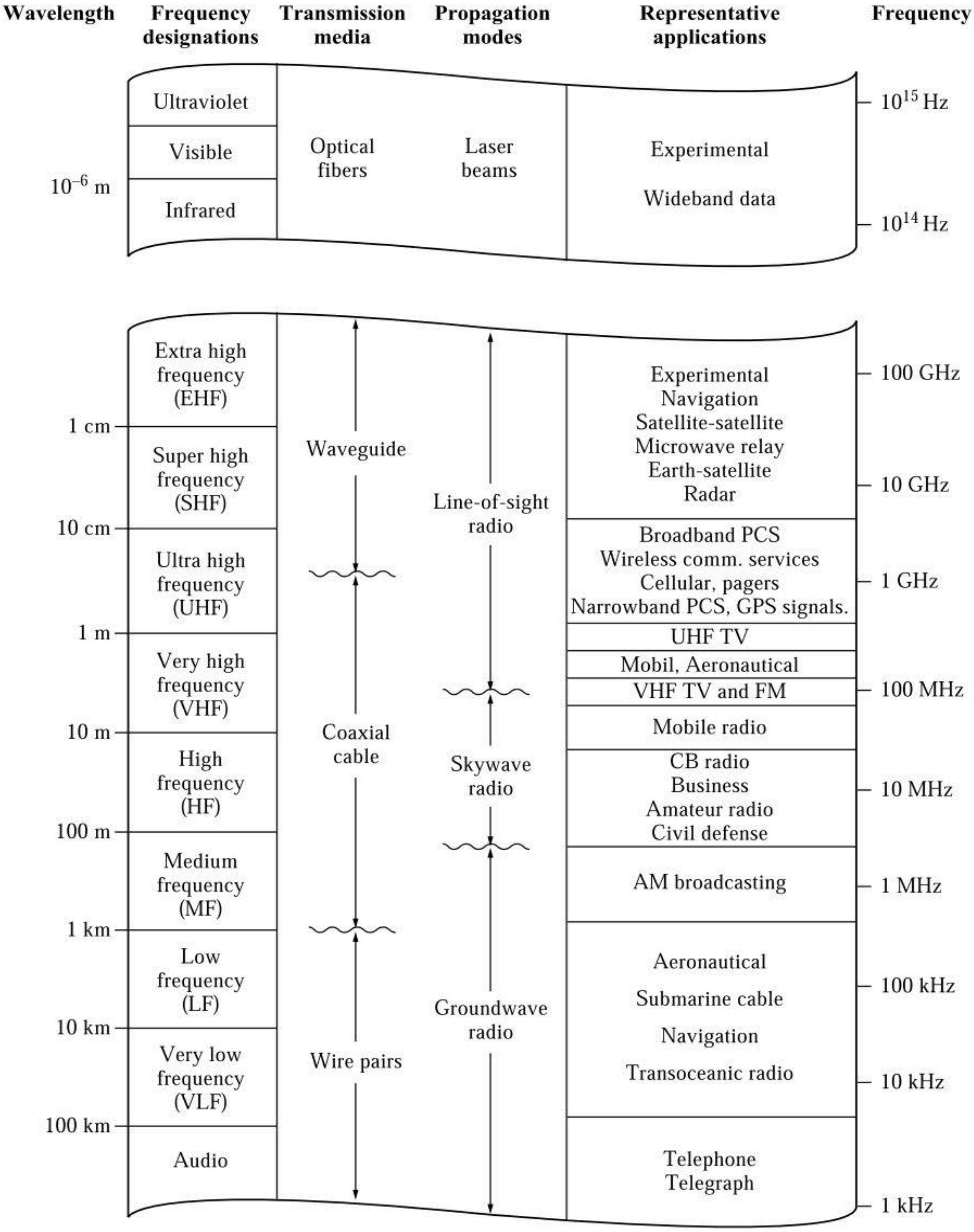


Figure 3.6 Elektromagnetic spectrum (Carlson 2010, page 9)

3.4 Copper and Aluminium Cables

In large aircraft, the transmission channels between different devices are usually in the lower three-digit meter range, while in private aircraft the distances are often only a few meters. This is done conventionally with copper wires for the power supply of the devices and to provide a data connection. In order to avoid too much transmission loss (especially in the cables with a high current flow), the wire diameters have to be large enough. Copper is relatively heavy at 8920 kg/m³ (see periodic table in Appendix A) and has become more expensive over the last few years with strong price fluctuations, many companies have looked for alternatives (see *Long-term copper price* in Figure 3.7).

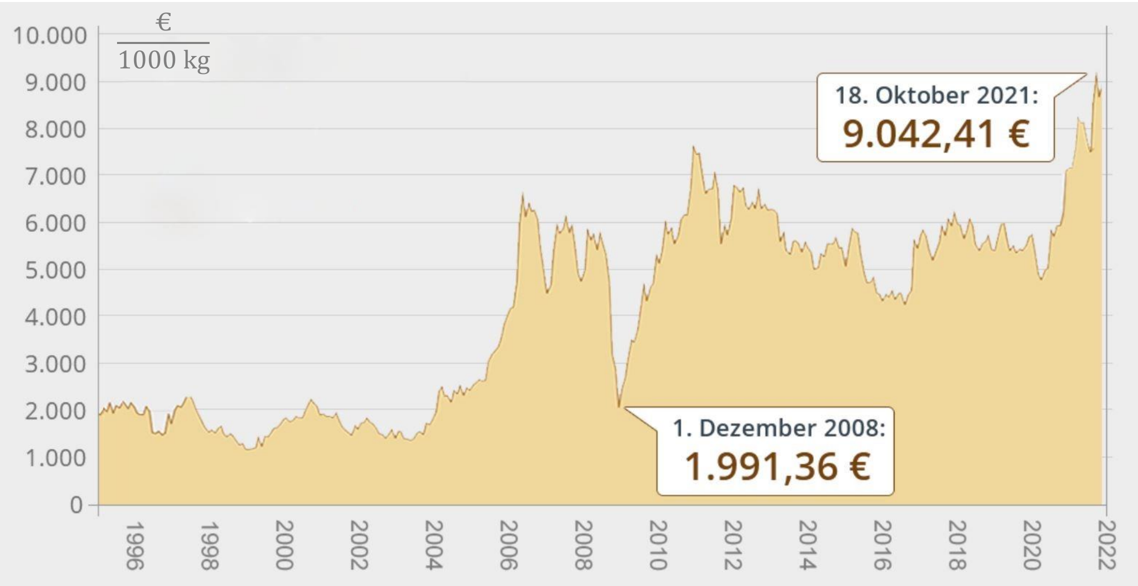


Figure 3.7 Long-term copper price, 1995 - 2021 in euros/tonnes (Gold 2021)

The choice increasingly shifts to aluminum with a weight of 2700 kg/m³ (see periodic table in Appendix A), although the light metal has only about 65 % of the conductivity of copper (see Table 3.3).

Table 3.3 Specific resistances of metallic materials (Fassbinder 2010)

Element	Symbol	Conductivity (at 20°) $[\Omega \cdot \mathrm{m}]$
Silver	*Ag*	$0.0160 \cdot 10^{-6}$
Copper	*Cu* (99,95%)	$0.0175 \cdot 10^{-6}$
Gold	*Au*	$0.0220 \cdot 10^{-6}$
Aluminum	*Al*	$0.0270 \cdot 10^{-6}$
Copper alloy	*CuCrZr*	$0.0375 \cdot 10^{-6}$
Tungsten	*W*	$0.0550 \cdot 10^{-6}$
Brass	*CuZn37*	$0.0645 \cdot 10^{-6}$
Iron	*Fe*	$0.1000 \cdot 10^{-6}$
Lead	*Pb*	$0.2080 \cdot 10^{-6}$
Coal	*C*	$40.000 \cdot 10^{-6}$

It can be seen that silver is the best-conducting metal, closely followed by copper and gold. Silver and gold are very expensive metals. **Silver** currently costs a little more than **650 €/kg** and **gold** currently costs a little more than **51,000 €/kg**. In comparison, Figure 3.7 shows that **copper** currently costs just over **9 €/kg**. Nevertheless, gold is used as a thin coating for electrical conductors due to its good oxidation properties in the field of connectors. Since gold is also a soft metal, switch bouncing is prevented. Small impurities such as dust particles are pressed into the metal surface, resulting in better switching properties. Hard contacts without gold coating keep bouncing off each other when switching and generate impulses and high contact heat at higher currents. Gold plated switches are therefore used more for signal connectors and low current applications. For higher current applications, a silver coating is usually used. This oxidizes quite quickly, but when the contact bounces during the switching process, a thin layer of silver is removed, leaving the contact surface blank. The electrical conductor of a general switch is usually made of copper with a silver layer and a thin gold coating. When switching signals and other low currents, the gold prevents the contacts from bouncing and heating up. When switching higher currents, the gold evaporates. Afterward, the contact surface consists of silver which remains blank when switching higher currents. If signals and low currents are switched, the oxidized surface of the silver cannot be removed and the switch gets more and more resistance in addition to the pulses of contact bouncing. A good source for further research on this topic is Song 2012 and Vinaricky 2016.

Aluminum as an electrical conductor material is quite suitable at a current price of just over **2.50 €/kg**. Therefore, aluminum costs about 28 % of copper.

$$\frac{\text{price } Al}{\text{price } Cu} = \frac{2.5\ \frac{€}{\text{kg}}}{9.0\ \frac{€}{\text{kg}}} = 0.2\overline{7}$$

The conductivity of aluminum is about 65 % of copper.

$$\frac{\text{conductivity } Cu}{\text{conductivity } Al} = \frac{0.0175 \cdot 10^{-6}\,\Omega \cdot \text{m}}{0.0270 \cdot 10^{-6}\,\Omega \cdot \text{m}} = 64.815\ \%$$

From this, it can be deduced that an aluminum cable has about 1.54 times the cable cross-section of copper and therefore about 1.24 times the cable diameter for the same connectivity.

$$\mathrm{A}_{Al} = \mathrm{A}_{Cu} \cdot \frac{0.0270}{0.0175} = \mathrm{A}_{Cu} \cdot 1.54286$$
$$\mathrm{A} = \pi \cdot r^2$$
$$\pi \cdot {r_{Al}}^2 = \pi \cdot {r_{Cu}}^2 \cdot 1.35185$$
$$r_{Al} = r_{Cu} \cdot \sqrt{1.54286}$$
$$r_{Al} = r_{Cu} \cdot 1.2422$$

Replacing copper cables with aluminum cables requires about 1.54 times more material at a price that is about 28 % of copper. For a company, simply replacing copper cable with aluminum cable results in price savings of about 57 %.

$$\text{price}_{Al} = \text{price}_{Cu} \cdot \left(\frac{\text{current aluminium price}}{\text{current copper price}}\right) \cdot 1.54286$$
$$\text{price}_{Al} = \text{price}_{Cu} \cdot 0.2\overline{7} \cdot 1.54286$$
$$\text{price}_{Al} = 0.42857 \cdot \text{price}_{Cu}$$

The pure weight saving when changing from copper cables to aluminum cables is about 46.7 %, taking into account that about 1.54 times more material is required.

$$\mathrm{m}_{Al} = 2\,700\ \frac{\mathrm{kg}}{\mathrm{m}^3}$$
$$\mathrm{m}_{Cu} = 8\,920\ \frac{\mathrm{kg}}{\mathrm{m}^3}$$
$$\frac{\mathrm{m}_{Al}}{\mathrm{m}_{Cu}} = 30.269\ \%$$

$$\frac{\mathrm{m}_{Al,new}}{\mathrm{m}_{Cu}} = \frac{2\,700\ \frac{\mathrm{kg}}{\mathrm{m}^3} \cdot 1.54286}{8\,920\ \frac{\mathrm{kg}}{\mathrm{m}^3}} = 46.701\ \%$$

However, there are also a few weaknesses that must be considered when designing aluminum cables. Aluminum does not bend well around corners, so larger curves must be planned. Also, due to the creep behavior of aluminum, classic connectors and connections cannot be used. A classic crimp connection would loosen over time and lead to loose contact. Since the surface of aluminum oxidizes quickly, the result is a loose connection and, in the worst case, a cable fire. Copper-plated aluminum connectors should be avoided if possible as copper and aluminum are electrochemically not compatible, which would lead to severe corrosion (exception and countermeasures in the following). Special aluminum plugs must be used that ensure sufficient contact pressure even after a long time, such as plugs with springs. In order to really save weight and costs with the extra effort, it is only worthwhile in civil aviation or in the automotive sector to design cables with high power transmissions in aluminum. These are usually the lines from the power generator on the engine (alternator) to the battery and to busbars as in the fuse bus.

Companies such as Helukabel, have specialized in aluminum cable to produce an extremely fine-stranded aluminum cable that has almost as good bending properties as copper cable (see Figure 3.8).

Figure 3.8 Aluminum cable from the company Heluwind (Helukabel 2015)

The cables shown in Figure 3.8 are mainly used for wind turbine wiring. Especially the aluminum cables with large core diameters such as the Heluwind WK Powerline Alu (in Figure 3.8, bottom left and top right). Due to the lower self-weight, longer sections of a cable can be routed to the converter without major interruptions. As already described, the connection of aluminum cables is a very special challenge. Herkules offers a C8 crimp as a solution, which penetrates very deeply into the aluminum strand bundle (filling value 95 %) and, due to the high pressure, rips open the individual strand surfaces for even contacting (see Figure 3.9, left). (Helukabel 2015)

Figure 3.9 Cross-section of an aluminum C8 crimp in the cable lug (left) and AL/CU cable lugs (Helukabel 2015)

As already mentioned, corrosion can occur especially when using a noble higher potential such as copper, iron or brass with aluminum. The electrochemical reaction or oxidation is only possible with electrolytes. This is usually condensation and general moisture. In an aluminum/copper cable lug, the copper is the anode, the aluminum is the cathode and the water is the electrolyte. The voltage generated is short-circuited across the copper/aluminum contact, which stimulates a decomposition process of the aluminum (copper is not decomposed). The contamination of the smallest copper particles causes blooming oxidation on the aluminum. This would lead to increased contact resistance and then to a drastic increase in temperature. The AL/CU connectors are manufactured using the friction welding process, which prevents the penetration of moisture. The middle cable lug in Figure 3.9, a shrink sleeve is pulled over the cable after crimping, protecting the metal surface from moisture. Good protection is also provided by internal adhesive shrink sleeves to prevent moisture penetration (see right cable lug in Figure 3.9). Nevertheless, it is recommended to perform regular maintenance according to the load of the connection area in order to control the creep behavior and possible surface oxidation. (Helukabel 2015, page 47-48)

The automotive and aviation industries have been researching aluminum cables for some time. For example, entire cable harnesses in the Airbus A380 aircraft are made of aluminum. High-voltage overhead transmission lines have also been made of aluminum for a long time because the low weight of the cable increases the distance between the masts and therefore reduces the cost of the entire installation.

One approach of wiring in the automotive sector is to try to change the type of wiring, for example by printing cables or installing cable harnesses automatically, as the manual effort for installing cable harnesses is currently very high. To save weight and costs, more and more aluminum cables are being used instead of copper, as well as fiber optics for optical data transmission. The cost of a complete wiring harness with installation is one of the most expensive components in a car. In the field of large-scale production, however, it is possible to develop company internal connectors, which means that only one connector is needed for an electrical device. Figure 3.10 shows on the left a car body after it comes out of the paint shop without seats and interior covering but with the first component installed, the wiring harness. The right side of Figure 3.10 shows various connectors, including a white connector with 40 pins for signals and low currents and other larger slots for higher currents. Such typical customized device connectors are usually found only in products of higher quantities. (Spilok 2021)

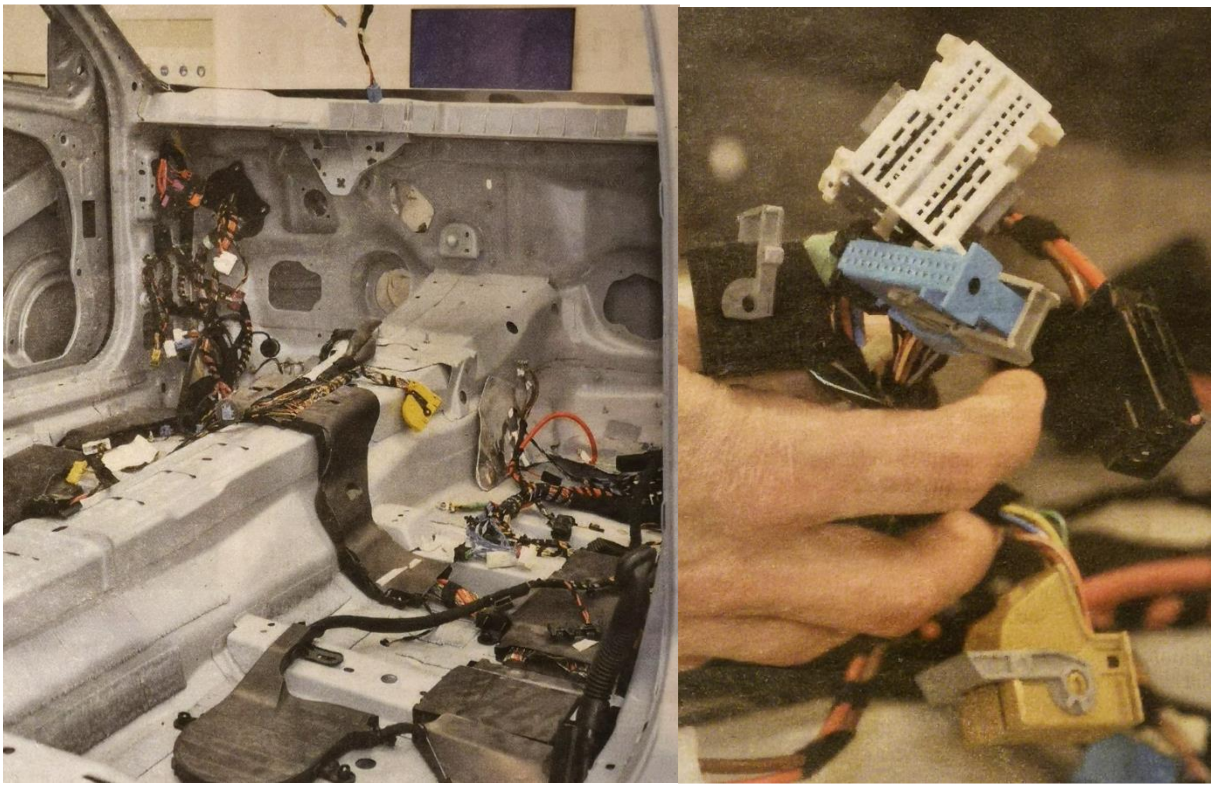


Figure 3.10 Car wiring harness (left) and plug connectors (right) (Spilok 2021)

In most cases, communication between components is still established via copper cables, which represent the transmission channel of the communication system. Copper cables have a long history and are more proven than aluminum. In addition, more electronic components, cables, connectors, connections are typically designed for copper cables. Copper cables are more expensive to purchase but require less maintenance and have smaller cable cross-sections. Aluminum cables are less expensive, lighter, but more prone to breakage (not as flexible) and require more expertise for proper installation. It makes more sense to make cable connections with shorter distances and less current flow in copper (data connection) while it can make more sense to make long-distance or high-current flow cables in aluminum.

For aircraft, the development and installation costs of the entire wiring harness assume a large part of the total costs. Airbus also had to delay delivery of the A380 several times around 2007 because of problems with the wiring.

The A380-800 contains around 100 000 individual cables with a total length of 470 km. The total weight of the cables is 5 700 kg with an additional 1 700 kg for cable guides and suspensions. Figure 3.11 shows a photo of the crown area (above ceiling panels) demonstrating the typical wiring installation with suspension points. Figure 3.12 shows the electrical system of an A350 in CAD. In a Wireless Avionics Intra-Communication (WAIC) study, it is said that about 30 % of the A380 cable weight could be reduced if certain components had WAIC (more in Chapters 4.10 and 6). (AVSI 2015)

The empty weight of an A380-800 is 277 000 kg (Haas 2016). The weight of the cables with suspension is about 2.67 % of the total weight.

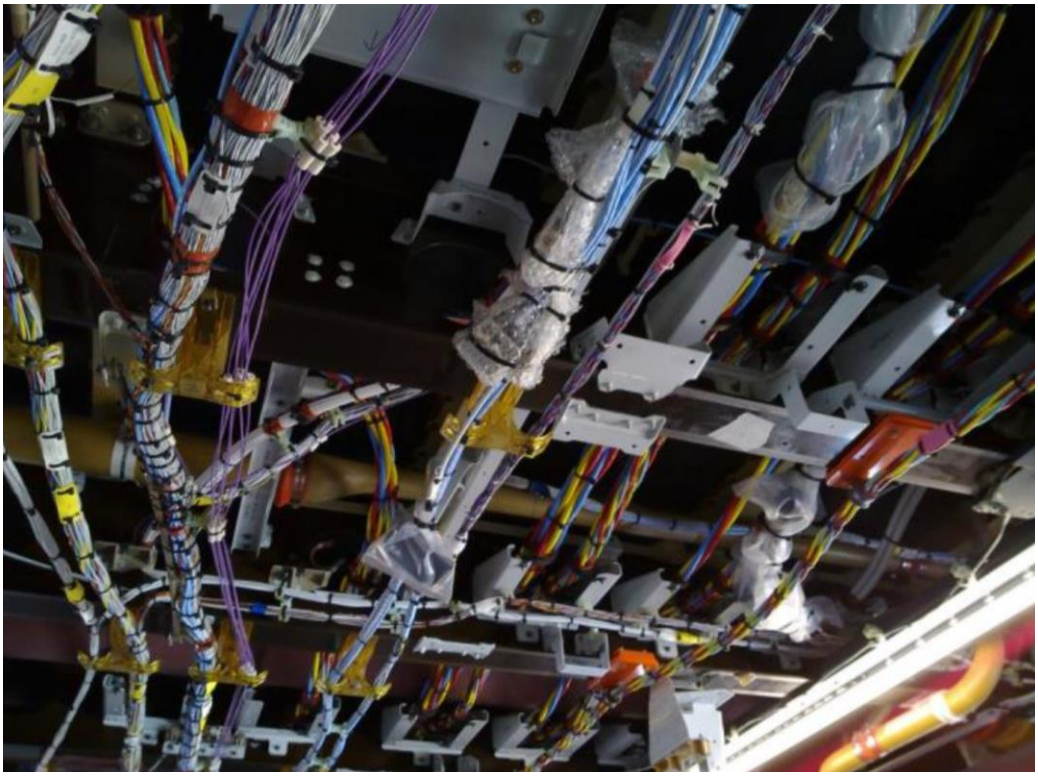


Figure 3.11 Typical wiring installation in A380 crown area (AVSI 2015)

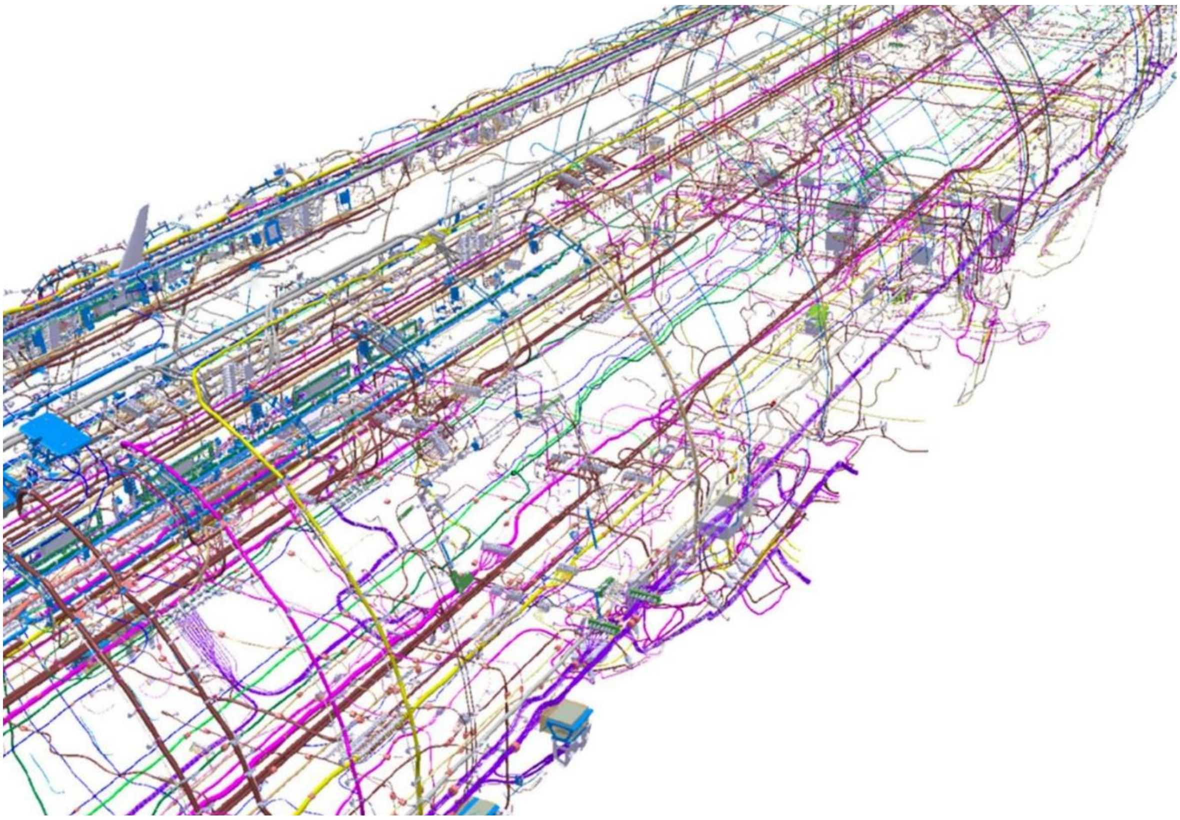


Figure 3.12 Installation of the electrical systems of the A350 (AVSI 2015)

4 Protocols

The Cambridge dictionary defines the word protocol as:

> *A computer language allowing computers that are connected to each other to communicate.*

A protocol is a set of rules that formats data for transmission and reception. It allows computers of different structures and performances to communicate with each other in a network. In order for information to be successfully transmitted, the transmitting and receiving devices must support the same protocols, which must be integrated into the electronic device by hardware, software, or both. Network protocols, such as those used in the PC for data transmission up to the display for the user, consist of cooperating protocols called protocol suite. The standard model is the Open Systems Interconnection (OSI) model defined by the International Organization for Standardisation ISO/IEC 7498 which can be seen in Figure 4.1. (Chai 2021)

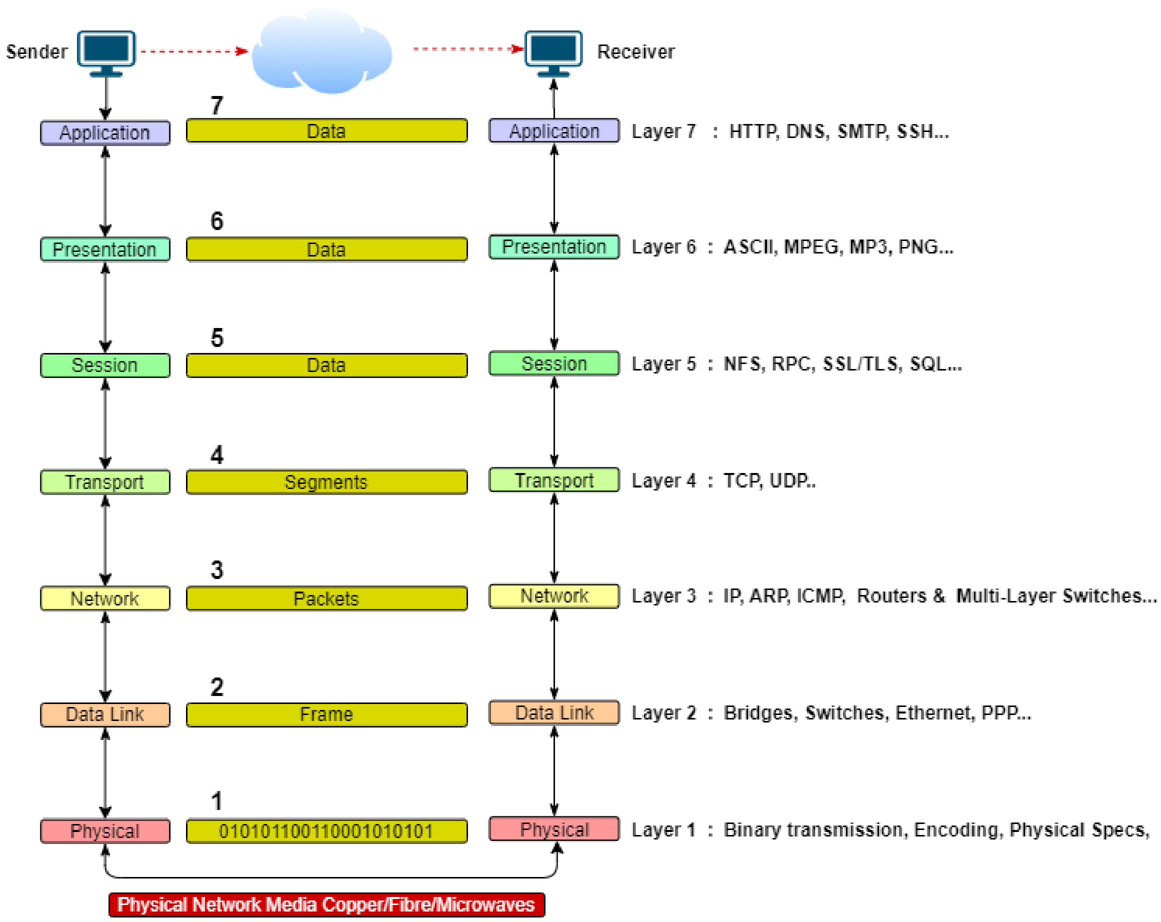


Figure 4.1 OSI model based on ISO/IEC 7498 (Mountaing 2020)

In the **first physical layer**, the raw data is transmitted between the transmitter and receiver. Here, the individual bits are translated into an electrical signal, an optical signal or a radio signal, depending on the selected transmission channel. Characteristics such as the voltage level, the timing of voltage changes, data rates, maximum transmission distances, modulation schemes, channel access methods and physical connections are also part of the layer specification. Since the physical layer also includes the interconnection of the hardware, pin arrangement, voltages, line impedance, cable specifications, signal timing, and frequencies are also included. The bit rate is defined in the physical layer where the transmission mode simplex, half-duplex or full-duplex (see definitions) can be defined. In standards such as **Bluetooth**, **Ethernet**, **USB** or **CAN**, the specifications for the physical layer are defined. (Wikipedia 2021)

The **second data link layer** defines the protocols that are responsible for creating and closing a data link. Errors that occur in the physical layer may also be detected and corrected there. The Institute of Electrical and Electronic Engineers IEEE 802 divides the layer into MAC and LLC. Medium Access Control (MAC) is responsible for the control and gives access rights for devices in the network to transmit data. Logical Link Control (LLC) is responsible for packaging and identifying network protocols, checking for data errors, and synchronizing clock rates. IEEE 802 networks such as 802.3 **Ethernet**, 802.11 **Wi-Fi** and 802.15.4 **ZigBee** operate in this layer. Point-to-Point Protocol (PPP) or the ITU-T G.hn standard also operates in this layer. The PPP can operate on several different physical layers (synchronous and asynchronous serial lines) for data transmission. (Wikipedia 2021)

The **third network layer** is responsible for sending a data package to the correct node. A node is a network access of an electronic system so that in a network many nodes can be connected. Each node has an address so that the data packages can be addressed to a specific node. In this way, the data package can be forwarded through other nodes until it arrives at the correct node. If the data packet is too large, it can be split into smaller data fragments that are then reassembled at the destination node. In the network layer are routing protocols that are responsible for managing groups and addressing assignments. This includes the Internet Protocol (**IP**), which uses rules for sending and receiving data at the IP address level, Address Resolution Protocol (**ARP**) and Internet Control Message Protocol (**ICMP**). (Wikipedia 2021)

The **fourth transport layer** is responsible for the quality of the data sequences to be transmitted from transmitter to receiver. Reliability can be controlled by flow control, segmentation/desegmentation and error control. Transport protocols such as User Datagram Protocol (**UDP**) are responsible for fast transmission and higher bandwidths without further package inspection, so that package losses, duplicates, and rewriting are accepted. The protocol is used in streaming media, real-time multiplayer games, and Voice over IP (VoIP) applications where packet loss is not a major problem. Multicast applications are also enabled by UDP. Transmission Control Protocol (**TCP**) ensures that no packages are lost with a data request, adds package numbering so that the correct sequence can be checked at the destination, is slower due to the higher effort and represents a point-to-point data connection. Intermediate solutions with advantages and disadvantages from both areas are also available, for example with TP0 (fast) to TP4 (safe). (Wikipedia 2021)

The **fifth session layer** checks connections between computers. It establishes, manages, and closes connections from local to remote applications. Checkpointing, restart, recovery or the proper ending of a session belong to this layer. Included are protocols such as Network File System (**NFS**) which allows access to files over a network, Remote Procedure Call (**RPC**) which can be used in programs to request services from another program located on a computer on another network (without having to understand the details of the network), Structured Query Language (**SQL**) which is a transaction protocol that records database changes and allows recovery in case of system failure, Secure Sockets Layer (**SLL**) and Transport Layer Security (**TLS**) for providing an encrypted Internet connection. (Wikipedia 2021)

The **sixth representation layer** is also called the syntax layer. Here the data is converted from the network format into the application format. Basic Abstract Syntax Notation One (ASN.1) encoding rules are used, for example, to convert an 8-bit Extended Binary Coded Decimal Interchange Code (EBCDIC) file to an American Standard Code for Information Interchange (**ASCII**) encoded text file. Data structures to or from Extensible Markup Language (**XLM**) are also enabled. User-related presentation file types include Moving Picture Experts Group (**MPEG**) for encoding media such as audio, video, and graphics; the **MP3** encoding format for audio files; and Portable Network Graphics (**PNG**) for compressing and presenting images. (Wikipedia 2021)

The **seventh application layer** is the layer closest to the user. The application layer and the user interact with the software application. Functions include identification of communication partners, determination of resource availability, and synchronization of communication. For example, a website might have two application entities that use Hypertext Transfer Protocol (**HTTP**) to communicate with the user and a database protocol to collect data for a remote database. The layer also includes protocols such as Domain Name System (**DNS**) as a decentralized name system for identifying other computers, services and resources; Simple Mail Transfer Protocol (**SMTP**) for receiving and sending mail; and Secure Shell (**SSH**) for secure operation of network services such as logins and remote command lines over an unsecured network. (Wikipedia 2021)

In practice, many protocols handle tasks from multiple layers. Therefore, the protocol suites look very different depending on the application. In the TCP/IP protocol suite, there is a four-layer division that packs OSI layers one and two together, takes over the third and fourth layers and packs the fifth, sixth and seventh layers together again. In this way, a connection from the application to the physical connection could take place from the protocols SMPT to TCP to IP over Ethernet. (Novell 2009)

When selecting a suitable data BUS in an aircraft (BUS topology see Figure 4.2), several points must be considered: Reliability, durability, long-term availability, number of nodes in the system (number of components), bandwidths to be transmitted, upgradeability, type of aircraft (civil or military), and price (hardware costs, software costs, maintenance costs). The hardware components must also be able to withstand all influences such as shock, vibration, temperature and electromagnetic interference during the operation of the aircraft. (Stock 2011, Chapter 1).

There are many different systems used in aircraft from different manufacturers. ARINC 429 is used in larger aircraft as an integrated BUS to connect guidance, altitude, altitude reference, flight management and more to one network. An ARINC 429 interface board is priced at over $3000 per piece. Due to the higher implementation effort in the aircraft and the high price, the data BUS is not suitable for smaller aircraft (more about the protocol in Chapter 4.5, *ARINC 429*) (Ueidaq 2021). ARINC 629 and ARINC 664 or AFDX (Avionics Full-Duplex Switched Ethernet) are used as the main data link data buses in large aircraft. The cost per node is over 500 dollars (more on the protocol in Chapter 4.6.1, *ARINC 629*). MIL-STD-1553 and MIL-STD-1773 data BUS are used in military aircraft. The cost per node is over 800 dollars (more in Chapter 4.6.3). The IEEE 1394b (FireWire) or AS5643 (Aerospace Standard) is used in unmanned vehicles and military aircraft (more about the protocol in Chapter 4.6.4). (Janu 2014 and Stock 2011, Chapter 1)

With the choice of a suitable protocol, there are different network topologies that a protocol supports. CAN, TTP/C, Flexray, ASCB-D and MIL-STD protocols have a linear/bus topology. ARINC 664 / AFDX is built in a star topology. AS5643 is built in a ring topology. Protocols can also support multiple topologies (see topologies in figure 4.2). (Stock 2011, Chapter 1)

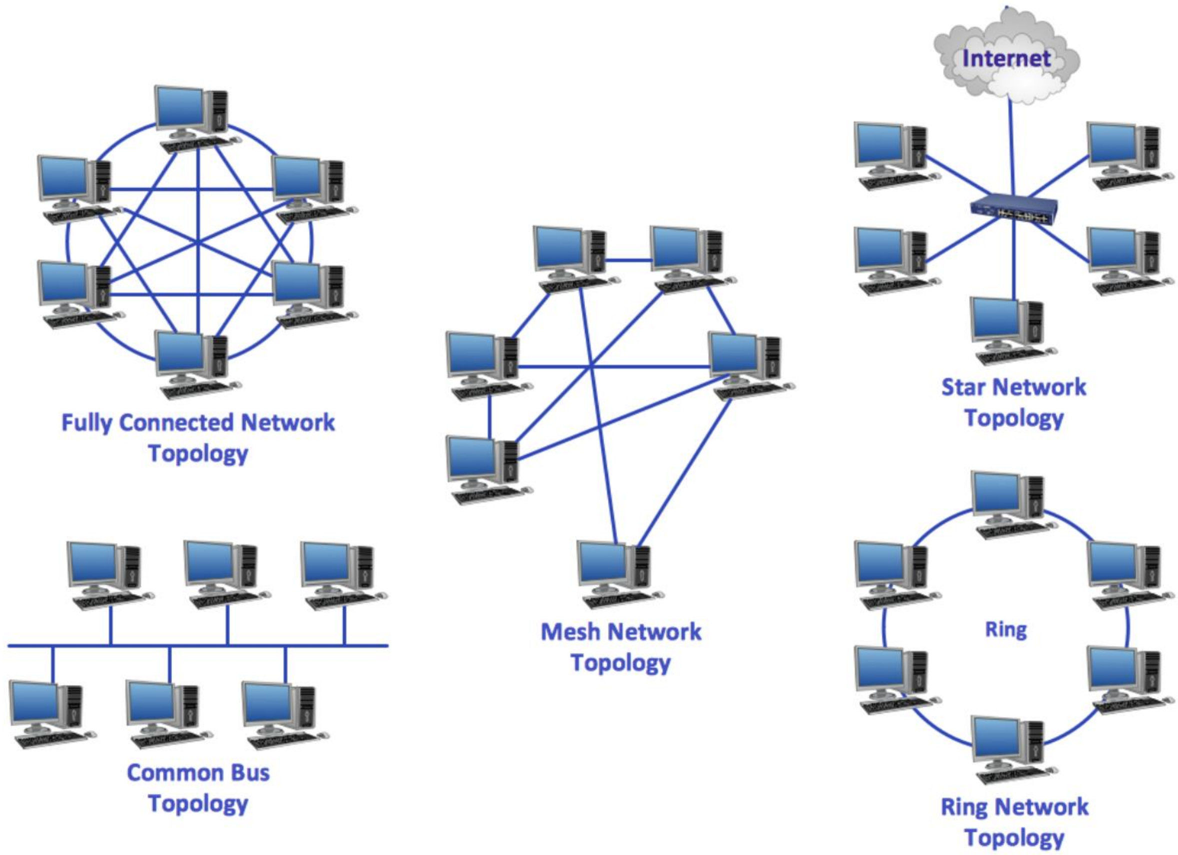


Figure 4.2 Various types of network topologies (Alexander 2017)

4.1 Controller Area Network (CAN)

Controller Area Network (CAN) systems are mainly used to save wires. The CAN reduces the required cable to 2 wires where the information is sent in a serialized format. The use of a CAN system also means weight savings and easier installation.

The use of digital avionics is a standard feature of modern aircraft. In large aircraft, various devices are connected using the Integrated Modular Avionics (IMA) concept, which gives relative flexibility in the choice of devices. The ARINC 629 protocol is often used, which enables a high data exchange rate (more in Chapter 4.6.1, *ARINC 629*). Unfortunately, the system is more complicated and the installation and maintenance costs are too high to be used in small aircraft. The CAN data bus configuration is a mature and suitable choice as it is already widely used in the industry and there is a lot of cheap hardware available to get a CAN BUS running. (Prasad 2014)

CAN was developed by Bosch in 1985 for the automotive industry. In contrast to USB or Ethernet, CAN does not send large blocks from point-to-point, meaning from node A to node B, but in a CAN network, short messages are sent into the BUS system which can be received by each node. The connection of different devices is therefore done through the bus-line nodes which are connected to the corresponding electrical systems (see Figure 4.3). (Corrigan 2016)

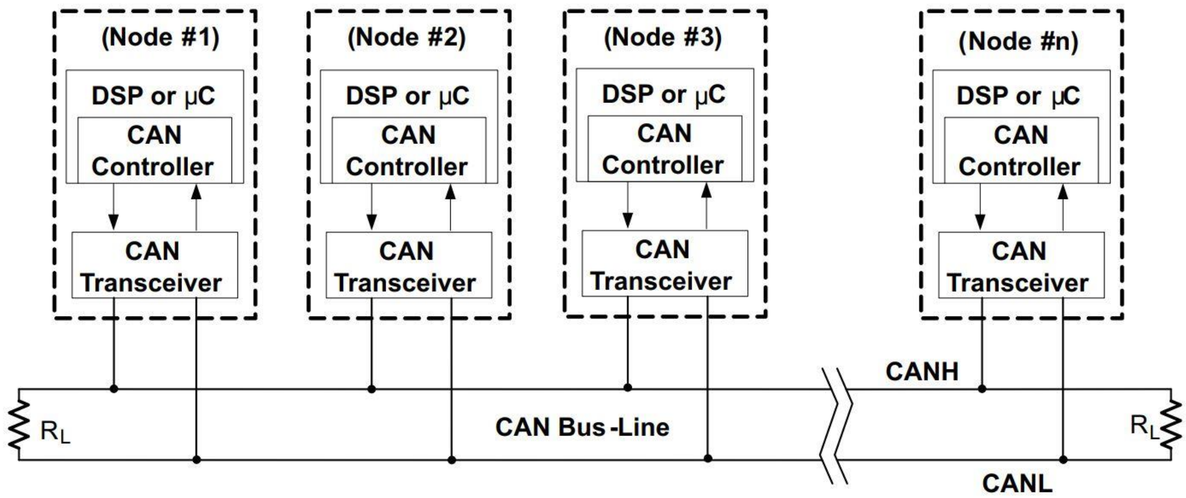


Figure 4.3 Structure of a CAN BUS (Corrigan 2016, Chapter 5)

CAN was standardized by ISO-11898 which defines CAN on the first two layers according to the OSI model, the physical layer and the data link layer (see Figure 4.1). The standard architecture according to ISO 11898 provides on the application layer a microcontroller that drives an integrated or separate CAN controller, then a CAN transceiver which then enters the BUS line via CANH (high) and CANL (low) (see Figure 4.3). The CAN protocol is also supported by the international users and manufacturers group under the name CAN in Automation (CiA). There are many CAN-based variations from different companies in different industries, such as

CAN Kingdom by KVASER (Fredriksson 1995) and DeviceNet by Rockwell Automation. Applications include cars, trucks, motorcycles, snowmobiles, trains, buses, aircraft, agricultural vehicles, construction machinery, mining equipment, marine vehicles and machine networking with automation. (Corrigan 2016)

CAN is a CSMA with CD+AMP. CSMA stands for Carrier-Sense, Multiple Access, which means that each node only enters the BUS if no other nodes have entered the BUS. If the BUS is empty, several nodes can try to enter the BUS at the same time. CD+AMP stands for Collision Detection and Arbitration on Message Priority and means that if several nodes enter the BUS and a collision occurs, the message with the higher priority is sent while the other messages have to wait. Each message is assigned an 11-bit identifier, which allows 2048 different message identifiers to be defined. The highest priority has the lowest number (0) up to the highest number (2048) with the lowest priority. Later, an extended CAN was introduced, which has a total of 29-bit identifiers, allowing 537 million identifiers to be defined (see Figure 4.4). (Corrigan 2016)

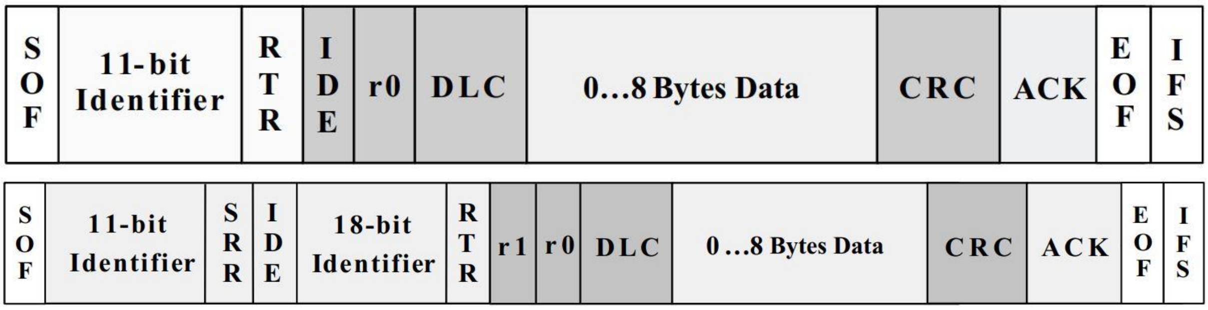


Figure 4.4 Standard CAN (top) and extendet CAN (bottom) (Corrigan 2016, Chapter 3)

The structure of a **normal CAN** message is: start of frame (SOF), 11-bit identifier, remote transmission request (RTR), identifier extension (IDE), reserved bit r0, data length code (DLC), 0 to 8 bytes of the message itself, cyclic redundancy check (CRC), acknowledges (ACK), end-of-frame (EOF), inter-frame space (IFS). The structure of an **extendet CAN** message is: SOF, 11-bit identifier, substitute remote request (SRR), IDE, additional reserved bit r1, reserved bit r0, DLC, 0 to 8 bytes of the actual message, CRC, ACK, EOF, IFS (see Figure 4.4). (Corrigan 2016, Chapter 3)

A basic feature of the CAN is the opposite logical state between the input and the CAN and the output and the CAN. The logical high state is normally a 1 while the low state is a 0. In the CAN it is the opposite. Figure 4.5 shows the driver input with a 1, 0, 1, 0, 1 which also arrives in the receiver output. In the CAN bus, however, the message is transmitted in such a way that the Recessive Low state is a 1 and the Dominant High state is a 0 (Corrigan 2016).

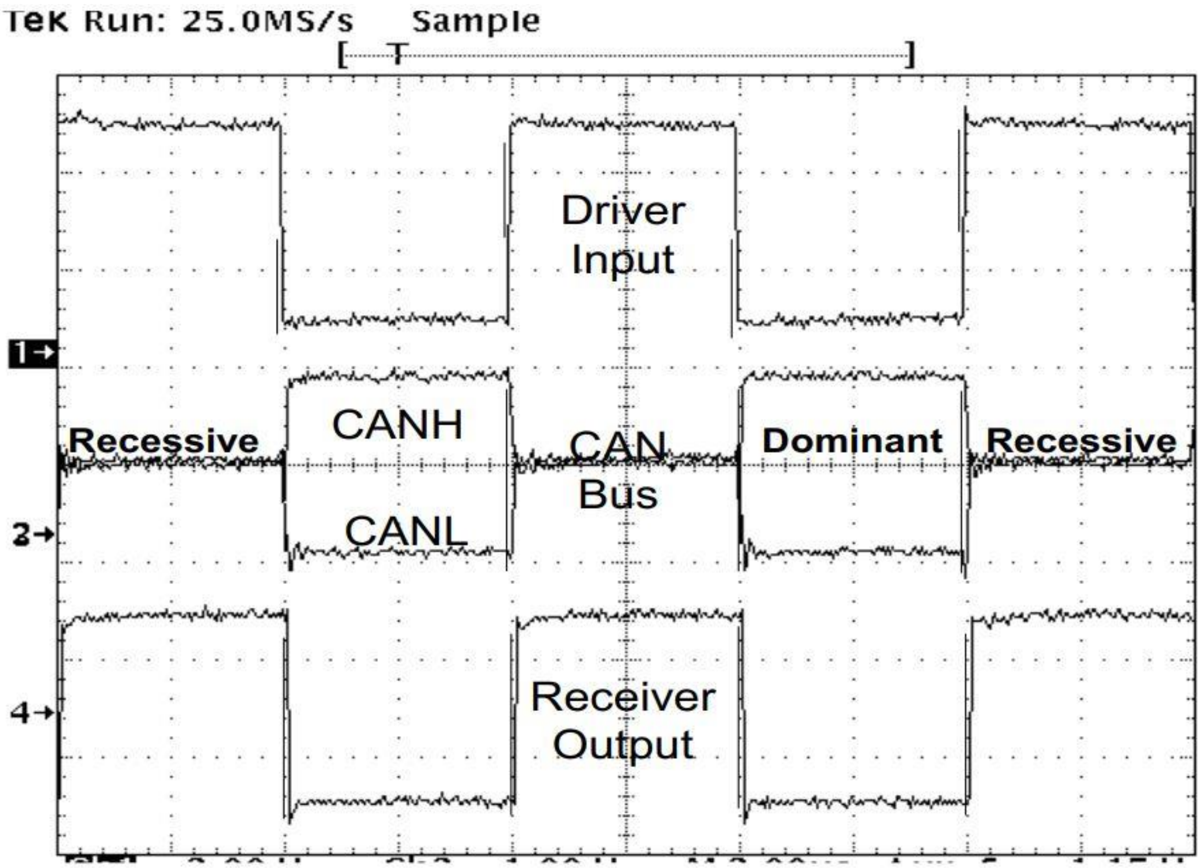


Figure 4.5 CAN dominant and recessive BUS states (Corrigan 2016, Chapter 5)

If two nodes try to enter the BUS at the same time, access is granted to the node that is longer dominant, which means it has more zeros and therefore a higher priority (nondestructive, bit-wise arbitration). The dominant node continues with its message because it sent a 0 last (dominant) while the other node sent a 1 (recessive) and waits until the BUS is free for another attempt. Figure 4.5 also shows very well that in the CAN BUS a dominant 0 can always overwrite a recessive 1. Figure 4.6 shows how 3 nodes use the CAN BUS to transmit information. Node A transmits its message after which Node B and C acknowledge the correct reception of the message. Node B and node C then start transmitting the message and node C wins arbitration because it has more zeros and sends a bit longer dominant. It ends the message transmission which node A and node B confirm. Now, node B is the last to transmit its message with the lowest priority. As already shown in Figure 4.3, this process can be used to implement a large number of nodes in a CAN BUS system. (Corrigan 2016)

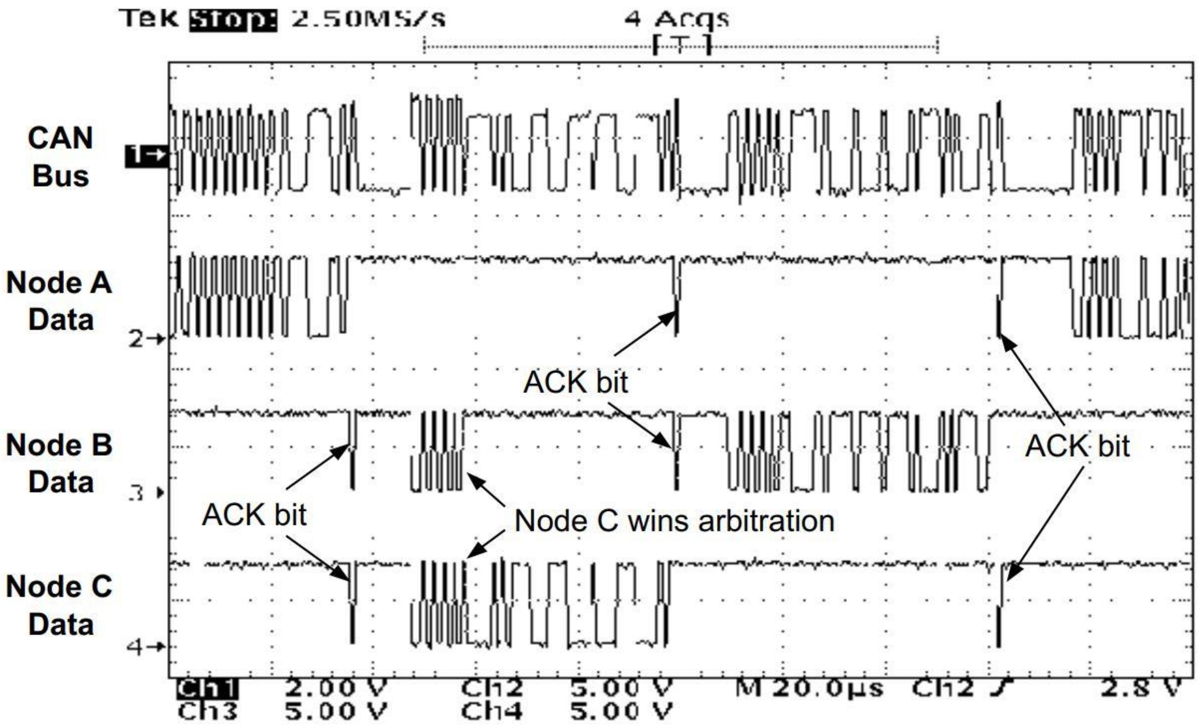


Figure 4.6 CAN BUS traffic with 3 nodes (Corrigan 2016, Chapter 5)

In the dimensioning and implementation of the CAN protocol, there are of course many things to consider. For example, a 120 ohm resistor is usually packed between the CANH and the CANL, which can also be split with a capacitor. Otherwise high common mode voltages could occur during periods of inactivity. Also high frequency noise of the BUS lines is filtered by connecting a capacitor to the ground. Figure 4.7 shows the hardware scheme. Depending on the selected signaling rate, a typical value of the capacitor for a high-speed CAN is $C_L = 4.7$ µF. The two 60 Ω resistors should also be as equal as possible ($\pm 1\%$) so that the effective magnetic noise immunity is not reduced. With a twisted pair or shielded twisted pair cable and the described technique, good electromagnetic compatibility of the network is created. (Corrigan 2016)

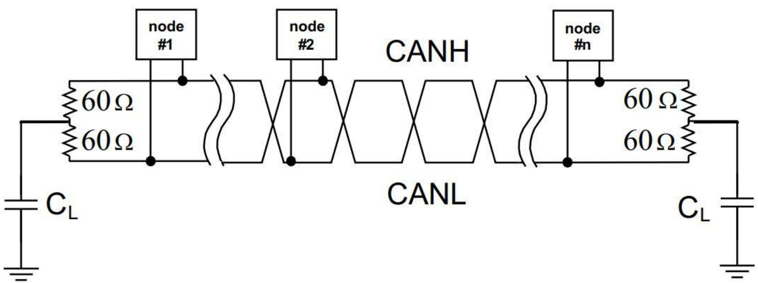


Figure 4.7 CAN BUS wiring scheme (Corrigan 2016, Chapter 5.1.13)

For the use of CAN BUS on board an aircraft, the system must be highly fail-safe and properly designed. In a paper published in the International Refereed Journal of Engineering and Science (IRJES), the use of CAN BUS for smaller aircraft is investigated. The study shows that a redundant architecture, which can be located in the firmware, can increase the reliability of the system. The safety requirements for the reliability and stability of the CAN BUS system are therefore sufficient for use in the aircraft. The CAN standard defines **five different error detection methods** to keep the system error-free. In **bit monitoring**, each transmitter on the CAN BUS compares the bit currently being transmitted with the bit originally transmitted; if a discrepancy is detected, a bit error is signaled. In **bit stuffing**, no more than five consecutive bits may be sent from the same source, otherwise, a stuff error is signaled. **Frame check** always checks the frame of each message with the logical sequence, for example, at the start of the frame a certain SOF level must be maintained, otherwise, a frame error is signaled. With the **acknowledgement check**, all nodes connected to the BUS must acknowledge each correctly received message in the ACK slot. If the sender does not receive an ACK from any receiver, the ACK error is signaled. **The cyclic redundancy check** is a 15-bit checksum in each message. If there are discrepancies between the received and calculated checksums, a CRC error occurs. If a fault is found, a backup strategy must be available to keep the CAN BUS running. In very important applications a completely redundant backup BUS system could also be available to switch to. (Prasad 2014, page 25)

A user-related CAN presentation with further design tips and tricks is available from Boys 2012.

4.2 CANaerospace

A modification of the CAN protocol for aviation was developed by Michael Stock in 1998 with the name **CANaerospace**. It was standardized around 2000 at NASA as **AGATE** data bus. CANaerospace is used since then in several aircraft and in the microlight engine, Rotax 912 iS for data transmission between the engine and the Engine Management System (EMS). In the SOPHIA research aircraft, CANaerospace is used to control the telescope. The CANaerospace protocol is also used in other systems inside the aircraft. (Stock 2005)

The AGATE Program was established in 1994 and completed in 2001. Its goal was to create a Small Aviation Transportation System (SATS) as an alternative to short-range automotive trips for personal and business purposes. Business people could visit several factories located far apart in just one day by reducing travel time. The project aims to make flying easier for everyone and is also intended for medical, public safety and recreational purposes, among others. With AGATE, market growth for intercity transport with small aircraft should be possible. Safety enhancements for single-pilot operations were also a priority. Figure 4.8 (affordable alternative transportation) shows ideas and configurations of possible small aircraft. Also, very innovative ideas for cockpit design and one-handed control to simplify the system for the pilot and make it safer. The system uses a computer to receive and display the latest information from weather, flight route navigation, map display and information on restricted areas, terrain hazards, airports and airspace conditions. (NASA 1996)

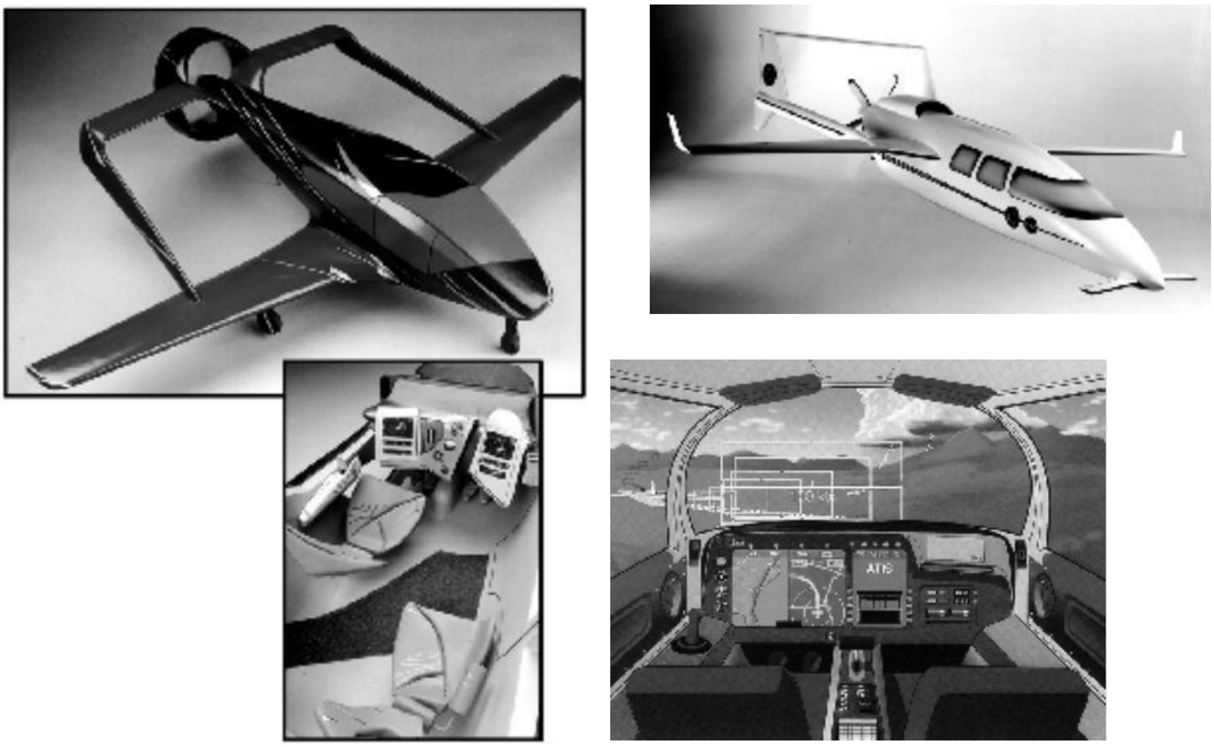
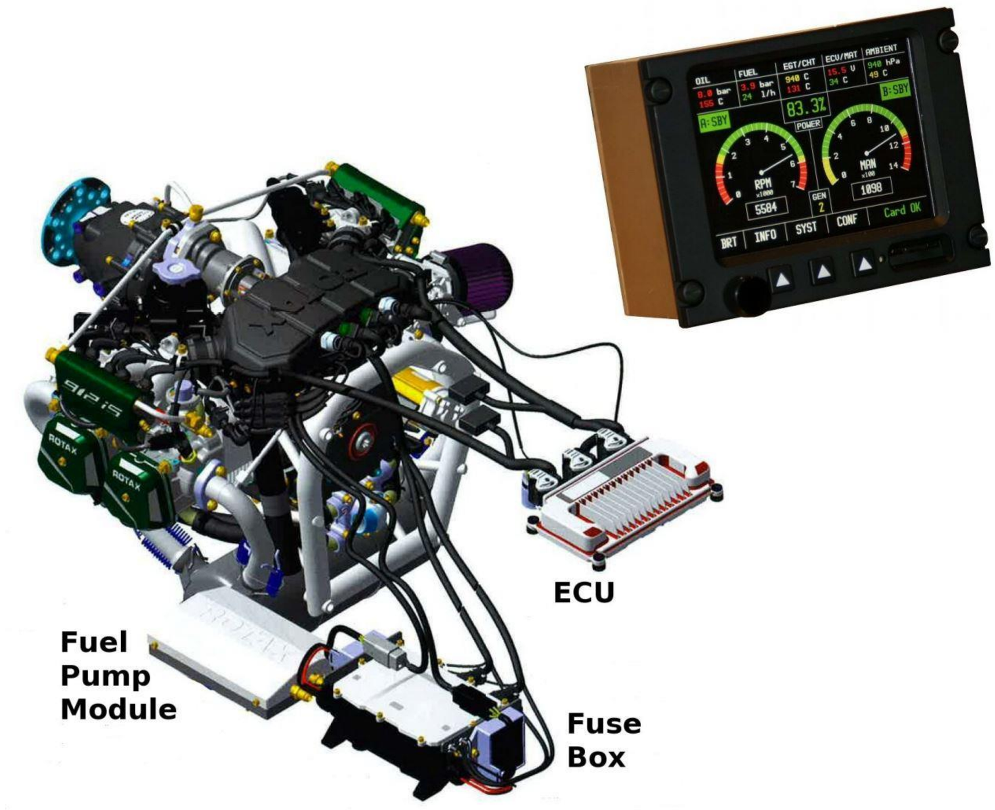


Figure 4.8 AGATE – affordable alternative transportation (NASA 1996)

Microlight aircraft from many manufacturers use the Rotax 912 iS engine because it is powerful, reliable and fuel-efficient (Rotax 2013). It transmits data from the engine with a double redundant CANaerospace BUS. This means that 4 cables go from the Engine Controller Unit (ECU) to the Engine Management Unit (EMU). The EMU displays engine and sensor data for the pilot. This is primarily engine speed, intake manifold pressure, and other temperatures and pressures in the engine. Secondary information such as the status of sensors and the status of the engine itself is also routed with the BUS to the EMU where it can be stored for maintenance or the recording of engine data with time stamps and GPS data (see figure 4.9 the connected ECU at the engine which transmits the data to the EMU in the aircraft with the redundant CAN BUS). By recording all this data with the sensor values from the motor, it is possible to make early predictions of problems or find faults better. The EMU also provides engine-specific information such as cold start procedures to guide the pilot. Recording and monitoring provides information over the entire life of the engine and comes from the world of jet engines. It can now increasingly be used for piston engines as well. (Stock 2012)



Engine Drawing: © BRP-Powertrain

Figure 4.9 ROTAX 912 iS engine with ECU and EMU (Stock 2011, Chapter 1: *CAN*)

The CANaerospace has a configurable data rate from 5 kBit/s to 1 MBit/s with an effective data rate of max 576 kBit/s (≤ 40 m BUS length), which is sufficient for most real-time control systems in small aircraft. (Stock 2005)

It should be noted that the highest possible data rate is strongly dependent on the bus length, which is shown in Figure 4.10. For a 100% loaded BUS, a BUS length of 40 meters is still possible at a data rate of 1 MBit/s. In small aircraft, however, the BUS lengths are usually less than 10 meters. (Stock 2011, Chapter 2: *CAN Physical Layer*)

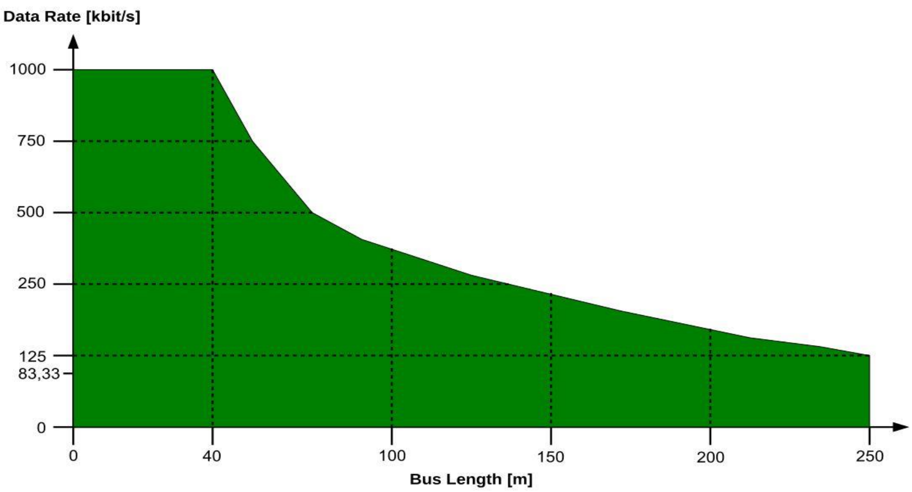


Figure 4.10 CAN data rate vs. BUS length (Stock 2011, Chapter 2)

Another factor limiting the transmission speed is the number of nodes in the system. The recommendation is to stay below 32 nodes at 1 Mbit/s, below 35 nodes at 500 kbit/s and below 40 nodes at 250 kbit/s. The reason for this is that each node and each component brings a certain resistance, capacity and inductivity into the BUS system. Depending on the quality of the CAN transceiver, the effects are stronger and weaker, so you have to make own tests with the chosen components. (Stock 2011, Chapter 2)

In a quality analysis of the CANaerospace protocol at the University of Defence in the Czech Republic, a CANaerospace BUS with 7 nodes was examined at various speeds. At a baud rate of 125 kbit/s, communication was unproblematic even at a BUS length of 40 m. At 500 kbit/s, communication quality decreased. Some messages were sent too often into the BUS and some were sent and received not enough. At 1 Mbit/s only one node in the BUS was examined. At a small distance of the node transceiver to the receiver, communication could take place, but at a higher distance more than 2.83 m, the communication quality became very bad so that communication was no longer possible. However, it must be said that for the test flat cables were used which were not even shielded. In general, the communication behavior of the designed system is very different at different baud rates and different time schedules. Therefore a detailed analysis must be made for the suitability of a system. (Janu 2014)

In a later paper, the author has also developed an application that can help during a design of communication algorithm of any avionic system based on MATLAB. In an avionic system, it is necessary to enter the BUS at predefined time intervals. To develop these time schedules for different nodes a cycle matrix can be used. This way all nodes have a certain time window to send messages into the BUS. In general, the CANaerospace specification recommends for the design 80 % BUS utilization for synchronous messages to leave room for asynchronous messages, for the control messages of the BUS communication and for emergency messages. (Janu 2016)

CAN hardware is produced by many manufacturers such as Motorola, Intel, Philips, Infineon, Toshiba, NEC and Texas Instruments with different performance of the microcontrollers, different connection types and quality. Since many BUS systems are used in industry, such as VME, PCI/PMC, ISA/EISA, SBUS, PC104, IP and others, there are many different BUS boards to choose from (see CAN BUS board in Figure 4.11). There are also many CAN BUS boards for Raspberry Pi and Arduino which is a good interface for smaller projects and test phases. A DB-9 connector (RS-232), which has been a recommended standard for serial data transmission since 1960, is often used to wire components (Wikipedia 2022). (Stock 2005)

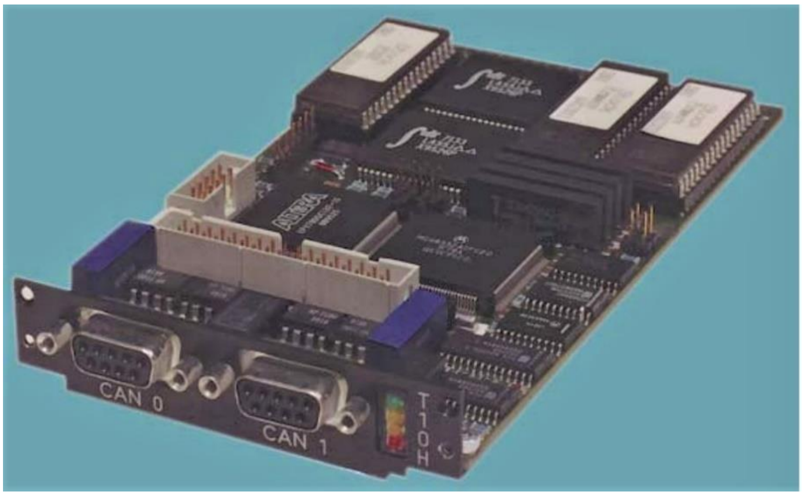


Figure 4.11 CAN board (Stock 2005, page 4)

In an aircraft BUS application, a DB-9 connector can be used to connect all the electronic components. Figure 4.12 shows a CANaerospace pin assignment proposal where pin 1 (VCC) and pin 5 (GND) are used to supply power to the respective device and pin 2 (CAN low) and pin 7 (CAN high) are used to control the CAN BUS. Pin 4, 6 and 9 are used for maintenance (TX, RX, GND) and pin 3 and 8 are unused. In the case of a redundant BUS system, these pins could be used for CAN low 2 and CAN high 2. (Stock 2005, page 12)

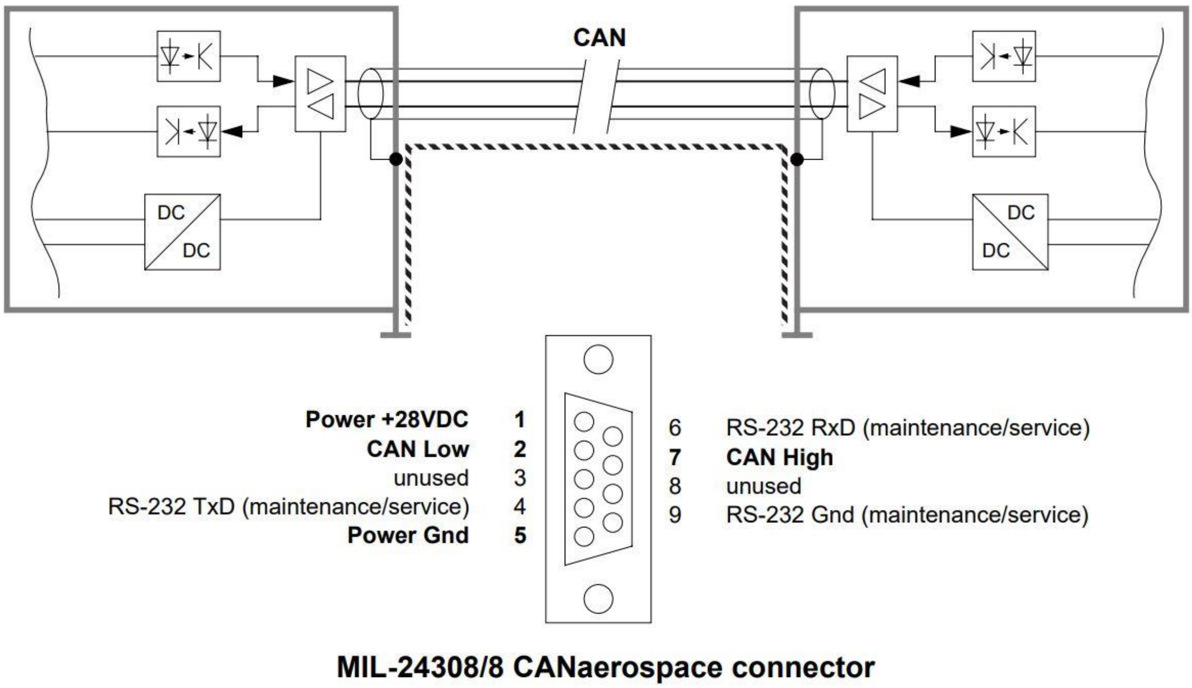


Figure 4.12 CANaerospace proposal for the physical interface (Stock 2005, page 12)

As already described in Chapter 4.1 *CAN*, CANaerospace works on the same principle. In the CANaerospace BUS, the identifiers have been divided into different areas with a general definition of priority (see Table 4.1).

Table 4.1 CANaerospace identifiers with priority (Stock 2005, page 10)

Desciption and priority		Identifier range	Number of identifiers
Emergency Event Data	highest	0 – 127	128 data objects
High Priority Node Service Data		128 – 199	72 data objects or 36 com. channels
High Priority User-Defined Data		200 – 299	100 data objects
Normal Operation Data		300 – 1799	1500 data objects
Low Priority User-Defined Data		1800 – 1899	100 data objects
Debug Service Data		1900 – 1999	100 data objects
Low Priority Node Service Data	lowest	2000 – 2031	32 data objects or 16 com. channels

For example, in identifier 317 is calibrated airspeed, on 321 heading angle, on 401 roll control position, on 1070 radio hight and on 1205 lateral center of gravity (for further CANaerospace identifier definition see Appendix C). For the data transmission of any CAN message, the probability of undetected data corruption is about $1 \cdot 10^{-13}$ per transmission, which is very low. (Stock 2005, page 3 and 11)

CANaerospace uses the same identifier structure as shown in Figure 4.4, with the change that 4 bytes of the message block are replaced by a **CANaerospace message header**, leaving only 4 message bytes for message transmission. In **byte 0** (node ID) a backup node can be addressed in case of node failure (mode change in redundancy management). In **byte 1** (data type) the data type of the message is specified since (backup) devices from different manufacturers can use different data types even if they perform an identical function. By specifying the data type with each message, a system configuration is automatically defined. In **byte 2** (service code) the status and thus the validity of each message is transmitted. The unit of the value to be transmitted can also be transmitted. In **byte 3** (message code) the messages are numbered so that the sequence and the absence of messages can be checked. In this way it can also be determined whether a transmitting unit is working properly (see Figure 4.13). (Stock 2005, page 9)

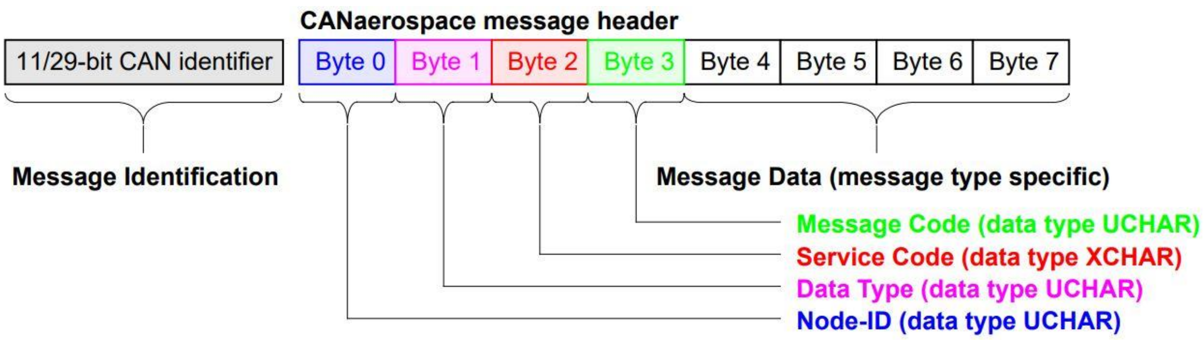


Figure 4.13 CANaerospace basic message format (Stock 2005, page 9)

There are many other parameters that must be considered when designing an aircraft system with CANaerospace. These include bandwidth management, redundancy management, selection of the right cables and connectors with shields and splice design, and others such as Quality increase by galvanic isolation of CAN transceiver and CAN controller in all nodes. (Stock 2011)

Bandwidth management is needed when there are many nodes in the BUS. The higher priority messages could block the lower priority messages over a longer period of time which would lead to excessive latency for the lower priority messages. Bandwidth management must also prevent peak loads scenarios. One concept is Time-Triggered BUS Scheduling (**TTBS**). In the quality analysis of Janu 2014, bandwidth management is performed using a cycle matrix. Keywords are time-triggered communication via CAN (**TTCAN**) and time division multiple access (**TDMA**). (Stock 2011) (Janu 2014)

Redundancy management is an important topic if avionic systems have to be designed fail-safe. For example, it is possible to enter **several redundant CAN BUS with one node** type or to enter a **CAN BUS with several redundant nodes** of the same type. A mixed solution with several redundant nodes and CAN BUS represents the best degree of redundancy but also the greatest effort (**cross-strapping**). Also in software design, the verification of redundant messages is important because sending a message with the same priority would lead to an uncovered error. This is the reason why a message code is sent with the message. (Stock 2011)

CAN BUS failures like a cable break (**open-circuit**) must be planned as well as a **short-circuit** in the BUS line or in the CAN transceiver or another component on the BUS. **Software errors** of any kind must also be considered with backup plans to avoid a total system failure in the air. In more complex BUS systems with longer distances, bit timing must also be considered. Cables, transceivers and other components increase the capacity of the system, making the system react more sluggish to current pulses. This also leads to the fact that the time t_{BitA} of the recessive 1 becomes longer to the time t_{BitB} of the dominant 0 (see figure 4.14). A CAN transceiver can in extreme cases also evaluate the data incorrectly. (Stock 2011, Chapter 3-4)

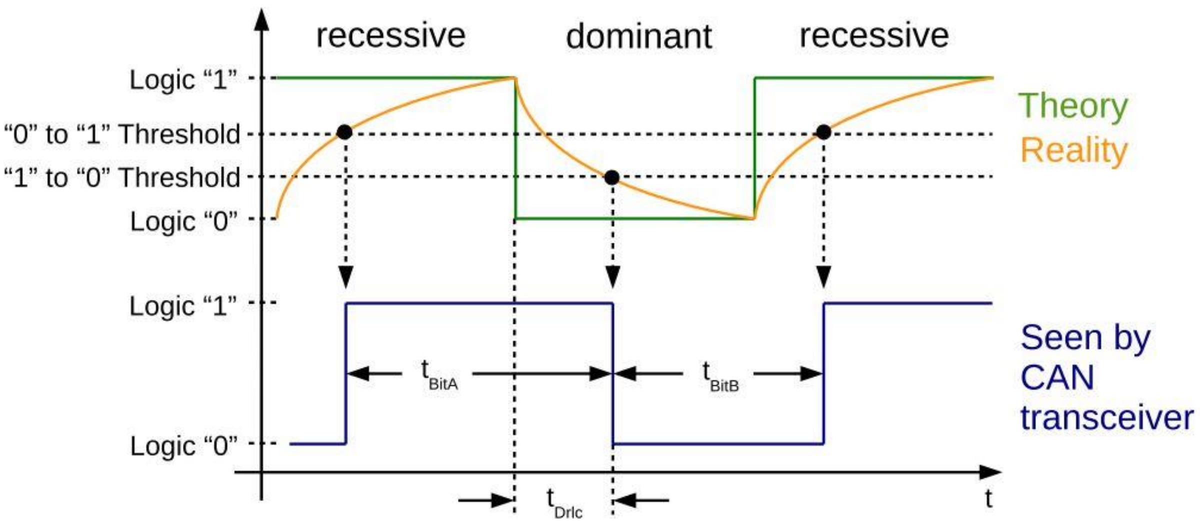


Figure 4.14 Influence of CAN Transceiver and Cable on bit timing (Stock 2011, Chapter 3)

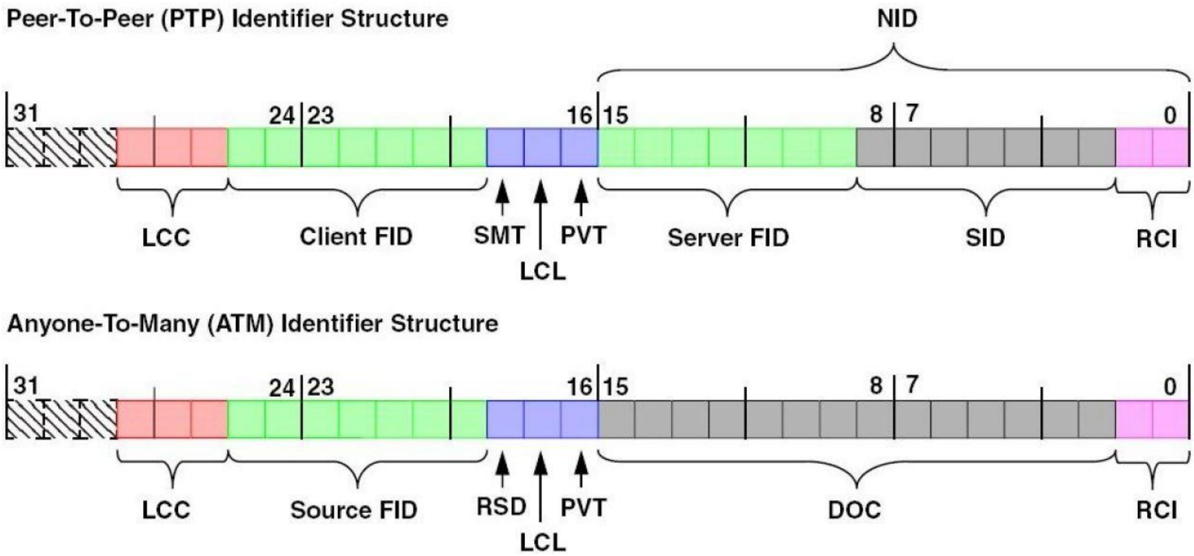
The CANaerospace demonstration software written in C can be found in Appendix B. The CANaerospace interface definition with all identifiers defined so far can be found in Appendix C. It is pointed out that the description of the CANaerospace protocol in this work together with the demonstration software cannot be used to develop a system for use in an aircraft. However, it is intended to provide the opportunity to get deeper into the subject matter in order to carry out test runs with CANaerospace. For the installation of a CAN BUS system in an aircraft, every detail must be taken into account in order to be able to successfully write software and ensure safe operation (more on the decision and design of a protocol in Chapter 4.9).

The open-source license and low chip costs on which CANaerospace can be deployed make the data BUS very attractive. For example, CAN is used as a throttle-by-wire data BUS in the Strato-2C high-altitude research aircraft or as an environmental control system and data bus for water waste management in the Airbus A340-600. In the Fairchild-Dornier 728 aircraft, the CAN BUS is used in the primary and secondary flight control actuators, as a secondary power supply system, for intercommunication and in the smoke detection system. In the A380, the CAN BUS is also used in subsystems such as the environmental control system, in the Multi-Function Display (MFD), and to control the power supply (see CAN protocols CANopen, ARINC 825, 826 and 812 in the next chapters). (Stock 2005)

CANaerospace was also selected as a suitable transmission protocol for networking the components of a Small Unmanned Aerial Vehicle (SUAV) SAFRAN flight system architecture. The possibility of optimizing the protocol for the application was used, but it was also pointed out that the numerous degrees of freedom of the CAN protocol requires a high degree of care. The additional identifiers required for SUAV have been extended in the user-defined area in order to maintain the complete standard assignment of CANaerospace. These are for example Ultrasonic high on ID 1500, distance to obstacle horizontal on ID 1501, RPM and temperatures of the motors from ID 1560 to 1567 and the status of the batteries at ID 1571-1574. If a CAN-BUS is used in the construction of drones and other SUAV, it is a good option to continue using the identifiers defined in the paper (see Appendix B *Identifier Distribution – NOD* of the paper by Horst 2018). (Horst 2018)

4.3 ARINC 825

On behalf of Airbus and Boing, a CAN Technical Working Group is formed by the Airlines Electronic Engineering Committee (AEEC) to define the ARINC 825 standard. Members of the CAN Working Group were from Airbus, Boing, Rockwell Collins, GE Aerospace and Stock Flight Systems. Due to different networks on board of an aircraft, the integration of different electrical components in the aircraft was associated with considerable problems, which is why Airbus and Boing formed a cooperation. ARINC 825 (Aeronautical Radio Incorporated) was published in November 2007. In large commercial air transport aircraft, ARINC 825 is used as a sub-system to ARINC 664 / AFDX. So sensors, actuators and other types of avionic devices which have low to medium data transmission are connected to the CAN which in turn is connected to an interface on the AFDX system. ARINC 825 is very similar to CANaerospace, but uses a 29-bit CAN identifier by default. Since CAN only defines the first two ISO/OSI layers, ARINC 825 also defines ISO/OSI layers 3, 4 and 6 to enable logical communication functions (peer-to-peer (PTP), anyone-to-many (ATM) and station addressing). Figure 4.15 shows the identifier structure of the ARINC 825 in PTP and ATM mode. (Stock 2009)



Field	Description
LCC	Logical Communication Channel
FID	Aircraft Function ID
RSD	Reserved
SMT	Service Message Type
LCL	Local Bus Only

Field	Description
PVT	Private Data
SID	Server ID
RCI	Redundancy Channel ID
NID	Node ID
DOC	Data Object Code

Figure 4.15 ARINC 825 CAN Identifier Structure (Stock 2011)

Figure 4.16 shows the communication channel assignment of ARINC 825 which also has the CAN typical message priority.

Channel Number	Channel Acronym	Communication Type	Description	LCC Bits	Message Priority
0	EEC	ATM	Exception Event Channel	000	Highest
1			Reserved	001	
2	NOC	ATM	Normal Operation Channel	010	
3			Reserved	011	
4	NSC	PTP	Node Service Channel	100	
5	UDC	ATM/PTP	User-defined Channel	101	
6	TMC	PTP	Test and Maintenance Channel	110	
7	FMC	ATM/PTP	CAN Base Frame Migration Channel	111	Lowest

Figure 4.16 ARINC 825 logical communication channel assignment (Stock 2009)

The Time-Triggered BUS Scheduling (TTBS) bandwidth management concept already developed at CANaerospace was also applied in the ARINC 825 standard (see Figure 4.17). (Stock 2009)

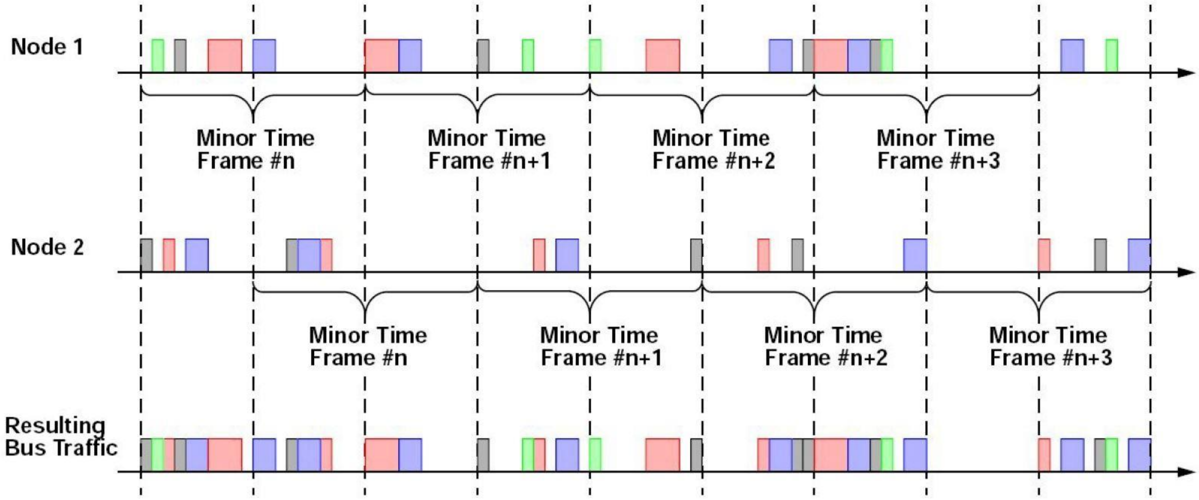


Figure 4.17 CAN message distribution using TTBS, predictable bus traffic at 50 % bus load (Stock 2011)

ARINC 825 is now used in new aircraft. The first aircraft with this protocol onboard was the Airbus A350. Various networks are installed which communicate with each other via gateways. Since bandwidths and communication types are very different, the ARINC 825 specifications provide a gateway model with substantial information about protocol conversion, bandwidth management, data buffering and fault isolation (Stock 2009).

Another good description of ARINC 825 can be found in the paper by Knueppel 2012, which also describes CAN BUS in general in aircraft. The smoke detection system of the Airbus A318 was one of the first CAN BUS controlled systems based on CANopen (see Figure 4.18). Since the concept was well established, it was used again in the Airbus A380. Several thousand flight hours proved that the system worked well and good experience was gained, which was used in the development of ARINC 825, which is now used in the Airbus A350. (Knueppel 2012)

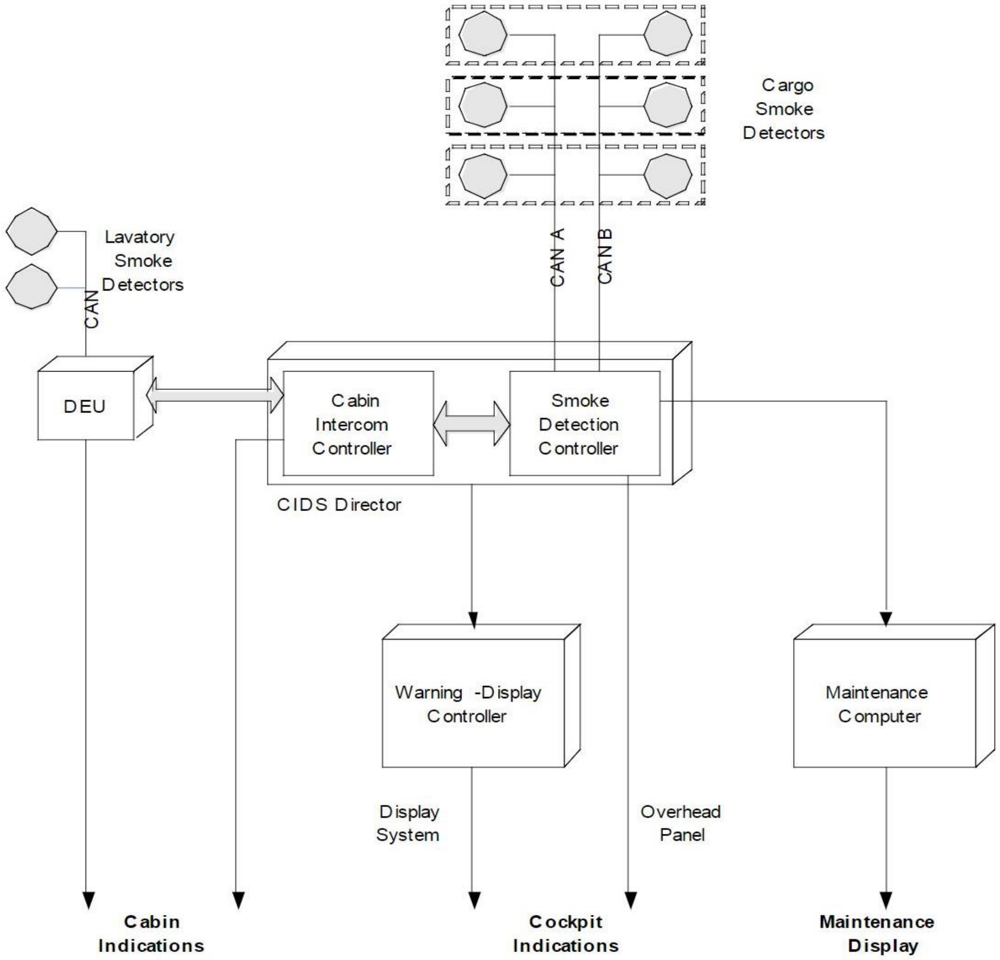


Figure 4.18 Smoke detection system using open standard CAN to interface detectors (Knueppel 2012)

4.4 More CAN Based Protocols in Aviation – CANopen, ARINC 826 and 812

Strictly speaking, **CANopen** is not a protocol for aviation, but a protocol for industrial use and automation. However, it is mentioned here because later CAN protocols are based on it and CAN-based protocols in aircraft use sensors and hardware parts originally designed for CAN-open (see smoke detection system in Figure 4.18). Since CANopen is used by many companies, there is a lot of hardware and software development based on the industrial standard. The protocol uses the 11-bit identifier as the default and is distributed by the international users and manufacturers group under CAN in Automation (CiA 2003). CAN protocols are also used in large aircraft cabins to control subsystems such as the catering elevator, water-waste systems and other cabin equipment that do not have high priority. (Klüster 2012)

A comparison of CANopen to CANaerospace has already been accurately made in the work of Horst 2018. CANopen, for example, is more specialized and does not have as many configuration options as CANaerospace. With CANopen there is not much room for personal adaptations. These are mostly given by numerous specialized device profiles which a manufacturer must comply with. In general, CANaerospace is designed simpler but needs more care and understanding in the BUS design for reliable operation. Especially for the design of a redundant CAN BUS system in an aircraft, CANaerospace offers much better possibilities. (Horst 2018, Chapter 3.4.1)

ARINC 826 is based on the protocol ARINC 825 with the specification "Software Data Loader via CAN Interface" it is also a CAN based protocol. The general specification for Software Data Load in Avionics is ARINC 615A. Since CAN is now increasingly used in aircraft systems, a data and software loader based on CAN has been created. ARINC 826 is also used beside ARINC 825 in the Airbus A350. (Klüster 2012)

ARINC 812 is also a CAN based protocol "Definition of Standard Data Interfaces for Galley Insert (GAIN) equipment, CAN Communication". It is very similar to ARINC 825 with a 29-bit identifier. ARINC 812 defines the protocol for control of equipment, power supply and monitoring of galleys. Originally commissioned by Airbus and Boing. **ARINC 810** defines the associated physical interface such as characteristics of connectors, cables, connector types, size and weight of various components in the galley. (Klüster 2012)

4.5 ARINC 429

ARINC 429 is a well-known older widely used Aircraft BUS protocol. It was published in 1977 as the Mark 33 Digital Information Transfer System (DITS), which is why it is referred to by this name in the literature. The protocol is a simplex PTP protocol that sends information purely from a transmitter to receivers. This one-directional data flow makes the protocol simple in terms of maintenance and long-term operational cost. The system consists of one transmitter and up to 20 receivers per BUS. The protocol uses a 32-bit word’s structure for data transmission with a bipolar return to zero characteristics. It can be operated at two different speeds, 12.5 and 100 kHz (see Figures 4.19 and 4.20). (Ueidaq 2021) (Kaviyarasu 2012)

Figure 4.19 Typical ARINC 429 communication line (Ueidaq 2021)

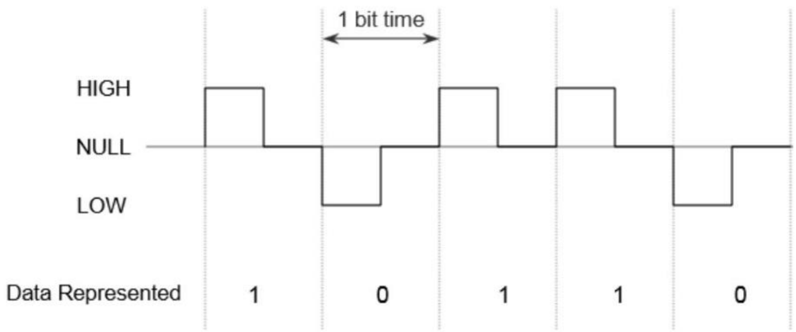


Figure 4.20 Transmission characteristic of ARINC 429 (Kaviyarasu 2012)

The transmitter transmission power is about 10 V (HIGH) and -10 V (LOW). Depending on the number of nodes and BUS length, the receiver will receive 6.5 to 13 V (HIGH) and -6.5 to -13 V (LOW). The typical logic is that high to zero is 1 and low to zero is 0. ARINC 429 uses a twisted shielded pair cable for data transmission. The resistance of the BUS system should be about 78 Ω (see cable characteristic in figure 4.21). (Kaviyarasu 2012)

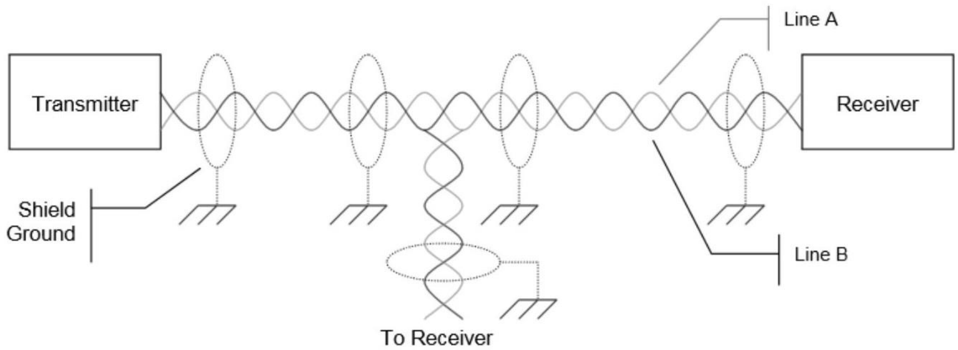


Figure 4.21 Cable characteristic of ARINC 429 (Kaviyarasu 2012)

For multidirectional communication with ARINC 429 (duplex mode), two-wire pairs are needed for both directions. The cost of implementation, hardware, software, sensors, and the limited customizability make protocols such as ARINC 429 unattractive and unsuitable for use in microlight aircraft.

4.6 Other Known Airborne Protocols

4.6.1 ARINC 629

ARINC 629 was introduced in 1995 and is used, among others, in the Airbus A330, A340 and the Boing 777. It is a full-duplex protocol that allows each terminal in the bus to send and receive messages. Up to 128 terminals are supported per BUS (see Figure 4.22). (Isik 2010)

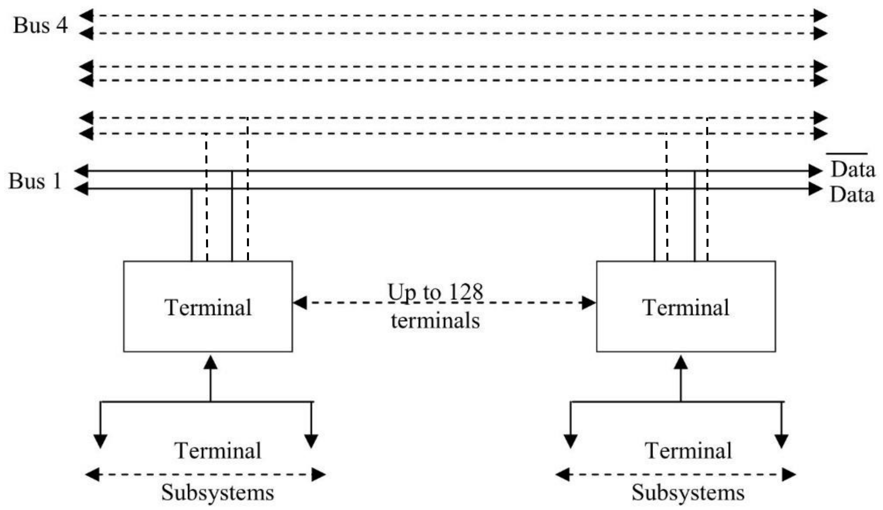


Figure 4.22 ARINC 629 data BUS (Isik 2010)

ARINC 629 is a time-based collision-avoidance protocol and transmits data at a speed of up to 2 Mbps. Each terminal in the BUS is provided with a timeslot via control timers in which it may enter the BUS. The protocol has a 20-bit word format where the first three bits are for time synchronization, 16 bits are for the data to be sent and the last bit is a parity bit that can be used for verification. (Isik 2010)

4.6.2 ARINC 664 / AFDX

ARINC 664 is also called **AFDX** and is a full-duplex ethernet link with a data transmission speed of over 100 Mbit/s (also found under **TTE**). It was developed on the basis of the industrial Ethernet with security changes in terms of package loss and real-time application. It is a newer protocol that can be used instead of ARINC 429 (only PTP) and ARINC 629 (too expensive). Since Ethernet is widely used, there is already a lot of know-how as well as cheap hardware components like computer chips and cables. In the context of ARINC 664 a protocol was created which allows a secure redundant communication in a star topology (switch in the middle, end nodes around, see figure 4.2). The message header defines the destination to which the message is sent. Smart switches buffer and route packages using a Virtual Link (VL). In this way, it is also possible to send a packet to several receivers with only one transmitted message via the smart switch (see figure 4.23, LRU 7 sends the message via the smart switch to LRU 9, 15 and 18). (Excalibur 2010)

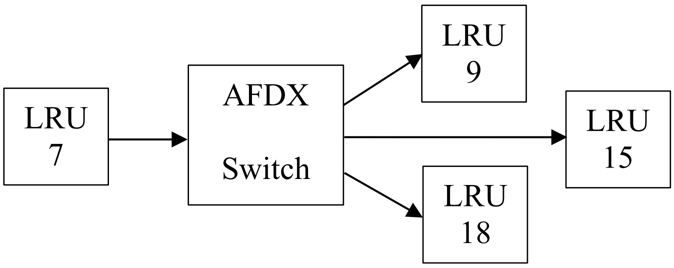


Figure 4.23 ARINC 664 / AFDX routing via smart switch (Excalibur 2010)

During transmission, a Bandwidth Allocation Gap (BAG) is assigned to each VL, which is 1, 2, 4, 6, 16, 32, 64 or 128 milliseconds long, depending on the message length. A Sequence Number (SN) is added to each message package so that the receiver can detect non-received packages and transmitter errors. In dual redundancy, messages are sent over two networks, using the SN to forward the faster message and drop the later arriving message (see figure 4.24). (Excalibur 2010)

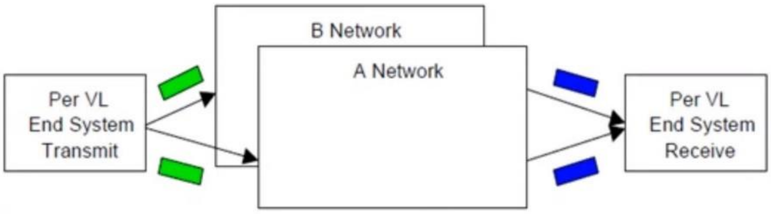


Figure 4.24 ARINC 664 / AFDX dual redundant network (Excalibur 2010)

Background data that is less prioritized can also be sent without redundancy. Figure 4.25 shows the **redundancy** function of ARINC 664. A transmitter sends 6 packages in the first line which are sent in network A (second line) and network B (third line). The first received packet is forwarded to the end system (fourth line). It can be seen that the first packet A1 and A2 are forwarded, then packet B3 arrives faster than A3 and is forwarded. A4 is not transmitted (**frame lost**) but due to the redundant system B4 is transmitted and then again the faster B5 and A6. (Excalibur 2010)

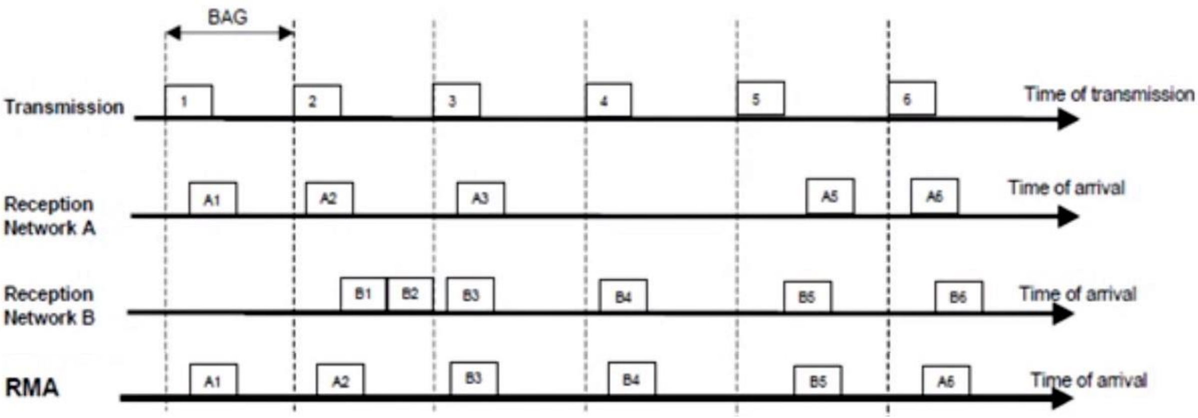


Figure 4.25 ARINC 664 / AFDX frame lost in network A (Excalibur 2010)

The entire system also supports optical transmission lines. For example, the LRUs are usually connected to smart switches with copper cables, but the switches themselves or the network backbone can transmit their data optically (see Figure 4.26). The network can also have different gateways to connect ARINC 629, 429, or CAN-based protocols. (Aviftech 2016)

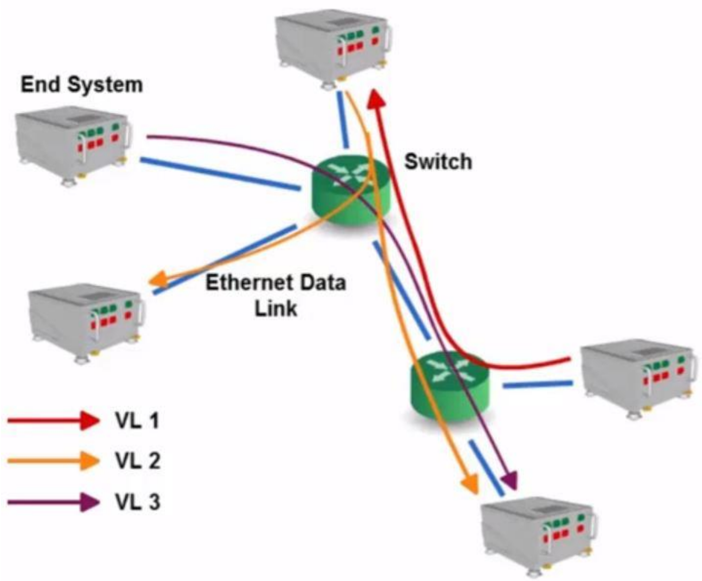


Figure 4.26 Virtual Links of the AFDX network (Aviftech 2016)

4.6.3 MIL-STD-1553, MIL-STD-1773 and ARINC 708

MIL-STD-1553 is a military standard protocol originally published by SAE in 1968, then by the US Air Force in 1973 as **Mil-Std-1553A** and in 1978 as **Mil-Std-1553B**. Initially the protocol was used by the military, but later by others such as NATO and NASA for rockets and space missions (ISS, Ariane 5, Gallileo for instance). The protocol runs on a twisted and shielded cable pair with a fixed data rate of 1 Mbps. Typically it is installed as a dual redundant system (line A and B). Typically there are three types of BUS involved: BUS Controller (BC), Remote Terminal (RT), and BUS Monitor (BM) (see Figure 4.27). (ESA 2012)

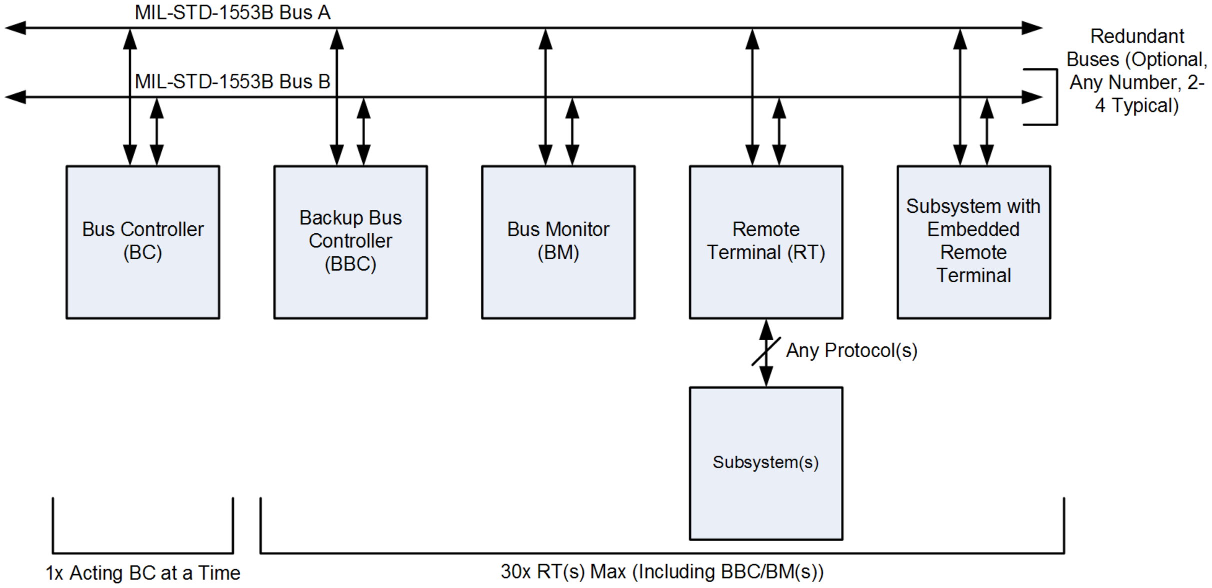


Figure 4.27 MIL-STD-1553B Data BUS Architecture (Wikipedia 2021b)

A typical simple BUS architecture with its components is shown in Figure 4.28.

Figure 4.28 MIL1553B BUS cable architecture, stub, coupler and terminator (ESA 2012)

The MIL-STD-1553B protocol is a TDM, half-duplex protocol which can handle up to 30 devices (RTs + BBCs + BMs) per BUS. The BUS cable resistance is 70 to 85 ohm. The signal transmission is a Manchester signal where a high-to-low (0.5 µs + 0.5 µs) is a 1 and a low-to-high is a 0. The operation voltage is about 18-27 volts. The protocol sends between 1 and 32 data words. Each data word can be considered as a 20-bit word, 3 bit for sinc, 16-bit payload and 1 bit for odd parity control. The transmission is always controlled by the BUS Controller (BC). (Wikipedia 2021b)

MIL-STD-1773 is the same functional design as MIL-STD-1553B with the difference that the protocol uses optical fibers. This makes it slightly lighter and resistant to electromagnetic interference (including EMP). NASA also uses the standard partially as AS 1773 with an optical dual rate of 1 to 20 Mbit/s. (Wikipedia 2021b)

ARINC 708 is also based on MIL-STD-1553B and is for airborne weather radar. The main difference is that the payload of the message can be much longer, for example, to transmit a radar image (1600 bits long). It can also use both BUS lines A and B for transmission. (Condor 2000)

4.6.4 IEEE 1394, FireWire and AS5643

IEEE 1394b is a high-speed communication standard developed in the early 1990s by Apple under the name **FireWire**. The SAE standard **AS5643** was published in 2004 and reaffirmed in 2013 as IEEE 1394 military and aerospace data network. (Wikipedia 2022d)

The IEEE 1394b high-speed interface can support many topologies, such as tree architecture, BUS (linear), and loops to increase the redundancy of the system. For example, the protocol is used in the Lockheed Martin F-35, where communication is established between 1394 devices. The data exchange takes place between computers that analyze data from the engine, flight controls, weapons systems and mission details. The AS5643 is also used for the X-47B Unmanned Combat Air System Demonstration (UCAS-D). Three data buses with three Vehicle Management Computers (VMC) were used, and the three VMC are cross-coupled for high redundancy. In addition, a loop topology is used so that even if a single port or cable failure occurs, the alternative communication path is used through intelligent configuration. (Mourn 2011)

4.6.5 ASCB

Avionics Standard Communication BUS (ASCB) is a communication protocol for small to medium-sized aircraft (up to about 100 seats). There are several versions of ASCB in which Honeywell is highly involved. The most widely used version is ASCB-D. The protocol is a fault-tolerant, periodic real-time protocol consisting of four Ethernet BUS systems. Two Ethernet BUSs run on the left side and two run on the right side with one BUS being a backup. The primary BUS of the left side also runs on the right side and the primary BUS of the right side also runs on the left side. The network interface controllers (NIC) on the left side can send messages via the primary BUS and the backup BUS on the left side, but they can also receive messages from the primary BUS on the right side (listen-only). In the same way on the right side, the network interface controllers (NIC) can send messages via the primary BUS and the backup BUS on the right side, but they can also receive messages from the primary BUS on the left (listen-only) (see Figure 4.29). (Paulitsch 2015)

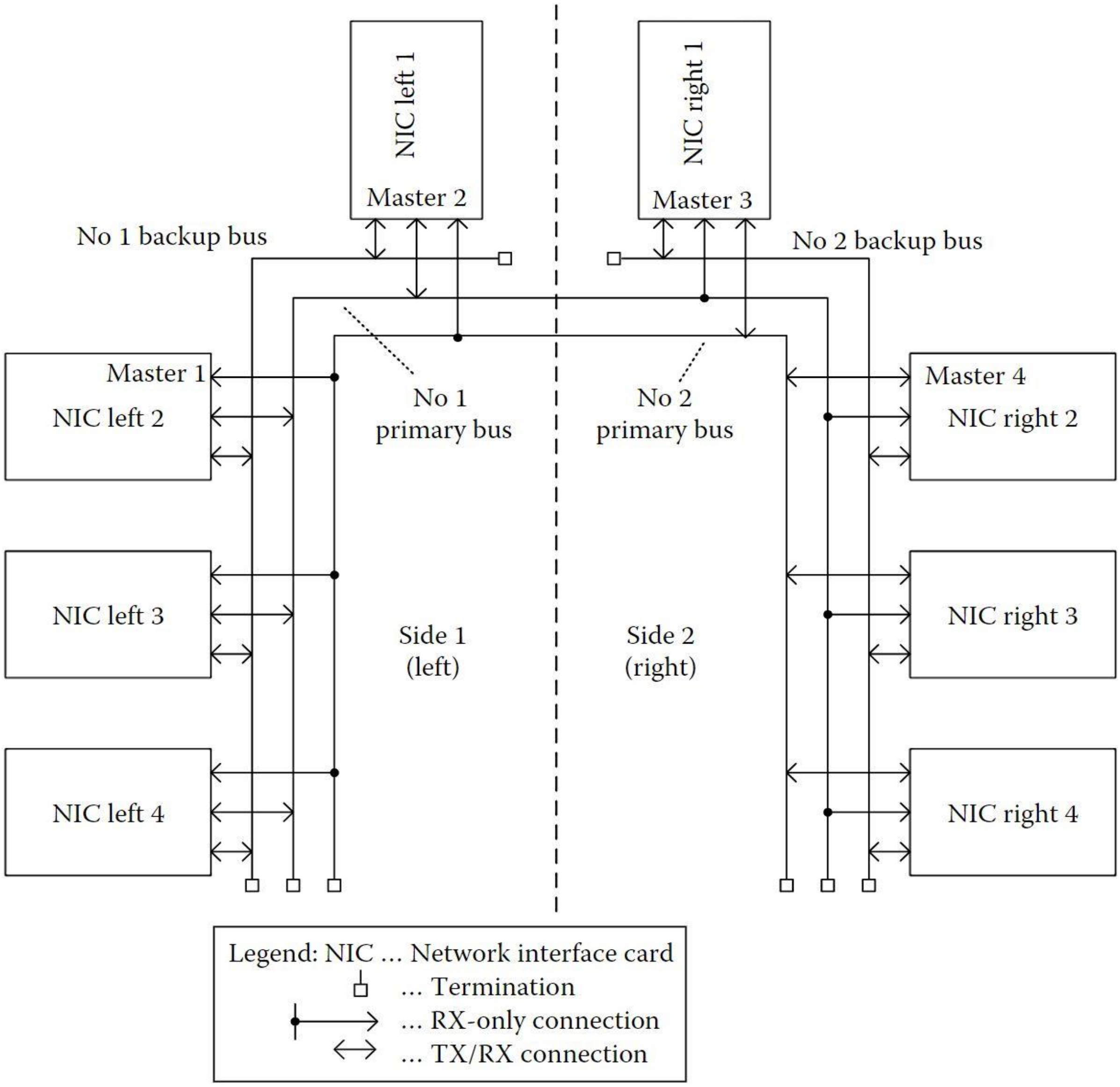


Figure 4.29 ASCB architecture (Paulitsch 2015, Chapter 44.2)

The ASCB-D is a TDMA based protocol with a speed of 10 Mbit/s (today even faster). It uses twinax cables and connectors (twinax or twinaxial cables are like coaxial cables with 2 inner conductors instead of one). In the field of ASCB-D avionics, Honeywell has designed the Primus Epic Suite as a Modular Avionics Unit (MAU) which is installed in many airframes like: Agusta AW139 (helicopter), Cessna Citation Sovereign, Dassault Falcon aircraft, Gulfstream aircraft, Embraer 190/195 and others. With the MAU, weight savings, small unit size, improved maintainability, less power consumption, less wiring and more functions could be achieved compared to older concepts. Figure 4.30 shows a dual redundant example system diagram with the MAU Primus Epic. (Paulitsch 2015)

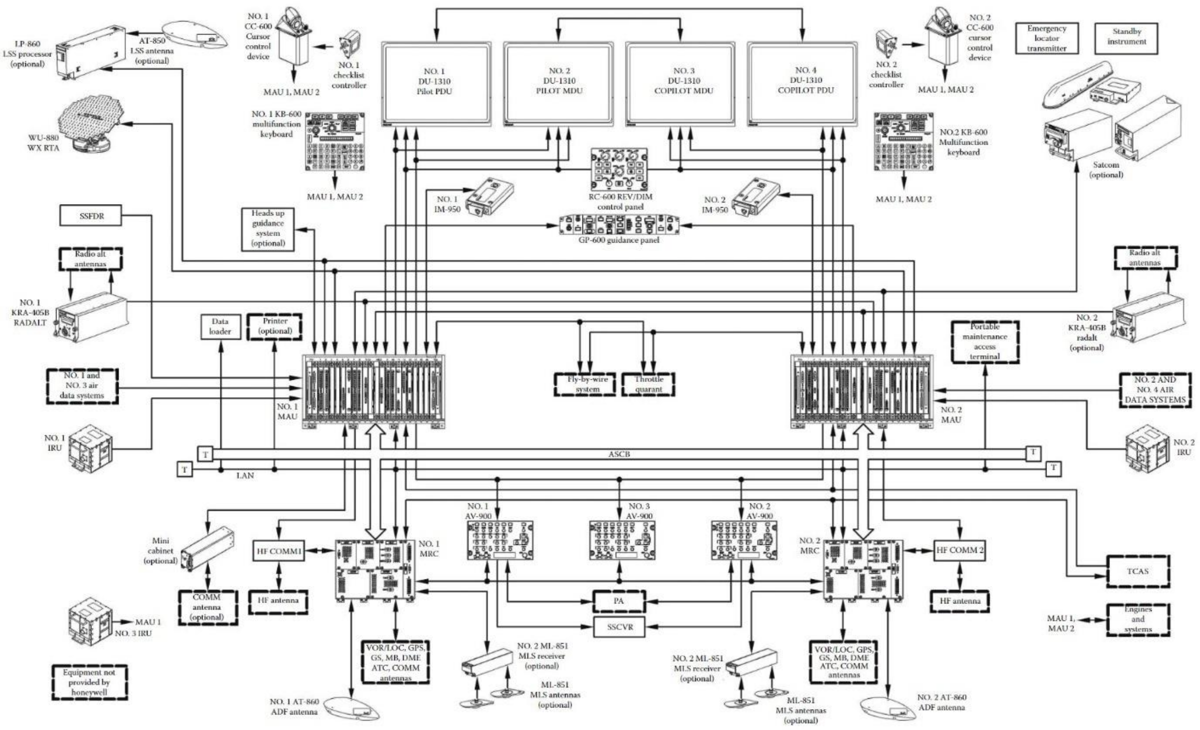


Figure 4.30 ASCB – Primus Epic example system diagram (Paulitsch 2015, Chapter 44.6)

In 2008, Stock flight systems implemented a Telemetry Interface System (TIS) on the DLR High Altitude and Long Research Aircraft (HALO) platform, providing an interface between ASCB-D and CANaerospace respectively ARINC 825. Live values from the ASCB-D could also be read out with the TIS during the flight (Stock 2011b).

4.7 RS-232

The Recommended Standard (RS) 232 is not a protocol for aviation but a serial communication standard which was already introduced in 1962 in the fifth version as EIA/TIA-232-E. EIA/TIA stands for Electronic Industry Association and the Telecommunication Industry Association where the standard was developed. The RS-232 standard defines voltage and signal levels as well as pin-wiring configurations. The maximum data transmission rate is 1 Mbit/s. However, the RS-232 standard also specifies a maximum data rate of 20 kBit/s to reduce the chance of crosstalk. In company applications, data rates of around 250 kBit/s are mostly used today. The output voltage is defined between 5 V and 15 V, with voltage drop and noise defining a voltage of 3 V to 15 V at the receiver. At the receiver, a 3 V to 15 V is a logic 0 and a -3 V to -15 V is a logic 1 (Maxim 2001).

Figure 4.31 shows on the left a schematic diagram and on the right a measured oscilloscope plot of a binary (1 + 2 + 0 + 8 + 0 + 0 + 64 + 0 =) 75 which stands for an ASCII "K" character. The measured time of one bit is 106 µs at a voltage of -5.68 V and 6.48 V (see Figure 4.31).

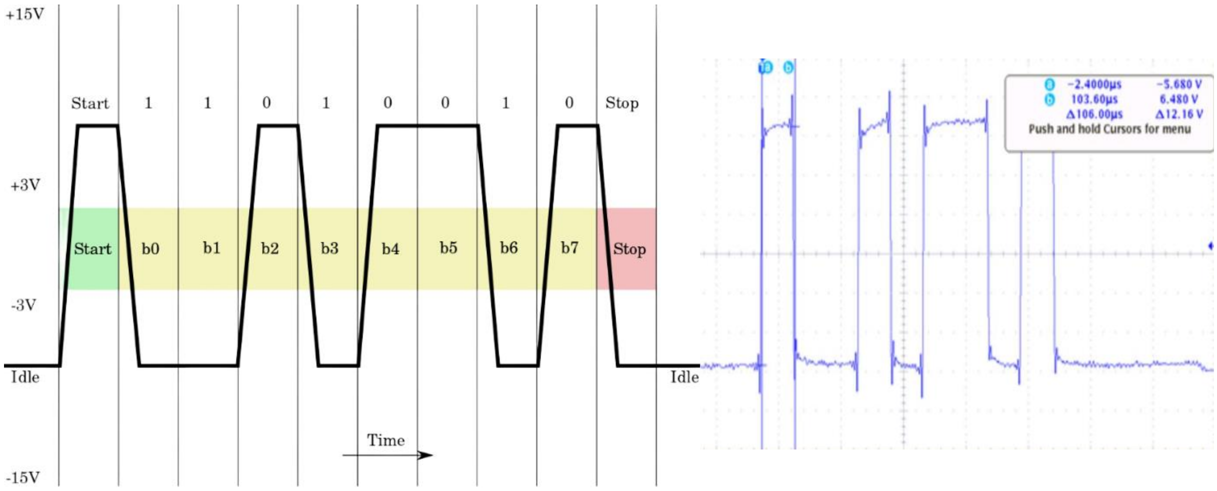


Figure 4.31 RS-232 data line and voltage level, schematic (left), oscilloscope on receiver (right) (Wikipedia 2022)

The communication with RS-232 takes place point-to-point. For a full-duplex communication, at least a 3-wire cable is needed where from device A one cable goes from TX to RX to device B and one cable from TX to RX to device A. The data lines share a GND cable from device A to device B. The resistance between the data line and GND is about 5 kΩ (3 to 7 kΩ). The maximum cable length is defined as a maximum cable capacitance of 2500 pF. (Maxim 2001)

RS-232 is often used as a general standard for connecting electrical devices from different companies. For example, the microlight avionic component iLevel 3 AW in Chapter 5.2.3 uses a DB15 connector where pin 2 is a TX RS-232 and pin 3 is a RX RS-232 (see Figure 5.6).

Also AUX0, AUX1 and AUX2 (RX and TX) on pins 10, 11, 12, 13, 4 and 5 use RS-232 standard to process data from ADS-B, GPS, AHRS and others (Levil 2018). See Chapter 5.2.3.

4.8 Radio Control (RC) Transmission Protocols

In the Radio Control (RC) area, there have been many developments in protocols from different manufacturers to reduce the number of cables for the individual channels to be controlled. The general control of individual RC parts such as servo motors and Electronic Speed Controllers (ESC) for brushless motors is done with a Pulse Width Modulation (PWM).

PWM is the modulation of an electrical voltage at a constant rectangular frequency and voltage. Due to the modulation and the on-off ratio of the PWM rectangle, an averaged counterpart output can be transmitted with demodulation. In figure 4.32 the PWM is shown in blue with the corresponding averaged counterpart in red. In the first section, the ratio of on to off is very small and increases up to 0.105 s. Then the ratio of on to off is very large and decreases to 0.110 s. The process is repeated except that the PWM signal is not transmitted at 100 V but at -100 V. The averaged counterpart is a sine wave with a length of 0.02 s. Using demodulation with a low-pass filter, it would be possible to output all volt ranges between -100 and +100 V as DC voltage with the PWM shown. Figure 5.20 shows a constant PWM with the corresponding average. Some LED lights can only be on or off. To dim such a LED light anyway it is modulated. Because the modulation frequency is much higher than what we can see, we see a constant light intensity. (Vujacic 2017 and Mike Cook's 2021)

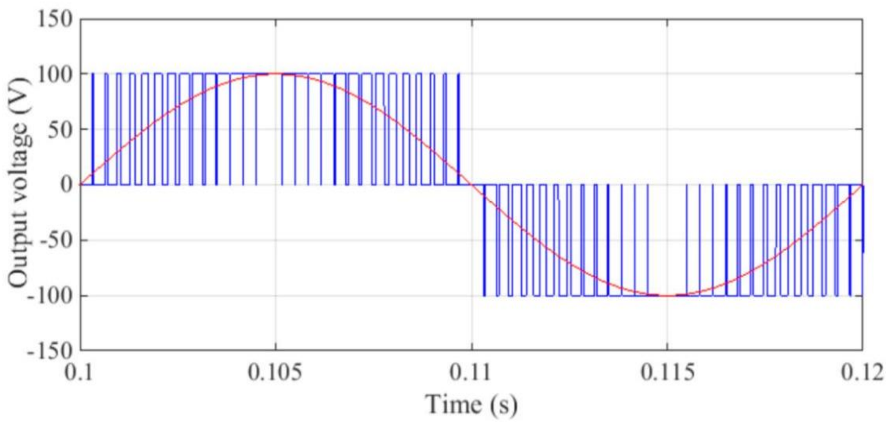


Figure 4.32 PWM output voltage and the averaged counterpart (Vujacic 2017)

In general, the PWM voltage in the RC sector is 5 V. Many RC components are connected with a wiring harness which includes the signal transmission and the voltage supply. Therefore you have three cables with VCC, PWM and GND. The general voltage is 5 V and is sufficient for most boards. Components that need more power or voltage usually have two more separate cables (VCC and GND). If a larger RC model has a flight controller with 10 channels to be controlled, then 10 harnesses (each with 3 cables) would have to be laid from the receiver to the controller. Since all PWM signals require only one power supply, this could be reduced to 12 cables (1st channel PWM, VCC and GND, 2nd to 10th channel only PWM cables). To reduce the cabling even further and to reduce the price, weight and size of receivers, protocols have been developed so that only one cable harness (with 3 cables) has to be laid from the receiver to the controller. Table 4.2 shows the most known protocols from the RC world with the channel number and the manufacturers.

Table 4.2 RC transmission protocols (Optimum 2016)

Protocol	Manufacturer	Number of channels
SBUS [a]	Futaba, FrSky	18 channels
IBUS [b]	Flysky	14 channels
XBUS [c]	JR	14 channels
MSP [d]	Multiwii	8 channels
SUMD [e]	Graupner	16 channels
SUMD3 [f]	Graupner	32 channels + 64 switches
CRSF [g]	TBS	12 channels

[a] Inverted Universal Asynchronous Receiver / Transmitter (UART) - COM signal. The UART is a hardware device for asynchronous serial communication where the data format and transmission speeds are configurable. It sends data bits from the lowest value to the highest value, framed by start and stop bits so that the communication channel handles the timing. It was one of the earliest computer communication devices used to connect teletypewriters for an operator console (Enzyklopädie 2020)

[b] Is a protocol that allows bidirectional communication (duplex)

[c] Low delay between individual channels

[d] a protocol with MultiWii developed (MultiWii is a general software for controlling multirotor RC models)

[f] (Badziong 2021)

[g] Crossfire is a similar protocol to SBUS which allows higher update speeds and bidirectional communication (duplex). It is designed for FPV RC flight with good long-range characteristics by adjusting the transmit power (TBS 2019).

Such simple protocols can also be used in microlight aircraft. For example, if the landing gear is to be extended electrically, the nose wheel as well as the main landing gear has electrically opening flaps that are operated with a servo motor, then the control unit does not need 3, 6 or more PWM outputs but only one SBUS output. There are inexpensive SBUS to PWM converters that can be installed just before the servo motors. Also for the installation of electrical trims in the airplane a simple BUS system from the RC world could be used. Fail-safes would have to be developed in individual cases together with the drive controller if needed.

4.9 ZigBee, Z-Wave and EnOcean

ZigBee is a low-cost, low-power protocol for device-to-device data connectivity. Over 400 companies define and use the standard for the communication of various devices such as home automation, building automation, sensor networks, smart energy and personal health care monitoring. The group of companies is united under the ZigBee Alliance. The ZigBee standard includes all ISO/OSI layers in its architecture (see Figure 4.33). (Silicon 2021)

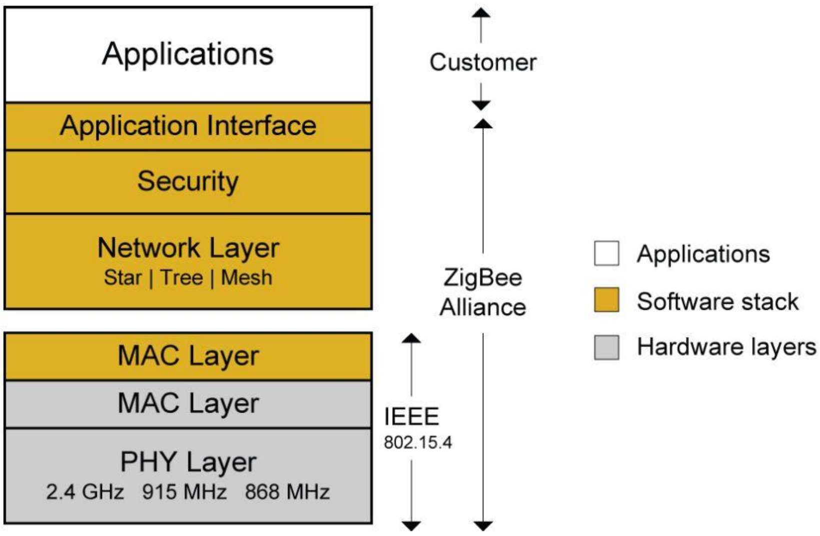


Figure 4.33 ZigBee ISO/OSI Architecture (Silicon 2021)

The ZigBee protocol is based on the IEEE 802.15.4 MAC and physical layer which operates standardly at 2.4 GHz with a speed of 250 kbyte/s. However, the real data speed through the ZigBee network is lower and it is generally designed for low data rates. The network performance is based on the payload size. Generally, a 127-byte packet is sent with a data payload of 68 bytes. If the message is larger than 68 bytes then the message is split into fragments that are sent in multiple packets (see Figure 4.34). (Silicon 2021)

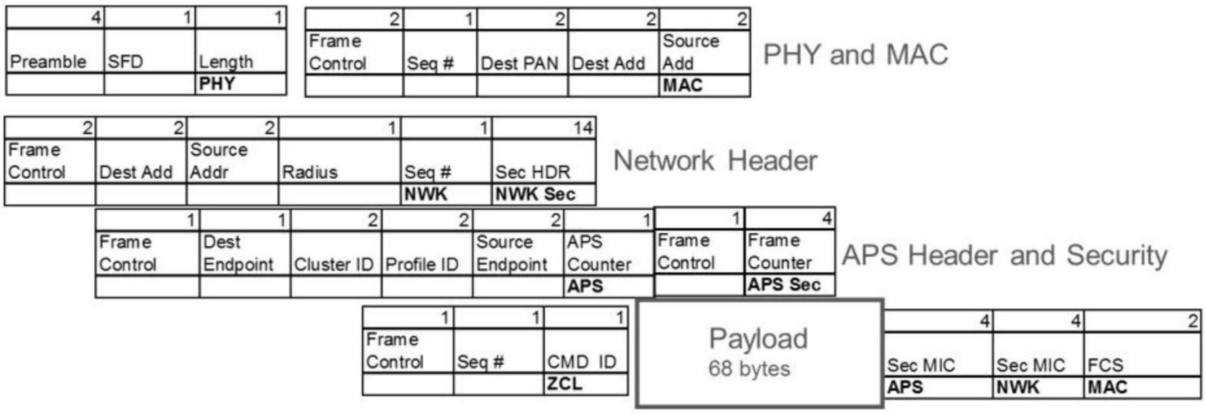


Figure 4.34 ZigBee packet format (Silicon 2022)

Depending on the number of devices, ZigBee can form any mesh for forwarding data. In this way, messages can be forwarded from a sending node via several intermediate network nodes to the receiver or a gateway. A ZigBee coordinator (ZC) is responsible for forming the network. It defines network addresses, selects the correct channels and is also responsible for security settings and authorizations. It monitors and rectifies network problems, manages channel changes in the event of faults, and more. Some functions can be defined in the user layer in ZigBee 3.0. (Silicon 2021)

In case ZigBee device A wants to send a message to device C but no route is available, the network software will initiate a route discovery. Device A sends a message to the entire network and asks device C for a response. C receives the message and the device list through which the message was forwarded. C can now send a response back to A via device B or device D. Silicon Labs uses a weighting algorithm to select what appears to be the most reliable next hop. C chooses B as the hop and sends a temporary route to A (blue) which A confirms as a permanent route (green). The route discovery process is complete and the route can be used to transmit data from A to C or from C to A (see Figure 4.35, 1. and 2.). When messages are transmitted, the application specifies whether an end-to-end acknowledgment should be sent from the receiver (called APS acknowledgment). In case of a time-out (no response) the route may need to be repaired and the sending device will be informed. If a node in the mesh is corrupted, it will be excluded from the selection when a new route is chosen. Device A updates the routing table so that the next hop on node is device D. A makes a temporary routing request via D to C which is confirmed by C as a permanet route (see Figure 4.35, 3. and 4.). (Silicon 2021, Chapter 4)

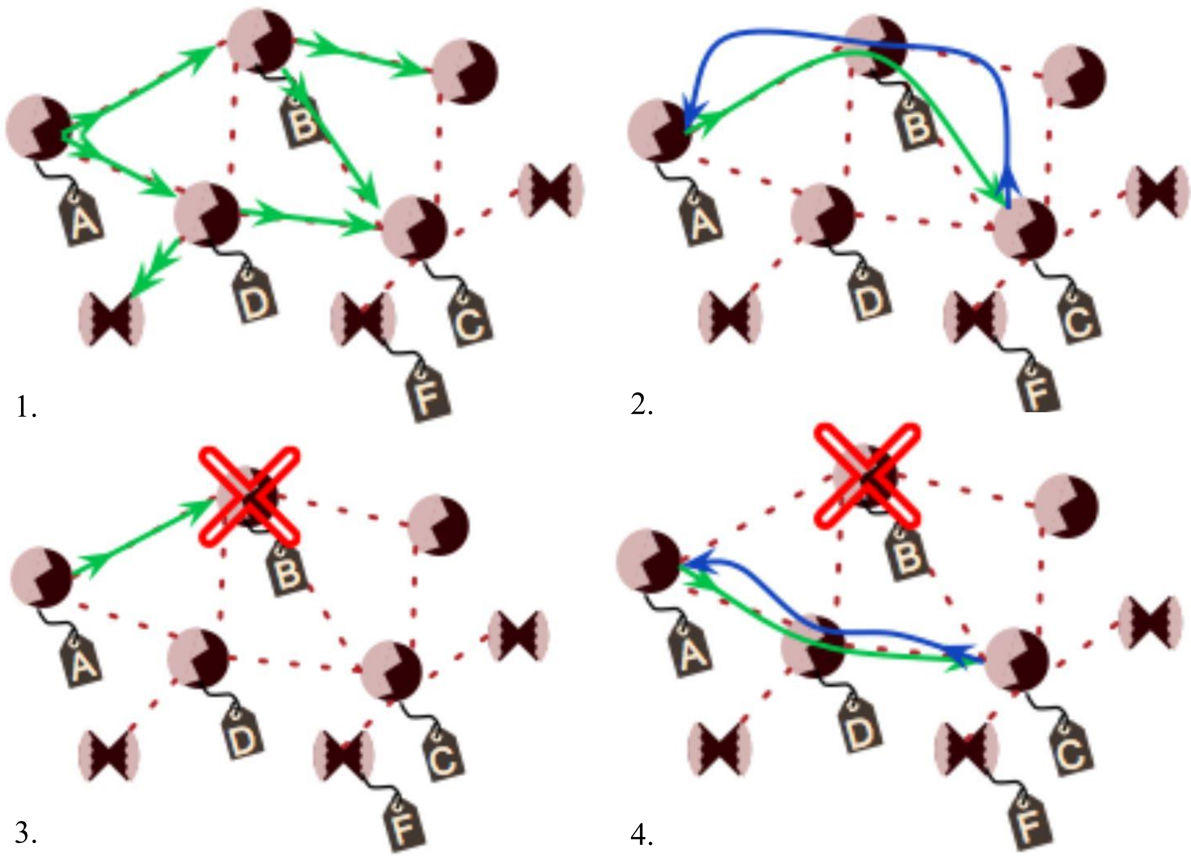


Figure 4.35 ZigBee network routing (Silicon 2021, Chapter 4)

Already in an older paper "Integrating Zigbee and CAN Networks in Industrial Application" it is described how the Zigbee mesh network can be connected to a CAN bus. This is investigated on the basis of wireless ZigBee sensors that are to be connected to a CAN network. If Zigbee sensors are to be used in aircraft, a gateway could also be used to connect them via CAN to a controller or another gateway for further data processing (Gomes 2010).

Various tests were performed by Silicon Labs, including packet loss and speeds at different ranges in the open (see Table 4.3). It can be seen that at different distances the average round trip time of a message remains almost the same at 18 ms. Only the susceptibility of packet loss is higher at longer distances. In this test, 50 bytes of payload data were sent in an interval of 100 ms without a hop and with disabled security (without APS confirmation). (Silicon 2022, Chapter 2)

Table 4.3 Packet loss for distance and average round trip (Silicon 2022, Chapter 2)

Distance	Average round trip	Packet loss
20 meters	18.0 milliseconds	0 %
50 meters	17.9 milliseconds	0 %
75 meters	17.9 milliseconds	0.75 %
85 meters	18.6 milliseconds	1.65 %

Another test result shows that with different payloads of 50 to 250 bytes, about 25 to 60 ms per hop is required (see Figure 4.36). (Silicon 2022, Chapter 4)

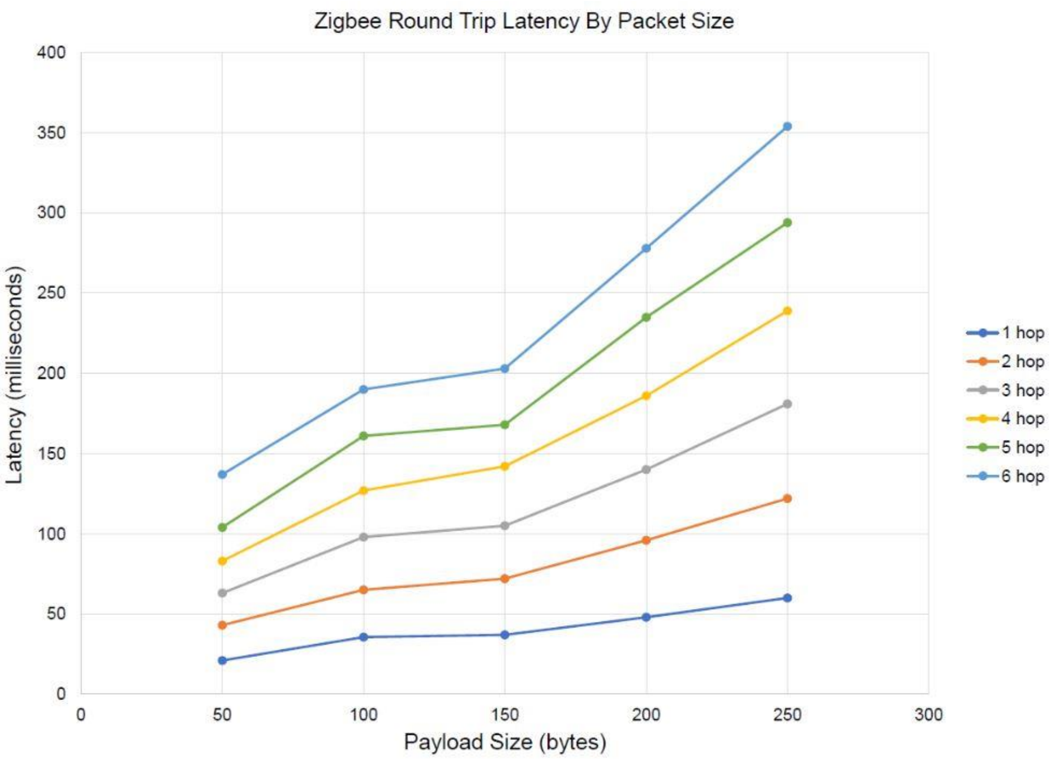


Figure 4.36 ZigBee round trip latency by packet size (Silicon 2022, Chapter 4)

The ZigBee transmission system and protocol show great potential in smart data transmission. Node errors and the self-repairing nature of the data mesh could ensure a high degree of reliability, especially with redundant nodes. Complete data exchange of sensors in the aircraft via a wireless connection would extremely reduce the wiring effort. ZigBee 3.0 already has many cluster libraries with useful functions. For aircraft use, however, an adapted ZigBee profile would probably have to be developed to meet the safety and redundancy requirements. Even in the case of a complete loss of signal through, for example, a jammer attack, an aircraft should never be in danger.

Z-Wave is similar to ZigBee a well-known smart home communication protocol for a wireless short-distance transmission with a mesh topology. While ZigBee operates at 2.4 GHz, Z-Wave operates in the low-frequency range at 868.42 MHz in Europe. In the USA Z-Wave runs mostly on 908.42 MHz. Depending on the country, the frequency varies from 865.2 MHz (India) to 926 MHz (Japan) (Wikipedia 2022b). Because ZigBee uses a higher frequency than Z-Wave, the transmission speed of up to 250 kbyte/s is higher than with Z-Wave with a maximum of 100 kbyte/s. In return, the range of Z-Wave is a bit better. ZigBee is also a more widespread standard with more profile specifications and many hardware manufacturers. Z-Wave is simpler and hardware parts are only available from Silicon labs. Z-Wave is limited to 4 hops with a maximum of 232 devices per mesh. ZigBee has no hop limit and supports up to 65 000 devices per mesh. Z-Wave is slightly more power-efficient so simple sensors that are not powered by a cable will last for many years on a coin cell battery. (Shea 2018)

EnOcean is a radio transmission technology for sensors and switches designed for ultra-efficient energy use, mostly battery-free and maintenance-free with a lifetime of over 20 years. Surrounding light can be converted into electricity with small solar panels, mechanical motion such as pressing a button can be converted into energy, and potential differences such as temperature can be converted into electricity with small Peltier elements. The current is sufficient to transmit a simple 14-byte long binary signal with a transmission rate of 125 kbit/s at a frequency of 902 MHz, 928.35 MHz, 868.3 MHz or 315 MHz. Generally, 3 packets are sent in pseudo-random spacing which reduces the probability of packet collision. Switches can send an additional data packet when released, enabling additional functions such as dimming lights. EnOcean GmbH was originally founded as a sub-company of Siemens AG in 2001. In 2012, the technology was ratified as the international wireless standard ISO/IEC 14543-3-10 "Information technology - Home Electronic Systems (HES) - Part 3-10: Wireless Short-Packet (WSP) protocol optimized for energy harvesting - Architecture and lower layer protocols" (Wikipedia 2022c).

For use in microlight aircraft, Z-Wave offers sufficient data speed alongside ZigBee for the transmission of sensor values and other device status information. EnOcean has great potential as a technology and will probably be encountered more in the future. In large passenger aircraft, a wireless mesh connection for all passenger service buttons would be conceivable, especially if they were wireless, battery-free and designed for the entire service life of the aircraft. However, any radio communication must not interfere with any other function of the aircraft, such as the aircraft radio. Manufacturers of avionics for small aircraft could create a complete ZigBee avionics bundle with sensors, displays and backup batteries for easy integration into the aircraft.

New microlight aircraft could be equipped with this and especially the complete homebuilder and prototyping market would be very interested in a well-working avionics system without wiring effort.

In a PhD thesis, Hope 2011 investigates the propagation environment of aircraft. Large sections of the aircraft may be highly resonant, which causes difficulties for wireless receivers. Two standards are recommended: 802.11n (Wi-Fi 4) and ZigBee. For these, the propagation in an airframe is studied in-depth and research is done on what data rates and which systems could be made wireless. The work also introduces a patch antenna (created by an algorithm) that can be written directly on the aircraft's skin (more on this in Chapter 6, *Future of Avionics*). (Hope 2011)

The author Hope has also previously published research on this topic in various papers with the keywords multipath resonant channel, fly-by-wireless, wireless sensor networks and ZigBee. (Panitz 2008 and Dawson 2008)

In the paper by Dawson 2008, it is mentioned that low-power wireless sensors could be powered by energy generated from the surrounding environment in the form of vibrations, thermal potentials or airflow. He refers to Paradiso 2005 who presents in his work a series of techniques to scavenge energy. (Dawson 2008)

See the next Chapter and Chapter 6 for more information on wireless communication.

4.10 Decision and Selection

In Chapter 4, many different protocols were presented. The focus is on CAN and CANaerospace for use in microlight aircraft. This is because the protocol is much cheaper than other aviation protocols and allows a lot of customization. The implementation workload also depends very much on the type of devices that are to be connected to the CAN BUS.

For example, if a simple sensor sends simple values cyclically over the BUS, the receiver electronics will filter the message and read the value. Such a simple communication is simple to program, simple to test, failure mode and effects analysis are simple to perform and the treatment is straightforward (layer 2 communication). A very complex design with a lot of effort and cost would be a sensor that has to be calibrated via the BUS or software configuration that has to be downloaded and flashed via the BUS. Since the software is larger than the payload of a single message, many messages would have to be assembled. The result would be the design of a transport protocol with a command interface which is very complex to design with expensive implementation and testing. Service and testing tools are also needed for a successful implementation. For example, ARINC 826 would be a suitable protocol for this. (Klüster 2012)

Depending on the requirements, there are many protocols and possibilities to design a system. No protocol can be defined as the best, but it can be said that for an avionics design with affordable sensors in microlight aircraft, a CAN BUS system is suitable. Because of the redundancy possibilities and the predefined identifiers for the flight application, CANaerospace is the right choice. This CAN BUS has already been tested in many applications, also with respect to trim and autopilot possibilities (see Stock 2009, *Fly-By-Wire for Experimental Aircraft?*). By default, in small aircraft with fewer electrical systems, the "star network" topology is used where all devices and sensors usually run together in the cockpit and are wired to controllers or displays (see Figure 4.2, *network topologies*). The additional cost of a CAN system design is usually too high to justify the cabling effort or the weight reduction. From a certain production quantity, however, microlight aircraft manufacturers should consider a general CAN system design.

Avionics manufacturers use their own protocols for their own components. Dynon Avionics (see Chapter 5.2.5), for example, uses its own serial data formats for the connection of different units to each other. One of these is the NMEA (National Marine Electronics Association) serial data format developed for use in boat navigation and control systems. The *SkyView System Installation Guide* shows all serial data formats with examples (see SkyView 2021, Chapter 26). The interest of avionics manufacturers to use general communication protocols across devices is not very high, since complete avionics packages are supposed to be sold. It is also not unusual to buy communication functions for other manufacturers' equipment. This is usually a feature that can be unlocked with a license key. When selecting a suitable protocol, the connection to other existing devices and systems must be carefully considered. Connection must also be possible for devices that could be retrofitted in the future.

If wireless communication between sensors and between electronic components is to be implemented on a larger scale, then according to Hope 2011 Wi-Fi 4 or ZigBee are suitable (see the end of Chapter 4.9). At the end of Chapter 3.4, a Wireless Avionics Intra-Communication (WAIC) system is introduced, which is focused less on the transmission technology than on describing why wireless transmission makes sense and which components can be considered.

They also propose to use the 4200-4400 MHz frequency for an aircraft WAIC. The frequency is used worldwide as Aeronautical Radio Navigation Service (ARNS), but only for radio altimeter assignment. See Appendix D for the U.S. frequency allocations, which shows that many frequency bands are reserved for aeronautical radio navigation (red). The 4.2 to 4.4 GHz range is in the second-lowest row (see Appendix D from Commerce 2016). In International Telecommunication Union Radio-communication Sector (ITU-R) studies, WAIC showed no negative impact on radio altimeter performance. (AVSI 2015)

In the World Radiocommunication Conference 2015 in Geneva, WAIC was presented and published in WRC-15, resolution 424 after which ICAO starts the development of Standards and Recommended Practices (SARPs) for WAIC. (WRC 2015, page 320-321)

In WRC 2015, page 26 and 30, the frequency 4 200 - 4 400 is now for AERONATICAL MOBILER (R) 5.436 and as before the AERONATICAL RADIONAVIGATION 5.438, 5.437, 5.439 and 5.440. The description of the 5.436 by WRC 2015 is as follows:

> *5.436 Use of the frequency band 4 200-4 400 MHz by stations in the aeronautical mobile (R) service is reserved exclusively for wireless avionics intra-communication systems that operate in accordance with recognized international aeronautical standards. Such use shall be in accordance with Resolution 424 (WRC-15).*

So there are already several approaches to using wireless communication in aircraft, whereby WAIC refers to commercial aircraft (more in Chapter 6, *Future of Avionics*). In order to install new technology, many certification steps must be overcome before a safe installation is permitted. Difficulties are also that frequency allocation can vary from country to country, which is why international frequency allocation is important. In the microlight sector, the installation of sensors that enable wireless data communication would be possible without further certification as long as the aircraft can continue to be flown (even in the event of a system failure) with the minimum equipment. As described in Chapter 4.9, ZigBee is very well suited to transmitting simple information in a mesh network. However, this is still a completely new field with high potential for the future. Especially for avionics manufacturers, a more detailed investigation and a possible profile definition for Zigbee is worthwhile, which could create a completely new product range.

Chapter 4.7 *RS-232* also shows a communication standard that is often used as a general standard by various companies. RS-232 is also only a point-to-point protocol, so cables cannot be reduced. But it can be implemented quite fast and devices from different manufacturers can be connected.

For microlight prototypes and small series production, the RS-232 standard is used as the communication system. If several aircraft of one type are to be equipped with avionics, or if an aircraft is to be equipped with a high number of avionic components, BUS cabling with CAN-aerospace is worthwhile. For systems that do not require fail-safe operation, wireless communication with ZigBee will come into question over time. These will be wireless sensors that send their data to a display or ADAHRS via serial interfaces such as RS-232, or directly to a smartphone or tablet via WiFi.

5 Flywhale Microlight Aircraft

5.1 Introduction of Company and Aircraft

Figure 5.1 Flywhale Aircraft - Uniplanes GmbH

I did my main internship at the company Uniplanes GmbH and worked on the Flywhale aircraft (Figure 5.1). Uniplanes GmbH (previously Flywhale Aircraft) is a small aircraft company located in Dötlingen, Northern Germany and was founded in 2012 by Elke and Helmut Rind. Mr. Rind started developing the first prototype of Flywhale in 2007 in his own workshop where the first prototype FW01 (Flywhale 1) was built. A test pilot flew the prototype in the ground effect of the airfield where important knowledge was gathered for modifications. Together with a small team, they developed the second prototype FW02 in the following years. In the development of the second prototype, there was also a fuselage for the breaking load test, where the requirements for the certification of the DAeC were proven. Since the prototype was successfully completed in 2015, further prototypes have been built and sold to customers.

A weight increase became possible in 2018 by EASA with Regulation 2018/1139. Germany then published in 2019 in the LTF-UL the national regulations, since then the ultralight aircraft are allowed to have a $m_{MTO} = 600$ kg, aircraft with swimmer (amphibious aircraft) are allowed to have a $m_{MTO} = 650$ kg (more information on this subject can be found under Gmelin 2021). So it started again with the work of planning, implementation and approval of the weight increase to an $m_{MTO} = 650$ kg. On 2019-08-13 the rejection test of the BRS (Ballistic Recovery Systems) was carried out, which was successful (Figure 5.2). Unfortunately, the company had to declare insolvency at the end of March 2020, after which it was sold and since August 2020 has new owners under the company name Uniplanes GmbH. For this purpose, investors consisting of experienced aviation specialists from Germany, Austria and Liechtenstein were grouped. The investors aim to optimize and produce the Flywhale Adventure single-engine ultralight amphibian aircraft. Other products to expand the portfolio are also being planned for the future. The current Flywhale aircraft is a modern flying boat made of lightweight CFRP (Carbon-fiber-reinforced polymers) composite material, equipped with a Rotax 912 iS Sport engine and a three-bladed constant-speed propeller.

Figure 5.2 Flywhale rejection test of the BRS (2019-08-13)

Since I already have a lot of experience with electronics in my private life, it quickly became clear that this is where my strengths lie, and I was made familiar with the aircraft's electrical system. I started with the panel wiring of the eleventh prototype FW11, having the disassembled panel what was previously planned and prepared for the aircraft. There were now several changes because on all previous models the circuit breakers were directly in the panel (see Figure 5.10 *Flywhale panel with fuses (left) and Flywhale panel with VPX box installed (right)*), but now they were replaced by the VP-X Sport Box, which is a digital circuit breaker (see Chapter 5.2.2). The FW11 aircraft is a very special one, because the customer doesn't want to have the Dynon system installed, but connects two iPads via the iLevel 3 AW (see Chapter 5.2.3). This is a more affordable option and opens up further possibilities for integrating other devices. The customer would like to have everything displayed on his tablet later, such as the Rotax engine data, VP-X circuit breaker status and the status of the left and right fuel level. Figure 5.25 shows the engine monitoring system (EMS) in the center of the cockpit. When integration into the iLevel is complete, this component will be omitted. For the integration of the fuel level indicator, hardware still has to be developed and code written, which the customer has not yet done but will do later. Until then, there must be a fuel level indicator to ensure safe flying with the aircraft. The other aircraft equipped with the Dynon Avionics already have the integration of the fuel indicator so that the problem now only occurs with the FW11 aircraft (see Chapter 5.3, *Fuel Level Indicator*)

5.2 Avionics

The Flywhale aircraft has a number of electrical components which are briefly described in the following chapters. Other components for the microlight aircraft sector are also partly described, although there are of course many other brands and devices that are suitable for the microlight aircraft sector and are not shown here.

Generally, in Germany it is possible to install digital flight instruments in addition to the instruments of the minimum equipment (airspeed indicator, altimeter, turn coordinator and heading indicator). By means of a paper from the DAeC it is also possible to exchange type certificated analog flight instruments for digital flight instruments (see Appendix F (page 142), *Supplementary Device Test DAeC*). In order to prove the corresponding safety requirements, a Quality Test Report (QTR) must be carried out for each electronic component to be used, which replaces a component from the minimum equipment. Flywhale would like to give customers the option to install a Garmin G5 display which includes an airspeed indicator, attitude indicator, altimeter, turn coordinator, heading indicator and vertical speed indicator. Using the QTR, a Garmin G5 can be installed instead of the analog indicators of the minimum equipment (see QTR in Appendix G (page 144) and Garmin G5 in the following Chapter).

A modern microlight aircraft is equipped with an Automatic Dependent Surveillance-Broadcast (ADS-B) which transmits position data of the own aircraft and receives and displays position data of other air traffic. Even newer drones are equipped with ADS-B receivers (for example DJI Air 2S) to display aircraft and helicopters to give a warning if they are in the area.

Transponders are mandatory in Germany for flights in airspace C, D and TMZ (Transponder Mandatory Zone) and for flights above 5000 ft Mean Sea Level (MSL) or 3500 ft GND (altitude coding 7000, mode ALT). In the transponder, a code is entered which is prescribed by the Air Traffic Control (ATC). (Kassera 2016, page 188).

In order to have the most freedom in flying, most microlight aircraft have a transponder installed which can also contain ADS-B functions.

FLARM (Flight Alarm) is also a traffic awareness and collision avoidance technology for smaller aircraft. It sends and compares GPS position data to provide an early warning (see more in Chapter 5.2.4).

A radio (COM) goes from 118.000 to 136.975 MHz with a channel spacing of 8.33 kHz. In rare cases, flights without radio communication are permitted at uncontrolled airfields, but in practice, every microlight aircraft has a radio onboard. (Kassera 2016, page 188)

5.2.1 Garmin G5

The Garmin G5 electronic flight instrument can independently read the pitot pressure and therefore the relative speed. The device has a display to indicate airspeed, altitude, altimeter, turn coordinator, heading and vertical speed. A system like the Garmin G5 that can display an artificial horizon is an Air Data and Heading Reference System (ADAHRS). Mostly the device is not used on its own but as a backup for other systems. In the event of a power failure, the device can be equipped with a self-contained battery, which extends the device runtime by four hours (see Figure 5.3). Two devices can also be connected to each other, allowing different information to be shown on the display and providing redundancy. It can also be used to control an autopilot. The Garmin G5 can be purchased for certificated aircraft for around 2 600 USD and for experimental/LSA aircraft for around 1 590 USD (with G5 battery pack). (Garmin 2021)

Figure 5.3 Garmin G5 with backup battery (Garmin 2021)

The Garmin G5 instrument can be installed with a QTR instead of the analog indicators of the minimum equipment (see QTR in Appendix G). After submission to the DAeC, an application for modification of the aircraft type-certificate data sheet must be made. In a technical bulletin from the manufacturer (Uniplanes GmbH), changes to the type-certificate data sheet are published. Supplemental entries of operating instructions for the Garmin G5 must be made in the aircraft flight manual. It may be required that the installation may only be carried out with other EFIS.

5.2.2 VP-X Sport

The VP-X Sport electronic circuit breaker system (Figure 5.4) is designed for microlight aircraft. The box has no moving parts and greatly simplifies the wiring of aircraft systems. All connected systems can be digitally monitored for power consumption and switches can be freely assigned. When a fuse blows, it can be digitally turned back on via the connected controls (Advanced Flight Systems, Dynon, Garmin, Grand Rapids Technologies, iLevil Avionics and MGL Avionics). The VP-X Sport Box also provides flap control, trim control, lights wig-wag control and starter protection (see Figure 5.5, *Function Diagram*). The price of a VP-X sport box is about 1 700 USD. (Vertical Power 2021)

Figure 5.4 VP-X installed in the Flywhale

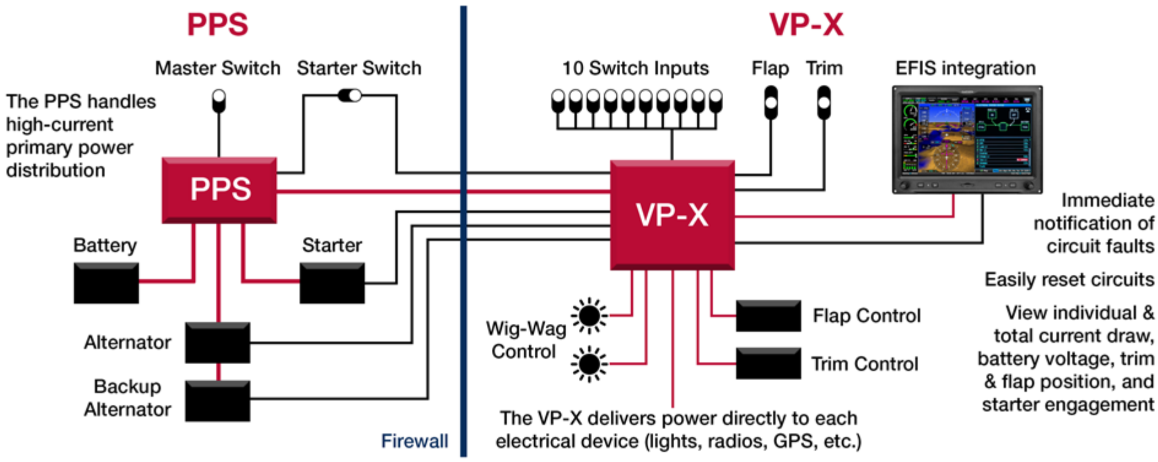


Figure 5.5 VP-X Sport Function Diagram (Vertical Power 2021)

Figure 5.10 (left) shows a panel of a Flywhale without VP-X where all required fuses are installed in the panel. Together with the associated switches, such a panel can quickly lose its overview, not to mention the quite complex wiring on the back. Figure 5.10 (right) shows a Flywhale panel with a VP-X box that has almost no fuses. Since the aircraft can be in the water for a long time, there is a bilge pump on board that can be turned on without switching on the aircraft's main circuit. For this purpose, there is a fuse in the panel. The flap controller is also not required as the VP-X box can also control this directly with the rocker switch (see Figure 5.5).

5.2.3 iLevel 3 AW

The iLevil 3 AW is a small compact unit that transmits data directly to a phone or tablet via wifi. It measures ADS-B weather and traffic information, GPS navigation, pitot connection (indicated airspeed and altitude), AHRS (Attitude and Heading Reference Systems) and data logging (.fdr .gpx and .csv). It has an internal battery, so in the event of a total electrical failure, the iLevel can continue to operate. On the device, there is a DB15 connector where all other devices like the VP-X box can be connected via the RS-232 (TX and RX) (see Figure 5.6). The device costs about 1 490 USD (Level 2021)

Figure 5.6 iLevel 3 AW (Level 2021)

In Chapter 4.7, the RS-232 standard is described in more detail. iLevel is designed to read as many devices as possible and has 4 RS-232 (RX and TX) ports (8 pins). The data can then be displayed via WiFi in an APP on the tablet or smartphone (see Figure 5.25 tablet screens).

5.2.4 FLARM – PowerFLARM Fusion

PowerFLARM Fusion is a small 250 g device that has Bluetooth and WiFi to display the data directly on a tablet or smartphone in APPs (Air Navigation Pro, SkyDemon, ForeFlight, EasyVFR, iPilot and XC Soar). It supports ADS-B and FLARM and has a DB9 connector with an RS-232 data link (see Figure 5.7 and for further information about RS-232 Chapter 4.7). The price is about 1 890 EUR. (Flarm 2021)

Figure 5.7 PowerFLARM Fusion (Flarm 2021)

5.2.5 Dynon Avionics

The SkyView HDX is an Electronic Flight Instrument System (EFIS) from Dynon for experimental and light sport avionics (see Figure 5.8). The displays (7" or 10") are compatible with all existing SkyView (SV) components but need additional modules to function. For primary flight instruments, an ADAHRS module (SV-ADAHRS-200) is needed, for engine monitoring a EMS module (SV-EMS-220) and for moving maps a GPS module (like the SV-GPS-250) is needed. The whole Dynon SkyView product range is very well connectable so that integrated devices can be shown on the display and you have a coherent system. For example, there is a VHF COM radio where the control panel can be mounted horizontally or vertically. A dual-band ADS-B traffic and weather receiver (SV-ADSB-427) or a Mode S transponder with ADS-B functions. A backup battery for the SkyView system (display and attached devices) provides at least one hour of runtime (one backup battery per display). (Dynon 2021)

The Flywhale aircraft, like many other microlight aircraft, uses a small bent plate on the leading edge of the wing which is pushed up just before a stall and triggers a switch. In the cockpit, a warning light and a warning sound come on to bring the aircraft into a safe position before the stall. Dynon offers a pitot probe (optionally heated) which has a second opening for an Angle Of Attack (AOA) sensing. This allows the SV display or EFIS to continuously indicate the angle of attack (see Figure 5.9).

Figure 5.8 SkyView HDX 10’’ and 7’’ (Dynon 2021)

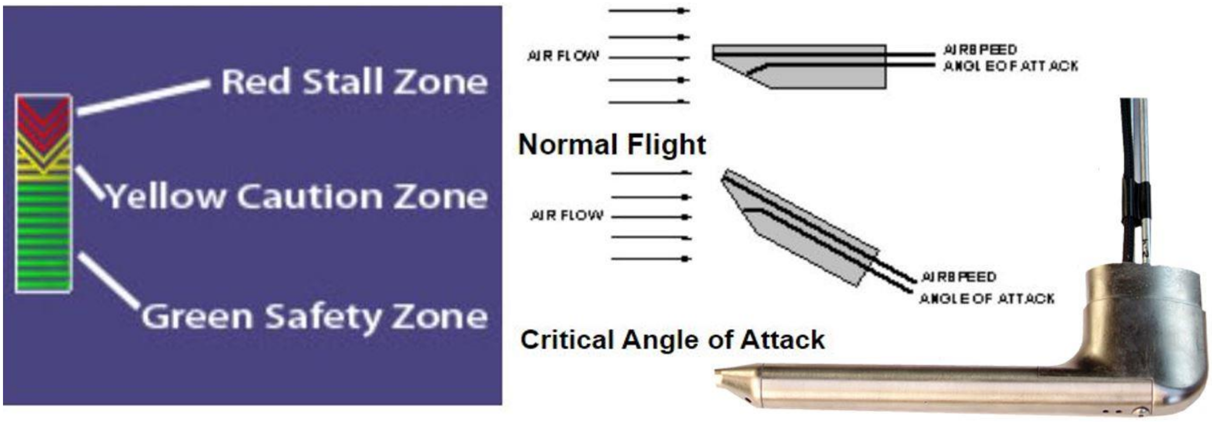


Figure 5.9 AOA principle and indicator on EFIS display (Dynon 2021)

In the following Chapter, Figure 5.12 shows a Dynon installation in a Flywhale aircraft where two backup batteries are installed at the bottom left of the picture, and the Dynon EMS controller and mode-s transponder are installed at the bottom right of the picture. At the top of the picture two 10" SkyView HDX displays can be seen from behind (see Figure 5.12).

5.2.6 Avionics and Wiring in the Microlight Aircraft

In Chapter 5.2 *Avionics* electrical components and systems that are used in the Flywhale Aircraft were introduced. Figure 5.10 (left) shows an older Flywhale panel where no VPX fuse box was used and therefore all fuses are located in the panel. Together with buttons, the layout can quickly become unclear and space can also become tight. Figure 5.10 (right) shows a newer panel where a VPX fuse box has been used and therefore almost all fuses have disappeared from the panel. Only the bilge pump on board still has a fuse in the panel so that it can operate continuously without switching on the main circuit when the aircraft is parked in the water. The flap position no longer needs a controller as this task is taken over by the VPX box and only one rocker switch is needed. The panel has become simpler and the installation effort of the panel has also been reduced. Fuses that are to be switched off or switched on again can be controlled via the connected EFIS (usually touch display) (see Figure 5.25, marked green in the lower right corner of the tablet: Fuse on).

Figure 5.10 Flywhale panel with fuses (left) and Flywhale panel with VPX box installed (right)

Despite the panel wiring simplification provided by the VPX fuse box, many cables go from the wings and the rear fuselage section forward into the cockpit to various electronic devices. In Figure 5.11 (left), the wiring harness has begun to be pulled into the aircraft from individual cables with additional cable length. Once the exact location of the electrical components has been determined, the cables can be shortened and crimped to be connected to the correct pin in the respective connector in the device. Figure 5.11 (right) shows the completed Flywhale cockpit.

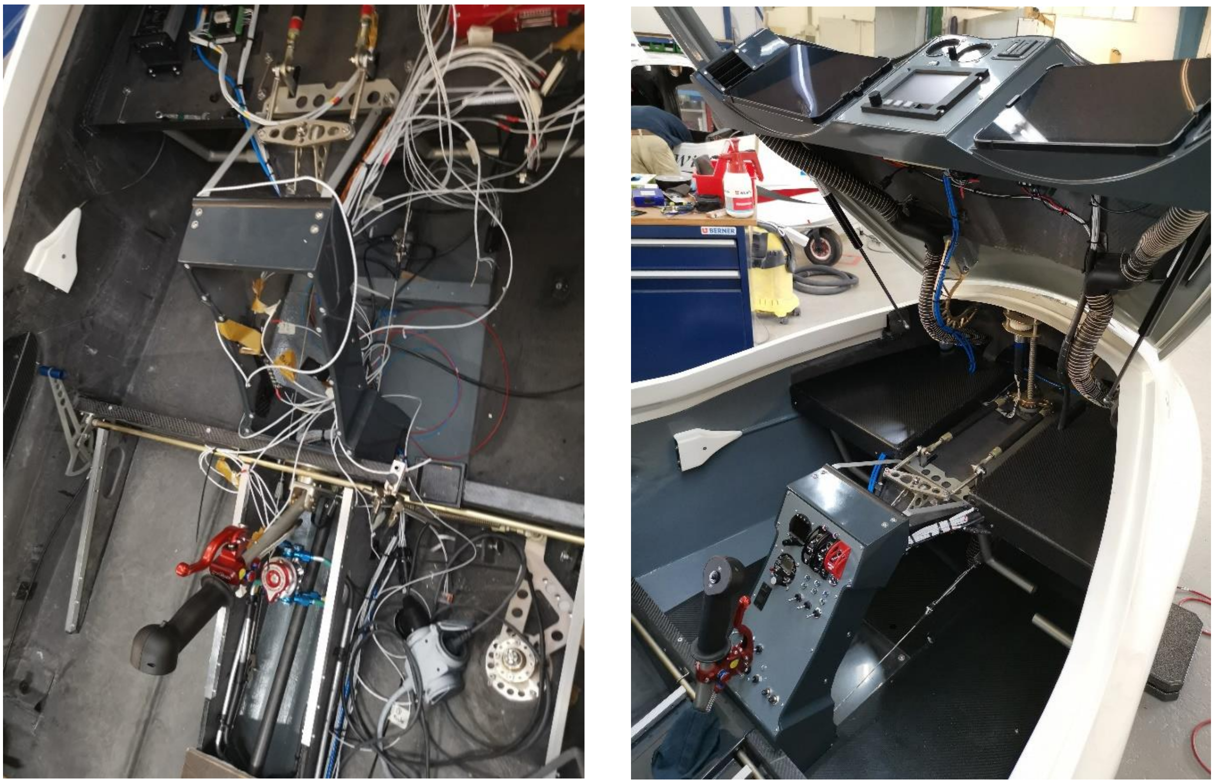


Figure 5.11 Flywhale cockpit before panel installation without connectors (left) and after finished installation (right)

Flywhale aircraft are prototypes as every aircraft built to date has been completely different in terms of avionics. The choice of avionics is made by the customer, which means that different brands with different numbers of devices and different positions in the aircraft have to be wired. This means that a new wiring harness has to be created for each aircraft, which increases the time and effort required for wiring and testing. Figure 5.12 shows a cockpit of a Flywhale aircraft without cover with a Dynon installation. In the lower-left of the picture are two backup batteries that keep the Dynon avionics system running for one hour in case of a power failure. At the bottom right of the picture, the Dynon EMS controller and the transponder are installed. At the top of the picture, the two SkyView HDX displays can be seen from behind. The whole system is connected in such a way that if one display fails, the data can be displayed on the other display, providing a sufficient level of redundancy.

Figure 5.12 Flywhale cockpit without cover, equipped with Dynon components

On the website Kitplanes, Bruch 2014 describes in *All About Avionics* his way of wiring his aircraft. He has a few tips and good pictures of the wiring. For example, he uses an aluminum profile for many cables, which can be removed later when the harness is finished and labels all his cables with shrink tubes. The cockpit panel is also first completely built up on a bench to be able to work from both sides and to be able to set up and program the system (see Figure 5.13 and 5.14). A VPX fuse box is also used (see Chapter 5.2.2). (Bruch 2014)

Figure 5.13 Mounted cockpit panel on a bench, front side (Bruch 2014)

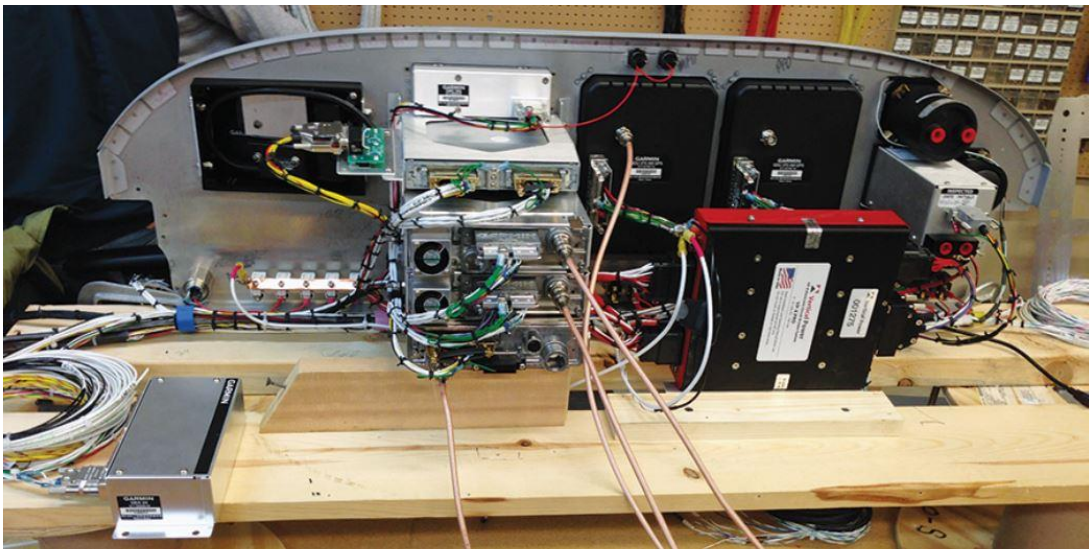


Figure 5.14 Mounted cockpit panel on a bench, back side (Bruch 2014)

5.3 Fuel Level Indicator

5.3.1 Problem and Task

In the Flywhale aircraft (FW11) is a fuel level sensor installed which outputs a voltage from 0 to 5 volts. This must be calibrated when empty and when full. In between the intermediate values are then output linearly via the signal out cable. The sensor is supplied with the aircraft's main voltage and can be between 9-34 volts (see Figure 5.15).

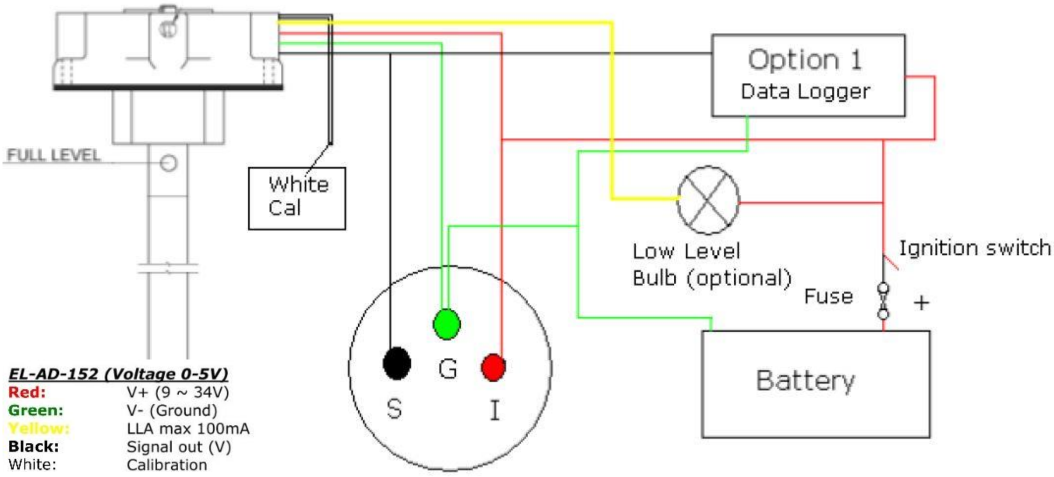


Figure 5.15 ATL Fuel Level Sender EL-AD-152 (Molloy 2012)

In order to get a fuel level display as soon as possible, a fuel display from Flybox (from Italy) was purchased and installed after a few days (Figure 5.16). In the Flybox fuel display, you can then fill the tank step by step (for example in 5 liter steps) to compensate the non-linearities of the tank (Figure 5.17). Unfortunately, it failed at the calibration, because always only 4.3 volts were displayed although the tank was still empty. After several emails to the manufacturer, it turned out that the output power of the ATL fuel level sender is too weak to overcome the internal resistance of the FlyBox fuel display. This could be due to the output power from the signal out of the ATL tank level transmitter is not strong enough, or the distance between the sensor and the FlyBox fuel display is too long, respectively the cable diameter is too small, resulting in power losses in the system.

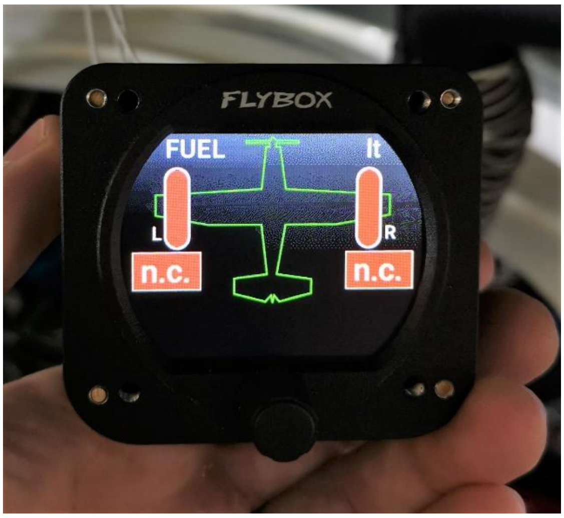
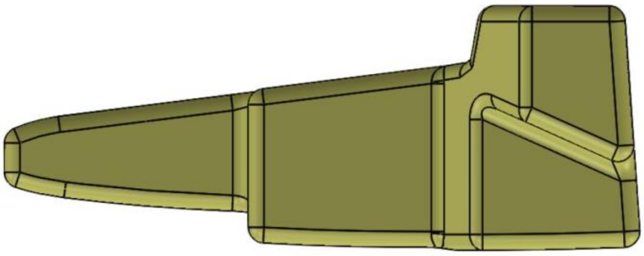


Figure 5.16 Flybox Omnia57 Fuel L-P



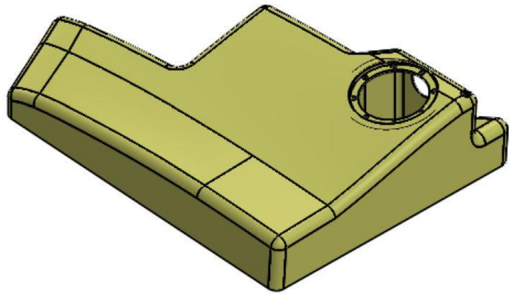


Figure 5.17 Flywhale tank container with 53 l volume

To solve the problem, I was given the task of developing a fuel level indicator that would process the incoming signal from the fuel level sender to drive the RP3 LED indicator that was available in the workshop (see Figure 5.18). Since the LED indicator cannot be connected directly to the fuel level sender, an intermediate module is to be used for data processing and for taking into account the non-linearities of the tank (calibratability).

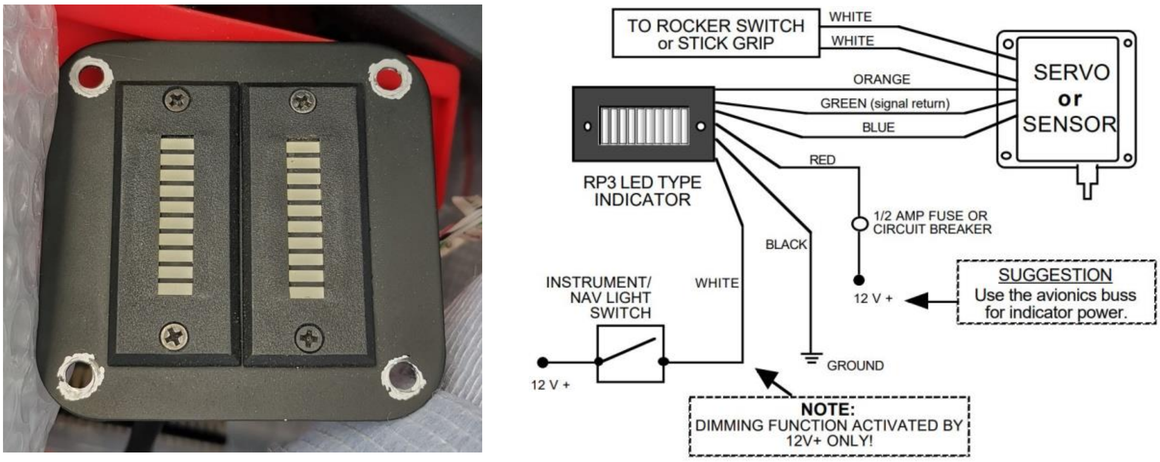


Figure 5.18 (left) RP3 LED indicator and (right) RP3 wiring (Ray Allen 2021)

Since I already have a lot of experience from previous projects with Arduino, I decide to use the Arduino Uno platform. The plan is to connect the signal output cables of the left and right tanks to the Arduino Uno, write a code for the required task and then control the RP3 LED indicator. When connecting the components I quickly noticed a problem. The Arduino Uno board has 6 analog inputs, but unfortunately no analog output pins. It can only output a digital PWM (Pulse Width Modulation) signal (see Chapter 4.8). So I could read a voltage with the analog inputs, but unfortunately, I could not control the RP3 LED display with the modulated voltage.

The keyword is DAC (Digital Analog Converter). There is already a board on the market "MCP4725 Breakout Board 12 bit DAC" (Figure 5.19), but the fuel level indicator has to be finished as soon as possible so that the already finished aircraft can be delivered to the customer. After a little research, I figured out how to build a DAC myself with simple electrical components I had at home (home office due to the corona pandemic). A capacitor and a resistor.

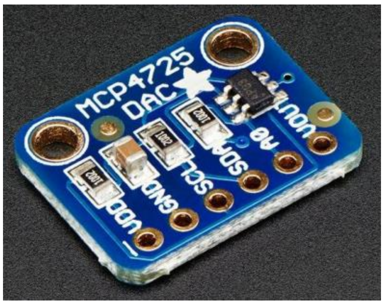


Figure 5.19 MCP4725 breakout board 12 bit DAC (Adafruit 2021)

5.3.2 PWM and Electrical Understanding of a Low-Pass Filter

To understand the problem as well as the solution we have to go a bit deeper into the matter of the PWM output signal of an Arduino. In the code, the PWM signal is defined by analog write from 0-255. This wave signal (Figure 5.20) must now be smoothed with a filter so that a pure DC signal of 0 to 5 volts remains. Since we do this with the low-frequency average and not with the high-frequency pulses, this is called a low-pass filter. Since filters are not ideal, it must be weighed, how far one should reduce the high frequencies, since the response time of the output voltages reacts slower and slower to the PWM signal and therefore becomes more and more sluggish. The more complex a low pass filter is built, the better the response time and the attenuation. (Mike Cook's 2021)

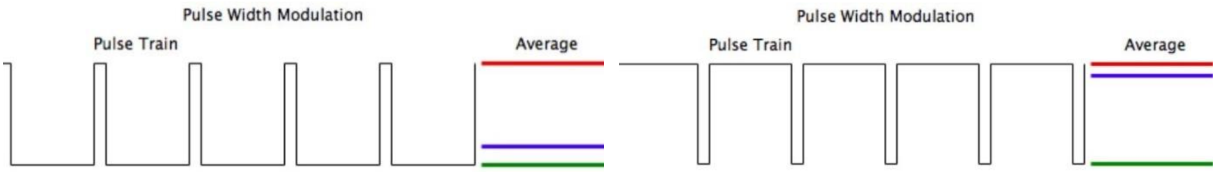


Figure 5.20 PWM pulse train (left side low, and right side high average) (Mike Cook's 2021)

A low-pass filter is characterized by two main features: cut-off frequency and slope. The frequency cutoff is the point at which the output drops by half compared to the input. This point is also called the 3 dB point. The slope is the rate at which the output falls off after the 3 dB point in the unit dB per octave where an octave is a duplication of frequency. The slope is also called order. A first order filter has 3 dB per octave. A second order filter drops at 6 dB per octave and a third order filter drops at 9 dB per octave. Since each 3 dB represents a halving of the signal, a third-order filter has one-eighth the input amplitude. A first order filter consists of a capacitor and a resistor connected in series. The value of the resistor depends on the value of the capacitor (in farads) and the value of the frequency (in Hz). This is called "capacitive reactance" X_C. The wiring of a potential divider and a frequency dependent potential divider are shown in Figure 5.21 and 5.22 with the corresponding Formula 5.2 and 5.3. (Mike Cook's 2021)

$$X_C = \frac{1}{2 \cdot \pi - F \cdot C} \tag{5.1}$$

with $X_C = R = R_2$

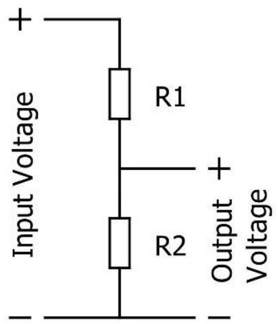


Figure 5.21 Potential divider (created with QElectroTech 2019)

$$\text{Output Voltage} = \frac{\text{Input Voltage } \cdot R_2}{R_1 + R_2} \tag{5.2}$$

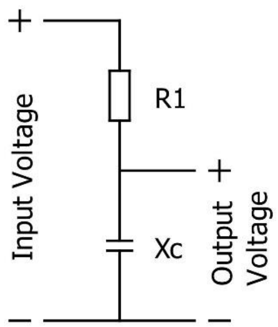


Figure 5.22 Frequency dependent potential divider (low pass filter) (QElectroTech 2019)

$$\text{Output Voltage} = \frac{\text{Input Voltage } \cdot X_C}{R_1 + X_C} \tag{5.3}$$

When designing a simple filter, it is not the absolute values that are important, but the ratio of the capacitor to the resistor. It depends on the application how fast the output signal can be changed. The cutoff frequency should then be set to a quarter. In my case of the fuel level indicator, one change per second is sufficient, so the design cutoff frequency is set to a quarter of a second or $F = 4$ Hz.

$$R \cdot C = \frac{1}{2 \cdot \pi \cdot F} = \frac{1}{2 \cdot \pi \cdot 4\ \text{Hz}} = 0.0398 \tag{5.4}$$

From this, a suitable resistance to a capacitor can be found. I still had capacitors with 100 µF at home, which leads to the following resistance with Formula 5.4.

$$\mathrm{R} = \frac{0{,}0398}{100 \cdot 10^{-6}\ \mathrm{F}} = 398\ \Omega$$

I then chose a 470 Ω resistor with which the cutoff frequency with Formula 5.4 is

$$\mathrm{F} = \frac{1}{2 \cdot \pi \cdot \mathrm{R} \cdot \mathrm{C}} = \frac{1}{2 \cdot \pi \cdot 470\ \Omega \cdot 100\ \mu\mathrm{F}} = 3.3863\ \mathrm{Hz}$$

resulting in a response time of 1.2 seconds which is quite sufficient for the tank indicator.

The remaining ripple of the PWM signal can also be determined. The default Arduino PWM frequency is 490 Hz at 5 V, which with our 100 µF capacitor and Formula 5.4 results in a

$$X_C = \mathrm{R}_2 = \frac{1}{2 \cdot \pi \cdot \mathrm{F} \cdot \mathrm{C}} = \frac{1}{2 \cdot \pi \cdot 490\ \mathrm{Hz} \cdot 100\ \mu\mathrm{F}}$$

$$\mathrm{R}_2 = 3.2481\ \Omega$$

and with Formula 5.2 to a

$$\text{Ripple Voltage} = \frac{5\ \mathrm{V}\ \cdot 3.2481\ \Omega}{470\ \Omega + 3.2481\ \Omega} = 0.03432\ \mathrm{V}.$$

The ripple voltage is small enough for the RP3 LED indicators. (Mike Cook's 2021)

5.3.3 Wired Hardware for Fuel Level Indicator

Figure 5.23 shows the finished hardware. The cables are soldered directly to the board and hot glued. This reduces the risk of vibration brittle cracks in the solder and improves the moisture resistance. When installed, the entire board was put into a large heat shrink tube.

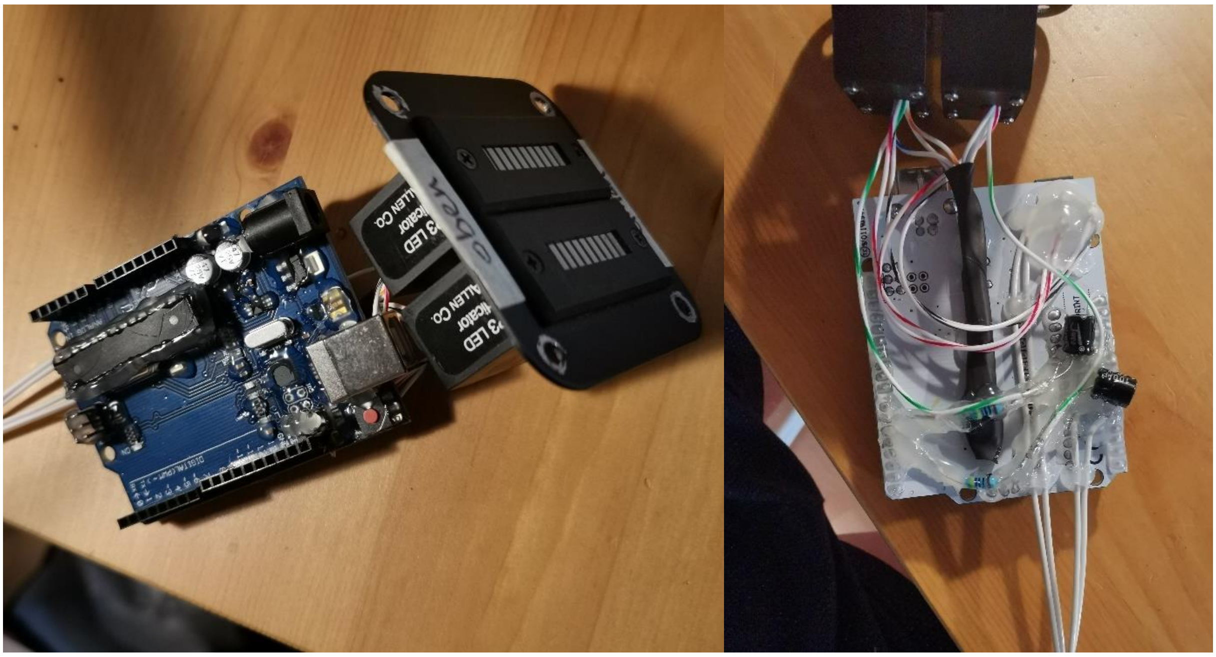


Figure 5.23 Finished hardware of the fuel level indicator

Figure 5.24 shows the wiring and connection of the different components.

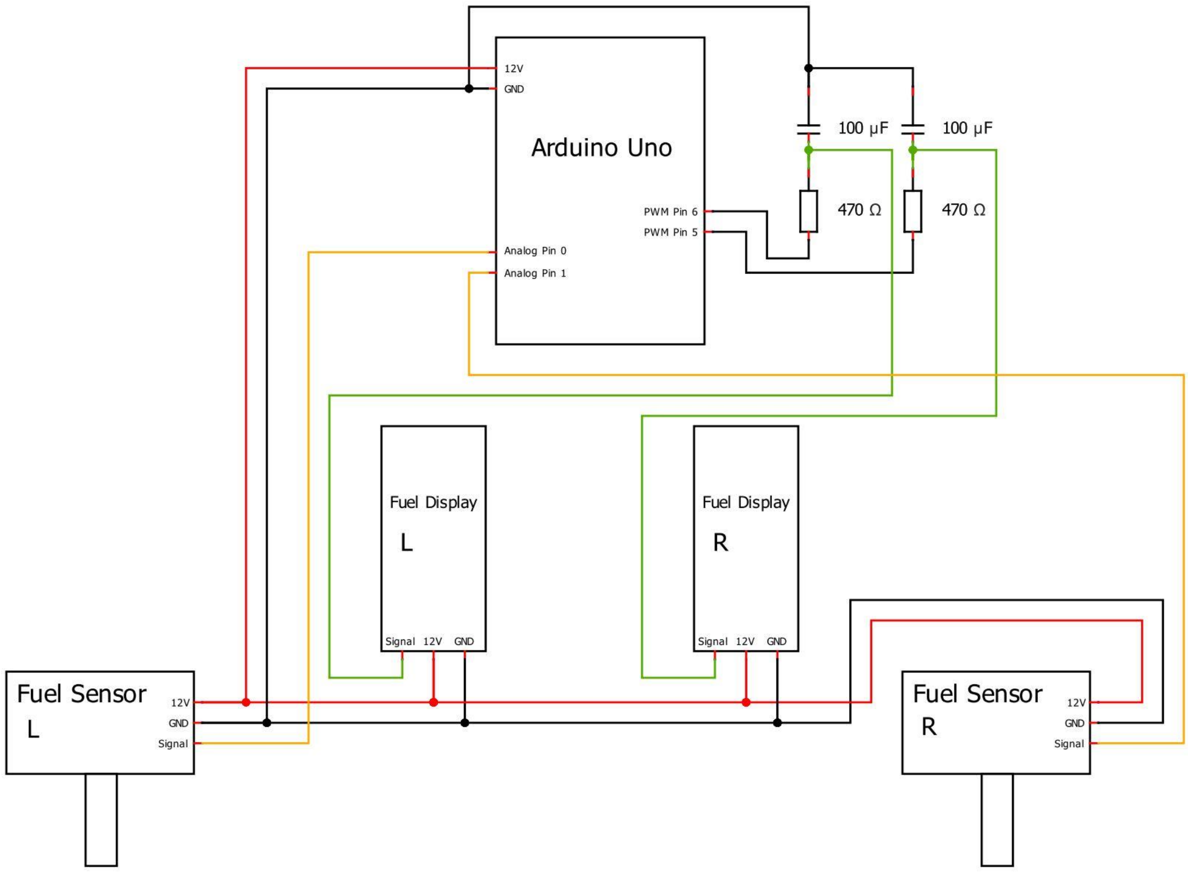


Figure 5.24 Wiring of the fuel level indicator (created with QElectroTech 2019)

During the development of the fuel level indicator, the Arduino code was written and adapted to the configuration data. Jan Kaminski (engineer from uniplanes GmbH) has measured the two fuel tanks on the Flywhale aircraft with a voltmeter on the signal out cable in 5 liter steps (see Table 5.1). These voltages ended up as calibration input in the Arduino code. Then I tested with an adjustable power supply (LW-K3010D) at which voltage which LED in the RP3 indicator is indicated.

Table 5.1 ATL signal out calibration Voltages of the left and right tank

Added fuel [l]	Calibration Voltage left tank [V]	Calibration Voltage right tank [V]
0	0,27	0,27
5	0,83	0,61
10	1,09	1,14
15	1,56	1,62
20	2,02	2,05
25	2,47	2,50
30	2,93	2,95
35	3,41	3,41
40	3,89	3,89
45	4,44	4,44
50	4,85	4,83

5.3.4 Software for Fuel Level Indicator

In Appendix E the complete Arduino code can be seen. The code also has a voltage offset of 0.035 V so that the LEDs do not flicker back and forth in case of vibrations or slight fluctuations of the level. Until the end, it was possible to read out via the serial print (serial monitor in Arduino) which LED is controlled by what value which was very practical for testing and the last calibration (see Figure 5.25). Also in the last calibration step, the values for the full state were adjusted in the code. The serial print was then commented out of the code.

Figure 5.25 Cockpit of Flywhale Aircraft – last calibration of fuel level indicator in arduino code

6 Future of Avionics

Microlight aircraft are equipped with different avionics depending on the customer's requirements. People who fly microlight aircraft are usually not dependent on the aircraft. Therefore, flying microlight aircraft is more of a recreational activity. Flying is therefore a form of entertainment and aerial fun. Design and technical gadgets are therefore main reasons for choosing the desired avionics. After a few years, new avionics may need to be installed to replace old components or to complement the existing system. To meet the requirements in the best possible way, a modular avionics system with minimal installation effort is desired. Switches, knobs and fuses are increasingly replaced by components that can be operated via redundant touch panels, where the pilot decides which data he wants to be displayed. (see Figure 6.1).

Honeywell introduces a new scalable avionics system "Anthem". It is always connected to the cloud when possible so that all kinds of data can be loaded directly into the avionics system and pilots can upload their flight plans to the flight management system before landing at the airport. It has a landing assistance system that helps the pilot to always perform soft landings by hand and many 3D functions like 3D maps near the airport which simplifies IFR approaches. In general, the new Anthem avionics system makes flying easier and more fun because it intelligently suggests things to do at the right time. It is easy to use and looks like a normal tablet, making tasks intuitive and easy to perform. The most important feature of Anthem, however, is its scalability to different types of aircraft. The number of touch screens can also be varied (see Figure 6.1). (Plane 2021)

Figure 6.1 Anthem, scalable avionics system from Honeywell (Plane 2021)

Athem is not an avionics system for smaller aircraft like microlight aircraft, but Athem shows that fully scalable networked avionics is the future.

In the paper of Hope 2011 a novel patch antenna is presented which can be created using a genetic algorithm. This can be applied directly to the aircraft skin as a configurable antenna (see Figure 6.2).

Figure 6.2 Genetically evolved patch antenna (Hope 2011, page 285)

The algorithm can be used to create antennas that are precisely designed for ZigBee use and the installation position in the aircraft. These can be written directly to the aircraft's skin (Hope 2011). Electrical components connected to ZigBee can be located in the left-wing, right-wing or fuselage and have an enhanced radio direction to the fuselage center or cockpit. The radio mesh network cannot be disconnected or suffer cable breakage or short circuit. Redundancy could also be implemented with double ZigBee transmitters. However, the many advantages can only be realized if a complete avionics system is equipped with the wireless standard. For first tests, converters like ZigBee to RS-232, ZigBee to RS-428, ZigBee to USB or ZigBee to Ethernet could be used to connect different systems via radio. Later, the electrical avionics component should support ZigBee by itself for an improved fail-safe data connection.

NASA uses an international Deep Space Network (DSN) to maintain a data link at all times during space missions or with interplanetary robots. Three main sites in Goldstone (California), Madrid (Spain) and Canberra (Australia) ensure that a connection can always be maintained as our planet spins. The DSN also provides data for radar astronomy and radio astronomy. (NASA 2020)

In a broader sense, the DSN is a huge global mesh network with 3 nodes located on Earth and other nodes in space. If the data connection becomes weaker and weaker (node disappears behind the horizon), a new node is selected for data transfer.

Future Combat Air System (FCAS) was launched as a European project by Germany and France in 2017. As an industry leader, Airbus and Dassault Avioation agreed to cooperate on this project. (Airbus 2022)

FCAS is supported by a Multi-Domain Combat Cloud (MDCC), which is a decentralized cyber-resistant mesh network (see Figure 6.3). It connects nodes across all forces and enables real-time information sharing. By providing specific information at a specific location, the effectiveness of the MDCC network is increased. The cloud enables FCAS to deploy manned and unmanned units efficiently and effectively in all domains. The Observe, Orient, Decide and Act (OODA) cycle will also be enhanced by the MDCC through the increasing use of Artificial Intelligence (AI). (Airbus 2022b)

Figure 6.3 FCAS (MDCC) mesh network (Airbus 2022)

As shown in Figure 6.3, with a complete decentralized mesh data connection, information is shared that also increases the redundancy of individual systems to a high degree. Drones and remotely piloted missiles could continue to function properly in the event of a single system failure such as GPS position data or magnetic sensor failure by using data from the next node in the data mesh. Radar surveillance material and other data are collected by many devices and can be combined by MDCC to provide a high-resolution image.

In FCAS, data processing plays a very important role through the data cloud collection of all networked objects. These are evaluated to the extent that only the useful data are sent to a manned jet, for example. These tasks are performed by agorythms with artificial intelligence. Position and control decisions of drones can also be given by FCAS. (Koch 2019)

Air traffic data could be uploaded from all aerial vehicles to the cloud, so that air traffic is displayed before the aircraft's own sensors can detect it. Even if anti-collision systems fail, air traffic can still be displayed and the own position data can still be uploaded. The example shows a functional improvement and redundancy increase of a single system.

As already mentioned at the end of Chapter 3.4, the use of Wireless Avionics Intra-Communication (WAIC) could save about 30 % in cable weight. As already explained in more detail in Chapter 4.10, the first steps for approvals and certification of WAIC are on their way.

Examples of WAIC controlled units are the release mechanisms of the oxygen masks, audio announcements and the screens with safety information for the passenger which will be controlled wirelessly. Through WAIC, it is also possible to quickly enable a new seating arrangement by moving the mounting locations of the electrical units (see Figure 6.4). (AVSI 2015)

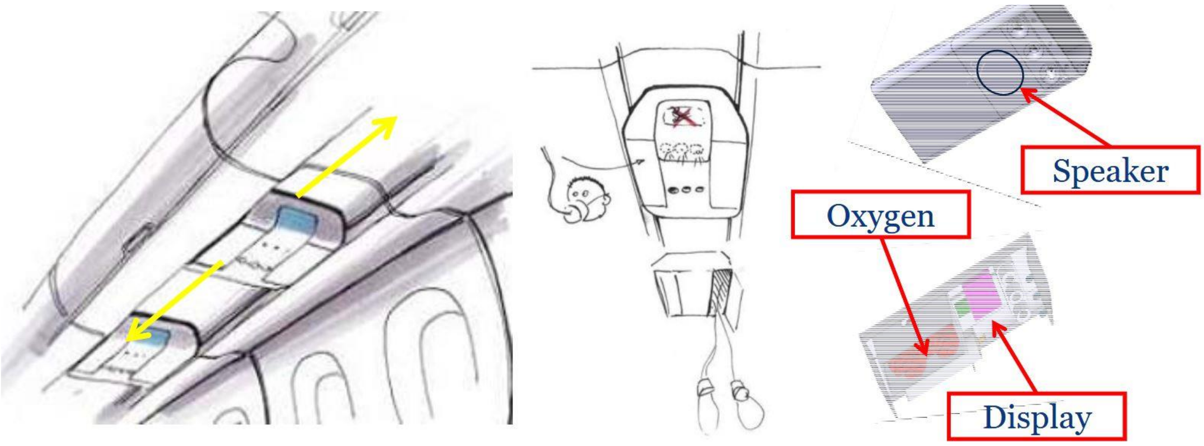


Figure 6.4 Wireless devices on the aircraft (AVSI 2015)

Modular adaptability through wireless data communication also enables better maintainability or replacement of individual components.

Anthem, as a non-military avionics system, provides the cloud interface that enables the benefits of a decentralized mesh network in the civil sector. It remains to be seen how long it will take for microlight aircraft avionics to use a mesh data connection to other aircraft, helicopters and drones, for example, to combine weather data and position data into a huge real-time database. Initially, it will probably be a modular avionics setup with a wireless data link to equip many different aircraft with a reliable modular system. Reliability through hardware simplification and minimal installation effort are of particular interest to small microlight companies, prototyping and for kit plane builders.

In January 2022, the entire Flywhale aircraft project (intellectual property and production assets) was sold by Uniplanes GmbH to Dornier Seawings GmbH, which will produce and sell the aircraft under the name DS-2C. Dornier Seawings also wants to use the DS-2C as a research platform to test new systems and to present in the future the first amphibious VTOL under the name DS-2C-X-eVTOL (see Figure 6.5). (Dornier 2022)

Figure 6.5 Future research platform DS-2C-X-eVTOL (Dornier 2022)

Dornier Seawings intends to enter the Advanced Air Mobility (AAM) segment with the DS-2C in order to offer customers a more economical and environmentally friendly product. In Dornier's history there are already many amphibious aircraft, whose tradition was continued with the Seastar with a maximum take-off weight of 5100 kg and up to 12 passengers. The Flywhale aircraft (DS-2C) has now been handed over to Dornier Seawings, which already has a lot of experience in aviation. (Dornier 2022)

7 Summary and Conclusion

Aircraft avionics equipment is becoming more digital to make flying easier and safer for pilots. If, as planned, the international decision is made to navigate only according to true north and no longer according to magnetic north, more and more analog instruments will be replaced by digital avionics. This also affects all Microlight aircraft, which are already equipped with large screens according to customer demands. Due to this change, well-functioning communication between electrical components is more important than ever before. Depending on the requirements and the amount of data to be transmitted, various options are available in terms of the transmission channel (electrical cable, optical fiber or radio transmission) and protocols. Especially for the power supply of electrical components, conventional copper cables can have a high weight, which can be replaced by the lighter and cheaper material aluminum. The material properties, however, must be taken into account, especially at connections and crimps. The current wiring in large commercial aircraft has been simplified by BUS systems, but still represents a big part of the installation effort and the weight of the aircraft (>2.6 %). There are many different airborne protocols for different applications such as ARINC 429, 629, 664, 708, 812, 825, 826, and MIL-STD-1553, MIL-STD-1773 as well as IEEE 1384 (FireWire) and ASCB to name a few which are explained in more detail in this paper. In the protocol functionality are differences in cable and connector characteristics and differences in the possible network connection types (topologies). There are also many differences in terms of transmission speed and redundancy possibilities of the different standards.

The development of WAIC for large aircraft, which enables wireless communication with cabin units, could be the introduction of wireless communication in the aviation industry. For microlight aircraft, the hardware is far too expensive to use ARINC protocols and WAIC is not suitable as a cabin communications system. Many electrical avionics devices rely on the RS-232 data transfer standard to exchange ADS-B and GPS data with ADAHRS and other devices. Since microlight aircraft do not require any further certification for equipment in addition to the minimum equipment (airspeed indicator, altimeter, turn coordinator and heading indicator), a wireless data link could be established with less effort. Studies on the use of ZigBee and other radio links on 2.4 GHz have already investigated this in an aircraft environment. An algorithm generated patch antenna was presented, which can be printed on the aircraft skin.

The current state of avionics with examples of equipment and conventional wiring is shown using the Flywhale microlight aircraft as an example. The manual effort of wiring a harness is very large every time a prototype is built such as the Flywhale aircraft. With a standardized wiring protocol or the use of ZigBee sensors, the manual effort could be greatly reduced. A fuel level indicator is planned, built and tested from scratch. The microcontroller Arduino Uno serves as data processor where the resistance of the fuel level sensor is used as input and a PWM signal is output. This is converted by a low-pass filter into a smoothed DC signal for the control of a LED indicator. A calibration due to the nonlinearity of the fuel tank is implemented in the Arduino software. If standardized protocols were used by the fuel level sender and LED indicator, the effort required for the connection could have been significantly reduced. If the LED indicator and the fuel level sender supported ZigBee, there would have been no hardware effort; the system would only have to be configured.

The future of avionics is a fully integrated data network across devices and aircraft, which is made possible in the military sector by FCAS and MDCC. In the civil sector, a new avionics system Anthem from Honeywell is presented, which also has a cloud connection. Modularity is also a major strength of Anthem, which means that the system can be integrated into a wide variety of aircraft types. In the future, WAIC will enable a reduction in cables, easier maintenance and better interchangeability of cabin units such as speakers and displays.

The trend shows that future avionics will use wireless technologies and can be modularly adapted to the aircraft or customer's requirements. Cloud networked avionics systems can be enhanced by real-time databases and increase the level of redundancy. However, it will be a few years before this technology enters the microlight sector and becomes well established.

When looking for a suitable communication system for microlight aircraft, there are several possible solutions. Small microlight aircraft companies, prototypers or kit plane builders will continue to use the RS-232 ports provided by the electrical components to communicate with the equipment. Components such as sensors will also continue to be wired point-to-point to the respective processing system. For larger microlight companies in series production, it is possible to save cable, weight and installation effort by designing a wiring harness with BUS systems. With CANaerospace, many sensors and control components (for example trim servos, landing gear and landing gear doors to ADAHRS) can be connected to a BUS which runs through the microlight aircraft. A very detailed customization and programming of CANaerospace has to be done once and can then be used for all aircraft of the same type. For smaller tasks, a simple RC protocol such as SBUS can also reduce the amount of wires needed to control trim tabs and other servo motors.

A complete modular avionic system for microlight aircraft with a wireless data communication to sensors has yet to be developed. First steps in this direction have already been made, but a breakthrough will only be achieved when a larger company creates and sells a complete product suite. ZigBee is a very good transmission protocol because of the high bandwidth of 250 kbyte/s and the self-regulating and self-repairing mesh network. Therefore, there is great potential in ZigBee as a wireless data protocol for future avionics applications in microlight aircraft.

List of References

ADAFRUIT, 2021. *MCP4725 Breakout Board – 12-bit DAC w/12C Interface*. Adafruit
Available from: https://www.adafruit.com/product/935
Archived at: https://perma.cc/VD4X-Q6WC
Datasheet available from: https://cdn-shop.adafruit.com/datasheets/mcp4725.pdf
Archived at: https://perma.cc/L32T-8BHK

AIRBUS, 2022. *Future Combat Air System (FCAS)*. Airbus
Available from: https://bit.ly/3L2lYhB
Archived at: https://perma.cc/8MM6-Q9B4

AIRBUS, 2022b. *Multi-Domain Combat Cloud*. Airbus
Available from: https://bit.ly/3AR6kRw
Archived at: https://perma.cc/SR7D-FQVC

ALEXANDER, 2017. *Die verschiedenen Arten von Netwerktopologien.* 6004 Luzern, Swiss: Swissns GmbH. 2017-06-03
Available from: https://bit.ly/34U4Vhb
Archived at: https://perma.cc/TL8H-4Y86

ANTONSUSI, 2019. *File: Periodensystem Einfach.svg.* Wikimedia, 2019-10-06
Available from: https://bit.ly/3qEOePa
Archived at: https://perma.cc/2J4P-T78T

AVIFTECH, 2016. *Testing Avionics Ethernet Networks & Switches.* In: Youtube. 2016-10-20 Omaha, USA: Avionics Interface Technologies
Available from: https://youtu.be/PNAvGVv0qQ8
Archived at: https://perma.cc/4Z3Y-HJXM

AVSI, 2015. *Wireless Avionics Intra- Communications (WAIC).* ICAO Regional WRC-15 Preparatory Workshop Cairo, Egypt: Aerospace Vehicle Systems Institute. 2015-02
Available from: https://bit.ly/3GYnLlw
Archived at: https://perma.cc/5EHX-9YD2

BADZIONG, Klaus, 2021. *SUMD, SUMH Protokolle…* [Email: klaus@graupner-service.de] Graupner Service Center Deutschland UG. 2021-12-20

BOYS, Robert, 2012. *CAN Primer.* bob.boys@arm.com V 4.0 ARM
Available from: https://bit.ly/3I8yWYI
Archived at: https://perma.cc/CW87-A7XX

BRUCH, Stein, 2014. *All About Avionics: Panel Upgrades – Wiring your panel.* Kitplanes. Minnesota, USA: SteinAir (Avionics Shop). 2014-08-21
Available from: https://www.kitplanes.com/all-about-avionics-8/
Archived at: https://perma.cc/3T8V-JSPU

CARLSON, A. Bruce and Paul B. CRILLY, 2010. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication.* Fifth Edition. New York, NY 10020: McGraw - Hill Education. ISBN 978-007-126332-0.
Available from: https://bit.ly/32FK945
Archived at: https://perma.cc/BQ9Q-JN9D

CHAI, Wesley and Alissa IREI, 2021. *Netzwerkprotokoll.* TechTarget, ComputerWeekly
Available from: https://www.computerweekly.com/de/definition/Protokoll
Archived at: https://perma.cc/2QQG-GJPF

CiA, 2003. *CANopen: Framework for CANopen Managers and Programmable CANopen Devices.* CiA Draft Standard Proposal 302. CAN in Automation e.V. Vers. 3.2.1. 2003-04-09
Available from: https://bit.ly/3IhqJSK (Closed Access)
Archived at: https://perma.cc/CF88-CP7A

COMMERCE, 2016. *United States Frequency Allocations: The Radio Spectrum.* U.S. Department of Commerce. National Telecommunications and Information Administration.
Available from: https://bit.ly/3IsoZ8P
Archived at: https://perma.cc/W9CT-BH9L

CONDOR, 2000. *ARINC Protocol: Tutorial.* Santa Barbara, USA: Condor Engineering, Inc. 2000-06-07
Available from: https://bit.ly/3FKGYWm
Archived at: https://perma.cc/4YLQ-364R

CORRIGAN, Steve, 2016. *Introduction to the Controller Area Network (CAN).* SLOA101B August 2002 - Revised May 2016. Dallas, Texas: Texas Instruments Inc.
Available from: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf
Archived at: https://perma.cc/KZF6-J987

DAWSON, J. F., D. C. HOPE, M. PANITZ and C. CHRISTOPOULOS, 2008. *Fly by wireless: Evaluation of Zigbee radio system for in vehicle connectivity.* In: Proceedings of EMC UK, Newbury, UK: University of York. Nutwood UK Limited, page 93-99
Available from: https://bit.ly/3GW1YdY
Archived at: https://perma.cc/6QLF-E4UV

DORNIER, 2022. *Dornier Seawings acquires Flywhale project from Uniplanes GmbH*. Wessling, Germany: Dornier Seawings 2022-01-19
Available from: https://bit.ly/3oY5uxI
Archived at: https://perma.cc/M6DB-LNDM

DYNON, 2021. *SkyView HDX: The Premiere SkyView Experience*. USA: Dynon Avionics
Available from: https://dynonavionics.com/skyview-hdx.php
Archived at: https://perma.cc/RP36-3XVB

ENZYKLOPÄDIE, 2020. *Universeller asynchroner Empfänger-Sender - Wikipedia.* Enzyklopädie, 2020-12-21
Available from: https://bit.ly/3E9JDrI
Archived at: https://perma.cc/Z98A-SAS6

ESA, 2012. *Mil-STD-1553.* ESA. Space Engineering Technology
Available from: https://bit.ly/3KjzCMS
Archived at: https://perma.cc/248U-ZNGF

EXCALIBUR, 2010. *AFDX / ARINC-664P7 Tutorial | Excalibur Systems.* In: Youtube. 2010-10-21
Available from: https://youtu.be/LxTVxz5ogag
Archived at: https://perma.cc/Q3B2-XHU9

FASSBINDER, Stefan, 2010. *Elektrischer Leiter – Alternativen zu Kupfer?* Dresden: Elektrotechnik 10/09, September 2005
Available from: https://bit.ly/3qyAJ23
Archived at: https://perma.cc/V2L5-NL4T

FIBERLABS, 2016. *Wavelength-Division Multiplexing (WDM).* U.S. and Canada: FiberLabs Inc., 2016-08-23
Available from: https://bit.ly/3t5WLvZ
Archived at: https://perma.cc/52RA-MVAZ

FLARM, 2021. *PowerFLARM Fusion.* FLARM Technology Ltd. 2021-07-20
Available from: https://bit.ly/3ANC89Z
Archived at: https://perma.cc/DR95-RUFX

FOSKO, 2010. *Optical Fiber Loss And Attenuation.* 269 Mavis Drive, Pleasanton, CA 94566 USA: Fiber Optics For Sale Co., 2010-09-20
Available from: https://bit.ly/3tdcedI
Archived at: https://perma.cc/97XJ-FEFS

FREDRIKSSON, Lars-Berno, 1995. *A CAN Kingdom.* Rev.3.01. Kinnahult Sweden: KVASER AB
Available from: http://www.kvaser.cn/wp-content/uploads/2014/02/ck301p.pdf
Archived at: https://perma.cc/A7GA-EGSF

GARMIN, 2021. *G5 Electronic Flight Instrument for Experimental/LSA Aircraft*. Garmin Ltd.
Available from: https://www.garmin.com/en-US/p/514383
Archived at: https://perma.cc/5H5R-4NBS

GMELIN, Philipp, 2021. *Requirements from National Regulations for Microlight Aircraft and Statistical Parameters for Preliminiary Sizing*. Hamburg, Germany: University of Applied Science, HAW. 2021-04-04
Available from: https://nbn-resolving.org/urn:nbn:de:gbv:18302-aero2021-04-04.014
Archived at: https://perma.cc/664H-7ZWM

GOLD 2021. *Aktueller Kupferpreis. Kupferpreis-Entwicklung 1995-2021.* Deutschland: Gold.de
Available from: https://www.gold.de/kurse/kupferpreis/
Archived at: https://perma.cc/L2VF-DLED

GOMES, Rui P. N., Jose E. G. OLIVEIRA and Francisco J. A. CARDOSA, 2010. *Integrating Zigbee and CAN Networks in Industrial Application.* Conference: Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010-06 IEEE. DOI: 10.1109/DCOSSW.20105593271
Available from: https://bit.ly/3AbgxaX
Archived at: https://perma.cc/AR4P-BHJC

HAAS, Michael 2016. *Airbus A 380-800.* Airplanefacts by airplanepics.ch
Available from: http://www.airplanepics.ch/Flugzeugtypen/A%20380-800.html
Archived at: https://perma.cc/8C4U-GXAH

HARA, Tenshi C., 2015. *Optische Übertragungstechnik. Grundlegende Prinzipien und Kopplungstechnologien.* Dresden: Technische Universität Dresden, Fakultät Informatik, Institut für Systemarchitektur, Professur Rechnernetze, 2015
Available from: https://docplayer.org/8549544-Optische-uebertragungstechnik.html
Archived at: https://perma.cc/7USV-X7YZ

HELUKABEL, 2015. *Aluminiumkabel & -leitungen: Verarbeitung und Anschlusstechnik.* Hemmingen, Germany: Helukabel GmbH
Available from: https://bit.ly/3nI6lSs
Archived at: https://perma.cc/W9XF-FU9X

HOPE, David C., 2011. *Towards a Wireless Aircraft: Propagation, Antennas and Radio Standrads.* PhD Thesis. York, UK: University of York. 2011-11-07
Available from: https://etheses.whiterose.ac.uk/1900/1/Thesis1x5spacing.pdf
Archived at: https://perma.cc/YLM3-QM7U

HORST, Ulrich ter, 2018. *Ein generischer CANaerospace-Knoten für redundante UAS-Architekturen basierend auf MicroPython.* Hamburg: Hochschule für Angewandte Wissenschaften
Available from: https://bit.ly/3FzRY8X
Archived at: https://perma.cc/A3J7-MVRH

ISIK, Yasemin, 2010. *ARINC 629 Data Bus Standard on Aircrafts.* Recent Researches in Circuits, Systems, Electronic, Control & Signal Processing. Eskisehir Turkey: Anadolu University, Civil Aviation School. ISBN: 978-960-474-262-2
Available from: https://bit.ly/3nE2BkP
Archived at: https://perma.cc/67XZ-Q74H

JANU, Premysl, 2014. *Analysis of CANaerospace Protocol Communication Quality in Avioation System.* Czech Republic: University of Devence. In: Advances in Electrical and Computer Engineering (AECE), Vol 14, Number 1, 2014, page 81-86. DOI: 10.4316/AECE.2014.01.013. ISSN: 1582-7445
Available from: https://bit.ly/33pruK5 (Closed Access)
Archived at: https://perma.cc/SMJ8-GJ97

JANU, Premysl, 2016. *Application for Designing Communication Algorithm Using CAN with CANaerospace Protocol.* Czech Republic: University of Devence. Advances in Military Technology. Vol. 11, No. 1, 2016-06-08
Available from: https://bit.ly/3qEkulH (Closed Access)
Archived at: https://perma.cc/6RQF-Q44X

KASSERA, Winfried, 2016. *Ultraleicht Fliegen Kompakt: Das Grundwissen zur UL-Lizenz.* 2. Auflage. Stuttgart: Motorbuch Verlag. ISBN: 978-3-613-03844-8

KAVIYARASU, A., 2012. *ARINC 429: Data bus for Civil Aircraft*. Chromepet, Chennai, India: Madras Institute Of Technology. Anna University
Available from: https://mitindia.edu/images/pdf/avionics_ppt/ARINC_429.pdf
Archived at: https://perma.cc/PSF4-CM5A

KLÜSTER, Jürgen, 2012. *CAN-based Protocols in Avionics.* Vector Informatik GmbH. Version 1.1 Application Note AN-ION-1-0104, 2012-04-12
Available from: https://bit.ly/3qxwgxX
Archived at: https://perma.cc/B2W4-KWDL

KNUEPPEL, Ralph, 2012. *Standardizationof CAN networks for airborne use through ARINC 825*. iCC 2012. CAN in Automation. Bremen, Germany: Airbus Operations GmbH
Available from: https://bit.ly/3qC59lo
Archived at: https://perma.cc/3GKC-B36M

KOCH, Wolfgang, 2019. *FCAS – Herausforderungen für Sensordatenfusion und Ressourcenmanagement.* Frauenhofer CPM. Frauenhofer-Institute für Kommunikation, Informationsverarbeitung und Ergonomie FKIE, Fellow des IEEE
Available from: https://bit.ly/3KWrNwW
Archived at: https://perma.cc/7AJ5-89G3

LEARMOUNT, David, 2021. *A Change of Heading in the Skies: The Navigation Flashlight.* 2021-03, page 7-9
Available from: http://ovn.at/fileadmin/flashlight/OVN-Flashlight_03-2021.pdf
Archived at: https://perma.cc/9RP3-7NXR

LEVIL AVIATION CORP., 2018. *iLevil 3 AW Installation Instructions*. 1704 Kennedy Point, Oviedo, FL32765 USA.
Available from: https://bit.ly/3g7ZQEp
Archived at: https://perma.cc/JG8Y-H5P7

LEVIL AVIATION CORP., 2021. *iLevil 3 AW*. 1704 Kennedy Point, Oviedo, FL32765 USA.
Available from: https://shop.levil.com/collections/all-products/products/ilevil-3-aw
archived at: https://perma.cc/EN3Q-JJ8M

MAXIM, 2001. *Fundamentals of RS-232 Serial Communications.* Tutorials 83, Maxim Integrated Products, Inc. 2001-03-29
Available from: https://bit.ly/32LgGpv
Archived at: https://perma.cc/CE85-K5UL

McBRIDE, William V., 1976. *Instrument Flying.* AF Manual 51-37. Headquarters US Air Force Washington DC 20330: Department of the Air Force, 1976-12-01
Available from: https://code7700.com/pdfs/usaf/afm_51-37_instrument_flying.pdf
Archived at: https://perma.cc/2DLJ-4GSL

MIKE COOK'S, 2021. *PWM: The Arduino doesn't have any analog outputs but it does have the nest best thing, PWM outputs. But what are they and how do you use them?* The Box. Tutorials as a mixture of techniques and experimentation in the art of electronics.
Available from: http://www.thebox.myzen.co.uk/Tutorial/PWM.html
Archived at: https://perma.cc/GM72-UE9V

MOLLOY, K., 2012. *ATL Fuel Level Sender Probes: Technical Specification*. Denbigh Rd, MK1 1DF, United Kingdom: ATL, AERO TEC LABORATORIES LTD. 2012-01-11
Available from: https://bit.ly/3nElTGH
Archived at: https://perma.cc/CX4H-4R4M

MOUNTAING, 2020. *What is the open systems interconnection model?: The OSI Model by Inernational Organization for Standardization (ISO).* Lightnetics, topic 24727
Available from: https://bit.ly/3eU1iJG
Archived at: https://perma.cc/XB25-N96T

MOURN, Richard 2011. *UAVs leverage IEEE-1394b data buses for success.* EE Times. Military & Aerospace Designline. 2011-07-09
Available from: https://bit.ly/33SyEGz
Archived at: https://perma.cc/YG9K-JBZJ

NASA, 1996. *Affordable Alternative Transportation: AGATE – Revitalizing General Aviation.* Hampton, VA 23666, United States: NASA Langley Research Center
Available from: https://www.nasa.gov/centers/langley/news/factsheets/AGATE.html
Archived at: https://perma.cc/SAZ3-LNSA

NASA, 2020. *Space Communications: What is the Deep Space Network?.* NASA. Pasadena, United States: Jet Propulsion Laboratory. 2020-03-30
Available from: https://go.nasa.gov/3ISQb0E
Archived at: https://perma.cc/WQ4Z-SVNV

NOAA 2019. *US/UK World Magnetic Model – Epoch 2020.0. Main Field Declination (D).* National Centers for Enviromental Information, National Oceanic and Atmospheric Administration, December 2019
Available from: https://bit.ly/32IBE88
Archived at: https://perma.cc/A6QT-NALV

NOVELL, 2009. *The TCP/IP Suite of Protocols.* Novell, 2009-11
Available from: https://bit.ly/3qQKzwt
Archived at: https://perma.cc/8GUH-RLN2

OPTIMUM, 2016. *RC Protokolle – TX / RX.* Delemont, Suisse: Optimum Racing
Available from: https://bit.ly/329tENo
Archived at: https://perma.cc/4W6X-7RC2

PANITZ, M., C. CHRISTOPOULOS, P. SEWELL, D. HOPE, J. DAWSON and A. MARVIN, 2008. *Modelling wireless communication in highly-multipath low-loss enviroments.* 2008 International Symposium on Electromagnetic Compatibility – EMC Europe, 2008
Available from: https://bit.ly/3Iv9zk4 (Closed Access)
Archived at: https://perma.cc/M4PD-VY3F

PARADISO, J. A. and T. STARNER, 2005. *Energy scavenging for mobile and wireless electronics.* In: IEEE Pervasive Computing, vol. 4, no. 1, page 18-27. 2005-03
Available from: https://bit.ly/3J7RN73 (Closed Access)
Archived at: https://perma.cc/WZ4F-TGS2

PAULITSCH, Michael, 2015. *ASCB: Avionisc Standard Communications Bus.* Taylor & Francis Group, LLC 2015. Part 44. Airbus Group Innovations. 2015-03-03
Available from: https://bit.ly/3qJcXly
Archived at: https://perma.cc/7382-7F5T

PLANE, 2021. *Honeywell Launches Anthem Scalable Avionics System.* Plane & Pilot. 2021-10-08
Available from: https://bit.ly/3GFIYQS
Archivet at: https://perma.cc/W9JN-WUDW

PRASAD, E. Giri, S. ARUNVINTHAN and Niladri Shekhar DAS, 2014. *Real-Time System Bus Architecture for Small Aircraft Using Can-Protocol.* In: International Refereed Journal of Engineering and Science (IRJES). Volume 3, Issue 5 (May 2014), page 24-29. ISSN: 2319-1821
Available from: http://www.irjes.com/Papers/vol3-issue5/D352429.pdf
Archived at: https://perma.cc/HJ6M-9DL9

QELECTROTECH, 2019. *QelectroTech.* Is a free software to create electric diagrams. GNU/GPL license.
Available from: https://qelectrotech.org/
Archived at: https://perma.cc/Q7C7-R8WB

RAY ALLEN, 2021. *The Ray Allen Company: RP3 LED Type Position Indicator*. Distribution Way Ste 15, Vista, CA 92081 USA.
Available from: http://rayallencompany.com/RACmedia/instructionsRP3.pdf
Archived at: https://perma.cc/KE5R-D2ZW

ROTAX, 2013. *Rotax 912 iS exceeds the assumed fuel efficiency.* Rotax, 2013-04-26
Available from: https://bit.ly/3n12IXm
Archived at: https://perma.cc/P9PE-AB66

SCHOLZ, Dieter, 2002. *Avionic Systems*. Lecture Notes. Hamburg University of Applied Science, Berliner Tor 9, 20099 Hamburg, Germany.
Available from: https://bit.ly/3GKrfIe
Archived at: https://perma.cc/X93Q-B25U

SHEA, Sharon, 2018. *Z-Wave*. Internet of Things, IoT Agenda, TechTarget
Available from: https://bit.ly/3ruRxaL
Archived at: https://perma.cc/GT32-NLSN

SILICON, 2021. *UG103.2: Zigbee Fundamentals.* Austin, USA: Silicon Labs. Silicon Laboratories Inc. Rev 1.5. 2021
Available from: https://bit.ly/3qBXXpt
Archived at: https://perma.cc/VP9Z-ZCH5

SILICON, 2022. *AN1138: Zigbee Mesh Network Performance.* Austin, USA: Silicon Labs. Silicon Laboratories Inc. Rev 1.5
Available from: https://bit.ly/3AddPlC
Archived at: https://perma.cc/HYX9-89CW

SKYVIEW, 2021. *SkyView SE, SkyView Classic, SkyView Touch, SkyView HDX: System Installation Guide.* Woodinville, USA: Dynon. 2021-11
Available from: https://bit.ly/3shTVC3
Archivet at: https://perma.cc/6ST8-X5EL

SLIDETODOC, 2013. *CSC 311 Data Communication and Networking Introduction* [Presentation]. SlideToDoc
Available from: https://bit.ly/3GGrWSP
Archived at: https://perma.cc/FJE2-7CST

SMITH, Dee, 2014. *Solar Flares And Electromagnetic Pulses.* CEO of Strategic Intelligence Group. Octavian Report
Available from: https://bit.ly/3GRqqNC
Archived at: https://perma.cc/G6KP-FRYJ

SONG, Jian, Liangliang WANG, Andre ZIBART and Christian KOCH, 2012. *Corrosion Protection of Electrically Conductive Surface.* Metals. Lemgo, Germany: Precision Engineering Laboratory. ISSN: 2075-4701
Available from: https://bit.ly/34X9mYq
Archived at: https://perma.cc/57J3-XGP8

SPILOK, Kathleen, 2021. *Raus aus dem Kabeldschungel.* In: *VDI nachrichten*. 2021-09-10 Nr. 36., page 6-7. ISSN 0042-1758

STOCK, Michael and Jim DEAS, 2009. *Fly-By-Wire for Experimental Aircraft?: A Vision based on CANaerospace/AGATE Data Bus Technology.* Stock Flight Systems and JAD Systems, Santa Clarita, USA
Available from: https://bit.ly/3nB7gEh
Archived at: https://perma.cc/2UNS-NXJV
Presentation available from: https://bit.ly/3fyED6c
Archived at: https://perma.cc/DM52-EZVC

STOCK, Michael, 2005. *CANaerospace/AGATE data bus.* Schützenweg 8a, Berg 82335, Germany: Stock Flight Systems, 2005
Available from: https://bit.ly/3yFp9pO ,
archived at: https://perma.cc/RC9K-2LWR

STOCK, Michael, 2009. *Arinc 825 sepcification for CAN in airborne application.* CAN Newsletter 4/2009. Schützenweg 8a, Berg 82335, Germany: Stock Flight Systems
Available from: https://bit.ly/3Fy95YG
Archived at: https://perma.cc/8DTR-C9TY

STOCK, Michael, 2011. *CANaerospace/ARINC825 Training Course* [Email: michael@stockflightsystems.com]. 2022-01-12. Schützenweg 8a, Berg 82335, Germany: Stock Flight Systems, 2011

STOCK, Michael, 2011b. *HALO ASCB-D/CANaerospace/ARINC825 Telemetry Interface System (TIS) User's Manual.* Rev. 1.5. Stock Flight Systems, 2011
Available from: https://bit.ly/3FO1N34
Archived at: https://perma.cc/RG4E-GEXS

STOCK, Michael, 2012. *Interview with Michael Stock – Inventor of the CANaerospace protocol and the new 912iS EMU.* Interview by: WILLIAM, under Aviation News. Rotax News – Your source for Rotax Aircraft Engine News
Available from: http://rotaxnews.net/?p=461
Archived at: https://perma.cc/9NLQ-TS6M

TBS, 2019. *TBS CROSSFIRE Funksystem: Adaptives Long Range- Fernsteuerungssystem.* TBS Crossfire. Rev 2019-03-24
Available from: https://www.team-blacksheep.com/tbs-crossfire-manual-de.pdf
Archived at: https://perma.cc/HF7F-X8YP

UEIDAQ, 2021. *Understanding ARINC-429: ARINC-429 Tutorial & Reference.* United Electronic Industries
Available from: https://www.ueidaq.com/arinc-429-tutorial-reference-guide
Archived at: https://perma.cc/89HW-T5YC

VERTICAL POWER, 2021. Willows Road NE, Kirkland, WA 98034 USA. Astronics Corp.
Available from: https://verticalpower.com/
Archived at: https://perma.cc/TD47-L9C6

VINARICKY, Eduard, 2016. *Elektrische Kontakte, Werkstoffe und Anwendungen: Grundlagen, Technologien, Prüfverfahren.* 3rd edition. Heidelberg: Springer Vieweg. ISBN 978-3-642-45426-4
Available from: https://bit.ly/3sd1MlC
Archived at: https://perma.cc/8YUG-UZY7

WIKIPEDIA, 2021. *OSI model.* Wikipedia, 2021-12-18
Available from: https://en.wikipedia.org/wiki/OSI_model
Archived at: https://perma.cc/UX8V-KGWS

WIKIPEDIA, 2021b. *MIL-STD-1553.* Wikipedia, 2021-11-14
Available from: https://en.wikipedia.org/wiki/MIL-STD-1553
Archived at: https://perma.cc/T65X-NER3

WIKIPEDIA, 2022. *RS-232.* Wikipedia, 2022-01-14
Available from: https://en.wikipedia.org/wiki/RS-232
Archived at: https://perma.cc/RLY5-WJBQ

WIKIPEDIA, 2022b. *Z-Wave.* Wikipedia, 2022-01-04
Available from: https://en.wikipedia.org/wiki/Z-Wave
Archived at: https://perma.cc/V9UU-SYS5

WIKIPEDIA, 2022c. *EnOcean.* Wikipedia, 2022-01-17
Available from: https://en.wikipedia.org/wiki/EnOcean
Archived at: https://perma.cc/PV28-96GU

WIKIPEDIA, 2022d. *IEEE 1394.* Wikipedia, 2022-01-14
Available from: https://en.wikipedia.org/wiki/IEEE_1394
Archived at: https://perma.cc/UY5R-KG6W

WRC, 2015. *Final Acts WRC-15: World Radiocommunication Conference.* Geneva Switzerland: ITU-R. 2015
Available from: https://bit.ly/3KAhrTc
Archived at: https://perma.cc/VZ7T-CXQM

YOUNIS, Marwan and Sevda ABADPOUR, 2018. *Lecture Notes for: Advanced Radio Communication I.* WiSe 2018/2019. Karlsruhe: Institut für Hochfrequenztechnik und Elektronik
Available from: https://www.ihe.kit.edu/img/studium/ARC_Main.pdf
Archived at: https://perma.cc/2QRT-T37L

On 2022-02-23 all links were tested and access was possible.

Appendix A Periodic Table

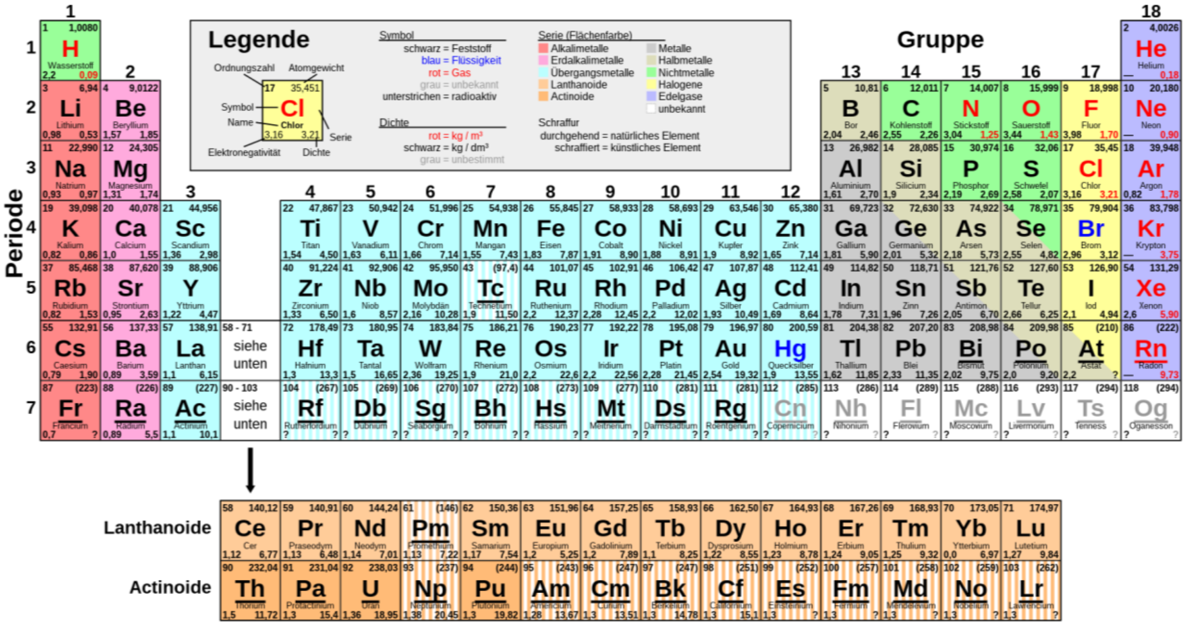


Figure A.1 Periodic table (Antonsusi 2019)

Appendix B CANaerospace Demonstrations Software

```
/*****************************************************************************************************************************
* CANaerospace demonstration software
*
* This is free source code which may be used by anyone for any purpose without
* restrictions. The author refuses any responsibility for the use of this code
* in any application, however.
*
* This program relies on two functions for CAN message transmission and
* reception which are defined as follows:
*
* short can_read(short dd, short can_id, CAN_AS_OBJ *msg_buf)
* short can_write(short dd, short can_id, CAN_AS_OBJ *msg_buf)
*
* Filename: canas_demo.c
*
*
* Function names
* ________________________________________________________________________________________
*
* void main(void)
*
* MODIFICATIONS:
*
* When          Version     What                            Who
* ________________________________________________________________________________________
*
* 12.07.1999    1.0         Initial Version                 M. Stock
* 10.09.2001    1.1         FLOAT data transmission added   M. Stock
*
*****************************************************************************************************************************/

#include "can_as.h"

/*
 * Definitions.
 */

#define HW_REVISION     0x23
#define SW_REVISION     0x45

/*
 * Externals.
 */

extern short can_read(short dd, short can_id, CAN_AS_OBJ *msg_buf);
extern short can_write(short dd, short can_id, CAN_AS_OBJ *msg_buf);

/*
 * main() starts here.
 */
```

```
void main(void)
{
char msg_count, last_msg_nr, our_node_id, *buf_ptr;
short cd, ret, loops;
CAN_AS_OBJ Switches_CAN, NS_Request_CAN, NS_Response_CAN, Lights_CAN;
CAN_AS_OBJ Cyclic_Lat_CAN, Cyclic_Long_CAN;
unsigned int *lights, *switches;
int *ns_resp_int, download_in_progress, chksum, buf_start;
float *cyclic_long, *cyclic_lat;

/*
 * Initialize variables.
 */

Cyclic_Lat_CAN.frame_type = DATA;
Cyclic_Lat_CAN.byte_count = 8;
Cyclic_Lat_CAN.node_id = our_node_id;
Cyclic_Lat_CAN.data_type = AS_FLOAT;
Cyclic_Lat_CAN.msg_code = 0;
Cyclic_Lat_CAN.service_code = 0;
cyclic_lat = (float *) &(Cyclic_Lat_CAN.data[0]);
*cyclic_lat = 0.0;

Cyclic_Long_CAN.frame_type = DATA;
Cyclic_Long_CAN.byte_count = 8;
Cyclic_Long_CAN.node_id = our_node_id;
Cyclic_Long_CAN.data_type = AS_FLOAT;
Cyclic_Long_CAN.msg_code = 0;
Cyclic_Long_CAN.service_code = 0;
cyclic_long = (float *) &(Cyclic_Long_CAN.data[0]);
*cyclic_long = 0.0;

Switches_CAN.frame_type = DATA;
Switches_CAN.byte_count = 8;
Switches_CAN.node_id = our_node_id;
Switches_CAN.data_type = AS_BLONG;
Switches_CAN.msg_code = 0;
Switches_CAN.service_code = 0;

switches = (unsigned int *) &(Switches_CAN.data[0]);
*switches = 0;

lights = (unsigned int *) &(Lights_CAN.data[0]);
*lights = 0;

NS_Response_CAN.frame_type = DATA;
NS_Response_CAN.node_id = our_node_id;

ns_resp_int = (int *) &(NS_Response_CAN.data[0]);

buf_start = 0x10000;        /* 64k memory area for data downloads */

/*
 * Now go into an endless loop sending and receiving normal operation
 * data. We also process incoming node service requests to show how
```

```
 * this may be implemented.
 */
for (;;)
 {
 /*
  * Transmit pitch/roll control position and trim switch states. Note
  * that the data transmission rate should be limited to avoid unnecessary
  * bus load.
  */

 *cyclic_long = 0.5;   /* actually, sensor data should be used here */

 Cyclic_Long_CAN.msg_code++;
 can_write(cd, PITCH_CTRL_POS_ID, &Cyclic_Long_CAN);

 *cyclic_lat = 0.5;      /* actually, sensor data should be used here */

 Cyclic_Lat_CAN.msg_code++;
 can_write(cd, ROLL_CTRL_POS_ID, &Cyclic_Lat_CAN);

 Switches_CAN.msg_code++;
 ret = can_write(cd, TRIM_SWITCH_ID, &Switches_CAN);

 ret = can_read(cd, TRIM_LIGHTS_ID, &Lights_CAN);

 if (ret == NS_OK)
  {
  /*
   * This is where incoming trim light data would be processed.
   */
  }

 /*
  * Process node service requests (Node Service Channel #0). As an example,
  * we demonstrate the IDS and DDS services.
  */

 ret = can_read(cd, NS_REQ_0_ID, &NS_Request_CAN);

 if ((ret == NS_OK) && (NS_Request_CAN.node_id == our_node_id))
  {
  switch(NS_Request_CAN.service_code)
   {
   /*
    * Node identification service.
    */

   case IDS:
     if (NS_Request_CAN.msg_code == 0x00)
      {
      NS_Response_CAN.data[0] = HW_REVISION;
      NS_Response_CAN.data[1] = SW_REVISION;
      NS_Response_CAN.data[2] = 0;
      NS_Response_CAN.data[3] = 0;
      NS_Response_CAN.msg_code = NS_OK;
      }
```

```
  else
   {
   NS_Response_CAN.data[0] = 0;
   NS_Response_CAN.data[1] = 0;
   NS_Response_CAN.data[2] = 0;
   NS_Response_CAN.data[3] = 0;
   NS_Response_CAN.msg_code = NS_UNKNOWN_SVC;
   }

  NS_Response_CAN.service_code = IDS;
  NS_Response_CAN.data_type = AS_UCHAR_4;
  NS_Response_CAN.byte_count = 8;

  can_write(cd, NS_RSP_0_ID, &NS_Response_CAN);
  break;

/*
 * Data download service.
 */

case DDS:

  if ((NS_Request_CAN.data_type == AS_MEMID) && (!download_in_progress))
   {
   if (NS_Request_CAN.msg_code == 1)
    {
    /*
     * We only support memory destination identifier "1" here.
     */

    NS_Response_CAN.byte_count = 8;
    NS_Response_CAN.data_type = AS_LONG;
    NS_Response_CAN.service_code = NS_Request_CAN.service_code;
    NS_Response_CAN.msg_code = NS_Request_CAN.msg_code;
    *ns_resp_int = NS_XON;
    buf_ptr = (char *) buf_start;
    chksum = 0;
    msg_count = 0;
    last_msg_nr = NS_Request_CAN.msg_code;
    download_in_progress = 0;
    }
   else
    {
    NS_Response_CAN.byte_count = 8;
    NS_Response_CAN.data_type = AS_LONG;
    NS_Response_CAN.service_code = NS_Request_CAN.service_code;
    NS_Response_CAN.msg_code = NS_Request_CAN.msg_code;
    *ns_resp_int = NS_INVALID;
    }
   can_write(cd, NS_RSP_0_ID, &NS_Response_CAN);
   }

  else if (download_in_progress)
   {
   for (loops = 0; loops < NS_Request_CAN.byte_count-4; loops++)
    {
```

```
          *buf_ptr = NS_Request_CAN.data[loops];
          chksum += NS_Request_CAN.data[loops];
          buf_ptr++;
          msg_count++;
          if (msg_count > last_msg_nr)
           {
           download_in_progress = -1;
           NS_Response_CAN.byte_count = 8;
           NS_Response_CAN.data_type = AS_CHKSUM;
           NS_Response_CAN.service_code = NS_Request_CAN.service_code;
           NS_Response_CAN.msg_code = NS_Request_CAN.msg_code;
           *ns_resp_int = chksum;
           can_write(cd, NS_RSP_0_ID, &NS_Response_CAN);
           }
          }
         }
        break;

      /*
       * Other services may be supported here.
       */

      default:
        break;
      }
    }
  }
}

/*
 * End of file.
 */
```

Appendix C CANaerospace Interface Definitions

```
/*****************************************************************************************************************************
* CANaerospace interface definitions
*
* (C) 1998 - 2015 Stock Flight Systems. All rights reserved.
*
* Filename: can_as.h
*
* This file contains definitions and the structures used for the CANaerospace
* interface definition.
*
* MODIFICATIONS:
*
* When        Version   What                                    Who
* ---------------------------------------------------------------------------------------
*
* 24.02.1998  1.00      Initial Version                         M. Stock
* 26.06.1998  1.05      CAN_AS_OBJ definition changed           M. Stock
* 15.07.1998  1.10      AS_BSHORT_2 definition added            M. Stock
* 21.01.1999  1.20      Identifier ranges changed               M. Stock
* 13.04.1999  1.30      Identifier definitions modified         M. Stock
* 10.05.1999  1.31      Some identifiers added                  M. Stock
* 10.05.1999  1.32      Nav sensor definitions changed          K. Heidenreich
* 14.05.1999  1.33      Some identifiers rearranged             M. Stock
* 19.05.1999  1.34      DME frequencies added                   K. Heidenreich
* 12.07.1999  1.40      Command definitions added               M. Stock
* 30.10.1999  1.41      CAN_AS_OBJ expanded                     M. Stock
* 23.10.2000  1.42      Node service error return codes added   M. Stock
* 29.12.2000  1.43      Some service code definitions added     M. Stock
* 19.06.2001  1.44      29-bit ID support for CAN_AS_OBJ        M. Stock
* 17.09.2001  1.45      State transmission service code added   M. Stock
* 14.02.2002  1.46      Some more GPS/flightstate identifiers   M. Stock
* 05.06.2002  1.47      Density alitude identifier added        M. Stock
* 18.07.2002  1.48      Wind speed/direction dentifiers added   M. Stock
* 31.03.2003  1.49      3-byte data types added                 M. Stock
* 29.06.2004  1.50      CIS/BSS node services added             M. Stock
* 04.07.2005  1.51      NIS node service added                  M. Stock
* 16.07.2005  1.52      MIS/MCS/CSS/DSS node services added     M. Stock
* 10.02.2006  1.53      RCS node service added                  M. Stock
* 04.10.2008  1.54      Some identifiers added                  M. Stock
* 08.01.2012  1.55      ECU_x_HOURS_y_ID identifiers added      M. Stock
* 21.09.2012  1.56      SIS/SXS node services added             M. Stock
* 03.04.2015  1.57      TOT_FUEL_FLOW_ID added                  M. Stock
* 09.10.2018  1.58      LANE_X_TIMESTAMP_ID added               M. Stock
*
*****************************************************************************************************************************/

/*
 * Data type definitions.
 */

#define     AS_NODATA     0
#define     AS_ERROR      1
```

```
#define     AS_FLOAT        2
#define     AS_LONG         3
#define     AS_ULONG        4
#define     AS_BLONG        5
#define     AS_SHORT        6
#define     AS_USHORT       7
#define     AS_BSHORT       8
#define     AS_CHAR         9
#define     AS_UCHAR        10
#define     AS_BCHAR        11
#define     AS_SHORT_2      12
#define     AS_USHORT_2     13
#define     AS_BSHORT_2     14
#define     AS_CHAR_4       15
#define     AS_UCHAR_4      16
#define     AS_BCHAR_4      17
#define     AS_CHAR_2       18
#define     AS_UCHAR_2      19
#define     AS_BCHAR_2      20
#define     AS_MEMID        21
#define     AS_CHKSUM       22
#define     AS_ACHAR        23
#define     AS_ACHAR_2      24
#define     AS_ACHAR_4      25
#define     AS_CHAR_3       26
#define     AS_UCHAR_3      27
#define     AS_BCHAR_3      28
#define     AS_ACHAR_3      29

/*
 * Node-ID definitions.
 */

#define     ALL_NODES       0

/*
 * Service code definitions.
 */

#define         IDS     0    /* identification service */
#define         NSS     1    /* node synchronisation service */
#define         DDS     2    /* data download service */
#define         DUS     3    /* data upload service */
#define         SCS     4    /* simulation control service */
#define         TIS     5    /* transmission interval service */
#define         FPS     6    /* FLASH programming service */
#define         STS     7    /* state transmission service */
#define         FSS     8    /* filter setting service */
#define         CIS     9    /* CAN identifier setting service */
#define         BSS     10   /* CAN baudrate setting service */
#define         NIS     11   /* node-ID setting service */
#define         MIS     12   /* module information service */
#define         MCS     13   /* module configuration service */
#define         CSS     14   /* CAN identifier setting service */
#define         DSS     15   /* ID distribution setting service */
#define         RCS     16   /* Data recording control service */
```

```
#define         SIS         17      /* Signal ID setting service */
#define         SXS         18      /* Signal xmit interval service */

/*
 * Command definitions for identification service (IDS).
 */

#define         NS_STD_INFO_REQ         0x00    /* get standard revision information */
#define    NS_UNKNOWN_CODE 0xff

/*
 * Command definitions for data upload/download service (UDS/DDS).
 */

#define         NS_XOFF         0       /* halt transmission */
#define         NS_XON          1       /* resume transmission */
#define         NS_ABORT        -1      /* abort upload/download */
#define         NS_INVALID      -2      /* invalid operation or memory ID */

/*
 * Node service error return codes. User-defined codes start from -40.
 */

#define         NS_OK                   0
#define         NS_UNKNOWN_SVC          -1      /* 0xff */
#define         NS_UNKNOWN_MSG          -2      /* 0xfe */
#define         NS_INVALID_MODE         -3      /* 0xfd */
#define         NS_INVALID_CHAN         -4      /* 0xfc */
#define         NS_INVALID_ADDR         -5      /* 0xfb */
#define         NS_OUT_OF_RANGE         -6      /* 0xfa */
#define         NS_OUT_OF_BUFS          -7      /* 0xf9 */
#define         NS_BUF_FULL             -8      /* 0xf8 */
#define         NS_SVC_FAILED           -9      /* 0xf7 */
#define         NS_DISABLED             -10     /* 0xf6 */
#define         NS_OVERLAPPING          -11     /* 0xf5 */
#define         NS_BUSY                 -12     /* 0xf4 */
#define         NS_NOEXIST              -13     /* 0xf3 */
#define         NS_INVALID_CMD          -14     /* 0xf2 */

/*
 * Definitions for frequently used conversions.
 */

#define         M_2_FT          3.28213         /* meter to feet */
#define         FT_2_M          0.30468         /* feet to meter */
#define         MPS_2_FPM       196.9279        /* m/s to feet/min. */
#define         MPS_2_KTS       1.9438          /* m/s to kts */
#define         KTS_2_MPS       0.51450         /* kts to m/s */
#define         FPM_2_MPS       0.00508         /* feet/min. to m/s */
#define         KMH_2_KTS       0.53996         /* km/h to kts */
#define         KTS_2_KMH       1.852           /* kts to km/h */
#define         MPS_2_KMH       3.6             /* m/s to km/h */
#define         NM_2_KM         1.852           /* nautical miles to km */
#define         HG_2_HPA        33.8759         /* inch Hg to hPa */
#define         HPA_2_HG        0.02952         /* hPa to inch Hg */
#define         KG_2_LBS        2.205           /* kg to lbs */
```

```
#define         LBS_2_KG    0.45351           /* kg to lbs */
#define         LBSSQFT_2_KGM2 4.8854         /* lbs/sqft to kg/m^2 */
#define         KGM2_2_LBSSQFT 0.20469        /* lbs/sqft to kg/m^2 */
#define         GALS_2_LTR3.7853              /* US gallons to liters */
#define         LTR_2_GALS0.2642              /* liters to US gallons */
#define         LTR_2_QUARTS    1.0567        /* liters to US liquid quarts */
#define         QUARTS_2_LTR    0.9463        /* US liquid quarts to liters */

/*
 * Node service identifier definitions.
 */

#define     NS_REQ_0_ID         128     /* high priority node service request */
#define     NS_RSP_0_ID         129     /* high priority node service response*/
#define     NS_REQ_1_ID         NS_REQ_0_ID+2
#define     NS_RSP_1_ID         NS_RSP_0_ID+2
#define     NS_REQ_2_ID         NS_REQ_0_ID+4
#define     NS_RSP_2_ID         NS_RSP_0_ID+4
#define     NS_REQ_3_ID         NS_REQ_0_ID+6
#define     NS_RSP_3_ID         NS_RSP_0_ID+6
#define     NS_REQ_4_ID         NS_REQ_0_ID+8
#define     NS_RSP_4_ID         NS_RSP_0_ID+8
#define     NS_REQ_5_ID         NS_REQ_0_ID+10
#define     NS_RSP_5_ID         NS_RSP_0_ID+10
#define     NS_REQ_6_ID         NS_REQ_0_ID+12
#define     NS_RSP_6_ID         NS_RSP_0_ID+12
#define     NS_REQ_7_ID         NS_REQ_0_ID+14
#define     NS_RSP_7_ID         NS_RSP_0_ID+14
#define     NS_REQ_8_ID         NS_REQ_0_ID+16
#define     NS_RSP_8_ID         NS_RSP_0_ID+16
#define     NS_REQ_9_ID         NS_REQ_0_ID+18
#define     NS_RSP_9_ID         NS_RSP_0_ID+18
#define     NS_REQ_10_ID        NS_REQ_0_ID+20
#define     NS_RSP_10_ID        NS_RSP_0_ID+20
#define     NS_REQ_11_ID        NS_REQ_0_ID+22
#define     NS_RSP_11_ID        NS_RSP_0_ID+22
#define     NS_REQ_12_ID        NS_REQ_0_ID+24
#define     NS_RSP_12_ID        NS_RSP_0_ID+24
#define     NS_REQ_13_ID        NS_REQ_0_ID+26
#define     NS_RSP_13_ID        NS_RSP_0_ID+26
#define     NS_REQ_14_ID        NS_REQ_0_ID+28
#define     NS_RSP_14_ID        NS_RSP_0_ID+28
#define     NS_REQ_15_ID        NS_REQ_0_ID+30
#define     NS_RSP_15_ID        NS_RSP_0_ID+30
#define     NS_REQ_16_ID        NS_REQ_0_ID+32
#define     NS_RSP_16_ID        NS_RSP_0_ID+32
#define     NS_REQ_17_ID        NS_REQ_0_ID+34
#define     NS_RSP_17_ID        NS_RSP_0_ID+34
#define     NS_REQ_18_ID        NS_REQ_0_ID+36
#define     NS_RSP_18_ID        NS_RSP_0_ID+36
#define     NS_REQ_19_ID        NS_REQ_0_ID+38
#define     NS_RSP_19_ID        NS_RSP_0_ID+38
#define     NS_REQ_20_ID        NS_REQ_0_ID+40
#define     NS_RSP_20_ID        NS_RSP_0_ID+40
#define     NS_REQ_21_ID        NS_REQ_0_ID+42
#define     NS_RSP_21_ID        NS_RSP_0_ID+42
```

```
#define     NS_REQ_22_ID      NS_REQ_0_ID+44
#define     NS_RSP_22_ID      NS_RSP_0_ID+44
#define     NS_REQ_23_ID      NS_REQ_0_ID+46
#define     NS_RSP_23_ID      NS_RSP_0_ID+46
#define     NS_REQ_24_ID      NS_REQ_0_ID+48
#define     NS_RSP_24_ID      NS_RSP_0_ID+48
#define     NS_REQ_25_ID      NS_REQ_0_ID+50
#define     NS_RSP_25_ID      NS_RSP_0_ID+50
#define     NS_REQ_26_ID      NS_REQ_0_ID+52
#define     NS_RSP_26_ID      NS_RSP_0_ID+52
#define     NS_REQ_27_ID      NS_REQ_0_ID+54
#define     NS_RSP_27_ID      NS_RSP_0_ID+54
#define     NS_REQ_28_ID      NS_REQ_0_ID+56
#define     NS_RSP_28_ID      NS_RSP_0_ID+56
#define     NS_REQ_29_ID      NS_REQ_0_ID+58
#define     NS_RSP_29_ID      NS_RSP_0_ID+58
#define     NS_REQ_30_ID      NS_REQ_0_ID+60
#define     NS_RSP_30_ID      NS_RSP_0_ID+60
#define     NS_REQ_31_ID      NS_REQ_0_ID+62
#define     NS_RSP_31_ID      NS_RSP_0_ID+62
#define     NS_REQ_32_ID      NS_REQ_0_ID+64
#define     NS_RSP_32_ID      NS_RSP_0_ID+64
#define     NS_REQ_33_ID      NS_REQ_0_ID+66
#define     NS_RSP_33_ID      NS_RSP_0_ID+66
#define     NS_REQ_34_ID      NS_REQ_0_ID+68
#define     NS_RSP_34_ID      NS_RSP_0_ID+68
#define     NS_REQ_35_ID      NS_REQ_0_ID+70
#define     NS_RSP_35_ID      NS_RSP_0_ID+70

#define     NS_REQ_100_ID     2000   /* low priority node service request */
#define     NS_RSP_100_ID     2001   /* low priority node service response */
#define     NS_REQ_101_ID     NS_REQ_100_ID+2
#define     NS_RSP_101_ID     NS_RSP_100_ID+2
#define     NS_REQ_102_ID     NS_REQ_100_ID+4
#define     NS_RSP_102_ID     NS_RSP_100_ID+4
#define     NS_REQ_103_ID     NS_REQ_100_ID+6
#define     NS_RSP_103_ID     NS_RSP_100_ID+6
#define     NS_REQ_104_ID     NS_REQ_100_ID+8
#define     NS_RSP_104_ID     NS_RSP_100_ID+8
#define     NS_REQ_105_ID     NS_REQ_100_ID+10
#define     NS_RSP_105_ID     NS_RSP_100_ID+10
#define     NS_REQ_106_ID     NS_REQ_100_ID+12
#define     NS_RSP_106_ID     NS_RSP_100_ID+12
#define     NS_REQ_107_ID     NS_REQ_100_ID+14
#define     NS_RSP_107_ID     NS_RSP_100_ID+14
#define     NS_REQ_108_ID     NS_REQ_100_ID+16
#define     NS_RSP_108_ID     NS_RSP_100_ID+16
#define     NS_REQ_109_ID     NS_REQ_100_ID+18
#define     NS_RSP_109_ID     NS_RSP_100_ID+18
#define     NS_REQ_110_ID     NS_REQ_100_ID+20
#define     NS_RSP_110_ID     NS_RSP_100_ID+20
#define     NS_REQ_111_ID     NS_REQ_100_ID+22
#define     NS_RSP_111_ID     NS_RSP_100_ID+22
#define     NS_REQ_112_ID     NS_REQ_100_ID+24
#define     NS_RSP_112_ID     NS_RSP_100_ID+24
#define     NS_REQ_113_ID     NS_REQ_100_ID+26
```

```
#define     NS_RSP_113_ID          NS_RSP_100_ID+26
#define     NS_REQ_114_ID          NS_REQ_100_ID+28
#define     NS_RSP_114_ID          NS_RSP_100_ID+28
#define     NS_REQ_115_ID          NS_REQ_100_ID+30
#define     NS_RSP_115_ID          NS_RSP_100_ID+30

/*
 * Flight state identifier definitions.
 */

#define     BODY_LONG_ACC_ID              300   /* body longitudinal acceleration */
#define     BODY_LAT_ACC_ID               301   /* body lateral acceleration */
#define     BODY_NORM_ACC_ID              302   /* body normal acceleration */
#define     BODY_PITCH_RATE_ID            303   /* a/c pitch rate */
#define     BODY_ROLL_RATE_ID             304   /* a/c roll rate */
#define     BODY_YAW_RATE_ID              305   /* a/c yaw rate */
#define     RUDDER_POS_ID                 306   /* rudder position */
#define     STABILIZER_POS_ID             307   /* horizontal stabilizer position */
#define     ELEVATOR_POS_ID               308   /* elevator position */
#define     LEFT_AILERON_POS_ID           309   /* left aileron position */
#define     RIGHT_AILERON_POS_ID          310   /* right aileron position */
#define     BODY_PITCH_ANGLE_ID           311   /* a/c pitch angle */
#define     BODY_ROLL_ANGLE_ID            312   /* a/c roll angle */
#define     BODY_SIDESLIP_ID              313   /* a/c sideslip */
#define     ALTITUDE_RATE_ID              314   /* vertical speed */
#define     IND_AIRSPEED_ID               315   /* indicated airspeed */
#define     TRUE_AIRSPEED_ID              316   /* true airspeed */
#define     CAL_AIRSPEED_ID               317   /* calibrated airspeed */
#define     MACH_NUMBER_ID                318   /* mach number */
#define     BARO_CORRECTION_ID            319   /* barometric correction (QNH) */
#define     BARO_ALTITUDE_ID              320   /* barometric altitude */
#define     HEADING_ANGLE_ID              321   /* heading angle */
#define     STANDARD_ALTITUDE_ID          322   /* standard altitude */
#define     TOTAL_AIR_TEMP_ID             323   /* total air temperature */
#define     STATIC_AIR_TEMP_ID            324   /* static air temperature */
#define     DIFFERENTIAL_PRESS_ID         325   /* differential pressure */
#define     STATIC_PRESS_ID               326   /* static pressure */
#define     HEADING_RATE_ID               327   /* magnetic heading rate */
#define     PORT_AOA_ID                   328   /* port side angle-of-attack */
#define     STARBOARD_AOA_ID              329   /* starboard side angle-of-attack */
#define     DENSITY_ALT_ID                330   /* density altitude */
#define     TURN_COORD_RATE_ID            331   /* turn coordination rate */
#define     TRUE_ALTITUDE_ID              332   /* temperature corrected altitude */
#define     WIND_SPEED_ID                 333   /* wind speed */
#define     WIND_DIRECTION_ID             334   /* wind direction in degrees */
#define     OUTSIDE_AIR_TEMP_ID           335   /* outside air temperature */
#define     BODY_NORM_VEL_ID              336   /* body normal velocity */
#define     BODY_LONG_VEL_ID              337   /* body longitudinal velocity */
#define     BODY_LAT_VEL_ID               338   /* body latral velocity */
#define     TOTAL_PRESS_ID                339   /* total pressure */
#define     FLAPS_POS_ID                  340   /* flaps position */
#define     SLATS_POS_ID                  341   /* slats position */
#define     SPEED_BRAKE_POS_ID            342   /* speed brake position */
#define     LAT_TRIM_SURFACE_POS_ID       343   /* lateral trim surface position */
#define     LON_TRIM_SURFACE_POS_ID       344   /* longitudinal trim surface position */
#define     DIR_TRIM_SURFACE_POS_ID       345   /* directional trim surface position */
```

```
#define     PORT_BETA_ID                346   /* beta measured on port side */
#define     STARBOARD_BETA_ID           347   /* beta measured on starboard side */
#define     MAIN_GEAR_POS_ID            348   /* main landing gear position */
#define     FRONT_GEAR_POS_ID           349   /* front landing gear position */
#define     RETRACT_ENGINE_POS_ID       350   /* rectractable engine pos. (gliders) */
#define     STARB_FLAPERON_POS_ID       351   /* flaperon position (starboard side) */
#define     PORT_FLAPERON_POS_ID        352   /* flaperon position (port side) */
#define     STARB_CANARD_POS_ID         353   /* canard position (starboard side) */
#define     PORT_CANARD_POS_ID          354   /* canard position (port side) */
#define     STARB_TAILERON_POS_ID       355   /* taileron position (starboard side) */
#define     PORT_TAILERON_POS_ID        356   /* taileron position (port side) */
#define     AIR_DATA_SIGMA_ID           360   /* Rho/Rho_0 */

/*
 * Flight controls identifier definitions.
 */

#define     PITCH_CTRL_POS_ID           400   /* pitch control position */
#define     ROLL_CTRL_POS_ID            401   /* roll control position */
#define     LAT_TRIM_POS_ID             402   /* lateral stick trim position */
#define     YAW_CTRL_POS_ID             403   /* yaw control position */
#define     COLLECTIVE_CTRL_POS_ID      404   /* collective control position */
#define     LONG_TRIM_POS_ID            405   /* longitudinal stick trim position */
#define     PEDAL_TRIM_POS_ID           406   /* directional pedals trim position */
#define     COLLECTIVE_TRIM_POS_ID      407   /* collective stick trim position */
#define     CONTROL_SWITCH_ID           408   /* control stick switches */
#define     LAT_TRIM_SPEED_ID           409   /* lateral trim speed */
#define     LONG_TRIM_SPEED_ID          410   /* longitudinal trim speed */
#define     PEDAL_TRIM_SPEED_ID         411   /* pedal trim speed */
#define     COLLECTIVE_TRIM_SPEED_ID    412   /* collective trim speed */
#define     NOSE_WHEEL_STEER_POS_ID     413   /* nose wheel steering position */
#define     POWER_LEVER_1_POS_A_ID      414   /* power lever position (engine #1) */
#define     POWER_LEVER_2_POS_A_ID      415   /* power lever position (engine #2) */
#define     POWER_LEVER_3_POS_A_ID      416   /* power lever position (engine #3) */
#define     POWER_LEVER_4_POS_A_ID      417   /* power lever position (engine #4) */
#define     COND_LEVER_1_POS_A_ID       418   /* condition lever pos. (engine #1) */
#define     COND_LEVER_2_POS_A_ID       419   /* condition lever pos. (engine #2) */
#define     COND_LEVER_3_POS_A_ID       420   /* condition lever pos. (engine #3) */
#define     COND_LEVER_4_POS_A_ID       421   /* condition lever pos. (engine #4) */
#define     POWER_LEVER_1_POS_B_ID      422   /* power lever position (engine #1) */
#define     POWER_LEVER_2_POS_B_ID      423   /* power lever position (engine #2) */
#define     POWER_LEVER_3_POS_B_ID      424   /* power lever position (engine #3) */
#define     POWER_LEVER_4_POS_B_ID      425   /* power lever position (engine #4) */
#define     COND_LEVER_1_POS_B_ID       426   /* condition lever pos. (engine #1) */
#define     COND_LEVER_2_POS_B_ID       427   /* condition lever pos. (engine #2) */
#define     COND_LEVER_3_POS_B_ID       428   /* condition lever pos. (engine #3) */
#define     COND_LEVER_4_POS_B_ID       429   /* condition lever pos. (engine #4) */
#define     FLAPS_LEVER_POS_ID          430   /* flaps lever position */
#define     SLATS_LEVER_POS_ID          431   /* slats lever position */
#define     PARK_BRAKE_LEVER_POS_ID     432   /* park brake lever position */
#define     SPEED_BRAKE_LVR_POS_ID      433   /* speed brake lever position */
#define     POWER_LEVER_MAX_POS_ID      434   /* maximum power lever position */
#define     PLT_LEFT_BRAKE_POS_ID       435   /* pilot's left foot brake position */
#define     PLT_RIGHT_BRAKE_POS_ID      436   /* pilot's right foot brake position */
#define     CPLT_LEFT_BRAKE_POS_ID      437   /* copilot's left foot brake position */
#define     CPLT_RIGHT_BRAKE_POS_ID     438   /* copilot's right foot brake pos. */
```

```
#define     TRIM_SWITCH_ID              439    /* trim system switches */
#define     TRIM_LIGHTS_ID              440    /* trim system indicator lights */
#define     COLLECTIVE_SWITCH_ID        441    /* collective control stick switches */
#define     STICK_SHAKER_ID             442    /* stick shaker stall warning device */
#define     PITOT_HEAT_SWITCH_ID        443    /* pitot heating switches */
#define     LDG_LIGHT_SWITCH_ID         444    /* landing light switches */
#define     TAXI_LIGHT_SWITCH_ID        445    /* taxi light switches */
#define     STROBE_LIGHT_SWITCH_ID      446    /* strobe light switches */
#define     CABIN_LIGHT_SWITCH_ID       447    /* cabin light switches */
#define     PANEL_LIGHT_SWITCH_ID       448    /* panel light switches */
#define     AVIONICS_BUS_SWITCH_ID      449    /* avionicd bus switches */
#define     ENG_START_SWITCH_ID         450    /* enigne start switches */

/*
 * Engine/fuel system data identifier definitions.
 */

#define     ENGINE_1_N1_A_ID            500    /* engine #1 N1/rpm */
#define     ENGINE_2_N1_A_ID            501    /* engine #2 N1/rpm */
#define     ENGINE_3_N1_A_ID            502    /* engine #3 N1/rpm */
#define     ENGINE_4_N1_A_ID            503    /* engine #4 N1/rpm */
#define     ENGINE_1_N2_A_ID            504    /* engine #1 N2/prop rpm */
#define     ENGINE_2_N2_A_ID            505    /* engine #2 N2/prop rpm */
#define     ENGINE_3_N2_A_ID            506    /* engine #3 N2/prop rpm */
#define     ENGINE_4_N2_A_ID            507    /* engine #4 N2/prop rpm */
#define     ENGINE_1_TORQUE_A_ID        508    /* engine #1 torque */
#define     ENGINE_2_TORQUE_A_ID        509    /* engine #2 torque */
#define     ENGINE_3_TORQUE_A_ID        510    /* engine #3 torque */
#define     ENGINE_4_TORQUE_A_ID        511    /* engine #4 torque */
#define     ENGINE_1_TIT_A_ID           512    /* engine #1 turbine inlet temp */
#define     ENGINE_2_TIT_A_ID           513    /* engine #2 turbine inlet temp */
#define     ENGINE_3_TIT_A_ID           514    /* engine #3 turbine inlet temp */
#define     ENGINE_4_TIT_A_ID           515    /* engine #4 turbine inlet temp */
#define     ENGINE_1_ITT_A_ID           516    /* engine #1 interturbine temp */
#define     ENGINE_2_ITT_A_ID           517    /* engine #2 interturbine temp */
#define     ENGINE_3_ITT_A_ID           518    /* engine #3 interturbine temp */
#define     ENGINE_4_ITT_A_ID           519    /* engine #4 interturbine temp */
#define     ENGINE_1_TOT_A_ID           520    /* engine #1 turbine outlet temp */
#define     ENGINE_2_TOT_A_ID           521    /* engine #2 turbine outlet temp */
#define     ENGINE_3_TOT_A_ID           522    /* engine #3 turbine outlet temp */
#define     ENGINE_4_TOT_A_ID           523    /* engine #4 turbine outlet temp */
#define     ENGINE_1_FUEL_FLOW_A_ID     524    /* engine #1 fuel flow rate */
#define     ENGINE_2_FUEL_FLOW_A_ID     525    /* engine #2 fuel flow rate */
#define     ENGINE_3_FUEL_FLOW_A_ID     526    /* engine #3 fuel flow rate */
#define     ENGINE_4_FUEL_FLOW_A_ID     527    /* engine #4 fuel flow rate */
#define     ENGINE_1_MAN_PRESS_A_ID     528    /* engine #1 manifold pressure */
#define     ENGINE_2_MAN_PRESS_A_ID     529    /* engine #2 manifold pressure */
#define     ENGINE_3_MAN_PRESS_A_ID     530    /* engine #3 manifold pressure */
#define     ENGINE_4_MAN_PRESS_A_ID     531    /* engine #4 manifold pressure */
#define     ENGINE_1_OIL_PRESS_A_ID     532    /* engine #1 oil pressure */
#define     ENGINE_2_OIL_PRESS_A_ID     533    /* engine #2 oil pressure */
#define     ENGINE_3_OIL_PRESS_A_ID     534    /* engine #3 oil pressure */
#define     ENGINE_4_OIL_PRESS_A_ID     535    /* engine #4 oil pressure */
#define     ENGINE_1_OIL_TEMP_A_ID      536    /* engine #1 oil temp */
#define     ENGINE_2_OIL_TEMP_A_ID      537    /* engine #2 oil temp */
#define     ENGINE_3_OIL_TEMP_A_ID      538    /* engine #3 oil temp */
```

```
#define    ENGINE_4_OIL_TEMP_A_ID      539   /* engine #4 oil temp */
#define    ENGINE_1_CHT_A_ID           540   /* engine #1 cylinder head temp */
#define    ENGINE_2_CHT_A_ID           541   /* engine #2 cylinder head temp */
#define    ENGINE_3_CHT_A_ID           542   /* engine #3 cylinder head temp */
#define    ENGINE_4_CHT_A_ID           543   /* engine #4 cylinder head temp */
#define    ENGINE_1_OIL_QUANT_A_ID     544   /* engine #1 oil quantity */
#define    ENGINE_2_OIL_QUANT_A_ID     545   /* engine #2 oil quantity */
#define    ENGINE_3_OIL_QUANT_A_ID     546   /* engine #3 oil quantity */
#define    ENGINE_4_OIL_QUANT_A_ID     547   /* engine #4 oil quantity */
#define    ENGINE_1_COOL_TEMP_A_ID     548   /* engine #1 cooland temp */
#define    ENGINE_2_COOL_TEMP_A_ID     549   /* engine #2 cooland temp */
#define    ENGINE_3_COOL_TEMP_A_ID     550   /* engine #3 cooland temp */
#define    ENGINE_4_COOL_TEMP_A_ID     551   /* engine #4 cooland temp */
#define    ENGINE_1_POW_RATIO_A_ID     552   /* engine #1 power ratio */
#define    ENGINE_2_POW_RATIO_A_ID     553   /* engine #1 power ratio */
#define    ENGINE_3_POW_RATIO_A_ID     554   /* engine #1 power ratio */
#define    ENGINE_4_POW_RATIO_A_ID     555   /* engine #1 power ratio */
#define    ENGINE_1_STATUS_1_A_ID      556   /* engine #1 status word 1 */
#define    ENGINE_2_STATUS_1_A_ID      557   /* engine #2 status word 1 */
#define    ENGINE_3_STATUS_1_A_ID      558   /* engine #3 status word 1 */
#define    ENGINE_4_STATUS_1_A_ID      559   /* engine #4 status word 1 */
#define    ENGINE_1_STATUS_2_A_ID      560   /* engine #1 status word 2 */
#define    ENGINE_2_STATUS_2_A_ID      561   /* engine #2 status word 2 */
#define    ENGINE_3_STATUS_2_A_ID      562   /* engine #3 status word 2 */
#define    ENGINE_4_STATUS_2_A_ID      563   /* engine #4 status word 2 */

#define    ENGINE_1_N1_B_ID            564   /* engine #1 N1/rpm */
#define    ENGINE_2_N1_B_ID            565   /* engine #2 N1/rpm */
#define    ENGINE_3_N1_B_ID            566   /* engine #3 N1/rpm */
#define    ENGINE_4_N1_B_ID            567   /* engine #4 N1/rpm */
#define    ENGINE_1_N2_B_ID            568   /* engine #1 N2/prop rpm */
#define    ENGINE_2_N2_B_ID            569   /* engine #2 N2/prop rpm */
#define    ENGINE_3_N2_B_ID            570   /* engine #3 N2/prop rpm */
#define    ENGINE_4_N2_B_ID            571   /* engine #4 N2/prop rpm */
#define    ENGINE_1_TORQUE_B_ID        572   /* engine #1 torque */
#define    ENGINE_2_TORQUE_B_ID        573   /* engine #2 torque */
#define    ENGINE_3_TORQUE_B_ID        574   /* engine #3 torque */
#define    ENGINE_4_TORQUE_B_ID        575   /* engine #4 torque */
#define    ENGINE_1_TIT_B_ID           576   /* engine #1 turbine inlet temp */
#define    ENGINE_2_TIT_B_ID           577   /* engine #2 turbine inlet temp */
#define    ENGINE_3_TIT_B_ID           578   /* engine #3 turbine inlet temp */
#define    ENGINE_4_TIT_B_ID           579   /* engine #4 turbine inlet temp */
#define    ENGINE_1_ITT_B_ID           580   /* engine #1 interturbine temp */
#define    ENGINE_2_ITT_B_ID           581   /* engine #2 interturbine temp */
#define    ENGINE_3_ITT_B_ID           582   /* engine #3 interturbine temp */
#define    ENGINE_4_ITT_B_ID           583   /* engine #4 interturbine temp */
#define    ENGINE_1_TOT_B_ID           584   /* engine #1 turbine outlet temp */
#define    ENGINE_2_TOT_B_ID           585   /* engine #2 turbine outlet temp */
#define    ENGINE_3_TOT_B_ID           586   /* engine #3 turbine outlet temp */
#define    ENGINE_4_TOT_B_ID           587   /* engine #4 turbine outlet temp */
#define    ENGINE_1_FUEL_FLOW_B_ID     588   /* engine #1 fuel flow rate */
#define    ENGINE_2_FUEL_FLOW_B_ID     589   /* engine #2 fuel flow rate */
#define    ENGINE_3_FUEL_FLOW_B_ID     590   /* engine #3 fuel flow rate */
#define    ENGINE_4_FUEL_FLOW_B_ID     591   /* engine #4 fuel flow rate */
#define    ENGINE_1_MAN_PRESS_B_ID     592   /* engine #1 manifold pressure */
#define    ENGINE_2_MAN_PRESS_B_ID     593   /* engine #2 manifold pressure */
```

```
#define    ENGINE_3_MAN_PRESS_B_ID    594    /* engine #3 manifold pressure */
#define    ENGINE_4_MAN_PRESS_B_ID    595    /* engine #4 manifold pressure */
#define    ENGINE_1_OIL_PRESS_B_ID    596    /* engine #1 oil pressure */
#define    ENGINE_2_OIL_PRESS_B_ID    597    /* engine #2 oil pressure */
#define    ENGINE_3_OIL_PRESS_B_ID    598    /* engine #3 oil pressure */
#define    ENGINE_4_OIL_PRESS_B_ID    599    /* engine #4 oil pressure */
#define    ENGINE_1_OIL_TEMP_B_ID     600    /* engine #1 oil temp */
#define    ENGINE_2_OIL_TEMP_B_ID     601    /* engine #2 oil temp */
#define    ENGINE_3_OIL_TEMP_B_ID     602    /* engine #3 oil temp */
#define    ENGINE_4_OIL_TEMP_B_ID     603    /* engine #4 oil temp */
#define    ENGINE_1_CHT_B_ID          604    /* engine #1 cylinder head temp */
#define    ENGINE_2_CHT_B_ID          605    /* engine #2 cylinder head temp */
#define    ENGINE_3_CHT_B_ID          606    /* engine #3 cylinder head temp */
#define    ENGINE_4_CHT_B_ID          607    /* engine #4 cylinder head temp */
#define    ENGINE_1_OIL_QUANT_B_ID    608    /* engine #1 oil quantity */
#define    ENGINE_2_OIL_QUANT_B_ID    609    /* engine #2 oil quantity */
#define    ENGINE_3_OIL_QUANT_B_ID    610    /* engine #3 oil quantity */
#define    ENGINE_4_OIL_QUANT_B_ID    611    /* engine #4 oil quantity */
#define    ENGINE_1_COOL_TEMP_B_ID    612    /* engine #1 cooland temp */
#define    ENGINE_2_COOL_TEMP_B_ID    613    /* engine #2 cooland temp */
#define    ENGINE_3_COOL_TEMP_B_ID    614    /* engine #3 cooland temp */
#define    ENGINE_4_COOL_TEMP_B_ID    615    /* engine #4 cooland temp */
#define    ENGINE_1_POW_RATIO_B_ID    616    /* engine #1 power ratio */
#define    ENGINE_2_POW_RATIO_B_ID    617    /* engine #1 power ratio */
#define    ENGINE_3_POW_RATIO_B_ID    618    /* engine #1 power ratio */
#define    ENGINE_4_POW_RATIO_B_ID    619    /* engine #1 power ratio */
#define    ENGINE_1_STATUS_1_B_ID     620    /* engine #1 status word 1 */
#define    ENGINE_2_STATUS_1_B_ID     621    /* engine #2 status word 1 */
#define    ENGINE_3_STATUS_1_B_ID     622    /* engine #3 status word 1 */
#define    ENGINE_4_STATUS_1_B_ID     623    /* engine #4 status word 1 */
#define    ENGINE_1_STATUS_2_B_ID     624    /* engine #1 status word 2 */
#define    ENGINE_2_STATUS_2_B_ID     625    /* engine #2 status word 2 */
#define    ENGINE_3_STATUS_2_B_ID     626    /* engine #3 status word 2 */
#define    ENGINE_4_STATUS_2_B_ID     627    /* engine #4 status word 2 */

#define    ENGINE_1_EGT_1_A_ID        628    /* engine #1 exhaust gas temp cyl 1 */
#define    ENGINE_2_EGT_1_A_ID        629    /* engine #2 exhaust gas temp cyl 1 */
#define    ENGINE_1_EGT_2_A_ID        630    /* engine #1 exhaust gas temp cyl 2 */
#define    ENGINE_2_EGT_2_A_ID        631    /* engine #2 exhaust gas temp cyl 2 */
#define    ENGINE_1_EGT_3_A_ID        632    /* engine #1 exhaust gas temp cyl 3 */
#define    ENGINE_2_EGT_3_A_ID        633    /* engine #2 exhaust gas temp cyl 3 */
#define    ENGINE_1_EGT_4_A_ID        634    /* engine #1 exhaust gas temp cyl 4 */
#define    ENGINE_2_EGT_4_A_ID        635    /* engine #2 exhaust gas temp cyl 4 */
#define    ENGINE_1_EGT_5_A_ID        636    /* engine #1 exhaust gas temp cyl 5 */
#define    ENGINE_2_EGT_5_A_ID        637    /* engine #2 exhaust gas temp cyl 5 */
#define    ENGINE_1_EGT_6_A_ID        638    /* engine #1 exhaust gas temp cyl 6 */
#define    ENGINE_2_EGT_6_A_ID        639    /* engine #2 exhaust gas temp cyl 6 */
#define    ENGINE_1_MAN_TEMP_A_ID     640    /* engine #1 manifold air temperature */
#define    ENGINE_2_MAN_TEMP_A_ID     641    /* engine #2 manifold air temperature */
#define    ENGINE_1_AMB_TEMP_A_ID     642    /* engine #1 ambient air temperature */
#define    ENGINE_2_AMB_TEMP_A_ID     643    /* engine #2 ambient air temperature */

#define    ENGINE_1_EGT_1_B_ID        644    /* engine #1 exhaust gas temp cyl 1 */
#define    ENGINE_2_EGT_1_B_ID        645    /* engine #2 exhaust gas temp cyl 1 */
#define    ENGINE_1_EGT_2_B_ID        646    /* engine #1 exhaust gas temp cyl 2 */
#define    ENGINE_2_EGT_2_B_ID        647    /* engine #2 exhaust gas temp cyl 2 */
```

```
#define    ENGINE_1_EGT_3_B_ID       648    /* engine #1 exhaust gas temp cyl 3 */
#define    ENGINE_2_EGT_3_B_ID       649    /* engine #2 exhaust gas temp cyl 3 */
#define    ENGINE_1_EGT_4_B_ID       650    /* engine #1 exhaust gas temp cyl 4 */
#define    ENGINE_2_EGT_4_B_ID       651    /* engine #2 exhaust gas temp cyl 4 */
#define    ENGINE_1_EGT_5_B_ID       652    /* engine #1 exhaust gas temp cyl 5 */
#define    ENGINE_2_EGT_5_B_ID       653    /* engine #2 exhaust gas temp cyl 5 */
#define    ENGINE_1_EGT_6_B_ID       654    /* engine #1 exhaust gas temp cyl 6 */
#define    ENGINE_2_EGT_6_B_ID       655    /* engine #2 exhaust gas temp cyl 6 */
#define    ENGINE_1_MAN_TEMP_B_ID    656    /* engine #1 manifold air temperature */
#define    ENGINE_2_MAN_TEMP_B_ID    657    /* engine #2 manifold air temperature */
#define    ENGINE_1_AMB_TEMP_B_ID    658    /* engine #1 ambient air temperature */
#define    ENGINE_2_AMB_TEMP_B_ID    659    /* engine #2 ambient air temperature */

#define    FUEL_PUMP_1_FLOW_ID       660    /* fuel pump #1 flow */
#define    FUEL_PUMP_2_FLOW_ID       661    /* fuel pump #2 flow */
#define    FUEL_PUMP_3_FLOW_ID       662    /* fuel pump #3 flow */
#define    FUEL_PUMP_4_FLOW_ID       663    /* fuel pump #4 flow */
#define    FUEL_PUMP_5_FLOW_ID       664    /* fuel pump #5 flow */
#define    FUEL_PUMP_6_FLOW_ID       665    /* fuel pump #6 flow */
#define    FUEL_PUMP_7_FLOW_ID       666    /* fuel pump #7 flow */
#define    FUEL_PUMP_8_FLOW_ID       667    /* fuel pump #8 flow */
#define    TANK_1_FUEL_QUANT_ID      668    /* tank #1 fuel quantity */
#define    TANK_2_FUEL_QUANT_ID      669    /* tank #2 fuel quantity */
#define    TANK_3_FUEL_QUANT_ID      670    /* tank #3 fuel quantity */
#define    TANK_4_FUEL_QUANT_ID      671    /* tank #4 fuel quantity */
#define    TANK_5_FUEL_QUANT_ID      672    /* tank #5 fuel quantity */
#define    TANK_6_FUEL_QUANT_ID      673    /* tank #6 fuel quantity */
#define    TANK_7_FUEL_QUANT_ID      674    /* tank #7 fuel quantity */
#define    TANK_8_FUEL_QUANT_ID      675    /* tank #8 fuel quantity */
#define    TANK_1_FUEL_TEMP_ID       676    /* tank #1 fuel temp */
#define    TANK_2_FUEL_TEMP_ID       677    /* tank #2 fuel temp */
#define    TANK_3_FUEL_TEMP_ID       678    /* tank #3 fuel temp */
#define    TANK_4_FUEL_TEMP_ID       679    /* tank #4 fuel temp */
#define    TANK_5_FUEL_TEMP_ID       680    /* tank #5 fuel temp */
#define    TANK_6_FUEL_TEMP_ID       681    /* tank #6 fuel temp */
#define    TANK_7_FUEL_TEMP_ID       682    /* tank #7 fuel temp */
#define    TANK_8_FUEL_TEMP_ID       683    /* tank #8 fuel temp */
#define    FUEL_SYS_1_PRESS_ID       684    /* fuel system #1 pressure */
#define    FUEL_SYS_2_PRESS_ID       685    /* fuel system #2 pressure */
#define    FUEL_SYS_3_PRESS_ID       686    /* fuel system #3 pressure */
#define    FUEL_SYS_4_PRESS_ID       687    /* fuel system #4 pressure */
#define    FUEL_SYS_5_PRESS_ID       688    /* fuel system #5 pressure */
#define    FUEL_SYS_6_PRESS_ID       689    /* fuel system #6 pressure */
#define    FUEL_SYS_7_PRESS_ID       690    /* fuel system #7 pressure */
#define    FUEL_SYS_8_PRESS_ID       691    /* fuel system #8 pressure */
#define    ENGINE_1_THROT_POS_A_ID   692    /* engine #1 throttle position */
#define    ENGINE_2_THROT_POS_A_ID   693    /* engine #1 throttle position */
#define    ENGINE_1_AMB_PRESS_A_ID   694    /* engine #1 ambient air pressure */
#define    ENGINE_2_AMB_PRESS_A_ID   695    /* engine #2 ambient air pressure */
#define    ENGINE_1_THROT_POS_B_ID   696    /* engine #1 throttle position */
#define    ENGINE_2_THROT_POS_B_ID   697    /* engine #1 throttle position */
#define    ENGINE_1_AMB_PRESS_B_ID   698    /* engine #1 ambient air pressure */
#define    ENGINE_2_AMB_PRESS_B_ID   699    /* engine #2 ambient air pressure */

/*
 * Power transmission system data identifier definitions.
```

```
 */

#define ROTOR_1_RPM_ID              700   /* rotor 1 RPM */
#define ROTOR_2_RPM_ID              701   /* rotor 2 RPM */
#define ROTOR_3_RPM_ID              702   /* rotor 3 RPM */
#define ROTOR_4_RPM_ID              703   /* rotor 4 RPM */
#define GEARBOX_1_RPM_ID            704   /* gearbox 1 speed */
#define GEARBOX_2_RPM_ID            705   /* gearbox 2 speed */
#define GEARBOX_3_RPM_ID            706   /* gearbox 3 speed */
#define GEARBOX_4_RPM_ID            707   /* gearbox 4 speed */
#define GEARBOX_5_RPM_ID            708   /* gearbox 5 speed */
#define GEARBOX_6_RPM_ID            709   /* gearbox 6 speed */
#define GEARBOX_7_RPM_ID            710   /* gearbox 7 speed */
#define GEARBOX_8_RPM_ID            711   /* gearbox 8 speed */
#define GEARBOX_1_OIL_PRESS_ID      712   /* gearbox 1 oil pressure */
#define GEARBOX_2_OIL_PRESS_ID      713   /* gearbox 2 oil pressure */
#define GEARBOX_3_OIL_PRESS_ID      714   /* gearbox 3 oil pressure */
#define GEARBOX_4_OIL_PRESS_ID      715   /* gearbox 4 oil pressure */
#define GEARBOX_5_OIL_PRESS_ID      716   /* gearbox 5 oil pressure */
#define GEARBOX_6_OIL_PRESS_ID      717   /* gearbox 6 oil pressure */
#define GEARBOX_7_OIL_PRESS_ID      718   /* gearbox 7 oil pressure */
#define GEARBOX_8_OIL_PRESS_ID      719   /* gearbox 8 oil pressure */
#define GEARBOX_1_OIL_TEMP_ID       720   /* gearbox 1 oil temperature */
#define GEARBOX_2_OIL_TEMP_ID       721   /* gearbox 2 oil temperature */
#define GEARBOX_3_OIL_TEMP_ID       722   /* gearbox 3 oil temperature */
#define GEARBOX_4_OIL_TEMP_ID       723   /* gearbox 4 oil temperature */
#define GEARBOX_5_OIL_TEMP_ID       724   /* gearbox 5 oil temperature */
#define GEARBOX_6_OIL_TEMP_ID       725   /* gearbox 6 oil temperature */
#define GEARBOX_7_OIL_TEMP_ID       726   /* gearbox 7 oil temperature */
#define GEARBOX_8_OIL_TEMP_ID       727   /* gearbox 8 oil temperature */
#define GEARBOX_1_OIL_QTY_ID        728   /* gearbox 1 oil temperature */
#define GEARBOX_2_OIL_QTY_ID        729   /* gearbox 2 oil oil quantity */
#define GEARBOX_3_OIL_QTY_ID        730   /* gearbox 3 oil oil quantity */
#define GEARBOX_4_OIL_QTY_ID        731   /* gearbox 4 oil oil quantity */
#define GEARBOX_5_OIL_QTY_ID        732   /* gearbox 5 oil oil quantity */
#define GEARBOX_6_OIL_QTY_ID        733   /* gearbox 6 oil oil quantity */
#define GEARBOX_7_OIL_QTY_ID        734   /* gearbox 7 oil oil quantity */
#define GEARBOX_8_OIL_QTY_ID        735   /* gearbox 8 oil oil quantity */

/*
 * Hydraulic system data identifier definitions.
 */

#define HYDRAULIC_1_PRESS_ID        800   /* hydraulic system 1 pressure */
#define HYDRAULIC_2_PRESS_ID        801   /* hydraulic system 2 pressure */
#define HYDRAULIC_3_PRESS_ID        802   /* hydraulic system 3 pressure */
#define HYDRAULIC_4_PRESS_ID        803   /* hydraulic system 4 pressure */
#define HYDRAULIC_5_PRESS_ID        804   /* hydraulic system 5 pressure */
#define HYDRAULIC_6_PRESS_ID        805   /* hydraulic system 6 pressure */
#define HYDRAULIC_7_PRESS_ID        806   /* hydraulic system 7 pressure */
#define HYDRAULIC_8_PRESS_ID        807   /* hydraulic system 8 pressure */
#define HYDRAULIC_1_TEMP_ID         808   /* hydraulic system 1 temperature */
#define HYDRAULIC_2_TEMP_ID         809   /* hydraulic system 2 temperature */
#define HYDRAULIC_3_TEMP_ID         810   /* hydraulic system 3 temperature */
#define HYDRAULIC_4_TEMP_ID         811   /* hydraulic system 4 temperature */
#define HYDRAULIC_5_TEMP_ID         812   /* hydraulic system 5 temperature */
```

```
#define HYDRAULIC_6_TEMP_ID            813    /* hydraulic system 6 temperature */
#define HYDRAULIC_7_TEMP_ID            814    /* hydraulic system 7 temperature */
#define HYDRAULIC_8_TEMP_ID            815    /* hydraulic system 8 temperature */
#define HYDRAULIC_1_QTY_ID             816    /* hydraulic system 1 fluid quantity */
#define HYDRAULIC_2_QTY_ID             817    /* hydraulic system 2 fluid quantity */
#define HYDRAULIC_3_QTY_ID             818    /* hydraulic system 3 fluid quantity */
#define HYDRAULIC_4_QTY_ID             819    /* hydraulic system 4 fluid quantity */
#define HYDRAULIC_5_QTY_ID             820    /* hydraulic system 5 fluid quantity */
#define HYDRAULIC_6_QTY_ID             821    /* hydraulic system 6 fluid quantity */
#define HYDRAULIC_7_QTY_ID             822    /* hydraulic system 7 fluid quantity */
#define HYDRAULIC_8_QTY_ID             823    /* hydraulic system 8 fluid quantity */


/*
 * Electric system data identifier definitions.
 */

#define    AC_VOLTAGE_1_ID             900    /* AC system #1 voltage */
#define    AC_CURRENT_1_ID             910    /* AC current */
#define    DC_VOLTAGE_1_ID             920    /* DC system #1 voltage */
#define    DC_VOLTAGE_2_ID             921    /* DC system #2 voltage */
#define    DC_CURRENT_1_ID             930    /* DC current */
#define    PROP_ICEGUARD_1_CURR_ID     940    /* prop iceguard amps */
#define    ECU_VOLT_1_A_ID             950    /* engine control unit #1 voltage */
#define    ECU_VOLT_2_A_ID             951    /* engine control unit #2 voltage */
#define    ECU_VOLT_3_A_ID             952    /* engine control unit #3 voltage */
#define    ECU_VOLT_4_A_ID             953    /* engine control unit #4 voltage */
#define    ECU_VOLT_1_B_ID             954    /* engine control unit #1 voltage */
#define    ECU_VOLT_2_B_ID             955    /* engine control unit #2 voltage */
#define    ECU_VOLT_3_B_ID             956    /* engine control unit #3 voltage */
#define    ECU_VOLT_4_B_ID             957    /* engine control unit #4 voltage */

/*
 * Navigation system data identifier definitions.
 */

#define ACT_NAV_LATITUDE_ID            1000   /* active nav system latitude */
#define ACT_NAV_LONGITUDE_ID           1001   /* active nav system longitude */
#define ACT_NAV_HEIGHT_ID              1002   /* active nav system height */
#define ACT_NAV_ALTITUDE_ID            1003   /* active nav system altitude */
#define ACT_NAV_GND_SPEED_ID           1004   /* active nav system ground speed */
#define ACT_NAV_TT_ID                  1005   /* active nav system true track */
#define ACT_NAV_MT_ID                  1006   /* active nav system magnetic track */
#define ACT_NAV_XTK_ID                 1007   /* active nav system cross track err. */
#define ACT_NAV_TKE_ID                 1008   /* active nav system track err. angle */
#define ACT_NAV_TTG_ID                 1009   /* active nav system time-to-go */
#define ACT_NAV_ETA_ID                 1010   /* active nav system ETA */
#define ACT_NAV_ETE_ID                 1011   /* active nav system ETE */

#define NAV_WP_IDENT_1_ID              1012   /* WP identifier char 0-3 */
#define NAV_WP_IDENT_2_ID              1013   /* WP identifier char 4-7 */
#define NAV_WP_IDENT_3_ID              1014   /* WP identifier char 8-11 */
#define NAV_WP_IDENT_4_ID              1015   /* WP identifier char 12-15 */
#define NAV_WP_TYPE_ID                 1016   /* WP route segment type */
                                              /* also VOR/VOR-DME/DME/NDB/INTRSCT */
#define NAV_WP_LATITUDE_ID             1017   /* waypoint latitude */
```

```
#define NAV_WP_LONGITUDE_ID        1018   /* waypoint longitude */
#define NAV_WP_MIN_ALT_ID          1019   /* minimum WP altitude */
#define NAV_WP_MIN_FL_ID           1020   /* minimum WP flight level */
#define NAV_WP_MIN_RDRHGT_ID       1021   /* minimum WP radar height */
#define NAV_WP_MIN_HGT_WGS_ID      1022   /* minimum WP height above ellipsoid */
#define NAV_WP_MAX_ALT_ID          1023   /* maximum WP altitude */
#define NAV_WP_MAX_FL_ID           1024   /* maximum WP flight level */
#define NAV_WP_MAX_RDRHGT_ID       1025   /* maximum WP radar height */
#define NAV_WP_MAX_HGT_WGS_ID      1026   /* maximum WP height above ellipsoid */
#define NAV_WP_PLAN_ALT_ID         1027   /* planned WP altitude */
#define NAV_WP_PLAN_FL_ID          1028   /* planned WP flight level */
#define NAV_WP_PLAN_RDRHGT_ID      1029   /* planned WP radar height */
#define NAV_WP_PLAN_HGT_WGS_ID     1030   /* planned WP height above ellipsoid */
#define NAV_WP_DIST_ID             1031   /* WP Distance to Waypoint */
#define NAV_WP_TTG_ID              1032   /* WP time-to-go */
#define NAV_WP_ETA_ID              1033   /* WP ETA */
#define NAV_WP_ETE_ID              1034   /* WP ETE */
#define NAV_WP_TO_FR_FLG_ID        1035   /* FROM/TO/APCHNG/OFF flag */

#define GPS_AC_LATITUDE_ID         1036   /* GPS aircraft latitude */
#define GPS_AC_LONGITUDE_ID        1037   /* GPS aircraft longitude */
#define GPS_AC_HGT_ABV_EL_ID       1038   /* GPS aircraft height above WGS 84 */
#define GPS_AC_GND_SPEED_ID        1039   /* GPS ground speed */
#define GPS_AC_TT_ID               1040   /* GPS true track */
#define GPS_AC_MT_ID               1041   /* GPS magnetic track */
#define GPS_AC_XTK_ID              1042   /* GPS cross track error */
#define GPS_AC_TKE_ID              1043   /* GPS track error angle */
#define        GPS_AC_GS_DEV_ID    1044   /* GPS GS deviation */
#define GPS_PRED_RAIM_ID           1045   /* GPS predicted RAIM */
#define GPS_VERT_POS_TOL_ID        1046   /* vertical figure of merit */
#define GPS_HOR_POS_TOL_ID         1047   /* horizontal figure of merit */
#define GPS_MODE_ID                1048   /* GPS operation mode */

#define INS_AC_LATITUDE_ID         1049   /* INS aircraft latitude */
#define INS_AC_LONGITUDE_ID        1050   /* INS aircraft longitude */
#define INS_AC_HGT_ABV_EL_ID       1051   /* INS aircraft height above WGS 84 */
#define INS_AC_GND_SPEED_ID        1052   /* INS ground speed */
#define INS_AC_TT_ID               1053   /* INS true track */
#define INS_AC_MT_ID               1054   /* INS magnetic track */
#define INS_AC_XTK_ID              1055   /* INS cross track error */
#define INS_AC_TKE_ID              1056   /* INS track error angle */
#define INS_VERT_POS_TOL_ID        1057   /* vertical figure of merit */
#define INS_HOR_POS_TOL_ID         1058   /* horizontal figure of merit */

#define AUX_AC_LATITUDE_ID         1059   /* AUX aircraft latitude */
#define AUX_AC_LONGITUDE_ID        1060   /* AUX aircraft longitude */
#define AUX_AC_HGT_ABV_EL_ID       1061   /* AUX aircraft height above WGS 84 */
#define AUX_AC_GND_SPEED_ID        1062   /* AUX ground speed */
#define AUX_AC_TT_ID               1063   /* AUX true track */
#define AUX_AC_MT_ID               1064   /* AUX magnetic track */
#define AUX_AC_XTK_ID              1065   /* AUX cross track error */
#define AUX_AC_TKE_ID              1066   /* AUX track error angle */
#define AUX_VERT_POS_TOL_ID        1067   /* vertical figure of merit */
#define AUX_HOR_POS_TOL_ID         1068   /* horizontal figure of merit */

#define        MAG_HEADING_ID      1069   /* magnetic heading */
```

```
#define      RADIO_HEIGHT_ID         1070  /* radio height */
#define DME_1_DISTANCE_ID            1071  /* DME #1 distance */
#define DME_2_DISTANCE_ID            1072  /* DME #2 distance */
#define DME_3_DISTANCE_ID            1073  /* DME #3 distance */
#define DME_4_DISTANCE_ID            1074  /* DME #4 distance */
#define DME_1_TTG_ID                 1075  /* DME #1 time to station */
#define DME_2_TTG_ID                 1076  /* DME #2 time to station */
#define DME_3_TTG_ID                 1077  /* DME #3 time to station */
#define DME_4_TTG_ID                 1078  /* DME #4 time to station */
#define DME_1_GND_SPD_ID             1079  /* DME #1 ground speed */
#define DME_2_GND_SPD_ID             1080  /* DME #2 ground speed */
#define DME_3_GND_SPD_ID             1081  /* DME #3 ground speed */
#define DME_4_GND_SPD_ID             1082  /* DME #4 ground speed */
#define ADF_1_REL_BRG_ID             1083  /* ADF #1 relative bearing */
#define ADF_2_REL_BRG_ID             1084  /* ADF #2 relative bearing */
#define ADF_3_REL_BRG_ID             1085  /* ADF #3 relative bearing */
#define ADF_4_REL_BRG_ID             1086  /* ADF #4 relative bearing */
#define      ILS_1_LOC_DEV_ID        1087  /* ILS #1 LOC deviation */
#define      ILS_2_LOC_DEV_ID        1088  /* ILS #2 LOC deviation */
#define      ILS_3_LOC_DEV_ID        1089  /* ILS #3 LOC deviation */
#define      ILS_4_LOC_DEV_ID        1090  /* ILS #4 LOC deviation */
#define      ILS_1_GS_DEV_ID         1091  /* ILS #1 GS deviation */
#define      ILS_2_GS_DEV_ID         1092  /* ILS #2 GS deviation */
#define      ILS_3_GS_DEV_ID         1093  /* ILS #3 GS deviation */
#define      ILS_4_GS_DEV_ID         1094  /* ILS #4 GS deviation */
#define      FD_1_PITCH_DEV_ID       1095  /* flight director #1 pitch deviation */
#define      FD_2_PITCH_DEV_ID       1096  /* flight director #2 pitch deviation */
#define      FD_1_ROLL_DEV_ID        1097  /* flight director #1 roll deviation */
#define      FD_2_ROLL_DEV_ID        1098  /* flight director #2 roll deviation */
#define DECISION_HEIGHT_ID           1099  /* decision height */
#define VHF_1_FREQ_ID                1100  /* VHF COM #1 frequency */
#define VHF_2_FREQ_ID                1101  /* VHF COM #2 frequency */
#define VHF_3_FREQ_ID                1102  /* VHF COM #3 frequency */
#define VHF_4_FREQ_ID                1103  /* VHF COM #4 frequency */
#define VOR_ILS_1_FREQ_ID            1104  /* VOR ILS #1 frequency */
#define VOR_ILS_2_FREQ_ID            1105  /* VOR ILS #2 frequency */
#define VOR_ILS_3_FREQ_ID            1106  /* VOR ILS #3 frequency */
#define VOR_ILS_4_FREQ_ID            1107  /* VOR ILS #4 frequency */
#define ADF_1_FREQ_ID                1108  /* ADF #1 frequency */
#define ADF_2_FREQ_ID                1109  /* ADF #2 frequency */
#define ADF_3_FREQ_ID                1110  /* ADF #3 frequency */
#define ADF_4_FREQ_ID                1111  /* ADF #4 frequency */
#define DME_1_FREQ_ID                1112  /* DME #1 frequency */
#define DME_2_FREQ_ID                1113  /* DME #2 frequency */
#define DME_3_FREQ_ID                1114  /* DME #3 frequency */
#define DME_4_FREQ_ID                1115  /* DME #4 frequency */
#define XPDR_1_CODE_ID               1116  /* transponder #1 code */
#define XPDR_2_CODE_ID               1117  /* transponder #2 code */
#define XPDR_3_CODE_ID               1118  /* transponder #3 code */
#define XPDR_4_CODE_ID               1119  /* transponder #4 code */

#define DESIRED_TRK_MAG_ID           1120  /* desired track angle */
#define MAG_VARIATION_ID             1121  /* magnetic variation */
#define SEL_GPATH_ANGLE_ID           1122  /* selected glidepath angle */
#define SEL_RWY_HDG_ID               1123  /* selected runway heading */
#define COMPUTED_VERT_VEL_ID         1124  /* computed vertical velocity */
```

```
#define SEL_COURSE_ID               1125   /* selected course */

#define VOR_1_RADIAL_ID             1126   /* VOR #1 radial bearing */
#define VOR_2_RADIAL_ID             1127   /* VOR #2 radial bearing */
#define VOR_3_RADIAL_ID             1128   /* VOR #3 radial bearing */
#define VOR_4_RADIAL_ID             1129   /* VOR #4 radial bearing */

#define TRUE_EAST_VEL_ID            1130   /* true east velocity */
#define TRUE_NORTH_VEL_ID           1131   /* true north velocity */
#define TRUE_UP_VEL_ID              1132   /* true up velocity */
#define     TRUE_HEADING_ID         1133   /* true heading angle */

/*
 * Landing gear system data identifier definitions.
 */

#define     GEAR_SWITCH_ID          1175   /* gear lever switches */
#define     GEAR_LIGHTS_WOW_ID      1176   /* gear lever indicator lights/WOW */
#define     TIRE_PRESS_1_ID         1177   /* landing gear #1 tire pressure */
#define     TIRE_PRESS_2_ID         1178   /* landing gear #2 tire pressure */
#define     TIRE_PRESS_3_ID         1179   /* landing gear #3 tire pressure */
#define     TIRE_PRESS_4_ID         1180   /* landing gear #4 tire pressure */
#define     BRAKE_PAD_THCK_1_ID     1181   /* landing gear #1 brakepad thickness */
#define     BRAKE_PAD_THCK_2_ID     1182   /* landing gear #2 brakepad thickness */
#define     BRAKE_PAD_THCK_3_ID     1183   /* landing gear #3 brakepad thickness */
#define     BRAKE_PAD_THCK_4_ID     1184   /* landing gear #4 brakepad thickness */

/*
 * Miscellaneous data identifier definitions.
 */

#define UTC_ID                      1200   /* Universal Time Coordinated */
#define CABIN_P_ID                  1201   /* cabin pressure */
#define CABIN_ALT_ID                1202   /* cabin altitude */
#define CABIN_T_ID                  1203   /* cabin temperature */
#define CG_LONG_ID                  1204   /* longitudinal center of gravity */
#define CG_LAT_ID                   1205   /* lateral center of gravity */
#define DATE_ID                     1206   /* date */
#define TOP_MARKER_ID               1207   /* flight data recording TOP marker */

#define ENGINE_1_HOURS_A_ID         1208   /* engine #1 operating hours */
#define ENGINE_2_HOURS_A_ID         1209   /* engine #2 operating hours */
#define ENGINE_3_HOURS_A_ID         1210   /* engine #3 operating hours */
#define ENGINE_4_HOURS_A_ID         1211   /* engine #4 operating hours */
#define ENGINE_1_HOURS_B_ID         1212   /* engine #1 operating hours */
#define ENGINE_2_HOURS_B_ID         1213   /* engine #2 operating hours */
#define ENGINE_3_HOURS_B_ID         1214   /* engine #3 operating hours */
#define ENGINE_4_HOURS_B_ID         1215   /* engine #4 operating hours */

#define ECU_1_HOURS_A_ID            1216   /* ECU #1 operating hours */
#define ECU_2_HOURS_A_ID            1217   /* ECU #2 operating hours */
#define ECU_3_HOURS_A_ID            1218   /* ECU #3 operating hours */
#define ECU_4_HOURS_A_ID            1219   /* ECU #4 operating hours */
#define ECU_1_HOURS_B_ID            1220   /* ECU #1 operating hours */
#define ECU_2_HOURS_B_ID            1221   /* ECU #2 operating hours */
#define ECU_3_HOURS_B_ID            1222   /* ECU #3 operating hours */
```

```
#define ECU_4_HOURS_B_ID                 1223   /* ECU #4 operating hours */

#define     LANE_A_TIMESTAMP_ID          1224   /* ECU A Timestamp */
#define     LANE_B_TIMESTAMP_ID          1228   /* ECU B Timestamp */

#define     TOT_FUEL_FLOW_ID             1226   /* Total fuel flow */
#define     TOT_FUEL_CONSUMED_ID         1227   /* Total fuel consumed */

#define     FUEL_PRESS_RED_A_ID          1234   /* Fuel press - manifold press ECU A */
#define     FUEL_PRESS_RED_B_ID          1235   /* Fuel press - manifold press ECU B */

/*
 * Traffic Collision Avoidance Systems (TCAS) data identifier definitions.
 */

#define FLARM_MSG_1_ID                   1300   /* FLARM message #1 */
#define FLARM_MSG_2_ID                   1301   /* FLARM message #2 */
#define FLARM_MSG_3_ID                   1302   /* FLARM message #3 */
#define FLARM_MSG_4_ID                   1303   /* FLARM message #4 */
#define FLARM_MSG_5_ID                   1304   /* FLARM message #5 */
#define FLARM_MSG_6_ID                   1305   /* FLARM message #6 */
#define FLARM_MSG_7_ID                   1306   /* FLARM message #7 */
#define FLARM_MSG_8_ID                   1307   /* FLARM message #8 */
#define FLARM_SENS_MSG_1_ID              1308   /* FLARM sensor box message #1 */
#define FLARM_SENS_MSG_2_ID              1301   /* FLARM sensor box message #2 */
#define FLARM_SENS_MSG_3_ID              1302   /* FLARM sensor box message #3 */
#define FLARM_SENS_MSG_4_ID              1303   /* FLARM sensor box message #4 */
#define FLARM_SENS_MSG_5_ID              1304   /* FLARM sensor box message #5 */
#define FLARM_SENS_MSG_6_ID              1305   /* FLARM sensor box message #6 */
#define FLARM_SENS_MSG_7_ID              1306   /* FLARM sensor box message #7 */
#define FLARM_SENS_MSG_8_ID              1307   /* FLARM sensor box message #8 */

/*
 * User defined data identifier definitions.
 */

#define     FIRST_USER_DEFINED_ID        1500
#define     LAST_USER_DEFINED_ID         1799

/*
 * End of file.
 */
```

Appendix D Frequency Allocations USA 2016

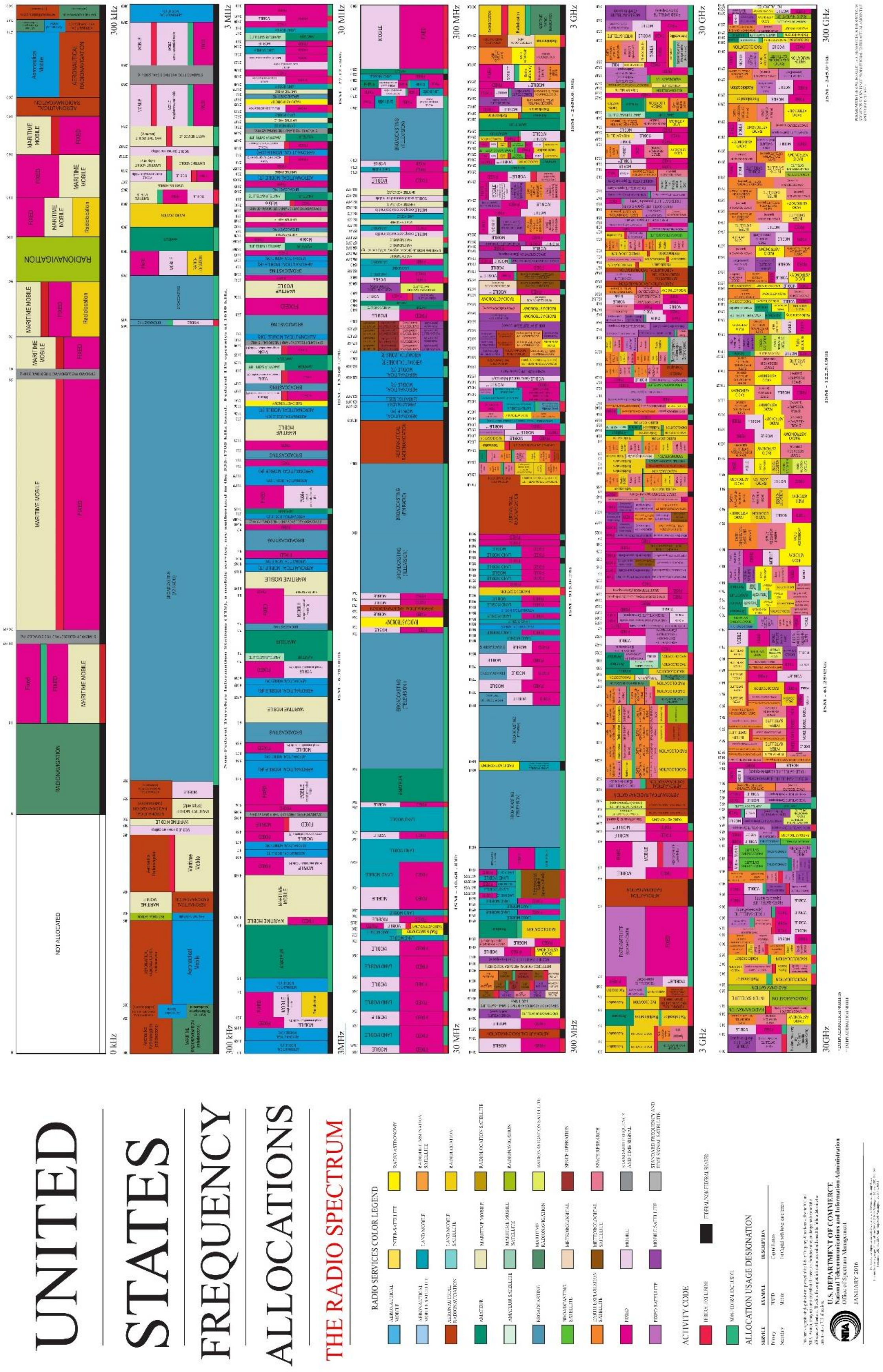


Figure D.1 Frequency allocations USA – the radio spectrum (Commerce 2016)

Appendix E Fuel Level Indicator Arduino Code

```
const float offset = 0.035;
int pwmVal[] = {8, 15, 21, 27, 34, 40, 46, 52, 58, 65};

//linker tank
int stateL = 0;
const int  pwmL = 6;
float links[] = {0.0, 1.09, 1.56, 2.02, 2.47, 2.93, 3.41, 3.89, 4.44, 4.80,
99};

//rechter tank
int stateR = 0;
const int  pwmR = 5;
float rechts[] = {0.0, 1.14, 1.62, 2.05, 2.50, 2.95, 3.41, 3.89, 4.44,
4.79, 99};


void setLedL(int state){
    analogWrite(pwmL, pwmVal[state]);
    // Serial.println(state+1);
    // Serial.println(" ");
}

void setLedR(int state){
    analogWrite(pwmR, pwmVal[state]);
    // Serial.println(state+1);
    // Serial.println();
}


// the setup routine runs once when you press reset:
void setup() {
  pinMode(pwmL, OUTPUT);
  pinMode(pwmR, OUTPUT);

  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);

  int sensorValueL = analogRead(A0);
  float Lvoltage = sensorValueL * (5.0 / 1023.0);

  int sensorValueR = analogRead(A1);
  float Rvoltage = sensorValueR * (5.0 / 1023.0);

}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValueL = analogRead(A0);
  int sensorValueR = analogRead(A1);

  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0
- 5V):
  float Lvoltage = sensorValueL * (5.0 / 1023.0);
  float Rvoltage = sensorValueR * (5.0 / 1023.0);
```

```
  if( Lvoltage > (links[stateL+1] + offset)){
    stateL = stateL + 1;
    // Serial.println(Lvoltage);
    setLedL(stateL);
  } else if ( Lvoltage < (links[stateL] - offset)){
    stateL = stateL - 1;
    // Serial.println(Lvoltage);
     setLedL(stateL);
  }

  if( Rvoltage > (rechts[stateR+1] + offset)){
    stateR = stateR + 1;
    // Serial.println(Rvoltage);
    setLedR(stateR);
  } else if ( Rvoltage < (rechts[stateR] - offset)){
    stateR = stateR - 1;
    // Serial.println(Rvoltage);
    setLedR(stateR);
  }

}
```

Appendix F Supplementary Device Test DAeC

	Deutscher Aero Club e.V. **Luftsportgeräte-Büro** Beauftragter des Bundesministeriums für Verkehr *Authorized Representative*	**Ergänzende Musterprüfung Digitale Instrumente (Glascockpit)**

Merkblatt

Digitale Instrumente in LUFTSPORTGERÄTEN

Grundsätzlich können digitale Instrumente unabhängig von der Mindestausrüstung (mechanische Fahrtmesser, Höhenmesser, Kompass) als Zusatzausrüstung eingebaut werden. Die analogen Instrumente waren durch die jeweilige Musterzulassung zugelassen.

Nachfolgend geben wir eine Information, welche Forderungen für eine ergänzende Musterzulassung erfüllt sein müssen, wenn digitale Instrumente anstelle der Mindestausrüstung eingebaut werden.

Digitale Grund-Instrumente (Fahrtmesser, Höhenmesser, Kompass) dürfen in Luftsportgeräten dann als Hauptinstrumente eingesetzt werden, wenn sie in Genauigkeit und Funktionssicherheit mindestens gleichwertig wie mechanische Instrumente und für das Muster zugelassen sind.

Die ergänzende Musterzulassung für den Einsatz digitaler Instrumente als Hauptinstrumente ist durch den Hersteller / Musterbetreuer zu beantragen. Digitale Instrumente werden in diesen Fällen im Gerätekennblatt als Ausrüstung oder als zugelassene Ausrüstungsvariante eingetragen.

Der Hersteller / Musterbetreuer muss die Einhaltung folgender Spezifikationen der digitalen Instrumente gewährleisten und die Musterakte mit den Nachweisen vervollständigen:

1. **Bei Ausfall des Hauptstromnetzes an Bord muss die elektrische Stromversorgung aller digitalen Instrumente von einem unabhängigen Backup-System für mindestens 30 Minuten sichergestellt sein.**
2. **Die Aktivierung und der Status des Backup-Systems muss durch eine sichtbare Anzeige am Instrumentenbrett dauerhaft erkennbar sein.**
3. **Die Batterien des Backup-Systems müssen nach einer völligen Entladung innerhalb von 3 Stunden wieder voll aufladbar sein.**
4. **Das Backup-System und die digitalen Instrumente müssen nach dem Einbau vom Hersteller oder einem von ihm Beauftragten auf Funktionstüchtigkeit überprüft werden.**

	Deutscher Aero Club e.V. Luftsportgeräte-Büro Beauftragter des Bundesministeriums für Verkehr *Authorized Representative*	Ergänzende Musterprüfung Digitale Instrumente (Glascockpit)

5. **Die Anzeige des Backup-Systems und die digitalen Instrumente müssen bei hellem Tageslicht, bei direkter Sonnenstrahlung und in der Dämmerung bei einem Blickwinkel von mehr als 30° zweifelsfrei von der Position des Piloten ablesbar sein.**
6. **Das Backup-System und die digitalen Instrumente müssen bei Betriebstemperaturen von -10°C bis +50°C und Lagerungstemperaturen von - 30°C bis +60°C funktionstüchtig sein.**
7. **Das Backup-System und die Instrumente müssen bei einer relativen Luftfeuchtigkeit von bis zu 80% (bei einer Lufttemperatur von 20°C, ohne Kondensation) funktionstüchtig sein.**
8. **Die digitalen Instrumente müssen kalibrierbar sein.**
9. **Im Betriebshandbuch müssen Arbeitsweise und Bedienung der digitalen Instrumente beschrieben sein. Außerdem sind bei Geräteausfall einzuhaltende Verfahren aufzuführen.**
10. **Die Geschwindigkeitsbereiche des digitalen Fahrtmessers müssen gemäß der Werte des Luftsportgerätes farblich markiert sein.**
11. **Auf die Pflicht zur jährlichen Nachprüfung der digitalen Instrumente muss im Betriebshandbuch hingewiesen werden.**
12. **Die Prüfverfahren für das Backup-System und die digitalen Instrumente müssen in den Wartungsanleitungen aufgeführt sein.**
13. **Die technischen Daten und Bezeichnungen der zugelassenen Backup-Systeme und der digitalen Instrumente müssen im Betriebshandbuch angegeben sein.**

Appendix G QTR Garmin G5

Test, Garmin G5

QTR Nr.: QTR_GarminG5_17_08_2021

Serial Nr.: 4JQ046934

Erstellt: Philipp Gmelin

List of Revisions

Revision	*Datum*	*Changes*
A	17.08.2021	first release

Tabel of Contents

1 Introduction

The report shows the test procedure for a supplemental type certification for the use of a digital Garmin G5 instead of analog instruments. The "Ergänzende Musterprüfung Digitaler Instrumente (Glascockpit)" published on 22.06.2017 by "Deutscher Aero Club e.V. Luftsportgeräte-Büro" is used.

1.1 *Requirements for the Garmin G5 according to DAeC e.V.*

1. In case of failure of the main power supply on board, the electrical power supply of of all digital instruments must be ensured by an independent backup system for at least 30 minutes.
2. The activation and status of the backup system shall be permanently indicated by a visible display on the instrument panel.
3. The batteries of the backup system must be fully rechargeable within 3 hours after a complete discharge.
4. The backup system and the digital instruments must be checked for proper functioning by the manufacturer or his authorized representative after installation.
5. The display of the backup system and the digital instruments must be visible in bright daylight, in direct sunlight and at dusk at an angle of view of more than viewing angle of more than 30° can be read without doubt from the pilot's position.
6. The backup system and digital instruments must be functional at operating temperatures from -10°C to +50°C and storage temperatures from - 30°C to +60°C.
7. The backup system and instruments must function at a relative humidity of up to 80% (at an air temperature of 20°C, without condensation).
8. The digital instruments must be calibrateable.
9. The operating manual must describe the functioning and operation of the digital instruments. In addition, procedures to be followed in the event of instrument failure shall be listed
10. The speed ranges of the digital airspeed indicator must be color-coded in accordance with the of the airborne equipment.
11. The requirement for annual inspection of the digital instruments must be stated in the operations manual.
12. The test procedures for the backup system and the digital instruments must be listed in the maintenance manuals.
13. The technical data and designations of the approved backup systems and the digital instruments must be specified in the operating manual.

1.2 *Solutions and Test Setup*

1. The G5 lithium-ion Battery Pack (Part Number: 010-12493-01) is designed for the Garmin G5 electronic flight instrument and provides up to 4 hours of emergency operation. In the test, the main power supply of the Garmin G5 must be switched off and it must continue to run with the backup battery for at least 30 minutes.
2. When the main power is switched off, the estimated time until the battery is empty is displayed. When the on-board power supply is switched on again, a lightning bolt symbol appears over the battery icon. The battery level is then displayed in percentage.

3. After a complete discharge, the G5 battery pack must be fully charged within 3 hours. In the test, the on-board power supply is switched on after a complete discharge and the charging time is recorded.
4. The manufacturer will check all important functions after installation of the Garmin G5, as well as the automatic backup system in the event of an onboard power failure.
5. In a test, the display must be readable without problems up to an angle of 30° in direct sunlight. The Garmin G5 automatically adjusts the backlight intensity to the lighting conditions in Auto mode.
6. In this test, the Garmin G5 is brought to -30°C in the switched-off state, after 2 minutes it should be warmed up to -10°C. When the temperature is reached, the device should be switched on in backup mode. The device must correctly display the heading (artificial horizon) and settings can be made in the menu (change a unit) without problems. After successfully passing the test, the Garmin G5 is turned off and warmed up to +60°C. The temperature is maintained for 2 minutes and then cooled down to +50°C. When the temperature is reached, the function test of the Garmin G5 is performed again in backup mode.
7. The Garmin G5 is exposed to a relative humidity of 80% at a temperature of 20°C. The device must run in backup mode for 2 minutes in this environment and the display must not show any condensation.
8. The Garmin G5 can be calibrated as described in Chapter 8 of the Installation Manual.
9. The most important operating elements must be described in the Flywhale operating manual, as well as the behavior in the event of a device failure.
10. The speed ranges are correctly displayed by the Garmin G5 after entering the different airspeeds in the device configuration.
11. The operating manual refers to the obligation of annual rechecking of the digital instruments.
12. When the aircraft is put into operation, the main switch of the aircraft's power supply system is flipped, after which all electronic devices in the aircraft turn on. To verify that the Garmin G5 backup system is working, simply turn the main power switch off. The Garmin G5 should automatically go into backup mode and stay on. After this test, the main power switch can be turned back on and the Garmin G5 should indicate that it is being powered again from the onboard power supply. This can be done during the pre-flight check but also after a maintenance to check the correct activation of the backup mode of the Garmin G5.

13. The technical data of the backup systems are recorded in the operating manual.

1.3 Personnel Performing and Witnessing the Test

The test shall be witnessed by additional personnel with equivalent qualification, but shall not directly be involved into the test. It is acceptable that the personnel for testing and witnessing are recruited from the company's staff. It is not required that test or witnessing personnel hold a special license, such as aircraft inspector (e.g. EASA Part 66, German Prüfer Klasse 1/3/5, etc.).

1.4 Documentation

The test shall sufficiently be documented with cameras, statements, and/or records. Following items need to be documented:

- conformity of the test specimen (Test Plan NR. FW TP 32 0010 R01)
- photos of the test setup prior to test
- photos of the scenery after the test
- eventually video of the test

2 Test Protocol

2.1 Backup System Minimum Runtime

After fully charging the Garmin G5, a minimum runtime of more than 50 minutes was measured in backup mode.

Test passed (>30 min) ☒

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758

After the "Environment and Humidity Test" (picture through oven window) at 15:54, the Garmin G5 was left running with a start capacity of 38% (Figure 2.1). After 50 min at 16:44, the Garmin G5 had a remaining runtime of 38 min (Figure 2.2).

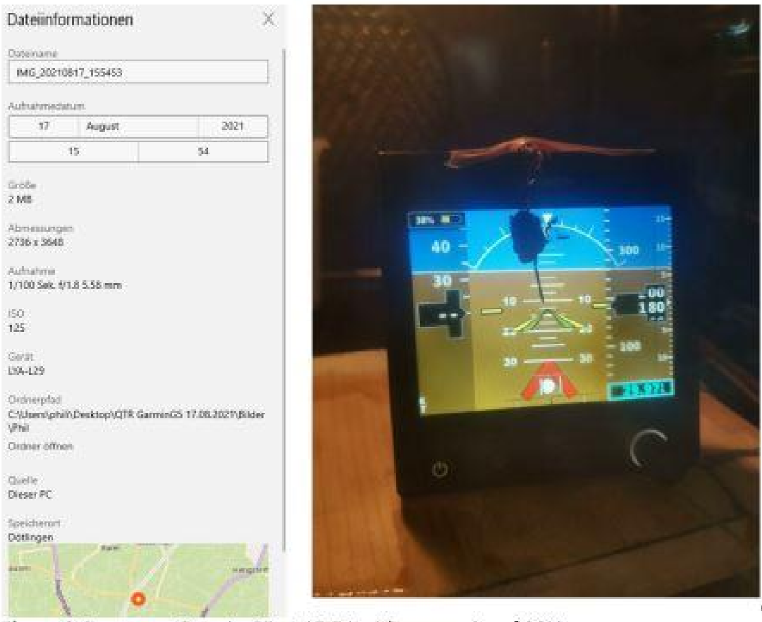


Figure 2.1 Garmin G5 at 15:54 with a capacity of 38%.

Figure 2.2 Garmin G5 at 16:44 with a with a remaining runtime of 38 min.

-Quality Test Report-
Uniplanes

Revision A

2.2 Backup System Minimum Recharge Time

After the Garmin G5 was fully discharged, a complete recharge time of less than 120 minutes was measured.

Test passed (<180 min) ☒

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758

2.3 *Display Reading Angle*

☒ The display brightness reacts to the brightness of the environment

☒ The display can still be read up to a reading angle of 30° with a bright direct beam of a lamp.

Test passed ☒

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758

The reading angle of the Garmin G5 is surprisingly good. When fully illuminated with a bright lamp, the display could still be read at over 80° angle (Figure 2.3 and 2.4).

Figure 2.3 Garmin G5 reading angle at about 55°

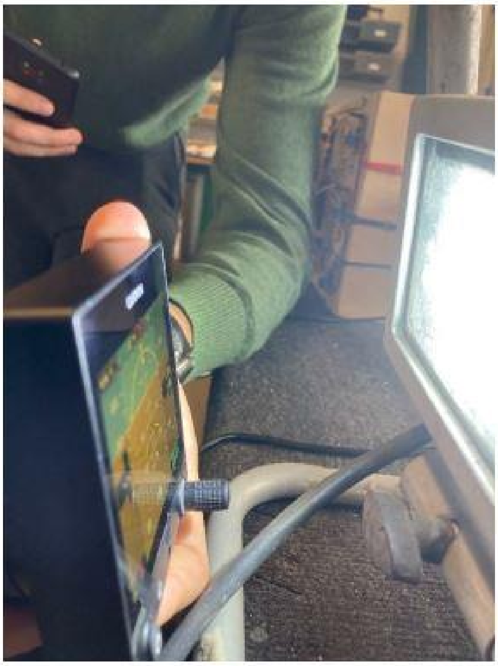


Figure 2.4 Garmin G5 reading angle over 80°

2.4 Operation Temperatures

☒ The Garmin G5 is brought to -30°C in the switched-off state

☒ After 2 minutes it is warmed up to -10°C

☒ The temperature is reached, the device is switched on in backup mode

☒ The device must correctly display the heading (artificial horizon) and settings can be made in the menu (change a unit) without problems

☒ The Garmin G5 is turned off and warmed up to +60°C

☒ The temperature is maintained for 2 minutes and then cooled down to +50°C

☒ The temperature is reached, the device is switched on in backup mode

☒ The device must correctly display the heading (artificial horizon) and settings can be made in the menu (change a unit) without problems

Test passed ☒

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758

The test was performed in an insulated box with dry ice cooling (Figure 2.6). Two temperature sensors were attached to the Garmin G5, one directly on the display and one directly on the battery (Figure 2.5). This made it possible to record the temperature of the device.

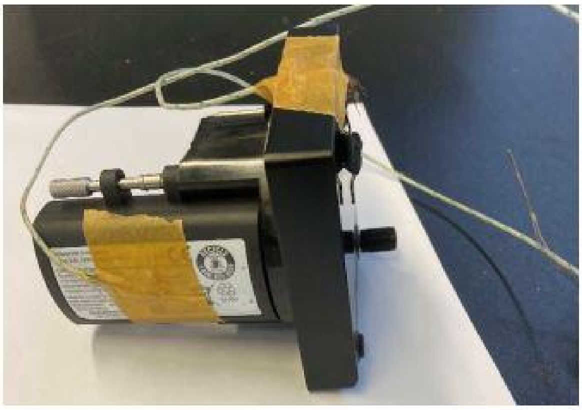


Figure 2.5 Garmin G5 with temperature sensors

To ensure an even temperature in the box, the air was circulated with small fans. The Garmin G5 was cooled to below -30 °C when it was turned off.

Figure 2.6 Garmin G5 in the insulated box with dry ice cooling

Since the sensors are not directly in the device but have a slight distance to the device, the boundary layer temperature was measured at the device, which is why strong temperature changes can be seen in the temperature curve when the lid of the box was opened to set a temperature below -10 °C. In the following figure 2.7 it can be seen that the temperature below -30 °C was kept for about 2 minutes and 50 seconds. After that, the lid was opened to switch on the device and to set a temperature below -10°. The lid had to be opened several times, which caused the temperature sensor in front of the display to fluctuate as warmer air was let into the box. (Figure 2.7)

Gerätename: Flywhale		17.08.2021 15:14:40			Seite 1/1
Startzeit: 17.08.2021 14:39:28		Minimum	Maximum	Mittelwert	Grenzwerte
Endzeit: 17.08.2021 15:09:40	Oberfläche Garmin [°C]	-36,3	21,7	-9,684	-195,0/1000,0
Messkanäle: 3	Oberfläche Backup Akku [°C]	-39,2	23,3	-13,595	-195,0/1000,0
Messwerte: 454	Umgebungstemperatur [°C]	20,7	21,3	21,089	-195,0/1000,0
SN 40718071					

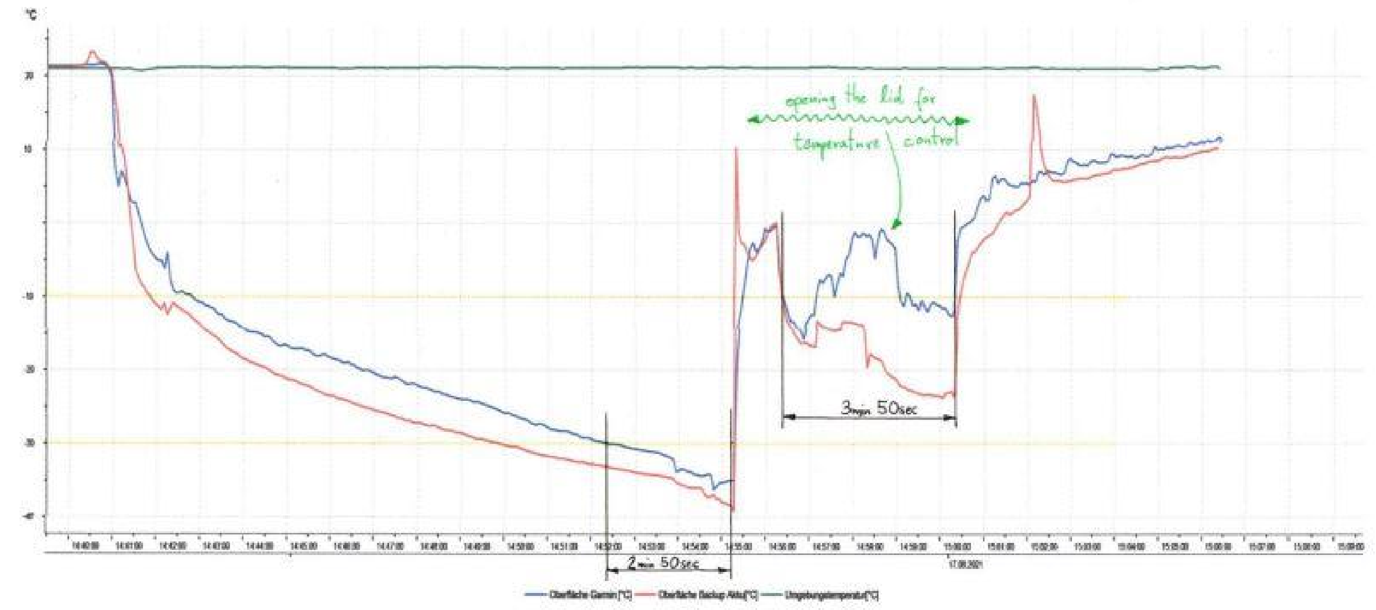


Figure 2.7 Garmin G5 temperature diagram of the display, backup battery and environment temperature (-30 °C and -10 °C)

By moving the box, the functionality was proven by the movement of pitch and roll in the Garmin G5 display (Figure 2.8).

Figure 2.8 Garmin G5 in switched-on state with dry ice cooling

In the next step, the Garmin G5 was heated to over 60 °C in the oven and held for about 4 minutes and 20 seconds (Figure 2.9 and 2.10).

Gerätename: Flywhale		17.08.2021 16:08:51			Seite 1/1
Startzeit: 17.08.2021 15:25:52		Minimum	Maximum	Mittelwert	Grenzwerte
Endzeit: 17.08.2021 15:38:00	Oberfläche Garmin [°C]	26,7	76,1	55,642	-195,0/1000,0
Messkanäle: 3	Oberfläche Backup Akku [°C]	23,8	78,1	59,038	-195,0/1000,0
Messwerte: 183	Umgebungstemperatur [°C]	----	----	----	-195,0/1000,0
SN 40718071					

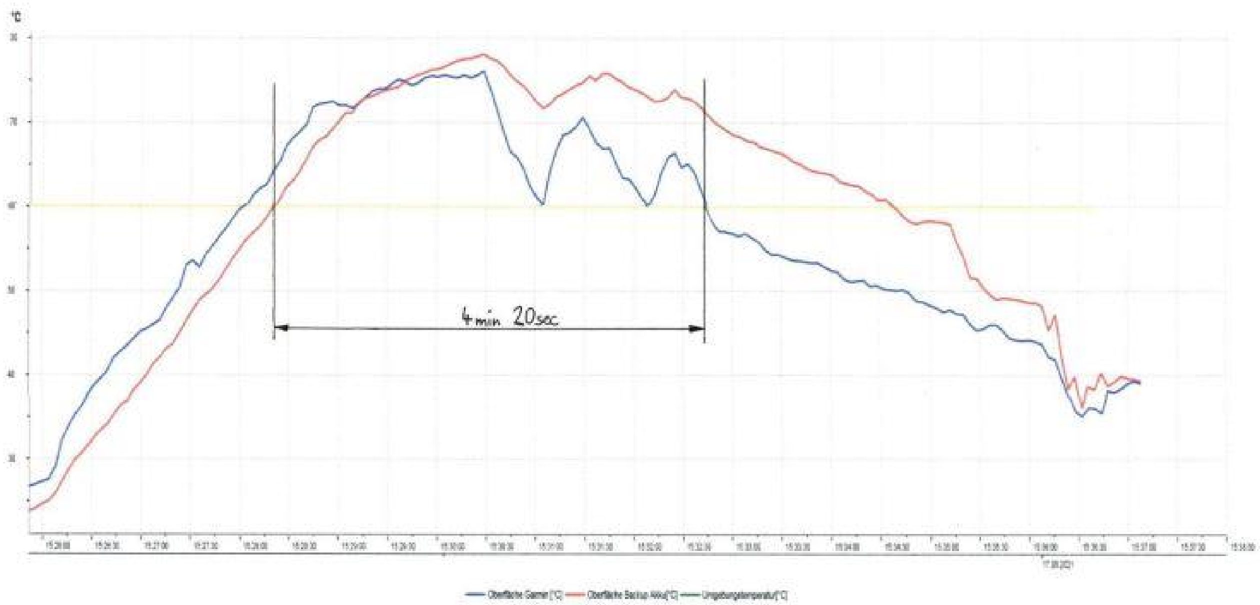


Figure 2.9 Garmin G5 temperature diagram of the display and the backup battery (+60 °C)

Figure 2.10 Garmin G5 in the oven at more than 60 °C

Then the unit was switched on at about 50 °C and held at temperature for 2 min and 30 seconds (Figure 2.11 and 2.12). After that, the oven was opened and immediately proved to be functional by moving roll and pitch and changing the unit in the menu.

Figure 2.11 Garmin G5 in the oven at about 50 °C when switched on

Gerätename: Flywhale		21.08.2021 12:02:40			Seite 1/1
Startzeit: 17.08.2021 15:41:14		Minimum	Maximum	Mittelwert	Grenzwerte
Endzeit: 17.08.2021 16:02:26	Oberfläche Garmin [°C]	32,2	51,6	42,233	-195,0/1000,0
Messkanäle: 3	Oberfläche Backup Akku [°C]	30,6	50,4	42,476	-195,0/1000,0
Messwerte: 319	Umgebungstemperatur [°C]	----	----	----	-195,0/1000,0
SN 40718071					

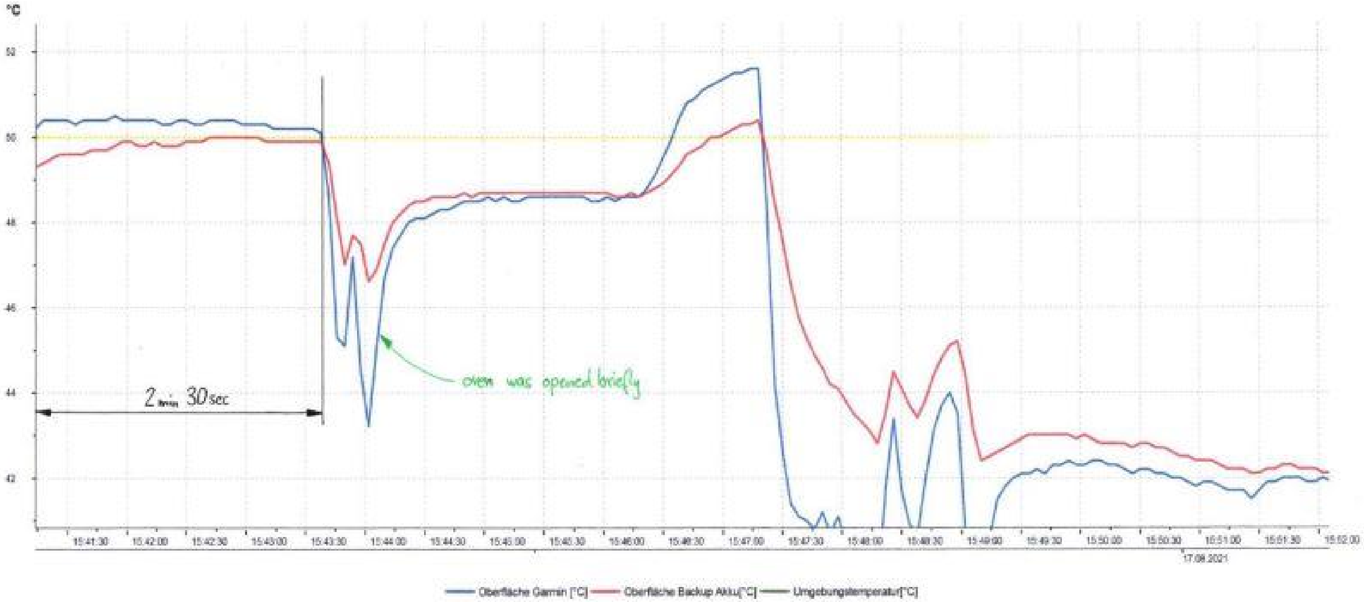


Figure 2.12 Garmin G5 temperature diagram of the display and the backup battery (+50 °C)

2.5 Environment and Humidity Test

☒ The Garmin G5 is put into the prepared box/oven with about 20°C and 80% relative humidity

☒ After 2 minutes of runtime in backup mode, there is no sign of condensation

Test passed ☒

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758

The test was performed directly after the previous test. An ultrasonic water nebulizer was turned on in the oven, which could quickly increase the relative humidity (Figure 2.13). At about 39 °C, the relative humidity of 80 % was then reached and held for 3 minutes 30 seconds (Figure 2.14).

Figure 2.13 Garmin G5 in the oven with increasing relative humidity

Gerätename: Flywhale		21.08.2021 12:04:01			Seite 1/1
Startzeit: 17.08.2021 15:41:14		Minimum	Maximum	Mittelwert	Grenzwerte
Endzeit: 17.08.2021 16:02:26	Oberfläche Garmin [°C]	32,2	51,6	42,233	-195,0/1000,0
Messkanäle: 3	Oberfläche Backup Akku [°C]	30,6	50,4	42,476	-195,0/1000,0
Messwerte: 319	Umgebungstemperatur [°C]	----	----	----	-195,0/1000,0
SN 40718071					

Figure 2.14 Garmin G5 in the oven with a relative humidity of about 80%

Then the oven was opened to see if the Garmin G5 had any signs of condensation. The Garmin G5 was completely dry and had no signs of condensation (Figure 2.15).

Figure 2.15 Video snapshot when opening the oven at 82% relative humidity at about 39 °C

3 Result

The Garmin G5 has successfully passed all required tests and can be installed in an ultralight aircraft without hesitation instead of the minimum equipment.

Serial Nr.: 4JQ046934

Date: 17.08.2021
Prepared by: Philipp Gmelin
Dipl. Ing. (FH) Jan Kaminski
Andreas Sosnowski

Philipp Gmelin

Jan Kaminski

UNIPLANES GmbH
Schubertstrasse 14
D-60325 Frankfurt/Main
HRB120758