EUROPEAN ORGANISATION
FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

EUROCONTROL EXPERIMENTAL CENTRE

PC BASED AIRCRAFT DATA LINK PROCESSOR

EEC Note No. 14/99

Project SUR-3-E1

Issued : December 1999

REPORT DOCUMENTATION PAGE

Reference : EEC Note No. **14/99**	**Security Classification :** Unclassified
Originator : EEC - COM (Communications)	**Originator (Corporate Author) Name / Location** : EUROCONTROL Experimental Centre B.P. 15 F -91222 Brétigny sur Orge Cedex Telephone : +33 (0)1 69 88 75 00
Sponsor : EATCHIP Development Directorate DED.3	**Sponsor** (**Contract Authority**) **Name** / **Location** : EUROCONTROL Agency Rue de la Fusée, 96 B -1130 BRUXELLES Telephone : +32 (0)2 729 9011

TITLE :

PC based Aircraft Data Link Processor

Author	Date	Pages	Figures	Tables	Annexes	References
P.Brun	12/99	X + 85	14	-	10	13

EATCHIP Task Specification	Project	Task No. Sponsor	Period
-	SUR-3-E1	SUR-3-E1	3/98-12/98

Distribution Statement :
(a) Controlled by : Head of COM and P. HUNT (SUR-3-E1 project leader)
(b) Special Limitations : None
(c) Copy to NTIS : NO

Descriptors (keywords) :
MODE S, ADLP, GDLP, FITAMS, AIRBORNE ROUTER

Abstracts :
As no ADLPs exist yet on the market, it was decided to build an ADLP using a PC for experiments in the laboratory or in test aircraft. The software for the ADLP was developed by TUB in ‘C’ language under an Eurocontrol contract. Eurocontrol implemented all the ARINC hardware and software interfaces with the other equipment such as the Mode S Transponder, the Trial ATN router, broadcast buses, and GNSS / GPS receivers.

This document has been collated by mechanical means. Should there be missing pages, please report to:

EUROCONTROL Experimental Centre
Publications Office
B.P. 15
91222 – BRETIGNY-SUR-ORGE CEDEX
France

EEC Note No. 14/99
Issued : December 1999

PC Based Aircraft Data Link Processor

by

P. Brun

Summary

As no ADLPs exist yet on the market, it was decided to build an ADLP using a ruggedised or industrial PC for experiments in the laboratory or in test aircraft. The software for the ADLP was developed by TUB in ‘C’ language under an Eurocontrol contract.

Eurocontrol implemented all the ARINC hardware and software interfaces with the other equipment such as the Mode S Transponder, the Trial ATN router, broadcast buses, and GNSS / GPS receivers.

Having a PC based ADLP presented several advantages for experimental work. It was easily mountable on a test aircraft or could be readily installed in any laboratory, was portable and mobile. The other great advantage was that EEC was in full control of the hardware and software used and upgrades, corrections and modifications could be applied in minutes and not in months as is the case with industry based certified aircraft equipment, not to mention the costs involved.

Since building the first PC based ADLP there has been great interest in this product and we have supplied several ADLPs to those interested, namely DERA, DFS, STNA for various projects including Mode S sub-network SARPs Validation, FITAMS flight trials and T-GDLP, Air / Ground router implementation in the ATN environment, as well as ADS-B and DAPs and data-link trials.

TABLE OF CONTENTS

1. INTRODUCTION **1**

1.1 OBJECTIVE 1
1.2 GENERAL OVERVIEW 1

2. PC-BASED ADLP **3**

2.1. HARDWARE DESCRIPTION 3
2.2. ADVANCED ARINC CARDS AND FIRMWARE 5
2.2.1. Firmware ARINC 429 Broadcast *6*
2.2.2. Firmware ARINC 718 protocol *6*
2.2.3. Firmware GPFT (Williamsburg) *7*
2.3. SOFTWARE DESCRIPTION 8
2.3.1. TUB ADLP software tasks *8*
2.3.2. Transponder and broadcast bus process *10*
2.3.3. ARINC 429 GPFT and broadcast bus process *10*
2.3.4. Transponder up-link and down-link processes *11*

3 USER'S MANUAL **12**

3.1. LOGIN 12
3.2 AUTOMATIC START-UP OF ADLP 12
3.3. MANUAL START 12
3.4. CONFIGURATION FILES 13
3.4.1. ADLP.CFG *13*
3.4.2. FITAMS.CFG *15*
3.5. TASKS 16
3.6. ADLP LOG FILES 16
3.7. STOP ADLP 16
3.8. COPY TO AND FROM FLOPPY 17
3.9. INTERNET WEB SITES 17

FRENCH TRANSLATION **19**

1. INTRODUCTION 19
2. LE LABORATOIRE DE BRETIGNY 20
2.1. Intégration du sous-réseau ATN avec le Mode S *21*
2.2. ADS-Broadcast et DAPs *21*
2.3. Passerelle pour les interfaces ARINC 718 et Williamsburg *22*

ABBREVIATIONS **23**

REFERENCES **25**

LINKS **26**

APPENDIX A BRETIGNY LABORATORY **29**

A.1. LABORATORY APPLICATIONS 29
A.2. ATN AND MODE S SUB-NETWORK INTEGRATION 29
A.3. ADS-BROADCAST AND DAPS 32

A.4. GATEWAY FOR 718 AND WILLAMSBURG INTERFACES 34
A.4.1. Objective....................34
A.4.2. Transponder interface format 34
A.4.3. ISO8208 interface format....................34

APPENDIX B TRT TRANSPONDER TEST RACK (1) 39

APPENDIX C TRT TRANSPONDER TEST RACK (2) 43

C.1. ARINC 600 CONNECTOR 43
C.2. FRONT PANEL AND TEST BUTTON 43

APPENDIX D TAR / ADLP CABLE 47

APPENDIX E GPFT INTERFACE SPECIFICATION....................51

APPENDIX F OPERATING SYSTEM FREEBSD....................57

APPENDIX G ADVANCED ARINC CARD DRIVER 61

G.1. DEFINITION AND EQUIVALENCES 61
G.2. FUNCTIONS AND PROCEDURES 62

APPENDIX H ADLP INSTALLATION....................67

H.1. PREPARE 67
H.2. PROCEDURE 67
H.3. CONFIGURE THE SYSTEM 69
H.4. KERNEL AND ARINC CARD CONFIGURATION....................72

APPENDIX I SARPS VALIDATION PROJECT 75

I.1. INTRODUCTION 75
I.2. TEST BENCH....................75
I.3. SSE SCENARIO....................76
I.4. SVC SCENARIO....................77

APPENDIX J DFS MODE S SUB-NETWORK....................81

J.1. INTRODUCTION 81
J.2. INVESTIGATED DATA LINK ENVIRONMENT 81
J.3. INVESTIGATED DATA LINKS....................81
J.4. INVESTIGATED PARAMETERS 82
J.5. APPROACH 83

LIST OF ILLUSTRATIONS

FIGURE 1 FUNCTIONAL ELEMENTS OF THE MODE S SUBNETWORK 2
FIGURE 2 ARINC PROTOCOLS STACK 3
FIGURE 3 PC BASED ADLP 4
FIGURE 4 ADVANCED ARINC CARD 5
FIGURE 5 SOFTWARE ARCHITECTURE 11
FIGURE 6 FITAMS - ATN MODE S AND SDU 30
FIGURE 7 ADS-BROADCAST AND DAPS 33
FIGURE 8 GATEWAY FOR SVC AND MODE S 35
FIGURE 9 TRANSPONDER TEST RACK 39
FIGURE 10 TRANSPONDER FRONT PANEL 43
FIGURE 11 ADLP / TAR CABLE 47
FIGURE 12 DERA DATALINK TEST BENCH 75
FIGURE 13 DFS OVERALL DATA LINK ENVIRONMENT 81
FIGURE 14 DFS SUB-NETWORK 84

1. Introduction

1.1 Objective

To provide a flexible ADLP for test and trials purposes, where EEC have full control over all hard ware and software aspects.

1.2 General overview

The purpose of this document is to provide an overall description of the software and the hardware architectural design and a user manual of the PC Based Aircraft Data Link Processor including the description of the configurations installed at Brétigny, DERA, DFS and STNA Mode-S laboratories.

The ADLP nucleus software was written by TUB and EEC designed the hardware and software concerning the ARINC interfaces.

An Airborne Data Link Processor (ADLP) provides intercommunication functions necessary to interface :

- The ATN router on the aircraft side using a standard ISO 8208 and Williamsburg interface
- The Mode S transponder (XPDR) using an ARINC 718 protocol
- The Mode S Specific Services Entities using an ARINC 429-15 GPFT protocol
- The GNSS or Aircraft data equipment using an ARINC 429 broadcast protocol

The ADLP standards provide for full compliance of the Mode S sub-network with ATN requirements. Basically this sub-network is composed of a Mode S transponder (XPDR), a Mode S interrogator (MODE S Radar) and a Ground Data Link Processor (GDLP) which is a peer process of the ADLP on the ground side.

The Aeronautical Telecommunication Network (ATN) is an inter-network architecture that provides the interchange of digital data between ground/ground, air/ground and avionics data sub-networks (Mode S, Satellite/ADS, and others).

This interchange of information based on several sub-networks (MODE-S, Satellite/ADS, and others) adopts common interface services and protocols based on the International Organisation for Standardisation (ISO) Open Systems Interconnection (OSI) reference model.

Basically a Mode S ground station communicates with a Mode S transponder over a RF data link. Messages with a length of 56 or 112 bits are transmitted between ground station and transponder. The transponder will forward these messages to the ADLP. The frame-processor of the ADLP will link messages together to form frames of data. For the down-link direction the ADLP will split frames into small messages and forward those to the transponder. The frame-processor of the ground is now responsible of linking the messages into complete frames.

Frames leaving the frame-processor of the ADLP are interpreted on the contents of the Mode S Sub-network header. This header can contain basically two types of protocols:

- Mode S Sub-network protocol
- Mode S Specific protocol.

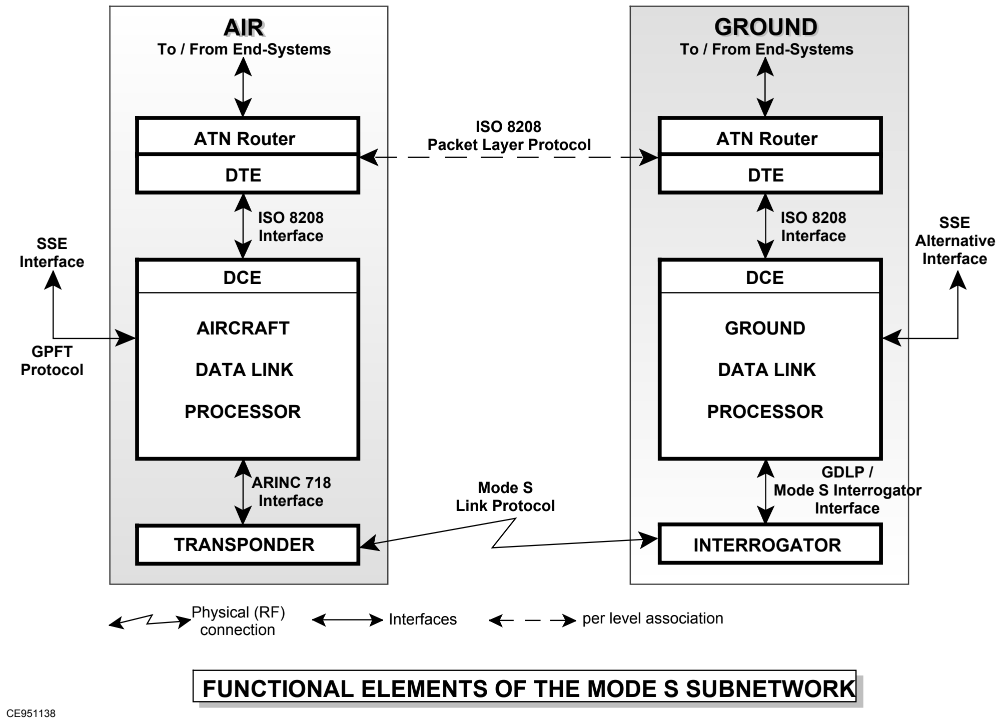


Figure 1 Functional elements of the Mode S Subnetwork

This router may be either mobile (aircraft) or fixed (ground based). The ATN router selects ground and air/ground sub-networks based on user-specified communication requirement and sub-network availability to route data information.

This action is transparent to the user who therefore does not need to know the area of coverage of particular sub-networks nor to change communications procedures depending upon the sub-network that is in use.

2. PC-Based ADLP

2.1. Hardware description

The ADLP is an ADVANTECH Pentium PC with up to four EEC ARINC cards inserted on the PC ISA bus. These ARINC cards are designed in house by EEC Brétigny and provide the ARINC 429 interfaces required.

The ADLP is connected to the aircraft sub systems with standard ARINC429 buses. For this purpose the ADLP has a total of 8 ARINC429 input and 16 ARINC429 output channels available.

On top of the physical ARINC429 channels the following ARINC protocols are available for communication with aircraft sub-systems :

- ARINC718 protocol used for communication with the transponder.
- ARINC429-15 protocol. This is the General Purpose File transfer protocol used in conjunction with ISO interface and SSI interface (known also as Williamsburg)
- ARINC429-broadcast. This is the normal ARINC429 protocol and is used in conjunction with the ADI (Aircraft Data Input) application of the ADLP

The ARINC718 and ARINC429-15 protocols need one physical input and one physical output channel for operation.

The ARINC429 broadcast channels require only physical input channels.

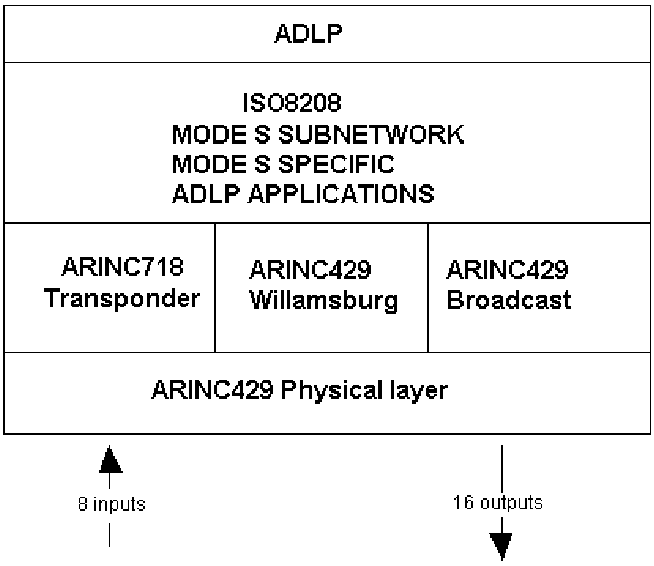


Figure 2 ARINC protocols stack

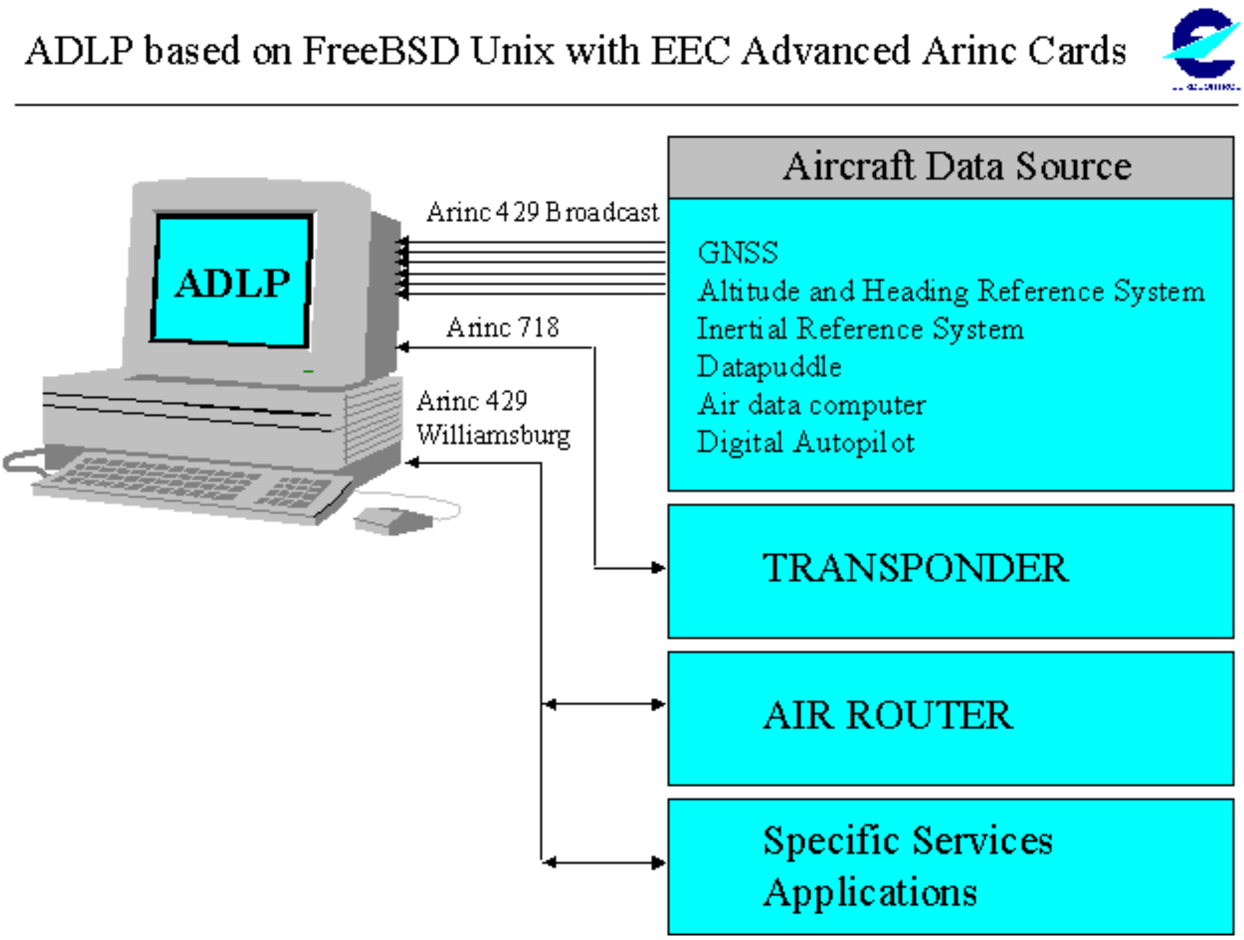


Figure 3 **PC Based ADLP**

2.2. Advanced ARINC Cards and firmware

Note: For more details see **Eurocontrol Notes N° 17/94 and 27/97**

The EEC Advanced ARINC card exists in several versions but with the same functionality, the latest version uses surface mounted integrated circuits.

For this card, a lot of software was produced for dialogue applications. Each firmware is developed and downloaded from the PC in the dual ported memory of the ARINC card.

As the ARINC card contains a common memory area which is accessible to both the PC and the local onboard processor, the PC may access the shared memory at any time to fetch or store data.

The host can stop, reset and start the card via an I/O port, the ARINC card can also interrupt the PC

The Advanced ARINC card has a local Intel 8088 processor and it makes all protocol handling on board and is able to respect the requested time-outs for ARINC718 (4 ms).

For the ADLP we use three different firmware or combinations of them:

- ARINC718 protocol
- ARINC429 general purpose file transfer “Williamsburg”
- ARINC429 broadcast

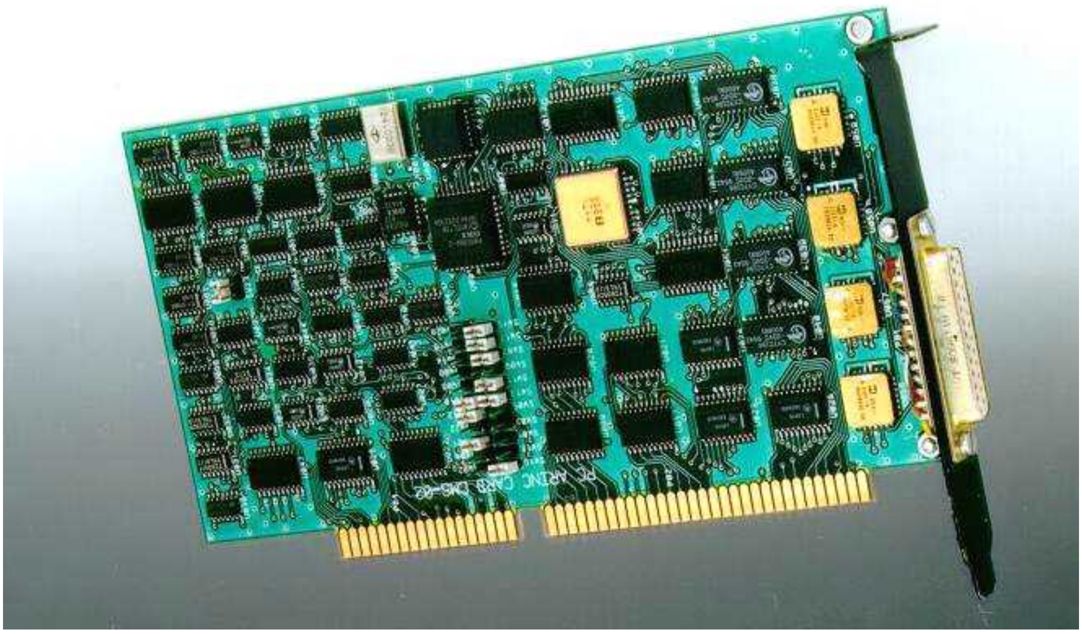


Figure 4 Advanced Arinc card

2.2.1. Firmware ARINC 429 Broadcast

The ADI firmware can read two aircraft data buses simultaneously in order to update the GICB registers in the transponder including BDS 0,5 and 0,6 with CPR coding for the extended long squitter (ADS-B) or DAPs as BDS 4,0, 5,0, 6,0.

The interface between PC and the ARINC card is made in predefined memory areas.

The firmware has for each of the two input channels, all possible labels occurring on an aircraft broadcast bus.

According to a configuration file, the PC downloads these tables for the labels defined for each channel (20 per channel max)

2.2.2. Firmware ARINC 718 protocol

This firmware establishes the interface to the Mode S transponder according to the ARINC 718 characteristic.

The ADLP software configures and runs the firmware on the card according to the following definition:

The following items define the interface:

Control, Altitude and flight ident.
The ARINC 718 firmware can generate the ARINC words necessary for a transponder without the need of a control box. There are default values in the firmware and the ADLP software can also modify these values and then change altitude, Mode A or flight ident. according to the ARINC 429 DITS representation.

Speed
This parameter defines the speed of the ARINC 718 interface, for the transponder the valid value is HIGH.

Activate handshake
This parameter defines whether or not if it necessary to implement a handshake between ADLP software and the 718 firmware on the card. Valid values are 0 or 1. Taking into account the data rate of this connection, handshaking is used to avoid losing transponder messages.

Clear the handshake
If the handshake is active and when the ADLP receives up-link message from the transponder it must clear the handshake to receive another up-link, if not the reception will be blocked and the 718 firmware send “not clear to send” indefinitely.

Send buffer and send byte count
This buffer is used to send data to the transponder according to the length specified by the “send byte count” register.

Received buffer and received byte count
This buffer is used to receive data from the transponder under control of the interruption handler, the length of the data received is given by the "receive byte count" register.

Note : The firmware implements all the ARINC 718 characteristic, and the ARINC protocol is completely transparent to the ADLP.

2.2.3. Firmware GPFT (Williamsburg)

The firmware for the General Purpose File Transfer is used for standard point to point connections with an airborne router using ISO 8208 protocol and with an MSP airborne system using GICB or MSP protocols.

The ADLP software configures and runs the firmware on the card according to the following definition :

System address label
This parameter is a specific Williamsburg field, it identified the remote machine (TAR). This value is defined according to the Trial ATN router configuration file (SAL = 224 octal)

General File Identifier
This parameter is a type field (GFI) to differentiate between MSP protocol and ISO8208 protocol (ISO= 1 and MSP=3).

Speed
This parameter defines the speed of the ARINC 429 interface, according to the remote machine configuration. Valid values are HIGH or LOW.

Calling system
This parameter is used to define who is the initiator of the communication.(ADLP or TAR) Valid values are 1 or 0 (1= ADLP initiates the communication).

Send buffer and send byte count
This buffer is used to send data to the Williamsburg interface according to the length specified by the "send byte count" register.

Received buffer and received byte count
This buffer is used to receive data from the Williamsburg interface under the interruption handler, the length of the data received is given by the "receive byte count" register.

Activate handshake
This parameter defines whether or not if it necessary to implement a handshake between the ADLP software and the Williamsburg firmware on the card. Valid values are 0 or 1. The handshake is used to prevent the loss of messages.

Clear the handshake
If the handshake is active and when the ADLP receives up-link message from the Williamsburg interface it must clear the handshake to receive another up-link, if not the reception will be blocked.

2.3. Software description

The ADLP is designed to run under control of a multitasking operating system, to transport packets from one process to another we use a FIFO First in / First out buffer having a shared memory area.

The following processes constitute the ADLP software functions:

- ↳ TUB tasks
- ↳ Transponder interface
- ↳ ARINC 429 broadcast interface
- ↳ ARINC 429 GPFT
- ↳ Transponder up-link task
- ↳ Transponder down-link task
- ↳ SSE task

2.3.1. TUB ADLP software tasks

Interfaces

To communicate with this software we use 4 sockets interfaces to deliver data to and from the TUB software, these sockets are the following :

- ↳ up-link transponder socket
- ↳ down-link transponder socket
- ↳ ISO 8208 sockets
- ↳ SSE sockets

Action

The TUB software is split into different functional modules as specified by the SARPs. Each module is represented by a software module, consisting of several functions, where one or more software modules form a process.

The SARPs specify the following functional modules :

- ↳ DCE
- ↳ Re-formatter
- ↳ ADCE
- ↳ Re-sequencing
- ↳ Frame processor
- ↳ Mode S Specific Services Entity
- ↳ Mode S Sub-network Management

These modules are implemented as several processes running in parallel. The table below shows the functional modules and the related processes

Functional module	Process	Module	Comment
DCE	DRA000	FTP DCE	ISO8208 conversion to/from an internal format DCE processing
Reformatter	DRA000	RFM	Reformatting
ADCE	DRA000	ACE	ADCE and parts of Sub-network management
Resequencing	DRA000	ACE	Re-sequencing is a part of ADCE
Frame Processor	DWN000 DEC000 LNK000		Down-link part of frame processor; includes some sub-network management functions ARINC 718 decoder of frame processor Segment Linker and De-multiplexer of frame processor
Mode S Specific Services Entity	SSE000		MSP, GICB, and BC processing
Mode S Sub-network Management			Located in DRA000 and DWN000
System Administration	ADP000 ERR000		ADLP start and debug control process ADLP error handler

The Mode S Sub-network Management functions are implemented in the down-link part of the frame processor and in the ADCE part of the DRA000 process.

2.3.2. Transponder and broadcast bus process

This process has the following functions:

- It is a gateway between the ARINC card driver and the up-link / down-link processes

  - It receives under interruption ARINC words from the 718 firmware using the driver and then forwards a frame into a FIFO towards the up-link process

  - It receives data from down-link process using read procedure in FIFO and then forwards the packet to the ARINC board

- It generates a control word, altitude and flight ident. to the transponder using the ARINC board

- It receives ARINC broadcast words from ARINC bus to update the ARINC broadcast table in share memory area

During the initialisation, it enables the handshake with card, and loads the ARINC labels table to the ARINC board according to a configuration file list. This labels list is used as ARINC word filter by the board on broadcast bus.

2.3.3. ARINC 429 GPFT and broadcast bus process

This process has the following functions :

- It is a gateway between ARINC card driver and the ISO8208 socket interface used to communicate with TUB ADLP software :

  - It receives under interruption a GPFT frame from the GPFT firmware using the driver and then forwards it towards the ISO8208 socket interface if GFI = 1

  - It receives under interruption a GPFT frame from the GPFT firmware using the driver and then forwards it towards the MSP socket interface if GFI = 3

  - It receives data from socket interface and forwards it to the GPFT ARINC board filling in the GFI value

- It receives ARINC broadcast words from the ARINC bus to update the ARINC broadcast table in the shared memory area

During the initialisation, it enables the handshake with card, and loads the ARINC labels table to the ARINC board according to a configuration file list. This label list is used as an ARINC word filter by the board on the broadcast bus.

2.3.4. Transponder up-link and down-link processes

This processes has two functions :

↳ The down-link process receives data from TUB software using socket interface and forwards it into the down-link FIFO towards the transponder interface process.

↳ The up-link process receives data from the transponder interface process using a FIFO and forwards it into the up-link socket interface used by TUB ADLP software.

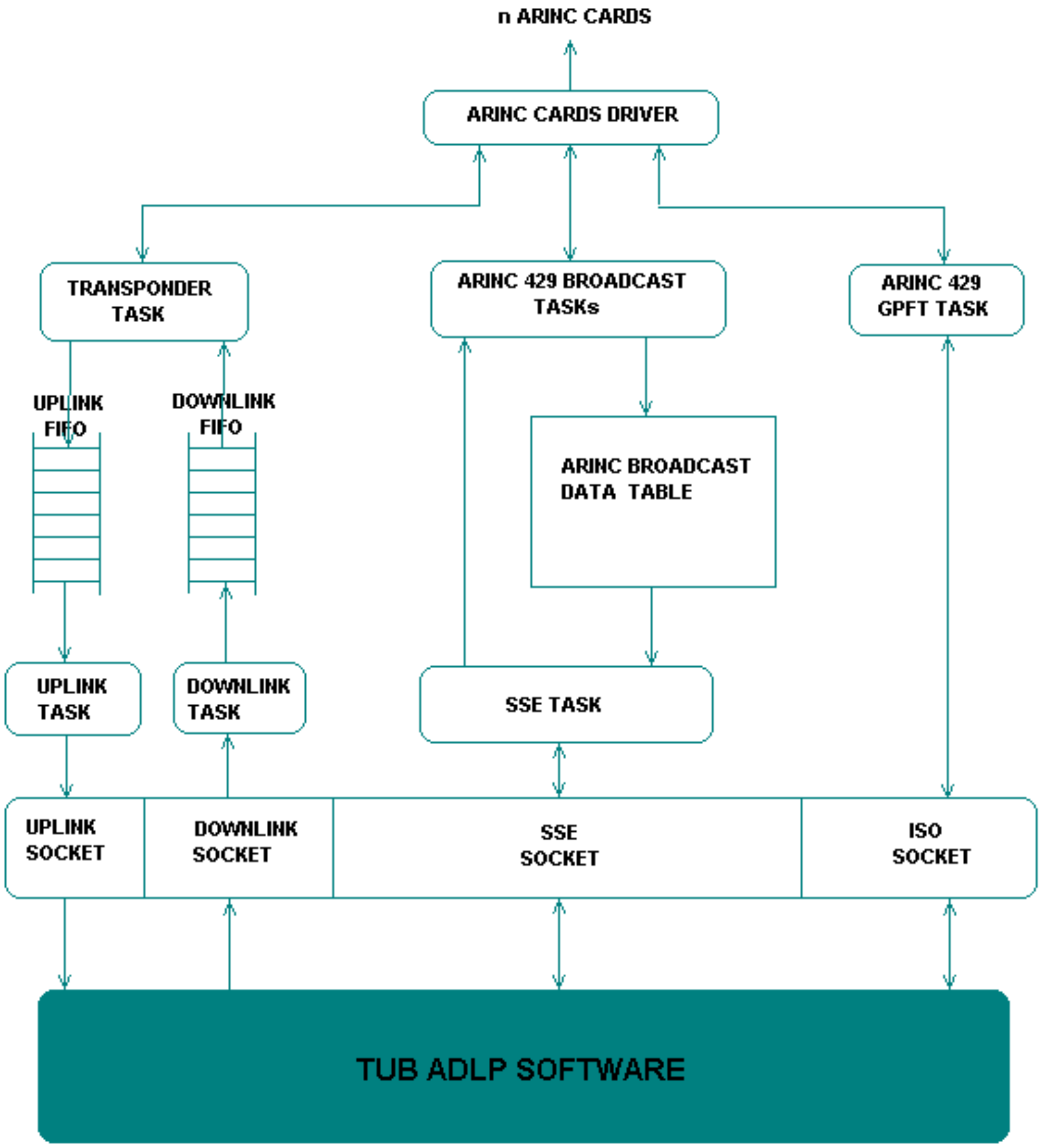


Figure 5 Software architecture

3 User's Manual

This chapter describes the ADLP user's manual

3.1. LOGIN

To connect to the system in user mode :

Login : **obione**

Password : **eureka**

To connect to the system in supervisor mode connect as user and type :

Calife

Password : **eureka**

To return to the user mode type :

exit

3.2 AUTOMATIC START-UP OF ADLP

For the ADLP to start automatically, modify the file **/etc/rc.local** in supervisor mode.

Add to the end of the file the line :

/usr/local/bin/tcsh /home/staff/obione/disk_p/adlp30/bin/bootadlp

To get control of the system type **CTRL-C**

3.3. MANUAL START

For the ADLP to start manually, modify the file **/etc/rc.local** in supervisor mode and restart the PC.

Make the following line a comment by adding # at start of line :

#/usr/local/bin/tcsh /home/staff/obione/disk_p/adlp30/bin/bootadlp

Then execute the following commands in supervisor mode.

Cd /dev

chmod 666 ttyv*

bin

cleanipc

To start the ADLP manually in user mode select directory bin ::

bin

bootadlp

3.4. CONFIGURATION FILES

3.4.1. ADLP.CFG

The ADLP configuration file is used to set up ADLP operation for different hardware or software configuration.

The ADLP configuration file defines :

- Transponder level
- Transponder up-link ELM capability
- Transponder down-link ELM capability
- ICAO 24 bit Mode S address
- Transponder interface : separate/common interface lines for ELM
- DTE interface assignment table
- MSP / GICB / BC interface assignment table
- Mode S Sub-network version number

The file name for the ADLP configuration file is **adlp.cfg** in the bin directory.

The file is a plain-ASCII file containing keywords, parameters and comments. Lines that start with a hash (#) are ignored (comments). A keyword has to start in column 1 of the line. Each keyword requires a parameter, separated by one or more spaces between the keyword and the parameter. All characters behind the parameter are ignored.

The following keywords are defined :

VERSION
The parameter denotes the current Sub-network number. According to the current SARPs this number has to be 1.

UP-LINK-ELM
The parameter denotes the transponder up-link ELM capability according to SARPs 2.9.1. Valid values are 0 to 6.

DOWN-LINK-ELM
The parameter denotes the transponder down-link ELM capability according to SARPs 2.9.1. Valid values are 0 to 6.

ICAO-ADDRESS
The parameter denotes the ICAO 24 bits address of the aircraft. The parameter has to be an 8 digit octal number where each digit represents 3 bits of the ICA0-address. For example $38019B is 16000633 in octal representation.

SPLIT-ELM
The parameter has to be YES or NO (only the first character of the parameter is interpreted). This keyword is used to switch on or off the split transponder interface option. With the parameter YES the ADLP deliver and receives ELM frames on a separate physical ARINC718 output line. With the parameter NO, ELM and SLM frames are delivered and received on a same physical ARINC718 interface.

DTES
The parameter defines, which DTEs are connected to the ADLP on which physical interface line. The parameter has to be a string with 16 characters length. Each character position in the string represents an aircraft DTE address where the leftmost character denotes DTE 0. The dash (-) denotes that a DTE address is not used. For each DTE that is connected to the ADLP the physical number (from 0 to 3) has to be at the position relating to the DTE address in the parameter string.

LEVEL5
The parameter has to be YES or NO (only the first character of the parameter is interpreted). The parameter YES indicates that a level-5 transponder is connected to the ADLP. The parameter NO indicates that a level-2 to level-4 transponder is connected to the ADLP.

UMSP0, UMSP1, UMSP2, UMSP3
The parameter of these 4 keywords determines the assignment between the 64 MSP channels and the physical specific services interfaces at up-link. Each parameter has to be a string with 16 characters length. Each character position in the string denotes a MSP channel number where the leftmost character of the UMSP0 parameter denotes M/CH 0 and the rightmost character of the UMSP3 parameter denotes M/CH 63 (UMSP0: M/CH 0..15; UMSP1: M/CH 16..31; UMSP2: M/CH 32..47; UMSP3: M/CH 48..63). The dash (-) denotes that a MSP channel is not supported. For each supported MSP channel put the physical interface number (0 or 1) at the position relating to the MSP channel number address in the parameter of UMSP0..3.

DMSP0, DMSP1, DMSP2, DMSP3
The parameter of these 4 keywords determines the assignment between the 64 MSP channels and the physical specific services interfaces at down-link. The assignment is identical to that described for the UMSP keywords.

UP-LINK-BC-PORT
The parameter of this keyword determines the physical ARINC 429 file transfer interface used for up-link broadcast messages. Valid values are 0 and 1.

3.4.2. FITAMS.CFG

The FITAMS.CFG configuration file is used to set up ARINC channels.

The file defines :

- speed of the channels
- ARINC 429 broadcast labels
- Socket port base
- SAL for the Williamsburg protocol

The file is in the bin directory.

The file is a plain-ASCII file containing keywords, parameters and comments. Lines that start with a hash (#) are ignored (comments). A keyword has to start in column 1 of the line. Each keyword requires a parameter, separated by one or more spaces between the keyword and the parameter. All characters behind the parameter are ignored.

The following keywords are defined :

CHANNEL1, CHANNEL2, CHANNEL3, CHANNEL4, CHANNEL5, CHANNEL6
The parameter of these 6 keywords determines the assignment between the 6 ARINC 429 broadcast channels and the list of the ARINC labels to be handled per channel. The parameters (20 max per channel) have to be a 2 digit octal number.
For example CHANNEL3 63 64 74 76 103 110 111 112 120 121
The CHANNEL3 keyword is used by the transponder interface card
The CHANNEL4 .is used by the TAR interface card.
The CHANNEL1..2 and CHANNEL5..6 are used by 2 additional ARINC card dedicated to the DAPs process.

SPEED1, SPEED2, SPEED3, SPEED4, SPEED5, SPEED6
The parameter of these 6 keywords determines the speed of each channel. Valid values are HIGH or LOW. The SPEED3 parameter has to be HIGH according to the transponder interface. SPEED1 and SPEED2 must have the same value HIGH or LOW. The same for SPEED5 and SPEED6.

BDS
The parameters of this keyword denote the BDS list computed by the ADLP Actually there are only 4 parameters implemented 05, 40, 50, 60. The parameters are hexadecimal values.

HOST
The parameter of this keyword is only used in Brétigny because the Williamsburg interface is not in the same PC as are the other ADLP processes. If all the processes are in the same PC, the syntax is HOST 127.0.0.1 otherwise the syntax is HOST *IP address of the ADLP PC.*

PORTBASE
The parameter denotes the TCP port base for all the sockets interfaces according to the /etc/services file.

SAL
The parameter denotes the TAR SAL number according to the TAR configuration file. The value is a decimal number.

TRACE
The parameter enables/disables the ARINC word traces. Valid values are YES or NO.

3.5. TASKS

When the ADLP starts you can see the following messages :

---adp000 < /dev/ttyv2 | tee ./trace/recadlpxxxxxx.xxxxx > /dev/ttyv2 ---

---down-link | tee ./trace/recdown-linkxxxxxx.xxxxx > /dev/ttyv3 ---

---up-link | tee ./trace/recup-linkxxxxxxx.xxxx > /dev/ttyv4 ---

---gicbsse > /dev/ttyv5 ---

---gpftsse | tee ./trace/recgpftadixxxxxx.xxxxx > /dev/ttyv6 ---

---xpdrirs | tee ./trace/recxpdrirsxxxxxx.xxxx > /dev/ttyv7 ---

xxxxxxx.xxxxx is the date and time when the ADLP started

To follow the behaviour of the six tasks type **ALT-Fx**

To see ADLP traces type **ALT-F3**

To see the transponder down-link task type **ALT-F4**

To see the transponder up-link task type **ALT-F5**

To see the task which computes BDS (gicbsse) type **ALT-F6**

To see ISO8208 and MSP task (gpftsse) type **ALT-F7**

To see the state of the transponder interface type **ALT-F8**

3.6. ADLP LOG FILES

To edit ADLP log files go to the trace directory by typing **bin and cd trace**

To choose the file name see chapter " Tasks ".

3.7. STOP ADLP

To stop ADLP type **q or Q**.

Before switching off the PC type **CTRL-ALT-SUPPR** then switch off the PC.

3.8. COPY TO AND FROM FLOPPY

You must be in supervisor mode and stop the ADLP software

Type **mtools** to have some help

To copy binary files to the floppy type :

mcopy nomfic a:

To copy text files to the floppy type :

mcopy -t nomfic a:

To copy binary files from the floppy type :

mcopy a: nomfic

To copy text files from the floppy type :

mcopy -t a: nomfic

3.9. Internet Web sites

To have more information concerning the UNIX FREE BSD have a look on the following site:

http://www.fr.FREEBSD.org/

FRENCH TRANSLATION

1. Introduction

Comme l'ADLP n'existait pas sur le marché, nous avons décidé d'en réaliser un en utilisant un PC industriel durci destiné à être utilisé en laboratoire ou dans des avions expérimentaux.

Le logiciel a été développé en langage C par TUB sous contrat d'Eurocontrol.

Eurocontrol Brétigny a réalisé toutes les interfaces ARINC logiciel et matériel avec les équipements comme le transpondeur Mode S, le routeur ATN, les bus DAP, et le récepteur GNSS / GPS.

Avoir un ADLP sur PC présente plusieurs avantages, notamment la maîtrise complète du matériel et du logiciel pour de la maintenance corrective et évolutive par Eurocontrol Brétigny.

Depuis sa réalisation, l'ADLP a suscité un grand intérêt auprès des Etats membres ; nous avons fourni l'équipement à la DERA, la DFS, le STNA pour des projets variés tels que :

- Mode S sub-network SARPs Validation,
- vols expérimentaux FITAMS,
- implémentation air/sol du sous-réseau ATN,
- expérimentation ADS-B,
- expérimentation DAPs.

Un processeur de liaison de données bord (ADLP) fournit les interfaces de communication nécessaires pour dialoguer avec :

- un routeur ATN à bord utilisant une interface au standard ISO 8208 et Williamsburg,
- un transpondeur Mode S (XPDR) utilisant le protocole ARINC 718,
- un GNSS / GPS ou un équipement de données avion utilisant le protocole ARINC 429 broadcast.

Le sous-réseau Mode S est composé d'un transpondeur Mode S (XPDR), d'un interrogateur Mode S (MODE S Radar) et d'un processeur de liaisons de données sol (GDLP) qui est l'homologue de l'ADLP au sol.

L’ADLP est constitué d’un PC Pentium ADVANTECH et de quatre cartes ARINC connectées sur le bus ISA du PC. Ces cartes ARINC sont conçues et fabriquées à Brétigny. Elles fournissent toutes les interfaces ARINC 429 requises.

L’ADLP s’exécute sous un environnement UNIX multitâches FREEBSD.

Les tâches suivantes constituent le logiciel de l’ADLP :

- les tâches TUB,
- la tâche d’interface transpondeur,
- la tâche d’interface broadcast ARINC 429,
- la tâche d’interface ARINC 429 Williamsburg,
- la tâche SSE (entité des services spécifiques du Mode S).

Nous utilisons des interfaces «sockets» pour communiquer entre les tâches et notamment avec celles de TUB :

- socket de communication descendante avec le transpondeur,
- socket de communication montante avec le transpondeur,
- sockets ISO 8208,
- sockets SSE

2. Le laboratoire de Bretigny

Les services rendus par le laboratoire mode S de Brétigny sont:

- la recherche et le développement,
- la fourniture d’équipements pour les états membres,
- la mise en œuvre et la simulation d’équipements avioniques embarqués.

Sur le toit du bâtiment de Brétigny nous avons :

- une antenne directionnelle dirigée vers le radar Mode S expérimental d’Orly pour être accessible par le sous-réseau Mode S français.
- une antenne sectorisée et une antenne omnidirectionnelle pour les évaluations de technologie ADS-B

L’ADLP est utilisé pour les applications :

- intégration du sous-réseau ATN avec Mode S,
- ADS-Broadcast et acquisition des squitters longs,
- passerelle ARINC 718 and Williamsburg pour des intégrations distantes

2.1. Intégration du sous-réseau ATN avec le Mode S

Le routeur ATN (TAR) est le premier module entièrement compatible avec la recommandation de l'OACI CNS/ATM-1. Le TAR est utilisé dans les expérimentations d'ADS l'Europe et c'est l'élément principal dans le déploiement de l'infrastructure d'ATN.

Consulter le site suivant pour plus d'information à ce sujet :

http://www.eurocontrol.fr/projects/atn/projects-nav-1.html

Pour préparer l'expérimentation FITAMS (Flight Trials with ATN and Multiple Sub-networks) nous avons effectué l'intégration complète de l'ATN pour les sous-réseaux satellite et Mode S, c'est dans ce cadre que nous avons utilisé l'ADLP sur PC.

Nous avons utilisé un transpondeur Mode S de TRT de niveau 4 supportant les messages descendants de 16 segments et les squitters long.

2.2. ADS-Broadcast et DAPs

ADS-Broadcast est un des futurs moyens possibles pour la surveillance, le transpondeur dans l'avion transmet spontanément les informations suivantes :

- position de l'avion : Latitude, longitude et altitude
- position de l'avion au sol : Latitude, Longitude, vitesse
- Vélocité : Vélocité Est/West et Nord/Sud, taux de monté et de virage
- Identifiant du vol et le type d'avion
- information d'intention

L' ADLP reçoit des mots ARINC d'un GPS et des systèmes de données avion. Avec ces informations, il met à jour des registres du transpondeur à intervalle fréquent. Le transpondeur se charge d'envoyer des squitters contenant ces informations ou de les délivrer au sol sur interrogations de surveillance enrichie du radar.

L'ADLP met à jour dans le transpondeur les registres GICB 05, 06, 40, 50 and 60. Le format de ces registres est données dans la note technique :

- http://www.eurocontrol.fr/public/reports/eecnotes/1998/20.htm

Pour plus d'informations consulter les notes techniques :

- http://www.eurocontrol.fr/public/reports/eecnotes//1995/not1-95.htm
- http://www.eurocontrol.fr/public/reports/eecnotes/1997/not17-97.htm
- http://www.eurocontrol.fr/public/reports/eecnotes/1998/11.htm

2.3. Passerelle pour les interfaces ARINC 718 et Williamsburg

Cette passerelle a été mise en œuvre par EEC pour utiliser à distance le banc de test transpondeur mode S de Brétigny et le routeur ATN bord en utilisant le réseau IP.

Le transpondeur est dans la couverture du sous-réseau Mode S français (radar mode S d'Orly / T-GDLP à Toulouse).

NLR l'utilise pour réaliser à Amsterdam l'intégration du produit ADLP certifié aéronautique.

Cette passerelle est un serveur IP offrant deux interfaces ARINC 429 :

- ARINC 429 pour communiquer avec le transpondeur Mode S en utilisant le protocole ARINC 718,
- ARINC429 pour communiquer avec le routeur bord ATN en utilisant les protocoles Williamsburg et ISO8208.

Le serveur tourne à Bretigny sur un PC sous FREE BSD, il crée deux sockets de communication TCP/IP :

- 718_SOCKET port 7041
- ISO_SOCKET port 7040

Le client tourne sur l'ADLP à NLR, il est connecté sur les port sockets et échange des données avec le serveur en utilisant les deux interfaces.

ABBREVIATIONS

AAC	Advanced ARINC Card
ADCE	Airborne DCE
ADI	Aircraft Data Interface
ADP	Aircraft Data Processing
ADLP	Airborne Data Link Processor
ADS	Automatic Dependent Surveillance
ADS-B	Automatic Dependent Surveillance (Broadcast data)
AICB	Air Initiated Comm-B
AICD	Air Initiated Comm-D
ADCB	Air Directed Comm-B
ADCD	Air Directed Comm-D
ARINC	Aeronautical Radio Inc.
ATN	Aeronautical Telecommunication Network
BDS	Comm-B DATA Selector
BCD	Binary Code Decimal
BC	Broadcast
BSD IPC	Berkeley Software Distribution Inter-process Communication
CDI	Coding Identification
CPR	Compact Position Reporting
DAP	Down-link Aircraft Parameters
DCE	Data Control Equipment
DTE	Data Terminal Equipment
DLPU	Data Link Processor Unit
ELM	Extended Length Message
EFMS	Experimental Flight Management System
FITAMS	Flight Trial for ATN and Multiple Sub-network
GDLP	Ground Data Link Processor
GFI	General Format Identifier
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GICB	Ground Initiated Comm-B Messages

GPFT	General Purpose File Transfer Protocol
ICAO	International Civil Aviation Organisation
II	Interrogator Identifier
ISO	International Organisation for Standardisation
LCI	Logical Channel Identifier
LSB	Least Significant Bit
MCDU	Multi-purpose Control and Display Unit
MODE S	Mode Select
MSB	Most Significant Bit
MSP	Mode S Specific Protocol
PC	Personal computer
RF	Radio frequency
SLM	Short Length Message
SICAS	SSR Improvements and Collision Avoidance Systems
SICASP	SICAS Panel
SSE	Mode S Specific Services Entity
SSI	Mode S Specific Services Interface
SSM	Status Sign Matrix
SSR	Secondary Surveillance Radar
SVC	Switched Virtual Circuits
TUB	Technische Universität Braunschweig
XPDR	Mode S Transponder

- “Advanced PC ARINC card Version 2” EEC note N° 17/94
- “Firmware of the EEC PC ARINC Card EEC” Note N° 27/97
- ICAO ANNEX 10 “Surveillance radar and collision avoidance systems”
- ICAO “Manual on Mode S Specific services” N°9688-AN/952
- “Interface between DTE and DCE Data networks” ITU-T Recom. X.25 10/96
- “Data communications X.25 packet layer protocol for DTE” ISO / IEC 8208
- ARINC specification DITS 429P1-15
- ARINC specification DITS 429P2-15
- ARINC specification DITS 429P3-15
- “Minimum Operational Performance Specification for Secondary Surveillance Radar Mode S Transponders“ ED-73 EUROCAE
- “MARK 3 Air Traffic Control Transponder ” ARINC Characteristic 718-4 15/12/89
- “MOPS ADLP” DO 203A RTCA
- FREE BSD 2.X Handbook, driver note and FAQ

For further information have a look to the EEC technical notes :

http://www.eurocontrol.fr/public/reports/eecnotes//1995/not1-95.htm

http://www.eurocontrol.fr/public/reports/eecnotes/1995/notes_95.htm

http://www.eurocontrol.fr/public/reports/eecnotes//1995/not5-95.htm

http://www.eurocontrol.fr/public/reports/eecnotes//1995/not16-95.htm

http://www.eurocontrol.fr/public/reports/eecnotes//1995/not25-95.htm

http://www.eurocontrol.fr/public/reports/eecnotes//1996/not3-96.htm

http://www.eurocontrol.fr/public/reports/eecnotes/1997/not17-97.htm

http://www.eurocontrol.fr/public/reports/eecnotes/1997/not27-97.htm

http://www.eurocontrol.fr/public/reports/eecnotes/1998/11.htm

http://www.eurocontrol.fr/public/reports/eecnotes/1998/20.htm

APPENDIX A - BRETIGNY LABORATORY

APPENDIX A BRETIGNY LABORATORY

A.1. Laboratory applications

The services provided by the Mode S laboratory at EEC Brétigny are for research and development and providing facilities for the member States.

The implementation and the simulation of Mode S airborne equipment are made available in the laboratory.

On the roof of the EEC building, there are:

- One directional antenna directed towards the Orly radar to be reachable from Mode S sub-network
- One sectorised and one omni-directional antenna for ADS-B evaluation

The following applications will be described below:

- ATN and Mode S sub-network integration
- ADS-Broadcast and long squitter acquisition
- Gateway for ARINC 718 and Willamsburg interfaces

A.2. ATN and Mode S sub-network integration

The Trials ATN Router (TAR) is the first fully ICAO CNS/ATM-1 Package compliant router and as such is particularly important in the validation of the CNS/ATM-1 Package SARPs. The TAR is currently used in the ADS Europe trials and follow-on projects and was a contribution to that project. The TAR is also a key element in the ATN Infrastructure deployment.

The Trials Transport Server (TTS) provides CNS/ATM-1 Package compliant Transport Service. It allows the integration of user supplied applications, e.g. TES applications. Transport API complies to industry standard XTI (X/OPEN Transport Interface).

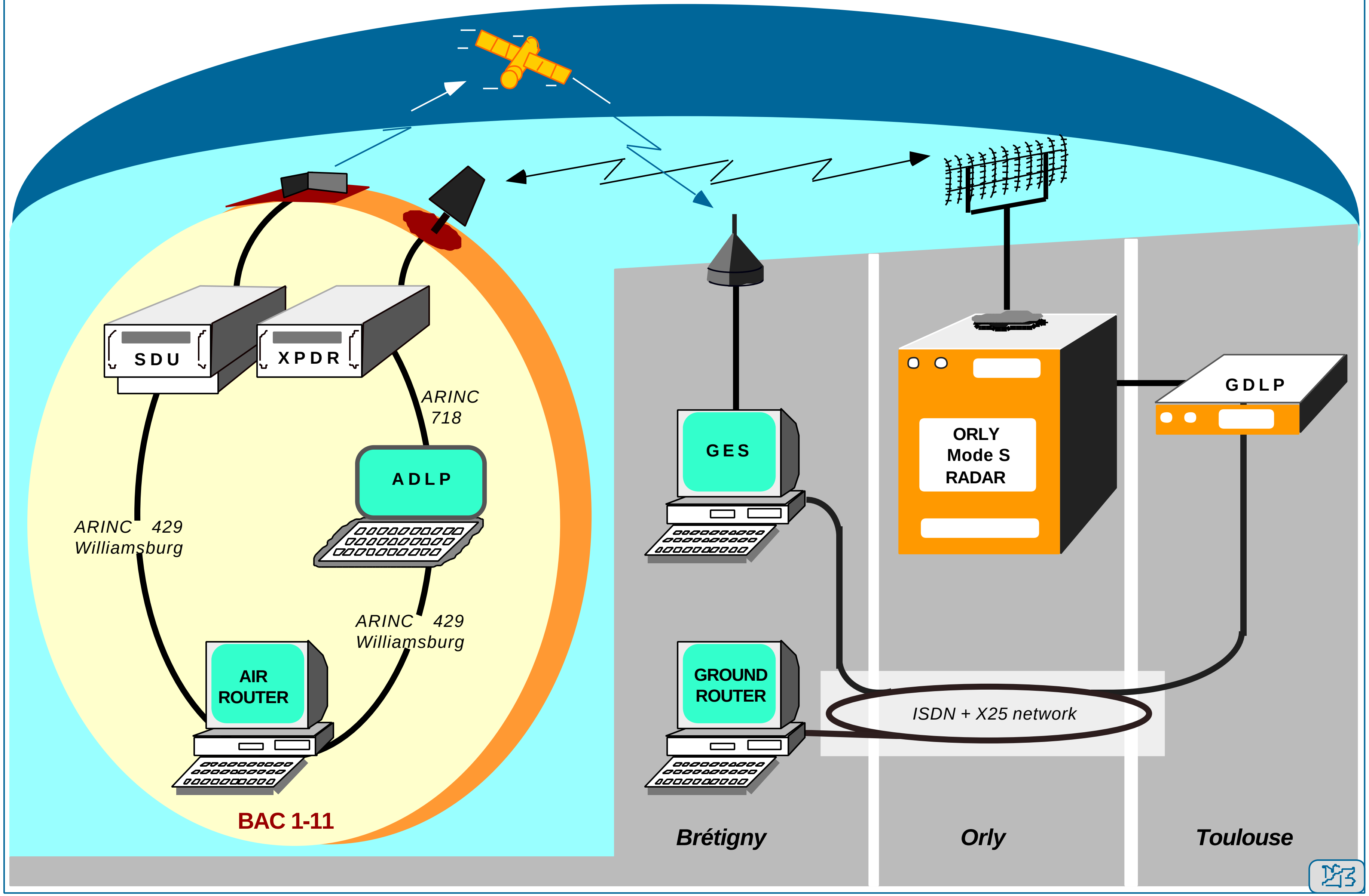
Together TAR and TTS provide a configurable platform for the deployment of ATN systems (Routers, End Systems or combined Routers/End Systems) in the ATIF. TAR-TTS main features:

The Trials End System (TES) experimental software is an implementation of ATN upper layers and application service entities (ASEs) conforming to ICAO SARPs.

See the following web site for more information :
http://www.eurocontrol.fr/projects/atn/projects-nav-1.html

FITAMS ATN - Mode S SDU Subnetworks

EEC - CoE COM



To prepare FITAMS (Flight Trials with ATN and Multiple Sub-networks) we made complete ATN and Multiple sub-networks integration tests at Brétigny, for this purpose we used the PC based ADLP.

FITAMS purpose is to demonstrate the first ever ICAO SARPs compliant end to end (7 layer) data link via multiple mobile sub-networks with live experimental aircraft. The aircraft is the BAC 1-11 from DERA, UK equipped with an airborne version of TAR-TTS-TES.

The Airborne Router is connected via ARINC 429 buses to SATCOM SDU and Mode S ADLP on the communication side, and to Experimental FMS and cockpit display on the application side.

Demonstration applications selected for FITAMS are:

ADS
A simplified ADS application is implemented which enables the down-linking of aircraft parameters in ADS reports at regular intervals. Aircraft parameters are gathered from FMS data on ARINC 429 dedicated buses.

CPDLC
A subset of CPDLC messages set is implemented. Airborne pilot interaction takes place via a touch screen interface.

To prepare FITAMS experimentation we used for the integration in the laboratory the following components:

At Brétigny :

- the transponder :

  The transponder used is a Level 4 TRT transponder provided by EUROCONTROL capable of supporting 16 segment down-link ELMs and extended squitter

- the PC based ADLP
- SDU
- Airborne ATN router
- Ground ATN router
- GES

At Orly :

- Mode S Radar

At Toulouse :

T-GDLP :Trials-Ground Data Link Processor.
The T-GDLP was procured by EUROCONTROL and runs on proprietary hardware. It communicates with the Orly Mode S radar using the Asterix protocol and with ground ATN router using ISDN connection and X25 network.

A.3. ADS-Broadcast and DAPs

ADS-Broadcast is one of the future possible means of surveillance, the transponder in the aircraft transmits spontaneously the following information:

- Airborne position : Latitude, longitude and altitude
- Or Ground position : Latitude, Longitude, ground track, speed
- Velocity : E/W & N/S velocity , vertical & turn rate
- Flight identical and aircraft type
- Intent information

In the Bac 1-11 aircraft, the ADLP receives ARINC words from GNSS equipment containing position information from the from GPS system. With this information the ADLP computes some gicb register contents and updates the transponder with it at frequent intervals.

In the laboratory, it is possible to simulate the GNSS equipment with an ARINC word generator.

Only Airborne / Ground position and flight ident. Squitter data are computed actually by the ADLP using for the position the CPR algorithm, the velocity squitter is not yet implemented.

On the ground we use a Mode S Data Link ground station with an omni-directionnal antenna to receive and analyse the squitters.

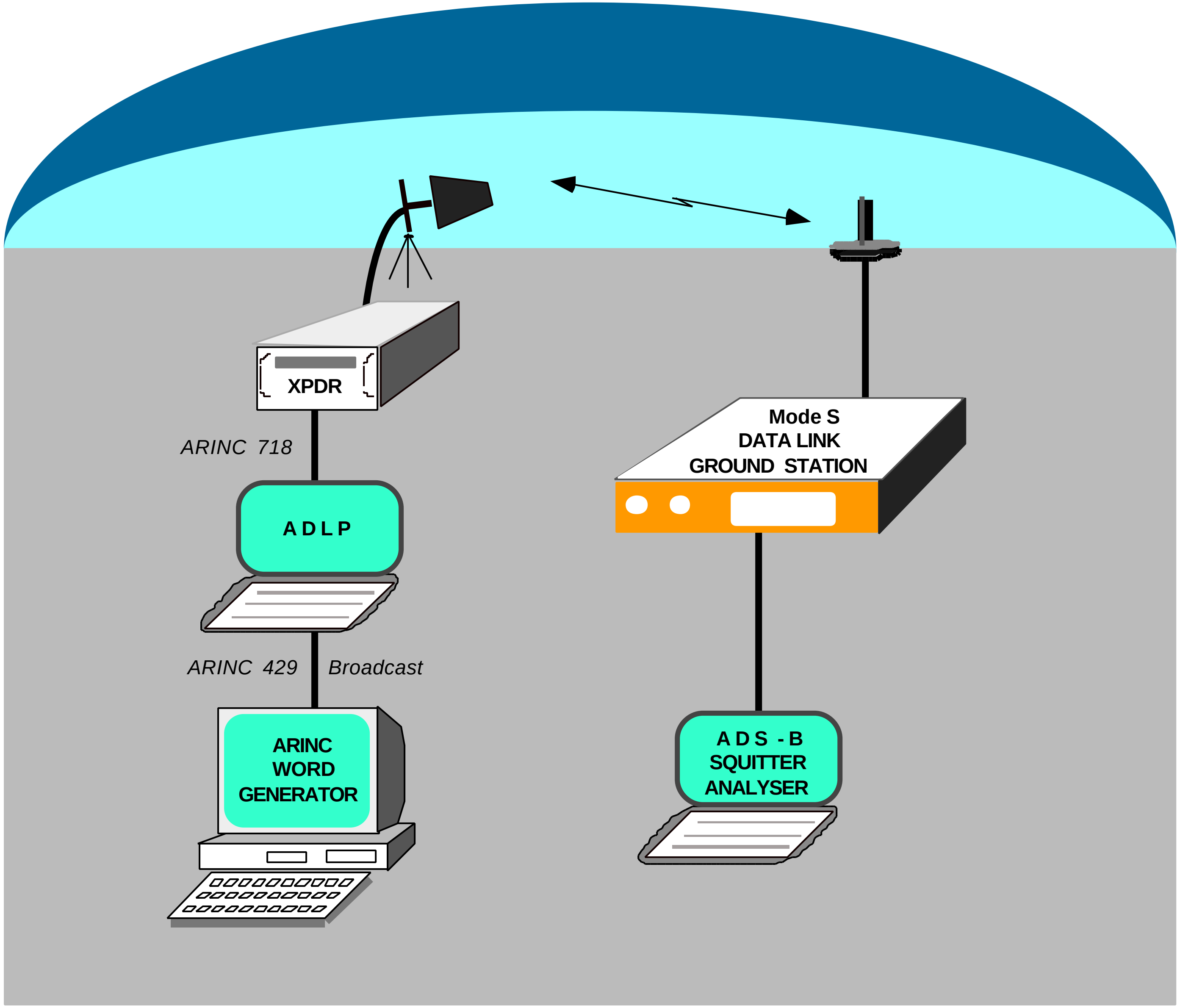
For more information see the following technical notes:

- http://www.eurocontrol.fr/public/reports/eecnotes//1995/not1-95.htm
- http://www.eurocontrol.fr/public/reports/eecnotes/1997/not17-97.htm
- http://www.eurocontrol.fr/public/reports/eecnotes/1998/11.htm

The ADLP computes the contents of GICB registers 4,0, 5,0 and 6,0. The format of these registers is shown in the following technical note :

- http://www.eurocontrol.fr/public/reports/eecnotes/1998/20.htm

A D S - Broadcast



A.4. Gateway for 718 and Willamsburg interfaces

A.4.1. Objective

This gateway was implemented by EEC to use remotely the Brétigny Mode S transponder test bench and ATN airborne router. The transponder is on the French Mode S ground sub-network coverage (Orly Mode S radar / T-GDLP at Toulouse / ATN Route at Brétigny).

This gateway is remotely used by NLR to perform tests with the NLR ADLP located in the Netherlands.

This Gateway offers two different ARINC 429 interfaces on the IP network :

- ARINC 429 to communicate with the Mode A/C/S transponder using ARINC718 protocol
- ARINC429 to communicate with the ATN airborne router using Williamsburg and ISO8208 protocols

With the Gateway you can connect a client application to a Transponder and airborne router data server based at Brétigny using two software socket TCP/IP communication interfaces.

The server process are tasks on the Pentium PC running the FREE BSD operating system which creates two sockets :

- 718_SOCKET port 7041
- ISO_SOCKET port 7040

The client process is running on the ADLP at NLR, it is connected to the sockets and exchanges data using these two interfaces.

A.4.2. Transponder interface format

The up-link and down-link data packet have the same format :

Length [8] In byte	ARINC 718 words [nx16] (max 72 words)

The ARINC718 data interface in both directions only contains the 24 databits of each 718-word. (The other 8 bits of each 718-word contains no "user"-data).

A.4.3. ISO8208 interface format

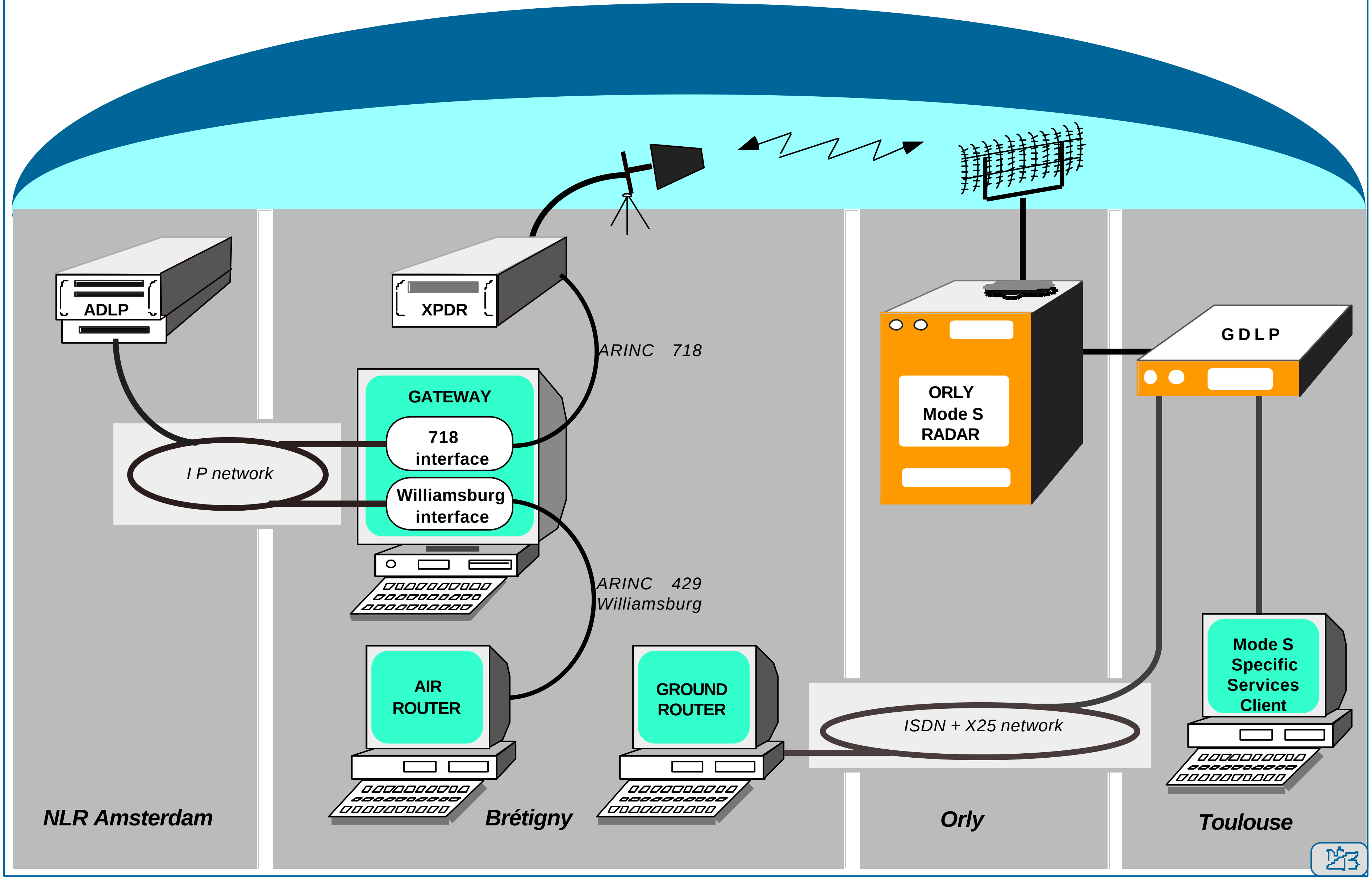
The ISO8208 data is preceded by 1 header-byte containing the GFI. The GFI is required to separate MSP packets (GFI = 3) from ISO8208 packets (GFI = 1).

Length [8] In byte	GFI = 1 or 3	n bytes [nx8]

GATEWAY for SVC & Mode S Specific Services Testing

EEC - CoE COM

EUROCONTROL



SITE	EQUIPMENT	DESCRIPTION
Brétigny	XPDR	Transponder. The transponder used is a Level 4 TRT transponder provided by EUROCONTROL capable of supporting 16 segment down-link ELMs and extended squitter
	Gateway	Server software based on PC running FREE BSD including 718 et Williamsburg interfaces using 2 advanced ARINC cards
	Airborne router	
	Ground router	
NLR	ADLP	Airborne Data-Link Processor Certified avionic equipment made by NLR including for the integration the client processes to communicate with gateway
Orly	Mode S Radar	
Toulouse	T-GDLP	Trails-Ground Data Link Processor. The T-GDLP was procured by EUROCONTROL and runs on proprietary hardware. It communicates with Radar using the Asterix protocol
	MSP Client	Mode S Specific services applications including GICB extraction and Dataflash client process

The gateway components are :

- Unix like FREE BSD operating system
- software for the transponder data server
- software for the TAR data server
- 2 advanced ARINC cards for ARINC429 interfaces
- 1 Ethernet card for IP network interface

APPENDIX B - TRT TRANSPONDER TEST RACK (1)

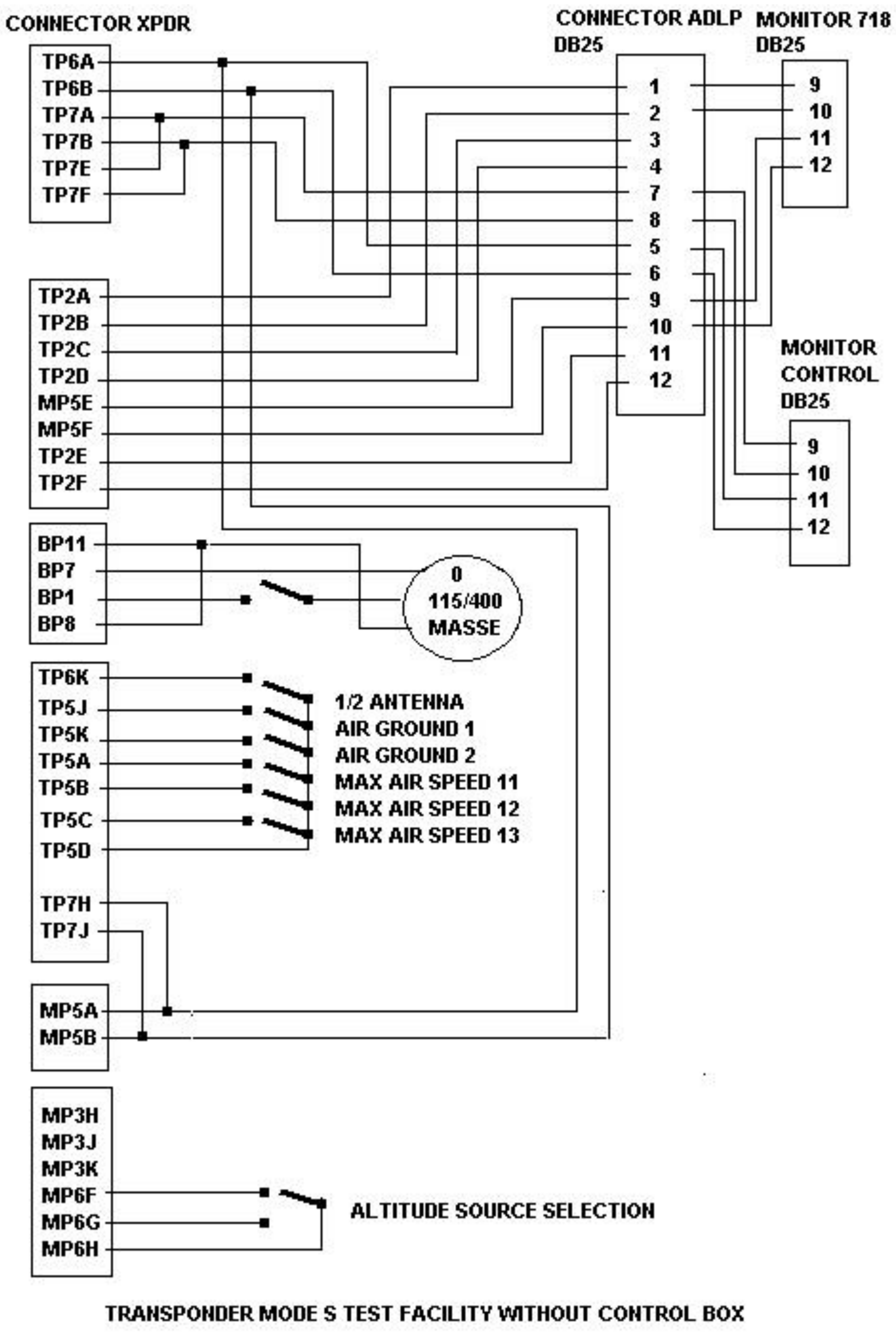


Figure 9 TRANSPONDER TEST RACK

APPENDIX C - TRT TRANSPONDER TEST RACK (2)

C.1. ARINC 600 connector

TP6K antenna diversity (1 = bottom antenna 0 = both antennas)

TP5J Air/Ground 1 (1 = ground 0 = air)

TP5K Air/Ground 2 (1 = ground 0 = air)

C.2. Front panel and test button

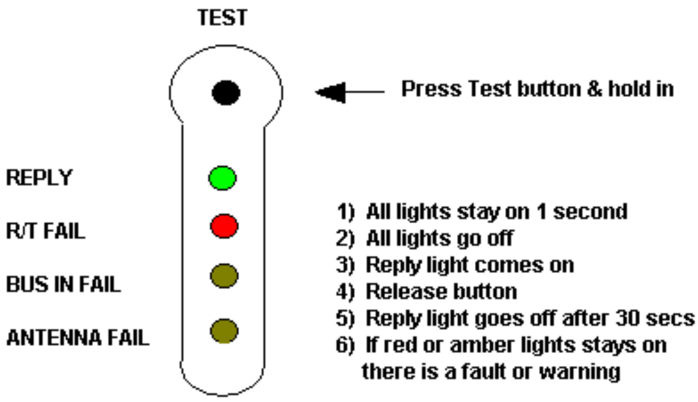


Figure 10 Transponder front panel

Reply "On" indicates Mode A/C reply
R/T Fail "On" indicates an internal transponder fault
Bus in fail "On" indicates that either:

- the altitude is not correct
- or not received by the transponder
- or the ARINC control word is not correct
- or not received by the transponder

This is normally a fault external to the transponder (Control unit fault, ADLP fault, …)

Antenna fail "On" indicates an antenna problem, this is normally external to the transponder.

Internal straps

There are 14 straps in the TRT transponder, only three straps are of interest for the ADLP applications. To change straps open left hand side of the transponder. Open = 0 soldered = 1

E8 (normally open)	open = old control unit	soldered = new control unit
E7 (soldered for STNA)	open = squitters enabled	soldered = squitters disabled
E6 (normally open)	open = external Mode S address & flight ident soldered = internal Mode S address ($F000XX) & flight id	

APPENDIX D - TAR / ADLP CABLE

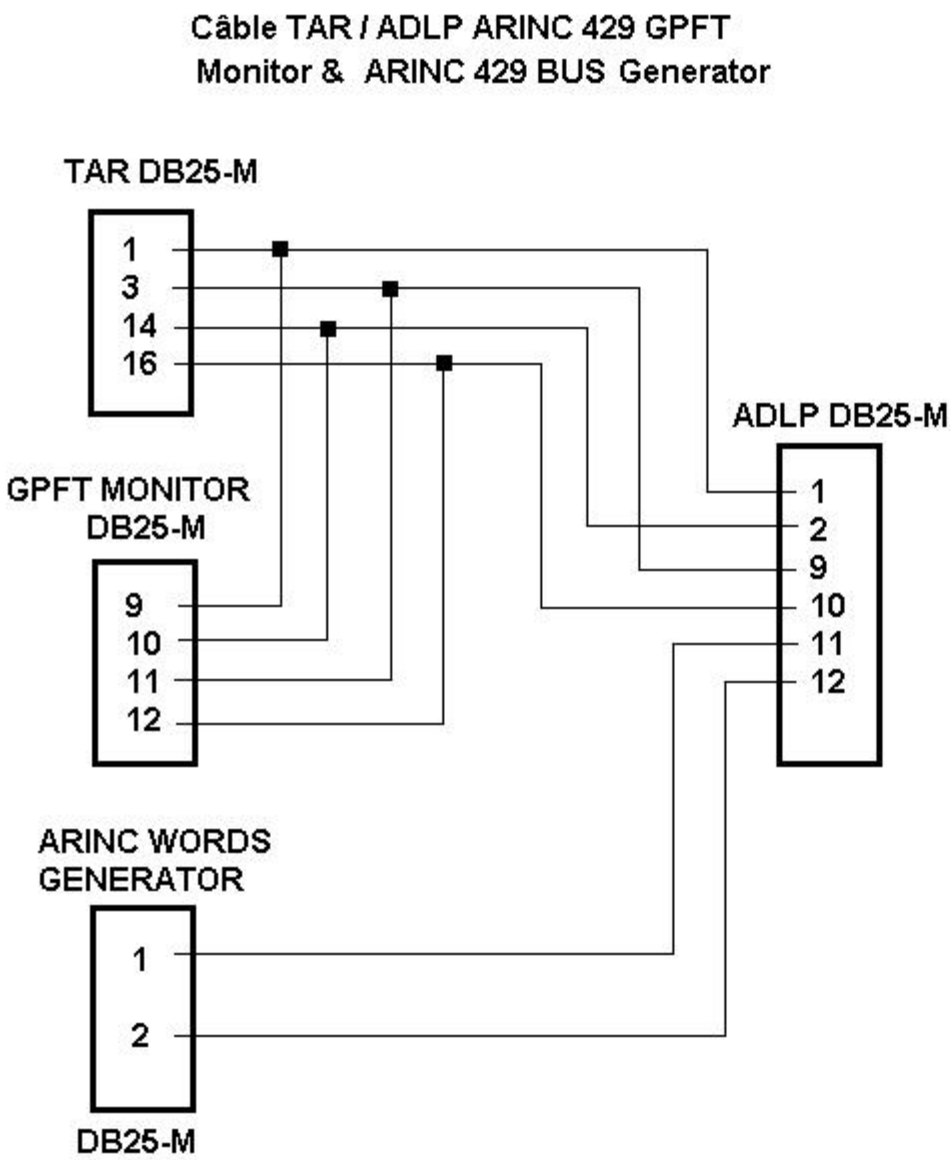


Figure 11 ADLP / TAR cable

APPENDIX E - GPFT INTERFACE SPECIFICATION

In the ADLP, a single ARINC-429 interface is used to support all communications between the Airborne End System and the ADLP. The following message types are present on the same interface:

- SVC traffic
- Connectivity Reports (Join or Leave events)
- Up-link SSE (MSPs and Broadcasts)
- Down-link SSE (MSPs, Broadcasts and GICB Register Updates)
- Delivery Indications for down-link SSE (ADLP to AES only)

All these messages are transferred using the Williamsburg General Purpose File Transfer (GPFT) Protocol defined in ARINC-429-12.

The GFI in the SOT message is 01_{HEX} for SVC traffic (1 above) and 03_{HEX} for all other traffic, including Connectivity Reports.

This GFI should **NOT** be transferred in the data words of the LDU.

E.1. SVC Related Formats

E.1.1 SVC Packets

The format for SVC packets is as specified in ISO8208 and is represented exactly without any conversions.

E.1.2 Connectivity Reports

The format for the Connectivity Report specified is:

Ground Address (AG) [8 bits]	Connectivity (0=leave, 1=join) [8]

As this message uses an ARINC-429 GFI of 03_{HEX}, it could be confused with an up-link SSE packet or SSE delivery status indication within the AES. However, the Connectivity Reports are only two bytes long, so the AES can take appropriate measures.

E.2 SSE related formats

Data formats are required for the following message types:

- Up-link SSE (MSPs and Broadcasts)
- Down-link SSE (MSPs, Broadcasts and GICB Register Updates)
- Delivery Indications for down-link SSE (ADLP to AES only)

E.2.1 Up-link SSE

Up-link SSE messages use the following format:

Format ID [8]	Packet Indicator [8]	Protocol Data [n*8]

E.2.1.1 Format ID

The format ID is set to 2 for up-link SSE data.

E.2.1.2 Packet Indicator

The Packet Indicator is set according to the following table:

Value (Hex)	Meaning
00 to 3F	MSP Ch No.
80	Broadcast
90	GICB Update

E.2.1.3 Protocol Data

For Up-link MSPs the Protocol Data is:

Message Type=0 [4]	II-code [4]	MSP Message [1x8..151x8]

Except for MSPs transmitted as a single up-link SLM, in which case the Protocol Data is:

Message Type=1 [4]	II-code [4]	SLM Header [32]	MSP Message [1x8..151x8]

For Up-link Broadcast the Protocol Data is:

SLM Header [32]	Broadcast ID [8]	Broadcast Message [6x8]

E.2.2 Down-link SSE Data

Down-link SSE messages use the following format:

Format ID [8]	Packet Indicator [8]	Sequence Number [8]	Protocol Data [nx8]

E.2.2.1 Format ID

The Format ID is set to 1 for down-link SSE data.

E.2.2.2 Packet Indicator

The Packet Indicator is set as specified in section 2.1.2 above.

E.2.2.3 Sequence Number

Sequence Numbers start at zero and are incremented modulo 256. The sequence restarts following a successful ALO/ALR exchange.

E.2.2.4 Protocol Data

For down-link MSPs the Protocol Data is:

Message Type=1 [4]	II-code [4]	MSP Message [1x8..159x8]

For Down-link Broadcasts the Protocol Data is:

Broadcast ID [8]	Broadcast Message [6x8]

For Down-link GICB updates the Protocol Data is:

Register Number [8]	GICB Message [7x8]

E.2.3 SSE Delivery Indications

The SSE Delivery Indications use the following format:

Format ID [8]	Packet Indicator [8]	Sequence Number [8]	Delivery Indicator [8]

E.2.3.1 Format ID

The Format ID is set to 0 for SSE Delivery Indications.

E.2.3.2 Packet Indicator

The Packet Indicator is set as specified in section 2.1.2 above.

E.2.3.3 Sequence Number

The Sequence Number is set to the same value as the Sequence Number used in the down-link message being acknowledged.

E.2.3.4 Delivery Indicator

The Delivery Indicator is set to 1 if the packet has been successfully forwarded, otherwise it is set to zero.

APPENDIX F - OPERATING SYSTEM FREEBSD

FREEBSD is an advanced BSD UNIX operating system for "PC-compatible" computers, developed and maintained by a large team of individuals.

FREEBSD offers advanced networking, performance, security and compatibility features today which are still missing in other operating systems, even some of the best commercial ones.

FREEBSD makes an ideal Internet or Intranet server. It provides robust network services, even under the heaviest of loads, and uses memory efficiently to maintain good response times for hundreds, or even thousands, of simultaneous user processes. Visit our gallery for examples of FREEBSD powered applications and services.

The quality of FREEBSD combined with today's low-cost, high-speed PC hardware makes FREEBSD a very economical alternative to commercial UNIX workstations.

It is well-suited for a great number of both desktop and server applications.

FREEBSD can be installed from a variety of media including CD-ROM, floppy disk, magnetic tape, an MS-DOS partition, or if you have a network connection, you can install it directly over anonymous FTP or NFS. All you need is pair of blank, 1.44MB floppies and these directions.

While you might expect an operating system with these features to sell for a high price, FREEBSD is available free of charge and comes with full source code. If you would like to try it out, more information is available.

FREEBSD provides you with many advanced features previously available only on much more expensive computers. These features include:

- **Pre-emptive multitasking** with dynamic priority adjustment to ensure smooth and fair sharing of the computer between applications and users.

- **Multi-user** access means that many people can use a FREEBSD system simultaneously for a variety of things. System peripherals such as printers and tape drives are also properly SHARED BETWEEN ALL users on the system.

- Complete **TCP/IP networking** including SLIP, PPP, NFS and NIS support. This means that your FREEBSD machine can inter-operate easily with other systems as well act as an enterprise server, providing vital functions such as NFS (remote file access) and e-mail services or putting your organisation on the Internet with WWW, ftp, routing and firewall (security) services.

- **Memory protection** ensures that applications (or users) cannot interfere with each other. One application crashing will not affect others in any way.

- FREEBSD is a **32-bit operating system** and was designed as such from the ground up.

- The industry standard **X Window System** (X11R6) provides a graphical user interface (GUI) for the cost of a common VGA card and monitor and comes with full sources.

- **Binary compatibility** with many programs built for SCO, BSDI, NetBSD, Linux and 386BSD.

- Hundreds of **ready-to-run** applications are available from the FREEBSD **ports** and **packages** collection.

- Thousands of additional and **easy-to-port** applications available on the Internet. FREEBSD is source code compatible with most popular commercial Unix systems and thus most applications require few changes to compile.

- Demand paged **virtual memory** and `merged VM/buffer cache' design efficiently satisfies applications with large appetites for memory while still maintaining interactive response to other users.

- **Shared libraries** (the Unix equivalent of MS-Windows DLLs) provide for efficient use of disk space and memory.

- A full complement of **C, C++ and Fortran** development tools. Many additional languages for advanced research and development are also available in the ports and packages collection.

- **Source code** for the entire system means you have the greatest degree of control over your environment.

- Extensive **on-line documentation**.

FREEBSD is based on the 4.4BSD-Lite release from Computer Systems Research Group (CSRG) at the University of California at Berkeley, and carries on the distinguished tradition of BSD systems development. In addition to the fine work provided by CSRG, the FREEBSD Project has put in many thousands of hours in fine tuning the system for maximum performance and reliability in real-life load situations. As many of the commercial giants struggle to field PC operating systems with such features, performance and reliability.

For more details see following web site and news group

- http://www.fr.freebsd.org/

- FREE BSD-questions@FREE BSD.ORG.

http://www.freebsd.org

FreeBSD

FreeBSD:The Power To Serve

APPENDIX G - ADVANCED ARINC CARD DRIVER

APPENDIX G ADVANCED ARINC CARD DRIVER

To use the EEC Advanced ARINC Card in a PC running FREE BSD UNIX, we had to develop a driver using the source code of an existing generic ISA driver.

The driver runs in supervisor mode and allows a user application to communicate with card, the user program uses the following functions to interface with card :

- open the driver
- close the driver
- read from the card
- write to the card
- I/O Control function to command the behaviour of the card :
    - Interrupt handler attachment
    - Interrupt handler detachment
    - End of Interrupt
    - Start the firmware in the card
    - Stop the firmware in the card

G.1. Definition and equivalences

Definition of the driver are :

PORT_USE	Port number use by the kernel
MAX_CARD	Number maximum of card (max 4)
ARINC_ID	Identify the card
ARINC_SIZE	Ram size in the card
ARINC_START	firmware start address in card memory
ARINC_MIN(a,b)	macro to find the minimum value
UNIT(dev)	macro to associate minor number and card
ResetStart(a)	macro to reset the card
Hold(a)	macro to hold the firmware in the card
ResetInt(a)	macro to make end of interrupt
CDEV_MAJOR	Major number of the driver

This driver enables the use of up to four cards in the PC.

The data structures of the drivers are indicated by the driver suffix number which enables to define each card. (see chapter device installation)

The driver descriptor indexed by the suffix number is :

dp_mem_addr	the ram card pointer
dp_size	memory ram size
io_regs[3]	I/O port to command the card
io_port_size	size of I/O port
irq_level	to save the IRQ level
*proc_to_sig	interrupt handler pointer
Status	card status

G.2. Functions and procedures

Probe function

This function is called by the kernel during the boot sequence to initialises the driver to detect the hardware and sets up all relevant data-structures.

This function is also use to make the conversion between the virtual to physical address ram in the card.

To detect the hardware of the card, the only verification is a ram check made by write and read pattern in a memory of the card.

This function returns “0” if the system has detected a card.

Attach function

This function is called by the kernel during the boot sequence for each device minor number, It sets the data structure of the driver according to the parameter of this device.

This function returns a value greater than zero if the device is correctly attached.

Open and close function

This function is called by the user program to open or close the device driver, to allocate or free allocated resources.

Write and read function

This function is called by the user program to write or read a packet to the Advanced ARINC card. Access to the card is made by 16 bit words. The length is given by the number of 16 bit words

IOCTL functions

This function is called by the user program to control the behaviour of the card, this function provides the following actions :

- attach an interrupt handler
- detach an interrupt handler

- ↳ start the card
- ↳ stop the card
- ↳ generate an end of interrupt

Interrupt handler

This handler receives an IRQ from the card and sends a signal to the signal handler associated user application.

APPENDIX H - ADLP INSTALLATION

APPENDIX H   ADLP INSTALLATION

We used a floppy disk and the network to install FREE BSD

H.1. Prepare

Suppress all the partitions and the logical units on the hard disk and create a DOS partition (100 M for example) using ***fdisk.***

To prepare the installation it is necessary to create a bootable system floppy disk.

Format a floppy disk using DOS and use the software ***rawrite.exe*** to copy ***boot.flp*** onto it.

This software is available on the Web site:

- http://www.fr.freebsd.org/

Follow "Getting FREE BSD" and the directories /pub/FREE BSD/2.2.5-RELEASE/floppies and /pub/FREE BSD/tools for 2.2.5 version

The command to make a floppy disk is :

- rawrite boot.flp a:

H.2. Procedure

Verbose boot

Boot with the floppy disk and when the boot prompt appears hit **-v** :

- boot : **-v**

Hardware detection

Suppress all the not present peripherals in the PC and after hit A or Q to save the configuration detected.

Kernel configuration

Choose the following option :

- Kernel configuration in visual mode

Alt-F1 and Alt-F2 keys commute between boot traces and configuration.

To change a value in a configuration panel use keys up, down, enter or space.

Keymap option
Chose the keyboard type in the list

Options
Change only the parameter debug = YES using down and space keys

Partition
Create a partition using [C] key and values proposed by the software without changing, we use the maximum size available on the hard drive for this partition. Set bootable this slice using [S] key, then quit with [Q] key and choose the **Bootmgr** option with the space bar.

Labels
Define the label and the size of each partition in the slice :

- 30 M for **/** option mount = Y and option File system
- 64 M for the swap (2 * the / label) and option swap
- 500 M for **/usr** with mount = Y and option File system
- 100 M for **/var** with mount = Y and option File system
- 100 M for **/tmp** with mount = Y and option File system
- the rest for **/home** with mount = Y and option File system

To create label use [C] key or [T] and [M] keys to modify an existing label.

Distribution
Select **custom** with space bar and after select :

- bin, dict, doc, info, man,
- space bar on **src** field and after select **sys**

Select **basic** with enter key and after select :

- bin, cfg, doc, html, lib, lkit, man, prog, set

Select **Xserver** with enter key and after select :

- SVGA, VGA16, S3

Select **Fonts** with enter key and after select :

- fnts, fscl (speedo)

Media
Choose the media for the installation

- FTP site France 1 and ethernet

Fill the following fields :

- **Host** : adlp.uneec.eurocontrol.fr
- **Domain** : uneec.eurocontrol.fr
- **Gateway** : 147.196.6.100
- **Name server** : 147.196.69.9
- **IP address** : 147.196.6.155
- **Network mask** : 255.255.255.0

adlp is machine name, characteristics are EEC network specific, ask to your network administrator for your characteristics.

Commit
Before running this command check the successful DNS access with Alt-F1 key
This option install FREE BSD.

H.3. Configure the system

Users
To configure the users use the software **vipw** , each user has to be in the WHEEL group.

device

To create terminal devices run the following commands in the **/dev** directory :

- mknod ttyv0 c 12 0
- mknod ttyv1 c 12 1
- mknod ttyv2 c 12 2
- mknod ttyv3 c 12 3
- mknod ttyv4 c 12 4
- mknod ttyv5 c 12 5
- mknod ttyv6 c 12 6
- mknod ttyv7 c 12 7
- mknod ttyv8 c 12 8
- mknod ttyv9 c 12 9
- mknod ttyv10 c 12 10
- mknod ttyv11 c 12 11

To create devices for three advanced ARINC card use the following commands:

- mknod ac0 c 81 0
- mknod ac1 c 81 1
- mknod ac2 c 81 2

and enable user access :

- chmod 666 ac0
- chmod 666 ac1
- chmod 666 ac2

shells

Edit the **/etc/shells** file and add the following lines:

- /bin/sh
- /bin/csh
- /usr/local/bin/zsh
- /usr/local/bin/tcsh

passwd

Edit the **/etc/passwd** file and add the following line :

obione:*:1001:20:Philippe Brun:/home/staff/obione:/usr/local/bin/tcsh

ttys

Edit the **/etc/ttys** file and insert the following lines :

Ttyv0	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv1	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv2	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv3	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv4	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv5	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv6	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv7	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv8	"/usr/libexec/getty Pc"	Cons25	on secure
Ttyv9	"/usr/libexec/getty Pc"	Cons25	on secure

services

Edit the **/etc/services** file and insert the following lines according to the **netdef.h** file gave by TUB :

ADLP_SERVICE	7020/tcp
ADLP_ISO8208_1	7021/tcp
ADLP_ISO8208_2	7022/tcp
ADLP_ISO8208_3	7023/tcp
ADLP_ISO8208_4	7024/tcp
ADLP_XPDR_UP	7025/tcp
ADLP_XPDR_DN	7026/tcp
ADLP_XPDR_UP2	7027/tcp
ADLP_XPDR_DN2	7028/tcp
ADLP_SSE_1	7029/tcp
ADLP_SSE_2	7030/tcp
ADLP_SSE_3	7031/tcp

calife

With this software you can log in as root access and keep your user configuration and environment . To use this software add int the **/etc** directory the following files :

/etc/calife.auth : user list

```
roberto:
obione:
```

/etc/calife.auth-dist :

```
fcb
roberto:/bin/tcsh
pb::guest,blaireau
```

chsh

The ***chsh*** command is used to change the shell, run this command in user mode and after change the following line :

↳ shell : /usr/local/bin/tcsh

support.s

The **/usr/src/sys/i386/i386/support.s** file must be change to have word access to the advanced ARINC card, edit the file and and the following function::

```
ENTRY(bcopyw)
bcopyw:
        pushl   %esi
        pushl   %edi
        movl    12(%esp),%esi
        movl    16(%esp),%edi
        movl    20(%esp),%ecx
        movl    %edi,%eax
        subl    %esi,%eax
        cmpl    %ecx,%eax
        jb      1f
        cld
        rep
        movsw
        popl    %edi
        popl    %esi
        ret
        ALIGN_TEXT

1:      addl    %ecx,%edi
        addl    %ecx,%esi
        decl    %edi
        decl    %esi
        std
        rep
```

```
movsw
popl    %edi
popl    %esi
cld
ret
```

H.4. Kernel and ARINC card configuration

modification

The **/usr/src/sys/i386/conf/nom_machine** file must contain the advanced ARINC card description and the ADLP necessary resources:

```
device ac0 at is a? port 0x300 irq 5 iomem 0xF00000 iosiz 65536 vector accintr
device ac1 at isa? port 0x380 irq 3 iomem 0xF20000 iosiz 65536 vector accintr
device ac2 at isa? port 0x340 irq 7 iomem 0xF40000 iosiz 65536 vector accintr
options         SYSVSHM
options         SYSVSEM
options         "SHMSEG=64"
```

The **/usr/src/sys/i386/isa** directory must contains the following files :

- ARINC1.c
- ARINC.h

Makefile, .depend et files.i386 files
In the **/usr/sr/sys/compile/mon_machine** file add the ARINC1 description (same syntax as atapi for example)

Kernel generation
Run the following command :

```
config    nom_machine
```

In **../../compile/nom_machine** directory run the following commands :

make depend
make
make install (if make finish successfully)
after reboot.

APPENDIX I - SARPs VALIDATION PROJECT

I.1. Introduction

This study was made by the Mode S Data Link team of the Air Traffic Control System Group at the Defence Evaluation and Research Agency (DERA, Malvern.

I.2. Test bench

The Data Link Test Bed (DLTB) was developed by DERA, it incorporates all Mode S airborne and ground functional elements. Communication between the simulated aircraft installation and the ground elements employs a true Mode S link right down to the RF signals.

The following diagram illustrates the DLTB configuration :

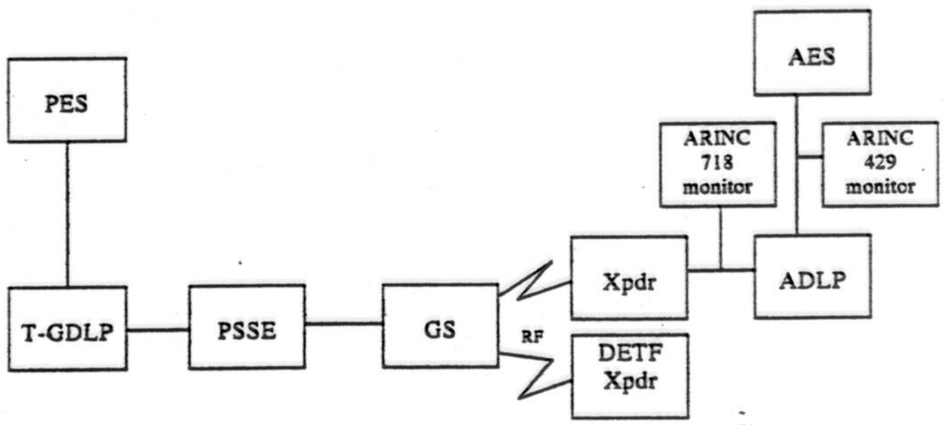


Figure 12 DERA DataLink Test Bench

The DLTB components are :

PES Private End System. The PES is a standalone PC running the Solaris operating system which communicates with the T-GDLP using an X25 card and driver software.

AES Airborne End System. The AES is a standalone PC running the Solaris operating system which communicates with the ADLP using an ARINC 429 card and driver software. The ISO8208 DTE functionality is provided using pre-written scripts.

T-GDLP Trails-Ground Data Link Processor. The T-GDLP was procured by EUROCONTROL and runs on proprietary hardware. It communicates with PSSE/GS using CAA Level 4 protocol

ADLP Airborne Data Link. The ADLP was procured by Brétigny and runs on a standalone PC running the Free BSD Unix operating system

PSSE	Private Specific Services Entity. The PSSE is a standalone PC running Windows 3.1. The PSSE provides a conversion function between the versions of the CAA Level 4 protocol used by the T-GDLP and ground station. The PSSE also provides an automatic acknowledgement for all up-link request
GS	Ground Station. The Malvern Ground Station is a Type 1 interrogator. This type of interrogator does not support cancellation (or acknowledgement) of up-link requests (SLM, ELM or broadcast)
XPDR	Transponder. The transponder used is a Level 4 TRT transponder provided by EUROCONTROL capable of supporting 16 segments ELMs
MONITOR	The ARINC 718 monitor runs on a standalone DOS PC fitted with a Brétigny ARINC 429 interface card

Over The past few years, EUROCONTROL and the UK National Air Traffic Services Ltd (NATS) have sponsored a number of activities at DERA Malvern aimed towards eventual validation of the Mode S Sub-network SARPs. The purpose of this additional EUROCONTROL project was to achieve substantial completion of this validation exercise through further complementary tests including investigation of the changes proposed at SICASP/6.

Validation of the Mode S Sub-network SARPs is required to be thorough and to demonstrate, with a high degree of confidence, that a system which satisfies the SARPs requirements can meet the requirements for the Mode S data-link.

The tests had two main objectives. Firstly, they validated the SARPs through investigations of the end-to-end operation of the Mode S Sub-network, secondly, they tested prototype Sub-network components, particularly the ADLP and T-GDLP procured by EUROCONTROL, for compliance with the SARPs including SICASP/6 amendments.

I.3. SSE scenario

The SSE scenario consists of sending ten up-link and ten down-link 10 byte MSPs followed by ten up-link and ten down-link 100 byte MSPs. The scenario is conducted twice, the second time with a background extraction of three GICBs per antenna revolution.

The scenario analysis performs two functions :

- Establishes that all messages were correctly transferred to the appropriate end system;
- Provides rudimentary timing analysis for up-link and down-link SLM and ELM transfer delay for the MSP service

I.4. SVC scenario

The SVC scenario consists of establishing an SVC and transferring ten up-link and ten down-link 10 byte data packets followed by ten up-link and ten down-link 100 byte data packets.

The scenario is conducted twice, the second time with a background extraction of three GICBs per antenna revolution.

The scenario analysis takes two forms :

- verification that all messages were correctly delivered
- rudimentary timing analysis for up-link and down-link SLM and ELM transfer delay for the SVC service.

APPENDIX J - DFS MODE S SUB-NETWORK

APPENDIX J   DFS MODE S SUB-NETWORK

J.1. Introduction

Several air-ground Data Links are currently becoming available for ATC and other purposes. These Data Links have different properties which make them differently suited for expected applications. In order to identify the key properties of the individual Data Links a comparative investigation of the Data Link media is currently prepared by Deutsche Flugsicherung GmbH (DFS).

J.2. Investigated Data Link Environment

The investigated overall Data Link environment is shown in figure 1

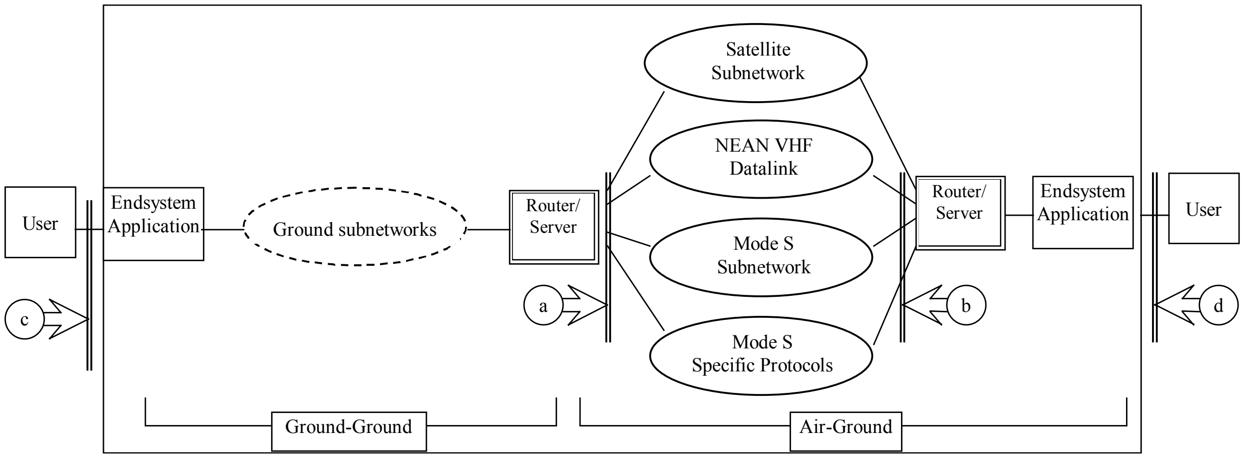


Figure 13 DFS Overall Data Link Environment

Figure 1 shows the overall Data Link environment in which the investigations are intended to take place. It should be noted that ground Sub-networks are excluded since their properties are in general known. The properties are analysed at two different communication levels which are, (1) the Data Link/Network level as between interfaces "a" and "b" and (2) the application level as indicated by interfaces "c" and "d".

J.3. Investigated Data Links

DFS is currently working on a detailed planning document which will provide the basis to investigate the following Data Links:

1. AMSS Data 3 service
2. Mode S Sub-network
3. VHF based North European ADS-B Network (NEAN) (communication facilities only)
4. Mode S Specific Protocol MSP communication facilities
5. ATN CPDLC Application via AMSS and Mode S

It is clearly visible that the first four entries in the list represent general purpose Data Links while the last entry in the list includes the communication stacks up to the application layer. This is intended to investigate the potential difference between the performance of the Data Links themselves and the communication at application level. Although the ATN represents one future solution, it might be that Data Links are used for dedicated purposes outside the framework of ATN. Application level behaviour of new applications can furthermore be deduced from the properties of the Data Links.

J.4. Investigated Parameters

The investigations of the Data Links address the following parameters:

1. Availability

   This parameter indicates at which probability the services of the Data Link are available.

2. Call Establishment Latency

   This parameter indicates how long it takes until a data communication service becomes available after a user wishes to send data to a destination. It hence has an impact on how call establishment is used; whether it can be performed just prior to the communication request or needs to be performed longer before (e.g. when an aircraft reaches a particular area).

3. Data Transmission Delay

   This parameter indicates how long it takes to transmit data from one location to another.

4. Data Integrity

   This parameter indicates how likely it is that data is received uncorrupted.

5. Available User Data rates

   This parameter indicates which data rate can be expected to be available for the transfer of data to/from one user.

From these parameters a test approach was derived which consists of the performance of three different experiments.

J.5. Approach

To perform the investigations it is planned to carry out the following three experiments for each of the Data Links and in both directions:

1. Call Establishment
2. Bit rate Determination
3. Data Exchange

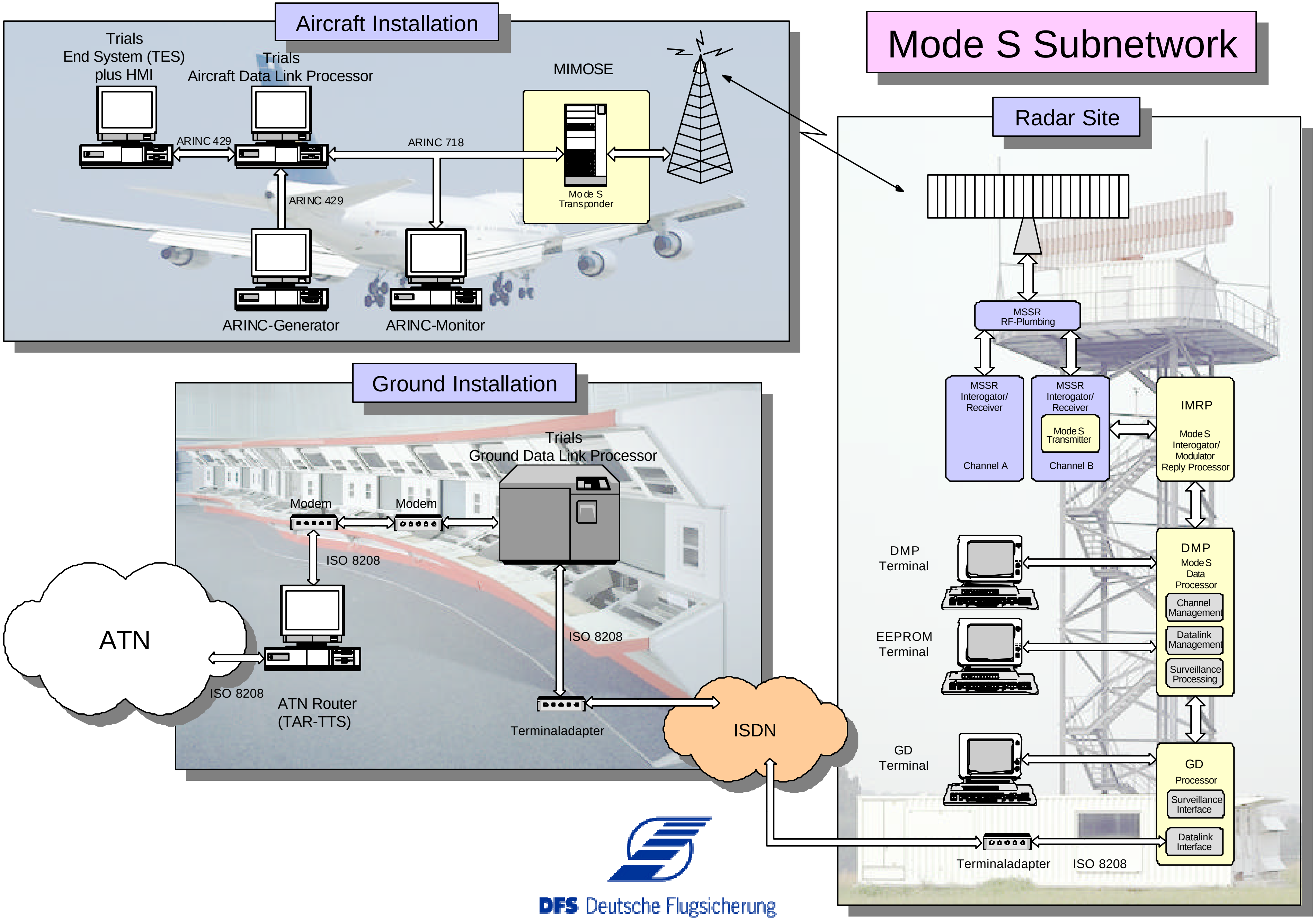
In a first step, basic experiments on application level will be performed by using the CPDLC HMI, which was provided in the scope of the flight trials over multiple ATN Sub-networks (FITAMS).

The ground installation components are :

ATN router	The Trials ATN Router (TAR) is the first fully ICAO CNS/ATM-1 Package compliant router and as such is particularly important in the validation of the CNS/ATM-1 Package SARPs.
T-GDLP	Trails-Ground Data Link Processor. The T-GDLP was procured by EUROCONTROL and runs on proprietary hardware. It communicates with ATN router and with the Mode S radar using ISDN link

The radar site installation components are :

GD	It is an ASTERIX gateway converter between T-GDLP and DMP.
DMP	This equipment is a software ground segment of the Mode S radar
IMRP	This equipment constitutes the extractor of the Mode S radar
Secondary antenna	The antenna of the Mode S radar is a standard secondary surveillance radar antenna



Aircraft Installation
Trials
End System (TES)
plus HMI
Trials
Aircraft Data Link Processor
MIMOSE
ARINC 429
ARINC 718
Mode S
Transponder
ARINC 429
ARINC-Generator
ARINC-Monitor
Mode S Subnetwork
Radar Site
MSSR
RF-Plumbing
MSSR
Interogator/
Receiver
Channel A
MSSR
Interogator/
Receiver
Mode S
Transmitter
Channel B
IMRP
Mode S
Interogator/
Modulator
Reply Processor
Ground Installation
Trials
Ground Data Link Processor
Modem
Modem
ISO 8208
ATN
ISO 8208
ATN Router
(TAR-TTS)
ISO 8208
Terminaladapter
ISDN
DMP
Terminal
DMP
Mode S
Data
Processor
Channel
Management
EEPROM
Terminal
Datalink
Management
Surveillance
Processing
GD
Terminal
GD
Processor
Surveillance
Interface
Datalink
Interface
Terminaladapter
ISO 8208
DFS Deutsche Flugsicherung

The aircraft installation components are:

ADLP	Airborne Data Link. The ADLP was procured by Brétigny and runs on a standalone PC running the Free BSD Unix operating system
XPDR	Transponder. The transponder used is a Level 4 TRT transponder provided by EUROCONTROL capable of supporting 16 segments ELMs
718 MONITOR	The ARINC 718 monitor runs on a standalone DOS PC fitted with a rétigny ARINC 429 interface card
GPFT MONITOR	The ARINC Williamsburg monitor runs on a standalone DOS PC fitted with a Brétigny ARINC 429 interface card
MIMOSE	This transponder test facility is a DFS product which provides a full configuration of the discreet available to configure the transponder behaviour.