EUROPEAN ORGANISATION
FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL EXPERIMENTAL CENTRE

TEST DATA LINK PROCESSOR

November 1993
EEC Task No. AT 58
EEC Note No. 19/93

Approved for publication by
the Head of Division B2

Issued : NOVEMBER 1993

The information contained in this document is the property of the EUROCONTROL Agency and no part should be reproduced in any form without the Agency's permission. The views expressed herein do not necessarily reflect the official views or policy of the Agency.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Security Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEC Note No. 19/93</td>
<td>Unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Originator Code</th>
<th>Originator (Corporate Author) Name/Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEC Division B2</td>
<td>EUROCONTROL Experimental Centre</td>
</tr>
<tr>
<td>Mr. H.P. ENGLMEIER</td>
<td>B. P. 15</td>
</tr>
<tr>
<td>Mr. J. M. MATEUS-MARTINS</td>
<td>F - 91222 BRETIGNY SUR ORGE Cedex</td>
</tr>
<tr>
<td></td>
<td>Telephone (1) 69 88 75 00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sponsor Code</th>
<th>Sponsor (Contract Authority) Name/Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering Directorate</td>
<td>EUROCONTROL Agency</td>
</tr>
<tr>
<td>Division D.ED3</td>
<td>72, rue de la Loi</td>
</tr>
<tr>
<td></td>
<td>B - 1040 BRUSSELS</td>
</tr>
<tr>
<td></td>
<td>Telephone (2) 729 35 11</td>
</tr>
</tbody>
</table>

| Title                          | TEST DATA LINK PROCESSOR                                  |

<table>
<thead>
<tr>
<th>Author</th>
<th>Date</th>
<th>Pages</th>
<th>Figs.</th>
<th>Refs</th>
<th>Annexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.P. ENGLMEIER</td>
<td>11/93</td>
<td>39</td>
<td>17</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>J.M. MATEUS-MARTINS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AT 58</td>
<td>06.92 to 03.93</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Statement</th>
<th>(a) Controlled by: Head of Division B2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(b) Special limitations: None</td>
</tr>
<tr>
<td></td>
<td>(c) Sent to NTIS: None</td>
</tr>
</tbody>
</table>

| Descriptors (keywords)        | MODE-S, ADLP, DLPU, XPDR, TDLP                          |

| Abstract                      | This note describes the MODE-S Test Data Link Processor (TDLP), which was developed by the EUROCONTROL Experimental Centre. The TDLP is able to emulate a MODE-S Data Link Processor (DLPU) for test and experiments where a ruggedized and certified system is not required. Being PC based, the TDLP is cheaper and easier to operate and to modify than a DLPU. The TDLP can be upgraded to emulate an Aircraft Data Link Processor (ADLP). |
TEST DATA LINK PROCESSOR

by

H.P. ENGLMEIER
and
J. M. MATEUS MARTINS

SUMMARY

This note describes the MODE-S Test Data Link Processor (TDLP), which was developed by the EUROCONTROL Experimental Centre. The TDLP is able to emulate a MODE-S Data Link Processor (DLPU) for test and experiments where a ruggedized and certified system is not required. Being PC based, the TDLP is cheaper and easier to operate and to modify than a DLPU. The TDLP can be upgraded to emulate an Aircraft Data Link Processor (ADLP).
I. INTRODUCTION

1. ACRONYM AND ABBREVIATION DEFINITIONS

2. PURPOSE AND SCOPE

II. GENERAL OVERVIEW

III. THE TEST DATA LINK PROCESSOR

1. GENERAL ASPECTS

2. HARDWARE ENVIRONMENT

3. SOFTWARE ENVIRONMENT

4. ARCHITECTURAL DESIGN

4.1. TXID.ASM General Overview

4.2. TDLP.PAS General Overview

4.3. TDLP Real Time Simulation Facility

4.3.1. Uplink Process

4.3.1.1. Validate Word / Record

4.3.1.2. Messages Types

Test Data Link Processor
4.3.1.3. Messages Validation

4.3.1.4. Screen Management

4.3.2. Downlink Process

4.3.2.1. Keyboard Management

4.3.2.2. Messages Library

4.3.2.3. Messages Generation

4.3.2.4. Ground Initiated COMM-B Library

5. TDLP USER MANUAL

5.1. Introduction

5.2. Hardware Requirements

5.3. Software Requirements

5.4. Starting TDLP

5.5. Main Menu

5.6. TDLP Real Time Simulation

5.6.1. Avionics File Input

5.6.2. Sending Messages

5.6.3. Leaving the Application

5.7. Messages Generation

5.7.1. Creating or Modifying Messages

5.7.2. Loading Free Message

5.8. AVIONICS Data

6. MEANS INVOLVED / TEST PROCEDURES
7. COSTS OF THE TLDP DEVELOPMENT
   7.1. Staff Effort
   7.2. Hardware and Software
   7.3. Cost Efficiency

8. GENERAL CONSTRAINTS, ASSUMPTIONS AND DEPENDENCIES

9. CONCLUSIONS

10. REFERENCES

Annex 1: TXID.ASM Detailed Architectural Design

1. TXID.ASM DOWNLOAD FLOWCHART
   1.1. AAC Low Level Main Program Flowchart
   1.2. AAC Low Level Interrupt Procedure Flowchart
   1.3. Control Word Format at Buffer Level
   1.4. Data Word Format at Buffer Level
   1.5. HARRIS Processor Word Format for ARINC
   1.6. Low Level versus High Level Dialogue Process

Annex 2: Uplink versus Downlink Flowcharts

1. BASIC MECHANISM TO RECEIVE A WORD
   1.1. Main Program Condition
   1.2. Interrupt Procedure Condition

Test Data Link Processor
2. BASIC MECHANISM TO RECEIVE A RECORD
   2.1. Main Program Condition
   2.2. Interrupt Procedure Condition
   2.3. Uplink Record Processing

3. BASIC MECHANISM TO SEND A WORD
   3.1. Main Program Condition

4. MAIN MECHANISM TO SEND A RECORD
   4.1. Main Program Condition

Annex 3: Ground Initiated Comm-B Message Generation
   1. BASIC MECHANISM TO SEND A GICB

Annex 4: User Free Text Message Generation
   1. BACKUP MESSAGES FORMAT PROCESS
   2. ON LINE MESSAGE

Annex 5: Downlink Messages Photos
   Photo No. 1
   Photo No. 2
   Photo No. 3
   Photo No. 4
I. INTRODUCTION

1. ACRONYM AND ABBREVIATION DEFINITIONS

AAC       Advanced ARINC Card
ACAS      Airborne Collision Avoidance System
ADLP      Aircraft Data Link Processor
AGLAEE    Air Ground data Link Applications Experimentations
AICB      Air Initiated COMM-B Message
AP field  Address/Parity field
ARINC     Aeronautical Radio INcorporated
ATCRBS    Air Traffic Control Radar Beacon System
ATC       Air Traffic Control
ATIS      Automatic Terminal Information System
BACK1     BACKup message file N° 1
BACK2     BACKup message file N° 2
BACK3     BACKup message File N° 3
BACK4     BACKup message File N° 4
BDS       Binary Data Store
DLPU      Data Link Processor Unit
EASIE     Enhanced Air Traffic Management and Mode-S Implementation in Europe
EPROM     Erasable Programmable / Read Only Memory
GICB      Ground Initiated Comm-B message
GPFT      General Purpose File Transfer
MA        Field Message COMM-A
MC        Field Message COMM-C
SD        Field Special Designator (16 bit)
VME       Versatile Multiprocessor Europe
XPDR      Transponder

2. PURPOSE AND SCOPE

This document describes the TEST DATA LINK PROCESSOR in its functional and operative aspects.

Taking into account the existence of two types of DLPU's, the DATA LINK PROCESSOR UNIT laboratory model and the C2 model certified for commercial aircraft, EUROCONTROL Agency, in the context of the EASIE programme, decided to develop a DLPU for test purposes based on a PC and using an ADVANCED ARINC CARD (EEC Note No. 09/91).
The performance capabilities of this project are similar in the major aspects to the existing DLPU's concepts. The software implemented design includes, beside a real time working function, the possibility to generate user avionics data and user free text messages for data link purposes.

A wide number of users can be foreseen considering the implementation of a large number of MODE-S radars in Europe on which the TDLP can be useful to test and implement air-ground data link capabilities.

This document includes:

a) An introduction describing the purpose of this document.

b) A global software description for all the processes which work in real time.

c) User facilities description for the generation of user avionics data and user free text messages.

d) Means involved / Test procedures.

e) Costs of the TDLP development.

f) General constraints, assumptions and dependencies.

g) Conclusions.

h) References.

i) Annexes.
II. GENERAL OVERVIEW

The EUROCONTROL AGENCY, in the context of the EASIE programme, has made available to European countries participating in the MODE-S trials, MODE-S Transponders and several models of DLPUs (DATA LINK PROCESSOR UNIT) which can be described in the following way:

- 3 DLPUs model A/B for experimental aircraft.
- 3 DLPUs model laboratories for ground installations.
- 12 DLPUs model C2 certified for commercial aircraft, with data link terminals and ARINC printers.

The task of the DLPU is to assure the data link between, on one side, the pilot and AVIONICS equipments (ARINC 429 protocol) and, on the other side, the MODE-S transponder (ARINC 718 protocol) which is in charge to support High Frequency data link with the MODE-S ground stations.

The link supported by the DLPU can be divided into two types. The first concerns control messages or text messages like meteorological information, flight plans, pilot's navigation procedures, etc. The other type concerns aircraft flight data parameters which are read from the aircraft ARINC bus and updated through appropriate messages in the MODE-S transponder.

The DLPU is a not a standardized version of the ADLP - AIRCRAFT DATA LINK PROCESSOR, but the hardware of the DLPU can, if available, execute the ADLP software (connection to the aircraft router) plus the application processes of the DLPU.
A modern laboratory DLPUs, shown in Figure No. 1, is composed of 3 VME cards, a CPU with memory and SCSI interface for hard disc, floppy, operator terminal and pilot terminal, 2 VME ARINC cards, one for the transponder interface and a second VME ARINC card for the aircraft interface.

To run this DLPUs, an aircraft data simulator must be available. Inputs for downlink data are done manually, which for test purposes is not the best solution.
The number of laboratory DLPUs is not sufficient because each radar needs a test MODE-S transponder connected to a DLPU. Also, the laboratory DLPU, a copy of a DLPU in a standard VME chassis, needs an aircraft data simulator.

Even a laboratory DLPU built up with standard material still is quite expensive, about 16,000 ECU per unit. Therefore, the EEC developed a TDLP - TEST DATA LINK PROCESSOR, a test DLPU based on a PC. The first version of this TDLP, the less expensive one, running under MS-DOS, is described in this technical Note.
III. THE TEST DATA LINK PROCESSOR

1. GENERAL ASPECTS

The TEST DATA LINK PROCESSOR was born from the need to create an easy and flexible instrument to monitor and test the MODE-S DATA LINK at several levels.

Figure No. 2 - TDL P TEST CHAIN

Test Data Link Processor
Figure No. 3 - TDLP GENERAL OVERVIEW
Figures Nos. 2 and 3 show the basic TDLP implementation for a data link test chain with the purpose of monitoring uplink messages with real time analysis and a downlink messages generation facility including automatic update of avionics data in the XPDR as well as user free messages generation. There is no aircraft interface, the aircraft data to be sent to the XPDR are in pre-prepared selectable files. Using a simple message data base or typing his one message, the user, by simply pressing a button of the keyboard, can activate a downlink message process. Uplink messages are displayed on the screen, downlink messages excluding avionics data are also displayed but using reversed video.

2. HARDWARE ENVIRONMENT

Seen from the MODE-S ground station, there is no difference whether the transponder is connected to a DLPU, a laboratory DLPU or a TDLP. The ADVANCED PC ARINC CARD (EEC Note No. 09/91) installed on a PC-AT are the two hardware components of the TDLP. The dual in quadruple out ADVANCED PC ARINC CARD (AAC) was specially programmed to accept full duplex dialogue in ARINC 718 protocol between a TRANSPONDER (XPDR) and a PC. Providing common PC memory data share, the card is able to insure low level access to words or files by the PC or vice-versa.

3. SOFTWARE ENVIRONMENT

The Software Project Component is a successful mix, on a DOS environment, of low level and high level software languages.

For these purposes, two different types of software have been developed, the first running on the card, is the running 'FIRMWARE', the other one running on the PC is the software of the TDLP.

The firmware (TXID.ASM) was written on MicroSoft Macro ASSEMBLER language and basically is a ARINC 718 dialogue support program. This program is able to perform an efficient real time ARINC 718 data dialogue and data formatting processes (about 200 μs by ARINC 718 word). The possibility of using a high level language like Logitech MODULA-2 instead of ASSEMBLER proved successful only for the ARINC 429 protocol due to time restrictions for the data formatting.

The program has an important feature that profits from the card hardware design which is an automatic program loading capacity. The use of this technique can be adapted to a wide range of applications providing an easy software expansion capability with on line debugging and memory management. In parallel and also to simplify its use, memory analysis software tools were developed creating a simple and transparent memory view including real time access for reading or writing processes. These software tools were developed internally and will be described in a future document.

Test Data Link Processor
The PC program (TDLP.PAS) was written in Borland PASCAL. In general terms, it supports screen and keyboard management, AAC operating processes including IN/OUT common memory access, a user facility to generate avionics data with consequent on line loading capacity on transponder BDS (Binary Data Store) tables and finally user free text messages generation for downlink purposes.
4. ARCHITECTURAL DESIGN

Figure No. 4: TDLP Architectural Design
4.1. TXID.ASM General Overview

In reality, TXID.ASM, which is described in Annex 1, is composed of two different parts, a loader and the loaded part. Using the card’s possibility of sharing the PC local RAM memory, this program loads a part of its own, which corresponds to the card Firmware (Figure No. 4), to the card memory which is shared by the PC. To prevent the card microprocessor from running during the load operation, a HALT command is executed. With this capability, the common use of EPROMS and similar techniques were replaced by a more flexible and reliable way of using firmware in Random Access Memory (RAM).

4.2. TDLP.PAS General Overview

In order to produce a simple software design associated with evolutive maintainability, Borland PASCAL was used as a high level language to implement this software package.

There are 4 basic high level processes to perform the real time operation of the TDLP; Uplink, Downlink, keyboard control and screen control processes, (Figure No. 4). Associated with the downlink process, there are two data bases; Avionics data base and downlink messages data base. The role and influence of all these features will be described in the next chapters.

As mentioned above, the operating system used was MS-DOS. During real time operation, due to the fact that DOS is not a multitask operating system, the reception of any uplink message has full priority among all the others tasks.

So, when this happens an interrupt signal is produced by the card. This incoming information is kept in a standby buffer waiting to be treated. The use of the MS-DOS environment as a non multitasking and a non reentrant operating system creates several constraints on the implementation of a real time test data link processor unit.

Due to this fact, keyboard and screen management were implemented in a non standard way. These constraints and the actions taken to overcome these problems will be explained in Chapter no. 8.

The structure of the program may be described as a normal sequence of actions that can be software interrupted at any time without breaking the normal functioning of the program.
4.3. **TDLP Real Time Simulation Facility**

4.3.1. *Uplink Process*

4.3.1.1. Validate Word / Record .

The meaning of word on this process level corresponds to a 31 bit string, without parity bit, already back formatted and ready to be analyzed. A record information can correspond from a minimum group of 6 words (COMM-A) to a maximum group of 24 words (COMM-C) including the checksum.

Depending on whether the input information is a record or a control word, one of two types of buffers is allocated to store this information assuming that no errors were detected on the transmission between the transponder and the TDLP. The PC is informed of these two events in two ways: the card sends an interrupt signal and at the same time it modifies a global variable. These two processes are described in more detail in *Annex 2*.

4.3.1.2. Messages Types

For display purposes, the card monitors two types of uplink messages; COMM-A and COMM-C (ELM). Surveillance messages are simply monitored.

A clear description of these three types of message format can be found in *ICAO - Annex 10* or in the ARINC ATCRBS / MODE-S document.

All these messages are sent by the transponder (XPDR) to the TDLP without the AP field.

Surveillance messages are sent in 3 words and unlinked COMM-A messages are sent in 5 words (1 record). A linked COMM-A message can go to a maximum of 4 records.

An unlinked COMM-C message also known as an extended length message (ELM) is sent on a minimum of 5 words (1 segment) to a maximum of 23 words (6 segments - 1 record ).
An unlinked ELM can eventually go to 16 segments (Figure No. 5) which means sending 2 records with 6 segments each and a last one with 16 words (4 segments).

<table>
<thead>
<tr>
<th>NUMBER OF SEGMENTS</th>
<th>NUMBER OF WORDS</th>
<th>NUMBER OF RECORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>23 + 5</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>23 + 9</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>23 + 12</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>23 + 16</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>23 + 20</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>23 + 23</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>23 + 23 + 5</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>23 + 23 + 9</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>23 + 23 + 12</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>23 + 23 + 16</td>
<td>3</td>
</tr>
</tbody>
</table>

**Figure No. 5**

*Note:* This Figure shows the number of records as well as the number of words necessary to transmit from 1 segment to 16 segments ELM message.

4.3.1.3. Messages Validation

The transponder introduces the code for the first ARINC word in any uplink message.
Assuming that there is no message code error, the received record is assembled at this process stage. For this, two conditions must be verified at the same time; Correct word 1 and TDLP header format. These format codes are shown below.

<table>
<thead>
<tr>
<th>BIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>52 51 50 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>L</th>
<th>CRM</th>
<th>NOT DEFINED</th>
<th>UII</th>
<th>RCN</th>
<th>UMT</th>
<th>UDT</th>
<th>C</th>
<th>T</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

**Figure No. 6 : XPDR to TDLP Word 1 Format versus Field's Bit Number**

**Comments:**

Field content and coding (all fields have MSB at right)

T : Transmission Control Bit Field

CTS : Clear to Send Bit Field

UDT : Uplink Data Transfer

  0 = Data transfer of an interrogation  
  (with interrogation type specified in UMT field)

  1 = Comm-B close-out (with RCN for current register number)
  2 = Comm-B broadcast time-out
  3 = Downlink ELM close-out (with RCN for current register number)
  4-7 = Not assigned

UMT : Uplink Message Type

  0 = No uplink data
  1 = Surveillance interrogation
  2 = Comm-A interrogation
  3 = Comm-A broadcast interrogation
  4 = Uplink ELM
  4-7 = Not assigned

RCN : Register Control Number

UII : Uplink II, Interpolator Identification

*Test Data Link Processor*
CRN : Comm-C record number

0 = No ELM data
1 = 1st ELM record (segments 0 to 5)
2 = 2nd ELM record (segments 6 to 11)
3 = 3rd ELM record (segments 12 to 15)

LRI : Last Record Indicator

0 = First or intermediate record
1 = Last record

P : Parity

SD<---|--->MA

<table>
<thead>
<tr>
<th>LAS</th>
<th>FTS</th>
<th>Spare</th>
<th>PRS</th>
<th>UTF</th>
<th>NTC</th>
<th>APN</th>
<th>GPN</th>
<th>RCO</th>
<th>MSN</th>
<th>UPM</th>
<th>CID</th>
<th>URD</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0,3</td>
<td>3</td>
<td>5</td>
<td>31</td>
</tr>
</tbody>
</table>

Figure No. 7 : TDLP Header Format for a COMM-A Message
with Header Fields versus Bit Number

+---->MC

<table>
<thead>
<tr>
<th>FTS</th>
<th>LES</th>
<th>Spare</th>
<th>PRS</th>
<th>UTF</th>
<th>NTC</th>
<th>APN</th>
<th>GPN</th>
<th>RCO</th>
<th>MSN</th>
<th>UPM</th>
<th>CID</th>
<th>URD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0,3</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Figure No. 8 : TDLP Header Format for a COMM-C Message
with Header Fields versus Bit Number

Test Data Link Processor
Comments:

MA : Message COMM-A
MC : Message COMM-C
SD : Special Designator

LAS : Linked COMM-A Subfield

0 = Unlinked COMM-A
1 = First record of a linked message
2 = Second record of a linked message of 3 or 4 records
3 = Third record of a linked message of 4 records
4 = Last record of a linked message of 4 records
5 = Last record of a linked message of 2 records
6 = Last record of a linked message of 3 records

LES : Linked ELM Subfield

0 = Unlinked ELM
1 = Linked ELM (initial component)
2 = Linked ELM (intermediate component)
3 = Linked ELM (final component)

FTS : Format Type Subfield. Always 1. ( Datagram header)

PRS : Priority Subfield

0 = Standard priority
1 = High priority

UTF : Uplink Transport Format. Always 1

NTC : Network Type Code. Irrelevant for present implementation

APN : Airborne Process Name

1 = Airborne ATC process
2 = Airborne weather process
3 = Airborne GPFT process (General Purpose File Transfer)

GPN : Ground Process Name

1 = Ground ATC process
2 = Ground weather process
3 = Ground GPFT process (General Purpose File Transfer)
RCO : Response Control Option

0 = No acknowledgment request
1 = Acknowledgment request
2 = Acknowledgment for message number

MSN : Message Number to acknowledge (0 - 7) if RCO indicates reply

UPM : Uplink Process Message number (0 - 7)

CID : (Raw Data) Coding Identification

0 = 5-Bit character coding
1 = 6-Bit character coding
2 = Phrase coding

URD : Uplink Raw Data

A full detailed description of the TDLP uplink header format can be found in TUB Laboratory DLPU document. Considering the 56 bit of the MA field of a COMM-A message, the first component contains the complete header (25 bits if MSN=0) and only 31 bits of URD. The following components (if present) contain the segmented data (URD) without any header in the MA field. For a COMM-C (ELM) message, the first component contains the complete header (27 bits if MSN=0) and some bits of URD (53 bits). A linked ELM is not implemented, LES field is ignored in this version of the TDLP.

As specified in the CID field, there are 3 coding types for uplink messages. In order to optimize software work and also having in mind users environment necessities, for the moment, any 5 bit coding message is simply indicated on the screen. The 6 bit coding is implemented as shown in Figure No. 9.

Phrase coding is implemented only with the indication on the screen if it is an ATC or WEATHER phrase coded process and the corresponding phrase dictionary number. This phrase dictionary is described in TUB Laboratory DLPU document.

The ATC phrase number is coded in the next 4 bits after the header and the WEATHER phrase number is coded in the 10 bits following the header.
Figure No. 9 : 6 Bit Code Table

4.3.1.4. Screen Management

The screen management is basically a circular buffer which has a common part with the PC RAM area that contains the video display information which is itself operated by the DOS BIOS routines. Using this technique, any interrupt signal coming from the card will not provoke any "crash" on the program main body. This characteristic reflects the unavailability of procedure re-entry under DOS when an interrupt occurs inside a conventional PASCAL screen writing routine.

4.3.2. Downlink Process

4.3.2.1. Keyboard Management

The keyboard management is performed by a polling action on the PC keyboard processor which contains the code of the last pressed button. Also in this feature, standard PASCAL routines cannot be used because of the same reasons mentioned above.
4.3.2.2. Messages Library

This library is composed of 18 messages. As shown in Figure No. 10, 14 Messages are pre-programmed. They can be only modified in the source code (TDLP.PAS); the last 4 Messages can be modified by the user (BACK1, BACK2, BACK3, BACK4).

To differentiate the detection of all these messages, it became necessary to allocate for each message a keyboard button code, as mentioned in "KEY" field in Figure No. 10. Each message is coded according to the code table shown in Figure No. 9.
## AZERTY Keyboard Message Allocation

<table>
<thead>
<tr>
<th>KEY</th>
<th>MESSAGE No.</th>
<th>MESSAGE TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'1'</td>
<td>1</td>
<td>'COMM-B'</td>
</tr>
<tr>
<td>'2'</td>
<td>2</td>
<td>'COMM-B LINKED 2'</td>
</tr>
<tr>
<td>'3'</td>
<td>3</td>
<td>'TEST COMM-B LINKED 3'</td>
</tr>
<tr>
<td>'4'</td>
<td>4</td>
<td>'TEST COMM-B LINKED 4'</td>
</tr>
<tr>
<td>'5'</td>
<td>5</td>
<td>'............TEST ELM COMM-D MESSAGE WITH 6 SEGMENTS............'</td>
</tr>
<tr>
<td>'6'</td>
<td>6</td>
<td>'............TEST ELM COMM-D MESSAGE WITH 10 SEGMENTS......'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'............TEST ELM COMM-D MESSAGE WITH 10 SEGMENTS..'</td>
</tr>
<tr>
<td>'7'</td>
<td>7</td>
<td>'............THIS A DOWNLINK'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'N LINK MESSAGE FOR TESTING 4 ELM W'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'ITH 16 SEGMENTS WITH APN = 1 .......'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'...............THIS IS A DOW'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'N LINK MESSAGE FOR TESTING A ELM W'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'ITH 16 SEGMENTS WITH APN = 1..'</td>
</tr>
<tr>
<td>'8'</td>
<td>8</td>
<td>'AF405 REQUEST FL345 AT NANTES WAYPOINT'</td>
</tr>
<tr>
<td>'9'</td>
<td>9</td>
<td>'SAB546 REQUEST DIRECT ROUTING ON FL240 ON PARIS/ORLY'</td>
</tr>
<tr>
<td>'0'</td>
<td>10</td>
<td>'TAP420 REQUEST METEO REPORT FOR EDDF'</td>
</tr>
<tr>
<td>'A'</td>
<td>11</td>
<td>'LUF345 REQUEST ATIS INFORMATION FOR LPPO'</td>
</tr>
<tr>
<td>'Z'</td>
<td>12</td>
<td>'BRI345 REPORTING CB TURBULENCE AT FL300'</td>
</tr>
<tr>
<td>'E'</td>
<td>13</td>
<td>'ABC345 REPORTING UFO AT BRETIGNY REGION'</td>
</tr>
<tr>
<td>'R'</td>
<td>14</td>
<td>'AF345 REPORTING ALL FREQUENCIES OFF LINE INCLUDING EMERGENCY 119.1'</td>
</tr>
<tr>
<td>'F1'</td>
<td>BACK1</td>
<td></td>
</tr>
<tr>
<td>'F2'</td>
<td>BACK2</td>
<td></td>
</tr>
<tr>
<td>'F3'</td>
<td>BACK3</td>
<td></td>
</tr>
<tr>
<td>'F4'</td>
<td>BACK4</td>
<td></td>
</tr>
</tbody>
</table>

Figure No. 10: Downlink Messages versus Keyboard Allocations

*Test Data Link Processor*
4.3.2.3. Messages Generation

A flow chart analysis of this process can be found in *Annex 4.*

There are three types of messages implemented by the TDLP; Ground Initiated COMM-B (GICB) messages, Air Initiated COMM-B (AICB) messages and finally COMM-D Extended Length Messages.

GICB messages, that will be described in more detail in Chapter 4.3.2.4., carry exclusively aircraft data parameters, like wind speed, time to go, latitude, etc.

AICB and COMM-D messages carry meteorological or ATC information. All these three types of messages are sent by the TDLP without the AP field. A clear description of these three types of message’s format can be found in ICAO *Annex 10* or in ARINC ATCRBS / MODE-S document.

Unlinked AICB messages are sent in 4 words (1 record). Linked AICB messages can go to a maximum of 4 records.

COMM-D and COMM-C messages are formally identical. A segment corresponds to a MD field and linked COMM-D messages are not considered on this TDLP version.

The TDLP introduces the code to the first ARINC word in any downlink message. Downlink header formats are implemented only in AICB’s and ELM messages. Word 1 and header format code are shown below:

<table>
<thead>
<tr>
<th>BIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>L</th>
<th>DRN</th>
<th>NOT DEFINED</th>
<th>DII</th>
<th>RCN</th>
<th>DMT</th>
<th>DDT</th>
<th>C</th>
<th>T</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

*Figure No. 11: TDLP to XPDR Word 1 Format versus Field’s Bit Number*

**Comment:**

Field content and coding (all fields have MSB at right)
T : Transmission Control Bit Field

CRS : Clear to Send Bit Field

DDT : Downlink Data Transfer
0  =  Transfer of reply data
1  =  Cancel message (associated with DMT and RCN)
2-7 =  Not assigned

DMT : Downlink Message Type
0  =  Ground-initiated COMM-B
1  =  Air-initiated COMM-B
2  =  Air-initiated COMM-B broadcast
3  =  Downlink ELM
4-7 =  Not assigned

RCN : Register Control Number

UII : Downlink II, responsible for present message

BDS : Register address of COMM-B buffer for present message segment

DRN : COMM-D Record Number
0  =  No ELM data
1  =  1st ELM record (segments 0 to 5)
2  =  2nd ELM record (segments 6 to 11)
3  =  3rd ELM record (segments 12 to 15)

LRI : Last Record Indicator
0  =  First or intermediate record
1  =  Last record

P : Parity

+-----+ MB, MD

<table>
<thead>
<tr>
<th>PTS</th>
<th>LSS/LES</th>
<th>PRS</th>
<th>NTC</th>
<th>GPN</th>
<th>APN</th>
<th>RCO</th>
<th>DDM</th>
<th>RCO</th>
<th>MSN</th>
<th>DPM</th>
<th>CID</th>
<th>DRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0,3</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

**Figure No. 12 : TDLP Downlink Header Format with Header Fields versus Bit Number**

*Test Data Link Processor*
**Comments:**

MB : Message COMM-B  
MD : Message COMM-D  

FTS : Format Type Subfield. Always 1 (Datagram header)  

LBS : Linked COMM-B Subfield  

0 = Unlinked COMM-B  
1 = Linked COMM-B (2 component message)  
2 = Linked COMM-B (3 component message)  
3 = Linked COMM-B (4 component message)  

LES : Linked ELM Subfield  

0 = Unlinked ELM  
1 = Linked Comm-B (initial component)  
2 = Linked Comm-B (intermediate component)  
3 = Linked Comm-B (final component)  

PRS : Priority Subfield  

0 = Standard priority  
1 = High priority  

NTC : Network Type Code - Irrelevant for present implementation  

GPN : Ground Process Name  

1 = Ground ATC process  
2 = Ground weather process  
3 = Ground GPFT process (General Purpose File Transfer)  

APN : Airborne Process Name  

1 = Airborne ATC process  
2 = Airborne weather process  
3 = Airborne GPFT process (General Purpose File Transfer)  

DDM : Downlink Data Link Message Number. Irrelevant for present implementation  

RCO : Response Control  

0 = No acknowledgment request.  
1 = Acknowledgment request  
2 = Acknowledgment for message number  

*Test Data Link Processor*
MSN : Message number to acknowledge (0 - 7) if RCO indicates reply

DPM : Downlink Process Message Number (0 - 7)

CID : Coding Identifier

0 = 5-Bit character coding
1 = 6-Bit character coding
2 = Phrase coding

DRD : Downlink Raw Data. User data

A full detailed description of the TDLP uplink header format can be found in TUB Laboratory DLPDU document. Considering the 56 bits of the MB field of a COMM-B message, the first component contains the complete header (29 bits) and only 27 bits of URD. The following components (if present) contain the segmented data (URD) without any header in the MB field. For a COMM-D (ELM) message the first component contains the complete header (29 bits) and some bits of URD (51 bits). A linked ELM is not implemented (LES field is ignored).

4.3.2.4. Ground Initiated COMM-B Library

A most useful feature of the TDLP is the possibility to generate simulated Avionics data. This data will update the BDS (Binary Data Store) of the XPDR.

The user is able to create his sequence of data and to store them on an Avionics data file which is processed at the beginning of the program.

How to handle that file is a subject for the operating aspects part (Chapter 5.8.). Avionics data specifications can be found on ARINC DOCUMENT MARK 33 DITS 429-14.

There is no data variation during the real time simulation and time out data flow (refresh time) is constant for all parameters.

Ground initiated COMM-B messages (GICB’s) are responsible to transfer these information from the TDLP to the MODE-S XPDR (Annex 3). A GICB message is also composed of 4 words.

On the 56 bits of the MB field several Avionics data information are concatenated, each message will load a different BDS address on the XPDR. The TDLP implements 26 BDS that are described on the table below (Figure No. 13):

Test Data Link Processor
<table>
<thead>
<tr>
<th>BDS</th>
<th>ALLOCATED TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Capability Report</td>
</tr>
<tr>
<td>20</td>
<td>Aircraft Identification</td>
</tr>
<tr>
<td>40</td>
<td>Path &amp; Attitude Report</td>
</tr>
<tr>
<td>50</td>
<td>Speed &amp; Track Report</td>
</tr>
<tr>
<td>60</td>
<td>Ground Referenced Speed Vector</td>
</tr>
<tr>
<td>70</td>
<td>Aircraft Intention No. 1</td>
</tr>
<tr>
<td>80</td>
<td>Aircraft Intention No. 2</td>
</tr>
<tr>
<td>90</td>
<td>Aircraft Intention No. 3</td>
</tr>
<tr>
<td>A0</td>
<td>Deviation Report</td>
</tr>
<tr>
<td>B0</td>
<td>Airspeeds Report</td>
</tr>
<tr>
<td>C0</td>
<td>Altitude Report</td>
</tr>
<tr>
<td>21</td>
<td>Aircraft Referenced State Vector</td>
</tr>
<tr>
<td>31</td>
<td>Waypoint Information</td>
</tr>
<tr>
<td>41</td>
<td>Waypoint Identifier</td>
</tr>
<tr>
<td>51</td>
<td>Position Report</td>
</tr>
<tr>
<td>61</td>
<td>G.M.T. Report</td>
</tr>
<tr>
<td>71</td>
<td>Meteorological Report</td>
</tr>
<tr>
<td>24</td>
<td>Weight &amp; Balance Report</td>
</tr>
<tr>
<td>17</td>
<td>Engine No. 1 Report</td>
</tr>
<tr>
<td>27</td>
<td>Engine No. 2 Report</td>
</tr>
<tr>
<td>37</td>
<td>Engine No. 3 Report</td>
</tr>
<tr>
<td>47</td>
<td>Engine No. 4 Report</td>
</tr>
<tr>
<td>18</td>
<td>B767 Status Words</td>
</tr>
<tr>
<td>19</td>
<td>B767 Status Words</td>
</tr>
<tr>
<td>1A</td>
<td>Aircraft Identification Part 1</td>
</tr>
<tr>
<td>1B</td>
<td>Aircraft Identification Part 2</td>
</tr>
</tbody>
</table>

Figure No. 13 : BDS Allocation List

The precise description of all the fields (Avionics data) existing in these BDS group can be found in EEC Note No. 10 / 92 "BDS Allocation for DLP-U- C".
5. **TDLP USER MANUAL**

5.1. **Introduction**

The purpose of this Chapter is to give a general overview how a user can implement this software package and, at the same time, to describe how to use all the TDLP software facilities.

In addition to DLPU capabilities, two more features were implemented. The first enables the user to write his own downlink messages and the second one concerns the generation of an Avionics data file for GICB generation purposes. These two possibilities cannot work in parallel due to DOS mono-tasking operating system.

5.2. **Hardware Requirements**

There are no special hardware requirements besides a PC-AT and an ADVANCED ARINC CARD to implement the TDLP software package. This advanced ARINC card can be made available through the EUROCONTROL Experimental Centre.

5.3. **Software Requirements**

To run TDLP it is necessary to have several distinct files that will be explained as follows:

(a) GICB.DAT; Default file with AVIONICS data,

(b) BACK1, BACK2, BACK3, BACK4; Backup files for user downlink messages generation;

(c) TDLP.EXE; The host executable file,

(d) TXID.EXE; The firmware download program,

(e) TDLPB.BAT; Batch file to start running the Test Data Link Processor Facilities.

5.4. **Starting TDLP**

To start running the program, the user should type "TDLPB". This batch file, before calling TDLP.EXE, starts loading the card firmware TXID.EXE. When the download is completed, the following message is displayed:

"TXID LOAD COMPLETED"
Afterwards for each downlink message code generation, the following message is displayed:

"DOWNLINK MESSAGE x GENERATION FINISHED"

When all messages are processed, the main menu is displayed.

5.5. Main Menu

DATA LINK PROCESSOR UNIT Real Time Simulation Facilities Program
E.E.C. October 1993 by H.P. ENGLMEIER and J.M. MATEUS MARTINS

TEST DATA LINK PROCESSOR < 1 >

User Free Text Messages Generation < 2 >

Creating User AVIONICS Data < 3 >

Type < Q/q > to Quit to DOS

According to your application, choose your option:
Are you sure? :

Figure No. 14: Main Menu

5.6. TDLP Real Time Simulation

The first option (< 1 >) will initiate the real time simulation process.
5.6.1. **Avionics File Input**

After having chosen option < 1 >, the user should introduce the file name where the program will get all the Avionics data information needed for the simulation. If pressing < ENTER >, GICB.DAT default file is loaded. After having processed this information, the program sends the following message:

" GICB GENERATION FINISHED "

5.6.2. **Sending Messages**

As mentioned above (Figure No. 10), messages can be sent by simply pressing some keyboard keys. When the downlink process is concluded, the message is shown on the screen in reverse video. Because the program is not based on a multitask environment it is possible that this action does not take place immediately, for example, if the program is sending GICB messages or if an uplink message is being processed.

The keyboard routine considers only once the last pressed button, to repeat the message, type a non allocated button, e.g. < ESC >.

5.6.3. **Leaving the Application**

To leave the application type < 'Q' > or < 'q' > and the program will return to the main menu.
5.7. Messages Generation

The second option (< 2 >) will initiate the user free text messages generation facility, that will display the menu shown in Figure No. 15.

Data Link Processor Unit Real Time Simulation Facilities Program
E.E.C. October 1993 by H.P. Englmeier and J.M. Mateus Martins

For generate free text messages you have two options:

Create or modify a message < 1 >

Load a message < 2 >

According to your application choose your option:

Figure No. 15: User Free Text Message Generation Menu

5.7.1. Creating or Modifying Messages

Selecting keyboard key "1", the user has the possibility of creating a message with a maximum of 208 characters which represent an ELM COMM-D of 16 segments. It is also possible to modify any existing file. Backup files (BACK1..4) can be a good example. In a future version, the user will be able to define his own backup files.
5.7.2. **Loading Free Message**

Taking into consideration Chapter 4.3.2.2., this facility permits the implementation of the 19th message allocated to any other free key button. The user only has to introduce the file name where the message is stored.

5.8. **AVIONICS Data**

With the third option (< 3 >), the user has the possibility to create or modify AVIONICS data files. Starting with GICB.DAT default file, any other file can be created. The interface display image is shown in *Figure No. 16.*

**BDS Generation Table**

<table>
<thead>
<tr>
<th>BDS 40</th>
<th>BDS 50</th>
<th>BDS 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fli. Path Angle: Ram. +180° Res. = 0.18***</td>
<td>Fli. Path Accler.: Ram. = 4g Res. = 0.001</td>
<td>Track Angle Magn.: Ram. +180° Res.</td>
</tr>
<tr>
<td>00000000000000</td>
<td>000000000000</td>
<td>00000000000000</td>
</tr>
<tr>
<td>000000000000</td>
<td>000000000000</td>
<td>00000000000000</td>
</tr>
</tbody>
</table>

| Mack Number: Ram. 4.096 Mach Res. 0.001 | Comp. Air Speed: Ram. 1024Kt Res. 0.001 | Track Angle Magn.: Ram. +180° Res. 0.044 |
| 000000000000            | 000000000000           | 00000000000000         |
| Track Angle True: Ram. +180° Res. 0.044 | Track Angle Magn.: Ram. +180° Res. 0.044 | Static Air Temper. Ram. +512° Res. 0.5 |
| 00000000000000         | 00000000000000         | 00000000000000         |
| Altit. Rate: Ram. +16384ft/1min Res. 64** | Static Air Temper. Ram. +512° Res. 0.5 | 000000000000 |
| 00000000000000         | 00000000000000         | 00000000000000         |

*Test Data Link Processor*
BDS 70
Selected Altitude: Ran. 65536Ft Res. 16**Selected Heading: Ran.+180° Res. 0.8
0000000000000000
000000000000

Status Word 1 : LABEL 270  Status Word 2 : LABEL 271
"Binary input" 0000000000000000  "Binary input" 0000000000000000

BDS 80
Selec. Air Speed: Ran. 512Kt Res. 0.25 **Selec. Alt. Rate: Ran. 16384Ft/Min Res. 16
0000000000000000
000000000000

Selected Altitude : Ran. 65536Ft Res. 16  Status Word 1 : LABEL 270
0000000000000000  "Binary input" 0000000000000000

BDS 90
Sel.Mach Numb: Ran. 4096Mach Res. 0.002***Sel. Alt. Rate: Ran. 16384Ft/Min Res. 16
0000000000000000
000000000000

Selected Altit. : Ran. 65536Ft Res. 16  Status Word 2 : LABEL 271
0000000000000000  "Binary input" 0000000000000000

BDS A0
Cross Track Error: Ran. 128Nm Res. 0.004 **Vertical Deviation: Ran. 2048Ft Res. 1
0000000000000000
000000000000

Acceleration : Ran. +1g Res. 0.016
000000000000

BDS B0
Comput. Air Speed: Ran. 1024Kt Res. 0.25**True Air Speed : Ran. 2048Kt Res. 0.5
0000000000000000
000000000000

Mach : Ran. 4.096Mach Res. 0.00025  Heading (Mag) : Ran. +180° Res. 0.18
0000000000000000
000000000000

Test Data Link Processor
BDS 21
Magnetic Heading : Ran. +180° Res. 0.18**True Air Speed : Ran. 2048Kt Res. 0.5
0000000000000000
000000000000
Static Air Temper.:Ran. +512°C Res. 0.5*Normal Acceler. : Ran. +1g Res. 0.016
0000000000000000
00000000000000

BDS 31
Bearing To WayPoin.:Ran.+180° Res. 0.18 Time To Go: Ran. 399.9Min Res. 0.1
0000000000000000
00000000000000

Distance : Ran. 3999.9Nm Res. 0.1
00000000000000

BDS 41
Active Waypoint 1 : 3 Chr ISO 5 Active Waypoint 2 : 3 Chr ISO 5
0000000000000000
00000000000000

Desired Track : Ran. +180° Res. 0.18
00000000000000

BDS 51
Latitude : Ran. +180° Res. 0.0007 Longitude : Ran. +180° Res. 0.0007
0000000000000000
00000000000000

Altitude : Ran.131072Ft Res. 8
00000000000000

BDS 61
G.M.T. Binary Word : HHMMSS Flight Number : XXXX
000000
0000

BDS 71
Roll Angle : Ran.+90° Res. 0.35 Altitude Rate : Ran.+8192Ft/Min Res. 128
0000000000000000
00000000000000

Test Data Link Processor
Wind Speed: Ran. 256Kt Res. 1 0000000000000000
Wind Angle: Ran.+180° Res. 0.703 0000000000000000

BDS C0
Radio Height: Ran. 8192 Ft Res. 1 0000000000000000
Averag. Stat.Press.: Ran. 1024Mb Res. 0.25 0000000000000000

Altitude 1013.25Mb: Ran. 131072Ft Res. 1 0000000000000000

BDS 24
Gravity Center: Ran. 99.9% Res. 0.1 0000000000000000
Gross Weight: Ran. 131072Lbs Res. 160 0000000000000000

BDS 18
FCC Status Word # 1 LABEL 272 "Binary input" 0000000000000000
FCC Status Word # 2 LABEL 273 "Binary input" 0000000000000000

BDS 19
FCC Status Word # 3 LABEL 274 "Binary input" 0000000000000000
FCC Status Word # 4 LABEL 275 "Binary input" 0000000000000000

BDS 17
N1 Act. Engi.#1: Ran. 256%rpm Res. 0.13**Fuel Flow Engi.#1: Ran. 32768Lb/Hr Res. 8 0000000000000000

Total Air Temper.: Ran.+512°C Res. 0.5 Static Air Temper.: Ran.+512°C Res. 0.5 0000000000000000

BDS 27
N1 Act. Engi.#2: Ran. 256% rpm Res. 0.13Fuel Flow Engi.#2: Ran. 32768Lb/Hr Res. 8 0000000000000000

Total Air Temper.: Ran.+512°C Res. 0.5 Static Air Temper.: Ran.+512°C Res. 0.5 0000000000000000

Test Data Link Processor
BDS 37
N1 Act. Engi.#3: Ran. 256% rpm Res. 0.13*Fuel Flow Engi.#3: Ran. 32768 Lb/Hr Res. 8
000000000000 000000000000

Total Air Temper.: Ran.+ -512°C Res. 0.5 Static Air Temper.: Ran.+ -512°C Res. 0.5
000000000000 000000000000

BDS 47
N1 Act. Engi.#4: Ran. 256% rpm Res. 0.13*Fuel Flow Engi.#4: Ran. 32768 Lb/Hr Res. 8
000000000000 000000000000

Total Air Temper.: Ran.+ -512°C Res. 0.5 Static Air Temper.: Ran.+ -512°C Res. 0.5
000000000000 000000000000

BDS 1A
Aircraft Id. Char.2/Char.1: LABEL 233 Aircraft Id. Char.4/Char.3: LABEL 234
00 00

Aircraft Id. Char.5/Char.6: LABEL 235
00

BDS 1B
Aircraft Id. Char.8/Char.7: LABEL 236 Aircraft Id. Char.10/Char.9: LABEL 237
00 00

Figure No. 16

Comments:

1. In each BDS for each item the left bit always indicates the status bit. If status bit is equal to 1 then the data is valid otherwise data is too old or not valid.
2. Range and resolution are indicated in each item.
3. Inputs are decimal if not otherwise indicated.
4. Negative numbers should be indicated with a negative sign.
5. Use arrow keys to access all BDS values.
6. To leave the editor type < ENTER >.
6. MEANS INVOLVED / TEST PROCEDURES

Two test environments were used for the TDLP. One using the TRT test bench available at the EEC and the other using the test chain mentioned in Figure No. 2 with ORLY MODE-S radar. In this case, an end-to-end test could be performed, thanks to the ground data processor AGLAE kindly made available by the Centre d'Etudes de la Navigation Aérienne (CENA) and the MODE-S station which was kindly made available by the Service Technique de la Navigation Aérienne (STNA).

![Diagram of test setup](image)

**Figure No. 17: MODE-S Data Link Test Chain (CENA, STNA, EEC)**

**Comment:** The Orly MODE-S test station is within the line of sight of the transponder antenna which is mounted on top of the EEC building.

*Test Data Link Processor*
7. COSTS OF THE TDLP DEVELOPMENT

7.1. Staff Effort

The software project took approximately 16 man-weeks of work (about 15,000 ECU). Including software maintenance for about 2 years, the project is estimated at 24,000 ECU.

7.2. Hardware and Software

The hardware and the software packages involved can be estimated at 1500 ECU. Altogether, the price of an additional TDLP unit is estimated at 2100 ECU (cost of PC + card), which is less expensive than a VME-based DLPU (16,000 ECU, see page 5) or a DLPU, model C2, 100,000 ECU.

7.3. Cost Efficiency

The costs of the TDLP developments are paid back after the second TDLP which is well below the expected demand.

8. GENERAL CONSTRAINTS, ASSUMPTIONS AND DEPENDENCIES

As mentioned above, the operating system used was MS-DOS. During real time operation, due to the fact that DOS is not a multitasking operating system the reception of any uplink record has full priority among all the others tasks. So, when this happens the full priority is established by an interrupt signal produced by the card. This incoming information is stored in a stand-by buffer waiting to be treated.

This stand-by buffer has a maximum capacity of 25 records. The use of MS-DOS environment as a non re-entrant operating system creates several constraints to implement a real time test data link processor unit.

In order to overcome these constraints, the following solutions were implemented:

(a) Interrupt service routines were implemented to access data from the card using common share variables.

(b) Keyboard management was implemented on a sequential process done basically with polling actions on the PC keyboard processor because keyboard scan routines are excessively long in PASCAL for use with the high speed protocol communication with the MODE-S transponder.

Test Data Link Processor
On this protocol, ARINC 718, that will be described in Annex No. 1, the transmission speed is about 100K Hz.

(c) Screen management was also implemented on a sequential way using DOS facilities of screen addressing.

The TDLP is not prepared to handle information from more than 3 Mode-S radars and also does not implement the following processes:

A - ELM linked messages:

Only the first block is displayed.

B - Phrase coded messages:

For the uplink messages only the process type (ATC or METEO) and the phrase number are displayed.

Only some essential downlink phrase coded messages are implemented.

C - 5 Bit coded messages:

Considering the fact that this coding is very rarely used, the TDLP does not support it.

D - More than 24 BDS loaded.

E - More than 14 messages programmed.

F - More than 4 user backup files.

G - More than 1 user free text file.

H - General Purpose File Transfer (GPFT) still is not implemented even if it has already been tested.
9. CONCLUSIONS

Considering the limitations mentioned in Chapter 8, the PC-AT based TDLP is a very useful and affordable tool for MODE-S Data-Link test purposes. Considering the limitations mentioned in Chapter 8, a more powerful workstation with a multi-tasking operating system would considerably increase the potential of the TEST DATA LINK PROCESSOR with a slight increase in costs. Therefore, it is envisaged that a TDLP application will be developed on a UNIX workstation.
10. REFERENCES

   by P. HUNT

E.E.C. Note No. 10/92 - BDS Allocations for DLPU MODEL-C for Trials with
   AIR FRANCE A310 and BRITANIA B767 Aircraft
   by P. HUNT

E.E.C. Note NO. 09/91 - Advanced PC ARINC card
   by H.P. ENGLMEIER

DLPU User's Guide (B-Model) Version 1.0, 30.05.1988
   by M. LEHMENSIEK
   Technische Universitat Braunschweig Institut für Nachrichtentechnik

ARINC Document - MARK 3 Air Traffic Control Transponder (ATCRBS / MODE- S)
   ARINC Characteristic 718-4, 15.12.1989

ARINC Document - MARK 33 Digital Information Transfer System (DITS)
   ARINC Characteristic 429-12, 1.7.1990

ICAO - Aeronautical Telecommunications (Annex 10)
1. TXID.ASM Download Flowchart

Figure No. 1: TXID.ASM Download Flowchart
Comments:

The on-board processor is set in a HALT position (ADDRESS $301$), the 128 KBytes are tested in two steps, a jump instruction to the low level program is written in the re-start entry location, the low level program is loaded and the on-board processor is restarted (ADDRESS $300$).

1.1. AAC Low Level Main Program Flowchart

![Flowchart]

Figure No. 2: Main Process
Comments:

SENDFLAG control functions:

0 : No operation.

1 : Transmission of a word: the word is fetched out of a buffer (WRDBUF), and is formatted according to ARINC 718 protocol (Chapter 1.5) and ARINC processor (HS 3282) data handling. Finally the word is sent.

2 : Transmission of a record: The transmission of a record starts with the initiation of the checksum and the fetch of the word count. Then, the record is sent word by word with formatting of the word and checksum update (addition module 24) during sending of the previous word. If the word count reaches zero as last word, the checksum is sent.

3 : Before the sending of a word the record input buffer is cleared.
1.2. AAC Low Level Interrupt Procedure Flow Chart

Figure No. 3: Interrupt Process

Test Data Link Processor
Comments:

RECFLAG Control Functions.

0 : Input in a dummy buffer (reset interrupt !).
1 : Reception of a word.

If the received word is a control word, the word is formatted backwards and stored in a circular buffer and the vector for the PC is updated. If the vector flag (VFLAG) is zero, the vector flag is set to one, a word interrupt indicator is set and the PC is interrupted. The VFLAG is a control flag necessary to guarantee synchronous PC data access and corresponding VECTOR pointer update (avoid 8 bit and 16 bit processor conflict).

If the received word is a data word, record reception will be initiated. The reception word count (RWCNT) is fetched, the checksum will be initialized, the RECFLAG is set to 2 and the program continues as if the RECFLAG was 2.

2 : Record word reception.

After having received a data word, immediately during the reception of the next word, the data word is formatted backwards, the checksum is updated and the word is put back in the reception record buffer (RFILE). The last word of the received record is considered as the checksum and compared with the calculated (updated) one. If the two are not equal an error flag is set. Further, the RECFLAG is reset to 1, a record interrupt indicator is set and the PC is interrupted.

The word or record interrupt indicator is a memory location called FLINT and can have the following values:

1 : A discrete word was received.

2 : A record was received.

3 : Error.

This conclusion will be taken by the high level program (TDLP.PAS).
### 1.3. Control Word Format at Buffer Level

<table>
<thead>
<tr>
<th>Address x</th>
<th>Bit</th>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address x+1</td>
<td>Bit</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>09</td>
</tr>
<tr>
<td>Address x+2</td>
<td>Bit</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>Address x+3</td>
<td>Bit</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
</tr>
</tbody>
</table>

*Figure No. 4: Type Format used in WRDBUF and CBUFSTR Buffers*

### 1.4. Data Word Format at Buffer Level

<table>
<thead>
<tr>
<th>Address x</th>
<th>Bit</th>
<th>08</th>
<th>09</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address x+1</td>
<td>Bit</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>Address x+2</td>
<td>Bit</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
</tbody>
</table>

*Figure No. 5: Type Format used in FILE and RFILE Buffers*
1.5. **HARRIS Processor Word Format for ARINC 718**

![Diagram of HARRIS 3282 Processor Format](image)

**Figure No. 6 : HARRIS 3282 Processor Format Diagram**

<table>
<thead>
<tr>
<th>31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1</th>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB COUNT LSB</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>MSB COUNT LSB</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>DATA</td>
<td>0 1</td>
<td>WORD NUMBER</td>
</tr>
<tr>
<td>CHECK SUM</td>
<td>0 1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>IF WORD NUMBER X RECEIVED</td>
</tr>
</tbody>
</table>

**Figure No. 7 : ARINC 718 Words Format**

*Test Data Link Processor*
1.6. **Low Level Versus High Level Dialogue Process**

The low level program (TXID.ASM) dialogue with the high level program (TDLP.PAS) in two ways: common share variables or by interrupt process.

The 'FIRMWARE' offset listing (TXID.LST) gives the following addresses:

<table>
<thead>
<tr>
<th>SEGMENT</th>
<th>OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENDFLAG</td>
<td>$xxxx</td>
</tr>
<tr>
<td>WRDBUF</td>
<td>$xxxx</td>
</tr>
<tr>
<td>WCNT</td>
<td>$xxxx</td>
</tr>
<tr>
<td>FILE</td>
<td>$xxxx</td>
</tr>
<tr>
<td>RECFLAG</td>
<td>$xxxx</td>
</tr>
<tr>
<td>CBUFSTR</td>
<td>$xxxx</td>
</tr>
<tr>
<td>RWCNT</td>
<td>$xxxx</td>
</tr>
<tr>
<td>RFILE</td>
<td>$xxxx</td>
</tr>
<tr>
<td>VECTOR</td>
<td>$xxxx</td>
</tr>
<tr>
<td>VFLAG</td>
<td>$xxxx</td>
</tr>
<tr>
<td>FLINT</td>
<td>$xxxx</td>
</tr>
</tbody>
</table>

Strapping parameters:

- Segment address = $C000
- Interrupt request level = IRQ3

I/O Addresses:

- HALT AAC PROCESSOR = $301
- RESTART AAC PROCESSOR = $300
- RESET AAC INTERRUPT = $302

These parameters can be changed by a proper strapping on the AAC. The high level program makes use of the same addresses to get or to introduce data and control commands.

*Test Data Link Processor*
1. BASIC MECHANISM TO RECEIVE A WORD

1.1. Main Program Condition

The initial value of RECFLAG is 0. To start the reception process, the TDLP.PAS program only needs to set it to 1 at the beginning and during the rest of the time TXID.ASM will control it.

Words can be received in two different ways. The first can be considered as a random way (RTS) and the other, an expected way (CTS or ACK). An RTS word can arrive at any moment and for that reason his process is treated at the interrupt procedure level. For CTS and ACK words a time delay can be adapted using a flag indicator (INTERRUPT FLAG) associated with a timeout routine.
1.2. Interrupt Procedure Condition

Figure No. 1: Interrupt Procedure Flowchart for a Word

Test Data Link Processor
Comments:

SAVE REGISTERS:

Common initial procedure of saving all processor registers including code segment passing value to a typed constant avoiding DOS non re-entry.

RESET INTERRUPTIONS:

PORT [$302] := 1 : Reset interruption on the AAC.
PORT [$20] := 1 : Reset interruption on the PC.

GET WORD:

Using VECTOR pointer, last word is read from CBUFSTR buffer.

INTERRUPT FLAG:

When the incoming word is a CTS or a ACK this logical flag is set.

WORD COUNTER:

RTS/CTS data field (Bits 27 to 31) ; Meaning the number of words to receive or to send.

After a RTS/CTS dialogue, RWCNT is loaded with the value of WORD COUNTER register which informs the AAC about the number of words to receive.

SEND_CTS:

Output process of a CTS word. The transponder, if no other activity is present on the bus, scans the presence of the TDLP, by sending in a cyclic way (3 sec) a RTS of 0 words which receives an immediate response from the TDLP, a CTS of 0 words. This procedure is called "BUS TEST".

RESTORE REGISTERS:

Common final procedure of restoring all processor registers.

2. BASIC MECHANISM TO RECEIVE A RECORD

2.1. Main Program Condition

In fact, the only prior condition is RECFLAG :=1 which is responsible for the chaining actions mention in Annex 1.
In other hand, when a record has arrived there is an increment of a stack message register which provokes record processing when the register is checked. The stack register associated with a circular multi-record buffer prevents any loss of information caused by record overload.

2.2. Interrupt Procedure Condition

![Flowchart](image)

Figure No. 2: Interrupt Procedure Flowchart for a Record

*Test Data Link Processor*
Comments:

When the PC receives an interrupt corresponding to a file reception indication (FLINT = 2), the interrupt procedure executes the following actions:

FLINT := 0:
FLINT flag is reset

INC (STACK_MESSAGES, 1):
STACK_MESSAGES register is incremented with 1.

SEND_ACK (RWCNT):
The dialogue protocol is completed with the output of a ACK word which confirm having received RWCNT words.

MESSAGE TRANSFER:
The incoming message is loaded on a multi file buffer.

If the condition (FLINT=3) appears, it means that a control word arrived just before or after a file transmission and the PC has missed the interrupt which cause an ERROR condition (restart process).
2.3. Uplink Record Processing

Figure No. 3

Test Data Link Processor
Comments:

FINAL TRANSFER:

The file is transferred from the circular buffer to a single buffer where the data will be analysed depending the UMT field.

SURVEILLANCE INTERROGATION: FORMAT process

COMM-A INTERROGATION: FORMAT process

COMM-A BROADCAST INTERROGATION: FORMAT process

COMM-C ELM: FORMAT process

SCREEN MANAGEMENT:

If the message is completed, a video buffer is loaded with it. Using screen addressing facilities, this buffer is transferred to a circular video buffer (Screen) displaying the message.

Test Data Link Processor
Example of a COMM-A Message

```
3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1

<table>
<thead>
<tr>
<th>P</th>
<th>L</th>
<th>CRN</th>
<th>NOT DEFINED</th>
<th>UII</th>
<th>RCN</th>
<th>UMT</th>
<th>UDT</th>
<th>CT</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>MA2</td>
<td>MA1</td>
<td></td>
<td>SD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>MA5</td>
<td>MA4</td>
<td></td>
<td>MA3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>NOT USED</td>
<td>MA7</td>
<td></td>
<td>MA6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>CHECKSUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
```
Example of a COMM-C ELM Message of 6 Segments

Figure No. 5 : Line Aspect Diagram of a COMM-C Message of 6 Segments

Note : All fields have MSB at right except COUNT. For the end user, bits are swapped according with Figure No. 6 of Annex 1. X = 0 or 1.
3. BASIC MECHANISM TO SEND A WORD

3.1. Main Program Condition

![Diagram]

Figure No. 6

*Note:* This mechanism is used at different stages including downlink and uplink processes.

**Comments:**

SENDFLAG = 0:

The AAC is ready to start the process

LOAD WRDBUF:

WRDBUF is loaded as mention in Figure No. 6.

CLEAR_BUFER_FLAG:

Flag that indicates if RFILE buffer should be cleaned to receive a file. On that case SENDFLAG := 3 which is a typical situation for a CTS word.

*Test Data Link Processor*
4. **BASIC MECHANISM TO SEND A RECORD**

4.1. **Main Program Condition**

![Diagram](image)

**Figure No. 7**
Comments:

POLLING_KEYBOARD:

This procedure scans directly from PC-Address $60 the code of the last keyboard pressed button. When a key is pressed INKEY flag gets the logical value of TRUE.

MESSAGE LIBRARY(Keycode):

When starting TDLP.PAS, 18 messages (14 pre-prepared and 4 user-prepared) are formatted and loaded on a downlink message buffer. Each message is allocated to a certain keycode. To avoid any time lost, when a button is pressed the corresponding message is directly loaded on the FILE buffer. With this, real time reply "criterium" is accomplished.

Example of a COMM-B Message

<table>
<thead>
<tr>
<th>P</th>
<th>L</th>
<th>DRN</th>
<th>BOS</th>
<th>DII</th>
<th>RCN</th>
<th>DMT</th>
<th>DDT</th>
<th>CT</th>
<th>COUNT</th>
</tr>
</thead>
</table>
| X | 1 | 0 | 0 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | WORD 1
| P | MB3 | | MB2 | | MB1 | | | | | | | | | | | | | | | COUNT |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 0 | 1 | 0 | 0 | 0 | 1 | 0 | WORD 2
| P | | MB6 | | MB5 | | MB4 | | | | | | | | | | | | | | | COUNT |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 1 | 0 | 0 | 0 | 1 | 1 | WORD 3
| P | | NOTUSED | | MB7 | | | | | | | | | | | | | | | | | COUNT |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 1 | 0 | 0 | 1 | 0 | 0 | WORD 4
| P | | CHECKSUM | | | | | | | | | | | | | | | | | | | COUNT |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | 0 | 1 | 1 | 1 | 1 | 1 | WORD 5

Figure No. 8: ARINC Line Aspect Diagram of a COMM-B Message

Note: All fields have MSB at right except COUNT. X = 0 or 1. For the end user, bit are swapped according with Figure No. 6 of Annex 1.
Example of a COMM-D ELM Message of 6 Segments

![Message Structure](image)

**Figure No. 9 : Line Aspect Diagram of a COMM-D Message of 6 Segments**

**Note :**

DF - Downlink format

ND - Segment number

All fields have MSB at right except COUNT. For the end user, bit are swapped according with *Figure No. 6 of Annex 1*. $X = 0$ or 1.

*Test Data Link Processor*
1. BASIC MECHANISM TO SEND A GICB

Figure No. 1

Test Data Link Processor
Comments:

POLLING KEYBOARD: described in Annex 2.

CHECK_GICB_TIMEOUT:

This procedure detects a relative time out (approx. 3 sec. with only BUS TEST dialogue) and put GICB_TIMEOUT flag at logical value TRUE.

SEND GICB's:

When starting TDLP.PAS, and after reading GICB file, 24 GICB messages are formatted and loaded on a GICB message buffer. This procedure reads each message and perform a complete transmission process.
1. BACKUP MESSAGES FORMAT PROCESS

```
READ_BACKUP_FILES

GET_MESSAGE_LENGTH

BITS_MESSAGE_LENGTH < 57

BITS_MESSAGE_LENGTH < 225

BITS_MESSAGE_LENGTH < 1281

MESSAGE_OVERFLOW

LOAD_DOWNLINK_BUFFER

YES

UNLINKED A.I.C.B.

YES

LINKED A.I.C.B.

YES

COMM-D ELM

Test Data Link Processor
```
Note:

This process is part of the initial setup proceedings of TDLP real time application.

Comments:

These messages are read from 4 different files (BACK1, BACK2, BACK3, BACK4). The process of formatting and loading is the same that for the 14 programmed messages.

READ_BACKUP_FILES:

Reading procedure from disc of the 4 backup files.

GET_MESSAGE_LENGTH:

Get the number of characters on each message.

BITS_MESSAGE_LENGTH:

Considering 6 bits character coding, this variable contains the number bit to format.

UNLINKED AIR INITIATED COMM-B:

Format process.

LINKED AIR INITIATED COMM-B:

Format process.

COMM-D EXTENDED LENGTH MESSAGE:

Format process.

MESSAGE OVERFLOW:

Linked ELM messages are not implemented.

LOAD_DOWNLINK_BUFFER:

This buffer receives all formatted messages and ready to be sent.
2. ON LINE MESSAGE

This single message is read from a different user file, is formatted and loaded as the 19th message and its key allocation is any other free key from keyboard (Chapter 4.3.2.2.).
ANNEX No. 5

TDLP DOWNLINK MESSAGES DISPLAY

Test Data Link Processor
COMB
COMMB LINKED 2
TEST COMM-B LINKED 3
TEST COMM-B LINKED 4

.............TEST ELM COMM-D MESSAGE WITH 6 SEGMENTS.............

.............TEST ELM COMM-D MESSAGE WITH 10 SEGMENTS.............

....TEST ELM COMM-D MESSAGE WITH 10 SEGMENTS..

.............THIS IS A DOWNLINK MESSAGE FOR TESTING A ELM WITH 16 SEGMENTS WITH APM = 1.............THIS IS A DOWNLINK MESSAGE FOR TESTING A ELM WITH 16 SEGMENTS WITH APM = 1..

AFR485 REQUEST FL345 AT NANTES WAYPOINT
SAB546 REQUEST DIRECT ROUTING ON FL240 AT PARIS/ORLY
TAP426 REQUEST METEO REPORT FOR EDDF
LUF345 REQUEST ATIS INFORMATION FOR LPPO

Photo n° 1

BRI345 REPORTING CB TURBOLENCE AT FL300
ABC345 REPORTING UFO AT BRETIGNY REGION
AFR345 REPORTING ALL FREQUENCIES OFF LINE INCLUDING EMERGENCY 119.1

GOOD MORNING AT THIS MUNS
HELLO AT THIS MUNS.
MESSAGE RECEIVED.

WELCOME TO THE TEST DATA LINK PROCESSOR ENVIRONMENT.

COMB
COMMB LINKED 2
TEST COMM-B LINKED 3
TEST COMM-B LINKED 4

.............TEST ELM COMM-D MESSAGE WITH 6 SEGMENTS.............
GOOD MORNING AF320

CommA Unlinked 5 bits coded Message

CommA Unlinked 5 bits coded Message

THERE

CommA phrase coded ATC message number = 1

..........................TEST ELM COMM-D MESSAGE WITH 6 SEGMENTS..........................

CommA phrase coded ATC message number = 11
CommA phrase coded ATC message number = 28

THY345 REPORTING UFO AT BRETIGNY REGION

CommA phrase coded MET message number = 2
CommA phrase coded MET message number = 3

AFR345 REPORTING ALL FREQUENCIES OFF LINE INCLUDING EMERGENCY 119.1