

ARL-SYS-TM-139

AR-006-079

AD-A236 028



DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

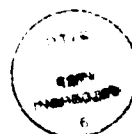
MELBOURNE, VICTORIA

Aircraft Systems Technical Memorandum 139

PROGRAMMABLE COCKPIT - INTER-COMPUTER

COMMUNICATIONS AND DATA FLOW

by



D.A. CRAVEN

Submission for	
FILE NAME	☒
DATE	☒
REVIEW	☒
APPROVAL	☒
By	
Distribution	
Availability	
AVAL	
A-1	

Approved for public release

(C) COMMONWEALTH OF AUSTRALIA 1990

91-01075



MARCH 1990

This work is copyright. Apart from any fair dealing for the purpose of study, research, criticism or review, as permitted under the Copyright Act, no part may be reproduced by any process without written permission. Copyright is the responsibility of the Director Publishing and Marketing, AGPS. Inquiries should be directed to the Manager, AGPS Press, Australian Government Publishing Service, GPO Box 84, CANBERRA ACT 2601.

**DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY**

Aircraft Systems Technical Memorandum 139

**PROGRAMMABLE COCKPIT - INTER-COMPUTER
COMMUNICATIONS AND DATA FLOW**

by

D.A. CRAVEN

SUMMARY

The Programmable Cockpit is a multi-computer system which is used for research into display design, cockpit layouts, crew workload and other human factors issues. This document covers the communication strategy (how information is conveyed between constituent units of the system) employed in Stage 1 of the project.



(C) COMMONWEALTH OF AUSTRALIA 1990

**POSTAL ADDRESS: Director, Aeronautical Research Laboratory
506 Lorimer Street, Fishermens Bend Victoria 3207
Australia**

CONTENTS		Page No.
1.	INTRODUCTION	1
2.	SYSTEM ARCHITECTURE	1
3.	COMMUNICATION CHANNELS	2
3.1	Dual Port RAM	2
3.2	Parallel I/O	2
3.3	Serial I/O	2
4.	SOFTWARE	2
4.1	Background Tasks	2
4.2	Interface to Amiga Units	3
4.3	Interface to AT units	3
4.4	The Communication Rendezvous	4
5.	MODIFYING THE DATA STRUCTURES	5
6.	CONCLUDING REMARKS	5
FIGURE 1 :	Communication Path	6
FIGURE 2 :	Flow Chart - PCPTx Process (Amiga)	7
FIGURE 3 :	Flow Chart - PCPRx Process (Amiga)	8
FIGURE 4 :	Flow Chart - Init_Comms Procedure (AT)	9
FIGURE 5 :	Flow Chart - Write_Data Procedure (AT)	9
FIGURE 6 :	Flow Chart - Read_Data Procedure (AT)	9
APPENDIX	- Listing of AircraftData.def	10
REFERENCES		12
ACKNOWLEDGEMENTS		13

1. INTRODUCTION

The Programmable Cockpit is a multi-computer system which is used for research into display design, cockpit layouts, crew workload and other human factors issues. It currently consists of four closely arranged screens onto which different cockpit instruments can be displayed, all driven from a common aircraft model. This provides a test-bed for determining the effectiveness of new or modified instruments, as they can be assessed in an environment simulating the cockpit as a whole.

The system incorporates a distributed processing network of five processors, each performing a distinctive part of the simulation. The tasks are broken up as follows:

- **FDM (Flight Dynamic Model)** : The FDM controls the aircraft simulation, receiving pilot input (via the control column, throttles, etc.) and providing the other processors with the appropriate data to generate their displays. Input to this unit is also required from the Control Display. The FDM uses an 80286 based IBM PC-AT compatible computer (with co-processor). (Ref 1)
- **CD (Control Display)** : The CD utilises a touch screen in order to simulate switches, navigational computers, etc. that are available to the pilot. Engine instruments also appear on this screen. A 68020 based Amiga (with co-processor) is used for this unit. (Ref 2)
- **HUD (Head Up Display)** : The HUD provides an outside view over which an overlay is drawn to simulate a head-up display. A 68020 based Amiga (with co-processor) is used for this unit. (Ref 3)
- **HDD (Head Down Display)** : The HDD shows the main bank of aircraft instruments. The HDD uses an 80286 based IBM PC-AT compatible computer (with co-processor). (Ref 4)
- **MMD (Moving Map Display)** : The MMD provides a north - up moving map display intended as a navigator's aid. Position, waypoint (as set from the navigational computer in the CD) and track information are superimposed over the map. A 68000 based Amiga was used for this display (Ref 5)

In order to minimise the delay involved in distributing the relevant information to all of the displays, an efficient communications system is required.

2. SYSTEM ARCHITECTURE

Two Amiga 2500s (a "hybrid" computer containing both MC68020/68881 and Intel 80286/80287 processor combinations) and one Amiga 500 are used in the system. The 2500s possess a small area of dual-port RAM (available to both processors) which is able to be used for communication. The setup chosen is depicted in Figure 1.

Information required by the various displays has been divided into two separate structures : the Engine Data structure, which contains engine information only relevant to the CD; and the Aircraft Data structure, containing entries used by all units. The Engine Data structure is therefore kept local to the Amiga hosting the FDM and CD and is not transmitted to the other computers. The appendix shows the Modula-2 source defining the current versions of these structures.

All of the display units operate independently on information originating from the FDM. The CD, however, also has a need to pass information back to the FDM (eg position of Gear Up/Down switch). In order to keep the external communication path as simple as

possible, the FDM and CD both reside on the same Amiga 2500 host, allowing different areas of the dual port RAM to be used for communications in both directions. This enables the link between the two A2500s to be unidirectional, eliminating the potential problem of delays incurred in waiting for the channel to become free.

Serial communication (via an RS232C interface) is used to pass data to the MMD. This display is not as dynamic as the others, and so does not suffer from the use of a slower channel.

3. COMMUNICATION CHANNELS

3.1 Dual Port RAM

The Amiga 2500's "bridgeboard" (Ref. 6) includes 128 kilobytes of dual port memory for the specific purpose of allowing efficient communication between the two constituent computers. The two processors accessing this memory have inherently different expectations on how information is stored for data elements exceeding one byte in length. The Motorola processor (of the Amiga) uses the convention that the most significant byte occupies the lowest address, whereas the Intel processor (AT side) expects the least significant byte at this address. To help compensate for this, the DPR is connected to logic performing a "byte-swapping" operation, so that if a 16-bit word is accessed by the Motorola processor, the high and low bytes will be transposed automatically. However, when accessing 32-bit data items, user software must still be used to exchange the positions of the high and low word. The Motorola 68020's "SWAP Dn" instruction efficiently handles this operation in as little as one processor clock cycle.

3.2 Parallel I/O

The Amiga has standard bi-directional parallel ports that provide an efficient means for high speed data transfer. The device driver for the parallel port allows a task to "wait" for a specific number of bytes to be received at the port. While waiting, the task is removed from the system's "ready" list, meaning it will receive no processor time (Ref. 7). The driver "signals" the process when the requested data has been received, causing the process to again be placed on the ready list and resume execution. As a result, the communication tasks on either side of the parallel link are absent from the ready list for most of the time, minimising the overheads involved in communication.

3.3 Serial I/O

Existing RS232C serial ports are used, operating at 9600 baud. The serial device driver operates on the same principles as that of the parallel device (above).

4.0 SOFTWARE

Software has been written to implement the abovementioned methodology and to provide an interface for the various languages used in the Programmable Cockpit.

4.1 Background Tasks

Distribution of the data structures is handled by two background tasks, one on each Amiga 2500. Since all displays need access to this information when updating, it is critical that this distribution be carried out quickly. As such, both of these tasks are given a higher priority than the HUD and CD units they share their processor with. A description of the tasks follows :

• **PCPTx (Programmable Cockpit Transmitter - Fig. 2)** This process (executed on the same Amiga as the CD) waits for the FDM to signal that it has initialised the Dual Port Memory and then obtains the relevant information from the AT regarding its use. It then allocates local memory for the CD to use for its data structures, creating a RAM file (used later by the CD) containing pointers to this area. The process then executes a loop where it repeatedly waits for a signal from the AT to indicate when new data is to be distributed.

• **PCPRx (Programmable Cockpit Receiver - Fig. 3)** This process, residing on the Amiga hosting the HUD, is similar to PCPTx except that it waits for a signal from the parallel device indicating that a new data packet (sent by PCPTx) has been received. After checking this data for errors, it is distributed to the HUD, HDD and, via the serial port, to the MMD.

4.2 Interface to Amiga Units

The communication interface to the Amiga units has been written in Modula-2 with embedded 68020 assembler. The defined variables/procedures are as follows :

```
VAR   FSDDataPtr      : POINTER TO AircraftData.DataStructure;  
      EngineDataPtr   : POINTER TO AircraftData.EngineDataStructure;
```

These variables point to the areas of memory required to be used for the Amiga's local copy of the AircraftData and EngineData structures (see appendix) respectively. The latter is only used by the CD.

PROCEDURE Init;

This procedure initialises the above pointers. In the case of the CD and HDD, if the unit on the AT side of the Dual Port RAM channel is not yet executing, Init will wait until it is signalled by the AT. For the MMD, Init configures the serial channel and returns immediately.

PROCEDURE Finished () : BOOLEAN;

This procedure returns TRUE if the AT program on the other side of the Dual Port RAM "bridge" has stopped executing. (Not valid for the MMD)

PROCEDURE Preset (number : CARDINAL);

Preset is used by the CD and passes the argument "number" to the FDM. This is used to force the FDM to one of a number of pre-defined flight conditions when required.

PROCEDURE Request2 (number : INTEGER);

Request2 is another procedure to pass a data item from the Amiga to the AT via the Dual Port RAM. It is used by the CD to pass runway height information to the FDM.

4.3 Interface to AT Units

The communication interface for the AT based units has been written in 80286 assembler, and is able to be called from both Microsoft C and Pascal (as used in the HDD and FDM respectively). The software packages **FDMComms.asm** and **HDDComms.asm** contain identical procedure specifications and differ only in their usage of the Dual Port Memory (since the FDM has the additional Engine Data structure). The Pascal definition

of these procedures, as well as a brief description of their operation, follows :

FUNCTION Init_Comms () : INTEGER : EXTERNAL;

Init_Comms initialises the Dual Port RAM communication channel and allocates memory for the data structures. If no problems are encountered, the Amiga is signalled (via an MS-DOS interrupt) to indicate that the link has been made and the function returns a zero (successful) error code. (Flow chart - Fig. 4)

PROCEDURE Close_Comms : EXTERNAL;

Close_Comms removes the communication link between the Amiga and AT and frees all associated resources.

PROCEDURE Write_Data (VARs data : DataStructure) : EXTERNAL;

Write_Data copies the AT's local data structures into the Dual Port RAM area, making them accessible to the Amiga. This procedure is only used by the FDM. (Flow chart - Fig. 5)

PROCEDURE Read_Data (VARs data : DataStructure) : EXTERNAL;

Read_Data copies the Aircraft Data structure from the DPR into the AT's local memory area. This procedure is only used by the HDD. (Flow chart - Fig. 6)

FUNCTION Preset_Req () : INTEGER : EXTERNAL;

This procedure returns a request number (as set by the Amiga) or zero if none is pending. The request is cleared before the procedure returns. Preset_Req is currently only used by the FDM.

FUNCTION Request_2 () : INTEGER : EXTERNAL;

Request_2 operates in the same way as Preset_Request with the exception that the request is not cleared. the FDM uses this function to obtain runway height information from the CD.

4.4 The Communication Rendezvous

The communication process is initiated from the FDM. When new data has been calculated (i.e. at the end of an iteration of the model) a call to Write_Data updates the structures in the Dual Port RAM and signals the PCPTx process on the Amiga. This process then copies (and translates where necessary) the appropriate structures into the Amiga's local data area, and requests the parallel port to transmit the Aircraft Data. The other Amiga (controlling the HUD) is in turn interrupted by its parallel port indicating a full packet has been received. After checking for errors, the new data is copied into the HUD's local memory area as well as the Dual Port RAM, and finally queued to the serial port for the MMD.

5. MODIFYING THE DATA STRUCTURES

The software has been designed to facilitate modifications to the data structures so that new entities may be added/deleted as required. Those packages requiring modifications to implement changes in the Data Structure are :

- *AircraftData.def* : The DataStructure is changed to represent the new data;
- *FSTx* : The constant FDMScopeSize must be changed to reflect the size (in bytes) of the data structure valid for the FDM;
- *FSRx* : The constant HDJanusSize must be similarly modified to reflect the scope of the HDD;
- *FDMComms.asm/HDDComms.asm* : The DPRSize constants need to be changed to allow sufficient room in the Dual Port RAM for the new structures.

The source code for all of the above packages is commented with details on making these changes.

Simple version checking has been implemented to aid in detecting conflicting data structures.

6. CONCLUDING REMARKS

The communication system as used in stage 1 of the programmable cockpit operated reliably and introduced little overhead to the system.

FIGURE 1 : Communication Path

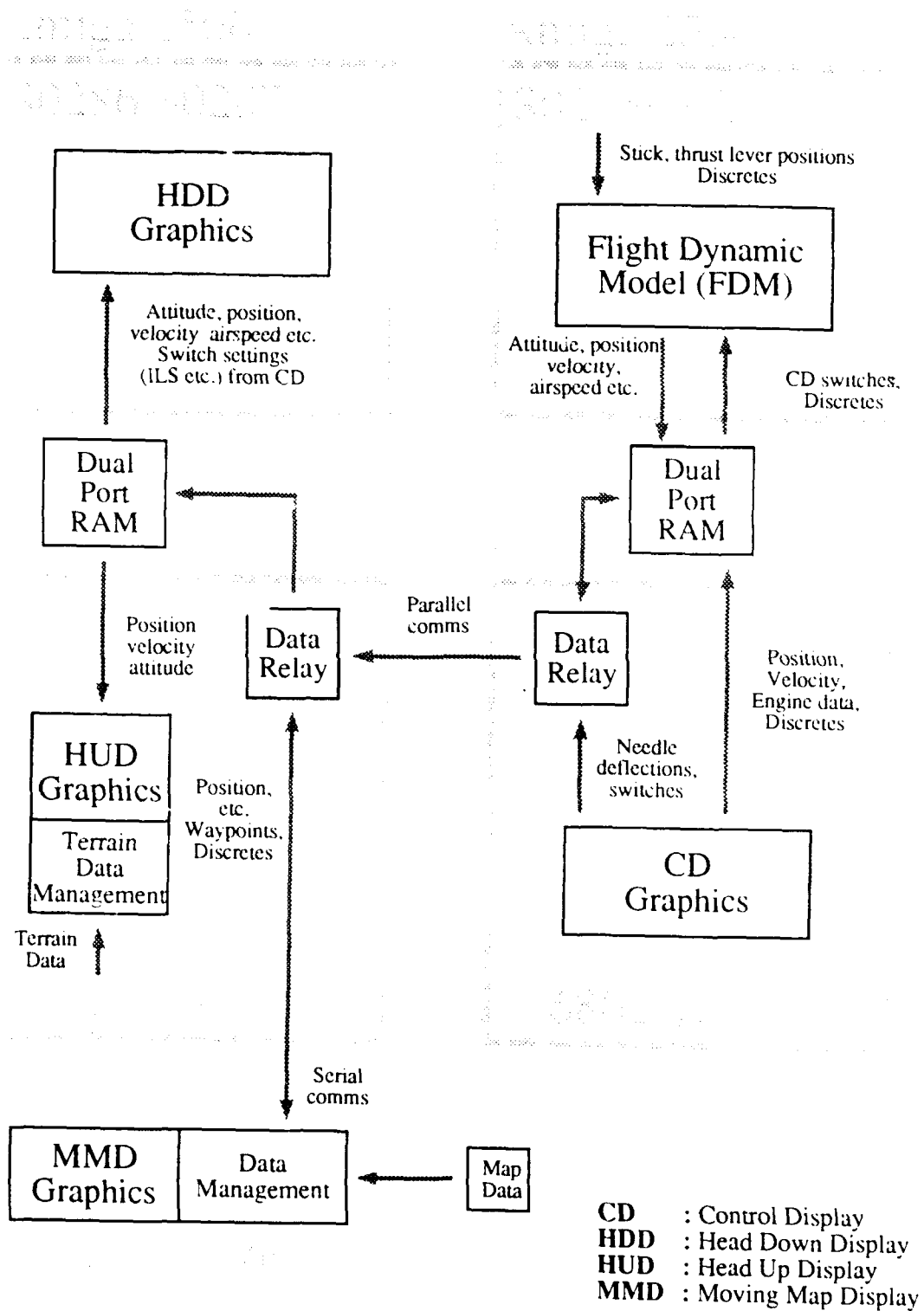


FIGURE 2 : FLOW CHART - PCPTx Process
(Background process on Amiga hosting CD)

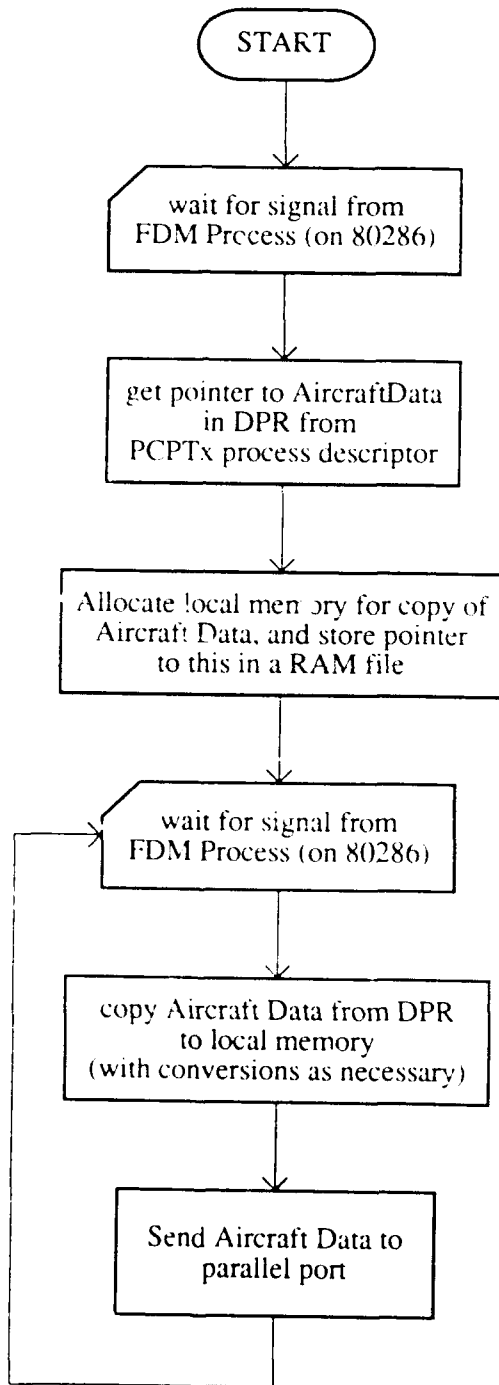
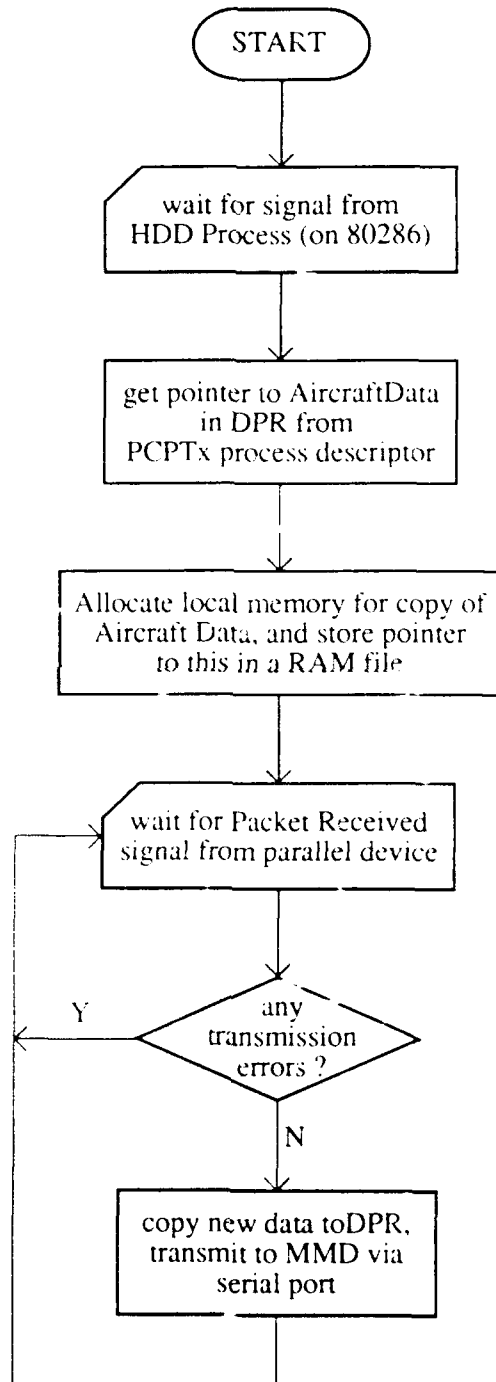
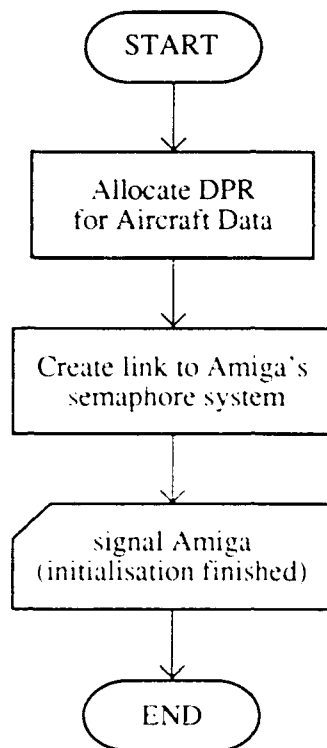


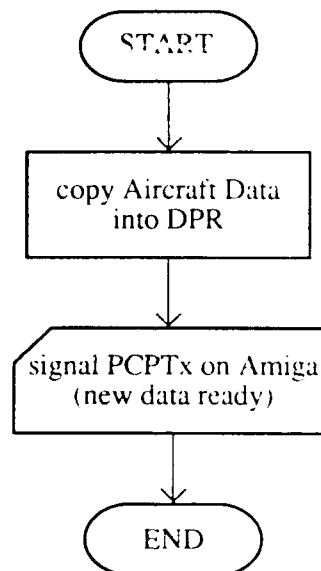
FIGURE 3 : FLOW CHART - PCPRx Process
(Background process on Amiga hosting HUD)



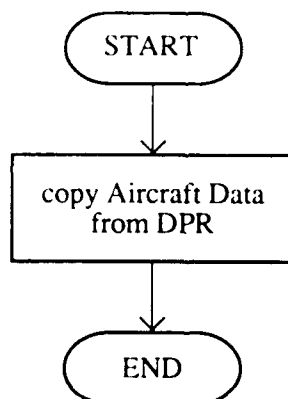
**FIGURE 4 : Init_Comms
Flow Chart (AT Procedure)**



**FIGURE 5 : Write_Data
Flow Chart (AT Procedure)**



**FIGURE 6 : Read_Data
Flow Chart (AT Procedure)**



APPENDIX : Listing of AircraftData.def

DEFINITION MODULE AircraftData;

FROM SYSTEM IMPORT BYTE, TSIZE;

CONST NumLongWords = 4 (* number of LONGINTs at start of AEM's scope *)

(* these need to have their high & low words
swapped to allow for different processor
conventions *);

TYPE WarningType = (w0, w1, w2, w3, w4, w5, w6, w7,
gearDown, stall, w10, w11, w12, w13, w14, buttonDown);

WarningSet = SET OF WarningType; (* size 16 bits *)

DataStream = RECORD

(*	SCOPE	*)	(* UNITS *)
(*	CD MMD HUD	*) FollWPN : LONGINT;	(* decimeters *)
(*		*) FollWPE : LONGINT;	(* decimeters *)
(*		*) ToWPN : LONGINT;	(* decimeters *)
(*		*) ToWPE : LONGINT;	(* decimeters *)
(*		*) FromWPN : LONGINT;	(* decimeters *)
(*		*) FromWPE : LONGINT;	(* decimeters *)
(*		*) NameFoll : ARRAY [0..5] OF CHAR;	
(*		*) NameFrom : ARRAY [0..5] OF CHAR;	
(*		*) NameTo : ARRAY [0..5] OF CHAR;	
(*	HDD	*) AdvancedDisplay : BOOLEAN;	

(* space for another boolean or byte here -
all non-byte elements are word aligned *)

(*	AEM	*) NPosition : LONGINT;	(* decimeters *)
(*		*) EPosition : LONGINT;	(* decimeters *)
(*		*) Altitude : LONGINT;	(* decimeters *)
(*		*) AltFeet : LONGINT;	(* feet *)

(* add further LONGWORD items here
and increase NumLongWords (CONST) accordingly*)

(*		*) CursorX : INTEGER;	
(*		*) CursorY : INTEGER;	
(*		*) Warnings : WarningSet;	
(*		*) Heading : INTEGER;	(* 1/10th deg *)

(*		*) AircraftType : INTEGER;	
(*	MMD	*) DriftAngle : INTEGER;	(* 1/10th deg *)
(*		*) FPDepressionAngle : INTEGER;	(* 1/10th deg *)
(*		*) LateralAcc : INTEGER;	(* 1/10th ft/s *)
(*		*) AngleOfAttack : INTEGER;	(* 1/10th deg *)

(*					*) MachNumber	: INTEGER;	(* 1/10th deg *)
(*					*) GNumber	: INTEGER;	(* 1/10th deg *)
(*					*) Pitch	: INTEGER;	(* 1/10th deg *)
(*					*) Roll	: INTEGER;	(* 1/10th deg *)
(*					*) Speed	: INTEGER;	(* knots *)
(*					*) RateOfTurn	: INTEGER;	(* /second *)
(*					*) RateOfClimb	: INTEGER;	(* feet/min *)
(*		AEM			*) Power	: INTEGER;	(* % *)
(*					*) HeadingBug	: INTEGER;	(* degrees *)
(*					*) AltitudeBug	: INTEGER;	(* 100s ft *)
(*					*) SpeedBug	: INTEGER;	(* knots *)
(*					*) Flaps	: BYTE;	(* degrees *)
(*					*) SpeedBrake	: BYTE;	(* degrees *)
(*					*) ILSFlags	: BITSET;	
(*					*) ILSx	: INTEGER;	(* minutes *)
(*					*) ILSy	: INTEGER;	(* minutes *)
(*					*) ADFHeading	: INTEGER;	(* degrees *)
(*					*) ADFReqCourse	: INTEGER;	(* degrees *)
(*					*) ToWPElevation	: INTEGER;	(* feet *)

(* DME made last in record as it is LONG *)

(* CD HDD HUD *) DME :LONGINT; (* decimeters *)

END;

EngineDataStructure = RECORD

EPRL	: INTEGER;
EPRR	: INTEGER;
EGTL	: INTEGER;
EGTR	: INTEGER;
N1L	: INTEGER;
N1R	: INTEGER;

END;

CONST num = TSIZE (DataStructure);

END AircraftData.

REFERENCES:

1. Flight Management Group. *Programmable Cockpit - Stage 1 Overview*
ARL-SYS-TM-136 1990
2. Iob, M. - *Programmable Cockpit - Flight Dynamic Model*
ARL-SYS-TM-138 1990
3. Page, A.G. - *Programmable Cockpit - Head Up Display and Outside View*
ARL-SYS-TM-137 1990
4. Selvestrel, M.C. - *Programmable Cockpit - Head Down Display*
ARL-SYS-TM-140 1990
5. Craven, D.A. - *A Demonstration of an Aircraft Navigator's Moving Map Display*
ARL-SYS-TM-141 1991
6. Mical, R.J. - *Amiga/PC Services: Bridge Board* Commodore- Amiga Inc.
1988
7. Commodore Business Machines, Inc. - *Amiga ROM Kernel Reference Manual*
-Exec Addison - Wesley Publishing Company, Inc. 1986

ACKNOWLEDGEMENTS

Acknowledgement of the efforts of the members of the Programmable Cockpit team is due :

- I.V. Lloyd** - Project supervisor
- M. Iob** - Flight Dynamic Model
- B.D.L. Neil** - Control Display
- A.G. Page** - Head Up Display / Outside View graphics
- M. Selvestrel** - Head Down Display
- P.W. Futschik** - Hardware interfacing to Control column and throttles

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central

Chief Defence Scientist
AS, Science Corporate Management (shared copy)
FAS Science Policy (shared copy)
Director, Departmental Publications
Counsellor, Defence Science, London (Doc Data Sheet Only)
Counsellor, Defence Science, Washington (Doc Data Sheet Only)
OIC TRS, Defence Central Library
Document Exchange Centre, DSTIC (8 copies)
Defence Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne

Aeronautical Research Laboratory

Director
Library
Chief Aircraft Systems Division
Author: D.A. Craven (3 copies)
PO/Flight Management Group
I. Lloyd
B. Neil
M. Iob
A. Page
M. Selvestrel
P. Futschik

Materials Research Laboratory

Director/Library

Defence Science & Technology Organisation - Salisbury

Library

Navy Office

Navy Scientific Adviser (3 copies Doc Data Sheet)

Army Office

Scientific Adviser - Army (Doc Data Sheet Only)

Air Force Office

Air Force Scientific Adviser (Doc Data Sheet Only)
Aircraft Research and Development Unit
Library
Director Research Requirements - Air Force

HQ ADF

Director General Force Development (Air)

Universities and Colleges

Adelaide
Barr Smith Library

Flinders
Library

LaTrobe
Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library
Professor R. Telfer, Institute of Aviation

New England
Library

Sydney
Engineering Library

NSW
Library, Australian Defence Force Academy

Queensland
Library

Tasmania
Engineering Library

Western Australia
Library

RMIT
Library
Mr M.L. Scott, Aerospace Engineering

University College of the Northern Territory
Library

SPARES (10 COPIES)

TOTAL (57 COPIES)

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER AR-006-079	1b. ESTABLISHMENT NUMBER ARL-SYS-TM-139	2. DOCUMENT DATE MARCH 1990	3. TASK NUMBER DST 89/096
4. TITLE Programmable Cockpit - Inter-Computer Communications and Data Flow		5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF. (C) RESTRICTED (R), UNCLASSIFIED (U)). U U U DOCUMENT TITLE ABSTRACT	6. NO. PAGES 15 7. NO. REFS. 7
8. AUTHOR(S) D.A. Craven		9. DOWNGRADING/DELIMITING INSTRUCTIONS Not applicable	
10. CORPORATE AUTHOR AND ADDRESS AERONAUTICAL RESEARCH LABORATORY 506 LORIMER STREET FISHERMENS BEND VIC 3207		11. OFFICE/POSITION RESPONSIBLE FOR: SPONSOR <u>DSTO</u> SECURITY <u>-</u> DOWNGRADING <u>-</u> APPROVAL <u>CSYD</u>	
12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for public release OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601			
13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO No limitations			
13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE ☒ UNRESTRICTED OR ☐ AS FOR 13a.			
14. DESCRIPTORS Cockpits Flight crews Human factors engineering Computer interfaces			15. DISCAT SUBJECT CATEGORIES 0104
16. ABSTRACT <i>The Programmable cockpit is a multi-computer system which is used for research into display design, cockpit layouts, crew workload and other human factors issues. This document covers the communication strategy (how information is conveyed between constituent units of the system) employed in stage 1 of the project.</i>			

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT)

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER

Aircraft Systems Technical
Memorandum 139

19. COST CODE

74 7312

20. TYPE OF REPORT AND PERIOD COVERED

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)