University of the Witwatersrand

A HIGH SPEED RADAR DATA ACQUISITION AND PROCESSING SYSTEM FOR AN EXPERIMENTAL MONOPULSE TRACKING RADAR

By

Norman Keith Burgess

A project report submitted to the Faculty of Engineering,
University of the Witwatersrand,
in partial fulfilment of the requirements for the degree of
Master of Science in Engineering

Johannesburg, April 1994

I hereby declare that all the material incorporated into this project report is my own original un-aided work, except where specific reference is made to other contributors. The work contained herein has not been submitted for a degree at any other university.

Signed ____________________ Date 11 APRIL 1994

N K Burgess

ABSTRACT

This project report describes the development of a high speed data acquisition and processing system (DAPS) for an experimental monopulse tracking radar at the Division of Manufacturing and Aeronautical Systems Technology (Aerotek), of CSIR.

The system development involved replacement of existing data acquisition hardware and software with an enhanced, PC based, integrated high speed data acquisition system. The system is used for radar acceptance testing, radar monitoring during trials and capture of large volumes of data for off-line processing to pursue research into the phenomenology of radar cross section.

The work performed during the execution of this project represents an effort in three technology areas. These are:

- Digital hardware design of custom interface hardware for the ISA bus as implemented on most PC's. This was not a major aim of the project, but was required in the absence of suitable hardware.
- Formal software specification and design using the real time extensions of Hatley and Pirbhai to the Yourdon and de Marco data flow analysis methodology. The existing software had not been designed using any formal method, and suffered a lack of documentation throughout all phases of the development. The way this project was undertaken has served as an exercise in following the approach, which resulted in a product which is better documented and more maintainable. There are elements of total quality management that are still missing though.
- Quality management during software development. At the time of commencement of the project there was no formal commitment to software quality management at Aerotek. Commitment to ISO9000 accreditation in future has required that a solid quality ethic and management system be instituted. Current software development follows this approach. A discussion of the topic appears in the conclusion.

The DAPS has been in use for approximately two years now. With continued minor modification to meet new user requirements, it has successfully enhanced the usability of the radar system during trials and measurements with the SAAF.

ACKNOWLEDGEMENTS

The work presented in this project report was carried out under the supervision of Professor A J Walker of the Department of Electrical Engineering, University of the Witwatersrand, South Africa.

I would like to thank the following people and organisations:

Professor Walker for his guidance during the writing up of the project report.

My wife Sharon, for encouragement to finish the project and report.

Aerotek (CSIR), my employers, and my colleagues there for the opportunity to perform the work and then submit the work for the degree.

NOMENCLATURE

There are a number of definitions, acronyms and abbreviations used in this report which are specific to the radar system for which this data capture and analysis system was developed; others are more general, well defined radar system terms which require no elaboration. The reader is referred to either of [4] or [5] for clarification. In addition, a number of features existed prior to the commencement of this work, along with certain user defined concepts regarding data processing and graphical output. It is considered appropriate to provide footnote definitions of the nomenclature at the first place they occur. A glossary of terms (Appendix C) is also provided for easy reference.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1

INTRODUCTION

The Division of Aeronautical and Production Systems Technology (Aerotek) of CSIR have developed an experimental, monopulse, tracking radar. The radar system[1] is used in a number of experimental and measurement applications, each with different requirements in terms of processing and display of radar data. The applications include:

- Experimental radar cross section (RCS) measurements. Radar returns are gathered by a data capture system and stored for post processing. Amplitude and fluctuation statistics of the radar return are processed and analysed. The return of interest may be from land clutter, sea clutter, rain clutter or aircraft. During measurements of aircraft the radar system is also used to vector (direct the flight trajectories) of the aircraft being measured.

- Radar calibration and performance measurements. These serve to assist with calibrated RCS measurements as well as provide inputs for radar systems analysis and testing. Performance measurements include sensitivity, clutter rejection, noise levels and stability measurements.

- Reference tracking during SAAF (South African Airforce) exercises. The radar system is used to monitor three dimensional flight profiles along with velocity and acceleration characteristics during the flights.

- Radar acceptance testing to verify operation of the radar system prior to its use in experiments of the above types. This is a standard process where the radar is qualified and trimmed to optimum and stable performance.

[1] In all occurrences, unless otherwise indicated the terms radar or radar system refer specifically to the experimental, monopulse, tracking radar system of Aerotek of the CSIR for which the data acquisition and processing system has been developed.

The radar data stream[2] which is captured by the high speed radar data acquisition and processing system (DAPS)[3] is logically separate from the data and signal processing required to control and keep the system functioning as a radar tracking a target; although some of that data is obviously included in the radar data stream. The data captured and processed by the DAPS serves to assist the user with his use of the system, and to provide increased understanding of radar targets and radar system theory.

This project report describes the implementation of the current DAPS, which comprises PC based, high speed data acquisition hardware, associated driver software, and capture, analysis and display software. Moderately high speed acquisition, processing and display of data on machines of the PC family is by no means a new concept. There exist many D/A hardware interface cards for the PC family with associated software to capture, analyse and display data. Uses of data acquisition systems are as numerous as the number of such systems available. In this particular instance, the existing radar system has a custom data format and requires a high throughput (a moderately high average data rate with word sized samples for each radar pulse being delivered in high speed bursts), which could not be achieved with commercially available products at the time this project was initiated.

1.1 Project Overview

During early development of the radar system some ten years ago, there was already a requirement for data capture and processing. At the time this was implemented with a HP based equipment (HP9920 series computer) and a custom interface to a 9mm tape drive.

[2] This refers to the digital data stream from which radar data is captured for processing. The stream is a pre-defined set of radar parameter samples passed to the capture and processing system by the radar monopulse video processor and controller. There are 60 word or integer components representing either radar target information (such as azimuth, range or signal amplitude), or radar status indicators and set points. In the existing radar system this data set is also called **RDO**.

[3] The current version of this equipment comprises a 50MHz 486 ruggedised PC clone processor card with 16MB of RAM; 1MB VGA adapter, Super VGA display and standard I/O ports; Adaptec 1542B SCSI host adapter; 425MB, SCSI hard drive; 1GB, SCSI, Tahiti I Magneto Optical drive; combined 1.2MB/1.4MB, floppy/stiffy drive; high speed serial to parallel data filter card; FIFO capture interface card; time code reader card and a Postscript printer.

The context of the data falls into two categories, namely high speed data acquisition (**HSDA**)[4], and medium speed data acquisition (**MSDA**)[5]. HSDA facilitates capture of data for off-line, post-processing. RCS measurements are a typical example. MSDA facilitates more rapid processing, such as required for acceptance tests and calibration. The specifications of the radar data parameters, rates and data processing appears later.

HSDA was achieved with the custom interface and 9mm tape system. The tapes could be used on the DAPS or any other system. MSDA was achieved with the GPIO[6] interface on the HP computer.

As the radar moved from the experimental demonstration stage to an integrated product, it was decided that the DAPS should be upgraded. Apart from the fact that the existing system was not able to meet the full user requirement regarding data processing and output, it was also prone to failure. The tape drive was operated at the maximum data rate, and tape media imperfections frequently caused data overrun. In addition the user requirement grew with the system. PC based software development is considerably easier than HP based development, which also indicated replacement of the proprietary systems in place.

This project report described the system which was designed and implemented to replace the historic DAPS. This includes the GPIO type interface for the PC and the software to receive, store, retrieve and process data. At the time of commencement of this project (1990), there was no GPIO interface card (not to be confused with GPIB) for PC's or compatibles. This dictated that a custom interface card needed to be developed to meet the interface which existed in the DAPS.

[4] This is an abbreviation for High Speed Data Acquisition. In this report this refers to capture of all of the available radar data to a mass storage device.

[5] This is an abbreviation for Medium Speed Data Acquisition. In this report this refers to capture of a subset of the available radar data.

[6] This is the Hewlett Packard term describing their General Parallel Input Output bus and its associated standards. This bus is described completely in their documentation. The implementation of the particular interface board of this project is described in more detail elsewhere. The interface used in this project is a subset of the full GPIO standard. The subset is really just a high speed, parallel, externally clocked data bus with data on/off control.

1.2 Structure of Project Report

The presence of the existing radar data acquisition system imposed tight constraints on the options for a solution to the problem; a relatively simple decision was taken to move to a PC clone, develop a custom interface card and program in C. For this reason, there is no appreciable literature or product survey.

The structure of the project report is detailed below.

In Chapter 2, the detailed problem definition is given. The historic system is described, and the broad outline of the enhanced system is presented.

In Chapter 3, the user specification of the system is presented. This includes a functional description of the various logical modules, both hardware and software.

Chapter 4 presents the specification of the hardware requirements.

Chapter 5 deals with the software specification. This is modelled on the IEEE standard for such a specification [1]. The high level software specification is presented in this section. Detailed specification down to a functional level appear in Appendix B. A functional description of the software implementation appears in Chapter 7. Detailed software listings are not presented, as some of the information contained in them is proprietary.

Chapter 6 presents the hardware implementation of the FIFO interface card, which is the 16-bit interface card for the PC-AT compatible or higher. The details of aspects of the design appear in Appendix A.

Conclusions concerning the system and project, with remarks about further development are presented in Chapter 8.

This project report reviews:

- The system specification of the DAPS.
- Development of the hardware capture card to interface to the radar data stream.
- Specification of the system software.

- Implementation of the MSDA functions.
- The experience gained in formal specification, design and quality management.

The above aspects were solely my responsibility. Detailed design of the data filter hardware and driver software, time-code interface, some aspects of the HSDA software were delegated to colleagues. The system integration, debugging and modification of all software, and system acceptance were my sole responsibility.

CHAPTER 2

PROBLEM DEFINITION

The aim of this project has been to upgrade the DAPS to modernise the equipment, reduce the size and improve the functionality of the system. This has involved replacement of the HP computer, dedicated interface and tape drive with a dedicated PC interface card and software.

As already stated, the major reason for replacement of the existing hardware and software is that it did not fulfil all of requirements for testing and analysis of the radar; nor could it be integrated into the mobile radar laboratory easily. The existing system did not lend itself to easy transportation of the data and results for further analysis and reporting purposes, because most of the infrastructure for general analysis and word processing at Aerotek is PC based. Finally, with the advance of technology, much more functionality and performance are achievable with the enhanced system than the historic. The reliability of the historic system was also a problem.

2.1 System Requirements

A brief system requirement is given here, the full user specification appears in Chapter 3. A block diagram of the radar system, and peripheral components appears below.

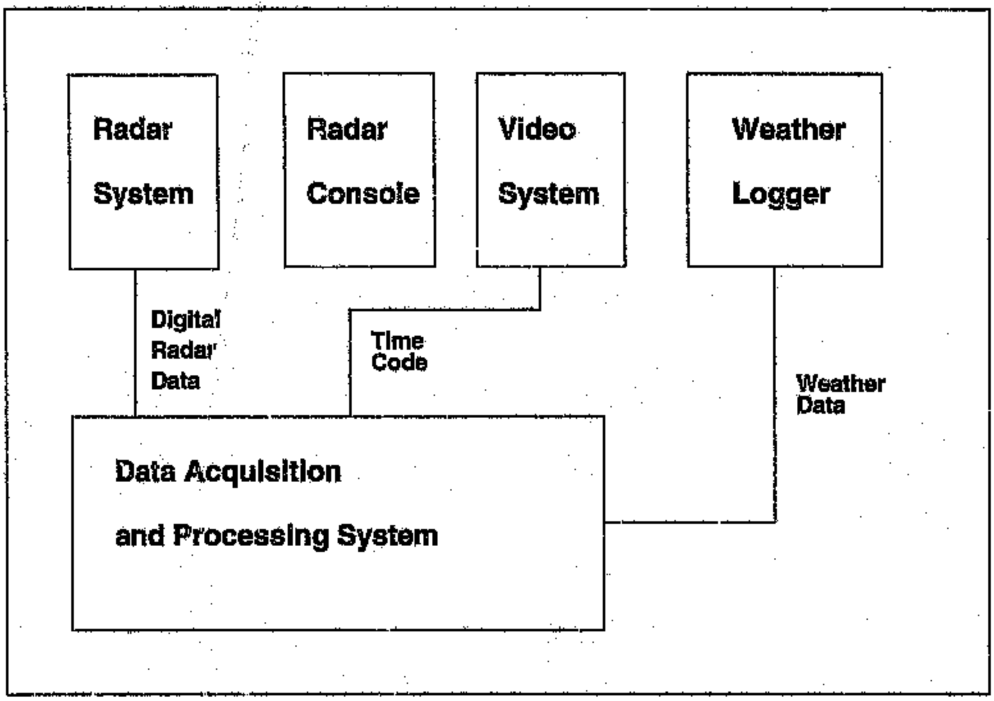


Figure 2.1 Radar System

The video system is used to record boresight video footage with overlaid radar display data, from the console. This system is used to select data for post processing. To this end the DAPS has a longitudinal time code interface used for logging of timecode during data capture. There is also a weather monitoring module with a serial port interface. This is used to log weather data during data capture. Each data file can then be referenced to a video recording, and has atmospheric data for the experiment. The atmospheric conditions affect propagation and radar return.

The DAPS is required to perform mutually exclusive MSDA and HSDA of digital radar data. MSDA is used for real time[7] and semi-real time[8] processing. Real time processing

[7] This term refers to the standard interpretation of the phrase, where the real time action occurs whilst and in pace with the constraining action of some piece of equipment.

[8] This term refers specifically in this document where data is captured in real time and analysis and display takes place within a maximum time of 5 minutes from the data capture.

is required for the vectoring[9] of aircraft. Typical functions which require semi-real time processing antenna scan plots, or calibration measurements. The HSDA function is used to collect large volumes, typically 5 to 20 MB per recording, of data which are processed off-line to obtain RCS information.

The user requirement was stated as follows:

> To replace the existing DAPS with a PC based system. The enhanced system has to preserve the existing interfaces and functionality, and improve on functionality to meet a number of processing requirements. The specification of the processing requirements is to be included in the design of the necessary hardware and software.

2.2 First Generation Data Acquisition System

At the commencement of this project the radar system was in the process of being integrated into a mobile laboratory. A block diagram of the then existing DAPS, comprising separate HSDA and MSDA systems, appears below.

[9] A term used by the SAAF to refer to guidance of aircraft under radar control. The radar operator gives course correction information to the pilot to assist him in traversing a specified path.

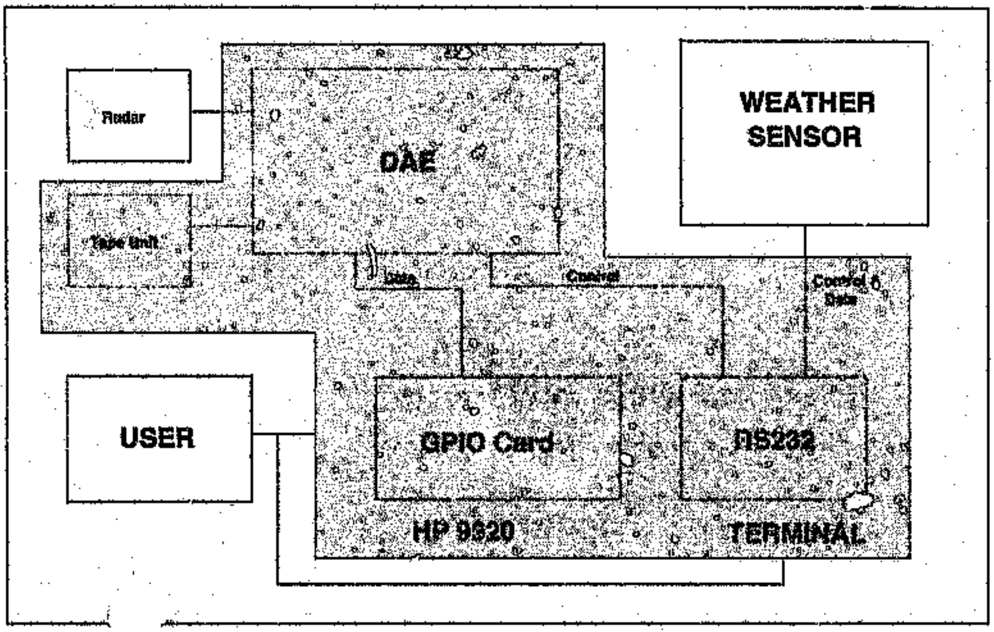


Figure 2.2 Block Diagram of First Generation Data Acquisition System

The DAE (data acquisition equipment) was the custom data filter and tape driver. This constituted the high speed capture path. The medium speed interface from the DAE was designed to suit GPIO capture to the HP9920.

The DAE module interfaced the radar data stream to two recipients. The Cypher, nine track tape unit was controlled completely by the DAE through resident software accessed by a terminal through an RS232 port. Embedded software in the DAE controlled two data filters of identical design. Each filter could be set up to pass from none to all of the data through to either the tape or the HP 9920; these being at the output of each of one of the filters. The HP 9920 data stream was passed out of a GPIO port in the DAE.

2.3 Deficiencies in First Generation System

The system detailed above had several limitations:

- It was difficult to generate processed data results from MSDA. Far more analysis and presentation software exists on PC based systems. The turn-around time, ease of presentation, and reporting of results are features which can be enhanced with the use of a PC family computers to perform the analysis.

- The system required an RS232 type terminal and the HP computer in order to be used for measurements. Having both of these pieces of auxilliary equipment in the confined laboratory space around the radar posed a space problem.

- Data acquired on the HP computer could not be easily transported to PC compatible (IBM-PC and family) machines. It was not easily portable to Unix based systems, where some of the more voluminous data processing has to occur.

- Data analysis software on the HP 9920 had to be custom expanded each time that different analysis functions were required. This is costly in terms of manpower and neither the resulting software, nor the data, was readily transportable to PC compatible machines.

- The system did not allow for expansion to include control and data collection from weather sensing equipment, which has since taken place.

- Expansion to include capture of time code was not possible on the historic system.

- The tape drive unit could not capture all of the radar parameters, even when operating at the limits of its specification. This was because media faults caused data overruns.

These problems came about because the system was developed at a time when a most suitable machine available to collect and analyse data was the HP 9720. The HP data interface buses were also suitably fast to meet the requirements. Also, the tape drive was the only high speed mass storage device readily available.

2.4 Enhanced System

The enhanced system replaces the HP 9920, terminal, tape storage device and DAE module with a single PC compatible machine. The data filtering functions of the DAE are accomplished by one hardware card. The buffering and capture of the filtered data is accomplished by a separate FIFO card. Large volumes of data are gathered to hard and magneto optical disk drives. The time code capture function is accomplished by a third card. Control of the weather sensing equipment is easily achieved through a standard RS232 interface.

There are two major software functions, namely HSDA and MSDA. The complete specification of the system appears in Chapter 3. The block layout of the enhanced system appears below.

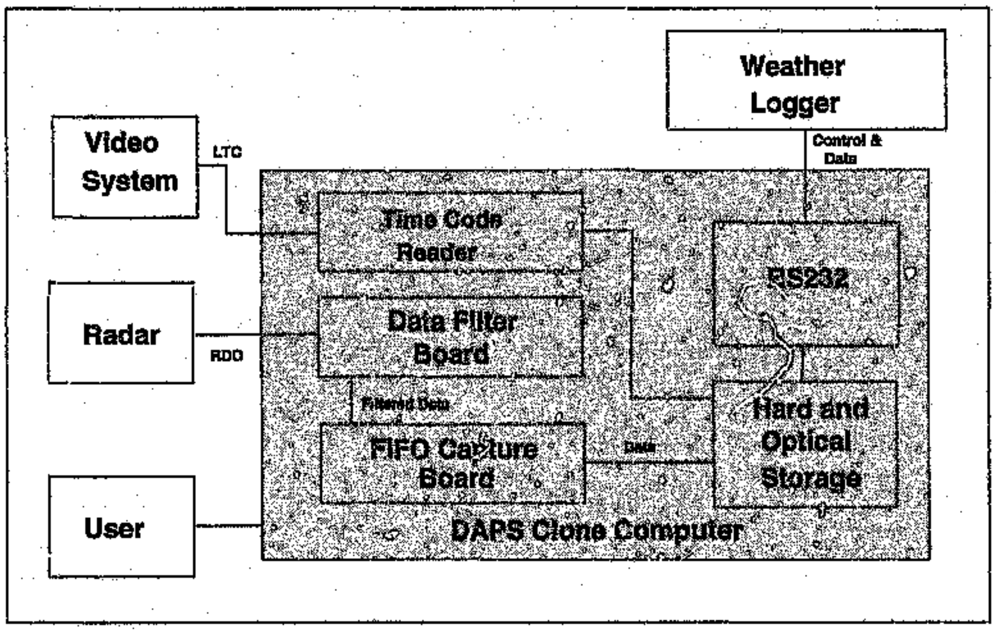


Figure 2.3 Block Diagram of Enhanced DAPS System

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

The specification of the DAPS is derived from the requirements of the type of measurement to be made with the system. The user requirement was given in the broadest of terms and the specification, which appears below, was developed from this. Detailed hardware specification follows in Chapter 4. The hardware specification is based upon the existing interfaces. The software specification which facilitates the HSDA and MSDA processes follows in Chapter 5. The former DAPS was able to perform both processes simultaneously, a requirement which has been removed from the enhanced system, ie. MSDA and HSDA functions are mutually exclusive.

3.1 HSDA

The function of the HSDA process is to capture a user specified subset of the radar data stream. This subset can be the full set if required. User selection of relevant (to a particular session) parameters allows for efficient use of storage capacity. The user makes selections of the data subset by selecting the parameters[10] to be recorded. The user can save and retrieve default settings for re-use.

Associated with each capture session (data file generated) is a textual description of the capture scenario. The system is required to automatically distinguish data files with a unique number. In the textual description of the scenario, is a summary of key weather information obtained from the weather equipment.

[10] The term parameters refers to individual data items, such as sum channel voltages, or antenna azimuth position. A complete set of each parameter selected is obtianed for each transmitted pulse of the radar. In the specification of functional processing of parameters, each parameter is conceptually an array containing pulse by pulse samples of the RDO data parameter of the same name, eg. *I*, where individual samples are referred to as *I(i)*. This would be the I channel voltage.

The user initiates the start and stop of recording. At each start and stop, a new file is generated. During the logging of data the synchronised longitudinal time code is to be inserted in to the records of data, to allow post synchronisation with events and video information.

After capture the user to be able to set a filter for extraction of data from recorded files. This filter allows for selection of a subset of parameters to be extracted from the file, the start and stop points and the type of file which is generated. At present this includes ASCII, binary or Matlab. Some of the post processing is done in Matlab. Binary files allow for transport to Unix systems and other environments. Filtered data files are then generated.

Some of the radar data has a 10 ms update rate (control signals), even though the data is framed every pulse. If a user selects 10 ms data and no pulse by pulse data, then the data filter is to extract data at the appropriate rate. This eliminates repetition and reduces the size of output files.

3.2 MSDA

The function of the MSDA process is to capture and process a user specified subset of the radar parameters, determined by the function (eg. antenna pattern) specification. For functions such as aircraft vectoring and radar alignment this takes place in real time. For functions such as calibration measurements and antenna pattern plots this takes place in a semi-real time fashion.

For real time functions the user is able to start and stop the function. At any time a hard copy of the current graph and results is generated upon user request.

For semi-real time functions, the user selects a function (type of data processing) from the available choices. The user is able to adjust default settings of the specific function chosen. Upon user command, the required data is selected, captured, processed and displayed; under user menu control where necessary. The user is then able to generate hard copies of the graphs which are plotted on the screen. Functions can be performed in a cyclic fashion until the user terminates the function in favour of other options.

CHAPTER 4

HARE WARE SPECIFICATION

Selection of a PC clone for the DAPS was a simple process, based upon available technology. The current version of this equipment comprises a 50MHz 486 ruggedised PC clone processor card with 16MB of RAM; 1MB VGA adapter, Super VGA display and standard I/O ports; Adaptec 1542B SCSI host adapter; 425MB, SCSI hard drive; 1GB, SCSI, Tahiti I Magneto Optical drive; combined 1.2MB/1.4MB, floppy/stiffy drive and a Postscript printer.

The system was selected to be fast enough to sustain the required data rate when logging data to disk (HSDA). The processor was chosen to obtain the maximum throughput for MSDA functions. The requirement for these functions is that as much data as possible should be processed as fast as possible. Thus the fastest available processor has been chosen.

As can be seen in Figure 2.3, there are four hardware interfaces in the DAPS:

- A standard RS232 port which interfaces to the weather logging equipment outside the laboratory.
- A time code reader card which interfaces to a longitudinal time code (LTC) generator which provides time code to the video system as well.
- A data filter board which interfaces to the radar data stream, and allows for selection of parameters to be extracted from each frame (at every radar pulse) of data. In the original system there were two, independent filters. Data was passed to a GPIO type output port and the tape. For ease of transition between the two systems it was decided initially to keep the functions separate and the data filter board performs the same function. This allowed for integration of the PC based

DAPS with the existing system in place. MSDA functions could be taken on by the enhanced DAPS alone at first. Development of the HSDA functions then replaced the need for the old components.

- The output of the data filter board is sent to the FIFO capture board. This board buffers the data to reduce the instantaneous data rates. Data is captured to PC memory and then disk as appropriate for MSDA or HSDA functions.

The hardware requirements for all interfaces are presented below.

4.1 Weather Sensing Equipment

The weather sensing equipment consists of a CR10 Measurement and Control Module from Campbell Scientific, Inc. This unit is a fully programmable data logger and controller which in this application is used for logging of wet and dry bulb temperatures, rainfall rates and amounts, wind speed and air pressure. The unit is capable of downloading present measurements and stored histories of data. The interface [illegible] the unit is through a standard RS232 interface. The DAPS requires one RS232 serial port to be available to the weather data logger interface.

4.2 Time Code Reader Card

There is a standard LTC chip which can decode time code and user bits from an LTC source; two of which are used here, one for time code and the other for user bit information. In the radar system the user bits are used to indicate modes and status information. LTC updates time 25 times per second. The requirement is for interrupt driven logging of time code and user bit information at every update. This requires allocation of I/O space and interrupts.

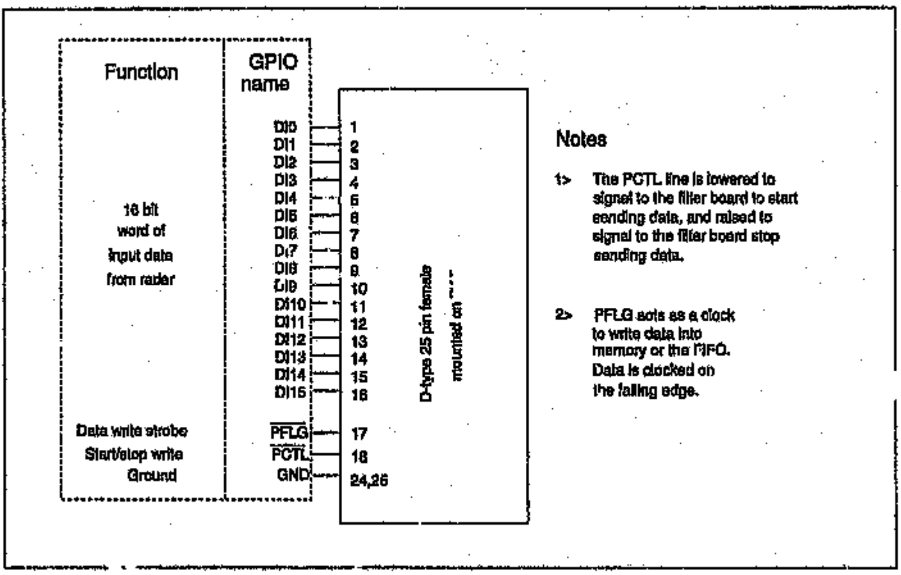
4.3 Data Filter Board Control and Data Interfaces

The data filter board is controlled through I/O ports inside the PC. The radar data stream is a high speed serial data set consisting of 60 radar parameters per frame. The data filter board performs a serial to parallel conversion on all the data. Dependent upon the programmed extraction filter, a subset of parallel data is presented to the GPIO type port on the FIFO card. Data is captured to this port for MSDA and HSDA functions. The existing data filter hardware (in the DAE) is replicated on this card. The former embedded software control which controlled the data selection through I/O ports and latches in the DAE is replicated in the PC, through ports in the I/O address space.

The synchronisation, decoding and serial to parallel conversion of the radar data are not specified here, since the functionality of the DAE was replicated.

4.4 FIFO Capture Board (with GPIO type data port)

The data filter flows to the FIFO capture board over a reduced form of the HP GPIO data bus. The GPIO bus is not used in full in the case of the PC (the interface on the HP 9920 was of course a standard GPIO card) and so it is called a reduction of the full standard. The only non-standard interface required is in fact the reduced GPIO subset used here, and this is presented in Figure 4.1.



1> The PCTL line is lowered to signal to the filter board to start sending data, and raised to signal to the filter board stop sending data.

2> PFLG acts as a clock to write data into memory or the FIFO. Data is clocked on the falling edge.

Figure 4.1 Subset of GPIO Standard Used in DAPS

The data format and associated rates are completely defined by the existent radar system. This is detailed below, with reference to Figure 4.2.

The data to be captured over the 16 bit parallel data bus is pre-defined to consist of from 1 to 60, 16 bit parameters. The instantaneous and average rates of transfer vary, depending upon how many data parameters are selected and extracted from each frame. The maximum duration of a single data frame transfer over the bus is 192 µs. This is the duration of one frame of all 60 parameters, with 3,2 µs between adjacent parameters. The data frame is repeated every pulse repetition interval (pri) of the radar, which varies from 384,6 µs to 454,5 µs. In the worst case (minimum pri with all 60 parameters selected), the maximum data clock rate (burst) on the $\overline{\text{PFLG}}$ line will be 312,5 kHz and the maximum average data rate is, however, lower at 156 000, 16 bit parameters (words) per second. This is because the transfer takes place in a burst at every frame refresh period and does not utilise the entire refresh period available.

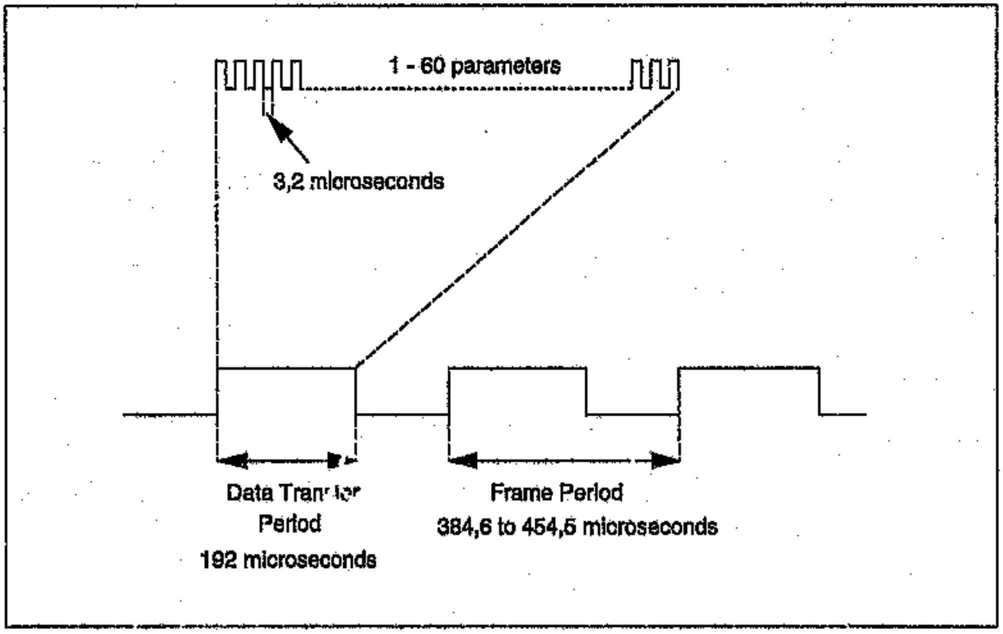

1 - 60 parameters
3,2 microseconds
Data Transfer Period 192 microseconds
Frame Period 384,6 to 454,5 microseconds

Figure 4.2 Data Frame Format

The HSDA process is required to capture data to disk at the above rates. No processing is required during capture. The process must acquire and store all 60 parameters continuously. This is an average throughput to disk of 305 kB per second (from 156 000 16 bit parameters per second).

The maximum number of parameters that the PC is required to collect simultaneously for real and semi-real time processing is determined by the type of processing required. A minimum specification of the quantities of data required to satisfy computational throughput of the MSDA process has been determined from the type of experimental procedure to be performed by the radar system. In each case the minimum set is captured to allow the maximum processing throughput.

As will be seen in the specification of the MSDA and processing functions there are two types of capture and processing required. In the first form, semi-real time functions, the maximum requirement is that the DAPS is capable of acquiring up to 10 parameters for as long as 15 seconds, after which the data is processed and results displayed. For efficiency, this data is to be stored in memory and not to disk. It is of little concern to those functions and the results that data is lost cyclic execution of capture and processing. There is an obvious trade-off between the number of parameters and overall volume of data collected (collection time), and the ability of the machine to process and display results completely before the next capture can occur. For the type of result required, the limitations are acceptable, and the exact limits only emerged as the system was developed.

The second type of processing is true real time processing. Only one of these is specified here and it is required to run continuously in real time. The effective data rate is much lower because some of the parameters used are only updated at 10 ms intervals, even though they are included in every frame. The specification of the function implemented requires that 10 parameters with 10 ms update rates be acquired, and two with pri update rates. This reduces to an average throughput of 12,1 kB per second with continuous processing of data and display of results.

4.5 System Hardware Specifications Summary

The following table summarises the key hardware specifications of the DAPS PC and its associated interfaces, as determined from the above constraints.

Table 4.1 Hardware System Specification

Item	Specification
Weather Sensor Interface	Standard RS232 port on DAPS PC
Time Code Reader Card	Port allocation in PC I/O space for control and data flow. Interrupt allocation for interrupt driven logging of time code.
Data Filter Interface	Port allocation in PC I/O space for control functions. Data direct to FIFO card for HSDA/MSDA through residual GPIO type interface
FIFO Capture card Interface	16-bit parallel interface with 2 control lines and data ground Maximum Burst Data Rate - 312 500 words per second Maximum Average Data Rate (throughput) - 156 000 words per second Storage Capacity to Disk - no more than 305 kB per second continuously until disk full. Storage Capacity to Memory - 26 000 words per second for 15 seconds (no more than 762 kB at no more than 51 kB per second).

CHAPTER 5

SOFTWARE SPECIFICATION

The structure of this specification is based upon the ANSI/IEEE Std 830-1984, entitled "IEEE Guide to Software Requirements Specifications" [1]. The sections dealing with specification of the functions are done using De Marco style data flow diagrams as extended to real-time systems by Hatley and Pirbai [2]. The high level specifications are presented here. Detailed specification of some of the software functions appear in Appendix B.

5.1 Introduction

5.1.1 Purpose

This Software Requirement Specification (SRS) serves to define the objectives and functions of data acquisition control and analysis software for the DAPS, a sub-system of the radar system. The primary motivators for the collection of data from the radar system are the need to analyse the performance and calibrate the system, and measurement of target parameters during track.

The users of the software and hardware resulting from this project will be the developers of the DAPS system, and the team involved in the development of the radar system.

5.1.2 Scope

The software described in this document is the Data Acquisition Processing Package, the term DAPS being used to refer to the hardware and software system as a whole.

The following specifications are applicable to this SRS, but only relevant information will filter through to section 5.3, the bulk of such documentation being beyond the scope of this SRS.

- The specification of the hardware and detailed specification of the software driver for the data filter board. The relevant information to this document is the software interface.
- All documentation on the weather logging equipment. The only relevant information is the control interface subset which allows weather data to be retrieved for logging to disk files which are processed off-line by other systems.
- The user requirements for real and semi-real time analyses (MSDA) which stem from radar systems analysis, design and testing considerations. Software specification is derived from these user requirements; Eg. 'Calibration of the radar system for the purpose of measuring radar cross-section (RCS)' or 'measurement of the improvement factor of the radar MTI'. Definitions of these concepts are readily available in radar literature. Implementation of these functions is an open-ended topic, as more and more functions may be required from the user, which would lead to expansion of the specification and associated software. An initial set of such functions is included here as required by the user on acceptance of the system.
- The user requirements for high speed acquisition and storage to disk which were (HSDA).

5.1.3 Definitions, Acronyms, and Abbreviations

The terms defined throughout this report as footnotes and in the glossary (Appendix C) are applicable here.

5.1.4 Overview

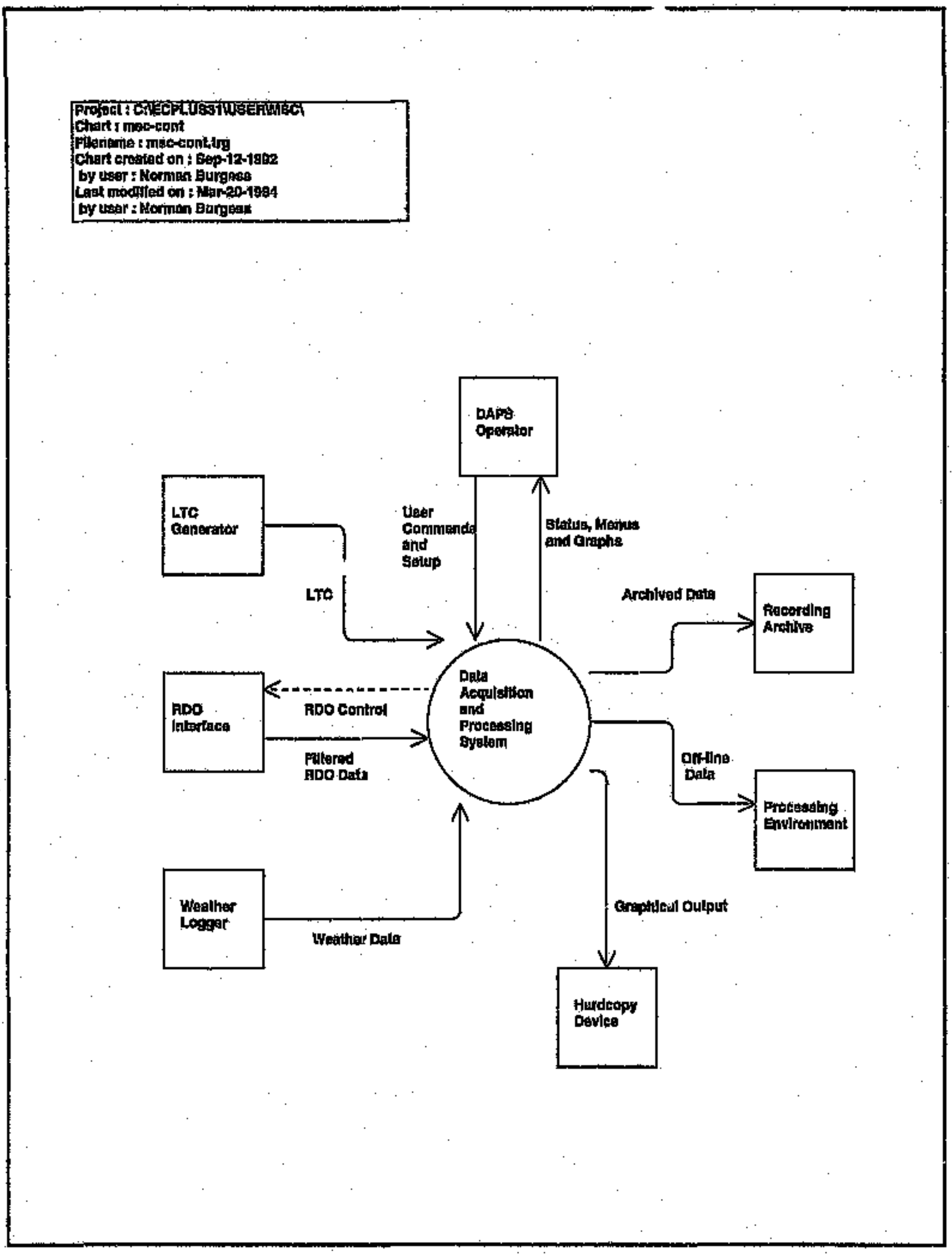
The rest of this section concerns itself with a more detailed description of the software package and its fulfilment of the two major types of acquisition, namely HSDA and MSDA. The software resulting from this specification is specific to the hardware interface cards used, and is not generally portable to other PC's. The uniqueness of the system removes constraints on the use of specific hardware features of the PC family member used, although these are as always to be avoided if equivalent performance can be achieved by more generic means.

5.2 General Description

5.2.1 Product Perspective

The DAPS system for which this software is intended has been described in Chapter 2. The product of this SRS is the software to implement HSDA functions and MSDA functions. This includes driver software for the hardware interfaces and the user interface.

To place the software product in perspective, the architecture of the system is analysed in detail first. Three concept diagrams are useful in this regard. The first is an architectural context diagram to establish where the DAPS fits into the system logically. It serves to identify external entities and the data flows to and from these entities.



Project : C:\ECPLUS31\USER\MSC\
Chart : msc-cont
Filename : msc-cont.trg
Chart created on : Sep-12-1992
by user : Norman Burgess
Last modified on : Mar-20-1994
by user : Norman Burgess

Figure 5.1 DAPS Architecture Context Diagram

An architecture flow diagram gives a more detailed breakdown of the control and data flow between hardware components inside the DAPS, as opposed to the flows to and from external entities. The software will be required to drive each of the interfaces shown.

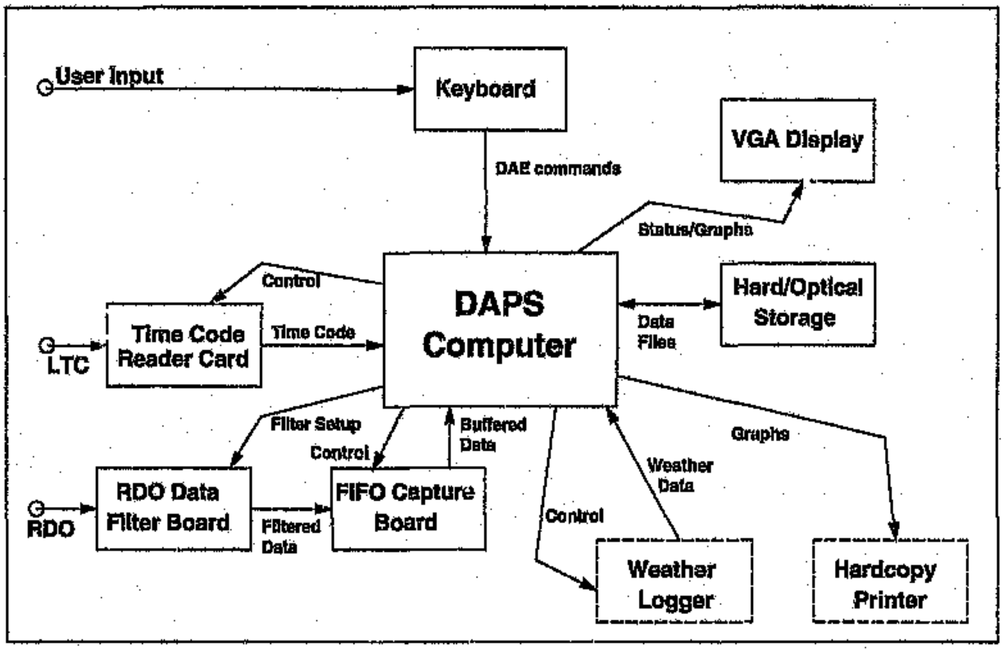


Figure 5.2 DAPS Architecture Flow Diagram

A detailed view of the system architecture, and in particular shared PC bus, is shown below. This serves to identify the shared resources. This diagram assists with the design phase by identifying the devices which need to be accessed by a number of higher level functions.

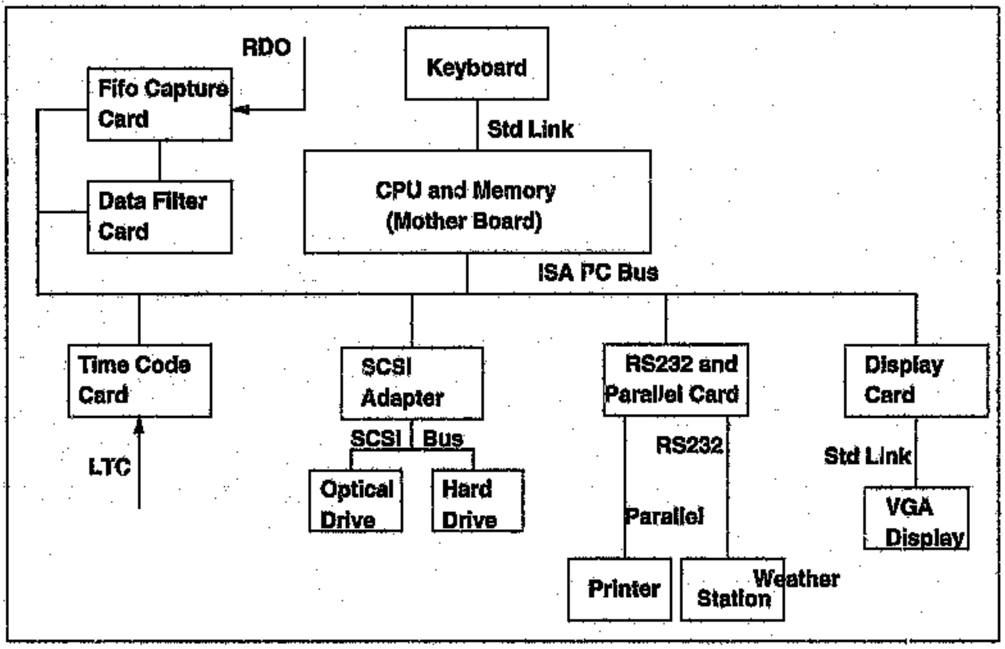


Figure 5.3 DAPS Architecture Interface Diagram

5.2.2 Product Functions

There are two major functions of the DAPS software, namely:

- HSDA. This is the control of the high speed data capture process, including verification of the validity of the data. This includes control of all aspects of the storage to optical disk and the associated data filter use during storage. The time code driver and weather logger driver are included in this major function.

- MSDA. This is capture, processing and display of a data subset for radar system performance evaluation, radar calibration and measurement of certain target phenomena which are applicable to the control of an experiment, eg. Maximum and minimum RCS.

Common to both functions is the control of the data filter board and FIFO capture board, which is used to capture data for both.

The software is to be installed on the ruggedised PC clone that has been purchased for the purpose of performing the functions described above. The DAPS is mounted in one of the three radar consoles.

The functional breakdown below gives a broad description of the five functional modules in the DAPS. This breakdown identifies the logical functions which stem from the major functions of MSDA and HSDA. These are then expanded into detailed functional specifications. The five major functions are identified in the following data flow diagram. The processes reflected here are the five major processes required to implement the DAPS.

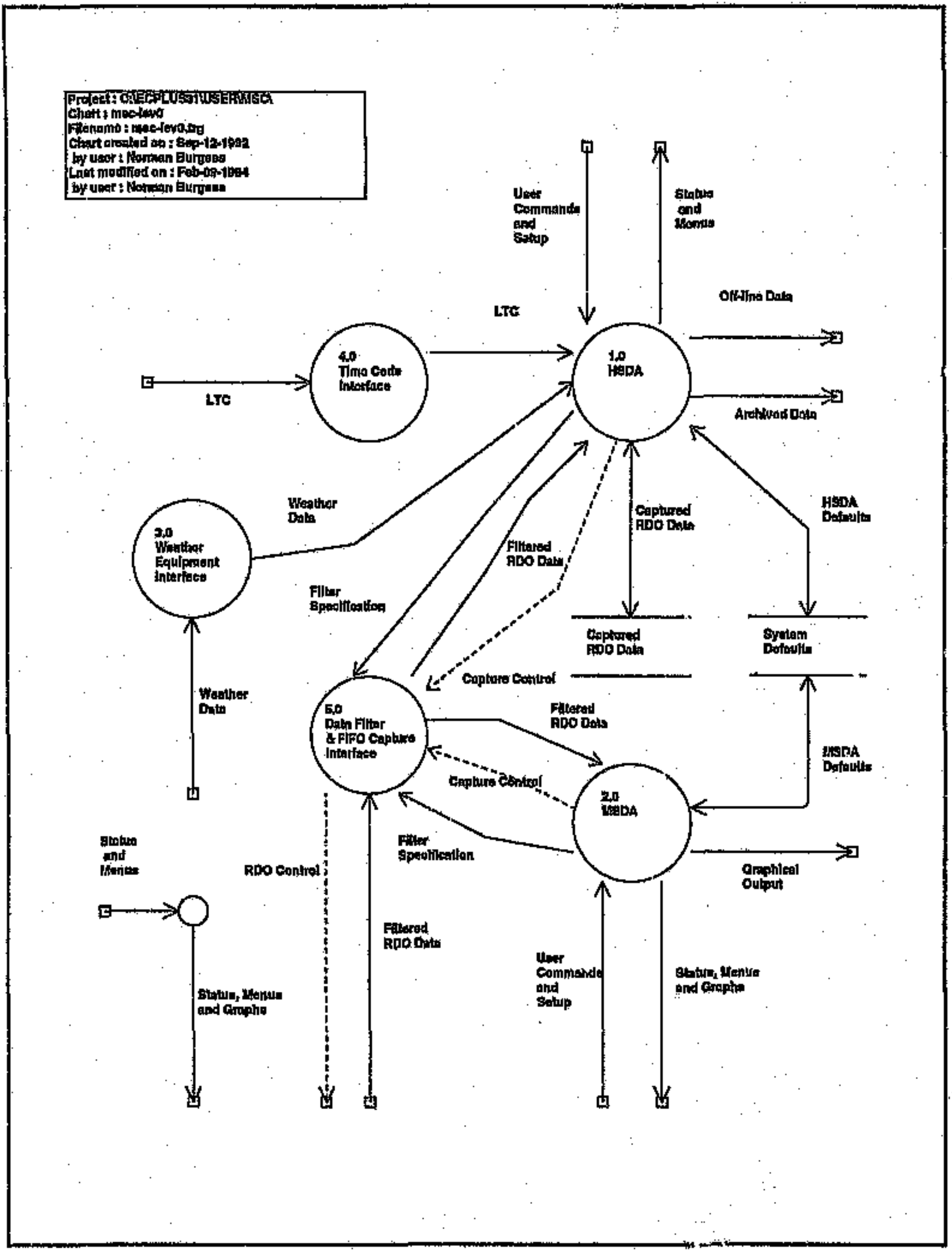


Figure 5.4 DAPS Level 0 Data Flow Diagram (DFD)

5.2.2.1 HSDA System Functions

Selection of a data filter (selection of parameters) for the HSDA process. The filter allows user selection of a number of parameters (1 to 60) for recording. A complete specification of the file header. Control of (starting and stopping) the storage to hard and optical disk. Functions to manage default settings and free space. Retrieval and further filtering of data (amount and parameters) for post processing.

5.2.2.2 MSDA Capture and Analysis Functions

These are high level functions dealing with capture of filtered data to memory for the purposes of real time or semi-real time analysis and display of the results. Only a subset of possible analysis options is specified and completed here; a modular approach will ensure that other functions can be specified and implemented.

5.2.2.3 Weather Equipment Interface Functions

Weather data consisting of temperature, humidity, and wind speed and direction are to be logged onto the headers of data files on the optical disk automatically at the start of each HSDA recording. This driver can also be used to make weather information available to the user upon request from other applications.

5.2.2.4 Time Code Interface Functions

A driver is used to set up time code card to decode time code information and prompt the main functions on an interrupt driven basis with current time code and user bit information. This information is used by the HSDA function during data capture.

5.2.2.5 Data Filter Board and FIFO Capture Board Control Functions

These are the functions which take care of setting of filter options on the data filter board (from HSDA and MSDA specifications); initialisation of the FIFO capture board, starting and stopping of the data stream and writing of FIFO buffers to memory or disk. These functions are used by both HSDA and MSDA functions.

5.2.3 User Characteristics

The users of this software are all to be familiar with the radar system and its function as well as with the hardware of the system. The analyses referred to in section 5.2.2.2, and defined in Appendix B, are those regularly required for operation of the radar. The provision of files of data on the PC gives access to the data from other environments (ASYST, Matlab, Mathematica and compiled languages) for user specific analyses which are the responsibility of the user. The user is expected to familiarise himself with the operation of the DAPS. Elementary operating instructions will be provided in written form.

5.2.4 General Constraints

The constraints on the design of the DAPS software are primarily hardware constraints which have already become clear in the preparation of this SRS. The control of the weather monitoring equipment takes place through the RS-232 serial interface ports. The control of the data filter board, time code reader card and FIFO capture card takes place through driver software which has to co-exist with this software. The data capture to the PC takes place through the custom GPIO interface. The data parameters available and the formats and rates of transfer are dictated by the radar system. The constraints on this system emerge from the hardware and software design of the systems that interface to it.

5.2.5 Assumptions and Dependencies

In preparation of this SRS it has been assumed that the software is to be implemented on an MS-DOS based 80486, PC clone having 2 RS-232 ports, one parallel port and three custom prototype boards. The Borland C++ and Turbo Assembler environments are assumed to be available for the development of code. The software interface of the weather equipment is pre-defined; this system has embedded software which is driven from the serial interface. The control interface and filter setup of the data filter board are defined by the existing hardware, which is replicated. The time code reader chip has a pre-defined control method. The software developed from this SRS has to accommodate these specifications.

5.3 Specific Requirements

5.3.1 Functional Requirements

The functional requirements are presented in the five categories developed above. The detailed specification of each of the functions is presented in Appendix B. Certain of the functions described here do use, or depend on others in the list. The manner in which functions are integrated (menu forms and structure), and some of the specific aspects of individual functions (eg. scaling factors for parameters) are implementation issues.

5.3.1.1 HSDA System Functions

The detailed data flow diagram, and specification of the functions is contained in Appendix B. The HSDA process is required to manage the setup, capture, retrieval and further filtering of RDO data.

5.3.1.2 MSDA Capture and Analysis Functions

The detailed data flow diagram, and specification of the functions is contained in Appendix B. The MSDA process is required to manage the setup, capture, processing and display of processed RDO data.

5.3.1.3 Weather Equipment Interface Functions

The detailed specification of this function is contained in Appendix B.

5.3.1.4 Time Code Interface Functions

The detailed specification of this function is contained in Appendix B.

5.3.1.5 Data Filter Board and FIFO Capture Board Control Functions

The detailed data flow diagram, and specification of the functions is contained in Appendix B. This process is required to manage programming of the RDO filters and control of the capture process.

5.3.2 External Interface Requirements

5.3.2.1 Weather Equipment Software

The weather equipment is driven through the serial port by standard drivers provided by the manufacturers of the equipment, which access the embedded software of the device. This interface has to be accommodated.

5.3.2.2 Data Parameter Set

The table on the following page describes the data parameters as they appear in the RDO data stream. The first column is the number of the parameter in the stream, considering the data stream as word serial. Although there is a difference in the update period of different parameters, they all appear every pulse repetition interval (pri) of the radar. Typically there are two categories resulting in the distinction. The track parameters are computed every 10 ms, and hence their update accordingly. Instantaneous detection information and pulse amplitude measurements are updated every pri.

There are a few additional terms which require explanation to assist with interpretation of this table.

PRFTEL is a pulse repetition interval counter which counts from zero with a mark every 10 ms. This counter is used as an absolute reference for the pulse by pulse data.

Predicted parameters are values stabilised values from the tracking filters of the radar with a small look ahead time. Typically these are the most stable for measurement purposes.

Predicted values wrt boresight are values predicting the offset from the boresight position.

RPG stands for Radar Positioner Group. This is the entire assembly comprising motorised positioner, transmitter, receiver and down convertors.

Designator can refer to a search radar, optical putter on, console designation or optical (TV) tracking system.

Error refers to the monopulse measured errors from the monopulse antenna configuration, and the calculated range errors.

Range of Detection refers to the ranges at which detections are reported by the CFAR thresholding processor.

Channel parameters are the raw, measured voltages of each of the monopulse channel components.

Early and Late samples are used to establish range tracking.

Table 5.1 RDO Data Stream Parameters

	Data Word Name	Description	Update Period	Valid Time w. r. t. PRFTEL=
1	RNK2K	Predicted target range (20ms ahead)	10ms	+5ms
2	AZK2K	Predicted target azimuth (20ms ahead)	10ms	+5ms
3	ELK2K	Predicted target elevation (20ms ahead)	10ms	+5ms
4	RNK2KV	Predicted target range rate	10ms	+5ms
5	AZK2KV	Predicted target azimuth rate	10ms	+5ms
6	ELK2KV	Predicted target elevation rate	10ms	+5ms
7	BAZK1K	Predicted target azimuth wrt. boresight (10ms ahead)	10ms	-5ms
8	BELK1K	Predicted target elevation wrt. boresight (10ms ahead)	10ms	-5ms
9	RPGAZC	Azimuth command to RPG	10ms	0ms
10	RPGELC	Elevation command to RPG	10ms	0ms
11	ADCDLC	Range Gate position command	10ms	0ms
12	RPGAZ	RPG azimuth angle	10ms	-5ms
13	DESRNG	Designator range	10ms	
14	RPGEL	RPG elevation angle	10ms	-5ms
15	DESAZI	Designator azimuth	10ms	
16	ELTR	Real elevation error (ru. *32)	10ms	
17	ELTI	Imaginary elevation error (ru. *32)	10ms	
18	AZTR	Real azimuth error (ru. *32)	10ms	
19	AZTI	Imaginary azimuth error (ru. *32)	10ms	
20	RNGF1	RANGE OF DETECTION_1	PRI	
21	RNGF2	RANGE OF DETECTION_2	PRI	
22	RNGF3	RANGE OF DETECTION_3	PRI	
23	RNGF4	RANGE OF DETECTION_4	PRI	
24	RNGF5	RANGE OF DETECTION_5	PRI	
25	RNGF6	RANGE OF DETECTION_6	PRI	
26	RNGF7	RANGE OF DETECTION_7	PRI	
27	RNGF8	RANGE OF DETECTION_8	PRI	
28	BDCDLY	Position of B gate (SUMG1)	PRI	
29	SUMG1I	First B gate	PRI	
30	SUMG1Q		PRI	
31	SUMI	Sum I channel	PRI	
32	SUMQ	Sum Q channel	PRI	
33	AZI	Azimuth I channel	PRI	
34	AZQ	Azimuth Q channel	PRI	
35	ELI	Elevation I channel	PRI	
36	ELQ	Elevation Q channel	PRI	

Table 5.1 RDO Data Stream Parameters (continued)

	Data Word Name	Description	Update Period	Valid Time w. r. t. PRFTEL=
37	ESI	Early Sample I channel	PRI	
38	ESQ	Early Sample Q channel	PRI	
39	LSI	Late Sample I channel	PRI	
40	LSQ	Late Sample Q channel	PRI	
41	ADCER	Range error	10ms	
42	RNPOSM	Range position measurement	10ms	-15ms
43	AZPOSM	Azimuth position measurement	10ms	-15ms
44	ELPOSM	Elevation Position measurement	10ms	-15ms
45	XBWI	Bandwidth index of X-filter	10ms	0ms
46	YBWI	Bandwidth index of Y-filter	10ms	0ms
47	ZBWI	Bandwidth index of Z-filter	10ms	0ms
48	SUMG2I	Second B gate	PRI	
49	SUMG2Q		PRI	
50	SUMG3I	etc.	PRI	
51	SUMG3Q		PRI	
52	SUMG4I		PRI	
53	SUMG4Q		PRI	
54	SUMG5I		PRI	
55	SUMG5Q		PRI	
56	PRFTEL	PRI's in 10ms	10mPRIs	0ms
57	MODE	SPARE(9), TGT_IN_RSW(1) TGT_IN_RG(1), MODE(5)	10ms	0ms
58	STATES	DESIGNATORS(3), MTI_ON/OFF(1), RW_ON/OFF(1), DUMMYLOAD(1), ABN_THRESH(1), PRF/TRACK(1), TRACK_ON_JAM(1), PRF_CONTROL(3), AGC_OFF(1), RX_CAIN(3)	10ms	0ms
59	SCAN	SPARE(3), SCAN_RATE(2), SCAN_PATTERN(3), SCAN_HEIGHT(4), SCAN_WIDTH(4)	10ms	0ms
60	SWITCH	SPARE(10), AZ HAND/FOOT(2), EL HAND/FOOT(2), RANGE HAND/FOOT(2)	10ms	0ms

5.3.3 Performance Requirements

5.3.3.1 Static Requirements

The system comprises a single stand alone PC clone with hard-co࿾y device as defined previously. This is a single user data capture and analysis system. Only one data file is handled at a time, but these files can grow to hundreds of Megabytes (HSDA functions).

5.3.3.2 Dynamic Requirements

During high speed capture, the PC is required to log all data at the prescribed rates as previously defined until the optical cartridge is full. The MSDA system is required to operate in semi-real time and real time modes.

For semi-real time operation, the graphical output of the processing is required to be available no more than 5 minutes after capture has taken place, whereafter hard-copy should be immediately available.

For real time functions, such as the flight control console, the system is required to log an process the required amount of data in real time, and continuously until the process is terminated.

5.3.4 Design Constraints

5.3.4.1 Features Common to the MSDA Functions

5.3.4.1.1 Calibration of Parameters

Calibration and scaling factors for radar information, target position and target trajectory are pre-defined by the hardware in the radar system. The calibration and scaling constants are used to convert collected data words and integer values to real units. This conversion should be automatically included in the implementation of MSDA functions as appropriate.

5.3.4.1.2 Graphical Outputs

The required graphical output for each function has much in common with other functions so far as axes, scaling and labelling are concerned. It is clear that high level graphics functions of the sort which can be purchased or are shipped with the development systems are required for common access.

5.3.4.1.3 Hard-copy Option

As usual for such a system, it is desirable to obtain a hard-copy of a graphs and associated means and variances. This option must be accessible from any of the MSDA functions.

Hard-copy is specified as through a Postscript printer.

5.3.4.1.4 Storage and Retrieval of Results (Reduced, Processed Data Sets)

It is desirable to allow the user to store and retrieve results from any of the functions in the MSDA. This function has to implemented inside each of the MSDA functions, since the format of the stored data is determined by the function.

5.3.5 Attributes

In order to ensure that the system is maintainable, a hardware and software configuration and version control is required. To facilitate proper collation of the large amounts of data to be collected, suitable standards for naming and archiving of data are required.

To achieve maximum availability it is required that spare interface cards are kept available. The entire software installation has to be backed up and available for re-installation while at remote sites.

5.3.6 Other Requirements

The radar environment is harsh, as are many of the deployment locations. For this reason the interface cards need to be ruggedised.

CHAPTER 6

HARDWARE DESIGN AND IMPLEMENTATION

The design of the data filter card and time code interface were tasked out to colleagues. The FIFO interface was designed by the author. This section gives a functional description of the hardware card designed to interface to the GPIO-type data stream. The detailed circuit diagrams and description are contained in Appendix A.

6.1 Hardware Design

Apart from good design practice, one factor dictates that the GPIO bus needs to be buffered from the PC to ensure complete data integrity. This is the high instantaneous data rate, detailed in Chapter 4. The average data rate is well within the capability of the 486 mother board used, but the instantaneous rate exceeds the 486's capability.

It was decided that a FIFO (First In First Out) principle would be used for data buffering. All machines from the PC-AT up are capable of handling 16-bit I/O, and so a 16-bit 16k deep FIFO was deemed suitable. In the worst case for capture to memory (MSDA) this depth provides a 630 ms buffer, while in the worst case for capture to disk (HSDA) it provides a 105 ms buffer.

The IBM-PC standard has a section of the port address space reserved for prototype development, from hexadecimal 300 to 31F (indicated from here on by a $ sign preceding the address eg.$300). It was decided that the interface card would occupy a portion of that space. The exact addresses used appear in Appendix A.

With a buffered FIFO system there are two clear modes of operation. In the first the system can acquire data independently up to the size of the buffer without any transfer to the PC memory or disk until after the transfer. The second mode, where the amount of captured data exceeds the capacity of the FIFO, requires dedication of the PC to the capture until it is finished, or until a critical function requires attention. With this in mind it was decided to implement a hardware counting system for data transfers involving 16 384 (the size of the FIFO) or less data values. This function, the use of a FIFO and the usual buffering of the output and inputs led to the following block diagram of Figure 6.1.

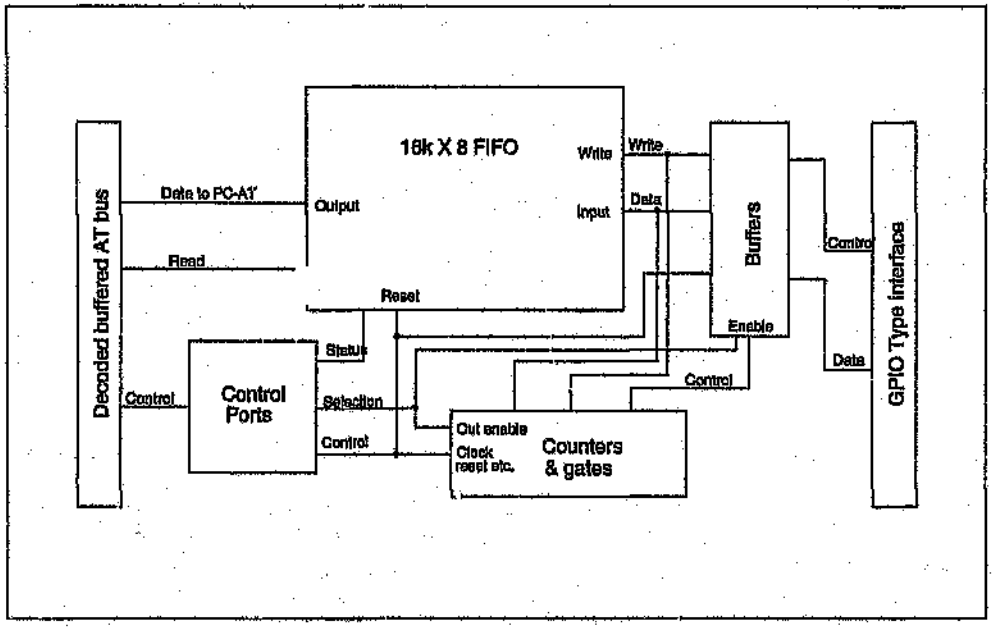


Figure 6.1 Block Diagram of FIFO Prototype Card

The counters serve an additional function during power-up and initialisation. They can be used to execute a self test function where their output is clocked through the FIFO. This provides a fairly extensive test of the control, counters and FIFO. The buffers between the GPIO port and the input to the FIFO allow for selection between the counters and the GPIO data as input to the FIFO.

The decoded and buffered AT bus is as described in technical documentation on the PC hardware [3]. I/O ports in the prototype address space of PC are allocated for the purpose of controlling the FIFO, counter and buffers. There is also one data address allocated for reading the incoming data.

Control functions from the control port of the block labelled "Control Ports" include the following:

- Clocking of the counters for self test modes and initialisation of a count for counted data modes
- Reset of the counters
- Reset of the FIFO buffer

The selection port provides a means for selection of the data input to the FIFO.

The status port is used to monitor the FIFO status. ie. Quantity of data in FIFO.

If the counted data mode is enable then the write line of the FIFO input is used to clock the counter. The counters in turn are connected to secondary control of the buffers so that data input can be terminated when the required amount of data has been clocked.

The control connection between the buffers and the GPIO port represent the flag and clock of the GPIO port.

Detailed circuit design took place. Appendix A contains a functional description of the ports which provides the interface information necessary for software implementation, and gives an overview of the steps for control of data transfer. The circuit diagrams are also presented in Appendix A along with a detailed description of the circuit.

CHAPTER 7

SOFTWARE DESIGN AND IMPLEMENTATION

7.1 Work Breakdown

The development of the software functions was broken down functionally according to the specification of Chapter 5. Colleagues were tasked with initial design and coding of the HSDA and Time Code processes. The other processes were designed and implemented by the author. Final integration and all debugging and subsequent modification of all processes was the author's responsibility.

In each case the processes and data dictionary items specified in Chapter 5 were finalised and a design of the process took place, followed by coding in C++, Assembler and Matlab.

The following sections give descriptions of how the functions were implemented.

7.2 HSDA Functions

This module was developed in Borland C++ 3.1 using the Turbo Vision user interface. This tool-box facilitates easy development of user input screens and the user control interface. The time code, weather equipment and RDO interface driver software is linked into this code.

7.3 MSDA Functions

The high level processing requirement of these functions made implementation in Matlab desirable. In addition, the desire of end users to process data in a custom fashion in Matlab indicated that as much of the development as possible took place in Matlab (Version 3.x).

The RDO interface driver software was used to develop a memory resident driver which could be called from Matlab to collect and transfer data to memory.

All functions were then developed in the Matlab environment. Rather than showing pictures of the graphs as they were planned, samples of actual output of the system are included to show the form of graph plotted by the functions

7.3.1 Radar Improvement Factor Measurement

The following graph is a representative plot of MTI improvement factor as a function of time over a period of 2 seconds. This particular plot indicates that the radar was not optimally set up for the measurement, since there is a large fluctuation, and poor improvement factor (35 dB is to be typically expected)

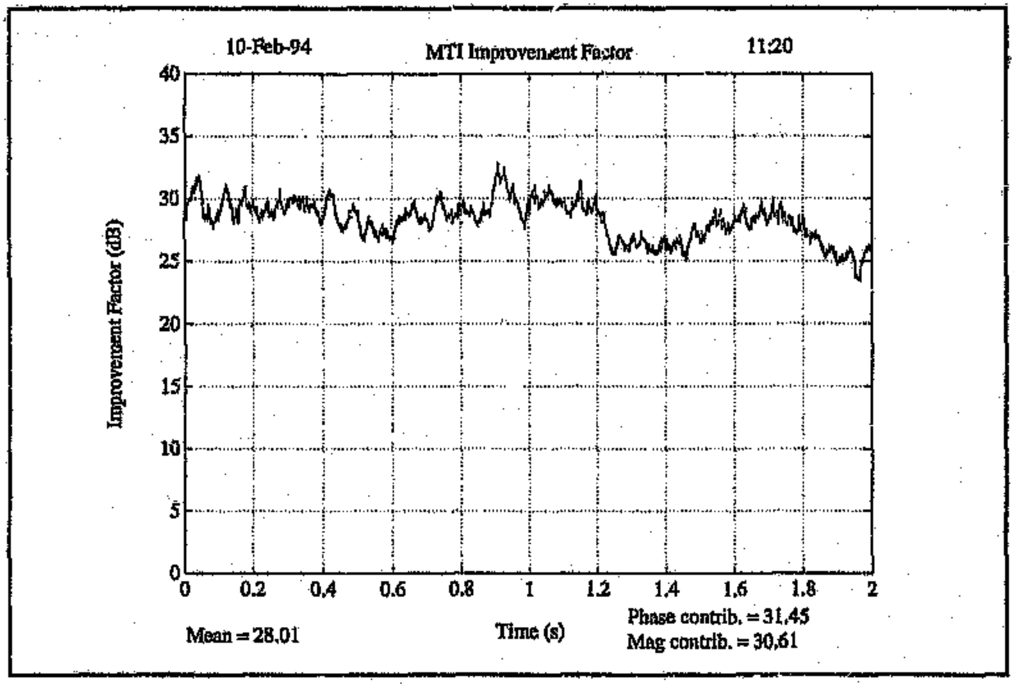


Figure 7.1 MTI Improvement Factor

7.3.2 Sum, Azimuth and Elevation I and Q Channel Characteristics

Sample graphical outputs which were developed for this purpose appear below. The target in this case was .. small corner reflector on a mast. For this reason there are slow fluctuations in the individual *I* and *Q* channel components as the reflector moves back and forth. The plot of the channel powers shows the power in the sum and difference channels of the radar. This is a typical return from a stationary target. The spectrum measurement is a function of radial velocity of the target [4], zero in this case, and is spread around this value for more complex targets.

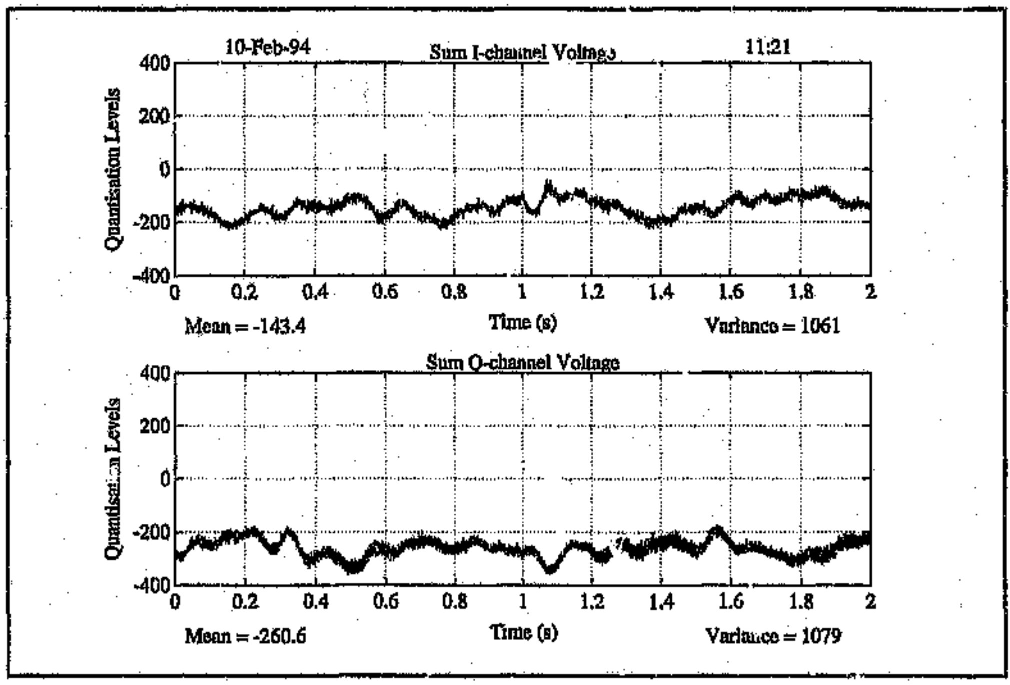


Figure 7.2 Sum Channel Voltages

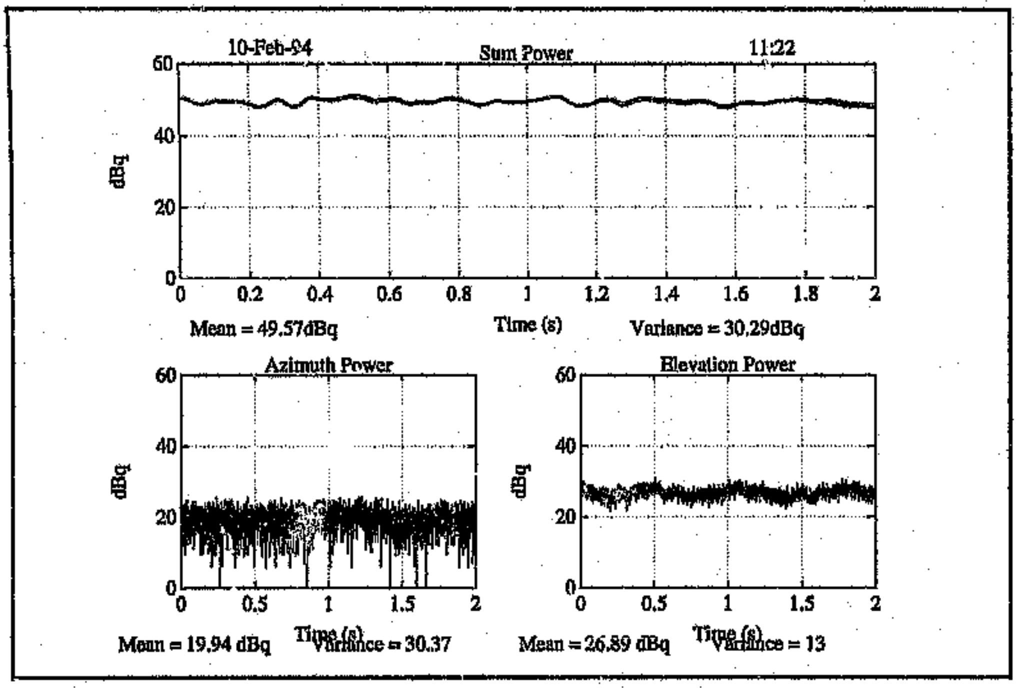


Figure 7.3 Receiver Channel Powers

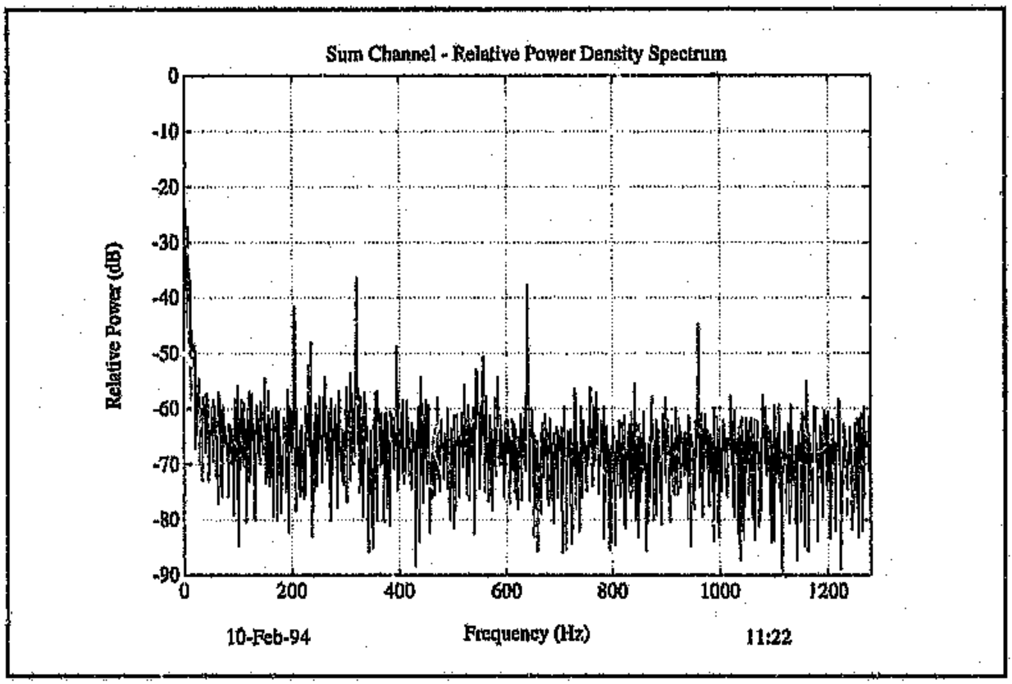


Figure 7.4 Sum Channel Power Density Spectrum

7.3.3 Antenna Pattern Measurement

These graphs serve to verify the radar antenna radiation pattern. The results presented below are representative of a typical radar antenna pattern. It is always useful to compare the sum and difference beam gains and widths, so these appear together. The error slope function is an indication of the performance of the antenna around the boresight, where tracking occurs.

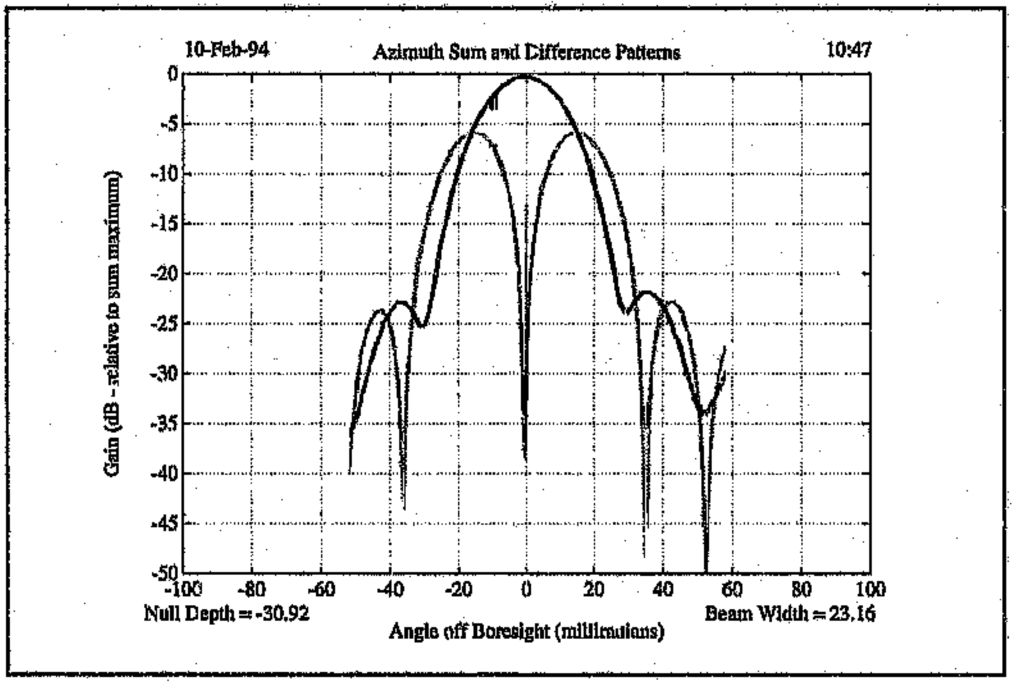


Figure 7.5 Azimuth Antenna Patterns Cross-section

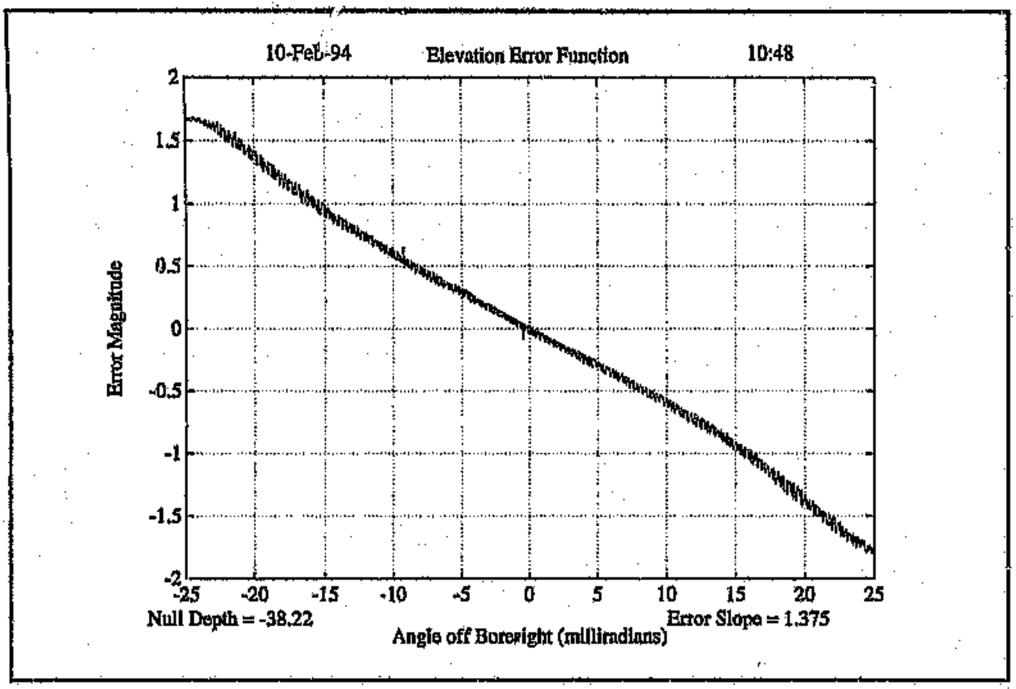


Figure 7.6 Elevation Error Slope

7.3.4 RPG Scan Pattern and Acceleration Measurement

This tool is used to assist with optimisation of the RPG control functions, so that the most efficien. scan patterns can be obtained. The graphical outputs presented here show a section of a box scan over a prescribed sector. The areas of concern during analysis are the corner points and accelerations and velocities at such points. The time plots of position, rate and acceleration; and the ablility to zoom any of the plots allows detailed analysis.

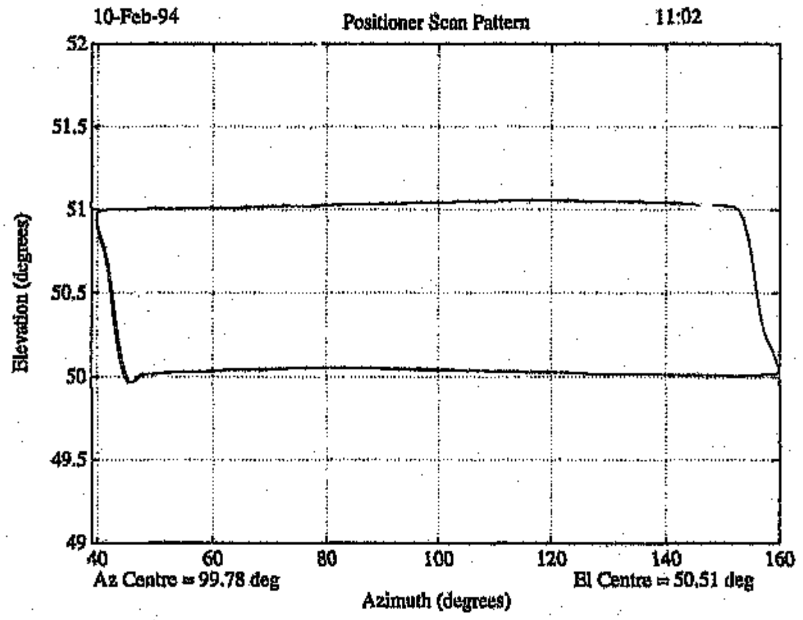


Figure 7.7 Simple Box Scan

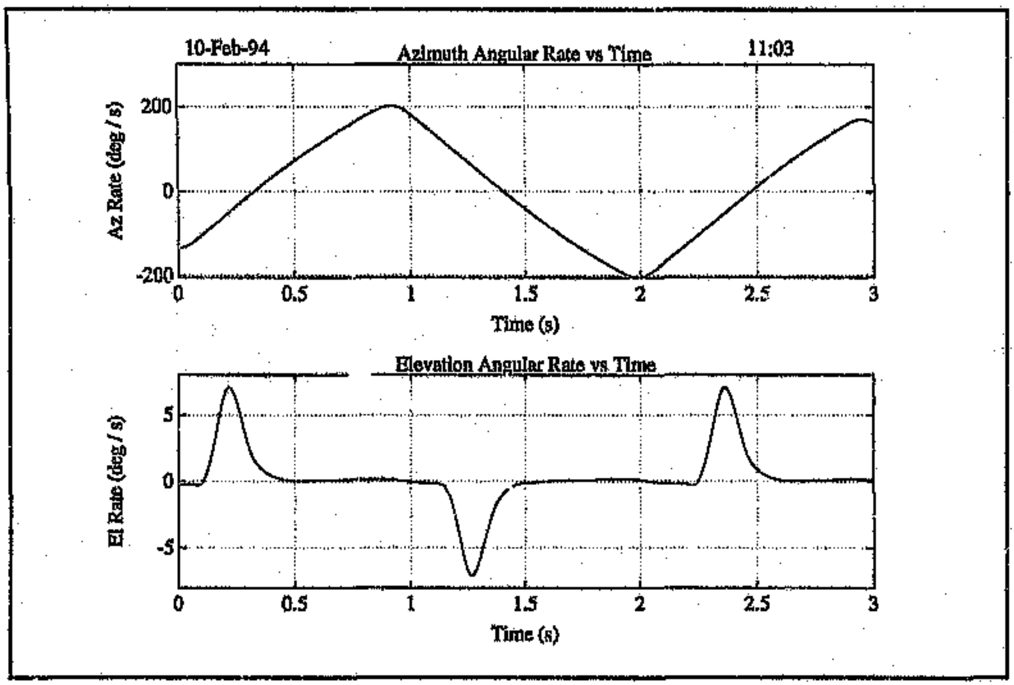


Figure 7.8 Angular Rates During Scan

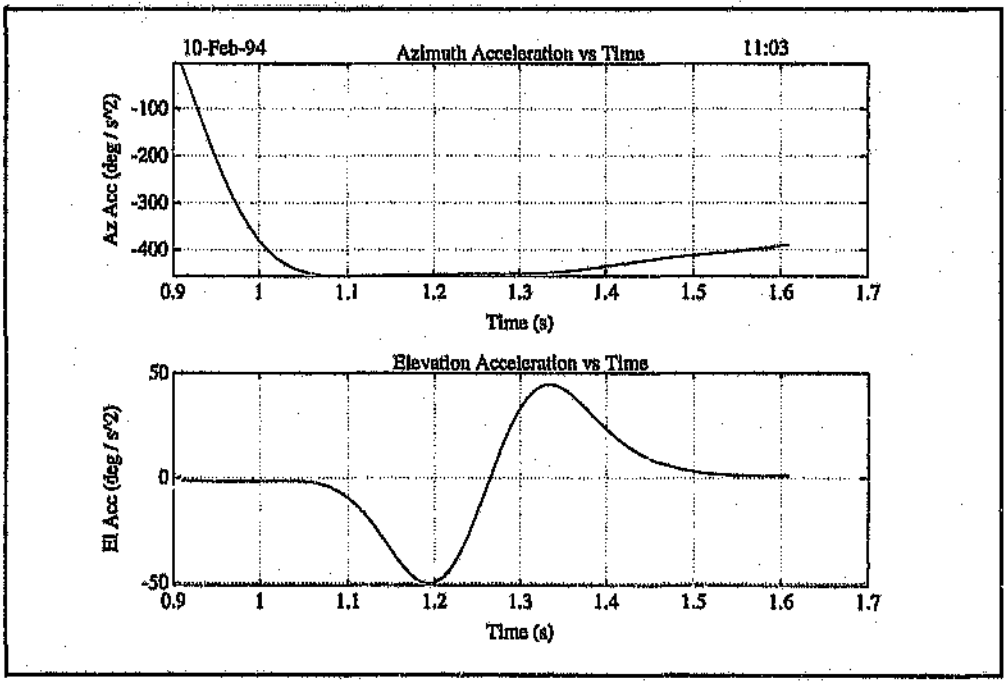


Figure 7.9 Zoomed Plots of Angular Acceleration versus Time

7.3.5 Calibration Constant on Small Sphere

The graph for this function is one simple plot of the calculated factor over the duration of the measurement window. The key value is the mean, which is used for RCS calculations. This measurement was made with a near maximum signal level, hence the apparent lack of fluctuation, however, a look at the scale of the *y*-axis will indicate that this is not the case.

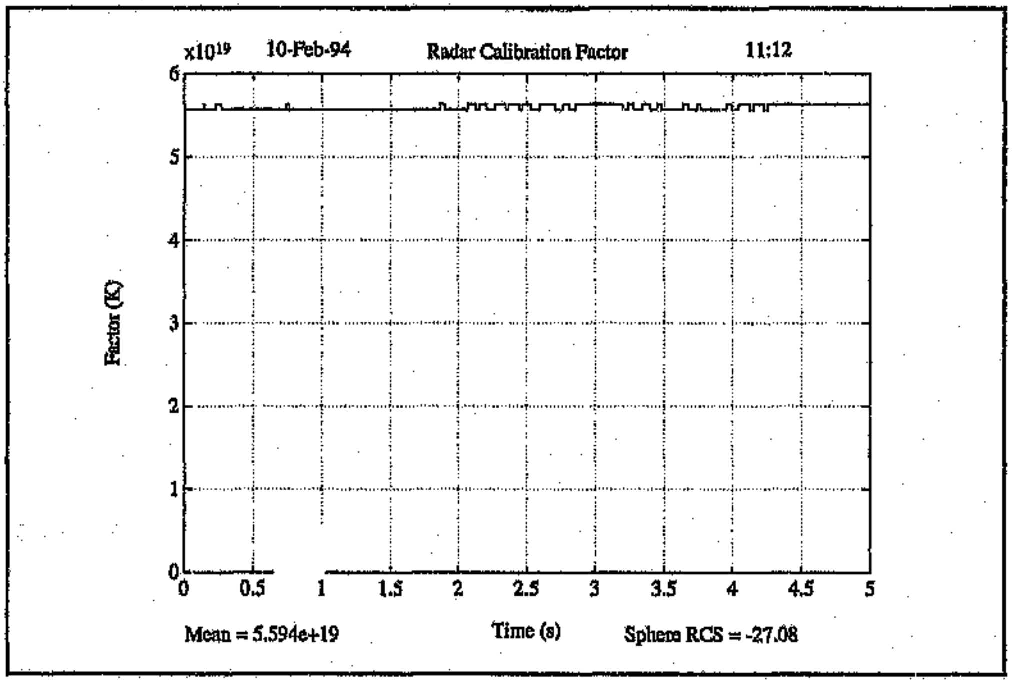


Figure 7.10 Radar Calibration Factor

7.4 Weather Equipment Interface Functions

The interface to the driver for the weather equipment was developed in Borland C++. This driver is linked into the HSDA software.

7.5 Time Code Interface Functions

An interface to the third party interrupt driven driver was developed for the time code reader. This driver is linked into other code as necessary.

7.6 Data Filter Board and FIFO Capture Board Control Functions

Some of the more critical functions were developed in Turbo Assembler; the bulk of the development took place in Borland C++. The drivers for the RDO card suite are linked into the HSDA and MSDA code.

CHAPTER 8

CONCLUSIONS

8.1 Results Achieved

The DAPS system has been installed in the radar system mobile laboratory for two years. During this period it has been successfully used on numerous field trials to capture high speed data for RCS, clutter and detailed performance analysis purposes. The MSDA functions have served to perform acceptance test procedures on the radar at each location, in order to validate the performance for measurement purposes. In addition some of the functions have served to assist with the experiment and high speed data capture process.

The success can be measured by the fact that external customers are satisfied with the results achieved, and are prepared to contract Aerotek to be at tests with the radar on a continuous basis.

8.2 Application Potential

So far as the technical aspects pertinent to data capture, and the application of the hardware developed here are concerned, there are a number of points to note:

- The technology and techniques used here have changed during the evolution of the project. The current data storage process can be made more efficient by using either DMA transfer, or special 80386/80486 instructions (such as INSW) which capture data even faster than standard ISA bus DMA.
- With the other two possible techniques in mind, and three other data capture applications being undertaken, there is a next generation card being designed for future use. This card uses the same FIFO storage to buffer data, but will allow

data transfer by any of three techniques; namely port I/O (as used here), DMA or repeat in-string (REP INSW) statements specific to the processors mentioned above.

- Much of the high level processing and storage functions are to be kept, and only the card interface modules need replacing.

8.3 Project Experience

This project has served as an exercise in system specification, embedded hardware design and software specification and design. Software specification was an exercise in standardised specification using the real-time extensions of data flow analyses.

The experience gained during the execution of this project has been used on other Aerotek projects of similar nature.

8.4 Quality Management in Software Development

CSIR has made a decision to pursue ISO9000 [8,9] accreditation. Aerotek is fully committed to a total quality management program and achievement of the accreditation. Accreditation per se is not a guarantee of quality, but should represent the ethic behind the quality management system which demonstrates and should measure the quality.

With regard to software development this has involved investigation of the total development life-cycle. The effort put into a formal methodology during specification of the DAPS system has ensured that the user requirement and design are well documented and understood. Some of the elements lacking during the development of the software were proper version control, inspection, testing, traceability and pursuing of documentation throughout these phases.

The current development method commences with a formal user requirement. This is followed by an SRS (Software Requirement Specification), very similar in form to the IEEE model used here. The design process is supported by a formal design document. Coding is according to an agreed formal coding standard, with appropriate documentation either in the code or as a separate document. The software is subject to testing according to an acceptance and test plan (ATP). At all stages inspection and review are undertaken to ensure the correctness, validity and suitability of the work. Software always forms part of a larger system. There is a standard numbering system for all components and sub-sytems in a system. This ensures traceability, and is applied to the software components; documentation, code and executables. The entire design process is subject to configuration and version control. This ensures that changes are carefully and appropriately applied throughout a project.

It is envisaged that commitment to quality will become a fundamental part of the CSIR way of doing business. A good software quality management system will support the work ethic.

At present the formal design tool which has been selected to assist with software development is a Unix based version of Cadre's Teamwork and other tools.

The selection of a documentation and version control system has not yet been finalised. Extensive investigation into the GNU RCS [10] version control system has demonstrated, as is often the case, that a freeware package is able to meet the requirements at present. A decision has been taken to move to the Polytron Version Control System (PVCS) [11] and associated products in future.

REFERENCES

[1] IEEE Standards Board, "IEEE Guide to Software Requirement Specificaᴜons", ANSI/IEEE Std 830-1984, September 1983.

[2] Hatley D J and Pirbhai I A, "Strategies for Real-Time System Specification", Dorset House Publishing, 1988.

[3] IBM Corpuration, "IBM Technical Reference: Personal Computer AT", IBM Part 1502243, 1984.

[4] Barton D K, "Modern Radar System Analysis", Artech House Inc., 1988.

[5] Eaves J L and Ready E K, "Principles of Modern Radar", Van Nostrand Reinhold Company, 1987.

[6] Texas Instruments, "The TTL Data Book for Design Engineers", Second Edition, Texas Instruments, 1981.

[7] Borland International, "Borland C++ 3.1 - User's Guide and other Documentation", Borland International, 1992.

[8] ISO 9001, Quality Assurance Standards, "Model for Quality Assurance in Design, Development, Production, Installation and Servicing", International Standards Organisation (ISO/TC176), 1987.

[9] ISO 9000-3, Quality Assurance Supplements, "Guidelines for the Applicatin of ISO 9001 to the Development, Supply and Maintenance of Software", International Standards Organisation (ISO/TC176), 1987.

[10] Tichy W L and Eggert P, "GNU Revision Control System (RCS) Version 5.6", under GNU General Public Licence to the Free Software Foundation, 1992.

[11] Sage Software, "Polytron Version Control System (PVCS) Version 3.3", Sage Software Inc, 1990.

APPENDIX A

DETAILED HARDWARE DESCRIPTION AND CIRCUIT DIAGRAMS

A.1 Hardware Description

A.1.1 Port Functional Description

As stated previously, four port addresses are used for the interface card. These are $300, $302, $304 and $306. The following sections summarise the functions of the ports and individual bits in the bytes driving those ports, where applicable.

Port $300

The function of this port is summarised by Table A.1.

Table A.1 Function of Port $300

Title	Data Port
Purpose	To read FIFO data (to PC)
Type	Read only
Width	16 bits
Individual Bit Descriptions	Entire width made up of a data parameter which is either a word (0 - 65 535) or an integer (-32 768 - 32 767)

Some of the data parameters are words and some are integers. Not all of the parameters use the full 16 bits, but are width and sign expanded to 16 bits. The type and sizing of individual parameters is a software function.

Port $302

The function of this port is summarised by Table A.2.

Table A.2 Function of Port $302

Title	Status Port	
Purpose	Monitoring of FIFO status lines	
Type	Read only	
Width	16 bits	
Individual Bit Descriptions	Low Order Byte	Connected to the Full Flags (active low) of the FIFO chips. Bits are paired off with the high and low order FIFO chips making up a data word. Bits 8 and 7 monitor the chip which is loaded first after a reset condition. Bits 6 and 5 monitor the next load and so on
	High Order Byte	Connected to the Empty Flags (active low) of the FIFO chips. Bit connections are the same as for the low order byte

If the FIFO is filled from a reset condition, then bits will toggle separately in the high and low order bytes from the most to least significant positions. Under normal operation (not immediately following a reset) the order of flag indications will depend upon which chips are full and empty. The interpretation and use of the flags is a software function.

Port $304

The function of this port is summarised by Table A.3.

Table A.3 Function of Port $304

Title	FIFO Control Port	
Purpose	Control of FIFO for reset, GPIO write enable and selection of counted data or continuous modes	
Type	Write only	
Width	8 bits, making up low order byte of word at address	
Individual Bit Descriptions	1 (LSB)	GPIO Write Enable. Drives PCTL line of GPIO bus through buffer. Signals to DAPS when to start/stop transmitting data (active low)
	2	GPIO Write Enable indicator LED (active low)
	3	Enables write strobe to clock counters at each write for implementation of counted data modes (active low)
	4	Enables counters to control write strobe to FIFO from GPIO, for implementation of counted data modes. A full count of 65 535 on counters then disables further writes (active low)
	5	FIFO Reset lines (active low)
	6	Reset indicator LED (active low)
	7	Not used
	8 (MSB)	Not used

The individual bit positions are accessed by performing AND and OR operations on the byte to be written to the control port, as dictated by the necessity to lower or raise a line respectively. The function of bits 3 and 4 ·will become clearer when considered in conjunction with some of the bits of port $306 as follows.

Port $306

The function of this port is summarised by Table A.4.

Table A.4 Function of Port $306

Title	Counter Control and FIFO Data Selection Port	
Purpose	Control of counters for clocking, reset and enable; as well as selection of GPIO, or counters or neither as FIFO input	
Type	Write only	
Width	8 bits, making up low order byte of word at address	
Individual Bit Descriptions	1 (LSB)	Counter Clear lines (active low)
	2	Counter output clock, to transfer latest count to counter output registers (rising edge trigger)
	3	Count register clock, OR'ed with write strobe as second mechanism for counting under control of software (rising edge trigger)
	4	Test indicator LED (active low)
	5	Enables write strobes from GPIO or counters to FIFO write lines (active low)
	6	In conjunction with bit 7 provides selection of FIFO data input. Operation is described in text below
	7	In conjunction with bit 6 provides selection of FIFO data input. Operation is described in text below
	8 (MSB)	Enable to decoder controlling access of counters and GPIO buffers to FIFO input (active low)

Once again, the individual bit positions are accessed by performing AND and OR operations on the byte to be written to the control port, as dictated by the necessity to lower or raise a line respectively. Bits 1, 2, 3 and 5 are used in conjunction with bits 3 and 4 of port $304, as has been described. Only certain combinations for bits 6 and 7 are valid, bo.h bits being used in conjunction to generate selection of the FIFO input. The valid 2 bit numbers with bit 7 as MSB are binary 10 and 11. These numbers select the counters and GPIO respectively as FIFO input. Usage of the port is, once again, a software function.

A.1.2 Data Modes

The only function on the card that requires explanation is that of the data modes. The workings of the rest of the card and the ports are relatively easily understood from the diagram. All functions will become clear in the software definition and description reports which are to follow. The actual implementation of any of these modes is the subject of software reports.

Continuous Data Mode

The counters can be completely isolated from the FIFO input, reset, and with the clock input isolated they will then exert no control over the GPIO buffers or write strobe to the FIFO chips. In this mode data reception is controlled solely by enabling of the GPIO buffers (port $306, bits 6 and 7), and delivery of a PCTL low to the GPIO bus (port $304, bit 1). Data will stream continuously until the PCTL line is raised again, whereupon the current set (frame) of data parameters will be sent to completion before termination of the transmission. The buffers should not be disabled before the transmission is complete, as integrity of the last frame is not assured if this happens. Clearly the status lines (port $302) should be monitored and data read from the FIFO fast enough to ensure that the FIFO does not overflow. T1e practical implementation of this mode and the limitations will emerge in the software design.

Counted Data Mode

With the internal count-registers (port $306 bit 1) and write strobe access (port $304 bit 3) to the counters enabled; and with the "feedback" path from the counter carry to the GPIO buffers and FIFO write enable open (port $304 bit 4); the counters will disable access to the FIFO inputs when a carry is generated at the highest count of 65 535. Thus if a calculated initial value is set in the counter registers, by means of the test channel, a predetermined number of samples can be loaded into the FIFO. Clearly it is a software function to check that no more than 16 384 samples are requested, and that the initial value is corrected to allow only values which will complete a frame of data parameters.

Test Mode

If the counters are selected as FIFO input (port $306 bits 6, 7 and 8) and the counter internal registers and output registers are clocked (port $306 bits 2 and 3), the FIFO can be filled with any desirable counter values, the counters can be set to a value and a complete test can be conducted on the card to establish that it is functioning correctly. The implementation of this test mode will be the subject of software reports.

A.2 Circuit Diagrams and Detailed Description

A survey of single-chip FIFO's available led to CMOS, 9 bit, parallel FIFO devices by Integrated Devices Technology, Inc. This range of components allows for depth and width expansion of the FIFO, with control functions such as full and empty indication. The range includes four chips, namely IDT7201S/L, IDT7202S/L, IDT7203S/L and IDT7204S/L. These correspond to 512, 1024, 2048 and 4096 by 9 FIFOs respectively. In line with the decision to implement a 16k FIFO, and the data rate, it was decided that the IDT7204S/L would be used, width expanded to 18 bits (only 16 used) and depth expanded to 16 384 values. The specification sheet of the 120 nanosecond, 24-pin DIP device, FIFO that was used here is readily available in the High Speed Memory Devices data book of the manufacturers.

The circuit diagram of the interface card appears in Figures A.1 and A.2. The circuit was built on a BICC-VERO, IBM AT Speedwire Interface Board with interface circuitry; part number 244-53101F. The interface circuitry of this board is the IBM AT recommended prototype interface.

Figure A.1: FIFO Circuit Diagram, Sheet 1

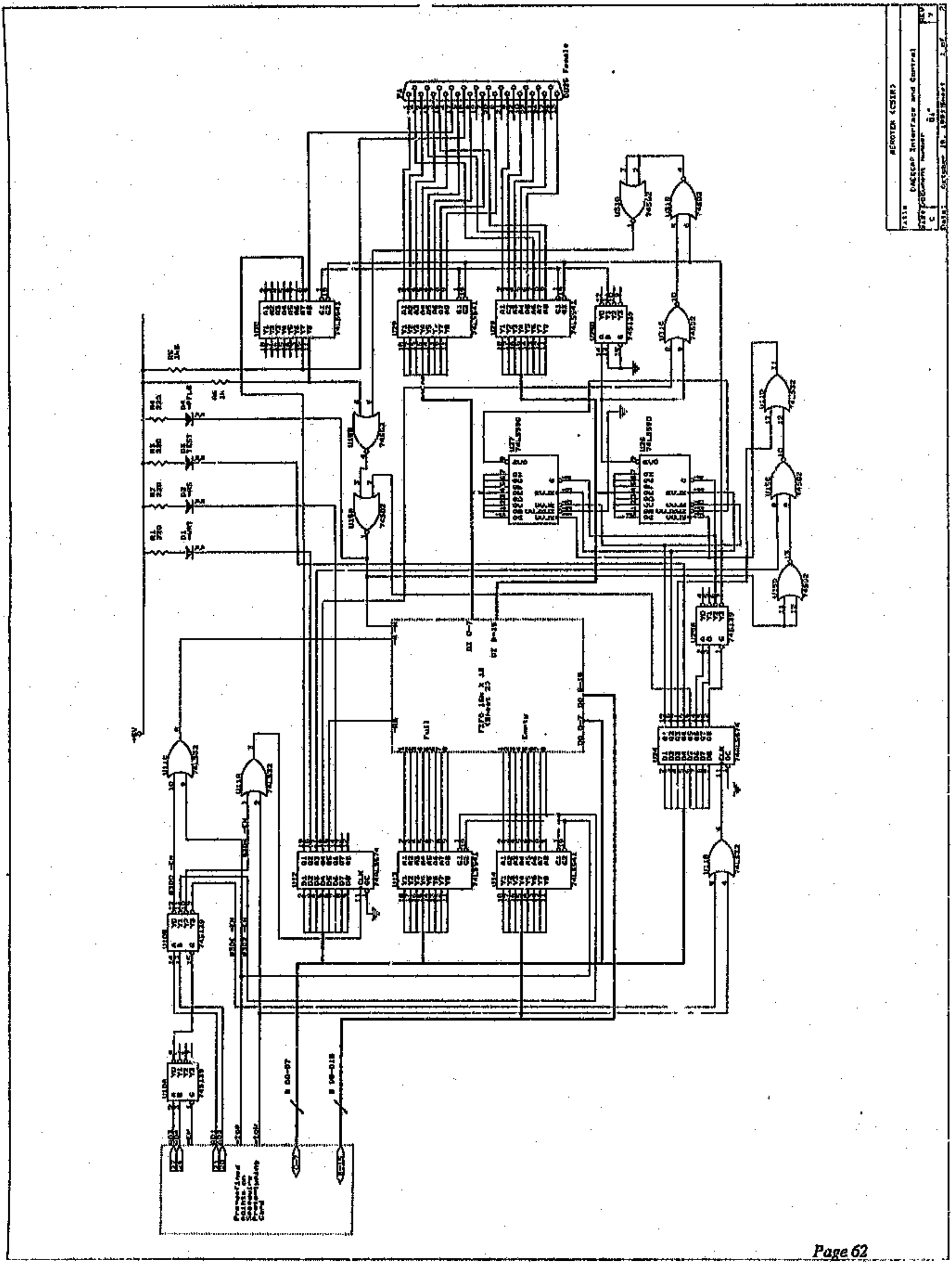
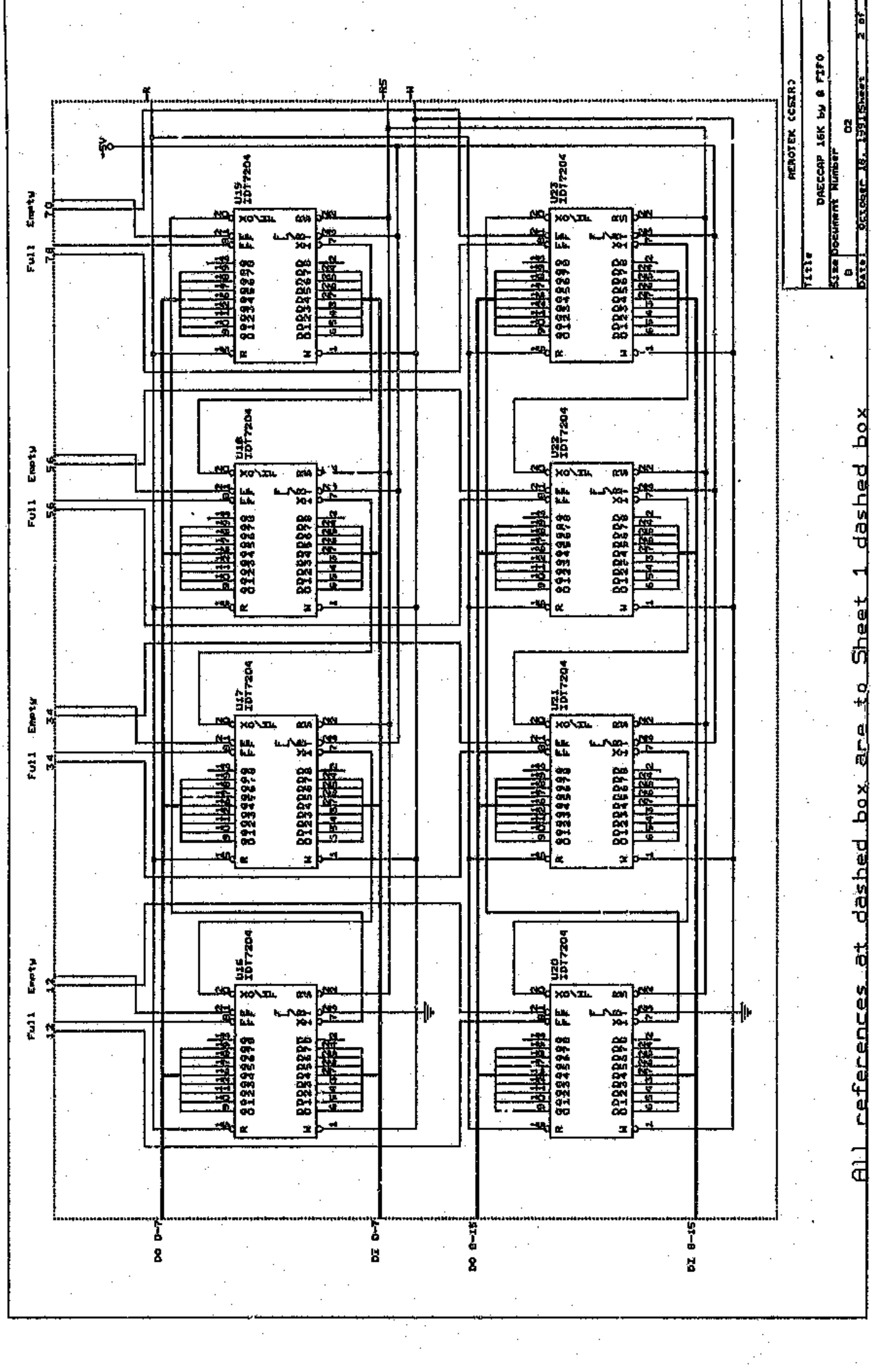


Figure A.2: FIFO Circuit Diagram, Sheet 2



Component numbering on the circuit diagrams takes cognizance of the numbering on the card and starts where the numbering on the bare interface card stops. The description below refers to the first sheet of the circuit diagram.

U10 serves to decode some of the prototype address space, namely $300, $302, $304 and $306. The interface card is set-up to function with 16 bit ports. This forces the use of only even addresses, hence the selection of the listed addresses. The functions of the ports have been described. U11A,B and C take care of I/O read and write instructions to the four addresses used on the card. U12 and U24 are latches which are used for control signals to the FIFO chips, data bus and counters. U13 and U14 are buffers for the FIFO status information. The interface of the card to the PC bus is formed by the chips mentioned so far and the data lines of the PC bus.

The four diodes; D1, D2, D3 and D4; are indicators used for low level indication and debugging of the card. D1 is an indicator that the input is open to receive data from the DAPS. D2 indicates a reset signal to the FIFO chips. D3 is used to indicate boot-up initialisation and testing of the card. D4 is a monitor of the input clock, ie. the data clock of the GPIO type bus.

U28 and U29 are buffers used to protect and isolate the input lines of the FIFO chips from the GPIO data lines. U30 performs the same task on the control lines. U26 and U27 are counters which are used for counting into the FIFO during testing and counting of incoming data in the mode where the amount of data captured does not exceed the capacity of the FIFO. U25A is used to allow a choice between either the counters or the GPIO data as input to the FIFO. U15D, U15C and U11D are used to control the counters. U25B, U31A, U31B and U31C implement hardware control of the GPIO data and control buffers to allow for the counted mode.

U15A and U15B are used for the FIFO write lines. They allow writing to take place, under the control of the PC, from the counters or the GPIO data lines.

The following description refers to the second sheet of the circuit diagram.

U16 - U19 are the FIFO chips used for the LSB of the data word. U20 - U23 form the MSB of the data word. These chips are used in the standard width and depth expansion modes that are detailed in the documentation on the devices. The resulting FIFO is 18 bits wide, although only 16 are used, and 16 384 words deep.

APPENDIX B

DETAILED SOFTWARE SPECIFICATION

The process, data flow, control flow and storage items on data flow diagrams used in specification of the DAPS were expanded to full descriptions of the flows and process specifications for each of the processes. For the sake of brevity not all details are included here. High level functional description is given of most items, and detailed specification of some of the processing options.

B.1 HSDA System Functions

The HSDA functions deal with both interface cards. Some control the data filter board, and others are implemented on the FIFO capture board. These functions are called either to execute HSDA functions, or prepare the data filter for MSDA functions.

The following data flow diagram identifies the processes which implement the HSDA system functions.

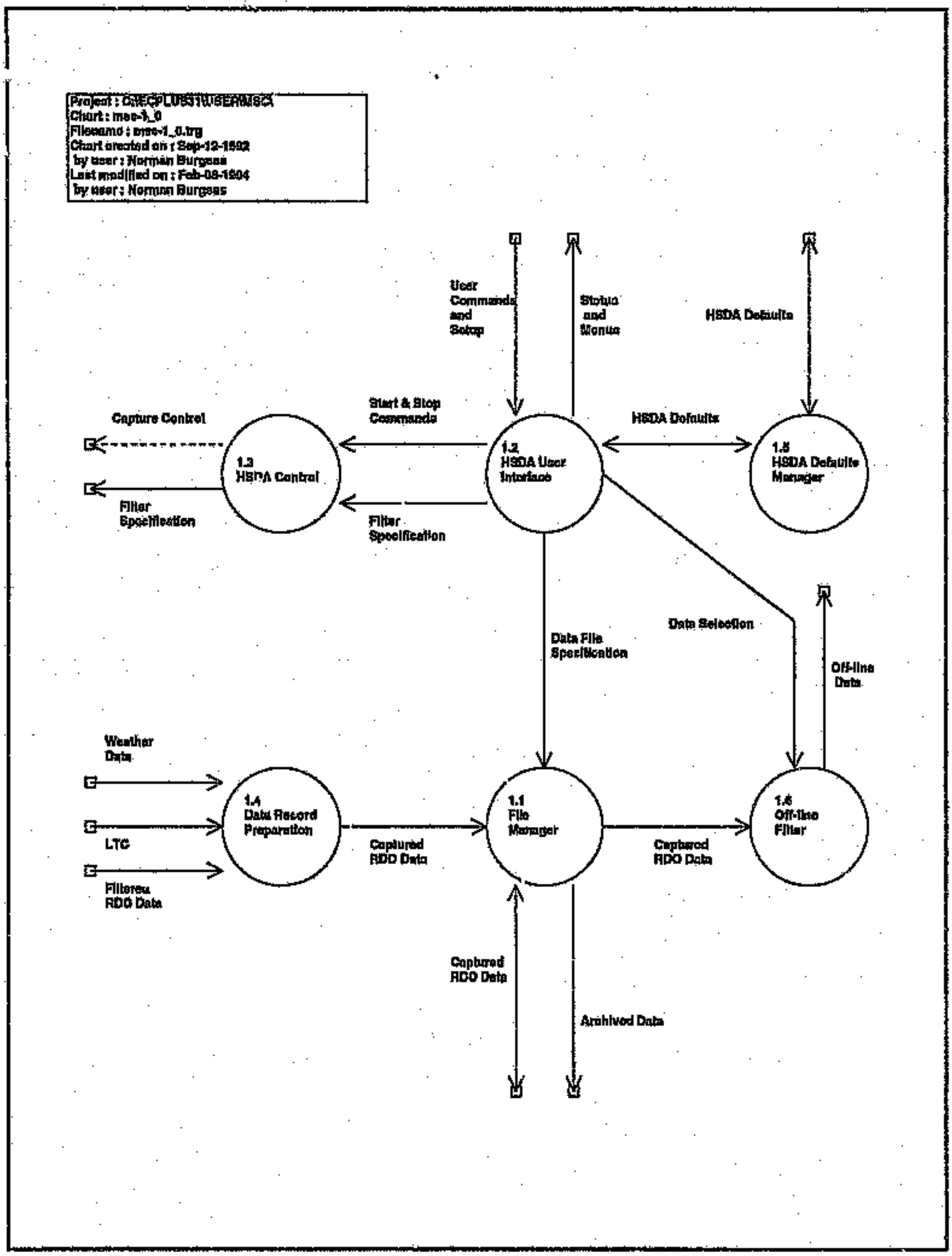


Figure B.1 HSDA Data Flow Diagram

B.1.1 File Manager

Introduction

This process manages the creation, storage and retrieval of RDO data files in which HSDA data is placed. Files are created during recordings. They are retrieved during filtering for off-line processing, or to be transferred for archival purposes.

Inputs

Data File Specification
Captured RDO Data

Outputs

Captured RDO Data
Archived Data

B.1.2 HSDA User Interface

Introduction

This process is the MMI to the HSDA system. The user is able to enter (and manage storage and retrieval of) the setup of the capture filters, data files, data filtering and system defaults. This process also serves as the control interface for starting and stopping the HSDA capture process.

Inputs

User Commands and Setup
HSDA Defaults

Outputs

Status and Menus
Start & Stop Commands
Filter Specification

Data File Specification
Data Specification
HSDA Defaults

B.1.3 HSDA Control

Introduction

This process delivers the filter specification and control commands to the FIFO interface, which controls the hardware. The process translates the user inputs (in logical understandable form) to the low level commands which are passed to the interface.

Inputs

Start & Stop Commands
Filter Specification

Outputs

Capture Control
Filter Specification

B.1.4 Data Record Preparation

Introduction

Data is collected simultaneously from three sources. This process organises the interleaving of the data and preparation of records for storage. The data is passed to a common file manager which is able to read and write RDO data files.

Inputs

Weather Data
LTC
Filtered RDO Data

Outputs

Data File Specification
Data Specification
HSDA Defaults

B.1.3 HSDA Control

Introduction

This process delivers the filter specification and control commands to the FIFO interface, which controls the hardware. The process translates the user inputs (in logical understandable form) to the low level commands which are passed to the interface.

Inputs

Start & Stop Commands
Filter Specification

Outputs

Capture Control
Filter Specification

B.1.4 Data Record Preparation

Introduction

Data is collected simultaneously from three sources. This process organises the interleaving of the data and preparation of records for storage. The data is passed to a common file manager which is able to read and write RDO data files.

Inputs

Weather Data
LTC
Filtered RDO Data

Outputs

Captured RDO Data

B.1.5 HSDA Defaults Manager

Introduction

This process interfaces the MMI process to the System Defaults store. It stores and retrieves HSDA defaults for re-use.

Inputs

HSDA Defaults

Outputs

HSDA Defaults

B.1.6 Off-Line Filter

Introduction

This process performs the function of filtering and converting specific parameters, in specified amounts, from the raw captured data. The data is passed out in any of a number of forms suitable for import into other environments.

Inputs

Data Selection
Captured RDO Data

Outputs

Off-line Data

B.2 MSDA Capture and Analysis Functions

The high level analysis functions described here achieve the major aim of this project. These functions are used for radar evaluation and calibration. The presentation of these higher level functions is slightly different from the lower level functions to facilitate better understanding of their purpose.

Each of the following sub-functions is implemented by first capturing a time-slice of RDO data (semi-real time processing). After this the data is processed according to the algorithm and displayed as indicated. Re-processing of the same data with different constant settings and hard copy are options which must always be available.

The exact form of displayed graphs was developed and evaluated as software was developed. For this reason no pictorial example is included in the specification. There are example outputs in Chapter 7. The data parameter selection descriptions are made with reference to Chapter 5.

The following data flow diagram identifies the processes which implement the MSDA system functions.

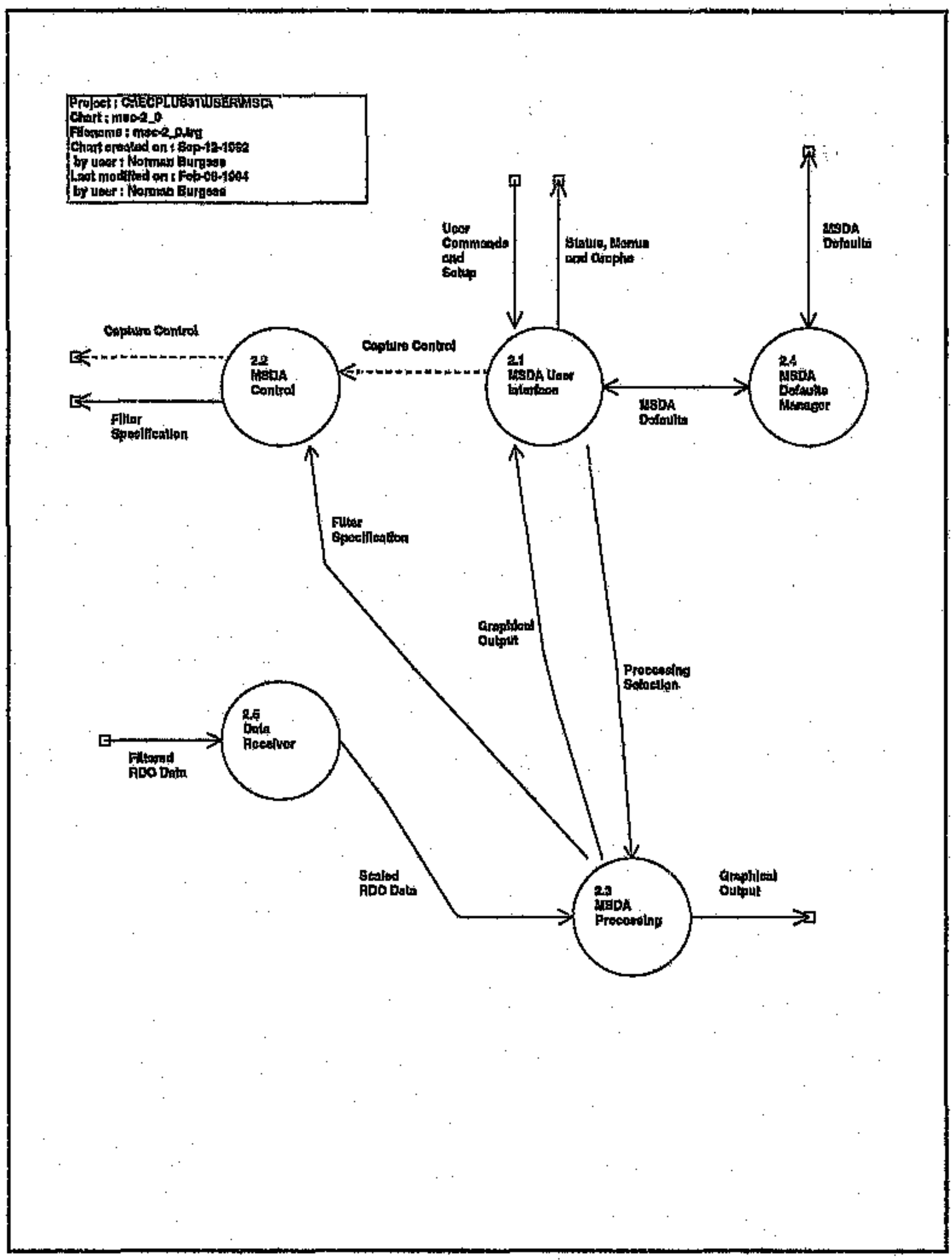


Figure B.2 MSDA Data Flow Diagram

B.2.1 MSDA User Interface

Introduction

This process is the MMI to the MSDA system. The user is able to enter (and manage storage and retrieval of) the setup of processing options and system defaults. This process also serves as the control interface for starting and stopping the MSDA capture process. The user processing options drive the MSDA process to automatically select (prepare the filter for) RDO data. The MMI also manages the display of graphs resulting from processing.

Inputs

User Commands and Setup
Graphical Output
MSDA Defaults

Outputs

Processing Selection
Status, Menus and Graphs
Capture Control
MSDA Defaults

B.2.2 MSDA Control

Introduction

This process is very similar to the HSDA control function. One essential difference is that this process automatically terminates the capture process when sufficient data has been collected for the process selection currently active.

Inputs

Capture Control
Filter Specification

Outputs

Capture Control

Filter Specification

B.2.3 MSDA Processing

Introduction

This process includes the algorithms of all of the processing and display options. The inputs and outputs are generically listed below. After this the initial set of processing options are specified in more detail. Depending upon the processing option selected, a filter specification is passed to the MSDA control process. There are a number of terms used to describe forms of processing and display. These are:

Constant/s Refers to the constants used when executing the capture and measurement functions. Examples are the number of samples, time-constant for smoothing and default plot settings. In all cases these should default to a value which is only re-specified when specifically accessed, and not at each measurement opportunity.

dBq The standard decibel scale, but referenced to one quantisation level for voltage measurements and one quantisation level squared for power measurements.

Magnified In the case of data output, this is a graphical plot where the axes system is scaled so that the maximum and minimum of the data just fit on the axes system.

Parameter/s This refers to the digital number which emerges from the A/D convertors in the I and Q channels of the radar system. These integer values are samples of the voltages in the channels.

Quantisation Level A value of 1 from an A/D convertor is referred to as one quantisation level. ie. This is the resolution of the A/D system, and the data acquisition process.

Zero-referenced In the case of data output, this is a graphical plot where the axes system has zero and some convenient maximum displayed.

Inputs

Scaled RDO Data

Processing Selection

Outputs

Filter Specification

Graphical Output

B.2.3.1 Radar Improvement Factor Measurement

Purpose

To calculate the MTI improvement factor of the radar. The improvement factor is often limited by noise and instabilities in the system. It is desirable to determine the improvement factor of the system to analyse current noise levels and stability, and to verify the radar's detection performance in a clutter environment.

Measurement Procedure

The radar is locked on to a strong clutter source, such as a corner reflector. In order for the measurement and calculation to be valid, it is essential that the signal to noise ratio (thermal, environmental noise) exceeds 50 dB, and this has to be established at the time of processing. RDO data is captured for a specified length of time. The MTI improvement factor is calculated and displayed.

Variations

Two of the distinct noise types in the radar are transmitted pulse amplitude noise and phase reference noise, both of which will lower the improvement factor. In order to further investigate cases where the measurement indicates a poor improvement factor, it is necessary to re-perform the calculation and display the independent results after removal of the phase noise and amplitude noise.

Inputs

Valid parameter inputs are the I and Q sum-channel components in quantisation levels.

Constant inputs are the data collection time (which determines the number of samples); radar scan rate (which determines the pulse integration process); a selection between a zero-referenced, or magnified plot; and a choice of normal, phased normalised or magnitude normalised variations.

The table below details the specification on the constants and the default values:

Constant	Range	Default
Data collection time (*T*)	0,5 to 5 seconds	1
Scan rate	40, 80, 160 and 320 degrees per second	80
Plot type	zero-referenced or magnified	zero-referenced
Thermal noise level	0 to 10 dBq	3
Variation	normal, phase normalised or magnitude normalised	normal

Outputs

An *x*-*y* plot of filtered, MTI improvement factor in dB (*y*) versus sample number.

Numeric indications of the mean improvement factor, mean IF phase and magnitude contributions (dB); signal to noise ratio (dB); and the absolute signal level (dBq). These means are calculated over the extent of the window.

Algorithm

The number of samples in a given window is *n*, which is calculated from the *T* as follows:

$$n = 2560 \cdot T$$

Given n - I and Q parameter samples from the sum channel of the radar RDO stream the improvement factor is calculated as follows.

First the double pulse MTI cancellor is implemented on each of the channels to obtain I_m and Q_m:

$$I_m(i) = I(i-2) - 2I(i-1) + I(i) \quad \text{and}$$

$$Q_m(i) = Q(i-2) - 2Q(i-1) + Q(i)$$

for $i \in (3,n)$ with $I_m(1) = I_m(2) = I_m(3)$ and $Q_m(1) = Q_m(2) = Q_m(3)$

In order to evaluate the effect of the radar integration, it is necessary to calculate the moving average of the improvement factor for the number of integrated samples, which is determined from the scan rate.

Scan rate (deg/s)	Number of integrated pulses (n_i)
40	88
80	44
160	22
320	11

The improvement factor I_m for a measurement on clutter is defined as:

$$I_m = G \cdot C_R$$

where C_R is the clutter rejection ratio (output clutter power to input clutter power), assuming that the noise is negligible. The improvement factor is calculated by multiplying this by the average MTI gain for a double pulse cancellor ($G = 6$).

To calculate the average improvement factor for any size windows, the relevant clutter powers must be averaged before division, so that:

$$\overline{C}_R = \frac{\frac{1}{n}\sum_{1}^{n}(I_m^2(i) + Q_m^2(i))}{\frac{1}{n}\sum_{1}^{n}(I^2(i) + Q^2(i))}$$

For the graphical plot, the value of n (to calculate each plotted value is n_l with the previous $n_l - 1$ values being used at the i-th position. For the entire data set n is n, which will deliver the mean improvement factor.

To calculate the normalised magnitude and phase means, as well as the graph values when either of these variations is selected, the data is processed as follows. For phase normalisation (which refers to the case where only magnitude fluctuations are significant), the phase differences are removed from the data set. First the complex magnitude and phase vector is calculated for each I and Q sample:

$$z(i) = m(i)e^{\theta(i)} \quad \text{where} \quad m(i) = \sqrt{I^2(i) + Q^2(i)} \quad \text{and} \quad \tan(\theta(i)) = \frac{Q(i)}{I(i)}$$

The phase, $\theta(i)$, is normalised to 45 degrees (to avoid numerical computation problems), whereafter the I and Q channel signals are extracted as follows:

$$I(i) = \Re(z(i)) \quad \text{and} \quad Q(i) = \Im(z(i))$$

The improvement factor and associated mean are recalculated from the start using this new data. To process the data with magnitude normalisation a similar process is followed:

$$z(i) = m(i)e^{\theta(i)} \quad \text{where} \quad m(i) = \sqrt{I^2(i) + Q^2(i)} \quad \text{and} \quad \tan(\theta(i)) = \frac{Q(i)}{I(i)}$$

The magnitude is normalised to 1. The I and Q channel signals are then extracted as above:

$$I(i) = \Re(z(i)) \quad \text{and} \quad Q(i) = \Im(z(i))$$

The mean absolute signal level is simply the mean of the input power over the window.

$$\overline{C}_I = \frac{1}{n}\sum_{1}^{n}(I^2(i) + Q^2(i))$$

Exceptions

This calculation is only valid for the parameters indicated.

B.2.3.2 Sum, Azimuth and Elevation I and Q channel Characteristics

Purpose

It is necessary to be able to view graphical plots of the *I* and *Q* channel signals of each of the three channels of the radar. These must be available as separate voltage plots, or combined power plots. Means and variances of the signals must also be calculated. In addition it is desirable to view the spectrum of the signal over the duration of the window.

Measurement Procedure

While the radar is tracking a target of interest, RDO data is captured for a specified length of time. The data is then processed and displayed.

Variations

The user requirement is to be able to view voltage plot-couples of the *I* and *Q* channel information in each of the difference channels, or the some channel. Alternate selections are a combined power plot of all three channels and the spectrum of the sum channel information.

Inputs

Valid parameter inputs are the I and Q components of the sum and two difference channels in quantisation levels.

Constant inputs are the data collection time (which determines the number of samples); a selection between a zero-referenced, or magnified plot; and a choice of one of five plots, namely, sum channel, elevation difference channel, azimuth difference channel, combined power and spectrum.

The table below details the specification on the constants and the default values:

Constant	Range	Default
Data collection time (*T*)	0,5 to 5 seconds	1
Plot type	zero-referenced or magnified	zero-referenced
Variation	sum, elevation, azimuth, power and spectrum	sum

Outputs

x-y plots, corresponding to the selections, with numeric indicators of means and variances.

In the case of voltage plots, two *x-y* plots of the voltages (in quantisation levels) are made, one above the other, on the screen. Numeric indicators of the average and variance of each of the six captured channels are displayed. A numeric indicator of the mean of the sum signal power is displayed (dBq).

The power plot comprises a plot of the power (dBq) in the sum and difference channels. A single plot is made, but the individual signals must be identifiable. Numeric indicators of the mean and variance of each signal are displayed.

The spectrum plot comprises an *x-y* plot of the magnitude of the complex frequency spectrum of the target from $-0,5f_s$ to $0,5f_s$, where f_s is the sampling frequency (2560 Hz).

Algorithm

For the voltage plot selections, the only processing per se is the extraction of means and variances, where the normal definitions apply. If there are *n* samples of variable *x* then:

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i \quad \text{and} \quad \sigma^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

The mean power of the signal in the sum channel is calculated as follows:

$$\bar{S}=\frac{1}{n}\sum_{i=1}^{n}(I^2(i)+Q^2(i))$$

For power plot selections the power is calculated for each of the sampled channels (sum, azimuth difference and elevation difference) at each sampled interval. These arrays are plotted. The means and variances are calculated and displayed, using the above definitions:

$$S_{az}(i)=I_{az}^2(i)+Q_{az}^2(i) \quad \text{and similarly for} \quad S_{el} \quad \text{and} \quad S_{sum}$$

The frequency spectrum is calculated using the *I* and *Q* channel signals as the input to a complex FFT calculation after windowing with a Hamming window, a standard function for reducing the . The number of samples has to be reduced to the nearest power of 2 below *n* (alternatively the sample size selected can be a power of 2) before the calculation is performed. The windowing function which is performed on each channel separately is defined below; where $w(i)$ is the weight applied to each sample in the set:

$$w(i)=0.54+0.46\cos\left(\frac{2\pi i}{n}-\pi\right)$$

Of primary concern is the magnitude of the frequency spectrum and this is plotted as a default. It is often desirable to smooth the frequency spectrum and re-plot it. This is a user selected option.

Exceptions

This calculation is only valid for the parameters indicated.

B.2.3.3 Antenna Pattern Measurement

Purpose

Antenna patterns and particularly the null depths of the difference channels are subject to change when the antenna is mounted onto the radar positioner. This makes it necessary to obtain a measure of antenna characteristics with the antenna mounted. Ideally this would be a complete 3D measurement over a solid angle of 2 beamwidths in cross-section. In practice this requires special scan patterns in the positioner control loops. Thus at this stage a reduced measurement is taken.

Measurement Procedure

The radar is locked onto the test target generator (TTG). With the necessary RDO data being captured, independent azimuth and elevation offsets are induced on the antenna with the thumb-wheels. This ensures that a cross-section is obtained in both the *E*-plane and the *H*-plane through the null of the antenna's monopulse radiation pattern.

Variations

There are three possible outputs for this function. Firstly, plots of the sum channel antenna gain along both sections. Secondly, the difference channel sections along both sections. The third plot results from a computation of the monopulse error function and the slope through the boresight of the antenna.

Inputs

Valid parameter inputs are the I and Q components of the sum and two difference channels in quantisation levels. Also necessary are the RPG azimuth and elevation co-ordinates.

Constant inputs are the data collection time (which determines the number of samples) and a choice of one of three plots, namely, sum channel beam patterns along both sections, difference channel beam patterns and the monopulse error function.

The table below details the specification on the constants and the default values:

Constant	Range	Default
Data collection time (T)	5 to 60 seconds	20
Variation	sum, difference and error slope	error slope

Outputs

x-y plots of the antenna beam patterns covering 1,5 degrees either side of the boresight of the antenna (this allows the first sidelobe to be measured), corresponding to the selections. Indicators of the measured 3 dB points of the antenna beam patterns and the amplitude of the first sidelobe. In all cases, beam patterns are normalised with 0 dB being the gain of the sum pattern at the boresight of the antenna.

Algorithm

The collection process means that the data is received in a disorganised fashion. There may be several cuts along each of the primary axes. The central tracking position at the start of the measurement determines the antenna boresight.

Firstly, the data is organised and the received amplitude (voltage) in quantisation levels is calculated for each sample in each of the received channels:

$$V_{sum}(i) = \sqrt{I_{sum}^2(i) + Q_{sum}^2(i)} \quad \text{and similarly for} \quad V_{az} \quad \text{and} \quad V_{el}$$

Secondly the received voltage levels are averaged at each RPG position. ie. All of the points with the same RPG co-ordinates are averaged. This produces a cross-sectional cut along each of the axes in each of the three channels. At each point $(\theta_{az}(i), \theta_{el}(i))$, there may be $n_{\theta(i)}$ data samples from the independent excursions during the measurement process. The voltage for each point is then calculated as follows:

$$\overline{V}_{sum}(i) = \frac{1}{n_{\theta(i)}} \sum_{k=1}^{n_{\theta(i)}} V_{sum}(\theta(i), k) \quad \text{and similarly for} \quad \overline{V}_{el}i \quad \text{and} \quad \overline{V}_{az}$$

The phasing of the antenna 'makes' the voltages in the difference channels negative on one side of the boresight and positive on the other. Conventionally, these are taken as negative to the left and below boresight and positive to the right and above. The data sets in each of these channels is negated for azimuth and elevation angle positions less than the measured boresight position. This data is now ready for plotting.

The monopulse error function in each of the directions is calculated by dividing the difference channel signal by the sum channel signal at each measured point. So:

$$E_{az}(i) = \overline{V}_{az}\frac{(i)}{\overline{V}_{sum}(i)} \quad \text{and similarly for} \quad E_{el}$$

The error slope at bor[illegible]ained by numerically differentiating the error function at 7 points sp[illegible]ually [illegible]d the boresight.

Exceptions

This calculation is only valid for the parameters indicated.

B.2.3.4 RPG Scan Pattern and Acceleration Measurement

Purpose

One of the important factors in determination of overall radar system performance is the radar positioner execution of scan patterns during acquisition processes, and the associated accelerations and decelerations. It would also be desirable to quantify stiction, but this is not possible at present.

Measurement Procedure

The radar is pointed in some arbitrary direction away from the intended scan volume. With the necessary RDO data being captured, the command to scan a particular pattern of interest is given. The positioner will slew around and execute the scan. After this the data collection process is terminated manually.

Variations

The first plot is a plot of the entire slew and scan. The next plot is a plot of the scan pattern over a user defined range of azimuth and elevation angles. A third plot is of the positioner angular acceleration and velocity during the execution of the scan, in both the azimuth and elevation directions.

Inputs

Valid parameter inputs are the azimuth and elevation co-ordinates of the RPG data set.

Constant inputs are the user defined region over which to plot data; and a choice between the default plot of all the data, or a zoomed plot of the user defined region. The user can also select to view the angular acceleration and velocity plot for each of the co-ordinate directions and along the direction of travel. ie. There is a choice between position, velocity or acceleration plots.

The table below details the specification on the constants and the default values:

Constant	Range	Default
User defined azimuth and elevation start and stop positions	Any range of the data actually collected	entire range
Variation	total or zoomed plot and position, velocity or acceleration plot.	total

Outputs

x-y plots of the antenna position, velocity or acceleration during the execution of the scan in either zoomed or total form.

Algorithm

The position information is inherently available from the data set. The velocity and acceleration of the positioner are cal lated by numeric differentiation of the data.

The data sets can then be plotted.

Exceptions

This calculation is only valid for the parameters indicated.

B.2.3.5 Calibration Constant Measurement on Small Sphere

Purpose

The radar has been used successfully in the past to measure RCS of many targets. These measurements require accurate calibration against a sphere. A frequent calibration verification exercise is that of calibration using a small 6" sphere of known RCS (0,01824 m^2, --17,39 dB). The purpose is to determine the calibration constant (*K*) of the radar which can be used to determine RCS from the received power in quantisation levels squared.

Measurement Procedure

The radar is locked onto the calibration sphere. Data is collected for the default time, and the calculation performed.

Variations

There is only one output for this calculation, namely a plot of the calibration constant *K*.

Inputs

Valid parameter inputs are the I and Q components of the sum channel, and the range

Constant inputs are the data collection time (which determines the number of samples).

The table below details the specification on the constants and the default values:

Constant	Range	Default
Data collection time (*T*)	0.5 to 10 seconds	5

Outputs

An x-y plots of the received signal level, and the calibration factor at each point. Indicators of the average *K* over the duration of the measurement.

Algorithm

The small calibration sphere is used at extremely close ranges, which allows the atmospheric attenuation to be ignored.

The received signal power (*S* in dBq) is calculated as for the power measurements. The calibration factor is defined at each point as:

$$K(i) = \frac{R^4(i)S}{\sigma}$$

where *R* is the range, and σ is the RCS (0,001824). The average value is calculated in the standard way:

$$\overline{K} = \frac{1}{n}\sum_{k=1}^{n} K(i)$$

The data can now be plotted.

Exceptions

This calculation is only valid for the parameters indicated.

B.2.3.6 Flight Control Console

Purpose

This is one of the more complex user requirements. A display is required which allows for control of an aircraft RCS measurement experiment while it is underway. This must therefore include the aircraft trajectory and position in radar (graphics) and navigational (text) co-ordinates to allow for vectoring of aircraft by airband radio communications, the maximum and minimum pulse (sum channel video) levels and radar status information such as MTI, STC (sensitivity time control) and mode. The display update rate will not be real time, since rates are then not suitable for user assimilation of the data. Background capture and processing has to occur in real time.

In addition it is desirable to display a background map in bitmap form of the area around the radar underneath the above display. The bitmap is supplied by the user for each deployment environment.

At the end of any particular session, the screen has to be saved or printed, and re-initialised for fresh tracks.

Measurement Procedure

During initialisation, the correct bitmap is loaded and a fresh co-ordinate system displayed. Once initialised there are four user commands which will start, stop, print or reset the display of current target parameters which are collected, processed, averaged and displayed.

Variations

For maximum performance at closer ranges, a zoom feature is required. Upon zoom, the existing track is re-displayed as determined by the boundaries of the zoomed display.

Inputs

The target parameters of range, azimuth, elevation and respective rates are collected at 10 ms intervals (this is the actual update rate of these parameters from the radar's monopulse video processor. Sum I and Q are collected at pri rates. The mode and status parameters are also collected at pri rates.

Additional constant inputs are summarised below.

Constant	Range	Default
Bitmap	Valid bitmap file name	Most recent selection
Zoom state	1, 2, 5, 10, 20 kilometre maximum range	20
Logging state	On/Off	On

Outputs

A graphic screen plot of trajectory in plan position form (PPI) is drawn over the bitmap of the region. Text indicators of the current aircraft radar co-ordinates, and navigational co-ordinates (as computed from the radar co-ordinates) are displayed alongside. A vertical bar graph of signal level with maximum and minimum indicators are plotted at the side of the display, along with the computed RCS. A text display of key status information.

Algorithm

The measurements of target position and rates occur in radar co-ordinates. Radar co-ordinate systems are unique in that they are not strictly polar or cylindrical as defined mathematically. The algorithm is presented with reference to the diagram below.

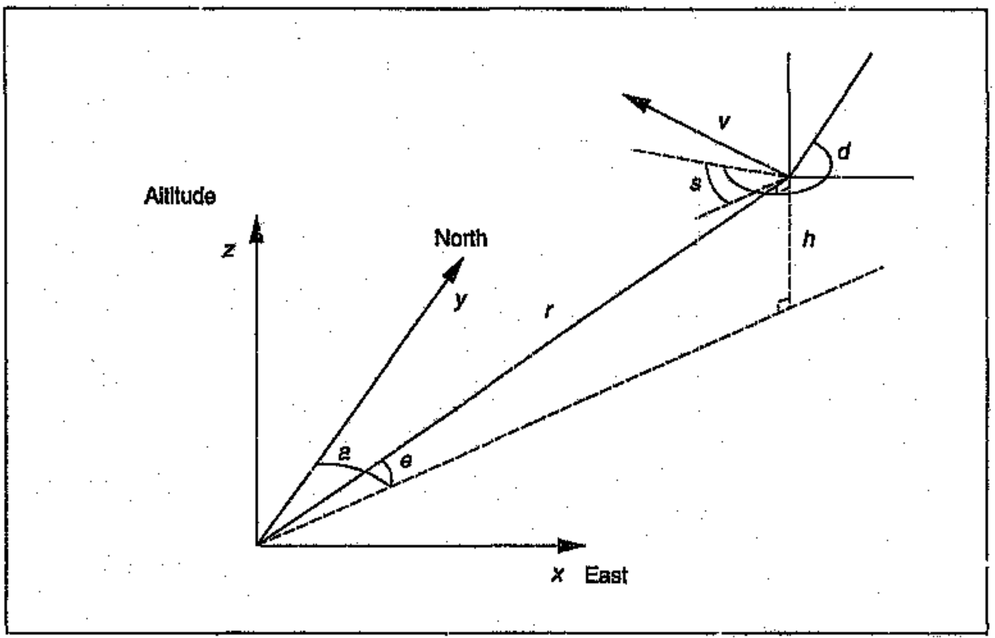


Figure B.3 Radar Co-ordinate System and Transformations

The target positional parameters collected every 10 ms are azimuth (a, θ) [illegible]on (e, ϕ) and range (r) and the associated rates of change. From this the height above ground (h), heading (d, α), aspect angle (s, β) and velocity (speed v) need to be calculated. The speed of the aircraft can be calculated directly from the measured co-ordinates as follows:

$$|v| = \sqrt{\dot{r}^2 + r^2(\dot{\theta}^2 + \dot{\phi}^2)}$$

The radar co-ordinates and rates are transformed to rectangular co-ordinates which can translate to the aircraft position. These can be used to calculate the 3D target angular parameters.

$$x = r\cos\phi\sin\theta$$

$$y = r\cos\phi\cos\theta$$

$$z = r\sin\phi$$

The height above ground, h, is equal to z. The rectangular co-ordinate rates are given by:

$$\dot{z} = \dot{r}\sin\phi + r\dot{\phi}\cos\phi$$

$$\dot{x} = ((\dot{r}\cos\phi - r\dot{\phi}\sin\phi)\sin\theta + \dot{\theta}\cos\theta$$

$$\dot{y} = ((\dot{r}\cos\phi - r\dot{\phi}\sin\phi)\cos\theta - \dot{\theta}\sin\theta$$

The heading (from the aircraft perspective) and horizontal aspect angle are then:

$$\alpha = arctan\left(\frac{\dot{x}}{\dot{y}}\right)$$

$$\beta = \alpha - (180^\circ + \theta)$$

From the above calculated parameters, a PPI type plot of the track of the aircraft can be plotted graphically. The navigational parameters are displayed in text form alongside.

The RCS calculation is the reverse of the calibration calculation presented earlier:

$$\sigma = \frac{R^4(i)S}{K}$$

where $S_{sum} = I_{sum}^2 + Q_{sum}^2$ as before

The RCS is displayed in text form. The maximum and minimum power levels are also displayed in text form alongside a bar graph with absolute scales to allow for continuous monitoring of the situation regarding noise and saturation levels in the radar system.

Finally, the mode and status words are monitored, and the bits indicating valid track (or lack thereof), MTI function (on or off), STC.

Exceptions

The radar data parameter selection for this function is automatic, and should not allow user intervention.

B.2.4 MSDA Defaults Manager

Introduction

Similar to the HSDA defaults manager this process allows for storage and retrieval of defaults for re-use.

Inputs

MSDA Defaults

Outputs

MSDA Defaults

B.2.5 Data Receiver

Introduction

This process receives RDO data from the interface to the hardware and scales it before passing it to the MSDA processing option currently selected. The scaling is predefined by the specification of the radar A/D convertors, shaft encoders and status word definitions.

Inputs

Filtered RDO Data

Outputs

Scaled RDO Data

B.3 Weather Equipment Interface Functions

The following data flow diagram identifies the processes which implement the weather equipment interface functions.

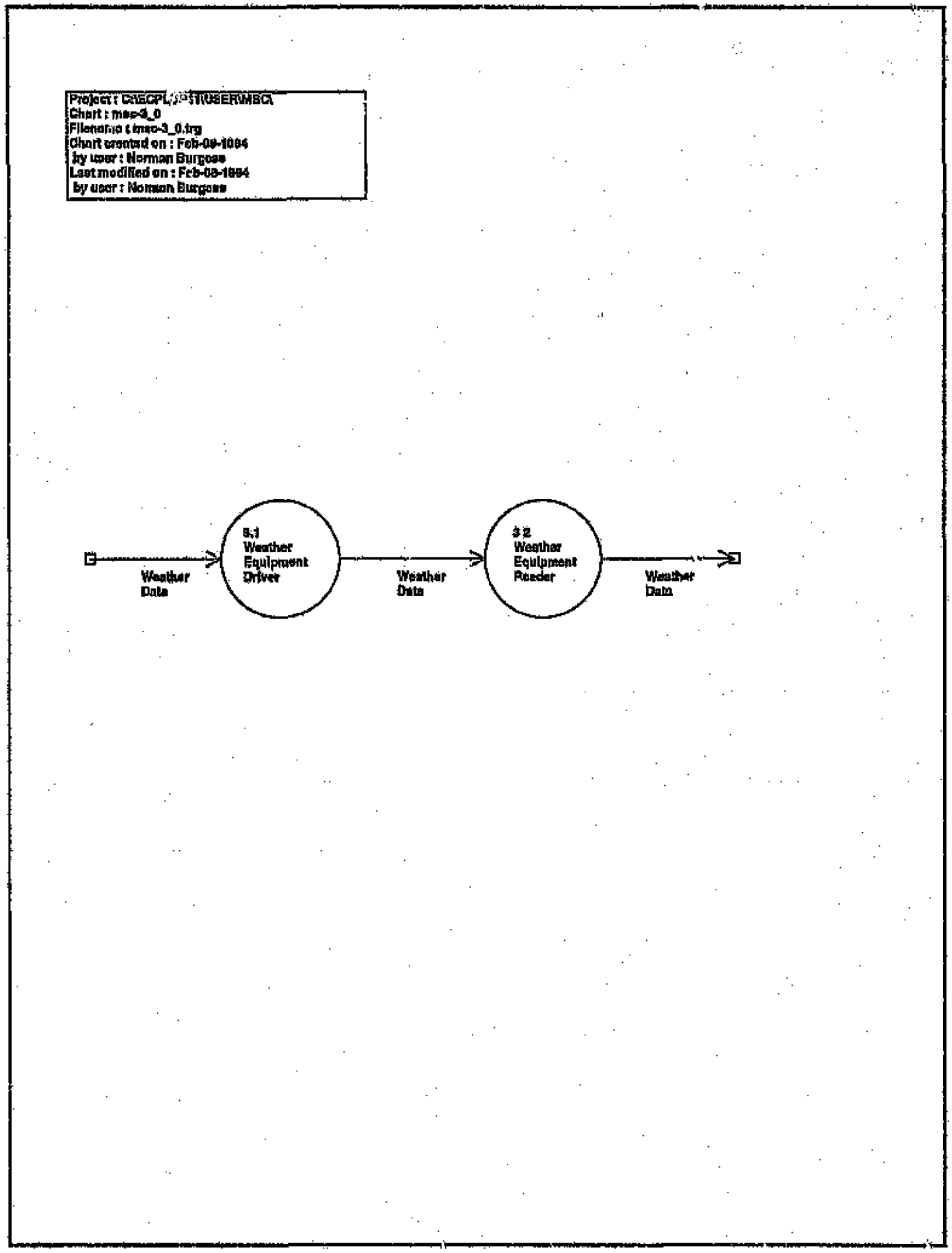

Project : CAECPL\[illegible]\USER\MSC\
Chart : map-3_0
Filename : map-3_0.trg
Chart created on : Feb-08-1994
by user : Norman Burgess
Last modified on : Feb-08-1994
by user : Norman Burgess
Weather Data
3.1 Weather Equipment Driver
Weather Data
3.2 Weather Equipment Reader
Weather Data

Figure B.4 Weather Equipment Interface Function DFD

B.3.1 Weather Equipment Driver

Introduction

The third party weather equipment is supplied with a driver which can be incorporated into user software. This driver interfaces to the weather equipment and allows for capturing of weather data.

Inputs

Weather Data

Outputs

Weather Data

B.3.2 Weather Equipment Reader

Introduction

This process interfaces the third party driver to the DAPS software, and allows for passing of weather data to the system.

Inputs

Weather Data

Outputs

Weather Data

B.4 Time Code Interface Functions

The following data flow diagram identifies the processes which implement the time code interface functions.

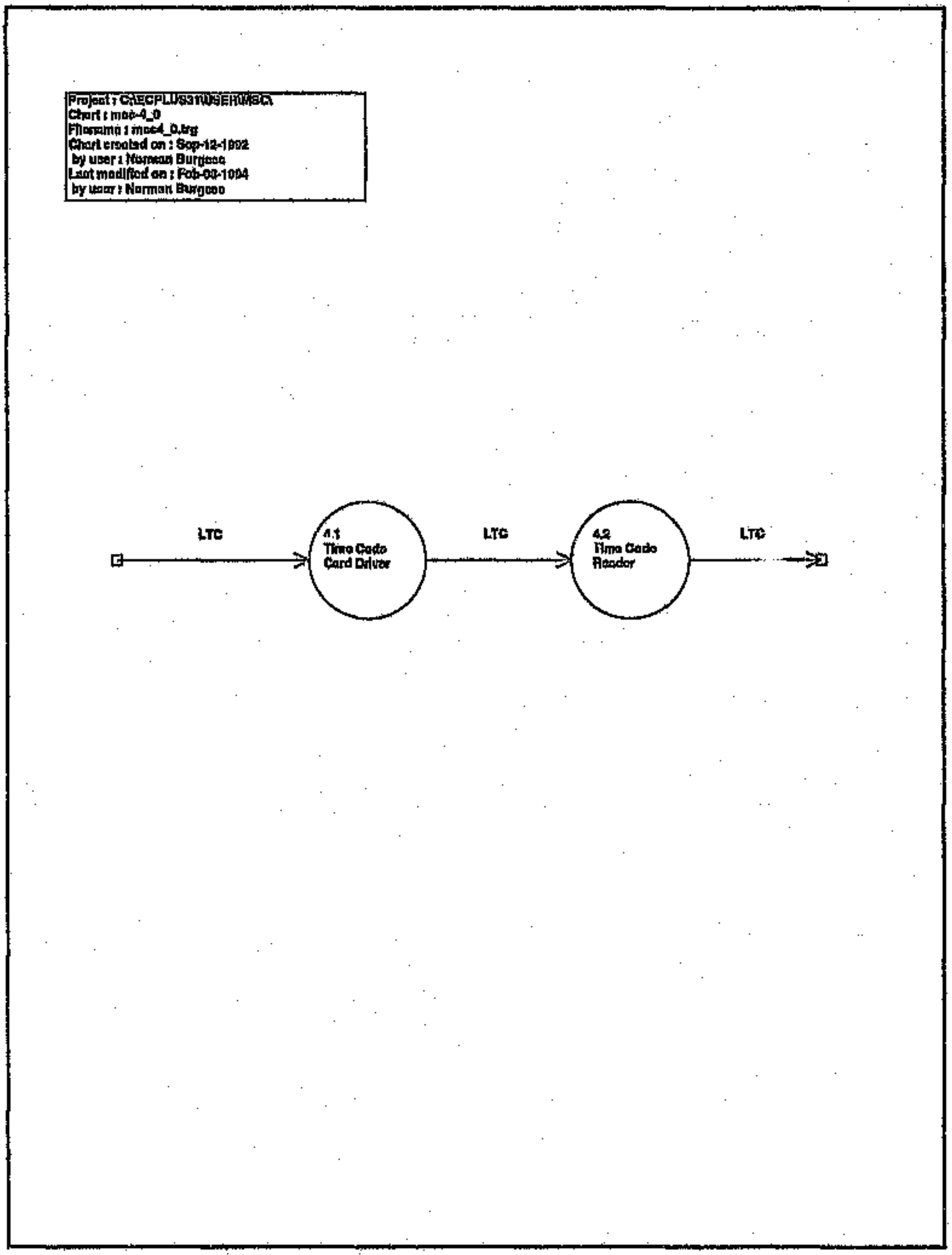


Figure B.5 Time Code Interface Function DFD

B.4.1 Time Code Card Driver

Introduction

The LTC card is supplied by an "in house" third party, along with driver software. This process represents that driver.

Inputs

LTC

Outputs

LTC

B.4.2 Time Code Reader

Introduction

This process interfaces the DAPS to the third party software, for delivery of LTC when required.

Inputs

LTC

Outputs

LTC

B.5 Data Filter Board and FIFO Capture Board Control Functions

The following data flow diagram identifies the processes which implement the data filter and FIFO control functions.

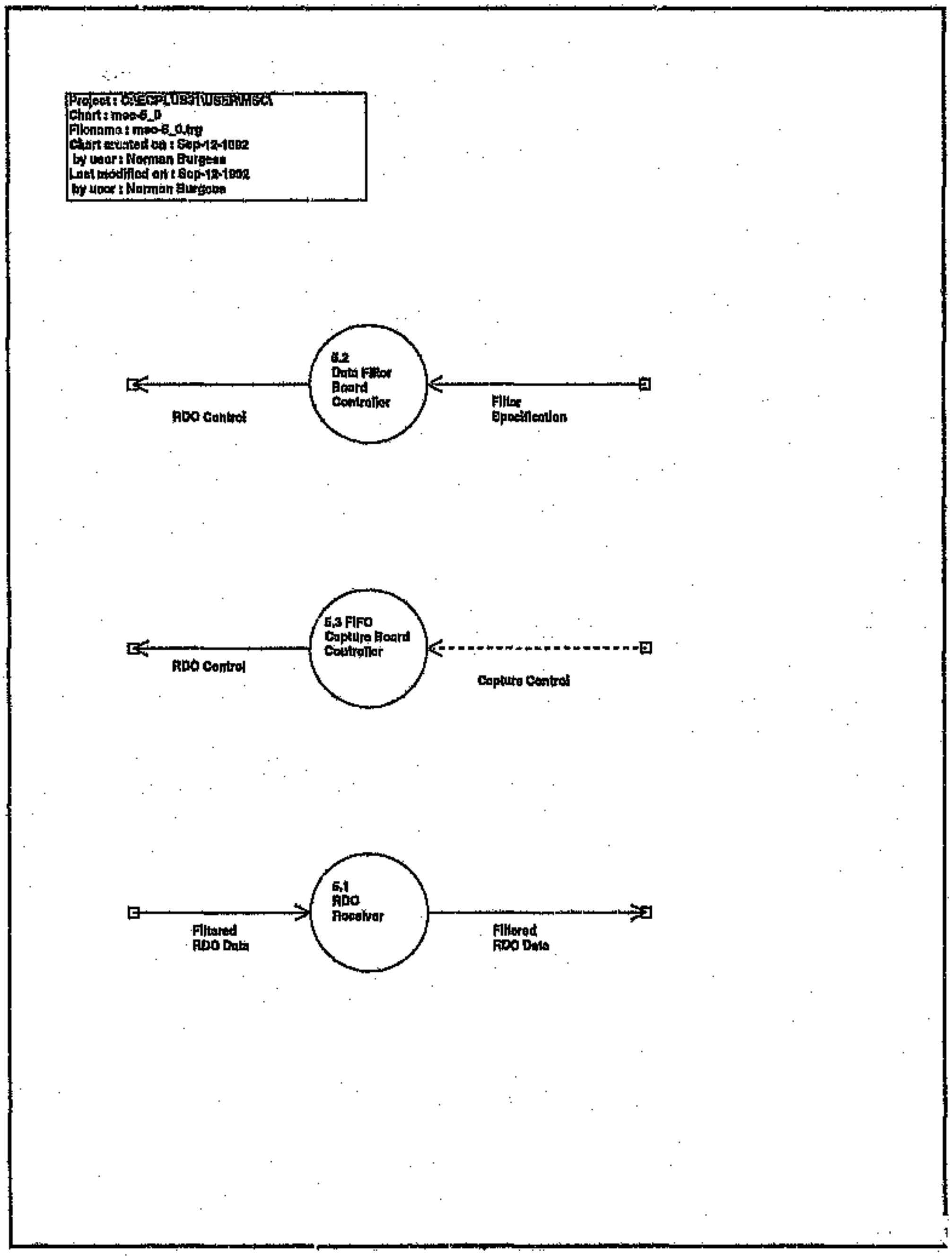


Figure B.6 Data Filter and FIFO Control DFD

B.5.1 RDO Receiver

Introduction

This process accepts RDO data from the FIFO's on the capture card and transfers them to RAM for use by other processes.

Inputs

Filtered RDO Data

Outputs

Filtered RDO

B.5.2 Data Filter Board Controller

Introduction

This process controls the data filter board. It performs the function of programming the registers with the data selection specified by the filter specification. This process is a low level driver for the filter board.

Inputs

Filter Specification

Outputs

RDO Control

B.5.3 FIFO Capture Board Controller

Introduction

This process controls the clearing, reset, starting and stopping of the transfer of data through the data filter to the FIFO capture card. This process is a low level driver for the FIFO card.

Inputs

Capture Control

Outputs

RDO Control

APPENDIX C

GLOSSARY OF TERMS

Data Acquisition and Processing System (DAPS) The current version of this equipment comprises a 50MHz 486 ruggedised PC clone processor card with 16MB of RAM; 1MB VGA a. pter, Super VGA display and standard I/O ports; Adaptec 1542B SCSI host adapter; 425MB, SCSI hard drive; 1GB, SCSI, Tahiti I Magneto Optical drive; combined 1.2MB/1.4MB, floppy/stiffy drive; high speed serial to parallel data filter card; FIFO capture interface card; time code reader card and a Postscript printer.

GPIO This is the Hewlett Packard term describing their General Parallel Input Output bus and its associated standards. This bus is described completely in their documentation. The implementation of the particular interface board of this project is described in more detail elsewhere. The interface used in this project is a subset of the full GPIO standard. The subset is really just a high speed, parallel, externally clocked data bus with data on/off control.

HSDA This is an abbreviation for High Speed Data Acquisition. In this report this refers to capture of all of the available radar data to a mass storage device.

MSDA This is an abbreviation for Medium Speed Data Acquisition. In this report this refers to capture of a subset of the available radar data.

Parameters The term parameters refers to individual data items, such as sum channel voltages, or antenna azimuth position. A complete set of each parameter selected is obtianed for each transmitted pulse of the radar. In the specification of functional processing of parameters, each parameter is conceptually an array containing pulse by pulse samples of the RDO data parameter of the same name, eg. *I*, where individual samples are referred to as *I(i)*. This would be the 1 channel voltage.

Radar Data Stream This refers to the digital data stream from which radar data is captured for processing. The stream is a pre-defined set of radar parameter samples passed to the capture and processing system by the radar monopulse video processor and controller. There are 60 word or integer components representing either radar target information (such as azimuth, range or signal amplitude), or radar status indicators and set points. In the existing radar system this data set is also called **RDO**.

Radar System In all occurrences, unless otherwise indicated the terms radar or radar system refer specifically to the experimental, monopulse, tracking radar system of Aerotek of the CSIR for which the data acquisition and processing system has been developed.

Real Time This term refers to the standard interpretation of the phrase, where the real time action occurs whilst and in pace with the constraining action of some piece of equipment.

Semi-real Time This term refers specifically in this document where data is captured in real time and analysis and display takes place within a maximum time of 5 minutes from the data capture.

Vectoring A term used by the SAAF to refer to guidance of aircraft under radar control. The radar operator gives course correction information to the pilot to assist him in traversing a specified path.