Design Improvement and Development of Avionic Systems for Flight Simulators

JIAN CHEN

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

April 2003

0-612-77697-2

ABSTRACT

Design Improvement and Development of Avionic Systems for

Flight Simulators

JIAN CHEN

Complicated navigation, costly inflight training, hazardous situations, accidents, and the inclination towards better quality of training have set the trend of novice pilot instructions to take place in flight simulators. One of the main components of such a flight simulator is the avionic system. In the Concordia Light Aircraft Simulator System, the avionic systems embody the radio stack and the associated navigational instruments. The contribution of this thesis is to document the enhancements of the Concordia flight simulator to function as a powerful academic tool, especially focusing on the design improvement and development of the radio stack system. The objective of this thesis is to upgrade the existing radio stack system. Essential for further developments, reverse engineering for the existing system is presented to understand its hardware configuration and operations. Instead of the common RS232 or parallel port, the multifunction data acquisition hardware made by National Instruments is used for the communication between the radio stack and the flight model computer. The VC++ and LabVIEW radio stack modules replace the existing assembly program to control and drive the upgraded radio stack system. A more sophisticated and versatile navigation model coded in VC++ is programmed for users to experience the radio navigation systems typically found on a general aviation aircraft.

ACKNOWLEDGMENTS

I wish to express my gratitude and deep appreciation to my supervisors, Dr. J. V. Svoboda and Dr. H. Hong, for their guidance and inspiration throughout my graduate studies.

I would like to express my gratitude to Mr. P. Lawn for his suggestion of this project and for providing guidance throughout the research.

The invaluable assistance of Mr. Gilles Huard in developing the custom electronic hardware and reverse engineering is greatly appreciated.

To my wife, YUN ZHANG

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

NOMENCLATURE

A	Amplitude
AC	Aircraft
ADF	Automatic Direction Finder
ATC XPNDR	Air Traffic Control Transponder
BCD	Binary Coded Data
BRG	VOR true bearing (degree)
C	Capacitance
CDI	Course Deviation Indicator
CFS	Canadian Flight Supplement
CNTL / CT	Control
COMM	Communication Transceiver
DIST	Slant range (NM)
DME	Distance Measuring Equipment
EAROM	Electrically Alterable Read Only Memory
f	Frequency
FIFO	First-In-First-Out
HSI	Horizontal Situation Indicator
ILS	Instrument Landing System
INCR	Iteration period
INTERVAL	Time between samples (2 seconds)
IRQ	Interrupt Request

$JITTER$	VOR bearing jitter amount (degree)
K	Multiplier factor (=0.28)
Lat	Latitude
Lon	Longitude
$MAGBRG$	The magnetic bearing for the indicator
MBR	Marker Beacon Receiver
MPU	Microprocessor
ms	Millisecond
N	NDB beacon location on Earth
NAV	Navigation Transceiver
NDB	Non-Directional Beacon
NM_FT	NM to Feet factor (=6076.1)
$OLDIST$	Old slant range (NM)
OBS	Omni-Bearing Selector
PNI	Pictorial Navigation Indicator
R	Resistance
R_{earth}	Radius of the Earth
RAM	Random-Access Memory
RAN	Aircraft range to tuned station
$RANDOM$	Random Number
$RNAV$	Area Navigation System
RMI	Radio Magnetic Indicator
S/E	Special Error

SH / SHIFT	Shift
ST	DME Station
STB	Strobed
T	Period
T_w	Pulse Width
UTC	Coordinated Universal Time
V	VOR station location on Earth
VELOCITY	Aircraft velocity (NM/iteration)
VH	Aircraft height above ground
VHF	Very High Frequency
VOR	Very High Frequency Omni-Range Receiver
VORDLN	VOR station magnetic declination (degree)
X	The Earth's X-axis
Y	The Earth's Y-axis
Z	The Earth's Z-axis
θ	Bearing from ground station to airborne receiver
θ_{ADF}	Relative bearing from aircraft to NDB station
Δt	Iteration period/Time increment

CHAPTER 1
INTRODUCTION

1.1. A Brief History of Aircraft Flight Simulation

From the Wrights to Concorde and the Space Shuttle - so much has happened in the first 100 years of powered flight. The importance of training has been realized since the inception of manned flight. As a flight training tool. flight simulator that simulates the flight characteristics of an aircraft could effectively prepare pilots for the task of flying.

1.1.1. Early History

In the early development, the emphasis was to control the motion of the simulator in response to disturbances. One of the first truly synthetic flight training devices was the Antoinette "Apprenticeship Barrel". The photograph shown in Figure 1.1.1-1 was published in 1910. It consisted of two half-sections of a barrel mounted and moved manually to represent the pitch and roll of an airplane. The prospective pilot sat in the top section of the device and was required to line up a reference bar with the horizon. The rough dynamic models and systems available at the time made the effectiveness of this and similar trainers debatable. [1.1]

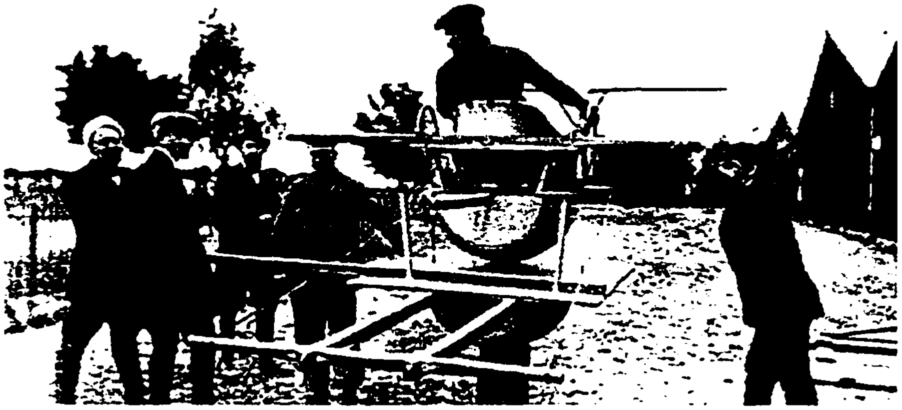


Figure 1.1.1-1 Apprenticeship Barrel [1.1]

The first description of the trainer made no reference to instruments and the device was therefore primarily intended to demonstrate to students the effects of the controls on the attitude of the simulated airplane and to train them in their operation. As with other synthetic devices of this time, the simulated effects of the ailerons, elevators and rudders were independent; they did not represent a true reproduction of the aircraft's coordinated behavior. Therefore, simulation was not seen as a substitute for actual flight. The acceptance of simulated flight as a useful training aid had to wait for further developments in the science of flying.

1.1.2. Instrument Flight Training

In the late 1920's, the need for the effective training of pilots in the skills of "blind" or instrument flying was starting to be felt. Two methods were developed: firstly, the existing moving trainers, such as the Links, were fitted with dummy instruments and the means for their actuation; secondly, non-movable devices were invented specifically for the task of instrument flight training. [1.2]

The Link Trainers were soon being fitted with instruments as standard equipment. Blind flying training was started by the Links at their flying school in the early 1930's and as the importance of this type of training was more fully appreciated, notably by the U.S. Army Air Corps, so the sales of Link Trainers increased. The newer Link Trainers were able to rotate through 360 degrees that allowed a magnetic compass to be installed, while the various instruments were operated either mechanically or pneumatically. Altitude, for example, was represented by the pressure of air in a tank directly connected to an altimeter. Rudder/aileron interaction was provided in the more advanced trainers, as was a stall feature. The reproduction of aircraft behavior and dynamics were produced in an empirical manner. [1.2]

The trainer shown in Figure 1.1.2-1 was being used to train pilots in the US Army. [1.3] It was an important advance in instrument flight training in that it enabled a closer match with the behavior of actual navigational aids. The visual system was based on a loop of film and simulated the effects of heading, pitch and roll movement. The trainer was widely demonstrated in the U.S., but was never commercially produced.

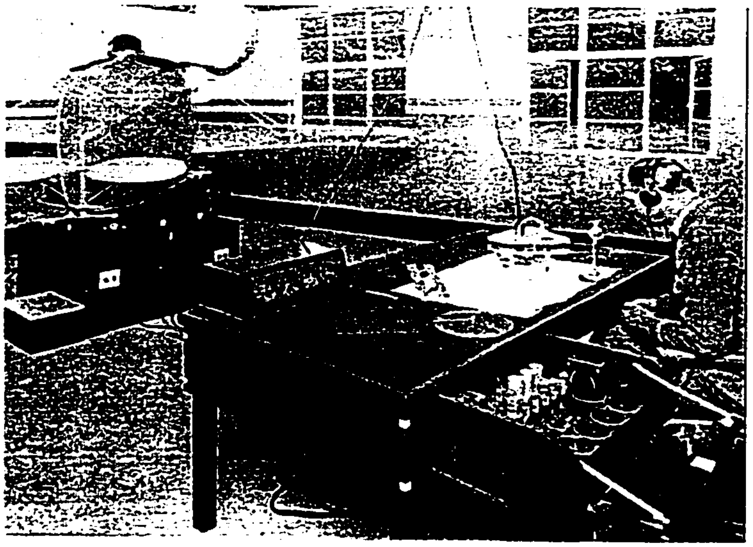


Figure 1.1.2-1 A Link Trainer [1.3]

1.1.3. World War II

At the start of the Second World War there was the requirement to train large numbers of people in the many individual and team skills involved in the operation of the various military aircraft. Developments in aircraft. such as variable pitch propellers. retractable undercarriage and higher speeds made training in cockpit drill essential. The mock-up fuselage was introduced as an aid to training in these procedures.

Of particular interest are the so-called Silloth Trainers, developed by Wing Commander Iles at RAF Silloth, south of Carlisle. Figure 1.1.3-2 shows one of these trainers for a Halifax bomber. The Silloth Trainer was designed for the training of all members of the crew, and was primarily a type familiarization trainer for learning drills and the handling of malfunctions. As well as the basic flying behaviors, all engines, electric and hydraulic systems were simulated. An instructor's panel, visible in the photo, was provided to enable monitoring of the crew and malfunction simulation. [1.2]

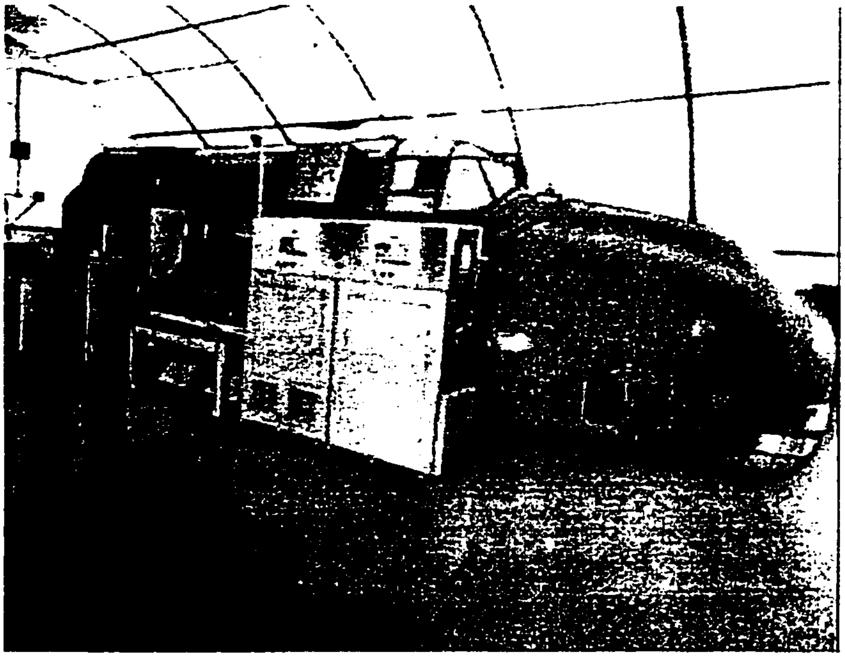


Figure 1.1.3-1 A Silloth Trainer [1.2]

In addition to the trainers mentioned, many other trainers were developed by adding extra features to the basic trainer for such tasks as gunnery instruction. The needs for training in more specialized skills were met by the adoption of a multitude of purpose-built devices.

1.1.4. Electronic Flight Simulation

A major advance in simulation during the war period was the use of the analogue computer to solve the equations of motion of the aircraft. The analogue computer enabled simulation of the response of the vehicle to aerodynamic forces as opposed to an empirical duplication of their effects. A number of these devices are the direct ancestors of the modern simulator.

The Z-1 trainer for the AT-6 airplane shown in Figure 1.1.4-1 was an example. It was built by Dr. Dehmel who had gained experience in analogue computing techniques through his work on Bell's M-9 anti-aircraft gun directors. He applied this knowledge to the design of an instrument flight simulator based on an analogue computer. [1.2]

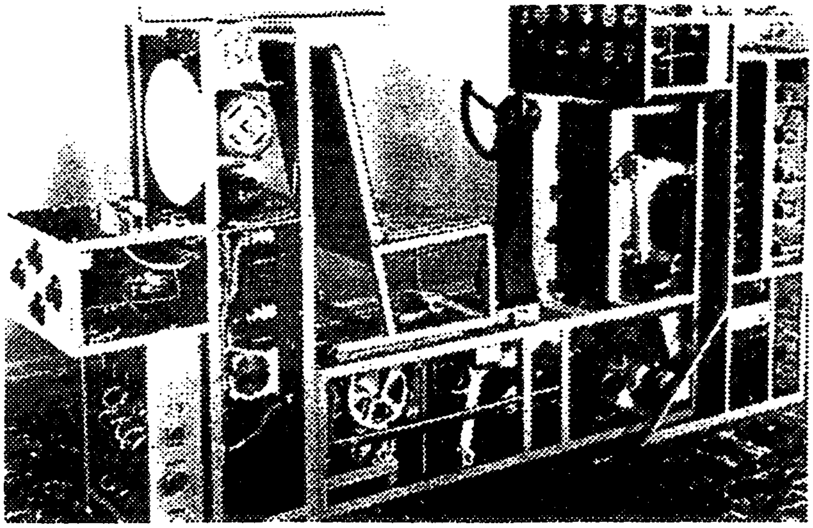


Figure 1.1.4-1 Z-1 Instrument Flight Simulator [1.2]

Analogue computers brought flight simulation into the electronics era. Electronic systems began to replace mechanical methods of simulating the dynamic model. The D.C. method was soon considered as a more demanding technology than A.C. carrier method of analogue computer, but one capable of superior precision in simulation.

1.1.5. Digital Simulators

A special machine was designed at the University of Pennsylvania for their simulator, which was named UDOFT (Universal Digital Operational Flight Trainer). The UDOFT project had demonstrated the feasibility of digital simulation and was mainly concerned with the solution of the aircraft dynamic equations. [1.2]

Nearly all of the simulators produced up to the mid 1950's had no fuselage motion systems. The fixed-base simulators did not feel like flying airplanes. It was found that a handling improvement could be made by empirical adjustment of the control loading and aircraft dynamics simulations that. in part compensated for the lack of motion. In 1958, Redifon received a contract from BOAC for the production of a pitch motion system as part of a Comet IV simulator shown in Figure 1.1.5-1. More complex motion systems were designed capable of producing motions in two and three degrees of freedom, and with the introduction of wide-bodied transport aircraft, such as the 747, a lateral acceleration was required which led to the introduction of four and six degrees of freedom systems. [1.1]

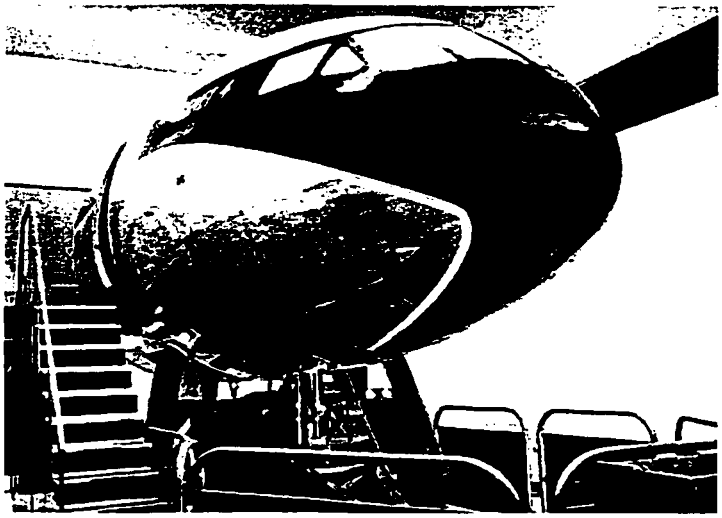


Figure 1.1.5-1 Comet IV FFS [1.1]

1.1.6. Modern Simulators

By offering key visual cues, it provides the final sensory input to the pilot to create a truly realistic training environment. Considerable computing power is necessary to generate digital pictures in real time, and the realistic and flexible visual attachments are fairly recent developments.

The first computer image generation systems for simulation were produced by the General Electric Company (USA) for the space program. [1.4] These systems were able to generate images of three-dimensional objects. Progress in this technology has been rapid and closely linked to developments in digital computer hardware technology. Figure 1.1.6-1 shows a computer-generated view of Frankfurt Airport at dusk.

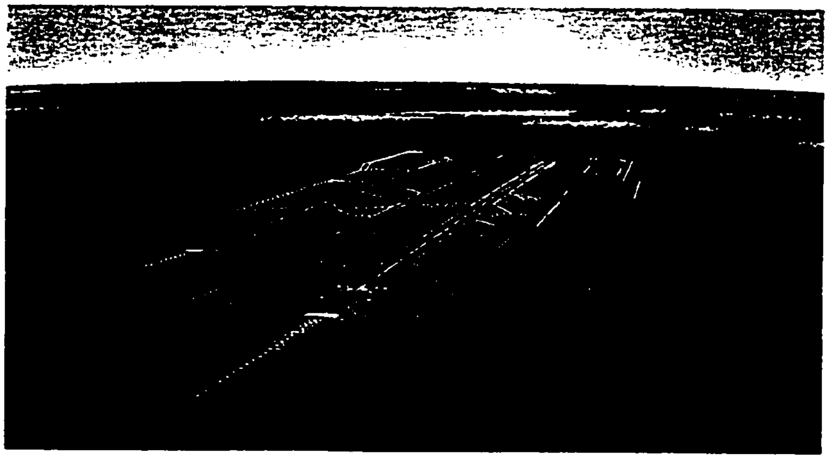


Figure 1.1.6-1 A Computer Generated View of Frankfurt Airport at Dusk [1.2]

Much effort has been devoted to improving the instructional facilities in the simulator. Provided are the use of high resolution touch screens for instructor control and substantial increases in the number of malfunctions and radio stations which can be offered. There are also facilities for exercise recording and playback, pilot performance recording and evaluation, separate pilot and flight engineer training in the same exercise and automated training.

1.2. Flight Training Devices and Full Flight Simulators

As shown previously, flight simulation has been developed in two distinct objectives. The first was as a procedure trainer to teach navigation and instrument flying procedures. Since navigation and instrument procedures training do not require a sophisticated motion system, visual system or accurate flight model. a simple fixed base simulator without these features would suffice. This category of flight simulators is known as Flight Training Devices (FTD). The second objective was to teach actual flying techniques. This develops a feel for the actual aircraft being simulated by providing the most realistic environment possible with motion. sound. visual. and instruments simulated as accurately as possible. This category of flight simulators is known as Full Flight Simulators (FFS).

1.2.1. Full Flight Simulators

Figure 1.2.1-1 shows a modern FFS. At the left of the picture. the rounded section at the front of the simulator is the visual mirror. which wraps round the cockpit 180 degrees giving the pilot and co-pilot an unobscured view all around.

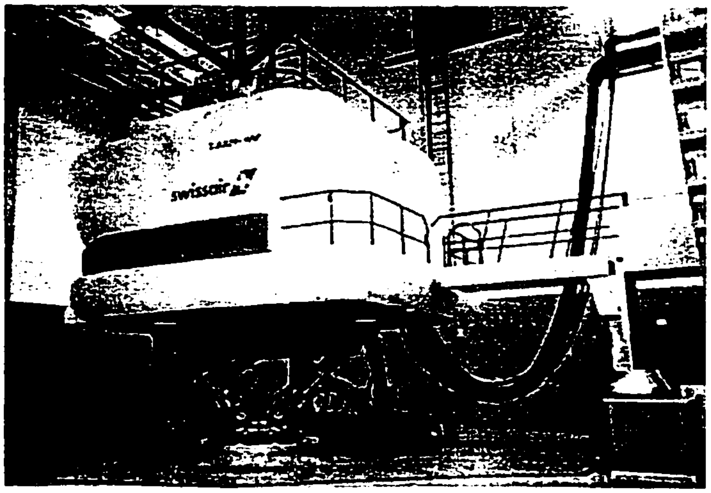


Figure 1.2.1-1 A Modern FFS [1.2]

1.2.2. Flight Training Devices

In 1990 and 1991, the FAA defined seven flight training device levels to provide more flexibility for operators to utilize FTDs of suitable complexity and capability in their training programs. Slowly, the FTD has been "catching on" and there are now a number of the devices in service in flight training programs, about a dozen in the United States and about twice that worldwide. There are more in service in ground training programs, where the devices usually do not require qualification by the regulatory authority. FTDs have not, however, served as effectively and efficiently as they can in air carrier flight training and certification programs. In many cases, especially among large operators, the entire flight-training program is done in an advanced simulator regardless of the training task and without regard to the minimum level of device required to support the particular training task. [1.5]

Air carriers, large and small, have long sought a low cost, highly effective means of pilot training and certification (licensing). Aviation authorities have been amenable to lower cost approaches as long as there was no compromise of training effectiveness or certification standards. Operators have taken the position that if the airplane has to be used at all, the entire training may as well be done in it. This attitude and the cost of FTDs has been a continuing barrier to gaining the advantages offered by FTDs. Attitudes are changing, however, and more interest in the use of FTDs, especially in conjunction with advanced simulators, is becoming evident. The addition of a visual system to an FTD offers the potential of even greater utilization and effectiveness. Visual FTDs will very likely prove to be an excellent complement to a simulator flight training program or flight training programs using airplanes. Figure 1.2.2-1 shows a modern FTD for the Airbus A320. [1.5]

Figure 1.2.2-1 A Modern FTD [1.5]

1.3. Concordia Flight Simulator

In the early 1980s, the Fluid Power Laboratory at Concordia University embarked on a joint project with CAE Electronics to study the feasibility of building a low cost light twin-engine flight simulator called Light Aircraft Simulator System (LASS). [1.6]

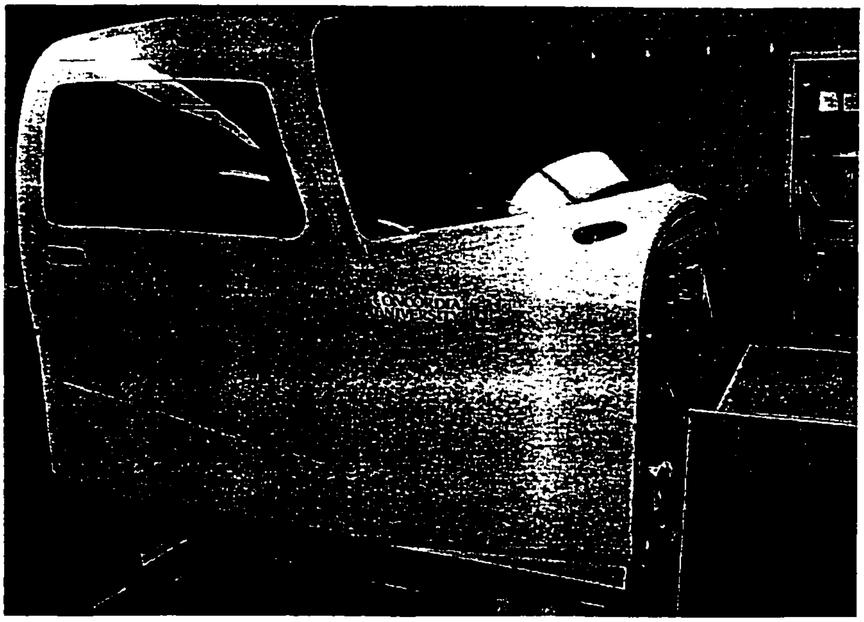


Figure 1.3-1 Light Aircraft Simulator System (LASS)

1.3.1. Light Aircraft Simulator System

The Light Aircraft Simulator System (LASS) was a visual FTD built from the cockpit of an actual Beech Duchess 76 aircraft. It is comprised of the following four major components. The block diagram of this system is shown in Figure 1.3.1-1.

1. The flight model computer and its interface
2. The simulated cockpit section containing all the flight, navigational instruments and controls
3. The visual system that generates the forward view projected ahead of the windshield
4. The instructor station to monitor and control the simulation

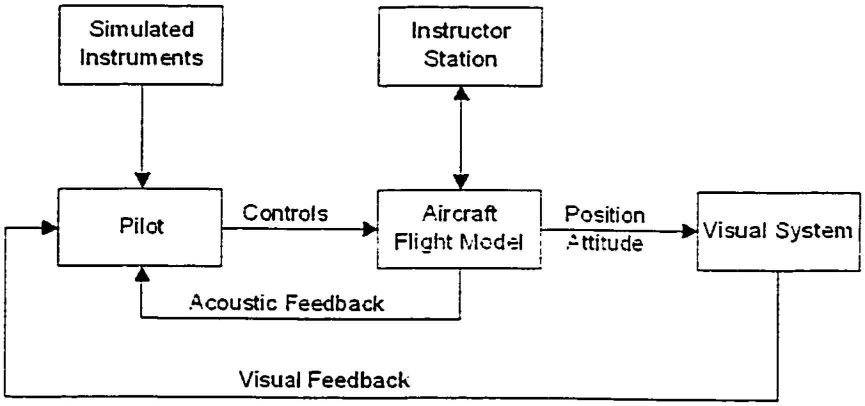


Figure 1.3.1-1 LASS Block Diagram

All the instruments and controls in the cockpit were retrofitted so that they could be realistically driven and read by the flight model computer. The variable controls were fitted with potentiometers to provide a voltage proportional to their position. The flight model computer was an Intel 8088 system with 128KB of memory, an 8-bit bus, and four RS232 serial communication ports to communicate with external devices such as an instructor station. All interface cards used to read and drive the simulated instruments and controls were custom built. Discrete digital input and output lines are used to read on/off devices such as switches and circuit breakers, and drive simple on/off displays such as lamps. The flight model program was coded in Fortran specifically to model the Duchess, and the interface routines that communicate with the instruments and controls were coded in Assembly language. [1.7]

Although unquestionably successful in demonstrating the feasibility of building a low cost FTD, CAE chose not to pursue this project further and the simulator has remained at Concordia University since it has been continuously improved over the years. Later enhancements would include the development of a simple computer generated visual system that would project the forward view from the aircraft on a screen mounted in front of the windshield. A motion system was considered but studies showed that an affordable limited capability motion system would actually detracted from the simulator's value. A decision was made not to pursue a motion system. [1.6]

1.3.2. Enhancement Done by Mr. P. Lawn

In the 1990s, Mr. P. Lawn did his great work to upgrade the computer hardware of the Concordia flight simulator and developed/implemented a more sophisticated and versatile simulation software. Through his research the evolution of the Concordia flight simulator to a powerful academic tool has begun.

A six-degree of freedom dynamic model had been implemented as a C program with features to completely define the stability and control coefficients, to graphically display several instruments, to monitor the performance, and to run in a number of configurations ranging from a PC or interfacing with a fully instrumented aircraft cockpit and visual system. The relationships between the components of the flight dynamic model are outlined in Figure 1.3.2-1. A flow chart shown in Figure 1.3.2-2 is used to implement the flight dynamic model. [1.7]

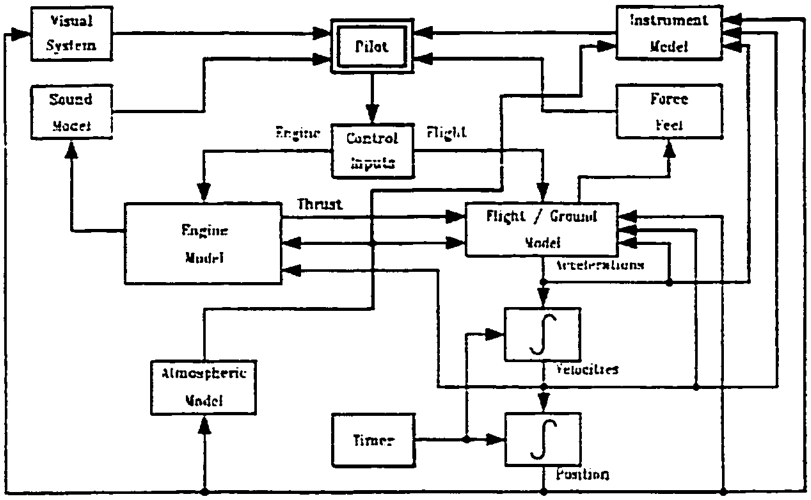


Figure 1.3.2-1 Flight Dynamic Model Block Diagram [1.7]

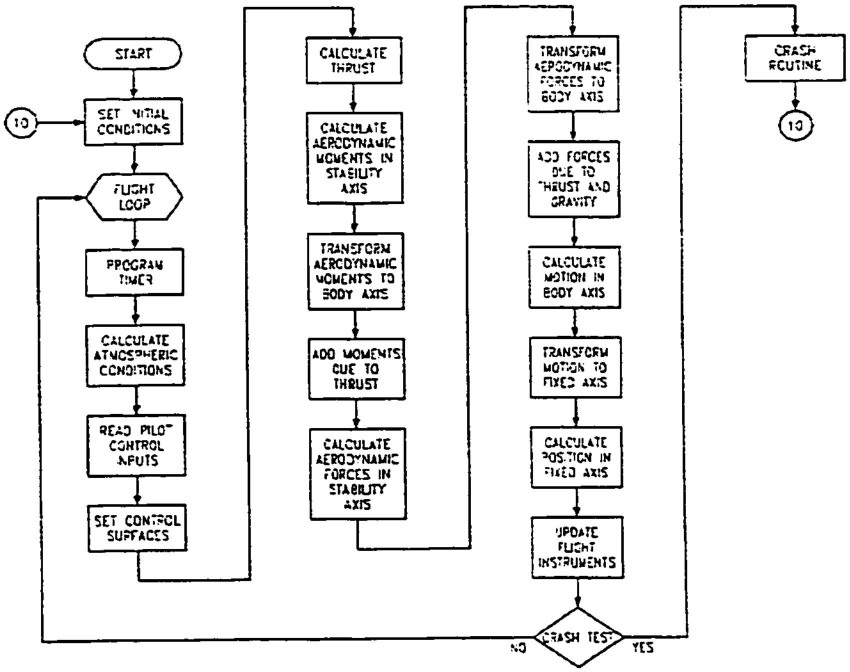

START
10
SET INITIAL CONDITIONS
FLIGHT LOOP
PROGRAM TIMER
CALCULATE ATMOSPHERIC CONDITIONS
READ PILOT CONTROL INPUTS
SET CONTROL SURFACES
CALCULATE THRUST
CALCULATE AERODYNAMIC MOMENTS IN STABILITY AXIS
TRANSFORM AERODYNAMIC MOMENTS TO BODY AXIS
ADD MOMENTS DUE TO THRUST
CALCULATE AERODYNAMIC FORCES IN STABILITY AXIS
TRANSFORM AERODYNAMIC FORCES TO BODY AXIS
ADD FORCES DUE TO THRUST AND GRAVITY
CALCULATE MOTION IN BODY AXIS
TRANSFORM MOTION TO FIXED AXIS
CALCULATE POSITION IN FIXED AXIS
UPDATE FLIGHT INSTRUMENTS
CRASH TEST
NO
YES
CRASH ROUTINE
10

Figure 1.3.2-2 Flight Dynamic Model Implementation Flowchart [1.7]

This flow chart shows the sequential steps of simulation loop taken to implement the flight dynamic model as a simulation code, including a routine to modify the configuration of the aircraft and display or store various program parameters in real time. The system is also a valuable tool to study flight dynamics and to examine the effects of program parameter variations. [1.7]

Under the advisement of Mr. P. Lawn, a plan was set up to continue the evolution of the Concordia flight simulator to a powerful academic tool, and especially focus on the development of the avionic system.

1.4. Thesis Objectives and Contributions

Modern computers have significant advances not only in computing speed, power, and performance, but also in smaller size and are available at lower cost. Many new technologies, such as data acquisition, have been developed for years. It is apparent that upgrading the existing system would allow significant enhancements to the capabilities of the Concordia flight simulator. This project was set up to upgrade the avionic system and rewrite the code to simulate the upgraded system. The objective was to create a useful academic tool for the teaching of avionic navigation systems, and to provide an effective tool for research into topics of aircraft avionic system and navigational instruments. This thesis documents the development of the avionics simulation program, and details the development of the radio navigation systems and the associated navigational instruments. It is also the intention of the author that this thesis offers the necessary insight to the hardware configuration and the avionics simulation software, and as such provides future students with the necessary information to continue in this field.

1.4.1. Reverse Engineering and Hardware Upgrade

To be useful as a teaching and research tool, the avionics simulation software should be able to run on a computer without the simulated cockpit that is only available in the laboratory. If achieved, with a projector to display the results on a screen, the avionics simulation could be run in a classroom environment. This would also allow students to experiment with the flight simulation on their own time at home. From these basic objectives, it is clear that the simulation software should be programmed to run in the Intel Pentium4 PC family of computers. The graphical representation of the navigational instruments would be displayed on the computer screen. A modern commercially available PC would run the complicated flight model in much higher speed. the simulation performance would be much better than ever. Selecting the Pentium4 family of computers would also ensure future compatibility since they would always be downward compatible.

In the laboratory, the flight simulation is run on the computer that drives the simulated cockpit. It is necessary to equip the hardware that interfaces the simulated instruments and controls to the flight simulation computer. In this project. a National Instruments (NI) PCI-6601 card was selected to interface the simulated radio stack to the flight model computer. This commercially available interface card is a timing and digital I/O board for use with the PCI bus in PC-compatible computers. It could perform a wide variety of tasks, and it well meets the need that it is a high-speed digital data acquisition with precise time control for this project.

The radio stack and the associated navigation indicators in the simulated cockpit did not need to be modified. These systems are still representative of the technology used in modern flight simulators.

Unfortunately, many important documents especially those about the simulated radio stack are not available any longer. It makes many difficulties to understand the operation and simulation principles of the radio navigation systems in the Concordia flight simulator. Reverse engineering therefore becomes the first challenge of this project.

1.4.2. Communication and Simulation Software Upgrade

In the original flight simulator, the simulated radio stack system communicates with the flight model computer through a RS232 connector that is linked to an original interface card. This kind of serial communication technology causes many software-programming troubles under the Microsoft Windows NT, 2K and XP that are mostly used in PCs today. In this project, all the software was developed under Windows XP and the advanced data acquisition technology was applied.

LabVIEW, which is the measurement and automation software from National Instruments, was used to develop the upgraded system in this project. It features interactive graphics, a state-of-the-art user interface, and a powerful graphical programming language. [1.8] Using the LabVIEW Data Acquisition VI Library, which is a series of virtual instruments (VIs) for using LabVIEW with National Instruments DAQ hardware (such as PCI 6601), greatly reduced the development time for this project.

C language has been upgraded to C++ for years. As the most productive C++ tool, Microsoft Visual C++ would create the highest performance applications for Windows.

In this project, a radio stack module and a navigation model were coded in VC++. It could provide future students with the necessary information to continue programming the entire flight model in VC++.

1.4.3. Summary of Objectives and Contributions

In summary, the commercially available better performance computers, the advanced modern communication technology and programming software have offered the opportunity to upgrade the Concordia flight simulator. Taking advantage of all these would help achieve the following goals in this project.

1. Run on a modern Intel Pentium4 PC with advanced interface technique
2. Upgrade the avionic system, simulate and model the radio stack system
3. Be capable of running with a full simulated cockpit or limited input devices such as a mouse and a keyboard
4. Develop the simulated radio stack system by using the graphical programming language LabVIEW, represent and control the upgraded system in real time
5. Offer Graphics User Interfaces (GUI) for students to experience the radio navigation systems without any simulated hardware connected to the PCs
6. Be written in VC++ to allow for modifications by students, and be integrated with the flight dynamic model for future research and projects
7. Present the reverse engineering and the development of the simulation in this thesis as a supporting document for students and professors using the Concordia flight simulator

CHAPTER 2

RADIO NAVIGATION SYSTEM OPERATION PRINCIPLES

The importance of accurate navigational techniques has been realized since early aviation. The aviation pioneers initially employed dead reckoning in which the ground speed was calculated from the time required to reach a known checkpoint. The velocity was then used to estimate their position after a measured elapsed time. Dead reckoning techniques are still taught to student pilots today, but it is obviously limited to clear days when landmarks are clearly visible from the air. The need for more reliable methods leads to the development of modern navigation systems.

The radio stack system is the aim object of this project. Before starting to simulate and model the radio stack system, it is necessary to fully understand the operational principles of the actual radio navigation system in the radio stack. Figure 2-1 shows the simulated radio stack in the Concordia flight simulator. The common systems found in general aviation aircraft, which are also equipped in the Concordia flight simulator, are described in this chapter. They consist of Automatic Direction Finder (ADF), Distance Measuring Equipment (DME), Very High Frequency omni-range receiver (VOR), Instrument Landing System (ILS), and Area Navigation (RNAV).

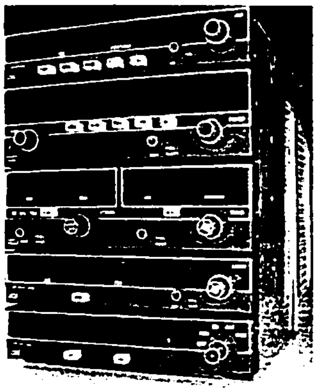


Figure 2-0-1 Radio Stack

2.1. Automatic Direction Finder

ADF is one of the simplest radio navigation systems. It provides the pilot with a relative bearing from the aircraft to a ground station. The bearing is indicated to the pilot over a compass card as shown in Figure 2.1-1. The compass card always indicates the forward direction of the aircraft, and the arrow always points to the ground transmitter. The compass card could be slaved to the gyroscopic compass to maintain the correct direction. In less sophisticated systems, the pilot must manually adjust the direction. In either case, the arrow is not affected by the orientation of the compass card. By following this bearing, the pilot could direct the aircraft to the selected transmitter. Since it only provides a relative bearing from the aircraft to the transmitter. any cross wind would tend to push the aircraft off the direct route and cause it to fly the path as shown in Figure 2.1-2. The pilot could correct for the cross wind by flying a heading in which the indicated relative bearing remains constant but does not necessarily point toward the front of the aircraft. [2.1]

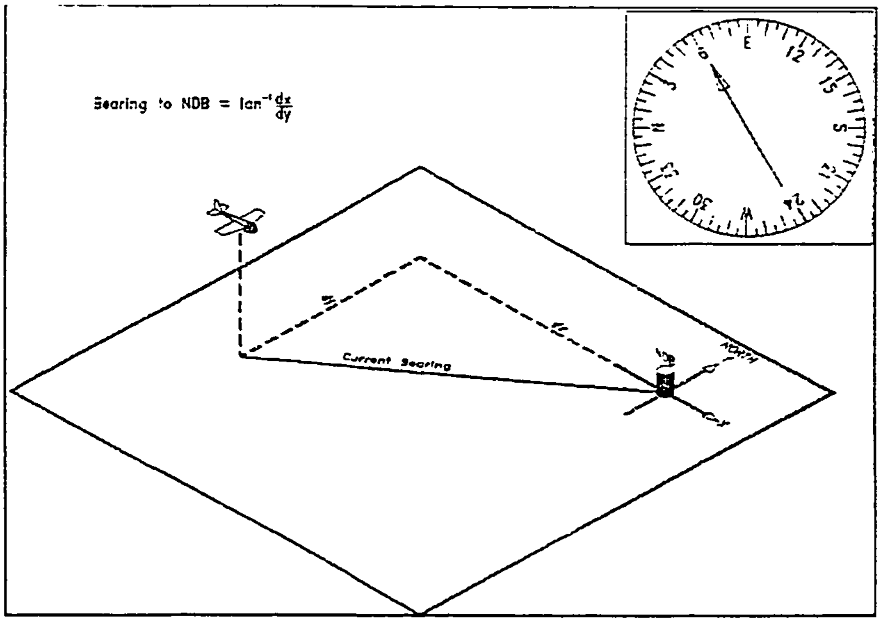


Figure 2.1-1 Aircraft Bearing to a Non Directional Beacon (NDB) [2.2]

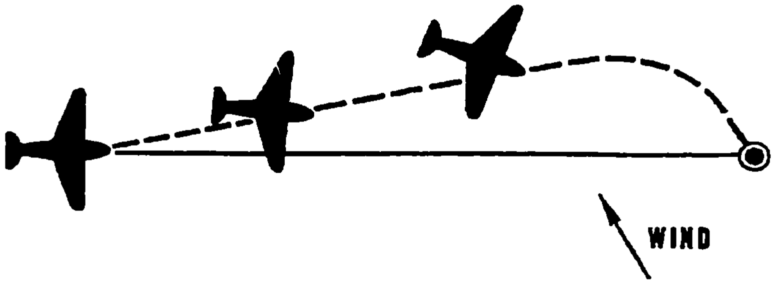


Figure 2.1-2 Aircraft Tracking an NDB Without Compensating for Cross Wind [2.2]

As shown in Figure 2.1-3, a simple single antenna placed in the alternating electromagnetic field of a radio receiver will have a small alternating voltage induced on it, and the voltage is proportional to the radio signal and the distance from the transmitter.

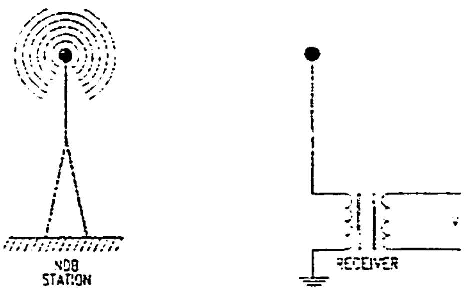


Figure 2.1-3 NDB Station and ADF Receiver [2.3]

Figure 2.1-4 shows a pair of receiver antennas where A and B are each connected to one end of a transformer primary winding. If the antenna is arranged such that A is closer to the transmitter than B, then the amplitude at A will be slightly greater than at B and will result in an oscillation voltage across the transformer of a given phase. As the antennas are rotated so that A and B are equidistant to the transmitter, both A and B will have an equal voltage induced on them, and thus no current will flow through the transformer resulting in a null. As the antennas continue to rotate past this point, B will be closer to the transmitter that will again result in an oscillating voltage across the transformer but of the opposite phase. [2.3]

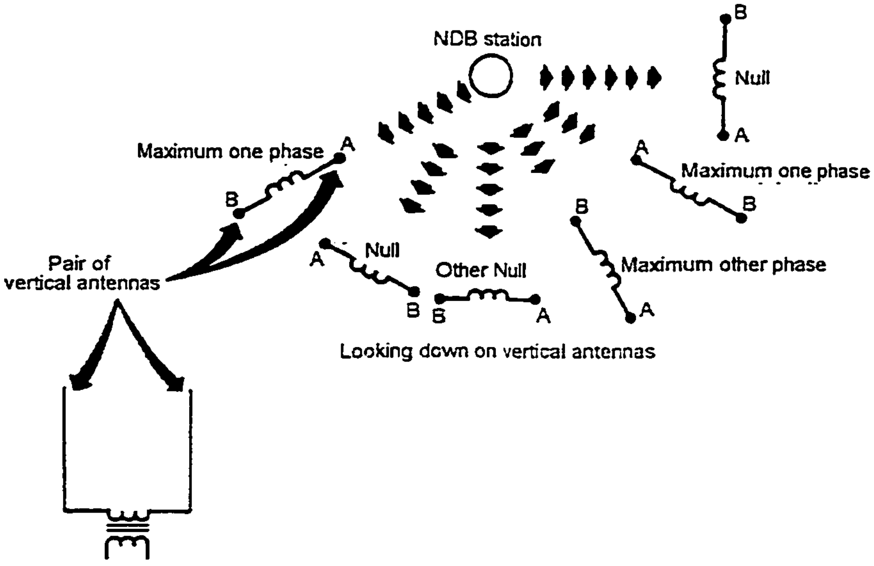


Figure 2.1-4 ADF Operation Principle [2.3]

Since the magnitude of the induced voltage is very small, a loop antenna that could effectively increase the length of A and B is used. Figure 2.1-5 shows the KT44B combined loop/sense ADF antenna for originally use with KR87 ADF receiver that is found in the Concordia flight simulator. The relative bearing to a transmitter could be determined by rotating the antenna until a null is detected at which point the transmitter would be in the direction perpendicular to the antenna loop. Depending on the direction of the phase shift as the antenna passes the null, the position of the transmitter with respect to the loop (in front of or behind) could be determined. [2.3]

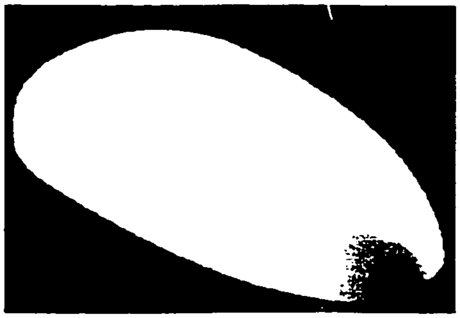


Figure 2.1-5 KA-44B Combined Loop/Sense ADF Antenna [2.4]

This technique of rotating the antenna was used on early systems, but it required a constant adjustment by the navigator. By mounting two mutually perpendicular loop antennas L1 and L2, the amplitude of the resulting signals from each antenna would be a function of the relative bearing to the transmitter θ_{ADF} and would vary by 90 degrees according to Equations 2.1 and 2.2. [2.3]

$$A_1 = A\sin(\theta_{ADF}) \tag{2.1}$$

$$A_2 = A\sin(\theta_{ADF} + 90'') = A\cos(\theta_{ADF}) \tag{2.2}$$

Solving these equations simultaneously yields a solution for the relative bearing to the transmitter in terms of the two resulting amplitudes from each antenna. The result is shown in Equation 2.3. [2.3]

$$\theta_{ADF} = \arctan(\frac{A_1}{A_2}) \tag{2.3}$$

Dedicated non directional beacons (NDB) operate in the low to medium frequency range of 200-415 KHz and 510-535 KHz. however most ADF receivers are designed to tune signals from 190 up to 1750 KHz allowing commercial AM broadcasting stations to be used as navigational aids. [2.5] Figure 4.1-6 shows a typical ADF receiver control panel.

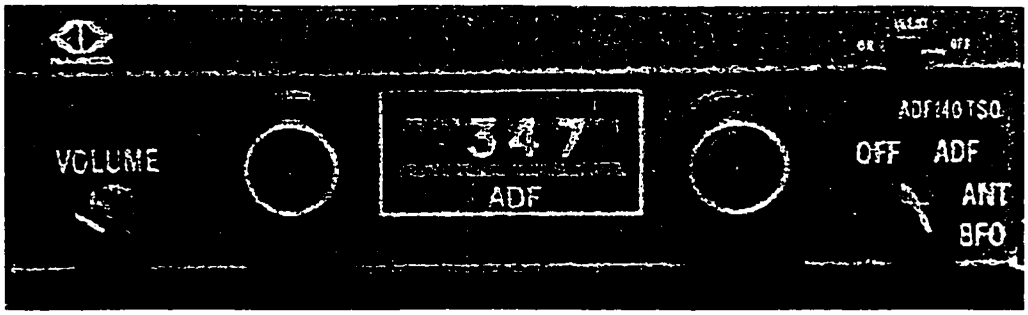


Figure 2.1-6 Typical ADF Control Panel [2.6]

Being low to medium frequency signals, NDB signals traveling over land are less susceptible to ground attenuation than higher frequency signals. The degree of attenuation of a signal traveling over land depends on the frequency of the signal (lower signals suffer less attenuation), and the conductivity of the surface below it. NDB signals traveling skyward can be reflected back to the Earth by the ionosphere. It can greatly increase the effective range of the beacon. However, the extent to which the signal is reflected and the increase in range is a function of both the frequency and the condition of the ionosphere. As a result, there is no simple means to determine the range of an NDB beacon, and it can vary significantly based on several factors. [2.2]

The locations of the ADF transmitters are defined in published documents such as the Canadian Flight Supplement (CFS) in terms of their latitude and longitude. [2.7]

2.2. Very High Frequency Omni-Range Receiver

Using the VOR, the pilot can fly a track directly to the station even in the presence of a crosswind. It overcomes the limitations of the ADF by providing the pilot with an indication of any deviation from a selected course to the VOR station. This information is indicated either on a Horizontal Situation Indicator (HSI) shown in Figure 2.2-1 or a Course Deviation Indicator (CDI) shown in Figure 2.2-2.

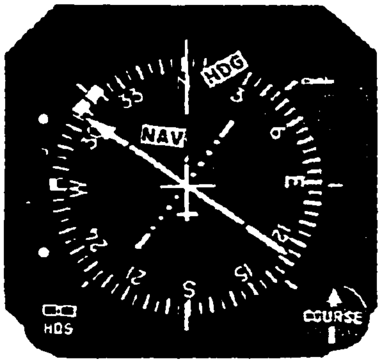


Figure 2.2-1 Horizontal Situation Indicator [2.8]

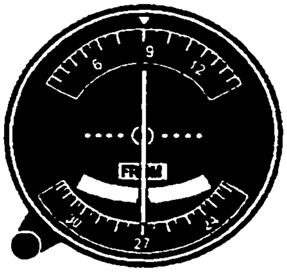


Figure 2.2-2 Course Deviation Indicator [2.8]

A HSI is slaved to the gyroscopic compass to ensure that the compass card is always oriented correctly to indicate the aircraft's heading. A radial is defined as a straight path away from a station at a defined bearing. The pilot selects the desired radial to fly towards or away from the VOR station by orienting the arrow on the compass card using the omni-bearing selector (OBS) knob. The deviation from the selected radial is indicated by the deviation of the center portion of the arrow as shown in Figure 2.2-3. Different indicators may have different amounts of dots but the full-scale deviation is plus or minus 10 degrees from the selected track. [2.3] For example, each dot represents 2 degrees from the selected track in the HSI shown in Figure 2.2-1. As a rule, if the selected track points to the VOR station as the aircraft flies towards it. and away from the station as the aircraft flies away, then the pilot should correct the course in the direction of the needle's deviation. [2.1]

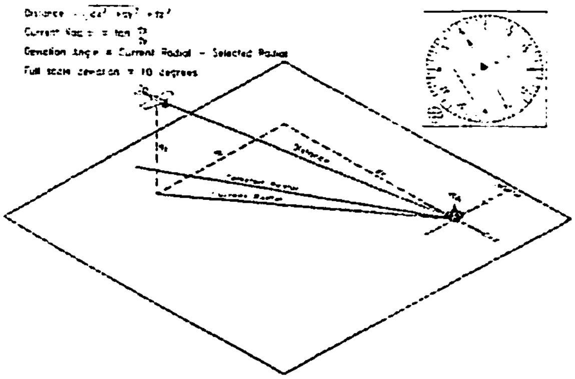

Distance
Current Radial = tan
Deviation Angle = Current Radial − Selected Radial
Full scale deviation = 10 degrees
Distance
Selected Radial
Current Radial

Figure 2.2-3 Aircraft Course Deviation When Approaching a VOR [2.3]

Another indication on the HSI is the "TO" and "FROM" indicator. If the absolute value of the deviation from the selected radial is less than 90 degrees, the indicator will indicate "TO". If the absolute value of the deviation exceeds 90 degrees, the indicator will indicate "FROM". Figure 2.2-4 illustrates the indication and each dot on the indicator represents 5 degrees. The two navigational indicators used in the Concordia flight simulator will be described in Chapter 6.

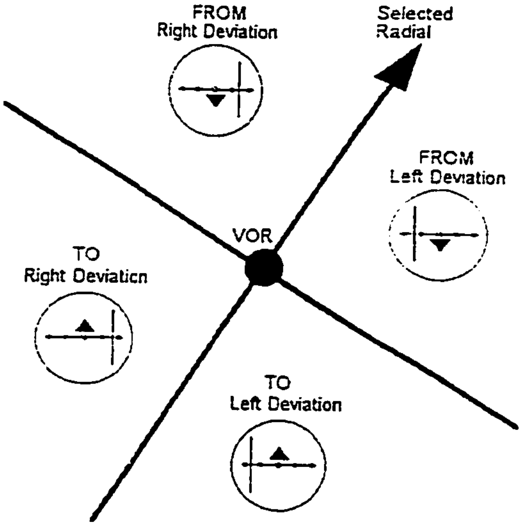


Figure 2.2-4 VOR TO/FROM Indication [2.3]

The VOR operates in the very high frequency range from 108.00 to 117.95 MHz. [2.9] It is the same frequency range used by the instrument landing system (ILS) localizer beam. Appendix 1 gives the list of frequencies in the range, and indicates those assigned for VOR and those for ILS system. In this frequency range, the signal is not reflected by the ionosphere or subject to ground attenuation. The VOR operates along a line of sight and therefore requires an unobstructed path from the beacon to the aircraft. Equation 2.4 estimates the range of reception for a VOR, where the altitude is in feet and the range is in nautical miles. [2.1]

$$Range = 1.23\sqrt{Altitude} \qquad (2.4)$$

A VOR station is often paired with a Distance Measuring Equipment (DME) to provide an indication of the distance remaining to the station. It will be discussed further in the DME section.

A VOR station transmits on a carrier wave in the frequency range from 108.00 to 117.95MHz. [2.9] Placed on this carrier is a frequency modulated 9960 Hz sub-carrier modulated from 9480 to 10440Hz at 30Hz according Equation 2.5. [2.1]

$$f = 9960 + 480\sin(30 \times 2\pi t) \qquad (2.5)$$

The resulting signal is transmitted on a rotating directional antenna that also turns at 30 revolutions per second so that the signal is strongest in the northern direction when the frequency-modulated component is at its maximum frequency. Since the antenna is rotating, a receiver would detect not only the carrier with its frequency modulated signal, but also the signal as being amplitude modulated as the antenna is directed towards and away from it 30 times per second. [2.3] The amplitude-modulated component would vary according to Equation 2.6, where $A_{\max}$ is the maximum amplitude and θ is the bearing from the station to the receiver, measured clockwise from the north. [2.1]

$$A = A_{\max}\sin(30 \times 2\pi t - \theta) \qquad (2.6)$$

A block diagram of a typical VOR receiver is shown in Figure 2.2-5. The 9960 sub-carrier is removed from the carrier, and then fed into a 30Hz amplitude detector and a 30Hz frequency modulation detector. The phase difference between the amplitude and frequency modulated signals is equal to the bearing of the transmitter to the aircraft. VOR periodically identifies itself by Morse code and may have an additional voice identification feature.

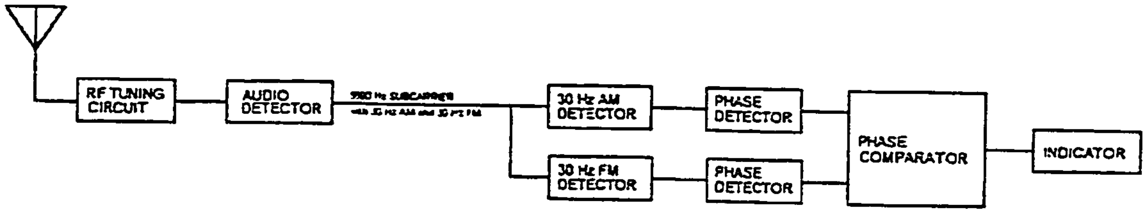


Figure 2.2-5 VOR Receiver Block Diagram [2.3]

2.3. Distance Measuring Equipment

The DME system shown in Figure 2.3-1 provides the pilot with a digital display of the slant range defined as the straight-line distance from the aircraft to a ground station up to 199 nautical miles. A DME station is often collocated with a VOR station and provides the pilot with both range and bearing information. Although the ground distance is the information required by the pilot, the difference between the slant and ground range is small over larger distances. To obtain the ground distance would require additional calculations, as well as interfacing the unit to the altimeter, creating unnecessary complications. It was decided by the avionic manufacturers that. in the interest of simplicity and reliability, the ground distance would not be calculated. [2.10] Many DME systems will differentiate the distance to determine the velocity and estimated time to arrival.

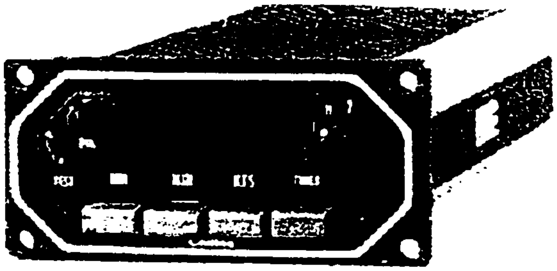


Figure 2.3-1 A Typical DME [2.11]

The DME interrogator transmits a pair of randomly spaces pulses to the ground station on a specific carrier frequency. When the ground station receivers the two pulses, it waits 50 microseconds and then retransmits a pair of pulses with the same spacing as the original one. Measuring the time from the transmitted signal to the reply and subtracting the 50-microsecond delay, the distance to the station could be calculated. Since the time between the pulses is random, an interrogator can use this distinctive spacing information to differentiate between replies to other aircraft and those replying to its own transmission. [2.12]

The DME interrogation carrier frequency from the aircraft is in the range from 1025 to 1150MHz, and the ground DME station replies on a carrier frequency of 63MHz above or below the interrogating signal.

2.4. Instrument Landing System

The final phase of any successful flight is the approach and landing at an airport. For both safety and the economic importance of maintaining a schedule, an Instrument Landing System (ILS) was developed to guide a pilot during the final approach without a visual reference to the runway or ground. At a predefined altitude before touchdown, called the "Decision Height", the landing must be aborted if the runway is not in sight. ILS includes marker radio beacons (outer, middle, and inner). The three marker beacons aligned with the runway indicate the distance of the runway threshold. The marker beacon receiver announces the beacon passing both visually and audibly.

When approaching a runway equipped with an ILS system, the pilot tunes the frequency of the ILS for the intended runway. As the VOR frequencies, the ILS frequencies range from 108.01 to 119.90MHz. Appendix 1 lists the frequencies that are assigned to VORs and ILSs. [2.3] When tuned, the pilot will receive information about the aircraft's horizontal alignment with the runway from the localizer, and the vertical deviation from an established approach angle from the glideslope. The information is presented to the pilot on a Course Deviation Indicator (CDI) similar with the one used for the VOR system, except that it also provides the glideslope deviation with the deflection of a horizontal bar, and does not require the course be set on the CDI. A typical CDI with glideslope indication is shown in Figure 2.4-1.

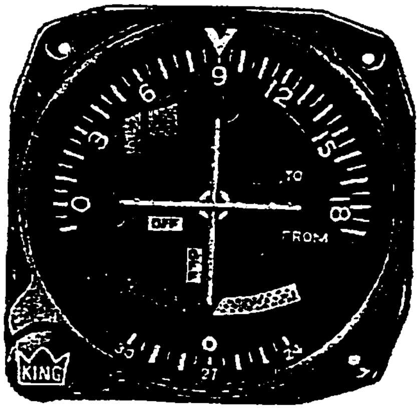


Figure 2.4-1 CDI with Glideslope Indication [2.3]

The same CDI is often used for both the VOR and the ILS systems, with a red glideslope flag in view when in the VOR mode to indicate that there is no glideslope indication. The indications of the CDI are shown in Figure 2.4-2 and 2.4-3. If the aircraft drifts below the glideslope, the horizontal bar will shift up. If the aircraft drifts above the glideslope, then the horizontal bar will shift down. A full-scale deflection of the horizontal bar represents a deviation of plus or minus 0.7 degrees from the glideslope. A full-scale deflection of the vertical bar represents a deviation of plus or minus 2 degrees from the localizer. [2.12]

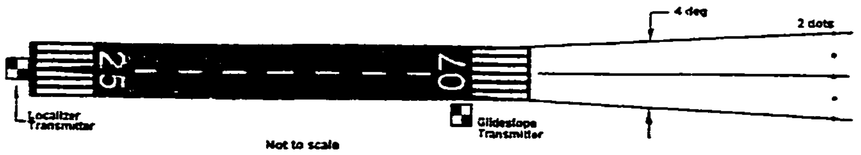


Figure 2.4-2 Localizer Deviation Indication [2.12]

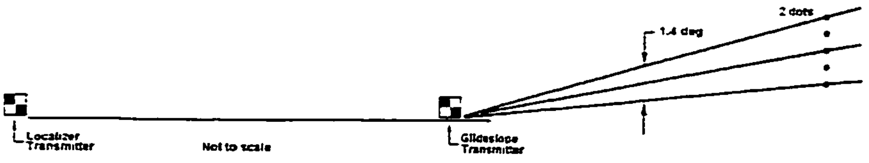


Figure 2.4-3 Glideslope Deviation Indication [2.12]

Both the localizer and the glideslope operate on a similar principle. For the localizer, two signals of 90 and 150Hz are transmitted on a carrier frequency that the pilot tunes. The two signals are transmitted directionally as that their signal strength varies as shown in Figure 2.4-4. The ILS receiver detects both frequencies and compares their relative strength, and then displays the difference between them as a deviation of the vertical bar on the CDI. If the aircraft is exactly on the localizer, both signals will have equal strength and the vertical bar will be centered. If the aircraft is to the left of the localizer beam, the 90Hz signal will be stronger than the 150Hz and the difference in strength will be indicated as a displacement to the right on the CDI. Similarly if the aircraft is to the right of the localizer beam, then the 150Hz signal will be stronger and the CDI will show a displacement of the vertical bar to the left. [2.12]

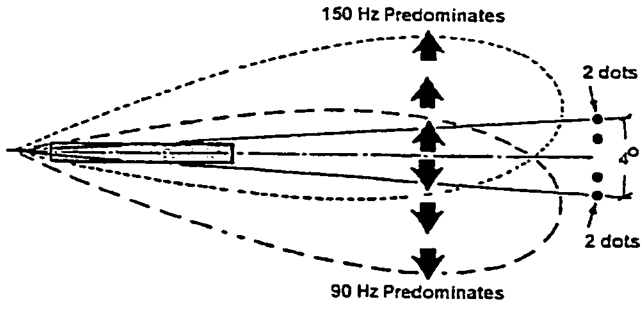


Figure 2.4-4 Localizer Signal Pattern [2.12]

For the glideslope, it operates in the same manner as the localizer except that the 90 and 150Hz signals are directionally transmitted as shown in Figure 2.4-5 and on a carrier frequency several times higher than the carrier of the localizer. Appendix 1 lists the glideslope frequencies associated with each localizer frequency. Note that the pilot does not need to tune both the localizer and the glideslope frequencies. When a localizer frequency is selected, the ILS receiver will automatically select the corresponding glideslope frequency. If the aircraft is approaching the runway below the glideslope, then the 150Hz signal will be stronger and the CDI will show an upward deviation of the horizontal bar. Similarly, if the aircraft drifts above the glideslope, then the 90Hz signal will be stronger and the CDI will show a downward deviation of the horizontal bar. Figure 2.4-6 shows a block diagram representation of an ILS receiver. [2.12]

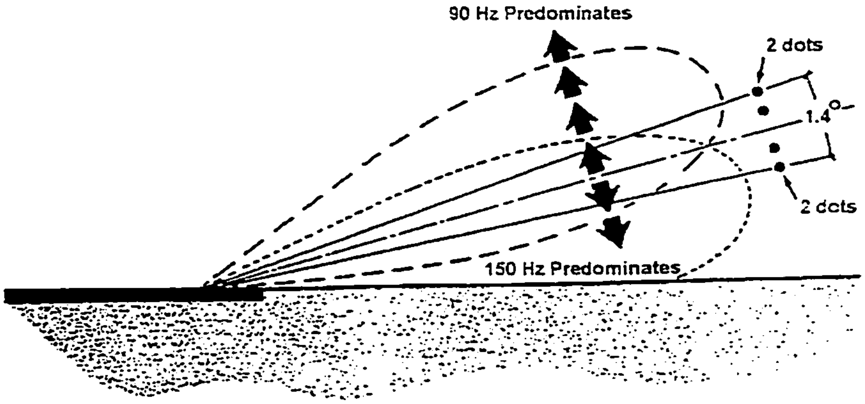


Figure 2.4-5 Glideslope Signal Pattern [2.12]

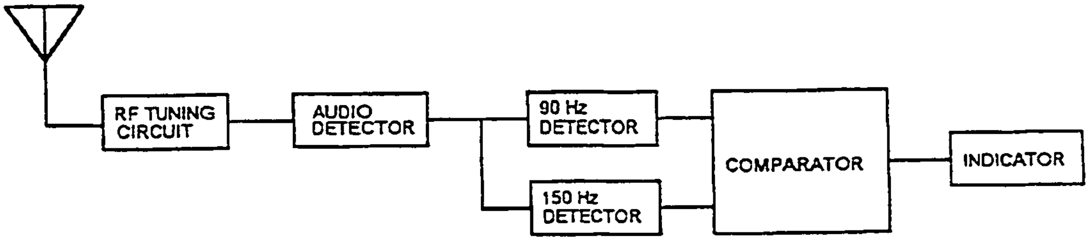


Figure 2.4-6 Typical ILS Glideslope or Localizer Receiver [2.12]

2.5. Area Navigation

Area navigation (RNAV) is a navigation and guidance system that uses VOR bearing, DME slant ranging, and barometric altitude as its basic signal inputs to compute course and distance to a waypoint. Since the system can only function within the service area of a VOR/DME station, it cannot be used for overseas navigation. Figure 2.5-1 illustrates the operation principle of RNAV. [2.12]

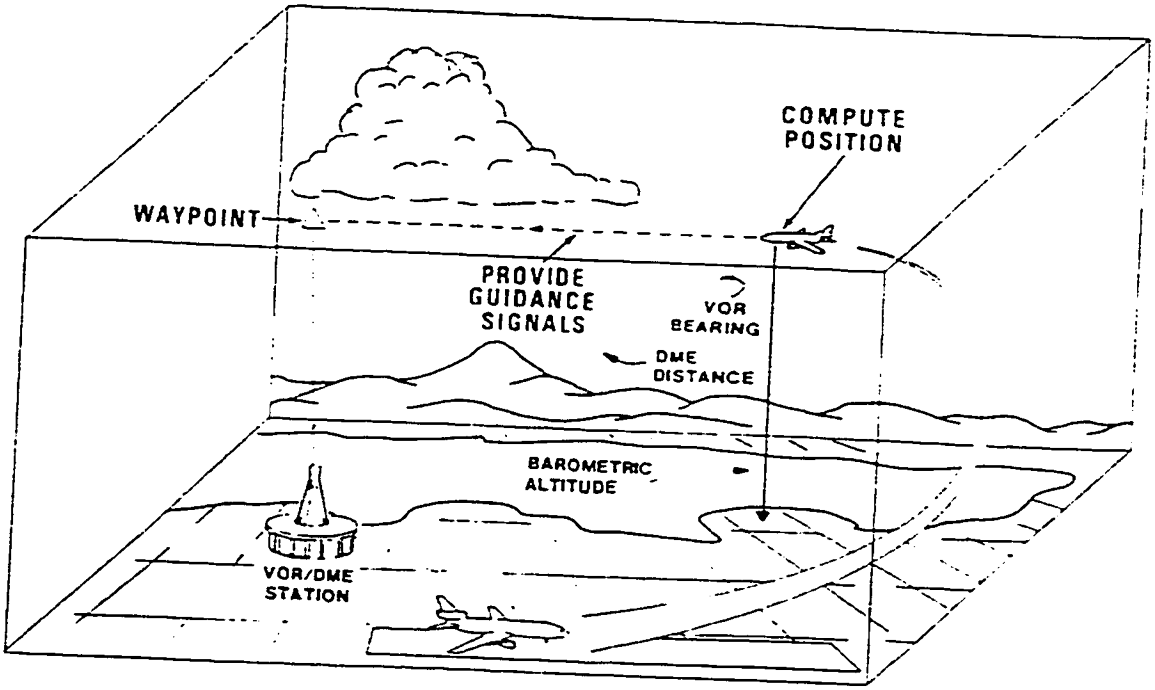


Figure 2.5-1 RNAV Operation [2.12]

A block diagram of a typical system is shown in Figure 2.5-2. The navigation computer receives a VOR bearing from the VOR receiver, DME distance from the DME interrogator, and altitude from the central air data computer.

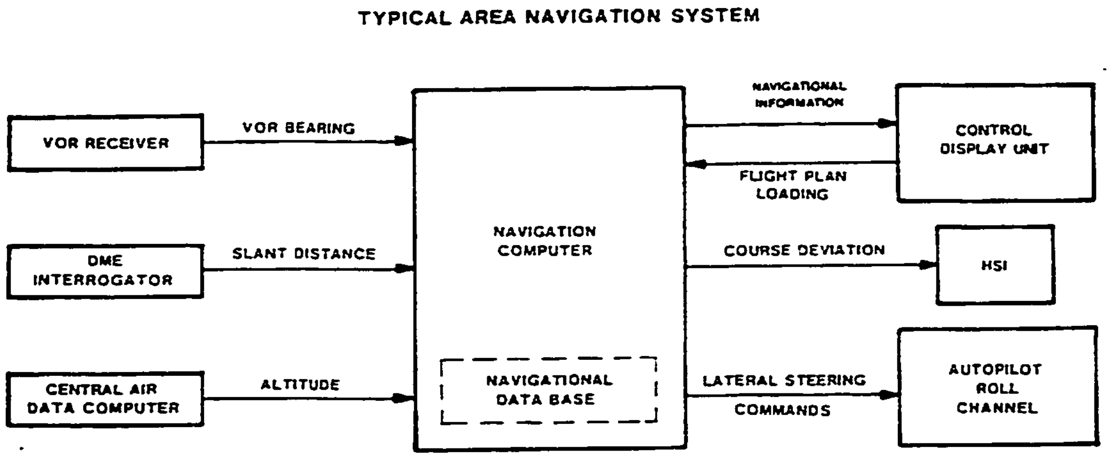


Figure 2.5-2 Typical RNAV Block Diagram [2.12]

A navigational database is stored either within the navigation computer or in an external storage unit. The navigational database contains all information needed regarding the routes between cities, the navigation aids (VOR/DME stations) and waypoints.

The control display unit is used to enter information into the computer and to display navigation information. In a typical commercial airplane installation, the computer may also send course deviation signals to the course deviation indicator, and lateral steering commands to the autopilot.

A flight plan without RNAV could be as shown in Figure 2.5-3. The aircraft flies from one VOR station to another, by a round about path, until it arrives at destination. Unfortunately, there cannot be an ideal line-up of VOR stations between all stations. The area navigation concept provides direct routes between airports. [2.12]

Along each route there are waypoints towards that the airplane flies. The waypoint locations are established when the route is designed. Each waypoint is associated with a specific NAV aid or VOR/DME station, such as in Figure 2.5-4. [2.12]

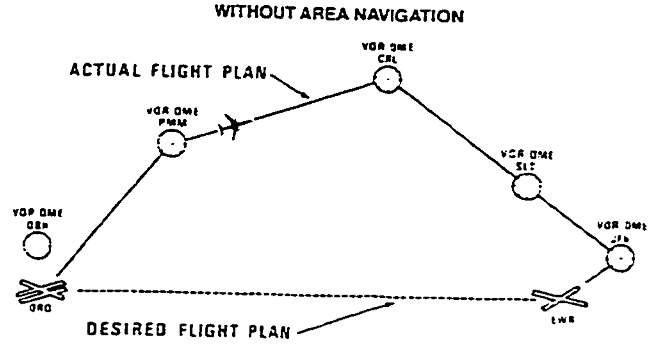


Figure 2.5-3 Without Area Navigation [2.12]

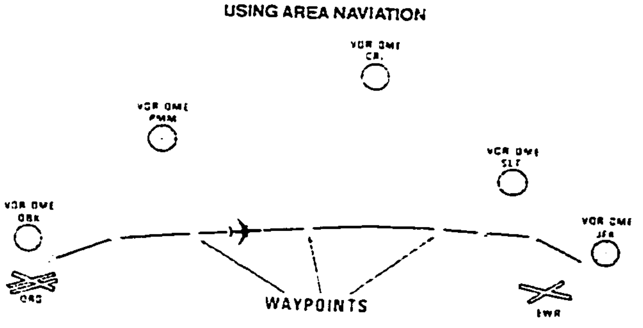


Figure 2.5-4 Using Area Navigation [2.12]

The navigational database stored in the computer contains the following characteristics of each waypoint, as shown in Figure 2.5-5: latitude and longitude, altitude, frequency of its NAV aid, distance from the NAV aid, and magnetic bearing from the NAV aid. [2.12]

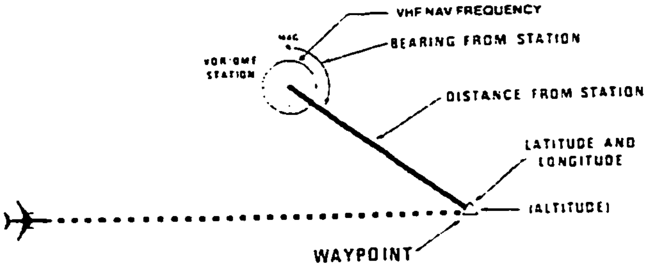


Figure 2.5-5 Waypoint Characteristics [2.12]

If the VHF navigation system is tuned to the proper NAV aid, the area navigation computer will receive the information regarding the position of the aircraft in respect to the NAV aid as shown in Figure 4.5-6. [2.12]

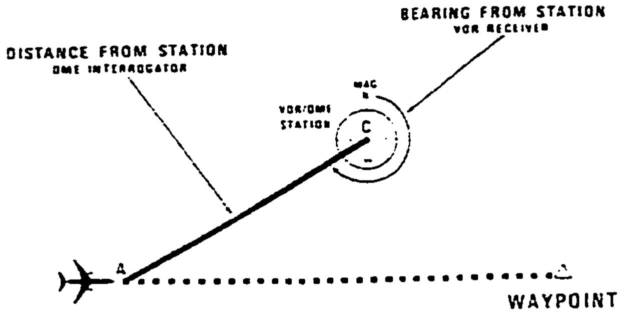


Figure 2.5-6 NAV Aid [2.12]

As shown in Figure 2.5-7, knowing the length of side A (DME distance), the length of side B (from the database), and the angle of A (difference between the bearing of the aircraft and the bearing of the waypoint), the length of side A-B can be computed, which gives the distance to the waypoint; and angle B, which is the course or track angle to the waypoint. This combination is referred to as the Rho-Theta mode of area navigation, where Rho is the DME distance between the aircraft and the VOR/DME station, and Theta is the VOR angle that is the difference between the bearing of the aircraft and the bearing of the waypoint. [2.12]

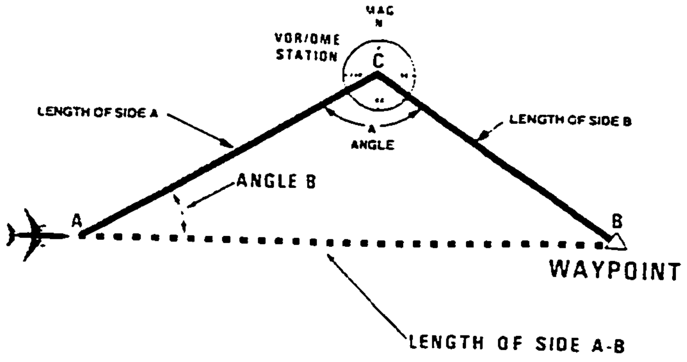


Figure 2.5-7 Rho-Theta Mode [2.12]

An improvement over Rho-Theta is possible using two DME distances. The navigation database would be expanded to provide each waypoint with two NAV aid references. Improved position accuracy is achieved along with improved navigational accuracy. Rho-Rho is the preferred method of area navigation, as shown in Figure 2.5-8. [2.12]

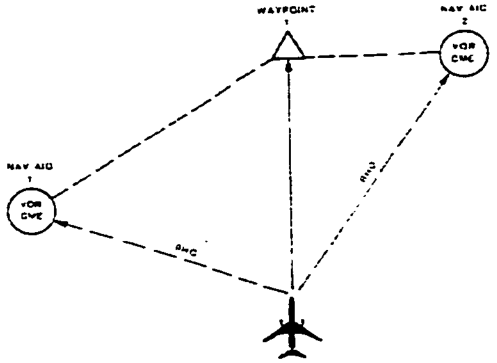


Figure 2.5-8 Rho-Rho Mode [2.12]

Area navigation provides more airspace between cities. As shown in Figure 2.5-9. three additional routes are established, each with its own series of waypoints. referencing the same NAV aids as the first route. This allows four times more traffic in the same route area. [2.12]

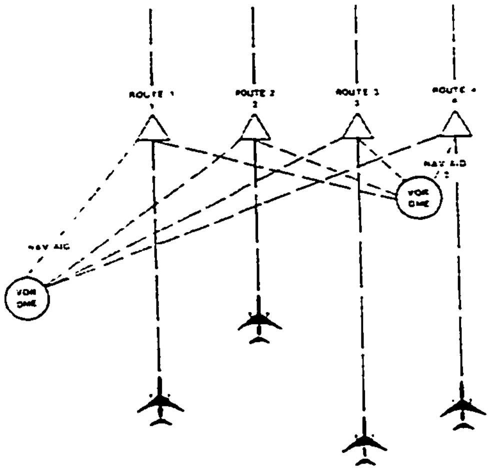


Figure 2.5-9 Airspace Using Area Navigation [2.12]

If the parallel routes are added and more flights are added to a geographic area, it is desirable to achieve the best possible navigational accuracy. This can be accomplished through use of frequency scanning DME interrogators. The interrogators can report distances to as many as five NAV aids, providing that the receptions are good.

The RNAV computer will use the ones that provide the best angles for mathematical computation, resulting in excellent course, distance and present position computation, as shown in Figure 2.5-10. [2.12] If a NAV aid with the aircraft and the waypoint could geometrically form a smaller acute triangle, it would be the better NAV aid used for the computation.

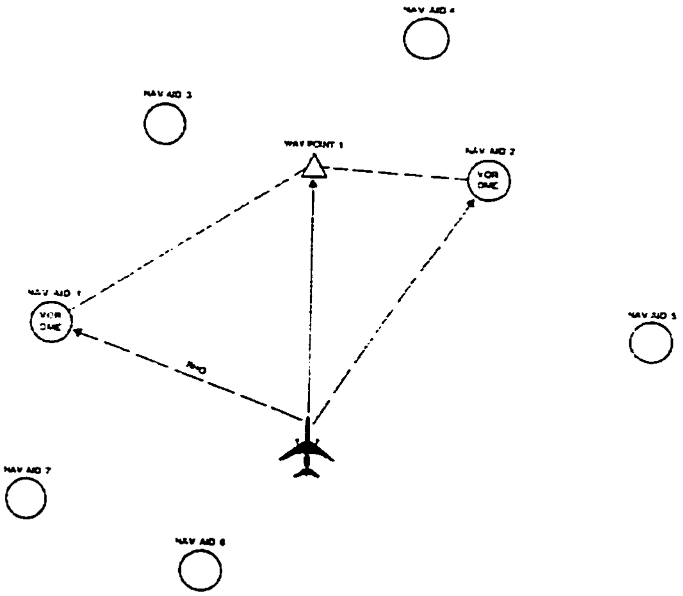


Figure 2.5-10 Frequency Scanning DME [2.12]

In summary, RNAV determines “fictitious” waypoints equivalent to real NAV aids. These waypoints are computed in reference to existing NAV aid by using its VOR frequency and specifying the waypoint by its radial and DME distance. [2.13]

2.6. Summary

One of the objectives of this project is to simulate and model the radio stack system. The radio navigation systems, which are found in the radio stack of the Concordia flight simulator, include the ADF, VOR, ILS, DME, and RNAV systems. In this chapter, the operation principles of these systems were presented.

Automatic Direction Finder (ADF) is the radio device that is used to sense and indicate the direction to a low/medium frequency non-directional radio beacon (NDB) ground transmitter. NDB is superimposed with a Morse code identifier. On the ADF instrument in the cockpit, the needle points towards the selected beacon and enables the pilot to fly the required procedure. Typical procedures in which NDBs are used are in approaches to airfields and keeping track of the aircraft when flying en-route. [4.14]

VHF Omni-Range (VOR) is a ground-based electronic navigational aid transmitting very high-frequency navigation signals, 360 deg in azimuth, oriented from magnetic north. VOR periodically identifies itself by Morse code and may have an additional voice identification feature. This VHF Omni-Range navigation method relies on the ground based transmitters that emit signals to VOR receivers. The aircraft equipment receives these signals, calculates the difference between them, and interprets the result as a radial or bearing From/To the ground station. VOR is the most commonly used radio navigation aid is in use today. [4.14]

Distance Measuring Equipment (DME) provides distance information and primarily serve operational needs of en-route or terminal area navigation. In the operation of DME, paired pulses at a specific spacing are sent out from the aircraft (this is the interrogation) and are received at the ground station. The ground station (transponder) then transmits paired pulses back to the aircraft at the same pulse spacing but on a different frequency. The time required for the round trip of this signal exchange is measured in the airborne DME unit and is translated into distance (nautical miles) from the aircraft to the ground station. [4.14]

Instrument Landing System (ILS) is a precision-instrument approach system that consists of the localizer, the glideslope and marker radio beacons (outer, middle, and inner). It provides lateral, longitudinal and vertical guidance for the approach. The relative position of the aircraft to the ideal flight path is indicated by the needles of the localizer/glideslope (LOC/GS) indicator or by the HSI. The three marker beacons aligned with the runway indicate the distance of the runway threshold. The marker beacon receiver announces the beacon passing both visually and audibly. [4.14]

Area Navigation (RNAV) is a system of radio navigation that permits direct point-to-point off-airways navigation by means of an on-board computer creating phantom VOR/DME transmitters termed waypoints. RNAV equipment can compute the airplane position, actual track and ground speed and then provide meaningful information relative to a route of flight selected by the pilot. Distance, time, bearing and cross-track error relative to the selected "TO" or "active" waypoint and the selected route are provided to the pilot. RNAV was developed to provide more lateral freedom and thus more complete use of available airspace. [4.14]

The operation theories of these radio navigation systems provide the fundamental knowledge for processing this project.

CHAPTER 3
REVERSE ENGINEERING

It has been widely acknowledged that aircraft avionics require human factors resources. The general causes of human error and the means for error prevention, originate in basic human capabilities and limitations. [3.1] In the Concordia flight simulator, the radio stack called Silver Crown Plus Series was originally manufactured by the BENDIX/KING Company. These panel-mounted avionics have the contemporary faceplate that helps improve ergonomics and makes these units easier to operate. They are equipped with backlighting of the bezel nomenclature to make night use easier, and sturdy knobs to give a more precise feel. [3.2]

One of the objectives of this project is to simulate and model this radio stack system. The radio stack system interface is shown in Figure 3-1, and each component will be described in the following sections.

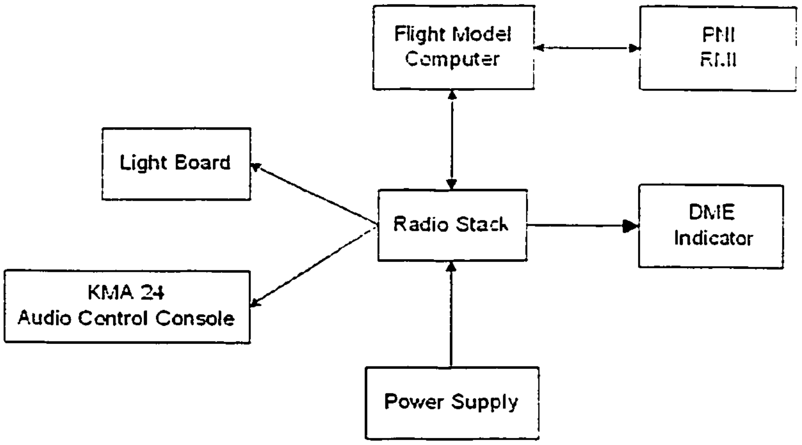


Figure 3-0-1 Simulated Radio Stack System Interface

Unfortunately, several important documents about the simulated radio stack are not available any longer. It made many difficulties to progress this project. Reverse engineering for the radio stack therefore becomes the first challenge in this work. This chapter presents the hardware configuration and the simulation principle of the radio stack system in the Concordia flight simulator.

3.1. Radio Stack Hardware Configuration

The simulated radio stack was built in the early 1980s. One of the objectives of this project is to preserve the basic configuration while upgrading the computer, the interface and the software. The new configuration supports the development of advanced navigation model using high level programming languages, and has the capability to read and drive many devices by using advanced communication technology.

All the radio navigation systems and the navigational instruments were originally manufactured by the BENDIX/KING. The radio navigation system used in the Concordia flight simulator consists of the following components. The letters and numbers in front of the system names are quoted from the original KING documentations.

- KR 87 Automatic Direction Finder (ADF)
- KNS 81 Digital Area Navigation System (RNAV)
- KX 165 Very High Frequency Navigation and Communication Transceiver (NAV/COMM)
- KY 196 Very High Frequency Communication Transceiver (COMM)
- KT 79 Air Traffic Control (ATC) Transponder (XPNDR)

These five components are mounted in the radio stack. They are called as the five stations of the radio stack. Most of the stations contain a switchboard, a display board including an automatic photocell dimming circuit, and an interface board. In some stations, the switchboard and the display board are integrated together. The design philosophy of the simulated switchboard and display board remains similar with the system mounted in the real aircraft.

- KDI 572 Distance Measuring Equipment (DME)
- KMA 24 Marker Beacon Receiver / Isolation Amplifier (MBR),

These two components are respectively interfaced to the radio stack via a DME connector and an Audio connector. A custom Audio/DME interface board was built and mounted in the back portion of the radio stack.

- KI 229 Radio Magnetic Indicator (RMI),
- KI 525A Pictorial Navigation Indicator (PNI)

These two associated navigation indicators are much more complicated. The operation and simulation of them will be described in Chapter 6.

In the original radio stack system, a specified power supply and a microprocessor system were also included.

3.2. Power Supply and Back Panel

3.2.1. The Power Supply

The power supply of the radio stack shown in Figure 3.2.1-1 is quite special since a high voltage +185V is required to drive the gas discharge display on the front panel of each station. The +5V is required to drive the TTL type Integrated Circuit (IC) chips. and the $\pm 9V$ voltages are used for some CMOS type IC chips.

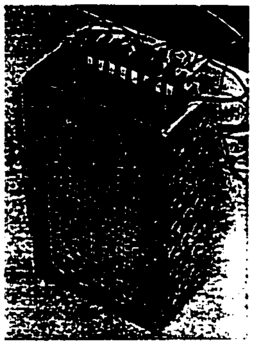


Figure 3.2.1-1 Power Supply of Radio Stack

In the original system, the main power supply of the simulator provided all the required power sources for the simulated cockpit. However, in this project, the radio stack power supply inputs shown in Table 3.2.1-1 were directly provided by a laboratorial power supply for convenience.

Input Voltage	Type of Power
120V	AC
+15V	DC
−15V	DC

Table 3.2.1-1 Power Supply Inputs

The power supply provides five output voltages listed in Table 3.2.1-2 to drive the entire radio stack, including switches, display, and all interfaces. A step-up transformer is required to transfer 120V AC to 185V DC, and the 7805 and 7905 regulators are respectively used to get $\pm 9V$ from $\pm 15V$. Since there is a high voltage in our system, a heat sink, a current limit (a 25-amp slow-blow fuse), and an over-voltage protection circuit are also included in the power supply.

Output Voltage	Color of the Power Line
GND	Black
+5V	White
+9V	Green
−9V	Red
+185V	Blue

Table 3.2.1-2 Power Supply Outputs

3.2.2. The Back Panel

The back panel shown in Figure 3.2.2-1 contains all the interface ports for the radio stack. The power supply socket inputs the power sources from the radio stack power supply. An RS232 port was used to interface the radio stack to the flight model computer in the original system. The DME connector is used to interface the DME indicator. The audio connector is used to interface the KMA 24 Audio Panel/Marker Beacon Receiver. Another connector is used to interface the simulated radio stack to a light board. These five sockets are described in Table 3.2.2-1.

Figure 3.2.2-1 Back Panel of Radio Stack

Socket	Description
Power supply connector	Male 5-pin
RS232 port	Female 25-pin
DME connector	Female 25-pin
Audio connector	Male 25-pin
Light board connector	Male 25-pin

Table 3.2.2-1 Sockets on Back Panel

The detail signal information of each pin in the sockets is shown in Appendix 2.

3.3. Microprocessor System

The microprocessor system shown in Figure 3.3-1 is one of the challenges to upgrade the original radio stack in this project. In the original radio stack system, the microprocessor is an AMD D8085A-2 shown in Figure 3.3-2. [3.3] The microprocessor system could input/output data to the radio stack via an ISA bus, and input/output data to the flight model computer via a RS232 port. This microprocessor system controls the programmable keyboard/display controller 8279 mounted in each station. A total of seven 8279 controllers are being used in the radio stack system. The 8279 will be described in detail in Section 4.2.1.3.

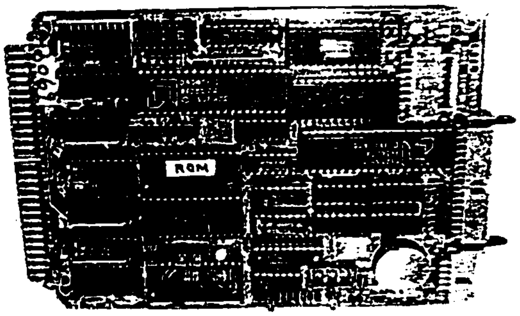
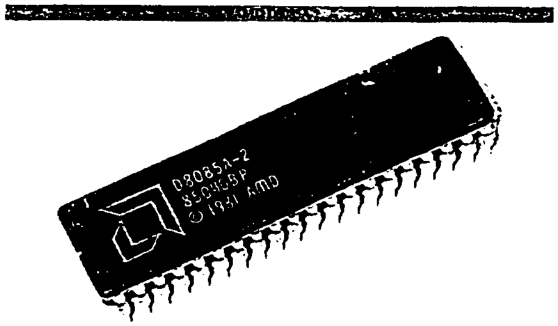


Figure 3.3-1 Microprocessor System of Original Radio Stack



Type	Microprocessor
Data bus width	8 bit
Package	40-pin ceramic DIP
Speed (MHz)	5
Physical memory (KB)	64
V core (V)	5 ± 5%
Min/Max operating temperature (°C)	0 - 70
Max power dissipation (W)	0.35

Figure 3.3-2 AMD D8085A-2 [3.3]

The required signals from the microprocessor system can be classified in three groups listed in Table 3.3-1. Table 3.3-2 describes the more detail information about the Address lines.

Group of Signals	Description
8-bit Data Lines	D0-D7
8-bit Address Lines	A0-A7
4-bit Control lines	Read ($\overline{RD}$), Write ($\overline{WR}$), Clock (CLK), Reset

Table 3.3-1 Required Signals From Microprocessor System

Signal Name	Function Description
A0	Control signal for the 8279 display/keyboard interface controller in each station
A1-A4	Inputs for the decoder 74LS154 that is mounted on the Audio/DME interface board. it decodes the chip select signals for the different 8279 chips in each station
A5-A6	Control signals for the decoder 74LS154
A7	For the microprocessor battery-low alert LED that is mounted on the Audio/DME Interface Board

Table 3.3-2 Address Lines of Microprocessor System

The microprocessor system card is mounted on an ISA slot that is mounted in the back portion of the radio stack. The power signals and the signals from the microprocessor system can be found on this ISA bus. The detail information of each pin on the ISA bus is shown in Appendix 3.

One thing is noticed that all of these signals flow into the Audio/DME interface board via the ISA bus. The proper signals that are required by each station are processed there, and then sent to the proper stations of the radio stack along a 64-pin connector called the Common Bus Connector.

3.4. Audio/DME Interface Board

The Audio/DME Interface Board shown in Figure 3.4-1 is the main interface section of the radio stack. It has six important functions shown in Table 3.4-1.

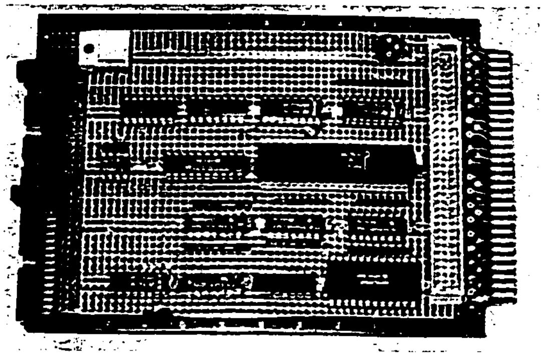


Figure 3.4-1 The Audio/DME Interface Board of Radio Stack

Functions	Main Functional Chips
Decode address lines	Decoder 74LS154
Verify control lines from the microprocessor system	OR Gate 74LS32, Inverter 74LS04
Transfer data lines	Buffer 7407
Generate High Frequency 7.2KHz	555 Timer Circuit Design
Amplify Audio signals from the stations	Operational Amplifier LM 324
Interface function for the DME Indicator	Programmable Keyboard/Display Controller 8279

Table 3.4-1 Functions of Audio/DME Interface Board

In order to fully understand these six functions, the reverse engineering was started from the four connectors shown in Table 3.4-2 on the Audio/DME interface board. As long as the function of each pin in these connectors is determined, along the signal cable connections, the function of each chip on each board can be figured out and then the operation of the Audio/DME interface board can be understood.

Connector	Description
Audio/DME ISA Slot	56-pin
Common Bus Connector	Male 64-pin
DME Connector	Male 25-pin
Audio Connector	Male 25-pin

Table 3.4-2 Connectors on Audio/DME Interface Board

The KDI 572 DME indicator is shown in Figure 3.4-2. and the KMA 24 Marker Beacon Receiver/Isolation Amplifier is shown in Figure 3.4-3. The Audio/DME interface board provides the required signals for these two devices, respectively via the DME connector and the Audio connector.

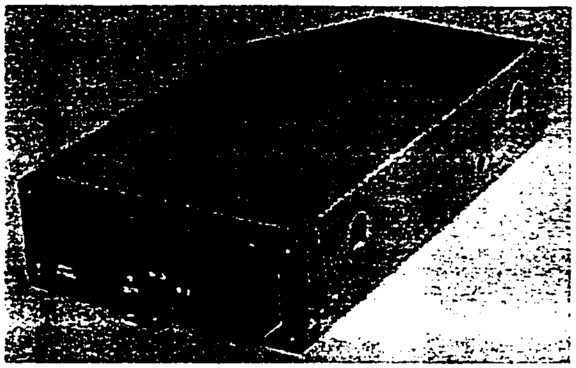


Figure 3.4-2 Simulated DME

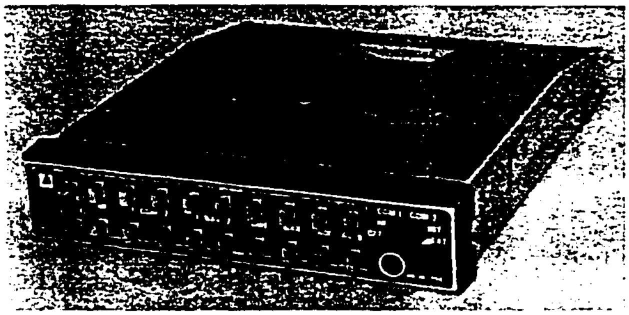


Figure 3.4-3 Simulated KMA

3.4.1. Audio/DME ISA Slot

The signals from the microprocessor system are communicated through the IAS bus. The Audio/DME interface board is used to process these signals for different applications. All the signals on the ISA bus could be classified into four groups listed in Table 3.4.1-1.

Group of Signals	Description
Power lines	GND, +5V, ±9V, and +185V
Data lines	D0-D7
Address lines	A0-A7
Control lines	$\overline{RD}$, $\overline{WR}$, CLK, Reset

Table 3.4.1-1 Signals on Audio/DME ISA Bus

The power lines have the same connection manner on every electrical board in the radio stack. In every connector, they are separated from data lines, address lines and control lines for safety reason. The data lines are either directly processed by Programmable Keyboard/Display Controller 8279 chips or are buffered for the further applications.

The A0 signal is sent to the DME 8279 chip as a control signal, and also to the Common Bus Connector for the use of other 8279 chips in other stations. The A1-A4 signals are sent to a Decoder 74LS154 for decoding the chip select signals to enable one 8279 chip in the radio stack at a time. [3.4] The complete information concerning the address decoding will be shown in the Chapter 4. The A5 and A6 signals are used to enable the Decoder 74LS154 as two strobe signals. The A7 signal that is also called LED HI is sent to a LED on the Audio/DME interface board, and also sent to the Audio connector to drive a LED on the KMA 24. The A7 signal is used to indicate that the microprocessor system is powered on.

Both $\overline{RD}$ and $\overline{WR}$ signals are sent to an OR Gate 74LS32. They OR a strobe signal of the Decoder 74LS154 to generate the reliable Read and Write signals since this kind of design can control the sequence of I/O commands and address decoding that generates the chip select signals for 8279 chips. It makes sure that the specified 8279 chip has been selected before the I/O starts. The CLK signal. like the A0 signal. is sent to the DME 8279 chip on the interface board, and also sent to the Common Bus Connector for 8279 chips in other stations. The Reset signal is sent to an Inverter 74LS04 to generate the usable Reset signal for all the 8279 chips.

3.4.2. Common Bus Connector

The Common Bus Connector is mounted on the Audio/DME interface board. It contains all the required signals to control the five stations, and interfaces them to the radio stack system.

The following signals can be found in the common bus connector: some signals from the ISA bus, proper signals for each station. the Audio signals. and the DME indicator signals. All these signals can be classified in five groups listed in Table 3.4.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Chip Select signals	CS0-CS7
Verified Control signals	$\overline{RD}$, $\overline{WR}$, CLK, A0, Reset. High Frequency (FRQ), Clear (CLR)
Audio signals	From several stations
Light signals	To several stations

Table 3.4.2-1 Signals in Audio/DME Common Bus Connector

The chip select signals are the outputs from the Decoder 74LS154 to enable specified 8279 chips in different stations. Both $\overline{RD}$ and $\overline{WR}$ signals are the reliable Read and Write signals, and the Reset signal is the usable Reset signal, as described in section 3.4.1. The CLK and A0 signals are from the ISA bus. The CLR signal is one output of the Decoder 74LS154. The FRQ signal is a 7.2KHz pulse signal generated by a LM555 timer circuit as shown in Figure 3.4.2-1. [3.5] Equation 3.1 is used to calculate the frequency of oscillation, where $R_A = 1K\Omega$, $R_B = 100K\Omega$ and $C = 1000pF$.

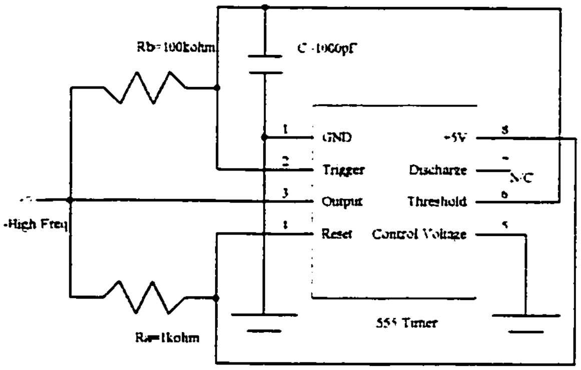


Figure 3.4.2-1 High Frequency Generation Circuit

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C} \qquad (3.1)$$

Both CLR and FRQ signals are used to acknowledge the status changes of the switches in the simulated system. Actually, the switches' Common lines are generated at FRQ 7.2KHz. It means that one Common line is generated every 0.14ms. A research to test the human reaction time had been done in Cornell University. [3.6] The result demonstrates that the typical human reaction times vary from around 80ms to about 400ms. [3.7] The FRQ 7.2KHz is therefore called as High Frequency. It is fast enough to enable the Common line, which corresponds to a rotated or pushed switch, before the pilot releases it.

The audio signals are amplified by the operational amplifier LM324, [3.8] and then sent to the Audio connector. The light signals are used for the backlights in the ADF, RNAV and ATC XPNDR stations. They are also sent to the Audio connector, but a separated cable will lead these light signals to the light board connector on the back panel.

3.4.3. DME Connector

The DME connector interfaces the DME indicator KDI 572 to the radio stack system. The signals in the DME connector could be classified in four groups listed in Table 3.4.3-1.

Group of Signals	Description
Power lines	GND, +5V, and +185V
Data lines	D0-D7
Switch signals	Common signal and switch signal
Display control signals	SCAN, SYNC, PROG, PHOTO

Table 3.4.3-1 Signals in Audio/DME DME Connector

The data lines are the signals from the DME 8279 data outputs. Before being sent to the DME indicator, they are buffered by the Buffer 7407. In addition, the buffered data signals D0-D3 could be used for the message display on the DME indicator.

The switch lines are used to acknowledge the status changes of the switches on the front panel. A Decoder 74LS139 decodes the scan lines SL0 and SL1 of the DME 8279 to generate the signals that affect what will be read from the return lines of the 8279. [3.9] The return lines RL0-RL3 of the DME 8279 are connected with the switches on the front panel to acknowledge the switch status. The switch technology used in this simulated radio stack system will be described in more detail in the ADF system.

The display control signals include the SCAN. SYNC. PROG and PHOTO. The SCAN signal is derived from the $\overline{BD}$ output of the 8279 and sent to the MUX CLK pin of an Anode Selection Johnson Counter (anode counter CD4022B). The SYNC signal is derived from the SL2 output of the 8279 and sent to the MUX RESET pin of the anode counter CD4022B. Both SCAN and SYNC signals are used to program the anode counter CD4022B. The PROG signal is used to program the cathode drivers. The PHOTO signal is generated by a photocell. A dimming circuit is used to automatically adjust the brightness by compensating the changes in ambient light level. Automatic dimming is controlled with both the photocell and the light dimming circuit. Dimming is accomplished by varying the duty cycle of the current programming circuitry of the cathode drivers. The anode counter CD4022B, the display circuit and the dimming circuit will be described in more detail in the ADF system.

3.4.4. Audio Connector

The Audio connector interfaces the KMA 24 Marker Beacon Receiver/Isolation Amplifier to the radio stack system. The signals on the Audio connector could be classified in four groups listed in Table 3.4.4-1.

Group of Signals	Description
Power lines	Audio GND
Volume control lines	From stations
LED signals	LED LO, LED HI
Light signals	LTNG LO, LTNG HI

Table 3.4.4-1 Signals in Audio/DME Audio Connector

The volume control lines, which are originally inputs from several stations with volume control function via the Common Bus Connector, are the audio signals amplified by the operational amplifier LM324. The LED signals including LED LO (power GND) and LED HI (A7) are the inputs from the microprocessor system via the ISA bus. The light signals LTNG LO and LTNG HI, which are the inputs from the light board, are used to drive the backlights in the ADF, RNAV, and ATC XPNDR stations.

The detail information of each pin in these three connectors is shown in Appendix 4. The drawing had also been made using the PROTEL software and saved as an electronic copy. The switchboard and the display board of the simulated DME Indicator KDI 573 have almost the same design as the original one: refer to the corresponding KING document. [3.10] The data sheet of the 8279 Programmable Keyboard/Display Controller is attached in Appendix 6. [3.11]

3.5. ADF

The simulated ADF system in the radio stack shown in Figure 3.5-1 has two functions listed in Table 3.5-1.

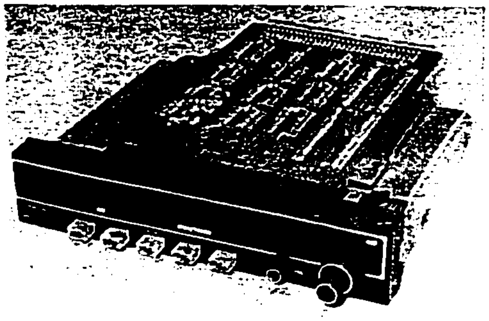


Figure 3.5-1 Simulated ADF

Function	Description
Display the ADF information	Messages and Numbers
Acknowledge the switches' status	1. Acknowledge switches' status via the ADF 8279 return lines 2. Send key codes to the microprocessor 3. Receive data from the microprocessor for display

Table 3.5-1 Functions of ADF in Radio Stack

To fully understand these two functions, the reverse engineering was started from three connectors shown in Table 3.5-2 on the ADF interface board. As long as the function of each pin in these connectors is determined. along the signal cable connections. the function of each chip on each board can be figured out and then the operations of the entire ADF system can be understood.

Connector	Description
ADF Bus Connector	Male 64-pin
ADF Switchboard Connector	Female 40-pin
ADF Display Board Connector	Female 16-pin

Table 3.5-2 Connectors on ADF Interface Board

3.5.1. ADF Bus Connector

The ADF bus connector is mounted on the ADF interface board. It interfaces the ADF station to the radio stack system. All the signals in the ADF bus connector can be classified in six groups listed in Table 3.5.1-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	D0-D7 to 8279
Chip select signal	CS1 for the ADF 8279
Control signals	$\overline{RD}$, $\overline{WR}$, CLK, A0, Reset, High Frequency (FRQ), Clear (CLR)
Audio signals	Audio GND, Audio OUT, Audio IN
Light signals	LTNG LO, LTNG HI

Table 3.5.1-1 Signals in ADF Bus Connector

The control signals $\overline{RD}$, $\overline{WR}$, CLK, A0, and Reset are used to control the ADF 8279 chip. The CLR signal is sent to a flip-flop 74LS74 to generate a reliable CLR signal for two Register 74LS173 chips that are used for scanning the status of the switches. The control signal FRQ is sent to a Decoder 74LS139 to enable one of the switch Common lines at a time. The Audio lines include Audio GND, Audio OUT, and Audio IN. These signals are used to control the ADF audio volume. The Light lines consist of LTNG LO and LTNG HI. They are used to drive the backlights in the ADF station.

3.5.2. ADF Switchboard Connector

The ADF switchboard connector is used to interface the ADF switchboard to the ADF interface board. The signals required to acknowledge the switch statuses are found in this connector. Since the ADF switchboard is integrated with the ADF display board, a display signal PHOTO is also found in this connector. All these signals can be classified in five groups listed in Table 3.5.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Switch signals	Pushbutton switches, knob switches
Audio signals	Audio GND. Audio OUT. Audio IN
Light signals	LTNG LO. LTNG HI
Display signals	PHOTO

Table 3.5.2-1 Signals in ADF Switchboard Connector

3.5.2.1. Pushbutton Switch

The switches used in the simulated ADF system can be classified in two groups: pushbutton switch and knob switch. For the pushbutton switches. the Dual-Decoder 74LS139 decodes the scan lines SL0 and SL1 of the ADF 8279 to generate a chip select signal that enables a Buffer 74LS244 to send the pushbutton switch signals on the return lines of the 8279. Each return line is originally connected to a High +5V power source with a pull-up resistor. When one switch is pushed. its switch line will be linked to its Common line. thus a Low signal will be read on its corresponding return line. According to the signals read from these return lines. a proper key code will be generate in the 8279 FIFO (first-in-first-out) buffer. The key code can be read from the FIFO buffer by the microprocessor and used to determine which switch is pushed. Any status change of the pushbutton switches in this radio stack is detected in this manner. The timing for the scanning of switch statuses will be described later. The return lines of the ADF 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.5.2.1-2.

Switch	Control signal
FRQ SW	RL0
FLT/ET SW	RL1
SET/RST SW	RL2
ADF SW	RL4
BFO SW	RL5
Pull-Up SW	RL6
ON SW	RL7

Table 3.5.2.1-1 Pushbutton Switches on ADF

3.5.2.2. Knob Switch

The knob switch includes three INC, DEC, MSD/LSD position lines, and two Common lines called Channel Common 1 and Channel Common 2. Only one of these two Common lines is generated at a time by decoding the FRQ 7.2KHz signal in the other channel of the Dual-Decoder 74LS139. When the knob stops at a position, one and only one position line will be linked to the Common source, and it becomes a Low signal from the original High signal. These three INC, DEC, MSD/LSD position lines are the control signals for two Register 74LS173 to respectively send Channel Common 1 and Cannel Common 2 on the RL0, RL1 and RL2, RL3 return lines of the ADF 8279. Rotating the knob therefore affects the signals that are read from the return lines of the 8279, thus the proper key code will be generated corresponding to the position of the knob. By reading two successive key codes, the difference between them is used to detect the rotating direction of the knob.

The reason for using this kind of design is because the pushbutton switches and the knob switch share the same RL0-RL2 lines of the ADF 8279. It is also a solution to use another 8279 specifically for the knob switch. This alternate solution is actually applied in the simulated RNAV system. The point is that the signals read from return lines of an 8279 determine the key code, and the key code is used to detect the status of the switches. As long as the system can generate sufficient key codes to differentiate every status of all the switches, the design is acceptable.

3.5.3. ADF Display Board Connector

The ADF display board connector is used to interface the ADF display board to the ADF interface board. The signals required for the gas discharge display are found in this connector. They can be classified in three groups listed in Table 3.5.3-1.

Group of Signals	Description
Power lines	GND
Data lines	Data outputs from ADF 8279
Display signals	PROG, SCAN, SYNC

Table 3.5.3-1 Signals in ADF Display Board Connector

3.5.3.1. Display Circuit

The display-timing diagram of the original KR 87 ADF is shown in Figure 3.5.3.1-1. The anode counter CD 4022B is clocked every 1ms by the original ADF microprocessor, so it turns on each of the anodes once every 8ms. This 1KHz display clock is sufficient because a 125Hz display rate is fast enough to produce a steady image to the eye. Each anode is turned on for 1ms, and during this time the microprocessor outputs the proper display data to the cathode drivers. These cathode drivers decode the BCD (binary coded data) and pull the corresponding cathode lines low, firing the proper segments or letters. In order to prevent a visual rippling effect, adjacent anodes are not fired in order, but rather the anode firing sequence was shuffled to produce a smooth continuous display. [3.12]

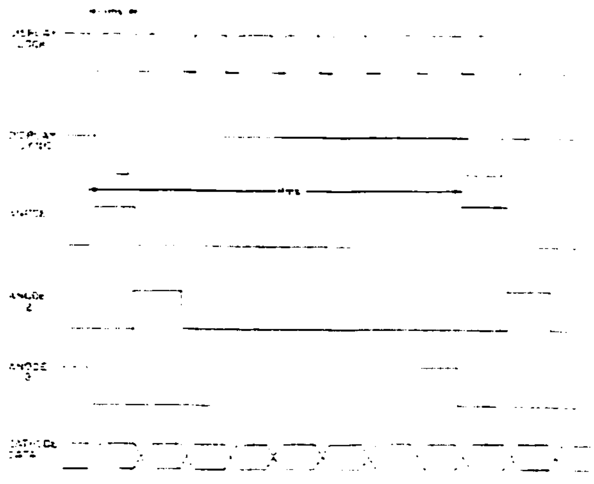


Figure 3.5.3.1-1 Display Timing Diagram [3.12]

In the simulated radio stack system, the CLK input of the 8279 is a 3MHz pulse signal that is generated by the microprocessor system. Since the 8279 controller is designed to run at 100KHz, a 5-bit programmable prescaler is provided within the 8279 to divide down the input frequency. The information about programming the prescaler will be described in Chapter 4. Since the fixed internal frequency is 100KHz, the cycle time t_{CY} of the 8279 should be $10\mu s$. In the timing diagram shown in Figure 5.5.3.1-2, it is found that the period of the $\overline{BD}$ signal is $640\mu s$ and its pulse width is $490\mu s$. This $\overline{BD}$ blank display output signal is sent to the anode counter CD4022B as the MUX CLK clock signal. In other words, the display clock is about 1.6KHz and it turns on each anode for $640\mu s$ in the simulated ADF system. This display clock is faster than the original one, but the anode counter CD4022B would operate in the same manner since its maximum clock input frequency can be up to 2.5MHz and its minimum clock pulse width is 200ns. [3.13]

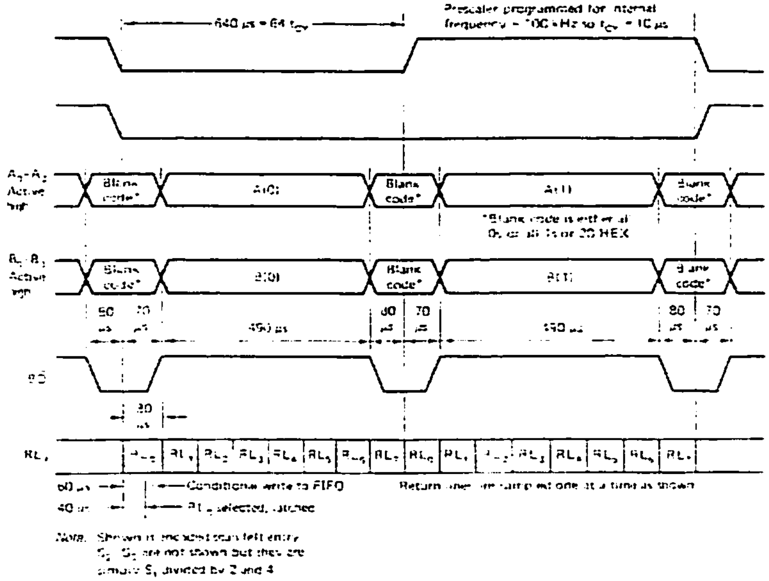


Figure 3.5.3.1-2 Keyboard and Display Signal Timing Diagram [3.14]

Figure 3.5.3.1-2 also shows that each return line is selected, latched and writes a corresponding key code to the FIFO buffer in $80\mu s$. It means that a key code would be generated every $80\mu s$, and the frequency would be 12.5KHz. In the simulated radio stack system, the microprocessor is an AMD D8085A-2 at 5MHz. This frequency is fast enough for the microprocessor to read a key code, generate new data and change display on the front panel without visible delay.

The scan lines of the 8279 operate in the encoded-mode by software programming that will be described in Chapter 4. Figure 3.5.3.1-3 shows the timing diagram of the encoded-mode scan-line signals. The SL2 signal, which is a input to a Retriggerable Monostrable Multivibrator 74LS123 shown in Figure 3.5.3.1-4, changes from High to Low every $8 \times 640\mu s$ that is 5.12ms. The truth table of the 74LS123, which is presented in Table 3.5.3.1-1, shows that a pulse output will be triggered when the SL2 input changes from High to Low. This pulse signal called SYNC is sent to the MUX RESET pin of the anode counter CD4022B. Therefore, the anode counter CD4022B will be reset every 5.12ms after firing all the eight anodes.

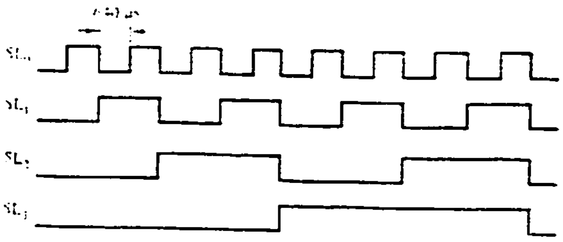


Figure 3.5.3.1-3 Encoded-Mode Scan Line Signals Timing Diagram [3.14]

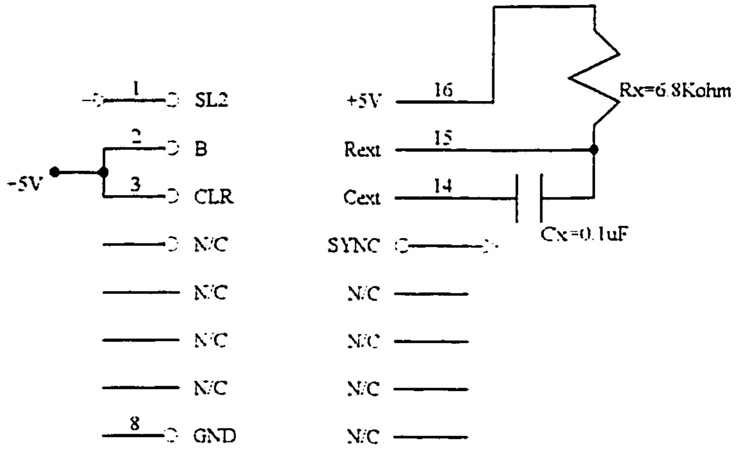
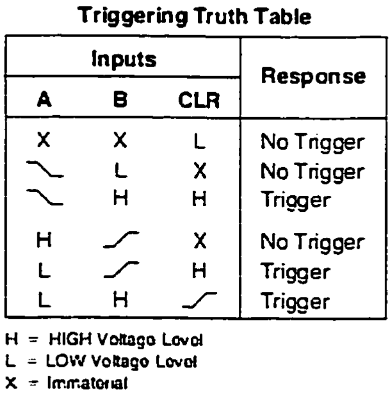


Figure 3.5.3.1-4 74LS123 Circuit



Triggering Truth Table

Inputs			Response
A	B	CLR	
X	X	L	No Trigger
↓	L	X	No Trigger
↓	H	H	Trigger
H	↑	X	No Trigger
L	↑	H	Trigger
L	H	↑	Trigger

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

Table 3.5.3.1-1 74LS123 Truth Table [3.15]

Figure 3.5.3.1-3 also shows the information about the timing of scanning switches' status. In this simulated radio stack system, the SL0 and SL1 scan lines are the inputs to a Decoder, and the outputs of the Decoder are used to enable Buffers or Registers to send the proper signals on return lines of an 8279. Since the comparison of the SL0 and SL1 inputs repeats the same result every $4 \times 640\mu s$, the Decoder enables each output every 2.56ms. Therefore, the signals on return lines of an 8279 might be changed every 2.56ms. Since a key code could be generated every $80\mu s$ as described previously, the 8279 would never miss a change on the return line. In other words, the 8279 could acknowledge every status change of the switches in this system.

Equation 3.2 [3.14] calculates the output pulse's width of the 74LS123, where K=0.28, $R_X = 6.8K\Omega$, $C_X = 0.1\mu F = 100000pF$. The calculated pulse width T_W is about $220\mu s$. It would be able to reset the anode counter CD4022B since its the minimum reset pulse width is 260ns. [3.12] It is this reset signal that shuffles the anode firing sequence, thus a smooth continuous display is produced.

$$T_W = KR_X C_X (1 + \frac{1}{R_X}) \qquad (3.2)$$

In summary, the anode counter CD4022B enables each anode every $640\mu s$, and it will be reset every 5.12ms and the reset period is $220\mu s$. All the stations of the simulated radio stack use the same multiplexed technique for the gas discharge display. The simulated system would provide the better display performance than the original one since the display clock is about 1.6KHz rather than 1KHz.

3.5.3.2. Dimming Circuit

The PHOTO and PROG signals are used in a dimming circuit. Varying both the duty cycle of the cathode drivers and the cathode current controls the brightness of the display. The cathode current varies over a 3:1 range, while the duty cycle varies over a 10:1 range. Thus, a 30:1 dimming range is possible. Maximum brightness occurs at high ambient light levels, while minimum brightness when the cockpit is dark. [3.12]

As shown in Figure 3.5.3.2-1, the amplifier I505A is configured as a voltage-to-current converter whose output current is determined by the voltage at the junction of R568, R573 and the photocell. The photocell acts as a light-sensitive resistor whose resistance decreases with increasing light intensity. The photocell forms a voltage divider with R573 at the input of I505A, as changes in light intensity vary the divider ratio and hence the current control voltage. Meanwhile, the display clock drives the positive input of the comparator I503B, which functions as an open-collector buffer. The display clock consists of narrow, negative going pulsed which pull down the output of the comparator, discharging C540. While the display clock is high, the open-collector output allows the current output of I505A to charge C540 linearly. When the voltage on the capacitor exceeds the threshold voltage on pin 9 of comparator I503C, its output pulls low, forward biasing diode CR516 and shunting the bias current to the cathode drivers that will be described in detail in Chapter 4.

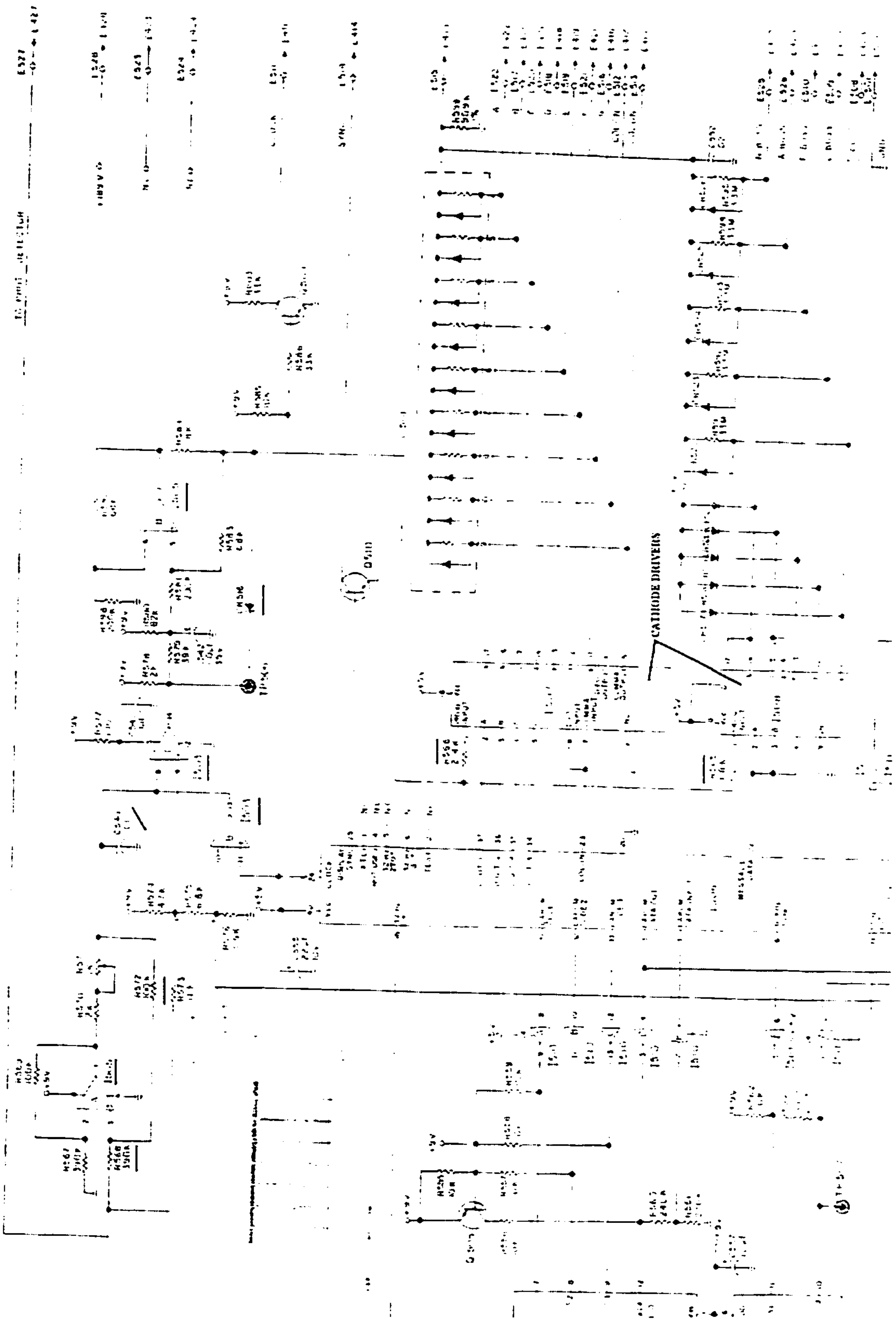


Figure 3.5.3.2-1 Dimming Circuit Design [3.12]

When high ambient light level exists as shown in Figure 3.5.3-6, the photocell exhibits a low resistance and little voltage appears at the input to I505A. Consequently, C540 charges very slowly and never reaches the threshold voltage of I503C before the display clock discharges it. Thus, I503C never gets tuned off, resulting in maximum duty cycle.

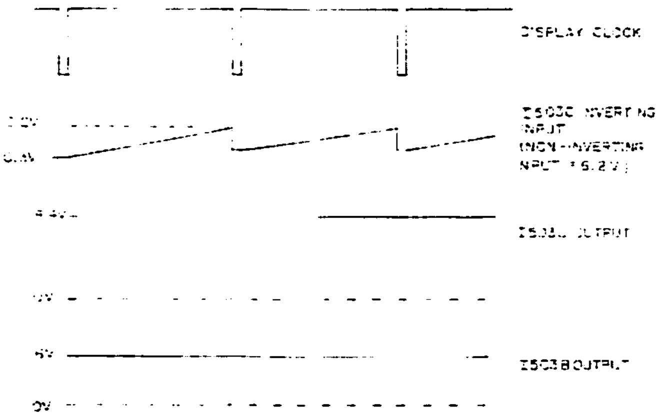


Figure 3.5.3.2-2 High Ambient Light Condition [3.12]

At low light levels shown in Figure 3.5.3-7, the photocell exhibits a high resistance, and a large current from I505A charges C540 rapidly. The voltage across the capacitor quickly triggers I503C, resulting in a very short duty cycle.

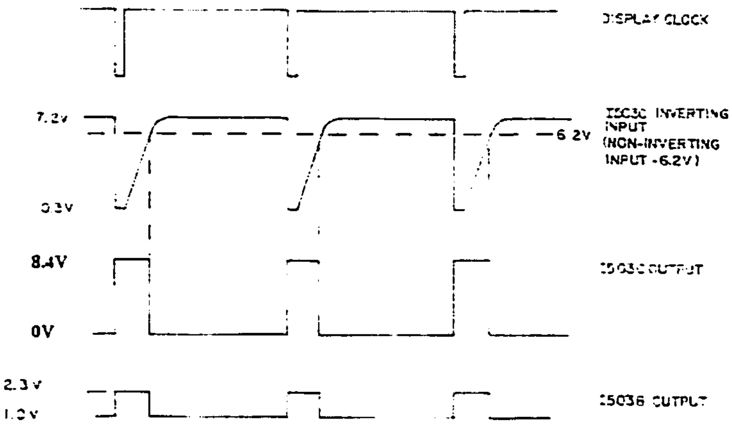


Figure 3.5.3.2-3 Low Ambient Light Condition [3.12]

Amplifier I505B is also configured as a voltage-to-current converter. It provides the proper current to the cathode drivers through programming resistors R566 and R565. Filtering the variable duty cycle pulse at the output of I503C derives the current control voltage. In this manner, current control is obtained with duty cycle control.

This kind of dimming circuit can be found on every display board in this simulated radio stack system.

The detail information of each pin in the three connectors is shown in Appendix 10. The drawing had been made using the PROTEL software and saved as an electronic copy. The display board and the switchboard of the simulated ADF KR 87 have almost the same design as the original one; refer to the corresponding KING document. [3.12] The data sheet of the Anode Selection Johnson Counter CD4022B is attached in Appendix 11. [3.13] The data sheet of the Retriggerable Monostrable Multivibrator 74LS123 is attached in Appendix 12. [3.15] The data sheet of the Cathode Driver DS8884A is attached in Appendix 13. [3.16]

3.6. RNAV

The RNAV system in the radio stack shown in Figure 3.6-1 has two functions listed in Table 3.6-1.

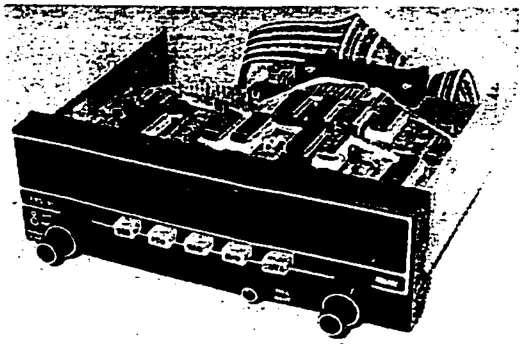


Figure 3.6-1 RNAV in Radio Stack

Function	Description
Display the RNAV information	Messages and Numbers
Acknowledge the switches' status	1. Acknowledge switches' status via the RNAV 8279 return lines 2. Send key codes to the microprocessor 3. Receive data from the microprocessor for display

Table 3.6-1 Functions of RNAV in Radio Stack

To completely understand these two functions, the reverse engineering was started from the three connectors shown in Table 3.6-2 on the RNAV interface board. As long as the function of each pin in these connectors is determined, along the signal cable connections, the function of each chip on each board can be figured out and then the operations of the entire RNAV system can be understood.

Connector	Description
RNAV Bus Connector	Male 64-pin
RNAV Switchboard Connector	Female 40-pin
RNAV Display Board Connector	Female 24-pin

Table 3.6-2 Connectors on RNAV Interface Board

3.6.1. RNAV Bus Connector

The RNAV bus connector is mounted on the RNAV interface board. It interfaces the RNAV station to the radio stack system. All the signals in the RNAV bus connector can be classified in six groups listed in Table 3.6.1-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	D0-D7 to 8279
Chip select signals	CS4 and CS5 for Two RNAV 8279 Chips
Control signals	$\overline{RD}$, $\overline{WR}$, CLK, A0, Reset, High Frequency (FRQ), Clear (CLR)
Audio signals	Audio GND, Audio OUT, Audio IN
Light signals	LTNG LO, LTNG HI

Table 3.6.1-1 Signals in RNAV Bus Connector

The control signals, the Audio lines, and the Light lines work in the exactly same manner as that in the ADF system. It is noted that two chip select signals are required for the two 8279 chips in the simulated RNAV system.

3.6.2. RNAV Switchboard Connector

The RNAV switchboard connector is used to interface the RNAV switchboard to the RNAV interface board. The signals required to acknowledge the switches' status are found in this connector. All the signals are classified in five groups listed in Table 3.6.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Switch signals	Pushbutton switches, knob switches
Audio signals	Audio GND, Audio OUT, Audio IN
Light signals	LTNG GND/LO, LTNG HI
Display signals	PHOTO

Table 3.6.2-1 Signals in RNAV Switchboard Connector

The switch lines are also classified in two groups: the pushbutton switches lines. and the knob switch lines. The pushbutton switches operate on the same principle as that in ADF system. The return lines of the RNAV 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.6.2-2.

Switch	Control signal
ON SW	RL0
RAD SW	RL1
Pull-Up SW	RL2
DATA SW	RL4
USE SW	RL5
RTN SW	RL6
CHK SW	RL7

Table 3.6.2-2 Pushbutton Switches on RNAV

In the simulated RNAV system. a second 8279 chip is specifically used to scan the knob switch rather than using only one 8279 for both knob switch and pushbutton switch. One decoder is also used to enable one Common line at a time. The return lines of the second 8279 scan the knob position signals P1L/P2L, P1R/P2R. LL/RL. LR/RR. WPT WP, MDE WP, FRQ KHz WP, and FRQ MHz WP. By using the same technique as that in the ADF system, the direction of rotating the knob switch can be determined.

3.6.3. RNAV Display Board Connector

The RNAV display board connector is used to interface the RNAV display board to the RNAV interface board. The signals required for the gas discharge display are found in this connector. All the signals are classified in four groups listed in Table 3.6.3-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	Data outputs from one RNAV 8279
Message data lines	Data outputs from the other RNAV 8279
Display signals	PHOTO, PROG, SCAN, SYNC

Table 3.6.3-1 Signals in RNAV Display Board Connector

The PHOTO, PROG, SCAN, and SYNC display control signals work in the same manner as that in the ADF station. It is noted that there are message data signals in the simulated RNAV system. Therefore, the second 8279 and two more cathode drivers are required in the simulated RNAV system.

The detail information of each pin in the three connectors is shown in Appendix 14. The drawing is available as a paper copy. The display board and the switchboard of the simulated RNAV KNS 81 have almost the same design as the original one; refer to the corresponding KING document. [3.17]

3.7. NAV/COMM

The NAV/COMM system in the radio stack shown in Figure 3.7-1 has two functions listed in Table 3.7-1.

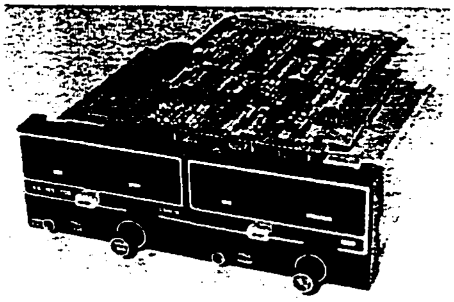


Figure 3.7-1 NAV/COMM in Radio Stack

Function	Description
Display the NAV/COMM information	Messages and Numbers
Acknowledge the switches' status	1. Acknowledge switches' status via the NAV/COMM 8279 return lines 2. Send key codes to the microprocessor 3. Receive data from the microprocessor for display

Table 3.7-1 Functions of NAV/COMM in Radio Stack

To fully understand these two functions, the reverse engineering was started from the three connectors shown in Table 3.7-2 on the NAV/COMM interface board. As long as the function of each pin in these connectors is determined, along the signal cable connections, the function of each chip on each board can be figured out and then the operations of the entire NAV/COMM system can be understood.

Connector	Description
NAV/COMM Bus Connector	Male 64-pin
NAV/COMM Switchboard Connector	Female 40-pin
NAV/COMM Display Board Connector	Female 16-pin

Table 3.7-2 Connectors on NAV/COMM Interface Board

3.7.1. NAV/COMM Bus Connector

The NAV/COMM bus connector is mounted on the NAV/COMM interface board. It interfaces the NAV/COMM station to the radio stack system. All the signals in the NAV/COMM bus connector can be classified in five groups listed in Table 3.7.1-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	D0-D7 to 8279
Chip select signals	CS2: CLK signal for Flip Flop 74LS374 CS3: to select 8279 chip
Control signals	$\overline{RD}$, $\overline{WR}$, CLK, A0, Reset.
Audio signals	Audio GND, Test SW, Test SW Common. Test Volume Hi, IDT Volume Hi

Table 3.7.1-1 Signals in NAV/COMM Bus Connector

The control signals and the Audio lines work in the same manner as that in the ADF system. It is noted that there is no FRQ or CLR signal required and there are more audio signals. The Test audio signals are used for COMM, and the IDT audio signals are used for NAV. It is also noted that two chip select signals are required: one is used as CLK signal for a flip-flop 74LS374, and the other one is used to select the NAV/COMM 8279. This flip-flop is used to process the message data signal called DP. Its function is similar to the second 8279 in the RNAV system.

3.7.2. NAV/COMM Switchboard Connector

The NAV/COMM switchboard connector is used to interface the NAV/COMM switchboard to the NAV/COMM interface board. Since the NAV/COMM switchboard is integrated with the NAV/COMM display board, the display signals are also found in this connector. All the signals can be classified in five groups listed in Table 3.7.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, +185V, Audio GND
Data lines	Data outputs from NAV/COMM 8279 DP signals
Switch signals	Pushbutton switch signals
Audio signals	Audio GND, Test SW, Test SW Common, Test Volume Hi, IDT Volume Hi
Display signals	CLK for dimming circuit

Table 3.7.2-1 Signals in NAV/COMM Switchboard Connector

An Anode Selection Johnson Counter is mounted on the NAV/COMM switchboard. It is specifically used to display the DP messages. The Test volume control is useable only when the Test volume control knob is pulled up, and the IDT volume control is useable only when the IDT volume control knob is pulled up.

The switch lines are used for the two pushbutton switches called XFER SW. The left XFER SW is used for COMM, and the right one is used for NAV. The return lines of the NAV/COM 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.7.2-2.

Switch	Control signal
ON SW	RL3
COMM 25K SW	RL4
NAV RAD SW	RL5
Right XFER SW	RL6
Left XFER SW	RL7

Table 3.7.2-2 Pushbutton Switches on NAV/COMM

3.7.3. NAV/COMM Display Board Connector

The NAV/COMM display board connector is used to interface the NAV/COMM display board to the NAV/COMM interface board. Since the NAV/COMM switchboard is integrated with the NAV/COMM display board, the switch signals are found in this connector. All the signals can be classified in two groups listed in Table 3.7.3-1.

Group of Signals	Description
Power lines	GND
Switch signals	Pushbutton switches, knob switches

Table 3.7.3-1 Signals in NAV/COMM Display Board Connector

The pushbutton switches operate on the same principle as that in the ADF station. The return lines of the NAV/COM 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.7.2-2.

The knob switch operates on the similar principle as that in the simulated ADF system. The three position lines are called the LEFT, PULSE, and RIGHT signals.

The detail information of each pin in these connectors is shown in Appendix 15. The drawing had also been made using the PROTEL software and saved as an electronic copy. The display board and the switchboard of the simulated COMM/NAV KX 165 have almost the same design as the original one; refer to the corresponding KING document. [3.18]

3.8. COMM

The COMM system in the radio stack shown in Figure 3.8-1 has two functions listed in Table 3.8-1.

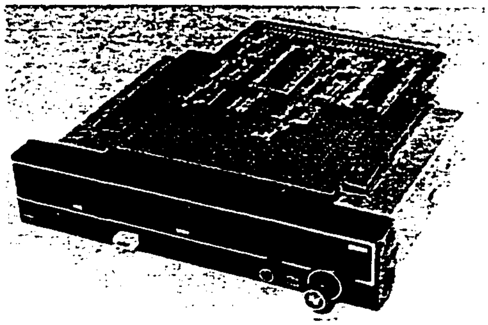


Figure 3.8-1 COMM in Radio Stack

Function	Description
Display the COMM information	Messages and Numbers
Acknowledge the switches' status	1. Acknowledge switches' status via the COMM 8279 return lines 2. Send key codes to the microprocessor 3. Receive data from the microprocessor for display

Table 3.8-1 Functions of COMM in Radio Stack

To fully understand these two functions, the reverse engineering was started from the two connectors shown in Table 3.8-2 on the COMM interface board. As long as the function of each pin in these connectors is determined, along the signal cable connections, the function of each chip on each board can be figured out and then the operations of the entire COMM system can be understood.

Connector	Description
COMM Bus Connector	Male 64-pin
COMM Switch/Display Board Connector	Female 40-pin

Table 3.8-2 Connectors on COMM Interface Board

3.8.1. COMM Bus Connector

The COMM bus connector is mounted on the COMM interface board. It interfaces the COMM station to the radio stack system. All the signals in the COMM bus connector can be classified in five groups listed in Table 3.8.1-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	D0-D7 to 8279
Chip select signals	CS6 to select 8279 chip
Control signals	$\overline{RD}$, $\overline{WR}$, CLK. A0. Reset, CLR
Audio signals	Audio GND, Test SW, Test SW Common. POT HI, POT LO

Table 3.8.1-1 Signals in COMM Bus Connector

The control signals and the Audio lines operate in the same manner as that in the ADF station. However, there is no FRQ signal required since fewer switches are used in the COMM station. The Test signals are active only when the volume control knob is pulled up. It is noted that the POT volume control is useable only when the COMM system operates at Test mode.

3.8.2. COMM Switch/Display Board Connector

The COMM switch/display board connector is used to interface the COMM switch/display board to the COMM interface board. All the switch and display signals can be found in this connector. All of these signals can be classified in five groups listed in Table 3.8.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, +185V, Audio GND
Data lines	4-bit data outputs from COMM 8279 1-bit data for DP signal
Switch signals	Pushbutton switches and knob switches
Audio signals	Audio GND, Test SW, Test SW Common. POT HI, POT LO

Table 3.8.2-1 Signals in COMM Switch/Display Board Connector

There is a pushbutton switch also called XFER SW. It works in the same manner as that in the NAV/COMM station. The return lines of the COMM 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.8.2-2.

Switch	Control signal
XFER SW	RL4
ON SW	RL5
Pull-Up 25K SW	RL6

Table 3.8.2-2 Pushbutton Switches on COMM

In this COMM station, there are fewer switches used. The knob switch operates on the similar principle as that for the pushbutton switches. There are sufficient return lines for all the switches. The return lines RL0-RL2 of the COMM 8279 are used to scan the INC, PULSE, and DEC position signals. These return lines are also originally linked to a high +5V power source. When the knob stops at a position, one and only one position line is connected to its Common line. A Low signal is read on the return line, and the corresponding key code will be changed. By reading two successive key codes, the difference between them is used to detect in which direction the knob is rotated. This kind of technique to detect the direction of rotating the knob switch is simpler than the one used the simulated ADF system.

Since there are only 4-bit data and 1-bit DP message signals required, only one 8279 chip is sufficient in the simulated COMM system.

The detail information of each pin in the two connectors is shown in Appendix 16. The drawing is available as a paper copy. The switch/display board of the simulated COMM KY 196 has almost the same design as the original one; refer to the corresponding KING document. [3.19]

3.9. ATC XPNDR

The ATC XPNDR system in the radio stack shown in Figure 3.9-1 has two functions listed in Table 3.9-1.

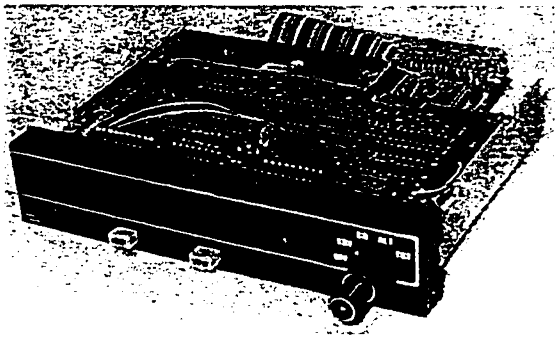


Figure 3.9-1 ATC XPNDR in Radio Stack

Function	Description
Display the ATC XPNDR information	Numbers
Acknowledge the switches' status	1. Acknowledge switches' status via the ATC XPNDR 8279 return lines 2. Send key codes to the microprocessor 3. Receive data from the microprocessor for display

Table 3.9-1 Functions of ATC XPNKR in Radio Stack

To completely understand these two functions, the reverse engineering was started from the two connectors shown in Table 3.9-2 on the ATC XPNDR interface board. As long as the function of each pin in these connectors is determined, along the signal cable connections, the function of each chip on each board can be figured out and then the operations of the entire ATC XPNDR system can be understood.

Connector	Description
ATC XPNDR Bus Connector	Male 64-pin
ATC XPNDR Switch/Display Board Connector	Female 40-pin

Table 3.9-2 Connectors on ATC XPNDR Interface Board

3.9.1. ATC XPNDR Bus Connector

The ATC XPNDR bus connector is mounted on the ATC XPNDR interface board. It interfaces the ATC XPNDR station to the radio stack system. All the signals in the ATC XPNDR bus connector can be classified in five groups listed in Table 3.9.1-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, and +185V
Data lines	D0-D7 to 8279
Chip select signals	CS0 to select 8279 chip
Control signals	$\overline{RD}$, $\overline{WR}$, CLK, A0, Reset
Light signals	LTNG HI and LTNG LO

Table 3.9.1-1 Signals in ATC XPNDR Bus Connector

The control signals and the lights signals works in the same manner as that in the ADF station. It is noticed that there is no FRQ and CLR signals required since fewer switches are used in the simulated ATC XPNDR system.

3.9.2. ATC XPNDR Switch/Display Board Connector

The ATC XPNDR switch/display board connector is used to interface the ATC XPNDR switch/display board to the ATC XPNDR interface board. All the switch and display signals are found in this connector. All the signals can be classified in five groups listed in Table 3.9.2-1.

Group of Signals	Description
Power lines	GND, +5V, +9V, +185V
Data lines	8-bit data outputs from COMM 8279
Switch signals	Pushbutton switches, knob switches
Display signals	PHOTO, PROG SCAN and SYNC
Light signals	LTNG HI and LTNG LO

Table 3.9.2-1 Signals in ATC XPNDR Switch/Display Board Connector

The return lines of the ATC XPNDR 8279 that are used for scanning different pushbutton switches are respectively described in Table 3.9.2-2. It is noted: VFR SW and V SW share the same RL7 line; VFR SW and IDT SW share the same Common line. When the VFR SW is pushed, the IDT SW switch line will be forced to lead to the Common line. Therefor, both the RL6 and RL7 return lines are read as a Low signal. Table 3.9.2-3 shows how to determine which pushbutton switch is pushed.

Switch	Control signal
IDT SW	RL6
VFR SW and V SW	Shared RL7

Table 3.9.2-2 Pushbutton Switches on ATC XPNDR

Switch	Detected signal
IDT SW	RL6 led to Common
VFR SW	RL6, RL7 led to Common
V SW	RL7 led to Common

Table 3.9.2-3 Determine Pushbutton Switches on ATC XPNDR

For the INC/DEC knob, the return lines RL0-RL2 of the ATC XPNDR 8279 are used to scan the position lines INC/DEC0, INC/DEC1, INC/DEC2. The direction of rotating the knob switch can be determined by using the same technique as that in the COMM station.

For the mode selection knob, the return lines RL3-RL5 of the ATC XPNDR 8279 are used to scan the position lines Mode1, Mode2 and Mode3. Table 3.9.2-4 shows how to determine which mode is selected for the simulated ATC XPNDR system.

Mode	Detected signal
OFF	Mode1 led to Common
SBY (standby mode)	Mode1, Mode2 led to Common
ON	Mode1, Mode2, Mode 3 led to Common
ALT (altitude mode)	Mode2, Mode3 led to Common
TST (test mode)	Mode2 led to Common

Table 3.9.2-4 Determine Selected Mode on ATC XPNDR

The detail information of each pin in the connectors is shown in Appendix 17. The drawing had also been made using the PROTEL software and saved as an electronic copy. The switch/display board of the simulated XPNDR KT 79 has almost the same design as the original one; refer to the corresponding KING document. [3.20]

3.10. Summary

Upgrading the avionic system in the Concordia flight simulator is one of the objectives of this project. Without the reverse engineering, the upgrade would not have been possible because several important documents of the radio stack were not available any longer. This chapter described the hardware reverse engineering work for the simulated radio stack system.

The maintenance manuals from the radio stack manufacturer BINDEX/KING Company were the fundamental materials for the design of the simulated radio stack system. Without the help of them, the reverse engineering would not have been able to progress. This simulated radio stack system consists of ADF, RNAV, NAV/COMM, COMM and ATC XPNDR stations with the associated DME, MBR and the two navigation indicators PNI and RMI.

Most of the stations in the radio stack contain three electrical printed circuit boards: interface board, display board and switchboard. In the ADF section, the switch circuit, display circuit and dimming circuit were described in detail. Similar designs can be found in every station of the simulated radio stack system. All the interface boards, which were custom designed and built, interfaced the simulated stations to a microprocessor system that was used to control the entire radio stack system.

The reverse engineering was always started from the connector on each board. As long as the function of each pin in the connector was determined, along the signal cable connections, the function of each chip on each board could be figured out. Through the reverse engineering, the hardware configuration and the operation principles of the simulated radio stack system are presented.

CHAPTER 4
UPGRADED RADIO STACK SYSTEM AND RADIO STACK MODULE

One of the objectives of this project is to upgrade the avionic system, simulate and model the radio stack system in the Concordia flight simulator. The reverse engineering presented in Chapter 3 gives the possibility to upgrade the simulated radio stack system. The required software for simulation and modeling includes a radio stack module and a navigation model. The so-called radio stack module is a software program used to control and drive the radio stack hardware so that it can be used in the simulated cockpit environment. The so-called navigation model is a software package used to implement the radio stack system's navigation functions in a flight simulation.

This chapter presents the upgraded radio stack system and radio stack module. The navigation model will be described in Chapter 5.

4.1. Upgrade Radio Stack System

4.1.1. Original Radio Stack System

In the original radio stack system as illustrated in Figure 4.1.1-1, the microprocessor system communicates with the flight model computer via a serial RS232 link at 9600 baud, and communicates with the radio stack via an 8-bit ISA bus. There was the existing radio stack module coded in Assembly language stored in an EAROM (Electrically Alterable Read Only Memory) mounted on the microprocessor board.

This radio stack module initializes and drives the entire simulated radio stack system. It is able to report the status of the radio stack to a flight model, and receive commands from the flight model via the RS232 port. This module is also able to acknowledge the status of switches on the front panel of each station, and change display according to the status changes via the ISA bus.

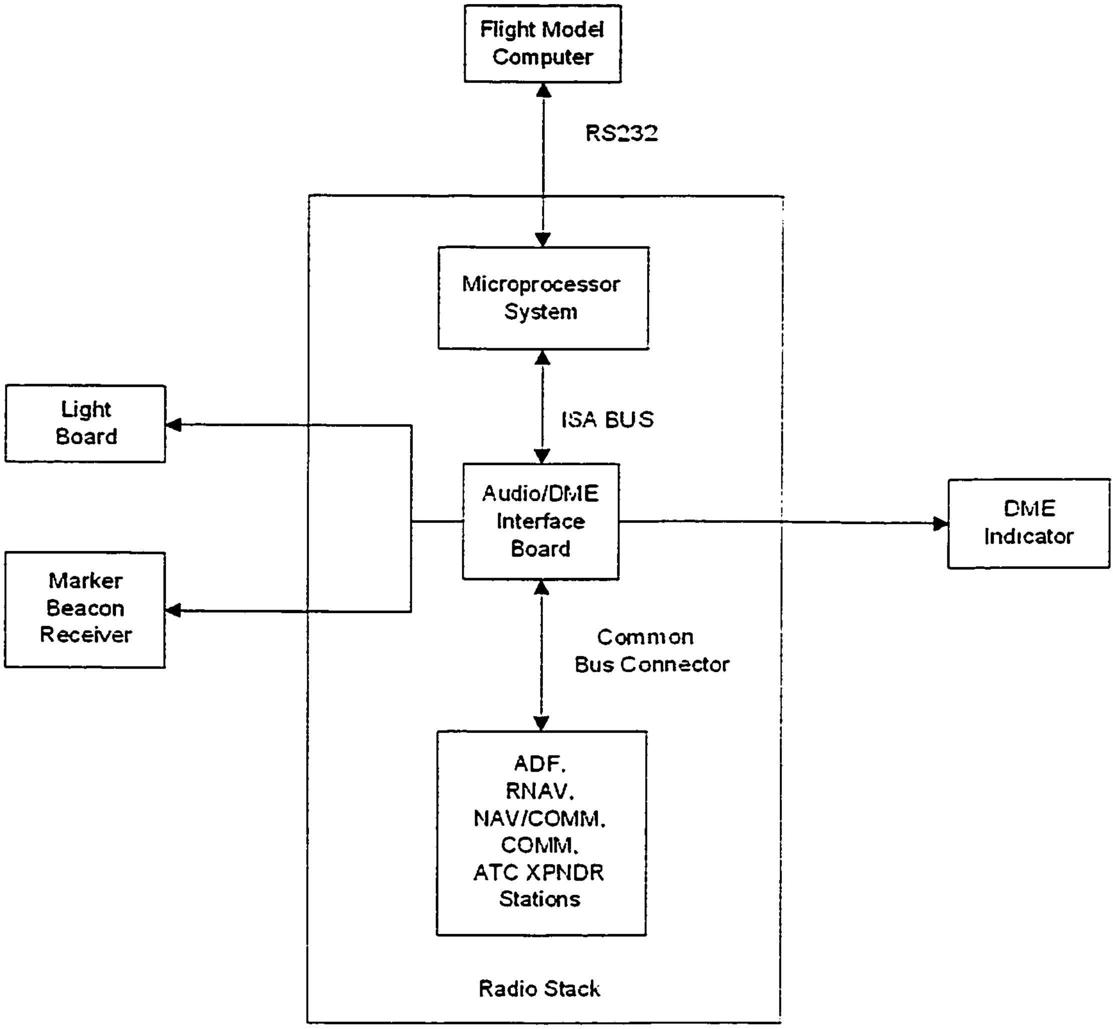


Figure 4.1.1-1 Original Radio Stack System Interface

Unfortunately, nothing was found in the EAROM during the reverse engineering for the microprocessor system. It was obvious that the chip had been destroyed. In keeping the same design philosophy, a well-made microprocessor board must be purchased from a professional microprocessor system manufacturer. A decision was made not to pursue this kind of design. An upgraded radio stack system will replace the original one, and the radio stack control module will be upgraded as well.

4.1.2. Upgraded Radio Stack System

In the upgraded system illustrated in Figure 4.1.2-1, a custom interface board is mounted in the radio stack. It interfaces the radio stack to the flight model computer via a high-speed communication port that will be described later. The flight model computer instead of the microprocessor system drives the simulated radio stack system while running the flight model. Modern computers, which offer high performance at a low cost, have created the possibility to implement this new design.

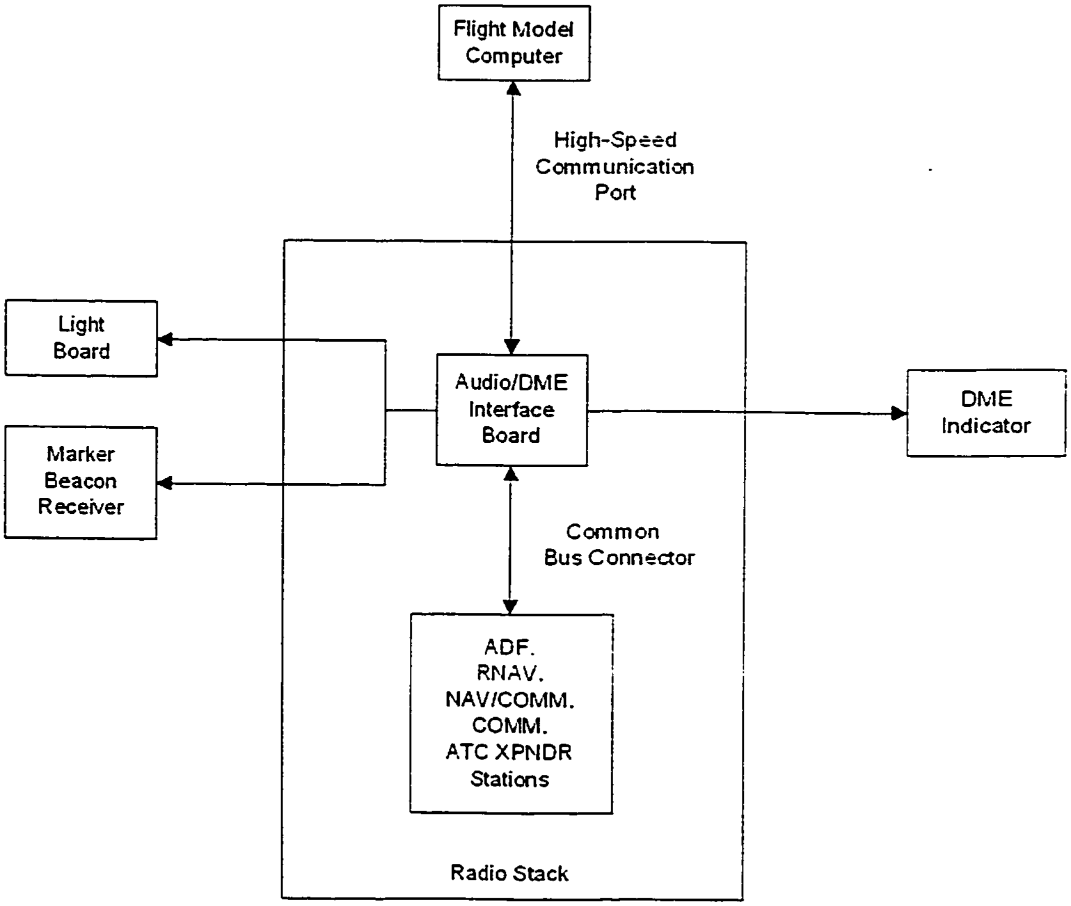


Figure 4.1.2-1 Upgraded Radio Stack System Interface

In the upgraded system, the radio stack module will be stored in the flight model computer. It has a set of databases including an Address database, a key code database and a display database. The program is able to acknowledge the requirements from the navigation model, and then send the proper data to the radio stack for display. It is also able to acknowledge the status changes of the switches on the front panel of each station, then process the changes and send the new data for display. If the radio stack module and the navigation model were integrated with the flight dynamic model, an advanced flight simulation would be implemented with the cooperation of the radio stack system.

4.2. Radio Stack Module

In order to control and drive the radio stack in the simulated cockpit environment, the radio stack module must be able to program several control chips, and the high-speed communication protocol between the radio stack and the flight model computer needs to be selected as well.

In this project, one radio stack module was coded in VC++ so that it could be integrated with the flight dynamic model in the future research, and the other radio stack module in LabVIEW was programmed while developing the upgraded system.

4.2.1. Programming Control Chips

4.2.1.1. Address Decoder

The 4-line to 16-line decoder 74LS154 is mounted on the Audio/DME interface board in the back portion of the simulated radio stack. It is used for decoding the four Address lines A1-A4 to generate the chip select signals for the 8279 chips and some other control signals. Table 4.2.1.1-1 gives the necessary information to build the Address database of the radio stack module. [4.1]

INPUTS				OUTPUTS									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
A4	A3	A2	A1	CS0	CS1	CS6	CS2	CS3	CS5	CS4	CLR	CS7	CLK
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

Table 4.2.1.1-1 Address Decoder 74LS154 Application

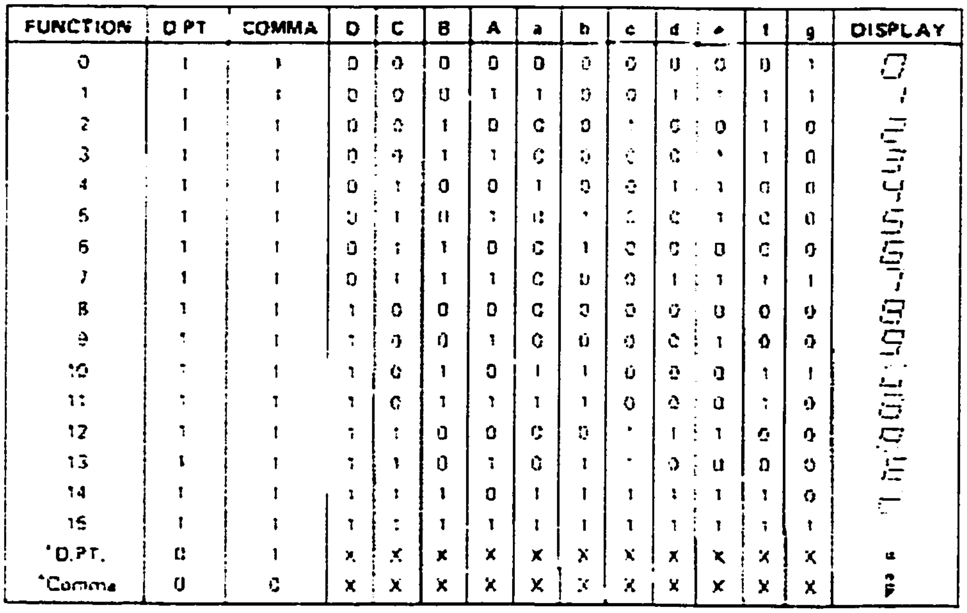
Table 4.2.1.1-2 shows the chip select signals are used for which specified 8279 chip and the control signals are used for which specified chips on the interface boards. In the radio stack module, the program will handshake with each station in turn.

Signal	Controlled Chip	Function Description
CS0	ATC XPNDR 8279	Switches and display
CS1	ADF 8279	Switches and display
CS2	NAV/COMM Flip-flop 74LS374	Message display
CS3	NAV/COMM 8279	Switches and display
CS4	RNAV 8279 No.1	Pushbutton switches Number display
CS5	RNAV 8279 No.2	Knob switch Message display
CS6	COMM 8279	Switches and display
CS7	DME 8279	Switches and display
CLR	ADF, RNAV, COMM Flip-flop 74LS74 chips	Switches
CLK	DME Flip-flop 74374	Message display

Table 4.2.1.1-2 Chip Select and Control Signals from 74LS154

4.2.1.2. Cathode Driver

The high voltage cathode decoder/driver DS8884A is mounted on every display board in the simulated radio stack. It is used to decode the four lines of BCD code inputs and drive the seven-segment digits of gas-filled readout displays. Table 4.2.1.2-1 gives the necessary information to build the Display database of the radio stack module. The detail date sheet of the DS8884A is attached in Appendix 13. [4.2]



FUNCTION	D PT	COMMA	D	C	B	A	a	b	c	d	e	f	g	DISPLAY
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	1	1	0	0	1	1	1	1	1
2	1	1	0	0	1	0	0	0	1	0	0	1	0	2
3	1	1	0	0	1	1	0	0	0	0	1	1	0	3
4	1	1	0	1	0	0	1	0	0	1	1	0	0	4
5	1	1	0	1	0	1	0	1	0	0	1	0	0	5
6	1	1	0	1	1	0	0	1	0	0	0	0	0	6
7	1	1	0	1	1	1	0	0	0	1	1	1	1	7
8	1	1	1	0	0	0	0	0	0	0	0	0	0	8
9	1	1	1	0	0	1	0	0	0	0	1	0	0	9
10	1	1	1	0	1	0	1	1	0	0	0	1	1	
11	1	1	1	0	1	1	1	1	0	0	0	1	0	
12	1	1	1	1	0	0	0	0	1	1	1	0	0	
13	1	1	1	1	0	1	0	1	1	0	0	0	0	
14	1	1	1	1	1	0	1	1	1	1	1	1	0	
15	1	1	1	1	1	1	1	1	1	1	1	1	1	
*D.PT.	0	1	X	X	X	X	X	X	X	X	X	X	X	.
*Comma	0	0	X	X	X	X	X	X	X	X	X	X	X	,

Table 4.2.1.2-1 Truth Table of DS8884A

Another kind of cathode driver DM8889N is used to drive gas discharge display devices from signals originating from MOS or TTL circuitry. It is used for the message display on the front panel. For example, in order to display the gas-filled readout message "VOR" on the RNAV front panel, the DM8889 will pull the corresponding outputs low since each letter of this message is connected to one of its outputs. The detail date sheet of the equivalent chip DI-230 is attached in Appendix 18. [4.3] Both kinds of cathode drivers receive the display data inputs from the 8279 programmable keyboard/display controllers.

4.2.1.3. Programmable Keyboard/Display Controller

The 8279 programmable keyboard/display interface controller is able to scan and encode up to a 64-key keyboard and controls up to 16-digit seven-segment displays. The key code is stored in a built-in first-in-first-out (FIFO) 8-character buffer. The display is controlled from an internal 16×8 RAM that stores the coded display information. [4.4] In the simulated radio stack system, an imaginary keyboard is formed by all the switches instead of a real keyboard connected to the 8279, but the operation remains the same.

A block diagram of the 8279 is shown in Figure 4.2.1.3-1(a) and its pin layout in Figure 4.2.1.3-1(b). [4.4] This diagram shows that there are four signal sections: the MPU interface, the key data inputs, the display data outputs, and the scan lines that are used by both the keyboard and display.

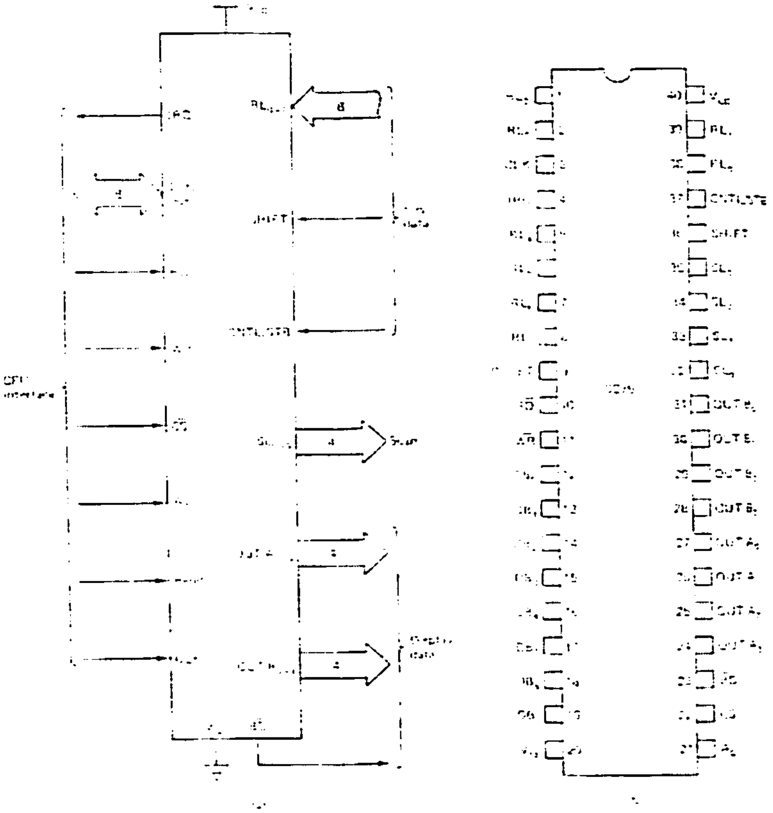


Figure 4.2.1.3-1 (a) Block diagram of the 8279 (b) Pin layout [4.4]

The bus interface of the 8279 consists of the eight data bus lines DB_0 through DB_7. These are the lines over which the MPU outputs data to the display, inputs key codes, issues command to the controller, and reads status information from the controller. Other signals found at the interface are the read ($\overline{RD}$), write ($\overline{WR}$), chip-select ($\overline{CS}$), and address buffer (A_0) control signals. They are the signals that control the data bus transfers that take place between the microprocessor and 8279. Another signal in this interface is interrupt request (IRQ). Since the simulated radio stack system does not operate with interrupt function, the IRQ signal is not used in this project. [4.4]

The key data lines include the eight return lines RL_0 through RL_7. These lines receive inputs from the keyboard. If logic 0 is detected at a return line, a key code will be generated to indicate which key is pressed. The key code format will be described later. This key code input is first debounced and loaded into an 8×8 key code FIFO (first-in-first-out) within the 8279, then read by the MPU. [4.4] There are shift (SHIFT) and control/strobed (CNTL/STB) signals in this section, but they are not used in this project.

The display data lines include two 4-bit output ports, $OUTA_0$ through $OUTA_3$ and $OUTB_0$ through $OUTB_3$, that are used as display segment drive lines. Segment data that are output on these lines are held in a dedicated display RAM area within the 8279. This RAM is organized as 16×8 and must be loaded with segment data by the MPU. [4.4]

The scan lines are used as row-drive signals for the keyboard and digit-drive signals for the display. There are just four of these lines, SL_0 through SL_3. However, they can be configured for different modes of operation through software: the decoded mode and the encoded mode. [4.4] In this project, the 8279 controllers operate in encoded scan mode as described in Section 3.5.3.1.

4.2.1.3.1. Command Words

The 8279 controller must be initialized, and its operation is configured through software. When $\overline{WR} = 0$ and $A0 = 1$, a command data on the input data lines is loaded into the corresponding control register of the 8279. As shown in Table 4.2.1.3.1-1, the first 3 bits of the command data sent to the control register selects one of the eight command words.

D7	D6	D5	Function	Purpose
0	0	0	Mode set	Selects the number of display positions, type of key scan...
0	0	1	Clock	Programs internal CLK, sets scan and debounce times.
0	1	0	Read FIFO	Selects type of FIFO read and address of the read.
0	1	1	Read Display	Selects type of display read and address of the read.
1	0	0	Write Display	Selects type of write and the address of the write.
1	0	1	Display write inhibit	Allows half-bytes to be blanked.
1	1	0	Clear	Clears the display or FIFO
1	1	1	End interrupt	Clears the IRQ signal to the microprocessor.

Table 4.2.1.3.1-1Eight Command Words of 8279 [4.4]

4.2.1.3.1.1. The Command Word 0

When the first three bits of the command data is 000, the command word 0 will be loaded into the control register. This command word is used to set the mode of operation for the keyboard and display. Its format is 000DDMMM.

The 3 MSBs of the command word are always 0 that indicates its mode set function. The next 2 bits, DD, is used to set the mode of display. The truth table is shown in Table 4.2.1.3.1.1-1. This DD field selects either 8-digit or 16-digit display, and whether new data are entered to the rightmost or leftmost display position. [4.4]

DD	Function
00	8-digit display with left entry
01	16-digit display with left entry
10	8-digit display with right entry
11	16-digit display with right entry

Table 4.2.1.3.1.1-1 Display Modes of 8279 [4.4]

The 3 LSBs are used to set the keyboard mode. The truth table is shown in Table 4.2.1.3.1.1-2. In the Encoded mode, the scan line outputs are active-high and follow binary bit pattern 0-7 or 0-15. In the Decoded mode, the scan line outputs are active-low (only one low at any time). Its pattern output will be 1110, 1101, 1011, and then 0111. In the strobed mode, the rising edge on the CNTL/STB input pin strobes data from the return lines into an internal FIFO for reading. In the 2-key lockout mode, it prevents 2 keys from being recognized if pressed simultaneously. In the N-key rollover mode, it accepts all keys pressed from 1st to last. [4.4]

KKK	Function
000	Encoded keyboard with 2-key lockout
001	Decoded keyboard with 2-key lockout
010	Encoded keyboard with N-key rollover
011	Decoded keyboard with N-key rollover
100	Encoded sensor matrix
101	Decoded sensor matrix
110	Strobed input, encoded keyboard scan
111	Strobed input, decoded keyboard scan

Table 4.2.1.3.1.1-2 Keyboard Modes of 8279 [4.4]

In the radio stack module, the command word 0 is programmed to 0EH. The 8279 will operate at Strobed input. encoded keyboard scan. and 16-digit display with left entry display mode in the simulated radio stack system.

4.2.1.3.1.2. The Command Word 1

When the first three bits of the command data is 001, the command word 1 will be loaded into the control register. This command word is used to set the operation frequency of the 8279. Its format is 001PPPPP.

This clock command word programs the internal clock driver. The code PPPPP is called the 5-bit programmable prescaler and is used to divide down the input frequency (CLK) to achieve the desired operating frequency. For example, since the 8279 controller is designed to operate at 100KHz, a prescaler of 01010 is required for a 1 MHz CLK input. [4.4] In the radio stack module, the command word 1 is programmed to 3EH because the input CLK signal is 3MHz and the divider is 30.

4.2.1.3.1.3. The Command Word 2

When the first three bits of the command data is 010, the command word 2 will be loaded into the control register. This command word is used to issues a read FIFO command. Its format is 010Z0AAA.

The key code FIFO buffer is read-only. Before the MPU can access it, a read FIFO command must be issued to the 8279. This read FIFO control word selects the address (AAA) of a keystroke from the FIFO buffer (000 to 111). Z selects auto-increment for the address. When the 8279 is set up for keyboard scanning, the Z and AAA bits are don't-care states, and the command word 2 will be therefore 40H. [4.4]

In the radio stack module, the command word 2 is programmed to 40H. It will initialize read (input) cycles to the 8279. In each subsequent read bus cycle, the key code at the top of FIFO buffer will be read into MPU.

4.2.1.3.1.4. The Command Word 3

When the first three bits of the command data is 011, the command word 3 will be loaded into the control register. This command word is used for MPU to read the contents of the display RAM through the data port. Its format is 011ZAAAA. Z selects auto-increment, so subsequent reads go to subsequent display positions. AAAA is the address of the first location to be accessed. [4.4]

In the upgraded radio stack system, the display data feedback is not required. The radio stack module outputs data for display, but does not need to read the display RAM. Therefore, this command word does not need to be sent.

4.2.1.3.1.5. The Command Word 4

When the first three bits of the command data is 100, the command word 4 will be loaded into the control register. This command word is used for MPU to send new data to the display RAM. Its format is 100ZAAAA, where Z and AAAA have the same definitions as the command word 3.

In the radio stack module, the command word 4 is programmed to 90H. It enables the auto-increment addressing and the Address 0000 will be the first location in the display RAM that is accessed.

4.2.1.3.1.6. The Command Word 5

When the first three bits of the command data is 101, the command word 5 will be loaded into the control register. This command word is used to enable the display Write inhibit or blanking. Its format is 1010WWBB.

This display Write inhibit control word WW inhibits writing to either the leftmost 4 bits of the display (WW=10 corresponds to $OUTA_0$ - $OUTA_3$) or the rightmost 4 bits (WW=01 corresponds to $OUTB_0$ - $OUTB_3$). The BB works similarly except that they blank (turn off) half of the output pins. [4.4]

In the radio stack module, the system does not require these special functions. Therefore, this command word does not need to be sent.

4.2.1.3.1.7. The Command Word 6

When the first three bits of the command data is 110, the command word 6 will be loaded into the control register. This command word is used to initialize the complete display RAM, FIFO status, and the interrupt-request output line. Its format is 110CCCFA.

If bit F is set, the FIFO status is cleared and the IRQ output is reset. If CCC is 100, it clears all the display RAM locations to become 00000000. If CCC is 010, the display RAM locations become 00100000. If CCC is 011, the display RAM locations become 11111111. Bit A is called the Clear All bit, which clears the FIFO status and all display RAM locations. [4.4]

In the radio stack module, the command word 6 is programmed to D3H. It initializes the FIFO status, the display RAM locations, and the IRQ line to all zeros.

4.2.1.3.1.8. The Command Word 7

When the first three bits of the command data is 111, the command word 7 will be loaded into the control register. Only bit 4 of the command word 7 is functional. This bit is an enable signal for what is called the special error (S/E) mode. When this mode is enabled and the keyboard has N-key rollover selected, a multiple-key depression causes the S/E flag of the FIFO status register to be set. This flag can be read by the MPU through software. [4.4]

In this project, the upgraded system operates at the handshaking mode and not in the special error mode. Therefore, this command word does not need to be sent.

4.2.1.3.2. Programming Keyboard/Display Controller

The initialization process is shown in Figure 4.2.1.3.2-1. The Clock must be programmed first. Since a 3MHz frequency drives the CLK input, the PPPPP is programmed to 3EH. In the simulated radio stack system, the imaginary keyboard is encoded type and external decoders are used to drive the switches that form the keyboard. After programming the keyboard type, the FIFO buffer should be cleared. Finally, a procedure is needed to read data from the FIFO to determine if a switch has been operated during initialization.

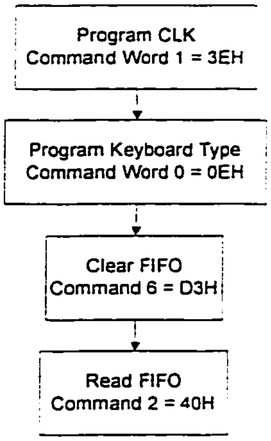


Figure 4.2.1.3.2-1 8279 Initialization Flowchart

To determine if a switch is pushed or the knob stops at which position, first of all, the FIFO status register needs to be checked. When $\overline{RD} = 0$ and $A0 = 1$, the contents of the FIFO status word as shown in Figure 4.2.1.3.2-2 will be read by the MPU. If there is nothing wrong as shown in the FIFO status register, a key code will be read after it.

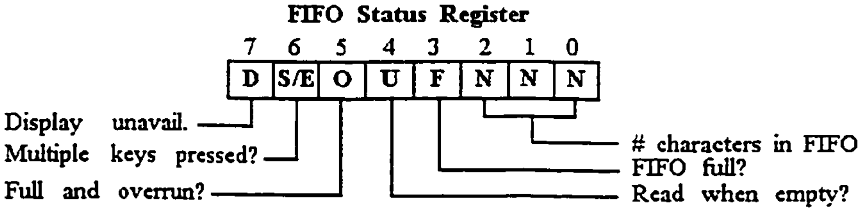


Figure 4.2.1.3.2-2 FIFO Status Register [4.4]

As shown in the left diagram of Figure 4.2.1.3.2-3, row and column numbers correspond respectively to the scan and return rightmost 6 bits in the scanned key code read from the 8279. The CT and SH indicate whether the control or shift keys are pressed. Since a real keyboard does not exist in the simulated radio stack system, the scanned key code is not used in this project. As shown in the right diagram of Figure 4.2.1.3.2-3, the Strobed key code indicates the state of the RLx bits at the time of the rising edge on the strobe input pin. This kind of Strobed key code is used to determine the status of the switches in this project.

Scanned Keyboard Code

7	6	5	4	3	2	1	0
CT	SH	Scan			Return		

Strobed Keyboard Code

7	6	5	4	3	2	1	0
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

Figure 4.2.1.3.2-3 Key Code Format [4.4]

4.2.2. Communication Protocol

In the original system, the microprocessor AMD D8085A-2 that is used to drive the radio stack operates at 5MHz, and a RS232 port interfaces it to the flight model computer at 9600 baud. In the upgraded system, the flight model computer replaces the microprocessor system to drive the radio stack. As described in Chapter 3, a key code might be generated every $80\mu s$, and the frequency is 12.5KHz. The flight model computer is required to read a key code, generate new data and change display without visible delay. Therefore, the communication rate between the computer and the radio stack should be much faster than 12.5KHz.

4.2.2.1. RS232, USB and Parallel Port

As shown in Chapter 3, the simulated radio stack requires 8-bit data lines, 8-bit address lines and 4-bit control lines from the original microprocessor system. The communication between them is via the ISA bus at 5MHz.

It is obvious that the original RS232 at 9600 baud cannot be used in the upgraded system. Even though the data flow rate of RS232 could be up to 115.2K baud, [4.5] it still seems too slow for this application. The USB is the advanced serial communication port. Its data transfer rate could be maximum 480Mb/s in the new USB 2.0 standard. [4.6]

The reason not to use a RS232 or a USB port in this project is because of their serial communication principle. A serial port is harder to interface than a parallel port. In most cases, the device connected to the serial port will need the serial transmission converted back to parallel so that it can be used. It requires a quite complicated hardware design on the new custom interface board that will be mount on the ISA bus in the back portion of the radio stack.

A parallel port could be used to transfer data at a rate up to 2Mb/s, [4.7] which is faster than the RS232 port, but slower than the USB port. In the beginning of this project, a parallel protocol was used to test the radio stack hardware and the radio stack module. A Kernel driver for the parallel port is required since all the software is programmed under Windows XP operation system in this project. [4.8] The Kernel driver was coded in VC++ and attached in Appendix 19.

The parallel communication is sufficient for this application, but it does not improve the original communication between the microprocessor system and the radio stack since the 8-bit ISA bus is running at 5MHz bus clock. On the other hand, there are only 17 pins that can be used for signal transfer in a parallel port. The data signals and the address signals must share the same 8-bit data lines. It still requires the additional hardware on the new custom interface board to control the data flow. Therefore, a better communication technology needs to be found and applied in this project.

4.2.2.2. Ethereal Network and Data Acquisition Technology

Ethernet is a local area network (LAN) technology that transmits information between computers at speeds of 10 and 100 million bits per second (Mbps). Currently the most widely used version of Ethernet technology is the 10-Mbps twisted-pair variety. The 10-Mbps Ethernet media varieties include the original thick coaxial system, as well as thin coaxial, twisted-pair, and fiber optic systems. The most recent Ethernet standard defines the new 100-Mbps Fast Ethernet System that operates over twisted-pair and fiber optic media. [4.9]

The Ethernet network is very effective for communication between computers. Since the radio stack is not as complicated as a computer system and an additional Ethernet network card is required, a decision was made not to use the Ethernet network communication in this project. However, it would be an inspired idea to build the Ethernet network among the flight model computer, the instructor station and the visual system in the future research.

Data acquisition (DAQ) technology has been studied for years. Recently the industrial digital I/O devices could achieve up to 80 Mbytes/s transfer rate and up to 96 I/O lines per device. [4.10] As shown in Chapter 3, a 3MHz clock signal is required in the simulated radio stack system. If the data acquisition card can also generate this clock signal, it would be perfect for this application.

The National Instruments (NI) Counter/Timer PCI-6601 as shown in Figure 4.2.2.2-1 has up to 32 digital I/O lines, 20MHz maximum source frequency and 4 up/down 32-bit counter/timers. [4.11] It meets all the communication requirements for the simulated radio stack system. The parallel 32-bit digital signals can be transferred at maximum 20MHz and the 3MHz frequency signal can be generated by one of the four counter/timers. In this project, a PCI 6601 mounted in the flight model computer was used to interface the radio stack. Therefore, the new custom interface card mount in the radio stack could be designed as simply as possible, and the communication performance would be much better than before.

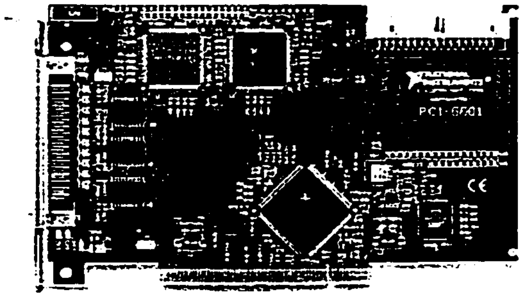


Figure 4.2.2.2-1 NI Counter/Timer PCI-6601 [4.11]

4.2.3. Radio Stack Module in VC++

Using the NI data acquisition card has another advantage that NI DAQ driver software is included in the hardware package. This driver software has an extensive library of functions that can be called from a user application-programming environment such as VC++. The NI DAQ library can be directly used to driver the PCI 6601 hardware in VC++ environment, and the Kernel driver is not required.

One radio stack module was coded in VC++ under Windows XP. The flowchart of Read routine is shown in Figure 4.2.3-1. The flowchart of Write routine is shown in Figure 4.2.3-2. It is noted that the Data input/output only happens when the A0 pin of 8279 is reset to 0. Appendix 20 lists the VC++ functions with the NI DAQ library used in the radio stack module.

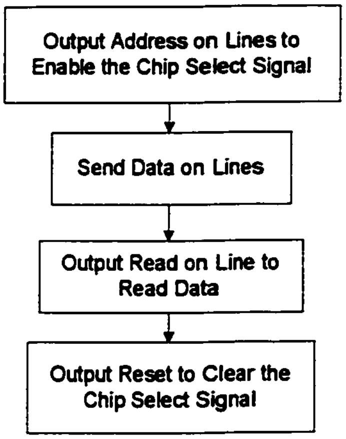

Output Address on Lines to Enable the Chip Select Signal
Send Data on Lines
Output Read on Line to Read Data
Output Reset to Clear the Chip Select Signal

Figure 4.2.3-1 Read Routine Flowchart

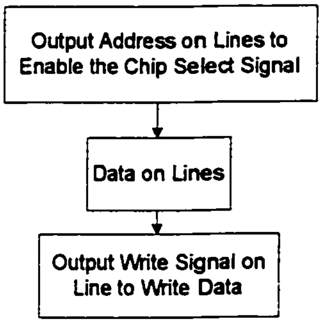

Output Address on Lines to Enable the Chip Select Signal
Data on Lines
Output Write Signal on Line to Write Data

Figure 4.2.3-2 Write Routine Flowchart

One objective of programming this VC++ radio stack module demonstrates that using the NI DAQ library instead of a complicated Kernel driver can directly access the PCI 6601 card under Windows XP. In addition, this VC++ module could be easily integrated with the C flight dynamic model in the future research.

4.2.4. Radio Stack Module in LabVIEW

VC++ is not the best software to develop the simulated radio stack system in this project. The NI LabVIEW application software, which was purchased with the PCI 6601 Counter/Timer hardware, is a more advanced graphical development environment with build-in functionality for this data acquisition and control application. It features interactive graphics, a state-of-the art user interface, and a powerful graphical programming language. [4.12] LabVIEW could be used to achieve one of the objectives of this project that is to offer graphical user interfaces (GUI) for students to experience radio navigation systems without any simulated hardware connected to the PCs. Furthermore, by using LabVIEW with the built-in I/O functions, it could be very easy to graphically represent and control the upgraded radio stack system on a computer screen.

4.2.4.1. Introduction to LabVIEW

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages (such as VC++), where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution. [4.12]

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources, and displays that information or moves it to other files or other computers. In LabVIEW, a user interface is built by using a set of tools and objects. The user interface is known as the front panel shown in Figure 4.2.4.1-1. The user then adds code using graphical representations of functions to control the front panel objects. The block diagram shown in Figure 4.2.4.1-2 contains this code. In some ways, the block diagram resembles a flowchart. Another component of LabVIEW is the icon and connector palette, such as the Tools Palette shown in Figure 4.2.4.1-3. It is used to identify the VI so that user can use the VI in another VI. A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages. [4.12]

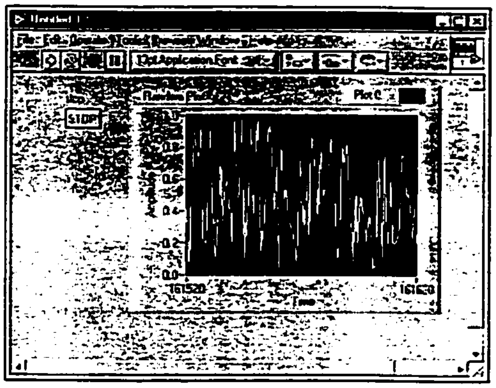


Figure 4.2.4.1-1 LabVIEW Front Panel [4.12]

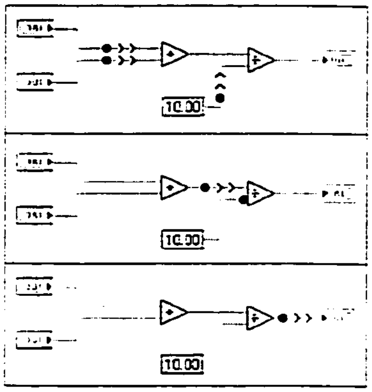


Figure 4.2.4.1-2 LabVIEW Block Diagram [4.12]

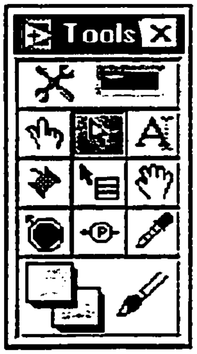
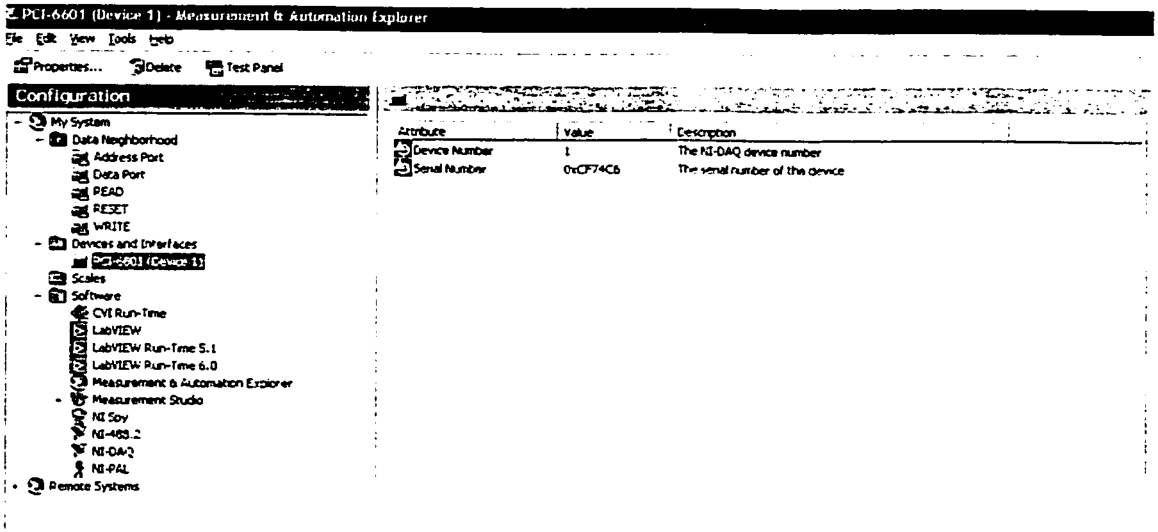


Figure 4.2.4.1-3 LabVIEW Tools Palette [4.12]

4.2.4.2. Measurement and Automation Explorer

Before using LabVIEW to program the radio stack module, the NI PCI-6601 must be configured. With Measurement & Automation Explorer (MAX), the user can configure the National Instruments hardware and software; add new channels, interfaces, and virtual instruments; execute system diagnostics; view devices and instruments connected to the system; schedule updates to the National Instruments software. [4.12]

The configurations of NI PCI-6601 in MAX are shown in Figure 4.2.4.2-1. The PCI-6601 is named as Device 1. In the PCI-6601 board, the DIO_0 - DIO_7 lines in the digital I/O lines are used as the Address Port, the DIO_8 - DIO_15 lines are used as the Data Port. The Read signal is on the DIO_16, the Write signal is on the DIO_17, and the Reset signal is on the DIO_18. The clock signal 3MHz is generated in Counter 0 and found on pin PFI_36.



Attribute	Value	Description
Device Number	1	The NI-DAQ device number
Serial Number	0xCF74C6	The serial number of the device

Figure 4.2.4.2-1 NI PCI-6601 Configuration in MAX

4.2.4.3. ADF in LabVIEW

In this project, the LabVIEW radio stack module was programmed with the help of Pilot Guide documents that were provided by the radio stack manufacturer BENDIX/KING Company. Users could experience the radio navigation systems like a pilot in a real flight, and it could be with or without the simulated hardware connected to the PCs. Therefore, this application could be used as a powerful teaching and research tool in avionics navigation system topic.

The graphical user interface (GUI) for the ADF system is shown in Figure 4.2.4.3-1. Compared with the actual front panel of the original KR 87 ADF, the GUI programmed by LabVIEW gives the amazing effect to users.

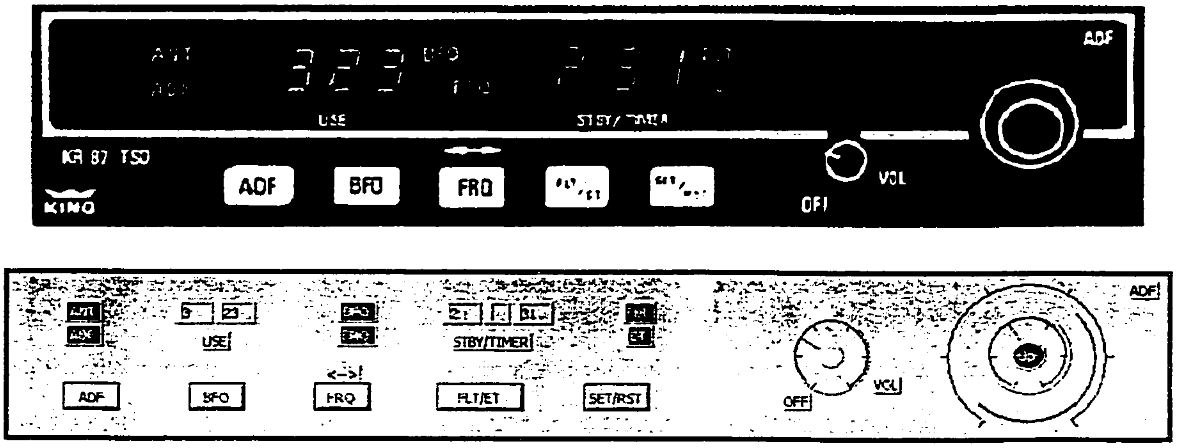


Figure 4.2.4.3-1 ADF GUI

The ADF VI and subVIs in block diagram are attached in Appendix 21. Although not all of subVIs is shown, the block diagram like a flowchart clearly indicates the programming logic. The LabVIEW program is able to track the flow of any data. This kind of advanced software makes the programming and debugging work much easier than the traditional text-based programming languages (such as VC++).

4.2.4.4. RNAV in LabVIEW

The graphical user interface (GUI) for the RNAV system is shown in Figure 4.2.4.4-1. The RNAV VI and subVIs in block diagram are attached in Appendix 21.

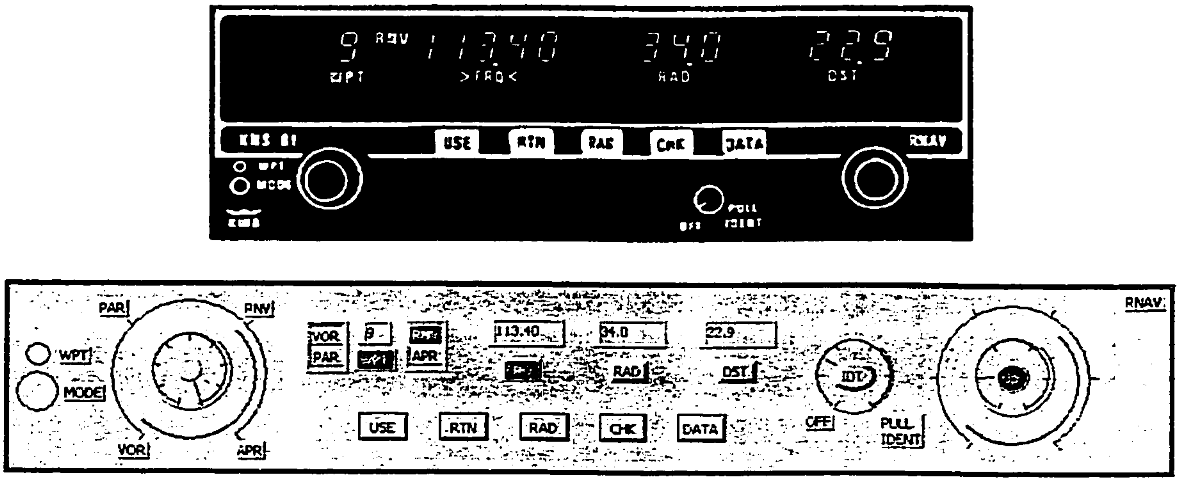


Figure 4.2.4.4-1 RNAV GUI

4.2.4.5. NAV/COMM in LabVIEW

The graphical user interface (GUI) for the NAV/COMM system is shown in Figure 4.2.4.5-1. The NAV/COMM VI and subVIs in block diagram are attached in Appendix 21.

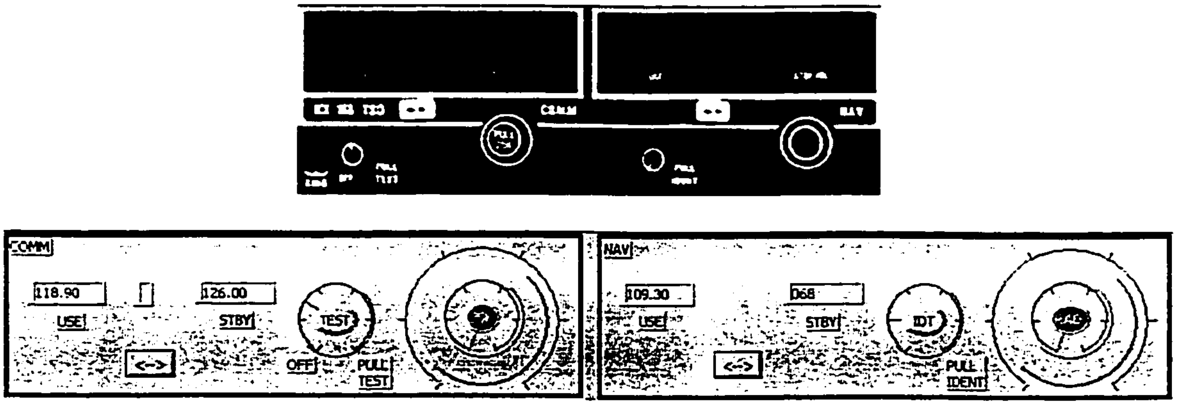


Figure 4.2.4.5-1 NAV/COMM GUI

4.2.4.6. COMM in LabVIEW

The graphical user interface (GUI) for the COMM system is shown in Figure 4.2.1.6-1. The COMM VI and subVIs in block diagram are attached in Appendix 21.

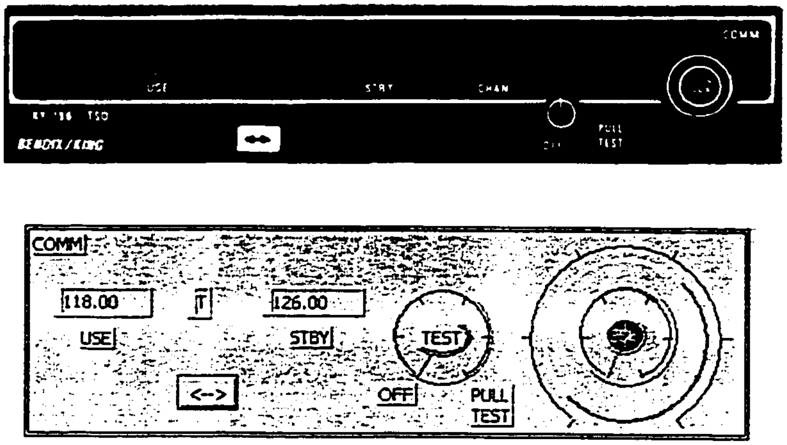


Figure 4.2.4.6-1 COMM GUI

4.2.4.7. ATC XPNDR in LabVIEW

The graphical user interface (GUI) for the ATC XPNDR system is shown in Figure 4.2.4.7-1. The ATC XPNDR VI and subVIs in block diagram are attached in Appendix 21.

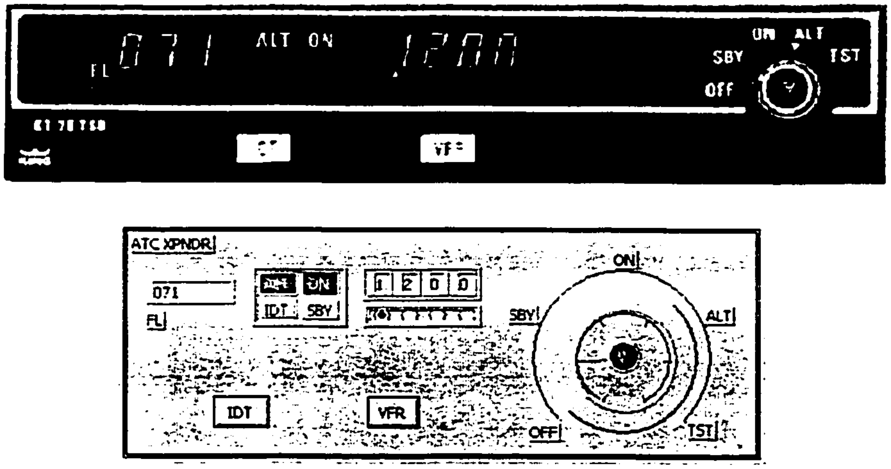


Figure 4.2.4.7-1 ATC XPNDR GUI

4.2.4.8. DME in LabVIEW

The graphical user interface (GUI) for the DME system is shown in Figure 4.2.4.8-1. The DME VI and subVIs in block diagram are attached in Appendix 21.

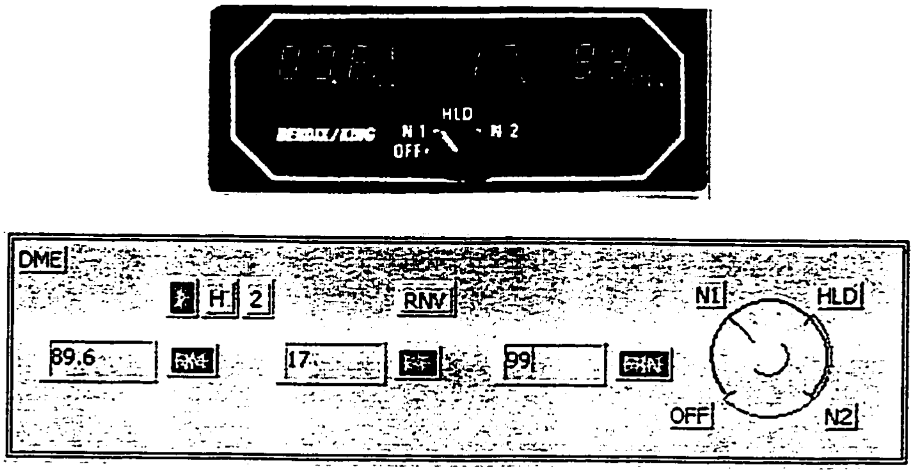


Figure 4.2.4.8-1 DME GUI

4.2.4.9. MBR in LabVIEW

The graphical user interface (GUI) for the MBR system is shown in Figure 4.2.4.9-1. The MBR VI switches controls in block diagram are attached in Appendix 21.

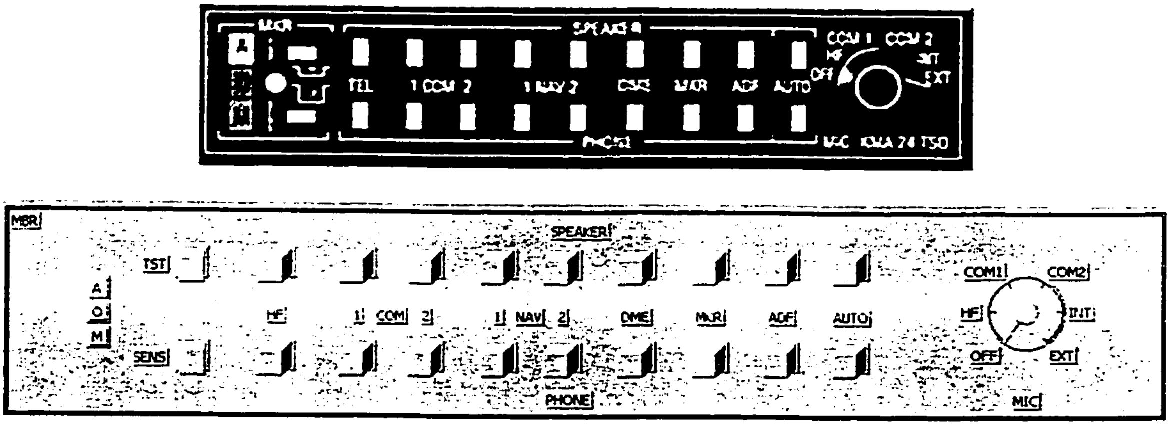


Figure 4.2.4.9-1 MBR GUI

4.3. Summary

This chapter presented the upgraded radio stack system and the computer code implementation of the radio stack module. The upgraded radio stack system represents the advanced interface technology and provides better performance. The radio stack module, which is used to control and drive the upgraded radio stack system, is able to acknowledge the status of every switch on the front panel of each station and generate the corresponding data for display.

Since the original radio stack module was damaged, an upgraded radio stack module stored in the flight model computer was programmed to replace it. First of all, several programmable functional IC chips were programmed. Then, the high-speed communication between the simulated radio stack and the flight model computer was selected. The considerations included RS232, USB, Parallel Port, Ethernet, and finally the advanced data acquisition card NI PCI-6601 was selected.

The VC++ radio stack module demonstrates that using the NI DAQ library instead of a complicated Kernel driver can directly access the PCI 6601 card under Windows XP. In addition, this VC++ module could be easily integrated with the existing C flight dynamic model in the future research.

The LabVIEW radio stack module was programmed to achieve one of the objectives of this project that is to offer graphical user interfaces (GUI) for students to experience radio navigation systems without any simulated hardware connected to the PCs. Furthermore, by using LabVIEW with the built-in I/O functions, it became easy to graphically represent and control the upgraded radio stack system on a computer screen.

CHAPTER 5
COMPUTER CODE IMPLEMENTATION OF NAVIGATION MODEL

The so-called navigation model is a software package used to implement the radio stack system's navigation functions in a flight simulation. In the existing system, a simple navigation model done by Mr. P. Lawn was integrated with his flight dynamic model. In this project, a more sophisticated and versatile navigation model was programmed for students to experience the radio navigation systems without the complicated flight dynamic model since this project focused on the development of avionic system. The upgraded navigation model was coded in VC++ so that it could be integrated with the flight dynamic model, and be modified by students in the future research.

The main components of this navigation model include the Timer, ADF, DME, and VOR/ILS modules.

5.1. Timer Module

In order to run the program in real time, it is essential to keep track of the elapsed time and the iteration period Δt. The iteration period is used to calculate the velocities, attitude and position by numerically integrating accelerations. The PC has an internal system timer that returns the elapsed time since 00:00:00 UTC (Coordinated Universal Time), January 1, 1970. [5.1] The resolution of this returned time is within 55ms because the system timer uses IRQ0 (Interrupt Request 0) to interrupt the processor 18.2 times per second and keep the time-of-day clock updated. [5.2] In the Concordia flight simulator, to be a dynamic representation of the behavior of the aircraft in a manner that allows the human operator to interact with the simulation in real time, [5.3] the flight simulation is required to cycle at a minimum of 50Hz (a iteration period of 20ms). [5.4] The Timer module uses the returned elapse time from the PC internal system timer to measure the iteration period. Since the variation of this returned time is within 55ms, a single measurement may result in incorrect data. This error is reduced by taking a number of cycles and averaging the elapsed time over two seconds according to the current period. The program then compares the measured elapsed time to the iteration period and the number of cycles since the previous measurement. If the computed iteration period is greater than 20ms, that means the simulation program are running too slow and does not meet the real time requirement, an unacceptable error will occur and the simulation program will stop. In other words, the Timer module checks the iteration period every two seconds, and ends the simulation if there is an unacceptable error. In this way, the simulation program will always run as fast, and with the highest fidelity, as the PC can support. Since the elapsed time is always measured since the simulation began, there are no cumulative errors in calculating the iteration period. The logic flow shown in Figure 5.1-1 illustrates the sequential equation execution within the Timer module and the routine used to monitor the iteration period is given in Appendix 22.

In the cases where it is not necessary to run the simulation in real time, for example when it is being controlled by command stored in a file, the iteration period could be set to any value. It would increase the fidelity of the simulation, and allow slower computers to get the result as faster computers.

In this project, there is no requirement to execute the navigation model at a rate greater than 7.5Hz (a iteration period of 133ms). [5.5] The Timer module described here provides the necessary information for future students to integrate the navigation model with the flight dynamic model.

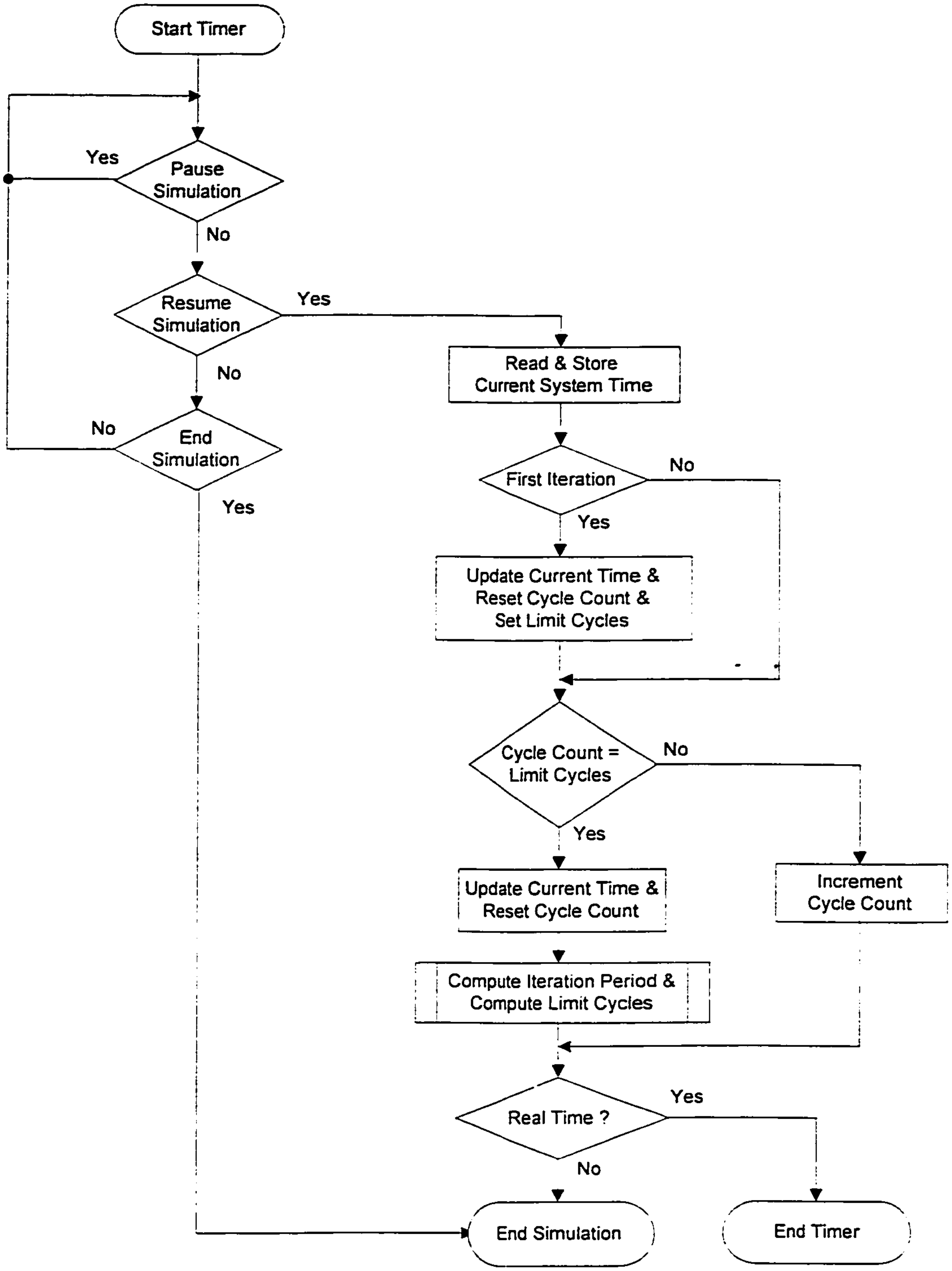

Start Timer
Yes
Pause Simulation
No
Resume Simulation
Yes
No
Read & Store Current System Time
No
End Simulation
Yes
First Iteration
No
Yes
Update Current Time & Reset Cycle Count & Set Limit Cycles
Cycle Count = Limit Cycles
No
Yes
Update Current Time & Reset Cycle Count
Increment Cycle Count
Compute Iteration Period & Compute Limit Cycles
Real Time ?
Yes
No
End Simulation
End Timer

Figure 5.1-1 Timer Module Flow Chart

5.2. ADF Module

The ADF module simulates the operation of the ADF receiver and the associated controls, senses the selections from the front panel and determines the system operating mode and frequency. The task also processes the various inputs of the following:

- Aircraft position
- Selected station frequency
- Altitude, longitude, elevation
- Type and power range to compute aircraft bearing to the selected station
- Audio identification information
- Associated data validities

The logic flow shown in Figure 5.2-1 illustrates the sequential equation execution within the ADF module. The VC++ function used to calculate the bearing to the selected station is shown in Appendix 23.

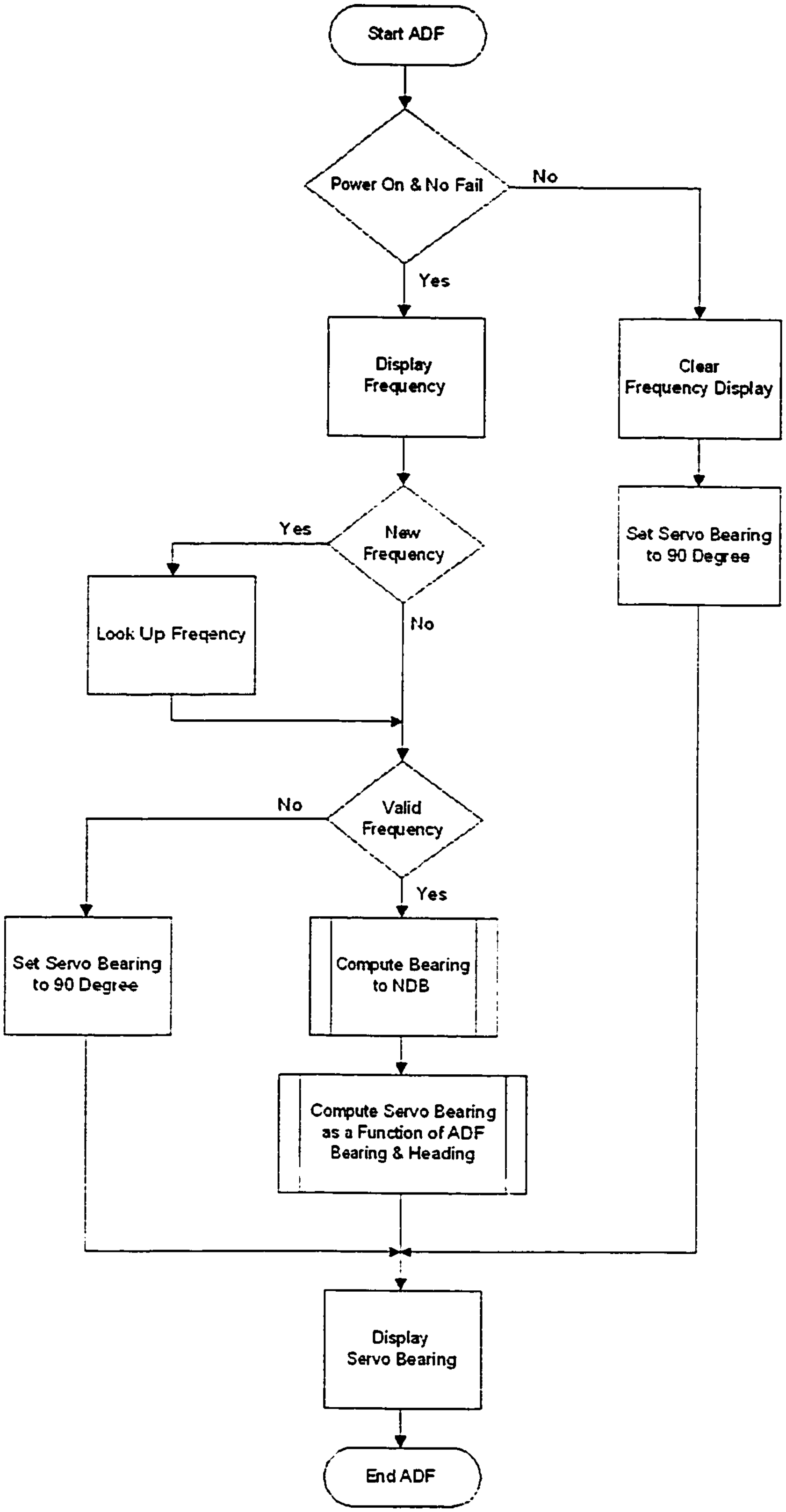


Figure 5.2-1 ADF Module Flowchart

Calculating the relative bearing from the aircraft to the transmitter simulates the ADF. The Earth centered fixed coordinate system is used for these calculations.

Consider an aircraft locates at A_1 and an NDB beacon locates at N_1 on the Earth as shown in Figure 5.2-2. The calculation to determine the relative bearing to the aircraft is achieved through two transformations. [5.4]

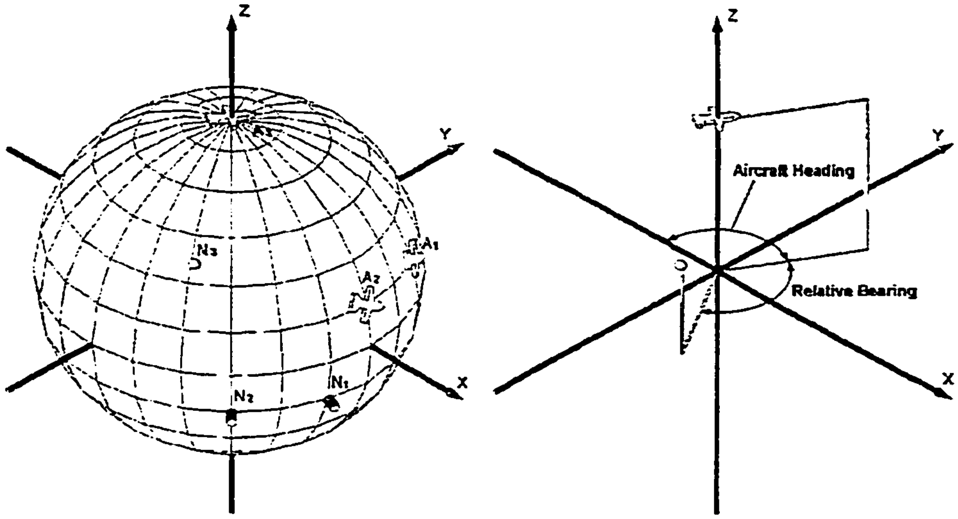


Figure 5.2-2 Relative Bearing from Aircraft to NDB Beacon [5.4]

The aircraft and NDB are located respectively at coordinates $A_1(Lat_{A_1}, Lon_{A_1})$ and $N_1(Lat_{N_1}, Lon_{N_1})$. Firstly, the aircraft and NDB are equally rotated about the Z-axis (Earth's axis) by the negative of the aircraft's longitude so that the aircraft is on zero longitude to position A_2 and N_2 respectively. [5.4]

$$A_2(Lat_{A_1}, 0) \tag{5.1}$$

$$N_2(Lat_{N_1}, Lon_{N_1} - Lon_{A_1}) \tag{5.2}$$

$$\begin{aligned} N_2(X_{N_2}, Y_{N_2}, Z_{N_2}) = (&R_{earth}\cos(Lat_{N_2})\cos(Lon_{N_2}), \\ &R_{earth}\cos(Lat_{N_2})\sin(Lon_{N_2}), \\ &R_{earth}\sin(Lat_{N_2})) \end{aligned} \tag{5.3}$$

Next, the aircraft and NDB are equally rotated about the Y-axis so that the aircraft is at the North Pole at position A_3 and N_3 respectively. [5.4]

$$\begin{bmatrix} X_{N_3} \\ Y_{N_3} \\ Z_{N_3} \end{bmatrix} = \begin{bmatrix} \cos(90^o - Lat_{A_2}) & 0 & -\sin(90^o - Lat_{A_2}) \\ 0 & 1 & 0 \\ \sin(90^o - Lat_{A_2}) & 0 & \cos(90^o - Lat_{A_2}) \end{bmatrix} \begin{bmatrix} X_{N_2} \\ Y_{N_2} \\ Z_{N_2} \end{bmatrix} \qquad (5.4)$$

Finally the bearing to NDB is calculated from Equation 5.5 and Equation 5.6 gives the relative bearing to the NDB. [5.4]

$$Bearing_{NDB} = \tan^{-1}(\frac{-X_{N_3}}{Y_{N_3}}) \qquad (5.5)$$

$$\text{Re}\,lative \quad Bearing = Bearing_{NDB} - True \quad Heading_{AC} \qquad (5.6)$$

Equation 5.7 is used to determine the output pointer bearing. The bearing servo updates the ADF pointer by driving it to the target servo bearing at the rate of 25 degrees per second. The pointer bearing is limited to plus or minus 180 degrees and sent to the ADF indicator. [5.5]

ADF Magnetic Bearing = ADF Bearing – Compass Heading (5.7)

5.3. DME Module

The DME module simulates the operation of the DME transponder and the associated controls, senses the selections from the front panel, determines the system operating mode, and then the various inputs of the following are processed. These inputs are used to compute the aircraft ground speed, time-to-go, slant range, and audio identification from the selected station and associated data validities.

- Aircraft position
- Selected station frequency
- Latitude
- Longitude
- Elevation
- Type and power range to compute aircraft range to the selected station

The logic flow shown in Figure 5.3-1 illustrates the sequential equation execution within the DME module. Appendix 23 lists the VC++ function used to calculate the distance to the selected station.

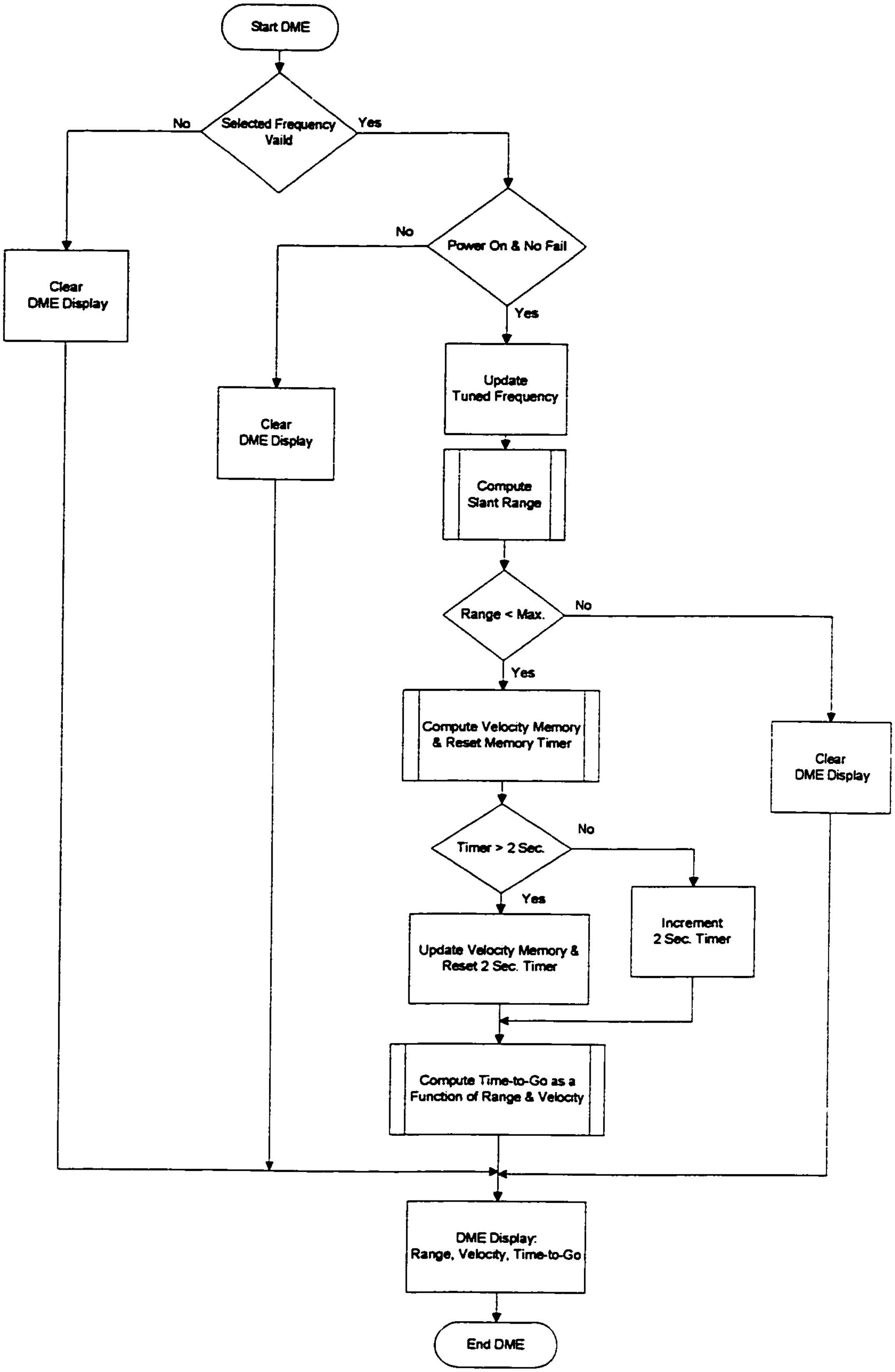


Figure 5.3-1 DME Module Flowchart

Calculating the straight-line distance from the aircraft to the DME station simulates the DME system. Both the coordinates of the aircraft and the station are converted from latitude, longitude and altitude to Cartesian coordinates using Equation 5.8 and 5.9, and Equation 5.10 is used to calculate the distance between the aircraft and the DME station. [5.4]

$$\begin{aligned}(X_{AC}, Y_{AC}, Z_{AC}) = (&R_{earth}\cos(Lat_{AC})\cos(Lon_{AC}),\\ &R_{earth}\cos(Lat_{AC})\sin(Lon_{AC}),\\ &R_{earth}\sin(Lat_{AC}))\end{aligned} \tag{5.8}$$

$$\begin{aligned}(X_{ST}, Y_{ST}, Z_{ST}) = (&R_{earth}\cos(Lat_{ST})\cos(Lon_{ST}),\\ &R_{earth}\cos(Lat_{ST})\sin(Lon_{ST}),\\ &R_{earth}\sin(Lat_{ST}))\end{aligned} \tag{5.9}$$

$$DIST = \sqrt{(X_{AC} - X_{ST})^2 + (Y_{AC} - Y_{ST})^2 + (Z_{AC} - Z_{ST})^2} \tag{5.10}$$

If the computed slant range is less than the maximum receiver operating range, then the memory timer is reset for subsequent operation and a velocity (range rate) is computed and used for updating when the DME goes into memory operation. Equation 5.11 is used to compute velocity memory and reset memory timer. [5.5]

$$VELOCITY = \frac{(DIST - OLDIST) \times INCR}{INTERVAL} \tag{5.11}$$

Where:

VELOCITY: Aircraft velocity (NM/iteration)

DIST: Slant range (NM)

OLDIST: Old slant range (NM)

INCR: Iteration period

INTERVAL: Time between samples (2 seconds)

The time-to-go is a function of the computed range and the velocity, as shown in Equation 5.12. [5.5]

$$Time-to-Go = \frac{DIST}{VELOCITY} \tag{5.12}$$

5.4. VOR/ILS Module

The VOR/ILS module simulates the operation of the VOR/ILS receiver and the associated controls. It senses the selections from the front panel and determines the system operating mode and frequency. The task also processes the various inputs of the following. These inputs are used to compute aircraft range/bearing to the selected station. deviation from a selected radial, azimuth and vertical deviation information, audio identification information, and associated data validities.

- Aircraft position
- Selected station frequency
- Latitude
- Longitude
- Elevation
- Type and power range to compute aircraft range/bearing to the selected station

The logic flow shown in Figure 5.4-1 illustrates the sequential equation execution within the VOR/ILS module. Appendix 23 lists the VC++ function used to calculate the bearing to the selected station.

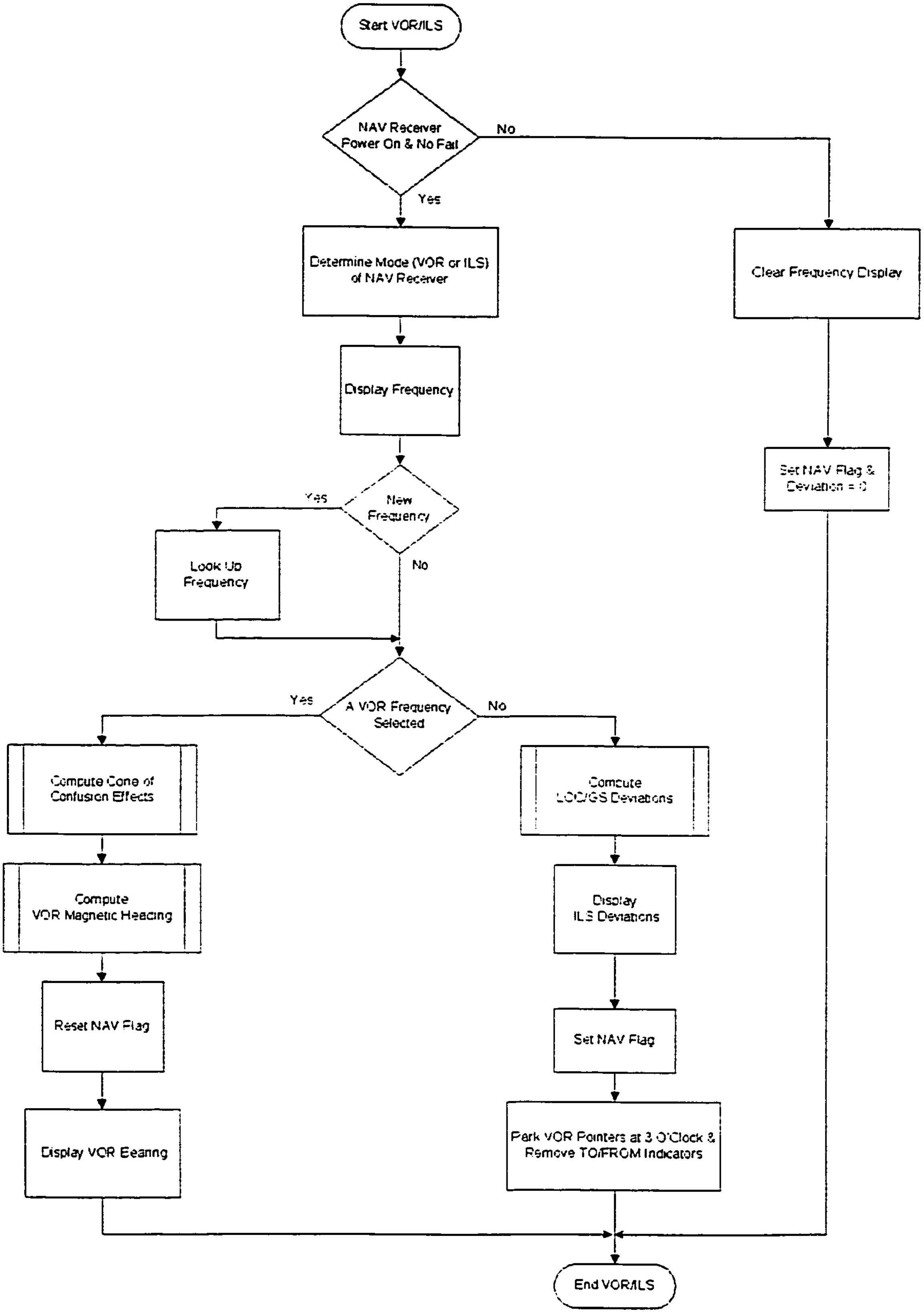

Start VOR/ILS
NAV Receiver Power On & No Fail
No
Yes
Determine Mode (VOR or ILS) of NAV Receiver
Clear Frequency Display
Display Frequency
Set NAV Flag & Deviation = 0
New Frequency
Yes
No
Look Up Frequency
A VOR Frequency Selected
Yes
No
Compute Cone of Confusion Effects
Compute LOC/GS Deviations
Compute VOR Magnetic Heading
Display ILS Deviations
Reset NAV Flag
Set NAV Flag
Display VOR Bearing
Park VOR Pointers at 3 O'Clock & Remove TO/FROM Indicators
End VOR/ILS

Figure 5.4-1 VOR/ILS Module Flowchart

5.4.1. VOR Simulation

The VOR is simulated in a manner similar to that of the ADF. The only exception is that the bearing from the station to the aircraft instead of the bearing from the aircraft to the station is calculated.

Since the true radial from a VOR station are defined relative to the Earth's magnetic field, the true radial used in the calculations is determined by subtracting the magnetic variation at the VOR station from the selected magnetic bearing.

Firstly, the coordinates of the aircraft and transmitter are rotated by an equal amount about the Earth's axis to position A_2 and V_2 so that the station's longitude is at zero as shown in Figure 5.4.1-1. The coordinates are then converted to Cartesian coordinates using Equation 5.15, with the Z-axis aligned with the Earth's northerly axis and the station on the Y-Z plane. [5.4]

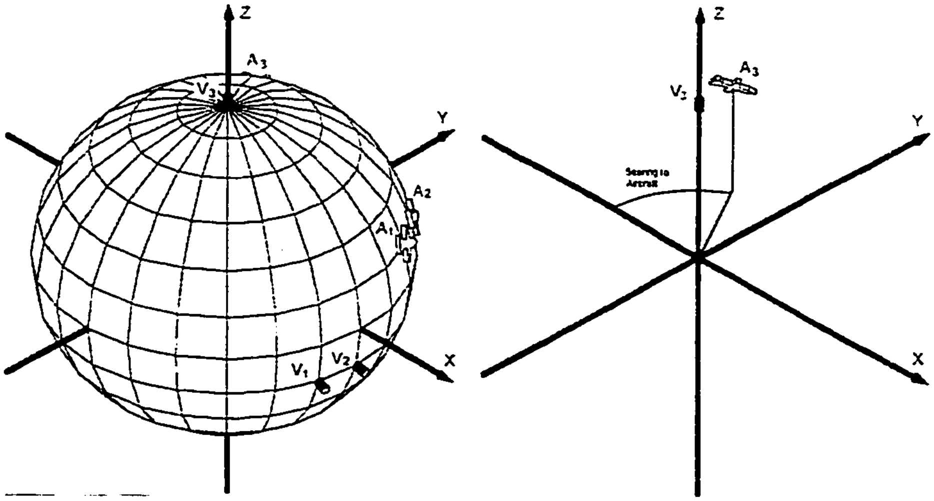


Figure 5.4.1-1 Bearing from VOR Station to Aircraft [5.4]

$$V_2(Lat_{V_1},0) \tag{5.13}$$

$$A_2(Lat_{A_1},Lon_{A_1}-Lon_{V_1}) \tag{5.14}$$

$$\begin{aligned} A_2(X_{A_2}, Y_{A_2}, Z_{A_2}) = (&R_{earth}\cos(Lat_{A_2})\cos(Lon_{A_2}), \\ &R_{earth}\cos(Lat_{A_2})\sin(Lon_{A_2}), \\ &R_{earth}\sin(Lat_{A_2})) \end{aligned} \tag{5.15}$$

Next, using the Equation 5.16, the aircraft and the station are rotated by an equal amount about the Y-axis to position A_3 and V_3 so that the VOR station is at the North Pole. The bearing from the station to the aircraft is calculated from the aircraft's X and Y coordinates. The selected true radial is subtracted from this bearing to give the deviation shown in Equation 5.17. [5.4]

$$\begin{bmatrix} X_{A_3} \\ Y_{A_3} \\ Z_{A_3} \end{bmatrix} = \begin{bmatrix} \cos(90'' - Lat_{V_2}) & 0 & -\sin(90'' - Lat_{V_2}) \\ 0 & 1 & 0 \\ \sin(90'' - Lat_{V_2}) & 0 & \cos(90^{o} - Lat_{V_2}) \end{bmatrix} \begin{bmatrix} X_{A_2} \\ Y_{A_2} \\ Z_{A_2} \end{bmatrix} \tag{5.16}$$

$$Deviation = \tan^{-1}\left(\frac{-X_{A_3}}{Y_{A_3}}\right) - Selected \quad Radial \tag{5.17}$$

In VOR mode, the aircraft ground range and ground altitude are used to determine whether or not the aircraft has penetrated the cone of confusion shown in Figure 5.4.1-2. If so, the signal strength and amount of random bearing noise are computed as a function of the aircraft range from the center of the cone. The bearing validity is reset for complete signal loss when inside a 40-degree cone. The amount of bearing jitter is a function of the aircraft distance from the center of the cone. A random number is used to generate random fluctuations from the target bearing. Equation 5.18 is used to calculate the VOR bearing jitter amount. [5.5]

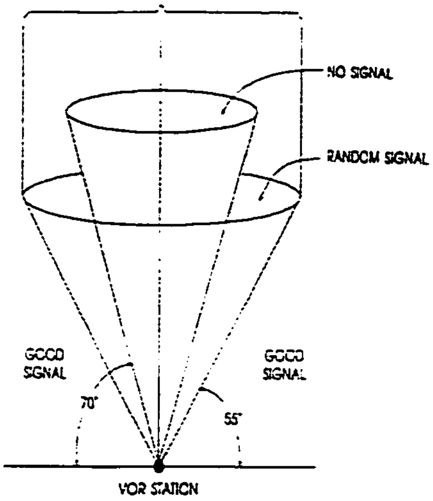


Figure 5.4.1-2 Cone of Confusion [5.5]

$$JITTER = \left(\frac{VH}{NM\_FT \times RAN + 100} - 1.0\right) \times RANDOM \times 5 \qquad (5.18)$$

Where: VH: Aircraft height above ground (ft)

RAN: Aircraft range to tuned station (nm)

RANDOM: Random Number

NM_FT: nm to feet factor (=6076.1)

The magnetic bearing to the station is computed from the true bearing, the selected station magnetic declination and the bearing jitter. Equation 5.19 is used to calculate the magnetic bearing for the indicator. [5.5]

$$MAGBRG=BRG + JITTER - VORDLN \qquad (5.19)$$

Where: BRG: VOR true bearing (degree)

JITTER: VOR bearing jitter amount (degree)

VORDLN: VOR station magnetic declination (degree)

5.4.2. ILS Simulation

The localizer portion of the ILS system is simulated in the same manner as the course deviation from a VOR beacon except that in the case of the ILS system, the pilot does not need to select the course or radial like in the case of a VOR, but rather the runway heading is used. Equations 5.12 to 5.16 are used to calculate the deviation of an aircraft from the runway heading. The deviation is displayed on a CDI by the position of the vertical bar where a full-scale deflection would typically represent a deviation of plus or minus 2 degrees from the proper approach.

The deviation from the glideslope is calculated in a similar manner to the DME calculations. Since the ILS system is used over the relatively small area around an airport, it is assumed that the curvature of the Earth is negligible. It greatly simplifies the calculations. Firstly, the straight-line distance from the aircraft to the glideslope transmitter is calculated in exactly the same manner used to calculate the distance to a DME station. The approach angle is calculated by taking the arcsine of the quotient of altitude above the runway divided by the straight line distance, as given in Equation 5.20. [5.4]

$$Approach\quad Angle = \arcsin\left(\frac{Altitude\quad Above\quad Runway}{Straight\quad Line\quad Dis\tan ce}\right) \tag{5.20}$$

The approach angle is compared to the glideslope angle, typically 2.5 to 3 degrees, and the deviation is displayed on the CDI.

5.5. Summary

This chapter presented the computer code implementation of the navigation model. The so-called navigation model is a software package used to implement the radio stack system's navigation functions in a flight simulation.

In the existing system, a simple navigation model done by Mr. P. Lawn was integrated with his flight dynamic model. In this project, a more sophisticated and versatile navigation model without the complicated flight dynamic model was programmed for students to experience the radio navigation systems typically found on a general aviation aircraft. The upgraded navigation model was coded in VC++ so that it could be integrated with the existing flight dynamic model, and be modified by students in the future research.

This navigation model consists of the Timer, ADF, DME, and VOR/ILS modules. The Timer module keeps track of the elapsed time and the period of iteration Δt so that the program could run in real time. The ADF, DME, and VOR/ILS modules respectively simulate the ADF, DME, VOR and ILS systems in the similar manner to the existing navigation model, but more functions are implemented in the upgraded navigation model.

The navigation model is able to provide the proper range and bearing information for different radio navigation systems and navigational instruments. It would make more sense when it is integrated with the flight dynamic model in a completed flight simulation.

CHAPTER 6
ASSOCIATED NAVIGATION INDICATORS OPERATION AND SIMULATION

In the Concordia flight simulator, two associated navigation indicators are cooperated with the simulated radio stack system: KI 525A Pictorial Navigation Indicator (PNI) and KI 229 Radio Magnetic Indicator (RMI). They are quite difficult to simulate and model because of their complicated mechanical and electrical characteristics. In this chapter, the operation and simulation of these two indicators are described.

6.1. KI 525A Pictorial Navigation Indicator

Originally, the KI 525A pictorial navigation indicator is one of the components in the King KCS 55A Compass System that provides the pilot with a simple, comprehensive visual display of the aircraft's heading and position in relation to a desired course. [6.1] The display of the KI 525A indicator is shown in Figure 6.1-1.

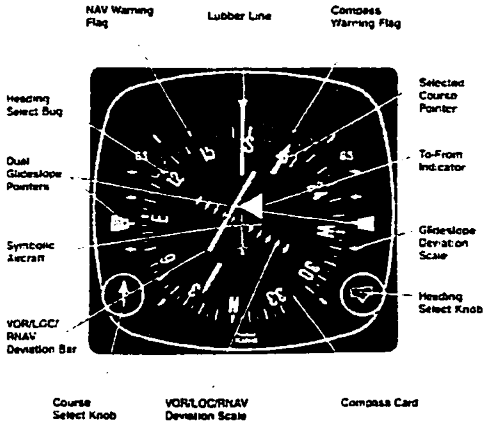


Figure 6.1-1 KI 525A Display [6.1]

6.1.1. KI 525A PNI Operation

The panel-mounted KI 525A PNI combines the display functions of the standard Directional Gyro with VOR/LOC course deviation indication and Glideslope deviation and flag into one compact display. Consequently, pilot workload is considerably reduced and visual coordination between several separate indicators is eliminated. The KI 525A simplifies course orientation, intercept and tracking and will generally result in better piloting technique and skill. [6.1]

The functional sections of the original KI 525A indicator include digitally driven heading display card, course datum and heading select optically derived autopilot outputs, a servo driven glideslope pointer using an optical position sensor, a glideslope retract circuit to detect an invalid GS signal, a NAV flag circuit that monitors NAV receiver power and video signal level, a HDG flag that monitors system power, gyro spin motor operation and slaving activity; plus the normal course deviation bar, TO-FROM meter, slaving CT, heading transmitter and course resolver. [6.2]

The KI 525A indicator provides a pictorial display of the horizontal navigation situation. It also provides manual controls for course and heading datum selections. Outputs from the KI 525A can be used for automatic pilot or flight director, the VOR receiver, and additional compass loads. [6.2] The KCS 525A Compass System block diagram is shown in Figure 6.1.1-1.

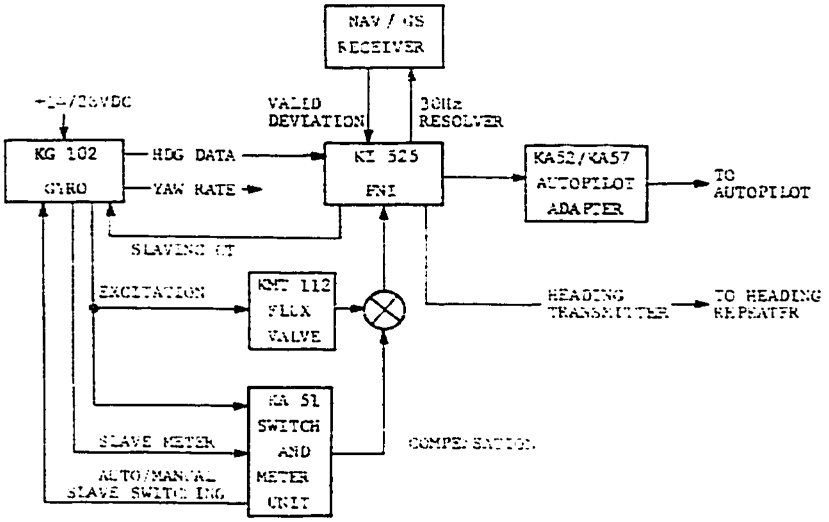


Figure 6.1.1-1 KI 525A PNI Block Diagram [6.2]

6.1.2. KI 525A PNI Simulation

In the KI 525A indicator, three-phase synchros and resolvers are used. The existing interface boards in the Concordia flight simulator are not able to interface them. In this project, a custom interface board was used to try to read and drive them, but there were still problems to completely simulate the PNI.

The KI 525A can be combined with the KI 229 RMI (Radio Magnetic Indicator) and the KX165 NAV/COMM transceiver to form the NAV/COMM systems as shown in Figure 6.1.2-1. The radio magnetic indicator KI 229 will be described later, and the KX165 NAV/COMM transceiver was as described in Chapter 3. The inputs of the KI 525A are the VOR/LOC and the GS bearing information from the KX165 NAV/COMM transceiver. The output of the KI 525A is the heading information for the KI 229 indicator.

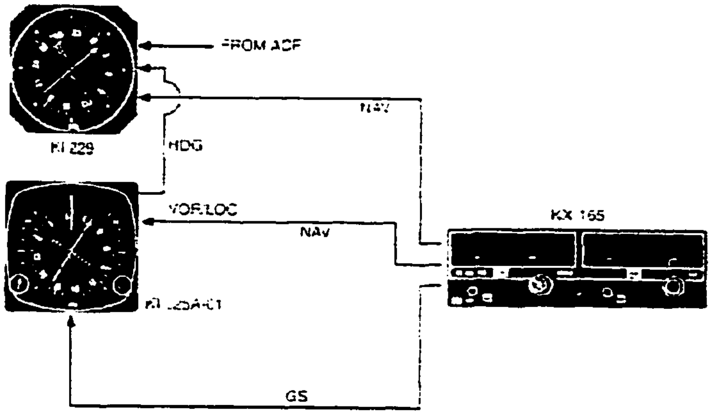


Figure 6.1.2-1 NAV/COMM System [6.1]

The KI 525A can be also combined with the KI 229 RMI, the KY 196 COMM transceiver and the KNS 81 RNAV system to form the NAV/RNAV/COMM systems as shown in Figure 6.1.2-2. The KY 196 COMM transceiver and the KNS 81 RNAV system were as described in Chapter 3. The inputs of the KI 525A are the RNAV/VOR/LOC and

GS bearing information from the KNS 81 RNAV system. Its output is the heading information for the KI 229 indicator.

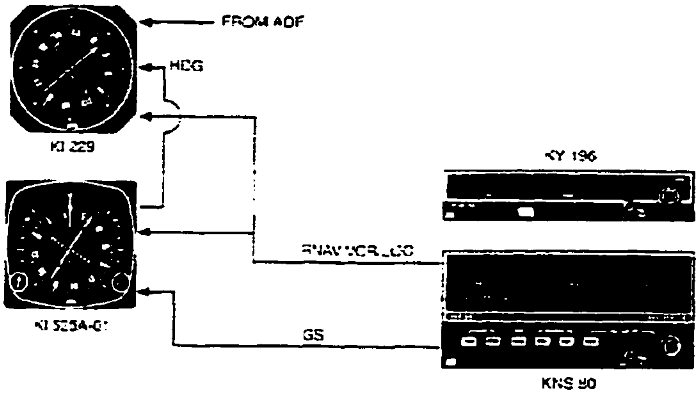


Figure 6.1.2-2 NAV/RNAV/COMM System [6.1]

In a flight simulation, the navigation model provides all the required navigation bearing information and heading information to the PNI. First of all, the navigation model sends the initial signals to the KI 525A indicator and the navigation systems in the radio stack. Then, the indicator could send the manual controls for course and heading datum selections to the navigation model, and the radio stack could send the changed RNAV/VOR/LOC/GS information to the navigation model as well. Finally, the flight model processes these manual changes and sends the proper data to the corresponding systems. Figure 6.1.2-3 shows this simulation logic.

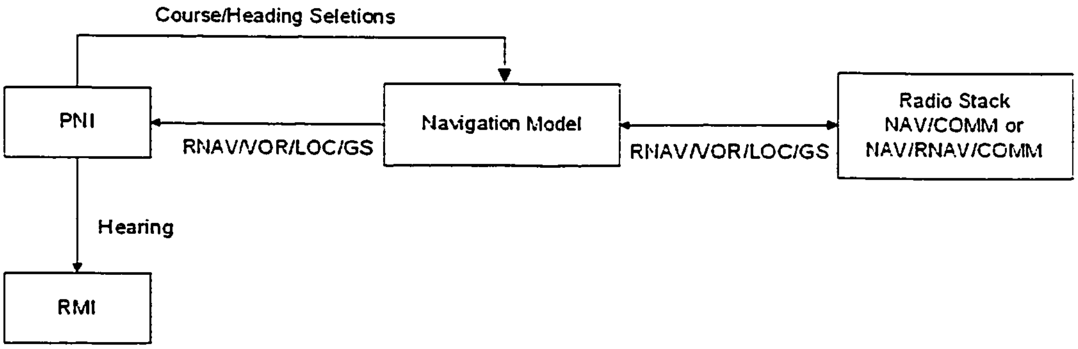


Figure 6.1.2-3 PNI Simulation Block Diagram

6.2. KI 229 Radio Magnetic Indicator

The KI 229 Radio Magnetic Indicator (RMI) shown in Figure 6.2-1 provides bearing information to either ADF or VOR stations by means of two pointers, each of which is read against the compass card. The "double" yellow pointer is dedicated solely to an ADF. The single green pointer is dedicated to VOR. A servo drive compass card with a fixed orange lubber line on the top of the display indicates aircraft heading information. [6.1] The KI 229 RMI can be used as a bearing repeater of the KI 525A indicator within the KCS 55A Compass system. As shown in Figure 6.1.2-1 and 6.1.2-2, the KI 229 indicator receives the ADF bearing information from the ADF system, the VOR bearing information from the NAV/COMM transceiver or the RNAV system, and the heading information from the KI 525A indicator.

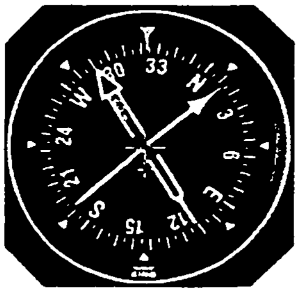


Figure 6.2-1 KI 229 RMI Display [6.1]

6.2.1. KI 229 RMI Operation

The original KI 229 utilizes a ringing choke power supply enabling it to operate from +13.75VDC or +27.5VDC aircraft power. The ringing choke circuit is a continuous energy flow power supply, limited like the linear to single-voltage input, but using higher frequency to pass the energy to the load. An incoming digital word presents a new available VOR bearing for display and the AC 10VRMS 400Hz SIN/COS signals are used to determine input VOR bearing. The 3VDC SIN/COS signals are used to determine input ADF bearing. Applying the digital signals to synchro DC repeaters drives the two needles. When a VOR input is determined invalid by the source or in the case of a composite signal, the associated needle is parked in a horizontal position pointing to the right. [6.3]

The KI 229 can be used as a bearing repeater within a compass system or other slaved directional gyro system. Aircraft heading is read under the lubber line of the KI 229. When the ADF receiver is tuned to a station, the yellow ADF pointer indicates the magnetic heading to the station. Thus, if the pilot desires to fly toward the station, he or she merely turns the aircraft to the magnetic heading indicated by the ADF pointer. When the VOR receiver is tuned to a VOR station, the green VOR pointer indicates the magnetic heading to the station. If the KI 229 is used in an Area Navigation system, the VOR pointer indicates the magnetic heading to the waypoint. Should a localizer frequency be selected, or the VOR receiver indicates a flagged condition, the VOR pointer is parked 90" to the right of the lubber line. [6.4] The KI 229 block diagram is shown in Figure 6.2.1-1.

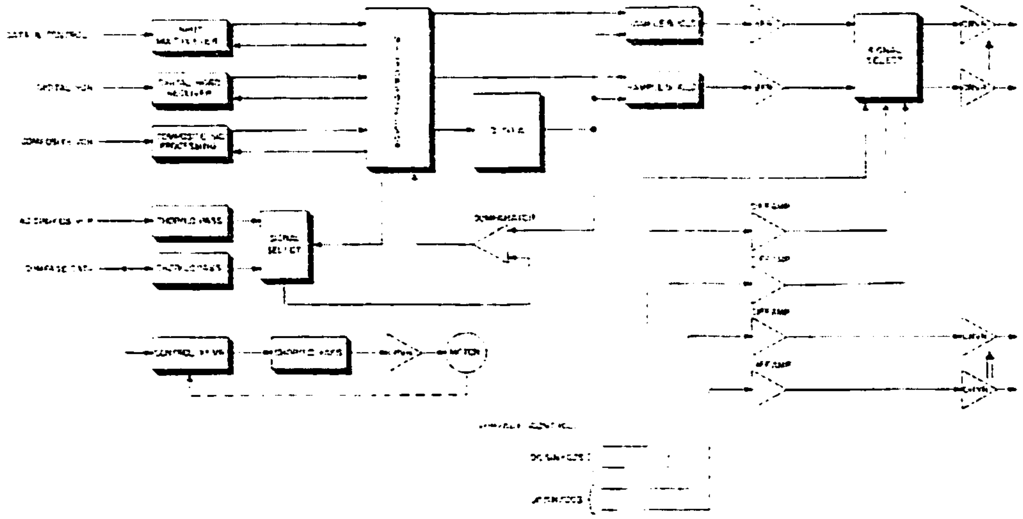


Figure 6.2.1-1 KI 229 Block Diagram [6.3]

6.2.2. KI 229 RMI Simulation

In order to simulate the KI 229, the AC 10VRMS 400Hz Sine/Cosine signals and digital words are required for VOR bearing display, the 3VDC Sine/Cosine signals are required for ADF bearing display, and the simulated KI 229 should be also able to receive the heading information provided by the KI 525A indicator.

In the Concordia flight simulator, the main power supply could provide the power for the KI 229 RMI. However, the existing interface boards are still not able to interface all of the required signals for the RMI. The interface board must be able to generate or interface the required AC SIN/COS and DC SIN/COS signals. In this project, a custom interface board was used to try to read and drive the RMI, but there were still problems to completely simulate the RMI.

As shown in Figure 6.2.2-1, the PNI provides the required heading information for the heading display on the RMI. Both the ADF/RNAV/NAV/COMM stations in the radio stack and the PNI provide the necessary information for the navigation model to generate the required signals for the ADF and VOR bearing displays on the RMI.

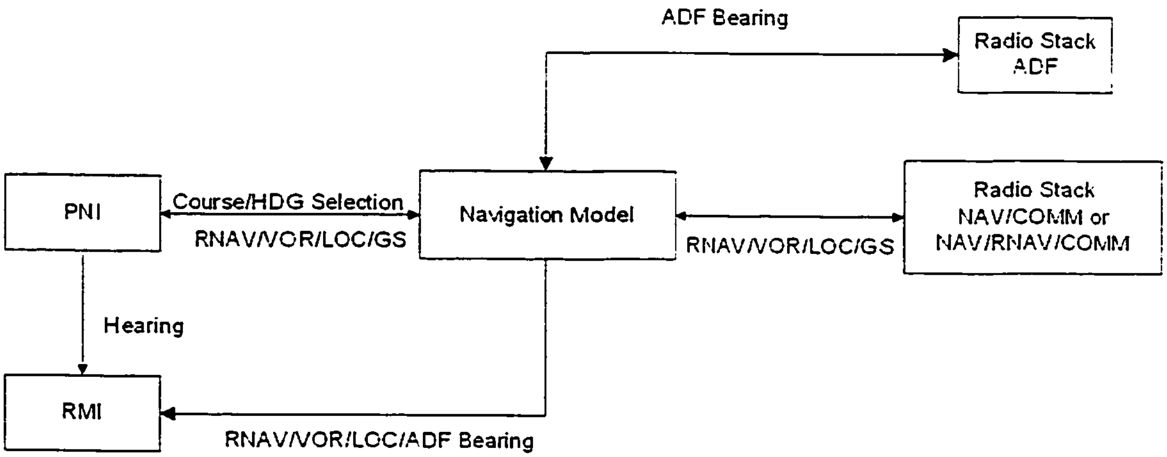


Figure 6.2.2-1 RMI Simulation Block Diagram

6.3. Summary

In this chapter, the operation and simulation of Radio Magnetic Indicator (RMI) and Pictorial Navigation Indicator (PNI) were introduced. These two associated navigation indicators could be used with the radio stack to form the advanced radio navigation systems, such as NAV/COMM systems and NAV/RNAV/COMM systems.

The KI 525A PNI provides a pictorial display of the horizontal navigation situation. It also provides manual controls for course and heading datum selections. The KI 229 RMI provides bearing information to either ADF or VOR stations by means of two pointers, each of which is read against the compass card.

The navigation model, which implements the navigation functions in the flight model, is used to provide the required information for the displays on the indicators. The simulation of these two indicators is quite difficult because of their complicated mechanical and electrical characteristics. The hardware and software required to read and drive these navigation indicators are still to be implemented. Better interface boards must be built to drive the three-phase synchros and resolvers in the PNI, and generate or interface AC/DC Sine/Cosine signals for the RMI.

CHAPTER 7
CONCLUSION

7.1. General Review

The purpose of this project was to enhance the development of the Concordia flight simulator system as a versatile research and teaching simulation tool. especially focus was on the development of the avionic system. This objective was achieved by replacing the original radio stack microprocessor hardware with a modern computer and advanced interfaces. The system allowed for the incorporation of a flexible radio stack module, navigation model and enhanced flight simulation software to be developed to exploit the academic potential of this system. This thesis aims at presenting not only the hardware configuration of the simulated radio stack system, but also the upgraded control module and navigation model. After introducing the history of flight simulation and the objectives of this project, the operation principles of a typical radio navigation system was introduced. Then, the development of the simulated radio stack system was documented, starting with the description of the reverse engineering for the radio stack hardware, and then the implementation of the upgraded radio stack module and navigation model into a working simulation program. A detail description of the operations of the radio stack was presented with a description of the method used to simulate them in program code. Finally, an insight into the two associated navigation indicators the PNI (Pictorial Navigation Indicator) and the RMI (Radio Magnetic Indicator), which could be used with the radio stack to form the advanced radio navigation systems, was offered.

7.2. Overview and Discussion

7.2.1. Radio Navigation Systems

One of the objectives of this project is to simulate and model the radio stack system. Chapter 2 presented the operation principles of the radio navigation systems found in the radio stack of the Concordia flight simulator. These systems include the ADF, VOR, ILS, DME, and RNAV systems.

Automatic Direction Finder (ADF) is the radio device that is used to sense and indicate the direction to a low/medium frequency non-directional radio beacon (NDB) ground transmitter. NDB is superimposed with a Morse code identifier. On the ADF instrument in the cockpit, the needle points towards the selected beacon and enables the pilot to fly the required procedure. Typical procedures in which NDBs are used are in approaches to airfields and keeping track of the aircraft when flying en-route.

VHF Omni-Range (VOR) is a ground-based electronic navigational aid transmitting very high-frequency navigation signals, 360 deg in azimuth, oriented from magnetic north. VOR periodically identifies itself by Morse code and may have an additional voice identification feature. This VHF Omni-Range navigation method relies on the ground based transmitters that emit signals to VOR receivers. The aircraft equipment receives these signals, calculates the difference between them, and interprets the result as a radial or bearing From/To the ground station. VOR is the most commonly used radio navigation aid in use today.

Distance Measuring Equipment (DME) provides distance information and primarily serve operational needs of en-route or terminal area navigation. In the operation of DME, paired pulses at a specific spacing are sent out from the aircraft and are received at the ground station. The ground transponder then transmits paired pulses back to the aircraft at the same pulse spacing but on a different frequency. The time required for the round trip of this signal exchange is measured in the airborne DME unit and is translated into distance from the aircraft to the ground station.

Instrument Landing System (ILS) is a precision-instrument approach system that consists of the localizer, the glideslope and marker radio beacons (outer, middle, and inner). It provides lateral, longitudinal and vertical guidance for the approach. The relative position of the aircraft to the ideal flight path is indicated by the needles of the LOC/GS indicator or by the HSI. The three marker beacons aligned with the runway indicate the distance of the runway threshold. The marker beacon receiver announces the beacon passing both visually and audibly.

Area Navigation (RNAV) is a system of radio navigation that permits direct point-to-point off-airways navigation by means of an on-board computer creating phantom VOR/DME transmitters termed waypoints. RNAV equipment can compute the airplane position, actual track and ground speed and then provide meaningful information relative to a route of flight selected by the pilot. Distance, time, bearing and cross-track error relative to the selected waypoint and the selected route are provided to the pilot. RNAV was developed to provide more lateral freedom and thus more complete use of available airspace.

The operation theories of these radio navigation systems provide the fundamental knowledge for processing this project.

7.2.2. Reverse Engineering

Upgrading the avionic system in the Concordia flight simulator is one of the objectives of this project. Without the reverse engineering, the upgrade would not have been possible because several important documents of the radio stack were not available any longer. Chapter 3 described the hardware reverse engineering work for the simulated radio stack system.

The maintenance manuals from the radio stack manufacturer BINDEX/KING Company were the fundamental materials for the design of the simulated radio stack system. Without the help of them, the reverse engineering would not have been able to progress. This simulated radio stack system consists of ADF, RNAV, NAV/COMM, COMM and ATC XPNDR stations with the associated DME, MBR and the two navigation indicators PNI and RMI.

Most of the stations in the radio stack contain three electrical printed circuit boards: interface board, display board and switchboard. In the ADF section, the switch circuit, display circuit and dimming circuit were described in detail. Similar design can be found in every station of the simulated radio stack system. All the interface boards, which were custom designed and built, interfaced the simulated stations to a microprocessor system that was used to control the entire radio stack system.

The reverse engineering was always started from the connector on each board. As long as the function of each pin in the connector was determined, along the signal cable connections, the function of each chip on each board could be figured out. Through the reverse engineering, the hardware configuration and the operation principles of the simulated radio stack system are presented.

7.2.3. Radio Stack System and Radio Stack Module

Chapter 4 presented the upgraded radio stack system and the computer code implementation of the radio stack module. The upgraded radio stack system represents the advanced interface technology and provides better performance. The radio stack module, which is used to control and drive the upgraded radio stack system, is able to acknowledge the status of the switches on the front panels and generate the corresponding data for display.

Since the original radio stack module was damaged, an upgraded radio stack module stored in the flight model computer was programmed to replace it. First of all, several programmable functional IC chips were programmed. Then, the high-speed communication between the simulated radio stack and the flight model computer was selected. The considerations included RS232, USB, Parallel Port, Ethernet, and finally the advanced data acquisition card NI PCI-6601 was selected.

The VC++ radio stack module was programmed to demonstrate that the PCI 6601 card could be accessed without a complicated Kernel driver under Windows XP. In addition, this VC++ module could be easily integrated with the existing C flight dynamic model in the future research.

The LabVIEW radio stack module was programmed to achieve one of the objectives of this project that is to offer graphical user interfaces (GUI) for students to experience radio navigation systems without any simulated hardware connected to the PCs. Furthermore, by using LabVIEW with the built-in I/O functions, it became easy to graphically represent and control the upgraded radio stack system on a computer screen.

7.2.4. Navigation Model

Chapter 5 presented the computer code implementation of the navigation model. The so-called navigation model is a software package used to implement the radio stack system's navigation functions in a flight simulation.

In the existing system, a simple navigation model done by Mr. P. Lawn was integrated with his flight dynamic model. In this project, a more sophisticated and versatile navigation model was programmed for students to experience the radio navigation systems without the complicated flight dynamic model since this project focuses on the development of the avionic system. The upgraded navigation model was coded in VC++ so that it could be integrated with the existing flight dynamic model, and be modified by students in the future research.

This navigation model includes the Timer, ADF, DME, and VOR/ILS modules. The Timer module keeps track of the elapsed time and the period of iteration Δt so that the program could run in real time. The ADF, DME, and VOR/ILS modules respectively simulate the ADF, DME, VOR and ILS systems in the similar manner to the existing navigation model, but more functions are implemented in the upgraded navigation model. This navigation model is able to communicate with the radio stack and the associated navigational instruments, and provide all the required information (such as range and bearing) for the avionic navigation systems in the Concordia flight simulator.

The navigation model would make more sense when it is integrated with the flight dynamic model in a completed flight simulation.

7.2.5. Associated Navigation Indicators

In Chapter 6, the operation and simulation of the Radio Magnetic Indicator (RMI) and the Pictorial Navigation Indicator (PNI) were presented. These two indicators could be used with the radio stack to form the advanced radio navigation systems, such as NAV/COMM systems and NAV/RNAV/COMM systems.

The KI 525A PNI provides a pictorial display of the horizontal navigation situation with manual controls for course and heading datum selections. The KI 229 RMI provides bearing information to either ADF or VOR stations by means of two pointers.

The navigation model. which implements the navigation functions in the flight model, is used to provide the required information for the displays on these two indicators. However, the hardware and software required to read and drive these navigation indicators are still to be implemented. Better interface boards must be used to drive the three-phase synchros and resolvers in the PNI, and generate or interface AC/DC Sine/Cosine signals for the RMI.

7.3. Future Work

This thesis continued the new level of development of the Concordia flight simulator, especially focused on the development of the avionic system. The simulated radio stack system, the radio stack module and the navigation model have been upgraded to improve the performances, include more functions, and enhance the Concordia flight simulator as an useful academic tool. Most features necessary to drive the upgraded radio stack system have been completed, but there still remain a number of aspects to be addressed as future work. Many of them could be the topic of student course projects.

7.3.1. VC++ Radio Stack Module and Navigation Model

In this project, the radio stack control module and navigation model were upgraded and coded in VC++. In the future research, these two software packages should be integrated with the flight dynamic model to complete the flight simulation.

The radio stack module is used to control and drive the radio stack hardware. In the upgraded radio stack system, this module is stored in the flight model computer instead of the original microprocessor system. If a complete flight simulator including the radio stack was required, the radio stack module should be integrated with the flight dynamic model.

The navigation model is only able to implement the navigation functions. It would make more sense for its integration with the flight dynamic model in a complete flight simulator.

7.3.2. Navigational Instruments

Through this project the simulation of the Radio Magnetic Indicator (RMI) and the Pictorial Navigation Indicator (PNI) has begun. While the code has been written to simulate these two indicators, the hardware and software required to read and drive them is still to be implemented. Custom interface boards must be built to drive the three-phase synchros and resolvers in the PNI, and generate or interface AC/DC Sine/Cosine signals for the RMI. These interface cards could be also used to simulate other complicated navigational instruments in the future research.

7.3.3. Sound Simulation

The audio cues to a pilot are key feedback (such as from engine, radio and other hardware) sounds to enhance situation awareness. Many actual avionic navigation systems have the function of audio announcement. such as Morse code and voice identification. In anticipation of this, a commercially available Sound Blaster card has been installed in the new computer. Through the many commercially available programs on the market that make extensive use of this card, there is no question that it is capable of generating realistic sounds for the simulation. In the future research. the sound simulation would be a great enhancement for the Concordia flight simulator.

7.3.4. Ethernet Network

During the discussion to select the interface between the simulated radio stack and the flight model computer. the Ethernet network had been considered. Although it was not used in this application. it would be an inspired idea to build an Ethernet network among the flight model computer, the instructor station and the visual system. Since the flight model computer is now required to control more hardware and run more complicated simulation software, it should not have any noticeable time waste in communication. In the near future, the video signals from the visual system computer would require larger communication bandwidth and faster data transfer rate. The advanced Ethernet network could provide an effective method to organize the different working sections in the future flight simulator system.

7.4. Concluding Remarks

The contribution of this thesis is to document the further enhancements of the Concordia flight simulator to function as a powerful academic tool, especially focusing on the design improvement and development of the radio stack system. From reverse engineering to simulation software programming, while there still remains work to be done to complete the task, it has provided undeniable insight and understanding of the knowledge. Hopefully, it is just a hint of what it will finally be capable of, once completed.

REFERENCES

[1.1] J.M.Rolfe & K.J.Staples, Flight Simulation. Cambridge University Press, New York, 1986

[1.2] Flights of Fancy, http://www.bleep.demon.co.uk/SimHist1.html

[1.3] Aeroplane Maintenance & Operation Series. Vol. 8: The Link Trainer

[1.4] 50 Years of Flight Simulation: Conference Proceedings. Royal Aeronautical Society, London, 1979

[1.5] Atlantis Systems International, http://www.atlantissi.com/7ftds.htm

[1.6] J. Svoboda, S. Lasnier, G.M. McKinnon, Microprocessor-Bases Light Aircraft Simulator, Canadian Industrial Computer Society, Ottawa, 1986

[1.7] Peter Lawn, The enhancement of a flight simulator system with teaching and research applications. Thesis (M.A.Sc.)--Dept. of Mechanical Engineering, Concordia University, 1998

[1.8] PCI 6601, http://sine.ni.com/apps/we/nioc.vp?cid=3589&lang=US, National Instruments Inc.

[1.9] DAQ 6601/6602 User Manual, National Instruments Inc., January 1999

[2.1] M. J. Culhane, Commercial Pilot Ground School Course, Accelerated Aviation Training, Vancouver, 1997

[2.2] From The Ground Up, 25th Edition, Aviation Publishers, Ottawa, 1987

[2.3] Myron Kayton and Walter R. Fried, Avionics Navigation Systems, John Wiley & Sons, 1969

[2.4] Online Catalog Avionics: ADF, http://www.seaerospace.com/king/ka44b.htm, Southeast Aerospace Inc., 2002

[2.5] Airborne ADF System Marks, Arinc Characteristic 570, Aeronautical Radio Inc., 1968

[2.6] Transport Canada, Flight Training Manual, 3rd Edition, Gage Educational Publishing Company, Toronto, 1989

[2.7] Canada Flight Supplement, Natural Resources Canada, July 17, 1997

[2.8] Transport Canada, Flight Training Manual, 4th Edition, Gage Educational Publishing Company, Toronto, 1994

[2.9] Airborne VOR Receiver, Arinc Characteristic 579-2, Aeronautical Radion Inc., 1989

[2.10] Precision Airborne Distance Measuring Equipment, Arinc Characteristic 709A, Aeronautical Radio Inc., 1994

[2.11] Online Catalog Avionics: DME, http://www.seaerospace.com/collins/ind41.htm, Southeast Aerospace Inc., 2002

[2.12] Avionics Fundamentals, International Standard Book Number 0-89100-293-6

[2.13] J. Svoboda, Avionics Navigation Systems Class Notes, Concordia University, 2000

[2.14] Joan L. Tomsic, SAE Dictionary of Aerospace Engineering, 2nd Edition, Society of Automotive Engineers, Inc., 1998

[3.1] The Royal Aeronautical Society, Key Trends in Human Factors of Aircraft Maintenance, One-Day Conference, London, October 1991

[3.2] KING, KY196A, KY196B & KY197A BENDIX/King COMM Transceivers, BENDIX/King Honeywell Inc, 2000

[3.3] http://www.shvets.com/genna/CPUs/8085/AMD-D8085A-2.html

[3.4] SN54154. SN74154 4-Line to 16-Line Decoders/Demultiplexers, Texas Instruments Corporation, March 1988

[3.5] LM555 Timer, National Semiconductor Corporation, February 2000

[3.6] Cornell University, http://instruct1.cit.cornell.edu/courses/ee476/labs/1997/lab2.html

[3.7] A.L. Benton, N.R.Varney and K.d. Hamsher, Visuopotential Judgement. A Clinical Test. Arch Neurol Vol 35, June 1978

[3.8] LM324 Quadruple Operational Amplifiers, Texas instruments Corporation, 1997

[3.9] DM74LS138. DM74LS139 Decoders/Demultiplexers. Fairchild Semiconductor Corporation, March 1988

[3.10] KING, KDI 572/573/574 DME Indicators Maintenance/Overhaul Manual, King Radio Corporation, November 1978

[3.11] C8279 Programmable Keyboard/Display Interface. CAST Inc., February 2000

[3.12] KING, KR87 Automatic Direction Finder Maintenance/Overhaul Manual, King Radio Corporation, July 1979

[3.13] Texas instruments, CD4017B CD4022B Types COMS Counter/Drivers, Texas Instruments Corporation, 1999

[3.14] Walter A. Triebel, The 80386. 80486. and Pentium Processor Hardware. Software. and Interfacing, Prentice-Hall Inc., May 1999

[3.15] Texas instruments, SN74LS123 Retriggerable Monostable Multivibrators, Texas instruments Corporation, March 1988

[3.16] National Semiconductor, DS8884A High Voltage Cathode Decoder/Driver, National Semiconductor Corporation, May 1986

[3.17] KING, KNS 81 Digital Area Navigation System Maintenance/Overhaul Manual, King Radio Corporation, January 1980

[3.18] KING, KX 155/165 NAV COMM Receiver Maintenance/Overhaul Manual, King Radio Corporation, January 1981

[3.19] KING, KY 196 COMM Transceiver Maintenance/Overhaul Manual, King Radio Corporation, August 1979

[3.20] KING, KT 79 Transponder Maintenance/Overhaul Manual, King Radio Corporation, April 1982

[4.1] SN54154, SN74154 4-Line to 16-Line Decoders/Demultiplexers, Texas Instruments Corporation, March 1988

[4.2] National Semiconductor, DS8884A High Voltage Cathode Decoder/Driver, National Semiconductor Corporation, May 1986

[4.3] Dionics, Gas Discharge Display Segment Driver DI-230A DI-240A, Dionics Inc., January 2002

[4.4] Walter A. Triebel, The 80386, 80486, and Pentium Processor Hardware, Software, and Interfacing, Prentice-Hall Inc., May 1999

[4.5] Kathleen Conlon Hinge, Electronic data interchange : from understanding to implementation, AMA Membership Publications Division, American Management Association, New York, 1988

[4.6] Universal Serial Bus Specification Revision 2.0, Compaq/Hewlett-Packard/Intel/Lucent/Microsoft/NEC/Philips, April 2000

[4.7] William J. Beyda, Basic data communications: a comprehensive overview, Englewood Cliffs, N.J.: Prentice Hall, 1989

[4.8] MSDN Online Library - Parallel Ports and Devices, Microsoft Corporation http://msdn.microsoft.com/library/default.asp?url=/library/en-us/parallel/hh/parallel/vspd_461z.asp

[4.9] Charles E. Spurgeon, Ethernet: The Definitive Guide, O'Reilly and Associates, February 2000

[4.10] Digital Input-Output - Products and Services - National Instruments. http://sine.ni.com/apps/we/nioc.vp?cid=1102&lang=US

[4.11] National Instruments, DAQ 6601/6602 User Manual, National Instrument Corporation. January 1999

[4.12] National Instruments, Getting Started with LabVIEW, National Instrument Corporation, November 2001

[5.1] Raj Jain, The Art of Computer Systems Performance Analysis, John Wiley & Sons Inc., 1991

[5.2] William Buchanan, PC Interfacing, Communications and Windows Programming, John Wiley & Sons Inc., 1991

[5.3] Peter H. Zipfel, Modeling and Simulation of Aerospace Vehicle Dynamics, American Institute of Aeronautics and Astronautics Inc., 2000

[5.4] Peter Lawn, The enhancement of a flight simulator system with teaching and research applications, Thesis (M.A.Sc.)--Dept. of Mechanical Engineering, Concordia University, 1998

[5.5] Software Design Document for the Radio Navigation System of the Dornier 328 Flight Simulator, CAE Electronics, Montreal, 1998

[6.1] Pilot's Guide KCS 55A Bendix/King Compass System, BENDIX/KING Inc., 1992

[6.2] KI 525A Pictorial Navigation Indicator Maintenance/Overhaul Manual, KING Avionics, May 1980

[6.3] KI 229 Radio Magnetic Indicator Installation Manual, KING Avionics, May 1980

[6.4] KI 229 Radio Magnetic Indicator Maintenance/Overhaul Manual, KING Avionics, May 1980

APPENDIX 1

VOR, LOCALIZER AND GLIDESLOPE FREQUENCIES

Freq (MHz)	VOR/LOC	GS (MHz)
108.00	VOR	
108.05	VOR	
108.10	ILS	334.70
108.15	ILS	334.55
108.20	VOR	
108.25	VOR	
108.30	ILS	334.10
108.35	ILS	333.95
108.40	VOR	
108.45	VOR	
108.50	ILS	329.90
108.55	ILS	329.75
108.60	VOR	
108.65	VOR	
108.70	ILS	330.50
108.75	ILS	330.35
108.80	VOR	
108.85	VOR	
108.90	ILS	329.30
108.95	ILS	329.15
109.00	VOR	
109.05	VOR	
109.10	ILS	331.40
109.15	ILS	331.25
109.20	VOR	
109.25	VOR	
109.30	ILS	332.00
109.35	ILS	331.85
109.40	VOR	
109.45	VOR	
109.50	ILS	332.60
109.55	ILS	332.45
109.60	VOR	
109.65	VOR	
109.70	ILS	333.20
109.75	ILS	333.05
109.80	VOR	
109.85	VOR	
109.90	ILS	333.80
109.95	ILS	333.65
110.00	VOR	
110.05	VOR	
110.10	ILS	334.40
110.15	ILS	334.25
110.20	VOR	
110.25	VOR	
110.30	ILS	335.00
110.35	ILS	334.85
110.40	VOR	
110.45	VOR	
110.50	ILS	329.60
110.55	ILS	329.45
110.60	VOR	
110.65	VOR	
110.70	ILS	330.20
110.75	ILS	330.05
110.80	VOR	
110.85	VOR	
110.90	ILS	330.80
110.95	ILS	330.65
111.00	VOR	
111.05	VOR	
111.10	ILS	331.70
111.15	ILS	331.55
111.20	VOR	
111.25	VOR	
111.30	ILS	332.30
111.35	ILS	332.15
111.40	VOR	
111.45	VOR	
111.50	ILS	332.90
111.55	ILS	332.75
111.60	VOR	
111.65	VOR	
111.70	ILS	333.50
111.75	ILS	333.35
111.80	VOR	
111.85	VOR	
111.90	ILS	331.10
111.95	ILS	330.95
112.00	VOR	
112.05	VOR	
112.10	VOR	
112.15	VOR	
112.20	VOR	
112.25	VOR	
112.30	VOR	
112.35	VOR	

112.40	VOR	114.65	VOR
112.45	VOR	114.70	VOR
112.50	VOR	114.75	VOR
112.55	VOR	114.80	VOR
112.60	VOR	114.85	VOR
112.65	VOR	114.90	VOR
112.70	VOR	114.95	VOR
112.75	VOR	115.00	VOR
112.80	VOR	115.05	VOR
112.85	VOR	115.10	VOR
112.90	VOR	115.15	VOR
112.95	VOR	115.20	VOR
113.00	VOR	115.25	VOR
113.05	VOR	115.30	VOR
113.10	VOR	115.35	VOR
113.15	VOR	115.40	VOR
113.20	VOR	115.45	VOR
113.25	VOR	115.50	VOR
113.30	VOR	115.55	VOR
113.35	VOR	115.60	VOR
113.40	VOR	115.65	VOR
113.45	VOR	115.70	VOR
113.50	VOR	115.75	VOR
113.55	VOR	115.80	VOR
113.60	VOR	115.85	VOR
113.65	VOR	115.90	VOR
113.70	VOR	115.95	VOR
113.75	VOR	116.00	VOR
113.80	VOR	116.05	VOR
113.85	VOR	116.10	VOR
113.90	VOR	116.15	VOR
113.95	VOR	116.20	VOR
114.00	VOR	116.25	VOR
114.05	VOR	116.30	VOR
114.10	VOR	116.35	VOR
114.15	VOR	116.40	VOR
114.20	VOR	116.45	VOR
114.25	VOR	116.50	VOR
114.30	VOR	116.55	VOR
114.35	VOR	116.60	VOR
114.40	VOR	116.65	VOR
114.45	VOR	116.70	VOR
114.50	VOR	116.75	VOR
114.55	VOR	116.80	VOR
114.60	VOR	116.85	VOR
116.90	VOR	116.95	VOR

117.00	VOR
117.05	VOR
117.10	VOR
117.15	VOR
117.20	VOR
117.25	VOR
117.30	VOR
117.35	VOR
117.40	VOR
117.45	VOR
117.50	VOR
117.55	VOR
117.60	VOR
117.65	VOR
117.70	VOR
117.75	VOR
117.80	VOR
117.85	VOR
117.90	VOR
117.95	VOR

APPENDIX 2

POWER SUPPLY AND BACK PANEL PINOUTS

```
1                13
 ________________
\ . . . . . . . . . . . .. /
 \ . . . . . . . . . . . /
  -----------------
14               25
```

LIGHT BOARD CONNECTOR (Back Panel)

LTNG LO	1.	14.	LTNG LO
LTNG HI (RNAV)	2.	15.	LTNG HI (ADF)
	3.	16.	LTNG HI (IFF)
	4.	17.	
	5.	18.	
	6.	19.	
	7.	20.	
	8.	21.	
	9.	22.	
	10.	23.	
	11.	24.	
	12.	25.	
	13.		

AUDIO CONNECTOR (Back Panel)

LED GND	1.	14.	LED HI
TEST SW (C/N)	2.	15.	TEST Common (C/N)
TEST Common (COMM)	3.	16.	TEST SW (COMM)
Audio OUT (ADF)	4.	17.	Audio IN (ADF)
IDT Vol. HI (C/N)	5.	18.	IDT Vol. Center (C/N)
Audio OUT (RNAV)	6.	19.	Audio IN (RNAV)
Test Vol. HI (C/N)	7.	20.	Test Vol. Center (C/N)
POT HI (COMM)	8.	21.	POT LO (COMM)
Audio GND	9.	22.	Audio GND
	10.	23.	
	11.	24.	
	12.	25.	
	13.		

DME CONNECTOR (Back Panel)

	GND	1.	14.	N2 SW	
8279/2	HLD SW	2.	15.	N1 SW	
	GND	3.	16.	OFF SW	
8889/2	OUTA3	4.	17.	D0	7407/10 (M)
8889/4	OUTB0	5.	18.	D1	7407/12 (M)
8889/5	OUTB1	6.	19.	D2	7407/2 (R)
8889/6	OUTB2	7.	20.	Dimming CLK	8279/23 555/4
8889/7	OUTB3	8.	21.	MODE SW Common	74LS139/4
8889/8	OUTA0	9.	22.	D3	7407/4 (R)
8889/9	OUTA2	10.	23.	H.V. +185V	
8889/3	OUTA1	11.	24.	+5V	
7408/8	SYNC	12.	25.	GND	
8279/23	SCAN	13.			

POWER SUPPLY JACK (Back Panel)

No.	1	2	3	4		5
Voltage	+5V	+185V	-9V	+9V		GND
Color	White	Blue	Red	Green		Black

Power Supply Lines (On ISA Bus)

1	2	3	4	5	6	7	8
+5V	GND	+185V			-9V	+9V	
White	Black	Blue			Red	Green	

Radio Stack Stations Position

TOP	ADF
	RNAV
	COMM/NAV (C/N)
	COMM
Bottom	IFF

APPENDIX 3
ISA BUS PINOUTS

ISA BUS

+5V	1.	56	+5V
GND	2.	55	GND
H.V. +185V	3.	54	H.V. +185V
D7	4.	53	D3
D6	5.	52	D2
D5	6.	51	D1
D4	7.	50	D0
	8.	79	
	9.	78	
A (5)	10.	47	
A (4)	11.	46	
A (3)	12.	45	
A (2)	13.	44	
A (1)	14.	43	
RD	15.	42	A0
	16.	41	WR
	17.	40	A (6)
	18.	39	
	19.	38	
	20.	37	
	21.	36	
	22.	35	
	23.	34	
A(7) LED HI	24.	33	RESET
	25.	32	CLK
	26.	31	
	27.	30	
-9V	28.	29	+9V

APPENDIX 4

AUDIO/DME INTERFACE BOARD PINOUTS

COMMON BUS CONNECTOR

(LO) Audio GND	1.	64	Audio GND (LO)
(COMM) Pot HI	2.	63	Pot LO (COMM)
(C/N) Test Vol. HI	3.	62	Test Vol. Center (C/N)
(RNAV) IDT	4.	61	Audio IN (RNAV)
(C/N) IDT Vol. HI	5.	60	IDT Vol. Center (C/N)
(ADF) Audio OUT	6.	59	Audio IN (ADF)
Test Common (COMM)	7.	58	Test SW (COMM)
(C/N) Test SW	8.	57	Test Common (C/N)
	9.	56	
(NAV/COMM) CS2	10.	55	
(NAV/COMM) CS3	11.	54	CS0 (IFF)
	12.	53	CS1 (ADF)
D0	13.	52	CS6 (COMM)
D1	14.	51	-RD
D2	15.	50	-WR
D3	16.	49	CLK
D4	17.	48	A0
D5	18.	47	RESET
D6	19.	46	
D7	20.	45	CLEAR
(RNAV) CS4	21.	44	High Freq.
(RNAV) CS5	22.	43	CS7 (DME)
	23.	42	
	24.	41	
	25.	40	LTNG HI (RNAV)
	26.	39	
5V	27.	38	LTNG HI (IFF)
9V	28.	37	LTNG HI (ADF)
GND	29.	36	GND
GND	30.	35	H.V. +185V
	31.	34	
LTNG LO	32.	33	LTNG LO

C/N: COMM/NAV

DME BUS CONNECTOR

Audio/1	(LO) Audio GND	1.	64	Audio GND (LO)	Audio/26
Audio/25	(COMM) Pot HI	2.	63	Pot LO (COMM)	LM324/3 (L)
Audio/24	(C/N) Test Vol. HI	3.	62	Test Vol. Center (C/N)	LM324/5 (L)
Audio/23	(RNAV) IDT	4.	61	Audio IN (RNAV)	LM324/10 (L)
Audio/22	(C/N) IDT Vol. HI	5.	60	IDT Vol. Center (C/N)	LM324/12 (L)
Audio/21	(ADF) Audio OUT	6.	59	Audio IN (ADF)	LM324/3 (R)
Audio/20	Test Common (COMM)	7.	58	Test SW (COMM)	Audio/7
Audio/19	(C/N) Test SW	8.	57	Test Common (C/N)	Audio/8
		9.	56		
74LS154/4	(NAV/COMM) CS2	10.	55		
74LS154/5	(NAV/COMM) CS3	11.	54	CS0 (IFF)	74LS154/1
		12.	53	CS1 (ADF)	74LS154/2
8279/12	D0	13.	52	CS6 (COMM)	74LS154/3
8279/13	D1	14.	51	-RD	8279/10 74LS32/6
8279/14	D2	15.	50	-WR	8279/11 74LS32/3
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9 74LS04/4
8279/18	D6	19.	46		
8279/19	D7	20.	45	CLEAR	74LS154/8
74LS154/7	(RNAV) CS4	21.	44	High Freq.	74LS04/6
74LS154/6	(RNAV) CS5	22.	43	CS7 (DME)	74LS154/9
		23.	42		
		24.	41		
		25.	40	LTNG HI (RNAV)	Audio/12
		26.	39		
	5V	27.	38	LTNG HI (IFF)	Audio/16
	9V	28.	37	LTNG HI (ADF)	Audio/15
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
	LTNG LO	32.	33	LTNG LO	

DME ISA BUS

	+5V	1.	56	+5V	
	GND	2.	55	GND	
	H.V. +185V	3.	54	H.V. +185V	
8279/19	D7	4.	53	D3	8279/15
8279/18	D6	5.	52	D2	8279/14
8279/17	D5	6.	51	D1	8279/13
8279/16	D4	7.	50	D0	8279/12
		8.	79		
		9.	78		
74LS04/1	A (5)	10.	47		
74LS154/20	A (1)	11.	46		
74LS154/21	A (2)	12.	45		
74LS154/22	A (3)	13.	44		
74LS154/23	A (4)	14.	43		
74LS32/5	RD	15.	42	A0	8279/21
		16.	41	WR	74LS32/1
		17.	40	A (6)	74LS123/1 (T)
		18.	39		
		19.	38		
		20.	37		
		21.	36		
		22.	35		
		23.	34		
LED Audio/9	A (7) LED HI	24.	33	RESET	73LS04/3
		25.	32	CLK	8279/3
		26.	31		
		27.	30		
-9V LM324/11	-9V	28.	29	+9V	+9V LM324/4

DME CONNECTOR (DME Interface)

7407/6(M)	CLK SCAN	1.	26	GND	
7407/8(M)	RESET SYNC	2.	25	+5V	
7407/12(L)	OUTA1	3.	24	H.V. +185V	
7407/2(M)	OUTA2	4.	23	D3	7407/4(R)
7407/10L(M)	OUTA0	5.	22	SW COMMON	74LS139/4
7407/8(L)	OUTB3	6.	21	CLK SCAN (Dimming)	7407/6(M)
7407/6(L)	OUTB2	7.	20	D2	7407/2(R)
7407/4 (L)	OUTB1	8.	19	D1	7407/12(M)
7407/2 (L)	OUTB0	9.	18	D0	7407/10(M)
7407/4 (M)	OUTA3	10.	17	RL0	8279/38 R-
	GND	11.	16	RL1	8279/39 R-
R- 8279/1	RL2	12.	15	RL3	8279/2 R-
		13.	14		

AUDIO CONNECTOR (DME Interface)

BUS/1	Audio GND	1.	26	Audio GND	BUS/64
LM324/1,2 (L)	Pot LO (COMM)	2.	25	Pot HI (COMM)	BUS/2
LM324/6,7 (L)	Test Vol. Center (C/N)	3.	24	Test Vol. HI (C/N)	BUS/3
LM324/8,9 (L)	Audio IN (RNAV)	4.	23	Audio OUT (RNAV)	BUS/4
LM324/13,14 (L)	IDT Vol. Center (C/N)	5.	22	IDT Vol. HI (C/N)	BUS/5
LM324/1,2 (R)	Audio IN (ADF)	6.	21	Audio OUT (ADF)	BUS/6
BUS/58	Test SW (COMM)	7.	20	Test GND (COMM)	BUS/7
BUS/57	Test GND (C/N)	8.	19	TEST SW (C/N)	BUS/8
ISA/24	LED HI	9.	18	LED LO	GND
		10.	17		
		11.	16	LTNG HI (IFF)	BUS/38
BUS/40	LTNG HI (RNAV)	12.	15	LTNG HI (ADF)	BUS/37
BUS/32	LTNG LO	13.	14	LTNG LO	BUS/33

APPENDIX 5

SELECTED 8279 CONTROLLER SPECIFICATIONS

CAST

C8279
Programmable Keyboard Display Interface Megafunction

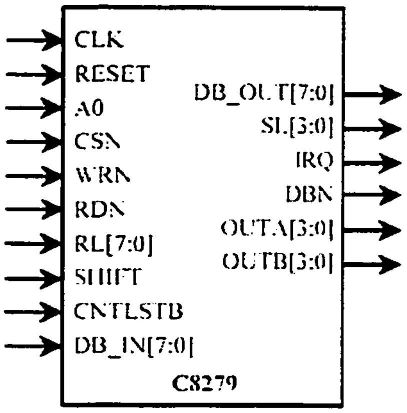
General Description

The C8279 is a programmable keyboard and display interface designed for use with microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The display portion provides a scanned display interface for LED, incandescent and other display technologies.

Features

- Simultaneous Keyboard Display Operation
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-key Rollover with Contact Debounce
- Dual 8- or 16-Numeric Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Interrupt Scan Timing
- Interrupt Output on Key Entry
- Functionality based on the Intel 8279

Symbol



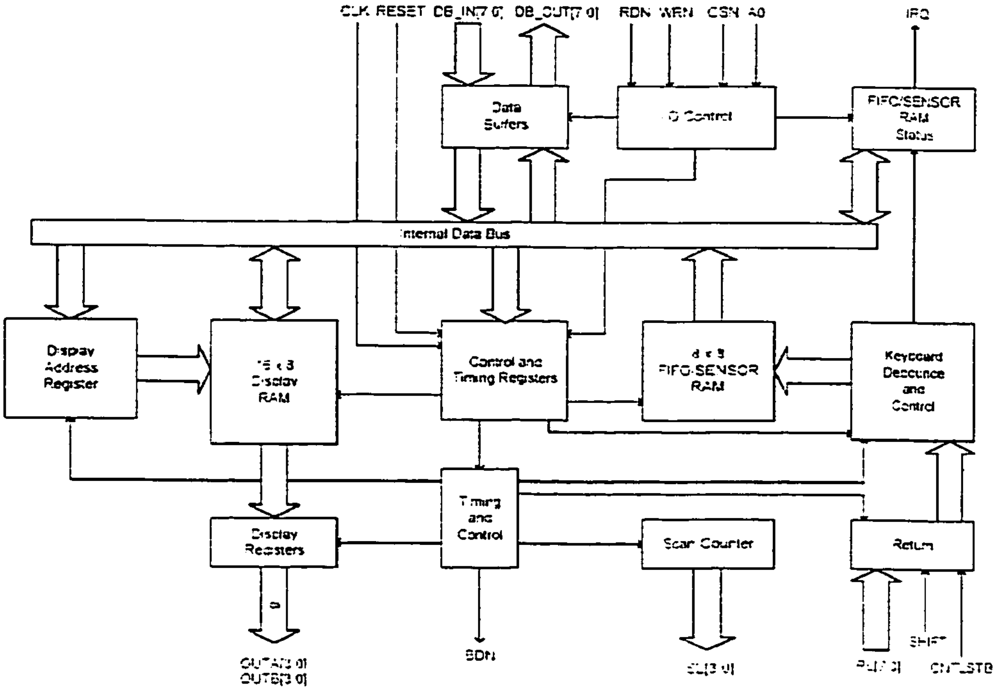
Pin Description

Name	Type	Polarity	Description
CLK	In	Rising	Clock
RESET	In	High	Reset
CSN	In	Low	Chip Select
A0	In	-	Buffer Address
RDN	In	Low	Input/Output Read
WRN	In	Low	Input/Output Write
RL[7:0]	In	-	Return Lines
SHIFT	In	-	Shift Input Status
CNTSTB	In	-	Control/Strobed Input Mode
DB_IN[7:0]	In	-	Data Bus (input side)
DB_OUT[7:0]	Out	-	Data Bus (output side)
PDBTRI	Out	Low	Tri-State signal for DB_OUT
IRQ	Out	-	Interrupt Request
SL[3:0]	Out	-	Scan Lines
OUTA[3:0]	Out	-	Outputs A for 16x4 display refresh registers
OUTB[3:0]	Out	-	Outputs B for 16x4 display refresh registers
BDN	Out	Low	Blank Display

Applications

- Point-of-contact Kiosk
- Medical instrumentation
- Test & Measurement Instrumentation
- Industrial Equipment
- Avionics
- Gaming & Amusement Machines

Block Diagram



Device Utilization & Performance

Target Device	Speed Grade	Utilization		Performance	Availability
		LCs	EABs	F$_{max}$	
EPF10K50S	-1	1801	-	144 MHz	Now
EP1K50	-1	1801	-	136 MHz	Now
EP1S10	-5	1627	-	189 MHz	Now

APPENDIX 6

ADF PINOUTS

ADF BUS CONNECTOR

SWITCH/1	Audio GND	1.	64		
		2.	63		
		3.	62		
		4.	61		
		5.	60		
SWITCH/40	Audio OUT	6.	59	Audio IN	SWITCH/39
		7.	58		
		8.	57		
		9.	56		
		10.	55		
		11.	54		
		12.	53	CS1	8279/22
8279/12	D0	13.	52		
8279/13	D1	14.	51	-RD	8279/10
8279/14	D2	15.	50	-WR	8279/11
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9
8279/18	D6	19.	46		
8279/19	D7	20.	45	CLEAR	74LS74/1
		21.	44	High Freq.	74LS139/14
		22.	43		
		23.	42		
		24.	41		
		25.	40		
		26.	39		
	5V	27.	38		
	9V	28.	37	LTNG HI	SWITCH/24
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
SWITCH/17	LTNG LO	32.	33		

Audio OUT: ADF SW, only when ADF is switched on, the volume control is available.

ADF SWITCH BOARD CONNECTOR

BUS/1	Audio GND	1.	40	Audio OUT	BUS/6
		2.	39	Audio IN	BUS/59
		3.	38		
7407/1 74LS139/11	Channel Common 1	4.	37	Channel Common 2	7407/3 74LS139/12
74LS173/7 (R) R-	INC SW	5.	36	DEC SW	74LS173/7 (L) R-
74LS74/4 R-	MSD/LSD	6.	35	MSD/LSD	74LS74/4 R-
		7.	34		
		8.	33	ON SW	8279/8
74LS139/4	Pull-Up SW Common	9.	32	ON SW Common	74LS139/4
8279/7	Pull-Up SW	10.	31	ADF SW Common	74LS139/4
74LS244/2	FRQ SW	11.	30	ADF SW	8279/5
74LS244/4	FLT/ET SW	12.	29	BFO SW	8279/6
74LS244/6	SET/RST SW	13.	28	BFO SW Common	74LS139/4
74LS139/5	SET/RST SW GND	14.	27		
		15.	26		
		16.	25	PHOTO	Dimming
BUS/32	LTNG LO	17.	24	LTNG HI	BUS/37
	+5V	18.	23	+9V	
	GND	19.	22	GND	
	GND	20.	21	H.V. +185V	

ADF DISPLAY BOARD CONNECTOR

8279/27	OUTA0	1.	16	OUTB2	8279/29
8279/26	OUTA1	2.	15	OUTB2	8279/29
8279/25	OUTA2	3.	14	OUTB2	8279/29
8279/24	OUTA3	4.	13		
8279/28	OUTB3	5.	12		
		6.	11	SCAN	7407/8
Dimming	PROG	7.	10	SYNC	7407/6
	GND	8.	9	GND	

APPENDIX 7

SELECTED CD4022B ANODE COUNTER SPECIFICATIONS

TEXAS INSTRUMENTS
Data sheet acquired from Harris Semiconductor
SCHS027

CD4017B, CD4022B Types

CMOS Counter/Dividers

High-Voltage Types (20-Volt Rating)
CD4017B—Decade Counter with 10 Decoded Outputs
CD4022B—Octal Counter with 8 Decoded Outputs

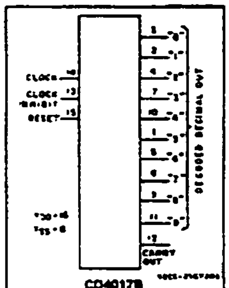
■ CD4017B and CD4022B are 5-stage and 4-stage Johnson counters having 10 and 8 decoded outputs, respectively. Inputs include a CLOCK, a RESET, and a CLOCK INHIBIT signal. Schmitt trigger action in the CLOCK input circuit provides pulse shaping that allows unlimited clock input pulse rise and fall times.

These counters are advanced one count at the positive clock signal transition if the CLOCK INHIBIT signal is low. Counter advancement via the clock line is inhibited when the CLOCK INHIBIT signal is high. A high RESET signal clears the counter to its zero count. Use of the Johnson counter configuration permits high-speed operation, 2-input decode-gating and spike-free decoded outputs. Anti-lock gating is provided, thus assuring proper counting sequence. The decoded outputs are normally low and go high only at their respective decoded time slot. Each decoded output remains high for one full clock cycle. A CARRY-OUT signal completes one cycle every 10 clock input cycles in the CD4017B or every 8 clock input cycles in the CD4022B and is used to ripple-clock the succeeding device in a multi-device counting chain.

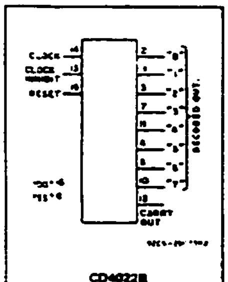
The CD4017B and CD4022B-series types are supplied in 16-lead hermetic dual-in-line ceramic packages (D and F suffixes), 16-lead dual-in-line plastic package (E suffix), 16-lead ceramic flat packages (K suffix), and in chip form (H suffix).

Features:
- Fully static operation
- Medium-speed operation . . . 10 MHz (typ.) at V_{DD} = 10 V
- Standardized, symmetrical output characteristics
- 100% tested for quiescent current at 20 V
- 5-V, 10-V, and 15-V parametric ratings
- Meets all requirements of JEDEC Tentative Standard No. 13A, "Standard Specifications for Description of 'B' Series CMOS Devices"

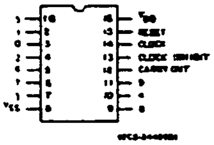
Applications:
- Decade counter/decimal decode display (CD4017B)
- Binary counter/decoder
- Frequency division
- Counter control/timers
- Divide-by-N counting
- For further application information, see ICAN-6166 "COS/MOS MSI Counter and Register Design and Applications"



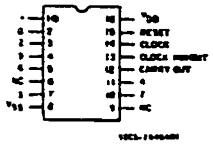
CD4017B
Functional Diagram



CD4022B
Functional Diagram



TOP VIEW
CD4017B
TERMINAL DIAGRAM



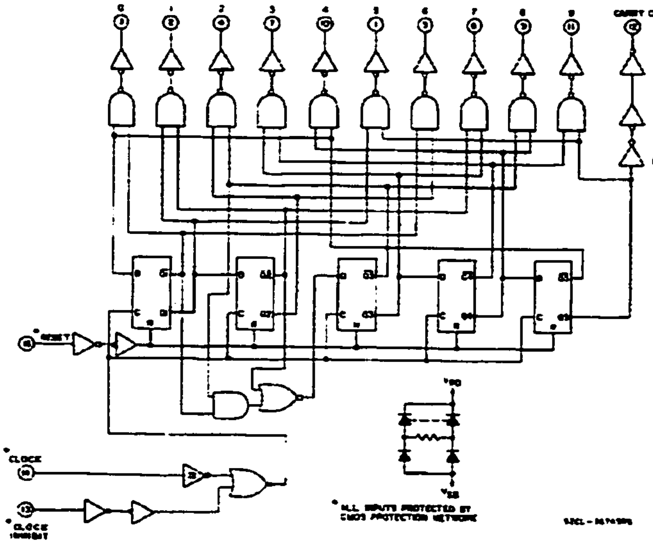
TOP VIEW
NC - no connection
CD4022B
TERMINAL DIAGRAM

RECOMMENDED OPERATING CONDITIONS

For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTICS	V_{DD} (V)	LIMITS		UNITS
		Min.	Max.	
Supply-Voltage Range (For T_A = Full Package-Temperature Range)		3	18	V
Clock Input Frequency, f_{CL}	5 10 15	– – –	2.5 5 5.5	MHz
Clock Pulse Width, t_W	5 10 15	200 90 60	– – –	ns
Clock Rise & Fall Time, t_{rCL}, t_{fCL}	5 10 15	UNLIMITED*		
Clock Inhibit Setup Time, t_s	5 10 15	230 100 70	– – –	ns
Reset Pulse Width, t_{RW}	5 10 15	260 110 60	– – –	ns
Reset Removal Time, t_{rem}	5 10 15	400 280 150	– – –	ns

*Only if Pin 14 is used as the clock input. If Pin 13 is used as the clock input and Pin 14 is tied high (for advancing count on negative transition of the clock), rise and fall time should be ≤ 15 μs.



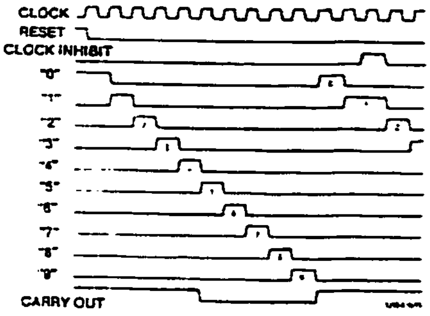


Fig. 2 - Timing diagram for CD4017B.

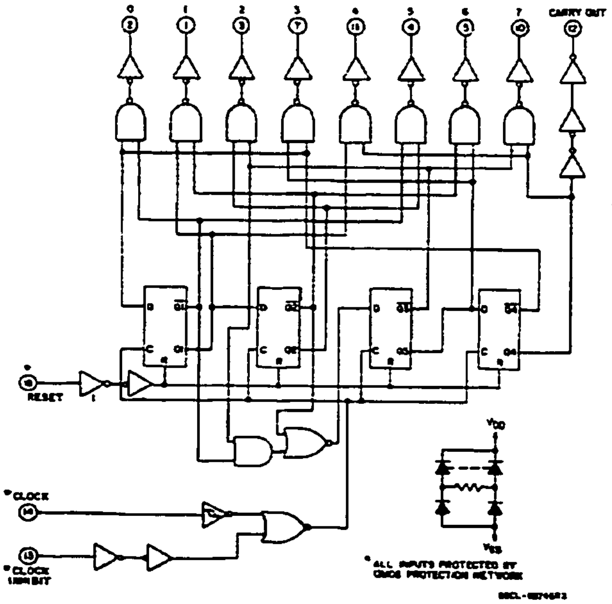


Fig. 3 - Logic diagram for CD4022B.

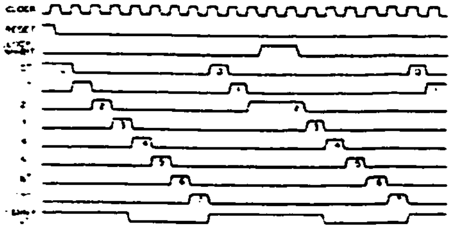


Fig. 4 - Timing diagram for CD4022B.

DYNAMIC ELECTRICAL CHARACTERISTICS
At T_A = 25°C, Input t_r, t_f = 20 ns, C_L = 50 pF, R_L = 200 kΩ

CHARACTERISTIC	CONDITIONS V_{DD} (V)	LIMITS Min.	Typ.	Max.	UNITS
CLOCKED OPERATION					
Propagation Delay Time, t_{PHL}, t_{PLH} Decode Out	5	–	325	650	ns
	10	–	135	270	
	15	–	85	170	
Carry Out	5	–	300	600	
	10	–	125	250	
	15	–	80	160	
Transition Time, t_{THL}, t_{TLH} Carry Out or Decode Out Line	5	–	100	200	ns
	10	–	50	100	
	15	–	40	80	
Maximum Clock Input Frequency, f_{CL}*	5	2.5	5	–	MHz
	10	5	10	–	
	15	5.5	11	–	
Minimum Clock Pulse Width, t_W	5	–	100	200	ns
	10	–	45	90	
	15	–	30	60	
Clock Rise or Fall Time, t_rCL, t_fCL	5, 10, 15	UNLIMITED			
Minimum Clock Inhibit to Clock Setup Time, t_s	5	–	115	230	ns
	10	–	50	100	
	15	–	35	70	
Input Capacitance, C_{IN}	Any Input	–	5	–	pF
RESET OPERATION					
Propagation Delay Time, t_{PHL}, t_{PLH} Carry Out or Decode Out Lines	5	–	265	530	ns
	10	–	115	230	
	15	–	85	170	
Minimum Reset Pulse Width, t_W	5	–	130	260	ns
	10	–	55	110	
	15	–	30	60	
Minimum Reset Removal Time	5	–	200	400	ns
	10	–	140	280	
	15	–	75	150	

* Measured with respect to carry output line.

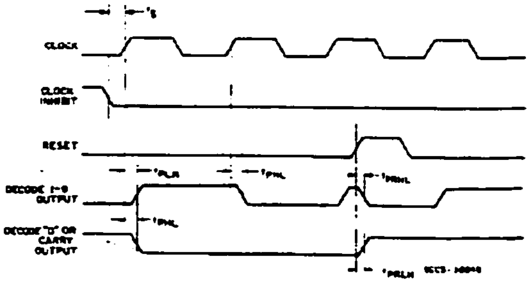


Fig 9 – Propagation delay, setup, and reset removal time waveforms.

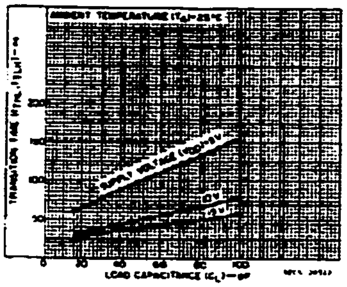


Fig. 10 – Typical transition time as a function of load capacitance.

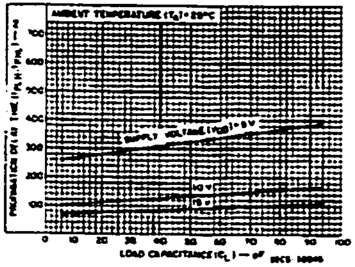


Fig. 11 – Typical propagation delay time as a function of load capacitance (clock to decode output).

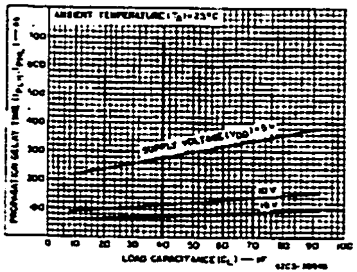


Fig. 12 – Typical propagation delay time as a function of load capacitance (clock to carry-out).

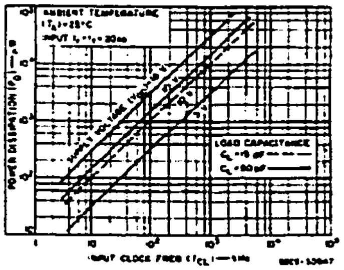


Fig. 13 – Typical dynamic power dissipation as a function of clock input frequency.

APPENDIX 8

SELECTED 74LS123 VIBRATOR SPECIFICATIONS

SN54122, SN54123, SN54130, SN54LS122, SN54LS123, SN74122, SN74123, SN74130, SN74LS122, SN74LS123 RETRIGGERABLE MONOSTABLE MULTIVIBRATORS

SDLS043 – DECEMBER 1983 – REVISED MARCH 1988

- **D-C Triggered from Active-High or Active-Low Gated Logic Inputs**
- **Retriggerable for Very Long Output Pulses, Up to 100% Duty Cycle**
- **Overriding Clear Terminates Output Pulse**
- **'122 and 'LS122 Have Internal Timing Resistors**

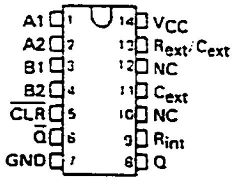
description

These d-c triggered multivibrators feature output pulse-duration control by three methods. The basic pulse time is programmed by selection of external resistance and capacitance values (see typical application data). The '122 and 'LS122 have internal timing resistors that allow the circuits to be used with only an external capacitor, if so desired. Once triggered, the basic pulse duration may be extended by retriggering the gated low-level-active (A) or high-level-active (B) inputs, or be reduced by use of the overriding clear. Figure 1 illustrates pulse control by retriggering and early clear.

The 'LS122 and 'LS123 are provided enough Schmitt hysteresis to ensure jitter-free triggering from the B input with transition rates as slow as 0.1 millivolt per nanosecond.

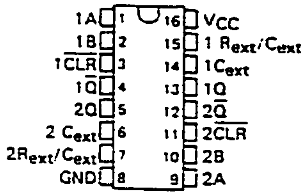
The R_{int} in nominall 10 kΩ for '122 and 'LS122.

SN54122, SN54LS122 . . . J OR W PACKAGE
SN74122 . . . N PACKAGE
SN74LS122 . . . D OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)

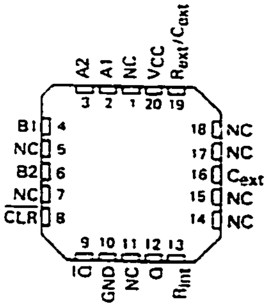


NOTES: 1. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).
2. To use the internal timing resistor of '122 or 'LS122, connect R_{int} to V_{CC}.
3. For improved pulse duration accuracy and repeatability, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
4. To obtain variable pulse durations, connect an external variable resistance between R_{int} or R_{ext}/C_{ext} and V_{CC}.

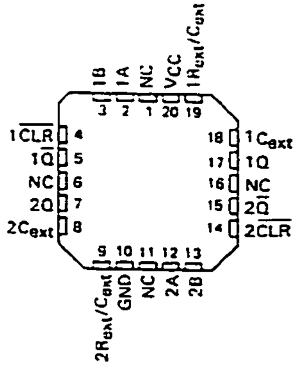
SN54123, SN54130, SN54LS123 . . . J OR W PACKAGE
SN74123, SN74130 . . . N PACKAGE
SN74LS123 . . . D OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



SN54LS122 . . . FK PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)



SN54LS123 . . . FK PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)

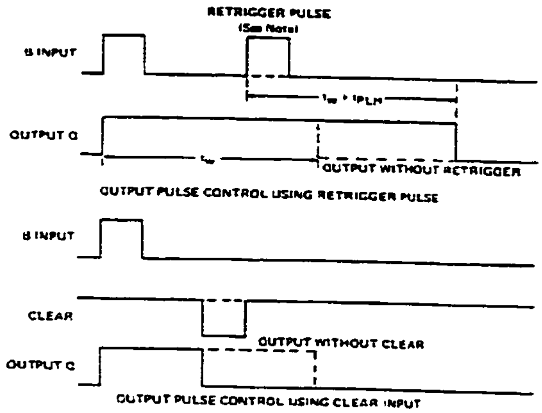


NC - No internal connection

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS

description (continued)



OUTPUT PULSE CONTROL USING RETRIGGER PULSE

OUTPUT PULSE CONTROL USING CLEAR INPUT

NOTE: Retrigger pulses starting before 0.22 C_{ext} (in picofrads) nanoseconds after the initial trigger pulse will be ignored and the output duration will remain unchanged.

FIGURE 1—TYPICAL INPUT/OUTPUT PULSES

'122, 'LS122
FUNCTION TABLE

INPUTS					OUTPUTS	
CLEAR	A1	A2	B1	B2	Q	Q̄
L	X	X	X	X	L	H
X	H	H	X	X	L†	H†
X	X	X	L	X	L†	H†
X	X	X	X	L	L†	H†
H	L	X	↑	H	⊓	⊔
H	L	X	H	↑	⊓	⊔
H	X	L	↑	H	⊓	⊔
H	X	L	H	↑	⊓	⊔
H	H	↓	H	H	⊓	⊔
H	↓	↓	H	H	⊓	⊔
H	↓	H	H	H	⊓	⊔
↑	L	X	H	H	⊓	⊔
↑	X	L	H	H	⊓	⊔

'123, '130, 'LS123
FUNCTION TABLE

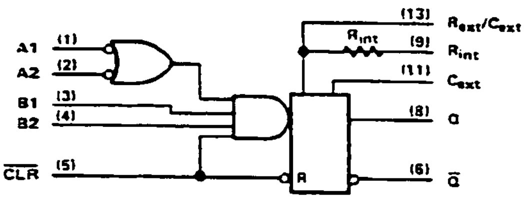
INPUTS			OUTPUTS	
CLEAR	A	B	Q	Q̄
L	X	X	L	H
X	H	X	L†	H†
X	X	L	L†	H†
H	L	↑	⊓	⊔
H	↓	H	⊓	⊔
↑	L	H	⊓	⊔

See explanation of function tables on page

† These lines of the functional tables assume that the indicated steady-state conditons at the A and B inouts have been set up long enough to complete any pulse started before the set up.

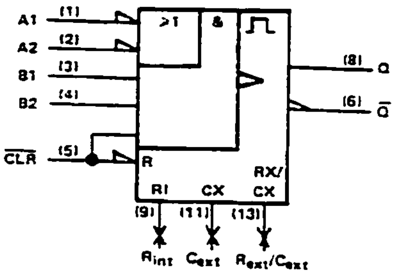
logic diagram (positive logic)

'122, 'LS122



logic symbol†

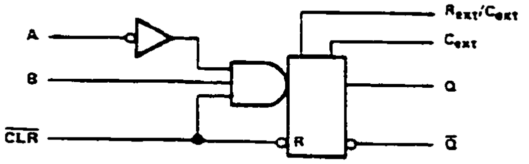
'122, 'LS122



R_{int} is nominally 10 kΩ for '122 and 'LS122

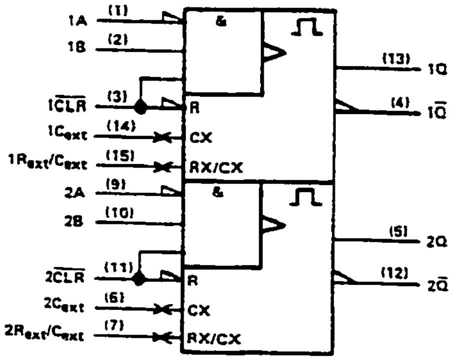
logic diagram (positive logic) (each multivibrator)

'123, '130, 'LS123



logic symbol†

'123, '130, 'LS123

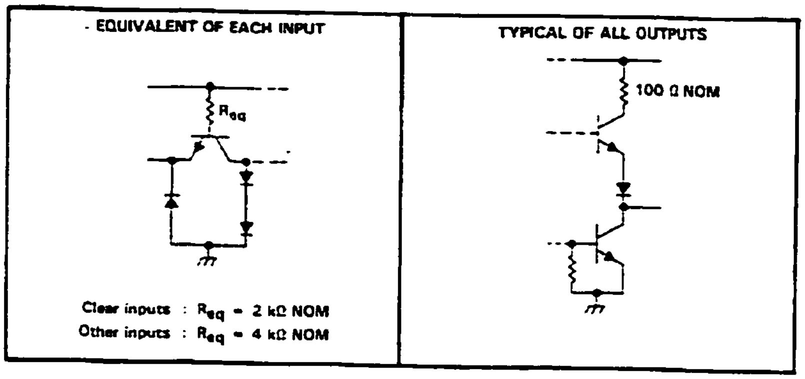


Pin numbers shown are for D, J, N, and W packages.

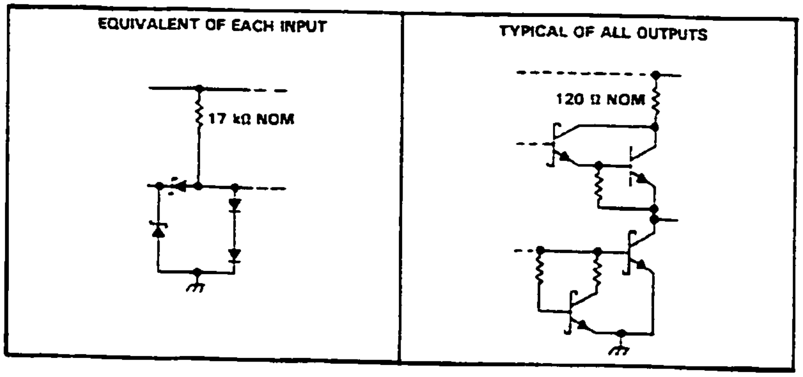
†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

schematics of inputs and outputs

'122, '123, '130 CIRCUITS


EQUIVALENT OF EACH INPUT
R_{eq}
Clear inputs : R_{eq} = 2 kΩ NOM
Other inputs : R_{eq} = 4 kΩ NOM
TYPICAL OF ALL OUTPUTS
100 Ω NOM

'LS122, 'LS123 CIRCUITS



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: '122, '123, '130	5.5 V
'LS122, 'LS123	7 V
Operating free-air temperature range: SN54'	−55°C to 125°C
SN74'	0°C to 70°C
Storage temperature range	−65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

TYPICAL APPLICATION DATA FOR '122, '123, '130

For pulse durations when $C_{ext} \leq 1000$ pF, see Figure 4.

The output pulse duration is primarily a function of the external capacitor and resistor. For $C_{ext} > 1000$ pF, the output pulse duration (t_W) is defined as:

$$t_W = K \cdot R_T \cdot C_{ext} \left(1 + \frac{0.7}{R_T}\right)$$

where

K is 0.32 for '122, 0.28 for '123 and '130

R_T is in kΩ (internal or external timing resistance.)

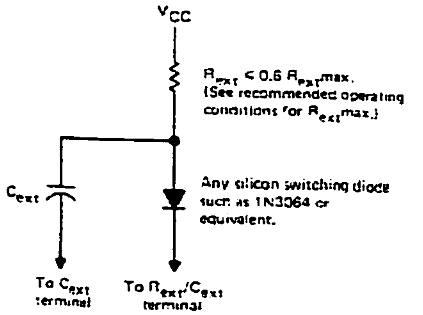
C_{ext} is in pF

t_W is in ns

To prevent reverse voltage across C_{ext}, it is recommended that the method shown in Figure 2 be employed when using electrolytic capacitors and in applications utilizing the clear function. In all applications using the diode, the pulse duration is:

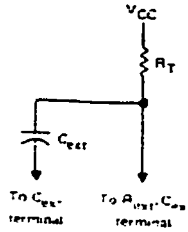
$$t_W = K_D \cdot R_T \cdot C_{ext} \left(1 + \frac{0.7}{R_T}\right)$$

K_D is 0.28 for '122, 0.25 for '123 and '130


V_{CC}
$R_{ext} \leq 0.6\ R_{ext}max$.
(See recommended operating conditions for $R_{ext}max$.)
C_{ext}
Any silicon switching diode such as 1N3064 or equivalent.
To C_{ext} terminal
To R_{ext}/C_{ext} terminal

TIMING COMPONENT CONNECTIONS WHEN $C_{ext} > 1000$ pF AND CLEAR IS USED
FIGURE 2

Applications requiring more precise pulse durations (up to 28 seconds) and not requiring the clear feature can best be satisfied with the '121.



TIMING COMPONENT CONNECTIONS
FIGURE 3

TYPICAL OUTPUT PULSE DURATION
vs
EXTERNAL TIMING CAPACITANCE

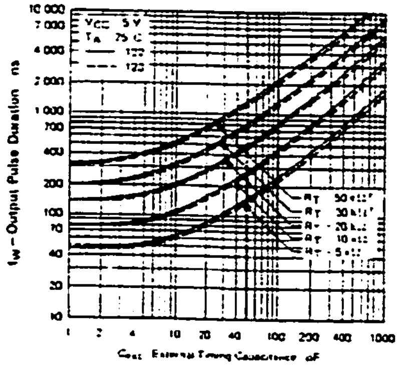


FIGURE 4

†These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54' circuits.

TYPICAL APPLICATION DATA FOR 'LS122, 'LS123

The basic output pulse duration is essentially determined by the values of external capacitance and timing resistance. For pulse durations when $C_{ext} \leq 1000$ pF, use Figure 6, or use Figure 7 where the pulse duration may be defined as:

$$t_w = K \cdot R_T \cdot C_{ext}$$

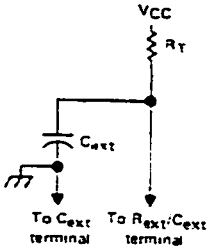
When $C_{ext} \geq 1$ μF, the output pulse width is defined as:

$$t_w = 0.33 \cdot R_T \cdot C_{ext}$$

For the above two equations, as applicable:

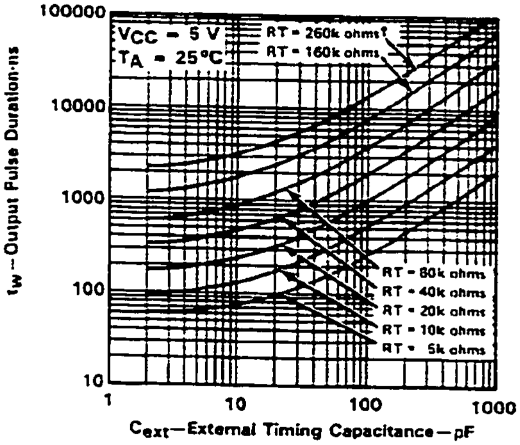
K is multiplier factor, see Figure 7
R_T is in kΩ (internal or external timing resistance)
C_{ext} is in pF
t_w is in ns

For maximum noise immunity, system ground should be applied to the C_{ext} node, even though the C_{ext} node is already tied to the ground lead internally. Due to the timing scheme used by the 'LS122 and 'LS123, a switching diode is not required to prevent reverse biasing when using electolytic capacitors.



TIMING COMPONENT CONNECTIONS
FIGURE 5

'LS122, 'LS123
TYPICAL OUTPUT PULSE DURATION
vs
EXTERNAL TIMING CAPACITANCE

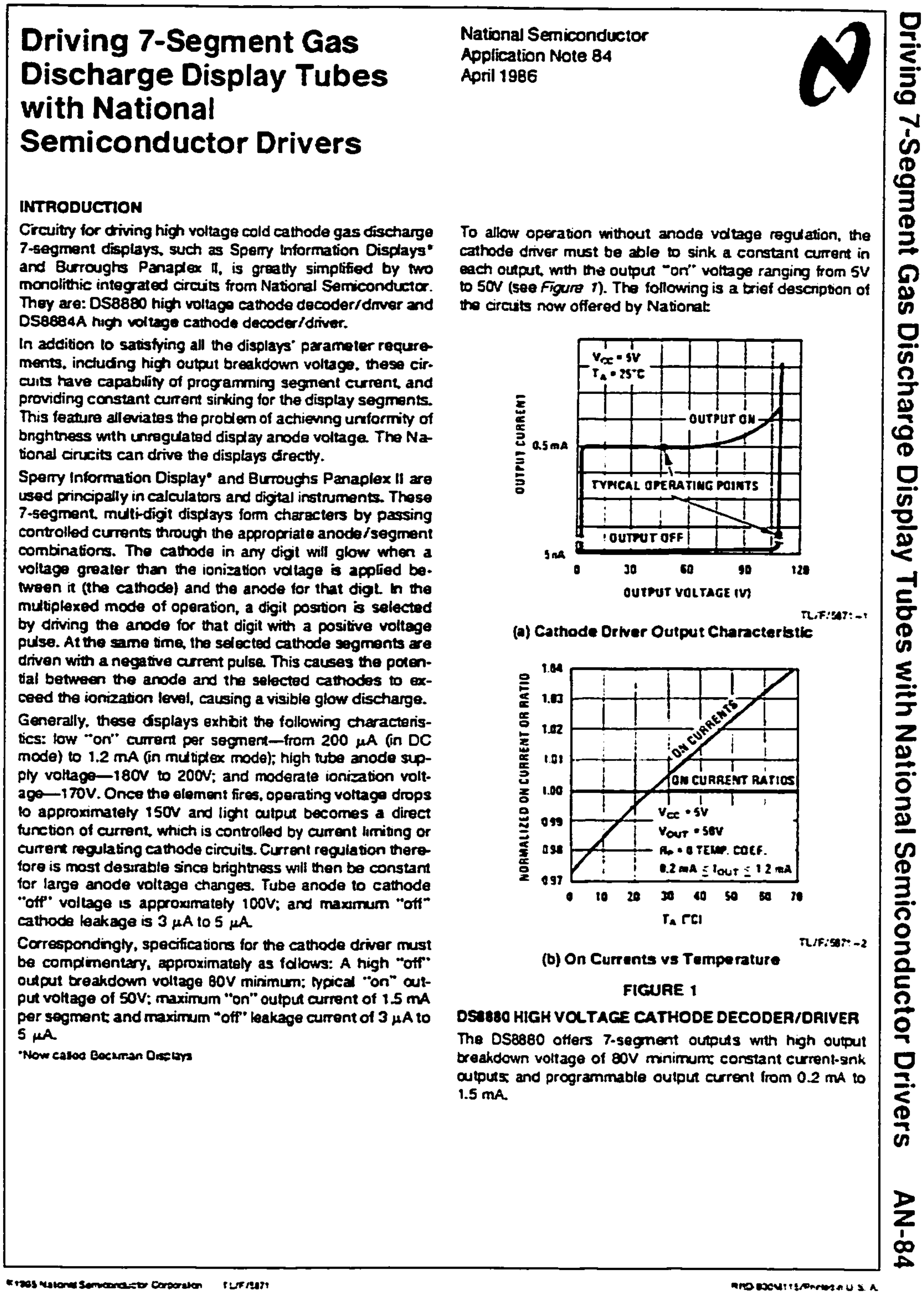


†This value of resistance exceeds the maximum recommended for use over the full temperature range of the SN54LS circuits.

FIGURE 6

APPENDIX 9

SELECTED DS8884 CATHODE DRIVER SPECIFICATIONS



Driving 7-Segment Gas Discharge Display Tubes with National Semiconductor Drivers

National Semiconductor
Application Note 84
April 1986

INTRODUCTION

Circuitry for driving high voltage cold cathode gas discharge 7-segment displays, such as Sperry Information Displays* and Burroughs Panaplex II, is greatly simplified by two monolithic integrated circuits from National Semiconductor. They are: DS8880 high voltage cathode decoder/driver and DS8884A high voltage cathode decoder/driver.

In addition to satisfying all the displays' parameter requrements, including high output breakdown voltage, these circuits have capability of programming segment current, and providing constant current sinking for the display segments. This feature alleviates the problem of achieving uniformity of brghtness with unregulated display anode voltage. The National cirucits can drive the displays drectly.

Sperry Information Display* and Burroughs Panaplex II are used principally in calculators and digital instruments. These 7-segment, multi-digit displays form characters by passing controlled currents through the appropriate anode/segment combinations. The cathode in any digit will glow when a voltage greater than the ionization voltage is applied between it (the cathode) and the anode for that digit. In the multiplexed mode of operation, a digit position is selected by driving the anode for that digit with a positive voltage pulse. At the same time, the selected cathode segments are driven with a negative current pulse. This causes the potential between the anode and the selected cathodes to exceed the ionization level, causing a visible glow discharge.

Generally, these displays exhibit the following characteristics: low "on" current per segment—from 200 μA (in DC mode) to 1.2 mA (in multiplex mode); high tube anode supply voltage—180V to 200V; and moderate ionization voltage—170V. Once the element fires, operating voltage drops to approximately 150V and light output becomes a direct function of current, which is controlled by current limiting or current regulating cathode circuits. Current regulation therefore is most desirable since brightness will then be constant for large anode voltage changes. Tube anode to cathode "off" voltage is approximately 100V; and maximum "off" cathode leakage is 3 μA to 5 μA.

Correspondingly, specifications for the cathode driver must be complimentary, approximately as follows: A high "off" output breakdown voltage 80V minimum; typical "on" output voltage of 50V; maximum "on" output current of 1.5 mA per segment; and maximum "off" leakage current of 3 μA to 5 μA.

To allow operation without anode voltage regulation, the cathode driver must be able to sink a constant current in each output, with the output "on" voltage ranging from 5V to 50V (see *Figure 1*). The following is a brief description of the circuits now offered by National:

TL/F/5871-1

(a) Cathode Driver Output Characteristic

TL/F/5871-2

(b) On Currents vs Temperature

FIGURE 1

DS8880 HIGH VOLTAGE CATHODE DECODER/DRIVER

The DS8880 offers 7-segment outputs with high output breakdown voltage of 80V minimum; constant current-sink outputs; and programmable output current from 0.2 mA to 1.5 mA.

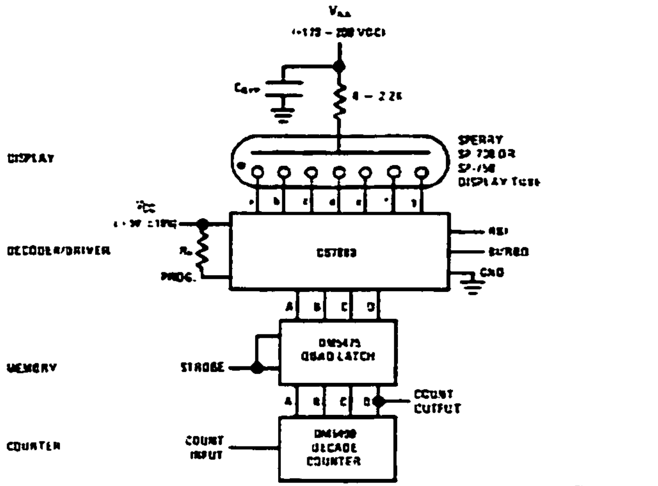
*Now called Beckman Displays

RRD-B30M115/Printed in U. S. A.

Driving 7-Segment Gas Discharge Display Tubes with National Semiconductor Drivers AN-84

APPLICATION

The circuit has a built-in BCD decoder and can interface directly to Sperry and Panaplex II displays, minimizing external components *(Figure 2)*. The inputs can be driven by TTL or MOS outputs directly. It is optimized for use in systems with 5V supplies.

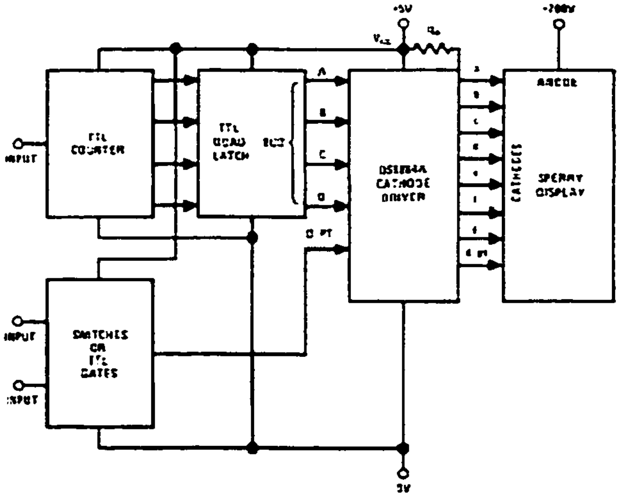


TL/F/5871–3

FIGURE 2. DC Operation From TTL

The DS8880 decoder/driver provides for unconditional as well as leading and trailing zero blanking. It utilizes negative input voltage clamp diodes. Typically, output current varies only 1% for output voltage changes of 3V to 50V. Operating power supply voltage is 5V. The device can be used for multiplexed or DC operation.

Available in 16-pin cavity DIP packages, the DS7880 is guaranteed over the full military operating temperature range of −55°C to +125°C; the DS8880 in molded DIP over the industrial range of 0°C to +70°C.



TL/F/5871–4

FIGURE 3. Interfacing Directly With TTL Output

DS8884A HIGH VOLTAGE CATHODE DECODER/DRIVER

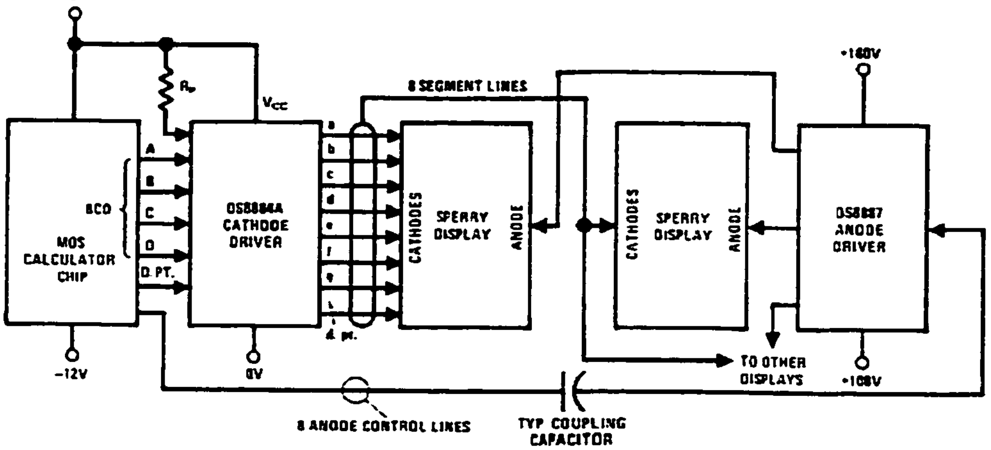
The DS8884A offers 9-segment outputs with high output breakdown voltage of 80V minimum: constant current-sink outputs, programmable from 0.2 mA to 1.2 mA. It also offers input negative and positive voltage clamp diodes for DC restoring, and low input load current of −0.25 mA maximum.

APPLICATION

DS8884A decodes four lines of BCD input and drives 7-segment digits of gas-filled displays. There are two separate inputs and two additional outputs for direct control of decimal point and comma cathodes. The inputs can be DC coupled to TTL *(Figure 3)* or MOS outputs *(Figure 4)*, or AC-coupled to TTL or MOS outputs *(Figure 5)* using only a capacitor. This means the device is useful in applications where level shifting is required. It can be used in multiplexed operation, and is available in an 18-pin molded DIP package.

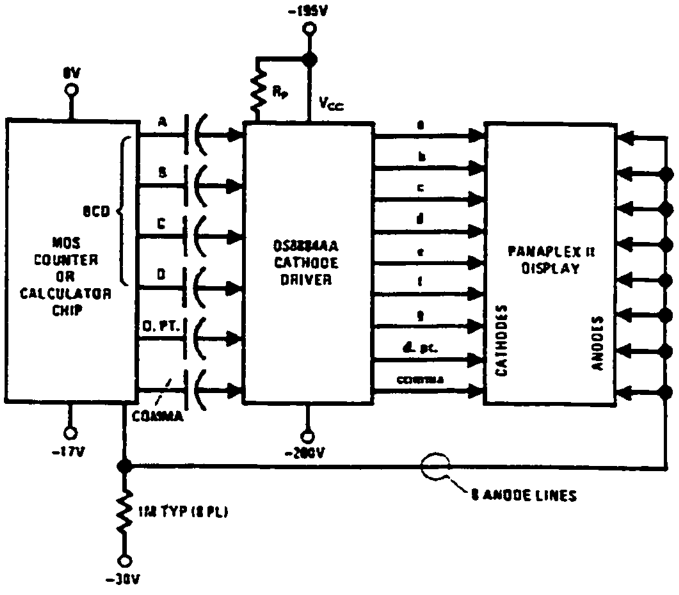
Other advantages of the DS8884A are: typical output current variation of 1% for output voltage changes of 3V to 50V; and operating power supply voltage of 5V. Inputs have pull-up resistors to increase noise immunity in AC coupled applications.

The DS8884A is guaranteed over the 0°C to +70°C operating temperature range.



TL/F/5871-5

FIGURE 4. BCD Data Interfacing Directly With MOS Output



TL/F/5871-6

Note: Capacitive coupling between the logic and the segment drivers may be used only when the segment drivers are turned "OFF" during digit-to-digit transitions.

FIGURE 5. Cathode BCD Data AC Coupled From MOS-Output

APPENDIX 10

RNAV PINOUTS

RNAV BUS CONNECTOR

		1.	64	Audio GND (LO)	SWITCH/39
		2.	63		
		3.	62		
SWITCH/40	Audio OUT	4.	61	Audio IN	SWITCH/38
		5.	60		
		6.	59		
		7.	58		
		8.	57		
		9.	56		
		10.	55		
		11.	54		
		12.	53		
8279/12	D0	13.	52		
8279/13	D1	14.	51	-RD	8279/10
8279/14	D2	15.	50	-WR	8279/11
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9
8279/18	D6	19.	46		
8279/19	D7	20.	45	CLEAR	74LS74/1
8279/22 (T)	CS4	21.	44	High Freq.	74LS74/11
8279/22 (B)	CS5	22.	43		
		23.	42		
		24.	41		
		25.	40	LTNG HI	
		26.	39		
	5V	27.	38		
	+9V	28.	37		
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
		32.	33	LTNG LO	

Audio OUT: IDT

RNAV SWITCH BOARD CONNECTOR

BUS/4	Audio OUT	40	1.	GND	
BUS/64	Audio GND	39	2.	Audio OUT	BUS/4
BUS/61	Audio IN	38	3.	WPT WP	74LS07/5
8279/1 (B)	Pull-Up SW	37	4.	LL	74LS173/7 (T)
74LS139/11	Pull-Up SW COMMON	36	5.	P1L	74LS74/4
		35	6.	P2L	74LS74/4
8279/38 (B)	ON SW	34	7.	RL	74LS173/7 (B)
74LS139/11	ON SW COMMON	33	8.	MDE WP	74LS07/9
74LS07/1	FRQ WP (KHz)	32	9.	COMMON SW (ROW)	74LS139/12
74LS173/7 (T)	LR	31	10.	RAD SW COMMON	74LS139/11
74LS74/4	P1R	30	11.	DATA SW	8279/5 (B)
74LS74/4	P2R	29	12.	RAD SW	?
74LS173/7 (B)	RR	28	13.	USE SW	8279/6 (B)
74LS07/3	FRQ WP (MHz)	27	14.	RTN SW	8279/7 (B)
		26	15.	CHK SW	8279/8 (B)
		25	16.	RAD SW (IN)	8279/39 (B)
		24	17.		
		23	18.		
	GND	22	19.	GND	
	LTNG HI	21	20.	LTNG LO	

RNAV DISPLAY BOARD CONNECTOR

8279/31 (T)	R1	24	1.	SCAN (MPX CLK)	8279/23 (T)
8279/30 (T)	R2	23	2.	SYNC (RESET)	74LS123/13
8279/29 (T)	R4	22	3.	DM1	8279/31 (B)
8279/28 (T)	R8	21	4.	DM2	8279/30 (B)
8279/27 (T)	L1	20	5.	PROG	Dimming
8279/26 (T)	L2	19	6.	PHOTO	Dimming
8279/25 (T)	L4	18	7.		
8279/24 (T)	L8	17	8.		
	GND	16	9.	+9V REG	
	+5V REG	15	10.	GND	
	GND	14	11.	H.V.	
		13	12.		

APPENDIX 11

NAV/COMM PINOUTS

NAV/COMM BUS CONNECTOR

		1.	64	Audio GND	SWITCH/1
		2.	63		
SWITCH/40	Test Vol. Hi	3.	62	Test Vol. Center	SWITCH/39
		4.	61		
SWITCH/5	IDT Vol. Hi	5.	60	IDT Vol. Center	SWITCH/37
		6.	59		
		7.	58		
SWITCH/3	Test SW	8.	57	Test SW Common	SWITCH/2
		9.	56		
74LS374/11	CS2	10.	55		
8279/22	CS3	11.	54		
		12.	53		
8279/12	D0	13.	52		
8279/13	D1	14.	51	-RD	8279/10
8279/14	D2	15.	50	-WR	8279/11
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9
8279/18	D6	19.	46		
8279/19	D7	20.	45		
		21.	44		
		22.	43		
		23.	42		
		24.	41		
		25.	40		
		26.	39		
	5V	27.	38		
	9V	28.	37		
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
		32.	33		

IDT Vol. HI: IDT SW, only when IDT is switched on, the volume control is available.

NAV/COMM SWITCH BOARD CONNECTOR

BUS/64	Audio GND	1.	40	Test Vol. HI	BUS/3
BUS/57	Test SW Common	2.	39	Test Vol. Center	BUS/62
BUS/8	Test SW	3.	38		
		4.	37	IDT Vol. Center	BUS/60
BUS/5	IDT Vol. HI	5.	36		
		6.	35	L SW	8279/8
		7.	34	L SW Common	R-
556/4,6,8	CLK (-BD)	8.	33	R SW	8279/7
8884/2 (L)	OUTB0 A	9.	32	R SW Common	R-
8884/3 (L)	OUTB1 B	10.	31		
8884/4 (L)	OUTB2 C	11.	30	OUTA0 A	8884/2 (R)
8884/5 (L)	OUTB3 D	12.	29	OUTA1 B	8884/3 (R)
8884/6 (L) 74LS02/4	DP	13.	28	OUTA2 C	8884/4 (R)
74LS05/4	SCAN	14.	27	OUTA3 D	8884/5 (R)
74LS05/6	SYNC	15.	26	DP	8884/6 (R) 74LS02/1
		16.	25		
		17.	24		
	+5V	18.	23	+9V	
	GND	19.	22	GND	
	GND	20.	21	H.V. +185V	

NAV/COMM DISPLAY BOARD CONNECTOR

	GND	1.	16	ON SW	8279/2
74LS139/4	SW GND	2.	15	ON SW Common	R-
74LS139/5	SW GND	3.	14	C 25K SW	8279/5
74LS139/6	SW GND	4.	13	C 25K SW Common	R-
74LS139/7	SW GND	5.	12	GND	
8279/1	LEFT SW	6.	11	N RAD SW	8279/6
8279/39	PULSE SW	7.	10	N RAD SW Common	R-
8279/38	RIGHT SW	8.	9		

APPENDIX 12
COMM PINOUTS

COMM BUS CONNECTOR

		1.	64	Audio GND	SWITCH/1
SWITCH/39	POT HI	2.	63	POT LO	SWITCH/40
		3.	62		
		4.	61		
		5.	60		
		6.	59		
SWITCH/38	Test SW Common	7.	58	Test SW	SWITCH/2
		8.	57		
		9.	56		
		10.	55		
		11.	54		
		12.	53		
8279/12	D0	13.	52	CS6	8279/22
8279/13	D1	14.	51	-RD	8279/10
8279/14	D2	15.	50	-WR	8279/11
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9
8279/18	D6	19.	46		
8279/19	D7	20.	45	CLEAR	74LS74/1
		21.	44		
		22.	43		
		23.	42		
		24.	41		
		25.	40		
		26.	39		
	5V	27.	38		
	9V	28.	37		
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
		32.	33		

COMM SWITCH/DISPLAY BOARD CONNECTOR

BUS/64	Audio GND	1.	40	POT LO	BUS/63
BUS/58	Test SW	2.	39	POT HI	BUS/2
	GND	3.	38	Test SW Common	BUS/7
7407/1 74LS139/11	Common K	4.	37	Common M	7407/3 74LS139/12
74LS173/7 (L) R-	DEC SW	5.	36	INC SW	74LS173/7 (R) R-
		6.	35	PULSE SW	74LS74/4 R-
		7.	34		
74LS139/4	XFER SW Common	8.	33	ON SW Common	74LS139/4
8279/5	XFER SW	9.	32	ON SW	8279/6
8884/2	OUTB0 A	10.	31		
8884/3	OUTB1 B	11.	30		
8884/4	OUTB2 C	12.	29	PULL 25K SW	8279/7
8884/5	OUTB3 D	13.	28	PULL 25K SW GND	74LS139/5
8884/6	OUTA3 DP	14.	27	SCAN	7407/8 R-
		15.	26	SYNC	7407/6 R-
		16.	25		
		17.	24		
	+5V	18.	23	+9V	
	GND	19.	22	GND	
	GND	20.	21	H.V. +185V	

APPENDIX 13

ATC XPNDR PINOUTS

ATC XPNDR BUS CONNECTOR

		1.	64		
		2.	63		
		3.	62		
		4.	61		
		5.	60		
		6.	59		
		7.	58		
		8.	57		
		9.	56		
		10.	55		
		11.	54	CS0	8279/22
		12.	53		
8279/12	D0	13.	52		
8279/13	D1	14.	51	-RD	8279/10
8279/14	D2	15.	50	-WR	8279/11
8279/15	D3	16.	49	CLK	8279/3
8279/16	D4	17.	48	A0	8279/21
8279/17	D5	18.	47	RESET	8279/9
8279/18	D6	19.	46		
8279/19	D7	20.	45		
		21.	44		
		22.	43		
		23.	42		
		24.	41		
		25.	40		
		26.	39		
	5V	27.	38	LTNG HI	SWTICH/17
	9V	28.	37		
	GND	29.	36	GND	
	GND	30.	35	H.V. +185V	
		31.	34		
		32.	33	LTNG LO	SWITCH/24

ATC XPNDR SWITCH/DISPLAY BOARD CONNECTOR

	GND	1.	40	Mode1 SW	8279/2
74LS138/15	V SW Common	2.	39	Mode2 SW	8279/5
8279/8	V SW	3.	38	Mode3 SW	8279/6
74LS138/13	VFR SW Common	4.	37	MODE SW Common	74LS138/15
8279/8	VFR SW	5.	36	INC/DEC 2	8279/1
74LS138/13	IDT SW Common	6.	35	INC/DEC 1	8279/39
8279/7	IDT SW	7.	34	INC/DEC 0	8279/38
8884/3	OUTA1 B	8.	33	INC/DEC Common	74LS138/14
8884/2	OUTA0 A	9.	32		
8884/4	OUTA2 C	10.	31		
8884/5	OUTA3 D	11.	30		
8889/2	OUTB0	12.	29	PHOTO	Dimming
8889/3	OUTB1	13.	28	SYNC	74LS05/8 R-
8889/4,5	OUTB2	14.	27	SCAN	74LS05/10 R-
8889/6,7	OUTB3	15.	26	PROG	Dimming
		16.	25		
BUS/38	LTNG HI	17.	24	LTNG LO	BUS/33
	+5V	18.	23	+9V	
	GND	19.	22	GND	
	GND	20.	21	H.V. +185V	

APPENDIX 14

SELECTED DM8889N CATHODE DRIVER SPECIFICATIONS

DIONICS, INC.
65 Rushmore Street
Westbury, NY 11590

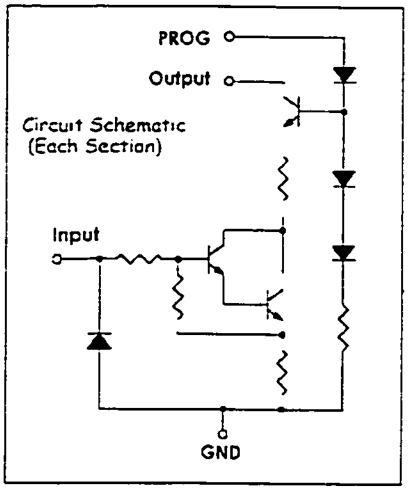
Phone: (516) 997-7474
Fax: (516) 997-7479
Website: www.dionics-usa.com

GAS DISCHARGE DISPLAY SEGMENT DRIVERS

DI-230A DI-240A

General Description:

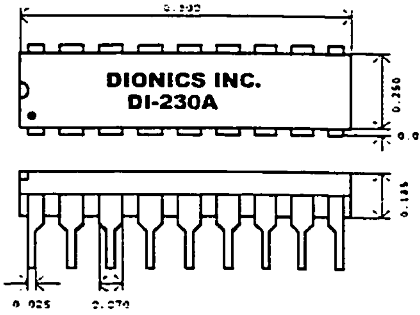
The DIONICS DI-230A · DI-240A Series circuits are designed to drive gas discharge display devices from signals originating from MOS or TTL circuitry. Each output is a switched. programmable constant current sink with a voltage compliance of 80 or 125 Volts.
These circuits provide for simple interfaces with displays such as the Beckman. Burroughs Panaplex [7], Cherry or equivalents.



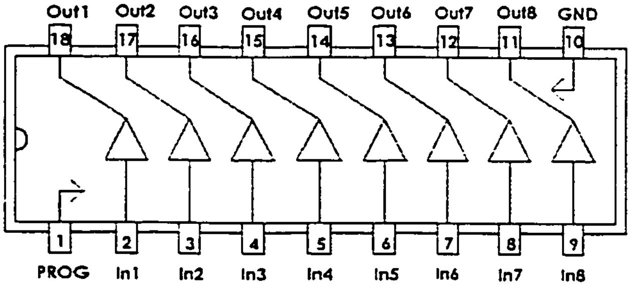
Features:

- ✓ High Breakdown Voltage: 80V or 125V.
- ✓ High Input Voltage Capability: 40V
- ✓ TTL or MOS Compatible
- ✓ All Output Currents Programmed with Single Resistor
- ✓ Requires Few Additional Components
- ✓ Equivalent To DM-8889, MC3491, ITT-505
- ✓ DC Restoring Input Diode

Package Layout:



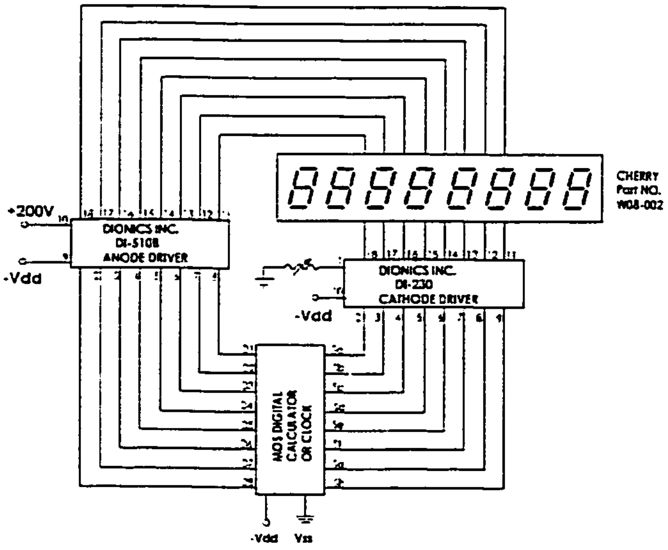
Pin Connections

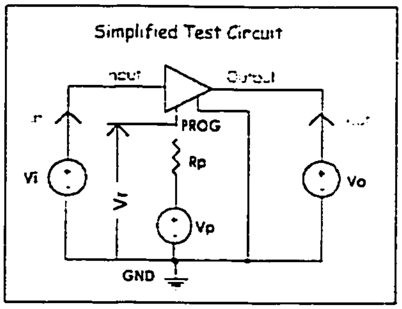


Absolute Maximum Rating (Ta = 25 °C)				
Characteristic	**Symbol**	**Notes**	**Limits**	**Units**
Input Voltage	V_{in}	Measured With Respect to GND Terminal	40	V
Output Current	I_o		5	mA
Output Voltage DI-230	V_o	Measured With Respect to GND Terminal	80	V
Output Voltage DI-240	V_o	Measured With Respect to GND Terminal	100	V
Power Dissipation DI-230; DI-240	P_D	Derate at 8 mW/ °C Above 25°C Ambient	800	mW
Storage Temperature	T_s		-55 to +125	°C
Operating Temperature*	T_{o}		0 to +70	°C

Electrical Characteristics (Ta = 25 °C)						
Parameter	**Symbol**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Units**
Output Saturation Voltage	V_o (SAT)	I_o=1mA; V_i=3.5V; R_p=27kΩ; V_p=10V		5		V
Output Leakage Current	I_o (OFF)	V_o = Rated Voltage; V_i = 0.4V; R_p = 27kΩ; V_p = 10V		0.1	10	µA
Output Current Match	$\Delta I_o / I_o$	V_o=60V; V_i=3.5V; R_p=27kΩ; V_p=10V		± 5	± 10	%
Output Current	I_o (ON)	V_o=60V; V_i=3.5V; R_p=27kΩ; V_p=10V	0.85	1.00	1.15	mA
Input Current	I_i	V_i = 7.0V	250	370	500	µA

Typical Application:





APPENDIX 15

KERNEL PARALLEL PORT DRIVER

```
/*Kernel Parallel Port Driver*/

/*Port instructions are not included in development environment VC++ because direct I/O access isn't
allowed by the operation system (Windows NT, 2000, XP). Therefore, it is needed to include a portion of
assembler code into the software in order to access the hardware. */


/*Assembler Code*/

/*BYTE Input*/

BYTE inportb (UINT portid)

{
        unsigned char value;

        _asm mov edx, portid
        _asm in al, dx
        _asm mov value, al

        return value;
}


/*BYTE Output*/

void outportb(UINT portid, BYTE value)

{
        _asm mov edx, portid
        _asm mov al, value
        _asm out dx, al
}


/*Ex: Print character "C"*/

if (inportb(0x379) & 0x10)                        /*Check Select Pin*/

{
        outportb(0x378, 'B');                     /*Write character "C" to Parallel Port*/
        outportb(0x37a, inportb(0x37a) | 0x01);   /*Set "Strobe"*/

        sleep(1);                                 /*Wait 1ms*/

        outportb(0x37a, inportb(0x37a) | 0xfe);   /*Clear "Strobe"*/
}
```

APPENDIX 16

VC++ RADIO STACK MODULE

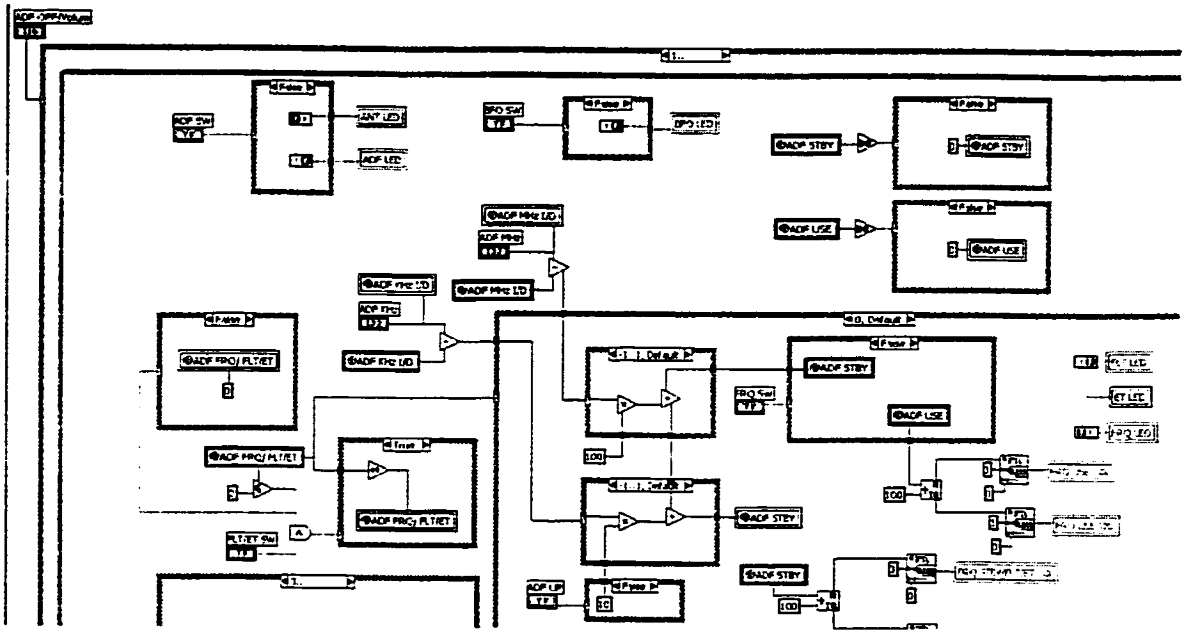
```
/*PCI-6601 I/O and Frequency Generation Functions*/

/*include files*/

#include "nidaqex.h"

/*Globe Variable Declarations*/
i16 iStatus = 0;
i16 iRetVal = 0;
i16 iDevice = 1;
i16 iIgnoreWarning = 0;
i16 iPort = 0;

/*Subroutine Declarations*/

void Clock(void);
void StopClock(void);
void InputPort(void);
void OutputPort(long iPattern);
void OutputLine(int iLine, int iState);


/*3 MHz Clock Generation*/

void Clock(void)
{
  /*Local Variable Declarations*/

  u32 ulGpctrNum = ND_COUNTER_0;
  u32 ulGpctrOutput = ND_GPCTR0_OUTPUT;
  u32 ulLOWcount = 7;
  u32 ulHIGHcount = 7;


  iStatus = GPCTR_Control(iDevice, ulGpctrNum, ND_RESET);

  iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Control/RESET", iIgnoreWarning);

  iStatus = GPCTR_Set_Application(iDevice, ulGpctrNum, ND_PULSE_TRAIN_GNR);

  iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Set_Application", iIgnoreWarning);

  iStatus = GPCTR_Change_Parameter(iDevice, ulGpctrNum, ND_SOURCE,
              ND_INTERNAL_20_MHZ);

  iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Change_Parameter/SOURCE", iIgnoreWarning);

  iStatus = GPCTR_Change_Parameter(iDevice, ulGpctrNum, ND_COUNT_1, ulLOWcount);

  iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Change_Parameter/COUNT1", iIgnoreWarning);
```

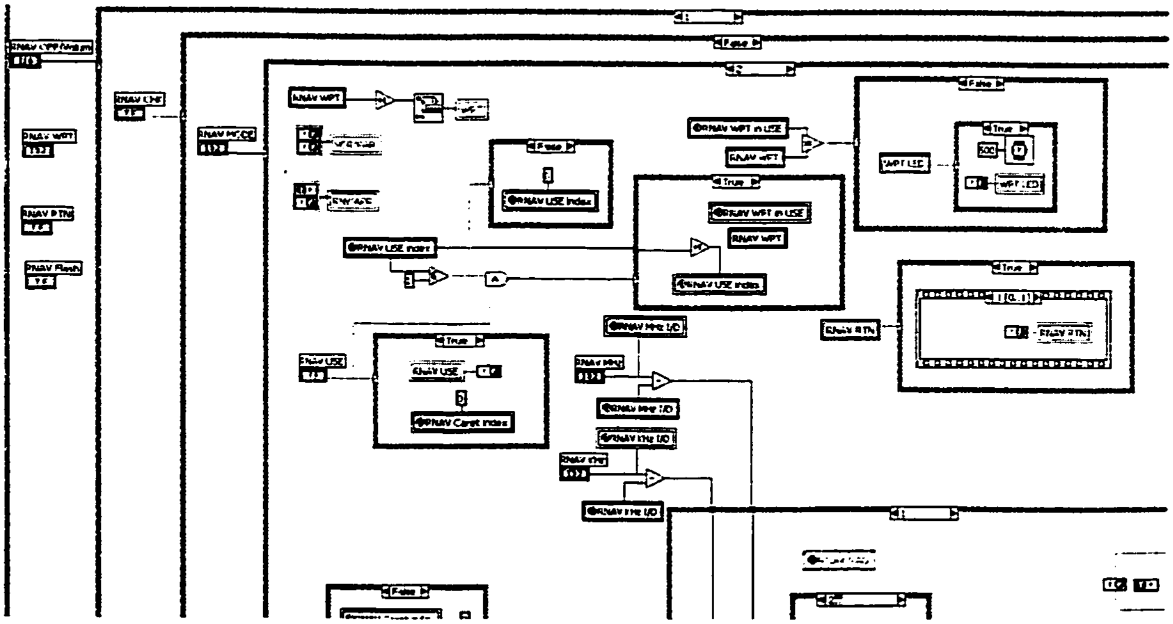
```
    iStatus = GPCTR_Change_Parameter(iDevice, ulGpctrNum, ND_COUNT_2, ulHIGHcount);

    iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Change_Parameter/COUNT2", iIgnoreWarning);

    /* To output a counter pulse, you must call Select_Signal. */

    iStatus = Select_Signal(iDevice, ulGpctrOutput, ulGpctrOutput, ND_LOW_TO_HIGH);

    iRetVal = NIDAQErrorHandler(iStatus, "Select_Signal/GpctrOutput", iIgnoreWarning);

    iStatus = GPCTR_Control(iDevice, ulGpctrNum, ND_PROGRAM);

    iRetVal = NIDAQErrorHandler(iStatus, "GPCTR_Control/PROGRAM", iIgnoreWarning);

    printf(" Light Aircraft Radio Stack Simulation Software started...\n");

    /* HINT: If you don't see pulses at GPCTR0_OUTPUT, check your connections. */

}

/* End of 3 MHz Clock Generation program */


/*StopClock*/

void StopClock(void)
{
        /*Local Variable Declarations*/

        u32 ulGpctrNum = ND_COUNTER_0;


   printf(" Hit any key to stop simulation.\n");

   iRetVal = NIDAQWaitForKey(0.0);

   /* CLEANUP - Don't check for errors on purpose. */

   /* Reset GPCTR. */

   iStatus = GPCTR_Control(iDevice, ulGpctrNum, ND_RESET);

   /* Disconnect GPCTR0_OUTPUT. */

   /* Note that the following Select_Signal call will cause the output to be high impedance which will
         most likely bring the logic level HIGH if there is a pull-up resistor on this pin. (Check your
    hardware user manual.) If you do not want this behavior, comment out the next line. */

   iStatus = Select_Signal(iDevice, ND_GPCTR0_OUTPUT, ND_NONE, ND_DONT_CARE);

   printf(" Light Aircraft Radio Stack Simulation Software done!\n");
}

/* End of StopClock program */
```

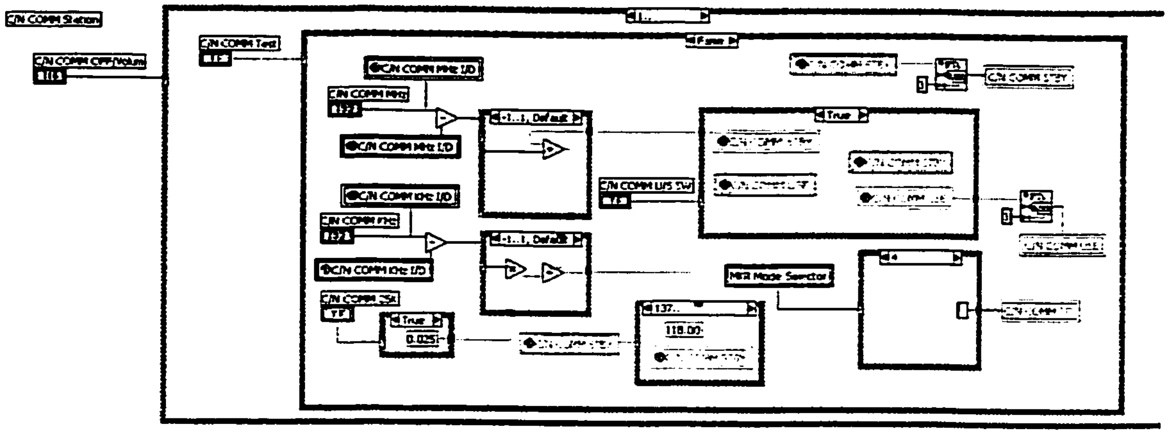
```
/*OutputLine*/

void OutputLine(int iLine, int iState)
{

  /*Local Variable Declarations*/

  i16 iDir = 1;

  /* Configure line as output. */

  /* NOTE: Some devices do not support DIG_Line_Config. Use DIG_Prt_Config instead. */

  iStatus = DIG_Line_Config(iDevice, iPort, iLine, iDir);

  iRetVal = NIDAQErrorHandler(iStatus, "DIG_Line_Config". iIgnoreWarning);

  printf(" The digital state on port %d line %d is set to %d\n", iPort, iLine, iState);

  iStatus = DIG_Out_Line(iDevice. iPort, iLine, iState);

  iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Line", iIgnoreWarning);


}

/* End of OutputLine program */


/*InputPort*/

void InputPort(void)
{
  /*Local Variable Declarations*/

  i16 iMode = 0;
  i16 iDir = 0;
  i32 iPattern = 0;

  /* Configure port as input, no handshaking. */

  iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);

  iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);

  iStatus = DIG_In_Prt(iDevice, iPort, &iPattern);

  iRetVal = NIDAQErrorHandler(iStatus, "DIG_In_Prt", iIgnoreWarning);

  printf(" The digital pattern on port %d is (DECIMAL) %ld\n". iPort, iPattern);

}

/* End of InputPort program */
```

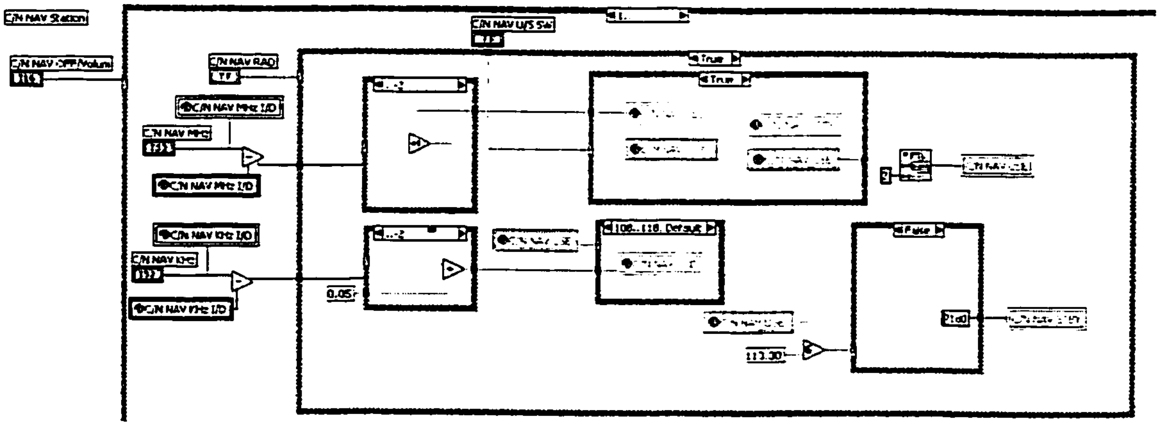
```
/*OutputPort*/

void OutputPort(long iPattern)
{
   /*Local Variable Declarations*/

   i16 iMode = 0;
   i16 iDir = 1;
   i16 iYieldON = 1;

   /* Configure port as output, no handshaking. */

   iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);

   iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);

   iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);

   iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", iIgnoreWarning);

   printf(" The digital pattern on port %d is set to (DECIMAL) %ld\n", iPort, iPattern);

          iRetVal = NIDAQYield(iYieldON);

}

/* End of OutputPort program */
```

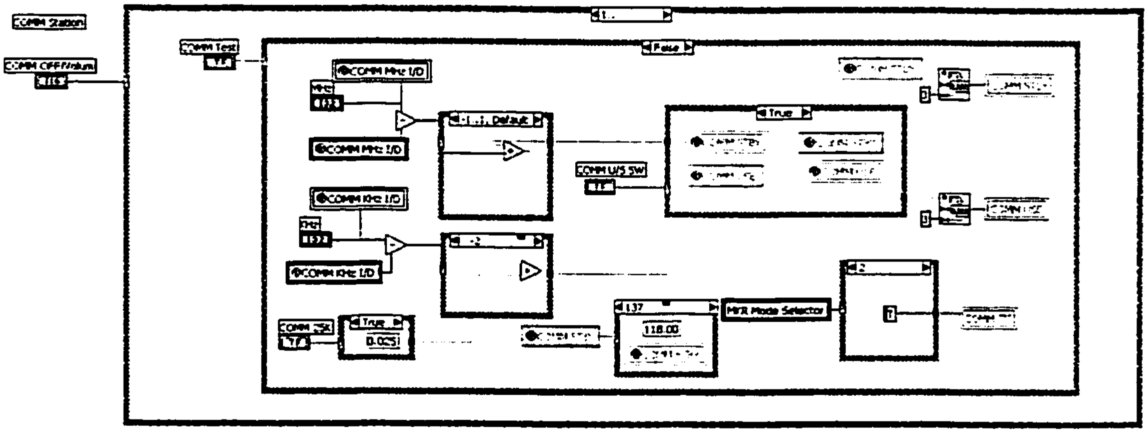
APPENDIX 17
LABVIEW RADIO STACK MODULE

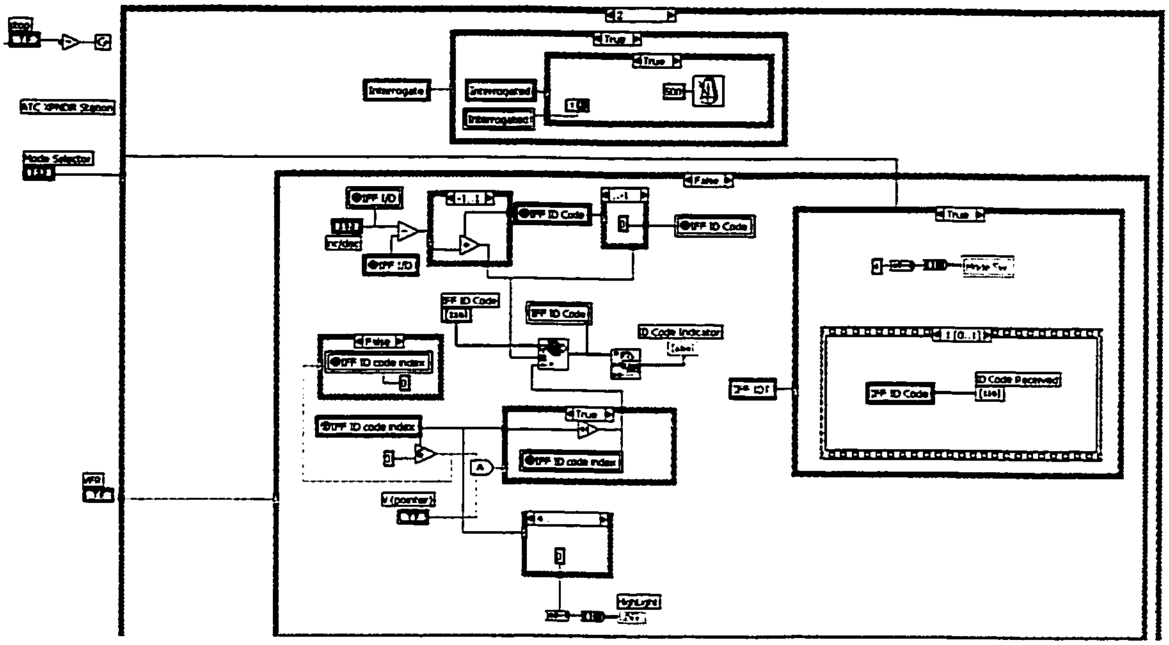




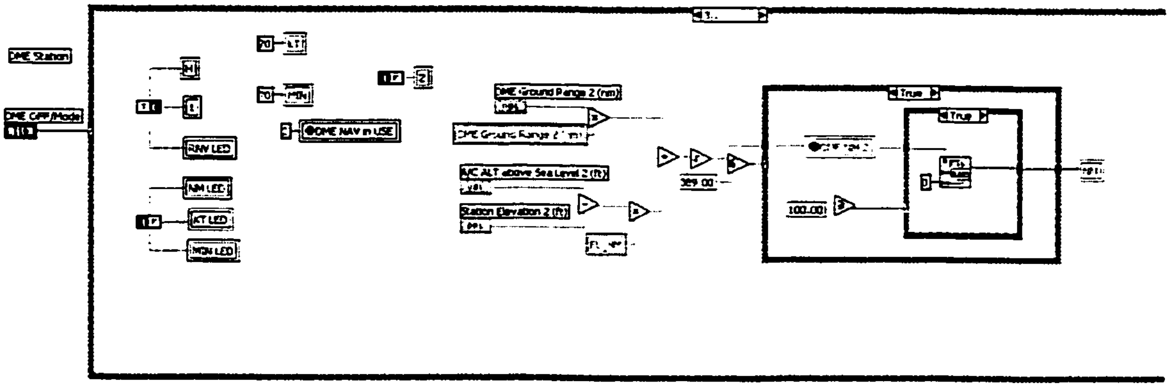

C/N COMM Station
C/N COMM OFF/Volum
C/N COMM Test
False
C/N COMM MHz I/D
C/N COMM MHz
C/N COMM MHz I/D
-1..1, Default
C/N COMM KHz I/D
C/N COMM KHz
C/N COMM KHz I/D
-1..1, Default
C/N COMM 25K
True
0.025
137..
118.00
True
C/N COMM U/S SW
MFR Mode Selector
4
C/N COMM STBY


C/N NAV Station
C/N NAV OFF/Volum
C/N NAV U/S SW
C/N NAV RAD
C/N NAV MHz I/D
C/N NAV MHz
C/N NAV MHz I/D
C/N NAV KHz I/D
C/N NAV KHz
C/N NAV KHz I/D
True
True
..-2
..-2
0.05
108..118, Default
113.30
False
C/N NAV USE

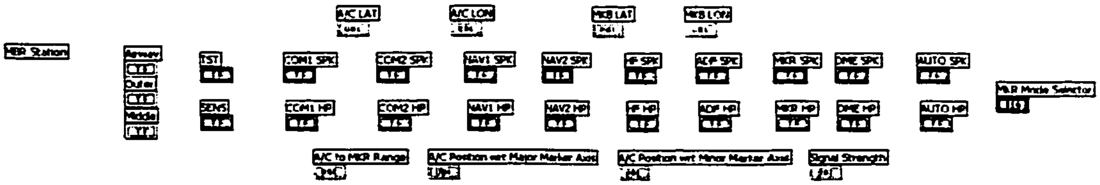

COMM Station
COMM OFF/Volum
COMM Test
False
COMM MHz I/D
MHz
COMM MHz I/D
-1..1, Default
COMM KHz I/D
KHz
COMM KHz I/D
-2
COMM 25K
True
0.025
137
118.00
True
COMM U/S SW
MFR Mode Selector
2



stop
ATC XPNDR Station
Mode Selector
2
True
True
Interrogate
Interrogated
Interrogated
500
False
IFF I/D
inc/dec
IFF I/D
IFF ID Code
IFF ID Code
IFF ID Code
IFF ID Code
ID Code Indicator
False
IFF ID code index
IFF ID code index
True
IFF ID code index
A
VFD
V (pointer)
HighLight
True
Mode Sw
IFF IDT
IFF ID Code
ID Code Received



3
DME Station
DME OFF/Mode
RNV LED
NM LED
KT LED
MIN LED
KT
MIN
DME NAV in USE
DME Ground Range 2 (nm)
DME Ground Range 2
A/C ALT above Sea Level 2 (ft)
Station Elevation 2 (ft)
389.00
100.00
True
True



MKR Station
Airway
Outer
Middle
A/C LAT
A/C LON
MKB LAT
MKB LON
TST
SENS
COM1 SPK
COM2 SPK
NAV1 SPK
NAV2 SPK
HF SPK
ADF SPK
MKR SPK
DME SPK
AUTO SPK
COM1 HP
COM2 HP
NAV1 HP
NAV2 HP
HF HP
ADF HP
MKR HP
DME HP
AUTO HP
MKR Mode Selector
A/C to MKR Range
A/C Position wrt Major Marker Axis
A/C Position wrt Minor Marker Axis
Signal Strength

APPENDIX 18

TIMER MODULE

```
/* Cycle Timer */

void CYCLE_TIMER()
{
# include <time.h>
# include <sys\timeb.h>

# include "glob_inc.c"

static unsigned long Start_Sec;     /* Starting time in seconds */

unsigned long Curr_Sec;             /* Current time in seconds */

static int Start_Thou,              /* Starting thousands of a second */
        Cycle_Count,                /* Cycle counter */
        2_Sec_Limit;                /* Cycles for 2 seconds */

int Curr_Thou;                      /* Current thousands of a second */
struct timeb t;
char dummy[30];
int Key;


/* PAUSE SIMULATION */

/*
If a simulation pause is requested then the program freezes here until
<F11> is pressed to resume the simulation or <F12> to end the simulation.
*/


        if (Cycle_Pause)
        {
                Key = 0;
                while (Key!=11)
                {
                        while(kbhit()) getch();
                        while(!kbhit());
                        while(kbhit()) Key = getch()-58;
                        if (Key>70) Key = Key - 64;
                        if(Key==26) Key=11;
                        if (Key==27) Key=12;
                        if(Key==12)
                        {
                        Cycle_Quit=1;
                        Key=11;
                        }
                }
                Cycle_Pause = 0;
                First_Cyc = 2;
        }
/*  READ AND STORE THE SYSTEM TIME */

          ftime(&t);                /* Read the system time */
          Curr_Sec = t.time;        /* Store current seconds */
          Curr_Thou= t.millitm;     /* Store current thousands */


/* STORE TIME ON FIRST PASS */

          if (First_Cyc)            /* If this is the first iteration */
          {
          Start_Sec = Curr_Sec;
          Start_Thou = Curr_Thou;
          Cycle_Count= 0;
          2_Sec_Limit = 100;
          }

/* NORMAL OPERATION*/

/*Every 2 seconds the iteration rate is checked and updated if needed */

          else
          {
                    Cycle_Count++;
                    if (Cycle_Count==2_Sec_Limit)
                    {
                              Cycle_Time = ((Curr_Sec-Start_Sec)
                                                  +0.001*(Curr_Thou-Start_Thou))/2_Sec_Limit;
                              Start_Sec = Curr_Sec;
                              Start_Thou = Curr_Thou;
                              Cycle_Count= 0;
                              2_Sec_Limit = 2/Cycle_Time; /* Cycles/2sec */
                    }
          }

/* NORMAL OPPERATION */

          if(!Real_Time)
                    Cycle_Time=Fixed_Time/1000.0;

return;
}
```

APPENDIX 19

BEARING AND RANGE FUNCTION

```
/* Range and Bearing Functions*/

#define deg2rad 3.141592653/180.0          /* Deg to Rad conversion factor */
#define rad2deg 180.0/3.141592653          /* Rad to Deg conversion factor */
#define Rearth 3437.746771                 /* Radius of the earth in nm */
#define m2nm 1.0/1852.0                    /* m to nm conversion */
#define ft2nm 1.0/6076.1                   /* ft to nm conversion */

#include <math.h>
#include "glob_inc.c"

void DISP_DME(int DD_Hours, int DD_Mins, float DD_Vel, float DD_Dist);
void DISP_ADF(float Angle);
void DISP_VOR();

/*Calculate the distance in nm between two points */

float AV_RANGE(double Lat1, double Lon1, double Alt1,
               double Lat2, double Lon2, double Alt2)
{
        double X1,Y1,Zl,X2,Y2,Z2;

        float Range;

/* Convert coordinated from degrees to radians */

        Lat1 = Lat1*deg2rad;
        Lon1 = Lon1*deg2rad;

        Lat2 = Lat2*deg2rad;
        Lon2 = Lon2*deg2rad;

/* Convert altitude from m to nm */

        Alt1 = Alt1*m2nm + Rearth;
        Alt2 = Alt2*m2nm + Rearth;

/* Convert coordinated to from polar to Cartesian */

        Xl = Alt1*cos(Lat1)*cos(Lon1);
        Y1 = Alt1*cos(Lat1)*sin(Lon1);
        Zl = Alt1*sin(Lat1);

        X2 = Alt2*cos(Lat2)*cos(Lon2);
        Y2 = Alt2*cos(Lat2)*sin(Lon2);
        Z2 = Alt2*sin(Lat2);

/* Calculate range in nm */

        Range = sqrt((Xl-X2)*(Xl-X2)+(Yl-Y2)*(Yl-Y2)+(Zl-Z2)*(Zl-Z2));
        return Range;
}
```

```
/* Calculate the bearing from one point to another */

float AV_BEARING(double LatF, double LonF, double AltF,
                                        double LatT, double LonT, double AltT)
{
        double XF,YF,ZF,XT,YT,ZT;

        static float Bearing;

/* Convert coordinated from degrees to radians */

        LatF = LatF*deg2rad;
        LonF = LonF*deg2rad;

        LatT = LatT*deg2rad;
        LonT = LonT*deg2rad;

/* Convert altitude from feet to nm */

        AltF = AltF*m2nm + Rearth;
        AltT = AltT*m2nm + Rearth;

/* Rotate both points so that FROM is at 0 longitude */

        LonT = LonT - LonF;

/* Convert coordinated to from polar to Cartesian */

        XT = AltT*cos(LatT)*cos(LonT);
        YT = AltT*cos(LatT)*sin(LonT);
        ZT = AltT*sin(LatT);

/* Rotate both points so that FROM is at N Pole */

        XT = XT*sin(LatF) - ZT*cos(LatF);

/* Calculate bearing from FROM to TO */

        if(XT == 0.0)
        {
                if(YT>=0.0)
                        Bearing = 90.0;
                else
                        Bearing = -90.0;
        }
        else
        {
                Bearing = atan2(YT,-XT)*rad2deg;
        }

        return Bearing;
}
```