A DATA COLLECTION SYSTEM FOR THE STUDY OF RF INTERFERENCE FROM
INDUSTRIAL, SCIENTIFIC, AND MEDICAL EQUIPMENT

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
William B. Drury
March, 1986
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. PROBLEM DEFINITION</td>
<td>5</td>
</tr>
<tr>
<td>A. ILS REGULATIONS</td>
<td>5</td>
</tr>
<tr>
<td>B. POSSIBLE AVIONICS INTERFERENCE</td>
<td>7</td>
</tr>
<tr>
<td>C. DATA COLLECTION REQUIREMENTS</td>
<td>8</td>
</tr>
<tr>
<td>D. DATA INTERPRETATION</td>
<td>11</td>
</tr>
<tr>
<td>III. EQUIPMENT</td>
<td>13</td>
</tr>
<tr>
<td>A. SELECTION</td>
<td>13</td>
</tr>
<tr>
<td>B. CALIBRATION</td>
<td>16</td>
</tr>
<tr>
<td>1. Antennas</td>
<td>16</td>
</tr>
<tr>
<td>2. Receiver</td>
<td>24</td>
</tr>
<tr>
<td>IV. DATA COLLECTION SYSTEM SOFTWARE</td>
<td>30</td>
</tr>
<tr>
<td>A. SYSTEM OVERVIEW</td>
<td>30</td>
</tr>
<tr>
<td>B. DEVICE DRIVERS</td>
<td>35</td>
</tr>
<tr>
<td>1. MR.DVD - Miniranger Device Driver</td>
<td>36</td>
</tr>
<tr>
<td>2. CK.DVD - Clock Device Driver</td>
<td>37</td>
</tr>
<tr>
<td>3. BB.DVD - Byte Bucket Device Driver</td>
<td>42</td>
</tr>
<tr>
<td>4. SL.DVD - SL-803-A Device Driver</td>
<td>43</td>
</tr>
<tr>
<td>C. FORTH DATA COLLECTION SOFTWARE DESCRIPTION</td>
<td>45</td>
</tr>
<tr>
<td>1. Definitions</td>
<td>48</td>
</tr>
<tr>
<td>2. Device Driver Words</td>
<td>50</td>
</tr>
<tr>
<td>3. Specific Device Driver Functions</td>
<td>50</td>
</tr>
<tr>
<td>4. Data Collection Loop</td>
<td>52</td>
</tr>
<tr>
<td>5. User Interface Software</td>
<td>56</td>
</tr>
<tr>
<td>V. DATA EVALUATION</td>
<td>60</td>
</tr>
<tr>
<td>A. PLOT PROCEDURE</td>
<td>64</td>
</tr>
<tr>
<td>B. WATERMAN, ILLINOIS DATA COLLECTION</td>
<td>67</td>
</tr>
<tr>
<td>C. ILS LOCALIZER DATA</td>
<td>95</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI. CONCLUSIONS</td>
<td>104</td>
</tr>
<tr>
<td>VII. ACKNOWLEDGEMENTS</td>
<td>107</td>
</tr>
<tr>
<td>VIII. REFERENCES</td>
<td>108</td>
</tr>
<tr>
<td>IX. APPENDICES</td>
<td>111</td>
</tr>
<tr>
<td>A. Data Collection Software - FORTH Listings</td>
<td>112</td>
</tr>
<tr>
<td>B. Miniranger Device Driver Assembler Listing</td>
<td>132</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>EMCO 3104 Biconical Broadband Antenna.</td>
<td>19</td>
</tr>
<tr>
<td>3-2</td>
<td>Biconical Antenna Propagation Pattern.</td>
<td>20</td>
</tr>
<tr>
<td>3-3</td>
<td>Antenna Factor for Bent-wire Antenna, Run 1.</td>
<td>22</td>
</tr>
<tr>
<td>3-4</td>
<td>Antenna Factor for Bent-wire Antenna, Run 2.</td>
<td>23</td>
</tr>
<tr>
<td>3-5</td>
<td>Calibration of EMC-25 Amplitude Signal.</td>
<td>26</td>
</tr>
<tr>
<td>3-6</td>
<td>Calibration of EMC-25 Frequency Signal, 25-50 MHz Band.</td>
<td>27</td>
</tr>
<tr>
<td>3-7</td>
<td>Calibration of EMC-25 Frequency Signal, 100-210 MHz Band.</td>
<td>28</td>
</tr>
<tr>
<td>4-1</td>
<td>Data Collection System Configuration.</td>
<td>31</td>
</tr>
<tr>
<td>4-2</td>
<td>Computation of True Position in Space Relative to ISM Unit Under Test.</td>
<td>33</td>
</tr>
<tr>
<td>4-3</td>
<td>Format of Recorded Data.</td>
<td>34</td>
</tr>
<tr>
<td>4-4</td>
<td>Miniranger Device Driver Flow Chart.</td>
<td>38</td>
</tr>
<tr>
<td>4-5</td>
<td>TICCNT Interrupt Service Flow of Control.</td>
<td>41</td>
</tr>
<tr>
<td>4-6</td>
<td>SL-803-A Device Driver Character Read Routine Flow Chart.</td>
<td>44</td>
</tr>
<tr>
<td>4-7</td>
<td>Sample CRT Display During Data Collection.</td>
<td>55</td>
</tr>
<tr>
<td>4-8</td>
<td>Menu of Options Displayed on CRT.</td>
<td>57</td>
</tr>
<tr>
<td>4-9</td>
<td>System Parameter Settings Displayed on CRT.</td>
<td>59</td>
</tr>
<tr>
<td>4-10</td>
<td>RS-232 Interconnection Diagram.</td>
<td>62</td>
</tr>
<tr>
<td>5-1</td>
<td>Block Diagram of Data Collection System Used for Waterman, Illinois Data Collection Flights.</td>
<td>69</td>
</tr>
<tr>
<td>5-2</td>
<td>Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.</td>
<td>72</td>
</tr>
<tr>
<td>5-3</td>
<td>Flight Data Machine A, 25 kW, 457 M Altitude, 180° Azimuth, RFI Shields Removed.</td>
<td>73</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page No.</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>5-4</td>
<td>Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.</td>
<td>74</td>
</tr>
<tr>
<td>5-5</td>
<td>Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.</td>
<td>75</td>
</tr>
<tr>
<td>5-6</td>
<td>Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields in Place.</td>
<td>76</td>
</tr>
<tr>
<td>5-7</td>
<td>Flight Data Machine B, 2 kW, 152 M Altitude, 300° Azimuth, RFI Shields in Place.</td>
<td>77</td>
</tr>
<tr>
<td>5-8</td>
<td>Flight Data Machine B, 2 kW, 457 M Altitude, 300° Azimuth, RFI Shields in Place.</td>
<td>78</td>
</tr>
<tr>
<td>5-9</td>
<td>Flight Data Machine B, 2 kW, 457 M Altitude, 0° Azimuth, RFI Shields in Place.</td>
<td>79</td>
</tr>
<tr>
<td>5-10</td>
<td>Flight Data Machine B, 2 kW, 457 M Altitude, 240° Azimuth, RFI Shields in Place.</td>
<td>80</td>
</tr>
<tr>
<td>5-11</td>
<td>Flight Data Machine B, 2 kW, 152 M Altitude, 140° Azimuth, RFI Shields in Place.</td>
<td>81</td>
</tr>
<tr>
<td>5-12</td>
<td>Flight Data Machine B, 2 kW, 152 M Altitude, 0° Azimuth, RFI Shields in Place.</td>
<td>82</td>
</tr>
<tr>
<td>5-13</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 320° Azimuth, RFI Shields in Place.</td>
<td>83</td>
</tr>
<tr>
<td>5-14</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 20° Azimuth, RFI Shields in Place.</td>
<td>84</td>
</tr>
<tr>
<td>5-15</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 260° Azimuth, RFI Shields in Place.</td>
<td>85</td>
</tr>
<tr>
<td>5-16</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 260° Azimuth, RFI Shields Removed.</td>
<td>86</td>
</tr>
<tr>
<td>5-17</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 230° Azimuth, RFI Shields Removed.</td>
<td>87</td>
</tr>
<tr>
<td>5-18</td>
<td>Flight Data Machine C, 3 kW, 152 M Altitude, 20° Azimuth, RFI Shields Removed.</td>
<td>88</td>
</tr>
<tr>
<td>5-19</td>
<td>Flight Data Machine D, 2 kW, 152 M Altitude, 230° Azimuth, RFI Shields in Place.</td>
<td>89</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page No.</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>5-20</td>
<td>Flight Data Machine D, 2 kW, 152 M Altitude, 20° Azimuth, RFI Shields in Place.</td>
<td>90</td>
</tr>
<tr>
<td>5-21</td>
<td>Flight Data Machine D, 2 kW, 152 M Altitude, 260° Azimuth, RFI Shields in Place.</td>
<td>91</td>
</tr>
<tr>
<td>5-22</td>
<td>Flight Data Machine D, 2 kW, 152 M Altitude, 200° Azimuth, RFI Shields in Place.</td>
<td>92</td>
</tr>
<tr>
<td>5-23</td>
<td>Flight Data Machine D, 2 kW, 152 M Altitude, 60° Azimuth, RFI Shields in Place.</td>
<td>93</td>
</tr>
<tr>
<td>5-24</td>
<td>Albany, Ohio Localizer Signal Flight 1.</td>
<td>98</td>
</tr>
<tr>
<td>5-25</td>
<td>Albany, Ohio Localizer Signal Flight 2.</td>
<td>99</td>
</tr>
<tr>
<td>5-26</td>
<td>Albany, Ohio Localizer Signal Flight 3.</td>
<td>100</td>
</tr>
<tr>
<td>5-27</td>
<td>Albany, Ohio Localizer Signal Flight 4.</td>
<td>101</td>
</tr>
<tr>
<td>5-28</td>
<td>Albany, Ohio Localizer Signal Flight 5.</td>
<td>102</td>
</tr>
<tr>
<td>5-29</td>
<td>Albany, Ohio Localizer Signal Flights 1-5.</td>
<td>103</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page No.</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>3.1</td>
<td>EMI Calibration Data</td>
<td>17</td>
</tr>
<tr>
<td>5.1</td>
<td>Ground-based Data versus Airborne Data</td>
<td>96</td>
</tr>
</tbody>
</table>
Interest in measuring absolute RF field strength in space comes from the Federal Aviation Administration's (FAA) concern regarding the levels of interference in the aircraft instrument landing system (ILS) localizer band caused by Industrial, Scientific, and Medical (ISM) equipment. ISM equipment is any apparatus that uses RF energy in one of the frequency ranges reserved by the International Telecommunications Union (ITU) for non-communication purposes. A typical industrial application of ISM equipment would be a machine that uses RF energy for the heat sealing of plastics.

The source of possible ILS localizer interference is the fourth harmonic frequency component of the ISM unit. The fourth harmonic of the 26.96 - 27.28 MHz ISM frequency band [1] is 107.84 - 109.12 MHz which overlaps the localizer band of 108.0 - 117.9 MHz [2]. Interference in this frequency range could be received by the localizer receiver and may produce erroneous readings on navigation instruments. The Federal Communications Commission (FCC) has placed limits on the amount of harmonic energy which may be radiated by an ISM device; however, these limits are specified for ground-based measurements only and so have little relevance to localizer interference.

Conventional techniques to model the RF power in free space are not practical in the study of ISM equipment because the transmitting characteristics of the machines are unknown. Therefore, actual
measured data are required. Heretofore, no formal studies have been conducted to measure the free space emissions of ISM equipment. A need has arisen for a system to collect data which will provide sufficient information to study these emissions.

The system developed to collect these data points employs an RF voltmeter, a range measurement device, a data storage unit and a computer to control the operation. The computer controlling the system provides a fast sample rate so that the distance between sample points is relatively short, providing complete coverage of the test area. A short sampling time also ensures that the distance and RF field level measurements are taken nearly simultaneously.

To obtain an accurate measurement of the RF energy transmitted by an ISM device into free space, both the measure of the electric field in absolute volts/meter and the corresponding position of measurement relative to the source are recorded. Evidence of data validity is provided by recording the time of each measurement and the frequency of the received signal. The time of measurement is used to confirm that the data points are taken at small intervals and the recorded frequency verifies that the receiver has been properly tuned. Also, a sufficient number of file descriptors and operator notes are required so that the data files are evaluated using the correct assumptions such as ISM device model, RF output power, and equipment configuration.

The result of the computer-controlled data collection system is a body of data that can be used to evaluate the free-space RF
emissions of a particular ISM unit. The output files provide data to plot the RF field strength in volts/meter versus distance from the unit. The outputs of this system are actual measurements of the ISM emissions in absolute values. This system provides the capability to obtain original, accurate information about the RF emissions produced by ISM equipment.

In this project several pieces of actual ISM equipment were measured in a simulated production environment at an FCC-certified open field test site. These ISM units were evaluated according to FCC certification procedures at the same time that the data collection flights were being performed. To study the effects of RF shielding on the units, measurements were taken both with the shielding removed and with the machine set up according to the manufacturer's specification. Also, the E-field of a localizer antenna array was recorded using this system. These measurements were taken so that the strengths of the ISM equipment and of typical localizer signals could be compared in absolute terms.

This data collection system has been used in actual field trials and has produced data files of sufficient detail for determining the radiation of several sources. The E-field measurements are accurate to within 1.5 dB and the distance to within 2 meters. These points are taken every 400 milliseconds or approximately every 20 meters with an aircraft velocity of 100 knots.

E-fields measured while flying over the site of an operating ISM unit were 16.9 to 42.9 decibels higher than ground-based measure-
ments taken at the same time. These measurements were taken at an FCC-certified open-field test site in Waterman, Illinois. For one ISM unit the removal of RFI shielding resulted in E-field emissions as much as 25 dB higher than those measured with all shielding properly installed; another ISM unit, with lower RF output power, exhibited less than a 7 dB increase with removal of shielding. All data from Waterman, Illinois flight tests are given in Chapter V.

Measurements of the ILS localizer signal at Albany, Ohio are also given in Chapter V. Data from the ILS localizer measurements and the ISM equipment were used to determine that ISM units which pass FCC certification requirements may produce signal levels greater than localizer signals at certain points. An ISM unit that meets FCC standards may produce signal levels 2.8 dB higher than the ILS localizer; an ISM unit that passes CISPR standards may produce signals 32.4 dB lower than the ILS localizer [3]. An ILS receiver susceptibility report cited in Chapter II states that the signal-to-interference ratio must be greater than 20 dB to maintain receiver errors less than ten percent of full-scale.
II. PROBLEM DEFINITION

The purpose of this section is to create a basis for understanding the data collection system functions and requirements. The ISM equipment regulations will be defined, interference concerns will be explained, data collection system requirements will be detailed and output interpretation procedures will be outlined.

A. **ISM REGULATIONS.** Industrial, Scientific, and Medical (ISM) frequency bands are defined by the International Telecommunications Union (ITU) as any of the frequency ranges reserved for equipment that use RF power for purposes other than communications. ISM frequency bands were created so that equipment using RF power could be allowed to produce unlimited levels of emissions in the band without interfering with other equipment. A piece of ISM equipment would typically be used for such purposes as the heat sealing of plastics in an industrial application and surgical diathermy equipment in medical usage. In all applications the equipment produces heat in the subject material by the dissipation of RF energy. These machines are available in many different RF power ratings, and nearly all units contain some RF shielding to suppress unwanted emissions.

Regulatory agencies such as the FCC and CISPR (International Special Committee on Radio Interference) have set legal limits on the strength of harmonic emissions permitted for ISM equipment and have developed a set of guidelines for the measurements of these harmonic emissions.
emissions. Currently the FCC specifies that no equipment shall produce signal levels of magnitude greater than 10 µV/m measured one mile from the unit at any frequency other than the allotted ISM band [4]. CISPR allows signal levels no greater than 54.0 dBµV/m at 30 meters for frequencies in the aeronautical bands in question.

For ISM equipment to be operated legally, each piece must pass a certification test performed by a company using FCC-approved facilities and procedures to conduct these tests. The current procedure by which the certification tests are conducted is one which involves determination of the RF decay factor by measurement and evaluation of RF harmonic field strength at one mile by extrapolation from a measured value at a shorter distance. The RF emissions are evaluated in the range from the fundamental frequency of the unit to the ninth harmonic. Measurements are normally conducted in a controlled environment at an open field test site designed expressly for this purpose, with the equipment set up according to manufacturer's specifications [5].

Existing FCC guidelines for ISM certification require only that measurements be made by use of a suitable antenna placed at relatively small distances above the ground. Therefore, interfering signals launched at high elevation angles are not measured. The resulting lack of information concerning signals launched from ISM devices is one impetus behind the FAA's interest in measuring ISM emissions at altitudes above ground level.
B. POSSIBLE AVIONICS INTERFERENCE. FAA concern for ISM emissions centers on radiation at the fourth harmonic frequency of the 26.96 - 27.28 MHz ISM band (107.86 - 109.12 MHz). These harmonic frequencies overlap the frequency band used by the (Instrument Landing System) ILS localizer. The localizer is an integral part of the ILS that transmits RF signals with audio frequency modulation which are received and decoded by a receiver on board an aircraft. These signal levels are interpreted by the receiver to indicate the aircraft's lateral position in space relative to the localizer centerline (usually coincident with the runway centerline). Interference with the localizer signal could produce an erroneous indication which would cause the pilot to approach the airport on an incorrect or non-linear path, with the potential for reduced safety.

Studies have been conducted to examine the performance of localizer receivers in the presence of co-channel interference. In one such report written at the Institute for Telecommunication Sciences [6] in Boulder, Colorado, four localizer receivers were tested with four types of interference. In each case, receiver performance was most affected by a continuous wave (CW) interference signal. This is significant to ISM interference considerations because the signal produced by most ISM equipment is essentially CW. This localizer interference study used several different receiver conditions as the criteria for an interfering signal large enough to cause significant degradation of the receiver's performance. In the conclusion of this report it is stated that the required signal-to-interference ratio to
maintain less than 15 μA course deviation error, or 10% of full scale, is nearly constant for each receiver and that the localizer receivers require about 20 dB signal-to-interference ratios.

All FCC certification guidelines for ISM equipment specify only ground-based measurements and no correlation has been shown between ground-based ISM measurements and free space measurements; this has generated concern over the applicability of present-day FCC limits to ILS navigation integrity. Therefore, a need has arisen for a procedure to collect data concerning RF field strengths in free space near an ISM device. The following sections of this paper describe the development of such a data collection system and provide illustrations of output data analysis.

C. DATA COLLECTION REQUIREMENTS. The data collection procedures used must have the capability of measuring absolute RF field strengths in free space and of determining the precise location each measurement was taken. In addition, the measurements need to be taken at enough positions that a complete view of the free space emissions can be seen. To take measurements in the air around an RF source implies that some type of aircraft must be used to transport the measurement device and therefore that the devices used must be compatible with the aircraft (in this case a single-engine airplane) environment.

The use of an airplane to carry the data collection system places many constraints on the type of system that can be used. Since
there is no easy and accurate position reference available, some type of three-dimensional position measurement system must be incorporated. The limitation on the operator's movement while in the small airplane requires that the system be made very easy to use. The need for data points to be close together for complete analysis, coupled with the required velocity of the airplane, requires that the interval between measurements be minimized. The above requirements necessitate the implementation of an automatic data collection system employing a computer as a controlling mechanism. Another constraint placed on the data collection system by the use of an airplane is the type of antenna used. For reasons of practicality the antennas used must either be those existing on the airplane or one designed for use on an airplane.

To measure the ISM emissions in absolute values the entire measurement system must be calibrated in absolute terms. First, the antenna must be calibrated so that an antenna factor can be derived which, when added to the receiver measurement, will provide the RF E-field in dBµV/m (decibels relative to 1 microvolt/meter). This antenna calibration can be complicated by the fact that the antenna is attached to the airplane which is a large metallic object that changes position relative to the RF source. Second, the receiver must be calibrated for an accurate reading, and finally, any equipment such as A/D converters used to encode the receiver signal levels must also be calibrated.
Use of a computerized system requires that software be written to perform the data collection operations. This software must satisfy several needs. The controlling software must execute so that all data pertinent to the RF field analysis be collected and recorded for future use. Data that need to be stored include the position, detected field strength, receiver frequency, and the time of measurement. The software execution should be fast enough so that data are updated at a rapid rate for closely-spaced points, and there should be some type of information passed to the operator that indicates what is being recorded. In addition, the system should operate with interactive control by the operator, and for reliable operation it should be very simple to use.

The choice of equipment for the airborne data collection system is limited by several factors. All equipment directly involved in the data collection and storage operations must be computer compatible, or some type of interface to the computer must be produced. An important consideration for this type of operation is the portability and power requirement of each equipment item, since each item will either be carried in a small airplane or transported to a remote test site in a land vehicle. Airborne equipment must be small and light and able to operate from the power supplied by the airplane. The data collection system hardware also must be able to operate in the ambient conditions encountered in the airplane such as low/high temperatures, turbulence and vibration.
The computer used for this operation is also subject to the constraints of portability and various ambient conditions. The system computer must be able to support several peripheral communications channels and have available a fast software system. In addition, the computer must be easily transportable and capable of operating on electrical power available at a remote test site.

D. DATA INTERPRETATION. The purpose of the data collection system is to provide a set of data that can be evaluated to indicate RF radiation levels in free space near a radiating source. Interpretation of the collected data should provide detailed information that can be evaluated to detect the presence or absence of any vertical or horizontal RF lobing.

The interpretation should also provide a quantitative measure of the absolute E-field existing over the site. This information should give the measured E-field values and frequencies along with the position of the measurement. Absolute RF levels are needed to study the effects of the ISM equipment on the ILS localizer, and to compare airborne measurements to ground-based certification measurements. The reported levels must be independent of the system that performed the measurement. System independence is maintained by calibrating the receiving equipment so that the signals are measured in absolute terms. E-fields are measured in units of decibels relative to 1 microvolt/meter (dBµV/m).

Both the lobing and the absolute E-field measurement can best be evaluated if plots of the recorded data are made. These plots can
be of received signal value versus horizontal position or of any other two parameters that produce plots which allow easy visual evaluation.
III. EQUIPMENT

A. SELECTION. The equipment chosen to execute the tasks set forth in the previous chapter was selected by engineering judgement to be the most suitable. Factors that were taken into consideration for equipment selection included the applicability to the problem solution, immediate availability and the economics of using existing equipment versus new purchases.

The computer chosen to control the data collection was a Heath H89 [7]. This computer was selected because the H89 was deemed capable of performing the required functions, and a unit plus a backup were readily available. The H89 is a complete functional computer that supports a console screen, console keyboard, multiple disk drives, and three RS-232 serial data ports. In addition, the FORTH language is available for use on the H89. Use of FORTH has resulted in reduced software development time as compared to assembler coding or machine language, and in faster execution time compared to the BASIC language.

An Electro-Metrics EMC-25 interference analyzer receiver [8] was chosen to detect the RF signals. This receiver was provided to the Avionics Engineering Center by the FAA as government-furnished equipment for this work. The EMC-25 is not only capable of detecting the RF emissions of ISM equipment, but also provides external signals that indicate the received signal level and the receiver control settings. The EMC-25 also contains a rechargeable battery pack as a
power source that will provide enough power for the unit to operate approximately 12 hours between charges [9]. This is an important consideration when operating in a small airplane with limited electrical power.

Outputs provided by the EMC-25 to indicate received signal amplitudes and frequency are DC voltage levels of 0 to +1.5v nominal [10]. The DC voltage for signal amplitude is derived from the meter terminal voltage and therefore is an indication of the meter deflection, while the frequency output is a measure of the tuner setting. In addition to the above signals, there are four binary data lines encoded as a hexadecimal digit that indicates the frequency band number, and seven binary data lines from the attenuator switch. Each data line from the attenuator switch indicates that a particular attenuator setting has been selected. These seven data lines are encoded by an 8 to 3 line encoder to give a 3 bit octal representation of the attenuator switch position.

A Serial Lab Products SL-803-A Intelligent Remote Serial I/O unit [11] was incorporated to convert the analog signals from the EMC-25 into ASCII characters and to make available upon request all of the EMC-25 signals on an RS-232 data communications link. The SL-803-A was chosen for its wide range of capabilities and for its ease of application. Up to sixteen channels of analog signals may be input to the SL-803-A for A/D conversion and up to eight digital input lines may be used. This exceeds the system requirement for two chan-
nels for A/D conversion and seven digital input lines. The SL-803-A, controlled by characters sent over the RS-232 line, is transparent to any transmissions until it detects an ASCII character that has been selected by the user as its control character. It then reads the subsequent ASCII codes and acts according to the designed command convention [12]. Enabling of specified channels and the selection of either ±2v or ±10v A/D conversion are among the programmable modes of the unit.

To measure the distance from the airplane to a point on the ground and provide a data link between the airplane and ground station, a Motorola Miniranger is used together with a custom Ground Telemetry Processor (GTP) [13]. The Miniranger provides a measurement of distance every 7.5 milliseconds between the airborne and ground Miniranger transponders accurate to ±2 meters [14] and outputs the range in ASCII characters from the base unit. The Miniranger system data link is a transparent two-way communication link which is used in this system to transmit ASCII characters between the SL-803-A in the airplane and the H89 computer on the ground. For this system the Miniranger remote transponder is in the airplane and the base station on the ground with the H89 computer and Byte Bucket [15] tape drive. The SL-803-A, located in the airplane, communicates with the H89 computer by sending and receiving characters over the Miniranger's telemetry data link. Miniranger-determined slant range is formatted by the Ground Telemetry Processor into an ASCII character string. This character string is read by the H89 computer and the range data are recovered.
Data collected by the equipment described above require storage for subsequent evaluation. To accomplish this task an ADPI Byte Bucket digital cassette tape player/recorder [15] was used. The Byte Bucket is a cassette tape drive that can be placed into its different operating modes either manually or by the transmission of control characters on the RS-232 data link. The Byte Bucket uses digital cassette tapes capable of storing up to 230,000 bytes of data per track with one track on each side [16].

B. CALIBRATION. To make E-field measurements in absolute $\mu$V/m, the system that collects the data must be calibrated. This section details the procedures used for calibrating the antennas, receiver, and SL-803-A A/D converters used in the data collection system. Table 3.1 contains the calibration data for the antennas and cables used for this project.

1. **Antennas.** Three antennas were selected for the project, two mounted on the airplane for signal detection during data collection, and the other a broadband biconical antenna used for calibrating the airplane antennas.

The antenna used for receiving signals of the fourth harmonic of the ISM frequency (108.48 MHz) is a bent dipole antenna mounted near the top of the airplane's vertical stabilizer. This antenna had already been a part of the airplane; it was designed to be used with the VOR (VHF Omnidirectional Range) [17] navigation system operating in the frequency band 108.0 to 117.9 MHz [18].
Biconical Antenna
Antenna Factor = 16.4 dB @ 27 MHz
Antenna Factor = 13.1 dB @ 109 MHz
Source: Three antenna method calibration. Sept. 9, 1983

Bent-dipole antenna on Saratoga aircraft N8238C
Antenna factor = 53.4 dB @ 27 MHz
Antenna factor = 13.1 dB @ 109 MHz
Source: Calibration versus Biconical antenna using substitution. Nov. 7, 1983

CB antenna on Saratoga N8238C
Antenna Factor = 9 dB @ 27 MHz
Source: Data collected on January 3, 1984 versus values predicted by Geometrical Optical Theory model.

Cables
EMI Cable A (35 feet)
-0.7 dB @ 27 MHz
-1.2 dB @ 109 MHz
EMI Cable B (80 feet)
-1.6 dB @ 27 MHz
-3.2 dB @ 109 MHz
EMI Cable C (5 feet)
-0.2 dB @ 27 MHz
-0.4 dB @ 109 MHz
Source: All cables calibrated Sept. 12, 1983

Dual Directional Coupler - HP778D serial no. 1144A04704
27 MHz - Both ports -32.6 dB
109 MHz - Both ports -22.0 dB

Table 3.1 EMI Calibration Data
For the reception of the fundamental frequency 27.12 MHz, an antenna was mounted to the underside of the left wing. This antenna, designed for 27 MHz citizen's band radio communications uses in an airplane, was a bent wire antenna with an impedance matching network in its base for matching to a 50 ohm coaxial transmission line. A temporary mounting fixture was manufactured for this antenna so that it may be removed from the airplane when not being used.

An EMCO 3104 biconical broadband antenna (see figure 3-1) is used at both the fundamental and fourth harmonic frequencies for ground-based operation. This antenna is designed for an operating frequency range of 20-200 MHz and has a donut-shaped propagation pattern similar to that of a dipole antenna (see figure 3-2). The biconical antenna was calibrated at an open field site using the "three antenna" method [19] and found to have an antenna factor of 16.4 dB at 27 MHz and 13.1 dB at 109 MHz [20].

The calibrated biconical antenna is used to transmit a signal of known strength for calibrating the airplane antennas. This is accomplished by open-field location of the biconical, driven by a signal generator, and using the data collection system to measure the resulting electric field while flying over the biconical antenna location. The collected data are then analyzed for the propagation factor using a least-squared-error curve fit. This propagation factor is used to predict the field strength levels that appear at the airplane. Measured values of field strength are subtracted from values obtained
Figure 3-2. Biconical Antenna Propagation Pattern.
using a mathematical model based on a geometrical optical theory modified for finite conductivity [21]. The difference between the measured E-fields and those predicted by the mathematical model is the antenna factor. Figures 3-3 and 3-4 are plots of the antenna factor for the 27 MHz bent-wire antenna obtained using this method; this antenna has an antenna factor of 8.65 dB.

The method detailed above was applicable to both airplane antennas; however, the bent-dipole antenna was calibrated at 109 MHz with considerably less expense using the calibrated biconical antenna and the substitution method. An uncalibrated dipole antenna was driven to transmit an RF field at 109 MHz; the calibrated biconical antenna was placed 50 feet from this antenna at height equal to the height of the bent-dipole antenna on the aircraft. Received signal at the biconical antenna was measured by the EMC-25 receiver; the biconical antenna was then removed and the airplane was positioned so that the bent-dipole antenna occupied the same space the biconical antenna had previously occupied. The received signal level of the aircraft antenna was measured using the EMC-25 receiver and subtracted from the first signal level measurement. The difference between the two values is the difference between the antenna factor of the biconical antenna and the bent-dipole antenna. Since the antenna factor of the biconical antenna is known, the antenna factor of the bent-dipole antenna is then also known. The antenna factor of the bent-dipole antenna was determined from these measurements to be 13.1 dB at 109 MHz.
Figure 3-3. Antenna Factor for Bent-wire Antenna, Run 1, 27 MHz, Data Set 1.
Figure 3-4. Antenna Factor for Bent-wire Antenna, Run 2, 27 MHz, Data Set 2.
2. Receiver. The Electro-Metrics Interference Analyzer model EMC-25 is used as the major component of interference analysis systems for the 14 kHz to 1 GHz range [22]. The EMC-25 is tunable in 15 frequency bands for the frequency range specified and is capable of measuring signal levels from 20 dBµV to 120 dBµV within 1.5 dBµV at frequencies above 25 MHz (0 dBµV to 100 dBµV below 25 MHz).

The EMC-25 receiver is periodically calibrated by the manufacturer to ensure the accuracy of the meter reading; however, the DC amplitude and frequency outputs require calibration for this project. The amplitude output was calibrated by providing various levels of signal input to the receiver and measuring the voltage output for each corresponding meter reading. The frequency output was similarly calibrated by tuning the receiver to a signal provided by a frequency-calibrated RF source and measuring the corresponding analog output voltage.

The EMC-25 signal outputs were calibrated with all the connections made as they would be in actual use, to eliminate the possibility of invalid calibration due to loading effects of external circuitry. For this data collection system, the previous statement implies having the outputs of the EMC-25 wired to the analog inputs of the SL-803-A during the calibration procedures. The analog-to-digital conversion of the SL-803-A also requires calibration. This calibration involves providing a known voltage to each analog input and interrogating the unit to read the digital output.
To simplify calibration and eliminate possible errors in the calibration of the SL-803-A, both the EMC-25 receiver outputs and the A/D conversion of the SL-803-A were calibrated simultaneously. Receiver and A/D calibration were accomplished together by providing the proper calibrated RF input signal to the EMC-25 and requesting the converted digital data string from the SL-803-A I/O unit. The H89 computer was used to read the SL-803-A character string output and display this string on the CRT for the operator to record.

The SL-803-A Intelligent Remote Serial I/O Unit performs A/D conversion to the nearest millivolt when in the ±2V mode. However, the output of this conversion is not absolutely stable; with a constant voltage input to the SL-803-A, the digitized output varies by less than 3 millivolts around an average value. To minimize error caused by the slight instability of the A/D conversion, a small routine was written for the H89 computer that causes it to request output from the SL-803-A fifteen times and then determine the mean of the sampled values. The computed average was output on the CRT along with a list of all the sampled values and this average was recorded as the calibrated output.

Calibration plots of the digitized voltage output vs. receiver parameter are given in figures 3-5 through 3-7. Figure 3-5 is a plot of the EMC-25 meter reading vs. digitized output. The solid line on the plot is a linear approximation used during data collection to compute the meter reading. Figures 3-6 and 3-7 are plots of digitized
Figure 3-5. Calibration of EMC-25 Amplitude Signal.
Figure 3-6. Calibration of EMC-25 Frequency Signal, 25-50 MHz Band.
Figure 3-7. Calibration of EMC-25 Frequency Signal, 100-210 MHz Band.
output vs. frequency for the 25-50 MHz and 100-210 MHz bands respectively. These plots show a piecewise linear approximation used for frequency computation during data collection. These linear approximations are not the result of any best curve fit, but were chosen to give numbers convenient for computation during data collection. These approximations do not in any way affect the validity of the collected data since they are only used for providing a display to the operator. The data recorded are raw data sent to the H89. It is left to the analysis routine to determine a digital approximation of the calibration curves with sufficient accuracy for valid interpretation. An example of data analysis is given in Chapter IV; this analysis uses the same data shown above except that the data used for analysis are not rounded to the nearest integer as is done for the FORTH software operator-display routines.
IV. DATA COLLECTION SYSTEM SOFTWARE

A. SYSTEM OVERVIEW. The data collection system configuration (shown in figure 4-1) consists of a Heath H89 computer that controls several peripheral devices to collect and record the RF interference E-field amplitude and frequency, and the position and time of measurement. To measure RF interference levels, an Electro-Metrics EMC-25 Interference Analyzer is incorporated in the system. The EMC-25, as detailed in Chapter III, is capable of measuring the RF fields from ISM equipment and provides external signals so that the detected field can be documented.

Making the EMC-25 signals intelligible to the H89 computer entails converting the two analog signals to digital form, encoding the EMC-25's binary data lines, and making all the information available to the H89 on request in a serial ASCII character format. These tasks are accomplished by the remote serial I/O unit. This unit provides both the A/D conversion and the serial ASCII character data stream.

For the RF field measurement to be useful in determining the propagation pattern, the position for each measurement must be recorded. The Miniranger with telemetry data link is used to measure the slant range from a ground point to the airplane, while the altitude and airplane heading are manually read from the instruments in the airplane. In performing data collection maneuvers, the pilot flies in a straight line at a constant altitude directly over the test
Figure 4-1. Data Collection System Configuration.
site. When this is done, the position in space at every point can be calculated by equation 4.1 from the recorded altitude, heading and Miniranger distance (see figure 4-2).

\[ R = \sqrt{R_m^2 + 2Y_{off}} - \sqrt{R_m^2 - Alt^2 \sin\theta - Y_{off}} \]  
Eq. 4-1

A system clock is also kept so that the time of each measurement can be recorded with the other data. Time of measurement is useful in data reduction by providing evidence of any data collection interruption. The clock is a software counter which keeps time via interrupts provided by the H89 at precise two-millisecond intervals (see discussion of the clock device driver below).

Data is stored on a digital cassette tape using an Analog & Digital Peripherals Inc. Byte Bucket digital cassette tape player/recorder. The Byte Bucket is controlled by transmitting appropriate ASCII control characters to it. Each data file contains a header block that contains a user-defined description of the file, data records, and a trailer string formatted as a data record, but with all numbers set to "9" to mark the close of the file. The data set format is given in figure 4-3.

Data transfer between peripheral devices is controlled by a routine written in FORTH and executed by the H89 computer. While performing data collection, the routine runs in a continuous loop that inputs data from the three sources and stores it on tape. The loop
Figure 4-2. Computation of True Position in Space Relative to ISM Unit under Test.
FILE: N
DATE: DD-MMM-YY
NAME:
EQUIPMENT:
ALTITUDE:
HEADING:
RECORD 1
RECORD 2
RECORD 3
RECORD 4
... 
RECORD M
END-OF-FILE RECORD

FILE FORMAT

<table>
<thead>
<tr>
<th>TIME</th>
<th>AMPLITUDE SIGNAL</th>
<th>FREQUENCY SIGNAL</th>
<th>HEX CODE</th>
<th>RANGE</th>
</tr>
</thead>
</table>

TIME: 'HH:MM:SS'

AMPLITUDE SIGNAL: 'C,+V.VVV,'

C = SLP-803-A channel number
V.VVV = Amplitude Voltage Reading

FREQUENCY SIGNAL: 'C,+V.VVV'

C = SLP-803-A channel number
V.VVV = Frequency Voltage Reading

HEX CODE: Two characters representing two hexadecimal bytes encoded as 'XBBBBAAA'.

BBBB = Frequency Band Number
AAA = Attenuator Setting Code

RECORD FORMAT

'99:99:999,+9.999,9,+9.9999999999'

END-OF-FILE RECORD

Figure 4-3. Format of Recorded Data.
also creates a display on the CRT to give the operator an indication of data contents and checks for input from the console keyboard to accept user commands. User commands are software-limited to a predefined set of inputs that control when data collection and data storage are enabled, plus commands to change the file header and to quit data collection (details of the provided user commands are given in the section below, entitled "DATA COLLECTION LOOP").

B. DEVICE DRIVERS. For each device external to the FORTH environment, a device driver routine tailored to perform that device's specific control functions is loaded in the memory of the H89. A device driver is a routine written in H89 assembler language that defines the character length, parity, and number of stop bits; sets the data transmission rate; and provides proper character timing to avoid overrun.

In some cases the device driver will maintain a data buffer so that the calling program need only read the buffer when it needs data. In this way the data may be sent from the peripheral device to the device driver at any time and is available when the calling program requests it. A device driver routine can be made to read, write, open, or close by passing the proper codes in the CPU (Central Processing Unit) registers upon calling it [23].

Calling a device driver from FORTH is accomplished by pushing the proper codes onto the system stack and executing the word SCALL [24]. SCALL pops the codes from the stack, puts them in the CPU
registers, and performs a jump to subroutine (JSR) with the device driver as the destination. The special functions of each device driver used in the data collection are discussed below.

1. **MR.DVD - Miniranger Device Driver.** The Miniranger driver contains a polling routine that checks for character input on the port to which the Ground Telemetry Processor (GTP) auxiliary data port is connected. The GTP auxiliary port is an RS-232 connection through which characters are continuously sent in a fixed ASCII format; this character stream contains the Miniranger slant range.

   If there is no character present when the port is polled, the routine returns to the calling program immediately. The special way that this routine returns will be explained in the discussion of loading the clock device driver. See the section entitled CK.DVD. When an input character is detected, it is read and either stored or discarded depending on its position in the data stream. If the character is a valid range digit (known by counting the number of characters detected since the start of the fixed block), it is stored in one of two buffer areas; otherwise, it is discarded and no changes are made to the buffer. The exception to the previous statement occurs upon receipt of the character marking the end of the data stream. When this sentinel is detected, the routine checks to see if the proper number of characters were received in the data block. Upon receipt of the last character and confirmation of a proper number of characters, the buffer area that was just filled with range data is
flagged as the output buffer and a second buffer previously flagged as output is marked for storage of range data in the next data block. When a calling program requests range data from the Miniranger device driver the information located in the buffer flagged as output is transferred into the memory space provided by the calling program. The Miniranger device driver flow chart is shown in figure 4-4.

The Miniranger polling routine is executed every two milliseconds to ensure it is performed often enough to receive all incoming data. The special loading procedure of the Miniranger device driver that causes its polling routine to be performed every two milliseconds is discussed below in the presentation of the clock device driver (CK.DVD) since they are loaded in exactly the same way.

2. CK.DVD - Clock Device Driver. The clock device driver routine is the actual system clock. It does not perform I/O to a hardware clock but performs the clock functions itself, in software. In the most strict sense this is not really a device driver because it does not perform any I/O operation, but the functions it provides makes it appear as though I/O operations are being performed with a clock. This device driver is loaded in a manner such that on every occurrence of the system TICCNT [25] interrupt, at two millisecond intervals, a routine in the clock driver is executed that increments the clock counter. The routine then checks the clock counter to see if 500 TICCNTs have been received; if so the clock counter is cleared and the stored time-of-day string is incremented by one second.
Figure 4-4. Miniranger Device Drive Flow Chart.
The time-of-day string is an eight-byte area where the ASCII character representation for current time is stored. The clock driver polling routine does the appropriate data handling so that time is stored in a proper form using a 24-hour clock format (00:00:00 - 23:59:59). When a time read is requested by another program, this string is transferred. When a time string is written to the clock driver, the current time string is replaced by the string supplied by the calling program.

Upon receipt of a TICCNT interrupt, the H89 vectors to a service routine whose address is stored at the system location UIVEC+1. On system reset, the H89's monitor loads an address into UIVEC+1 that is the beginning of a TICCNT service routine located in the monitor. This routine restores saved registers and returns from interrupt. The loading routines of both the clock and the Miniranger device drivers alter the flow of the TICCNT interrupt service routine by taking the address stored at UIVEC+1 and storing it as the address of a jump instruction at the end of their routines. They then store the beginning address of their respective routines at UIVEC+1 so that each will be executed on the occurrence of the 2-millisecond TICCNT interrupt.

To illustrate the procedure and effects of loading the clock and Miniranger device drivers in the manner above using the location UIVEC+1, the following example is given. In this example the clock device is loaded first. However, it is noted that the order of the
load makes very little difference in the results; it merely changes
the order in which the two routines are updated.

When the clock device driver is loaded, the address of the
monitor's TICCNT service routine is transferred from UIVEC+1 to the
address of a jump at the end of the clock update routine and the
starting address of the clock update routine is placed at UIVEC+1.
When the Miniranger driver is loaded, the starting address of the
clock routine is transferred from UIVEC+1 to the address of a jump at
the end of the Miniranger routine and beginning address of the
Miniranger routine is placed at UIVEC+1.

On the occurrence of a TICCNT interrupt, control is given to
the start of the Miniranger routine (its address is at UIVEC+1). When
the Miniranger routine is finished it will cause a jump to the start
of the clock routine, and when the clock routine is finished it will
cause a jump to the TICCNT service routine which will cause return
from interrupt. This ensures both that the clock is updated every two
milliseconds and that the Miniranger driver checks for input every two
milliseconds, fast enough so that no data are lost if the GTP auxi-
liary data port is operated at 4800 Baud or lower. The flow of
control for a TICCNT interrupt service is shown in figure 4-5 after
the clock and the Miniranger device driver have been loaded.

Since the TICCNT interrupt can occur (and the clock and
Miniranger polling routines be executed) at any time, care must be
taken that the polling routines are executed fast enough to return
Figure 4-5. TICCNT Interrupt Service Flow of Control.
from interrupt without destructively interfering with a process in action. The time that the TICCNT service routine requires is critical to the SL-803-A device driver. The SL-803-A driver makes requests on a regular basis that instruct the SL-803-A I/O device to send data, and then waits until all characters have been received. The TICCNT interrupt cannot be disabled while this driver is receiving data, for this would slow down the clock. Thus the TICCNT service routine must be fast enough to return from interrupt before an input character is lost. If the SL-803-A is operating at its fastest rate of 9600 baud then each character requires slightly more than 1 millisecond to be transmitted. Therefore, the TICCNT service routine is limited to less than one millisecond so that no interference will occur.

3. **BB.DVD - Byte Bucket Device Driver.** The Byte Bucket device driver acts as a controlling link to the digital tape player/recorder on output and as a character buffer on input.

The only special function of the output routine is the ability of the device driver to respond to XON and XOFF ASCII commands sent by the Byte Bucket. This allows for temporary suspension of transmission so that the Byte Bucket may be used in its write-verify mode. In this mode, the Byte Bucket accepts a certain number of characters and then sends an XOFF so that it can write those characters to tape and read them back to verify that they were written properly. When the Byte Bucket has verified the write operation it sends an XON to restart the driver.
On input the Byte Bucket device driver maintains a buffer of characters read from tape so that when a user program requests a certain number of characters they are taken from the buffer. When the buffer area is empty and more characters are needed, the device driver activates the Byte Bucket until it has received approximately 253 characters (250 plus the number sent during the time it takes for the Byte Bucket to respond to an XOFF). These characters are stored in the device driver buffer so the calling program may read them.

4. SL.DVD - SL-803-A Device Driver. The SL-803-A device driver is a character timing link to the I/O port upon a write request and a character timeout link on read request (see figure 4-6). The driver maintains no data buffer area nor character counters as in previous driver routines.

When the SL-803-A device driver is requested by a calling program to write a string of characters, the routine executes a loop that checks the port to see if it is ready to accept characters, outputs a new character if the port is ready, and loops until the port is ready again. Included in the loop is a check of input from the console keyboard to see if a control character signaling a premature stop has been detected. The loop will be executed by the driver routine until either all of the supplied string characters have been output or a premature end has been entered from the keyboard.

When a read request is made to the SL-803-A device driver it goes into a loop that looks for input characters to satisfy the
Figure 4-6. SL-308-A Device Driver Character Read Routine Flow Chart.
request. The read loop is similar to the write loop; the difference is in the amount of time the read loop allows for character input. The read routine of this driver contains a timing loop such that only a certain amount of time is allowed for a character to appear at the port after the previous character has been read. If the character timing loop detects that too much time has been taken, the routine exits the read loop and fills the calling program's read request with a string of nines. The read timing loop is implemented so that the routine will not loop forever if a character is lost in transmission to the port, or if too many characters are requested. Providing a read timeout loop is important when using the Miniranger datalink to send data since characters are occasionally lost in transmission on this radio link.

C. FORTH DATA COLLECTION SOFTWARE DESCRIPTION. FORTH [26] is a threaded interpretive programming language. The basic programming unit of the FORTH language is a "word." A word, in FORTH, is the definition of some procedure or function. Each word is created in FORTH either by making reference to words previously defined or by the creation of machine code from an assembly language routine. Most FORTH words are created by references to existing words; this makes development time faster and eliminates the creation of redundant segments of machine code. A compiled FORTH word is a list of addresses that give the location of other FORTH words, except in the case of an assembly-coded word. When a word is executed, the FORTH interpreter initially goes to the first address in the word definition
and executes the word located there; then it goes to the second address in the word definition to execute that word; this process continues until the list of addresses has been exhausted. This process is known as "threading" together the addresses to form a new word; hence the description "threaded interpretative language."

FORTH words that are created by writing assembler code segments do not consist of a list of addresses; they are executable machine code. The FORTH kernel is a collection of machine code segments that can be referenced to create more complex words. All FORTH word definitions eventually reference machine code segments in the FORTH kernel, or user defined code segments, to perform their desired tasks. By this process a FORTH word becomes a list of address references to small groups of machine code routines. These addresses are linked together so that the execution of small machine segments will proceed in a manner as to produce the desired result.

The process of creating new FORTH words by making reference to existing words means that the function of each new word becomes increasingly more complex. However, the definition of new words becomes easier since words previously defined provide individual functions and using these functions only requires references to existing words. This manner of creating functions can ultimately lead to the definition of one word that will cause execution of an entire system. In this particular application, that one word is "MENU." MENU is a word that provides a list of options to the system operator, prompts the operator for input and causes execution of any valid user command.
Parameters are passed among routines in an area of memory designated as the system stack. Passing parameters on the system stack minimizes the amount of overhead needed to make calls to other words, but forces all words to adhere to strict rules. The amount of information left on the system stack after execution may vary among words, but any single word must always leave the same amount, regardless of the operation's result. Correspondingly, a sufficient amount of information must be placed on the system stack before calling a word that uses data from the stack. A FORTH program must be written such that all parameters left on the stack by one word are taken off the stack by another due to the limited memory space allotted to the system stack. For most applications the system stack is large enough to hold all parameters to be passed among words. However, if too many parameters are placed on the stack (usually the result of a programming error) the system may crash.

The purpose of this FORTH software is to provide a master controlling program that collects the desired data, processes the data, creates a display on the CRT, and stores the data on magnetic tape. The finished program requires very little user input; messages explain each available command, and each command is initiated by a single key stroke.

The FORTH words can be divided into general categories by the purpose for which they were written. These categories are described as follows:
1) Definitions of constants, control and initialization of strings, and memory allocation for data I/O.

2) Interface to H89 system device drivers.

3) I/O operations for each device driver routine.

4) Continuous collection, computation, display, and tape storage of data to/from each peripheral unit.

5) User interface that provides explanations, lists menu options, and prompts the user for input.

1. Definitions. These definitions consist of constants used in calculations, fixed control strings, and the memory space allocation for variables and I/O strings. The Byte Bucket tape drive and SL-803-A serial I/O device both require specific control strings, and all data transfer operations require allocated memory space.

Control of the Byte Bucket is accomplished by transmitting to it ASCII character strings which are explicitly defined so that they may be accessed by name within the definition of a FORTH word. This makes the job of the programmer easier and makes the FORTH code more readable. The following eight commands are given by sending the appropriate one or two character strings to the Byte Bucket (definitions of the strings are in the Appendix).

(1) enter play mode

(2) enter record mode
(3) exit play mode  
(4) exit record mode  
(5) rewind to beginning of tape  
(6) skip to beginning of next file  
(7) rewind to beginning of current file  
(8) request status word

The SL-803-A also has control strings to define the mode of operation as well as those to tell it what data to transmit and when to transmit it. These control strings consist of a unit select control code character followed by a sequence of characters that define some operating parameter. The control code character can be any ASCII character that is not part of a defining string. It is selected by the setting of a 8-pin DIP switch on the back of the SL-803-A. The FORTH name of each string defined for the SL-803-A and the function of each is given below.

(1) @DLY - Define turnaround delay  
(2) @F03 - Specify two analog channels  
(3) @H1 - Enable channel A  
(4) @V - Specify ±2v A/D conversion  
(5) @8 - Read all enabled analog channels  
(6) @A - Read digital data

In addition to the control and initialization strings, some strings are allocated space but the content of the string space is undefined. These strings are used as holding areas for transfer of
data to and from tape. Also stored in memory are the ASCII character values for the names of the device drivers and for the default disk drive "SY0:." These character values must be in memory to open a device driver from FORTH using the word SCALL.

2. Device Driver Words. To communicate with a device driver from FORTH, the FORTH word SCALL is executed with the proper codes on the system stack. Coding can become complicated in FORTH when setting up these SCALLs, so a group of words were written that take care of the bookkeeping necessary for communication with a device driver. These words were made versatile enough to work with any device driver.

With these FORTH words, it is much easier to program communication with a device driver, and they make the FORTH code more readable. To open a channel for read (or write) all that need be given is the channel number, the location of the ASCII code for the device driver name, and the command OPEN-READ (OPEN-WRITE). To close a channel, the channel number and the command CLOSE-DD are required. To read from (or write to) a device driver, the user supplies the channel number of the opened driver along with the string space for the data to be read into (written from) and the command READ-DD (WRITE-DD).

3. Specific Device Driver Functions. The next level of FORTH words consists of those that define the specific I/O functions of each device driver used in the system. These are the words that utilize the general device driver words of section two applied to a specific
device driver. They make available the outputs of the measurement devices in a string form compatible with the FORTH environment.

For the Byte Bucket these are the words that use the Byte Bucket control string definitions of section one and the device driver words of section two to construct words such as PLAY, REWIND, RECORD, etc. that cause the Byte Bucket to perform the specified action. Also provided for Byte Bucket I/O is the word "BYTEBUCKET" which writes the contents of the reserved string BB$ to tape. BB$ is filled with all the data required for one data point.

FORTH words are defined to control the operation of the SL-803-A I/O unit. An initialization word (INIT) is provided so that the SL-803-A can be set in the desired mode of operation before data collection starts. Since the SL-803-A device only sends data when requested to do so, words are defined that send the data request string to the unit through the device driver, followed immediately by a read request to the device driver to capture the data as it comes back. This procedure is followed for both the analog and digital outputs of the SL-803-A as defined by the words "ANALOG" and "DISCRETE" respectively.

Both the clock and the Miniranger devices require very little control from FORTH since their respective device drivers are both interrupt driven to configure the data in a location that can easily be transferred to FORTH. Since both the clock and Miniranger device drivers maintain the data in memory, FORTH need only read these characters using READ-DD.
4. Data Collection Loop. This is the "working level" software of the system. Here all of the above functions are put together in a data collection loop that reads data from the various sources, stores it on magnetic tape if storage is enabled (explained later), and provides a display on the CRT for the user. There is a small amount of calculation involved with providing the display, but data are stored on tape exactly as they are received.

The data collection routine starts by initializing several operating flags and the SL-803-A, then clears the CRT screen and enters the collection loop. The loop, provided all functions are enabled, consists of inputting the time, analog EMC-25 data, digital EMC-25 data and Miniranger range data, then storing this data on magnetic tape followed by configuring a display on the CRT and finally checking the keyboard for input before returning to the start of the loop.

Due to the amount of time it takes to calculate the data for a CRT display and the relatively low importance of the CRT display compared to tape operations, there is a counter in the collection loop that enables the display only on every fifth sample. Analysis of data collection loop execution time showed that the CRT display required about 375 milliseconds out of total loop time of 700 milliseconds. Therefore, configuring the CRT display on every fifth sample provides for 75 percent more recorded samples in a given amount of time than does configuring the CRT display with every sample. The CRT display
is provided expressly for purposes of giving the operator a quick qualitative look at the data being collected so displaying only 20 percent of the data is not considered a problem. With the data displayed on the CRT on every sample, the average sample time is 700 milliseconds; with a display every fifth sample, the sample time is 390 milliseconds; and with no display the sample time is 320 milliseconds.

The enabled functions described above allow for the data collection loop to run in several modes. The file being written to the Byte Bucket can be closed and a new file opened, or an open file can be turned on and off. When the Byte Bucket file mode is either off or closed, nothing is stored on tape, the loop only collects data and shows the result on the CRT. In addition to stopping tape storage the entire sampling process can be suspended. When this occurs the loop only checks the keyboard for input while the information that was displayed on screen when the sampling was turned off is "frozen" on the screen.

The process of checking the keyboard for input is done to provide interactive control of the data collection routine. Control of the routine is accomplished by pressing one of eight system-defined function keys. All other keys are software-disabled when in the collection loop. Six of these keys are used to set or clear three flags that provide the conditional enabling discussed in the previous paragraph. One key gives the user the opportunity to change the
descriptive header that is written to tape at the beginning of a data file, and the last key provides for a logical exit from the data collection loop. These eight keys and the function they provide are:

f1 - Turn Byte Bucket on  
f2 - Turn Byte Bucket off  
f3 - Open file on Byte Bucket  
f4 - Close file on Byte Bucket  
f5 - Set file parameters  
Blue - Turn sampling ON  
Red - Turn sampling OFF  
White - Exit from collection loop

The CRT display places on the 25th line the current time, the values of the computed frequency and amplitude from the EMC-25, the attenuator setting of the EMC-25, the distance from the airplane to the ground station and the status of the Byte Bucket. On lines 1 through 24 is a strip chart display of the relative measured amplitude, with the most recent measurement placed on line 24 and the older measurements scrolled up and off the screen. The last 24 measurements sent to the CRT are displayed at all times. The displayed amplitude is a graphics character placed on line 24 at a position determined by the value of the amplitude input string, scaled between 0 and 80 (see figure 4-7).

It should be noted that the frequency and amplitude displayed on line 25 are the result of a calculation based on the input data and
a piecewise linear approximation of the EMC-25's output characteristics. These values may or may not be "true," but they provide a fast approximation to give the user an estimate of what is being measured. However, due to the nearly linear characteristics of the EMC-25 these values reported on line 25 tend to be very close to the real amplitude and frequency (see figures 3-5 to 3-7). The EMC-25 outputs received in data transmission, not the values shown on line 25, are stored on tape to allow the data analysis routines the freedom to compute the frequency and amplitude in any way desired. Data analysis routines explained in Chapter V use the same method to determine amplitude and frequency; however, the accuracy of the calculation is much better in the data analysis routines.

5. User Interface Software. The purpose of this software is to make the operation of the data collection system as simple and self-explanatory as possible, as well as to provide for error checking of parameters input by the operator. The result of this software is a system that displays a menu of options on the CRT (figure 4-8), with a short explanation of the action each option takes, and waits for a command input from the user. All user commands are entered by typing only the first letter of the given command. Any command that may cause harm if executed at the wrong time requires user confirmation. Any input other than one of the defined commands is ignored.

The header for each file stored on tape is a list of parameters that describe the file. These parameters, along with the correct
Antenna - Show frequency and amplitude
Eye - Exit to HDOS - FFORTH to get back
Data - Show raw receiver data
Factor - Check antenna factor
Go - Start data collection
Initialize - The SL-803-A unit (to FORTH input level if MENU not TURNKEY)
Menu - Type this list
Parameters - Check file parameters
Range - Show slant range
Time - Display the time

Figure 4-8. Menu of Options Displayed on CRT.
time, must be input by the user. Before the system begins data
collection, current parameter settings are shown to the user with the
option either to make changes or continue as they are (figure 4-9).
The parameters may also be viewed, with the option of changing, by
entering the "P" command from the menu of user options. The parame-
ters contained in this file are the date, a file name, a descriptor
string for identification of the equipment being measured, and an
altitude. The file name and equipment type are each allotted space
for 80 characters so that any input string up to 80 characters in
length may entered; the altitude is allotted 12 characters. Both the
time and date strings are of fixed form and only the proper form will
be accepted. The time and date input routines prompt the user for
input; if an invalid input is entered they will issue an error message
and prompt for another input until the proper form is given (the
proper form is shown to the user in the prompting string).

The menu options are listed below; the letter in quotes is the
command input and the word to the right gives the function, followed
by a short explanation.

"A" Antenna - Show measured amplitude, frequency, and
attenuation.

"B" Bye - Return to HDOS (Heath Disk Operating System)
command level (requires confirmation).

"F" Factor - Show the antenna factor being used with option
to change.
TIME: 12:34:46
DATE: 15-NOV-83
ALTITUDE: 500

FILE NAME: EM1 DATA, OPEN-FIELD TEST SITE, WATERMAN, ILLINOIS

EQUIPMENT: MODEL A, 5 KW RF POWER, ALL SHIELDING IN PLACE

SYSTEM PARAMETERS -- (RETURN) TO START DATA COLLECTION, ANY OTHER
KEY TO SET NEW PARAMETERS. (CTRL-C ABORTS)

Figure 4-9. System Parameter Settings Displayed on CRT.
"G" Go - Start data collection.

"I" Initialize - Initialize devices if MENU TURNKEYed (see below).

"M" Menu - Type this menu.

"P" Parameters - Show file parameters with option to change.

"R" Range - Show Miniranger slant range.

"T" Time - Display the time.

To simplify usage even further the system can be made self-loading with automatic execution of the above menu. To operate the system, all four device drivers (CK:, SL:, MR:, BB:) must be loaded at HDOS command level and then the configured FORTH file executed from command level. Once executing in FORTH, the entire system is started by executing the word MENU. To make loading easier, an assembly language routine called LOADDVD was written that loads all four device drivers and then links to the compiled FORTH file. The assembled code from LOADDVD can be assigned the file name "PROLOGUE.SYS" so that it is executed when the disk is initially loaded. In addition to this, the FORTH word "MENU" can be TURNKEYed so that it is executed when FORTH is entered. Therefore, if the disk being used contains LOADDVD as PROLOGUE.SYS and MENU is TURNKEYed in the compiled FORTH file then all that the user needs to do is load the disk and follow the directions.

D. OPERATING THE DATA COLLECTION SYSTEM. The required components for operating this data collection system are 1) an H89 com-
puter, 2) a Motorola Miniranger III and Ground Telemetry Processor, 3) an Electro-Metrics EMC-25 interference analyzer, 4) a Serial Lab Products SL-803-A remote serial I/O device, 5) a Byte Bucket magnetic tape player/recorder and 6) the software developed to control these devices.

All data communication connections between devices are accomplished by making RS-232 connections to the proper device ports. The RS-232 interconnection diagram is shown in figure 4-10. This diagram does not show any power connections or other system data connections associated with normal use of the Miniranger III and Ground Telemetry Processor. The switch-selected operating mode of each device is to be set in the following way:

SL-803-A: 4800 Baud, control character=hexadecimal 26
Byte Bucket: 9600 Baud, remote mode ON, verify mode OFF, parity OFF (verify mode may be ON if desired)
Miniranger: Set for normal operation, data link set for 4800 Baud.
GTP: Auxiliary data port set for 4800 Baud.

The minimum software required for operation consists of the four device drivers; BB.DVD, CK.DVD, MR.DVD, and SL.DVD; plus the compiled FORTH code written to operate this system. It is not necessary to have the routine LOADDVD, but it makes the loading easier and does not occupy RAM space after execution, so it does not detract from system performance in any way.
Figure 4-10. RS-232 Interconnection Diagram.
When the complete system is used with LOADDVD as PROLOGUE and MENU TURNKEYed, the operation of the system is simple. The user makes sure that the switch settings on the Byte Bucket and SL-803-A are correct and that power is applied to all devices, and then loads the disk. The system will load itself and come up in the menu input mode, showing the available commands and waiting for input.
V. DATA EVALUATION

Raw data acquired by the system while flying over ISM sites must be transformed into a more convenient form to be evaluated. This transformation must create distinguishable forms of data output from the lists of numbers recorded during flight. FORTRAN programs have been written for the Ohio University 370 computer system that create plots of E-field levels versus distance from the unit under test. These plots are a data form which can easily be interpreted to determine E-field levels at a certain point or group of points.

A. PLOT PROCEDURE. Data are transferred from the Byte Bucket digital cassette tapes to the IBM 370 computer system so that the computation and plot facilities of this system can be utilized. A Calcomp plot package [27] exists on the IBM 370 system; this plot package provides the resources for development of FORTRAN plotting programs to output the data. Data transfer is accomplished by use of a routine called FREACH, written for use on the Heath H89 computer. This routine executes the proper H89 port control to transfer data from an external source to an H89 system disk and from an H89 system disk to the IBM 370 system.

Data that has been uplinked to the IBM 370 system are in a form that can be used by FORTRAN programs. These data files contain points in which one or more system parameters are recorded as zero if the Miniranger data link or GTP output of range failed at any time during data collection. These points need to be eliminated from the data.
file because they represent impossible states. A FORTRAN program
(DATAFLTR) was written to eliminate these points.

The FORTRAN subroutine RFSIG calculates the amplitude and fre-
quency from receiver signals. Calculations in RFSIG are based on data
from the EMC-25/SLP-803-A calibration presented in Chapter III. The
calibration data used in RFSIG differs from the calibration data in
Chapter III only by the accuracy of its entry. Data shown in Chapter
III and used in the data collection software to calculate values for a
CRT display were rounded off so that integer numbers could be entered
into the FORTH software; the data entered into RFSIG were not rounded
off.

The formula used in RFSIG to calculate the EMC-25 meter reading
from the amplitude signal is given in equation 5.1. Meter reading is
calculated in dB\mu V; therefore, measured signal strength in dB\mu V/m is
obtained by adding the meter reading, attenuator setting, and antenna
factor as given in equation 5.2. Attenuator setting used in equation
5.2 is calculated by RFSIG from the receiver code recorded as a
hexadecimal byte. Antenna factor is a constant supplied by the
calling program; this constant is the measured antenna factor for the
antenna used. Frequency is calculated by RFSIG using the same
piecewise-linear approximations as shown in figures 3-6 and 3-7,
except that data in RFSIG are not rounded to the nearest integer as is
done for these figures.

\[
E_{\text{meter}} = (V_{\text{ampl}} \times 42.17 \text{ dB\mu V/Volt}) - 46.63 \text{ dB\mu V} \tag{5.1}
\]
where: \( E_{\text{meter}} = \) EMC-25 meter reading

\( V_{\text{ampl}} = \) Amplitude signal voltage

\[ E\text{-field} = E_{\text{meter}} + \text{Attenuator} + \text{Antenna Factor} \]  

where: \( E\text{-field} = \) Electric field strength in dBμV/m

To plot the data a FORTRAN program must be written that reads the raw data, calls RFSIG to calculate the E-field and frequency, processes the range in any way desired and calls the Calcomp plot routines to plot the E-field levels versus distance. An example of this is the routine EFLDPLLOT listed in the Appendix. EFLDPLLOT calls the plot routines to plot E-field levels versus ground distance from the test site. This routine also uses the Calcomp plot facilities to print information regarding data collection setup and signal frequency. EFLDPLLOT also plots the FCC and CISPR limits for E-field radiation at the measured signal frequency; these limits are useful as references for comparing plots.

Plotting the data on graphs reduces the bulk of information presented while increasing intelligibility. The time of measurement, although helpful to assure proper operation, is not important to signal evaluation and therefore may be discarded. Also, the frequency setting of the receiver need not be listed for each data point since it is nearly constant throughout the entire flight. Test flight data show that the frequency setting signal varies during flight by only 2 to 3 millivolts around an average value; this reflects the stability of the A/D converter used for signal processing. The mean frequency
value is printed on the output plot to indicate the receiver frequency setting.

B. WATERMAN, ILLINOIS DATA COLLECTION. During the week of October 11-15, 1983, measurements were taken of the E-fields produced by four ISM units at an FCC-certified open-field test site in Waterman, Illinois. These measurements were taken in cooperation with the Elite Electronics Engineering Company [28] and the manufacturer of the ISM equipment. Measurements of the E-fields produced by this equipment were obtained by three methods: E-fields were recorded using the automated data collection system while flying over the test site; measurements were taken by the Elite Electronics Engineering Company according to FCC rules and regulations; and E-fields were measured using the EMCO 3104 biconical antenna mounted on top of a tower.

The open-field test site at Waterman, Illinois consists of a wooden building with a minimal amount of metallic structure near the area containing the unit under test. The building contained all of the resources required for operation of the ISM equipment in a mode simulating a production environment; these resources consisted of electrical power, pressurized water, and pressurized air. The unit under test was placed on a wooden turntable inside the building so that the unit could be tested at different azimuths while ground-based equipment remained stationary.

Airborne data collection was conducted for the four ISM units by flying the airplane over the test site in north-to-south passes.
E-field measurements for different machine orientations were obtained by turning the equipment on the turntable. Airborne measurements were conducted only at the fourth harmonic ISM frequency (108.48 MHz) because an antenna of sufficient sensitivity at the ISM frequency (27.12 MHz) was not available at that time. The data collection flights were performed at altitudes of 500 and 1500 feet above ground level for each unit; two of the four units were tested at three different azimuths, one unit was tested at five azimuths, and the fourth unit was tested at only one azimuth. A total of 29 data collection flights were performed with approximately 1200 data points recorded.

The airborne data collection system used at Waterman, Illinois was an earlier version of the one described in the preceding chapters. This system used a TI9900 Loran-C navigation receiver [29] to determine the aircraft position during data collection. The TI9900 outputs receiver information on an RS-232 data connection; this information contains the aircraft latitude and longitude, and the range and bearing from a designated point. During data collection the location of the open-field test site was entered into the receiver as the "to waypoint" so that range and bearing angle outputs were referenced to the test site. Output of the TI9900 was received by a H89 device driver and the latitude, longitude, range and bearing angle were stored in memory. These receiver parameters were input by the FORTH controlling software and stored on tape. The data collection system configuration used at Waterman, Illinois (see figure 5-1) did not utilize the Miniranger system; the TI9900 data output was connected to
Figure 5-1. Block Diagram of Data Collection System used for Waterman Illinois Data Collection Flights.
the computer port to which the GTP auxiliary data port was later connected (port 340) and all data collection equipment was carried inside the airplane. There are three major areas that made this data collection system less effective than the present system 1) accuracy of the range measurement; 2) ease of operation; and 3) electrical power requirements. The TI9900 provided measurements of range rounded to the nearest tenth of a nautical mile (607.6 feet); this range was updated approximately every twelve seconds [30]. Although interpolation of the range between updates improved the accuracy at individual data points, accuracies better than 100 feet could not be obtained from the TI9900 output. Since there was no telemetry data link with this system, all data collection equipment had to be operated from inside the airplane. Operation of both the EMC-25 receiver and the H89 computer were accomplished by one person riding in the airplane; however, this type of activity is prone to operator error as evidenced by several missed data files at Waterman. The environment of the airplane is not conducive to complicated operation of equipment, so the operator cannot easily operate both the EMC-25 receiver and control the operation of the data collection system. Operating the H89 computer on electrical power supplied by the square-wave inverters in the airplane degraded the computer's performance. During data collection flights the CRT display of the H89 computer intermittently went blank; this prohibited the operator from seeing data being collected. The source of CRT malfunction was probably due to the nature of the supplied electrical power; if so, this problem
has since been corrected by the installation of sine-wave inverters in the aircraft.

Data from Waterman was uplinked to the Ohio University IBM 370 computer system and reduced to plot form by FORTRAN programs. The routine RBINT was used to interpolate between TI9900 range and bearing points; this routine calculated the x and y position of the aircraft relative to the test site. Subroutine RFSIG was used to calculate the E-field levels and frequency, and EFLDPLT created plots of the data by utilizing the Calcomp plot package. Output plots, shown in figures 5-2 through 5-23 were then used to evaluate the emissions of the four machines tested. The plots display the E-field magnitude in dBµV/m as the ordinate and the horizontal distance from the test site as the abscissa. Horizontal distance from the test site represents the distance from a point on the ground directly beneath the aircraft to the test site. FCC and CISPR E-field limits are also shown on these data plots for use as references.

Plots of flight data from Waterman are shown in figures 5-2 through 5-23. All four machines tested at Waterman were dielectric sealers used to seal vinyl or other similar material. Machines B and D produced 2 kW; machine C produced 3 kW; and machine A produced 25 kW of RF output power. These machines are referred to as A, B, C and D to preserve the anonymity of the manufacturer.

Effects of RFI (Radio Frequency Interference) Shielding and ISM unit azimuth can be seen by comparing plots of data taken under the
Figure 5-2. Flight Data Machine A, 25kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.
Figure 5-3. Flight Data Machine A, 25 kW, 457 M Altitude, 180° Azimuth, RFI Shields Removed.
Figure 5-4. Flight Data Machine, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.
Figure 5-5. Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields Removed.
Figure 5-6. Flight Data Machine A, 25 kW, 152 M Altitude, 180° Azimuth, RFI Shields in Place.
Figure 5-7. Flight Data Machine B, 2 kW, 152 M Altitude, 300° Azimuth, RFI Shields in Place.
Figure 5-8. Flight Data Machine B, 2 kW, 457 M Altitude, 300° Azimuth, RFI Shields in Place.
Figure 5-9. Flight Data Machine B, 2 kW, 457 M Altitude, 0° Azimuth, RFI Shields in Place.
Figure 5-10. Flight Data Machine B, 2 kW, 457 M Altitude, 240° Azimuth, RFI Shields in Place.
Figure 5-11. Flight Data Machine B, 2 kW, 152 M Altitude, 240° Azimuth, RFI Shields in Place.
Figure 5-12. Flight Data Machine B, 2 kW, 152 M Altitude, 0° Azimuth, RFI Shields in Place.
Figure 5-13. Flight Data Machine C, 3 kW, 152 M Altitude, 320° Azimuth, RFI Shields in Place.
Figure 5-14. Flight Data Machine C, 3 kW, 152 M Altitude, 20° Azimuth, RFI Shields in Place.
Figure 5-15. Flight Data Machine C, 3 kW, 152 M Altitude, 260° Azimuth, RFI Shields in Place.
Figure 5-16. Flight Data Machine C, 3 kW, 152 M Altitude, 260° Azimuth, RFI Shields in Place.
Figure 5-17. Flight Data Machine C, 3 kW, 152 M Altitude, 230° Azimuth, RFI Shields Removed.
Figure 5-18. Flight Data Machine C, 3 kW, 152 M Altitude, 20° Azimuth, RFI Shields Removed.
Figure 5-19. Flight Data Machine D, 2 kW, 152 M Altitude, 230° Azimuth, RFI Shields in Place.
Figure 5-20. Flight Data Machine D, 2 kW, 152 M Altitude, 20° Azimuth, RFI Shields in Place.
Figure 5-21. Flight Data Machine D, 2 kW, 152 M Altitude, 260° Azimuth, RFI Shields in Place.
Figure 5-22. Flight Data Machine D, 2 kW, 152 M Altitude, 200° Azimuth, RFI Shields in Place.
Figure 5-23. Flight Data Machine D, 2 kW, 152 M Altitude, 60° Azimuth, RFI Shields in Place.
different conditions. Figure 5-6 shows data from machine A with all RFI shielding properly installed; figure 5-2 is from the same machine with the RFI shielding for the oscillator circuitry removed; and figures 5-4 and 5-5 show E-field levels with all RFI shielding removed. Analysis of these plots shows that removal of the oscillator circuitry shielding increased the signal level in space by approximately 5 dB; removal of all RFI shielding resulted in E-field levels as much as 25 dB higher.

Plots for machine B are shown in figures 5-7 through 5-12. Figures 5-7 and 5-8 can be compared to see the effects of changing the altitude for data collection. For this comparison the location of the data relative to the FCC and CISPR limits is useful; as expected from wave propagation theory, this relative measurement is nearly the same for the two altitudes. Effects of changing the azimuth of this unit can be seen by comparing figures 5-7, 5-11, and 5-12; this machine exhibited less than 5 dB change in signal amplitude for these three azimuths.

Machine C was tested at three azimuths both with RFI shields in place and with shields removed; these plots are shown in figures 5-13 through 5-18. Machine C exhibited as much as 20 dB change in signal level with change in azimuth (figures 5-13 and 5-14); and no more than 7 dB change with removal of shielding (figures 5-14 and 5-18).

Data collected for machine D represents the E-field for this machine at 5 different azimuths. All data collected for machine D
were taken at an altitude of 152 meters with all RFI shielding in place. E-field levels measured for machine D differ by less than 10 dB; this difference exists between plots of 200 degree and 320 degree azimuths (figures 5-19 and 5-22).

Ground-based measurements were also conducted at Waterman, Illinois according to FCC and CISPR certification procedures [31]. All four ISM units met or exceeded FCC emission standards based on the ground measurements; two of the units met CISPR standards. However, based on airborne data none of the four units met either FCC or CISPR limits; this appears to be due to RF absorption for low elevation angles and the E-field boundary conditions for receiving antennas close to a ground plane [32]. Table 5.1 lists data from both ground-based and airborne data collection; the column on the right side of this table lists the difference between ground-based and airborne data. Table 5.1 shows that in all cases the airborne measurements exceeded the ground-based measurements; the airborne data were a minimum of 16.9 dB above ground-based data (machine B at 0 degree azimuth) and a maximum of 42.9 dB above ground-based data (machine D at 200 degree azimuth).

C. ILS LOCALIZER DATA. On March 7, 1984 the E-fields produced by the ILS localizer antenna at the Albany, Ohio airport were measured using the data collection system. This data collection provided measurements of the localizer antenna output in absolute dBμV/m for the area directly in front of the antenna array. Data collected at
Ground-based Data versus Airborne Data
(dB relative to FCC limit of 20 dBμV/m at one mile)

<table>
<thead>
<tr>
<th>Machine A</th>
<th>Azimuth</th>
<th>Ground Data</th>
<th>Airborne Data</th>
<th>Airborne-Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>-36.4</td>
<td>+5</td>
<td>41.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine B</th>
<th>Azimuth</th>
<th>Ground Data</th>
<th>Airborne Data</th>
<th>Airborne-Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-8.9</td>
<td>+8</td>
<td>16.9</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>-14.1</td>
<td>+7</td>
<td>21.1</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>-14.7</td>
<td>+6</td>
<td>20.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine C</th>
<th>Azimuth</th>
<th>Ground Data</th>
<th>Airborne Data</th>
<th>Airborne-Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>-39.5</td>
<td>-14</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>-35.9</td>
<td>-8</td>
<td>27.9</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>-35.0</td>
<td>+5</td>
<td>40.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine D</th>
<th>Azimuth</th>
<th>Ground Data</th>
<th>Airborne Data</th>
<th>Airborne-Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>-40.8</td>
<td>-5</td>
<td>35.8</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>-43.3</td>
<td>-7</td>
<td>36.3</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>-37.9</td>
<td>+5</td>
<td>42.9</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>-37.2</td>
<td>+1</td>
<td>38.2</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>-34.5</td>
<td>+2</td>
<td>36.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1
Albany, Ohio was uplinked to the Ohio University IBM 370 computer system and reduced using the FORTRAN subroutine RFSIG and a main program very similar to EFLDPLCT. Five flight tests were conducted at Albany, Ohio; data plots of these flight tests are shown in figures 5-24 through 5-28. Figure 5-29 is a combination of plots 5-24 through 5-28; this plot gives a more comprehensive picture of the E-field levels produced by the localizer. All of the data plots show nearly the same E-field level at common points; figure 5-29 demonstrates this by its continuity. The continuity of the plot in figure 5-29 also demonstrates the repeatability of the measurement system.

Measurement of the ILS localizer allows direct comparison between ISM radiation and localizer signal levels. In [33] a comparison of these localizer signal levels with ISM signal levels was performed. This comparison shows that ISM units which pass ground-based FCC certification tests may produce signal levels 2.8 dB higher than ILS localizer signals. This comparison is based on the findings at Waterman which indicate E-fields above ISM units are at least 16.9 dB higher than those detected using FCC procedures. Units that meet CISPR standards would produce signal levels in space 32.4 dB lower than the ILS localizer. The ILS susceptibility report cited in Chapter II concluded that signal-to-interference ratios of at least 20 dB must be maintained for dependable ILS operation; therefore, ISM units that pass FCC certification could potentially interfere with ILS receiver operation.
Figure 5-24. Albany, Ohio Localizer Signal Flight 1.
Figure 5-25. Albany, Ohio Localizer Signal Flight 2.
Figure 5-26. Albany, Ohio Localizer Signal Flight 3.
Figure 5-27. Albany, Ohio Localizer Signal Flight 4.
Figure 5-28. Albany, Ohio Localizer Signal Flight 5.
Figure 5-29. Albany, Ohio Localizer Signal Flights 1-5.
VI. CONCLUSIONS

The goal of this work was to engineer a data collection system that was capable of measuring data points which permit the analysis of E-fields produced by ISM equipment; this has been accomplished. The computer-controlled system provides the necessary update rate, E-field measurement accuracy and distance measurement accuracy to permit analysis of ISM emissions.

Time between samples for this system is 390 milliseconds; this provides for a new electric field measurement approximately every 20 meters in flight. Data from flight tests at Waterman, Illinois (see figures 5-2 through 5-23) shown that narrow vertical lobes of radiation emitted by the ISM units were detected by the system. ISM machines used at Waterman exhibited narrow vertical lobing directly overhead; this is consistently shown in the data plots.

Range measurement accuracy of the Miniranger with custom Ground Telemetry Processor far exceed system requirements; this system is capable of range measurements within ±2 meters. In normal operation, the distance from the ISM unit to the airplane will not be less than 150 meters; therefore a 2 meter range error represents only 1.33 percent. A range error of 1.33 percent represents a 0.23 decibel change in E-field; this amount of change is not reliably detectable by most RF receivers, including the EMC-25 used for these measurements. Repeatability of measurement data from the Albany, Ohio ILS localizer provides evidence that the system functions appropriately.
This system requires two operators in addition to the pilot; one operator rides in the aircraft and operates the EMC-25 receiver; the other operator controls the computerized data collection from the keyboard of the H89 computer on the ground. The entire ground-based system can be operated inside a van; all test equipment can be transported to the test site in this same vehicle, or in a light aircraft.

This data collection system is not limited to collection of EMI data; the architecture of the collection system software makes it conducive to modifications which result in sensitivity to different input values. The structure of the data collection loop would be nearly the same for any data desired; each set of data collected would have its own FORTH words that define how the data is read in. Therefore, to alter the system to collect different data entails changing only those FORTH words, not the entire collection structure. Future uses now planned include Loran-C navigational system measurements.

Outputs of data collected by this system show evidence that E-field emissions at high elevation angles from ISM units are much greater than those measured from the ground. E-fields measured above ISM units ranged from 16.9 - 42.9 dB above ground-based measurements. All of these measurements were taken at an open-field test site; measurements from actual production sites could help validate these measurements, but the opportunity to perform actual site measurements did not exist. However, measurements of E-fields produced by the
localizer at Albany, Ohio provide a direct comparison between localizer signals and ISM signals. Comparisons of these localizer signals to signals produced by ISM equipment show there is a very real possibility that ISM equipment does produce E-field emissions large enough to interfere with ILS receiver operation.

The data collection system presented in this paper provides the capability to measure free-space electric field strength in absolute terms. This capability is essential for measuring emissions at high elevation angles from ISM equipment; it is desirable to measure emissions at high elevation angles since measurements made at ground level fail to represent accurately field strengths at higher elevation angles. In regards to ILS navigation systems, field strength levels at high elevation angles are the only measurements that are valid since localizer receivers are used when the aircraft is in the air.

The data obtained by this data collection system indicates that a potential interference problem could result when ILS localizer receivers are operated near a piece of ISM equipment. No attempt has been made to characterize fully the emissions of ISM equipment or prove that ISM equipment does interfere with the safe operation of ILS localizer receivers. However, the data collection system presented in this thesis does provide the capability to conduct studies in these areas.
VII. ACKNOWLEDGEMENTS

The author would like to thank the following persons: Dr. Robert Lilley for his guidance in preparation of this thesis and his skill as the pilot of the aircraft during flight tests; Mr. James Nickum, project engineer, for his technical expertise and his assistance in operating the system; Mr. Richard Zoulek who completed the mechanical details of the flight instrumentation package. Finally, I would like to thank Dr. Richard McFarland and the entire Avionics Engineering Center staff for allowing me to be a part of that fine organization, and especially Alicya Shade for her many hours of dedicated word processing.
VIII. REFERENCES


[12] Ibid. pp. 3-3 - 3-9.


[20] Ibid., pp. 7-8.


[27] "Functional Description of the FORTRAN Subroutine Package CALCOMP," Ohio University Computer Center, Ohio University, Athens, Ohio.


A. Data Collection Software - FORTH Listings

B. Miniranger Device Driver Assembler Listing
APPENDIX A

DATA COLLECTION SOFTWARE—

FORTH LISTINGS
SCREEN NUMBER 1
0 ( SLP COMMAND DEFINITIONS)
1
2
3 2 STRING @8 " &8" @8 S! READ ALL CHAN
4 5 STRING @DLY " @H01" @DLY S! SPL DELAY
5 3 STRING @H1 " &H1" @H1 S! ENABLE CHAN A
6 4 STRING @F03 " &F03" @F03 S! 2 ANALOG CHAN
7 2 STRING @A " &A" @A S! READ DIGITAL DATA
8 2 STRING @V " &V" @V S! + - 2 VOLTS
9
10 CREATE DFLT 83 C, 89 C, 48 C, 0 C, 0 C, 0 C,
11 CREATE BB: 66 C, 66 C, 58 C, 0 C,
12 CREATE CK: 67 C, 75 C, 58 C, 0 C,
13 CREATE MR: 77 C, 82 C, 58 C, 0 C,
14 CREATE SL: 83 C, 76 C, 58 C, 0 C,
15 CREATE DVDFLG 10 ALLOT

SCREEN NUMBER 2
0 ( EMI DEFINITIONS CONTINUED)
1 10 STRING T$10 T$ DROP 1- C! TIME STRING
2 9 STRING T1$
3 17 STRING A1$ ANALOG TRUNCATED STRING
4 19 STRING A$ 19 A$ DROP 1- C! ANALOG INPUT STRING
5 4 STRING D$ 4 D$ DROP 1- C! DISCRETE INPUT STRING
6 2 STRING D1$
7 5 STRING R1$ 5 R1$ DROP 1- C! RANGE STRING
8 1 STRING CLEAR$ 1 CLEAR$ DROP 1- C! CLEAR SLP STRING
9
10 15 STRING SLPINIT$
11 9 @DLY SLPINIT$ S+
12 @H1 SLPINIT$ S+
13 @F03 SLPINIT$ S+
14 @V SLPINIT$ S+
15 80 STRING BB$ -->

SCREEN NUMBER 3
0 ( EMI DEFINITIONS JIM NICKUM)
1 CREATE PLAY$ 1 C, 17 C,
2 CREATE RECORD$ 1 C, 18 C,
3 CREATE PLAY-OFF$ 1 C, 19 C,
4 CREATE RECORD-OFF$ 1 C, 20 C,
5 CREATE BOT$ 2 C, 16 C, 49 C,
6 CREATE SKIP$ 2 C, 16 C, 50 C,
7 CREATE REWIND$ 2 C, 16 C, 51 C,
8 CREATE STATUS$ 2 C, 16 C, 52 C,
9 CREATE LFCR$ 1 C, 10 C,
10
11 0 VARIABLE FILENO.
12 ;S
13
14
15

01MAR84WBD
SCREEN NUMBER 4

0 ( DEVICE DRIVER DEFINITIONS) 06FEB83JDN

1

2 ( N1 N2 --- )
3 ( N1 = CHANNEL #, N2 = DEVICE DRIVER NAME EG: [ CK: ] )

4 : OPEN-READ
5 SWAP DFLT SWAP 0 SWAP 1 34 SCALL DROP ;
6
7

8 ( N1 STRING --- )
9 ( N1 = CHANNEL #, STRING = STRING TO BE READ TO )

10 : OPEN-DD
11 >R >R >R 0 >R SWAP R> SWAP 1 4 SCALL DROP ;
12
13
14 -->
15

SCREEN NUMBER 5

0 ( N1 --- ) 21JAN83JDN
1 ( N1 = CHANNEL # )

2 : CLOSE-DD
3 0 0 ROT 0 SWAP 1 38 SCALL DROP ;
4
5 ( N1 N2 --- )
6 ( N1 = CHANNEL #, N2 = ADDR DEVICE DRIVER NAME EG: CK:) 

7 : OPEN-WRITE
8 SWAP DFLT SWAP 0 SWAP 1 35 SCALL DROP ;
9
10
11
12
13
14 -->
15

SCREEN NUMBER 6

0 06FEB83JN

1 ( N1 STRING --- )
2 ( N1 = CHANNEL #, STRING = STRING TO BE WRITTEN )
3 : WRITE >R >R >R 0 >R SWAP R> SWAP 1 5 SCALL DROP ;
4
5 : INIT 3 SLPINIT$ WRITE ;
6
7 : CLEARSL 2 CLEAR$ READ-DD ;
8
9 : ANALOG 2 A$ 3 @8 WRITE READ-DD A$ 17 LEFT$ A1$ S!
10 A$ 2- + C@ 10 = NOT IF CLEARSL THEN ;
11
12 : DISCRETE 2 D$ 3 @A WRITE READ-DD D$ 2 LEFT$ D1$ S!
13 D$ 2- + C@ 10 = NOT IF CLEARSL THEN ;
14 : TIME 4 T$ READ-DD T$ 8 LEFT$ T1$ S! ;
15 -->
SCREEN NUMBER 7
0 ( DEVICE DRIVER JIM NICKUM )
1 : PLAY 0 PLAY$ COUNT WRITE ;
2 : RECORD 0 RECORD$ COUNT WRITE ;
3 : PLAY-OFF 0 PLAY-OFF$ COUNT WRITE ;
4 : RECORD-OFF 0 RECORD-OFF$ COUNT WRITE ;
5 : BOT 0 BOT$ COUNT WRITE ;
6 : SKIP 0 SKIP$ COUNT WRITE ;
7 : REWIND 0 REWIND$ COUNT WRITE ;
8 : STATUS 0 STATUS$ COUNT WRITE ;
9
10
11
12
13
14
15 -->

SCREEN NUMBER 8
0 ( DEVICE DRIVER JIM NICKUM )
1
2
3 : RANGE 1 R1$ READ-DD ;
4
5 : OPENALL 0 BB: OPEN-WRITE 1 MR: OPEN-READ 2 SL: OPEN-READ
6
7
8
9
10
11
12
13
14
15 -->

SCREEN NUMBER 9
0 ( DEVICE DRIVERS J. NICKUM)
1 20 STRING TEMP$
2
3 : +TST SWAP DUP C@ 43 =
4
5
6
7 -->
8
9
10
11
12
13
14
15
SCREEN NUMBER 10
0 ( DEVICE DRIVER J NICKUM) 06APR83WBD
1 : CLOSEALL 0 CLOSE-DD 1 CLOSE-DD 2 CLOSE-DD 3 CLOSE-DD
2 4 CLOSE-DD ;
3
4 VARIABLE BBONFLG
5 VARIABLE BBOPENFLG
6 VARIABLE SAMPLEFLG
7
8 : BBON? BBONFLG @ ;
9
10 : BBOPEN? BBOPENFLG @ ;
11
12 : SAMPLEON? SAMPLEFLG @ ;
13
14
15 ;S

SCREEN NUMBER 11
0 ( TIME SET ROUTINE) 12MAR83WBD
1 This routine requests the input of the correct time and
2 writes it to the clock. The current time may be used by
3 hitting <RETURN> when input is requested. A flag is put
4 on the stack on exit to indicate no trouble if true and
5 invalid entry if false.
6
7 10 STRING TIME$
8
9 : BADDATA BELL CR " INVALID INPUT- RETYPE YOUR INPUT" CR ;
10
11 : WRTTIME TIME$ 1+ 1+ TIME$ S! 4 CLOSE-DD 4 CK: OPEN-WRITE
12 4 TIMES$ WRITE 4 CLOSE-DD 4 CK: OPEN-READ ;
13 -->
14
15

SCREEN NUMBER 12
0 ( TIME SET ROUTINES) 12MAR83WBD
1 : SECCHK TIME$ 7 2 MID$ VAL 2DUP 60. D<
2 IF 0. D<
3 IF 0
4 ELSE 1
5 THEN
6 ELSE 2DROP 0
7 THEN ;
8
9 : MINCHK TIME$ 4 2 MID$ VAL 2DUP 60. D<
10 IF 0. D<
11 IF 0
12 ELSE 1
13 THEN
14 ELSE 2DROP 0
15 THEN ; -->
SCREEN NUMBER 13
0 ( TIME SET ROUTINES)
1
2 : HRCHK TIMES 2 LEFT$ VAL 2DUP 24. D<
3       IF 0. D<
4       IF 0
5       ELSE 1
6       THEN
7             ELSE 2DROP 0
8       THEN ;
9
10 -->
11
12
13
14
15

SCREEN NUMBER 14
0 ( TIME SET ROUTINES)
1
2 : GETTIME 7 GET$ TIMES 0 HRCHK
3       IF
4       ELSE 1+
5       THEN
6       TIME$ 3 1 MID$ " : " S=
7       IF
8       ELSE 1+
9       THEN
10 MINCHK
11       IF
12       ELSE 1+
13       THEN
14       TIME$ 6 1 MID$ " : " S=
15

SCREEN NUMBER 15
0 ( TIME SET ROUTINES)
1       IF
2       ELSE 1+
3       THEN
4 SECCHK
5       IF
6       ELSE 1+
7       THEN
8
9       IF
10       BADDATA 0
11       ELSE
12       WRTTIME 1
13       THEN ;
14
15 -->
SCREEN NUMBER 16
0 ( TIME SET Routines) 12MAR83WBD
1
2 : INPTIME 1 GET$ TIME$ S! TIME$ DUP 0=
3     IF
4     2DROP 1
5     ELSE
6     GETTIME
7     THEN ;
8
9 : SETTIME CR CR ." INPUT TIME: TIME IS "
10     HERV TIME Ti$ TYPE HXRV
11     ." ? <RETURN IF CORRECT>"
12 INPTIME CR
13 -->

SCREEN NUMBER 17
0 ( ROUTINE TO SET DATE)
1 9 STRING DATE$
2 9 STRING DATE1$
3 3 12 STRING-ARRAY MONS " JAN" 0 MONS S! " FEB" 1 MONS S!
4 " MAR" 2 MONS S! " APR" 3 MONS S! " MAY" 4 MONS S!
5 " JUN" 5 MONS S! " JUL" 6 MONS S! " AUG" 7 MONS S!
6 " SEP" 8 MONS S! " OCT" 9 MONS S! " NOV" 10 MONS S!
7 " DEC" 11 MONS S!
8
9 : YEARCHK DATE1$ 8 2 MID$ VAL 2DUP 100. D<
10     IF 0. D<
11     IF 0
12     ELSE 1
13     THEN
14     ELSE 2DROP 0
15     THEN ; -->

SCREEN NUMBER 18
0 ( ROUTINE TO SET DATE)
1 : DAYCHK DATE1$ 1 2 MID$ VAL 2DUP 32. D<
2     IF 0. D<
3     IF 0
4     ELSE 1
5     THEN
6     ELSE 2DROP 0
7     THEN ;
8
9 : MONTHCHK 0 0 BEGIN
10     DUP MONS DATE1$ 4 3 MID$ S=
11     IF
12     2DROP 1 11
13     THEN
14     1+ DUP 12 =
15     UNTIL DROP ; -->
SCREEN NUMBER 19
0 (SETDATE CON.)
1: GETDATE 8 GET$ DATE1$ S+ 0 DAYCHK
  2: IF
  3:   ELSE 1+
  4: THEN
  5: DATE1$ 3 1 MID$ "-" S=
  6: IF
  7:   ELSE 1+
  8: THEN
  9: MONTHCHK
 10: IF
 11:   ELSE 1+
 12: THEN
 13: DATE1$ 7 1 MID$ "-" S=
 14
 15 -->

SCREEN NUMBER 20
0 (SETDATE CON.)
  1
  2: IF
  3:   ELSE 1+
  4: THEN
  5: YEARCHK
  6: IF
  7:   ELSE 1+
  8: THEN
  9
 10: IF
 11:   BADDATA 0
 12: ELSE
 13:   DATE1$ DATE$ S! 1
 14: THEN ;
 15 -->

SCREEN NUMBER 21
0 (SETDATE CON.)
  1
  2: INPDATE CR ." DATE IS ". HERV DATE$ TYPE HXRV
  3: ." ? ( <RETURN> IF CORRECT, ELSE ENTER DD-MMM-YY)"
  4: 1 GET$ DATE1$ S! DATE1$ DUP 0=
  5: IF
  6:    2DROP 1
  7: ELSE
  8:    GETDATE
  9: THEN CR ;
10
12: SETDATE BEGIN INPDATE UNTIL ;
13 -->
14
15
SCREEN NUMBER 22
0 ( INPUT THE FILE NAME) 04APR83WBD
1
2 80 STRING FILENAME$
3
4 : SETNAME CR ." FILE NAME:" CR HERV FILENAME$ TYPE HXRV CR ." INPUT FILE NAME OR <RETURN> IF OK" CR INPUT$ DUP 0=
8 IF 9 2DROP 10 ELSE 11 FILENAME$ S! 12 THEN CR ; 13
14
15 -->

SCREEN NUMBER 23
0 ( SET THE EQUIPMENT TYPE) 04APR83WBD
1
2 80 STRING EQUIP$
3
4 : SETEQUIP CR ." EQUIPMENT TYPE:" CR HERV EQUIP$ TYPE HXRV CR ." INPUT EQUIPMENT UNDER TEST OR <RETURN> IF OK" CR INPUT$ DUP 0=
8 IF 9 2DROP 10 ELSE 11 EQUIP$ S! 12 THEN CR ; 13
14
15 -->

SCREEN NUMBER 24
0 ( ROUTINE TO INPUT THE ALTITUDE) 01MAR84WBD
1
2 12 STRING ALTITUDE$
3
4 : SETALTITUDE CR ." ALTITUDE: " ALTITUDE$ HERV TYPE HXRV CR ." INPUT ALTITUDE OR <RETURN> IF OK" CR 12 GET$ DUP 0=
8 IF 9 2DROP 10 ELSE 11 ALTITUDE$ S! 12 THEN CR ; 13
14 -->
15
SCREEN NUMBER 25
0 ( ROUTINE TO SET THE AIRCRAFT HEADING )
1 2 3 STRING HEADING$
3
4 : SETHEADING CR ." HEADING: 
5 HEADING$ HERV TYPE HXRV CR
6 ." INPUT AIRCRAFT HEADING OR <RETURN> IF OK"
7 3 GET$ DUP 0=
8 IF
9 2DROP
10 ELSE
11 HEADING$ S!
12 THEN CR ;
13 ;S
14

SCREEN NUMBER 26
0 ( CONSTANTS FOR BAND11 FREQUENCY CALCULATION) 16JUN83WBD
1 2 25 CONSTANT FLO1-11 25 MHZ
3 24 CONSTANT SLOPE1-11 24 MHZ/VOLT
4 202. 2CONSTANT VLO1-11 .202 VOLTS
5 30 CONSTANT FLO2-11 30 MHZ
6 20 CONSTANT SLOPE2-11 20 MHZ/VOLT
7 400. 2CONSTANT VLO2-11 0.4 VOLTS
8 42 CONSTANT FLO3-11 42 MHZ
9 17 CONSTANT SLOPE3-11 17 MHZ/VOLT
10 1000. 2CONSTANT VLO3-11 1.0 VOLTS
11
12 -->
13
14

SCREEN NUMBER 27
0 ( CONSTANTS FOLR BAND13 FREQUENCY CALCULATION) 16JUN83WBD
1 2 100 CONSTANT FLO1-13 100 MHZ
3 113 CONSTANT SLOPE1-13 112.5 MHZ/VOLT
4 115. 2CONSTANT VLO1-13 .115 VOLTS
5 132 CONSTANT FLO2-13 131.5 MHZ
6 78 CONSTANT SLOPE2-13 77.5 MHZ/VOLT
7 400. 2CONSTANT VLO2-13 0.4 VOLTS
8 178 CONSTANT FLO3-13 178 MHZ
9 63 CONSTANT SLOPE3-13 62.7 MHZ/VOLT
10 1000. 2CONSTANT VLO3-13 1.0 VOLTS
11
12 -->
13
14
15
SCREEN NUMBER 28
0 ( COMPUTE FREQUENCY FROM SLP INPUTS) 22APR83WBD
1
2
3
4
5
6
7
8 -->
9
10
11
12
13
14
15

SCREEN NUMBER 29
0 ( DEFINITIONS OF BANDS) 16JUN83WBD
1 : NOBAND DROP 0. 2DUP ;
2
3
4
5
6
7
8
9
10
11
12 -->
13
14
15

SCREEN NUMBER 30
0 ( DEFINITIONS OF BANDS) 16JUN83WBD
1 : BAND11
2   CASE
3      1 OF FLO1-11 SLOPE1-11 VLO1-11 ENDOF
4      2 OF FLO2-11 SLOPE2-11 VLO2-11 ENDOF
5      3 OF FLO3-11 SLOPE3-11 VLO3-11 ENDOF
6    ENDCASE ;
7
8 : BAND13
9   CASE
10      1 OF FLO1-13 SLOPE1-13 VLO1-13 ENDOF
11      2 OF FLO2-13 SLOPE2-13 VLO2-13 ENDOF
12      3 OF FLO3-13 SLOPE3-13 VLO3-13 ENDOF
13  ENDCASE ;
14 -->
15
SCREEN NUMBER 31
0 ( FREQUENCY COMPUTE CONT.)
1 HEX
2 : BANDNO? HEX D1$ VAL DROP 78 AND 8 / DECIMAL
3 DECIMAL
4
5 : BANDLIMS BANDNO? GODO NOBAND NOBAND NOBAND NOBAND NOBAND
6 NOBAND NOBAND NOBAND NOBAND NOBAND BAND11
7 NOBAND BAND13 NOBAND NOBAND THEN;
8
9
10 -->
11
12
13
14
15

SCREEN NUMBER 32
0 ( FREQUENCY COMPUTE CONT.)
1
2 2VARIABLE FVIN
3
4 : FVOLTS A1$ 12 6 MID$ +TST PVAL DROP FVIN 2!;
5
6
7 -->
8
9
10
11
12
13
14
15

SCREEN NUMBER 33
0 ( FREQUENCY COMPUTE CONT.)
1
2 400. 2CONSTANT PNT4
3 1000. 2CONSTANT ONEPNT0
4
5 : BANDSECT FVIN 2@ 2DUP PNT4 D< BANDSECT leaves a 1
6 IF "if FVIN<400., a 2 if
7 2DROP 1 400.<FVIN<1000., and a
8 ELSE 3 otherwise
9 ONEPNT0 D<
10 IF
11 2
12 ELSE
13 3
14 THEN
15 THEN ; -->
SCREEN NUMBER 34
0 ( FREQUENCY COMPUTE CONT.)
1
2 0 VARIABLE FREQUENCY
3
4 ( --- fn ) fn is the frequency input
5 : FREQCOMP FVOLTS BANDSECT BANDLIMS
6 FVIN 2@ 2SWAP -1 D-- D+ DROP
7 1000 */ + FREQUENCY ! ;
8 -->
9
10
11
12
13
14
15

SCREEN NUMBER 35
0 ( EVALUATION OF ATTENUATION INPUT)
1 16JUN83WBD
2 HEX
3 : ANTINPUT HEX D1$ VAL DROP 07 AND DECIMAL ;
4 DECIMAL
5 -->
6
7
8
9
10
11
12
13
14
15

SCREEN NUMBER 36
0 ( EVALUATION OF ATTENUATION INPUT)
1 01MAR84WBD
2 VARIABLE ATTSET
3 : ATTEN ANTINPUT DUP 0= IF
4 DROP -99 ATTSET ! ELSE
5 BANDNO? 11 < IF
6 1 - 20 * ELSE
7 20 THEN
8 THEN ATTSET !
9 THEN ; -->
10
11
12
13
14
15
SCREEN NUMBER 37
0 ( COMPUTATION OF DISPLAYED AMPLITUDE) 29JUN83WBD
1
2 VARIABLE AMPLITUDE
3 42 CONSTANT AMPLMULT  42 DB/VOLT
4 47 CONSTANT AMPLDIFF  47 DB OFFSET
5 VARIABLE ANTFACCTOR antenna factor in DB's
6
7 : AMPLCOMP A1$ 3 6 MID$ +TST PVAL
8 2DROP
9 AMPLMULT 1000 */
10 AMPLDIFF -
11 ANTFACCTOR @ +
12 ATTSET @ +
13 AMPLITUDE  ! ;
14 ;S
15

SCREEN NUMBER 38
0 ( WRITES 25TH LINE WHEN COLLECTING ) 01MAR84WBD
1 : WRITE25 25ON 1 25 HVCUR
2 HERV T1$ TYPE
3 HERV ." RNG:" R1$ TYPE
4 HERV ." AMP:" AMPLITUDE @ . "DB" "MHz" 01MAR84WBD
5 HERV ." ATT:" ATTSET @ . "DB" 01MAR84WBD
6 HERV ." BB:" BBON?
7 IF ." ON "
8 ELSE ." OFF "
9 THEN 01MAR84WBD
10 BBOPEN?
11 IF ." OPEN "
12 ELSE ." CLOSED"
13 THEN 01MAR84WBD
14 HERV ." FILE:" FILENO. @ . 01MAR84WBD
15 HXRV ; -->

SCREEN NUMBER 39
0 ( MARK EOF ON TAPE ) 29JUN83WBD
1 VARIABLE FILES
2
3
4 : MARKFILE RECORD-OFF 0 FILES ! 750 MS RECORD ;
5
6 : POS-AMP AMPLITUDE @ 20 + 80 MIN 1 MAX 24 HVCUR HEGM
7 94 EMIT HXGM ;
8
9 0 VARIABLE SMPLNUM
10
11 -->
12
13
14
15
SCREEN NUMBER 40

0 01MAR84WBD
1 200 STRING LEADER$
2
3 : LEADER FILENO. @ 0 STR$ LEADER$ S! LFCSR$ COUNT LEADER$ S+
4 " DATE: " LEADER$ S+
5 DATE$ LEADER$ S+ LFCSR$ COUNT LEADER$ S+
6 " NAME: " LEADER$ S+
7 FILENAME$ LEADER$ S+ LFCSR$ COUNT LEADER$ S+
8 " EQUIPMENT: " LEADER$ S+
9 EQUIP$ LEADER$ S+ LFCSR$ COUNT LEADER$ S+
10 " ALTITUDE: " LEADER$ S+
11 ALTITUDE$ LEADER$ S+ LFCSR$ COUNT LEADER$ S+
12 " HEADING: " LEADER$ S+
13 HEADING$ LEADER$ S+ LFCSR$ COUNT LEADER$ S+ ;
14
15 -->

SCREEN NUMBER 41

0 01MAR84WBD
1 : BBOPEN BBOPEN?
2 IF
3 BELL BELL bytebucket already open
4 ELSE
5 FILENO. @ 1+ FILENO. ! RECORD LEADER
6 0 LEADER$ WRITE 1 BBOPENFLG !
7 RECORD-OFF 1000 MS RECORD
8 THEN ;
9
10 : SETPARAM
11 BEGIN SETTIME UNTIL SETDATE SETALTITUDE
12 SETNAME SETEQUIP SETHEADING ;
13
14
15 -->

SCREEN NUMBER 42

0 ( SERVICE ROUTINES FOR FUNCTION KEYS ) 01MAR84WBD
1 : BBON 1 BBONFLG ! ;
2 : BBOFF 0 BBONFLG ! ;
3 : SMPLTOGGL SAMPLEFLG @ NOT SAMPLEFLG ! ;
4 : RESETLINK 3 SLPINIT$ WRITE CLEARSL ;
5
6 : BBCLOSE BBOPEN?
7 IF
8 0 " 99:99:999,+9.999,9,+9.9999999999" WRITE
9 0 LFCSR$ COUNT WRITE RECORD-OFF 0 BBOPENFLG !
10 ELSE
11 BELL BELL BELL
12 THEN ;
13
14 : SHUTDOWN BBCLOSE CURON 25OFF 1 24 HVCUR HXRV CLOSEALL QUIT ;
15 -->
SCREEN NUMBER 43
0 ( SERVICES COMMANDS INPUT BY FUNCTION KEYS) 01MAR84WBD
1 : NEWPARAMS 1 24 HVCUR PAGE
2 BBOPEN? DUP IF BBCLOSE THEN SEPARAM
3 IF BBOPEN THEN :
4 : CHECKINPUT
5 BEGIN
6 [COMPILE] ?TERMINAL DUP
7 WHILE
8 27 =
9 IF
10 [COMPILE] ?TERMINAL
11 -->
12
13
14
15

SCREEN NUMBER 44
0 ( SERVICES COMMANDS INPUT BY FUNCTION KEYS) 01MAR84WBD
1 CASE
2 83 OF BBON ENDOF f1 KEY
3 84 OF BBOFF ENDOF f2 KEY
4 85 OF BBOPEN ENDOF f3 KEY
5 86 OF BBCLOSE ENDOF f4 KEY
6 87 OF NEWPARAMS ENDOF f5 KEY
7 80 OF SMPLTOGGL ENDOF BLUE KEY
8 81 OF RESETLINK ENDOF RED KEY
9 82 OF SHUTDOWN ENDOF WHITE KEY
10 BELL BELL FOR OTHER FUNCTION
11 EDCASE
12 THEN
13 REPEAT DROP ; -->
14
15

SCREEN NUMBER 45
0 ( INPUT THE DATA AND PROCESS ) 01MAR84WBD
1 : SAMPLE TIME ANALOG DISCRETE RANGE BBON? BBOPEN? AND
2 IF
3 BYTEBUCKET
4 THEN
5 SMPLNUM @
6 CASE
7 1 OF AMPLCOMP ENDOF COMPUTE AMPLITUDE
8 2 OF FREQCOMP ENDOF COMPUTE FREQUENCY
9 3 OF ATTEN POS-AMP CR ENDOF ATTENUATOR & BLIP
10 4 OF WRITE25 0 SMPLNUM ! ENDOF WRITE 25TH LINE
11 0 SMPLNUM ! OTHERWISE ZERO THE SAMPLE NO.
12 END CASE
13 SMPLNUM 1+! ; INCREMENT THE SAMPLE NUMBER
14 -->
15
SCREEN NUMBER 46
0 ( START UP DATA COLLECTION) 01MAR84WBD
1 : START
2 0 FILENO. ! 0 BBONFLG ! 1 SAMPLEFLG ! 5 SMPLNUM !
3 INIT PAGE CUROFF 0 BBOPENFLG ! 0 FILES ! CLEARSL 25ON
4 BEGIN SAMPLEON?
5 IF
6 SAMPLE
7 THEN
8 CHECKINPUT
9 AGAIN ; -->
10
11
12
13
14
15

SCREEN NUMBER 47
0 ( START UP OF TEST) 01MAR84WBD
1
2
3 : CHECKPARAM
4 PAGE TIME CR
5 ." TIME:" HERV T1$ TYPE HXRV CR
6 ." DATE" HERV DATE$ TYPE HXRV CR
7 ." ALTITUDE:" HERV ALTITUDE$ TYPE HXRV CR
8 ." HEADING:" HERV HEADING$ TYPE HXRV CR
9 ." FILE NAME:" CR HERV FILENAME$ TYPE HXRV CR
10 ." EQUIPMENT:" CR HERV EQUIP$ TYPE HXRV CR ;
11
12
13 -->
14
15

SCREEN NUMBER 48
0 ( BEGIN EVERYTHING) 15APR83WBD
1
2 : FLY
3 BEGIN
4 CHECKPARAM CR
5 ." SYSTEM PARAMETERS--<RETURN> TO START DATA COLLECTION "
6 ." ANY OTHER KEY TO SET NEW" CR ." PARAMETERS (CNTL-C "
7 ." ABORTS)"
8 KEY 13 = 13 IS <RETURN> KEY
9 IF
10 START
11 ELSE
12 SETPARAM
13 THEN
14 AGAIN ;
15 ;S
SCREEN NUMBER 49
0 ( TYPE THE MENU ON SCREEN) 04DEC83WBD
1 : MENU-TYPE PAGE
2 CR CR HERV ." A" HXRV ." ntenna-Show frequency,amplitude"
3 CR CR HERV ." B" HXRV ." ye-Exit to HDOS-FORTH to get back"
4 CR CR HERV ." D" HXRV ." ata- Show raw receiver data"
5 CR CR HERV ." F" HXRV ." actor-Check antenna factor "
6 CR CR HERV ." G" HXRV ." o-Start data collection"
7 CR CR HERV ." I" HXRV ." nitialize-The SL-803-A unit "
8 ." (to FORTH input level if MENU not TURNKEY)"
9 CR CR HERV ." M" HXRV ." enu-Type this list"
10 CR CR HERV ." P" HXRV ." arameters-Check file parameters"
11 CR CR HERV ." R" HXRV ." ange-Show slant range"
12 CR CR HERV ." T" HXRV ." ime-Display the time"
13 CR
14
15 -->

SCREEN NUMBER 50
0 27APR83WBD
1 : USURE? BELL ." -ARE YOU SURE (Y/N)?" KEY 89 = ;
2
3 : MENU-DATA ANALOG DISCRETE 10 SPACES D1$ A1$ TYPE TYPE
4 CR CR ;
5
6 0 VARIABLE MOUNTFLAG DISMOUNTDISK is used to prevent
7 possible harm to the disk when collecting data. It dismounts
8 ( --- ) the disk and prompts the user to
9 : DISMOUNTDISK MOUNTFLAG @ 13 <> remove it if it has not been
10 IF dismounted.
11 DISMOUNT 100 1 DO BELL LOOP PAGE 1 12 HVCUR HERV
12 ." THE DISK IS NOW DISMOUNTED, PLEASE REMOVE THE "
13 ." DISK. HIT ANY KEY WHEN READY" HXRV
14 13 MOUNTFLAG ! KEY DROP
15 THEN ; -->

SCREEN NUMBER 51
0 ( SHOW THE ANTENNA FACTOR, MAY CHANGE)
1 : MENU-FACTOR ANTFACTOR @
2 BEGIN DUP
3 CR ." ANTENNA FACTOR NOW = "
4 HERV ." DB" HXRV
5 CR ." <RETURN> IF OK,OTHER KEY TO RESET" CR
6 KEY 13 = NOT
7 WHILE 2DROP
8 CR ." INPUT THE ANTENNA FACTOR EXPRESSED IN DB"
9 CR ." ENTER A MAXIMUM OF 3 CHARACTERS (-99 TO 999) "
10 CR ." NO ' +' SIGN PLEASE"
11 CR 3 GET DROP CR CR
12 REPEAT ANTFACTOR ! MENU-TYPE ;
13
14
15 -->
SCREEN NUMBER 52
0 ( MENU PARAMETER OPTIONS ) 27APR83WBD
1
2 : MENU-PARAM
3     BEGIN CHECKPARAM CR
4     " FILE PARAMETERS <RETURN> IF OK"
5     " ANY OTHER KEY TO RESET"
6     KEY 13 = NOT
7     WHILE
8     SETPARAM
9     REPEAT
10     MENU-TYPE ;
11
12
13
14
15

SCREEN NUMBER 53
0 ( MENU OPTIONS ) 01MAR84WBD
1 : MENU-ANTENNA ANALOG DISCRETE ATTEN AMPLCOMP FREQCOMP
2     FREQUENCY @ ." MHZ" 3 SPACES
3     AMPLITUDE @ ." AMPL:" ." DB" 3 SPACES
4     ." ATT:" ATTSET @ ." DB" ;
5
6
7 : MENU-RANGE RANGE ." -SLANT RANGE IN METERS:" R1$ TYPE ;
8
9 : MENU-TIME TIME 3 SPACES T1$ TYPE ;
10
11
12
13
14
15

SCREEN NUMBER 54
0 ( PROCESS KEY INPUT ) 04DEC83WBD
1 : MENU-INPUT
2     CASE
3       65 OF ." ANTENNA" MENU-ANTENNA CR ENDOF
4       66 OF ." BYE" USURE? IF BYE THEN CR ENDOF
5       68 OF ." DATA" MENU-DATA ENDOF
6       70 OF ." FACTOR" MENU-FACTOR ENDOF
7       71 OF ." GO" FLY OPENALL MENU-TYPE CR ENDOF
8       73 OF ." INITIALIZE" CLOSEALL QUIT ENDOF
9       77 OF ." MENU" MENU-TYPE CR ENDOF
10      80 OF ." PARAMETERS" MENU-PARAM CR ENDOF
11      82 OF ." RANGE" MENU-RANGE CR ENDOF
12      84 OF ." TIME" MENU-TIME CR ENDOF
13
14     ENDCASE ;
15     -->
MENU is the word that will start all data collection routines. MENU is the system TURNKEY word.

: MENU DISMOUNTDISK
OPENALL INIT
MENU-TYPE BEGIN KEY MENU-INPUT AGAIN ;

;S
APPENDIX B

MINIRANGER DEVICE DRIVER ASSEMBLER LISTING
MINIRANGER DEVICE DRIVER ASSEMBLER LISTING

* MiniRanger Device Driver
*
* Written by Bill Drury - December 3, 1983
*
* Adapted from Loran-C device driver written
* by Bill Drury - November 1982
*
* TITLE 'MINIRANGER DEVICE DRIVER'
*
* This device driver polls the H89 port that is connected
to the MINIRANGER GTP output. The device driver checks
for input during the service of the system TICCNT
interrupts occurring every 2 mS. The polling routine saves
only the range data coming from the GTP port. This range
data is checked to be a valid ASCII number and the block
length is checked for validity. The LF at the end of the
GTP block is used as the delimiter.
*
* Incoming data is stored in an input buffer until a complete
* and valid buffer is received. At that time the input buffer
* is flagged for output and the output buffer is flagged for
* input. There is always a complete buffer available to be
* read by a calling program.
*
* DVDFLV EQU 0C7H
DVD.ENT EQU 200H
DC.MAX EQU 11
DT.CR EQU 00000010B CAPABLE OF READ
DT.CW EQU 00000100B CAPABLE OF WRITE
EC.EOF EQU 01H END OF FILE
EC.FNO EQU 09H FILE NOT OPEN
EC.ILR EQU 0AH ILLEGAL REQUEST
EC.FAO EQU 19H FILE ALREADY OPEN
UIVEC EQU 201FH
$TBRA EQU 193Eh
LORC EQU 340Q LORAN PORT ADDRESS
UR.RBR EQU 0 RECIIVER BUFFER REGISTER
UR.THR EQU 0 TRANSMITTER HOLDING REGISTER
UR.DLL EQU 0 DIVISOR LATCH LEAST
UR.DLM EQU 1 DIVISOR LATCH MOST
UR.IER EQU 1 INTERRUPT ENABLE
UR.IIR EQU 2 INTERRUPT IDENTIFICATION
UR.LCR EQU 3 LINE CONTROL
UR.MCR EQU 4 MODEM CONTROL
UR.LSR EQU 5 LINE STATUS
UR.MSR EQU 6 MODEM STATUS
UC.DR EQU 00000001B DATA READY MASK
ASC.STX EQU 02H ASCII STX
ASC.A EQU 41H ASCII A
ASC.BEL EQU 07H ASCII BEL
NL EQU 0AH NEW LINE CHAR.
ASC.X EQU 58H ASCII X
ASC.0 EQU 30H ASCII ZERO
ASC.COL EQU 3AH ASCII COLON
ASC.CR EQU 0DH ASCII CARRIAGE RETURN
ASC.LF EQU 0AH ASCII LINE FEED
NCHAR EQU 5 NUMBER OF CHARACTERS
XTEXT HOSEQU
XTEXT ESVAL
EJECT
STL 'PIC CODE HEADER'
CODE PIC
$ EQU *+DVD.ENT-6
DB DVDFLV
DB DT.CR+DT.CW
DB 1
DB 1
DB DT.CR+DT.CW
DS 7
DB 0
DS $-*
SPACE 3,9
START EQU *
CPI DC.MAX
JNC ILLREQ
CALL $TBRA
DB LCREAD-* F0-READ FROM DEVICE
DB LCWRIT-* F1-WRITE TO DEVICE
DB LCNOOP-* F2-READ REGARDLESS
DB LCOPEN-* F3-OPEN FOR READS
DB LCOPEN-* F4-OPEN FOR WRITES
DB ILLREQ-* F5-OPEN FOR UPDATES
DB LCNOOP-* F6-CLOSE CHANNEL TO DEVICE
DB LCNOOP-* F7-ABORT
DB ILLREQ-* F8-MOUNT DEVICE
DB LCLOAD-* F9-LOAD DEVICE
DB LCNOOP-* F10-EXAMINE DEVICE READY STATUS
EJECT
STL 'DATA READ ROUTINE'
*
* LCREAD-READS THE A BLOCK DATA COMPILED IN BUFFER
* POINTED TO BY OUTBUF TO LOCATION POINTED
* TO BY (DE)
* ENTRY-(DE)=TO ADDRESS
* (BC)=NO. OF BYTES REQUESTED (AT LEAST NCHAR)
* EXIT-NONE
* USES-ALL
*
LCREAD EQU *
LDA OPENFL
ORA  A,EC.FNO  TEST OPEN STATUS
MVI  A,EC.FNO  PREPARE FOR FILE NOT OPEN
STC
RZ
MOV  A,C  WITH FNO ERROR
CPI  NCHAR  AT LEAST NCHAR BYTES REQUESTED?
JNC  RDA1  IF YES, PROCEED
MOV  A,B
ORA  A  MULTIPLE OF 256 REQUESTED?
JZ  ILLREQ  IF NOT, RETURN ILLEGAL REQUEST

RDA1
PUSH  B  SAVE BYTES REQUESTED
LXI  B,NCHAR  FORCE TRANSFER OF NCHAR BYTES
LHLD  OUTBUF  POINT HL TO A BLOCK DATA
DI
CALL  MOVBUF  TRANSFER DATA
EI
POP  B  RESTORE BYTES REQUESTED
LXI  H,-NCHAR
DAD  B  SUBTRACT BYTES SENT
MOV  B,H
MOV  C,L  PUT REMAINING BYTE COUNT IN BC
MVI  L,NCHAR  LOAD L WITH BYTES SENT

RDA2
MOV  A,B
ORA  C  SEE IF BYTES REMAINING=0
RZ  IF SO RETURN
XRA  A
STAX  D  STORE NULL AT NEXT TO LOCATION
DCX  B  DECREMENT BYTES LEFT
INX  D  INC NEXT TO ADDRESS
INR  L  INCREMENT BYTES SENT
JNZ  RDA2  IF LESS THAN 256 SENT, GO AGAIN
MOV  A,B
ORA  C  SENT ALL BYTES REQUESTED?
RZ  IF YES, RETURN NO ERROR
MVI  A,EC.EOF  SET END OF FILE MESSAGE
STC  INDICATE AN ERROR
RET
EJECT
STL 'DRIVER FUNCTIONS'

LCLOAD EQU *
LHLD  UIVEC+1  GET HDOS TICCNT VECTOR
SHLD  RETURN+1  INSTALL AT END OF POLLMR ROUTINE
LXI  H,POLLMR  GET MINIRANGER'S START ADDRESS
SHLD  UIVEC+1  REPLACE HDOS TICCNT VECTOR
XRA  A  CLEAR CARRY
RET
ILLREQ EQU *
MVI  A,EC.ILR  ILLEGAL FUNCTION REQUESTED
STC  SET ILLEGAL REQUEST CODE
STC  INDICATE ERROR
RET
SPACE  3,9  NO ACTION TO BE TAKEN
LCNOOP EQU *
XRA A
RET
SPACE 3,9

LCOPEN EQU *
XRA A
OUT LORC+UR.LCR LORC+UR.IER
MVI A,80H
OUT LORC+UR.LCR
MVI A,140Q
OUT LORC+UR.DLL
MVI A,0
OUT LORC+UR.DLM
MVI A,7
OUT LORC+UR.LCR
IN LORC+UR.RBR
MVI A,-1
STA OPENFL
LXI H,BUFFA
SHLD INBUF
LXI H,BUFFB
SHLD OUTBUF

* TEMPORARY
MVI M,53H
INX H
MVI M,54H
INX H
MVI M,41H
INX H
MVI M,52H
INX H
MVI M,54H

* RET
SPACE 3,9

* LCWRIT EQU *
LDA OPENFL
ORA A
MVI A,EC.FNO
STC
RZ
WTLOOP LDA S.CAADDR+1
ANA A
RNZ
IN LORC+UR.LSR
ANI 0010000B
JZ WTLOOP
MVI A,2
OUT LORC+UR.THR

WITH NO ERROR

CLEAR CARRY
WITH NO ERROR

CLEAR A
CLEAR DLAB
DISABLE 8250 INTERRUPTS
SET DLAB
1200 BAUD LSB=140Q
1200 BAUD MSB=0
2 STOP BITS, 8 BIT WORDS
CLEAR GARBAGE
FLAG FILE OPEN
INITIALIZE BUFFA AS INPUT
INITIALIZE BUFFB AS OUTPUT

RETURN IF CTRL-A,-B,-C,-Z HIT
RETURN IF NOT EMPTY, LOOP
LOAD MODE 2 CODE
OUTPUT MODE 2 CODE
XRA A CLEAR CARRY
RET
EJECT
STL 'DATA MOVE ROUTINE'

* * 
MOVBUF-MOVES (BC) BYTES FROM ((HL)) TO ((DE))
* *
ENTRY:(BC)=NO. OF BYTES TO BE MOVED
(HL)=FROM ADDRESS
(DE)=TO ADDRESS
EXIT : (DE) AND (HL) INCREMENTED BY (BC)
(BC)=0
* 
USES : ALL
* 

MOVBUF EQU *
MOV A,B SEE IF FINISHED
ORA C
RZ RETURN IF (BC)=0
MOV A,M GET BYTE TO BE MOVED
STAX D STORE AT TO LOCATION
INX H BUMP FROM POINTER
INX D BUMP TO POINTER
DCX B DECREMENT BYTE COUNT
JMP MOVBUF GO AGAIN
EJECT
STL 'DATA COLLECTION ROUTINE'

* * 
POLLMR:COME HERE DURING THE SERVICE OF THE TICCNT INTERRUPT
* THAT OCCURS EVERY 2 MILLISECONDS.THE ROUTINE CHECKS
* THE MINIRANGER FOR DATA AND PROCESSES THE DATA IN
* THE PROPER MANNER.
* *
* UPON RECEIPT OF THE LAST DATA CHARACTER,
* THE OUTBUF POINTER IS SET TO THE BUFFER JUST COMPLETED
* AND THE INBUF POINTER IS SET TO THE OTHER BUFFER.
* 
POLLMR EQU *
IN LORC+UR.LSR READ STATUS REGISTER
ANI UC.DR
JZ RETURN RETURN IF DATA NOT READY
IN LORC+UR.RBR INPUT DATA
ANI 7FH MASK OFF PARITY BIT
MOV B,A SAVE CHARACTER
CPI ASC.LF CHECK FOR END OF BLOCK
JZ MARK
LDA COUNT
ADI 1 BUMP CHARACTER COUNTER
JC CLEAR CLEAR IF COUNTER OVERRUNS
STA COUNT
CPI 23
JC RETURN IF COUNT<24, NOT A RANGE
CPI 28
JNC RETURN IF COUNT > 28, NOT A RANGE
MOV A,B
CPI ASC.0
JC CLEAR IF CHAR < 0, NOT VALID
CPI ASC.COL
JNC CLEAR IF CHAR > 9, NOT VALID
LDA COUNT
SUI 23 FORM CHAR POSITION COUNT
MOV E,A
MVI D,0
LHLD INBUF GET INPUT BUFFER ADDR
DAD D FORM DESTINATION ADDR
MOV M,B STORE CHAR IN INBUF
JMP RETURN

MARK LDA COUNT
CPI 38
JNZ CLEAR CLEAR IF NOT 38 CHARs RCVD

* * SWITCH BUFFERS *

LHLD OUTBUF GET CURRENT OUTBUF ADDR
LXI D,BUFFA GET BUFFA ADDR
MOV A,D
CMP H SEE IF HI BYTES MATCH
JNE AOUTBIN
MOV A,E
CMP L
JNE AOUTBIN
LXI H,BUFFA BUFFA WAS OUTBUF
SHLD INBUF BUFFA NOW INBUF
LXI H,BUFFB BUFFB NOW OUTBUF
SHLD OUTBUF BUFFB NOW OUTBUF
JMP CLEAR
AOUTBIN LXI H,BUFFA BUFFA WAS INBUF
SHLD OUTBUF BUFFA NOW OUTBUF
LXI H,BUFFB BUFFB NOW OUTBUF
SHLD INBUF BUFFB NOW INBUF

* CLEAR XRA A
STA COUNT CLEAR CHAR COUNT
RETURN EQU *
JMP 0 ADDR CHANGED AT LOAD TIME

* * MR: STORAGE AREAS *

BUFFA DS 5 BUFFER FOR STORING DATA
BUFFB DS 5 BUFFER FOR STORING DATA
OUTBUF DS 2 POINTER TO OUTPUT BUFFER
INBUF DS 2 POINTER TO INPUT BUFFER
COUNT DB 0 CHARACTER COUNTER
SL-803-A DEVICE DRIVER ASSEMBLER LISTING

TITLE 'SLDVD - SL: Device Driver, H8-4'
***
SLDVD - SL-803-A DEVICE DRIVER.
***
* J.G. LETWIN
* G. Chandler 78.10
* 79.11
* 79.12
* Originally ATDVD for use as 'Alternate Terminal' device driver.
* Modified by Jim Nickum for use as SL: Device Driver,
* subsequently modified by Bill Drury to provide functions
* desired for SL-803-A communications for the Avionics EMI
* project.
*
SPACE 4
** ATDVD IS THE DEVICE DRIVER FOR THE DEVICE
*
*
AT:
*
* IF H8410=0
* THEN
* PORT = 374-5
* ELSE
* PORT = 320-7
*
XTEXT ASCII
XTEXT DDDEF
XTEXT MTR
XTEXT HOSEQU
XTEXT ESVG
XTEXT ECDEF
XTEXT PICDEF
XTEXT DEVDEF
XTEXT DVDDEF
XTEXT U8250
XTEXT U8251
*
*
SLPORT EQU 320Q
TIME EQU 06H
ASC.AST EQU 2AH
ASC.SHP EQU 23H
ASC.0 EQU 30H
*
*
NUMBER OF TIMEOUT LOOPS
ASCII CODE FOR *
ASCII CODE FOR #
ASCII CODE FOR 0
* AIO.UNI EQU 041061A ADDRESS OF I/O UNIT NUMBER
* *
  CODE HEADER
* *
  CODE PIC
EQU $+DVD.ENT-6
DB DVDFLV DEVICE DRIVER FLAG VALUE
DB DT.CR+DT.CW DEVICE CAPABILITY: READ AND WRITE
DB 00000001B MOUNTED UNIT MASK
DB 1 ONLY 1 UNIT
DB DT.CR+DT.CW 0: CAPABLE OF WRITE
DS 7 1-7: IGNORED
DB DVDFLV
DS $-*
*
EJECT
STL 'ASSEMBLY CONSTANTS'
*** ASSEMBLY CONSTANTS
*
*
** DEFAULT DEVICE DEFINITIONS
*
IF H8410
DFLT.AT EQU 320Q PORT ADDRESS
DFLT.BD EQU 030A 4800 BAUD
ELSE
DFLT.AT EQU 374Q PORT ADDRESS
DFLT.BD EQU 000A
ENDIF
*
*
DFLT.PD EQU 0 DEFAULT NUMBER OF PAD CHARACTERS
DFLT.WD EQU 80 80 COLUMN WIDTH
DFLT.CX EQU 1 INITIAL COLUMN INDEX
DFLT.CS EQU 0 DEFAULT CTL-S SETTING
SPACE 4
*
*
SB.1 EQU 00000000B ONE STOP BIT
SB.2 EQU 10000000B TWO STOP BITS
*
*
MLC EQU 00000000B MAP LOWER CASE
NOMLC EQU 00000001B NO MAP OF LOWER CASE
EJECT
STL 'MAIN-LINE'
*** ATDVD ENTRY POINT.
*
*
ENTRY (A) = PROCESS CODE
(BC) = BYTE COUNT (USUALLY)
* (DE) = MEMORY ADDRESS (USUALLY)
* EXIT 'C' CLEAR IF OK
* 'C' SET IF ERROR
* (A) = ERROR CODE
* USES ALL

ATDVS EQU * ENTRY POINT
CALL $TBRA ENTER PROCESSOR
DB SLREAD-* READ
DB ATWRITE-* WRITE
DB ATABTR-* READR
DB ATOPE-* OPENR
DB ATOPE-* OPENW
DB ATABTR-* OPENU
DB ATNAP-* CLOSE
DB ATABT-* ABORT
DB ATABTR-* MOUNT
DB ATLOAD-* LOAD

ATABT SPACE 4,10
** ATABTR - ISSUE DEVICE DRIVER ABORT TO REQUEST.

ATABT MVI A,EC.DDA DEVICE DRIVER ABORT
STC
RET

ATABT SPACE 4,10
** ATABT - ABORT DEVICE DRIVER

ATLOAD SPACE 4,10
** ATLOAD - LOAD DEVICE DRIVER

ATLOAD EQU *
ANA A CLEAR CARRY
RET

ATABT SPACE 4,10
** ATABT - OPEN (READ OR WRITE)

ATABT XRA A
STA EOFLG CLEAR EOF ON INPUT FLAG
LDA TAT.POR
LHLD TAT.BAU
IF H8410
CALL I8250
ELSE
CALL I8251
ENDIF
MVI A,-1
STA OPENFL FLAG FILE OPEN
MVI A,CR
CALL  TCH   \hspace{2cm} \text{TCH - \text{RESET COLUMN INDEX, AND RETURN CARRIAGE}}
RET

\text{ATNOP SPACE 4,10}
\hspace{2cm} \text{** ATNOP - IGNORE REQUEST.}
\hspace{2cm} \text{***}

\text{ATNOP ANA  A}
RET \hspace{2cm} \text{DO NOTHING}
STL 'ATREAD - \text{READ}'
EJECT
\hspace{2cm} \text{** ATREAD - READ DATA FROM CONSOLE.}
\hspace{2cm} \text{***}

\text{SLREAD READS BYTES \text{UNTIL THE REQUEST IS SATISFIED,}}
\hspace{2cm} \text{OR A CTL-D IS STRUCK. THE CTL-D IS TAKEN AS EOF.}
\hspace{2cm} \text{***}

\text{ATR2 STAX  D}
\hspace{2cm} \text{STORE CHAR}
INX  D
DCX  B

\text{SLREAD EQU *}
LDA  OPENFL
ORA  A \hspace{2cm} \text{CHECK OPEN STATUS}
MVI  A,EC.FNO
STC \hspace{2cm} \text{PREPARE FOR FILE NOT OPEN}
RZ
PUSH B
PUSH D

\text{CHECK MOV  A,B}
ORA  C \hspace{2cm} \text{SEE IF HAVE ALL CHARACTERS}
JNZ  INPUT \hspace{2cm} \text{IF NOT, GET ANOTHER}
POP  D
POP  B \hspace{2cm} \text{CLEAR PUSHED REGISTERS}
RET

\text{INPUT tries to receive a character from the given port.}
\hspace{2cm} \text{**}
\text{If the character is not immediately ready, INPUT will}
\hspace{2cm} \text{delay for a certain time waiting for it. If the character}
\hspace{2cm} \text{is not received in this time, INPUT will assume}
\hspace{2cm} \text{character transmission is faulty and so will arrange to}
\hspace{2cm} \text{send all 0's back to the calling program.}
\hspace{2cm} \text{***}

\text{INPUT MVI  A,TIME}
STA  THIGH \hspace{2cm} \text{STORE TIME SET}
XRA  A
STA  TLOW \hspace{2cm} \text{INITIALIZE TIME LOW COUNTER}

\text{GETCHAR IN  SLPRT+UR.LSR}
\hspace{2cm} \text{GET PORT STATUS}
ANI  UC.DR \hspace{2cm} \text{SEE IF CHARACTER READY}
JZ  NOTRDY
IN  SLPRT+UR.RBR \hspace{2cm} \text{READ IN CHARACTER}
ANI  7FH \hspace{2cm} \text{STRIP OF PARITY BIT}
STAX  D \hspace{2cm} \text{STORE CHARACTER}
INX  D \hspace{2cm} \text{INCREMENT DEST. POINTER}
DECENTRE COUNTER
CHECK
NOTRDY LDA TLOW
  CPI 0 SEE IF ZERO
  JZ CHECKHI IF SO, CHECK HI TIME
  DCR A DECREMENT TIMEOUT
  STA TLOW STORE NEW LOW TIME
  JMP DELAY GO TO DELAY
CHECKHI LDA THIGH
  CPI 0 SEE IF ZERO
  JZ LOST IF SO, ASSUME CHAR. LOST
  DCR A DECREMENT HIGH TIME
  MVI A, 255
  STA THIGH
  STA TLOW SET LOW TIMER

NOTE THAT THE SETTING OF THE LOW TIMER IS THE
DECIDING FACTOR FOR THE DURATION OF EACH TIMEOUT
CYCLE. THE DEFINITION OF TIME DETERMINES HOW
MANY TIMEOUT CYCLES ARE TO BE EXECUTED WHILE
WAITING FOR CHARACTERS.
TYPICAL TIMING LOOP REQUIRES 131 T-STATES OR
71 MICROSECONDS WITH 1.8432 MHZ CLOCK. A
TIMEOUT CYCLE WITH 255 LOOPS TAKES ABOUT 18
MILLISECONDS.

DELAY XTHL
  XTHL
  JMP GETCHAR GO CHECK FOR CHARACTER

IF NEEDED CHARACTER IS NOT RECEIVED IN GIVEN TIME, THE
ROUTINE WILL COME HERE TO STORE A 0 FOR ALL REQUESTED
CHARACTERS AND THEN GO TO TRANSFER THESE 0's BACK TO THE
CALLING PROGRAM.

RESTORE REQUEST
POPB

SEE IF DONE
ORA C

STORE CHARACTER
STAX D

INCREMENT DESTINATION PNTR
INX D

DECREMENT COUNTER
DCX B

EJECT

EOF FLAG
DB 0

'ATWRITE - WRITE TO AT'
STL 'ATWRITE - WRITE TO AT DEVICE.'
ATWRITE writes the data to the AT device.

The special characters:

- TAB
- FF
- NULL
- NL

are treated separately.

If an abort is posted before the operation completes, ATWRITE exits.

ATWRITE EQU
LDA S.CAADR+1 see if address
ANA A
RNZ A,B abort, claim all done
MOV A,B
ORA C check byte count left
RZ ALL DONE

(A) = character. See if needs special processing:

NULL
NL
TAB
FF
LDAX D
CALL TCH type character

ATW2
INX D increment pointer
DCX B decrement count
JMP ATWRITE

TCH SPACE 4,10

** TCH - type character

(A) = character
* exit none
* uses A,F

TCH
ANA A
RZ null
CPI NL
JE CRLF new line
CPI FF
JNE TCH2 not FF
MVI A,6

TCH1
PUSH PSW
CALL CRLF
POP PSW
```
DCR A
JNZ TCH1
RET

TCH2 CPI TAB
    JNE WCHAR IS NOT TAB, JUST PRINT IT
WCHAR CALL WCHAR WRITE BLANK
    LDA TAT.CX
    DCR A
    ANI 7
    JNZ WCH3
    RET
    STL 'SUBROUTINES'
    EJECT
** WAIT - WAIT FOR THE HANDSHAKE
*
    WAIT EQU *
    RET
    WCHAR SPACE 4,10

** WCHAR - WRITE CHARACTER
*
    ENTRY (A) = CHARACTER
    EXIT NONE
    USES A,F
WCHAR PUSH PSW
    CPI ' '
    JC WCHAR0 NOT PRINTABLE, SO SKIP COUNT CHECK!
    LDA TAT.CX
    DCR A
    LXI H,TAT.WID
    CMP M
    JC WCHAR0 TAT.CX-1 < TAT.WID
    CALL CRLF
WCHAR0 LDA TAT.CON
    ANI MLC+NOMLC
    JNZ WCHAR1 NO MAPPING
    POP PSW
    CALL $MCU
    PUSH PSW
WCHAR1 POP PSW
    CALL OUTCHAR
    CPI CR
    JZ WCHAR2
    CPI ' '
    JC WCHAR3 NOT PRINTABLE
    LDA TAT.CX
    INR A
    STA TAT.CX
    RET
WCHAR2 MVI A,1
    STA TAT.CX
WCHAR3 RET
```
CRLF SPACE 4,10
** CRLF - TYPE CRLF.
*
CRLF MVI A,CR
CALL WCHAR
MVI A,LF
CALL WCHAR
LDA TAT.PAD
ORA A
CRLF1 JZ CRLF2
PUSH PSW
XRA A
CALL WCHAR
POP PSW
DCR A
JMP CRLF1
CRLF2 RET
*
* MOVBUF-MOVES (BC) BYTES FROM ((HL)) TO ((DE))
*
* ENTRY: (BC)=NO. OF BYTE TO BE MOVED
* (HL)=FROM ADDRESS
* (DE)=TO ADDRESS
* EXIT: (DE) AND (HL) INCREMENTED BY ENTRY VALUE OF (BC)
* (BC)=0
* USES :ALL
*
MOVBUF EQU *
MOV A,B
ORA C SEE IF FINISHED
RZ
MOV A,M GET BYTE TO BE MOVED
STAX D STORE AT TO LOCATION
INX H BUMP FROM POINTER
INX D BUMP TO POINTER
DCX B DECREMENT BYTE COUNT
JMP MOVBUF GO AGAIN
XTEXT DVDIO
XTEXT MCU
EJECT
*** TAT.UNT - TABLE AT: UNIT CONSTANTS
*
TAT.UNA EQU *
TAT.UNT DB 0 UNIT NUMBER
7]=1 IF ASSIGNED
TAT.POR DB DFLT.AT PORT NUMBER
D.PORT EQU TAT.POR
TAT.BAU DW DFLT.BD BAUD RATE
7]=1 IF TWO STOP BITS
TAT.CON DB MLC CONFIGURATION BYTE
TAT.PAD DB DFLT.PD NUMBER OF PAD CHAR. FOR <CR>
TAT.WID DB DFLT.WD TERMINAL WIDTH
CLOCK DEVICE DRIVER ASSEMBLER LISTING

TITLE 'CLOCK DRIVER ROUTINE'
STL 'ADAPTED FROM D. LAMM'
SPACE 6

DVDFLV EQU 0C7H
DVD.ENT EQU 200H
DC.MAX EQU 11
DT.CR EQU 00000010B
DT.CW EQU 00000100B
EC.EOF EQU 01H
EC.FNO EQU 09H
EC.ILR EQU 0AH
EC.FAO EQU 19H
UIVEC EQU 201FH
$TBRA EQU 193EH
NL EQU 0AH
CAL EQU -1-500
SPACE 3
CODE PIC

$ EQU *+DVD.ENT-6
DB DVDFLV
DB DT.CR+DT.CW
DB 1
DB 1
DB DT.CR+DT.CW
DS 7
DB 0
DS $-*
STL 'DRIVER ENTRY POINT'
EJECT

START EQU *
CPI DC.MAX
JNC ILLEGAL
CALL $TBRA
DB READ-*
DB WRITE-*
DB IGNORE-*
DB OPREAD-*
DB OPWRITE-*
DB ILLEGAL-*
DB CLOSE-*
DB CLOSE-*
DB ILLEGAL-*
DB LOAD-*
DB IGNORE-*
STL 'CK.DVD FUNCTION PROCESSOR'
EJECT

ILLEGAL EQU *
MVI A,EC.ILR
STC
RET
SPACE 3,9

IGNORE EQU *
XRA A
RET
SPACE 3,9

LOAD EQU *
LHLD UIVEC+1
SHLD CLKRET+1
LXI H,CLOCK
SHLD UIVEC+1
XRA A
RET
SPACE 3,9

OPREAD EQU *
LDA RSTAT
ORA A
JZ OPREAD1
MVI A,EC.FAO
STC
RET

OPREAD1 MVI A,-1
STA RSTAT
XRA A
RET
SPACE 3,9

OPWRITE EQU *
LDA WSTAT
ORA A
JZ OPWRITE1
MVI A,EC.FAO
STC
RET

OPWRITE1 MVI A,-1
STA WSTAT
MVI A,8
STA XFERCNT
LXI H,TEMPBUF
SHLD BUFPTR
XRA A
RET
SPACE 3,9

CLOSE EQU *
XRA A
STA RSTAT
STA WSTAT
RET
SPACE 3,9

MOVE EQU *
MOV A,B
ORA C
RZ
MOV A,M
STAX D
INX H
INX D
DCX B
JMP MOVE

STL 'CK.DVD WRITE PROCESSOR'

EJECT

WRITE EQU *
LDA WSTAT
ORA A
MVI A,EC.FNO
STC
RZ

WRITE1 MOV A,B
ORA C
RZ
LDA XFERCNT
ORA A
JZ WRITE2
DCR A
STA XFERCNT
LHLD BUFPTR
LDAX D
MOV M,A
INX H
SHLD BUFPTR
INX D
DCX B
JMP WRITE1

WRITE2 LXI H,TEMPBUF
LXI D,TIMEBUF
LXI B,8
DI
CALL MOVE
EI
RET
SPACE 3,9
STL 'CK.DVD READ PROCESSOR'
EJECT

READ EQU *
LDA RSTAT
ORA A
MVI A, EC.FNO
STC
RZ
MOV A,C
CPI 9
JNZ READ1
MOV A,B
ORA A
JZ ILLEGAL

READ1
PUSH B
LXI B,9
LXI H, TIMEBUF
DI
CALL MOVE
EI
POP B
LXI H, -9
DAD B
MOV B,H
MOV B,H
MOV C,L
MVI L,9

READ2
MOV A,B
ORA C
RZ
XRA A
STAX D
DCX B
INX D
INR L
JNZ READ2
MOV A,B
ORA C
RZ
MVI A, EC.EOF
STC
RET
STL 'CK.DVD TICCNT ENTRY POINT'
EJECT

CLOCK EQU *
LHLD TICKS
INX H
SHLD TICKS
LXI B, CAL
DAD B
JNC CLKRET
SHLD TICKS
MVI C,'0'
INRS LXI H,TIMEBUF+7
INR M
MOV A,M
CPI '9'+1
JM CLKRET
MOV M,C
INRTS DCX H
INR M
MOV A,M
CPI '6'
JM CLKRET
MOV M,C
INRM DCX H
DCX H
INR M
MOV A,M
CPI '9'+1
JM CLKRET
MOV M,C
INRTM DCX H
INR M
MOV A,M
CPI '6'
JM CLKRET
MOV M,C
INRH DCX H
DCX H
INR M
MOV A,M
CPI '4'
JM CLKRET
INRH1 DCX H
MOV A,M
CPI '2'
JM INRH2
MOV M,C
INX H
MOV M,C
JMP CLKRET
INRH2 INX H
MOV A,M
CPI '9'+1
JM CLKRET
MOV M,C
INRTH DCX H
INR M
SPACE 3,9
CLKRET EQU *
BYTEBUCKET DEVICE DRIVER ASSEMBLER LISTING

H84IO  EQU  0          ASSEMBLE FOR H8-4 CARD
IF    H84IO
TITLE 'BBDVD - BB: Device Driver, H8-4'
ELSE
TITLE 'BBDVD - BB: Device Driver, H8-5'
ENDIF
EJECT
***  BBDVD - BB DEVICE DRIVER.
*
*   J.G. LETWIN
*
*   G. Chandler   78.10
*   79.11
*   79.12
*   J. Nickum     82.10
*
SPACE  4
**  BBDVD IS THE DEVICE DRIVER FOR THE DEVICE
*
*  BB:
*
*  IF H84IO=0
*  THEN
*      PORT = 374-5
*  ELSE
*      PORT = 330-7
*  *
XTEXT  HOSDEF
XTEXT  ASCII
XTEXT  DDDEF
XTEXT  MTR
XTEXT  HOSEQU
XTEXT ESVAL
XTEXT ECDEF
XTEXT PICDEF
XTEXT DEVDEF
XTEXT DSTDDEF
XTEXT U8250
XTEXT U8251

* *
XON EQU 11H TRANSMIT ON
XOFF EQU 13H TRANSMIT OFF
* *
* *
AIO.UNI EQU 041061A ADDRESS OF I/O UNIT NUMBER
* *
* *
CODE HEADER
* *
* *
CODE PIC
* *
* *
$ EQU *+DVD.ENT-6
DB DVDFLV DEVICE DRIVER FLAG VALUE
DB DT.CR+DT.CW DEVICE CAPABILITY: READ AND WRITE
DB 00000001B MOUNTED UNIT MASK
DB 1 ONLY 1 UNIT
DB DT.CR+DT.CW 0: CAPABLE OF WRITE
DS 7 1-7: IGNORED
DB DVDFLV
DS $-
*
EJECT
STL 'ASSEMBLY CONSTANTS'
*** ASSEMBLY CONSTANTS
* *
** DEFAULT DEVICE DEFINITIONS
**
IF H84IO
DFLT.AT EQU 330Q PORT ADDRESS
DFLT.BD EQU 014A 9600 BAUD
ELSE
DFLT.AT EQU 374Q PORT ADDRESS
DFLT.BD EQU 000A
ENDIF
*
DFLT.PD EQU 0 DEFAULT NUMBER OF PAD CHARACTERS
DFLT.WD EQU 80 80 COLUMN WIDTH
DFLT.CX EQU 1 INITIAL COLUMN INDEX
DFLT.CS EQU 0 DEFAULT CTL-S SETTING
**

* SB.1 EQU 00000000B ONE STOP BIT
SB.2 EQU 10000000B TWO STOP BITS
*
MLC EQU 00000000B MAP LOWER CASE
NOMLC EQU 00000001B NO MAP OF LOWER CASE
EJECT
STL 'MAIN-LINE'
*** BDVD ENTRY POINT.
*
ENTRY (A) = PROCESS CODE
(BC) = BYTE COUNT (USUALLY)
(DE) = MEMORY ADDRESS (USUALLY)
EXIT 'C' CLEAR IF OK
'C' SET IF ERROR
(A) = ERROR CODE
*
USES ALL
*
BBDVD EQU *
ENTRY POINT
CALL $TBRA ENTER PROCESSOR
DB BBREAD-* READ
DB BBWRITE-* WRITE
DB BBABTR-* READR
DB BBOPER-* OPENR
DB BBOPEW-* OPENW
DB BBABTR-* OPENU
DB BBCLOSE-* CLOSE
DB BBABT-* ABORT
DB BBABTR-* MOUNT
DB BBLOAD-* LOAD

BBABTR SPACE 4,10
** BBABTR - ISSUE DEVICE DRIVER ABORT TO REQUEST.
*
* BBABTR MVI A,EC.DDA DEVICE DRIVER ABORT
STC
RET
BBABT SPACE 4,10
** BBABT - ABORT DEVICE DRIVER
*
BBABT CALL CRLF
RET
*
* BBLOAD SPACE 4,10
** BBLOAD - LOAD DEVICE DRIVER
*
BBLOAD EQU *
ANA A CLEAR CARRY
RET

BBOPER SPACE 4,10
** BBOPER - OPEN BB: FOR READ

BBOPER EQU *

LDA RSTAT GET READ STATUS
ANA A
MVI A,EC.FAO READY FOR FILE ALREADY OPEN ERROR
STC
RNZ RETURN WITH ERROR IF RSTAT<>0

MVI A,OFFH SET OPEN READ FLAG
STA RSTAT

XRA A CLEAR A
STA OPTR CLEAR OUT POINTER
STA OPTR+1
STA IPTR CLEAR IN POINTER
STA IPTR+1

JMP BBOPE CONTINUE WITH OPEN PROCESS

BBPEW SPACE 4,10
** BBPEW - OPEN BB: FOR WRITE

BBPEW EQU *

LDA WSTAT GET WRITE STATUS
ANA A
MVI A,EC.FAO READY FOR FILE ALREADY OPEN ERROR
STC
RNZ RETURN WITH ERROR IF WSTAT<>0

MVI A,OFFH SET OPEN WRITE FLAG
STA WSTAT

JMP BBOPE CONTINUE WITH WRITE OPEN PROCESS

BBPE SPACE 4,10
** BBPE - OPEN (READ OR WRITE)

BBPE XRA A
STA EOFFLAG CLEAR EOF ON INPUT FLAG
LDA TAT.POR
LHLD TAT.BAU
IF H8410
CALL 18250
ELSE
CALL 18251
BBNOPS  SPACE  4,10
**  BBNOP  -  IGNORE REQUEST.
*
*
BBCLOSE  XRA    A    CLEAR A
          STA    RSTAT  CLEAR READ OPEN FLAG
          STA    WSTAT  CLEAR WRITE OPEN FLAG
          ANA    A
          RET
          STL   'BBREAD - READ'
          EJECT
**  BBREAD  -  READ DATA FROM CONSOLE.
*
*
BBREAD  READS BYTES UNTIL THE REQUEST IS SATISIFIED,
*  OR A CTL-D IS STRUCK. THE CTL-D IS TAKEN AS EOF.
*  A CNTL-Q WILL BE OUTPUT TO BEGIN READING THE FILE
*  A CNTL-S WILL BE OUTPUT WHEN EXITING THIS ROUTINE
*  THIS WILL ALLOW XON/XOFF CONTROL OF THE BYTE BUCKET
*
BBREAD  LDA    EOFLG   GET EOF FLAG
          RAR    SET CARRY AND EOF ERROR
          RC     RETURN IF EOF WAS SET
          MOV    A,B   TEST REQ > 1 BYTES
          ORA    C
          RZ     RETURN IF NOT
          PUSH   H     SAVE H AND L REGS
          ATR1   LDA    OPTR    GET OUTPUT POINTER
                   MOV    H,A
                   LDA    IPTR    GET INPUT POINTER
                   CMP    H     ARE THEY EQUAL ?
                   JNZ    GETDAT  JUMP IF NOT
                   LDA    OPTR+1  GET OUT PNTR HIGH BYTE
                   MOV    H,A
                   LDA    IPTR+1  GET IN PNTR HIGH BYTE
                   CMP    H     EQUAL?
                   JNZ    GETDAT  JUMP IF NOT
                   CALL   RDBUFF  GET NEW BUFFER FULL
                   JC     ERRR    CONSOLE INTERRUPT OR ERROR
                   GETDAT  CALL   BUFD   GET DATA THROUGH OPTR
                   ANA    A     TEST FOR NULL
                   JZ     IGNORE  IGNORE CHAR IF NULL
                   CPI    CTLD   IS CHAR CNTL-D
                   JZ     SOEF    SET EOF FLAG IF EQUAL
                   CPI    LF     IS IT A LINE FEED
                   JZ     NEWLINE CHANGE TO NL CHAR
                   CPI    CR     IS IT A CARRIAGE RETURN
                   JNZ    ATR4    JUMP IF NOT CR CHAR
                   IGNORE LHLD   OPTR   GET OUTPUT CHAR PTR
                   INX    H     BUMP POINTER
                   SHLD   OPTR   SAVE THE POINTER
JMP ATR1
NEWLINE MVI A, NL
ATR4 STAX D
INX D
LHLD OPTR
INX H
SHLD OPTR
DCX B
MOV A, B
ORA C
JZ TEOF
JMP ATR1
SEOF MVI A, EC.EOF*2+1
STA EOFLG
ATR2 MVI A, 0
STAX D
INX D
STNULL DCX B
MOV A, B
ORA C
JNZ ATR2
TEOF POP H
ORA A
RET
ERRR POP H
RET
*
*
*
STL 'BBWRITE - WRITE TO BB'
EJECT
*
***
BBWRITE - WRITE TO BB DEVICE.
*
BBWRITE WRITES THE DATA TO THE BB DEVICE.
*
THE SPECIAL CHARACTERS:
*
TAB
*
FF
*
NULL
*
NL

OUTPUT BUFFER POINTER
INPUT BUFFER POINTER
LOCAL TICCN
OPEN FOR READ FLAG
OPEN FOR WRITE FLAG
EOF FLAG
STOP TRANSMISSION FLAG

GO TO TEST IF BUFFER COMPLETE
CHANGE CHAR TO NL
SAVE CHAR IN MBASIC BUFFER
POINT TO NEXT MBASIC BUFFER LOC
GET OPTR
BUMP POINTER
SAVE THE POINTER
DECR MBASIC CHAR CNTR
TEST IF SECTOR CNTR = 0
IF = 0 TEST FOR EOF AND EXIT
NOT DONE GET ANOTHER CHAR.
ERROR CODE + CARRY WHEN SHIFT
SET ERROR FLAG ADDR
NULL CHAR
MBAS BUFFER
LOOP TILL B=0
RESTORE H & L REGS
CLEAR CARRY
RESTORE H & L
* * ARE TREATED SEPERATELY.
* * IF AN ABORT IS POSTED BEFORE THE OPERATION COMPLETS,
* BBWRITE EXITS.

BBWRITE EQU *
LDA WSTAT GET WRITE OPEN FLAG
ANA A
MVI A,EC.FNO READY FOR FILE NOT OPEN ERR
STC
RZ
RETURN ERROR IF WSTAT=0
TURNON XRA A CLEAR WRITE STOP FLAG
STA WSTOP
WRITE1 LDA S.CAADR+1 SEE IF ADDRESS
ANA A
RNZ ABORT, CLAIM ALL DONE
MOV A,B
ORA C CHECK BYTE COUNT LEFT
RZ ALL DONE

CALL INCHAR GET BB INPUT
JC GO IF NO INPUT GO ON
CPI XOFF SEE IF XOFF SENT
JNZ NOTOFF
MVI A,-1 YES,FLAG TO STOP WRITING
STA WSTOP
JMP WRITE1

NTOFF CPI XON SEE IF XON SENT
JZ TURNON YES,CLEAR STOP FLAG AND WRITE
GO LDA WSTOP CHECK IF STOP FLAG SET
ANA A
JNZ WRITE1 IF SET, LOOP

(A) = CHARACTER. SEE IF NEEDS SPECIAL PROCESSING:

NULL
NL
TAB
FF

LDAX D
CALL TCH TYPE CHARACTER
ATW2 INX D INCREMENT POINTER
DCX B DECREMENT COUNT
JMP WRITE1
TCH SPACE 4,10
** TCH - TYPE CHARACTER

(A) = CHARACTER
* EXIT NONE
* USES A,F
*
TCH ANA A
    RZ
    CPI NL       IS NULL
    JE CRLF     IS NEW LINE
    CPI CR       IS CHAR RTN
    JE CRLF       YES
    CALL OUTCHAR   PRINT IT
    RET

* *
IBUFF DS 1024    BB BUFFER AREA
* *
STL 'SUBROUTINES'
    EJECT
* *
BUFVAL   WILL TAKE IBUFF AND ADD TO IT THE OPTR VALUE
    TO PRODUCE AN Indexed ADDRESS
* *
ENTRY   NONE
* *
EXIT   (A)=VALUE OF CHAR IN IBUFF POINTED TO BY OPTR
* *
BUFVAL LXI H,IBUFF
    LDA OPTR
    ADD L
    MOV L,A
    LDA OPTR+1
    ADC H
    MOV H,A
    MOV A,M
    RET

* *
* IBUFVAL   WILL PRODUCE THE INDEXED ADDRESS FOR THE INPUT
    POINTER FOR THE BYTE BUCKET BUFFER
* *
* ENTRY   NONE
* *
EXIT   D & E REGS CONTAIN THE ADDRESS OF THE NEXT BUFFER
    ADDRESS
* *
* IBUFVAL LXI D,IBUFF
    LDA IPTR
    ADD E
    MOV E,A
    LDA IPTR+1
RDBUF  MVI    A,CTLQ    SEND XON
        CALL   OUTCHAR
        MVI    A,0      CLEAR OUT POINTER
        STA    OPTR
        STA    OPTR+1
        STA    IPTR    CLEAR IN POINTER
        STA    IPTR+1
        CALL   INBUFF  GET THE BUFFER FULL
        RET

INBUFF  EQU    *
        PUSH   B
        PUSH   D
        LOOP   LDA   .TICCNT+1
                STA   CNTR
        LOOP2  LDA   D.PORT
                MOV   H,A
                MVI   L,UR.LSR
                LDA   S.CAADR+1
                ANA   A
                STC
                JZ    LOOP1
                POP   D
                POP   B
                RET
        LOOP1  CALL   IN
                ANI   UC.DR
                JZ    INCRR1  IF DATA NOT READY, WAIT
                MVI   L,UR.RBR
                CALL   IN
                ANI   177Q    MASK TOP BIT
                JZ    LOOP     IGNORE NULL CHARACTER
                PUSH  PSW     SAVE A
                CALL   IBUFVAL GET INPUT BUFFER ADDR.
                POP   PSW     RESTORE A REG.
                STAX  D
                LHLD   IPTR
                INX    H
                SHLD   IPTR
MOV A,H
CPI 3 SEE IF READ 1020 CHAR.
JC LOOP IF H<3, NOT 1020 YET
MOV A,L
CPI 252 IF H<3 & L<252, NOT 1020
JC LOOP
JMP INCR2 H=3 & L=252, RETURN

INCR1 LDA CNTR DATA NOT READY
MOV B,A
LDA .TICCNT+1
SUB B
CPI 4 WAIT 2 SECONDS
JC LOOP2
LDA IPTR
MOV H,A TEST IF ZERO
ORA H
JNZ INCR2 IF NOT ZERO NO ERROR
MVI A,53 IF ZERO FILE NOT FOUND ERROR
STC FLAG ERROR

INCR2 POP D
POP B
RET

WAIT SPACE 4,10
** WAIT - WAIT FOR THE HANDSHAKE
*
WAIT EQU *
RET *
*

CRLF SPACE 4,10
** CRLF - TYPE CRLF.
*
CRLF MVI A,CR CALL OUTCHAR
MVI A,LF CALL OUTCHAR

CRLF2 RET *
*
*

XTEXT DVDIO
XTEXT MCU
EJECT

*** TAT.UNT - TABLE AT: UNIT CONSTANTS
*
TAT. UNT EQU *
TAT. UNT DB 0 UNIT NUMBER
7] = 1 IF ASSIGNED
TAT. POR DB DFLT. AT PORT NUMBER
D. PORT EQU TAT. POR
TAT. BAU DW DFLT. BD BAUD RATE
7] = 1 IF TWO STOP BITS
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAT.CON</td>
<td>MLC</td>
<td>CONFIGURATION BYTE</td>
</tr>
<tr>
<td>TAT.PAD</td>
<td>DFLT.PD</td>
<td>NUMBER OF PAD CHAR. FOR &lt;CR&gt;</td>
</tr>
<tr>
<td>TAT.WID</td>
<td>DFLT.WD</td>
<td>TERMINAL WIDTH</td>
</tr>
<tr>
<td>TAT.CX</td>
<td>DFLT.CX</td>
<td>COLUMN INDEX</td>
</tr>
<tr>
<td>TAT.CTS</td>
<td>DFLT.CS</td>
<td>CTL-S FLAG</td>
</tr>
<tr>
<td>EJECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STL</td>
<td>'SUBROUTINES'</td>
<td></td>
</tr>
<tr>
<td>XTEXT</td>
<td>TBRA</td>
<td></td>
</tr>
<tr>
<td>XTEXT</td>
<td>TYPTX</td>
<td></td>
</tr>
<tr>
<td>DW</td>
<td>'RL'</td>
<td>DUMMY ADDRESS FOR RELOCATION</td>
</tr>
<tr>
<td>DS</td>
<td>64</td>
<td>PATCH AREA</td>
</tr>
<tr>
<td>LON</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>