AIRCREW INFLIGHT PHYSIOLOGICAL
DATA ACQUISITION SYSTEM II

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Joseph G. Jolda
Capt USAF

and
Stephen J. Wanzek
Capt USAF
Graduate Electrical Engineering
December 1977

Approved for public release; distribution unlimited.
Preface

This thesis is an investigation into a second generation Inflight Physiological Data Acquisition System sponsored by the School of Aerospace Medicine (SAM) at Brooks AFB, Texas. The investigation entailed designing, building, and testing both the hardware and the software for a microprocessor-based prototype system. The resulting prototype, presented in this report, is a complete data acquisition system including the sensor interfaces, the microprocessor, and a permanent memory device. The body of the report is written in general terms for the user at SAM; a more detailed description including circuit diagrams, program listings, and technical discussions is given in the appendices.

This project would not have been possible without the encouragement and help of a number of people. We would like to acknowledge the individuals who helped to make this thesis a reality.

We are indebted to Dr. Mathew Kabrisky, our thesis advisor, who gave us the guidance and enthusiasm to pursue this project. We are grateful to Dr. Gary Lamont, Capt Mike Weber, and Capt Chuck Cornell whose suggestions and assistance helped immeasurably. We would also like to thank Bob Durham and Dan Zambon for their superb technical assistance. We are also grateful to the personnel from the Crew Technology Division at SAM for their support in obtaining the hardware. Our gratitude also goes to Al Haun of Analog Devices for his assistance with the data acquisition module and to Jack Capehart at the ASD computer center for his assistance in transferring the operating system to PROM.

Our deepest gratitude goes to Steve's wife, Cindy, and to Greg's wife, Sue, and son, Jason, for their continuing encouragement and patience while we pursued this project.
Contents

Preface ........................................................................................................................................ ii
List of Figures .......................................................................................................................... v
List of Tables ........................................................................................................................... vii
List of Abbreviations ................................................................................................................ viii
Notation ..................................................................................................................................... x
Abstract ..................................................................................................................................... xi

I. Introduction ............................................................................................................................... 1
   Background ................................................................................................................................. 1
      Current System ....................................................................................................................... 1
      System Standards .................................................................................................................. 2
   Scope of Thesis ........................................................................................................................ 3
      Feasibility ............................................................................................................................... 3
      Assumptions .......................................................................................................................... 4
      System Configuration ............................................................................................................ 5

II. Hardware .................................................................................................................................. 8
   SBC 80/20 ................................................................................................................................. 9
   DAS 1128 Data Acquisition Module ....................................................................................... 12
   Sensor Interface ....................................................................................................................... 13
      Oxygen Partial Pressure ....................................................................................................... 13
      Acceleration .......................................................................................................................... 14
      ECG (Heart Rate) ................................................................................................................. 14
   Data Storage ............................................................................................................................ 18
      Bubble Memory ...................................................................................................................... 18
      Hazeltine Video Terminal ...................................................................................................... 19

III. General Algorithms .............................................................................................................. 21
   Introduction .............................................................................................................................. 21
   Service Routines ..................................................................................................................... 22
      POZIN .................................................................................................................................. 22
      PO2OUT ................................................................................................................................ 23
   Flow Rate (FLWRT) .................................................................................................................. 23
   Absolute Pressure (ABSPR) .................................................................................................... 24
   Breathing Rate (BRRT) ............................................................................................................ 24
   X, Y, & Z G's ......................................................................................................................... 24
   Heart Rate (HEART) ................................................................................................................ 24
   System Support Software ....................................................................................................... 24
      Power-up Routine .................................................................................................................. 25
      Error Routine ....................................................................................................................... 25
      Permanent Data Storage Routine ....................................................................................... 25
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV.</td>
<td>Post-Flight Data Conversion</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Absolute Pressure</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Cabin Altitude</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Z-G's</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Heart Rate</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Breathing Rate</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>Minute Ventilation Volume</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>Oxygen Volumes</td>
<td>37</td>
</tr>
<tr>
<td>V.</td>
<td>System Tests</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Developmental Tests</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Final Tests</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Heart Rate</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Oxygen Partial Pressure</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Absolute Pressure</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Flow Rate</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Breathing Rate</td>
<td>48</td>
</tr>
<tr>
<td>VI.</td>
<td>Suggestions and Recommendations</td>
<td>49</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>52</td>
</tr>
<tr>
<td>Appendix A:</td>
<td>System Characteristics</td>
<td>55</td>
</tr>
<tr>
<td>Appendix B:</td>
<td>Operating System</td>
<td>63</td>
</tr>
<tr>
<td>Appendix C:</td>
<td>IFPDAS II Prototype Users Guide</td>
<td>106</td>
</tr>
<tr>
<td>Appendix D:</td>
<td>Post Flight Data Conversion Routine</td>
<td>109</td>
</tr>
<tr>
<td>Appendix E:</td>
<td>SBC 80/20 Hardware Description</td>
<td>123</td>
</tr>
<tr>
<td>Appendix F:</td>
<td>DAS 1128 Hardware Description</td>
<td>129</td>
</tr>
<tr>
<td>Appendix G:</td>
<td>Sensor Interfaces</td>
<td>139</td>
</tr>
<tr>
<td>Appendix H:</td>
<td>Magnetic Bubble Memory Interface Diagrams</td>
<td>152</td>
</tr>
<tr>
<td>Appendix I:</td>
<td>Oxygen Consumption Calculation for Single Breath Analysis</td>
<td>157</td>
</tr>
<tr>
<td>Vita</td>
<td></td>
<td>163</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Components</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>SBC 80/20 Functional Diagram</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>DAS 1128 Functional Diagram</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Heart Rate Detector Functional Diagram</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>ECG Lead Placement</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>ECG and R-wave Detector Output</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Cabin Absolute Pressure vs Time</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>Cabin Altitude vs Time</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>Minimum and Maximum Z-G's vs Time</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>Heart Rate vs Time</td>
<td>33</td>
</tr>
<tr>
<td>11</td>
<td>Breathing Rate vs Time</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>Minute Ventilation Volume vs Time</td>
<td>38</td>
</tr>
<tr>
<td>13</td>
<td>Inspired Oxygen Quantity vs Time</td>
<td>40</td>
</tr>
<tr>
<td>14</td>
<td>Expired Oxygen Quantity vs Time</td>
<td>41</td>
</tr>
<tr>
<td>15</td>
<td>Heart Rate Computations</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>Sine Wave Approximation of Flow Rate Signal</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>DAS 1128 Interconnections and Interface</td>
<td>61</td>
</tr>
<tr>
<td>18</td>
<td>Component Board Layout</td>
<td>62</td>
</tr>
<tr>
<td>19</td>
<td>Operating System Flow Chart</td>
<td>63</td>
</tr>
<tr>
<td>20</td>
<td>Post Flight Data Conversion Flow Chart</td>
<td>110</td>
</tr>
<tr>
<td>21</td>
<td>OM11 Interface Circuit</td>
<td>140</td>
</tr>
<tr>
<td>22</td>
<td>OM11 Compensator Circuit</td>
<td>142</td>
</tr>
<tr>
<td>23</td>
<td>Accelerometer Interface Circuit</td>
<td>143</td>
</tr>
</tbody>
</table>
### List of Figures

(Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ECG Amplifier and R-Wave Detector</td>
<td>145</td>
</tr>
<tr>
<td>25</td>
<td>Digital Interval Counter</td>
<td>146</td>
</tr>
<tr>
<td>26</td>
<td>R-wave Interval Counter Timing Diagram</td>
<td>146</td>
</tr>
<tr>
<td>27</td>
<td>Chassis Board Layout and Part Number/Function</td>
<td>149</td>
</tr>
<tr>
<td>28</td>
<td>External Chassis Interface Connections</td>
<td>150</td>
</tr>
<tr>
<td>29</td>
<td>Logic Diagram, Bubble Memory Controller (Page 1)</td>
<td>153</td>
</tr>
<tr>
<td>30</td>
<td>Logic Diagram, Bubble Memory Controller (Page 2)</td>
<td>154</td>
</tr>
<tr>
<td>31</td>
<td>Magnetic Bubble Memory Printed Circuit Evaluation Board (Page 1)</td>
<td>155</td>
</tr>
<tr>
<td>32</td>
<td>Magnetic Bubble Memory Printed Circuit Evaluation Board (Page 2)</td>
<td>156</td>
</tr>
<tr>
<td>33</td>
<td>Flow Rate Weighted Expired Oxygen Measurement</td>
<td>161</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Parameter Ranges</td>
<td>56</td>
</tr>
<tr>
<td>II</td>
<td>Prototype Power Supplies</td>
<td>57</td>
</tr>
<tr>
<td>III</td>
<td>Memory Map</td>
<td>57</td>
</tr>
<tr>
<td>IV</td>
<td>Pin Connections for DAS 1128 Interface</td>
<td>58</td>
</tr>
<tr>
<td>V</td>
<td>Pin Connections for Heart Rate Detector Interface</td>
<td>59</td>
</tr>
<tr>
<td>VI</td>
<td>Hazeltine Interface Connections</td>
<td>60</td>
</tr>
<tr>
<td>VII</td>
<td>Other Characteristics</td>
<td>60</td>
</tr>
<tr>
<td>VIII</td>
<td>8255 I/O Port Addresses</td>
<td>125</td>
</tr>
<tr>
<td>IX</td>
<td>DAS 1128 Hardwired Modifications</td>
<td>130</td>
</tr>
<tr>
<td>X</td>
<td>Chassis Pin Connections</td>
<td>151</td>
</tr>
<tr>
<td>XI</td>
<td>Oxygen Ratio</td>
<td>158</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>ABSPR</td>
<td>Absolute pressure sensor service routine</td>
<td></td>
</tr>
<tr>
<td>A/D</td>
<td>Analog to digital</td>
<td></td>
</tr>
<tr>
<td>AFIT</td>
<td>Air Force Institute of Technology</td>
<td></td>
</tr>
<tr>
<td>AMRL</td>
<td>Aerospace Medical Research Laboratory</td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Central processing unit</td>
<td></td>
</tr>
<tr>
<td>ECG</td>
<td>Electrocardiogram</td>
<td></td>
</tr>
<tr>
<td>EOC</td>
<td>End of conversion</td>
<td></td>
</tr>
<tr>
<td>EOI</td>
<td>End of interrupt</td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td>Error correcting routine</td>
<td></td>
</tr>
<tr>
<td>FLWRT</td>
<td>Flow rate sensor service routine</td>
<td></td>
</tr>
<tr>
<td>FORTRAN</td>
<td>(FROMula TRANslation) Engineering programming language</td>
<td></td>
</tr>
<tr>
<td>G's</td>
<td>Acceleration of Gravity</td>
<td></td>
</tr>
<tr>
<td>Hz</td>
<td>Hertz (cycles/sec)</td>
<td></td>
</tr>
<tr>
<td>ICW</td>
<td>Initialization command word</td>
<td></td>
</tr>
<tr>
<td>IFPDAS</td>
<td>Aircrew Inflight Physiological Data Acquisition System</td>
<td></td>
</tr>
<tr>
<td>I/O</td>
<td>Input/output</td>
<td></td>
</tr>
<tr>
<td>IR</td>
<td>Interrupt request</td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>In-service</td>
<td></td>
</tr>
<tr>
<td>LSB</td>
<td>Least significant bit</td>
<td></td>
</tr>
<tr>
<td>mm Hg</td>
<td>Millimeters of Mercury</td>
<td></td>
</tr>
</tbody>
</table>

viii
## List of Abbreviations

(Continued)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB</td>
<td>Most significant bit</td>
</tr>
<tr>
<td>MUX</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>OCW</td>
<td>Operational command word</td>
</tr>
<tr>
<td>PO2IN</td>
<td>Inspired oxygen partial pressure sensor service routine</td>
</tr>
<tr>
<td>PO2OUT</td>
<td>Expired oxygen partial pressure sensor service routine</td>
</tr>
<tr>
<td>PROM</td>
<td>Programmable read–only memory</td>
</tr>
<tr>
<td>RAM</td>
<td>Random access memory; read/write memory</td>
</tr>
<tr>
<td>ROM</td>
<td>Read–only memory</td>
</tr>
<tr>
<td>R–wave</td>
<td>Highest amplitude component of a normal ECG</td>
</tr>
<tr>
<td>SAM</td>
<td>School of Aerospace Medicine</td>
</tr>
<tr>
<td>SBC</td>
<td>Single board computer</td>
</tr>
<tr>
<td>USART</td>
<td>Universal synchronous/asynchronous receiver/transmitter</td>
</tr>
</tbody>
</table>
Notation

(name) Signifies positive true logic (e.g., EOC)

(name) Signifies negative true logic (e.g., $\text{EOC}$)

xxxx Signifies a decimal number (e.g., 3184)

xxxxH Signifies a hexadecimal number (e.g., 0C70H)
Abstract

This paper discusses a second generation microprocessor-based prototype system to acquire, analyze, and store selected environmental and physiological data from a pilot during flight. The Aircrew Inflight Physiological Data Acquisition System (IFPDAS) II consists of an input multiplexer and analog-to-digital converter, a heart rate detector, a microprocessor, and a permanent memory device. The microprocessor's operating system monitors eight sensors, extracts desired information, and stores these reduced data in permanent memory. After the flight, these data are transferred to a land-based computer which completes the data processing and graphs the following environmental and physiological information versus flight time: (1) cabin absolute pressure, (2) cabin altitude, (3) Z-G's, (4) heart rate, (5) breathing rate, (6) minute ventilation volume, (7) inspired oxygen quantity, and (8) expired oxygen quantity.

The completed IFPDAS II prototype provides the desired information well within the required accuracy. It provides the following parameter ranges: (1) heart rate from 53 ± .1 to 225 ± 2.2 b/min, (2) breathing rate from 4.7 ± .1 to 50 ± 1 b/min, (3) minute ventilation volume from 0 to 100 ± 2 l/min, (4) absolute pressure from 0 to 760 ± 2 mm Hg, and (5) G's from -3 to +12 ± .1 G.
AIRCREW INFILIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II

I Introduction

Background

The Crew Technology Division of the USAF School of Aerospace Medicine (SAM) at Brooks AFB, Texas, has recognized the need to relate pilot activity to physiological measurements, to apply these relationships to predict aircrew effectiveness, and to formulate equipment design and use criteria to optimize that effectiveness in present and projected flying roles.

Current System. SAM currently has an "Aircrew Inflight Physiological Data Acquisition System" (IFPDAS) which records seven analog functions on cassette tape:

1. A standard time code (to correlate flight events and physiological effects);
2. Pilot voice;
3. ECG;
4. Cabin pressure;
5. Oxygen consumption;
6. Expired flow;
7. Vertical acceleration.

The IFPDAS consists of two subsystems: one to sense and record the data (inflight), the other to reproduce the data (on the ground – after the flight). This data is then converted to digital signals and processed by a digital computer.
There are several problems with the current IFPDAS. It was designed and built using discrete components and is therefore not as reliable as a system based on modern components. Secondly, it doesn't have the capability to acquire triaxial G's, inspired flow volume, or separate input and output oxygen concentrations. Finally, it is highly specialized and, therefore, inflexible without costly design modifications.

Currently, the IFPDAS hardware is being modified by the U.S. Navy. This modification includes two additional analog functions as well as some replacement of discrete components by integrated circuits. The resulting modification should be more reliable; however, it still won't have the complete desired capability and flexibility.

System Standards. Personnel at SAM have projected the design requirements for the second generation system (IFPDAS II), due for production in the early 1980's. IFPDAS II must provide the following primary data:

1. Cabin pressure;
2. Time code;
3. G's (triaxial, if available);
4. Voice (this can be acquired separately if an all-digital system can be designed).

In addition, this new system should provide the capability to assess three or more of the following:

1. Inspired and expired flow;
2. Input and output oxygen concentrations;
3. Heart rate;
4. Mask pressure;
5. Garment pressure;
The desired range and accuracy for each of the sensors is included in Appendix A.

The IFPDAS II must meet several other system specifications. It must accept probe inputs from 0 to 5 volts which correspond to the appropriate range of each function (for example, 0 – 760 mm Hg for absolute pressure). It should provide a data acquisition time of at least three hours, and it must be time-synchronized to correlate flight events and physiological effects. The IFPDAS was designed to be carried in the pilot's survival vest and, therefore, the original size restrictions must still be met. This means that IFPDAS II must be no larger than 2" x 5" x 9" and must be self-contained, with no external connections to the aircraft. Finally, it is desirable to present a visual display of the status of the device and its probes.

Scope of Thesis

The purpose of this investigation was to determine the feasibility of implementing IFPDAS II as a completely digital system to eliminate the mechanical drives and reduce the post-flight computations; and, if digital implementation was practical, to update the existing equipment, increase its reliability, and extend its capabilities.

Feasibility. A search of the current literature revealed that the microprocessor is extending the capabilities of monitoring systems and data acquisition systems in the medical and engineering fields (Ref 1, 2, 3, & 4). A microprocessor controlled system offers several excellent features, the most important of which are flexibility and high reliability. In order to modify the function of a microprocessor-based system, all
that is generally required is a change in the software, often with little or no change in the hardware. In addition, the microprocessor can manipulate the data, extracting the significant information, thereby reducing the amount of permanent storage required. Finally, since the microprocessor incorporates numerous digital functions onto a single unit, or chip, it replaces an enormous number of discrete components, thereby greatly increasing system reliability. For these reasons, the investigation includes the development of a microprocessor-based IFPDAS II prototype.

**Assumptions.** This IFPDAS II prototype was designed making four assumptions. The first assumption is that the probes supply the desired data. This assumption is required since it is not within the scope of this investigation to redesign the probes. The second assumption is that there is a maximum period of four hours during which data is collected. This is required to establish the memory size needed to store the data, and is justified since the current system is limited to four hours - which personnel at SAM found to be satisfactory. The third assumption is that continuous storage of the data is not required. Instead, a periodic technique (i.e., every 10 to 30 secs) or a "store-on-significant-change" technique could be used. This, too, is necessary to limit the memory size. This assumption should not limit the usefulness of the data since unchanging data is generally not interesting. It is the changes in the data that are important, and both techniques will detect the changes. In addition, it will generally take 10 to 30 seconds to detect changes in, for example, heart rate or flow rate. The fourth and last assumption is that the power and size requirements would not
have to be met for this prototype development. This is required so that proven systems could be used for the development, with less emphasis on their size or power consumption.

System Configuration. The minimum configuration for a microprocessor-based digital data acquisition system would have to include a sensor, an analog-to-digital (A/D) interface, a microprocessor, and a memory device. When more than one sensor is required, the A/D interface becomes more complex. In order to keep this interface to a minimum, the sensor inputs can be multiplexed to one A/D converter, rather than using an A/D converter for each sensor. This not only reduces the number of converters, but reduces the number of inputs to the microprocessor. The multiplexer and single A/D converter configuration allows additional sensors to be interfaced to the system without changing the basic system hardware, while keeping the number of system components to a minimum. These concepts were used to design the IFPDAS II prototype, which consists of six major functional units as shown in Figure 1.

The multiplexer selects one from up to 16 different probe signals as the input to the A/D converter. This converter transforms the analog signal into a digital number which represents the signal for use by the microprocessor (CPU). The heart rate detector (implemented in hardware for required accuracy - see ECG section of Chapter II) supplies the CPU with a number representing the heart beat interval. The CPU combines these numbers with previous data (stored temporarily in the CPU memory), extracts the desired information, and stores the desired result in permanent memory.
Fig. 1. System Components
The microprocessor performs two major functions: data manipulation and system control. Both of these functions are implemented by a software program, called the operating system, which is stored in the CPU memory. The operating system directs selection of the proper input probe, initiates conversion of the analog signal, and routes data to and from both CPU and permanent memory. The operating system also ensures proper interpretation and reduction of the data.

The remaining chapters discuss the IFPDAS II prototype with respect to its hardware, the general algorithms designed to acquire the data, the post-flight data reduction, and the tests of the prototype system. Recommendations for further system development are also given.
II Hardware

Several microprocessors (including the Z80, the TMS9900, the 8080, and the 6800) were considered for use as the central processing unit for the IFPDAS II prototype. Intel Corporation's 8080A CPU has several advantages over the other microprocessors which make it the best choice for the system CPU. A major advantage is that the 8080 is a well established, low cost, highly reliable microprocessor (Ref 5:44-45) which readily integrates with Intel's general purpose peripherals. These peripherals provide a variety of special functions (such as timing and external interfaces) which, together with the 8080, make up a complete computer system. In addition, the devices selected for the A/D interface and the permanent memory had already been interfaced to the 8080. Other considerations include the author's previous experience with the 8080 and the support software available on AFIT's computer system. One final advantage is Intel's new generation 8080A—the 8085—which is 100% software compatible with the 8080 and integrates several functions of the 8080 system onto a single chip (Ref 6:109-113, Ref 7).

The IFPDAS II prototype consists of four major hardware components: Intel's SBC 80/20 (a single board computer containing the 8080 and several peripherals), the DAS 1128 Data Acquisition Module, the sensor interfaces, and the permanent data storage device. Each of these components will be described in the following sections. IFPDAS II prototype characteristics are listed in Appendix A.
Intel's 8080-based single board computer SBC 80/20 was purchased. This allowed engineering development of the IFPDAS II prototype using a proven computer system rather than devoting time to fabrication and testing of a specialized computer system.

The SBC 80/20 is a complete computer system on a single 6.75-by-12 inch printed circuit card. The CPU, system control functions, CPU memories, input/output (I/O) interfaces, interval timers, and interrupt controller all reside on the board (a block diagram of these functions is shown in Figure 2). The CPU functions have been discussed in the section on system configuration; each of the peripheral devices will be introduced in the remainder of this section. (Specific design and operational characteristics are discussed in Appendix E.)

The CPU memory consists of two types of memory: read/write memory (RAM) and read-only memory (ROM). Unlike the read/write memory, the read-only memory is non-volatile, which means that a program or data stored on the ROM will not be lost by turning the power off. For this reason, the operating system will be permanently stored in ROM. Read/write memory is used like a scratch pad by the CPU. Previous data samples, intermediate calculations, and event counters are stored on this temporary memory for later use by the CPU. Any data stored in RAM would be lost if power failed; however, the operating system would recognize the loss of power and reinitialize the system when power is restored.

Communication with the input devices (the data acquisition module and the heart rate detector) is accomplished through two 8255 Programmable Peripheral Interfaces. These 8255s receive data over 8 or 16 data
lines when informed by an external device that the data is ready for
input. The 8255s store the data, indicate to the CPU that the data is
available, and transfers the data to the CPU upon request.

Communication with the current memory device (the Hazeltine video
terminal) is accomplished through the 8251 Programmable Communication
Interface. This device communicates with external devices over a
standardized (RS232) interface. (The use of this standardized interface
allows communication with any RS232-compatible device—not just the
Hazeltine.) The rate of transfer (baud rate or bit rate) and format of
the data is controlled by the system software.

Timing and frequency division are accomplished by the 8253
Programmable Interval Timer. The 8253 contains three independent
timers, each of which is programmed by the operating system. One of
the timers is configured as a real time clock which informs the CPU
of every 50 msec interval. The other two timers are used as frequency
dividers, slowing the system clock to required frequencies. One of the
resulting frequencies is used to establish the baud rate for the 8251
communications interface. The other is used as the clock for the heart
rate detector.

The last peripheral device contained on the SBC 80/20 is the 8259
Programmable Interrupt Controller. An interrupt is notification to the
CPU by an external device that an event has occurred or that the device
requires servicing (Ref 8: Ch 5, 8-33). The 8259 intercepts the inter-
rupt request from the external device, and it informs the CPU that the
interrupt has occurred and where the software service routine can be
found.
DAS 1128 Data Acquisition Module

The DAS 1128 is a self-contained data acquisition system manufactured by Analog Devices. This compact module was selected for use in the prototype since it is a proven system which readily integrates with the 8080 through the 8255. It contains an analog input multiplexer, an A/D converter, and all of the timing and control circuitry needed to perform the complete data acquisition function (Figure 3).

seven physiological and environmental analog signals are input to the signal multiplexer. The multiplexer is directed by the CPU (via the STROBE input) to select these signals sequentially. Conversion of the selected signal is started when the module is triggered by the CPU (via the TRIG input). (These CPU commands are transmitted by the 8255 to the DAS 1128.) When the conversion is complete, the output of the A/D converter is a digital representation of the 0 - 5 volt input, accurate to within 10 mv of the input signal. (This provides the required
accuracy as listed in Appendix A.) At this time, an "end-of-conversion" (EOC) is returned to the 8255, signifying that the digital data and associated sensor identification are ready for transmission. Each conversion requires approximately 25 microseconds, which would allow a maximum of 40,000 conversions per second. This is more than adequate since the current application requires only 1120 conversions per second. (This is based on a sampling rate of 20 Hz. If a faster rate is desired, the IFPDAS II could sample the maximum 16 inputs at over 70 Hz.) The electrical and interface configurations are discussed in Appendix F.

Sensor Interfaces

The IFPDAS II prototype accepts eight physiological and environmental sensors. These probes measure partial pressure of oxygen inhaled and exhaled, respiratory flow rate (expired), triaxial G's, absolute pressure, and ECG. A mask and two oxygen partial pressure sensors (Beckman OM11) were provided by SAM at Brooks AFB, TX. An accelerometer (Statham F-15-340) was obtained from the Aerospace Medical Research Laboratory (AMRL) at Wright-Patterson AFB, OH; and the ECG leads were available in AFIT's Bioengineering Laboratory. (The flow rate and absolute pressure sensors were not available.) Each of the available probes was interfaced to the prototype and is discussed in the following sections.

Oxygen Partial Pressure. The oxygen sensors are located in the mask/hose assembly in the same configuration used for the original IFPDAS (Ref 9:24-25); however, the oxygen information has changed. IFPDAS I provided a measurement representing the instantaneous difference
between the inspired and expired oxygen partial pressures. This is not the information desired; therefore, IFPDAS II provides two measurements, one representing the inspired and one representing the expired oxygen partial pressure. The OM11 sensor output is amplified to provide a 0 - 5 volt signal corresponding to a pressure range of 0 - 760 mm Hg of oxygen.

The OM11 sensor has an 800 msec response time for a 0 to 100% oxygen transition. This is an acceptable time for the present application; however, it is too slow for a breath-by-breath analysis. A complete circuit description of the amplifier is included in Appendix G. Included also is a circuit to reduce the response time to 100 msec, which would be sufficient for a breath-by-breath analysis, if desired.

**Acceleration.** The IFPDAS II prototype was designed to monitor the acceleration forces in all three dimensions. AMRL at Wright-Patterson AFB has mounted three uni-directional accelerometers on a breast plate (Ref 10) to provide the triaxial G's. Only one sensor was available, so only one amplifier was constructed and tested. However, the other two amplifiers (required for full triaxial G measurement) would be identical to the one actually built. The full range of the Statham accelerometer is -15 to +15 G's. The amplification circuit offsets this range and provides a 0 - 5 volt signal which corresponds to -3 to +12 G's. A complete circuit description of the accelerometer interface is given in Appendix G.

**ECG (Heart Rate).** IFPDAS I stores the complete ECG on cassette tape for later analysis. This recording is only reliable enough to provide information for heart rate calculation, by hand, on the ground.
 Normally, the heart rate is the desired information; therefore a heart rate detector was designed and built for the IFPDAS II prototype. This detector provides a digital representation of the heart rate, along with the provision to record the entire ECG waveform with a separate analog recorder, if desired. The heart rate information is derived by counting the time interval between detected R-waves. This count is passed to the CPU, through the second 8255. A functional diagram of the heart rate detector is given in Figure 4.

![Heart Rate Detector Functional Diagram](image)

**Fig. 4. Heart Rate Detector Functional Diagram**

This heart rate detector was designed to provide a maximum amplitude R-wave signal, while eliminating base-line shifts and reducing muscle artifacts. The base-line shifts and muscle artifacts are undesirable since they add extraneous signals and make accurate R-wave detection very difficult. The lead placement shown in Figure 5 is a compromise placement which provides a good QRS-wave least disturbed by muscle artifacts.
Figure 6 shows typical ECG's provided by the amplifier, along with the associated R-wave detector output. Figure 6a was recorded with the subject at rest, while Figure 6b was recorded with the subject exercising heavily on a stationary bicycle. As can be seen from these figures, the R-wave detector provides a highly reliable R-wave indication even under exercise and movement conditions more severe than a pilot would experience during a flight.

The output of the R-wave detector is used to trigger a counter, which counts the time interval between R-waves. When a subsequent R-wave is detected, this count is passed to the 8255 and the new count is started. Appendix G contains a complete electrical description of the ECG amplifier, R-wave detector, and interval counter.
Fig. 6a. ECG & R-wave Detector Output (Rest)

Fig. 6b. ECG & R-wave Detector Output (Exercise)
Data Storage

A variety of current data storage devices were considered. Several of these were eliminated because of their physical size (magnetic core memory), volatility (RAM, charge-coupled devices), or impracticality (PROM). The only devices considered practical for use on the IFPDAS II prototype were magnetic tape (either wafer or standard cassette) and magnetic bubble memory. (Product information for the two tape storage systems is given in Ref 11, 12.) Both of these storage devices provide non-volatile storage, large capacity, and small size. The magnetic bubble memory was selected over the cassette tape systems in order to provide a completely digital system with no mechanical drives.

Bubble Memory. Bubble memories have several characteristics which make them ideal for use as a mass memory system in this application.

These attributes result primarily from the semiconductor-like processing used to fabricate bubble devices in conjunction with their non-volatile nature. ... Some of the important features resulting from the semiconductor-like processing include reliability, small size, fast access time (relative to electro-mechanical mass storage devices), low cost, and small incremental capacity. ... By virtue of its non-volatility, the bubble device offers removability, asynchronous access, and low power. (Ref 13:1)

Bubble memory systems are still in the developmental stage. It is expected that a complete memory system, including two bubble memories (capable of storing 184,000 bits of information) and all of the system controls and interfacing, will fit on a single 4.5" x 6" x .75" printed circuit board. It is also expected that the power required to operate the system will be approximately three watts. The IFPDAS II prototype only stores data for 10 msec every 10 sec which would allow the CPU
to turn off the memory system during periods of non-storage – thus significantly reducing the power consumption.

Texas Instruments supplied circuit diagrams for the bubble memory controller and functional timing generator (Appendix H). These circuits were wired on a component board and interfaced to the SBC 80/20. Due to limited bubble memory production and procurement difficulties, a complete bubble memory system could not be obtained in time to interface to the IFPDAS II prototype. For this reason a temporary storage method consisting of a Hazeltine 2000 video terminal and digital cassette recorder was used.

**Hazeltine Video Terminal.** The Hazeltine video terminal communicates with the 8251 Programmable Communication Interface discussed in the SBC 80/20 section of this chapter. The operating system transforms the data into a standard format (seven-bit ASCII code) which is sent to the video terminal and displayed on the video screen. This visual display was a valuable developmental tool because it allowed easy interpretation of the data for validating the software operations as well as checking the accuracy of the stored data. This temporary "storage" technique was used until the system was operating as desired. Then data was collected by displaying it on the video screen and automatically copying it permanently on a digital cassette tape. (Appendix C describes the transfer process.) This digital tape represents the final storage of the "inflight" data, which is ready for "post-flight" conversion and graphic display.
The IFPDAS II prototype hardware configuration includes the SBC 80/20 printed circuit board; and a component board containing the DAS 1128, the heart rate detector, and the bubble memory function timing generator. The analog interfaces that provide the 0 - 5 volt signals are packaged in a 3'' x 4'' x 5'' chassis.
III General Algorithms

Introduction

The operation of the hardware discussed in the previous chapter is under the control of a software program called the operating system (Appendix B). Execution of the operating system by the CPU results in initialization of the system and acquisition, reduction, and storage of the data. The operating system selects the proper input sensor, starts the A/D conversion, and inputs the data for reduction and permanent storage. In addition, it ensures that the proper subprogram, or service routine, receives the data.

The biggest constraint on the IFPDAS II is the amount of available permanent storage for the acquired data. Because of this limitation, it is not possible to store the complete waveforms, or even continuous samples of these waveforms, for the desired four hour period. For this reason, a periodic technique (storing the data every ten sec) was used.

Three of the service routines must sum the input data for each ten second interval. In order to constrain the ten second sum to 16 bits of storage, it was necessary to sample no more than 256 times during the 10 second period. A sampling rate of 20 Hz (200 samples in 10 sec) was selected because the input signals change very slowly (less than 5 Hz).

Data are collected from each input signal every 50 msec using a noise-reducing digital filtering technique. This technique consists of taking 8 consecutive samples of the waveform in .5 msec and averaging these 8 samples to produce the 50 msec reading.
In order to collect data from each input signal at the 20 Hz rate, the operating system sets a hardware timer which interrupts the CPU at the end of 50 msec. This interrupt informs the CPU to start the service routine sequence.

The remainder of the chapter discusses the sensor service routines and the additional system support software.

**Service Routines**

There is an independent service routine for each input probe since different information is required from each sensor. Each service routine accepts the 50 msec reading from the averaging routine and performs the required operations to prepare the data for permanent storage.

The following sections discuss the information to be extracted from the input waveforms and the algorithm used in each service routine to reduce the data to the desired form.

**PO2IN.** The personnel at SAM need to assess the pilot's oxygen consumption during flight. One of the required quantities needed to compute this consumption is the amount of oxygen inspired during each breath. To accurately compute the quantity of oxygen inspired, the oxygen partial pressure and the inspired flow rate are required. Currently, the mask/hose assembly does not contain a sensor to measure inspired flow; therefore, a program to approximate the quantity of inspired oxygen was written.

The PO2IN service routine was written to sum all of the inspired partial pressure readings for the ten second period and store the sum for averaging after the flight. This sum is then used in conjunction
with the expired flow rate (assuming equal inspired and expired rates) to compute the oxygen intake. (Appendix I discusses a single breath analysis method for computing oxygen uptake.)

**PO2OUT.** The second required quantity needed to compute the pilot's oxygen consumption is the amount of carbon dioxide produced during each breath. Current carbon dioxide sensors do not lend themselves to inflight applications due to their size and weight; therefore, an approximation of the expired oxygen quantity had to be made.

The PO2OUT service routine sums the expired partial pressure readings for the ten second period and stores the sum for the post-flight conversion, which is similar to the PO2IN methods. (Even though the PO2IN and PO2OUT service routines are inaccurate in their analysis of the oxygen quantities, the sensor interfaces and data handlers were successfully exercised and produce approximate values. A single breath analysis method for computing the expired oxygen quantity is included in Appendix I.)

**Flow Rate (FLWRT).** The third quantity required to compute the oxygen consumption is flow volume. A probe in the oxygen mask/hose assembly measures the expired volume flow rate, which can be integrated with respect to time to obtain flow volume.

The integral of the flow rate waveform can be computed by using a rectangular approximation technique, since there is very little change to the flow rate signal during a 50 msec interval. The FLWRT service routine sums and stores the 200 flow rate readings for subsequent multiplication (after the flight) by the known 50 msec interval to produce the resulting integral.
**Absolute Pressure** (ABSPR). The final quantity required to compute the oxygen consumption is the cabin absolute pressure. This is necessary in order to compensate the flow rate reading for altitude and also to compute percent oxygen in the inspired and expired air. The ABSPR service routine stores a representation of the cabin absolute pressure every ten seconds.

**Breathing Rate** (BRRT). The BRRT service routine uses the flow rate sample for breath detection. The routine determines when the exhaled breath has stopped and marks this event as the start of a new breath. The interval between breaths is counted and stored.

**X, Y, & Z G's.** These service routines search for the maximum and minimum acceleration in each direction during the ten second interval. In order to reduce the effects of transient G impulses, each routine averages eight readings before comparing to the previous minimum and maximum values.

**Heart Rate** (HEART). The HEART service routine checks to see if the heart rate circuit has input a new heart beat interval. If a new count is available, it is read in and stored. Eight heart beat counts are averaged together to provide a representative heart rate.

**System Support Software**

In order to form the complete operating system, several additional routines are necessary. These routines initialize the IFPDAS II prototype, correct detectable errors, store the data on the permanent storage device, and provide the additional support required by the service routines.
Power-up Routine. This routine is executed when the hardware detects that power has been applied to the IFPDAS II prototype. This program configures the IFPDAS to its data acquisition function. This includes programming the 8251 communication interface, the 8255 input ports, the 8253 timers, and the 8259 interrupt controller as described in the hardware chapter. The DAS 1128 is initialized so that probe "0" is the first to be sampled when the service routine series is started. The final tasks accomplished by the power-up routine are the initialization of the scratch pad storage and the initiation of the service routine series (program loop).

Error Routine. The operating system must ensure that the executing service routine is receiving data from its associated probe. Each service routine accomplishes this by checking the sensor identification information. If the service routine and sensor are mismatched, the operating system must select the proper sensor and return to the program loop. This is accomplished by the error routine.

Permanent Data Storage Routine. The operating system keeps track of the running time, and schedules the storage routine every ten seconds. The storage routine transfers a timing preamble and the data computed by the service routines from the scratch pad storage to the permanent storage device. The data is converted to the ASCII code required by the Hazeltine video terminal, and is transmitted over the RS232 interface. After all of the data is transferred, two control characters are transmitted which directs the Hazeltine system to write the data from the video screen onto the digital cassette tape. Finally, the scratch pad storage area is reinitialized. The data storage routine is the only
device-dependent program in the operating system and is, therefore, the only module that will require modification when the new memory system is interfaced.

All of the service routines and system support software are stored on a single 2708 Erasable and Electrically Programmable Read Only Memory chip. This allows program modification by simply erasing and reprogramming the memory.
IV Post-Flight Data Conversion

The post-flight conversions are accomplished by a land-based computer after the flight. The data from the memory device is transferred to the computer where a program completes the data conversion and displays the desired information in graphic form. The form of this display (list, tabular, plot versus time, etc.) can be varied by changing the program.

This investigation utilized AFIT's computer system as the land-based computer. The data from the digital cassette tape is transferred to the computer (Appendix C) for the "post-flight" conversions. A FORTRAN program (Appendix D) reads in this data, converts it back to basic form, completes the calculation of the desired information, and displays this information in graphic form. Three environmental and five physiological parameters are graphed versus time by this conversion program. These are absolute pressure, cabin altitude, Z-G's, heart rate, breathing rate, minute ventilation volume, and inspired and expired oxygen volumes.

The following sections describe these calculations and include a representative graph of the parameter.

Absolute Pressure

The absolute pressure data is a number from 0 to 250 which is directly proportional to a pressure range of 0 to 760 mm Hg. The data
is converted to actual cabin absolute pressure using the following formula:

\[
\text{abs pressure (mm Hg)} = \frac{\text{data}}{250} \times 760 \tag{1}
\]

The cabin absolute pressure is plotted versus time, as in Figure 7.

Cabin Altitude

The absolute pressure data is also used to compute cabin altitude. There is not a simple mathematical relationship between data in mm Hg and altitude in feet (Ref 14:587); therefore, an approximate relationship was derived using linear regression techniques. A logarithmic curve of the form

\[ y = a + b \ln(x) \tag{2} \]

was found to provide the best fit. For the altitude range of 0 to 25,000 feet, the following equation was used:

\[
\text{altitude (feet)} = 170,156 - 25,685 \ln(\text{abs pressure (mm Hg)}) \tag{3}
\]

This equation is accurate to within 275 feet for the 0 to 25,000 foot range. A sample cabin altitude versus time plot is shown in Figure 8.
Fig. 7. Cabin Absolute Pressure vs Time
Fig. 8. Cabin Altitude vs Time.
Z-G's

The acceleration data is a number from 0 to 250 which is directly proportional to a range of -3 to +12 G's. The data is converted to actual G's using the following formula:

\[
\text{acceleration (G's)} = \left\{ \frac{\text{(data)}}{250} \times 15 \right\} - 3 \tag{4}
\]

(The -3 term is necessary to correct for the 3 G offset of the acceleration circuitry.) Minimum and maximum Z-G's (for each ten second interval) are plotted on the same graph, as in Figure 9. (The graphs for X and Y-G's are similar and are not included.)

Heart Rate

The heart rate data is an average number of 4.44 msec (1/225 Hz) counts between detected R-waves. This number is converted to the heart rate using the following equation:

\[
\text{heart rate (beats/min)} = \frac{1}{(4.44 \text{ msec})(\text{count})} \times 60
\]

\[
= \frac{225 \text{ Hz}}{(\text{count})} \times 60 \tag{5}
\]

An example of the heart rate plot is shown in Figure 30.)
Fig. 10. Heart Rate vs Time
Breathing Rate

The breathing rate data is similar to the heart rate data, except that each count represents 50 msec. The breathing rate is calculated from the number of 50 msec counts in the following manner:

\[
\text{breathing rate (breaths/min)} = \frac{1}{(50 \text{ msec})(\text{count})} \times 60
\]

\[
= \frac{20 \text{ Hz}}{(\text{count})} \times 60
\]  

(6)

The sample breathing rate graph is given in Figure 11.

Minute Ventilation Volume

The flow volume equations were derived for the original IFPDAS (Ref 9:5). The equation for flow volume rate \( F \) is

\[
F = V_f \text{ (volts) } \times (24.8 \text{ l/min/volt}) \times \sqrt{\frac{760 \text{ mm}}{\text{abs pressure}}}
\]  

(7)

(The square root term corrects the flow volume rate for actual altitude.) The 50 msec volume flow rate reading \( R \) is a number from 0 to 250 which is directly proportional to a flow rate from 0 to 124 l/min. Each \( R \) could be used to compute an incremental flow volume expired during that 50 msec interval:
Fig. 11. Breathing Rate vs Time

BREATH RATE VS TIME
SUBJECT: JOH
START TIME: 10 OCT 77

BREATH RATE (B/MIN)
TIME (MIN)
volume/50 msec (1) = \( \frac{R}{250} \times (124 \text{ l/min}) \times \sqrt{760 \text{ mm Hg}} \times \frac{.05 \text{ sec}}{60 \text{ sec/min}} \)  

(8)

Summing each of the incremental volumes provides the flow volume for the ten second interval:

\[
\text{volume/10 sec (1)} = \sum_{i=1}^{200} \left\{ \frac{R_i}{250} \times (124 \text{ l/min}) \times \sqrt{760 \text{ mm Hg}} \times \frac{.05 \text{ sec}}{60 \text{ sec/min}} \right\}
\]

(9)

or,

\[
\text{volume/10 sec (1)} = \left\{ \frac{124 \text{ l/min}}{250} \times \sqrt{760 \text{ mm Hg}} \times \frac{.05 \text{ sec}}{60 \text{ sec/min}} \right\} \times \sum_{i=1}^{200} R_i
\]

(10)

The summation term is the value calculated and stored by the IFPDA5 II prototype. Combining constants, Eq (10) reduces to
volume/10 sec (l) = \( 0.0114 \times \frac{\text{summation}}{\sqrt{\text{abs pressure}}} \) \hspace{1cm} (11)

In order to obtain the desired minute ventilation volume, six 10-second volumes are summed. Minute ventilation volume is plotted versus time, as in Figure 12.

**Oxygen Volumes**

The inspired and expired oxygen partial pressure data is a sum of the 50 msec readings. This sum is divided by the number of readings, which gives an average oxygen partial pressure reading for the ten second period:

\[
\text{avg reading} = \frac{\text{sum}}{200} \hspace{1cm} (12)
\]

This average reading is a number between 0 and 250 which is directly proportional to a pressure range of 0 to 760 mm Hg. This data is converted to the fractional amount of oxygen in the air as follows:

\[
\text{fraction } O_2 = \left\{ \frac{\text{avg reading}}{250} \times 760 \right\} / \text{abs pressure} \hspace{1cm} (13)
\]

The quantity of oxygen in the inspired or expired air is then computed by

\[
\text{quantity of } O_2 (l) = (\text{fraction } O_2) \times \text{minute vent vol (l)} \hspace{1cm} (14)
\]
Combining Eqs (12), (13), and (14)

\[
\text{quantity of } O_2 \ (l) = \frac{\text{sum} \times 760}{200 \times 250 \times \text{abs pressure}} \quad (\text{min vent vol}) \quad (15)
\]

or,

\[
\text{quantity of } O_2 \ (l) = \frac{(\text{sum}) \times \text{min vent vol}}{65.79 \times \text{abs pressure}} \quad (16)
\]

The inspired oxygen quantity is graphed in Figure 13; and the expired oxygen quantity is graphed in Figure 14.

The graphs provide a means for easy correlation of the aircraft's pressure altitude and acceleration, and the pilot's heart rate, breathing rate, minute ventilation volume, and oxygen consumption.
Fig. 14. Expired Oxygen Quantity vs Time
V System Tests

The IFPDAS II prototype was continually tested during development. Each program module, and each sensor interface, was checked individually both before and after system integration. After all of the modules and available interfaces were integrated, the IFPDAS II prototype was tested as a complete system. The following sections describe the developmental and final tests.

Developmental Tests

Each of the software modules was designed, coded, and tested on AFIT's main computer system (an 8080 simulator was used to verify the software). The tests consisted of inputting simulated data to check that the modules were manipulating the data properly. After these tests, the programs were transferred to the IFPDAS II prototype's memory.

Then the DAS 1128 was interfaced to the prototype and its conversion and input selection functions were checked. Initially, the STROBE and TRIG signals were supplied by digital switches which were used to manually step through the probe selections and start the A/D conversion. Then the operating system-supplied STROBE and TRIG signals (through the 8255) were used to perform the same functions. Known DC voltages were used as inputs to the signal multiplexer to check the accuracy of the conversion. For example: with the input voltage at 3.00 volts, the A/D converter output was 96H (150), where:
This testing ensured accurate conversion of the data by the DAS 1128 and proper manipulation of the data by the operating system.

After the operation of the DAS 1128 was verified, the data handling technique of each of the service routines was tested by applying a known DC voltage to the seven analog inputs. For the 3.00 volt example listed above, the routines that sum the readings over the 10 second interval (PO2IN, PO2OUT, and FLWRT) calculated a sum of 7530H (30,000), where,

\[
\frac{30,000}{200} = 150
\]  

(18)

Also, the ABSPR routine stored a reading of 96H. To check the X, Y, and Z-G routines, two different DC voltages (3.00 and 4.00 volts) were used during the 10 second interval. Each of the routines stored 96H as the minimum reading and C8H (200) as the maximum reading. These tests showed that the prototype hardware was functioning as designed and that the operating system and its service routines were exercising the desired control and providing the desired data.

The error handling software was checked by forcing a sensor/service routine mismatch. The operating system identified this error and scheduled the ERROR routine which selected the proper sensor for the service routine. A test was also performed to determine how often the
ERROR routine was being executed. The system was run for four hours which provided over two million chances for a sensor/service routine mismatch. During the test period, the ERROR routine was never executed.

As each sensor interface circuit was developed, it was tested independently. It was then connected to the IFPDAS II prototype to test its integration with the system. Two sensors were not available (absolute pressure and expired flow rate); therefore, their hardware interface and software service routines could not be verified using actual sensor inputs. However, simulated inputs were used for the final tests.

The following sections describe the final tests of the completed IFPDAS II prototype and their results.

**Final Tests**

**Heart Rate.** The heart rate detector was checked in the following manner. ECG leads were attached to a subject and input to the detector. The output of the ECG amplifier was connected to channel 1 of a Gould Brush 440 strip chart recorder and the output of the R-wave detector was connected to channel 2. This allowed simultaneous recording of the ECG and R-wave detector outputs. Figure 15 shows a strip chart segment; and, a comparison of the IFPDAS II prototype calculated heart rate with a manual computation directly from the chart.

The IFPDAS II prototype uses eight R-R intervals to calculate a representative heart rate. The end of each averaging period was marked on the strip chart using the "mark-event" indicator; and the associated IFPDAS II calculation was recorded.
Manual Computation

Chart speed: 25 mm/sec
Distance for 8 beats: 155 mm
Time for 8 beats: 6.20 sec
Heart Rate: 77.4 b/min

IFPDAS Results

Average number of 4.44 msec counts: 175
Time for 1 beat: 0.777 sec
Heart Rate: 77.1 b/min

Fig. 15. Heart Rate Computation
The two calculations shown in Figure 15 are within .5% and are typical of the IFPDAS II prototype's accuracy (Appendix A). The differences between the two are due to variances in the chart speed and quantization of the count increment by the prototype.

**Oxygen Partial Pressure.** The oxygen partial pressure sensor amplifiers were calibrated to indicate 152 mm Hg (1.00 volt). The full scale range was then checked by exposing the OM11 sensor to a pure oxygen source. The amplifier output was 5 volts, indicating 760 mm Hg of oxygen. This 5 volt input (FAH) was properly interpreted and stored by the IFPDAS II operating system.

**Acceleration.** The Statham F-15-340 accelerometer was interfaced to the IFPDAS II prototype through its amplifier. The sensor axis was oriented to indicate acceleration in the vertical direction and the output of the amplifier was adjusted to 1.33 volts (+1 G). The accelerometer was then rotated $90^\circ$ to simulate a zero-G situation; and the output dropped to 1 volt (0 G's). Finally, the sensor was rotated an additional $90^\circ$ to simulate a negative G situation; and the output dropped to .67 volts (-1 G's). The IFPDAS II prototype recorded the minimum and maximum G's (within 0.1 G's) during each ten second interval of the test.

**Absolute Pressure.** As mentioned earlier, the absolute pressure probe was not available for use on the prototype. For this reason, a known DC voltage was used to simulate the absolute pressure input. As before, the operating system stored the proper value.

**Flow Rate.** Since the flow rate sensor was not available, a .20 Hz sine wave oscillator was used to supply the flow rate signal. The sine
wave approximates the flow rate signal as shown in Figure 16. (The samples of the negative portion of the sine wave are "0" since the A/D converter input range is 0 - 5 volts.)

Fig. 16. Sine Wave Approximation of Flow Rate Signal

The amplitude of the sine wave was set to 3.0 volts and the frequency to .20 Hz using an oscilloscope. (The frequency was confirmed by the breathing rate - see the breathing rate test.) The sum of the 200 readings ranged from 2535H to 253EH, averaging 253AH (9530). The corresponding approximation of the integral is

\[(9530 \text{ counts}) \times .05 \text{ sec} = 476.50 \text{ count-sec} \quad (19)\]
Since there are 50 counts/volt,

\[
\frac{476.50 \text{ count-sec}}{50 \text{ counts/volt}} = 9.53 \text{ volt-sec} \quad \text{(20)}
\]

The exact integral yields:

\[
2 \times \int_{0}^{\frac{2.5}{3}} 3 \sin(2\pi(.2)t) \, dt = \frac{12}{.4\pi} = 9.55 \text{ volt-sec} \quad \text{(21)}
\]

The approximation of the integral is within .3% of the actual value.
(This technique was also used to check the PO2IN and PO2OUT summing routines, with the same accuracy.)

**Breathing Rate.** The flow rate readings are used by the operating system to determine the breathing rate. During the flow rate test, the breathing rate routine recorded 64H (100) fifty msec counts per breath which is one breath every 5 seconds, or 12 breaths per minute.

The test results show that the IFPDAS II prototype monitors the sensor inputs accurately, and can extract and store the desired data. Appendix A summarizes these results and discusses the prototype's accuracy.
VI Suggestions and Recommendations

The BK10103 Magnetic Bubble Memory System and the TMS9916 controller should be interfaced to the IFPDAS II prototype and tested. This will allow evaluation of the IFPDAS II as a complete independent system.

The data handling technique for the bubble memory system will have to be developed. This includes a software routine to format the data and provide the commands necessary to store the data. In addition, a data retrieval method must be designed. This could include direct transfer from the bubble memory to a large land-based computer; or transfer of the data to a digital cassette tape. Transferring the data to cassette tape might be desirable in the field where access to a large computer is unlikely. The transfer of the entire memory contents would be accomplished by the IFPDAS II's 8080 microprocessor in a matter of minutes, thus, freeing the bubble memory for another mission. The data from the first mission is conveniently stored for later analysis.

The tradeoffs between a magnetic bubble memory system and digital cassette storage systems should be investigated further. This could include acquiring a cassette tape system and interfacing it to the IFPDAS. The performance of both systems could then be directly compared in terms of speed, ease of data transfer, power consumption, etc.

There are several methods to select the "important" data for reduction and storage, in addition to the one selected for the IFPDAS
II prototype. Two others are a breath-by-breath analysis and a "store-on-significant-change" technique. The breath-by-breath analysis would be similar to the method used in the prototype; except that the data would be stored every breath instead of every ten seconds. A digital differentiation technique could be employed to detect significant changes in selected parameters. The operating system would then analyze the change, then collect and store any required data. A thorough analysis of expected flight profiles must be made in order to determine which of the acquisition techniques, or combination of techniques, would be most desirable for a given profile. (This profile analysis may also effect the final choice of memory system.)

An absolute pressure sensor and a bi-directional flow sensor should be acquired and interfaced to the IFPDAS II prototype. With these sensors, and the present OM11 sensors, a more accurate analysis of the oxygen consumption can be made. A suggested method incorporating a single breath analysis is given in Appendix I.

The prototype should be reduced to the IFPDAS II size restrictions. This will include eliminating the unnecessary components on the SBC 80/20 board; and should include consideration of newly developed chips which combine the functions of several chips into one unit. (An example is Intel's new 8085 microprocessor and component family. The components on the SBC 80/20 board that are needed for the IFPDAS II system can be replaced by three 8085 system chips. These components use only 3 watts of power, can be placed on a 3.25" x 4.5" printed circuit board, and are 100% software compatible with the 8080 (Ref 6:109-113).) Once the configuration is reduced, the power consumption should be carefully
considered. At this time, the complete IFPDAS II can be evaluated against the design criteria specified by SAM.

Other areas that should be investigated further include: the capability to record the pilot's voice and synchronize it with the data; a probe check routine which would give a visual indication if the probes were not properly interfaced; and a standard test device to bench-check the entire IFPDAS II operation.
Bibliography


19. ———. Beckman Instruments, Inc. (telephone interview) 6 October 1977.


29. Celler, B. G. and Bason, P. T. "Hardware Detection of P-waves
   in the Electrocardiogram," Medical and Biological Engineering:
   501-507 (September 1976).

30. James, Gordon W., Paul, M. H., and Wessel, H. U. "Precision
    Digital Heart Rate Meter," Journal of Applied Physiology, 32 (5):
    718-723 (May 1972).

31. The TTL Data Book for Design Engineers (Second Edition). Dallas,
Appendix A

System Characteristics
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Desired Range</th>
<th>IFPDAS II Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Rate</td>
<td>50 to 180 ± 2 b/min</td>
<td>53 ± .1 to 225 ± 2.2 b/min</td>
</tr>
<tr>
<td>Breathing Rate</td>
<td>10 to 30 ± 2 b/min</td>
<td>4.7 ± .1 to 50 ± 1 b/min</td>
</tr>
<tr>
<td>Flow Rate</td>
<td>N/S</td>
<td>0 to 124 ± .25 l/min *</td>
</tr>
<tr>
<td>Flow Rate Integral</td>
<td>N/S</td>
<td>0 to 26.2 volt-sec (± .3%)</td>
</tr>
<tr>
<td>Minute Ventilation Volume</td>
<td>5 to 35 ± 2 l/min</td>
<td>0 to 100 ± 2 l/min **</td>
</tr>
<tr>
<td>Oxygen Partial Pressure</td>
<td>N/S</td>
<td>0 to 760 ± 2 mm Hg</td>
</tr>
<tr>
<td>Absolute Pressure</td>
<td>N/S</td>
<td>0 to 760 ± 2 mm Hg *</td>
</tr>
<tr>
<td>G's</td>
<td>-3 to +12 ± .25 G's</td>
<td>-3 to +12 ± .1 G</td>
</tr>
</tbody>
</table>

N/S = Not Specified

* Assuming accurate probe input and ± ½ bit accuracy (± 10 mv)

** At sea level, assuming an accurate probe input
### Table II

**Prototype Power Supplies**

<table>
<thead>
<tr>
<th>SBC 80/20</th>
<th>+ 5 VDC ± 5% @ 3.5 Amps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- 5 VDC ± 5% @ .180 Amps</td>
</tr>
<tr>
<td></td>
<td>+ 12 VDC ± 5% @ .467 Amps</td>
</tr>
<tr>
<td></td>
<td>- 12 VDC ± 5% @ .123 Amps</td>
</tr>
</tbody>
</table>

**DAS 1128 and Sensor Interfaces:**

+ 15 VDC ± 3% @ .050 Amps
- 15 VDC ± 3% @ .100 Amps
+ 5 VDC ± 3% @ .500 Amps

### Table III

**Memory Map**

<table>
<thead>
<tr>
<th>Hexadecimal Address</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 - 03DF</td>
<td>Operating System</td>
</tr>
<tr>
<td>03EO - 03FF</td>
<td>Interrupt Jump Table</td>
</tr>
<tr>
<td>0400 - 0FFF</td>
<td>Unused PROM</td>
</tr>
<tr>
<td>1000 - 37FF</td>
<td>Not Used</td>
</tr>
<tr>
<td>3800 - 3BFF</td>
<td>Unused RAM</td>
</tr>
<tr>
<td>3C00 - 3C2F</td>
<td>CPU Scratch Pad Area</td>
</tr>
<tr>
<td>3C30 - 3F4F</td>
<td>Unused RAM</td>
</tr>
<tr>
<td>3F50 - 3F80</td>
<td>CPU Stack Area</td>
</tr>
<tr>
<td>3F81 - 3FFF</td>
<td>Unused RAM</td>
</tr>
</tbody>
</table>
## Table IV

### Pin Connections for DAS 1128 Interface

<table>
<thead>
<tr>
<th>SBC 80/20 Board</th>
<th>Component Board</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal</strong></td>
<td><strong>J1 Pin Connection</strong></td>
</tr>
<tr>
<td>Port 2 - Bit 7</td>
<td>2</td>
</tr>
<tr>
<td>Bit 6</td>
<td>4</td>
</tr>
<tr>
<td>Bit 5</td>
<td>6</td>
</tr>
<tr>
<td>Bit 4</td>
<td>8</td>
</tr>
<tr>
<td>Bit 3</td>
<td>10</td>
</tr>
<tr>
<td>Bit 2</td>
<td>12</td>
</tr>
<tr>
<td>Bit 1</td>
<td>14</td>
</tr>
<tr>
<td>Bit 0</td>
<td>16</td>
</tr>
<tr>
<td>IBFA</td>
<td>18</td>
</tr>
<tr>
<td>STROBE</td>
<td>20</td>
</tr>
<tr>
<td>IBFB</td>
<td>22</td>
</tr>
<tr>
<td>TRIG</td>
<td>24</td>
</tr>
<tr>
<td>STBA</td>
<td>26</td>
</tr>
<tr>
<td>N/C</td>
<td>28</td>
</tr>
<tr>
<td>N/C</td>
<td>30</td>
</tr>
<tr>
<td>STBB</td>
<td>32</td>
</tr>
<tr>
<td>Port 1 - Bit 7</td>
<td>34</td>
</tr>
<tr>
<td>Bit 6</td>
<td>36</td>
</tr>
<tr>
<td>Bit 5</td>
<td>38</td>
</tr>
<tr>
<td>Bit 4</td>
<td>40</td>
</tr>
<tr>
<td>Bit 3</td>
<td>42</td>
</tr>
<tr>
<td>Bit 2</td>
<td>44</td>
</tr>
<tr>
<td>Bit 1</td>
<td>46</td>
</tr>
<tr>
<td>Bit 0</td>
<td>48</td>
</tr>
<tr>
<td>N/C</td>
<td>50</td>
</tr>
</tbody>
</table>

Odd numbered pins are GND
<table>
<thead>
<tr>
<th>Signal</th>
<th>J2 Pin Connection</th>
<th>J4 Pin Connection</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port 2 - Bit 7</td>
<td>2</td>
<td>2</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 6</td>
<td>4</td>
<td>4</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 5</td>
<td>6</td>
<td>6</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 4</td>
<td>8</td>
<td>8</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 3</td>
<td>10</td>
<td>10</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 2</td>
<td>12</td>
<td>12</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 1</td>
<td>14</td>
<td>14</td>
<td>N/C</td>
</tr>
<tr>
<td>Bit 0</td>
<td>16</td>
<td>16</td>
<td>N/C</td>
</tr>
<tr>
<td>IBFA</td>
<td>18</td>
<td>18</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>20</td>
<td>20</td>
<td>N/C</td>
</tr>
<tr>
<td>IBFB</td>
<td>22</td>
<td>22</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>24</td>
<td>24</td>
<td>N/C</td>
</tr>
<tr>
<td>STBA</td>
<td>26</td>
<td>26</td>
<td>STB</td>
</tr>
<tr>
<td>N/C</td>
<td>28</td>
<td>28</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>30</td>
<td>30</td>
<td>N/C</td>
</tr>
<tr>
<td>STBB</td>
<td>32</td>
<td>32</td>
<td>N/C</td>
</tr>
<tr>
<td>Port 1 - Bit 7</td>
<td>34</td>
<td>34</td>
<td>B1 (MSB)</td>
</tr>
<tr>
<td>Bit 6</td>
<td>36</td>
<td>36</td>
<td>B2</td>
</tr>
<tr>
<td>Bit 5</td>
<td>38</td>
<td>38</td>
<td>B3</td>
</tr>
<tr>
<td>Bit 4</td>
<td>40</td>
<td>40</td>
<td>B4</td>
</tr>
<tr>
<td>Bit 3</td>
<td>42</td>
<td>42</td>
<td>B5</td>
</tr>
<tr>
<td>Bit 2</td>
<td>44</td>
<td>44</td>
<td>B6</td>
</tr>
<tr>
<td>Bit 1</td>
<td>46</td>
<td>46</td>
<td>B7</td>
</tr>
<tr>
<td>Bit 0</td>
<td>48</td>
<td>48</td>
<td>B8 (LSB)</td>
</tr>
<tr>
<td>225 Hz Clock</td>
<td>50</td>
<td>50</td>
<td>Clock</td>
</tr>
</tbody>
</table>

Odd numbered pins are GND
### Table VI

**Hazeltine Interface Connections**

<table>
<thead>
<tr>
<th>Signal</th>
<th>J3 Pin Connection</th>
<th>Hazeltine Connection</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protective GND</td>
<td>2</td>
<td>1</td>
<td>Protective GND</td>
</tr>
<tr>
<td>Transmit Data</td>
<td>4</td>
<td>2</td>
<td>Transmit Data</td>
</tr>
<tr>
<td>Receive Data</td>
<td>6</td>
<td>3</td>
<td>Receive Data</td>
</tr>
<tr>
<td>Request to Send</td>
<td>8</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>Clear to Send</td>
<td>10</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>Signal GND</td>
<td>14</td>
<td>7</td>
<td>Signal GND</td>
</tr>
<tr>
<td>+12 volts</td>
<td>22</td>
<td>6</td>
<td>Data Set Ready</td>
</tr>
</tbody>
</table>

Unused pins are not shown

### Table VII

**Other Characteristics**

<table>
<thead>
<tr>
<th>Maximum analog inputs</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog input impedance</td>
<td>$&gt;10^{10}$ ohms</td>
</tr>
<tr>
<td>Analog input voltage</td>
<td>$0 - 5.12$ volts</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>20 Hz</td>
</tr>
<tr>
<td>Machine cycle time</td>
<td>465 nsec</td>
</tr>
<tr>
<td>Timer clock period</td>
<td>930 nsec</td>
</tr>
<tr>
<td>Heart rate clock</td>
<td>225 Hz</td>
</tr>
<tr>
<td>Transfer (baud) rate</td>
<td>1200 baud</td>
</tr>
</tbody>
</table>
Fig. 17. DAS 1128 Interconnections and Interface
Appendix B

Operating System

The operating system controls the operation of the IFPDAS II prototype hardware. Execution of the operating system by the CPU results in initialization of the system and acquisition, reduction, and storage of the data. The operating system selects the proper input sensor, starts the A/D conversion, and inputs the data for reduction and permanent storage. In addition, it ensures that the proper service routine receives the data.

A flow chart of the operating system starts on this page; a listing of the program follows the flow chart.

![Flow Chart](image)

**Fig. 19. Operating System Flow Chart (Sheet 1 of 9)**
Fig. 19. Operating System Flow Chart (Sheet 2 of 9)
PO2IN:

Correct probe?  
Yes → Add reading to running sum → Start the next conversion → PO2OUT ← NXTPR

No → 10

PO2OUT:

Correct probe?  
Yes → Add reading to running sum → Start the next conversion → FLWRT ← NXTPR

No → 10

1

Fig. 19. Operating System Flow Chart (Sheet 3 of 9)
Correct probe?

Yes

Add reading to running sum

Start the next conversion
GX ↔ NXTPR

Increment the breathing rate count

New breath?

Yes

Store count

Zero the count

No

Fig. 19. Operating System Flow Chart (Sheet 4 of 9)
Correct probe?

Yes

Average 8 readings

Yes

Is AVG < XMIN?

No

AVG → XMIN

Yes

Is AVG > XMAX?

No

AVG → XMAX

Start the next conversion

GY → NXTPR

Fig. 19. Operating System Flow Chart (Sheet 5 of 9)
Fig. 19. Operating System Flow Chart (Sheet 6 of 9)
GZ:

Fig. 19. Operating System Flow Chart (Sheet 7 of 9)
ABSPR:

Correct probe?

Yes

Store the reading

No

10

HEART:

Is there a new heart rate?

Yes

Average 8 readings

No

Store the average

CNTCK:

Is 10 sec up?

Yes

9

No

Halt

Wait for 50 msec timer interrupt

Fig. 19. Operating System Flow Chart (Sheet 8 of 9)
Fig. 19. Operating System Flow Chart (Sheet 9 of 9)
IFPDAAS II PROTOTYPE OPERATING SYSTEM

This program controls the operation of the IFPDAAS II prototype hardware. Execution of the operating system by the CPU results in initialization of the system and acquisition, reduction, and storage of the data. The operating system selects the proper input sensor, starts the A/D conversion, and inputs the data for reduction and permanent storage. In addition, it ensures that the proper service routine receives the data.

The operating system consists of the following program modules:

- PWRUP - CONFIGURES THE IFPDAAS II HARDWARE FOR THE DATA ACQUISITION FUNCTION
- TSMHS - RESETS THE 50 MSEC TIMER AND STARTS THE PROGRAM LOOP
- EOC - OBTAINS THE DATA AND PASSES IT TO THE APPROPRIATE SERVICE ROUTINE
- POZIN - INSPIRED OXYGEN PARTIAL PRESSURE SERVICE ROUTINE
- POZOUT - EXPIRED OXYGEN PARTIAL PRESSURE SERVICE ROUTINE
- FLWRT - FLOW RATE SERVICE ROUTINE
- BRRT - BREATHING RATE CALCULATIONS
GX - X-G's Service Routine
GY - Y-G's Service Routine
GZ - Z-G's Service Routine

ABSPr - Absolute Pressure Service Routine

HEART - Heart Rate Calculations

CNTCK - Ten Second Counter

STORE - Outputs the Data Every 10 Seconds

ERROR - Corrects Service Routine Probe Mismatches

The operating system includes the following subroutines:

STARTS CALLS STROBE AND TRIGGR
STROBE - Increments the DAS 1128 MUX Addr
TRIGGR - Starts the A/D Conversion
SUM - Performs Double Precision Addition
AVG - Performs Data Averaging
INIT - Initializes CPU Scratch Pad
DTOUT - Prepares Data for Output and Calls ASCII and PRINT
ASCII - Converts Binary Data to ASCII
PRINT - Outputs ASCII Character to the Console
### ASSEMBLER EQU STATEMENTS

<table>
<thead>
<tr>
<th>EQU Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>034F</td>
<td>MODE EQU 04EH</td>
</tr>
<tr>
<td>0377</td>
<td>PSTART EQU 027H</td>
</tr>
<tr>
<td>11E0</td>
<td>UART EQU 0E6H</td>
</tr>
<tr>
<td>3001</td>
<td>READY EQU 01H</td>
</tr>
<tr>
<td>30EC</td>
<td>CON EQU 0ECH</td>
</tr>
<tr>
<td>3F80</td>
<td>STACK EQU 3F80H</td>
</tr>
<tr>
<td>3F4</td>
<td>DATA EQU 0E4H</td>
</tr>
<tr>
<td>3F5</td>
<td>MUX EQU 0E5H</td>
</tr>
<tr>
<td>3F6</td>
<td>STAT1 EQU 0E6H</td>
</tr>
<tr>
<td>3F7</td>
<td>PRT1 EQU 0E7H</td>
</tr>
<tr>
<td>3F8</td>
<td>MFIN EQU 0E8H</td>
</tr>
<tr>
<td>3FA</td>
<td>STAT2 EQU 0EAH</td>
</tr>
<tr>
<td>3FC</td>
<td>PRT2 EQU 0EBH</td>
</tr>
<tr>
<td>3FE</td>
<td>AI1 EQU 0F6H</td>
</tr>
<tr>
<td>03F4</td>
<td>AI2 EQU 0F9H</td>
</tr>
<tr>
<td>03EB</td>
<td>C50FF EQU 0CH</td>
</tr>
<tr>
<td>03F5</td>
<td>C70FF EQU 0EH</td>
</tr>
<tr>
<td>03F6</td>
<td>AFFH EQU 020H</td>
</tr>
<tr>
<td>03F7</td>
<td>89FHSK EQU 002H</td>
</tr>
<tr>
<td>03FC</td>
<td>ST40N EQU 0DH</td>
</tr>
<tr>
<td>03FD</td>
<td>ST4OFF EQU 0CH</td>
</tr>
<tr>
<td>03FE</td>
<td>TR4ON EQU 0FH</td>
</tr>
<tr>
<td>040E</td>
<td>TR4OFF EQU 0EH</td>
</tr>
<tr>
<td>0417</td>
<td>ADCON EQU 0E7H</td>
</tr>
<tr>
<td>0439</td>
<td>CM43 EQU 036H</td>
</tr>
<tr>
<td>0470</td>
<td>C14C EQU 070H</td>
</tr>
<tr>
<td>0496</td>
<td>C243 EQU 086H</td>
</tr>
<tr>
<td>04C</td>
<td>CTR0 EQU 0DCH</td>
</tr>
<tr>
<td>101</td>
<td>CTR1 EQU 0DDH</td>
</tr>
<tr>
<td>130E</td>
<td>CTR2 EQU 0DEH</td>
</tr>
<tr>
<td>130F</td>
<td>TH0 EQU 0DFH</td>
</tr>
</tbody>
</table>

- **MODE SET FOR UART INIT**
- **CM0 INST TO RESET UART**
- **UART CONTROL PORT**
- **MASK FOR TRANSMITTER READY**
- **CONSOLE OUTPUT PORT**
- **INITIAL STACK POINTER VALUE**
- **MUX ADDR INPUT PORT**
- **CONVERTED DATA INPUT PORT**
- **STATUS PORT OF 8255 #1**
- **8255 #1 CONTROL PORT**
- **HEART RATE INPUT PORT**
- **STATUS PORT OF 8255 #2**
- **8255 #2 CONTROL PORT**
- **A255 CONTROL WORD 1**
- **A - MODE 1, INPUT 1**
- **B - MODE 1, INPUT 2**
- **A - MODE 1, OUTPUT 1**
- **B - MODE 1, OUTPUT 2**
- **PORT A OF 8255 #1**
- **PORT A OF 8255 #2**
- **PORT C OF 8255 #1**
- **PORT C OF 8255 #2**
- **PORT C OF 8255 #1**
- **VIA PORT C OF 8255 #1**
<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>EQU</th>
<th>16-bit Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>30F6</td>
<td>Interrupt CYD Word 1</td>
<td>EQU</td>
<td>0F6H</td>
<td></td>
</tr>
<tr>
<td>30F7</td>
<td>Interrupt CYD Word 2</td>
<td>EQU</td>
<td>0F7H</td>
<td></td>
</tr>
<tr>
<td>3004</td>
<td>Int Controller CHR Port 1</td>
<td>EQU</td>
<td>00H</td>
<td></td>
</tr>
<tr>
<td>090F</td>
<td>Int Controller CHR Port 2</td>
<td>EQU</td>
<td>0FH</td>
<td></td>
</tr>
<tr>
<td>0904</td>
<td>Int Mask Value</td>
<td>EQU</td>
<td>04H</td>
<td></td>
</tr>
<tr>
<td>330A</td>
<td>Masks Off Level 2</td>
<td>EQU</td>
<td>00H</td>
<td></td>
</tr>
<tr>
<td>032D</td>
<td>Int Mask Port</td>
<td>EQU</td>
<td>00H</td>
<td></td>
</tr>
<tr>
<td>110F</td>
<td>Breathing Rate Threshold</td>
<td>EQU</td>
<td>20H</td>
<td></td>
</tr>
<tr>
<td>122D</td>
<td>Set Flag Value</td>
<td>EQU</td>
<td>0FFH</td>
<td></td>
</tr>
<tr>
<td>33F0</td>
<td>Mask for MUX Addr</td>
<td>EQU</td>
<td>0F0H</td>
<td></td>
</tr>
<tr>
<td>1560</td>
<td>Probe Count (0 - 6)</td>
<td>EQU</td>
<td>06H</td>
<td></td>
</tr>
<tr>
<td>012D</td>
<td>End of Int CYD Word</td>
<td>EQU</td>
<td>020H</td>
<td></td>
</tr>
<tr>
<td>1311</td>
<td># of Output Bytes</td>
<td>EQU</td>
<td>0170</td>
<td></td>
</tr>
<tr>
<td>3309</td>
<td># of LOCs To Be Zeroed</td>
<td>EQU</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>287F</td>
<td>ASCII Code For &quot;CS.&quot;</td>
<td>EQU</td>
<td>7EH</td>
<td></td>
</tr>
<tr>
<td>101E</td>
<td>ASCII Code For &quot;PRINT&quot;</td>
<td>EQU</td>
<td>1EH</td>
<td></td>
</tr>
</tbody>
</table>
POWER UP ROUTINE

The power-up routine is executed when the hardware recognizes that power has been applied to the IFPDAS. (The power-up reset triggers off the +3 volt supply.) The #251 communication interface (USART), 3255 I/O ports, 8253 timers, 9259 interrupt controller, 16128 data acquisition module, and the CPU scratch pad storage area are all initialized.

The power-up routine starts the 50 usec timer; when it times out, the program loop is entered and the data acquisition function is begun.
**USART INITIALIZATION**

**MODE 4E**: 1 STOP 9IT
- NO PARITY
- 0 PITS
- 16x BAUD FACTOR

**CMD 37**: NO MUNT MODE
- NO INTERNAL RESET
- RTS HIGH
- RESET ERROR FLAGS
- NORMAL BREAK CHAR
- RECEIVE ENABLED
- DTR HIGH
- TRANSMIT ENABLED

**STACK POINTER INITIALIZATION**

SP STARTS AT $3F90H

```
0010 3E4E ORG 0
1010 MVI A,MODE \OUTPUT MODE SET TO USART
03E9 OUT USART
3E37 MVI A,PSTURY \OUTPUT RESET COMMAND WORD
03E9 OUT USART \TO USART
3183F LXI SP,STACK \INITIALIZES STACK POINTER
```
**8255 INITIALIZATION**

**8255 #1**

MODE 1 (STROBED I/O) 96

SET MODE

POFT A - 903E 1
PORT A - INPUT
C6, C7 - OUTPUT
POFT B - 903E 1
PORT B - INPUT

BITS C6 & C7 OFF

**8255 #2**

MODE 1 (STROBED I/O) 94

SET MODE

POFT A - 903E 1
PORT A - INPUT
C6, C7 - OUTPUT
POFT B - 903E 1
PORT B - OUTPUT

**INTERRUPT INITIALIZATION**

ICW1 F6 : JUMP TABLE AT
ICW2 03 : 03E0 TO 03FF
IMASK 04 : MASK OFF LEVEL 2

**Listing**

```
0019 3E5
1117 03A4
031E 3E63
1321 0339
0223 3E64
022F 0339

MVI A, ICW1 : OUTPUT COMMAND WORD 1
OUT ICWP1
MVI A, ICW2 : OUTPUT COMMAND WORD 2
OUT ICWP2
MVI A, IMASK
OUT MSKPT : OUTPUT MASK 4RD
```
TIMER INITIALIZATION

CONTROL WORD 36 1
SELECT COUNTER 0
LOAD 2 BYTES
MODE 3
BINARY COUNTER

CONTROL WORD 70 1
SELECT COUNTER 1
LOAD 2 BYTES
MODE 0
BINARY COUNTER

CONTROL WORD 86 1
SELECT COUNTER 2
LOAD 2 BYTES
MODE 3
BINARY COUNTER

SQUARE WAVE GEN SET TO 225 47
BAUD RATE SET TO 1200

227 3E35 MVI A,C6M3 :INIT COUNTER 0 FOR SQUARE
329 037F OUT THCP : WAVE GENERATOR
329 3EA3 MVI A,0ABH
32D 030C OUT CTR0 :ESTABLISHES 225 HZ CLOCK
33F 3E12 MVI A,012H : FOR HEART RATE CIRCUIT
131 030C OUT CTR0
137 3E70 MVI A,C1M0 :RESET TIMER 1
33F 039F OUT THCP
33F 3E95 MVI A,C2M3 :INIT COUNTER 2 FOR BAUD
33F 030F OUT THCP : RATE
33F 3E39 MVI A,56D :ESTABLISHES 1200 BAUD RATE
339 030E OUT CTR2 : FOR HAZELTIME CRT
13F AF XRA A
3340 030E OUT CTR2

79
DAS 1128 INITIALIZATION

SEQUENTIALLY TRIGGERED

ONE CONVERSION
READ MUX ADDRESS INFO
CONVERT UNTIL MUX ADDR EQUALS PRBCNT

| 0047 | C0C3D2 | PWRUP | CALL TRIGGR | STARTS A CONVERSION
| 0049 | C0E5 | IN | STAT1 | INPUT STATUS OF 8255 #1
| 0047 | 6E02 | ANI | 99FMSK | MASKS 1BF9
| 004A | C44500 | JZ | PWRUP | LOOP IF STILL CONVERTING
| 004C | DRES5 | IN | MUX | READS IN MUX ADDR & 4 LSB
| 004E | ESFO | ANI | MUXMSK | SAVES P7-84
| 0050 | FEF3 | CPI | PRBCNT | COMPARE MUX ADDR TO LAST
| 0052 | C65900 | JZ | IDA | JUMP IF MUX ADDR = PRBCNT
| 0055 | C09202 | CALL | STRTAD | START A CONVERSION
| 0058 | C34500 | JMP | PWRUP |
**THIS SECTION Initializes the Storage Area, Time, PINSM, POUTS, WRTAT, GXMAX, GYMAX, & GZMAX Are ZERoEd. Then OFFH Is LOADED INTO GXMIN, GYMIN, & GZMIN. 200D IS LOADED INTO THE TEN SEC COUNTER.**

```
3F9A     AF   IDA1: XRA     A
035C     21033C
125F     77
126G     23
1261     77
1262     C0E702
1265     21133C
126A     36C5
126A     AF
1069     160F
036D     23   IDA1: INX     H
036F     77
136F     15
0070     C25000
```

**THIS SECTION Starts the 50 MS TIMEp. When the Timer Expires the Program Loop Is ENTERed AT 99MS.**

```
0077     3E55
0075     0300
1077     3E01
1379     0393
3079     FA
0072     76
177D     00
```

- MVI     A,0E5H
- OUT     CTR1
- MVI     A,0D1H
- OUT     CTR1
- EI
- HLT
- NOP

- MOVES E5 TO .59 OF COUNTER
- MOVES P1 TO .59 OF COUNTER
- WAIT FOR 50 MS TIMER

81
PROGRAM LOOP

THE PROGRAM LOOP IS EXECUTED WHEN THE 50 µSEC TIMER EXPIRES. IT CONTAINS THE SERVICE ROUTINES - ONE FOR EACH PHYSIOLOGICAL AND ENVIRONMENTAL SENSOR. THE DATA IS COLLECTED BY THE END-OF-CONVERSION (EOC) ROUTINE WHICH AVERAGES 8 SAMPLES INTO 1 READING WHICH IT PASSES TO THE APPROPRIATE SUBROUTINE.

TIMER INTERRUPT HANDLER

THIS CODE IS EXECUTED WHEN THE
50 MS TIMER EXPIRES

THE TIMER IS RESET AND THE
FIRST CONVERSION (PO2 IN)
IS STARTED

NXTPR IS USED BY THE END OF
CONVERSION (EOC) ROUTINE
TO DETERMINE WHICH PROBE
WAS SAMPLED. THE ADDR OF
THE INDIVIDUAL PROBE SERVICE
ROUTINE IS STORED IN NXTPR.

| 007E | 3E71 | 053h | MVI | A, C16H | RESETS COUNTER 1 |
| 0260 | 030F |       | OUT | THCP   |               |
| 3782 | 3EE5 |       | MVI | A, 05H | MOVES E5 TO LS3 OF COUNTER |
| 33AF | 0303 |       | OUT | CTR1   |               |
| 0086 | 3E01 |       | MVI | A, 0DH | MOVES D1 TO MS3 OF COUNTER |
| 3119 | 0300 |       | OUT | CTR1   |               |
| 316A | 3E21 |       | MVI | A, EOC | RESETS "IN SERVICE" BIT |
| 01AC | 0304 |       | OUT | ICAP1  |               |
| 038F | C9902 |      | CALL | STRTAO |               |
| 13D1 | 21C300 |      | LXI | H, 02H |               |
| 3994 | 22113c | | SMLC | NXTPR  |               |
| 1397 | 33    |       | INX | SP     |               |
| 1394 | 33    |       | INX | SP     |               |
| 0039 | C39200 |       | JMP | EOC   |               |
END-OF-CONVERSION (EOC) ROUTINE

THIS CODE GENERATES EIGHT CONVERSIONS WHICH ARE SUMMED AND DIVIDED BY 8 TO OBTAIN AN AVERAGE READING - THUS REDUCING THE NOISE EFFECT.

THE SUM IS STORED IN MIL D CONTAINS THE COUNT OF THE CONVERSIONS, THE AVERAGE READING IS PASSED TO THE SERVICE ROUTINES IN E.

AFTER THE AVERAGE HAS BEEN COMPUTED, THE MUX ADDR IS OBTAINED AND MASKED OFF.

THEN A JUMP INDIRECT IS ACCOMPLISHED TO THE SERVICE ROUTINE ADDR STORED IN NXTPR.

**Assembly Code:**

```
0190 21300 EOC1 LXI H,
0192 00C302 CALL SUM
0194 09090 EOC2 JMP EOC1
0196 09E4 EOC2 IN DATA
0198 02F EOC2 IN X 8
019A 00C302 CALL SUM
019C 00C002 CALL AVG
019E 09E5 IN MUX
019F 09F0 ANI MUXMSK
01A1 24132 LMLD NXTPR
01A3 09 E9 PCNL
```

84
THIS IS THE SERVICE ROUTINE FOR PO2 IN

THE NEXT CONVERSION IS STARTED (PO2 OUT). THE ADDR OF PO2OUT IS STORED IN NXTPR. THE CURRENT READING OF PO2IN IS INPUT, THEN ADDED TO THE PREVIOUS SUM, AND STORED IN PINSM.

DOUBLE PRECISION ADDITION (DAD) IS USED

THE A REG CONTAINS THE 4JX ADDR IN BITS 7-4 (XDH)

33CA DSC1  PO2IN:  MVI  9,00H  INPUT DESIRED 4JX ADDR IN B
33CA 98       CMP  B  IS THE MUX ADDR 0 ?
337C C29982    JN7  ERROR  IIF NOT THERE IS AN ERROR
310F C09302    CALL SRTAD  START THE NEXT CONVERSION
310F 21E280    LXI  M,PO2OUT  GET ADDR OF "PO2 OUt"
310F 22113C    SHLD  NXTPR  STORE ADDR IN NXTPR
310F 2A023C    LMLD  PINSM  PLACE THE PO2IN RUNNING
310F 19        DAD  D  SUM IN MIL
310F 22023C    SHLC  PINSM  1NOTES: E HOLDS THE AVG DATA I OR D IS ZERO
00DF C39500    JMP  EOC  MIL = MIL + DIE

STOSES NEW SJM IN PINSM
THIS IS THE SERVICE ROUTINE
FOR PO2 OUT

THE NEXT CONVERSION IS STARTED
(FLOW RATE). THE ADDR OF
FLWRT IS STORED IN NXTPR.
THE CURRENT READING OF PO2OUT
IS INPUT, THEN ADDED TO
THE PREVIOUS SUM, AND STORED
IN POUTS.

DOUBLE PRECISION ADDITION
(DAD) IS USED

THE A REG CONTAINS THE 4JX ADDR
IN BITS 7-4 (XDH).

| 30F2 | 0610 | PO2OUT | 4VI | 9,10H | INPUT DESIRED 4JX ADDR IN B
| 30E4 | 30 | CMP | B | IS THE MUX ADDR 1 ?
| 30E5 | C29702 | JN7 | ERROR | IF NOT, THERE IS AN ERROR
| 30EA | C09082 | CALL | START | START THE NEXT CONVERSION
| 30ED | 21F900 | LVI | H,FLWRT | GET ADDR OF "FLOW RATE"
| 30EE | 22113C | SHLD | NXTPR | SERVICE ROUTINE
| 30F1 | 2A043C | LMLD | POUTS | STORE ADDR IN NXTPR
| 30F4 | 19 | DAD | D | PLACE THE PO2OUT RUNNING
| 30F7 | 22043C | SMLD | POUTS | SUM IN MSL
| 31F8 | C39309 | JMP | EOC | NOTE: F HOLDS THE AVG DATA
|                     |                     |     |       | IF D IS ZERO
|                     |                     |     |       | MSL = MSL + JXE
|                     |                     |     |       | STORES NEW SJH IN POUTS
THIS IS THE SERVICE ROUTINE FOR FLOW RATE

THE NEXT CONVERSION IS STARTED
THE ADDP OF GX IS STORED IN NXTPR
THE CURRENT READING OF FLOW RATE
IS INPUT, THEN ADDED TO THE
PREVIOUS SUM, AND STORED IN
FLRTS

DOUBLE PRECISION ADDITION (DAD)
IS USED

THE A RFG CONTAINS THE 4XJX ADDR
IN BITS 7-4 (XOH)

| 01FE | 029702 | FLRTS | MVI 9,20H |
| 0131 | C03022 | CALL STRTA |
| 0134 | 2140C1 | LXI H,GX |
| 0177 | 221130 | SHLD NXTPR |
| 010A | 24363C | LMLD FLRTS |
| 010D | 19 | DAD D |
| 010E | 22353C | SHLD FLRTS |

This page is best quality practicable
From copy furnished to DDC
THIS SECTION USES THE CURRENT READING OF FLOW RATE TO DETECT A NEW BREATH AND COUNT THE NUMBER OF INTERVALS BETWEEN BREATHS.

THE NEW BREATH IS DETECTED WHEN THE FLOW RATE READING IS FOUND TO BE BELOW THE THRESHOLD 4 TIMES IN A ROW.

THE NUMBER OF 50 MS INTERVALS BETWEEN BREATHS IS STORED IN BTNT.

E HOLDS THE FLOW RATE READING
TFLAG HOLDS THE THRESHOLD FLAG
THCNT HOLDS THE THRESHOLD COUNT
THCNT HOLDS THE 50 MS COUNT

0111 21173C  BRTT1: LXI  H,THCNT  ;GET ADDRESS OF 50 MS COUNT
0114  34      INR  H  ;INCREMENT THE COUNT
0115  7B      MOV  A,E  ;MOVE CURRENT READING TO A
0116  FE20    CPI  TRHLD  ;COMPARE TO THRESHOLD
0117  0A2501  JJC
0118  AF      XRA  A
0119  3C32183C STA  THCNT  ;STORE THRESHOLD COUNT
011E  3C32193G STA  TFLAG  ;AND THRESHOLD FLAG
0125  C93C00  JMP  EOC  ;DONE
0128  FEFF    CPI  TFLAG  ;IS IT SET?
012A  CA9D00  JZ  EOC  ;DONE IF ALREADY SET
0130  3A193C  LOA  TFLAG  ;GET THRESHOLD FLAG
0133  3C      INR  A  ;INCREMENT THE COUNT
0137  32193G STA  THCNT  ;STORE NEW THRESHOLD CNT
013C  FE14    CPI  4
013E  C93C00  JNZ  EOC  ;DONE IF .NE. 4
0140  3EFF    MOV  A,TFLAG
0144  32193C STA  TFLAG  ;GET THRESHOLD FLAG
0148  7E      MOV  A,THCNT  ; (MLC SET TO THCNT)
014F  21103C LXI  M,BTHRT  ;GET 50 MS CNT
0152  77      MOV  M,4  ;STORE COUNT FOR OUTPUT
0154  AF      XRA  4
0157  32173C STA  THCNT  ;ZERO THE 50 MS COUNT
015A  32193C STA  THCNT  ;AND THRESHOLD COUNT
015E  C93C00  JMP  EOC
THIS IS THE SERVICE ROUTINE FOR X-G'S

THE NEXT CONVERSION IS STARTED
 THE ADD OF GY IS STORED IN NXTPR
 THE CURRENT READING OF X-G'S
 INPUT, EIGHT READINGS ARE
 AVERAGED TO GIVE AN AVERAGE
 X-G'S READING.

IF THE AVG GT. GXMAX IT IS
 IS STORED IN GXMAX - IF LT.
 GXMIN IT IS STORED IN GXMIN

THE A REG CONTAINS THE MUX ADDR
 IN BITS 7-4 (XOH)

```
0140 0630 GX1: MOVE 9,3CH
0145 98 CMP 9
0150 C29402 JNZ ERROR
0155 209302 CALL STRTAO
0160 219501 LXI H, GY
0165 22113C SHLD NXTPR
0170 2A443C LNLX GX
0175 2T977 MOV A,E
0180 C00902 CALL XGCNT
0185 3A103C LOA INR A
0190 3C FE09 CPI B
0195 CA7501 JPZ GX1
0200 22143C SMID XG
0205 321C3C STA XGCNT
0210 39C300 JMP EOC
0215 2DCE02 GX11 CALL AVG
0220 AF STA XGCNT
0225 3213C STA XG
0230 32143C STA XG+1
0235 32193C STA A,E
0240 79 MOV M
0245 2193C LXI M,GXMAX
0250 9E CMP M
0255 0A1901 JC GX2
0260 77 MOV H, A
0265 2193C LXI H,GXMIN
0270 9E CMP M
0275 029300 JNC EOC
0280 77 MOV H, A
0285 C39300 JMP EOC
```
THIS IS THE SERVICE ROUTINE FOR Y-G'S

THE NEXT CONVERSION IS STARTED
(Z-G'S)

THE ADDR OF ZT IS STORED IN NXTPR
THE CURRENT READING OF Y-G'S IS
INPUT. EIGHT READINGS ARE
AVERAGED TO GIVE AN AVERAGE
Y-G'S READING.

IF THE AVG .GT. GYMAX IT IS
STORED IN GYMAX - IF .LT.
GYMIN IT IS STORED IN GYMIN

THE A REG CONTAINS THE MUX ADDR
IN BITS 7-4 (X0H)

0195 0641  GY1  4VI 9,40H  IPUT DESIRED MUX ADDR IN B
0196  98  CMP  9  IS THE MUX ADDR 4 ?
0197  C24902  JNY  ERROR  IF NOT, THERE IS AN ERROR
019C  000002  CALL  STRTAD  ISTART THE NEXT CONVERSION
019F  213F81  LXI  H,GZ  IGET ADDR OF "Z-G'S"
    ; SERVICE ROUTINE
01A2  22113C  LHLD NXTPR  ISTORE ADDR IN NXTPR
01A5  2A103C  LHLD YG  ILOADS HL WITH Y-G'S
    ; RUNNING SJM
01A8  79  MOV  A,E  IMOVES CURRENT READING TO A
01A9  C0C302  CALL  SUM  IACD DATA TO RUNNING SUM
01AC  3A1F3C  LOA  YGCNT  IGET Y-C COUNT
01AF  3C  INR A  INCREMENT THE COUNT
01B0  FE94F  CPI  B COMPARE WITH 8
01B2  CA4E81  JZ  GY1  JUMP IF YGCNT = EQ. 8
01B5  22133C  SHLD YG  ISTORE RUNNING SJM IN YG
01BA  321F3C  STA  YGCNT  ISTORE COUNT IN YGCNT
01BB  C39C00  JMP  AVG  IAVEAGE THE 8 READINGS
01BF  C0CE32  GY1  4FI  CALL  AVG  AVERAGE THE 8 READINGS
01C1  4F  XRA A  S0RE THE COUNT AND
01C2  321F3C  STA  YGCNT  THE RUNNING SUM
01C5  321D3C  STA  YG
01C8  321E3C  STA  YG+1
01CB  79  MOV  A,E  IMOVES AVG DATA TO A
01CC  21A43C  LXI  H,GYMAX  IGET ADDR OF MAX Y-G'S
01CF  9E  CMP M  ICMPARE AVG VALUE WITH MAX
01D0  DA4601  JC  GY2  JUMP IF A.LT.GYMAX
01D3  77  MOV  H,A  IPLACE IF A.LE.GYMAX
01D4  21993C  GY2  4FI  CALL  AVG  AVERAGE THE 8 READINGS
01D7  9E  LXI  H,GYMIN  IGET ADDR OF MIN Y-G'S
01DA  4  CMP  H  ICMPARE AVG VALUE WITH MIN
01DB  029C00  JNC  EOC  JUMP IF A.GT.GYMIN
01DE  77  MOV  H,A  IPLACE IF A.LT.GYMIN
01E1  C39C00  JMP  EOC
THIS IS THE SERVICE ROUTINE FOR Z-G'S

THE NEXT CONVERSION IS STARTED (ABS PRESS)

THE ADDR OF ABSPR IS STORED IN NXTPR

THE CURRENT READING OF Z-G'S IS INPUT. EIGHT READINGS ARE AVERAGED TO GIVE AN AVERAGE Z-G'S READING.

IF THE AVG .GT. GZMAX IT IS STOPPED IN GZMAX - IF .LT.

GZMIN IT IS STORED IN GZMIN

THE A REG CONTAINS THE YJX ADDR IN BITS 7-4 (XEN)

```
01DF 069C 674:  MOV 9,50H  INPUT DESIRED YUX ADDR IN B
01F1 0B8    CMP B  ITS THE YUX ADDR ?
01F2 C09362 JMP ERROR: IF NOT, THERE IS AN ERROR
01F3 C9002 CALL START: START THE NEXT CONVERSION
01F4 212902 LXI H,ABSPR  I GET ADDR OF "ABS PRESS"

: SERVICE ROUTINE

01FA 22113C SHLD NXTPR  ISTORE ADDR IN NXTPR
01FE 2A203C LHLD ZG  ILOADS HL WITH Z-G'S

: RUNNING SJM

01F1 79 MOV A,E  IMOVES CURRENT READING TO A
01F2 C0C902 CALL SUM  IADD DATA TO RUNNING SJM
01F3 3A223C LODA ZGCNT  IGET Z-G COUNT
01F4 3F39 JMP CPI  INCREMENT THE COUNT
01F5 C03702 JZ GZ1  IJUMP IF ZG CNT .EQ. 0
01F6 22203C SHLD ZG  ISTORE RUNNING SUM IN ZG
0201 32223C STA ZGCNT  ISTORE COUNT IN ZGCNT
020C C39C00 JMP EOC
0217 C0C002 G71: CALL AVG  IAVEARGF THE 9 READINGS
0220 AF XPA A  IZERO THE COUNT AND
0225 32223C STA ZGCNT  }: THE RUNNING SUM
022E 32213C STA ZG  I:
0231 32213C STA ZG+1
0234 79 MOV A,E  IMOVES AVG DATA TO A
023F 2103C LIXI H,GZMAX  IGET ADDR OF MAX Z-G'S
0244 9E CMP M  ICMPARE AVG VALUE WITH MAX
0247 041002 JC GZ2  IJUMP IF A.LT.GZMAX
024A 77 MOV M,A  IREPLACE IF A.GE.GZMAX
0251 21093C G72: LIXI H,GZMIN  IGET ADDR OF MIN Z-G'S
0256 9E CMP M  ICMPARE AVG VALUE WITH M
0259 029C00 JMP EOC  IJUMP IF A.GE.G7MIN
0264 77 MOV M,A  IREPLACE IF A.LT.GZMIN
026F C39C00 JMP EOC
```
THIS IS THE SERVICE ROUTINE
FOR ABS PRESS

THIS IS THE LAST CONVERSION
SO NO NEW CONVERSION IS
STARTED

THE CURRENT READING REPLACES
THE OLD. THE STORAGE
LOCATION IS CABPR

THE A REG CONTAINS THE MUX ADDR
IN BITS 7-4 (X0H)

027A 0663 A75PR1 4VI 9,60H 10UT DESIRED MUX ADDR IN 9
022A 49 CMP 9 IS THE MUX ADDR 6 ?
022B 023302 JN7 ERROR IF NOT THERE IS AN ERROR
022E 210E3C LXI M,CABPR 1GET ADDR OF CURRENT ABS PR
0231 73 MOV M,E 1STORE NEW VALUE IN CABPR
THIS IS THE SERVICE ROUTINE FOR HEART RATE.
THIS ROUTINE IS ENTERED RIGHT AFTER AB3 PRESS.

IF A NEW HEART RATE HAS BEEN COMPUTED (IBF=1) IT IS READ IN. EIGHT READINGS ARE AVERAGED TO GIVE AN AVERAGE HEART RATE. READINGS <LT. 10H ARE ASSUMED AS FALSE P-WAVES AND ARE ADDED TO THE COUNT TO BE AVERAGED - BUT DO NOT COUNT AS ONE OF THE 8. THE NEW AVERAGE HEART RATE IS THEN STORED.

```
0232 D9EA HEART1 IN STAT2 1 INPUT STATUS OF 8255 #2
0234 E620 ANI ABFMSK 2 MASKS IBFA
0235 CA6402 JZ CNTCK 3 JUMP IF NO NEW HEART RATE IS ENTERED
0239 D9E3 IN HRIN 4 INPUT THE NEW HEART RATE
023D 2F CHA 5 COMPLEMENTS DATA FROM INVERTING DRIVER (8226)
0240 2A143C LMLD 6 LOADS HRT WITH HEART RATE FROM UNPRES,
0243 D0C902 CALL 7 SUMS DATA TO RUNNING SUM
0247 FE10 CPI 8 10H PID (A HAS THE NEW READING)
024F D05A02 JC CNTCK 9 JUMP IF A <LT. 10H
0253 3A153C LOA HRCNT 10 GET HEART RATE COUNT
0257 3C INR A 11 INCREASE THE COUNT
0259 FE89 CPI 3 12 JUMP IF HRCNT <EQ. 8
025D 04B902 JC HRT1 13 JUMP IF HRCNT <EQ. 8
0261 22143C SHLP HRT 14 STORE RUNNING SUM IN HRT
0265 32157C STA HRCNT 15 STORE COUNT IN HRCNT
0269 C35A02 JMP CNTCK 16
026D 0CCE02 HRT1 CALL 17 AVG AVERAGE THE 8 READINGS
027C 210F3C LXI H,#HRT1 18 GET ADDR OF CURRENT HEART RATE
027F 73 MOV M,E 19 STORE NEW VALUE IN HRT
0280 4F XRA A 20
0281 32153C STA HRCNT 21 ZERO THE COUNT AND
0284 32143C STA HRT 22 THE RUNNING SUM
0287 32153C STA HRT+1
```
THIS SECTION DECREASES THE 10 SEC COUNTER

EACH 10 SEC THE STORAGE ROUTINE IS CALLED AND THE COUNTER IS RELOADED

AFTER COMPLETION OF THIS CODE, THE PROGRAM HALTS AND WAITS FOR THE 50 MS TIMER TO EXPIRE

026A 21133C  CNTCK1    LXI  4,TENSC  GET ADDR OF 10 SEC COUNT
0267 35  DCR  M  DECIMENT THE COUNT
026F CA7402  JZ  STORE  JUMP IF 10 SEC HAS ELAPSED
0271 FA  EI
0272 76  MLI
0273 00  NGP  WAIT FOR 50 MS TIMER
THIS SECTION OUTPUTS THE STORED INFORMATION EVERY 13 SECONDS.

THE DATA IS OUTPUT IN THE FOLLOWING ORDER:

PREAMBLE (LS3 FIRST)

PINSM (LS³ FIRST)

POUTS (LS³ FIRST)

FLRTS (LS³ FIRST)

GMAX, GMIN, GYMAX, GYMIN

GZMAX, GZMIN, ABSR, 4RTRT, 97MRT

0274 3601  STOREF1 MVI H,2000 :SETS THE COUNTER TO 200
0275 240030 CALL LTHDE TIME :PLACE OUTPUT COUNT IN HIL
0276 23 INX M :INCREMENT THE COUNTER
0277 220030 SHLD TIME :STORE THE NEW OUTPUT COUNT
0278 210030 LXI H,TIME :SAVE ADDRESS OF FIRST BYTE
0279 1611 MVI 0,BYTES :LOAD D WITH NUMBER OF OUTPUT BYTES

0280 7E STOREF1 MOV A,M :MOVE THE DATA TO A
0281 CDF02 CALL DOUT :OUTPUTS DATA TO CONSOLE
0282 23 INX H :GET NEXT OUTPUT BYTE ADDRESS
0283 15 DCR 0 :DECREMENT THE BYTE COUNTER
0284 C23202 JNZ STOR1 :JUMP IF MORE OUTPUT
0285 DE7E MVI C,CSP :CAUSES THE DATA ON SCREEN
0286 0E14 CALL PRINT :TO BE WRITTEN ON TAPE
0287 CD1303 MVI C,PR :INITIALIZES PINSM TO GZMIN
0288 E1 CALL INIT :WAIT FOR 50 45 TIMER
0289 08 CALL HLT
0290 76 CALL MOP
0291 00
THE FOLLOWING ERROR ROUTINE IS EXECUTED WHENEVER THE ACTUAL MUX ADDR IS NOT THE SAME AS THE EXPECTED MUX ADDR.


NOTE: MXTTP STILL CONTAINS THE ADDR OF THE DESIRED SERVICE ROUTINE.

THE FOLLOWING ERROR ROUTINE IS EXECUTED WHENEVER THE ACTUAL MUX ADDR IS NOT THE SAME AS THE EXPECTED MUX ADDR.


NOTE: MXTTP STILL CONTAINS THE ADDR OF THE DESIRED SERVICE ROUTINE.

0299 C09302 ERROR CALL STRTAD :STPOSE AND CONVERT
029E D3E5 ERR1 IN STAT1 :INPUT STATUS OF 8255 #1
02A0 E602 ANI 33FMSK :MASKS IFB
02A2 C9E02 JZ ERR1 :LOOP IF STILL CONVERSION
02A5 D7E5 IN MUX :READS MUX ADDR & 4 LS8
02A7 E6F0 ANI MUXMSK :SAVE 87-84
02AE 78 CMP 8 :COMPARE NEW ADDR WITH 8
02A2 C29332 JNZ ERROR :TRY AGAIN IF NOT EQ.
02A0 C39C00 JMP EOC :REENTER PROGRAM LOOP WHEN DESIRED ADDR IS FOUND

96
SYSTEM SUBROUTINES

These subroutines are used by the operating system and the service routines to accomplish tasks common to more than one module. There are subroutines to control the DAS 112A (start a conversion and increment the MUX address), manipulate the data (summing and averaging), initialize the storage area, and output the data (formatting the data and sending it to the console).
This surro RTE TAKES INPUT DATA FROM REG A AND ADDS IT TO HL

02C0h 4Fh SUM1 MOV C, A
02CAh 0600h MVI 9, 0
02CCh 09h MOV DATA TO C
02CDh C9h ZERO OUT B

This subroutine takes input data from reg A and adds it to HL.

02CEh 1Ah MOV A, L
02CFh 0Fh RRC
02D0h 0Fh RRC
02D1h 02002h JNC AVG1
02D5h F6E0h ORI 0E0H
02D7h 691h ADI 1
02D9h 02002h JNC AVG1
02DC 24h IMP H
02DDh E61Fh ANI 1FH
02DFh 5Fh MOV E, A
02E0h 7Ch MOV A, H
02E1h 0Fh RRC
02E2h 3Fh RRC
02E3h 0Fh RRC
02E4h 33h ADD E
02E5h 9Fh MOV E, A
02E6h 09h MOVES AVG READING TO E

This subroutine takes the running sum located in HL and divides it by 8. The resulting average is returned to the calling program in reg E.
THIS SECTION ZEROS THE FOLLOWING LOCATIONS: PINH, POUTS, FL1TS, GXMN, GYMNS, GZMX.

OFFSET IS PLACED IN THE FOLLOWING LOCATIONS: GXMIN, GYMNS, GZMIN.

THIS SUBROUTINE SEPARATES THE 8 BITS OF DATA IN THE A REG INTO TWO 4 BIT NUMBERS - EACH OF THESE NUMBERS IS CONVERTED TO AN ASCII CHARACTER AND OUTPUTED TO THE CONSOLE.

(BORROWED FROM THE SBC 80P20 USERS GUIDE PAGE B-29 (REF 15))
THIS SUBROUTINE CONVERTS THE 4 LSB'S OF THE A REG (0-9,A-F) INTO THE CORRESPONDING ASCII CHARACTER (30-39,41-46)

THE DAA (DECIMAL ADJUST ACC) INSTRUCTION PERFORMS THE FOLLOWING:
1. IF THE VALUE OF THE 4 LSB'S OF THE ACC IS >9 OR IF THE AC FLAG IS SET, 6 IS ADDED TO THE ACC
2. IF THE VALUE OF THE 4 MSB'S OF THE ACC IS NOW >9, OR IF THE CY FLAG IS SET, 6 IS ADDED TO THE 4 MSB'S OF ACC

(TAKEN FROM THE 8B32 USERS GUIDE PAGE 9-33 (REF 15))

0311 E60F ASCII: ANI OFH ; SAVES 4 LSB'S (1 HEX CHAR)
0313 C690 ADI 90H ; INSURE THAT 4-F CAUSES 1 A CARRY
0315 27 DAA ; ADJUST CONTENTS OF A REG
0316 CE40 ADD IN CARRY AND ADJUST IN THE HIGHER 4 BITS
0318 27 DAA ; ADJUST CONTENTS OF A REG
0319 4F MOV C,A ; MOVE ASCII CHAR TO C
031A C9 RET

THIS SUBROUTINE WAITS UNTIL THE CONSOLE IS READY TO ACCEPT A CHARACTER AND THEN OUTPUTS THE CHARACTER (STORED IN C REG) TO THE CONSOLE

0319 09E0 PRINT: IN USART INPUT STATUS OF CONSOLE
031A E631 ANI READY IMASK FOR TRANSMITTER READY
031F CA1903 J7 PRINT $LOOP IF NOT READY
0322 79 MOV A,C $MOVE CHARACTER TO A REG
0323 03EC OUT CON $SEND TO CONSOLE
0325 C9 RET
| $03E0$ | $C300$ | ORG $03E0$ |
| $03F0$ | $C300$ | JMP $0$ |
| $03F3$ | $00$ | NOP |
| $03F4$ | $C300$ | JMP $0$ |
| $03F7$ | $00$ | NOP |
| $03F8$ | $C300$ | JMP $0$ |
| $03FC$ | $C300$ | JMP $T50MS$ |
| $0400$ | $00$ | JMP $0$ |
| $0403$ | $C300$ | JMP $0$ |
| $0404$ | $00$ | NOP |
| $0404$ | $C300$ | JMP $0$ |
| $040F$ | $00$ | NOP |
| $040F$ | $C300$ | JMP $0$ |
| $040F$ | $00$ | NOP |
| $040F$ | $C300$ | JMP $0$ |
| $040F$ | $00$ | NOP |

This section contains the jump table for the interrupt controller.
This section contains the storage area for the program (CPU scratch pad storage)

These are the locations that will be stored every 10 sec

| 3G10 | 0001 | TIME | ORG | 3G90H |
| 3G3C | 0003 | TIME | DW | 0 |

- Contains running output
- 1 count, each count represents 10 sec
- Contains po2in running sum
- Contains po2out running sum
- Contains flow rate running sum

| 3G30 | 0000 | PINSM | DW | 0 |
| 3G34 | 0003 | POUTS | DW | 0 |
| 3G3A | 0003 | FLRTS | DW | 0 |

- Contains max x-G's
- Contains min x-G's
- Contains max y-G's
- Contains min y-G's
- Contains max z-G's
- Contains min z-G's

| 3G08 | 00 | GMAX | DB | 0 |
| 3G0C | 00 | GMIN | DB | 0 |
| 3G06 | 00 | G4MAX | DB | 0 |
| 3G07 | 00 | G4MIN | DB | 0 |
| 3G0F | 00 | CMAX1 | DB | 0 |
| 3G0F | 00 | CMIN1 | DB | 0 |
| 3G10 | 00 | ATHRT | DB | 0 |

- Contains current abs pressure
- Contains current heart rate
- Contains breath rate
### Additional Storage

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Description</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3011 0003</td>
<td>MXTRI</td>
<td>DW 0</td>
</tr>
<tr>
<td>3017 00</td>
<td>TENVSC</td>
<td>DB 0</td>
</tr>
<tr>
<td>3014 0003</td>
<td>MRTI</td>
<td>DW 0</td>
</tr>
<tr>
<td>3016 03</td>
<td>MRCNTI</td>
<td>DR 0</td>
</tr>
<tr>
<td>3017 00</td>
<td>TcoonT1</td>
<td>DB 0</td>
</tr>
<tr>
<td>3018 00</td>
<td>TTHCNT1</td>
<td>DB 0</td>
</tr>
<tr>
<td>3019 00</td>
<td>TFLAG1</td>
<td>DB 0</td>
</tr>
<tr>
<td>301A 0003</td>
<td>XGI</td>
<td>DW 0</td>
</tr>
<tr>
<td>301C 00</td>
<td>XGCNT1</td>
<td>DB 0</td>
</tr>
<tr>
<td>301D 0003</td>
<td>YGI</td>
<td>DW 0</td>
</tr>
<tr>
<td>301F 00</td>
<td>YTCNT1</td>
<td>DB 0</td>
</tr>
<tr>
<td>3020 0003</td>
<td>ZGI</td>
<td>DW 0</td>
</tr>
<tr>
<td>3022 00</td>
<td>ZGCNT1</td>
<td>DB 0</td>
</tr>
</tbody>
</table>

- **3011 0003**: MXTRI (Contains EOC Service Routine Address)
- **3017 00**: TENVSC (Contains 10 Sec Count)
- **3014 0003**: MRTI (Contains Heart Rate Routine Address)
- **3016 03**: MRCNTI (Contains Heart Rate Count 0-8)
- **3017 00**: TcoonT1 (Contains 50 MS Count)
- **3018 00**: TTHCNT1 (Contains Threshold CNT)
- **3019 00**: TFLAG1 (Contains Threshold Flag)
- **301A 0003**: XGI (Contains X-5'S Running Sum)
- **301C 00**: XGCNT1 (Contains X-5'S Count 0-8)
- **301D 0003**: YGI (Contains Y-5'S Running Sum)
- **301F 00**: YTCNT1 (Contains Y-5'S Count 0-8)
- **3020 0003**: ZGI (Contains Z-5'S Running Sum)
- **3022 00**: ZGCNT1 (Contains Z-5'S Count 0-8)

No program errors.
SYMBOL TABLE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3007</td>
</tr>
<tr>
<td>A17</td>
<td>1095</td>
</tr>
<tr>
<td>AVG1</td>
<td>2203</td>
</tr>
<tr>
<td>BRT1</td>
<td>1123</td>
</tr>
<tr>
<td>C0H3</td>
<td>1056</td>
</tr>
<tr>
<td>C7DFF</td>
<td>100E</td>
</tr>
<tr>
<td>C8P</td>
<td>107E</td>
</tr>
<tr>
<td>D</td>
<td>3002</td>
</tr>
<tr>
<td>ECC</td>
<td>309C</td>
</tr>
<tr>
<td>E2P1</td>
<td>129E</td>
</tr>
<tr>
<td>FLWCT</td>
<td>10FF</td>
</tr>
<tr>
<td>G</td>
<td>317E</td>
</tr>
<tr>
<td>H</td>
<td>3004</td>
</tr>
<tr>
<td>HRT</td>
<td>1C14</td>
</tr>
<tr>
<td>ICP2</td>
<td>3C03</td>
</tr>
<tr>
<td>IDA1</td>
<td>3C04</td>
</tr>
<tr>
<td>L</td>
<td>3005</td>
</tr>
<tr>
<td>MSKP</td>
<td>1099</td>
</tr>
<tr>
<td>PINN</td>
<td>3C22</td>
</tr>
<tr>
<td>PPI1</td>
<td>3C07</td>
</tr>
<tr>
<td>PRINT</td>
<td>1319</td>
</tr>
<tr>
<td>RSTNP</td>
<td>1037</td>
</tr>
<tr>
<td>STAT2</td>
<td>0864</td>
</tr>
<tr>
<td>STOR</td>
<td>1274</td>
</tr>
<tr>
<td>TIME</td>
<td>127E</td>
</tr>
<tr>
<td>TMCH</td>
<td>1258</td>
</tr>
<tr>
<td>TRGON</td>
<td>1C5F</td>
</tr>
<tr>
<td>XG</td>
<td>1C14</td>
</tr>
<tr>
<td>Z6</td>
<td>3C20</td>
</tr>
</tbody>
</table>
Appendix C

IFPDAS II Prototype User's Guide

Data Collection

This section describes the step-by-step procedure necessary to operate the IFPDAS II prototype in its data acquisition function.

1. AC Power — ON to both the Hazeltine video terminal and the cassette tape

2. DC Power — +5 VDC supply: OFF
   all others (-5, +12, -12, +15, & -15 VDC): ON

3. Video Terminal — Parity: 1
   Full Duplex
   Baud Rate: 1200
   Clear the screen

4. Cassette Recorder — Insert tape in Recorder 2 and engage
   Select CONT — OFF LINE — PAGE
   Depress RESET, then REWIND
   Depress INTERLOCK and RECORD button
   and wait for tape to stop
   (RECORD button stays lighted)

5. +5 VDC Supply — ON

The +5 VDC power supply resets the SBC 80/20 hardware and the operating system starts to execute. Data will be output to the video terminal every ten seconds and transferred from the screen to the cassette tape under program control. The 8080 CPU directs the writing of this data using two control characters. The CONTROL SHIFT PERIOD (cs.) character (7EH) tells the Hazeltine that a command will follow. The PRINT character (1EH) directs the transfer from the screen to the tape.

When all of the desired data has been recorded:
1. DC Power — OFF (this terminates the IFPDAS's operation)

2. Push RESET (the tape will advance momentarily, then stop and the RECORD light will go out)

3. Press REWIND

4. AC Power — OFF (to both the video terminal and the cassette recorder)

The data is now recorded on the cassette tape and is ready for transfer to the main computer.

**Data Transfer**

This section describes the step-by-step procedure to transfer the data from the cassette tape to a permanent file on AFIT's computer system.

1. Video Terminal — Parity: 1
   Half Duplex
   Baud Rate: 300

2. Cassette Recorder — Select CONT — OFF LINE — PAGE
   Depress RESET, then REWIND

3. Using the terminal, LOGIN and enter EDITOR

4. Enter: CREATE, SUPRESS line numbers (C,S)

5. After system responds ENTER LINES, depress PLAYBACK on tape channel

6. After the data is transferred, send an "=" to release the CREATE mode

7. List the file and check for errors

The data is now in the edit file. To store it permanently, enter the following commands:

1. REQUEST,Q,*PF
2. SAVE,Q,NOSSEQ,O  (NOSEQUence, Overwrite)
3. CATALOG,Q,DATA,ID=(problem #),RP=(# of days to retain)

The data is now stored on disc for later use by the post-flight data conversion routine.

Data Conversion

This section describes the procedure to execute the compiled post-flight data conversion routine (COMPCONVERT), using the file DATA as the data.

Enter the following commands:

1. ATTACH,LGO,COMPCONVERT  (attaches COMPCONVERT as a local file called LGO)
2. ATTACH,AFITSUBROUTINES,ID=AFIT (attaches the AFIT subroutines as a local file called AFITSUB)
3. LIBRARY,AFITSUB
4. ATTACH,TAPE10,DATA (attaches DATA as a local file called TAPE10)
5. REWIND,LGO
6. REWIND,TAPE10
7. LGO (executes the post-flight data conversion routine)

When the program completes its execution, it will have created a local file called PLOT containing the output graphs. To send these graphs to AFIT's plotter, enter:

ROUTE,PLOT,TID=BB,FID=(xxx),DC=PT

This routing completes the data conversion process.
Appendix D

Post Flight Data Conversion Routine

This program takes the data from the cassette tape (TAPE10) and performs the post flight data conversions. In addition to the calculations, each program module formats the labels for the graphs and calls the plotting routines.

The following parameters are plotted versus time by this program:

- Cabin Absolute Pressure
- Cabin Altitude
- Z - G's
- Heart Rate
- Breathing Rate
- Minute Ventilation Volume
- Inspired Oxygen Quantity
- Expired Oxygen Quantity

A flow chart of the post flight data conversion routine is given in Figure 20; a listing of the program follows the flow chart.
Fig. 20. Post Flight Data Conversion Flow Chart (Sheet 1 of 2)
Fig. 20. Post Flight Data Conversion Flow Chart (Sheet 2 of 2)
POST FLIGHT DATA CONVERSION

This program takes the data from the cassette tape (Tape10) and performs the post flight data conversions. In addition to the calculations, each program module formats the labels for the graphs and calls the plotting routines.

The following parameters are plotted versus time by this program:
- Cabin Absolute Pressure
- Cabin Altitude
- Z-g's
- Heart Rate
- Breathing Rate
- Minute Ventilation Volume
- Inspired Oxygen Quantity
- Expired Oxygen Quantity

```plaintext
PROGRAM CONVERT(INPUT, OUTPUT, TAPE10)
INTEGER TIME1, TIME2, A(100, 15)
DIMENSION X(102), Y(102), IN(17), IBUF(103), P(107, 2)
NROW = 0
DO 20 I = 1, 100
    READ(10, 100) TIME1, TIME2, (A(I, J), J = 1, 15)
  100 FORMAT(1722)
    IF(EOF(10)) 3, 10
  30 K = TIME1 + 256*TIME2
    IF(K .NE. I) STOP "DATA MISALIGNED"
    NROW = NROW + 1
  20 CONTINUE
```
CABIN ABSOLUTE PRESSURE

THE CABIN ABSOLUTE PRESSURE IS CALCULATED USING THE FOLLOWING FORMULA:

\[
\text{CABIN ABSOLUTE PRESSURE} = \frac{(\text{DATA}) \times 760}{250}
\]

30  \text{IN(1)} = "ABS PRESSU"
    \text{IN(2)} = "RE VS TIME"
    \text{IN(3)} = "SUBJECT"
    \text{IN(4)} = "JGJ"
    \text{IN(5)} = "START TIME"
    \text{IN(6)} = "1 1415"
    \text{IN(7)} = "19 OCT"
    \text{IN(8)} = "77"
    \text{IN(9)} = "TIME"
    \text{IN(10)} = "(MIN)"
    \text{IN(11)} = "ABS PRESSU"
    \text{IN(12)} = "RE (MM HG)"
    \text{IN(13)} = ""
    \text{IN(14)} = ""
    \text{IN(15)} = ""
    \text{IN(16)} = ""
    \text{IN(17)} = ""

DO 60 I=1, NROW
    \text{Y(I)} = \text{A(I,13)} \times 760 \div 250,
    \text{B(I,1)} = \text{Y(I)}
    \text{X(I)} = I \div 6.

60  \text{CONTINUE}

\text{X(NROW+1)} = 0.
\text{Y(NROW+1)} = 250.
\text{X(NROW+2)} = 1.
\text{Y(NROW+2)} = 100.
\text{CALL PLOTS(1,1,1,10,4,4PLOT)}
\text{CALL PLOT (0,-4,-3)}
\text{CALL PLOT (0,0.03,-3)}
\text{CALL HGRAPH (X,Y,NROW,IN,-1,0,0)}
CABIN ALTITUDE

THE CABIN ABSOLUTE PRESSURE IS CONVERTED TO ALTITUDE (IN FEET) USING THE FOLLOWING RELATIONSHIP:

ALTITUDE (FEET) = 170,156 - 25,685 LN (ABS PR (MM HG))

ID(1) = " ALTITUDE"
ID(2) = " VS TIME."
ID(11) = " ALTITUDE"
ID(12) = " (FEET)"
DO 55 I=1,NROW
   Y(I)=170516-25685* (ALOG (3(I,1)))
   X(I)=I/6.
CONTINUE
X(NROW+1)=0.
Y(NROW+1)=5.
X(NROW+2)=1.
Y(NROW+2)=6503.
CALL PLOT (0.0,-4.,-3)
CALL PLOT (0.,0.03,-3)
CALL HGRAPH (X,Y,NROW,ID,-1,0,0)
Z - G'S

The acceleration data is converted to G's using the following formula:

\[
\text{ACCELERATION (G'S)} = \left( \frac{\text{DATA}}{15} \right) - 3 \]

250

The maximum and minimum values are plotted on the same graph for comparison.

ID(1) = "7 - G'S"
ID(2) = "VS TIME"
ID(11) = "MAX AND MIN"
ID(12) = "N 7 - G'S"
DO 120 I = 1, NROW
   Y(I) = (A(I, 11) * .05) - 3.
   X(I) = I/6.
120 CONTINUE
X(NROW + 1) = 0.
Y(NROW + 1) = 0.5
X(NROW + 2) = 1.
Y(NROW + 2) = .4
CALL PLOT (0, ., -4, , -?)
CALL PLOT (0, .0, 0.03, -3)
CALL HGRAPH (X, Y, NROW, ID, -1, 0, 0)

C

ID(1) = 0
DO 130 I = 1, NROW
   Y(I) = (A(I, 12) * .05) - 3.
   X(I) = I/6.
130 CONTINUE
X(NROW + 1) = 0.
Y(NROW + 1) = 0.5
X(NROW + 2) = 1.
Y(NROW + 2) = .4
CALL PLOT (0, ., -4, , -3)
CALL PLOT (0, .0, 0.03, -3)
CALL HGRAPH (X, Y, NROW, ID, 2, 1, 1)

115
HEART RATE

Each heart rate count represents 4.44 msec. The count is converted to the heart rate using the following equation:

\[
\text{Heart Rate (beats/min)} = \frac{1}{4.44 \text{ msec}} \times 60 \times \text{count}
\]

ID(1) = "HEART RATE"
ID(2) = "E VS TIME"
ID(11) = "HEART RATE"
ID(12) = "(B/MIN)"
DO 40 I = 1, NROW
  \( Y(I) = (223 \times 60.) / A(I, 14) \)
  \( X(I) = I / 6. \\
40 \)
CONTINUE
CALL PLOT(0., 4., 3)
CALL PLOT(0., 3., 3)
CALL HGRAPH(X, Y, NROW, ID, 1, 0, 0)

BREATHING RATE

Each count is worth 50 msec. The breathing rate is calculated as follows:

\[
\text{Breathing Rate (breaths/min)} = \frac{1}{50 \text{ msec}} \times 60 \times \text{count}
\]

ID(1) = "BREATH RATE"
ID(11) = "BREATH RATE"
ID(12) = "E (B/MIN)"
DO 50 I = 1, NROW
  \( Y(I) = (20. \times 60.) / A(I, 15) \)
  \( X(I) = I / 6. \\
50 \)
CONTINUE
CALL PLOT(0., 4., 3)
CALL PLOT(0., 3., 3)
CALL HGRAPH(X, Y, NROW, ID, 1, 0, 0)
MINUTE VENTILATION VOLUME

The minute ventilation volume is calculated using the IFDOAS-supplied summation and the absolute pressure data. The formula is:

\[
\text{VOLUME} = \frac{(\text{SUM})}{10 \text{ sec} (L)} = \frac{\text{SUM}}{9.114} \\
\text{SORT(ABS P\_R)}
\]

DO 70 I=1,NROW

9(I+3,2)=(A(I,1)+2*6*A(I,6))/(SQRT(B(I,1))*87.75)

CONTINUE

B(1,2)=3(4,2)
B(2,2)=3(4,2)
B(3,2)=3(4,2)
B(NROW+4,2)=B(NROW+3,2)
B(NROW+5,2)=B(NROW+3,2)
DO 30 I=1,NROW

3(I,2)=3(I,2)+9(I-4,2)+9(I+1,2)+3(I+2,2)+9(I+3,2)+9(I+4,2)

&

3(I,2)=3(I,2)+9(I+1,2)+3(I+2,2)+9(I+3,2)+9(I+4,2)

&

70 CONTINUE

80 CONTINUE

ID(1)="MIN VENT V"
ID(2)="OL VS TIME"
ID(11)="MIN VENT V"
ID(12)="OL (L)"
CALL PLOT (J,4,-3)
CALL PLOT (J,0.07,-3)
CALL HGRAPH (X,Y,NROW,ID,1,3,0)

117
INSPIRED OXYGEN QUANTITY

THE QUANTITY OF INSPIRED OXYGEN IS COMPUTED BY:

\[
\text{QUANTITY OF O}_2 \text{ (L) } = \frac{(PC2IN \text{ SUM})(\text{MIN VENT VOL})}{65.79 \text{ (ABS PR)}}
\]

\[
\text{IN}(1) = "OXYGEN \text{ INT}" \\
\text{IN}(2) = "AKE VS T" \\
\text{IN}(11) = "OXYGEN \text{ INT}" \\
\text{IN}(12) = "AKE (L)"
\]

DO 90 I=1,NROW

\[
Y(I) = \frac{((A(I,1) + 256*A(I,2)) * 3.04*B(I,2))}{(200.0*3(I,1))}
\]

\[
X(I) = I / 6.
\]

90 CONTINUE

CALL PLOT (0., -4., -3)
CALL PLOT (0., 0.03, -3)
CALL HGRAPH (X, Y, NROW, IN, 1, 0, J)

EXPIRED OXYGEN QUANTITY

THE QUANTITY OF EXPIRED OXYGEN IS COMPUTED BY:

\[
\text{QUANTITY OF O}_2 \text{ (L) } = \frac{(PO2OUT \text{ SUM})(\text{MIN VENT VOL})}{65.79 \text{ (ABS PR)}}
\]

\[
\text{IN}(1) = "OXYGEN \text{ EXP}" \\
\text{IN}(2) = "IREO VS T" \\
\text{IN}(11) = "OXYGEN \text{ EXP}" \\
\text{IN}(12) = "IREO (L)"
\]

DO 110 I=1,NROW

\[
Y(I) = \frac{((A(I,3) + 256*A(I,4)) * 3.04*B(I,2))}{(200.0*3(I,1))}
\]

\[
X(I) = I / 6.
\]

110 CONTINUE

CALL PLOT (0., -4., -3)
CALL PLOT (0., 0.03, -3)
CALL HGRAPH (X, Y, NROW, IN, 1, 0, J)

C STOP "YOU HAVE IT"
END

118
The following subroutines are used by the post flight data conversion routine to format and plot the graphs. These subroutines were borrowed from AFIT's EE 6.91 course.
SUBROUTINE IGRAPH(X,Y,ID,NO,NP,NS)
DIMENSION X(1),Y(1),ID(1) $ IF (NO.EQ.2) CALL PLOT(-1.85,2.10,-3)
IF (NO.EQ.2) GO TO 30 $ IF (NO.LT.0) GO TO 10
CALL SCALE(X,7,N,1) $ CALL SCALE(Y,5,N,1)
10 CALL PLOT(0.0,11.,2) $ CALL PLOT(8.5,11.,2)
CALL PLOT(3.5,0.,2) $ CALL PLOT(0.,0.,2)
CALL PLOT(1.35,1.35,-3) $ CALL PLOT(0.,0.30,-2)
IF(ID(1).EQ.000) GO TO 25
CALL PLOT(-1.0,-1,-3) $ CALL PLOT(0.0,-2,-2)
DO 20 I=1,7,2
20 CALL SYMBOL(I+1.5)*1.,.1,97,ID(I),93.,20)
CALL PLOT(0.,0.,7) $ CALL PLOT(1.,0.,2)
CALL PLOT(1.,2.,2) $ CALL PLOT(0.,2.,-2)
CALL PLOT(-1.,1,-3)
25 CALL PLOT(5.,8.9,2) $ CALL PLOT(-3.8,0.0,-2)
CALL SYMBOL(5.,-2.,1,13,0.,50) $ CALL PLOT(5.3,75.,-3)
CALL AXIS(0.,0.,ID(9),-20,7,90.,X(N+1),X(N+2))
CALL AXIS(0.,0.,ID(11),20,5,186.,Y(N+1),Y(N+2))
30 Y(N+2)=Y(N+2) $ CALL LINE(Y,X,N,1,NS,NS)
Y(N+2)=Y(N+2) $ CALL PLOT(1.85,-2.10,-3)
RETURN $ END
SUBROUTINE AXIS(XO,YO,L,NC,RL,ANG,RMIN,DR)
DIMENSION L(1) $ A=ANG*3.14159/180. $ OX=1*COS(A) $ DY=1*SIN(A)
IC=ISIGN(1,NC) $ NNC=IAPS(NC) $ R=.1 $ N=1 $ X=XO $ Y=YO$
10 CALL PLOT(X,Y,3) $ X=X+DX $ Y=Y+DY $ CALL PLOT(X,Y,2)
CALL PLOT(X-.21*DY*IC,Y+.21*DX*IC,2)
IF(N.EQ.5) CALL PLOT(X-.42*DY*IC,Y+.42*DX*IC,2)
IF(N.EQ.10) CALL PLOT(X-.70*DY*IC,Y+.70*DX*IC,2)
N=MOD(N,10)+1 $ R=R+.1 $ IF(R.LT.RL) GO TO 10
A=ANG-(IC+1)*45. $ DX=10.*OX $ DY=IC.*OY
C=-.175+.125*IC $ D=.19+.35*IC
X=XO+C*DX-D*DY $ Y=YO+C*DY+D*DX
P=1+MAX1(ARS(DM1N),ARS(DMIN+DR*RL)) $ R=90+10*(R)
IR=INT(ARS(R)) $ IF(R.LT.0.) IR=-(IR+1) $ IR=IR-MOD(IR,3)
D=DM1N/IC.*IR $ DP1=DP/10.*IR $ P=0.
20 FNCOD(E7,101,S) R1 $ CALL SYM3OL(X,Y,.07,S,A,7) $ R1=R1+DR1
X=X+DX $ Y=Y+DY $ R=R+1. $ IF(R.LT.PL) GO TO 20
R=(RL-.1*NNC)/2. $ C=.1+.5*IC
X=XG+R+J*OX-C*OY $ Y=YO+R+J*DY+C*NX
CALL SYM3OL(X,Y,1,1,ANG,NNC) $ IF(IR.EQ.0) RETURN
ENCOD(E7,102,S) $ CALL SYM3OL(399,999,.10,S,ANG,5)
CALL WHERE(X,Y,A)
ENCOD(E7,103,S) I? $ CALL SYM3OL(X,Y,.07,S,ANG,3)
101 FORMAT(F7,2)
102 FORMAT(5H+10)
103 FORMAT(I3)
RETURN $ END
SUBROUTINE SCALE (DATA, LENGTH, N, K)
REAL DATA(N), LENGTH, SF(F)
DATA SF/1.0, 2.0, 2.5, 5.0, 10.0/
DMIN=OMAX=DATA(1)
DO 10 I=1,N
10 IF (DATA(I).LT.DMIN) DMIN=DATA(I)
IF (DATA(I).GT.OMAX) OMAX=DATA(I)
DATA(N+1)=DMIN ; DATA(N+2)=1.0
IF (LENGTH.LT.0.0) RETURN
PANSE=(OMAX-DMIN)/LENGTH
SFFXP=INT(ALOS10(PANSE))
IF (2*PANSE.LT.1.0) SFFXP=SFFXP-1.0
SFMANT=2*PANSE*10.0**(-SFFXP)
DO 20 I=1,5
20 IF (SF(I).GT.SFMANT) GO TO 30
PRINT*, "SCALE: SCALE FACTOR ERROR. " $ RETURN
30 SFNICE=SF(I)*10.0**SFFXP
ADMJN=INT(OMIN/SFNICE)*SFNICE
IF (ADJMIN.GT.OMIN) ADJMIN=ADJMIN-SFNICE
IF ((OMAX-ADJMIN)/SFNICE.LT.LENGTH) GO TO 40
IF (I.LT.5) SFNICE=SF(I+1)*10.0**SFFXP
IF (I.EQ.5) SFNICE=20.0*10.0**SFFXP
ADMJN=INT(OMIN/SFNICE)*SFNICE
IF (ADJMIN.GT.OMIN) ADJMIN=ADJMIN-SFNICE
DATA(N+1)=ADJMIN $ DATA(N+2)=SFNICE $ RETURN $ END
Appendix E

SBC 80/20 Hardware Description (Ref 16)

Memory

There are two types of memory on the SBC 80/20 board: random access memory (RAM) and read-only memory (ROM). Eight INTEL 2113 static RAM devices provide 2048 (2K) X 8-bits of read/write storage. The RAM address space is located from 3800H to 3FFFH by jumper connection 120-121. Four Intel 2708 Erasable and Electrically Programmable Read Only Memory chips provide 4096 (4K) X 8-bits of ROM. The ROM address space is located from 0000H to OFFFH. A complete memory map is given in Table III, Appendix A. (The functional characteristics of the memories are given in the SBC 80/20 Hardware Reference Manual (Ref 16: Ch 3, 21-29).)

Parallel I/O Interface

Two 8255 Programmable Peripheral Interfaces provide the input ports from the DAS 1128 and the heart rate detector. The remainder of this section describes these interfaces as configured for this specific application. A complete operational summary of the 8255 is available in the hardware reference manual (Ref 16: Ch 3, 51-73).

The 8255 contains three 8-bit ports (A, B, and C). The operating system configures these ports to the strobed input mode (mode 1, control word 'B6H'). This configuration provides for two input ports (A and B) and a control port (C). Each input port contains an input latch to hold the received data while the control port consists of six control bits and two output bits. (The SBC 80/20 modifications
listed in the hardware reference manual (Ref 16: Ch 4, 17-18 and 28-29) were accomplished.)

8255 #1 interfaces the DAS 1128 to the SBC 80/20. Port A receives the 8 most significant bits (MSB); Port B receives the 4 least significant bits (LSB) and the 4 multiplexer address bits. The Port C control bits are used as follows:

\[ C_0: \] \text{INTR}_B - interrupt request (not used)

\[ C_1: \] \text{IBF}_B - "high" indicates the data has been loaded into the input latch

\[ C_2: \] \text{STB}_B - "low" loads the data into the input latch

\[ C_3: \] \text{INTR}_A - (not used)

\[ C_4: \] \text{STB}_A - (same as above)

\[ C_5: \] \text{IBF}_A - (same as above)

\[ C_6: \] (output) - set by control word '0DH'
reset by control word '0CH'

\[ C_7: \] (output) - set by control word '0FH'
reset by control word '0EH'

The \text{EOC} signal from the DAS 1128 provides the \text{STB}_A and \text{STB}_B signals.

Bit C_6 provides the \text{STROBE} signal to the DAS 1128; while bit C_7 provides the \text{TRIG} signal (see Appendix F). The complete pin assignments for this interface are given in Table IV, Appendix A.

8255 #2 interfaces the heart rate detector to the SBC 80/20. Port A receives the heart rate count; Port B is not used. The Port C control bits are used as follows:

\[ C_0 - C_2: \] (not used)

\[ C_3: \] \text{INTR}_A - (not used)
\[ C_4 : \overline{STB} \_A \]
\[ C_5 : \overline{IBF} \_A \]
\[ C_6 - C_7: \text{ (not used)} \]

A strobe signal (\(\overline{STB}\)) is generated by the heart rate detector when the count is completed (see Appendix G). The complete pin assignments for this interface are given in Table V, Appendix A.

The 8255 I/O port addresses are given in Table VIII; a complete I/O port addressing table is given in the hardware reference manual (Ref 16: Ch 2, 7).

**Table VIII**

<table>
<thead>
<tr>
<th>I/O Device</th>
<th>I/O Port Address (hexadecimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8255 #1</td>
<td></td>
</tr>
<tr>
<td>Port A</td>
<td>E4</td>
</tr>
<tr>
<td>Port B</td>
<td>E5</td>
</tr>
<tr>
<td>Port C</td>
<td>E6</td>
</tr>
<tr>
<td>Control</td>
<td>E7</td>
</tr>
<tr>
<td>8255 #2</td>
<td></td>
</tr>
<tr>
<td>Port A</td>
<td>E8</td>
</tr>
<tr>
<td>Port B</td>
<td>E9</td>
</tr>
<tr>
<td>Port C</td>
<td>EA</td>
</tr>
<tr>
<td>Control</td>
<td>EB</td>
</tr>
</tbody>
</table>
Serial I/O Interface

The 8251 USART provides the output port to the Hazeltine 2000 video terminal (the interface connector pin assignments are listed in Table VI, Appendix A). The remainder of this section describes this interface as configured for the IFPDAS II prototype. A complete summary of the 8251 is available in the hardware reference manual (Ref 16: Ch 3, 34-51).

The system software configures the 8251 as an asynchronous receiver/transmitter. A '4EH' mode instruction programs the USART to the asynchronous mode with 1 stop bit, no parity check, 8 transmitted bits, and a baud rate factor of 16X. A '37H' command instruction sets the request-to-send and data-terminal-ready signals high, enables the receive and transmit capabilities, and resets the error flags. Interval timer '2' supplies the baud rate clock (see the following section).

Interval Timers

The 8253 Programmable Interval Timer includes three separate counters (0, 1, and 2). Counter 0 is used as a frequency divider to produce the 225 Hz clock required by the heart rate circuit (see Appendix G). Control word '36H' configures counter 0 as a square wave rate generator. The counter is then loaded with 12ABH (4779) which produces the desired frequency. Counter 1 is used as a real time clock to inform the CPU of every 50 msec interval. Control word '70H' configures counter 1 to interrupt the CPU when the count is complete. The counter is loaded with D1E5H (53,733) which is equivalent to 49.97 msec. The time required for the CPU to handle the interrupt and reset the timer brings the total time between interrupts to 50.00 msec.
Counter 2 is used as a frequency divider to produce the baud rate clock required by the 8251 USART. Control word 'B6H' configures counter 2 as a square wave rate generator. The counter is then loaded with 0038H (56) which produces the desired frequency. A complete summary of the 8253 is contained in the hardware reference manual (Ref 16: Ch 3, 73-87).

**Interrupt Controller**

The 8259 Programmable Interrupt Controller provides the capability to recognize interrupt requests, and based on that request, to jump to any location in the memory map. This section describes the operation of the 8259 in the IFPDAS II prototype. A complete operational summary of the 8259 is given in the hardware reference manual (Ref 16: Ch 3, 87-110).

The 8259 uses a jump table stored in PROM (03EOH to 03FFH) to pass control to the interrupt handling routine. When the 50 msec timer expires, an interrupt request (IR3) is sent by the timer to the 8259. The 8259 accepts this request and sends an interrupt to the 8080 CPU. After the CPU acknowledges the request, the 8259 "calls" the fourth entry of the jump table which causes a branch to the 50 msec timer interrupt handler (TSOMS).

The operating system programs the 8259, during the power-up routine, to accomplish this task. Two initialization command words (ICW) are required to inform the 8259 of the location and length of the jump table. ICW1 (= 'F6H') and ICW2 (= '03H') "tell" the 8259 that the jump table starts at 03EOH and that the call address interval is 4. After the 8259 receives these two words, it is in the normal (fully nested) mode and is ready to operate.
The 8259 is programmed by the CPU to ignore IR2. IR2 is generated by counter 0 which has a special function when the SBC 80/20 monitor is executing (Ref 15:8). Since counter 0 is used for a different purpose in the prototype than in the monitor, the 8080 sends an operational command word (OCW) to the 8259. OCW1 (="04H") masks off IR2 so that this request is never "seen" by the 8259.

The final command sent to the 8259 by the operating system is an end of interrupt (EOI) command word (OCW2 = "20H"). This command resets the in-service bit (IS3) which allows IR3 to request another interrupt.
Appendix F

**DAS 1128 Hardware Description** (Ref 17)

The DAS 1128 is a complete self-contained miniature high speed data acquisition system which is described in the attached 8-page pamphlet. The module is hard-wired to its IFPDAS II prototype configuration; these modifications are listed in Table IX along with their function. These hard-wired modifications are also reflected in Figure 17, Appendix A. The complete operating characteristics are given in the pamphlet.
### Table IX

**DAS 1128 Hardwired Modifications**

<table>
<thead>
<tr>
<th>Function</th>
<th>Jumper Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16 single-ended inputs</strong></td>
<td>11B to 11T, 12B to 2B, 17B to 19T, 18T to 18B</td>
</tr>
<tr>
<td><strong>Full range scale 0 - 5.12 volts</strong></td>
<td>12T to 13B, 14T and 14B to 13T, 15B to 16B</td>
</tr>
<tr>
<td><strong>Full 12 bit operation</strong></td>
<td>28T to DIG GND</td>
</tr>
<tr>
<td><strong>Output code: Unipolar Binary</strong></td>
<td>17T to -15 volts, 29T (B1) is MSB</td>
</tr>
<tr>
<td><strong>Sequentially triggered multiplexer addressing</strong></td>
<td>24B to +5 volts, STROBE to 8255 #1, TRIG to 8255 #1</td>
</tr>
<tr>
<td><strong>Sequence 0 to 6, then repeat</strong></td>
<td>4 OUT and 2 OUT to external NAND gate, Output of NAND gate to 25B</td>
</tr>
<tr>
<td><strong>Highest accuracy</strong></td>
<td>CLK TRIM to DIG GND (provides 2.08 microsec/bit conversion time), DLY TRIM to DIG GND, ±15 V return to ANA RTN, +5 V return to DIG RTN</td>
</tr>
</tbody>
</table>
FEATURES
Complete Data Acquisition System
12 Bit Digital Output
16 Single or 8 Differential Analog Inputs
High Throughput Rate
Selectable Analog Input Ranges
Versatile Input/Output/Control Format
Low 3 Watt Power Dissipation
Small 3” x 4.6” x 0.375” Module

GENERAL DESCRIPTION
The DAS1128 is a complete self-contained miniature high speed data acquisition system. The compact 3” x 4.6” x 0.375” module provides the designer with an easily implemented solution to the data acquisition problem. It contains an analog input signal multiplexer, a sample-and-hold amplifier, a 12 bit A/D converter, and all of the programming, timing and control circuitry needed to perform the complete data acquisition function.

The DAS1128 is a high performance device which can digitize an analog signal to an accuracy of ±½LSB out of 12 bits, relative to full scale. It has ±8ppm/°C gain temperature coefficient, and the maximum throughput rate can be varied from 50,000 conversions/second for a 12 bit conversion from different analog input channels, to 200,000 conversions/second for a successive 4 bit conversion made on a single channel.

Figure 1. Functional Block Diagram

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

Route 1 Industrial Park; P.O. Box 280; Norwood, Mass. 02062
Tel: 617/329-4700
TWX: 710/394-6577

West Coast
213/695-1783

Mid-West
312/894-3300

Texas
214/231-5094
### SPECIFICATIONS (typical @ +25°C and ±15V unless otherwise noted)

**ANALOG INPUTS**
- Number of Inputs to Multiplexer: 16 Single Ended, 8 True-Differential, 16 Pseudo-Differential
- **Input Voltage (Full Scale Range):** 10V to 10V, 0V to +10V, -5V to +5V, 0V to -10V, -20V to +20V, 0V to +12V, -12V to +5.12V, or 0V to +5.12V.
- **Maximum Input Voltage:** ±15V
- **Input Current (per channel):** 1mA max
- **Input Impedance:** >100kΩ for "ON" channel
- **Input Capacitance:** 10pF for "OFF" channel
- **Input Fault Current (power off or MUX failure):** Internally limited to 20mA
- **Direct ADC Input Impedance:** 10kΩ for each input line

**TEMP. COEFFICIENTS**
- **Gain Error Relative to F.S.:** ±0.1LSB
- **Quantization Error:** ±0.1LSB
- **Differential Nonlinearity Error:** @ 33kHz throughput rate
- **Differential Nonlinearity Error:** @ 50kHz throughput rate
- **Noise Error:** ±0.1LSB
- **-PS to +PS Error Between Successive Channel Transitions:** ±0.1LSB

**TRAVIS DYNAMICS**
- **Throughput Rate (16 bits):** 50kHz (max)
- **MUX Crosstalk ("OFF" channels to "ON" channels):** >80dB down @ 1kHz
- **Differential Amplifier CMRR:** 65dB ±3% at 1kHz
- **SHA Acquisition Time to 0.01%:** 4.5sec max
- **SHA Aperture Uncertainty:** ±10msec
- **SHA Feedthrough:** 10dB @ 1kHz

**DIGITAL INPUT SIGNALS**
- **MUX Address Inputs (8, 4, 2, 1):** 10µsec time plus 1µsec for each input line
- **MUX ENABLE HI (Pin 18T):** High (logic "1") input enables MUX "HI" output for inputs 0 through 7
- **MUX ENABLE LO (Pin 17B):** High (logic "1") input enables MUX "LO" output for inputs 8 through 15
- **STROBE (Pin 26T or 25T):** Negative going transition initiates A/D conversion. STROBE 1 must be at a logic "1" to enable STROBE 2. STROBE 2 must be at a logic "1" to enable STROBE 1.
- **LOAD ENABLE (Pin 248):**
  - Low (logic "0") input allows next strobe command to sequentially advance MUX address register.
  - Low (logic "0") input allows next strobe command to update MUX address register according to external address inputs.

**OUTPUT CODING**
- **MUX Address Outputs:** (IE, +5V, 1-ft pin 18B, 19T through 22T)
- **DELAY OUT (Pin 23T):**
- **EOC (Pin 27B):**

**DIGITAL OUTPUT SIGNALS**
- **Parallel Outputs:**
- **Coding:**
- **MUX Address Outputs:** (IE, +5V, 1-ft pin 18B, 19T through 22T)
- **DELAY OUT (Pin 23T):**

**POWER REQUIREMENTS**
- **+15V: 13%**
- **-15V: 13%**
- **+5V: 15%**
- **-5V: 15%**
- **Power Supply Sensitivity:**
- **Gain:** ±2.0mV/V
- **Offset:** ±4.0mV
- **ENHANCE:** ±10mV/V

**ENVIRONMENT & PHYSICAL**
- **Operating Temperature:** 0°C to +70°C
- **Storage Temperature:** -25°C to +85°C
- **Relative Humidity:** Up to 95% non-condensing
- **RFI & EMI 6 sides (except connector area):**
- **Power Supply Voltage:** 5.0V ± 0.6% ± 0.375V

**PRICE**
- $295.00 (1-9), price includes mating right-angle connector.

---

\[1\] Warm-up time to reach accuracy is 5 minutes.

\[2\] Specifications apply only when tracking +15V and -15V supplies are used, and for slowly varying variations in power supply voltages.

Specifications subject to change without notice.
THEORY OF OPERATION

A block diagram of the DAS1128 is shown in Figure 1. Analog input signals are applied to the various inputs of the 16 channel CMOS multiplexer. This multiplexer in conjunction with the differential amplifier that follows it, can be configured by the user to accept 16 single-ended analog inputs, or 8 fully differential analog inputs. It can also be connected as a 16 channel "pseudo-differential" input device, which permits some of the benefits of differential operation while maintaining a 16 channel input capability.

The differential buffer amplifier is gain programmable by the user via jumpers and input selection and the commencement of a conversion. The user can thus set the selected analog input signal at a constant level while the A/D converter is making a conversion.

The A/D converter is a high speed 12 bit successive approximation device that has been designed using the Analog Devices' AD562, 12 bit integrated circuit D/A. The reference voltage for the conversion is supplied by an adjustable precision reference circuit that has a temperature coefficient of 5ppm/°C.

In addition to these basic functional blocks, the DAS1128 also contains all of the clock circuitry necessary to perform the complete data acquisition function. The internal clock can be externally adjusted to provide various throughput rates at different accuracies. Input channel addressing logic is provided, as is the capability to short cycle the A/D converter (i.e. perform conversions of less than 12 bits resolution). It is also possible for the user to adjust the time interval between input channel selection and the commencement of a conversion. The user can thus trade off speed vs. accuracy in the settling time of the multiplexer and sample-and-hold amplifier, as well as speed versus accuracy of the A/D converter.

Figure 2. Simplified Timing Diagram, Showing Time-Interval Assignments and Constants.

INPUT CONNECTIONS

As shown in Figure 3, three input configurations can be used. 16 single-ended inputs (3a) can be connected to the multiplexer, all referenced to analog gnd. In the second configuration (3b), the inputs are connected individually as 8 true differential pairs. In this case the differential amplifier is connected "Differentially" with the output of the MUX. Finally, a "Quasi-Differential" connection (3c) can be realized under favorable ground path conditions. In this configuration the differential amplifier Lo terminal is used as the ground return for all sensors. In each of these input schemes, it should be noted that the input multiplexer has been designed to protect itself and signal sources from both overvoltage failure and from fault currents due to power-off loading or MUX failure.

Full scale range of the DAS1128 may be set by appropriate jumper connections for 8 different ranges: 0 to +10V; 0 to +5V; 0 to +10.24V; 0 to +5.12V; -10 to +10V; -5 to +5V; -10.24 to +10.24V; -5.12 to +5.12V.

Note that 10.24 and 5.12 ranges are commonly used since conversion increments become 5mV/bit, 2.5mV/bit, and 1.25mV/bit.

MUX AND S/H DYNAMICS — OVERLAP MODE

The overlap mode is defined as the ability of MUX to accept a new channel address thereby selecting the next channel to be sampled while the previously acquired sample is being held by the S/H for conversion. The dynamic characteristics of the S/H circuit are shown in Figure 4. Maximum throughput rates are obtainable when a single channel is held at a single address and the channel is sampled repeatedly. In a dynamic condition, data-throughput rates obtainable are shown in Figure 5.

Figure 3. Signal Input Connections for Three Different Configurations.

Figure 4. Sample-Hold Parameters Defined and Specified.
SHORT CYCLE

It is possible to short cycle the DAS1128, i.e. stop the conversion after less than 12 bits. This can be done by connecting an external jumper between short cycle terminal and one of the output terminals. With shorter cycles the attainable throughput rate increases, see Figure 5. In short cycle operation the output terminals will decrease proportionately to the number of bits selected. Note the short cycle terminal must be grounded for full 12-bit operation.

GROUNDING CONSIDERATIONS

Attention should be given to the methods of connection for electrical returns and voltage reference points. Analog return (ANA RTN) and digital return (DIG RTN) are provided. The following rules should be applied when integrating the DAS1128 into the system.

1. If the ±15V power supply is floating (for optimum analog accuracy), connect its return to ANA RTN (Pin 2B or 2T). If the ±15V power supply is not floating, connect its return to DIG RTN (Pin 35T or 36T).

2. Connect the +5V supply return to DIG RTN (Pin 35T or 36T). If this supply also powers additional equipment, run separate, parallel returns to the equipment ground and to DIG RTN (Pin 35T or 36T).

3. To minimize signal grounding problems, single-ended input signals should only be returned to ANA RTN (Pin 2B or 2T). If this is not possible, then connect the input signals in either the “true differential” or “pseudo-differential” configurations (see Figure 3).

4. Connect computer ground to DIG RTN (Pin 35T or 35B). Use heavy wire or ground planes.

5. The computer chassis should be connected to the computer and power supply grounds at only one point.

6. Connect the third-wire ground from main AC power input to the computer power supply return.

GAIN AND OFFSET ADJUSTMENTS

The DAS1128 is calibrated with external gain and offset adjustment potentiometers connected as shown in Figure 7 and 8. The offset adjustment potentiometer has an adjustment range of at least ±10LSB’s, and the gain range adjustment potentiometer has an adjustment range of at least ±10LSB’s. Offset calibration is not affected by changes in gain calibration, and should therefore be performed prior to gain calibration. Proper gain and offset calibration requires great care and the use of extremely sensitive and accurate reference instruments. The voltage standard used as a signal source must be very stable. It should be capable of being set to within ±10LSB of the desired value at any point within its range.

These adjustments are not made with zero and full scale input signals, and it may be helpful to understand why. An A/D converter will produce a given digital word output for a small range of input signals, the nominal width of the range being one LSB. If the input test signal is set to a value which should cause the converter to be on the verge of switching between two adjacent digital outputs, the unit can be calibrated so that it does switch at just that point. With a high speed convert command rate and a visual display, these adjustments can be performed in a very accurate and sensitive way. Analog Devices’ Conversion Handbook gives more detailed information on testing and calibrating A/D converters.

OFFSET CALIBRATION

For unipolar operation set the input voltage precisely to +0.0012V and adjust the offset potentiometer until the converter is just on the verge of switching from 0000000000000000000000001. For ±5V bipolar operation set the input voltage precisely to -4.9988V; for ±10V units set it to -9.9976V. Adjust the offset
potentiometer, Figure 7, until Offset Binary coded units are just on the verge of switching from 000000000000 to 000000000001 and Two's Complement coded units are just on the verge of switching 100000000000 to 100000000001.

CLOCK RATE ADJUSTMENT
The clock rate may be adjusted for best conversion time/accuracy trade-off. The conversion time is varied by means of the external circuitry shown in Figure 9. An open CLK TRIM terminal (Pin 26B) results in 1.25μsec/bit nominal conversion time. A grounded CLK TRIM terminal (for highest accuracy) results in 2.08μsec/bit conversion.

Figure 7. Ext. Offset Adjustment

GAIN CALIBRATION
Set the input voltage precisely to +9.9963V for unipolar operation, +4.9963V for inputs of +5V or +9.9926V for inputs of ±10V. Note that these values are 1¼LSB’s less than nominal full scale. Adjust the 20k variable gain resistor, Figure 8, until Binary and Offset Binary coded units are just on the verge of switching from 111111111111 to 111111111111 and Two’s Complement coded units are just on the verge of switching from 011111111110 to 011111111111.

Figure 8. Ext. Ref. Adjustment

INT. RANGE ADJUSTMENT

SET THE INPUT VOLTAGE PRECISELY TO +9.9963V FOR UNIPOLAR OPERATION, +4.9963V FOR INPUTS OF +5V OR +9.9926V FOR INPUTS OF ±10V. NOTE THAT THESE VALUES ARE 1¼LSB’S LESS THAN NOMINAL FULL SCALE. ADJUST THE 20K VARIABLE GAIN RESISTOR, Figure 8, UNTIL BINARY AND OFFSET BINARY CODED UNITS ARE JUST ON THE VERGE OF SWITCHING FROM 111111111111 TO 111111111111 AND TWO’S COMPLEMENT CODED UNITS ARE JUST ON THE VERGE OF SWITCHING FROM 011111111110 TO 011111111111.

INT. OFFSET ADJUSTMENT

DELAY TIME ADJUSTMENT
The DLY OUT signal may be adjusted to vary the A/D converter triggering time by means of the external circuitry shown in Figure 10. An open DLY TRIM terminal (Pin 23B) results in a nominal delay time of 3.0μsec. A grounded DLY TRIM terminal (for highest-accuracy) results in 20μsec delay time nominal.

Figure 9. Clock Trim

Figure 10. Delay Trim

TABLE I

<table>
<thead>
<tr>
<th>INPUT CONFIGURATION</th>
<th>ANALOG INPUT CONNECTIONS</th>
<th>ANALOG INPUT RETURN</th>
<th>JUMPER CONNECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 Single-Ended Inputs (Figure 3a)</td>
<td>3T thru 10T and 3B thru 10B</td>
<td>All input returns to 2B or 2T</td>
<td>11B to 11T, 12B to 2B or 2T, 17B to 19T, 187 to 18T</td>
</tr>
<tr>
<td>8 Differential Inputs (Figure 3b)</td>
<td>3T thru 10T</td>
<td>3B thru 10B</td>
<td>11B to 12B, 17B to 18T to “1”</td>
</tr>
<tr>
<td>16 Pseudo-Differential Inputs (Figure 3c)</td>
<td>3T thru 10T and 3B thru 10B</td>
<td>Common Input return to 12B</td>
<td>11B to 11T, 17B to 19T, 187 to 18B</td>
</tr>
</tbody>
</table>

RECOMMENDED SET-UP PROCEDURE

1. Select input configuration, see Table I.

2. Select MUX address mode.
The method of addressing the multiplexer can be selected by connecting the unit as follows:

RANDOM. Set Pin 24B (LOAD ENB) to logic "0". The next falling edge of STROBE will load the address presented to Pins 19B through 22B (8, 4, 2, 1). The code on these lines must be stable during the falling edge of STROBE plus 100nsec.

SEQUENTIAL FREE RUNNING. Set to logic "1", Pin 24B (LOAD ENB) and 25B (CLR ENB). Connect Pin 27B (EOC) to Pin 24T (STROBE 1). Connect Pin 23T (DLV OUT) to Pin 27T (TRIG). Use Pin 26T (TRIG) as a run/stop control (i.e. A/D conversion will continue while TRIG is high and will stop while TRIG is low).

SEQUENTIAL TRIGGERED. Set to logic "1", Pins 24B (LOAD ENB) and 25B (CLR ENB). Connect Pin 24T (STROBE) to external triggering source. The multiplexer address register will automatically advance by one channel whenever a STROBE command is received. The initial channel can be selected by setting Pin 24B (LOAD ENB) to logic "0" during only one STROBE command. The multiplexer address will then be determined by the logic levels on Pins 19B through 22B (the external MUX address lines). Channel "0" can be selected as the initial channel by setting Pin 25B (CLR ENB) to logic "0" during only one STROBE command. The final channel can be selected by following the procedure presented in Figure 6.

3. Select A-D conversion/channel select sequence (see Figure 5).

   (1) NORMAL (input channel remains selected during its A/D conversion). Connect Pin 23T (DLV OUT) to Pin 27T (TRIG).

   (2) OVERLAP (next channel is selected during A/D conversion). Connect Pin 27B (EOC) to TTL compatible inverter input. Connect inverter output to Pin 24T (STROBE). Connect Pin 23T (DLV OUT) to Pin 27T (TRIG). Adjust the delay to at least 4usec greater than EOC, 20usec max (see Figure 10). The signal on Pin 26T (TRIG) serves as RUN/STOP control.

   (3) REPETITIVE SINGLE CHANNEL. After selecting the input channel to be repetitively sampled (see MUX ADDRESS MODE, above), set Pin 27T (TRIG) to logic "0". Connect Pin 26T (TRIG) to a triggering source. Conversion process is initiated by positive edge of TRIG command.

4. Select output resolution.

   a. Full 12 bit resolution: connect Pin 28T (SHT CVC) to Pin 35B (DIG RTN).

   b. Bn (Bn < 12) bit resolution: connect Pin 28T to the output pin for Bn + 1.

5. Select optimum throughput rate.
The system clock frequency and the STROBE to TRIG delay (if used) can be trimmed to optimize the accuracy/throughput rate trade-off. See Figures 9 and 10.

6. Select input voltage full scale range. See Table II.

7. Select output digital coding. See Table III.

---

**TABLE II**

<table>
<thead>
<tr>
<th>FOR FULL SCALE RANGE OF:</th>
<th>MAKE THE FOLLOWING CONNECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to +10V</td>
<td>12T to 2T; 14T to 14B to ADC Source*</td>
</tr>
<tr>
<td>0 to +10.24V</td>
<td>same as 0 to +10V, plus 15B to 16B.</td>
</tr>
<tr>
<td>0 to +5V</td>
<td>12T to 13B; 14T and 14B to ADC Source*</td>
</tr>
<tr>
<td>0 to +5.12V</td>
<td>same as 0 to +5V, plus 15B to 16B</td>
</tr>
<tr>
<td>-10V to +10V</td>
<td>12T to 2T; 14T to 15T; and 14B to ADC Source*</td>
</tr>
<tr>
<td>-10.24V to +10.24V</td>
<td>same as -10V to +10V, plus 15B to 16B</td>
</tr>
<tr>
<td>-5V to +5V</td>
<td>12T to 13B; 14T to 15T and 14B to ADC Source*</td>
</tr>
<tr>
<td>-5.12V to +5.12V</td>
<td>same as -5V to +5V, plus 15B to 16B</td>
</tr>
</tbody>
</table>

*ADC Source is usually Sample and Hold Output (13T), but may be any signal source including Diff. Amp. Output (13B) if Sample and Hold is not desired.

---

**TABLE III**

<table>
<thead>
<tr>
<th>OUTPUT CODE</th>
<th>CONNECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unipolar</td>
<td>Connect 17T to -15V</td>
</tr>
<tr>
<td>Binary</td>
<td>Use 29T (B1) for MSB</td>
</tr>
<tr>
<td>2's Complement</td>
<td>Connect 17T to -15V</td>
</tr>
<tr>
<td></td>
<td>Use 28B (B1) for MSB</td>
</tr>
<tr>
<td>Offset Binary</td>
<td>Connect 17T to -15V</td>
</tr>
<tr>
<td></td>
<td>Use 29T (B1) for MSB</td>
</tr>
<tr>
<td>1's Complement</td>
<td>Connect 17T to 2B</td>
</tr>
<tr>
<td></td>
<td>Use 28B (B1) for MSB</td>
</tr>
</tbody>
</table>

---

136
Figure 11. Timing for Non-Overlap Operation in Both Random and Sequential Addressing Modes. For Status Keys and Signal Condition Data, Refer to Box Below.

**SIGNAL CONDITIONS AND STATUS KEYS FOR FIGURES 11 AND 12.**

| CH. 2 | -3.415V | CODE 010 101 010 101 |
| CH. 3 | +10.235V | CODE 111 111 111 111 |
| CH. 0 | -10.240V | CODE 000 000 000 000 |
| CH. 1 | +3.410V | CODE 101 010 101 010 |

ADC SET UP FOR ±10.24V. INPUT, OFFSET BINARY. (FOR TWO’S COMPLEMENT, USE B1 FOR M.S.B.)

**KEY**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>May change</td>
</tr>
<tr>
<td>///</td>
<td>May change 0 to 1</td>
</tr>
<tr>
<td>///</td>
<td>May change 1 to 0</td>
</tr>
<tr>
<td>Q/R</td>
<td>Must be stable</td>
</tr>
</tbody>
</table>

Figure 12. Timing Diagram for Overlap Operation in the Sequential Addressing Mode. For Status Keys and Signal Condition Data, See Box at Right.
Outline Drawings and Pin Designations

**DAS1128 Connector Pin Diagram**

- **+15V**: 1, 18
- **-15V**: 17
- **ANA RIN**: 27, 28
- **CH D IN**: 2, 3, 4, 5
- **CH 1 IN**: 9, 10, 11, 12
- **CH 2 IN**: 9, 10, 11, 12
- **CV IN**: 9, 10, 11, 12
- **MUX IN OUT**: 117, 118
- **RANGE SEL**: 127, 128
- **ENABLE IN**: 181, 182
- **8 OUT**: 197, 198
- **4 OUT**: 207, 208
- **2 OUT**: 217, 218
- **1 OUT**: 227, 228
- **DLY OUT**: 237, 238
- **STROBE 1**: 247, 248
- **STROBE 2**: 257, 258
- **CLOCK**: 267, 268
- **AUX CYC**: 277, 278
- **AD 10**: 287, 288
- **BB OUT**: 317, 318
- **BB OUT**: 327, 328
- **DIG RTN**: 347, 348

**Dimensions shown in inches and (mm).**

- **Module Mounting Holes**: Use 5/32" or 3/16" Lo Flat No Hardware
- **Module Layout**: 30 equal spaces, 0.200" x 0.300"
- **PC Board Layout**: 2.000" (50.8)
- **PC Board**: 3.300" (83.8)

Typical Applications

**DAS1128 WITH MOTOROLA 6800**

- **Address Bus**: 22, 23
- **Data Bus**: 24, 25

**DAS1128 WITH INTEL 8080**

- **Address Bus**: 22, 23
- **Data Bus**: 24, 25

NOTE:

1. **8255 USED IN MODE 1 (STROBED IO)**
2. **PCB INDEXES MUX TO DESIRED CHANNEL**
3. **CS TO A, WHERE A IS AN ADDRES BIT OTHER THAN A9 OR A11**
4. **CPC INITIATES CONVERSION**
5. **EOC STROBE IN DATA AND MUX INFO**
6. **8255 SHOWN, HOWEVER 6228 CAN ALSO BE USED**
Appendix G

Sensor Interfaces

This appendix discusses the three circuits that were designed and built to interface the oxygen partial pressure sensors, the accelerometers, and the heart rate detector to the IFPDAS II prototype.

OM11 Sensor Interface (Ref 18)

The circuit in Figure 21 interfaces both Beckman OM11 polarographic sensors to the IFPDAS II prototype. The OM11 sensor is biased to -740 mvolts and produces a current (approximately 2 microAmps for air at room temperature) that is proportional to the oxygen partial pressure of the gas that surrounds the sensor. An amplifier converts this current source to a voltage output. The output signal ranges from 0 to 5.0 volts corresponding to oxygen partial pressures of 0 to 760 mm Hg; the response time for the sensor is 800 msec when exposed to a pure oxygen source.

The $D_1-R_1-R_2-R_3$ combination acts as a voltage regulator and divider, reducing the -15 volts to the necessary -740 mvolts sensor bias. The sensor current output is then amplified through amplifier $A_1$ to produce 1.0 volts (152 mm Hg). The $C_1-R_5$ combination determines the amplifier gain and provides low pass filtering with a 5 Hz cutoff frequency. Temperature compensation for the sensor is provided by a built-in 10K thermistor $(R_7)$ which, in parallel with $R_8$, automatically adjusts the gain for changing sensor temperature. Calibration of the
Fig. 21. OM11 Interface Circuit

AI - LM312
DI - IN753
R7 - 10k Thermistor
(white wires from sensor)
All res. 5%, un.

OM11

140
interface is accomplished using a 100% oxygen source and adjusting
potentiometer $R_9$ for a signal output of 5.0 volts.

If faster sensor response time is necessary, the circuit of
Figure 22 can be used to compensate the sensor. This circuit was
provided by Beckman Instruments (Ref 19) and it reduces the sensor
response time to 100 msec. Power consumption of this circuit can be
minimized by utilizing all UA776 amplifiers which can be biased for
microwatt power consumption by appropriately selecting the bias
resistors (Ref 20: Ch 8, 458–466).
Accelerometer Interfaces

The Statham F-15-340 accelerometer interface circuit is shown in Figure 23.

![Accelerometer Interface Circuit Diagram]

The output range of the circuit is 0 - 5 volts corresponding to -3 to +12 G's. Resistors $R_1$ and $R_2$ provide a 1.0 volt (3 G) offset of the bridge. With the accelerometer oriented as shown, $R_1$ is adjusted to provide a 1.33 volt (1 G) output signal. Feedback network $C_1-R_3$ determines the amplifier gain; and capacitor $C_1$ provides filtering to reduce the effects of transient G's. The filter cutoff frequency is approximately 20 Hz.
The output of the amplifier can be tested by changing the orientation of the accelerometer about the longitudinal axis and observing the corresponding signal outputs. With a 90° clockwise rotation, the probe senses a zero-G condition and the amplifier output signal is 1.0 volts. With an additional 90° rotation, the probe senses a negative-one-G condition and the output of the amplifier is .67 volts.

Heart Rate Detector

The heart rate detector consists of an ECG amplifier, R-wave detector, and digital interval counter. The analog portions are shown in Figure 24; the digital portion is shown in Figure 25.

The ECG amplifier increases a 1 mvolt ECG signal to a minimum of 1.5 volts. The differential ECG (leads 1 and 2) is input to amplifiers A1 and A2, and lead 3 is used as a reference to reduce the common mode voltage. Feedback networks R1–C1 and R2–C2 provide the gain and high frequency filtering. Potentiometer R4 allows the gain of the differential input stage to be varied between 30 and 100.

The amplifier signal is capacitively-coupled to amplifier A3 through C3 and C4. This helps eliminate DC baseline shifts. Amplifier A3 provides final signal amplification with a differential mode gain of 50. The ECG signal is then filtered through a double-pole, low pass, active filter using amplifier A4. The filter bandwidth is 60 Hz and its low frequency gain is unity.

Resistors R1, R2, and R7 to R10 were chosen within .5% tolerance to reduce the common mode amplifier gain. Resistors R5, R6, R11, and R15 set bias currents for low power (500 microwatts) operation of the MC1776 operational amplifiers.
Fig. 24. ECG Amplifier and R-wave Detector

A1, A2, A3, A4 - MC1776
A5 - LM339
Res. - 5 kΩ unless otherwise specified
Fig. 25. Digital Interval Counter

Fig. 26. R-wave Interval Counter Timing Diagram
The filtered ECG signal is input to an inverting comparator (A5). A reference voltage, adjustable by potentiometer \( R_{16} \), is compared to the ECG input and a pulse is output when the peak of the ECG is above the reference voltage. This pulse is then output to the digital portion of the circuit.

The detector circuit is adjusted by monitoring the ECG on a strip chart or oscilloscope. The amplifier gain is adjusted by \( R_4 \) to obtain an ECG peak magnitude of 2 volts. The threshold voltage can be adjusted by \( R_{16} \) to detect all of the R-waves without mis-firing on the noise.

The output of the R-wave detector is input to the digital counting circuitry (Figure 25). This circuit counts the number of 4.44 msec (1/225 Hz) periods between detected R-waves. The dual-D flip-flop configuration (U33) shapes the variable width R-wave pulse to a single 4.44 msec pulse. This pulse is input to a monostable multivibrator (U32) to generate a 1.40 microsec pulse for the 8255. The 4.44 msec pulse is also gated to clear the counter prior to subsequent counting pulses. (A timing diagram of this sequence is given in Figure 26.)

The oxygen and accelerometer circuits are biased at ±15 volts so that the DAS 1128 power supply can be used. A +5 volt power supply can be used with low power operational amplifiers (i.e., MC1776 or LM312) by splitting the +5 volt power supply to provide the necessary ±2.5 volts.
Two OM11 sensor interfaces, the accelerometer interface, and the heart rate detector are contained on a 2.75" x 4.75" component board which is housed in a 3" x 4" x 5" aluminum box. The amplifier and potentiometer orientation is shown in Figure 27. The interface box external connections are shown in Figure 28. The pin assignments for the 25-pin interface connector are given in Table X.
### Table X

**Chassis Pin Connections**

<table>
<thead>
<tr>
<th>External Connection</th>
<th>Pin #</th>
<th>Chassis Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/C</td>
<td>1</td>
<td>N/C</td>
</tr>
<tr>
<td>thermistor (white)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>OM11 #1 thermistor (green) (PO2IN)</td>
<td>3</td>
<td>to PO2IN</td>
</tr>
<tr>
<td>cathode (red)</td>
<td>4</td>
<td>amplifier</td>
</tr>
<tr>
<td>anode (black)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>N/C</td>
<td>6</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>7</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>8</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>9</td>
<td>N/C</td>
</tr>
<tr>
<td>accelerometer pin 2</td>
<td>10</td>
<td>to accelerometer</td>
</tr>
<tr>
<td>accelerometer pin 3</td>
<td>11</td>
<td>amplifier</td>
</tr>
<tr>
<td>+15 VDC (red)</td>
<td>12</td>
<td>+15 volt bus</td>
</tr>
<tr>
<td>-15 VDC (white)</td>
<td>13</td>
<td>-15 volt bus</td>
</tr>
<tr>
<td>thermistor (white)</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>OM11 #2 thermistor (green) (PO2OUT)</td>
<td>15</td>
<td>to PO2OUT</td>
</tr>
<tr>
<td>cathode (red)</td>
<td>16</td>
<td>amplifier</td>
</tr>
<tr>
<td>anode (black)</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>N/C</td>
<td>18</td>
<td>N/C</td>
</tr>
<tr>
<td>N/C</td>
<td>19</td>
<td>N/C</td>
</tr>
<tr>
<td>OM11 shields</td>
<td>20</td>
<td>GND</td>
</tr>
<tr>
<td>N/C</td>
<td>21</td>
<td>GND</td>
</tr>
<tr>
<td>accelerometer pin 4</td>
<td>22</td>
<td>GND</td>
</tr>
<tr>
<td>GND (black)</td>
<td>23</td>
<td>GND</td>
</tr>
<tr>
<td>accelerometer pin 1</td>
<td>24</td>
<td>+5 volt bus</td>
</tr>
<tr>
<td>+5 VDC (blue)</td>
<td>25</td>
<td>+5 volt bus</td>
</tr>
</tbody>
</table>
Appendix H

Magnetic Bubble Memory Interface Diagrams

These diagrams show the interfacing required to test the bubble memory with the prototype. The diagrams were supplied by Gerald Cox of Texas Instruments, Inc., Dallas, Texas.
Appendix I

Oxygen Consumption Calculations for Single Breath Analysis

Oxygen consumption is measured by analyzing the mass of the gases of inspiration and expiration. When laboratory conditions permit, precise measurement of inspired and expired flow volumes can be made, and their gaseous contents can be accurately analyzed. Oxygen consumption is then calculated by subtracting the quantity of oxygen exhaled from the quantity of oxygen inhaled (Ref 21:681-685).

Since the IFPDAS II is used in flight, precise analysis of the respiratory gases and oxygen consumption measurements become more difficult. The diluter-demand oxygen regulator provides a mixture of oxygen and air to the pilot. The regulator adds a minimum of 0% oxygen to a maximum of 30% oxygen at sea level. The mixture varies according to outlet flow and altitude as shown in Table XI (Ref 22). Because of varying mixtures, the oxygen consumption analysis involves measuring the inspired and expired volumes and flow-volume averaging the inspired and expired oxygen contents (Ref 23). This method allows a single breath analysis of oxygen consumption using IFPDAS measured parameters. The Beckman OM11 oxygen sensors provide measurements of oxygen partial pressure and the flow sensors provide measurements of flow volume. For this application, the OM11 sensors require the lead compensation discussed in Appendix G which decreases the sensor response time to 100 msec. The remaining discussion presents the mathematical analysis and the IFPDAS II implementation.

157
**Table XI**

**Oxygen Ratio**

<table>
<thead>
<tr>
<th>ALTITUDE (1000 FEET)</th>
<th>OUTLET FLOW (LITERS/MINUTE)</th>
<th>% OXYGEN ADDED FROM SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MINIMUM</td>
</tr>
<tr>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>135</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>135</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td>135</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>15</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>135</td>
<td>40</td>
</tr>
<tr>
<td>28</td>
<td>15, 50, 135</td>
<td>60</td>
</tr>
<tr>
<td>32</td>
<td>135</td>
<td>98</td>
</tr>
</tbody>
</table>

With diluter at 100%

All altitudes 15, 50, 135 98 100

(Ref 22)

The instantaneous expired oxygen volume is related to the partial pressure of oxygen by

\[
\frac{d V_{E02}}{dt} = \frac{d \left\{ V_E \cdot \frac{P_{E02}}{P_{Abs}} \right\}}{dt}
\]

(22)
where,

\[ V_{E_{O2}} = \text{Volume of expired oxygen (liters)} \]
\[ V_E = \text{Volume of expired air (liters)} \]
\[ P_{E_{O2}} = \text{Partial pressure of oxygen expired (mm Hg)} \]
\[ P_{Abs} = \text{Absolute pressure (mm Hg)} \]

Since,

\[ \frac{P_{E_{O2}}}{P_{Abs}} = F_{E_{O2}} \]  \hspace{1cm} (23)

where \( F_{E_{O2}} \) is the oxygen fraction in the expired air.

Using Eq (23), Eq (22) becomes,

\[ \frac{d V_{E_{O2}}}{dt} = \frac{d}{dt} \left( V_E \cdot F_{E_{O2}} \right) \]  \hspace{1cm} (24)

The flow-weighted expired oxygen volume then becomes,

\[ V_{E_{O2}} = \int F_{E_{O2}} \cdot V_E \, dt \]  \hspace{1cm} (25)

where \( V_{E_{O2}} \) is the expired volume of oxygen in liters/breath
A similar computation for inspired oxygen volume is made and the per breath oxygen consumption can then be computed:

\[ V_{O2} \text{ (liters/breath)} = V_{I_{O2}} - V_{E_{O2}} \]  

(26)

where,

- \( V_{O2} \) = Volume of oxygen consumed (liters/breath)
- \( V_{I_{O2}} \) = Inspired volume of oxygen (liters/breath)

The expired oxygen volume measurement is made using the curves of expired flow volume rate and oxygen partial pressure as shown in Figure 33. The per breath expired oxygen volume is the area under the \( (F_{E_{02}})(V_{E}) \) curve measured over the period of one breath. A trapezoidal approximation to the area under this curve will yield the flow-weighted expired oxygen volume:

\[ V_{E_{O2}} = \frac{1}{2} \left( \frac{F_{E_{O2,1}} \cdot V_{E,1}}{V_{E,1}} \right) + \sum_{i=2}^{n-1} \frac{F_{E_{O2,i}} \cdot V_{E,i}}{V_{E,i}} \]

\[ + \frac{1}{2} \left( \frac{F_{E_{O2,n}} \cdot V_{E,n}}{V_{E,n}} \right) t \]  

(27)

where,

- \( F_{E_{O2,i}} \) = Expired oxygen fraction value
Fig. 33. Flow Rate Weighted Expired Oxygen Measurement
\[ V_{E_i} = \text{Expired volume flow rate sample value (liters/min)} \]
\[ t = \text{Sample to sample width (min)} \]

A similar method is used to determine the inspired oxygen volume and the pilot's oxygen consumption is then computed:

\[ V_{O2} \text{ (liters/min)} = V_{I_{O2}} - V_{E_{O2}} \]  \hspace{1cm} (28)

The IFPDAS II operating system can be modified to store all of the parameters required for an accurate calculation of the oxygen consumption if an inspired flow sensor is acquired. The required flow-weighted sums are calculated as follows: As the inhaled breath is detected, the inhaled flow rate/oxygen-fraction product is calculated. These samples are summed over the period of the inspired breath and stored. A similar product-sum is calculated for the period of the exhaled breath and stored. These values are then available for altitude correction and integration on the ground.

A sampling rate of 20 Hz is adequate to insure minimal error for the trapezoidal approximation of the integral as the breathing rate averages only .2 Hz. This routine also allows ample time to perform the necessary multiplication and division routines, each requiring a maximum of 380 microsec. Measurement delays can be compensated for by temporarily storing sample values and time-adjusting them to provide maximum accuracy.
VITA

Joseph Gregory Jolda was born on 12 January 1948 in Worcester, Massachusetts. He graduated from high school in Webster, Massachusetts, in 1965 and attended Northeastern University, Boston, Massachusetts, from which he received the degree of Bachelor of Science in Electrical Engineering in June 1970. Upon graduation he was employed as an associate engineer for Amperex Electronic Corporation, Slatersville, Rhode Island. He also taught Mathematics and Circuit Theory at Worcester Junior College, Worcester, Massachusetts. He then entered active duty in September 1970, and received his USAF commission from OTS in January 1971. He completed pilot training and received his wings in April 1972. He served as a T-33 pilot with the 67th Tactical Reconnaissance Wing at Bergstrom AFB, Texas, and then as a T-37 instructor pilot with the 37th Flying Training Squadron at Columbus AFB, Mississippi, until entering the School of Engineering, Air Force Institute of Technology, in June 1976. He is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent Address: Wawela Park
Webster, Massachusetts 01570
VITA

Stephen John Wanzek was born on 23 September 1948 in Racine, Wisconsin. He graduated from high school in Northbrook, Illinois in 1966 and attended the University of Illinois from which he received the degree of Bachelor of Science in Electrical Engineering and Computer Science in June 1970. After graduation, he was employed as a system programming engineer for Automatic Electric Company, Northlake, Illinois. In November 1970 he entered Officer Training School and was commissioned in February 1971. He completed pilot training and received his wings in February 1972. He then served as a T-37 instructor pilot in the 37th Flying Training Squadron and as a T-37 flight examiner in the 14th Flying Training Wing, Columbus AFB, Mississippi until entering the School of Engineering, Air Force Institute of Technology, in June 1976. He is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent Address: 1551 Chapel Court
Northbrook, Illinois 60062
**Title:** AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II.

**Author:** Joseph J. Jolda, Stephen V. Wanzek

**Performing Organization:** Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB, Ohio 45433

**Period Covered:** December 1777

**Report Date:** December 1777

**Pages:** 178

**DISTRIBUTION STATEMENT:** Approved for public release; distribution unlimited

**ABSTRACT:**

This paper discusses a second generation microprocessor-based prototype system to acquire, analyze, and store selected environmental and physiological data from a pilot during flight. The Aircrew Inflight Physiological Data Acquisition System (IFPDAS) II consists of an input multiplexer and analog-to-digital converter, a heart rate detector, a microprocessor, and a permanent memory device. The microprocessor's operating system monitors eight sensors, extracts desired information, and stores these reduced data in permanent memory. After the flight.
these data are transferred to a land-based computer which completes the
data processing and graphs the following environmental and physiological
information versus flight time: (1) cabin absolute pressure, (2) cabin
altitude, (3) Z-G's, (4) heart rate, (5) breathing rate, (6) minute
ventilation volume, (7) inspired oxygen quantity, and (8) expired
oxygen quantity.

The completed IFPDAS II prototype provides the desired information
well within the required accuracy. It provides the following parameter
ranges: (1) heart rate from 53 ± .1 to 225 ± 2.2 b/min, (2) breathing
rate from 4.7 ± .1 to 50 ± 1 b/min, (3) minute ventilation volume from
0 to 100 ± 2 l/min, (4) absolute pressure from 0 to 760 ± 2 mm Hg, and
(5) G's from -3 to +12 ± .1 G.