General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM
(PART III: UNATTENDED POWER PERFORMANCE MONITOR)

(NASA-CR-165540) THE DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM. PART 3: UNATTENDED POWER PERFORMANCE MONITOR
(Fairchild Weston Systems, Inc.) 63 p

by A. Hallett, D. Heidkamp, H. Neustadter, R. Olson

FAIRCHILD WESTON SYSTEMS INCORPORATED
DATA SYSTEMS DIVISION

prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract DEN 3-98
DOE/NASA WIND TURBINE
DATA ACQUISITION SYSTEM
(PART III: UNATTENDED POWER PERFORMANCE MONITOR)

by A. Hallett, D. Heidkamp, H. Neustadter, R. Olson

FAIRCHILD WESTON SYSTEMS INCORPORATED
DATA SYSTEMS DIVISION

prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Levis Research Center
Contract DEN 3-98
DOE/NASA WIND TURBINE DATA ACQUISITION SYSTEM
(PART III: UNATTENDED POWER PERFORMANCE MONITOR)

A. HALLETT, D. HEIDKAMP, H. NEUSTADTER, R. OLSON

FAIRCHILD WESTON SYSTEMS INCORPORATED
POST OFFICE BOX 3041
SARASOTA, FLORIDA 33578

PROJECT MANAGER: DR. HAROLD E. NEUSTADTER, NASA/LRC

THIS MANUAL PROVIDES SOFTWARE DOCUMENTATION, OPERATIONAL PROCEDURES, AND DIAGNOSTIC INSTRUCTIONS FOR DEVELOPMENT VERSION OF AN UNATTENDED WIND TURBINE PERFORMANCE MONITORING SYSTEM DESIGNED TO BE USED FOR OFF-LINE INTELLIGENT DATA ACQUISITION IN CONJUNCTION WITH THE NASA LEWIS RESEARCH CENTER CENTRAL HOST COMPUTER.

UNCLASSIFIED, LIMITED
<table>
<thead>
<tr>
<th>SECTION</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>GENERAL DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>PURPOSE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Scope</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>HARDWARE REQUIREMENTS</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Bubbl-Board</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Overall Block Diagram – Description</td>
<td>3</td>
</tr>
<tr>
<td>II</td>
<td>HOUSEKEEPING SOFTWARE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EVENT (System Initialization)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>RACKGD (Background Process)</td>
<td>5</td>
</tr>
<tr>
<td>III</td>
<td>REMOTE COMMUNICATIONS SOFTWARE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DMODEM (Driver, DRV11E/T103J Modem)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>DUMP (Dump Data Blocks)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>DPURGE (Dump and Purge Data Blocks)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>CHAPSWD (Change Password)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>XMITT (Line Transmitter)</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>RCV (Line Receiver)</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>DISPLAY (System Failure Messages)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>DMODEM Storage – Chart</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>DIAG (Software diagnostics)</td>
<td>16</td>
</tr>
<tr>
<td>IV</td>
<td>ADC CONTROLLER SOFTWARE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ADV11A (Driver, ADV11-A ADC)</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>LOGIC1 (One Minute Logic)</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>LOGIC6 (Six Minute Logic)</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>DTALOS (Data Loss Interrogation and Recovery)</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>DALOS (Store Data Loss Word)</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>PSTAT (Bump Day Counters and Initialization)</td>
<td>27</td>
</tr>
<tr>
<td>SECTION</td>
<td>TITLE</td>
<td>PAGE</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>TRBSEL (Select Data Block)</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>CLOCK (Update Data Base &quot;Time&quot;)</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>ECLOCK (Enable External Clock)</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>SCLOCK (Initialize External Clock)</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>CRESET (Reset External Clock)</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>BCDDEC (BCD to Decimal Converter)</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>BOPY (Disk I/O than DCER)</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>BASE (Memory/Disk Storage Allocation)</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>BB.SYS (I/O-Memory to Disk)</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>BBl.SYS (I/O-Disk to Memory)</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>RD RP (RCP Device Handlers)</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>RD.SYS (I/O-Line to Disk)</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>RP.SYS (I/O-Disk to Line)</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>ADV11A Storage - Chart</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>SUBPROCESS ENTRIES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTSB (No Operation)</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>AVGS (Accumulate and Average Data)</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>UNITS (Determine Turbine ON/OFF Status)</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>UNITM (Pack Turbine/Calibration ON/OFF Status)</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>AVG6 (Compute and Store 6 Minute Average)</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>UNIT6 (Store Turbine ON/OFF Status Word)</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>CALB6 (Store Calibration ON/OFF Status Word)</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>SYSTEM UTILITIES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRAP (Handler, System Failure)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>RCOPY (Peer File Transfers)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>RPIP (Remote File Transfers)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>DTOASC (Decimal to ASCII Conversions)</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>OCTA (Octal to ASCII Conversions)</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>DEFINITIONS</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>PRINTOUTS (24 Hour Data Dump)</td>
<td></td>
</tr>
</tbody>
</table>
SECTION I GENERAL DESCRIPTION
Section I
GENERAL DESCRIPTION

Introduction
This manual provides software documentation, operational procedures and diagnostic instructions for development version of an unattended wind turbine performance monitoring system designed to be used for off-line intelligent data acquisition in conjunction with the NASA Lewis Research Center central host computer.

References

Purpose
The purpose of this manual is to describe a software system which (when mated with appropriate hardware) collects power system performance data from a wind turbine station, processes the raw data and stores pertinent information in a data base which can be queried periodically via a commercial telephone link.

Scope
The Documentation contained in this manual describes the software system and related hardware components for a power system performance monitor centered around an existing LSI11/2 computer and R/T operating system. In its basic form this system monitors the parameters of as many as three wind turbines.

Hardware Requirements
Processor: The CPU is a DEC LSI11/2. It is configured with 32K of random access memory.

FWS 56A-7621 System Clock: This clock stores days, hours, minutes, and seconds in hardware registers. It also provides an external priority pulse to the CPU at a rate of 10SPS. This model is battery powered to be immune to station power fluctuations. It can be set to the proper date and time, both manually and from the software.
OM-11 BUBBL-BOARD

The QSB-11 BUBBL-BOARD is a complete non-volatile mass storage system implemented on a single dual height module suitable for operation in the LSI-11 microcomputer. The QSB-11 provides 128 K-bytes of mass storage in an all solid state package, completely eliminating all external chassis, power supplies, moving parts and mechanical storage media.

The QSB-11 is comprised of one bubble-memory device (with its attendant support circuitry), plus a complete controller which emulates the DEC RX01 single density floppy-disk system. The system also includes a built-in bootstrap.

The QSB-11 is compatible with the RT-11 operating system and is operated as the system mass-storage device using DEC's standard "DX" floppy-disk handler.

The QSB-11 plugs directly into the LSI-11 backplane bus. No other electrical connections are required. All required power (+5 and +12 volts) is derived from the bus.

The QSB-11 BUBBL-BOARD module is removable and transportable without loss of data contained within the magnetic bubble device. Therefore, data can be moved from one machine to another simply by pulling the BUBBL-BOARD out of one machine and installing it into the bus of another.

The QSB-11 system provides built-in self diagnostics which can be invoked with simple ODT commands. The self test mode is intended for use in acceptance testing, and as a field maintenance aid.

An activity indicator light (labeled RUN) on the QSB-11 indicates when a bubble device on the module is being accessed. Another indicator labeled ERROR illuminates if the controller ever detects an error in the bubble-stored data.

NOTE: DEC, LSI-11, AND RT-11 are registered trademarks of Digital Equipment Corporation, Maynard, Massachusetts, USA.

QSB-11 HARDWARE/SOFTWARE SPECIFICATIONS

Software Compatibility: Command compatible with DEC RXV11/RX01 floppy disk software.

Memory Capacity: 128 K-bytes (131,072 8-bit bytes).

Command Functions:
- Fill Buffer
- Empty Buffer
- Write Sector
- Read Sector
- Read Status
- Read Error Message
- Initialize
- Enable/Disable Interrupt
- Run Self Test

Sector Buffer Size: 128 bytes

Device Address Select: 177170

Interrupt Vector Select: 264

Bootstrap Address: 173000
Access Time (first byte): Less than 41 msec average, 82 msec maximum

Transfer Rate: In excess of 68,000 bits per second

Sector Transfer Time: Less than 18 milliseconds


Bubble Device Formatting: Automatic synchronization by controller at power-up.

Power Required (maximum): +5 volts DC at 1.4 amps. +12 volts DC at 0.05 amps, static; 0.2 amps when accessed.

Size: Standard dual-height (double-width) module.

Operating Temperature: 0 to 70 degrees Celsius, maximum, ambient.

Storage Temperature: -40 to 100° C, without loss of stored data.

Weight: 14 ounces.

Overall Block Diagram - Description

A block diagram representing the major components of the software system is given in Figure 1.4A.

The "event enable" block is the start of the program proper. It enables the required interrupts and sets system variables to their proper initialization values. The background logic serves two major functions. One is to provide a "wait loop" from which "foreground processing" (events) may interrupt the system. The other is to interrogate the modem logic to determine when a "ring" signal has been received. "Foreground" processing is derived from an external clock operating at 10 SPS. The ADV11A Handler is responsible for bringing in A to D data, accumulating same, averaging data every 6 minutes, storing averages in the data block, looking for power outage conditions and updating system pointers. The Modem Logic is accessed via telephone communication. It establishes caller protocol, priorities and protections, data block transmissions for certain callers, and permits diagnostic and program development measures to be taken. The Diagnostic Software facilitates hardware and software maintenance by detecting and isolating system malfunctions.
PERFORMANCE MONITOR

EVENT
EVENT
ENABLE
SOFTWARE

BACKGD
BACKGROUND
SOFTWARE

ADV11A
TELEMETRY
DATA "HANDLER"
SOFTWARE

DMODEM
MODEM
SOFTWARE

DATA BASE

DIAG
DIAGNOSTIC
SOFTWARE

RT11
SYSTEM
UTILITIES

FIGURE 1.4A MAJOR SOFTWARE SYSTEM COMPONENTS
SECTION II  HOUSEKEEPING SOFTWARE
PROGRAM: EVENT

PURPOSE: Initialize system.

CALLING SEQUENCE: Executed via system Bootstrap Loader.

Description:
Enables required interrupts, conditions A/D channels, establishes FWA of the Data Base and initializes system variables.

Initialization Procedure:
1. Set R4 to FWA of Data Base.
2. Clear Data Base, 8K locations.
3. Establish maximum number of channels to be monitored.
4. Set 4 locations in Purge equal to 1 corresponding to 4 days.
5. Specify storage used for counters.
6. Initialize day counters.
7. Clear storage used for counters.

Special Conditions:
Prior to entry, the global "LDFLG" must be cleared, otherwise, system variables will not be initialized.

Globals:
Data Base Variables, MINCNT, OFF, DLOSS, CALL, CAL2, CAL3, CAL4, CAL5, CAL6, CAL7, TB1180, TB124, XXCNT, XCNT, STARTD, STOPD, R2S, R0S, MSGFLG, PASS, ERR, CODES, CODEF, REQA, VBASE, LDFLG
EVENT

DISABLE SYSTEM INTERRUPTS

SAVE RT11 TRAP VECTOR ADDRESS

STORE DATA ACQUISITION TRAP VECTOR ADDRESSES

ZERO DATA BASE MEMORY AREA

LOAD SYSTEM POINTER AND DATA BLOCK

LDFLG = 20? = NO

INITIALIZE DATA BASE VARIABLES TO DAY 1

SCLOCK SET CLOCK TO D-1/H-0 M-0/S-0

A

CLEAR VARIABLES FOR DUMP/DPURGE

CLEAR VARIABLES FOR DMODEM

BACKGD
PROGRAM: BACKGD

PURPOSE: Background routine which has program control between external clock interrupts.

CALLING SEQUENCE: BR BACKGD

Real Time Entry:
While in the idle mode BACKGD monitors bit 15 of the DLVII-E RCSR and the diagnostic "REMFLOG" in anticipation of a Modem status change (ring) or a diagnostic monitor request. When a modem "ring" is detected BACKGD transfers program control to subroutine DMODEM. When the diagnostic REMFLOG is set BACKGD returns the RT11 operating system to the caller.

NOTE: DLVII-E Receiver Control Status Register is defined in DMODEM Documentation.

Special Conditions:
Upon every system "start up" and prior to entering BACKGD the system is checked for a power out. When a power out is not detected BACKGD is executed immediately. When a power out is detected a verification is made to determine whether power has been out for one or more minutes and/or one or more days.

When power is out for one or more days, the data block day pointers are reset equal to the current day. All data blocks associated with the power out day(s) are purged.

When power is out for one or more minutes the data loss word (2008) is stored into the data block of the current day starting at the current and subsequent locations of the pointer "n" until "n" is equal to the increment past midnight "TPM".

NOTE: Power out recovery as stated above, takes place on an as available basis and in between clock interrupts.

Special Requirements:
LSI11/2, DLVII-E Asynchronous Line Interface, Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements:
DMODEM, PFAIL, CLOCK, DTALOS

Globals:
RCSR, REMFLOG, IPM, N
PFAIL

NOP

TRBSSEL
GET DATA BLOCK CURRENT WORD LOCATION

STORE DATA LOSS WORD THROUGHOUT THIS DBLOCK

WRITE THIS DATA BLOCK

CLEAR N POINTER

BUMP DAY POINTER

TRBSSEL
GET NEW PURGE/DATA FWA

SET PURGE FLAG FOR REMAINDER OF POWER OUT DAYS

RESSET CLOCK AND CDAY TO 1, 2, 3, OR 4 FOR CURRENT DAY

DTALOS
GET NEW IPM

CLEAR ~ POINTER

READ THIS DATA BLOCK

RTS
SECTION III  REMOTE COMMUNICATIONS SOFTWARE
PROGRAM:  DMODEM

PURPOSE:  To maintain and support data communications between the DLV11-E Asynchronous Line Interface and the T103J Modem.

CALLING SEQUENCE:  DMODEM assumes the standard calling conventions which are established for MACRO compatibility.

Serial Communications Line Interface

GENERAL DESCRIPTION:

The DLV11-E asynchronous line interface module interfaces the LSI-11 bus to the Bell T103J Data Modem. This module receives serial data from peripheral devices, assembles it into parallel data and transfers it to the LSI-11 bus. In addition, the DLV11-E accepts data from the LSI-11 bus, converts it into serial data and transmits it to the peripheral devices.

DLV11-E OPERATIONS:

Baud Rate - 300

Data bit selection - 8 bits, EVEN Parity-generated, Received and Transmitted by hardware.

Sync - Two STOP bits; frames the data bits received and transmitted by hardware. "Halt on Framing Error-Disabled."

Receiver control/status register (RCSR) - Console module at memory address 177560.

Receiver Data Buffer (RBUF) - Console module at memory address 177562.

Transmit Control/Status (XCSR) - Console module at memory address 177564.

Transmit Data Buffer (XBUF) - Console module at memory address 177566.

Interrupts - Receiver and Transmitter are disabled by software. Console module address 300 and 304 are therefore not used by DMODEM software.

PERIPHERAL CHARACTERISTICS:

CPU Interface - RS-232-C

Alternate Mode - Places the keyboard or terminal in teletype mode, producing upper case characters.

Line - Full Duplex; input received from line can go to tape, memory or printer; output from any device other than line can only go to line. In addition, control characters for reading and writing can only come from the line or keyboard.

Format - USASCII code, all data internal to the terminal is handled with a 7 bit format and EVEN parity.
Real Time Entry:

DMODEM is entered via BACKGD (Background subroutine) upon acknowledgement of a modem status change. This status change is determined by monitoring bit 15 of the RCSR register. The DLVII-E hardware asserts bit 15 when a transition of any of the following lines is detected.

RING  
CLEAR TO SEND
SECONDARY RECEIVE DATA
DATA SET READY

RING (bit-15) - Reflects the state of the modem ring line, and when set indicates that a ring signal is being received from the modem.

CLEAR TO SEND (bit 14) - Reflects the state of the clear-to-send line from the modem, and when set indicates that the modem is ready to accept data from the interface for transmission.

CARRIER (bit 13) - Reflects the state of the modem carrier, and when set indicates that the carrier is up.

SECONDARY RECEIVED DATA (bit 11) - Provides a receive channel for supervisory data from the modem to the processor.

DATA SET READY (bit 10) - Reflects the state of the data set ready line from the modem, and when set indicates that the modem is powered-up and ready.
The RCSR Interrupt is enabled when bit 7 of the RCSR is set to '1'. However, this interrupt is not enabled for this system.

Real Time Entry Continued:
The current configuration of program DMODEM is set up such that all of the following conditions must be determined .TRUE. before data communication can begin.

1. Ring bit asserted
2. Modem power on
3. Modem ready to receive data from the interface for transmission (clear to send).

The .TRUE. state of the third condition is determined by the state of bit 15 RCSR. When set to '1' this bit reflects a .TRUE. condition. When set to a '0' this bit reflects a .NOT TRUE. condition. DMODEM monitors this bit periodically during data communication and transfers program control to the calling program when a .NOT TRUE. condition is acknowledged. In the event that all of the above conditions are .TRUE. all software requirements have been satisfied and data communication begins.
Telephone Caller Protocol:

The following information describes basic I/O between DMODEM and the telephone caller. In an effort to simplify the understanding of this basic I/O we shall first define the control characters that are "invisible" to the caller and required by both DMODEM and the Lerc IBM 3033/TSS System.

The control characters, DC1 and DC3, have been employed so that the TSS System can pace the input and output of DMODEM according to its TSS scheduling process.

**SYSTEM REQUIREMENTS:**

The DC1 (x-on) control character must be supplied by the caller's system to initiate character (message) transmissions from DMODEM to the caller.

The DC3 (x-off) control character must be supplied by the caller's system to stop character (message) transmissions from DMODEM to the caller. Characters that are transmitted to DMODEM in response to messages sent from DMODEM should always follow the DC3 control character.

Considering the above system requirements, we shall talk about "telephone caller protocol" as it is understood by the caller/operator.

Upon entry, DMODEM transfers a message to the telephone caller, then waits for a response. Message transmissions are in groups of 8-bit (EVEN Parity), ASCII characters. Character transmissions are complete when a carriage return is recognized by the caller, or when a DC3 character is recognized by DMODEM.

While waiting for the telephone caller's response, DMODEM monitors status lines of the DLV11-E. These status lines indicate whether the telephone caller hung up his/her telephone or whether a character has been received.

Each time a character is received from the caller, the Modem Logic determines whether an error was generated during transmission (Receive Error).

**ERROR PROCEDURES:**

Upon recognition of an error, the type of error is determined, then DMODEM waits for the completion of the telephone caller's character transfers. Thereafter, the incorrect characters and an error message are transmitted prior to the acknowledgement of the error. The telephone caller may make corrections, if applicable, then re-transmit.

**NOTE:** Incorrect password characters are not echoed.

**DATA COMMUNICATION PROCEDURES:**

When valid character strings have been received from the telephone caller, DMODEM translates and verifies the characters to determine whether or not the caller responded correctly. When it has been determined that the telephone caller's response is correct, DMODEM acts upon the response, transmits another message in a sequence, then re-initializes its protocol. When the last message in the sequence is transmitted and the telephone caller responds correctly, DMODEM transmits the first message in the sequence and communication continues.

Upon recognition of an incorrect response from the telephone caller, DMODEM performs the error procedures stated above.
Data communication and Modem Logic execution is discontinued when (1) the Modem power is turned off, or (2) the telephone caller hangs up his telephone.

Message I/O: Communication between DMODEM and the telephone caller is shown below. The DC1's, DC3's and CR's (carriage return) characters are displayed in an effort to give a complete understanding of the message I/O. However, DC1's and DC3's are supplied by the hardware and therefore "invisible" to the caller/operator.

Line in:  
Line out:  

Line in:  
Line out:  

Line in:  
Response:  
Line out:  

Line in:  
Response:  
Line out:  

Line in:  
Response:  
Line out:  

Note: DC1/DC3 characters are no longer required if the caller is Fairchild Weston.

Line in:  
Response:  
Line out:  

Line in:  
Response:  
Line out:  

Note: DMODEM transfers program control to subroutine DUMP, DPURGE, CHPSWD or DIAG to have the desired function performed. Subroutines DPURGE, CHPSWD, and DIAG require password identification before execution.

The following messages shall be transmitted to the caller upon recognition of an incorrect response or a Receiver Error.

Response Errors:

'Sorry Wrong Number'
'Sorry Wrong Function'
'Sorry Wrong Password'
RECEIVE ERRORS:

OVERRUN ERROR - CHARACTER LOST - The processor has failed to service the RX DONE flag within the time required to load another character into the Receiver Data Buffer, i.e., \((1/\text{baud rate}) \times (\text{bits per character})\) seconds. Hence, the previous character was overwritten (lost).

FRAMING ERROR - STOP BIT MISSING - The character received was not followed by a valid stop bit. In the Asynchronous mode, each received character must be preceded by a start bit and succeeded by a stop bit, which serves to synchronize the DRV11-E receive logic.

PARITY ERROR - REQUIRE EVEN PARITY - The parity of the received character does not agree with the parity programmed (EVEN).

Special Conditions:

Upon entry, the trap counters are interrogated to determine whether a system trap failure occurred since the last data transmission. When true, DMODEM sets the trap flag "TPERR" then prepares for telephone caller protocol. At such times as the caller properly identifies him/herself, the appropriate trap message is transmitted.

When not true, DMODEM continues the logic flow as stated above.

Special Requirements:

LSI11/2, DLV11-E Asynchronous Line Interface, Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements:

DUMP, DPURGE, DIAG, CHPSWD, XMITT, RCV

Globals:

FWPSWD, PLPSWD, MSGFLG, ERR, SAVEB, PASS, CODES, CODEF, MSG, MSG2, MSG3, ERRL, ERR3, ERR4, ERR5, ERR6, F,U,NC,T,I,O,NS, RCSR, RBUF, XCSR, XBUF, MODEMP, DRCSR.
PROGRAM: DUMP DPURGE

PURPOSE: Dump and/or Dump and Purge data blocks, transferring same via the DLV11-E Line Interface module and the T103J Modem.

CALLING SEQUENCE: JSR PC,DUMP
or
JSR PC,DPURGE

Data Block Transmissions:

Four data blocks per channel are contained in the Data Base. Each block is comprised of 120 words of storage, enough space to store processed information for up to 16 channels for four consecutive days. Each data block is maintained and updated by various subroutines and sub-processes described in this manual. Ref... Documentation for "BASE", this manual.

Real Time Entry:

DUMP: Transmits data blocks that are complete and that have not been purged.

DPURGE: Transmits data blocks that are complete and that have not been previously purged. Sets a purge flag for each data block transmitted. Password identification is required before execution.

Special Conditions:

Converts data to Hexadecimal then to ASCII. Transmits 4 MSB of byte followed by 4 LSB. Stops when all data for all channels of a data block have been transmitted. In addition, transmits a carriage return as "End of Block" indicator. Dumps all or selected data blocks. Selected data blocks may be dumped via the command "Du n" where "n" equals the data blocks associated Julian day number 0 to 365.

Special Requirements:

LS11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements:

XMITT

Globals:

XCN1, XXCN1, STARTD, STOPD, R2S, ROSS, ROSA, HEXASC, PURGE, M ADC1M1, MAXCH,
SRL, LF, CR, CODES, CODEF, M, XCSR, RCSR, XBUF
INITIALIZE
START/STOP
DAY DUMP
COUNTS

DISPLAY
START DATE
TIME
DEFINITIONS

DISPLAY
CURRENT
DAY
MESSAGE

START DAY=
STOP DAY?

YES
TERM

NO

HAS
CURRENT
DAY BEEN
PURGED?

NO

D

YES

TERM

XMITT
TRANSMIT
PURGE
MESSAGE

BUMP START
DAY AND
PURGE
INDICATORS

CLEAR ALL
PROGRAM
VARIABLES

CURRENT
DAY=DATA ACQ
DAY?

YES

DUMP

DISPLAY
START/STOP
DAY DUMP
COUNTS

TERM

TRANSMIT
DUMP DAY
MESSAGES

IS
CALLER
PLUMBROOK?

NO

FUNCTION
= DP?

NO

SET PURGE
FLAG FOR
CURRENT DAY

SET DUMP
FLAG
"ORCSR"

DUMP
THIS
DATA
BLOCK

NO

YES

E

F
PROGRAM: CHPSWD - Subroutine for DLV11-E/T103J Modem communications

PURPOSE: Change Your Password

CALLING ROUTINE: JSR PC, CHPSWD

I/O Communications:

CHPSWD performs the following functions:

1. Transmits the message "Enter new password."

2. Receives new password characters and checks for receiver errors. When Receiver errors are found, CHPSWD determines the type, transmits an error message then solicits the caller to "Enter New Password." When a new password is correctly received, CHPSWD changes the caller's password then,

3. Exits.

Special Conditions:

Password characters must not exceed 78 characters.

Special Requirements:

LSI11/2, DLV11-E Asynchronous Line Interface,
Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements: XMITT, RCV.

Globals: ERR, CR, LF, ERR4, ERR5, ERR6, FWPSWD, PLPSWD, SAVEB, CODES.
PROGRAM: XMITT  RCV

PURPOSE: XMITT: Consecutively, transmit 8-bits of a single word.
RCV: Consecutively, receive and transfer 8-bits of a single word to a buffer area supplied by the calling program. In addition, tests for receiver errors and sets an error flag when applicable.

CALLING SEQUENCE:

XMITT: MOV #TBUF,RO ; RO=FWA of user transmitter buffer.
       JSR PC,XMITT
EXIT1: BR OFF ; Return here if Modem power off
EXIT2: NOP and/or
       MOV #RBUF,R1 ; RL=FWA of used receiver buffer
       JSR PC,RCV
EXIT3: BR OFF ; Return here if Modem power off
EXIT4: NOP ; Return here if Modem power on

Prior to receiving or transmitting characters, RCV and XMITT monitor bit 14 of the RCSR. When set to '1', this bit indicates that the Clear to Send line of the DLV11-E is asserted and that the T103J Modem is powered on and ready. When set to 0, this bit indicates that the Clear to Send line has been disabled and that the T103J Modem is power off.

Dependent on the state of RCSR bit 14, XMITT performs the following functions:

NOTE: Upon entry, XMITT enables the DLV11-E RCSR Request to Send Line and prepares to accept transmission from the caller.

Bit 14 Asserted:

1. Test XCSR bit 8 (Transmitter Ready) to determine whether the DLV11-E Transmitter Buffer can accept another character.
2. When not true, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.
3. When true, test receiver buffer (RBUF) for a DC1 (x-on) character.
4. When not true, continues monitoring RCSR bit 14/8 and testing RBUF for DC1 character.
5. When true, transfers 8 bits of the current buffer location. Bumps R1 to point to the next 8 bit character. Checks for a line feed character and transfers program control to EXIT2 when encountered. Otherwise, continues monitoring RCSR bit 14 and bit 8, repeating the functions as stated above.

Bit 14 Disabled:

1. Returns program control to EXIT1 of the calling program.

Dependent on the state of RCSR bit 14, RCV performs the following functions:

NOTE: Upon entry, RCV disables the DLV11-E RCSR Request to Send Line and prepares to accept transmission from the caller.
Bit 14 Asserted:

1. Tests RCSR bit 8 (Receiver Done) to determine whether an entire character has been received.

2. When not true, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.

3. When true, test receiver buffer (RBUF) for a DC3 (x-off) character.

4. When not true, continues monitoring bits 14 and 8 and testing for a DC3 character.

5. When true, transfers the received character to the user receiver buffer. Checks for receiver errors. Sets an error flag when necessary. Makes provisions for rubout characters. Returns program control to EXIT4 upon receipt of a carriage return character. Otherwise, continues monitoring RCSR bit 14 and bit 8, performing the functions as stated above.

Bit 14 Disabled:

1. Returns program control to EXIT3 of the calling program.

User Receiver Buffer Upon Exit:

Character data is stored sequentially into the current and subsequent byte locations of the receiver buffer followed by a carriage return and a line feed.

Error Conditions:

The global ERR is set to a #4, 5, or 6 when a receiver error is encountered. These error numbers are defined as follows:

- ERR 4 = Overrun Error, character lost
- ERR 5 = Framing Error, stop bit missing
- ERR 6 = Parity Error, EVEN parity required

ERROR RECOVERY: At the discretion of the calling program.

Special Conditions:

Under normal conditions (Modem power on), the user supplied transmitter buffer (TBUF) must contain a line feed character in the last byte location to terminate transmission. Receiver operations are terminated only after a carriage return character is received. Also, the user should allocate an additional byte location at the end of RBUF to allow for a line feed character, supplied by RCV.

Special Requirements:

LSI11/2, DLV11-E Asynchronous Line Interface, Bell 103 Modem or equivalent, RS232 CPU Interface.

External Requirements: None

Globals: ERR, RCSR, XCSR, RBUF, XBUF, LF, CR, SR4
MODEM ON?

XMITTER READY?

TRANSMIT A CHARACTER-BYTE

NEXT CHARACTER = LINE FEED?

IS CALLER FAIRCHILD?

REMOTE INPUT

DC3?

MODEM ON?

YES

YES

YES

YES

BUMP STACK POINTER BY 2 NORMAL EXIT

RESET DC1/DC3 FLAGS ABNORMAL EXIT

RTS
PROGRAM: DISPLY

PURPOSE: Determine the type of system failure and transmit an appropriate message.

CALLING SEQUENCE: JSR PC,DISPLY

Upon entry, determines type of system failure then transmits an appropriate message to the telephone caller. This message will print prior to a channel dump, however, the dump will not be inhibited. The message(s) printed should be used when making decisions to troubleshoot the system via program DIAG.

The trap flag "TPERR" is reinitialized by DIAG.

Message Format:

TRAP VECTOR # - DESCRIPTION

Special Requirements: None

External Requirements: None

Globals: None

MODEM STORAGE AREA

<table>
<thead>
<tr>
<th>MSG</th>
<th>/Who is calling please?/ CR LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG2</td>
<td>/Please give function code/ CR LF</td>
</tr>
<tr>
<td>MSG3</td>
<td>/Please give password/ CR LF</td>
</tr>
<tr>
<td>MSG4</td>
<td>/Enter new password/ CR LF</td>
</tr>
<tr>
<td>ERR1</td>
<td>/Wrong number/ CR LF</td>
</tr>
<tr>
<td>ERR2</td>
<td>/Wrong function/ CR LF</td>
</tr>
<tr>
<td>ERR3</td>
<td>/Wrong password/ CR LF</td>
</tr>
<tr>
<td>ERR4</td>
<td>/Overrun Error Character Lost/ CR LF</td>
</tr>
<tr>
<td>ERR5</td>
<td>/Framing error stop bit missing/ CR LF</td>
</tr>
<tr>
<td>ERR6</td>
<td>/Parity error require even parity/ CR LF</td>
</tr>
<tr>
<td>CODES</td>
<td>Station code characters</td>
</tr>
<tr>
<td>CODEF</td>
<td>Function code characters</td>
</tr>
<tr>
<td>MSGFLG</td>
<td>Message flag</td>
</tr>
<tr>
<td>ERR</td>
<td>Error number</td>
</tr>
<tr>
<td>PASS</td>
<td>Password error flag</td>
</tr>
<tr>
<td>SAVEB</td>
<td>FWA of password characters</td>
</tr>
<tr>
<td>FWPSWD</td>
<td>Fairchild Weston Password Storage</td>
</tr>
<tr>
<td>PLPSWD</td>
<td>Plumbrook Password Storage</td>
</tr>
<tr>
<td>PRGO</td>
<td>/Current Day= / CR LF</td>
</tr>
<tr>
<td>PRGL</td>
<td>/Buffer has been purged/ CR LF</td>
</tr>
<tr>
<td>DMPDAY</td>
<td>/Buffer to follow/ CR LF</td>
</tr>
<tr>
<td>CHN</td>
<td>/Channel / CR LF</td>
</tr>
<tr>
<td>F</td>
<td>/DU/ CR LF</td>
</tr>
<tr>
<td>U</td>
<td>/DUMP/ CR LF</td>
</tr>
<tr>
<td>NC</td>
<td>/DUMP AND PURGE CR LF</td>
</tr>
<tr>
<td>T</td>
<td>/DUMP PURGE/ CR LF</td>
</tr>
<tr>
<td>I</td>
<td>/CH/ CR LF</td>
</tr>
<tr>
<td>O</td>
<td>/DI/ CR LF</td>
</tr>
<tr>
<td>NS</td>
<td>/DB/ CR LF</td>
</tr>
</tbody>
</table>
PROGRAMS: DIAG CHETR/2 RONV PROBEM COPY DELETE DUMP - Diagnostic's

PURPOSE: To facilitate hardware and software maintenance via a remote communication link.

OPERATOR INITIATION: ;REF...DIADE1 Documentation

CALLING SEQUENCE: JSR PC,DIAG

OVERVIEW

The diagnostic software is a segment of the ADV1A Data Acquisition software and facilitates hardware and software maintenance by detecting and isolating system malfunctions. In addition, the diagnostic software affords the system program manager with the capability of locating and correcting any errors in programs written for or supplied by the user. Diagnostic program execution for all routines except CHETR/2, Boot and Monitor takes place on an as available basis in between interrupts. Therefore, data acquisitioning is not inhibited during program/operator I/O.

DIAG

DIAG is the program shell of the diagnostic operation, supervisor of all diagnostic subroutine execution and the intermediary of operator/program communication. In addition, DIAG performs housekeeping functions which:

1. Optionally allow data acquisition start date and start time inputs to be made permanent on disk for program retrieval and display,
2. Process and print statistical information pertinent to the data acquisition of up to 3 wind turbines,
3. Redirect diagnostic program control to the RT11 operating system and
4. Permit system bootstrapping.

PROGRAM/OPERATOR COMMUNICATION

PROGRAM: SELECT A DIAGNOSTIC

OPERATOR: Input one of the diagnostic names in the menu selection or type "MENU" if uncertain.

MENU SELECTION

<table>
<thead>
<tr>
<th>Diagnostic Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prints</td>
<td>Print System Statistics</td>
</tr>
<tr>
<td>VCLOCK</td>
<td>Verify clock time increments and clock loading.</td>
</tr>
<tr>
<td>RONV</td>
<td>Display selected channels data.</td>
</tr>
<tr>
<td>CLEARP</td>
<td>Clear power out and system status flags.</td>
</tr>
<tr>
<td>PROBEM</td>
<td>Verify and/or correct system failures, or Load and/or execute PDPL machine language instructions.</td>
</tr>
<tr>
<td>BOOT</td>
<td>Boot in new NASA Software System.</td>
</tr>
<tr>
<td>MONITOR</td>
<td>Return to RT11 System Level.</td>
</tr>
<tr>
<td>DATE - MM-DD-YY</td>
<td>Define data acquisition start date.</td>
</tr>
<tr>
<td>TIME - HH:MM:SS</td>
<td>Define data acquisition start time.</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy files on disk.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete files from system disk.</td>
</tr>
<tr>
<td>DUMP</td>
<td>Dump file blocks from disk to line.</td>
</tr>
<tr>
<td>MENU</td>
<td>Display this menu.</td>
</tr>
<tr>
<td>EXIT</td>
<td>Leave diagnostic mode.</td>
</tr>
</tbody>
</table>
PRINTS

PRINTS prints the following information to the remote terminal.

(1) Total amount of elapsed clock time since the start of data acquisition.
(2) Total amount of elapsed clock time during power outs.
(3) The number of the current data block being updated.
(4) The number(s) of the data blocks that have been purged.
(5) The current on/off status of up to 3 turbines.
(6) Messages identifying system failures.

CLEARP

CLEARP resets power out and system status flags to inhibit power fail and system failure messages that are printed to the remote terminal.

BOOT

BOOT reloads the RT11 operating system and the ADV11-A Data Acquisition Software.

MONITOR

MONITOR terminates Data Acquisitioning and transfers program control to the RT11 operating system at system level.

DATE/TIME

DATE and TIME inputs are stored on disk in the event of power outages. Once defined, the time and date should remain unchanged until data acquisition is reinitialized to Julian Day 1.

COPY

COPY performs basic disk file transfer operations for system software maintenance. The command line format is as follows:

*DISK:FILEOUT.EXT=DISK:FILEIN.EXT <CR>

DELETE

DELETE is a software maintenance process which deletes files from the system disk that are outdated or obsolete. The command line format is as follows:

*FILENAME.EXT <M

DUMP

DUMP utilizes the remote communications driver BBI.SYS and dumps file blocks from the system disk to modem line. The command line format is as follows:

*MB:FILEOUT.EXT=DISK:FILEIN.EXT <CR>

Files are transmitted to line in 512 decimal ASCII word blocks.

MENU

Prints the menu selection to the remote terminal.

EXIT

Terminates diagnostic operations and returns program control to the calling program.
DIAGNOSTIC SUBROUTINES

CHETR1/2

CHETR1 determines whether or not the interrupt service of the 50A-7621 system clock maintains a consistency without error at one second intervals over an infinite period of time.

CHETR2 determines whether or not the 50A-7621 system clock programs correctly when loaded under software control.

CHETR1/2 are executed in sequence when the clock verification function "VCLOCK" is requested by the operator. Both routines handle and disclose to the operator any errors found applicable to their operation.

Upon acknowledgement of errors CHETR1/2 disclose the following information.

Output message:

Output message:

Clock SYL1 = ...................... ;minutes/seconds
Computed SYL1 = ...................... ;minutes/seconds
Clock SYL2 = ...................... ;days/hours
Computed SYL2 = ...................... ;days/hours

Clock syllable 1/2 represents the current clock time taken from the software clock when the error occurred. The computed syllables 1/2 represent the time computed by software based on an anticipated one second interval generated by the clocks' P.P.S. interrupt. When the clock syllables and computed syllables agree the 50A-7621 system clock is said to be incrementing correctly and providing a consistent one second interrupt pulse. When the syllables do not agree the messages shown above are disclosed to reflect the acknowledgement of an inconsistency in clock increments and interrupt pulses.

Output message:

Output message:

Programmed Table

* Seconds = 14xx minutes = 24xx hours = 44xx *
* Days = 104xx Hun's Days = 204xx *

CLOCK ERROR!!

Load Data = ............ ;minutes/seconds
SYL1 = ............ ;minutes/seconds
SYL2 = ............ ;days/hours

"LOAD DATA" represents the data written to the clock upon detection of an error. Syllable 1/2 represent the data read from the clock upon detection of an error. These error messages are disclosed when the data written to the clock is not equal to the data read.

The program table is printed to aid in diagnosing the clock error(s) and should be referenced when defining the octal value in the "octal data" column of the message.
CHETRI/2 PROGRAM TERMINATION

A termination sequence begins when the caller hangs up his/her phone. At the completion of the sequence the RTll operating system is rebooted and the data acquisition initialized to start acquiring data for Day 1. The start date and start time must be redefined by the operator.

RONV - Display ADVII-A Data

The RONV diagnostic reads selected channels data from the ADVII-A analog to digital converter and displays same on a remote terminal device. Upon entry, the following program/operator interchange is initiated.

PROGRAM: Enter channel number.
OPERATOR: Input the desired channel number in decimal, 16 channels max.
PROGRAM: How many samples this channel?
OPERATOR: Input any decimal number from 1 to 32767.

After the channel number and number of samples have been input RONV responds by printing the total number of samples requested for the channel. Each sample is printed in octal ASCII and represent the channels most recent data conversion. The samples printed are "raw data" with 12 significant bits.

At the completion of each channel data print segment the operator is permitted to continue by inputting another channel and sample count. The operator may terminate input by entering "EXIT" when the channel number is requested by the program.

PROBEM - Memory Probe

PROBEM performs functions similar to the RTII Online Diagnostic Routine (ODR). Basically, PROBEM makes available a method by which the system manager can manipulate memory addresses and locations thereof for program testing and system analysis.

PROGRAM/OPERATOR communication begins when an "at" (@) character prompt is printed to the remote terminal. The operator need only enter the address of interest followed by a slash character (/) and the program displays the address contents. This address can be altered by entering an octal value on the same line followed by a carriage return or line feed. Any READ/WRITE location of memory may be displayed and/or changed. Any number of memory locations containing valid program instructions can be executed by entering a first word address followed by a "G" character. A "P" character may be entered to terminate the PROBEM operation.

The following examples illustrate program/operator communication.

FORMAT
ADDRESS CONTENTS
@xxxxxx / xxxxxx

(1) Display the contents of memory location 1000g.

@1000 / 000416
(2) Change the contents of memory location 10008.

@1000 / 000416 207<CR> ;before
@1000 / 000207 <CR> ;after
@ ;prompt after carriage return

(3) Display memory location 10008 through 10108.

@1000 / 000207 <LF>
@1002 / 40260 <LF>
@1004 / 000000 <LF>
@1006 / 000000 <LF>
@1010 / 000000 <CR>
@

(4) Change memory locations 10008 through 10028.

@1000 / 000207 240 <LF> ;before
@1002 / 40260 207 <CR> ;before
@1000 / 000240 <LF> ;after
@1000 / 000207 <CR> ;after

(5) Execute the program starting at memory address 10008.

@1000G

(6) Boot the RT11 operating system and restart the data acquisitioning.

@173000G

(7) Display the contents of the general purpose registers and set R2 equal to 108.

@R0 / 000121 <LF>
@R1 / 000000 <LF>
@R2 / 000000 10 <LF>
@R3 / 000002 <LF>
@R4 / 40260 <LF>
@R5 / 000001 <LF>
@R6 / 000776 <CR>
@

(8) Terminate the PROBEM operation and return to the caller.

@P

Select a diagnostic output by program DIAG upon return.

NOTE: All memory and register changes are made permanent upon termination of the PROBEM operation.
SPECIAL CONDITIONS

The operator is required to enter a special access code when requesting execution of the following routines:

(1) CLEARP  
(2) PROBEM  
(3) BOOT  
(4) MONITOR  
(5) COPY  
(6) DELETE  
(7) DUMP

SPECIAL REQUIREMENTS

Data terminal must be set for 8 bits even parity. ADV11A Data Acquisition Software.

LIMITATIONS: None
PRINTS

GET CURRENT TIME OF DATA
DECREMENT DAYS

DTOASC
STORE IN MESSAGE 1

POWER OUT?

YES

GET CURRENT DAY NUMBER

DTOASC
STORE IN MESSAGE 1A

COMPRESS
ELAPSED TIME

DTOASC
STORE IN MESSAGE 2

GET PURGED DAY COUNTS

DTOASC
STORE IN MESSAGE 3

B

TRANSMIT MESSAGE 1

POWER OUT?

YES

TRANSMIT MESSAGE 1A

TRANSMIT MESSAGE 2

TRANSMIT MESSAGE 3

C
C

TURBINE 1 ON? NO -> TRANSMIT MESSAGE 4A

YES -> TRANSMIT MESSAGE 4

TURBINE 2 ON? NO -> TRANSMIT MESSAGE 5A

YES -> TRANSMIT MESSAGE 5

TURBINE 3 ON? NO -> TRANSMIT MESSAGE 6A

YES -> TRANSMIT MESSAGE 6

SYSTEM FAILURES DETACHED?

DISPLY

RTS
DATE

ECHO LINE
JUST INPUT

STORE DATA IN SYSTEM POINTER BLOCK

RTS

TIME

ECHO LINE
JUST INPUT

STORE TIME IN SYSTEM POINTER BLOCK

RTS

MENU

TRANSMIT MENU

RTS

EXIT

BACK TO DMODEM
CLEAR P

CLEAR ALL GLOBALED STATUS FLAGS

RTS

BOOT

DISABLE ALL INTERRUPTS

JMP @173000

EXIT

COPY

DELETE

DUMP

ILLEGAL FILE SPEC'S?

YES

RTS

NO

INPUT = "EX"?

YES

RTS

NO

DUMP REQUEST?

YES

BUMP GLOBAL DUMP FLAG

NO

F

G

DELETE REQUEST?

NO

BOPY

YES

DCOPY LOCATED IN MAIN LINE

RTS

F

G

RTS
CHETR2

DISABLE CLOCK INTERRUPT AND SAVE REGISTERS

TRANSMIT WARNING MESSAGES

RESET CLOCK TO ALL ZEROS

SET DATA = 1 SECOND

LOAD CLOCK WITH DATA

READ SYL1 AND SYL2

MODEM ON?

DATA READ .EQ. DATA WRITER?

ALL BIT POSITIONS TESTED THIS SYLLABLE?

SHIFT DATA ONE BIT POSITION

SET DATA = 1 HOUR

FIRST TIME IN?

DISPLAY DATA READ AND DATA WRITTEN

RESTORE REGISTERS

RTS
SECTION IV ADC CONTROLLER SOFTWARE
PROGRAM: ADV11A  LOGIC1  LOGIC6

PURPOSE: Controls and maintains the sampling selection of one of sixteen channels of the ADV11-A analog to digital converter. Isolates data samples, selecting only those of interest and transfers same to various subprocess.

CALLING SEQUENCE: Entered 10 times per second via the system interrupt address 3008.

Real Time Entry:

Upon acknowledgement of the external clock event, 10 times per second, all registers are saved, then set up to contain the following:

- R0 = Converted data
- R1 = FWA of next subprocess, turbine data
- R2 = FWA of channel table
- CHANNEL = Data Base Storage for Current Channel Number.

DATA REDUCTION AND ACQUISITION:

Figure 4.3A outlines sample tables consisting of subprocess addresses. The table in 4.3A allocates sixteen 16-bit memory locations, one location per address. Each address is referenced after the current channel data is (1) converted from serial to parallel, (2) reduced from 12 to 7 bits and (3) stored in R0.

DATA PROCESSING:

With each table reference, ADV11A transfers program control to a subprocess whose address is in an active location of the table. Location #1 in the table contains the FWA of a subprocess for channel #1 data processing, location 2, channel 2 data processing, and so on.

Upon return from each subprocess the current channel number and table pointer are incremented to point to the next channel number and associated subprocess. The channel number is compared with a maximum channel count (MAXCH) to determine whether all channels have been processed. In the event that all channels have not been processed, ADV11A repeats data reduction as stated above. If however, all channels have been processed ADV11A increments a sample counter (NSAMP) then reinitializes the channel number (CHANNEL). Thereafter, program control is transferred to LOGIC1.

NOTE: "NSAMP" is used by the subprocess AVGS' and represents the number of samples processed per channel over a period of six minutes.
<table>
<thead>
<tr>
<th>Corresponding Channel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FWA OF AVGS 1</td>
<td>1</td>
</tr>
<tr>
<td>FWA OF AVGS 2</td>
<td>2</td>
</tr>
<tr>
<td>FWA OF AVGS 3</td>
<td>3</td>
</tr>
<tr>
<td>FWA OF AVGS 4</td>
<td>4</td>
</tr>
<tr>
<td>FWA OF AVGS 5</td>
<td>5</td>
</tr>
<tr>
<td>FWA OF AVGS 6</td>
<td>6</td>
</tr>
<tr>
<td>FWA OF AVGS 7</td>
<td>7</td>
</tr>
<tr>
<td>FWA OF AVGS 8</td>
<td>8</td>
</tr>
<tr>
<td>FWA OF AVGS 9</td>
<td>9</td>
</tr>
<tr>
<td>FWA OF UNITS 10</td>
<td>10</td>
</tr>
<tr>
<td>FWA OF AVGS 11</td>
<td>11</td>
</tr>
<tr>
<td>FWA OF AVGS 12</td>
<td>12</td>
</tr>
<tr>
<td>FWA OF AVGS 13</td>
<td>13</td>
</tr>
<tr>
<td>FWA OF AVGS 14</td>
<td>14</td>
</tr>
<tr>
<td>FWA OF AVGS 15</td>
<td>15</td>
</tr>
<tr>
<td>FWA OF UNITS 16</td>
<td>16</td>
</tr>
</tbody>
</table>

FIGURE 4.3A
Subprocess Table
For Sample Entries
DATA ACQUISITION

ADV11A

SAVE REGISTERS

PICKUP FWA OF CHANNEL BUFFER

IS CHANNEL = 1?

GET FWA OF CURRENT CHANNEL IN CHANNEL BUFFER

PICKUP FWA OF SAMPLE TABLE

CLEAR STATUS REGISTER CSR1

SELECT A/D CHANNEL

INITIATE CONVERSATION

DATA PROCESSING

CONVERSION COMPLETE

YES

PICKUP DATA REDUCE TO 7 BITS STORE IN RO

BRANCH TO A SUBPROCESS FOR CURRENT DATA CHANNEL

BUMP CHANNEL NUMBER

ALL CHANNELS PROCESSED?

INCREMENTSAMPLE COUNTER

RESET CHANNEL NUMBER TO 0

PICKUP FWA OF 1 MINUTE TABLE FOR THIS TURBINE

LOGIC 1
TITLE: LOGIC1

PURPOSE: Transfers program control to predefined subprocesses whose objective is to process those channels data collected and saved during ADV11A data processing, performing various software functions at one minute intervals.

CALLING SEQUENCE: MOV #TMIN, RL ; RL FWA of one minute subprocess entries
BR LOGIC1

Real Time:

Upon entry the global NSAMP is compared with the minute count variable MINS to determine whether one minute has elapsed since the last entry. If one minute has not elapsed LOGIC1 restores those registers saved by ADV11A then issues a return from interrupt (RTI). If one minute has elapsed, NSAMP is reinitialized and program control is transferred to each subprocess in the TMIN table, Figure 4.8A.

Upon completion of the subprocess transfers, the channel count (CHANNEL) is reinitialized and a check is made to determine whether six minutes has elapsed. In six minute intervals, LOGIC1 reinitializes the global MINCNT (one minute counter) then transfers program control to subroutine LOGIC6, otherwise LOGIC1 restores ADV11A registers then issues an RTI.

TITLE: LOGIC6

PURPOSE: Transfers program control to predefined subprocesses whose objective is to process and/or store those channels data collected, saved and preprocessed by ADV11A and LOGIC1. The memory image of the data block and the system pointer block is written to disk.

CALLING SEQUENCE: MOV #TISIX, RL ; FWA of subprocess six minute entries
BR LOGIC6

Real Time:

Upon entry, LOGIC6 transfers program control to the subprocesses whose FWA's are located in the first and subsequent active locations of table TISIX (Six minute table, Figure 4.8B).

After completing the subprocess transfers, LOGIC6 reinitializes the channel counter (CHANNEL), writes the data block to disk, then transfers program control to subroutine PSTAT (performs day rollover, housekeeping functions). Upon return from PSTAT, LOGIC6 prepares for its exit, restoring all registers saved by ADV11A, then reinitializing the data loss flag. Thereafter, an RTI is issued to return from interrupt.

NOTE: When telephone communications are in progress the data block is written to disk only if day rollover occurs at the current 6 minute entry into LOGIC6. Otherwise, the memory image of the data block is updated and gets written to disk when telephone communications are terminated.
### Special Conditions:

None

### Special Requirements:

Digital (DEC) ADV11-A A/D Converter.

### External Requirements:

DATAOS, PSTAT

### Globals:

CALB, CALON, CALERR, CHANL, CHANNEL, MAXCH, MINS, MINCNT, TIMIN NSAMP, SAMP, SIX, TISIX AND DLOSS.
LOGIC 1

HAS 1 MINUTE ELAPSED?

NO

RETURN

YES

RESET SAMPLE COUNTER

BRANCH TO THE SUBPROCESS FOR THE CURRENT CHANNEL

INCREMENT CHANNEL COUNT

ALL CHANNELS PROCESSED?

NO

A

YES

RESET CHANNEL COUNT

INCREMENT MINUTE COUNT

B

B

HAS 6 MINUTES ELAPSED?

NO

RETURN

YES

RESET MINUTE COUNT

PICKUP FWA OF 6 MINUTE TABLE FOR THIS TURBINE

LOGIC 6
LOGIC 6

DTALOS

GET IPM

A

BRANCH TO A SUBPROCESS

INCREMENT CHANNEL COUNT

ALL CHANNELS PROCESSED?

COMPUTE FWA OF NEXT CHANNEL

LAST BUFFER BEFORE DAY ROLLOVER?

FUNCTION = DUMP?

WRITE POINTER AND DATA BLOCK TO DISK

RTN

PSTAT

BUMP N/M POINTER

RESET CHANNEL/DLOSS COUNTS

RESTORE REGISTERS

RTN
PROGRAM: DTALOS

PURPOSE: Compute increment past midnight and look for data loss.

CALLING SEQUENCE: JSR PC,DTALOS

Real Time Entry:

DTALOS is executed every 6 minutes to compute the increment past midnight (IPM). The computed IPM is compared with the data block pointer "n" to determine whether a data loss occurred since the last entry. Upon recognition of a data loss the data loss flag "DLOSS" is set equal to a number greater than zero. This flag is monitored by processes at the six minute entries.

IPM - Computes as follows:

IPM = (Hours x 10 + (minutes/6))
Hours x 10 = The total number of increments per hour.
Minutes/6 = The total number of increments per six minute interval within a period of one hour.

Special Conditions: None

External Requirements:

CLOCK, CRESET, TRBSEL, DTALOS

Globals:

CDAY, IDAY, JDAY, IFLAG, JFLAG, SAVEM, M, N, IPM, MAXCH, CHANEL, DLOSS, TIME

PROGRAM: DALOS

PURPOSE: Stores Data Loss word/s.

CALLING SEQUENCE: SSR PC,DALOS

Real Time Entry: R5 = FWA of Day-M(n)

Upon entry, DALOS stores the data loss word (2008) into the data base starting at the current and subsequent position of the pointer "n" until "n" is equal to the increment past midnight (IPM).

Special Conditions:

DALOS should be entered only when data losses are detected. The data loss flag "DLOSS" should be tested for > zero before entry.

Special Requirements: None

External Requirements: None

Globals:

IPM, N, DLOSS
PROGRAM:  PSTAT

PURPOSE:  Adjusts program variables

CALLING SEQUENCE:  JSR PC,PSTAT

Real Time Entry:

Updates, if necessary, the current day counters, N pointer, external clock and FWA of the next day's data block. Also makes provision for leap year clock rollover.

Special Conditions:  None

Special Requirements:  None

External Requirements:

SCLACK, TRBSEL

Globals

M, N, IDAY, JDAY, PURGE, IPM

PROGRAM:  TRBSEL

PURPOSE:  Establishes the FWA of the Data Block and purge flag for the current day.

CALLING SEQUENCE:  JSR PC,TRBSEL

Real Time Entry:

Upon Entry, TRBSEL verifies the globaled "M" count to determine the "current day." Thereafter, general purpose register R5 is set equal to the FWA of the Data Block for that day. R5 is then adjusted to point to the current Data Storage location for A/D channel #1. The purge flag pointer is positioned at the current day's purge flag location. In addition, the GLOBL "DBLOCK" is setup to contain the FWA of the command string for data block disk write functions (performed by BOPY).

Special Conditions:  None

Special Requirements:  None

External Requirements:  None

Globals:

PURGE, M, N, ADC1M1

27
PSTAT

BUMP DAY POINTER

END OF BLOCK?

YES

BUMP DAY POINTER

LEAP YEAR ROLLOVER?

NO

YES

SET IDHY = 1

RTS

DAY POINTER -ET- 4?

NO

YES

SET DAY POINTER = 1

TRBSEL

B

DAY POINTERS EQ 1?

NO

RTS

YES

SET CLOCK DAY = 1
BSEL
MAND
BUFFER DONE
WA
GET DATA
SET R5 = FWA
OF DAY n(n)
GET PURGE FLAG BUFFER FW
CLEAR PURGE FLAG FOR CURRENT DAY

DAY = 1
? YES
NO
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE

DAY = 2
? NO
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE

DAY = 3
? NO
STORE FWA OF WRITE C-STRING FOR THIS DAY
DONE

DAY = 4
? NO
STORE FWA OF WRITE C-STRING FOR DAY 4
DONE

DONE
PROGRAM: CICLOK

PURPOSE: Updates the Data Base "TIME" locations, once every 6 minutes.

CALLING SEQUENCE: JSR PC, CLOCK

Real Time Entry:

Upon entry, CICLOK sends a freeze command (400)g to the DRV11 output register. This freeze command switches the output data mux to syllable 1, minutes and seconds, and assures that the output data remains stable for at least 50 msec. After the freeze command is sent, syllable 1 is read from the DRV11 data register 167774 and saved. Thereafter, the DRV11 automatically switches the clock output data to syllable 2, hours and days, which is also read from register 167774 and saved. Syllable 1 and 2 are converted from BCD to Decimal then the Data Base Locations reserved for days, hours, minutes, and seconds are updated accordingly. In addition, the global CRAY is set equal to the current clock day. Program control is returned to the caller after the above procedures are complete.

CICLOK DESCRIPTION:

The external clock is fully programmable and provides BCD days, hours, minutes and seconds. The clock provides an external priority pulse to the CPU at the rate of 10SPS. It is battery powered to be immune from station power fluctuations.

Specifications:

1. Power requirements are .060 amps at 5v.DC, power provided by an internal power supply and battery backup.
2. Interfaces to an LSI/11 CPU via a DEC DRV11 parallel line unit.
3. Provides julian date or elapsed time to the DRV11 input port in two syllables; Syllable 1 is minutes and seconds, Syllable 2 is days and hours.
4. Time may be set to any valid time via the output port of the DRV11.
5. Provides two interrupt rates:

   Enable Interrupt
   (1) REQ A at 10 P.P.S. ; MOV #177770, RO
       ; MOVB #100,(RO)
   (2) REQ B at 1 P.P.S. ; MOV #177770, RO
       ; MOVB #10, (RO)

FORMAT OF TIME DATA READ FROM THE EXTERNAL CLOCK:

<table>
<thead>
<tr>
<th>Syllable 2</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOURS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syllable 1</th>
<th>X</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINUTES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECONDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Remote Programming:

<table>
<thead>
<tr>
<th>Command</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAR 37000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREEZE 400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET SECONDS 14XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET MINUTES 24XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET HOURS 44XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET DAYS 104XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET HUNS DAYS 2040X</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Programming Example:

```
SETDAY: MOV #177772, R0 ; R0 = FWA of DRV11 output register.
       MOV #37000, (R0) ; Clear days, hours, minutes and
       MOV #10401, (R0) ; Set clock Day = 1, Hours, minutes
                        and seconds remain unchanged.
```

Special Conditions:

In addition to updating the database "TIME" locations, CLOCK also processes
and stores the current time into the database "PTIME" locations. These time
locations are used when computing the total amount of elapsed time during
power out.

Special Requirements:

DRV11 Parallel Line Unit, FWS 56A-7621 Battery Powered External Clock.

External Requirements:

BCDDEC

Globals:

SRO, TIME, DRIBUF, DCAY, DRCSR, DROBUF, IDAY

PROGRAM: ECLOCK SCLOCK CRESET

PURPOSE: Enable, Set, and Reset the external clock.

CALLING

SEQUENCE: JSR PC,ECLOCK
           JSR PC,SCLOCK
           JSR PC,CRESET

Real Time Unit:

ECLOCK: Enables the external clock interrupt by placing a '1' bit in the
7th bit of the clock control status register.

SCLOCK: Sets the external clock equal to Day 1, 0 Hours, 0 Minutes, 0
Seconds.

CRESET: Clears the GLOBL "LDFLG" and writes system pointer block to disk.

Special Conditions: None

Special Requirements:

FWS 56A-7621 Battery Powered External Clock, DLV11 Parallel Line Unit.

External Requirements: None

Globals:

DROBUF, DRCSR, IDAY
CLOCK

POWER OUT SINCE LAST ENTRY?

YES

STORE DECIMAL SECONDS IN "TIME" AREA

NO

CLEAR POWER OUT TIME STORAGE AREA

SEND FREEZE COMMAND

STORE M/S IN RO. STORE D/H IN R1

MASK OUT BCD SECONDS

SEC

B

A

STORE DECIMAL SECONDS IN "TIME" AREA

POWER OUT SINCE LAST ENTRY?

YES

STORE DECIMAL SECONDS IN "PRIME" AREA

NO

CLEAR POWER OUT TIME STORAGE AREA

SEND FREEZE COMMAND

STORE M/S IN RO. STORE D/H IN R1

MASK OUT BCD SECONDS

MIN

STORE DECIMAL MINUTES IN "TIME" AREA
**B**

POWER OUT SINCE LAST ENTRY?

- YES
  - STORE DECIMAL MINUTES IN "PRIME" AREA
  - MASK OUT HOURS
    - HOURS
      - POWER OUT SINCE LAST ENTRY?
        - YES
          - STORE DECIMAL DAYS IN "PRIME" AREA
          - STORE DECIMAL HOURS IN "TIME" AREA
          - RTS
        - NO
          - STORE DECIMAL HOURS IN "TIME" AREA
          - RTS
      - NO
        - STORE DECIMAL HOURS IN "TIME" AREA
        - RTS

- NO
  - STORE DECIMAL MINUTES IN "PRIME" AREA
  - MASK OUT HOURS
    - HOURS
      - POWER OUT SINCE LAST ENTRY?
        - YES
          - STORE DECIMAL DAYS IN "PRIME" AREA
          - RTS
        - NO
          - STORE DECIMAL HOURS IN "TIME" AREA
          - RTS

**C**

MASK OUT DAYS

**D**

COMPUTE ELAPSED TIME

STORE FOR DIAGNOSTIC DISPLAY

SET POWER OUT FLAG

RTS
CRESET

DISABLE CLOCK INTERRUPT

CLEAR LDFLG

WRITE SYSTEM POINTER BLOCK

RTS

ECLOCK

ENABLE CLOCK INTERRUPT

RTS

SCLOCK

CLEAR CLOCK REGISTERS

SET CLOCK DAY = 1

RTS
**PROGRAM:**

CALLING

SEQUENCE:

MOV DAYS,RO
MOV RO,SR0
JSR PC,DAYS

or

MOV HOURS,RO
MOV RO,SR0
JSR PC,HOURS

or

MOV MIN,RO
MOV RO,SR0
JSR PC,MIN

or

MOV SEC,RO
MOV RO,SR0
JSR PC,SEC

**Real Time Entry:**

Upon entry,
RO = BCD Day, hours minutes or seconds (LSB of RO).
SRO = Globaled storage area set equal to RO.
Converts days, hours, minutes, and/or seconds from BCD to decimal.

**Globals:**

SRO

**PROGRAM:**

BOPY - Disk I/O Handler

**PURPOSE:**

Used by ADV1LA software to transfer data blocks from memory to disk and vice versa.

CALLING

SEQUENCE:

DBLOCK = 1014; FWA of command string
MOV #Cstring, DBLOCK; store your string address.
JSR PC,BOPY

CSTRING: .ASCIZ /BB:FILEOUT.EXT=DX:FILEIN.EXT/

Upon entry, the command string supplied by the caller is transferred to the MACRO Command String Interpreter (CSI). CSI interrogates the command string for legal and illegal file specifications. When CSI determines that the command string is in an acceptable RT11 format, .LOOKUP and/or .ENTER request are performed on the files. CSI aborts the operation and returns to the caller when illegal file specs are found.

Memory location 1012 contains error status set and cleared by BOPY. This memory location should be monitored by the caller after each call to BOPY. Before exiting BOPY location 1012 will be set to zero if the operation is successful, less than zero if a write error is detected or greater than zero if a read error is detected.

After it has been determined that the command string supplied by the user is in a legal RT11 format BOPY performs a copy operation using synchronous I/O, and the output file is made permanent on the device.

Read/write operations are performed on channels 3 and 0 respectively and are transferred in 512 decimal word blocks.
SPECIAL CONDITIONS

Any driver defined in the command string must be resident on the system storage medium.

When linking, BOPY must be the last object in the link string.

SPECIAL REQUIREMENTS

MACRO library must be resident on the system storage medium before linking.

EXTERNAL REQUIREMENTS

.CSIGEN, .READW, .WRITW, .CLOSE, .SRESET, BASE

GLOBALS

DSpace, DEXT, SPACE, OUTSP, INSPI, DBASE

ASSEMBLY

BOPY=BOPY,BASE

PROGRAM:  BASE

PURPOSE:  Used to define storage allocation for all data base variable and constants.

ASSEMBLY:  MACRO

*BOPY=BOPY,BASE

The module base contains the data base variable described below. The variables names used in the data base are all globled and should not be duplicated by newly incorporated software routines.

GENERAL:

The data base allocates 8K words of memory and is comprised of the following:

1. System pointers, program variables that identify relative memory locations within the data block.

2. System flags, program variables that assist in the decision making process of the incorporated software routines.

3. System counters, program variables that establish and retain points of reference during data processing and program execution.

4. Disk storage allocation. Data blocks, four (4) total, comprised of memory storage locations, which allocate enough space to store data for up to 16 A/D channels consecutively for four days. The composite size for each block (Data Only) is 3840K words of memory of which 240K bytes are reserved per channel. In addition, 128K bytes of storage is made available for system pointers.
BOPY

SAVE REGISTERS
DISABLE
CLOCK
INTERRUPT

CSI

ERROR NO INPUT SPEC'S
YES
C

ERROR NO INPUT SPEC'S
NO
B

DIAG "DUMP" REQUEST ?
YES

MODEM ON ?
NO
CLOSE CHANNELS
RELEASE HANDLERS
RESTORE REGISTERS

READ A BLOCK ON CHANNEL #3

END OF FILE
YES
CLOSE CHANNELS
RELEASE HANDLERS
RESTORE REGISTERS
NO
A
C
RTS

READ ERROR ?
YES
SET READ ERROR FLAG

WRITE A BLOCK ON CHANNEL #0

WRITE ERROR ?
NO
CLOSE CHANNELS
RELEASE HANDLERS
RESTORE REGISTERS

WRITE ERROR ?
YES
SET WRITE ERROR FLAG

B

RTS
CLEAR LEFT 12 BITS OF INITIAL VALUE AND SAVE REMAINDER IN R5

CLEAR ALL BITS EXCEPT 5 IN INITIAL VALUE

YES

BIT 6 = 0?

NO

ADD 10 TO DECIMAL VALUE, R5

PICKUP ORIGINAL VALUE

CLEAR ALL BITS EXCEPT 6 IN INITIAL VALUE

YES

BIT 6 = 0?

NO

ADD 20 TO DECIMAL VALUE, R5

PICKUP ORIGINAL VALUE

PU

NO

BIT 8 = 0?

NO

ADD 80 TO DECIMAL VALUE, R5

PICKUP ORIGINAL VALUE

QU

YES

BIT 8 = 0?

NO

ADD 100 TO DECIMAL VALUE

PICKUP ORIGINAL VALUE

RTS

NO

BIT 10 = 0?

NO

ADD 200 TO DECIMAL VALUE

PICKUP ORIGINAL VALUE

QU

YES

BIT 10 = 0?

NO

ADD 100 TO DECIMAL VALUE

PICKUP ORIGINAL VALUE

RTS
Main memory storage allocation allocates enough storage for two data blocks. The first is used for current day data storage and the second is used for "DUMP" data storage. In addition, 128K bytes of storage have been added to maintain system pointers and program variables. This block is stored on disk every 6 minutes and may be referenced as "SYSTEM.PNT".

"DUMP" reads data blocks from disk to main memory and transfers same via a modem link.

The current day data block is updated every 6 minutes by the data acquisition software and written to disk. This procedure is repeated for 24 hours. At the 24 hour mark the data block last written will be located on disk as DAY1.DAT, DAY2.DAT, DAY3.DAT or DAY4.DAT.

Once all four data blocks have been written to disk the above procedure is repeated, overlaying DAY1.DAT, 2, 3 and so on.

5. Common Storage, global scratch area shared by all system software routines.
BEGIN DATA BASE VARIABLES ADDRESS ASSIGNMENTS HERE

; COMMON STORAGE AREA

; GLOBLS....

; DBASE, LDFLG, IPM, N, CHANL, IUCNT, IUCNT1, IUCNT2, JDAY
; LDFLG, BASE, LOAD FLAG,... EVENT
; IPM = LDFLG + 2 ; INCREMENT SINCE MIDNIGHT
; N = IPM + 2 ; RELATIVE POINTER TO CURRENT "M" ARRAYS
; CHANL = N + 2 ; CURRENT A/D CHANNEL 0-15
; IUCNT = CHANL + 2 ; UNITARY BIT COUNT, TRB ON/OFF STATUS BITS
; IUCNT1 = IUCNT + 2 ; UNITARY BIT COUNT, TRB 1
; IUCNT2 = IUCNT1 + 2 ; UNITARY BIT COUNT, TRB 2
; IUCNT3 = IUCNT2 + 2 ; UNITARY BIT COUNT, TRB 3
; JDAY = IUCNT3 + 2 ; JULIAN DAY COUNT
; MINS = JDAY + 2 ; ONE MINUTE COUNT
; CDAY = MINS + 2 ; CURRENT DAY COUNTER
; LEEPYR = CDAY + 2 ; LEEP YEAR COUNT
; SIX = LEEPYR + 2 ; SIX MINUTE COUNT
; SAVES = SIX + 2 ; MOST RECENT SAMPLE
; SAVES1 = SAVES + 2 ; MOST RECENT SAMPLE TRB 1 ON/OFF
; SAVES2 = SAVES1 + 2 ; MOST RECENT SAMPLE TRB 2 ON/OFF
; SAVES3 = SAVES2 + 2 ; MOST RECENT SAMPLE TRB 3 ON/OFF
; CSTAT = SAVES3 + 2 ; CAL ON/OFF STATUS BITS
; CALON = CSTAT + 2 ; CALIBRATION ON/OFF FLAG
; TRBN = CALON + 2 ; TURBINE #'S, 0=NONE, 1=THREE
; MAXCH = TRBN + 2 ; MAXIMUM CHANNEL
; IDAY = MAXCH + 2 ; INITIAL DAY COUNT
; M = IDAY + 2 ; RELATIVE DAY POINTER TO DATA BLOCK ARRAYS
; TIME = M + 2 ; EXTERNAL CLOCK TIME DECIMAL
; ; DATED, HOURS, MINUTES, SECONDS
; ; UPDATED EVERY 6 MINUTES
; PURGE = TIME + 10 ; PURGE FLAG LOCATIONS, DAY 1-4
; PTIME = PURGE + 6 ; POWER OUT TIME STORAGE
; DABAS = DABAS + 42. ; DATE RESTORE... STORAGE, ASCII STRING
; DTBAS = DABAS + 44. ; END OF SYSTEM POINTERS, EOF
; ; DATA ONLY; STORED EVERY 6 MINUTES
; ADC1M1 = ENDFLG + 2 ; DBLOCK-M, CH-1 CURRENT DAY STORAGE
; ADC2M1 = ADC1M1 + 360 ; DBLOCK-M, CH-2 CURRENT DAY STORAGE
; ADC3M1 = ADC2M1 + 360 ; DBLOCK-M, CH-3 CURRENT DAY STORAGE
; ADC4M1 = ADC3M1 + 360 ; DBLOCK-M, CH-4 CURRENT DAY STORAGE

; **END OF DATA BASE VARIABLE ADDRESS ASSIGNMENTS**
ADC5M1 = ADC4M1 + 360 ; DBLOCK-M, CH-5 CURRENT DAY STORAGE
ADC6M1 = ADC5M1 + 360 ; DBLOCK-M, CH-6 CURRENT DAY STORAGE
ADC7M1 = ADC6M1 + 360 ; DBLOCK-M, CH-7 CURRENT DAY STORAGE
ADC8M1 = ADC7M1 + 360 ; DBLOCK-M, CH-8 CURRENT DAY STORAGE
A9 = ADC8M1 + 360 ; DBLOCK-M, CH-9 CURRENT DAY STORAGE
A10 = A9 + 360 ; DBLOCK-M, CH-10 CURRENT DAY STORAGE
A11 = A10 + 360 ; DBLOCK-M, CH-11 CURRENT DAY STORAGE
A12 = A11 + 360 ; DBLOCK-M, CH-12 CURRENT DAY STORAGE
A13 = A12 + 360 ; DBLOCK-M, CH-13 CURRENT DAY STORAGE
A14 = A13 + 360 ; DBLOCK-M, CH-14 CURRENT DAY STORAGE
A15 = A14 + 360 ; DBLOCK-M, CH-15 CURRENT DAY STORAGE
A16 = A15 + 360 ; DBLOCK-M, CH-16 CURRENT DAY STORAGE
STOPE = A16 + 362 ; EOF.....6 MINUTE DATA STORAGE AREA.

; THE FOLLOWING VARIABLES MUST BE INITIALIZED UPON POWER UP

; FROM POWER OUT.
NSAMP = STOPE + 2 ; # OF ACCUMULATED SAMPLES
SR0 = NSAMP + 2 ; REGISTER
SR1 = SR0 + 2
SR2 = SR1 + 2
SR3 = SR2 + 2 ; STORAGE
SR4 = SR3 + 2
SR5 = SR4 + 2 ; AREA
SHIFT = SR5 + 2 ; SHIFT COUNT....UNITM
SHFT1 = SHFT + 2 ; TRB #1 SHIFT COUNT
SHFT2 = SHFT1 + 2 ; SHIFT COUNT FOR TRB 2
SHFT3 = SHFT2 + 2 ; TRB #3 SHIFT COUNT
ERBYT = SHFT3 + 2 ; DISK ERROR FLAG
DBLOCK = ERBYT + 2 ; FPGA OF CURRENT COMMAND LINE FOR BODY'
IFLAG = DBLOCK + 2 ; INITIAL DAY FLAG
JFLAG = IFLAG + 2 ; JULIAN DAY FLAG
MODEMF = JFLAG + 2 ; MODEM "QUE" FLAG

; BEGIN DUMP/DPURGE DATA STORAGE AREA HERE....
DMPS = MODEMF + 2 ; STORAGE AREA FOR DUMP/DPURGE

; BEGIN DRIVER LOAD AREA HERE....
OUTSP = DMPS + 10004 ; DRIVERS
INSP = OUTSP + 36 ; START
SPACE = INSP + 39 ; LOADING
DSPACE = OUTSP ; HERE
PROGRAM: BB.SYS  BBL.SYS - Device Handlers

PURPOSE: Handle I/O from disk to memory and visa versa.

BB.SYS: Reads and prepares data blocks for transmission to disk as files.

BBL.SYS: Reads files from disk and transmits same into prescribed memory locations or remotely across telephone lines.

CALLING SEQUENCE: Invoked via RT11 command string interpreter (CSI).

BB/BBL

BB and BBL where written and structured according to standard RT11 programming conventions for device handlers. Each contain the following sections:

- Preamble
- Header
- I/O Initiation
- Interrupt Service
- I/O Completion
- Handler Termination

These sections are discussed in the RT11 Software Support Manual - section 7.2.

LOADING AND EXECUTING

Read/write request are submitted to BB/BBL.SYS via the ADV11A Data Acquisition software or the ADV11A Diagnostic Software. Both utilize the file copy subroutine 90PY to define command string interpreter (CSI) file specifications and load BB/BBL and disk drivers.

BB.SYS

BB.SYS is a read only driver whose basic function is to pass data from a data block or system pointer block storage area in memory to an I/O buffer area shared by resident disk drivers. The data in the I/O buffer area is used by the disk drivers to create files on disk.

When entered, BB.SYS first determines whether or not a request has been made to read a file from disk. If a "READ FILE" request is issued BB.SYS transfers program control to BBL.SYS. If not, BB.SYS determines whether or not a request has been made to write a data block or system pointer block to disk.

BB.SYS obtains the appropriate memory address of the data block or system pointer block then begins transferring the data contained in the block to the I/O buffer shared by a resident disk driver.

The disk driver transfers the data in the I/O buffer area to disk in 512 decimal word blocks. When the last block of the data block or system pointer block is written to disk a filename and extension is written in the RT11 header block (block 6) and made permanent on the disk medium. The file name/extension written is passed to the disk driver by the CSI processor.

BB.SYS terminates its operation after passing eight 512 decimal word blocks of the data block or after passing one 512 decimal word block of the system pointer block.
BB1.SYS

BB1.SYS is a write only driver whose basic function is to read and transmit files from the disk drivers I/O buffer area to memory locations reserved for data block and system pointer storage. In addition, BB1.SYS performs the function of transmitting files data across the phone line during diagnostic dump operations.

BB1.SYS is entered via BB.SYS. Its first operation makes a decision whether or not to transfer the I/O buffer data to the data block and system pointer storage area or across the phone line.

The total number of blocks (8 or 1) for the data block or system pointer file is stored in their respective memory storage area when the user request is not a diagnostic dump operation.

When a diagnostic dump request is issued BB1.SYS reads the files data from the I/O buffer, unpacks and converts each word to ASCII then transmits same down the phone line. Each 512 decimal word block transmitted is followed by a carriage return line feed until the last block is transmitted. At the completion of the diagnostic dump operation the ASCII characters ".EXIT" are transmitted and BB1.SYS aborts.

SPECIAL REQUIREMENTS

Modem device assignments must start at memory address 1775608.

MACRO library.

EXTERNAL REQUIREMENTS

.DRBBEG, .DRAST, .FORK, .DRFIN, .DREAD

LIMITATIONS

Restricted for use by ADV11A Data Acquisition and Diagnostic Software only.

PROGRAM: RD RP - RCP Device Handlers

PURPOSE: Remote communications I/O processing.

RD.SYS: Read data from Line/Write to Disk.

RP.SYS: Read data from Disk/Write to Line.

CALLING SEQUENCE: Envoked via RT11 command string interpreter (CSI).

RD/RP

RD and RP were written and structured according to standard RT11 programming conventions for device handlers. Each contain the following sections:

* Preamble
* Header
* I/O Initiation
* Interrupt Service
* I/O Completion
* Handler Termination

These sections are discussed in the RT11 Software Support Manual section 7.2.
RD.SYS

RD.Sys is a read only driver whose basic function is to read data from an incoming modem line and write same to disk.

Upon entry, RD transmits a DC1 (218) character down the line to initiate telephone conversation. The driver at the other end (RP) is expected to respond by sending the total block count of the file to be read followed by the files contents.

RD reads and packs bytes from the line, storing same into an I/O buffer area shared by a resident disk handler. This buffer area is 512 decimal words in length. Each time the buffer area is exhausted RD decrements the block count and the buffer area gets written to disk. This procedure is repeated until the block count equals zero or until the modem carrier is dropped.

Block transfers less than 512 are written to disk with trailing zeros.

After the last block is written to disk RD performs handler termination procedures then exits to the caller.

RP.SYS

RP.Sys is a write only driver whose basic function is to transmit data from a file located on a resident disk to a modem line.

Upon entry, RP sits in a loop awaiting a DC1 (218) character from the incoming line. If the DC1 character is received RP transmits a block count followed by a file of data in 512 decimal word blocks. If the DC1 character is not received and the carrier drops RP exits to the caller.

The block count of the file is preloaded by the RCOPY/RPIP routines and is resident within these programs at memory location 10028.

The data from the file to be transferred is stored in an I/O buffer area shared and maintained by a resident disk handler. The resident disk handler fills the buffer area with data from the file and RP extracts it and sends byte for byte down the line. After the last block of the file is transmitted RP performs handler termination procedures then exits to the caller.

Special Conditions - RP/RD

Following each byte transmission RP transmits a status byte. The status byte when set to a 1 indicates to RD.Sys that the byte preceding the status should have bit 8 set. RD sets bit 8 dependent on the status byte then packs same into its I/O buffer. The status byte is merely tested then ignored.

Status byte transmissions have been incorporated to accommodate the 8-bit even parity configuration of the modem.
LOADING AND EXECUTING

See ROOPY/RPIP Documentation, this manual.

LIMITATIONS

File transfers are restricted to 127 decimal blocks maximum, per transfer.

SPECIAL REQUIREMENTS

The CSR address for the disk driver must be set at 177170. Modem device assignments must start at 177560. MACRO library must be resident on the system storage medium before linking.

EXTERNAL REQUIREMENTS

.DRFIN, .DREN, .DRBEG, .DRAST, .FORK
FORK - PROGRAM REENTRY

NO

Y

PROCESSI

SYSTEM

POINTER

..........

YES

BB1.SYS

GET DISK
BUFFER FWA
GET DISK
BUFFER SIZE

ILLEGAL
FILE
SPEC'S?

NO

FIRST
TIME IN?

YES

DATA
BLOCK
PROCESSING?

YES

PICKUP
FWA OF
DATA AREA

NO

SYSTEM
 POINTER

BLOCK
PROCESSING?

YES

PICKUP FWA
OF POINTER
AREA

NO

DIRECTORY
REQUESTED
BY DIAG?

YES

TRANSMIT
ASCII DIRECTORY
BLOCK TO LINE

NO

MODEM
ON?

YES

NC

C

B

B

B
READ A DATA BYTE FROM LINE

READ A STATUS BYTE FROM LINE

STATUS > 0?

LOR 200B WITH DATA BYTE

STORE DATA BYTE IN DISK BUFFER

DECREMENT BYTE COUNT

TRANSMIT DC1

TRANSMIT DATA AND STATUS BYTES

RECEIVE DC1
B

CLEAR BIT 8 OF DATA BYTE

TRANSMIT DATA BYTE TO LINE

TRANSMIT STATUS BYTE TO LINE

CLEAR STATUS BYTE

BUMP BYTE COUNT

A
### ADVILA STORAGE AREA

<table>
<thead>
<tr>
<th>CHANL</th>
<th>(Ch-1)</th>
<th>(Ch-2)</th>
<th>(Ch-3)</th>
<th>(Ch-4)</th>
<th>(Ch-5)</th>
<th>(Ch-6)</th>
<th>(Ch-7)</th>
<th>(Ch-8)</th>
<th>(Ch-9)</th>
<th>(Ch-10)</th>
<th>(Ch-11)</th>
<th>(Ch-12)</th>
<th>(Ch-13)</th>
<th>(Ch-14)</th>
<th>(Ch-15)</th>
<th>(Ch-16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMP</td>
<td>AVGS</td>
<td>AVGS</td>
<td>AVGS</td>
<td>AVGS</td>
<td>AVGS</td>
<td>AVGS</td>
<td>AVGS</td>
<td>UNITIS</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
</tr>
<tr>
<td></td>
<td>Ch-1</td>
<td>Ch-2</td>
<td>Ch-3</td>
<td>Ch-4</td>
<td>Ch-5</td>
<td>Ch-6</td>
<td>Ch-7</td>
<td>Ch-8</td>
<td>Ch-9</td>
<td>Ch-10</td>
<td>Ch-11</td>
<td>Ch-12</td>
<td>Ch-13</td>
<td>Ch-14</td>
<td>Ch-15</td>
<td>Ch-16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMIN</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>UNITIM</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
<td>RTSB</td>
</tr>
<tr>
<td></td>
<td>Ch-1</td>
<td>Ch-2</td>
<td>Ch-3</td>
<td>Ch-4</td>
<td>Ch-5</td>
<td>Ch-6</td>
<td>Ch-7</td>
<td>Ch-8</td>
<td>Ch-9</td>
<td>Ch-10</td>
<td>Ch-11</td>
<td>Ch-12</td>
<td>Ch-13</td>
<td>Ch-14</td>
<td>Ch-15</td>
<td>Ch-16</td>
</tr>
</tbody>
</table>

Specified channel number contained in most significant byte of each word.

Address of appropriate subprocess for each channel for each sample taken.

Address of appropriate subprocess for each channel. Entered once per minute.
<table>
<thead>
<tr>
<th>VBASE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>366</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TLSIX</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG6</td>
<td>Ch-1</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-2</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-3</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-4</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-5</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-6</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-7</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-8</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-9</td>
</tr>
<tr>
<td>UNIT6</td>
<td>Ch-10</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-11</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-12</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-13</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-14</td>
</tr>
<tr>
<td>AVG6</td>
<td>Ch-15</td>
</tr>
<tr>
<td>UNIT6</td>
<td>Ch-16</td>
</tr>
</tbody>
</table>

Address of appropriate subprocess for each channel. Entered once every six minutes.

Maximum channel
6 minute count constant
6 minute count constant
Leap year constant
SECTION V  SUBPROCESS ENTRIES
PROGRAM: RTSB

PURPOSE: No Operation

CALLING SEQUENCE: MOV #TABLE,RI ; #TABLE = FWA of subprocess table
JSR PC,@(R1)

Real Time Entry:
The address of process RTSB, a no operation, is contained in the inactive locations of the subprocess tables, SAMP, TIMIN, and TISIX. The execution of the process RTSB initiates a Return From Subroutine (RTS).

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals: None

PROGRAM: AVGS

PURPOSE: Accumulate and average groups of sample data

CALLING SEQUENCE: MOV #SAMP,RI ; FWA of sample table
JSR PC,@(R1)

Upon entry, RO = FWA of Turbine Data current channel.

Real Time Entry:
When no power outages are evident, AVGS is entered 10 times per second, via the A/D software driver ADV11A. AVGS accumulates data samples in Table TB1180, takes an average of the result, then accumulates the average quotient in a common storage area, Table TB124. This procedure takes place 4 times per minute and 24 times per six minutes. At the six minute mark, the quotients summed for each channel are averaged and stored into the data block by process AVG6. Within each one minute period, the first three accumulations consist of 180 data samples per channel. The fourth accumulation consists of 60 data samples per channel. The common storage areas TB1180 and TB124 are set equal to 0 by EVENT and LOGIC6 routines.

The Mathematical formula used to compute the average of 180 data samples per channel is as follows:

\[
\text{Common (channel)} = \frac{\sum_{i=1}^{180} \text{Data } i + \text{Common (channel)}}{180}
\]

where ... Common = common storage area for quotient accumulations
Channel = Channel data locations 1 through 7 in the common storage area, stated above.
Data = 180 data samples per channel
The mathematical formula used to compute the average of 60 data samples per channel is as follows:

\[ \text{Common (channel)} = \frac{\sum_{i=1}^{60} \text{Data}_i + \text{Common (channel)}}{60} \]

where...
- **Common** = common storage area for quotient accumulations
- **Channel** = channel data locations 1 through 7 in the common storage area, stated above.
- **Data** = 60 data samples per channel

**NOTE:** Each TB1180 Location is reinitialized before the respective quotient is accumulated in TB124.

**Special Conditions:** None

**Special Requirements:** None

**External Requirements:** None

**Globals:**
- TB1180, TB124, CHANEL, NSAMP
**AVGS STORAGE AREA**

<table>
<thead>
<tr>
<th>TB1180</th>
<th>(Ch-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Ch-2)</td>
</tr>
<tr>
<td></td>
<td>(Ch-3)</td>
</tr>
<tr>
<td></td>
<td>(Ch-4)</td>
</tr>
<tr>
<td></td>
<td>(Ch-5)</td>
</tr>
<tr>
<td></td>
<td>(Ch-6)</td>
</tr>
<tr>
<td></td>
<td>(Ch-7)</td>
</tr>
<tr>
<td></td>
<td>(Ch-8)</td>
</tr>
<tr>
<td></td>
<td>(Ch-9)</td>
</tr>
<tr>
<td></td>
<td>(Ch-10)</td>
</tr>
<tr>
<td></td>
<td>(Ch-11)</td>
</tr>
<tr>
<td></td>
<td>(Ch-12)</td>
</tr>
<tr>
<td></td>
<td>(Ch-13)</td>
</tr>
<tr>
<td></td>
<td>(Ch-14)</td>
</tr>
<tr>
<td></td>
<td>(Ch-15)</td>
</tr>
<tr>
<td></td>
<td>(Ch-16)</td>
</tr>
</tbody>
</table>

Stores sum of data samples
AVGS

SAVE REGISTERS

POSITION POINTERS AT CURRENT CHANNEL STORAGE LOCATION IN ACCUMULATION BUFFER

ADD CURRENT DATA TO PREVIOUS TOTAL IN ACCUMULATION BUFFER

IS NSAMP = 175, 359 OR 539?

YES

DIVIDE ACCUMULATION BUFFER VALUE BY 180 & ROUND

NO

IS NSAMP = 599?

YES

DIVIDE ACCUMULATION BY 60 & ROUND (THIS CHANNEL)

RESTORE REGISTERS

RTS

NO

DIVIDE ACCUMULATION BUFFER

ADD QUOTIENT TO ACCUMULATED AVERAGES BUFFER (THIS CHANNEL)

CLEAR ACCUMULATION BUFFER (THIS CHANNEL)

ADD QUOTIENT TO ACCUMULATED AVERAGES BUFFER

RTS

RTS
PROGRAM: UNITS

PURPOSE: Save present ON/OFF turbine status

CALLING SEQUENCE: MOV #SAMP, R1 ; FWA of sample table
                 JSRPC, @R1

Real Time Entry:
R0 - FWA of turbine data.
UNITs monitors channels data 10 times per second over a period of one minute
to determine (1) whether the turbine is on or off and/or (2) whether "Data
Calibration" is in progress. The turbine is determined to be on or off based
on a tolerance defined and supplied by the user. The data associated with
the channel connected to the process UNITS is compared to this tolerance and
when greater than or equal to denotes the on status of the turbine.

The globalled SAVES is set to a '1' or '0' after the turbine is determined to
be on or off. When set to a '0', SAVES reflects the OFF status of the turbine.
When set to a '1', SAVES reflects the ON status of the turbine. This ON/OFF
status bit is unitarily packed into 6 bits of a single word every minute for
6 minutes then stored in the data block. Packing and storing channels data
is a function of the processes UNITM and UNIT6.

Data calibration is considered "in progress" when the channels data is stable at
0 to 15 counts for a period of at least 30 seconds. Consequently, the globalled
"CALON" flag is set to a 1 to reflect the calibration ON status and to a 0
to reflect the calibration OFF status as determined by the state of the channels
data within this 30 second period. Within the subprocess UNITM and once per
minute, CALON bit 1 is unitarily packed into 6 bit positions of the calibration
status word CSTAT. CSTAT represents the calibration status over a period of
6 minutes.

In addition, UNITS monitors data calibration to determine whether any channel
calibrates consecutively for more than 1 minute. When any calibration channel
exceeds a 1 minute calibration limit the calibration error flag, CALERR, is set
to 1. This flag is monitored by the sub-process CALB6. CALB6 sets an error
bit in bit position #8 of CSTAT as determined by the state of CALERR.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals: SAVEs, MINS
UNITS

TURBINE ON?

NO A

YES

CALIBRATION ON?

NO B

YES

DATA CAL FOR 30 SECONDS?

NO

YES

SET CAL ON FLAG

DATA CAL EXCEED 1 MINUTE?

NO B

YES

SET CAL ERROR FLAG

SET TURBINE ON STATUS

SET TURBINE OFF STATUS

CLEAR ALL WORKING VARIABLES

RTS
PROGRAM: UNITM

PURPOSE: Pack turbine and calibration ON/OFF status

CALLING SEQUENCE: MOV #TMIN, R1 ; FWA of minute table
JSR PC, @(R1)

Real Time Entry:
Upon entry,
SAVES = Turbine ON/OFF status.
CAL/N = Calibration ON/OFF status.
UNITM is entered once per minute to retrieve and unitarily pack channels data into 6 bit positions of the gloabed IUCNT and CSTAT. Each of the globals are contained in the data base and made accessible to all subprocesses.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:
SAVES, CALON, SHIFT, IUCNT, CSTAT, SHIFT1, SHIFT2, IUCUT1, IUCUT2, SAVES1, SAVES2, SAVES3, IUCNT3

PROGRAM: AVG6

PURPOSE: Computes final average of data for a six minute period.

CALLING SEQUENCE: MOV #TISIX, R1 ; R1 = FWA of 6 minute table
JSR PC, @(R1)

Real Time Entry:
Upon entry,
RS = FWA of Day-M(n) for the current channel, located within the data block of the current day.
AVG6 divides the accumulated averages of Table Tbl24 by 24 and stores the quotient in the data block. These quotients are the final average for each data channel over a six minute period and are within 1% of the absolute value.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals:
Tb124, CHANEL
<table>
<thead>
<tr>
<th>TB 124</th>
<th>(Ch-1)</th>
<th>(Ch-2)</th>
<th>(Ch-3)</th>
<th>(Ch-4)</th>
<th>(Ch-5)</th>
<th>(Ch-6)</th>
<th>(Ch-7)</th>
<th>(Ch-8)</th>
<th>(Ch-9)</th>
<th>(Ch-10)</th>
<th>(Ch-11)</th>
<th>(Ch-12)</th>
<th>(Ch-13)</th>
<th>(Ch-14)</th>
<th>(Ch-15)</th>
<th>(Ch-16)</th>
</tr>
</thead>
</table>

Stores sum of averaged data samples
AVG6

PICKUP FWA OF CHANNELS STORAGE LOCATION

C-LOCATION = 0 ?

YES

CLEAR RØ FOR INTEGER DIVIDE

NO

CLEAR RØ FOR INTEGER DIVIDE

DIVIDE ACCUMULATED AVERAGE BY 24 AND ROUND

DLOSS FLAG GT 0 ?

YES

DALOS

NO

STORE DATA INTO DATA BLOCK

CLEAR ACCUMULATED AVERAGE

RTS
PROGRAM: UNIT6

PURPOSE: Stores turbine ON/OFF status word in the data block.

CALLING SEQUENCE: MOV #T1SIX,R1 ; six minute table
                  JSR PC,@(R1)

Real Time Entry:
Upon entry,
R5 = FWA of Day-M(n) located within the data block of the current day.
IUCNT = Turbine ON/OFF status word, updated by UNITM.
Every six minutes UNIT6 stores the turbine ON/OFF status word into the data
block at a location reserved for the channel. Upon exit, IUCNT is reinitialized.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals: IUCNT, CALERR, CSTAT, IUCNT1, IUCNT2, IUCNT3

PROGRAM: CALB6

PURPOSE: Stores calibration status word and checks for a calibration error.

CALLING SEQUENCE: MOV #T1SIX,R1 ; R1 = FWA of 6 minute table
                  JSR PC,@(R1)

Real Time Entry:
Upon entry,
R5 = FWA of Day-M(n) located within the data block of the current day.
CALERR = Calibration error flag, set by UNITM
CSTAT = Calibration ON/OFF status word, updated by UNITM
Every 6 minutes CALB6 performs the following functions:
1. Verifies the CALERR flag;
2. Sets bit 8 of CSTAT to '1' when CALON is greater than 0,
3. And/or, stores CSTAT into the current data block location.
Upon exit CSTAT and CALERR are reinitialized.

Special Conditions: None

Special Requirements: None

External Requirements: None

Globals: CALERR, CSTAT, IUCNT
CALB6

CALIBRATION ERROR?

YES

SET ERROR STATUS BIT IN CAL WORD

NO

1

DLOSS FLAG -GT- 0?

YES

DALOS

NO

STORE CAL ON/OFF STATUS INTO DATA BLOCK

1

2

RTS
SECTION VI  SYSTEM UTILITIES
PROGRAM: TRAP


CALLING SEQUENCE: Executed by the RTU monitor via some trap vector address (described below).

<table>
<thead>
<tr>
<th>Vector</th>
<th>Trap Entry Name</th>
<th>Trap Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>004</td>
<td>TRAP4</td>
<td>Caused by a reference to a non-existing device or odd address, or by a bus time-out.</td>
</tr>
<tr>
<td>010</td>
<td>TRAP10</td>
<td>Caused by an attempt to execute a reserved or illegal instruction.</td>
</tr>
<tr>
<td>014</td>
<td>TRAP14</td>
<td>Breakpoint Trap, usually issued by a debugging utility program such as ODT.</td>
</tr>
<tr>
<td>020</td>
<td>TRAP20</td>
<td>Input Output TRAP, Program I/O TRAP.</td>
</tr>
<tr>
<td>030</td>
<td>TRAP30</td>
<td>BMT Emulator Trap</td>
</tr>
<tr>
<td>034</td>
<td>TRAP34</td>
<td>Program Trap Instruction, issued by a program to change the flow of execution.</td>
</tr>
<tr>
<td>114</td>
<td>T114</td>
<td>Memory Parity Trap, caused by a memory parity error.</td>
</tr>
<tr>
<td>244</td>
<td>T244</td>
<td>FPU Trap, caused by the Floating Point Unit exception or error.</td>
</tr>
<tr>
<td>264</td>
<td>T264</td>
<td>Disk I/O error.</td>
</tr>
</tbody>
</table>

Special Conditions:
The address of each entry per trap must be stored in the respective Trap Vector Address before program execution can take place.

Globals:
TP4, TP4A, TP10, TP10A, TP30, TP30A, TP34, TP34A, TP114, TP114A, TP244, TP244A, TP264, TP264A, TP14, TP14A

PROGRAM: RCOPY RPIP - Remote file XFR processing.

PURPOSE: Establish and oversee I/O between the remote communications drivers RD.SYS and RP.SYS.

CALLING SEQUENCE: RUN RCOPY
or
RUN RPIP

RCOPY.SAV
RCOPY communicates with the performance monitor data acquisition software via modem.

In an effort to simplify yet afford RCOPY with adaptability it has been incorporated as a stand alone system "RUN" routine for a RT11 base system.

In its basic form RCOPY solicits information from the user via the system terminal. This information is used when preparing communication between the RCP drivers.
Command strings describing the drivers and file names of the files to be read remotely and transferred locally are input to ROOPY by the user. ROOPY interrogates the remote command string and looks for valid RCP driver definitions and string formats. Invalid string formats are not accepted. The user is requested to reenter the command upon error. However, valid command strings are passed down the modem line to the data acquisition software. When received, the data acquisition software loads and executes the appropriate drivers. Thereafter the drivers await file transmission.

After ROOPY transmits the remote command string the user is asked to input the block count of the file. Once entered, ROOPY decodes and stores the block count for later use by the RP driver. The block count is resident at memory location 10028.

Next, ROOPY invokes the command string interpreter (CSI) to solicit from the user a local command string file specifications. CSI interrogates the string for legal and illegal file specifications. Illegal file specs are not accepted. The user is requested to reenter the specs upon error.

CSI performs .LOOKUP and/or .ENTER request and loads the drivers defined in the local command string, when legal file specs are input by the user.

After the local drivers are loaded ROOPY issues READ/WRITE commands to the drivers.

Upon each READ/WRITE request the drivers perform the functions defined in the local command string file specification and transmits down the modem line the block count followed by file data. A file transfer is completed when the last block of the file is transmitted. The block count is transmitted once per file.

At the completion of each file transfer ROOPY is reinitialized and the above procedures repeated.

RPIP.SAV

RPIP performs functions identical to ROOPY. However, as a run routine RPIP communicates with the performance monitor at the RT11 executive or monitor level. That is, RPIP is designed to communicate with the RT11 system routine PIP (Peripheral Interchange Program).

RPIP.SAV was incorporated to safeguard and reassure file transfer capability in the event that the performance monitor data acquisition software gets corrupted and is unusable.
PROGRAM/OPERATOR INTERCHANGE - Example Format

PROGRAM: Enter command line for remote filespecs - [Type "HELP" if not sure]
OPERATOR: DXO:FILEOUT:EXT=RD:FILEIN.EXT

PROGRAM: Enter block count for local filespecs (Octal)
OPERATOR: 41

PROGRAM: Enter command line for local filespecs.
OPERATOR: RP:FILEOUT.EXT=DXO:FILEIN.EXT

In the example above, FILEOUT.EXT is the name of the file to be written, FILEIN.EXT is the name of the file to be read. Similarly, FILEIN.EXT will be read locally from DXO: and written to modem via RP:. RD will read FILEOUT.EXT from RP: and transfer same to DXO:. The total number of blocks transferred for FILEIN or FILEOUT is 418. When the transfer is complete DXO: will contain a file of 418 blocks named FILEOUT.EXT.

Filenames FILEOUT.EXT and FILEIN.EXT may be replaced by a file name resident on the local and/or remote disk storage medium.

I/O ERRORS

PROGRAM: Illegal command line for remote FILESPEC.S.
OPERATOR: Reenter legal command line.

PROGRAM: Input error.
OPERATOR: The input filename can not be found on the local storage medium. Reenter new local filespecs.

PROGRAM: Output error.
OPERATOR: The channel to the modem is not on. Dial and connect modem to the channel. Operator must start RPIP/RCOPY program from the beginning.

DIRECTORY LIST

File names, extentions and block counts may be obtained via the following directory command.

.DIR/OCTAL FILENAME.EXT

Special Conditions: None

Special Requirements:

MACRO library must be resident on the system medium before linking.

External Requirements:

.CSIGN, .PRINT, .EXIT, .READW, .WRITW, .CLOSE, .SRESET

ASSEMBLY

RCOPY = RCOPY
RPIP = RPIP

LINKING

RCOPY = RCOPY
RPIP = RPIP
RCOPY

RPIP

S

GET REMOTE FILE SPEC'S

NO

HELP REQUEST?

NO

GET BLOCK COUNT

SAVE AT MEMORY LOCATION 10028

C

IlLEGAL FILE SPEC'S?

YES

NO

S

S

S

S

S

NO

TYPE HELP FILE

YES
PROGRAM: DTOASC OCTA

PURPOSE:

DTOASC: Decimal to ASCII Conversion process
OCTA: Octal to ASCII Conversion process.

CALLING SEQUENCE:

DTOASC

.GLOBL NDTA,NDTAB
MOV Value,NDTA ;NData=Decimal Value
JSR PC,DTOASC
NOP
MOV #NDTAB,R0 ;R0=FWA of 3 word ASCII Decimal Value
.
.
.END

OCTA

.GLOBL BDATAB,CDATA
MOV Value,BDATAB; BDATAB=Octal Value
JSR PC,OCTA
NOP
MOV #CDATA,R0;R0=FWA of 3 word ASCII Octal Value
.
.
.END

Special Conditions: None

Special Requirements:
DTOASC, Floating Point Hardware

External Requirements: None

Globals:
DTOASC - NDTA, NDTAB
OCTA - BDATAB, CDATA
SECTION VII DEFINITIONS
DEFINITIONS

absolute address - A binary number that is assigned as the address of a physical memory storage location.

algorithm - A prescribed set of well-defined rules or processes for the solution of a problem; a program.

analog - Data characters which are not coded into discreet separate pulses or signal levels.

ASCII - The American Standard Code for Information Interchange, consisting of 128 7-bit binary codes for upper and lower case letters, numbers, punctuation, and special communication control characters.

asynchronous - A mode of operation in which an operation is started by a signal that the operation on which it depends is completed. When referring to hardware devices, it is a method in which each character is sent with its own synchronizing information. The hardware operations are scheduled by ready and done signals rather than by time intervals. In addition, it implies that a second operation can begin before the first operation is completed.

baud - (a) A unit of signaling speed equal to the number of signal events per second. (b) For asynchronous events, the unit of modulation rate corresponding to one unit interval per second. If, for example, the length of the unit's interval is 25 milliseconds, the modulation rate is 40 baud. Baud is frequently, though erroneously, used as a synonym for bits per second.

BCD - Binary Coded Decimal

bit - a binary digit.

bootstrap loader - A routine whose first instructions are sufficient to load the remainder of itself into memory from an input device and normally start a complex system of programs.

buffer - A storage area used to temporarily store information being transferred between two devices or between a device and a memory. A buffer is often a special register or a designated area of memory.

byte - The smallest memory addressable unit of information in a computer system. A byte is equivalent to 8 bits.

calibration - The encoding, per channel, of standard voltages by which data may be sequentially adjusted for slope and offset errors.

carrier - Two sets of frequencies used to transmit data over lines where Frequency 1 represents a '0' and Frequency 2 represents a '1'.

console module - A hardware device which is programmable through specified hardware registers.

data base - A collection of interrelated data items organized by a consistent scheme that allows one or more applications to process the items without regard to physical storage locations.
diagnostic - Pertaining to the detection and isolation of malfunctions or mistakes.

digital - Data characters which are coded into discreet separate pulses or signal levels.

dump - To copy the content of all or part of core memory, usually onto an external storage medium. Also the copy so produced.

FWA - First Word Address

function code - The name of an algorithm contained in the system software.

global - A value defined in one program module and used in others. Globals are often referred to as entry points in the module in which they are defined, and externals in the other modules which use them.

hexadecimal - Pertaining to the Base 16 number system.

interrupt - A signal which, when activated, causes a transition of control to a specific location in memory, thereby breaking the normal flow of control of the routine being executed. An interrupt is normally caused by an external event such as a "done" condition in a peripheral. It is distinguished from a trap which is caused by the execution of a processor instruction.

Julian day - Yearly day count 1 to 365.

MCR (Monitor Control Routine) - The executive routine that allows the user to communicate with the system using an on-line terminal device. MCR accepts and interprets commands typed on the terminal keyboard and calls appropriate routines to execute the specified requests.

msec - milliseconds, \((10^{-3})\) seconds.

mux - (multiplexing) Use of one communications line circuit for two or more simultaneous data paths.

parity - When a binary bit is appended to a group of bits to make the sum of all the bits always odd (ODD Parity) or always even (EVEN Parity).

purge - to delete

SPS - Samples per Second

TPERR - Trap Error flag

trap - The central processor immediately going to a fixed memory location because of an error or programming condition.

TTY - teletype

XXXg- Representation of an Octal number.
TYPICAL 24 HOUR DATA DUMP

DATA TAKEN 29TH OF OCTOBER 1981

FROM THE MODOA @ CLAYTON, N.M.
<table>
<thead>
<tr>
<th>6 MIN AVG IN HEX</th>
<th>CH1 POWER KW</th>
<th>CH2 WIND SPDM MPH</th>
<th>CH3 WIND SPDIN MPH</th>
<th>CH4 NAC,DIR DEG</th>
<th>CH5 YAW ERROR DEG</th>
<th>CH6 BLADE ANGLE DEG</th>
<th>CH7 KVARS ON/OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>7F</td>
<td>-300</td>
<td>0.0</td>
<td>50.0</td>
<td>0</td>
<td>+90</td>
<td>+10</td>
<td>-300</td>
</tr>
<tr>
<td>79</td>
<td>-270</td>
<td>2.5</td>
<td>47.5</td>
<td>18</td>
<td>+81</td>
<td>+5</td>
<td>-270</td>
</tr>
<tr>
<td>73</td>
<td>-240</td>
<td>5.0</td>
<td>45.0</td>
<td>36</td>
<td>+72</td>
<td>0</td>
<td>-240</td>
</tr>
<tr>
<td>6C</td>
<td>-210</td>
<td>7.5</td>
<td>42.5</td>
<td>54</td>
<td>+63</td>
<td>-5</td>
<td>-210</td>
</tr>
<tr>
<td>66</td>
<td>-180</td>
<td>10.0</td>
<td>40.0</td>
<td>72</td>
<td>+54</td>
<td>-10</td>
<td>-180</td>
</tr>
<tr>
<td>60</td>
<td>-150</td>
<td>12.5</td>
<td>37.5</td>
<td>90</td>
<td>+45</td>
<td>-15</td>
<td>-150</td>
</tr>
<tr>
<td>59</td>
<td>-120</td>
<td>15.0</td>
<td>35.0</td>
<td>108</td>
<td>+36</td>
<td>-20</td>
<td>-120</td>
</tr>
<tr>
<td>53</td>
<td>-90</td>
<td>17.5</td>
<td>32.5</td>
<td>126</td>
<td>+27</td>
<td>-25</td>
<td>-90</td>
</tr>
<tr>
<td>4C</td>
<td>-60</td>
<td>20.0</td>
<td>30.0</td>
<td>144</td>
<td>+18</td>
<td>-30</td>
<td>-60</td>
</tr>
<tr>
<td>46</td>
<td>-30</td>
<td>22.5</td>
<td>27.5</td>
<td>162</td>
<td>+9.0</td>
<td>-35</td>
<td>-30</td>
</tr>
<tr>
<td>3F</td>
<td>00</td>
<td>25.0</td>
<td>25.0</td>
<td>180</td>
<td>0.0</td>
<td>-40</td>
<td>0.0 (OFF)</td>
</tr>
<tr>
<td>39</td>
<td>+30</td>
<td>27.5</td>
<td>22.5</td>
<td>198</td>
<td>-9.0</td>
<td>-45</td>
<td>+30</td>
</tr>
<tr>
<td>33</td>
<td>+60</td>
<td>30.0</td>
<td>20.0</td>
<td>216</td>
<td>-18</td>
<td>-50</td>
<td>+60</td>
</tr>
<tr>
<td>2C</td>
<td>+90</td>
<td>32.5</td>
<td>17.5</td>
<td>234</td>
<td>-27</td>
<td>-55</td>
<td>+90</td>
</tr>
<tr>
<td>26</td>
<td>+120</td>
<td>35.0</td>
<td>15.0</td>
<td>252</td>
<td>-36</td>
<td>-60</td>
<td>+120</td>
</tr>
<tr>
<td>1F</td>
<td>+150</td>
<td>37.5</td>
<td>12.5</td>
<td>270</td>
<td>-45</td>
<td>-65</td>
<td>+150</td>
</tr>
<tr>
<td>19</td>
<td>+180</td>
<td>40.0</td>
<td>10.0</td>
<td>288</td>
<td>-54</td>
<td>-70</td>
<td>+180</td>
</tr>
<tr>
<td>13</td>
<td>+210</td>
<td>42.5</td>
<td>7.5</td>
<td>306</td>
<td>-63</td>
<td>-75</td>
<td>+210</td>
</tr>
<tr>
<td>0C</td>
<td>+240</td>
<td>45.0</td>
<td>5.0</td>
<td>324</td>
<td>-72</td>
<td>-80</td>
<td>+240</td>
</tr>
<tr>
<td>06</td>
<td>+270</td>
<td>47.5</td>
<td>2.5</td>
<td>342</td>
<td>-81</td>
<td>-85</td>
<td>+270</td>
</tr>
<tr>
<td>00</td>
<td>+300</td>
<td>50.0</td>
<td>0.0</td>
<td>360</td>
<td>-90</td>
<td>-90</td>
<td>+300</td>
</tr>
</tbody>
</table>