COMMUNICATION SOFTWARE FOR
THE PDP11/45 MINI COMPUTER

AMAL AWAD

A Technical Report
in
The Department
of
Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Engineering at
Concordia University
Montreal, Québec, Canada

March, 1979
© Amal Awad, 1979
ABSTRACT

COMMUNICATION SOFTWARE FOR THE PDP11/45

MINI COMPUTER

AMAL AWAD

The objective of this project is to develop communication software for the PDP11/45 Mini Computer. To permit the PDP11/45 to communicate with a remote terminal, remote PDP11, or with a full size computer such as the CDC-6000 Computet series.

This communication will be via data sets and private or public switched telephone facilities. This is accomplished by programming the Serial Line Interface DL11-W which is a character-buffered communication interface designed to assemble or disassemble the serial information required by the communication device for parallel transfer to or from the PDP11/45 Unibus. The interface provides the user with a choice of line speeds, character size, stop-code length, parity selection, line control function, and status indication.

The DL11-W Serial Line Interface flexibility provide complete data set control.
## CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>HARDWARE DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Scope</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>General Description</td>
<td>2</td>
</tr>
<tr>
<td>2.3</td>
<td>Functional Description</td>
<td>3</td>
</tr>
<tr>
<td>2.4</td>
<td>Configuration</td>
<td>4</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Baud Rate</td>
<td>4</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Address and Vectors Selections</td>
<td>4</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Data Format</td>
<td>6</td>
</tr>
<tr>
<td>2.5</td>
<td>Specification</td>
<td>6</td>
</tr>
<tr>
<td>2.6</td>
<td>Detailed Description</td>
<td>12</td>
</tr>
<tr>
<td>2.7</td>
<td>Installation and Connection</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>SOFTWARE CONFIGURATION OF THE SERIAL INTERFACE</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Scope</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Device Registers</td>
<td>21</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Interrupt</td>
<td>26</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Timing Consideration</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>THE COMMUNICATION PROGRAM DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Scope</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Program Description</td>
<td>28</td>
</tr>
</tbody>
</table>
## CONTENTS (Cont)

<table>
<thead>
<tr>
<th>Chapter / Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Procedure Information</td>
<td>30</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>Program Flow-Char</td>
<td>35</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>Program Listing</td>
<td>37</td>
</tr>
</tbody>
</table>

### TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>DL11-W Baud Rates</td>
<td>7</td>
</tr>
<tr>
<td>2-2</td>
<td>DL11-W Operating Specification</td>
<td>8</td>
</tr>
<tr>
<td>2-3</td>
<td>DL11-W Functional Units</td>
<td>12</td>
</tr>
<tr>
<td>2-4</td>
<td>Data Format Switches</td>
<td>10</td>
</tr>
<tr>
<td>2-5</td>
<td>Address Selection Logic Output Signals</td>
<td>17</td>
</tr>
<tr>
<td>2-6</td>
<td>Pin Connection</td>
<td>19</td>
</tr>
<tr>
<td>3-1</td>
<td>DL11-W Register Assignment</td>
<td>21</td>
</tr>
</tbody>
</table>

### FIGURES

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>DL11-W Block Diagram</td>
<td>5</td>
</tr>
<tr>
<td>2-2</td>
<td>DL11-W Data Format</td>
<td>7</td>
</tr>
<tr>
<td>2-3</td>
<td>Address Selection Logic</td>
<td>16</td>
</tr>
<tr>
<td>2-4</td>
<td>Address Format</td>
<td>11</td>
</tr>
<tr>
<td>2-5</td>
<td>DL11-W Modem Cable Connection Diagram</td>
<td>18</td>
</tr>
</tbody>
</table>
INTRODUCTION

This report provides a complete description of the hardware and software components used to interface the PDP11/45 Mini Computer with a remote computer via telephone lines facilities.

The report presents: The general description, operating specification, detailed description, installation and connection, programming information, hardware and software configuration of the DLI1-W Serial Line Interface.

A detailed description of the communication program features, program listing and flow-chart, a detailed procedure of a communication session with the compass computer are provided.

The report presents tables and illustrations to support the description of the DLI1-W Unit.

The report is divided into four major chapters: Introduction, hardware configuration, software configuration, and the communication program description.
2.1 Scope
This chapter presents the functional description and operating specification of the DLll-W interface unit. This information is the basic tool in understanding and analyzing the communication program described in chapter 3.

2.2 General Description
The DLll-W Serial Line Unit is a character-buffered communication interface designed to assemble or disassemble the serial information required by the modem for parallel transfer to or from the PDP-ll/unibus.

The DLll-W consists of a single integrated circuit quad board containing two independent communication units (Receiver and Transmitter) that are capable of simultaneous 2-way communication.

The DLll-W interface provide the logic and buffer registers necessary for program-controlled transfer of data between the PDP-ll system requiring parallel data and an external device requiring serial data. The interface also provide status and control bits that may be controlled by the program, the interface, or the modem for command, monitoring and interrupt functions.

The DLll-W provides the user with the choice of line speeds, baud rates, character size, stop-code length, line control functions, and status indicators.

The DLll-W has ETA level converter to change bipolar serial input data to TTL logic levels and TTL logic level serial outputs to the bipolar signals required by the modem.
2.3 Functional Description

DL11-W Modem Interface: The following discussion is supported by the DL11-W block diagram, Figure 2-1.

Serial information received or transmitted by the modem is assembled or disassembled by the DL11-W interface for parallel transfer to or from the PDP-11 UNIBUS. When the processor puts an address on the bus, the DL11-W decodes the address to determine the selected external device, and whether it is to perform an input or output operation.

The Transmitter Function:

It performs parallel-to-serial conversion of 5, 6, 7, or 8 level codes.

Data from the UNIBUS is loaded into parallel into the DL11 holding register. When the transmitter shift register is empty, the content of the holding register is shifted into the transmitter shift register and the XMIT RDY flag comes up. A second character from the bus can then be loaded into the holding register. However, because the shift register is still working on previous data, the shifting operation of the second character is delayed until previous character has been completely transmitted. Once the last bit of the character is transmitted to the modem, the interface indicates an interrupt request (XMIT RDY) to indicate that the buffer is empty and can now be loaded with another character for transfer to the modem.

The Receiver Function:

When data is received from the modem, the START bit of the serial data activates the interface receiver logic and data is loaded one bit at a time into the receiver buffer register. When buffer loading is complete, the buffer content is transferred to the holding register.
The interface sets the (RCVR_DONE) flag indicating to the program that a character has been assembled and is ready for transfer to the bus. If (RCVR INT ENB) is also set, the (RCVR DONE) flag initiates an interrupt sequence, thereby causing a vectored interrupt.

2.4 Configurations

The DL11-W consists of an M7865 quad module with five dip mounted switch packs. Each pack contains either eight or ten individual toggle switches. The packs are labeled S1 through S5 on the board; each switch on the pack numbered 1 through 8 or 10. Positions for On and Off are clearly indicated on the hardware.

Switch selection on the DL11-W interface provides the flexibility needed to handle a variety of functions. The user has a choice of speeds, character size, stop code length, parity, error detection, 20 mA current, loop or EIA, addresses, and vectors, and active or passive modes.

2.4.1 Baud Rates

Table 2-1 lists the eight different baud rates available on the DL11-W interface.

In our configuration the baud rate is selected to be 300 for both transmitter and receiver. For compatibility with the Bell Modem.

2.4.2 Addresses and vectors selection

Refer in the following discussion to the simplified diagram of the address selection logic, figure 2-3.

The switches on the logic can be altered so that the module responds to any address within the range of 774000 to 777777. However, standard address assignment for the DL11-W normally fall within the ranges of 775610 to 776177 or 776500 to 776677.
FIGURE 2.1 DL11-W BLOCK DIAGRAM

ICL = INTERRUPT CONTROL LOGIC
ASL = ADDRESS SELECTION LOGIC
RCSR = RECEIVER STATUS
XCSR = TRANSMITTER STATUS
RBUF = RECEIVER BUFFER
XBUF = TRANSMITTER BUFFER
MML = MAINT. MODE LOOP
The Interrupt Vector is determined by the interrupt control logic. Each
DL11-W within the system has a unique address and a unique vector. These
are determined by the switches on the module.

In our case, the addresses reserved for the DL11-W in the memory space
of the PDP-11/45 covers the area from 176560 to 176566 in octal base.
The VECTORS addresses are 360 for the RECEIVER and 364 for the
TRANSMITTER.

Switch S5 control the address selection. It indicates logic one when
turned off. For vector selection, on the other hand, switch S2 control
the VECTORS addresses, and it indicates logic one when it is on.

2.4.3 Data Format

The data format consists of a START bit, five to eight DATA bits, a
PARITY bit or no PARITY bit, and one, one and one half, or two STOP bits.
When less than eight DATA bits are selected the hardware justifies the
bits into the least significant bit positions for characters received
by the interface when transmitting characters. The program provides the
 justification into the least significant bits.

All variable items within any data format are selected by switches on
the DL11-W module. None of the variable can be controlled by the program.

These switches are listed in table 2-4.

The DL11-W data format is shown in figure 2-2.

The data format configuration for the communication program are as
follows: one START bit, one STOP bit, 8-bit data and no parity.

2.5 SPECIFICATIONS

Operating and physical specifications for the DL11-W Serial Line Unit
are given in table 2-2.
### Table 2-1 DL1-W Baud Rates

<table>
<thead>
<tr>
<th>Baud Rate</th>
<th>Transmit</th>
<th>Receive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S4-10</td>
<td>S3-1</td>
</tr>
<tr>
<td>110</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>150</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>300</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>600</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>1200</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>2400</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>4800</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>9600</td>
<td>OFF</td>
<td>ON</td>
</tr>
</tbody>
</table>

### Figure 2-2 DL1-W Data Format

- **Idle State of Line**: 5 to 8 data bits, odd/even, return to unused or idle state.
- **Start**: one bit time = $\frac{1}{\text{BAUD RATE}}$.
- **New Char.**: Start bit, 1.5 units, 2 units.

---

**Note**: The table and figure provide a detailed description of the baud rates and data format for the DL1-W interface, highlighting the specific values and the timing for transmitting and receiving data.
<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registers</td>
<td>Receiver Status Register (RCSR)</td>
</tr>
<tr>
<td></td>
<td>Receiver Buffer Register (RBUF)</td>
</tr>
<tr>
<td></td>
<td>Transmitter Status Register (XCSR)</td>
</tr>
<tr>
<td></td>
<td>Transmitter Buffer Register (XBUF)</td>
</tr>
<tr>
<td>Register Addresses</td>
<td>RCSR  176560</td>
</tr>
<tr>
<td></td>
<td>RBUF  176562</td>
</tr>
<tr>
<td></td>
<td>XCSR  176564</td>
</tr>
<tr>
<td></td>
<td>XBUF  176566</td>
</tr>
<tr>
<td>Interrupt Vector Address</td>
<td>360 Receiver</td>
</tr>
<tr>
<td></td>
<td>364 Transmitter</td>
</tr>
<tr>
<td>Priority Level</td>
<td>BR4 Serial Line Unit</td>
</tr>
<tr>
<td>Interrupt Type</td>
<td>Transmitter Ready (XMIT RDY)</td>
</tr>
<tr>
<td></td>
<td>Receiver Done (RCVR DONE)</td>
</tr>
<tr>
<td>Commands</td>
<td>Receiver Interrupt Enable (RCVR INT ENB)</td>
</tr>
<tr>
<td></td>
<td>Transmitter Interrupt Enable (XMIT INT ENB)</td>
</tr>
<tr>
<td></td>
<td>Maintenance Mode (MAINT)</td>
</tr>
<tr>
<td></td>
<td>Break (BREAK)</td>
</tr>
<tr>
<td>Status Indicators</td>
<td>Receiver Active (RCVR.ACT)</td>
</tr>
<tr>
<td></td>
<td>Transmitter Ready (XMIT RDY)</td>
</tr>
<tr>
<td></td>
<td>Receiver Done (RCVR DONE)</td>
</tr>
<tr>
<td></td>
<td>Error (ERROR)</td>
</tr>
<tr>
<td></td>
<td>Overrun (OR ERR)</td>
</tr>
<tr>
<td></td>
<td>Framming Error (FR ERR)</td>
</tr>
</tbody>
</table>
Parity Error (P ERR)
Serial data, 20 mA active current loop
Serial data, 20 mA passive current loop
Serial data, conforms to EIA and CCITT
specifications.
One START bit; 5-, 6-, 7-, or 8-bit DATA
character; PARITY bit (odd, even, or unused);
1 or 2 STOP bits with 6, 7, 8 DATA bits
selected; 1 or 1.5 STOP bits with 5 DATA bit
selected.

Baud Rates
Baud rates may be 100, 150, 300, 600, 1200, 2400
4800, or 9600. Any split speed combination
possible (transmitter and receiver speeds
may differ).

Bit Transfer Order
Low-order bit (LSB) first

Parity
Computed on incoming data or inserted on
outgoing data, depending on type of parity
(odd or even) used.
Parity may be odd or even

Power Required
2.0 A at +5 V
150 mA at -15 V
50 mA at level between +9 V and +15 V

Temperature Range
10 to 50 °C
<table>
<thead>
<tr>
<th>Name</th>
<th>Switch</th>
<th>Pin No.</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Parity</td>
<td>S4-6</td>
<td>35</td>
<td>Enable or disable the parity bit in the data character. When enabled, the value of the parity bit is dependent on the type of parity (odd- or even) selected by the even parity select (S4-2) switch. When disabled, the STOP bits immediately follow the last DATA bit during transmission. During reception, the receiver does not check for parity. Switch ON - parity enabled Switch OFF - parity disabled</td>
</tr>
<tr>
<td>Even Parity</td>
<td>S4-2</td>
<td>39</td>
<td>Determine whether odd or even parity is to be used. The receiver checks the incoming character for appropriate parity; the transmitter inserts the appropriate parity value. Switch ON - odd parity Switch OFF - even parity</td>
</tr>
<tr>
<td>STOP Bit</td>
<td>S4-5</td>
<td>36</td>
<td>Selects the desired number of stop bits. Switch ON - One stop bit. Switch OFF - Two STOP bits, but if five DATA bits are selected, one and one half STOP bits will be selected.</td>
</tr>
</tbody>
</table>
These two switches are used together to provide a code that selects the desired number of DATA bits in the character.

<table>
<thead>
<tr>
<th>S4-4</th>
<th>S4-3</th>
<th>No. of DATA bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>ON</td>
<td>5</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>6</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>7</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 2-4 Interface Select Address Format

Decoded When Line Clock is Enabled..
2-6 Detailed Description

The complete DL11-W may be divided into 12 functional areas. Table 2-3 lists these areas and explains the general purpose of each.

Table 2-3 DL11-W Functional Units

<table>
<thead>
<tr>
<th>Functional Unit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection Logic</td>
<td>Determine if the interface has been selected and what type of operation (transmitter, or receiver) has been selected. Permit selection of one of four internal registers and determines if the register is to perform an input or output function.</td>
</tr>
<tr>
<td>Register Logic</td>
<td>Four internal registers, addressable by the program, provide data transfer, command and control, and status monitoring functions for the interface.</td>
</tr>
<tr>
<td>Interrupt Request Logic</td>
<td>The receiver or transmitter can request control of the Unibus for a vectored interrupt.</td>
</tr>
<tr>
<td>Interrupt Logic</td>
<td>Permits the DL11-W to gain control of the Unibus for a vectored interrupt.</td>
</tr>
<tr>
<td>Transmitter Control Logic</td>
<td>Provides necessary input control signals for the UART when it is used to convert parallel data from the Unibus to serial data required by the Modem.</td>
</tr>
</tbody>
</table>
Receiver Control Logic

Provides necessary input control signals for the UART when it is used to convert serial data to parallel data required for transmission to the bus.

Universal Asynchronous Receiver/Transmitter (UART)

Perform the necessary serial-to-parallel or parallel-to-serial on the data, and supplies control and error detecting bits.

Baud Rate Logic

Determine the clock frequencies and, therefore, the baud rate for transmitter and receiver sections system of the UART. Eight baud rates are derived from a single oscillator and are independently switch-selectable.

Maintenance Mode Logic

Perform a closed loop test of the serial line unit control logic by tying the serial output of the transmitter into the receiver input, forcing the receiver clock to the same frequency as the transmitter clock.

Break Generation Logic

Permit the transmission of continuous space or "break." The duration of break can be timed by the pseudo-transmission of specific number of character.

EIA Logic

Provides necessary level converters for use with EIA level devices.
2.6.1 Address Selection

The following description is supported by figure 2-3.

The address logic decodes the incoming address information from the bus and provides the signal that determines which register has been selected and whether it is to perform an input or output function. Jumpers on the logic can be altered so that the module responds to any address within the range of 775610 to 776177 or 776500 to 776677.

The first five octal digits of address indicate that the serial line unit has been selected. The final octal digits consist of the A02, A01, and A00, determine which register has been selected and whether a word or byte operation is to be performed. The two-mode control line C01 and C00 determine whether the selected register is to perform an input or an output operation.

The address decoding is performed by a series of logic gates inputs to two 32×8 Read Only Memory (ROM). Basically, the state of the five input lines define one of 32 unique addresses. The content of the ROM corresponding to that unique address is then available at the output of the ROM. Each ROM provides 8 outputs for a total of 16, although only 14 of the 16 are used.

2.6.1.1 Inputs

Refer to the simplified block diagram of the address selection logic shown in figure 2-3. Note that IN and OUT are used with respect to the master (controlling) device. Thus, when the DL11-W is used, an OUT transfer is transfer of data out of the master (the processor), and into the interface. Similarly, an IN transfer is the operation of the interface furnishing data to the processor.
The address selection lines consists of 18 address lines on the bus (A17-00), bus control lines C1 and C0, and a master synchronization (MSYN) line.

The address selection logic decodes the addresses on the bus.

The address format is shown in figure 2-4.

1. Address lines A17-A11 must be all 1s. This specifies an address in the top 4K addresses for device registers.

2. Decoding of address lines A10-05 and A3 is determined by switches. When a given line switch is ON, the address logic searches for a 0 on that line. If the switch is OFF, the logic searches for a 1. If only the serial line unit is to be enabled, then decoding of A04 will also be determined by a switch.

3. Lines A01, A02 and A04 are decoded to select one of the five addressable device registers.

4. Line C1 is used to select either an input or output function. When C1 is false, an input (read) operation is selected, when it is true an output (write or load) operation is selected.

5. Line A00 is used for byte control in such a manner that no register control signals are generated when a byte operation is performed on the high-order byte of any register.

2.6.1.2 OUTPUT

The address selection logic output signals are used to permit selection of four 16-bit registers, and determine whether information is to be gated into or out of the master device. All output signals are listed in table 2-5.
FIGURE 2.3 ADDRESS SELECTION LOGIC SIMPLIFIED DIAGRAM
<table>
<thead>
<tr>
<th>Function Selected</th>
<th>Bus Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus to transmitter buffer</td>
<td>DATO or DATOB</td>
</tr>
<tr>
<td>RBUF BUS H</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>RCSR CLK ENB L</td>
<td>DATO or DATOB</td>
</tr>
<tr>
<td>XCSR CLK ENB L</td>
<td>DATO or DATOB</td>
</tr>
<tr>
<td>SSYN EN L</td>
<td>DATO, DATOB, DATI</td>
</tr>
<tr>
<td>E14 EN L</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>Enables bus drivers 001, 003, 004 and 005</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E22SO H</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>Select either buffer(H) or transmitterstatus(L) to bus (bits 0 and 2)</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E22STB L</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>Enables bits 0 and 2 (above) to bus drivers</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E21 S1 H</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>Bits 6 and 7 of</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E22 SO H</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>receiver buffer(SO=1, S1=1)</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>receiver status(SO=1, S1=1)</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>transmitter status (SO=1, S1=1)</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E6 EN L</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>Enable bus drivers D00, D02 and D07</td>
<td>DATI or DATIP</td>
</tr>
<tr>
<td>E23. EN L</td>
<td>DATI or DATIPu</td>
</tr>
<tr>
<td>Receiver status (bit 11) to bus</td>
<td>DATI or DATIPu</td>
</tr>
</tbody>
</table>
2-7 INSTALLATION AND CONNECTION

2-7-1 Mounting

The DL11-W can be mounted in either a small peripheral controller slot in
the PDP-11/45 processor (DD11-C) or an SPC slot in a DD11-D or a
DD11-P in the backplanes.

2.6.2 Connection

Once the 17856 has been installed, an appropriate cable must be
connected as shown in Figure 2-5.

Figure 2-5 DL11-W Modem Connection Diagram

Table 2-6 lists connector pin numbers for the BC05C cable connector.
<table>
<thead>
<tr>
<th>Berg Pin</th>
<th>M7856 Module</th>
<th>BCO5C Modem Cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>B</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>Force Busy</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>Secondary Clear to Send</td>
</tr>
<tr>
<td>E</td>
<td>Serial Input(TTL)</td>
<td>Interlock In</td>
</tr>
<tr>
<td>F</td>
<td>Serial Output((EIA))</td>
<td>Transmitted Data</td>
</tr>
<tr>
<td>H</td>
<td>20 mA Interlock</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>Serial Input (EIA)</td>
<td>Received Data</td>
</tr>
<tr>
<td>K</td>
<td>+Serial Input(20mA)</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td>External Clock</td>
</tr>
<tr>
<td>M</td>
<td>EIA Interlock</td>
<td>Interlock Out</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>Serial Clock Xmit</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td>Secondary Request to Send</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>Serial Clock Receiver</td>
</tr>
<tr>
<td>S</td>
<td>-Serial Input (20mA)</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>Clear to Send</td>
</tr>
<tr>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Request to Send(EIA)</td>
<td>Request to Send</td>
</tr>
<tr>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>Ring</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>+Power</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>AA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>Carrier</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DD</td>
<td>Data Terminal Ready (EIA)</td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td>-Reader Run (20mA)</td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KK</td>
<td>-Serial Output (20mA)</td>
<td></td>
</tr>
<tr>
<td>LL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PP</td>
<td>+Reader Run (20mA)</td>
<td></td>
</tr>
<tr>
<td>RR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TT</td>
<td>+5 V</td>
<td></td>
</tr>
<tr>
<td>UU</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>VV</td>
<td>Ground</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
CHAPTER 3
SOFTWARE CONFIGURATION
OF THE DL11-W INTERFACE

3.1 SCOPE
This chapter presents general programming information for software
control of the DL11-W interface.

3.2 Programming Information
Programming of the DL11-W is controlled by device registers, interrupt
and timing considerations.

3.2.1 Device Registers
There are four device registers. These registers have been assigned bus
addresses and can be read or loaded using any PDP-11/45 instruction
which refer to their addresses.

The four device registers and associated bus address are listed in
table 3-1.

<table>
<thead>
<tr>
<th>Register</th>
<th>Mnemonic</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Status</td>
<td>RCSR</td>
<td>176560</td>
</tr>
<tr>
<td>Receiver Buffer</td>
<td>RBUF</td>
<td>176562</td>
</tr>
<tr>
<td>Transmitter Status</td>
<td>XCSR</td>
<td>176564</td>
</tr>
<tr>
<td>Transmitter Buffer</td>
<td>XBUF</td>
<td>176566</td>
</tr>
</tbody>
</table>

Table 3-1 DL11-W Register Assignment
for Interface With BELL Modem 103
The bit assignments for device registers are shown below. The unused and write-only bits are always read as 0s. Writing unused or read-only bits has no effect on bit position but is not considered good programming practice. The mnemonic INIT refers to the initialization signal issued by the processor. Initialization is caused by one of the following: issuing a programmed RESET instruction, pressing the START switch on the processor console, or the occurrence of a power-up or power-down condition on the processor power supply.

**Receiver Status Register Bit Format**

<table>
<thead>
<tr>
<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>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT USED</td>
<td>NOT USED</td>
<td>NOT USED</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>

RCVR        RCVR        RCVR        RDR
ACT         DONE        INT           ENB

**Bit Meaning and Operation**

15-12. Unused

11 Receive Active-Read-only. When set this bit indicates that the receiver interface is active. This bit is set at the center of the start bit, which is the beginning of the input serial data from the Modem, and cleared by the leading edge of Receiver Done. Also may be cleared by INIT.

10-8 Unused

7 Receiver Done-read-only. Set when an entire character has been received and is ready for transfer to the Unibus. Cleared by addressing (read or write) RBUF, or INIT.
Starts an interrupt sequence (bit 6) is also set.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning and Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Error-Read-only. Logical OR of Overrun, Framing Error, and Parity Error. Cleared by removing the error conditions. Error is not tied to the interrupt logic.</td>
</tr>
<tr>
<td>14</td>
<td>Overrun-Read-only. Set if previously received character is not read (Receiver Done is not reset) before the present character is received.</td>
</tr>
<tr>
<td>13</td>
<td>Framing Error-Read-only. Set if the character read has no valid STOP bit. Also used to detect Break.</td>
</tr>
<tr>
<td>12</td>
<td>Receive parity error-Read-only. Set if received parity does not agree with the expected parity. Always zero if no parity is selected.</td>
</tr>
<tr>
<td>11-8</td>
<td>Unused</td>
</tr>
<tr>
<td>7-0</td>
<td>Received data bits-Read-only. These bits contain the character just read. If less than 8 bits are selected, the data will be right-justified into the least significant bits.</td>
</tr>
</tbody>
</table>
Transmitter Status Register Bit Format

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning and Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-8</td>
<td>Unused</td>
</tr>
<tr>
<td>7</td>
<td>Transmitter Ready—Read-only. Set by INIT. Cleared when XBUF can accept another character. When set it will start an interrupt sequence if transmitter interrupt enable is also set.</td>
</tr>
<tr>
<td>6</td>
<td>Transmitter Interrupt Enable—Read/Write. Cleared by INIT.</td>
</tr>
</tbody>
</table>
When set it will start an interrupt sequence if Transmitter Ready is also set.

Unused

Maintenance-Read/write. Cleared by INIT. When set, it disables the serial line unit to the receiver and send the serial output of the transmitter into the serial input of the receiver. Forces receiver to run at transmitter speed.

Unused

Break-Read/write. Cleared by INIT. When set, it transmits a continuous space. May be disabled via a switch.

Transmitter Data Buffer Bit Format

15 8 7 0

NOT USED Transmitter Data Buffer

Bit Meaning and Operation

15-8 Unused

7-0 Transmitter Data Buffer-Write-only. If less than eight bits are selected, the character must be right-justified into the least significant bits.
3.2.2 Interrupts

The DLLL-W interface uses BR interrupt to gain control of the bus to perform vectored interrupt, thereby causing transfer of control to a handling routine. The routines control data transfer to and from the interface.

The DLLL-W has two independent channels; one for receiver section, the other for the transmitter section. These two channels operate independently. However, if simultaneous interrupt request occurs the receiver has higher priority than transmitter.

A transmitter interrupt can occur only if the interrupt enable (XMIT INT ENB) bit in the transmitter status register is set, setting the transmitter ready (XMIT RDY) bit initiates an interrupt request. When (XMIT RDY) is set, it indicates that the transmitter buffer is empty and ready to accept another character from the bus for transfer to the modem.

A receiver interrupt can occur only if the interrupt enable (RCVR INT ENB) bit in the receiver status register is set. Setting the receiver done (RCVR DONE) bit initiates an interrupt request. When (RCVR DONE) is set, it indicates that an entire character has been received and ready for transfer by the bus.

The interrupt priority level is 4 for the receiver and transmitter. The vector address assigned for the receiver and transmitter are 360, 364 respectively.

The vector addresses can be changed by resetting switches in the interrupt control logic.

3.2.3 Timing Considerations
When programming the DLI-1-W interface it is important to consider the timing of certain functions to be able to use the system in most efficient manner. Timing consideration for the receiver, transmitter and break generation logic are discussed in the following paragraphs.

3.2.3.1 Receiver
The (RCVR DONE) flag (bit 7 in RCSR) sets when the Universal Asynchronous Receiver/Transmitter (UART) has assembled a full character. This occur at the middle of the first STOP bit, because the UART is double-buffered, data remain valid until the next character is received and assembled. This permits one full character time for servicing the (RCVR DONE) flag.

3.2.3.2 Transmitter
The transmitter section of the UART is also double-buffered. The (XMIT RDY) flag (bit 7 of XCSR) is set after initialization. When the buffer (XBUF) is loaded with the first character from the bus, the flag clears but then sets again within a fraction of a bit time. A second character can then be loaded which clears the flag again. The flag then remain clear for nearly one full character time.

3.2.3.4 Break Generation Logic
When the BREAK bit (bit 0 in the XCSR) is set, it causes transmission of a contiguous space. Because the (XMIT RDY) continue to function normally, the duration of a break can be timed by the pseudo-transmission of a number of characters. However, because the transmitter section of the UART is double-buffered, a null character (all 0s) should precede transmission of the break to insure that all previous characters clear the line. In a similar manner, the final pseudo-transmitted character in the break should be null.
CHAPTER 4
THE COMMUNICATION PROGRAM

DESCRIPTION

4.1 Scope
This chapter describes in detail the program structure, the program
main features, and the complete procedure on how to use the program
to communicate with the compass computer (CDC-6000 Series).
The program flowchart and listing are presented in appendices A and B
respectively.

4.2 Program Description
The program handles data transfer between the PDP11/45 Mini-Computer
and the DL11-W Serial Line Unit configured to interface Bell modem.
The transfer is done asynchronously and on vectored interrupt bases.
The program is independent of the PDP11/45 operating system, so it can
be used with most PDP-11 Computer Series.
The program design is based on modular program development. The program
is built from the main routine and sub-programs. The main routine is
written in Fortran language, the subroutines are in PDP11/45 Assembly
language. The advantages of such an approach are fast program
devolution, the facility of using different source languages,
whichever suits the task best, easy debugging; and fast program
modification. The program code is re-locatable. It can be loaded anywhere in
memory as required. This feature is exceedingly valuable for
those utilities as the disk-resident routines which are subjected to
loading in a dynamically changing program environment.
The program initialization is a table driven utilizing the auto-increment mode of addressing. This mode provides for automatic stepping of a pointer through sequential elements of a table of operands. It assumes the content of the selected general register to be the address of the operand. Contents of register is stepped to address the next sequential location.

This mode of addressing facilitates the processing of tabular data.

The program structure is based on subroutine linkage as explained before. Now let us explain with more detail the argument transmission of the call subroutine instruction in the PDP11/45.

The memory location pointed to by the linkage register of the JSR (jmp subroutine) instruction may contain argument or addresses of arguments. These arguments may be accessed from the subroutine in several ways.

Using the register R5 as the linkage register, the first argument could be obtained by using the addressing modes indicated by (R5), (R5)+, X(R5) for actual data. The auto-increment mode could be used so that the linkage register is automatically updated to point to the next argument.

Some of the PDP11 subroutine advantage:

a-Argument can be quickly passed between the calling program and the subroutine.

b-Many subroutine calls can be executed without the need to provide any saving procedure for the linkage information. Since the linkage information is automatically pushed onto the stack in sequential order, return can simply made by automatically popping this information from the stack in the opposite order.
4.3 Procedure Information

This paragraph describes in detail how to execute a program on the
compass CDC computer with the PDP11/45 as a terminal using the
developed program.

Starting with the PDP11/45 system bootstrap, program assembly, linking
and executing.

A brief note about time-sharing on the CDC-6000 series is also provided.

System bootstrapping:

Switch on the master power switch.

Mount the RT-11 disk, which has the communication program file, on
drive number 0.

Set the Load/Run switch to Run. Wait until the Ready lamp is on.

Set Enable/Halt switch to Halt position

Set the address 773010 in the switch register.

Depress the Load Address switch.

Set the Enable/Halt switch to Enable.

Depress the Start switch.

On the system terminal type:

GT ON followed by carriage return.

The monitor will prompt with (.) and it is now ready to receive
commands.

Program Assembly:

Call the Macro software as follows:

R MACRO (CR)

* COM.OBJ = COM.MAC (CR)

To compile the main fortran program call the fortran software as
follows;
.R FORTRAN (CR)
* COMM.IN=COMM.FOR (CR)

Program Linking:
To link the compiled Fortran program with the assembled Macro routines and the Fortran Library, call the linker software
.R link (CR)
*COMM.TST=COMM.IN,COM.OBJ,SYSLIB/F (CR)

Program Execution:
To run the program switch to the monitor mode by typing CNTR/C key. Type the following;
.R COMM (CR)

SET the 3-position switch located on the video terminal is at CDC position.

Call the CDC computer by dialling the number 7366 from the telephone set connected to the modem. If the call is successful, a continuous buzzer will be heard.

Set the Talk/Data switch on the modem to the Data position.

Type the numeric character ONE on the PDP-11 terminal followed by a carriage return. The CDC responds as follows;

CONCORDIA UNIVERSITY - CYPER 172/2 MOS 1.2-419

USER NUMBER:
Type the user account number followed by carriage return. The machine responds;

PASSWORD:
To which you must reply by typing Your PASSWORD. The machine then type

TERMINAL : 10,TTY
RECOVER / SYSTEM :
Your are now loged on. to enter the CDC command mode type

BATCH (CR)

To create a new file type.

NEW, Fname (CR)

To initiate TEXT entering mode type

TEXT (CR)

The machine will replay with the message

ENTER TEXT MODE

Enter the text of the file line by line each followed by a carriage return.

To exit the TEXT mode strike the (@) key followed by a carriage return.

The machine will respond

EXIT TEXT MODE

To save the text as a permanent file type

REWIND,Fname (CR)

PACK,Fname (CR)

After this the file is saved in the normal fashion.

-Compiling and Assembling

Let us assume we dealing with CFORT compiler. To activate the compiler type the command

CFORT,I = Fname

which instructs the CFORT compiler to read the source file from file 'fname' rather than from file INPUT.

To specify that the compiler is to write a listing to a local file called "Listing". This local file may be saved for subsequent printing.

CFORT,I=Fname,l=Listing

If no listing is desired at all we may type
CFORT, I=FileName, L=0

This suppresses the listing entirely, except that if any compilation errors occur in the program, the errors detected will be printed on file OUTPUT.

The compilation step creates another local file. The file contains the machine-language of the program being compiled. The name of this local file is "LGO". To execute this program type LGO.

We may save the local file and subsequently get it and execute it in another session. We may instruct the compiler to write a machine-language file of different name than LGO by indicating further parameter as follows:

CFORT, I=FileName, L=Listing, B=bfile

Which creates a local file whose name is symbolized by "bfile".

This file may be saved with SAVE, bfile

The parameters have the same meaning whether we are using CFORT compiler, FTN, COBOL, or the Compass Assembler.

Note: One of the most used text editor is NTXED (also known as TXED). It can be used to add or delete lines from the local text files, or to change lines in local files.

How to use NTXED? To edit a local file "Fname" first issue a command from BATCH:

+NTXED, Fname

The computer will echo the character string "?" indicating to the user that it is ready to receive further text editing instructions.

NTXED has an internal pointer which at a given time during an editing-
Indicates one of the line in the text file. The most common of the
positioning instruction to the editor are listed below:

F
Move to the next line (forward)

T
Move to the first line (top of the file)

B
Move to the last line (bottom of the file)

U
Move to the previous line (up)

A given line in the text file may be deleted from the file by first
locating the internal pointer to it and then issuing the command "D".
To add a new lines to the a text file the internal pointer must be
first set to the line after which the new line must be inserted. Then
issue the instruction "I".

You can always tell where you are in TINPUT mode, since instead of
printing "?", which the editor does to request an editing command, the
string ">~" is printed when in TINPUT mode.

Once all the desired changes has been made to the text file the
command

/ E
Causes the termination of the editor session and save all the changes
made to the local file in that local file.

If it is discovered that there has been a big mistake made in the
current editing session then that session may be terminated without
having any changes made in the editing session being made to the local
file. This is effectively an abandonment of the session, the instruct to
be given to abandon the session is

Q
The file in question will be as it was before the text editor was
initiated.
CALL TEST (FLAG)

FLAG

= 0

CALL SEND (OCHAR)

A

B

Communication Program Flow Chart
(Cont'd.)
MICRO COMMUNICATION ROUTINE

TABLE Driven, DINAMICALLY RE-ALLOCATABLE

.TITLE COMM

.NLIST TTM

.GLOBAL, START, BREAK, RECD, TEST, SEND, ITYOUT, ITYIN

.MCAll .REGDEF,.EXIT,.TTYOUT,.TTINR

.REGDEF

JSW = 44
SWR = 177570

DL-11 INTERFACED DIRECTLY TO MODEM ASYNC. UART

MCSRI = 176560 ; RECEIVER STATUS REG.
MDATI = MCSRI+2 ; RECEIVER BUFFER REG.
MCSRO = MCSRI+4 ; TRANSMITTER STATUS REG.
MDATO = MCSRI+6 ; TRANSMITTER BUFFER REG.
MVRCI = 360 ; RECEIVER VECTOR
MVRCO = MVRCI+4 ; TRANSMITTER VECTOR

KBBUF = 177562 ; KEYBOARD BUFFER REG.

INSET = 2
INABLE = 102
ENSET = 2
ENABLE = 100
PRI4 = 200

; CR = 15
DEL = 177
PAR = 200.

START:
CLR RBUF ; CLEAR TEMPORARY BUFFERS
CLR SBUF

MOV #VTABLE,R1

INIT:
MOV (R1)+, @(R1)+ ; INITIALIZE
BNE INIT

RETURN:
BIC #10100,@JSW ; CLEAR SPECIAL MODE BIT, CONSOLE I/O
CLC ; CLEAR CARRY BIT
RTS PC ; RETURN, CHECK, & RE-CALL

RECD:

BIC #PAR,RBUF ; CLEAR PARITY
CMPB #DEL,RBUF ; RUBOUT?
BNE 18

MOV #1,RBUF ; YES, SUBSTITUTE 1

18:

MOV RBUF,@2(R5) ;RECD FROM MODEM

CLR RBUF
BR RETURN

TEST:

MOV SBUF,@2(R5) ;SBUF CLEAR TO SEND?
BR RETURN

SEND:

MOV @2(R5),SBUF ;SEND TO MODEM

SEND1:

MOV RCSRO-2,@RCSRO ;ENABLE OUTPUT INTERRUPT
BR RETURN

ITYOUT:

TTYOUT @2(R5) ;TYPE THE RECD DATA FROM MODEM
BR RETURN

ITYIN:

CLR @2(R5)
BIS #10100.@JSW

B-3
BREAK:

BIS #1,RCSRO-2
MOV #10,R1

LOOP:

JSR PC,SEND1
SOB R1,LOOP
BIC #1,RCSRO-2
CLR SBUF
BR RETURN

VTABLE:

MODEM’S DL-11 I/O VECTORS

.WORD MINTI
RVECI: .WORD MVECI
.WORD PRI4
.WORD MVECI+2
RVECO:
.WORD

B-4
.WORD PRI4
.WORD MVECI+2
RVECO: .WORD MINTO
.WORD MVECO
.WORD PRI4
.WORD MVECO+2

; MODEMS I/O CSR REGISTERS

.WORD INABLE
RCSRI: .WORD MCSRI
.WORD INSET
RDATI: .WORD MDATI

; .WORD ENABLE
RCSRO: .WORD MCSRO
.WORD CR
RDATAO: .WORD MDATO

ETABLE:
.WORD 0
.WORD 0

; INTERRUPT ROUTINES

INPUT INTERRUPT TO MODEM TO DL-11W

B-5

MINTI:

MOV @RDATI,RBUF ; MODEM TO USER

MOV RCSRI-2,@RCSRRI ; RE-ENABLE INPUT INTERRUPT

RTI

OUTPUT INTERRUPT TO MODEM FROM DL11-W FROM USERS INPUT

MINTO:

CLR @RCSRO ; DIABLE OUTPUT INTERRUPT

MOVB SBUF,@RDATO ; USER BYTE OUTPUT TO MODEM

CLR SBUF

RTI

IMMEDIATE BUFFERS & COUNTER

RBUF: .WORD 0
SBUF: .WORD 0

.TEAM 0
.TEAM 0

TEN: .WORD 32000

-END START
THE FORTRAN PROGRAM LISTING

C       TEST COMM. SUBRoutines MICRO
C
C       INTEGER ICHAR,OCHAR,FLAG
C
C       BYTE CHARS(2),ESC(2)
C
C       EQUIVALENCE.(OCHAR,CHARS)
C
C       DATA ESC/"@",033/
C
C       DATA ICHAR/0/,OCHAR/0/,FLAG/0/
C
C       INITIALIZATION
C
C       CALL START
C
C       5       CALL RECEIVED(ICARH)
C
C       IF (ICARH) 10,20,10
C
C       10       CALL ITYOUT(IDCARH)
C
C       20       CALL ITYIN(CHAR)
C
C       IF (CHAR) 5,5,30