DATA REDUCTION SYSTEM UTILIZING GOLDEN RIVER COUNTING AND RECORDING EQUIPMENT

in cooperation with the Department of Transportation Federal Highway Administration

RESEARCH REPORT 290-2
STUDY 2-18-81-290
FREeway DATA COLLECTION SYSTEM
### Data Reduction System Utilizing Golden River Counting and Recording Equipment

**Authors:**
Jae Y. Lee and Gene P. Ritch

**Performing Organization Name and Address**
Texas Transportation Institute
Texas A&M University System
College Station, Texas 77843

**Sponsoring Agency Name and Address**
Texas State Department of Highways and Public Transportation; State Transportation Planning Division,
P. O. Box 5051
Austin, Texas 78763

**Abstract**
There is a need for large quantities of traffic data to plan, design and operate urban freeways. Portable traffic counters and records, utilizing microprocessors and memory storage, have increased the efficiency in the collection of volume data, and to a limited extent, the processing and reporting of the data. The increased need for timely data requires systems that automatically process and store data in a computer data management system.

For the current microprocessor technology in portable data collection systems of traffic counters, this project develops software programs that transfer the raw data to a large computer, process and store the data in forms from which the user can request a large number of analyses and report formats with little manual intervention.

**Key Words**
Traffic Counting Systems, Data Analysis, Data Storage, Data Retrieval, Traffic Data Base, Microprocessor Applications.

**Distribution Statement**
No restriction. This document is available to the public through the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161.
DATA REDUCTION SYSTEM UTILIZING
GOLDEN RIVER COUNTING AND RECORDING EQUIPMENT

by
Jae Y. Lee
and
Gene P. Ritch

Research Report 290-2
Research Study Number 2-18-81-290
Developing a Freeway Data Collection System

Sponsored by
Texas State Department of Highways and Public Transportation
in cooperation with the
U. S. Department of Transportation, Federal Highway Administration

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas

November 1983
ABSTRACT

There is a need for large quantities of traffic data to plan, design and operate urban freeways. Portable traffic counters and recorders, utilizing microprocessors and memory storage, have increased the efficiency in the collection of volume data, and to a limited extent, the processing and reporting of the data. The increased need for timely data requires systems that automatically process and store data in a computer data management system.

For the current microprocessor technology in portable data collection systems of traffic counters, this project develops software programs that transfer the raw data to a large computer, process and store the data in forms from which the user can request a large number of analyses and report formats with little manual intervention.
DISCLAIMER

The contents of the report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

Key Words: Traffic Counting Systems, Data Analysis, Data Storage, Data Retrieval, Traffic Data Base, Microprocessor Applications
SUMMARY

This report is concerned with expanding the analysis of traffic volume data as counted and recorded by the Golden River traffic counter system. This system stores data in the counter in a memory controlled by a microprocessor. These data are transferred to another larger microprocessor called a Retriever for further processing. Two reports can be printed on a terminal directly from the Retriever. The formats of these reports are provided by the suppliers of the equipment. Different formats can be ordered from the suppliers, but the flexibility of the type of analyses and reports provided by the Retriever is limited.

This study developed programs that transfer the raw data from the Retriever processes and stores the data in a large computer in a manner that permits further analysis and reports specified by the user. Programs for daily and weekly summary reports are provided.
IMPLEMENTATION STATEMENT

The traffic data analysis programs are written in the WATFIV FORTRAN language and, with modifications, can be utilized on the Department's computer system. The software modifications are mainly associated with the input/output Fortran functions and can be implemented with minimal time and effort. The more difficult task is transmitting the traffic data from the Golden River retriever to the Department's computer system. The retriever transmits data as an asynchronous communications device. The Department's communications network is built around IBM's System Network Architecture (SNA) concept and its ability to communicate with synchronous data link controllers (SDLC). As such, the retriever must appear to be one of several SDLC devices to the Department's computer system. At present, there are only two methods to employ that will enable the retriever to communicate with the Department's computer system in this manner and both methods include additional equipment. The present FACTS* system can communicate to both the retriever and the main computer system, but software must be developed for the FACTS that will enable the traffic data to be read. The FACTS system will in turn communicate to the main computer system via a SDLC linkup. The other method is to utilize a microprocessor to read the traffic data and through software emulate a SDLC device when communicating to the Department's computer system. This approach is being studied in the current Research Project 2-18-84-421, "Developing A Freeway Data Base." The microprocessor will emulate the current data terminal through a software program and a hardware 3270 SNA/SDLC communication package.

*Flexible Advanced Computer Traffic Signal
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Situation</td>
<td>1</td>
</tr>
<tr>
<td>Procedures</td>
<td>2</td>
</tr>
<tr>
<td>Data File Set-up Procedure</td>
<td>3</td>
</tr>
<tr>
<td>Data Output Procedure</td>
<td>10</td>
</tr>
<tr>
<td>File Separation</td>
<td>11</td>
</tr>
<tr>
<td>Data Reduction</td>
<td>15</td>
</tr>
<tr>
<td>Conclusions</td>
<td>19</td>
</tr>
<tr>
<td>Appendix A</td>
<td>22</td>
</tr>
<tr>
<td>Appendix B</td>
<td>26</td>
</tr>
<tr>
<td>Appendix C</td>
<td>30</td>
</tr>
<tr>
<td>Appendix D</td>
<td>55</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>GR-0308 Interface/Charger Unit Configuration</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>Block Diagram of System Hook-Up</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>Golden River Retriever Face</td>
<td>8</td>
</tr>
<tr>
<td>4.</td>
<td>Data File Listing</td>
<td>12</td>
</tr>
<tr>
<td>5.</td>
<td>System Logic of PARTSAVE</td>
<td>14</td>
</tr>
<tr>
<td>6.</td>
<td>System Logic of RUNNER</td>
<td>16</td>
</tr>
<tr>
<td>7.</td>
<td>Daily Time Interval Report</td>
<td>17</td>
</tr>
<tr>
<td>8.</td>
<td>Weekly Hourly Report</td>
<td>18</td>
</tr>
<tr>
<td>9.</td>
<td>System Logic of REPORTER</td>
<td>20</td>
</tr>
</tbody>
</table>
LIST OF APPENDICES

1. APPENDIX A
   Listing of Execute File PARTSAVE . . . . . . . . . . 22

2. APPENDIX B
   Listing of Execute File RUNNER . . . . . . . . . . 26

3. APPENDIX C
   Listing of REPORTER . . . . . . . . . . . . . . . . . . . 30

4. APPENDIX D
   Interface Signal Setting Information . . . . . . 55
INTRODUCTION

The Golden River traffic counter system is one of the most advanced traffic counter systems available. Until recently, a sizeable and time consuming effort was required to transcribe the data from the output format of the Retriever to one which was more easily understood. This report explains the development of a computer assisted method to reduce this data reduction effort. The programs which were developed aid the user in removing the data from the Retriever, storing them in a computer file, and producing printed outputs similar to those which were obtained manually.

The procedures and programs discussed in this report were developed under Wylbur/370 system at Texas A&M University. Wylbur/370 is an on-line interactive text editing system that helps the user to create files, submit and retrieve jobs from remote terminals. Wylbur also has executing capabilities that can be used as a preprocessor to better aid the user in executing the programs. The execute files discussed in this report uses this preprocessor capability to enter different header information to the program.

The use of these programs results in a simplified and efficient reduction of traffic volume data for the analysis process.

SITUATION

The Golden River Retriever is being used in connection with a Silent-700 terminal to produce a raw data output of traffic volume counts. The Silent-700 terminal is used as an auxiliary printer connected to the
Golden River Retriever. The reduction of these raw data into useful information is a tedious manual process that occupies much of the user's time. For more efficiency, a computer can be utilized to remove the data from the retriever and store it in a computer file. Once the data are stored, a program can manipulate the data and produce reports similar to those obtained manually. This process could aid the user in a faster and more efficient analysis of traffic volume data. The data can be stored permanently in a computer memory and be used at a later date to produce reports. This is advantageous when the Retriever memory space is filled. The data can be transferred to the computer, making the Retriever available for continued data collection.

PROCEDURES

The first step in computer utilization is to remove the data from the Retriever and store it in a computer file. It is desirable that the data be transferred directly from the Retriever into a computer file with the least amount of human intervention. The transmission of data by individual files is more time consuming. It requires an interaction with Wylbur after each file is transmitted to the computer. Direct data transfer allows for a faster and more accurate data transfer and frees the user to concentrate on other activities during the data transmission.

A special cable was developed to serve as an interface between the Retriever and a Silent-700 terminal. This cable enables communication between the Retriever and the Wylbur system at the Texas A&M host computer. The data are transmitted via the RS-232 serial port connector of the
GR-0308 Interface/Charger unit to the Silent-700 terminal. The Silent/700 terminal is used as a modem connection that transmits the serial data to the host computer (Amdahl 470). The transmitted data are stored as a sequential data file on a computer disk memory.

Once the data is removed from the Retriever and stored in a computer memory, the Retriever may be cleared and used for more data retrieval. The procedure to remove the data from the Retriever and store them in a computer file is described in the following "DATA FILE SET-UP" procedure.

DATA FILE SET-UP PROCEDURE

The following procedure is to be utilized for a direct transfer of traffic volume data from a Golden River Retriever to a data file on the Amdahl 470 V6/V8 system at Texas A&M University.

• Turn on the Interface/charge unit.
• Turn on the Silent-700 terminal.
• Make the following cable hook-ups on the Interface unit:
  1) Connect P01 to P01-A.
  2) Connect P02 to On-Line Computer.
  3) Connect P03 to On-Line Terminal.
  4) Connect P04 to Silent-700 rear plug.
  5) Set the toggle switch on the switch box on "ON" position.

This switch box disables the data transmission to the printer but allows the communication between the Retriever and the Wylbur system. The user is advised not to disturb this switch position during the data transmission.
See Figure 1 for description of the Interface unit. See Figure 2 for the block diagram of the entire system hook up.

- Set the toggle switch on the Interface unit to the "On-Line Terminal" side (Right).
- Make the following settings on the Silent-700 terminal.
  1) Speed setting to High-Speed.
  2) Duplex setting to FULL Duplex.
  3) On-Line switch to "On-Line" position.

The interface signals transmitted during the data transfer from the Retriever are an important factor of data transfer. These signals control "line feed" and "carriage return" as well as other signals that can be programmed to be transmitted at the end of each line and end of each file.

- To make the proper signals, the following steps should be used on the Retriever:
  1) Place Retriever into the Interface mode by sliding the indicator bar next to "INTERFACE".
  2) The display will show three numbers.
  3) Push "2" to step to Interface mode 2. Interface 2 should be set at 300.
  4) Push "3" and the Retriever will display three numbers.
- Press "#" and "*" simultaneously and the Retriever will display
Figure 1
GR-0308 Interface/Charger Unit
Figure 2
Block Diagram of System Hook-Up
• Type in the following sequence of numbers:
  1) 013 + Enter key
  2) 999 + Enter key
  3) 999 + Enter key

These are minimum required signals and more can be found in the user's manual for the Golden River Retriever. See Figure 3 for the description of Retriever keyboard.

• Push "4" and make sure that the Interface 4 is set to "1" which automatically stops data transmission after each line, and wait for DC4 signal from Wylbur before continuing to the next line of data.

• Push 5; Interface 5 should be set on "0". See Appendix D for more information on interface signal setting procedure.

• After all the hook-ups and settings are made, on the telephone dial the dial-up number of the host computer, and wait for the high pitch. Place the phone receiver onto the cups on the Silent-700 terminal.

• The "On-Line" light should appear on the right corner of the terminal.

• Press "Return" twice and the computer will respond with a beep and a "GO".

• Press "Return" twice more and the computer will respond with "Invalid Switch Chars".

• Log-on to computer using the appropriate Texas A&M Data Processing Service Request number.

• Once logged on, get into collect mode by pressing "BREAK" key on the right-upper corner of the terminal. The computer will respond with a line number and a question mark, i.e., "1?".
Figure 3
Golden River Retriever Face
• Place the toggle switch on the Interface unit to the "RETRIEVER" side (Left).

• Get into the Output mode in the Retriever by lining up the indicator bar next to the "OUTPUT DATA" column.

• Press "*" and "#" simultaneously. Retriever will display "_ _".

• Enter a file number to transfer a single file. The user could enter "99" which will transfer all the data files on the Retriever. The user may use a special command, Collect Unnumbered, which will squeeze each line of data into a single line on the printer. To use this command, type in "COL UNN" under "COMMAND" mode and the computer will no longer print out the line numbers.

• Examine the data as they are being transferred to the Amdahl computer.

Sometimes the user may want to disable the printer and proceed with the data transfer silently.

• To disable the printer, turn the "On-Line" switch on the Silent-700 to "OFF" position. The "On-Line" light should remain on.
  1) Data Transfer is shown by the blinking light on "DATA" indicator bulb on the Interface unit.
  2) The Retriever should display four dashes during the data transfer. Wait until both the blinking light and the dashes disappear which indicates the end of data transfer.

• Now the user must save the data transferred under a unique name for later usage.

• To get back to Wylbur, set the toggle switch on the Interface unit back to "On-Line Terminal" side and the "On-Line" switch on the
Silent-700 back to "On-Line" position.

- Press "BREAK" and the computer will respond with "COMMAND?"
- Save the data file under a unique alphabetic name of six characters or less.
- Examine the data type "LIST" and inspect the data as they are printed across the printer. "LIST UNN" will squeeze each line of data onto single line on the printer.
- Log-off by typing LOGOFF CLE.

**DATA OUTPUT PROCEDURE**

Sometimes the user may wish to obtain raw data printout without creating a computer file. The following procedure can be used to obtain print out of the data from the Retriever.

- Turn on the GR-0308 Interface/Charger unit.
- Turn on the Silent-700 terminal.
- Make the following cable hook-ups on the GR-0308 Interface unit:
  1) Connect P01 to P01-A.
  2) Connect P02 to On-Line Computer.
  3) Connect P03 to Off-Line Terminal.
  4) Connect P04 to Silent-700 rear plug.
  5) Set the toggle switch on the switch box on "ON" position.
- Set the toggle switch on the GR-0308 Interface unit to the "Off-Line Terminal" side (Left).
- Following the same procedure described on page 4, make the following
interface signals:
1) 013 + Enter key
2) 010 + Enter key
3) 099 + Enter key
4) 099 + Enter key

• The Interface 4 should be set to "Ø" which allows continuous data transmission.

• Get the Retriever into output mode by lining up the bar indicator next to the "OUTPUT DATA" column.

• Press "*" and "," simultaneously. Retriever will display "__".

• Enter a file number to transfer a specific file. The user could enter "99" which will transfer all the data files on the Retriever.

• Examine the data as they are printed.

This procedure produces simple listing of the collected data and does not create a computer file.

FILE SEPARATION

Once the data are stored as a sequential file, it must be subdivided into several files. Each segment of the original file is separated by marker statements "*BEGIN" and "*END" as shown in Figure 4. The segments delimited by "*BEGIN" and "*END" of the original data file must be subdivided into individual files. Then the program can reduce the data within the individual files and produce a report. Partitioned data sets are used to group the individual files together under a master file. A partitioned data set will save the entire file under a master file name and each individual file will be given a distinct member name. This allows the user to
<table>
<thead>
<tr>
<th>1.</th>
<th>*BEGIN 00 01 03310482 0015 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>830613 1415 0080 0097 0101 0115 0092 0099 0098 0086 0080 0073 0100 0099</td>
</tr>
<tr>
<td>3.</td>
<td>830613 1715 0122 0125 0092 0098 0074 0081 0101 0065 0068 0065 0051 0049</td>
</tr>
<tr>
<td>4.</td>
<td>830613 2015 0050 0048 0061 0043 0052 0039 0024 0036 0036 0027 0023 0019</td>
</tr>
<tr>
<td>5.</td>
<td>830613 2315 0021 0027 0021 0013 0019 0004 0007 0011 0002 0006 0007 0002</td>
</tr>
<tr>
<td>6.</td>
<td>830614 0215 0007 0011 0014 0002 0005 0004 0003 0003 0006 0011 0010</td>
</tr>
<tr>
<td>7.</td>
<td>830614 0515 0012 0024 0043 0050 0064 0122 0126 0136 0132 0171 0197 0223</td>
</tr>
<tr>
<td>8.</td>
<td>830614 0815 0196 0172 0176 0139 0131 0145 0142 0098 0087 0093 0096</td>
</tr>
<tr>
<td>9.</td>
<td>*END 00 01 03310482 0015 00</td>
</tr>
<tr>
<td>10.</td>
<td>*BEGIN 00 02 03310780 0015 00</td>
</tr>
<tr>
<td>11.</td>
<td>830613 1100 0114 0142 0141 0162 0136 0151 0165 0157 0157 0162 0179 0169</td>
</tr>
<tr>
<td>12.</td>
<td>830613 1400 0198 0158 0142 0181 0161 0155 0151 0261 0152 0218 0168 0207</td>
</tr>
<tr>
<td>13.</td>
<td>830613 1700 0147 0233 0128 0145 0111 0094 0084 0090 0065 0072 0064 0053</td>
</tr>
<tr>
<td>14.</td>
<td>830613 2000 0067 0051 0044 0046 0047 0031 0023 0026 0033 0027 0021 0031</td>
</tr>
<tr>
<td>15.</td>
<td>830613 2300 0014 0018 0020 0009 0013 0008 0015 0066 0002 0012 0004 0011</td>
</tr>
<tr>
<td>16.</td>
<td>830614 0200 0012 0009 0010 0004 0003 0002 0003 0007 0005 0006 0006 0112</td>
</tr>
<tr>
<td>17.</td>
<td>830614 0500 0009 0019 0015 0018 0018 0053 0050 0101 0118 0143 0185 0225</td>
</tr>
<tr>
<td>18.</td>
<td>830614 0800 0208 0185 0168 0149 0141 0129 0134 0127 0120 0133 0117 0128</td>
</tr>
<tr>
<td>19.</td>
<td>830614 1100 0129 0130</td>
</tr>
<tr>
<td>20.</td>
<td>*END 00 02 03310780 0015 00</td>
</tr>
<tr>
<td>21.</td>
<td>*BEGIN 00 03 03310777 0015 00</td>
</tr>
<tr>
<td>22.</td>
<td>830613 1115 0056 0081 0070 0048 0062 0068 0063 0041 0055 0059 0054 0056</td>
</tr>
<tr>
<td>23.</td>
<td>830613 1415 0060 0052 0069 0058 0053 0060 0082 0075 0066 0084 0071 0074</td>
</tr>
<tr>
<td>24.</td>
<td>830613 1715 0084 0096 0074 0077 0057 0043 0049 0042 0046 0036 0041 0032</td>
</tr>
<tr>
<td>25.</td>
<td>830613 2015 0034 0035 0026 0036 0034 0033 0029 0027 0021 0025 0022 0028</td>
</tr>
<tr>
<td>26.</td>
<td>830613 2315 0013 0018 0005 0013 0009 0010 0005 0007 0003 0003 0007 0002</td>
</tr>
<tr>
<td>27.</td>
<td>830614 0215 0003 0005 0002 0004 0007 0001 0003 0003 0001 0004 0002 0003</td>
</tr>
<tr>
<td>28.</td>
<td>830614 0515 0005 0003 0016 0028 0038 0024 0048 0047 0049 0064 0063 0054</td>
</tr>
<tr>
<td>29.</td>
<td>830614 0815 0067 0067 0062 0061 0061 0065 0062 0068 0061 0036 0049 0050</td>
</tr>
<tr>
<td>30.</td>
<td>830614 1115 0050</td>
</tr>
<tr>
<td>31.</td>
<td>*END 00 03 03310777 0015 00</td>
</tr>
<tr>
<td>32.</td>
<td>*BEGIN 00 04 03310483 0015 00</td>
</tr>
<tr>
<td>33.</td>
<td>830613 1130 0163 0176 0145 0141 0118 0116 0130 0106 0120 0134 0171 0129</td>
</tr>
<tr>
<td>34.</td>
<td>830613 1430 0152 0177 0169 0149 0154 0150 0100 0096 0135 0146 0151 0192</td>
</tr>
<tr>
<td>35.</td>
<td>830613 1730 0204 0176 0149 0085 0093 0102 0113 0086 0081 0058 0064 0059</td>
</tr>
<tr>
<td>36.</td>
<td>830613 2030 0041 0058 0054 0040 0043 0042 0037 0035 0035 0040 0032 0026</td>
</tr>
<tr>
<td>37.</td>
<td>830613 2330 0026 0024 0021 0014 0011 0014 0007 0010 0011 0005 0009 0005</td>
</tr>
<tr>
<td>38.</td>
<td>830614 0230 0006 0001 0002 0004 0002 0003 0003 0002 0007 0003 0006</td>
</tr>
<tr>
<td>39.</td>
<td>830614 0530 0012 0025 0029 0026 0036 0076 0096 0124 0105 0120 0143 0156 0142</td>
</tr>
<tr>
<td>40.</td>
<td>830614 0830 0153 0130 0159 0136 0149 0141 0198 0139 0100 0121 0110 0149</td>
</tr>
<tr>
<td>41.</td>
<td>*END 00 04 03310483 0015 00</td>
</tr>
</tbody>
</table>
address the individual files and also store all the individual files together in one master file.

The execute file named PARTSAVE was developed to aid the user in dividing the original data set into a partitioned data set. The following procedure is used to run the execute file PARTSAVE.

- Log-On to Wylbur
- Type "EXEC FROM PARTSAVE CLE"
- The execute file will interact with the user in conversational style to aid the user in dividing the data set.

The following algorithm is used in the execute file PARTSAVE. The execute file prompts the user to create a master file and provides instructions if the user needs assistance. Once the master file is set-up, the execute file will ask the user for the original file, working from end of the file and prints each individual section for verification. The user has the choice to either save or delete that portion of the data file. If the user chooses to save that segment, the execute file will ask for a member name to be assigned to that segment. The execute file will continue this process until it reaches the end of data set or the user "BREAK" operation. After the completion of the execute file "PARTSAVE", the user will have the original data file plus a master partitioned data set with its individual members. The system logic of PARTSAVE described in the flowchart in Figure 5. See Appendix A for the complete listing of the execute file PARTSAVE.
INTRODUCTION

CREATE PDS?

YES

INSTRUCTIONS TO SET-UP PDS

NO

INPUT MASTER, ORIGINAL FILE NAME

LIST EACH SEGMENT

SAVE THIS SEGMENT?

YES

SAVE UNDER MEMBER NAME

NO

DELETE THIS SEGMENT

CONTINUE?

YES

NO

STOP

Figure 5

System Logic of PARTSAVE
DATA REDUCTION

After the data set is divided into a partitioned data set, the user is ready to use the program named REPORTER, and reduce the raw data. The execute file "RUNNER" has been developed to aid the user in running the program REPORTER, which reduces data and produces the reports. The program RUNNER is used in the following manner. It interacts with the user to obtain the specific data set to be reduced, the specific count location, time interval, weather condition, counter, etc. Once this information is obtained, it passes on these parameters to the program REPORTER. They are to be used in producing reports. Program RUNNER will initiate the execution of the program REPORTER. REPORTER produces the report and returns to the execute file to check for continuing execution. If there are multiple files to be reduced, the user simply replies with the name of other files to be reduced. The system logic of RUNNER is described by the flowchart in Figure 6. See Appendix B for the complete listing of the execute file RUNNER.

The program REPORTER reduces the data according to the time interval and the appropriate configuration. REPORTER produces two reports: Daily time interval sums and weekly hourly totals. It also prints running subtotals and hourly totals for better data analysis. Two reports produced by REPORTER are shown in Figures 7 and 8.

The program REPORTER is designed with several modules, each with distinct functions which aid in producing the reports. Subroutine READER reads the specified data file and passes this information to module PREP. PREP is the core segment of REPORTER. It reduces the data
START

PRINT OUT INTRODUCTION

FAMILIAR?

YES

INPUT DATE, LOC, WEATHER, DATA FILE, COUNTER, ETC.

PLACE REPORTER IN ACTIVE FILE

ADD DATA FILE TO END

EXECUTE THE REPORTER

MORE FILES?

YES

NO

STOP

INSTRUCTIONS TO RUN PROGRAM

Figure 6

System Logic of RUNNER
### Daily Time Interval Report

**DATE:** 6-14-1983  
**WEATHER:** GOOD  
**LOCATION:** 145 • GREENS ROAD  
**SOURCE:** THIS IS 1 OF 2

<table>
<thead>
<tr>
<th>TIME</th>
<th>VOLUME</th>
<th>TIME</th>
<th>VOLUME</th>
<th>TIME</th>
<th>VOLUME</th>
<th>TIME</th>
<th>VOLUME</th>
<th>TIME</th>
<th>VOLUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00- 0.15</td>
<td>25</td>
<td>6.00- 6.15</td>
<td>87</td>
<td>12.00-12.15</td>
<td>0</td>
<td>18.00-18.15</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15- 0.30</td>
<td>14</td>
<td>6.15- 6.30</td>
<td>113</td>
<td>12.15-12.30</td>
<td>0</td>
<td>18.15-18.30</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.30- 0.45</td>
<td>19</td>
<td>6.30- 6.45</td>
<td>177</td>
<td>12.30-12.45</td>
<td>0</td>
<td>O</td>
<td>18.30-18.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.45- 1.00</td>
<td>11</td>
<td>6.45- 7.00</td>
<td>185</td>
<td>12.45-13.00</td>
<td>0</td>
<td>18.45-19.00</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00- 1.15</td>
<td>15</td>
<td>7.00- 7.15</td>
<td>232</td>
<td>13.00-13.15</td>
<td>0</td>
<td>19.00-19.15</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.15- 1.30</td>
<td>11</td>
<td>7.15- 7.30</td>
<td>267</td>
<td>13.15-13.30</td>
<td>0</td>
<td>19.15-19.30</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.30- 1.45</td>
<td>5</td>
<td>7.30- 7.45</td>
<td>299</td>
<td>13.30-13.45</td>
<td>0</td>
<td>O</td>
<td>19.30-19.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1.45- 2.00</td>
<td>10</td>
<td>7.45- 8.00</td>
<td>321</td>
<td>13.45-14.00</td>
<td>0</td>
<td>19.45-20.00</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00- 2.15</td>
<td>9</td>
<td>8.00- 8.15</td>
<td>252</td>
<td>14.00-14.15</td>
<td>0</td>
<td>20.00-20.15</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.15- 2.30</td>
<td>11</td>
<td>8.15- 8.30</td>
<td>262</td>
<td>14.15-14.30</td>
<td>0</td>
<td>20.15-20.30</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.30- 2.45</td>
<td>6</td>
<td>8.30- 8.45</td>
<td>206</td>
<td>14.30-14.45</td>
<td>0</td>
<td>O</td>
<td>20.30-20.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2.45- 3.00</td>
<td>5</td>
<td>8.45- 9.00</td>
<td>187</td>
<td>14.45-15.00</td>
<td>0</td>
<td>O</td>
<td>20.45-21.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3.00- 3.15</td>
<td>11</td>
<td>9.00- 9.15</td>
<td>161</td>
<td>15.00-15.15</td>
<td>0</td>
<td>O</td>
<td>21.00-21.15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3.30- 3.45</td>
<td>7</td>
<td>9.30- 9.45</td>
<td>153</td>
<td>15.30-15.45</td>
<td>0</td>
<td>O</td>
<td>21.30-21.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3.45- 4.00</td>
<td>12</td>
<td>9.45-10.00</td>
<td>174</td>
<td>15.45-16.00</td>
<td>0</td>
<td>O</td>
<td>21.45-22.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4.00- 4.15</td>
<td>4</td>
<td>10.00-10.15</td>
<td>133</td>
<td>16.00-16.15</td>
<td>0</td>
<td>O</td>
<td>22.00-22.15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4.15- 4.30</td>
<td>7</td>
<td>10.15-10.30</td>
<td>120</td>
<td>16.15-16.30</td>
<td>0</td>
<td>O</td>
<td>22.15-22.30</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4.30- 4.45</td>
<td>12</td>
<td>10.30-10.45</td>
<td>126</td>
<td>16.30-16.45</td>
<td>0</td>
<td>O</td>
<td>22.30-22.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4.45- 5.00</td>
<td>23</td>
<td>10.45-11.00</td>
<td>103</td>
<td>16.45-17.00</td>
<td>0</td>
<td>O</td>
<td>22.45-23.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5.00- 5.15</td>
<td>17</td>
<td>11.00-11.15</td>
<td>125</td>
<td>17.00-17.15</td>
<td>0</td>
<td>O</td>
<td>23.00-23.15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5.15- 5.30</td>
<td>30</td>
<td>11.15-11.30</td>
<td>145</td>
<td>17.15-17.30</td>
<td>0</td>
<td>O</td>
<td>23.15-23.30</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5.30- 5.45</td>
<td>33</td>
<td>11.30-11.45</td>
<td>153</td>
<td>17.30-17.45</td>
<td>0</td>
<td>O</td>
<td>23.30-23.45</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5.45- 6.00</td>
<td>73</td>
<td>11.45-12.00</td>
<td>0</td>
<td>17.45-18.00</td>
<td>0</td>
<td>O</td>
<td>23.45-24.00</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**AM PEAK (6:30 - 9:30) VOLUME:** 2689  
**AM PEAK HIGHEST HOUR:** 1139  
**PM PEAK (3:30 - 6:30) VOLUME:** 0  
**PM PEAK HIGHEST HOUR:** 0  
**NEAREST 15 MINUTE COUNT:** 0  
**STARTING TIME:** 7:15  
**STARTING TIME:** 0:00  

**Figure 7**
### Freeway Volume Count Summary Sheet

**Location:** I45 @ Greens Road  
**Date:** 6-13-1983  
**Direction of Flow:** Northbound on I45  
**Source:** This is 1 of 2  
**Site No:** 3310775

**Day of Week:**  
- Monday: 6/13  
- Tuesday: 6/14  
- Wednesday: 6/15  
- Thursday: 6/16  
- Friday: 6/17  
- Saturday: 6/18  
- Sunday: 6/19

<table>
<thead>
<tr>
<th>Time</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
<th>Hourly Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>69</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>41</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4-5</td>
<td>0</td>
<td>107</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5-6</td>
<td>0</td>
<td>562</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6-7</td>
<td>0</td>
<td>1119</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7-8</td>
<td>0</td>
<td>907</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8-9</td>
<td>0</td>
<td>628</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9-10</td>
<td>0</td>
<td>482</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10-11</td>
<td>0</td>
<td>270</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11-12</td>
<td>532</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12-13</td>
<td>555</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13-14</td>
<td>561</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14-15</td>
<td>525</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15-16</td>
<td>543</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16-17</td>
<td>700</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17-18</td>
<td>461</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18-19</td>
<td>363</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19-20</td>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20-21</td>
<td>315</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21-22</td>
<td>258</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22-23</td>
<td>172</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23-24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**AM Peak (6-9) Volume:** 2588  
**AM Peak Highest Hour:** 1119  
**PM Peak (3-6) Volume:** 1768  
**PM Peak Highest Hour:** 700  
**DT:** 5285  

*Figure 8*
according to time interval and configuration. PREP calculates the hourly totals, AM and PM peak period volumes, running subtotals, and other useful information. PREP then passes this information to sub-routines DAILY and WEEKLY.

There are three possible ways to obtain the print out. DAILY produces the 24-hour summary and weekly produces the weekly hourly volume report. Both DAILY and WEEKLY produces the reports on each individual channel. They can produce up to seven reports for six channel configuration where the seventh report will contain the sum of all the channels. The user may choose to print out DAILY and/or WEEKLY report. On a multi-channel configuration, the channels with no data, all zero recordings, will not be printed. The user makes the choice during the interaction with the execute file RUNNER which will make the proper changes in the REPORTER to print out the specified reports. The system logic of the program RUNNER is described in a flow chart in Figure 9. See Appendix C for the complete listing of the program REPORTER.

The entire system is designed in modular style to ease the modification of the programs to obtain other information needed by the user. It can be also modified to be used under different computer systems.

CONCLUSIONS

The proper use of the computer programs described within this report results in a more efficient use of the Golden River traffic counter system. The developed programs aid the users in producing printed reports with less overall effort than was previously required using manual methods.
SUBROUTINE READER
READS DATA INTO ARRAYS

SUBROUTINE PREP
REDUCES THE DATA
ACCORDING TO THE INTERVAL, CONFIGURATION

DAILY REPORT ?
YES
SUBROUTINE DAILY TO PRODUCE 24 HOUR REPORT

NO

WEEKLY REPORT ?
YES
SUBROUTINE WEEKLY TO PRODUCE THE WEEKLY REPORT

NO

STOP

Figure 9
System Logic of REPORTER
Computer disk storage of the data file, as developed here, will aid in the long term storage and accessibility of the field data. The future developments will be in the area of utilizing a microcomputer to aid in data reduction system. The data will be transmitted to a microcomputer and stored on a floppy diskette for permanent storage.
Listing of Execute File PARTSAVE

1. SET ESCAPE %
2. ; THIS IS AN EXECUTE FILE TO DIVIDE A DATA FILE
3. ; INTO EACH INDIVIDUAL SEGMENT. EACH SEGMENT IS
4. ; DELIMITED BY BOTH '*BEGIN' AND '*END' STATEMENT
5. ;
6. ;
7. COMM THIS PROGRAM WILL AID YOU IN DIVIDING YOUR DATA FILE INTO
8. COMM SEVERAL INDIVIDUAL FILES. YOU MUST ALREADY HAVE CREATED A
9. COMM DUMMY PARTITIONED DATA SET WITH ENOUGH DIRECTORY AND SPACE
10. COMM ALLOCATED. IF YOU HAVE NOT DONE SO, THIS PROGRAM WILL NOT
11. COMM WORK FOR YOU. IF YOU NEED INSTRUCTIONS TO SET UP THIS DUMMY
12. COMM FILE, TYPE "YES" AFTER THE PROMPT "DO YOU NEED INSTRUCTIONS ?"
13. COMM OTHERWISE THIS PROGRAM WILL ASSUME THAT YOU HAVE ALREADY DONE
14. COMM THE PREREQUISITE STEPS.
15. COMM
16. READ STRING S7 PROMPT 'DO YOU NEED INSTRUCTIONS ? (Y/N)'
17. IF (SUBSTR(S7,1) EQ 'Y') POINT ';CREATE' EXEC NOL
18. COMM
19. COMM THIS PROGRAM NEEDS THE FOLLOWING INFORMATION.
20. ;
21. ;
22. ; CREATE PDS ?
23. ;
24. ;
25. COMM
26. COMM NAME OF FILE TO BE DIVIDED -- YOUR ORIGINAL DATAFILE
27. COMM MASTER FILE NAME? -- MASTER NAME IN WHICH THE PARTITIONED
28. COMM DATA SETS WILL BE SAVED.
29. COMM MEMBER NAME? -- NAME OF THE INDIVIDUAL DATA FILE TO BE
30. COMM SAVED UNDER. MUST BE LESS THAN 8 CHARACTERS.
31. COMM
32. COMM REMEMBER THAT THE MASTER NAME CANNOT BE SAME AS THE
33. COMM ORIGINAL DATAFILE NAME!!!
34. COMM
35. COMM
36. ;RETURN POINT
37. ;
38. ;
39. ; INPUT MASTER, ORIGINAL FILE NAME
40. ;
41. ;
42. READ STRING S4 PROMPT 'NAME OF THE FILE TO BE DIVIDED: '
43. READ STRING S2 PROMPT 'MASTER FILE NAME: '
44. USE %S4 CLE
45. ;START
46. POINT '*BEGIN' NOL
47. SET VAL W1=CURRENT
48. POINT '*END' IN %W1/L NOL
49. SET VAL W2=CURRENT
50. ;
51. ;
52. ; LIST EACH SEGMENT
53. ;
54. ;
55. LIST %W1/%W2
56. ; LIST DATA TO VERIFY THE FILE
57. READ STRING S1 PROMPT 'SAVE THIS FILE? (Y/N) '
58. IF (SUBSTR(S1,1) EQ 'Y') POINT ';SAVE' EXEC NOL
59. DEL %W1/%W2
COMM LET'S CONTINUE WITH THE NEXT SEGMENT....

COMM

POINT ';START'; EXEC NOL
;
;
; SAVE UNDER MEMBER NAME
;
;
;SAVE

READ STRING S3 PROMPT 'MEMBER NAME: '
SAVE %S2(%S3) LINES %W1/%W2
DELETE %W1/%W2
;
;
; CONTINUE ?
;
;
;
; CHECK FOR END OF DATA FILE
IF (W1 EQ 1) POINT ';QUIT'; EXEC NOL
;
;
COMM LET'S CONTINUE WITH THE NEXT SEGMENT....

POINT ';START'; EXEC NOL
;
;
;
;CREATE DUMMY PARTITIONED DATASET
;
;
; INSTRUCTIONS TO SET UP DUMMY PARTITIONED DATASET

COMM

COMM FOLLOWING INFORMATION IS NEEDED TO SET UP A PARTITIONED DATASET

COMM

COMM DIRECTORY SIZE -- SINGLE DIRECTORY WILL HOLD ABOUT 10
COMM MEMBER FILES. DIVIDE THE TOTAL NUMBER OF FILES BY 10
COMM AND ADD 20% TO THIS SIZE FOR LATER ADDITION.
COMM SPACE -- NUMBER OF TRACKS TO BE ALLOCATED FOR THE DUMMY
COMM DATASET. SPACE SHOULD BE ESTIMATED 20% HIGHER THAN THE
COMM CURRENT NUMBER OF TRACKS OCCUPIED BY THE ORIGINAL FILE
COMM

;

READ STRING S9 PROMPT 'DO YOU NEED THIS INFORMATION? (Y/N)'

IF (SUBSTR(S9,1) EQ 'N') POINT ';QUIT'; EXEC NOL

READ STRING S8 PROMPT 'YOUR DPSR?'

READ STRING S7 PROMPT 'YOUR ID?'

READ STRING S6 PROMPT 'ORIGINAL DATAFILE NAME?'

;

SHOW DSN LIKE $USR.%S8.%S7.%S6 ALL
;
;
;
;
;
COMM

COMM DUMMY

READ STRING S5 PROMPT 'DO YOU WANT TO SET UP DUMMY FILE? (Y/N)'

IF (SUBSTR(S5,1) EQ 'N') POINT ';QUIT'; EXEC NOL

READ STRING S9 PROMPT 'MASTER FILE NAME?'

READ STRING S8 PROMPT 'FIRST MEMBER FILE NAME?'

READ VALUE N1 PROMPT 'DIRECTORY SIZE?'
120.       READ VALUE N2 PROMPT 'SPACE SIZE ?'
121.       COMM
122.       COMM
123.       ;
124.       ;
125.       ; SET UP MASTER PDS
126.       ;
127.       ;
128.       SAVE %S9(%S8) DIR %N1 CARD SPACE %N2
129.       COMM
130.       COMM
131.       COMM YOU ARE NOW READY TO DIVIDE THE FILES.
132.       POINT ';RETURN' EXEC NOL
133.       ;QUIT
134.       COMM HAVE A NICE DAY !!!!
135.       COMM
APPENDIX B
Listing of Execute File RUNNER

1. SET ESC &
2. ; THIS EXECUTE FILE AIDS THE USER IN PRODUCING THE SUMMARY REPORTS.
3. ; THIS EXECUTE FILE INTERACTS WITH THE USER TO OBTAIN INFORMATION
4. ; NEEDED TO PRODUCE THE SUMMARY REPORTS SUCH AS DATE, LOCATION,
5. ; TIME INTERVAL, WEATHER CONDITION, SPECIFIC DATA FILE, ETC.
6. ; IT THEN SELECTS THE SPECIFIED DATA FILE AND INITIATE THE ACTUAL
7. ; PROGRAM WHICH PRODUCES THE SUMMARY REPORTS.
8. ;
9. COMM WELCOME TO CONVERSATIONAL COMPUTER SYSTEM. IF YOU ARE ALREADY
10. COMM FAMILIAR WITH THIS PROGRAM PROCEDURES, TYPE "YES" TO SKIP THE
11. COMM INTRODUCTION. OTHERWISE YOU WILL BE GIVEN THE INSTRUCTIONS ON
12. COMM HOW TO USE THIS PROGRAM.
13. COMM
14. ;
15. ; FAMILIAR USER?
16. ;
17. READ STRING S9 PROMPT 'ARE YOU ALREADY FAMILIAR ? '
18. IF (S9 EQ 'YES') POINT ';START' EXEC NOL
19. COMM
20. ;
21. ; INSTRUCTIONS TO RUN THE PROGRAM
22. ;
23. ;
24. COMM YOU ARE NOW IN A POSITION TO CONTROL THE COMPUTER PROCESSING.
25. COMM COMPUTER WILL PROMPT YOU FOR SOME SPECIFIC INFORMATION WHICH
26. COMM YOU MUST SUPPLY BEFORE PROPER PROCESSING CAN BEGIN. YOUR
27. COMM COOPERATION IS A VITAL PART OF THIS PROGRAM.
28. COMM
29. COMM
30. COMM
31. COMM IF YOU MAKE A TYPING ERROR DURING YOUR CONVERSATION WITH
32. COMM THE COMPUTER, YOU CAN DO EITHER OF TWO THINGS.
33. COMM
34. COMM 1. IF YOU HAVE ALREADY PRESSED "RETURN", YOU WILL SIMPLY
35. COMM HAVE TO START ALL OVER.
36. COMM 2. IF YOU HAVE NOT PRESSED "RETURN", SIMPLY BACK SPACE
37. COMM UNTIL YOU REACH YOUR ERROR, THEN RETYPE THE REMAINING
38. COMM LINE OVER AGAIN. ONCE YOU BACK TRACK, YOU LOOSE THAT
39. COMM LETTER EVEN IF IT REMAINS ON YOUR TERMINAL.
40. COMM
41. COMM
42. COMM THANK YOU FOR YOUR COOPERATION.
43. COMM
44. COMM LET'S BEGIN....
45. COMM
46. COMM
47. COMM
48. ;
49. ;
50. ; INPUT DATE, LOC, WEATHER, ETC.
51. ;
52. ;START PROCESSING
53. COMM TYPE IN DATE AS MM-DD-YY
54. READ STRING S1 PROMPT 'DATE ? '
55. ;
56. READ STRING S2 PROMPT 'LOCATION ? '
57. READ STRING S3 PROMPT 'DIRECTION OF FLOW ? '
58. READ STRING S4 PROMPT 'WEATHER ? '
59. READ STRING S5 PROMPT 'COUNTER ? '

27
60. READ STRING S6 PROMPT 'WHICH DATA FILE ?'
61. READ VALUE N1 PROMPT 'TIME INTERVAL IN MINUTES ?'
62. 
63. 
64. COMM
65. COMM THANK YOU FOR YOUR COOPERATION
66. COMM
67. 
68. 
69. ; PLACE REPORTER IN ACTIVE FILE SPACE
70. 
71. 
72. USE REPORTER CLE
73. CH '&DATE' TO '&S1' N
74. CH '&LOC' TO '&S2' N
75. CH '&DIR' TO '&S3' N
76. CH '&WEATHER' TO '&S4' N
77. CH '&COUNTER' TO '&S5' N
78. CH '&DELTA' TO &N1 N
79. CH '&TITLE' TO '&S6' N
80. 
81. ; SELECT THE DATA FILE TO BE READ IN
82. ; ADD DATA FILE TO END OF THE PROGRAM
83. 
84. 
85. 
86. COPY FROM &S6 TO END
87. SET VALUE WO=LAST-1
88. &WO+.1 999999
89. 
90. 
91. ; MAKE CHANGES FOR PRINTING OUT THE TOTAL VOLUMES
92. 
93. 
94. COMM WHICH SUMMARY REPORT DO YOU WANT?
95. COMM D - DAILY REPORT
96. COMM W - WEEKLY REPORT
97. COMM B - BOTH DAILY AND WEEKLY REPORTS
98. COMM
99. READ STRING S1 PROMPT 'TYPE IN YOUR CHOICE (D,W,B)'
100. 
101. IF (S1 EQ 'D') CH '777' TO 'C77' NOL
102. IF (S1 EQ 'W') CH '888' TO 'C88' NOL
103. 
104. 
105. READ STRING S2 PROMPT 'DO YOU WANT THE CHANNEL TOTALS? (Y/N)'
106. IF (S2 EQ 'Y') CH 'LT. 7' TO 'LT. 8' NOL
107. 
108. 
109. ; EXECUTE THE PROGRAM REPORTER
110. 
111. 
112. 
113. RUN H
114. COMM IF YOU WISH TO CONTINUE PROCESSING, TYPE 'YES'
115. COMM AFTER THE PROMPT 'CONTINUE ?'.
116. COMM OTHERWISE, SIMPLY RETURN AND THIS PROGRAM WILL
117. COMM TERMINATE AUTOMATICALLY.
118. 
119. ; MORE FILES ?
120.  
121.  
122.  READ STRING S7 PROMPT 'CONTINUE ? '
123.  IF (S7 EQ 'YES') POINT ';'START'1 EXEC NOL
124.  COMM HAVE A NICE DAY!!!!
125.  COMM
APPENDIX C
Listing of REPORTER

1. //&TITLE JOB (W262,501D,S02,003,JL), 'REPORT'
2. //*XBM WATFIV
3. C
4. C
5. C
6. C
7. C******************************************************* ************
8. C
9. C
10. C THIS PROGRAM WAS DEVELOPED TO AID THE USER IN REDUCING THE
11. C TRAFFIC VOLUME DATA. DATA IS TRANSFERRED FROM THE GOLDEN
12. C RIVER RETRIEVER TO THE COMPUTER MEMORY BY DIRECT DATA COM-
13. C MUNICATION PROCEDURE. THE EXECUTE FILE RUNNER PASSES THE
14. C HEADER INFORMATION TO THIS PROGRAM AND CONCATENATE THE DATA
15. C FILE TO THE END OF THIS PROGRAM. THIS PROGRAM IS DESIGNED
16. C WITH MANY SUBROUTINES TO EASE THE MODIFICATION OF THE THIS
17. C PROGRAM AT A LATER DATE. THIS PROGRAM CAN BE MODIFIED TO
18. C WORK UNDER A DIFFERENT SYSTEM AS WELL AS FOR ADDITIONAL IN-
19. C FORMATION NEEDED BY THE USER.
20. C
21. C******************************************************* ************
22. C
23. COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
24. COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
25. COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
26. COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
27. COMMON DATE(200),TIME(200)
28. COMMON CONFIG,TCONFG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
29. COMMON TFilen,Tsiten,Intval,Tintvl,Status,Index
30. COMMON TStats,AMPpeak(7),PMpeak(7),AMhigh(7),PMhigh(7),RTIME
31. INTEGER P,NWtot(7,7,24),Check,APEAK(7),PPEAK(7),AHI(7),PHI(7)
32. CHARACTER*8 HEADER,TRAIL,TEST
33. INTEGER TSDate,DATE,CHOTOT,TESDAY
34. INTEGER COUNT,INDEX,TSITEN
35. INTEGER CONFIG,TCONFG,FILENO,TFilen,SITEN,TSITEN
36. INTEGER ARMON(7),ARDAY(7)
37. INTEGER INTVAL,TINTVL,STATUS,TSTATS,TIME
38. INTEGER BIN1,BIN2,BIN3,BIN4
39. INTEGER BIN5,BIN6,BIN7,BIN8
40. INTEGER BIN9,BIN10,BIN11,BIN12,MDayz
41. INTEGER AMPpeak,PMpeak,AMhigh,PMhigh,RTIME,IMAX
42. INTEGER ENDEX,D3,DAY,MONTH,YEAR
43. C
44. C
45. C
46. C IFLAG=0
47. C
48. C IFLAG IS USED TO DETECT THE MISMATCH OF THE TIME INTERVAL
49. C
50. C
51. DO 10 I = 1,7
52. APEAK(I) = 0
53. PPEAK(I) = 0
54. AHI(I) = 0
55. PHI(I) = 0
56. C
57. DO 5 J = 1,24
58. C
59. C******************************************************************************
THIS SEGMENT OF THE PROGRAM IS DESIGNED TO SET UP A THREE
DIMENSIONAL ARRAY IN ORDER TO ASSIST IN PRINTING OUT THE TABLE. THE FOLLOWING IS A DESCRIPTION OF THE ARRAY 'NWTOT'

K - THE FIRST SUBSCRIPT IS FOR THE CHANNEL NUMBER 1 - 6
I - THE SECOND SUBSCRIPT IS FOR THE DAY OF WEEK (1 - 7)
J - THE LAST SUBSCRIPT IS FOR THE NUMBER OF HOURS IN THE DAY (1 - 24)

DO 4 K = 1,7
   NWTOT(K,I,J) = 0
4 CONTINUE

CALL SUBROUTINE READER TO READ THE DATA FROM DISK FILE
CALL READER(IMAX)

CALL SUBROUTINE DAWEEK TO FIND THE DAY-OF-WEEK GIVEN THE DATE OF DATA COLLECTION.

TSDATE = DATE(1)
TESDAY = DATE(1)
CALL DAWEEK(TSDATE,ENDEX,ARMON,ARDAY,CHECK,DAY,MONTH,YEAR)
P = CHECK
IND=ENDEX*24+1
IF (IND .EQ. 0) IND = 1
INDEX=1
ITIME = TIME(INDEX)/100
IT = 1
IJ = 1
WHILE(INDEX .LE. IMAX) DO
INDEX IS THE VARIABLE THAT CHECKS AGAINST THE END OF FILE
CONTINUE LOOP UNTIL END OF FILE HAS BEEN REACHED
CALL SUBROUTINE INITAL TO INITIALIZE ARRAYS
CALL INITAL(MDAYZ,CHOTOT)
CALL SUBROUTINE PREP TO PREPARE THE OUTPUT REPORT

CALL PREP (P,NWTOT,D3,IJ,ITIME,IT,IFLAG)

CHECK FOR ABORTED ENDING DUE TO DIFFERENT TIME INTERVALS

IF (IFLAG .EQ. 1) GOTO 99

P = P + 1

IF (TIME(INDEX) .GT. 0) ITIME = TIME(INDEX)/100

CALL SUBROUTINE DAILY TO PRINT OUT DAILY REPORT

CALL DAILY (APEAK,PPEAK,AHI,PHI,P,D3,DAY,MONTH,YEAR)

DAY = DAY + 1

THIS IS THE END OF THE PROGRAM. LOOPING WITHIN THE SUBROUTINES ARE DONE THROUGH SUBROUTINE CALLS INSIDE SUBROUTINES.

END WHILE

THIS IS THE END OF THE PROGRAM. LOOPING WITHIN THE SUBROUTINES ARE DONE THROUGH SUBROUTINE CALLS INSIDE SUBROUTINES.

END

SUBROUTINE READER -- READS THE ENTIRE DATA FILE AND STORE THE DATA INTO BIN1 - BIN12 ARRAYS. PRINTS OUT THE DATA AT THE END OF THE SUBROUTINE TO VERIFY THE DATA

SUBROUTINE READER(IMAX)
COMMON HEADER, TRAIL, TEST, CHOTOT(7, 24)

COMMON BIN1(200), BIN2(200), BIN3(200), BIN4(200)
COMMON BIN5(200), BIN6(200), BIN7(200), BIN8(200)
COMMON BIN9(200), BIN10(200), BIN11(200), BIN12(200)
COMMON DATE(200), TIME(200)
COMMON CONFIG, TCONFIG, FILENO, TSDATE, MULTPL, MDAYZ(7, 288)
COMMON TFILEN, SITENO, TSITEN, INTVAL, TINTVL, STATUS, INDEX
COMMON TSTATS, AMPEAK(7), PMPEAK(7), AMHIGH(7), PMHIGH(7), RTIME

CHARACTER*8 HEADER, TRAIL, TEST
INTEGER P, NWTOT(7, 7, 24), CHECK, APEAK(7), PPEAK(7), AHI(7), PHI(7)
INTEGER CHECKR
INTEGER TSDATE, DATE, CHOTOT
INTEGER COUNT, INDEX, TSTIME
INTEGER CONFIG, TCONFIG, FILENO, SITENO, TSITEN
INTEGER INTVAL, TINTVL, STATUS, TSTATS, RTIME
INTEGER BIN1, BIN2, BIN3, BIN4
INTEGER BIN5, BIN6, BIN7, BIN8
INTEGER BIN9, BIN10, BIN11, BIN12, MDAYZ
INTEGER AMPEAK, PMPEAK, AMHIGH, PMHIGH, RTIME, IMAX

C READ THE HEADER LINE
C
READ(5, 201) HEADER, CONFIG, FILENO, SITENO, INTVAL, STATUS
FORMAT(A6, 1X, I2, 1X, I2, 1X, I8, 1X, I4, 1X, I2)
C
C SET UP LOOP TO READ AND STORE ALL THE DATA
C
I=1
J=1
C
READ THE DATA AND STORE INTO ARRAYS
C
READ(5, 202) DATE(I), TIME(I), BIN1(I), BIN2(I), BIN3(I), BIN4(I),
* BINS(I), BIN6(I), BIN7(I), BIN8(I), BIN9(I), BIN10(I),
* BIN11(I), BIN12(I)
C
C
C
C
FORMAT(I6, 1X, I4, 2X, I2(I4, 1X))
IF (DATE(I) .EQ. 999999 ) GOTO 1001
I=I+1
GOTO 49
C
READ(5, 201) TRAIL, TCONFIG, TFILEN, TSITEN, TINTVL, TSTATS
C
C COMPLETED READING THE ENTIRE FILE
C
IMAX=I-1
C
WRITE(6, 403) HEADER, CONFIG, FILENO, SITENO, INTVAL, STATUS
DO 13000 J=1, IMAX
WRITE(6, 402) DATE(J), TIME(J), BIN1(J), BIN2(J), BIN3(J), BIN4(J),
* BINS(J), BIN6(J), BIN7(J), BIN8(J), BIN9(J), BIN10(J),
* BIN11(J), BIN12(J)
13000 CONTINUE
C
WRITE(6, 404) TRAIL, TCONFIG, TFILEN, TSITEN, TINTVL, TSTATS
C
FORMAT('1', 10X, A6, 1X, I2, 1X, I2, 1X, I8, 1X, I4, 1X, I2, /)
SUBROUTINE PREP -- PREP REDUCES THE DATA ACCORDING TO THE TIME INTERVAL AND THE CONFIGURATION. IT CALCULATES THE MINUTE INTERVAL TOTAL, HOURLY TOTAL, RUNNING SUBTOTAL, AND PEAK HOUR TOTALS. THIS SUBROUTINE IS THE MAIN CORE OF THE ENTIRE PROGRAM.

SUBROUTINE PREP(P,NWTOT,D3,IJ,ITIME,IT,IFLAG)

FIRST = 1
CALCULATE MINUTE TOTAL AND HOURLY TOTAL

TSDATE = DATE(INDEX)
TSTIME = TIME(INDEX)
INDEX = INDEX
RTIME = &DELTA

IF (RTIME .NE. INTVAL) THEN DO
WRITE (6,21)
FORMAT (/11,110(*),/,'*** ERROR ****',/)
21
FORMAT (/*,110(*),/,'*** ERROR ****',/)
300. * T20, 'TIME INTERVAL SUPPLIED BY THE USER DOES NOT',
301. * 'MATCH THE DATA FILE INFORMATION', '//' ,','
302. * 110('*'))
303. IFLAG=1
304. RETURN
305. END IF
306. C
307. C
308. C
309. P1 = 0
310. MULTPL=60./RTIME
311. COUNT=(TSTIME/100)*MULTPL + (MOD(TSTIME,100)/RTIME)
312. IF (COUNT .EQ. 0) COUNT=1
313. WHILE (DATE(INDEX) .EQ. TSDATE) DO
314. C
315. C
316. SET UP LOOP TO CALCULATE THE MINUTE TOTALS
317. C
318. IF (CONFIG .LE. 3) THEN DO
319. MDAYZ(7,COUNT)=BIN1(INDEX)
320. MDAYZ(7,COUNT+1)=BIN2(INDEX)
321. MDAYZ(7,COUNT+2)=BIN3(INDEX)
322. MDAYZ(7,COUNT+3)=BIN4(INDEX)
323. MDAYZ(7,COUNT+4)=BIN5(INDEX)
324. MDAYZ(7,COUNT+5)=BIN6(INDEX)
325. MDAYZ(7,COUNT+6)=BIN7(INDEX)
326. MDAYZ(7,COUNT+7)=BIN8(INDEX)
327. MDAYZ(7,COUNT+8)=BIN9(INDEX)
328. MDAYZ(7,COUNT+9)=BIN10(INDEX)
329. MDAYZ(7,COUNT+10)=BIN11(INDEX)
330. MDAYZ(7,COUNT+11)=BIN12(INDEX)
331. C****************************************************
332. C* THIS SEGMENT PREPARES DATA FOR INDIVIDUAL LANE *
333. C* DAILY TIME INTERVAL REPORT. IT STORES THE *
334. C* VALUES INTO MDAYZ(7,288) WHICH COVERS ALL SIX *
335. C* LANES IN THE CASE OF SIX CHANNEL CONFIGURATION. *
336. C*
337. C*
338. C****************************************************
339. C
340. MDAYZ(FIRST,COUNT)=BIN1(INDEX)
341. MDAYZ(FIRST,COUNT+1)=BIN2(INDEX)
342. MDAYZ(FIRST,COUNT+2)=BIN3(INDEX)
343. MDAYZ(FIRST,COUNT+3)=BIN4(INDEX)
344. MDAYZ(FIRST,COUNT+4)=BIN5(INDEX)
345. MDAYZ(FIRST,COUNT+5)=BIN6(INDEX)
346. MDAYZ(FIRST,COUNT+6)=BIN7(INDEX)
347. MDAYZ(FIRST,COUNT+7)=BIN8(INDEX)
348. MDAYZ(FIRST,COUNT+8)=BIN9(INDEX)
349. MDAYZ(FIRST,COUNT+9)=BIN10(INDEX)
350. MDAYZ(FIRST,COUNT+10)=BIN11(INDEX)
351. MDAYZ(FIRST,COUNT+11)=BIN12(INDEX)
352. C
353. C
354. COUNT=COUNT+12
355. TSDATE=DATE(INDEX)
356. INDEX=INDEX+1
357. C
358. C****************************************************
359. C* DIMENSION 1 FOR SINGLE CHANNEL*
ELSE DO
  IF (CONFIG .LE. 15) THEN DO
    MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)
    MDAYZ(7,COUNT+1)=BIN3(INDEX)+BIN4(INDEX)
    MDAYZ(7,COUNT+2)=BIN5(INDEX)+BIN6(INDEX)
    MDAYZ(7,COUNT+3)=BIN7(INDEX)+BIN8(INDEX)
    MDAYZ(7,COUNT+4)=BIN9(INDEX)+BIN10(INDEX)
    MDAYZ(7,COUNT+5)=BIN11(INDEX)+BIN12(INDEX)
  ENDIF
  ELSE DO
    IF (CONFIG .EQ. 18) THEN DO
      MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)+BIN3(INDEX)+
                      BIN4(INDEX)+BIN5(INDEX)+BIN6(INDEX)
      MDAYZ(7,COUNT+1)=BIN7(INDEX)+BIN8(INDEX)+BIN9(INDEX)+
                        BIN10(INDEX)
      MDAYZ(7,COUNT+2)=BIN11(INDEX)+BIN12(INDEX)
    ENDIF
  ENDIF
  D3 = 2
  ELSE DO
    IF (CONFIG .EQ. 18) THEN DO
      MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)+BIN3(INDEX)+
                      BIN4(INDEX)+BIN5(INDEX)+BIN6(INDEX)
      MDAYZ(7,COUNT+1)=BIN7(INDEX)+BIN8(INDEX)+BIN9(INDEX)+
                        BIN10(INDEX)
      MDAYZ(7,COUNT+2)=BIN11(INDEX)+BIN12(INDEX)
    ENDIF
  ENDIF
MDAYZ(FIRST,COUNT)=BIN1(INDEX)
MDAYZ(FIRST+1,COUNT)=BIN2(INDEX)
MDAYZ(FIRST+2,COUNT)=BIN3(INDEX)
MDAYZ(FIRST+3,COUNT)=BIN4(INDEX)
MDAYZ(FIRST+4,COUNT)=BIN5(INDEX)
MDAYZ(FIRST+5,COUNT)=BIN6(INDEX)
MDAYZ(FIRST,COUNT+1)=BIN7(INDEX)
MDAYZ(FIRST+1,COUNT+1)=BIN8(INDEX)
MDAYZ(FIRST+2,COUNT+1)=BIN9(INDEX)
MDAYZ(FIRST+3,COUNT+1)=BIN10(INDEX)
MDAYZ(FIRST+4,COUNT+1)=BIN11(INDEX)
MDAYZ(FIRST+5,COUNT+1)=BIN12(INDEX)

C
C
COUNT=COUNT+2
TSDATE=DATE(INDEX)
INDEX=INDEX+1
C
C******************************************************************************************
C** DIMENSION 6 FOR MULTI-CHANNEL *
C******************************************************************************************
C
D3 = 6
C
ELSE DO
MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)+BIN3(INDEX)+
* BIN4(INDEX)+BIN5(INDEX)+BIN6(INDEX)+BIN7(INDEX)+
* BIN8(INDEX)+BIN9(INDEX)+BIN10(INDEX)+BIN11(INDEX)+
* BIN12(INDEX)
COUNT=COUNT+1
TSDATE=DATE(INDEX)
INDEX=INDEX+1
ENDIF
ENDIF
ENDIF
C
C
END WHILE
C
C SET DO LOOP TO CALCULATE THE HOURLY TOTAL VOLUME
C
NTEVAL=60/INTVAL
MAXI=24*NTEVAL
C
C
C
NTEVAL --- NUMBER OF ELEMENTS TO ADD TO FIND HOURLY TOTAL
C
C******************************************************************************************
C** LOOP THROUGH SIX TIMES TO FIND HOURLY TOTALS *
C** OF ALL THE CHANNELS FOR LATER USAGE *
C******************************************************************************************
C
DO 6000 M=1,7
NPLUS=0
L = 1
K=0
C DO 1000 I=1,MAXI,NTEVAL
   NPLUS=NPLUS+NTEVAL
   K=K+1
   DO 100 J=I,NPLUS
       CHOTOT(M,K)=CHOTOT(M,K)+MDAYZ(M,J)
   100 CONTINUE
   NWTOT(M,P,K)=CHOTOT(M,K)
1000 CONTINUE

C*****************************************************************
THE NEXT SEGMENT TO THIS ROUTINE IS TO DETERMINE IF THE
CHANNEL IS A DUAL CHANNEL, AND IF SO, IT WILL COMPUTE
THE NECESSARY HOURLY TOTALS FOR EACH OF THE CHANNELS
EXAMPLE:
NWTOT(1,K,J) - WILL CONTAIN INFORMATION FOR LANE 1*
NWTOT(2,K,J) - WILL CONTAIN INFORMATION FOR LANE 2*
C*****************************************************************
C** ******************************************************************
SUBROUTINE DAILY(APEAK,PPEAK,AHI,PHI,P,D3,DAY,MONTH,YEAR)
COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
COMMON DATE(200),TIME(200)
COMMON CONFIG,TCONFIG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
COMMON TFILEN,SITENO,TSITEN,TINTVL,TSTATS,AHI,PHI,P,TIME
DIMENSION NUAMHI(7),NUPMHI(7)
CHARACTER*8 HEADER,TRAIL,TEST
CHARACTER*2 CAMMIN(7),CPMMIN(7)
INTEGER NWTOT(7,7,24)
INTEGER TSDATE,DATE,CHOTOT
INTEGER COUNT,INDEX,TSTIME
INTEGER CONFIG,TCONFIG,FILENO,TFILEN,SITENO,TSITEN
INTEGER TINTVL,TSTATS,AHI,PHI,P,TIME
SUBROUTINE DAILY(APEAK,PPEAK,AHI,PHI,P,D3,DAY,MONTH,YEAR)
C*****************************************************************
C** ******************************************************************
C*****************************************************************
INTEGER BIN1,BIN2,BIN3,BIN4
INTEGER BIN5,BIN6,BIN7,BIN8
INTEGER BIN9,BIN10,BIN11,BIN12,MDAYZ
INTEGER AMPEAK, PMPEAK, AMHIGH, PMHIGH,RTIME
INTEGER APEAK(7),PPEAK(7),AHI(7),PHI(7),P
INTEGER HILIM,HIGH,AMINDX(7),PMINDX(7)
INTEGER RUNSUB(7,24),T,B,D3,DAY,MONTH,YEAR
INTEGER AMHOUR(7),PMHOUR(7),AMMIN(7),PMMIN(7)
DATA CAMMIN/7*'99'/, CPMMIN /7*'99'/
DO 6 K=1,7
    DO 5 T = 1,24
        RUNSUB(K,T) = 0
    CONTINUE
7 CONTINUE
DO 601 N1=1,7
    RUNSUB(N1,1)=CHOTOT(N1,1)
602 CONTINUE
DO 602 N2=2,24
    RUNSUB(N1,N2)=RUNSUB(N1,N2-1) + CHOTOT(N1,N2)
601 CONTINUE
K3=1
K4=1
CONTINUE
C CALCULATE THE AM-PEAK AND PM-PEAK VOLUME
C CALCULATE THE AM-HIGH AND PM-HIGH VOLUME
C
C******************************************************************************
C** ZERO OUT THE ARRAYS TO HOLD THE PEAK VOLUMES  **
C******************************************************************************
DO 700 J1=1,7
    AMHIGH(J1)=0
    PMHIGH(J1)=0
    AMPEAK(J1)=0
    PMPEAK(J1)=0
700 CONTINUE
C SET UP LOOPS TO CALCULATE AMPEAK, PMPEAK, AMHIGH AND PMHIGH VOLUMES
LOW=4*MULTPL
HIGH=11*MULTPL
LOWX2 = 12 * MULTPL

LOW -- LOWER BOUND FOR THE LOOP TO CALCULATE VALUES
HIGH - HIGHER"

DO 35 IT = 1, 7
  AMINDX(IT) = 0
  PMINDX(IT) = 0
  AMHOUR(IT) = 0
  AMMIN(IT) = 0
  PMHOUR(IT) = 0
  PMMIN(IT) = 0

DO 33 II = LOW, HIGH
C INITIALIZE NUAMHI AND NUPMHI TO ZEROS
NUAMHI(IT) = 0
NUPMHI(IT) = 0

DO 34 I2 = 1, MULTPL
  NUAMHI(IT) = NUAMHI(IT) + MDAYZ(IT, II + I2)
  NUPMHI(IT) = NUPMHI(IT) + MDAYZ(IT, LOWX2 + I2)
CONTINUE

CHECK TO EXCHANGE THE HIGHER VALUES TO AMHIGH AND PMHIGH
IF (NUAMHI(IT) .GT. AMHIGH(IT)) THEN DO
  AMHIGH(IT) = NUAMHI(IT)
  AMINDX(IT) = II
ENDIF

IF (NUPMHI(IT) .GT. PMHIGH(IT)) THEN DO
  PMHIGH(IT) = NUPMHI(IT)
  PMINDX(IT) = LOWX2
ENDIF

LOWX2 = LOWX2 + 1
INCREMENT LOWX2 TO ADD THE NUMBERS TO CALCULATE PMHIGH
CONTINUE

RESET LOWX2 BACK TO ORIGINAL VALUE FOR FURTHER PRINT OUT
LOWX2 = 12 * MULTPL

CONTINUE

FIND THE AM AND PM PEAK VOLUME STARTING TIME

DO 751 Nl = 1, 7
  AMHOUR(Nl) = AMINDX(Nl) / MULTPL
  AMMIN(Nl) = MOD(AMINDX(Nl), MULTPL) * INTVAL
ENDIF

CONVERT TO THE CHARACTER SET FOR MINUTES LESS THAN 10
IF (AMMIN(N1) .LT. 10) THEN DO
  IF (AMMIN(N1) .EQ. 0) CAMMIN(N1)='00'
  IF (AMMIN(N1) .EQ. 1) CAMMIN(N1)='01'
  IF (AMMIN(N1) .EQ. 2) CAMMIN(N1)='02'
  IF (AMMIN(N1) .EQ. 3) CAMMIN(N1)='03'
  IF (AMMIN(N1) .EQ. 4) CAMMIN(N1)='04'
  IF (AMMIN(N1) .EQ. 5) CAMMIN(N1)='05'
  IF (AMMIN(N1) .EQ. 6) CAMMIN(N1)='06'
  IF (AMMIN(N1) .EQ. 7) CAMMIN(N1)='07'
  IF (AMMIN(N1) .EQ. 8) CAMMIN(N1)='08'
  IF (AMMIN(N1) .EQ. 9) CAMMIN(N1)='09'
ENDIF

IF (PMMIN(N1) .LT. 10) THEN DO
  IF (PMMIN(N1) .EQ. 0) CPMMIN(N1)='00'
  IF (PMMIN(N1) .EQ. 1) CPMMIN(N1)='01'
  IF (PMMIN(N1) .EQ. 2) CPMMIN(N1)='02'
  IF (PMMIN(N1) .EQ. 3) CPMMIN(N1)='03'
  IF (PMMIN(N1) .EQ. 4) CPMMIN(N1)='04'
  IF (PMMIN(N1) .EQ. 5) CPMMIN(N1)='05'
  IF (PMMIN(N1) .EQ. 6) CPMMIN(N1)='06'
  IF (PMMIN(N1) .EQ. 7) CPMMIN(N1)='07'
  IF (PMMIN(N1) .EQ. 8) CPMMIN(N1)='08'
  IF (PMMIN(N1) .EQ. 9) CPMMIN(N1)='09'
ENDIF

PMHOUR(N1)=PMINDX(N1)/MULTPL
PMMIN(N1) =(MOD(PMINDX(N1),MULTPL)*INTVAL)

CONTINUE

SET UP LOOP TO FIND AMPEAK AND PMPEAK VALUES
BY SIMPLY ADDING UP THE VALUES OF 6:30-9:30 AND
15:30-18:30 RESPECTIVELY.

LOWLIM = INT(6.5*MULTPL)
MAXLIM=3*MULTPL
HILIM= INT(15.5*MULTPL)

DO 46 I4=1,7
DO 43 I3=1,MAXLIM
  AMPEAK(I4)=AMPEAK(I4)+MDAYZ(I4,LOWLIM+I3)
  PMPEAK(I4)=PMPEAK(I4)+MDAYZ(I4,HILIM+I3)
CONTINUE

CONTINUE

PRINT OUT THE SUMMARY PORTION OF THE REPORT

APEAK(P) = AMPEAK
PPEAK(P) = PMPEAK
AH1(P) = AMHIGH
C PHI(P) = PMHIGH

C***********************************************************************
C** START LOOP TO PRINT OUT SIX DAILY REPORTS. ***
C***********************************************************************

C I6=1

WHILE(I6 .LT. 7 ) DO

C CHECK FOR AM AND PM PEAK VOLUMES TO PRINT OUT THE NEXT
C CHANNEL REPORT. IF THE PEAK VOLUMES ARE ZERO THEN DO
C NOT PRINT OUT THE NEXT CHANNEL.

C IF (RUNSUB(I6,24) .NE. 0) THEN DO

WRITE(6,11)

11 FORMAT('1')

WRITE(6,21)

21 FORMAT(' ')

WRITE(6,31)

31 FORMAT(T40,'FREEWAY VOLUME COUNT - TEXAS TRANSPORTATION',

41 FORMAT( ' INSTITUTE','//)

42 WRITE(6,41)MONTH,DAY,YEAR,SITENO,I6,63

43 FORMAT(T10,'DATE:',3X,I2,'-',I2,'-',I4,T50,'SITE NO:',

53 FORMAT(3X,'COUNTER:',3X,'&COUNTER','//,

54 T10,'WEATHER:',3X,'&WEATHER','//,

55 T50,'DIRECTION OF FLOW:',3X,'&DIR','//,

56 T10,'LOCATION:',3X,

57 '&LOC','//,

58 T10,'SOURCE :',10X,'THIS IS ',I1,' OF ',I1,'//)

C

C

WRITE(6,51)

51 FORMAT(T5,125(''))

WRITE(6,61)

61 FORMAT(T26,'HOURLY',T57,'HOURLY',T88,'HOURLY',T119,'HOURLY')

WRITE(6,71)

71 FORMAT(4(9X,'TIME',4X,'VOLUME',2X,'VOLUME'))

WRITE(6,51)

72 FORMAT(4(9X,'TIME',4X,'VOLUME',2X,'VOLUME'))

WRITE(6,56)

56 FORMAT(' ',T5,125(''))

KOUNT = 1

COL1 = 0.0

COL2 = 6.0

COL3 = 12.0

COL4 = 18.0

TDELTA = 0.0

C START A LOOP TO COUNT THE TIMES AND PRINT OUT MINUTE TOTALS

C

C K3 WILL BE USED TO WRITE OUT THE MINUTE & HOURLY TOTALS

C

COL42 = 0.0

78 WHILE (COL42 .LE. (24.0-Delta)) DO
C IF COL42 .GE. 24. THEN END OF THAT DAY HAS REACHED
C
C KEEP COUNTER FOR PAGE EJECT IN CASE OF TIME INTERVAL 15 MINUTES
C
KOUNT = KOUNT+1
C
TDELTA = DELTA + TDELTA
COL12 = COL1 + DELTA
COL22 = COL2 + DELTA
COL32 = COL3 + DELTA
COL42 = COL4 + DELTA

IF (TDELTA .GE. (0.6-DELTA)) THEN DO
  COL12 = COL12 + 0.4
  COL22 = COL22 + 0.4
  COL32 = COL32 + 0.4
  COL42 = COL42 + 0.4
ENDIF
C
C CHOOSE WHICH INFORMATION TO PRINT OUT AND CORRESPONDING FORMAT

C IF (MOD((K3-(MULTPL-1)),MULTPL) .EQ. 0) THEN DO
C PRINT OUT MINUTE TOTAL AND HOURLY TOTAL
WRITE(6,93) COL1, COL12, MDAYZ(I6,K3),CHOTOT(I6,K4),
          COL2,COL22,MDAYZ(I6,K3+COLTWO),CHOTOT(I6,K4+6),COL3,
          COL32,MDAYZ(I6,K3+COLTRE),CHOTOT(I6,K4+12),COL4,COL42,
          MDAYZ(I6,K3+COLFOR),CHOTOT(I6,K4+18)
K4=K4+1
IF (K4 .GT. 6) K4=1
ELSE DO
C****************************************************************
C** PRINT OUT THE MINUTE INTERVALS WITHOUT THE HOURLY VOLUME *
C***********************************************************************************
WRITE(6,94) COL1,COL12,MDAYZ(I6,K3),COL2,COL22,
            MDAYZ(I6,K3+COLTWO),COL3,COL32,MDAYZ(I6,K3+COLTRE),
            COL4,COL42,MDAYZ(I6,K3+COLFOR)
ENDIF
C
C PAGE CONTROL FOR LONG REPORTS.
C
C91 FORMAT(T5,4(F5.2,'-',F5.2,I6,19,5X))
93 FORMAT(T5,4(F5.2,'-',F5.2,I6,14X))

840. COL3 = COL32
841. COL4 = COL42
842. IF (TDELTA .GT. (0.6-DELTA)) THEN DO
843. TDELTA=0.0
844. COL1 = IFIX(COL1+1.0)
845. COL2 = IFIX(COL2+1.0)
846. COL3 = IFIX(COL3+1.0)
847. COL4 = IFIX(COL4+1.0)
848. C
849. WRITE(6,56)
850. C
851. C
852. WRITE(6,95) RUNSUB(I6,B),RUNSUB(I6,B+6),RUNSUB(I6,B+12),
853. * RUNSUB(I6,B+18)
854. C
855. 95 FORMAT('+',T5,22X,'(',I5,')','24X','(',I5,')','24X,'(
856. * '(',I5,')','24X','(',I5,')')
857. B = B + l
858. IF (B .GT. 6) B=1
859. C
860. C
861. C CHECK FOR PAGE EJECT AND LINE CONTROL FOR LONG REPORTS
862. C
863. IF ((45-KOUNT) .LT. 15) THEN DO
864. KOUNT=1
865. C
866. C IF END-OF-DAY THEN DO NOT SKIP TO NEW PAGE. PRINT TOTALS.
867. C
868. IF (COL4 .LT. 24) THEN DO
869. C
870. WRITE(6,10)
871. C
872. WRITE(6,56)
873. ENDIF
874. ENDIF
875. ENDIF
876. K3=K3+1
877. C
878. C
879. C PRINT,'K4=',K4
880. END WHILE
881. C
882. C
883. C
884. C
885. WRITE(6,101) AMPEAK(I6),PMPEAK(I6)
886. 101 FORMAT(//,T10,'AM PEAK (6:30 - 9:30) VOLUME:',I10,
887. * T60,'PM PEAK (3:30 - 6:30) VOLUME:',I10,
888. * T10,'DT:',10('_'))
889. WRITE(6,102) RUNSUB(I6,24)
890. 102 FORMAT('+',T115,I5)
891. C
892. C
893. C
894. WRITE(6,111)AMHIGH(I6),PMHIGH(I6)
895. 111 FORMAT(T10,'AM PEAK HIGHEST HOUR:',8X,I10,5X,T60,
896. * 'PM PEAK HIGHEST HOUR:',8X,I10,5X)
897. C
898. C
899. 112 FORMAT(T10,'(NEAREST',I3,' MINUTE COUNT)',T60,
900. * '(NEAREST', I3, ' MINUTE COUNT')')
901. C
902. C
903. C PRINT OUT THE STARTING TIME DEPENDING ON THE MINUTE COUNT
904. C
905. C
906. C IF ((AMMIN(I6) .LT. 10) .AND. (PMMIN(I6) .LT. 10)) THEN DO
907. WRITE(6,114) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),CPMMIN(I6)
908. C
909. C
910. 114 FORMAT(/,T10,2('STARTING TIME :',18X,I3,'%A2,11X))
911. ELSE DO
912. C
913. C
914. C IF ((AMMIN(I6) .LT. 10) .OR. (PMMIN(I6) .LT. 10)) THEN DO
915. IF (AMMIN(I6) .LT. 10) THEN DO
916. WRITE(6,115) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),PMMIN(I6)
917. 115 FORMAT(/,T10,'STARTING TIME :',18X,I3,'%A2,11X,
918. * 'STARTING TIME :',18X,I3,'%A2)
919. ELSE DO
920. C
921. C
922. C WRITE(6,116) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),CPMMIN(I6)
923. 116 FORMAT(/,T10,'STARTING TIME :',18X,I3,'%A2,11X,
924. * 'STARTING TIME :',18X,I3,'%A2)
925. END IF
926. C
927. C
928. C ELSE DO
929. C
930. C WRITE(6,113) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),CPMMIN(I6)
931. 113 FORMAT(/,T10,2('STARTING TIME :',18X,I3,'%A2,11X))
932. END IF
933. C
934. C
935. C
936. C
937. C
938. C
939. C
940. C RESET K3 FOR NEXT CHANNEL PRINT OUT
941. C
942. C
943. C
944. C
945. C
946. C
947. C
948. C
949. C
950. C
951. C
952. C
953. C
954. C CHECK TO SEE IF END HAS REACHED
955. C
956. C
957. C
958. 10 FORMAT('1')
959. RETURN
SUBROUTINE INITAL -- SUBROUTINE TO INITIALIZE THE ARRAY HTOTAL AND MINTOT THAT ACCUMULATES THE HOURLY AND MINUTE INTERVAL TOTALS.

SUBROUTINE INITAL(MDAYZ,CHOTOT)
DIMENSION MDAYZ(7,288)
INTEGER CHOTOT(7,24)

DO 80 K=1,7
    DO 20 J=1,24
    CHOTOT(K,J)=0
    CONTINUE
80 CONTINUE

C****************************************************
ZERO OUT MDAYZ AFTER EACH DAY OF PREPARATION **
C****************************************************

DO 40 I=1,7
    DO 50 J=1,288
        MDAYZ(I,J)=0
    CONTINUE
50 CONTINUE

RETURN
END

SUBROUTINE DAWEEK -- THIS SUBROUTINE FINDS THE DAY OF THE WEEK GIVEN THE INITIAL DATE IN THE FORM OF YY-MM-DD. IT DETERMINES THE PROPER ELEMENT TO STORE THE

SUBROUTINE DAWEEK(TSDATE,ENDEX,ARMON,ARDAY,CHECK,DAY,MONTH,YEAR)
CHARACTER*9 DATES(7),DAYOWK
INTEGER DATE,TSDATE,DAY,MONTH,YEAR
INTEGER ENDEX,D3,FACTOR,CHECK
INTEGER ARMON(7),ARDAY(7)

C INITIALIZE THE DATES ARRAY

DO 21 I = 1, 7
    ARMON(I) = 0
    ARDAY(I) = 0
21    CONTINUE

DATES(1) = 'SATURDAY'
DATES(2) = 'SUNDAY'
DATES(3) = 'MONDAY'
DATES(4) = 'TUESDAY'
DATES(5) = 'WEDNESDAY'
DATES(6) = 'THURSDAY'
DATES(7) = 'FRIDAY'

C FIND THE DAY, MONTH AND YEAR FROM THE GIVEN TEST DATE
C
DAY = (MOD(TSDATE,100))
MONTH = (MOD(TSDATE,10000) - DAY)/100
YEAR = (TSDATE/10000) + 1900
C PRINT OUT DAY, MONTH, AND YEAR
PRINT,'DAY=',DAY,'MONTH=',MONTH,'YEAR=',YEAR
C CONVERSION OF THE DATE INTO ABSOLUTE JULIAN DATES
C IF MONTH IS EITHER JAN. OR FEB., USE THE FOLLOWING FORMULA.
IF (MONTH .LE. 2) THEN DO
    FACTOR = 365*YEAR + DAY + 31*(MONTH-1) +
    IFIX((YEAR-1)/4.) - IFIX(3/4.*(IFIX((YEAR-1)/100.)+1))
ELSE DO
    FACTOR = 365*YEAR+DAY+31*(MONTH-1) - IFIX(0.4*MONTH+2.3)
    +IFIX(YEAR/4.) - IFIX(3/4.*(IFIX(YEAR/100.)+1))
END IF
C FOR ALL OTHER MONTHS, USE THE FOLLOWING FORMULA.
C
C FIND THE INDEX FOR THE ARRAY DATES TO CONVERT TO DATES
ENDEX = FACTOR + (IFIX(-FACTOR/7.)*7)+1
DAYOWK = DATES(ENDEX)
C PRINT OUT THE DAY OF WEEK
PRINT, 'DAY OF WEEK = ', DATES(ENDEX)
CALL A PROGRAM TO DETERMINE 7-DAY WEEK

CALL FINDAY(DAYWK, DAY, MONTH, YEAR, ARMON, ARDAY, CHECK)

IF (ENDEX .GE. 3) THEN DO
   ENDEX = ENDEX - 3
ELSE DO
   ENDEX = ENDEX + 4
ENDIF

RETURN
END

SUBROUTINE WEEKLY
-- SUBROUTINE TO PRODUCE THE WEEKLY
SUMMARY REPORT. THIS REPORT INCLUDES THE HOURLY TOTAL
VOLUMES, SUMMARY OF WEEKLY DATA. IT CAN PRODUCE UP TO
SIX REPORTS FOR THE 6-CHANNEL CONFIGURATION.

SUBROUTINE WEEKLY(ARMON, ARDAY, NWTOT, APEAK, PPEAK, AHI, PHI, D3, SITENO, TESDAY)

INTEGER ARMON(7), ARDAY(7), SITENO, TESDAY
INTEGER NWTOT(7, 7, 24), DAY, MONTH, YEAR
INTEGER APEAK(7), PPEAK(7), AHI(7), PHI(7)
INTEGER L, D3, IDTOT(7), ITOTAL(7)

C*** THIS WILL HELP PRINT THE NECESSARY AMOUNT OF TABLES*

C*** FIND OUT THE MONTH, DAY, AND YEAR FROM THE GIVEN DATE

DAY = MOD(TESDAY, 100)
MONTH = (MOD(TESDAY, 10000) - DAY) / 100
YEAR = (TESDAY / 10000) + 1900
CALCULATE THE TOTAL VALUES OF DAILY HOURLY VOLUMES

L=1

WHILE (L .LT. 7 ) DO

SET UP LOOP TO PRINT OUT THE WEEKLY SUMMARY REPORT

ITOTAL(L)=0

DO 10009 K=1,7
    IDTOT(K)=0
    DO 10010 IL=1,24
        IDTOT(K)=IDTOT(K)+NWTOT(L,K,IL)
    CONTINUE
    ITOTAL(L) = ITOTAL(L) + IDTOT(K)
    CONTINUE

IF IT IS A TWO CHANNEL, THEN CLEAR THE PREVIOUS PEAK Periods

IF (D3 .EQ. 2) THEN
    DO 15000 J=1,7
        APEAK(IK) = 0
        PPEAK(IK) = 0
        AHI(IK) = 0
        PHI(IK) = 0
    CONTINUE
END IF

IF IT IS A TWO CHANNEL TABLE, THIS WILL HELP PRINT
OUT THE NECESSARY AM & PM PEAK PERIODS.
DO 20000 I =1,24
CITEMP = NWTOT(L,J,I)
C IF (I .LE. 12) THEN DO
C IF ((I .GT. 6) .AND. (I .LE. 9)) APEAK(J) = APEAK(J) + ITEMP
C IF (ITEMP.GT. AHI(J)) AHI(J) = ITEMP
C ELSE DO
C IF (ITEMP.GT. PHI(J)) PHI(J) = ITEMP
C END IF
C 20000 CONTINUE
C 15000 CONTINUE
C END IF
C
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************
C****************************************************************

C**START A LOOP TO PRINT OUT THE WEEKLY REPORTS. IF ALL OF
C**THE ENTRIES ARE ZEROS, DO NOT PRINT OUT THAT REPORT. *
C**
C**
C** C** IF (ITOTAL(L) .NE. 0) THEN DO
C**
C** WRITE(6,10) FORMAT('1')
C**
C** WRITE(6,20) MONTH,DAY,YEAR
C**
C** FORMAT(///,'T48,'FREEWAY VOLUME COUNT SUMMARY SHEET',///,
C** 'T15,'LOCATION :',5X,'&LOC','
C** '/,T15,'DATE :','9X,I2,'-',I2,'-',I4,T50,
C** 'T50,'DIRECTION OF FLOW :','5X,'&DIR','
C** '
C**
C** WRITE(6,35)L,D3,SITENO
C**
C** FORMAT(T15,'SOURCE :',T30,'THIS IS ',I2, ' OF ',I2,
C** 'T50,'SITE NO :','5X,I8,///)
C**
C** WRITE(6,40)
C**
C** FORMAT(T15,'DAY OF WEEK :',T40,'MONDAY',T50,'TUESDAY',
C** 'T60,'WEDNESDAY',T70,'THURSDAY',T80,'FRIDAY',
C** 'T90,'SATURDAY',T100,'SUNDAY',///)
C**
C** WRITE(6,41) ARMON(1),ARDAY(1),ARMON(2),ARDAY(2),
C** ARMON(3),ARDAY(3),ARMON(4),ARDAY(4),ARMON(5),ARDAY(5),
C** ARMON(6),ARDAY(6),ARMON(7),ARDAY(7)
C**
C** FORMAT(T39,I2, ' ',I2,T50,I2,' ',I2,T60,I2,' ',I2,T70,I2,
C** ' ',I2,T99,I2,' ',I2,//)

51
C
1261. WRITE(6,45)
1262. 45 FORMAT(T15,93(' '))
1263. WRITE(6,50)
1264. 50 FORMAT(T20,' TIME',T40,'HOURLY',T50,'HOURLY',T60,'HOURLY',
1265. * T70,'HOURLY',T80,'HOURLY',T90,'HOURLY',T100,
1266. * 'HOURLY')
1267. WRITE(6,60)
1268. 60 FORMAT(T40,'VOLUME',T50,'VOLUME',T60,'VOLUME',T70,
1269. * 'VOLUME',T80,'VOLUME',T90,'VOLUME',T100,'VOLUME')/
1270. N=0
1271. C
1272. PRINT,'NWTOT =',NWTOT(1,1),NWTOT(2,1),NWTOT(3,1),NWTOT(4,1)
1273. WHILE(N.LT. 24) DO
1274.  N=N+1
1275.  K = 3
1276.  WRITE(6,70) N,NONE,NWTOT(L,1,NONE),NWTOT(L,2,NONE),
1277.  * NWTOT(L,3,NONE),NWTOT(L,4,NONE),NWTOT(L,5,NONE),
1278.  * NWTOT(L,6,NONE), NWTOT(L,7,NONE)
1279.  70 FORMAT(T20,I2,'-',I2,T42,7(I4,6X))
1280. N=N+1
1281. END WHILE
1282. WRITE(6,46)
1283. 46 FORMAT(' ')
1284. WRITE(6,45)
1285. 45 FORMAT(' ')
1286. WRITE(6,80) APEAK(1),APEAK(2),APEAK(3),APEAK(4),APEAK(5),
1287.  * APEAK(6),APEAK(7)
1288.  80 FORMAT(' ',T39,7(3X,I4,3X))
1289. WRITE(6,85)
1290.  85 FORMAT('+',T15,'AM PEAK (6-9) VOLUME:',T38,7(3X,5(' '),2X))
1291. WRITE(6,90) AHI(1),AHI(2),AHI(3),AHI(4),AHI(5),
1292.  AHI(6),AHI(7)
1293.  90 FORMAT(' ',T39,7(3X,I4,3X))
1294. WRITE(6,95)
1295.  95 FORMAT('+',T15,'AM PEAK HIGHEST HOUR:',T38,7(3X,5(' '),2X))
1296. WRITE(6,100) PPEAK(1),PPEAK(2),PPEAK(3),PPEAK(4),PPEAK(5),
1297.  * PPEAK(6),PPEAK(7)
1298.  100 FORMAT(' ',T39,7(3X,I4,3X))
1299. WRITE(6,105)
1300.  105 FORMAT('+',T15,'PM PEAK (3-6) VOLUME:',T38,7(3X,5(' '),2X))
1301. WRITE(6,110) PHI(1),PHI(2),PHI(3),PHI(4),PHI(5),
1302.  PHI(6),PHI(7)
1303.  110 FORMAT(' ',T39,7(3X,I4,3X))
1304. WRITE(6,115)
1305.  115 FORMAT('+',T15,'PM PEAK HIGHEST HOUR:',T38,7(3X,5(' '),2X))
1306. WRITE(6,120)
1307.  120 FORMAT(' ',T15,'DT:',T38,7(3X,5(' '),2X))
1308. C
1309.  125 FORMAT('+',T38,7(3X,I5,2X))
1310. C
1311. C
1312. C
1313. C
1314. C
1315. ELSE DO
1316.  L=L+1
1317. ENDIF
1318. C
1319. C
1320. L=L+1
1321. ELSE DO
SUBROUTINE 'FINDAY' TO FIND 7-DAY WEEK

THIS SUBROUTINE INPUTS THE DAY, MONTH, YEAR OF THE CURRENT WEEK OF THE DATA COLLECTION. IT THEN ATTEMPTS TO DETERMINE THE NUMBER OF THE DAY OF THAT WEEK. IN DOING SO THE SUBROUTINE NEEDS TO KNOW WHAT MONTH IT IS SO THAT IT CAN KNOW HOW MANY DAYS IN IT.

SUBROUTINE FINDAY(DAYOWK, DAY, MONTH, YEAR, ARMON, ARDAY, CHECK)

INTEGER DAY, MONTH, YEAR
CHARACTER*(9) DAYOWK
INTEGER NWTOT(7,7,24)
INTEGER DAYZ(12), CHECK, CHECK1, CHECK2, MON1, MON2, ARMON(7), ARDAY(7)

C***********************************************************
C***********************************************************

C***********************************************************
C***********************************************************

THE FOLLOWING STATEMENTS WILL DETERMINE HOW MANY DAYS ARE IN THE RESPECTIVE MONTHS OF THE YEAR AND PLACE IN THEIR RESPECTIVE ARRAYS.

C***********************************************************
C***********************************************************

DAYZ(1) = 31
DAYZ(2) = 28
DAYZ(3) = 31
DAYZ(4) = 30
DAYZ(5) = 31
DAYZ(6) = 30
DAYZ(7) = 31
DAYZ(8) = 31
DAYZ(9) = 30
DAYZ(10) = 31
DAYZ(11) = 30
DAYZ(12) = 31
IF (MOD(YEAR, 4) .EQ. 0) DAYZ(2) = 29

C***********************************************************
C***********************************************************

THE FOLLOWING USES A 'CHECK' TO FIND OUT WHAT DAY HAS BEEN USED AS INPUT. (EX. MONDAY = 1, TUESDAY = 2.)

C***********************************************************
C***********************************************************

IF (DAYOWK .EQ. 'MONDAY') CHECK = 1
IF (DAYOWK .EQ. 'TUESDAY') CHECK = 2
IF (DAYOWK .EQ. 'WEDNESDAY') CHECK = 3
IF (DAYOWK .EQ. 'THURSDAY') CHECK = 4
IF (DAYOWK .EQ. 'FRIDAY') CHECK = 5
IF (DAYOWK .EQ. 'SATURDAY') CHECK = 6
IF (DAYOWK .EQ. 'SUNDAY') CHECK = 7
CHECK1 = CHECK
CHECK2 = CHECK - 1
MON1 = MONTH
MON2 = MONTH
IDAY1 = DAY
IDAY2 = DAY

WHILE(CHECK1 .GT. 0) DO
  ARMON(CHECK1) = MON1
  ARDAY(CHECK1) = IDAY1
  CHECK1 = CHECK1 - 1
  IDAY1 = IDAY1 - 1
  IF(IDAY1 .EQ. 0) THEN
    IF(MON1 .EQ. 1) MON1 = 13
  END IF
END WHILE

WHILE(CHECK2 .LT. 7) DO
  CHECK2 = CHECK2 + 1
  ARMON(CHECK2) = MON2
  ARDAY(CHECK2) = IDAY2
  IDAY2 = IDAY2 + 1
  IF(IDAY2 .GT. DAYZ(MON2)) THEN
    IF(MON2 .EQ. 12) MON2 = 0
    MON2 = MON2 + 1
    IDAY2 = 1
  END IF
END WHILE

IF(ARMON(CHECK2) .NE. MON1) THEN
  END IF
ENDIF

RETURN
END
APPENDIX D
The following pages are taken from the MK3 AUTOMATIC TRAFFIC COUNTING EQUIPMENT USER MANUAL. This section describes the procedures to set up interface signals. The user may find more ASCII codes from the IBM System/370 Reference card. Any ASCII character set can be used as EOL and EOF signals. The user simply enters the decimal equivalent of the ASCII character set during the interface signal setting procedure.
3.3 Interface

This mode is used to programme the RS-232 interface to be compatible with the computer or off-line terminal which is to be used with the Retriever.

There are 4 functions available to initialise the interface parameters and there is one function available to allow the user to test the interface. These five functions correspond to 5 code numbers. The code number is displayed on the left hand digit of the display. The 6 right hand digits are used as a parameter field. To change the current code number, press the appropriate key in the range 1-5. Pressing a key outside this range will result in Interface 1 being selected.

The parameters displayed for each code are:

<table>
<thead>
<tr>
<th>Interface Code</th>
<th>Parameters Displayed</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Retriever Status; Battery Voltage</td>
<td>C S V.VV</td>
</tr>
<tr>
<td>2</td>
<td>Baud Rate</td>
<td>C BBBBBB</td>
</tr>
<tr>
<td>3</td>
<td>No. of End of Line and End of File Characters</td>
<td>C LL FF</td>
</tr>
<tr>
<td>4</td>
<td>Code for Autostop option setting</td>
<td>C A</td>
</tr>
<tr>
<td>5</td>
<td>Code for Status Line option setting</td>
<td>C S</td>
</tr>
</tbody>
</table>

C = Interface Code Number  
S = Status Code  
V = Battery Voltage  
B = Baud rate  
L = End of Line characters  
F = End of File characters  
A = Autostop Code  
S = Status Line Code

Following is a description of the 5 Interface codes.

3.3.1 Interface 1 - Display Retriever status and battery voltage:

This function has three characteristics – it displays the present Retriever Status and battery voltage, and it can be instructed to output a test message to a computer or off-line terminal via the RS-232 Interface.

Select Interface 1 by pressing key "1", and the display will show 3 numbers. The left hand digit shows the Interface code selected, in this case 1. The digit in the centre of the display indicates the status of the Retriever. If this digit is 0, the Retriever is O.K., otherwise a fault condition is indicated by a non-zero number. Status is reset when the data files are cleared.

The battery voltage is shown on the right hand side of the display. When the battery is fully charged this should be in excess of 6.30 volts with the charger disconnected. As the battery discharges, this voltage will reduce towards 5.70 volts over a period of 6 to 8 days. At 5.50 volts a cut-out circuit will switch the power to the Retriever off (see section 1.3.1).

WARNING: When the power to the Retriever is switched off, ALL Data will be lost. For this reason it is suggested that the GR 0308 Charger/Interface unit is connected to the Retriever whenever possible. (It should be noted that the
Interface parameters, as described in the following sections, will not be lost when power to the Retriever is switched off).

Once the Interface parameters 2-5 have been initialised, (see the following sections), the user can check the RS-232 interface is matched to the computer or off-line terminal being used.

With Interface 1 selected, and the terminal or computer connected, depress "*" and "#" simultaneously. The Retriever will output repeatedly a test message of numbers and characters. Any End of Line characters (as described in Section 3.3.3) will follow each line, and transmission will stop at the end of each line if the Autostop option has been enabled (see Section 3.3.4) When the "*" and "#" are released, the Retriever will complete output of the current test message and End of Line characters, and it will output the End of File characters (see Section 3.3.3) followed by the End of Line characters again. During this output the Retriever will respond to any status line control as described in Section 3.3.5.

3.3.2 Interface 2 - Baud Rate

Select Interface code number 2. The right hand digits show the current Retriever baud rate. To alter the baud rate press the "*" and "#" simultaneously, and when the display shows 5 dashes enter the required baud rate, (with leading zeros), followed by the 'enter' key. The Retriever will confirm the baud rate selected on the display.

The available baud rates are:

19200  9600  4800  2400  1200  600  300  110

Entry of a baud rate other than shown above will result in the nearest available baud rate being selected.

The average transmission rate will be less than the baud rate, whilst the Retriever calculates data for output. This delay will be approximately 5-500 milliseconds at start of each line of output, proportional to the recording interval of the data file being output. In format 2, longer delays may occur (up to 15 seconds) whilst the output is being compiled.

3.3.3 Interface 3 - End of Line and End of File characters

The operator can specify up to 20 characters to be added to the end of each line of output in the Output Data mode, and up to 20 characters to be added at the end of each File in the Output Data mode.

Select Interface Code Number 3; the centre two digits show the present number of characters transmitted at the end of each line, and the right hand two digits show the present number of characters transmitted at the end of each file. To alter the current E.O.L. (End of Line) and E.O.F. (End of File) character strings, depress "*" and "#" simultaneously, and when:
ASCII code for first EOL character + "ENTER"
ASCII code for second EOL character + "ENTER"

ASCII code for last EOL character + "ENTER"
999 + "ENTER" (terminating code)
ASCII code for first EOF character + "ENTER"
ASCII code for second EOF character + "ENTER"

ASCII code for last EOF character + "ENTER"
999 + "ENTER" (terminating code)

For example:

For the E.O.L. characters to be CR, LF and six nulls, and the E.O.F. characters to be four asterisks then:

<table>
<thead>
<tr>
<th>Display shows</th>
<th>Press Keys</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 6.40</td>
<td>3</td>
<td>Select Interface 3 function</td>
</tr>
<tr>
<td>3 x y</td>
<td>* &amp; #</td>
<td>To change stored characters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x and y are previous values)</td>
</tr>
<tr>
<td>013 + ENTER</td>
<td>Load ASCII Carriage Return</td>
<td></td>
</tr>
<tr>
<td>010 + ENTER</td>
<td>Load ASCII Line Feed</td>
<td></td>
</tr>
<tr>
<td>000 + ENTER</td>
<td>Load ASCII Null</td>
<td></td>
</tr>
<tr>
<td>000 + ENTER</td>
<td>Load ASCII Null</td>
<td></td>
</tr>
<tr>
<td>000 + ENTER</td>
<td>Load ASCII Null</td>
<td></td>
</tr>
<tr>
<td>000 + ENTER</td>
<td>Load ASCII Null</td>
<td></td>
</tr>
<tr>
<td>999 + ENTER</td>
<td>Terminate EOL / Commence EOF</td>
<td></td>
</tr>
<tr>
<td>042 + ENTER</td>
<td>Load ASCII *</td>
<td></td>
</tr>
<tr>
<td>042 + ENTER</td>
<td>Load ASCII *</td>
<td></td>
</tr>
<tr>
<td>042 + ENTER</td>
<td>Load ASCII *</td>
<td></td>
</tr>
<tr>
<td>999 + ENTER</td>
<td>Terminate loading EOF chars</td>
<td></td>
</tr>
</tbody>
</table>

3 8 4 – Indicates 8 EOL and 4 EOF characters have been accepted.

All ASCII codes in the range 000 to 127 are accepted, any code in excess of 127 will terminate that part of the loading. Attempted entry of more than 20 E.O.L. or 20 E.O.F. characters will be ignored, but the first 20 characters loaded will be stored. If no E.O.F. characters are required then enter the terminating code "999" twice after the last E.O.L. character code.

All numbers entered and displayed on the Retriever are in the normal decimal numbering system. If you wish to enter a character code whose value is known in binary, octal, or hexadecimal, conversion to a decimal number must be done before entry of decimal equivalent into the Retriever.

For example:

<table>
<thead>
<tr>
<th>ASCII Line Feed</th>
<th>Decimal</th>
<th>Binary</th>
<th>Octal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>010</td>
<td>0001010</td>
<td>012</td>
<td>0A</td>
</tr>
</tbody>
</table>
### ASCII Codes for Control Characters

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>NULL</td>
<td>A null character usually to fill time</td>
</tr>
<tr>
<td>001</td>
<td>SQH</td>
<td>Start of Heading</td>
</tr>
<tr>
<td>002</td>
<td>STX</td>
<td>Start of Text</td>
</tr>
<tr>
<td>003</td>
<td>ETX</td>
<td>End of Text</td>
</tr>
<tr>
<td>004</td>
<td>EOT</td>
<td>End of Transmission</td>
</tr>
<tr>
<td>005</td>
<td>ENQ</td>
<td>Enquiry</td>
</tr>
<tr>
<td>006</td>
<td>ACK</td>
<td>Acknowledge</td>
</tr>
<tr>
<td>007</td>
<td>BELL</td>
<td>Bell</td>
</tr>
<tr>
<td>008</td>
<td>BS</td>
<td>Backspace</td>
</tr>
<tr>
<td>009</td>
<td>HT</td>
<td>Horizontal Tab</td>
</tr>
<tr>
<td>010</td>
<td>LF</td>
<td>Line Feed</td>
</tr>
<tr>
<td>011</td>
<td>VT</td>
<td>Vertical Tab</td>
</tr>
<tr>
<td>012</td>
<td>FF</td>
<td>Form Feed</td>
</tr>
<tr>
<td>013</td>
<td>CR</td>
<td>Carriage Return</td>
</tr>
<tr>
<td>016</td>
<td>DLE</td>
<td>Data Link Escape</td>
</tr>
<tr>
<td>017</td>
<td>DC1</td>
<td>Device Code 1</td>
</tr>
<tr>
<td>018</td>
<td>DC2</td>
<td>Device Code 2</td>
</tr>
<tr>
<td>019</td>
<td>DC3</td>
<td>Device Code 3</td>
</tr>
<tr>
<td>020</td>
<td>DC4</td>
<td>Device Code 4</td>
</tr>
<tr>
<td>027</td>
<td>ESC</td>
<td>Escape</td>
</tr>
<tr>
<td>127</td>
<td>DEL</td>
<td>Delete</td>
</tr>
</tbody>
</table>

Only commonly used control characters have been listed, but the user can input any valid ASCII codes required.

### 3.3.4 Interface 4 - Autostop Option Code

When enabled, this option halts all output from the Retriever to the computer or terminal after completion of each line of data output in the Output Data mode (including programmed end of line characters). The computer or terminal must then send a DC1 code to recommence transmission.

Select Interface code number 4. The right hand digit shows the current option as follows:

<table>
<thead>
<tr>
<th>Autostop Option Code</th>
<th>Resulting Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal operation, no stop at end of line</td>
</tr>
<tr>
<td>1</td>
<td>Stops transmission at end of each line of output in Output Data mode (and Interface 1 test mode)</td>
</tr>
</tbody>
</table>

To alter the Autostop Option Code, depress "*" and "#" simultaneously, and when the display shows 1 dash, enter the appropriate one digit code.

### 3.3.5 Interface 5 - Status Line Input Option

Some terminals and/or communications devices present a "device ready" signal as an indication that serial data (such as output from the Retriever) can now be sent, or must now be terminated. On an RS232 device this will usually be on pin 20 of the 25 way connecting plug and normally has high or positive level to indicate "device ready" for receipt of data.
Status Line (Pin 20 of 25 way RS232 "D" connector)

<table>
<thead>
<tr>
<th>Input Code</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ignore Status Input</td>
</tr>
<tr>
<td>1</td>
<td>Initialize transmission of next character only whilst Status input is high</td>
</tr>
<tr>
<td>2</td>
<td>Initialize transmission of next character only whilst Status input is low</td>
</tr>
</tbody>
</table>

In this context Status Line input is defined as:

-15 to 0 volts — Status Low
5 to +15 volts — Status High

(0 to 5 volts undefined).

Select Interface Code Number "5". The right hand display will show the current value of Status Line Control.

To alter, press "*" and "#" simultaneously and when display shows one dash enter one digit number 0 through 2 as indicated above.

This facility could be used as follows with an external switch:

Pin 1 ————0

5 PIN XLR CONNECTOR

Pin 5 ————0

(Switch open equivalent to Low status input, switch closed equivalent to high status input)

Diagram shows switch closed

Note that if Status Line Input is left disconnected it assumes a low level, i.e. no transmission would occur with Status Line Control = 1, but continuous transmission with 0 or 2.
<table>
<thead>
<tr>
<th>FRONT</th>
<th>BACK</th>
<th>FRONT</th>
<th>BACK</th>
<th>FRONT</th>
<th>BACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>form</td>
<td>25</td>
<td>31</td>
<td>32</td>
<td>49</td>
</tr>
<tr>
<td>2</td>
<td>title</td>
<td>26</td>
<td>33</td>
<td>34</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>vi</td>
<td>27</td>
<td>35</td>
<td>36</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>vi</td>
<td>29</td>
<td>37</td>
<td>38</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>vii</td>
<td>29</td>
<td>39</td>
<td>40</td>
<td>53</td>
</tr>
<tr>
<td>6</td>
<td>vi</td>
<td>blank</td>
<td>30</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>31</td>
<td>43</td>
<td>44</td>
<td>55</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>32</td>
<td>45</td>
<td>46</td>
<td>56</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>33</td>
<td>47</td>
<td>48</td>
<td>57</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>34</td>
<td>49</td>
<td>50</td>
<td>59</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>35</td>
<td>51</td>
<td>52</td>
<td>59</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>36</td>
<td>53</td>
<td>54</td>
<td>60</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>37</td>
<td>55</td>
<td>57</td>
<td>62</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>39</td>
<td>57</td>
<td>blank</td>
<td>61</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>39</td>
<td>59</td>
<td>59</td>
<td>63</td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>40</td>
<td>60</td>
<td>61</td>
<td>64</td>
</tr>
<tr>
<td>17</td>
<td>21</td>
<td>blank</td>
<td>41</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>blank</td>
<td>42</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>23</td>
<td>blank</td>
<td>43</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>25</td>
<td>blank</td>
<td>44</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>26</td>
<td>blank</td>
<td>45</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>27</td>
<td>28</td>
<td>46</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>29</td>
<td>blank</td>
<td>47</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>30</td>
<td>blank</td>
<td>48</td>
<td>72</td>
<td></td>
</tr>
</tbody>
</table>