CLEMSON

CWRR / Report No. 113

WITHDRAWN

USE OF DIGITIZED RADAR AND MICROCOMPUTERS TO FORECAST LOCAL FLOODS AND IRRIGATION NEEDS

by S. Nnaji, D. F. Linvill, and K. D. Robbins

Clemson University, Water resources research institute

Clemson, South Carolina 29631

Project No.A-054-SC

Annual Cooperative Grant Agreement No.14-34-0001-2143

Final Technical Completion Report

to

The Office of Water Policy

U.S. Department of the Interior

Washington, D.C. 20240

TECHNICAL COMPLETION REPORT
A - 054 - SC

USE OF DIGITIZED RADAR AND MICROCOMPUTERS TO FORECAST LOCAL FLOODS AND IRRIGATION NEEDS

by

Soronadi Nnaji
Department of Civil Engineering

Dale F. Linvill
Kevin D. Robbins
Department of Agricultural Engineering
Clemson University

Submitted to

The Office of Water Policy
United States Department of the Interior
Washington, D.C. 20240

The work on which this report is based was supported in part by funds provided by the United States Department of the Interior as authorized under the Water Research and Development Act of 1978 (PL95-467), and the Water Resources Commission.

WATER RESOURCES RESEARCH INSTITUTE
CLEMSON UNIVERSITY
CLEMSON, SOUTH CAROLINA 29631

June, 1983

Publications

Nnaji, S., 1982, "Statistical Flow Forecasting Model for Small Catchments," in the Proceedings of the International Symposium on Hydrometeorology, Denver, CO.

Nnaji, S., 1983, "A Microcomputer Based Radar Information Management System," in the Proceedings of the International Technical Conference of the National Weather Service Real Time Data Collection System and its application to mitigating natural flood hazards, Sacramento, CA.

Thesis

Robbins, K. D., 1983, "Microcomputer Acquisition and Utilization of Digitized Radar Rainfall Data Signals" M.S. Thesis, Department of Agricultural Engineering, Clemson University.

TABLE OF CONTENTS

Abstract ii

Introduction 1

Previous Studies 1

System Development 4
- Hardware 4
- Software 8

Application 17
- Soil Moisture Modeling 17
- Runoff Prediction 18
- Irrigation Scheduling 19

Implementation 19
- Data Collection 21
- Data Formatting 22
- Data Transfer 25
- Data Display 26
- Application 28

Recommendations 34
- Areas Requiring Further Development 34
- Selecting the System 35

Summary and Conclusions 37

References 39

Appendix 41

FIGURES

Figure 1	Equipment Configuration	5
Figure 2	Software modules	9
Figure 3	Schematic of Runoff Prediction	20
Figure 4	Sample Output of Program SHOWRAD	24
Figure 5	Graphics Printer Output with All DVIP Levels	29
Figure 6	Graphics Printer Output with DVIP Levels greater than one . . .	30
Figure 7	Sample Alphanumeric Printer Output	31
Figure 8	Sample Printout from the runoff Model - FORCSTR	32

TABLES

Table 1	Co-ordinates of Selected Cities	15
Table 2	Athens Radar Data for Clemson Watershed	16

APPENDIX

Appendix 1 System Software Programs 41

ABSTRACT

The objective of this study is to develop an inexpensive microcomputer based system for making weather sensitive decisions in real time. The focus is towards the small scale weather information user. To this end the report describes a Radar Information Management System which uses a microcomputer to access, reformat, localize, store and retrieve radar rainfall (or other weather) data in a digitized form. The system will also produce a facimile on command.

In addition to the system software, interactive applications software have been developed that will allow use of the system for local flood forecasting and assessment of irrigation needs given soil moisture predictions based on real time radar rainfall.

An implementation procedure and sample outputs are included.

INTRODUCTION

Obtaining representative and timely estimates of local rainfall have always been an operational problem in soil moisture assessment for irrigation scheduling and flash flood forecasting. The reason for this is, in part, that rainfall events are random in magnitude, in time of occurrence and also in areal coverage. Meteorological observation stations are usually located miles apart and interpolating between measurements at these stations lead to large estimation errors at ungauged sites. Rain gage networks, when they exist at a density required to adequately describe spatial variability of rainfall, involve high maintenance costs. The difficulty exists of obtaining data in real time particularly from remote gages.

Recent advances in computer and radar technology now make it possible to collect, analyze, store, retrieve and disseminate large amounts of weather information in real time and for large areas. At present, however, this information is not available directly to the small scale user, who may include county agents, sewage plant operators and individual farmers.

The objective is to develop an inexpensive microcomputer based system for accessing digitized weather radars and converting their output (storm movement and coverage and rainfall intensities) into information usable in a variety of agricultural and hydrologic decision scenerios. The decision maker is the small scale user who must make real-time decisions for implementing irrigation or treated sewage discharge schedules, pest control strategies and issuance of flash flood warnings among others.

PREVIOUS STUDIES

Literature reviewed include studies that combined radar and computers to produce weather information applicable to hydrologic forecasting and irrigation needs assessment. Initial efforts in hydrologic forecasting have been documented by Curry, Clark and Runnels (1970), Grayman and Eagleson (1970) and Bigler et.al. (1970). A later generation of studies include those by Hudlow (1972), Curry (1973), Greene (1973), Hummels (1975), Sisk (1975), Wiggert and Ostlund (1975) and Tetzloff (1976). Green (1974) expanded on Hudlow's (1972) description of the NWS's D/RADEX system located in the Midwest region. An objective of this system is to improve river stage forecasting techniques using radar rainfall estimates. These estimates are provided at three-hour intervals to NWS regional forecast centers in the service area.

Wasserman (1980) described the radar and satellite observations available in the Eastern region of the NWS. He also discussed the weaknesses in the current systems providing the radar information to the offices responsible for warning the public. He described the NWS's Network and Local Warning radar systems as well as the non-government remote systems used mainly by television stations to prepare radar information directly to the public. Among the weaknesses listed by Wasserman are the frequency and resolution of reported observations from network radars, lack of personnel for continuous monitoring of local warning radars and poor or no communications of real time radar data to offices with warning responsibility.

To alleviate these and other limitations in its operations, the NWS has developed an 'Automation of Field Operations and Services' (AFOS) system. Many papers have been written about the AFOS: its development and implementation (Klein, 1976), its revisions and updates (Mielke, 1979a, 1979b, 1982) and its use for verifying forecasts (Heffernan et.al., 1980, Dunn, 1982). Included in the works reported by Mielke are provisions for adding new capabilities in terms of software into the system. Of immediate interest are the Convective

Parameters Program (Mielke, 1979a) and the Crop and Soil Information Report (Mielke, 1979b). The former predicts potential convective activity while the latter calculates temperature and precipitation data on a weekly basis for selected stations in the Western Region of the NWS. This information is calculated from an AFOS input file updated daily.

An extension of the capabilities of AFOS that enables the system to provide local users with high resolution timely radar data has been developed by the Western Region of the NWS (Mathewson, 1982). This extension, AFOS Radar Processor (ARAP), generates information in two formats - AFOS graphic and AFOS alphanumeric. The latter is produced for users who do not have access to AFOS. The products include meteorologic parameters such as intensity, echo tops, accumulation etc. provided in real time in 5 - 60 minute intervals for both small (city-county) and large (drainage basin) areas. The product of this mini computer based system may also include areal coverage vs. time plots of monitored parameters as well as an alarm/alert warning message.

While the products of ARAP are readily available to users through commercial telephone call up, availability is limited to the service areas of a few network radars located in the Western Region. There is no parallel to the ARAP in the Eastern Region of the NWS within which the State of South Carolina is located. Thus the weaknesses discussed by Wasserman (1982) and resolved for the most part by ARAP in selected service areas in the Western Region remains unaddressed in the East.

The pioneering efforts in the area of irrigation scheduling are those of Jensen (1969) and Jensen et. al. (1970) who developed and implemented the U.S. Department of Agriculture's irrigation scheduling program that was run on a mainframe computer. The program uses meteorological data to calculate water use and maintains a soil water budget. Other mainframe based scheduling programs and consultancies which utilize these programs to provide scheduling services exist. One may expect, however, that only large scale farmers will find such services cost-effective.

More recent work (Kincard and Heerman (1974), Harrington and Heerman (1981), Fereres et. al. (1981) and Lambert, Doty and Quisenberry (1981)) have adopted the approach of using personal computers which can be afforded by the family farmer. Particularly germane to this study is the work of Lambert et. al. These researchers developed a water-budget based irrigation program for the Radio Shack TRS-80 personal computer. The program is based on measured daily radiation maximum and minimum daily temperatures, rain or irrigation and five (5) day forecast of radiation and temperatures.

SYSTEM DEVELOPMENT

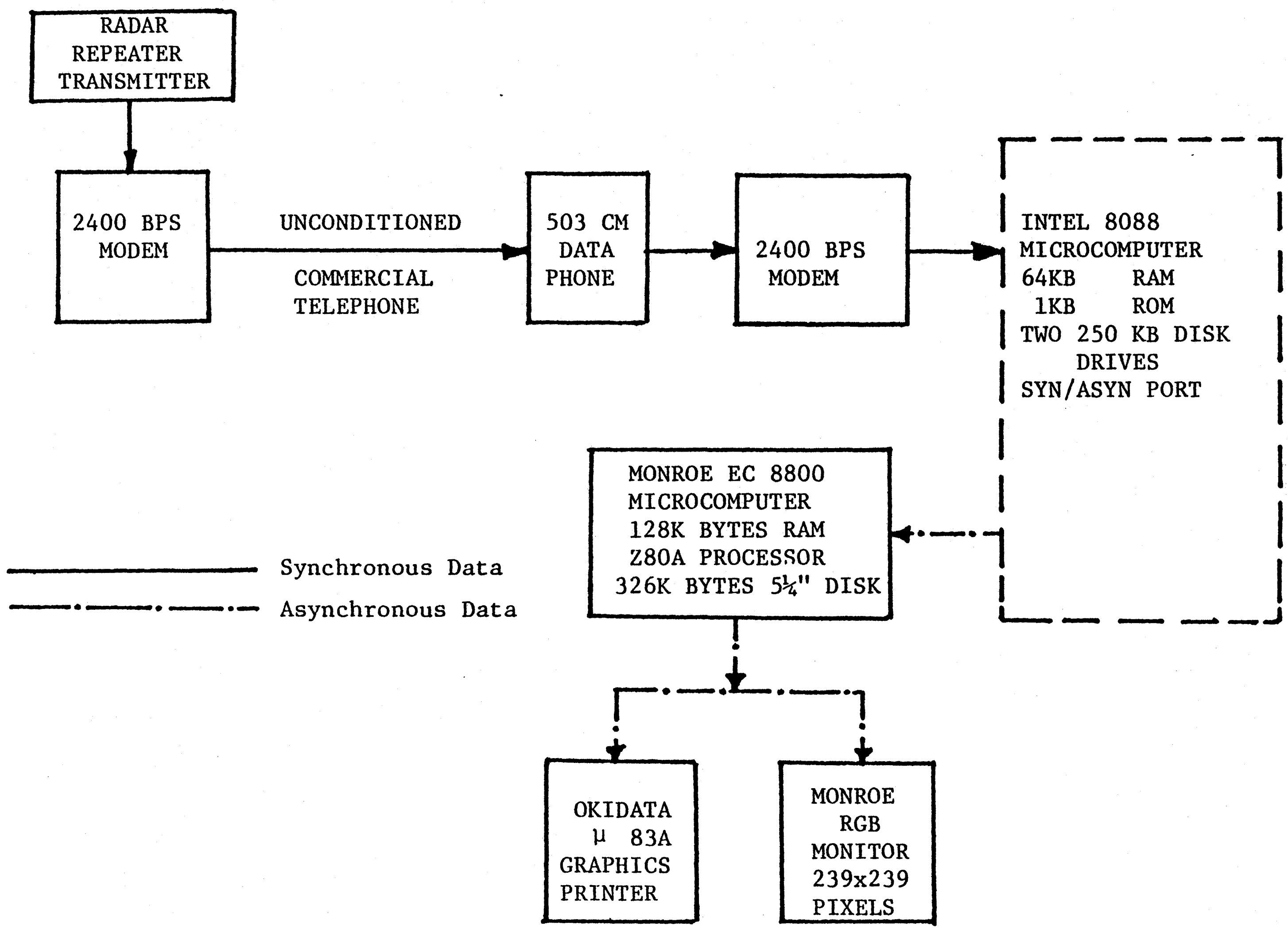
The basic system includes the digitized radar, a modem for receiving radar transmitted rainfall (echo) intensities and a microcomputer for processing the transmitted data. The microcomputer houses the software modules for data processing and modules for application purposes. Figure 1 shows the hardware configuration and the data transmission path.

Hardware

Radar

The NWS radar sets in and around South Carolina, are equipped with digitizing units accessible via telephone call-up. Six of these radars cover the state. The Athens, Georgia and the Charleston, South Carolina radars are part of the NWS network system while those at Columbia, SC, Charlotte, NC, and Augusta and Atlanta, GA are part of the NWS Local Warning radars. In addition

FIGURE 1. EQUIPMENT CONFIGURATION AND DATA TRANSMISSION PATH



to NWS radar, some television stations own and operate weather radars most of which are equipped with digitizers though none has been calibrated. The NWS network radar at Athens is used for this project. The digitized radar image contains 6 levels indicating the strength of the return signal. For the Athens radar, the rainfall intensity bounds corresponding to the digitized video integrator and processor (DVIP) levels are:

DVIP Level	Rain Intensity (in/hr)
0	0
1	.01 - .1
2	.1 - .5
3	.5 - 1
4	1 - 2
5	2 - 5
6	> 5

Enterprise Radar Repeater Transmitter

In order to directly access the digitized radar data it is necessary to physically attach a transmitter equipment to the radar. Only two commercial companies, under contract with the NWS, have such access. The companies operate processing equipment which transmit the digitized data, at cost, over commercial telephone lines to subscribing users such as T.V. stations. The Enterprise Electronics Inc. gave permission for the access to its transmitted data at no cost to the project.

Modem

The word MODEM is a contraction for modulator-demodulator. The modem is an equipment which allows communication of digital data signals over telephone and other limited bandwidth communication channels. The characteristics of the modem selected are rigidly defined by the incoming radar signal characteristics:

rate - 1200 baud

modulation - differentially phase shift keyed (DPSK), differential guadrature phase shift (DQPS) or equivalent

Carrier frequency - 1800 hz

Line compatibility - Bell 201B

Data format - serial synchronous

The key to modem selection is the type of modulation. Suitable modems for use in gathering data from the Enterprise Electronics WR100/77 radar data remoting system include the UDS-201B modem and the RIXON DS-2401 modem. While this does not constitute an endorsement of the above mentioned products, they are both known to be suitable for gathering the radar data. In particular the UDS-201B modem was used both for the system shakedown and for routine data collection.

Microcomputer

The first phase of the project was acquisition of a suitable microcomputer for processing of radar data. The criteria to be met were high resolution color graphics for the display of the six levels of radar rainfall intensities, large storage capacity (nearly 300KB) for storage of a complete radar picture (sweep), convenient graphic commands, relatively low cost which would be affordable to future users and the capability for expansion to meet future needs.

The Monroe EC8800 Educational Microcomputer was chosen (February 1982) to best meet the needs of the project. This computer has only one built-in disk drive and accepts only asynchronous data while the transmitted data is in a synchronous format. These limitations necessitated our insertion of an intermediate unit between the modem and the Monroe. This unit is an INTEL 8088 microprocessor with 64KB of RAM, IKB of ROM, two 250 KB disk drives and a synchronous/asynchronous series part. We hasten to add that recent advances in microcomputer technology have produced inexpensive stand alone systems having all the necessary attributes for data processing and hydrologic modeling. Lu (1982) has compiled a partial list of such microcomputers by make and their built-in attributes.

Software

System software modules were written for the INTEL processor unit and the MONROE microcomputer depending on the data processing function performed in the unit. Figure 2 shows the modules which are described below. The programs are listed in Appendix 1. The modules RADARXFR, RADARTRANS, SATGET and SATIN are not required if all data handling and applications modeling are done on a single microcomputer having minimum recommended attributes.

Radarp

The program Radarp is written in the PL/I programming language. Its purpose is to gather synchronous radar data through a serial USART (universal synchronous/asynchronous receiver/transmitter) port. The source of the port data is a UDS-201B modem connected to an exclusion key data telephone set. The set is connected to a standard two wire telephone line. The signal coming down

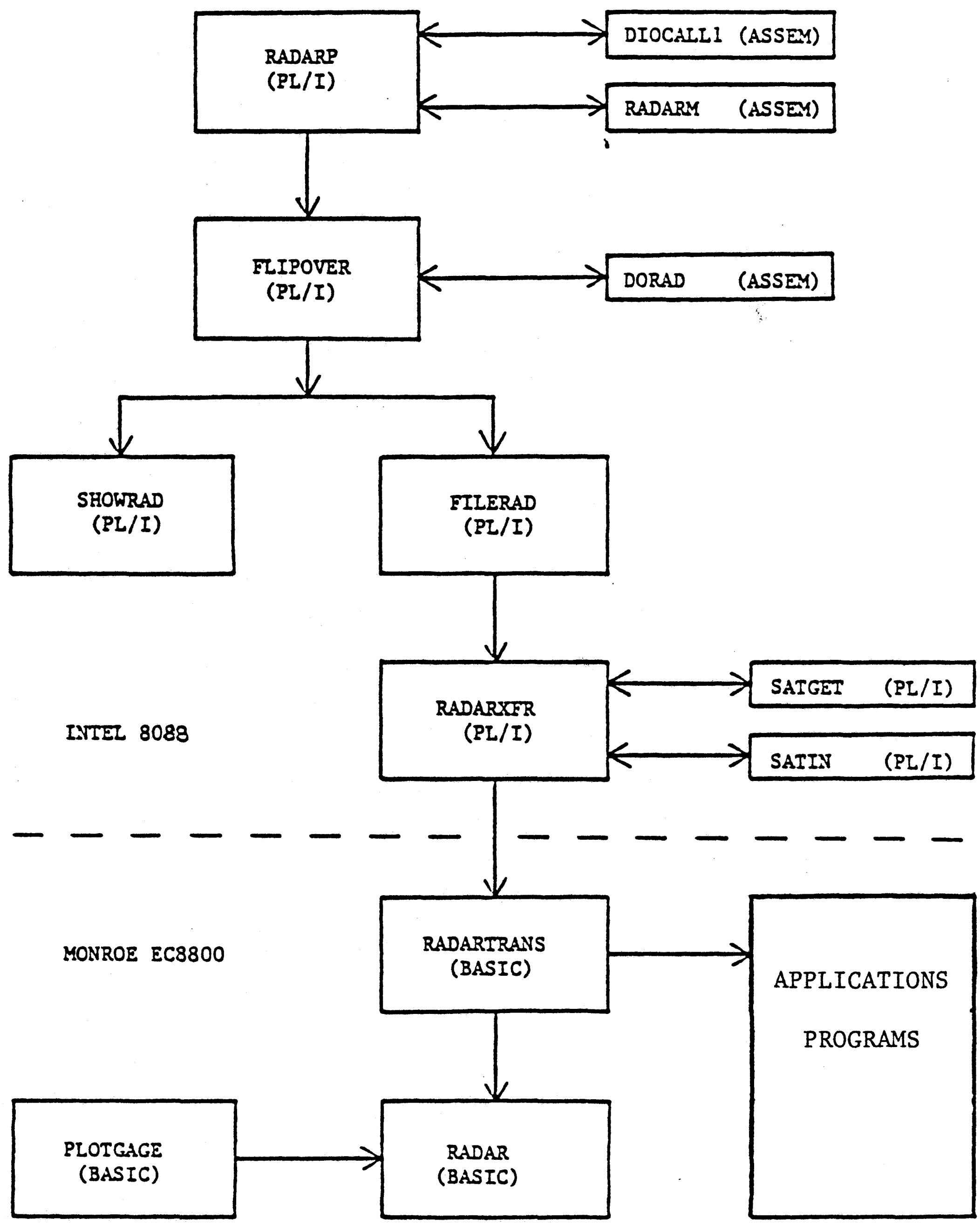


Figure 2. Software Modules and Information Transmission Path

the line is received from the Enterprise Electronics transmitter, connected to the Athens NWS WSR-57 radar set.

Radarp senses the output of the modem and begins by first looking for a valid synchronization word. Following the receipt of the synchronization the program reads rainfall intensity, time of day, antenna azimuth and elevation. The rainfall intensities represent data for 20-270km along each radial. Data is available up to 450km but is not collected past 270km. After collecting data along a radial, the program waits for the next synchronization word and repeats the process. In all 360 radials of data are collected and stored in a sequential disc file called 'RADAR.DAT'.

The data collection process takes about 2 min 10 sec to complete while the file storage takes about 1 min 10 sec.

Diocall 1

This assembler program is called by Radarp. It directs the various program calls within Radarp.

Radarm

This is an assembler program which is called from Radarp. Its purpose is to operate the USART port. It initializes the port and also operates the synchronization function of the port.

Flipover

This program is written in PL/I and operates on the file "RADAR.DAT" which was produced by Radarp. It reverses the order of all bits within each input byte and also reverses the order of all bytes within each radial. The bit reversal is required because of the way the USART port inputs the radar data. The USART assumes that the least significant bit (LSB) is entering first, when in actuality, the most significant bit (MSB) of data enters the USART first. The "RADAR.FLP" is created after the reversals are complete.

Dorad

This assembler program is called from Flipover to accomplish the bit and Byte reversals. It has the additional function of removing unwanted bits from the data stream. These unwanted bits are 16 bits of unwanted synchronization as well as the first and last bits of each data word. It also discards the first data word which is currently undefined and has no function.

Filerad

This PL/1 program works on the file "RADAR.FLP". Its purpose is to decode the time of day, antenna elevation and azimuth and the rainfall data. After each radial is decoded it is written to a file in ASCII code. The resulting file is called "RADAR.XFR" and is approximately 52 bytes long. This is the final file in the data collection process and is ready to be transferred to the Monroe EC8800 for archival.

Radarxfr

This PL/1 file controls the transfer of "RADAR.XFR" from the Intel 8088 to the Monroe EC8800. The transfer procedure occurs at 1200 BAUD and is an image ASCII transfer.

Satget

This PL/1 program is called by Radarxfr to declare various character functions used in the data transfer process.

Radars

This CP/M submit file is used to link various programs used by Radarp.

Bardset

This CP/M submit file is used to initialize various input and output ports and to declare ports as files so that the CP/M operating system can access them.

Showrad

This PL/1 program is functionally identical to Filerad with the exception that no additional files are created. All decoded data is output to a CRT. It is used to preview the data during the collection process to give the operator an idea of the character of the rainfall event.

Radartrans

The transfer of ASCII files from the INTEL 8088 to the Monroe EC8800 required two programs to be written. The first program, Radarxfr was an output program written in PL/I. The second program, Radartrans, was written for the EC8800 in Basic. It controls the input of the data to a disk file.

The operation of the program involves naming the input file, collecting data from the INTEL 8088 one radial at a time and writing the radial to the disk file. In all, 360 radials of data are transferred comprising 52190 Bytes of data. Transfer of a file takes about 5 minutes to complete.

Radar

Once the radar data has been transferred to the EC8800, processing and display programs can be run. Radar is the display program and is written in

Basic. It works by reading a radial of data from the disk file which contains information on time of observation, antenna altitude, antenna azimuth and rainfall DVIP levels.

Screen plotting begins at the first radial of the disk file. Since the data is in polar format and the Basic plotting routines work in rectangular coordinates, the first step is to accomplish a polar to rectangular conversion on the data. This is done for each "point" of radar data. Data with a zero value is not converted to rectangular coordinates, in order to save plotting time.

The program has the option of 5 display formats. The operator can choose a full display in which the radar is located at the center of the screen and the radar data is plotted around the radar site. In this mode only even numbered radials are plotted and every other data point along a radial (every 4km) is plotted. Thus, only 1/4 of the data is actually untilized. Plotting time varies from 2 to 4 minutes depending upon the amount of rain occurring in the event.

The full screen display also displays a map overlay which shows state boundaries within the radar coverage area as well as the location of cities or raingages. This is useful in identifying areas of interest relating to the rainfall event.

The other four options enable the operator to plot one of the 4 quadrants (0°-90°, 90°-180°, 180°-270° or 270°-359°). In this mode, the radar is displayed in one of the screen corners. The quadrant is then displayed using all radials in the quadrants range as well as plotting all data along the radial (2km increment). The scale of the display is changed by a factor of two. While the plotting density visually appears to be the same, the amount of data in the quadrant is increased by a factor of 4. This gives more detail in the display. Plotting times for a quadrant are comparable to a full screen display. The quadrant display section plots gages, but does not create a map overlay at this time.

Plotgage

In order to locate a point or area within the display, a program was required which would calculate bearing and distance from the radar, using latitude and longitude coordinates. This is accomplished by Plotgage. The parameter are calculated using Great Circle Route formulas. Table 1 shows these information for locations within the project area.

The operator can route different displays created by the program to a printer for later use. The program outputs the bearing in degrees and radians and distance in statute miles, radians, nautical miles and kilometers. It also converts the parameter to screen coordinates appropriate to the plotting scale used in Radar. The coordinates are then used as input data points for plotting gage locations as well as plotting the map overlay.

Convert

The purpose of this Basic routine is to convert the collected instantaneous DVIP levels into rainfall intensity bounds using the conversion code for the Athens radar. The intensities in inches/hour are then converted to intensities in inches per radar scan interval. Corresponding cummulative rainfall bounds at a point can be calculated for multiples of scanning intervals or the entire scanning period. Table 2 shows the output from this routine for the rainfall event of June 7, 1983.

Table 1. Co-ordinates of Selected Cities Within the Athens Radar Range

```
RADAR = ATHENS GA
LATITUDE = 33.57.00      LONGITUDE = 83.19.00

ALLENDALE SC
LATITUDE = 33.00.00      LONGITUDE = 81.18.00
BEARING(DEG) FROM ATHENS GA = 118.90
DISTANCE(KM) FROM ATHENS GA = 214.66
MONITOR "X" COORDINATE 213.0
MONITOR "Y" COORDINATE  67.1

ANDERSON SC
LATITUDE = 34.30.00      LONGITUDE = 82.43.00
BEARING(DEG) FROM ATHENS GA =  41.88
DISTANCE(KM) FROM ATHENS GA =  82.30
MONITOR "X" COORDINATE 146.5
MONITOR "Y" COORDINATE 149.6

CLEMSON SC
LATITUDE = 34.41.00      LONGITUDE = 82.49.00
BEARING(DEG) FROM ATHENS GA =  29.24
DISTANCE(KM) FROM ATHENS GA =  93.52
MONITOR "X" COORDINATE 141.8
MONITOR "Y" COORDINATE 159.8

COLUMBIA SC
LATITUDE = 33.57.00      LONGITUDE = 81.07.00
BEARING(DEG) FROM ATHENS GA =  89.39
DISTANCE(KM) FROM ATHENS GA = 202.79
MONITOR "X" COORDINATE 220.4
MONITOR "Y" COORDINATE 120.1

GRNVLL-SPRTNBRG SC
LATITUDE = 34.54.00      LONGITUDE = 82.13.00
BEARING(DEG) FROM ATHENS GA =  43.37
DISTANCE(KM) FROM ATHENS GA = 145.98
MONITOR "X" COORDINATE 169.1
MONITOR "Y" COORDINATE 172.1

ROCK HILL SC
LATITUDE = 34.59.00      LONGITUDE = 80.58.00
BEARING(DEG) FROM ATHENS GA =  61.27
DISTANCE(KM) FROM ATHENS GA = 243.99
MONITOR "X" COORDINATE 226.0
MONITOR "Y" COORDINATE 177.6
```

Table 2. Athens Radar Rainfall Data for Clemson Watershed
Event of June 7, 1983

-071217
15 MINUTE LOW RANGE RAINFALL = 0.003 in.
15 MINUTE MED RANGE RAINFALL = 0.013 in.
15 MINUTE HIGH RANGE RAINFALL = 0.025 in.

071248
29 MINUTE LOW RANGE RAINFALL = 0.005 in.
29 MINUTE MED RANGE RAINFALL = 0.024 in.
29 MINUTE HIGH RANGE RAINFALL = 0.048 in.

071300
12 MINUTE LOW RANGE RAINFALL = 0.002 in.
12 MINUTE MED RANGE RAINFALL = 0.008 in.
12 MINUTE HIGH RANGE RAINFALL = 0.016 in.

071313
13 MINUTE LOW RANGE RAINFALL = 0.002 in.
13 MINUTE MED RANGE RAINFALL = 0.008 in.
13 MINUTE HIGH RANGE RAINFALL = 0.016 in.

071325
12 MINUTE LOW RANGE RAINFALL = 0.002 in.
12 MINUTE MED RANGE RAINFALL = 0.010 in.
12 MINUTE HIGH RANGE RAINFALL = 0.020 in.

071343
19 MINUTE LOW RANGE RAINFALL = 0.003 in.
19 MINUTE MED RANGE RAINFALL = 0.016 in.
19 MINUTE HIGH RANGE RAINFALL = 0.032 in.

071427
43 MINUTE LOW RANGE RAINFALL = 0.007 in.
43 MINUTE MED RANGE RAINFALL = 0.033 in.
43 MINUTE HIGH RANGE RAINFALL = 0.066 in.

071459
32 MINUTE LOW RANGE RAINFALL = 0.005 in.
32 MINUTE MED RANGE RAINFALL = 0.027 in.
32 MINUTE HIGH RANGE RAINFALL = 0.053 in.

071537
39 MINUTE LOW RANGE RAINFALL = 0.000 in.
39 MINUTE MED RANGE RAINFALL = 0.000 in.
39 MINUTE HIGH RANGE RAINFALL = 0.000 in.

071653
81 MINUTE LOW RANGE RAINFALL = 0.014 in.
81 MINUTE MED RANGE RAINFALL = 0.068 in.
81 MINUTE HIGH RANGE RAINFALL = 0.135 in.

RAINFALL TOTALS FOR PERIOD

LOW TOTAL = 0.041 in.
MED TOTAL = 0.206 in.
HIGH TOTAL = 0.411 in.

APPLICATION

Central to the project is the estimation of soil moisture in a watershed or agricultural field since the amount of antecedent soil moisture determines, to various extent, how much runoff occurs during and after a rainfall event. It also determines the amount of irrigation to be applied over a crop land.

Runoff occurs when the rainfall intensity exceeds infiltration capacity of the soil or when the soil matrix within the zone of major hydrologic activity is saturated with infiltrated rain. The latter case may result also when intense rain falls on vegetated areas such that the canopy absorbs some of the impact force and consequently reduces the effective rain intensity on the soil surface.

While rainfall replenishes soil moisture, evapotranspiration, deep seepage and interflow abstract water from the soil. A complete modeling of soil moisture dynamics should consider all the processes present. For our purpose, which is to demonstrate the use of radar rainfall in real time runoff prediction, and irrigation scheduling, it suffices to adopt a simple model of soil moisture dynamics. To this end we do not consider deep seepage and interflow. Evapotranspiration is assumed to be constant throughout the event.

Soil Moisture Modeling

A water balance approach for soil moisture accounting is used as a basis for developing the simple runoff model. Rainfall is assumed to be intercepted in proportion (α) to the density of the canopy over the catchment. Once the interception capacity (C^*) is satisfied all rain find their way to the soil surface. At the surface, the rainfall either replenishes the soil moisture or is available for runoff as effective precipitation depending on the current soil moisture content. The soil moisture content at the end of the kth time interval since rainfall monitoring started is given by:

$$SM_k = SM_{k-1} + RAIN_k - ET_k - RUNOFF_{k-1} \quad (1)$$

where ET is evapotranspiration and T_k, define below,

$$\tau_k = \sum_{j=1}^{k} \Delta_j \quad (2)$$

is the time since monitoring started while j is the time interval in minutes) between the (j-1) and the jth radar scan.

The soil moisture at the onset of an event depends on the antecedent moisture which in turn is a function of the (initial) soil moisture immediately after a previous event. To obtain the antecedent soil moisture, an exponential soil moisture decay model, equation (3), is adopted

$$S_t = S_{t'} \, Exp(-gT_t) \quad (3)$$

where T_t is the temperature and S_t is the soil moisture for time, t > t', since the last irrigation or rain and before rainfall monitoring started. The exponent, g, is a function of soil cover and soil type. The temperature sequence at time instants less than t are assumed to be observed values. The value T_t may also be observed or may be a forecast. The forecast case is not considered in this report.

Runoff Prediction

The effective rainfall is the rainfall in excess of that required to satisfy the soil moisture capacity and evaporation. It may also be called the potential runoff. The effective rainfall, R(s') is convoluted with the impulse response function, H estimated for the catchment in order to obtain a runoff hydrograph. The expression for the convolution is

$$Q(s) = \sum_{s^1 = o}^{s^1 = s} R(s')H(s-s') \tag{4}$$

where $R(s') = 0$ when $s' > T_k$. The variable Q(s) is discharge in inches/hr at the s th instant since rainfall monitoring started. Both the impluse response function and Q(s) are defined continuously though the input is defined in quantized form being constant for each of the k intervals between radar scans. This hydrograph is a prediction given the rainfall sequence so far. Figure 3 is a schematic of the runoff prediction process. The process of runoff generation and convolution is repeated as new rainfall observations are received.

Irrigation Scheduling

Using the model adopted for soil moisture depletion, equation 3 and the estimated parameters for the site, one can obtain an estimate of soil moisture at the end of an event. This estimate now becomes, S_o, the starting point of soil moisture depletion monitoring. Given temperature forecasts, soil moisture deficit and hence the next irrigation time and quantity, if it does not rain meanwhile, may be predicted.

IMPLEMENTATION

In order to collect and process transmitted radar data, both the system and applications software must be implemented in a prescribed manner. This section gives the steps with the user commands and the corresponding system response at each step. The steps are based on current system configuration which consists of the MONROE EC8800 augmented by the INTEL 8088.

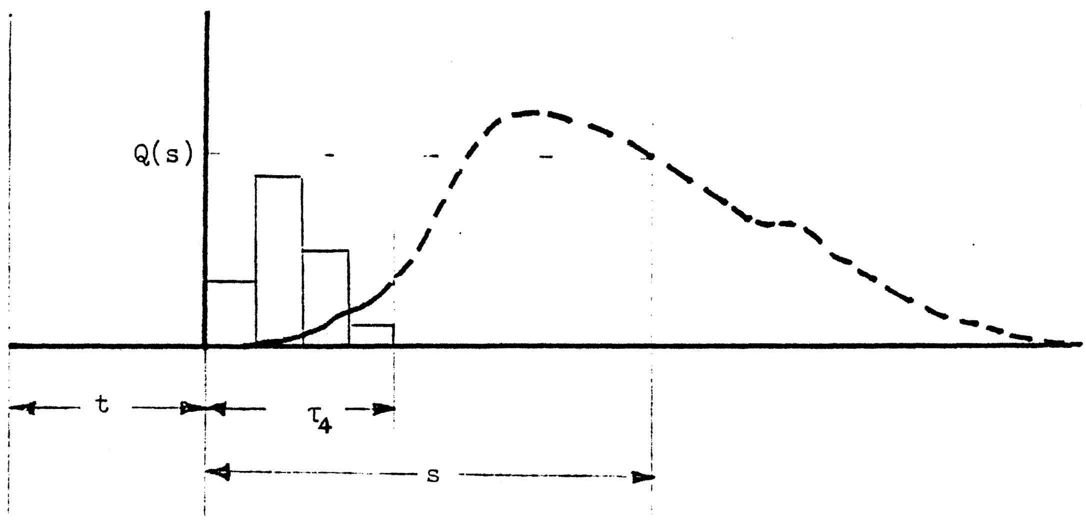


Figure 3. Schematic of Runoff Prediction Q(s) is predicted flow at time S based on rainfall through τ_4 since rain monitoring started, t is duration of soil moisture depletion monitoring.

Data Collection

Data collection begins by first loading the 8" disk containing the system software programs into the INTEL 8088. The first operator command instructs the INTEL processing unit to initialize ports and load program overlays. This function is performed by program IO IO which is initiated by typing

IO IO

to which the computer will respond by displaying

INITIALIZING PORTS
OVERLAYS IN PLACE

The processing unit dynamic debugging task (DDT) must then be loaded. This is done by typing

DDT

to which the computer will respond with

DDT VERS 1.4.

The computer will then display several memory addresses which must be loaded by operator responses. These addresses disable bit stripping commands present in the IO IO program. Computer prompts and operator responses are given below.

```
sfe9f
E6 00          note:  responses are zeroes
7F 00
C9 .
-g0
```

After this has been done, the logical input device must be assigned as reader input by typing the following command string.

STAT RDR: = UR2:

The computer is now ready for the data collection program to be loaded. This is done by typing

RADARP

which will load the program into RAM. The program will monitor the input port for proper data synchronization words to begin data input.

The operator must ensure that the modem power switch is turned on and then dial the number for the selected radar transmitter. As soon as the transmitter answers the telephone, data will be transmitted along the telephone line. This data can be aurally detected and is characterized by a wavering "white noise" sound. When this sound is detected, the exclusion key on the data telephone should be lifted which will transfer control of the line to the modem. Data will be transferred from the modem to the microcomputer and input via RADARP. Upon receipt of 360 radials of data, the program will store the data to a disk file and terminate.

Data Formatting

The collected file must be converted to a format that can be decoded. This function is performed by the program FLIPOVER which is initiated by typing the command

FLIPOVER .

The file created by FLIPOVER, "B:RADAR.FLP", is operated on by the next program in the series, FILERAD. If immediate processing on the file is desired, this program can be initiated directly. However, it is desirable to rename "B:RADAR.FLP" to allow collection of another radar data set. This protects the file from being overwritten by subsequent operation of FLIPOVER. An advisable file format which fits into the file specifications of CP/M is "B:NNddhhmm", where "B:" specifies disk drive B, "NN" is a two letter identifier for the radar transmitter, "dd" is the day of the month, "hh" is the hour (Greenwich Mean Time) and "mm" is minutes. It would be desirable to include a designation for the month, but a limit on filename length precludes this feature. To rename the file in this format, the REName utility is used by typing

REN B:NNddhhmm = B:RADAR.FLP

which will preserve the file in the renamed format. To enable this file to be processed by FILERAD, the file must be renamed to "B:RADAR.FLP" using the same CP/M command.

To operate on "B:RADAR.FLP", the decoding program, FILERAD, is loaded by typing

FILERAD .

This creates a decoded file of radar information. The resulting file name is "B:RADAR.XFR". This is an expanded length ASCII file and is ready for transfer to the EC800.

The program, SHOWRAD, is executed by typing the command

SHOWRAD .

It is very similar to FILERAD in that it operates on the same input file and has the same decoding routines. The major difference is that SHOWRAD does not produce an output file. Output is sent to the CRT to enable previewing of the data. A sample of the data viewing format is given in figure 4. It was obtained by typing a "control p" at the beginning operation.

```
HOUR: 13      MINUTE:  0

ELEVATION:    0.2       AZIMUTH:  29

  20 KM       1 1 1 1 1 1 1 1
  36 KM       1 3 1 2 1 1 1 1
  52 KM       1 1 2 1 0 0 0 0
  68 KM       0 0 0 0 0 1 1 1
  84 KM       1 1 1 1 1 1 1 1
 100 KM       1 1 1 1 1 1 1 1
 116 KM       1 1 1 1 1 1 1 1
 132 KM       0 0 0 0 0 0 1 1
 148 KM       1 1 2 3 3 6 2 1
 164 KM       0 0 0 0 0 0 0 0
 180 KM       0 0 0 0 1 0 0 0
 196 KM       1 0 0 0 0 0 0 0
 212 KM       0 0 0 0 0 0 0 0
 228 KM       0 0 0 0 0 0 0 0
 244 KM       0 0 0 0 0 0 0 0
 260 KM       0 0 0 0 0 0 0 0

TYPE "NEW LINE" TO SEE NEXT RECORD
"C" FOR CONTINUOUS DISPLAY, "Q" TO QUIT

HOUR: 13      MINUTE:  0

ELEVATION:    0.2       AZIMUTH:  30

  20 KM       1 1 1 1 1 1 1 1
  36 KM       1 3 1 1 1 1 1 1
  52 KM       1 1 1 1 0 0 1 1
  68 KM       1 1 0 0 0 0 0 0
  84 KM       1 1 1 1 1 1 1 1
 100 KM       1 1 1 1 1 1 1 1
 116 KM       1 1 1 1 1 1 1 1
 132 KM       1 0 1 1 0 0 1 0
 148 KM       0 0 1 2 2 6 2 2
 164 KM       0 0 0 0 0 0 0 0
 180 KM       0 0 0 0 1 0 0 0
 196 KM       1 0 0 0 0 0 0 0
 212 KM       0 0 0 0 0 0 0 0
 228 KM       0 0 0 0 0 0 0 0
 244 KM       0 0 0 0 0 0 0 0
 260 KM       0 0 0 0 0 0 0 0

TYPE "NEW LINE" TO SEE NEXT RECORD
"C" FOR CONTINUOUS DISPLAY, "Q" TO QUIT
```

Figure 4. - Sample Output of Development System Program SHOWRAD. DVIP levels are given for 2Km increments along the radial shown in the heading. The bottom of the output shows program control choices available to the operator.

Data Transfer

Data transfer requires the operation of two programs running concurrently. One program, RADARXFR, will operate within the Development System and the other, RADARTRANS, will operate in the EC8800. The operator must first load RADARTRANS into the EC8800 from the 5 1/4" source program disk. This is done by inserting the disk which will load the Monroe Operating System into RAM. The Basic language and RADARTRANS must then be loaded into RAM by typing

BASIC,,30000 RADARTRANS .

After searching the diskette and loading the programs, the computer will display a prompt menu as follows

DISC VOLUME "RAIN:" MUST BE IN FPYO:

TO CHANGE DISKS, TYPE THE FOLLOWING

-VOL RAIN:
CLOSE FPYO:

INSERT FPYO:

OPEN FPYO:
CONTINUE .

This sequence of commands is necessary in order to change disks. The program disk (SYS:) is replaced with a data disk (RAIN). After disks are changed, the screen is cleared and the following prompt appears

INPUT STATION
A. ATHENS
C. CHARLESTON .

INPUT STATION (A OR C)

The operator responds by selecting a radar station with the appropriate letter. The screen is again cleared and the prompt

ENTER FILE DATE & TIME ddhhmm

appears. The operator responds by entering the date identifier. Before the "RETURN" key is pressed, RADARXFR must be started on the Development System. This ,is done by typing

RADARXFR

on the INTEL 8088 keyboard. The "RETURN" key can then be depressed on the EC8800 and data transfer will begin. After transfer of 360 radials of data, the EC8800 will display

TRANSFER COMPLETE

in large red letters. Transfer of all files can be accomplished in this manner, remembering that the files on the INTEL 8088 must be created by the operation of FILERAD before initiating the transfer programs. Data files are fairly large (about 52K), and disk storage capacity of the EC8800 will only allow five files per disk.

Data Display

Operation of the radar display program, RADARQUAD, requires insertion of the program diskette (SYS:). Once the operating system is loaded, the command string

BASIC,,30000 RADARQUAD

will load the program. The program begins by prompting the operator to change the program disk to a data disk, select a radar name and choose an event time in the same manner as was described for RADARTRANS. The screen is then cleared and an additional prompt appears which allows selection of a full, 360 degree plotting format or a choice of one of the four 90° quadrants. The prompt message appears as

```
PLOT FULL SCREEN (F)
      OR
QUADRANTS (1,2,3,4)

      2  1
      3  4
```

to which the operator must respond with the appropriate character. The screen is again cleared and a high resolution display of state boundaries and major city locations are displayed. Radar DVIP level are plotted on this background. After plotting, a prompting message

```
PRINT RESULTS (Y OR N)
```

appears at the bottom of the screen. A negative response will end execution of the program. A positive response will cause the graphics to be hidden and another prompting message

```
SELECT PRINT SIZE

1.  6 LPI, 10 CPI (LARGE SCALE)
2.  6 LPI, 16.5 CPI
3.  8 LPI, 10 CPI
4.  8 LPI, 16.5 CPI (SMALL SCALE)

SELECTION ?

1.  PRINT ALL COLORS
2.  PRING COLORS  LEVEL 1

SELECTION ?
```

appears on the screen. The "small scale" selection will create a printout that fits on 8 1/2" x 11" paper, while the "large scale" printout will be nearly 15" on a side. The color selection prompt is included so that outputs can be obtained for all DVIP levels or for more intense levels, only. Samples of this output are given in figures 5 and 6.

A text printout of the radar image can be obtained with RADARPRINT. The program is loaded in the same manner as the others by typing

BASIC,,30000 RADARPRINT .

A prompting message is generated, as before, which tells the operator to switch to a data disk and specify an input file. In addition, the operator is reminded to position the paper at the top of the form. An output is then printed which gives DVIP levels in two kilometer increments for each of the 360 radials. The output is formatted for teletype paper, 8 1/2"V x 11"H. This format was selected so that the data would fit on the paper, yet be easy to file. Nine pages of output are produced for each radar picture. Samples of this output are given in figure 7.

Application

The runoff forecasting model and the irrigation scheduling model were coded into interactive computer programs FORCSTR and FORCSTI respectively. For example, FORCSTR can be loading by typing

BASIC,,30000 FORCSTR .

As before, a prompting message is displayed to enable data disk insertion, radar file selection and radar transmitter specification. The prompt has an additional feature which allows specification of the number of radar files present in the rainfall event. Figure 8 contains a sample printout for FORCSTR. The variables S*, ID and TEMP are, respectively, the soil moisture at

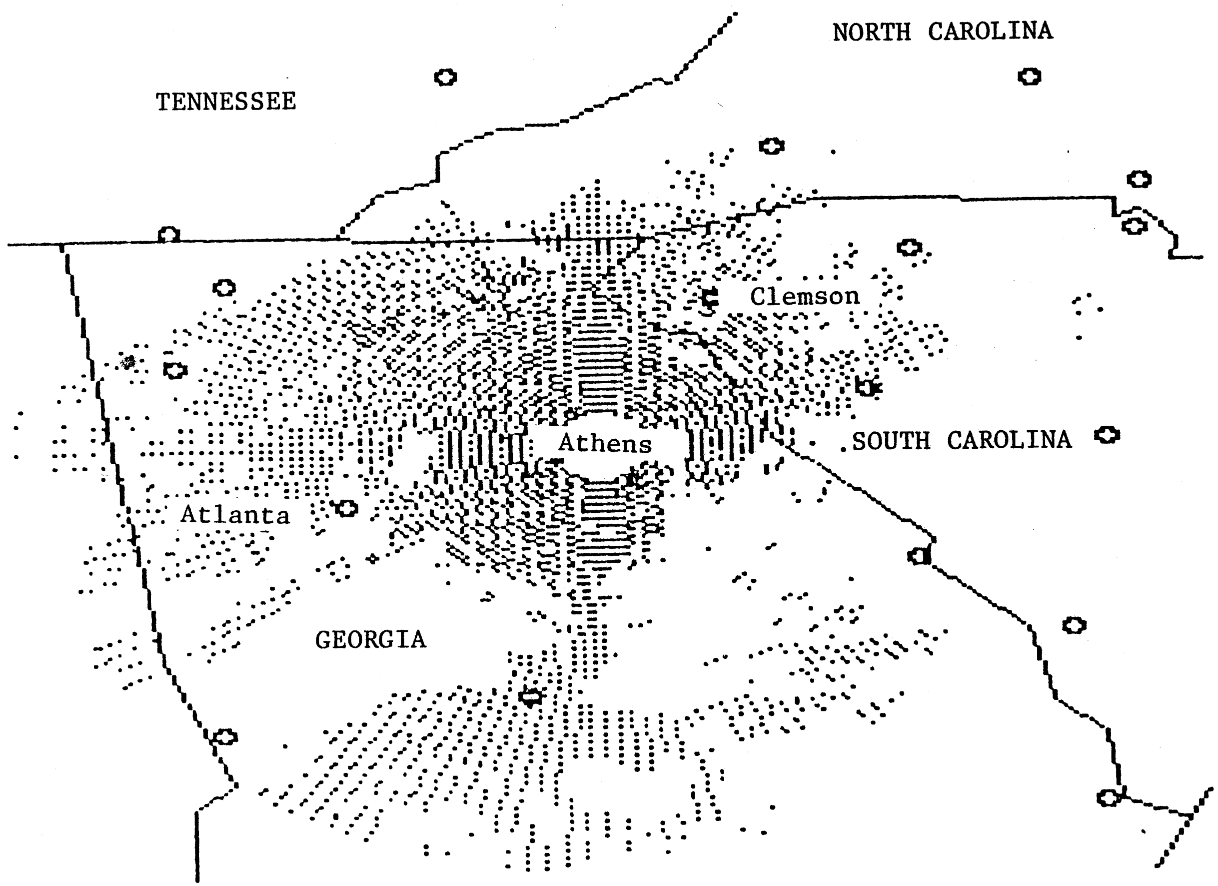


Figure 5 - Graphics Printer Output Of Radar Rainfall Event (Athens, Ga., June 7, 1983, 1300 GMT) as Produced by RADARQUAD. All DVIP levels are displayed.

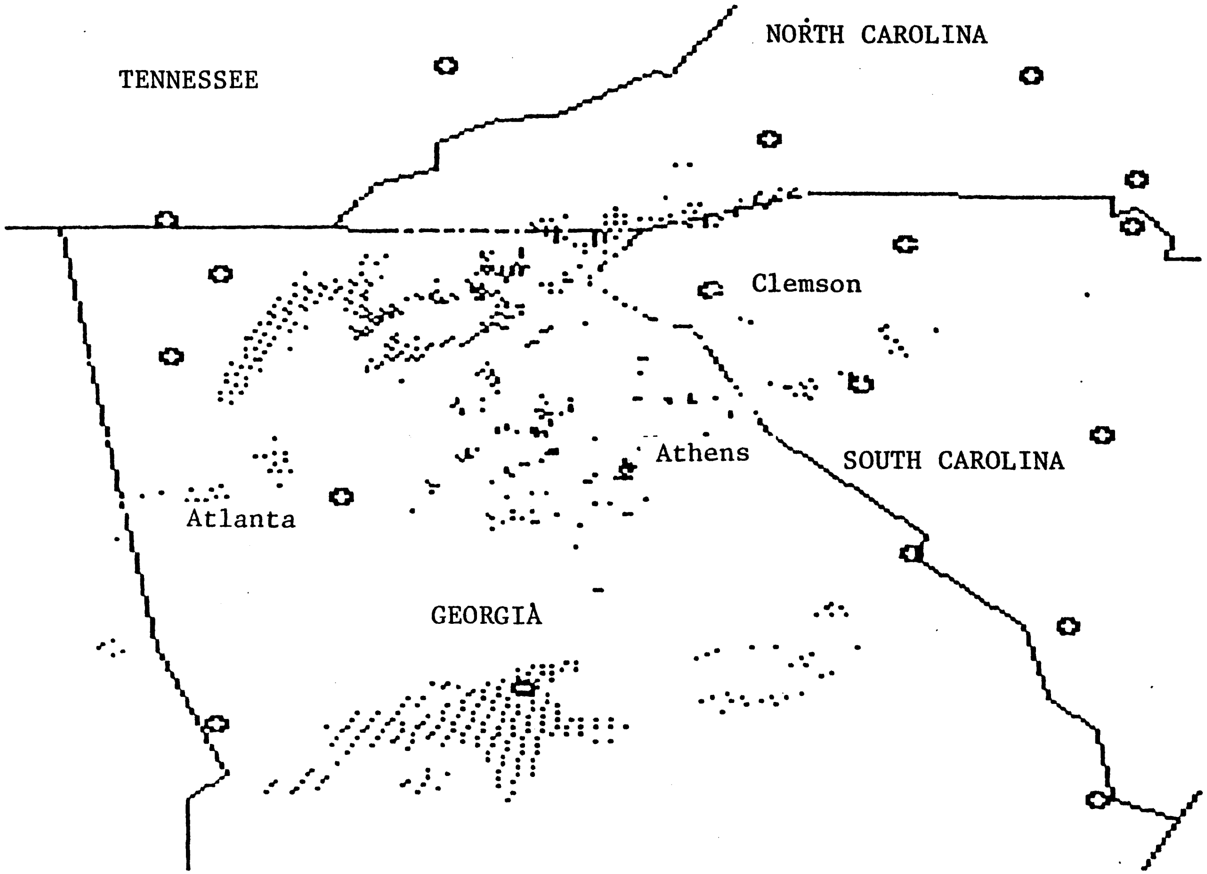
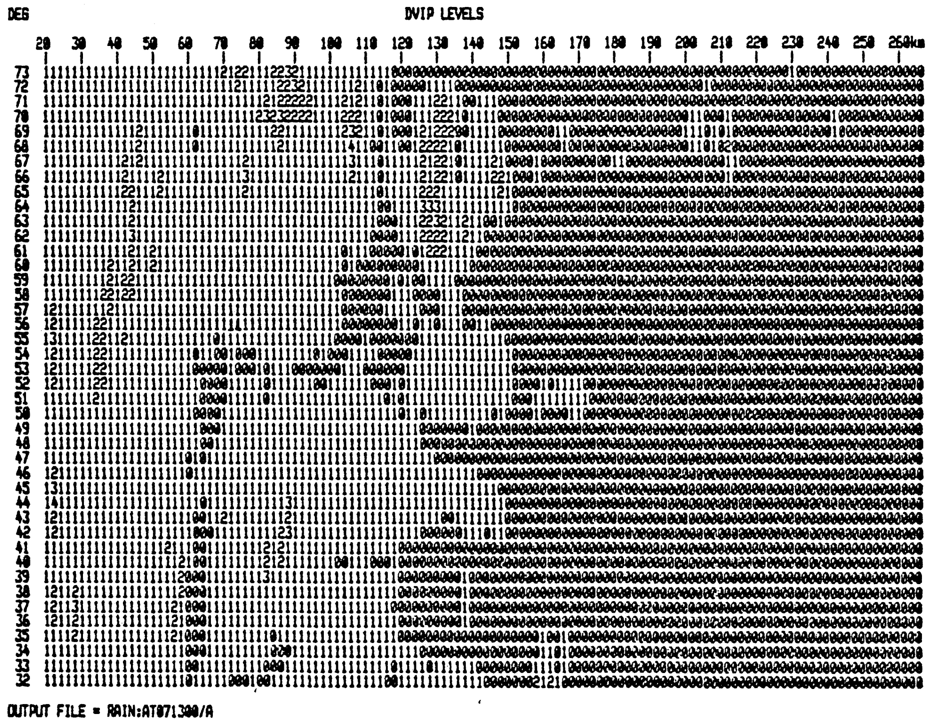


Figure 6 - Graphics Printer Output Of Radar Rainfall Event (Athens, Ga., June 7, 1983, 1300 GMT) as Produced by RADARQUAD. DVIP levels greater than level 1 are displayed.



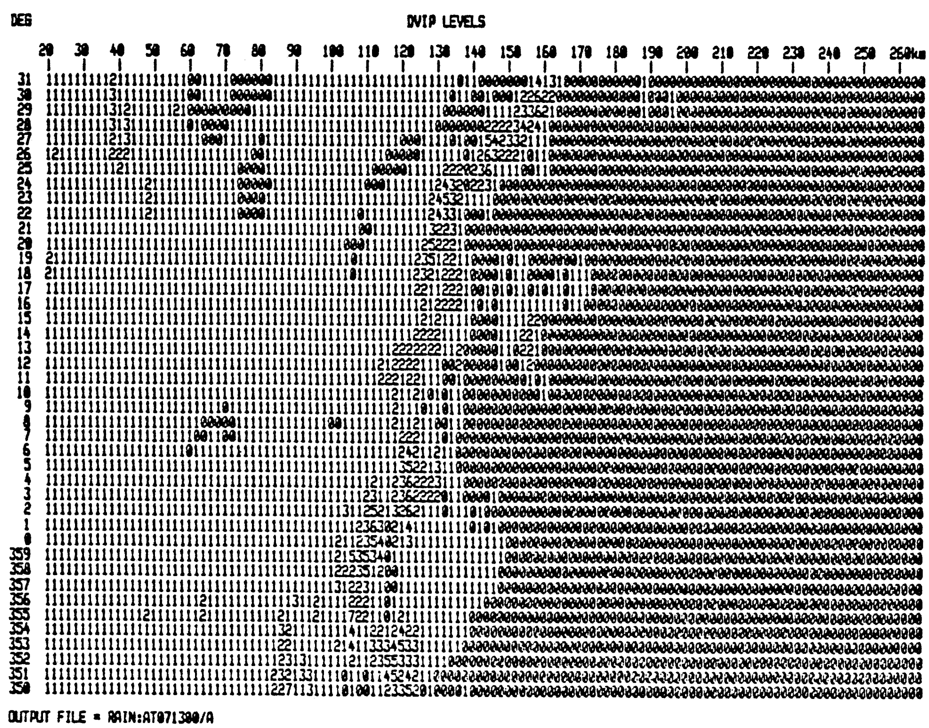


Figure 7. - Alphanumeric Printer Output Of Radar Rainfall Event (Athens, Ga., June 7, 1983, 1300 GMT) as Produced by RADARPRINT. Printout of two pages of output data includes DVIP levels for Clemson, S.C., and Clemson Watershed.

Figure 8. Sample Printout from Runoff Model (FORCSTR)

INITIAL CONDITIONS

INTERCEPTION PARAMETER	ANTECEDENT SOIL MOIST.	MAXIMUM INTERCEPTION	MAXIMUM SOIL MOIST.	DEPLETION EXPONENT
.2	1.4	.1	1.5	0

TEMPERATURE = 82.5
TEMPERATURE = 78.5

DATA FOR INTERVAL(S)

SAMPLING INTERVAL	START TIME THIS INTERVAL	RAINFALL IN THIS INTERVAL	SOIL MOISTURE	POTENTIAL RUNOFF
15.00	0.00	0.013	1.408	0.000
29.00	0.25	0.024	1.426	0.000
12.00	0.73	0.010	1.433	0.000
13.00	0.93	0.009	1.438	0.000
12.00	1.15	0.010	1.444	0.000
19.00	1.35	0.034	1.470	0.000
43.00	1.67	0.036	1.497	0.000

Figure 8 ... Continued

DATA FOR INTERVAL(S)

SAMPLING INTERVAL	START TIME THIS INTERVAL	RAINFALL IN THIS INTERVAL	SOIL MOISTURE	POTENTIAL RUNOFF
32.00	2.38	0.027	1.500	0.016

CONVOLUTION

Time(Hr)	Flow(In/Hr)
0.25	0.000
0.73	0.000
0.93	0.000
1.15	0.000
1.35	0.000
1.67	0.000
2.38	0.000
2.63	0.000
2.88	0.001
3.13	0.001
3.38	0.002
3.63	0.002
3.88	0.002
4.13	0.001
4.38	0.001
4.63	0.000
4.88	0.000

DATA FOR INTERVAL(S)

SAMPLING INTERVAL	START TIME THIS INTERVAL	RAINFALL IN THIS INTERVAL	SOIL MOISTURE	POTENTIAL RUNOFF
39.00	2.92	0.000	1.498	0.000

saturation, the number of temperature observations (forecasts) and the corresponding temperature values.

Statements starting with word "ENTER" are self prompts to enable the user to know which parameters to enter. Given the rainfall rate and the sampling interval the program determines the potential runoff. If this is zero the program skips the convolution routine to await further rainfall observation. For this illustrative example we have used the mid range of the radar rainfall.

RECOMMENDATIONS

Areas Requiring Further Development

This system was a preliminary attempt at radar data acquisition which was meant to show the feasibility of a process. While the project goals were met, further development of the system is required to make it operationally useful. Specifically, the system needs to (1) operate faster, (2) provide automatic data collection and (3) be operationally verified.

To obtain these improvements, a major change in equipment must be made. The system currently uses two separate microcomputers. The INTEL 8088, which is used for data collection and formatting, is an older system which operates at a fairly slow rate. This necessitates separation of collection and processing procedures to allow acceptable time intervals between radar events. Post-processing must be performed on the collected files in order to transfer them to the application programs. Faster operating equipment would enable processing of the collected files between radar events, making the data available in real time. Some of this problem would be alleviated if all programs in this system were run on one machine. This would eliminate some of the file creation and handling which further slows the process.

Another problem of the system is the amount of personnel time needed to collect the data. As the system is currently configured, an operator is required to manually load programs, dial the radar transmitter and ensure the proper storage of files. For relatively short duration rainfall events this is only an inconvenience. For events which last longer than several hours, it can become a tiring and expensive procedure. Automatic dialing modems and multitasking programs could alleviate this problem by making the data collection process automatic. Data collection and processing program could operate as high priority tasks, while application programs could operate at lower priorities. This type of system would not only be more cost efficient, but would be more reliable due to decreased chances of operator errors.

Verification of the accuracy of the collected data needs to be made against a collection standard. This can be done by performing radar-raingage comparisons under a variety of conditions. Comparisons should be made for type of rainfall, terrain, season and radar antenna elevation, to name a few. This data then needs to be applied against current collection methods to determine the best method of utilization of the data.

Selecting the System

The requirements placed on the microcomputer allow for more flexibility in the choice of equipment. Characteristics to consider include the amount of random access memory (RAM), disk storage capacity, processor speed, synchronous data port capability, high level language support and graphic display characteristics. Secondary considerations should be given to the microcomputer's ability to perform tasks in excess to the radar project (ex. - word processing, database management), the reliability of the microcomputer company and availability of software to do secondary tasks.

The availability of a synchronous communications port should be a primary concern. Without this funciton, data cannot be entered from the modem to the microcomputer. This microcomputer specifications should declare a universal synchronous/asynchronous receiver/transmitter (USART) chip as its input/output hardware.

In order to handle the data transfer and processing, a high level, compiler language such as PASCAL, PL/I or FORTRAN must be available. These are required to handle the control of the input port and assembly language calls required to collect and process data on a real-time basis. An interpreter language such as BASIC would be too slow to do data processing between events.

Storage of the incoming data must be handled without disk access. The RAM must be large enough to handle the collection and processing programs as well as the collected data. The data storage area will vary depending on the amount of signal processed, but will be approximately 22KB for a radar coverage area of 276 km (radius). Data is available for 450 km which would require approximately 40KB of data storage. For this reason, a minimum configuration of 128KB RAM is recommended. Once the data is collected, it must be stored on a disk for permanent/semi-permanent archival. Converting the collected, binary data to an ASCII format further increases its size to 53KB for the 276 km areal radius. Disk storage capacity should allow for storing as many events on a single disk as possible. Furthermore, a two-drive system should be used so that one drive can be reserved for program storage while the second drive is used for data storage. Two drives will also facilitate program development and source program backup copying.

Processor speed should be another consideraton in choosing a microcomputer. The processor clock frequency should be as fast as possible in order to allow data processing during the collection process. While this characteristic is not a definitive measure of processing speed, due to differences in hardware architecture, a faster clock will allow performance of more instructions per second. If the processor is too slow, data processing cannot be performed in real-time.

The final consideration should be given to graphics capabilities. In order to give an adequate graphic representation of the radar data, high resolution color graphics should be used. Since six levels of rainfall plus a background color will have to be displayed, a microcomputer capable of displaying these seven colors should be used. Horizontal and vertical resolution should be high enough to provide reasonable detail. In our study, four displayed colors were available with a 239 H x 239 V pixel resolution. While this was adequate to display data, availability to display all rainfall levels in a separate color would have improved the visual display.

SUMMARY AND CONCLUSIONS

Radar measurement of rainfall offers a valuable method for the estimation of areal rainfall. It provides spatial information about rainfall events which are not readily measured by other methods. Furthermore, it allows for data acquisition in near real time. Heavy rainfall events often lead to correspondingly intense runoff and flash floods. Access to radar data at a local level will enable relief agencies to quickly target their efforts into areas with greatest needs.

This report presented a microcomputer based system with which radar data can be accessed, processed and applied to practical forecasting situations.

Further system development can provide rainfall information needed to improve a variety of hydrologic and agricultural modelling procedures.

Lastly, this system will be inexpensive and hence affordable by the small scale user if access to the radar data can be made at a nominal fee. This is possible if the non-government owner of the radar transmitter attached to the NWS radar offers a low cost alternative to current access mechanism, for example, an automatic billing mechanism using a 900 type area code.

REFERENCES

Bigler, S. G., R. C. McGrew and M. St. Clair, 1970, An Experiment in Digitizing Weather Radar Data from a Four Station Network, Preprints, 14th Radar Meteorology Conference, American Meteorological Society, Tucson, AZ.

Curry, R. G., R. A. Clark and R. C. Runnels, 1970, "Hydrograph Synthesis from Digitized Radar data by Streamflow Routing," Preprints, 14th Radar Meteorology Conference, American Meteorological Society, Tucson, AZ.

Curry, R. G., 1973, "A Study of the Application of Weather Radar in Streamflow Forecasting," Dept. of Oceonography and Meteorology, Texas A & M University, College Station, TX, Report TR-4.

Dunn, L. B., 1982, "AFOS Era Automated Forecast Verification," Preprints, Ninth Conference on Weather Forecasting and Analysis, American Meteorological Society, Seattle, WA.

Fereres, E., R. E. Goldfien, W. O. Pruitt, D. W. Henderson and R. M. Hagan, 1981, "The Irrigation Management Program: A New Approach to Computer Assisted Irrigation Scheduling," American Society of Agricultural Engineers, Publ. 23-81.

Grayman, W. M. and P. S. Eagleson, 1970, "The Use of Radar Measurements in the Prediction of Streamflow Hydrographs," Preprints, 14th Radar Meteorology Conference, American Meteorological Society, Tucson, AZ.

Greene, D. R., 1973, "Numerical Techniques for the Analysis of Digital Radar Data with Applications to Meteorology and Hydrology," Dept. of Oceonography and Meteorology, Texas A & M University, College Station, TX, Report TR-7.

Greene, D. R. and R. A. Clark, 1974, "The operational use of digital radar data for flash flood monitoring," Flash Floods Symposium IAHS-AISH Publ. 112, American Geophysical Union, Washington, DC.

Greene, D. R., 1975, "Hydrologic Application of Digital Radar Data," Preprints, 16th Radar Meteorology Conference, American Meteorological Society, Houston, TX.

Harrington, G. J. and D. F. Heermann, 1981, "State of the Art Irrigation Scheduling Computer Program," American Society of Agricultural Engineers, Publ. 23-81.

Heffernan, M. M. and H. R. Glahn, 1980, "AFOS-Era Forecast Verification Program," Preprints, Eighth Conference on Weather Forecasting and Analysis, American Meteorological Society, Denver, CO.

Hudlow, M. D., 1972, "Use of Radar Data from D/RADEX for Operational Hydrology," Preprints 15th Radar Meteorology Conference of American Meteorological Society, Boston, MA.

Hummels, D. R., 1975, "A Digitized Radar for Precipitation Measurements and Applications to Hydrology," Kansas Water Resources Research Institute, Kansas State University, Manhattan, KS, Report No. 170.

Jensen, M. E., 1969, "Scheduling Irrigation Using Computers," Journal of Soil and Water Conservation, Vol. 24, No. 5.

Jensen, M. E., D. C. N. Robb and C. E. Franzoy, 1970, "Scheduling Irrigation Using Climate Crop-Soil Data," American Soc. Civil Engr., Journal of Irrigation and Drainage Division, Vol. 96, IR1.

Kincaid, D. C. and D. F. Heermann, 1974, Scheduling Irrigations Using a Programmable Calculator, U.S.D.A. Bulletin, ARS-NC-12.

Klein, W. H., 1976, "The AFOS Program and Future Forecast Applications," Monthly Weather Review, 104.

Lambert, J. R., G. W. Doy and V. L. Quisenberry, 1981, "Irrigation Scheduling in Humid Areas," American Society of Agricultural Engineers, Publ. 23-81.

Lu, C., 1982, "Microcomputers: The Second Wave," High Technology, Sept./Oct. 1982.

Mathewson, M. A., 1982, "ARAP (AFOS Radar Processor)," Preprints, Ninth Conference on Weather Forecasting and Analysis, American Meteorological Society, Seattle, WA.

Mielke, K. B., 1979a, "AFOS Crop and Soil Information Report Program," Western Region Computer Programs and Problems, National Weather Service, WCRP No. 2.

Mielke, K. B., 1979b, "A Computer Program for Convective Parameters," National Weather Digest, Vol. 4. No. 3.

Mielke, K. B., 1982, "Operational Forecasting in the AFOS Era" Preprints, Ninth Conference on Weather Forecasting and Analysis, American Meteorological Society, Seattle, WA.

Sisk, D. D., 1975, "Digital Radar Data and its Application in Flash Flood Forecasting" NWS Technical Memo, NOAA-TM-NWS-ER-59.

Tetzloff, R. P., 1976, "The Adequacy of Digitized Radar Data for Operational River Forecasting," Preprints, 16th Radar Meteorology Conference, American Meteorological Society, Seattle, WA.

Wasserman, S. E., 1980, "Real Time Radar and Satellite Information Available for Operational Short-Range Forecasting," Preprints, Eighth Conference on Weather Forecasting and Analysis of the American Meteorological Society, June 10-13, Denver, CO.

Wiggert, J. W. and E. A. Brandes, 1979, "Computerized Rain Assessment and Tracking of South Florida Weather Radar Echoes," Bull. American Meteorological Society, Vol. 14(1).

APPENDIX 1

```
/*      RADARP.PLI       */
/*                       */
/* PROGRAM TO GATHER BINARY DATA FROM SYNCHRONOUS MODEM, AND STORE IT ON DISK.
   CP/M SYSTEM FUNCTIONS ARE CALLED.
   SUBROUTINE PORT IS ASSEMBLER LANGUAGE TO INITIALIZE THE SERIAL HARDWARE.
   SUBROUTINE HUNT IS ASSEMBLER TO CAUSE THE 2651 TO ENTER HUNT MODE,
   LOOKING FOR TWO SYNC BYTES.

   TO RUN THE RADAR DATA GATHERING PROGRAM, FIRST RUN THE
   INPUT/OUTPUT PORT INITIALIZATION PROGRAM BY TYPING:

   io io

   THE PRESENT VERSION OF IO.COM STRIPS OFF THE MOST SIGNIFICANT
   BIT OF EACH BYTE COMMING INTO THE READER.  THE FOLLOWING
   PATCH WILL DISABLE THE STRIPPING.  USER RESPONSE IS IN
   LOWER CASE, EXCEPT THE ZEROS, WHICH ARE ALSO USER SUPPLIED.

   ddt
   DDT VERS 1.4
   sfe9f
   E6 00
   7F 00
   C9 .
   -g0

   AFTER THIS HAS BEEN DONE, THE LOGICAL DEVICE "USER READER #2"
   SHOULD BE ASSIGN AS READER INPUT AS FOLLOWS:

   stat rdr: = ur2:

   THE WEATHER RADAR COLLECTION PROGRAM ASSUMES THAT THE
   RADAR MODEM IS CONNECTED TO PORT 2, MARKED WITH A "2"
   ON THE BACK OF THE DEVELOPMENT COMPUTER.  THE PROGRAM
   MAY NOW BE RUN BY TYPING:

   radarp

   28 MARCH 1983  -- JLD
*/
RADAR:
    PROCEDURE OPTIONS(MAIN);

      %REPLACE DEG_LIM BY 360; /* NUMBER OF "RADIANS" TO COLLECT */
      %REPLACE WORD_LIM BY 62; /* MAX WORDS PER RADIAN
        /* The specification calls for: (26-8+(30*26))/8=100-->120 */
      DECLARE
```

```
    PORT ENTRY,
    WORD_MARK FIXED,
    DEGREE_MARK FIXED,
    TEMPCHR FIXED(8),
    RAW_RADIAL(DEG_LIM, WORD_LIM) FIXED(8); /* ARRAY HOLDS RADAR IMAGE*/

  CALL PORT(); /* INITIALIZE I/O PORT FOR SYNCHRONOUS OPERATION. */
  CALL BITSIN; /* GET ONE ENTIRE RADIAL FROM SYNCHRONOUS DATA PORT. */
  CALL TIDY; /* SAVE IMAGE ON DISK, THEN STOP PROGRAM. */
  STOP; /* THAT'S ALL. */

 BITSIN:
  PROCEDURE;
  DECLARE
    HUNT ENTRY,
    RDSTAT ENTRY RETURNS( FIXED(7) ),
    RDRDR ENTRY RETURNS( CHAR(1) );
  DO DEGREE_MARK = 1 TO DEG_LIM;
    PUT SKIP LIST(DEGREE_MARK);
   REHUNT:
    CALL HUNT();
    IF (RANK(RDRDR()) ^= 0) THEN GOTO REHUNT;
    IF (RANK(RDRDR()) ^= 0) THEN GOTO REHUNT;
    DO WORD_MARK = 1 TO WORD_LIM;
      TEMPCHR = RANK(RDRDR());
      RAW_RADIAL(DEGREE_MARK, WORD_MARK) = TEMPCHR;
    END;
  END;
 END BITSIN;

  TIDY:
  PROCEDURE;
    DCL SAVE FILE;
    DCL OUT_STRING CHAR(WORD_LIM);
    OPEN FILE(SAVE) OUTPUT SEQUENTIAL TITLE('B:RADAR.DAT');
    DO DEGREE_MARK = 1 TO DEG_LIM;
      PUT SKIP LIST(DEGREE_MARK);
      DO WORD_MARK = 1 TO WORD_LIM;
        SUBSTR(OUT_STRING,WORD_MARK,1) =
              ASCII(RAW_RADIAL( DEGREE_MARK, WORD_MARK ));
      END;
      WRITE FILE(SAVE) FROM(OUT_STRING);
    END;
    CLOSE FILE(SAVE);
  END TIDY;

END RADAR;
```

```
****** RADARM.MAC ******

; RADAR INPUT DRIVER
; 11/18/82 -- JLD
;
        CSEG
        ENTRY   PORT, HUNT
BASE    EQU     08H
DATARG  EQU     BASE+0
STATUS  EQU     BASE+1
MODE    EQU     BASE+2
COMAND  EQU     BASE+3

MWORD1  EQU     00001100B
; D7    DOUBLE SYNC
; D5-6  NORMAL ( NOT TRANSPARENT )
; D4    NO PARITY
; D2-3  8-BIT WORD
; D0-1  SYNCHRONOUS

MWORD2  EQU     000H

COMWD   EQU     00100010B
; D6-7  NORMAL (NO SYN-DLE STRIPPING)
; D5    RTS ACTIVE
; D4    NOT RESETTING ERRORS
; D3    NO DLE
; D2    RECEIVER NOT ENABLED YET
; D1    DTR ACTIVE
; D0    TRANSMITTER NOT ENABLED

SYN1    EQU     0FFH
SYN2    EQU     00H

; WHEN OR'D WITH MODE WORD 1 MWORD1 THESE TWO VALUES WILL ENABLE
; AND DISABLE THE RECIEVER SECTION OF THE 2651.

RCVON   EQU     0100B
RCVOFF  EQU     (NOT RCVON) AND OFFH

; CODE STARTS HERE.

PORT:
        PUSH    PSW
        IN      COMAND  ;RESET MODE REGISTER POINTER.
        MVI     A,MWORD1
        OUT     MODE
```

```
        MVI     A,MWORD2
        OUT     MODE
        MVI     A,SYN1
        OUT     STATUS
        MVI     A,SYN2
        OUT     STATUS
        MVI     A,COMWD
        OUT     COMAND
        POP     PSW
        RET

; THIS SUBROUTINE FORCES THE SERIAL INPUT PORT INTO THE HUNT MODE;

HUNT:
        PUSH    PSW
        MVI     A,COMWD AND RCVOFF
        OUT     COMAND
        MVI     A,COMWD OR RCVON
        OUT     COMAND
        POP     PSW
        RET

        END
```

```
****** DIOCALL1.MAC ******

        TITLE   'DIRECT CP/M CALLS FROM PL/I-80'
;
        PUBLIC  DFCB0   ;RETURNS ADDRESS OF DEFAULT FCBP
        PUBLIC  DFCB1   ;RETURN ADDRESS OF DEFAULT FCB1
        PUBLIC  SEAR    ;SEARCH FOR FILE (#17)
        PUBLIC  SEARN   ;SEARCH FOR NEXT(#18)
        PUBLIC  SETDMA  ;SET NEW DMA.
        PUBLIC  GETDPB  ;GET BASE OF DISK PARM BLOCK (#31).
        PUBLIC  RESDSK  ;RESET DISK SYSTEM
        PUBLIC  RESDRV  ;RESET DISK SYSTEM, #37.
        PUBLIC  RDSTAT  ;CONSOLE STATUS
        PUBLIC  RDRDR   ;READER CHARACTER INPUT (#3)
;
        EXTRN   ?BEGIN  ;BEGINNING OF FREE LIST
        EXTRN   ?BOOT   ;SYSTEM REBOOT ENRY POINT
        EXTRN   ?BDOS   ;BDOS ENTRY POINT
        EXTRN   ?DFCB0  ;DEFAULTFCB 0
        EXTRN   ?DFCB1  ;DEFAULT FCB 1
        EXTRN   ?DBUFF  ;DEFAULT BUFFER
;
        PRINTF  EQU     9       ;PRINT STRING FUNCTION.
        SERCHF  EQU     17
        VERSF   EQU     12      ;GET CURRENT VERSION NUMBER.
        RDRF    EQU     3
        RESETF  EQU     13      ;RESET DISK SYSTEM.
        SERCHN  EQU     18
        SETDMF  EQU     26
        GETDPF  EQU     31      ;GET DISP PARAMETER BLOCK.
        RSDRVF  EQU     37
        CR      EQU     0DH
        LF      EQU     0AH
;
GETP2:
GETP2I:
        CALL    GETP1
        INX     H
        MOV     D,M
        RET
;
GETP1:
        MOV     E,M
        INX     H
        MOV     D,M
        XCHG
        MOV     E,M
```

```
        RET
;
DFCB0:
        LXI     H,?DFCB0
        RET
;
DFCB1:  LXI     H,?DFCB1
        RET
;
SEAR:   CALL    GETP2I
        MVI     C,SERCHF
        JMP     ?BDOS
;
SEARN:  MVI     C,SERCHN
        JMP     ?BDOS
;
SETDMA:
        CALL    GETP2
        MVI     C,SETDMF
        JMP     ?BDOS
;
GETDPB:
        CALL    CHKV20  ;CHECK FOR 2.0 OR GREATER
        MVI     C,GETDPF
        JMP     ?BDOS
;
RESDSK: MVI     C,RESETF
        CALL    ?BDOS
        LXI     H,080H
        RET
;
RESDRV: ;RESET DRIVE, FUNCTION #37.
                ;HL--> DRIVE VECTOR BIT(16)).
        CALL    CHKV22
        CALL    GETP2
        MVI     C,RSDRVF
        JMP     ?BDOS
GETVER:
        PUSH    H
        MVI     C,VERSF
        CALL    ?BDOS
        POP     H
        RET
;
CHKV20:
        CALL    GETVER
        CPI     20
        RNC
```

```
        JMP     VERERR
;
CHKV22:
        CALL    GETVER
        CPI     22H
        RNC
;
; FALL ON THROUGH IF NOT A 2.2 SYSTEM.
;
VERERR:
        LXI     D,VERMSG
        MVI     C,PRINTF
        CALL    ?BDOS
        JMP     ?BOOT
VERMSG:
        DB      CR,LF,'LATER CP/M OR MP/M VERSION REQUIRED$'

RDSTAT:                  ;DIRECT CONSOLE STATUS READ.
        LXI     H,RDSRET        ;READ STATUS RETURN
        PUSH    H               ;RETURN TO RDSRET
        LHLD    ?BOOT+1         ;BASE OF JMP VECTOR
        LXI     D,1*3           ;OFFSET TO .JMP CONST
        DAD     D               ;HL = .JMP CONST
        PCHL

RDSRET:                         ;RETURN CLEAN TRUE VALUE
        ORA     A               ;ZERO?
        RZ                      ;RETURN IF SO
        MVI     A,OFFH          ;CLEAN TRUE VALUE.
        RET

RDRDR:                          ;READ READER CHARACTER (#3)
        MVI     C,RDRF          ;READER FUNCTION
CHRIN:
        ;COMMON CODE FOR CHARACTER INPUT
        CALL    ?BDOS           ;VALUE RETURNED TO A
        POP     H               ;RETURN ADDRESS
        PUSH    PSW             ;CHARACER TO STACK
        INX     SP              ;DELETE FLAGS
        MVI     A,1             ;CHARACTER LENGTH IS 1
        PCHL
;
        END
```

```
/*      FLIPOVER.PLI       */
/*                         */
FLIPOVER:
    PROCEDURE OPTIONS(MAIN);

      %REPLACE DEG_LIM BY 360; /* NUMBER OF "RADIANS" TO COLLECT */
      %REPLACE WORD_LIM BY 62; /* MAX WORDS PER RADIAN */
      %REPLACE OUT_LIM BY 57;
/*  */
      DCL DORAD ENTRY(CHAR(OUT_LIM),CHAR(WORD_LIM));
      DECLARE
        DEGREE_MARK FIXED;
      DCL SAVE FILE;
      DCL MESS FILE;
      DCL IN_STRING CHAR(WORD_LIM);
      DCL OUT_STRING CHAR(OUT_LIM);
/*  */
        OPEN FILE(SAVE) OUTPUT SEQUENTIAL TITLE('B:RADAR.FLP');
        OPEN FILE(MESS) INPUT SEQUENTIAL TITLE('B:RADAR.DAT');
        DO DEGREE_MARK = 1 TO DEG_LIM;
          PUT SKIP LIST(DEGREE_MARK);
          READ FILE(MESS) INTO(IN_STRING);
          CALL DORAD(OUT_STRING,IN_STRING);
          WRITE FILE(SAVE) FROM(OUT_STRING);
        END;
        CLOSE FILE(SAVE);
        CLOSE FILE(MESS);

  END FLIPOVER;
```

```
        ****** DORAD.MAC ******
;
;       DORAD(OUT_STRING, IN_STRING)
;
;       STRIP FIRST TWO BITS (FROM SYNCH),
;         THEN STRIP FIRST AND LAST BIT FROM EACH 26-BIT GROUP.
;         ALSO REVERSE EACH RESULTING BYTE.
;
;
                CSEG
                ENTRY   DORAD
;
DORAD           EQU     $
                PUSH    PSW                 ;SAVE ALL REGS
                PUSH    B
                PUSH    D
                PUSH    H
;
                MOV     E,M                 ;FETCH POINTERS TO STRINGS
                INX     H
                MOV     D,M
                INX     H
                MOV     C,M
                INX     H
                MOV     B,M
                MOV     H,B
                MOV     L,C
;
                MOV     A,M                 ;GET FIRST SOURCE BYTE
                INX     H                   ;POINT TO NEXT BYTE
                RAR                         ;STRIP TWO BITS
                RAR
                MOV     B,A
                MVI     C,6                 ;SIX GOOD BITS LEFT
;
                MVI     A,19                ;CONVERT 19 26-BIT GROUPS
LOOP26          EQU     $
                CALL    GET26
                DCR     A
                JNZ     LOOP26
                POP     H
                POP     D
                POP     B
                POP     PSW
;
```

```
                RET
        ;
                PAGE
        ;
        ;     GET26 -- STRIP FIRST AND LAST BITS FROM 26-BIT STRING,
        ;                INVERT EACH OF RESULTING BYTES.
        ;
        GET26   EQU     $
                PUSH    PSW             ;SAVE ALL REGS
                CALL    ROTOFF          ;DROP FIRST BIT
                MVI     A,3             ;3 BYTES
        FLIPLP  EQU     $
                CALL    FLIP
                DCR     A
                JNZ     FLIPLP
                CALL    ROTOFF          ;DROP LAST BIT
        ;
                POP     PSW
        ;
                RET
        ;
                PAGE
        ;
        ROTOFF  EQU     $
                PUSH    PSW
                MOV     A,B             ;USE CURRENT BYTE
                RAR
                MOV     B,A
                DCR     C               ;ONE LESS BIT
                JNZ     NONEW           ;USED ALL THESE BITS?
        ;
                MOV     B,M             ;GET NEXT BYTE
                INX     H               ;UPDATE POINTER
                MVI     C,8             ;8 GOOD BITS HERE
        ;
        NONEW:  POP     PSW
                RET
        ;
                PAGE
        ;
        FLIP    EQU     $
                PUSH    PSW
                PUSH    D
        ;
                MVI     D,8             ;NUMBER OF BITS TO FLIP
```

```
        XRA     A
        STA     NEWBYT
;
FLOOP   EQU     $
        MOV     A,B             ;GET A BIT
        RAR
        MOV     B,A
        LDA     NEWBYT
        RAL
        STA     NEWBYT
;
        DCR     C               ;USED UP THIS SOURCE BYTE?
        JNZ     NOTNEW
        MOV     B,M
        MVI     C,8
        INX     H
;
NOTNEW  EQU     $
        DCR     D               ;FLIPPED 8 BITS?
        JNZ     FLOOP
        POP     D
        XCHG
        LDA     NEWBYT
        MOV     M,A
        XCHG
        INX     D
;
        POP     PSW
        RET
;
        DSEG
NEWBYT: DS      1
;
        END
```

```
FILERAD:
    PROCEDURE OPTIONS(MAIN);

      %REPLACE DEG_LIM BY 360; /* NUMBER OF "RADIANS" TO COLLECT */
      %REPLACE OUT_LIM BY 57;
/*  */
/*  */
      DCL (DEGREE_MARK, WORD_NUM) FIXED;
      DCL SAVE FILE;
      DCL OUTFIL FILE;
      DCL IN_STRING CHAR(OUT_LIM);
      DCL ANSWER CHAR(30) VARYING;
/*  */
/*  */
        OPEN FILE(SAVE) INPUT SEQUENTIAL TITLE('B:RADAR.FLP');
        OPEN FILE(OUTFIL) OUTPUT STREAM TITLE('B:RADAR.XFR');

        DO DEGREE_MARK = 1 TO DEG_LIM;
          READ FILE(SAVE) INTO(IN_STRING);
          WORD_NUM = 2;    /* SKIP WORD OF ZEROS */
          CALL SHOW_TIME;
          CALL SHOW_LOCATION;
          CALL SHOW_RAIN;
        END;
        CLOSE FILE(SAVE);

/*   */
/*   */
/*   */
/*   */
/*   */
SHOW_TIME:
****************************************
*                                      *
*  PROGRAM CODE FOR THIS SECTION IS    *
*  NOT INCLUDED IN THIS LISTING IN     *
*  ORDER TO PROTECT PROPRIETARY        *
*  INFORMATION OF ENTERPRISE           *
*  ELECTRONICS CORPORATION             *
*                                      *
****************************************
  END SHOW_TIME;
/*   */
/*   */
/*   */
/*   */
/*   */
SHOW_LOCATION:
```

```
****************************************
*                                      *
*  PROGRAM CODE FOR THIS SECTION IS    *
*  NOT INCLUDED IN THIS LISTING IN     *
*  ORDER TO PROTECT PROPRIETARY        *
*  INFORMATION OF ENTERPRISE           *
*  ELECTRONICS CORPORATION             *
*                                      *
****************************************
  END SHOW_LOCATION;
/*    */
/*    */
/*    */
/*    */
/*    */
SHOW_RAIN:
****************************************
*                                      *
*  PROGRAM CODE FOR THIS SECTION IS    *
*  NOT INCLUDED IN THIS LISTING IN     *
*  ORDER TO PROTECT PROPRIETARY        *
*  INFORMATION OF ENTERPRISE           *
*  ELECTRONICS CORPORATION             *
*                                      *
****************************************
  END SHOW_RAIN;
  END FILERAD;
```

```
/*      SHOWRAD.PLI       */
/*                        */
SHOWRAD:
    PROCEDURE OPTIONS(MAIN);

      %REPLACE DEG_LIM BY 360; /* NUMBER OF "RADIANS" TO COLLECT */
      %REPLACE OUT_LIM BY 57;
/*  */
/*  */
      DCL (DEGREE_MARK, WORD_NUM) FIXED;
      DCL SAVE FILE;
      DCL IN_STRING CHAR(OUT_LIM);
      DCL ANSWER CHAR(30) VARYING;
      DCL (QUIT_FLAG, ROLL_FLAG) BIT(1);
/*  */
        OPEN FILE(SAVE) INPUT SEQUENTIAL TITLE('B:RADAR.FLP');

        QUIT_FLAG = '0'B;
        ROLL_FLAG = '0'B;

        DO DEGREE_MARK = 1 TO DEG_LIM WHILE(QUIT_FLAG = '0'B);
          READ FILE(SAVE) INTO(IN_STRING);
          WORD_NUM = 2;     /* SKIP WORD OF ZEROS */
          CALL SHOW_TIME;
          CALL SHOW_LOCATION;
          CALL SHOW_RAIN;
          IF ROLL_FLAG = '0'B THEN

          DO;
            PUT SKIP(2) LIST('TYPE "NEW LINE" TO SEE NEXT RECORD');
            PUT SKIP LIST('"C" FOR CONTINUOUS DISPLAY, "Q" TO QUIT');
            ANSWER = '';
            READ INTO(ANSWER);
            IF LENGTH(ANSWER) > 2 THEN
              IF SUBSTR(ANSWER,1,1) = 'Q' THEN
                QUIT_FLAG = '1'B;
              ELSE IF SUBSTR(ANSWER,1,1) = 'C' THEN
                ROLL_FLAG = '1'B;
          END;
        END;
        CLOSE FILE(SAVE);

/*    */
/*    */
/*    */
/*    */
/*    */
SHOW_TIME:
```

```
*************************************
*                                   *
*  PROGRAM CODE FOR THIS SECTION IS *
*  NOT INCLUDED IN THIS LISTING IN  *
*  ORDER TO PROTECT PROPRIETARY     *
*  INFORMATION OF ENTERPRISE        *
*  ELECTRONICS CORPORATION          *
*                                   *
*************************************
  END SHOW_TIME;
/*   */
/*   */
/*   */
/*   */
/*   */
SHOW_LOCATION:
*************************************
*                                   *
*  PROGRAM CODE FOR THIS SECTION IS *
*  NOT INCLUDED IN THIS LISTING IN  *
*  ORDER TO PROTECT PROPRIETARY     *
*  INFORMATION OF ENTERPRISE        *
*  ELECTRONICS CORPORATION          *
*                                   *
*************************************
  END SHOW_LOCATION;
/*   */
/*   */
/*   */
/*   */
/*   */
SHOW_RAIN:
*************************************
*                                   *
*  PROGRAM CODE FOR THIS SECTION IS *
*  NOT INCLUDED IN THIS LISTING IN  *
*  ORDER TO PROTECT PROPRIETARY     *
*  INFORMATION OF ENTERPRISE        *
*  ELECTRONICS CORPORATION          *
*                                   *
*************************************
  END SHOW_RAIN;
  END SHOWRAD;
```

```
/*      RADARXFR.PLI        */
/*                          */
RADARXFR:
  PROC OPTIONS(MAIN);
  DCL SATPUT EXTERNAL ENTRY(CHAR(80) VARYING);
  DCL SATGET EXTERNAL ENTRY(CHAR(80) VARYING);
  DCL OUTLINE CHAR(80) VARYING;
  DCL INLINE CHAR(80) VARYING;
  DCL INFILE FILE;
  DCL END_OF_FILE BIT(1);
  /*   */
  OPEN FILE(INFILE) STREAM INPUT TITLE('B:RADAR.XFR');
  END_OF_FILE = '0'B;
  ON ENDFILE(INFILE) END_OF_FILE = '1'B;
  DO WHILE (END_OF_FILE = '0'B);
    READ FILE(INFILE) INTO(OUTLINE);
    IF (END_OF_FILE = '0'B) THEN
    DO;
      CALL SATPUT(OUTLINE);
      CALL SATGET(INLINE);
      PUT SKIP LIST(OUTLINE);
    END;
    END;
    CLOSE FILE(INFILE);
    END RADARXFR;
```

```
****** SATGET.PLI ******
SATGET:
  PROC(INLINE);
  DCL CR CHAR(1) STATIC INIT('K');
  DCL I FIXED(15);
  DCL INCHR CHAR(1);
  DCL INLINE CHAR(80) VARYING;
  DCL SATIN EXTERNAL ENTRY(CHAR(1));
  INLINE = '';
  CALL SATIN(INCHR);
  DO WHILE (INCHR ¬= CR);
    INLINE = INLINE !! INCHR;
    CALL SATIN(INCHR);
  END;
  END SATGET;
```

Blank

```
10 ! *********** RADARTRANS ***********
20 ! *                                *
30 ! *         ASCII TRANSFER         *
40 ! *         OF RADAR DATA          *
50 ! *                                *
60 ! *   KEVIN ROBBINS- MAY 12,1983   *
70 ! *                                *
80 ! **********************************
90 !
100 ! *
110 ! DATA IS TRANSFERED AS ASCII CHAR'S
120 ! FROM THE INTEL DEVELOPEMENT SYSTEM
130 ! TO THE MONROE EC 8800.
140 ! *
150 ! 360 RADIALS ARE TRANSFERRED.
160 ! *
170 ! AFTER DATA IS TRANSFERRED, IT IS
180 ! STORED IN A DISC FILE ON
190 ! VOLUME "RAIN:".
200 ! *
210 EXTEND ! ALLOWS 32 CHARACTER VARIABLE NAMES
220 ! * DIMENSION CHARACTER STRING LENGTH
230 DIM B$=66% ! DATA CHARACTER STRING
240 DIM Ans$=1%,Name$=1%,N$=6%,File$=15%,Term$=2%
250 ! *
260 ; CHR$(12%) ! CLEARS LO-RES SCREEN
270 ! *
280 ; 'DISK VOLUME "RAIN:" MUST BE IN FPY0:'
290 ! *
300 ! * SCREEN PROMPT MESSAGE *
310 ; "TO CHANGE DISCS, TYPE THE FOLLOWING" : ;
320 ; YEL "-VOL RAIN:"
330 ; YEL "-CLOSE FPY0:"
340 ; 'INSERT VOLUME "RAIN:"'
350 ; YEL "-OPEN FPY0:"
360 ; YEL "CONT" : ;
370 ; "CHANGE DISCS? (Y/N)"
380 ! *
390 GET Ans$ ! TO CHANGE DISK
400 IF Ans$="Y" OR Ans$="y" PAUSE ! REVERT TO OPERATING SYSTEM
410 ! *
420 ; CHR$(12%)
430 ; "INPUT STATION "
440 ; "A  ATHENS"
450 ; "C  CHARLESTON"
460 ! *
470 ; : INPUT "STATION (A OR C) "Name$
480 ; "ENTER FILE DATE & TIME  ddhhmm"; : ; CUR(6%,22%); : INPUT N$
490 IF Name$="A" OR Name$="a" File$="RAIN:AT"+N$ ELSE File$="RAIN:CH"+N$
500 ; : ; "FILE NAME IS " File$
```

```
510 ! *
520 PREPARE File$ AS FILE 2 ! OUTPUT FILE
530 OPEN "AUX:R7" AS FILE 1 ! INPUT PORT
540 ! * AUX:  DENOTES COMMUNICATIONS PORT *
550 ! *  R7   DENOTES 9600 BAUD *
560 ! *
570 ; "COMMUNICATIONS PORT IS OPEN" : ;
580 ! *
590 ! * DATA TRANSFER SECTION *
600 Term$="AK" ! CNTRL CHAR SENT TO INTEL SIGNIFYING MONROE READY FOR INPUT
610 FOR I%=1% TO 360% ! 360 RADIALS OF DATA TRANSFER
620   ; CUR(11%,0%) I%
630   INPUT LINE #1,A$ : ; #1,Term$ ! TIME,ELEV,AZ
640   INPUT #1,B$ : ; #1,Term$ ! 64 BYTES DATA
650   INPUT #1,C$ ! 64 BYTES DATA
660   PRINT #2,A$; : ; A$
670   PRINT #2,B$ : ; LEFT$(B$,32%) : ; RIGHT$(B$,33%)
680   PRINT #2,C$ : ; LEFT$(C$,32%) : ; RIGHT$(C$,33%) : ;
690   ; #1,Term$
700 NEXT I% ! NEXT RADIAL
710 ! *
720 CLOSE 1 ! INPUT PORT
730 CLOSE 2 ! OUTPUT FILE
740 ! *
750 ; : ; : ; RED DBLE "TRANSFER COMPLETE"
760 ! *
770 STOP
```

```
10 ! ************ RADARQUAD ************
20 ! *                                *
30 ! *    PROGRAMMER-KEVIN ROBBINS    *
40 ! *     REVISED - JUN 04, 1983     *
50 ! *                                *
60 ! **********************************
70 !
80 !
90 EXTEND ! ALLOWS LONGER VARIABLE NAMES
100 ; CHR$(12%) ! CLEARS LO-RES SCREEN
110 !
120 DIM Ans$=1%,N$=6%,Name$=1%
130 DIM Radar$=16%,Continue$=1%
140 !
150 ; RED DBLE "RADAR PLOTTING PROGRAM"
160 ; CUR(4%,0%) 'DISK VOLUME "RAIN:" MUST BE IN FPY0:"'
170 ; "TO CHANGE DISKS, TYPE THE FOLLOWING"
180 ; CUR(7%,0%) YEL "-VOL RAIN:"
190 ; YEL "-CLOSE FPY0:"
200 ; 'CHANGE DISK TO VOLUME "RAIN"'
210 ; YEL "-OPEN FPY0:"
220 ; YEL "CONTINUE"
230 ; CUR(13%,0%) "CHANGE DISK? (Y/N)"
240 !
250 GET Ans$
260 IF Ans$="Y" OR Ans$="y" PAUSE ! BREAK TO OPERATING SYSTEM
270 !
280 ; CHR$(12%)
290 ; YEL "NAME OF RADAR?"; : ; WHT "(ENTER LETTER)" : ;
300 ; "   A. ATHENS"
310 ; "   B. CHARLESTON"
320 !
330 GET Name$
340 !
350 ; CUR(2%,0%) : ; CHR$(17%)
360 IF Name$="A" OR Name$="a" THEN ; CUR(3%,0%) "ATHENS" ELSE ; CUR(3%,0%) "CHARLE
    STON"
370 !
380 INPUT "ENTER EVENT DATE & TIME (ddhhmm) "N$
390 IF Name$="A" OR Name$="a" THEN Radar$="RAIN:AT"+N$+"/A" ELSE Radar$="RAIN:CH"+
    N$+"/A"
400 ; : ; "OPENING " Radar$
410 !
420 OPEN Radar$ AS FILE 2 ! OPEN DATA FILE
430 !
440 DIM Color%(128%) ! COLOR ARRAY
450 DIM Color1$=64%,Color2$=64% ! DATA INPUT CHARACTERS
460 DIM Hour$=2%,Min$=2% ! DATA INPUT CHARACTERS
470 DIM Zero$=1%
480 Zero$="0"
```

```
490 !
500 ; CHR$(12%)
510 ; "PLOT FULL SCREEN (F)"
520 ; "        OR"
530 ; "QUADRANTS (1,2,3,4)"
540 ; : ; "2  1" : ; "3  4" : ;
550 GET Ans$
560 IF Ans$="F" OR Ans$="f" THEN ; CHR$(12%) ELSE GOSUB 1760
570 !
580 ! ** HI-RES SCREEN INITIALIZATION **
590 FGCTL 33% ! SETS COLORS (BLACK,BLUE,MAGENTA,CYAN)
600 FGPOINT 0%,0%,0%
610 FGFILL 239%,239%
620 !
630 RESTORE 1340
640 GOSUB 1250 ! PLOT GAGES
650 RESTORE 1630 ! GAGE DATA SET
660 GOSUB 1360 ! PLOT MAP
670 ; CHR$(12%)
680 !
690 ! ** DEF. COLOR OF RAINFALL RATES **
700 ; CUR(0%,34%) BLU "1 ";CHR$(127%);" 4"
710 ; CUR(1%,34%) MAG "2 ";CHR$(127%);" 5"
720 ; CUR(2%,34%) CYA "3 ";CHR$(127%);" 6"
730 !
740 FGPOINT 119%,119%,3% ! RADAR LOCATION
750 !
760 Delrad%=2% : Plotfactx%=1% : Plotfacty%=1% ! PLOTTING PARAMETERS
770 Xorig%=119% : Yorig%=119%
780 FOR I%=1% TO 360% ! 360 RADIALS OF DATA FILE
790   !
800   GOSUB 2250 ! READS RADIAL FROM FILE
810   !
820   IF MOD(Azimuth%,2%)=1% GOTO 1090 ! SELECT EVEN # RADIALS FOR PLOTTING
830   ! ** SKIPS DATA IF ALL ZEROES **
840   IF COMP%(Color1$,Zero$) FOR J%=1% TO 64% STEP Delrad% : Color%(J%)=VAL(MID$(
      Color1$,J%,1%)) : NEXT J% ELSE GOSUB 1710
850   IF COMP%(Color2$,Zero$) FOR J%=65% TO 128% STEP Delrad% : Color%(J%)=VAL(MID
      $(Color2$,J%-64%,1%)) : NEXT J% ELSE GOSUB 1730
860   !
870   ! ** ANGULAR CONVERSION PARAMETERS **
880   Azimuth=-Azimuth%*PI/180%+PI/2%
890   U=SIN(Azimuth)
900   V=COS(Azimuth)
910   !
920   FOR Rad%=10% TO 118% STEP Delrad% ! STEPS ALONG A RADIAL
930     K%=Rad%-9% ! DATA ARRAY INDEX
940     IF Color%(K%)=0% OR Color%(K%)=7% GOTO 1060 ! TO NEXT RAD%
950     ! ELIMINATES THE MAPPING OF NULL RADAR DATA.
960     !
```

```
970     ! ** SETS FLAG TO MAP INTENSE RAINS **
980     IF Color%(K%)>3% Color%(K%)=Color%(K%)-3% : Intense%=1% ELSE Intense%=0%
990     ON ERROR GOTO 2310 ! TRAPS SCREEN PLOTTING ERRORS
1000      !
1010      ! ** PLOTS INDIVIDUAL DATA POINTS **
1020      FGPOINT Xorig%+(Rad%*Plotfactx%*V),Yorig%+(Rad%*Plotfacty%*U),Color%(K%)

1030      !
1040      IF Intense%=1% GOSUB 1210 ! PLOTS INTENSE RAINS (DVIP 4,5,6)
1050      !
1060    NEXT Rad%
1070    !
1080    IF Ans$="F" 1090 ELSE RETURN ! TO 1860 WHEN PLOTTING QUADRANT
1090 NEXT I%
1100 !
1110 IF Name$="A" OR Name$="a" ; CUR(0%,0%) YEL "ATHENS         " ELSE ; "CHARLEST
     ON"
1120 ; CUR(20%,0%)
1130 ; CUR(23%,0%) "PRINT SCREEN ? "; : GET Ans$
1140 IF Ans$="Y" OR Ans$="y" GOSUB 2320
1150 STOP
1160 STOP
1170 !
1180 !
1190 ! ** PLOTS "INTENSE" RAINS **
1200 !
1210 FGPOINT Xorig%+((Rad%+1%)*Plotfactx%*V),Yorig%+((Rad%+1%)*Plotfacty%*U),Color
     %(K%)
1220 FGPOINT Xorig%+((Rad%-1%)*Plotfactx%*V),Yorig%+((Rad%-1%)*Plotfacty%*U),Color
     %(K%)
1230 RETURN ! TO ***
1240 !
1250 ! ** GAGE PLOTTING SUBROUTINE **
1260 READ Numgage%
1270 FOR I%=1% TO Numgage%
1280   READ Px%,Py%
1290   FGPOINT Px%+2%,Py%,3%
1300   FGCIRCLE Px%,Py%
1310 NEXT I%
1320 !
1330 RETURN ! TO 650/1820
1340 DATA 18,68,103,182,87,43,40,44,165,104,50,34,142,219,19,220,120,142,160
1350 DATA 181,173,33,180,89,223,154,202,227,191,226,178,172,134,213,67,206,220
1360 ! ** MAP PLOTTING **
1370 RESTORE 1630
1380 READ Px%,Py%
1390 FGPOINT Px%,Py%,3%
1400 FOR I%=1% TO 11%
1410   READ X%,Y%
1420   FGLINE X%,Y%
```

```
1430 NEXT I%
1440 READ Px%,Py%
1450 FGPOINT Px%,Py%,3%
1460 FOR I%=1% TO 12%
1470   READ X%,Y%
1480   FGLINE X%,Y%
1490 NEXT I%
1500 READ Px%,Py%
1510 FGPOINT Px%,Py%,3%
1520 FOR I%=1% TO 8%
1530   READ X%,Y%
1540   FGLINE X%,Y%
1550 NEXT I%
1560 FOR I%=1% TO 2%
1570   READ Px%,Py%,X%,Y%
1580   FGPOINT Px%,Py%,3%
1590   FGLINE X%,Y%
1600 NEXT I%
1610 RETURN ! TO 670
1620 !
1630 DATA 129,177,117,163,133,150,138,150,154,120,185,92,183,86
1640 DATA 204,67,209,47,219,38,222,19,234,14
1650 DATA 239,169,233,169,234,175,227,183,222,181,222,186,162,187
1660 DATA 129,177,11,178,31,61,45,26,37,19,37,0
1670 DATA 66,178,75,190,87,194,87,201,99,208,111,209,131,221,134,220,147,239
1680 DATA 11,178,0,178,228,0,239,21
1690 !
1700 ! ** NULL DATA SUBROUTINE **
1710 FOR J%=1% TO 64% STEP Delrad% : Color%(J%)=0% : NEXT J%
1720 RETURN ! to 1850
1730 FOR J%=65% TO 128% STEP Delrad% : Color%(J%)=0% : NEXT J%
1740 RETURN ! TO 860
1750 !
1760 ! ** QUADRANT PLOTTING SUBROUTINE **
1770 FGPOINT 0%,0%,0%
1780 FGFILL 239%,239%
1790 FGCTL 33% ! SETS COLORS (BLACK,BLUE,MAGENTA,CYAN)
1800 ; CHR$(12%)
1810 ON VAL(Ans$) GOSUB 1950,2020,2090,2160
1820 Delrad%=1% : Plotfactx%=2% : Plotfacty%=2%
1830 GOSUB 1250 ! PLOT GAGES
1840 GOSUB 2250 ! INPUT DATA FROM FILE
1850 !
1860 WHILE Azimuth%)=L1% AND Azimuth%(=U1%
1870   GOSUB 840 ! PLOT DATA
1880   GOSUB 2250 ! INPUT DATA FROM FILE
1890   Ctr%=Ctr%+1% ! PLOTTED RADIAL COUNTER
1900   IF Ctr%)90% STOP
1910 WEND
1920 IF Ctr%(90 1840
```

```
1930 STOP
1940 !
1950 ! QUADRANT 1
1960 L1%=0% : U1%=90%
1970 Xorig%=0% : Yorig%=0%
1980 RESTORE 1990
1990 DATA 9,55,61,46,82,203,2,100,106,106,30,214,117,71,165,216,143,174,202
2000 RETURN ! TO 1820
2010 !
2020 ! QUADRANT 2
2030 L1%=270% : U1%=359%
2040 Xorig%=239% : Yorig%=0%
2050 RESTORE 2060
2060 DATA 1,119,119
2070 RETURN ! TO 1820
2080 !
2090 ! QUADRANT 3
2100 L1%=180% : U1%=270%
2110 Xorig%=239% : Yorig%=239%
2120 RESTORE 2130
2130 DATA 1,119,119
2140 RETURN ! TO 1820
2150 !
2160 ! QUADRANT 4
2170 L1%=90% : U1%=180%
2180 Xorig%=0% : Yorig%=239%
2190 RESTORE 2200
2200 DATA 1,119,119
2210 RETURN ! TO 1820
2220 !
2230 !
2240 ! ** DATA INPUT SUBROUTINE **
2250 INPUT #2,Hour$,Min$,Elev,Azimuth%
2260 INPUT #2,Color1$
2270 INPUT #2,Color2$
2280 ; CUR(0%,0%); : ; USING "###" Azimuth%; : ; " DEG"
2290 ; USING "&" CUR(1%,0%) Hour$ ":"; : ; USING "&" Min$ " Z"
2300 RETURN ! TO 800/1830/1870
2310 RESUME 1060 ! ERROR HANDLER,INPUT NEXT POINT
2320 ! ************ OKIGRAPH *************
2330 ! *                                 *
2340 ! *     GRAPHICS SCREEN DUMP TO     *
2350 ! *         OKIDATA PRINTER         *
2360 ! *                                 *
2370 ! *  KEVIN ROBBINS - JULY 13,1983   *
2380 ! *                                 *
2390 ! ***********************************
2400 !
2410 FGCTL 0 ! GRAPHICS HIDDEN
2420 OPEN "PR:" AS FILE 1
```

```
2430 ; #1 CHR$(24) ! CANCEL PRINT BUFFER
2440 FGCTL 0 : ; CHR$(12)
2450 ; CUR(2,0) RED DBLE "SELECT PRINT SIZE"
2460 ; CUR(6,0) "1.  6 LPI, 10 CPI (LARGE SCALE)"
2470 ; "2.  6 LPI, 16.5 CPI"
2480 ; "3.  8 LPI, 10 CPI"
2490 ; "4.  8 LPI, 16.5 CPI (SMALL SCALE)"
2500 !
2510 ; CUR(11,0) "SELECTION ? "; : GET Ans$ ! PRINT SIZE SELECTION
2520 ON VAL(Ans$) GOSUB 2870,2920,2970,3020
2530 FGCTL 33 ! COLOR CONTROL GROUP FOR SCREEN
2540 DIM Swath%(3,240),Oki%(120)
2550 FOR X%=0% TO 239% STEP 3%
2560   ; CHR$(12) : ; "READING"
2570   K%=0%
2580   FOR Y%=0% TO 239% STEP 2%
2590     IF FGPOINT(X%,Y%))0 Swath%(K%,Y%)=1% ELSE Swath%(K%,Y%)=0%
2600     IF FGPOINT(X%,Y%+1%))0 Swath%(K%,Y%+1%)=2% ELSE Swath%(K%,Y%+1%)=0%
2610   NEXT Y%
2620   K%=1%
2630   FOR Y%=0% TO 239% STEP 2%
2640     IF FGPOINT(X%+K%,Y%))0 Swath%(K%,Y%)=4% ELSE Swath%(K%,Y%)=0%
2650     IF FGPOINT(X%+K%,Y%+1%))0 Swath%(K%,Y%+1%)=8% ELSE Swath%(K%,Y%+1%)=0%
2660   NEXT Y%
2670   K%=2%
2680   FOR Y%=0% TO 239% STEP 2%
2690     IF FGPOINT(X%+K%,Y%))0 Swath%(K%,Y%)=16% ELSE Swath%(K%,Y%)=0%
2700     IF FGPOINT(X%+K%,Y%+1%))0 Swath%(K%,Y%+1%)=32% ELSE Swath%(K%,Y%+1%)=0%
2710   NEXT Y%
2720   ; #1 CHR$(27,71); ! PRINTER GRAPHICS MODE
2730   FOR Y%=0% TO 239% STEP 2%
2740     Y2%=Y%/2%
2750     Oki%(Y2%)=128
2760     FOR K%=0% TO 2%
2770       Oki%(Y2%)=Oki%(Y2%) OR Swath%(K%,Y%)
2780       Oki%(Y2%)=Oki%(Y2%) OR Swath%(K%,Y%+1%)
2790     NEXT K%
2800   NEXT Y%
2810   ; CHR$(12) : ; "WRITING"
2820   FOR I%=0% TO 119%
2830     ; #1 CHR$(Oki%(I%)); ! GRAPHICS DATA
2840   NEXT I%
2850   ; #1 CHR$(27,84) ! PRINTER TEXT MODE FOR LINE FEED
2860 NEXT X%
2870 !
2880 ! 6 LPI, 10 CPI
2890 !
2900 ; #1 CHR$(27,54); : ; #1 CHR$(30)
2910 RETURN
2920 !
```

```
2930 ! 6 LPI, 16.5 CPI
2940 !
2950 ; #1 CHR$(27,54); : ; #1 CHR$(29)
2960 RETURN
2970 !
2980 ! 8 LPI, 10 CPI
2990 !
3000 ; #1 CHR$(27,56); : ; #1 CHR$(30)
3010 RETURN
3020 !
3030 ! 8 LPI, 16.5 CPI
3040 !
3050 ; #1 CHR$(27,56); : ; #1 CHR$(29)
3060 RETURN
```

```
10 ! ********** RADARPRINT *************
20 ! *                                 *
30 ! *  RADAR DVIP LEVELS ARE OUTPUT   *
40 ! * FOR SPECIFIED RADAR INPUT FILES *
50 ! *                                 *
60 ! *   PROGRAMMER - KEVIN ROBBINS    *
70 ! *          JUNE, 1983             *
80 ! *                                 *
90 ! ***********************************
100 !
110 ! *
120 EXTEND ! ALLOWS 32 CHARACTER VARIABLE NAMES
130 ; CHR$(12%) ! CLEAR CHARACTER SCREEN
140 ! *
150 ! * CHANGE DISKS TO INPUT DATA *
160 ; "INSERT PROPER DATA DISK"
170 ; : ; "BE SURE TO CHANGE DISK VOLUME NAME" : ;
180 PAUSE ! ENTER OPERATING SYSTEM
190 ; CHR$(12%)
200 ; CUR(10%,0%) "DATE & TIME OF OBS ddhhmm"; : ; CUR(10%,19%);
210 INPUT ""Date$
220 Date$="RAIN:AT"+Date$+"/A"
230 OPEN Date$ AS FILE 1
240 ! *
250 ; CHR$(12%)
260 ; CUR(10%,0%) "POSITION PAPER TO TOP OF FORM"
270 ; "PRESS RETURN TO CONTINUE"
280 GET Ans$
290 ; CHR$(12%) : ; CUR(10%,0%) "PRINTING"
300 ! *
310 ! * OPEN PRINTER FILE *
320 OPEN "PR:" AS FILE 2
330 ! *
340 ! * PRINT PARAMETERS - 16.5 CPI, 8LPI *
350 ; #2 CHR$(29) : ; #2 CHR$(27,56)
360 ! *
370 ! * PRINT HEADINGS *
380 ; #2 "DEG",,,,"DVIP LEVELS" : ; #2
390 ; #2 "    20   30   40   50   60   70   80   90   100  110  120  130  140  150
        160  170  180  190  200  210  220  230  240  250  260km"
400 ; #2 "     |    |    |    |    |    |    |    |    |    |    |    |    |    |
         |    |    |    |    |    |    |    |    |    |    |"
410 ! *
420 ! * PRINT 360 RADIALS OF RADAR, 42 RADIALS/PAGE *
430 ! *
440 FOR I%=1% TO 42%
450   Ctr%=Ctr%+1%
460   IF Ctr%>360% THEN ; #2 : ; #2 "OUTPUT FILE = " Date$ : CLOSE : STOP
470   INPUT #1,Hour$,Min$,Elev,Azimuth%
480   INPUT #1,Color1$ ! DVIP DATA
```

```
490   INPUT #1,Color2$ ! DVIP DATA
500   ; #2 USING "###" Azimuth%; : ; #2 "  " Color1$; : ; #2 LEFT$(Color2$,60)
510 NEXT I%
520 ! *
530 ; #2 : ; #2 "OUTPUT FILE = " Date$
540 ; CHR$(12%)
550 ; CUR(10%,0%) "POSITION PAPER TO TOP OF FORM"
560 ; "PRESS RETURN TO CONTINUE"
570 GET Ans$
580 ; CHR$(12%) : ; CUR(10%,0%) "PRINTING"
590 GOTO 380
```

```
10 ! ************ PLOTGAGE ************
15 ! *                                *
20 ! *   DISTANCE BETWEEN TWO POINTS   *
25 ! * ON THE EARTH'S SURFACE FOR USE  *
30 ! *   IN DETERMINING THE DISTANCE   *
35 ! * BETWEEN A RADAR AND RAINGAGE    *
40 ! *                                *
45 ! *   KEVIN ROBBINS  MAY 19, 1983   *
50 ! *                                *
55 ! *                                *
60 ! * SOURCE OF EQUATIONS:           *
65 ! * "SPHERICAL TRIGONOMETRY WITH    *
70 ! *  NAVAL & MILITARY APPLICATIONS" *
75 ! *  -LYMAN M. KELLS               *
80 ! *  -WILLIS F. KERN               *
85 ! *  -JAMES R. BLAND               *
90 ! *  McGRAW-HILL BOOK COMPANY  1942 *
95 ! *                                *
100 ! **********************************
110 ! *
120 DOUBLE ! 16 DIGIT ACCURACY FOR CALCULATIONS
130 EXTEND ! ALLOWS 32 CHARACTER VARIABLE NAMES
140 ! *
150 FGPOINT 0%,0%,0% : FGFILL 239%,239% ! CLEAR HI-RES SCREEN
160 ! *
170 ! * DIMENSION CHARACTER STRINGS AND ARRAYS *
180 DIM Lon2$(40%)=10%,Lat2$(40%)=10%,Gage$(40%)=20%
190 DIM Bearing(40%),Distance(40%)
200 DIM Temp$=10%,Radar$=10%,Ans$=1%
210 DIM Gagepx(40%),Gagepy(40%),Gagecx(40%),Gagecy(40%),Bearing(40%),Distance(40%)

220 ! *
230 N%=0% ! INITIALIZE GAGE COUNTER
240 ; CHR$(12%) ! CLEAR LO-RES SCREEN
250 ! *
260 INPUT "NAME OF RADAR (ORIGIN)? "Radar$
270 N%=N%+1% ! UPDATE GAGE COUNTER
280 IF Radar$="" Radar$=Temp$ : Repeat%=1% ELSE Repeat%=0%
290 ! *
300 ; "NAME OF RAINGAGE" N%;
310 INPUT Gage$(N%)
320 ;
330 IF Repeat%=1% GOTO 390
340 ! *
350 INPUT "LATITUDE OF RADAR (HH.MM.SS)? "Lat1$
360 INPUT "LONGITUDE OF RADAR (HH.MM.SS)? "Lon1$
370 ;
380 ! *
390 INPUT "LATITUDE OF GAGE (HH.MM)? "Lat2$(N%)
400 INPUT "LONGITUDE OF GAGE (HH.MM)? "Lon2$(N%)
```

```
410 ! *
420 ! * LATITUDE/LONGITUDE STRING MANIPULATION *
430 Hlat1$=LEFT$(Lat1$,2%) ! HOURS
440 Mlat1$=MID$(Lat1$,4%,2%) ! MINUTES
450 Hlon1$=LEFT$(Lon1$,2%) ! HOURS
460 Mlon1$=MID$(Lon1$,4%,2%) ! MINUTES
470 Slat1$=RIGHT$(Lat1$,7%) ! SECONDS
480 Slon1$=RIGHT$(Lon1$,7%) ! SECONDS
490 Hlat2$=LEFT$(Lat2$(N%),2%) ! HOURS
500 Mlat2$=MID$(Lat2$(N%),4%,2%) ! MINUTES
510 Hlon2$=LEFT$(Lon2$(N%),2%) ! HOURS
520 Mlon2$=MID$(Lon2$(N%),4%,2%) ! MINUTES
530 Slat2$=RIGHT$(Lat2$(N%),7%) ! SECONDS
540 Slon2$=RIGHT$(Lon2$(N%),7%) ! SECONDS
550 ! *
560 Rad=180/PI ! RADIAN CONVERSION FACTOR
570 ! *
580 ! * CONVERT COORDINATES TO DECIMAL RADIANS *
590 Lat1=(VAL(Hlat1$)+VAL(Mlat1$)/60+VAL(Slat1$)/3600)/Rad
600 Lon1=(VAL(Hlon1$)+VAL(Mlon1$)/60+VAL(Slon1$)/3600)/Rad
610 Lat2=(VAL(Hlat2$)+VAL(Mlat2$)/60+VAL(Slat2$)/3600)/Rad
620 Lon2=(VAL(Hlon2$)+VAL(Mlon2$)/60+VAL(Slon2$)/3600)/Rad
630 ! *
640 ! * CALCULATION OF BEARING & DISTANCE *
650 Tanphi=COS(Lon1-Lon2)*TAN(PI/2-Lat2)
660 Phi=ATN(Tanphi)
670 Phiprime=PI/2-Lat1-Phi
680 Cotcourse=(1/TAN(Lon1-Lon2))*SIN(Phiprime)*(1/(SIN(Phi)))
690 Course=-ATN(Cotcourse)+PI/2
700 Cosdist=(COS(Phiprime)*(1/COS(Phi))*COS(PI/2-Lat2))
710 Dist=-ATN(Cosdist/SQR(-Cosdist*Cosdist+1))+PI/2
720 ! *
730 IF Lon2>Lon1 Course=Course+PI ! ALLOWS FULL 360 DEG RESULTS
740 ! *
750 ! * OUTPUT RESULTS TO SCREEN *
760 ; CHR$(12%)
770 ; YEL Radar$; : ; " COORDINATES (DEGREES)"
780 ; "LAT. = " Lat1$, : ; "LONG. = " Lon1$
790 ; : ; YEL Gage$; : ; " COORDINATES (DEGREES)"
800 ; "LAT. = " Lat2$(N%), : ; "LONG. = " Lon2$(N%)
810 ; CUR(8%,0%)
820 ; YEL "BEARING FROM " Radar$ " TO " Gage$(N%)
830 ; USING "###.#" Course; : ; " RADIANS"
840 Bearing(N%)=Course*Rad
850 ; USING "###.#" Bearing(N%); : ; " DEGREES" : ;
860 ; CUR(13%,0%) YEL "DISTANCE FROM " Radar$ " TO " Gage$(N%)
870 ; USING "####.#" Dist*60*Rad; : ; " NAUTICAL MILES"
880 ; USING "####.#" Dist*60*Rad*1.1507794; : ; " STATUTE MILES"
890 Distance(N%)=Dist*60*Rad*1.852
900 ; USING "####.#" Distance(N%); : ; " KILOMETERS"
```

```
910 ! *
920 Temp$=Radar$
930 GOSUB 1260 ! PLOTTING & STORAGE SUBROUTINE
940 ! *
950 ; : ;
960 ; "OUTPUT GAGES TO PRINTER & QUIT (Q) "
970 ; "NEW STATION (S)"
980 ; "OUTPUT SCREEN TO PRINTER (P) "
990 ! *
1000 GET Ans$
1010 IF Ans$="Q" OR Ans$="q" GOTO 1440 ELSE IF Ans$="P" OR Ans$="p" GOSUB 1050
1020 GOTO 240 ! INPUT ANOTHER STATION
1030 ! *
1040 ! * SCREEN DUMP SUBROUTINE *
1050 OPEN "PR:" AS FILE 1 ! OPEN PRINTER FILE
1060 ; #1 "***************************************"
1070 ; #1 Radar$ " COORDINATES (DEGREES)"
1080 ; #1 "LAT. = " Lat1$,"LONG. = " Lon1$
1090 ; #1 : ; #1 Gage$(N%); : ; #1 " COORDINATES (DEGREES)"
1100 ; #1 "LAT. = " Lat2$(N%), : ; #1 "LONG. = " Lon2$(N%)
1110 ; #1 : ; #1
1120 ; #1 "BEARING FROM " Radar$ " TO " Gage$(N%)
1130 ; #1 USING "#.###" Course; : ; #1 " RADIANS"
1140 ; #1 USING "###.#" Course*Rad; : ; #1 " DEGREES" : ; #1
1150 ; #1 : ; #1 "DISTANCE FROM " Radar$ " TO " Gage$(N%)
1160 ; #1 USING "####.#" Dist*60*Rad; : ; #1 " NAUTICAL MILES"
1170 ; #1 USING "####.#" Dist*60*Rad*1.1507794; : ; #1 " STATUTE MILES"
1180 ; #1 USING "####.#" Dist*60*Rad*1.852; : ; #1 " KILOMETERS"
1190 ; #1 "***************************************"
1200 ! *
1210 Temp$=Radar$
1220 IF MOD(N%,3%)=0% ; #1 CHR$(12%) ! FORM FEED TO PRINTER
1230 RETURN
1240 ! *
1250 ! * PLOTTING AND STORAGE SUBROUTINE *
1260 FGCTL 33% ! SETS COLOR GROUP (BK-B-M-C + CHARACTERS)
1270 FGPOINT 119%,119%,3% ! RADAR LOCATION (CENTER OF SCREEN)
1280 ! *
1290 Course=Course+3*PI/2 ! ANGLE OF GAGE FROM RADAR (RADIANS)
1300 U=SIN(-Course)
1310 V=COS(-Course)
1320 Dist=Dist*60*Rad*1.852 ! DISTANCE FROM RADAR TO GAGE (KM)
1330 ! *
1340 FGPOINT 119%+(Dist/2*V)+2%,119%+(Dist/2*U),3%
1350 FGCIRCLE 119%+(Dist/2*V),119%+(Dist/2*U)
1360 ! *
1370 Gagepx(N%)=119%+(Dist/2*V)+2%
1380 Gagepy(N%)=119%+(Dist/2*U)
1390 Gagecx(N%)=119%+(Dist/2*V)
1400 Gagecy(N%)=119%+(Dist/2*U)
```

```
1410 RETURN
1420 ! *
1430 ! * SEND RESULTS TO PRINTER *
1440 ; CHR$(12%)
1450 OPEN "PR:" AS FILE 1 ! OPEN PRINTER FILE
1460 ; #1 CHR$(12%) ! ADVANCE PAPER TO TOP OF FORM
1470 ; #1 "RADAR = " Radar$ ! RADAR NAME
1480 ; #1 "LATITUDE = " Lat1$, : ; #1 "LONGITUDE = " Lon1$
1490 ; #1 : ; #1 : ; #1
1500 FOR I%=1% TO N% ! PROCESS ALL GAGES
1510   ; #1 Gage$(I%)
1520   ; #1 "LATITUDE = " Lat2$(I%), : ; #1 "LONGITUDE = " Lon2$(I%)
1530   ; #1 "BEARING(DEG) FROM " Radar$; : ; #1 " = "; : ; #1 USING "###.##" Beari
     ng(I%)
1540   ; #1 "DISTANCE(KM) FROM " Radar$; : ; #1 " = "; : ; #1 USING "###.##" Dista
     nce(I%)
1550   ; #1 'MONITOR "X" COORDINATE '; : ; #1 USING "###.#" Gagecx(I%)
1560   ; #1 'MONITOR "Y" COORDINATE '; : ; #1 USING "###.#" Gagecy(I%)
1570   ; #1
1580   IF MOD(I%,7%)=0% ; #1 CHR$(12%) : ; #1 "RADAR = " Radar$ : ; #1 "LATITUDE =
     " Lat1$, : ; #1 "LONGITUDE = " Lon1$ : ; #1 : ; #1
1590 NEXT I% ! NEXT GAGE
1600 STOP
```

```
10 ! ************** RADAR **************
20 ! *                               *
30 ! *      RAINFALL ACCUMULATION     *
40 ! *                               *
50 ! *     KEVIN ROBBINS-JUNE,1983    *
60 ! *                               *
70 ! *   RAINFALL AMOUNTS FROM RADAR.  *
80 ! *   AMOUNTS ARE GIVEN IN INCHES.  *
90 ! *                               *
100 ! **********************************
110 !
120 ! *
130 EXTEND ! ALLOWS 32 CHAR VAR NAMES
140 ; CHR$(12%) ! CLEARS LO-RES SCREEN
150 ! *
160 ! * DIMENSION ARRAYS  AND CHARACTER STRINGS *
170 DIM Color1$=64%,Color2$=64% ! DATA STRINGS
180 DIM Color%(128%) ! RAIN RATE ARRAY
190 DIM Ans$=2% ! ALL-PURPOSE KEYBOARD INPUT VARIABLE
200 ! *
210 OPEN "GAGES" AS FILE 1 ! FILE OF AVAILABLE GAGE SITES
220 INPUT #1,N% ! NUMBER OF GAGES FROM FILE
230 ; "SELECT A GAGE FOR PROCESSING "
240 ! *
250 ! * DIMENSION GAGE ARRAYS *
260 DIM Dist%(N%),Bearing%(N%) ! GAGE COORDINATES
270 DIM Name$(N%)=16% ! GAGE NAMES
280 ! *
290 ! * INPUT GAGES FROM FILE *
300 FOR I%=1% TO N%/2%
310   FOR J%=1% TO 2%
320     K%=I%+J%-1%
330     INPUT #1,Name$(K%),Dist%(K%),Bearing%(K%) ! DATA ENTERED IN "NORMALIZED" C
    OORDINATES
340     ; CUR(I%+2%,(J%-1%)*20%); : ; USING "##." K%; : ; " " Name$(K%)
350   NEXT J%
360 NEXT I%
370 ! *
380 ;
390 CLOSE 1 ! CLOSE GAGES FILE
400 ! *
410 INPUT Ans$ ! GAGE NUMBER
420 ; CHR$(12%)
430 K%=VAL(Ans$)
440 Rad%=Dist%(K%)/2%-9% ! DATA ARRAY INDEX
450 ! *
460 INPUT "RADAR NAME ? "Radar$
470 IF Radar$="ATHENS" File$="RAIN:AT" ELSE File$="RAIN:CH"
480 ; CHR$(12%) : ; YEL Radar$; : ; WHT "RADAR"
490 ; "HOW MANY RADAR EVENTS  "
```

```
500 INPUT Num%
510 ! *
520 ! * DIMENSION PARAMETER ARRAYS FOR NUMBER OF EVENTS *
530 DIM Hour%(Num%),Interval%(Num%),Totrain(Num%),Min%(Num%),Date$(Num%)=6% ! RAIN
      ACCUMULATION VARIABLES
540 ! *
550 ; CUR(2%,0%); : ; CHR$(17%) ! CLEAR SCREEN BELOW CURSOR
560 ; CUR(3%,0%) "ACCUMULATING RAINFALL FOR"; : ; YEL Name$(K%)
570 ! *
580 ! * DATA INPUT AND PROCESSING LOOP *
590 FOR I%=1% TO Num%
600   ; CUR(4%,0%) "DATE & TIME OF OBS  ddhhmm"; : ; CUR(4%,19%);
610   INPUT " "Date$(I%)
620   ; CUR(12%,0%)
630   CLOSE 1
640   ; "CHANGE DISKS? "
650   PAUSE ! TO CHANGE DISKS, THEN CONTINUE
660   ; CUR(5%,0%); : ; CHR$(17%)
670   OPEN File$+Date$(I%) AS FILE 1 ! RADAR DATA FILE
680   GOSUB 1150 ! INPUT DATA
690   ! *
700   WHILE Azimuth%(Bearing%(K%)-2% OR Azimuth%)Bearing%(K%)+2% ! SKIP UNUSED DAT
      A
710     GOSUB 1150 ! INPUT DATA
720     ; CUR(6%,0%); : ; USING "###" Azimuth%; : ; " DEG  (NULL DATA)"
730   WEND
740   ! *
750   IF I%=1% Hour%(0%)=VAL(Hour$) : Min%(0%)=VAL(Min$)
760   ! *
770   FOR J%=1% TO 3% ! 3 POINTS ALONG SELECTED RADIAL
780     GOSUB 1150 ! INPUT DATA
790     ; CUR(6%,0%) YEL; : ; USING "###" Azimuth%; : ; " DEG  (USABLE RADIAN)" :
      ; CHR$(17)
800     ! *
810     ! * LOAD ARRAY WITH RADAR DATA *
820     FOR I1%=1% TO 64%
830       Color%(I1%)=VAL(MID$(Color1$,I1%,1%))
840       Color%(I1%+64%)=VAL(MID$(Color2$,I1%,1%))
850     NEXT I1%
860     ! *
870     ON J% GOSUB 1210,1370,1210 ! "WEIGHTING" DATA POINTS
880   NEXT J%
890   ! *
900   ! * TOTAL RAINFALL FROM 9 DATA POINTS *
910   Ltotrain(I%)=(Lrain(I%)*Interval%(I%)/60)/4.416 ! AVERAGING COEFFICIENT
920   Mtotrain(I%)=(Mrain(I%)*Interval%(I%)/60)/4.415 ! AVERAGING COEFFICIENT
930   Htotrain(I%)=(Hrain(I%)*Interval%(I%)/60)/4.416 ! AVERAGING COEFFICIENT
940   ! *
950   ! * OUTPUT RAINFALL AMOUNTS TO SCREEN *
960   ; CUR(6%,0%) Interval%(I%); : ; "MINUTE RAINFALL = "; : ; USING "#.###" Ltot
      rain(I%); : ; "IN  (LOW)"
```

```
970   ; CUR(7%,0%) Interval%(I%); : ; "MINUTE RAINFALL = "; : ; USING "#.###" Mtot
      rain(I%); : ; "IN  (MED)"
980   ; CUR(8%,0%) Interval%(I%); : ; "MINUTE RAINFALL = "; : ; USING "#.###" Htot
      rain(I%); : ; "IN  (HIGH)"
990   ! *
1000   GOSUB 2130 ! SOIL MOISTURE SUBROUTINE
1010 NEXT I%
1020 ! *
1030 ! * CALCULATE TOTALS FOR EVENT *
1040 FOR I%=1% TO Num%
1050   Ltotal=Ltotal+Ltotrain(I%)
1060   Mtotal=Mtotal+Mtotrain(I%)
1070   Htotal=Htotal+Htotrain(I%)
1080 NEXT I%
1090 ! *
1100 CLOSE 1 ! CLOSE DATA FILE
1110 ! *
1120 GOTO 1930 ! PRINTER SUBROUTINE
1130 ! *
1140 ! ** DATA INPUT SUBROUTINE **
1150 INPUT #1,Hour$,Min$,Elevation,Azimuth%
1160 INPUT #1,Color1$
1170 INPUT #1,Color2$
1180 RETURN
1190 ! *
1200 ! * DEPTH CALCULATION SUBROUTINE *
1210 Hour%(I%)=VAL(Hour$) : Min%(I%)=VAL(Min$)
1220 IF I%=1% Interval%(I%)=15% : GOTO 1250
1230 IF Hour%(I%-1%)=Hour%(I%) THEN Interval%(I%)=Min%(I%)-Min%(I%-1%) ELSE Interv
      al%(I%)=Min%(I%)+60%-Min%(I%-1%)
1240 ! *
1250 FOR Ptr%=Rad%-1% TO Rad%+1% ! PTR=ARRAY POINTER
1260   R%=Ptr%-Rad%+2% ! RADIAL ARRAY POINTER
1270   ON Color%(Ptr%)+1% GOSUB 1480,1520,1580,1640,1700,1760,1820,1860 ! JUMP ON
      DVIP LEVELS
1280 NEXT Ptr%
1290 ! *
1300 ! * RAIN RATE CALCULATIONS FOR A POINT *
1310 Lrain(I%)=(Lrain(I%)+Lrate(1%)*.354+Lrate(2%)*.5+Lrate(3%)*.354)
1320 Mrain(I%)=(Mrain(I%)+Mrate(1%)*.354+Mrate(2%)*.5+Mrate(3%)*.354)
1330 Hrain(I%)=(Hrain(I%)+Hrate(1%)*.354+Hrate(2%)*.5+Hrate(3%)*.354)
1340 RETURN
1350 ! *
1360 ! ** DEPTH CALCULATION SUBROUTINE **
1370 FOR Ptr%=Rad%-1% TO Rad%+1% ! PTR=ARRAY POINTER
1380   R%=Ptr%-Rad%+2% ! RADIAL ARRAY POINTER
1390   ON Color%(Ptr%)+1% GOSUB 1480,1520,1580,1640,1700,1760,1820,1860
1400 NEXT Ptr%
1410 ! *
1420 ! * RAIN RATE CALCULATIONS FOR A POINT *
```

```
1430 Lrain(I%)=(Lrain(I%)+Lrate(1%)*.5+Lrate(2%)+Lrate(3%)*.5)
1440 Mrain(I%)=(Mrain(I%)+Mrate(1%)*.5+Mrate(2%)+Mrate(3%)*.5)
1450 Hrain(I%)=(Hrain(I%)+Hrate(1%)*.5+Hrate(2%)+Hrate(3%)*.5)
1460 RETURN
1470 ! *
1480 ! ** DVIP 0, RAINFALL RATE = 0 in/hr **
1490 Lrate(R%)=0 : Mrate(R%)=0 : Hrate(R%)=0
1500 RETURN
1510 ! *
1520 ! ** DVIP 1, RAINFALL RATE = .01-.1 in/hr **
1530 Lrate(R%)=.01
1540 Mrate(R%)=.05
1550 Hrate(R%)=.1
1560 RETURN
1570 ! *
1580 ! ** DVIP 2, RAINFALL RATE = .1-.5 in/hr **
1590 Lrate(R%)=.1
1600 Mrate(R%)=.25
1610 Hrate(R%)=.5
1620 RETURN
1630 ! *
1640 ! ** DVIP 3, RAINFALL RATE = .5-1 in/hr **
1650 Lrate(R%)=.5
1660 Mrate(R%)=.75
1670 Hrate(R%)=1
1680 RETURN
1690 ! *
1700 ! ** DVIP 4, RAINFALL RATE = 1-2 in/hr **
1710 Lrate(R%)=1
1720 Mrate(R%)=1.5
1730 Hrate(R%)=2
1740 RETURN
1750 ! *
1760 ! ** DVIP 5, RAINFALL RATE = 2-5 in/hr **
1770 Lrate(R%)=2
1780 Mrate(R%)=3.5
1790 Hrate(R%)=5
1800 RETURN
1810 ! *
1820 ! ** DVIP 6, RAINFALL RATE = >5 in/hr **
1830 Lrate(R%)=5 : Mrate(R%)=5 : Hrate(R%)=5
1840 RETURN
1850 ! *
1860 ! ** DVIP 7, ERRONEOUS DATA HANDLER **
1870 ! * SET DVIP LEVEL TO LEVEL OF PREVIOUS POINT *
1880 Lrate(R%)=Lrate(R%-1%)
1890 Mrate(R%)=Mrate(R%-1%)
1900 Hrate(R%)=Hrate(R%-1%)
1910 RETURN
1920 ! *
```

```
1930 ! ** PRINTER SUBROUTINE **
1940 OPEN "PR:" AS FILE 5
1950 ; #5 CHR$(27,66) ! SHORT LINE (8.5") PAPER
1960 ; #5 Radar$; : ; #5 " RADAR RAINFALL DATA FOR " Name$(K%) : ; #5
1970 ! *
1980 FOR I%=1% TO Num%
1990   ; #5 Date$(I%) ! TIME OF OBSERVATION
2000   ; #5 Interval%(I%); : ; #5 "MINUTE LOW RANGE RAINFALL = "; : ; #5 USING "#.
     ###" Ltotrain(I%); : ; #5 " in."
2010   ; #5 Interval%(I%); : ; #5 "MINUTE MED RANGE RAINFALL = "; : ; #5 USING "#.
     ###" Mtotrain(I%); : ; #5 " in."
2020   ; #5 Interval%(I%); : ; #5 "MINUTE HIGH RANGE RAINFALL = "; : ; #5 USING "#
     .###" Htotrain(I%); : ; #5 " in."
2030   ; #5
2040 NEXT I%
2050 ! *
2060 ; #5 : ; #5 : ; #5 "RAINFALL TOTALS FOR PERIOD"
2070 ; #5 : ; #5 "LOW TOTAL = "; : ; #5 USING "##.###" Ltotal; : ; #5 " in."
2080 ; #5 "MED TOTAL = "; : ; #5 USING "##.###" Mtotal; : ; #5 " in."
2090 ; #5 "HIGH TOTAL = "; : ; #5 USING "##.###" Htotal; : ; #5 " in."
2100 CLOSE 5 ! CLOSE PRINTER FILE
2110 ! *
2120 STOP
2130 ! **********************************
2140 ! *      RAINFALL CONVOLUTION      *
2150 ! *   FOR STREAMFLOW GENERATION    *
2160 ! *                                *
2170 ! *  ADAPTED BY KEVIN ROBBINS      *
2180 ! *  FROM FORTRAN VERSION WRITTEN  *
2190 ! *  BY S. NNAJI, MAY, 1983.       *
2200 ! *                                *
2210 ! **********************************
2220 !
2230 CLOSE 1
2240 IF I%>1 GOTO 2660 ! SKIP INITIALIZATION SECTION
2250 EXTEND ! ALLOWS 32 CHAR. VAR.NAMES
2260 DIM Cumrain(30),Rain(30),Q(30),Sm(30),T(30),Temp(30),Time(30)
2270 DIM Et(30),Tfall(30),Drain(30),Flow(30),H(15)
2280 !
2290 ; CHR$(12%)
2300 ; "ENTER THE FOLLOWING VARIABLES:" : ;
2310 INPUT "INTERCEPTION PARAMETER = "Intercept
2320 INPUT "ANTECEDENT SOIL MOISTURE = "Asm
2330 INPUT "INTERCEPTION CAPACITY = "Maxintercept
2340 INPUT "SOIL MOISTURE SATURATION = "Maxsm
2350 INPUT "SOIL DEPLETION EXPONENT = "G
2360 INPUT "# OF TEMPERATURE OBS = "Numtemp
2370 FOR I=1 TO 10
2380   READ H(I)
2390 NEXT I
```

```
2400 OPEN "PR:" AS FILE 5 : ; #5 CHR$(29)
2410 ; #5 "INITIAL CONDITIONS" : ; #5
2420 ; #5 "INTERCEPTION  ANTECEDENT    MAXIMUM       MAXIMUM       DEPLETION"
2430 ; #5 "PARAMETER     SOIL MOIST.  INTERCEPTION  SOIL MOIST.   EXPONENT"
2440 ; #5 "__________________________________________________________________" :
     ; #5
2450 ; #5 Intercept,Asm,Maxintercept,Maxsm,G
2460 ; #5 : ; #5
2470 FOR I=1 TO Numtemp
2480   ; "TEMPERATURE" I "= "; : INPUT ""Temp(I)
2490   ; #5 "TEMPERATURE = " Temp(I)
2500 NEXT I
2510 ; #5 : ; #5 : ; #5 : ; #5
2520 Ff=1 ! FORM FEED CONTROL
2530 FOR I=1 TO 30
2540   Cumrain(I)=0
2550   Drain(I)=0
2560   Et(I)=.002
2570   Rain(I)=0
2580   Q(I)=0
2590   Sm(I)=0
2600   T(I)=0
2610 NEXT I
2620 FOR I=1 TO Numtemp
2630   Asm=Asm*EXP(-G*Temp(I))
2640 NEXT I
2650 Sm(0)=Asm
2660 ; CHR$(12%)
2670 J=I%
2680 Delt=Interval%(J)
2690 Rain(J)=Mtotrain(J)
2700 T(J)=Delt/60+T(J-1)
2710 Cumrain(J)=Cumrain(J-1)+Rain(J-1)
2720 IF C(J)(Maxintercept Tfall(J)=(1-Intercept)*(Rain(J)-Et(J))
2730 IF C(J))=Maxintercept Tfall(J)=Rain(J)-Et(J)
2740 Sm(J)=Sm(J-1)+Tfall(J)-Drain(J)
2750 IF Sm(J)(=0 Sm(J)=0
2760 IF Sm(J)(=Maxsm Q(J)=0 : IF Ff=1 Ff=0 : GOSUB 3050 ELSE GOSUB 3090
2770 IF Sm(J))=Maxsm Q(J)=Sm(J)-Maxsm : Sm(J)=Maxsm
2780 !
2790 ! CONVOLUTING EFFECTIVE RUNOFF
2800 !
2810 IF Q(J)=0 RETURN
2820 ; #5 CHR$(12) : GOSUB 3050
2830 ; #5 : ; #5 : ; #5 : ; #5 : ; #5 "CONVOLUTION" : ; #5 : ; #5 " Time(Hr) "," F
     low(In/Hr) "
2840 ; #5 "_______________________"
2850 Nh=10
2860 Ny=Nh+J-1
2870 L1=1
```

```
2880 FOR I=1 TO Ny
2890   IF I(J Time(I)=T(I) : Del=T(I+1)-T(I)
2900   IF I)=J Del=.25 : Time(I)=Time(I-1)+Del
2910   Lu=1
2920   Flow(I)=0
2930   IF I)J Lu=J
2940   IF I)Nh Ll=I-Nh+1
2950   FOR L=Ll TO Lu
2960     K=I-L+1
2970     Flow(I)=Flow(I)+Del*Q(L)*H(K)
2980   NEXT L
2990   ; #5 USING "##.##" Time(I); : ; #5 TAB(15); : ; #5 USING "##.###" Flow(I)
3000 NEXT I
3010 IF I%=Num% ; #5 CHR$(12)
3020 Ff=1
3030 RETURN
3040 DATA 0,.15,.32,.5,.46,.38,.25,.19,.07,0
3050 ; #5 "DATA FOR INTERVAL(S)" : ; #5
3060 ; #5 "SAMPLING      START TIME     RAINFALL IN    SOIL          POTENTIAL"
3070 ; #5 "INTERVAL      THIS INTERVAL  THIS INTERVAL  MOISTURE      RUNOFF"
3080 ; #5 "_________________________________________________________________"
     : ; #5
3090 ; #5 USING "##.##" Delt; : ; #5 TAB(15); : ; #5 USING "##.##" T(J-1); : ; #5
     TAB(30); : ; #5 USING "##.###" Rain(J);
3100 ; #5 TAB(45); : ; #5 USING "##.###" Sm(J); : ; #5 TAB(60); : ; #5 USING "##.#
     ##" Q(J)
3110 RETURN
```