

AD-A103 000

AIR FORCE GEOPHYSICS LAB HANSCOM AFB MA

F/G 4/2

A DIGITAL PROCESSING AND DISPLAY SYSTEM FOR THE ROTATING BEAM C--ETC(U)

FEB 81 J C WEYMAN, R H LYNCH

UNCLASSIFIED

AFOL-TR-81-0027

NL

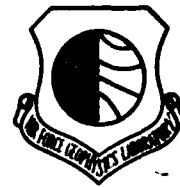
100

4.0100

END
DATE
FILMED
9-8
OTIC

AFGL-TR-81-0027
INSTRUMENTATION PAPERS, NO. 299

LEVEL



Q

**A Digital Processing and Display System for
the Rotating Beam Ceilometer (AN/GMQ-13)**

JAMES C. WEYMAN, Capt, USAF
RICHARD H. LYNCH



4 February 1981

Approved for public release; distribution unlimited.

METEOROLOGY DIVISION PROJECT 6670
AIR FORCE GEOPHYSICS LABORATORY
HANSCOM AFB, MASSACHUSETTS 01731

AIR FORCE SYSTEMS COMMAND, USAF



81 8 17 025

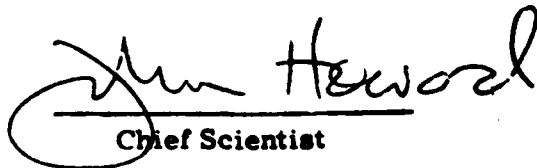
AD A103000

DTIC FILE COPY

This report has been reviewed by the ESD Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER


Chief Scientist

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFGI-TR-81-0027	2. GOVT ACCESSION NO. AD-A103000	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A DIGITAL PROCESSING AND DISPLAY SYSTEM FOR THE ROTATING BEAM CEILOMETER (AN/GMQ-13)	5. TYPE OF REPORT & PERIOD COVERED Scientific. Interim.	
7. AUTHOR(s) James C. Weyman, Capt, USAF Richard H. Lynch	6. PERFORMING ORG. REPORT NUMBER IP No. 299	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Geophysics Laboratory (LYU) Hanscom AFB Massachusetts 01731	8. CONTRACT OR GRANT NUMBER(s) 21	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory (LYU) Hanscom AFB Massachusetts 01731	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 66701007	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE 4 February 1981	
	13. NUMBER OF PAGES 68	
	15. SECURITY CLASS (of this report) Unclassified	
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Rotating Beam Ceilometer Interrupt-driven Processor/display Binomial smoothing function Cloud base height Microprocessor subsystems		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A microprocessor-based automated display system for the Rotating Beam Ceilometer (RBC) was developed to demonstrate the feasibility of modernizing and improving the present display system. The new system consists of a Burroughs self-scan panel display (model number SSD0124-0039) and an Intel 8080A-based microcomputer. The program acquires the signal returns from the RBC, converts the analog signal to a digital value, and displays the most recent scan. The system allows for three modes of operation or processing.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract (Continued)

Two modes, a one-scan mode and a five-scan mode, require no further inputs from the user once they have been selected. A manual mode which is user-interactive is also available. The basic display capabilities consist of a depiction of signal intensity vs. height, a numerical display of the height of the peak value and an illuminated cursor positioned at the peak value. Evaluation of the system at AFGL and Scott AFB, IL has confirmed that the display is more readable, provides more accurate cloud heights and permits better interpretation of the data obtained than does the current CRT display system. Further considerations which will be incorporated before operational implementation are discussed.



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Accession For	
1.1.1.1.1.1	☒
2.2.2.2.2	☐
3.3.3.3.3	☐
4.4.4.4.4	
By	
Distribution/	
Availability Codes	
5.5.5.5.5/or	
6.6.6.6.6	
A	

Preface

The successful fabrication and testing of the processor/display for the Rotating Beam Ceilometer (RBC) could not have been achieved without the technical support provided by TSgt James Boyce. In addition, the authors are grateful to Mr. William Lamkin for his assistance in the fabrication of the equipment, to Mr. Donald A. Chisholm and Lt. Col. William Wright, HQ AWS, for valuable discussions concerning the project, and to Mr. Chisholm for his many helpful comments on the paper. Lastly, the contribution of Miss Karen Sullivan in typing the manuscript is gratefully acknowledged.

DISCLAIMER

The United States Air Force does not approve, recommend, or endorse any product except for its own use, and naming an item is for the purpose of identification only and shall not be used in advertising, sales promotion, or to indicate in any manner, either implicitly or explicitly, endorsement by the U.S. Air Force.

Contents

1. INTRODUCTION	7
2. EQUIPMENT	8
2.1 AN/GMQ-13 Rotating Beam Ceilometer (RBC)	8
2.2 AFGL RBC Processor/Display	10
3. ROTATING BEAM CEILOMETER DATA	21
4. AUTOMATED DATA PROCESSING PROCEDURES	24
5. FUTURE CONSIDERATIONS	29
6. SUMMARY AND CONCLUSIONS	30
REFERENCES	32
APPENDIX A: Flow Chart of Processing Program	33
APPENDIX B: PLM-80 Software of Processing Program	39
APPENDIX C: Conversion Table from Pulse Counts to Heights	65

Illustrations

1. System Diagram	9
2. RBC Display Processor (Block Diagram)	11
3. Analog Demodulator and Signal Conditioner (Schematic Diagram)	13
4. RBC Display Computer Card (Schematic Diagram)	14

Illustrations

5. Double Dual-Port Display Memory (Simplified Schematic)	17
6. "A" Memory and Display Clock (Schematic Diagram)	19
7. "B" Memory and Toggle Circuit (Schematic Diagram)	20
8. Display Memory Load/Run Detail	22
9. RBC Measurement Geometry	23
10. Diagram of Old RBC Display	23

A Digital Processing and Display System for the Rotating Beam Ceilometer (AN/GMQ-13)

1. INTRODUCTION

The Rotating Beam Ceilometer (RBC) is one of the elder statesmen in the Air Weather Service's inventory of sensors. As with many complex sensors that have been in service an extended period of time, maintenance problems (decreased mean time between failures, availability of parts, etc.) become aggravated. This has been true of the RBC, particularly its display portion. In response to this need, the Air Force Geophysics Laboratory (AFGL) undertook the development of a new and improved display system.

An exploratory development program was initiated at AFGL to design, fabricate, test, and evaluate an automated cloud display system which relied on microprocessor technology. This system was designed to be a low cost, compact package which could provide sufficient detailed information to the operator so that he could rapidly make an accurate observation. The developed display accomplishes this goal. In addition, the microprocessor-based system analyzes the data and produces an objectively determined cloud base height from the five most recent scans. This objective cloud base height could be helpful to a busy observer, but more importantly could be transmitted through a serial channel as part of the

(Received for publication 4 February 1981)

automated observation (AUTO OBS) portion of the Automated Weather Distribution System (AWDS).¹

This report will review the specific design of the RBC, the microprocessor-based display system, the data received from the RBC, how that data are processed and displayed, and future considerations for the system.

2. EQUIPMENT

2.1 AN/GMQ-13 Rotating Beam Ceilometer (RBC)

The standard RBC system consists of a projector, a detector, and a CRT indicator as shown in Figure 1. The projector contains two back-to-back lamp assemblies on a motor driven mount which continuously rotates in the vertical plane. These lamp assemblies each contain a high current incandescent lamp inside a rotating shutter and a 24 in. parabolic mirror behind the lamp. As the mount rotates, each lamp will project a 120 Hz modulated beam which sweeps from 0 to 90 deg or from parallel to perpendicular to the ground. During the 0 to 90 deg sweep a filter on the housing attenuates visible light allowing infrared light to be transmitted. At 90 deg, the beam is reduced in power and blocked by the housing during the remaining three quadrants of mount revolution. The two-lamp arrangement doubles the atmospheric illumination portion of each revolution. This produces more frequently updated information and hardware redundancy but can introduce problems, discussed later, when computer-processing these signals. Magnets actuate a reed switch when either lamp is horizontal (at 0 deg) to develop a synchronizing pulse for the indicator.

The detector consists of a vertically viewing, parabolic mirror optical system with a photocell at its focal point. A honeycomb collimator is mounted above the mirror to reduce effects caused by sunlight and other stray light. An amplifier boosts photocell output for transmission to the indicator. A locally-fabricated reflector is usually mounted on the housing to reflect some light directly into the optical system when illuminated by a projector's lamp in the 0 deg or horizontal position. This produces a strong return on the indicator and is used to verify system operation. This feature of the system is used by the microcomputer and will be discussed later.

The operational RBC system output is displayed on a cathode ray tube (CRT) indicator. The switch closure which occurs when either lamp is at 0 deg is used to synchronize the sweep mechanism. The electron beam is swept from the bottom to the top of the CRT in synchronization with the projector's rotation from

1. Air Force Communications Service (1977) Automated Weather Distribution System (AWDS) Required Operational Capability (AFSC ROC 601-77).

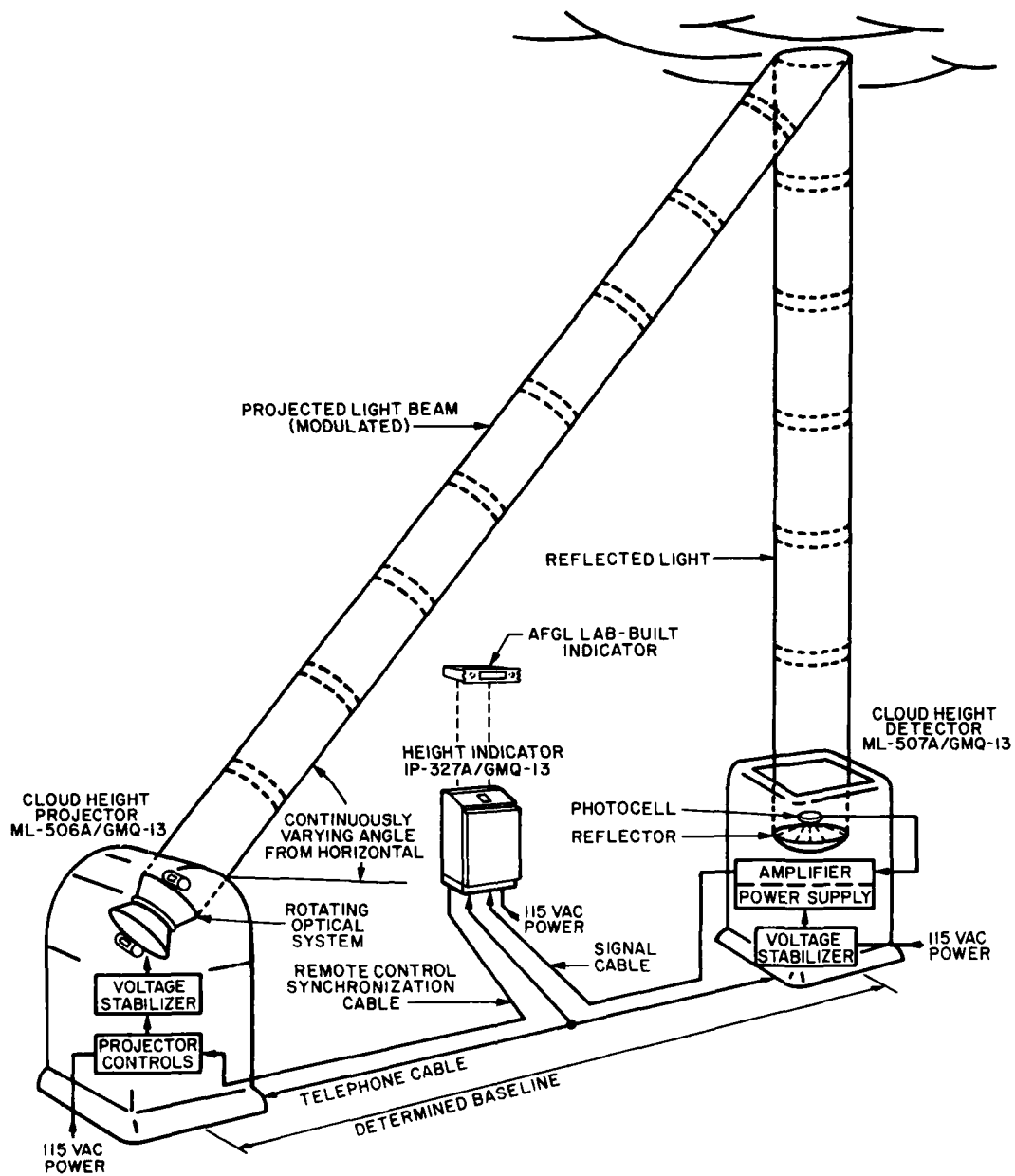


Figure 1. System Diagram

0 to 90 deg. The deflection of the beam left and right of the centerline is proportional to signal strength; that is, a strong cloud return gives a wide deflection. Operation of the indicator often requires numerous adjustments and considerable operator skill in interpreting signals resulting from multiple cloud layers, snow, fog, etc.

Actual cloud height is determined by watching the trace for the 3 sec it takes to scan from bottom to top of the CRT. If a cloud return is present, the point of maximum beam deflection is taken to be the height of the cloud's base. This is read directly from an overlay indicating height vs. projector angle or by reading an angle-only overlay and referring to a conversion table to obtain the cloud height. Numerous operator adjustments must be correctly done to maintain system accuracy. In addition, synchronization with the projector must be manually initiated.

2.2 AFGL RBC Processor/Display

The AFGL RBC processor/display unit (see Figure 2) consists of the unit's front panel and five functionally separate subsystems contained on three printed circuit cards. The five subsystems are:

1. Analog demodulator and converter,
2. Microcomputer,
3. Dot matrix display,
4. Display controller,
5. Power supply.

The analog demodulator and converter circuit is shown schematically in Figure 3. This circuit accepts the 120 Hz amplitude modulated signal from the RBC. An absolute value rectifier and integrator (M31) detects and slightly filters the envelope of the signal which is amplified by the programmable gain differential amplifier (M32). Gains of 1.00, 1.33, 1.67 and 2.00 are available under program control by means of a two pole, solid state analog switch (M30). This switch is driven by two computer output lines and alters the amplifier's feedback resistance. In addition, a dc voltage is applied to the amplifier's differential input in order to offset the residual dc baseline voltage. The offset voltage can be one of 16 values developed by a digital to analog converter (M30) and ranges from near zero to approximately 25 percent of the full scale 10 V input. The firmware description in this report (see Section 4) details the methods used to determine gain and offset values based on analysis of the signal. An 8 bit analog to digital converter (M28) completes the processing of the signal. A programmable input/output port (M27) serves as the interface between the analog and computer subsystems.

The microcomputer in this system is based on an Intel 8080A microprocessor and is shown schematically in Figure 4. Program memory capacity is 12K of 2716 type EPROM while random access memory capacity is 4K of 8108 static RAM.

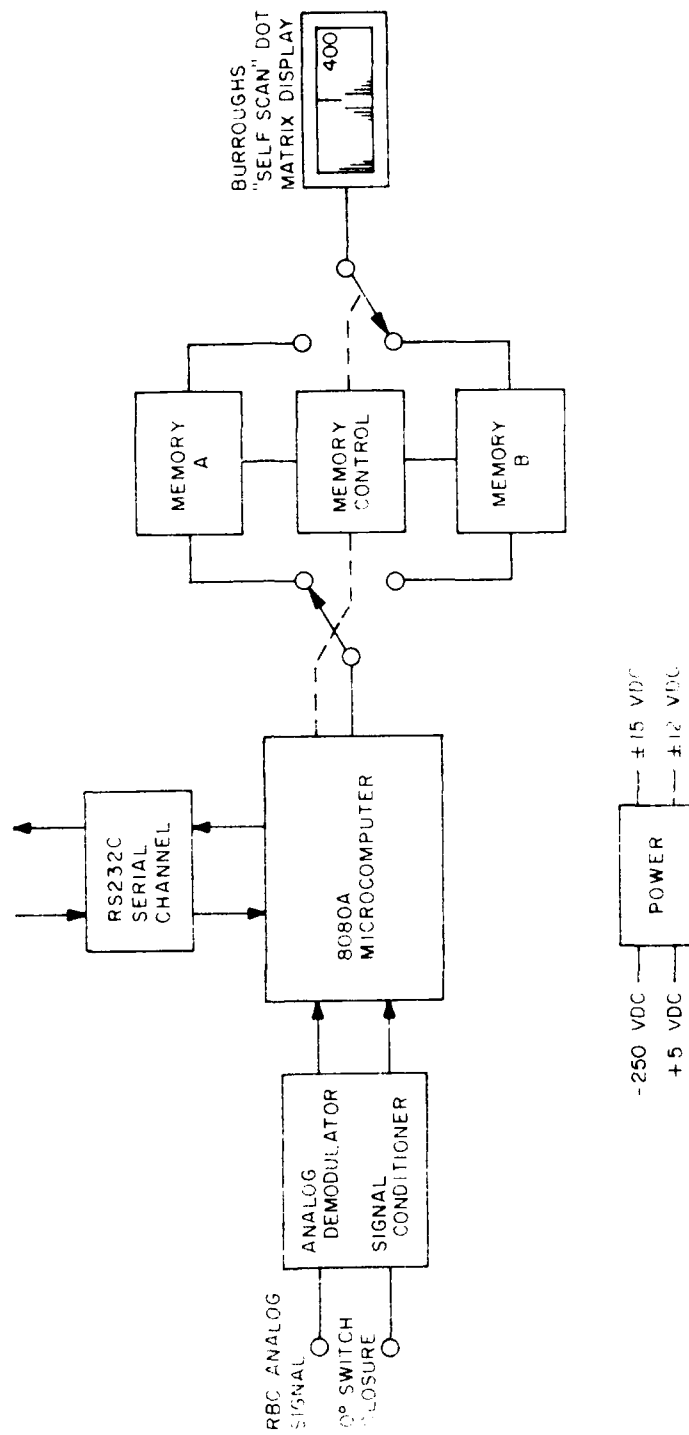
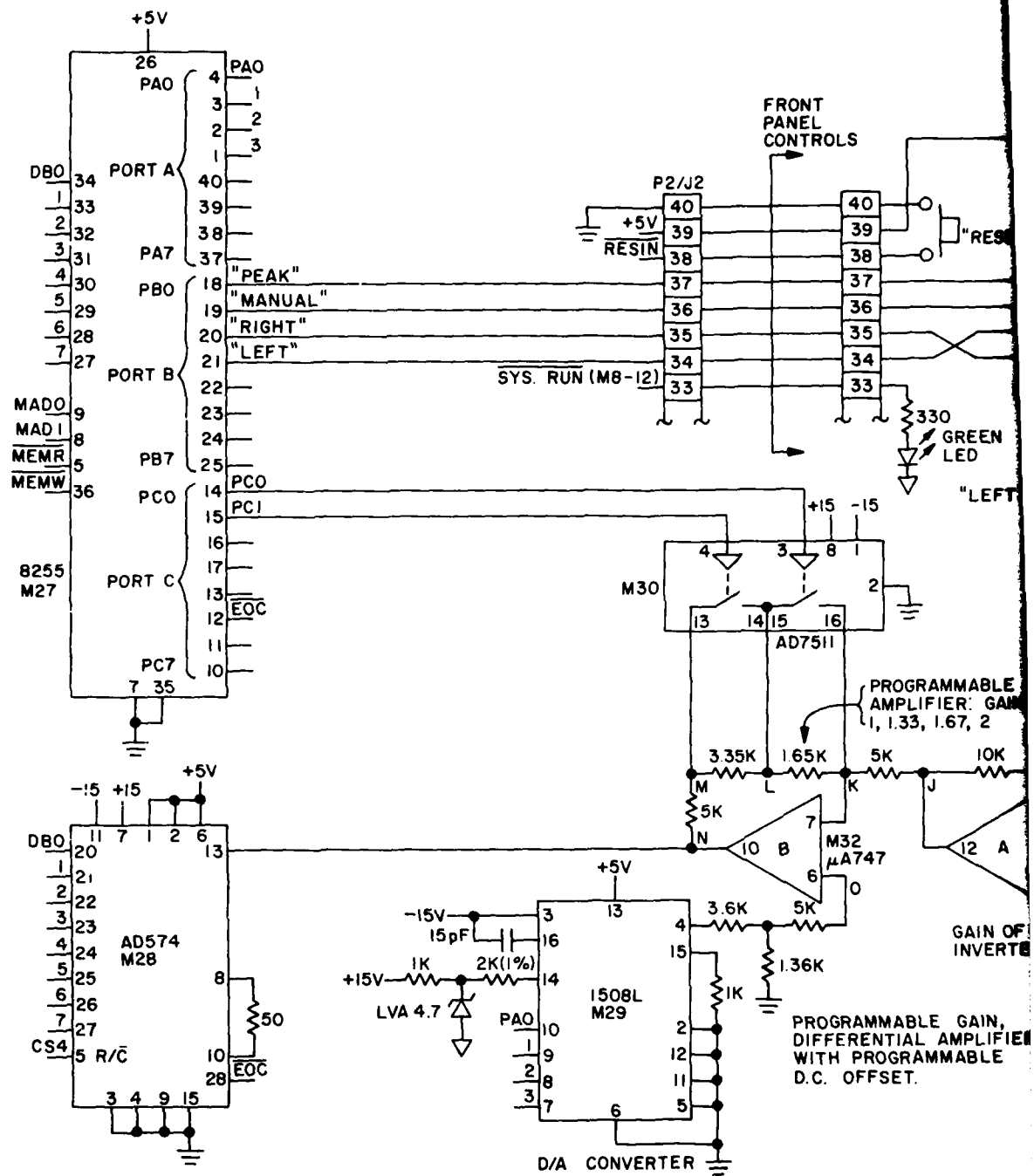
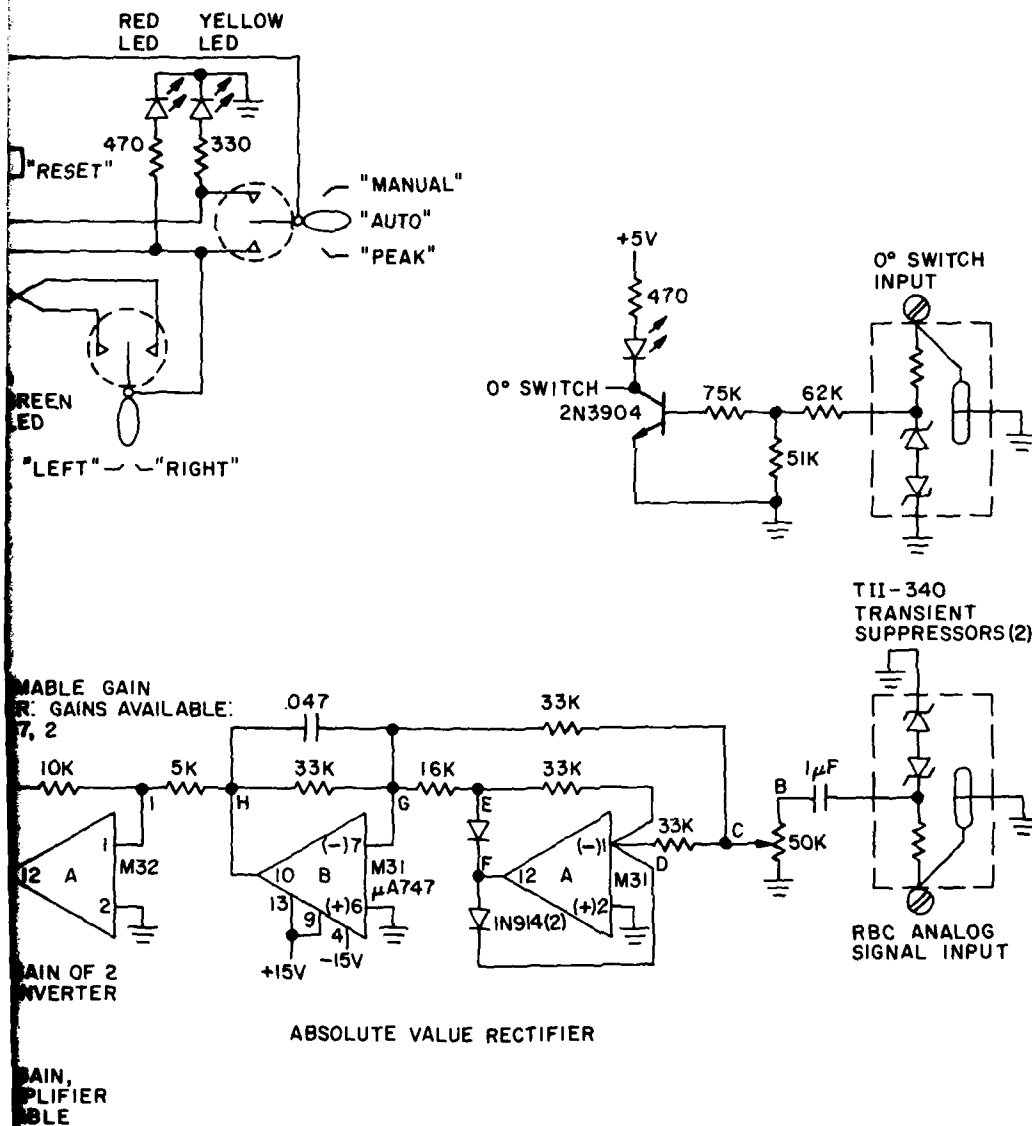


Figure 2 RBC Display Processor (Block Diagram)





THIS CIRCUITRY IS LOCATED ON DISPLAY
CONTROLLER BOARD (EXCEPT AS NOTED)

Figure 3. Analog Demodulator and Signal Conditioner (Schematic Diagram)

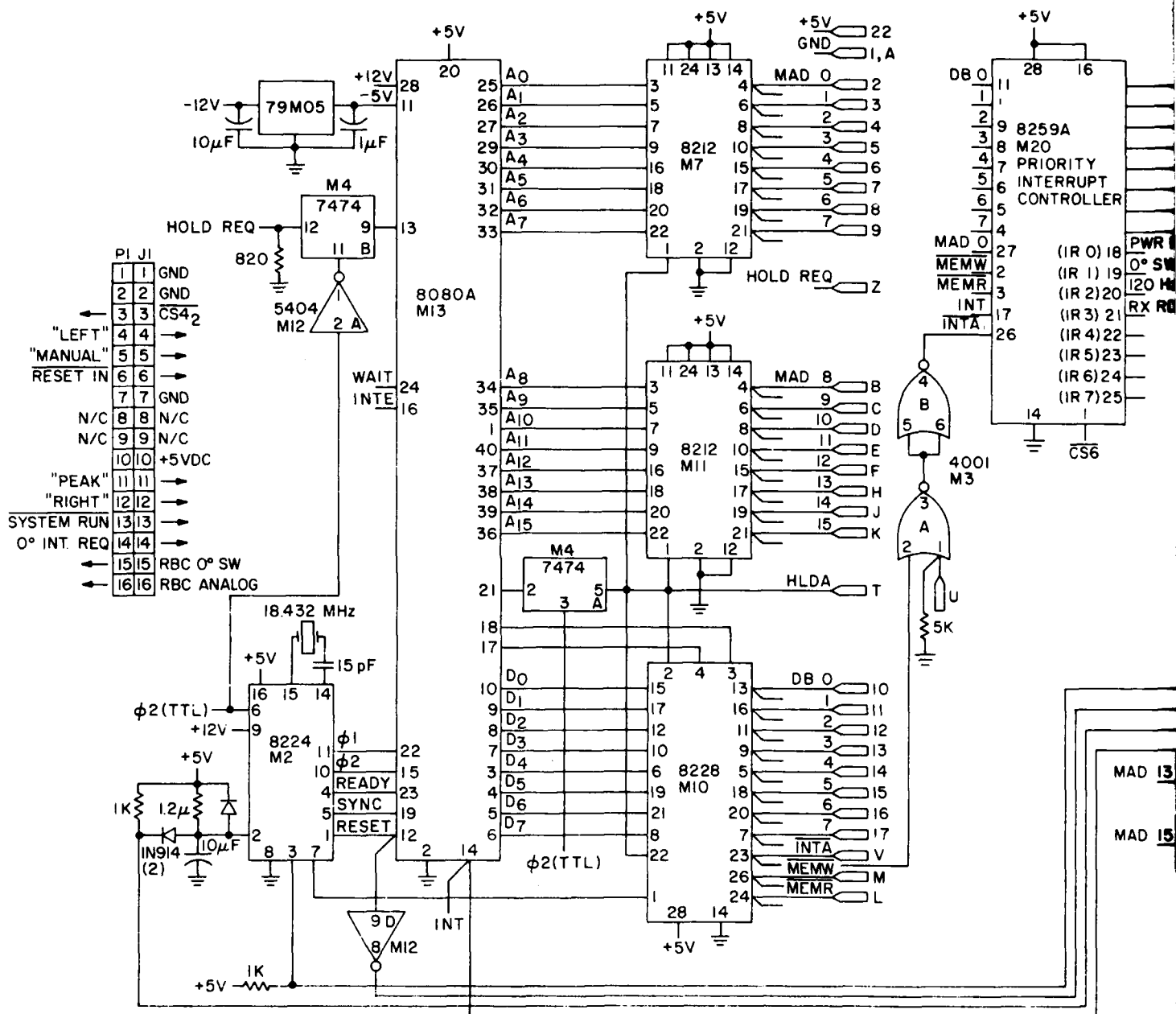
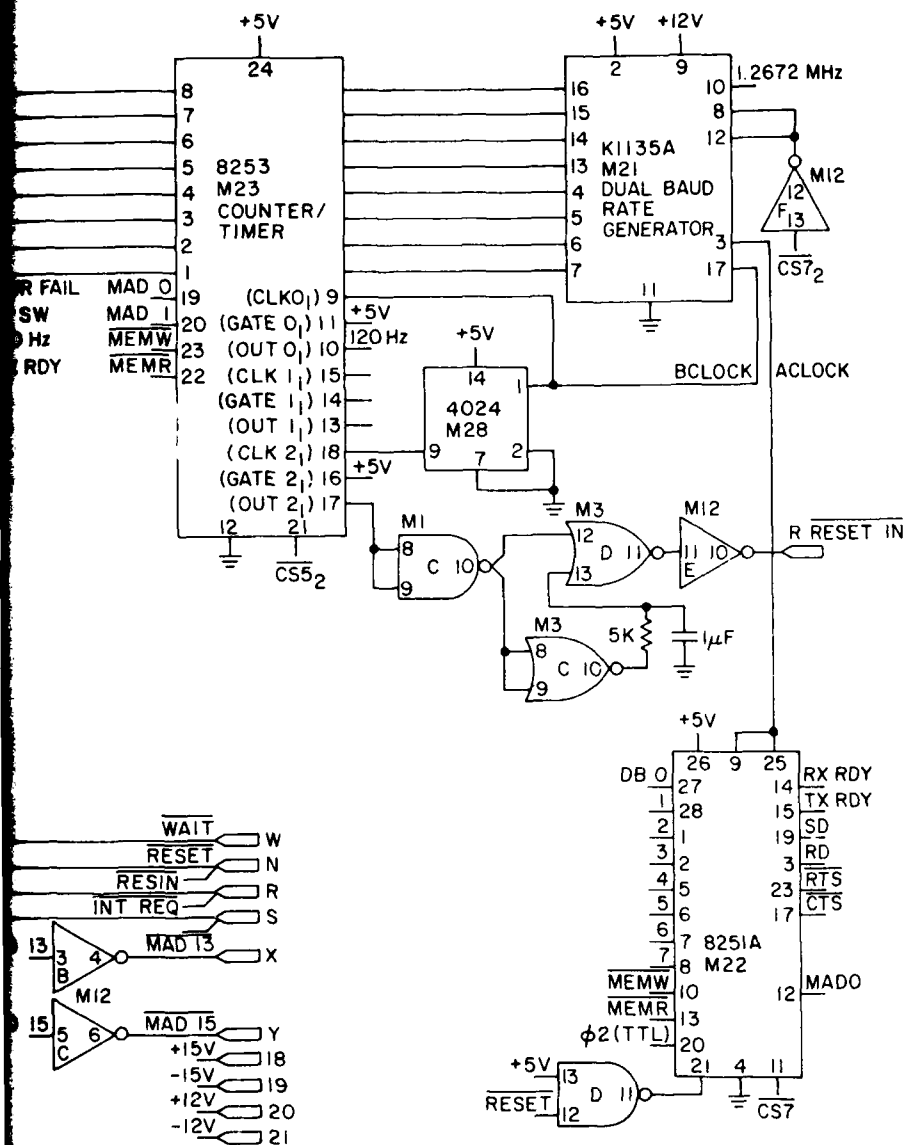
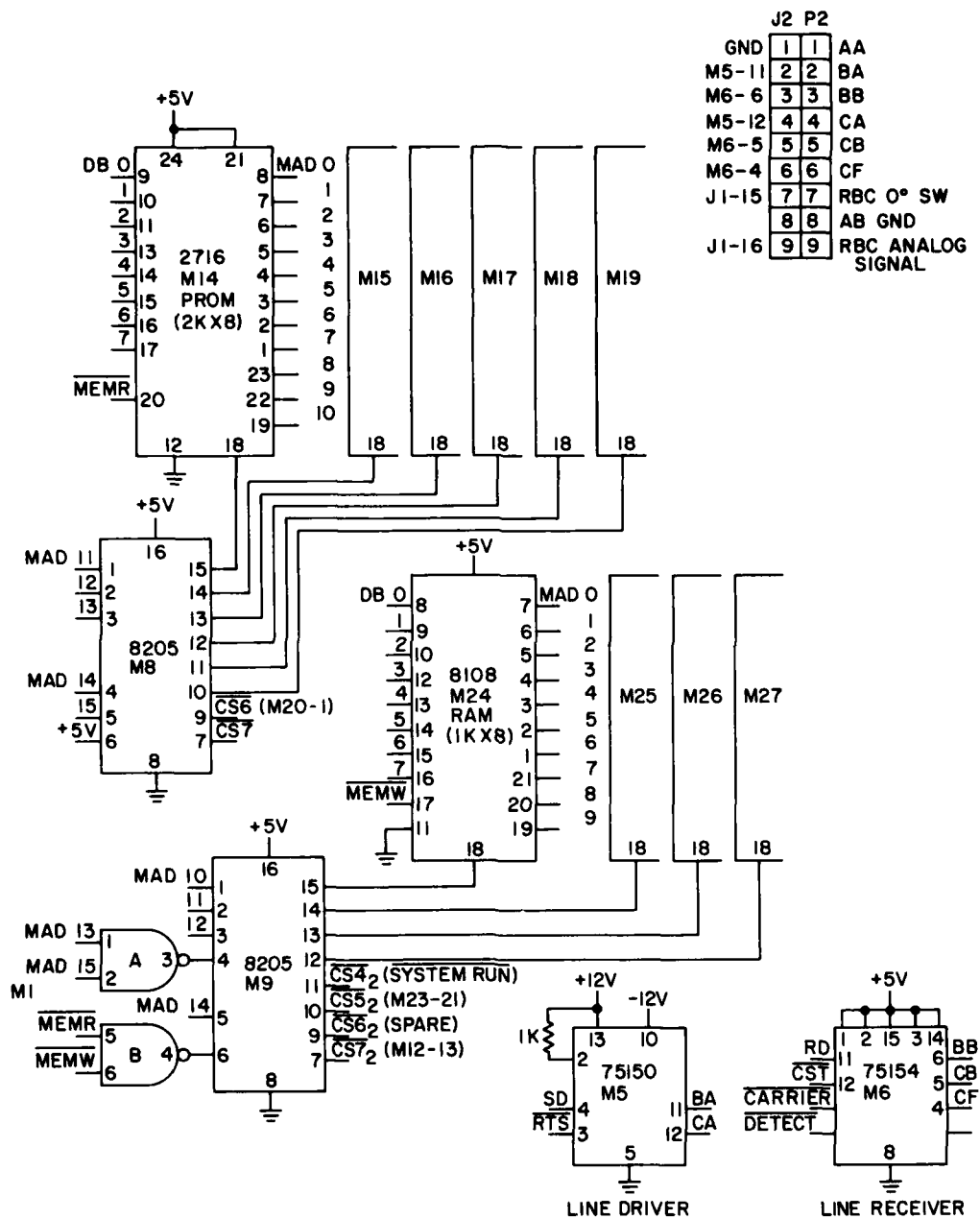


Figure 4. RBC Display Computer Card
(Schematic Diagram) (Sheet 1 of 2)





The system incorporates an eight level priority interrupt controller (M20), a triple 16 bit counter/timer (M23), a USAR/T serial port (M22) with RS 232C interface and a programmable baud rate generator (M21). This combination of devices, contained on one board, has been found to be a very useful "workhorse" micro-computer in earlier lab designs and provides considerable computer power and versatility in this application.

The dot matrix display includes a Burrough's SSD-0124-0039 panel and driver assembly.² The assembly consists of a SELF-SCAN panel display and driver board. The SELF-SCAN panel display is a flat panel gas discharge display device capable of displaying a matrix of 17 cells by 192 cells which can be configured in any font or graphic application. The cells are located on 0.040 in. centers vertically and 0.030 in. centers horizontally.

The driver board consists of a 3-phase counter, anode drivers, phase drivers, missing pulse detector, blanking control, and scan anode disable control circuit. The display clock signal is used to sequentially address the cathode phases. The data input signals are used to address the display cells at the appropriate location. The driver board contains a protection circuit. If loss of the clock or reset signal is detected, it switches off the display and scan anodes to protect the display panel from being damaged while operating in a non-scan mode.

The display operates in a scanning mode, scanning from left to right, one full column at a time. A clock must be included with a period of one column time. At the end of the 192nd display column time, the negative edge of the reset pulse must be generated to initiate a new scan cycle. The reset duration must be at least one clock period.

Seventeen data input lines are provided to address the PNP transistors which are operated in a constant current source configuration to drive the display cells.

The protection circuit consists of two retriggerable monostable multivibrators arranged in a missing pulse detector configuration to monitor the clock and reset input signals. Failure to receive high to low transitions on either line at appropriate time intervals disables the display anodes, scan anodes, and cathode phase drivers. The presence of these signals at a later time would automatically resume the normal operation of the system.

The simplified schematic of the display controller circuit is shown in Figure 5 (Figures 6 and 7 are the detailed schematics). The Burroughs SELF SCAN dot matrix display is organized into 192 sixteen bit vertical columns. A 17th bit is wired true to provide an illuminated baseline.

2. Burroughs Bulletin No. 3511A, Burroughs Corporation, OEM Division, P.O. Box 1226, Plainfield, New Jersey 07061.

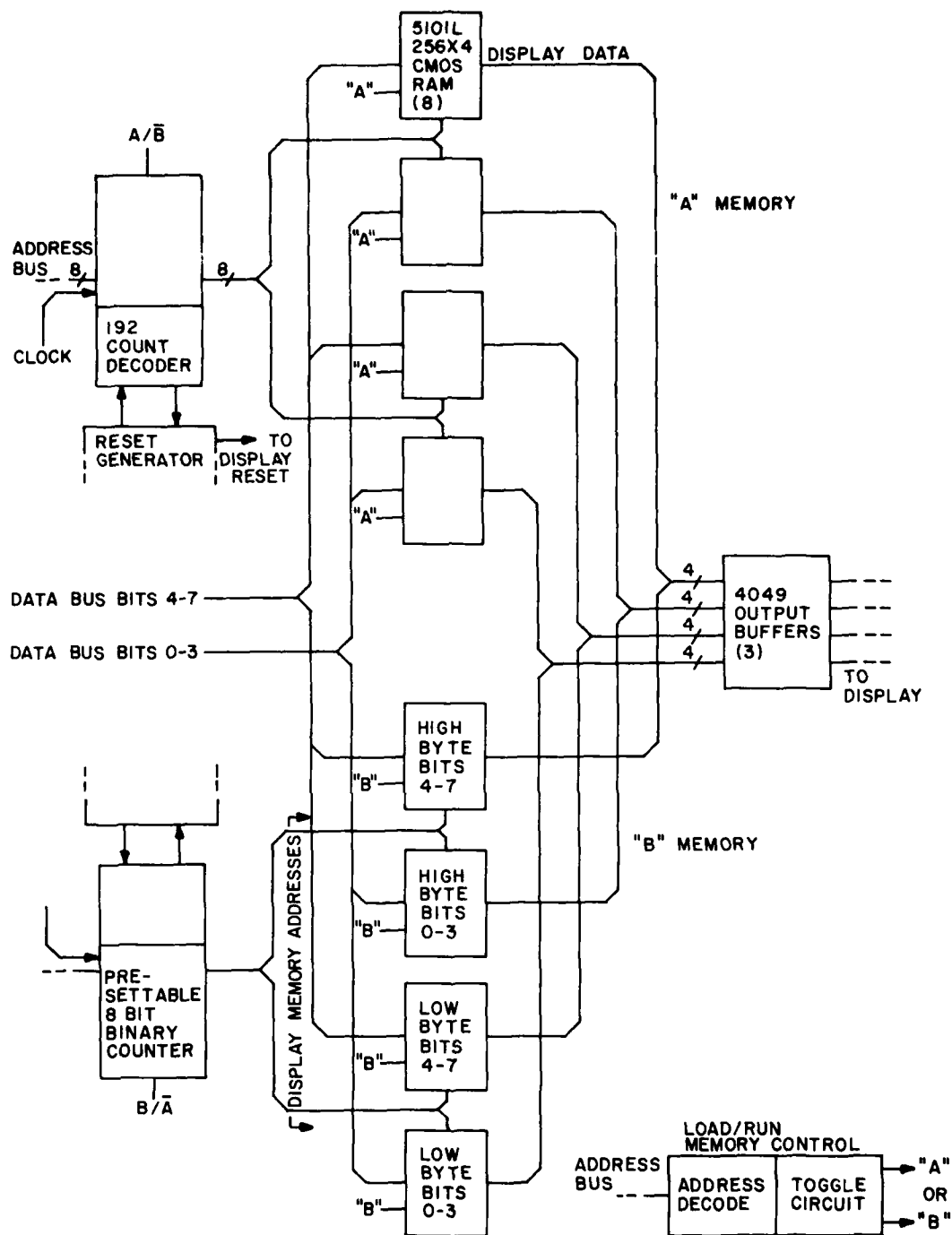
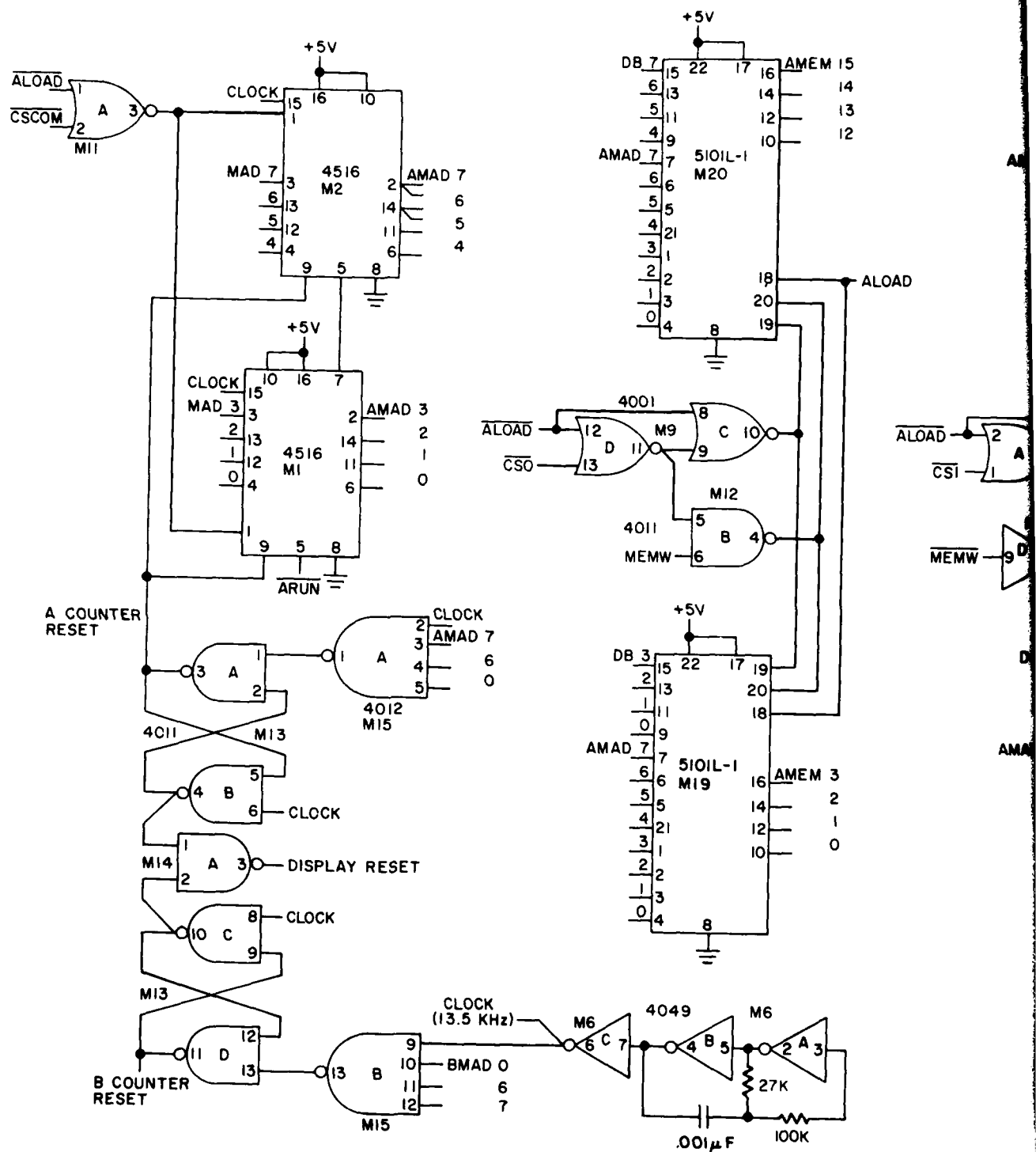
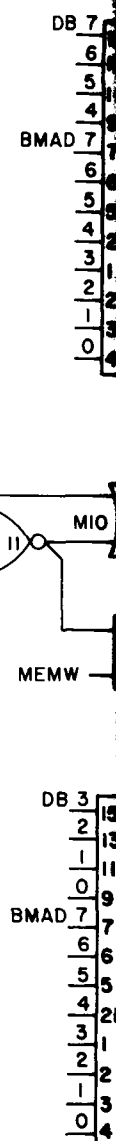
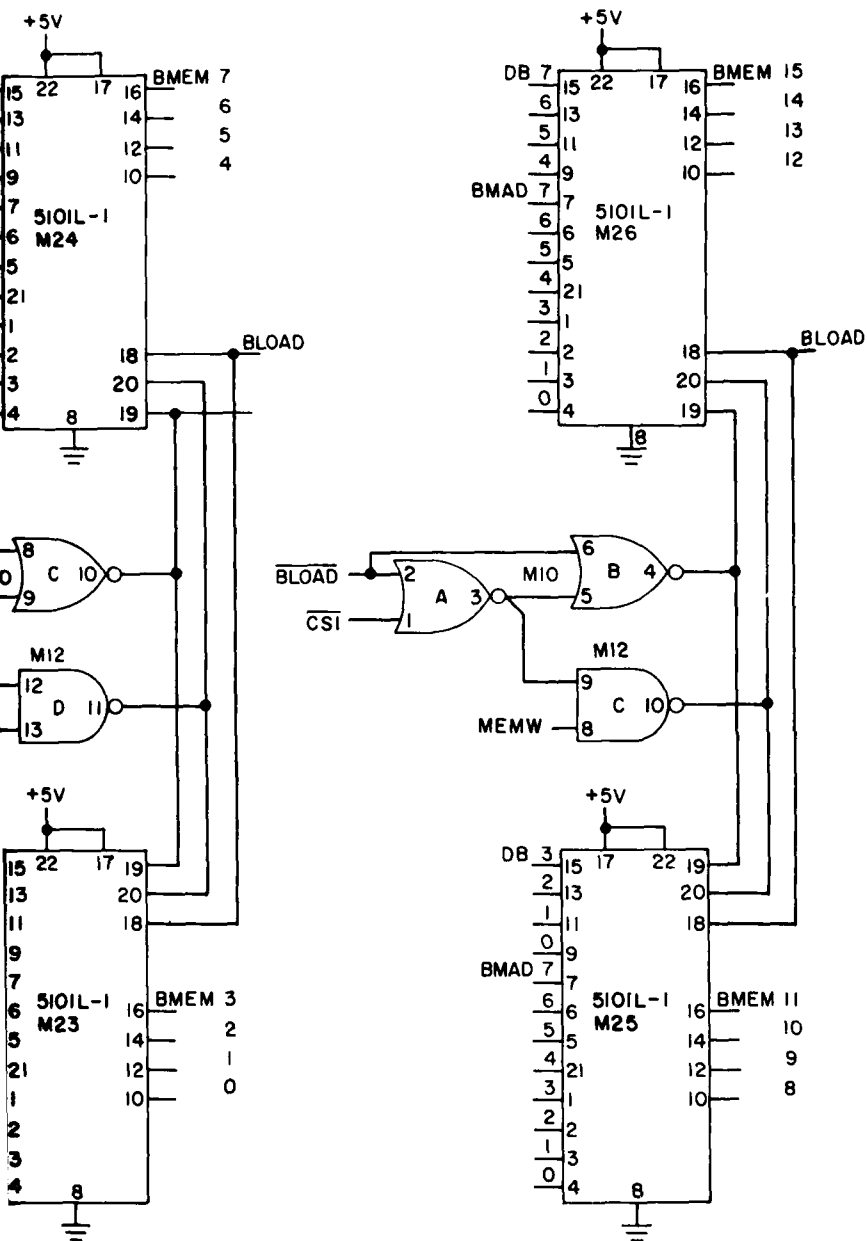


Figure 5. Double Dual-Port Display Memory (Simplified Schematic)





1



To create an image, a data field is stored in the display memory in the form of 192 low byte/high byte pairs or columns. Approximately 70 times each second an address generator made up of an oscillator (M6) and binary counters (M1 and M2 in A memory or M3 and M4 in B memory) provides sequential addresses and data strobes to write a data field to the display from left to right. The 193rd clock pulse, conditioned by the counter's decoder (M15), develops a Display Reset strobe. If the computer is not requesting a memory "toggle" or update, the memory merely cycles and continues to display the same data. A "toggle" request, stored by M7, will cause a switch to the alternate memory synchronized by Display Reset. Contents of the alternate memory are now clocked to the display in the above manner.

A dual purpose 8 bit presettable binary counter (whose operation is illustrated in Figure 8) develops or buffers display memory addresses depending on memory selection. If in "load" state, the counter is in its presettable condition and merely buffers the lower 8 bits of the computer's address buss directly to the selected display memory. When data is to be written to this memory, the computer uses an ordinary memory-write instruction. When toggled to "run" state, display memory addresses are developed as the counter accumulates column-strobing clock pulses.

A total of seven dc voltages are required to operate this equipment: +5, -5, ± 12 , ± 15 , and -250. The +5 Vdc logic power is developed from the 120 V, 60 Hz line by means of a high efficiency, switching type power supply. With the exception of -5 Vdc, the other voltages are developed by dc to dc converters powered by the +5 V supply. The -5 Vdc power for the 8080A is developed from the -12 Vdc source through a three terminal regulator.

3. ROTATING BEAM CEILOMETER DATA

The configuration and operation of the RBC was described in some detail in Section 2.1. Basically, the detector is set a fixed distance away from the projector (normally 400 ft), with its field of view vertical and coplanar with the rotating projector beam. The sensor's intersection volume advances up the detector's vertical beam as the projector's beam rotates from the horizontal. When the volume coincides with a cloud, as illustrated in Figure 9, backscatter of the projector's beam by water droplets in the cloud is detected by the receiver and depicted on the indicator. The elevation angle (α) at which the maximum backscatter return occurs yields the cloud height (h) by triangulation.

It is characteristic for the RBC response to increase slowly above the noise level, reach a maximum, and decrease gradually to the noise level. This is illustrated in Figure 10. Referring to Figures 9 and 10, the return begins to

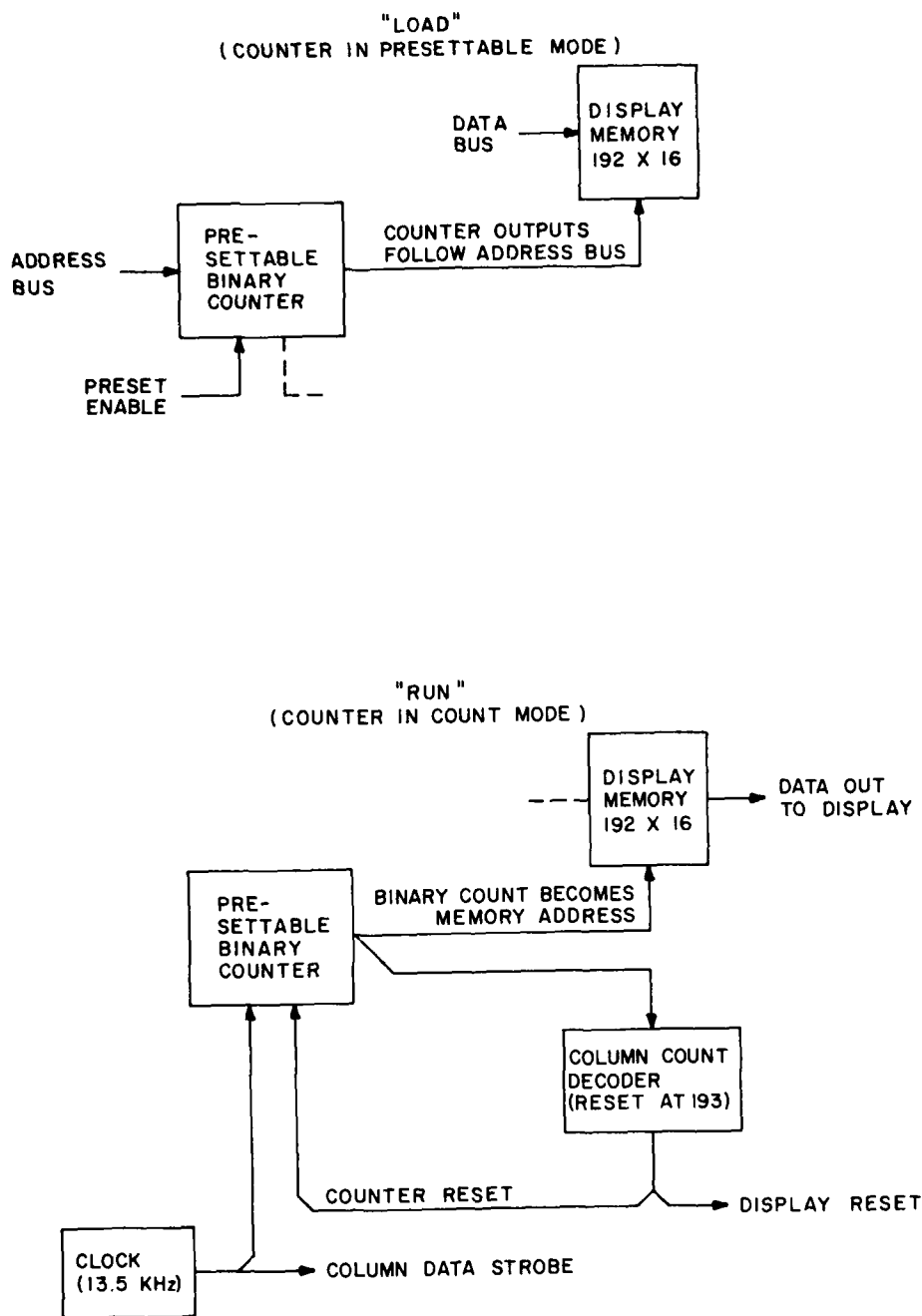


Figure 8. Display Memory Load/Run Detail

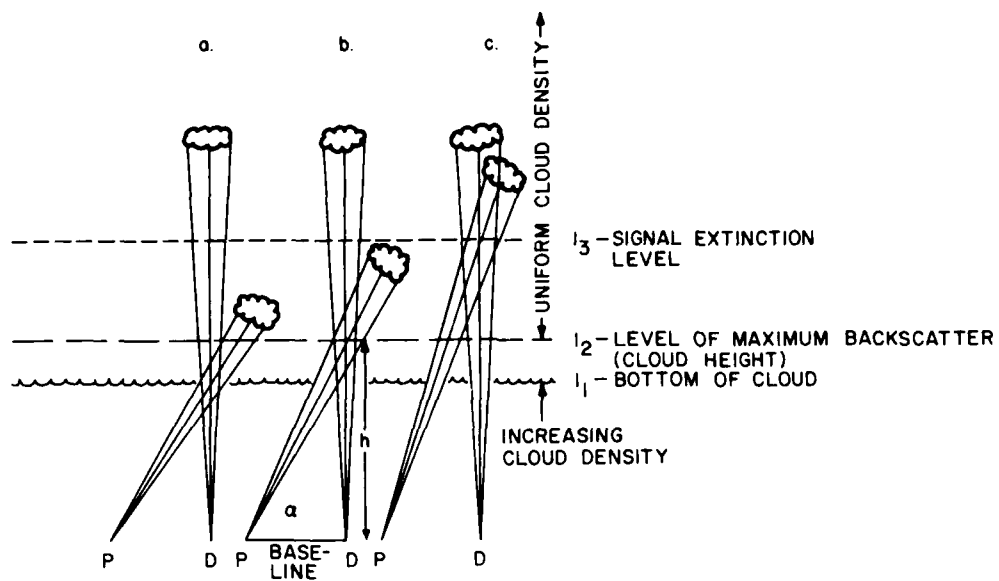


Figure 9. RBC Measurement Geometry

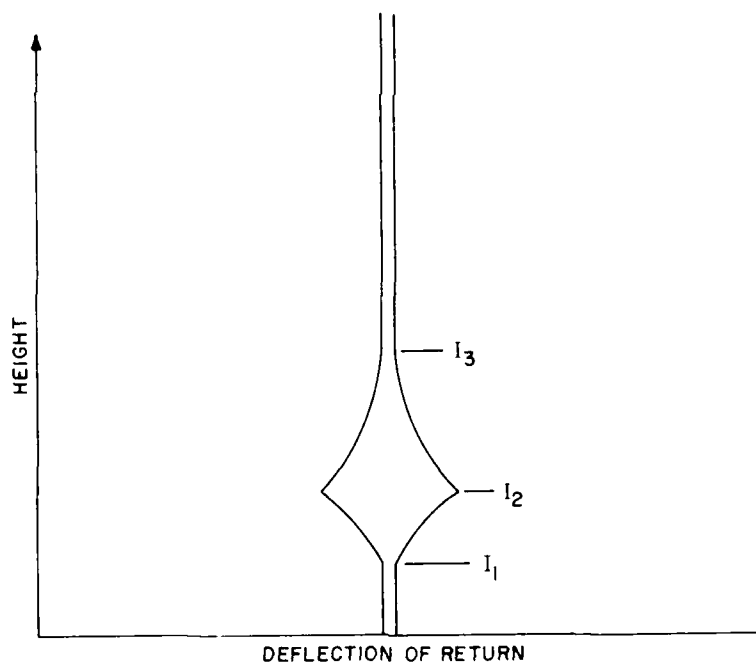


Figure 10. Diagram of Old RBC Display

increase when the optical axes intersect at I_1 , the bottom of the cloud. The signal strength reaches a maximum at the level of maximum backscatter I_2 , and then decreases to the noise level at the signal extinction level, I_3 .

Moroz and Travers³ have suggested that this idealized return is altered by the effects of multiple scatter. They postulated that sufficient multiple scatter occurs to create signals above noise level which would be detected just before the two beams intersect the cloud. In addition, multiple scatter between the two beams reinforces the receiver's signal, suggesting that the maximum return would occur at a higher altitude than it exists. Quantitative adjustment for multiple scatter effects were not estimatable from their study.

For the purposes of our development and test, cloud heights were determined in accordance with Federal Meteorological Handbook No. 1 guidelines which state, "The base of sky cover aloft is represented by points of maximum reaction (peak return) of the ceilometer to light reflected from the base of layers detected."⁴

4. AUTOMATED DATA PROCESSING PROCEDURES

The computer program developed to handle the RBC processing is shown diagrammatically in Appendix A and the PL/M-80 source code is listed in Appendix B. It was written to process the output from a single Rotating Beam Ceilometer (RBC) using an Intel 8080A microprocessor. It has three main objectives. The first is to represent the data received from the RBC on the display system in a manner similar to the operational CRT display. The second is to filter out spurious or insignificant returns. The last objective is to examine the retained data and to determine a representative cloud base height for the single RBC observing point. It should be noted that each of the two lamps in the RBC is processed independently, almost as if they were two separate instruments. This prevents minor differences in one lamp (for example, lamp intensity variation, a slightly different calibration) from influencing the other.

The logic of the program is based upon the acquisition of certain start signals from the RBC and internally generated 120 Hz clock pulses. Within the RBC, each 6 sec scan of information from one lamp has two parts; a 3 sec, 0 to 90 deg cloud base height data scan and a 3 sec, 90 to 180 deg lamp inactive scan. However, for processing purposes, the 180 deg scan is divided into three parts. The first 5 deg

3. Moroz, E. Y., and Travers, G. A. (1975) Measurement of Cloud Height, AFCRL-TR-75-0306, AD A015 737.

4. Federal Meteorological Handbook No. 1, Surface Observations, Second Edition (1979), Washington, DC, pp A5-5 to A5-29.

is used to calculate a gain adjustment, 5 to 90 deg is taken as cloud base height data, and 90 to 180 deg is stored to calculate an offset adjustment for the next cycle of the lamp. A zero degree switch closure signal received from the projector is used to initiate data collection.

The program operates on an interrupt-driven basis. This allows the micro-processor to continuously execute its main program, stopping only to service peripheral devices when it is told to do so by the device itself. In effect, the interrupt method provides an external input that directs the processor to complete the instruction currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, the processor automatically resumes exactly where it left off. This allows the processor to operate in a very efficient manner. In addition, a priority or level of importance can be placed on each interrupt. This permits a higher priority interrupt to stop the processing of a lower priority interrupt, allowing the higher one to be serviced immediately. In this system the zero degree switch closure is assigned the highest priority followed by the internally generated 120 Hz clock pulse.

When first put into (or returned to) operation, the computer initialization program formats certain component chips and presets certain variables. The system then acquires the first zero degree switch closure to signal the upcoming receipt of data pulses. Once the program has received the first zero degree switch closure, data acquisition is synchronized by the 120 Hz clock. When a 120 Hz clock pulse is received, the main processor is interrupted and a data acquisition sequence is accomplished. This produces data samples every 8.3 msec or 0.25 degree of elevation, thereby dividing the 180 deg scan into 720 data samples.

Two corrections, obtained from previous scans of the same lamp, are applied to each received data scan. The first, a gain correction applied to pulses 20-719 (5 to 179.75 deg), is determined for each lamp from its previous scan. During the initial segment of the scan a strong signal return is developed by a small reflector (mounted on the detector housing) which is illuminated by the projector beam. The signal reaches a maximum value near zero degrees elevation and decays to a background level of 0.3 to 0.5 V before reaching 5 deg of elevation. This signal approximates a strong cloud base return and is used as a reference for the gain correction. This correction, which compensates for degraded signal intensity, is the factor needed to adjust the signal level that occurs in the first 5 deg, to a value representing a full intensity lamp. Four discrete gain adjustments are permitted: 1.00, 1.33, 1.67, and 2.00. The second correction, an offset applied to pulses 20 to 360 (5 to 90 deg), is determined from a 10-scan running average of the noise or background level of the detector. It is determined during the 3 sec portion of each sweep when the projector beam does not intercept the detector beam (pulses

361 to 719 covering the range 90.25 to 179.75 deg). A simple average of the low-level noise (background) peaks received during this part of the scan is integrated into a 10-scan running mean (which spans 2 min) to provide the offset factor for the next rotation of the same lamp. For this purpose, a noise peak is defined as an increase in the amplitude of the return signal followed by a decrease. Offset factors range from near 0 to 2.35 V in 16 discrete and equal intervals. After one completes the 3 sec lamp inactive period, the other lamp trips the zero degree switch, becoming the active lamp, and the processing repeats itself alternating back and forth from one lamp to the other. The primary data acquisition phase of the program is completed with the switch-over to the other lamp.

One additional piece of preprocessing should be noted. During cloudless daylight periods varying amounts of sunlight may, on occasion, be scattered into the detector depending on time of day and orientation of RBC. If these returns occur during the cloud base data segment (0 to 90 deg), erroneous cloud base heights may be reported. Because these returns are produced by sunlight and not the RBC lamp, they are just as likely to occur during the 3 sec inactive scan of the lamp. Therefore, if two or more peaks collected during the inactive scan have magnitudes of 25 percent or more of full scale, sunglint is assumed and a cloud base height is reported as none. Two or more peaks are used to reduce the possibility that other random noise sources, which may produce one large noise peak, are not misinterpreted as sunlight.

Once the 360th data pulse has been collected, the program automatically begins processing the cloud height data scan. While this processing is occurring, the program, through the interrupt method discussed before, continues to gather data (from pulses 361 to 719) to be used to calculate the offset correction. The first processing step is to prepare the data for display. The maximum return in the first 20 pulses (5 deg) is located. If the zero degree switch is perfectly positioned, this maximum return should occur at the first pulse (zero degrees). If this is not the case, the program shifts the largest return into the zero degree position. This self-calibrating procedure ensures a correct height being determined and prevents inconsistency between the height reported by two lamps which may not have been calibrated in exactly the same manner.

The geometry of the RBC and experience with its performance suggest clouds above 87.5 deg would not scatter sufficient light back into the detector to be discernable with confidence. Therefore, subsequent processing is restricted to pulses 0 to 352 (0 to 87.5 deg). The next step is to reduce the range of pulses (0 to 352) and the range of signal intensity (0 to 255) to fit the array set aside for the 192 by 17 dot matrix display. Selecting every other pulse reduces its effective range to 0 to 176. Dividing the signal intensity by 16 reduces its range to 0 to 15. Thus, if original pulse number 36 had a signal intensity of 158, then the 18th

element of the display array would have bits 0 to 8 lit and bits 9 to 15 unlit. (Internally the program would assign a 1 bit to array dots on the display to be lit.) The last portion of this pass through the data is to locate the maximum return within the pulse range 20 to 352. A cursor is displayed at the element of the array which represents this maximum return. For this element of the display array, the lit/unlit logic is reversed. Thus, in the example above, bits 0 to 8 in element 18 would be unlit and bits 9 to 15 would be lit. If the magnitude of all data points in the range 20 to 352 was zero, the cursor would not be displayed.

The system allows for three modes of operation or processing which will be described in subsequent paragraphs. They are the one-scan and the five-scan modes (which require no further inputs from the user once they have been selected) and the manual mode (which is user-interactive). These options are exercised by the user/operator through a switch on the display's front panel. Switching from one mode to another, as required to satisfy the operator's data needs, is easily handled by the display's microprocessor.

The full range data array (0 to 352 pulses and 0 to 255 signal intensity) is filtered and analyzed to find a cloud base height. The first step in this stage involves the detection and rejection of large amplitude single pulse noise spikes. Any pulse (from numbers 21 to 351) whose magnitude is greater than the magnitude of the average of its two adjacent neighbors by 200 percent or more is rejected and replaced by the average of the adjacent values. Then a binomial function is applied to eliminate high frequency components whose duration is less than four pulses. Ripples of less than four pulse duration which are part of substantially higher peaks will also be filtered out. However, since the major peaks persist over a large number of pulses, these will not be eliminated. In this process the absolute location of slope reversal (which identifies each peak) will be preserved. A 7 weight filter is used in this binomial function with the following weights: $1/64$, $6/64$, $15/64$, $20/64$, $15/64$, $6/64$, $1/64$. The filter weights are successively applied to the set of pulses, using each pulse as the central value in turn and adjacent pulses to define the wings of the function. The 7 point filter process is initially centered on pulse 24 (using pulses 21 to 27) and ends with pulse 348 (using pulses 345 to 351).

The identification of peaks in the filtered scan is restricted to pulses between pulse count 24 and 348. Simply stated, a series of pulses constitutes a valid peak if the pulse magnitudes are increasing for at least four consecutive pulses and decreasing for the two succeeding pulses. When a peak is identified, the pulse count and the magnitude at the point of slope reversal (indicating the summit of the peak) are stored for later analysis. After the fully filtered scan has been examined, the magnitudes of the stored peaks are ranked in descending order. The program then stores the peak count with largest magnitude for latter analysis. In its R&D

capacity, this automated basic processing system served the needs of a follow-on study, which used the largest and second highest peak in the evaluation of automated cloud observation systems.⁵ Therefore, the second largest peak was also stored. If no peaks were identified during the scan, the program stored 1000 for both the maximum and secondary maximum peaks. However if a maximum peak exists and a secondary peak does not, 1000 is stored only for the secondary maximum.

The determination of the objective cloud base height is initiated after five scans have been obtained for a lamp. Then after each successive scan an objective cloud base height is calculated for each lamp using the maximum peak pulse count determined for each of the most recent five scans. Recall that five scans of one lamp span 1 min in total time. Initially, the five peaks are compared and if two or more of them have the same pulse count, that count is selected as the cloud base height. If two values have multiple occurrences, the lowest is selected. For example, if peak pulse count values of 140, 120, 120, 100, and 100 are reported, 100 is selected as the cloud base height. If all five values are different, a second check is made to determine if two or more pulse counts are within 4 counts of each other. If they are, the lowest pulse count is stored as the cloud base height. For example, if values of 92, 54, 96, 97, and 52 are reported, 52 is stored. If the first two tests fail, a final attempt is made. If but one of the five values is within 6 pulse counts of the value identified in the previous cloud base height determination for the lamp under consideration, then the lower of these two values is retained. If all three tests fail then the cloud base height is reported as none by storing 1000. At this point in the program, a check is made to determine if the one-scan or the five-scan mode has been selected. In the one-scan mode, the five-scan cloud base height determination is made but not used. This permits a rapid update when the five-scan mode is reselected. The pulse height count, representing the largest return received in the current scan, is converted to a height in feet using Appendix C. The numeric characters for the height, the stored cursor position, and the stored display array are then sent to update the dot matrix display. If the magnitudes of the data points were zero, "NO PEAK" would be displayed for the height and the cursor would be suppressed. In the five-scan mode, the most recent stored data array, the stored cursor position, and the numeric characters for the cloud base height are displayed. Again, the CBH (in feet) is determined using Appendix C. If a cloud base height was determined to be none or produced by sunglint, "NO CBH" is displayed.

After the data for a given lamp have been analyzed and displayed, the program completes the collection of offset data and waits for the other lamp's zero degree

5. Geisler, E. B. and Chisholm, D. A. (1980) An Automatic Cloud Observation System (ACOS), AFGL-TR-81-0002, AD A094 330.

switch closure. When the zero degree switch closure occurs, cloud base data collection proceeds using the interrupt mode for the new lamp. An offset correction for the previous lamp is calculated and stored for the next cycle while the basic processing described above continues for the new lamp.

The third and final mode of operation is the manual mode. While this mode continues to allow data to be collected, it inhibits the display from updating, thereby allowing the operator to evaluate the current display at length. The cursor positioner, a front panel switch, is enabled so that the operator can move the cursor to the left or to the right to investigate secondary maxima or other interesting features in the trace. When the positioner is toggled and not held, the cursor moves only one column to allow exact positioning. However, if the switch is held in the toggled position, the cursor moves very rapidly to permit longer moves. In either the toggled or the toggled and hold operation, a conversion of the cursor position to the equivalent height in feet is accomplished and automatically updated in the display's upper right corner.

5. FUTURE CONSIDERATIONS

In the studies conducted thus far, an effective method to process and report an obscuration has not been found. This is largely due to the fact that the RBC's response to an obscuration varies tremendously depending on the weather element causing it and its intensity and thickness. Procedures tested thus far, in the demonstration of the Modular Automated Weather System at Scott AFB, IL⁶ and more recently at AFGL and Otis AFB, have not yielded a fully adequate solution. Data being gathered from a three RBC network at AFGL's Weather Test Facility on Otis AFB, MA are being used to try to resolve this difficult automation task. When developed, obscuration procedures could be incorporated into the processing program.

In a related study,⁵ hierarchical clustering techniques for automated cloud observations have been evaluated using data gathered by the three RBC network at the Weather Test Facility at Otis AFB. This clustering technique, applied to the latest 30 min of cloud data provided by our RBC processing procedures, yielded reliable and representative observation of low cloud height, low cloud amount, and ceiling. It also found that specification accuracy is not measurably improved by using basic data from a second or third nearby RBC and that the use of secondary peak information does not add to improved specification. It is our contention,

6. Chisholm, D.A., Lynch, R.H., Weyman, J.C., and Geisler, E.B. (1980) A Demonstration Test of the Modular Automated Weather System (MAWS), AFGL-TR-80-0087, AD A087 070.

however, that if secondary peaks survive the filters and tests applied to them in the program, they could be valuable information in support of aircraft operations. Expanding the processing and display capability to include these added pieces of information (low cloud height, amount and ceiling based on updated 30 min data sets and the secondary peaks) would require added display space beyond that presently included. An alternative solution would allow the operator to display low cloud observations and related data using a front panel switch which could direct the computer to temporarily hold RBC scans and display alphanumeric data instead.

Another change would be required in order to implement these modifications throughout the Air Weather Service inventory. Procedures are needed to account for differences in the RBC baseline (distance from projector to detector) and line frequency which can be 50 cps (cycles per second) at some overseas bases compared to 60 cps in the U.S. These variations can be accounted for through internal switches which can be set during installation consistent with local conditions.

Finally, a test routine, which would be implemented at the option of the observer-operator, could be included in the system. Its purpose would be to assist in the diagnosis of suspected system malfunctions. When the program appears to be functioning improperly, a known data set would be generated by an internal signal simulator. Comparison of anticipated outcomes with actual outcomes of the several processing steps would aid the operator in isolating system malfunctions (if any). Work is presently underway to incorporate the test routine and the inventory-wide option capability into the display system.

6. SUMMARY AND CONCLUSIONS

Although the RBC is an aging sensor, with proper maintenance and handling it should be useful to the AWS for several more years. Further, the cost to completely replace the RBCs in the Air Force inventory would be very high. This study has demonstrated the advantages of using a low cost microprocessor-based automated display system to replace the display portion of the current system. Although the prototype display system can continue to undergo modifications and improvements, it goes a long way towards overcoming the operational and maintenance problems encountered with the current CRT display.

In addition to solving the maintenance problems, the display/processor provides an added dimension that was not previously available to the operator. It provides a reliable, repeatable scan which minimizes the subjectivity of the operator and the misalignment of certain components. In addition, the system displays the exact height of the maximum return for each scan to an accuracy never obtainable before. Fine scale variations can also be seen. The system provides the operator with

options to select one of three modes to process and display the RBC data. Two modes are fully automated: a one-scan mode which operates only on the latest scan, and a five-scan mode which objectively determines an updated 1 min cloud base height. The third, a manual-interactive mode, allows the observer to hold the current scan and fully investigate, by means of a moving cursor, primary peaks, secondary peaks, or other areas of interest. This mode permits the observer to extract in detail all the information a scan has to offer.

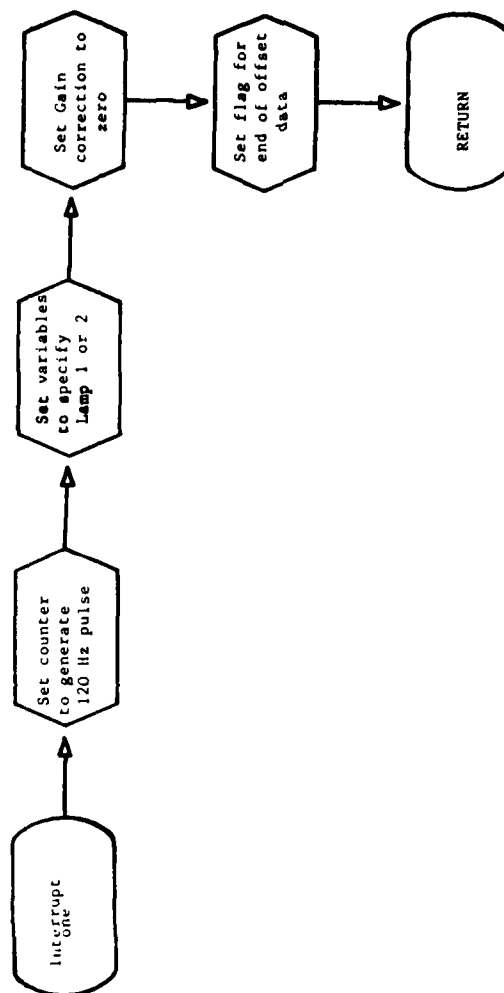
In November 1980, a prototype RBC display system was installed in the Base Weather Station (BWS) at Scott AFB, IL for a MAC Service Test. It was interfaced to the operational RBC sensors and was located in close proximity to the operational CRT display. Operating procedures and evaluation criteria were briefed to BWS, Hq AWS and AFCC personnel. From our experience with the system and from responses received from Scott personnel, the automated display system has proven to be very beneficial. The display is more readable, provides more accurate heights, and permits better interpretation of the data obtained. In its present form, or more so with the improvements suggested above, this display system can significantly enhance the utility of the operational RBC and is a step towards more accurate and reliable observations.

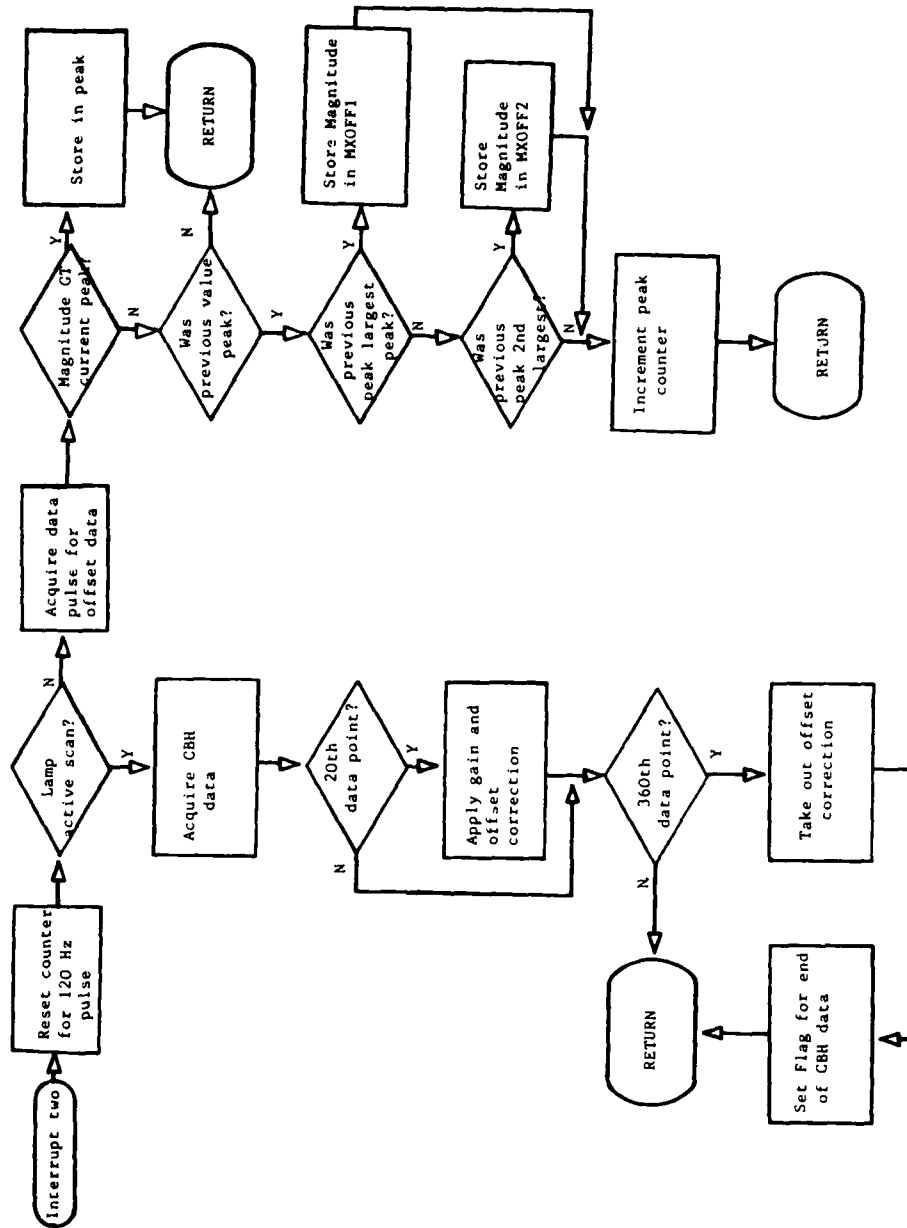
References

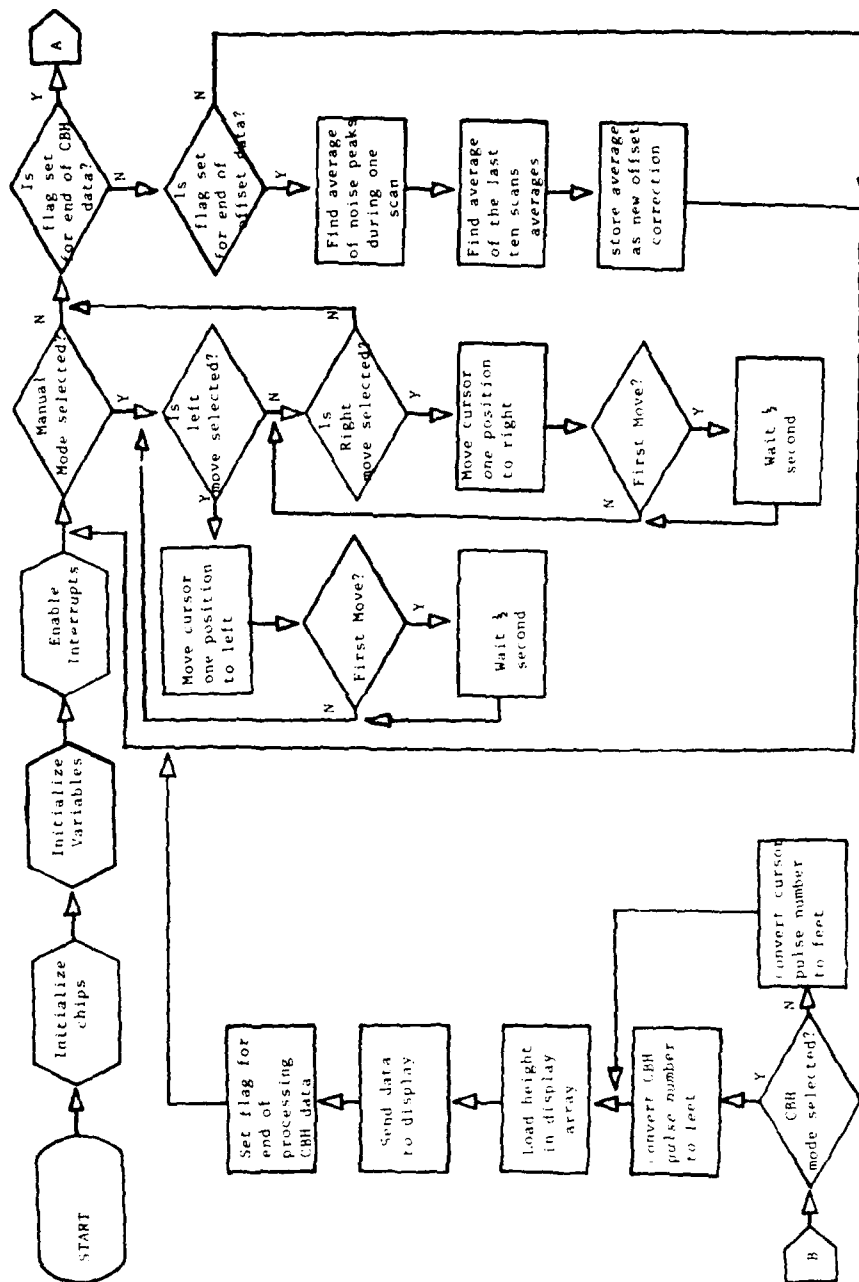
1. Air Force Communications Service (1977) Automated Weather Distribution System (AWDS) Required Operational Capability (AFSC ROC 601-77).
2. Burroughs Bulletin No. 3511A, Burroughs Corporation, OEM Division, P.O. Box 1226, Plainfield, New Jersey 07061.
3. Moroz, E. Y., and Travers, G. A. (1975) Measurement of Cloud Height, AFCRL-TR-75-0306, AD A015 737.
4. Federal Meteorological Handbook No. 1, Surface Observations, Second Edition (1979), Washington, DC, pp A5-5 to A5-29.
5. Geisler, E. B. and Chisholm, D. A. (1980) An Automated Cloud Observation System (ACOS), AFGL-TR-81-0002, AD A094 330.
6. Chisholm, D. A., Lynch, R. H., Weyman, J. C., and Geisler, E. B. (1980) A Demonstration Test of the Modular Automated Weather System (MAWS), AFGL-TR-80-0087, AD A087 070.

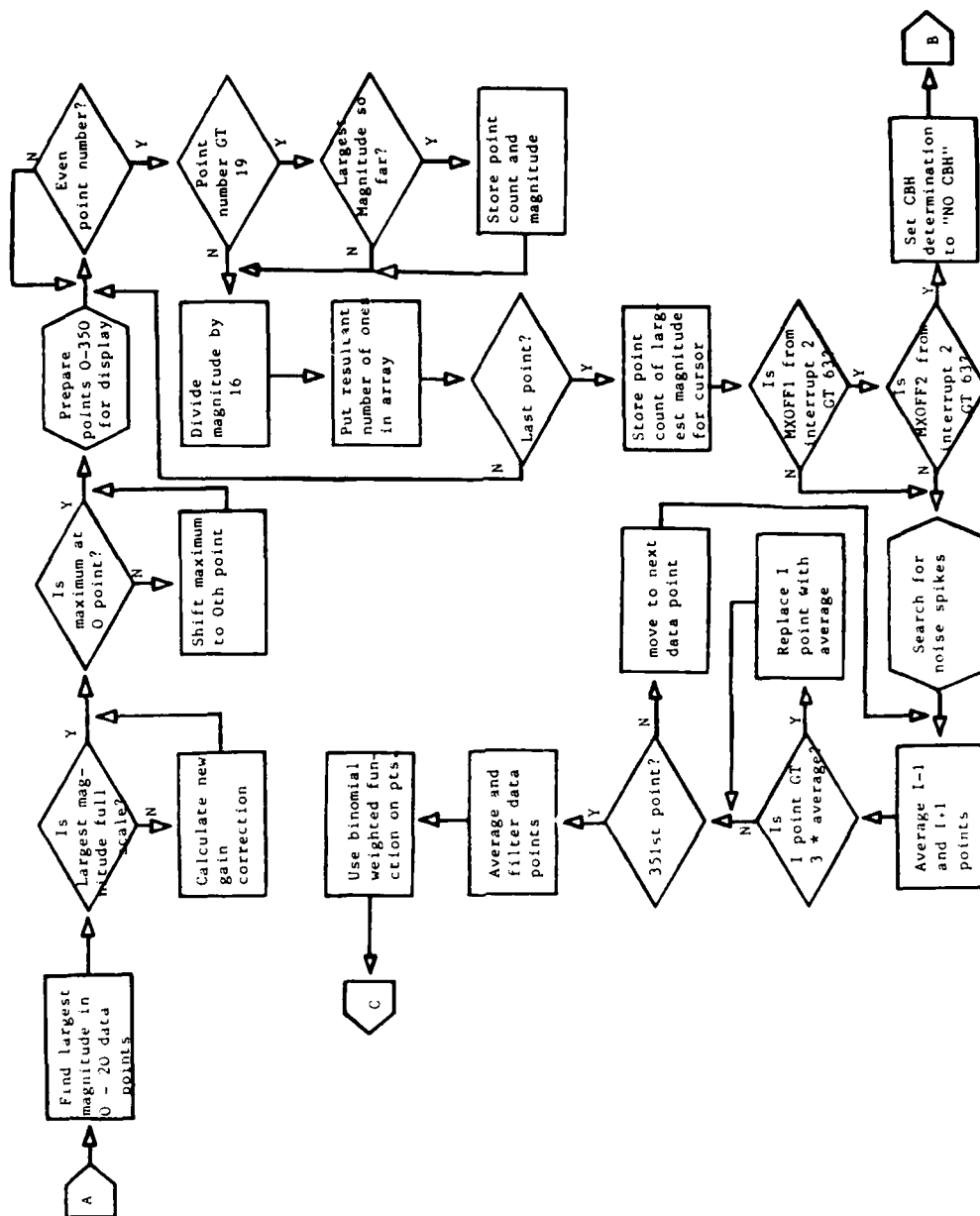
Appendix A

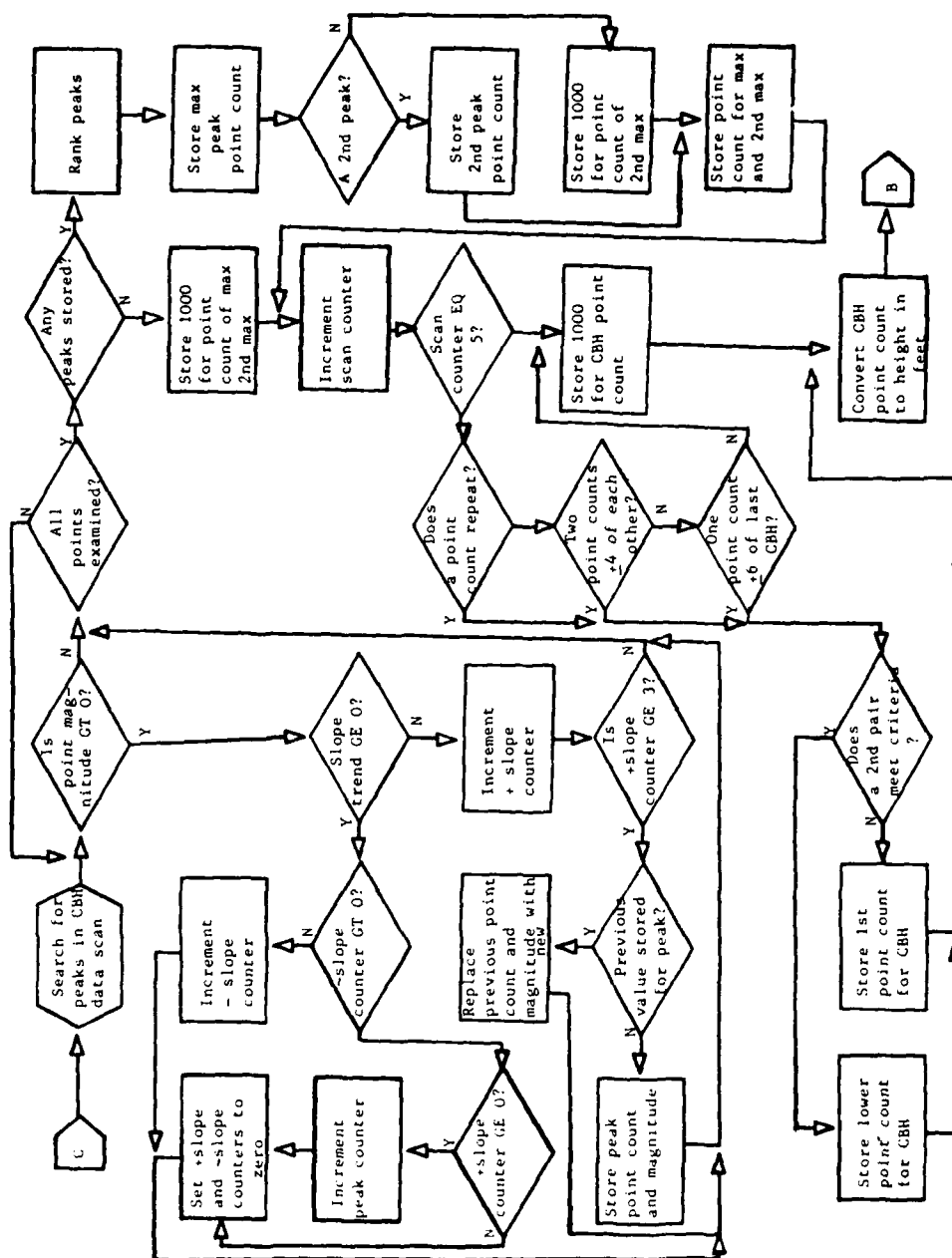
Flow Chart of Processing Program











Appendix B

PLM-80 Software of Processing Program

PL/M-80 COMPILER RBC DISPLAY - MAIN PROGRAM

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE RBCDISPLAYPROGRAM
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 :F1:RBCDIS.SAV TITLE('RBC DISPLAY - MAIN PROGRAM') NOOBJECT

```

1      $PRINT(1LP:)  DEBUG INTVECTOR(4,40H)
2      RBCDISPLAYPROGRAM: DO:
3      DECLARE DCL LITERALLY 'DECLARE':
4      DCL (CHECK,CNT,COFF,CURSOR,DONE,GETOFF,J,K1,K2,K3,K4,KS,K6,K7,LEFT) BYTE PUBLIC:
5      DCL (MANUAL,MAX,MAX1,MAXN,MXOFF1,MXOFF2,NO,NO2,OFFS,REFLAG) BYTE PUBLIC:
6      DCL (RIGHT,SCAN,X) BYTE PUBLIC:
7      DCL (CTR,I,IC,K,POFF) ADDRESS PUBLIC:
8      DCL A(361) BYTE PUBLIC, BEIL(2) ADDRESS PUBLIC, C(20) BYTE PUBLIC:
9      DCL CEL(2) ADDRESS PUBLIC, CLOUD(2) ADDRESS PUBLIC, CLOUDY(22) ADDRESS PUBLIC:
10     DCL DIS(200) ADDRESS PUBLIC, GAIN(2) BYTE PUBLIC, GRAND(2) BYTE PUBLIC:
11     DCL HEIT(20) ADDRESS PUBLIC, OCT(2) BYTE PUBLIC, OFFSET(2) BYTE PUBLIC:
12     DCL PEAK(80) BYTE PUBLIC, PEAKO(200) BYTE PUBLIC, PULSE(80) ADDRESS PUBLIC:
13     DCL RCT(2) BYTE PUBLIC:
14     DCL HEAD(16) ADDRESS PUBLIC DATA(
15     01H,03H,07H,0FH,1FH,3FH,7FH,FFH,1FFH,3FFH,7FFH,OFFFH,1FFFF,3FFFF,7FFFF,OFFFFF):
16     CHECK1: PROCEDURE (NO2) EXTERNAL:
17     DCL NO2 BYTE:
18     END CHECK1:
19     DATACHECK: PROCEDURE (P,Y,Z) EXTERNAL:
20     DCL (Y,Z) BYTE, P ADDRESS:
21     END DATACHECK:
22     DETOFF: PROCEDURE EXTERNAL:
23     END DETOFF:
24     FILL: PROCEDURE EXTERNAL:
25     END FILL:
26     GAINER: PROCEDURE (X,Y) EXTERNAL:
27     DCL (X,Y) BYTE:
28     END GAINER:
29     INTONE: PROCEDURE EXTERNAL:
30     END INTONE:
31     INTTWO: PROCEDURE EXTERNAL:
32     END INTTWO:
33     MTIME: PROCEDURE (X) EXTERNAL:
34     DCL X BYTE:
35     END MTIME:
36     OFFCAL: PROCEDURE EXTERNAL:
37     END OFFCAL:

```

PL/M-80 COMPILER RBC DISPLAY - MAIN PROGRAM

```

36 1 RESET: PROCEDURE EXTERNAL;
37 2 END RESET;

38 1 SCANS: PROCEDURE(N02) EXTERNAL;
39 2 DCL N02 BYTE;
40 2 END SCANS;

41 1 VARY: PROCEDURE EXTERNAL;
42 2 END VARY;

43 1 INTER1: PROCEDURE INTERRUPT 1;
44 2 CALL INTONE1;
45 2 MEMORY(0000H) = 20H;
46 2 END INTER1;

47 1 INTER2: PROCEDURE INTERRUPT 2;
48 2 CALL INTTWO1;
49 2 MEMORY(0000H) = 20H;
50 2 END INTER2;

51 1 BEGIN: DISABLE;
52 1 MEMORY(8403H) = 30H; /*8253 COUNTER 0 LD/RD*/
53 1 MEMORY(8400H) = 0FH; /*8253 LS BYTE LOAD*/
54 1 MEMORY(8400H) = 0FH; /*8253 MS BYTE LOAD*/
55 1 MEMORY(8C00H) = 0E7H; /*PROGRAM RAUD RATE 1200*/
56 1 MEMORY(0000H) = 5AH; /*8259 ICW1 ADDRESS START 0040H,EDGE TRIGGERED,INTERVAL 4,SINGLE, NO ICW4*/
57 1 MEMORY(0001H) = 00H; /*8259 ICW2 TOP BYTE OF ADDRESS*/
58 1 MEMORY(0001H) = 0FH; /*MASK FOR 8259*/
59 1 MEMORY(0801H) = 7AH; /*8251 MODE WORD: XMIT ENABLE,RECVR ENABLE,RESET ERR FLAGS,REQUEST TO SEND*/
60 1 MEMORY(0801H) = 35H; /*8251 COMMAND WORD: XMIT ENABLE,RECVR ENABLE,RESET ERR FLAGS,REQUEST TO SEND*/
61 1 I = MEMORY(0800H); /*CLEAR RECEIVER BUFFER*/

62 1 MEMORY(6A03H) = 6AH; /*FORMAT 8255 FOR RBC*/
63 1 MEMORY(6A00H) = 00H; /*SET BIAS LEVEL 0 OF 16*/
64 1 MEMORY(6A02H) = 60H; /*SET GAIN OF 1.0H*/

/*INITIALIZE VARIABLES FOR PROPER STARTING CONDITIONS*/

65 1 CHECK = 0AH;
66 1 GAIN(0),GAIN(1) = 00H;
67 1 OFFSET(0),OFFSET(1) = 02H;
68 1 BEIL(0),BEIL(1),CEIL(0),CEIL(1) = 1000;
69 1 RCT(0),RCT(1),DCT(0),DCT(1) = 0;
70 1 GRAND(0),GRAND(1),CLOUD(0),CLOUD(1) = 0;
71 1 K1,K2,K3,K4,K5,K6,K7 = 0;
72 1 CNT,CODE,IC,MXOFF1,MXOFF2 = 0;
73 1 DONE,GETOFF,POFF = 1;
74 1 RFLAG = 4;
75 1 IC = 362;
76 1 MEMORY(0001H) = 0F9H;
77 1 ENABLE;
78 1 LOOP: MEMORY(8000H) = 00H;
79 1 MANUAL = MEMORY(6A01H) AND 03H;
/*SET MASK FOR 8259,ALLOW INTER 1 AND 2*/
/*SYSTEM RUN LIGHT*/
/*WHEN THE SYSTEM IS PLACED IN MANUAL MODE BIT 1 OF THIS
LOCATION BECOMES 1*/

```

PL/H-80 COMPILER RBC DISPLAY - MAIN PROGRAM

```

80 1 IF MANUAL = 02H THEN DO1
82 2 DO WHILE MANUAL = 02H1
83 3 MANUAL = MEMORY(6601H) AND 02H1
84 3 LEFT = MEMORY(6601H) AND 04H1
85 3 RIGHT = MEMORY(6601H) AND 08H1
86 3 CTR = 01
87 3 DO WHILE LEFT = 04H1
88 4 IF CURSOR > 0 THEN DO1
90 5 DIS(CURSOR) = NOT DIS(CURSOR)1
91 5 IF CURSOR = 192 THEN CURSOR = 1751
93 5 ELSE CURSOR = CURSOR - 11
94 5 IF NO = 1 THEN DO1
96 6 NO = 01
97 6 DO 1 = 151 TO 1651
98 7 DIS(1) = DIS(1) AND 00FFH1
99 7 END1
101 7 CALL FILL1
102 7 CALL RESET1
103 5 END1
104 4 CALL VARY1
105 4 LEFT = MEMORY(6601H) AND 04H1
106 4 END1
107 3 DO WHILE RIGHT = 08H1
108 4 IF CURSOR < 175 THEN DO1
110 5 DIS(CURSOR) = NOT DIS(CURSOR)1
111 5 CURSOR = CURSOR + 11
112 5 CALL FILL1
113 5 CALL RESET1
114 5 END1
115 4 CALL VARY1
116 4 RIGHT = MEMORY(6601H) AND 08H1
117 4 END1
120 1 IF DONE = 0 AND (RELAG = 2 OR RELAG = 4) THEN DO1
122 2 RCT(K3) = RCT(K3) + 11
123 2 GRAND(K3).MAX = 01
124 2 DO 1 = 0 TO 191
125 3 IF A(1) > GRAND(K3) THEN DO1
127 4 GRAND(K3) = A(1)1
128 4 MAX = 11
129 4 END1
131 2 CALL GAINER(GRAND(K3).K3)1
132 2 IF MAX > 0 AND MAX < 9 THEN DO1
134 3 DO 1 = 0 TO (360 - MAX)1
135 4 A(1) = A(1)+MAX1
136 4 END1
138 2 MAX1 = 01

```

```

/*DO IF IN MANUAL MODE*/
/*REMAIN IN LOOP UNTIL TAKEN OUT OF MANUAL MODE*/
/*WHEN IN MANUAL MODE AND TOGGLE SWITCH IS PLACED IN LEFT
CURSOR THEN BIT TWO WILL BECOME TRUE*/
/*WHEN IN MANUAL MODE AND THE TOGGLE SWITCH
IS PLACED IN RIGHT CURSOR THEN BIT THREE WILL BECOME TRUE*/
/*COUNTER TO MAKE FIRST MOVE SLOW AND THEN SPEED UP*/
/*CONTINUE TO DO UNTIL FALSE*/
/*CORRECT LAST CURSOR POSITION TO REGULAR VIDEO*/
/*CHANGE CURSOR POSITION TO THE LEFT*/
/*NO PEAK OR NO CBH WAS LAST THING PRINTED, AND
NOW MUST CLEAR THAT AREA IN MEMORY*/
/*FILL ARRAY AND SEND TO DISPLAY*/
/*CHECK ON DONE BECAUSE WE ARE NOT GOING TO PROCESS DATA*/
/*CHECK IF A SLOW MOVE IS REQUESTED OR FAST ON*/
/*CHECK IF TOGGLE IS STILL ENGAGED*/
/*DO UNTIL TOGGLE IS DISENGAGED*/
/*ONLY RIGHT AS FAR AS DATA IS SHOWN*/
/*CORRECT POSITION OF LAST CURSOR SO IT BECOMES REGULAR VIDEO*/
/*INCREASE CURSOR COUNT BY ONE*/
/*FILL NEW ARRAY AND DISPLAY*/
/*CHECK ON STATUS OF DONE! WILL NOT PROCESS DATA*/
/*FAST OR SLOW MOVE*/
/*CHECK IF TOGGLE IS STILL ENGAGED*/
/*THEN DO1 /*CBH DATA HAS ALL BEEN RECEIVED AND MUST BE
PROCESSED*/
/*RUNNING TOTAL OF THE NUMBER OF SCANS THAT HAVE BEEN
PROCESSED TO NOW! MAX OF 5*/
/*GRAND WILL BE USED TO STORE THE INTENSITY OF LARGEST PEAK
FOUND IN THE FIRST 4.75 DEGREES*/
/*MAX WILL BE USED TO STORE THE LOCATION OF THE LARGEST PEAK
IN THE FIRST 4.75 DEGREES*/
/*DO LOOP TO FIND THE MAX INTENSITY IN THE FIRST 4.75 DEGREES*/
/*LARGER MAGNITUDE FOUND IN THE SCAN*/
/*STORE MAGNITUDE*/
/*STORE LOCATION*/
/*CALCULATE A NEW GAIN CORRECTION BASED ON THE VALUE OF GRAND*/
/*IF MAX INTENSITY WAS NOT AT ZERO DEGREES SHIFT IT!
THE MOST IT CAN BE SHIFTED IS 8 POINTS (2 DEGREES)*/
/*DO LOOP TO SHIFT THE DATA POINTS BECAUSE THE MAX INTENSITY
WAS NOT AT ZERO DEGREES*/
/*MAX1 IS THE MAXIMUM INTENSITY OF THE CBH DATA COLLECTED
DURING THE LAST SCAN! USED WITH CURSOR PLACEMENT*/

```

PL/M-80 COMPILER

```

/*H*V*IN IS THE MAXIMUM NUMBER WHICH IS THE COUNT OF THE MAX
PEAK IN THE SCA*V*
/*DO LOOP TO GO THROUGH THE CBH DATA TO FIND POSITION OF
MAX PEAK AND VALUE THERE*/
/*K* IS A DUMMY VARIABLE FOR PLACEMENT OF CURSOR AND MAIN
SINCE THE DISPLAY ONLY SHOWS EVERY OTHER DATA POINT*/
/*IF CBH DATA POINT IS ZERO PUT ZERO IN DISPLAY ARRAY*/
/*ALL OTHER NON ZERO CBH DATA POINTS*/
/*DO NOT WANT A PEAK IN FIRST FIVE DEGREES BECAUSE THAT IS
WHERE THE START BANG IS*/
/*CURRENT CBH DATA POINT IS LARGER IN MAGNITUDE THAN PREVIOUS
MAX FOUND*/
/*STORE MAGNITUDE OF MAX FOUND*/
/*STORE COUNT NUMBER OF MAX FOUND*/
/*RANGE OF THE DIGITAL VALUES IS 0 -255 HOWEVER ONLY 0 - 15
CAN BE SHOWN ON DISPLAY SO DIVIDE BY 16*/
/*SET CURSOR TO COUNT OF THE MAX MAGNITUDE FOUND*/

DAY THE RBC SIGNAL IS VERY NOISY AND ALOT
S. TO ELIMINATE THOSE RETURNS A CHECK OF THE
RING THE OFF SCAN. IF THESE TWO ARE FOUND
IS ASSUMED AND THE MAX AND 2ND MAX FOR
NO PEAK OR A NO CBH CONDITION*/

FIRST MAX GREATER THAN A 1/4 SCALE*/
SECOND MAX GREATER THAN A 1/4 SCALE*/
PREVIOUS CBH WAS "NO CBH"*/
K1 WILL FIND THE CORRECT PLACE IN HEIT*/
MAKE SURE A NO CBH IS REPORTED AS A 5 SCAN VALUE*/
DO LOOP TO PLACE THE VALUE FOR NO PEAKS IN HEIT*/

DETERMINE THE MAX AND 2ND MAX PEAK FOR THE CURRENT SCAN
TER FILTERING AND AVERAGING*/
/*FINDS A 5 SCAN REPRESENTATIVE VALUE*/

/*THE OFFSCAN NOISE DATA FOR OFFSET DETERMINATION HAS
ALL BEEN COLLECTED*/
/*OFF WILL BE THE SUM OF ALL THE NOISE PEAKS COLLECTED
DURING THE OFFSCAN AND K1 WILL BE HOW MANY PEAKS
WENT INTO THE TOTAL*/
/*DO LOOP TO FIND TOTAL OF NOISE PEAKS*/

/*RUNNING TOTAL IN POF*/*
/*RUNNING TOTAL OF NUMBER OF PEAKS GOING INTO TOTAL*/

/*IF ANY NOISE PEAKS FIND THE AVERAGE OF ALL */
/*IF NONE SET AVERAGE TO ZERO*/
/*DETERMINE NEXT OFFSET CORRECTION FOR LAST LAMP*/
/*OFFSET NOISE DATA HAS BEEN PROCESSED*/

```

```

PL/M-80 COMPILER   RBC DISPLAY - MAIN PROGRAM

194 1      END RBC$DISPLAY$PROGRAM;

MODULE INFORMATION:
CODE AREA SIZE      = 04B6H   1206D
VARIABLE AREA SIZE = 0555H   1365D
MAXIMUM STACK SIZE = 000AH   10D
223 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

```

PL/M-80 COMPILER RBC DISPLAY - ASCII

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE ASCII.MOD
OBJECT MODULE PLACED IN: I:\ASCII.OBJ
COMPILER INVOKED BY: PLM80 I:\ASCII.SRC TITLE('RBC DISPLAY - ASCII')

```

1  $PRINT(:LF:) DEBUG
   ASCII$MOD: DO:

/*THIS PROCEDURE TAKES THE POSITION OF THE PEAK AS SHOWN BY THE CURSOR IN
THE ONE SCAN MODE OR THE POSITION OF THE DETERMINED CBH BOTH OF WHICH ARE
IN 1/4 DEGREE INCREMENTS AND CONVERTS IT TO A HEIGHT IN FEET. THEN THE FEET
HEIGHT IS CONVERTED TO A DOT PATTERN ON THE DISPLAY WHICH WILL PRINT OUT
THE APPROPRIATE NUMBER. */

2  DECLARE DCL LITERALLY 'DECLARE':
3  DCL NO BYTE EXTERNAL:
4  DCL DIS(200) ADDRESS EXTERNAL:
5  DCL HEAD(118) BYTE DATA(3EH,41H,41H,41H,3EH,01H,21H,7FH,01H,01H,
   21H,43H,45H,49H,31H,22H,41H,49H,49H,36H,0CH,14H,24H,7FH,04H,72H,51H,51H,4EH,
   3EH,49H,49H,49H,06H,41H,42H,44H,48H,70H,36H,49H,49H,36H,30H,49H,44H,4CH,38H,
   7FH,20H,1CH,02H,7FH,00H,3EH,41H,41H,41H,3EH,00H,00H,00H,00H,00H,00H,0FFH,0FFH,
   7FH,48H,48H,48H,30H,00H,7FH,49H,41H,00H,1FH,24H,44H,24H,1FH,00H,7FH,08H,
   14H,22H,41H,0FFH,0FFH,3EH,41H,41H,41H,22H,00H,7FH,49H,49H,37H,00H,7FH,08H,08H,
   08H,7FH,00H,00H,00H,00H,00H,00H):

/*HEAD IS A FIVE BY SEVEN MATRIX WHICH WRITES OUT ON THE DISPLAY

0 1 2 3 4 5 6 7 8 9 N O P E A K C B H
DUMMY CHARACTERS OF OFFH ARE ADDED AFTER NO AND PEAK TO MAKE THE COUNT BETTER
WHEN USED IN A DO LOOP*/

6  HEIGHT: PROCEDURE(Y) ADDRESS EXTERNAL:
7  DCL Y ADDRESS:
8  END HEIGHT:

9  PUT: PROCEDURE(X,Y,Z):
10 DCL (I,K,X,Y,Z) BYTE, J ADDRESS:
11 K = 0:
12 Z = Z + 5:

/*K IS A DUMMY VARIABLE TO COUNT THRU HEAD THE CORRECT NUMBER*/
/*Z IS THE NUMBER TO BE CONVERTED INTO A 5 X 7 NUMBER TO FIND
CORRECT PLACE MULTIPLY BY FIVE. ALSO AT THE END IS THE WORDS NO,
PEAK, AND CBH WITH OFFH FILLERS*/
/*DO LOOP TO PLACE THE BIT PATTERNS OF THE NUMBERS OR LETTERS*/
/*HEAD HAS THE CORRECT BIT PATTERNS FOR CONVERTING*/
/*THE NUMBERS OR LETTERS ARE PLACED IN THE TOP HALF OF DISPLAY*/
/*INCREMENT THE COUNTER FOR PROCEEDING THROUGH HEAD*/

13 DO I = X TO Y:
14 J = HEAD(Z*K):
15 DIS(I) = DIS(I) + SHL(J,8):
16 K = K + 1:
17 END:
18 END PUT:

19 ASCII: PROCEDURE(X) PUBLIC:
20 DCL (FIRST,J,P,SECOND) BYTE, (X,Y) ADDRESS:
21 IF X < 351 THEN DO:
22 NO = 0:
23 Y = HEIGHT(X):
24 /*PEAK OR CBH FOUND TO BE CONVERTED*/
   /*DUMMY VARIABLE FOR "NO" NOT PRINTED THIS SCAN*/
   /*CONVERTING NUMBER OF 1/4 DEGREES TO NUMBER OF FEET*/

```

PL/M-80 COMPILER RBC DISPLAY - ASCII

```

25 3 FIRST,SECOND = 0;
26 3 P = (Y/1000);
27 3 IF P > 0 THEN DO;
29 4 CALL PUT(169,173,P);
30 4 END;
31 4 ELSE FIRST = 1;
32 3 Y = Y - (P*1000);
33 3 P = (Y/100);
34 3 IF P > 0 OR FIRST = 0 THEN DO;
36 4 CALL PUT(175,179,P);
37 4 END;
38 3 ELSE SECOND = 1;
39 3 Y = Y - (P*100);
40 3 P = (Y/10);
41 3 IF P > 0 OR FIRST = 0 OR SECOND = 0 THEN DO;
43 4 CALL PUT(181,185,P);
44 4 END;
45 3 P = Y - (P*10);
46 3 CALL PUT(187,191,P);
47 3 END;
48 3 ELSE DO;
49 3 NO = 1;
50 3 DO J = 151 TO 168;
51 4 DIS(J) = DIS(J) AND 00FFH;
52 4 END;
53 3 CALL PUT(151,168,10);
54 3 IF X = 384 THEN CALL PUT(169,191,14);
55 3 ELSE CALL PUT(169,191,19);
56 3 END;
57 3 END ASCII;
58 2 END ASCII#MOD;
59 1

```

```

/*DUMMY VARIABLES TO SUPPRESS LED ZEROS*/
/*GET THOUSAND DIGIT*/
/*IF NON-ZERO PUT VALUE FROM HEAD (BIT PATTERN FOR THOUSAND
DIGIT) INTO ARRAY DIS*/

/*IF THOUSAND DIGIT ZERO SUPPRESS AND SET DUMMY VARIABLE TO 1*/
/*SUBTRACT THOUSAND DIGIT FROM ORIGINAL NUMBER */
/*GET HUNDRED DIGIT*/
/*IF HUNDRED DIGIT NONZERO OR THOUSAND DIGIT WAS NONZERO,
THEN GET BIT PATTERN OF NUMBER*/

/*IF THOUSAND AND HUNDRED DIGIT BOTH ZERO, SUPPRESS BOTH AND
SET DUMMY VARIABLE TO 1*/
/*SUBTRACT HUNDRED DIGIT OUT OF ORIGINAL NUMBER*/
/*GET TENS DIGIT*/
/*IF ANY ONE OF THE THOUSANDS, HUNDREDS, OR TENS
DIGITS ARE NONZERO THEN PLACE TENS DIGIT IN ARRAY*/

/*SUBTRACT OUT TENS DIGIT FROM REMAINING NUMBER TO GET ONES DIGIT*/
/* PLACE ONES DIGIT CONVERTED IN ARRAY REGARDLESS OF PAST OR
PRESENT DIGITS*/

/*NO PEAK OR NO CBH WAS FOUND SO PRINT LETTERS IN ARRAY*/
/*DUMMY VARIABLE FOR "NO" PRINTED THIS SCAN*/
/*DO LOOP TO CLEAR TOP HALF OF ARRAY TO PLACE NO CBH OR PEAK*/
/*ZERO OUT TOP HALF OF ARRAY BUT LEAVE BOTTOM HALF*/

/*PLACE "NO " IN ARRAY */
/*PUT PEAK INTO DIS ARRAY ONE SCAN MODE*/
/*PUT CBH INTO DIS ARRAY 5 SCAN MODE*/

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0245H
VARIABLE AREA SIZE = 000FH
MAXIMUM STACK SIZE = 0006H
37 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - CHECK1

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE CHECK1MOD
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 F1:CHECK1.SAV TITLE('RBC DISPLAY - CHECK1') NOOBJECT

```

1      SPRINT(1LP1) DEBUG
2      CHECK1MOD: DO1
3
4      /*THIS PROCEDURE TAKES THE 352 CBH DATA POINTS AND ELIMINATES ANY SHARP NOISE
5      SPIKES, FILTERS AND AVERAGES THE POINTS (USING A 64 PART FILTER: 1 FOR I +- 3
6      .6 FOR I +-2, .15 FOR I +-1, AND 20 FOR CENTER VALUE), AND THEN FINDS THE
7      MAXIMUM AND SECONDARY MAXIMUM PEAK FOR EACH SCAN (PEAK IS DEFINED AS FOUR
8      INCREASING VALUES AND TWO DECREASING VALUES). THEN STORES HEIGHT IN DEGREES
9      OF MAXIMUM AND SECONDARY MAXIMUM PEAK IN HEIT. */
10
11     DECLARE A(361) BYTE EXTERNAL, HEIT(20) ADDRESS EXTERNAL
12     DECLARE PEAK(80) BYTE EXTERNAL, PULSE(80) ADDRESS EXTERNAL, RCT(2) BYTE EXTERNAL
13
14     CHECK1: PROCEDURE(NO2) PUBLIC
15
16     DECLARE (DOWN,IX,J,NO2,Q,SVE1,UP) BYTE
17     DECLARE (I,P) ADDRESS
18
19     DO P = 21 TO 351
20     I = (A(P-1) + A(P+1))/2
21     IF DOUBLE(A(P)) >= 3 * I THEN A(P) = I
22     /*DO LOOP WHICH CHECKS FOR LARGE NOISE SPIKES*/
23     /*FIND AVERAGE OF TWO ADJACENT POINTS*/
24     /*IF CENTER POINT GREATER THAN 3 TIMES THE AVERAGE
25     /*THEN NOISE AND FILL WITH AVERAGE OF ADJACENT POINTS*/
26     END
27
28     DO I = 24 TO 348
29     P = A(I-3) + 6*(DOUBLE(A(I-2)) + A(I+2)) + 15*(DOUBLE(A(I-1))
30     + A(I+1)) + 20*A(I) + A(I+3)
31     A(I) = P/64
32     /*DO LOOP WHICH FILTERS AND AVERAGES*/
33     /*FILTER AND AVERAGE BY WEIGHTS 1.6,15,20,15,6,IFOR I-3,I-2,....,I+3*/
34     /*DIVIDE SUM BY THE TOTAL OF THE HEIGHTS*/
35     DO I = 0 TO 79
36     PEAK(I),PULSE(I) = 0
37     END
38     DOWN,IX,UP = 0
39
40     SVE1 = 0
41     DO I = 24 TO 347
42     IF IX < 79 THEN DO
43     IF A(I+1) > A(I) THEN DO
44     IF DOWN = 1 THEN DO
45
46     IF SVE1 > A(I+1) THEN DO
47     IF UP > 2 THEN DO
48
49     I = I + 1
50     IX = IX + 1
51     END
52     DOWN,UP = 0
53     END
54
55     /*DOWN = NUMBER OF DECREASING INTERVALS AFTER PEAK, IX = COUNTER
56     FOR NUMBER OF PEAKS UP = NUMBER OF INCREASING INTERVALS*/
57     /*SET MAXIMUM IN PEAK SO FAR TO ZERO*/
58     /*DO LOOP TO GO THROUGH CBH DATA POINTS TO FIND PEAKS*/
59     /*LIMIT NUMBER OF PEAKS TO 80*/
60     /*CHECK FOR INCREASING INTERVALS*/
61     /*LAST POINT WAS A DECREASE, BUT THE POINT BEFORE THAT WAS NOT.
62     NEEDED TWO DECREASING TO GET A PEAK, SO NEED TO CHECK IF
63     SECOND PREVIOUS POINT WAS LESS THAN PRESENT*/
64     /*SVE1 IS THE SECOND PREVIOUS POINT, IF SVE1 IS LARGER THAN A(I+1)
65     WE HAVE TWO DECREASING VALUES*/
66     /*CHECK NOW TO SEE IF 3 INCREASING INTERVALS OCCURRED BEFORE 2 DECREASING
67     THIS DETERMINES A PEAK, IF PEAK FOUND DO FOLLOWING*/
68     /*INCREMENT I SO THAT POINT WILL NOT BE COUNTED AGAIN*/
69     /*INCREMENT PEAK COUNTER*/
70
71     /*EITHER A PEAK OR NOT, TWO DECREASING VALUES FOUND SO RESET DOWN AND UP*/

```

PL/M-80 COMPILER RBC DISPLAY - CHECK1

```

37 6 ELSE DO:
38 7 DOWN = 0;
39 7 UP = UP + 1;
40 7 SVE1 = A(I+1);
41 7 IF UP > 2 THEN DO:
42 8 PEAK(I) = A(I+1);
43 8 PULSE(I) = 1 + 1;
44 8 END; END;
45 8 ELSE DO:
46 8 UP = UP + 1;
47 8 SVE1 = A(I+1);
48 8 IF UP > 2 THEN DO:
49 9 PEAK(I) = A(I+1);
50 9 PULSE(I) = 1 + 1;
51 9 END; END;
52 9 ELSE DO:
53 9 IF DOWN = 0 THEN DOWN = 1;
54 9 IF UP > 2 THEN DO:
55 10 PEAK(I) = A(I+1);
56 10 PULSE(I) = 1 + 1;
57 10 END; END;
58 10 ELSE DO:
59 10 IF DOWN = 0 THEN DOWN = 1;
60 10 IF UP > 2 THEN DO:
61 11 PEAK(I) = A(I+1);
62 11 PULSE(I) = 1 + 1;
63 11 END; END;
64 11 ELSE DO:
65 11 IF UP > 2 THEN DO:
66 12 PEAK(I) = A(I+1);
67 12 PULSE(I) = 1 + 1;
68 12 END; END;
69 12 ELSE DO:
70 12 IF UP > 2 THEN DO:
71 13 PEAK(I) = A(I+1);
72 13 PULSE(I) = 1 + 1;
73 13 END; END;
74 13 ELSE DO:
75 13 IF UP > 2 THEN DO:
76 14 PEAK(I) = A(I+1);
77 14 PULSE(I) = 1 + 1;
78 14 END; END;
79 14 ELSE DO:
80 14 IF UP > 2 THEN DO:
81 15 PEAK(I) = A(I+1);
82 15 PULSE(I) = 1 + 1;
83 15 END; END;
84 15 ELSE DO:
85 15 IF UP > 2 THEN DO:
86 16 PEAK(I) = A(I+1);
87 16 PULSE(I) = 1 + 1;
88 16 END; END;
89 16 ELSE DO:
90 16 IF UP > 2 THEN DO:
91 17 PEAK(I) = A(I+1);
92 17 PULSE(I) = 1 + 1;
93 17 END; END;
94 17 ELSE DO:
95 17 IF UP > 2 THEN DO:
96 18 PEAK(I) = A(I+1);
97 18 PULSE(I) = 1 + 1;
98 18 END; END;
99 18 ELSE DO:
100 18 IF UP > 2 THEN DO:
101 19 PEAK(I) = A(I+1);
102 19 PULSE(I) = 1 + 1;
103 19 END; END;
104 19 ELSE DO:
105 19 IF UP > 2 THEN DO:
106 20 PEAK(I) = A(I+1);
107 20 PULSE(I) = 1 + 1;
108 20 END; END;
109 20 ELSE DO:
110 20 IF UP > 2 THEN DO:
111 21 PEAK(I) = A(I+1);
112 21 PULSE(I) = 1 + 1;
113 21 END; END;
114 21 ELSE DO:
115 21 IF UP > 2 THEN DO:
116 22 PEAK(I) = A(I+1);
117 22 PULSE(I) = 1 + 1;
118 22 END; END;
119 22 ELSE DO:
120 22 IF UP > 2 THEN DO:
121 23 PEAK(I) = A(I+1);
122 23 PULSE(I) = 1 + 1;
123 23 END; END;
124 23 ELSE DO:
125 23 IF UP > 2 THEN DO:
126 24 PEAK(I) = A(I+1);
127 24 PULSE(I) = 1 + 1;
128 24 END; END;
129 24 ELSE DO:
130 24 IF UP > 2 THEN DO:
131 25 PEAK(I) = A(I+1);
132 25 PULSE(I) = 1 + 1;
133 25 END; END;
134 25 ELSE DO:
135 25 IF UP > 2 THEN DO:
136 26 PEAK(I) = A(I+1);
137 26 PULSE(I) = 1 + 1;
138 26 END; END;
139 26 ELSE DO:
140 26 IF UP > 2 THEN DO:
141 27 PEAK(I) = A(I+1);
142 27 PULSE(I) = 1 + 1;
143 27 END; END;
144 27 ELSE DO:
145 27 IF UP > 2 THEN DO:
146 28 PEAK(I) = A(I+1);
147 28 PULSE(I) = 1 + 1;
148 28 END; END;
149 28 ELSE DO:
150 28 IF UP > 2 THEN DO:
151 29 PEAK(I) = A(I+1);
152 29 PULSE(I) = 1 + 1;
153 29 END; END;
154 29 ELSE DO:
155 29 IF UP > 2 THEN DO:
156 30 PEAK(I) = A(I+1);
157 30 PULSE(I) = 1 + 1;
158 30 END; END;
159 30 ELSE DO:
160 30 IF UP > 2 THEN DO:
161 31 PEAK(I) = A(I+1);
162 31 PULSE(I) = 1 + 1;
163 31 END; END;
164 31 ELSE DO:
165 31 IF UP > 2 THEN DO:
166 32 PEAK(I) = A(I+1);
167 32 PULSE(I) = 1 + 1;
168 32 END; END;
169 32 ELSE DO:
170 32 IF UP > 2 THEN DO:
171 33 PEAK(I) = A(I+1);
172 33 PULSE(I) = 1 + 1;
173 33 END; END;
174 33 ELSE DO:
175 33 IF UP > 2 THEN DO:
176 34 PEAK(I) = A(I+1);
177 34 PULSE(I) = 1 + 1;
178 34 END; END;
179 34 ELSE DO:
180 34 IF UP > 2 THEN DO:
181 35 PEAK(I) = A(I+1);
182 35 PULSE(I) = 1 + 1;
183 35 END; END;
184 35 ELSE DO:
185 35 IF UP > 2 THEN DO:
186 36 PEAK(I) = A(I+1);
187 36 PULSE(I) = 1 + 1;
188 36 END; END;
189 36 ELSE DO:
190 36 IF UP > 2 THEN DO:
191 37 PEAK(I) = A(I+1);
192 37 PULSE(I) = 1 + 1;
193 37 END; END;
194 37 ELSE DO:
195 37 IF UP > 2 THEN DO:
196 38 PEAK(I) = A(I+1);
197 38 PULSE(I) = 1 + 1;
198 38 END; END;
199 38 ELSE DO:
200 38 IF UP > 2 THEN DO:
201 39 PEAK(I) = A(I+1);
202 39 PULSE(I) = 1 + 1;
203 39 END; END;
204 39 ELSE DO:
205 39 IF UP > 2 THEN DO:
206 40 PEAK(I) = A(I+1);
207 40 PULSE(I) = 1 + 1;
208 40 END; END;
209 40 ELSE DO:
210 40 IF UP > 2 THEN DO:
211 41 PEAK(I) = A(I+1);
212 41 PULSE(I) = 1 + 1;
213 41 END; END;
214 41 ELSE DO:
215 41 IF UP > 2 THEN DO:
216 42 PEAK(I) = A(I+1);
217 42 PULSE(I) = 1 + 1;
218 42 END; END;
219 42 ELSE DO:
220 42 IF UP > 2 THEN DO:
221 43 PEAK(I) = A(I+1);
222 43 PULSE(I) = 1 + 1;
223 43 END; END;
224 43 ELSE DO:
225 43 IF UP > 2 THEN DO:
226 44 PEAK(I) = A(I+1);
227 44 PULSE(I) = 1 + 1;
228 44 END; END;
229 44 ELSE DO:
230 44 IF UP > 2 THEN DO:
231 45 PEAK(I) = A(I+1);
232 45 PULSE(I) = 1 + 1;
233 45 END; END;
234 45 ELSE DO:
235 45 IF UP > 2 THEN DO:
236 46 PEAK(I) = A(I+1);
237 46 PULSE(I) = 1 + 1;
238 46 END; END;
239 46 ELSE DO:
240 46 IF UP > 2 THEN DO:
241 47 PEAK(I) = A(I+1);
242 47 PULSE(I) = 1 + 1;
243 47 END; END;
244 47 ELSE DO:
245 47 IF UP > 2 THEN DO:
246 48 PEAK(I) = A(I+1);
247 48 PULSE(I) = 1 + 1;
248 48 END; END;
249 48 ELSE DO:
250 48 IF UP > 2 THEN DO:
251 49 PEAK(I) = A(I+1);
252 49 PULSE(I) = 1 + 1;
253 49 END; END;
254 49 ELSE DO:
255 49 IF UP > 2 THEN DO:
256 50 PEAK(I) = A(I+1);
257 50 PULSE(I) = 1 + 1;
258 50 END; END;
259 50 ELSE DO:
260 50 IF UP > 2 THEN DO:
261 51 PEAK(I) = A(I+1);
262 51 PULSE(I) = 1 + 1;
263 51 END; END;
264 51 ELSE DO:
265 51 IF UP > 2 THEN DO:
266 52 PEAK(I) = A(I+1);
267 52 PULSE(I) = 1 + 1;
268 52 END; END;
269 52 ELSE DO:
270 52 IF UP > 2 THEN DO:
271 53 PEAK(I) = A(I+1);
272 53 PULSE(I) = 1 + 1;
273 53 END; END;
274 53 ELSE DO:
275 53 IF UP > 2 THEN DO:
276 54 PEAK(I) = A(I+1);
277 54 PULSE(I) = 1 + 1;
278 54 END; END;
279 54 ELSE DO:
280 54 IF UP > 2 THEN DO:
281 55 PEAK(I) = A(I+1);
282 55 PULSE(I) = 1 + 1;
283 55 END; END;
284 55 ELSE DO:
285 55 IF UP > 2 THEN DO:
286 56 PEAK(I) = A(I+1);
287 56 PULSE(I) = 1 + 1;
288 56 END; END;
289 56 ELSE DO:
290 56 IF UP > 2 THEN DO:
291 57 PEAK(I) = A(I+1);
292 57 PULSE(I) = 1 + 1;
293 57 END; END;
294 57 ELSE DO:
295 57 IF UP > 2 THEN DO:
296 58 PEAK(I) = A(I+1);
297 58 PULSE(I) = 1 + 1;
298 58 END; END;
299 58 ELSE DO:
300 58 IF UP > 2 THEN DO:
301 59 PEAK(I) = A(I+1);
302 59 PULSE(I) = 1 + 1;
303 59 END; END;
304 59 ELSE DO:
305 59 IF UP > 2 THEN DO:
306 60 PEAK(I) = A(I+1);
307 60 PULSE(I) = 1 + 1;
308 60 END; END;
309 60 ELSE DO:
310 60 IF UP > 2 THEN DO:
311 61 PEAK(I) = A(I+1);
312 61 PULSE(I) = 1 + 1;
313 61 END; END;
314 61 ELSE DO:
315 61 IF UP > 2 THEN DO:
316 62 PEAK(I) = A(I+1);
317 62 PULSE(I) = 1 + 1;
318 62 END; END;
319 62 ELSE DO:
320 62 IF UP > 2 THEN DO:
321 63 PEAK(I) = A(I+1);
322 63 PULSE(I) = 1 + 1;
323 63 END; END;
324 63 ELSE DO:
325 63 IF UP > 2 THEN DO:
326 64 PEAK(I) = A(I+1);
327 64 PULSE(I) = 1 + 1;
328 64 END; END;
329 64 ELSE DO:
330 64 IF UP > 2 THEN DO:
331 65 PEAK(I) = A(I+1);
332 65 PULSE(I) = 1 + 1;
333 65 END; END;
334 65 ELSE DO:
335 65 IF UP > 2 THEN DO:
336 66 PEAK(I) = A(I+1);
337 66 PULSE(I) = 1 + 1;
338 66 END; END;
339 66 ELSE DO:
340 66 IF UP > 2 THEN DO:
341 67 PEAK(I) = A(I+1);
342 67 PULSE(I) = 1 + 1;
343 67 END; END;
344 67 ELSE DO:
345 67 IF UP > 2 THEN DO:
346 68 PEAK(I) = A(I+1);
347 68 PULSE(I) = 1 + 1;
348 68 END; END;
349 68 ELSE DO:
350 68 IF UP > 2 THEN DO:
351 69 PEAK(I) = A(I+1);
352 69 PULSE(I) = 1 + 1;
353 69 END; END;
354 69 ELSE DO:
355 69 IF UP > 2 THEN DO:
356 70 PEAK(I) = A(I+1);
357 70 PULSE(I) = 1 + 1;
358 70 END; END;
359 70 ELSE DO:
360 70 IF UP > 2 THEN DO:
361 71 PEAK(I) = A(I+1);
362 71 PULSE(I) = 1 + 1;
363 71 END; END;
364 71 ELSE DO:
365 71 IF UP > 2 THEN DO:
366 72 PEAK(I) = A(I+1);
367 72 PULSE(I) = 1 + 1;
368 72 END; END;
369 72 ELSE DO:
370 72 IF UP > 2 THEN DO:
371 73 PEAK(I) = A(I+1);
372 73 PULSE(I) = 1 + 1;
373 73 END; END;
374 73 ELSE DO:
375 73 IF UP > 2 THEN DO:
376 74 PEAK(I) = A(I+1);
377 74 PULSE(I) = 1 + 1;
378 74 END; END;
379 74 ELSE DO:
380 74 IF UP > 2 THEN DO:
381 75 PEAK(I) = A(I+1);
382 75 PULSE(I) = 1 + 1;
383 75 END; END;
384 75 ELSE DO:
385 75 IF UP > 2 THEN DO:
386 76 PEAK(I) = A(I+1);
387 76 PULSE(I) = 1 + 1;
388 76 END; END;
389 76 ELSE DO:
390 76 IF UP > 2 THEN DO:
391 77 PEAK(I) = A(I+1);
392 77 PULSE(I) = 1 + 1;
393 77 END; END;
394 77 ELSE DO:
395 77 IF UP > 2 THEN DO:
396 78 PEAK(I) = A(I+1);
397 78 PULSE(I) = 1 + 1;
398 78 END; END;
399 78 ELSE DO:
400 78 IF UP > 2 THEN DO:
401 79 PEAK(I) = A(I+1);
402 79 PULSE(I) = 1 + 1;
403 79 END; END;
404 79 ELSE DO:
405 79 IF UP > 2 THEN DO:
406 80 PEAK(I) = A(I+1);
407 80 PULSE(I) = 1 + 1;
408 80 END; END;
409 80 ELSE DO:
410 80 IF UP > 2 THEN DO:
411 81 PEAK(I) = A(I+1);
412 81 PULSE(I) = 1 + 1;
413 81 END; END;
414 81 ELSE DO:
415 81 IF UP > 2 THEN DO:
416 82 PEAK(I) = A(I+1);
417 82 PULSE(I) = 1 + 1;
418 82 END; END;
419 82 ELSE DO:
420 82 IF UP > 2 THEN DO:
421 83 PEAK(I) = A(I+1);
422 83 PULSE(I) = 1 + 1;
423 83 END; END;
424 83 ELSE DO:
425 83 IF UP > 2 THEN DO:
426 84 PEAK(I) = A(I+1);
427 84 PULSE(I) = 1 + 1;
428 84 END; END;
429 84 ELSE DO:
430 84 IF UP > 2 THEN DO:
431 85 PEAK(I) = A(I+1);
432 85 PULSE(I) = 1 + 1;
433 85 END; END;
434 85 ELSE DO:
435 85 IF UP > 2 THEN DO:
436 86 PEAK(I) = A(I+1);
437 86 PULSE(I) = 1 + 1;
438 86 END; END;
439 86 ELSE DO:
440 86 IF UP > 2 THEN DO:
441 87 PEAK(I) = A(I+1);
442 87 PULSE(I) = 1 + 1;
443 87 END; END;
444 87 ELSE DO:
445 87 IF UP > 2 THEN DO:
446 88 PEAK(I) = A(I+1);
447 88 PULSE(I) = 1 + 1;
448 88 END; END;
449 88 ELSE DO:
450 88 IF UP > 2 THEN DO:
451 89 PEAK(I) = A(I+1);
452 89 PULSE(I) = 1 + 1;
453 89 END; END;
454 89 ELSE DO:
455 89 IF UP > 2 THEN DO:
456 90 PEAK(I) = A(I+1);
457 90 PULSE(I) = 1 + 1;
458 90 END; END;
459 90 ELSE DO:
460 90 IF UP > 2 THEN DO:
461 91 PEAK(I) = A(I+1);
462 91 PULSE(I) = 1 + 1;
463 91 END; END;
464 91 ELSE DO:
465 91 IF UP > 2 THEN DO:
466 92 PEAK(I) = A(I+1);
467 92 PULSE(I) = 1 + 1;
468 92 END; END;
469 92 ELSE DO:
470 92 IF UP > 2 THEN DO:
471 93 PEAK(I) = A(I+1);
472 93 PULSE(I) = 1 + 1;
473 93 END; END;
474 93 ELSE DO:
475 93 IF UP > 2 THEN DO:
476 94 PEAK(I) = A(I+1);
477 94 PULSE(I) = 1 + 1;
478 94 END; END;
479 94 ELSE DO:
480 94 IF UP > 2 THEN DO:
481 95 PEAK(I) = A(I+1);
482 95 PULSE(I) = 1 + 1;
483 95 END; END;
484 95 ELSE DO:
485 95 IF UP > 2 THEN DO:
486 96 PEAK(I) = A(I+1);
487 96 PULSE(I) = 1 + 1;
488 96 END; END;
489 96 ELSE DO:
490 96 IF UP > 2 THEN DO:
491 97 PEAK(I) = A(I+1);
492 97 PULSE(I) = 1 + 1;
493 97 END; END;
494 97 ELSE DO:
495 97 IF UP > 2 THEN DO:
496 98 PEAK(I) = A(I+1);
497 98 PULSE(I) = 1 + 1;
498 98 END; END;
499 98 ELSE DO:
500 98 IF UP > 2 THEN DO:
501 99 PEAK(I) = A(I+1);
502 99 PULSE(I) = 1 + 1;
503 99 END; END;
504 99 ELSE DO:
505 99 IF UP > 2 THEN DO:
506 100 PEAK(I) = A(I+1);
507 100 PULSE(I) = 1 + 1;
508 100 END; END;
509 100 ELSE DO:
510 100 IF UP > 2 THEN DO:
511 101 PEAK(I) = A(I+1);
512 101 PULSE(I) = 1 + 1;
513 101 END; END;
514 101 ELSE DO:
515 101 IF UP > 2 THEN DO:
516 102 PEAK(I) = A(I+1);
517 102 PULSE(I) = 1 + 1;
518 102 END; END;
519 102 ELSE DO:
520 102 IF UP > 2 THEN DO:
521 103 PEAK(I) = A(I+1);
522 103 PULSE(I) = 1 + 1;
523 103 END; END;
524 103 ELSE DO:
525 103 IF UP > 2 THEN DO:
526 104 PEAK(I) = A(I+1);
527 104 PULSE(I) = 1 + 1;
528 104 END; END;
529 104 ELSE DO:
530 104 IF UP > 2 THEN DO:
531 105 PEAK(I) = A(I+1);
532 105 PULSE(I) = 1 + 1;
533 105 END; END;
534 105 ELSE DO:
535 105 IF UP > 2 THEN DO:
536 106 PEAK(I) = A(I+1);
537 106 PULSE(I) = 1 + 1;
538 106 END; END;
539 106 ELSE DO:
540 106 IF UP > 2 THEN DO:
541 107 PEAK(I) = A(I+1);
542 107 PULSE(I) = 1 + 1;
543 107 END; END;
544 107 ELSE DO:
545 107 IF UP > 2 THEN DO:
546 108 PEAK(I) = A(I+1);
547 108 PULSE(I) = 1 + 1;
548 108 END; END;
549 108 ELSE DO:
550 108 IF UP > 2 THEN DO:
551 109 PEAK(I) = A(I+1);
552 109 PULSE(I) = 1 + 1;
553 109 END; END;
554 109 ELSE DO:
555 109 IF UP > 2 THEN DO:
556 110 PEAK(I) = A(I+1);
557 110 PULSE(I) = 1 + 1;
558 110 END; END;
559 110 ELSE DO:
560 110 IF UP > 2 THEN DO:
561 111 PEAK(I) = A(I+1);
562 111 PULSE(I) = 1 + 1;
563 111 END; END;
564 111 ELSE DO:
565 111 IF UP > 2 THEN DO:
566 112 PEAK(I) = A(I+1);
567 112 PULSE(I) = 1 + 1;
568 112 END; END;
569 112 ELSE DO:
570 112 IF UP > 2 THEN DO:
571 113 PEAK(I) = A(I+1);
572 113 PULSE(I) = 1 + 1;
573 113 END; END;
574 113 ELSE DO:
575 113 IF UP > 2 THEN DO:
576 114 PEAK(I) = A(I+1);
577 114 PULSE(I) = 1 + 1;
578 114 END; END;
579 114 ELSE DO:
580 114 IF UP > 2 THEN DO:
581 115 PEAK(I) = A(I+1);
582 115 PULSE(I) = 1 + 1;
583 115 END; END;
584 115 ELSE DO:
585 115 IF UP > 2 THEN DO:
586 116 PEAK(I) = A(I+1);
587 116 PULSE(I) = 1 + 1;
588 116 END; END;
589 116 ELSE DO:
590 116 IF UP > 2 THEN DO:
591 117 PEAK(I) = A(I+1);
592 117 PULSE(I) = 1 + 1;
593 117 END; END;
594 117 ELSE DO:
595 117 IF UP > 2 THEN DO:
596 118 PEAK(I) = A(I+1);
597 118 PULSE(I) = 1 + 1;
598 118 END; END;
599 118 ELSE DO:
600 118 IF UP > 2 THEN DO:
601 119 PEAK(I) = A(I+1);
602 119 PULSE(I) = 1 + 1;
603 119 END; END;
604 119 ELSE DO:
605 119 IF UP > 2 THEN DO:
606 120 PEAK(I) = A(I+1);
607 120 PULSE(I) = 1 + 1;
608 120 END; END;
609 120 ELSE DO:
610 120 IF UP > 2 THEN DO:
611 121 PEAK(I) = A(I+1);
612 121 PULSE(I) = 1 + 1;
613 121 END; END;
614 121 ELSE DO:
615 121 IF UP > 2 THEN DO:
616 122 PEAK(I) = A(I+1);
617 122 PULSE(I) = 1 + 1;
618 122 END; END;
619 122 ELSE DO:
620 122 IF UP > 2 THEN DO:
621 123 PEAK(I) = A(I+1);
622 123 PULSE(I) = 1 + 1;
623 123 END; END;
624 123 ELSE DO:
625 123 IF UP > 2 THEN DO:
626 124 PEAK(I) = A(I+1);
627 124 PULSE(I) = 1 + 1;
628 124 END; END;
629 124 ELSE DO:
630 124 IF UP > 2 THEN DO:
631 125 PEAK(I) = A(I+1);
632 125 PULSE(I) = 1 + 1;
633 125 END; END;
634 125 ELSE DO:
635 125 IF UP > 2 THEN DO:
636 126 PEAK(I) = A(I+1);
637 126 PULSE(I) = 1 + 1;
638 126 END; END;
639 126 ELSE DO:
640 126 IF UP > 2 THEN DO:
641 127 PEAK(I) = A(I+1);
642 127 PULSE(I) = 1 + 1;
643 127 END; END;
644 127 ELSE DO:
645 127 IF UP > 2 THEN DO:
646 128 PEAK(I) = A(I+1);
647 128 PULSE(I) = 1 + 1;
648 128 END; END;
649 128 ELSE DO:
650 128 IF UP > 2 THEN DO:
651 129 PEAK(I) = A(I+1);
652 129 PULSE(I) = 1 + 1;
653 129 END; END;
654 129 ELSE DO:
655 129 IF UP > 2 THEN DO:
656 130 PEAK(I) = A(I+1);
657 130 PULSE(I) = 1 + 1;
658 130 END; END;
659 130 ELSE DO:
660 130 IF UP > 2 THEN DO:
661 131 PEAK(I) = A(I+1);
662 131 PULSE(I) = 1 + 1;
663 131 END; END;
664 131 ELSE DO:
665 131 IF UP > 2 THEN DO:
666 132 PEAK(I) = A(I+1);
667 132 PULSE(I) = 1 + 1;
668 132 END; END;
669 132 ELSE DO:
670 132 IF UP > 2 THEN DO:
671 133 PEAK(I) = A(I+1);
672 133 PULSE(I) = 1 + 1;
673 133 END; END;
674 133 ELSE DO:
675 133 IF UP > 2 THEN DO:
676 134 PEAK(I) = A(I+1);
677 134 PULSE(I) = 1 + 1;
678 134 END; END;
679 134 ELSE DO:
680 134 IF UP > 2 THEN DO:
681 135 PEAK(I) = A(I+1);
682 135 PULSE(I) = 1 + 1;
683 135 END; END;
684 135 ELSE DO:
685 135 IF UP > 2 THEN DO:
686 136 PEAK(I) = A(I+1);
687 136 PULSE(I) = 1 + 1;
688 136 END; END;
689 136 ELSE DO:
690 136 IF UP > 2 THEN DO:
691 137 PEAK(I) = A(I+1);
692 137 PULSE(I) = 1 + 1;
693 137 END; END;
694 137 ELSE DO:
695 137 IF UP > 2 THEN DO:
696 138 PEAK(I) = A(I+1);
697 138 PULSE(I) = 1 + 1;
698 138 END; END;
699 138 ELSE DO:
700 138 IF UP > 2 THEN DO:
701 139 PEAK(I) = A(I+1);
702 139 PULSE(I) = 1 + 1;
703 139 END; END;
704 139 ELSE DO:
705 139 IF UP > 2 THEN DO:
706 140 PEAK(I) = A(I+1);
707 140 PULSE(I) = 1 + 1;
708 140 END; END;
709 140 ELSE DO:
710 140 IF UP > 2 THEN DO:
711 141 PEAK(I) = A(I+1);
712 141 PULSE(I) = 1 + 1;
713 141 END; END;
714 141 ELSE DO:
715 141 IF UP > 2 THEN DO:
716 142 PEAK(I) = A(I+1);
717 142 PULSE(I) = 1 + 1;
718 142 END; END;
719 142 ELSE DO:
720 142 IF UP > 2 THEN DO:
721 143 PEAK(I) = A(I+1);
722 143 PULSE(I) = 1 + 1;
723 143 END; END;
724 143 ELSE DO:
725 143 IF UP > 2 THEN DO:
726 144 PEAK(I) = A(I+1);
727 144 PULSE(I) = 1 + 1;
728 144 END; END;
729 144 ELSE DO:
730 144 IF UP > 2 THEN DO:
731 145 PEAK(I) = A(I+1);
732 145 PULSE(I) = 1 + 1;
733 145 END; END;
734 145 ELSE DO:
735 145 IF UP > 2 THEN DO:
736 146 PEAK(I) = A(I+1);
737 146 PULSE(I) = 1 + 1;
738 146 END; END;
739 146 ELSE DO:
740 146 IF UP > 2 THEN DO:
741 147 PEAK(I) = A(I+1);
742 147 PULSE(I) = 1 + 1;
743 147 END; END;
744 147 ELSE DO:
745 147 IF UP > 2 THEN DO:
746 148 PEAK(I) = A(I+1);
747 148 PULSE(I) = 1 + 1;
748 148 END; END;
749 148 ELSE DO:
750 148 IF UP > 2 THEN DO:
751 149 PEAK(I) = A(I+1);
752 149 PULSE(I) = 1 + 1;
753 149 END; END;
754 149 ELSE DO:
755 149 IF UP > 2 THEN DO:
756 150 PEAK(I) = A(I+1);
757 150 PULSE(I) = 1 + 1;
758 150 END; END;
759 150 ELSE DO:
760 150 IF UP > 2 THEN DO:
761 151 PEAK(I) = A(I+1);
762 151 PULSE(I) = 1 + 1;
763 151 END; END;
764 151 ELSE DO:
765 151 IF UP > 2 THEN DO:
766 152 PEAK(I) = A(I+1);
767 152 PULSE(I) = 1 + 1;
768 152 END; END;
769 152 ELSE DO:
770 152 IF UP > 2 THEN DO:
771 153 PEAK(I) = A(I+1);
772 153 PULSE(I) = 1 + 1;
773 153 END; END;
774 153 ELSE DO:
775 153 IF UP > 2 THEN DO:
776 154 PEAK(I) = A(I+1);
777 154 PULSE(I) = 1 + 1;
778 154 END; END;
779 154 ELSE DO:
780 154 IF UP > 2 THEN DO:
781 155 PEAK(I) = A(I+1);
782 155 PULSE(I) = 1 + 1;
783 155 END; END;
784 155 ELSE DO:
785 155 IF UP > 2 THEN DO:
786 156 PEAK(I) = A(I+1);
787 156 PULSE(I) = 1 + 1;
788 156 END; END;
789 156 ELSE DO:
790 156 IF UP > 2 THEN DO:
791 157 PEAK(I) = A(I+1);
792 157 PULSE(I) = 1 + 1;
793 157 END; END;
794 157 ELSE DO:
795 157 IF UP > 2 THEN DO:
796 158 PEAK(I) = A(I+1);
797 158 PULSE(I) = 1 + 1;
798 158 END; END;
799 158 ELSE DO:
800 158 IF UP > 2 THEN DO:
801 159 PEAK(I) = A(I+1);
802 159 PULSE(I) = 1 + 1;
803 159 END; END;
804 159 ELSE DO:
805 159 IF UP > 2 THEN DO:
806 160 PEAK(I) = A(I+1);
807 160 PULSE(I) = 1 + 1;
808 160 END; END;
809 160 ELSE DO:
810 160 IF UP > 2 THEN DO:
811 161 PEAK(I) = A(I+1);
812 161 PULSE(I) = 1 + 1;
813 161 END; END;
814 161 ELSE DO:
815 161 IF UP > 2 THEN DO:
816 162 PEAK(I) = A(I+1);
817 162 PULSE(I) = 1 + 1;
818 162 END; END;
819 162 ELSE DO:
820 162 IF UP > 2 THEN DO:
821 163 PEAK(I) = A(I+1);
822 163 PULSE(I) = 1 + 1;
823 163 END; END;
824 163 ELSE DO:
825 163 IF UP > 2 THEN DO:
826 164 PEAK(I) = A(I+1);
827 164 PULSE(I) = 1 + 1;
828 164 END; END;
829 164 ELSE DO:
830 164 IF UP > 2 THEN DO:
831 165 PEAK(I) = A(I+1);
832 165 PULSE(I) = 1 + 1;
833 165 END; END;
834 165 ELSE DO:
835 165 IF UP > 2 THEN DO:
836 166 PEAK(I) = A(I+1);
837 166 PULSE(I) = 1 + 1;
838 166 END; END;
839 166 ELSE DO:
840 166 IF UP > 2 THEN DO:
841 167 PEAK(I) = A(I+1);
842 167 PULSE(I) = 1 + 1;
843 167 END; END;
844 167 ELSE DO:
845 167 IF UP > 2 THEN DO:
846 168 PEAK(I) = A(I+1);
847 168 PULSE(I) = 1 + 1;
848 168 END; END;
849 168 ELSE DO:
850 168 IF UP > 2 THEN DO:
851 169 PEAK(I) = A(I+1);
852 169 PULSE(I) = 1 + 1;
853 169 END; END;
854 169 ELSE DO:
855 169 IF UP > 2 THEN DO:
856 170 PEAK(I) = A(I+1);
857 170 PULSE(I) = 1 + 1;
858 170 END; END;
859 170 ELSE DO:
860 170 IF UP >
```

PL/M-80 COMPILER RBC DISPLAY - DATA CHECK

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE DCMOD
 OBJECT MODULE PLACED IN F11DC.OBJ
 COMPILER INVOKED BY: PLM80 F11DC.SRC TITLE('RBC DISPLAY - DATA CHECK')

```

1      @PRINT(1:LP:)) DEBUG
2      DC#MOD: DO:
3
4      MTIME: PROCEDURE(X) EXTERNAL:
5      DECLARE X BYTE:
6      END MTIME:
7
8      DECLARE J BYTE EXTERNAL:
9
10     DATA#CHECK: PROCEDURE (P,Y,Z) PUBLIC REENTRANT:
11
12     /*PROCEDURE IS A FLAG CHECKER. THE PROGRAM REMAINS IN THIS PROCEDURE FOR 100MS
13     OR UNTIL A TRUE CONDITION EXISTS. J IS AN IDENTIFIER WHICH CAN BE CHECKED TO
14     DETERMINE THE REASON FOR EXIT ( J=0 TRUE CONDITION, J = 10 100MS EXIT)*/
15
16     DECLARE (Y,Z) BYTE, (I,P) ADDRESS:
17     J = 0:
18     DO I = 0 TO 847:
19     IF (MEMORY(P) AND Y) = Z THEN RETURN:
20     CALL MTIME(1):
21     END:
22     J = 10:
23     END DATA#CHECK:
24     END DC#MOD:
25
26     /*SET IDENTIFIER TO ZERO*/
27     /*DO LOOP TO CHECK CONDITION AND TO WASTE TIME IN BETWEEN*/
28     /*CHECK FOR CONDITION TRUE, IF TRUE RETURN*/
29     /*TOTAL WAIT OF 847 X 118MICRO SEC = APPROX 100M SEC*/
30     /*FINISHED DO LOOP WITHOUT A TRUE CONDITION, SET J*/

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0069H      105D
VARIABLE AREA SIZE  = 0000H       0D
MAXIMUM STACK SIZE  = 000AH      10D
24 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - DETOFF

ISIS-11 PL/M-80 V3.1 COMPILATION OF MODULE DETOFFMOD
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 IF1:DETOFF.SAV TITLE:RBC DISPLAY - DETOFF(') NOOBJECT

```

1      PRINT(1,LP1) DEBUG
2      DETOFFMOD: DO1
3      DECLARE (K6,K7,OFFS) BYTE EXTERNAL
4      DECLARE C(20) BYTE EXTERNAL, OCT(2) BYTE EXTERNAL, OFFSET(2) BYTE EXTERNAL
5      DETOFF: PROCEDURE PUBLIC
6      /*PROCEDURE TAKES THE ONE SCAN AVERAGE OF DATA COLLECTED DURING THE OFF SCAN
7      AND FINDS THE AVERAGE FOR THE LAST SCANS(UP TO A MAXIMUM OF 10). STORES IN OFFSET(0)
8      FOR LAMP 1 OR OFFSET(1) FOR LAMP 2*/
9      DECLARE (I,P) ADDRESS
10     P = OFFS
11     IF P > 3CH THEN P = 3CH
12     C(OCT(K6)+K7) = P
13     P = 0
14     DO I = 0 TO OCT(K6)
15     P = P + C(I+K7)
16     ENDI
17     IF OCT(K6) > 0 THEN P = P/OCT(K6)
18     OFFSET(K6) = SHR(P,2) + 2
19     IF OCT(K6) > 8 THEN DO1
20     OCT(K6) = 8
21     DO I = 0 TO 9
22     C(I+K7) = C(I+K7+1)
23     ENDI
24     OCT(K6) = OCT(K6) + 1
25     END DETOFF
26     END DETOFFMOD

```

/*OFFS IS AVERAGE NOISE FROM ONE SCAN*/
/*ONLY 4 SIO BITS FOR OFFSET, BITS 2,3,4,5*/
/*OFFSET IS CALCULATED AFTER ZERO DEGREE SWITCH FOR NEXT LIGHT, THEREFORE
K6 IS OPPOSITE OF K310 FOR LAMP 1 AND 1 FOR LAMP 2*/
/*K7 IS USED TO STORE THE TEN CURRENT OFFS IN C, THEREFORE K7 IS EITHER
0 OR 10 DEPENDING ON LAMP*/
/*P SET TO ZERO TO BE USED FOR TOTAL OF PAST OFFS UP TO 10*/
/*DO LOOP TO SUM ALL OFFS*/

/*AVERAGE OF PAST OFFS*/
/*ONLY 4 SIO BITS, BUT MUST BE IN BITS 0,1,2,3/
/*ADD 2 OFFSET TO CALCULATED VALUE TO REDUCE NOISE*/
/*LARGEST OFFSET CAN BE 15 4 BITS ON*/
/*IF ALREADY HAVE TEN OFFS MOVE THEN TENTH OUT AND
MAKE ROOM FOR THE NEXT ONE */

/*MOVE 1 TO 0, 2 TO 1 ETC*/
/*ADD ONE TO THE COUNT OF OFFS*/

MODULE INFORMATION:

CODE AREA SIZE = 0120H 298D
VARIABLE AREA SIZE = 0004H 4D
MAXIMUM STACK SIZE = 0004H 4D
38 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - FILL
 IS15-11 PL/M-80 V3.1 COMPILATION OF MODULE FILLMOD
 NO OBJECT MODULE REQUESTED
 COMPILER INVOKED BY: PLM80 IF1FILL.SAV TITLE('RBC DISPLAY - FILL') NOOBJECT

```

1  $PRINT('LP1') DEBUG
2  FILLMOD: DO1
3  ASCII: PROCEDURE(X) EXTERNAL1
4  DECLARE X ADDRESS1
5  END ASCII1
6  DECLARE (CURSOR,K3,SCAN) BYTE EXTERNAL1
7  DECLARE BEIL(2) ADDRESS EXTERNAL1 DIS(200) ADDRESS EXTERNAL1
8  FILL: PROCEDURE PUBLIC1
9  /*THIS PROCEDURE CAUSES THE CURSOR TO BE REVERSE VIDED, CLEANS OUT THE SPACE AT
10 THE END OF THE DISPLAY FOR THE NUMBERS TO BE DISPLAYED, CALLS ASCII WHICH CONVERTS
11 THE PEAK OR CBH TO A NUMBER AND PUTS INTO DISPLAY ARRAY, WRITES THE ARRAY INTO
12 THE MEMORY FOR THE DISPLAY UNIT, AND THEN GIVES THE SWITCH SIGNAL TO UPDATE
13 THE DISPLAY*/
14 DECLARE I ADDRESS1
15 /*DO LOOP TO CLEAN OUT MEMORY AT THE END WHERE NO DATA IS STORED*/
16 DO I = 176 TO 1921
17   DIS(I) = 00H1
18 END1
19 DIS(CURSOR) = NOT DIS(CURSOR)1
20 /*REVERSES THE BIT IN MEMORY WHERE THE CURSOR WILL BE STORED
21 CAUSING THE CURSOR TO BE REVERSE VIDEO*/
22 DO I = 169 TO 1751
23   /* CLEANS OUT THE TOP PART OF DISPLAY WHERE THE NUMBER WILL
24   BE STORED, ONLY THE TOP HALF BECAUSE DATA MIGHT ALSO BE HERE*/
25   DIS(I) = DIS(I) AND 00FFH1
26   SCAN = MEMORY(6601H) AND 03H1
27   IF SCAN > 0 THEN CALL ASCII(CURSOR*2)1 /*IF IN ONE SCAN OR MANUAL MODE PRINT NUMBER FOR CURSOR*/
28   ELSE CALL ASCII(BEIL(K3))1 /*IF IN FIVE SCAN MODE PRINT CBH HEIGHT*/
29   DO I = 0 TO 1921
30     MEMORY(6001H+I) = LOW(DIS(I))1 /*DO LOOP TO PUT DISPLAY UPDATE INTO DISPLAY MEMORY*/
31     MEMORY(6202H+I) = HIGH(DIS(I))1 /*TOP HALF OF DISPLAY*/
32   END1
33   CALL TIME(20)1
34   MEMORY(6400H) = 00H1
35   /*UPDATE SWITCH TO CAUSE DISPLAY MEMORIES TO BE REVERSED
36   THEREFORE DISPLAY CURRENT SCAN*/
37 END FILL1
38 END FILLMOD1
  
```

MODULE INFORMATION:

CODE AREA SIZE = 00F1H 241D
 VARIABLE AREA SIZE = 0002H 2D
 MAXIMUM STACK SIZE = 0002H 2D
 42 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - GAIN

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE GAINMOD
 OBJECT MODULE PLACED IN F1:GAIN.OBJ
 COMPILER INVOKED BY: PLM80 F1:GAIN.SRC TITLE('RBC DISPLAY - GAIN')

```

* PRINT(1LP)  DEBUG
1  GAINMOD: DD1
2  1  DECLARE GAIN(2) BYTE EXTERNAL
3  1  GAINER: PROCEDURE(X,Y) PUBLIC

/*THIS PROCEDURE CALCULATES THE GAIN IF ANY NEEDED TO BRING THE START BANG UP
TO FULL SCALE. THE MAXIMUM VALUE IN THE FIRST 5 DEGREES OF THE SCAN IS THE VARIABLE
X AND Y IS THE LAMP THIS VALUE APPLIES TO. THE FULL SCALE VALUE (255) IS DIVIDED BY
THE MAXIMUM SIGNAL OBTAINED. ONLY FOUR DISCREET GAINS ARE POSSIBLE:

0  1.00
1  1.33
2  1.67
3  2.00

THIS GAIN IS THEN APPLIED TO THE CORRESPONDING LAMP FOR THE NEXT FIVE SCANS.*/

4  2  DECLARE (X,Y) BYTE, P ADDRESS
5  2  P = 25500 / X
6  2  X = 0;

7  2  IF P > 133 THEN X = 1;
9  2  IF P > 167 THEN X = 2;
11 2  IF P > 200 THEN X = 3;
13 2  GAIN(Y) = X
14 2  END GAINER
15 1  END GAINMOD;

/*DIVIDE FULL SCALE(255) * 100 BY LARGEST VALUE FROM 0-5DEGREES*/
/*GAIN OF 1.00 IS ASSUMED UNLESS CHANGED BELOW*/
/*CALCULATE HOW MUCH IF ANY GAIN IS NEEDED*/
/*GAIN OF 1.33 IS NEEDED*/
/*GAIN OF 1.67 IS NEEDED*/
/*GAIN OF 2.00 IS NEEDED*/
/*PLACE GAIN NEEDED INTO GAIN FOR APPROPRIATE LAMP*/

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0058H  880
VARIABLE AREA SIZE = 0004H   40
MAXIMUM STACK SIZE = 0004H   40
30 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - HEIGHT

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE HEIGHTMOD
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 :F1:HEIGHT.SAV TITLE('RBC DISPLAY - HEIGHT') NOOBJECT

```

1      #PRINT(1LP:) DEBUG
2      HEIGHT#MOD: DO:
   DECLARE DCL LITERALLY 'DECLARE':
3      HEIGHT: PROCEDURE(Y) ADDRESS PUBLIC:
/*THIS PROCEDURE CONVERTS THE HEIGHT OF THE PEAKS IN 1/4 DEGREES TO HEIGHT IN
FEET*/
4      DCL Y ADDRESS:
/*VALUES GO FROM 0.00 TO 88.25 DEGREES AT 1/4 DEGREE INTERVALS BUT FOR CBH READINGS EXPECT VALUES FROM
6.75 TO 86.5 DEGREES ONLY BECAUSE OF FILTERING, AVERAGING, AND DEFINITION OF PEAK*/
5      DCL HIT(354) ADDRESS DATA(0.2,3.5,7.9,10.12,14.16,17,
19,21,23,24,26,28,30,31,33,35,37,39,40,42,44,46,47,49,51,53,
54,56,58,60,62,63,65,67,69,71,72,74,76,78,80,81,83,85,87,89,
91,92,94,96,98,100,102,103,105,107,109,111,113,115,117,118,120,122,124,126,
128,130,132,134,136,138,140,142,144,146,148,150,152,154,156,158,160,162,164,166,
168,170,172,174,176,178,180,182,184,186,188,190,192,194,196,198,199,202,204,206,208,
210,213,215,217,219,222,224,226,229,231,233,235,238,240,243,245,248,250,252,255,
257,260,262,265,267,270,272,275,278,280,283,285,288,291,293,296,299,302,304,307,
310,313,315,318,321,324,327,330,333,336,339,342,345,348,351,354,357,360,363,367,
370,373,376,380,383,386,390,393,397,400,404,407,411,414,418,422,425,429,433,437,
440,444,448,452,456,460,464,468,472,476,481,485,490,494,498,503,507,512,517,521,
526,531,536,541,546,551,556,561,566,571,577,582,587,593,599,604,610,616,622,628,
634,640,646,653,659,666,672,679,686,693,700,707,714,722,729,737,744,752,760,768,
777,785,794,802,811,820,829,839,848,858,868,878,888,898,909,920,931,942,954,966,
978,990,1003,1015,1029,1042,1056,1070,1084,1099,1114,1130,1145,1162,1178,1195,1213,1231,1250,1269,
1288,1308,1329,1350,1372,1395,1418,1442,1467,1493,1519,1547,1575,1604,1635,1666,1699,1733,1768,1804,
1842,1882,1923,1966,2011,2058,2107,2158,2212,2268,2328,2390,2456,2525,2599,2676,2759,2846,2939,3038,
3144,3259,3380,3510,3652,3806,3972,4155,4353,4572,4814,5082,5383,5720,6103,6540,7044,7632,8328,9162,
10181,11455,13092)

```

```

6      2      RETURN HIT(Y)
7      2      END HEIGHT:
8      1      END HEIGHT#MOD:

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0207H      727D
VARIABLE AREA SIZE = 0002H      2D
MAXIMUM STACK SIZE = 0000H      0D
37 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE INTONEMOD
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 IF1:INTONE.SAV TITLE('RBO

MODULE INFORMATION:

```

CODE AREA SIZE      = 005EH      94D
VARIABLE AREA SIZE = 0000H      0D
MAXIMUM STACK SIZE = 0002H      2D
40 LINES READ
0 PROGRAM ERROR(S)
END OF PL/M-80 COMPILATION

```

PL/M-80 COMPILER RBC DISPLAY - INTTWO

:SIS-II PL/M-80 V3.1 COMPILATION OF MODULE INTTWO
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLM80 IF1:INTTWO.SAV TITLE('RBC DISPLAY - INTTWO') NOOBJECT

```

1      $PRINT(1LP:)  DEBUG
2      INTTWO:MOD: DO:
3      1      DATACHECK: PROCEDURE(P,Y,Z) EXTERNAL:
4      2      DECLARE (Y,Z) BYTE, P ADDRESS:
5      3      END DATACHECK:
6      4      OFFCAL: PROCEDURE EXTERNAL:
7      5      END OFFCAL:
8      6      DECLARE (CNT,COFF,DONE,K3,MXCT,MXOFF1,MXOFF2,MXPK,RFLAG) BYTE EXTERNAL:
9      7      DECLARE (IC,POFF) ADDRESS EXTERNAL:
10     8      DECLARE A(361) BYTE EXTERNAL, GAIN(2) BYTE EXTERNAL, OFFSET(2) BYTE EXTERNAL:
11     9      DECLARE PEAKO(200) BYTE EXTERNAL:
12    10      INTTWO: PROCEDURE PUBLIC:
13    11      /*THIS PROCEDURE PROCESSES THE 120HZ INTERRUPT EITHER FOR COLLECTING CBH DATA
14    12      OR FOR COLLECTING NOISE DATA FOR THE OFFSET CORRECTION. IT FIRST RESET THE
15    13      COUNTER USED TO PROVIDE THE 120HZ INTERRUPT, AND THEN CHECKS TO SEE IF CBH OR
16    14      NOISE DATA ARE BEING COLLECTED. IF NOISE DATA ARE BEING COLLECTED OFFCAL IS
17    15      CALLED. OTHERWISE THE PROGRAM HERE IS EXECUTED FOR CBH DATA. THE DATA IS READ
18    16      FROM THE A TO D CONVERTER AND STORED IN ARRAY A. THEN IF THE SCAN IS AT 3
19    17      DEGREES THE GAIN AND OFFSET CORRECTIONS ARE APPLIED. IF THE SCAN IS AT 68
20    18      DEGREES THE OFFSET CORRECTION IS TAKEN OUT AND CONDITIONS ARE SET TO PROCESS
21    19      THE CBH DATA JUST COLLECTED AND TO BEGIN COLLECTED OFFSET DATA. */
22    20      MEMORY(8403H) = 30H: /*LOAD 8253 COUNTER #1 FOR USE AS THE 120HZ INTERRUPT*/
23    21      MEMORY(8400H) = 90H: /*LOAD LS BYTE FIRST! USING 137 BECAUSE INPUT CLOCK TO COUNTER
24    22      IS 90 MICRO SECS AND ABOUT 150 MICRO SECS ARE USED PROCESSING
25    23      THE INTERRUPT ITSELF*/
26    24      MEMORY(8400H) = 00H: /*LOAD MS BYTE SECOND*/
27    25      IF RFLAG = 2 OR RFLAG = 4 THEN CALL OFFCAL: /*CBH DATA ALREADY COLLECTED SO USE OFF SCAN
28    26      TO COLLECT NOISE DATA TO BE USED FOR OFFSET CORRECTION*/
29    27      ELSE DO: /*A READ OF THIS POSITION BEGINS THE CONVERT OF THE DATA*/
30    28      A(IC) = MEMORY(6800H): /*THIS WAITS FOR END OF CONVERSION FLAG*/
31    29      CALL DATACHECK(6A02H,20H,00H): /*READ AND STORE AWAY CONVERTED DATA 0 - 255*/
32    30      A(IC) = MEMORY(6800H): /*INCREASE COUNTER FOR CBH DATA POINTS*/
33    31      IC = IC + 1: /*LOOK FOR START BAND ONLY IN THE FIRST 4.75 DEGREES! THIS DATA WILL
34    32      BE USED FOR THE GAIN CORRECTION*/
35    33      IF IC = 20 THEN DO: /*FOR THE DEGREES AFTER 4.75 GAIN AND OFFSET CORRECTION ARE APPLIED*/
36    34      MEMORY(6602H) = GAIN(K3): /*ONLY COLLECT CBH DATA UP TO 90 DEGREES! AFTER THAT PROCESS
37    35      MEMORY(6600H) = OFFSET(K3): THE DATA AND START COLLECTING NOISE DATA FROM OFF SCAN FOR OFFSET*/
38    36      END: /*CHANGE RFLAG FROM 1 TO 2 OR FROM 3 TO 4*/
39    37      IF IC > 360 THEN DO: /*TAKE OUT OFFSET CORRECTION! BECAUSE NOISE VALUES WILL BE READ
40    38      RFLAG = RFLAG + 1: TO DETERMINE NEW OFFSET*/
41    39      MEMORY(6600H) = 00H: /*DUMMY VARIABLE DONE = 0 CBH DATA TO BE PROCESSED*/
42    40      DONE = 0:
43    41

```

PL/M-80 COMPILER RBC DISPLAY - INTTMO

32 4 COFF.MXOFF1.MXOFF2 = 0;

33 4 POFF = 1;

34 4 DO CNT = 0 TO 199;

35 5 PEAKO(CNT) = 0;

36 5 END; END; END;

39 2 END INTTMO;

40 1 END INTTMO#MOD1

/*COFF.MXOFF1, AND MXOFF2 ARE USED IN OFFCAL1 COFF IS THE COUNTER
FOR THE NOISE PEAKS FOUND IN THE OFF-SCAN, AND MXOFF1 AND MXOFF2
ARE THE MAXIMUM AND 2ND MAXIMUM PEAKS FOUND IN THE OFFSCAN.
MXOFF1 AND 2 ARE USED TO DETERMINE NO CBH ON A SUNNY DAY WHEN
THE SIGNAL CBH DATA IS VERY NOISY*/
/*POFF IS USED IN OFFCAL TO SPECIFY WHETHER THE LAST VALUE
WAS A PEAK OR NOT;POFF =1, LAST WAS NOT A PEAK*/
/*DO LOOP TO ZERO OUT PEAKO WHICH IS USED IN OFFCAL TO STORE
DATA COLLECTED IN THE OFF SCANS*/

MODULE INFORMATION:

CODE AREA SIZE = 00C3H 197D
VARIABLE AREA SIZE = 0000H 0D
MAXIMUM STACK SIZE = 0004H 4D
44 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - MTIME

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE MTIME.MOD
 OBJECT MODULE PLACED IN: F:\MTIME.OBJ
 COMPILER INVOKED BY: PLM80 IF: MTIME.SRC TITLE: RBC DISPLAY - MTIME

```

1      $PRINT(1LP); DEBUG
      MTIME$MOD; DOT
2      1      MTIME: PROCEDURE(X) PUBLIC REENTRANT;
      /*THE PURPOSE OF THIS PROCEDURE IS TO SIMPLY WASTE TIME. CALLING MTIME WITH A
      ZERO IS ILLEGAL. CALLING MTIME WITH A ONE RESULTS IN A WAIT OF 118 MICRO SECS*/
3      2      DECLARE (X,J) BYTE;
4      2      LOOP: J = 21
5      2      LOOP1: J = J - 11
6      2      IF J <> 0 THEN GO TO LOOP1;
7      2      X = X - 11
8      2      IF X <> 0 THEN GO TO LOOP;
9      2      END MTIME;
11     2      END MTIME$MOD;
12     1

```

/*DOES THIS LOOP X TIMES WHEN CALLED*/
 /*DOES THIS LOOP 2 TIMES FOR EVERY X TIMES*/

MODULE INFORMATION:

CODE AREA SIZE = 0028H 40D
 VARIABLE AREA SIZE = 0000H 0D
 MAXIMUM STACK SIZE = 0002H 2D
 18 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - OFFCAL
 ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE OFFCAL.MOD
 NO OBJECT MODULE REQUESTED
 COMPILER INVOKED BY: PLM80 IF1:OFFCAL.SAV TITLE('RBC DISPLAY - OFFCAL') NOOBJECT

```

1      $PRINT(ILIPI)  DEBUG
      OFFCAL#MOD: DO1

2      1      DATACHECK: PROCEDURE(P,Y,Z) EXTERNAL;
3      2      DECLARE (Y,Z) BYTE, P ADDRESS;
4      3      END DATACHECK;

5      1      DECLARE PEAKO(200) BYTE EXTERNAL;
6      2      DECLARE (COFF,MXOFF1,MXOFF2) BYTE EXTERNAL, POFF ADDRESS EXTERNAL;
7      3      OFFCAL: PROCEDURE PUBLIC;

/*THIS PROCEDURE IS USED BY THE 120HZ INTERRUPT DURING THE OFF SCAN TO COLLECT
NOISE DATA TO BE USED TO CALCULATE THE OFFSET NECESSARY. IT CONVERTS THE DATA TO
A DIGITAL VALUE FIRST THEN FINDS THE NOISE PEAKS (UP TO A MAXIMUM OF 200) IN THIS SCAN.
A NOISE PEAK IS DEFINED AS A NON-ZERO VALUE FOLLOWED BY A LESSER VALUE.
ALSO THE LARGEST AND SECOND LARGEST NOISE PEAKS ARE FOUND. THROUGH PAST EXPERIENCE
ON SUNNY DAYS A LARGE AMOUNT OF NOISE IS FOUND. THESE TWO MAXIMUMS ARE USED LATER
TO DIFFERENTIATE BETWEEN LARGE NOISE RETURNS ON THE CBH DATA SCAN AND REAL CLOUD
RETURNS.
IF THESE TWO MAXIMUMS ARE OVER A QUARTER SCALE THEN A SUNNY DAY IS ASSUMED AND
NO CBH IS REPORTED FOR THE ONE MINUTE READING. */

8      2      DECLARE X BYTE;

9      2      X = MEMORY(6800H);
10     2      CALL DATACHECK(6602H,20H,00H);
11     2      X = MEMORY(6800H);
12     2      IF X > PEAKO(COFF) THEN DO1;
13     3      PEAKO(COFF) = X;
14     3      POFF = 0;
15     3      END1;
16     3      ELSE DO1;
17     3      IF POFF = 0 THEN DO1;
18     3      POFF = 1;
19     3      IF PEAKO(COFF) > MXOFF1 THEN MXOFF1 = PEAKO(COFF);
20     3      IF PEAKO(COFF) > MXOFF2 THEN MXOFF2 = PEAKO(COFF);
21     3      ELSE DO1;
22     3      IF PEAKO(COFF) > MXOFF1 THEN MXOFF1 = PEAKO(COFF);
23     3      IF PEAKO(COFF) > MXOFF2 THEN MXOFF2 = PEAKO(COFF);
24     3      END1;
25     3      IF COFF < 199 THEN COFF = COFF + 1;
26     3      END1;
27     3      ELSE POFF = 1;
28     3      END1;
29     3      ELSE POFF = 1;
30     3      END1;
31     3      END OFFCAL;
32     2      END OFFCAL#MOD;
33     1

```

MODULE INFORMATION:

CODE AREA SIZE = 00A3H 1630
 VARIABLE AREA SIZE = 0001H 1D
 MAXIMUM STACK SIZE = 0004H 4D
 45 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - RESET

ISIS-11 PL/M-80 V3.1 COMPILATION OF MODULE RESETMOD
 NO OBJECT MODULE REQUESTED
 COMPILER INVOKED BY: PLM80 IF:RESET.SAV TITLE('RBC DISPLAY - RESET') NOOBJECT

```

1      @PRINT(1LP1) DEBUI
      RESETMOD: DO:
2      1      DECLARE DONE BYTE EXTERNAL, IC ADDRESS EXTERNAL:
3      1      RESET: PROCEDURE PUBLIC:
      /*THIS PROCEDURE RESETS THE NECESSARY VARIABLES AFTER THE CURRENT SCAN HAS BEEN
      PROCESSED*/
4      2      IF DONE = 0 THEN DO:
6      3      DONE = 1:
7      3      IC = 362:
8      3      END:
9      2      END RESET:
10     1      END RESETMOD:
      /*DONE = 0 MEANS A SCAN HAS FINISHED BUT HAS NOT BEEN FULLY PROCESSED*/
      /*DONE = 1 MEANS LAST SCAN HAS BEEN PROCESSED AND WAITING FOR NEW SCAN*/
      /*PROCESSING OF PREVIOUS LAMP IS DONE! THEREFORE REENABLE INTER ONE*/

MODULE INFORMATION:
CODE AREA SIZE      = 0014H 20D
VARIABLE AREA SIZE = 0000H 0D
MAXIMUM STACK SIZE = 0000H 0D
18 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

```

PL/N-80 COMPILER RBC DISPLAY - SCANS

ISIS-11 PL/N-80 V3.1 COMPILATION OF MODULE SCANSMOD
NO OBJECT MODULE REQUESTED
COMPILER INVOKED BY: PLN80 \$FI:SCANS.SAV TITLE('RBC DISPLAY - SCANS') NOOBJECT

```

1      @PRINT(1,LP1)  DEBUG
2
3      SCANSMOD: DO1
4
5      HEIGHT: PROCEDURE(Y) ADDRESS EXTERNAL;
6      DECLARE Y ADDRESS;
7      END HEIGHT;
8
9      DECLARE (K3,K4,K5) BYTE EXTERNAL;
10     DECLARE BEIL(2) ADDRESS EXTERNAL, CEIL(2) ADDRESS EXTERNAL, CLOUD(2) ADDRESS EXTERNAL;
11     DECLARE CLOUDY(22) ADDRESS EXTERNAL, HEIT(20) ADDRESS EXTERNAL, RCT(2) BYTE EXTERNAL;
12
13     SCANS: PROCEDURE (NO2) PUBLIC;
14
15     /* THIS PROCEDURE TAKES THE MAXIMUM PEAK HEIGHT IN 1/4 DEGREES FOR 5 SCANS AND
16     FINDS A REPRESENTATIVE CLOUD BASE HEIGHT(CBM) FOR 5 SCANS. THE FIVE HEIGHTS(IN
17     1/4 DEGREES) ARE FIRST TEST TO SEE IF TWO VALUES ARE EQUAL. IF THERE ARE THIS
18     VALUE IS CHOSEN AS THE CBM. IF THERE ARE MORE THAN ONE PAIR THAT ARE EQUAL, THE
19     LOWER PAIR IS SELECTED. IF THE FIRST TEST FAILS, THE FIVE ARE TESTED TO SEE IF
20     TWO HEIGHTS ARE WITHIN +-1 DEGREE. IF BOTH METHODS SATISFY THIS
21     CRITERIA THE LOWER ONE OF THE LOWER PAIR IS SELECTED. IF BOTH METHODS FAIL A
22     FINAL ATTEMPT IS MADE TO SEE IF ONE OF THE FIVE IS WITHIN +-1/2 DEGREES
23     OF THE PREVIOUS CBM PROVIDED THAT THE PREVIOUS CBM WAS NOT "NO CBM". IF THERE ARE,
24     THE LOWER OF THE TWO IS CHOSEN. IF MORE THAN ONE MATCHES, THEN THE LOWEST IS
25     SELECTED. */
26
27     DECLARE (J,K,NO2,Q) BYTE;
28     DECLARE (I,X1,X2,X3,X4,X5) ADDRESS;
29
30     Q,CEIL(NO2) = 0;
31     K = 10 * NO2;
32
33     /* Q IS IDENTIFIER TO SHOW WHEN A CBM HAS BEEN FOUND,
34     CEIL(NO2) IS WHERE THE HEIGHT OF CBM WILL BE STORED WHEN FOUND.
35     /*K WILL SHOW WHERE THE CURRENT MAX PEAKS OF THE LAST 5 SCANS
36     ARE FOUND IN HEIT.*/
37
38     /* *****FIRST METHOD FIND TWO VALUES EQUAL***** */
39
40     DO I = K TO K+3;
41     DO J = (I+1) TO K+4;
42     IF HEIT(I) < 349 THEN DO;
43     /*IF TWO HEIGHTS ARE EQUAL*/
44     IF HEIT(I) = HEIT(J) THEN DO;
45     /*FIRST TIME ANY TWO VALUES ARE EQUAL*/
46     CEIL(NO2) = HEIT(I);
47     Q = 1;
48     /*Q=1 MEANS A MATCH HAS BEEN FOUND*/
49     ELSE DO;
50     IF CEIL(NO2) > HEIT(I) THEN DO;
51     /*SECOND PAIR OF DATA POINTS ARE AT A LOWER HEIGHT THAN FIRST*/
52     CEIL(NO2) = HEIT(I);
53     /*REPLACE FIRST PAIR WITH SECOND PAIR, BECAUSE 2ND PAIR LOWER*/
54     END;
55     END;
56     END;
57
58     /* ***** SECOND METHOD (FIRST METHOD FAILED) FIND TWO VALUES WITHIN +-1 DEGREE ***** */

```

```

34 2 IF Q = 0 THEN DO1
36 3 X2 = 01
37 4 DO I = K TO K+31
38 5 IF HEIT(I) < 349 THEN DO1
40 6 X5 = HEIT(I)
41 7 IF X5 < 32 THEN X4 = 271
43 8 ELSE X4 = X5 - 41
44 9 IF X5 > 343 THEN X3 = 3481
46 10 ELSE X3 = X5 + 41
47 11 DO J = (I+1) TO K+41
48 12 IF HEIT(J) < 349 THEN DO1
49 13 IF HEIT(J) >= X4 THEN DO1
50 14 IF HEIT(J) <= X3 THEN DO1
51 15 IF HEIT(J) < HEIT(I) THEN X1 = HEIT(J)
52 16 ELSE X1 = HEIT(I)
53 17 IF X2 = 0 THEN X2 = X11
54 18 ELSE DO1
55 19 IF X1 < X2 THEN X2 = X11
60 20 END1
61 21 CEIL(N02) = X21
62 22 Q = 11
63 23 END1
64 24 END1
65 25 END1

/* TWO HEIGHTS NOT EQUAL-NOW CHECK TO FIND 2 HEIGHTS CLOSE TO EACH OTHER*/
/* DUMPY VARIABLE TO STORE ANY MATCHES FOUND*/
/* DO LOOP TO CHECK SCANS 1 THRU 4*/
/* ONLY WANT TO COMPARE THOSE SCANS WITH PEAKS! NOT WITH NOCBH*/
/* X5 WILL BE THE CURRENT HEIGHT TO BE COMPARED */
/* X4 WILL BE CURRENT HEIGHT MINUS FOUR VALUES EXCEPT LOWEST VALUE
CAN BE ONLY 27*/
/* HEIT(I) - 4 IN X4*/
/* X3 WILL BE CURRENT HEIGHT PLUS FOUR VALUES EXCEPT HIGHEST VALUE
CAN ONLY BE 348*/
/* HEIT(I) + 4 IN X3*/
/* CHECK CURRENT HEIT(I) AGAINST OTHER HEIGHTS FROM SCAN 2-5*/
/* ONLY WANT TO COMPARE THOSE SCANS WITH PEAKS! NOT WITH NOCBH*/
/* GREATER THAN CURRENT HEIT(I)-4*/
/* LESS THAN CURRENT HEIT(I)+4*/
/* IF A MATCH HAS BEEN FOUND NOW PUT LOWER HEIGHT IN X1*/
/* FIRST PAIR OF DATA TO MATCH SO PUT THE LOWER VALUE FOUND IN X2*/
/* SECOND PAIR OF DATA TO MATCH SO MUST CHOOSE LOWER ONE*/
/* IF CURRENT MATCH IS LOWER THAN THE PREVIOUS ONE PUT IN X1*/
/* PUT LOWER MATCH FOUND IN CEIL(N02)*/
/* IDENTIFIER THAT A MATCH HAS BEEN FOUND*/
END1
END1
END1

/* ***** THIRD METHOD (FIRST TWO FAILED) TEST IF ONE VALUE WITHIN +-6 VALUES(+-1 1/2 DEGREES)
OF PREVIOUS CBH ***** */

/* NOT EQUAL 1/2 VALUES NOT CLOSE, SO NOW TRY CLOSE TO LAST REPORTED CBH*/
/* LAST MINUTE'S VALUE MUST BE A PEAK VALUE NOT A NOCBH*/
/* X2 WILL BE USED TO STORE LOWER OF MATCH FOUND*/
/* COMPARE 5 SCANS AGAINST LAST MINUTE'S VALUE*/
/* THE SCAN BEING COMPARED MUST BE A PEAK VALUE NOT NOCBH*/
/* THE CURRENT VALUE BEING COMPARED IS IN X3*/
/* X4 WILL BE CURRENT VALUE MINUS 6 WITH A MINIMUM OF 27*/
/* HEIT(J) - 6 IN X4*/
/* X3 WILL BE CURRENT VALUE PLUS 6 WITH A MAXIMUM OF 348*/
/* HEIT(J) + 6 IN X3*/
/* PREVIOUS HEIGHT > HEIT(J) - 6*/
/* PREVIOUS HEIGHT <= HEIT(J) + 6*/
/* BEIL(N02)1 / LOWER HEIGHT IN X1*/
/* FIRST MATCH WITH PREVIOUS REPORTED CBH*/
/* SECOND MATCH WITH PREVIOUS REPORTED CBH*/
/* SECOND MATCH WAS LOWER THAN THE FIRST*/

END1
CEIL(N02) = X21
Q = 11
END1
END1
END1
IF CEIL(N02) = 0 THEN CEIL(N02) = 10001
/* CEIL IS FROM 27 TO 348 FOR CLOUDS, 1000 FOR NOC10*/

/* NOW THAT A CBH HAS BEEN FOUND THE FOLLOWING CONVERTS FROM 1/4 DEGREES TO FEET*/
IF CEIL(K3) > 348 THEN DO1
IF CEIL(K3) = 1000 THEN CLOUD(K3) = 100001
/* CEIL FOUND IS NO CBH OR A MALFUNCTION*/
/* CEIL FOUND IS NO CBH REPRESENTED BY 10000*/

```

PL/M-80 COMPILER RBC DISPLAY - SCANS

```

114 3      ELSE CLOUD(K3) = 0;
115 3      END;
116 2      ELSE CLOUD(K3) = HEIGHT(CEIL(K3));
117 2      CEIL(K3) = CEIL(K3);
118 2      DISABLE;
119 2      K4 = K3 * 10;
120 2      K5 = K3 * 11;
121 2      CLOUDY(K5) = CLOUD(K3);
122 2      K3 = K5 + 1;
123 2      DO I = 0 TO 9;
124 3      IF HEIT(K4+1) = 1000 THEN CLOUDY(K5+1) = 1000;
125 3      ELSE CLOUDY(K5+1) = HEIGHT(HEIT(K4+1));
126 3      END;
127 2      ENABLE;
128 2      RCT(K3) = 4;
129 2      DO I = 0 TO 3;
130 3      HEIT(K4+1) = HEIT(K4+I+1);
131 3      HEIT(K4+1+5) = HEIT(K4+I+5);
132 3      END;
133 2      END SCANS;
134 2      END SCANS;
135 1      END SCANS;MOD;

```

```

/*A MALFUNCTION SOMEWHERE*/
/*A CBN HEIGHT HAS BEEN FOUND NON COMMENT TO FEET*/
/*SAVE THE CURRENT CBN IN HEIL TO BE USED
AS PREVIOUS READING IN NEXT DETERMINATION*/
/*DISABLE SO WE CAN STORE ALL CURRENT MAX.2ND MAX.
AND CBN IN CASE LATER WE WANT TO SEND OUT*/
/*K4 WILL TELL WHERE CURRENT VALUEDSTONED IN HEIT*/
/*K5 WILL SHOW WHERE CURRENT VALUEDSTONED IN CLOUDY*/
/*FIVE SCAN VALUE FIRST IN DATA*/
/*DO LOOP TO STORE MAX AND 2ND MAX PEAKS' HEIGHTS*/
/*CONVERT 1/4DEGREE HEIGHTS TO FEETHEIGHTS*/
/*SET SCAN COUNTER BACK TO FOUR*/
/*DO LOOP TO DISCARD OLDEST VALUE AND MAKE
ROOM FOR NEXT VALUE*/
/*FOR MAX PEAK*/
/*FOR SECONDARY MAX PEAK*/

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0556H 1346D
VARIABLE AREA SIZE = 0010H 16D
MAXIMUM STACK SIZE = 0004H 4D
132 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-80 COMPILATION

PL/M-80 COMPILER RBC DISPLAY - VARY

ISIS-11 PL/M-80 V3.1 COMPILATION OF MODULE VARYMOD
 OBJECT MODULE PLACED IN IF1:VARY.OBJ
 COMPILER INVOKED BY: PLM80 IF1:VARY.SRC TITLE('RBC DISPLAY - VARY')

```

1  VARYMOD: DO1
2  DECLARE CTR ADDRESS EXTERNAL;
3  VARY: PROCEDURE PUBLIC;
/*THIS PROCEDURE CAUSES THE CURSOR TO MOVE VERY SLOWLY FOR THE FIRST CHANGE
  BUT THEN AFTER THE FIRST, THE CURSOR WILL MOVE VERY RAPIDLY.
4  DECLARE I BYTE;
5  IF CTR < 1 THEN DO1
6  DO I = 0 TO 19;
7  CALL TIME(250);
8  END;
9  CTR = CTR + 1;
10 IF CTR = 20 THEN DO1
11 ELSE CALL TIME(200);
12 END;
13 END VARY;
14
/*CHECK TO SEE IF FIRST MOVE OF CURSOR OR A SUCCEEDING ONE*/
/*IF FIRST MOVE WAIT 1/2 SECOND BEFORE GOING ON*/
/*JUST KILLING TIME*/
/*INCREMENT MOVE COUNTER*/
/*WASTE ONLY 20 MSEC BEFORE PROCEEDING FOR ANY MOVE AFTER FIRST ONE*/

```

MODULE INFORMATION:

CODE AREA SIZE = 0035H 53D
 VARIABLE AREA SIZE = 0001H 1D
 MAXIMUM STACK SIZE = 0002H 2D
 22 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION

Appendix C

Conversion Table from Pulse Counts to Heights

Elevation Angle	Pulse Count	Height (ft)	Elevation Angle	Pulse Count	Height (ft)
6.75	27	47	14.00	56	100
7.00	28	49	14.25	57	102
7.25	29	51	14.50	58	103
7.50	30	53	14.75	59	105
7.75	31	54	15.00	60	107
8.00	32	56	15.25	61	109
8.25	33	58	15.50	62	111
8.50	34	60	15.75	63	113
8.75	35	62	16.00	64	115
9.00	36	63	16.25	65	116
9.25	37	65	16.50	66	118
9.50	38	67	16.75	67	120
9.75	39	69	17.00	68	122
10.00	40	71	17.25	69	124
10.25	41	72	17.50	70	126
10.50	42	74	17.75	71	128
10.75	43	76	18.00	72	130
11.00	44	78	18.25	73	132
11.25	45	80	18.50	74	134
11.50	46	81	18.75	75	136
11.75	47	83	19.00	76	138
12.00	48	85	19.25	77	140
12.25	49	87	19.50	78	142
12.50	50	89	19.75	79	144
12.75	51	91	20.00	80	147
13.00	52	92	20.25	81	148
13.25	53	94	20.50	82	150
13.50	54	96	20.75	83	152
13.75	55	98	21.00	84	154

Elevation Angle	Pulse Count	Height (ft)	Elevation Angle	Pulse Count	Height (ft)
21.25	85	156	35.25	141	283
21.50	86	158	35.50	142	285
21.75	87	160	35.75	143	288
22.00	88	162	36.00	144	291
22.25	89	164	36.25	145	293
22.50	90	166	36.50	146	296
22.75	91	168	36.75	147	299
23.00	92	170	37.00	148	302
23.25	93	172	37.25	149	304
23.50	94	174	37.50	150	307
23.75	95	176	37.75	151	310
24.00	96	178	38.00	152	313
24.25	97	180	38.25	153	315
24.50	98	182	38.50	154	318
24.75	99	184	38.75	155	321
25.00	100	187	39.00	156	324
25.25	101	189	39.25	157	327
25.50	102	191	39.50	158	330
25.75	103	193	39.75	159	333
26.00	104	195	40.00	160	336
26.25	105	197	40.25	161	339
26.50	106	199	40.50	162	342
26.75	107	202	40.75	163	345
27.00	108	204	41.00	164	348
27.25	109	206	41.25	165	351
27.50	110	208	41.50	166	354
27.75	111	210	41.75	167	357
28.00	112	213	42.00	168	360
28.25	113	215	42.25	169	363
28.50	114	217	42.50	170	367
28.75	115	219	42.75	171	370
29.00	116	222	43.00	172	373
29.25	117	224	43.25	173	376
29.50	118	226	43.50	174	380
29.75	119	229	43.75	175	383
30.00	120	231	44.00	176	386
30.25	121	233	44.25	177	390
30.50	122	235	44.50	178	393
30.75	123	238	44.75	179	397
31.00	124	240	45.00	180	400
31.25	125	243	45.25	181	404
31.50	126	245	45.50	182	407
31.75	127	248	45.75	183	411
32.00	128	250	46.00	184	414
32.25	129	252	46.25	185	418
32.50	130	255	46.50	186	422
32.75	131	257	46.75	187	425
33.00	132	260	47.00	188	429
33.25	133	262	47.25	189	433
33.50	134	265	47.50	190	437
33.75	135	267	47.75	191	440
34.00	136	270	48.00	192	444
34.25	137	272	48.25	193	448
34.50	138	275	48.50	194	452
34.75	139	278	48.75	195	456
35.00	140	280	49.00	196	460

Elevation Angle	Pulse Count	Height (ft)	Elevation Angle	Pulse Count	Height (ft)
49.25	197	464	63.25	253	794
49.50	198	468	63.50	254	802
49.75	199	472	63.75	255	811
50.00	200	478	64.00	256	820
50.25	201	481	64.25	257	829
50.50	202	485	64.50	258	839
50.75	203	490	64.75	259	848
51.00	204	494	65.00	260	858
51.25	205	498	65.25	261	868
51.50	206	503	65.50	262	878
51.75	207	507	65.75	263	888
52.00	208	512	66.00	264	898
52.25	209	517	66.25	265	909
52.50	210	521	66.50	266	920
52.75	211	526	66.75	267	931
53.00	212	531	67.00	268	942
53.25	213	536	67.25	269	954
53.50	214	541	67.50	270	966
53.75	215	546	67.75	271	978
54.00	216	551	68.00	272	990
54.25	217	556	68.25	273	1003
54.50	218	561	68.50	274	1015
54.75	219	566	68.75	275	1029
55.00	220	571	69.00	276	1042
55.25	221	577	69.25	277	1056
55.50	222	582	69.50	278	1070
55.75	223	587	69.75	279	1084
56.00	224	593	70.00	280	1099
56.25	225	599	70.25	281	1114
56.50	226	604	70.50	282	1130
56.75	227	610	70.75	283	1145
57.00	228	616	71.00	284	1162
57.25	229	622	71.25	285	1178
57.50	230	628	71.50	286	1195
57.75	231	634	71.75	287	1213
58.00	232	640	72.00	288	1231
58.25	233	644	72.25	289	1250
58.50	234	653	72.50	290	1269
58.75	235	659	72.75	291	1288
59.00	236	666	73.00	292	1308
59.25	237	672	73.25	293	1329
59.50	238	679	73.50	294	1350
59.75	239	686	73.75	295	1372
60.00	240	693	74.00	296	1395
60.25	241	700	74.25	297	1418
60.50	242	707	74.50	298	1442
60.75	243	714	74.75	299	1467
61.00	244	722	75.00	300	1493
61.25	245	729	75.25	301	1519
61.50	246	737	75.50	302	1547
61.75	247	744	75.75	303	1575
62.00	248	752	76.00	304	1604
62.25	249	760	76.25	305	1635
62.50	250	768	76.50	306	1666
62.75	251	777	76.75	307	1699
63.00	252	785	77.00	308	1733

Elevation Angle	Pulse Count	Height (ft)	Elevation Angle	Pulse Count	Height (ft)
77.25	309	1768	82.00	328	2846
77.50	310	1804	82.25	329	2939
77.75	311	1842	82.50	330	3038
78.00	312	1882	82.75	331	3144
78.25	313	1923	83.00	332	3259
78.50	314	1966	83.25	333	3380
78.75	315	2011	83.50	334	3510
79.00	316	2058	83.75	335	3652
79.25	317	2107	84.00	336	3806
79.50	318	2158	84.25	337	3972
79.75	319	2212	84.50	338	4155
80.00	320	2268	84.75	339	4353
80.25	321	2328	85.00	340	4572
80.50	322	2390	85.25	341	4814
80.75	323	2456	85.50	342	5082
81.00	324	2525	85.75	343	5383
81.25	325	2599	86.00	344	5720
81.50	326	2676	86.25	345	6103
81.75	327	2759	86.50	346	6540