DOE/ET/34109--T1

MASTER

DEL Manufacturing Company

905 Monterey Pass Road

Monterey Park, CA. 91754

Phone: 264-5900

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DEL Manufacturing Company

FINAL TECHNICAL REPORT

TRACKING INSTRUMENT AND CONTROL FOR SOLAR CONCENTRATORS

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

FINAL TECHNICAL REPORT

Reporting Period: October 1979 - January 1981

TRACKING INSTRUMENT AND CONTROL
FOR SOLAR CONCENTRATORS

CONTRACT NO. DE-AC01-79ET21009

This report was sponsored by the
Department of Energy
under the contract cited above.

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product, or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

G K Systems Inc. has made application for a patent to protect technical disclosures embodied in this report.

G K SYSTEMS, INC.
749 Lakefield Road, Suite 'A'
Westlake Village
California 91361

Authors - John Gray
John Kuhlman

31st January 1981

TABLE OF CONTENTS

PARAGRAPH	DESCRIPTION	PAGE NO.
1.0	Distribution List	1-1
2.0	Abstract	2-1
3.0	Description of Project	3-1
4.0	Solar Tracker	4-1
.1	General	4-1
.2	Specifications	4-2
4.1	The "POINT" Solar Tracking System	4-3
.1	The System	4-3
.2	Central Controller Based System	4-3
.3	Local Controller Based System	4-4
4.2	Tracking Head - Mechanical	4-5
.1	General	4-5
.2	Design Objectives	4-5
.3	Design Achievements	4-6
4.3	Final Design - Tracking Head	4-8
4.4	Final Design - Local Controller	4-9
.1	General	4-9
.2	System - Module Boards	4-11
.3	Interconnections	4-11
.4	Schematic and Cable Diagrams	4-11
.5	Component List - System Control Board	4-12
.6	Component List - Module Boards	4-13

TABLE OF CONTENTS (cont.)

PARAGRAPH	DESCRIPTION	PAGE NO.
5.0	Circuit Descriptions	5-1
.1	Basic Operation	5-2
.2	Functional Considerations	5-3
.1	Scan Address Counter and Address Latches	5-3
.2	Active - Passive	5-3
.3	PROM Memory	5-3
.4	Switch Terminations	5-3
.5	Switch Status Register	5-3
.6	Motor Control Register	5-3
.7	Position Motor Control Module	4-4
.8	Position Motor Power Module	5-4
.9	Scan Motor Module	5-4
.10	Photodiode and Index Amplifiers	5-4
.11	Analog Select and Sample Hold	5-4
.12	Signal Synchronization and ADC Control	5-4
.13	Unit Address and Local Select	5-4
.14	Command Decode	5-4
.15	Microcomputer	5-5
.16	Send/Receive	5-5
5.3	Electrical Diagrams	5-6
6.0	"POINT" Operating System Description	6-1
.1	Application Program Considerations	6-3
.2	"POINT" Program Tables	6-5
6.1	"POINT" Operating System Definitions	6-6

TABLE OF CONTENTS (cont.)

PARAGRAPH	DESCRIPTION	PAGE NO.
6.2.0	"POINT" Application Program Description	6-35
.1	Program Sequences	6-36
.2	Application Program Module Descriptions	6-38
6.3	"POINT" Application Definitions	6-40
7.0	Project Evaluation	7-1
.1	Results	7-4
.2	Conclusions	7-5
.3	Enhancements	7-6

APPENDIXES

I	"POINT" Application Program Modules	
II	Demonstration Unit	

1.0 DISTRIBUTION LIST

1.1 General

The Report Distribution List furnished with the contract indicates the following recipients for the report.

Mr. Martin Gutstein
Program Manager
Advanced Thermal Technology
Office 408, 600 E Street N.W.
Washington, D.C. 20545

Mr. Howard Coleman
Program Official
Advanced Thermal Technology
600 E. Street N.W.
Washington, D.C. 20545

Chief, MIS and Control Branch
Doe/Solar Technology
Mail Stop 404, 600 E Street, N.W.
Washington, D.C. 20545

In addition copies will also be sent to:

Department of Energy
Technical Information Center (T.I.C.)
Special Assistant for Reproduction and Processing
Post Office Box 62
Oak Ridge, Tennessee 37830

Solar Energy Research Institute
1617 Cole Boulevard
Golden, CO 80401
Solar Thermal Program Office 17/2
Attention: Contracts Manager

2.0 ABSTRACT

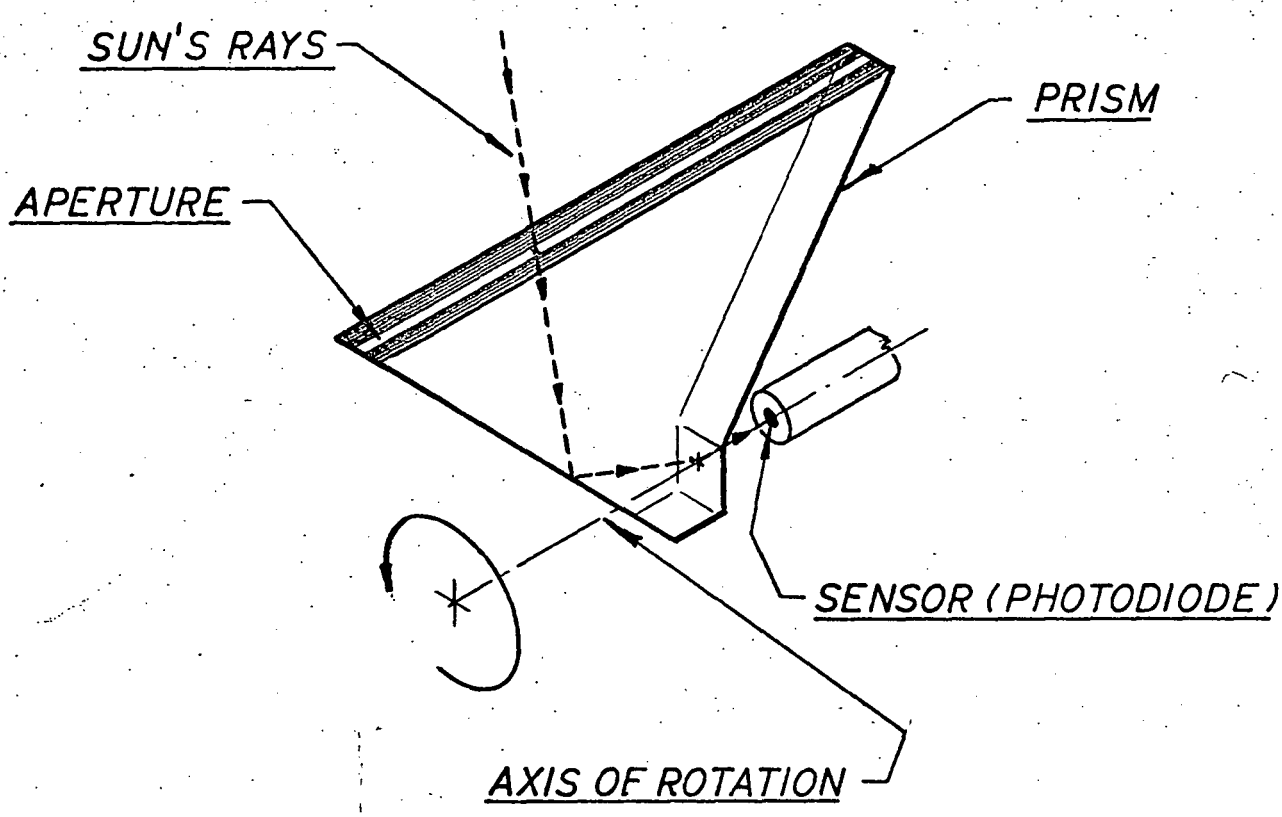
The solar tracking instrument and control concept investigated has application to both line and point focus collectors.

As detailed in our proposal (offered in response to PRDA No. RA01-79-ET21008 and dated May 22, 1978) we consider a digital technology based approach to solar tracking is mandatory in order to overcome the inherent problems possessed by existing equipment. Such an approach has advantage of providing a basis for coordinated control and management of system requirements for solar concentrator installations.

The digital design philosophy has been pursued from the onset on this contract. Both the circuit design and equipment configuration have been structured to provide flexible system capabilities. This flexibility will permit development of special capabilities compatible with the existing design to meet the needs of future installations without the necessity of system redesign.

3.0 DESCRIPTION OF PROJECT

Development of a tracking instrument and associated controller based on digital technology. This equipment will be used to test the single sensor concept for solar tracking and insulation measurement and in addition provide data on the coordinated control of solar concentrator ancillary functions.



4.0 SOLAR TRACKER

4.0.1 GENERAL

As illustrated, the G K Systems Tracker uses a single photo sensor, and a rotating aperature to obtain tracking accuracies better than 1.5 mrads (0.1 degs). Peak signal detection is used to eliminate tracking of false sources, i.e., clouds, etc. A prism is employed to obtain an extended field of view (150 degs axially - 360 degs radially). The tracker digitally measures the Suns displacement angle relative to the concentrator axis, and repositions it incrementally. This arrangement permits the use of low cost non-servo motors.

The local controller contains microprocessor based electronics, incorporating digital signal processing. A single controller may be time shared by a maximum of sixteen trackers, providing a high performance, cost effective solar tracking system, suitable for both line and point focus concentrators.

An installation may have the local controller programmed as a standalone unit or slaved to a central controller. When used with a central controller, dynamic data monitoring and logging is available, together with the ability to change system modes and parameters, as desired.

4.0.2 SPECIFICATIONS

TRACKING HEAD

Tracking Accuracy	-	1.5 mrads (<0.1 degs)
Field of View	-	150 degs axially 360 degs radially
Insolation Level	-	1% of maximum

LOCAL CONTROLLER

Input Voltage	-	115 VAC, 1 phase, 60 Hz
Input Current	-	1/2 AMP plus Positioning Motor current
Addressing	-	256 unique addresses available
Serial Port	-	Synchronous send/receive, transformer coupled
Operating System	-	Real Time, Foreground/Background
Outputs	-	1-16 Positioning Units
System Parameters	-	Many are user selectable, if other than the default values are desired e.g. Tracking Accuracy Ephemeris mode insolation level Tracking mode insolation level

POSITIONING UNIT

Switch Inputs	-	2 Travel Limits 1 Unassigned
Digital Inputs	-	1 Scan Index 2 Unassigned
Analog Inputs	-	1 Photo Diode 4 Unassigned
Digital Outputs	-	3 Unassigned
Motor Drives	-	1 Scan Motor 1 Positioning Motor 120 VAC, specify current NOTE: Unassigned inputs and outputs are user programable.

OPTIONS

Standalone Module	-	Replaces serial port for standalone inputs and outputs
Scan Inverter	-	Used with 12 volt DC systems
DC Motor Control	-	Used with DC positioning motors, Specify voltage and current
Program Modules	-	User-defined program modules may be added or substituted to application programs

4.1 THE "POINT" SOLAR TRACKING SYSTEM

POINT is an acronim for Precision Optical INstrument Tracker which describes the fundimental concept underlying the design. The device is an instrument capable of digitally measuring the suns displacement relative to a fixed axis with a high degree of precision (1.5 mrads - 0.1 degs.). When used in conjunction with it's associated LOCAL CONTROLLER the TRACKER provides a new approach to solar tracking effectively overcoming the inherent problems of shadow band based designs.

4.1.1 THE SYSTEM

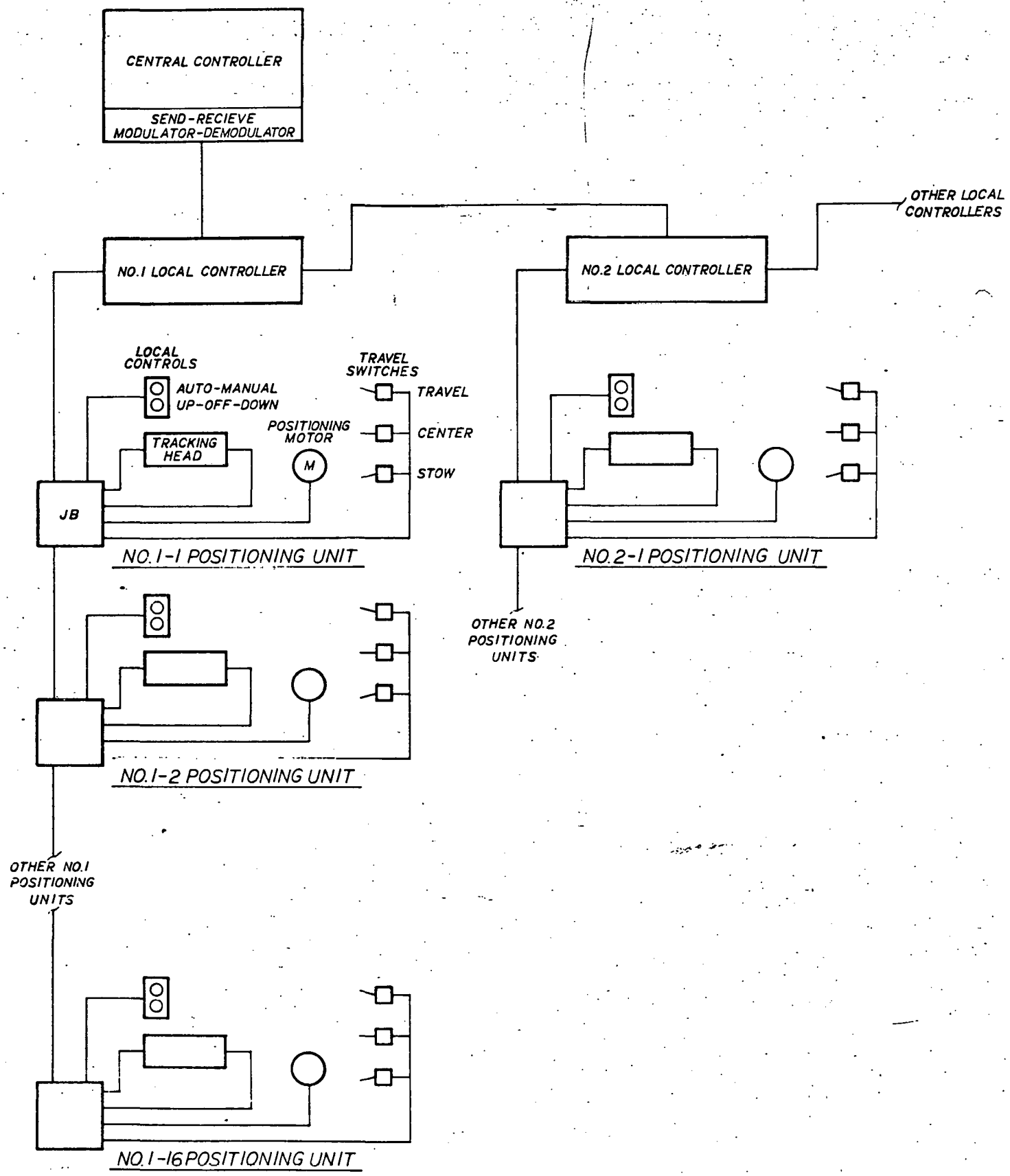
The system is a modular arrangement permitting one central controller to be time shared by a maximum of 256 LOCAL CONTROLLERS. These in turn may be each time shared by a maximum of 16 POSITIONING UNITS see Drg. No. 100.

A POSITIONING UNIT comprises all equipment necessary for operation of one concentrator drive system, i.e., motor, TRACKING HEAD, travel switches and local controls.

4.1.2 CENTRAL CONTROLLER BASED SYSTEM

It is anticipated that most medium size and larger solar concentrator installations will have a mini-computer for controlling the process system incorporated in their basic design. This computer could be arranged to act as the central controller for the tracking system. Communications with the LOCAL CONTROLLERS would be via the SEND-RECEIVE MODULATOR - DEMODULATOR compatable with most

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED	TITLE SYSTEM DIAGRAM	NEXT ASM / USED ON	GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361
	CUSTOMER DOE	PROJ NO / REF	JOB NO / DRN FG / CKD / DATE 1-6-81 / DRG NO. 100 / SHT 1 OF 1 / REV A

mini-computers having a USRT or USART. A central controller provides the capability of changing operating parameters when the situation dictates via entries through the computer's keyboard. It also provides dynamic data monitoring and logging permitting more efficient management of the installation.

4.1.3 LOCAL CONTROLLER BASED SYSTEM

For less sophisticated installations the LOCAL CONTROLLER may be arranged to operate as a stand alone unit. Such an arrangement limits the systems capabilities to tracking to pre-established criteria without the ability to change parameters, without reprogramming. Data monitoring and logging are unavailable without a central controller. Several special purpose inputs are provided for external devices, such as clocks, rain detectors, hail detectors, etc.

* * * * *

4.2 TRACKING HEAD - MECHANICAL

4.2.1 GENERAL

As stated in our proposal the fundamental concept used in the design of the test device, i.e., "A precisely sized aperture being rotated with constant angular velocity"...was retained when the TRACKING HEAD was redesigned.

The need to increase the field of view, so that the TRACKING HEAD would be capable of scanning "From horizon to horizon along the North-South axis and simultaneously view the Northern-Southern horizon...", was given paramount importance in the final configuration to ensure that the unit could readily acquire the sun regardless of it's and the concentrators orientation, and provide a flexible approach to tracking.

4.2.2 DESIGN OBJECTIVES

<u>Increasing field of view</u>

Two methods of accomplishing this were suggested in the proposal:

a. Mounting the photodiode stationary on the drum's axis of rotation and transmitting the suns rays to it by a fiber optic, which would be attached to the rotating drum. Such an approach could provide a full 360° scanning on the North-South axis, and by forming a lens on the receiving end of the fiber, the acceptance angle on the East-West axis might be increased.

b. Devise a prism/lens-fiber optic arrangement which would use a photodiode mounted in the manner described above. By selection

of optical materials with suitable index of refraction, this arrangement could further enhance the acceptance angle and the associated field of view on the East-West axis.

Tracking accuracy

Discussions with various agencies and concentrator manufactures prior to the commencement of the design phase indicated that a tracking accuracy of $\pm 0.1^{\circ}$ was desirable for most applications. The design was predicated upon attaining a tracking accuracy equal to or better than this value.

4.2.3 DESIGN ACHIEVEMENTS

Increasing field of view

The use of optical fibers in the manner described in a.) above was explored as the initial arrangement that appeared to offer the most cost effective configuration to attain an increased field of view with a stationary mounted photodiode. However, after extensive analysis the approach was recognized to be deficient and was abandoned. The major problem encountered was radiation losses. These occur when the fibers are bent, and in order to reduce them to acceptable level the radius of curvature of the fiber must be at least 40 times its diameter. It was determined that radii of that magnitude would increase the dimensions and cost of the TRACKING HEAD to an unacceptable extent. Once this problem of radiation loss and its influence on equipment size was fully appreciated it was apparent that it would have a similar impact on a design embracing the alternative concept b.) outlined in the proposal. The recognition that optical fibers were an

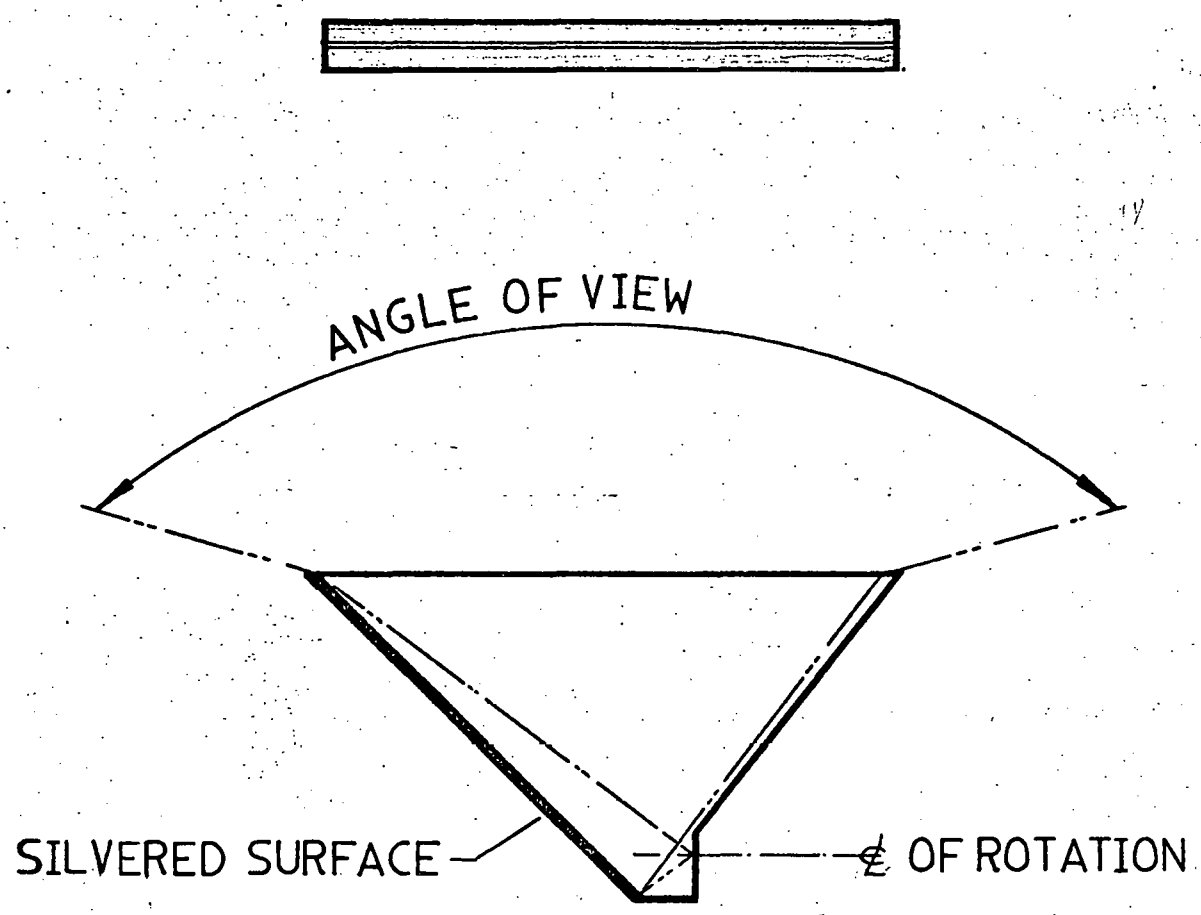


FIG. 4.2-1 PRISM

undesirable feature in the design prompted the search for a prism/lens arrangement which would dispense with their need, but still accomplish the desired increase in field of view, and simultaneously permit the photodiode to be mounted stationary.

After further study, a prism with an inclined mirror surface to redirect the sun's rays was conceived and incorporated into the design. It is clear that this is a superior approach to the need to achieve an increase in field view, as not only does it allow the photodiode to be mounted stationary but also it provides some optical gain.

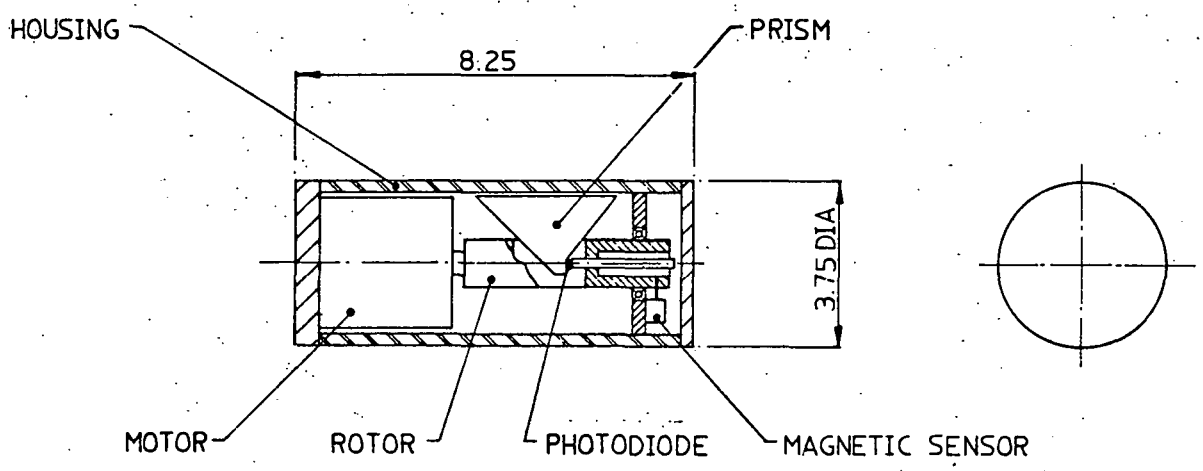
The arrangement used in the final design is illustrated in fig. 4.2-1

Tracking accuracy

In order to attain the desired tracking accuracy it was necessary to improve the quality of signals received from the photo and magnetic sensors on the TRACKING HEAD as compared to those generated by the test device. To achieve this improvement the optimum dimensions of both the rotating optical aperture and the thickness of the armature passing through the magnetic sensors flux field were established, and the values obtained were used in the design, within the constraints imposed by practical manufacturing considerations.

* * * * *

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .xx ± .010 .xxx ± .001 ANGLES ± 1/2° UNLESS OTHERWISE

TITLE: TRACKING HEAD

CUSTOMER: DOE

NEXT ASM | USED ON | PROJ NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO | DWN FG | D 1-6-80 | DWG NO 101

SHT 1 OF 1 | REV

4.3 FINAL DESIGN - TRACKING HEAD

The final design of the TRACKING HEAD is shown diagrammatically on drawing 101.

It will be seen that it is comprised of the following major components:

a. Prism

b. Motor

c. Rotor

d. Photodiode sensor

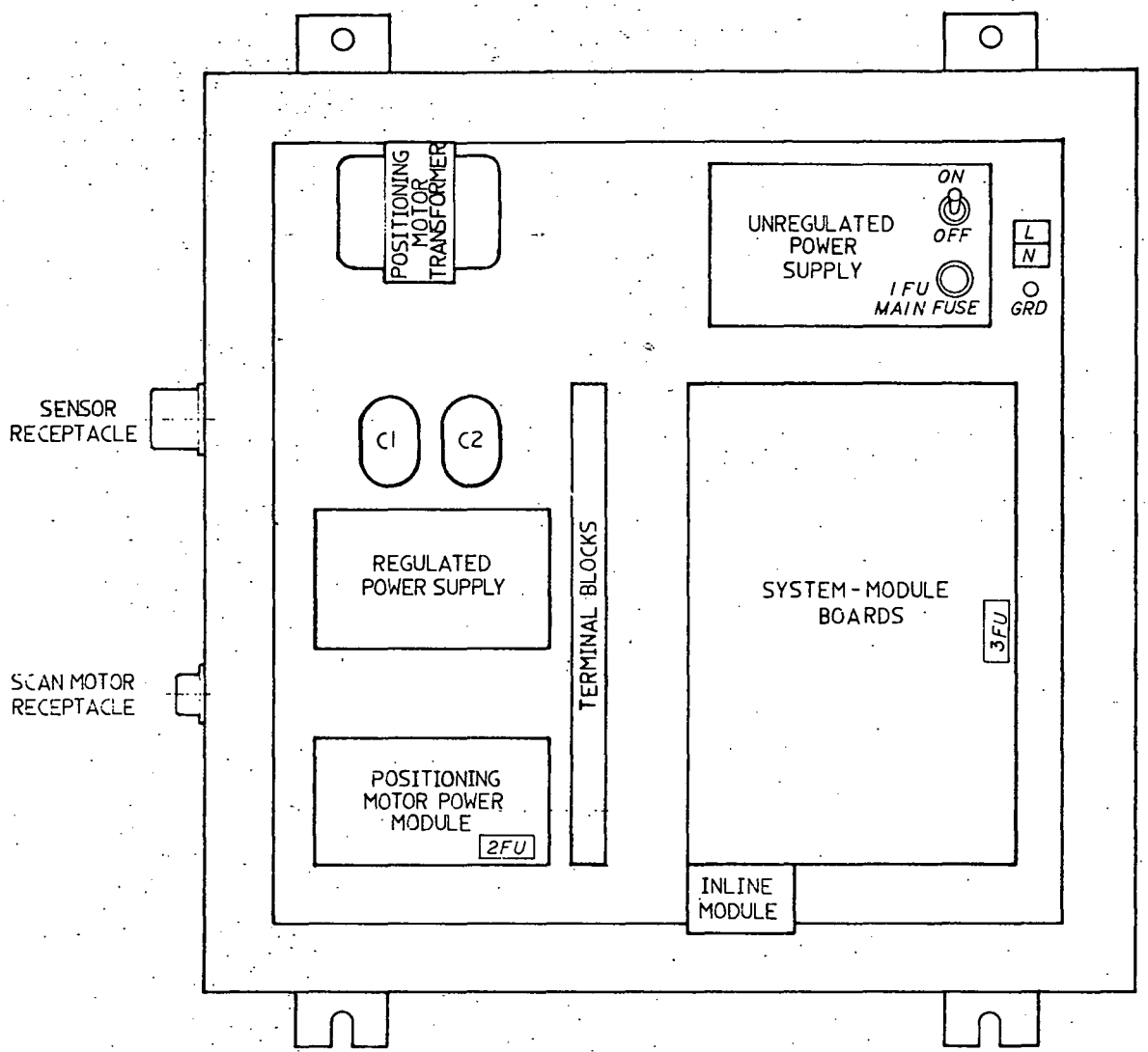
e. Magnetic sensor

f. Housing

The PRISM see Fig 4.2-1 is mounted on the ROTOR which is turned with constant angular velocity by the hysterisis synchronous MOTOR. The PHOTODIODE sensor which is mounted on an adjustable rod is positioned adjacent to the PRISM and receives the sun's rays entering it's apertured edge via an inclined reflective surface. The MAGNETIC SENSOR is mounted on the TRACKING HEAD frame. A soft iron armature attached to the ROTOR cut it's flux field once every MOTOR revolution. The HOUSING is made of acrylic tubing. It provides both a hermetically sealed enclosure for the components and a mechanical support for the sensors.

The signals generated by the PHOTODIODE and MAGNETIC SENSORS are transmitted to the LOCAL CONTROLLER for processing by shielded cable. Both sensors are provided with adjustable mountings to permit optimisation of signals.

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



FUSE		
	1FU	1A – 250V
	2FU	2A – 250V
	3FU	1A – 250V

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± ½° UNLESS OTHERWISE SPECIFIED

TITLE: LOCAL CONTROLLER

CUSTOMER: DOE

NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO | SCALE 1/2F | DRN | CKD | APP | DATE 1-15-80 | ORG NO 102

SHT 1 OF 1 | REV A

4.4 FINAL DESIGN - LOCAL CONTROLLER

4.4.1 GENERAL

The LOCAL CONTROLLER is housed in a NEMA 12 oil - dust resistant enclosure 20 inch high - 20 inch wide - 9 inch deep. See drawing 102.

Input power requirements are 120 volts, 1 , 60 hz. 2 amp maximum. The LOCAL CONTROLLER comprises the following major components;

a. Unregulated power supply - Provides unregulated +5 volts, +12 volts, and -12 volts to the control system.

b. Regulated power supply - Regulates the output from a) to levels acceptable to logic system devices. (See component specifications for further details.)

c. System - Module boards - Both contain logic control devices. The SYSTEM CONTROL BOARD provides the basic control function, which are standard for all applications. The MODULE BOARDS control the inputs and outputs,which may vary depending upon external equipment specified. They plug into the SYSTEM CONTROL BOARD,and both have been designed with sufficient flexibility, to permit the systems use with most concentrator manufactures drive systems.

Modules;

1. SCAN MOTOR MODULE - Provides both the control function and power to the TRACKING HEAD'S scan motor.

2. POSITIONING MOTOR CONTROL MODULE - Provides a low power 120 VAC, forward, and reverse supply for a small motor,

normally a Position Motor Power Module is slaved to this module, to provide drive for much larger motors.

3. SEND - RECEIVE MODULE - Provides for communication with a central controller in installations were dynamic data monitoring and logging plus the ability to change system modes and parameters is required. In installations where these capabilities are not desired the SEND - RECEIVE MODULE is replaced with a STANDALONE MODULE (Future) which provides the system with multiple external inputs, clocks, detectors, instruments, switches, etc.

4. EXPANSION MODULE (Future) - Permits controller to be time shared by a maximum of sixteen TRACKING HEADS.

d. Inline Module - A signal conditioning device for the PHOTODIODE SENSOR signal.

e. Positioning Motor Power Module - Provides power to the CONCENTRATOR POSITIONING MOTOR for forward or reverse operation in response to signals from the POSITIONING MOTOR CONTROL MODULE.

f. C1 - Positioning motor phase shift capacitor.

g. C2 - Scan motor phase shift capacitor.

h. Terminal blocks - Inputs and outputs from SYSTEM - MODULE BOARDS. (For terminal assignments see drawing no. 2D0044.

j. Scan motor receptacle - M.S. receptacle for SCAN MOTOR connection.

k. Sensor receptacle - M.S. receptacle for PHOTODIODE and MAGNETIC SENSOR connections.

4.4.2 SYSTEM - MODULE BOARDS

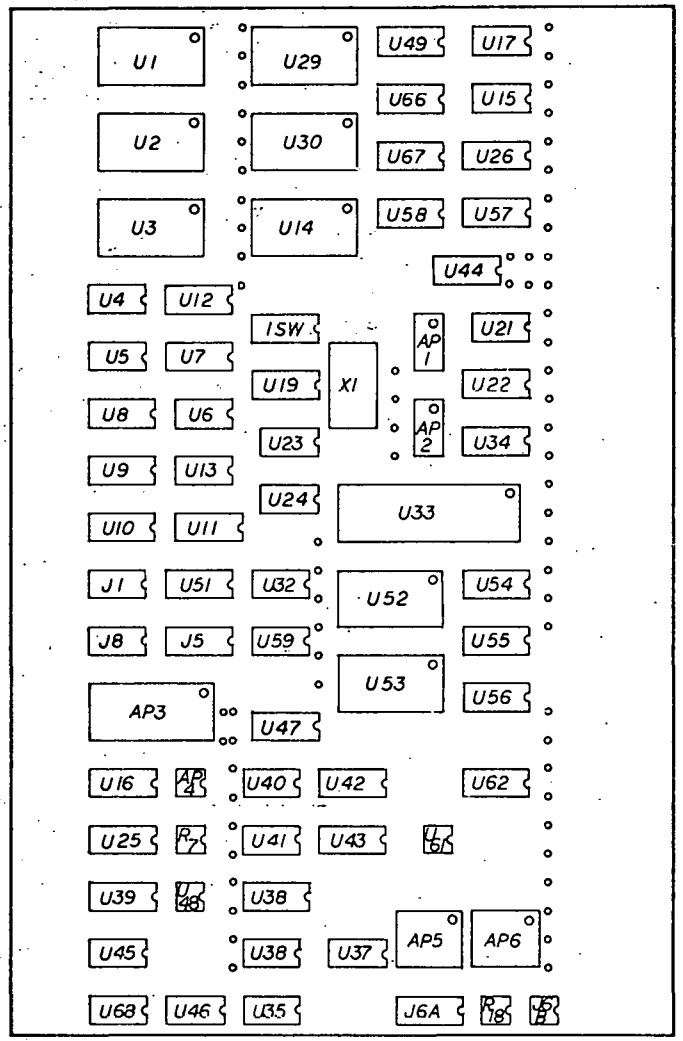
The component layout of the SYSTEM - MODULE BOARDS are shown on drawings No. 103 and 104 respectively. They contain a combination of integrated and discrete devices. The latter are mounted on Adaptor Plugs which are shown on drawings No. 105 and 104. The Component Lists identify the items by number and indicate the part number of the device used. The location of the #1 pin for both integrated devices and adaptor plugs is also tabulated to facilitate component replacement. Location of MODULES BOARDS on the SYSTEM BOARD are given on drawing No. 106. The co-ordinants given are referenced to the SYSTEM BOARD. The appropriate MODULE BOARD pins are listed in the Component List.

4.4.3 INTERCONNECTIONS

The location of plugs connecting external modules to the SYSTEM - MODULE BOARD assembly is shown on drawing No. 2D0044. This drawing also shows the terminal block connections for both internal and external equipment.

4.4.4 SCHEMATIC AND CABLE DIAGRAMS

Schematic diagrams are provided in Section 5.3.



COMPONENT LAYOUT

SYSTEM CONTROL BOARD

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .xx ± .010 .xxx ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED	TITLE SYSTEM CONTROL BOARD	NEXT ASM	GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361
		USED ON	
	CUSTOMER DOE	PROJ NO	JOB NO / DRN / CKD / APP / DATE / DRG NO 103
		REF NO	SCALE / REV A

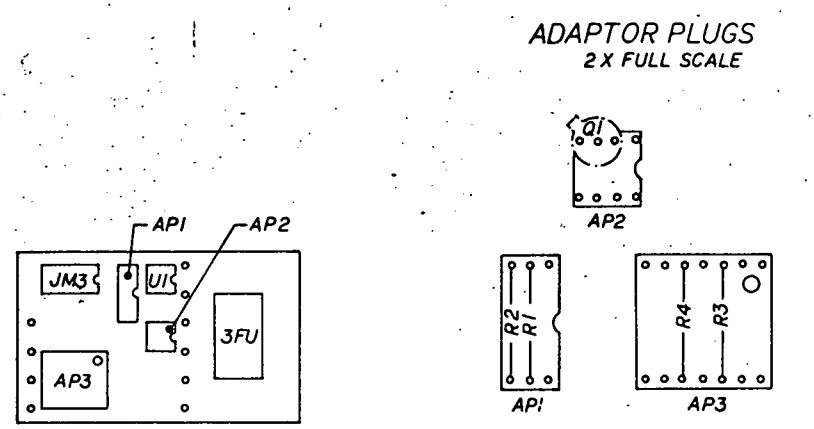
SHT 1 OF 1

22234 PRINTED ON NO. 1000H-10 CLEARPRINT FADE-OUT

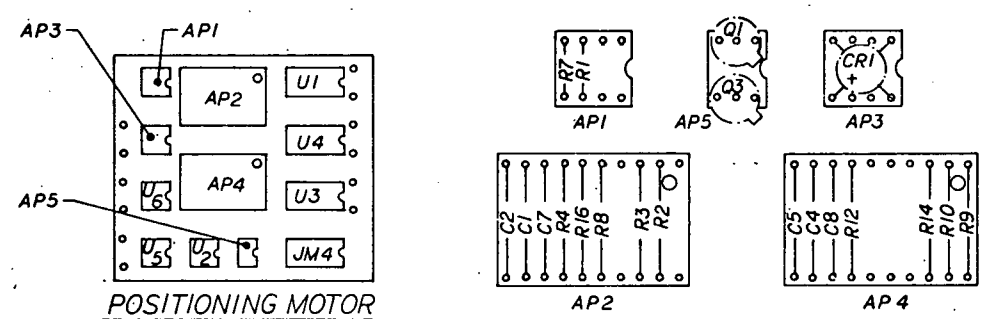
4.4.5 COMPONENT LIST - SYSTEM CONTROL BOARD

Identity	Device	Location	Identity	Device	Location
U1	4508B	AA38	U43	4051B	DB19
U2	4508B	AD38	U44	4042B	AJ07
U3	4508B	AG38	U45	4011B	DF44
U4	4024B	BA44	U46	74LS26	DH36
U5	4024B	BC44	U47	LF311	CG26
U6	4001B	BE35	U48	LF356	DD38
U7	4049B	BC35	U49	4174B	AA13
U8	4585B	BE41	U50	Unassigned	
U9	4585B	BG41	U51	4503B	CB35
U10	4585B	BJ41	U52	2716	CB13
U11	4049B	BJ34	U53	2716	CE13
U12	4027B	BA35	U54	4042B	CB04
U13	4013B	BG35	U55	4042B	CD04
U14	4508B	AG22	U56	4042B	CF04
U15	4002B	AC04	U57	4042B	AG04
U16	4174B	CJ43	U58	4042B	AG13
U17	4011B	AA04	U59	314 Res. 270 OHM	CD27
U18	Unassigned		U60	LF356 (inline module)	
U19	4506B	BD26	U61	LF356	DB12
U20	Unassigned		U62	LF311	CJ04
U21	4011B	BB04	U63	Unassigned	
U22	4538B	BD04	U64	Unassigned	
U23	4025B	BF26	U65	Unassigned	
U24	4013B	BH26	U66	4503B	AC13
U25	4503B	DB43	U67	4503B	AE13
U26	4503B	AE04	U68	4023B	DH44
U27	Unassigned		1SW	8P Rocker Switch	BB26
U28	Unassigned		J1	Jack 1 Power Supply	CB44
U29	4514B	AA22	J5	Jack 5 Switch Input	CD35
U30	4514B	AD22	J6A	Jack 6 index input	DH11
U31	Unassigned		J6B	Jack 6B inline mod.	DJ04*
U32	4001B	CB27	J8	Jack 8 external	CD44
U33	8035	BH05	R7	Potentiometer	DB38
U34	316 Res 10K	BF04	R18	Potentiometer	DH07
U35	4012B	DH28			
U36	4013B	DF28	AP1	Adaptor Plug 1	BB13
U37	4001B	DF19	AP2	Adaptor Plug 2	BE13
U38	4049B	DD27	AP3	Adaptor Plug 3	CF37
U39	4043B	DD43	AP4	Adaptor Plug 4	CJ38
U40	4023B	CJ28	AP5	Adaptor Plug 5	DE11
U41	4013B	DB28	AP6	Adaptor Plug 6	DE03
U42	4051B	CJ19	X1	Crystal	

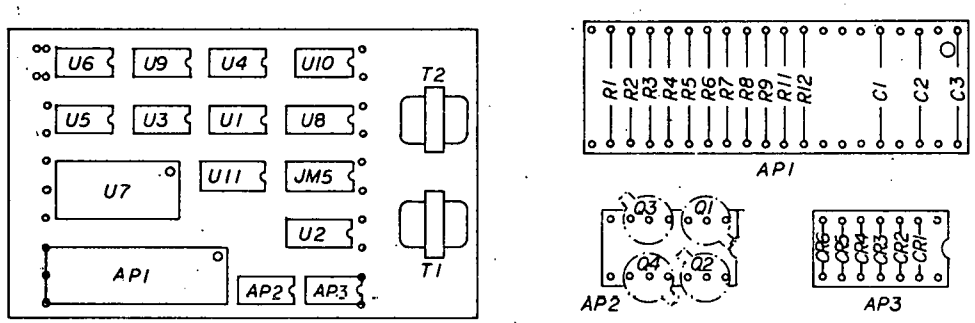
* CAUTION - Reverse orientation



SCAN MOTOR MODULE



POSITIONING MOTOR CONTROL MODULE



SEND-RECIEVE MODULE

COMPONENT LAYOUT
MODULE BOARDS

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .xx ± .010 .xxx ± .001 ANGLES ± ½ UNLESS OTHERWISE SPECIFIED	TITLE MODULE BOARDS	NEXT ASM	GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361	SHT 1 OF 1
	CUSTOMER DOE	USED ON		
		PROJ NO	JOB NO	DRN / CKD / APP / DATE 1-15-81 / DRG NO 104 / REV A
		REF NO	SCALE	

PRINTED ON NO. 1000H-10 CLEARPRINT FADE-OUT

4.4.6 COMPONENT LIST - MODULE BOARDS

Scan Motor Module

Identity	Device	Location
Ref. Pin		BB01
U1	MCS 6200	BB02
AP1	Adaptor Plug #1	BB06
AP2	Adaptor Plug #2	BD02
AP3	Adaptor Plug #3	BE10
JM3	Module Jack #3	BB10
3FU	Fuse 1A	

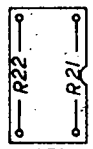
Positioning Motor Control Module

Identity	Device	Location
Ref. Pin		BB20
U1	4001B	BB21
U2	MCS6200	BH34
U3	4011B	BF21
U4	4001B	BD21
U5	MCS6200	BH39
U6	MCS6200	BF39
AP1	Adaptor Plug #1	BB39
AP2	Adaptor Plug #2	BB29
AP3	Adaptor Plug #3	BD39
AP4	Adaptor Plug #4	BE29
AP5	Adaptor Plug #5	BH29
JM4	Module Jack #4	BH21

Send - Receive Module

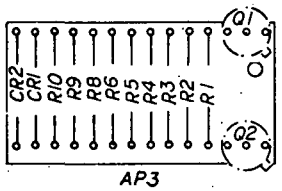
Identity	Device	Location
Ref. Pin		AA1
U1	LM741	AC11
U2	LF311	AG02
U3	4013B	AC19
U4	4001B	AA11
U5	4070B	AC27
U6	4013B	AA27
U7	2651	AE20
U8	4027B	AC02
U9	4042B	AA19
U10	4001B	AA02
U11	4042B	AE11
T1	Transformer	
T2	Transformer	
AP1	Adaptor Plug #1	AH15
AP2	Adaptor Plug #2	AJ08
AP3	Adaptor Plug #3	AJ01
JM5	Module Jack #5	AE02

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			

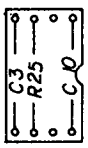


AP1

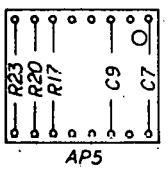
AP2



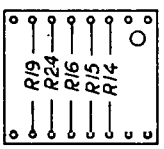
AP3



AP4



AP5



AP6

ADAPTOR PLUGS FOR
SYSTEM CONTROL BOARD
2 X FULL SCALE

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied with out written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± ½° UNLESS OTHERWISE SPECIFIED

TITLE: ADAPTOR PLUGS

CUSTOMER: DOE

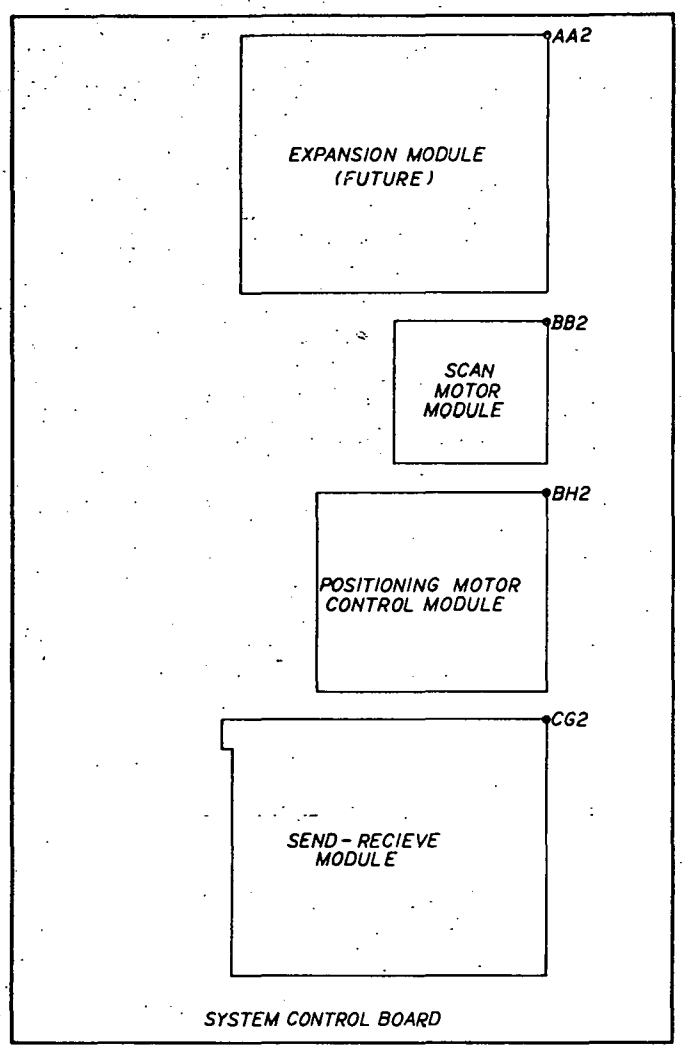
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO | SCALE | DRN | CKD | APP | DATE 1-16-81 | DRG NO 105

SHT 1 OF 1 | REV A

22534 PRINTED ON NO. 1000H-10 CLEARPRINT FADE-OUT



LOCATION OF MODULES

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED	TITLE LOCATION OF MODULES ON SYSTEM CONTROL BOARD	NEXT ASM	GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361	SHT 1 OF 1
	CUSTOMER DOE	USED ON	JOB NO. / SCALE / DRN / CKD / APP / DATE 1-16-81	DWG NO 106 / REV A
		PROJ NO / REF NO		

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

PRINTED ON NO. 1000R-10 CLEARPRINT FADE-OUT

5.0 CIRCUIT DESCRIPTIONS

5.1 BASIC OPERATION: Re: Fig. 5.1-1

The angular measurement of the Sun relative to the tracker's axis is basically determined from the subtraction of a counter value obtained at the moment of peak insolation from the counter value at the time the optics are inline with the trackers reference axis. This subtraction is necessary since the counter value is only relative to the instantaneous optical position. Once operating this phase angle is fixed, but it will change each time the system scan motor is reactivated. Counter overflow corresponds to one revolution of the optics. When in the tracking mode of operation the positioning motor is ran until the error angle has decreased until it is within the tracking window.

Counter values are latched in AL1 and AL2, where AL1 contains the index value, and AL2 the position at which Peak insolation occurs.

```
                                                  +-------------+
+---------------+        Data Buss                ! POSITIONING !
! MICROCOMPUTOR !----*--------------*------->! MOTOR       !--> MOTOR
+---------------+    ^              ^           ! DRIVE       !
                     !              !           +-------------+
                +-----------+       !
Index Strobe-->! AL1 Latch !        !
                +-----------+       !
                     ^        +-----------+
Peak Strobe----------!------->! AL2 Latch !
                     !        +-----------+
                     !              ^
                     !              !
                      --------------*
                                    !
                              +-----------+
                +------+      !  SCAN     !
                ! OSC. !------>!  ADDRESS  !
                +------+      !  COUNTER  !
                              +-----------+
```

Fig. 5.1.-1 BASIC OPERATION

5.2 FUNCTIONAL CONSIDERATIONS: Re: Dwg. 1D0018

Many other functions must be performed to implement this strategy. The following is a description of the circuit blocks shown in Dwg. 1D0018.

5.2.1 SCAN ADDRESS COUNTER & ADDRESS LATCHES Re: Dwg. 1C0022

Contains the scan address counter, and latches AL1 and AL2. The essentials have been described in the Basic Operation Section.

5.2.2 ACTIVE/PASSIVE Re: Dwg. 1C0020

There are two basic machine states. In the Active mode data acquistion is performed, wherein the address latches are loaded. The passive mode disables this function, and the data contained in the latches is processed. Also in this mode the Analog to Digital Conversion may be performed on the Peak Insolation Level contained in the Sample Hold Capacitor. The transition between these states occurs when the counter value equals the comparison register value, and the Cycle Enable is True. When a 360 deg. scan is desired, the comparison register is normally set to the index value +180 degs. The first comparison will set the active state, and the second will set the passive state. Other scan limits are similarily possible by setting any comparison register value, and after the first comparison (entering the Active state), loading another value, at which the transition to the Passive state is to be made.

5.2.3 PROM MEMORY Re: Dwg. 1C0024

The Prom Memory block includes the proms, address latches, and has contained in it the Comparison register outputs, and Status Register inputs.

5.2.4 SWITCH TERMINATIONS Re: Dwg. 1C0026

Provides pull up and pull down switch input terminations.

5.2.5 SWITCH STATUS REGISTER Re: Dwg. 1C0025

This is the input register for all Unit #0 switches.

5.2.6 MOTOR CONTROL REGISTER Re: Dwg. 1C0030

Output register for Unit #0 motors including ON/OFF, and direction of rotation.

5.2.7 POSITION MOTOR CONTROL MODULE Re: Dwg. 1C0035

Converts the digital outputs to the positioning motor into low power AC.

5.2.8 POSITION MOTOR POWER MODULE Re: Dwg. 1C0035

Converts low power positioning motor AC drives into a high power drives.

5.2.9 SCAN MOTOR MODULE Re: Dwg. 1C0035

Converts a digital output to the Scan motor into low power AC.

5.2.10 PHOTODIODE and INDEX AMPS. Re: Dwg. 1C0028

The Photo diode amp is an inline transconductance amp, and the Index amp is a voltage amplifier.

5.2.11 ANALOG SELECT and SAMPLE HOLD Re: Dwg. 1C0027

Contains the circuitry to perform analog input selection, Peak detection, Zero detection, and performs the actual analog to digital conversion, for the sampled, or peak signal, held in the sample hold capacitor.

5.5.12 SIGNAL SYNCHRONIZATION & ADC CONTROL Re: Dwg. 1C0029

Synchronizes asynchronous digital signals with the system clock, and has the control circuitry for the analog to digital converter.

5.2.13 UNIT ADDRESS and LOCAL SELECT Re: Dwg. 1C0037

Permits selection of 1 of 256 local unit addresses. The address is read by the microcomputor for local unit selection when used with a Send/Receive Module. Local Select decodes I/O unit address #0 to select onboard rather than expansion circuitry.

5.2.14 COMMAND DECODE Re: Dwg. 1C0021

Performs address decoding for mapped I/O operation.

5.2.15 MICROCOMPUTOR Re: Dwg. 1C0021

Contains the crystal, microcomputor, and watch dog timer for system control.

5.2.16 SEND/RECEIVE MODULE Re: Dwg. 1C0032, Fig. 5.2-1,2

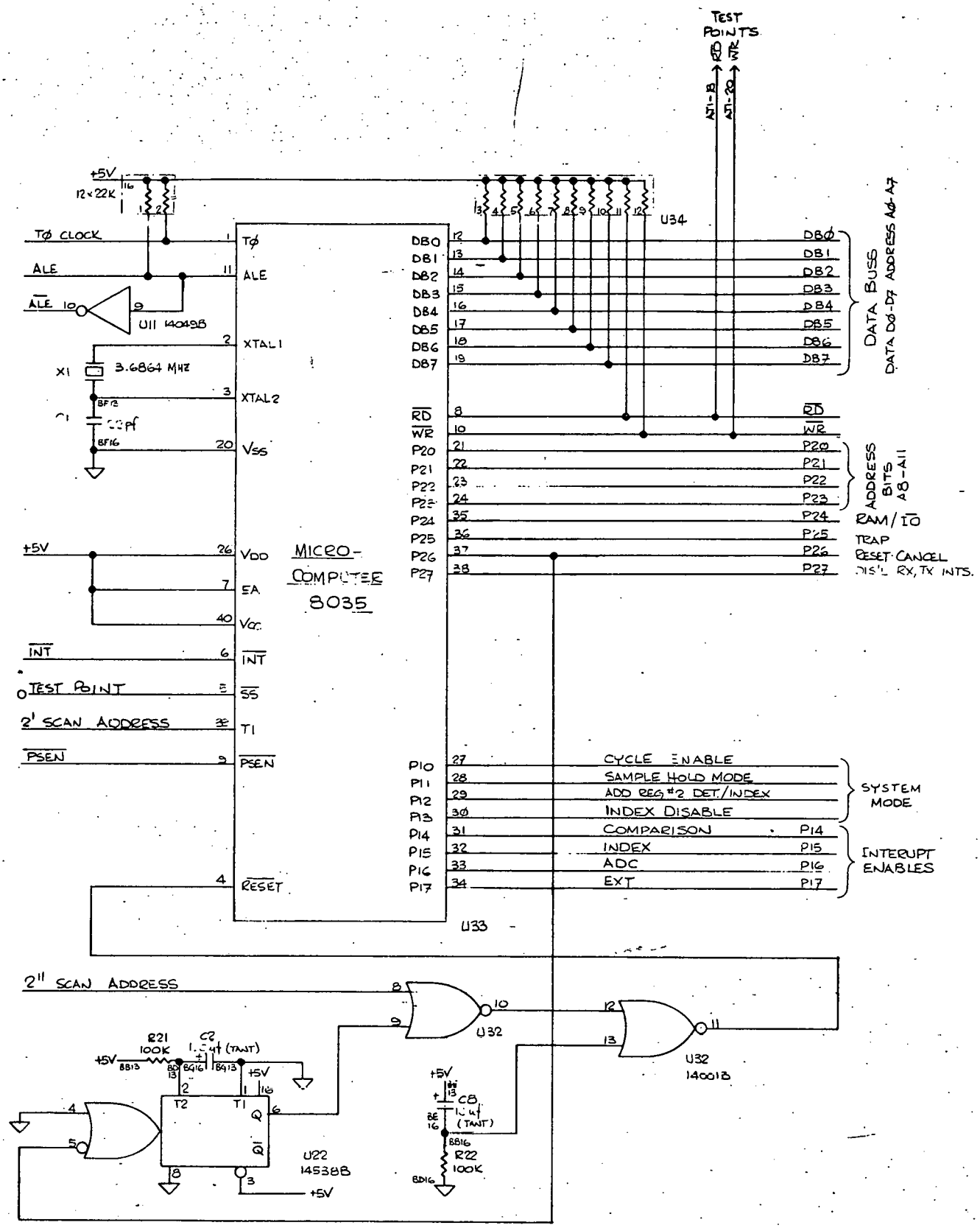
Provides synchrounous comunications with a central controller. A 2DB Chebyshev filter was chosen with D=.895, and F=1635.84 Hz, having a 3Db attenuation at 1920 Hz, and a high pass filter with F(-3Db)=480Hz, for a 60 Hz rejection of -18Db. The combined filter has an approximate constand delay of about 184 microseconds, or 35.33% for a 1920 Hz signal.

5.3 ELECTRICAL DIAGRAMS

Below is a list of electrical diagrams included in this section.

	BLOCK DIAGRAM	
100018	D	"Point" Block Diagram
	SCHEMATIC DIAGRAMS	
1C0019	D	Microcomputer
1C0020	D	Prom memory
1C0021	E	Command decode
1C0022	D	Scan address counter & latches
1C0023	D	Active - passive
1C0024	E	Status register & interupts
1C0025	D	Switch status register
1C0026	B	Switch terminations
1C0027	E	Analog select & sample hold
1C0028	F	Photodiode & indexamplifiers
1C0029	D	Signal sync. & A.D.C. control
1C0030	D	Motor control register
1C0031	D	Power supplies
1C0032	E	Send/Receive module
1C0033	B	Scan motor module (option #1)
1C0035	D	Positioning motor module
1C0037	A	Unit address & local select
1C0040	B	Unregulated power supply
	CABLE DIAGRAMS	
1C0042	B	Sensor cable
1C0043	B	Motor cable

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		[illegible]
B	U50-U66-U67-68 ADDED	[illegible]		[illegible]
C	GENERALLY REVISED	[illegible]		[illegible]
D	GENERALLY REVISED	[illegible]		[illegible]



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied with or written permission it must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ± .010, ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: MICRO-COMPUTER

CUSTOMER: D.O.E

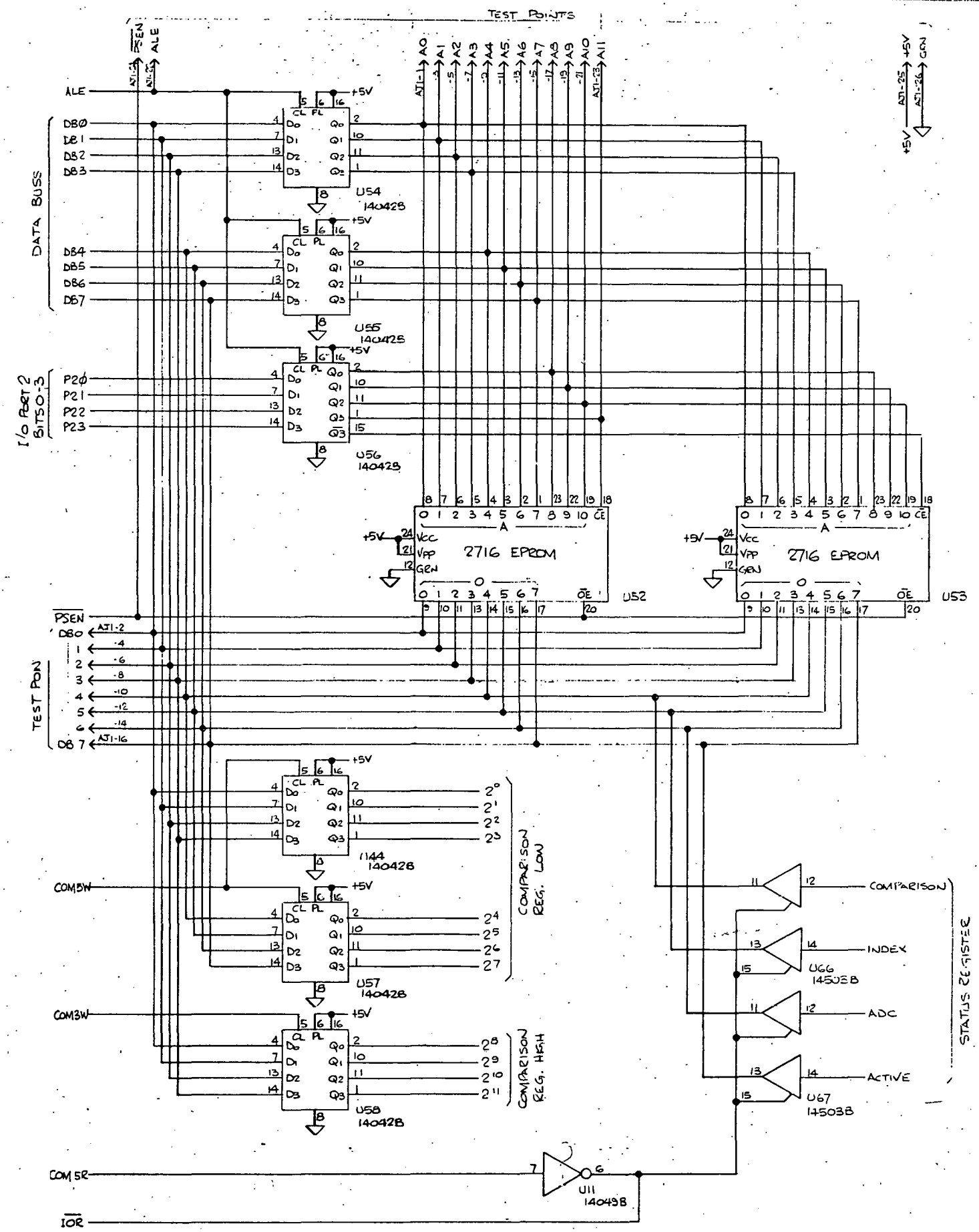
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN | CKD | APP | DATE 5/1/84 | DRG NO 1C0019 | SHT 1 OF 1 | REV 1

SCALE NONE

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
D	[illegible]			



This information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ± .010, ANGLES ± ½°, UNLESS OTHERWISE SPECIFIED

TITLE: PROM MEMORY

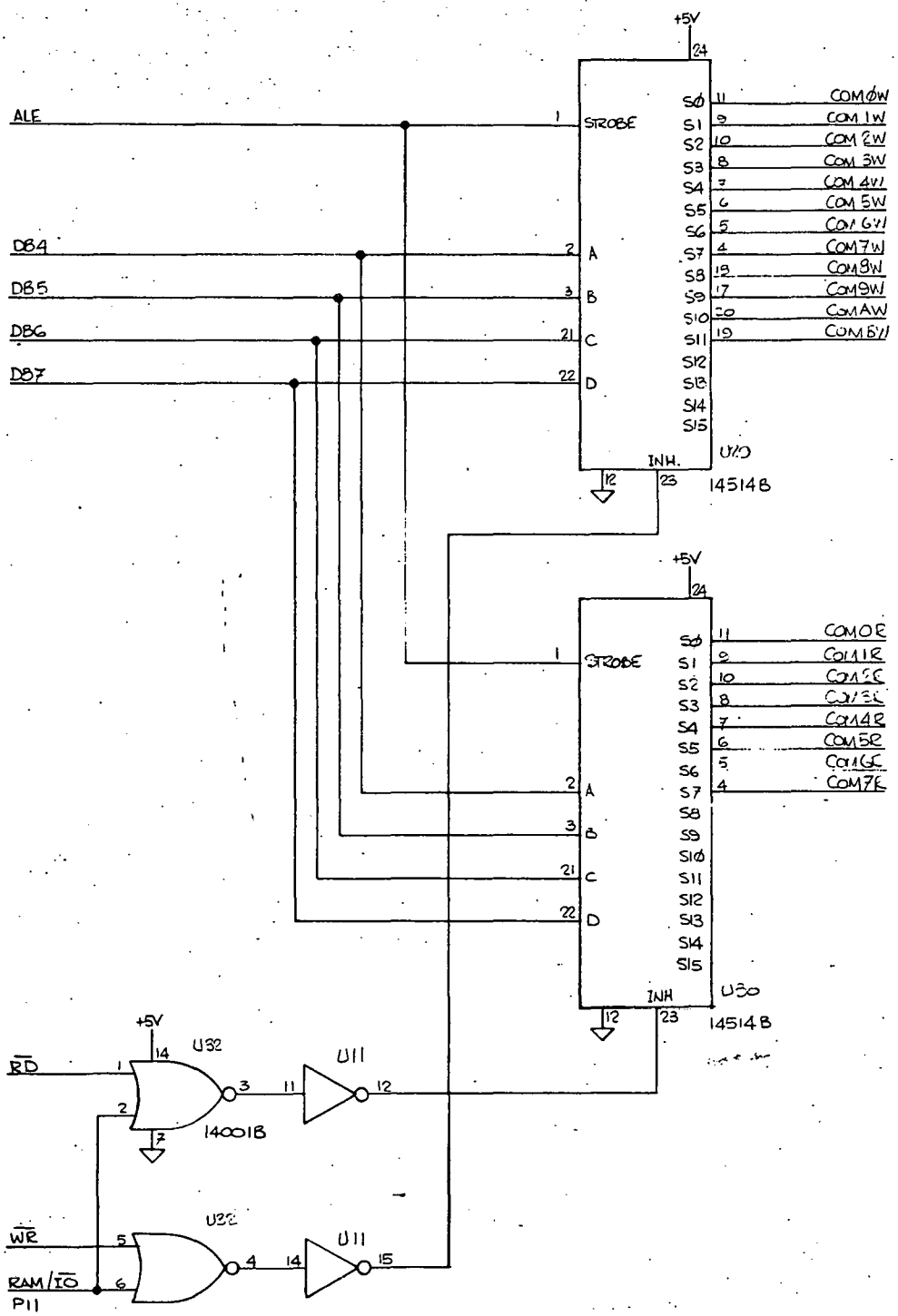
CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO: D100 | SCALE: NONE | DRN | CKD | APP | DATE | DRG NO. 1C0020 | REV D | SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
C	REDESIGNED - REDRAWN			
D	REVISED			
E	REVISED			



The information contained herein is the property of GK Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: COMMAND DECODE

CUSTOMER: DOE

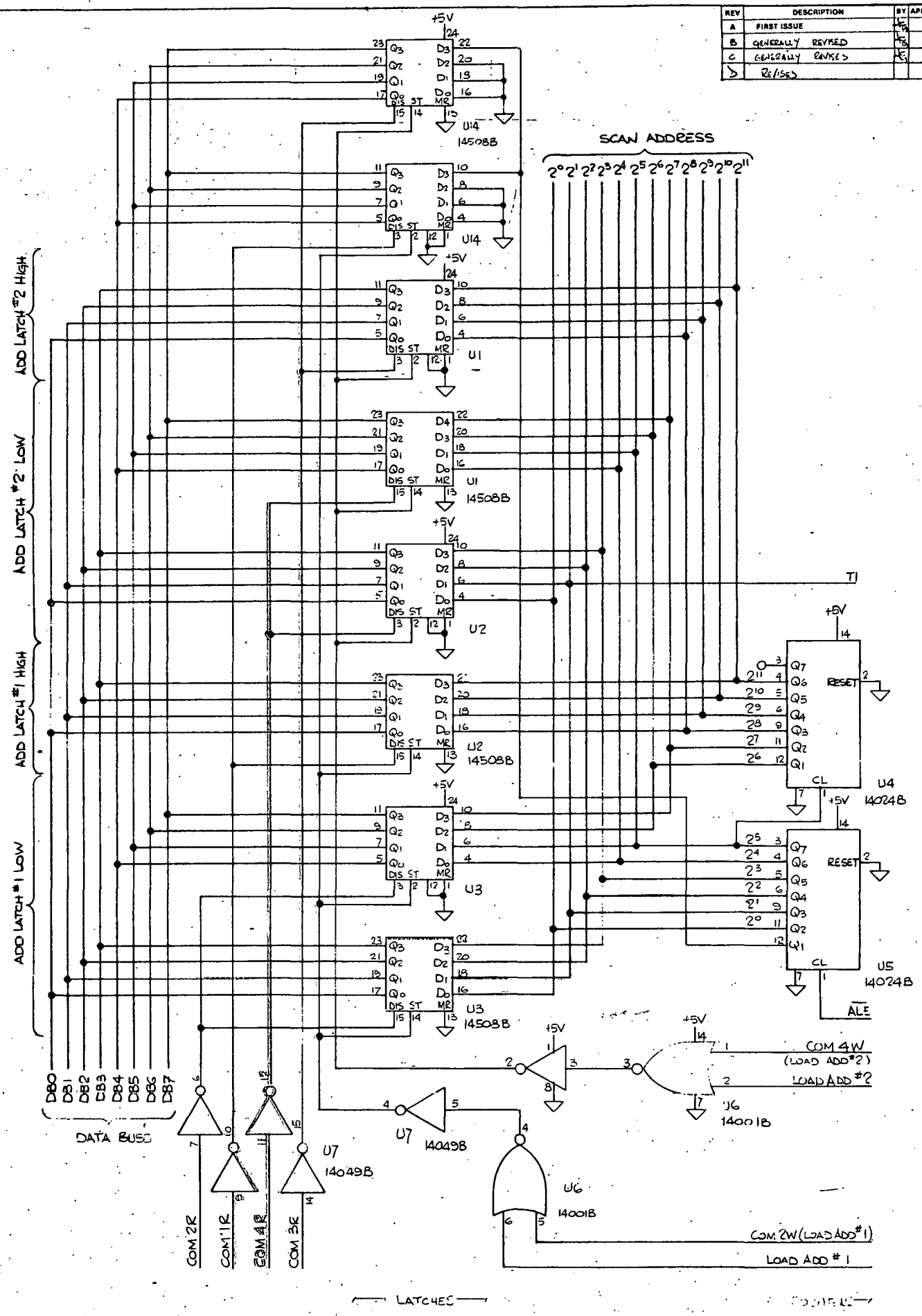
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO. 0100 | DRN | CKD | APP | DATE | DWG NO. 1C0021 | SHT 1 OF 1 | REV E

SCALE NONE

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	GENERALLY REVISED			4/21/80
C	GENERALLY REVISED			12/1/82
D	REVISED			12/1/80



This information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: SCAN ADDRESS COUNTER & SCAN ADDRESS LATCHES

CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

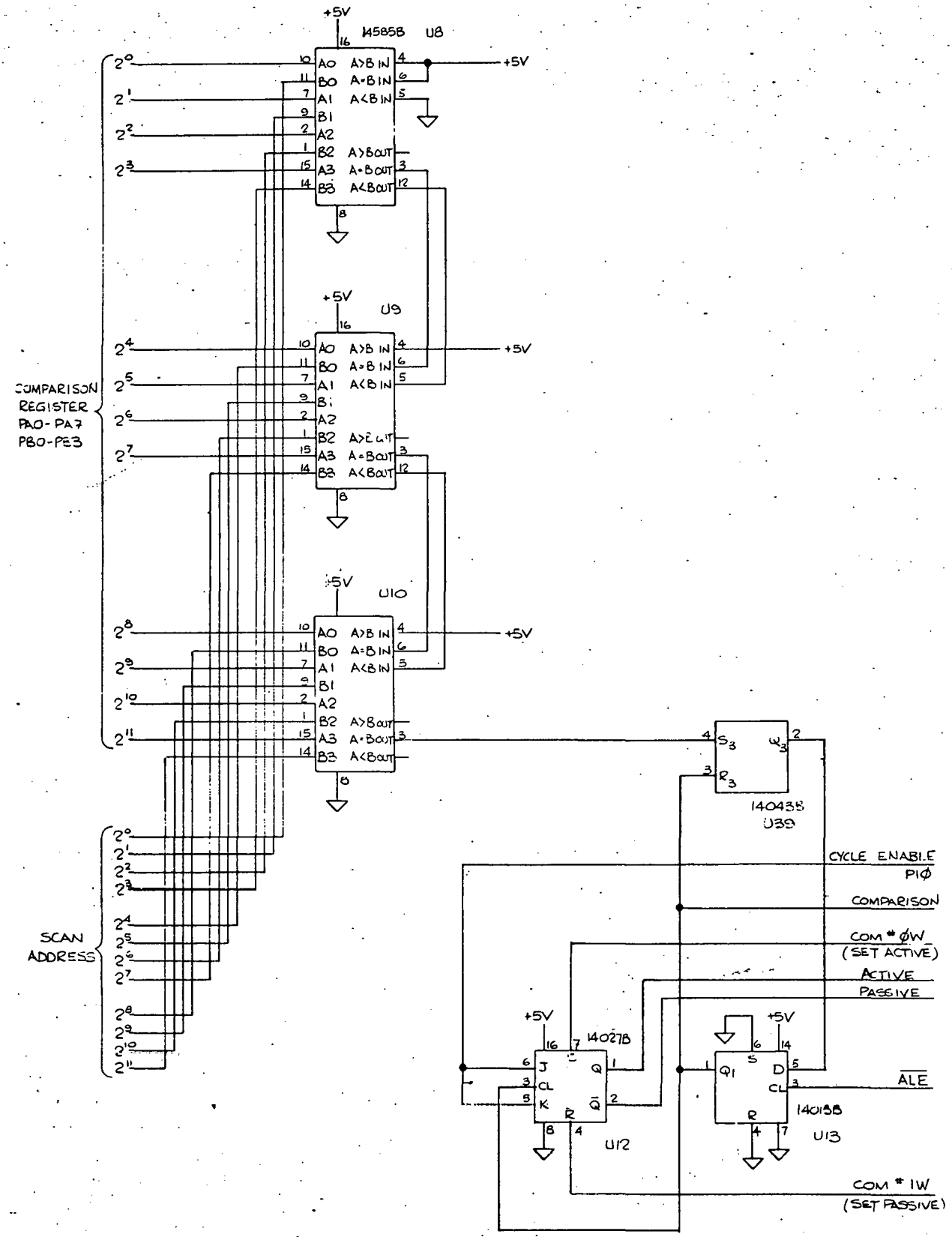
JOB NO: 0100 | SCALE: NONE | DRN | CKD | APP | DATE 3/1/80

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

DWG NO: 1C0022

SHT 1 OF 1 | REV 1.1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	FG		3/19/80
B	COM#1W WAS COM#1	FG		3/28/80
C	GENERALLY REVISED	FG		4/21/80
D	REVISED	FG		1/5/81



The information contained herein is the property of GH Systems Inc. The drawing may not be used or copied without written permission or it must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .xx ± .010 .xxx ± .001 ANGLES ± ½° UNLESS OTHERWISE SPECIFIED

TITLE: ACTIVE - PASSIVE

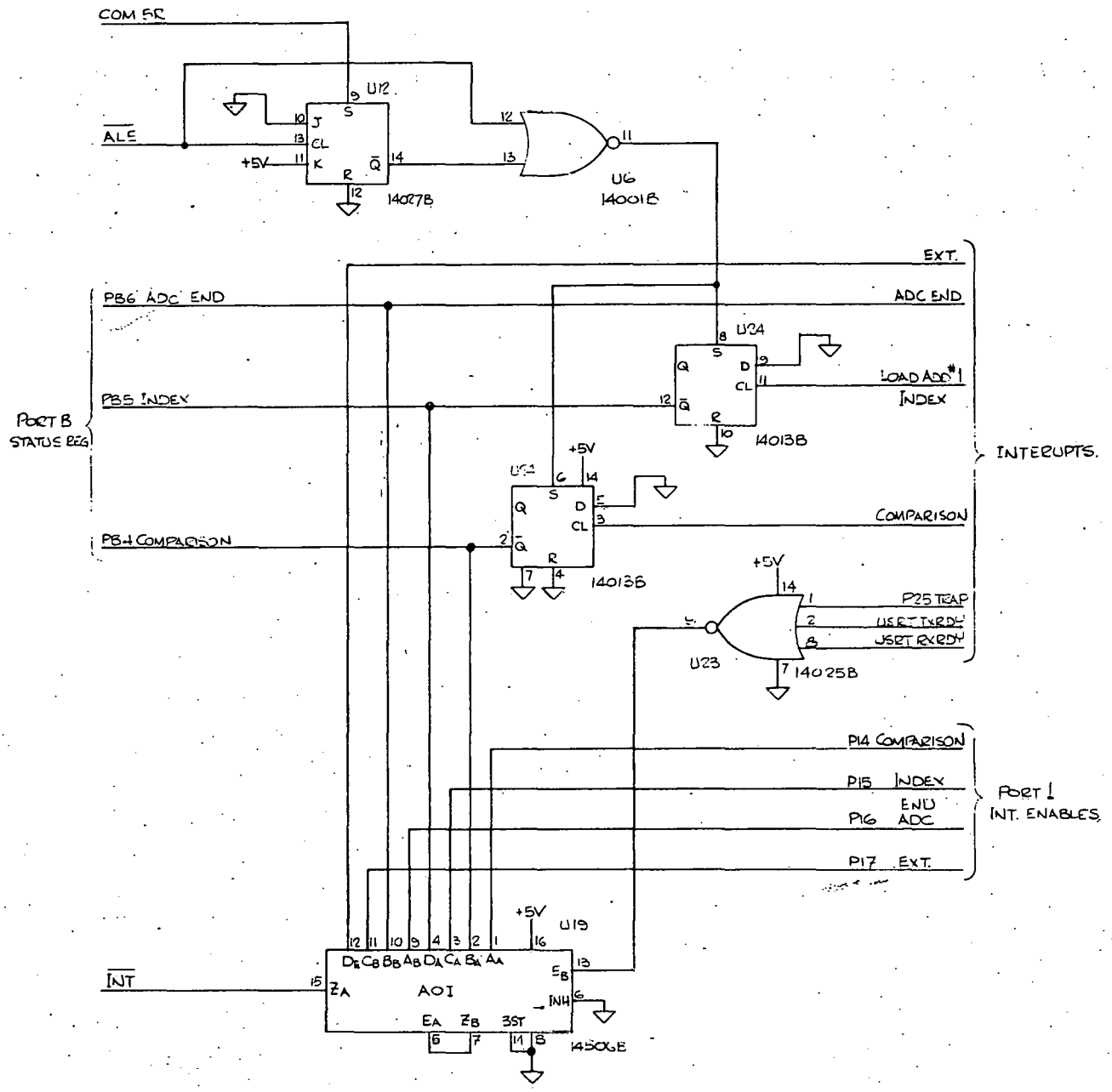
CUSTOMER: D.O.E

HEAT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | SCALE NONE | DRN | CKD | APP | DATE 3/19/80 | DRG NO 10.0023 | SHT 1 OF 1 | REV D

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		3/7/80
C	REDESIGNED & REDRAWN.	[illegible]		4/2/80
D	REVISED	[illegible]		10/1/80
E	REVISED	[illegible]		12/18/80



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS ± .010 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED

TITLE: STATUS REGISTER INTERUPTS

CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

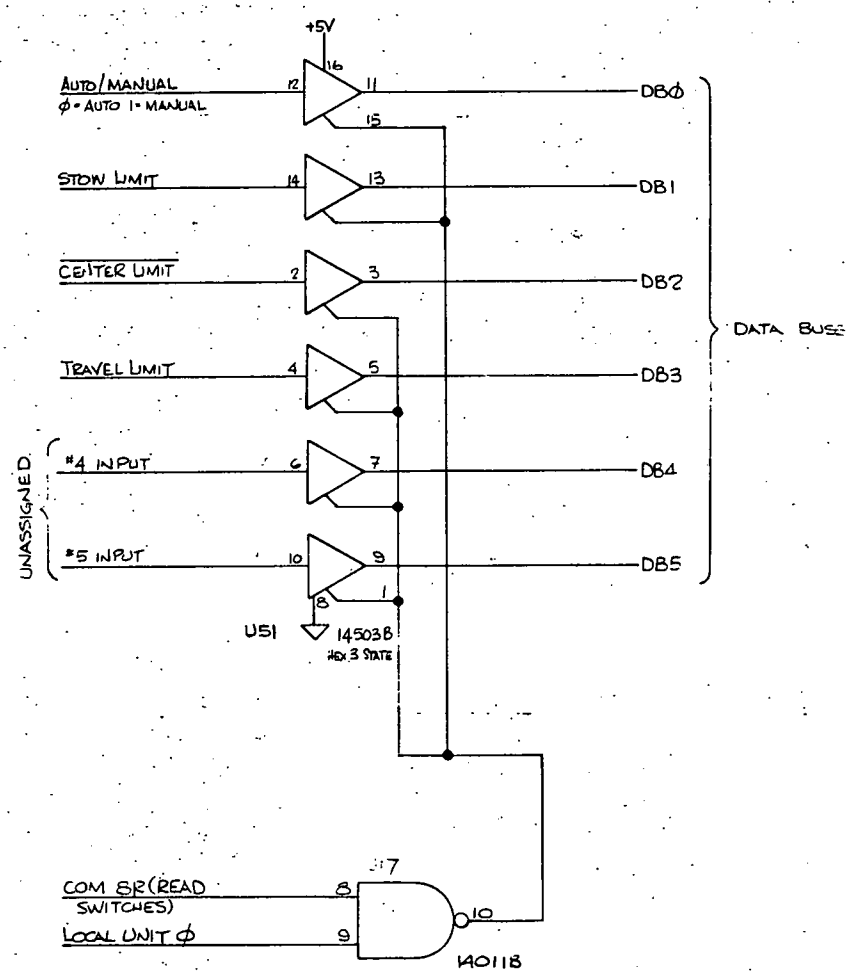
GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN | CKD | APP | DATE | DRG NO IC0024 | REV F.

SCALE NONE

SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		[illegible]
B	GENERALLY REVISED	[illegible]		[illegible]
C	GENERALLY REVISED	[illegible]		[illegible]
D	REVISED	[illegible]		[illegible]



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned up in request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ± .010, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: SWITCH STATUS REGISTER

CUSTOMER: D.O.E

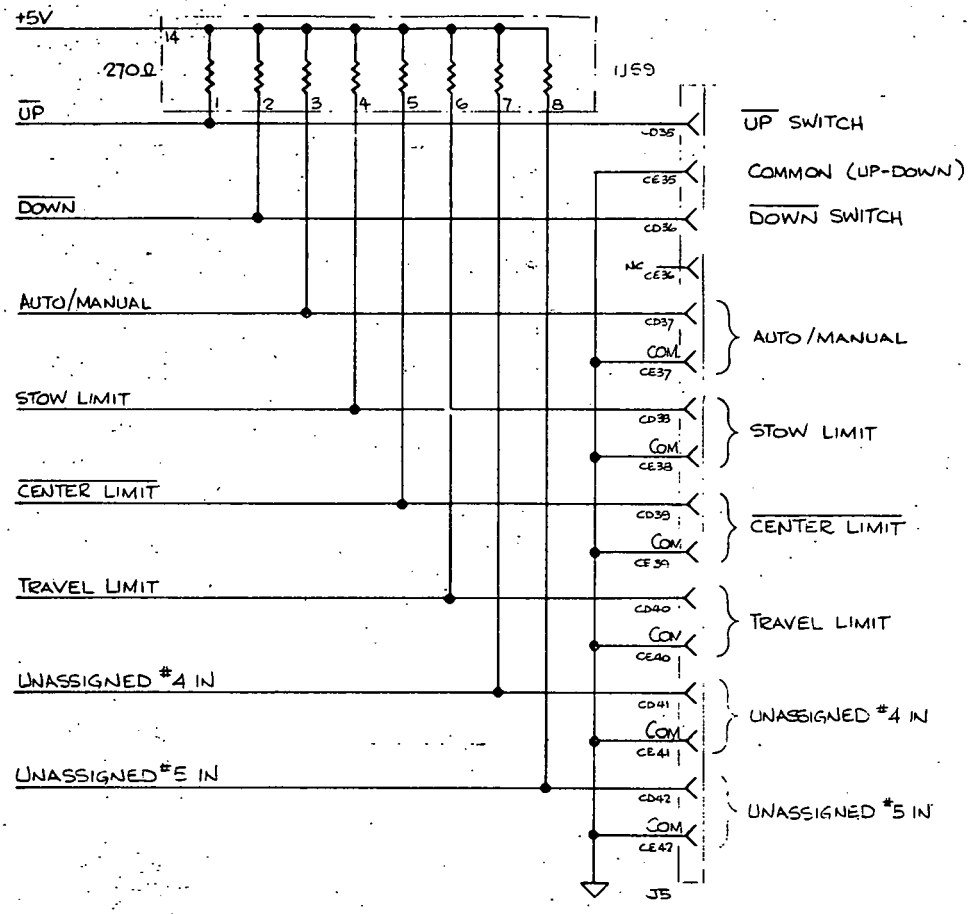
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN | CKD | APP | DATE | DWG NO 1C0025 | SHT 1 OF 1 | REV D

SCALE NONE

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		5/19/[illegible]
B	GENERALLY REVISED.	[illegible]		5/20/[illegible]



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: SWITCH TERMINATIONS

CUSTOMER: D.O.E

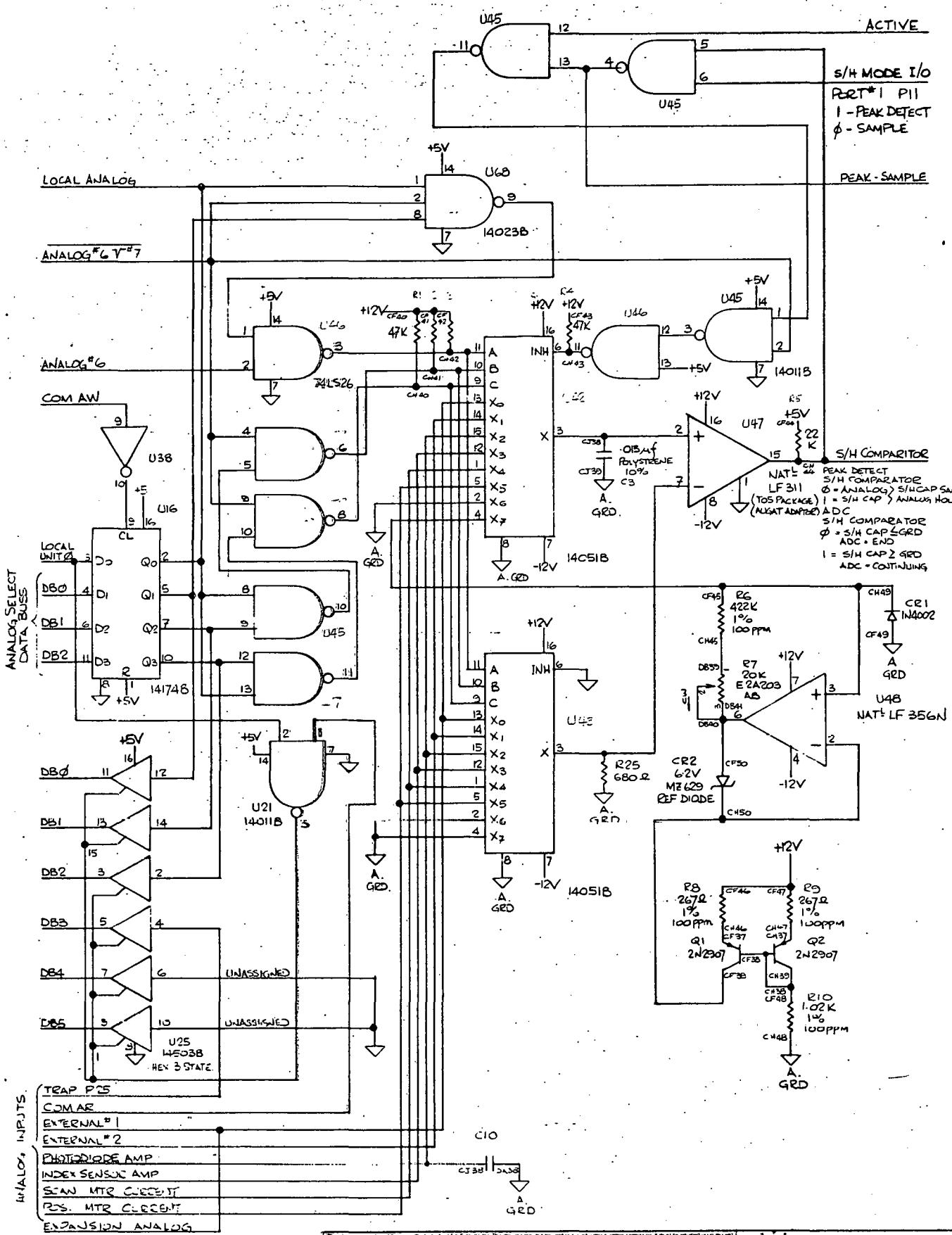
NEXT ASM / USED ON / PROJ NO / REF NO

GH Systems, Inc.
WESTLAKE VILLAGE CALIFORNIA 91361

JOB NO 0100 | DRN JRC | DATE 37-7-[illegible] | DRG NO 1C0026 | SHT 1 OF 1 | REV B

SCALE NONE | CKD | APP

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		[illegible]
B	GENERALLY REVISED	[illegible]		[illegible]
C	GENERALLY REVISED	[illegible]		[illegible]
D	REVISED	[illegible]		[illegible]
E	REVISED	[illegible]		[illegible]



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: ANALOG SELECT & SAMPLE HOLD

CUSTOMER: D.O.E

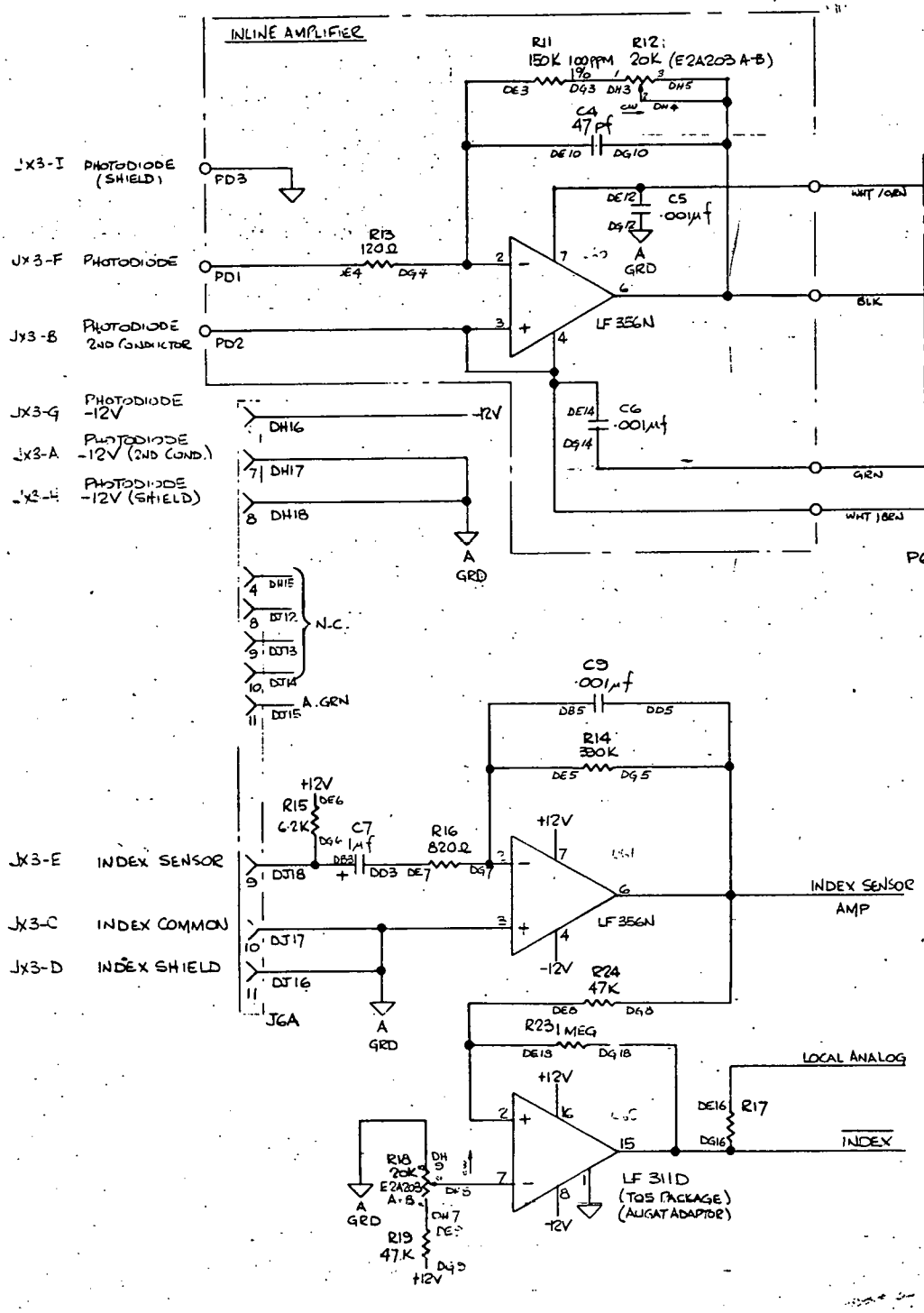
NEXT ASM / USED ON / PROJ NO / REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN FG | DATE 5/1/80 | DWG NO 1C0027 | SHT 1 OF 1 | REV E

SCALE NONE | CKD | APP

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	INDEX WAS INDEX			
C	REVISED			
D	REVISED			
E	REVISED			
F	REVISED			



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED

TITLE: PHOTODIODE + INDEX AMPS.

CUSTOMER: D.O.E

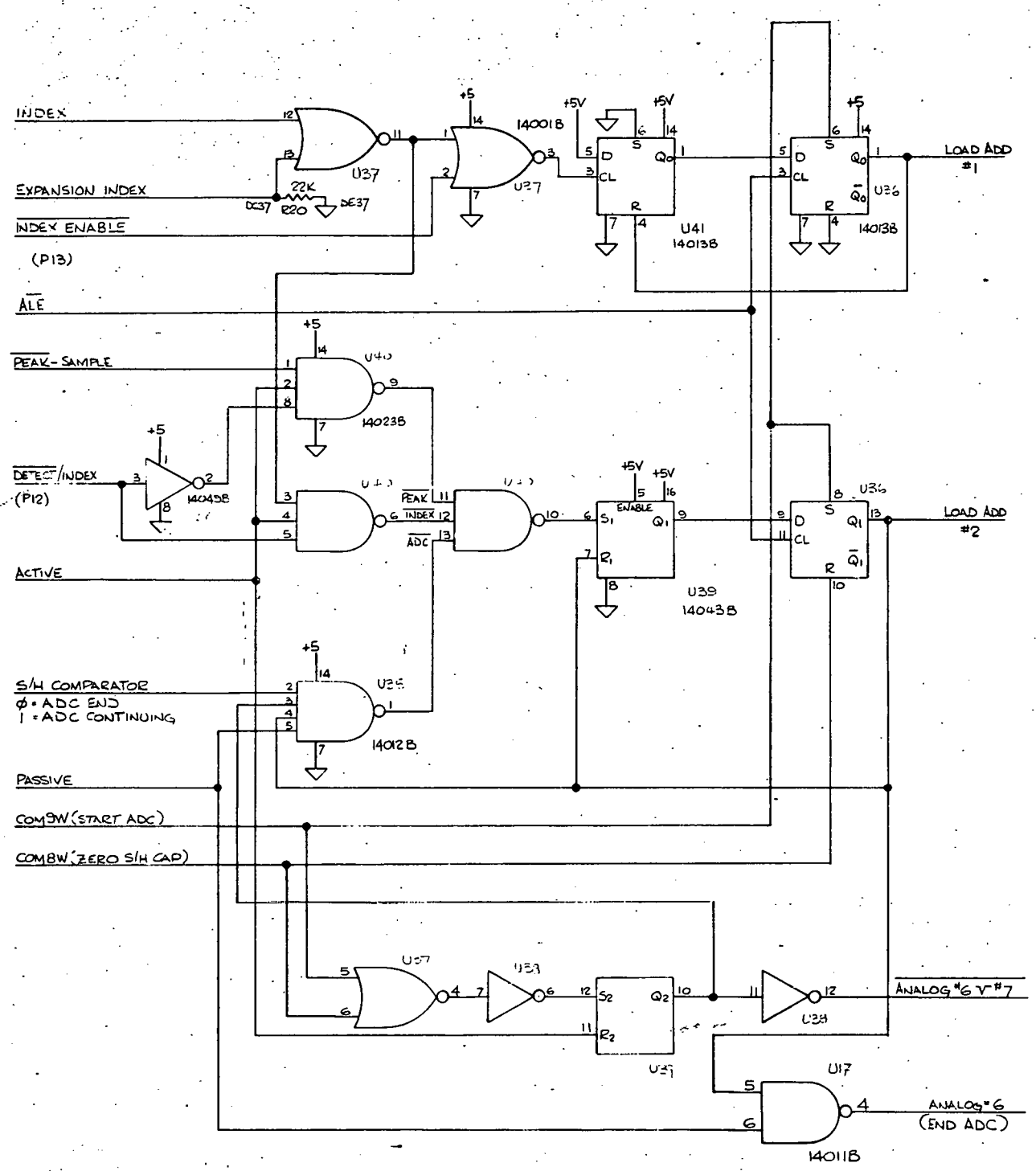
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | SCALE NONE | DRN | CKD | APP | DATE

DWG NO IC 0028 | REV F | SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		[illegible]
B	GENERALLY REVISED	[illegible]		3/21/80
C	GENERALLY REVISED	[illegible]		4/21/80
D	REVISED	[illegible]		[illegible]



The information contained herein is the property of GK Systems Inc. This drawing may not be used or copied with or written permission or must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: SIGNAL SYNCHRONIZATION & ADC CONTROL

CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

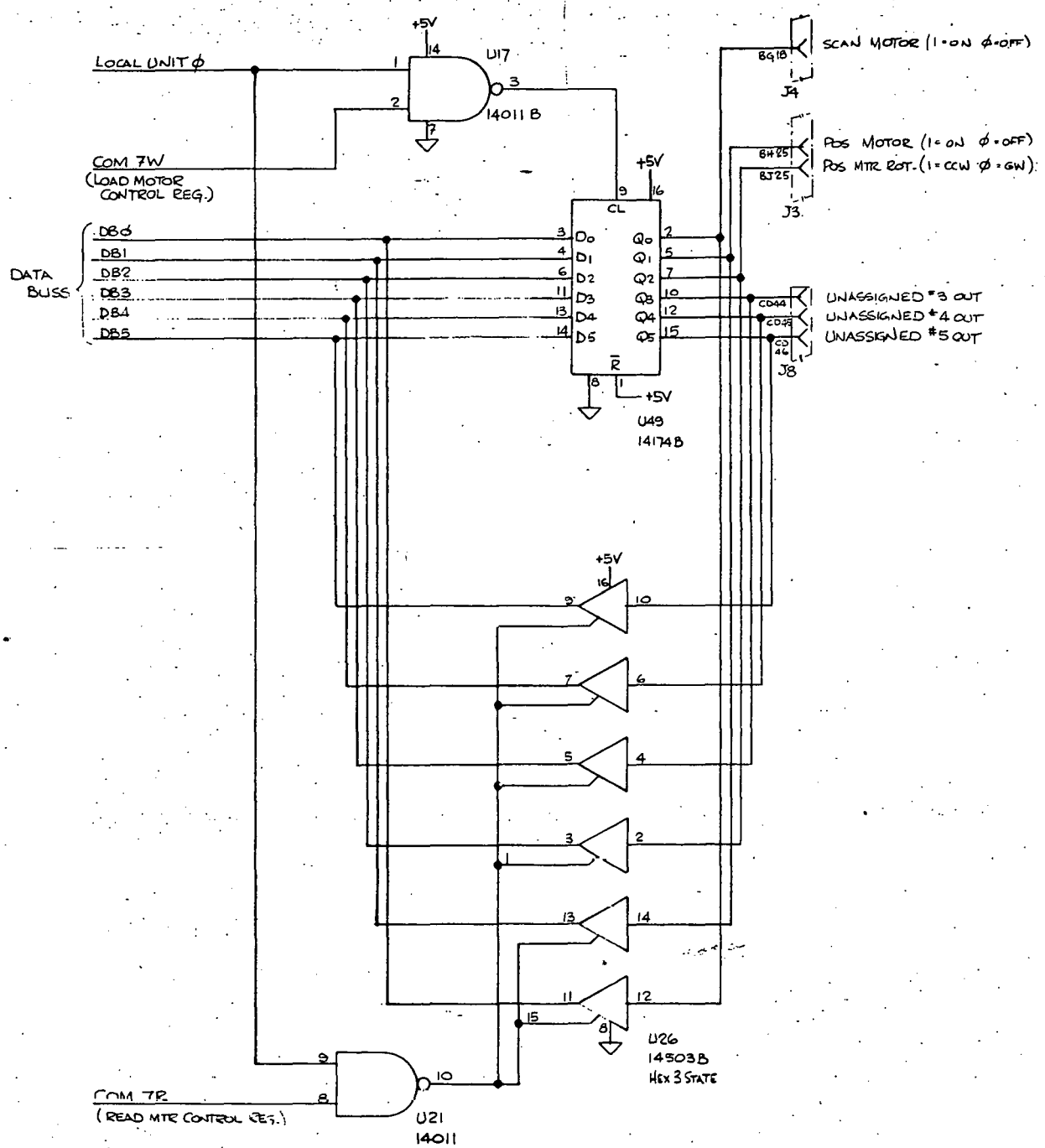
GK Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN [illegible] | CKD | APP | DATE | DRG NO 1C 0029 | REV D

SCALE NONE

SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		3/4/80
B	GENERALLY REVISED	[illegible]		3/31/80
C	GENERALLY REVISED	[illegible]		4/21/80
D	GENERALLY REVISED	[illegible]		11/6/80



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED

TITLE: MOTOR CONTROL REGISTER

CUSTOMER: D.O.E

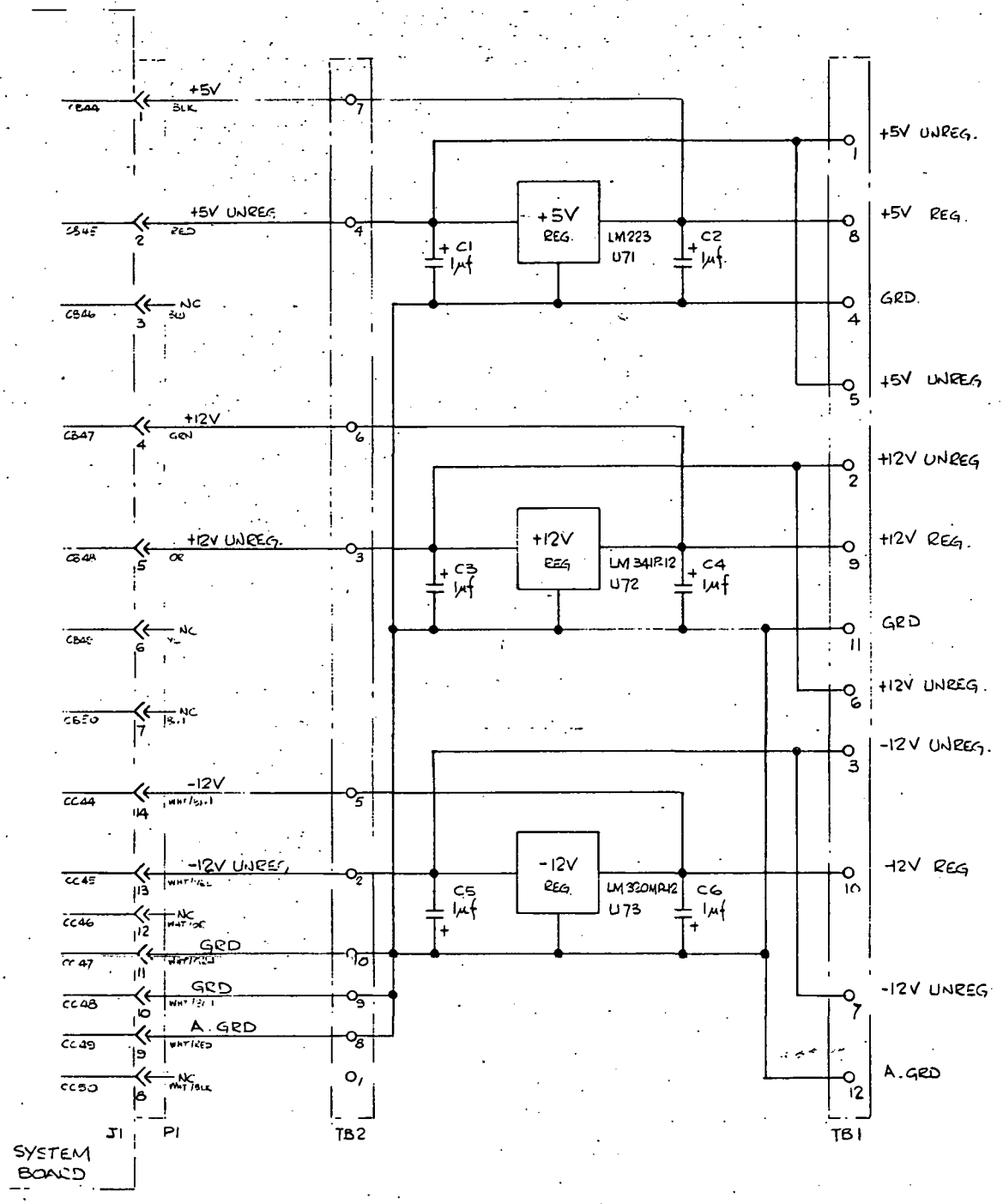
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE CALIFORNIA 91361

JOB NO 0100 | DRN | CKD | APP | DATE 3/1/80 | DWG NO 1C0030 | SCALE NONE

SHT 1 OF 1 | REV 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE	[illegible]		[illegible]
B	GENERALLY REVISED	[illegible]		5/20/80
C	-12V UNREG ADDED	[illegible]		7/15/80
D	TB ADDED	[illegible]		[illegible]



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission if it must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ± .010, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: POWER SUPPLIES

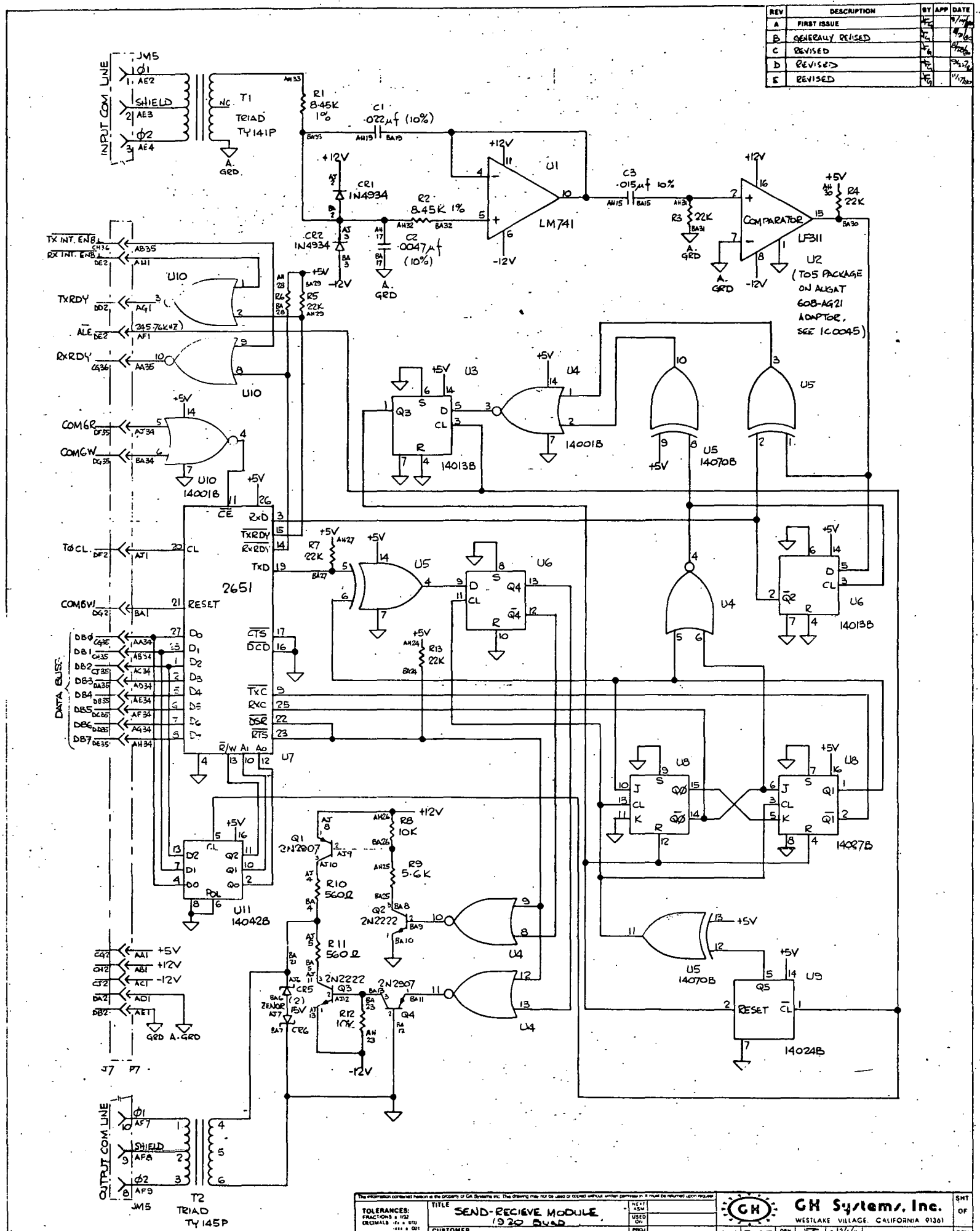
CUSTOMER: DOE

NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO C1100 | DRN [illegible] | CKD | APP | DATE [illegible] | DRG NO 1C0031 | SCALE NONE | SHT 1 OF 1 | REV D

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	GENERALLY REVISED			
C	REVISED			
D	REVISED			
E	REVISED			



The information contained herein is the property of GK Systems Inc. This drawing may not be used or copied without written permission or it must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED

TITLE: SEND-RECIEVE MODULE 1920 BUAD

CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

JOB NO 0100 | SCALE NONE | DRN | CKD | APP | DATE 5/1/82

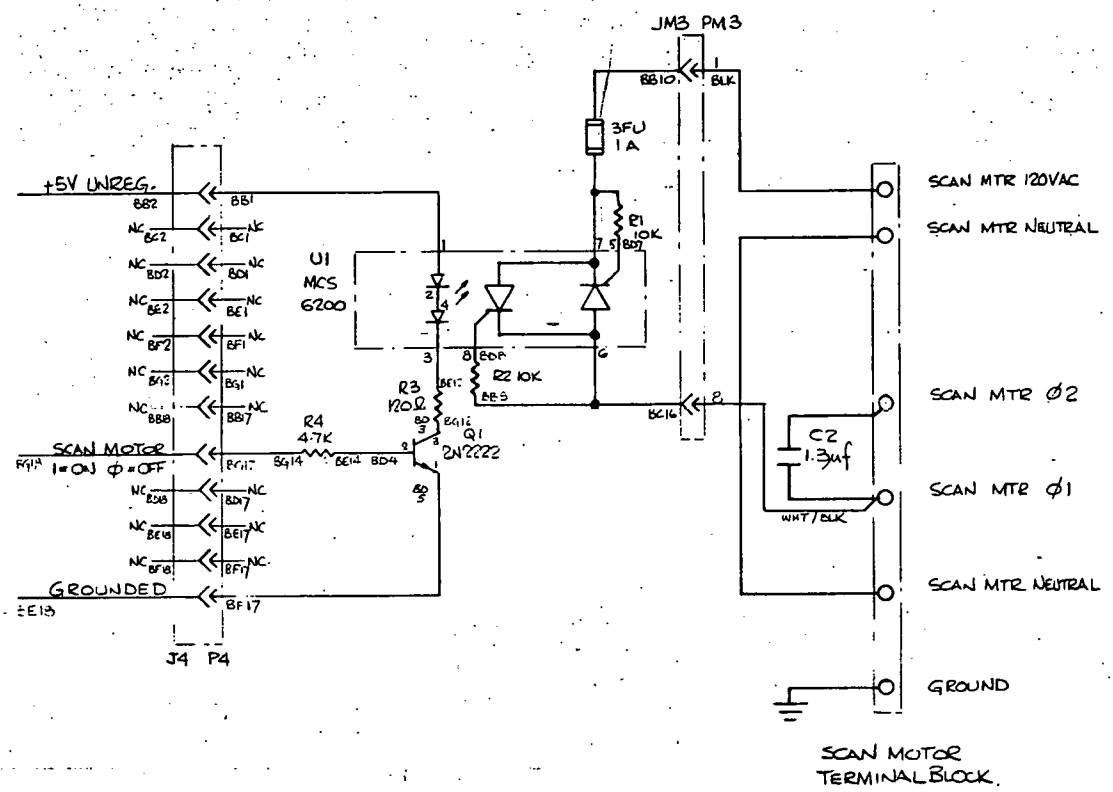
GK Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

DWG. NO. IC0032

SHT OF

REV E

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	REVISED			



NOTE ON JM3 BB11, 12, 13, 14, 15, 16
& BC10, 11, 12, 13, 14, 15
ARE NOT CONNECTED.

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ±.010, ±.001, ANGLES ± ½°, UNLESS OTHERWISE SPECIFIED

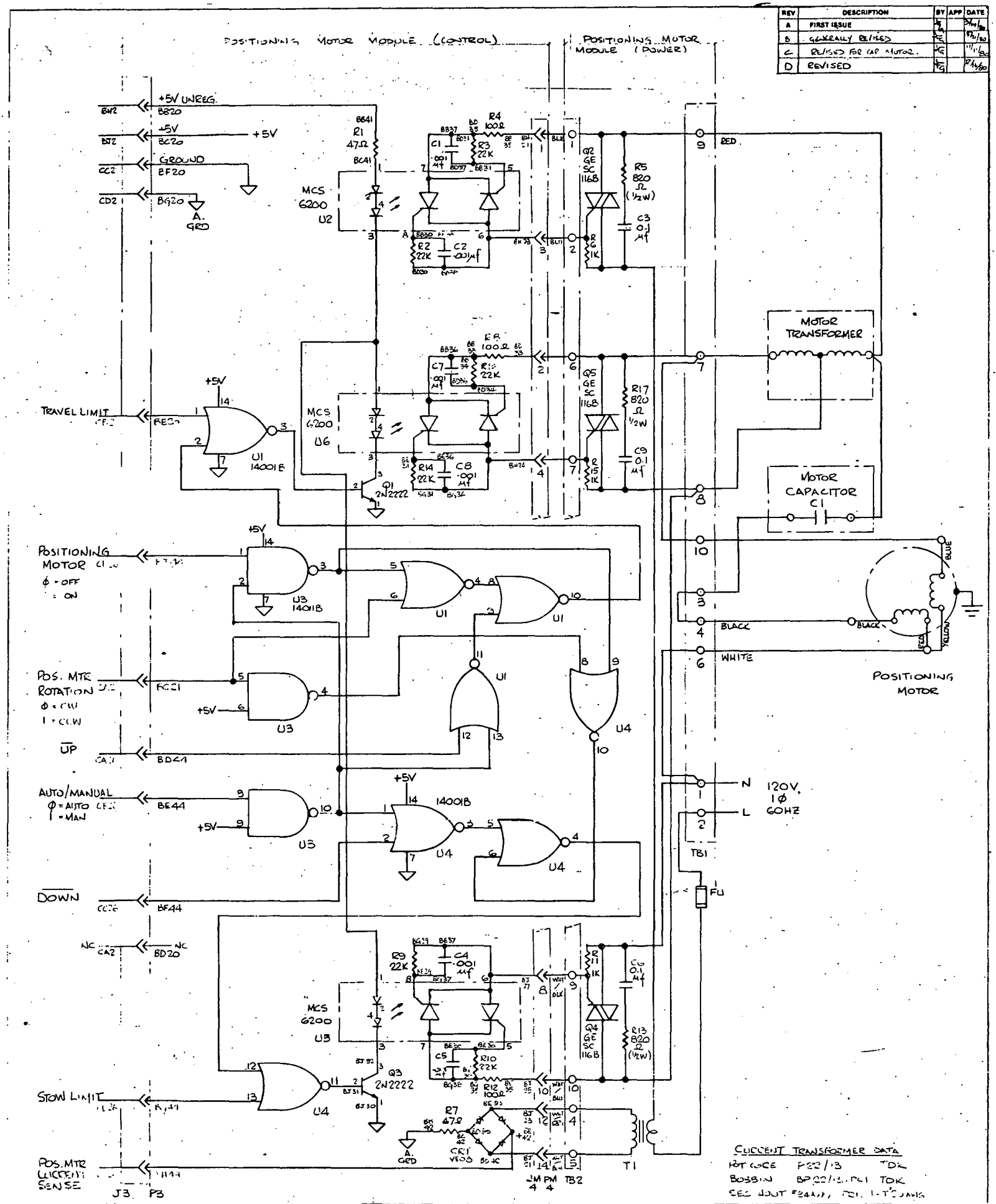
TITLE: SCAN MOTOR MODULE OPTION #1

CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

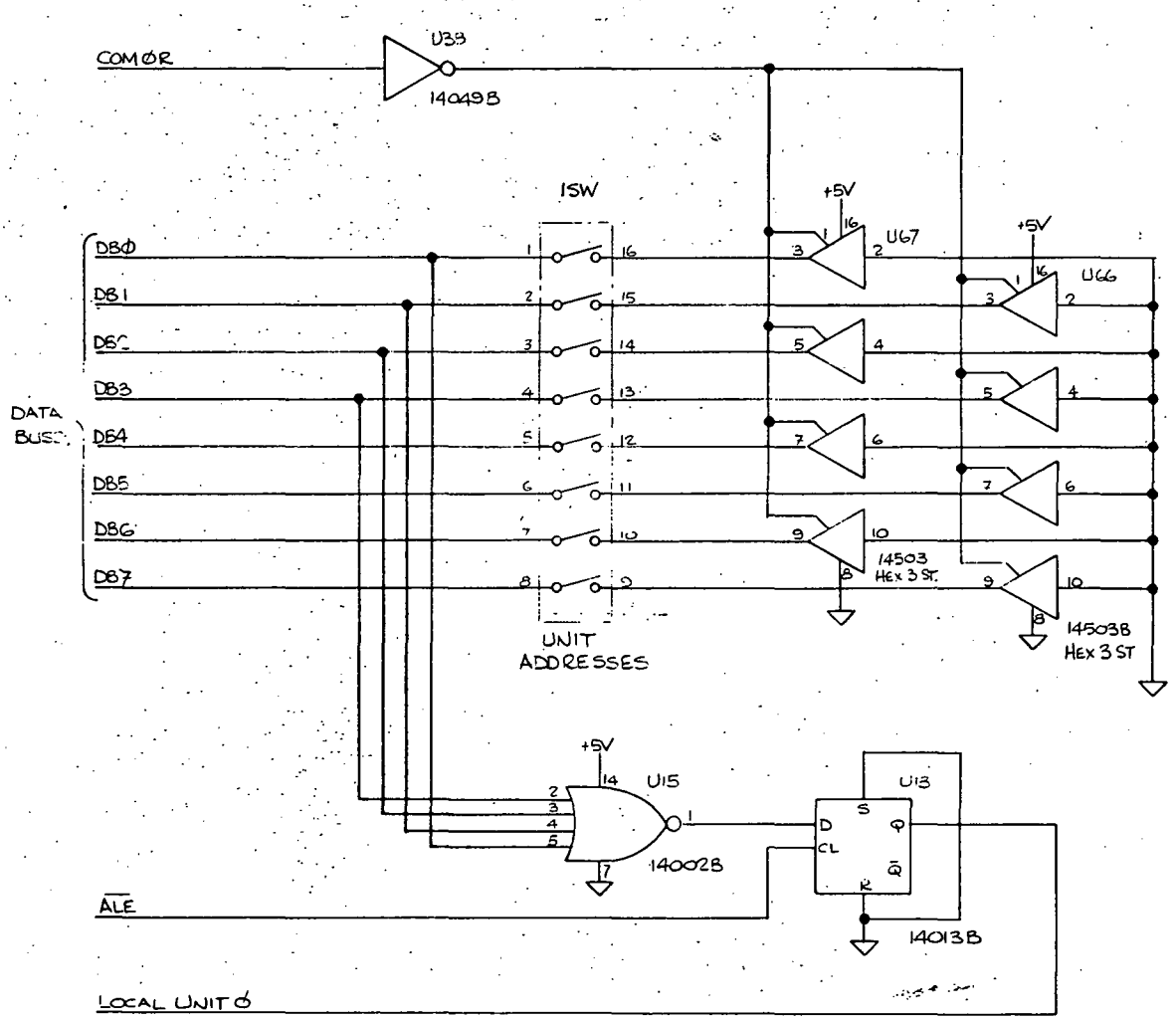
GH Systems, Inc.
WESTLAKE VILLAGE CALIFORNIA 91361

JOB NO 0100 | DRN JEG | CKD | APP | DATE 7/7/80 | DRG NO 1C0033 | SCALE NONE | SHT 1 OF 1 | REV B



REV | DESCRIPTION | BY | APP | DATE
A | FIRST ISSUE
B | GENERALLY REVISED
C | REVISED FOR CAP MOTOR.
D | REVISED
POSITIONING MOTOR MODULE (CONTROL)
POSITIONING MOTOR MODULE (POWER)
+5V UNREG.
+5V
GROUND
A. GRD
MCS 6200 U2
MCS 6200 U6
MCS 6200 U3
Q2 GE SC 116B
Q5 GE SC 116B
Q4 GE SC 116B
Q1 2N2222
Q3 2N2222
R1 47Ω
R2 22K
R3 22K
R4 100Ω
R5 820 Ω (1/2W)
R6 1K
R7 47Ω
R8 100Ω
R9 22K
R10 22K
R11 1K
R12 100Ω
R13 820 Ω (1/2W)
R14 22K
R15 1K
R17 820 Ω 1/2W
C1 .001 μf
C2 .001 μf
C3 0.1 μf
C4 .001 μf
C5
C6 0.1 μf
C7 .001 μf
C8 .001 μf
C9 0.1 μf
TRAVEL LIMIT
U1 14001B
U3 14011B
U4 14001B
POSITIONING MOTOR
ϕ = OFF
= ON
POS. MTR ROTATION
ϕ = CW
1 = CCW
UP
AUTO/MANUAL
ϕ = AUTO
1 = MAN
DOWN
STOW LIMIT
POS. MTR CURRENT SENSE
J3 P3
MOTOR TRANSFORMER
MOTOR CAPACITOR C1
RED
BLUE
BLACK
WHITE
YELLOW
POSITIONING MOTOR
N
L
120V. 1ϕ 60HZ
TB1
FU
T1
JM PM TB2
CURRENT TRANSFORMER DATA
POT CORE P22/13 TDK
BOBBIN BP22/13-PC1 TDK
SEC 400T #24AWG, PRI. 1-T'C JANG
TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± 1/2° UNLESS OTHERWISE SPECIFIED
TITLE POSITIONING MOTOR MODULE
CUSTOMER D.O.E
NEXT ASM
USED ON
PROJ NO
REF NO
GH Systems, Inc.
WESTLAKE VILLAGE CALIFORNIA 91361
JOB NO 0100
SCALE NONE
DRN
CKD
APP
DATE
DRG NO 1C0035
SHT 1 OF 1
REV

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied with out written permission. It must be returned upon request

TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± ½° UNLESS OTHERWISE SPECIFIED

TITLE: UNIT ADDRESS & LOCAL SELECT

CUSTOMER: D-O-E

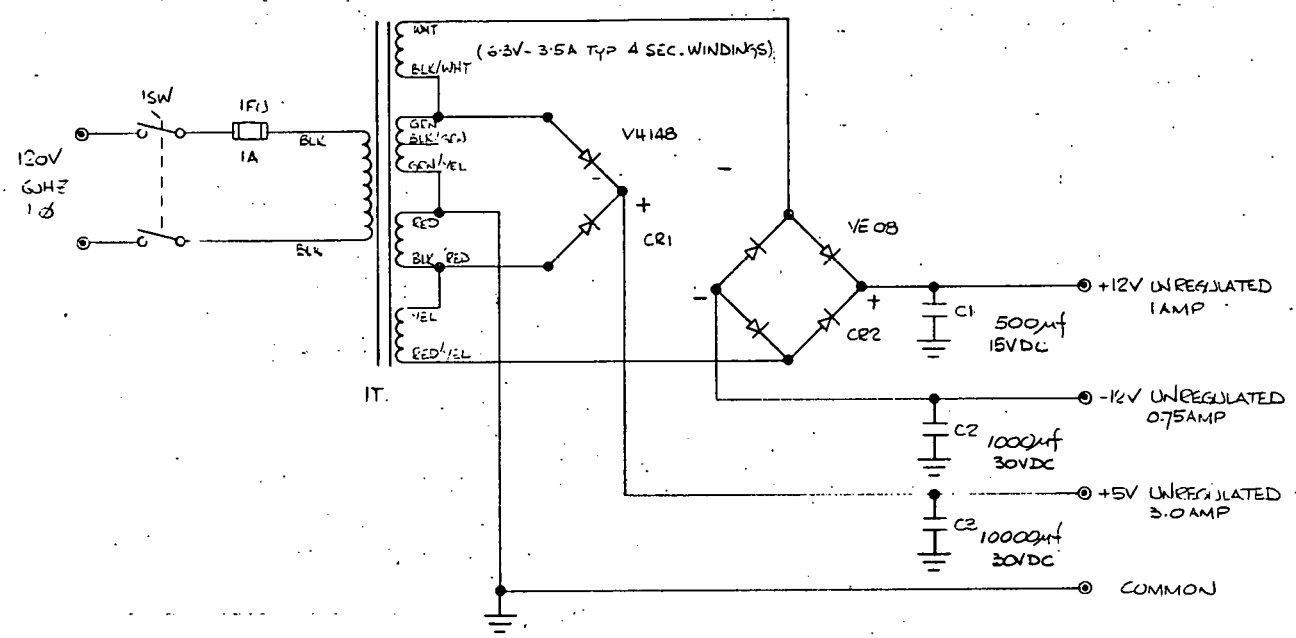
NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO 0100 | DRN | CKD | APP | DATE | SCALE NONE | DWG NO 1C0037

SHT 1 OF 1 | REV A

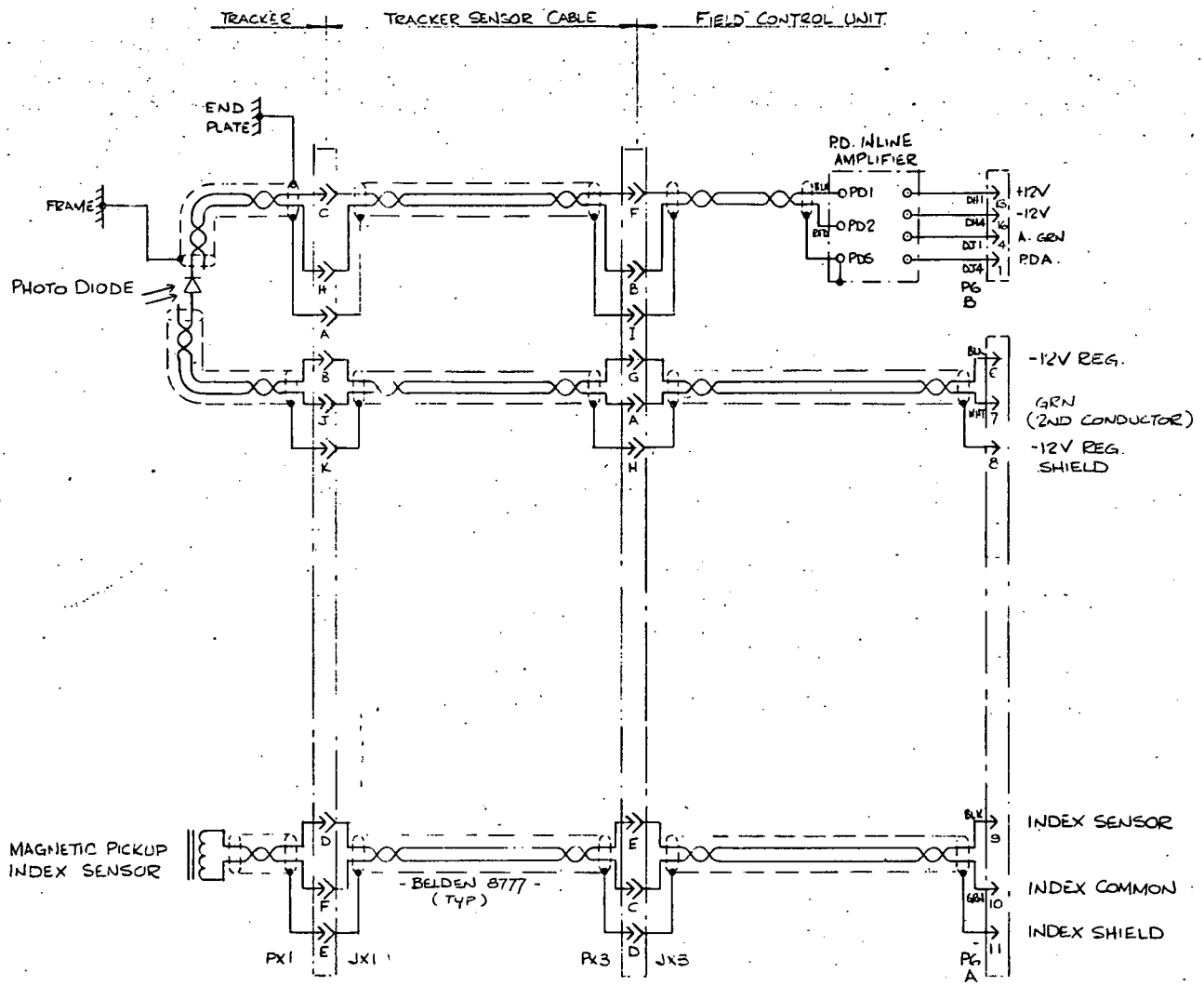
REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	REVISED			1/5/81



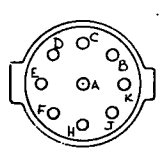
The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS .XX ± .010, .XXX ± .001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED	TITLE: UNREGULATED POWER SUPPLY	NEXT ASM	GH Systems, Inc. WESTLAKE VILLAGE, CALIFORNIA 91361
	CUSTOMER: D.O.E	USED ON	JOB NO 0100 · DRN JFG · CKD · APP · DATE 6-13-80 · DWG NO 10 0040
		PROJ NO	SCALE NONE
		REF NO	

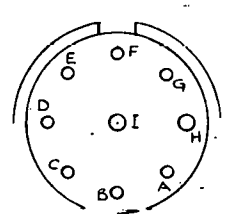
REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	REVISED	[illegible]		[illegible]



PX1	PIN CODING
A	SHIELD (RED)
B	BLK (WHT)
C	BLK (RED)
D	BLK (GRN)
E	SHIELD (GRN)
F	GRN
H	RED
J	WHT
K	SHIELD (WHT)



VIEW ON END OF PX1
(TRACKER SENSOR CABLE CONNECTOR AT TRACKER)



VIEW ON END OF PX3
(TRACKER SENSOR CABLE CONNECTOR AT FIELD UNIT)

PX3	PIN CODING
A	WHT
B	RED
C	GRN
D	SHIELD (GRN)
E	BLK (GRN)
F	BLK (RED)
G	BLK (WHT)
H	SHIELD (WHT)
I	SHIELD (RED)

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ±.010, ±.001, ANGLES ±1/2° UNLESS OTHERWISE SPECIFIED

TITLE: SENSOR CABLE DIAGRAM

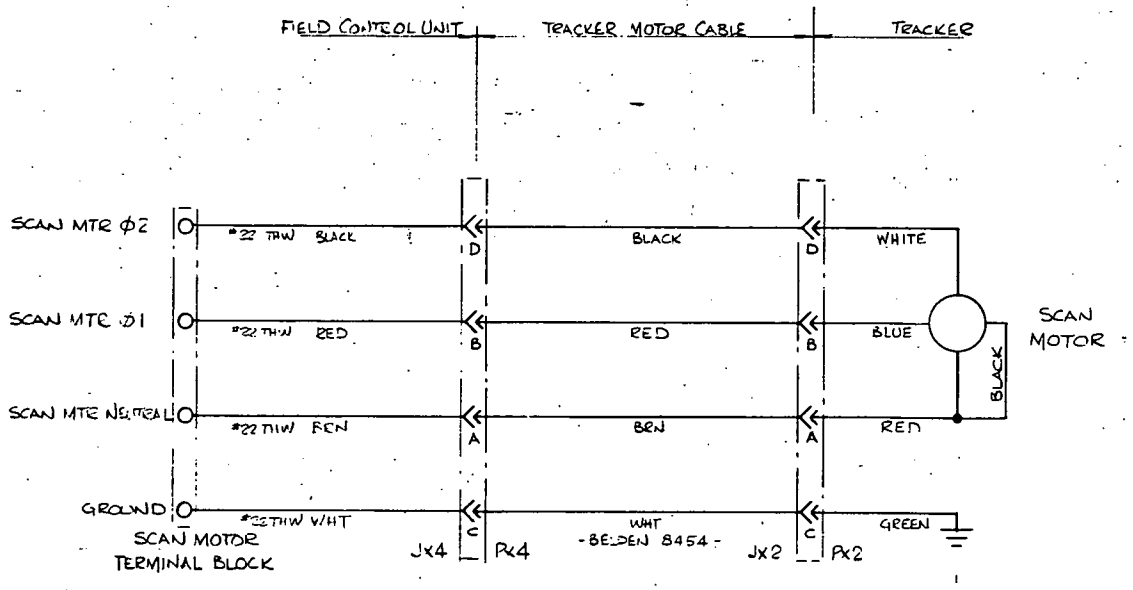
CUSTOMER: D.O.E

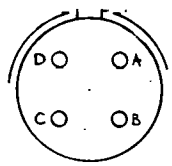
NEXT ASM / USED ON / PROJ NO / REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO / SCALE: NONE / DRN / CKD / APP / DATE / DRG NO: IC 0042 / REV: B / SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	REVISED	[illegible]		12/14/[illegible]





VIEW ON END OF PX4
(TRACKER MOTOR CABLE CONNECTOR AT FIELD UNIT)

VIEW ON END OF PX2
(TRACKER MOTOR CABLE CONNECTOR AT TRACKER)

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ±.010, ±.001, ANGLES ± 1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: MOTOR CABLE DIAGRAM

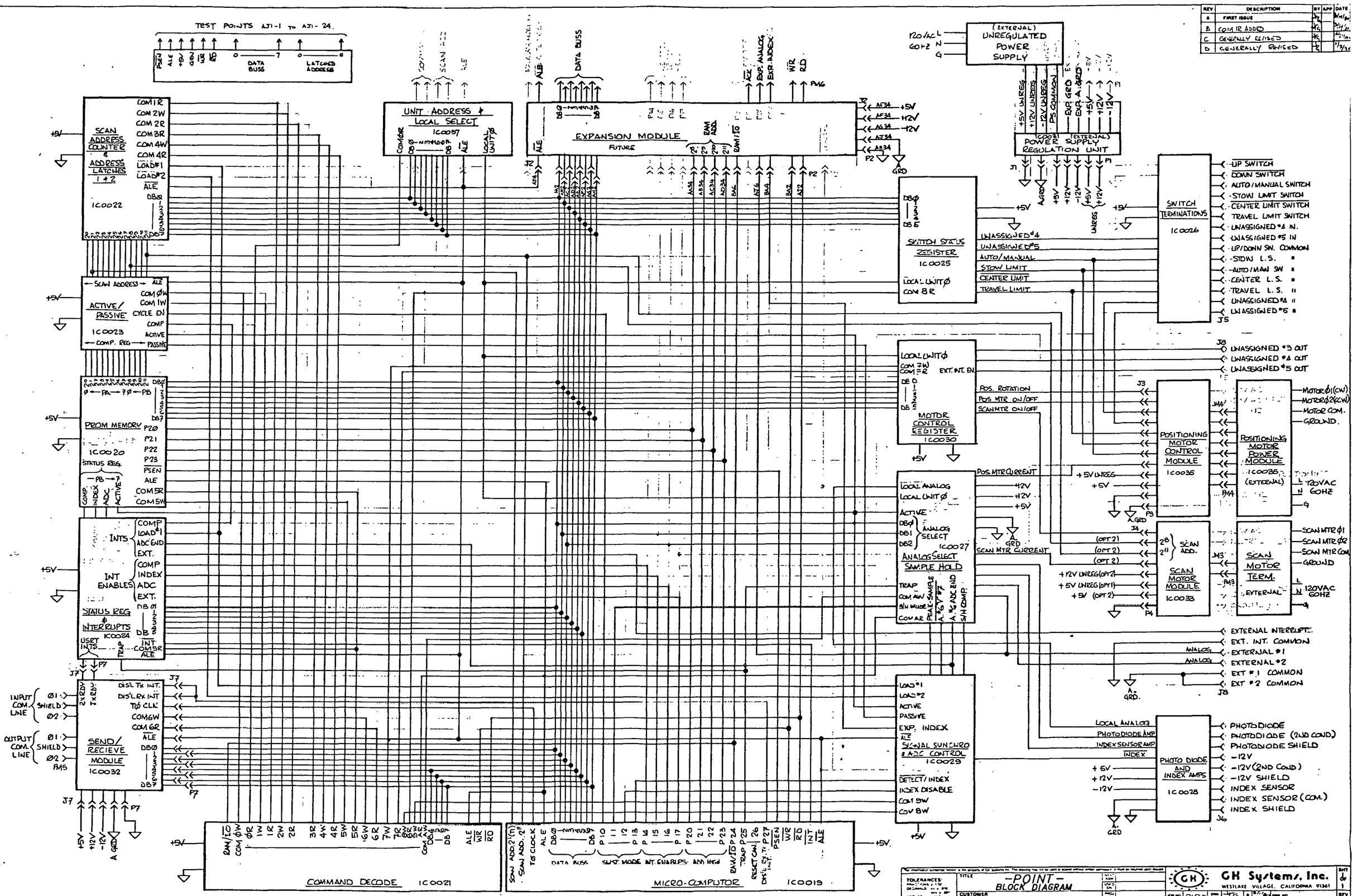
CUSTOMER: D.O.E

NEXT ASM | USED ON | PROJ NO | REF NO

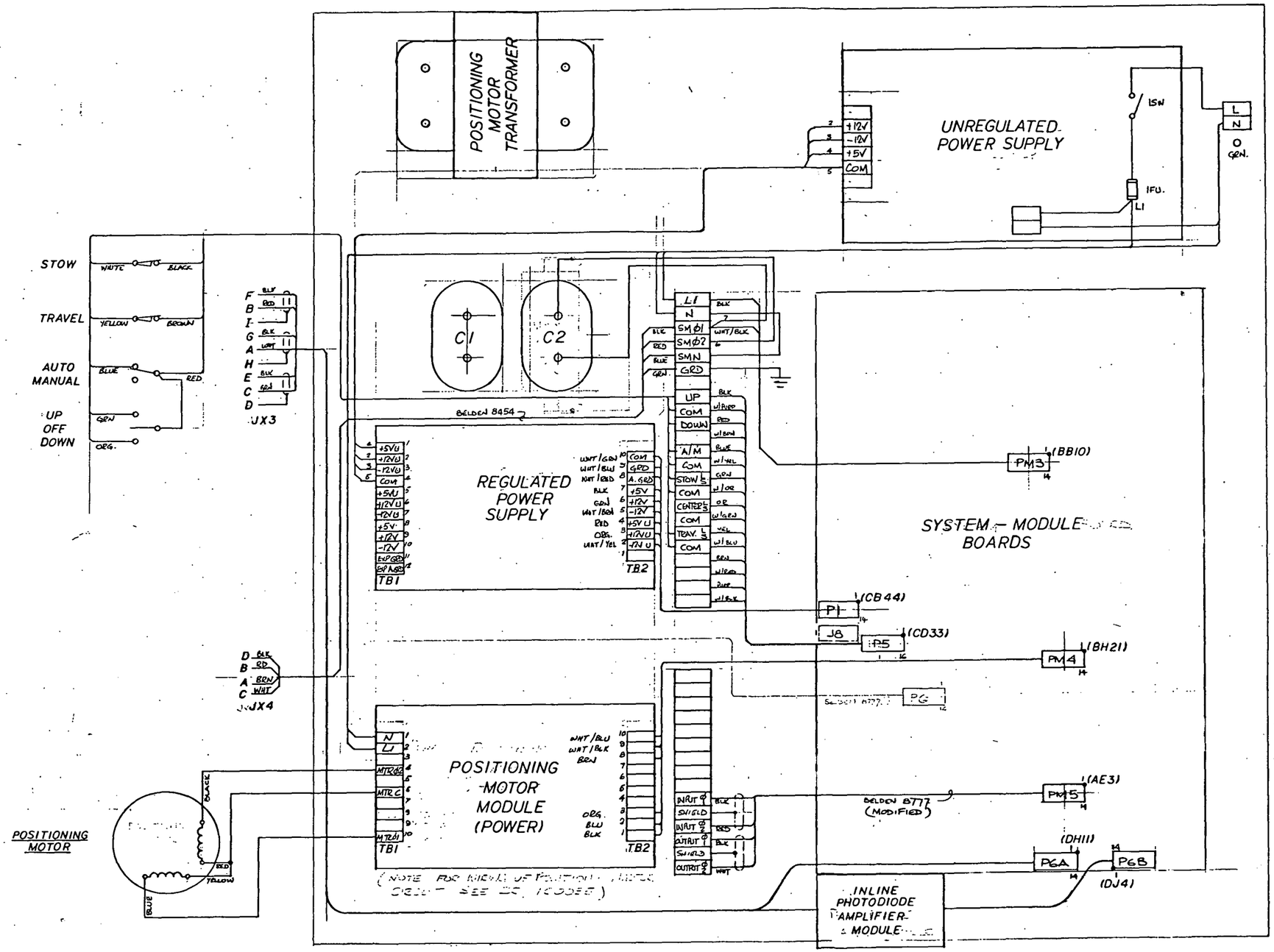
GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

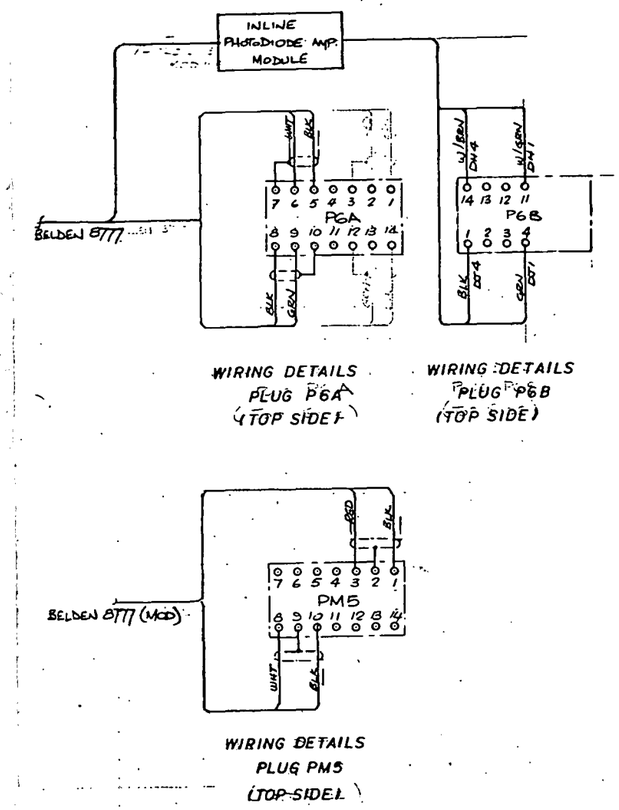
JOB NO 0100 | SCALE NONE | DRN [illegible] | CKD | APP | DATE [illegible] | DRG NO 1C0043 | REV B | SHT 1 OF 1

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			
B	COM IR ADDED			
C	GENERALLY REVISED			
D	GENERALLY REVISED			



REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			





TOLERANCES: FRACTIONS ± 1/32 DECIMALS .XX ± .010 .XXX ± .001 ANGLES ± ½° UNLESS OTHER WISE SPECIFIED

TITLE CONTROL PANEL LAYOUT

CUSTOMER DOE

GH Systems, Inc. WESTLAKE VILLAGE CALIFORNIA 91361

JOB NO 0100 SCALE FULL

DWG NO 2D0044

SHT 1 OF 1 REV A

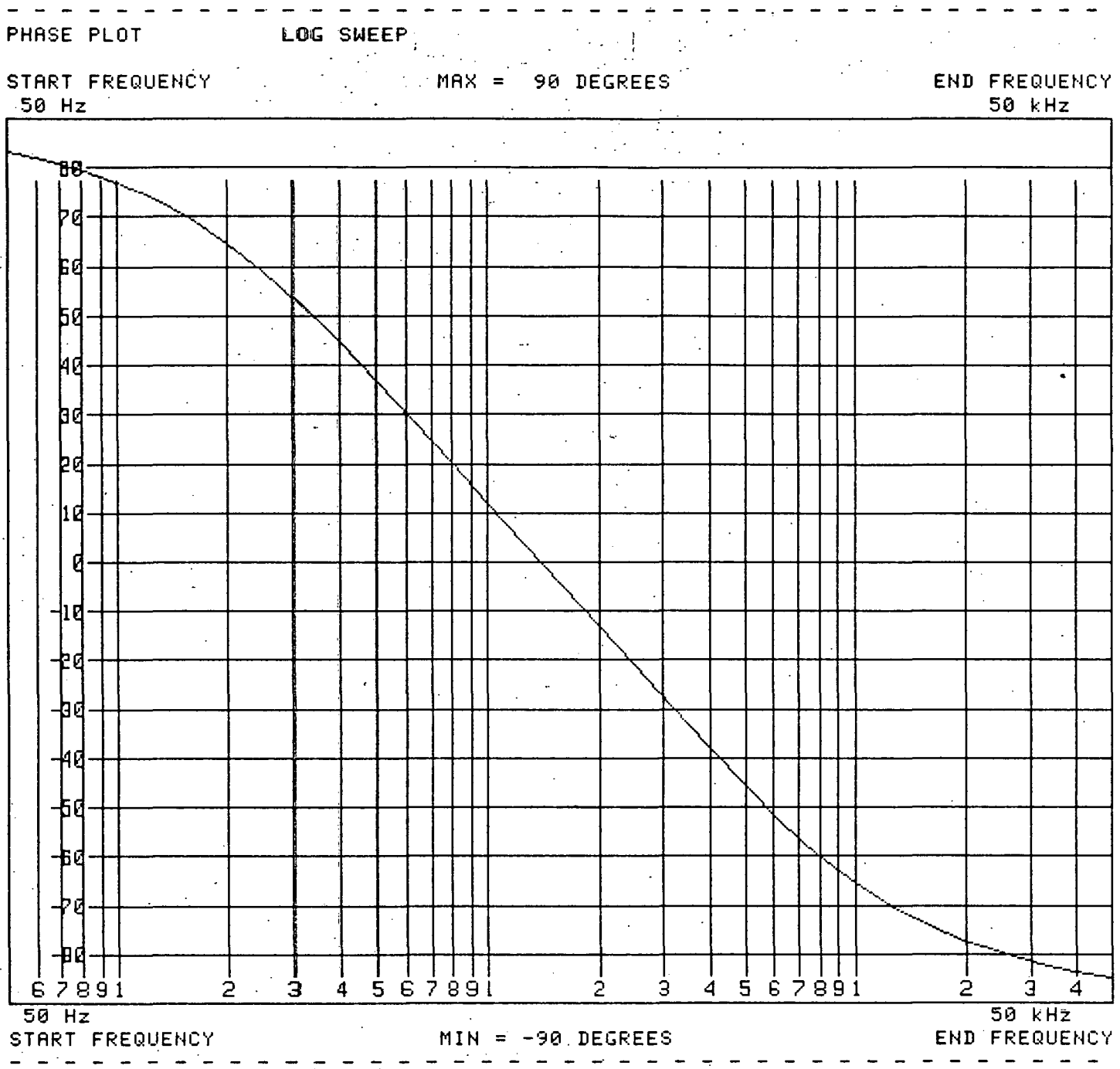

PHASE PLOT
LOG SWEEP
START FREQUENCY
50 Hz
MAX = 90 DEGREES
END FREQUENCY
50 kHz
80
70
60
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
50 Hz
START FREQUENCY
MIN = -90 DEGREES
50 kHz
END FREQUENCY

Fig. 5.2-1

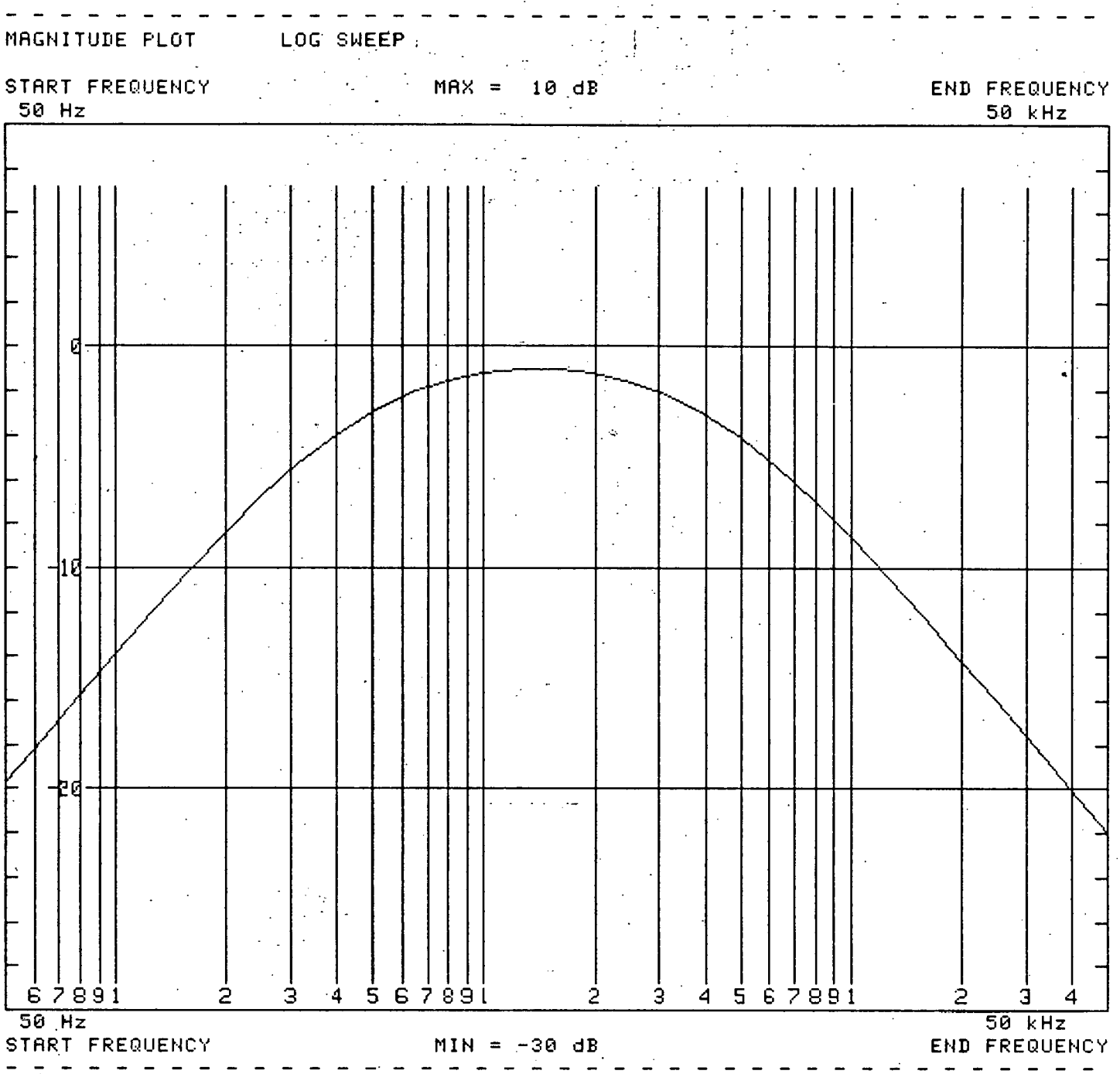

MAGNITUDE PLOT
LOG SWEEP
START FREQUENCY
50 Hz
MAX = 10 dB
END FREQUENCY
50 kHz
0
-10
-20
6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4
50 Hz
START FREQUENCY
MIN = -30 dB
50 kHz
END FREQUENCY

Fig. 5.2-2

6.0 POINT OPERATING SYSTEM DESCRIPTION

6.0 POINT OPERATING SYSTEM

The POINT Operating System is a real time foreground/background system. The operating system is completely interrupt driven. Program cycles as as short as 125 msecs may be programed. Foreground and background program modules execute concurrently. The operating system contains a supervisor, monitor, scheduler, and interrupt handlers.

Foreground program modules are normally associated with data acquisition, and background modules are usually used for data reduction and output. Foreground modules have priority over background modules. Application functions are accomplished by running a program sequence, which define an operation mode. Each sequence is specified as a list of foreground and background modules, executing in parallel. Sequence execution begins with the first foreground module and continues to the end of the the list. Background modules are executed during wait periods in the foreground, or, if needed, after execution of the foreground. Each module completes execution before the next module in the list (foreground or background) is started. Both foreground and background are completed before the sequence is restarted.

Within the appropriate cycle time and tracking precision, a maximum of 16 tracking heads (units) and associated outputs may be supported. The unit addresses are transparent to application program modules through the use of subroutines available to the application program.
The operating system supports several different program TRAPS, and are available to the foreground modules. A Time Queue is provided by the operating system, to provide a time delay before the selected function is performed by the operating system. The Time Queue has 8 msec resolution, and delays up to 8 mins may be used.

Functions supported by the POINT operating system include:

Serial Port	- A transformer coupled sychronous send/receive port is used for installations having a central controller.
I/O Protocol	- Handshaking and I/O operations on ram memory and I/O operations to and from the serial port is provided.
Event Traps	- Many event traps are provided for suspending foreground modules until a specified event(s) occurs.
Program Sequences	- Program modules (foreground and background) are linked together through program sequence definitions. The modules may be used in different program sequences.
Prog.Seq.Control	- Program sequences can be aborted, truncated, or changed by the program modules.
Time Queue	- Performs memory or I/O operations after a selected delay (8 msec resolution).
Module Scheduling	- Foreground/background program modules are dynamically scheduled.
Unit Scheduling	- Tracking units are scheduled and processed by the program sequences.

Operating System Con'd

6.0.1 APPLICATION PROGRAM CONSIDERATIONS:
At present the system is used to control a single field sensor, but in the future this will be expanded to sixteen. The Unit Address is contained in the high order nibble of ACPRG.

Foreground and background modules differ in the functions they may perform. Since foreground modules normally are event dependant, performing data acquisiton and other associated tasks, a major part of the time is usually spent waiting for events to occurr. Wait loops are not permitted, waiting is accomplished by issuing the proper Wait Trap, and passing any required parameters to the operating system, this distinguishes the foreground from the background program modules, where Wait Traps are not permitted.

Foreground or background segments may issue Time Traps, after which control returns immediately to the issuing program module. A Time Trap performs the specified operation after the end of the selected time period. The execution of this operation may be determined by a program module only by testing for the result, or having the time queue set a semaphore.

All program modules are terminated by executing an End Trap. The next program module in the program sequence is then executed. Program sequences may be truncated by issuing an End Sequence Trap, or aborted, Program Sequence 0 will then be executed on the next cycle, after an Abort Trap.

A program sequence may be changed by any program module at any time. The current sequence will continue to completion, and on the next cycle the new sequence will execute. At the end of the program sequence, the next Unit/Program Sequence is started. With a single field sensor, it is the same program sequence, unless it has been changed by Central Control, operating system, or a program module.

At the two major breaks in the program sequence foreground to background, and background(current sequence) to foreground(next program sequence of the same unit), a fixed number of parameters are passed forward in ACTEXAR. The mechanics of this are discussed later.

At system initialization a background sequence of programs must not execute, since there is as yet no information to be processed. ACTEXAR has been cleared and program modules may test B7 of PRGLSTAC to determine if it is the initial cyle, and execute an End Trap or End Sequence Trap.

POINT SYSTEM OPERATING SYSTEM STATE DIAGRAM

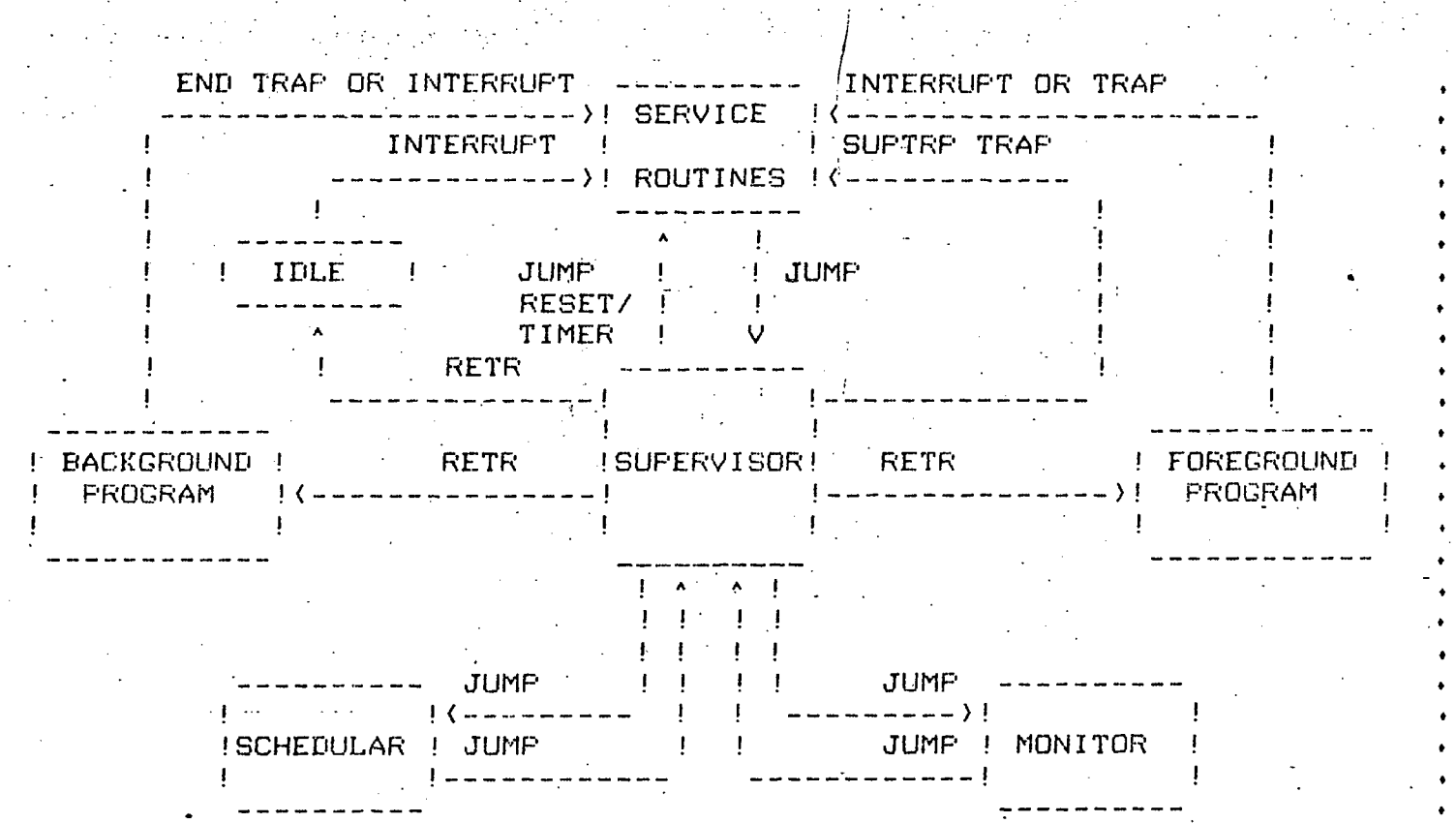


Fig. 6.0.1-1

6.0.2 POINT PROGRAM TABLES

All the program tables necessary to perform a program sequence are contained in Page 3(300-3FF) of the Point Operating System prom; this is not to be confused with Page B of the application program prom.

PRGLST is the start of the program sequence table. A particular program sequence is selected by a 6 bit relative address from PRGLST. This function is entirely under control of the operating system. The entry in PRGLST is a 1 byte address relative to PRGTABL. This address is a start address of the foreground and background module identification list. Each program module ID uses a nibble. Foreground modules are given first. The background and foreground portions are limited to sixteen entries max(8 bytes). If less than the maximum number of entries is made, a NULL entry must be made in the high order nibble, if the first unused program module occurs in the low order nibble to NULL nibbles must be used. Each nibble is a relative address to either BKPROG or FRPROG. Here the absolute starting address of the program module is stored(12 bits). Additionally BKPROG and FRPROG contain 4 additional bits. B12-B14 control the number of bytes(parameters) passed from the opposite mode (foreground or background) to the current mode. These parameters are loaded at the start of ACTEXAR towards the end. The parameters are obtained from buffer areas maintained by the operating system. Regardless of the numnber specified the transfer only occurs before the first program module execution of the foreground and background portions of the program sequence. This transfer is ignored for all other program modules.

6.1 POINT OPERATING SYSTEM DEFINITIONS

POINT REGS,COMMANDS,AND CONVENTIONS

SYMBOLS AND ABBREVIATIONS

A	HEXIDECIMAL ADDRESS CHARACTER
A1H	ADDRESS LATCH #1 HIGH; (A11-A8)=(DB3-DB0); READ ONLY
A1L	ADDRESS LATCH #1 LOW; (A7-A0)=(DB7-DB0); READ ONLY
A2H	ADDRESS LATCH #2 HIGH; (A11-A8)=(DB3-DB0); READ ONLY
A2L	ADDRESS LATCH #2 LOW; (A7-A0)=(DB7-DB0); READ ONLY
ACC	ACCUMULATOR; (READ/WRITE)
(ACC)	HH, D7-D0
ACT	ACTIVE CYF BOOLEAN STATE
AL1	ADDRESS LATCH #1; (A11-A8)(A7-A0)=(DB3-DB0)(DB7-DB0); READ ONLY
AL2	ADDRESS LATCH #2; (A11-A8)(A7-A0)=(DB3-DB0)(DB7-DB0); READ ONLY
AN	BINARY ADDRESS BIT N = 0-7
ANS	ANALOG SELECT REGISTER; (A2-A0)=(DB2-DB0); WRITE ONLY SELECT 1 OF 8 ANALOG SOURCES
BN	BINARY BIT N = 0-7
CPH	COMPARISON REGISTER HIGH; (A11-A8)=(DB3-DB0); WRITE ONLY
CPL	COMPARISON REGISTER LOW; (A7-A0)=(DB7-DB0); WRITE ONLY
CPR	COMPARISON REGISTER; (A11-A8)(A7-A0)=(DB3-DB0)(DB7-DB0); WRITE ONLY
CYF	CYCLE FF; (BOOLEAN); READ/WRITE
D	HEXIDECIMAL DATA CHARACTER
DN	BINARY DATA BIT N = 0-7
H	HEXIDECIMAL CHARACTER
LUNT	LOCAL UNIT ADDRESS; (A3-A0)=(DB3-DB0); WRITE ONLY
MTR	MOTOR CONTROL REGISTER; (D5-D0)=(DB7-DB0); WRITE ONLY
P24	PORT #2 B4; RAM/^IO; WRITE ONLY
PAS	PASSIVE; CYF BOOLEAN STATE; PAS=^ACT
R	REGISTER NUMBER; (READ/WRITE)
(R)	HH, D7-D0
SCAN	SCAN ADDRESS COUNTER (A11-A0)
STT	STATUS REGISTER; (D3-D0)=(DB7-DB0); PORT B; READ ONLY
SWS	SWITCH STATUS REGISTER; (D5-D0)=(DB7-DB0); READ ONLY
UNT	UNIT NUMBER; (A7-A0); READ ONLY; ADDRESS SWITCH SELECTABLE
URD	USRT RECEIVE DATA REGISTER; (D7-D0)=(DB7-DB0); READ ONLY
USD	USRT SEND DATA REGISTER; (D7-D0)=(DB7-DB0); WRITE ONLY
UST	USRT STATUS REGISTER; (D7-D0)=(DB7-DB0); READ ONLY
URM	USRT MODE REGISTERS; (D7-D0)=(DB7-DB0); READ ONLY
UWM	USRT MODE REGISTERS; (D7-D0)=(DB7-DB0); WRITE ONLY
USY	USRT SYNC REGISTERS; (D7-D0)=(DB7-DB0); WRITE ONLY
URC	USRT COMMAND REGISTER; (D7-D0)=(DB7-DB0); READE ONLY
UCM	USRT COMMAND REGISTER; (D7-D0)=(DB7-DB0); WRITE ONLY

I/O PORTS AND DESCRIPTIONS

```
PORT#1  [P17][P16][P15][P14][P13][P12][P11][P10]      WRITE ONLY
        <---INT.ENABLES----><---SYSTEM MODE---->
         SYSTEM MODE
         P10      CYCLE ENABLE
                  0. CYCLE FF IS DISABLED FROM TOGGLING.
                  1. CYCLE FF TOGGLES ON POSITIVE COMPARISON.
         P11      SAMPLE/HOLD MODE
                  0. SAMPLE
                  1. PEAK DETECTION
         P12      ^DETECT/INDEX
                  0. ADDRESS LATCH #2 LOADS ON PEAK DETECT.
                  1.    "       "    "    "     "  INDEX.
         P13      INDEX DISABLE
                  0. ADDRESS LATCH #1 LOADS ON INDEX.
                  1.    "       "    "  DISABLED FROM INDEX.
         INTERRUPT ENABLES
         P14      COMPARISON INTERRUPT ENABLE
         P15      INDEX          "        "
         P16      ADC END        "        "
         P17      EXTERNAL       "        "
PORT #2  [P27][P26][P25][P24][P23][P22][P21][P20]      WRITE ONLY
         <UNASSIGN><TRP>R/^IO<--HIGH ORDER ADD-->
         P20-P23 PROGRAM MEMORY ADDRESS 2*11-2*8
                  IF P24 IS SET SELECTS EXPANSION RAM MEMORY BANKS.
         P24      SELECTS EXTERNAL EXPANSION RAM OR IO COMMANDS
                  0. ENABLES ALL SYSTEM IO COMMANDS
                  1. READ/WRITE EXTERNAL RAM MEMORY
         P25      TRAP COMMAND IS EXECUTED, THIS BIT IS
                  NEVER SET IN THE RAM PORT2 BYTE
         P26      CANCEL RESET, RESET CANCELED BY ALTERNATING THE BIT.
         P27      DISABLE SEND/RECEIVE INTERRUPTS
CPL      COMPARISION REG LOW                                WRITE ONLY
         [ A7 ][ A6 ][ A5 ][ A4 ][ A3 ][ A2 ][ A1 ][ A0 ] ADDR.= 5X,P24=0
         <2*7-----------------------------------------2*0> HRDWRE COM.=5W
         <DB7-----------------------------------------DB0>
CPH      COMPARISION REG HIGH                               WRITE ONLY
         [ A11][ A10][ A9 ][ A8 ]                           ADDR.=3X,P24=0
         <2*11---------------2*8>                           HRDWRE COM.=3W
         <DB3----------------DB0>
STT      STATUS REG (PORT B)                                READ ONLY
         [ D3 ][ D2 ][ D1 ][ D0]                            ADDR.50,P24=0
         <DB7---------------DB4>                            HRDWRE COM.=5R
         D4       COMPARISON
         D5       INDEX
         D6       ADC END
         D7       ACTIVE
```

```
UNT       UNIT NUMBER                                          READ ONLY
          [ A7 ][ A6 ][ A5 ][ A4 ][ A3 ][ A2 ][ A1 ][ A0 ]     ADDR.0X,P24=0
          (2*7-----------------------------------------2*0)   HRDWRE COM.=0R
          (DB7-----------------------------------------DB0)
A1L       ADDRESS LATCH #1 LOW                                 READ ONLY
          [ A7 ][ A6 ][ A5 ][ A4 ][ A3 ][ A2 ][ A1 ][ A0 ]     ADDR.2X,P24=0
          (2*7-----------------------------------------2*0)   HRDWRE COM.=2R
          (DB7-----------------------------------------DB0)
A1H       ADDRESS LATCH #1 HIGH                                READ ONLY
          [ A11][ A10][ A9 ][ A8 ]                             ADDR.1X,P24=0
          (2*11----------------2*8)                            HRDWRE COM.=1R
          (DB3-----------------DB0)
A2L       ADDRESS LATCH #2 LOW                                 READ ONLY
          [ A7 ][ A6 ][ A5 ][ A4 ][ A3 ][ A2 ][ A1 ][ A0 ]     ADDR.4X,P24=Q
          (2*7-----------------------------------------2*0)   HRDWRE COM.=4R
          (DB7-----------------------------------------DB0)
A2H       ADDRESS LATCH #2 HIGH                                READ ONLY
          [ A11][ A10][ A9 ][ A8 ]                             ADDR.3X,P24=0
          (2*11----------------2*8)                            HRDWRE COM.=3R
          (DB3-----------------DB0)
MTR(N)    MOTOR CONTROL REGISTER                               READ/WRITE
          [ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ]                 ADDR.7N,P24=0
          (DB5-----------------------------DB0)                HRDWRE COM.=7W
          D0,      SCAN MOTOR ON                               HRDWRE COM.=7R
          D1,      POSITIONING MOTOR ON                        N=LOCAL UNIT ADDR.
          D2,      POSITIONING MOTOR ROTATION
                   0, CW
                   1, CCW
          D3,      APPLICATION PROGRAM FLAG
          D4-D5,   UNASSIGNED OUTPUTS
SWS(N)    SWITCH STATUS REGISTER                               READ ONLY
          [ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ]                 ADDR.8N,P24=0
          (DB5-----------------------------DB0)                HRDWRE COM.=8R
          D0,      ^AUTO/MANUAL                                N=LOCAL UNIT ADDR.
          D1,      STOW LIMIT
          D2,      ^CENTER LIMIT
          D3,      TRAVEL LIMIT
          D4-D5,   UNASSIGNED INPUTS
ANS(N)    ANALOG SELECT REGISTER                               READ/WRITE
          [ D5 ][ D4 ][ D3 ][ A2 ][ A1 ][ A0 ]                 ADDR.AN,P24=0
          (DB5-----------------------------DB0)                HRDWRE COM.=AW
          A0-A2    ANALOG SELECT          READ/WRITE           HRDWRE COM.=AR
                   0,    EXTERNAL #1                            N=LOCAL UNIT ADDR
                   1,    EXTERNAL #2
                   2,    PHOTO DIODE AMP
                   3,    INDEX SENSOR AMP
                   4,    SCAN MOTOR CURRENT
                   5,    POSITIONING MOTOR CURRENT
                   6-7,  INTERNAL HARDWARE
          D3,      TRAP STATUS            READ ONLY
          D4-D5,   UNASSIGNED             READ ONLY
```

```
URD     USRT RECEIVE DATA REGISTER                              READ ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR.60,P24=0
        (DB7----------------------------------------DB0) HRDWRE COM.=6R

UST     USRT STATUS REGISTER                                    READ ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR. 61,P24=0
        (DB7----------------------------------------DB0) HRDWRE COM.=6R
        D0.     TXRDY       SEND DATA BUFFER EMPTY
        D1.     RXRDY       READ CHARACTER AVAILABLE
        D2.     TXEMT/DSCHG TRANSMITER EMPTY OR DCD, DSR CHANGE
        D3.     PE          PARITY ERROR OR DLE CHARACETER RECEIVED
        D4.     OE          OVERRUN ERROR
        D5.     FE/SYN      SYN CHARACTER DETECTED
        D6.     DCD         DATA CARRIER DETECT
        D7.     DSR         DATA SET READY

        READ OR WRITES OF THE STATUS REGISTERS INCREMENT AN INTERNAL
        ROTARY ADDRESS POINTER. THE POINTER IS RESET BY A COMMAND
        REGISTER READ.
URM     USRT READ MODE REGISTERS                                READ ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR. 62,P24=0
        (DB7----------------------------------------DB0) HRDWRE COM.=6R
UWM     USRT WRITE MODE REGISTERS                               WRITE ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR. 66,P24=0
        (DB7----------------------------------------DB0) HRDWRE COM.=6W
        MODE REGISTER 1
        D0-1.   MODE/BAUD   00=SYN; 1=1X, 2=16X, 3=64X ASYN BAUD RATE
        D2-3.   CHARACTER LENGTH
                0.  5 BITS
                1.  6 BITS
                2.  7 BITS
                3.  8 BITS  *
        D4.     PEN         PARITY ENABLE
        D5.     EP          EVEN PARITY
        D6-7.   STOP BITS   ASYNC,0=INVALID,1=1 BIT,2=11/2BITS,3=2 BITS
        D6      TRANSPARENCY CONTROL
                0.          NORMAL
                1.          TRANSPARENT
        D7.     SCS         SINGLE CHARACTER SYNC
                0.          TWO SYN CHARACTERS
                1.          SINGLE SYN CHARACTER
        MODE REGISTER 2
        D0-3.   BAUD RATE WITH INTERNAL CLOCK
                0.  50      1.  75      2.  110     3.  134.5
                4.  150     5.  300     6.  600     7.  1200
                8.  1800    9.  2000    10. 2400    11. 3600
                12. 4800    13. 7200    14. 9600    15. 19200
        D4.     RECEIVOR CLOCK
                0.          EXTERNAL
                1           INTERNAL
        D5.     TRANMITTER CLOCK
                0.          EXTERNAL
                1           INTERNAL
        D6-7    NOT USED
```

```
URC     USRT COMMAND REGISTERS                                  READ ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR.63,P24=0
        (DB7-----------------------------------------DB0) HRDWRE COM.=6R
UCM     USRT COMMAND REGISTERS                                  WRITE ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR.67,P24=0
        (DB7-----------------------------------------DB0) HRDWRE COM.=6W
        D0.     TXEN    TRANSMIT ENABLE
        D1.     DTR     DATA TERMINAL READY
        D2.     RXE     RECEIVE ENABLE
                        SYN, 0 TO 1 TRANSITIONS PLACE USART IN HUNT MODE
        D3.     1.      ASYN, SEND BREAK CHARACTER
                2.      SYN, SEND DLE CHAR BEFORE DATA CHARACTER
        D4.     ER      ERROR RESET PE,OE, AND FE
        D5      RTS     REQUEST TO SEND, ENABLES OUTPUT MODULATOR
        D6-7.   OPERATING MODE
                0.      NORMAL
                1.      ASYN, AUTOMATIC ECHO MODE
                        SYN, SYN AND/OR DLE STRIPING MODE
                2.      LOCAL LOOP BACK
                3.      REMOTE LOOP BACK

USD     USRT SEND DATA REGISTER                                 WRITE ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR.64,P24=0
        (DB7-----------------------------------------DB0) HRDWRE COM.=6W


        WRITES OF THE SYNC REGISTERS INCREMENT AN INTERNAL
        ROTARY ADDRESS POINTER. THE POINTER IS RESET BY A COMMAND
        REGISTER READ.
USY     SYNC CHARACTER REGISTER(S)                              WRITE ONLY
        [ D7 ][ D6 ][ D5 ][ D4 ][ D3 ][ D2 ][ D1 ][ D0 ] ADDR.65,P24=0
        (DB7-----------------------------------------DB0) 1ST, 2ND AND 3RD,
                                                            BYTES.
                                                            HRDWRE COM.=6W
        1.      FIRST BYTE IS THE SYN1 CHARACTER
        2.      SECOND BYTE IS THE SYN2 CHARACTER
        3.      THIRD BYTE IS THE DLE CHARACTER
```

I/O COMMANDS

```
READ EXTERNAL
LABEL     LD       A,X(ADDR)
LABEL     LD       A,X@(REG OR A)
LABEL     OPCODE   REG,X(ADDR)
LABEL     OPCODE   REG,@(REG OR A)
LABEL     OPCODE   (INTERNAL MEMORY),X(ADDR)
LABEL     OPCODE   (INTERNAL MEMORY),X@(REG OR A)
WRITE EXTERNAL
LABEL     LDX      (EXTERNAL ADDR),A
LABEL     LDX      (EXTERNAL ADDR),REG
LABEL     LDX      @(REG OR A),A OR REG
LDX       COMMANDS ARE FOR EXTERNAL WRITE ONLY OPERATIONS
X(ADDR)   EXTERNAL MEMORY ADDRESS READ ONLY ASSUMED
X@REG     INDIRECT EXTERNAL MEMORY ADDRESS,READ ONLY ASSUMED
LDX @REG  INDIRECT EXTERNAL MEMORY ADDRESS,WRITE ONLY ASSUMED
REG       R0-R7
A         ACCUMULATOR
```

INPUT/OUTPUT ADDRESS MAP, P24=0

	READ	WRITE	
UNIT NUMBER	0X. UNT	0X. ACT	CYCLE FF ACTIVE, NO PARM.
ADDRESS LATCH #1 HIGH	1X. A1H	1X. PAS	CYCLE FF PASSIVE, NO PARM.
ADDRESS LATCH #1 LOW	2X. A1L	2X. AL1	LOAD ADD. LATCH#1,NO PARM.
ADDRESS LATCH #2 HIGH	3X. A2H	3X. CPH	COMP.REG HIGH
ADDRESS LATCH #2 LOW	4X. A2L	4X. AL2	LOAD ADD. LATCH#2,NO PARM
STATUS REGISTER	50. STT	50. CPL	COMP. REG. LOW
USRT RECIEVE DATA REG.	60. URD		
USRT STATUS REGISTER	61. UST		
USRT MODE REGISTERS	62. URM		
USRT COMMAND REGISTERS	63. URC		
		64. USD	USRT SEND DATA REGISTER
		65. USY	USRT SYNC REGISTERS
		66. UWM	USRT MODE REGISTERS
		67. UCM	USRT COMMAND REGISTER
MOTOR CONTROL REGISTER	7N. MTR(N)	7N. MTR(N)	MOTOR CONTROL REGISTER
SWITCH STATUS REGISTER	8N. SWS(N)	8X. SHC	ZERO S/H CAP., NO PARM.
	9N. *	9X. ADC	START ADC, NO PARM.
ANALOG SELECT REGISTER	AN. ANS(N)	AN. ANS(N)	ANALOG SELECT REGISTER
		B0. URES	RESET USART
	BX-FX. *	CX-FX. *	

Fig. 6.1-1

```
*         UNASSINGED
X         DON'T CARE
N         LOW ORDER HEX ADDRESS PARAMETER
```

INTERNAL OPERATIONS

ALL INTERNAL SOURCE OR DESTINATIONS ARE VALID, USING ACCUMULATOR, REGISTER, INDIRECT REGISTER ADDRESS OF MEMORY, OR MEMORY. ONLY INDIRECT REGISTER SOURCE AND DESTINATION ADDRESSING OF INTERNAL MEMORY IS NOT VALID. MACROS ACCOMPLISH THESE FUNCTIONS. OPCODES USED ARE THE SAME AS THE 8048 SET WITH A M SUFFIXED TO THE OPCODE. ONLY THE LD COMMAND OPERATES WITHOUT THE M SUFFIX.

TRAPS

THERE ARE SEVERAL SYSTEM TRAPS. THE END TRAP IS USED AT THE COMPLETION OF THE FOREGROUND, OR BACKGROUND PROGRAM. AN ABORT TRAP IS USED TO ABORT A PROGRAM SEQUENCE FROM EITHER THE FOREGROUND OR THE BACKGROUND. A WAIT IS USED BY THE FOREGROUND PROGRAM TO SUSPEND OPERATION UNTIL AN EXTERNAL EVENT HAS OCCURED. THE WAIT TRAP PASSES A CODE TO THE SUPERVISOR INDICATING THE TYPE OF TRAP BEING EXECUTED IN THE ACCUMULATOR AND THE EXTERNAL EVENT THAT WILL RESUME EXECUTION OF THE FOREGROUND PROGRAM. A TIME TRAP PASSES TWO PARAMETERS IN REGISTERS R2 AND R3(RB0). R2 IS FUNCTION OPERAND DESCRIBING IF A TIME OR EVENT FUNCTION IS TO BE EXECUTED, IF A TIME FUNCTION, HIGH OR LOW TIME MODE, AND A TABLE ADDRESS IN PAGE 3 OF PROGRAM MEMORY DESCRIBING THE THE FUNCTION TO BE EXECUTED. AN END SEQUENCE TRAP TRUNCATES THE PROGRAM SEQUENCE.

```
TRPINT, ACCUMULATOR
[B7][B6][B5][B4][B3][B2][B1][B0]
<-------TRAP CODE------>< TRAP >
B0-B1   TYPE TRAP
        0.      NOT VALID, OR FALSE TRAP
                B7-B2.   UNASSINGED
        1.      ACTION  TRAP   INTERUPT-
                   B2.   UNASSIGNED
                B7-B3.   TYPE ACTION TRAP
                    0.   END   TRAP
                    1.   TIME/EVENT TRAP, R2 AND R3(RB0) CONTAIN PARAMETERS
                         EXECUTION OF PROGRAM RESUMES.
                    2.   ABORT TRAP; ABORTS PROGRAM SEQUENCE
                    3.   END SEQUENCE TRAP
                 4-31.   NOT ASSIGNED
        2.      SUPERVISOR TRAP
                B7-B2.   UNASSINGED
        3.      WAIT TRAP INTERUPT
                B3-B2.   UNASSINGED
                   B4.   COMPARISON   TRAP
                   B5.   INDEX          "
                   B6.   ADC END        "
                   B7.   EXTERNAL       "
      UNASSINGED, OR INVALID TRAPS ARE INTERPRETED AS FALSE TRAPS
      AND EXECUTION RESUMES
```

```
MONITOR ENTRY CONDITIONS:
UPON ENTRY TO THE MONITOR THE ACCUMULATOR CONTAINS THE ENTRY CODE.
[B7][B6][B5][B4][B3][B2][B1][B0]
  0   0   0   0   0   0   0   0   NOP
  0   0   0   0 [  ENTRY CODE  ]  ENTRY OR SYS ERROR CONDITION
                [       1      ]  SERIAL WORD SENT
                [       2      ]  RECEIVE CHARACTER PARITY ERROR
                [       3      ]  RECEIVE CHARACTER OVERRUN ERROR
                [       4      ]  SEND MESSAGE TERMINATED
                [       5      ]  TIMER
                [       6      ]  NOT ASSIGNED
                [       7      ]  RESET
                [     8 - F    ]  NOT ASSIGNED
  0 [ERROR CODE][ UNIT NUMBER  ]  UNIT ERROR CONDITION
    [  1 - 7   ][     0 - F    ]  UNASSIGNED ERROR CODES
  1 [ SERIAL CHARACTER RECEIVED]  USRT RECEIVE WORD
ENTRY LOCATION: MONITOR IN POINT.MON


MONITOR EXIT CONDITIONS:
UPON EXIT FROM THE MONITOR THE ACCUMULATOR CONTAINS THE EXIT CODE.
[B7][B6][B5][B4][B3][B2][B1][B0]
  0   0   0   0   0   0   0   0   NOP
  0   0   0   0 [  EXIT  CODE  ]  USART SEND/RECEIVE ERROR CONDITION
                [       1      ]  RECEIVE PROTOCOL SYNTAX ERROR MSG
                [       2      ]  RECEIVE COMMAND SYNTAX ERROR MSG
                [       3      ]  RECEIVE OVERLAY ERROR MSG
                [       4      ]  RECEIVE PARITY ERROR MSG
                [       5      ]  RECEIVE OVERRUN ERROR MSG
                [       6      ]  TRANSMIT OVERRUN ERROR MSG
                [       7      ]  PROGRAM STATE ERROR MSG
                [       8      ]  EVENT QUEUE FULL
                [       9      ]  PROGRAM SEQUENCE RECYCLE
  0   0   0   1 [  EXIT  CODE  ]  CENTRAL CONTROL MESSAGE
                [       0      ]  SYSTEM RESET COMMAND (NOT AN ERROR)
EXIT LOCATION: RMONITOR IN POINT.SUP
```

SCHEDULAR ENTRY AND EXIT CONDITIONS:

ENTRY LOCATION: SCHED IN POINT.SCH
EXIT LOCATION: SCHSUPER IN POINT.SUP

ON ENTRY THE ACCUMULATOR CONTAINS A EXECUTION CODE
ENTRY CODES:

- 0. ACTIVE PROGRAM HAS ENDED
  - 1. EXECUTE ACTPRGDN
    - 1. FETCH NEXT ACTIVE PROGRAM IN THE SEQUENCE
    - 2. IF ACTIVE PROGRAM SEQUENCE DONE, DELETE RETURN FROM STACK
    - 3. IF ACTIVE PROGRAM DONE,
      - 1. SET ACTIVE PROGRAM DONE
      - 2. DELETE RETURN FROM THE STACK
    - 4. IF BAD STACK
      - 1. INACTIVE PROGRAM SET DONE
      - 2. INACTIVE END OF STACK=0
      - 3. STACK POINTER=0
  - 2. ACCUMULATOR CONTENTS ON EXIT:
    - 0. NORMAL EXIT
    - 1. ACTIVE PROGRAM SEQUENCE DONE
    - 2. ACTIVE AND INACTIVE PROGRAM SEQUENCES DONE
    - 3. ERROR, ACTIVE PROGRAM SEQUENCE ALREADY DONE
    - 4. BAD STACK
- 1. ACTIVE PROGRAM IS SUSPENDED, OR FINISHED
  - 1. EXECUTE ACTINACT
    - 1. SWAP ACTIVE PROGRAM FOR INACTIVE PROGRAM
    - 2. IF BAD STACK
      - 1. NO SWAP
      - 2. INACTIVE PROGRAM SET DONE
      - 3. INACTIVE END OF STACK=0
      - 4. STACK POINTER=0
  - 2. ACCUMULATOR CONTENTS ON EXIT:
    - 0. NORMAL EXIT
    - 1. NOT USED IN ACTINACT
    - 2. NOT USED IN ACTINACT
    - 3. NOT USED IN ACTINACT
    - 4. BAD STACK
- 2. RESET PROGRAM SEQENCES USING PRGLSTUO
  - 1. EXECUTE RESCHED
    - 1. IF B7=0 CLEAR THE BUFFER AREAS
    - 2. REINITIATE STACK
    - 3. RESET BCKPRG, AND FRGPRG
  - 2. ACCUMULATOR CONTENTS ON EXIT:
    - 0. NOT USED IN RESCHED
    - 1. NOT USED IN RESCHED
    - 2. ACTIVE AND INACTIVE PROGRAM SEQUENCES DONE
    - 3. NOT USED IN RESCHED
    - 4. NOT USED IN RESCHED

INTERRUPT ROUTINE CONDITIONS:

SYTEM RESET

EXIT IS MADE THROUGH THE TIMER INTERRUPT ROUTINE. F1 IS THE ONLY INDICATION A RESET HAS BEEN PERFORMED

```
  ENTRY LOCATION: 1. 000 HEX
ENTRY CONDITIONS: 1. RESET INTERRUPT
 EXIT CONDITIONS: 1. MEMORY BANK 0 SELECTED
                  2. TIMER INTERRUPTS ENABLED
                  3. TIMER ON
                  4. EXTERNAL INTERRUPTS ENABLED
                  5. USART INITIALIZED
                  6. USART CLOCK ON
                  7. PRGSTAT; ACTIVE AND INACTIVE PROGRAMS=DONE
                  8. STACK POINTER=0
                  9. RAM MEMORY CLEARED
                 10. DISABLE EXTERNAL INTERRUPT
                 11. EXTERNAL RAM CLEARED
                 12. EXTERNAL I/O CLEARED
   EXIT LOCATION: 1. RRESINT IN POINT.SUP
```

TIMER INTERRUPT

```
  ENTRY LOCATION: 1. 007 HEX
ENTRY CONDITIONS: 1. TIMER INTERRUPT
 EXIT CONDITIONS: 1. MEMORY BANK 0 SELECTED
                  2. REG BANK 1 SELECTED
                  3. TIMER INTERRUPT RESET
                  4. REG BANK 1 SELECTED
                  5. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                  6. EXTERNAL I/O OPERATIONS SELECTED
                  7. ACCUMULATOR=INTMSG5
   EXIT LOCATION: 1. RTIMINT IN POINT.SUP
```

INTERRUPT ROUTINE CONDITIONS CON'D

EXTERNAL INTERRUPT

```
      ENTRY LOCATION: 1. 003 HEX
    ENTRY CONDITIONS: 1. *INT=0, EXTERNAL INTERRUPT
                         AND
                      2. UST B1=1, USART CHARACTER RECEIVED
                         OR
                      3. UST B0=1, USART CHARACTER SENT
                         OR
                      4. STATUS INTERRUPT
                         1. STT B0 ^ PORT1 B4=1, COMPARISON INTERRUPT
                            OR
                         2. STT B1 ^ PORT1 B5=1, INDEX INTERRUPT
                            OR
                         3. STT B2 ^ PORT1 B6=1, ADC END INTERRUPT
                      5. TRAP INTERRUPT                ANSO B3=1
                         OR
                      6. FALSE INTERRUPT (NONE OF THE ABOVE)
     EXIT CONDITIONS: 1. MEMORY BANK 0 SELECTED
                      2. REG BANK 1 SELECTED
                      3. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                      4. EXTERNAL I/O OPERATIONS SELECTED
      EXIT LOCATION: 1. IN POINT.INT
                         1. USTREC (USART CHARACTER RECEIVED)
                            EXIT CONDITIONS:
                            1. ACCUMULATOR=UST
                            OR
                         2. USTXMT (USART CHARACTER TRANSMITTED)
                            EXIT CONDITIONS:
                            1. ACCUMULATOR=UST
                            OR
                         3. STTINT (STATUS INTERRUPT)
                            EXIT CONDITIONS:
                            1. ACC.=INTERRUPT DISABLE BIT=
                               (STT*16)^PORT1^F0H
                            OR
                         4. TRPINT (TRAP INTERRUPT)
                            EXIT CONDITIONS:
                            1. ACCUMULATOR=ANSO
                            2. ANSO(B3)=1
                         OR
                      2. RFLSINT (FALSE INTERRUPT) IN POINT.SUP
                         EXIT CONDITION: 1. ACCUMULATOR=0
```

INTERRUPT ROUTINE CONDITIONS CON'D

USTREC INTERRUPT, USART CHARACATER RECEIVED INTERRUPT

```
        ENTRY LOCATION: 1. USTREC, IN POINT.INT
      ENTRY CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                        2. ACCUMULATOR=UST
                           1. UST B1=1,          USART CHARACTER RECEIVED
                              AND
                           2. UST B4=1,          RECEIVE OVERRUN ERROR
                              OR
                           3. UST B3=1,          RECEIVE PARITY ERROR
                              OR
                           4. UST *B4 V *B3=1, URD=VALID RECEIVE CHARACTER
       EXIT CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                           AND
                        2. MESSAGE PREAMBLE OR ERROR
                           1. ACCUMULATOR=0, PREAMBLE
                              OR
                           2. ERROR
                              1. ACCUMULATOR=INTMSG2, PARITY ERROR
                                 OR
                              2. ACCUMULATOR=INTMSG3, OVERRUN ERROR
                           EXIT LOCATION: 1. RUSTERR IN POINT.SUP
                           OR
                        3. CHARACTER RECEIVED
                           1. ACCUMULATOR=B7+ASCII CHARACTER
                           EXIT LOCACTION: 1. RRECINT IN POINT.SUP
```

USTXMT INTERRUPT, USART CHARACATER SENT INTERRUPT

```
        ENTRY LOCATION: 1. USTXMT, IN POINT.INT
      ENTRY CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                        2. ACCUMULATOR=UST
                           1. UST B0=1, USART CHARACTER SENT
                        3. CHARACTER IN XMITCHR
                           1. XMITCHR=0 TERMINATE TRANSMISSION
                              OR
                           2. XMITCHR/=0 SEND XMITCHR
       EXIT CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                           AND
                        2. ACCUMULATOR=INTMSG1, CHARACTER SENT
                           OR
                        3. ACCUMULATOR=INTMSG4, TRAMISSION TERMINATED
         EXIT LOCATION: 1. RSNDINT, IN POINT.SUP
```

INTERRUPT ROUTINE CONDITIONS CON'D

TRPINT INTERRUPT, TRAP INTERRUPT

```
        ENTRY LOCATION: 1. TRPINT, IN POINT.INT
      ENTRY CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                           B1-B0. TYPE TRAP
                                0. FALSE OR INVALID TRAP
                                   B7-B2. UNASSIGNED
                                1. ACTION TRAP
                                      B2. UNASSIGNED
                                   B7-B3. TYPE ACTION TRAP
                                       0. END TRAP
                                       1. TIME/EVENT TRAP
                                          1. R2(RB0)=FUNCTION PARAMETER
                                          2. R3(RB0)=TIME PARAMETER
                                       2. ABORT TRAP
                                       3. END SEQUENCE TRAP
                                    4-31. UNASSIGNED
                                2. UNASSIGNED
                                   B7-B2. UNASSIGNED
                                3. WAIT TRAP
                                   B3-B2. UNASSIGNED
                                   B7-B4. TYPE OF WAIT TRAPS ISSUED
                                      B4. COMPARISON TRAP
                                      B5. INDEX TRAP
                                      B6. ADC END TRAP
                                      B7. EXTERNAL
                        2. ACCUMULATOR=ANS0
                        3. ANS0(B3)=1
         EXIT LOCATION: 1. RTRPINT, IN POINT.SUP
       EXIT CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                           AND
                        2. ACCUMULATOR=TRPMSG1, END TRAP
                           OR
                        3. TIME TRAP
                           1. ACCUMULATOR=TRPMSG2, ENTRY MADE
                             OR
                           2. ACCUMULATOR=B4+TRPMSG2, QUEUE FULL
                           OR
                        4. ACCUMULATOR=TRPMSG3, ABORT TRAP
                           OR
                        5. ACCUMULATOR=TRPMSG4, END SEQUENCE TRAP
                           OR
                        7. WAIT TRAP
                           1. ACCUMULATOR=[R3(RB0)^F0H]+TRPMSG7
                           2. PORT1=(PORT1^0FH)+[R3(RB0)^F0H]
                           3. P1=(PORT1^0FH)+[R3(RB0)^F0H]
```

INTERRUPT ROUTINE CONDITIONS CON'D

STTINT INTERRUPT, STATUS INTERRUPT

```
        ENTRY LOCATION: 1. STTINT, IN POINT.INT
      ENTRY CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                        2. ACCUMULATOR=INT. DISABLE BIT=(STT*16)^PORT1^FOH
                        3. R0=PORT1 ADDRESS
       EXIT CONDITIONS: 1. R3=ACCUMULATOR OF INTERRUPTED PROGRAM
                        2. PORT1=PORT1(+)INT.DIS
                        3. ACCUMULATOR=INT.DIS+B3+INTMSG7
                           B3:=PORT1^FOH=CONTINUEING INTERRUPT
        EXIT LOCATION: 1. RSTTINT, IN POINT.SUP
```

SUPERVISOR ENTRY CONDITIONS:

RRECINT, ENTRY AFTER USART RECEIVE INTERRUPT
UPON ENTRY TO THE SUPERVISOR THE ACCUMULATOR CONTAINS THE ENTRY CODE.
[B7][B6][B5][B4][B3][B2][B1][B0]
1 [SERIAL CHARACER RECEIVED] USRT RECEIVE WORD

RSTTINT, RSNDINT, RTIMINT, RUSTERR, AND RFLSINT
ENTRY AFTER STATUS, USART SEND, TIMER, USART ERROR, OR FALSE INTERRUPT
UPON ENTRY TO THE SUPERVISOR THE ACCUMULATOR CONTAINS THE ENTRY CODE.
[B7][B6][B5][B4][B3][B2][B1][B0]
<CONTEXT SENSITIVE->(-INT CODE->

ACCUMULATOR CODES
B2-B0. TYPE INTERRUPT
0. NO ACTION REQUIRED, OR FALSE INTERRUPT
1. SERIAL WORD SENT INTERRUPT
2. RECEIVE PARITY ERROR
3. RECEIVE OVERRUN ERROR
4. SEND MESSAGE TERMINATED
5. TIMER INTERRUPT
6. UNASSIGNED
7. STATUS INTERRUPT
B3. INDICATES A CONTINUING STATUS INTERRUPT
B7-B4. TYPE STATUS INTERRUPT
B4. COMPARISON INTERRUPT
B5. INDEX INTERRUPT
B6. ADC END INTERRUPT
B7. EXTERNAL INTERRUPT

RTRPINT
SUPERVISOR ENTRY POINT AFTER A TRAP
[B7][B6][B5][B4][B3][B2][B1][B0]
<-CONTEXT SENS.)<--TRAP CODE--->

B2-B0 TYPE TRAP
0. FALSE TRAP
1. END TRAP
2. TIME TRAP
B7-B4. TIME TRAP CONDITION CODE
0. ENTRY MADE
1. QUEUE FULL
3. ABORT TRAP
4. END SEQUENCE TRAP
5-6. UNASSIGNED TRAP
7. WAIT TRAP TRAP
B7-B4. WAIT TRAP PARAMETER
B4. COMPARISON TRAP
B5. INDEX TRAP
B6. ADC END TRAP
B7. EXTERNAL TRAP
B3 =0

SUPERVISOR ENTRY CONDITIONS CON'D:

SCHSUPER
ENTRY AFTER SCHECULAR ROUTINE
ACCUMULATOR CODES:
0. NORMAL EXIT
1. ACTIVE PROGRAM SEQUENCE DONE
2. ACTIVE AND INACTIVE PROGRAM SEQUENCES DONE
3. ACTIVE PROGRAM SEQUENCE DONE
4. BAD STACK

RMONITOR
ENTRY AFTER MONITOR ROUTINE
ACCUMULATOR CODES:

[B7]	[B6]	[B5]	[B4]	[B3]	[B2]	[B1]	[B0]	
0	0	0	0	0	0	0	0	NOP
0	0	0	0	[EXIT	CODE]			USART SEND/RECEIVE ERROR CONDITION
				[	1]			RECEIVE PROTOCOL SYNTAX ERROR MSG
				[	2]			RECEIVE COMMAND SYNTAX ERROR MSG
				[	3]			RECEIVE OVERLAY ERROR MSG
				[	4]			RECEIVE PARITY ERROR MSG
				[	5]			RECEIVE OVERRUN ERROR MSG
				[	6]			TRANSMIT OVERRUN ERROR MSG
				[	7]			PROGRAM STATE ERROR MSG
				[	8]			EVENT QUEUE FULL
				[	9]			PROGRAM SEQUENCE RECYCLE
0	0	0	1	[EXIT	CODE]			CENTRAL CONTROL MESSAGE
				[	0]			SYSTEM RESET COMMAND (NOT AN ERROR)

EXIT LOCATION: RMONITOR IN POINT.SUP

SYSTEM REFRESH OPERATIONS

CERTAIN ITEMS REQUIRE PERIODIC REFRESHMENT IN ORDER TO AVOID LOOSING CERTAIN FUNCTIONS OR HAVING A SYSTEM CRASH.

1. CANCEL RESET FUNCTION
   BIT P26 IS COMPLIMENTED IN ORDER TO RETRIGGER THE RESET MONOSTABLE AND ABORT A SYSTEM RESET.
   R. THE BIT IS COMPLIMENTED ON TIME ENTRY TO THE MONITOR
2. ENABLE EXTERNAL INTERUPTS
   R. ENABLED ON EACH TIME ENTRY TO THE MONITOR
3. ENABLE TIMER INTERRUPTS
   R. ENABLED ON EXIT FOR SUPERVISOR
4. ENABLE CLOCK TO USART
   R. ENABLED ABOUT EVERY 32 SECONDS FROM MONITOR, WHEN TIME EQUALS X800 HEX.
5. INITIALIZE USART
   R. INITIALIZED ABOUT EVERY 34 SECONDS FROM MONITOR, WHEN TIME EQUALS X000 HEX, IF NO USART RECEIVE ACTIVITY, OR IF SEND ACTIVITY HAS BEEN DETECTED IN THE PRIOR 34 SECOND PERIOD AND THERE HAS NOT BEEN A SUBSEQUENT SEND TERMINATION

POINT INTERNAL RAM MEMORY ALLOCATION

```
+-------------------------------------------------------------------------------------+
!0. R0 ACT   ! REG.!20. PORT #1, PORT1!40.-FORWRK---- +0-!60.-EVEQUE------0-!
!1. R1 ACT   ! BANK!21. VECTOR        !41.----------- +1-!61.          !     !
!2. R2 ACT   ! 0   !22. *             !42.-MSGBUFF0-- +0-!62. T/E,A    !  1  !
!3. R3 ACT   !     !23. USART XMITCHR !43.----------- +1-!63. OPER     !     !
!4. R0'INACT! R4   !24. TIMER TIMVALL !44.-FRBUFF0--- +0-!64. T/E,A    !  2  !
!5. R1'INACT! R5   !25. TIMER TIMVALH !45. FOREGROUND +1 !65. OPER     !     !
!6. R2'INACT! R6   !26. SNDRECST      !46. BUFFER     +2 !66. T/E,A    !  3  !
!7.-R3'INACT--R7--!27. SRCOMMSG      !47. UNIT #0    +3 !67. OPER     !     !
!8.-TOP-----------!28. SRPARM1       !48.            +4 !68. T/E,A    !  4  !
!9.          !     !29. SRPARM2       !49.            +5 !69. OPER     !     !
!A.          !     !2A. PRGLSTUO      !4A.            +6 !6A.          !  5  !
!B.          !     !2B. PRGLSTAC      !4B.            +7 !6B.          !     !
!C.          !     !2C. DROPIN        !4C.            +8 !6C. EVENT QUEUE 6 !
!D.          !     !2D. DROPOUT       !4D.----------- +9-!6D. 2BYTES PER    !
!E.          !     !2E. WINDOW        !4E.-BKBUFF0--- +0-!6E. ENTRY    !  7  !
!F.          !     !2F.-MSGBUFF--- +0-!4F. BACKGROUND +1 !6F.          !     !
!10.         !STACK!30.----------- +1-!50. BUFFER     +2 !70.TIME/EVENT,ADDR!
!11.         !     !31.ICPRG:UNT#PROG#!51. UNIT #0    +3 !71.OPERAND   !     !
!12.         !     !32.ACPRG:UNT#PROG#!52.            +4 !72.          !  9  !
!13.         !     !33.-ACTEXAR--- +0-!53.            +5 !73.          !     !
!14.         !     !34. ACTIVE     +1 !54.            +6 !74. T/E,A    !  A  !
!15.         !     !35. PROGRAM    +2 !55.            +7 !75. OPER     !     !
!16.         !     !36. EXECUTION  +3 !56.            +8 !76. T/E,A    !  B  !
!17.-BOTTOM-------!37. AREA       +4 !57.----------- +9-!77. OPER     !     !
!18.-R0-SYS-------!38.            +5 !58. *              !78. T/E,A    !  C  !
!19. R1 SYS ! REG !39.            +6 !59. *              !79. OPER     !     !
!1A. R2 SYS ! BANK!3A.            +7 !5A. *              !7A. T/E,A    !  D  !
!1B. R3 SYS ! 1   !3B.            +8 !5B. *              !7B. OPER     !     !
!1C.ACTACC  ! R4  !3C.----------- +9-!5C. *              !7C. T/E,A    !  E  !
!1D.INACTACC! R5  !3D.-BCKWRK---- +0-!5D. *              !7D. OPER     !     !
!1E.PRGSTAT ! R6  !3E.            +1 !5E. *              !7E. T/E,A    !  F  !
!1F.OPSTAT  ! R7  !3F.----------- +2-!5F. *              !7F.-OPER----------!
+-------------------------------------------------------------------------------------+
```

* UNASSIGNED

Fig. 6.1-2

PRGSTAT PROGRAM STATUS ADDR. 1E

```
[ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
( F/B)(DONE)( PROG STAT)(UINT)(F/B END OF STACK)
```

B0-B2	F/B END OF STACK END OF STACK FOR INACTIVE PROGRAM
B3	UNINT FOREGROUND UNINTERRUPTABLE
B4-B5	PROGRAM STATUS, STATUS OF INACTIVE PROGRAM 0. FINISHED, INDICATES THAT INACTIVE PROGRAM HAS ENDED. 1. SUSPENDED 2. WAITING 3. SCHEDULED
B6	DONE, SCHEDULAR INDICATES ACTIVE PROGRAM SEQUENCE HAS ENDED.
B7	F/B INDICATES ACTIVE PROGRAM 0. BACKGROUND ACTIVE 1. FOREGROUND ACTIVE

RAM CON'D

OPSTAT OPERATING SYSTEM STATUS ADDR. 1F

```
[ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
<---UNASSIGNED--->(RECI)(SNDA)S.MODE(CHAR MODE )
```

B0-B1 RECEIVE MODE
- 0. 1ST ADDRESS CHARACTER RECEIVE MODE
- 1. 2ND ADDRESS CHARACTER RECEIVE MODE
- 2. MESSAGE CHARACTER RECEIVE MODE
- 3. CHARACTER AVAILABLE RECEIVE MODE

B2 SEND MODE
- 0. SEND MODE IDLE
- 1. CHARACTER SENT

B3 SEND ACTIVITY
USED TO GAURANTEE THAT SEND ACTIVITY TERMINATES
- 0. BIT CLEARED BY SEND TERMINATION
- 1. BIT SET BY MONITOR WHEN SEND ACTIVITY DETECTED

B4 RECEIVOR IDLE
USED FOR LONG RECEIVOR IDLE PERIODS TO GUARANTEE THAT THE RECEIVER IS NOT DEAF.
- 0. BIT CLEARED BY MONITOR
- 1. BIT SET BY RECEIVER ACTIVIY

ACPRG AND ICPRG ADDR. 32 AND 31

ONE BYTE EACH, CONTAINING THE CURRENT UNIT NUMBER AND PROGRAM NUMBER OF THE ACTIVE OR INACTIVE PROGRAM SEQUENCES. THE PROGRAM NUMBER IS A RELATIVE NIBBLE ADDRESS INTO EITHER FRGLST OR BCKLST.

ACPRG AND ICPRG

```
[ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
<------UNIT NUMBER-----><----PROGRAM NUMBER---->
```

PRGLSTUO ADDR. 2A
PRGLSTAC ADDR. 2B

PRGLSTUO IS MOVED TO PRGLSTAC AT THE END OF A PROGRAM CYCLE, PERMITTING PROGRAM MODULES TO CHANGE THE PROGRAM SEQUENCE IDENTIFICATION OF THE PROGRAM LIST THAT WILL BE USED TO FILL FRGLST AND BCKLST. PRGLST IS LOCATED IN PAGE 3 OF PROGRAM MEMORY.

```
[ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
<NIR )( NA )<-----------PRGLST ID--------------->
```

B0-B6. RELATIVE ADDRESS INTO PAGE 3 OF PROGRAM MEMORY FROM THE START OF THE PROGRAM LIST. FOREGROUND AND BACKGROUND SECTIONS ARE TERMINATED BY A NULL ENTRY.

B6. NOT ASSIGNED

B7. NOT INITIAL RUN
- 0. INITIAL RUN OF THE PROGRAM LIST, AS LOADED BY THE MONITOR.
- 1. SET TO ONE AFTER THE FIRST CYCLE BY THE SCHEDULAR

RAM CON'D

ACTEXAR ADDR. 33-37
FIVE GENERAL PURPOSE BYTES USED BY THE ACTIVE PROGRAM DURING EXECUTION AS RAM MEMORY. THIS AREA AT THE TIME OF EXECUTION BELONGS TO A PARTICULAR UNIT# AND EITHER THE FOREGROUND OR BACKGROUND PROGRAM.
THE AREAS ARE CLEARED BY THE SCHEDULAR BEFORE THE 'FIRST' EXECUTION CYCLE OF THE PROGRAM SEQUENCE, AND MAY BE TESTED BY THE FOREGROUND OR BACKGROUND PROGRAM TO INDICATE THIS CONDITION.

FOREBUFFO AND BCKFUFFO ADDR. 56-5A AND 5B-5E
UNIT 0 FOREGROUND AND BACKGROUND BUFFER AREAS. THE APPOPRIATE AREA IS LOADED BY THE SCHEDULAR FROM THE ACTEXAR AT THE TERMINATION OF THE RESPECTIVE PROGRAM SEQUENCE. THESE AREAS MAY BE USED FOR FOREGROUND/BACKGROUND COMMUNICATION. THE AREAS ARE CLEARED BY THE SCHEDULAR BEFORE THE 'FIRST' EXECUTION CYCLE OF THE PROGRAM SEQUENCE AND MAY BE TESTED BY THE FOREGROUND OR BACKGROUND PROGRAM TO INDICATE THIS CONDITION.

MSGBUFF ADDR. 2F-30
MESSAGE BUFFER SET BY THE OPERATING SYSTEM OR CENTRAL CONTROL UNIT AS A MESSAGE TO THE FOREGROUND OR BACKGROUND PROGRAM. CURRENTLY WITH ONLY ONE FIELD UNIT MSGBUFFO IS NOT BEING USED. THE MESSAGE BUFFER IS USED AND CLEARED BY THE APPLICATION PROGRAM.

MSGBUFFO ADDR. 53-54
MESSAGE BUFFER ASSOCIATED WITH UNIT#1 IN A MULTI UNIT ENVIROMENT, NOT CURRENTLY USED.

EVENQUE ADDR. 60-7F
EVENT QUEUE HAVING SIXTEEN ENTRIES MAXIMUM, TWO BYTES PER ENTRY
TIMER/EVENT FUNCTION BYTE #1

```
[ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
<---------EVENT QUEUE ENTRY---------><L/H><T/E>
B0-B7. =0 EMPTY ENTRY
B0.    TIME/EVENT    EVENT MODE NOT CURRENTLY SUPPORTED
       0. TIME FUNCTION
       1. EVENT FUNCTION
B1.    LOW/HIGH TIME MODE
       0. 8 MILLESECOND RESOLUTION
       1. 2.048 SECOND RESOLUTION
B2-B7. RELATIVE TABLE ENTRY INTO EVENTQUE
       EVENTQUE IS A TABLE CONTAINING THREE BYTES PER ENTRY,
       DEFINING THE ACTION TO BE TAKEN AFTER A TIMER/EVENT
       OCCURANCE
```

TIME/EVENT OPERAND BYTE #2

```
ONE BYTE OF TIME OPERAND, TIME RANGE DETERMINED BY BYTE #1, B1
1. 8-2,048        MSEC,.8    'MSEC RESOLUTION, +0-8 MS ACCURACY
2. 2,048-524.288 SECS, 2.048 SECS RESOLUTION, +0-8 MS ACCURACY
```

RAM CON'D

```
SNDRECST                                    ADDR. 40
      SEND/RECEIVE STATES
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <SRM >(----SNDSTATE----)(-SUBSTATE-)(-RECSTATE-)
      B1-B0. RECEIVE PROGRAM STATE
             0. PROMPT MODE
             1. COMMAND MODE
             2. PARAMETER MODE
             3. FINISHED MODE
      B3-B2  SUBSTATE
             SEND STATE 2
                       0. EXECUTE
                       1. 1ST NIBBLE REPLY
                       2. 2ND NIBBLE REPLY
                       3. ETX SEND STATE
      B6-B4. SEND PROGRAM STATE
             0. IDLE MODE
             1. ACKNOWLEDGE MODE
             2. EXECUTE MODE
             3. TERMINATE TEXT MODE
             4. TERMINATE MESSAGE MODE
             5. TERMINATE SEND MODE
      B7.    SRCOMMSG MODE
             0. COMMAND MODE
             1. MESSAGE MODE
SRCOMMSG                                    ADDR. 41
      SEND/RECEIVE ERROR MESSAGE BITS
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <C/E >(-----------CONTEXT SENSITIVE------------)
      COMMAND MODE
             IF B3 OF SNDRECST=0
                       RECEIVE MODES 1 AND 2
                           B2-B0. NUMBER OF RECEIVE PARAMETERS TO BE STORED
                       SEND MODE 2
                           B4-B0. 2ND NIBBLE REPLY MESSAGE
                           B7-B4. COMMAND OPCODE
             IF B3 OF SNDRECST=1
                       B3-B0. HIGH ORDER NIBBLE OF REPLY
      MESSAGE MODE
                  B7-B4. TYPE MESSAGE MODE
                       0.  ERROR MESSAGE MODE
                       B3-B0. MESSAGES
                           0.      NO OP
                           1.      RECEIVE PROTOCOL SYNTAX ERROR MSG
                           2.      RECEIVE COMMAND SYNTAX ERROR MSG
                           3.      RECEIVE OVERLAY ERROR MSG
                           4.      RECEIVE PARITY ERROR MSG
                           5.      RECEIVE OVERRUN ERROR MSG
                           6.      TRANSMIT OVERRUN ERROR MSG
                           7.      PROGRAM STATE ERROR MSG
                           8.      EVENT QUEUE FULL
                       1.  CENTRAL CONTROL MESSAGE MODE
                           0.      SYSTEM RESET COMMAND (NOT AN ERROR)
```

RAM CON'D

```
PORT1                                          ADDR. 20
      RAM IMAGE OF PORT1
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <-------------PORT #1 DEFINITION---------------->
VECTOR                                         ADDR. 21
      RAM IMAGE OF PORT1
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <-------GENERAL PURPOSE ADDRESS VECTOR--------->
COMREG                                         ADDR. 22
      RAM IMAGE OF USART COMMAND REG (UCM)
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <--------------UCM DEFINITION----------------->
XMITCHR                                        ADDR. 23
      USART ASCII TRANSMIT CHARACTER
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <-NA-><-----------ASCII CHARACTER------------->
TIMVALL                                        ADDR. 24
      LOW BYTE TIMER
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <--------------------TIME---------------------->
TIMVALH                                        ADDR. 25
      HIGH BYTE TIMER
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <--------------------TIME---------------------->
SRPARM1                                        ADDR. 42
      SEND/RECEIVE ERROR MESSAGE BITS
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <------------SEND/RECEIVE PARAMETER------------>
SRPARM2                                        ADDR. 43
      SEND/RECEIVE ERROR MESSAGE BITS
      [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
      <------------SEND/RECEIVE PARAMETER------------>
```

POINT OPERATING SYSTEM PROGRAM MEMORY MAP

SUBMODULE/ MODULE	MODULE, AS REQUIRED	ADDRESSES BEGIN	END	(HEX) TOTAL	AVAIL. MEMORY
POINT.INT		0000	0131	0131	----
POINT.SCH		0140	02C6	0186	000F
PAGE3	SUBPG3	0300	0400	0100	003A
POINT.MON		0400	05E0	01E0	0000
POINT.SUP		0640	06F6	00B6	0060
SUBROUTINES	SUBPG3	071E	0800	00E2	0028
SYSTEM PROM			SUBTOTAL	072F	00D1
PAGEB	ASBPG3	0B00	0C00	0100	0300
SUBROUTINES	ASBPG3	0EEB	1000	0115	02EB
APPLICATION PROM			SUBTOTAL	0215	05EB
TOTAL OPERATING SYSTEM			TOTAL	0944	06BC

NOTE: PAGE3 AND PAGEB ARE ONLY PARTIALLY USED.

Fig. 6.1-3

STACK LIMITATIONS: FOREGROND, BACKGROUND, AND SYSTEM PROGRAM

1. SUBROUTINES MAY ONLY BE NESTED TWO DEEP IN F/B/SUPER/SCHED/MON.
2. SUBROUTINES MAY ONLY BE ONE DEEP IN INTERRUPT ROUTINES.
3. INTERRUPTS ARE ENABLED IN FOREGROUND, BACKGROUND, AND SCHEDULAR, EXCEPT SHORT FOREGROUND PROGRAMS WHICH MAY BE SPECIFIED UNINTERRUPTABLE.
4. INTERRUPTS IN THE MONITOR MAY ONLY BE ENABLED WHEN NO SUBROUTINES ARE BEING EXECUTED, AND WHEN THE ACCUMULATOR, REGISTERS, AND F1 ARE NOT BEING USED, ONLY THE CONDITION CODES ARE SAVED ON THE STACK.
5. A WAIT TRAP MAY ONLY BE EXECUTED IN THE FOREGROUND.

PRGTABL PROGRAM TABLE

TABLE OF FOREGROUND AND BACKGROUND STARTING ADDRESSES. THE PROGRAM ID FORMS A RELATIVE ADDRESS INTO THE TABLE.

PROGRAM MEMORY ADDRESS=300H, PAGE 3 OF PROGRAM MEMORY.

```
    !<-------------------TWO BYTES-------------------->!
    !-----------------------------------------------------!
ID  !    PROGRAM CODE     ! HIGH ORDER ADDR! LOW ORDER ADDR! RELATIVE
    !-------------------+----------------+---------------! ADDRESS
    !<D15><D14><D13><D12>!<A11><----->(A8)!<A7><----->(A0)!
00  !          0          ! BCKGND EOF  0  !       0       ! 00
 .  !  0    T    T    T  ! BACKGROUND     !               !  .
 .  !                     ! PROGRAM        !               !  .
0F  !                     ! ENTRIES        !               ! 1E
    !-------------------+----------------+---------------!
10  !          0          ! FRGND EOF   0  !       0       ! 20
 .  !  U    T    T    T  ! FOREGROUND     !               !  .
 .  !                     ! PROGRAM        !               !  .
1F  !                     ! ENTRIES        !               ! 3E
    +-----------------------------------------------------+
```

Fig. 6.1-4

A0-A11 PROGRAM STARTING ADDRESS

D12-D14 BACKGROUND, NUMBER OF BYTES TO BE TRANSFERRED FROM THE FOREGROUND OR BACKGROUND BUFFER AREA FRBUFF(N)/ BKBUFF(N) TO THE OPPOSITE COM AREA BKGNDCOM OR FRGNDCOM FOR THE FIRST PROGRAM OF A FOREGROUND OR BACKGROUND PROGRAM SEQUENCE.

D15 1. U: FOREGROUND PROGRAM UNINTERRUPTABLE
2. BACKGROUND UNASSIGNED = 0

ID00-10 EMPTY ENTRIES. USED FOR EOF PROGRAM SEQUENCE

EVENTQUE EVENT QUEUE EXECUTION TABLE

TABLE OF SINGLE OPERATIONS TO BE EXECUTED UPON THE OCCURANCE OF AN EVENT IN THE EVENT QUEUE.

PROGRAM MEMORY = PAGE 3 OF APPLICATION PROGRAM MEMORY.

```
   !<--------------------THREE BYTES----------------------------!
   !-------------------------------------------------------------!
   ! BYTE #3      ! BYTE #2      ! BYTE #1                       !
ID ! DESTINATION  ! SOURCE       ! FUNCTION                      ! RELATIVE
   !--------------+--------------+-------------------------------! ADDRESS
   !<A8-------A0>!<A8-------A0>!<D7--D6><D5--D4><D3--D2><D1--D0>!
00 !  NOT VALID   !  NOT VALID   !          NOT VALID            !
01 !  ADDRESS     !ADDR.OR IMED.!  DEST.   SRC.     N.A.   OP CODE! 00
02 !              !              !                                ! 03
 . !              !              !                                !  .
 . !              !              !                                !  .
   +-------------------------------------------------------------+
```

Fig. 6.1-5

ID 00 IS NOT PERMITTED SINCE THE MONITOR USES A NULL ENTRY TO INDICATE AN EMPTY QUEUE ENTRY.

```
BYTE #1  FUNCTION
      D1-D0. OP CODE
          0. MOVE
          1. OR
             1. NOT APPROPRIATE FOR PORT2 DESTINATION
          2. AND
             1. NOT APPROPRIATE FOR PORT2 DESTINATION
          3. XOR
             1. NOT APPROPRIATE FOR PORT2 DESTINATION
      D3-D2. NOT ASSIGNED
      D4-D5. SOURCE MODE
          0. I/O
          1. EXTERNAL RAM
          2. INTERNAL RAM
          3. IMMEDIATE, THE VALUE IS CONTAINED IN BYTE2
             PAGE THREE WHERE THE IMMEDIATE OPERAND IS LOCATED
      D6-D7. DESTINATION MODE
          0. I/O
          1. EXTERNAL RAM
          2. INTERNAL RAM
          3. PORT 1 OR 2
BYTE #2  SOURCE ADDRESS OR IMMEDIATE DATA
BYTE #3  DESTINATION ADDRESS
          1. IF PORT DESTINATION
             1. PORT 1 IF BYTE #3=0
             2. PORT 2 IF BYTE #3 NOT=0
```

POINT CENTRAL CONTROL COMMAND

OPCODE HEX	NUM. PARM	2ND PARM. SRPARM1	1ST PARM SRPARM2	NMEMONIC	ACTION TAKEN
00	0			COMNOP	NO OP
01	0			SYSRES	SYSTEM RESET
02	2	ADDRESS	READ CNT	RIRAM	READ INTERNAL RAM
03	2	ADDRESS	DATA	WIRAM	WRITE INTERNAL RAM
04	2	ADDRESS	READ CNT	RXRAM	READ EXTERNAL RAM
05	2	ADDRESS	DATA	WXRAM	WRITE EXTERNAL RAM
06	2	ADDRESS	READ CNT	RXIO	READ EXTERNAL I/O
07	2	ADDRESS	DATA	WXIO	WRITE EXTERNAL I/O
08	1	PORT #	DATA	OPRT	OR PORT
09	2	PORT #	DATA	APRT	AND PORT
0A	2	FUNCTION	OPERAND	WEVQ	EVENT QUEUE ENTRY
0B	0			RPSW	READ PSW

Fig. 6.1-6

POINT SEND/RECEIVE TIMING

MINIMUM BIT SYNCHRONIZATION IS ACHIEVED BY RECEIVING A BINARY 01, OR 10 IN THE BIT STREAM. IN PRACTICE PROPER BIT SYCHRONIZATION IS ACQUIRED BY INSERTING A MINIMUM OF ONE ASCII NAK CHARACTER AS A FILLER WHICH HAS SUFFICIENT ALTERNATE ONES AND ZEROES BEFORE ANY SYNC. CHARACTERS. THE COMBINATION OF ALTERNATE ONES AND ZEROES ELIMINATES ANY LOGIC TRANSITIONS AT BIT BOUNDARIES, AND GAURANTEES UNAMBIGUOUS BIT SYNC. THE TRANSMISSION FROM THE CENTRAL CONTROLLER TO THE FIELD UNIT USES THE STANDARD 7 BIT ASCII TRANSMISSION CONVENTION. STX IS USED AS THE MESSAGE SYNC CHARACTER. MESSAGES ARE TERMINATED WITH ETX. EOT IS USED TO RETURN THE USART TO THE HUNT MODE.

MESSAGE SYNTAX, CENTRAL CONTROLLER TO FIELD UNIT

SINCE THE COMMUNICATION IS SYNCHRONOUS IN NATURE "NAK" CHARACTERS MAY BE UNPREDICABLY INSERTED BY THE USART INTO THE MESSAGE STREAM, THE FIELD UNIT MUST THEREFORE IGNORE THEM.

```
                !<--UNIT ADDRESS--->!<--------COMMAND------->!


NAK,..,NAK,STX,H HEX ADD,L HEX ADD,$,C(HEX COMMAND)
                ,ASCII OR L HEX PARM,A OR H PARM,....,$,C....,ETX,EOT,NAK..


$           COMMAND OR MESSAGE DELIMITER
HEX CHAR    ASCII 3N, WHERE N IS A HEXIDECIMAL NUMBER
```

FIELD UNIT REPLY MESSAGE SYNTAX

SINCE THE COMMUNICATION IS SYNCHRONOUS IN NATURE "NAK" CHARACTERS MAY BE UNPREDICABLY INSERTED BY THE USART INTO THE MESSAGE STREAM, THE CENTRAL CONTROLLER MUST THEREFORE IGNORE THEM.

```
  MSG := "NAK",REPLY                  SEND NAK REPLY
REPLY := "ACK",REPLY                  NORMAL COMMAND ACKNOWLEDGE, NO REPLY
REPLY := "ACK",PARM                   NORMAL COMMAND ACKNOWLEDGE, WITH REPLY
REPLY := TERM                         END MESSAGE TERMINATE
 PARM := 2HEX,PARM                    SEND PARAMETER,GET NEXT PARAMETER
 PARM := 2HEX,"ETX",REPLY             REPLY PARAMETERS,TERMINATE TEXT
 TERM := "ACK","EOT","NULL"           NORMAL MESSAGE TERMINATION
 TERM := "CAN","EOT","NULL"           ERROR MESSAGE TERMINATION
 2HEX := LNIB,HNIB                    BYTE REPLY
 LNIB := 3,HEX CHAR                   LOW ORDER NIBBLE
 HNIB := 3,HEX CHAR                   HIGH ORDER NIBBLE
```

MESSAGE SYNCHRONIZATION AT THE CENTRAL CONTROLLER IS ACHIEVED BY THE FIELD CONTROLLER SENDING TWO CONSECUTIVE NAK'S. BIT SYNCHRONIZATION AT THE CENTRAL CONTROLLER RECEIVER IS GAURANTEED SINCE THE TRANSMISSION CLOCK IS GENERATED THERE FOR THE ENTIRE INSTALLATION, AND THE FIELD CONTROLLERS DECODE THIS CLOCK FROM THE SEND LINE. DUE TO FILTERING THERE IS HOWEVER A CONSTANT TIME DELAY BETWEEN THE SEND BIT STREAM AND THE RECEIVE BIT STREAM.

```
+-------------+          RECEIVE LINE
!             !<------*----------------*------------<
!  CENTRAL    !       ^                ^              ADDITIONAL UNITS
! CONTROLLER  !       ! SEND LINE      ¬
!             !-------------*----------------*-------->
+-------------+       !    !            !    !
                      !    V            !    V
                 +-------------+   +-------------+
                 !   FIELD     !   !   FIELD     !
                 !   UNIT      !   !   UNIT      !
                 !   #1        !   !   #2        !
                 +-------------+   +-------------+
```

Fig. 6.1-7

SEND/RECIEVE FIELD CONTROLLER

SINCE THE SEND/RECEIVE IS INTERRUPT DRIVEN, THE MONITOR MUST INITIATE SEND MESSAGES BY MAKING A CALL TO XMITINT. THE INTERUPT ROUTINE NOW CONTINUES THE TRANSMISSSION. THE SUPERVISOR WILL REQUEST NEW CHARACTERS TO BE LOADED IN XMITCHR AS REQUIRED. THE MESSAGE IS TERMINATED BY LOADING XMITCHR WITH ZERO.

6.2 POINT APPLICATION PROGRAM DESCRIPTION

6.2 GENERAL:

Each specific operation mode is a sequence of individual programs operating in the foreground and background, that give the desired results, after execution. Each sequence repeats indefinately until changed by a reset condition, central controller, or a program module.

These programs are assembled and linked at locations 800-1000 hex. Page B (B00-BFF) is reserved for tables that can be used by the SUPERVISOR and program modules.

A program sequence is an ordered sequence of program modules used to achieve a desired result.

The program module is the fundamental program unit. In different program sequences the order, number, and specific modules are changed to modify system performance.

These are rotary sequences that periodically repeat. A programable number of parameters (7 max) may be transferred from the foreground portion of the sequence to the background, and also from background to foreground.

The application program modules also have a message buffer two bytes long MSGBUFF0 used for principly semaphores, but also can be used for parameter storage.

Each program module additionally owns the accumulator and R0-R3, this is in additon to ACTEXAR, and may also be used to pass parameters between program modules within the foreground or background portion of the program sequence.

The 8048 family of microprocessors has a rotary stack limited to eight levels, and does not permit pushing of parameters onto it. The application programs are limited, by necessity, to nesting of subroutines to two deep.

6.2.1 PROGRAM SEQUENCES

Program Sequence #0	Power Up or Idle program sequence loads default parameters on initial pass, but otherwise idles.
Program Sequence #1	System Initialization to perform tracking. The program sequence will automatically switch to Program Sequence #4 if there is sufficient insolation for tracking, if not it will execute Program Sequence #2 (Clock Mode), and if there is a fault condition it will return to Program Sequence #0.
Program Sequence #3	Stow concentrator.

APPLICATION PROGRAM STATE DIAGRAM

```
                                 'IPRG'
                                 -----------   in stow
                     PROGRAM   ! INITIAL    !<------------------
                     SEQUENCE  ! PROGRAM    !  Central Command  !
                        0      !            !--------------     !
                                 -----------                 !    !
                                      !                      !    !
     <-------PROGRAM SEQUENCE 1---------->                    !    !
     'RDSNR'                          ! Central Command      !    !
     'FILTERS'          'ONEPASS'   V            'NPRG'     V    !
      -----------  warmup      -----------                 ----------
     !DATA        ! done       ! EQUIPMENT !              ! STOW     ! PROGRAM
     !ACQUISITION!<----------! INITIAL-  !              ! CONCEN-  ! SEQUENCE
     !FILTERING   !            ! IZATION   !              ! TRATOR   !    3
      -----------               -----------                 ----------
     Set    !          End Seq    !      ^ warmup            !     ^ not in
     Prog.  !          Trap       !      ! not               !     ! stow
     Seq.   !                      -----  completed          ----  position
      2     !
     <--------PROGRAM SEQUENCE 4--------->
 'RDSNR',   !
 'FILTERS' V                    'TRACK'
      -----------               -----------    ^
     !DATA        !<----------!            !   !
     !ACQUISITION!            !  TRACK     !   !
     !FILTERING   !---------->!            !   !
      -----------               -----------    !
        ^      !   SUNF>DROPIN              PROGRAM
        !      !                            SEQUENCE 2
'CLKDR' !      V   SUNF<DROPOUT                !
      -----------                              !
     ! EPHEMERIS !                             !
     ! TRACK     !                             !
     !           !                             !
      -----------                              V
```

Fig. 6.2.1-1

6.2.2 APPLICATION PROGRAM MODULE DESCRIPTIONS

IPRG - Initial Program Foreground Module

Description:

Initialization is performed.

Algorithm:

If initial cycle,

Load default values into DROPIN, DROPOUT, and WINDOW.

Otherwise,

NPRG - Night Program Foreground Module

Description:

Stows the concentrator.

Algorithm:

If not in stow position,

Run positions motor in direction of stow

Otherwise,

Set Program Sequence 0,

Position motor is shut OFF.

ONEPASS - One Pass Foreground Module

Description:

Setup conditons to perform tracking.

Algorithm:

If initial cycle,

Run Scan Motor,

Issue Time Trap to set semaphore after delay period.

If not initial cycle and NO semaphore,

Issue End Sequence Trap to truncate Program Sequence 1.

Otherwise,

Set Passive Mode,

Issue Wait Trap for Index.

Add 180 degs to Index value, and load Comparison Register.

Set Program Sequence 2.

RDSNR - Read Sensors Foreground Module

Description:

Acquires angular position, and insolation level data.

Algorithm:

Set Passive State,

Wait for Active State by issuing Comparison Trap,

Wait for Passive State by issuing Comparison Trap,

Add 180 degs to Index value, and load Comparison Register,

Convert Address Latch data to relative position,

Add position OFFSET,

Perform Analog to Digital conversion on Insolation Level

APPLICATION PROGRAM MODULE FUNCTIONS con'd

FILTERS - Digital Data Filters Background Module

Description:

Filters Angular Position Data and Insolation Level

Algorithm:

```
If Insolation Level not within limits,
   Set overflow value in SUNF,
   Set BETAF equal to zero
If Insolation level NOT equal to OVERFLOW
   Filter angular position data, generate BETAF,
   Filter insolation level, generate SUNF
```

TRACK - Track Concentrator Background Module

Description:

Track concentrator to within WINDOW value.

Algorithm:

```
If SUNF < DROPOUT,
   Set Program Sequence 2,
   Exit
If Manual Mode
   Stop positioning motor,
   Exit.
If BETAF > Continous Run Limit
   Run positioning motor in appropriate direction,
   Exit
If Position Motor ON,
   If no Time Event pending (error)
      Stop motor,
      Exit
   Otherwise,
      Exit
Otherwise,
   Calculate positions motor run time form WINDOW value,
   Run Position motor in appropriate direction
   Issue Time Trap to stop motor after calculated delay
```

CLKDR - Ephemeris Tracking Background Module

Description:

Periodically run position motor, to maintain approximate alignment with the SUN.

Algorithm:

```
If SUNF > DROPIN,
   Set Program Sequence 4
   Exit.
If Manual Mode,
   Stop positioning motor,
   Exit.
Otherwise,
   Reposition concentrator after delay.
```

6.3 POINT APPLICATION DEFINITIONS

APPLICATION PROGRAM RAM MEMORY MAP

	BUFFER ADDR.	ACTEXAR ADDR.	REL. ADDR.	ITEM	TRANSFERRED
FRBUFFO	44	33	+0	NEWBETA H.	To Background
	45	34	+1	NEWBETA L.	To Background
	46	35	+2	NEWSUN	No
	47	36	+3	OFFSET	No
	48	37	+4	unassigned	No
	49	38	+5	unassigned	No
	4A	39	+6	unassigned	No
	4B	3A	+7	unassigned	No
	4C	3B	+8	unassigned	No
	4D	3C	+9	unassigned	No
BKFUFFO	4E	33	+0	NEWBETA H.	From Foreground
	4F	34	+1	NEWBETA L.	From Foreground
	50	35	+2	NEWSUN	No
	51	36	+3	BETAF H.	No
	52	37	+4	BETAF L.	No
	53	38	+5	P1	No
	54	39	+6	P2	No
	55	3A	+7	P3	No
	56	3B	+8	SUNF	No
	57	3C	+9	AVGDPL	No

Fig. 6.3-1

APPLICATION PROGRAM PROM MEMORY MAP

SUBMODULE/ MODULE	MODULE, as required	ADDRESSES BEGIN	END		AVAIL. MEMORY
IPRG		0800	081C	001C	----
NPRG		0820	084C	002C	0004
ONEPAS		0860	08BB	005B	0014
FILTERS		08D0	099F	00CF	0015
RDSNR		09B0	0A4B	009B	0011
PAGEB	APPTBL	0B00	0C00	0100	00B5
TRACK		0C00	0CBC	00BC	0000
APPSUB		0E0B	0EE1	00D6	014F
Oper. Sys.	ASBPG3	0EEB	0FE0	00F5	000A
ENVTQUE	APPTBL	0FE0	1000	0020	0000
APPLICATION PROM			TOTAL	05B4	024C

Fig. 6.3-2

NOTE: PAGEB and EVNTQUE are only partially used.

RAM MEMORY DEFINITONS

```
NEWBETA MEASURED POSITION                    BUFFER ADDR. 44, 4E
H.      ACTEXAR OFFSET +0                   ACTEXAR ADDR. 33
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        HIGH BYTE OF ANGULAR DISPLACEMENT VALUE MEASURED IN
        RDSNR AND PASSED TO BACKGROUND.

NEWBETA MEASURED POSITION                    BUFFER ADDR. 45, 4F
L.      ACTEXAR OFFSET +1                   ACTEXAR ADDR. 34
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        LOW BYTE OF ANGULAR DISPLACEMENT VALUE MEASURED IN
        RDSNR AND PASSED TO BACKGROUND.

NEWSUN  MEASURED INSOLATION VALUE            BUFFER ADDR. 46, 50
        ACTEXAR OFFSET +2                   ACTEXAR ADDR. 35
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        INSOLATION LEVEL MEASURED IN RDSNR AND PASSED TO
        PASSED TO BACKGROUND.

OFFSET  POSITION OFFSET                      BUFFER ADDR. 47
        ACTEXAR OFFSET +3                   ACTEXAR ADDR. 36
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        POSITION OFFSET, MAY BE MODIFIED BY CENTRAL CONTROL.
        THE DEFAULT VALUE IS 0.

BETAF   FILTERED POSITION                    BUFFER ADDR. 51
H.      ACTEXAR OFFSET +3                   ACTEXAR ADDR. 36
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        HIGH BYTE OF FILTERED NEWBETA.

BETAF   FILTERED POSITION                    BUFFER ADDR. 52
L.      ACTEXAR OFFSET +4                   ACTEXAR ADDR. 37
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        LOW BYTE OF FILTERED NEWBETA.

P1      POSITION FILTER PARAMETER            BUFFER ADDR. 53
        ACTEXAR OFFSET +5                   ACTEXAR ADDR. 38
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        BETAF FILTER PARAMETER

P2      POSITION FILTER PARAMETER            BUFFER ADDR. 54
        ACTEXAR OFFSET +6                   ACTEXAR ADDR. 39
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        BETAF FILTER PARAMETER
```

RAM MEMORY DEFINITONS con'd

```
P3      POSITION FILTER PARAMETER                BUFFER ADDR. 55
        ACTEXAR OFFSET +7                       ACTEXAR ADDR. 3A
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        BETAF FILTER PARAMETER

SUNF    FILTERED INSOLATION VALUE                BUFFER ADDR. 56
        ACTEXAR OFFSET +8                       ACTEXAR ADDR. 3B
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        FILTERED NEWSUN

AVGDPL  MEAN AVERAGED DISPLACEMENT               BUFFER ADDR. 57
        ACTEXAR OFFSET +9                       ACTEXAR ADDR. 3C
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        MEAN AVERAGE OF THE POSITION CHANGE, OR
        ANGULAR POSITIONAL VELOCITY.  IT IS USED IN
        THE CALCULATION OF BETAF, WHERE IT IS USED
        AS PARAMETER CONTROLLING THE ADAPTIVE FILTER
        PORTION.

MSGBUFF0 UNIT #0 MESSAGE BUFFER                          ADDR. 42
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        (MTR0)(UNA )(--WINDOW--)(----UNA---)(-SUNF LSB-)
         WARM          COUNT                   B(-1) B(-2)
                                                         ADDR. 43
        [ B7 ][ B6 ][ B5 ][ B4 ][ B3 ][ B2 ][ B1 ][ B0 ]
        (----------------UNASSIGNED-------------------)
```

7.0 PROJECT EVALUATION

7.0.1 GENERAL

The Statement of Work, attachment B of the Contract indicates that the Proposal should be used as "the basis for technical approach," ...."to be applied under this contract,"....etc.

To establish a foundation for Project Evaluation the objectives delineated in the Abstract of the Proposal were reviewed. The salient items are:

a. Systems approach to design.

b. Management and Control of both Tracker and auxillary equipment.

c. Measurement of Suns Position and Insolation level.

d. Overcome inherent problems of multi-sensor null type trackers.

7.0.2 SYSTEMS APPROACH

Taking a Systems Approach at the hardware design stage develops a product which is adaptable to all foreseeable requirements. The resulting configuration is normally a basic unit which provides the fundamental features necessary for all applications and modules, which may be added or removed as necessary, to meet the varying special requirements that may be encountered. This modular approach is also desirable in the software design so that it too can be adapted to differing applications with the minimum of change. Such modularity of both hardware and software exists in the "POINT" tracking system.

7.0.3 MANAGEMENT AND CONTROL

In addition to providing the control functions for a maximum of sixteen POSITIONING UNITS, the CONTROLLER was designed to provide

a variety of unassigned inputs and outputs (see Specification page 4-2 for details) which may be used for Management and Control of auxillary equipment. The purpose to which they may be applied is flexible, and specific assignments may be accomplished by programming the microprocessor in the case of the LOCAL CONTROLLER based system, or keyboard entries in a central controller based system.

7.0.4 MEASUREMENT

Solar tracking is a dynamic activity and therefore for the Peak Detecting, Single Sensor concept to be established as superior to existing technology it is necessary that the measurements of the Sun's position and insolation level made by the Tracking Head be suitable in both timing and accuracy for convertion into commands to the device which keeps the concentrator focused on the Sun.

7.0.5 OVERCOMING INHERENT PROBLEMS OF EXISTING TRACKERS

To demonstrate that the Peak Detecting, Single Sensor technology is a superior approach to tracking, it is merely necessary to show that a tracker designed to that concept, will discriminate and ignore light sources which confuse multi-sensor null type trackers (shadow band). It is clear that the elimination of need to balance two analog circuits, and the suitability of digital technology for overall system control are significant advantages that accrue from this approach but in the final analysis the ability to track accurately and reliably must be the criteria by which a tracker is evaluated.

7.0.6 METHODOLOGY

The Tracking System was connected to the Demonstration Unit described in appendix II. This unit simulates the dynamic operating conditions for a line focus concentrator and provided the means of displaying the systems Tracking Capability, measuring it's dynamic Tracking Accuracy and also it's Field of View.

7.0.7 TRACKING ACCURACY

The "POINT" system accomplishes solar tracking in incremental steps. The Microprocessor is programmed to respond when the concentrators axis is displaced on either side of the Sun by a preselected angle. The CONTROLLER repositions the axis in the appropriate direction beyond the Sun where it dwells, until the next command. Together these angles, which may be altered if desired, form a tracking "Window".

Tracking accuracy is dependent on the resolution of the data received by the CONTROLLER from the TRACKING HEAD. The "design objective" for this parameter was ±0.1 degrees or better, i.e. a tracking "WINDOW" 0.2 degrees wide. It was decided that to obtain the "design objective" accuracy "the least significant bit" should represent a maximum of 0.05 degrees in axis displacement. In practice it has been established that the use of this value results in an accuracy of ±0.05 degrees.

If desired the tracking "WINDOW" can be arranged to either lead or lag the concentrator's axis permitting intentional defocussing. The "offset" can either be entered by an operator via the keyboard

in a Central Controller based system or automatically by an external sensor in both Central and Local Controller based systems. The value is programmable and may be changed in 0.02 degree increments to a maximum of 3.2 degrees leading or lagging.

The "offset" capability may be used for:

a. Precise tracking alignment

b. Emergency defocussing

c. Temperature control or a combination of these as desired.

7.1 RESULTS

7.1.1 TRACKING ACCURACY

A series of Accuracy observations were made over a period of one month. These observations took place under conditions varing from cloudless clear sky to cloudiness where the sun was observed or partially obscured 90% of the time. The observations were made at varing times throughout the day, the earliest at 8:30 A.M. and the latest at 4:30 P.M. when the sun is blocked by a local obstruction. The system performed reliable under all of these varing conditions acquiring the sun when activated and tracking it with an accuracy of better than $\pm$ 0.1 degrees when it was visable and remaining within 0.75 degrees (the programmed amount) when it was obscured.

7.1.2 FIELD OF VIEW

The Field of View was measured at noon on a cloudless day. The results obtained are shown on Fig. 7.1-1. It will be seen that

the measured value to the Polar Horizons is 131 degrees as compared to the calculated value based on optical fundimentals of 148 degrees. The total Field of View of the Tracking Head is shown diagramatically in Fig. 7.1-2.

7.2 CONCLUSIONS

It has been demonstrated that the Single Sensor, Peak Detecting method of tracking the sun is a superior approach for both Line and point focus concentrators. It offers the advantages of precise tracking, insulation level and tracking error measurement, dynamically modifiable tracking limits and offset adjustment.

In addition, the "Point" Control System provides unassigned inputs and outputs both analog and digital for control of ancillary equipment.

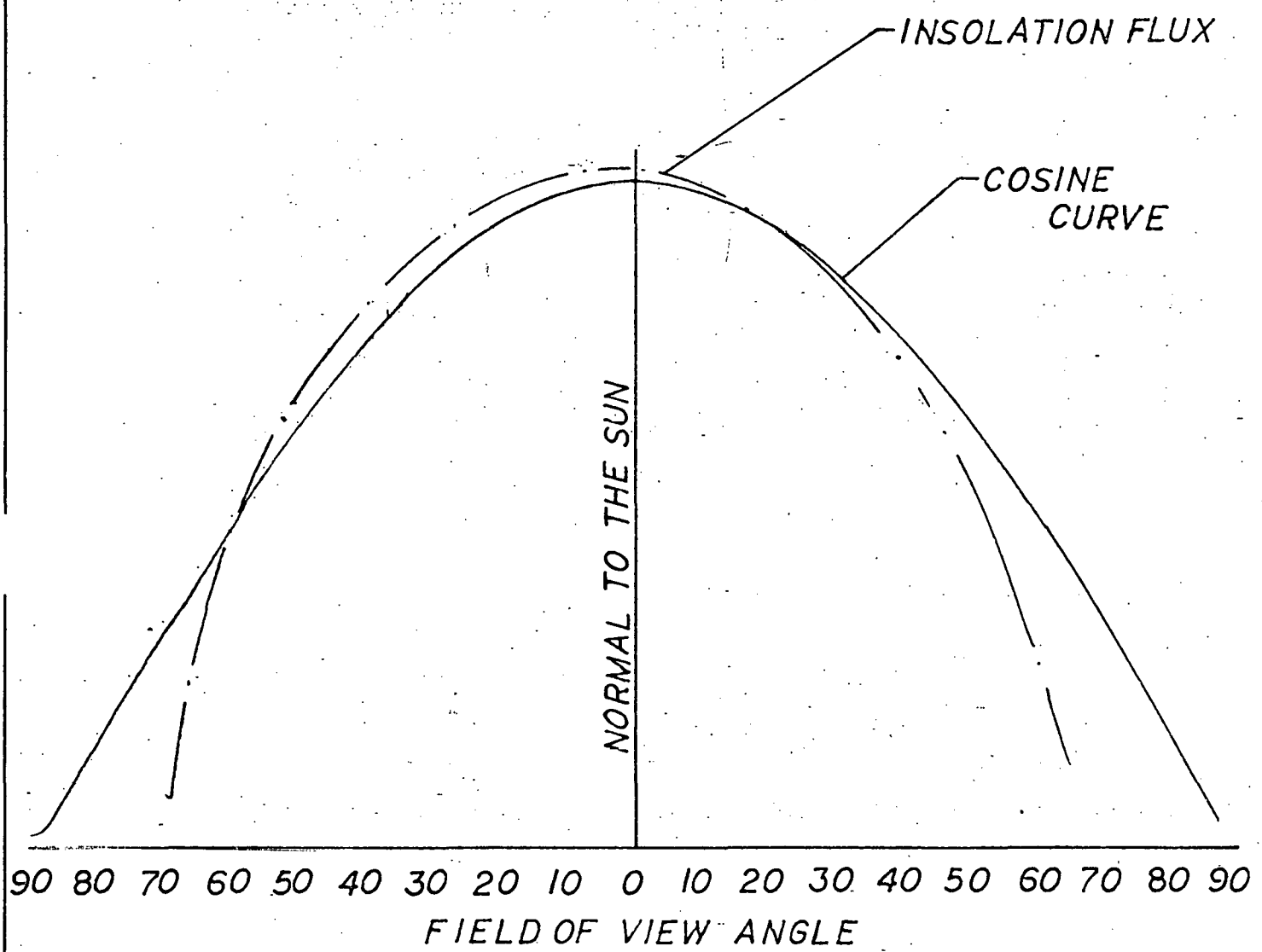


FIG 7.1-1 FIELD OF VIEW/INSOLATION FLUX

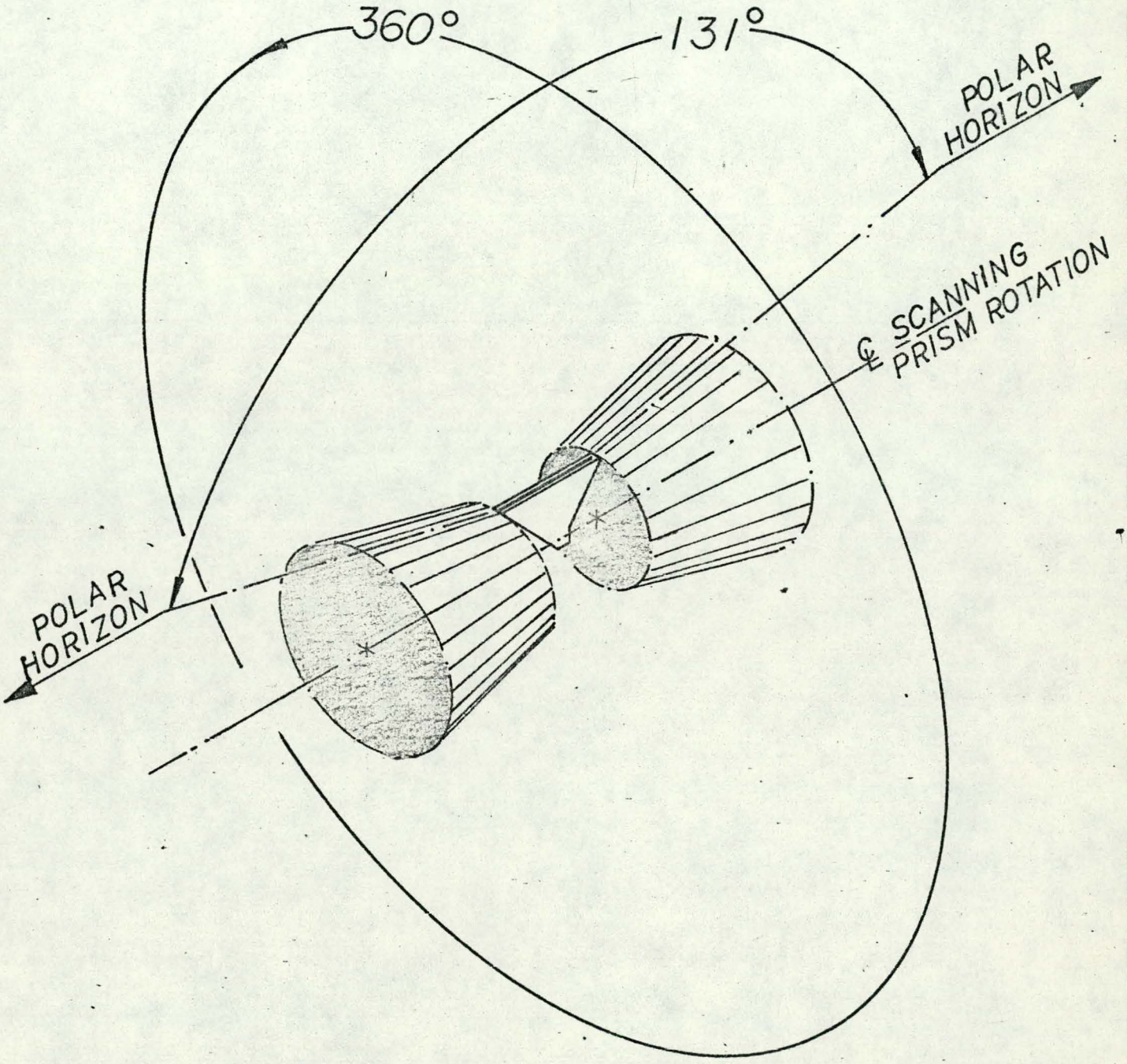


FIG 7.1-2 FIELD OF VIEW BOTH AXES

7.3 ENHANCEMENTS

7.3.1 LOCAL CONTROLLER - HARDWARE

It is possible to increase the tracking accuracy by utilizing the three existing data "bits" currently uncommitted. This would entail changing the asynchronous counters to synchronous counters, and running them directly from the crystal oscillator or phase lock loop. The necessary data latches currently exist. At present a simple peak detection circuit is used, it will be necessary to augment this with a signal differentiating circuit and a more precise magnetic index sensor. However, the design for the magnetic sensor exists, and it will be a standard feature in future models. In addition there are some associated software changes that would need to be implemented.

7.3.2 LOCAL CONTROLLER - SOFTWARE

Several enhancements are being made to the Operating System, to reduce it's overhead, and simplify the operation.

By adapting some of the Application Program modules, it will be possible to extend the operation to Point Focus systems.

7.3.3 TRACKING HEAD - HOUSING

The prototype tracking head was fabricated from commercially available materials which were cut and machined as necessary to make the individual components. It was discovered during testing that the cast acrylic tubing, used as protection from the environment, had imperfections (ripples) which on occasions distorted the Sun's rays. When this occurred the equipment continued to track to the ± 0.1 degree value, but would tend to hunt until the imperfection was no longer in the Sun's ray path.

On future models the acrylic housing would either be polished to remove the ripples, or alternatively the housing would be made from scientific glass tubing.

7.3.4 TRACKING HEAD - CONFIGURATION

The prototype tracking head was fabricated to investigate the Peak Detecting concept and for practical applications it will be necessary to repackage it in a form more convenient for installation, field maintenance and adjustment. See fig. 7.3-1.

It can be seen that in this arrangement the motor and signal connections are both located at one end and access to the prism, magnetic sensing head, motor and other components is possible by removing either the transparent or metal cover without disturbing the mounting.

7.3.5 FIELD OF VIEW

The present 131 degree field of view is adequate all point focus concentrators and line focus concentrator which are orientated on a North-South axis. For line focus concentrators with East-West orientation the field of view needs extending to a minimum of 170 degrees. This can be accomplished optically by modifying the prism.

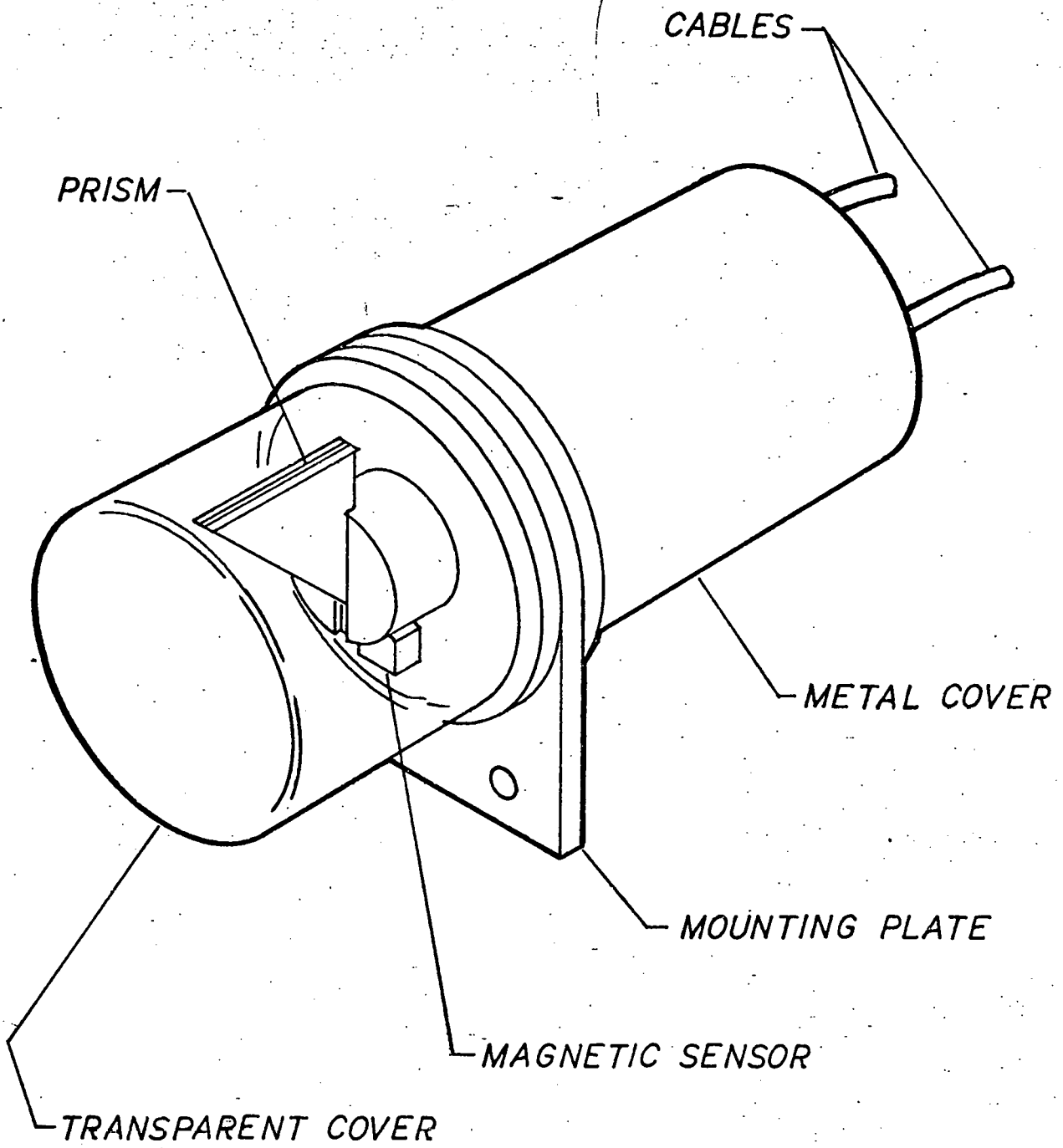


FIG 7.3-1 TRACKING HEAD

APPENDIX I

POINT APPLICATION PROGRAM MODULES

```
                 ************************************************************
                 ;SYSYMTBL

                 ;SYSTEM CONFIGURATION INFORMATION
0001             NUMUNITS  EQU     1 ;NUMBER OF TRACKERS PER CONTROLER
000A             LACTEXAR  EQU    10 ;LENGTH OF PROGRAM EXECUTION AREA
0010             NUMEVQ    EQU    16 ;MAX NUMBER OF EVENT QUEUE ENTRIES
0020             LPRGLST   EQU    32 ;LENGTH PROGRAM LIST TABLE(65 MAX
0020             LEVNTQUE  EQU    32 ;LENGTH EVENT QUEUE OPERATION TABLE(129 MAX
0040             LPRGSEQ   EQU    64 ;LENGTH PROGRAM MODULE SEQUENCE LIST TABLE
000C             NUMCOM    EQU X'0C' ;NUMBER OF COMMANDS=NUMBCOM IN SUBPG3
000F             RECYCPER  EQU X'0F' ;PROGRAM SEQUENCE RECYCLE PERIOD MASK

                 ;INTERNAL RAM ASSIGNMENTS
0003             REG03     EQU     X'03'        ;RAM ADDR REG3,RB0
001C             ACTACC    EQU     X'1C'        ;FOREGROUND ACCUMULATOR
001D             INACTACC  EQU     X'1D'        ;BACKGROUND ACCUMULATOR
001E             PRGSTAT   EQU     X'1E'        ;RAM ADDRESS
001F             OPSTAT    EQU     X'1F'        ;OPSYS STATUS BYTE
0020             PORT1     EQU     X'20'        ;RAM PORT#1 BYTE
0021             VECTOR    EQU     X'21'        ;RETURN VECTOR ADDRESS
0023             XMITCHR   EQU     X'23'        ;USART XMIT CHAR
0024             TIMVALL   EQU     X'24'        ;TIMER VALUE LOW BYTE
0025             TIMVALH   EQU     X'25'        ;TIMER VALUE HIGH BYTE
0026             SNDRECST  EQU     X'26'        ;SEND/REC PROG STATES
0027             SRCOMMSG  EQU     X'27'        ;SEND/REC ERROR MSG BITS
0028             SRPARM1   EQU     X'28'        ;SEND/REC PARAMETER #1
0029             SRPARM2   EQU     X'29'        ;SEND/REC PARAMETER #2
002A             PRGLSTUO  EQU     X'2A'        ;PROGRAM LIST UNIT#0
002B             PRGLSTAC  EQU     X'2B'        ;ACTIVE PROGRAM LIST
002C             DROPIN    EQU     X'2C'        ;ENTER TRACK MODE
002D             DROPOUT   EQU     X'2D'        ;ENTER CLOCK MODE
002E             WINDOW    EQU     X'2E'        ;MAX TRACK ERROR
002F             MSGBUFF   EQU     X'2F'        ;ACTIVE MESSAGE BUFFER
0031             ICPRG     EQU     X'31'        ;ADDR FRGND COM. AREA
0032             ACPRG     EQU     X'32'        ;ADDR BKGND COM. AREA
0033             ACTEXAR   EQU     X'33'        ;FOREGROUND UNIT#,PROG#
003D             ACTDIGFL  EQU     X'3D'        ;ACTIVE POSITON DIGITAL FILTER
0042             MSGBUFFO  EQU     X'42'        ;UNIT#0 MESSAGE BUFFER
0044             FRBUFFO   EQU     X'44'        ;FRGND BUFFER UNIT#0
004E             BKBUFFO   EQU     X'4E'        ;BKGND BUFFER UNIT#0
0060             EVEQUE    EQU     X'60'        ;START OF EVENT QUEUE

                 ;NO CURRENT EXTERNAL RAM

                 ;MESSAGE EQUATES
0001             INTMSG1   EQU     X'01'        ;SERIAL WORD SENT
0002             INTMSG2   EQU     X'02'        ;RECEIVE PARITY ERROR MSG
0003             INTMSG3   EQU     X'03'        ;RECEIVE OVERRUN ERROR MSG
0004             INTMSG4   EQU     X'04'        ;SEND MSG TERMINATED
0005             INTMSG5   EQU     X'05'        ;TIME INTERRUPT MSG
0006             INTMSG6   EQU     X'06'        ;UNASSIGNED
0007             INTMSG7   EQU     X'07'        ;STATUS INTERRUPT MSG
0001             TRPMSG1   EQU     X'01'        ;END TRAP MSG
0002             TRPMSG2   EQU     X'02'        ;TIME TRAP MSG
0003             TRPMSG3   EQU     X'03'        ;ABORT TRAP MSG
0004             TRPMSG4   EQU     X'04'        ;END SEQUENCE TRAP MSG
```

```
0007        TRPMSG7   EQU   X'07'    ;WAIT TRAP MSG
  01        SCHMSG1   EQU   X'01'    ;ACT SEQ DONE,DELETE FROM STK
0002        SCHMSG2   EQU   X'02'    ;ACT/INACT PROG SEQ DONE
0003        SCHMSG3   EQU   X'03'    ;ERROR,ACT PROG SEQ DONE
0004        SCHMSG4   EQU   X'04'    ;ERROR, BAD STACK
0001        MONMSG1   EQU   X'01'    ;RECEIVE PROTOCOL SYNTAX ERR
0002        MONMSG2   EQU   X'02'    ;RECEIVE COMMAND SYNTAX ERR
0003        MONMSG3   EQU   X'03'    ;RECEIVE OVERLAY ERROR
0004        MONMSG4   EQU   X'04'    ;RECEIVE PARITY ERROR MESSAGE
0005        MONMSG5   EQU   X'05'    ;RECEIVE OVERRUN ERROR MESSAGE
0006        MONMSG6   EQU   X'06'    ;TRANSMIT OVERRUN MSG
0007        MONMSG7   EQU   X'07'    ;PROGRAM STATE ERROR MSG
0008        MONMSG8   EQU   X'08'    ;QUEUE FULL ERROR MSG
0009        MONMSG9   EQU   X'09'    ;PROG SEQ RECYCLE SEMAPHORE
0010        MONRESMG  EQU   X'10'    ;SYSTEM RESET MESSAGE

            ;POINT TRAPS
0002        SUPTRP    EQU   X'02'    ;SUPERVISOR TRAP
                                     ;TO REENTER INTERRUPT STATE

            ;CONTROL CHARACTER EQUATES
0002        STX       EQU   X'02'    ;STX CHARACTER
0003        ETX       EQU   X'03'    ;ETX OR EOM CHAR.
0015        NAK       EQU   X'15'    ;NAK CHAR
0006        ACK       EQU   X'06'    ;ACK CHARACTER
0018        CAN       EQU   X'18'    ;CAN OR SO CHARACTER
0004        EOT       EQU   X'04'    ;EOT CHAR

            ;COMMAND EQUATES
0000        COMNOP    EQU   X'00'    ;NO OP
0001        SYSRES    EQU   X'01'    ;SYSTEM RESET
0002        RIRAM     EQU   X'02'    ;READ INTERNAL RAM
0003        WIRAM     EQU   X'03'    ;WRITE INTERNAL RAM
0004        RXRAM     EQU   X'04'    ;READ EXTERNAL RAM
0005        WXRAM     EQU   X'05'    ;WRITE EXTERNAL RAM
0006        RXIO      EQU   X'06'    ;READ EXTERNAL I/O
0007        WXIO      EQU   X'07'    ;WRITE EXTERNAL I/O
0008        RPRT      EQU   X'08'    ;READ PORT
0009        WPRT      EQU   X'09'    ;WRITE PORT
000A        WEVQ      EQU   X'0A'    ;EVENT QUEUE ENTRY
000B        RPSW      EQU   X'0B'    ;READ PSW

            ;EXTERNAL I/O HARDWARE ADDRESSES
0000        UNT       EQU   X'00'    ;RO FIELD UNIT ADDR
0000        ACT       EQU   X'00'    ;WO CYCLE FF=ACTIVE
0010        A1H       EQU   X'10'    ;RO ADDR.LATCH#1 HIGH
0010        PAS       EQU   X'10'    ;WO CYCLE FF=PASSIVE
0020        A1L       EQU   X'20'    ;RO ADDR.LATCH#1 LOW
0020        AL1       EQU   X'20'    ;WO LOAD ADDR LATCH#1
0030        A2H       EQU   X'30'    ;RO ADDR.LATCH#2 HIGH
0030        CPH       EQU   X'30'    ;WO B7-B4 COMP REG HIGH
0040        A2L       EQU   X'40'    ;RO ADDR.LATCH#2 LOW
  40        AL2       EQU   X'40'    ;WO LOAD ADDR LATCH#2
0050        STT       EQU   X'50'    ;RO  B3-B0 STATUS REG
0050        CPL       EQU   X'50'    ;WO COMP REG LOW
0060        URD       EQU   X'60'    ;RO USART RECEIVE REG
0061        UST       EQU   X'61'    ;RO USART STATUS REG
```

```
0062        URM     EQU     X'62'     ;RO USART MODE REGS
0063        URC     EQU     X'63'     ;RO USART COMMAND REG
0064        USD     EQU     X'64'     ;WO USART SEND REG
0065        USY     EQU     X'65'     ;WO USART SYNC REGS
0066        UWM     EQU     X'66'     ;WO USAR MODE REGS
0067        UCM     EQU     X'67'     ;WO USART COMMAND REG
0070        MTRO    EQU     X'70'     ;R/W MOTOR CONTROL REGISTER UNIT#0
            *MTR1   EQU     X'71'     ;R/W UNASSIGNED             UNIT#1
            *MTR2   EQU     X'72'     ;R/W UNASSIGNED             UNIT#2
            *MTR3   EQU     X'73'     ;R/W UNASSIGNED             UNIT#3
            *MTR4   EQU     X'74'     ;R/W UNASSIGNED             UNIT#4
            *MTR5   EQU     X'75'     ;R/W UNASSIGNED             UNIT#5
            *MTR6   EQU     X'76'     ;R/W UNASSIGNED             UNIT#6
            *MTR7   EQU     X'77'     ;R/W UNASSIGNED             UNIT#7
            *MTR8   EQU     X'78'     ;R/W UNASSIGNED             UNIT#8
            *MTR9   EQU     X'79'     ;R/W UNASSIGNED             UNIT#9
            *MTRA   EQU     X'7A'     ;R/W UNASSIGNED             UNIT#10
            *MTRB   EQU     X'7B'     ;R/W UNASSIGNED             UNIT#11
            *MTRC   EQU     X'7C'     ;R/W UNASSIGNED             UNIT#12
            *MTRD   EQU     X'7D'     ;R/W UNASSIGNED             UNIT#13
            *MTRE   EQU     X'7E'     ;R/W UNASSIGNED             UNIT#14
            *MTRF   EQU     X'7F'     ;R/W UNASSIGNED             UNIT#15
0080        SHC     EQU     X'80'     ;WO ZERO SAMPLE HOLD CAPACITOR
0080        SWSO    EQU     X'80'     ;RO SWITCH STATUS REGISTER  UNIT#0
            *SWS1   EQU     X'81'     ;RO UNASSIGNED              UNIT#1
            *SWS2   EQU     X'82'     ;RO UNASSIGNED              UNIT#2
            *SWS3   EQU     X'83'     ;RO UNASSIGNED              UNIT#3
            *SWS4   EQU     X'84'     ;RO UNASSIGNED              UNIT#4
            *SWS5   EQU     X'85'     ;RO UNASSIGNED              UNIT#5
            *SWS6   EQU     X'86'     ;RO UNASSIGNED              UNIT#6
            *SWS7   EQU     X'87'     ;RO UNASSIGNED              UNIT#7
            *SWS8   EQU     X'88'     ;RO UNASSIGNED              UNIT#8
            *SWS9   EQU     X'89'     ;RO UNASSIGNED              UNIT#9
            *SWSA   EQU     X'8A'     ;RO UNASSIGNED              UNIT#A
            *SWSB   EQU     X'8B'     ;RO UNASSIGNED              UNIT#B
            *SWSC   EQU     X'8C'     ;RO UNASSIGNED              UNIT#C
            *SWSD   EQU     X'8D'     ;RO UNASSIGNED              UNIT#D
            *SWSE   EQU     X'8E'     ;RO UNASSIGNED              UNIT#E
            *SWSF   EQU     X'8F'     ;RO UNASSIGNED              UNIT#F
            ***     EQU     X'90'     ;RO UNASSIGNED
0090        ADC     EQU     X'90'     ;WO START ANALOG TO DIGITAL CONV.
00A0        ANSO    EQU     X'A0'     ;R/W ANALOG SELECT REGISTER UNIT#0
            *ANS1   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#1
            *ANS2   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#2
            *ANS3   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#3
            *ANS4   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#4
            *ANS5   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#5
            *ANS6   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#6
            *ANS7   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#7
            *ANS8   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#8
            *ANS9   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#9
            *ANSA   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#10
            *ANSB   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#11
            *ANSC   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#12
            *ANSD   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#13
            *ANSE   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#14
            *ANSF   EQU     X'A1'     ;R/W UNASSIGNED             UNIT#16
```

```
00B0          URES      EQU     X'B0'        ;WO  RESET USART
              ***************************************************************
00B0                    END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
K	0006	ACPRG	0032	ACT	0000	ACTACC	001C
ACTDIGFL	003D	ACTEXAR	0033	ADC	0090	AL1	0020
AL2	0040	ANSO	00A0	BKBUFFO	004E	CAN	0018
COMNOP	0000	CPH	0030	CPL	0050	DROPIN	002C
DROPOUT	002D	EOT	0004	ETX	0003	EVEQUE	0060
FRBUFFO	0044	ICPRG	0031	INACTACC	001D	INTMSG1	0001
INTMSG2	0002	INTMSG3	0003	INTMSG4	0004	INTMSG5	0005
INTMSG6	0006	INTMSG7	0007	LACTEXAR	000A	LEVNTQUE	0020
LPRGLST	0020	LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002
MONMSG3	0003	MONMSG4	0004	MONMSG5	0005	MONMSG6	0006
MONMSG7	0007	MONMSG8	0008	MONMSG9	0009	MONRESMG	0010
MSGBUFF	002F	MSGBUFFO	0042	MTRO	0070	NAK	0015
NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001	OPSTAT	001F
PAS	0010	PORT1	0020	PRGLSTAC	002B	PRGLSTUO	002A
PRGSTAT	001E	RECYCPER	000F	REG03	0003	RIRAM	0002
RPRT	0008	RPSW	000B	RXIO	0006	RXRAM	0004
SCHMSG1	0001	SCHMSG2	0002	SCHMSG3	0003	SCHMSG4	0004
SHC	0080	SNDRECST	0026	SRCOMMSG	0027	SRPARM1	0028
SRPARM2	0029	STT	0050	STX	0002	SUPTRP	0002
SWSO	0080	SYSRES	0001	TIMVALH	0025	TIMVALL	0024
TRPMSG1	0001	TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004
TRPMSG7	0007	UCM	0067	UNT	0000	URC	0063
URD	0060	URES	00B0	URM	0062	USD	0064
UST	0061	USY	0065	UWM	0066	VECTOR	0021
WEVQ	000A	WINDOW	002E	WIRAM	0003	WPRT	0009
0	0007	WXRAM	0005	XMITCHR	0023		

NO ERRORS

```
                ;APPADD.G
                ;FILE USED FOR SUBROUTINE LINKAGE

                ;THIS ADDRESS FILE MUST BE UPDATED WITH
                ;CHANGES IN APPSUB.S
                ;FUTUREDATA DOES NOT
                ;ALLOW GLOBAL SYMBOLS IN THE 8048 ASSEMBLER

                ;ADDRESSES CONTAINED IN APPSUB.S

0E0B            TRUNADP   EQU     X'E0B'
0E52            CLRWRD    EQU     X'E52'
0E58            SSUB      EQU     X'E58'
0E73            FMINUS    EQU     X'E73'
0E82            FPLUS     EQU     X'E82'
0E8D            FASR      EQU     X'E8D'
0E98            FTRUN     EQU     X'E98'
0EAF            FRNDOFF   EQU     X'EAF'
0EBA            DSADD     EQU     X'EBA'
0ECC            DSSUB     EQU     X'ECC'
0000                      END
```

CLRWRD	0E52	DSADD	0EBA	DSSUB	0ECC	FASR	0E8D
INUS	0E73	FPLUS	0E82	FRNDOFF	0EAF	FTRUN	0E98
SSUB	0E58	TRUNADP	0E0B				

NO ERRORS

```
                ;APPTBL.S

                          PRNT    GEN
0000                      ORG     X'B00'

                ;PAGE3 IN APPLICATION PROGRAM MEMORY

                ********************
0B00            PRGTABL   EQU     *
                ;PRGTABLE: NIBBLES 0-2 CONTAIN THE STARTING ADDRESS OF
                ;THE APPLICATION PROGRAM MODULE. B2-B0 OF NIBBLE #3
                ;IS THE NUMBER OF BYTES TO BE TRANSFER TO THE OPPOSITE
                ;BUFFER (FRBUFFO OR BKBUFFO) AT THE TERMINATION OF THE
                ;FOREGROUND OR BACKGROUND SEQUENCE. THE VALUE ONLY HAS
                ;SIGNIFIGANCE IF IT IS THE LAST MODULE OF EITHER THE
                ;FOREGROUND OR BACKGROUND. B3 IS USED TO INDICATED AN
                ;UNINTERRUPIBLE FOREGROUND MODULE. IT IS UNASSIGNED
                ;FOR BACKGROUND MODULES.

0B00            FRPROG    EQU     *
                          ;16 ENTRIES MAX, INCLUDING DUMMY
                          ;                          NAME      ADDR. DATA REC'D
0B00 0000                 DC      B(X'0000')      0. DUMMY     000   0
0B02 0008                 DC      B(X'0800')      1. IPRG      800   0
0B04 2008                 DC      B(X'0820')      2. NPRG      820   0
0B06 6008                 DC      B(X'0860')      3. ONEPAS    860   0
(   B B009                DC      B(X'09B0')      4. RDSNR     9B0   0
(__A                      ORG     X'B20'
0B20            BKPROG    EQU     *
                          ;16 ENTRIES MAX, INCLUDING DUMMY
                          ;                          NAME      ADDR. DATA REC'D
0B20 0000                 DC      B(X'0000')      0. DUMMY     000   0
0B22 500A                 DC      B(X'0A50')      1. CLKDR     A50   0
0B24 D038                 DC      B(X'38D0')      2. FILTERS   8D0   3
                          ;                          NEWBETA   +0    2
                          ;                          NEWSUN    +2    1
0B26 000C                 DC      B(X'0C00')      3  TRACK     B00   0
0B28                      ORG     X'B40'
                ********************
                ;PRGLST IS A INDEX TABLE CONTAINING THE START ADDRESSES
                ;IN PAGE3 OF FRGLST/BCKLST PROGRAM SEQUENCES
                          ;32 ENTRIES CURRENT MAXIMUM. 64 ENTRIES MAXIMUM
                          ;POSSIBLE NUMBER OF ENTRIES
0B40            PRGLST    EQU     *
0B40 60                   DC      PRGSEQ0.AND.X'00FF'   INITIAL PROG SEQUENCE
0B41 63                   DC      PRGSEQ1.AND.X'00FF'   SENSOR INITIALIZATION
0B42 66                   DC      PRGSEQ2.AND.X'00FF'   CON'TRATOR CLOCK DRIVEN
0B43 6A                   DC      PRGSEQ3.AND.X'00FF'   STOW CONCENTRATOR
0B44 A0                   DC      PRGSEQ4.AND.X'00FF'   TRACK CONCENTRATOR
0B45                      ORG     X'B60'
                ********************
                ;PROGRAM SEQUENCES, FOREGROUND SEQUENCE FIRST, THEN
                ;BACKGROUND SEQUENCE, EACH USES A ZERO EOF TERMINATOR
                ;EACH PROGRAM SEQUENCE HAS AN ADDRESS ENTRY IN PRGLST
                          ; EACH FOREGROUND OR BACKGROUND SECTION MAY HAVE
                          ; A MAXIMUM OF 16 ENTRIES WHICH EQUAL 8 BYTES
0B60            PRGSEQ    EQU     *
```

```
0B60            PRGSEQ0   EQU     *          ;IPRC INITIAL PROG SEQ
  60 01                   DC      X'01'      ;FORGND PROG  1
  61 00                   DC      0          ;EOF FORGND PROG SEQ
0B62 00                   DC      0          ;EOF BCKGND PROG SEQ

0B63            PRGSEQ1   EQU     *          ;ONEPAS,RDSNR TRACK INIT
0B63 43                   DC      X'43'      ;FORGND PROG  3,4
0B64 00                   DC      0          ;EOF FORGND
0B65 00                   DC      0          ;EOF BCKGND

0B66            PRGSEQ2   EQU     *          ;RDSNR,BFILTER,CLOCK DRIVE
0B66 04                   DC      X'04'      ;FORGND PROG  4
0B67 00                   DC      0          ;EOF FORGND
0B68 12                   DC      X'12'      ;BCKGND PROG  2,1
0B69 00                   DC      0          ;EOF BCKGND

0B6A            PRGSEQ3   EQU     *          ;NPRG STOW COLLECTOR
0B6A 02                   DC      X'02'      ;FORGND PROG  2
0B6B 00                   DC      0          ;EOF FORGND
0B6C 00                   DC      0          ;EOF BCKGND

0B6D                      ORG     X'BA0'

0BA0            PRGSEQ4   EQU     *          ;RDSNR,BFILTER,TRACK
0BA0 04                   DC      X'04'      ;FORGND PROG  4
0BA1 00                   DC      0          ;EOF FORGND
  A2 32                   DC      X'32'      ;BCKGND PROG  2,3
  A3 00                   DC      0          ;EOF BCKGND

                ;INSERT ADDITIONAL TABLES HERE

                ;END OF PAGE3  X'BFE'  DON'T USE THIS LOCATION
```

```
0BA4                    ORG     X'FE0'

                ;EVNTQUE CONTAINS OPERATION, FUNCTIONS, AND ADDRESSES
                ;WHICH ARE ADDRESSED BY EVENT QUEUE OPERATIONS.
                ;64 ENTRIES MAXIMUM

0FDD            EVNTQUE   EQU     *-3
                      ;QUE1=AND, MTR(0) B3,B2,B1=0, MTR(0)^(X'F1')
                      ;AND IMEDIATE DATA WITH I/O
0FE0 32                   DC      X'32',X'F1',MTR0
0FE1 F1
0FE2 70
                      ;QUE2=OR, MTR(0) D3=1, MTR(0)v(X'08')
                      ;OR IMEDIATE DATA WITH I/O
0FE3 31                   DC      X'31',X'08',MTR0
0FE4 08
0FE5 70
                      ;QUE3=OR, MSGBUFF0 D7=1, MSGBUFF0v(X'80')
                      ;OR IMEDIATE DATA WITH INTERNAL RAM
0FE6 B1                   DC      X'B1',X'80',MSGBUFF0
0FE7 80
0FE8 42


                ********************
                ;SYSYMTBL
                          PRNT    GEN
                ********************
0FE9                      END
```

''H	0010	A1L	0020	A2H	0030	A2L	0040
K	0006	ACPRG	0032	ACT	0000	ACTACC	001C
ACTDIGFL	003D	ACTEXAR	0033	ADC	0090	AL1	0020
AL2	0040	ANS0	00A0	BKBUFFO	004E	BKPROG	0B20
CAN	0018	COMNOP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	EOT	0004	ETX	0003
EVEQUE	0060	EVNTQUE	0FDD	FRBUFFO	0044	FRPROG	0B00
ICPRG	0031	INACTACC	001D	INTMSG1	0001	INTMSG2	0002
INTMSG3	0003	INTMSG4	0004	INTMSG5	0005	INTMSG6	0006
INTMSG7	0007	LACTEXAR	000A	LEVNTQUE	0020	LPRGLST	0020
LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002	MONMSG3	0003
MONMSG4	0004	MONMSG5	0005	MONMSG6	0006	MONMSG7	0007
MONMSG8	0008	MONMSG9	0009	MONRESMG	0010	MSGBUFF	002F
MSGBUFFO	0042	MTRO	0070	NAK	0015	NUMCOM	000C
NUMEVQ	0010	NUMUNITS	0001	OPSTAT	001F	PAS	0010
PORT1	0020	PRGLST	0B40	PRGLSTAC	002B	PRGLSTUO	002A
PRGSEQ	0B60	PRGSEQO	0B60	PRGSEQ1	0B63	PRGSEQ2	0B66
PRGSEQ3	0B6A	PRGSEQ4	0BA0	PRGSTAT	001E	PRGTABL	0B00
RECYCPER	000F	REGO3	0003	RIRAM	0002	RPRT	0008
RPSW	000B	RXIO	0006	RXRAM	0004	SCHMSG1	0001
SCHMSG2	0002	SCHMSG3	0003	SCHMSG4	0004	SHC	0080
SNDRECST	0026	SRCOMMSG	0027	SRPARM1	0028	SRPARM2	0029
STT	0050	STX	0002	SUPTRP	0002	SWS0	0080
SYSRES	0001	TIMVALH	0025	TIMVALL	0024	TRPMSG1	0001
TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004	TRPMSG7	0007
UCM	0067	UNT	0000	URC	0063	URD	0060
ES	00B0	URM	0062	USD	0064	UST	0061
USY	0065	UWM	0066	VECTOR	0021	WEVQ	000A
WINDOW	002E	WIRAM	0003	WPRT	0009	WXIO	0007
WXRAM	0005	XMITCHR	0023				

NO ERRORS

```
                ;IPRG                                          11/24/80

                          PRNT    GEN
 0000                     ORG     X'800'


                ;FUNCTION:          IPRG - Initial Program Cycle
                ;INPUTS:            PRGLSTAC B7 (inital program cycle)
                ;OUTPUTS:           DROPIN
                ;                   DROPOUT
                ;                   WINDOW
                ;DESTROYS:          ACC, R0
                ;DESCRIPTION:       IPRG - If initial program cycle loads default
                ;                   values into Dropin and Dropout. Otherwise,
                ;                   waits until central control changes the program
                ;                   sequence.


 0800           IPRG      EQU     *
 0800                     LD      A,(PRGLSTAC)              ;get initial cycle
+0800                     SELMAC  MOV,A,(PRGLSTAC)
+0800                     OPRM    MOV,A,PRGLSTAC        ;R0 OR R1 LOST
+0800 B82B                MOV     R0,#PRGLSTAC              ;R0 LOST
+0802 F0                  MOV A,@R0                     ;LEVEL1 OPCODE R,(MEM)
 0803 F215                JB7     ENDIPRG                   ;not initial cycle
 0805                     LD      (DROPIN),#IDRPIN
+0805                     SELMAC  MOV,(DROPIN),#IDRPIN
+0805                     OPMIMD  MOV,DROPIN,#IDRPIN        ;R0 LOST
+0805 B82C                MOV     R0,#DROPIN                ;R0 LOST
+0807 B050                MOV @R0,#IDRPIN
 0809                     LD      (DROPOUT),#IDRPOUT
+0809                     SELMAC  MOV,(DROPOUT),#IDRPOUT
+0809                     OPMIMD  MOV,DROPOUT,#IDRPOUT       ;R0 LOST
+0809 B82D                MOV     R0,#DROPOUT               ;R0 LOST
+080B B040                MOV @R0,#IDRPOUT
 080D                     LD      (DROPOUT),#IDRPOUT
+080D                     SELMAC  MOV,(DROPOUT),#IDRPOUT
+080D                     OPMIMD  MOV,DROPOUT,#IDRPOUT       ;R0 LOST
+080D B82D                MOV     R0,#DROPOUT               ;R0 LOST
+080F B040                MOV @R0,#IDRPOUT
 0811                     LD      (WINDOW),#IWINDOW
+0811                     SELMAC  MOV,(WINDOW),#IWINDOW
+0811                     OPMIMD  MOV,WINDOW,#IWINDOW       ;R0 LOST
+0811 B82E                MOV     R0,#WINDOW                ;R0 LOST
+0813 B003                MOV @R0,#IWINDOW
 0815 2301      ENDIPRG   MOV     A,#ENDT
 0817                     TRAP
+0817 35                  DIS     TCNTI          ;START CRITICAL REGION
+0818 8A20                ORL     P2,#X'20'
+081A 05                  EN      I
+081B 00                  NOP                    ;END CRITICAL REGION
                ***************
                ;SYSYMTBL
                          PRNT    GEN
                ***************
                ***************
                ;APSYMTBL
```

```
                    PRNT    GEN
            ***************
            ***************
            ;APPADD.G
                    PRNT    GEN
            ***************
081C                END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
]RTT	0011	ACK	0006	ACPRG	0032	ACT	0000
ACTACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETAF	0036	BKBUFF0	004E	CAN	0018
CLRWRD	0E52	COMNOP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	DSADD	0EBA	DSSUB	0ECC
ENDIPRG	0815	ENDSEQ	0019	ENDT	0001	EOT	0004
ETX	0003	EVEQUE	0060	FASR	0E8D	FMINUS	0E73
FORWRK	0040	FPLUS	0E82	FRBUFF0	0044	FRNDOFF	0EAF
FTRUN	0E98	ICPRG	0031	IDRPIN	0050	IDRPOUT	0040
INACTACC	001D	INTMSG1	0001	INTMSG2	0002	INTMSG3	0003
INTMSG4	0004	INTMSG5	0005	INTMSG6	0006	INTMSG7	0007
IPRG	0800	ISTT	0008	IWINDOW	0003	LACTEXAR	000A
LEVNTQUE	0020	LPRGLST	0020	LPRGSEQ	0040	MONMSG1	0001
MONMSG2	0002	MONMSG3	0003	MONMSG4	0004	MONMSG5	0005
MONMSG6	0006	MONMSG7	0007	MONMSG8	0008	MONMSG9	0009
MONRESMG	0010	MSGBUFF	002F	MSGBUFF0	0042	MTR0	0070
NAK	0015	NEWBETA	0033	NEWSUN	0035	NUMCOM	000C
NUMEVQ	0010	NUMUNITS	0001	OFFSET	0036	OPSTAT	001F
P1	0038	P2	0039	P3	003A	PAS	0010
PORT1	0020	PRGLSTAC	002B	PRGLSTU0	002A	PRGSTAT	001E
QUE1	0004	QUE2	000A	QUE3	000E	RECYCPER	000F
REG03	0003	RIRAM	0002	RPRT	0008	RPSW	000B
RUNDLY	0008	RUNLIM	0070	RXIO	0006	RXRAM	0004
SCHMSG1	0001	SCHMSG2	0002	SCHMSG3	0003	SCHMSG4	0004
:	0080	SNDRECST	0026	SRCOMMSG	0027	SRPARM1	0028
L... ARM2	0029	SSUB	0E58	STT	0050	STX	0002
SUNF	003B	SUPTRP	0002	SWS0	0080	SYSRES	0001
TIMVALH	0025	TIMVALL	0024	TRPMSG1	0001	TRPMSG2	0002
TRPMSG3	0003	TRPMSG4	0004	TRPMSG7	0007	TRUNADP	0E0B
TTRP	0009	UCM	0067	UNT	0000	URC	0063
URD	0060	URES	00B0	URM	0062	USD	0064
UST	0061	USY	0065	UWM	0066	VECTOR	0021
WARM	0004	WATADC	0043	WATCOMP	0013	WATIND	0023
WEVQ	000A	WINDOW	002E	WIRAM	0003	WPRT	0009
WXIO	0007	WXRAM	0005	XMITCHR	0023		

NO ERRORS

```
                ;NPRG                                          11-24-80

                          PRNT    GEN
 0000                     ORG     X'820'


                ;FUNCTION:        NPRG - Night program
                ;INPUTS:          None
                ;OUTPUTS:         Positioning Motor, Rotation
                ;CALLS:           SWSTAT
                ;DESTROYS:        ACC,R0,R1
                ;DESCRIPTION:     NPRG - Stows the mirror only


 0820           NPRG      EQU     *
 0820 B820                MOV     R0,#PORT1
 0822 2308                MOV     A,#ISTT                  ; initial status
 0824 30                  XCHD    A,@R0                    ; store low nibble
 0825 F0                  MOV     A,@R0                    ; get Port 1
 0826 39                  OUTL    P1,A                     ; output status
 0827                     LDX     A,(SWS0)                 ;read switch status
+0827                     SELMAC  MOVX,A,(SWS0)
+0827                     OPRM    MOVX,A,SWS0        ;R0 OR R1 LOST
+0827 B880                MOV     R0,#SWS0              ;R0 LOST
+0829 80                  MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 082A 1234                JB0     NPOUT                    ; manual mode,wait
   2C 323B                JB1     ENDNP                    ; stow limit
   2E                     LD      A,X(MTR0)                ; get motor status
+082E                     SELMAC  MOV,A,X(MTR0)
+082E                     OPRM    MOVX,A,MTR0
+082E B870                MOV     R0,#MTR0              ;R0 LOST
+0830 80                  MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 0831 4306                ORL     A,#X'06'                 ; motor ON, CCW
 0833 90                  MOVX    @R0,A                    ; do it
 0834 2301      NPOUT     MOV     A,#ENDT                  ; set end trap code
 0836                     TRAP
+0836 35                  DIS     TCNTI          ;START CRITICAL REGION
+0837 8A20                ORL     P2,#X'20'
+0839 05                  EN      I
+083A 00                  NOP                    ;END CRITICAL REGION
 083B           ENDNP     EQU     *                        ; Turn Motor OFF
 083B                     LDX     A,(MTR0)
+083B                     SELMAC  MOVX,A,(MTR0)
+083B                     OPRM    MOVX,A,MTR0        ;R0 OR R1 LOST
+083B B870                MOV     R0,#MTR0              ;R0 LOST
+083D 80                  MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 083E 53F9                ANL     A,#X'F9'                 ; motor OFF
 0840 90                  MOVX    @R0,A                    ; output it
 0841                     LD      (PRGLSTU0),#X'80'        ; prog seq#0
+0841                     SELMAC  MOV,(PRGLSTU0),#X'80'
+0841                     OPMIMD  MOV,PRGLSTU0,#X'80'        ;R0 LOST
+ 41 B82A                MOV     R0,#PRGLSTU0                ;R0 LOST
+  43 B080                MOV @R0,#X'80'
                                                          ; not initial cycle
 0845 2301                MOV     A,#ENDT                  ; set end trap code
 0847                     TRAP
+0847 35                  DIS     TCNTI          ;START CRITICAL REGION
```

```
+0848 8A20                ORL     P2,#X'20'
+   A 05                  EN      I
+084B 00                  NOP                     ;END CRITICAL REGION
                 **************
                 ;SYSYMTBL
                          PRNT    GEN
                 **************
                 **************
                 ;APSYMTBL
                          PRNT    GEN
                 **************
                 **************
                 ;APPADD.G
                          PRNT    GEN
                 **************
 084C                     END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
ABRTT	0011	ACK	0006	ACPRG	0032	ACT	0000
ACTACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETAF	0036	BKBUFFO	004E	CAN	0018
CLRWRD	0E52	COMNOP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	DSADD	0EBA	DSSUB	0ECC
ENDNP	083B	ENDSEQ	0019	ENDT	0001	EOT	0004
ETX	0003	EVEQUE	0060	FASR	0E8D	FMINUS	0E73
FORWRK	0040	FPLUS	0E82	FRBUFFO	0044	FRNDOFF	0EAF
FTRUN	0E98	ICPRG	0031	IDRPIN	0050	IDRPOUT	0040
INACTACC	001D	INTMSG1	0001	INTMSG2	0002	INTMSG3	0003
INTMSG4	0004	INTMSG5	0005	INTMSG6	0006	INTMSG7	0007
ISTT	0008	IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020
LPRGLST	0020	LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002
MONMSG3	0003	MONMSG4	0004	MONMSG5	0005	MONMSG6	0006
MONMSG7	0007	MONMSG8	0008	MONMSG9	0009	MONRESMG	0010
MSGBUFF	002F	MSGBUFFO	0042	MTRO	0070	NAK	0015
NEWBETA	0033	NEWSUN	0035	NPOUT	0834	NPRG	0820
NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001	OFFSET	0036
OPSTAT	001F	P1	0038	P2	0039	P3	003A
PAS	0010	PORT1	0020	PRGLSTAC	002B	PRGLSTUO	002A
PRGSTAT	001E	QUE1	0004	QUE2	000A	QUE3	000E
RECYCPER	000F	REG03	0003	RIRAM	0002	RPRT	0008
RPSW	000B	RUNDLY	0008	RUNLIM	0070	RXIO	0006
RXRAM	0004	SCHMSG1	0001	SCHMSG2	0002	SCHMSG3	0003
SCHMSG4	0004	SHC	0080	SNDRECST	0026	SRCOMMSG	0027
SRPARM1	0028	SRPARM2	0029	SSUB	0E58	STT	0050
STX	0002	SUNF	003B	SUPTRP	0002	SWS0	0080
SYSRES	0001	TIMVALH	0025	TIMVALL	0024	TRPMSG1	0001
TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004	TRPMSG7	0007
TRUNADP	0E0B	TTRP	0009	UCM	0067	UNT	0000
URC	0063	URD	0060	URES	00B0	URM	0062
USD	0064	UST	0061	USY	0065	UWM	0066
VECTOR	0021	WARM	0004	WATADC	0043	WATCOMP	0013
WATIND	0023	WEVQ	000A	WINDOW	002E	WIRAM	0003
WPRT	0009	WXIO	0007	WXRAM	0005	XMITCHR	0023

NO ERRORS

```
                ;CLKDR                                          10-8-80

                        PRNT    GEN
0000                    ORG     X'A50'


                ;FUNCTION:          CLKDR - Clock Drive Module
                ;INPUTS:            MTRO (D1=POSIT.MTR ON,D2=ROTAT.,D3=INIT.STATE)
                ;                   DROPIN
                ;                   SUNF
                ;                   EVEQUE
                ;OUTPUTS:           MTRO (D1=POSIT.MTR ON,D2=ROTAT.,D3=INIT.STATE)
                ;                   PRGLSTUO, if SUNF > DROPIN value
                ;                   EVEQUE
                ;CALLS:             NONE
                ;DESTROYS:          Accumulator, and RO
                ;DESCRIPTION:       Initially the positioning motor is off, the
                ;                   program waits the OFF period, sets the motor
                ;                   On and waits the ON period. The sequence then
                ;                   repeats.
                ;                   If SUNF is equal to or greater than DROPIN the
                ;                   program sequence is changed to the track mode
                ;                   of operation. The operation is only performed
                ;                   in STATE #1, and the pending time trap is
                ;                   deleted from EVEQUE.
                ;                   MTRO contains the entry code
                ;                   Entry States:                   Responses:
                ;                   0. D1=0 Positioning Motor       Initial State
                ;                      D2=0 Rotation                1. Set D2=1
                ;                      D3=X                         2. D3=0 No change
                ;                                                   3. Wait Time Trap
                ;                                                      D3>1
                ;                      Illegal Conditions:
                ;                      1. Current time trap         1. Clr time trap
                ;                   1. D1=0 Positioning Motor       Motor Off wait
                ;                      D2=1 Rotation                1. None
                ;                      D3=0 State xsition flag
                ;                      Illegal Conditions:
                ;                      1. No Current time trap      1. Execute State 0
                ;                   2. D1=0 Positioning Motor       Motor Off xsition
                ;                      D2=1 Rotation                1. D1=1 Motor ON
                ;                      D3=1 State xsition flag      2. D2=N Rotation
                ;                                                   3. D3=0 No change
                ;                                                   3. Wait Time Trap
                ;                                                      D1>0,D2>0
                ;                      Illegal Conditions:
                ;                      1. Current time trap         1. Clr time trap
                ;                   3. D1=1 Positioning Motor       Motor ON wait
                ;                      D2=X Rotation                1. None
                ;                      D3=0 State xsition flag
                ;                      Illegal Conditions:
                ;                      1. No Current time trap      1. Execute State 0


0A50            CLKDR   EQU     *
0A50 B960               MOV     R1,#EVEQUE
0A52                    LDX     A,(SWSO)        ; read switch status
```

```
+0A52                    SELMAC  MOVX,A,(SWS0)
+0A52                    OPRM    MOVX,A,SWS0        ;R0 OR R1 LOST
+0A52 B880               MOV     R0,#SWS0              ;R0 LOST
+0A54 80                 MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 0A55 B870               MOV     R0,#MTR0
 0A57 12AD               JB0     CLKDR4             ; MANUAL MODE,Scan MTR only
 0A59 80                 MOVX    A,@R0              ; get motor status
 0A5A 32A3               JB1     CLKDR3             ; state 3 NO ACTION
 0A5C 5273               JB2     CLKDR1             ; state 1 or 2
            ;Entry State 0  Initial State  D1,D2=0
                         ;Error reentry point from states 1 and 3
 0A5E B100      CLKDRERR MOV     @R1,#0             ; clr any event queue entry
 0A60 80                 MOVX    A,@R0              ; get motor status
 0A61 4304               ORL     A,#X'04'           ; set D2=1 Rotation
 0A63 53F5               ANL     A,#X'F5'           ; set D3=0 Xsition
                                                    ; set D1=0 Mtr OFF
 0A65 90                 MOVX    @R0,A              ; store new mtr status
 0A66 BB58               MOV     R3,#X'58'          ; 180 sec's to mtr ON
 0A68 BA0A               MOV     R2,#QUE2           ; event queue parm
 0A6A                    LD      A,#TTRP            ; MTR0 D3=1 Xsition
+0A6A                    SELMAC  MOV,A,#TTRP
+0A6A                    OPRIMD  MOV,A,#TTRP
+
+0A6A 2309               MOV A,#TTRP
 0A6C                    TRAP
+0A6C 35                 DIS     TCNTI            ;START CRITICAL REGION
+   D 8A20               ORL     P2,#X'20'
+   F 05                 EN      I
+0A70 00                 NOP                      ;END CRITICAL REGION
 0A71 44A6               JMP     CLKDREX            ; leave
            ;Entry State 1 or 2
 0A73       CLKDR1       EQU     *                  ; not initial cycle
 0A73 728F               JB3     CLKDR2             ; state 2
            ;Entry State 1, Normally no ACTION
 0A75 F1                 MOV     A,@R1              ; check for event queue entry
 0A76 C65E               JZ      CLKDRERR           ; Error no event queue entry
                                                    ; reinitiate
 0A78       CLKDR1A      LD      A,(DROPIN)
+0A78                    SELMAC  MOV,A,(DROPIN)
+0A78                    OPRM    MOV,A,DROPIN       ;R0 OR R1 LOST
+0A78 B82C               MOV     R0,#DROPIN            ;R0 LOST
+0A7A F0                 MOV A,@R0                     ;LEVEL1 OPCODE R,(MEM)
 0A7B 37                 CPL     A                  ; 1's CPL DROPIN
 0A7C                    ADDM    A,(SUNF)           ; SUNF-DROPIN-1
+0A7C                    SELMAC  ADD,A,(SUNF)
+0A7C                    OPRM    ADD,A,SUNF         ;R0 OR R1 LOST
+0A7C B83B               MOV     R0,#SUNF              ;R0 LOST
+0A7E 60                 ADD A,@R0                     ;LEVEL1 OPCODE R,(MEM)
 0A7F 17                 INC     A                  ; SUNF-DROPIN
 0A80 F2A6               JB7     CLKDREX            ; SUNF(DROPIN keep clock mode
                                                    ; state verification as #3
                         ;SUNF)DROPIN-1, Set program sequence to track mode
   82                    LDX     (MTR0),#X'01'      ; scan motor only
+0A82                    SELMAC  MOVX,(MTR0),#X'01'
+0A82                    OPMIMD  MOVX,MTR0,#X'01'      ;R0 LOST
+0A82 B870               MOV     R0,#MTR0              ;R0 LOST
+0A84 2301               MOV     A,#X'01'                 ;ACC LOST
```

```
+0A86 90                MOVX @R0,A
   87 B100              MOV     @R1,#0          ; clr any event queue entry
 0A89                   LD      (PRGLSTU0),#X'84'
+0A89                   SELMAC  MOV,(PRGLSTU0),#X'84'
+0A89                   OPMIMD  MOV,PRGLSTU0,#X'84'       ;R0 LOST
+0A89 B82A              MOV     R0,#PRGLSTU0               ;R0 LOST
+0A8B B084              MOV @R0,#X'84'
                                                  ; program sequence 4
                                                  ; not initial cycle
                                                  ; TRACK Drive
 0A8D 44A6              JMP     CLKDREX          ; leave
            ;Entry State 2
 0A8F       CLKDR2      EQU     *
 0A8F B100              MOV     @R1,#0           ; clr any event queue entry
 0A91 4302              ORL     A,#X'02'         ; D1=1 turn mtr on
 0A93 53F3              ANL     A,#X'F3'         ; D2=0 rotation
                                                  ; D3=0 no xsition
 0A95 90                MOVX    @R0,A            ; store new mtr status
 0A96 BB7D              MOV     R3,#X'7D'        ; 1 sec wait to mtr OFF
 0A98 BA04              MOV     R2,#QUE1         ; turn mtr OFF event
 0A9A                   LD      A,#TTRP          ; MTR0 D3,D2,D1=0
+0A9A                   SELMAC  MOV,A,#TTRP
+0A9A                   OPRIMD  MOV,A,#TTRP
+
+0A9A 2309              MOV A,#TTRP
 0A9C                   TRAP
+   C 35                DIS     TCNTI          ;START CRITICAL REGION
+0A9D 8A20              ORL     P2,#X'20'
+0A9F 05                EN      I
+0AA0 00                NOP                    ;END CRITICAL REGION
 0AA1 44A6              JMP     CLKDREX          ; leave
            ;Entry State 3, Normally no ACTION
 0AA3 F1    CLKDR3      MOV     A,@R1            ; check for event queue entry
 0AA4 C65E              JZ      CLKDRERR         ; Error no event queue entry
                                                  ; reinitiate
            ;Exit
 0AA6 2301  CLKDREX     MOV     A,#ENDT          ; SET END TRAP CODE
 0AA8                   TRAP
+0AA8 35                DIS     TCNTI          ;START CRITICAL REGION
+0AA9 8A20              ORL     P2,#X'20'
+0AAB 05                EN      I
+0AAC 00                NOP                    ;END CRITICAL REGION
            ;Manual override
 0AAD B001  CLKDR4      MOV     @R0,#X'01'       ; scan motor MTR0 only
 0AAF 4478              JMP     CLKDR1A          ; check Prog Seq


            **************
            ;SYSYMTBL
                        PRNT    GEN
            **************
            **************
            ;APSYMTBL
                        PRNT    GEN
            **************
            **************
            ;APPADD.G
```

```
                    PRNT    GEN
            **************
..31                END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
ORTT	0011	ACK	0006	ACPRG	0032	ACT	0000
ACTACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETAF	0036	BKBUFFO	004E	CAN	0018
CLKDR	0A50	CLKDR1	0A73	CLKDR1A	0A78	CLKDR2	0A8F
CLKDR3	0AA3	CLKDR4	0AAD	CLKDRERR	0A5E	CLKDREX	0AA6
CLRWRD	0E52	COMNOP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	DSADD	0EBA	DSSUB	0ECC
ENDSEQ	0019	ENDT	0001	EOT	0004	ETX	0003
EVEQUE	0060	FASR	0E8D	FMINUS	0E73	FORWRK	0040
FPLUS	0E82	FRBUFFO	0044	FRNDOFF	0EAF	FTRUN	0E98
ICPRG	0031	IDRPIN	0050	IDRPOUT	0040	INACTACC	001D
INTMSG1	0001	INTMSG2	0002	INTMSG3	0003	INTMSG4	0004
INTMSG5	0005	INTMSG6	0006	INTMSG7	0007	ISTT	0008
IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020	LPRGLST	0020
LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002	MONMSG3	0003
MONMSG4	0004	MONMSG5	0005	MONMSG6	0006	MONMSG7	0007
MONMSG8	0008	MONMSG9	0009	MONRESMG	0010	MSGBUFF	002F
MSGBUFFO	0042	MTRO	0070	NAK	0015	NEWBETA	0033
NEWSUN	0035	NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001
OFFSET	0036	OPSTAT	001F	P1	0038	P2	0039
P3	003A	PAS	0010	PORT1	0020	PRGLSTAC	002B
PRGLSTUO	002A	PRGSTAT	001E	QUE1	0004	QUE2	000A
QUE3	000E	RECYCPER	000F	REG03	0003	RIRAM	0002
RPRT	0008	RPSW	000B	RUNDLY	0008	RUNLIM	0070
IO	0006	RXRAM	0004	SCHMSG1	0001	SCHMSG2	0002
SCHMSG3	0003	SCHMSG4	0004	SHC	0080	SNDRECST	0026
SRCOMMSG	0027	SRPARM1	0028	SRPARM2	0029	SSUB	0E58
STT	0050	STX	0002	SUNF	003B	SUPTRP	0002
SWS0	0080	SYSRES	0001	TIMVALH	0025	TIMVALL	0024
TRPMSG1	0001	TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004
TRPMSG7	0007	TRUNADP	0E0B	TTRP	0009	UCM	0067
UNT	0000	URC	0063	URD	0060	URES	00B0
URM	0062	USD	0064	UST	0061	USY	0065
UWM	0066	VECTOR	0021	WARM	0004	WATADC	0043
WATCOMP	0013	WATIND	0023	WEVQ	000A	WINDOW	002E
WIRAM	0003	WPRT	0009	WXIO	0007	WXRAM	0005
XMITCHR	0023						

NO ERRORS

```
                ;ONEPAS

                          PRNT    GEN
 0000                     ORG     X'860'


                ;
                ;Forsround Module #1 - Both TRACK and CLOCK
                ;FUNCTION:          ONEPAS - One pass; First time set-up
                ;INPUTS:            PRGLSTAC(B7),MSGBUFFO(B7)
                ;OUTPUTS:           AL1=Index Position
                ;                   MTRO,BO=1 Scan Motor On
                ;                   P13 Index Disable,(PORT1,B3)=0
                ;                   ANSO=Photo Diode
                ;CALLS:             None
                ;DESTROYS:          ACC,RO,R2,R3
                ;DESCRIPTION:       Performs all startup functions ; modes, enables


 0860           ONEPAS     EQU     *
 0860                      LD      A,(PRGLSTAC)        ;set initial cyc bit
+0860                      SELMAC  MOV,A,(PRGLSTAC)
+0860                      OPRM    MOV,A,PRGLSTAC        ;RO OR R1 LOST
+0860 B82B                 MOV     RO,#PRGLSTAC              ;RO LOST
+0862 FO                   MOV A,@RO                    ;LEVEL1 OPCODE R,(MEM)
 0863 F7                   RLC     A                   ;B7)CARRY
    )4 E672                JNC     ONEPAS2             ;initial cycle
 0866                      LD      A,(MSGBUFFO)        ;read mss flags
+0866                      SELMAC  MOV,A,(MSGBUFFO)
+0866                      OPRM    MOV,A,MSGBUFFO        ;RO OR R1 LOST
+0866 B842                 MOV     RO,#MSGBUFFO              ;RO LOST
+0868 FO                   MOV A,@RO                    ;LEVEL1 OPCODE R,(MEM)
 0869 F290                 JB7     ONEPAS3             ;scan mtr warm,set parm's
 086B           ONEPAS1    EQU     *
 086B 2319                 MOV     A,#ENDSEQ           ;END SEQUENCE Trap
 086D                      TRAP
+086D 35                   DIS     TCNTI          ;START CRITICAL REGION
+086E 8A20                 ORL     P2,#X'20'
+0870 05                   EN      I
+0871 00                   NOP                    ;END CRITICAL REGION
 0872           ONEPAS2    EQU     *
 0872                      ANLM    (MSGBUFFO),#X'7F' ;set B7=0 Mtr not warm
+0872                      SELMAC  ANL,(MSGBUFFO),#X'7F'
+0872                      OPMIMD  ANL,MSGBUFFO,#X'7F'       ;RO LOST
+0872 B842                 MOV     RO,#MSGBUFFO               ;RO LOST
+0874 20                   XCH     A,@RO
+0875 537F                 ANL A,#X'7F'
+0877 20                   XCH     A,@RO
 0878 2308                 MOV     A,#ISTT
 087A                      LD      (PORT1),A           ;Set INIT Status
+087A                      SELMAC  MOV,(PORT1),A
+   A                     OPMR    MOV,PORT1,A       ;RO OR R1 LOST
+   'A B820                MOV     RO,#PORT1                 ;RO LOST
+087C AO                   MOV @RO,A                    ;LEVEL1 MOV (MEM),R
 087D 39                   OUTL    P1,A
 087E                      LDX     (MTRO),#X'01'       ;turn on scan motor
+087E                      SELMAC  MOVX,(MTRO),#X'01'
```

```
+087E                    OPMIMD  MOVX,MTR0,#X'01'      ;R0 LOST
+  7E B870               MOV     R0,#MTR0              ;R0 LOST
+0880 2301               MOV     A,#X'01'              ;ACC LOST
+0882 90                 MOVX @R0,A
 0883 BA0E               MOV     R2,#QUE3              ;set EVENT QUE#3
 0885 BB04               MOV     R3,#WARM              ;Warm Up
 0887 2309               MOV     A,#TTRP               ;set time trap message
 0889                    TRAP
+0889 35                 DIS     TCNTI          ;START CRITICAL REGION
+088A 8A20               ORL     P2,#X'20'
+088C 05                 EN      I
+088D 00                 NOP                    ;END CRITICAL REGION
 088E 046B               JMP     ONEPAS1               ;End Sequence Trap
                         ;Scan Motor Warm initialize Parameters
 0890          ONEPAS3   EQU     *
 0890                    ANLM    (PORT1),#X'F7'        ;index disable,B3=0
+0890                    SELMAC  ANL,(PORT1),#X'F7'
+0890                    OPMIMD  ANL,PORT1,#X'F7'      ;R0 LOST
+0890 B820               MOV     R0,#PORT1             ;R0 LOST
+0892 20                 XCH     A,@R0
+0893 53F7               ANL A,#X'F7'
+0895 20                 XCH     A,@R0
 0896 99F7               ANL     P1,#X'F7'
 0898                    LDX     (PAS),A               ;set passive mode,No parms
+0898                    SELMAC  MOVX,(PAS),A
+0898                    OPMR    MOVX,PAS,A       ;R0 OR R1 LOST
+   8 B810               MOV     R0,#PAS               ;R0 LOST
+089A 90                 MOVX @R0,A                    ;LEVEL1 MOV (MEM),R
 089B 2323               MOV     A,#WATIND             ;wait trap for index
 089D                    TRAP
+089D 35                 DIS     TCNTI          ;START CRITICAL REGION
+089E 8A20               ORL     P2,#X'20'
+08A0 05                 EN      I
+08A1 00                 NOP                    ;END CRITICAL REGION
                         ;LOAD INDEX VALUE+180 deg's in COMPARISON REG
 08A2                    LDX     A,(A1H)               ;set index High
+08A2                    SELMAC  MOVX,A,(A1H)
+08A2                    OPRM    MOVX,A,A1H       ;R0 OR R1 LOST
+08A2 B810               MOV     R0,#A1H               ;R0 LOST
+08A4 80                 MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 08A5 D308               XRL     A,#X'08'              ;add 180 Deg's
 08A7                    LDX     (CPH),A               ;load comp reg High
+08A7                    SELMAC  MOVX,(CPH),A
+08A7                    OPMR    MOVX,CPH,A       ;R0 OR R1 LOST
+08A7 B830               MOV     R0,#CPH               ;R0 LOST
+08A9 90                 MOVX @R0,A                    ;LEVEL1 MOV (MEM),R
 08AA                    LDX     A,(A1L)               ;set index Low
+08AA                    SELMAC  MOVX,A,(A1L)
+08AA                    OPRM    MOVX,A,A1L       ;R0 OR R1 LOST
+08AA B820               MOV     R0,#A1L               ;R0 LOST
+08AC 80                 MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
    D                    LDX     (CPL),A               ;load index Low
+   D                    SELMAC  MOVX,(CPL),A
+08AD                    OPMR    MOVX,CPL,A       ;R0 OR R1 LOST
+08AD B850               MOV     R0,#CPL               ;R0 LOST
+08AF 90                 MOVX @R0,A                    ;LEVEL1 MOV (MEM),R
 08B0                    LD      (PRGLSTU0),#X'02' ;select program sequence #2
```

```
+08B0                    SELMAC  MOV,(PRGLSTU0),#X'02'
+  B0                    OPMIMD  MOV,PRGLSTU0,#X'02'        ;R0 LOST
+08B0 B82A               MOV     R0,#PRGLSTU0                ;R0 LOST
+08B2 B002               MOV @R0,#X'02'
 08B4 2301               MOV     A,#ENDT            ;END Trap
 08B6                    TRAP
+08B6 35                 DIS     TCNTI          ;START CRITICAL REGION
+08B7 8A20               ORL     P2,#X'20'
+08B9 05                 EN      I
+08BA 00                 NOP                    ;END CRITICAL REGION
              **************
              ;SYSYMTBL
                         PRNT    GEN
              **************
              **************
              ;APSYMTBL
                         PRNT    GEN
              **************
              **************
              ;APPADD.G
                         PRNT    GEN
              **************
 08BB                    END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
[illegible]RTT	0011	ACK	0006	ACPRG	0032	ACT	0000
[illegible]ACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETAF	0036	BKBUFFO	004E	CAN	0018
CLRWRD	0E52	COMNOP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	DSADD	0EBA	DSSUB	0ECC
ENDSEQ	0019	ENDT	0001	EOT	0004	ETX	0003
EVEQUE	0060	FASR	0E8D	FMINUS	0E73	FORWRK	0040
FPLUS	0E82	FRBUFFO	0044	FRNDOFF	0EAF	FTRUN	0E98
ICPRG	0031	IDRPIN	0050	IDRPOUT	0040	INACTACC	001D
INTMSG1	0001	INTMSG2	0002	INTMSG3	0003	INTMSG4	0004
INTMSG5	0005	INTMSG6	0006	INTMSG7	0007	ISTT	0008
IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020	LPRGLST	0020
LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002	MONMSG3	0003
MONMSG4	0004	MONMSG5	0005	MONMSG6	0006	MONMSG7	0007
MONMSG8	0008	MONMSG9	0009	MONRESMG	0010	MSGBUFF	002F
MSGBUFFO	0042	MTRO	0070	NAK	0015	NEWBETA	0033
NEWSUN	0035	NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001
OFFSET	0036	ONEPAS	0860	ONEPAS1	086B	ONEPAS2	0872
ONEPAS3	0890	OPSTAT	001F	P1	0038	P2	0039
P3	003A	PAS	0010	PORT1	0020	PRGLSTAC	002B
PRGLSTUO	002A	PRGSTAT	001E	QUE1	0004	QUE2	000A
QUE3	000E	RECYCPER	000F	REG03	0003	RIRAM	0002
RPRT	0008	RPSW	000B	RUNDLY	0008	RUNLIM	0070
RXIO	0006	RXRAM	0004	SCHMSG1	0001	SCHMSG2	0002
[illegible]MSG3	0003	SCHMSG4	0004	SHC	0080	SNDRECST	0026
[illegible]OMMSG	0027	SRPARM1	0028	SRPARM2	0029	SSUB	0E58
STT	0050	STX	0002	SUNF	003B	SUPTRP	0002
SWS0	0080	SYSRES	0001	TIMVALH	0025	TIMVALL	0024
TRPMSG1	0001	TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004
TRPMSG7	0007	TRUNADP	0E0B	TTRP	0009	UCM	0067
UNT	0000	URC	0063	URD	0060	URES	00B0
URM	0062	USD	0064	UST	0061	USY	0065
UWM	0066	VECTOR	0021	WARM	0004	WATADC	0043
WATCOMP	0013	WATIND	0023	WEVQ	000A	WINDOW	002E
WIRAM	0003	WPRT	0009	WXIO	0007	WXRAM	0005
XMITCHR	0023						

NO ERRORS

```
                ;RDSNR

                        PRNT    GEN
 0000                   ORG     X'9B0'


                ;FUNCTION:          RDSNR - Read Sensors
                ;INPUTS:            AL1 - Address #1, index position
                ;                   APFLAG - (in ACTEXAR+0)
                ;                   INDEX DISABLE - 0
                ;                   CYCLE FF - Passive
                ;OUTPUTS:           APFLAG - (B7 first pass flag)
                ;                   NEWBETA - new error angle (ACTEXAR+2, +3)
                ;                   NEWSUN - new sun level (ACTEXAR+1)
                ;                   CYCLE FF - Passive
                ;                   INDEX DISABLE - 0
                ;CALLS:             BETA,SUNLVL
                ;DESTROYS:          All
                ;DESCRIPTION:       RDSNR - Checks scan motor warm, then reads
                ;                           pointing error angle (NEWBETA), and
                ;                           insolation level (NEWSUN).  Sets
                ;                           1st pass flag to 'continuing' before
                ;                           exiting.


 09B0           RDSNR   EQU     *
   0                    LDX     A,(MTR0)                ;insure scan mtr ON
+09B0                   SELMAC  MOVX,A,(MTR0)
+09B0                   OPRM    MOVX,A,MTR0         ;R0 OR R1 LOST
+09B0 B870              MOV     R0,#MTR0                ;R0 LOST
+09B2 80                MOVX A,@R0                      ;LEVEL1 OPCODE R,(MEM)
 09B3 4301              ORL     A,#X'01'
 09B5 90                MOVX    @R0,A                   ;store it
                        ;1. WAIT FOR COMPARISON ENTER ACTIVE STATE
                        ;2. ACQUIRE POSITION OF PEAK SIGNAL AND PEAK ANALOG
                        ;   LEVEL
                        ;3. ON NEXT COMPARISON, REENTER PASSIVE STATE
                ;DESCRIPTION:       BETA sets up for PEAK DETECTION. Sets Modes,
                ;                   ANALOG SELECT = #2 (Photo Diode). BETA Zero's
                ;                   SAMPLE/HOLD Capacitor, and sets passive. Loads
                ;                   COMPARISON REGISTER and starts SYNC COMPARISON
                ;                   TRAP, to start PEAK CYCLE (Active/Passive
                ;                   Enabled).  Runs Peak Detection cycle (Ends with
                ;                   2nd Comparison Trap).  Read Results, Compare
                ;                   and Calculate NEWBETA, and calculate New ANTI
                ;                   INDEX (Disable Active/Passive Cycle).


 09B6           BETA    EQU     *
 09B6                   LDX     (PAS),A                 ;set Passive
+09B6                   SELMAC  MOVX,(PAS),A
+   36                  OPMR    MOVX,PAS,A          ;R0 OR R1 LOST
+   36 B810             MOV     R0,#PAS                 ;R0 LOST
+09B8 90                MOVX @R0,A                      ;LEVEL1 MOV (MEM),R
 09B9                   LDX     (SHC),A                 ;Zero S/H capicator
+09B9                   SELMAC  MOVX,(SHC),A
+09B9                   OPMR    MOVX,SHC,A          ;R0 OR R1 LOST
```

```
+09B9 B880                MOV     R0,#SHC                 ;R0 LOST
+   BB 90                 MOVX @R0,A                      ;LEVEL1 MOV (MEM),R
 09BC                     LDX     (ANS0),#X'02'           ;Photo Diode Input
+09BC                     SELMAC  MOVX,(ANS0),#X'02'
+09BC                     OPMIMD  MOVX,ANS0,#X'02'        ;R0 LOST
+09BC B8A0                MOV     R0,#ANS0                ;R0 LOST
+09BE 2302                MOV     A,#X'02'                ;ACC LOST
+09C0 90                  MOVX @R0,A
 09C1 B820                MOV     R0,#PORT1               ;load P1 and PORT1
 09C3 2303                MOV     A,#X'03'                ;B0 Cycle Enable=1
                                                          ;B1 S/H mode=1 Peak
                                                          ;B2 AL2 LOAD MODE=0 Peak
                                                          ;B3 Index Disable=0
 09C5 A0                  MOV     @R0,A                   ;load PORT1
 09C6 39                  OUTL    P1,A                    ;output it
                          ;wait for ACTIVE state
 09C7 2313                MOV     A,#WATCOMP              ;Comparison trap
 09C9                     TRAP
+09C9 35                  DIS     TCNTI           ;START CRITICAL REGION
+09CA 8A20                ORL     P2,#X'20'
+09CC 05                  EN      I
+09CD 00                  NOP                     ;END CRITICAL REGION
                          ;wait for PASSIVE state
 09CE 2313                MOV     A,#WATCOMP              ;Comparison trap for
 09D0                     TRAP                            ;Peak Detection
+09D0 35                  DIS     TCNTI           ;START CRITICAL REGION
+   D1 8A20               ORL     P2,#X'20'
+09D3 05                  EN      I
+09D4 00                  NOP                     ;END CRITICAL REGION
 09D5 B820                MOV     R0,#PORT1               ;Disable Cycle
 09D7 230A                MOV     A,#X'0A'                ;B0 Cycle Enable=0
                                                          ;B1 S/H mode=1 Peak
                                                          ;B2 AL2 LOAD MODE=0 Peak
                                                          ;B3 Index Disable=1
 09D9 A0                  MOV     @R0,A                   ;load PORT1
 09DA 39                  OUTL    P1,A
                ;LOAD INDEX VALUE+180 deg's in COMPARISON REG
 09DB                     LDX     A,(A1H)                 ;get index High
+09DB                     SELMAC  MOVX,A,(A1H)
+09DB                     OPRM    MOVX,A,A1H      ;R0 OR R1 LOST
+09DB B810                MOV     R0,#A1H                 ;R0 LOST
+09DD 80                  MOVX A,@R0                      ;LEVEL1 OPCODE R,(MEM)
 09DE D308                XRL     A,#X'08'                ;add 180 Deg's
 09E0                     LDX     (CPH),A                 ;load comp reg High
+09E0                     SELMAC  MOVX,(CPH),A
+09E0                     OPMR    MOVX,CPH,A      ;R0 OR R1 LOST
+09E0 B830                MOV     R0,#CPH                 ;R0 LOST
+09E2 90                  MOVX @R0,A                      ;LEVEL1 MOV (MEM),R
 09E3                     LDX     A,(A1L)                 ;get index Low
+09E3                     SELMAC  MOVX,A,(A1L)
+09E3                     OPRM    MOVX,A,A1L      ;R0 OR R1 LOST
+   E3 B820               MOV     R0,#A1L                 ;R0 LOST
+   E5 80                 MOVX A,@R0                      ;LEVEL1 OPCODE R,(MEM)
 09E6                     LDX     (CPL),A                 ;load index Low
+09E6                     SELMAC  MOVX,(CPL),A
+09E6                     OPMR    MOVX,CPL,A      ;R0 OR R1 LOST
+09E6 B850                MOV     R0,#CPL                 ;R0 LOST
```

```
+09E8 90                  MOVX @R0,A                    ;LEVEL1 MOV (MEM),R
                ;CONVERT TO RELATIVE POSITION
 ..E9 5428                CALL    SUBMATH               ;calculate rel. position
                                                        ;AL2 minus AL1
                                                        ;ACC=B15-B8
                                                        ;R3=B8-B0
 09EB                     LD      (NEWBETA),A           ;store B15-B8
+09EB                     SELMAC  MOV,(NEWBETA),A
+09EB                     OPMR    MOV,NEWBETA,A        ;R0 OR R1 LOST
+09EB B833                MOV     R0,#NEWBETA                ;R0 LOST
+09ED A0                  MOV @R0,A                    ;LEVEL1 MOV (MEM),R
 09EE 18                  INC     R0
 09EF FB                  MOV     A,R3                  ;get B8-B0
 09F0 A0                  MOV     @R0,A                 ;store NEWBETA+1
 09F1 C8                  DEC     R0                    ;restore R0
                          ;ADD POSITION OFFSET
                                  ;Convert 1/4(1/10)Deg's to 1/16(1/10)Deg's
 09F2                     LD      A,(OFFSET)
+09F2                     SELMAC  MOV,A,(OFFSET)
+09F2                     OPRM    MOV,A,OFFSET        ;R0 OR R1 LOST
+09F2 B836                MOV     R0,#OFFSET              ;R0 LOST
+09F4 F0                  MOV A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 09F5 97                  CLR     C
 09F6 F7                  RLC     A                     ;times 2,save sign
 09F7 AB                  MOV     R3,A                  ;save it
 09F8 27                  CLR     A
 ( 9 E6FC                 JNC     *+3
    B 37                  CPL     A
 09FC AA                  MOV     R2,A
 09FD B802                MOV     R0,#2                 ;addr. R2
 09FF B902                MOV     R1,#2                 ;addr. R2
 0A01 D482                CALL    FPLUS                 ;times2=times4
 0A03 B833                MOV     R0,#NEWBETA
 0A05 D482                CALL    FPLUS                 ;NEWBETA+4(OFFSET)
                ;CONVERT PEAK LEVEL TO DIGITAL
 0A07           SUNLVL    EQU     *
 0A07                     LDX     (ADC),A               ;start A/D conversion
+0A07                     SELMAC  MOVX,(ADC),A
+0A07                     OPMR    MOVX,ADC,A        ;R0 OR R1 LOST
+0A07 B890                MOV     R0,#ADC                ;R0 LOST
+0A09 90                  MOVX @R0,A                    ;LEVEL1 MOV (MEM),R
 0A0A 2343                MOV     A,#WATADC
 0A0C                     TRAP                          ;wait for END ADC
+0A0C 35                  DIS     TCNTI           ;START CRITICAL REGION
+0A0D 8A20                ORL     P2,#X'20'
+0A0F 05                  EN      I
+0A10 00                  NOP                     ;END CRITICAL REGION
 0A11 5428                CALL    SUBMATH               ;calculate ADC value
                                                        ;AL2 minus AL1
                                                        ;ACC=B15-B8
                                                        ;R3=B8-B0
                          ;DELETE Nibbles 0 and 3,Restructure to 1Byte
 (  3 B803                MOV     R0,#X'03'             ;addr. R3
 0A15 30                  XCHD    A,@R0                 ;R3=ValueNibblesReversed
 0A16 53F0                ANL     A,#X'F0'              ;delete LoNibble
 0A18 C61C                JZ      SUNLVL1
 0A1A BBFF                MOV     R3,#X'FF'             ;Overflow set value=FF
```

```
 0A1C          SUNLVL1   EQU      *
   1C FB                MOV      A,R3
   1D 47                SWAP     A                    ;ProperNibbleAlignment
 0A1E                   LD       (NEWSUN),A           ;store it
+0A1E                   SELMAC   MOV,(NEWSUN),A
+0A1E                   OPMR     MOV,NEWSUN,A       ;R0 OR R1 LOST
+0A1E B835              MOV      R0,#NEWSUN             ;R0 LOST
+0A20 A0                MOV @R0,A                   ;LEVEL1 MOV (MEM),R
 0A21 2301              MOV      A,#ENDT              ;end TRAP
 0A23                   TRAP
+0A23 35                DIS      TCNTI          ;START CRITICAL REGION
+0A24 8A20              ORL      P2,#X'20'
+0A26 05                EN       I
+0A27 00                NOP                     ;END CRITICAL REGION


               ;SUBROUTINES
                        ;CALCULATE PEAK POSITION RELATIVE TO INDEX
                        ;Subtract Peak-Index  (AL2-AL1) 12 bits
                        ;after nibble moves
 0A28          SUBMATH   EQU      *
 0A28                   LDX      A,(A1L)              ;set index low
+0A28                   SELMAC   MOVX,A,(A1L)
+0A28                   OPRM     MOVX,A,A1L       ;R0 OR R1 LOST
+0A28 B820              MOV      R0,#A1L            ;R0 LOST
+0A2A 80                MOVX A,@R0                  ;LEVEL1 OPCODE R,(MEM)
   2B 47                SWAP     A                   ;reorder bytes and nibbles
   2C AA                MOV      R2,A
 0A2D                   LDX      A,(A1H)             ;set index high
+0A2D                   SELMAC   MOVX,A,(A1H)
+0A2D                   OPRM     MOVX,A,A1H       ;R0 OR R1 LOST
+0A2D B810              MOV      R0,#A1H            ;R0 LOST
+0A2F 80                MOVX A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0A30 47                SWAP     A                   ;reorder bytes and nibbles
 0A31 B802              MOV      R0,#X'02'           ;addr. R2
 0A33 30                XCHD     A,@R0               ;swap low nibbles A,R2
                                                     ;A=High Byte,R2=Low Byte
 0A34 37                CPL      A                   ;A=1'sCPL High Byte
 0A35 2A                XCH      A,R2                ;A=Low Byte,R2=1'sHigh Byte
 0A36 37                CPL      A
 0A37 AB                MOV      R3,A                ;R3=1's CPL Low Byte
 0A38                   LDX      A,(A2H)             ;get PEAK high
+0A38                   SELMAC   MOVX,A,(A2H)
+0A3A                   OPRM     MOVX,A,A2H       ;R0 OR R1 LOST
+0A38 B830              MOV      R0,#A2H            ;R0 LOST
+0A3A 80                MOVX A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0A3B 47                SWAP     A                   ;reorder bytes and nibbles
 0A3C A9                MOV      R1,A                ;save it
 0A3D                   LDX      A,(A2L)             ;get PEAK low
+0A3D                   SELMAC   MOVX,A,(A2L)
+0A3D                   OPRM     MOVX,A,A2L       ;R0 OR R1 LOST
+0A3D B840              MOV      R0,#A2L            ;R0 LOST
+   3F 80               MOVX A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0A40 47                SWAP     A                   ;reorder bytes and nibbles
 0A41 B801              MOV      R0,#X'01'           ;addr. R1
 0A43 30                XCHD     A,@R0               ;A=LoByte,R1=HiByte
 0A44 97                CLR      C
```

```
0A45 A7                 CPL     C                   ;C=1 for 2's CPL add
0A46 7B                 ADDC    A,R3                ;AL2-AL1 low
  47 AB                 MOV     R3,A                ;R3=LoByte Difference
0A48 F9                 MOV     A,R1                ;get HiByte
0A49 7A                 ADDC    A,R2                ;A=HiByte Difference
0A4A 83                 RET

              ***************
              ;SYSYMTBL
                        PRNT    GEN
              ***************
              ***************
              ;APSYMTBL
                        PRNT    GEN
              ***************
              ***************
              ;APPADD.G
                        PRNT    GEN
              ***************
0A4B                    END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
ABORTT	0011	ACK	0006	ACPRG	0032	ACT	0000
TACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETA	09B6	BETAF	0036	BKBUFFO	004E
CAN	0018	CLRWRD	0E52	COMNOP	0000	CPH	0030
CPL	0050	DROPIN	002C	DROPOUT	002D	DSADD	0EBA
DSSUB	0ECC	ENDSEQ	0019	ENDT	0001	EOT	0004
ETX	0003	EVEQUE	0060	FASR	0E8D	FMINUS	0E73
FORWRK	0040	FPLUS	0E82	FRBUFFO	0044	FRNDOFF	0EAF
FTRUN	0E98	ICPRG	0031	IDRPIN	0050	IDRPOUT	0040
INACTACC	001D	INTMSG1	0001	INTMSG2	0002	INTMSG3	0003
INTMSG4	0004	INTMSG5	0005	INTMSG6	0006	INTMSG7	0007
ISTT	0008	IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020
LPRGLST	0020	LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002
MONMSG3	0003	MONMSG4	0004	MONMSG5	0005	MONMSG6	0006
MONMSG7	0007	MONMSG8	0008	MONMSG9	0009	MONRESMG	0010
MSGBUFF	002F	MSGBUFFO	0042	MTRO	0070	NAK	0015
NEWBETA	0033	NEWSUN	0035	NUMCOM	000C	NUMEVQ	0010
NUMUNITS	0001	OFFSET	0036	OPSTAT	001F	P1	0038
P2	0039	P3	003A	PAS	0010	PORT1	0020
PRGLSTAC	002B	PRGLSTUO	002A	PRGSTAT	001E	QUE1	0004
QUE2	000A	QUE3	000E	RDSNR	09B0	RECYCPER	000F
REGO3	0003	RIRAM	0002	RPRT	0008	RPSW	000B
RUNDLY	0008	RUNLIM	0070	RXIO	0006	RXRAM	0004
SCHMSG1	0001	SCHMSG2	0002	SCHMSG3	0003	SCHMSG4	0004
SHC	0080	SNDRECST	0026	SRCOMMSG	0027	SRPARM1	0028
ARM2	0029	SSUB	0E58	STT	0050	STX	0002
SUBMATH	0A28	SUNF	003B	SUNLVL	0A07	SUNLVL1	0A1C
SUPTRP	0002	SWSO	0080	SYSRES	0001	TIMVALH	0025
TIMVALL	0024	TRPMSG1	0001	TRPMSG2	0002	TRPMSG3	0003
TRPMSG4	0004	TRPMSG7	0007	TRUNADP	0E0B	TTRP	0009
UCM	0067	UNT	0000	URC	0063	URD	0060
URES	00B0	URM	0062	USD	0064	UST	0061
USY	0065	UWM	0066	VECTOR	0021	WARM	0004
WATADC	0043	WATCOMP	0013	WATIND	0023	WEVQ	000A
WINDOW	002E	WIRAM	0003	WPRT	0009	WXIO	0007
WXRAM	0005	XMITCHR	0023				

NO ERRORS

```
                ;FILTERS                                        12-19-80

                          PRNT    GEN
0000                      ORG     X'8D0'


                ;PROCEDURE:       FLTSW - Filter switch
                ;INPUTS:          NEWSUN
                ;OUTPUTS:         NEWSUN <- FF hex,
                ;                           if  C0 hex < NEWSUN
                ;                 NEWBETA <- 0000 hex,
                ;                           if  C0 hex < NEWSUN < 1/2 DROPOUT
                ;CALLS:           None
                ;DESTROYS:        ACC, R0, R1
                ;DESCRIPTION:     If the insolation level is greater than C0
                ;                 C0 hex or less than 1/2 the DROPOUT VALUE,
                ;                 don't drive the filter. The ADC is set so
                ;                 there is never an insolation level greater
                ;                 than C0 hex, for values greate than C0 hex
                ;                 set NEWSUN equal to FF hex to indicate an
                ;                 overflow.


08D0            FLTSW     EQU     *
08D0 B935                 MOV     R1,#NEWSUN
08D2 B82D                 MOV     R0,#DROPOUT
08D4 F0                   MOV     A,@R0           ;get DROPOUT
  D5 97                   CLR     C
08D6 67                   RRC     A               ;A=1/2 DROPOUT
08D7 37                   CPL     A               ;1's CPL 1/2 DROPOUT
08D8 61                   ADD     A,@R1           ;NEWSUN-1/2 DROPOUT-1
08D9 E6E2                 JNC     FLTSW1          ;NEWSUN<1/2 DROPOUT
08DB F1                   MOV     A,@R1           ;get NEWSUN
08DC 0340                 ADD     A,#(0-X'C0').AND.X'00FF'
08DE E6E6                 JNC     FLTSW2          ;SUNF<C0 hex,no overrange
08E0 B1FF                 MOV     @R1,#X'FF'      ;overrange NEWSUN <- FF hex
08E2 B833       FLTSW1    MOV     R0,#NEWBETA     ;1/2 DROPOUT<NEWSUN< C0 hex
08E4 D452                 CALL    CLRWRD          ;NEWBETA<-0 (2 bytes)
08E6            FLTSW2    EQU     *
```

```
;PROCEDURE:        BFILTER - Filter for BETA (error angle)
;INPUTS:           NEWSUN, NEWBETA, BETAF, P1, P2, P3
;OUTPUTS:          BETAF, P1, P2, P3
;CALLS:            FPLUS, DSADD, DSSUB, FASR, FRNDOFF,
;                  FTRUN, TRUNADP
;DESTROYS:         All
;DESCRIPTION:      BFILTER - Performs a 3 pole, 3 zero low pass
;                  filter on input NEWBETA and filter parameters
;                  P1, P2, P3 and generates output BETAF.
;                  FILTER EQUATION:
;                  Y(Z)/X(Z)=Y(Z)W(Z)/W(Z)X(Z)
;                          3  2
;                   1     Z  +Z  +Z+1
;                  -- * ---------------
;                  16      3      2
;                         Z 1.5Z   +Z-.25
;                  WHERE:
;                                        -1        -2          -3
;                  W(Z)=X(Z)+1.5W(Z)Z   -W(Z)Z   +.25W(Z)Z
;                  AND,
;                                       -1        -2        -3
;                  16Y(Z)=W(Z)+W(Z)Z   +W(Z)Z   +W(Z)Z
;                  BETAF=1/16Y(Z)
;                  EXPRESSING ALL VALUES RELATIVE TO BETAF:
;                  PI=(NEWBETA-BETAF)
;                  W(Z)Z=W(0)=P0+BETAF
;                        -1
;                  W(Z)Z   =P1+BETAF(Z)
;                        -2
;                  W(Z)Z   =P2+BETAF(Z)
;                        -3
;                  W(Z)Z   =P3+BETAF(Z)
;                  P0=1/4PI+1.5P1-P2+1/4P3
;                                          -1
;                  D=BETAF(Z)Z-BETAF(Z)Z   ,Displacement
;                  D=1/4(P0+P1+P2+P3)
;                                        -1
;                  BETAF(Z)=BETAF(Z)Z   +D
;                                 -1
;                  P3(Z)=P2(Z)Z   -D
;                                 -1
;                  P2(Z)=P1(Z)Z   -D
;                                 -1
;                  P1(Z)=P0(Z)Z   -D
```

```
                ;       ------------------------------------
                ;      ! -------------------------          !
                ;      !!                         !          !
                ;      !!              ---        ---        ---
                ;      !! -----------!1.5!      !-1 !      !.25!
                ;      !!!             ---        ---        ---
                ;      vvv              ^          ^          ^
                ;      ---      -----   !   -----  !   -----  !   ---
                ; X->!SUM!-*->!Z(-1)!-*->!Z(-1)!-*->!Z(-1)!-*->!SUM!->Y
                ;      ---   !W0 -----  !W1 -----  !W2 -----   W3 ---
                ;            !          !          !              ^^^
                ;            !          !          !              !!!
                ;            !          !           -------------- !!
                ;            !           ------------------------- !
                ;             ------------------------------------
                ;
                ;      ----
                ; Y->!1/16!->BETAF
                ;      ----

08E6            BFILTER   EQU     *
                ;Calculate P0
                          ;Do fractional math
08E6 B833                 MOV     R0,#NEWBETA
08E8 B936                 MOV     R1,#BETAF
[illegible]A D473         CALL    FMINUS          ;PI(NEWBETA)=NEWBETA-BETAF
[illegible]C B93C         MOV     R1,#AVGDPL      ;adaptive truncation parm.
08EE D40B                 CALL    TRUNADP         ;Truncate
08F0 B93A                 MOV     R1,#P3
08F2 D4BA                 CALL    DSADD           ;add P3
08F4 D48D                 CALL    FASR            ;div by 2,=1/2PI+1/2P3
08F6 B938                 MOV     R1,#P1
08F8 D4BA                 CALL    DSADD           ;add P1
08FA D48D                 CALL    FASR            ;div by 2
08FC D4AF                 CALL    FRNDOFF         ;(R0)=1/4PI+.5P1+1/4P3
                          ;Do integer math
08FE D4BA                 CALL    DSADD           ;add P1,=1/4PI+1.5P1+1/4P3
0900 B939                 MOV     R1,#P2
0902 D4CC                 CALL    DSSUB           ;sub P2,=1/4PI+1.5P1-P2+1/4P3
                          ;P0=1/4PI+1.5P1-P2+1/4P3
0904 B902                 MOV     R1,#2           ;temp. R2 addr.
0906 D498                 CALL    FTRUN           ;truncate R2=P0, -128<P0<+127
                ;Calculate BETAF
0908 B938                 MOV     R1,#P1
090A D4BA                 CALL    DSADD           ;add P1
090C B939                 MOV     R1,#P2
090E D4BA                 CALL    DSADD           ;add P2
0910 B93A                 MOV     R1,#P3
0912 D4BA                 CALL    DSADD           ;add P3
0914 D48D                 CALL    FASR            ;div. by 2
[illegible]6 D48D         CALL    FASR            ;(R0)=1/4(P0+P1+P2+P3)
                                                  ;displacement value
0918 D4AF                 CALL    FRNDOFF         ;round off value
091A B903                 MOV     R1,#3           ;temp R3 addr.
091C D498                 CALL    FTRUN           ;truncate R3=D, -128<P0<+127
091E B836                 MOV     R0,#BETAF
```

```
 0920 D4BA                CALL    DSADD          ;BETAF(0)=BETAF(1)+D
                ;Perform time correlation
 0922 B93A                MOV     R1,#P3
 0924 B839                MOV     R0,#P2         ;R0=P2,R1=P3
 0926 F0                  MOV     A,@R0
 0927 A1                  MOV     @R1,A          ;P2->P3
 0928 B938                MOV     R1,#P1         ;R1=P1,R0=P2
 092A F1                  MOV     A,@R1
 092B A0                  MOV     @R0,A          ;P1->P2
 092C FA                  MOV     A,R2           ;get P0,R1=P1
 092D A1                  MOV     @R1,A          ;P0->P1
                ;Rereferance Pn's to current BETAF
 092E B903                MOV     R1,#3          ;R3=DISPLACEMENT
 0930 B83A                MOV     R0,#P3
 0932 D458                CALL    SSUB           ;single precision SUB
 0934 B839                MOV     R0,#P2
 0936 D458                CALL    SSUB           ;single precision SUB
 0938 B838                MOV     R0,#P1
 093A D458                CALL    SSUB           ;single precision SUB
                ;Cauculate new averaged displacement value AVGDPL
 093C 85                  CLR     F0             ;F0=0 no roundup
 093D                     LD      A,(AVGDPL)
+093D                     SELMAC  MOV,A,(AVGDPL)
+093D                     OPRM    MOV,A,AVGDPL      ;R0 OR R1 LOST
+093D B83C                MOV     R0,#AVGDPL            ;R0 LOST
+093F F0                  MOV A,@R0                  ;LEVEL1 OPCODE R,(MEM)
   40 F7                  RLC     A              ;save sign
   41 F0                  MOV     A,@R0          ;get AVGDPL
 0942 67                  RRC     A              ;div by 2
 0943 E646                JNC     *+3            ;no roundup
 0945 95                  CPL     F0             ;set roundup flag
 0946 AA                  MOV     R2,A           ;save 1/2 AVGDPL
 0947 FB                  MOV     A,R3           ;get Displacement
 0948 F7                  RLC     A              ;save sign
 0949 FB                  MOV     A,R3           ;get Displacement
 094A 67                  RRC     A              ;div by 2
 094B E64E                JNC     *+3
 094D 85                  CLR     F0             ;roundup in ADDC, no F0
 094E 7A                  ADDC    A,R2           ;add prior roundup, if any
 094F 95                  CPL     F0             ;F0=1 no roundup
 0950 B653                JF0     *+3            ;no roundup
 0952 17                  INC     A              ;F0 roundup
 0953 A0                  MOV     @R0,A          ;store it
```

```
                ;PROCEDURE:        SUNFLT - Average sun level
                ;INPUTS:           NEWSUN - New value
                ;                  SUNF - Old value
                ;                  MSGBUFFO B1,B0 = SUNF B(-1),B(-2)
                ;OUTPUTS:          SUNF - Averaged insolation level
                ;                  MSGBUFFO B1,B0 = SUNF B(-1),B(-2)
                ;CALLS:            FASR, FPLUS
                ;DESTROYS:         All
                ;DESCRIPTION:      Filter is Y(0) = 3/4 Y(1) + 1/4 X
                ;                       Where :        X = Present input
                ;                                   Y(0) = Present output
                ;                                   Y(1) = Past output

 0954           SUNFLT    EQU     *
                ;Don't filter if NEWSUN > C0 hex
 0954                     LD      A,(NEWSUN)
+0954                     SELMAC  MOV,A,(NEWSUN)
+0954                     OPRM    MOV,A,NEWSUN        ;R0 OR R1 LOST
+0954 B835                MOV     R0,#NEWSUN              ;R0 LOST
+0956 F0                  MOV A,@R0                   ;LEVEL1 OPCODE R,(MEM)
 0957 37                  CPL     A
 0958 C698                JZ      SUNFLT1         ;NEWSUN > C0 hex, error
                ;Assemble complete SUNF, in R2,3 and BCKWRK
 095A 97                  CLR     C
 095B B93D                MOV     R1,#BCKWRK
 095D                     LD      A,(SUNF)
+095D                     SELMAC  MOV,A,(SUNF)
+095D                     OPRM    MOV,A,SUNF          ;R0 OR R1 LOST
+095D B83B                MOV     R0,#SUNF                ;R0 LOST
+095F F0                  MOV A,@R0                   ;LEVEL1 OPCODE R,(MEM)
 0960 F7                  RLC     A
 0961 AA                  MOV     R2,A
 0962 27                  CLR     A
 0963 F7                  RLC     A
 0964 2A                  XCH     A,R2
 0965 F7                  RLC     A
 0966 AB                  MOV     R3,A
 0967 FA                  MOV     A,R2
 0968 F7                  RLC     A
 0969 AA                  MOV     R2,A            ;R2,3=ASL (2) SUNF
 096A A1                  MOV     @R1,A
 096B 19                  INC     R1              ;R1=BCKWRK+1
 096C                     LD      A,(MSGBUFFO)    ;get SUNF B(-1),(-2)
+096C                     SELMAC  MOV,A,(MSGBUFFO)
+096C                     OPRM    MOV,A,MSGBUFFO      ;R0 OR R1 LOST
+096C B842                MOV     R0,#MSGBUFFO            ;R0 LOST
+096E F0                  MOV A,@R0                   ;LEVEL1 OPCODE R,(MEM)
 096F 53FC                ANL     A,#X'FC'
 0971 30                  XCHD    A,@R0           ;clr B1,0
 0972 5303                ANL     A,#X'03'        ;B(-1),B(-2) only
 0974 4B                  ORL     A,R3
 0975 AB                  MOV     R3,A            ;R2,3=SUNF,B7-B(-2)
 0976 A1                  MOV     @R1,A           ;BCKWRK=SUNF,B7-B(-2)
 0977 C9                  DEC     R1              ;R1=BCKWRK
                ;Calculate 3*SUNF
 0978 B802                MOV     R0,#2           ;addr R2,3
```

```
 097A D482                CALL    FPLUS
   7C D482                CALL    FPLUS          ;R2,3=3*4*SUNF
   7E D48D                CALL    FASR
 0980 D48D                CALL    FASR           ;R2,3=3*SUNF
                ;Calculate 3*SUNF+NEWSUN
 0982 B935                MOV     R1,#NEWSUN
 0984 F1                  MOV     A,@R1          ;get NEWSUN
 0985 6B                  ADD     A,R3
 0986 AB                  MOV     R3,A
                          ;Load new SUNF B(-1),(-2) in MSGBUFF B1,0
 0987 B942                MOV     R1,#MSGBUFF0
 0989 5303                ANL     A,#X'03'       ;B(-1),(-2) only
 098B 41                  ORL     A,@R1
 098C 31                  XCHD    A,@R1          ;store B(-1),(-2)
                          ;Add carry to SUNF B7-B0
 098D 27                  CLR     A
 098E 7A                  ADDC    A,R2
 098F AA                  MOV     R2,A           ;R2,3=NEWSUN+3*SUNF
                ;Calculate 3/4 SUNF+1/4 NEWSUN
 0990 D48D                CALL    FASR           ;R2,3=1/2NEWSUN+3/2SUNF
 0992 D48D                CALL    FASR           ;R2,3=1/4NEWSUN+3/4SUNF
 0994 FB                  MOV     A,R3
 0995                     LD      (SUNF),A
+0995                     SELMAC  MOV,(SUNF),A
+0995                     OPMR    MOV,SUNF,A       ;R0 OR R1 LOST
+0995 B83B                MOV     R0,#SUNF               ;R0 LOST
+0997 A0                  MOV @R0,A                  ;LEVEL1 MOV (MEM),R
   98            SUNFLT1  EQU     *
                ;End of module
 0998 2301                MOV     A,#ENDT
 099A                     TRAP
+099A 35                  DIS     TCNTI          ;START CRITICAL REGION
+099B 8A20                ORL     P2,#X'20'
+099D 05                  EN      I
+099E 00                  NOP                    ;END CRITICAL REGION
                **************
                ;SYSYMTBL
                          PRNT    GEN
                **************
                **************
                ;APSYMTBL
                          PRNT    GEN
                **************
                **************
                ;APPADD.G
                          PRNT    GEN
                **************
 099F                     END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
ORTT	0011	ACK	0006	ACPRG	0032	ACT	0000
TACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDFL	003C
BCKWRK	003D	BETAF	0036	BFILTER	08E6	BKBUFFO	004E
CAN	0018	CLRWRD	0E52	COMNOP	0000	CPH	0030
CPL	0050	DROPIN	002C	DROPOUT	002D	DSADD	0EBA
DSSUB	0ECC	ENDSEQ	0019	ENDT	0001	EOT	0004
ETX	0003	EVEQUE	0060	FASR	0E8D	FLTSW	08D0
FLTSW1	08E2	FLTSW2	08E6	FMINUS	0E73	FORWRK	0040
FPLUS	0E82	FRBUFFO	0044	FRNDOFF	0EAF	FTRUN	0E98
ICPRG	0031	IDRPIN	0050	IDRPOUT	0040	INACTACC	001D
INTMSG1	0001	INTMSG2	0002	INTMSG3	0003	INTMSG4	0004
INTMSG5	0005	INTMSG6	0006	INTMSG7	0007	ISTT	0008
IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020	LPRGLST	0020
LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002	MONMSG3	0003
MONMSG4	0004	MONMSG5	0005	MONMSG6	0006	MONMSG7	0007
MONMSG8	0008	MONMSG9	0009	MONRESMG	0010	MSGBUFF	002F
MSGBUFFO	0042	MTRO	0070	NAK	0015	NEWBETA	0033
NEWSUN	0035	NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001
OFFSET	0036	OPSTAT	001F	P1	0038	P2	0039
P3	003A	PAS	0010	PORT1	0020	PRGLSTAC	002B
PRGLSTUO	002A	PRGSTAT	001E	QUE1	0004	QUE2	000A
QUE3	000E	RECYCPER	000F	REG03	0003	RIRAM	0002
RPRT	0008	RPSW	000B	RUNDLY	0008	RUNLIM	0070
RXIO	0006	RXRAM	0004	SCHMSG1	0001	SCHMSG2	0002
[illegible]MSG3	0003	SCHMSG4	0004	SHC	0080	SNDRECST	0026
[illegible]OMMSG	0027	SRPARM1	0028	SRPARM2	0029	SSUB	0E58
STT	0050	STX	0002	SUNF	003B	SUNFLT	0954
SUNFLT1	0998	SUPTRP	0002	SWS0	0080	SYSRES	0001
TIMVALH	0025	TIMVALL	0024	TRPMSG1	0001	TRPMSG2	0002
TRPMSG3	0003	TRPMSG4	0004	TRPMSG7	0007	TRUNADP	0E0B
TTRP	0009	UCM	0067	UNT	0000	URC	0063
URD	0060	URES	00B0	URM	0062	USD	0064
UST	0061	USY	0065	UWM	0066	VECTOR	0021
WARM	0004	WATADC	0043	WATCOMP	0013	WATIND	0023
WEVQ	000A	WINDOW	002E	WIRAM	0003	WPRT	0009
WXIO	0007	WXRAM	0005	XMITCHR	0023		

NO ERRORS

```
                ;TRACK

                          PRNT    GEN
 0000                     ORG     X'C00'


                ;FUNCTION:          TRACK - Track Concentrator module
                ;INPUTS:            BETAF
                ;                   SUNF
                ;                   MTR(0)
                ;                   DROPOUT
                ;OUTPUTS:           MTR(0)
                ;                   PRGLSTU0, if SUNF.not.>DROPOUT
                ;                   EVEQUE
                ;CALLS:             None
                ;DESTROYS:          ACC,R0,R1,R2,R3
                ;DESCRIPTION:


 0C00           TRACK      EQU     *
                           ;Test fo Program Sequence Change
 0C00                      LD      A,(SUNF)
+0C00                      SELMAC  MOV,A,(SUNF)
+0C00                      OPRM    MOV,A,SUNF        ;R0 OR R1 LOST
+0C00 B83B                 MOV     R0,#SUNF              ;R0 LOST
+0C02 F0                   MOV A,@R0                     ;LEVEL1 OPCODE R,(MEM)
   03 37                   CPL     A                 ;1's CPL
 ..04                      ADDM    A,(DROPOUT)       ;DROPOUT-SUNF-1
+0C04                      SELMAC  ADD,A,(DROPOUT)
+0C04                      OPRM    ADD,A,DROPOUT         ;R0 OR R1 LOST
+0C04 B82D                 MOV     R0,#DROPOUT               ;R0 LOST
+0C06 60                   ADD A,@R0                     ;LEVEL1 OPCODE R,(MEM)
 0C07 F218                 JB7     TRACK1B           ;SUNF>DROPOUT, Keep Track Mode
                           ;Set program sequence to clock drive mode
 0C09                      LD      (EVEQUE),#0       ;clear any EVEQUE entry
+0C09                      SELMAC  MOV,(EVEQUE),#0
+0C09                      OPMIMD  MOV,EVEQUE,#0         ;R0 LOST
+0C09 B860                 MOV     R0,#EVEQUE                ;R0 LOST
+0C0B B000                 MOV @R0,#0
 0C0D                      LDX     (MTR0),#X'01'  ;scan MTR ON only
+0C0D                      SELMAC  MOVX,(MTR0),#X'01'
+0C0D                      OPMIMD  MOVX,MTR0,#X'01'      ;R0 LOST
+0C0D B870                 MOV     R0,#MTR0              ;R0 LOST
+0C0F 2301                 MOV     A,#X'01'                ;ACC LOST
+0C11 90                   MOVX @R0,A
 0C12                      LD      (PRGLSTU0),#X'82'
+0C12                      SELMAC  MOV,(PRGLSTU0),#X'82'
+0C12                      OPMIMD  MOV,PRGLSTU0,#X'82'       ;R0 LOST
+0C12 B82A                 MOV     R0,#PRGLSTU0              ;R0 LOST
+0C14 B082                 MOV @R0,#X'82'
 0C16 84B5                 JMP     TRACK13           ;Exit
   18           TRACK1B    LDX     A,(SWS0)          ;read switch status
+..18                      SELMAC  MOVX,A,(SWS0)
+0C18                      OPRM    MOVX,A,SWS0       ;R0 OR R1 LOST
+0C18 B880                 MOV     R0,#SWS0              ;R0 LOST
+0C1A 80                   MOVX A,@R0                    ;LEVEL1 OPCODE R,(MEM)
 0C1B 12A6                 JB0     TRACK10           ;Manual Mode
```

```
                                ;Convert Angular error to Absolute
   1D 85                        CLR     F0
 0C1E B936                      MOV     R1,#BETAF       ;Source
 0C20 B83D                      MOV     R0,#BCKWRK      ;Destination
 0C22 D452                      CALL    CLRWRD          ;CLR 2 bytesBCKWK
 0C24 F1                        MOV     A,@R1           ;get msbyte BETAF
 0C25 F22B                      JB7     TRACK1          ;BEATF negative
 0C27 D482                      CALL    FPLUS           ;BETAF positive
 0C29 842E                      JMP     TRACK2
 0C2B D473          TRACK1      CALL    FMINUS          ;BETAF negative
 0C2D 95                        CPL     F0              ;F0=1 error negative
 0C2E               TRACK2      EQU     *               ;BCKWRK=!BETAF!
                    ;Test if Error is greater than Continuous Run Limit
                    ;C=1, if !ERROR!)RUNLIM
                                ;R0=BCKWRK
 0C2E BAFF                      MOV     R2,#X'FF'       ;Default Error value
 0C30 F0                        MOV     A,@R0
 0C31 9636                      JNZ     TRACK3          ;value)255
 0C33 18                        INC     R0              ;addr lsbyte
 0C34 F0                        MOV     A,@R0
 0C35 AA                        MOV     R2,A            ;value(256
 0C36 2390          TRACK3      MOV     A,#(0-RUNLIM).AND.X'00FF'
 0C38 6A                        ADD     A,R2
 0C39 E646                      JNC     TRACK5          ;fime track mode
                    ;Continous Run, if !ERROR!)RUNLIM
 0C3B 2307                      MOV     A,#X'07'        ;scan  MTR(0) On
                                                        ;posit MTR(0) On
                                                        ;rotat MTR(0) CCW
 0C3D B641                      JF0     TRACK4          ;error negative
 0C3F D304                      XRL     A,#X'04'        ;rotat MTR(0) CW
 0C41               TRACK4      LDX     (MTR0),A        ;run motor
+0C41                           SELMAC  MOVX,(MTR0),A
+0C41                           OPMR    MOVX,MTR0,A       ;R0 OR R1 LOST
+0C41 B870                      MOV     R0,#MTR0              ;R0 LOST
+0C43 90                        MOVX @R0,A                  ;LEVEL1 MOV (MEM),R
 0C44 84AB                      JMP     TRACK11
                    ;Fine track, if !ERROR!(RUNLIM
 0C46               TRACK5      EQU     *
                                ;Convert WINDOW in 1/4(1/10) Deg's to 1/16(1/10)Deg's
 0C46 BA00                      MOV     R2,#0
 0C48                           LD      A,(WINDOW)
+0C48                           SELMAC  MOV,A,(WINDOW)
+0C48                           OPRM    MOV,A,WINDOW      ;R0 OR R1 LOST
+0C48 B82E                      MOV     R0,#WINDOW            ;R0 LOST
+0C4A F0                        MOV A,@R0                   ;LEVEL1 OPCODE R,(MEM)
 0C4B AB                        MOV     R3,A
 0C4C B802                      MOV     R0,#2           ;addr R2
 0C4E B902                      MOV     R1,#2           ;addr R2
 0C50 D482                      CALL    FPLUS           ;2(WINDOW)
 0C52 D482                      CALL    FPLUS           ;4(WINDOW)
 0C54 B83D                      MOV     R0,#BCKWRK      ;addr !ERROR ANGLE!
 0C56 D473                      CALL    FMINUS
 0C58 F0                        MOV     A,@R0           ;get sign bit
 0C59 F29B                      JB7     TRACK8          ;!ERROR ANGLE!(WINDOW
                    ;!ERROR ANGLE!)WINDOW
 0C5B                           LDX     A,(MTR0)        ;get posit MTR staus
+0C5B                           SELMAC  MOVX,A,(MTR0)
```

```
+0C5B                    OPRM    MOVX,A,MTR0          ;R0 OR R1 LOST
+  5B B870               MOV     R0,#MTR0                 ;R0 LOST
+0C5D 80                 MOVX A,@R0                       ;LEVEL1 OPCODE R,(MEM)
 0C5E 32A1               JB1     TRACK9             ;Motor ON
                         ;Posit. Motor OFF
 0C60                    LD      A,(EVEQUE)         ;must NOT have EVEQUE entry
+0C60                    SELMAC  MOV,A,(EVEQUE)
+0C60                    OPRM    MOV,A,EVEQUE          ;R0 OR R1 LOST
+0C60 B860               MOV     R0,#EVEQUE                ;R0 LOST
+0C62 F0                 MOV A,@R0                       ;LEVEL1 OPCODE R,(MEM)
 0C63 96A6               JNZ     TRACK10            ;there is an entry, RESET
                         ;Increment !ERROR ANGLE!>WINDOW count
 0C65                    LD      A,(MSGBUFF0)
+0C65                    SELMAC  MOV,A,(MSGBUFF0)
+0C65                    OPRM    MOV,A,MSGBUFF0          ;R0 OR R1 LOST
+0C65 B842               MOV     R0,#MSGBUFF0                ;R0 LOST
+0C67 F0                 MOV A,@R0                       ;LEVEL1 OPCODE R,(MEM)
 0C68 47                 SWAP    A
 0C69 17                 INC     A
 0C6A 5303               ANL     A,#X'03'           ;A=0, if count was 4
 0C6C C676               JZ      TRACK6
 0C6E 47                 SWAP    A                  ;restore nibble positions
 0C6F 20                 XCH     A,@R0
 0C70 53CF               ANL     A,#X'CF'           ;clr B5,B4
 0C72 60                 ADD     A,@R0
 0C73 20                 XCH     A,@R0              ;store new count
    4 84B5               JMP     TRACK13            ;Exit
                         ;Compute Run Time
                         ;R2,R3=4(WINDOW)
 0C76          TRACK6    LD      A,(WINDOW)
+0C76                    SELMAC  MOV,A,(WINDOW)
+0C76                    OPRM    MOV,A,WINDOW          ;R0 OR R1 LOST
+0C76 B82E               MOV     R0,#WINDOW                ;R0 LOST
+0C78 F0                 MOV A,@R0                       ;LEVEL1 OPCODE R,(MEM)
 0C79 97                 CLR     C
 0C7A 67                 RRC     A                  ;.5(WINDOW)
 0C7B 7B                 ADDC    A,R3               ;roundoff
 0C7C AB                 MOV     R3,A               ;R3=4.5(WINDOW)
 0C7D 27                 CLR     A
 0C7E 7A                 ADDC    A,R2
 0C7F 96A6               JNZ     TRACK10            ;OVERFLOW EXIT
 0C81 FB                 MOV     A,R3               ;get MTR run time
 0C82 0308               ADD     A,#RUNDLY          ;add MTR run delay time
 0C84 F6A6               JC      TRACK10            ;OVERFLOW EXIT
 0C86 AB                 MOV     R3,A               ;store run time
 0C87 BA04               MOV     R2,#QUE1
 0C89 2307               MOV     A,#X'07'           ;scan  MTR(0) On
                                                    ;posit MTR(0) On
                                                    ;rotat MTR(0) CCW
 0C8B B68F               JF0     TRACK7             ;Error Negative
 0C8D D304               XRL     A,#X'04'           ;rotat MTR(0) CW
 (   F          TRACK7    LDX     (MTR0),A           ;Run Motor
+(   F                   SELMAC  MOVX,(MTR0),A
+0C8F                    OPMR    MOVX,MTR0,A          ;R0 OR R1 LOST
+0C8F B870               MOV     R0,#MTR0                  ;R0 LOST
+0C91 90                 MOVX @R0,A                      ;LEVEL1 MOV (MEM),R
 0C92 2309               MOV     A,#TTRP            ;time trap
```

```
 0C94                    TRAP
+  94 35                 DIS     TCNTI          ;START CRITICAL REGION
+  95 8A20               ORL     P2,#X'20'
+0C97 05                 EN      I
+0C98 00                 NOP                    ;END CRITICAL REGION
 0C99 84AF               JMP     TRACK12
                ;!ERROR ANGLE!<WINDOW
 0C9B           TRACK8   EQU     *              ;get posit MTR staus
 0C9B                    LDX     A,(MTR0)
+0C9B                    SELMAC  MOVX,A,(MTR0)
+0C9B                    OPRM    MOVX,A,MTR0      ;R0 OR R1 LOST
+0C9B B870               MOV     R0,#MTR0            ;R0 LOST
+0C9D 80                 MOVX A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0C9E 37                 CPL     A
 0C9F 32A6               JB1     TRACK10        ;MTR OFF
                         ;Posit Motor ON
 0CA1           TRACK9   LD      A,(EVEQUE)     ;Must have EVEQUE entry
+0CA1                    SELMAC  MOV,A,(EVEQUE)
+0CA1                    OPRM    MOV,A,EVEQUE      ;R0 OR R1 LOST
+0CA1 B860               MOV     R0,#EVEQUE           ;R0 LOST
+0CA3 F0                 MOV A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0CA4 96AF               JNZ     TRACK12        ;There is an entry, EXIT
                ;Manual MOde or !ERROR ANGLE!<WINDOW motor off
 0CA6           TRACK10  EQU     *
 0CA6                    LDX     (MTR0),#X'01'  ;Scan Motor ON only
+0CA6                    SELMAC  MOVX,(MTR0),#X'01'
+  A6                    OPMIMD  MOVX,MTR0,#X'01'      ;R0 LOST
+  A6 B870               MOV     R0,#MTR0              ;R0 LOST
+0CA8 2301               MOV     A,#X'01'                ;ACC LOST
+0CAA 90                 MOVX @R0,A
                ;Clear any EVEQUE entry
 0CAB           TRACK11  EQU     *
 0CAB 27                 CLR     A
 0CAC                    LD      (EVEQUE),A
+0CAC                    SELMAC  MOV,(EVEQUE),A
+0CAC                    OPMR    MOV,EVEQUE,A      ;R0 OR R1 LOST
+0CAC B860               MOV     R0,#EVEQUE              ;R0 LOST
+0CAE A0                 MOV @R0,A                  ;LEVEL1 MOV (MEM),R
                ;Clear !ERROR ANGL!>WINDOW count
 0CAF           TRACK12  EQU     *
 0CAF                    LD      A,(MSGBUFF0)
+0CAF                    SELMAC  MOV,A,(MSGBUFF0)
+0CAF                    OPRM    MOV,A,MSGBUFF0      ;R0 OR R1 LOST
+0CAF B842               MOV     R0,#MSGBUFF0            ;R0 LOST
+0CB1 F0                 MOV A,@R0                  ;LEVEL1 OPCODE R,(MEM)
 0CB2 53CF               ANL     A,#X'CF'       ;clear B5,B4
 0CB4 A0                 MOV     @R0,A          ;store it
                ;Exit
 0CB5 2301      TRACK13  MOV     A,#ENDT
 0CB7                    TRAP
+0CB7 35                 DIS     TCNTI          ;START CRITICAL REGION
+  B8 8A20               ORL     P2,#X'20'
+  A 05                 EN      I
+0CBB 00                 NOP                    ;END CRITICAL REGION
                **************
                ;SYSYMTBL
                         PRNT    GEN
```

```
                 ***************
                 ***************
                 ;APSYMTBL
                           PRNT    GEN
                 ***************
                 ***************
                 ;APPADD.G
                           PRNT    GEN
                 ***************
0CBC                       END
```

A1H	0010	A1L	0020	A2H	0030	A2L	0040
[illegible]RTT	0011	ACK	0006	ACPRG	0032	ACT	0000
[illegible]ACC	001C	ACTDIGFL	003D	ACTEXAR	0033	ADC	0090
AL1	0020	AL2	0040	ANS0	00A0	AVGDPL	003C
BCKWRK	003D	BETAF	0036	BKBUFF0	004E	CAN	0018
CLRWRD	0E52	COMNDP	0000	CPH	0030	CPL	0050
DROPIN	002C	DROPOUT	002D	DSADD	0EBA	DSSUB	0ECC
ENDSEQ	0019	ENDT	0001	EOT	0004	ETX	0003
EVEQUE	0060	FASR	0E8D	FMINUS	0E73	FORWRK	0040
FPLUS	0E82	FRBUFF0	0044	FRNDOFF	0EAF	FTRUN	0E98
ICPRG	0031	IDRPIN	0050	IDRPOUT	0040	INACTACC	001D
INTMSG1	0001	INTMSG2	0002	INTMSG3	0003	INTMSG4	0004
INTMSG5	0005	INTMSG6	0006	INTMSG7	0007	ISTT	0008
IWINDOW	0003	LACTEXAR	000A	LEVNTQUE	0020	LPRGLST	0020
LPRGSEQ	0040	MONMSG1	0001	MONMSG2	0002	MONMSG3	0003
MONMSG4	0004	MONMSG5	0005	MONMSG6	0006	MONMSG7	0007
MONMSG8	0008	MONMSG9	0009	MONRESMG	0010	MSGBUFF	002F
MSGBUFF0	0042	MTR0	0070	NAK	0015	NEWBETA	0033
NEWSUN	0035	NUMCOM	000C	NUMEVQ	0010	NUMUNITS	0001
OFFSET	0036	OPSTAT	001F	P1	0038	P2	0039
P3	003A	PAS	0010	PORT1	0020	PRGLSTAC	002B
PRGLSTU0	002A	PRGSTAT	001E	QUE1	0004	QUE2	000A
QUE3	000E	RECYCPER	000F	REG03	0003	RIRAM	0002
RPRT	0008	RPSW	000B	RUNDLY	0008	RUNLIM	0070
RXIO	0006	RXRAM	0004	SCHMSG1	0001	SCHMSG2	0002
SCHMSG3	0003	SCHMSG4	0004	SHC	0080	SNDRECST	0026
[illegible]COMMSG	0027	SRPARM1	0028	SRPARM2	0029	SSUB	0E58
[illegible]	0050	STX	0002	SUNF	003B	SUPTRP	0002
SWS0	0080	SYSRES	0001	TIMVALH	0025	TIMVALL	0024
TRACK	0C00	TRACK1	0C2B	TRACK10	0CA6	TRACK11	0CAB
TRACK12	0CAF	TRACK13	0CB5	TRACK1B	0C18	TRACK2	0C2E
TRACK3	0C36	TRACK4	0C41	TRACK5	0C46	TRACK6	0C76
TRACK7	0C8F	TRACK8	0C9B	TRACK9	0CA1	TRPMSG1	0001
TRPMSG2	0002	TRPMSG3	0003	TRPMSG4	0004	TRPMSG7	0007
TRUNADP	0E0B	TTRP	0009	UCM	0067	UNT	0000
URC	0063	URD	0060	URES	00B0	URM	0062
USD	0064	UST	0061	USY	0065	UWM	0066
VECTOR	0021	WARM	0004	WATADC	0043	WATCOMP	0013
WATIND	0023	WEVQ	000A	WINDOW	002E	WIRAM	0003
WPRT	0009	WXIO	0007	WXRAM	0005	XMITCHR	0023

NO ERRORS

```
                ;APPSUB.S

                          PRNT    GEN
0000                      ORG     X'EOB'


                ;TRUNADP

                ;SUBROUTINE:      TRUNADP - TRUNCATES A 2 BYTE VALUE TO
                ;                 10,8,7, or 6 bits, including sign.
                ;INPUTS:          R0 - pointer to msby 2 byte source,
                ;                 and destination
                ;                 R1 is a pointer to AVGDL.
                ;                 AVGDPL - Averaged displacement value used
                ;                 to determine the level of truncation.
                ;OUTPUTS:         (R0) <-  +31>(R0))-32 , if  +5>AVGDPL>-5
                ;                 (R0) <-  +63>(R0))-64 , if  +9>AVGDPL>-9
                ;                 (R0) <- +127>(R0))-128, if +17>AVGDPL>-17
                ;                 (R0) <- +511>(R0))-512, if +16<AVGDPL<-16
                ;CALLS:           None
                ;DESTROYS:        ACC,R1,R2,F0
                ;DESCRIPTION:     This subroutine is used as a part of an
                ;                 adaptive filter. To reduce the effect of
                ;                 non-correlated noise. It performs an
                ;                 adaptive truncation.

   0B 85        TRUNADP   CLR     F0
   0C F0                  MOV     A,@R0
0E0D F210                 JB7     TRUNADP1
0E0F 95                   CPL     F0                ;F0=1,0=Pos.,Neg
0E10            TRUNADP1  EQU     *
0E10 F1                   MOV     A,@R1             ;get AVGDPL
0E11 F217                 JB7     TRUNADP2          ;negative number
0E13 C62F                 JZ      TRUNADP3          ;0 not mapped
0E15 07                   DEC     A                 ;symetrical +,- numbers
0E16 37                   CPL     A                 ;uniform bit testing
0E17 F7         TRUNADP2  RLC     A                 ;B7->C
0E18 D900                 MOV     R1,#X'00'         ;lsbyte mask
0E1A BAFE                 MOV     R2,#X'FE'         ;msbyte mask
0E1C F7                   RLC     A                 ;get B6
0E1D E633                 JNC     TRUNADP5          ;10 Bit truncation
0E1F F7                   RLC     A                 ;get B5
0E20 E633                 JNC     TRUNADP5          ;10 Bit truncation
0E22 F7                   RLC     A                 ;get B4
0E23 E633                 JNC     TRUNADP5          ;10 Bit truncation
0E25 B980                 MOV     R1,#X'80'
0E27 F7                   RLC     A                 ;get B3
0E28 E631                 JNC     TRUNADP4          ; 8 Bit truncation
0E2A B9C0                 MOV     R1,#X'C0'
0E2C F7                   RLC     A                 ;get B2
0E2D E631                 JNC     TRUNADP4          ;7 Bit truncation
   F B9E0       TRUNADP3  MOV     R1,#X'E0'         ;lsbyte,6 Bit truncation
   1 BAFF       TRUNADP4  MOV     R2,#X'FF'         ;msbyte mask
0E33 F0         TRUNADP5  MOV     A,@R0             ;get msbyte
0E34 B637                 JF0     TRUNADP6          ;positive number
0E36 37                   CPL     A
0E37 5A         TRUNADP6  ANL     A,R2              ;B15-B8 MASK
```

```
0E38 9643                JNZ     TRUNADP8        ;must truncate
  3A 18                  INC     R0
0E3B F0                  MOV     A,@R0           ;get lsbyte
0E3C B63F                JFO     TRUNADP7        ;positive number
0E3E 37                  CPL     A
0E3F 59        TRUNADP7  ANL     A,R1            ;B7-B0 mask
0E40 C8                  DEC     R0              ;restore R0
0E41 C651                JZ      TRUNADPB        ;no truncation
                         ;must truncate
0E43 FA        TRUNADP8  MOV     A,R2            ;B15-B8 mask
0E44 37                  CPL     A
0E45 B648                JFO     TRUNADP9        ;pos.number
0E47 37                  CPL     A               ;neg. number
0E48 A0        TRUNADP9  MOV     @R0,A           ;load msbyte
0E49 18                  INC     R0              ;addr.lsbyte
0E4A F9                  MOV     A,R1            ;B7-B0 mask
0E4B 37                  CPL     A
0E4C B64F                JFO     TRUNADPA        ;pos. number
0E4E 37                  CPL     A               ;neg. number
0E4F A0        TRUNADPA  MOV     @R0,A           ;load lsbyte
0E50 C8                  DEC     R0              ;restore R0
0E51 83        TRUNADPB  RET


               ;CLRWRD

               ;SUBROUTINE:      CLRWRD - sets 2 bytes equal to zero
               ;INPUTS:          R0 - pointer to 2 byte destination
               ;OUTPUTS:         (R0),(R0+1)(-0
               ;CALLS:           None
               ;DESTROYS:        ACC

0E52 27        CLRWRD    CLR     A
0E53 A0                  MOV     @R0,A           ;(R0)(-0
0E54 18                  INC     R0
0E55 A0                  MOV     @R0,A           ;(R0+1)(-0
0E56 C8                  DEC     R0              ;restore R0
0E57 83                  RET


               ;SSUB

               ;SUBROUTINE:      SSUB - single precision subtraction
               ;INPUTS:          R0 - pointer to 1 byte source
               ;OUTPUTS:         (R0)= (R0)-(R1)
               ;CALLS:           None
               ;DESTROYS:        ACC,R2,F0
               ;DESCRIPTION:     (R0)=-128 if (R0)-(R1)(-128
               ;                 (R0)=+127 if (R0)-(R1)>+127
               ;                 Otherwise:
               ;                 (R0)=(R0)-(R1)
               ;                 OVERFLOW CHECKED

0E58 F1        SSUB      MOV     A,@R1
0E59 37                  CPL     A
0E5A 17                  INC     A
0E5B AA                  MOV     R2,A            ;save 2's CPL
```

```
0E5C F0                 MOV     A,@R0
   D D1                 XRL     A,@R1          ;opposite signs?
0E5E F7                 RLC     A
0E5F F0                 MOV     A,@R0          ;get value
0E60 85                 CLR     F0
0E61 F664               JC      SSUB1          ;signs opposite possible ovl
                        ;signs are the opposite no overflow possible
0E63 95                 CPL     F0
                        ;opposite signs, possible overflow
0E64 6A         SSUB1   ADD     A,R2
0E65 A0                 MOV     @R0,A          ;store result
0E66 B672               JF0     SSUB2          ;never overflow
0E68 F7                 RLC     A              ;save sign
0E69 F0                 MOV     A,@R0          ;get result
0E6A D1                 XRL     A,@R1
0E6B F272               JB7     SSUB2          ;sign didn't change
                        ;result overflow  C=1 if positive overflow
0E6D B07F               MOV     @R0,#X'7F'     ;preload max.pos.
0E6F F672               JC      SSUB2          ;positive overflow
0E71 10                 INC     @R0            ;C=0,negative underflow
0E72 83         SSUB2   RET


                ;FMINUS

                ;SUBROUTINE:        FMINUS - Double byte substraction
                ;INPUTS:            R0 - pointer to msby minuend
                ;                   R1 - pointer to msby subtrahend
                ;OUTPUTS:           in minuend
                ;CALLS:             FRNDOFF
                ;DESTROYS:          ACC,R2
                ;DESCRIPTION:       (R0)((R0)-(R1)

0E73 18         FMINUS  INC     R0             ;point to lsby's
0E74 19                 INC     R1
0E75 F1                 MOV     A,@R1
0E76 37                 CPL     A              ;1's CPL lsby (R1)
0E77 97                 CLR     C
0E78 A7                 CPL     C              ;C=1
0E79 70                 ADDC    A,@R0          ;2's CPL add
0E7A A0                 MOV     @R0,A          ;save it
0E7B C9                 DEC     R1             ;addr. msby's
0E7C C8                 DEC     R0             ;addr. msby's
0E7D F1                 MOV     A,@R1          ;A=msby(R1)
0E7E 37                 CPL     A              ;1's CPL msby (R1)
0E7F 70                 ADDC    A,@R0          ;A=2'sCPLmsby
0E80 A0                 MOV     @R0,A          ;save it
0E81 83                 RET


                ;FPLUS

                ;SUBROUTINE:        FPLUS - Double byte addition
                ;INPUTS:            R0 - pointer to msby A
                ;                   R1 - pointer to msby B
                ;OUTPUTS:           in ((R0))  C=A+B
                ;CALLS:             None
```

```
                ;DESTROYS:          ACC
                ;DESCRIPTION:       (R0)((R0)+(R1)
                ;                   NOTE: Doesn't check for overflow

0E82 18         FPLUS     INC       R0              ;point to lsby's
0E83 19                   INC       R1
0E84 F0                   MOV       A,@R0
0E85 61                   ADD       A,@R1
0E86 A0                   MOV       @R0,A           ;store lsby
0E87 C8                   DEC       R0              ;point,msby's
0E88 C9                   DEC       R1
0E89 F0                   MOV       A,@R0
0E8A 71                   ADDC      A,@R1
0E8B A0                   MOV       @R0,A           ;store msby
0E8C 83                   RET


                ;FASR

                ;SUBROUTINE:        FASR - Double byte arithmetic shift right
                ;INPUTS:            R0 - pointer to msby
                ;OUTPUTS:           (R0) <- (R0)/2, C <- B0
                ;CALLS:             None
                ;DESTROYS:          ACC
                ;DESCRIPTION:       (R0)((R0)/2

0E8D F0         FASR      MOV       A,@R0           ;get sign
0E8E F7                   RLC       A               ;B7>C,sign extension
0E8F F0                   MOV       A,@R0           ;get msby
0E90 67                   RRC       A               ;ASR msby
0E91 A0                   MOV       @R0,A           ;store msby
0E92 18                   INC       R0              ;lsby
0E93 F0                   MOV       A,@R0           ;get lsby
0E94 67                   RRC       A               ;ASR lsby
0E95 A0                   MOV       @R0,A           ;store lsby
0E96 C8                   DEC       R0              ;restore R0 addr
0E97 83                   RET


                ;FTRUN

                ;SUBROUTINE:        FTRUN - Truncates a 2 byte value to 1 byte
                ;INPUTS:            R0 - pointer to msby 2 byte source
                ;                   R1 - pointer to 1 byte destination
                ;OUTPUTS:           (R1) <- (R0),          if +128>(R0)>-129
                ;                   (R1) <- +127(7F hex), if +128>(R0)
                ;                   (R1) <- -128(80 hex), if      (R0)>-129
                ;CALLS:             None
                ;DESTROYS:          ACC
                ;DESCRIPTION:       (R1)<-(R0)
                ;                   NOTE: Doesn't check for overflow

0E98 B17F       FTRUN     MOV       @R1,#X'7F'      ;initial value
0E9A F0                   MOV       A,@R0           ;get sign
0E9B 18                   INC       R0              ;addr lsby
0E9C F2A6                 JB7       FTRUN3          ;negative number
                          ;positive number
```

```
0E9E 96A4                JNZ     FTRUN2+1      ;truncate,positive number(+255
  A0 F0                  MOV     A,@R0         ;get lsby
0EA1 F2A4                JB7     FTRUN2+1      ;truncate,positive number(+127
                         ;no truncation, +128)(R0))-129
0EA3 A1        FTRUN2    MOV     @R1,A         ;store number
0EA4 C8                  DEC     R0            ;restore R0
0EA5 83                  RET
                         ;negative number
0EA6 11        FTRUN3    INC     @R1           ;(R1)=-128
0EA7 37                  CPL     A
0EA8 96A4                JNZ     FTRUN2+1      ;truncate,negative number(-255
0EAA F0                  MOV     A,@R0         ;get lsby
0EAB F2A3                JB7     FTRUN2        ;store it
                                               ;truncate,negative number(-128
                         ;truncate number (R1)=+127 or -128
0EAD C4A4                JMP     FTRUN2+1      ;truncate


               ;FRNDOFF

               ;SUBROUTINE:      FRNDOFF - Rounds off a 2 byte value, using C
               ;INPUTS:          R0 - pointer to msby
               ;OUTPUTS:         (R0) <- (R0)+C
               ;CALLS:           None
               ;DESTROYS:        ACC
               ;DESCRIPTION:     (R0)((R0)+(R1)

0EAF 18        FRNDOFF   INC     R0            ;point to lsby
0EB0 F0                  MOV     A,@R0         ;get lsby
0EB1 1300                ADDC    A,#0
0EB3 A0                  MOV     @R0,A         ;store lsby
0EB4 C8                  DEC     R0            ;point to msby
0EB5 F0                  MOV     A,@R0         ;get msby
0EB6 1300                ADDC    A,#0
0EB8 A0                  MOV     @R0,A         ;store msby
0EB9 83                  RET


               ;DSADD

               ;SUBROUTINE:      FPLUS - Double byte addition using A
               ;                 double byte destination, and a single
               ;                 byte source, B7(R1) is extended.
               ;INPUTS:          R0 - pointer to msby A Double precision
               ;                 R1 - pointer to      B Single precison
               ;OUTPUTS:         in ((R0))  C=A+B
               ;CALLS:           None
               ;DESTROYS:        ACC
               ;DESCRIPTION:     (R0)((R0)+(R1)
               ;                 NOTE: Doesn't check for overflow

  3A 18        DSADD     INC     R0            ;point to lsby
  3B F0                  MOV     A,@R0         ;get lsby
0EBC 61                  ADD     A,@R1
0EBD A0                  MOV     @R0,A         ;store lsby
0EBE C8                  DEC     R0            ;point msby's
0EBF F1                  MOV     A,@R1         ;get sign bit
```

```
0EC0 F2C7                JB7     DSADD1          ;negative number
                         ;positive number
0EC2 F0                  MOV     A,@R0           ;get msby
0EC3 1300                ADDC    A,#0
0EC5 C4CA                JMP     DSADD2
0EC7 F0          DSADD1  MOV     A,@R0           ;get msby
0EC8 13FF                ADDC    A,#X'FF'
0ECA A0          DSADD2  MOV     @R0,A           ;store msby
0ECB 83                  RET


                 ;DSSUB

                 ;SUBROUTINE:    DSSUB - Double byte subtraction using A
                 ;               double byte destination, and a single
                 ;               byte source. B7(R1) is extended.
                 ;INPUTS:        R0 - pointer to msby A Double precision
                 ;               R1 - pointer to      B Single precison
                 ;OUTPUTS:       in ((R0))  C=A-B
                 ;CALLS:         FRNDOFF
                 ;DESTROYS:      ACC
                 ;DESCRIPTION:   (R0)((R0)-(R1)
                 ;               NOTE: Doesn't check for overflow

0ECC 18          DSSUB   INC     R0              ;point to lsby
0ECD F1                  MOV     A,@R1           ;get single precision
0ECE 37                  CPL     A               ;1's CPL single prec.
0ECF 97                  CLR     C
0ED0 A7                  CPL     C               ;C=1 for 2's CPL add
0ED1 70                  ADDC    A,@R0           ;2's CPL add
0ED2 A0                  MOV     @R0,A           ;store lsby
0ED3 C8                  DEC     R0              ;point msby's
0ED4 F1                  MOV     A,@R1           ;get sign bit
0ED5 F2DC                JB7     DSSUB1          ;negative number
                         ;positive number
0ED7 F0                  MOV     A,@R0           ;get msby
0ED8 13FF                ADDC    A,#X'FF'
0EDA C4DF                JMP     DSSUB2
                         ;negative number
0EDC F0          DSSUB1  MOV     A,@R0           ;get msby
0EDD 1300                ADDC    A,#X'00'
0EDF A0          DSSUB2  MOV     @R0,A           ;store msby 1's CPL
0EE0 83                  RET


0EE1                     END
```

CLRWRD	0E52	DSADD	0EBA	DSADD1	0EC7	DSADD2	0ECA
SUB	0ECC	DSSUB1	0EDC	DSSUB2	0EDF	FASR	0E8D
FMINUS	0E73	FPLUS	0E82	FRNDOFF	0EAF	FTRUN	0E98
FTRUN2	0EA3	FTRUN3	0EA6	SSUB	0E58	SSUB1	0E64
SSUB2	0E72	TRUNADP	0E0B	TRUNADP1	0E10	TRUNADP2	0E17
TRUNADP3	0E2F	TRUNADP4	0E31	TRUNADP5	0E33	TRUNADP6	0E37
TRUNADP7	0E3F	TRUNADP8	0E43	TRUNADP9	0E48	TRUNADPA	0E4F
TRUNADPB	0E51						

NO ERRORS

APPENDIX II

DEMONSTRATION UNIT

In order to evaluate the "POINT" solar tracking system's:

a. Tracking capability;

b. Tracking accuracy; and

c. Field of View.

a demonstration unit was designed and built. This unit simulates the dynamic operating conditions of a line focus concentrator, and thereby provided the means to measure the parameters given above under realistic rather than idealistic circumstances. This is important especially with respect to item b), as a systems TRACKING ACCURACY encompasses not only it's ability to determine the sun's position to the specified accuracy but also it's ability to respond to this data by repositioning the concentrator within the limits of the specified value.

The design of the DEMONSTRATION UNIT is shown diagrammatically on drawing No. 107.

It is comprised of the following major components;

a. Motor;

b. Gearbox;

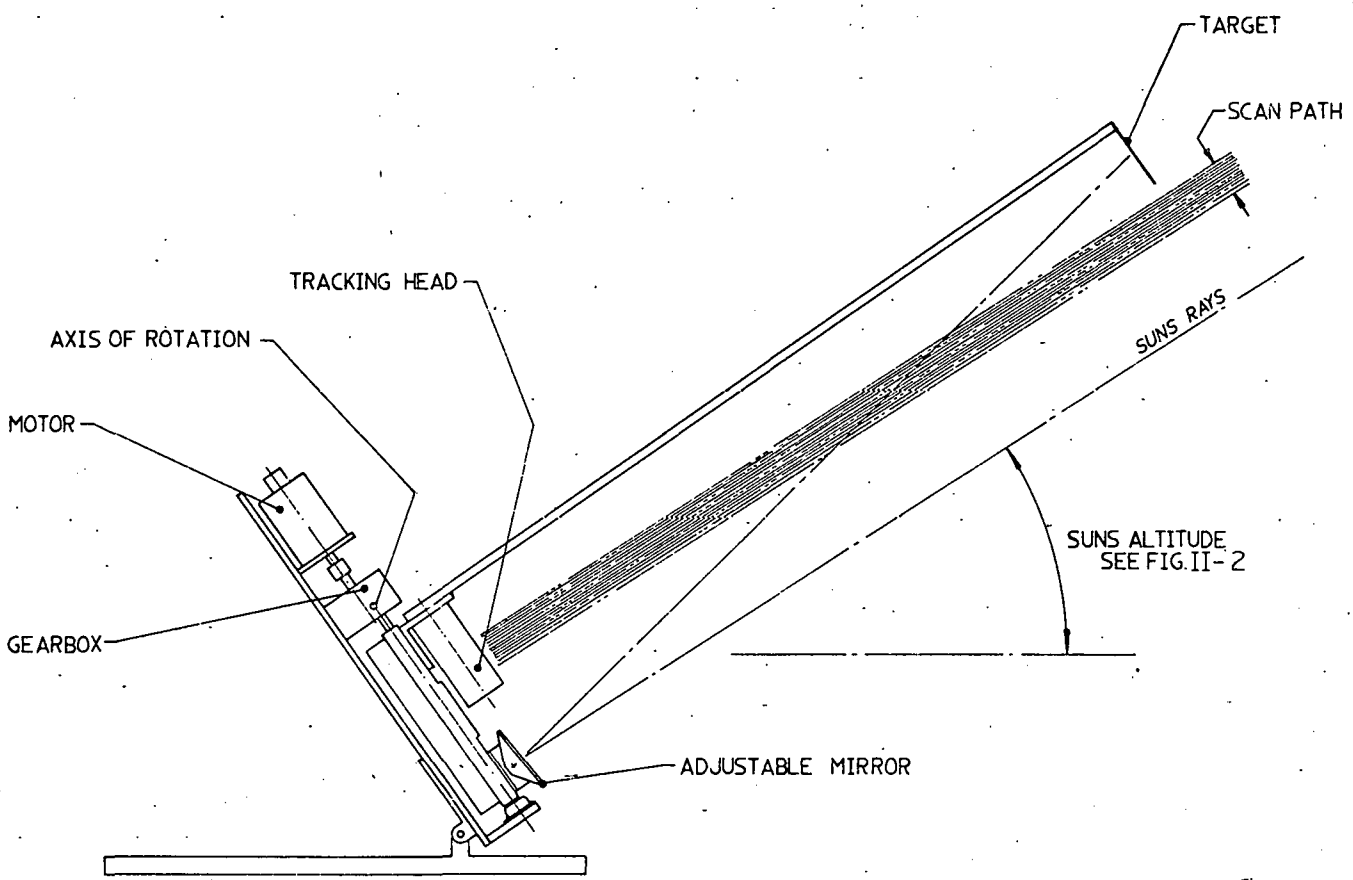
c. Adjustable mirror;

d. Target;

e. And of course the TRACKING HEAD.

The single phase, reversable MOTOR is connected to the GEARBOX by a flexible coupling. The MOTOR and GEARBOX arrangement provide a speed of 0.118 r.p.m. at the GEARBOX output shaft. (approximating the 0.125 r.p.m. speed of the concentrators installed at the Home Laundry in Pasadena).

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



TOLERANCES:
FRACTIONS ± 1/32
DECIMALS .XX ± .010
.XXX ± .001
ANGLES ± 1/2°
UNLESS OTHERWISE SPECIFIED

TITLE TRACKER DEMONSTRATION UNIT

CUSTOMER D.O.E.

NEXT ASM -
USED ON
PROJ NO
REF NO
JOB NO
SCALE

DRN
CKD
APP
DATE

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

DWG NO 107

SHT 1 OF 2

REV A

The ADJUSTABLE MIRROR, TARGET and TRACKING HEAD are mounted on a common shaft and all move simultaneously when the shaft is rotated.

When aligned, the sun's rays are reflected from a 0.06 wide unmasked strip of the mirror to the TARGET 72 inches distant, see fig. II-1. The image at the TARGET is 0.70 wide and is visually compared to a reference scale having 0.125 divisions.

As the sun traverses the sky it's displacement angle relative to the DEMONSTRATION UNIT'S axis is measured and in accordance with a pre-established value the control system repositions the shaft. This repositioning is a incremental rather than a continuous movement which permits the sun's image at the TARGET to be observed moving relative to the stationary reference scale and precise measurements of the tracking accuracy made.
Two limit switches, one signalling "FULL TRAVEL" and the other "STOW" were installed to indicate extremes of travel.

The DEMONSTRATION UNIT could be adjusted to alter the inclination of the TRACKING HEAD so it's field of View could be measured, see drawing no. 108.

The Sun's altitude at 34° N (Noon) is given in Fig. II-2. This was used to determine the correct value for testing.

* * * * *

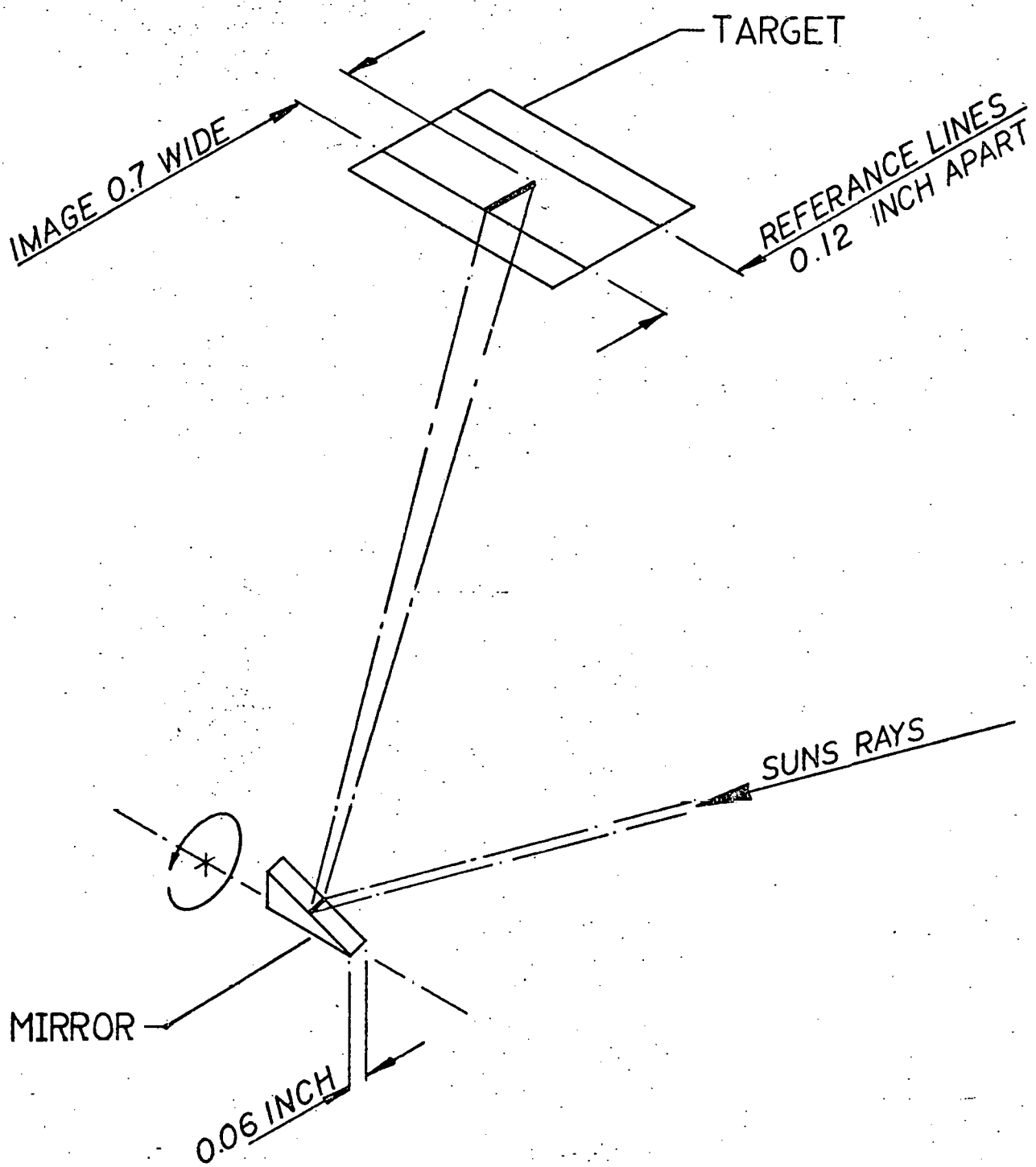


FIG II-1
OPTICS OF DEMONSTRATION UNIT

ALTITUDE	June 21	13	25	37	50	62	74	79
	Mar.-Sept. 21	—	12	25	36	46	53	56
	Dec. 21	—	—	9	18	26	31	33
AZIMUTH	June 21	110	103	95	90	78	58	0
	Mar.-Sept. 21	90	82	72	61	46	26	0
	Dec. 21	—	—	54	43	30	16	0

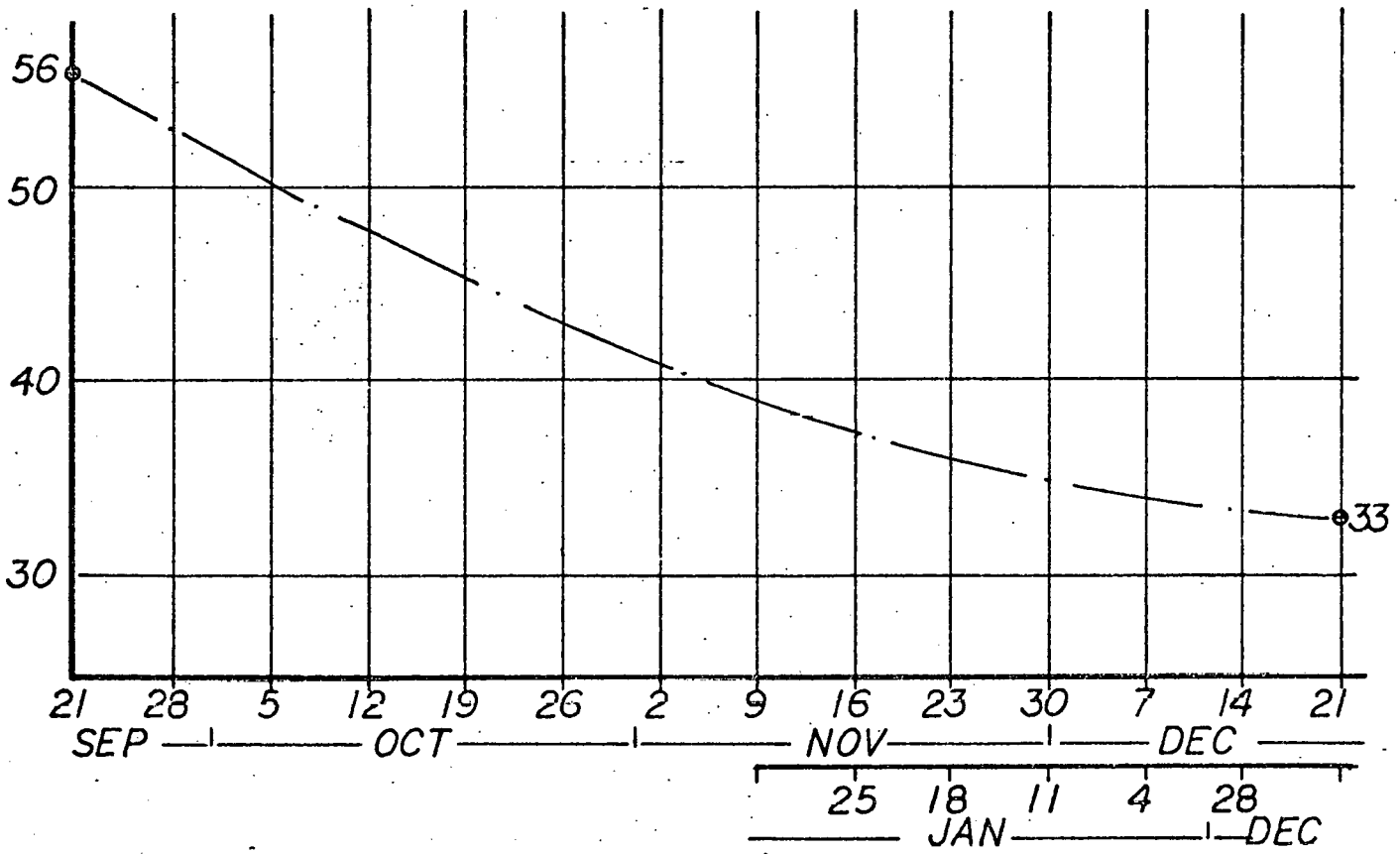
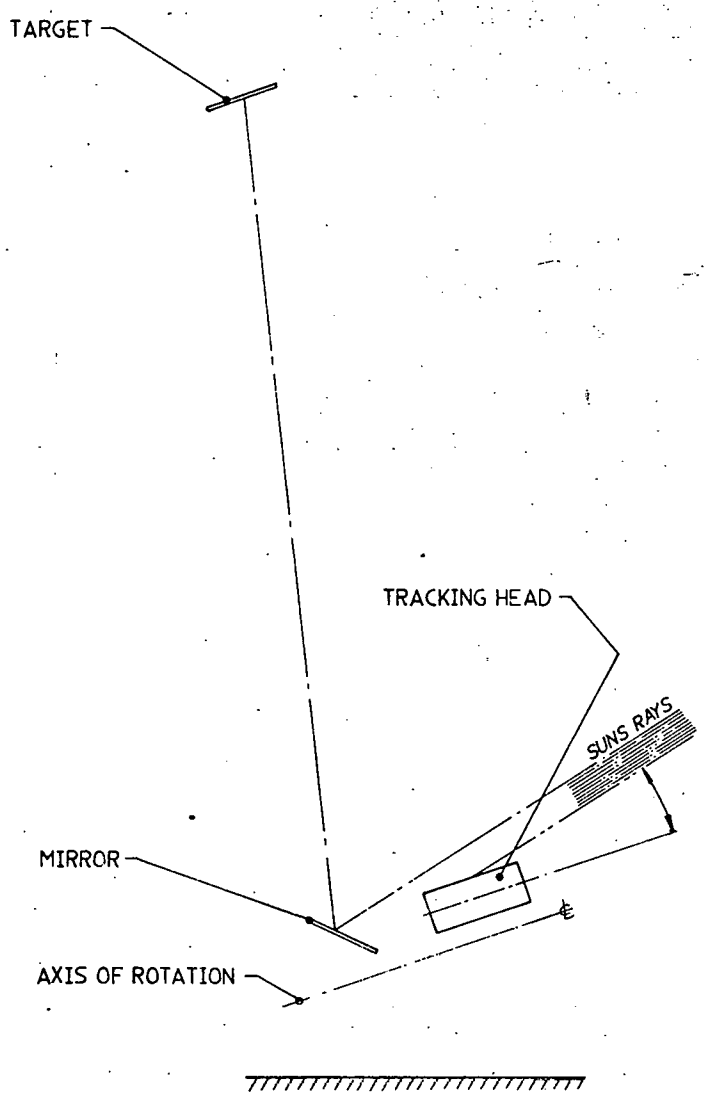


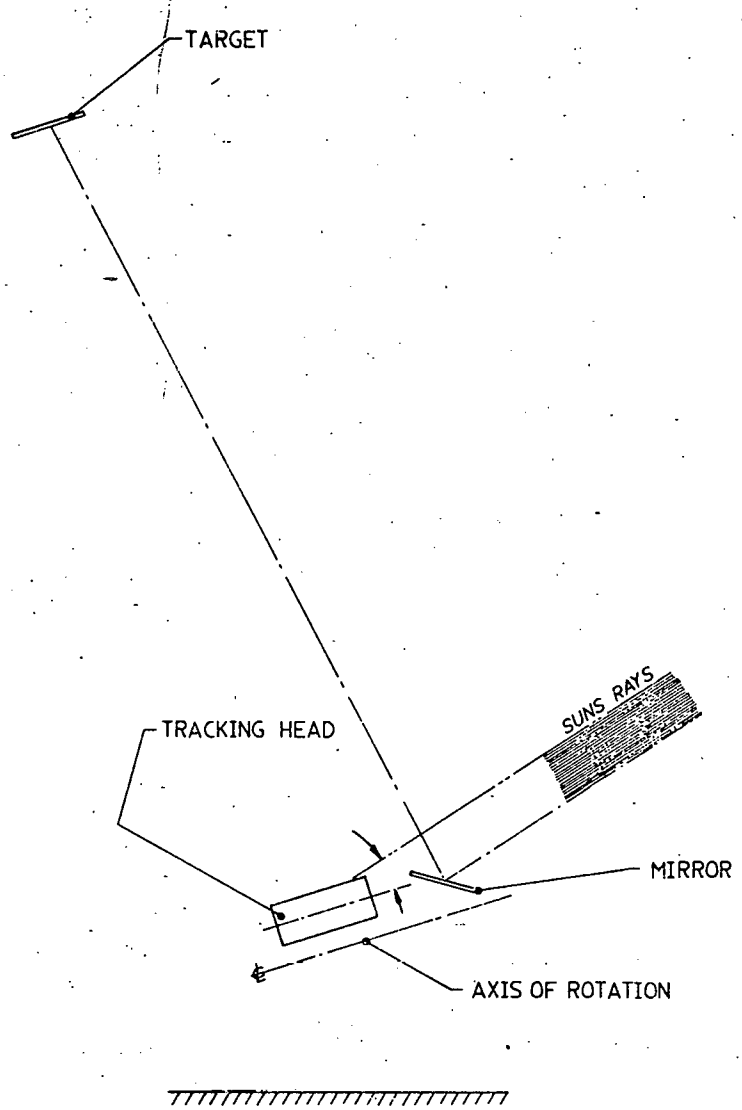
FIG. II-2

SUNS ALTITUDE AT 34 N NOON SEPT -JAN

REV	DESCRIPTION	BY	APP	DATE
A	FIRST ISSUE			



EXTREME FIELD OF VIEW
INBOARD (MOTOR) END



EXTREME FIELD OF VIEW
OUTBOARD END

The information contained herein is the property of GH Systems Inc. This drawing may not be used or copied without written permission. It must be returned upon request.

TOLERANCES: FRACTIONS ± 1/32, DECIMALS ±.010, ±.001, ANGLES ±1/2°, UNLESS OTHERWISE SPECIFIED

TITLE: TRACKER DEMONSTRATION UNIT

CUSTOMER: DOE

NEXT ASM | USED ON | PROJ NO | REF NO

GH Systems, Inc.
WESTLAKE VILLAGE, CALIFORNIA 91361

JOB NO | SCALE | DRN | CKD | APP | DATE | DRG NO 108

SHT 1 OF 1 | REV A