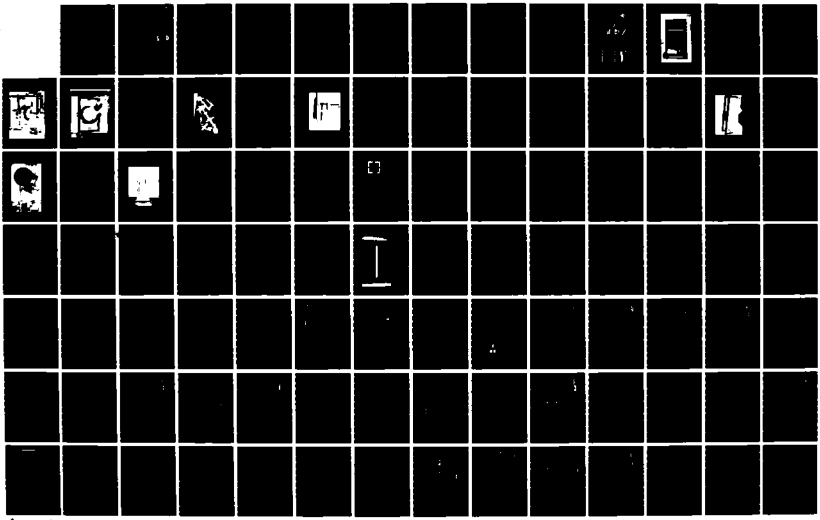
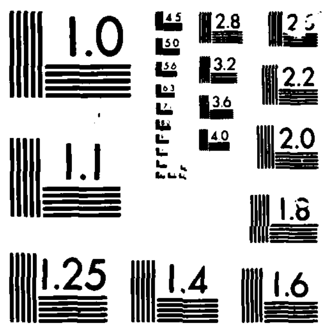
AD-A166 821 RADOME POSITIONER FOR THE RFSS (RADIO FREQUENCY SIMULATION SYSTEM)(U) GEORGIA INST OF TECH ATLANTA ENGINEERING EXPERIMENT STATION D O GALLENTINE ET AL. 1/3

UNCLASSIFIED 27 FEB 78 DAAK40-77-C-0047 F/G 17/9 NL





MICROCOP' CHART

(2)

PROJECT NO. A-1954

BASIC AND APPLIED RESEARCH
SYSTEMS ENGINEERING SUPPORT

SUB-TASK-A-1954-070 & 090 Final Report

RADOME POSITIONER FOR THE RFSS

D. O. Gallentine,
J. A. Stratigos, J. M. Schuchardt
C. J. Bowick, R. W. Bird
31 December 1977

CONTRACT NO. DAAK40-77-C-0047

U. S. Army Missile Research and Development Command
Redstone Arsenal, Alabama 35809

Prepared by the

Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332

Rev: 27 February 1978

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER A-1945-070 & 090	2. GOVT ACCESSION NO. AD-A 166821	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Radome Positioner for the RFSS		5. TYPE OF REPORT & PERIOD COVERED Final Sub Task Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) D. O. Gallentine, J. A. Stratigos, J. M. Schuchardt, C. J. Bowick, R. W. Bird		8. CONTRACT OR GRANT NUMBER(s) DAAK40-77-C-0047
9. PERFORMING ORGANIZATION NAME AND ADDRESS Engineering Experiment Station Georgia Institute of Technology		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Missile Research and Development Command Redstone Arsenal, Alabama 35809 M. M. Hallum (DRDMT-TDF)		12. REPORT DATE 31 December 1977
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Radio Frequency Simulation System RFSS, Radome, Radome Measurements, Gimbal Mechanical Design, Electrical Controls

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the design of a versatile gimbal structure for use in the West Aperture Room of the Army's Radio Frequency Simulation System, for performing RF evaluation measurements of missile radomes and RF seekers. This gimbal is capable of rotating radomes of up to 18 inches in diameter and weighing as much as 50 lbs about fixed RF seeker antennas. The radome motion limit is ± 40° about boresight in both azimuth and elevation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A second element used is a sturdy hinged seeker antenna mount that will permit accurate positioning of the seeker antenna when the radome is mounted or removed from the gimbal. The third element used is a movable cart that can lift the entire assembly off the door frame and carry it out of the way. A fourth element used is a specially designed cart to remove the aperture door and store it while the radome positioner is being used.

The radome positioner is controlled by a microcomputer that permits manual operation and the selection of one of several raster scan patterns to ease data taking. Precision speed and position controlling is accomplished by a closed-loop servo approach where the loop is closed via the computer.

The positioner provides additional capability to perform antenna pattern measurements using the RFSS array and anechoic chamber and may possibly function as a general purpose gimbal permitting closed-loop tracking of the array by experimental RF seekers.

TABLE OF CONTENTS

Pages 120 and 121 do not contain proprietary information.
Per Mr. Kevin Jackson, Army Missile Comd, Systems Simulation Dir.

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 INTRODUCTION

As originally designed, the Radio Frequency Simulation System (RFSS) provides a unique capability for closed-loop testing of RF guidance systems. The elements of the RFSS, notably the multi-element target array and the large anechoic chamber, can service three test point locations - the main central flight table and two off-axis aperture rooms. The two-axis gimbal described in this report is intended to be a portable structure that can be installed when needed in the West Aperture Room* providing the RFSS with an expanded capability for radome and RF seeker testing and for other uses such as microwave antenna testing and RF seeker evaluations.

The new radome positioner shown in Figure 1 utilizes a heavy duty structure capable of handling missile radomes from a variety of Army missiles from the large Pershing to the smaller Patriot and Hawk radomes. Other radomes from a variety of systems can also be handled. The positioner operation is controlled by a microcomputer providing two-axis closed-loop control via a simple keyboard and digital display. These elements are shown in Figure 2. The operator proceeds through a few simple steps to bring the unit to operational readiness and then selects one of several preprogrammed raster scan patterns, or complete manual positioning is possible.

The positioner then moves the radome while the RF seeker antenna remains stationary. This approach is based on the fundamental concept that if only the radome is moving, only the radome is contributing to the indicated boresight error. As a result, even small radome errors can be conveniently and accurately ascertained.

The following sections describe in detail background information associated with the positioner and the mechanical, electrical and software portions of the radome positioner.

---

*Only minor modifications would be required to install the radome positioner in the East Aperture Room.

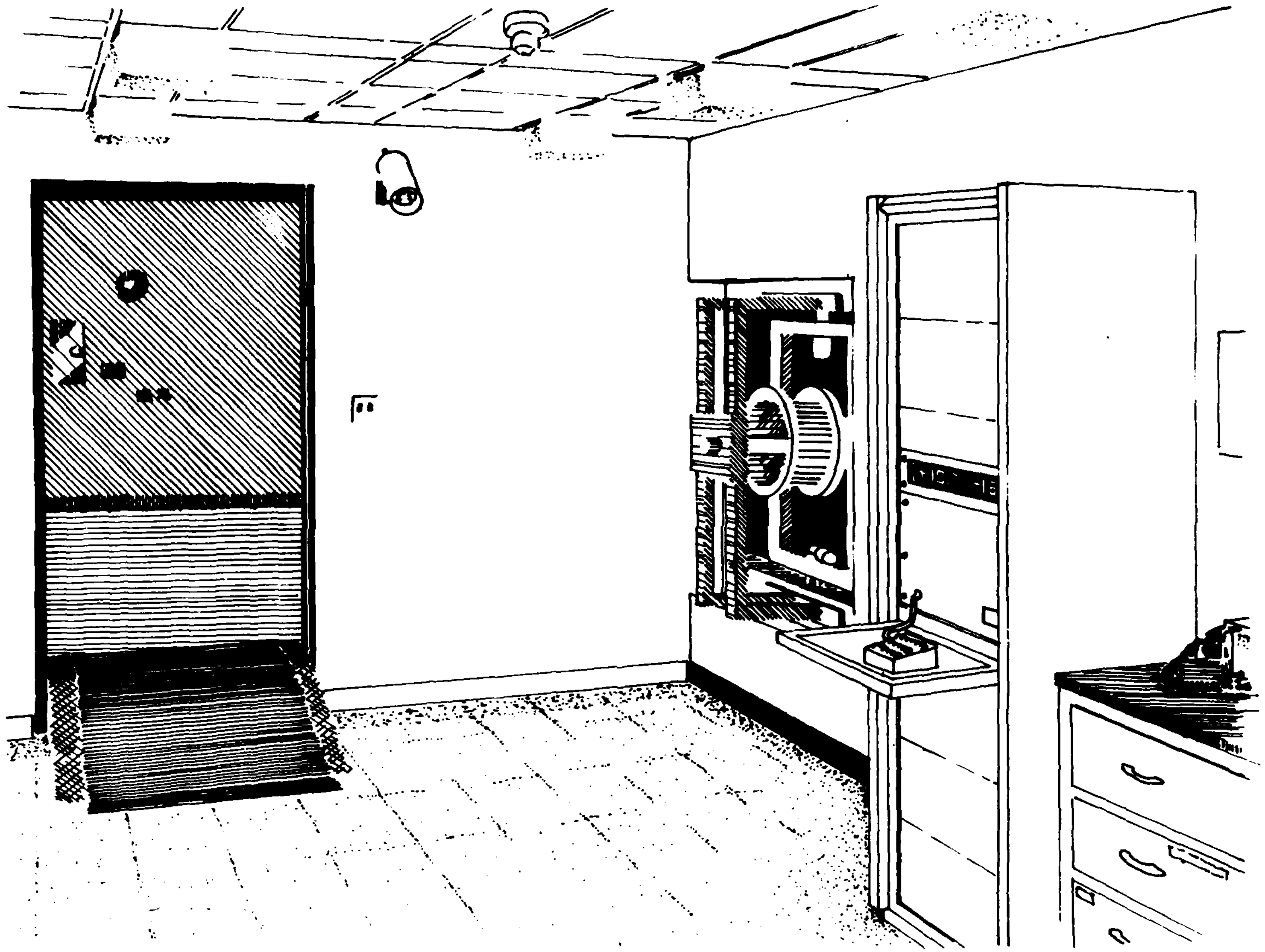


Figure 1. Sketch Showing the Radome Positioner and Electronics Installed in the West Aperture Room of the RFSS.

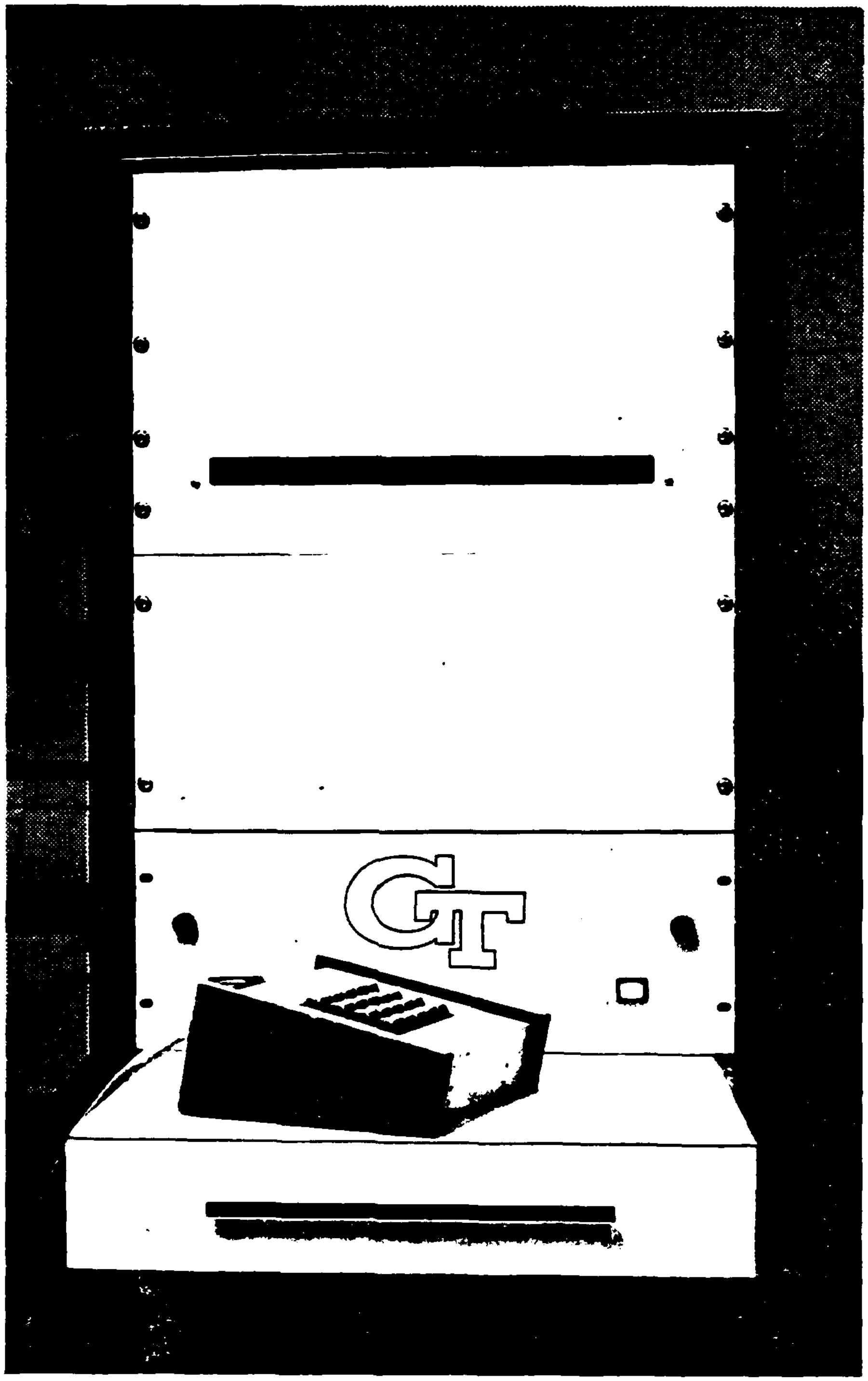


Figure 2. Microcomputer Rack and Control Console for the RFSS Radome Positioner

2.0 BACKGROUND

In the past few years, closed-loop tests of RF seekers in the RFSS with and without radomes have suggested the need for a separate portion of the facility be made available for convenient radome and seeker evaluations that do not tie up the main flight table. The term convenience as used here has several meanings:

1. It is certainly desirable to make things easier for the designer and evaluator to install the bulky and typically awkward missile radome with ease and, above all, accurately and repeatably. This factor improves operator morale and the quality of test data.
2. It is desirable to have a unit that is capable of automatic operation since a large amount of angle space must be sampled to fully characterize the nature of the radome under evaluation and manual operations are not geared to taking a lot of data.
3. A versatile implementation is needed to permit the operator to examine in minute detail peculiarities that might be uncovered in an initial screening of a production or unusual radome specimen.
4. It is desirable for a government agency to have a facility capable of independent evaluation of radomes and RF sensors in a complementary manner to their existing facilities and with known radome test procedures as are used by radome manufacturers.

Thus, the current design described in this report was conceived and developed to meet the needs. Further, the unit as implemented is fully compatible with other Army RFSS computers and offers addition flexibility for testing in the RFSS itself, perhaps allowing improved facility utilization in the future.

3.0 MECHANICAL DESCRIPTION

The radome positioner consists of a large two-axis gimbal assembly designed to fit into the current opening in the west aperture room upon removal of the shielded door. Figures 3 and 4 show the final unit prior to installation. The mechanical design philosophy used to meet the positioner requirements listed in Table 1 was to use large standard size ball and thrust bearings, steel pins compatible with the bearings, aluminum alloys, and standard structural members. Previous experience has shown these methods improve operational reliability and are cost effective.

The inner gimbal is a ring 18 inches inside diameter with a 3-inch square tubular cross-section. This inner gimbal is supported by the outer gimbal through two 0.984 inch diameter steel pins and four single row, deep groove ball bearings. The outer gimbal is a rectangular structure 26 inches by 38.5 inches inside dimensions with a 3-inch square tubular cross-section. It is supported in the azimuth axis with two 0.984 inch diameter steel pins and four single row, deep groove ball bearings. There is also a thrust ball bearing mounted below the outer gimbal to carry the vertical loads of the entire gimbal system. The entire elevation and azimuth gimbal system is mounted to an external frame. This external frame is aluminum angle 4 inches by 3 inches by 0.250 inch thick and is mounted into the opening of the West Aperture Room of the RFSS. The entire gimbal and outer frame assembly is clamped to the internal edge of the aperture opening with eight special design C-clamps (Figure 5). The C-clamps have a large clamping surface to spread the clamping pressure over a broad area, thereby preventing local distortion to the contact finger brass extrusion mounted on the inner edge of the aperture opening.

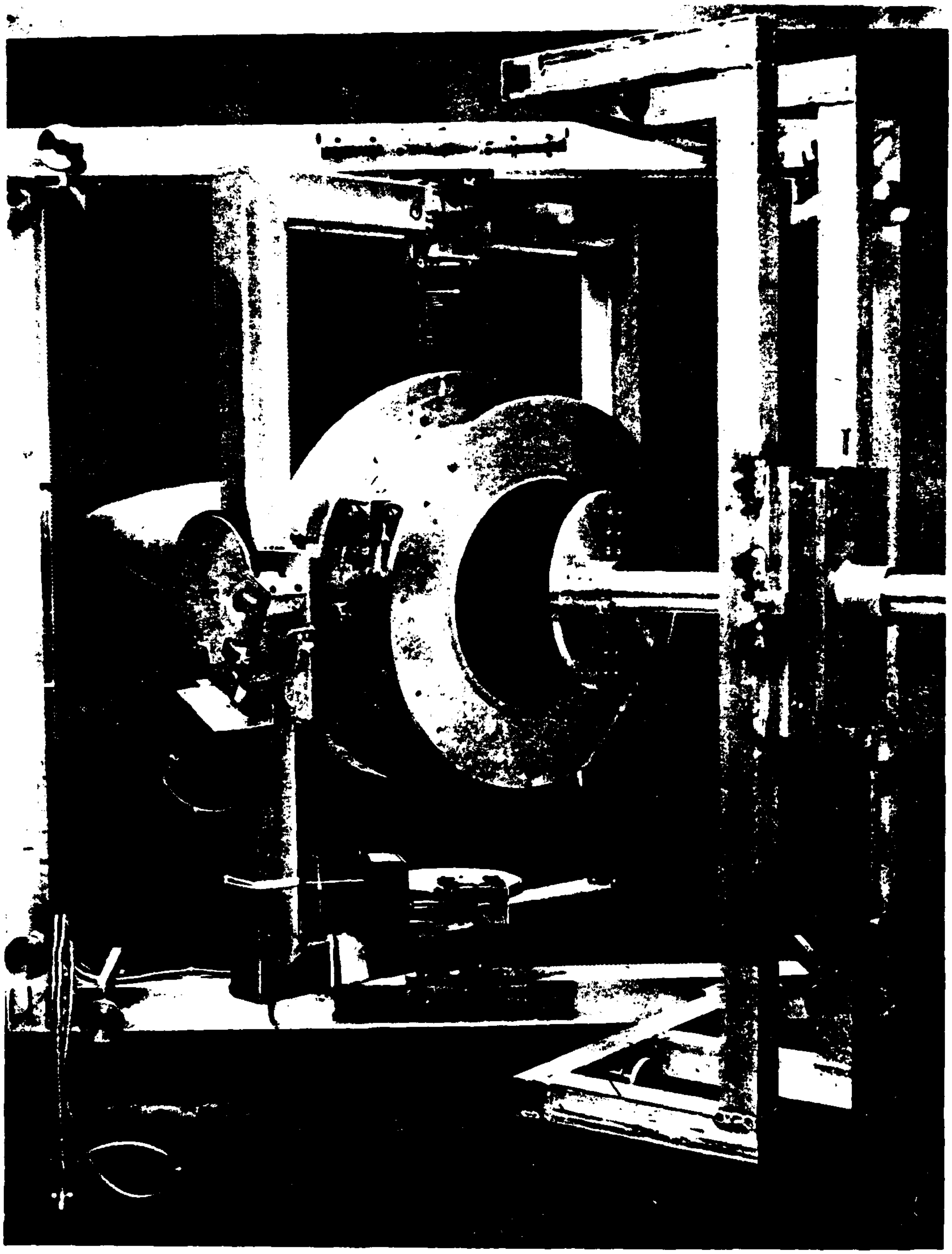


Figure 3. Radome and Gimbals and Seeker Mount Rear View.

Figure 4. Radome and Gimbals Front View.

TABLE 1

RFSS RADOME POSITIONER REQUIREMENTS

Radome Diameter	18 inches, maximum
Radome Weight	50 pounds, maximum
Seeker Antenna Weight	20 pounds, maximum
Readout Accuracy	± 0.1 degrees
Scan Angle Elevation	± 40 degrees
Scan Angle Azimuth	± 40 degrees
Scan Rate (AZ or EL)	2 degrees/sec, nominal
Repositioning Accuracy	± 0.1 degrees
Seeker Repositioning after Boresighting	± 0.005 inches
Sign of Angle (When Looking Toward the RFSS Array)	Up and to the Right (1st Quadrant) +AZ, +EL Up and to the Left (2nd Quadrant) -AZ, +EL Down to the Left (3rd Quadrant) -AZ, -EL Down to the Right (4th Quadrant) +AZ, -EL

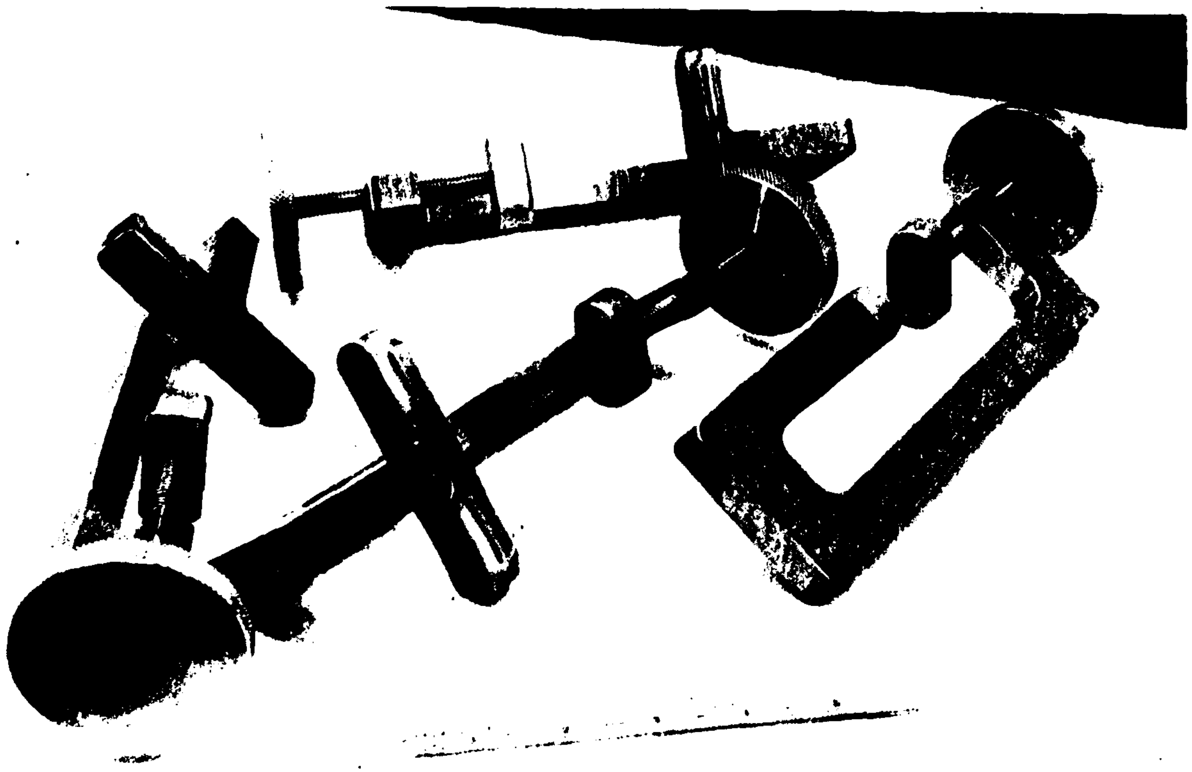


Figure 5. Special C-Clamps for Attaching to RFSS Frame.

The seeker antenna mount and boresight adjustment mechanism is mounted directly to the external frame independent of the radome gimbal system mounting. The seeker antenna is located at the intersection of the gimbal's azimuth and elevation axis. It is mounted on a cantilevered 2-inch diameter aluminum tube supported at the rear-end by a plate which is attached to the boresight adjustment mechanism. This mechanism is an integral part of a stiff tubular structure that is attached to the external frame. Attachment to the external frame is accomplished through two large hand-operated screw locks. There are also four precision steel guide pins with stops located near the screw locks to act as precision references for the repositioning of the seeker antenna and its supporting structure after the test radome is installed. A hinge mechanism is used to swing the seeker antenna and its supporting structure out of the way while the test radome is being installed. This technique allows the seeker antenna to be installed or removed from the inside of the radome.

Associated with the positioner is a separate cart shown in Figure 6 used only for installation, removal and storage of the entire gimbal system. The cart is a modified commercial unit having a hydraulic lift and is mounted on wheels for ease of movement.

3.1 Design Approach

Three basic methods were used to design a gimbal assembly capable of meeting the requirements: 1 structural analysis was performed to select materials and geometries capable of meeting the desired very low deflections that occur as seeker and radome are mounted and moved, 2 basic drive train components were selected to have sufficient inherent accuracy needed to position the gimbal axes

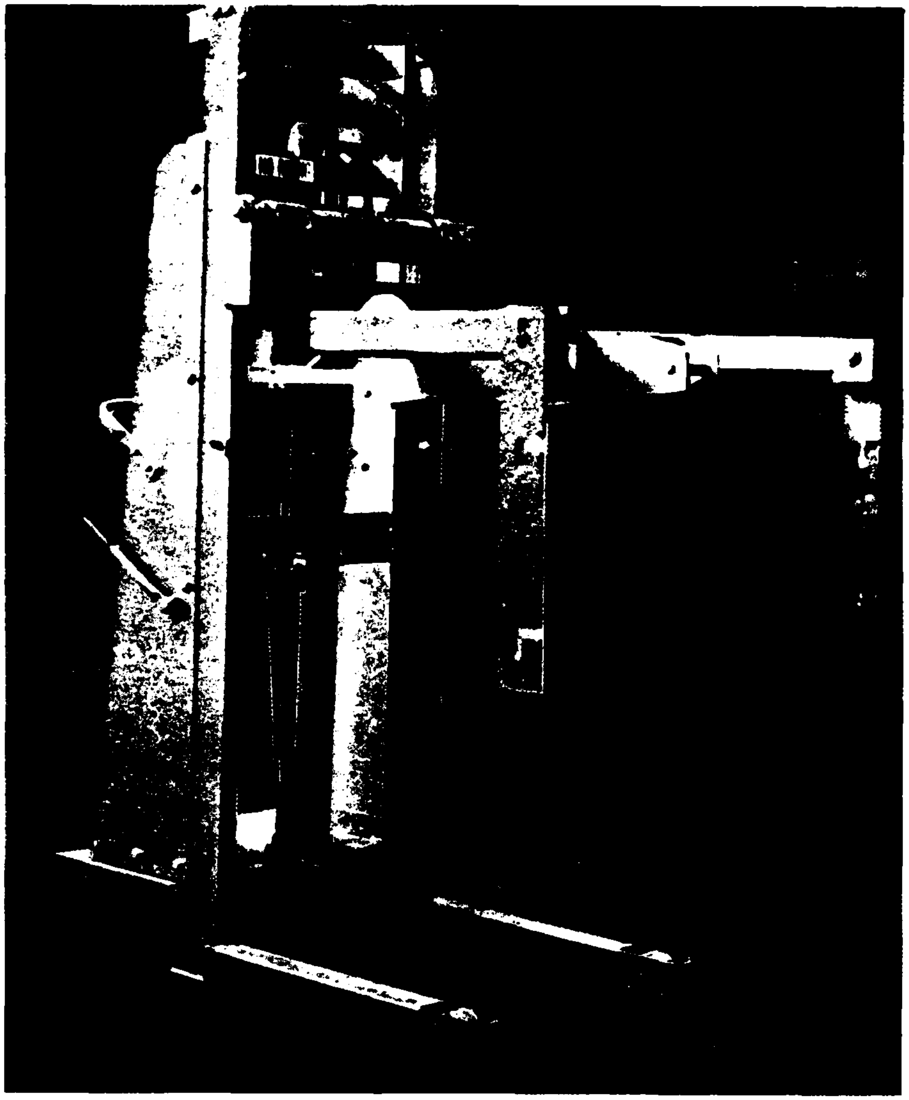


Figure 6 . Cart Used to Install, Remove and Store the RFSS Radome Positioner

accurately, and 3 critical components were accurately machined and when precise alignment was needed stainless steel was used for improved ruggedness.

Structural Analysis

During the design study phase preceeding the hardware fabrication a structural analysis of this approach was completed. This analysis indicates the following:

a] The maximum possible rotation of the outer gimbal frame at the location of the optical encoder is less than 0.00367 degrees. This is based on a maximum external moment of 525 pounds/inch caused by a 50 pound radome with a lever arm of 10.5 inches. This is not a system error, because it can be calibrated out at assembly and periodically thereafter.

b] The maximum possible vertical deflection of the outer gimbal frame with reference to the thrust bearing is less than 8×10^{-4} inches. This deflection is based on the addition of a 50 pound radome.

c] The vibration frequency of the positioner structure is calculated to be a minimum of 170 Hz.

Readout Accuracy

Readout accuracy is a summation of the errors of the various elements of the system; these include the encoders, perpendicularity of the azimuth and elevation axis and the deflections within the structure that are caused by the installation of the test radome after the seeker antenna has been boresighted. A discussion of how each of these error sources is minimized is given below.

The encoders selected for this system are Itek Ral3/23C with an accuracy of plus or minus 0.03 degrees (1/3 bit). The encoder is an absolute type that is connected directly to the shaft in each axis. (The encoder is described in detail in Appendix C.)

The error caused by the perpendicularity of the azimuth and elevation axes is dependent on the error (tolerance) in machining of the two axes in the outer gimbal. A typical machine tolerance would be approximately ± 0.005 inches which would result in an angular error of about 0.02°*.

The error caused by the installation of the radome after the seeker antenna has been boresighted is dependent on the deflection of the entire gimbal system with reference to the seeker antenna mount. In this case, the calculated deflection caused by the installation of a 50 pound radome is 0.0008 inches. The readout error caused by this deflection divided by the distance (48 feet) from the seeker antenna to the source array antenna at the far end of the microwave chamber is negligible (less than 1.5×10^{-3} milliradians).

Other sources of error are an accumulation of miscellaneous machining and assembly tolerances which are estimated to be less than ±0.015° (see Table 2).

Repositioning Accuracy of the Seeker Antenna and Radome

The seeker antenna must be swung out of the way while the test radome is being installed and must be repositioned to its original boresighted position to within ±0.005 inches. This repositioning accuracy is built into the basic structure and is dependent on the machine tolerances of the location precision reference guide pins. This tolerance can be easily held to less than ±0.005 inches by proper machining methods.

The seeker antenna will sag when placed on the 2 inch diameter attachment. The total calculated deflection caused by a 20 pound antenna system is less than 0.001 inches producing an angular rotation of less than 0.5 milliradians in the apparent antenna boresight axis..

The angular location of the test radome must be positioned to within ±0.1 degree. This is accomplished by the use of an accurately machined adapter/fixture that attaches the test radome to the inner gimbal ring. Normal machining methods to tolerances of ± 0.005 inches with well-made radomes will be adequate. Alignment dowel pins are also located on the

---

* $$\tan \theta_{error} = \frac{\text{Machine Tolerance}}{\text{Smallest Width/2}}$$ (See Figure 7)

$$\tan \theta_{error} = \frac{0.005}{24.5/2} = 0.00041$$

$$\theta_{error} = 0.023°$$

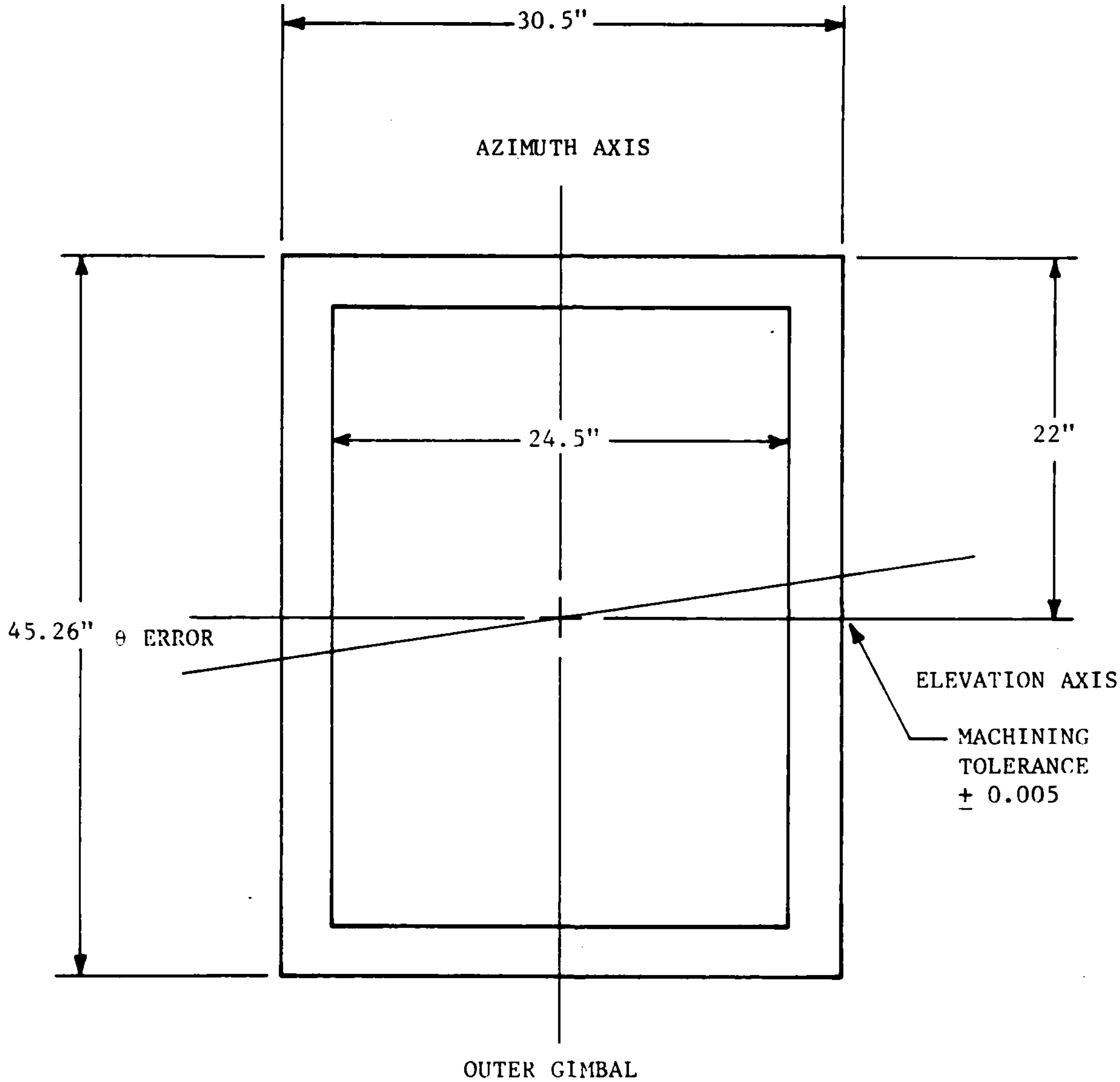


Figure 7. Diagram Illustrating Gimbal Misalignment Factors due to Machining Tolerances.

TABLE 2
TOTAL READOUT ACCURACIES

Error Source	Error Degrees
Encoder Readout	± 0.030
Outer Gimbal/Perpendicularity	± 0.023
Structure Deflection	Negligible (1.5 x 10^{-3} milliradian)
Structure Twisting	Negligible (0.004 degrees)
Miscellaneous Machining & Assembly Tolerances	± 0.015
TOTAL SINGLE PLANE ERROR	± 0.065 peak
	± 0.056 RMS

fixture so that the radome can be removed and re-attached to the inner gimbal ring and maintain the same rotation position relative to the azimuth and elevation axes to within ±0.1 degree.

The weight of radome positioner less radome, seeker antenna and counter weights is:

Inner Gimbal		43.67 pounds
Gimbal Ring	19.01	
Radome Adapter	14.17	
Radome Mount	6.79	
Elevation Shafts	3.70	
Outer Gimbal		84.62
Gimbal Frame	41.59	
Bearings Az and El	5.40	
Azimuth Shafts	3.98	
Drive Motors	10.00	
Gear Trains	7.00	
Encoders	2.00	
Motor Mounting Brackets	4.65	
Miscellaneous Brackets and Clamps	10.00	
Outer Frame		30.83
Outer Frame	23.53	
Alignment Plate	3.20	
Antenna Mount Hinge Brackets	4.10	
Seeker Antenna Mount		50.88
Antenna Mount	48.98	
Dowel Pins, Hand Screw Lock, Hinge Pins	1.90	
Miscellaneous Hardware and Wiring		5.00
	Total Weight Less Radome Seeker Antenna and Counter Weights	215.00 pounds

Note this weight is less than the door (~400 pounds) used to seal the West Aperture Room of the RFSS.

3.2 Mechanical Components

The overall design of the radome positioner system is based on a two axis system (elevation over azimuth) mounted in an aluminum angular outer frame designed to closely fit the aperture door opening. A rigid seeker antenna mounting frame shown in Figure 8 is also attached directly to the angular outer frame.

The outer mounting frame (See Drawing 5*) is fabricated from 4" by 4" by 3/8" aluminum angle-alloy 6061T6. The angle is welded into a rectangular frame and machined to fit the aperture opening, allowing a nominal 1/16" clearance on all four sides. During the machining operations the mounting surfaces for the azimuth upper and lower gimbal shaft (See Encoder Mounting Shaft Drawing 25) are machined parallel to each other to within 0.002". Also, the surfaces for mounting the seeker antenna alignment plates (See Drawing 6) and the seeker antenna support bracket hinges are machined perpendicular to the gimbal shaft mounting surfaces to within 0.002" and parallel to each other to within 0.002". Holes are also provided for mounting the four lifting buttons (See Drawing 28) on the sides.

The outer gimbal (Drawing 8) is fabricated from 3" readily available square aluminum of alloy 6061-T6. Solid aluminum blocks are also incorporated into the overall weldment at the azimuth and elevation bearing axis to provide solid cross-sections sufficiently large to support the large azimuth and elevation bearings. During the machining operation, the perpendicularity of the two axes was maintained to within ± 0.005". The main drive motor mounting brackets in both axes are also mounted on the outer gimbal. The mounting surfaces for these brackets are machined on the sides of the gimbal to be parallel to the appropriate axis within ± 0.005". The encoder mounting surfaces are also located on the outer gimbal. These encoder mounting surfaces are machined to locate encoders concentric and perpendicular to the center lines of the azimuth and elevation axis to within 0.001 inches to prevent excessive side loading on the encoder bearings.

*All mechanical and electrical drawings appear in Appendix A.

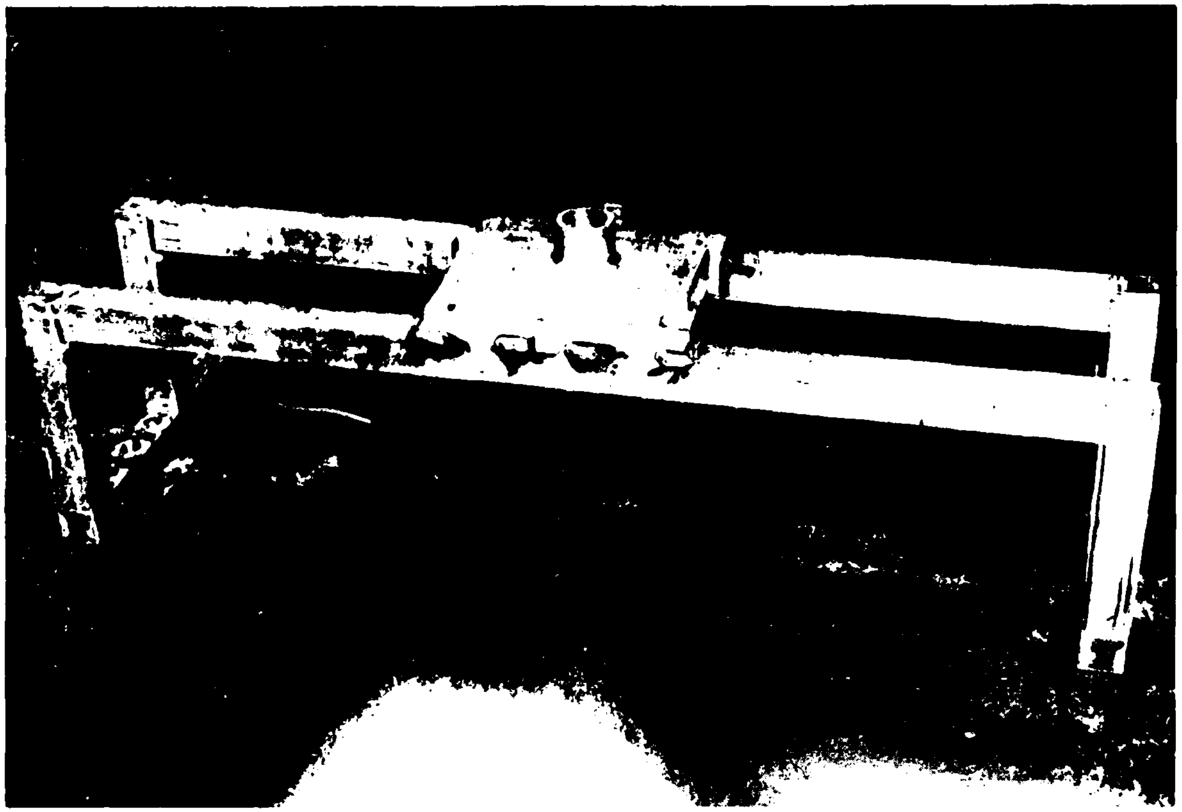


Figure 8. Seeker Antenna Mounting Frame Prior to Installation

The inner gimbal (See Drawing 9) is a total weldment fabricated from various thicknesses of aluminum sheet and solid aluminum blocks. Excess stock is left on all surfaces of the weldment to allow machining to final dimensions. General tolerances are held to ± 0.005 inches although clearance between the radome mounting ring (See Drawing 27) and the inner gimbal is held to less than 0.002 inches by hand fitting during the machining operations.

The seeker antenna bracket (See Drawing 11) is a machined weldment fabricated from standard 2 inch by 2 inch by 1/8 inch square aluminum tubing alloy 6061-T6. Solid aluminum sections are added where compression loading is required to prevent localized distortions of the tubing wall. Parallel tolerances of mounting surfaces are held to 0.002 inches. The location of main alignment dowel pin holes is accomplished by transferring the holes to the antenna alignment plate (See Drawing 6) after attachment to the outer frame. The seeker adjustment mechanism (See Drawings 22, 23, and 24) attaches directly to the seeker antenna mount and provides vertical and horizontal adjustments in both planes up to ± 0.750 inches. Adjustments in the third plane are accomplished with the 2 inch diameter round tube located at the center of the adjustment mechanism.

Drive motors and worm gear assemblies shown in Figure 9 are identical in both azimuth and elevation axis. There is an additional thrust bearing located in the worm gear mounting block (See Drawing 15) for the elevation axis. This thrust bearing is located above the worm to accept the loads of the radome when insufficient counter weight is applied.

The counter weights (See Drawing 41) are located on each side of the radome mounting. Eight counter weights are supplied and can be applied in increments of 10 pounds up to 80 pounds.

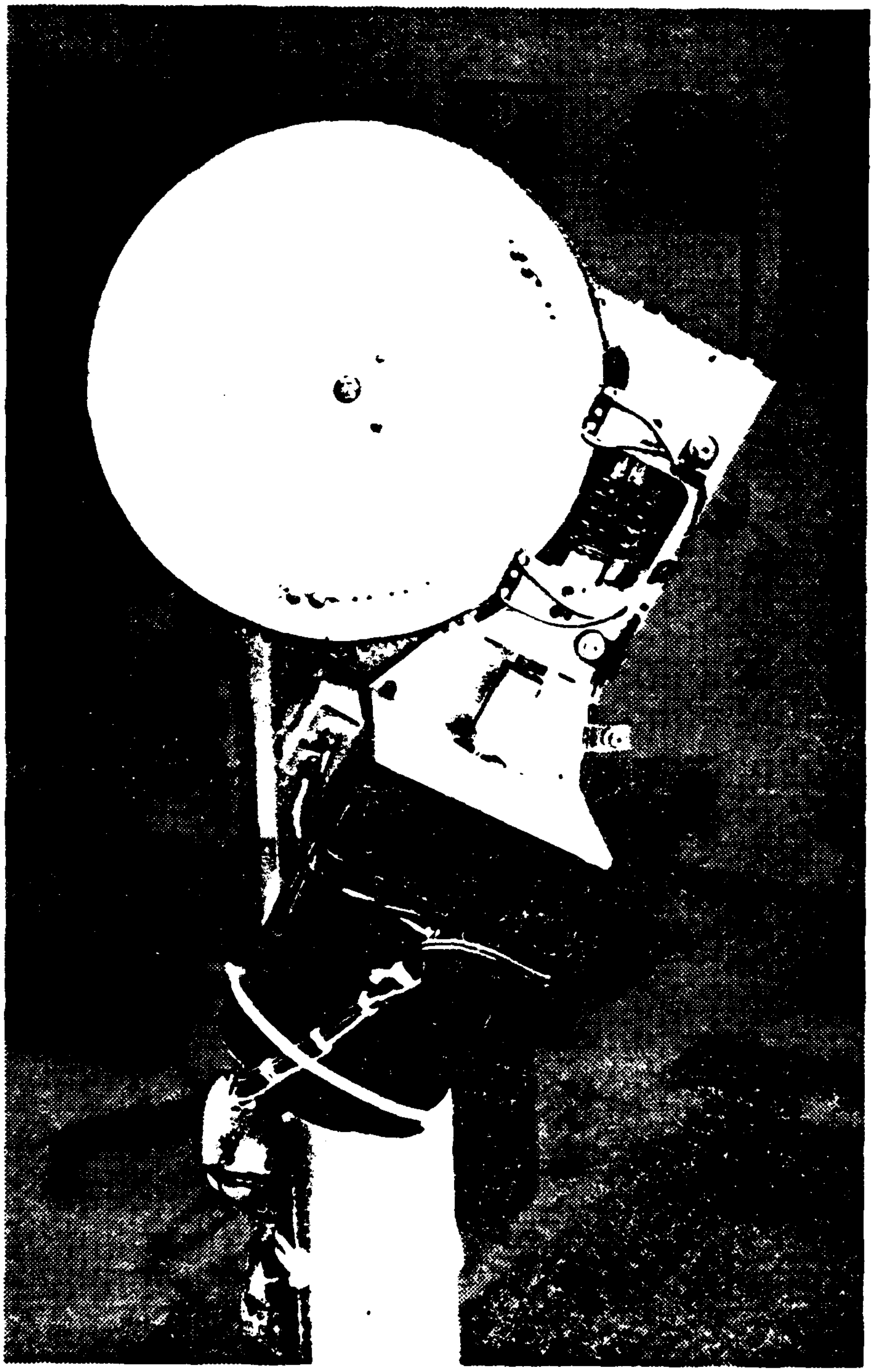


Figure 9 . Elevation Axis Motor and Worm Gear Drive Assembly.

3.3 Installation Procedures

In order to install the Radome Positioner in the aperture door opening, the aperture door and its associated parts including door latch, limit stops, etc., must first be removed and stored for future replacement. The door removal is accomplished with a special cart shown in Figure 10 that is fabricated for that purpose.

The suggested procedure for removal of the door and installation of the Radome Positioner is:

A. Remove door closure mechanism from both the door and the wall above the door. Also remove the door latch and the electrical door-closed indicator switch from the wall on the right side of the door. (Save all of the hardware for reinstallation of the door.) Open the door to approximately 45° angle.

B. Manually move the door-removal hand cart under the door until the approximate center of gravity of the door is aligned with the approximate center of the platform on the cart. Align the edge of the door with groove on the top of the platform. With the foot lever raise the platform until the weight of the door is being supported by the cart. Tighten the upper door clamp. With the foot lever raise the door until the door hinges are separated (approximately 2 inches). Move the cart, with the door attached, away from the door opening until the door clears the wall. Slowly lower the platform with the door attached by releasing the hydraulic valve. The door and cart can now be moved for storage.

C. Manually move the Radome Positioner cart, with the Radome Positioner attached, into position in front of the aperture opening. Raise the Radome Positioner until the outer frame

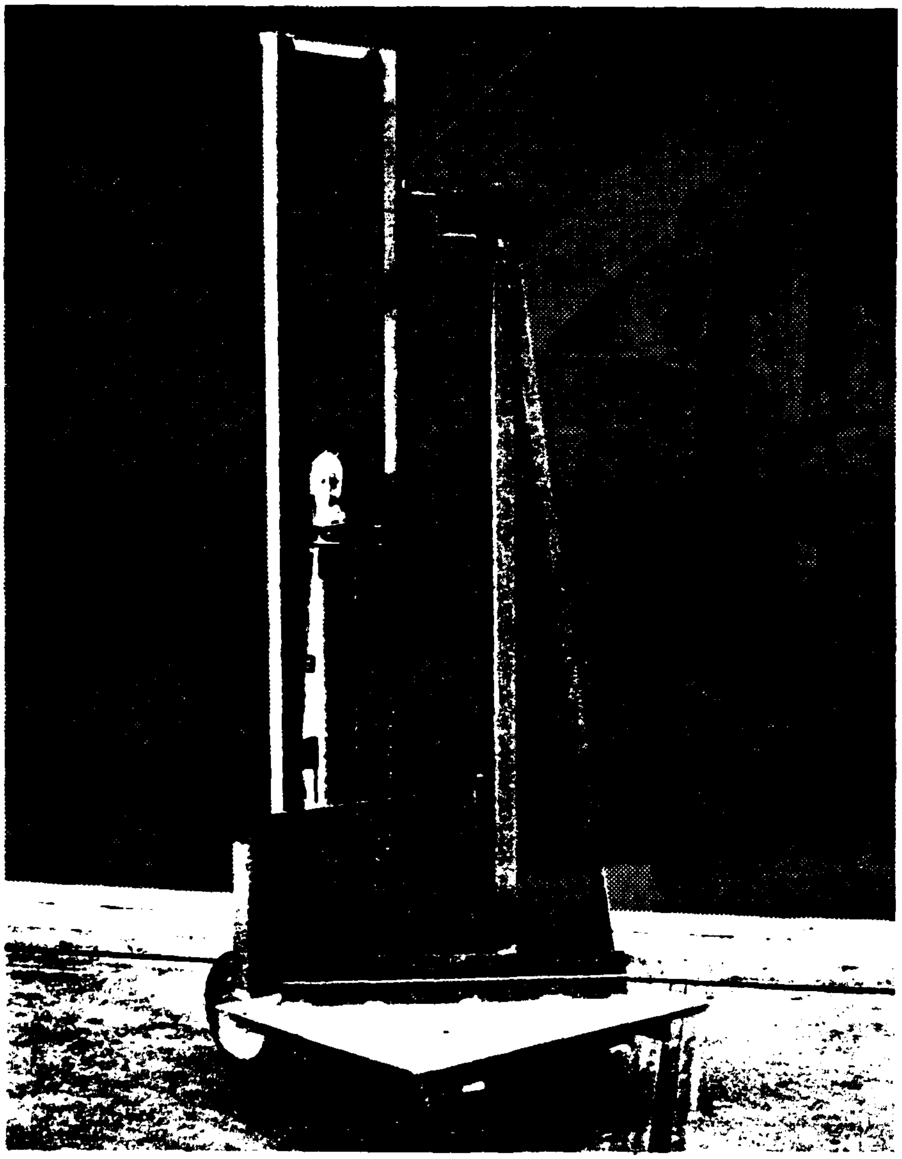


Figure 10. Cart Used to Remove the RFSS West Aperture Room Door.

aligns with the aperture opening. Lock the foot brake on the cart and move the Radome Positioner forward into the opening by rotating the hand wheel located between the forks at the cart.

D. Install and tighten the eight special C-clamps around the periphery of the outer frame.

E. Remove the cart by lowering the forks with the lifting hooks approximately 1.5 inches. Release the foot brake and move the cart backward separating the lifting hooks from the lifting buttons located on the outer frame.

F. After completing the electrical connections to the drive motor and readout devices, the system is ready for the electrical check-out and operation.

G. Calibration adjustments and zero alignment of the optical encoders can also then be made.

To reinstall the aperture room door, the procedure given above is carried-out essentially in reverse.

4.0 ELECTRONICS

The basic requirement of the radome positioner electronics is to allow an operator to accurately position the azimuth and elevation gimbals. A simple solution would be a manual control of the motors for each axis and a calibrated scale to indicate angle. A more sophisticated system eases the operator's task permitting the taking of a lot of data from many radomes. Until recently the conventional approach to the problem involved a complex servo feedback system and electro-mechanical indicators and position control. With the introduction of the microprocessor in 1971, the means to implement a programmable and sophisticated control system with a minimum of hardware became readily available. In addition, a microprocessor is capable of performing a wide variety of complex logical operations under control of an easily modified program stored in digital memory.

The use of a microprocessor and its associated components forming a microcomputer, as the basis of the radome positioner electronics, results in a very flexible and easy to use system. To implement a similar system with conventional digital logic would require 400 - 500 integrated circuits compared to the 30 integrated circuits which actually comprise the heart of the microcomputer. This dramatic reduction in parts count results in a similar reduction in cost and power consumption and an increase in reliability and flexibility.

A block diagram of the radome positioner electronics is shown in Figure 11. The microcomputer closes a digital control loop between each shaft angle encoder and the azimuth or elevation motor and displays the current gimbal position. Commands from the keyboard or serial ASCII (American Standard Code for Information Interchange) data from an external source cause the microcomputer to update angle inputs to the control loop and to open or close the loop as required. Internally stored programmable raster patterns are provided for automatic positioning.

With the exception of the display and motor controller interface, all of the positioner electronics are contained on 5 plug-in circuit boards located in the main electronics chassis. Four of these boards, the enclosure,

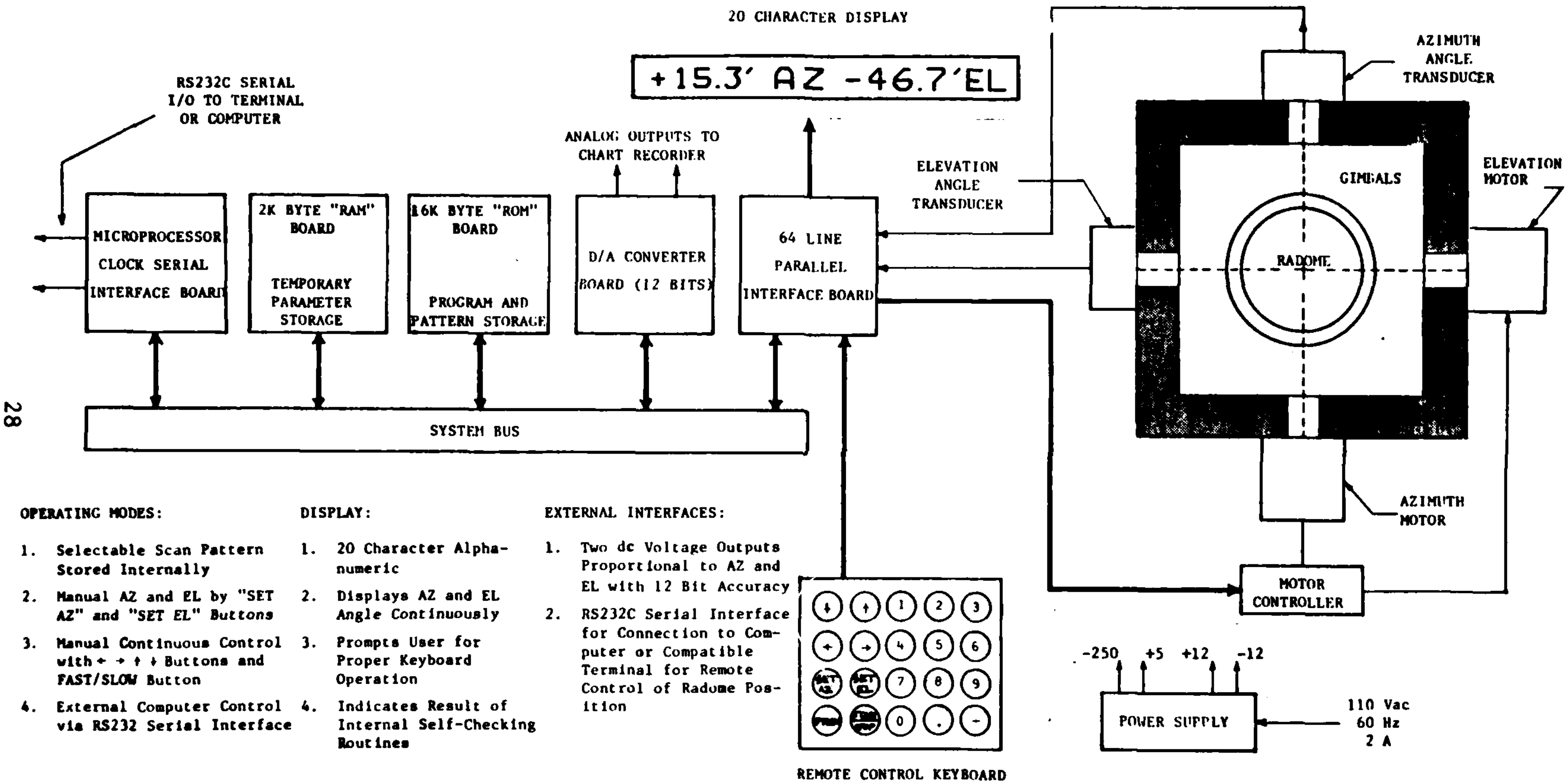


Figure 11. Block Diagram of RFSS Radome Positioner.

and power supply are commercially available microcomputer and peripheral circuit cards that have been modified slightly for this particular application. A fifth wire-wrap board in the main chassis contains miscellaneous interface circuits for the keyboard, display and analog angle outputs. The only other non-standard electronics assembly is the dual, optically isolated D/A converter and amplifiers which are used to convert digital information from the microcomputer into isolated dc signals used to control the speed and direction of the gimbal motors. This is referred to as the motor controller and is contained in an enclosure mounted behind a rack panel in the bottom of the rack.

4.1 Electronics Components

The positioner electronics are comprised of several distinct components, each of which will be described individually. These components are: microprocessor board, RAM board, ROM board, parallel I/O board, I/O buffer circuits, display, keyboard, power supplies, shaft encoders, motors, and motor controller.

Microprocessor Board

The heart of the entire controller is a Motorola M6800 micropressor and its associated circuits contained on the Motorola M68MM01A-1 micromodule (See Drawing 64). This board contains the 6800 microprocessor. 1k bytes of RAM, provisions for up to 4k bytes of ROM, two 20 bit parallel I/O ports and an RS232 serial I/O port. The controller software is contained in the four 2708 EPROMS located on this board. The two parallel I/O ports are used to operate the display and keyboard.

2k Static RAM

The Motorola M68MM06 board holds 2048 bytes of RAM that are used for temporary variable storage by the positioner software. (See Drawing 66) Sixteen 2102-1 1k bit static RAMs and their associated buffering and address decoding logic are contained on this board.

16k ROM Board

As additions are made to the software and if new rasters are added the 4k bytes of ROM space available on the microprocessor board may not be adequate. The 16k ROM board M68MM04 (See Drawing 67) has room for 16 1k byte EPROMS should additional memory be required.

32/32 Parallel I/O Board

The M68MM0 3-1 32/32 parallel I/O board (See Drawing 68) provides 32 parallel inputs and 32 parallel outputs to the microprocessor. These inputs and outputs appear as four memory locations beginning at address 8E00 and are used for motor speed and direction control and to read the two 13 bit shaft angle encoders. All inputs and outputs are buffered.

I/O Buffer Board

This board is a custom wire-wrapped board containing miscellaneous interface circuitry for the keyboard and display as well as a modular dc/dc converter and two 12 bit D/A converters used to provide analog azimuth and elevation angle outputs. A schematic and parts placement for these circuits are shown in Drawings 67 and 70.

Display

A Burroughs Self-Scan II 20 character alphanumeric display (See Appendix C) is used to output messages and position information to the operator. This display was chosen because it is entirely self-contained and required only power supply voltages and parallel ASCII data. The display requires +5 Vdc, -12 Vdc and -250 Vdc to operate. The -250 Vdc is provided by a modular dc/dc converter mounted on the rear inside wall of the main chassis. Characters in the display are multiplexed and thus require a periodic "refresh". This is provided by an external 100 ns clock on the I/O buffer board and appropriate interrupt driven software (See Section 4.3).

Keyboard

Commands to the positioner are entered via a custom 20-key keyboard attached to the front of the main chassis (See Drawing 71). Hall-effect switches were used to minimize key bounce problems. Integrated circuit U1 generates an interrupt to the microprocessor whenever a key is pressed causing the positioner software to execute a keyboard parsing routine which reads the switch closure via PIA lines PAØ-PA7 and PBØ-PB7 from the I/O buffer board.

Power Supplies

A power supply (See Drawing 75) within the main electronic chassis supplies +5 Vdc, and ±12 Vdc to the microcomputer components, keyboard, display and motor control electronics. A dc/dc converter located on the I/O buffer board converts 5 Vdc to ±15 Vdc for the D/A converters used to produce analog angle outputs and to the D/A converters in the motor speed controller. *Another dc/dc converter in the rear of the main electronic chassis supplies* 250 Vdc to the display.

Shaft Encoders

Two identical 13 bit shaft angle encoders located on the azimuth and elevation gimbal shafts converts angular displacement to parallel binary data used by the microcomputer to read the gimbal angles to within 0.09 degrees. +5 Vdc is supplied to each encoder via the 50 conductor cable connecting the encoders and microcomputer (See Appendix C for a complete description of the shaft angle encoders).

Motors

Identical 1/25 horsepower ac motors (See Figure 9) and gear reduction boxes are used to drive each gimbal axis. Motor speed is regulated by a triac speed controller which in turn is driven by optically isolated control signals from the motor controller. A tachometer on the motor shaft provides an ac voltage proportional to speed which is fed back to the motor speed controller to maintain constant torque under varying load conditions over a wide range of speeds.

4.2 Motor Control Electronics

A block diagram of the electronics utilized for motor speed and direction control (azimuth and elevation) is shown in Figure 12. The electronics perform three basic functions. They are:

1. Control of the rotational speed of the motor on each axis.
2. Directional control for each motor.
3. Manual over-ride control.

The heart of the motor control electronics is the Q-CON C-10P-4 variable speed motor control. A block diagram of this motor controller is shown in Figure 13.

The controller maintains constant speed over a wide range of load torque by means of a feedback loop between the tachometer and a triac operated motor driver. The amplifier amplifies the error signal received from the tachometer in the feedback loop and drives the motor. The effective gain in the loop can be adjusted to be sufficiently high so that even a small error voltage will initiate a corrective action. The effective gain in the loop when using the C-10P-4 controller approaches 100 causing the speed of the motor to change a few percent with changes in the load.

Separate C-10P-4 motor speed controllers are used for the azimuth and elevation axes motors. The only modification made to the controller was to bring out a connection to allow motor speed to be controlled (V_1) externally by a D/A converter which is driven by an output of the micro-computer.

Details of the controller are discussed below.

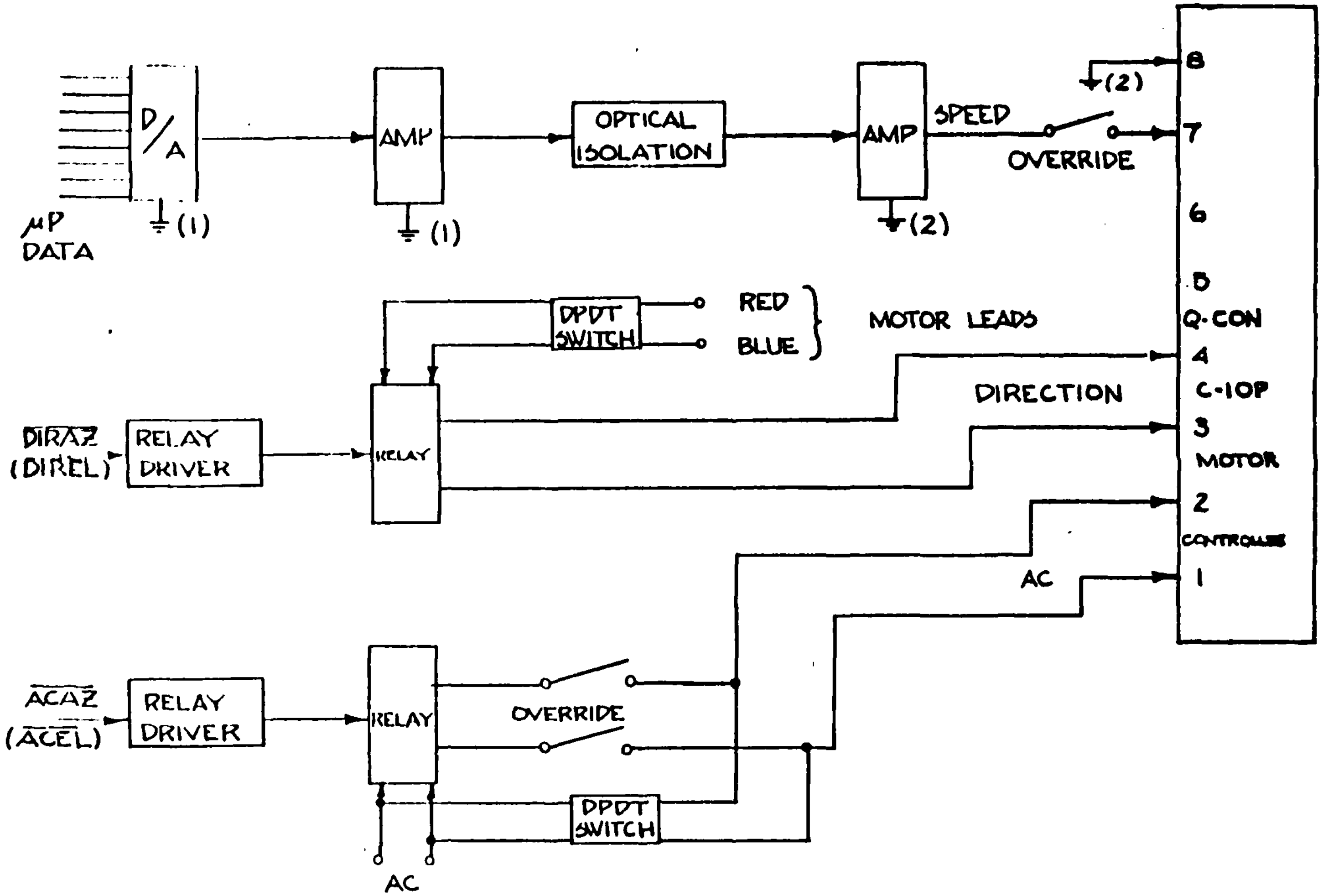


Figure 12. Motor Direction and Speed Control Block Diagram.

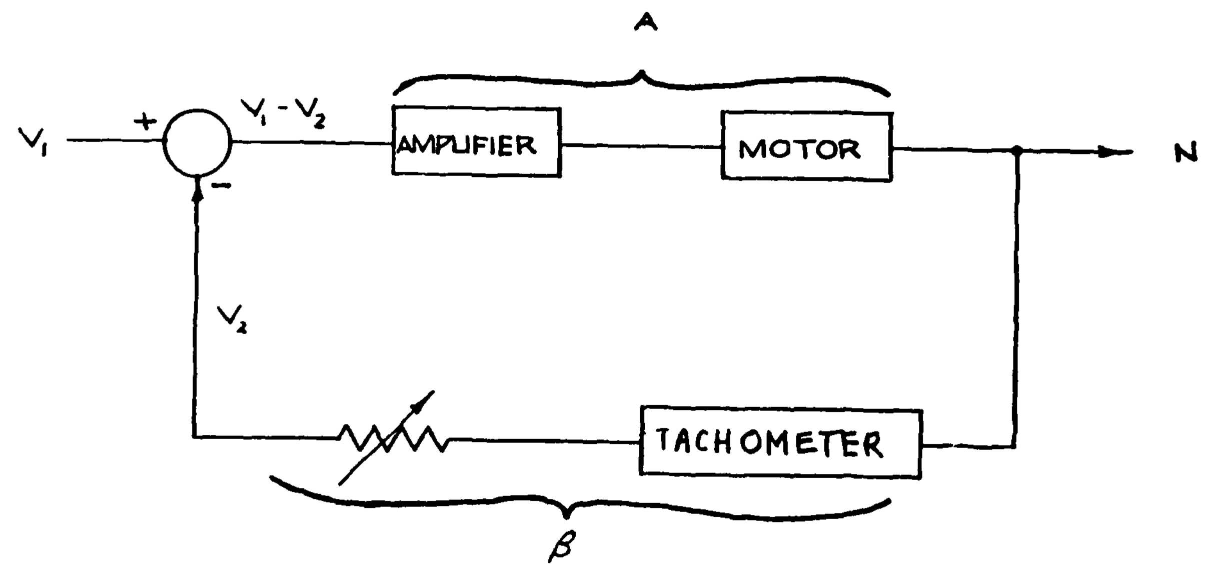


Figure 13. Motor Controller Feedback Loop.

Control of Motor Speed

Control of the rotational speed for each motor (azimuth and elevation) is accomplished by applying positive dc voltage reference (V_1) to pin 7 of the respective Q-CON C-10P-4 controller (See Figure 14). The motor's rotational speed is proportional to the reference voltage V_1, and at a reference voltage of zero (0) volts, each motor will be stopped. Maximum rotational speed is obtained with an input voltage of approximately 20 volts.

When the microcomputer issues a speed command to a motor (azimuth or elevation) it delivers an 8-bit digital word to its respective digital to analog D/A converter (Figure 12). The D/A then translates this digital word into a corresponding dc voltage level. This dc level is then amplified by U1 and applied to an optically coupled isolator through U2. The isolator's function is to isolate the C-10P-4 controller from the microcomputer and its associated components, thus providing protection against ground loops, power line transients, and noise. The isolated dc signal is then amplified by U2 and U3 and applied as the reference voltage (V_1) to pin 7 of the C-10P-4 controller.

The D/A used in this design is the Datel model 98BIR. Its 8 inputs provide for 2^8 or 256 different dc voltage levels which are used to control the motor speed.

Directional Control

Directional Control of the rotation of each motor is accomplished by:

1. Disconnecting ac power from the C-10P-4 controller.

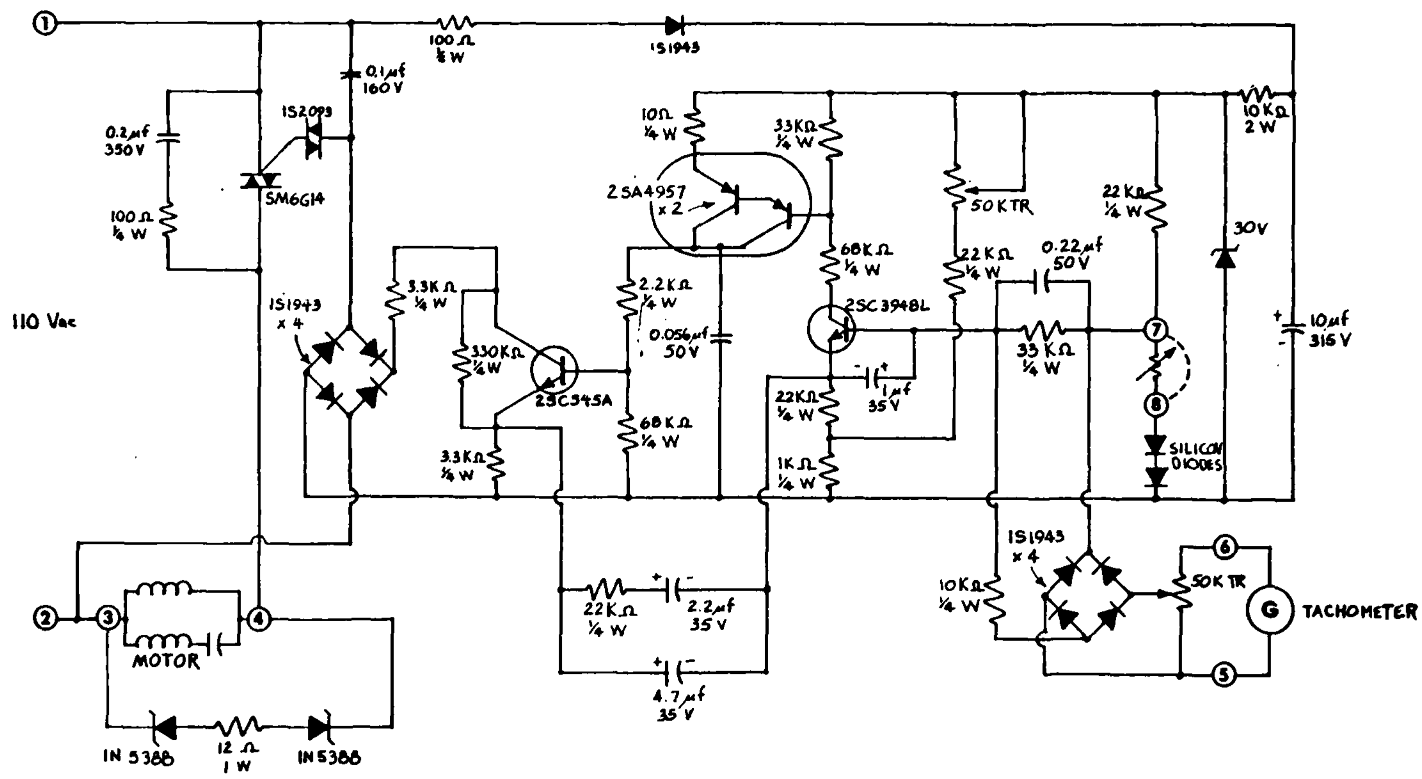


Figure 14. Q-Con C-10P-4 Motor Speed Controller.

2. Reversing the red and blue motor leads between pins 3 and 4 of the controller.

3. Reapplying the ac power.

The sequence must be followed as it is stated above. MOTOR DIRECTION MUST NOT BE CHANGED WITH POWER APPLIED TO THE CONTROLLER.

During program control, if the microcomputer determines the need for direction reversal, it first sets ACEZ (ACEL) high turning off relay RY1 (RY2) and disconnecting power from the controller. Once ac power is disconnected, the motor's direction may be reversed. To do this the microcomputer complements DIRAZ (DIREL) reversing the red and blue motor leads between pins 3 and 4 of the C-10P-4 controller. Power is then re-applied to the controller by the microcomputer, and the direction reversal is complete. This is all done through program control and takes place automatically.

Manual Over-ride Control

During its normal operation, the microcomputer software will not allow the radome to be positioned at an angle greater than ± 40 degrees with respect to its center position in either azimuth or elevation. Should the microcomputer fail however, a standby method of shutting down the system, if it tries to rotate beyond its ± 40 degree limit, has been installed. This standby system is a set of four limit switches, two for each axis, which will interrupt the ac power to the C-10P-4 controllers if ± 40 degrees in either direction is reached by the radome. If this occurs, there is no way of returning the radome to within its boundaries through microcomputer control. For this reason, manually operated override switches have been provided to reposition the radome to within its boundaries so the microcomputer can regain control. THE OVERRIDE SWITCHES SHOULD NEVER BE OPERATED WHILE THE POSITIONER IS UNDER MICROCOMPUTER CONTROL.

Should the microcomputer software limits fail, and the limit switches engage, then the following directions apply:

1. Determine which limit has been reached (Azimuth or Elevation)
2. Throw the OVERRIDE switch to OVERRIDE.
3. Throw the switch corresponding to the limited position (Azimutn or Elevation) to its position opposite NORMAL. The radome will then reverse direction and proceed into its normal operating boundary.
4. When the radome has reached the desired position, return the switch thrown in step 3 to its NORMAL position. This will stop the radome.
5. If it is desired to reposition the other axis at this time, throw the appropriate switch (azimuth or elevation) to the position opposite its normal position. This axis will then reverse from the direction it was going when limiting occurred. When the radome is in the desired position on this axis, return the switch to its NORMAL position.
6. After repositioning the radome, return the OVERRIDE switch to its NORMAL position. The radome is then again under microcomputer control.

<u>CAUTION</u>

NEVER TRY TO MANUALLY REPOSITION THE RADOME WITH THE OVERRIDE SWITCH IN ITS NORMAL POSITION.

It should be noted here that once the OVERRIDE switch is returned to NORMAL, the microcomputer is again in control of the positioner. The possibility therefore exists that whatever caused limiting in the first place, could again cause limiting to occur. If this is the case, manually reposition the radome as in steps 1 thru 5 above, but do not carry out step 6. This will prevent the microcomputer from taking control of the positioner and thus prevents limiting from again occurring.

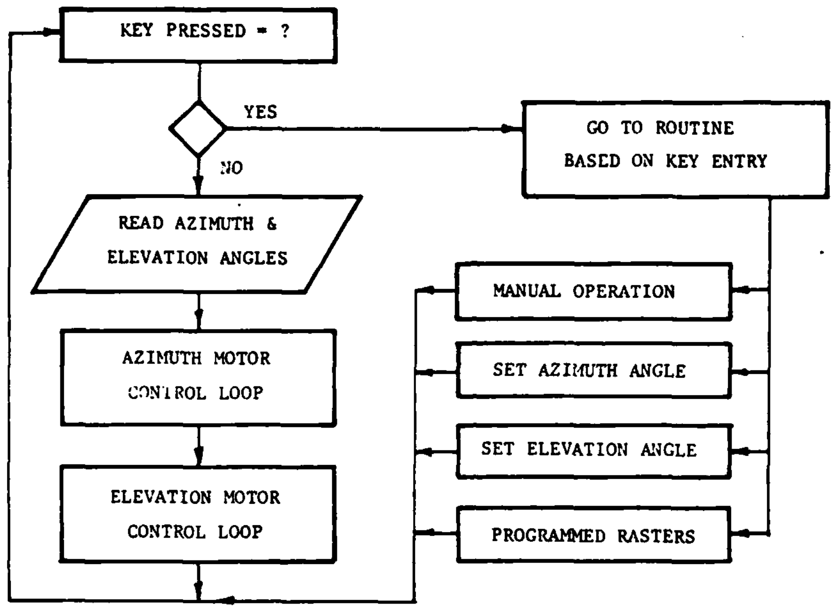
Provision for manual control of the positioner was not meant as a secondary positioning system. Its sole purpose is to place the radome back within its boundary should the microcomputer software limits ever fail. It should be used only for this purpose.

4.3 Microcomputer Software

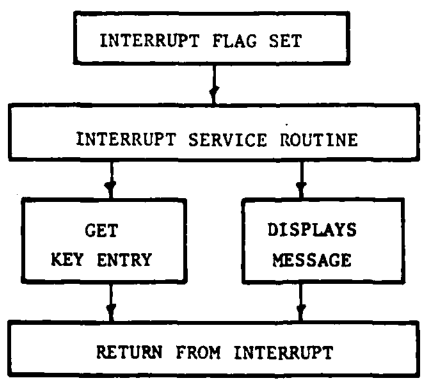
Operation of the Radome Positioner is controlled by a program which is stored in 6k bytes of Read Only Memory (ROM). This program is executed by the 6800 microprocessor and defines all the operations of the positioner. Logical operations defined by the program along with inputs from the shaft angle encoders, keyboard, limit switches and in the case of remote operation, serial ASCII data from an external source, completely specify the motion of the gimbals.

The controller software was written in assembly language for the 6800 microprocessor and converted into machine language instructions by a cross assembler on the Georgia Tech CDC CYBER 74 computer. A listing of this program appears in Appendix B. Each line of the listing consists of a line number followed by an address and either one, two, or three hex bytes corresponding to the machine language equivalent of the assembly language statement on the remainder of the line.

A simple flow chart of the positioner software is shown in Figure 15. This diagram represents the logical flow of the program with each block corresponding to several subroutines of from 10 to 100 lines of program. Program operation can best be understood by considering the main task of the microprocessor as the azimuth and elevation control loop. This portion of the software is constantly comparing the actual position of the gimbals with the desired position which is stored in Random Access Memory (RAM) that is accessed by the microprocessor and is modified by a number of other subroutines. The control loop program can alter the speed and



a. Main Program Loop



b. Interrupt Service Routines

Figure 15. RFSS Radome Positioner Microcomputer Software Flow Chart

direction of each gimbal in order to make the actual and desired angles equal. This digital control loop is analogous to a conventional analog servo control system with the servo amplifiers and loop filters replaced by a microprocessor program.

While the control loop program is operating, it is periodically briefly "interrupted" from one of three sources. An interrupt results in the execution of the current program being briefly suspended while the microprocessor executes another program called the Interrupt Service Routine. After the service routine is completed, the microprocessor resumes execution of the main program (in this case the control loop) at the exact place it was interrupted. The system is designed so that the interrupt service routines take only a few hundred microseconds to complete so in effect the processor's time is "shared" by more than one program without degrading its performance on the main program. The three interrupting devices in the case of the radome positioner are the display, the keyboard and the serial interface for remote operation. Since the display requires periodic refreshing, an interrupt from a 100 μs clock causes the microprocessor to transfer the contents of a specific 20 byte portion of RAM at the rate of one character (1 byte) every 100 μs. Thus the entire display is updated or "refreshed" every 2 ms, which is faster than the flicker response of the eye resulting in a display that appears to be continuous. (See Figure 16).

An interrupt is also generated whenever a key on the positioner keyboard is pressed. The keyboard interrupt service routine decodes the key that was pressed and determines what action to take by means of a key state table within the program. This table specifies what action the program will take based on which key was pressed and the current status or "state" of the program. For example, a particular state is associated with the prompt "Enter Azimuth Angle" which results from pressing the "SET AZ" key. If a number key is pressed, the keyboard service routine

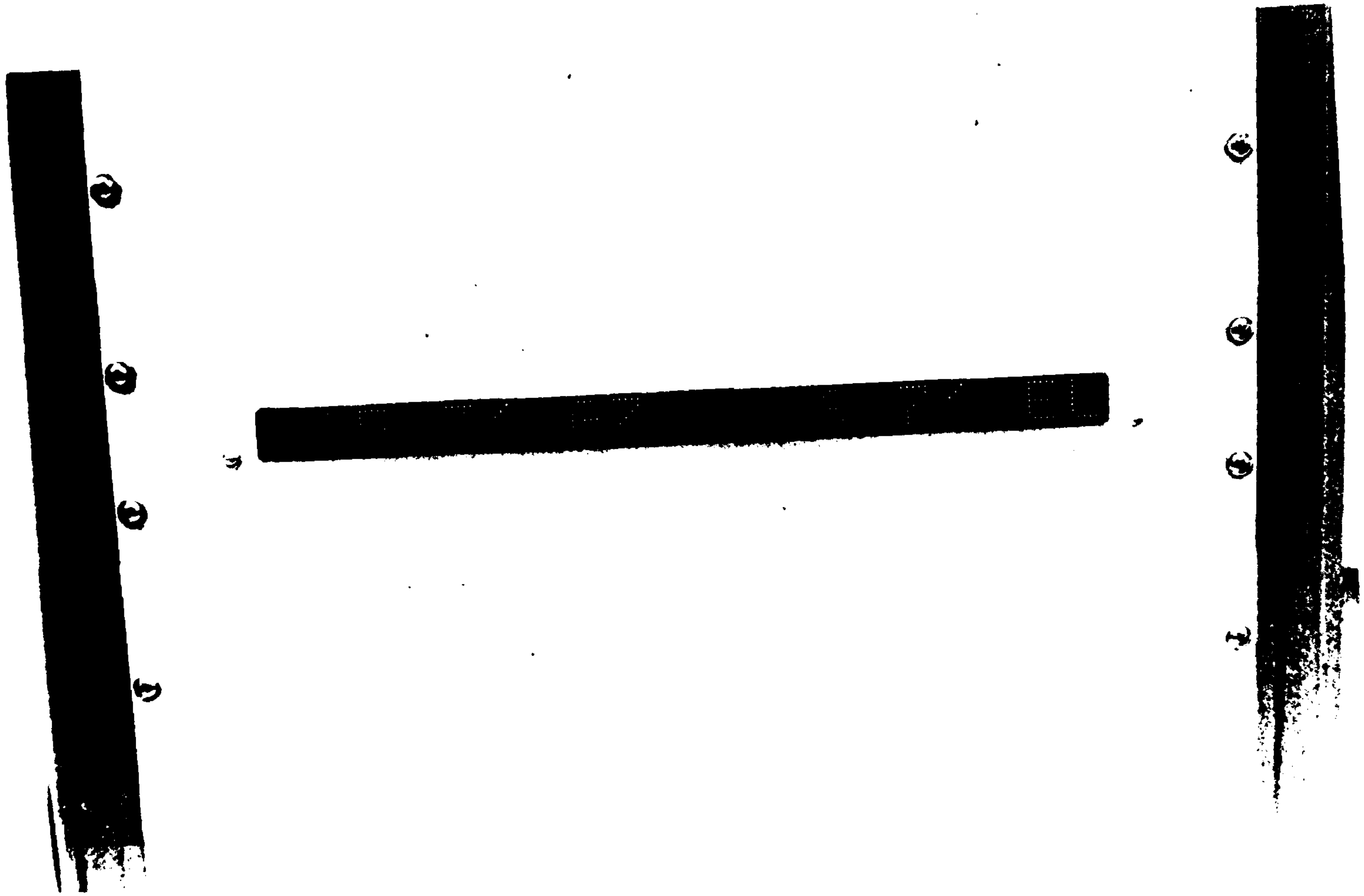


Figure 16. Alphanumeric Display Showing Angle Readout.

recognizes it as a proper response and stores it as the first digit of the desired angle. If, however, the "SET EL" key or one of the manual direction keys are pressed, the keyboard program will recognize it as an invalid key resulting in the display of an "Invalid Entry" message and will then return to state 0 which is the "Resting State" and display the current azimuth and elevation angle.

A similar sequence of operations results from commands that are received via the serial interface from a remote device in the Remote Operation Mode. In this case the interrupts are generated by the interface each time a valid ASCII character is received. Error messages and prompts are sent over the interface as well as being displayed.

5.0 POSITIONER OPERATION

The radome positioner is capable of four basic modes of operation: manual azimuth and elevation movement, preset azimuth and elevation angles, internally stored raster patterns, and external computer control via a serial interface. The first three modes are accessed via the controller keyboard and the fourth via an RS232C serial interface when the local/remote switch is in the REMOTE position. The alphanumeric display will prompt the operator for any required keyboard inputs and provide error messages if any invalid key sequence is entered. In addition, failure of any one of several self-test programs will be indicated by an appropriate message. The display will also echo numeric entries and indicate the present azimuth and elevation angle.

System Initialization

When power is applied to the positioner by pressing the power switch located in the lower right-hand corner of the controller front panel, the button will be illuminated and an automatic reset sequence will be initiated in the microcomputer. This will cause the microcomputer to begin executing the controller program. A power-up message will be displayed on the controller's display indicating successful system initialization. See Table 3 for a complete list of system messages.

Manual Operation

Manual control of gimbal position is available via the four arrow keys on the keyboard. Depressing any of these keys will cause the gimbals to move in the indicated direction. Only one key at a time may be depressed. The azimuth and elevation angles will be displayed on the controller display. Gimbal motion will continue as long as the key is depressed or until the gimbal angle limit is exceeded.

TABLE 3. SYSTEM MESSAGES

MESSAGE	MEANING
THE GA. TECH - RFSS RADOME POSITIONER VERSION 1.0.	Power on message giving current software version number.
RADOME POS. READY	Indicates proper initialization sequence complete.
XX.X' AZIMUTH XX.X' ELEVATION	Present azimuth and elevation angle. Denotes idle state if constant or follows gimball motion.
ENTER ELEVATION ANGLE ENTER AZIMUTH ANGLE	Prompt for angle entry after pressing SET AX or SET EL key.
ERROR-INVALID ENTRY	Improper key sequence entered.
ANGLE TOO LARGE . . .	Entry of angle greater than azimuth or elevation angle limit.
POSITIONER HALTED	Result of pressing $\frac{\text{START}}{\text{STOP}}$ key when gimbals are in motion.
ANGLE LIMIT EXCEEDED	Azimuth or elevation gimbal has reached preset limit.
ENTER "PROGRAM" NUMBER	Response to PGRM key (see raster pattern descriptions).

Preset Angle Mode

The gimbals may be commanded to any azimuth and elevation angle via the SET EL and SET AZ keys. When either of these keys are depressed the operator will be prompted to enter the desired angle. Each angle may be set independently of the other. Gimbal motion will begin when the $\frac{\text{START}}{\text{STOP}}$ key is depressed following the angle entry. The gimbals may be halted at any time by pressing the $\frac{\text{START}}{\text{STOP}}$ key again. If entry of an angle greater than the gimbal angle limit is attempted, an error message will be displayed (See Table 3).

Internally Stored Patterns

Several commonly used raster pattern programs are stored internally in the controller's microcomputer. These programs may be accessed via the PRGM key on the controller keyboard followed by a program number. Each program will prompt the operator to enter any required pattern variables. A complete description of each program is given in Appendix E. Briefly, the choice of rasters is vertical and horizontal linear rasters, or circle or star rasters.

External Computer Control

The radome positioner has the capability to be controlled by an external device such as a computer or computer terminal that has the capability to send and receive ASCII characters over an RS232C interface. When the local/remote switch located on the front panel is placed in the remote position all of the functions available through the keyboard can also be commanded over the RS232C interface. In addition, messages and prompts similar to those displayed on the controller display are sent over the interface. Thus, an operator or computer program can operate the positioner in essentially the same manner as when the keyboard and display are used. See Appendix E for a detailed description of the computer interface operation.

Gimbal Angle Limits

In order to prevent the gimbals from being moved past an angle that could cause damage to the seeker antenna and hit the sides of the aperture or gimbal frame, two limiting mechanisms were implemented. The first is a set of four "soft" limit angles that can be manually set from the keyboard. These angles correspond to the positive and negative extreme of each axis and are initially set to 40° when power is first applied to the system. They may be changed by entering the key sequence; "SET AZ (EL)" followed by one of the four manual direction controlled buttons; "↑", "↓", "→", "←". The display will then show the current limit and allow it to be changed. Attempting to set a limit greater than 40° or less than 1° for any axis will result in an error message and no change in the limit will occur. Should the angle reach the preset limit during positioner operation, the message ANGLE LIMIT EXCEEDED will be displayed and the gimbals will automatically move the opposite direction until within the limits.

A second fail-safe angle limit system consists of four microswitches mounted on the azimuth and elevation axis gears. These switches will interupt ac power to the motors should the gimbals ever exceed the angle at which they are set. They are ajustable from 20° to 40° and are normally set at 40°. See Section 4.2 for instructions on how to reposition the gimbals after these "hard" limits are exceeded.

Rear Panel Connections

The display, motor controller, shaft encoders, and an external RS232 device connect to the system via connectors located on the rear panel of the main chassis (See Drawing 81). The two BNC connectors labeled "Azimuth dc Out" and "Elevation dc Out" provide dc voltages proportional to gimbal position. From $\pm$ 40° to − 40° on either axis corresponds to + 4 Vdc to − 4 Vdc output with 0° = 0 Vdc. These outputs are unbuffered D/A converter outputs and can provide up to 5 mA of drive current.

APPENDIX A

RADOME POSITIONER DRAWINGS

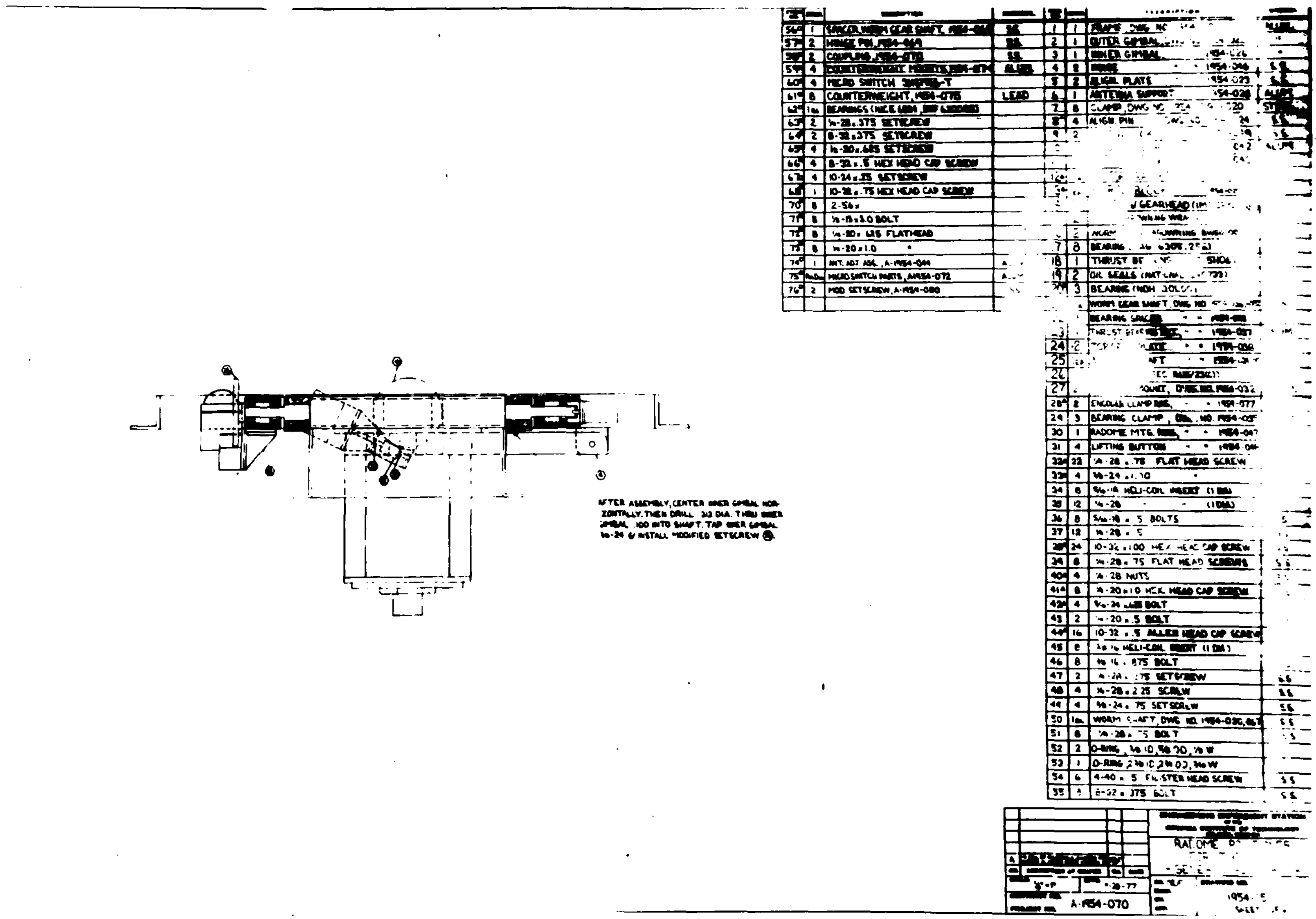
DRAWING LIST

Radome Positioner Drawing No.	Title
1	Radome Positioner for the RFSS (Sheet 1 of 2)
2	Radome Positioner for the RFSS (Sheet 2 of 2)
3	Clamp for Radome Positioner
4	Screw Assembly for Radome Positioner Clamp
5	Outer Mounting Frame for Radome Positioner
6	Alignment Plate for Seeker Antenna
7	Alignment Pin for Seeker Antenna
8	Outer Gimbal Assembly
9	Outer Gimbal Assembly
10	Insert for Outer Gimbal
11	Seeker Antenna Support Bracket
12	Bearing Block and Motor Mtg. Plate for Azimuth Axis
13	Worm Shaft for Azimuth Axis
14	Bearing Block Mounting Bracket for Azimuth Axis
15	Encoder Mounting Block
16	Bearing Spacer - Az. & El. Axis
17	Encoder Mounting Shaft - Azimuth Axis
18	Bearing Clamp - El. Axis
19	Worm Gear Shaft - Azimuth Axis
20	Thrust Bearing Plate - Azimuth Axis
21	Top Seal Plate Worm Gear Shaft - Azimuth Axis
22	Seeker Antenna Alignment Assembly
23	Seeker Antenna Horizontal Adjustment Plate
24	Seeker Antenna Vertical Adjustment Plate
25	Encoder Mounting Shaft - El. Axis
26	Hinge for Seeker Antenna Mtg. Bracket
27	Radome Mounting Ring
28	Lifting Button for Outer Frame
29	Screw Lock for Seeker Antenna Mtg. Frame
30	Worm Gear Mounting Shaft - El. Axis
31	Worm Gear Modifications
32	14" Radome Adapter Ring
33	Bearing Block and Motor Mounting Plate - El. Axis
34	Bearing Block Mounting Bracket - El. Axis
35	Worm Shaft - El. Axis
36	Worm Gear Shaft Spacer - El. Axis
37	Hinge Pin for Seeker Antenna Bracket
38	Shaft Coupling Motor to Worm
39	Micro-Switch Actuator and Mounting Plate
40	Counter Weight Mounting Blocks

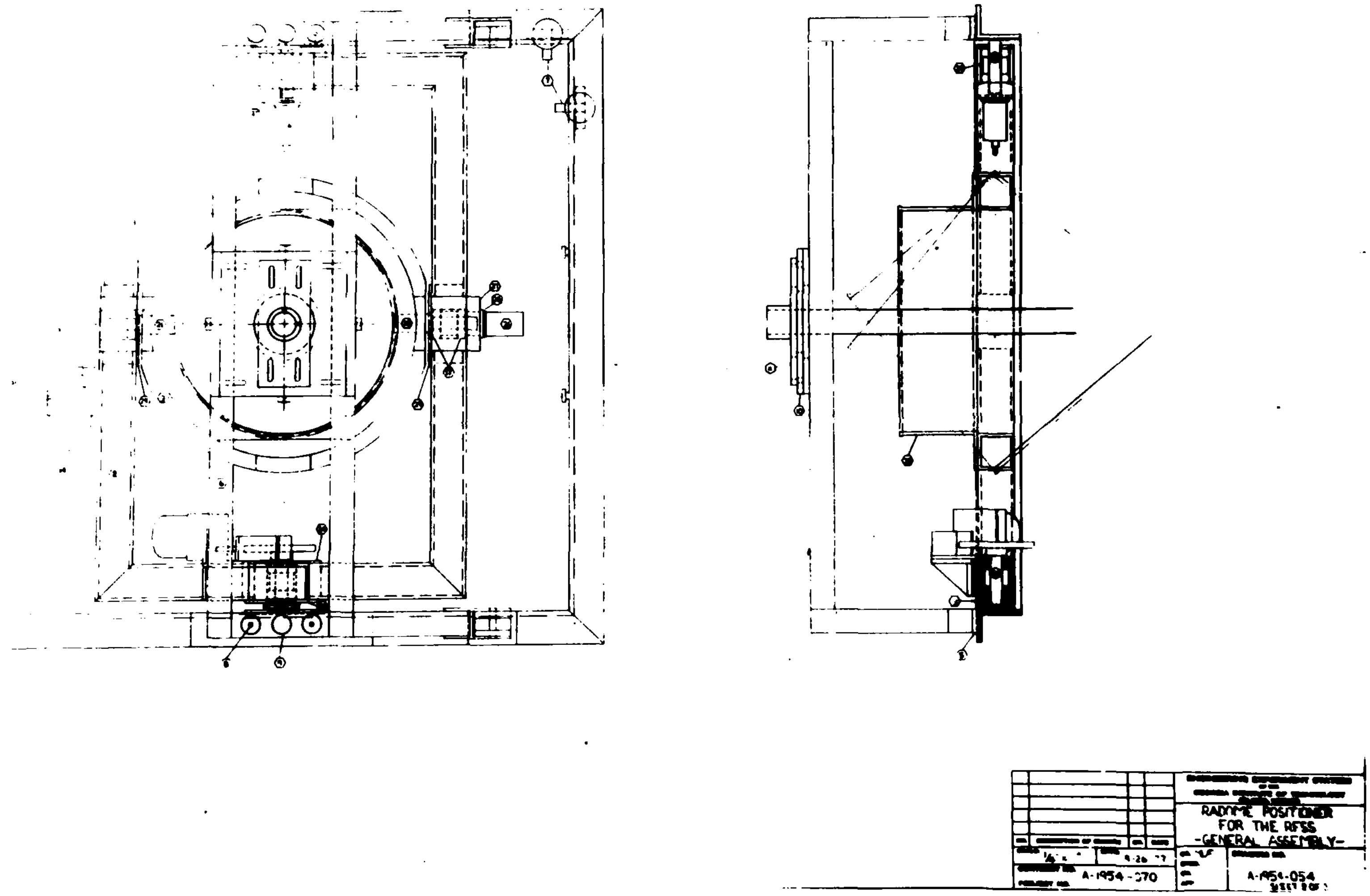
DRAWING LIST (CONTINUED)

Drawing No.	Title
41	Counter Weight
42	Encoder Clamp Ring
43	Modified Set Screw for Centering of Inner Gimbal
44	Modification of Front Panel M68MMLC1 Micro-Module
45	Modification of Back Panel M68MMLC1 Micro-Module
46	Keyboard Assembly
47	Box for Keyboard
48	Key Mounting Board and Bracket
49	Panel Modifications for Digital Readout
Cart for Positioner	
50	Lifting Hook Assembly for Radome Positioner
51	Lifting Hook Details
52	Threaded Plate for Lifting Hook
53	Retractor Mechanism Components for Radome Positioner Removal Cart
Door Removal Cart	
54	Modified Hand Truck for Removing Door of RFSS
55	Door Bottom Support Bracket
56	Door Clamp Screw Top
57	Door Clamp Top
58	Cap Support Bracket
59	Upright Support
60	Gusset Plate
61	Angle Support
62	Channel Support
63	Angle Stop
Electronics	
64	M68MM01A-1 Microcomputer Schematic
65	Parallel and Serial Interface Schematic for M68MM01A-1
66	M68MM06 2k Byte Static RAM Board Schematic
67	M68MM03 16k Byte EPROM Board Schematic
68	M68MM03 32 Channel I/O Board Schematic
69	I/O Buffer Board Schematic
70	I/O Buffer Board Parts Placement

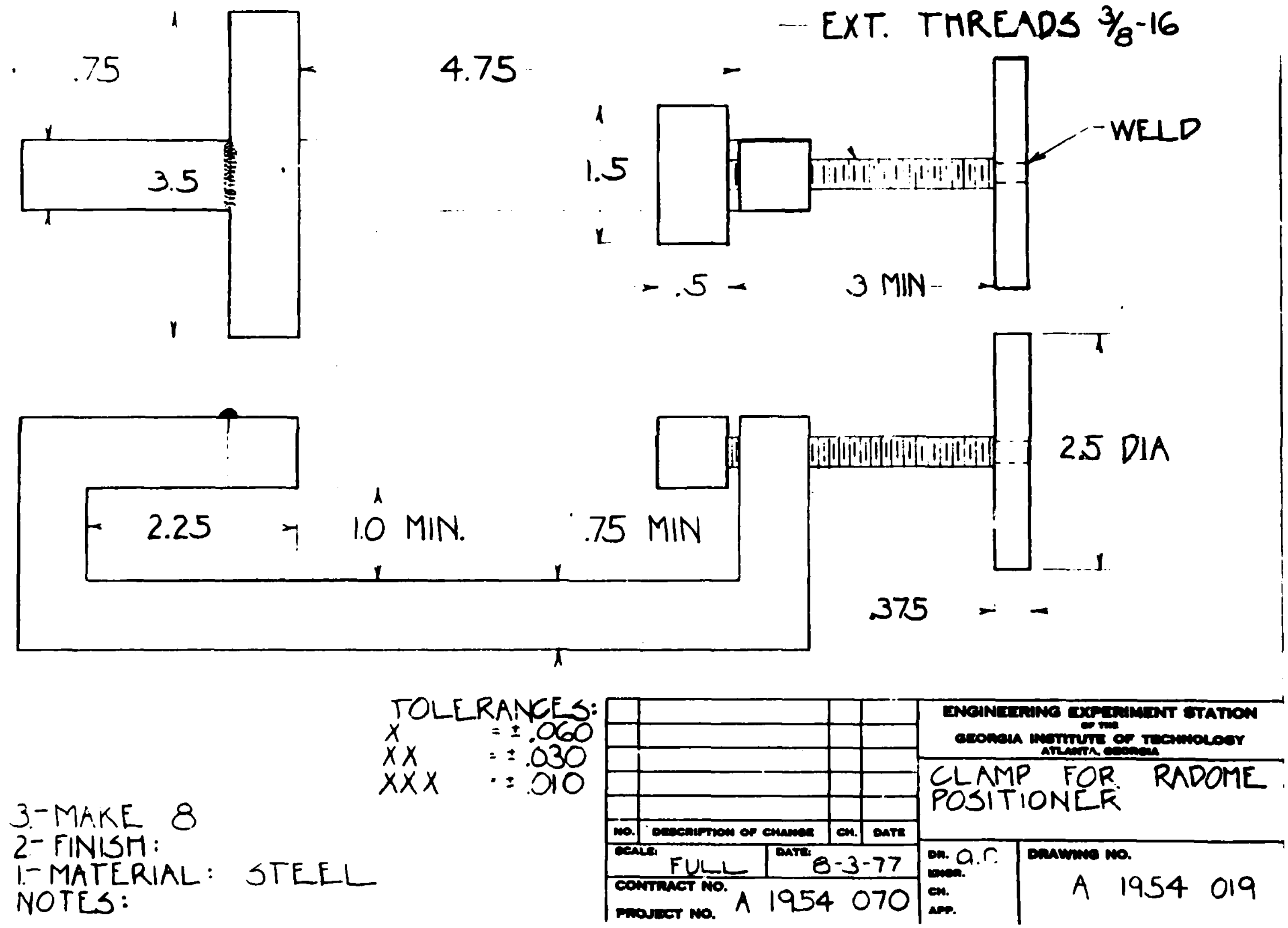
DRAWING LIST (CONTINUED)



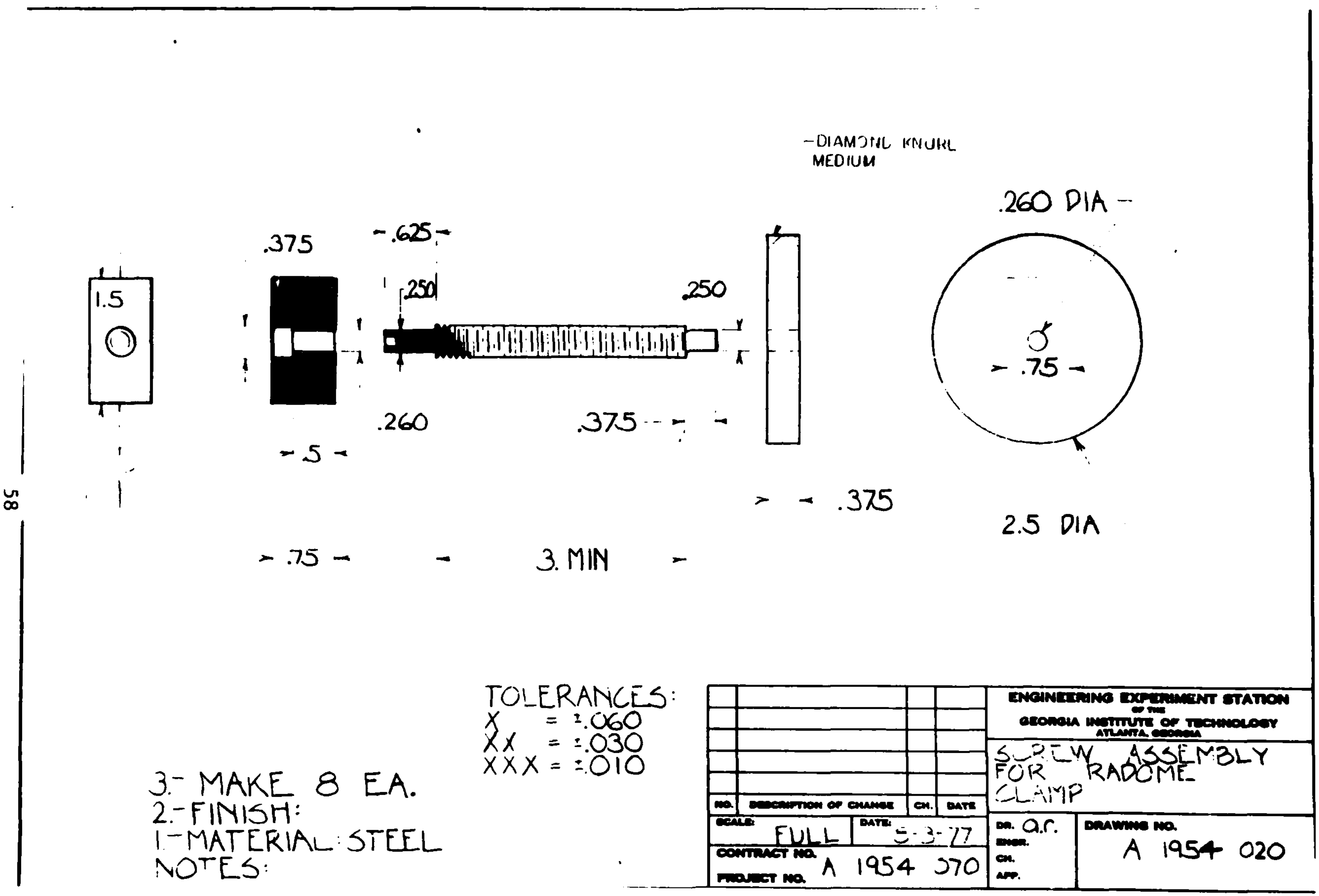
Drawing 1. Radome Positioner for the RFSS (Sheet 1 of 2)



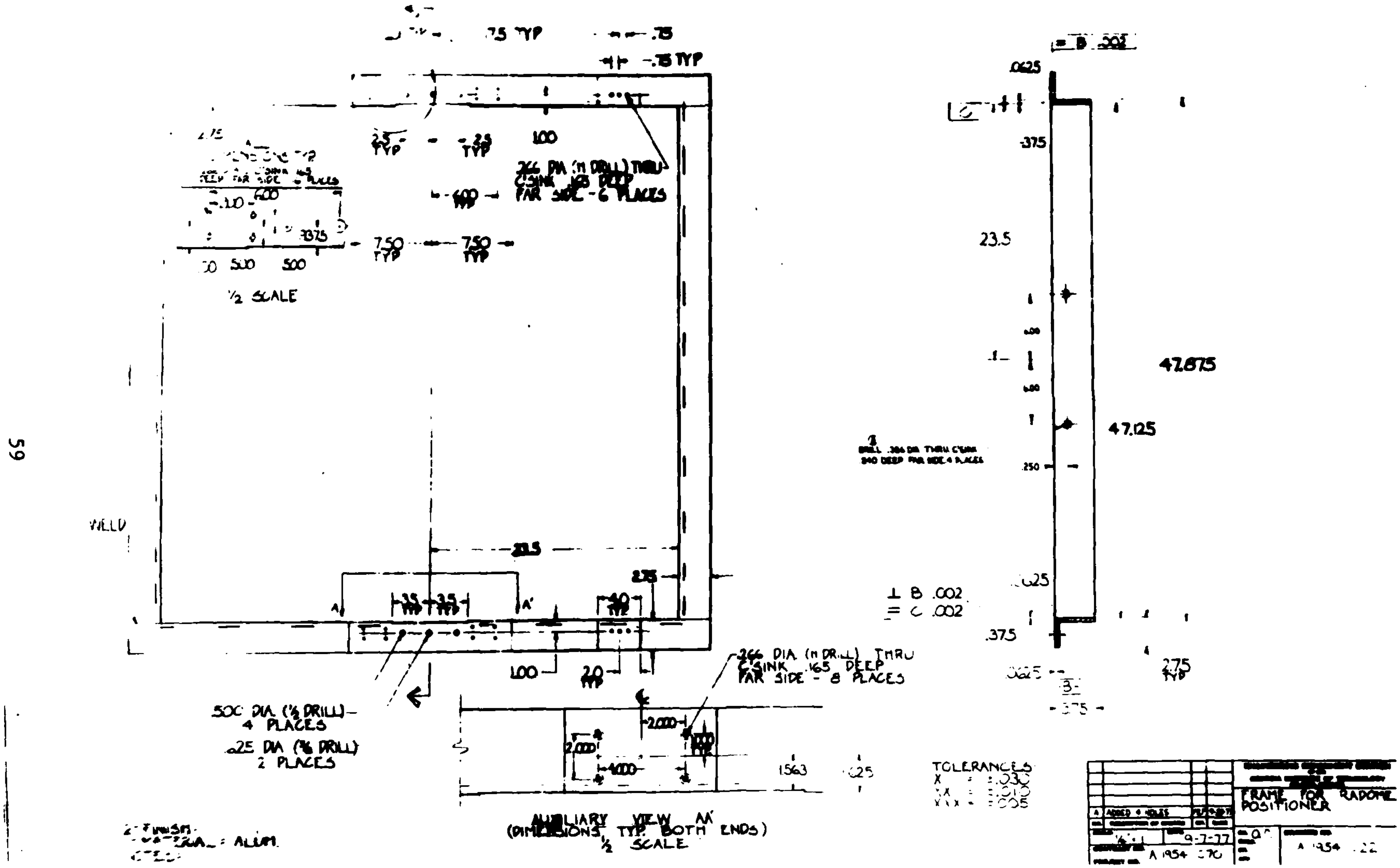
Drawing 2. Radome Positioner for the RFSS (Sheet 2 of 2)



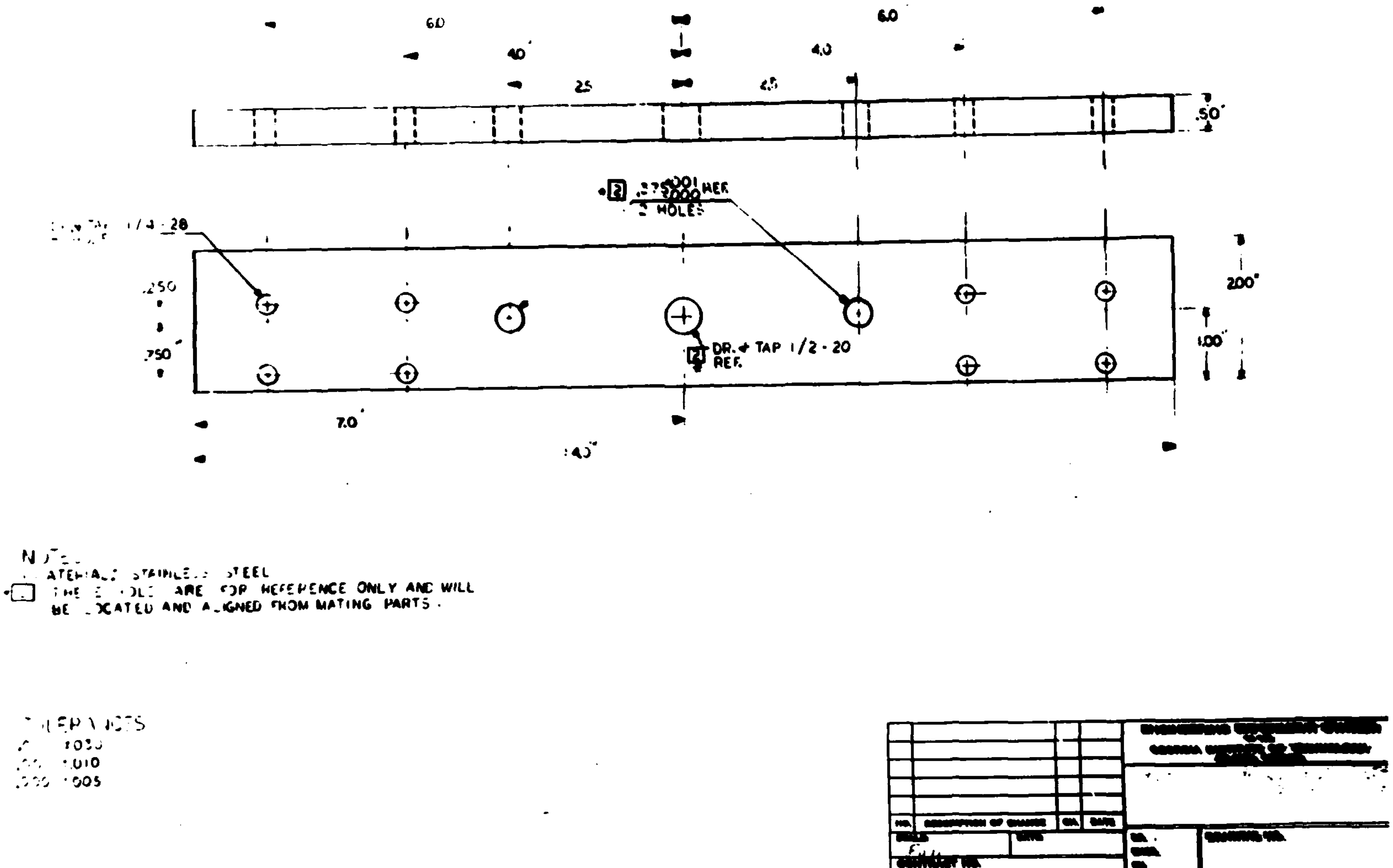
Drawing 3. Clamp for Radome Positioner



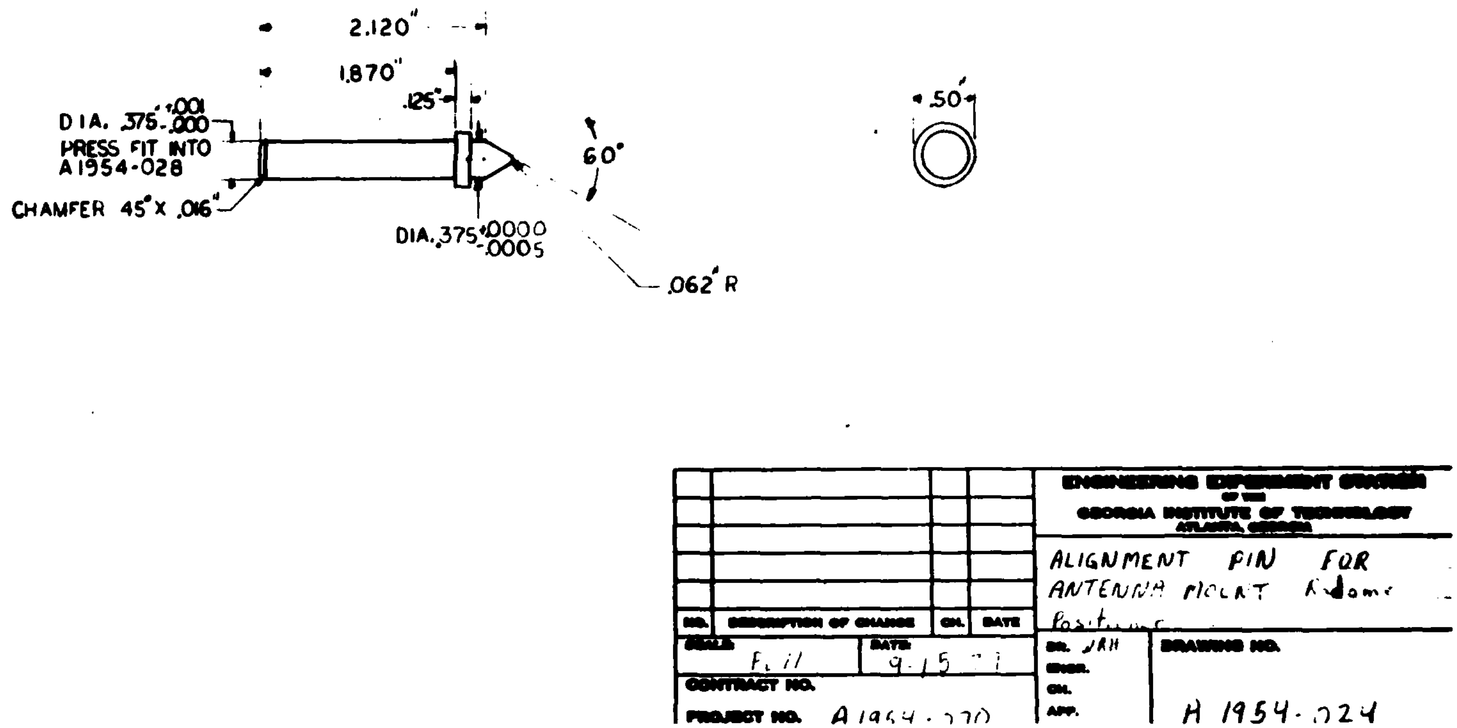
Drawing 4. Screw Assembly for Radome Positioner Clamp



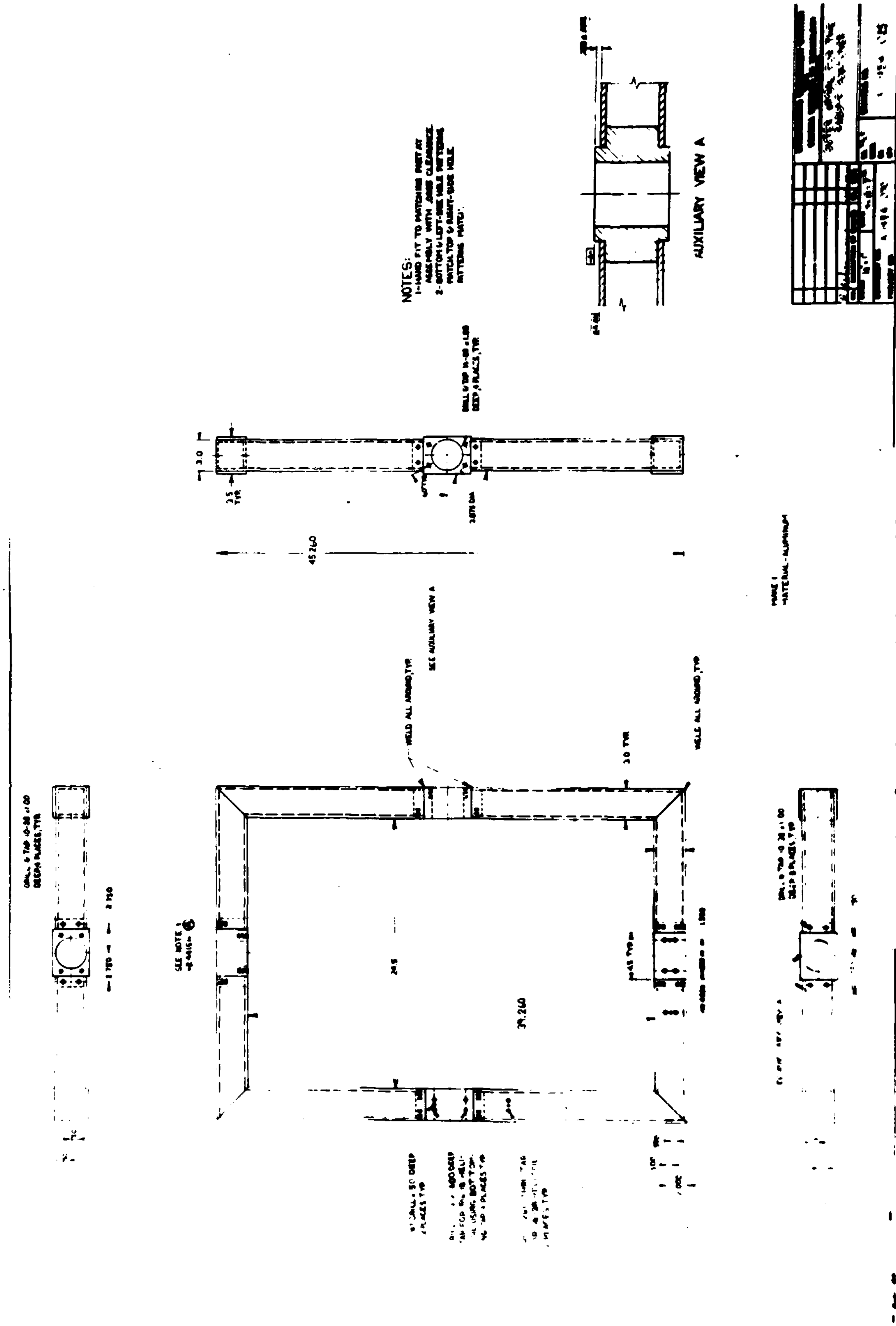
Drawing 5. Outer Mounting Frame for Radome Positioner



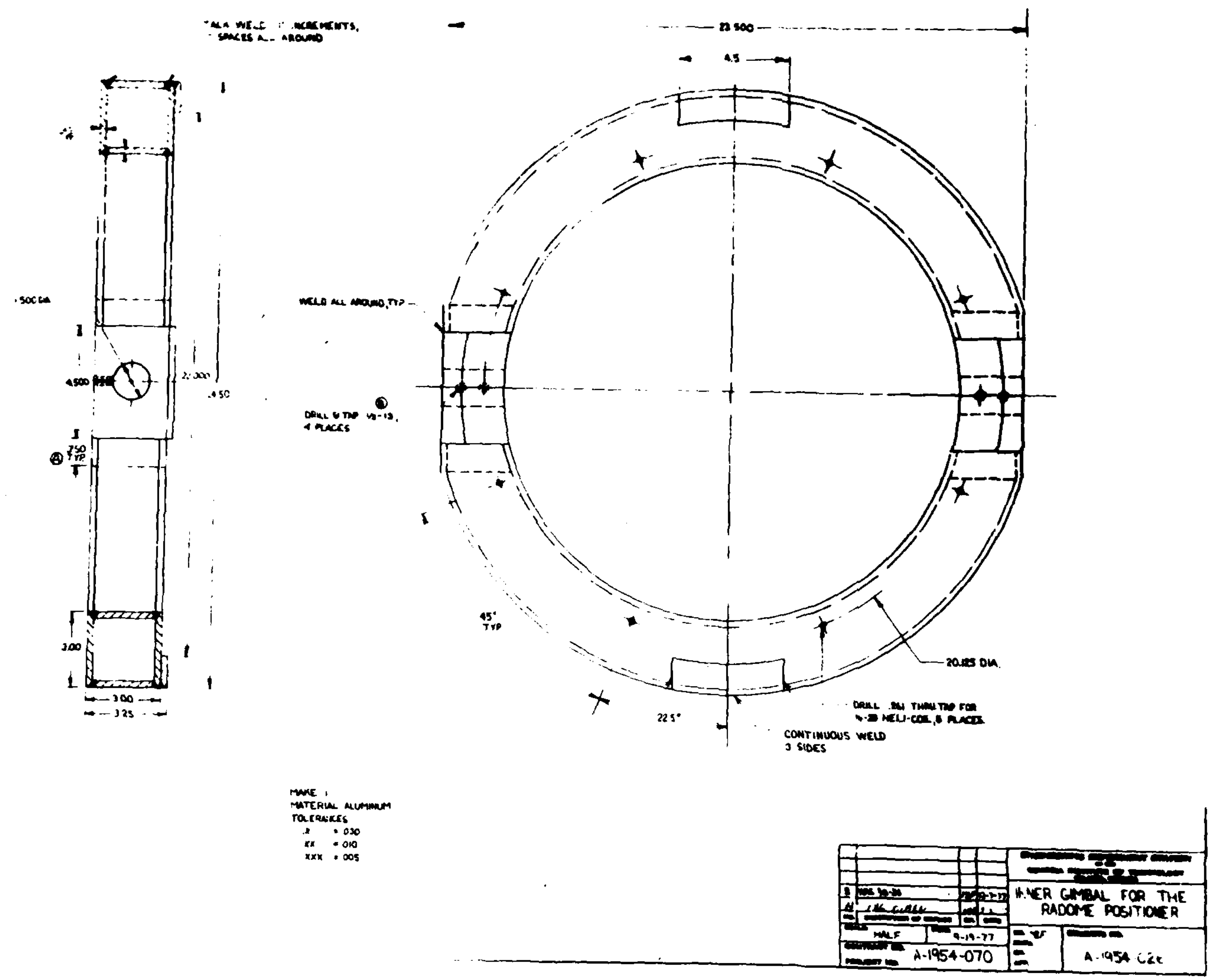
Drawing 6. Alignment Plate for Seeker Antenna.



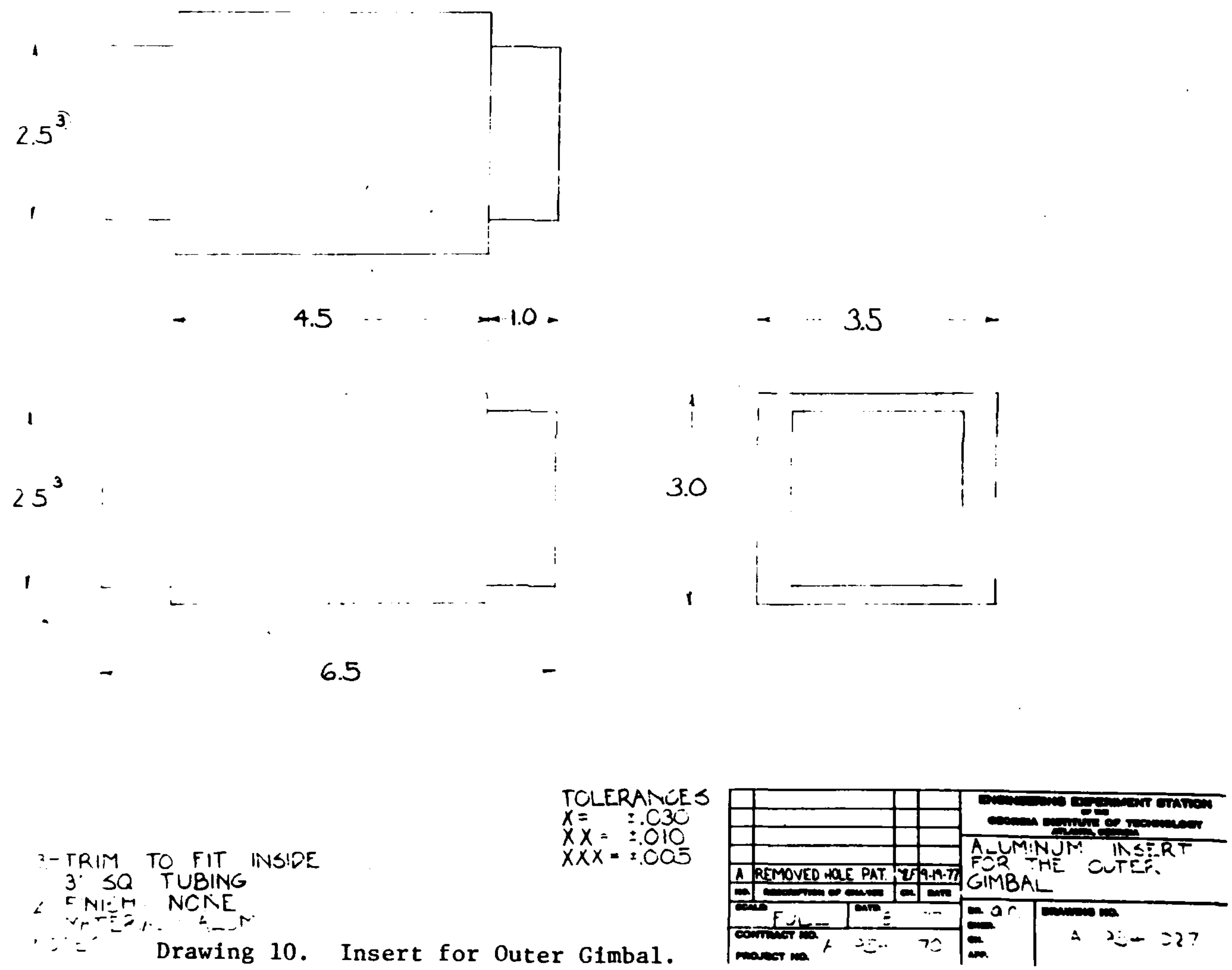
Drawing 7. Alignment Pin for Seeker Antenna



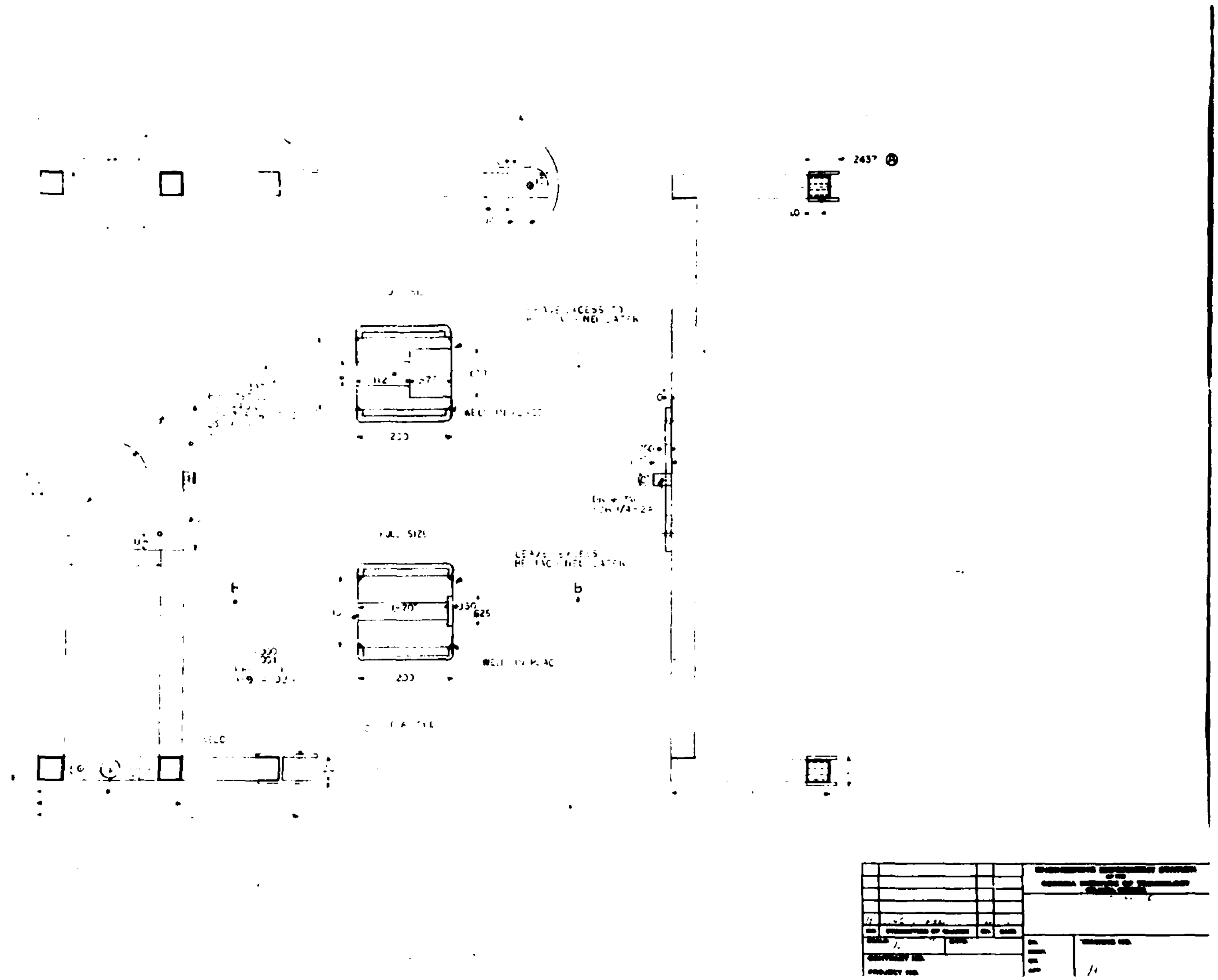
Drawing 8. Outer Gimbal Assembly.



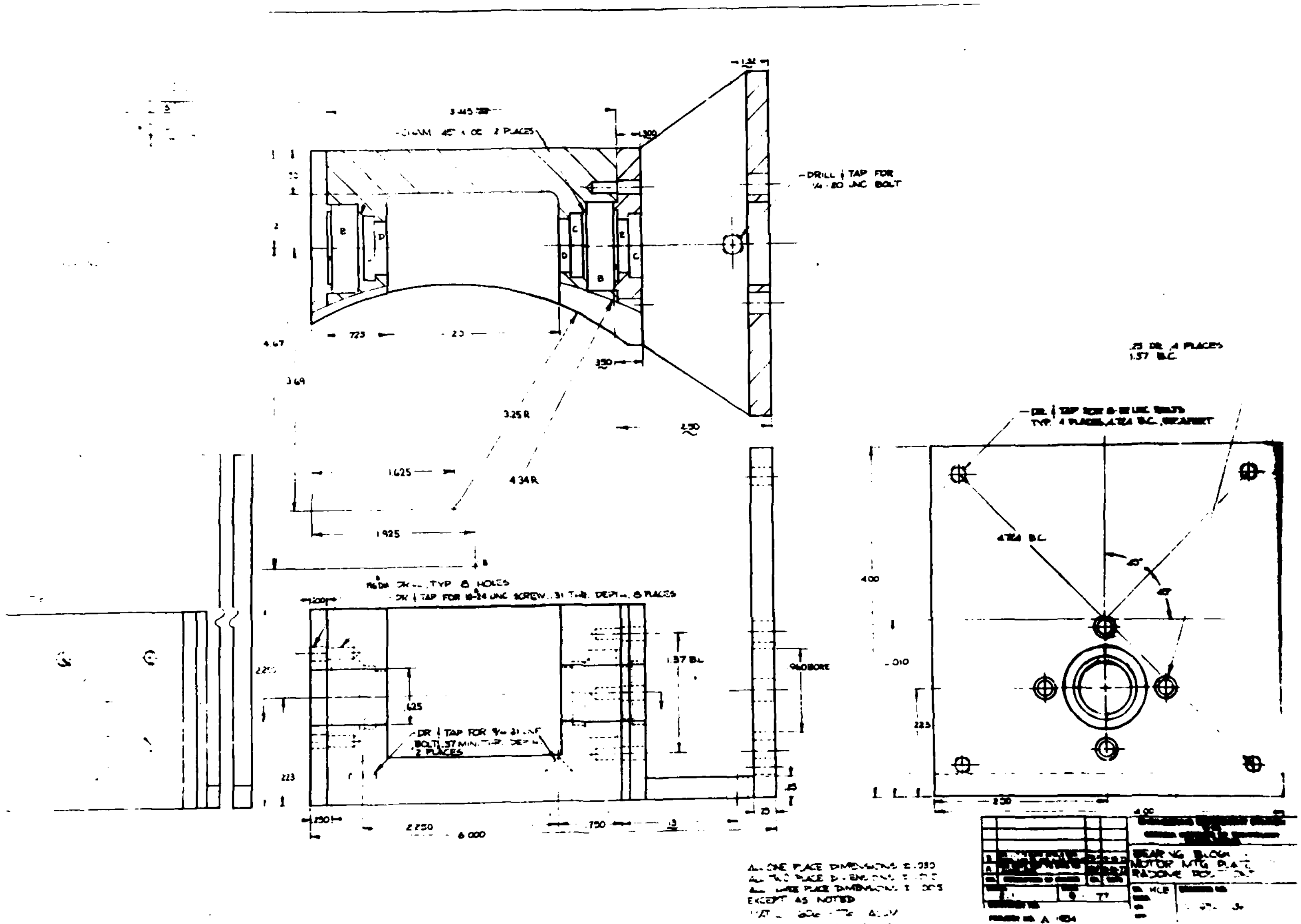
Drawing 9. Inner Gimbal Assembly



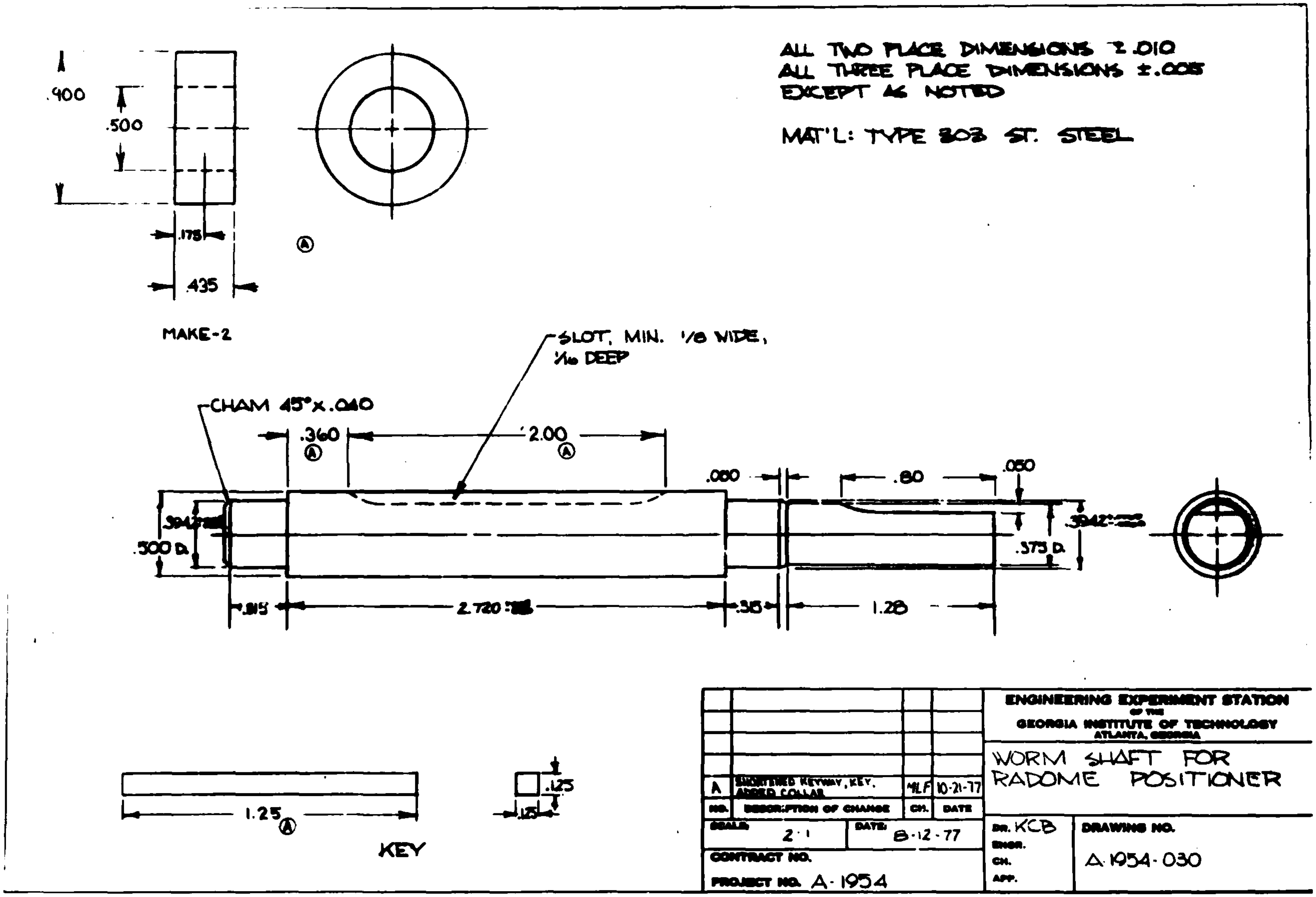
Drawing 10. Insert for Outer Gimbal.



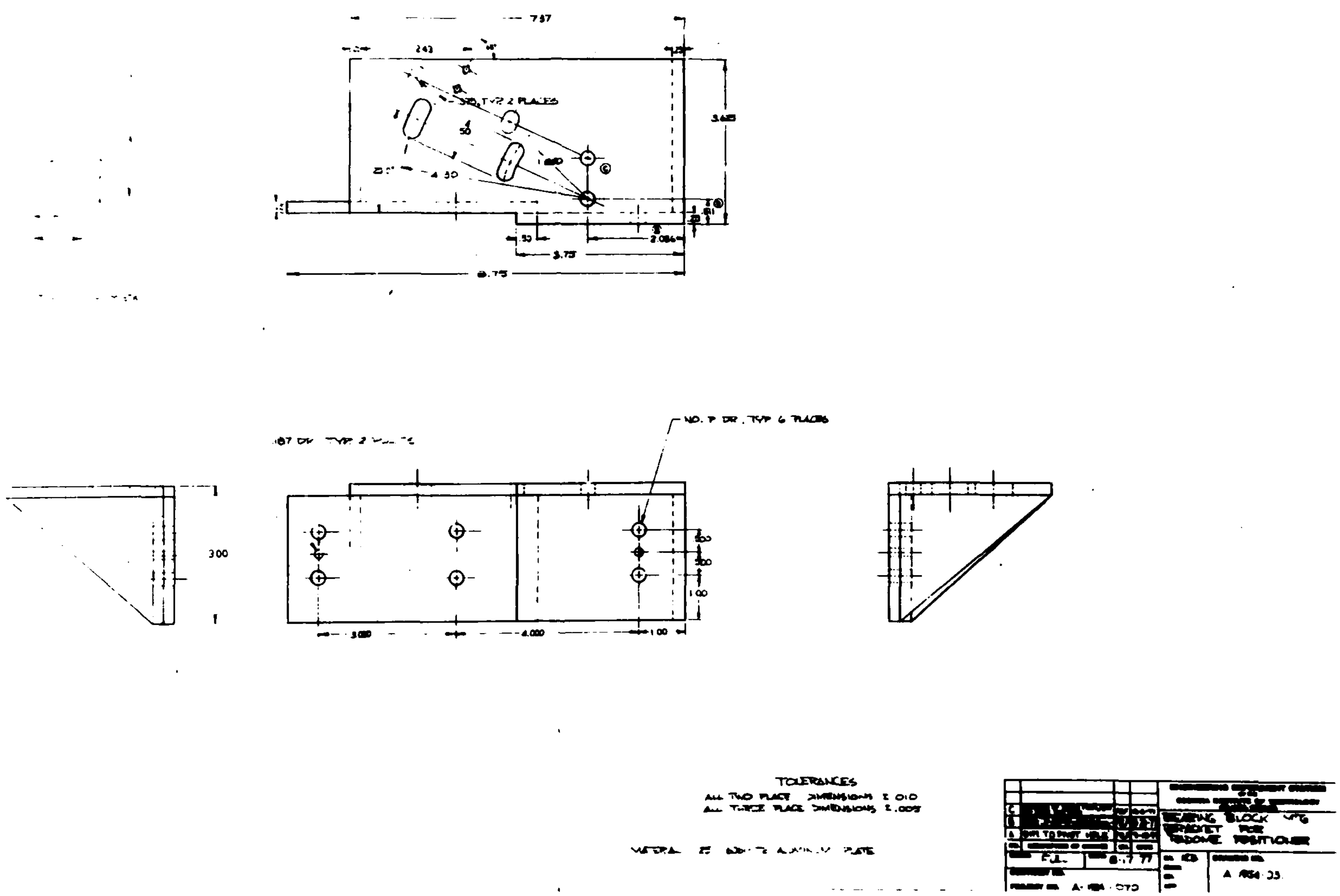
Drawing 11. Seeker Antenna Support Bracket



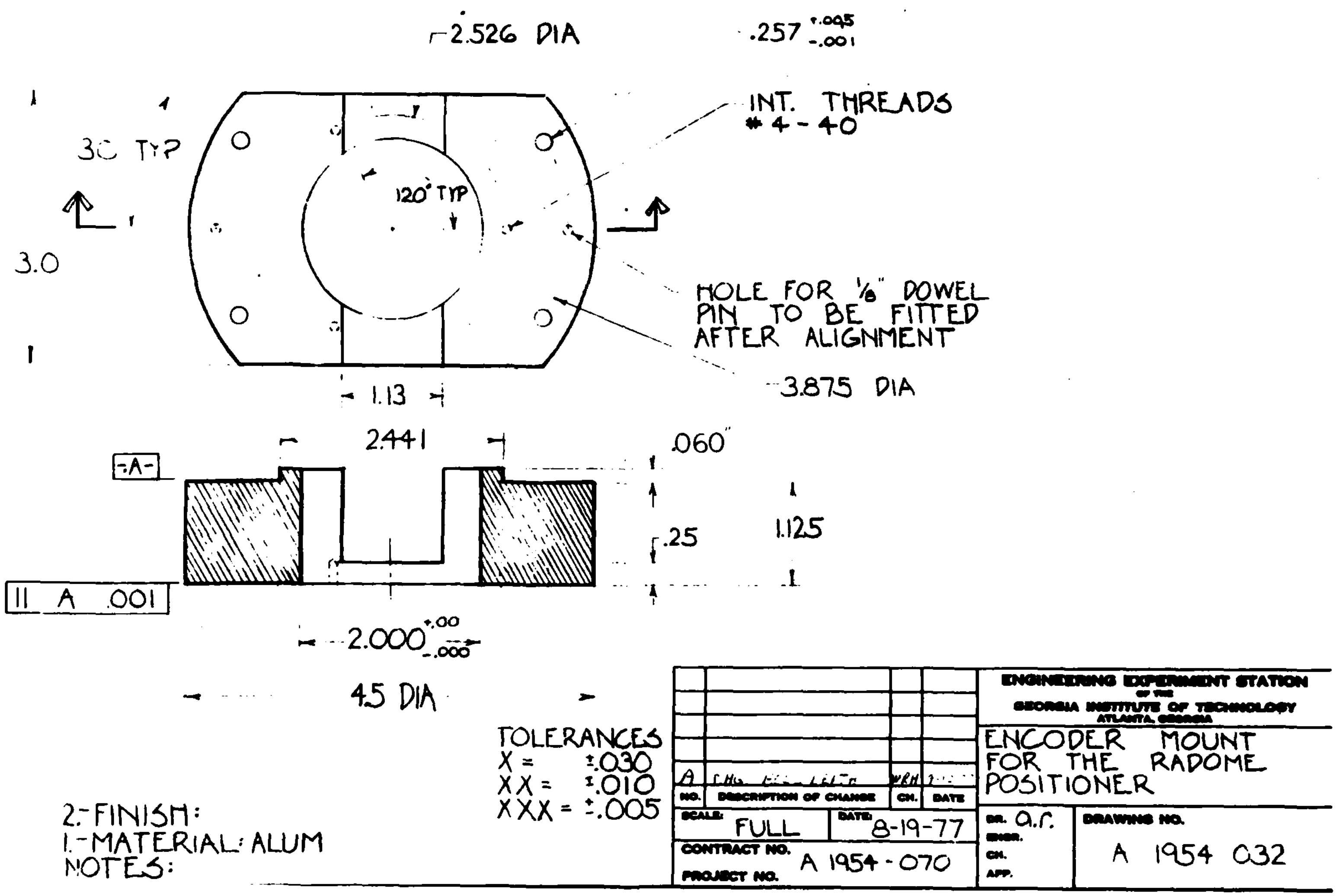
Drawing 12. Bearing Block and Motor Mtg. Plate for Azimuth Axis.



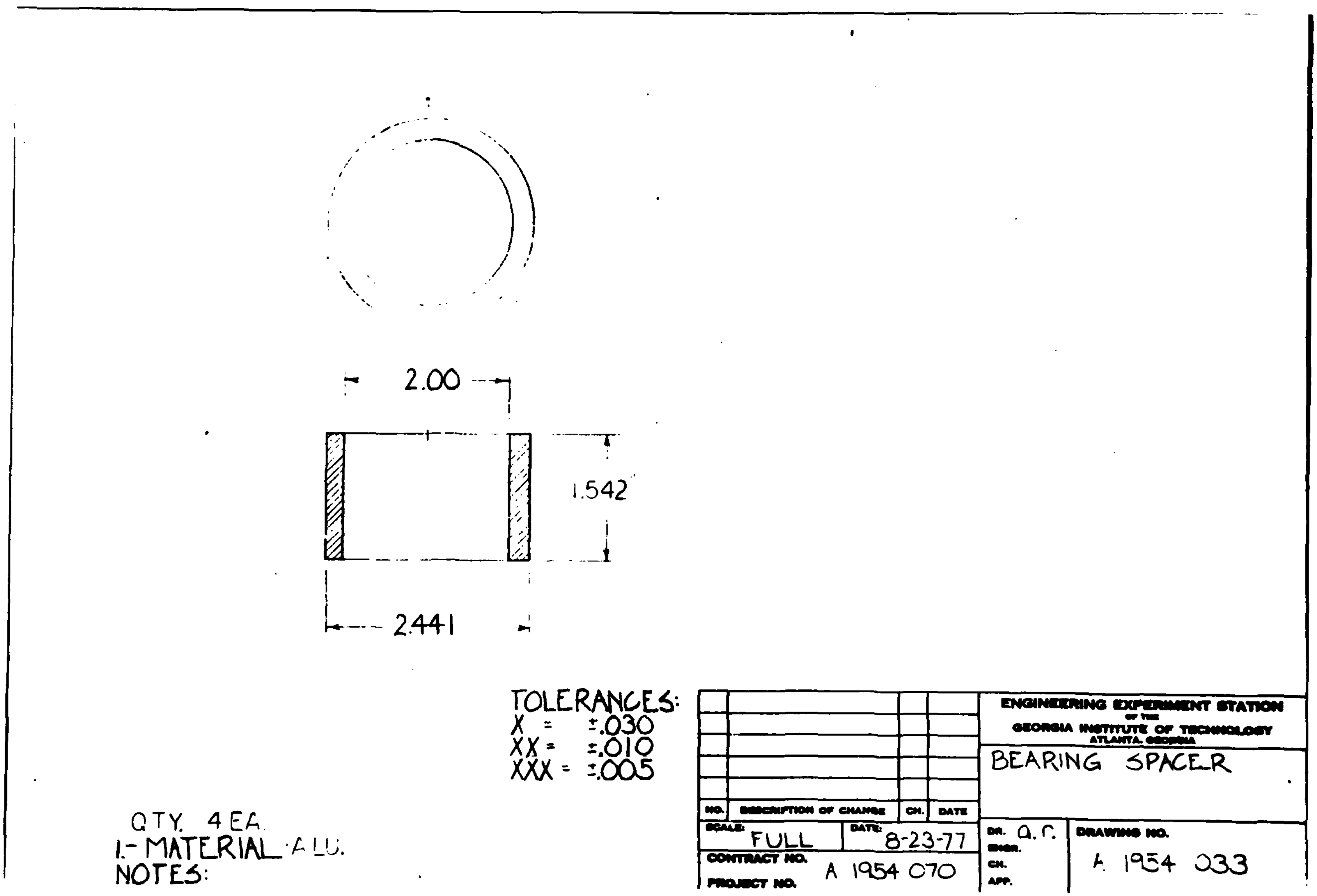
Drawing 13. Worm Shaft for Azimuth Axis



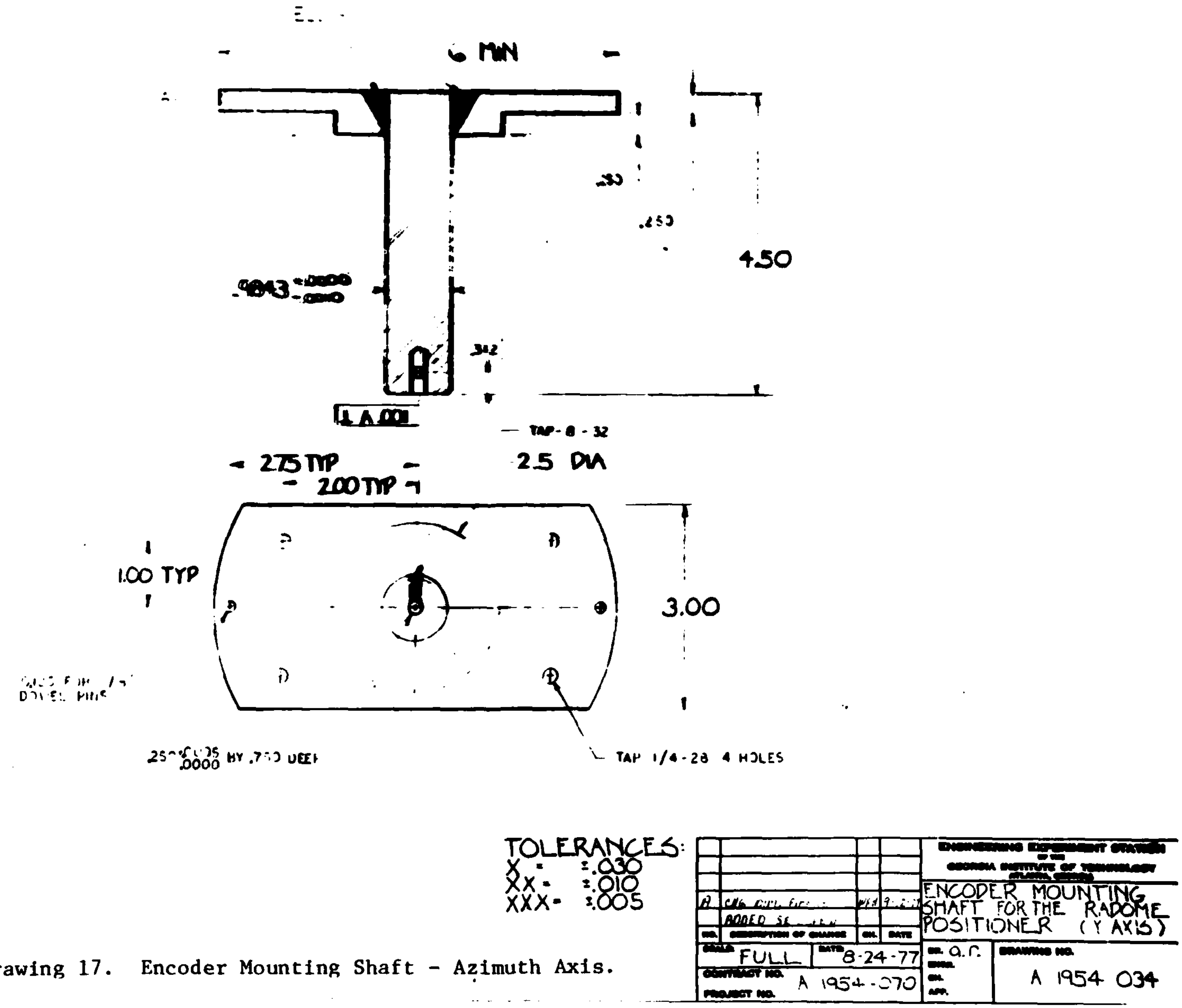
Drawing 14. Bearing Block Mounting Bracket for Azimuth Axis



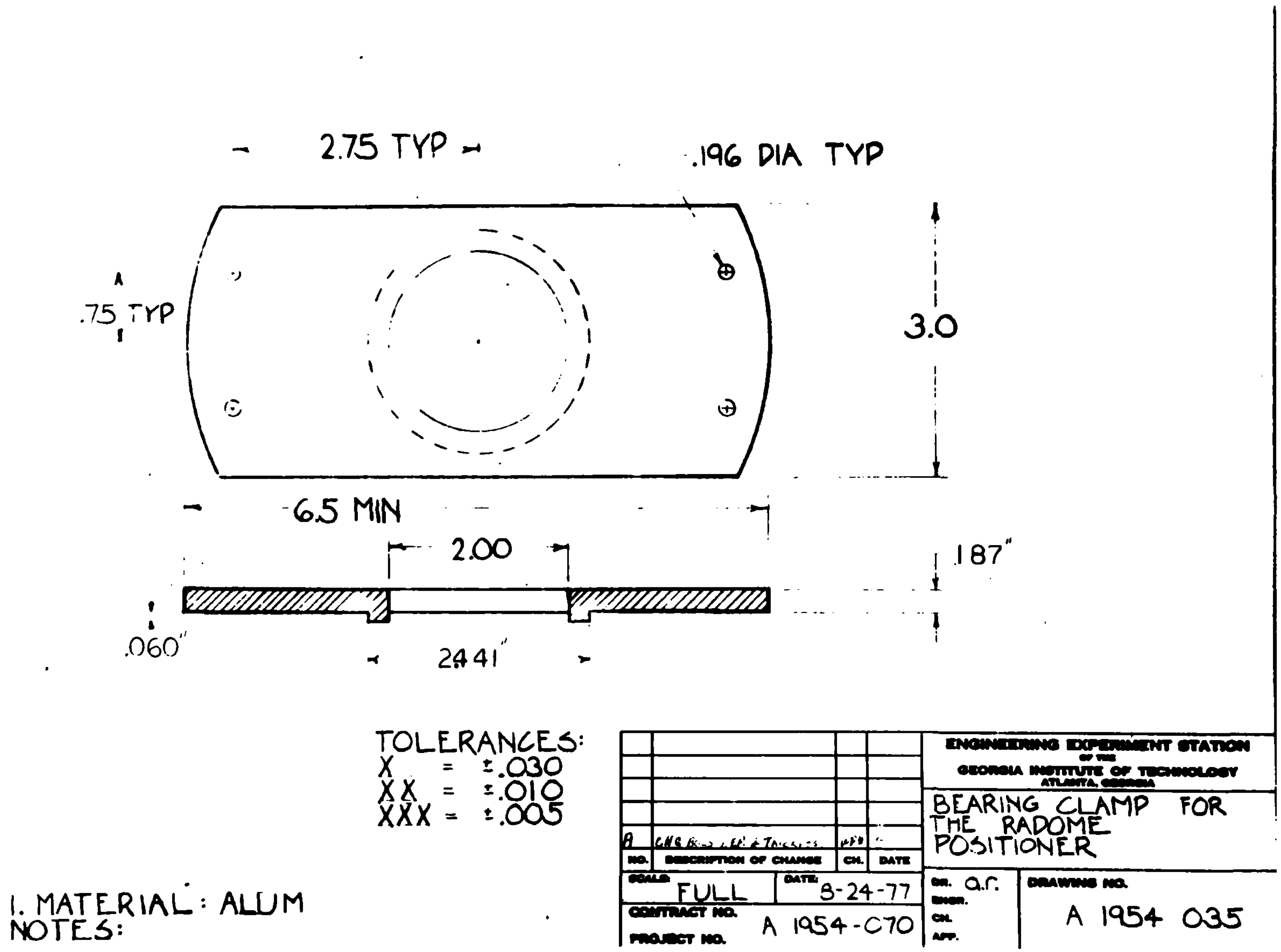
Drawing 15. Encoder Mounting Block



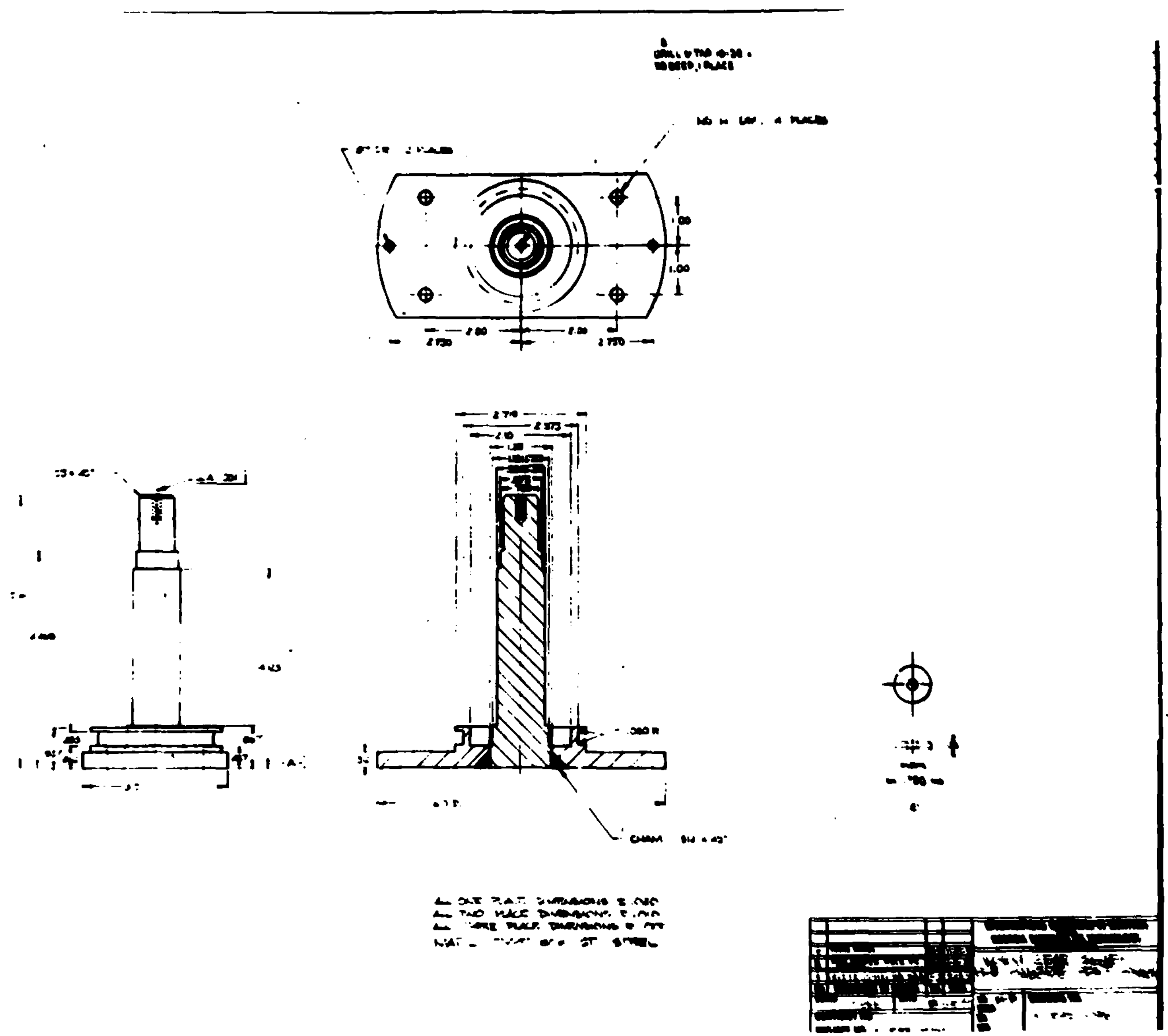
Drawing 16. Bearing Spacer - Az & El Axis



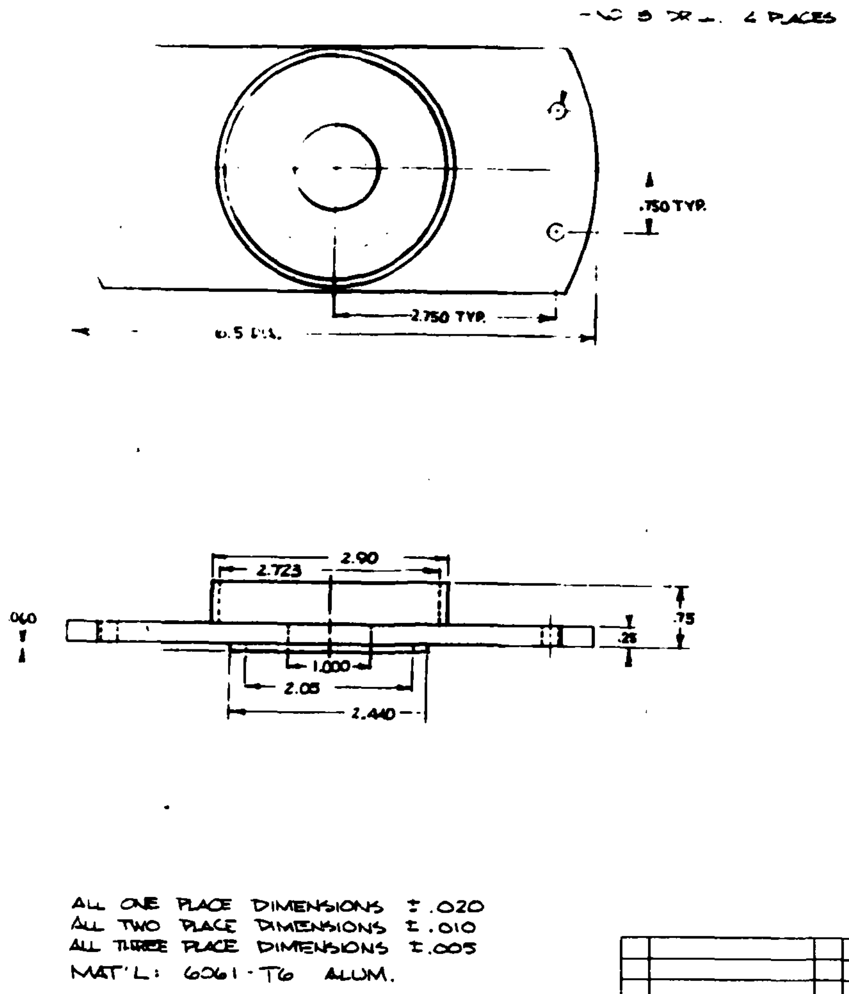
Drawing 17. Encoder Mounting Shaft - Azimuth Axis.

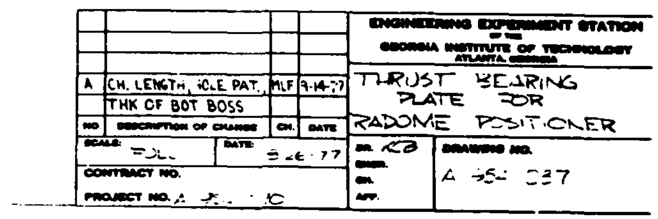


Drawing 18. Bearing Clamp - El. Axis

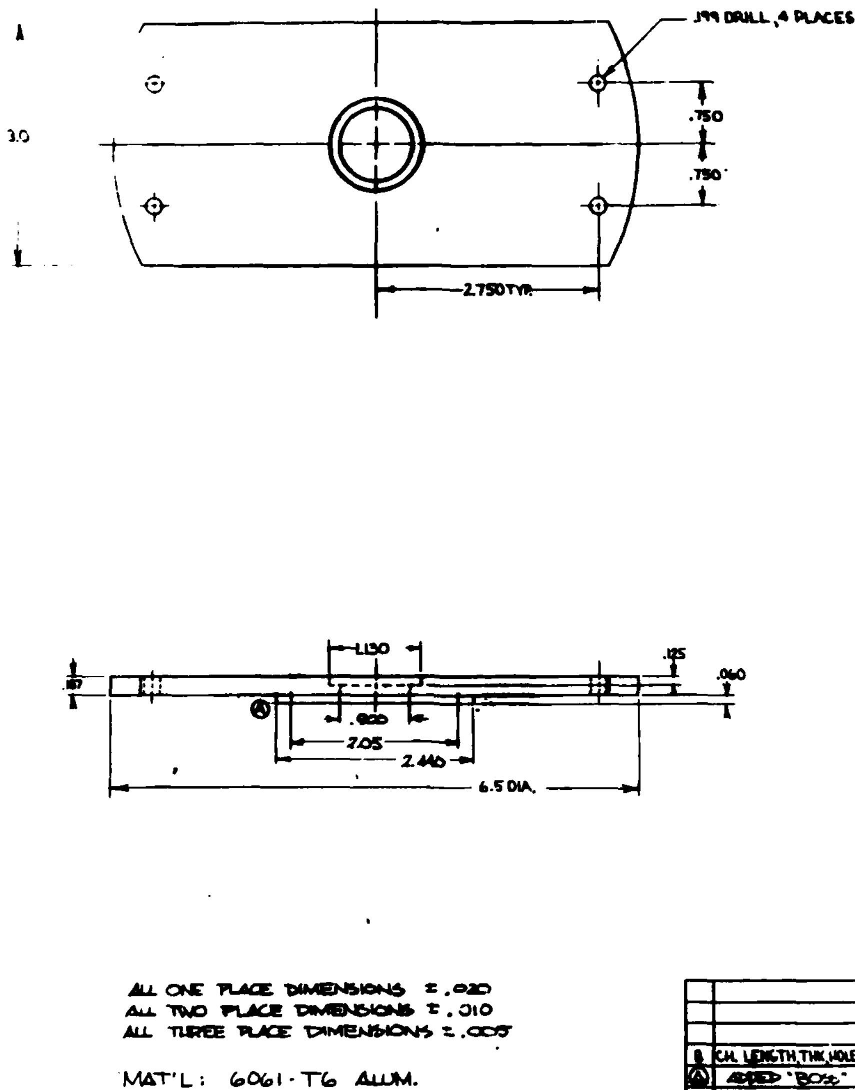


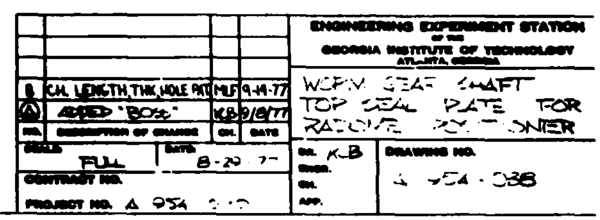
Drawing 19. Worm Gear Shaft - Azimuth Axis



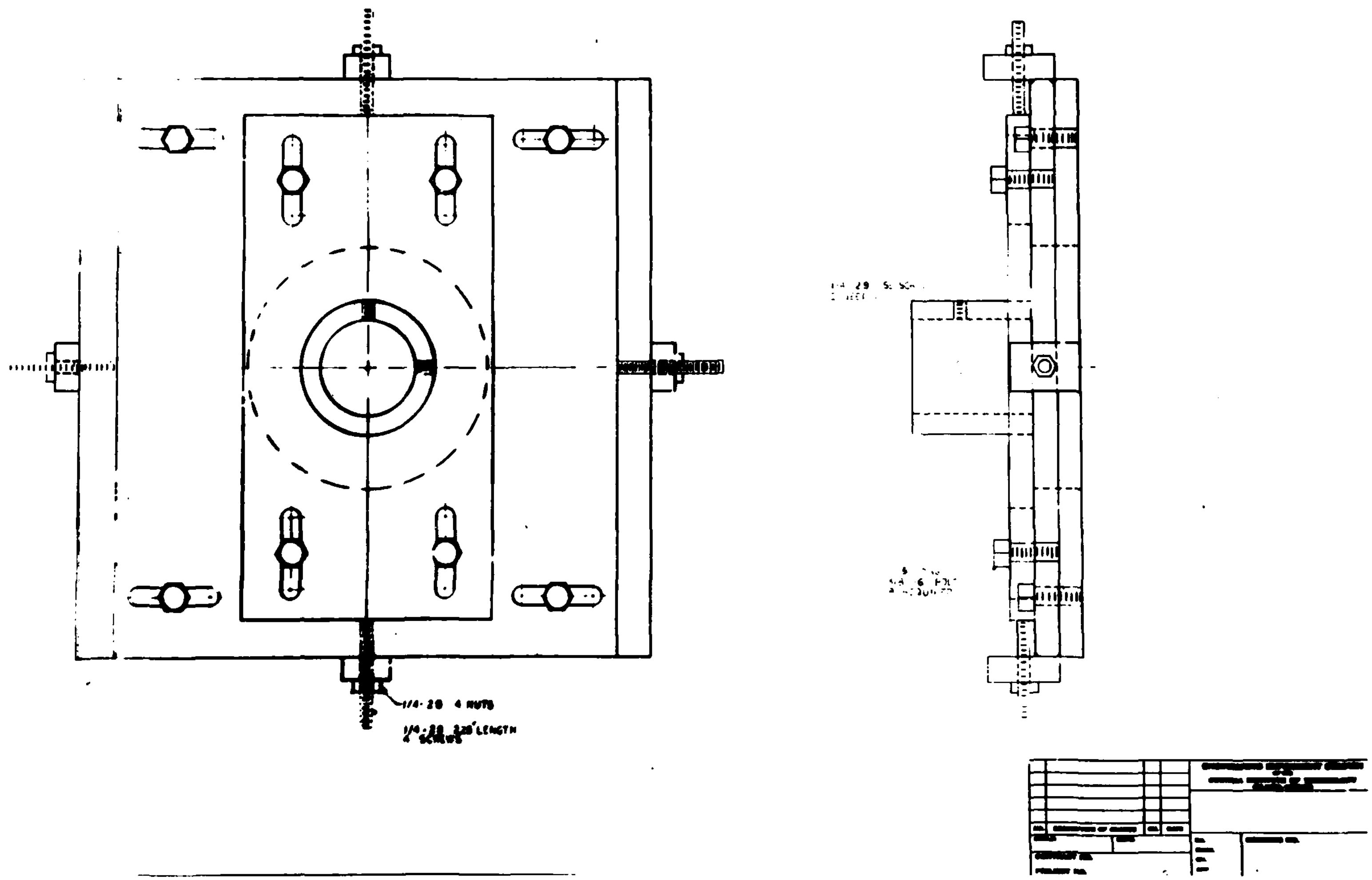


Drawing 20. Thrust Bearing Plate - Azimuth Axis.

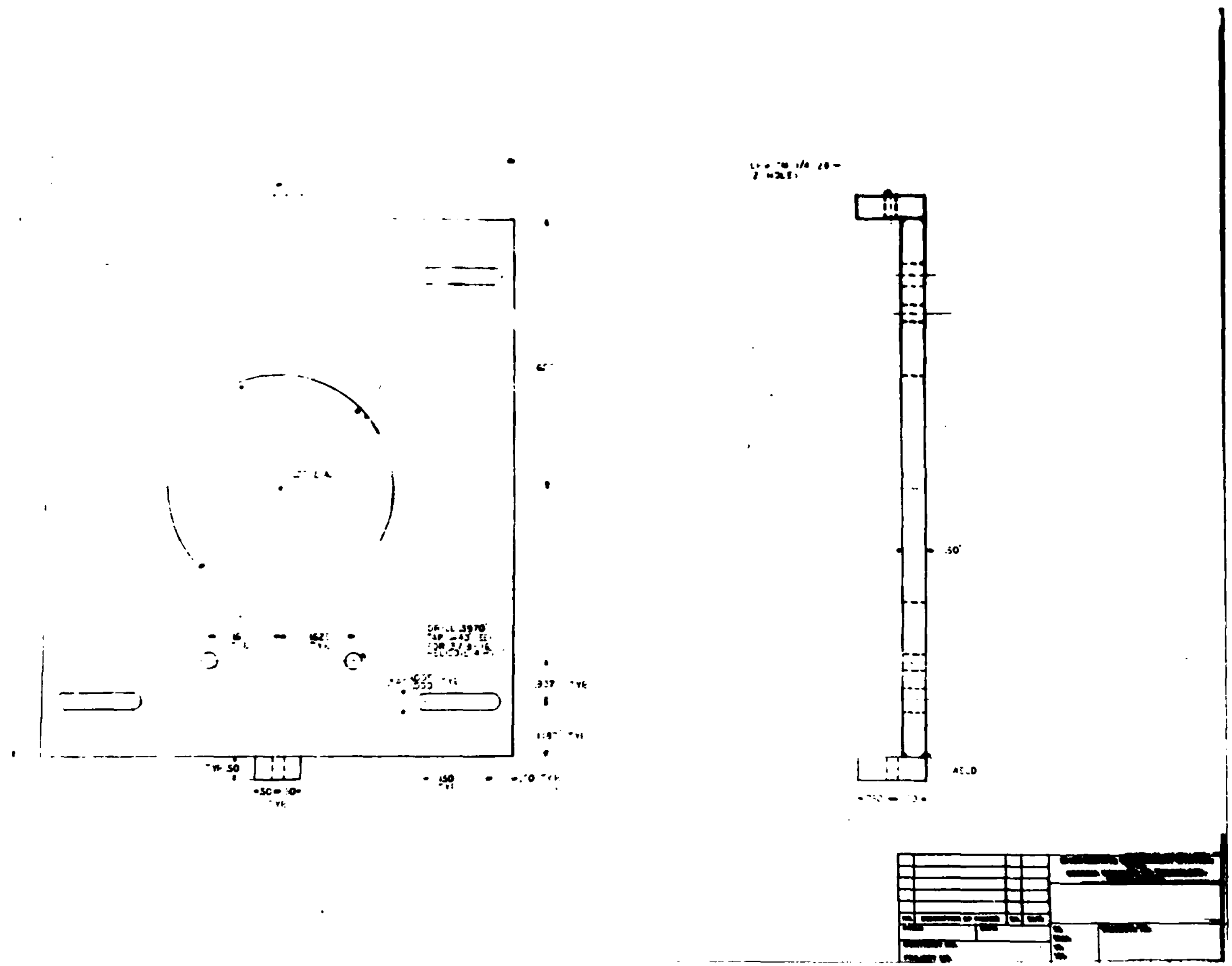




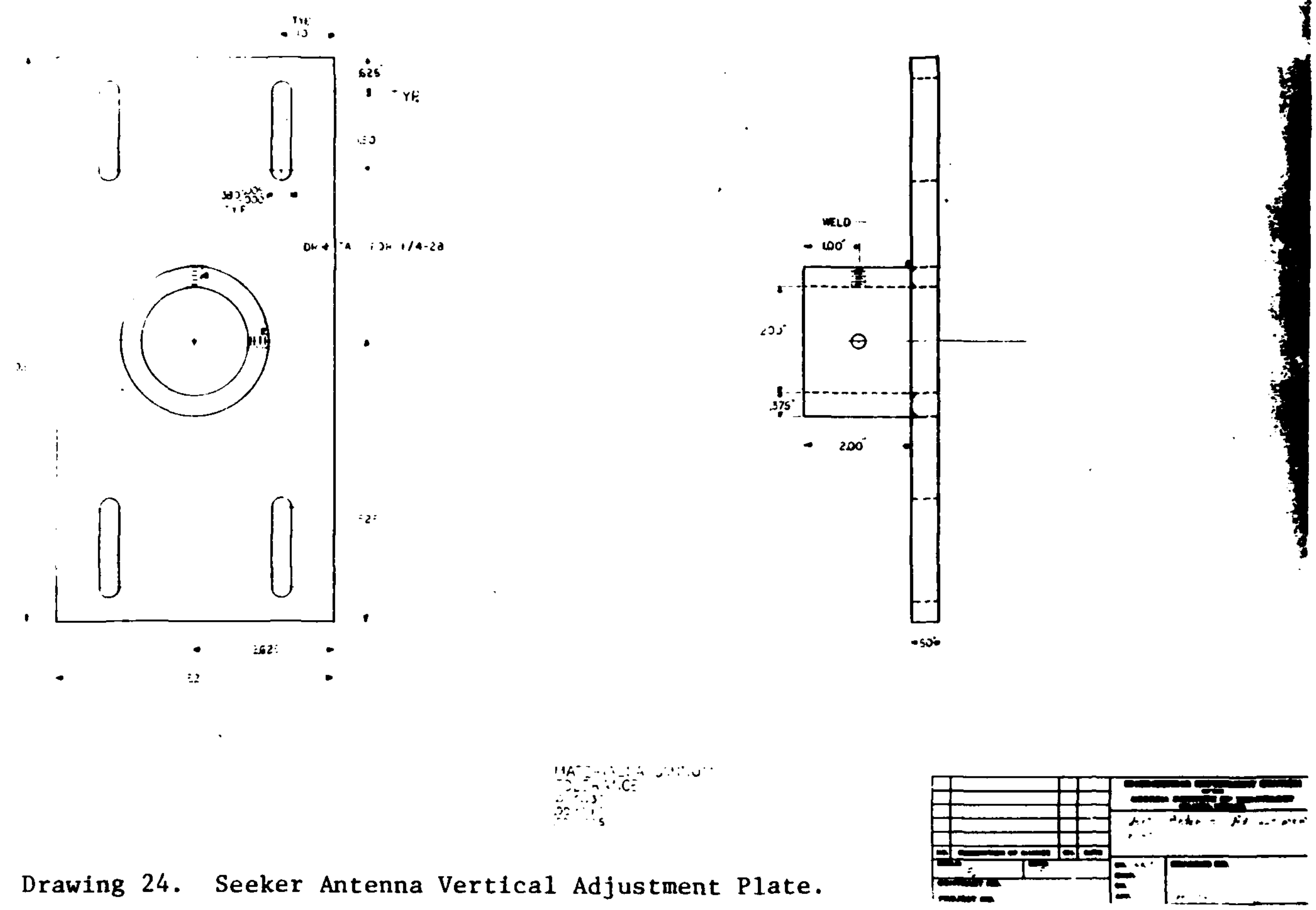
Drawing 21. Top Seal Plate Worm Gear Shaft - Azimuth Axis.



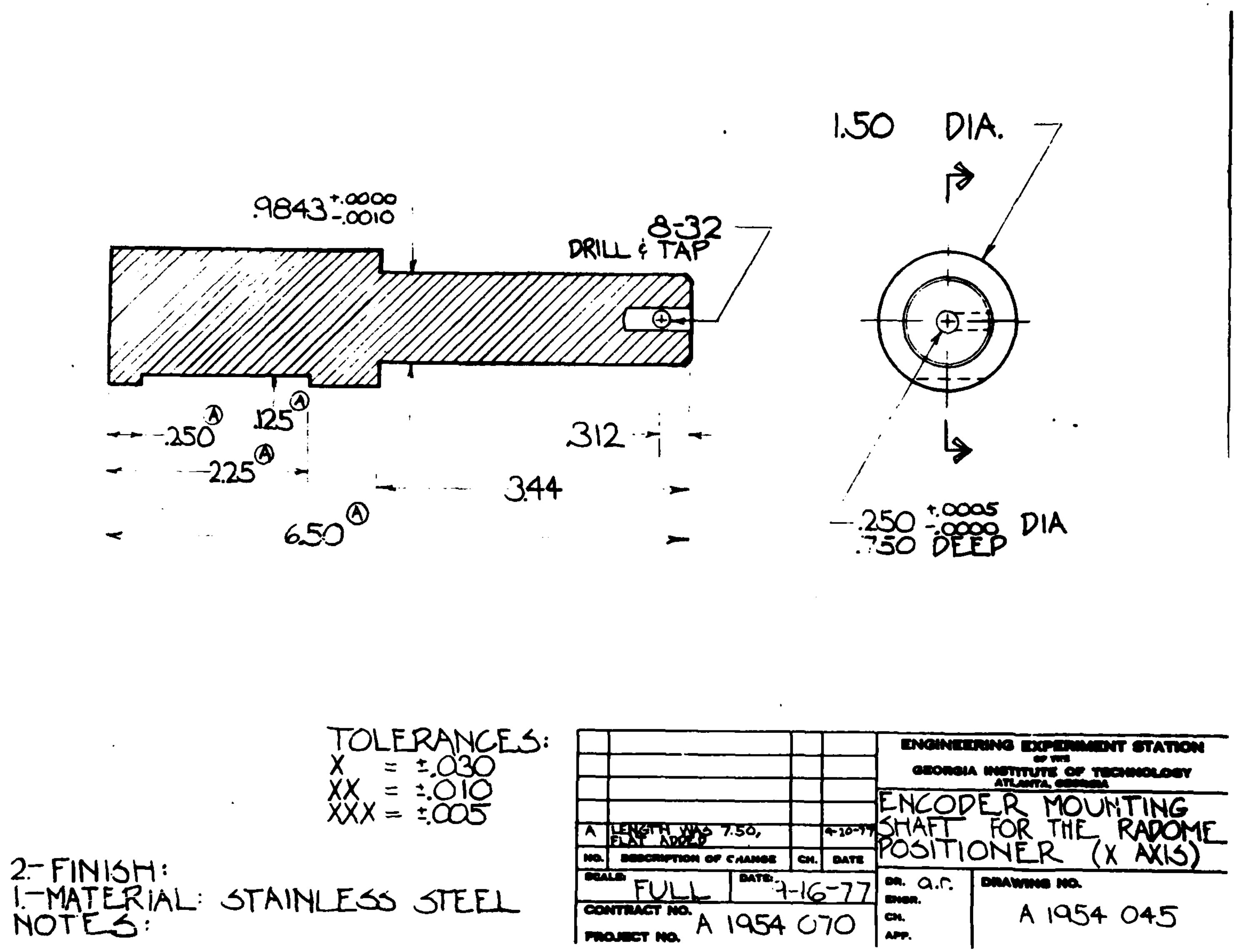
Drawing 22. Seeker Antenna Alignment Assembly



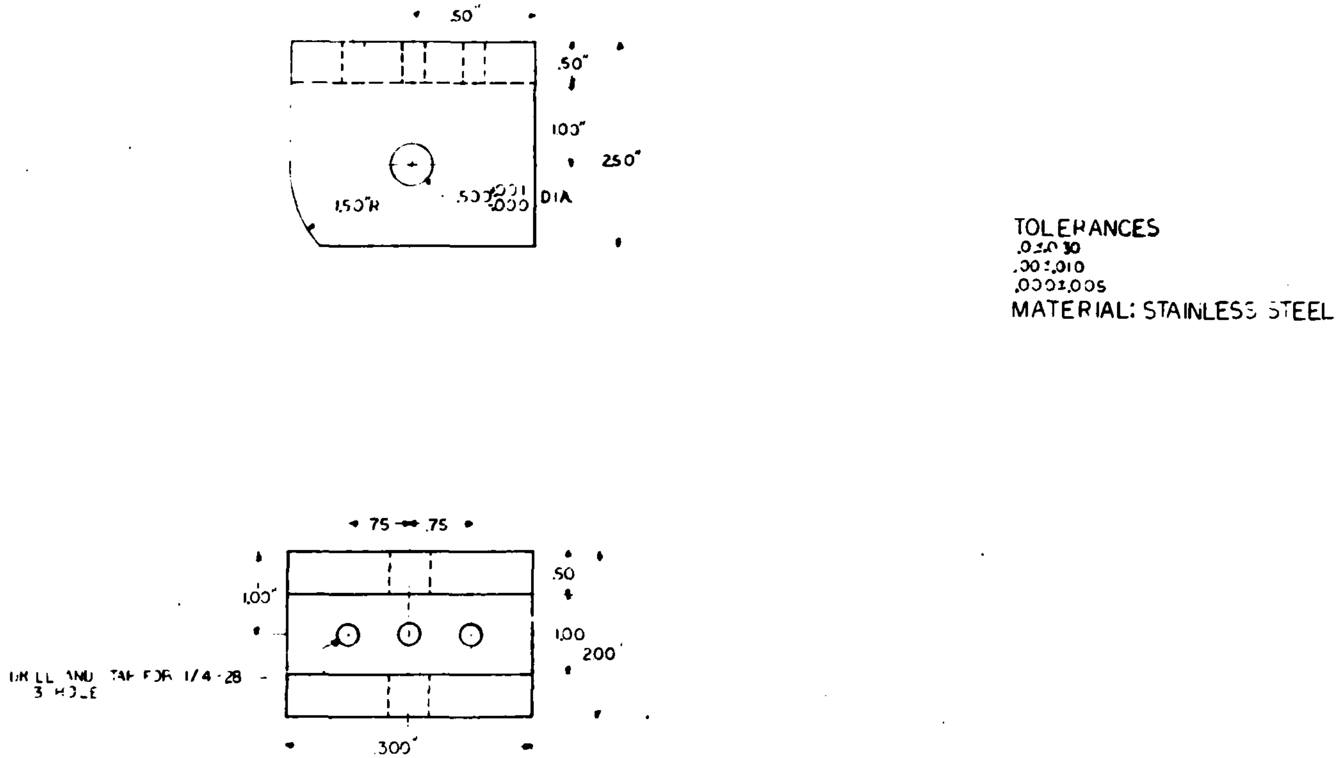
Drawing 23. Seeker Antenna Horizontal Adjustment Plate

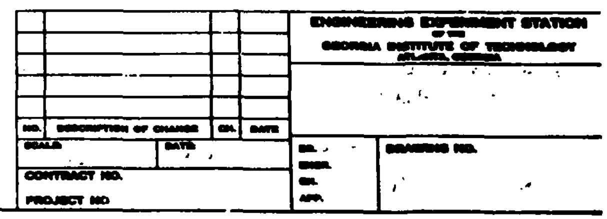


Drawing 24. Seeker Antenna Vertical Adjustment Plate.

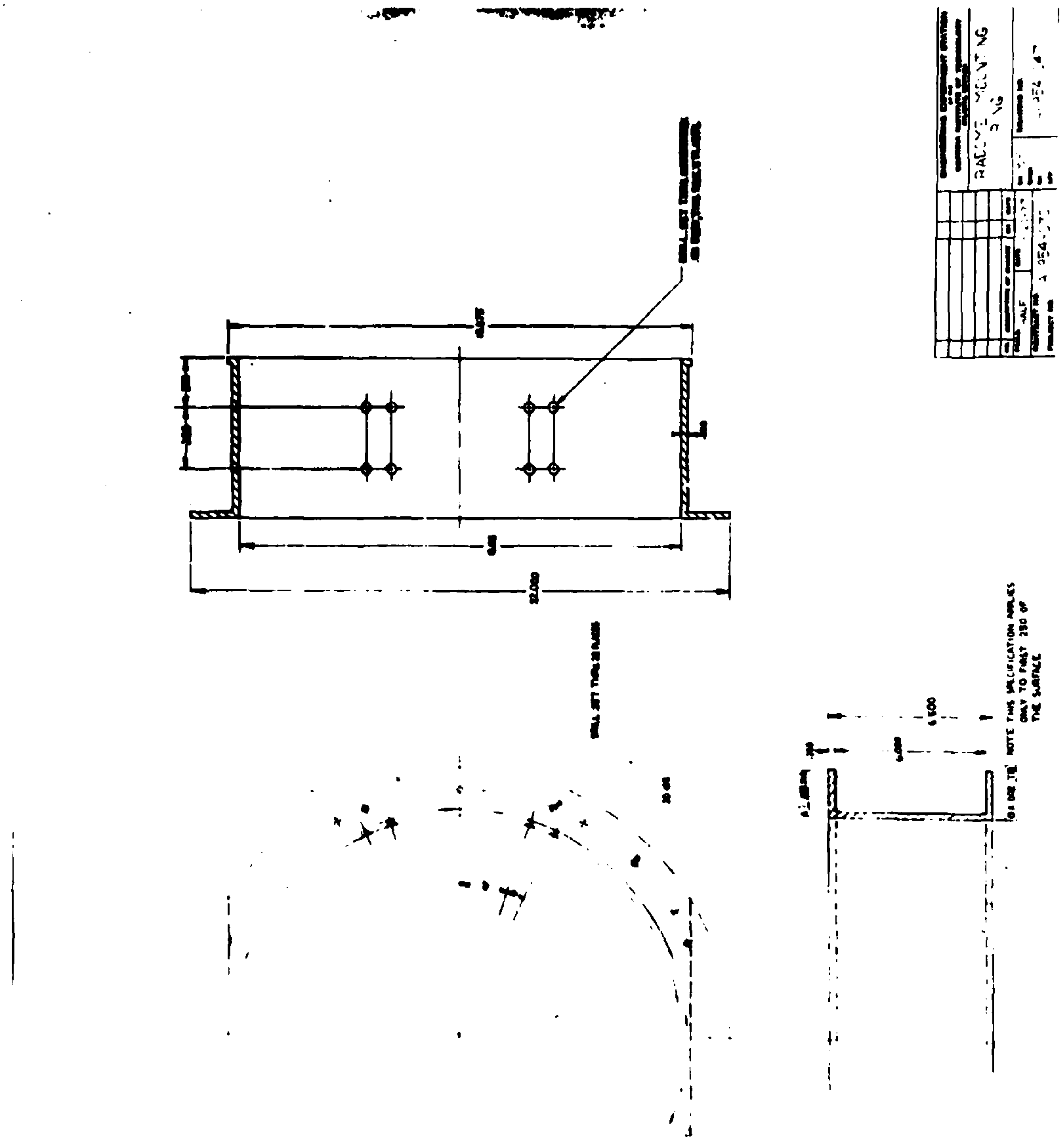


Drawing 25. Encoder Mounting Shaft - El. Axis

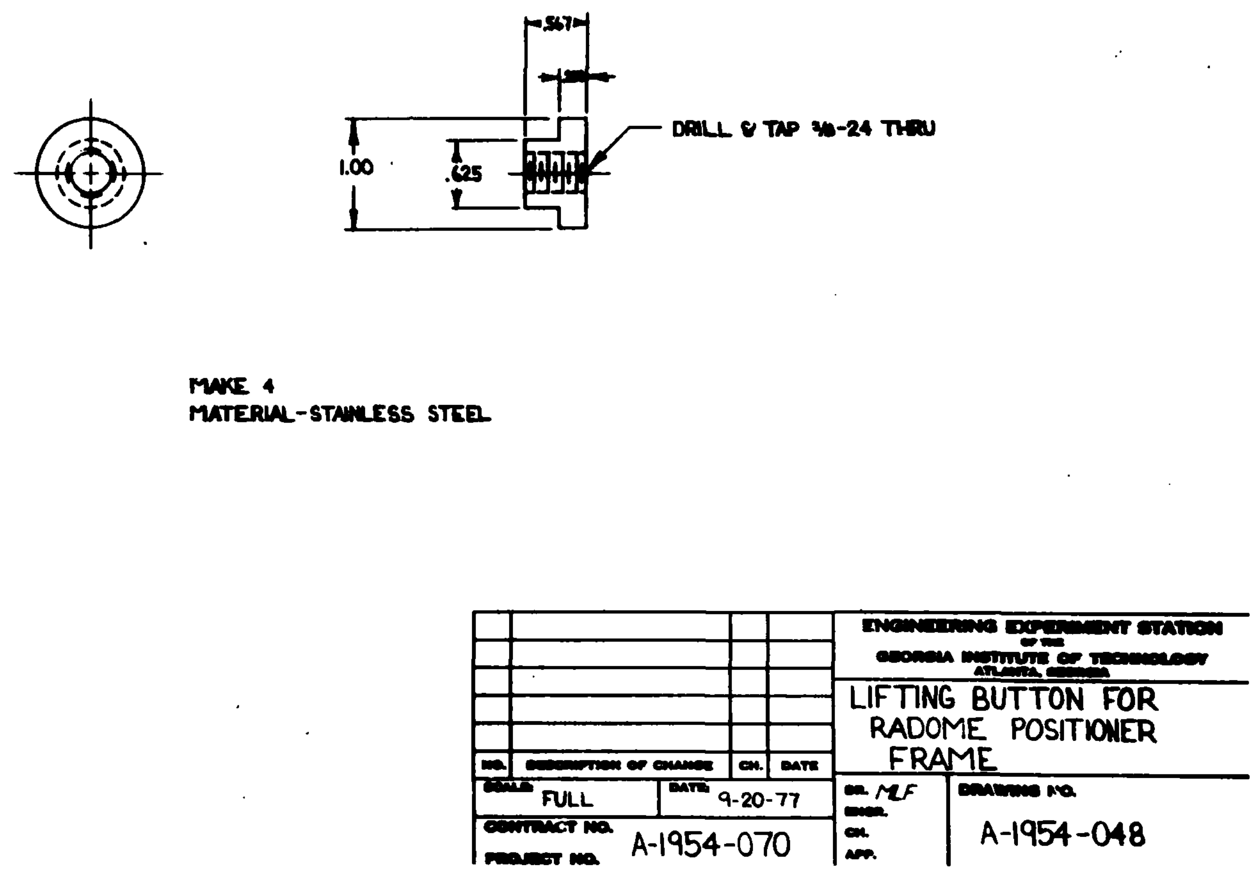




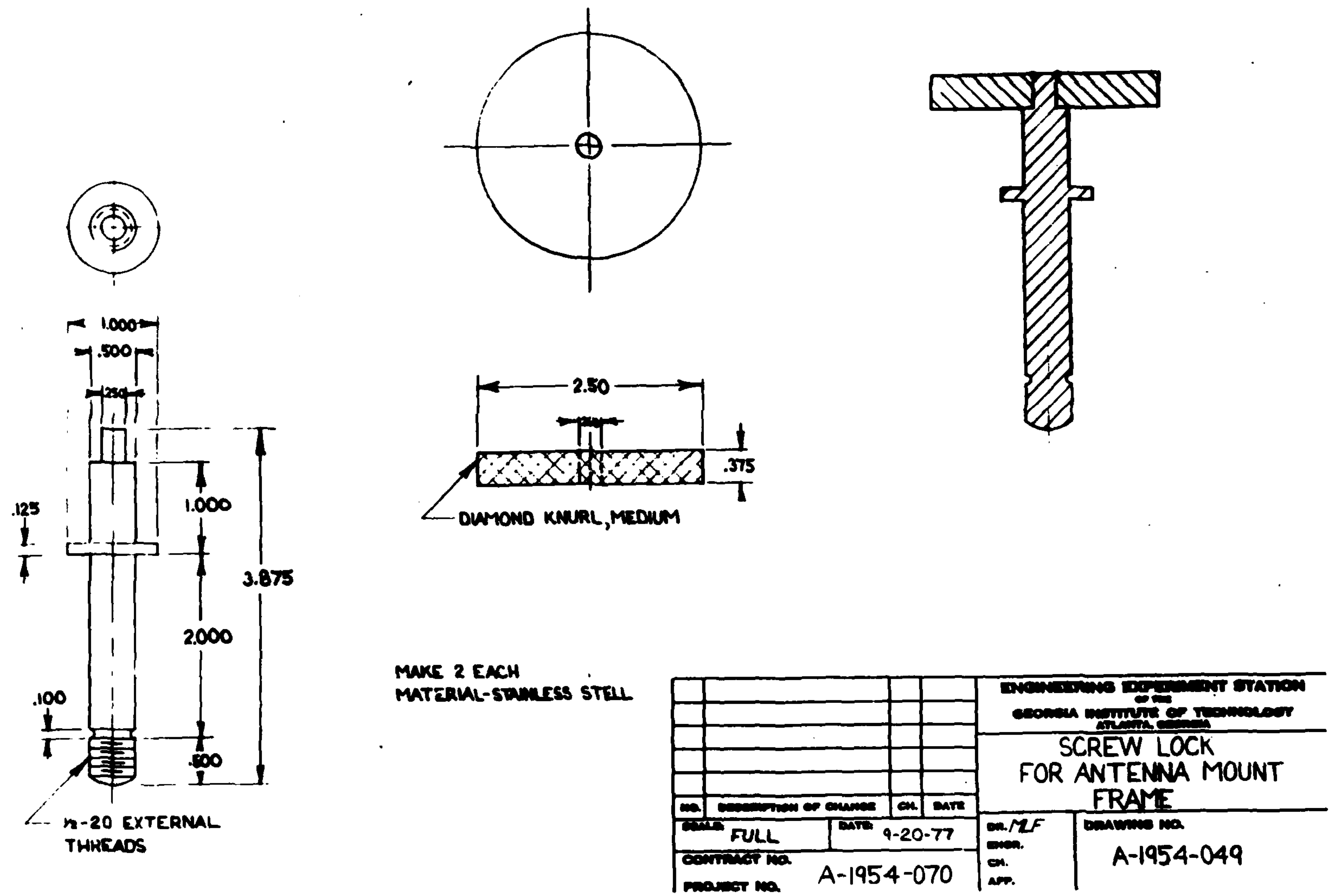
Drawing 26. Hinge for Seeker Antenna Mtg. Bracket.



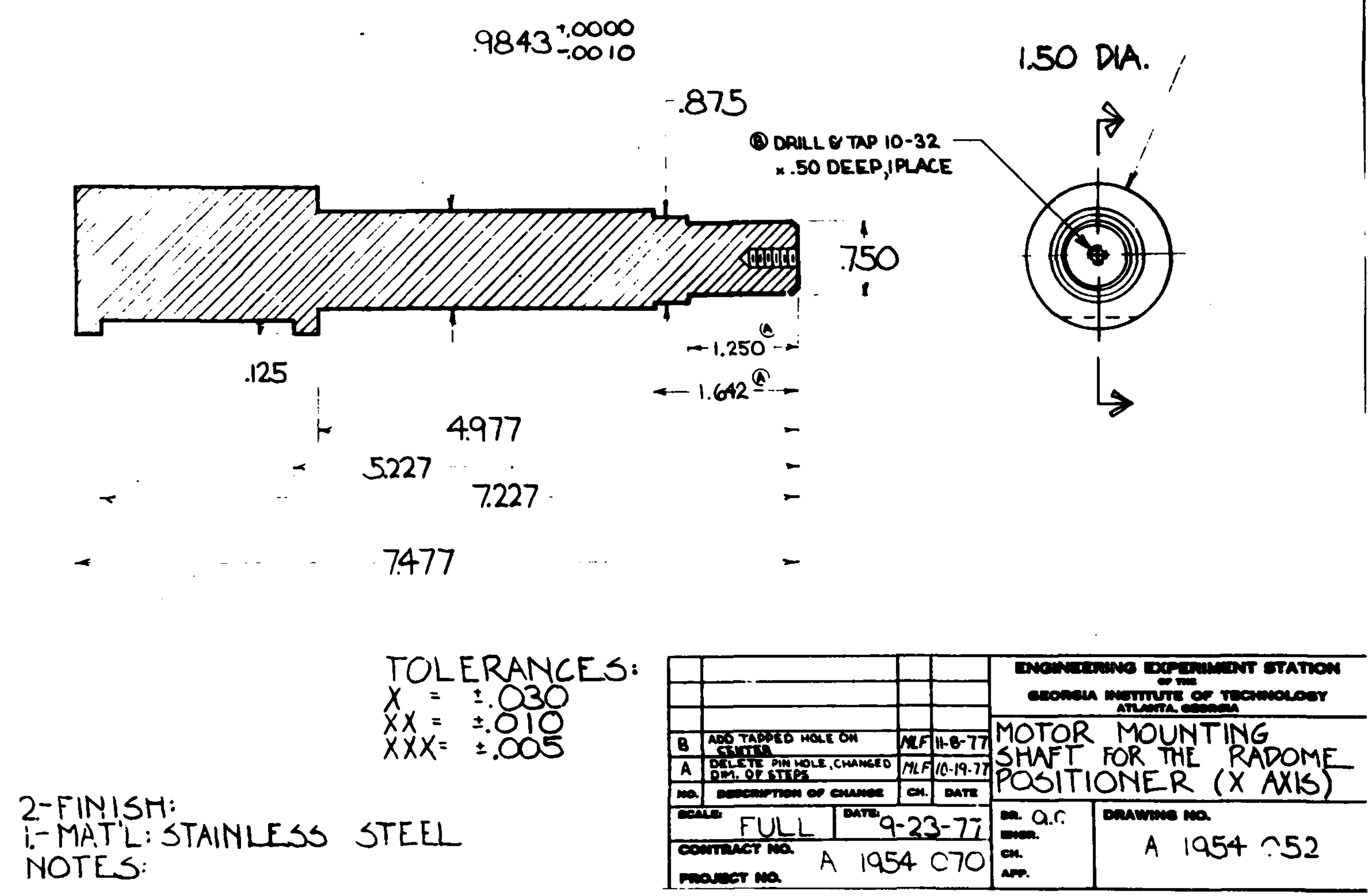
Drawing 27. Radome Mounting Ring



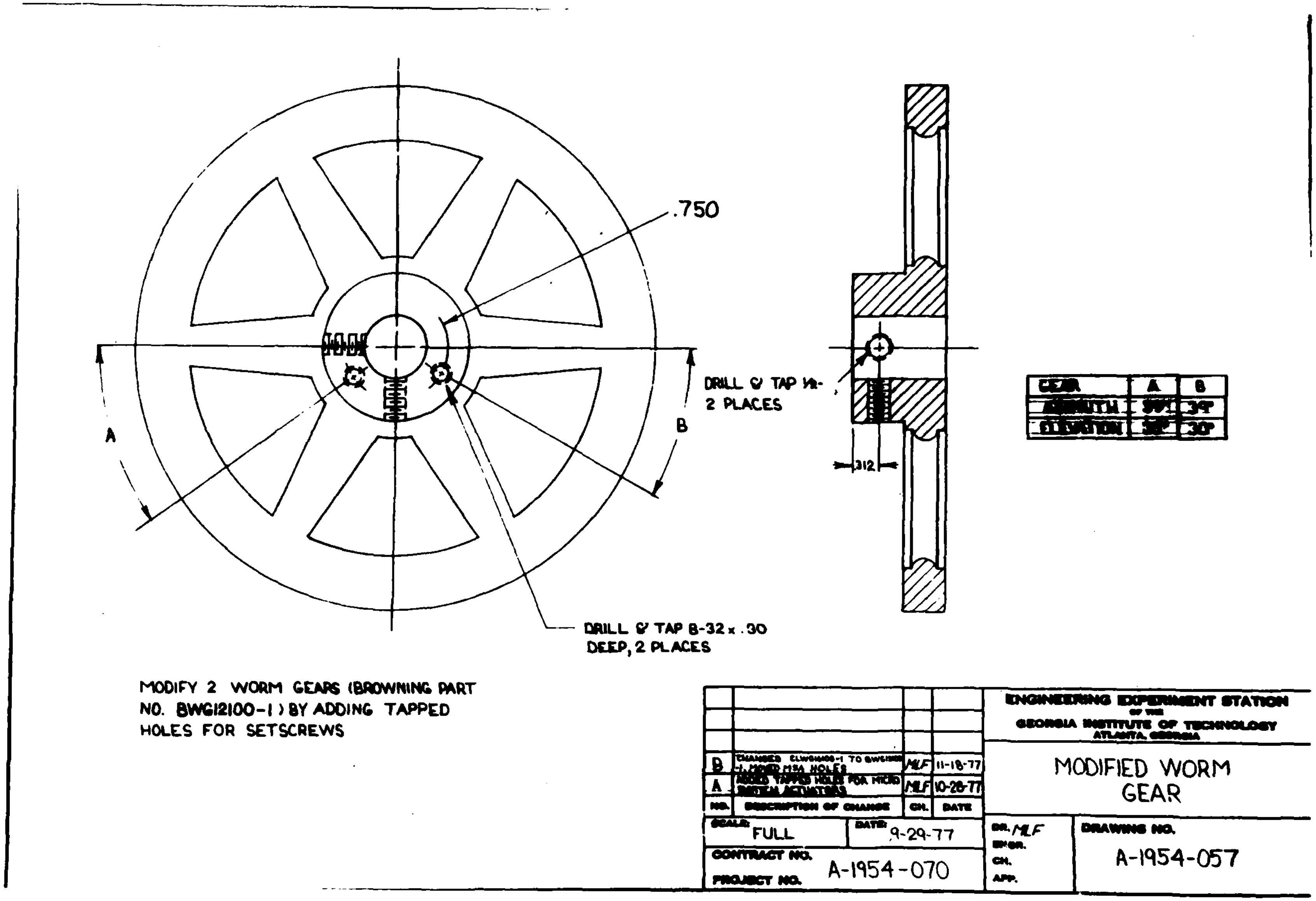
Drawing 28. Lifting Button for Outer Frame



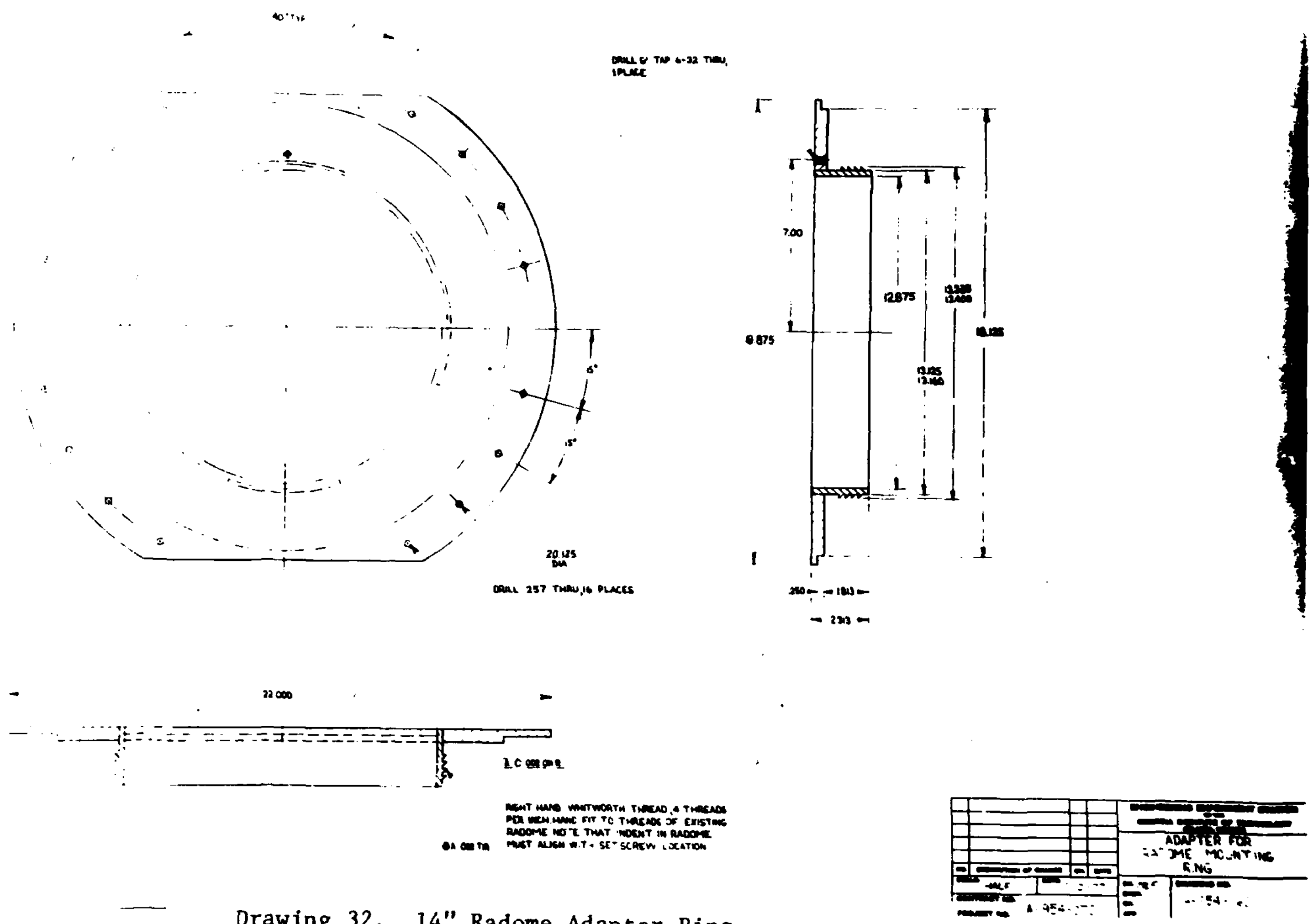
Drawing 29. Screw Lock for Seeker Antenna Mtg Frame



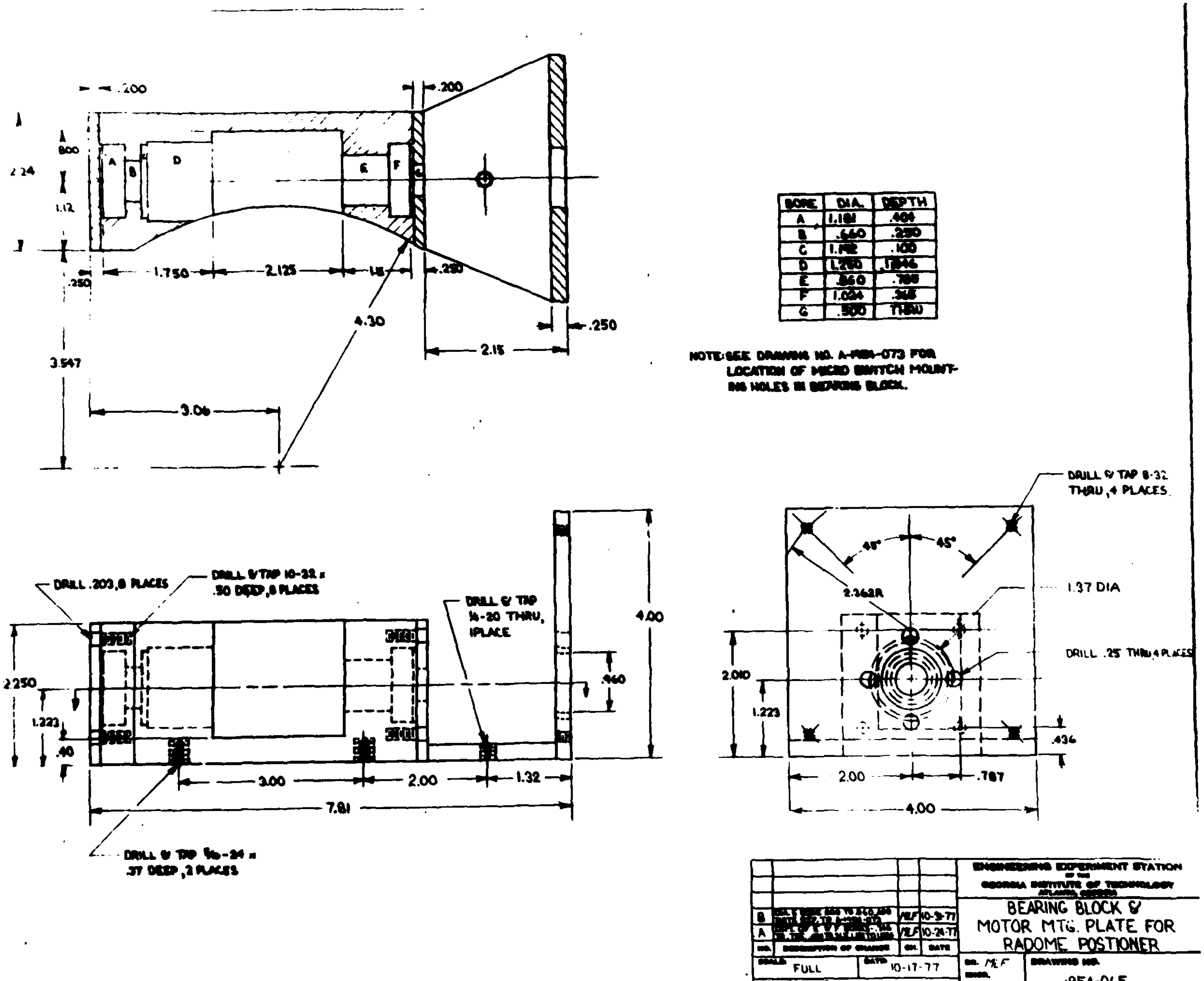
Drawing 30. Worm Gear Mounting Shaft - El. Axis



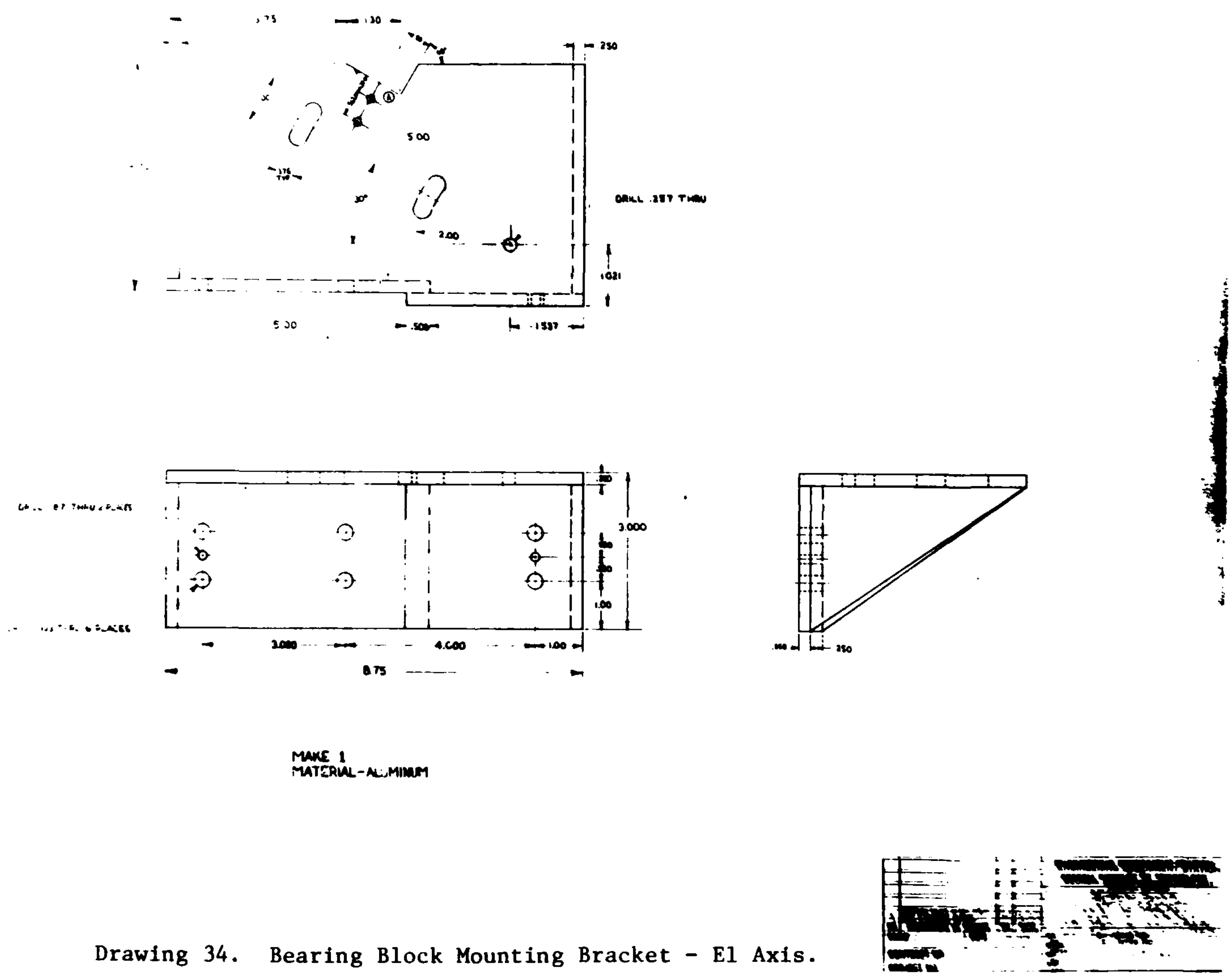
Drawing 31. Worm Gear Modifications



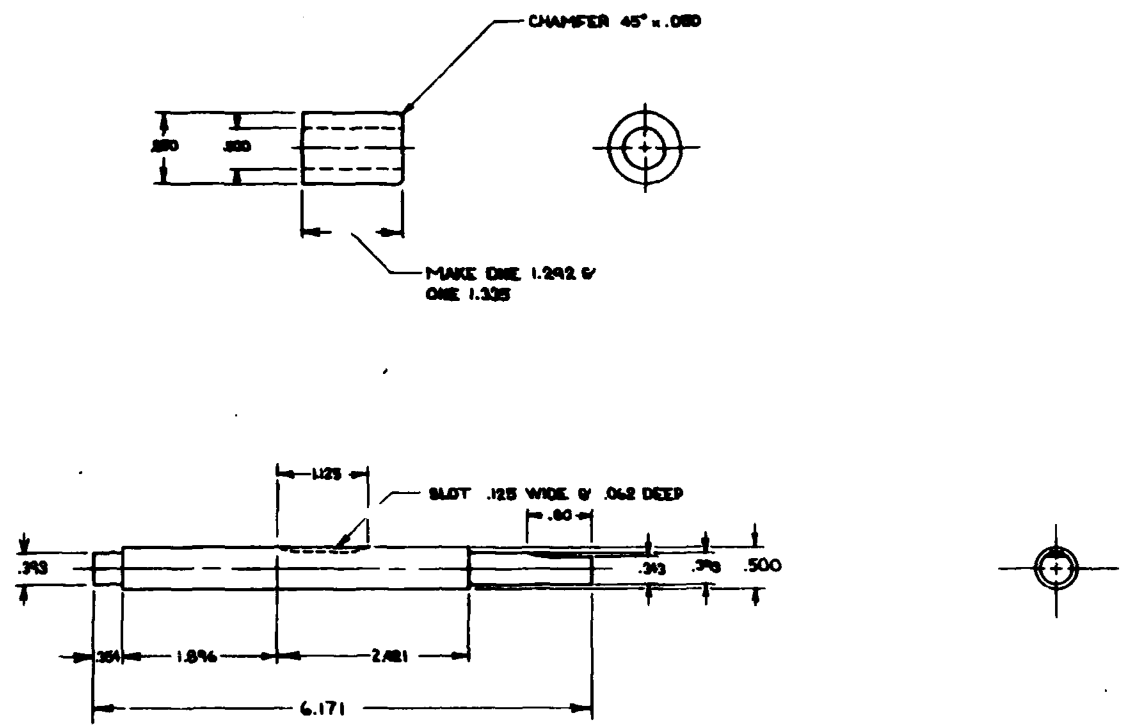
Drawing 32. 14" Radome Adapter Ring.



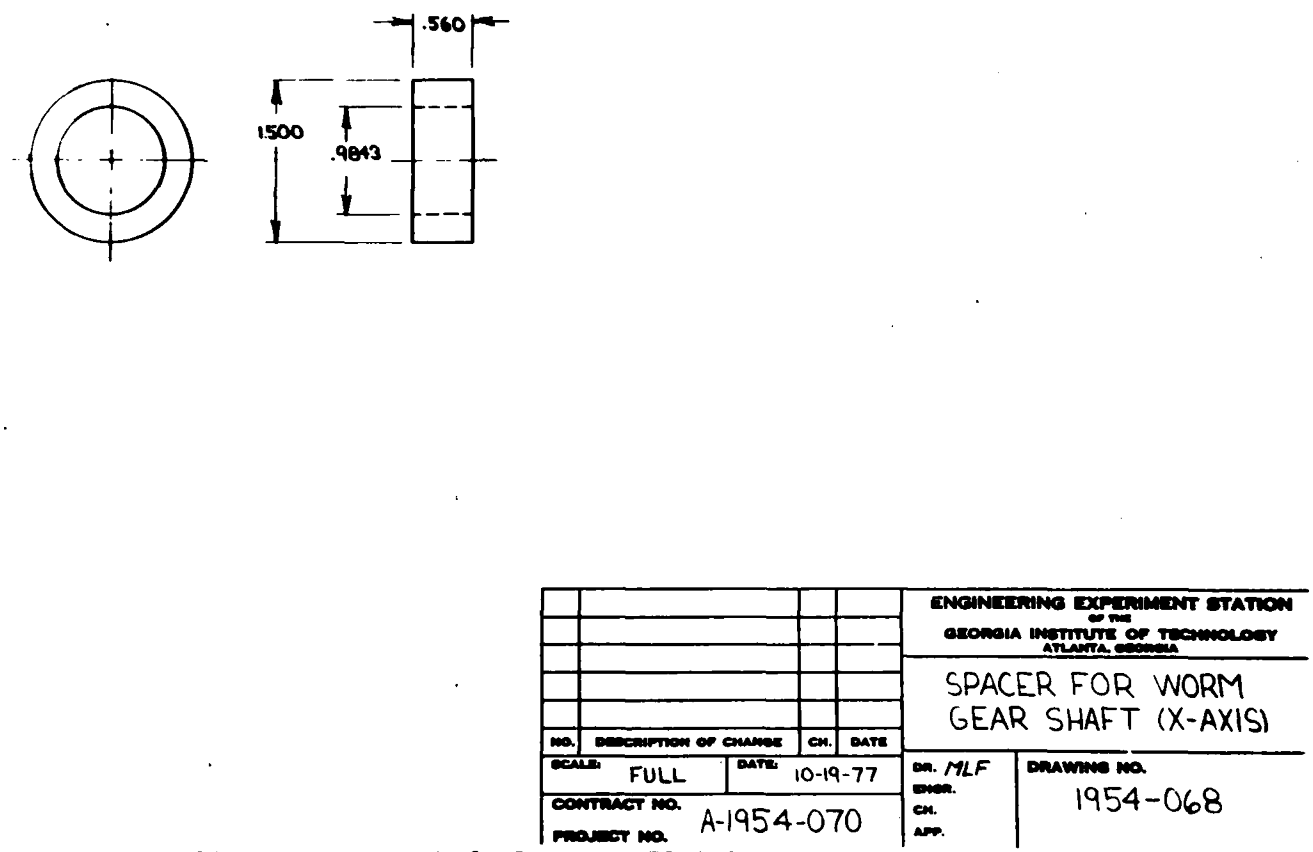
Drawing 33. Bearing Block and Motor Mounting Plate - El Axis.



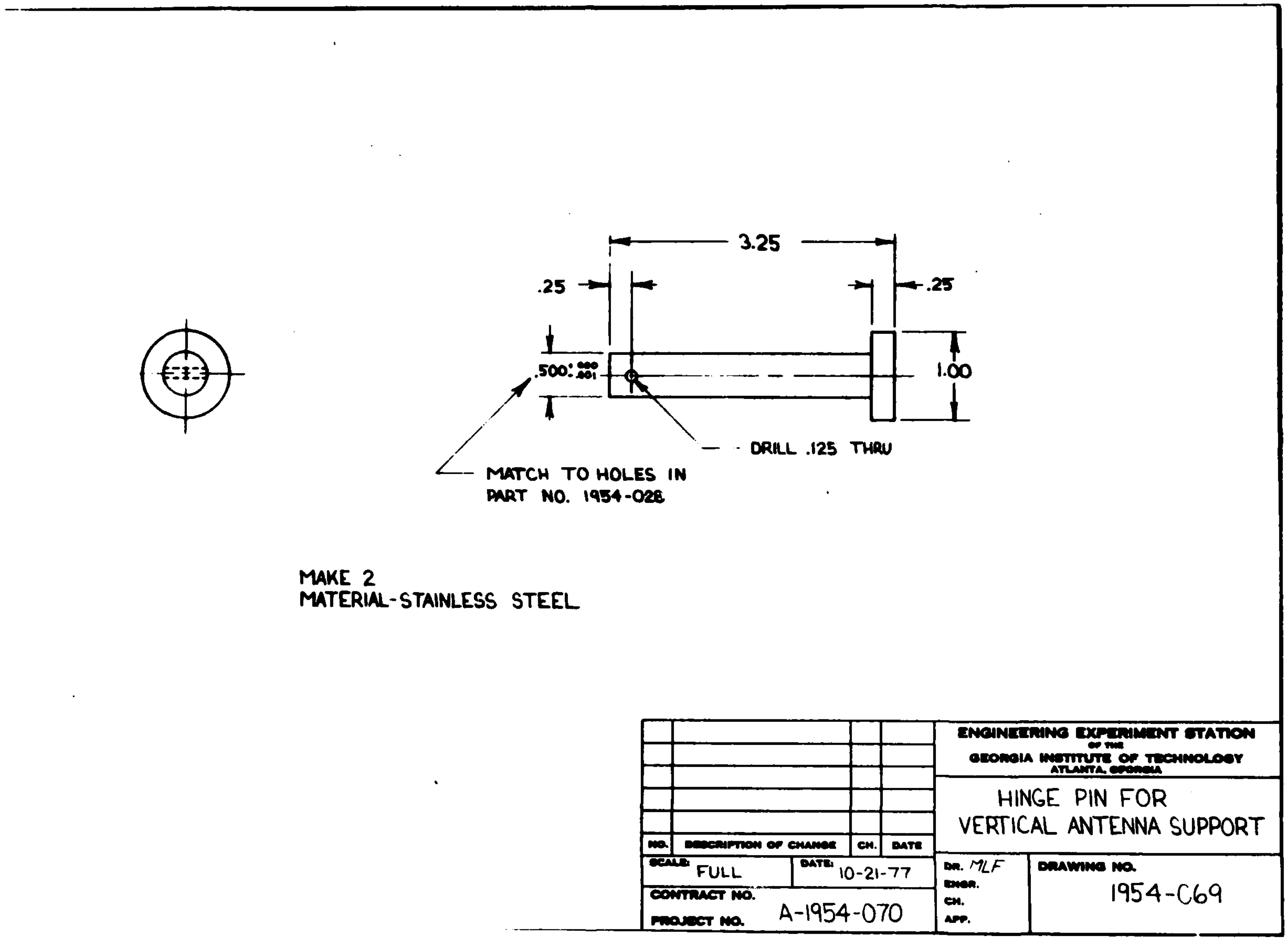
Drawing 34. Bearing Block Mounting Bracket - El Axis.



Drawing 35. Worm Shaft - El Axis.



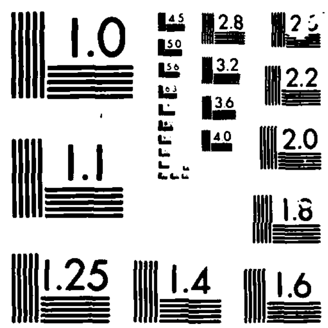
Drawing 36. Worm Gear Shaft Spacer - El Axis.



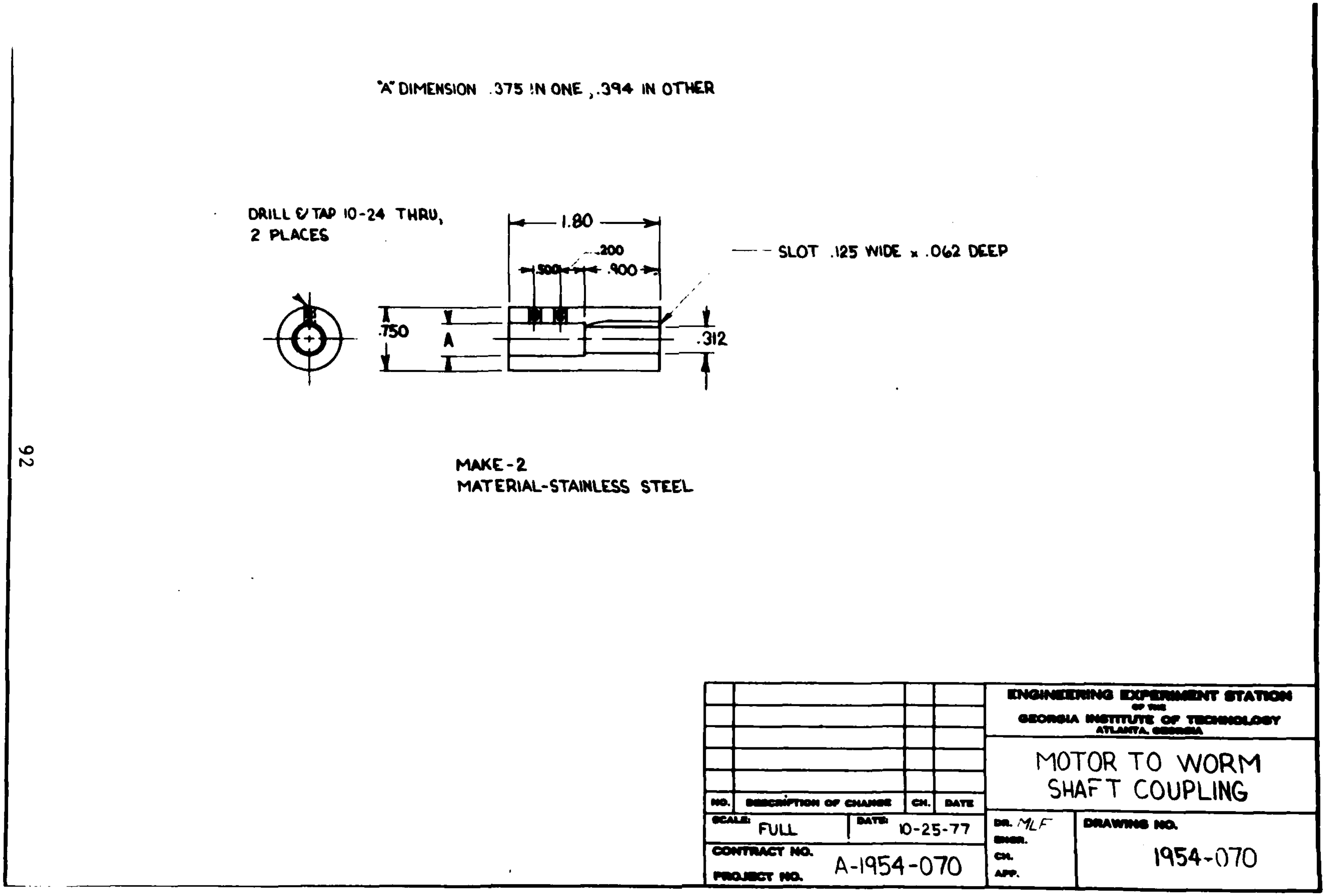
Drawing 37. Hinge Pin for Seeker Antenna Bracket

AD-A166 821 RADOME POSITIONER FOR THE RFSS (RADIO FREQUENCY SIMULATION SYSTEM)(U) GEORGIA INST OF TECH ATLANTA ENGINEERING EXPERIMENT STATION D O GALLENTINE ET AL. 2/3

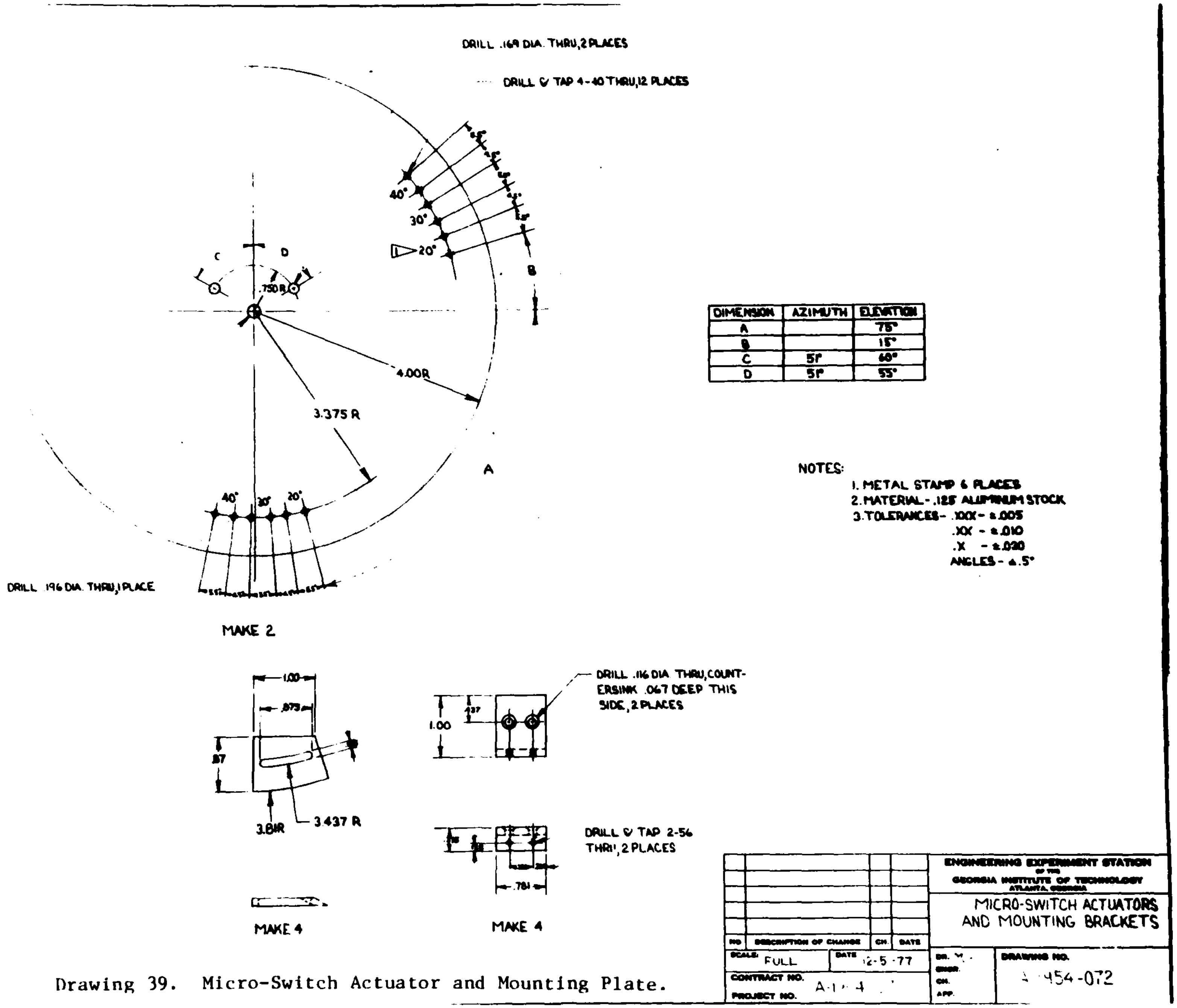
UNCLASSIFIED 27 FEB 78 DAAK40-77-C-0047 F/G 17/9 NL



MICROCOPY CHART

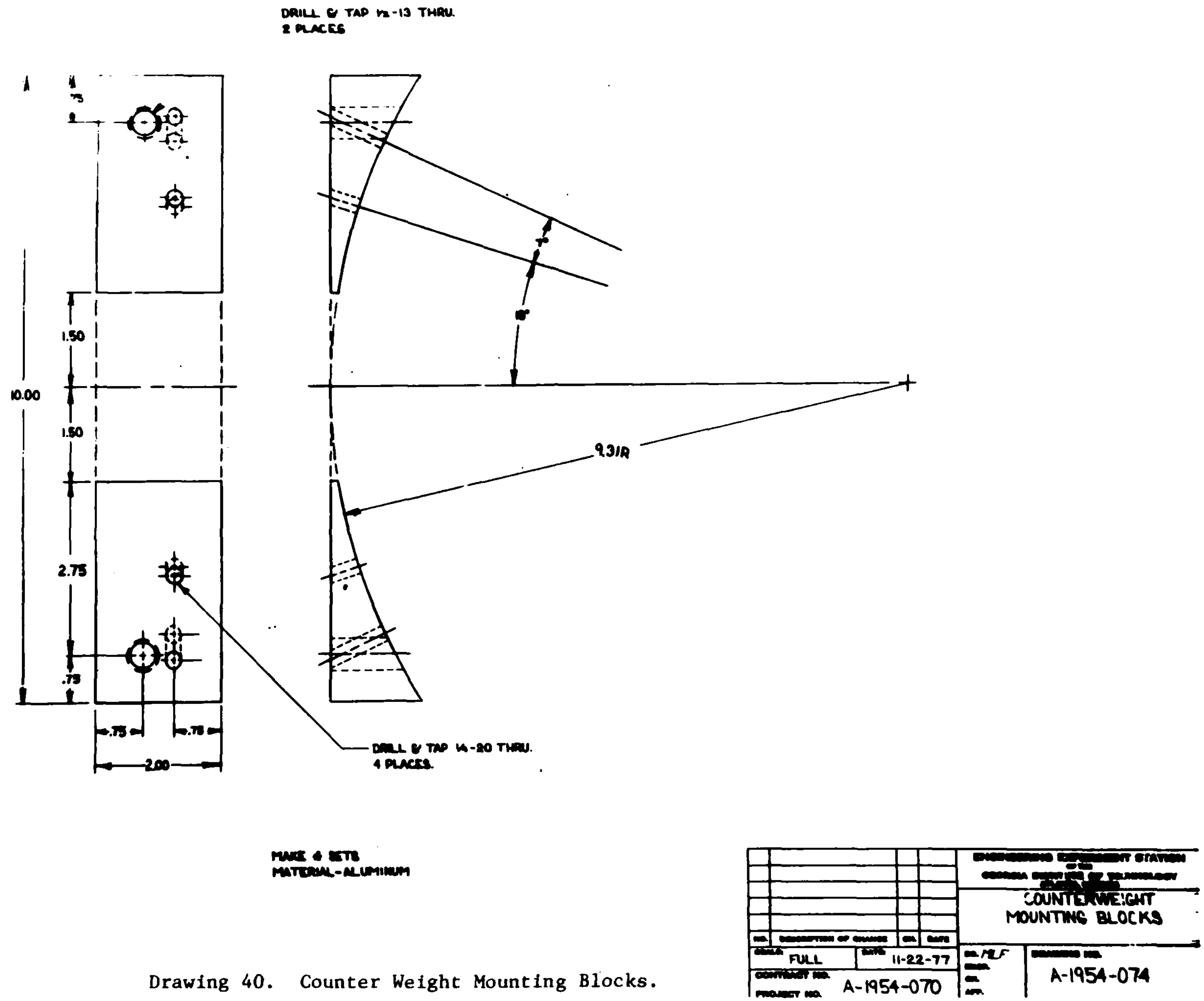


Drawing 38. Shaft Coupling Motor to Worm

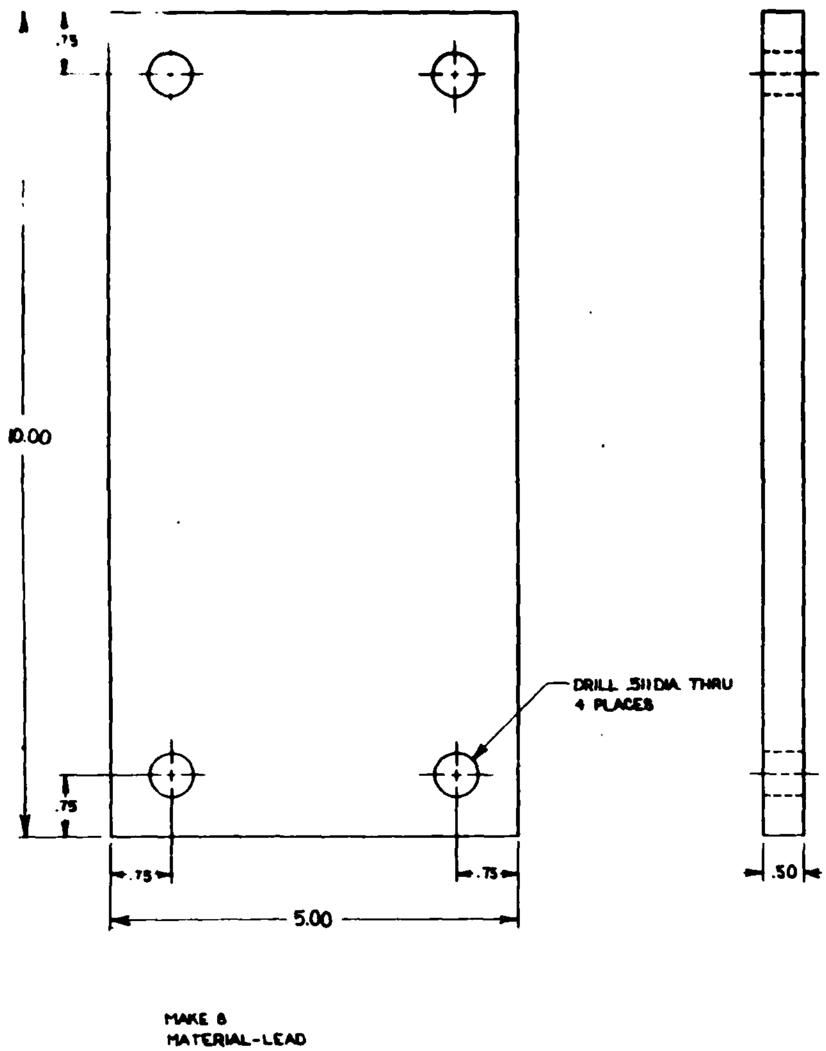


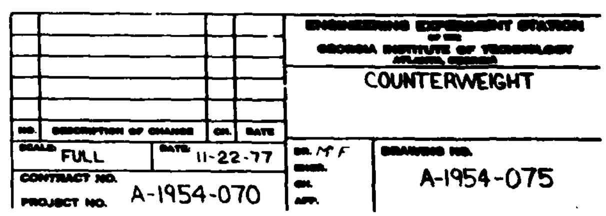
DIMENSION	AZIMUTH	ELEVATION
A		75°
B		15°
C	51°	60°
D	51°	55°

Drawing 39. Micro-Switch Actuator and Mounting Plate.

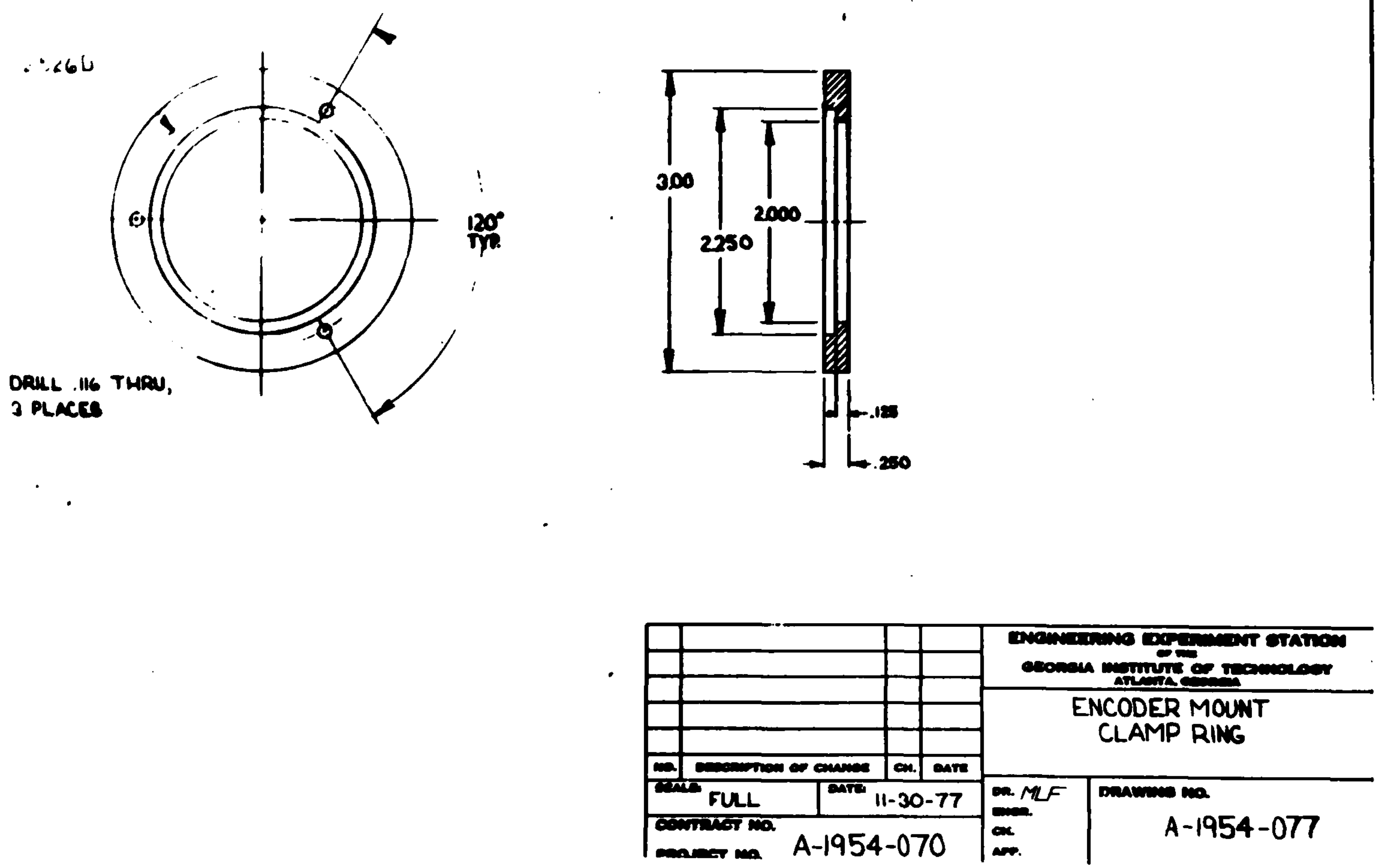


Drawing 40. Counter Weight Mounting Blocks.

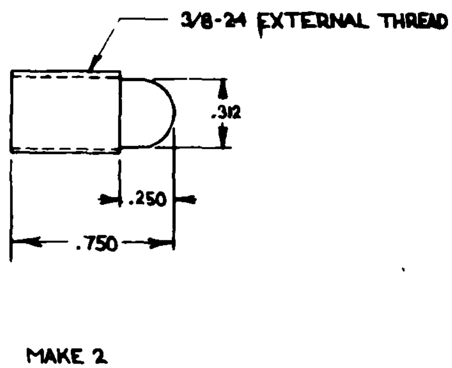




Drawing 41. Counter Weight.

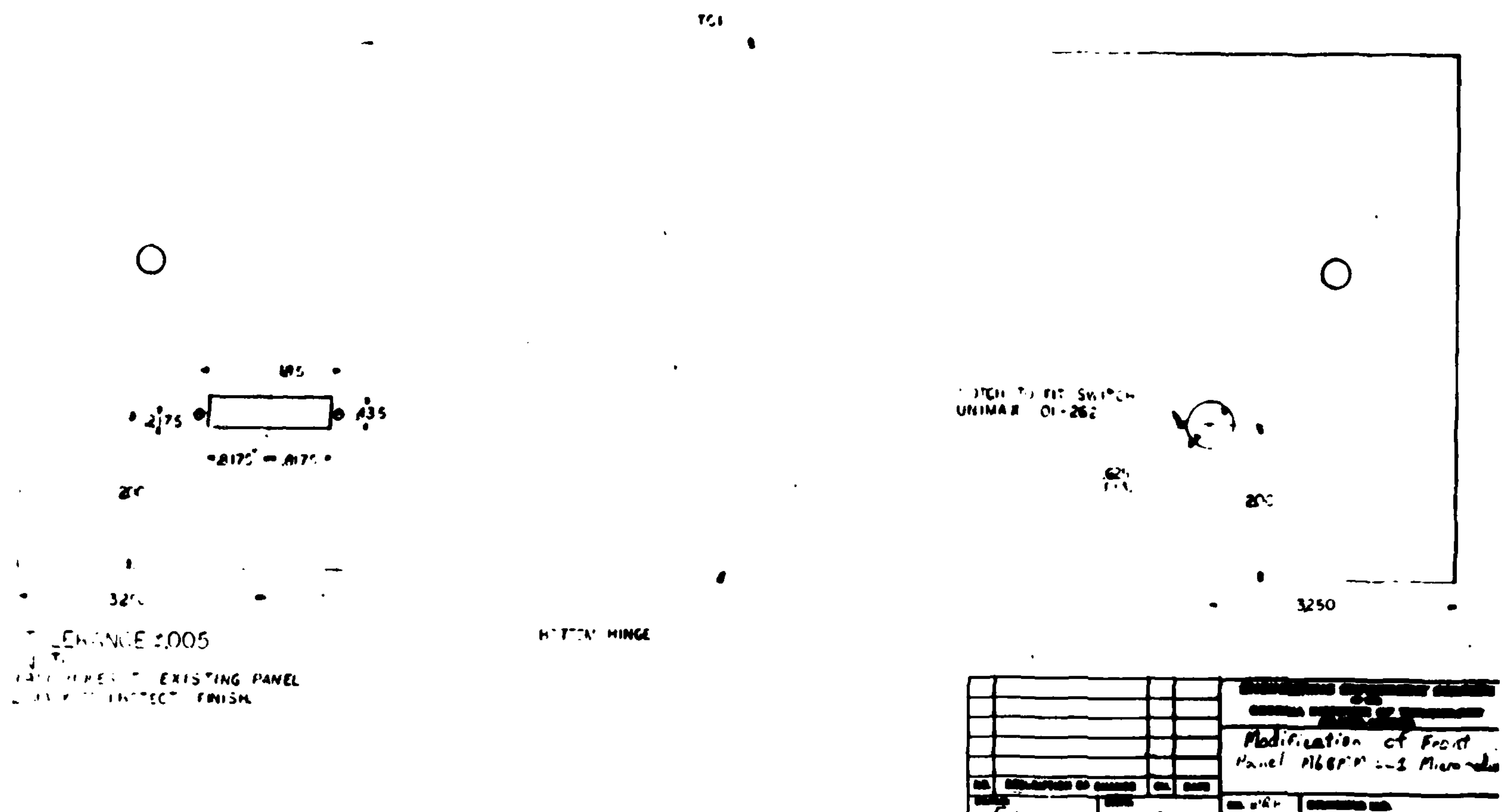


Drawing 42. Encoder Clamp Ring

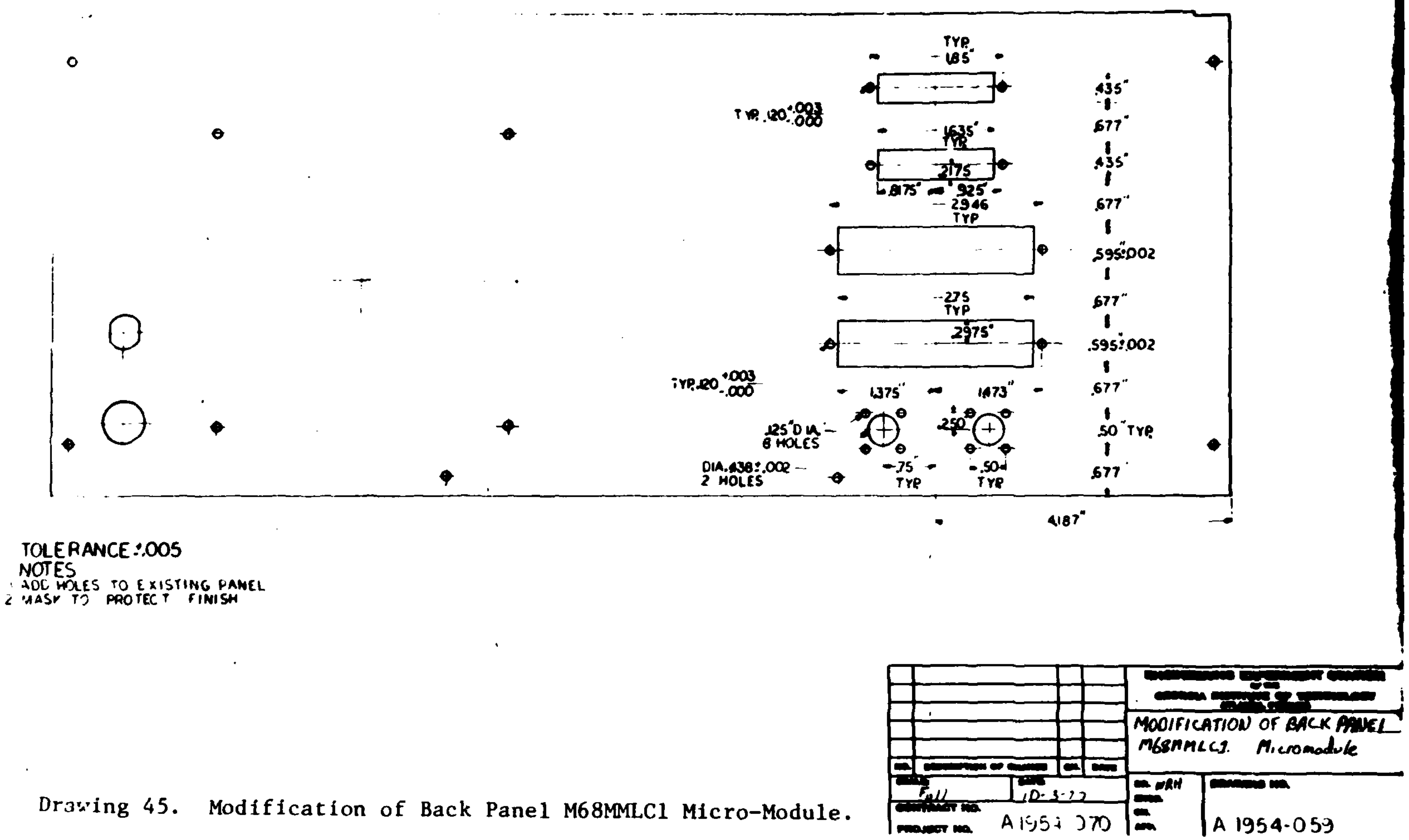


NO.	DESCRIPTION OF CHANGE	CH.	DATE	ENGINEERING EXPERIMENT STATION OF THE GEORGIA INSTITUTE OF TECHNOLOGY ATLANTA, GEORGIA	
				MODIFIED SETSCREW FOR HORIZONTAL POSITIONING OF INNER GIMBAL	
SCALE 2:1	DATE 12-7-77			DR. MLF ENGR. CH. APP.	DRAWING NO. A-1954-080
CONTRACT NO. PROJECT NO. A-1954-070					

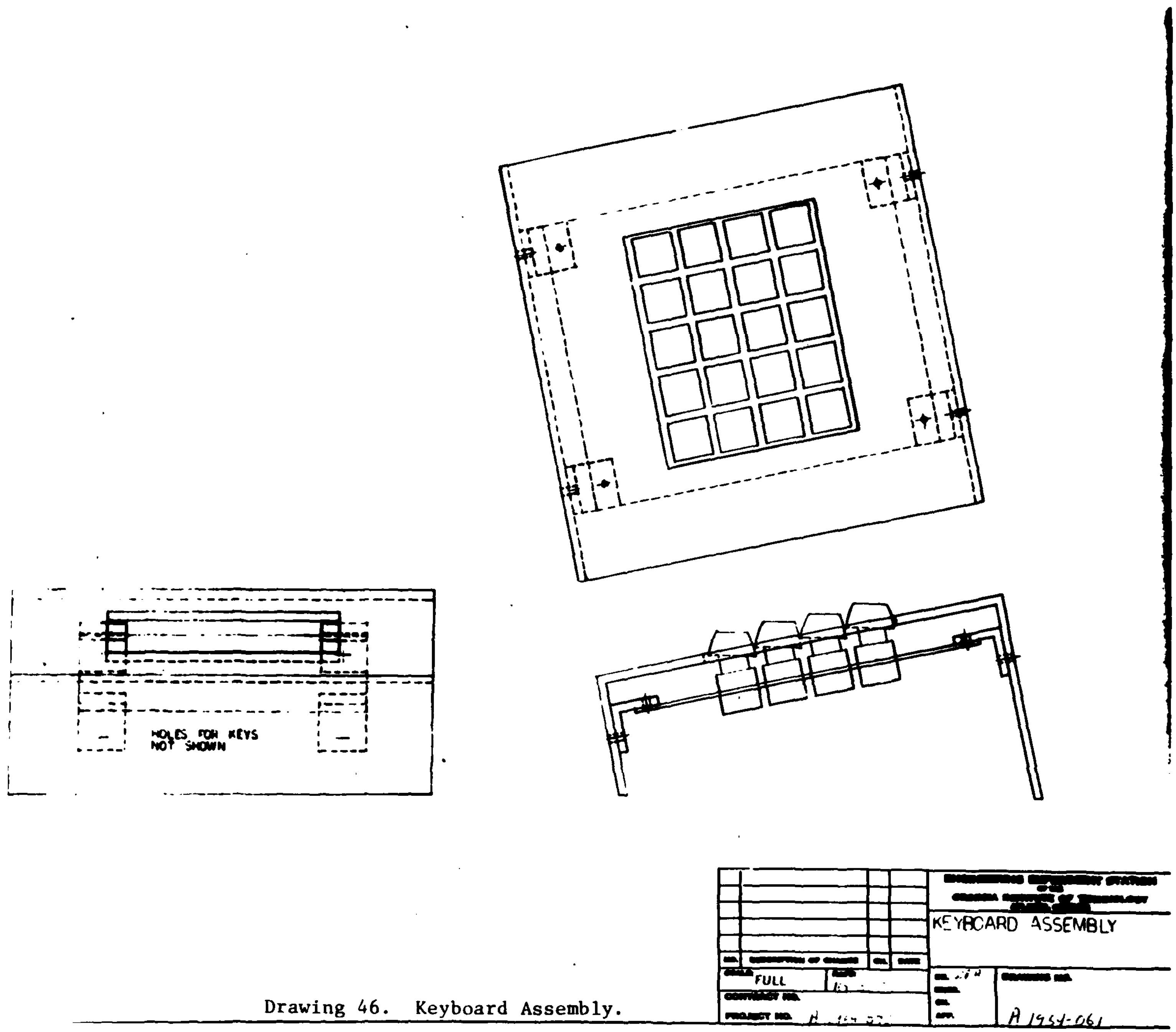
Drawing 43. Modified Set Screw for Centering of Inner Gimbal



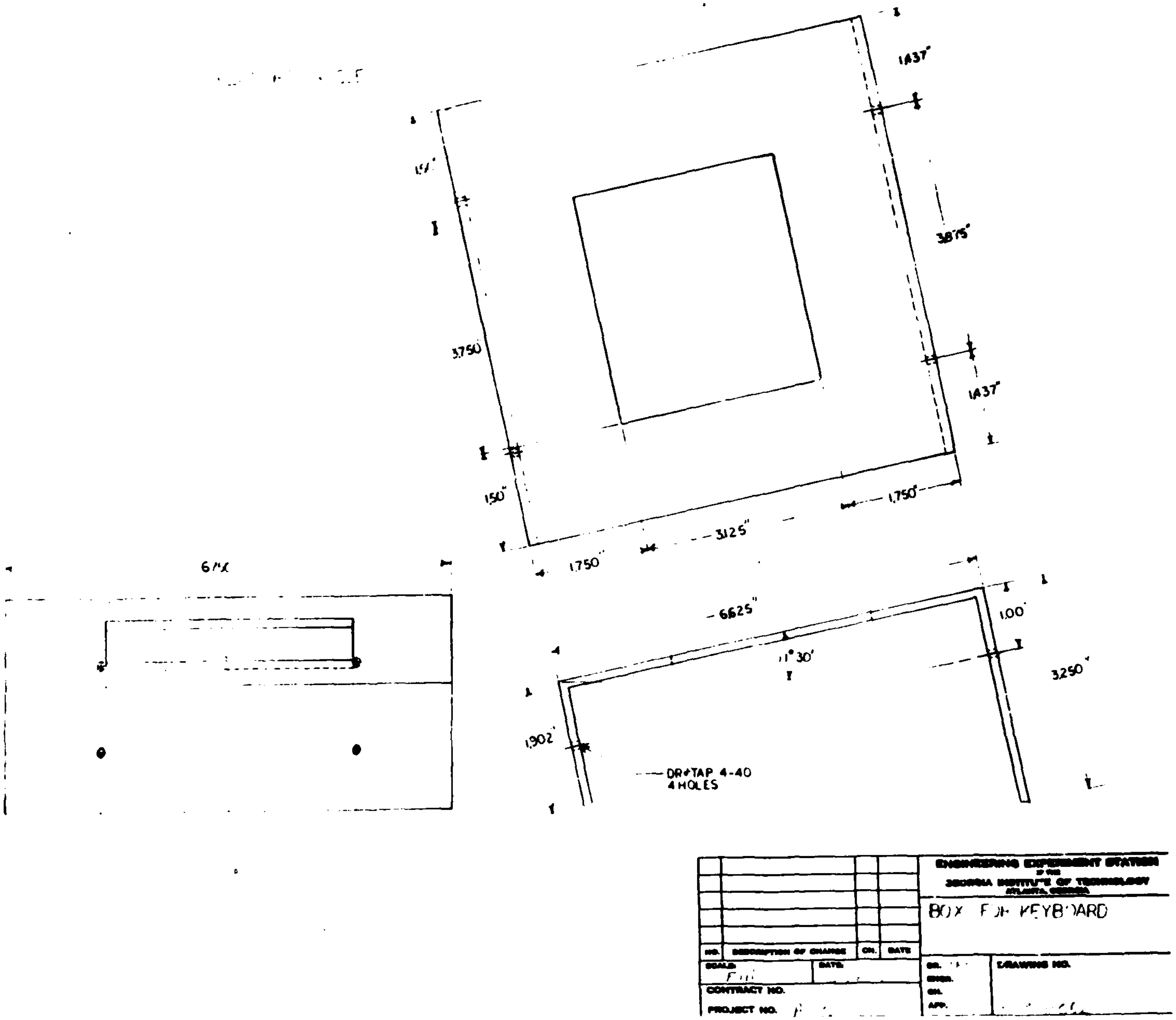
Drawing 44. Modification of Front Panel M68MMLC1 Micro-Module.



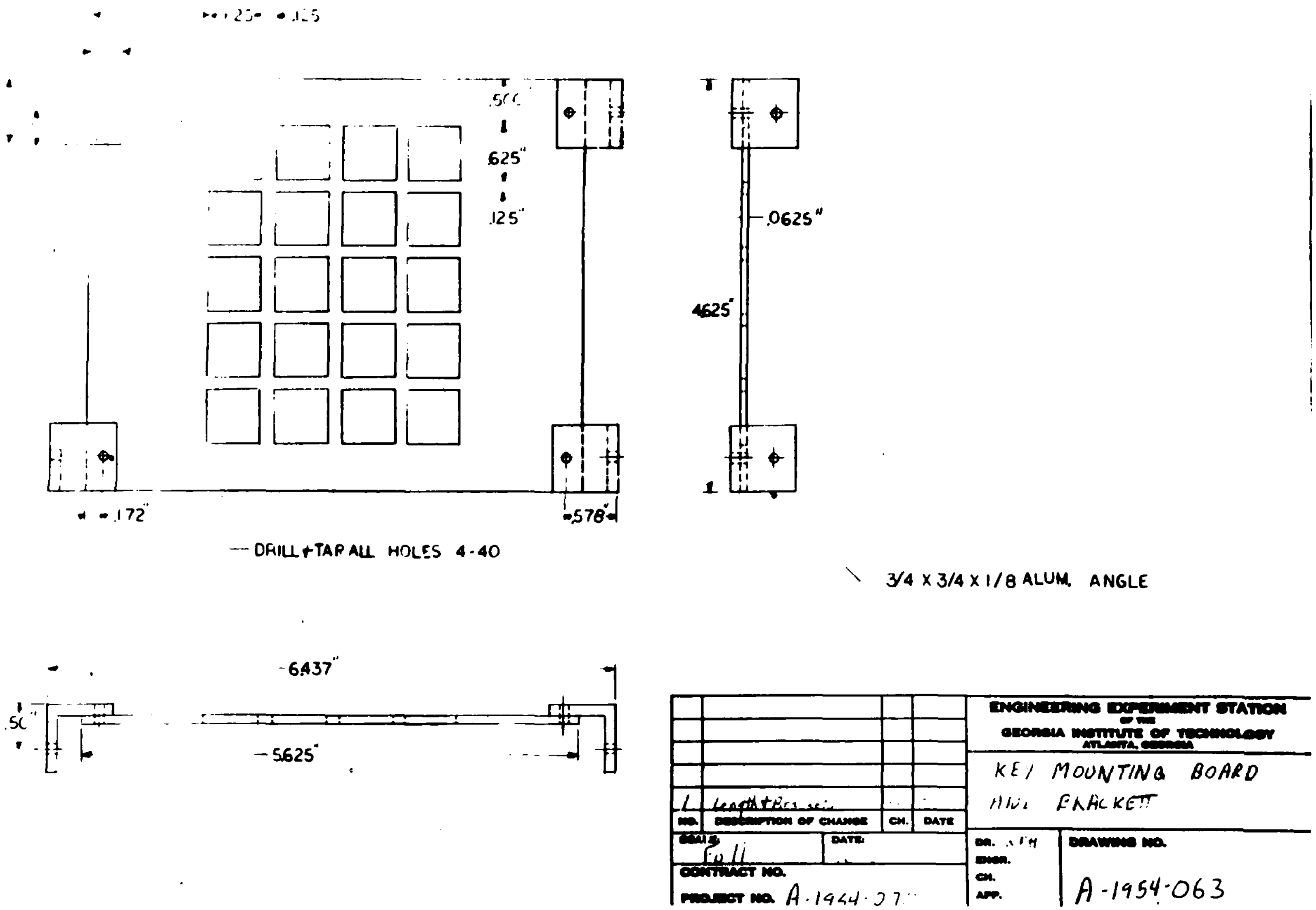
Drawing 45. Modification of Back Panel M68MMLC1 Micro-Module.



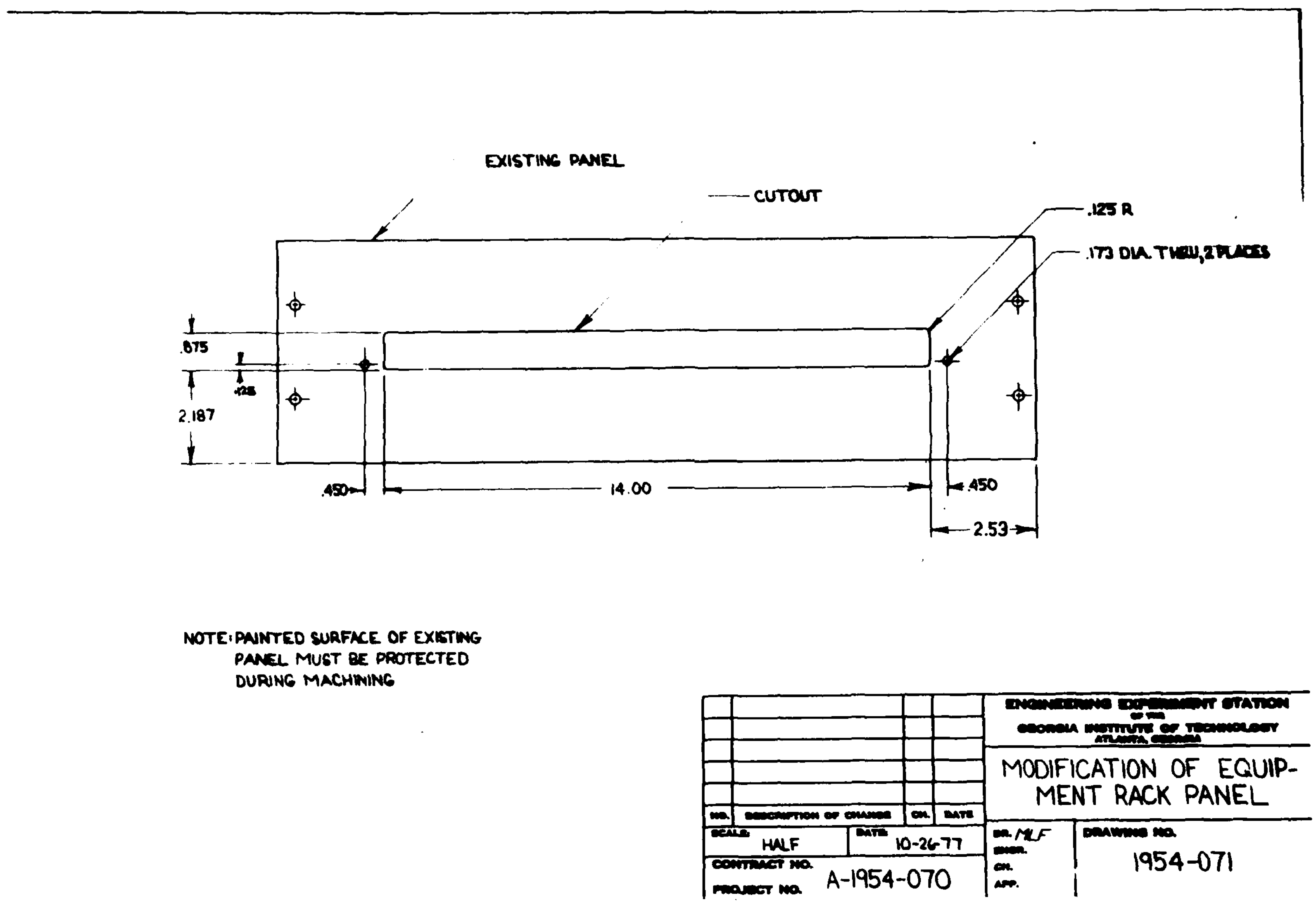
Drawing 46. Keyboard Assembly.



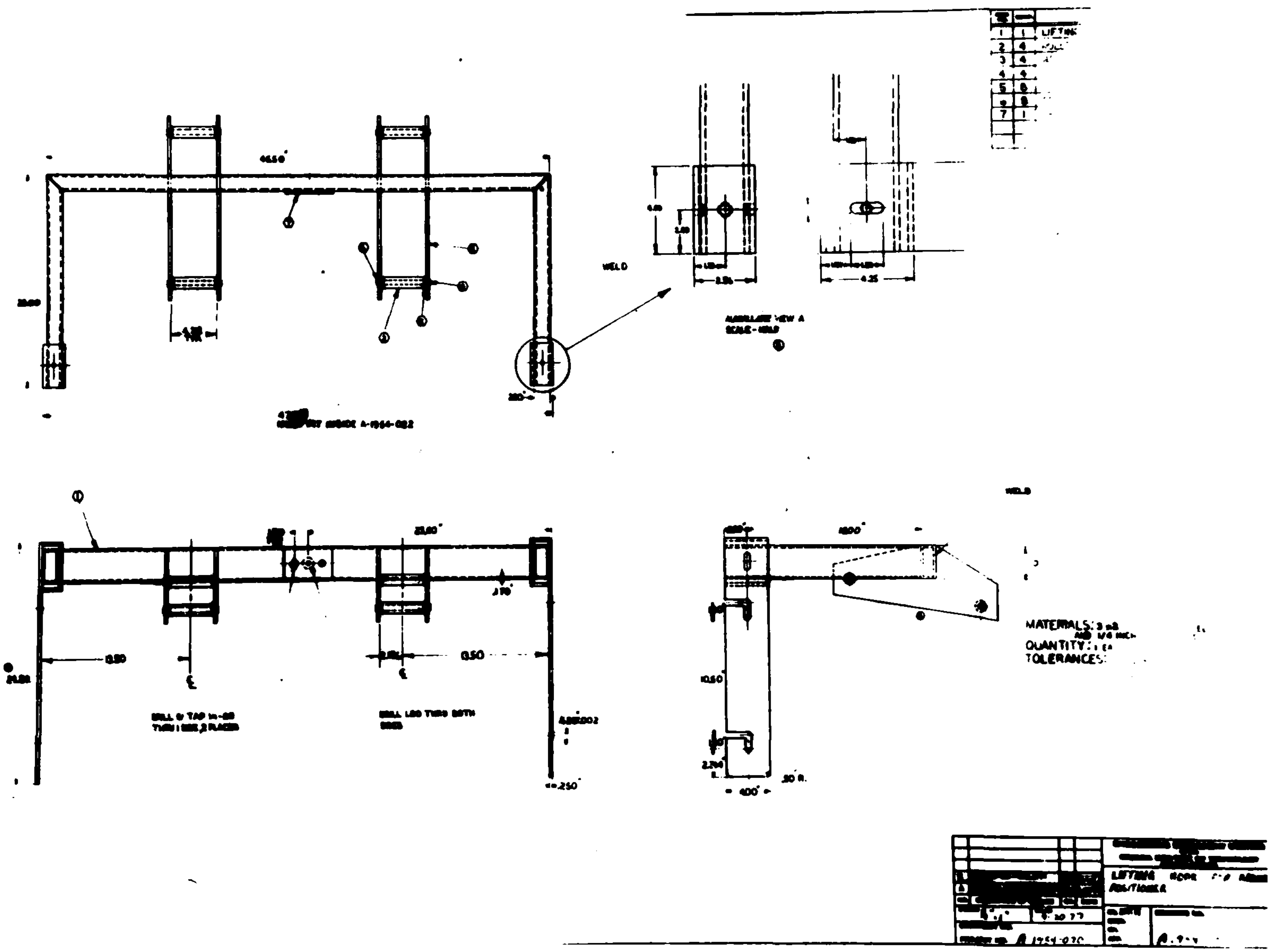
Drawing 47. Box for Keyboard



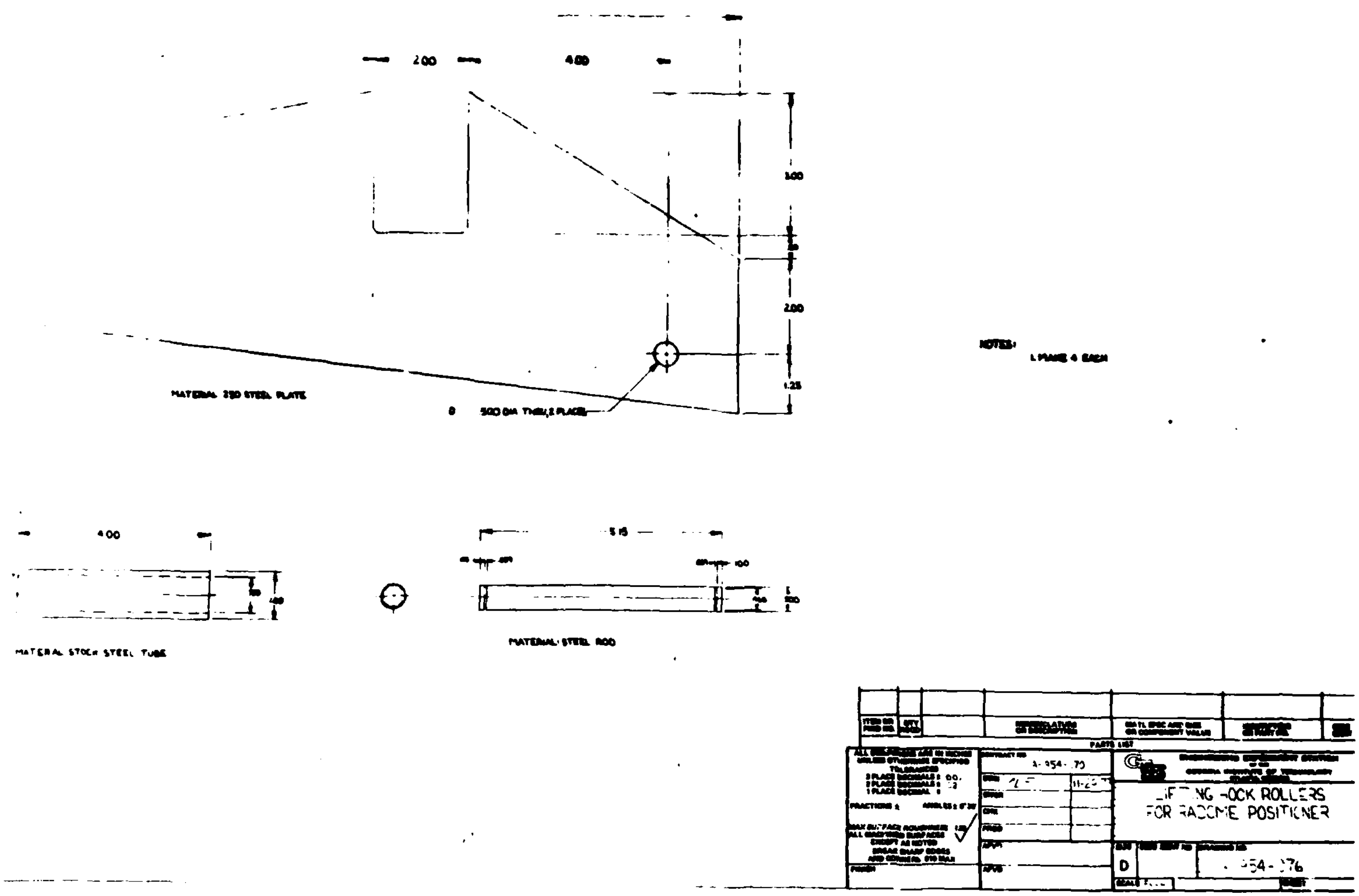
Drawing 48. Key Mounting Board and Bracket



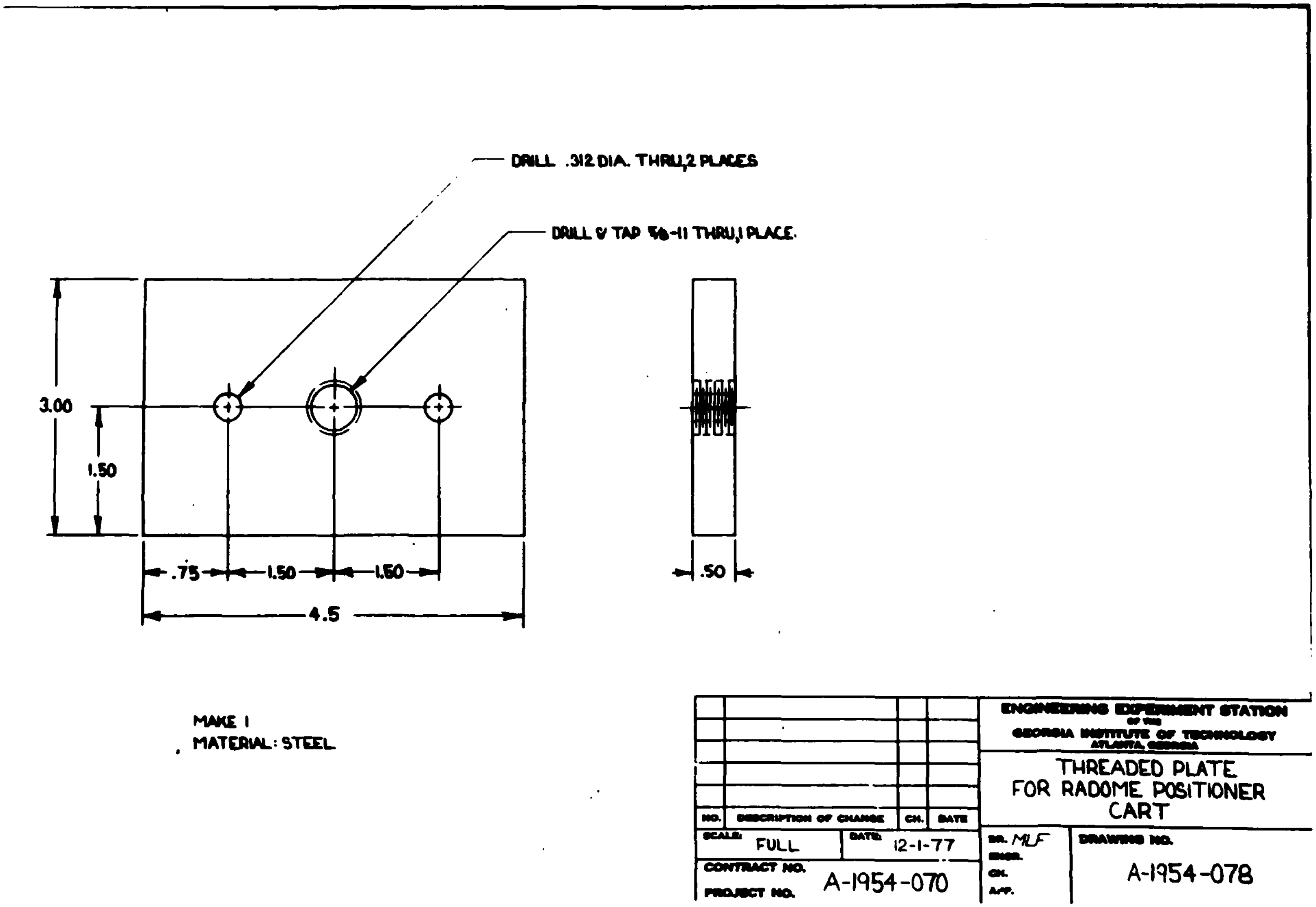
Drawing 49. Panel Modifications for Digital Readout



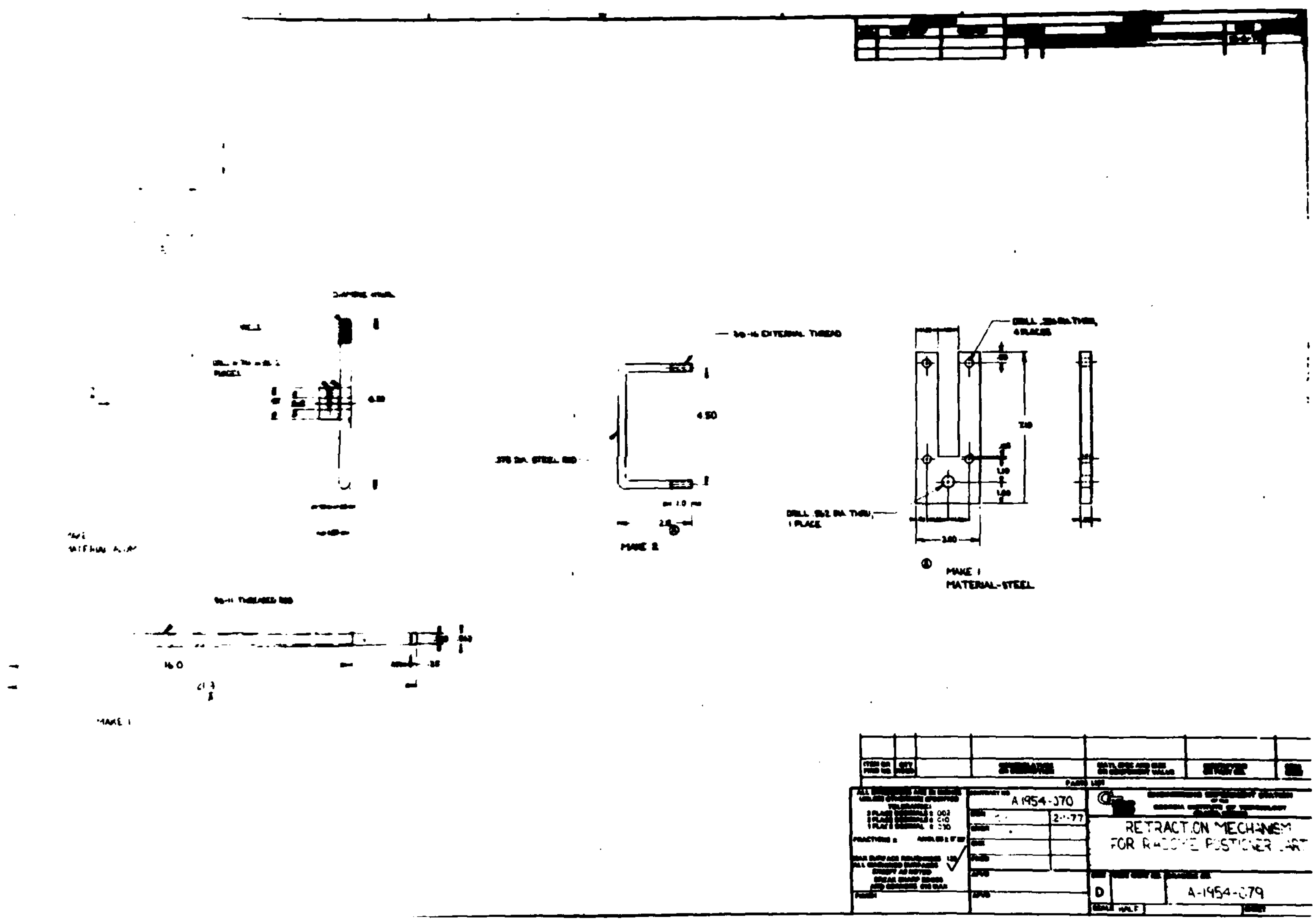
Drawing 50. Lifting Hook Assembly for Radome Positioner



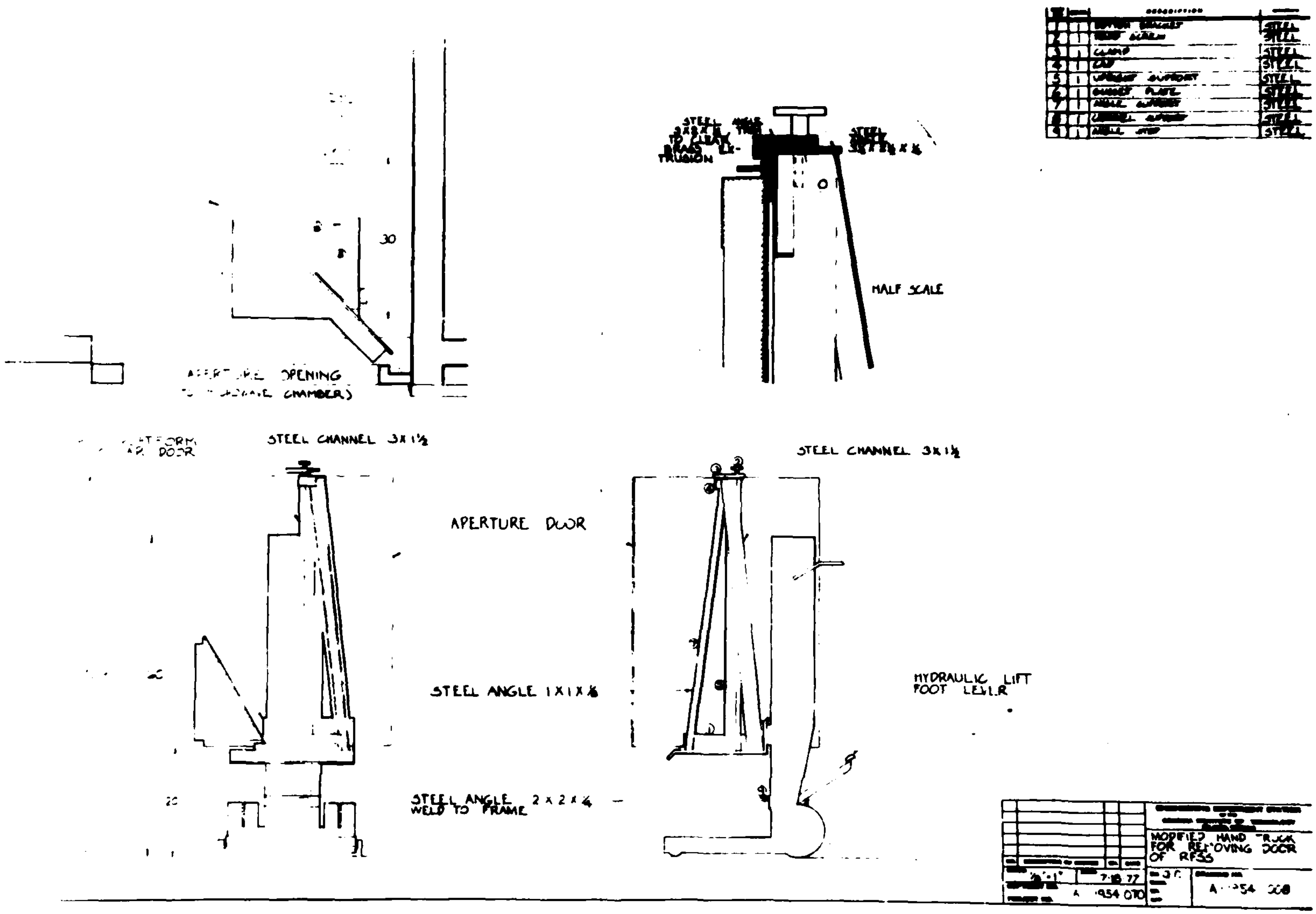
Drawing 51. Lifting Hook Details



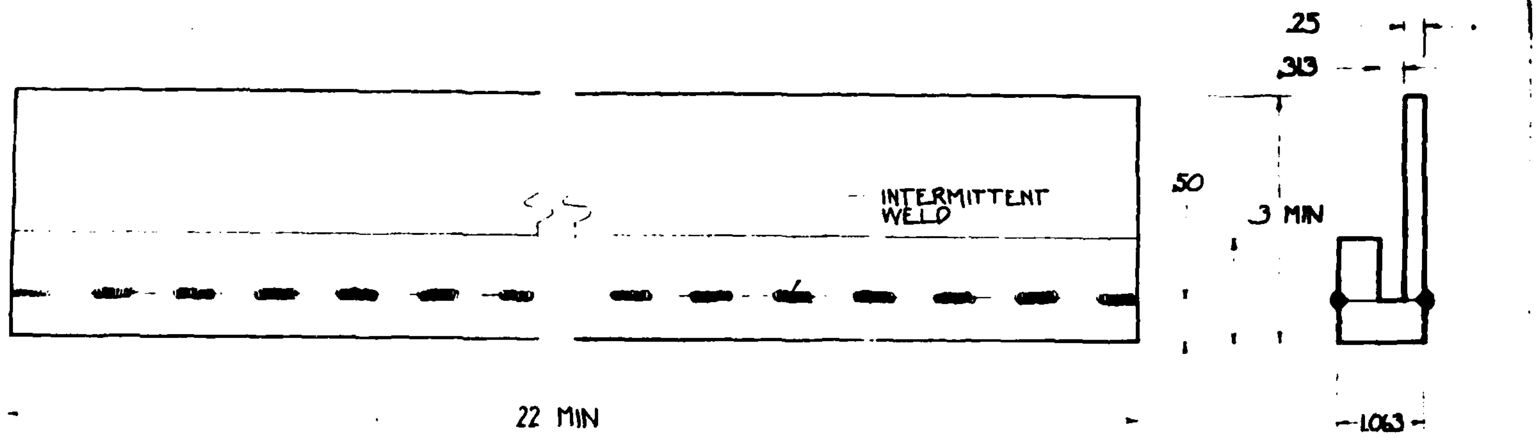
Drawing 52. Threaded Plate for Lifting Hook



Drawing 53. Retractor Mechanism Components for Radome Positioner Removal Cart



Drawing 54. Modified Hand Truck for Removing Door of RFSS



2.- FINISH: NONE
1- MATERIAL: STEEL
NOTES:

TOLERANCES:
X = ±.060
XX = ±.030
XXX = ±.010

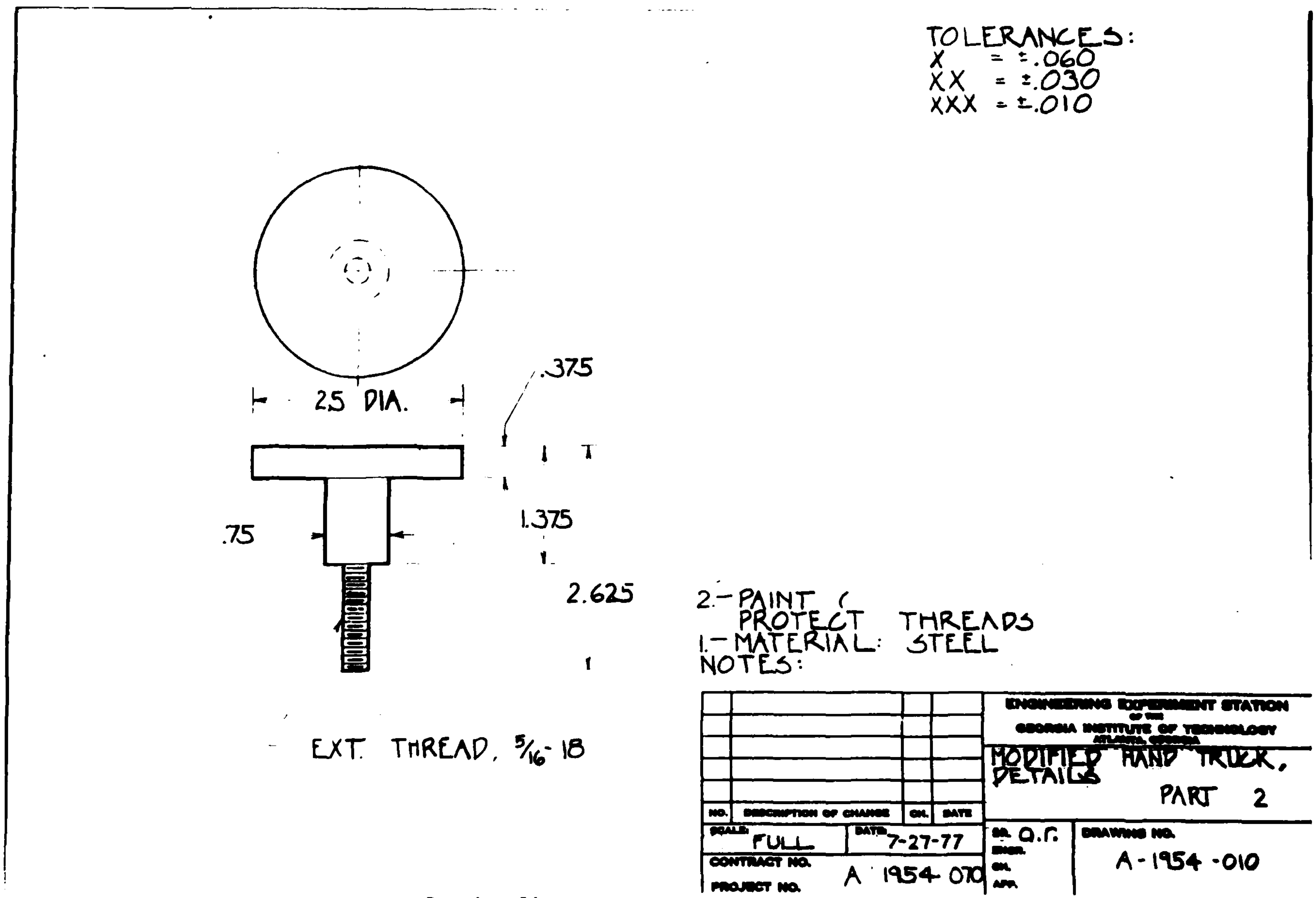
NO	DESCRIPTION OF CHANGE	CH.	DATE

ENGINEERING EXPERIMENT STATION
OF THE
GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA

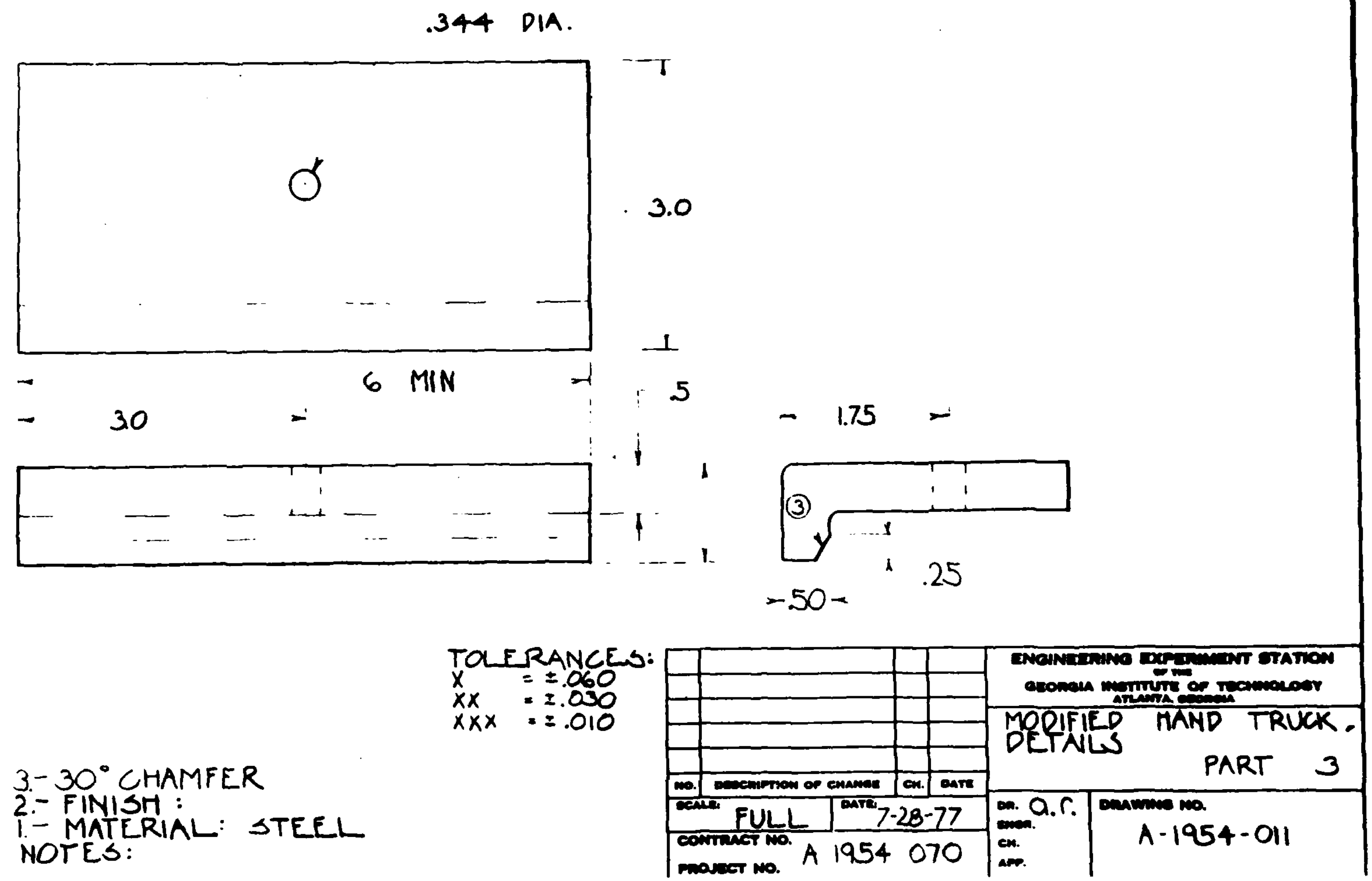
BOTTOM BRACKET FOR MODIFIED HAND TRUCK PART 1

SCALE FULL
DATE 7-27-77
DR. Q.C.
CHECK
CH.
APP.
CONTRACT NO.
PROJECT NO. A 1954 073
DRAWING NO. A-1954-009

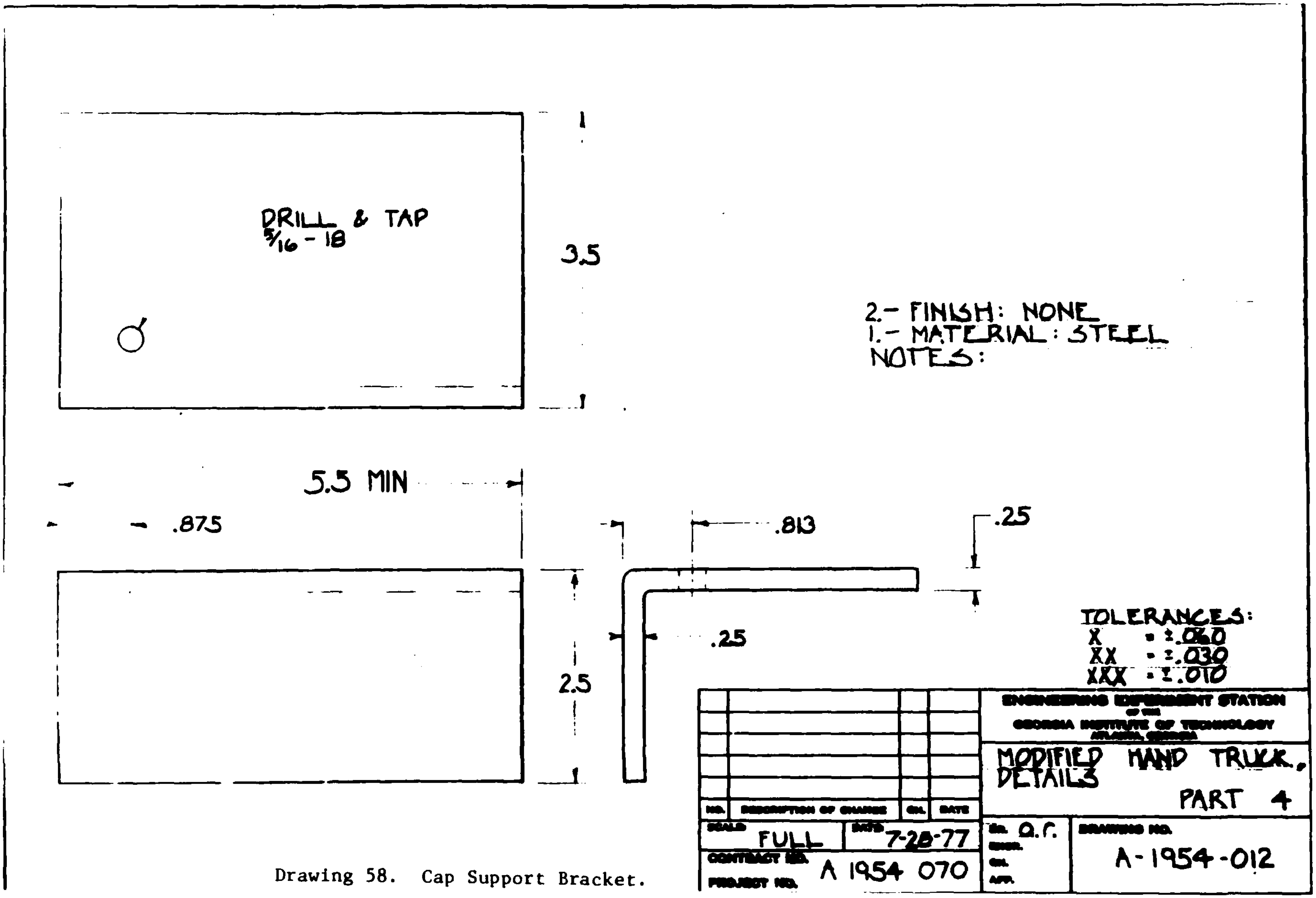
Drawing 55. Door Bottom Support Bracket



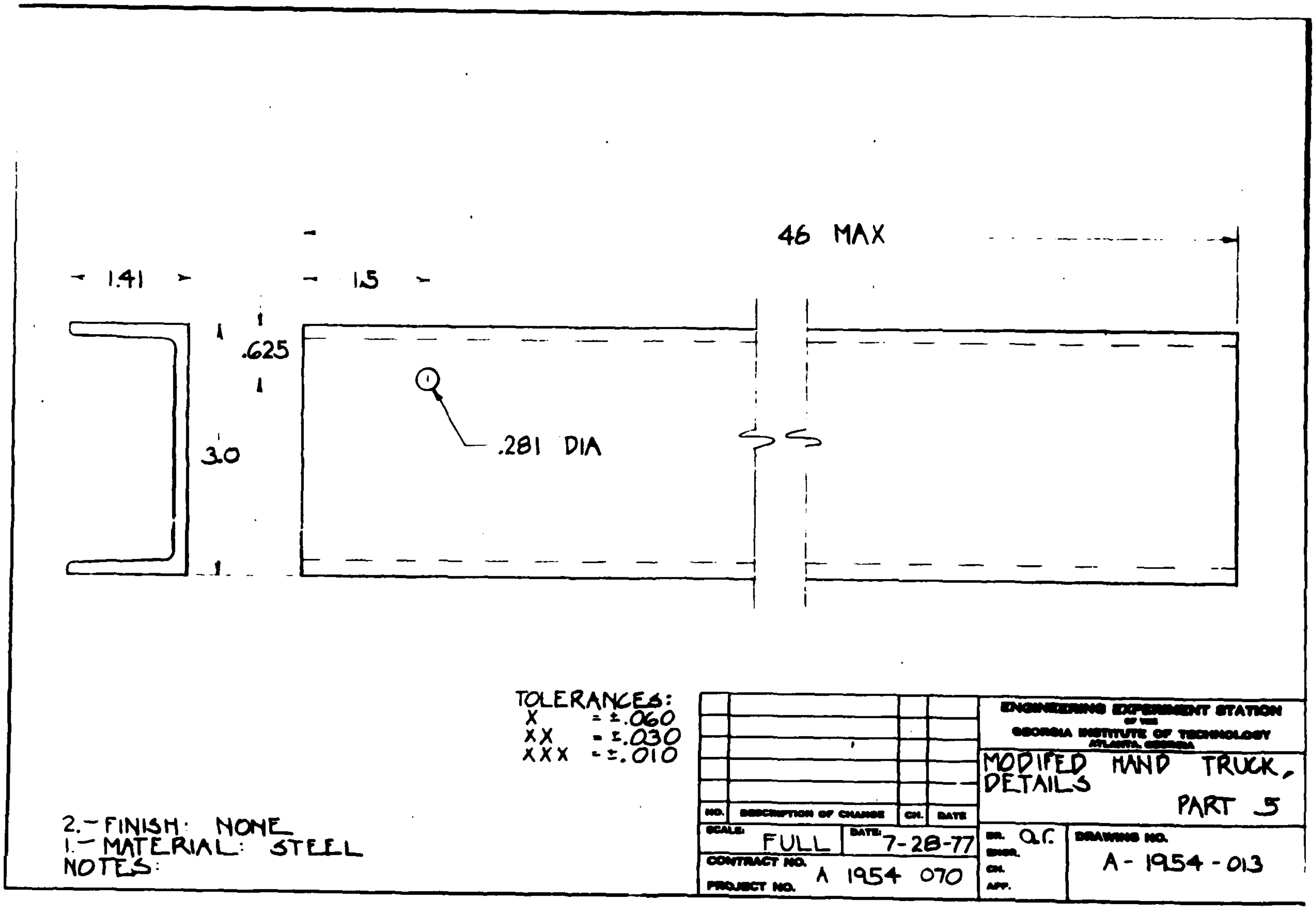
Drawing 56. Door Clamp Screw Top.



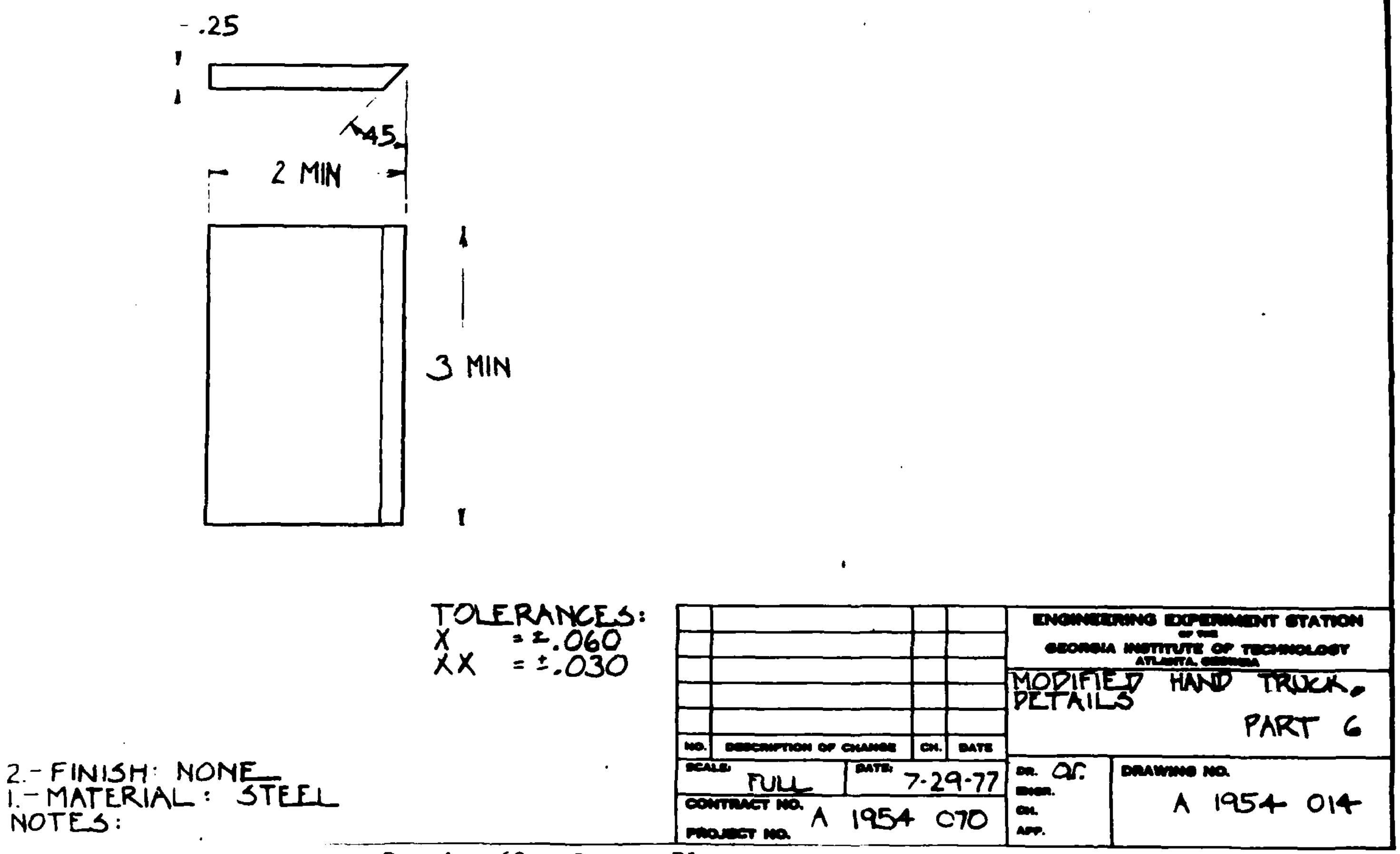
Drawing 57. Door Clamp Top



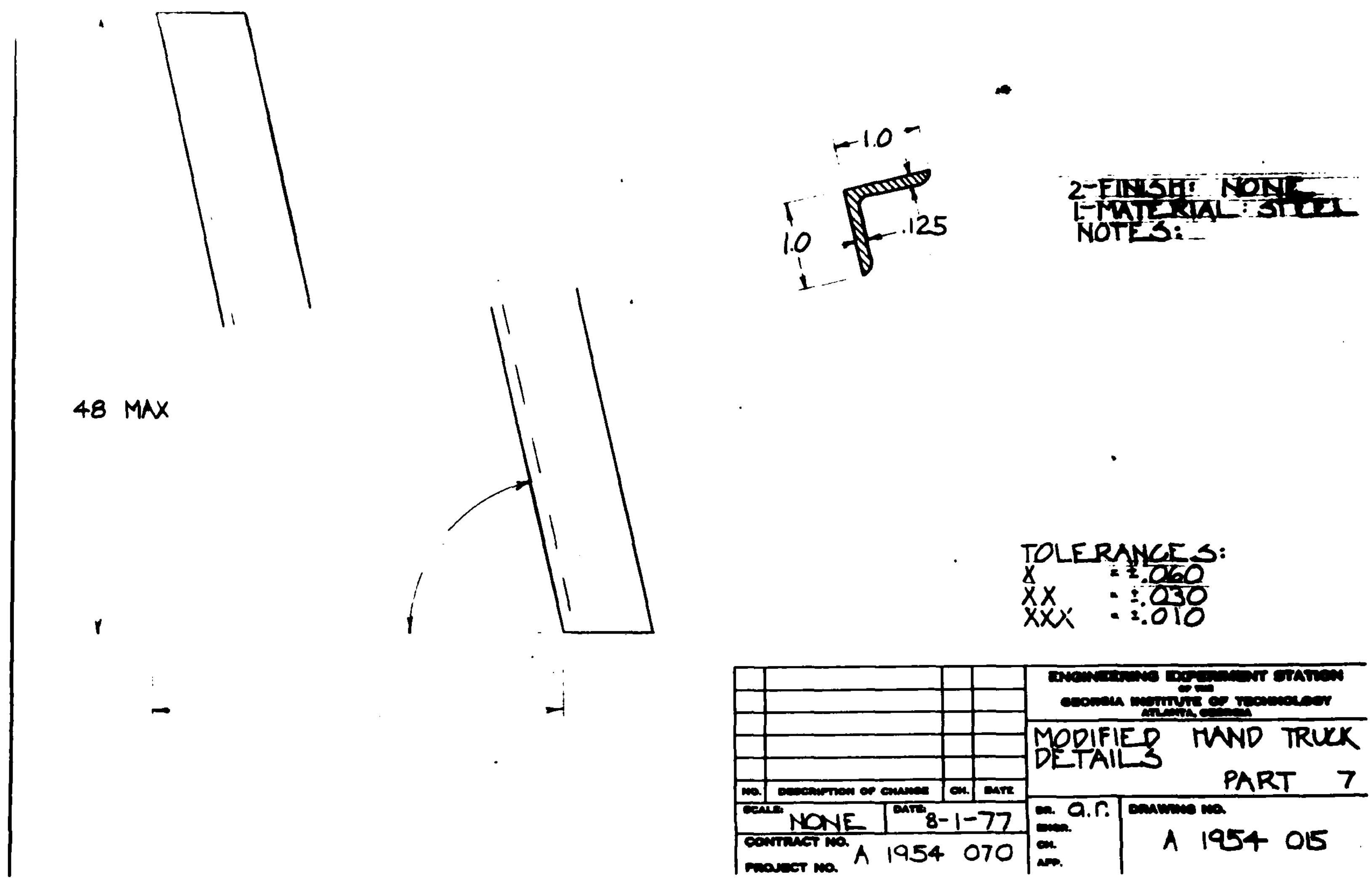
Drawing 58. Cap Support Bracket.



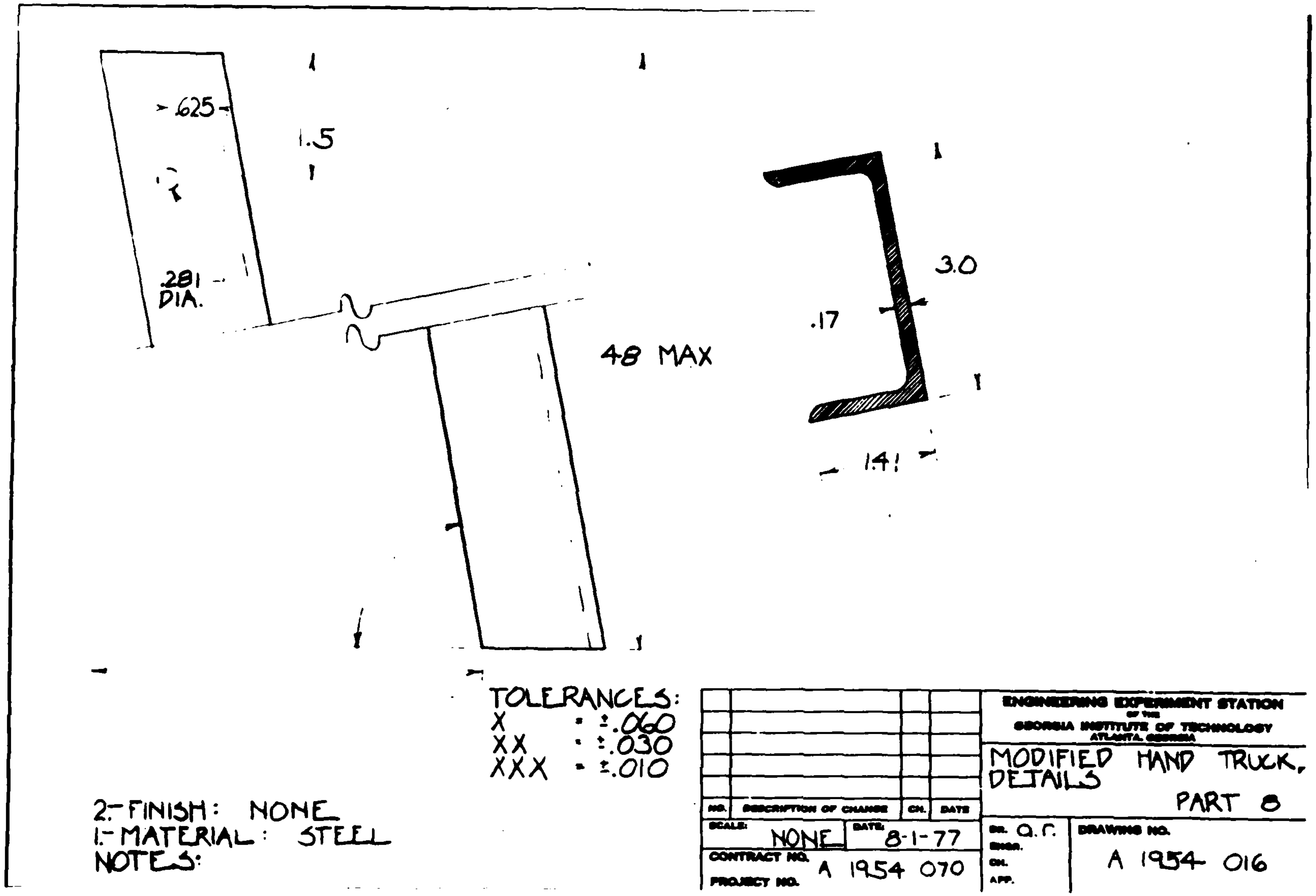
Drawing 59. Upright Support



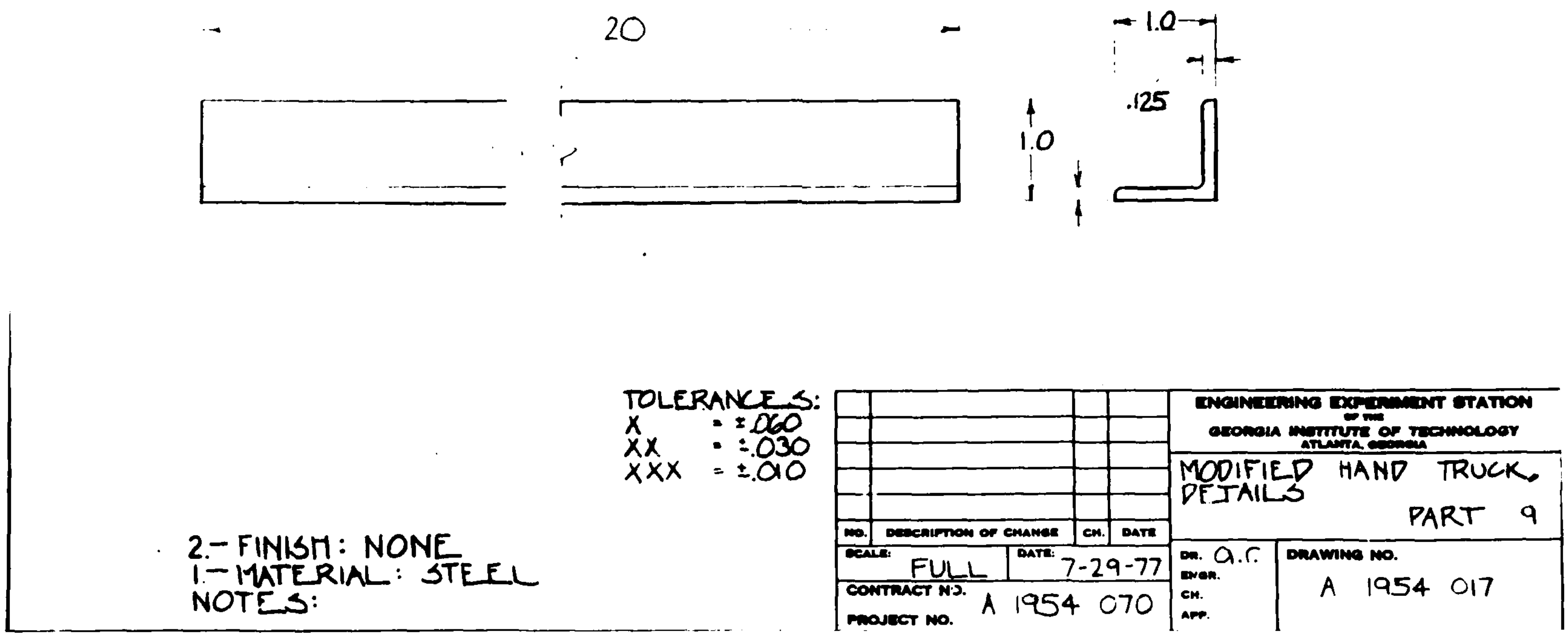
Drawing 60. Gusset Plate.



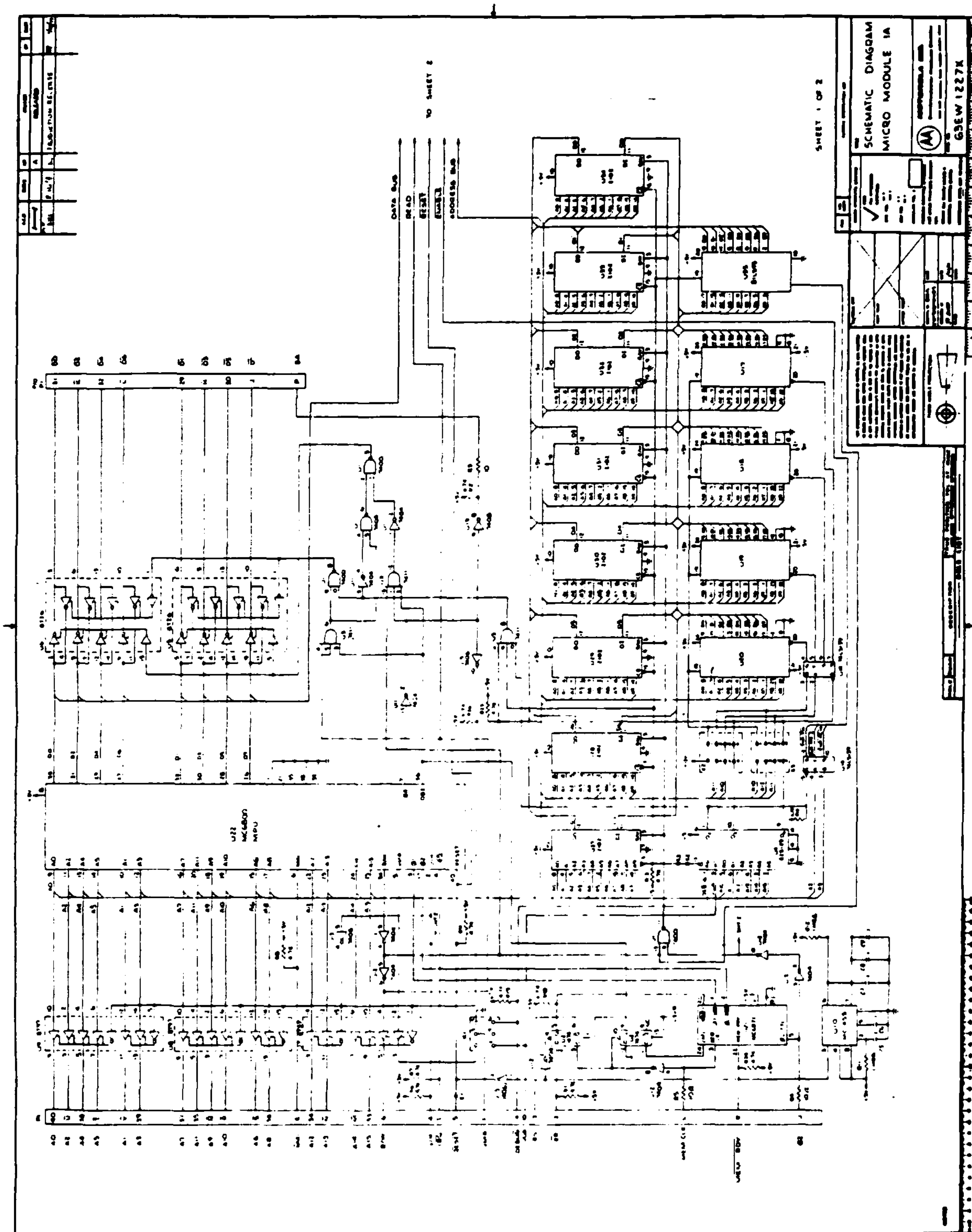
Drawing 61. Angle Support



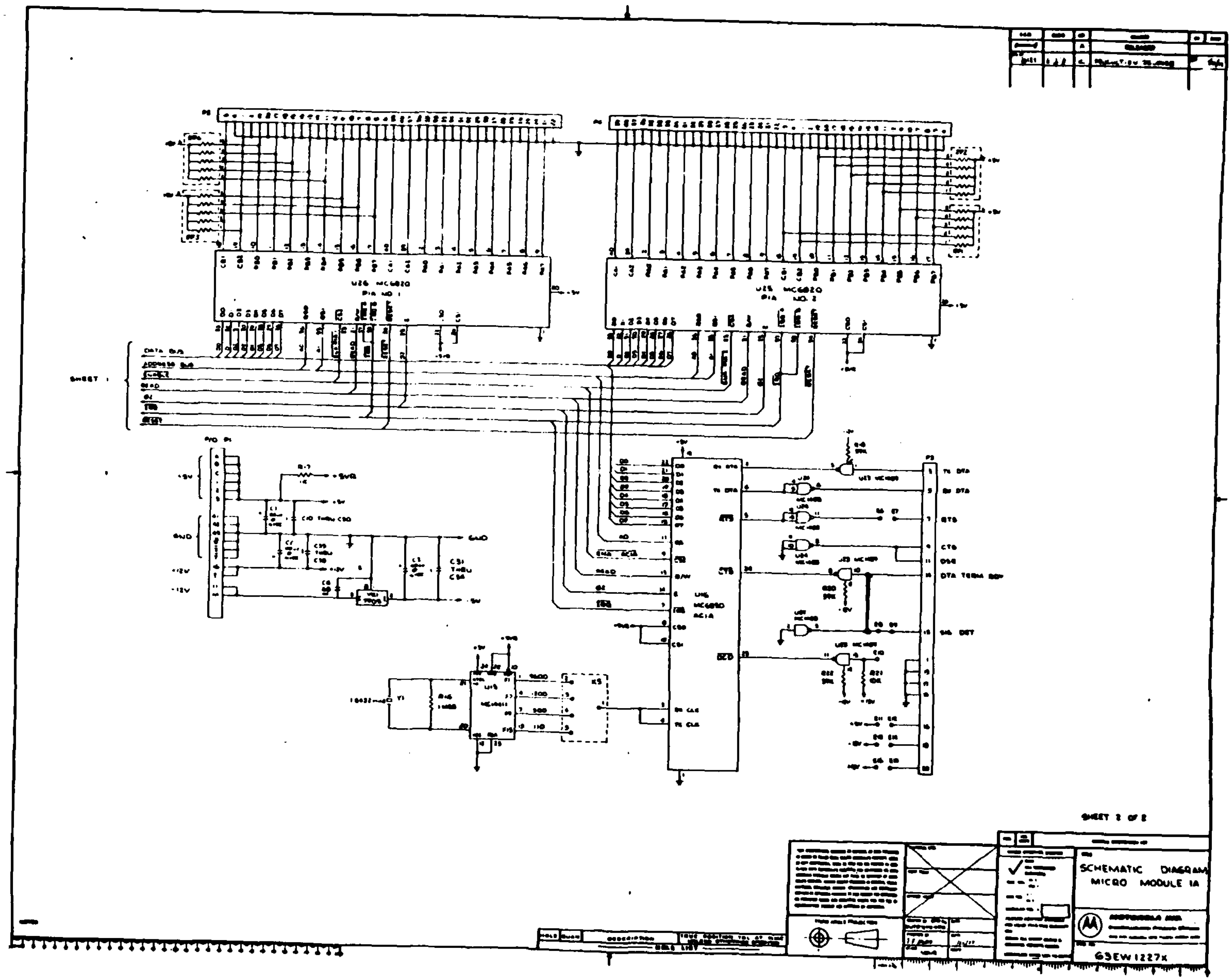
Drawing 62. Channel Support



Drawing 63. Angle Stop



Drawing 64. Motorola M68MM01A-1 Microcomputer.



Drawing 65. Parallel and Serial Interface Schematic for M68MM01A-1.

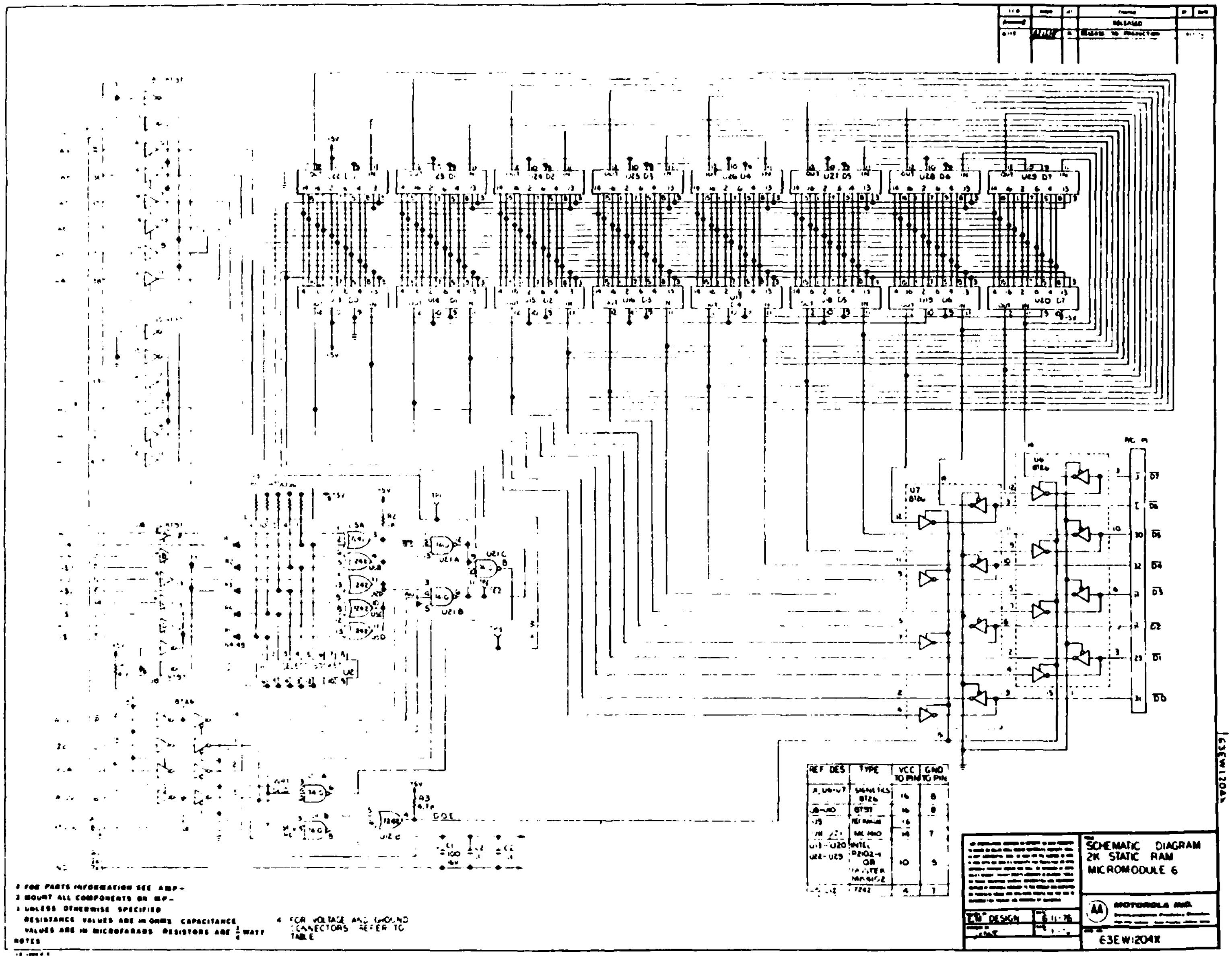


FIGURE 3-2. 2K Static RAM Module Schematic Diagram

Drawing 66. M68MM06 2k Byte Static RAM Board Schematic.

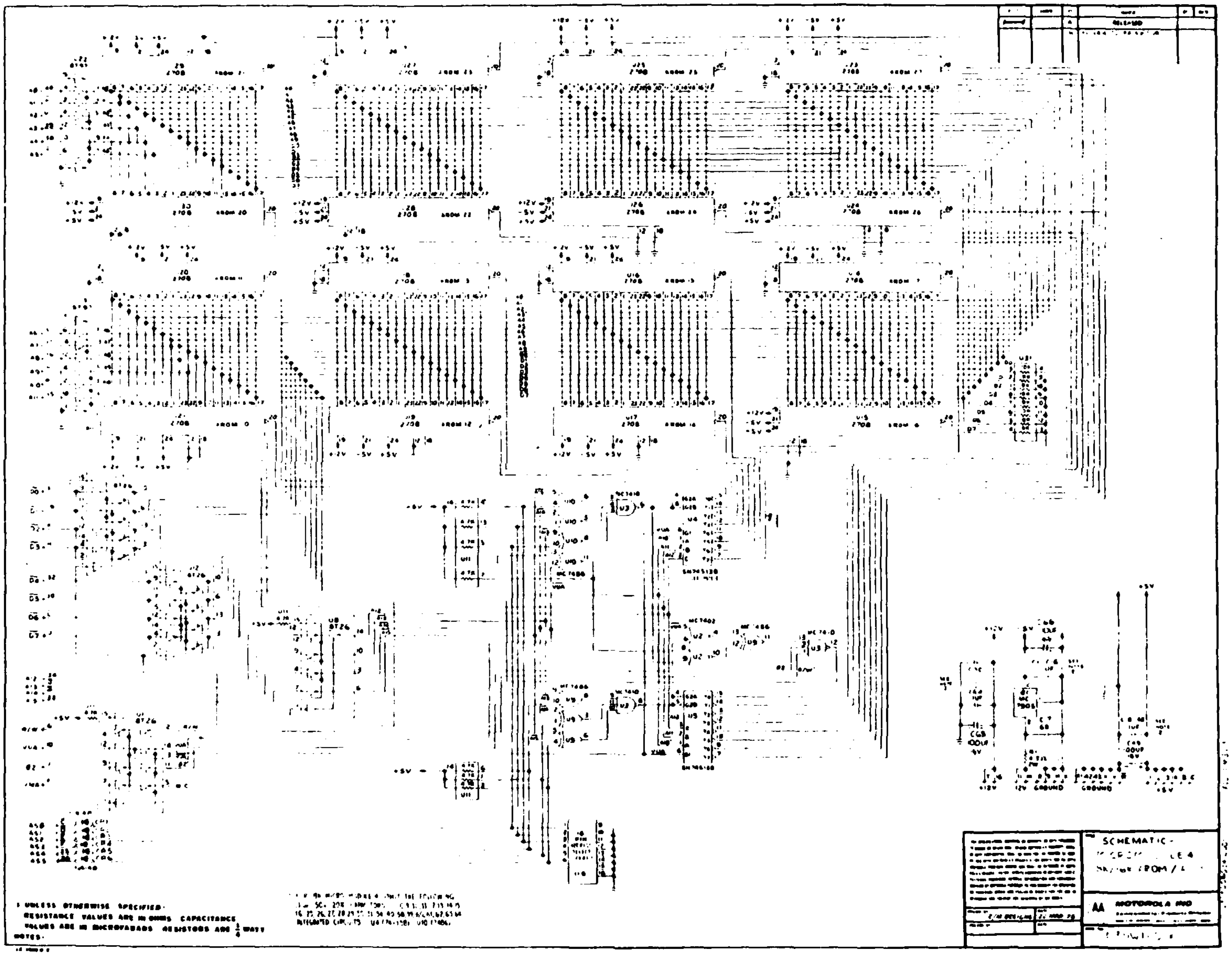


FIGURE 3-2. 8K/16K AROM/ROM MODULE SCHEMATIC DIAGRAM 3-3

Drawing 67. M68MM04 16k Byte EPROM Board Schematic.

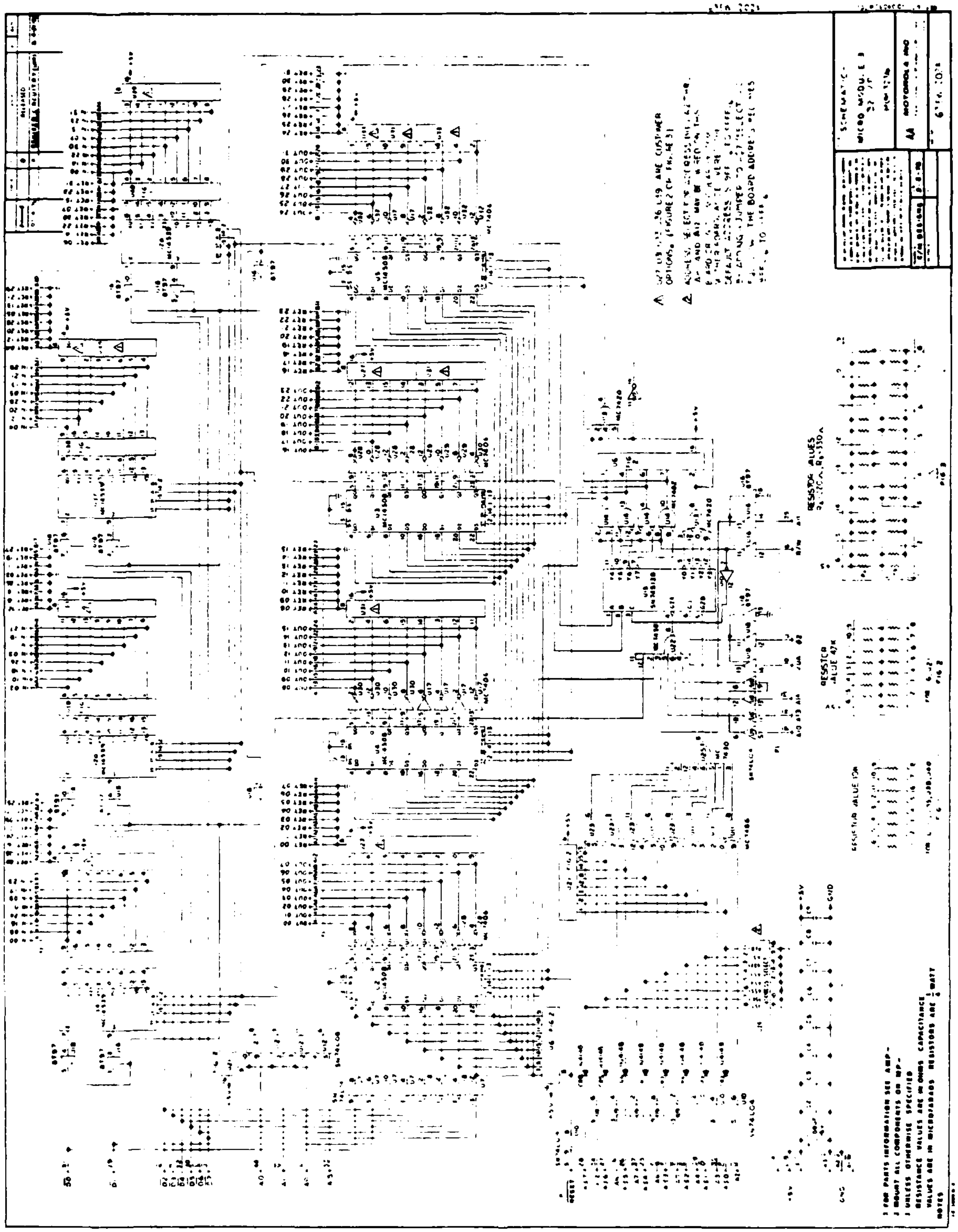
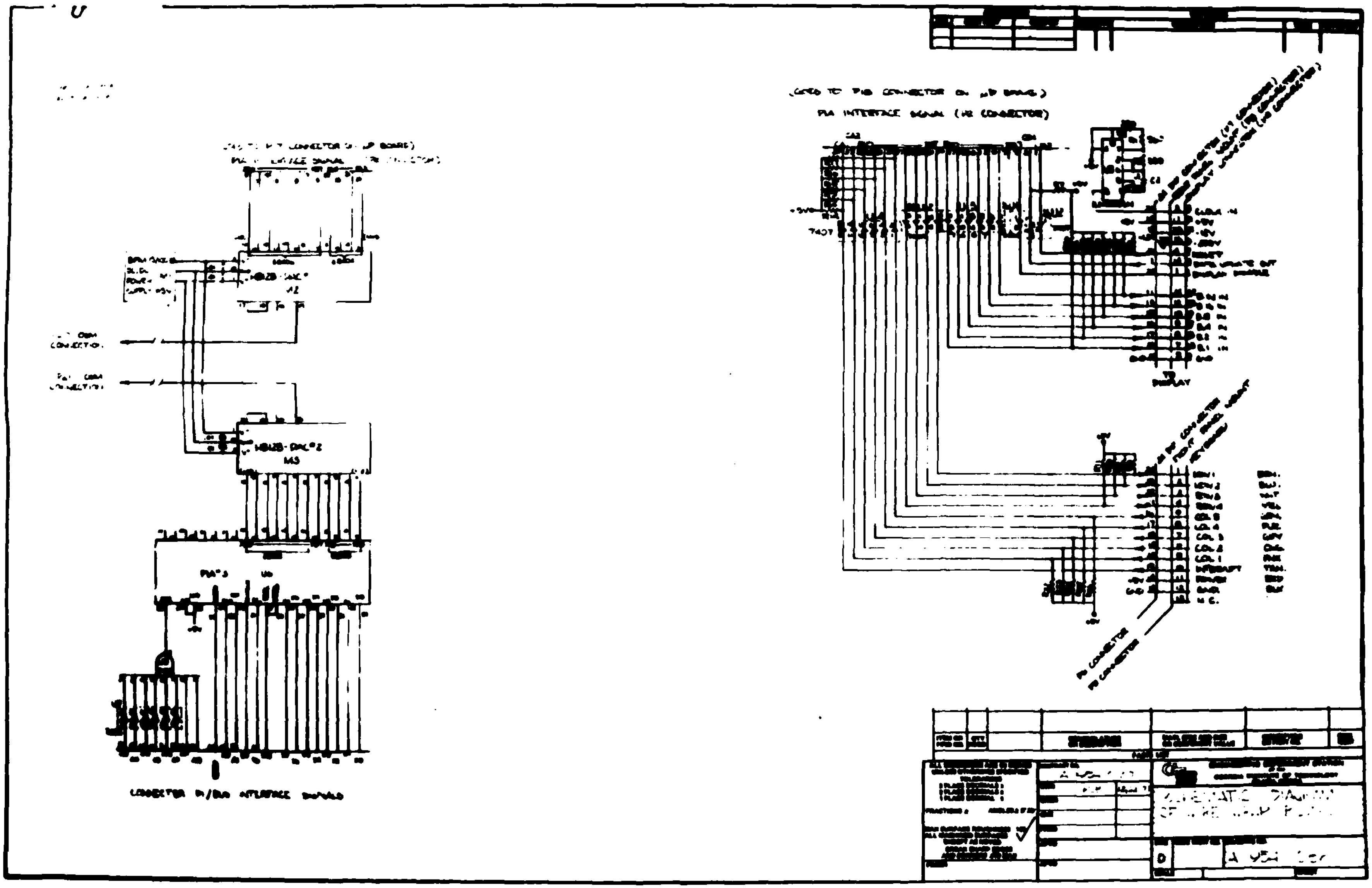
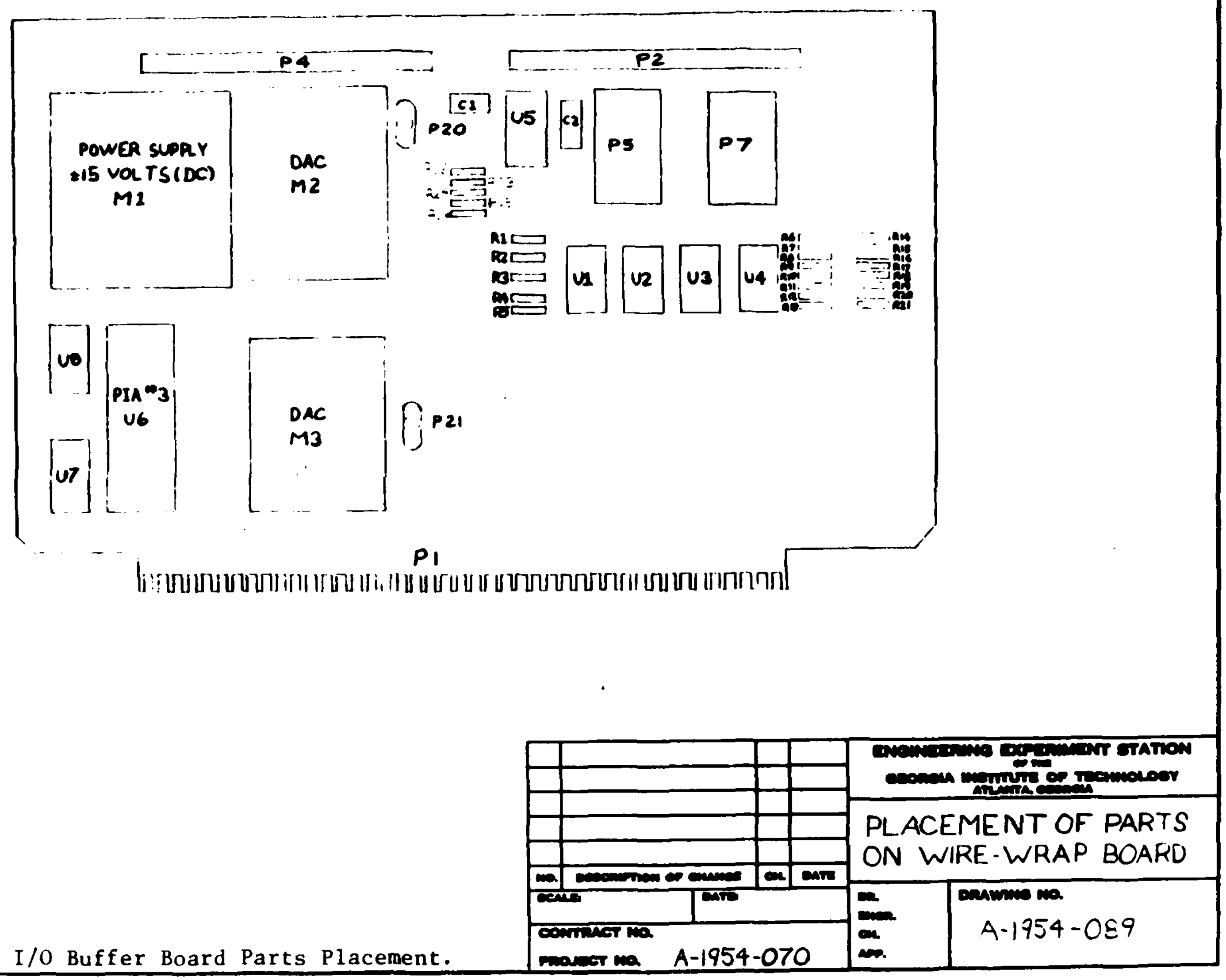


FIGURE 3-2. 32/32 INPUT/OUTPUT MODULE SCHEMATIC DIAGRAM 3-5

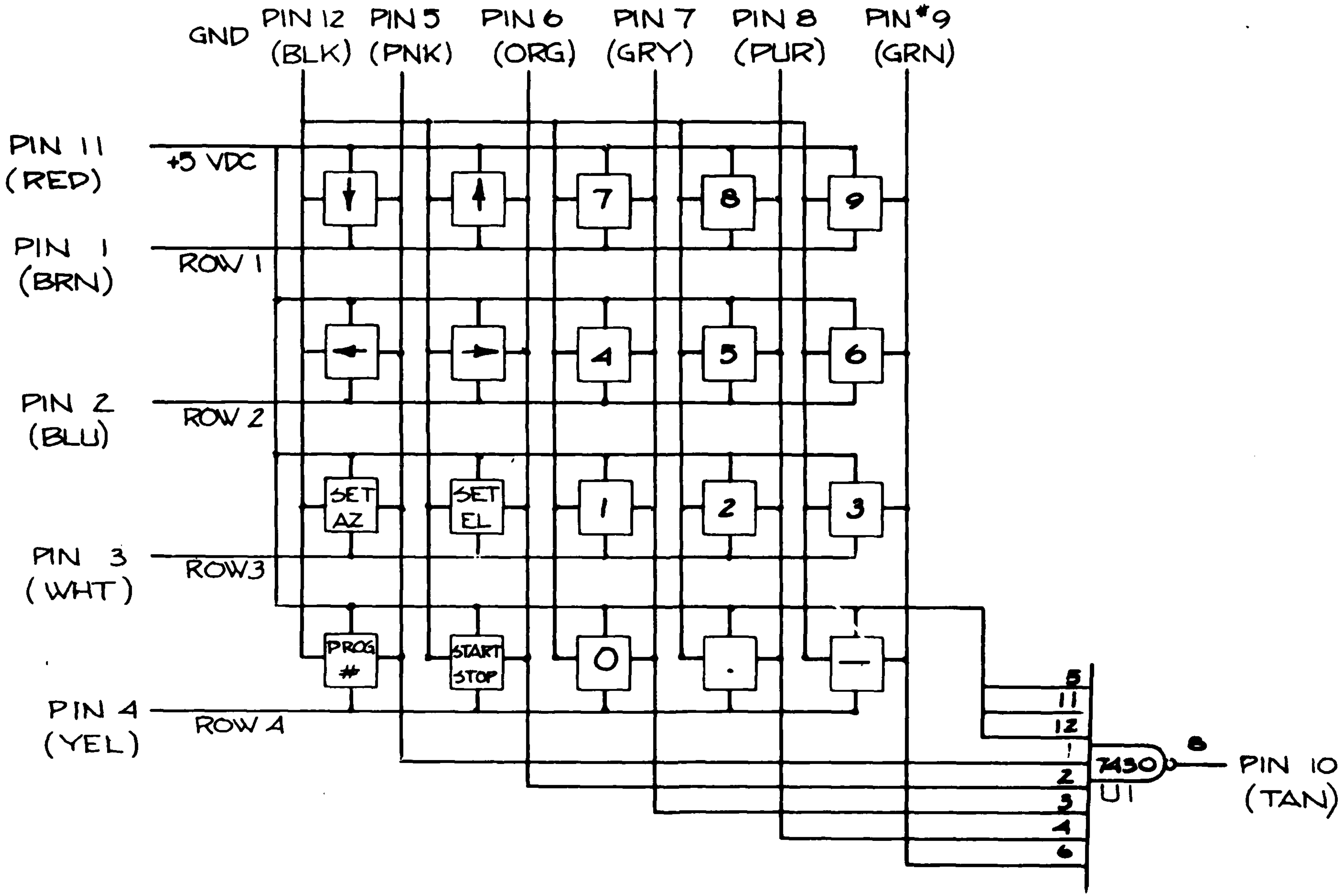
Drawing 68. M68MM03 32 Channel I/O Board Schematic.



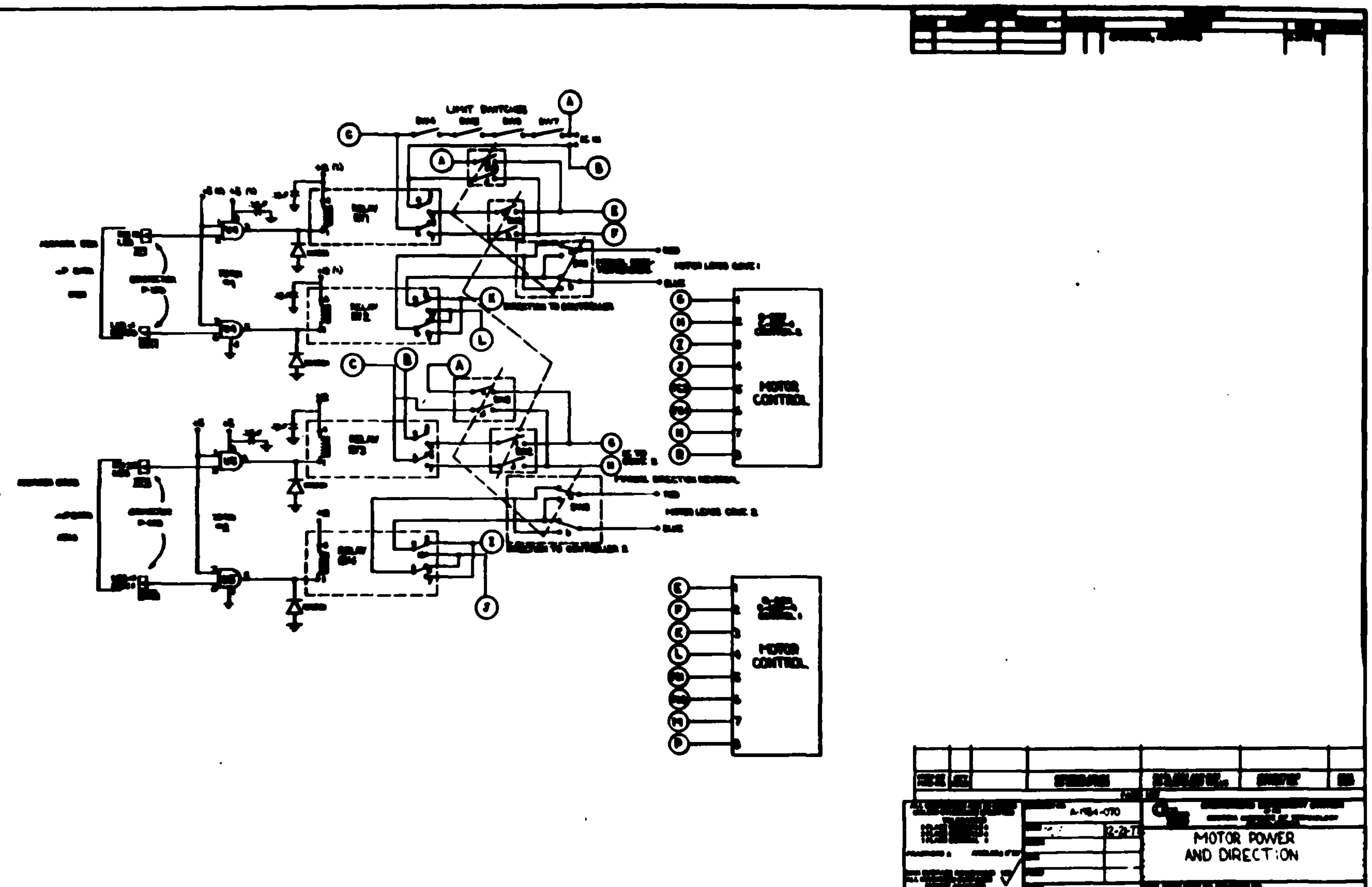
Drawing 69. I/O Buffer Board Schematic



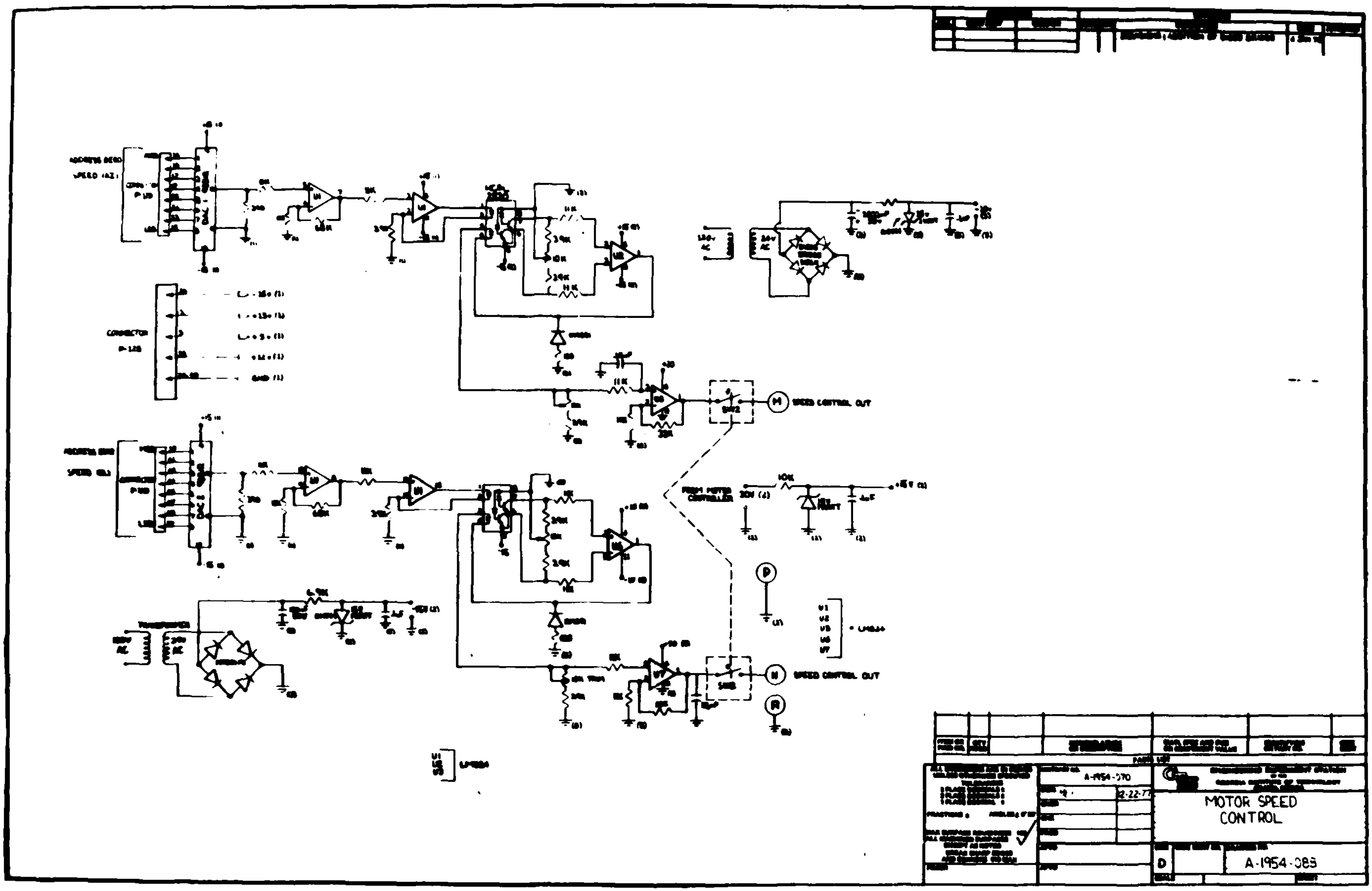
Drawing 70. I/O Buffer Board Parts Placement.



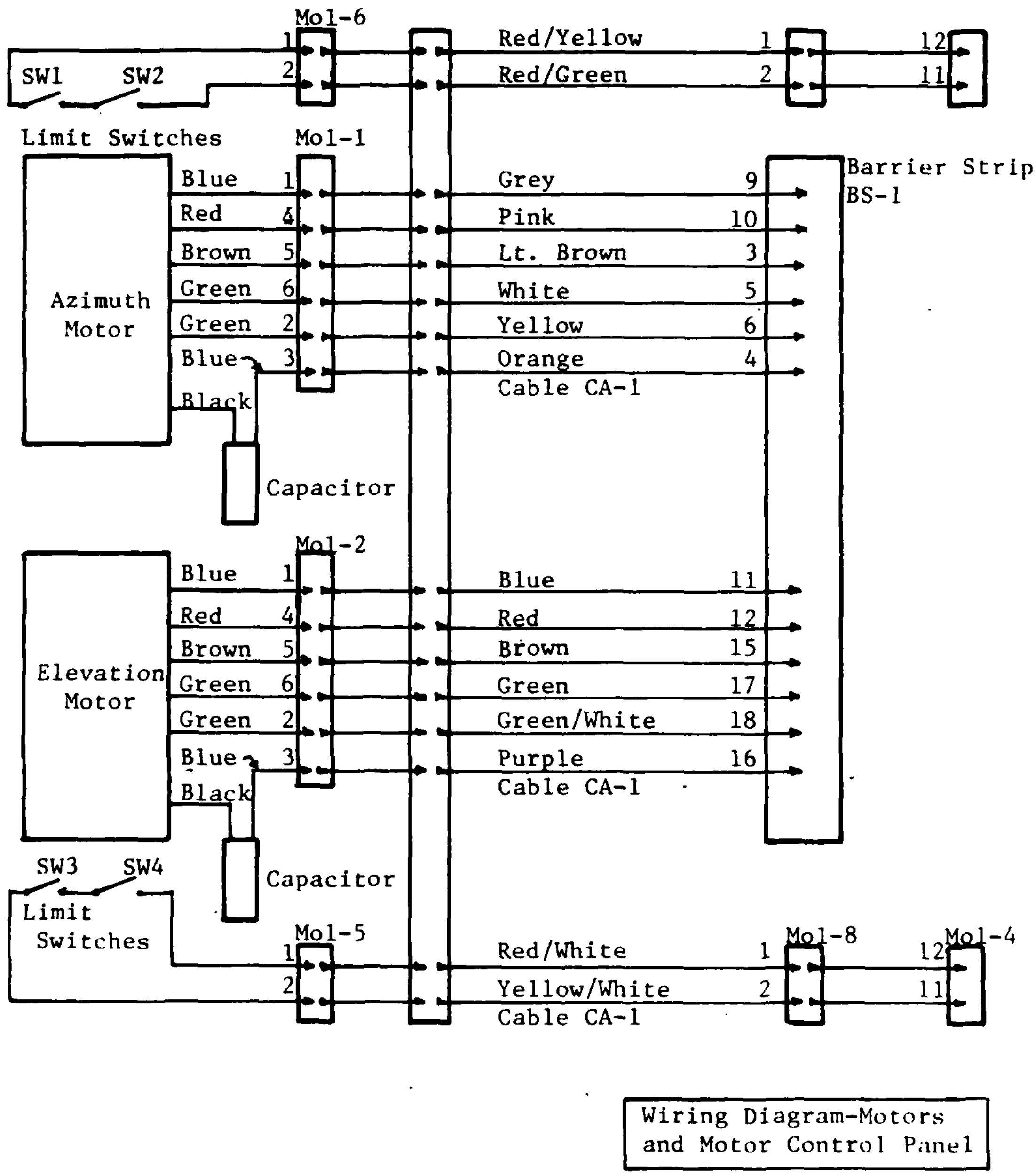
Drawing 71. Keyboard Schematic.



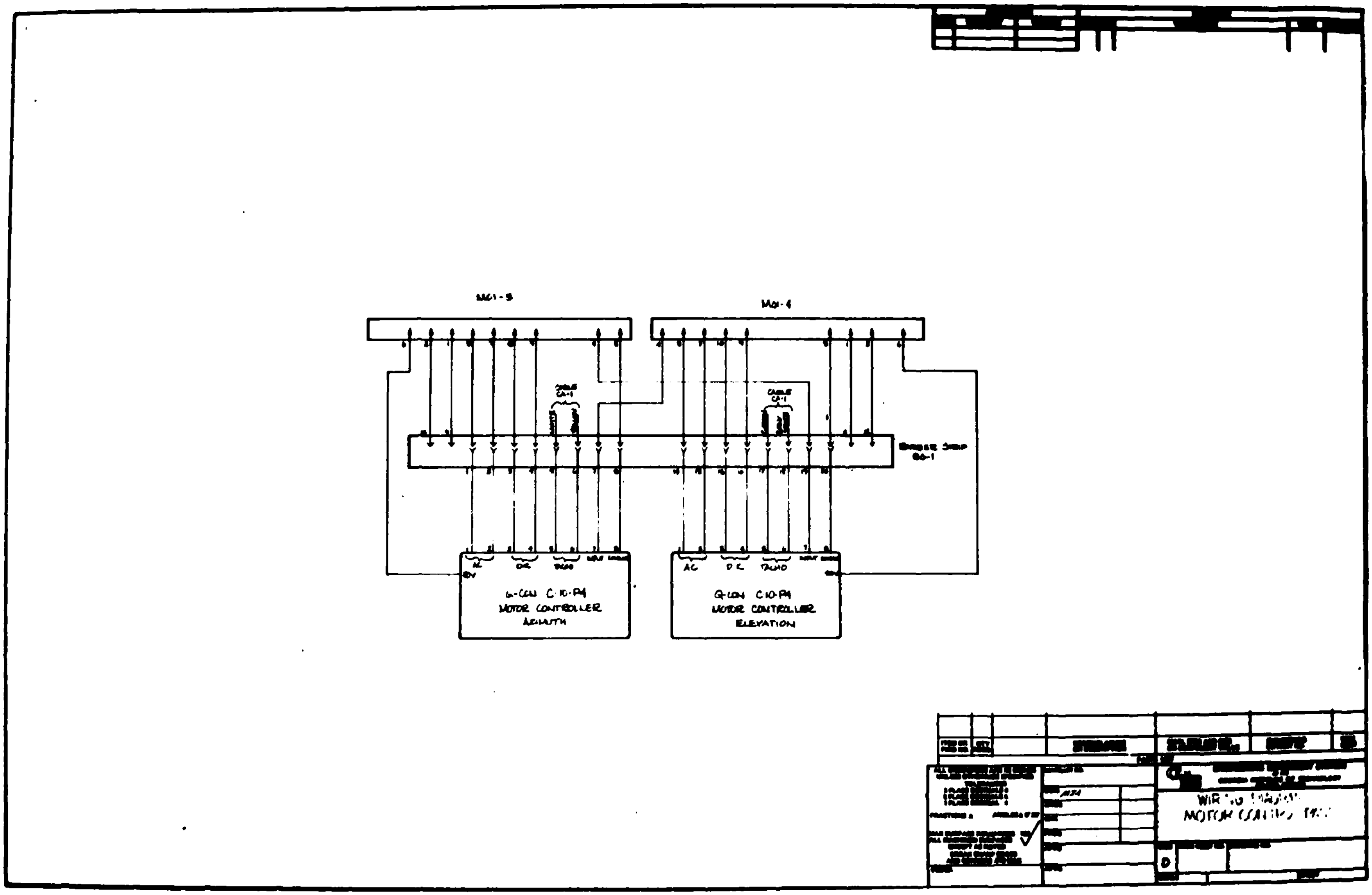
Drawing 72. Motor Power and Direction Control.



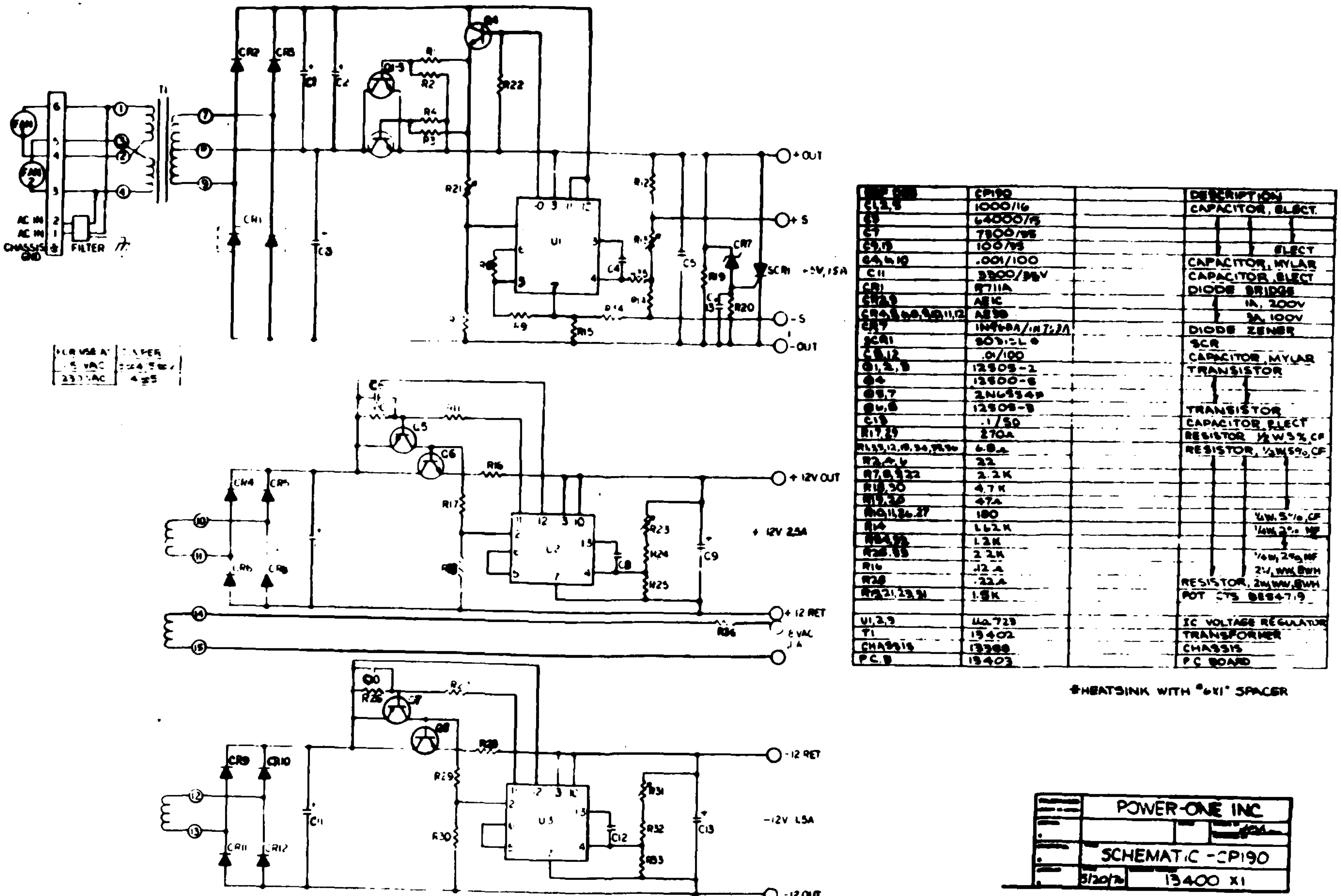
Drawing 73. D/A Converter and Isolated Amplifier.



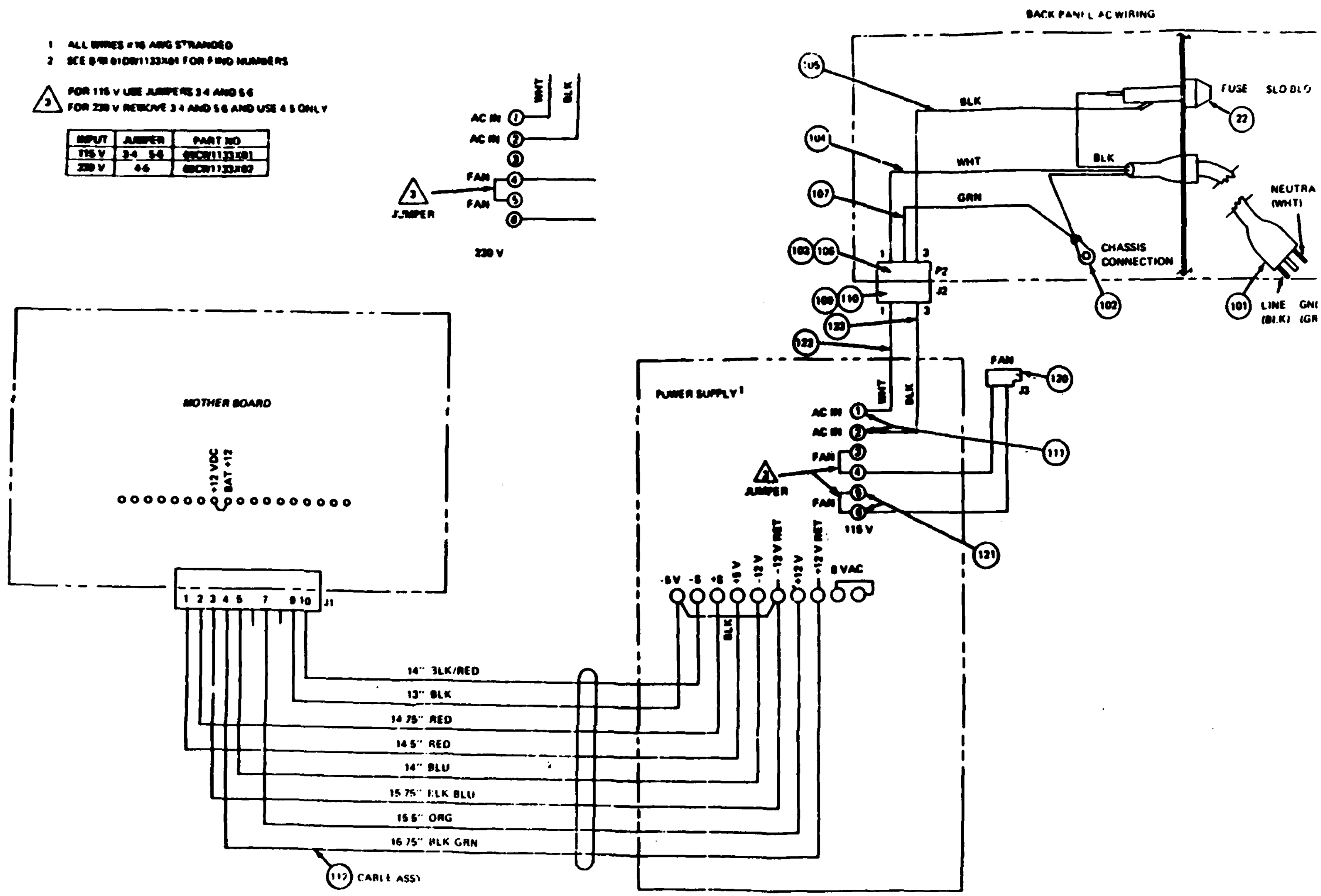
Drawing 74. Motors and Limit Switches Wiring.



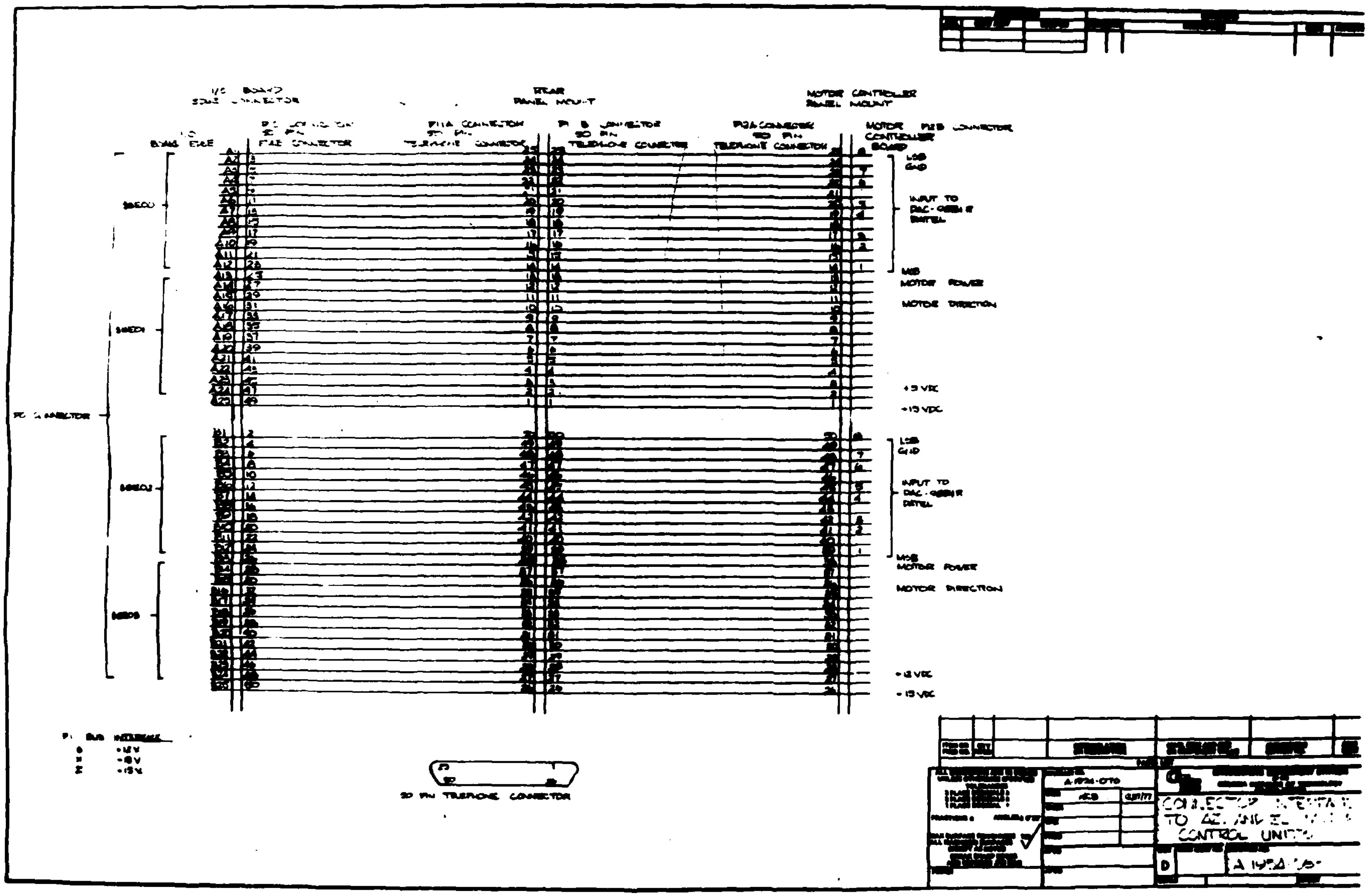
Drawing 75. Motor Controller Panel Wiring.



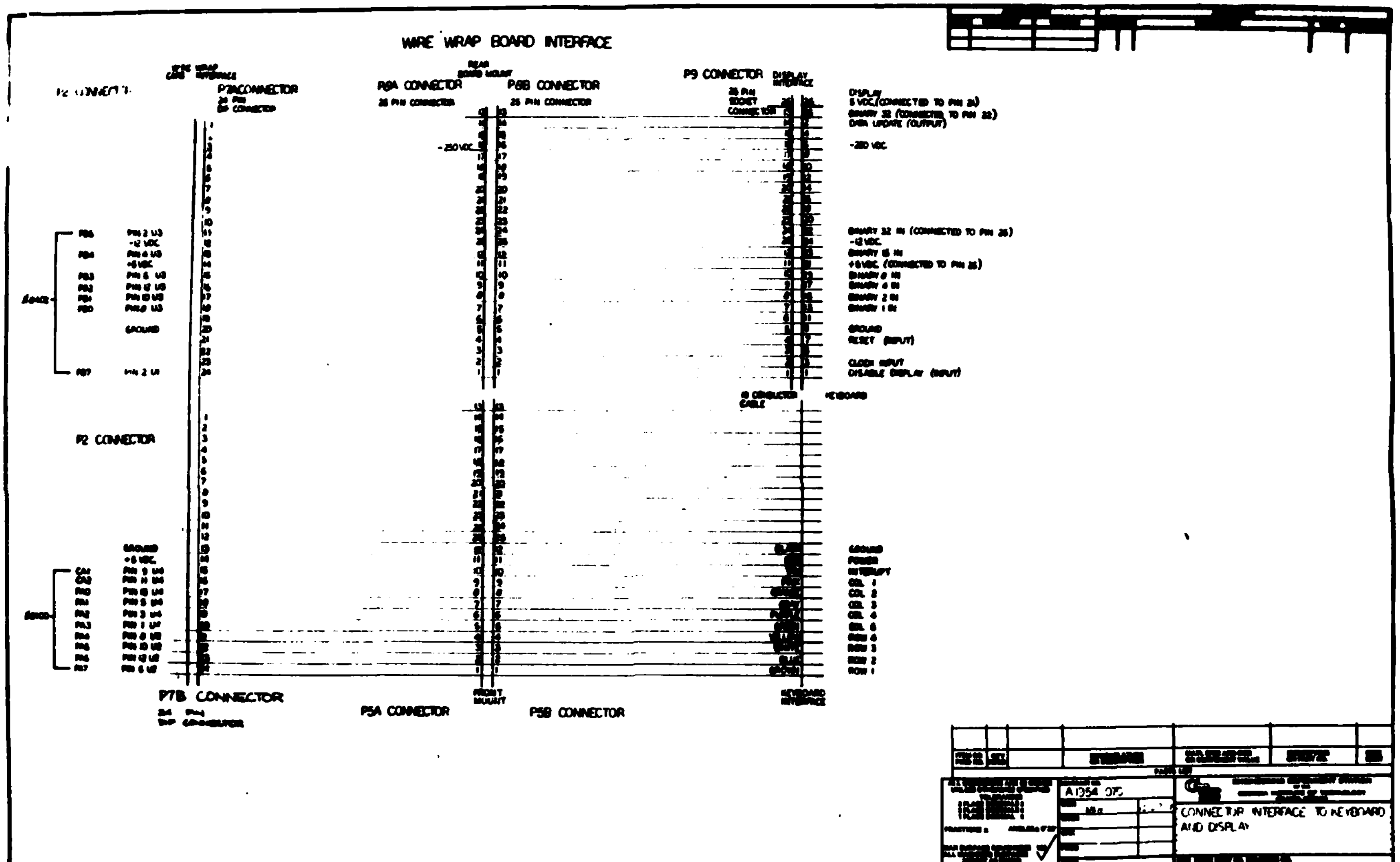
Drawing 76. Microcomputer Power Supply Schematic.



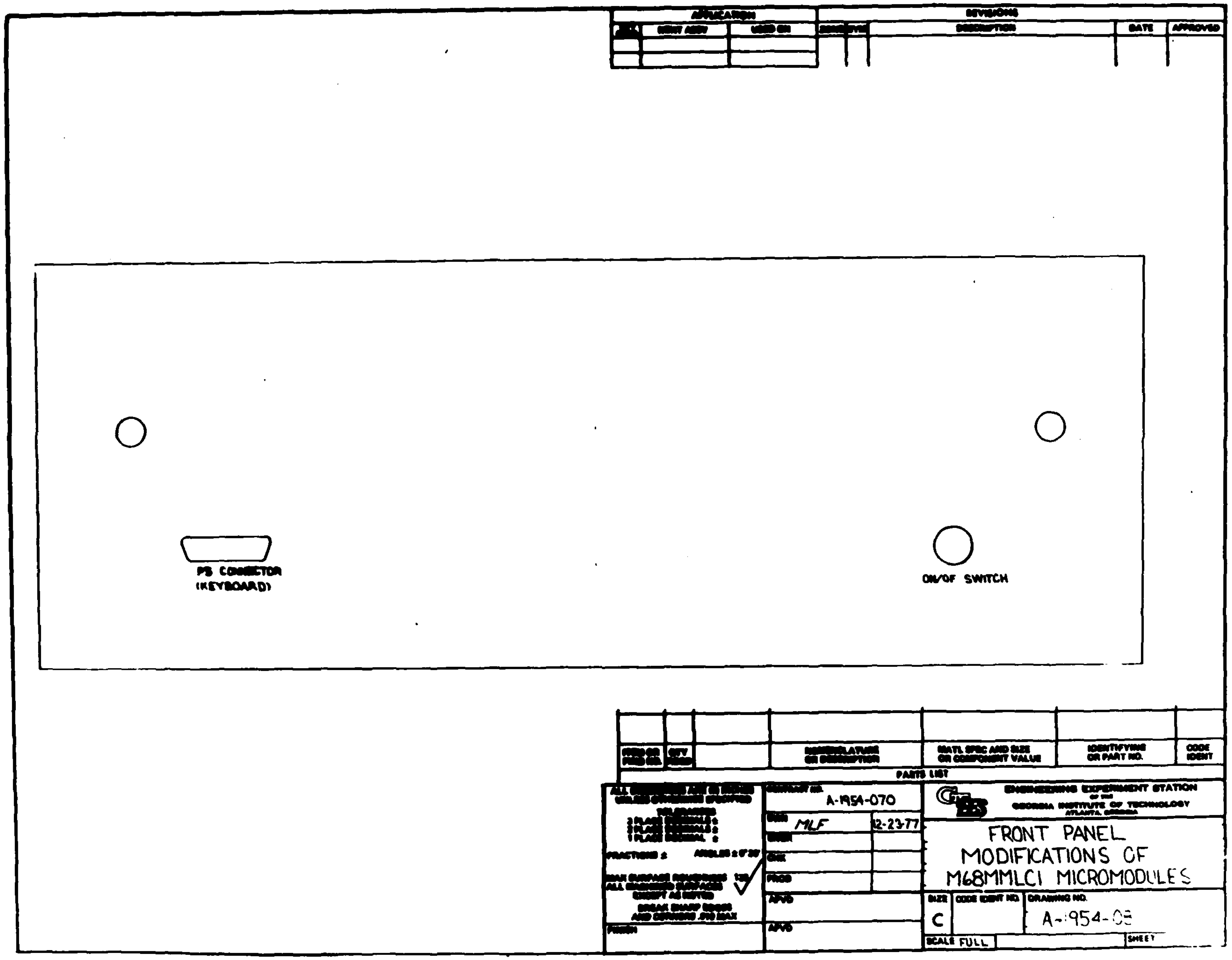
Drawing 77. Main Chassis Wiring.



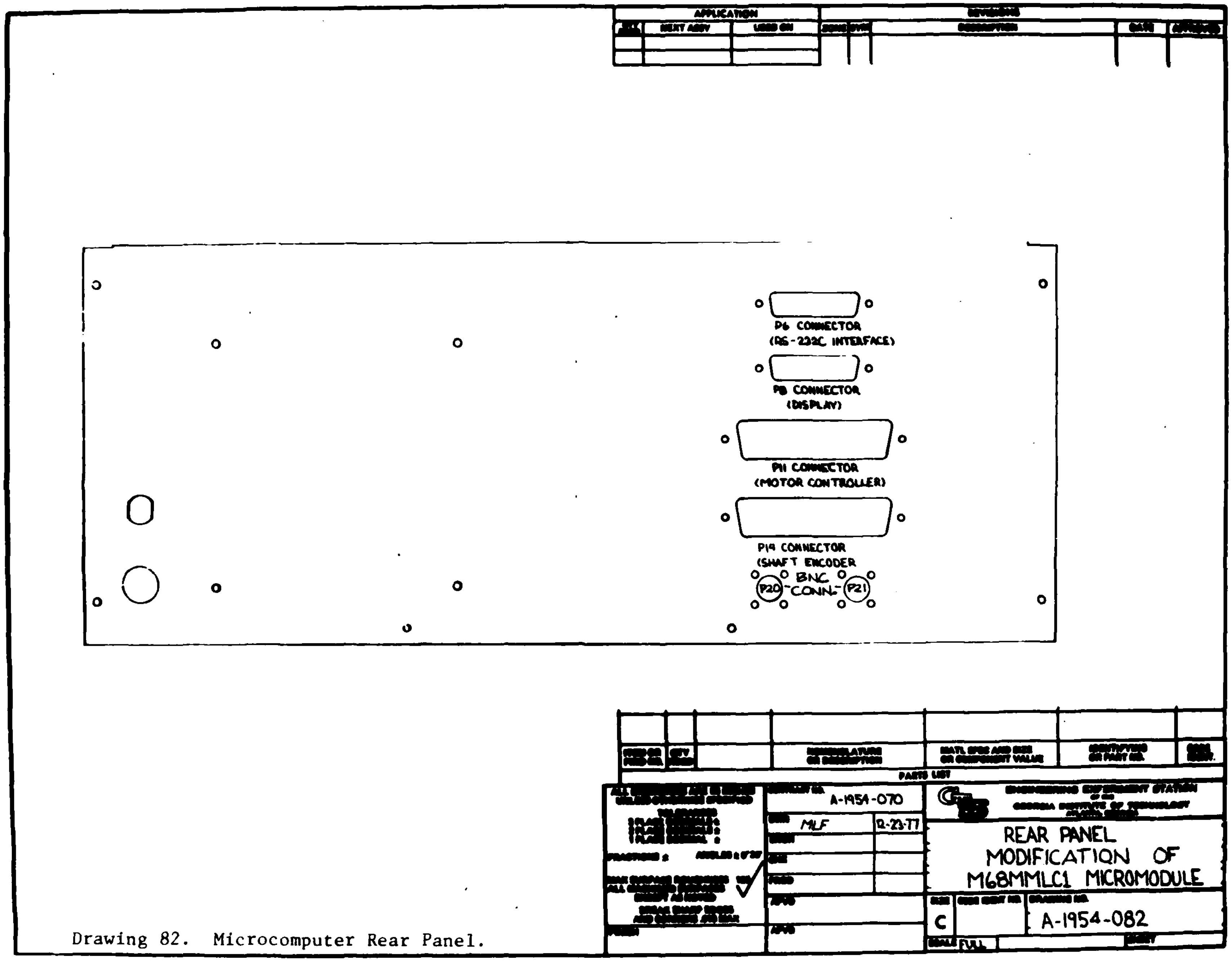
Drawing 79. Motor to Controller Cable.



Drawing 80. Keyboard and Display Cables.



Drawing 81. Microcomputer Front Panel.



Drawing 82. Microcomputer Rear Panel.

APPENDIX B

RADOME POSITIONER SOFTWARE LISTING

```

                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                    ;                                          ;
                    ;     RADOME POSITIONER FOR THE (RFSS)     ;
                    ;     RADIO FREQUENCY SIMULATION SYSTEM    ;
                    ;     SUB-TASK-A-1954-070                  ;
                    ;     WRITTEN BY ROBERT W. BIRD            ;
                    ;     USING THE M6800 MICROPROCESSOR       ;
                    ;     VERSION 1.1                          ;
                    ;                                          ;
                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      0000                 ORG    0000H
0000  0014          DISBUF BLOCK  20
0014  0015          SIBUF  BLOCK  21
0029  0002          TEMPX  BLOCK  2
002B  0001          CHARBF BLOCK  1
002C  0002          CHARPT BLOCK  2
002E  0002          CHARCT BLOCK  2
0030  0001          CHRNUM BLOCK  1
0031  0001          SAVEA  BLOCK  1
0032  0002          SAVEX  BLOCK  2
0034  0002          SAVEX1 BLOCK  2
0036  0001          TEMPA  BLOCK  1
0037  0001          TEMPB  BLOCK  1
0038  0001          MSBENC BLOCK  1
0039  0001          LSBENC BLOCK  1
003A  0001          LETA   BLOCK  1
003B  0001          LETB   BLOCK  1
003C  0001          BCDA   BLOCK  1
003D  0001          BCDB   BLOCK  1
003E  0001          SAVDEC BLOCK  1
003F  0003          ANGLE  BLOCK  3
0042  0001          SIGN   BLOCK  1
0043  0001          AZSIGN BLOCK  1
0044  0001          ELSIGN BLOCK  1
0045  0002          AZBCD  BLOCK  2
0047  0002          ELBCD  BLOCK  2
0049  0002          TEMPX1 BLOCK  2
004B  0002          TEMPX2 BLOCK  2
004D  0001          ENTRYA BLOCK  1
004E  0001          ENTRYB BLOCK  1
004F  0001          KEYENT BLOCK  1
0050  0001          TEMPA1 BLOCK  1
0051  0001          TEMPB1 BLOCK  1
0052  0001          KEYC   BLOCK  1
0053  0002          AZKEY  BLOCK  2
0055  0002          ELKEY  BLOCK  2
0057  0001          AZKEYS BLOCK  1
0058  0001          ELKEYS BLOCK  1
0059  0001          MFLAG  BLOCK  1
005A  0001          MINUEN BLOCK  1
005B  0001          SPEEDA BLOCK  1
005C  0001          SPEEDE BLOCK  1
```

```
005D 0002      AZMAG  BLOCK   2
005F 0002      ELMAG  BLOCK   2
0061 0001      AZEL   BLOCK   1
0062 0001      TEMPS  BLOCK   1
0063 0001      PROGN  BLOCK   1
0064 0001      PROGL  BLOCK   1
0065 0002      PROGA  BLOCK   2
0067 0002      PROGB  BLOCK   2
0069 0002      PROGC  BLOCK   2
006B 0001      KFLAG  BLOCK   1
006C 0001      DFLAGA BLOCK   1
006D 0001      DFLAGE BLOCK   1
006E 0001      SFLAGA BLOCK   1
006F 0001      SFLAGE BLOCK   1
0070 0002      TEMPD  BLOCK   2
0072 0002      BCDVSR BLOCK   2
0074 0002      FPTEL  BLOCK   2
0076 0002      FPTAZ  BLOCK   2
0078 0001      FPTELS BLOCK   1
0079 0001      FPTAZS BLOCK   1
007A 0001      PROCNT BLOCK   1
007B 0002      STADDR BLOCK   2
007D 0002      PFLAG  BLOCK   2
007F 0002      PROANG BLOCK   2
0081 0002      BINANG BLOCK   2
0083 0002      SINE   BLOCK   2
0085 0002      COSINE BLOCK   2
0087 0001      SSIGN  BLOCK   1
0088 0001      CSIGN  BLOCK   1
0089 0001      SAVE1  BLOCK   1
008A 0001      BINUPR BLOCK   1
008B 0002      PELLIM BLOCK   2
008D 0002      NELLIM BLOCK   2
008F 0002      PAZLIM BLOCK   2
0091 0002      NAZLIM BLOCK   2
0093 0001      LFLAGE BLOCK   1
0094 0001      LFLAGA BLOCK   1
0095 0001      MEGFLG BLOCK   1
0096 0002      SAVEX2 BLOCK   2
0098 0001      CARRY  BLOCK   1
     00D0      HAFSPD EQU    0D0H
     00E0      QUASPD EQU    0E0H
     8404      DDRA2  EQU    08404H   ;MS 4 BITS OF DAC #1-AZIMUTH
     8405      CRA2   EQU    08405H
     8406      DDRB2  EQU    08406H   ;LS 8 BITS OF DAC #1-AZIMUTH
     8407      CRB2   EQU    08407H
     8800      DDRA3  EQU    08800H   ;MS 4 BITS OF DAC #2-ELEVATION
     8801      CRA3   EQU    08801H
     8802      DDRB3  EQU    08802H   ;LS 8 BITS OF DAC #2-ELEVATION
     8803      CRB3   EQU    08803H
     0000      DISAZ  EQU    00000H
     000A      DISEL  EQU    0000AH
     8E03      MEBSEL EQU    08E03H
```

```
      8E02        LSBSEL  EQU    08E02H
      8E01        MSBSAZ  EQU    08E01H
      8E00        LSBSAZ  EQU    08E00H
      8400        DDRA    EQU    08400H
      8401        CRA     EQU    08401H
      8402        DDRB    EQU    08402H
      8403        CRB     EQU    08403H
      8408        ACIAS   EQU    8408H     ;ACIA STATUS/CONTROL REGISTER
      8409        ACIAD   EQU    8409H     ;ACIA DATA REGISTER
      00F0        ETHSPD  EQU    0F0H
      2000                ORG    2000H
                  ;
                  ;       PIA INITIALIZATION
                  ;
2000  8E0FFF      GOE     LDS    #0FFFH
2003  0F                  SEI
2004  8603                LDA A  #3        ;"00000011" = MASTER RESET
2006  B78408              STA A  ACIAS     ;RESET ACIA
2009  8681                LDA A  #81H      ;"10000001" = 7 BITS, EVEN PARITY, AND 2 STOP BITS
200B  B78408              STA A  ACIAS     ;SET ACIA FOR RCVR INTERRUPT, TXMIT INTERRUPT OFF
200E  4F                  CLR A
200F  B78401              STA A  CRA       ;CLEARS CONTROL REGISTER A
2012  B78403              STA A  CRB       ;CLEARS CONTROL REGISTER B
2015  B78405              STA A  CRA2
2018  B78407              STA A  CRB2
201B  B78801              STA A  CRA3
201E  B78803              STA A  CRB3
2021  CE8400              LDX    #8400H
2024  86F0                LDA A  #0F0H
2026  A700                STA A  0,X
2028  8607                LDA A  #7
202A  A701                STA A  1,X
202C  860F                LDA A  #0FH
202E  A700                STA A  0,X
2030  A704                STA A  4,X
2032  86FF                LDA A  #0FFH
2034  A702                STA A  2,X
2036  A706                STA A  6,X
2038  B78800              STA A  DDRA3
203B  B78802              STA A  DDRB3
203E  8604                LDA A  #004H
2040  B78405              STA A  CRA2
2043  B78407              STA A  CRB2
2046  B78801              STA A  CRA3
2049  B78803              STA A  CRB3
204C  863D                LDA A  #03DH
204E  B78403              STA A  CRB       ;SELECTS OUTPUT REGISTER B
                  ;
                  ;       INITIALIZES MOTORS TO ZERO SPEED, ETC
                  ;
2051  CE0029              LDX    #TEMPX
2054  4F                  CLR A
2055  A700        NEXTC   STA A  0,X
```

```
2057 08                 INX
2058 8C0009             CPX     #CARRY+1
205B 26F8               BNE     NEXTC
205D CE3500             LDX     #3500H
2060 DF8B               STX     PELLIM
2062 DF8D               STX     NELLIM
2064 DF8F               STX     PAZLIM
2066 DF91               STX     NAZLIM
2068 860D               LDA A   #0DH      ;INITIALIZE (CR) IN SIBUF
206A 9729               STA A   SIBUF+21
206C 86FF               LDA A   #0FFH
206E B78E02             STA A   LSBSEL
2071 86FF               LDA A   #0FFH
2073 B78E00             STA A   LSBSAZ
2076 976E               STA A   SFLAGA    ;SETS AZ SPEED FLAG TO NOTE ZERO SPEED
2078 976F               STA A   SFLAGE    ;SETS EL SPEED FLAG TO NOTE ZERO SPEED
207A 8601               LDA A   #001H
207C B78E01             STA A   MSBSAZ    ;TURNS ON POWER TO AZ MOTOR
207F B78E03             STA A   MSBSEL    ;TURNS ON POWER TO EL MOTOR
                    ;
                    ;     INITIALIZES CONTROL LOOP SUCH THAT POSITIONER
                    ;     WILL NOT MOVE UPON POWER-UP.
                    ;     (MODIFICATION 1.1)
                    ;
2082 BD2AF8             JSR     SHAENC    ;READS ANGLES
2085 DE45               LDX     AZBCD
2087 DF53               STX     AZKEY     ;UPDATES AZIMUTH KEYENTRY WITH
2089 DE47               LDX     ELBCD     ;CURRENT AZIMUTH LOCATION
208B DF55               STX     ELKEY     ;UPDATES EL KEYENTRY WITH CURRENT EL LOCATION
208D 9643               LDA A   AZSIGN
208F 9757               STA A   AZKEYS    ;UPDATES AZIMUTH SIGN KEYENTRY WITH
2091 9644               LDA A   ELSIGN    ;CURRENT AZIMUTH SIGN STATUS
2093 9758               STA A   ELKEYS    ;UPDATES ELEVATION SIGN KEYENTRY WITH
                    ;                      CURRENT ELEVATION SIGN STATUS
2095 0E                 CLI               ;CLEARS INTERRUPT MASK, CPU NOW READY FOR INTERRUPTS
                    ;
                    ;     BEGIN STATE TABLE
                    ;
2096 CE3138       MSGA  LDX     #MSG12
2099 BD2C22             JSR     ASCDIS    ;DISPLAYS "THE GA. TECH-RFSS    "
209C C614               LDA B   #20
209E BD2E3E             JSR     WAITE     ;WAITE FOR 2 SECONDS
20A1 CE314C             LDX     #MSG13
20A4 BD2C22             JSR     ASCDIS    ;DISPLAYS "RADOME POSITIONER    "
20A7 C60A               LDA B   #10
20A9 BD2E3E             JSR     WAITE     ;WAIT FOR 1 SECOND
20AC CE3160             LDX     #MSG14
20AF BD2C22             JSR     ASCDIS    ;DISPLAYS "      VERSION 1.0    "
20B2 C60A         MSGB  LDA B   #10
20B4 BD2E3E             JSR     WAITE
                    ;
                    ;     BEGIN STATE ZERO, MAIN CONTROL LOOP
                    ;
```

```
20B7 966B    ST0    LDA A  KFLAG    ;IDLE STATE
20B9 2A0F           BPL    ST0A
20BB 7F006B         CLR    KFLAG    ;CLEARS KEY-PRESSED FLAG
20BE 964F           LDA A  KEYENT   ;GETS KEYCODE OF KEY PRESSED
20C0 CE31FF         LDX    #SP0     ;PUTS ZERO STATE POINTER IN INDEX REGISTER
20C3 BD2BE8         JSR    ADDCAL   ;JUMPS TO ROUTINE TO CALCULATE NEXT STATE
20C6 EE00           LDX    0,X
20C8 6E00           JMP    0,X
                    ;
                    ;      AZIMUTH MOTOR CONTROL LOOP
                    ;
20CA BD2AF8  ST0A   JSR    SHAENC   ;READS BOTH AZ AND EL ANGLES
20CD 9659           LDA A  MFLAG
20CF 2BE6           BMI    ST0
20D1 8600           LDA A  #00H
20D3 975B           STA A  SPEEDA   ;SET UP SPEED VARIABLE TO FULL SPEED
20D5 975C           STA A  SPEEDE
20D7 9657           LDA A  AZKEYS
20D9 9143           CMP A  AZSIGN   ;TEST TO SEE IF BOTH SIGNS ARE EQUAL
20DB 2712           BEQ    ST0X     ;BRANCH TO DO A BCDSUB IF SIGNS ARE THE SAME
20DD 9654           LDA A  AZKEY+1
20DF 9B46           ADD A  AZBCD+1  ; FIND LSBYTE OF AZ MAGNITUDE DIFF
20E1 19             DAA
20E2 975E           STA A  AZMAG+1
20E4 9653           LDA A  AZKEY
20E6 9945           ADC A  AZBCD    ;FIND MSBYTE OF AZ MAGNITUDE DIFF
20E8 19             DAA
20E9 D65E           LDA B  AZMAG+1
20EB 975D           STA A  AZMAG    ;AZ MAG DIFF NOW IN A AND B REGISTERS
20ED 200E           BRA    ST0X2
20EF 9653    ST0X   LDA A  AZKEY
20F1 D654           LDA B  AZKEY+1
20F3 CE0045         LDX    #AZBCD   ;PUT ADDRESS OF BCD CURRENT LOCATION IN INDEX REG.
20F6 BD2AC4         JSR    BCDSUB   ;JUMPS TO ROUTINE TO SUBTRACT BCD NUMBERS
20F9 975D           STA A  AZMAG
20FB D75E           STA B  AZMAG+1
                    ;
                    ;      ADDITION TO TIGHTEN CONTROL LOOP (AZIMUTH) TO .1 DEGREE
                    ;      (MODIFICATION 1.1)
                    ;
20FD 8100    ST0X2  CMP A  #00H     ;START <0.2 DEGREE TEST
20FF 260D           BNE    ST0X1    ;BRANCHES TO <0.5 DEGREE TEST IF BCD WORD NOT <0.2
2101 C120           CMP B  #20H     ;COMPARING TO 00.2 DEGREE
2103 2209           BHI    ST0X1    ;BRANCHES TO <0.5 DEGREE TEST IF BCD WORD NOT <0.2
2105 86FF           LDA A  #0FFH    ;CURRENT POSITION IS LESS THAN 0.2 DEGREE
2107 976E           STA A  SFLAGA   ;SETS AZ SPEED FLAG WITH CORRECT SPEED
2109 B78E00         STA A  LOBJAZ   ;STOP AZ MOTOR WITH ZERO SPEED
210C 204A           BRA    ST0E
210E 8100    ST0X1  CMP A  #00H     ;START <0.5 DEGREES TEST
2110 260A           BNE    ST0B     ;BRANCHES TO <5.0 DEG. TEST IF BCD WORD NOT <0.5
2112 C150           CMP B  #050H
2114 2206           BHI    ST0B     ;BRANCHES TO <5.0 DEG. TEST IF BCD WORD NOT <0.5
2116 86F0           LDA A  #ETHSPD  ;SET SPEED TO EIGHTH SPEED
```

```
2118 975B          STA A  SPEEDA  ;SET UP SPEED VARIABLE FOR USE LATER
211A 2012          BRA    STOD
                 :
211C 8104   STOB   CMP A  #004H   ;START TEST FOR <5.0 DEG
211E 2206          BHI    STOC    ;BRANCHES TO <10.0 TEST IF <5.0 TEST FAILS
2120 86E0          LDA A  #QUASPD ; SET SPEED TO QUARTER SPEED
2122 975B          STA A  SPEEDA  ;SET UP SPEED VARIABLE FOR USE LATER
2124 2038          BRA    STOD    ;BRANCH TO DECISION ROUTINE
2126 8109   STOC   CMP A  #009H   ;START <10.0 DEG. TEST
2128 2204          BHI    STOD    ;BRANCH TO DECISION ROUTINE IF TEST FAILS
212A 86D0          LDA A  #HAFSPD ;SET SPEED TO HALF SPEED
212C 975B          STA A  SPEEDA  ;SETS UP SPEED VARIABLE TO BE USED LATER
212E 9657   STOD   LDA A  AZKEYS  ;DESTINATION NOT REACHED, CHECKS SIGNS
2130 9143          CMP A  AZSIGN
2132 270A          BEQ    SAMEAZ
2134 812B   DIFFAZ CMP A  #02BH   ;DIFFERENT SIGNS
2136 2703          BEQ    B2
2138 7E2C0E        JMP    LEFT
213B 7E2C18 B2     JMP    RIGHT
213E 812B   SAMEAZ CMP A  #02BH   ;SAME SIGNS, WHICH ONE IS PLUS
2140 270B          BEQ    YESAZ
2142 7D0098 NOAZ   TST    CARRY
2145 2B03          BMI    B3
2147 7E2C0E        JMP    LEFT
214A 7E2C18 B3     JMP    RIGHT
214D 7D0098 YESAZ  TST    CARRY
2150 2B03          BMI    B4
2152 7E2C18        JMP    RIGHT
2155 7E2C0E B4     JMP    LEFT
                 :
                 ;        ELEVATION MOTOR CONTROL LOOP
                 :
2158 9658   STOE   LDA A  ELKEYS
215A 9144          CMP A  ELSIGN  ;TEST TO SEE IF SIGNS ARE THE SAME
215C 2712          BEQ    STOY    ;BRANCHES IF SIGNS ARE NOT THE SAME
215E 9656          LDA A  ELKEY+1
2160 9B48          ADD A  ELBCD+1 ; FIND LSBYTE OF EL MAG. DIFFERENCE
2162 19            DAA
2163 9760          STA A  ELMAG+1
2165 9655          LDA A  ELKEY
2167 9947          ADC A  ELBCD   ;FIND MSBYTE OF EL MAG. DIFFERENCE
2169 19            DAA
216A D660          LDA B  ELMAG+1
216C 975F          STA A  ELMAG
216E 200E          BRA    STOY1
2170 CE0047 STOY   LDX    #ELBCD  ;PUT ADDRESS OF CURRENT ANGLE IN INDEX REG.
2173 9655          LDA A  ELKEY
2175 D656          LDA B  ELKEY+1
2177 BD2AC4        JSR    BCDSUB  ;JUMPS TO ROUTINE THAT SUBTRACTS BCD NUMBERS
217A 975F          STA A  ELMAG
217C D760          STA B  ELMAG+1
                 :
                 :        ADDITION TO TIGHTED CONTROL LOOP (ELEVATION) TO .1 DEGREE
```

```
                        :       (MODIFICATION 1.1)
                        :
217E 8100       STOY1   CMP A   #00H      ;START <0.2 DEGREE TEST
2180 260E               BNE     STOZ      ;BRANCHES TO <0.5 DEGREE TEST IF ANGLE GREATER THAN 0.2 DEGREE
2182 C120               CMP B   #20H      ;COMPARES ANGLE TO 00.20 DEGREES
2184 220A               BHI     STOZ      ;BRANCHES TO <0.5 DEGREE TEST IF ANGLE GREATER THAN 0.2 DEGREE
2186 86FF               LDA A   #0FFH     ;POSITIONER NEEDS TO STOP VERY, VERY SOON
2188 976F               STA A   SFLAGE    ;SETS SPEED FLAG
218A B78E02             STA A   LSBSEL    ;STOPS ELEVATION MOTOR WITH ZERO SPEED
218D 7E294C             JMP     CPFLAG    ;POSITON REACHED, CHECK PROGRAM FLAG
2190 8100       STOZ    CMP A   #000H     ;START <0.5 DEG. TEST
2192 260A               BNE     STOF      ;BRANCH TO <5.0 DEG. TEST IF COMPARE FAILS
2194 C150               CMP B   #050H     ;COMPLETES <0.5 DEG. TEST
2196 2206               BHI     STOF      ;BRANCH TO <5.0 DEG. IF ACCB IS PLUS
2198 86F0               LDA A   #ETHSPD   ;SETS MOTOR SPEED TO EIGHTH SPEED
219A 975C               STA A   SPEEDE    ;SETS SPEED VARIABLE FOR USE LATER
219C 2012               BRA     STOH
                        :
219E 8104       STOF    CMP A   #094H     ;START <5.0 DEG TEST
21A0 2206               BHI     STOG      ;BRANCH TO <10.0 DEG TEST IF ACCA IS PLUS
21A2 86E0               LDA A   #QUASPD
21A4 975C               STA A   SPEEDE    ;SET UP SPEED VARIABLE FOR EL MOTOR SPEED
21A6 2008               BRA     STOH      ;BRANCH TO DECISION ROUTINE
21A8 8109       STOG    CMP A   #099H     ;START <10.0 DEG. TEST
21AA 2204               BHI     STOH      ;BRANCH TO DECISION ROUTINE IF ACCA IS PLUS
21AC 86D0               LDA A   #HAFSPD
21AE 975C               STA A   SPEEDE    ;SET UP SPEED VARIABLE FOR HALF SPEED
21B0 9658       STOH    LDA A   ELKEYS
21B2 9144               CMP A   ELSIGN
21B4 2708               BEQ     SAMEEL
21B6 812B       DIFFEL  CMP A   #02BH
21B8 2702               BEQ     B5
21BA 2017               BRA     B7
21BC 2012       B5      BRA     B8
21BE 812B       SAMEEL  CMP A   #02BH
21C0 2709               BEQ     YESEL
21C2 7D0098     NOEL    TST     CARRY
21C5 2B02               BMI     B6
21C7 200A               BRA     B7
21C9 2005       B6      BRA     B8
21CB 7D0098     YESEL   TST     CARRY
21CE 2B03               BMI     B7
21D0 7E2BFA     B8      JMP     UP
21D3 7E2C04     B7      JMP     DOWN
                        :
                        :       BEGIN STATE ONE, MANUAL DOWN BUTTON
                        :
21D6 8600       ST1     LDA A   #00H      ;BEGIN STATE 1, MANUAL DOWN
21D8 C6FF               LDA B   #0FFH
21DA BD2DE2             JSR     MOTEL     ;JUMPS TO ROUTINE TO COMMAND ELEVATION MOTOR
21DD 8670       ST1A    LDA A   #70H
21DF B78400             STA A   DDRA
21E2 B68400             LDA A   DDRA
```

```
21E5 8177          CMP A  #77H    ;HAS DOWN KEY BEEN LET UP
21E7 2608          BNE    ST1B    ;DOWN KEY IS NOT BEING PRESSED NOW
21E9 BD2936        JSR    RESTO   ;RESTORE KEYBOARD BEFORE READING ANGLES (MOD 1.1)
21EC BD2AF8        JSR    SHAENC  ;DOWN KEY HAS NOT BEEN LET UP, SO READ ANGLES
21EF 20EC          BRA    ST1A
21F1 7F0093  ST1B  CLR    LFLAGE  ;CLEAR LIMIT REACHED FLAG (MOD 1.1)
21F4 86FF          LDA A  #0FFH   ;TURN OFF MOTOR
21F6 C6FF          LDA B  #0FFH   ;CLOCKWISE MOTION STILL SET
21F8 BD2DE2        JSR    MOTEL
21FB BD2936        JSR    RESTO   ;RESTORE KEYBD PIA. THEN BACK TO STATE ZERO
21FE 7E20B7        JMP    ST0
                   ;
                   ;      BEGIN STATE TWO, MANUAL UP BUTTON
                   ;
2201 8600    ST2   LDA A  #000H   ;BEGIN STATE TWO, LOAD A WITH SPEED
2203 C600          LDA B  #000H   ;LOAD B WITH DIRECTION
2205 BD2DE2        JSR    MOTEL
2208 8670    ST2A  LDA A  #070H
220A B78400        STA A  DDRA
220D B68400        LDA A  DDRA
2210 817B          CMP A  #07BH   ;HAS UP KEY BEEN RELEASED?
2212 2608          BNE    ST2B
2214 BD2936        JSR    RESTO   ;RESTORE KEYBOARD BEFORE READING ANGLES (MOD 1.1)
2217 BD2AF8        JSR    SHAENC  ;UP KEY IS STILL BEING PRESSED, READ ANGLES
221A 20EC          BRA    ST2A
221C 7F0093  ST2B  CLR    LFLAGE  ;CLEAR LIMIT REACHED FLAG
221F 86FF          LDA A  #0FFH
2221 C600          LDA B  #000H   ;C-CLOCKWISE MOTION SET
2223 BD2DE2        JSR    MOTEL
2226 BD2936        JSR    RESTO   ;RESTORE KEYBD PIA, THEN BACK TO STATE ZERO
2229 7E20B7        JMP    ST0
                   ;
                   ;      BEGIN STATE THREE, MANUAL LEFT BUTTON
                   ;
222C 8600    ST3   LDA A  #000H   ;BEGIN STATE THREE, LOAD A WITH SPEED
222E C600          LDA B  #000H   ;LOAD B WITH DIRECTION
2230 BD2D86        JSR    MOTAZ   ;JUMP TO ROUTINE TO COMMAND AZIMUTH MOTOR
2233 86B0    ST3A  LDA A  #0B0H
2235 B78400        STA A  DDRA
2238 B68400        LDA A  DDRA
223B 81B7          CMP A  #0B7H   ;HAS RIGHT KEY BEEN RELEASED?
223D 2608          BNE    ST3B
223F BD2936        JSR    RESTO   ;RESTORE KEYBOARD BEFORE READING ANGLES (MOD 1.1)
2242 BD2AF8        JSR    SHAENC  ;KEY HAS NOT BEEN RELEASED, SO READ ANGLES
2245 20EC          BRA    ST3A
2247 7F0094  ST3B  CLR    LFLAGA  ;CLEAR LIMIT REACHED FLAG (MOD 1.1)
224A 86FF          LDA A  #0FFH   ;TURN AZIMUTH MOTOR OFF
224C C600          LDA B  #000H
224E BD2D86        JSR    MOTAZ
2251 BD2936        JSR    RESTO   ;RESTORE KEYBD PIA, THEN BACK TO STATE ZERO
2254 7E20B7        JMP    ST0
                   ;
                   ;      BEGIN STATE FOUR, MANUAL RIGHT BUTTON
```

```
                        ;
2257 8600       ST4     LDA A   #000H    ;BEGIN STATE FOUR, LOAD A WITH SPEED
2259 C6FF               LDA B   #0FFH    ;LOAD B WITH DIRECTION
225B BD2D86             JSR     MOTAZ
225E 86B0       ST4A    LDA A   #0B0H
2260 B78400             STA A   DDRA
2263 B68400             LDA A   DDRA
2266 81BB               CMP A   #0BBH    ;HAS LEFT KEY BEEN RELEASED?
2268 2608               BNE     ST4B
226A BD2936             JSR     RESTO    ;RESTORE KEYBOARD BEFORE READING ANGLES (MOD 1.1)
226D BD2AF8             JSR     SHAENC   ;LEFT KEY HAS NOT BEEN RELEASED, SO READ ANGLES
2270 20EC               BRA     ST4A
2272 7F0094     ST4B    CLR     LFLAGA   ;CLEAR LIMIT REACHED FLAG (MOD 1.1)
2275 86FF               LDA A   #0FFH
2277 C6FF               LDA B   #0FFH    ;CLOCKWISE MOTION
2279 BD2D86             JSR     MOTAZ
227C BD2936             JSR     RESTO    ;RESTORE KEYBD PIA, THEN BACK TO STATE ZERO
227F 7E20B7             JMP     ST0
                        ;
                        ;       BEGIN STATE FIVE, ERROR STATE
                        ;
2282 CE3098     ST5     LDX     #MSG4    ;PRINTS "ERROR..INVALID ENTRY"
2285 BD2C22             JSR     ASCDIS   ;PRINTS ASCII MESSAGE
2288 C60A               LDA B   #10
228A BD2E3E             JSR     WAITE
228D 7E20B7             JMP     ST0      ;BACK TO STATE ZERO
                        ;
                        ;       BEGIN STATE SIX, ERROR STATE
                        ;
2290 CE30E8     ST6     LDX     #MSG3
2293 BD2C22             JSR     ASCDIS   ;PRINTS "ANGLE TOO LARGE....."
2296 7E20B2             JMP     MSGB     ;WAIT 1 SECOND, THEN RETURN TO CONTROL LOOP
                        ;
                        ;       BEGIN STATE SEVEN, MESSAGE STATE
                        ;
2299 CE3124     ST7     LDX     #MSG11
229C BD2C22             JSR     ASCDIS   ;PRINTS "POSITIONER HALTED    "
229F 7E20B2             JMP     MSGB     ;WAIT 1 SECOND, THEN RETURN TO CONTROL LOOP
                        ;
                        ;       BEGIN STATE TEN, SET AZ
                        ;
22A2 9761       ST10    STA A   AZEL     ;REMEMBERS WHICH KEYCODE WAS PRESSED, SETAZ OR SETEL
22A4 CE3070             LDX     #MSG2
22A7 BD2C22             JSR     ASCDIS   ;PRINTS "ENTER AZIMUTH ANGLE"
22AA CE000B             LDX     #DISEL+1
22AD DF32               STX     SAVEX    ;KEEPS TRACK OF WHERE THINGS ARE ON THE DISPLAY
22AF 966B       ST10A   LDA A   KFLAG
22B1 2AFC               BPL     ST10A
22B3 7F006B             CLR     KFLAG    ;CLEARS KEYENTRY FLAG
22B6 CE30C0             LDX     #MSG6
22B9 BD2C22             JSR     ASCDIS   ;PRINTS "AZIMUTH              "
22BC 7F004D             CLR     ENTRYA   ;CLEARS BOTH REGISTERS TO BE USED WHEN PACKING ENTRIES
22BF 7F004E             CLR     ENTRYB
```

```
2202 964F            LDA A  KEYENT   ;GETS KEYENTRY
2204 CE3227          LDX    #SP10
2207 BD2BE8          JSR    ADDCAL
220A EE00            LDX    0,X
220C 6E00            JMP    0,X      ;JUMPS TO NEXT STATE DETEMINED BY KEYENTRY IN A
                 ;
                 ;      BEGIN STATE ELEVEN, SET EL
                 ;
22CE 9761    ST11    STA A  AZEL     ;REMEMBERS WHICH KEYENTRY WAS MADE...SET EL
22D0 CE305C          LDX    #MSG1
22D3 BD2C22          JSR    ASCDIS   ;PRINTS "ENTR ELEVATION ANGLE"
22D6 CE000B          LDX    #DISEL+1
22D9 DF32            STX    SAVEX    ;KEEPS TRACK OF WHERE THINGS ARE ON THE DISPLAY
22DB 966B    ST11A   LDA A  KFLAG    ;WAITS FOR NEXT KEYENTRY
22DD 2AFC            BPL    ST11A
22DF 7F006B          CLR    KFLAG    ;CLEARS KEYENTRY FLAG
22E2 CE30D4          LDX    #MSG7
22E5 BD2C22          JSR    ASCDIS   ;PRINTS "ELEVATION        "
22E8 7F004D          CLR    ENTRYA   ;CLEARS BOTH REGISTERS WITH WHICH THE KEYENTRIES ARE PACKED
22EB 7F004E          CLR    ENTRYB
22EE 964F            LDA A  KEYENT   ;GETS KEYENTRY
22F0 CE324F          LDX    #SP11
22F3 BD2BE8          JSR    ADDCAL   ;CALCULATES THE ADDRESS OF THE NEXT STATE
22F6 EE00            LDX    0,X
22F8 6E00            JMP    0,X      ;JUMPS TO NEXT STATE AS DETERMINED BY LAST KEYENTRY
                 ;
                 ;      BEGIN STATE TWELVE, DISPLAYS ENTERED PLUS SIGN AND FIRST NUMBER
                 ;
22FA C62B    ST12    LDA B  #02BH    ;BEGINS STATE TWELVE, PLUS SIGN AND MAGNITUDE
22FC DE32            LDX    SAVEX
22FE E700            STA B  0,X      ;DISPLAYS PLUS SIGN
2300 08              INX             ;INCREMENTS TRACKING POINTING
2301 D762            STA B  TEMPS    ;REMEMBERS THE SIGN OF THE KEYENTRY
2303 44              LSR A           ;CONVERTS KEYCODE TO BCD CODE
2304 16              TAB
2305 BD2A9E          JSR    PACK     ;ROUTINE TO PACK KEYENTRY
2308 CB30            ADD B  #030H    ;CONVERTS BCD CODE TO ASCII CODE
230A E700            STA B  0,X      ;ECHOES KEYENTRY ON THE DISPLAY
230C 08              INX             ;INCREMENTS TRACKING POINTER
230D DF32            STX    SAVEX    ;REMEMBERS NEW VALUE OF TRACKING POINTER
230F 966B    ST12A   LDA A  KFLAG    ;WAITS FOR ANOTHER KEYENTRY
2311 2AFC            BPL    ST12A
2313 7F006B          CLR    KFLAG    ;CLEARS KEYENTRY FLAG
2316 964F            LDA A  KEYENT   ;GETS KEYENTRY
2318 CE3277          LDX    #SP12    ;LOADS INDEX REGISTER WITH STATE 12 POINTER
231B BD2BE8          JSR    ADDCAL   ;CALCULATES THE NEXT ADDRESS
231E EE00            LDX    0,X
2320 6E00            JMP    0,X      ;JUMPS TO THE NEXT STATE
                 ;
                 ;      BEGIN STATE THIRTEEN, DISPLAYS ENTERED MINUS SIGN
                 ;
2322 C62D    ST13    LDA B  #02DH
2324 DE32            LDX    SAVEX
```

```
2326 E700          STA B  0,X     ;DISPLAYS THE ENTERED MINUS SIGN
2328 08            INX            ;INCREMENTS TRACKING POINTER
2329 DF32          STX    SAVEX   ;STORES TRACKING POINTER
232B D762          STA B  TEMPS   ;REMEMBERS THE SIGN
232D 966B   ST13A  LDA A  KFLAG   ;WAITS FOR NEXT KEYENTRY
232F 2AFC          BPL    ST13A
2331 7F006B        CLR    KFLAG   ;CLEARS KEYENTRY FLAG
2334 964F          LDA A  KEYENT  ;GETS KEYENTRY
2336 CE329F        LDX    #SP13
2339 BD2BE8        JSR    ADDCAL
233C EE00          LDX    0,X
233E 6E00          JMP    0,X     ;JUMPS TO NEXT STATE
                   ;
                   ;      BEGIN STATE FORTEEN, DISPLAYS ENTERED FIRST NUMBER AFTER MINUS SIGN
                   ;
2340 44     ST14   LSR A
2341 16            TAB
2342 BD2A9E        JSR    PACK    ;PACKS THE ENTERED BCD NUMBERS
2345 CB30          ADD B  #030H
2347 DE32          LDX    SAVEX
2349 E700          STA B  0,X     ;ECHOES THE ENTERED BCD CODE
234B 08            INX            ;INCREMENTS TRACKING POINTER
234C DF32          STX    SAVEX   ;KEEPS TRACK OF POINTER
234E 966B   ST14A  LDA A  KFLAG
2350 2AFC          BPL    ST14A
2352 7F006B        CLR    KFLAG   ;CLEARS KEYENTRY FLAG
2355 964F          LDA A  KEYENT  ;GETS KEYENTRY
2357 CE32C7        LDX    #SP14   ;LOADS INDEX REGISTER WITH STATE 14 POINTER
235A BD2BE8        JSR    ADDCAL
235D EE00          LDX    0,X
235F 6E00          JMP    0,X     ;JUMPS TO NEXT STATE AS DETERMINED BY LAST KEYENTRY
                   ;
                   ;      BEGIN STATE FIFTEEN, DISPLAYS ENTERED SECOND NUMBER AFTER EITHER + OR -
                   ;
2361 44     ST15   LSR A          ;CONVERTS KEYCODE TO BCD CODE
2362 16            TAB
2363 BD2A9E        JSR    PACK    ;PACKS ENTERED BCD NUMBERS
2366 CB30          ADD B  #030H   ;CONVERTS BCD TO ASCII
2368 DE32          LDX    SAVEX
236A E700          STA B  0,X     ;ECHOES THE ENTERED BCD NUMBER
236C 08            INX            ;INCREMENTS TRACKING POINTER
236D DF32          STX    SAVEX   ;STORES TRACKING POINTER
236F 966B   ST15A  LDA A  KFLAG
2371 2AFC          BPL    ST15A
2373 7F006B        CLR    KFLAG   ;CLEARS KEYENTRY FLAG
2376 964F          LDA A  KEYENT  ;GETS KEYENTRY
2378 CE32FF        LDX    #SP15
237B BD2BE8        JSR    ADDCAL  ;CALCULATES NEXT ADDRESS
237E EE00          LDX    0,X
2380 6E00          JMP    0,X     ;JUMPS TO NEXT STATE
                   ;
                   ;      BEGIN STATE SIXTEEN, DISPLAYS ENTERED DECIMAL POINT
                   ;
```

```
2382 C62E     ST16   LDA B  #02EH
2384 DE32            LDX    SAVEX
2386 E700            STA B  0,X      ;ECHOES THE DECIMAL POINT
2388 08              INX             ;INCREMENTS TRACKING POINTER
2389 DF32            STX    SAVEX    ;STORES TRACKING POINTER
238B 966B     ST16A  LDA A  KFLAG
238D 2AFC            BPL    ST16A
238F 7F006B          CLR    KFLAG    ;CLEARS KEYENTRY FLAG
2392 964F            LDA A  KEYENT   ;GETS KEYENTRY
2394 CE3317          LDX    #SP16
2397 BD2BE8          JSR    ADDCAL   ;CALCULATES NEXT ADDRESS
239A EE00            LDX    0,X
239C 6E00            JMP    0,X      ;JUMPS TO NEXT STATE
              ;
              ;      BEGIN STATE SEVENTEEN, DISPLAYS LAST ENTERED NUMBER
              ;
239E 44       ST17   LSR A           ;BEGIN STATE 17, CONVERTS KEYCODE TO BCD CODE
239F 16              TAB
23A0 BD2A9E          JSR    PACK     ;PACKS BCD NUMBERS
23A3 CB30            ADD B  #030H    ;CONVERTS BCD CODE TO ASCII CODE
23A5 DE32            LDX    SAVEX
23A7 E700            STA B  0,X      ;ECHOES ENTERED BCD NUMBER
23A9 08              INX             ;INCREMENTS TRACKING POINTER
23AA B531FD          LDA A  DEGMAR
23AD A700            STA A  0,X      ;DISPLAYS DEGREE MARK AFTER LAST ENTERED NUMBER
23AF 08              INX             ;INCREMENTS TRACKING POINTER
23B0 DF32            STX    SAVEX    ;STORES TRACKING POINTER
23B2 9661            LDA A  AZEL
23B4 811E            CMP A  #01EH    ;TEST TO SEE IF SET EL WAS ENTERED
23B6 2711            BEQ    ST17A
23B8 9662            LDA A  TEMPS    ;STORES ENTERED DATA INTO APPROPRIATE AZ REGISTERS
23BA 9757            STA A  AZKEYS
23BC 964D            LDA A  ENTRYA
23BE D64E            LDA B  ENTRYB
23C0 BD295E          JSR    TSTANG   ;TEST FOR ENTRY ANGLE >40.1
23C3 DE4D            LDX    ENTRYA
23C5 2F53            STX    AZKEY
23C7 2011            BRA    ST17C
23C9 9662     ST17A  LDA A  TEMPS    ;STORES ENTERED DATA INTO APPROPRIATE EL REGISTERS
23CB 9758            STA A  ELKEYS
23CD 964D            LDA A  ENTRYA
23CF D64E            LDA B  ENTRYB
23D1 BD295E          JSR    TSTANG
23D4 DE4D            LDX    ENTRYA
23D6 DF55            STX    ELKEY
23D8 2000            BRA    ST17C
23DA 966B     ST17C  LDA A  KFLAG    ;WAITS FOR START KEY TO BE PRESSED
23DC 2AFC            BPL    ST17C
23DE 7F006B          CLR    KFLAG
23E1 964F            LDA A  KEYENT   ;GETS KEYENTRY
23E3 CE333F          LDX    #SP17    ;LOADS INDEX REGISTER WITH STATE 17 POINTER
23E6 BD2BE8          JSR    ADDCAL
23E9 EE00            LDX    0,X
```

```
23EB 6E00              JMP    0,X      ;JUMPS TO NEXT STATE
                :
                :      BEGIN STATE EIGHTEEN, GO TO MOTOR CONTROL LOOP
                :
23ED 7F0059     ST18   CLR    MFLAG    ;CLEARS MOTOR FLAG
23F0 7F0093            CLR    LFLAGE   ;CLEAR LIMIT REACHED FLAG
23F3 7F0094            CLR    LFLAGA
23F6 7E20B7            JMP    ST0
                :
                :      BEGIN STATE NINETEEN, DISABLES CONTROL LOOP
                :      (MODIFICATION 1.1)
                :
23F9 86FF       ST19   LDA A  #0FFH
23FB 9759              STA A  MFLAG    ;SETS THE MOTOR FLAG SO THAT THE CNT. LOOP IS DISABLED
23FD BD2A8B            JSR    ALSTOP
2400 7D0094            TST    LFLAGA   ;CHECH FOR AZIMUTH LIMIT REACHED
2403 2716              BEQ    ST19A    ;BRANCHES IF AZ LIMIT NOT REACHED
2405 4F                CLR A           ;AZ LIMIT HAS BEEN REACHED
2406 D66C              LDA B  DFLAGA   ;GET CURRENT DIRECTION STATUS
2408 53                COM B           ;GET OPPOSITE DIRECTION
2409 BD2D36            JSR    MOTAZ    ;AZ MOTOR CONTROL SUBROUTINE
240C C60A              LDA B  #10
240E BD2E3E            JSR    WAITE    ;ALLOW TIME FOR AZ MOTOR TO REPOSITION GIMBAL
2411 86FF              LDA A  #0FFH
2413 D66C              LDA B  DFLAGA
2415 BD2D36            JSR    MOTAZ    ;STOP MOTOR@D LIMIT NOT EXCEEDED ANYMORE
2418 7E20B7            JMP    ST0      ;GO BACK TO CONTROL LOOP
241B 7F0094     ST19A  CLR    LFLAGA
241E 7D0093            TST    LFLAGE   ;CHECK FOR ELEVATION LIMIT EXCEEDED
2421 2719              BEQ    ST19B    ;BRANCHES IF EL LIMIT NOT EXCEEDED
2423 4F                CLR A           ;EL LIMIT HAS BEEN REACHED
2424 D66D              LDA B  DFLAGE   ;GET CURRENT DIRECTION STATUS
2426 53                COM B           ;GET OPPOSITE DIRECTION
2427 BD2DE2            JSR    MOTEL    ;EL MOTOR CONTROL SUBROUTINE
242A C60A              LDA B  #10
242C BD2E3E            JSR    WAITE    ;ALLOW TIME FOR EL MOTOR TO REPOSITION GIMBAL
242F 86FF              LDA A  #0FFH
2431 D66D              LDA B  DFLAGE
2433 BD2DE2            JSR    MOTEL    ;STOP EL MOTOR@D LIMIT NOT EXCEEDED ANYMORE
2436 7F0093            CLR    LFLAGE
2439 7E20B7            JMP    ST0      ;RETURN CONTROL TO ST0
243C 7E2299     ST19B  JMP    ST7      ;DISPLAYS "POSITIONER HALTED    ", WAITS 1 SEC, THEN ST0
                :
                :      BEGIN STATE TWENTY, INPUT BCD PARAMETERS FOR PROGRAMMED RASTER SCANS
                :
243F CE30FC     ST20   LDX    #MSG9
2442 BD2C22            JSR    ASCDIS   ;PRINTS "ENTER PROG NUMBER"
2445 CE000F            LDX    #DISEL+3
2448 DF32              STX    SAVEX
244A 966B       ST20A  LDA A  KFLAG    ;WAITS FOR ANOTHER KEYENTRY
244C 2AFC              BPL    ST20A
244E 7F006B            CLR    KFLAG    ;CLEARS THE KEYENTRY FLAG
2451 7F004D            CLR    ENTRYA
```

```
2454 7F004E          CLR    ENTRYB
2457 CE3110          LDX    #MSG10
245A BD2C22          JSR    ASCDIS   ;PRINTS "PROG  @DENTER      "
245D 964F            LDA A  KEYENT
245F CE3367          LDX    #SP20
2462 BD2BE8          JSR    ADDCAL   ;CALCULATES NEXT ADDRESS
2465 EE00            LDX    0,X
2467 6E00            JMP    0,X
                ;
                ;     BEGIN STATE TWENTY-ONE, DISPLAYS PROGRAM NUMBER
                ;
2469 44         ST21 LSR A           ;CONVERTS KEYCODE TO BCD CODE
246A 8B30            ADD A  #030H
246C 9705            STA A  DISAZ+5 ; ECHOES THE ENTER PROGRAM NUMBER
246E 9763            STA A  PROGN   ;REMEMBERS THE PROGRAM NUMBER
2470 C641            LDA B  #041H
2472 D70D            STA B  DISEL+3 ; DISPLAYS "A" AFTER ENTER
2474 D764            STA B  PROGL
2476 966B      ST21A LDA A  KFLAG    ;WAITS FOR THE FIRST DIGIT OF A
2478 2AFC            BPL    ST21A
247A 7F006B          CLR    KFLAG    ;CLEARS THE KEYENTRY FLAG
247D 964F            LDA A  KEYENT   ;GETS THE NEXT KEYENTRY
247F CE338F          LDX    #SP21    ;LOADS THE INDEX REGISTER WITH STATE 21 POINTER
2482 BD2BE8          JSR    ADDCAL
2485 EE00            LDX    0,X
2487 6E00            JMP    0,X      ;JUMPS TO THE NEXT STATE
                ;
                ;     BEGIN STATE TWENTY-TWO, DISPLAYS FIRST NUMBER OF ONE
                ;     OF THREE INPUTS THAT MAKE UP THE RASTER SCAN
                ;
2489 44         ST22 LSR A           ;CONVERTS KEYCODE TO BCD CODE
248A 16              TAB
248B BD2A9E          JSR    PACK     ;PACKS THE BCD NUMBER
248E CB30            ADD B  #030H    ;CONVERTS BCD TO ASCII
2490 DE32            LDX    SAVEX
2492 E700            STA B  0,X
2494 08              INX
2495 DF32            STX    SAVEX
2497 966B      ST22A LDA A  KFLAG    ;WAITS FOR NEXT ENTRY
2499 2AFC            BPL    ST22A
249B 7F006B          CLR    KFLAG    ;CLEARS KEYENTRY FLAG
249E 964F            LDA A  KEYENT   ;GETS KEYENTRY
24A0 CE33B7          LDX    #SP22    ;LOADS INDEX REGISTER WITH STATE 22 POINTER
24A3 BD2BE8          JSR    ADDCAL   ;CALCULATES NEXT ADDRESS
24A6 EE00            LDX    0,X
24A8 6E00            JMP    0,X      ;JUMPS TO NEXT STATE
                ;
                ;     BEGIN STATE TWENTY-THREE, DISPLAYS SECOND NUMBER OF ONE
                ;     OF THREE INPUTS THAT MAKE UP THE RASTER SCAN
                ;
24AA 44         ST23 LSR A           ;BEGIN STATE 23, CONVERTS KEYCODE TO BCD CODE
24AB 16              TAB
24AC BD2A9E          JSR    PACK     ;PACKS ENTERED BCD NUMBERS
```

```
24AF CB30           ADD B  #030H
24B1 DE32           LDX    SAVEX
24B3 E700           STA B  0,X       ;ECHOES ENTERED BCD NUMBER
24B5 08             INX
24B6 DF32           STX    SAVEX
24B8 966B    ST23A  LDA A  KFLAG     ;WAITS FOR ANOTHER KEYENTRY
24BA 2AFC           BPL    ST23A
24BC 7F006B         CLR    KFLAG     ;CLEARS KEYENTRY FLAG
24BF 964F           LDA A  KEYENT    ;GETS KEYENTRY
24C1 CE33DF         LDX    #SP23     ;LOADS INDEX REGISTER WITH STATE 23 POINTER
24C4 BD2BE8         JSR    ADDCAL
24C7 EE00           LDX    0,X
24C9 6E00           JMP    0,X       ;JUMPS TO NEXT STATE
                    ;
                    ;      BEGIN STATE TWENTY-FOUR, DISPLAYS DECIMAL POINT
                    ;
24CB C62E    ST24   LDA B  #02EH
24CD DE32           LDX    SAVEX
24CF E700           STA B  0,X       ;ECHOES DECIMAL POINT
24D1 08             INX
24D2 DF32           STX    SAVEX
24D4 966B    ST24A  LDA A  KFLAG
24D6 2AFC           BPL    ST24A
24D8 7F006B         CLR    KFLAG     ;CLEARS KEYENTRY FLAG
24DB 964F           LDA A  KEYENT    ;GETS KEYENTRY
24DD CE3407         LDX    #SP24
24E0 BD2BE8         JSR    ADDCAL    ;CALCULATES NEXT ADDRESS
24E3 EE00           LDX    0,X
24E5 6E00           JMP    0,X       ;JUMPS TO NEXT STATE
                    ;
                    ;      BEGIN STATE TWENTY-FIVE, DISPLAYS LAST ENTERED NUMBER
                    ;      OF ONE OF THREE INPUTS THAT MAKE UP THE RASTER SCAN
                    ;
24E7 44      ST25   LSR A            ;BEGIN STATE 25, CONVERTS KEYCODE TO BCD CODE
24E8 16             TAB
24E9 BD2A0E         JSR    PACK      ;PACKS BCD NUMBERS
24EC CB30           ADD B  #030H     ;CONVERTS BCD TO ASCII
24EE DE32           LDX    SAVEX
24F0 E700           STA B  0,X       ;ECHOES ENTERED NUMBER
24F2 08             INX
24F3 F631FD         LDA B  DEGMAR
24F6 E700           STA B  0,X       ;INCLUDES DEGREE MARK
24F8 CE000F         LDX    #DISEL+5
24FB DF32           STX    SAVEX     ;STORES TRACKING POINTER
24FD 9664           LDA A  PROGL
24FF 8142           CMP A  #042H
2501 2734           BEQ    ST25B     ;BRANCH IF PROGL IS A "B"
2503 8143           CMP A  #043H
2505 2756           BEQ    ST25C     ;BRANCH IF PROGL IS A "C"
2507 964D           LDA A  ENTRYA
2509 D64E           LDA B  ENTRYB
250B 9765           STA A  PROGA
250D D766           STA B  PROGA+1
```

```
250F C60A          LDA B  #10
2511 BD2E3E        JSR    WAITE    ;WAIT 1 SECOND
                ;
                ;      ADDITION TO ST25.  CHECKS TO SEE IF PROG # 3 OR #4 IS
                ;      BEING IMPLEMENTED.
                ;      (MODIFICATION 1.1)
                ;
2514 D663          LDA B  PROGN    ;GET PROGRAM NUMBER
2516 C133          CMP B  #33H
2518 274B          BEQ    ST25A    ;BRANCHES IF STATE=PROGRAM #3
251A C134          CMP B  #34H
251C 2747          BEQ    ST25A    ;BRANCHES IF STATE=PROGRAM #4
                ;
251E CE3084        LDX    #MSG3
2521 BD2C22        JSR    ASCDIS   ;BLANKS THE DISPLAY
2524 CE3110        LDX    #MSG10
2527 BD2C22        JSR    ASCDIS   ;DISPLAYS "PROG  @DENTER        "
252A 9663          LDA A  PROGN
252C 9705          STA A  DISAZ+5 ; DISPLAYS THE PROGRAM NUMBER
252E 8642          LDA A  #042H
2530 970D          STA A  DISEL+3 ; DISPLAYS A "B" AFTER ENTER
2532 9764          STA A  PROGL    ;STORES PROGRAM LETTER
2534 7E2476        JMP    ST21A
2537 964D   ST25B  LDA A  ENTRYA
2539 D64E          LDA B  ENTRYB
253B 9767          STA A  PROGB
253D D768          STA B  PROGB+1 ; STORES WHAT WAS ENTERED AS "B"
253F C60A          LDA B  #10
2541 BD2E3E        JSR    WAITE    ;WAIT 1 SECOND
2544 CE3084        LDX    #MSG3
2547 BD2C22        JSR    ASCDIS   ;BLANKS THE DISPLAY
254A CE3110        LDX    #MSG10
254D BD2C22        JSR    ASCDIS   ;DISPLAYS "PROG  @DENTER        "
2550 9663          LDA A  PROGN
2552 9705          STA A  DISAZ+5 ; DISPLAYS PROGRAM NUMBER
2554 8643          LDA A  #043H
2556 970D          STA A  DISEL+3 ; DISPLAYS A "C"
2558 9764          STA A  PROGL    ;STORES THE CURRENT PROGRAM LETTER
255A 7E2476        JMP    ST21A    ;JUMPS TO ENTER "C"
255D 964D   ST25C  LDA A  ENTRYA
255F D64E          LDA B  ENTRYB
2561 9769          STA A  PROGC
2563 D76A          STA B  PROGC+1 ; STORES WHAT WAS ENTERED AS "C"
2565 966B   ST25A  LDA A  KFLAG
2567 2AFC          BPL    ST25A
2569 7F006B        CLR    KFLAG    ;CLEARS KEYENTRY FLAG
256C 964F          LDA A  KEYENT
256E CE342F        LDX    #SP25    ;LOADS INDEX REGISTER WITH STATE 25 POINTER
2571 BD2BE8        JSR    ADDCAL   ;CALCULATES THE NEXT ADDRESS
2574 EE00          LDX    0,X
2576 6E00          JMP    0,X
                ;
                ;      BEGIN STATE 00, PROGRAM DISTRIBUTION STATE
```

```
                        ;       (MODIFICATION 1.1)
                        ;
2578 9663       ST00    LDA A   PROGN   ;GET PROGRAM NUMBER
257A 8131               CMP A   #31H
257C 270B               BEQ     ST001   ;BRANCHES IF CURRENTLY IN PROG #1
257E 8132               CMP A   #32H
2580 270A               BEQ     ST002   ;BRANCHES IF CURRENTLY IN PROG #2
2582 8133               CMP A   #33H
2584 2709               BEQ     ST003   ;BRANCEHES IF CURRENTLY IN PROG #3
2586 7E27A1             JMP     ST29
2589 7E2592     ST001   JMP     ST26    ;GO TO PROGRAM #1
258C 7E2655     ST002   JMP     ST27    ;GO TO PROGRAM #2
258F 7E2717     ST003   JMP     ST28    ;GO TO PROGRAM #3
                        ;
                        ;       BEGIN STATE TWENTY-SIX, PATTERN NUMBER ONE
                        ;       (MODIFICATION 1.1)
                        ;
2592 7F0059     ST26    CLR     MFLAG
2595 9665               LDA A   PROGA   ;GETS TWO-BYTE RASTER PARAMETER " A "
2597 D666               LDA B   PROGA+1
2599 9753               STA A   AZKEY   ;ENTERS AZIMUTH PART OF FIRST POINT
259B D754               STA B   AZKEY+1
259D F631FA             LDA B   MINUS
25A0 D757               STA B   AZKEYS  ;ENTERS SIGN OF AZIMUTH PART OF FIRST POINT
25A2 9669               LDA A   PROCC
25A4 D66A               LBA B   PROCC+1
25A6 9774               STA A   FPTEL   ;SAVES TWO-BYTE ANSWER
25A8 D775               STA B   FPTEL+1
25AA 9755               STA A   ELKEY   ;ENTERS ELEVATION PART OF FIRST POINT
25AC D756               STA B   ELKEY+1
25AE F631FB             LDA B   PLUS
25B1 D758               STA B   ELKEYS  ;ENTER SGN OF ELEVATION PART OF FIRTST POINT
25B3 D778               STA B   FPTELS  ;SAVES VALUE OF SIGN
25B5 CE25C4             LDX     #ST26A
25B8 DF7B               STX     STADDR  ;SAVES RETURN ADDRESS
25BA 7F007A             CLR     PROCNT  ;CLEARS PROGRAM STATE COUNTER
25BD C6FF               LDA B   #0FFH
25BF D77D               STA B   PFLAG   ;SETS PROGRAM FLAG
25C1 7E20B7             JMP     STO     ;GO TO CONTROL LOOP, ANTICIPATE RETURN
25C4 D644       ST26A   LDA B   ELSIGN
25C6 F131FA             CMP B   MINUS   ;CHECKING CURRENT STATUS OF POSITION
25C9 260E               BNE     ST26A1  ;BRANCHES IF HAS RASTER IS NOT DONE
25CB 9674               LDA A   FPTEL
25CD D675               LDA B   FPTEL+1
25CF CE0047             LDX     #ELBCD
25D2 BD2AC4             JSR     BCDSUB  ;COMPARES CURRENT POSITION TO RASTER LIMIT
25D5 9698               LDA A   CARRY   ;IF CARRY=0, SUBTRACTION OVERFLOW
25D7 2B76               BMI     ST26E   ;BRANCHES IF RASTER LIMIT REACHED
25D9 D67A       ST26A1  LDA B   PROCNT
25DB C101               CMP B   #01H    ;CHECKS STATUS OF PROGRAM COUNTER
25DD 2718               BEQ     ST26D   ;BRANCHES IF THIRD POINT OF SCAN NEEDED
25DF C103               CMP B   #03
25E1 2714               BEQ     ST26B   ;BRANCHES OF FIFTH POINT OF SCAN NEEDED
```

```
25E3 9643            LDA A  AZSIGN
25E5 812B            CMP A  #2BH
25E7 2604            BNE    NEGS
25E9 862D            LDA A  #2DH
25EB 2002            BRA    FIN
25ED 862B     NEGS   LDA A  #2BH
25EF 9757     FIN    STA A  AZKEYS
25F1 7C007A          INC    PROCNT   ;INCREMENTS PROGRAM STATE COUNTER
25F4 7E20B7          JMP    STO      ;GO TO CONTROL LOOP, EXPECTING A RETURN
25F7 D644     ST26B  LDA B  ELSIGN
25F9 F131FB          CMP B  PLUS
25FC 261A            BNE    ST26B1   ;BRANCHES IF IN THE PLUS SIDE
25FE 9647            LDA A  ELBCD    ;ELSIGN IS MINUS, SO SUBTRACT
2600 D648            LDA B  ELBCD+1  ;GET CURRENT POSITION
2602 CE0067          LDX    #PROGB
2605 BD2AC4          JSR    BCDSUB
2608 9755            STA A  ELKEY    ;ENTER IN THIRD POINT
260A D756            STA B  ELKEY+1
260C 7D0098          TST    CARRY    ;TEST TO SEE WHICH NUMBER IS BIGGER
260F 2724            BEQ    ST26B2   ;BRANCHES IF ANSWER STILL PLUS
2611 B631FA          LDA A  MINUS    ;ANSWER IS MINUS
2614 9758            STA A  ELKEYS
2616 201D            BRA    ST26B2
2618 9648     ST26B1 LDA A  ELBCD+1  ;ELSIGN IS MINUS, SO ADD
261A 9B68            ADD A  PROGB+1
261C 19              DAA
261D 9756            STA A  ELKEY+1  ;ENTER IN THIRD POINT
261F 9647            LDA A  ELBCD
2621 9967            ADC A  PROGB
2623 19              DAA
2624 9755            STA A  ELKEY    ;ENTER IN THIRD POINT
2626 9655            LDA A  ELKEY
2628 D656            LDA B  ELKEY+1  ;CHECK TO SEE IF RASTER IS FINISHED
262A CE0074          LDX    #FPTEL
262D BD2AC4          JSR    BCDSUB
2630 7D0098          TST    CARRY    ;CHECH FOR SUBTRACTION OVERFLOW
2633 261A            BNE    ST26E    ;BRANCHES IF RASTER IS FINISHED
2635 7C007A   ST26B2 INC    PROCNT   ;INCREMENT PROGRAM STATE COUNTER
2638 967A            LDA A  PROCNT
263A 8104            CMP A  #04H     ;IS THIS CALCULATING 5TH POINT
263C 2603            BNE    ST26B3   ;BRANCHES IF CALCULATING 3TH POINT
263E 7F007A          CLR    PROCNT   ;CLEAR COUNTER TO RESTART SCAN PERIOD
2641 7E20B7   ST26B3 JMP    STO      ;GO TO CONTROL LOOP, EXPECT A RETURN
2644 B631FA   ST26C  LDA A  MINUS    ;CALCULATES 4TH POINT
2647 9757            STA A  AZKEYS
2649 7C007A          INC    PROCNT   ;INCREMENT PROGRAM STATE COUNTER
264C 7E20B7          JMP    STO      ;GO TO CONTROL LOOP, EXPECT A RETURN
264F 7F007D   ST26E  CLR    PFLAG    ;CLEAR PROGRAM FLAG
2652 7E20B7          JMP    STO      ;GO TO CONTROL LOOP, DONT COME BACK
              ;
              ;      BEGIN STATE TWENTY-SEVEN, PATTERN NUMBER TWO
              ;      (MODIFICATION 1.1)
              ;
```

```
00955 2655 7F0059   ST27    CLR     MFLAG
00956 2658 9665             LDA A   PROGA      ;GETS TWO-BYTE RASTER PARAMETER " A "
00957 265A D666             LDA B   PROGA+1
00958 265C 9776             STA A   FPTAZ      ;SAVES TWO-BYTE QUOTIENT
00959 265E D777             STA B   FPTAZ+1
00960 2660 9753             STA A   AZKEY      ;ENTERS AZIMUTH PART OF FIRST POINT
00961 2662 D754             STA B   AZKEY+1
00962 2664 F631FA           LDA B   MINUS
00963 2667 D757             STA B   AZKEYS     ;ENTERS SIGN OF AZIMUTH PART OF 1ST POINT
00964 2669 D779             STA B   FPTAZS     ;SAVES AZIMUTH SIGN
00965 266B 9669             LDA A   PROGC      ;GETS TWO-BYTE RASTER PARAMETER " C "
00966 266D D66A             LDA B   PROGC+1
00967 266F 9755             STA A   ELKEY      ;ENTERS ELEVATION PART OF FIRST POINT
00968 2671 D756             STA B   ELKEY+1
00969 2673 F631FB           LDA B   PLUS
00970 2676 D758             STA B   ELKEYS     ;ENTERS SIGN OF ELEVATION PART OF FIRST POINT
00971 2678 7F007A           CLR     PROCNT     ;CLEARS PROGRAM STATE COUNTER
00972 267B CE2687           LDX     #ST27A
00973 267E DF7B             STX     STADDR     ;SAVES RETURN ADDRESS
00974 2680 C6FF             LDA B   #0FFH
00975 2682 D77D             STA B   PFLAG      ;SETS PROGRAM FLAG
00976 2684 7E2037           JMP     STO        ;GO TO CONTROL LOOP, EXPECT A RETURN
00977 2687 D643     ST27A   LDA B   AZSIGN     ;RETURN HERE FROM CONTROL LOOP
00978 2689 F131FB           CMP B   PLUS       ;CHECKING CURRENT STATUS OF POSITION
00979 268C 260F             BNE     ST27A1
00980 268E 9676             LDA A   FPTAZ      ;GETS TWO-BYTE AZIMUTH PART OF FIRSTN PART
00981 2690 D677             LDA B   FPTAZ+1
00982 2692 CE0045           LDX     #AZBCD
00983 2695 BD2AC4           JSR     BCDSUB     ;COMPARES CURRENT POSITION TO RASTER LIMIT
00984 2698 7D0098           TST     CARRY      ;IF CARRY = 0. THEN OVERFLOW HAS OCCURRED
00985 269B 2674             BNE     ST27E      ;BRANCHES IF RASTER LIMIT REACHED
00986 269D D67A     ST27A1  LDA B   PROCNT     ;RASTER LIMIT NOT REACHED
00987 269F C101             CMP B   #01H       ;CHECKS STATUS OF PROGRAM STATE COUNTER
00988 26A1 2713             BEQ     ST27B      ;BRANCHES IF THIRD POINT OF SCAN NEEDED
00989 26A3 C102             CMP B   #02H
00990 26A5 275C             BEQ     ST27C      ;BRANCHES IF FORTH POINT OF SCAN NEEDED
00991 26A7 C103             CMP B   #03
00992 26A9 270B             BEQ     ST27B      ;BRANCHES IF FIFTH POINT OF SCAN NEEDED
00993 26AB F631FA           LDA B   MINUS
00994 26AE D758             STA B   ELKEYS     ;ENTERS AZIMUTH PART OF SECOND POINT
00995 26B0 7C007A           INC     PROCNT     ;INCREMENTS PROGRAM STATE COUNTER
00996 26B3 7E2037           JMP     STO        ;GO TO CONTROL LOOP, EXPECTING A RETURN
00997 26B6 D643     ST27B   LDA B   AZSIGN     ;HANDLES CHANGE IN COORDINATE SYSTEM
00998 26B8 F131FB           CMP B   PLUS
00999 26BB 271A             BEQ     ST27B1     ;BRANCHES IF IN THE PLUS SIDE
01000 26BD 9645             LDA A   AZBCD      ;AZSIGN IS MINUS, SO SUBTRACT
01001 26BF D646             LDA B   AZBCD+1    ;GET CURRENT POSITION
01002 26C1 CE0067           LDX     #PROCB      ;GET " B " VALUE
01003 26C4 BD2AC4           JSR     BCDSUB
01004 26C7 9753             STA A   AZKEY      ;ENTER THIRD POINT
01005 26C9 D754             STA B   AZKEY+1
01006 26CB 7D0098           TST     CARRY      ;TEST TO SEE WHICH ONE IS BIGGER
01007 26CE 2724             BEQ     ST27B2     ;BRANCHES IF ANSWER STILL PLUS
```

```
26D0 B631FB          LDA A  PLUS     ;ANSWER IS MINUS
26D3 9757            STA A  AZKEYS
26D5 201D            BRA    ST27B2   ;BRANCH AROUND ADD ROUTINE
26D7 9646     ST27B1 LDA A  AZBCD+1  ;AZSIGN IS PLUS, SO ADD
26D9 9B68            ADD A  PROGB+1
26DB 19              DAA
26DC 9754            STA A  AZKEY+1  ;ENTER IN THIRD POINT
26DE 9645            LDA A  AZBCD
26E0 9945            ADC A  AZBCD
26E2 19              DAA
26E3 9753            STA A  AZKEY    ;ENTER IN THIRD POINT
26E5 9653            LDA A  AZKEY    ;BEGIN RASTER LIMIT TEST
26E7 9654            LDA A  AZKEY+1  ;GET NEXT POINT
26E9 CE0076          LDX    #FPTAZ    ;GET END POINT
26EC BD2AC4          JSR    BCDSUB   ;NEXT POINT MINUS END POINT
26EF 7D0098          TST    CARRY    ;CHECK FOR WHICH IS DOMINANT
26F2 261D            BNE    ST27E    ;BRANCHES IF SCAN IS FINISHED
26F4 7C007A   ST27B2 INC    PROCNT   ;INCREMENT PROGRAM STATE COUNTER
26F7 967A            LDA A  PROCNT
26F9 8104            CMP A  #04H     ;IS THIS CALCULATING 5TH POINT
26FB 2611            BNE    ST27D    ;BRANCHES IF THIS IS TRUE
26FD 7F007A          CLR    PROCNT   ;IS CALCULATING 3RD POINT
2700 7E20B7   ST27B3 JMP    ST0      ;GO TO CONTROL LOOP, EXPECT A RETURN
2703 B631FB   ST27C  LDA A  PLUS     ;CALCULATES 4TH POINT
2706 9758            STA A  ELKEYS
2708 7C007A          INC    PROCNT   ;INCREMENTS PROGRAM STATE COUNTER
270B 7E20B7          JMP    ST0      ;GO TO CONTROL LOOP, EXPECT A RETURN
270E 7E2641   ST27D  JMP    ST26B3
2711 7F007D   ST27E  CLR    PFLAG    ;CLEAR PROGRAM FLAG
2714 7E20B7          JMP    ST0      ;GO TO CONTROL LOOP, DONT COME BACK
              ;
              ;      BEGIN STATE TWENTY-EIGHT, PATTERN NUMBER THREE
              ;      ENTER ANGLE LESS THAN 100 DEGREES TO SEPARATE SCANS
              ;      PROGRAM WILL GENERATE 360 DEGREES OF RASTER SCANS
              ;      (MODIFICATION 1.1)
              ;
2717 7F007F   ST28   CLR    PROANG   ;INITIALIZE ANGLE COUNTER
271A 7F0059          CLR    MFLAG
271D 7F0085          CLR    COSINE   ;INITIALIZE TWO-BYTE VALUES
2720 7F0083          CLR    SINE
2723 9665            LDA A  PROGA    ;GET ANGLE
2725 D666            LDA B  PROGA+1
2727 54              LSR B           ;DIVIDE BY 100
2728 46              ROR A
2729 54              LSR B
272A 46              ROR A
272B 54              LSR B
272C 46              ROR A
272D 54              LSR B
272E 46              ROR A
272F BD29EB          JSR    BCDBIN   ;CONVERT THIS BCD VALUE TO BINARY
2732 9781            STA A  BINANG   ;SAVE TWO-BYTE RESULT
2734 D782            STA B  BINANG+1
```

```
2736 D67D            LDA B  PFLAG      ;SET PROGRAM FLAG
2738 CE0000   ST28A  LDX    #0000H     ;CALCULATE POINT NUMBER ONE
273B DF53            STX    AZKEY      ;SEND POSITIONER TO ORIGIN
273D DF55            STX    ELKEY
273F C62B            LDA B  #2BH       ;FIX SIGNS FOR PROPER QUADRANT
2741 D757            STA B  AZKEYS
2743 D758            STA B  ELKEYS
2745 CE274D          LDX    #ST28B
2748 DF7B            STX    STADDR     ;SAVES RETURN ADDRESS
274A 7E20B7          JMP    STO        ;GO TO CONTROL LOOP, EXPECT A RETURN
274D DE7F     ST28B  LDX    PROANG     ;CALCULATE POINT NUMBER TWO
274F 8C0168          CPX    #0168H     ;COMPARE TO 360 DEGREES
2752 2847            BVC    ST28D      ;BRANCHES IF PATTERN COMPLETED
2754 BD2A2A          JSR    TRIGAD     ;PATTERN NOT FINISHED
2757 DE83            LDX    SINE       ;GET SINE VALUE
2759 8640            LDA A  #40H       ;GET AZMAX
275B 5F              CLR B
275C BD296D          JSR    BCDMPY     ;SINE (ANGLE) TIMES AZMAX
275F 9753            STA A  AZKEY      ;SET AZIMUTH DESTINATION
2761 D754            STA B  AZKEY+1
2763 DE85            LDX    COSINE     ;GET COSINE VALUE
2765 8640            LDA A  #40H
2767 5F              CLR B
2768 BD296D          JSR    BCDMPY     ;COSINE (ANGLE) TIMES ELMAX
276B 9755            STA A  ELKEY      ;SET ELEVATION DESTINATION
276D D756            STA B  ELKEY+1
276F D687            LDA B  SSIGN      ;GET SIGN OF SINE VALUE
2771 D757            STA B  AZKEYS
2773 D688            LDA B  CSIGN      ;GET SIGN OF COSINE VALUE
2775 D758            STA B  ELKEYS
2777 CE277F          LDX    #ST28C
277A DF7B            STX    STADDR     ;STORE RETURN ADDRESS
277C 7E20B7          JMP    STO        ;GO TO CONTROL LOOP, EXPECT A RETURN
277F D657     ST28C  LDA B  AZKEYS     ;CALCULATE POINT NUMBER THREE
2781 53              COM B             ;FIX AZIMUTH SIGN
2782 D757            STA B  AZKEYS
2784 D658            LDA B  ELKEYS
2786 53              COM B             ;FIX ELEVATION SIGN
2787 D758            STA B  ELKEYS
2789 967F            LDA A  PROANG
278B D680            LDA B  PROANG+1
278D DB82            ADD B  BINANG+1
278F D780            STA B  PROANG+1;UPDATE PROGRAMMED ANGLE
2791 977F            STA A  PROANG
2793 CE2738          LDX    #ST28A
2796 DF7B            STX    STADDR     ;SAVE RETURN ADDRESS
2798 7E20B7          JMP    STO        ;GO TO CONTROL LOOP, EXPECT A RETURN
279B 7F007D   ST28D  CLR    PFLAG      ;SCAN FINISHED, CLEAR PROGRAM FLAG
279E 7E20B7          JMP    STO        ;GO TO CONTROL LOOP, DONT RETURN
              ;
              ;      BEGIN STATE TWENTY-NINE, PATTER NUMBER FOUR
              ;      ENTER RADIUS (IN DEGRESS) OF CIRCLE
              ;      PROGRAM WILL GENERATE ONE CIRCLE WITH RADIUS " A "
```

```
                        ;       (MODIFICATION 1.1)
                        ;
27A1 7F007F     ST29    CLR     PROANG  ;INITIALIZE ANGLE COUNTER
27A4 7F0059             CLR     NFLAG
27A7 7F0085             CLR     COSINE  ;INITIALIZE TWO-BYTE VALUE
27AA 7F0083             CLR     SINE
27AD C6FF               LDA B   #0FFH
27AF D77D               STA B   PFLAG   ;SET PROGRAM FLAG
27B1 DE7F       ST29A   LDX     PROANG  ;CALCULATE POINT ON CIRCLE
27B3 8C0168             CPX     #360    ;COMPARES ANGLE WITH 360 DEGREES
27B6 2838               BVC     ST29B   ;BRANCHES IF PROGRAM IS COMPLETED
27B8 BD2A2A             JSR     TRIGAD  ;FIND SINE AND COSINE OF ANGLE
27BB DE83               LDX     SINE
27BD 9665               LDA A   PROGA
27BF D666               LDA B   PROGA+1
27C1 BD296D             JSR     BCDMPY  ;SINE (ANGLE) TIMES ENTERED RADIUS
27C4 9755               STA A   AZKEY   ;SET AZIMUTH DESTINATION
27C6 D754               STA B   AZKEY+1
27C8 DE85               LDX     COSINE
27CA 9665               LDA A   PROGA
27CC D666               LDA B   PROGA+1
27CE BD296D             JSR     BCDMPY  ;COSINE (ANGLE) TIMES ENTERED RADIUS
27D1 D754               STA B   AZKEY+1
27D3 DE85               LDX     COSINE
27D5 9665               LDA A   PROGA
27D7 D666               LDA B   PROGA+1
27D9 BD296D             JSR     BCDMPY  ;COSINE (ANGLE) TIMES ENTERED RADIUS
27DC 9755               STA A   ELKEY   ;SET ELEVATION DESTINATION
27DE D756               STA B   ELKEY+1
27E0 D587               LDA B   SSIGN   ;GET SIGN OF SINE VALUE
27E2 D757               STA B   AZKEYS
27E4 D689               LDA B   CSIGN   ;GET SIGN OF COSINE VALUE
27E6 D758               STA B   ELKEYS
27E8 CE27B1             LDX     #ST29A
27EB DF7B               STX     STADDR  ;STORE RETURN ADDRESS
27ED 7E20B7             JMP     STO     ;GO TO CONTROL LOOP. EXPECT A RETURN
27F0 7F007D     ST29B   CLR     PFLAG   ;CIRCLE COMPLETED, CLEAR PROGRAM FLAG
27F3 7E20B7             JMP     STO     ;GO TO CONTROL LOOP, DONT COME BACK
                        ;
                        ;       BEGIN STATE THIRTY, GET NEGATIVE ELEVATION LIMIT
                        ;       (MODIFICATION 1.1)
                        ;
27F6 CE3188     ST30    LDX     #MSG16
27F9 BD2C22             JSR     ASCDIS  ;DISPLAYS "NEG EL LIMIT        "
27FC CE0010             LDX     #DISEL+6
27FF DF32               STX     SAVEX   ;SAVE DISPLAY TRACKING POINTER
2801 968D               LDA A   NELLIM  ;GET CURRENT NEGATIVE ELEVATION LIMIT
2803 D68E               LDA B   NELLIM+1
2805 BD2D2F             JSR     BCDDIS  ;DISPLAY CURRENT NEGATIVE ELEVATION LIMIT
2808 BD2943             JSR     MOVEP   ;FIX LAST ENTRY IN DECIMAL PT
280B CE2823             LDX     #ST30B
280E DF7B               STX     STADDR  ;STORES RETURN ADDRESS
2810 966B       ST30A   LDA A   KFLAG
```

```
2812 2AFC           BPL    ST30A    ;WAITS FOR KEYENTRY
2814 7F006B         CLR    KFLAG    ;A KEY IS PRESSED; CLEAR KEYENTRY FLAG
2817 964F           LDA A  KEYENT   ;GET KEYENTRY
2819 CE3457         LDX    #SP30    ;LOADS INDEX REGISTER WITH STATE 30 POINTER
281C BD2BE8         JSR    ADDCAL   ;SUBROUTINE CALCULATES NEXT ADDRESS
281F EE00           LDX    0,X
2821 6E00           JMP    0,X      ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
2823 DE4D    ST30B  LDX    ENTRYA   ;JUST RETURNED FROM ENTERING NELLIM
2825 DF8D           STX    NELLIM   ;UPDATE NEGATIVE ELEVATION LIMIT
2827 20E7           BRA    ST30A    ;BRANCH TO WAIT ON NEXT KEYENTRY
             ;
             ;      BEGIN STATE THIRTY-ONE, SET POSITIVE ELEVATION LIMITS
             ;      (MODIFICATION 1.1)
             ;
2829 CE319C  ST31   LDX    #MSG17
282C BD2C22         JSR    ASCDIS   ;DISPLAYS "POS EL LIMIT          "
282F CE0010         LDX    #DISEL+6
2832 DF32           STX    SAVEX
2834 968B           LDA A  PELLIM   ;GET POSITIVE ELEVATION LIMIT
2836 D68C           LDA B  PELLIM+1
2838 BD2D2F         JSR    BCDDIS   ;DISPLAYS CURRENT POSITIVE ELEVATION
283B BD2943         JSR    MOVED
283E CE2856         LDX    #ST31B
2841 DF7B           STX    STADDR   ;SAVES RETURN ADDRESS
2843 966B    ST31A  LDA A  KFLAG
2845 2AFC           BPL    ST31A    ;WAITS FOR KEYENTRY
2847 7F006B         CLR    KFLAG    ;A KEY IS PRESSED, CLEAR KEYENTRY FLAG
284A 964F           LDA A  KEYENT   ;GET KEYENTRY
284C CE347F         LDX    #SP31    ;LOADS INDEX REGISTER WITH STATE 31 POINTER
284F BD2BE8         JSR    ADDCAL   ;SUBROUTINE CALCULATES NEXT ADDRESS
2852 EE00           LDX    0,X
2854 6E00           JMP    0,X      ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
2856 DE4D    ST31B  LDX    ENTRYA   ;JUST RETURNED FROM ENTERING PELLIM
2858 DF8B           STX    PELLIM   ;UPDATE POSITIVE ELEVATION LIMIT
285A 20E7           BRA    ST31A    ;BRANCHES TO WAIT ON NEXT KEYENTRY
             ;
             ;      BEGIN STATE THIRTY-TWO, SET NEGATIVE AZIMUTH LIMIT
             ;      (MODIFICATION 1.1)
             ;
285C CE31B0  ST32   LDX    #MSG18
285F BD2C22         JSR    ASCDIS   ;DISPLAYS "NEG AZ LIMIT          "
2862 CE0010         LDX    #DISEL+6
2865 DF32           STX    SAVEX    ;SAVES DISPLAY TRACKING POINTER
2867 9691           LDA A  NAZLIM   ;GET POSITIVE ELEVATION LIMIT
2869 D692           LDA B  NAZLIM+1
286B BD2D2F         JSR    BCDDIS   ;DISPLAY CURRENT NEGATIVE AZIMUTH LIMIT
286E BD2943         JSR    MOVED
2871 CE2889         LDX    #ST32B
2874 DF7B           STX    STADDR   ;SAVES RETURN ADDRESS
2876 966B    ST32A  LDA A  KFLAG
2878 2AFC           BPL    ST32A    ;WAITS FOR KEYENTRY
287A 7F006B         CLR    KFLAG    ;A KEY IS PRESSED, CLEAR KEYENTRY
287D 964F           LDA A  KEYENT   ;GET KEYENTRY
```

```
287F CE34A7          LDX   #SP32    ;LOADS INDEX REGISTER WITH STATE 32 POINTER
2882 BD2BE8          JSR   ADDCAL   ;SUBROUTINE CALCULATES NEXT ADDRESS
2885 EE00            LDX   0,X
2887 6E00            JMP   0,X      ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
2889 DE4D     ST32B  LDX   ENTRYA   ;JUST RETURNED FROM ENTERING NAZLIM
288B DF91            STX   NAZLIM   ;UPDATE NEGATIVE AZIMUTH LIMIT
288D 20E7            BRA   ST32A    ;BRANCH TO WAIT ON NEXT KEYENTRY
              ;
              ;      BEGIN STATE THIRTY-THREE, SET POSITIVE AZIMUTH LIMIT
              ;      (MODIFICATION 1.1)
              ;
288F CE31C4   ST33   LDX   #MSG19
2892 BD2C22          JSR   ASCDIS   ;DISPLAYS "POS AZ LIMIT        "
2895 CE0010          LDX   #DISEL+6
2898 DF32            STX   SAVEX    ;SAVES DISPLAY TRACKING POINTER
289A 968F            LDA A PAZLIM   ;GET POSITIVE ELEVATION LIMIT
289C D690            LDA B PAZLIM+1
289E BD2D2F          JSR   BCDDIS   ;DISPLAY CURRENT POSITIVE AZIMUTH LIMIT
28A1 BD2943          JSR   MOVED
28A4 CE28BC          LDX   #ST33B
28A7 DF7B            STX   STADDR   ;SAVES RETURN ADDRESS
28A9 966B     ST33A  LDA A KFLAG
28AB 2AFC            BPL   ST33A    ;WAITS FOR KEYENTRY
28AD 7F006B          CLR   KFLAG    ;A KEY IS PRESSED, CLEAR KEYENTRY
28B0 964F            LDA A KEYENT   ;GET KEYENTRY
28B2 CE34CF          LDX   #SP33    ;LOADS INDEX REGISTER
28B5 BD2BE8          JSR   ADDCAL   ;SUBROUTINE CALCULATES NEXT ADDRESS
28B8 EE00            LDX   0,X
28BA 6E00            JMP   0,X      ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
28BC DE4D     ST33B  LDX   ENTRYA   ;JUST RETURNED FROM ENTERING PAZLIM
28BE DF8F            STX   PAZLIM   ;UPDATE POSITIVE AZIMUTH LIMIT
28C0 20E7            BRA   ST33A    ;BRANCH TO WAIT ON NEXT KEYENTRY
              ;
              ;      BEGIN STATE THIRTY-FOUR, INPUT NUMBERS FOR SETTING LIMITS--1ST
              ;      (MODIFICATION 1.1)
              ;
28C2 44       ST34   LSR A          ;BEGIN INPUT, CONVERT KEYCODE TO BCD CODE
28C3 16              TAB            ;SAVES THIS BCD INPUT
28C4 BD2A9E          JSR   PACK     ;PACKS BCD INPUT INTO PACKED BCD FORM
28C7 CB30            ADD B #30H     ;CONVERTS BCD INPUT INTO ASCII CODE
28C9 DE32            LDX   SAVEX    ;GETS DISPLAY TRACKING POINTER
28CB E700            STA B 0,X
28CD 08              INX            ;INCREMENTS TRACKING POINTER
28CE DF32            STX   SAVEX    ;SAVES INCREMENTED POINTER
28D0 966B     ST34A  LDA A KFLAG
28D2 2AFC            BPL   ST34A    ;WAITS FOR NEXT BCD INPUT FROM KEYBOARD
28D4 7F006B          CLR   KFLAG    ;KEY IS PRESSED CLEAR KEYENTRY FLAG
28D7 964F            LDA A KEYENT   ;GET KEYENTRY
28D9 CE34F7          LDX   #SP34    ;LOAD INDEX REGISTER WITH STATE 34 POINTER
28DC BD2BE8          JSR   ADDCAL   ;SUBROUTINE CALCULATES NEXT ADDRESS
28DF EE00            LDX   0,X
28E1 6E00            JMP   0,X      ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
              ;
```

```
                     ;      BEGIN STATE THIRTY-FIVE, INPUT BCD CHARACTER NUMBER TWO
                     ;      (MODIFICATION 1.1)
                     ;
28E3 44         ST35   LSR A           ;2ND INPUT, CONVER KEYCODE TO BCD CODE
28E4 16                TAB             ;SAVE THIS BCD INPUT
28E5 BD2A9E            JSR   PACK      ;PACKS BCD INPUT INTO PACKED BCD FORM
28E8 CB30              ADD B #30H
28EA DE32              LDX   SAVEX     ;GETS DISPLAY TRACKING POINTER
28EC E700              STA B 0,X       ;ECHOS KEYENTRY
28EE 08                INX             ;INCREMENTS TRACKING POINTER
28EF DF32              STX   SAVEX     ;SAVES INCREMENTED POINTER
28F1 966B       ST35A  LDA A KFLAG
28F3 2AFC              BPL   ST35A     ;WAITS FOR NEXT BCD INPUT FROM KEYBOARD
28F5 7F006B            CLR   KFLAG     ;A KEY IS PRESSED, CLEAR KEYENTRY FLAG
28F8 964F              LDA A KEYENT    ;GET KEYENTRY
28FA CE351F            LDX   #SP35     ;LOAD INDEX REGISTER WITH STATE 35 POINTER
28FD BD2BE8            JSR   ADDCAL    ;SUBROUTINE CALCULATES NEXT ADDRESS
2900 EE00              LDX   0,X
2902 6E00              JMP   0,X       ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
                     ;
                     ;      BEGIN STATE THIRTY-SIX, INPUT DECIMAL POINT
                     ;      (MODIFICATION 1.1)
                     ;
2904 C62E       ST36   LDA B #2EH      ;3RD INPUT, CONVERT INPUT TO BCD CODE
2906 DE32              LDX   SAVEX     ;GETS TRACKING POINTER
2908 E700              STA B 0,X       ;ECHOS KEYENTRY
290A 08                INX             ;INCREMENTS TRACKING POINTER
290B DF32              STX   SAVEX
290D 966B       ST36A  LDA A KFLAG
290F 2AFC              BPL   ST36A     ;WAITS FOR LAST BCD CHARACTER
2911 7F006B            CLR   KFLAG     ;KEY IS PRESSED, CLEAR KEYENTRY FLAG
2914 964F              LDA A KEYENT    ;GET KEYENTRY
2916 CE3547            LDX   #SP36     ;LOAD INDEX REGISTER WITH STATE 36 POINTER
2919 BD2BE8            JSR   ADDCAL    ;SUBROUTINE CALCULATES NEXT ADDRESS
291C EE00              LDX   0,X
291E 6E00              JMP   0,X       ;JUMPS TO CALCULATED ADDRESS OF NEXT STATE
                     ;
                     ;      BEGIN STATE THIRTY-SEVEN, INPUT LAST BCD CHARACTER
                     ;      (MODIFICATION 1.1)
                     ;
2920 44         ST37   LSR A           ;LAST INPUT, CONVERT INPUT TO BCD CODE
2921 16                TAB             ;SAVES THIS BCD INPUT
2922 BD2A9E            JSR   PACK      ;PACKS BCD INPUT INTO ASCII CODE
2925 CB30              ADD B #30H
2927 DE32              LDX   SAVEX
2929 E700              STA B 0,X       ;ECHOS KEYENTRY
292B 964D       ST37A  LDA A ENTRYA    ;GET BCD ANGLE
292D D64E              LDA B ENTRYB
292F BD295E            JSR   TSTANG    ;TEST ANGLE FOR LIMTI CONDITION
2932 DE7B              LDX   STADDR    ;GET RETURN ADDRESS
2934 6E00              JMP   0,X       ;GO BACK TO THAT STATE
                     ;
                     ;      "RESTO" SUBROUTINE
```

```
                        ;       THIS SUBROUTINE RETURNS THE KEYBOARD TO AN INITIALIZED STATE
                        ;       SO THAT ANY KEY PRESS WILL GENERATE AN INTERRUPT.
                        ;
2936 860F        RESTO  LDA A   #00FH
2933 B78400             STA A   DDRA     ;RESTORE THE ROWS OF KEYBD FOR NEXT KEY PUSHED
293B B68400             LDA A   DDRA     ;CLEARS IRQ BITS IN CONTROL REGISTER A
293E 86FF               LDA A   #0FFH
2940 9759               STA A   CFLAG    ;DISABLES THE CONTROL LOOP
2942 39                 RTS
2943 9612        MOVED  LDA A   DISEL+8
2945 C62E               LDA B   #2EH
2947 D712               STA B   DISEL+8
2949 9713               STA A   DISEL+9
294B 39                 RTS
                        ;
                        ;       "CPFLAG" ROUTINE
                        ;       (MODIFICATION 1.1)
                        ;       THIS ROUTINE IS REACHED ONLY AFTER POSITIONER HAS STOPPED
                        ;       CHECKS PROGRAM FLAG (PFLAG) TO SEE IF IT IS CURRENTLY IN
                        ;       A PROGRAMMED SEQUENCE.
                        ;
294C 966E        CPFLAG LDA A   SFLAGA
294E 81FF               CMP A   #0FFH
2950 2609               BNE     CP1
2952 7D007D             TST     PFLAG    ;LOOKS AT PROGRAM FLAG
2955 2704               BEQ     CP1      ;BRANCHES TO CONTROL LOOP IF NOT IN A PROGRAMMED SEQUENCE
2957 DE7B               LDX     STADDR   ;YES, A PROGRAMMED SEQUENCE IS CURRENTLY IN OPERATION
2959 6E00               JMP     0,X      ;JUMPS BACK TO THE PROGRAMMED CONTROL ROUTINE.
295B 7E20B7      CP1    JMP     ST0      ;PROGRAM FLAG CLEARED, GO TO CONTROL LOOP
                        ;
                        ;       "TSTANG" SUBROUTINE
                        ;       COMPARES THE TWO ACCXS WITH THE CONTENTS OF THE
                        ;       INDEX REGISTER.  RETURNS FROM SUB. IF CONTENTS
                        ;       OF INDEX REGISTER ARE LARGER THAN THE CONTENTS
                        ;       OF THE ACCXS.  BRANCHES TO ERROR 6 IF NOT.
                        ;
295E CE31F6      TSTANG LDX     #LIMIT
2961 BD2AC4             JSR     BCDSUB
2964 7D0098             TST     CARRY
2967 2B03               BMI     NOPE
2969 7E2290             JMP     ST6      ;BRANCHES IF ENTERED ANGLE EXCEEDS LIMIT OF +/-40.0
296C 39          NOPE   RTS
                        ;
                        ;       "BCDMPY" SUBROUTINE
                        ;       REVISION 2.1
                        ;       MULTIPLIES TWO PACKED BCD (16-BIT) VALUES
                        ;       RE-ENTRANT CODE (USES 7 BYTES ON STACK)
                        ;       ACCA, ACCB TIMES (X), (X+1)
                        ;       RESULT IN ACCA, ACCB
                        ;       (MODIFICATION 1.1)
                        ;
296D 37          BCDMPY PSH B            ;PUSH MULTIPLIER ONTO STACK
296E 36                 PSH A
```

```
2 96F A601          LDA A  1,X
2971 36             PSH A          ;PUSH MULTIPLICAND ONTO STACK
2972 A600           LDA A  0,X
2974 36             PSH A
2975 8610           LDA A  #16
2977 36             PSH A          ;PUSH BIT COUNTER ONTO STACK
2978 30             TSX            ;TRANSFERS STACK ADDRESS TO INDEX REGISTER
               ;    STACK NOW LOOKS LIKE
               ;    +0 COUNT
               ;    +1 MS BYTE OF MULTIPLICAND
               ;    +2 LS BYTE
               ;    +3 MS BYTE OF MULTIPLIER
               ;    +4 LS BYTE
               ;    +5 MS BYTE OF RETURN ADDRESS
               ;    +6 LS BYTE
2979 A603           LDA A  3,X
297B 58      MPY163 ASL B
297C 49             ROL A          ;FORM RESULT
297D 6802           ASL    2,X     ;SHIFT MULTIPLICAND
297F 6901           ROL    1,X
2981 2400           BCC    MPY167  ;BRANCHES IF NO RESTORE IS NEEDED
2983 9736           STA A  TEMPA   ;SAVE ACCA
2985 17             TBA            ;NEED TO DO BCD ADD IN ACCA
2986 AB04           ADD A  4,X
2988 19             DAA            ;DECIMAL ADJUST
2989 16             TAB            ;TRANSFER BACK TO ACCB
298A 9636           LDA A  TEMPA   ;GET ACCA THAT WAS SAVED
298C A903           ADC A  3,X     ;FINISH BCD MATH
298E 19             DAA
298F 6A00    MPY167 DEC    0,X
2991 26E8           BNE    MPY163  ;BRANCHES IF COUNT NOT ZERO
               ;    CLEAN UP STACK
2993 31             INS
2994 31             INS
2995 31             INS
2996 31             INS
2997 31             INS
2998 39             RTS
               ;
               ;    "BCDDIV" SUBROUTINE
               ;    REVISION 1.2
               ;    BCD (16-BIT) VALUE INTEGER DIVIDE ROUTINE
               ;    LOAD ACCA. ACCB WITH BCD DIVIDEND AND LOAD
               ;    INDEX REGISTER WITH BCD DIVISOR
               ;    RETURNS WITH QUOTIENT IN ACCA, ACCB.
               ;    (MODIFICATION 1.1)
               ;
2999 37      BCDDIV PSH B          ;FORMS DIVIDEND IN STACK
299A 36             PSH A
299B A600           LDA A  0,X
299D E601           LDA B  1,X
299F 37             PSH B
29A0 36             PSH A          ;FORMS DIVISOR IN STACK
```

```
29A1 34                  DES
29A2 30                  TSX                ;POINTER IN INDEX REGISTER TO STACKED DATA
29A3 8601                LDA A  #1
29A5 6D01                TST    1,X
29A7 2B0B                BMI    BCD153
29A9 4C          BCD151  INC A              ;COUNT NUMBER OF TIMES DIVISOR IS SHIFTED LEFT
29AA 6802                ASL    2,X
29AC 6901                ROL    1,X         ;SHIFTS DIVISOR LEFT ONE BIT
29AE 2B04                BMI    BCD153      ;BRANCHES IF DIVISOR HAS BEEN SHIFTED LEFT 16 TIMES
29B0 8111                CMP A  #17         ;CHECK FOR DIVISOR EQUAL TO ZERO
29B2 26F5                BNE    BCD151      ;BRANCHES IF DIVISOR NOT LEFT JUSTIFIED
29B4 A700        BCD153  STA A  0,X         ;SAVES SHIFT COUNT ON STACK
29B6 A603                LDA A  3,X         ;GET DIVIDEND FROM STACK
29B8 E604                LDA B  4,X
29BA 6F03                CLR    3,X         ;INITIALIZE DIVIDEND TO BECOME QUOTIENT
29BC 6F04                CLR    4,X
                 ;       STACK LOOKS LIKE@D
                 ;       +0 COUNT
                 ;       +1 MS BYTE OF DIVISOR
                 ;       +2 LS BYTE
                 ;       +3 MS BYTE OF DIVIDEND
                 ;       +4 LS BYTE
                 ;       +5 MS BYTE OF RETURN ADDRESS
                 ;       +6 LS BYTE
29BE DF32        BCD163  STX    SAVEX       ;SAVE POINTER TO STACK
29C0 EE01                LDX    1,X         ;GET DIVISOR
29C2 BD2AC4              JSR    BCDSUB      ;DIVIDEND MINUS DIVISOR
29C5 7D0098              TST    CARRY       ;CARRY = 0, SUBTRACTION OVERFLOW
29C8 270E                BEQ    BCD165      ;BRANCHES IF DIVISOR STILL OKAY
29CA 9736                STA A  TEMPA       ;BEGIN RESTORE, SAVE VALUE IN ACCA
29CC 17                  TBA                ;BEGIN BCD MATH
29CD AB02                ADD A  2,X
29CF 19                  DAA
29D0 16                  TAB                ;PUT BCD VALUE BACK IN ACCB
29D1 9636                LDA A  TEMPA       ;GET ACCA THAT WAS SAVED
29D3 A901                ADC A  1,X         ;FINISH BCD MATH
29D5 0C                  CLC                ;CLEAR CARRY BIT, SHIFT IN ZERO
29D6 2001                BRA    BCD167      ;BRANCHES TO SHIFT IN ZERO TO QUOTIENT
29D8 0D          BCD165  SEC                ;SET CARRY BIT, SHIFT IN ONE (1)
29D9 6904        BCD167  ROL    4,X         ;SHIFTS IN CARRY BIT INTO LSB OF QUOTIENT
29DB 6903                ROL    3,X
29DD 6401                LSR    1,X         ;SHIFTS DIVISOR RIGHT WITH ZERO FILL
29DF 6602                ROR    2,X
29E1 6A01                DEC    1,X         ;DECREMENTS COUNT
29E3 26D9                BNE    BCD163      ;BRANCHES IF COUNT EQUAL TO ZERO (0)
                 ;       CLEAN UP STACK
29E5 31                  INS
29E6 31                  INS
29E7 31                  INS
29E8 32                  PUL A              ;PULLS CORRECTED BCD VALUE FROM STACK
29E9 33                  PUL B              ;DESTROYS REMAINDER
29EA 39                  RTS
                 ;
```

```
                    ;       "BCDBIN" SUBROUTINE
                    ;       CONVERTS FOUR BINARY CODED DECIMAL DIGITS
                    ;       TO A BINARY EQUIVALENT.  THE BCD DIGITS ARE
                    ;       PACKED TWO PER BYTE.  THE BINARY RESULT
                    ;       OCCUPIES TWO BYTES.  THE BCD DIGITS ARE LOADED
                    ;       INTO THE ACCA AND ACCB (MSD TO ACCA) AND THE
                    ;       BCDBIN SUBROUTINE IS CALLED.  THE ROUTINE EXITS
                    ;       WITH THE BINARY RESULT IN ACCA AND ACCB.
                    ;       (MODIFICATION 1.1)
                    ;
29EB 9789           BCDBIN STA A  SAVE1    ;SAVE 2 MS BCD VALUES
29ED 7F008A                CLR    BINUPR
29F0 17                    TBA
29F1 C40F                  AND B  #0FH     ;SAVE ONLY LS BCD VALUE
29F3 44                    LSR A           ;MOVE TENS BCD VALUE OF ACCA
29F4 44                    LSR A
29F5 44                    LSR A
29F6 44                    LSR A
29F7 2705           TENLP  BEQ    DOHUND   ;GO DO HUN WHEN TEN IS ZERO
29F9 CB0A                  ADD B  #10      ;ADD TEN TO BINARY TOTAL
29FB 4A                    DEC A           ;DECREMENT TENS DIGIT AND
29FC 20F9                  BRA    TENLP    ;REPEAT UNTIL ZERO
29FE 0C             DOHUND CLC             ;RESET CARRY
29FF 9689                  LDA A  SAVE1    ;GET HUN AND THOU DIGIT
2A01 840F                  AND A  #0FH     ;SAVE ONLY HUN DIGIT
2A03 270A           HUNLP  BEQ    DOTHOU   ;GO DO THOU IF HUN IS ZERO
2A05 CB64                  ADD B  #100     ;ADD 100 TO BINARY VALUE
2A07 2403                  BCC    HUN00
2A09 7C008A                INC    BINUPR   ;ADD 256 TO BIN UPPER VALUE
2A0C 4A             HUN00  DEC A           ;DECREMENT HUN DIGIT OND
2A0D 20F4                  BRA    HUNLP    ;REPEAT UNTIL ZERO
2A0F 9689           DOTHOU LDA A  SAVE1    ;GET THOU DIGIT
2A11 44                    LSR A           ;MOVE THOU BCD VALUE TO
2A12 44                    LSR A           ;LOWER FOUR BITS OF ACCA.
2A13 44                    LSR A
2A14 44                    LSR A
2A15 9789                  STA A  SAVE1    ;SAVE THOU DIGIT
2A17 2604                  BNE    THOU00   ;BRANCH IF THOU DIGIT IS ZERO
2A19 968A                  LDA A  BINUPR   ;GET BINARY UPPER VALUE
2A1B 200C                  BRA    XITBIN
2A1D 968A           THOU00 LDA A  BINUPR   ;GET BINARY UPPER VALUE
2A1F 0C             THOULP CLC             ;RESET CARRY
2A20 CBE8                  ADD B  #232     ;ADD 232 TO BINARY LOWER
2A22 8903                  ADC A  #03H     ;ADD 768 TO BINARY UPPER
2A24 7A0089                DEC    SAVE1    ;DECREMENT THOU DIGIT
2A27 26F6                  BNE    THOULP   ;REPEAT UNTIL THOU DIGIT ZERO
2A29 39             XITBIN RTS
                    ;
                    ;       "TRIGAD" SUBROUTINE
                    ;       SCALES ANGLE TO BETWEEN 0 AND 90 DEGREES
                    ;       FINDS SINE AND COSINE OF ANGLE
                    ;       RETURNS WITH RESULT IN ACCA (SINE) AND ACCB (COSINE)
                    ;       (MODIFICATION 1.1)
```

```
                    ;
2A2A DF36     TRIGAD STX   TEMPA    ;SAVES UPDATED ANGLE
2A2C 8C010E          CPX   #010EH   ;COMPARES ANGLE TO 270 DEGREES
2A2F 2916            BVS   TRIGA    ;BRANCHES IF ANGLE LESS THAN 270 DEGREES
2A31 8601            LDA A #01H     ;ANGLE IN 2ND QUADRANT
2A33 C668            LDA B #68H     ;GET 360 DEGREES
2A35 D037            SUB B TEMPB    ;360DEGREES MINUS (270 TO 359 DEGREES)
2A37 9236            SBC A TEMPA    ;DIFFERENCE BETWEEN 0 AND 90 DEGREES
2A39 D737            STA B TEMPB    ;SAVE DIFFERENCE
2A3B 9736            STA A TEMPA
2A3D C62B            LDA B #2BH     ;FIX PROPER SIGNS FOR CURRENT QUADRANT
2A3F D787            STA B SSIGN
2A41 C62D            LDA B #2DH
2A43 D788            STA B CSIGN
2A45 203A            BRA   TRIGD    ;GO TO FIND SINE AND COSINE
2A47 8C00B4   TRIGA  CPX   #00B4H   ;COMPARES ANGLE TO 180 DEGREES
2A4A 2914            BVS   TRIGB    ;BRANCHES IF ANGLE LESS THAN 180 DEGREES
2A4C 8601            LDA A #01H     ;ANGLE IN 3RD QUADRANT
2A4E C60E            LDA B #0EH     ;GET 270 DEGREES
2A50 D037            SUB B TEMPB    ;270 DEGREES MINUS (180 TO 269 DEGREES)
2A52 9236            SBC A TEMPA    ;DIFFERENCE BETWEEN 0 AND 90 DEGREES
2A54 D737            STA B TEMPB    ;SAVE DIFFERENCE
2A56 9736            STA A TEMPA
2A58 C62D            LDA B #2DH     ;FIX PROPER SIGNS FOR CURRENT QUADRANT
2A5A D787            STA B SSIGN
2A5C D788            STA B CSIGN
2A5E 2021            BRA   TRIGD    ;GO TO FIND SINE AND COSINE
2A60 8C005A   TRIGB  CPX   #005AH   ;COMPARES ANGLE TO 90 DEGREES
2A63 2916            BVS   TRIGC    ;BRANCHES IF ANGLE LESS THAN 90 DEGREES
2A65 8600            LDA A #00H     ;ANGLE IN 4TH QUADRANT
2A67 C6B4            LDA B #0B4H    ;GET 180 DEGREES
2A69 D037            SUB B TEMPB    ;180 DEGREES MINUS (90 TO 179 DEGREES)
2A6B 9236            SBC A TEMPA    ;DIFFERENCE BETWEEN 0 AND 90 DEGREES
2A6D D736            STA B TEMPA    ;SAVE DIFFERENCE
2A6F 9736            STA A TEMPA
2A71 C62B            LDA B #2BH     ;FIX PROPER SIGNS FOR CURRENT QUADRANT
2A73 D788            STA B CSIGN
2A75 C62D            LDA B #2DH
2A77 D787            STA B SSIGN
2A79 2006            BRA   TRIGD    ;GO TO FIND SINE AND COSINE
2A7B C62B     TRIGC  LDA B #2BH     ;ANGLE ALREADY BETWEEN 0 AND 90 DEGRREES
2A7D D787            STA B SSIGN    ;FIX PROPER SIGNS FOR CURRENT QUADRANT
2A7F D788            STA B CSIGN
2A81 9636     TRIGD  LDA A TEMPA    ;GET PROCESSED ANGLE
2A83 D637            LDA B TEMPB
2A85 DB37            ADD B TEMPB    ;DOUBLE ANGLE
2A87 9936            ADC A TEMPA
2A89 FB2FA7          ADD B TRIGTB+1;ADD TO BEGIN ADDRESS OF TABLE
2A8C B92FA6          ADC A TRIGTB
2A8F 9736            STA A TEMPA    ;STORE RESULT
2A91 D737            STA B TEMPB
2A93 DE36            LDX   TEMPA    ;LOAD RESULT INTO INDEX REGISTER
2A95 A600            LDA A 0,X      ;GET SINE
```

```
2A97 E601          LDA B  1,X      ;GET COSINE
2A99 9784          STA A  SINE+1   ;SAVE ONE-BYTE RESULTS
2A9B D786          STA B  COSINE+1
2A9D 39            RTS
                 ;
                 ;       "PACK" SUBROUTINE
                 ;       PACKS BINARY NUMBERS INTO BCD FORM
                 ;       ACCA SHOULD CONTAIN THE UNPACKED BCD FORM
                 ;       ROUTINE DESTROYS CONTENTS OF ACCA.
                 ;
2A9E 9B4E        PACK   ADD A  ENTRYB   ;ENTRYB LOOKS LIKE "X0"
2AA0 974E               STA A  ENTRYB   ;PACKS IN ANOTHER UNPACKED BCD FORM
2AA2 78004E             ASL    ENTRYB   ;THEN DOES 1-BIT LEFT SHIFT WITH ZERO FILL
2AA5 79004D             ROL    ENTRYA
2AA8 78004E             ASL    ENTRYB
2AAB 79004D             ROL    ENTRYA
2AAE 78004E             ASL    ENTRYB
2AB1 79004D             ROL    ENTRYA
2AB4 78004E             ASL    ENTRYB
2AB7 79004D             ROL    ENTRYA   ;SHIFTS 16-BIT BINARY INFORMATION OVER ONE CHARARTER
2ABA 39                 RTS
                 ;
                 ;       "ALSTOP" SUBROUTINE
                 ;       ROUTINE THAT STOPS BOTH MOTORS FOR EXCEEDING ANGLE LIMIT
                 ;
2ABB 86FF        ALSTOP LDA A  #0FFH
2ABD B78E00             STA A  LSBSAZ   ;STOPS AZIMUTH MOTOR
2AC0 B78E02             STA A  LSBSEL   ;STOPS ELEVATION MOTOR
2AC3 39                 RTS
                 ;
                 ;       "BCDSUB" SUBROUTINE
                 ;       SUBTRACTS TWO 16-BIT BCD PACKED NUMBERS
                 ;       SUBTRACTS INDEXED ADDRESS FROM ACCA, ACCB
                 ;       INDEX REG. CONTAINS STARTING ADDRESS
                 ;       ACCA, ACCB CONTAINS NUMBER TO BE SUBTRACTED
                 ;       FROM.  RETURNS WITH RESULT FROM SUBTRACTION IN
                 ;       ACCA, ACCB.
                 ;       ******************
                 ;       *     9999       *
                 ;       *    -IXRG       *
                 ;       *    ------      *
                 ;       *     DIFF       *
                 ;       *    +   1       *
                 ;       *    ------      *
                 ;       *    DIFF+1      *
                 ;       *     +BCD?      *
                 ;       *    ------      *
                 ;       *    ANSWER      *
                 ;       ******************
                 ;       BCS TESTS OVERFLOW CONDITION
                 ;       BCC TESTS NO OVERFLOW CONDITION
                 ;
2AC4 7F0098      BCDSUB CLR    CARRY
```

```
2AC7 A100          CMP A  0,X
2AC9 2210          BHI    SUBT
2ACB 2604          BNE    SWAP
2ACD E101          CMP B  1,X
2ACF 220A          BHI    SUBT
2AD1 37      SWAP  PSH B
2AD2 36            PSH A
2AD3 7A0098        DEC    CARRY
2AD6 A600          LDA A  0,X
2AD8 E601          LDA B  1,X
2ADA 30            TSX
2ADB 9736    SUBT  STA A  TEMPA
2ADD D737          STA B  TEMPB
2ADF 8699          LDA A  #99H
2AE1 16            TAB
2AE2 A001          SUB A  1,X
2AE4 E000          SUB B  0,X
2AE6 0D            SEC
2AE7 9937          ADC A  TEMPB
2AE9 19            DAA
2AEA 36            PSH A
2AEB 17            TBA
2AEC 9936          ADC A  TEMPA
2AEE 19            DAA
2AEF 33            PUL B
2AF0 7D0098        TST    CARRY
2AF3 2702          BEQ    BACK
2AF5 31            INS
2AF6 31            INS
2AF7 39      BACK  RTS
             ;
             ;     "SHAENC" SUBROUTINE
             ;     ROUTINE THAT TAKES CARE OF READING SHAFT ANGLE ENCODERS

2AF8 B68E01  SHAENC LDA A MSBSAZ  ;READS AZIMUTH ANGLE
2AFB F68E00        LDA B  LSBSAZ
2AFE 9738          STA A  MSBENC  ;STORES ANGLE IN TEMPORARY LOCATION
2B00 D739          STA B  LSBENC
2B02 58            ASL B          ;SCALE DAC OUTPUT BY A FACTOR OF 2
2B03 49            ROL A
2B04 B78404        STA A  DDRA2   ;OUTPUT MS 4 BITS OF AZ TO DAC
2B07 F78406        STA B  DDRB2   ;OUTPUT LS 8 BITS OF AZ TO DAC
2B0A 9638          LDA A  MSBENC  ;GET OLD A AND B
2B0C D639          LDA B  LSBENC
2B0E CE31F8        LDX    #DIVISO
2B11 BD2CAE        JSR    DIVIDE  ;DIVIDES ANGLE BY THE CONSTANT 14.912
2B14 CE0045        LDX    #AZBCD
2B17 BD2C3D        JSR    BINBCD  ;RETURNS WITH A PACKED BCD NUMBER
2B1A 36            PSH A
2B1B 37            PSH B
             ;
             ;     ADDITION TO "SHAENC" SUBROUTINE
             ;     CHECKS BOTH POS AND NEG AZIMUTH LIMITS
```

```
                       ;       (MODIFICATION 1.1)
                       ;
2B1C D737                       STA B  TEMPB    ;SAVES ACCB TEMP
2B1E D642                       LDA B  SIGN     ;START LIMIT CHECK
2B20 F131FB                     CMP B  PLUS
2B23 271C                       BEQ    P.L      ;BRANCHES IF AZIMUTH COORDINATES ARE POSITIVE
2B25 D637                       LDA B  TEMPB    ;AZIMUTH COORDINATES ARE NEGATIVE
2B27 CE0091                     LDX    #NAZLIM  ;GET NEGATIVE AZIMUTH LIMIT
2B2A BD2AC4                     JSR    BCDSUB   ;ACCX MINUS NAZLIM
2B2D 9698                       LDA A  CARRY
2B2F 261C                       BNE    SHA2     ;BRANCHES IF THE ACCXS ARE LARGER THAN NAZLIM
2B31 C6FF              PAL1     LDA B  #0FFH    ;POSITIONER HAS EXCEEDED THE LIMIT
2B33 D794                       STA B  LFLAGA   ;SET AZIMUTH LIMIT FLAG
2B35 CE3174                     LDX    #MSG15
2B38 BD2C22                     JSR    ASCDIS   ;DISPLAYS "ANGLE LIMIT EXCEEDED"
2B3B BD2ABB                     JSR    ALSTOP   ;BRANCHES TO STOP BOTH MOTORS
2B3E 7E20B2                     JMP    MSGB     ;RETURNS TO CONTROL LOOP AFTER 1 SEC. WAIT
2B41 D637              PAL      LDA B  TEMPB.   ;CHECKING FOR POSITIVE AZIMUTH LIMIT
2B43 CE008F                     LDX    #PAZLIM  ;GETING POSITIVE AZIMUTH LIMIT
2B4[illegible] BD2AC4           JSR    BCDSUB   ;ACCXS MINUS PAZLIM
2B49 9698                       LDA A  CARRY
2B4B 27E4                       BEQ    PAL1
2B4D 9642              SHA2     LDA A  SIGN
2B4F 9743                       STA A  AZSIGN
2B51 8641                       LDA A  #041H
2B53 973A                       STA A  LETA
2B55 865A                       LDA A  #05AH
2B57 973B                       STA A  LETB
2B59 33                         PUL B
2B5A 32                         PUL A
2B5B CE003F                     LDX    #ANGLE
2B5E BD2D5F                     JSR    BCDDIS   ;UNPACKS BCD ANGLE
2B61 CE0000                     LDX    #DISAZ
2B64 BD2D39                     JSR    ANG2     ;DISPLAYS PACKED BCD ON THE PANEL
                       ;
2B67 B68E03                     LDA A  MSBSEL   ;READS ELEVATION ANGLE
2B6A F68E02                     LDA B  LSBSEL
2B6D 9738                       STA A  MSBENC   ;STORES ANGLE TEMPORARILY
2B6F D739                       STA B  LSBENC
2B71 58                         ASL B
2B72 49                         ROL A
2B73 B78C00                     STA A  DORA3    ;OUTPUT MS 4 BITS OF EL TO DAC
2B76 F78C02                     STA B  DORB3    ;OUTPUT LS 8 BITS OF EL TO DAC
2B79 9638                       LDA A  MSBENC   ;GET OLD A AND B
2B7B D639                       LDA B  LSBENC
2B7D CE31F8                     LDX    #DIVISO
2B80 BD2CAE                     JSR    DIVIDE   ;DIVIDES ANGLE BY THE CONSTANT 14,912
2B83 CE0047                     LDX    #ELBCD
2B86 BD2C3D                     JSR    BINBCD   ;RETURNS WITH PACKED BCD NUMBER
2B89 36                         PSH A
2B8A 37                         PSH B
2B8B D737                       STA B  TEMPB    ;HANDLES CHANGE IN COORDINATE SYSTEM
2B8D D642                       LDA B  SIGN
```

```
2B8F F131FB            CMP B  PLUS
2B92 2605              BNE    SHA3
2B94 F631FA            LDA B  MINUS
2B97 2003              BRA    SHA4
2B99 F631FB     SHA3   LDA B  PLUS
2B9C D742       SHA4   STA B  SIGN
                ;
                :      ADDITION TO "SHAENC" SUBROUTINE.
                :      CHECKS BOTH POS AND NEG ELEVATION LIMITS
                ;      (MODIFICATION 1.1)
                ;
2B9E D642              LDA B  SIGN    ;START LIMIT CHECK
2BA0 F131FB            CMP B  PLUS
2BA3 D637              LDA B  TEMPB
2BA5 271A              BEQ    PEL     ;BRANCHES IF ELEVATION COORDINATE IS NEGATIVE
2BA7 CE008D            LDX    #NELLIM ;GET NEGATIVE ELEVATION LIMIT
2BAA BD2AC4            JSR    BCDSUB  ;ACCXS MINUS NELLIM
2BAD 9698              LDA A  CARRY
2BAF 261C              BNE    SHA1    ;BRANCHES IF THE ACCXS ARE LARGER THAN NELLIM
2BB1 C6FF       SHA    LDA B  #0FFH   ;POSITIONER HAS EXCEEDED THE LIMIT
2BB3 D793              STA B  LFLAGE  ;SET ELEVATION LIMIT FLAG
2BB5 CE3174            LDX    #MSG15
2BB8 BD2C22            JSR    ASCDIS  ;DISPLAY "ANGLE LIMIT EXCEEDED"
2BBB BD2ABB            JSR    ALSTOP  ;BRANCHES TO STOP BOTH MOTORS
2BBE 7E20B2            JMP    MSGB    ;RETURN TO CONTROL LOOP AFTER 1 SEC. WAIT
S^%68((L^#D6u7    PEL     LDA B  TEMPB   ;CHECKING POSITIVE ELEVATION LIMIT
2BC3 CE008B            LDX    #PELLIM ;GETTING POSITIVE ELEVATION LIMIT
2BC6 BD2AC4            JSR    BCDSUB  ;ACCXS MINUS PELLIM
2BC9 9698              LDA A  CARRY
2BCB 27E4              BEQ    SHA
2BCD 9642       SHA1   LDA A  SIGN
2BCF 9744              STA A  ELSIGN
2BD1 8645              LDA A  #045H
2BD3 973A              STA A  LETA
2BD5 864C              LDA A  #04CH
2BD7 973B              STA A  LETB
2BD9 33                PUL B
2BDA 32                PUL A
2BDB CE003F            LDX    #ANGLE
2BDE BD2D2F            JSR    BCDDIS  ;UNPACKS BCD ANGLE
2BE1 CE000A            LDX    #DISEL
2BE4 BD2D59            JSR    ASC2    ;DISPLAYS PACKED BCD ON PANEL
2BE7 39                RTS
                ;
                :      "ADDCAL" SUBROUTINE
                ;      ROUTINE THAT CALCULATES ADDRESSES FOR NEXT STATE
                ;      ACCA CONTAINS VARIABLE INDEX.
                :      INDEX REG. CONTAINS CURRENT STATE TABLE
                ;      ADDRESS FOR REFERENCING NEXT STATE.
                :
2BE8 9736       ADDCAL STA A  TEMPA
2BEA DF29              STX    TEMPX   ;STORES INDEX REGISTER TEMPORARILY
2BEC 5F                CLR B
```

```
2BED 9B2A                ADD A  TEMPX+1
2BEF D929                ADC B  TEMPX    ;ADDS KEYCODE TO INDEX REGISTER
2BF1 D729                STA B  TEMPX
2BF3 972A                STA A  TEMPX+1
2BF5 DE29                LDX    TEMPX    ;UPDATES INDEX REGISTER
2BF7 9636                LDA A  TEMPA
2BF9 39                  RTS
                 ;
                 ;       "UP" ROUTINE
                 ;       INSTRUCTS EL. MOTOR TO GO UP (CCW)
                 ;
2BFA 965C        UP      LDA A  SPEEDE
2BFC C600                LDA B  #000H
2BFE BD2DE2              JSR    MOTEL
2C01 7E20B7              JMP    STO
                 ;
                 ;       "DOWN" ROUTINE
                 ;       INSTRUCTS EL. MOTOR TO GO DOWN (CW)
                 ;
2C04 965C        DOWN    LDA A  SPEEDE
2C06 C6FF                LDA B  #0FFH
2C08 BD2DE2              JSR    MOTEL
2C0B 7E20B7              JMP    STO
                 ;
                 ;       "LEFT" ROUTINE
                 ;       INSTRUCTS AZ. MOTOR TO GO LEFT (CCW)
                 ;
2C0E 965B        LEFT    LDA A  SPEEDA
2C10 C600                LDA B  #000H
2C12 BD2D36              JSR    MOTAZ
2C15 7E2158              JMP    STOE
                 ;
                 ;       "RIGHT" ROUTINE
                 ;       INSTRUCTS AZ. MOTOR TO GO RIGHT (CW)
                 ;
2C18 965B        RIGHT   LDA A  SPEEDA
2C1A C6FF                LDA B  #0FFH
2C1C BD2D36              JSR    MOTAZ
2C1F 7E2158              JMP    STOE
                 ;
                 ;       "ASCDIS" SUBROUTINE
                 ;       DISPLAYS ASCII MSG. ON DISPLAY PANEL
                 ;       INDEX REG. HAS STARTING ADDRESS OF MSG.
                 ;
2C22 DF49        ASCDIS  STX    TEMPX1
2C24 CE0000              LDX    #0000H
2C27 DF4B                STX    TEMPX2
2C29 DE49        OVER    LDX    TEMPX1
2C2B A600                LDA A  0,X
2C2D 08                  INX
2C2E DF49                STX    TEMPX1
2C30 DE4B                LDX    TEMPX2
2C32 A700                STA A  0,X
```

```
2C34 08                     INX
2C35 FF4B                   STX     TEMPX2
2C37 8C0014                 CPX     #0014H
2C3A 26ED                   BNE     CVTR
2C3C 39                     RTS
                    ;
                    ;       "BIN3CD" SUBROUTINE
                    ;       BINARY TO PACKED BCD CONVERSION ROUTINE
                    ;       REVISION 1.1
                    ;       LOAD ACCA, ACCB WITH A 16-BIT BINARY NUMBER.
                    ;       LOAD INDEX REG. WITH ADDRESS OF MS BYTE OF
                    ;       WHERE PACKED BCD;INFORMATION IS TO BE KEPT.
                    ;       ROUTINE DETERMINES SIGN OF 16-BIT ENTRY
                    ;       AND SAVES THAT INFORMATION IN "SIGN".
                    ;       ROUTINE RETURNS WITH PACKED BCD IN THE
                    ;       SPECIFIED MEMORY LOCATION AND ALSO IN ACCA,
                    ;       ACCB.  THE LEAST FOUR MS DECIMAL VALUES
                    ;       WILL BE CONTAINED IN THE PACKED BCD NUMBER
                    ;
2C3D DF32           BINBCD  STX     SAVEX   ;SAVE DATA POINTER
2C3F 9736                   STA A   TEMPA
2C41 D737                   STA B   TEMPB   ;SAVE ACCA AND ACCB TEMPORARILY
2C43 9638                   LDA A   MSBENC
2C45 D639                   LDA B   LSBENC  ;SAVE THE 16-BIT SHAFT ANGLE CODE
2C47 C000                   SUB B   #000H
2C49 8210                   SBC A   #010H
2C4B 2B11                   BMI     POS     ;BRANCH IF 16-BIT VALUE IS POSITIVE
2C4D 863C                   LDA A   #03CH   ;16-BIT VALUE MUST BE NEGATIVE
2C4F C69F                   LDA B   #09FH
2C51 D037                   SUB B   TEMPB   ;AFTER SUBTRACT ACCB WILL CONTAIN CORRECT VALUE
2C53 9236                   SBC A   TEMPA   ;SUBTRACT 16-BIT VALUE FROM 360 DEGREES
2C55 9736                   STA A   TEMPA
2C57 B631FA                 LDA A   MINUS
2C5A 9742                   STA A   SIGN    ;SAVE ASCII VALUE OF MINUS SIGN
2C5C 2007                   BRA     POSN
2C5E B631FB         POS     LDA A   PLUS
2C61 9742                   STA A   SIGN    ;SAVE ASCII VALUE OF PLUS SIGN
2C63 D637                   LDA B   TEMPB   ;RESTORE ACCB TO PROPER VALUE
2C65 9636           POSN    LDA A   TEMPA   ;RESTORE ACCA TO PROPER VALUE
2C67 CE31EC                 LDX     #K10K   ;INITIALIZES INDEX REG. FOR FIRST BCD CONVERSION CONSTANT
2C6A 7F0031         CVDEC1  CLR     SAVEA   ;CLEAR BCD CONVERSION COUNTER
2C6D E001           CVDEC2  SUB B   1,X
2C6F A200                   SBC A   0,X
2C71 2505                   BCS     CVDEC5  ;BRANCH IF SUBTACTION PRODUCES OVERFLOW
2C73 7C0031                 INC     SAVEA   ;DECIMAL CHARACTER BEING BUILT, INCREMENT SAVEA
2C76 20F5                   BRA     CVDEC2
2C78 EB01           CVDEC5  ADD B   1,X
2C7A A900                   ADC A   0,X     ;RESTORES PARTIAL RESULT UPON OVERFLOW
2C7C 36                     PSH A           ;SAVES ACCA
2C7D 78003D                 ASL     BCDB    ;16-BIT SHIFTS LEFT WITH ZERO FILL
2C80 79003C                 ROL     BCDA    ;SO, NO NEED TO CLEAR EITHER REGISTER
2C83 78003D                 ASL     BCDB
2C86 79003C                 ROL     BCDA
```

```
2C89 78003D          ASL     BCDB
2C8C 79003C          ROL     BCDA
2C8F 78003D          ASL     BCDB
2C92 79003C          ROL     BCDA     ;SHIFTS LAST BCD VALUE OVER ONE CHARACTER
2C95 963D            LDA A   BCDB     ;LOOKS LIKE "X0"
2C97 9B31            ADD A   SAVEA    ;PRODUCES BCD CHARACTER
2C99 973D            STA A   BCDB     ;RESTORES PACKING REGISTER BCDB WITH CORRECT VALUE
2C9B 32              PUL A            ;RESTORES ACCA TO FORMER VALUE
2C9C 08              INX
2C9D 08              INX              ;INCREMENTS INDEX REGISTER TO NEXT CONSTANT
2C9E 8C31F6          CPX     #K10K+10 ; TESTS TO SEE LAST CONSTANT HAS BEEN USED
2CA1 26C7            BNE     CVDEC1   ;BRANCHES IF LAST CHARACTER HAS NOT BEEN REACHED YET
2CA3 963C            LDA A   BCDA
2CA5 D63D            LDA B   BCDB
2CA7 DE32            LDX     SAVEX
2CA9 A700            STA A   0,X
2CAB E701            STA B   1,X      ;SAVES 16-BIT PACKED BCD NUMBER
2CAD 39              RTS
                ;
                ;    "DIVIDE" SUBROUTINE
                ;    16-BIT INTEGER DIVIDE ROUTINE
                ;    REVISION 1.1
                ;    LOAD ACCA, ACCB WITH 16-BIT DIVIDEND.
                ;    LOAD INDEX REGISTER WITH 16-BIT DIVISOR.
                ;    RETURNS WITH 16-BIT QUOTIENT IN ACCA, ACCB.
                ;    THIS IS AN ASSUMED DIVIDE ROUTINE.
                ;
2CAE 9736       DIVIDE STA A TEMPA    ;SAVES ACCA TEMPORARILY
2CB0 8600            LDA A   #000H
2CB2 36              PSH A
2CB3 36              PSH A            ;CLEARS SPACE ON STACK FOR QUOTIENT
2CB4 36              PSH A
2CB5 36              PSH A            ;PRODUCES CLEARED SPACE FOR THE LS BYTES OF DIVIDEND
2CB6 9636            LDA A   TEMPA
2CB8 37              PSH B
2CB9 36              PSH A            ;FORMS REST OF DIVIDEND BY ADDING ACCA, ACCB TO STACK
2CBA A600            LDA A   0,X
2CBC E601            LDA B   1,X      ;FETCHES 16-BIT DIVISOR
2CBE 37              PSH B
2CBF 36              PSH A            ;FORMS TWO LS BYTES OF DIVISOR BY ADDING ACCA, ACCB TO STACK
2CC0 8600            LDA A   #000H
2CC2 36              PSH A
2CC3 36              PSH A            ;PRODUCES REST OF 32-BIT DIVISOR BY ADDING ZEROS TO STACK
2CC4 34              DES
2CC5 30              TSX              ;POINTER IN INDEX REG. TO STACKED DATA
2CC6 8601            LDA A   #1
2CC8 6D01            TST     1,X
2CCA 2B0F            BMI     DIV153   ;BRANCHES IF DIVISOR HAS BEEN LEFT JUSTIFIED
2CCC 4C         DIV151 INC A          ;COUNTS NUMBER OF TIMES DIVISOR IS SHIFTED LEFT
2CCD 6804            ASL     4,X
2CCF 6903            ROL     3,X
2CD1 6902            ROL     2,X
2CD3 6901            ROL     1,X      ;SHIFTS DIVISOR LEFT ONE BIT
```

```
2CD5 2B04                BMI    DIV153
2CD7 8121                CMP A  [illegible]
2CD9 26F1                BNE    DIV151   ;BRANCHES IF DIVISOR HAS BEEN SHIFTED LEFT 32 TIMES
2CDB A700         DIV153 STA A  0,X      ;SAVES SHIFT COUNT ON STACK
                  ;      STACK LOOKS LIKE THIS@D
                  ;      +0  XX COUNT
                  ;      +1  00 MS BYTE OF DIVISOR
                  ;      +2  00
                  ;      +3  XX
                  ;      +4  XX LS BYTE OF DIVISOR
                  ;      +5  XX MS BYTE OF DIVIDEND
                  ;      +6  XX
                  ;      +7  00
                  ;      +8  00 LS BYTE OF DIVIDEND
                  ;      +9  00 MS BYTE OF QUOTIENT
                  ;      +10 00 LS BYTE OF QUOTIENT
                  ;      +11 XX MS BYTE OF RETURN ADDR.
                  ;      +12 XX LS BYTE OF RETURN ADDR.
2CDD A607         DIV163 LDA A  7,X      ;BEGIN TO FORM NEW REMAINER FROM OLD DIVIDEND
2CDF E608                LDA B  8,X
2CE1 E004                SUB B  4,X
2CE3 A203                SBC A  3,X
2CE5 A707                STA A  7,X
2CE7 E708                STA B  8,X
2CE9 A605                LDA A  5,X
2CEB E606                LDA B  6,X
2CED E202                SBC B  2,X
2CEF A201                SBC A  1,X      ;SUBTRACTS DIVISOR FROM DIVIDEND
2CF1 A705                STA A  5,X
2CF3 E706                STA B  6,X
2CF5 241E                BCC    DIV165   ;BRANCHES IF NO BORROW FROM SUBTRACT
2CF7 A607                LDA A  7,X      ;BEGINS TO RESTORE REMAINDER TO PREVIOUS DIVIDEND
2CF9 E608                LDA B  8,X
2CFB E@A04                ADD B  4,X
2CFD A903                ADC A  3,X
2CFF A707                STA A  7,X
2D01 E708                STA B  8,X
2D03 A605                LDA A  5,X
2D05 E606                LDA B  6,X
2D07 E902                ADC B  2,X
2D09 A901                ADC A  1,X      ;RESTORES LAST SUBTRACT OPERATION IN CASE OF BORROW
2D0B A705                STA A  5,X
2D0D E706                STA B  6,X
2D0F 0C                  CLC
2D10 2001                BRA    DIV167
2D12 0D           DIV165 SEC             ;SETS, ASSUMES BINARY 1 FOR THIS PART OF THE DIVIDE
2D13 690A         DIV167 ROL    10,X
2D15 6909                ROL    9,X      ;SHIFTS BINARY 1 OR 0 INTO LSB OF QUOTIENT
2D17 6401                LSR    1,X
2D19 6602                ROR    2,X
2D1B 6603                ROR    3,X
2D1D 6604                ROR    4,X      ;SHIFTS DIVISOR RIGHT WITH ZERO FILL
2D1F 6A00                DEC    0,X      ;DECREMENTS COUNT
```

```
02015 2D21 25BA            BNE    DIV163  ;BRANCHES IF COUNT = 0
02016 2D23 31              INS
02017 2D24 31              INS
02018 2D25 31              INS
02019 2D26 31              INS
02020 2D27 31              INS
02021 2D28 31              INS
02022 2D29 31              INS
02023 2D2A 31              INS
02024 2D2B 31              INS            ;CLEANS UP STACK
02025 2D2C 32              PUL A
02026 2D2D 33              PUL B          ;PULLS CORRECTED 16-BIT QUOTIENT FROM STACK, DESTROYS REMAINDER
02027 2D2E 39              RTS
02028                ;
02029                ;      "BCDDIS" SUBROUTINE
02030                ;      REVISION 1.1
02031                ;      CONVERTS PACKED BCD ANGLES TO UNPACKED ASCII VALUES
02032                ;      TO BE DISPLAYED IN AZ, EL FORMAT.  INDEX REGISTER SHOULD
02033                ;      CONTAIN STARTING ADDRESS OF WHERE INFORMATION SHOULD BE
02034                ;      SAVED.  ROUTINE DESTROYS INITIAL ACCA, ACCB VALUES.
02035                ;      (MODIFICATION 1.1)
02036                ;
02037 2D2F 973C      BCDDIS STA A  BCDA
02038 2D31 8603             LDA A  #3
02039 2D33 973E             STA A  SAVDEC
02040 2D35 963C             LDA A  BCDA
02041 2D37 44               LSR A
02042 2D38 56               ROR B
02043 2D39 44               LSR A
02044 2D3A 56               ROR B
02045 2D3B 44               LSR A
02046 2D3C 56               ROR B
02047 2D3D 44               LSR A          ;SHIFTS PACKED BCD ANGLE RIGHT ONE BCD VALUE
02048 2D3E 56               ROR B          ;LOSING SECOND DECIMAL PLACE ACCURRACY
02049 2D3F 840F      ASC1   AND A  #00FH   ;MASKS OFF BCD VALUE
02050 2D41 8B30             ADD A  #030H   ;CONVERTS THIS VALUE TO ASCII
02051 2D43 A700             STA A  0,X     ;DISPLAYS THAT BCD VALUE
02052 2D45 7A003E           DEC    SAVDEC  ;DECREMENTS UNPACKING COUNT
02053 2D48 7D003E           TST    SAVDEC
02054 2D4B 270B             BEQ    ASC3    ;BRANCHES IF PACKED BCD NUMBER IS COMPLETELY UNPACKED
02055 2D4D 58               ASL B
02056 2D4E 49               ROL A
02057 2D4F 58               ASL B
02058 2D50 49               ROL A
02059 2D51 58               ASL B
02060 2D52 49               ROL A
02061 2D53 58               ASL B
02062 2D54 49               ROL A          ;SHIFTS PACKED BCD NUMBER LEFT ONE BCD VALUE
02063 2D55 08               INX            ;INCREMENTS TO NEXT DISPLAY LOCATION
02064 2D56 20E7             BRA    ASC1
02065 2D58 39        ASC3   RTS
02066                ;
02067                ;      "ASC2" SUBROUTINE
```

```
                          ;      DISPLAYS UNPACKED BCD NUMBERS IN AZ, EL FORMATS
                          ;      LETA, LETB SHOULD CONTAIN ASCII VALUES OF LETTERS.
                          ;      (MODIFICATION 1.1)
                          ;
2D59 B631FC       ASC2    LDA A  BLANK
2D5C A700                 STA A  0.X
2D5E 9642                 LDA A  SIGN
2D60 A701                 STA A  1,X      ;DISPLAYS SIGN OF ANGLE
2D62 963F                 LDA A  ANGLE
2D64 A702                 STA A  2.X
2D66 9640                 LDA A  ANGLE+1
2D68 A703                 STA A  3,X
2D6A B631FE               LDA A  POINT
2D6D A704                 STA A  4.X
2D6F 9641                 LDA A  ANGLE+2
2D71 A705                 STA A  5,X
2D73 B631FD               LDA A  DEGMAR
2D76 A706                 STA A  6,X      ;DISPLAYS ANGLE IN DEGREES
2D78 B631FC               LDA A  BLANK
2D7B A707                 STA A  7,X
2D7D 963A                 LDA A  LETA
2D7F A708                 STA A  8,X
2D81 963B                 LDA A  LETB
2D83 A709                 STA A  9,X      ;DISPLAYS WHAT THE ANGLE IS@D  AZ, EL.
2D85 39                   RTS
                          ;
                          ;      "MOTAZ" SUBROUTINE
                          ;      AZIMUTH MOTOR SUBROUTINE
                          ;      PLACE SPEED IN ACCA AND PLACE DIRECTION IN ACCB
                          ;
2D86 D16C         MOTAZ   CMP B  DFLAGA   ;DECIDE IF DESIRED DIRECTION MATCHES CURRENT DIRECTION
2D88 2752                 BEQ    SAME1    ;BRANCHES IF BOTH DIRECTIONS ARE THE SAME
2D8A D66C         DIFF1   LDA B  DFLAGA   ;DESIRED DIRECTION IS DIFFERENT FROM CURRENT DIRECTION
2D8C 2B28                 BMI    ONE1     ;BRANCHES IF CURRENT DIRECTION IS CW (RIGHT)
2D8E C600         ZERO1   LDA B  #000H    ;DIRECTION, HERE, IS CURRENTLY CCW (LEFT)
2D90 F78E01               STA B  MSBSAZ   ;TURNS POWER OFF TO AZIMUTH MOTOR
2D93 7D006E               TST    SFLAGA   ;FIND OUT WHAT THE SPEED IS
2D96 2704                 BEQ    Z2       ;BRANCHES IF MOTOR IS CURRENTLY IN MOTION
2D98 C601                 LDA B  #1       ;WAIT .1 SECOND
2D9A 2002                 BRA    Z3       ;BRANCH TO WAIT ONLY .1 SECOND INSTEAD OF 2
2D9C C614         Z2      LDA B  #20
2D9E BD2E3E       Z3      JSR    WAITE    ;WAIT FOR MOTOR TO STOP BEFORE SWITCHING DIRECTIONS
2DA1 C602                 LDA B  #002H
2DA3 F78E01               STA B  MSBSAZ   ;SWITCHING FROM CCW TO CW DIRECTION (RIGHT)
2DA6 C601                 LDA B  #1
2DA8 BD2E3E               JSR    WAITE    ;WAIT FOR DIRECTION RELAY TO HAVE TIME TO SWITCH
2DAB C603                 LDA B  #003H
2DAD F78E01               STA B  MSBSAZ   ;TURNS POWER ON, MOTOR NOW ORIENTED CW (RIGHT)
2DB0 C6FF                 LDA B  #0FFH
2DB2 D76C                 STA B  DFLAGA   ;SAVE DIRECTION OF AZIMUTH MOTOR BY SETTING DIRECTION FLAG
2DB4 2026                 BRA    SAME1
2DB6 C602         ONE1    LDA B  #002H    ;DIRECTION IS CURRENTLY CW (RIGHT)
2DB8 F78E01               STA B  MSBSAZ   ;TURNS POWER OFF OF AZIMUTH MOTOR
```

```
2DBB 7D006E          TST     SFLAGA   ;FIND OUT WHAT THE SPEED IS
2DBE 2704            BEQ     02       ;BRANCHES IF AZIMUTH MOTOR IS IN MOTION
2DC0 C601            LDA B   #1
2DC2 2002            BRA     03       ;BRANCHES TO WAIT ONLY FOR ON/OFF RELAY TO CHANGE
2DC4 C614     02     LDA B   #20
2DC6 BD2E3E   03     JSR     WAITE    ;WAITS FOR AZIMUTH MOTOR TO STOP
2DC9 C600            LDA B   #000H
2DCB F78E01          STA B   MSBSAZ   ;CHANGES DIRECTION FROM CW (RIGHT) TO CCW (LEFT)
2DCE C601            LDA B   #1
2DD0 BD2E3E          JSR     WAITE    ;WAIT FOR DIRECTION RELAY TO CHANGE
2DD3 C601            LDA B   #001H
2DD5 F78E01          STA B   MSBSAZ   ;TURNS POWER ON TO AZIMUTH MOTOR, ORIENTED IN CCW (LEFT)
2DD8 C600            LDA B   #000H
2DDA D76C            STA B   DFLACA   ;REMEMBERS DIRECTION OF AZ MOTOR BY SETTING DIR. FLAG
2DDC B78E00   SAME1  STA A   LSBSAZ   ;CHANGES SPEED OF AZIMUTH MOTOR
2DDF 976E            STA A   SFLACA   ;UPDATE RECORD OF SPEED OF AZIMUTH MOTOR
2DE1 39              RTS
              ;
              ;      "MOTEL" SUBROUTINE
              ;      ELEVATION MOTOR SUBROUTINE
              ;      PLACE SPEED IN ACCA, PLACE DIRECTION IN ACCB
              ;
2DE2 D16D     MOTEL  CMP B   DFLAGE   ;COMPARE DESIRED DIRECTION WITH CURRENT DIRECTION
2DE4 2752            BEQ     SAME     ;BRANCHES IF THE DIRECTIONS ARE THE SAME
2DE6 D66D     DIFF   LDA B   DFLAGE   ;DESIRED DIRECTION IS DIFFERENT FROM CURRENT DIRECTION
2DE8 2B28            BMI     ONE      ;BRANCHES IF CURRENT DIRECTION IS CW (DOWN)
2DEA C600     ZERO   LDA B   #000H    ;CURRENT DIRECTION IS CCW (UP)
2DEC F78E03          STA B   MSBSEL   ;TURNS POWER OFF TO ELEVATION MOTOR
2DEF 7D006F          TST     SFLAGE   ;CHECK ON SPEED OF EL MOTOR
2DF2 2704            BEQ     Z4       ;BRANCHES IF EL MOTOR IS IN MOTION
2DF4 C601            LDA B   #1       ;WAIT FOR .1 SECOND
2DF6 2002            BRA     Z5       ;BRANCHES TO WAIT ONLY FOR ON-OFF RELAY TO CHANGE
2DF8 C614     Z4     LDA B   #20
2DFA BD2E3E   Z5     JSR     WAITE    ;WAITS FOR MOTOR TO STOP MOVING
2DFD C602            LDA B   #002H
2DFF F78E03          STA B   MSBSEL   ;SWITCHES TO CW (DOWN) FROM CCW (UP)
2E02 C601            LDA B   #1
2E04 BD2E3E          JSR     WAITE    ;WAITS FOR DIRECTION RELAY TO SWITCH
2E07 C603            LDA B   #003H
2E09 F78E03          STA B   MSBSEL   ;TURNS POWER ON, ORIENTED IN CW (DOWN)
2E0C C6FF            LDA B   #0FFH
2E0E D76D            STA B   DFLAGE   ;REMEMBERS DIRECTION CHANGE BY SETTING DIRECTION FLAG
2E10 2026            BRA     SAME
2E12 C602     ONE    LDA B   #002H    ;DIRECTION IS CURRENTLY CW (DOWN)
2E14 F78E03          STA B   MSBSEL   ;TURNS POWER OFF OF ELEVATION MOTOR
2E17 7D006F          TST     SFLAGE   ;CHECK ON SPEED OF EL MOTOR
2E1A 2704            BEQ     04       ;BRANCHES IF ELEVATION MOTOR IS MOVING
2E1C C601            LDA B   #1
2E1E 2002            BRA     05       ;BRANCH TO WAIT .1 SECOND FOR ON-OFF RELAY TO SWITCH
2E20 C614     04     LDA B   #20
2E22 BD2E3E   05     JSR     WAITE    ;WAITS 2 SECONDS FOR EL MOTOR TO STOP MOVING
2E25 C600            LDA B   #000H
2E27 F78E03          STA B   MSBSEL   ;SWITCHES DIRECTION TO CCW (UP) FROM CW (DOWN)
```

```
2E2A C601             LDA B  #1
2E2C BD2E3E           JSR    WAITE    ;WAITS .1 SECOND FOR DIRECTION RELAY TO CHANGE
2E2F C601             LDA B  #001H
2E31 F78E03           STA B  MSBSEL   ;TURNS POWER ON, ORIENTED IN CCW (UP)
2E34 C600             LDA B  #000H
2E36 D76D             STA B  DFLAGE   ;REMEMBERS DIRECTION CHANGE BY SETTING DIRECTION FLAG
2E38 B78E02    SAME   STA A  LSBSEL   ;CHANGES SPEED OF ELEVATION MOTOR
2E3B 976F             STA A  SFLAGE   ;UPDATE RECORD OF EL MOTOR SPEED
2E3D 39               RTS
                ;
                ;     "WAITE" SUBROUTINE
                ;     DELAY IS IN HUNDREDS OF MILLISECONDS
                ;     PLACE DESIRED DELAY IN ACCB
                ;     THEN JUMP TO SUBROUTINE "WAITE"
                ;
2E3E CE4074    WAITE  LDX    #04074H
2E41 09        WAITE1 DEX
2E42 26FD             BNE    WAITE1
2E44 5A               DEC B
2E45 26F7             BNE    WAITE
2E47 39               RTS
                ;
                ;     KEYBOARD SERVICE ROUTINE
                ;     SERVICED ON INTERRUPT BASIS ONLY
                ;
2E48 86FF      KEYBD  LDA A  #0FFH
2E4A B78400           STA A  DDRA     ;MAKE ALL FOUR ROWS OF KEYBOARD INOPERABLE
2E4D B68400           LDA A  DDRA     ;NOW, CLEAR PIA FLAGS WITH MPU READ
2E50 8670      COLUMN LDA A  #070H
2E52 16               TAB
2E53 B78400    COLUM1 STA A  DDRA     ;TURN ON ROW #1 OF KEYBD
2E56 B68401           LDA A  CRA      ;READ CONTROL REGISTER OF PIA #1
2E59 48               ASL A
2E5A 2B11             BMI    DECODE   ;BRANCH IF KEY IN FIFTH COLUMN IS SET
2E5C 250C             BCS    ROW      ;BRANCH IF KEY IN COLUMNS 1-4 IS SET
2E5E C1E0             CMP B  #0E0H
2E60 2723             BEQ    K1       ;BRANCH IF NO KEY PRESSED, RESET PIA, RETURN FROM INTERRUPT
2E62 57               ASR B
2E63 CA80             ORA B  #080H
2E65 C4F0             AND B  #0F0H    ;PREPARES ACCB FOR NEXT ROW READ
2E67 17               TBA
2E68 20E9             BRA    COLUM1   ;BRANCH TO READ NEXT ROW
2E6A F68400    ROW    LDA B  DDRA     ;READS DDRA IF KEY IN COLUMNS 1-4 WAS PRESSED
2E6D CE2F6A    DECODE LDX    #KTABLE  ;BEGIN COMPARING KEYTABLE WITH KEY THAT WAS PRESSED
2E70 A600      DECOD1 LDA A  0,X
2E72 11               CBA
2E73 2708             BEQ    GTCHAR   ;FOUND MATCH BETWEEN KEYREAD AND KEYTABLE
2E75 8C2F7D           CPX    #KTABLE+19
2E78 270B             BEQ    K1       ;BRANCH IF NO MATCH IS FOUND
2E7A 08               INX
2E7B 20F3             BRA    DECOD1   ;BRANCH TO KEEP LOOKING FOR MATCH
2E7D A614      GTCHAR LDA A  20,X
2E7F C6FF             LDA B  #0FFH
```

```
2E81 D76B              STA B  KFLAG    ;SET FLAG TO KNOW WHEN KEY WAS PRESSED
2E83 974F              STA A  KEYENT   ;STORE THE KEYCODE OF THE KEY THAT WAS PRESSED
2E85 860F      K1      LDA A  #00FH
2E87 B78400            STA A  DDRA     ;RESTORE KEYBOARD PIA
2E8A B68400            LDA A  DDRA     ;CLEAR PIA FLAG BY MPU READ
2E8D 3B                RTI
                ;
                ;      DISPLAY SERVICE ROUTINE
                ;      (INTERRUPT BASIS ONLY)
                ;
2E8E B68402    DISPL   LDA A  DDRB     ;CLEAR PIA FLAG BY MPU READ
2E91 DE70              LDX    TEMPD
2E93 8C0014            CPX    #014H
2E96 2709              BEQ    START    ;BRANCH IF FIRST CHARACTER OF DISPLAY NEEDED
2E98 A600              LDA A  0,X
2E9A B78402            STA A  DDRB     ;SUPPLY CHARACTER TO DISPLAY
2E9D 08                INX
2E9E DF70              STX    TEMPD    ;INCREMENT CHARACTER COUNTER FOR DISPLAY
2EA0 3B                RTI
2EA1 CE0000    START   LDX    #000H
2EA4 A600              LDA A  0,X
2EA6 B78402            STA A  DDRB
2EA9 08                INX
2EAA DF70              STX    TEMPD
2EAC 8635      RESET   LDA A  #035H
2EAE B78403            STA A  CRB      ;MAKE PIA RESET BIT GO LOW TO RESET THE DISPLAY
2EB1 8601              LDA A  #1
2EB3 8D06              BSR    WAIT     ;WAIT ONE MILLISECOND
2EB5 863D              LDA A  #03DH
2EB7 B78403            STA A  CRB      ;MAKE PIA RESET BIT GO HIGH TO FINISH NEEDED RESET TO DISPLAY
2EBA 3B                RTI
2EBB C6A5      WAIT    LDA B  #0A5H
2EBD 5A        WAIT1   DEC B
2EBE 26FD              BNE    WAIT1
2EC0 4A                DEC A
2EC1 26F8              BNE    WAIT
2EC3 39                RTS
                ;
                ;      "RCVR" SUBROUTINE
                ;      ACIA RECEIVER HANDLER
                ;      OCCURS ONLY AFTER RCVR INTERRUPT
                ;      (MODIFICATION 1.1)
                ;
2EC4 B58409    RCVR    LDA A  ACIAD    ;GET CHARACTER
2EC7 CE2F..    RCVDEC  LDX    #CHARTB  ;LOAD POINTER TO COMMAND TABLE
2ECA 5F                CLR B
2ECB A100      RCVNEX  CMP A  0,X      ;COMPARE WITH VALID COMMAND
2ECD 2603              BNE    NMATCH   ;NO MATCH, TRY ANOTHER ONE
2ECF D74F              STA B  KEYENT   ;LOAD KEYBOARD COMMAND
2ED1 86FF              LDA A  #0FFH
2ED3 976B              STA A  KFLAG    ;SET KEYENTRY FLAG
2ED5 200?              BRA    RCVEND   ;RETURN FROM SUBROUTINE
2ED7 08        NMATCH  INX             ;MAKE NEXT COMPARISON AVAILABLE
```

```
2ED8 5C                   INC B            ;MAKE NEXT CODE
2ED9 5C                   INC B
2EDA C120                 CMP B  #20H      ;TABLE END ?
2EDC 26ED                 BNE    ROTNEX    ;NO!
2EDE 39          RCVEND   RTS              ;RETURN FROM SUBROUTINE, DONT GO INTO MCOMND
                 ;
                 ;        "MCOMND" SUBROUTINE
                 ;        TRANSFERS DISPLAY CONTENTS TO SERIAL INTERFACE BUFFER
                 ;        TO BE SEND OUT.
                 ;        (MODIFICATION 1.1)
                 ;
2EDF CE0014      MCOMND   LDX    #SIBUF    ;INITIALIZE VARIABLE WITH BEGINNING
2EE2 DF2C                 STX    CHARPT    ;ADDRESS OF SERIAL INTERFACE BUFFER
2EE4 DF96                 STX    SAVEX2
2EE6 7F0050               CLR    TEMPA1     ;INITIALIZE VARIABLE TO BEGINNING ADDRESS OF DISPLAY BUFFER
2EE9 7F0051               CLR    TEMPB1
2EEC DE50        MCOM1    LDX    TEMPA1     ;GET NEXT ADDRESS OF DISPLAY BUFFER
2EEE A600                 LDA A  0,X       ;FETCH ONE BYTE CONSTANT
2EF0 08                   INX              ;INCREMENT DISPLAY BUFFER ADDRESS
2EF1 DF50                 STX    TEMPA1     ;SAVE ADDRESS OF NEXT BYTE
2EF3 DE96                 LDX    SAVEX2     ;GET NEXT ADDRESS OF SERIAL INTERFACE BUFFER
2EF5 A700                 STA A  0,X       ;PUT BYTE INTO BUFFER
2EF7 08                   INX              ;INCREMENT SERIAL INTERFACE BUFFER ADDRESS
2EF8 DF96                 STX    SAVEX2     ;SAVE ADDRESS OF NEXT BYTE
2EFA 8C0028               CPX    #SIBUF+20;SAVE ADDRESS OF NEXT BYTE
2EFD 26ED                 BNE    MCOM1     ;NO
2EFF CE0014               LDX    #SIBUF
2F02 C615                 LDA B  #21
2F04 BD2F08               JSR    SEND
2F07 39                   RTS
                 ;
                 ;        "SEND" SUBROUTINE
                 ;        MESSAGE INITIATOR FOR INTERRUPT
                 ;        DRIVER ACIA TRANSMIT ROUTINE
                 ;        X POINTS TO MESSAGE. B=NO. OF BYTE
                 ;        (MODIFICATION 1.1)
                 ;
2F08 DF2C        SEND     STX    CHARPT    ;UPDATE CHARACTER POINTER
2F0A D72E                 STA B  CHARCT    ;SET BYTE COUNT
2F0C 86FF                 LDA A  #0FFH
2F0E 9795                 STA A  MSGFLG    ;SET MESSAGE FLAG
2F10 86A1                 LDA A  #0A1H
2F12 B78408               STA A  ACIAS     ;TURN ON TXMIT INTERRUPT
2F15 7F0030               CLR    CURNUM
2F18 39                   RTS
                 ;
                 ;        "TXMIT" SUBROUTINE
                 ;        INTERRUPT DRIVEN ACIA TRANSMITTER
                 ;        (MODIFICATION 1.1)
                 ;
2F19 D62E        TXMIT    LDA B  CHARCT
2F1B DE2C                 LDX    CHARPT    ;GET NUMBER OF CHARACTERS TO BE TRANSMITTED
2F1D D130                 CMP B  CURNUM    ;GET CHARACTER LENGTH
```

```
02333 2F1F 270D              BEQ     MSGDON   ;CHRNUM = CHARCT
02334 2F21 A600              LDA A   0,X
02335 2F23 B73409            STA A   ACIAD    ;INSERT ONE BYTE INTO ACIA DATA REGISTER
02336 2F26 08                INX
02337 2F27 DF2C              STX     CHARPT   ;SAVE UPDATED VALUE
02338 2F29 7C0030            INC     CHRNUM   ;UPDATE CURRENT # OF HOW MANY BYTES HAVE BEEN SENT OUT
02339 2F2C 2003              BRA     TXMEND   ;RETURN FROM SUBROUTINE
02340 2F2E BD2E0F     MSGDON JSR     MCOMND   ;RESTART MESSAGE
02341 2F31 39         TXMEND RTS
02342                 ;
02343                 ;      INTERRUPT SERVICING ROUTINE
02344                 ;      (MODIFICATION 1.1)
02345                 ;
02346 2F32 B68401     INT    LDA A   CRA      ;CHECKING ON STATUS OF KEYBOARD INTERRUPT FLAG
02347 2F35 2A10              BPL     INT1     ;NOT KEYBOARD, CHECK STATUS OF SERIAL INTERFACE
02348 2F37 B68404            LDA A   DDRA2    ;KEYBOARD DOES NEED SERVICE, BUT FIRST CHECK SWITCH
02349 2F3A 2B0B              BMI     INT1     ;SWITCH ON, IGNORE KEYBOARD
02350 2F3C B63409            LDA A   ACIAD    ;CLEAR ACIA INTERRUPT
02351 2F3F 8681              LDA A   #81H
02352 2F41 B73408            STA A   ACIAS
02353 2F44 7E2E48            JMP     KEYBD    ;SWITCH OFF, SERVICE KEYBOARD
02354 2F47 B68400     INT1   LDA A   DDRA     ;CLEAR KEYBD FLAG
02355 2F4A B63408            LDA A   ACIAS    ;CHECK ON STATUS OF SERIAL INTERFACE
02356 2F4D 2A17              BPL     INT3     ;NOT SERIAL INTERFACE, BAD INTERRUPT
02357 2F4F F68404            LDA B   DDRA2    ;CHECK SWITCH
02358 2F52 2A12              BPL     INT3
02359 2F54 47                ASR A            ;YES SERIAL INTERFACE, RCVR OR TXMIT ?
02360 2F55 2403              BCC     INT2     ;NOT RCVR, MAYBE TXMIT
02361 2F57 BD2EC4            JSR     RCVR     ;SERVICE RCVR, COME BACK
02362 2F5A B63408     INT2   LDA A   ACIAS    ;CHECK FOR TXMIT
02363 2F5D 47                ASR A
02364 2F5E 47                ASR A
02365 2F5F 2405              BCC     INT3     ;NOT TXMIT, BAD INTERRUPT
02366 2F61 BD2F19            JSR     TXMIT    ;SERVICE TXMIT
02367 2F64 2003              BRA     INT4
02368 2F66 B63409     INT3   LDA A   ACIAD    ;CLEAR ACIA INTERRUPT
02369 2F69 3B         INT4   RTI
02370                 ;
02371                 ;      BEGIN TABLE FOR KEYBOARD
02372                 ;
02373 2F6A 777B7D7E   KTABLE BYTE     077H,07BH,07DH,07EH,070H,0B7H,0BBH,0BDH,0BEH,0B0H
02373 2F6E 70B7BBBD
02373 2F72 BEB0
02374 2F74 D7DBDDDE          BYTE     0D7H,0DBH,0DDH,0DEH,0D0H,0E7H,0EBH,0EDH,0EEH,0E0H
02374 2F78 D0E7EBED
02374 2F7C EEE0
02375 2F7E 1416 0E10         BYTE     014H,016H,00EH,010H,012H,018H,01AH,008H,00AH,00CH
02375 2F82 12181A08
02375 2F86 0A0C
02376 2F88 1C1E0204          BYTE     01CH,01EH,002H,004H,006H,020H,022H,000H,024H,026H
02376 2F8C 06202200
02376 2F90 2426
02377                 ;
```

```
02378                     :       ASCII COMMAND TABLE
02379                     :
02380 2F92 30313233 CHARTB ASCII   "0123456789"
02380 2F96 34353637
02380 2F9A 3839
02381 2F9C 44554C52        ASCII   "DULRAEPS.-"
02381 2FA0 41455053
02381 2FA4 2E2D
02382                     :
02383                     :       BEGIN TRIG. TABLE FOR SINE AND COSINE VALUES
02384                     :       THESE VALUES ARE BETWEEN 0-90 DEGREES.
02385                     :       (MODIFICATION 1.1)
02386                     :
02387 2FA6 00990199 TRIGTB BYTE    00H,99H, 01H,99H, 03H,99H, 05H,99H, 07H,99H
02387 2FAA 03990599
02387 2FAE 0799
02388 2FB0 08991099        BYTE    08H,99H, 10H,99H, 12H,99H, 13H,99H, 15H,98H
02388 2FB4 12991399
02388 2FB8 1598
02389 2FBA 17981998        BYTE    17H,98H, 19H,98H, 20H,97H, 22H,97H, 24H,97H
02389 2FBE 20972297
02389 2FC2 2497
02390 2FC4 25962796        BYTE    25H,96H, 27H,96H, 29H,95H, 30H,95H, 32H,94H
02390 2FC8 29953095
02390 2FCC 3294
02391 2FCE 34943593        BYTE    34H,94H, 35H,93H, 37H,92H, 39H,92H, 40H,91H
02391 2FD2 37923992
02391 2FD6 4091
02392 2FD8 42904389        BYTE    42H,90H, 43H,89H, 45H,89H, 46H,88H, 48H,87H
02392 2FDC 45894688
02392 2FE0 4887
02393 2FE2 50865185        BYTE    50H,86H, 51H,85H, 53H,84H, 54H,83H, 55H,82H
02393 2FE6 53845483
02393 2FEA 5582
02394 2FEC 57815880        BYTE    57H,81H, 58H,80H, 60H,79H, 61H,78H, 62H,77H
02394 2FF0 60796178
02394 2FF4 6277
02395 2FF6 64766575        BYTE    64H,76H, 65H,75H, 66H,74H, 68H,73H, 69H,71H
02395 2FFA 66746873
02395 2FFE 6971
02396 3000 70707169        BYTE    70H,70H, 71H,69H, 73H,68H, 74H,66H, 75H,65H
02396 3004 73687466
02396 3008 7565
02397 300A 76647762        BYTE    76H,64H, 77H,62H, 78H,61H, 79H,60H, 80H,58H
02397 300E 78617960
02397 3012 8058
02398 3014 81578255        BYTE    81H,57H, 82H,55H, 83H,54H, 84H,53H, 85H,51H
02398 3018 83548453
02398 301C 8551
02399 301E 86508748        BYTE    86H,50H, 87H,48H, 88H,46H, 89H,45H, 89H,43H
02399 3022 88468945
02399 3026 8943
02400 3028 90429140        BYTE    90H,42H, 91H,40H, 92H,39H, 92H,37H, 93H,35H
```

```
02400 302C 92399237
02400 3030 9335
02401 3032 94349432          BYTE    94H,34H, 94H,32H, 95H,30H, 95H,29H, 96H,27H
02401 3036 95309529
02401 303A 9627
02402 303C 96259724          BYTE    96H,25H, 97H,24H, 97H,22H, 97H,20H, 98H,19H
02402 3040 97229720
02402 3044 9819
02403 3046 98179815          BYTE    98H,17H, 98H,15H, 99H,13H, 99H,12H, 99H,10H
02403 304A 99139912
02403 304E 9910
02404 3050 99089907          BYTE    99H,08H, 99H,07H, 99H,05H, 99H,03H, 99H,01H
02404 3054 99059903
02404 3058 9901
02405 305A 9900              BYTE    99H,00H
02406                    ;
02407                    ;   BEGIN TABLE FOR MESSAGES
02408                    ;
02409 305C 454E5452 MSG1     ASCII     "ENTR ELEVATION ANGLE"
02409 3060 20454C45
02409 3064 56415449
02409 3068 4F4E2041
02409 306C 4E474C45
02410 3070 454E5445 MSG2     ASCII     "ENTER AZIMUTH ANGLE-"
02410 3074 5220415A
02410 3078 494D5554
02410 307C 4820414E
02410 3080 474C452D
02411 3084 20202020 MSG3     ASCII     "                    "
02411 3088 20202020
02411 308C 20202020
02411 3090 20202020
02411 3094 20202020
02412 3098 4552524F MSG4     ASCII     "ERROR..INVALID ENTRY"
02412 309C 522E2E49
02412 30A0 4E56414C
02412 30A4 49442045
02412 30A8 4E545259
02413 30AC 5241444F MSG5     ASCII     "RADOME POS. READY..."
02413 30B0 4D452050
02413 30B4 4F532E20
02413 30B8 52454144
02413 30BC 592E2E2E
02414 30C0 415A494D MSG6     ASCII     "AZIMUTH             "
02414 30C4 55544820
02414 30C8 20202020
02414 30CC 20202020
02414 30D0 20202020
02415 30D4 454C4556 MSG7     ASCII     "ELEVATION           "
02415 30D8 4154494F
02415 30DC 4E202020
02415 30E0 20202020
02415 30E4 20202020
```

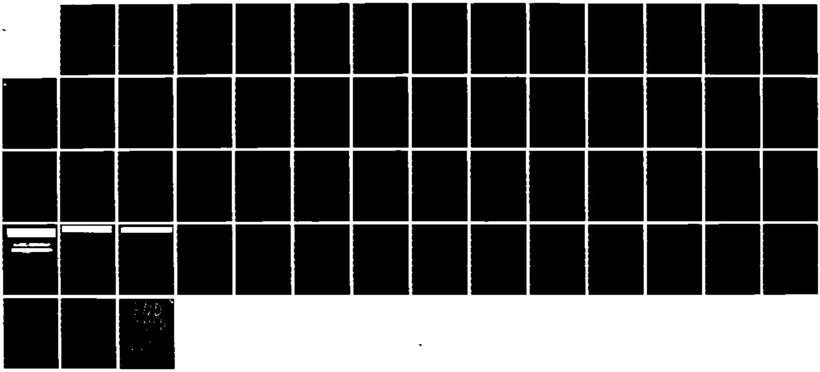
```
02416 30E8 414E474C MSG8   ASCII   "ANGLE TOO LARGE....."
02416 30EC 4520544F
02416 30F0 4F204C41
02416 30F4 5247452E
02416 30F8 2E2E2E2E
02417 30FC 454E5445 MSG9   ASCII   "ENTER PROGRAM NUMBER"
02417 3100 52205052
02417 3104 4F475241
02417 3108 4D204E55
02417 310C 4D424552
02418 3110 50524F47 MSG10  ASCII   "PROG  @DENTER          "
02418 3114 20203A45
02418 3118 4E544552
02418 311C 20202020
02418 3120 20202020
02419 3124 504F5349 MSG11  ASCII   "POSITIONER HALTED    "
02419 3128 54494F4E
02419 312C 45522048
02419 3130 414C5445
02419 3134 44202020
02420 3138 54484520 MSG12  ASCII   "THE GA. TECH-RFSS    "
02420 313C 47412E20
02420 3140 54454348
02420 3144 2D524653
02420 3148 53202020
02421 314C 5241444F MSG13  ASCII   "RADOME POSITIONER    "
02421 3150 4D452050
02421 3154 4F534954
02421 3158 494F4E45
02421 315C 52202020
02422 3160 20202020 MSG14  ASCII   "      VERSION 1.1     "
02422 3164 20564552
02422 3168 53494F4E
02422 316C 20312E31
02422 3170 20202020
02423 3174 414E474C MSG15  ASCII   "ANGLE LIMIT EXCEEDED"
02423 3178 45204C49
02423 317C 4D495420
02423 3180 45584345
02423 3184 45444544
02424 3188 4E454720 MSG16  ASCII   "NEG EL LIMIT         "
02424 318C 454C204C
02424 3190 494D4954
02424 3194 20202020
02424 3198 20202020
02425 319C 504F5320 MSG17  ASCII   "POS EL LIMIT         "
02425 31A0 454C204C
02425 31A4 494D4954
02425 31A8 20202020
02425 31AC 20202020
02426 31B0 4E454720 MSG18  ASCII   "NEG AZ LIMIT         "
02426 31B4 415A204C
02426 31B8 494D4954
```

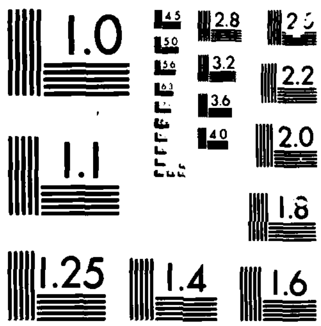
```
31BC 20202020
31C0 20202020
31C4 504F5320 MSG19  ASCII    "POS AZ LIMIT        "
31C8 415A204C
31CC 494D4954
31D0 20202020
31D4 20202020
31D8 56414C49 MSG20  ASCII    "VALID COMMAND RCVED "
31DC 4420434F
31E0 4D4D414E
31E4 44205243
31E8 56454420
                   ;
                   :      BEGIN ROM CONSTANTS
                   ;
31EC 2710     K10K   WORD     10000
31EE 03E8            WORD     1000
31F0 0064            WORD     100
31F2 000A            WORD     10
31F4 0001            WORD     1
31F6 4010     LIMIT  WORD     04010H
31F8 3A40     DIVISO WORD     03A40H
31FA 2D       MINUS  BYTE     02DH
31FB 2B       PLUS   BYTE     02BH
31FC 20       BLANK  BYTE     020H
31FD 27       DEGMAR BYTE     027H
31FE 2E       POINT  BYTE     02EH
                   ;
                   ;      BEGIN STATE TABLE ADDRESSES
                   ;
31FF 2282     SP0    WORD     ST5      ;KEY 0
3201 2282            WORD     ST5      ;KEY 1
3203 2282            WORD     ST5      ;KEY 2
3205 2282            WORD     ST5      ;KEY 3
3207 2282            WORD     ST5      ;KEY 4
3209 2282            WORD     ST5      ;KEY 5
320B 2282            WORD     ST5      ;KEY 6
320D 2282            WORD     ST5      ;KEY 7
320F 2282            WORD     ST5      ;KEY 8
3211 2282            WORD     ST5      ;KEY 9
3213 21D6            WORD     ST1      ;KEY DOWN
3215 2201            WORD     ST2      ;KEY UP
3217 222C            WORD     ST3      ;KEY LEFT
3219 2257            WORD     ST4      ;KEY RIGHT
321B 22A2            WORD     ST10     ;KEY SET AZ
321D 22CE            WORD     ST11     ;KEY SET EL
321F 243F            WORD     ST20     ;KEY PROG
3221 23F9            WORD     ST19     ;KEY START/STOP
3223 2282            WORD     ST5      ;KEY DECIMAL POINT
3225 2282            WORD     ST5      ;KEY MINUS SIGN
3227 22FA     SP10   WORD     ST12     ;0
3229 22FA            WORD     ST12     ;1
322B 22FA            WORD     ST12     ;2
```

```
322D 22FA            WORD    ST12      ;3
322F 22FA            WORD    ST12      ;4
3231 22FA            WORD    ST12      ;5
3233 22FA            WORD    ST12      ;6
3235 22FA            WORD    ST12      ;7
3237 22FA            WORD    ST12      ;8
3239 22FA            WORD    ST12      ;9
323B 2282            WORD    ST5       ;DOWN
323D 2282            WORD    ST5       ;UP
323F 285C            WORD    ST32       ;LEFT
3241 288F            WORD    ST33       ;RIGHT
3243 22A2            WORD    ST10      ;SET AZ
3245 22CE            WORD    ST11      ;SET EL
3247 243F            WORD    ST20      ;PROG
3249 20B7            WORD    ST0       ;START/STOP
324B 2382            WORD    ST16      ;DECIMAL POINT
324D 2322            WORD    ST13      ;MINUS SIGN
324F 22FA      SP11  WORD    ST12      ;0
3251 22FA            WORD    ST12      ;1
3253 22FA            WORD    ST12      ;2
3255 22FA            WORD    ST12      ;3
3257 22FA            WORD    ST12      ;4
3259 22FA            WORD    ST12      ;5
325B 22FA            WORD    ST12      ;6
325D 22FA            WORD    ST12      ;7
325F 22FA            WORD    ST12      ;8
3261 22FA            WORD    ST12      ;9
3263 27F6            WORD    ST30       ;DOWN
3265 2829            WORD    ST31       ;UP
3267 2282            WORD    ST5       ;LEFT
3269 2282            WORD    ST5       ;RIGHT
326B 22A2            WORD    ST10      ;SET AZ
326D 22CE            WORD    ST11      ;SET EL
326F 243F            WORD    ST20      ;PROG
3271 20B7            WORD    ST0       ;START/STOP
3273 2382            WORD    ST16      ;DECIMAL POINT
3275 2322            WORD    ST13      ;MINUS SIGN
3277 2361      SP12  WORD    ST15      ;0
3279 2361            WORD    ST15      ;1
327B 2361            WORD    ST15      ;2
327D 2361            WORD    ST15      ;3
327F 2361            WORD    ST15      ;4
3281 2361            WORD    ST15      ;5
3283 2361            WORD    ST15      ;6
3285 2361            WORD    ST15      ;7
3287 2361            WORD    ST15      ;8
3289 2361            WORD    ST15      ;9
328B 2282            WORD    ST5       ;DOWN
328D 2282            WORD    ST5       ;UP
328F 2282            WORD    ST5       ;LEFT
3291 2282            WORD    ST5       ;RIGHT
3293 22A2            WORD    ST10      ;SET AZ
3295 22CE            WORD    ST11      ;SET EL
```

AD-A166 821 RADOME POSITIONER FOR THE RFSS (RADIO FREQUENCY SIMULATION SYSTEM)(U) GEORGIA INST OF TECH ATLANTA ENGINEERING EXPERIMENT STATION D O GALLENTINE ET AL. 27 FEB 78 DAAK40-77-C-0047 3/3

UNCLASSIFIED F/G 17/9 NL





MICROCOPY CHART

```
3297 243F          WORD   ST20   ;PROG
3299 20B7          WORD   ST0    ;START/STOP
329B 2382          WORD   ST16   ;DECIMAL POINT
329D 2282          WORD   ST5    ;MINUS SIGN
329F 2340   SP13   WORD   ST14   ;0
32A1 2340          WORD   ST14   ;1
32A3 2340          WORD   ST14   ;2
32A5 2340          WORD   ST14   ;3
32A7 2340          WORD   ST14   ;4
32A9 2340          WORD   ST14   ;5
32AB 2340          WORD   ST14   ;6
32AD 2340          WORD   ST14   ;7
32AF 2340          WORD   ST14   ;8
32B1 2340          WORD   ST14   ;9
32B3 2282          WORD   ST5    ;DOWN
32B5 2282          WORD   ST5    ;UP
32B7 2282          WORD   ST5    ;LEFT
32B9 2282          WORD   ST5    ;RIGHT
32BB 22A2          WORD   ST10   ;SET AZ
32BD 22CE          WORD   ST11   ;SET EL
32BF 243F          WORD   ST20   ;PROG
32C1 20B7          WORD   ST0    ;START/STOP
32C3 2382          WORD   ST16   ;DECIMAL POINT
32C5 2282          WORD   ST5    ;MINUS SIGN
32C7 2361   SP14   WORD   ST15   ;0
32C9 2361          WORD   ST15   ;1
32CB 2361          WORD   ST15   ;2
32CD 2361          WORD   ST15   ;3
32CF 2361          WORD   ST15   ;4
32D1 2361          WORD   ST15   ;5
32D3 2361          WORD   ST15   ;6
32D5 2361          WORD   ST15   ;7
32D7 2361          WORD   ST15   ;8
32D9 2361          WORD   ST15   ;9
32DB 2282          WORD   ST5    ;DOWN
32DD 2282          WORD   ST5    ;UP
32DF 2282          WORD   ST5    ;LEFT
32E1 2282          WORD   ST5    ;RIGHT
32E3 22A2          WORD   ST10   ;SET AZ
32E5 22CE          WORD   ST11   ;SET EL
32E7 243F          WORD   ST20   ;PROG
32E9 20B7          WORD   ST0    ;START/STOP
32EB 2382          WORD   ST16   ;DECIMAL POINT
32ED 2282          WORD   ST5    ;MINUS SIGN
32EF 2290   SP15   WORD   ST6    ;0
32F1 2290          WORD   ST6    ;1
32F3 2290          WORD   ST6    ;2
32F5 2290          WORD   ST6    ;3
32F7 2290          WORD   ST6    ;4
32F9 2290          WORD   ST6    ;5
32FB 2290          WORD   ST6    ;6
32FD 2290          WORD   ST6    ;7
32FF 2290          WORD   ST6    ;8
```

```
02576 3301 2290            WORD    ST6     ;9
02577 3303 2282            WORD    ST5     ;DOWN
02578 3305 2282            WORD    ST5     ;UP
02579 3307 2282            WORD    ST5     ;LEFT
02580 3309 2282            WORD    ST5     ;RIGHT
02581 330B 22A2            WORD    ST10    ;SET AZ
02582 330D 22CE            WORD    ST11    ;SET EL
02583 330F 243F            WORD    ST20    ;PROG
02584 3311 20B7            WORD    ST0     ;START/STOP
02585 3313 2382            WORD    ST16    ;DECIMAL POINT
02586 3315 2282            WORD    ST5     ;MINUS SIGN
02587 3317 239E    SP16    WORD    ST17    ;0
02588 3319 239E            WORD    ST17    ;1
02589 331B 239E            WORD    ST17    ;2
02590 331D 239E            WORD    ST17    ;3
02591 331F 239E            WORD    ST17    ;4
02592 3321 239E            WORD    ST17    ;5
02593 3323 239E            WORD    ST17    ;6
02594 3325 239E            WORD    ST17    ;7
02595 3327 239E            WORD    ST17    ;8
02596 3329 239E            WORD    ST17    ;9
02597 332B 2282            WORD    ST5     ;DOWN
02598 332D 2282            WORD    ST5     ;UP
02599 332F 2282            WORD    ST5     ;LEFT
02600 3331 2282            WORD    ST5     ;RIGHT
02601 3333 22A2            WORD    ST10    ;SET AZ
02602 3335 22CE            WORD    ST11    ;SET EL
02603 3337 243F            WORD    ST20    ;PROG
02604 3339 20B7            WORD    ST0     ;START/STOP
02605 333B 2282            WORD    ST5     ;DECIMAL POINT
02606 333D 2282            WORD    ST5     ;MINUS SIGN
02607 333F 2282    SP17    WORD    ST5     ;0
02608 3341 2282            WORD    ST5     ;1
02609 3343 2282            WORD    ST5     ;2
02610 3345 2282            WORD    ST5     ;3
02611 3347 2282            WORD    ST5     ;4
02612 3349 2282            WORD    ST5     ;5
02613 334B 2282            WORD    ST5     ;6
02614 334D 2282            WORD    ST5     ;7
02615 334F 2282            WORD    ST5     ;8
02616 3351 2282            WORD    ST5     ;9
02617 3353 2282            WORD    ST5     ;DOWN
02618 3355 2282            WORD    ST5     ;UP
02619 3357 2282            WORD    ST5     ;LEFT
02620 3359 2282            WORD    ST5     ;RIGHT
02621 335B 22A2            WORD    ST10    ;SET AZ
02622 335D 22CE            WORD    ST11    ;SET EL
02623 335F 243F            WORD    ST20    ;PROG
02624 3361 23ED            WORD    ST18    ;START/STOP
02625 3363 2282            WORD    ST5     ;DECIMAL POINT
02626 3365 2282            WORD    ST5     ;MINUS SIGN
02627 3367 2282    SP20    WORD    ST5     ;0
02628 3369 2469            WORD    ST21    ;1
```

```
02629 336B 2469          WORD   ST21    ;2
02630 336D 2469          WORD   ST21    ;3
02631 336F 2469          WORD   ST21    ;4
02632 3371 2282          WORD   ST5     ;5
02633 3373 2282          WORD   ST5     ;6
02634 3375 2282          WORD   ST5     ;7
02635 3377 2282          WORD   ST5     ;8
02636 3379 2282          WORD   ST5     ;9
02637 337B 2282          WORD   ST5     ;DOWN
02638 337D 2282          WORD   ST5     ;UP
02639 337F 2282          WORD   ST5     ;LEFT
02640 3381 2282          WORD   ST5     ;RIGHT
02641 3383 22A2          WORD   ST10    ;SET AZ
02642 3385 22CE          WORD   ST11    ;SET EL
02643 3387 243F          WORD   ST20    ;PROG
02644 3389 20B7          WORD   ST0     ;START/STOP
02645 338B 2282          WORD   ST5     ;DECIMAL POINT
02646 338D 2282          WORD   ST5     ;MINUS SIGN
02647 338F 2489    SP21  WORD   ST22    ;0
02648 3391 2489          WORD   ST22    ;1
02649 3393 2489          WORD   ST22    ;2
02650 3395 2489          WORD   ST22    ;3
02651 3397 2489          WORD   ST22    ;4
02652 3399 2489          WORD   ST22    ;5
02653 339B 2489          WORD   ST22    ;6
02654 339D 2489          WORD   ST22    ;7
02655 339F 2489          WORD   ST22    ;8
02656 33A1 2489          WORD   ST22    ;9
02657 33A3 2282          WORD   ST5     ;DOWN
02658 33A5 2282          WORD   ST5     ;UP
02659 33A7 2282          WORD   ST5     ;LEFT
02660 33A9 2282          WORD   ST5     ;RIGHT
02661 33AB 22A2          WORD   ST10    ;SET AZ
02662 33AD 22CE          WORD   ST11    ;SET EL
02663 33AF 243F          WORD   ST20    ;PROG
02664 33B1 20B7          WORD   ST0     ;START/STOP
02665 33B3 24CB          WORD   ST24    ;DECIMAL POINT
02666 33B5 2282          WORD   ST5     ;MINUS SIGN
02667 33B7 24AA    SP22  WORD   ST23    ;0
02668 33B9 24AA          WORD   ST23    ;1
02669 33BB 24AA          WORD   ST23    ;2
02670 33BD 24AA          WORD   ST23    ;3
02671 33BF 24AA          WORD   ST23    ;4
02672 33C1 24AA          WORD   ST23    ;5
02673 33C3 24AA          WORD   ST23    ;6
02674 33C5 24AA          WORD   ST23    ;7
02675 33C7 24AA          WORD   ST23    ;8
02676 33C9 24AA          WORD   ST23    ;9
02677 33CB 2282          WORD   ST5     ;DOWN
02678 33CD 2282          WORD   ST5     ;UP
02679 33CF 2282          WORD   ST5     ;LEFT
02680 33D1 2282          WORD   ST5     ;RIGHT
02681 33D3 22A2          WORD   ST10    ;SET AZ
```

```
33D5 22CE         WORD  ST11   ;SET EL
33D7 243F         WORD  ST20   ;PROG
33D9 20B7         WORD  ST0    ;START/STOP
33DB 24CB         WORD  ST24   ;DECIMAL POINT
33DD 2282         WORD  ST5    ;MINUS SIGN
33DF 2290   SP23  WORD  ST6    ;0
33E1 2290         WORD  ST6    ;1
33E3 2290         WORD  ST6    ;2
33E5 2290         WORD  ST6    ;3
33E7 2290         WORD  ST6    ;4
33E9 2290         WORD  ST6    ;5
33EB 2290         WORD  ST6    ;6
33ED 2290         WORD  ST6    ;7
33EF 2290         WORD  ST6    ;8
33F1 2290         WORD  ST6    ;9
33F3 2282         WORD  ST5    ;DOWN
33F5 2282         WORD  ST5    ;UP
33F7 2282         WORD  ST5    ;LEFT
33F9 2282         WORD  ST5    ;RIGHT
33FB 22A2         WORD  ST10   ;SET AZ
33FD 22CE         WORD  ST11   ;SET EL
33FF 243F         WORD  ST20   ;PROG
3401 20B7         WORD  ST0    ;START/STOP
3403 24CB         WORD  ST24   ;DECIMAL POINT
3405 2282         WORD  ST5    ;MINUS SIGN
3407 24E7   SP24  WORD  ST25   ;0
3409 24E7         WORD  ST25   ;1
340B 24E7         WORD  ST25   ;2
340D 24E7         WORD  ST25   ;3
340F 24E7         WORD  ST25   ;4
3411 24E7         WORD  ST25   ;5
3413 24E7         WORD  ST25   ;6
3415 24E7         WORD  ST25   ;7
3417 24E7         WORD  ST25   ;8
3419 24E7         WORD  ST25   ;9
341B 2282         WORD  ST5    ;DOWN
341D 2282         WORD  ST5    ;UP
341F 2282         WORD  ST5    ;LEFT
3421 2282         WORD  ST5    ;RIGHT
3423 22A2         WORD  ST10   ;SET AZ
3425 22CE         WORD  ST11   ;SET EL
3427 243F         WORD  ST20   ;PROG
3429 20B7         WORD  ST0    ;START/STOP
342B 2282         WORD  ST5    ;DECIMAL POINT
342D 2282         WORD  ST5    ;MINUS SIGN
342F 2282   SP25  WORD  ST5    ;0
3431 2282         WORD  ST5    ;1
3433 2282         WORD  ST5    ;2
3435 2282         WORD  ST5    ;3
3437 2282         WORD  ST5    ;4
3439 2282         WORD  ST5    ;5
343B 2282         WORD  ST5    ;6
343D 2282         WORD  ST5    ;7
```

```
343F 2282           WORD    ST5     ;8
3441 2282           WORD    ST5     ;9
3443 2282           WORD    ST5     ;DOWN
3445 2282           WORD    ST5     ;UP
3447 2282           WORD    ST5     ;LEFT
3449 2282           WORD    ST5     ;RIGHT
344B 22A2           WORD    ST10    ;SET AZ
344D 22CE           WORD    ST11    ;SET EL
344F 243F           WORD    ST20    ;PROG
3451 2578           WORD    ST00    ;START/STOP
3453 2282           WORD    ST5     ;DECIMAL POINT
3455 2282           WORD    ST5     ;MINUS SIGN
3457 28C2      SP30 WORD    ST34    ;0
3459 28C2           WORD    ST34    ;1
345B 28C2           WORD    ST34    ;2
345D 28C2           WORD    ST34    ;3
345F 28C2           WORD    ST34    ;4
3461 28C2           WORD    ST34    ;5
3463 28C2           WORD    ST34    ;6
3465 28C2           WORD    ST34    ;7
3467 28C2           WORD    ST34    ;8
3469 28C2           WORD    ST34    ;9
346B 27F6           WORD    ST30    ;DOWN
346D 2829           WORD    ST31    ;UP
346F 285C           WORD    ST32    ;LEFT
3471 288F           WORD    ST33    ;RIGHT
3473 22A2           WORD    ST10    ;SET AZ
3475 22CE           WORD    ST11    ;SET EL
3477 243F           WORD    ST20    ;PROG
3479 20B7           WORD    ST0     ;START/STOP
347B 2904           WORD    ST36    ;DECIMAL POINT
347D 2282           WORD    ST5     ;MINUS SIGN
347F 28C2      SP31 WORD    ST34    ;0
3481 28C2           WORD    ST34    ;1
3483 28C2           WORD    ST34    ;2
3485 28C2           WORD    ST34    ;3
3487 28C2           WORD    ST34    ;4
3489 28C2           WORD    ST34    ;5
348B 28C2           WORD    ST34    ;6
348D 28C2           WORD    ST34    ;7
348F 28C2           WORD    ST34    ;8
3491 28C2           WORD    ST34    ;9
3493 27F6           WORD    ST30    ;DOWN
3495 2829           WORD    ST31    ;UP
3497 285C           WORD    ST32    ;LEFT
3499 288F           WORD    ST33    ;RIGHT
349B 22A2           WORD    ST10    ;SET AZ
349D 22CE           WORD    ST11    ;SET EL
349F 243F           WORD    ST20    ;PROG
34A1 20B7           WORD    ST0     ;START/STOP
34A3 2904           WORD    ST36    ;DECIMAL POINT
34A5 2282           WORD    ST5     ;MINUS SIGN
34A7 28C2      SP32 WORD    ST34    ;0
```

```
34A9 28C2          WORD   ST34    ;1
34AB 28C2          WORD   ST34    ;2
34AD 28C2          WORD   ST34    ;3
34AF 28C2          WORD   ST34    ;4
34B1 28C2          WORD   ST34    ;5
34B3 28C2          WORD   ST34    ;6
34B5 28C2          WORD   ST34    ;7
34B7 28C2          WORD   ST34    ;8
34B9 28C2          WORD   ST34    ;9
34BB 27F6          WORD   ST30    ;DOWN
34BD 2829          WORD   ST31    ;UP
34BF 285C          WORD   ST32    ;LEFT
34C1 288F          WORD   ST33    ;RIGHT
34C3 22A2          WORD   ST10    ;SET AZ
34C5 22CE          WORD   ST11    ;SET EL
34C7 243F          WORD   ST20    ;PROG
34C9 20B7          WORD   ST0     ;START/STOP
34CB 2904          WORD   ST36    ;DECIMAL
34CD 2282          WORD   ST5     ;MINUS SIGN
34CF 28C2    SP33  WORD   ST34    ;0
34D1 28C2          WORD   ST34    ;1
34D3 28C2          WORD   ST34    ;2
34D5 28C2          WORD   ST34    ;3
34D7 28C2          WORD   ST34    ;4
34D9 28C2          WORD   ST34    ;5
34DB 28C2          WORD   ST34    ;6
34DD 28C2          WORD   ST34    ;7
34DF 28C2          WORD   ST34    ;8
34E1 28C2          WORD   ST34    ;9
34E3 27F6          WORD   ST30    ;DOWN
34E5 2829          WORD   ST31    ;UP
34E7 285C          WORD   ST32    ;LEFT
34E9 288F          WORD   ST33    ;RIGHT
34EB 22A2          WORD   ST10    ;SET AZ
34ED 22CE          WORD   ST11    ;SET EL
34EF 243F          WORD   ST20    ;PROG
34F1 20B7          WORD   ST0     ;START/STOP
34F3 2904          WORD   ST36    ;DECIMAL POINT
34F5 2282          WORD   ST5     ;MINUS SIGN
34F7 28E3    SP34  WORD   ST35    ;0
34F9 28E3          WORD   ST35    ;1
34FB 28E3          WORD   ST35    ;2
34FD 28E3          WORD   ST35    ;3
34FF 28E3          WORD   ST35    ;4
3501 28E3          WORD   ST35    ;5
3503 28E3          WORD   ST35    ;6
3505 28E3          WORD   ST35    ;7
3507 28E3          WORD   ST35    ;8
3509 28E3          WORD   ST35    ;9
350B 2282          WORD   ST5     ;DOWN
350D 2282          WORD   ST5     ;UP
350F 2282          WORD   ST5     ;LEFT
3511 2282          WORD   ST5     ;RIGHT
```

```
3513 22A2           WORD   ST10     ;SET AZ
3515 22CE           WORD   ST11     ;SET EL
3517 243F           WORD   ST20     ;PROG
3519 20B7           WORD   ST0      ;START/STOP
351B 2904           WORD   ST36     ;DECIMAL POINT
351D 2282           WORD   ST5      ;MINUS SIGN
351F 2290    SP35   WORD   ST6      ;0
3521 2290           WORD   ST6      ;1
3523 2290           WORD   ST6      ;2
3525 2290           WORD   ST6      ;3
3527 2290           WORD   ST6      ;4
3529 2290           WORD   ST6      ;5
352B 2290           WORD   ST6      ;6
352D 2290           WORD   ST6      ;7
352F 2290           WORD   ST6      ;8
3531 2290           WORD   ST6      ;9
3533 2282           WORD   ST5      ;DOWN
3535 2282           WORD   ST5      ;UP
3537 2282           WORD   ST5      ;LEFT
3539 2282           WORD   ST5      ;RIGHT
353B 22A2           WORD   ST10     ;SET AZ
353D 22CE           WORD   ST11     ;SET EL
353F 243F           WORD   ST20     ;PROG
3541 20B7           WORD   ST0      ;START/STOP
3543 2904           WORD   ST36     ;DECIMAL POINT
3545 2282           WORD   ST5      ;MINUS SIGN
3547 2920    SP36   WORD   ST37     ;0
3549 2920           WORD   ST37     ;1
354B 2920           WORD   ST37     ;2
354D 2920           WORD   ST37     ;3
354F 2920           WORD   ST37     ;4
3551 2920           WORD   ST37     ;5
3553 2920           WORD   ST37     ;6
3555 2920           WORD   ST37     ;7
3557 2920           WORD   ST37     ;8
3559 2920           WORD   ST37     ;9
355B 2282           WORD   ST5      ;DOWN
355D 2282           WORD   ST5      ;UP
355F 2282           WORD   ST5      ;LEFT
3561 2282           WORD   ST5      ;RIGHT
3563 22A2           WORD   ST10     ;SET AZ
3565 22CE           WORD   ST11     ;SET EL
3567 243F           WORD   ST20     ;PROG
3569 20B7           WORD   ST0      ;START/STOP
356B 2282           WORD   ST5      ;DECIMAL POINT
356D 2282           WORD   ST5      ;MINUS SIGN
             ;
             ;      BEGIN RESET INTERRUPT VECTORS
             ;
     FFF8           ORG    0FFF8H
FFF8 2F02           WORD   INT      ;IRQ VECTOR
FFFA 2000           WORD   GOE      ;SWI VECTOR JUST IN CASE
FFFC 2E8E           WORD   DISPL    ;NMI VECTOR JUST IN CASE
```

```
02894 FFFE 2000           WORD     GOE      ;RESET VECTOR
***** P
02895                     END
```

```
ACIAD -- 8409     ACIAS -- 8408     ADDCAL - 2BE8     ALSTOP - 2ABB     ANGLE -- 003F
ASC1 --- 2D3F     ASC2 --- 2D59     ASC3 --- 2D58     ASCDIS - 2C22     AZBCD -- 0045
AZEL --- 0061     AZKEY -- 0053     AZKEYS - 0057     AZMAG -- 005D     AZSIGN - 0043
B2 ----- 213B     B3 ----- 214A     B4 ----- 2155     B5 ----- 21BC     B6 ----- 21C9
B7 ----- 21D3     B8 ----- 21D0     BACK --- 2AF7     BCD151 - 29A9     BCD153 - 29B4
BCD163 - 29BE     BCD165 - 29D8     BCD167 - 29D9     BCDA --- 003C     BCDB --- 003D
BCDBIN - 29EB     BCDDIS - 2D2F     BCDDIV - 2999     BCDMPY - 296D     BCDSUB - 2AC4
BCDVSR - 0072     BINANG - 0081     BINBCD - 2C3D     BINUPR - 008A     BLANK -- 31FC
CARRY -- 0098     CHARBF - 002B     CHARCT - 002E     CHARPT - 002C     CHARTB - 2F92
CHRNUM - 0030     COLUM1 - 2E53     COLUMN - 2E50     COSINE - 0035     CP1 ---- 295B
CPFLAG - 294C     CRA ---- 8401     CRA2 --- 8405     CRA3 --- 8801     CRB ---- 8403
CRB2 --- 8407     CRB3 --- 8303     CSIGN -- 0088     CVDEC1 - 2C6A     CVDEC2 - 2C6D
CVDEC5 - 2C78     DDRA --- 8400     DDRA2 -- 8404     DDRA3 -- 8800     DDRB --- 8402
DDRB2 -- 8406     DDRB3 -- 8802     DECOD1 - 2E70     DECODE - 2E6D     DEGMAR - 31FD
DFLAGA - 006C     DFLAGE - 006D     DIFF --- 2DE6     DIFF1 -- 2D8A     DIFFAZ - 2134
DIFFEL - 21B6     DISAZ -- 0000     DISBUF - 0000     DISEL -- 000A     DISPL -- 2E8E
DIV151 - 2CCC     DIV153 - 2CDB     DIV163 - 2CDD     DIV165 - 2D12     DIV167 - 2D13
DIVIDE - 2CAE     DIVISO - 31F8     DOHUND - 29FE     DOTHOU - 2A0F     DOWN --- 2204
ELBCD -- 0047     ELKEY -- 0055     ELKEYS - 0058     ELMAG -- 005F     ELSIGN - 0044
ENTRYA - 004D     ENTRYB - 004E     ETHSPD - 00F0     FIN ---- 25EF     FPTAZ -- 0076
FPTAZS - 0079     FPTEL -- 0074     FPTELS - 0078     GOE ---- 2000     GTCHAR - 2E7D
HAFSPD - 00D0     HUNLP -- 2A03     HUNOO -- 2A0C     INT ---- 2F32     INT1 --- 2F47
INT2 --- 2F5A     INT3 --- 2F66     INT4 --- 2F69     K1 ----- 2E85     K10K --- 31EC
KEYBD -- 2E48     KEYC --- 0052     KEYENT - 004F     KFLAG -- 006B     KTABLE - 2F6A
LEFT --- 2C0E     LETA --- 003A     LETB --- 003B     LFLAGA - 0094     LFLAGE - 0093
LIMIT -- 31F6     LSBENC - 0039     LSBSAZ - 8E00     LSBSEL - 8E02     MCOM1 -- 2EEC
MCOMND - 2EDF     MFLAG -- 0059     MINUEN - 005A     MINUS -- 31FA     MOTAZ -- 2D86
MOTEL -- 2DE2     MOVED -- 2943     MPY163 - 297B     MPY167 - 298F     MSBENC - 0038
MSBSAZ - 8E01     MSBSEL - 8E03     MSG1 --- 305C     MSG10 -- 3110     MSG11 -- 3124
MSG12 -- 3138     MSG13 -- 314C     MSG14 -- 3160     MSG15 -- 3174     MSG16 -- 3188
MSG17 -- 319C     MSG18 -- 31B0     MSG19 -- 31C4     MSG2 --- 3070     MSG20 -- 31D8
MSG3 --- 3084     MSG4 --- 3098     MSG5 --- 30AC     MSG6 --- 30C0     MSG7 --- 30D4
MSG8 --- 30E8     MSG9 --- 30FC     MSGA --- 2096     MSGB --- 20B2     MSGDON - 2F2E
MSGFLG - 0095     NAZLIM - 0091     NEGS --- 25ED     NELLIM - 008D     NEXTC -- 2955
NMATCH - 2ED7     NOAZ --- 2142     NOEL --- 21C2     NOPE --- 296C     02 ----- 2DC4
03 ----- 2DC6     04 ----- 2E20     05 ----- 2E22     ONE ---- 2E12     ONE1 --- 2DB6
OVER --- 2C29     PACK --- 2A9E     PAL ---- 2B41     PAL1 --- 2B31     PAZLIM - 008F
PEL ---- 2BC1     PELLIM - 003B     PFLAG -- 007D     PLUS --- 31FB     POINT -- 31FE
POS ---- 2C5E     POSN --- 2C65     PROANG - 007F     PROCNT - 007A     PROGA -- 0063
PROGB -- 0067     PROGC -- 0069     PROGL -- 0064     PROGN -- 0063     QUASPD - 00E0
RCVDEC - 2EC7     RCVEND - 2EDE     RCVNEX - 2ECB     RCVR --- 2EC4     RESET -- 2EAC
RESTO -- 2936     RIGHT -- 2C18     ROW ---- 2E6A     SAME --- 2E38     SAME1 -- 2DDC
SAMEAZ - 213E     SAMEEL - 21BE     SAVDEC - 003E     SAVE1 -- 0089     SAVEA -- 0031
SAVEX -- 0032     SAVEX1 - 0034     SAVEX2 - 0096     SEND --- 2F08     SFLAGA - 006E
SFLAGE - 006F     SHA ---- 2BB1     SHA1 --- 2BCD     SHA2 --- 2B4D     SHA3 --- 2B99
SHA4 --- 2B9C     SHAENC - 2AF8     SIBUF -- 0014     SIGN --- 0042     SINE --- 0083
SP0 ---- 31FF     SP10 --- 3227     SP11 --- 324F     SP12 --- 3277     SP13 --- 329F
SP14 --- 32C7     SP15 --- 32EF     SP16 --- 3317     SP17 --- 333F     SP20 --- 3367
SP21 --- 338F     SP22 --- 33B7     SP23 --- 33DF     SP24 --- 3407     SP25 --- 342F
SP30 --- 3457     SP31 --- 347F     SP32 --- 34A7     SP33 --- 34CF     SP34 --- 34F7
SP35 --- 351F     SP36 --- 3547     SPEEDA - 005B     SPEEDE - 005C     SSIGN -- 0037
ST0 ---- 20B7     ST00 --- 2578     ST001 -- 2589     ST002 -- 258C     ST003 -- 258F
ST0A --- 20CA     ST0B --- 211C     ST0C --- 2126     ST0D --- 212E     ST0E --- 2158
```

```
TEKTRONIX       M6800 ASM V2.2    SYMBOL TABLE LISTING         PAGE   60

    ST0F --- 219E      ST0G --- 21A8      ST0H --- 21B0      ST0X --- 20EF      ST0X1 -- 210E
    ST0X2 -- 20FD      ST0Y --- 2170      ST0Y1 -- 217E      ST0Z --- 2190      ST1 ---- 21E6
    ST10 --- 22A2      ST10A -- 22AF      ST11 --- 22CE      ST11A -- 22DB      ST12 --- 22FA
    ST12A -- 230F      ST13 --- 2322      ST13A -- 232D      ST14 --- 2340      ST14A -- 234E
    ST15 --- 2367      ST15A -- 236F      ST16 --- 2382      ST16A -- 238B      ST17 --- 239E
    ST17A -- 23C9      ST17C -- 23DA      ST18 --- 23ED      ST19 --- 23F9      ST19A -- 241B
    ST19B -- 243C      ST1A --- 21DD      ST1B --- 21F1      ST2 ---- 2201      ST20 --- 243F
    ST20A -- 244A      ST21 --- 2469      ST21A -- 2476      ST22 --- 2489      ST22A -- 2497
    ST23 --- 24AA      ST23A -- 24B8      ST24 --- 24CB      ST24A -- 24D4      ST25 --- 24E7
    ST25A -- 2565      ST25B -- 2537      ST25C -- 255D      ST26 --- 2592      ST26A -- 25C4
    ST26A1 - 25D9      ST26B -- 25F7      ST26B1 - 2618      ST26B2 - 2635      ST26B3 - 2641
    ST26C -- 2644      ST26E -- 264F      ST27 --- 2655      ST27A -- 2687      ST27A1 - 269D
    ST27B -- 26B6      ST27B1 - 26D7      ST27B2 - 26F4      ST27B3 - 2700      ST27C -- 2703
    ST27D -- 270E      ST27E -- 2711      ST28 --- 2717      ST28A -- 2738      ST28B -- 274D
    ST28C -- 277F      ST28D -- 279B      ST29 --- 27A1      ST29A -- 27B1      ST29B -- 27F0
    ST2A --- 2208      ST2B --- 221C      ST3 ---- 222C      ST30 --- 27F6      ST30A -- 2810
    ST30B -- 2823      ST31 --- 2829      ST31A -- 2843      ST31B -- 2856      ST32 --- 285C
    ST32A -- 2876      ST32B -- 2889      ST33 --- 288F      ST33A -- 28A9      ST33B -- 28BC
    ST34 --- 28C2      ST34A -- 28D0      ST35 --- 28E3      ST35A -- 28F1      ST36 --- 2904
    ST36A -- 290D      ST37 --- 2920      ST37A -- 292D      ST3A --- 2233      ST3B --- 2247
    ST4 ---- 2257      ST4A --- 225E      ST4B --- 2272      ST5 ---- 2282      ST6 ---- 2290
    ST7 ---- 2299      STADDR - 007B      START -- 2EA1      SUBT --- 2ADB      SWAP --- 2AD1
    TEMPA -- 0036      TEMPA1 - 0050      TEMPB -- 0037      TEMPB1 - 0051      TEMPD -- 0070
    TEMPS -- 0062      TEMPX -- 0029      TEMPX1 - 0049      TEMPX2 - 004B      TENLP -- 29F7
    THOULP - 2A1F      THOU00 - 2A1D      TRIGA -- 2A47      TRIGAD - 2A2A      TRIGB -- 2A60
    TRIGC -- 2A7B      TRIGD -- 2A81      TRIGTB - 2FA6      TSTANG - 295E      TXMEND - 2F31
    TXHIT -- 2F19      UP ----- 2BFA      WAIT --- 2EBB      WAIT1 -- 2EBD      WAITE -- 2E3E
    WAITE1 - 2E41      XITBIN - 2A29      YESAZ -- 214D      YESEL -- 21CB      Z2 ----- 2D9C
    Z3 ----- 2D9E      Z4 ----- 2DF8      Z5 ----- 2DFA      ZERO --- 2DEA      ZERO1 -- 2D8E

 2895 SOURCE LINES      1 ERROR
```

APPENDIX C

COMPONENT OPERATING MANUALS AND DATA SHEETS

Measurement Systems Division
Christina Street
Newton, Mass. 02161

OPERATION AND MAINTENANCE MANUAL

DIGISEC® RA __/23C SERIES ENCODERS

MANUAL NO. 2802
REV. E, JANUARY 1976

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1 SCOPE

This manual is to be used with the DIGISEC® RA __/23C series of absolute shaft position encoders. The manual covers installation, operation, theory of operation, and field maintenance. The discussion has general application inasmuch as design, operation, and maintenance features are similar for all encoders in this series. Refer to Section 6 for identifying nomenclature applicable to all models in this series. Differences in models are also tabulated in Section 6. Maintenance or repair beyond that covered in this manual must be performed by the manufacturer.

1.2 GENERAL DESCRIPTION

Encoders of the DIGISEC RA __/23C series are medium resolution, absolute shaft position encoders of the photoelectric, non-contacting type, which are designed for use wherever shaft position information is required in digital form. Typical applications include digital servos, stable platforms, navigation systems, theodolites, tracking radars, laser tracking systems, and numerical control systems.

An outline drawing applicable to all RA __/23C series encoders is contained in Section 6. All RA __/23C series encoders have a standard Size 23 synchro mount (2.25 inch diameter mounting flange) with 2-inch pilot diameters on both sides of the flange, as shown in the outline drawing. Thus, the encoder can be mounted with the flange located on either side of the mounting surface. The notch in the synchro flange mates with a standard zeroing ring (not supplied) which can be used to precisely align the encoder to the drive shaft zero reference.* RA --/23C series encoders are provided with either a plain or a standard splined 0.25 inch shaft. The drive shaft to be monitored is coupled to the plain shaft through a high accuracy flexible coupling, and to the splined shaft through a standard gear fastened to the latter. The choice of flexible coupling or gear hardware is left to the user.

RA __/23C series encoders are designed to operate from a +5V source (+6V optional). Except for this external source, the encoder is functionally self-contained. Within its cylindrical case are contained a shaft-mounted glass code disc, illuminating lamps, photodetectors, and signal processing solid state circuits, which provide a digital output word representing the instantaneous absolute angular position of the encoder shaft. The output word is in natural binary code and is provided in parallel format, with one bit per output channel. One pigtail cable supplies power to the encoder, brings out the parallel outputs, and provides a test point for the illuminating lamps. One cable lead (HOLD) is also used to apply an external HOLD

*Refer to MIL-HDBK-214A (Synchros) for information on applicable zeroing hardware (zeroing rings, pinion wrenches, etc.)

pulse when reading out "on the fly". To eliminate any possible ambiguity in the parallel readout, an inherent characteristic of the natural binary code, DIGISEC encoders utilize anti-ambiguity logic, which requires a finite amount of settling time for the signal to propagate from the least significant bit to the most significant bit. Application of the HOLD pulse freezes the state of the least significant bit and enables non-ambiguous parallel readout, subsequent to the settling period, for the remainder of the HOLD pulse duration.

All RA __/23C series encoders have field replaceable illuminating lamps[1] and signal processing electronics to facilitate maintenance.

1.3 SPECIFICATIONS

General specifications applicable to all encoders of the RA __/23C series are contained in Table 1-1. Additional detailed specifications showing differences between various models are contained in Section 6. These differences include resolution, power supply voltage, direction of rotation for increasing count, shaft style, and temperature range. Output stages on all encoder channels are either 5404 or 7404 TTL (transistor-transistor logic) elements. 5404 elements are used in encoders with a "military" temperature range. 7404 elements are used in encoders with a "commercial" temperature range. Performance characteristics are similar for both types. Figure 1-1 provides output/load interface information.

1.4 DESIGN FEATURES

The DIGISEC RA __/23C series has been designed to meet the requirements of the most demanding military and industrial applications with emphasis on ruggedness, long life, and reliability. All electronic circuits are solid state and of conservative design with components substantially derated. Noteworthy design features are the following:

a. Standard Size 23 synchro mount.

b. Optional shaft style (plain/splined).

c. One power supply voltage (+5V, +6V optional).

d. Optional temperature ranges (military/commercial).

e. Sealed bearings, field lubrication not required.

f. Hard-chrome-on-glass code disc.

1/ See Section 4.3.1

Table 1-1 General Specifications for DIGISEC RA __/23C Series Encoders*

Electrical

Resolution	Refer to Section 6
Accuracy	1/3 Bit RMS, excluding quantization
Output Signals (Fig. 1-1)	
Data format	Parallel, one output channel per bit
Logic levels	ONE: +3.5 to +5.5 vdc ZERO: ± 0.5 vdc
Rise and fall times	0.5 microseconds, maximum, with 3900-ohm load shunted by 1000 picofarads, or 10 TTL loads
Output stages	Fanout of ten 5404 or 7404 TTL elements
Settling time	3 microseconds, maximum

Note: Readout can be initiated 3 microseconds after application of external HOLD pulse

Input HOLD pulse	
Pulse levels	OFF (normal output): 0 to +0.5 vdc ON (frozen output): +3.5 to +5.5 vdc
Pulse width	Refer to Section 2.5
Sink current	7 milliamperes, maximum
Power requirements	
Voltage	Either: +5 vdc ± 2%, 1% max. peak to peak ripple Or: +6 vdc ± 2%, 1% max. peak to peak ripple (Refer to Section 6 for applicable voltage)
Current	600 milliamperes for 5V option 650 milliamperes for 6 V option

Mechanical

Outline dimensions	Refer to Section 6
Shaft	0.25-inch diameter, plain or splined
Weight	16 ounces
Rotation direction for increasing count	Refer to Section 6
Rotation rate	
Operating	360 rpm maximum
Slew	3600 rpm maximum
Rotor moment of inertia	0.4 oz.-in.^{2} maximum
Breakaway torque	0.5 oz.-in. maximum
Running torque	0.4 oz.-in. maximum
Shaft loading	
Axial	5 lbs. maximum
Radial	2 lbs. maximum at shaft end

*Refer to Section 6 for differences in models

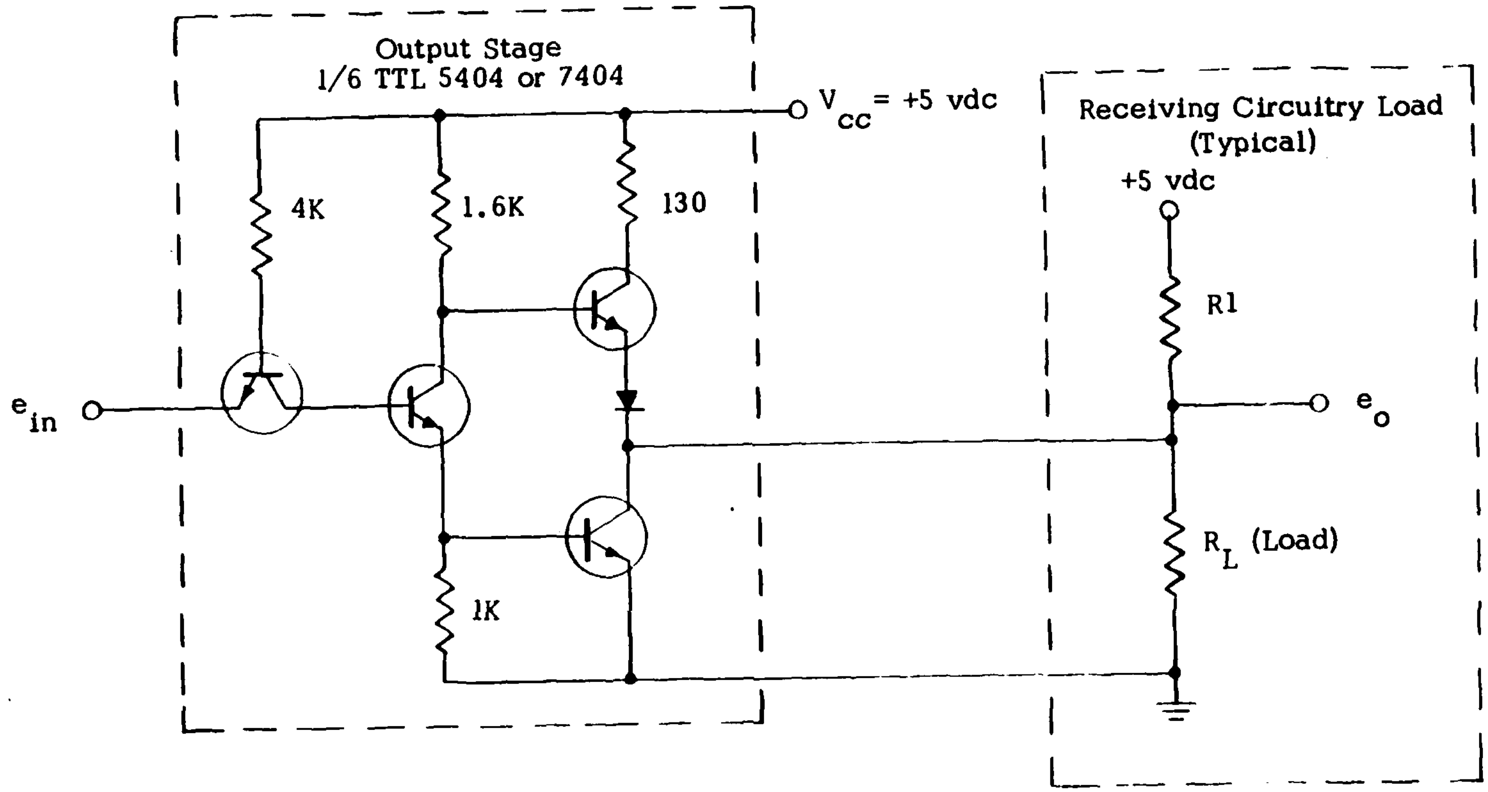
Table 1-1 General Specifications for DIGISEC RA __/23C Series Encoders (Cont.)

Environmental

	Military	Commercial
Temperature		
Operating	-40°C to +85°C	0 to +70°C
Non-Operating	-62°C to +90°C	-62°C to +90°C
Humidity	MIL-STD-202, Method 103, Condition B, (0-95%) operating. Will withstand 100% humidity with condensation non-operating.	
Shock	MIL-STD-202, Method 213, Test Condition A (50g peak, half sine wave, 11 ms duration, 3 shocks each direction each axis, 18 shocks total)	
Vibration	MIL-STD-202, Method 204, Condition D, except that vibration amplitude is .075 (total excursion) or 25g (peak) whichever is less. (swept sine, 10 hz to 2000 hz).	
Thermal shock	MIL-STD-202, Method 107C, Test Condition A-1 (25 cycles) except minimum temperature to be -62°C	
Salt atmosphere	MIL-STD-202, Method 101, Condition B, 5% salt solution	
Inclination	MIL-E-16400 Paragraph 4.5.14.2	

Rated Life

Mechanical, operating	10^9 revolutions minimum
MTBF	50,000 hours minimum calculated per MIL-HDBK-217A ground factors. 30,000 hours minimum calculated per MIL-HDBK-217A shipboard factors.



For e_{in} = 0 vdc, e_o High (One State)

$$e_o = \frac{5R_L}{R_L + \frac{R_1 \times 130}{R1 + 130}} - 0.6$$

For e_{in} = +5 vdc, e_o Low (Zero State)

$$e_o = \frac{5 \quad \frac{R_L \times 5}{RL + 5}}{R1 + \frac{R_L \times 5}{R_L + 5}}$$

Figure 1-1. Encoder Output and Load Interface Characteristics.

g. Small encoder diameter achieved by using integrated circuit modules.

h. Stainless steel case.

i. Low torque.

j. Field replaceable lamp assembly (long-life incandescent lamps) 1/

k. Field replaceable signal-processing integrated circuit modules requiring no field adjustment.

l. The use of anti-ambiguity logic which synchronizes all coarser data to the fine code track and thereby permits all but the fine track to be of relatively low accuracy.

m. Capability for readout on the fly at any speed up to the maximum rated operating speed. To allow for non-ambiguous readout on the fly, the encoders are designed to accept an external HOLD pulse which freezes the parallel outputs during readout.

1/ See Section 4.3.1

2. INSTALLATION AND OPERATION

2.1 HANDLING

DIGISEC RA __/23C series encoders are precision instruments and should be handled with care. Avoid shock to the encoder, particularly to the encoder shaft which is mounted on bearings to extremely fine tolerances. The plastic covering and the protective cap should remain in place during shipment or storage and should be removed only at the time that the encoder is installed in its operating location.

2.2 MECHANICAL ALIGNMENT

RA __/23C series encoders are supplied in standard Size 23 synchro mount configuration (see Fig. 2-1 and the outline drawing contained in Section 6). All encoders have a 1/4-inch OD shaft, either plain or splined.

CAUTION

> No alterations may be made to the encoder shaft or body except by the manufacturer, or warranty will be voided. Drilling or machining of the shaft will cause serious damage to the code disc, readout optics, or bearings.

CAUTION

> Do not use a rigid coupling between the encoder shaft and the drive shaft. A flexible coupling of high angular accuracy (Kinnemotive Corporation, Kinneflex series, or equivalent) must be used, unless the encoder is to be gear driven.

All splined shafts have a standard 22 teeth/96 pitch configuration with 1/4-28 outside thread, and are designed to accept a gear secured to the shaft by means of an MS 17186-4 or -8 drive washer and an MS 17178-3 drive nut.

The encoder may be installed in any attitude. However, the encoder shaft must be precisely aligned with the drive shaft because misalingment will degrade readout accuracy and shorten encoder life through excess loading of its bearings. The mounting hole must be bored to a diameter that is 0.001 inch (nominal) larger than the pilot diameter of the encoder.

CAUTION

> All misalignments between the encoder mounting surface and the drive shaft must be such that the radial and axial loading on the encoder shaft (through either the flexible coupling or drive gear) do not exceed the limits specified in Table 1-1.

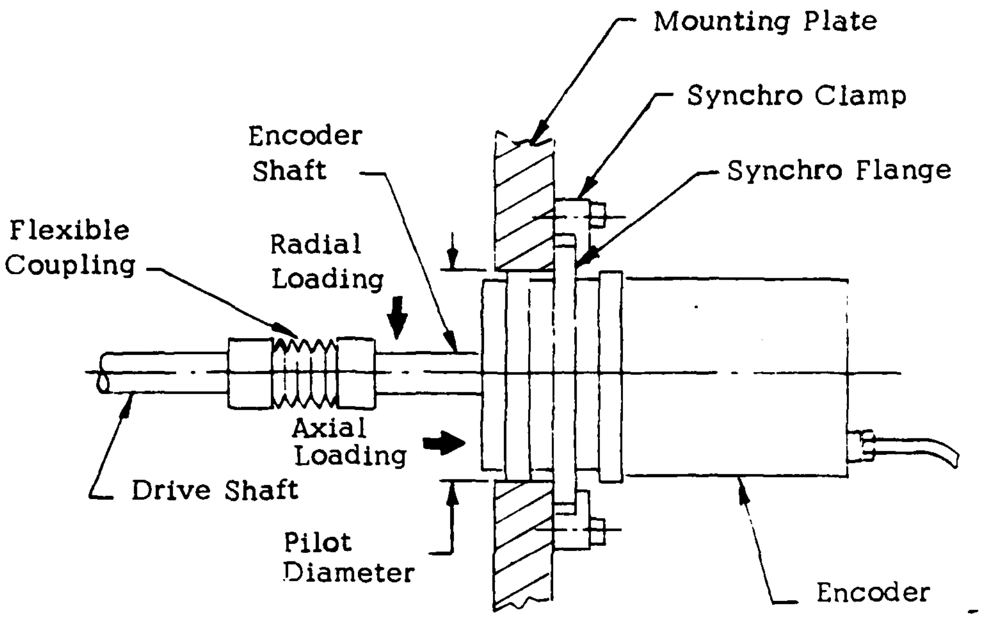


Figure 2-1 Installation of typical RA__/23C encoder with plain shaft

Note that the encoder has zero reference marks at the base of the shaft and on the case. These marks, when coincident, set the shaft angular position to a coarse zero count. The encoder must be oriented on its mounting surface so that its zero approximately coincides with that of the drive shaft. A standard Size 23 zeroing ring, to be driven by a pinion gear, may be inserted between the synchro flange and the mounting surface to facilitate precise zeroing after installation (see Section 2.4). Refer to MIL-HDBK-214A (Synchros) for information on applicable zeroing rings and associated components.

2.3 ELECTRICAL CONNECTIONS

2.3.1 Encoder Cable Wire Functions

All input/output electrical connections for the encoder cable are identified in the outline drawing contained in Section 6. The Lamp Test connection is used for troubleshooting (Section 4).

2.3.2 Grounding

Power and signal common are tied together within the encoder and are isolated from case ground since many applications require independent electrical and case grounds. In order to minimize noise problems, the noise level between the electrical and case grounds should be kept as low as possible. It is recommended that case ground be connected to electrical ground at only one point in the user's system, at a location to be determined experimentally for the particular installation.

2.3.3 Power Supply Considerations

RA __/23C series encoders are designed to operate from either +6vdc or +5vdc. The voltage applicable to a particular encoder can be found in Section 6.

NOTE

The external power supply must be set to provide +5V (or +6V), ± 2%, 1% maximum peak to peak ripple, at the encoder cable end in order to avoid possible erroneous readings caused by interconnection losses.

2.3.4 HOLD Pulse Line Driver Protection

The encoder HOLD pulse input is normally customer energized from a circuit that is powered from the same supply that operates the encoder, thus ensuring simultaneous application of power to all circuits of the encoder. In the event that the encoder proper and external HOLD driver circuits are energized from separate, non-interlocked supplies, this could result in a high state (5V) HOLD signal applied to an unenergized encoder, which could damage the encoder. Operation in this latter condition is allowed as long as a series protection diode is connected between the encoder HOLD line and the user's equipment. Any small signal diode with a PIV rating 50V or larger may be used. The anode of the diode should connect to the encoder HOLD line. The diode is considered part of the user's drive circuitry which should be capable of meeting the limits of para. 2.5 (Page 10) at the input of the encoder.

2.4 ZERO ALIGNMENT

1. Check that encoder is properly installed and that coarse zero has been set in accordance with instructions contained in Section 2.2.

2. Connect encoder to power supply and to output receiving circuitry.

3. Turn on power supply and receiving circuitry.

4. Slightly loosen the synchro clamps securing the encoder synchro flange to the mounting surface.

5. Set drive shaft to zero reference position.

6. While monitoring the encoder output with the receiving circuitry, carefully rotate the encoder case (either directly or through a gear-driven zeroing ring) until the zero is set to the desired tolerance.

7. Carefully tighten the synchro clamps. The encoder is now ready for operation.

2.5 OPERATION

After the encoder has been properly installed and connected to a power source and suitable receiving circuitry, operation involves only the application of the external HOLD pulse, as described below, for non-ambiguous readout on the fly. No adjustments or preventive maintenance are required aside from normal external cleaning procedures.

The encoder parallel outputs are always present as dc levels once external power is applied to the encoder. However, the anti-ambiguity logic within the encoder (Section 3.3) requires a certain amount of settling time (3 microseconds, maximum) which could cause improper readings if these were taken on the fly during a "settling cycle". To assure correct readout on the fly, DIGISEC encoders are designed to accept an external HOLD pulse. The net function of this pulse is to guarantee reliable readout if sampling is initiated 3 microseconds (or more) after the leading edge of the pulse and is terminated with the trailing edge. The HOLD pulse requirements are as follows:

OFF: ± 0.5 vdc, 7 ma Sink
ON: +3.5 to +5.5 vdc, 100 μa Source
Maximum Width: See below

The maximum width of the HOLD pulse can be determined from the following equation:

$$T_H = \frac{13.2 \times K \times R}{S}$$

Where T_H = Maximum width of HOLD pulse in microseconds

S = Shaft rotation speed in rpm

R = Encoder resolution in seconds of arc

K = A constant determined by the encoder type, as follows:

RA __/23C	K
10	1
11	1
12	1
13	2

If the HOLD pulse is applied for longer than T_H, the readout may show incorrect count. The maximum time, T_R, allowable for correct readout is therefore

T_R = (T_H-3) microseconds

3. THEORY OF OPERATION

A general functional block diagram applicable to all RA __/23C series encoders is contained in Fig. 3-1. The encoder consists of an optical subassembly, trim board A1, logic and hold board A2, and logic board A3. Field replaceable assemblies are the lamp assembly (which is part of the optical subassembly) and boards A2 and A3.

3.1 OPTICAL SUBASSEMBLY

The optical subassembly consists of a shaft-mounted glass code disc, a lamp assembly, and a slit and photodetector assembly. The code disc contains a series of concentric annular code tracks, each consisting of alternating transparent and opaque segments describing equal arcs along the circumference. The number of code cycles (one transparent segment followed by one opaque segment) varies by a factor of two from track to track, starting with one cycle on the inner track. The transparent and opaque states of all tracks thus represent the ONE and ZERO states of a multi-digit natural binary word, with one track per digit. The state of each track is sensed by illuminating the code disc and detecting the modulated light behind each track (as the disc is rotated) with a precisely registered slit/photodetector assembly. Each photodetector's output is essentially a square wave, with one cycle corresponding to a code cycle. The frequency of each detector's output is therefore a function of shaft rotation speed. When the shaft is stationary, the output of any detector is simply a dc level corresponding to either a ONE or ZERO state.

3.2 TRIM BOARD A1

Trim board A1 contains several trimming components which are factory set to provide the required parallel signal levels as they enter boards A2 and A3. The trim board components are precisely set for the detector outputs of its particular optical subassembly. Consequently, board A1 is not field replaceable.

3.3 LOGIC AND HOLD BOARD A2

Logic and hold board A2 performs two basic functions. It processes the parallel photodetector outputs corresponding to the three finer tracks on the code disc to provide the least significant digit (LSD) for all encoders as well as a few more significant digits for the higher resolution encoders (see Fig. 3-1). A2 also initiates anti-ambiguity control for the entire encoder and receives the input HOLD pulse enabling unambiguous readout on the fly.

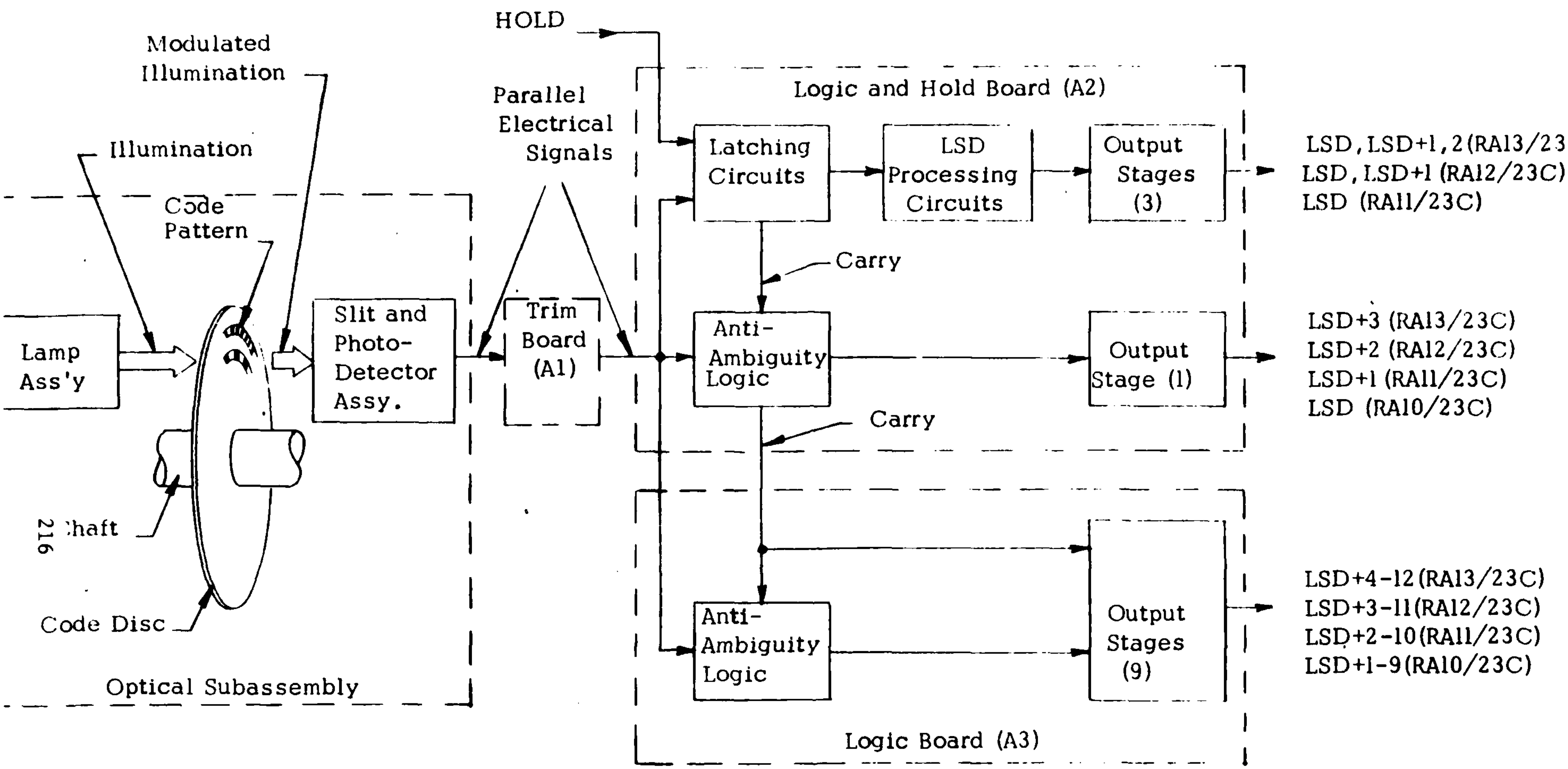


Figure 3-1 Encoder General Functional Block Diagram

216

The fine track circuit consists of the latching circuits, LSD processing circuits, and output stages shown in line in Fig. 3-1. This circuit generates one or more bits, depending on the encoder, from the fine track on the code disc. The CARRY signal represents the state of the fine track, which is used to synchronize the output transitions of all more significant bits. The CARRY cycle requires a maximum three microseconds of settling time after each fine track transition. Application of the HOLD pulse freezes the states of all bits through the latching circuits and through the nature of the CARRY logic. Reliable readout can then be initiated three microseconds after the leading edge of the HOLD pulse. The maximum duration of the HOLD pulse for reliable readout is dependent on the encoder resolution and shaft speed, as defined in Section 2.5. The output stages, which function as buffer amplifiers are either 5404 or 7404 TTL elements (refer to Section 1.3).

The two remaining tracks whose signals are processed by A2 each provide one of the parallel output bits. The CARRY from the fine track to the anti-ambiguity logic synchronizes the transitions of these two bits with those of the fine track. The more significant bit of the two, which is also the first CARRY for board A3, has its output stage on A3.

3.4 LOGIC BOARD A3

Logic board A3 processes the photodetector outputs of the remaining tracks to provide the remaining more significant bits. Each bit becomes the CARRY for the anti-ambiguity logic of the next more significant bit.

3.5 LAMP TEST CIRCUIT

The field replaceable lamp assembly[1/] contains several precisely aligned and potted lamps, all electrically connected in parallel. All RA 13/23C encoders contain four lamps. All remaining encoders (RA 12/23C, RA 11/23C, and RA 10/23C) contain three lamps. The LAMP TEST wire is connected to the less positive side of the parallel combination in each case. A voltmeter connected across the LAMP TEST wire and common will read the total current drain of all lamps through a series resistor. The normal readings can be found in Section 4. Failure of any lamp is indicated by a decrease in the current through the series resistor.

1/ See Section 4.3.1

4. MAINTENANCE

1.1 SCOPE

The optical subassembly of RA __/23C series encoders is factory aligned to extremely high precision. Therefore, field maintenance of encoders in this series is restricted to repair by replacement of the following three potted subassemblies: the lamp assembly, logic and hold board A2, and logic board A3. Refer to Section 6 for part numbers of replaceable assemblies applicable to any encoder. Trim board A1 is factory set for each encoder and is not field replaceable. It is partially hard-wired to the encoder.

The troubleshooting instructions which follow should help in isolating failure to either the external equipment, the three replaceable assemblies, or the rest of the encoder. Replace field replaceable assemblies in accordance with the instructions contained in Section 4.3. If failure is diagnosed in the non-replaceable portion of the encoder, no attempt should be made to correct the malfunction by opening the optical subassembly or forcing rotation of the encoder shaft. A detailed description of failure symptoms, suspected malfunctions, and operating conditions should be made. The encoder should then be carefully decoupled and removed from its mount, securely packed with its protective cap, plastic covering, and failure description, and returned to the manufacturer for repair.

If failure is diagnosed in the encoder cable, do not unsolder or solder wires where they connect to the encoder circuitry. Repair broken or shorted wires by splicing. If splicing does not correct the malfunction, replace the encoder.

RA __/23C series encoders have sealed bearings and no field lubrication is necessary.

CAUTION

> Do not open any portion of the encoder beyond providing access to the three field replaceable assemblies or warranty will be void. Repair of the optical subassembly beyond replacing the lamp assembly must be performed by the manufacturer.

.2 TROUBLESHOOTING

Troubleshooting the encoder involves first checking each of the parallel outputs for proper waveform amplitude and frequency as specified in Section 4.2.1. of the parallel output waveforms do not conform to the performance standards, the malfunction must be isolated either to the equipment external to the encoder (Section 4.2.2) or to the encoder itself (Section 4.2.3). Follow all steps in the order given. A voltmeter and a dual-trace oscilloscope equivalent to Tektronix Model 502A are required. Refer to the outline drawing contained in Section 6 for the encoder cable connections.

4.2.1 Encoder Output Test

Perform the output test as follows, using oscilloscope:

1. Shut off encoder power supply.

2. Disconnect encoder parallel outputs from receiving circuitry.

3. Turn on encoder power supply.

4. Rotate encoder shaft smoothly at maximum rated operating speed.

5. Connect oscilloscope (internal trigger) to parallel output in pairs, starting with LSD and LSD+1, then LSD+1 and LSD+2, (etc.) and check that all channel waveforms conform to the following standards:

   a. Each channel's output is a square wave with logic levels as follows:

      ONE: +3.5 to +5.5 vdc
      ZERO: 0.0 to +0.5 vdc

   b. Each channel's square wave has one half the frequency of the next less significant channel.

6. If any performance standard is not met, proceed with Sections 4.2.2 and 4.2.3 as judged necessary.

4.2.2 Troubleshooting External Equipment

If any performance standards are not met in Section 4.2.1, proceed with the following steps to check out the external equipment.

1. Check that encoder power supply voltage is within proper tolerance.

NOTE

Ensure that external power supply is set to +5V* (or +6V*), ± 2%, 1% maximum peak-to-peak ripple at the end of the encoder cable, to avoid possible erroneous readings caused by interconnection losses.

*Voltage depends on particular encoder (Section 6)

2. Check that output wires and receiving circuitry are free of shorts.

3. Check that encoder shaft is not binding and that coupling is not loose. If shaft is binding, check that encoder is installed in accordance with the requirements of Section 2.2. Encoder must be replaced if it is properly aligned mechanically but the shaft still binds.

4.2.3 Troubleshooting Encoder

If any performance standards are not met in Section 4.2.1, proceed with the following steps to check out the encoder itself.

1. Turn on encoder power supply.

2. Check encoder lamps 1/ by connecting voltmeter across LAMP TEST wire (+) and common.

Encoder Type	Performance Standard
RA 10/23C	0.35 vdc Minimum
RA 11/23C	0.35 vdc Minimum
RA 12/23C	0.35 vdc Minimum
RA 13/23C	0.4 vdc Minimum

If performance standard is not met, replace lamp assembly (refer to Section 4.3). * If in doubt, replace lamp assembly.

3. Check boards A2 and A3, in that order, by monitoring the following output channels on oscilloscope and checking for output waveform standards indicated in Section 4.2.1, Step 5.

Encoder Type	Board A2	Board A3
RA 10/23C	LSD	LSD+1 thru LSD+9
RA 11/23C	LSD, LSD+1	LSD+2 thru LSD+10
RA 12/23C	LSD, LSD+1, LSD+2	LSD+3 thru LSD+11
RA 13/23C	LSD, LSD+1 thru LSD+3	LSD+4 thru LSD+12

If any performance standard is not met, replace boards A2 and/or A3, in that order (refer to Section 4.3). Note that the anti-ambiguity control (CARRY) for the entire encoder is initiated on A2. If replacement of A2 and/or A3 does not correct the malfunction, failure resides in non-field-replaceable portions of the encoder.

* To test a lamp assembly outside the encoder, apply 3.5 to 4.0V to lamp block pins, and observe that all bulbs light. Bulbs also wear out due to gradual blackening, so that this test is not always conclusive. If in doubt, replace lamp assembly.

1/ See Section 4.3.1

4.3 PARTS REPLACEMENT

CAUTION

Shut off input power before removing or replacing components.

4.3.1 Removal/Replacement of Lamp Assembly (Figure 4-1)

NOTE

Replacement lamps are made up on a custom basis for specific encoders. Make sure the serial number on the replacement lamp matches the encoder serial number. Do not interchange lamps among encoders. The following three steps of this paragraph apply only to those encoders for which replacement lamps have been supplied.

Removal

1. CAREFULLY BRUSH AWAY ALL DIRT FROM THE FRONT OF THE ENCODER. Loosen two captive screws securing lamp assembly to encoder. Do not remove the screws from the lamp assembly.
2. Note the electrical contact pins, the alignment pins, and the sealing lip shown in Fig. 4-1.
3. Carefully remove the lamp assembly by pulling alternately on the two captive screws to overcome the friction from the sealing lip.

Replacement

Reverse removal procedures, taking care not to bend the electrical contact pins. Be careful not to get finger marks on the polished lamp reflectors. (See Page 19)

4.3.2 Removal/Replacement of Boards A2 and A3

Removal

1(a) Older Models - Pinch grommet at junction of encoder case and cable and push grommet and cable into case sufficiently to free cable.

1(b) New Models - Unscrew cable clamp packing nut and slide back nut and "O" ring.

2. Loosen two screws securing case to encoder on cable end of case. Remove screws.
3. Pull case back along cable to expose A2 and A3.
4. Carefully pull A2 and/or A3 back from its connector.

Replacement

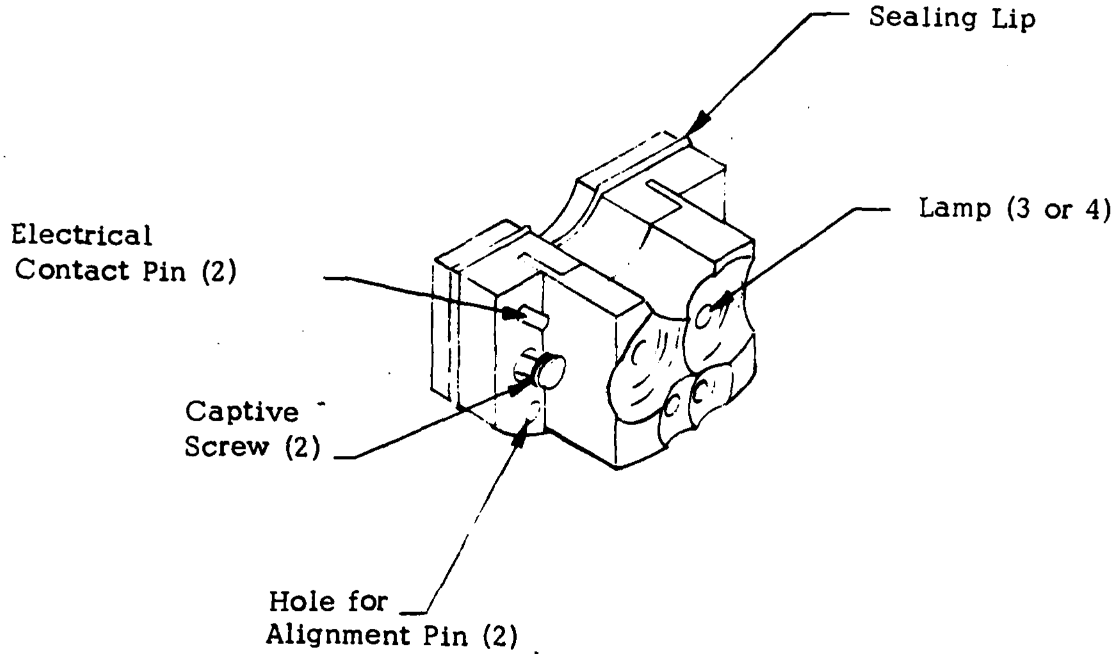
Reverse removal procedures.

CAUTION

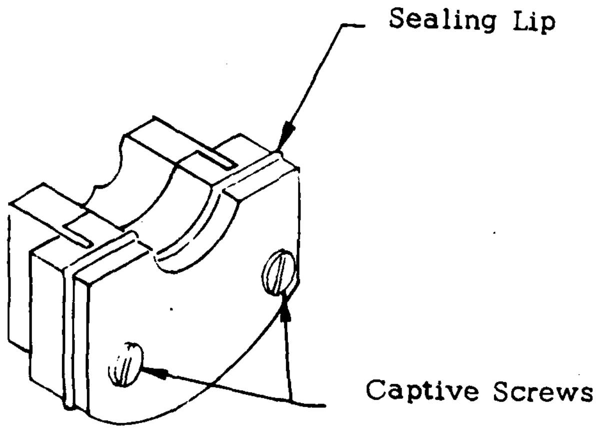
Plug-in boards are keyed to their proper sockets. Do not force a board into an improper socket or in reverse orientation.

NOTE

If cleaning of dirt or finger marks is required, use a cotton swab ("Q tip") and isopropyl alcohol. Swab gently and allow to dry. Do not use harsh or abrasive cleaning agents.



(a) Rear 3/4 View



(b) Front 3/4 View

Figure 4-1 Lamp Assembly

5. REPLACEMENT PARTS

Replacement parts applicable to any encoder of the DIGISEC __/23C series are listed in Table 5-1. These parts are the lamp assembly, logic and hold board A2, and logic board A3.

Table 5-1 Replacement Parts

Encoder Part Number (2785-__)	Replacement Part and Part Number		
	Lamp Assy. 1/ 2757-16G__	Logic and Hold Board (A2) 2757-33G__	Logic Board (A3) 2757-36G__
1, 33	1	6	2
2, 34	↓	3	1
3, 35	↓	6	2
4, 36	↓	3	1
5, 37	↓	6	2
6, 38	↓	3	1
7, 39	↓	6	2
8, 40	↓	3	1
9, 41	2	5	2
10, 42	↓	2	1
11, 43	↓	5	2
12, 44	↓	2	1
13, 45	↓	5	2
14, 46	↓	2	1
15, 47	↓	5	2
16, 48	↓	2	1
17, 49	↓	4	2
18, 50	↓	1	1
19, 51	↓	4	2
20, 52	↓	1	1
21, 53	↓	4	2
22, 54	↓	1	1
23, 55	↓	4	2
24, 56	↓	1	1
25, 57	↓	4	2
26, 58	↓	1	1
27, 59	↓	4	2
28, 60	↓	1	1
29, 61	↓	4	2
30, 62	↓	1	1
31, 63	↓	4	2
32, 64	↓	1	1

1/ See Section 4.3.1

6. DIFFERENCES IN MODELS

This section contains detailed specifications for all DIGISEC RA __/23C series encoders in addition to those listed in Table 1-1. These detailed specifications are listed in Table 6-1. Also contained in this section is an outline drawing (C 2000-583) that shows pertinent dimensions of all encoders, as well as optional shaft details. The drawing also identifies the electrical connections to the encoder.

All RA __/23C series encoders are identified by type number and part number. The type number gives the major (but not all) encoder characteristics as follows:

RA(a) /23C(b)X

Where R = rotary

A = absolute

(a) = resolution (Table 6-1, Column 1)

23 = standard Size 23 synchro configuration

C = contained electronics

(b) = temperature range (M - Military; C-Commercial)

X = modification of catalog unit; see supplement in front of manual for details.

The part number completely specifies the encoder.

Example: RA 12/23C(M), P/N 2757-47

Table 6-1 shows that this encoder has the following characteristics:

Resolution: 2^{12} transitions/revolution

Input voltage: +6VDC

Shaft style: Splined

Temperature range: Military

Direction of rotation for increasing count: CCW

Table 6-1 Detailed specifications for DIGISEC RA __/23C series encoders

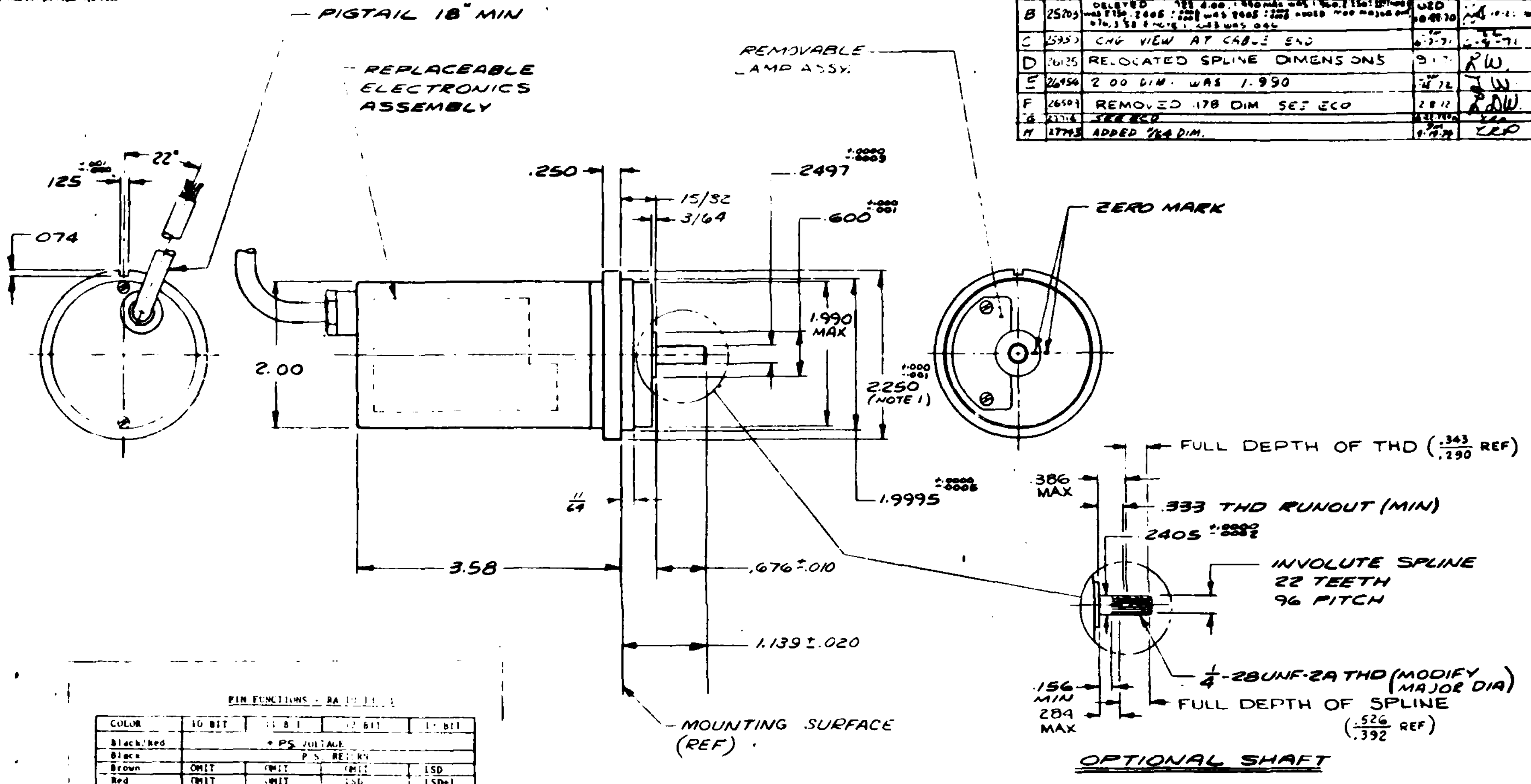
DIGISEC Type Number (RA __/23C)	Part Number (2785-__)	Angular Resolution (minutes)	Transitions per Revolution	Input Voltage (+ vdc)	Shaft (Note 5)	Temperature Range (Note 6)
13	1, 33	2.6	2^{13}	5	P	M
	2, 34				P	C
	3, 35				S	M
	4, 36				S	C
	5, 37			6	P	M
	6, 38				P	C
	7, 39				S	M
	8, 40				S	C
12	9, 41	5.3	2^{12}	5	P	M
	10, 42				P	C
	11, 43				S	M
	12, 44				S	C
	13, 45			6	P	M
	14, 46				P	C
	15, 47				S	M
	16, 48				S	C
11	17, 49	10.5	2^{11}	5	P	M
	18, 50				P	C
	19, 51				S	M
	20, 52				S	C
	21, 53			6	P	M
	22, 54				P	C
	23, 55				S	M
	24, 56				S	C
10	25, 57	21.1	2^{10}	5	P	M
	26, 58				P	C
	27, 59				S	M
	28, 60				S	C
	29, 61			6	P	M
	30, 62				P	C
	31, 63				S	M
	32, 64				S	C

Notes

(1) Outline dimensions shown on drawing C 2000-583
(2) Electrical connections listed on drawing C 2000-583
(3) Rotation for increasing count, defined looking at shaft end of encoder:
  (a) Part numbers 2757G1 through G32, clockwise
  (b) Part numbers 2757G33 through G64, counterclockwise
(4) Other specifications listed in Table 1-1
(5) P = plain, S = splined (see drawing C 2000-583)
(6) M = military, C = commercial (refer to Table 1-1)

REVISIONS

LTR	ECO	DESCRIPTION	DATE	CHKR APVD
A	24801	REPLACES REV - WITH CHANGE	6-26-70	
B	25705	DELETED ... [illegible]	10-28-70	
C	25953	CHG VIEW AT CABLE END	6-3-71	
D	26125	RELOCATED SPLINE DIMENSIONS	9-1-71	RW
E	26456	2.00 DIM. WAS 1.990	11-4-72	RW
F	26509	REMOVED .178 DIM SEE ECO	2-8-72	RDW
G	27116	SEE ECO		
H	27745	ADDED 11/64 DIM.	9-19-74	



PIN FUNCTIONS - RA [illegible]

COLOR	10 BIT	11 BIT	12 BIT	13 BIT
Black/Red	+ PS VOLTAGE			
Black	P.S. RETURN			
Brown	OMIT	OMIT	OMIT	LSD
Red	OMIT	OMIT	LSD	LSD+1
Orange	OMIT	LSD	LSD+1	LSD+2
Wht/Blk Orn	HOLD	HOLD	HOLD	HOLD
Yellow	LSD	LSD+1	LSD+2	LSD+3
Green	+1	+2	+3	+4
Blue	+2	+3	+4	+5
Violet	+3	+4	+5	+6
Gray	+4	+5	+6	+7
White	+5	+6	+7	+8
White Black	+6	+7	+8	+9
White Brown	+7	+8	+9	+10
White/Red	+8	+9	+10	+11
White/Orn	LSD+9	LSD+10	LSD+11	LSD+12
White Gray	CASE GROUND			
White/Yel	LAMP TEST			

1 MILITARY UNIT TO BE AS SHOWN.
COMMERCIAL UNIT TO BE 2.25 ±.010

NOTES:

QTY	ITEM NO	PART NO	DESCRIPTION	MATERIAL	SPECIFICATION	CODE IDENT

LIST OF MATERIAL

UNLESS OTHERWISE SPECIFIED
ALL DIMENSIONS ARE IN INCHES. DIM AND TOL PER USA STD Y14.5. DIM APPLY AFTER PROCESSING.
TOLERANCES
2 PLACE DEC ± .01 CHAMFER ± .01 × 45°
3 PLACE DEC ± .005 DEG ± 30 MIN
SURFACE ROUGHNESS PER USA STD B46.1 ✓
BREAK SHARP EDGES AND CORNERS .005 TO .020 RADIUS

		DATE
DWN	UZD	6-26-70
CHKR	A L	6-26-70
ENGR		6-26-70
QA	H M Cann	8-19-70
APVD		6-26-70

MATERIAL N/A
FINISH N/A
NEXT ASSY: USED ON 4-2757
APPLICATION

Itek Measurement Systems Division

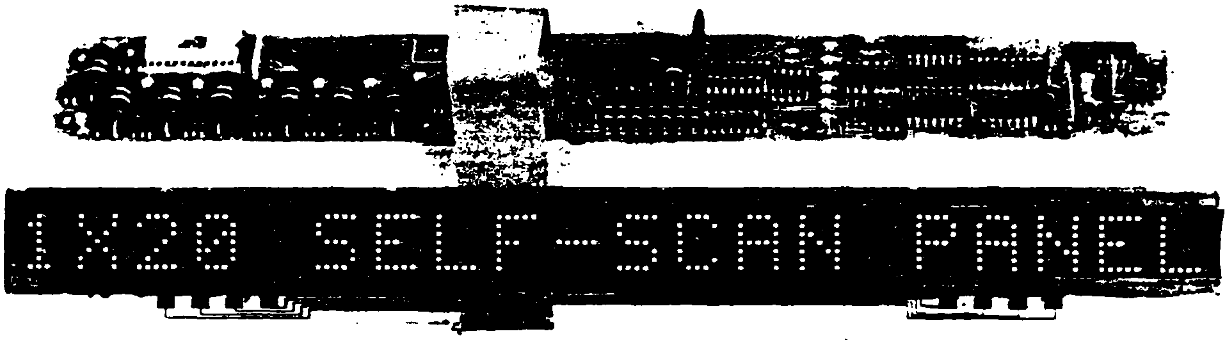
OUTLINE SIZE 23 RA-123C ENCODER

SIZE C | CODE IDENT NO 06130 | 2000-583 | REV H
SCALE 1/1 | WT | SHEET 1 OF 1

C 2000-583

SELF-SCAN® II PANEL DISPLAY

[illegible]



The model SII0120-0030 SELF-SCAN II panel display is a single-line, intermediate size, 20-character-wide, alphanumeric display that is ideal where readability and visibility are primary considerations. The display presents a bright, flicker-free, soft neon-orange glow that is characteristic of gas plasma technology. An additional feature of the panel is its buttability, which permits it to be assembled into multi-panel large displays. For example, a 1920-character display consisting of 24 rows of 80 characters each can be mounted in an enclosure 4 feet by 5 feet by 5 inches.

The panel display operates in a multiplexed scanning mode, with scanning being performed from left to right. Because of the internal panel address feature, only 14 external connections are required to control all of the functions of the panel. The internal address feature also substantially reduces the drive electronics required in comparison to a standard X-Y address matrix display.

The light output is generated by a neon glow discharge between transparent anodes on the front glass plate (for the horizontal rows) and the cathodes (corresponding to the columns) on the rear glass.

The cathodes are bussed in a six-phase arrangement so that Ø1 cathodes are columns 1, 7, 13, etc. While the common Ø1 cathodes are all driven low simultaneously during clock periods 1, 7, 13, etc., a glow occurs only under one cathode column due to the internal panel characteristics. This glow is under the anodes addressed by the character generator or auxiliary data inputs. This matrix address results in only those display cells needed in that one vertical column being on at a given moment of time.

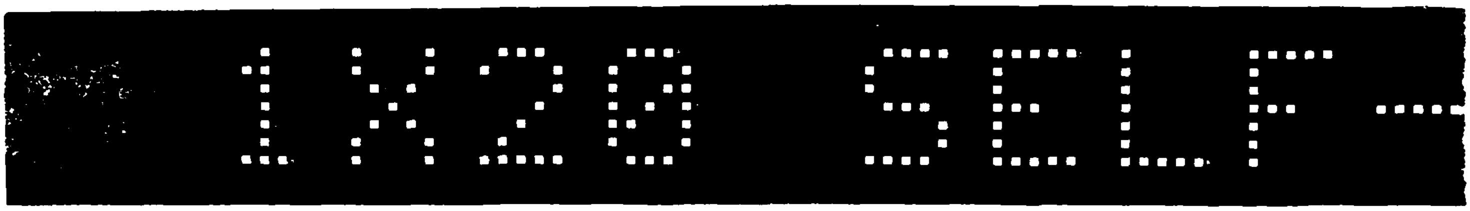
This display can be directly interfaced to computer/microprocessor based systems because all logic level inputs/outputs are TTL compatible. The display is ideal for applications where information must be presented to an operator,

ENVIRONMENTAL AND MECHANICAL CHARACTERISTICS

Operating Temperature	0° to 50°C
Storage Temperature	-40° to $+85^{\circ}$C
Relative Humidity (no condensation)	90% max.
Weight	14 ounces
Size	14" x 2" x 1½"
Shock	20 g, ½ sine wave, 11 ms in Y axis
Vibration	
Constant	2 g acceleration, 50–100 Hz, 10 min each axis
Sinusoid	0.018" double amplitude, 5–50 Hz
Operating Altitude	10,000 ft. max.
Storage Altitude	30,000 ft max

OPTICAL CHARACTERISTICS

Character Height	0.65 inch
Character Width	0.55 inch
Dot Size	0.05 inch square
Dot Center-to-Center Spacing	0.10 inch
Luminous Intensity	230 microcandelas
Light Output	60 ft-Lamberts (Note 1)
Contrast Ratio	5 to 1 at 300 ft.-L
Horizontal Viewing Angle	150°
Vertical Viewing Angle	50°
Color	Neon Orange



CHARACTER FORMAT (Actual Size)

as in a POS terminal. Each character is displayed in a 5 x 7 dot matrix, and formed of 0.050-inch square cells. Characters are defined by a positive logic six-bit ASCII code. Used in conjunction with the count logic, a character is formed by turning the display dot cells on and off as required at approximately 70 Hz.

The appropriate six-bit ASCII code for each desired character must be present for a minimum of five clock periods of each character position. After the 20th character is displayed, a reset pulse must be supplied to start a new scan. The character displayed in the extreme left location corresponds to the ASCII code present at the data input lines just after the reset pulse. The subsequent characters are displayed sequentially to the right according to the ASCII code provided to the display.

While the panel display is provided with a character generator capable of displaying a 64-character ASCII subset repertoire, seven auxiliary data input lines permit the character generator to be bypassed so that additional symbols or characters can be displayed. Each auxiliary data line controls one horizontal row of dot cells. A logic 0 at an auxiliary data input line turns on a cell; a logic 1 keeps the cell off.

When the auxiliary data inputs are used in conjunction with the character generator, either a logic 1 level must be applied to pin 1 (display disable) or a blanking code must be present at all the data input lines. In addition, a logic 1 level must also be present at all auxiliary data inputs during the entire reset period, during the last two columns of each character position, and for 14 us (min) after each high-to-low transition of the clock.

An external clock signal of 100 to 120 us provides the basic system timing. For complete scan cycle of the panel, 139 clock pulses are required: 138 clock pulses for the six-phase drive, and one pulse for scan reset. The screen of the panel can also be blanked by applying logic 0 level signal at the display disable input, provided all auxiliary data inputs are at logic 1 level.

The drive circuitry board is mounted with component side accessible to the user. This permits the character generator to be field-replaceable without dismantling the panel/driver board assembly.

For additional information or applications assistance on this panel, write to Burroughs Corporation, Electronic Components Division, P. O. Box 1226, Plainfield, New Jersey 07061; or call our special sales/applications number, (201) 757-3400 in New Jersey, or (714) 835-7335 in California.

NOTES:
1. ALL UNTOLERANCED DIMENSIONS ARE REFERENCE.
2. OPTIONAL PANEL SUPPORT CLIPS ARE SUGGESTED FOR ENVIRONMENTAL CONDITIONS WHERE EXTREME SHOCK AND VIBRATION IS ANTICIPATED.
3. SUGGESTED ALTERNATE MOUNTING PATTERN, HARDWARE KIT WILL SUPPLY (4) FOUR MOUNTING BRACKETS WHEN REQUESTED.
4. PRINTED WIRING BOARD INTERFACE CONSISTS OF 26 PINS .025 SQUARE .318 HIGH AND LOCATED (2) TWO ROWS ON .100 x .100 SPACING.

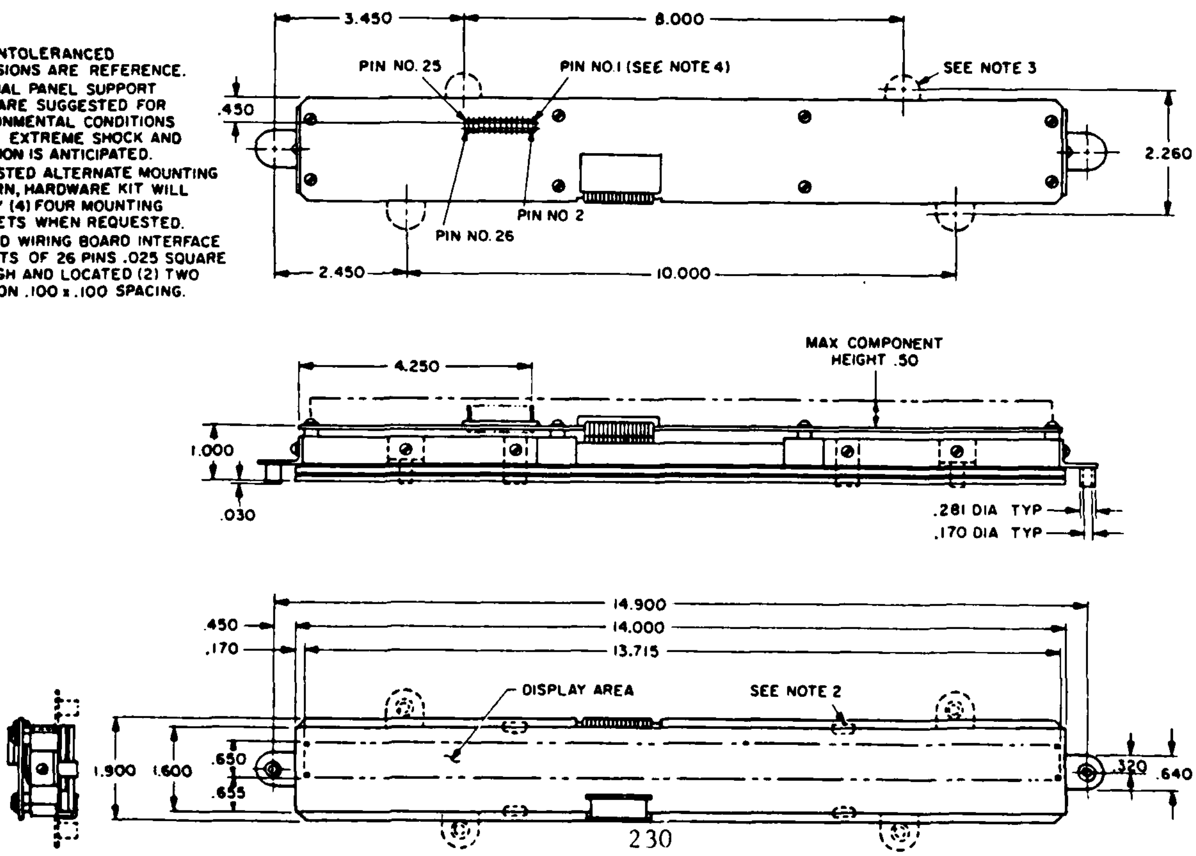


Figure 1. OUTLINE DRAWING

SCAN PANEL

ELECTRICAL CHARACTERISTICS (Note 4)

Power Required	
Positive Logic Supply	4.75 to 5.25V @ 350 mA max.
Negative Logic Supply	−11.4 to −12.6V @ −50 mA max.
Display Supply	−237.5 to −262.5V @ −110 mA max.
Clock Input Signal (See Figure 1)	
Logic 1 Level	2.0 to 5.25V @ 40 uA
Logic 0 Level	0 to 0.8V @ −7 mA
Clock Period	100 to 120 us
Logic 0 Voltage Duration	20 us to Clock Period −20 us
Data Input Signals	
Logic 1 Level	3.75 to 5.25V @ 10 uA max.
Logic 0 Level	−7.0 to 0.6V @ 10 uA max.
Duration (Note 2)	5 Clock Periods
Auxiliary Data Input Signals (Note 3)	
Logic 1 Level	4.35 to 5.55V @ 20 uA max.
Logic 0 Level	0 to 0.4V @ −4 mA max.
Logic 1 Duration	14 us min. to 1 Clock Period max.
Reset Input	
Logic 1 Level	2.0 to 5.25V @ 40 uA
Logic 0 Level	0 to 0.8V @ −7 mA
Duration	2 to 4 us
Reset Input Delay	0 to 1 us
Display Disable Input (Blanking Control)	
Logic 1 Level	2.0 to 5.25V @ 40 uA
Logic 0 Level	0 to 0.8V @ −7 mA
Data Update Output (Pulse Indicating End of Character)	
Logic 1 Level	2.0 to 5.25V @ −2 mA
Logic 0 Level	0 to 0.4V @ 10 mA

Table 2. TRUTH TABLE

BINARY INPUT	CHAR.	BINARY INPUT	CHAR.	BINARY INPUT	CHAR.
0	@	22	V	43	+
1	A	23	W	44	,
2	B	24	X	45	—
3	C	25	Y	46	.
4	D	26	Z	47	/
5	E	27	[	48	Ø
6	F	28	~	49	1
7	G	29	]	50	2
8	H	30	{	51	3
9	I	31	}	52	4
10	J	32	BLANK	53	5
11	K	33	!	54	6
12	L	34	"	55	7
13	M	35	#	56	8
14	N	36	$	57	9
15	O	37	%	58	:
16	P	38	&	59	;
17	Q	39	'	60	<
18	R	40	(	61	=
19	S	41	)	62	>
20	T	42	*	63	?
21	U				

Table 1. PIN CONNECTIONS

1	Display Disable In	14	Aux. Data 6 In
2	Data Update Out	15	Binary 2 In
3	Clock In	16	Aux. Data 7 In
4	Not Used	17	Binary 4 In
5	Not Used	18	Aux. Data 5 In
6	−250V	19	Binary 8 In
7	Reset In	20	Aux. Data 3 In
8	Not Used	21	Not Used
9	Ground	22	Aux. Data 1 In
10	Aux. Data 2 In	23	Binary 16 In
11	Not Used (Leave Open)	24	−12V
12	Aux. Data 4 In	25	Binary 32 In
13	Binary 1 In	26	+5V

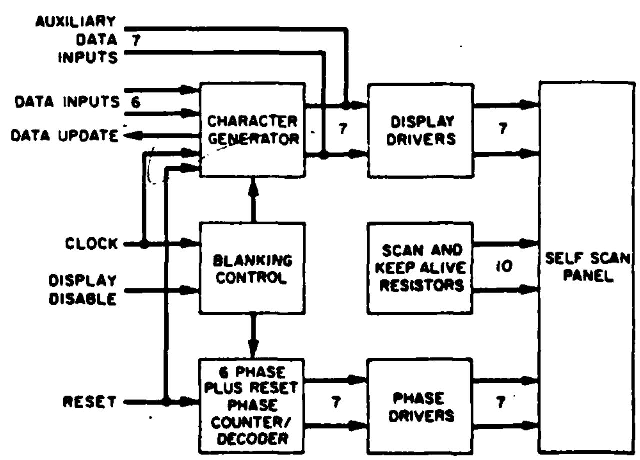


Figure 2. BLOCK DIAGRAM

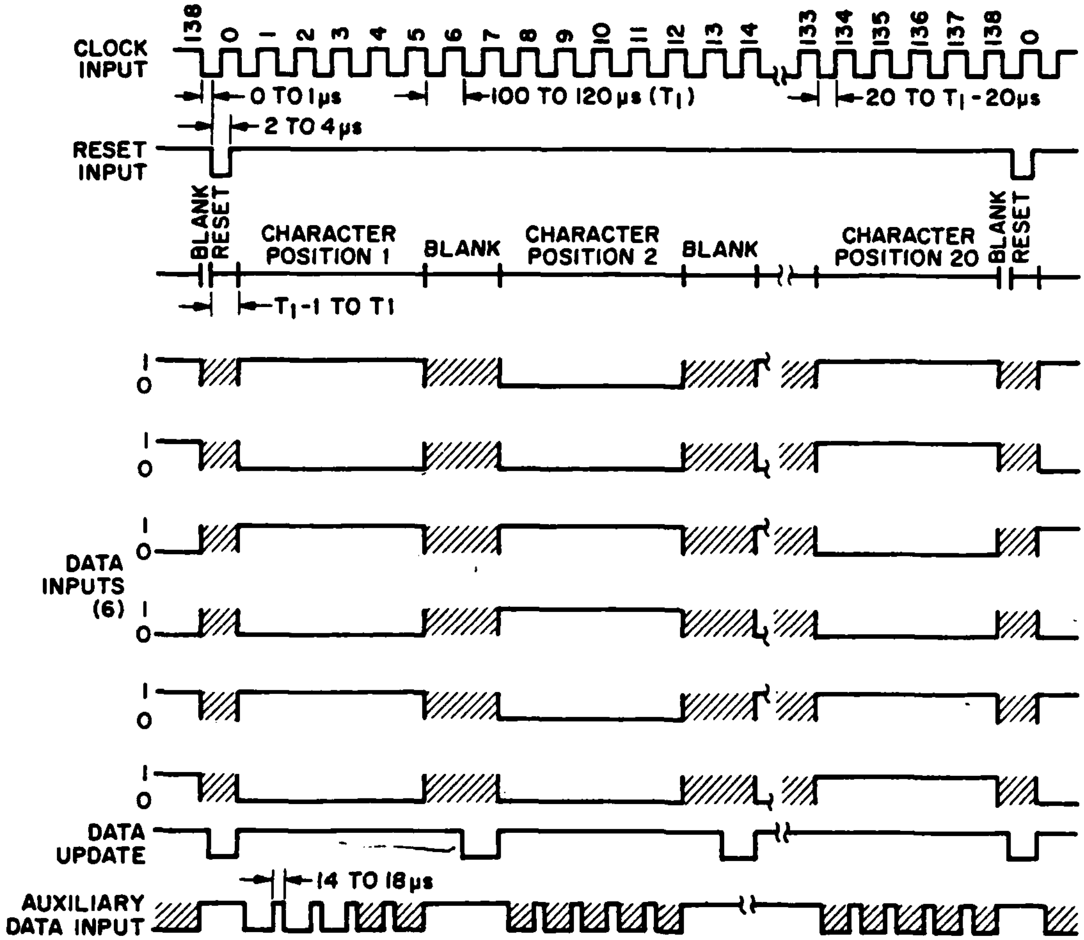


Figure 3. SYSTEM TIMING DIAGRAM

NOTES

1. This value is a typical time-averaged luminous intensity per dot at a current of 10 mA. The intensity may vary slightly with individual panels; but within any panel, all cells will have a constant luminous intensity.
2. Data input must remain constant for the first five clock periods of each character position. A logic 1 level is "true" data.
3. These inputs may be left open-circuited when not used. These inputs must be pulled up to positive logic supply voltage level when used. They must be in logic 1 state for at least 14 us after every negative clock transition and during the entire reset period.
4. Absolute ratings beyond which life and performance will be impaired.

PRINTED IN U.S.A.

APPENDIX D

EXTERNAL SERIAL INTERFACE

External Serial Interface

The serial interface on the microprocessor allows remote operation of the radome positioner. This interface is based on the RS-232-C interface standard*. ASCII commands are entered from a remote device which invokes the same responses as keyboard entries. A list of valid commands for the serial interface are given in Table D-1. A continuous display of current positioner status is sent to the external device. This device may at anytime send a valid command back through the interface to the microprocessor. Any invalid command received by the microprocessor will invoke an error message. The display on the radome positioner will echo any valid command just as it does for a keyboard entry. A switch, located at the front panel of the computer, will determine the mode of operation of the RFSS Radome Positioner. The two modes of operation are "Local" and "Remote". The local mode will allow only keyboard access and the remote mode will deny keyboard access and allow remote entry of valid commands. Note, however, that the arrow commands can only be used in setting their respective azimuth and elevation limits. Also, a valid command must be typed in to start the continuous display from the serial interface.

Access to the serial interface is by way of a EIA standard 25-pin connector located on the back panel of the microcomputer chassis. This connector, labeled "RS-232-C", is attached to connector P3 on the micromodule 1A board by way of solid wire ribbon cable as indicated on drawing #65. Pin identification using this standard is given in Table D-2.

The low data rates used in this system require no handshaking through the serial interface, therefore, a jumper from pin 15 to pin 14 of P3 (Drawing 65) has been used to constantly enable the I/O port of the microprocessor. For a software listing of the serial interface routine, please refer to Appendix B.

---

*EIA STANDARD NO. RS-232-C, "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange", August, 1969, Electronics Industries Association, Engineering Department, 2001 Eye Street N. W., Washington, D.C.

TABLE D-1

Valid Serial Interface Commands

S ≡ Start/Stop

E ≡ Set Elevation

A ≡ Set Azimuth

P ≡ Program

. ≡ Decimal Point

- ≡ Minus Sign

↓ ≡ DOWN

↑ ≡ UP

← ≡ LEFT

→ ≡ RIGHT

NOT ACCESSIBLE with in REMOTE (↓, ↑, ←, →)

Manual positioning

[illegible] 1 | E, 20.0, S

2 | E, 20.0, A, 10.0, S

To access limits

{E/A}, [↓, ↑, ←, →] [illegible]

[illegible] Limit: change it [illegible].

To exit limit ck S.

To run programs

P Enter {[illegible]} see [illegible] E{1, 2, 3, 4}

It will ask for parameters (again see appendix E) then S

TABLE D-2

Pin Description for Serial Interface Connector

Pin Number (25 Pin Connector)	P3-Edge Connector (20 Pin Connector Drawing 65)	Description	Abbreviation
1	1	Protective Ground	GND
2	3	Transmitted Data	Tx DATA
3	5	Received Data	Rx DATA
4	7	Request to Send	RTS
5	9	Clear to Send	CTS
6	11	Data Set Ready	DSR
7	13	Signal Ground	GND
8	15	Received Line Signal Detector	SIG DET
9	17	Ground	GND
10	19	Ground	GND
11 - 19	x,x,x,2,4,6,8,10,12	Not Used	—
20	14	Data Terminal Ready	DTR
21 - 25	16,18,20,x,x,	Not Used	—

APPENDIX E

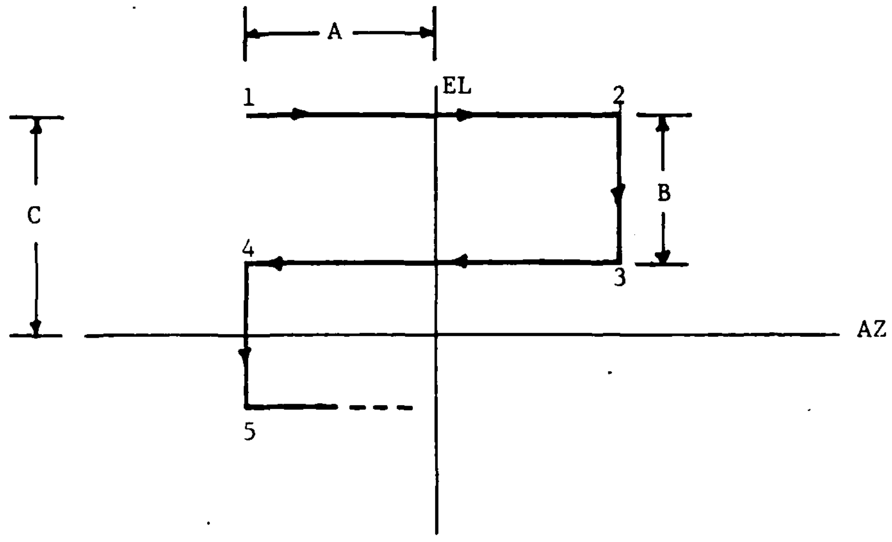
PRESTORED RASTER SCAN PATTERNS

Raster Scan Patterns

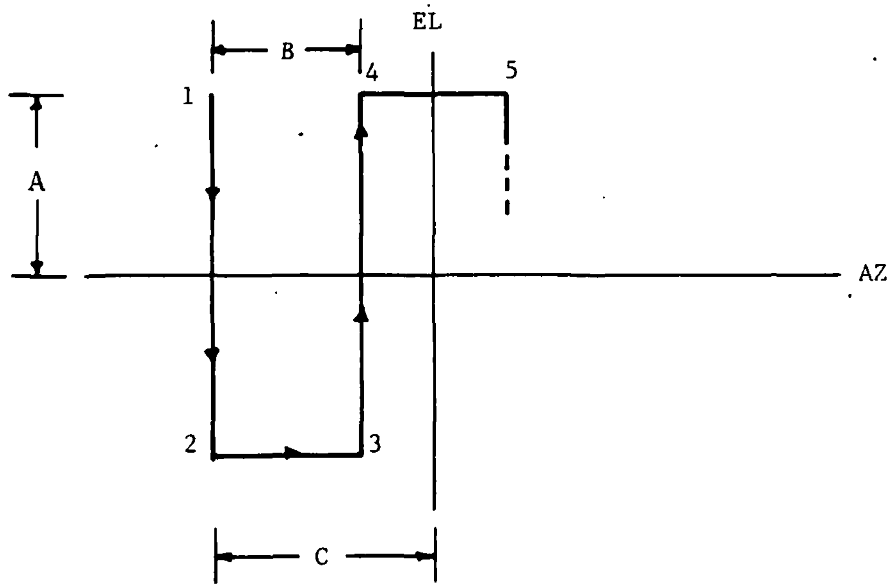
The patterns are generated from keyboard entries made by the user. The entries are variable parameters that determine how each pattern is generated. The display is used to prompt user access through either the keyboard or external serial interface by the use of these variable parameters: A, B, and C. Currently, there are four user programmable patterns that are invoked by activation of the "PROG" key on the keyboard. The display will then ask the user to enter a programmed pattern number from 1-4. The user will then be prompted by the display to enter the required parameters upon which the processor will then wait for a "START" key to be pressed before the desired program will start.

Patterns 1, 2 and their associated variables are defined in Figure E-1. The microcomputer, using the entered values of "A" and "C", computes point 1 and promptly moves the positioner to that point. The positioner will briefly stop and then move to the next calculated position, point 2. Point 3 is computed by the entered parameter "B". The positioner is moved to point 4 taking advantage of the change in coordinate signs and then finishs one period of the scan after arriving at point 5. The remainder of the raster positions are calculated in a similar manner. Pattern 2 is generated in a similar manner, the only difference being a 90 degree shift of the AZ/EL axes.

Patterns 3 and 4, shown in Figure E-2, access stored trigonometric values which are used to generate the desired patterns. These stored values can be found in the software listing of Appendix B. Activation of pattern 3 will initialize the positioner at the origin of the coordinate system. This position is referred to in the figure as point 1. The positioner will then move up in elevation until it reaches point 2. One leg of the star has now been generated. It will then move down in elevation and stop when point 3 is reached. The positioner will move back up to the origin (point 1). The positioner moves in a similar manner to complete the star raster. The legs of the star are separated by the entered angle

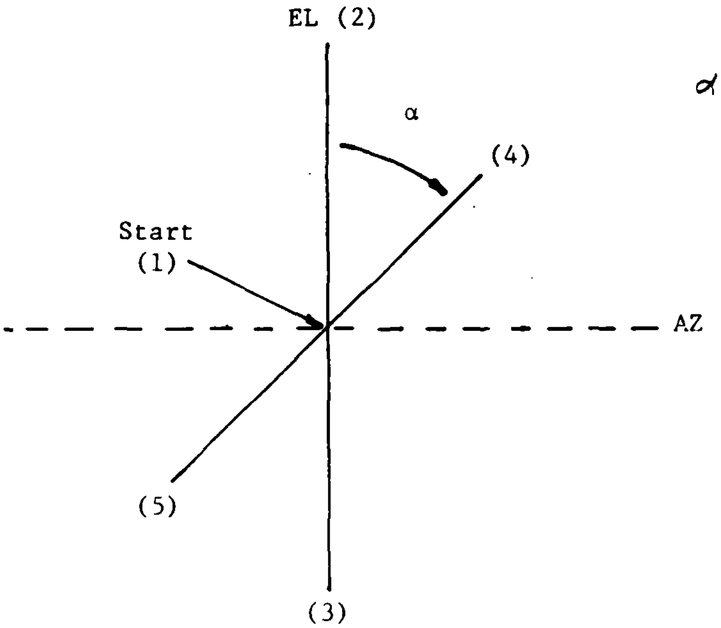


a) Pattern #1

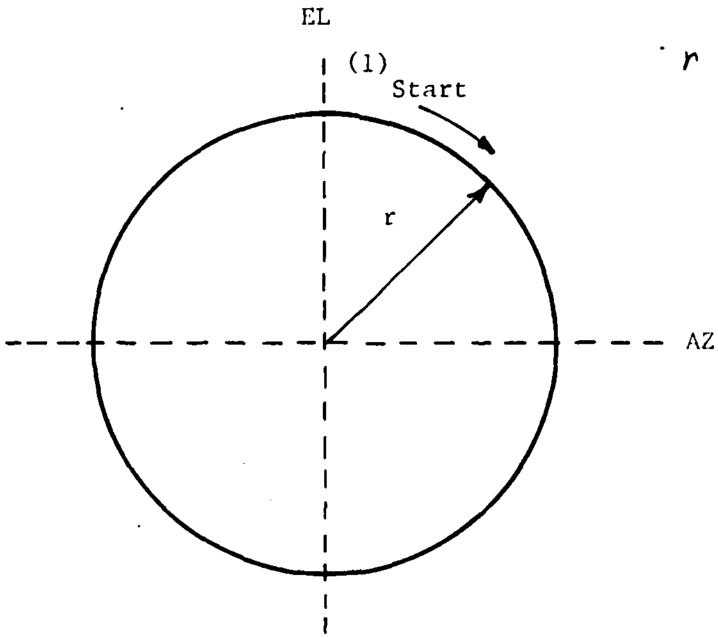


b) Pattern #2

Figure E-1. Linear Raster Patterns



a) Pattern #3 (Star)



b) Pattern #4 (Circle)

Figure E-2. Star and Circle Raster Patterns

α and all the end points of the legs are computed using analytical geometry. These points are defined as points on a circle with a maintained radius. Pattern 4 is a circle of which the radius is a variable through an entered parameter r. The circle is begun at point 1 on the elevation axis and moves in a clockwise direction with a constant increment of one degree. This angular resolution cannot be changed by the user and always completes a cycle of 360 degrees. The software listing of all four patterns can be found in Appendix B.

END
FILMED

5-86

DTIC