

AD-A152 715

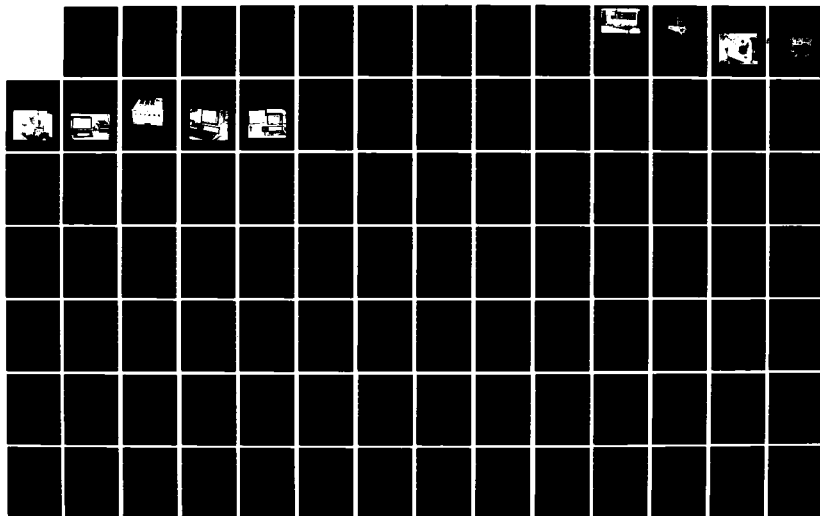
MANUFACTURING INFORMATION SYSTEM(U) BRIGHAM YOUNG UNIV
PROVO UT COMPUTER AIDED MFG LAB D K ALLEN ET AL.
26 DEC 84 AFOSR-TR-85-0275 AFOSR-82-0253

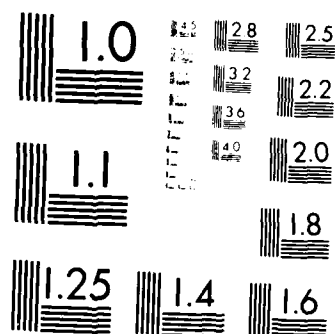
1/3

UNCLASSIFIED

F/G 13/8

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A152 715

AFOSR 72

275

(7)

FINAL REPORT
AFOSR Grant #82-0253
July 1, 1982 through October 31, 1984
MANUFACTURING INFORMATION SYSTEM

Submitted to
Air Force Office of Scientific Research
Building 410, Room 223
Bolling Air Force Base
Washington, D.C. 20322 N.C.

December 26, 1984

Principal Investigators:
D.K. Allen, P.R. Smith & M.J. Smart

Computer Aided Manufacturing Laboratory
Brigham Young University
Provo, Utah 84602

JB

DTIC FILE COPY

85 03 29 023

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified/Unlimited			1b. RESTRICTIVE MARKINGS None	
2a. SECURITY CLASSIFICATION AUTHORITY Public Law 97-258			3. DISTRIBUTION/AVAILABILITY OF REPORT Unrestricted	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Annual Report AFOSR 82-0253			5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR TR. 82-0253	
6a. NAME OF PERFORMING ORGANIZATION Brigham Young University		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6c. ADDRESS (City, State and ZIP Code) Research Division C-37 ASB Provo, UT 84602			7b. ADDRESS (City, State and ZIP Code) Building 410 Bolling AFB Washington, D.C. 20332	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Office of Scientific Research		8b. OFFICE SYMBOL (If applicable) AFOSR/	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-82-0253	
8c. ADDRESS (City, State and ZIP Code) AFOSR/NE Building 410 Bolling AFB, Washington, D.C. 20332			10. SOURCE OF FUNDING NOS.	
			PROGRAM ELEMENT NO. 61102F	PROJECT NO. 2305
TITLE (Include Security Classification) Manufacturing Information System (UNCL)				
PERSONAL AUTHOR(S) D.K. Allen, P.R. Smith, M.J. Smart				
11a. TYPE OF REPORT Final		13b. TIME COVERED FROM 82 Jly 01 TO 84 Oct 31		14. DATE OF REPORT (Yr., Mo., Day) 84 Dec 26
15. PAGE COUNT				
15. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.	Manufacturing, Integrated, Information, Flexible, Distributed, System, Assembly, Packaging, Computer Programs, Mechanical, etc.	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
This is the final report of a project to develop prototype miniature laboratory apparatus to be used in conducting a series of experiments and investigations relating to a Manufacturing Information System. The size and cost of manufacturing equipment has made it extremely difficult to perform realistic modeling and simulation of the manufacturing process in university research laboratories. Likewise the size and cost factors, coupled with many uncontrolled variables of the production situation has even made it difficult to perform adequate manufacturing research in the industrial setting. The difficulty of developing Integrated Manufacturing Systems is well documented by the large amount of funding and effort being spent by industry and government. One example MORE				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS PT. ☐ OTIC USERS ☐			21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL D.K. Allen, P.R. Smith, M.J. Smart			22b. TELEPHONE NUMBER (Include Area Code) 202-700-4433	22c. OFFICE SYMBOL N/A

No. 19

of a large scale industry/government program was the I-CAM project. This project lasting 10 years and \$250M has been only partially successful, attesting to the difficulty of the integration challenge.

It was the purpose for research funded under this grant to continue the development of miniature prototype equipment for use in an integrated CAD/CAM Laboratory. The equipment developed under this grant and ^{from} previous work is capable of actually performing production operations (e.g. drilling, milling, turning, punching, etc.) on metallic and non-metallic workpieces.

It is now expected that the prototype equipment developed or otherwise acquired under this grant will now provide the basis for extensive research on Manufacturing Information Systems, Common Database Development, CIM Application Program Development, Local Area Networking, and Knowledge-based CAD/CAM Training utilizing Interactive Videodisc Delivery Systems.

DDC
QUALITY
INSPECTED
1

MANUFACTURING INFORMATION SYSTEM

(Equipment Development Phase)

1. INTRODUCTION

This is the final technical report for research completed during a 28-month effort to develop prototype laboratory sized manufacturing equipment for use in Manufacturing Information System research.

1.1 NEED

In response to a long-felt need and effort to improve manufacturing engineering/technology education, the development of a full sized computer aided manufacturing (CAM) laboratory was undertaken at Brigham Young University in 1974 as a BYU Centennial year activity. The purpose of this laboratory was to provide advanced facilities for 1) teaching, 2) research, and 3) technology transfer. By 1976 a building had been erected for the laboratory and a number of pieces of full sized industrial CAD/CAM equipment had been acquired through cost-sharing and educational contributions. One hope for the CAM Laboratory was to develop a fully integrated system for linking the Applicon Computer Graphics System to a host computer PDP 11/40 Industrial Control Computer for process planning, scheduling, and N/C programming. The control information was then to be down-loaded to individual machines for producing the various parts. Machines acquired for the full-sized CAM Lab included a K & T MM100 CNC milling machine, a Sheldon Model 1710 tape lathe with an Allen-Bradley 7360 CNC control unit, An Eaton- Kenway Automated Storage and Retrieval System (AS/RS) for parts storage, and an ASEA Industrial Robot Model Rb-6 for processing, inspection and assembly.

1.2 PROBLEM

As the work of integration started at BYU, it was soon discovered that two classes of problems existed. First it became apparent that many smaller problems had to be solved in physically linking and logically inter-connecting the various full-sized pieces of equipment. For example, it was discovered that the Bendix Cordax inspection machine could not use the graphics data base for inspecting parts produced on the CNC Milling Machine. The graphics data base did not have tolerances stored for the various dimensions. Also, the graphics data base for the wire frame model did not store relationships between the various form features.

It was soon discovered that the BYU integration problems were not unique. The major Government/Industry Integration effort of I-CAM ran into problems as well. The initial work of the I-CAM project had as its central focus "Database and Database Automation". This central focus was changed after several months to that of developing "generic architecture" models. It is the considered opinion of at least one of the principal investigators of this project that there is not a "generic architecture" which will fit all of the manufacturing companies except at the highest levels of abstraction. The detailed architectural diagrams produced by various contractors have not provided the basis for integration. It was the intent of the Manufacturing Information System research to focus on database and database automation which had been earlier abandoned.

The second class of problems dealt with teaching. There was a pressing challenge of how to teach the various graduate and undergraduate classes dealing with advanced manufacturing technology and systems integration.

AIR FORCE

The full-sized production equipment of the CAM Lab requires a considerable amount of floor space, is expensive to operate and maintain, and because of this it was not feasible to provide multiple work stations. Furthermore, the equipment is a little dangerous to operate with large numbers of student operators and it requires quite a large amount of materials. In order to solve the materials problem styrofoam blocks were used for workpieces. Styrofoam was a good choice since it readily yields in case a student has incorrectly programmed the 20 h.p. milling machine and it suddenly starts moving at rapid traverse rate while the tool is still in the cut. However machining of styrofoam does not present a very realistic exposure to manufacturing problems. Realistic problems such as tool wear, cutting fluid secretion and application, etc. cannot be studied very efficiently when machining styrofoam.

1.3 Proposed Solution

It was postulated that the solution to integration of manufacturing systems is based on the development of a common data base, distributed computing, and local area networking, or in other words, the availability and use of an appropriate manufacturing information system. In order to evaluate this postulate, plans were made to construct a miniature factory and to install a series of microcomputers, databases, and a local area networking system for use in information systems research and teaching. AFOSR Grant No. 82-0253 has greatly assisted in accelerating the development of prototype laboratory equipment for this continuing work.

1.4 Historical Background

The concept of developing a miniature factory had been kicked around since 1976 when one of the principal investigators had visited Caterpillar Tractor Company and had seen some miniature materials handling equipment. The concept of developing a duplicate of the regular full-sized CAM Lab on a miniature scale was first described in a memorandum dated 22 May 1979 from D.K. Allen to J.J. Kunzler. In this memorandum it was mentioned that a Tektronix 4051 Graphics Display System had been donated and had funds to purchase a mini-lathe. Matching college and department funds were requested to purchase a Macsym II Process Control Computer so that interfacing it with the mini-lathe could begin.

Initially the Tektronix 4051 Graphics System had been used to introduce students to concepts of parametric design. With this approach, a part family, based on Group Technology concepts, could be displayed, dimensions added, and the drawing automatically scaled to size. The miniature lathe which was to be interfaced to the Macsym controller, which was to in turn be interfaced to the Tektronix 4051, was actually designed as a camera maker's lathe but had been retrofitted with ball-screws and digital stepper motors to permit its use for numerical control applications. In April, 1979, Superior Electric Company provided a very generous discount for stepping motor controllers to be used in conjunction with the Mini-Lab project.

With initial concepts and all of the equipment in place, Mr. Charles Snead, a graduate student enrolled in the M.S. Degree Program of Computer Aided Manufacturing was asked to begin the integration process by linking the Tektronix Graphics System, Macsym Process Control Computer and miniature lathe. The result was most encouraging. He was able to incorporate parametric design with N/C Cutterpath generation to produce a family of rotational parts having 1, 2, or 3 diameters.

One problem with the Macsym-based system was the very slow positioning rates for the lathe because all of the processing was being performed in BASIC. With this approach it would take nearly twenty minutes to machine a simple part. Subsequently, Mr. Steve Painter of Grady Moore Associates was asked to develop a higher speed stepper motor driver that could be used with the popular and low-cost Apple II Microcomputer instead of using the more expensive and slower approach using the Macsym II Computer. This development of the new controller took nearly 18 months, with one false start, but eventually ended up with a very fast 6-axis controller which could be controlled by the Apple II computer.

In order to expand the integration concept to include other equipment, the BYU Industrial Design Department was contacted in the late spring of 1981 and asked to produce mock-up equipment for use in the Mini-Lab. Professor Allen provided specifications and met with the students many times during the development phase to review and approve their work. On June 22, 1981, an open house was held in which students discussed their projects with members of the local industrial community.

In the fall of 1981, Mr. Forest Blair, a student in mechanical engineering was employed to produce assembly and retail drawings for the new miniature machine tools. Mr. Paul Smith worked closely with the project and with other graduate students who were developing and testing this miniature equipment as part of their thesis work. This work included designing the miniature turret punch, automated storage and retrieval system, and industrial robot. Because of difficulties in designing a milling machine in accordance with the plastic mock-up provided by the industrial design department, Professor Allen developed in May of 1982 an innovative design for a polar-coordinate milling machine. A patent has since been applied for on this design.

It was about this time that the CAM Laboratory was visited by Mr. Thomas Walsh, AFOSR for the purpose of exploring possible mutually beneficial research interests. As a follow up of his visit, an unsolicited proposal was submitted to AFOSR on November 6, 1981 to fund a 24 month grant for continued development of the Mini-Lab and to then use the apparatus so developed in research directed toward development of an integrated Manufacturing Information System. Although funding for the project did not begin until July of 1982, work was going on using internal funding.

The projected Manufacturing Information System research was described as including three phases. Phase I included the development of suitable laboratory apparatus, Phase II included development of software exchange specifications, standards, integration software, and necessary architecture models, and Phase III included subdividing the responsibility for creating applications programs among a number of research institutions.

Some slippage occurred in meeting this planned schedule because the principal investigators did not realize that the project would be funded on an annual basis beginning in July 1982 rather than on a 24-month basis as planned. Subsequently, a continuation proposal was prepared and submitted July 14, 1983. a no-cost extension was approved as was funding for the completion of Phase I of the project.

The following sections describe the mission and goals of the project, work performed to date, deliverables, and a summary and conclusions section.

2. MISSION AND GOALS

The mission of the project was to develop and test scaled-down manufacturing equipment and systems as a basis for manufacturing information system research, manufacturing simulation studies, and manufacturing education related to system integration. Eight goals for accomplishing the stated mission were identified. Only the three goals for Phase I are shown here. Goals for Phases II and III are contained in the original proposal.

Goal 1. Design and develop laboratory-size prototype production equipment for processing and handling of box-like, sheet, and rotational parts made from metallic and non-metallic materials.

Goal 2. Design and develop a small, microprocessor-based, computer numerical control (CNC) system for controlling both processing and handling equipment.

Goal 3. Develop and test the CAD/CAM interface between a small computer graphics display device and the CNC processing and handling system.

For each of the identified goals a series of smaller tasks were identified along with a time schedule for completion of each task.

3. PROJECT OVERVIEW AND DELIVERIES

In accordance with the mission and goals statement given above this section is devoted to a description of the hardware and software designs resulting from this project. The prototype hardware designs are divided into two broad categories: The first category deals with the design and development of laboratory-sized production equipment for fabricating metallic and non-metallic piece parts, and the second category deals with equipment control system hardware, interfaces, and electronic switching.

Software designs have also been divided into two categories: The first category deals with communication and control software required to communicate between the host computer and the equipment controllers themselves, and the second category deals with application software to demonstrate the feasibility of using parametric design concepts, and automatic cutterpath generation techniques in a "paperless factory" environment. The use of parametric techniques for defining individual pieceparts has been quite extensively used at BYU and is now being quite widely espoused. Some of the underlying work for parametric design is based on the Part Family Classification which contains some 240 basic shapes composed of primitive shapes and some 80 form features. With this system it is possible to generate literally millions of combinations of geometric configurations which can then be used in generative design, family of part programming, tool design, estimating, etc.

The overall architecture for the integrated manufacturing system is shown in Fig. 1. In this system, IBM PC's are used for design and scheduling, while Apple II microcomputers are used for machine controllers in conjunction with stepper motor drivers to actually position each of the machines.

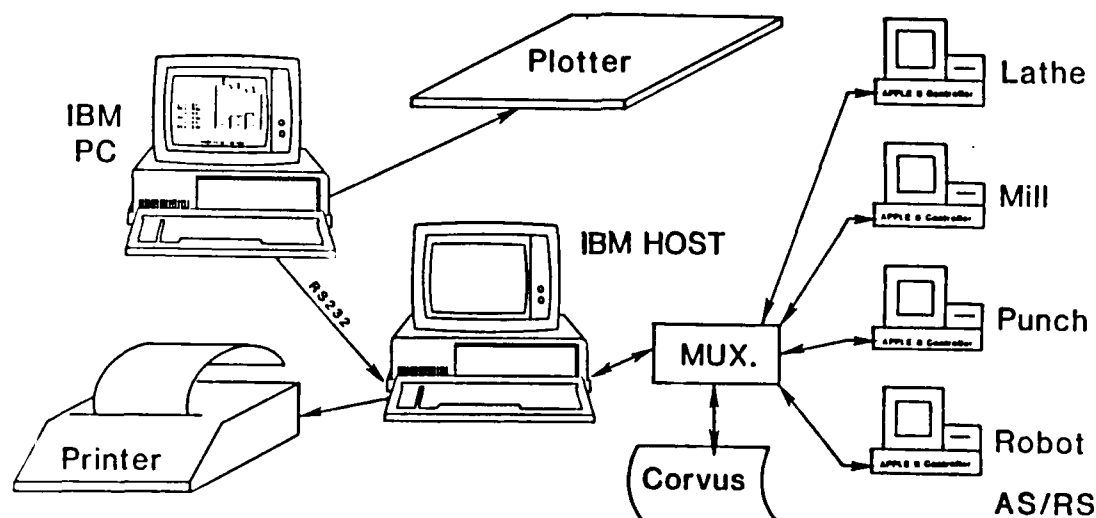


Fig. 1 - Architecture for Integrated Manufacturing Information System

3.1 Design of Laboratory Sized Production Equipment

Designs for each of the following prototype equipments are described below along with specifications, photographs and description. Detailed working drawings for machines produced under this grant and from other funding sources are provided in the appendices.

3.1.1 Automated Storage and Retrieval System (AS/RS)

Automated materials handling is one of the major components of an integrated factory. Information flow often parallels and preceeds material flow, thus design of the material handling system provides a basis for also designing the Manufacturing Information System.

General design specifications for the AS/RS are as follows:

- a) Bin capacity 2 lbs
- b) Bin size 3" x 4" x 8" long
- c) No. bins 18
- d) Traverse rate 100"/min

The AS/RS, shown in Fig. 2, is approximately 55" long and 23" high. It is equipped with ball screws and three slo-syn digital stepper-motor drives for positioning and bin retraction.

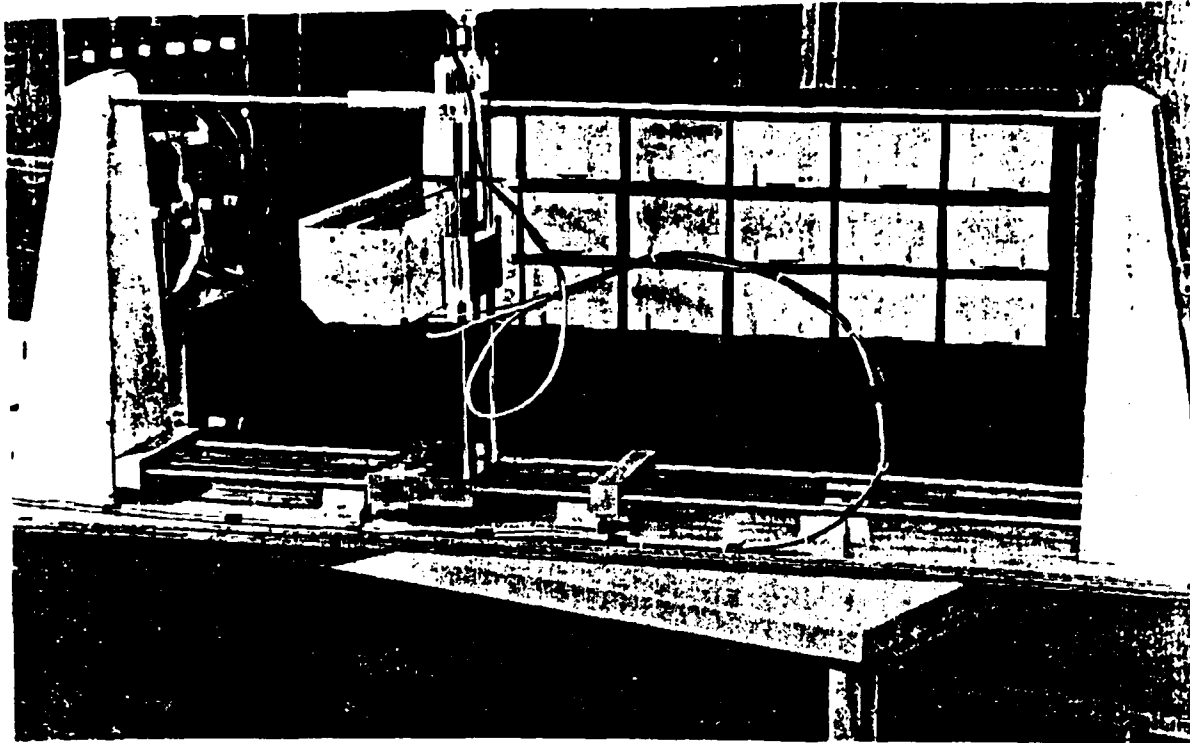


Fig. 2 - Miniature Automated Storage/Retrieval System

The AS/RS unit has 18 bins which can be used for storage of raw materials, in-process inventory, finished goods, or tooling. Box-like materials are staggered in height to facilitate removal. Rod-like materials are supported on an inclined plane so they roll into position to be gripped by the robot. Sheet stock is stored flat, and requires the development of a vacuum gripper mechanism to aid in its removal. Detailed design drawings for the miniature AS/RS are contained in Appendix A.

3.1.2 Miniature Sheet-metal Turret Punch

One of the important "wedges" in the I-CAM project was the development of a sheet-metal cell. Since there are a variety of expert opinions as to what should constitute such a cell, the intent of our project was to develop one standard miniature piece of equipment which would appear in most full-sized cells and which would provide the basis for information flow to a sheet metal center as well as provide a system which could be used for modeling and simulation of the sheet metal production process.

General design specifications for the turret punch are as follows:

- | | |
|------------------------------|-------------|
| a) Sheet stock size | 4" x 8" |
| b) Material paper card stock | .020" thick |
| c) Turret capacity | 8 tools |
| d) Punching capacity | 1/4" dia. |
| e) Positioning rate | 0-50"/min. |

The turret punch, shown in Fig. 3, can be used to produce holes, cut-outs, and re-entrants of various sizes depending upon which tools are in the turret. Some additional work must still be done to increase punching capacity and punch alignment.

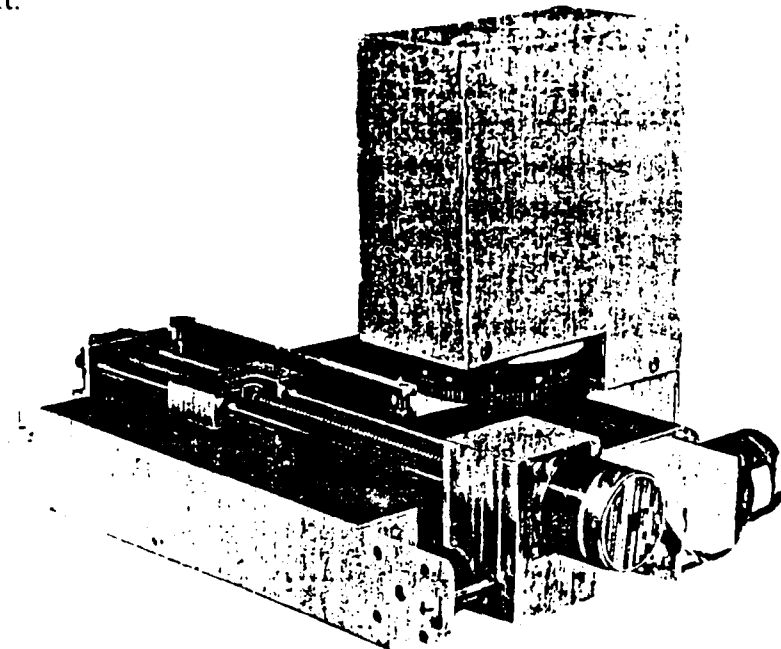


Fig. 3 - Miniature Sheet-metal Turret Punch

Detailed designs for the miniature turret punch are contained in Appendix B.

3.1.3 Milling Machine

Prior to the start of the AFOSR grant, a small horizontal spindle drilling machine had been acquired and outfitted with ball-screws and stepper motors to perform the functions of a milling machine. However, the spindle was too heavy to be elevated by the stepper-motor drives. Consequently, plans were made to design another horizontal mill and to make as many components as possible interchangeable with the new lathe to be designed. In the spring of 1981, the CAM Lab had also been approached by Digital Equipment Co. Industrial Design Department to develop a small milling machine to be connected to a graphic workstation for producing 3-dimensional sculptured surfaces. Rather than design a second horizontal spindle milling machine specifically for the Manufacturing Information System project, it was decided to use the design which had been made to satisfy the needs of Digital Equipment Company. This design, called the polar-coordinate mill, while especially suited for producing sculptured and contoured surfaces, seemed to be viable for use with the project at hand. Since the design was already available for the polar coordinate mill this approach provided a way to save some valuable development time.

The original design specifications for a horizontal milling machine have been met with the polar-coordinate milling machine with the exception of the spindle thread and the four position indexable turret. The available spindle had only a 1/4" dia. straight shaft. The indexable turret now has 360° of rotation instead of just four positions called for in the original specifications.

The design specifications for the milling machine are as follows:

- a) Size capacity 4" cube
- b) Drill motor 1/10 HP
- c) Spindle speeds 0-2000 rpm
- d) Positioning rate 0-50 in/min
- e) Resolution 0.005 in/step
- f) Materials, styrofoam, machinable wax, etc.

The polar-coordinate milling machine is shown in Figure 4. The machine utilizes a over-arm design in which adjustable track rollers ride in a hardened and ground V-groove. The overarm can sweep 180 degrees across the top of the object to be machined and at the same time the workpiece, mounted on the rotary table can be simultaneously rotated and traversed past the cutter. With this design it is now possible to machine five sides of a workpiece without refixturing it for each surface.

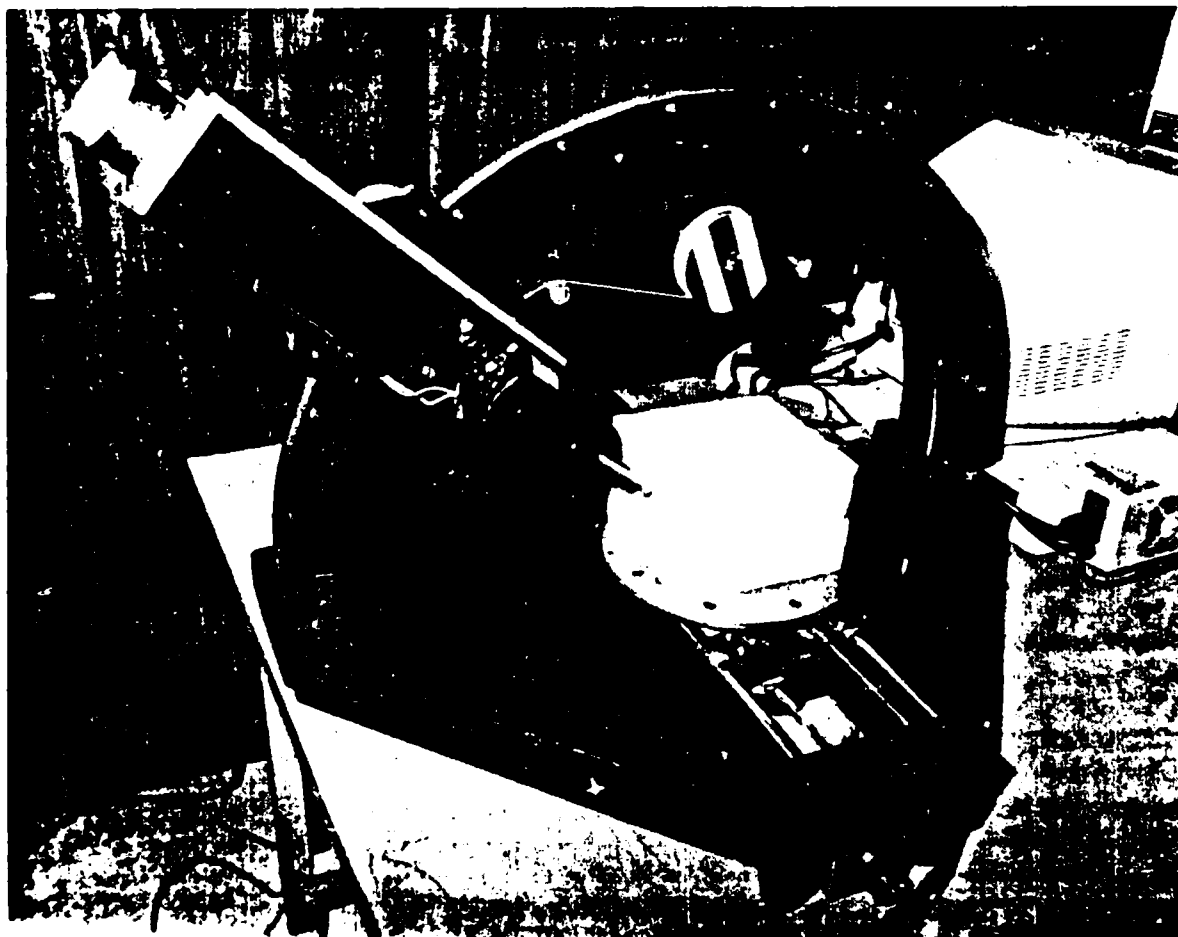


Fig. 4 - Polar-coordinate Milling Machine

Detailed designs for the polar-coordinate milling machine are contained in Appendix C.

3.1.4 CNC Turret Lathe

The original specifications for a computer controlled lathe called for: (a) 2" swing, (b) 6" between centers, (c) 1/10 H.P. spindle motor, 0-200 rpm, (d) positioning rate of 0-50"/minute, (e) positioning resolution 0.0005"/step, and a spindle with 5/8-11 UNC threads. The machine was to be equipped with 1/4" square shank tools, to be mounted in a four-station turret for machining aluminum, brass, plastic, wood and other soft materials.

At the beginning of the project, miniature working equipment was essentially unavailable for laboratory use and thus had to be designed and constructed from the ground up. The detailed designs for the miniature lathe are contained in Appendix D. However, a few miniature lathes have now appeared on the market which satisfy or exceed the original design specifications. By combining resources from several funding sources it was possible to purchase a commercially available miniature lathe and associated tooling rather than go through the development process. A photograph of the commercially available CNC turret lathe acquired for this project is shown in Figure 5.

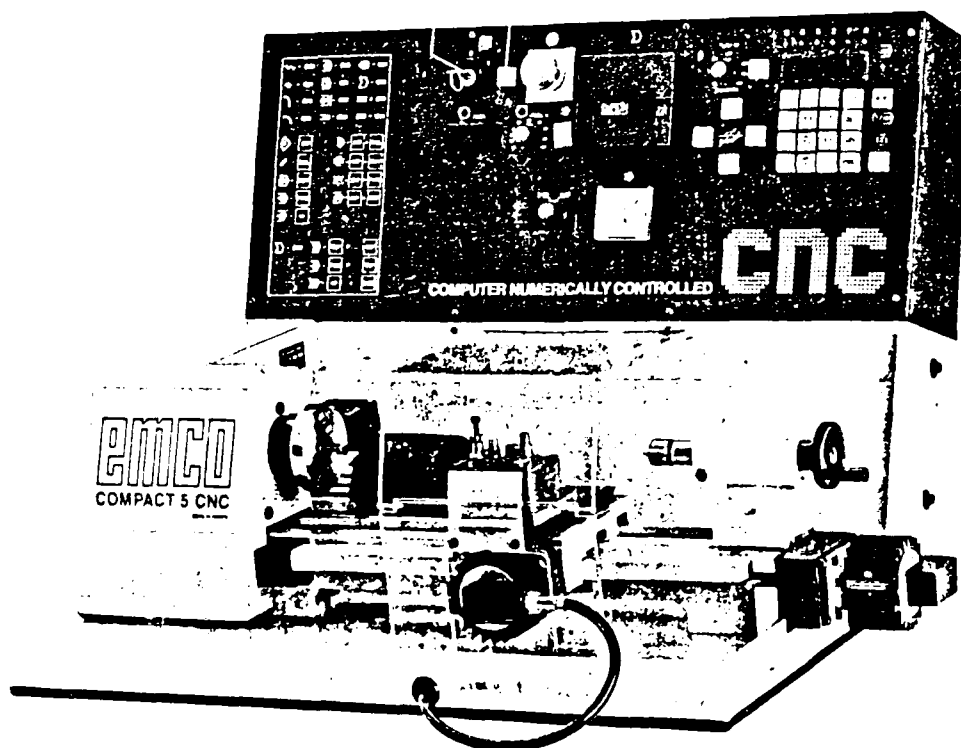
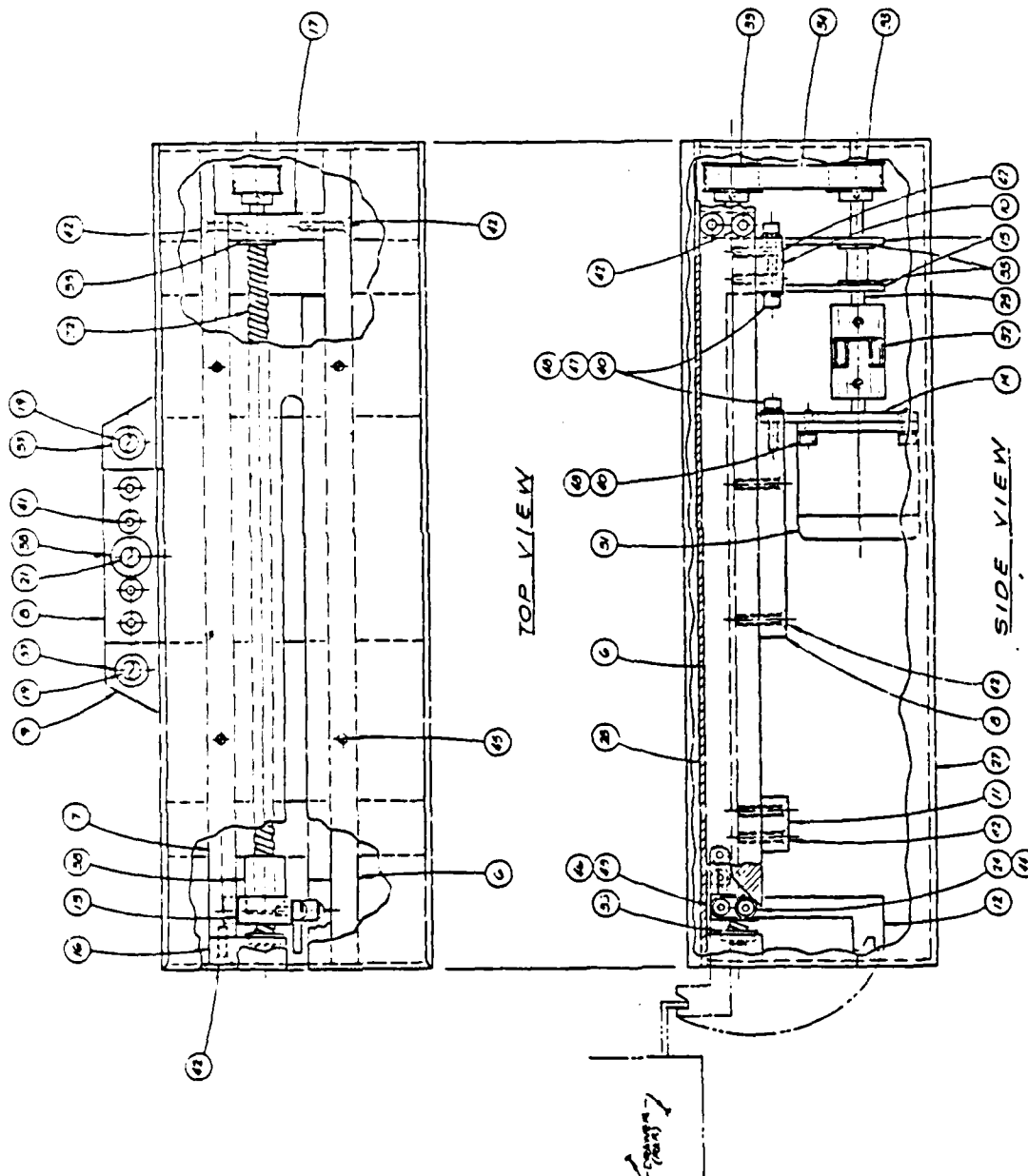


Fig. 5 - CNC Turret Lathe

The lathe does not quite have the resolution desired but it does more than compensate in other ways. For example the main drive motor is four times as powerful and it also was a six-station indexable turret instead of the four

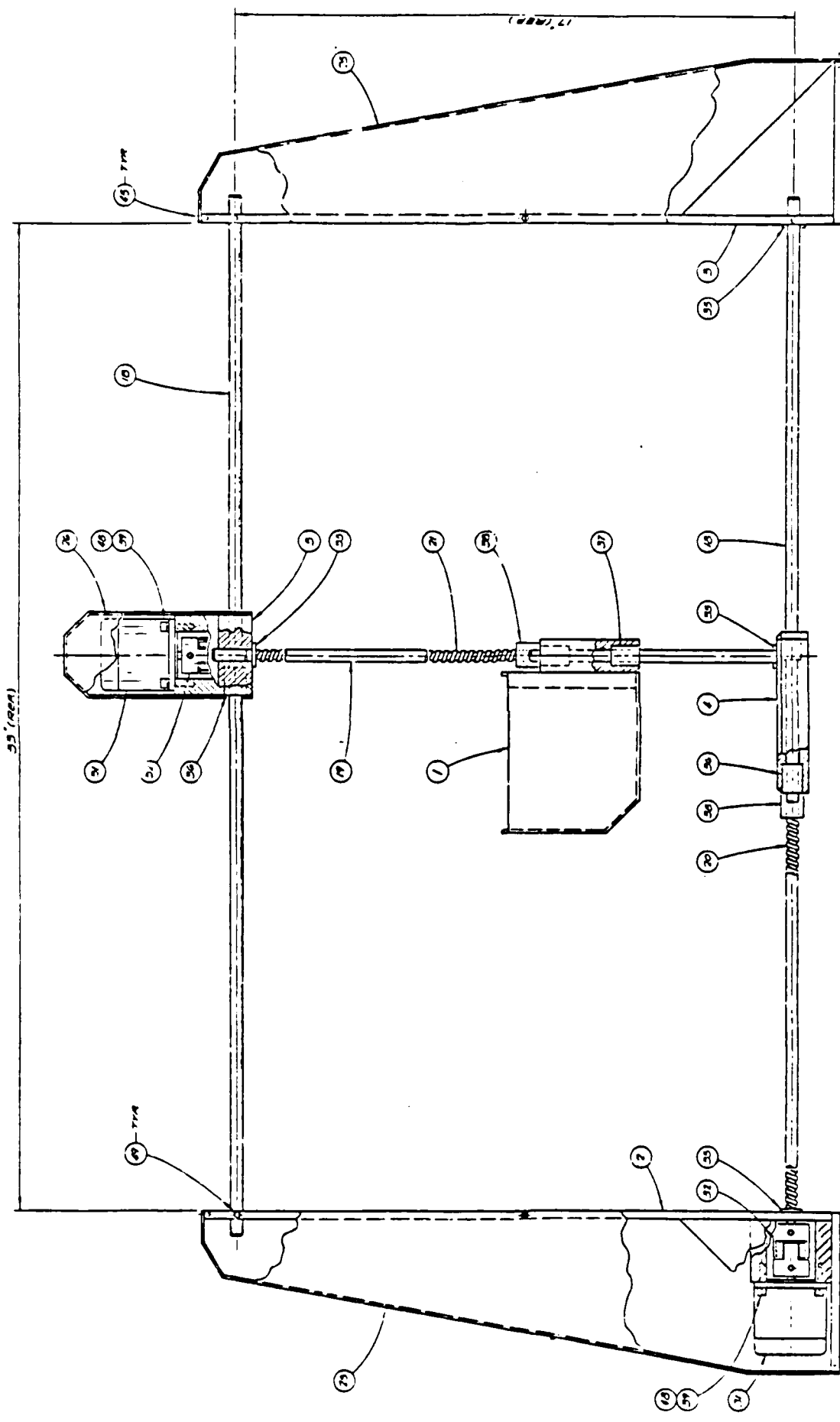


NOTES
 1 ALL ITEMS MARKED WITH
 ARE ACCOUNT ON ONE OF
 MATERIALS IN B MOOD.

DETAIL ITEM 1
 SCALE: FULL

SEPTEMBER 22, 1961	BRIGHAM YOUNG UNIVERSITY	MINI-STACKER	SUB-ASSEMBLY - DETAIL ITEM 1	FOREST CLIP	CC11
CAD-CAM DEPT	BRIGHAM YOUNG UNIVERSITY	MINI-STACKER	SUB-ASSEMBLY - DETAIL ITEM 1	FOREST CLIP	CC11

35" (approx)



FRONT ELEVATION
SLACKER-WELL

NOTES
ALL DIMENSIONS ARE TO CENTER
OF RODS UNLESS OTHERWISE SPECIFIED
UNLESS OTHERWISE SPECIFIED

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY		Project No.		Drawing No.		Sheet	
Designed by FORREST BLAIR		10-15-31		NO		6 of	
Checked by		Date		REVISIONS		DATE	
Project Engineer		Date		DESCRIPTION			
Report Name CAD-CAM		Date					
Drawing Title INL-STACKER		Date					
TBM DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE			
43 0011	Machine Screw - Slotted	1	C. Steel				
	Binding Head, #9-32 UNC x 1/9" Lg.						
44 0010	Machine Screw - Slotted Binding	1	C. Steel				
	Head, #8-32 UNC x 1/2" Long						
45 0010 & 11	Machine Screw - Slotted Flat	19	C. Steel				
	Head, #5-40 UNC x 1/4" Long						
46 0011	Washer - Flat, #8	1	C. Steel				
47 0011	Washer - Flat, #10	6	C. Steel				
48 0010 & 11	Washer - Lock, #10	18	C. Steel				
49 0010	Setscrew, #10	10	C. Steel				
50 0010	Bearing-1200 1, Single Shield,	1	Stock Drive Rod.				
	5/16" O.D., 1" Bore, Part No.						
	7855-156225						
51 0010	Bearing-Thrust, 1" Bore,	1	12. Std. Stock Drive Rod.				
	Part No. 2-7-025						

DRAWING MAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
FORREST BLAIR				10-15-41	10010	4 of 4
Prepared by				REVISIONS		
Approved by				NO	DESCRIPTION	DATE
Project Engineer						
Project Name						
CAD-CAM						
Drawing Title						
MINI-STACKER						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
30		Not Used				
31	0010 & 11	Motor-Stepping, Type SS25	3	Mfg. Std. SIO-SYN		
32	0010 & 11	Coupling-1/4" Nom. Bore, Model No. LC50	3	Mfg. Std. Love Joy		
33	0011	Pulley-Timing Belt, 1/5" Pitch, Planged for 3/8" Belt, .764 P.D., 1/4" Bore, Cat. No. 6A1-12DF03708	2	Mfg. Std. Stock Drive Prod.		
34	0011	Belt-Timing, 1/5" Pitch, 3/3" Width, 35 Grooves, 7.000" Pitch Length, Cat. No. 6G1-15-037	1	Mfg. Std. Stock Drive Prod.		
35	0010 & 11	Bearing-ABEC 3, Flanged, Single Shield, 5/8" O.D., 1/4" Bore, Part No. 7Y55-FS6225	2	Mfg. Std. Stock Drive Prod.		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
FORREST BLAIR				10-15-21	10010	5 of 5
Prepared by				REVISIONS		
Approved by				NO	DESCRIPTION	DATE
Project Engineer						
Project Name						
CAD-CAM						
MINI-STACKER						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
36	0010	Bearing-Linear, 7/8" O.D. x 1 1/4" Long, 1/2" Bore, Cat. No. A-81420	5	Mfg. Std. Thompson		
37	0010 & 11	Bearing-Linear, 5/8" O.D. x 7/8" Long, 3/8" Bore, Cat. No. A-61014	4	Mfg. Std. Thompson		
38	0010 & 11	Ball Nut-For 3/8" Nom. Screw, Cat. No. RM-0375-0125	3	Mfg. Std. Paceaway		
39	0010	Cap Screw - Hex Socket, #10-24 UNC x 1" Long	3	C. Steel		
40	0011	Cap Screw - Hex Socket, #10-24 UNC x 3/4" Long	10	C. Steel		
41	0011	Cap Screw - Flathead Hex Socket, #10-24 UNC x 1" Long	4	C. Steel		
42	0011	Cap Screw - Flathead Hex Socket, #10-24 UNC x 3/4" Long	19	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				Date 10-15-41	NO	2 of
Approved by				Date	REVISIONS DESCRIPTION	DATE
Project Engineer				Date		
Project Name CAD-CAM				Date		
Drawing Title MINI-STACKER				Date		
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
10	0014	Rear Rail Block	1	Aluminum		
11	0014	Front Rail Block	1	Aluminum		
12	0014	Retrieval Hook	1	Aluminum		
13	0014	Hook Attachment Block	1	Aluminum		
14	0015	Motor Support Plate	1	Aluminum		
15	0015	Drive Shaft Support Plate	2	Aluminum		
16	0015	Front Screw Brg. Mount	1	Aluminum		
17	0015	Rear Screw Brg. Mount	1	Aluminum		
18	0016	Horizontal Support Rod	1	C. Steel		
19	0016	Vertical Support Rod	2	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				Date 10-15-41	NO	3 of
Approved by				Date	REVISIONS DESCRIPTION	DATE
Project Engineer				Date		
Project Name CAD-CAM				Date		
Drawing Title MINI-STACKER				Date		
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
20	0016	Horizontal Drive Screw	1	C. Steel		
21	0016	Vertical Drive Screw	1	C. Steel		
22	0016	Retrieval Mechanism Drive Screw	1	C. Steel		
23	0016	Retrieval Mechanism Drive Shaft	1	C. Steel		
24	0016	Roller	1	C. Steel		
25	0017	Horizontal Support Cover	2	Acrylic or Equal		
26	0017	Vertical Drive Mount Cover	1	Acrylic or Equal		
27	0018	Retrieval Mechanism Bottom Cover	1	Acrylic or Equal		
28	0018	Retrieval Mechanism Top Cover	1	Acrylic or Equal		
29		Not Used				

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT

APPENDIX A

Detailed Design for Miniature Automated Storage and Retrieval System

3.3.3 CAD/CAM Demonstration Software

A demonstration program was created using the DCLASS System to provide a menu to design a particular part to be produced on the lathe and then transferring this information by means of a parametric program to the NC cutter path generation routine. Following this transfer of data a series of machine motion commands were executed to produce a miniature part as described by the geometry on the CAD system. This demonstration was also extended to generate cutter path moves on the milling machine as well as to exercise the storage and retrieval system to retrieve various materials from the bins.

The demonstration software was used to verify the fact that the machines performed as desired and that the machine controllers were adequate. Documentation and program listings for this software is contained in Appendix M. Some of this work represents the first phases of the integration which is to be conducted under subsequent research activities.

4.0 SUMMARY, And CONCLUSIONS

This development project was successful in designing, constructing, and testing miniature prototype laboratory equipment for use in undertaking a series of investigations relative to information systems for manufacturing. It is believed, based on the results of this development project, that miniature machine tools can be used to provide a viable alternative to full-sized equipment for use in research and teaching. The miniature equipment with its IBM personal computers and Apple II microcomputers when linked with the CORVUS local area networking system provided the basis for some very preliminary work in developing a fully integrated Manufacturing Information System.

The funding provided was only sufficient to develop prototype equipment, and additional work now needs to be undertaken to enhance this equipment and to use it for more fully investigating requirements of a Manufacturing Information System. It is believed that these requirements include common database development, further exploration of Local Area Networking, and additional work on defining parametric designs, as well as developing scheduling and control algorithms for the miniature factory.

While there has only been a limited amount of contract AFOSR and Brigham Young University the principal investigators and research staff have tried diligently to live up to the original grant proposal.

Additional funding is now being sought from other agencies and industrial sponsors to continue with the development of this miniature factory to fully exploit the possibilities of developing transportable Manufacturing Information Systems which can be implemented in full-sized factories.

The principal investigators and research staff wish to extend their appreciation to AFOSR for support of this grant.

The Engineering Design Terminal consists of an IBM PC-XT with 10-mb hard disc, 256K memory, parallel and serial ports for communication to other computers in the system and to the combination printer/plotter.

3.3 Communications and Control Software

A variety of software had to be written for use with the Apple II process control computers, the digital stepping motor drivers, hand shaking with the host computer, and demonstration CAD/CAM software to test system communication capabilities.

3.3.1 Machine Controller Software

Each of the miniature machines has several axis of motion ranging from two axis on the lathe to four axis on the polar-coordinate milling machine. Each of these axis is positioned by means of slo-syn digital stepping motors. As many as six of these motors can be controlled from one stepping motor controller interface box. Since each machine is controlled by the Apple II process control computer it is necessary for this computer to send command to each machine via the CY512 controller chip. In each Apple II microcomputer process controller is a Z80 microprocessor. This Z80 microprocessor has been programmed by Prof Smart to use assembly language to pass data from the Apple II process control system to the step motor controller chips located in the step motor controller box. Schematics and assembly code listings for the machine controller software is contained in Appendix G.

3.3.2 Communication Software

A variety of communication packages were created to provide menus for the machine control programs and then transferring these files from the Apple II process control computer to the machine control unit. Both manual and automatic control features were developed. The sequence of program control includes creating a text file containing the NC machine code file with traditional NC commands using N, X, Y, Z, and F functions to represent line number, X coordinate, Y coordinate, Z coordinate and feed rate respectively. The next piece of software was made to interpret and run this command file using the CP/M operating system. The machine language interface and communication program listing is contained in Appendix H. In Appendix I are contained the operating instructions for running the miniature lathe and the miniature milling machine. Appendix J contains a series of CNC programs for the miniature lathe and milling machine. A program written in Applesoft Basic for manually controlling the Turret Punch is contained in Appendix K. A variety of programs used to control the Storage and Retrieval system with the IBM PC via the Apple computer are contained in Appendix M. The first program called STACKER.BAS runs on the IBM PC and communicates with the Apple through the serial interface. The second program is also called STACKER.BAS but this one runs on the Apple computer and accepts commands from the IBM PC to produce the desired moves in the miniature AS/RS system.

The computer has a 10-MB Winchester Disk, and 256K memory. It is used to generate process plans based on pre-stored decision tree logic. This terminal is connected to the engineering design terminal via a multiport controller (right) from which it receives part family codes and part parameter values. As mentioned above, these codes and values are then used down-stream by the process control computer to generate machine motion commands.

3.2.4 Engineering Design Terminal

The engineering design terminal shown in Fig. 10 provides both a monochrome menu display and a color graphics display. A three-dimensional graphics software package and tablet/arm have been ordered using university funds for use with this equipment. Composite parametric designs based on group technology principles have been implemented with this system using a new version of the popular BYU Part Family Classification System.

It is expected in a subsequent design of the overall system that other intelligent terminals will be added to perform functions of bill of material processing, master scheduling, market forecasting, and management spread-sheet analysis.

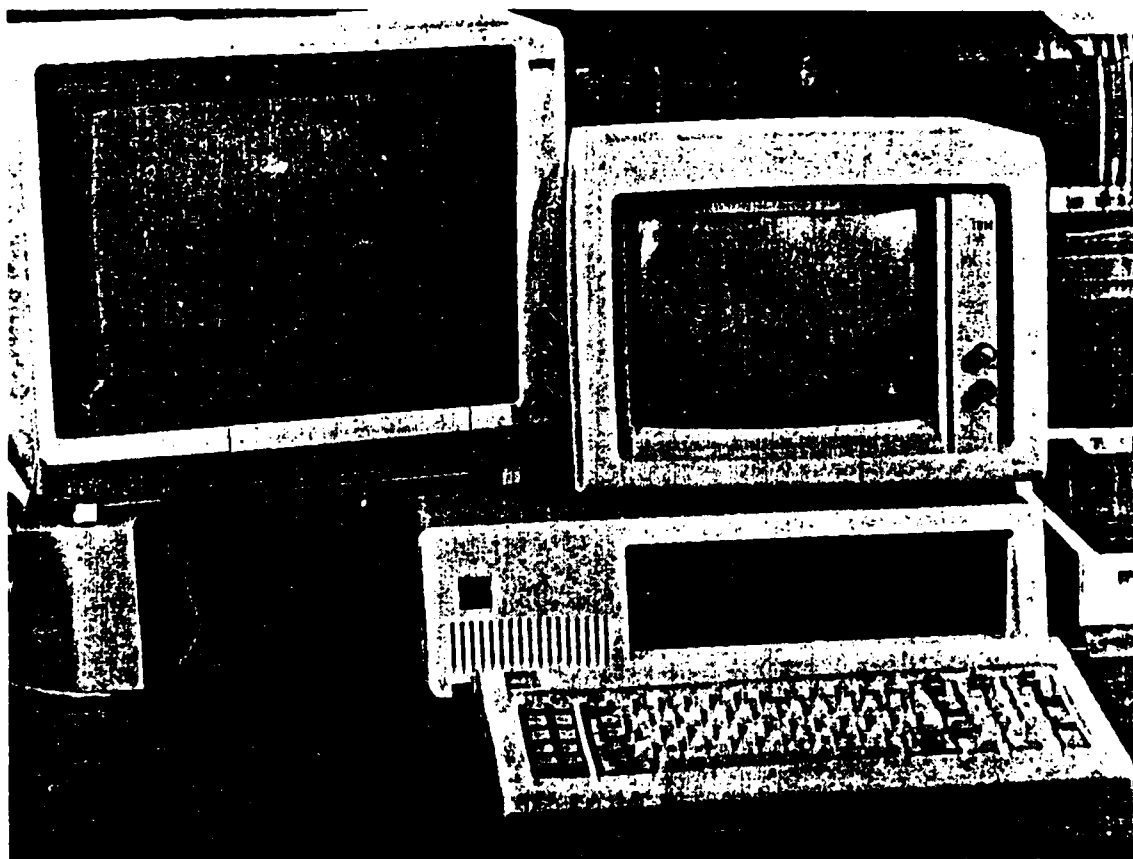


Fig. 10 - Engineering Design Terminal with Monochrome and Color Graphics Display

3.2.2 Computer Controlled Power Switch

A special switch was designed and constructed by Prof Smart to permit the stepper motor control to activate an AC motor upon request by utilizing an optically coupled triac circuit. For example, when a spindle motor is to be turned on, the control program sends a command to the CY512, Stepper Motor Controller device. The CY512 accepts the command and sets a logic bit at one of its output pins. This bit activates the optically coupled triac power switch and the motor turns on. Each computer controlled power switch is capable of controlling three 115 volt AC circuits of up to 12 amps each. Documentation and schematics for the power switch is contained in Appendix F.

3.2.3 Host Computer System

In the present configuration the host computer shown in the schematic of Fig. 1 would actually be serving a dual function. It would be used for process planning and scheduling as well as performing the functions of the host computer. The host computer functions involve assembling data required for production, hand-shaking with the process control computers, and down-loading parametric data to be used in machine control and cutter-path generation. A photograph of the IBM PC-XT host computer used in the miniature laboratory as part of the manufacturing information system is shown in Fig. 9.

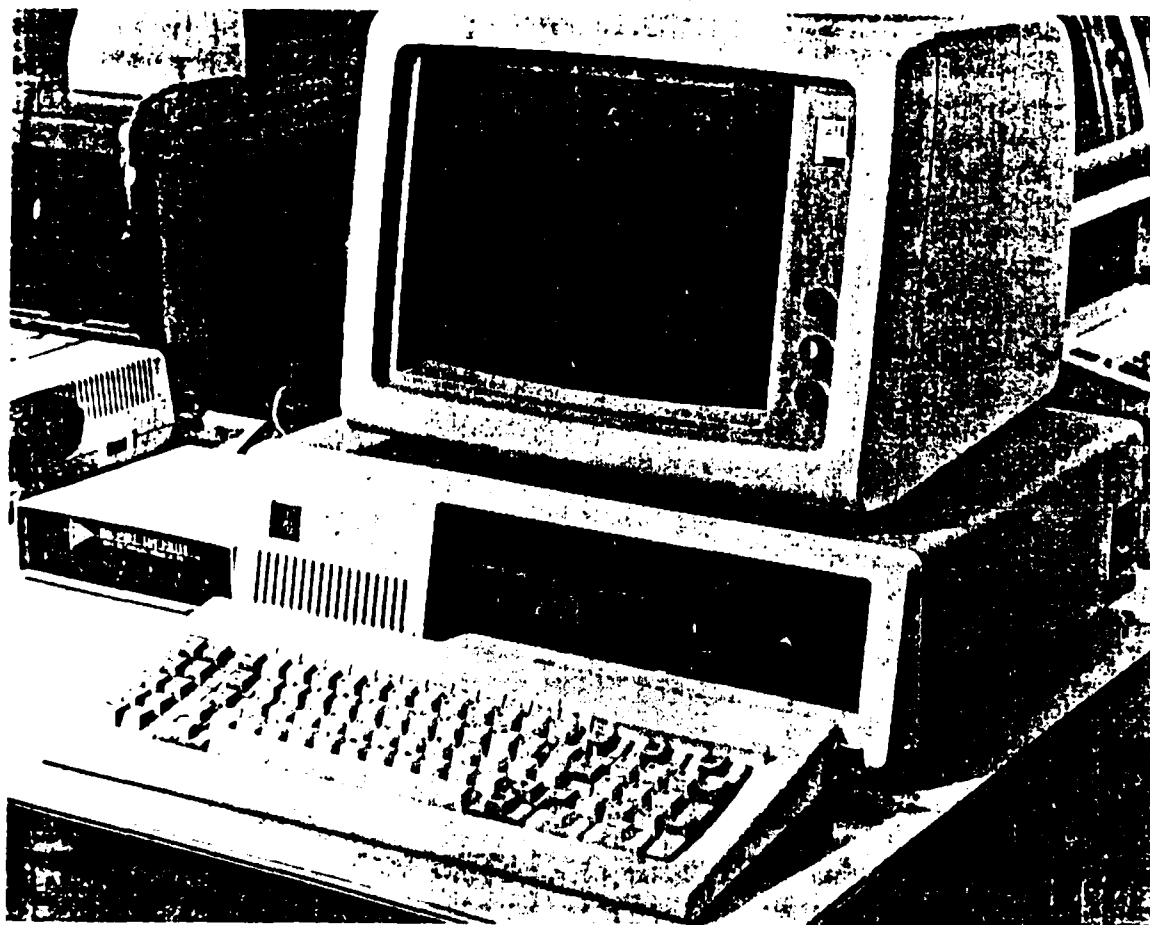


Fig. 9 - IBM PC-XT Host Computer System

Under the present configuration, one process control computer is used for controlling each machine. Each Apple computer was configured with 48K or main memory, 64-K ROM cards, and Microsoft Softcards to facilitate running the CP/M operating system, and a serial interface for communication with the IBM PC-XT host computer.

One of the six-axis stepper motor controllers with integral power supply is shown in Fig. 8. This unit is interfaced directly to the Apple II microcomputer process control system by means of a parallel interface cable.

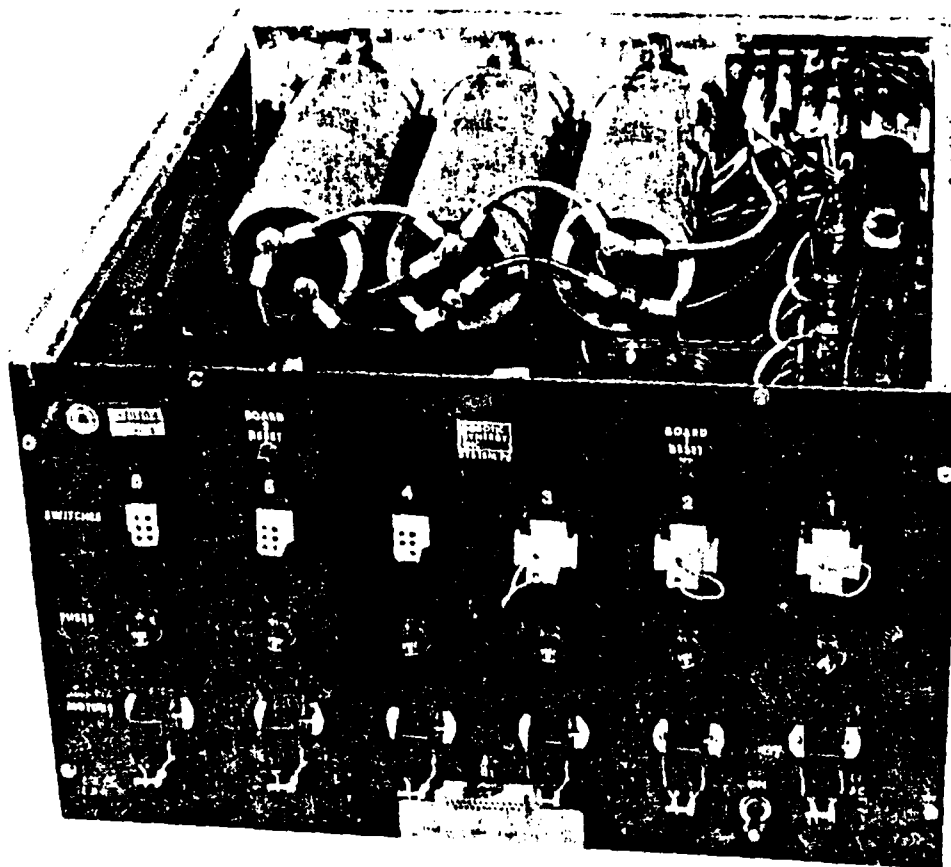


Fig. 8 - Six-axis Stepper Motor Controller and Power Supply

The stepper motor controller was built by Robotic Synergy of Salt Lake City. This controller contains the popular CY512 stepper motor control circuit. This circuit provides automatic acceleration and deceleration, allows various slewing rates and provides synchronization input and output signals.

Some work still remains to be done on the robot manipulator to bring it to the desired level of positioning accuracy and to develop required interchangeable end-effectors needed to handle a variety of materials.

Some work was done in investigating various types of feed-back and adaptive control systems such as could be used with either stationary or mobile robots, but because of problems in getting the robot operational, these features have not been incorporated.

3.2 Computing and Control System Hardware

Designs were made and prototype equipment was fabricated or purchased to: (a) control the stepper motors used for positioning each of the various machine axes, (b) remotely control spindle drive motors, and (c) to perform required computations for design, planning, scheduling, and transfer of data between the host computers and process control computers.

3.2.1 Equipment Controllers

Each of the miniature machine tools is positioned by means of digital stepper motor drives attached to ball screws or to special cogged belts. These drives are in turn controlled by means of stepper motor drivers which act under direction of Apple II microcomputers which serve the function of process control computers.

One of the Apple II microcomputers is shown in Fig. 7 along with the Corvus winchester shared-disc system.



Fig. 7 - Process Control Computer and Shared Winchester Disc

station turret. An attempt was made to purchase the machine without a control unit but to no avail. The controller will be by-passed or removed as it is integrated into the manufacturing information system.

3.1.5 Industrial Robot Manipulator

As with the lathe, there was not a viable miniature robot available at the start of the project. Consequently, designs were made and a prototype unit was constructed. Original specifications for the five axis robot manipulator included:

- a) Lifting capacity 1 lb.
- b) Range 8" x 17"
- c) Positioning accuracy $\pm .005"$
- d) Positioning rate 200 in/min

The robot is shown in Fig. 6 alongside other equipment to be used in developing the integrated manufacturing system and evaluating system performance under a range of operating conditions.

Detailed designs for the Industrial Robot Manipulator are contained in Appendix E.

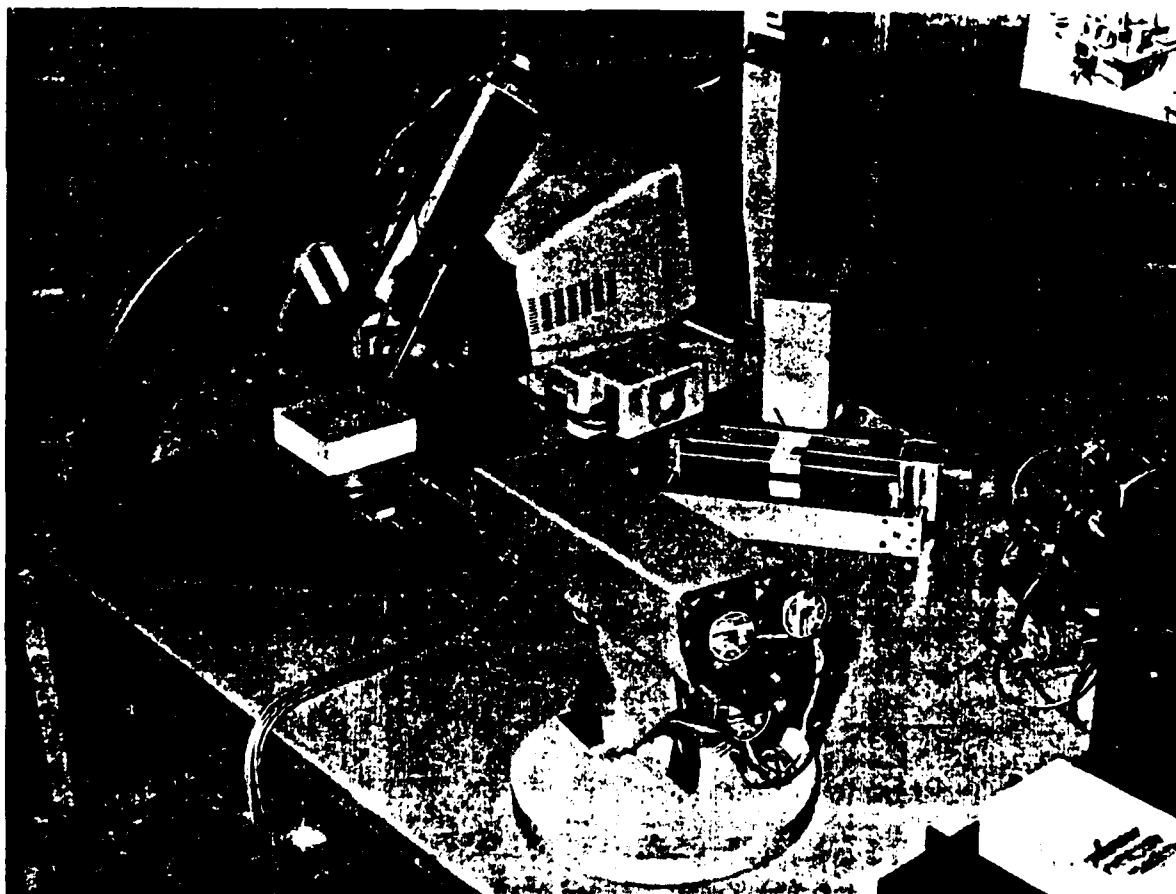


Fig. 6 - Industrial Robot Manipulator (Foreground)

NOTES

1. WORK THIS DRWG. WITH BILL OF MATERIALS MOODIO.
2. ALL FRACTIONAL DIMENSIONS TO BE (±) 1/32".
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



1. WORK THIS DRG. WITH BILL OF MATERIALS "MCOIO."
2. ALL FRACTIONAL DIMENSIONS TO BE (2) $\frac{1}{16}$.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



SCALE: 5/1-6

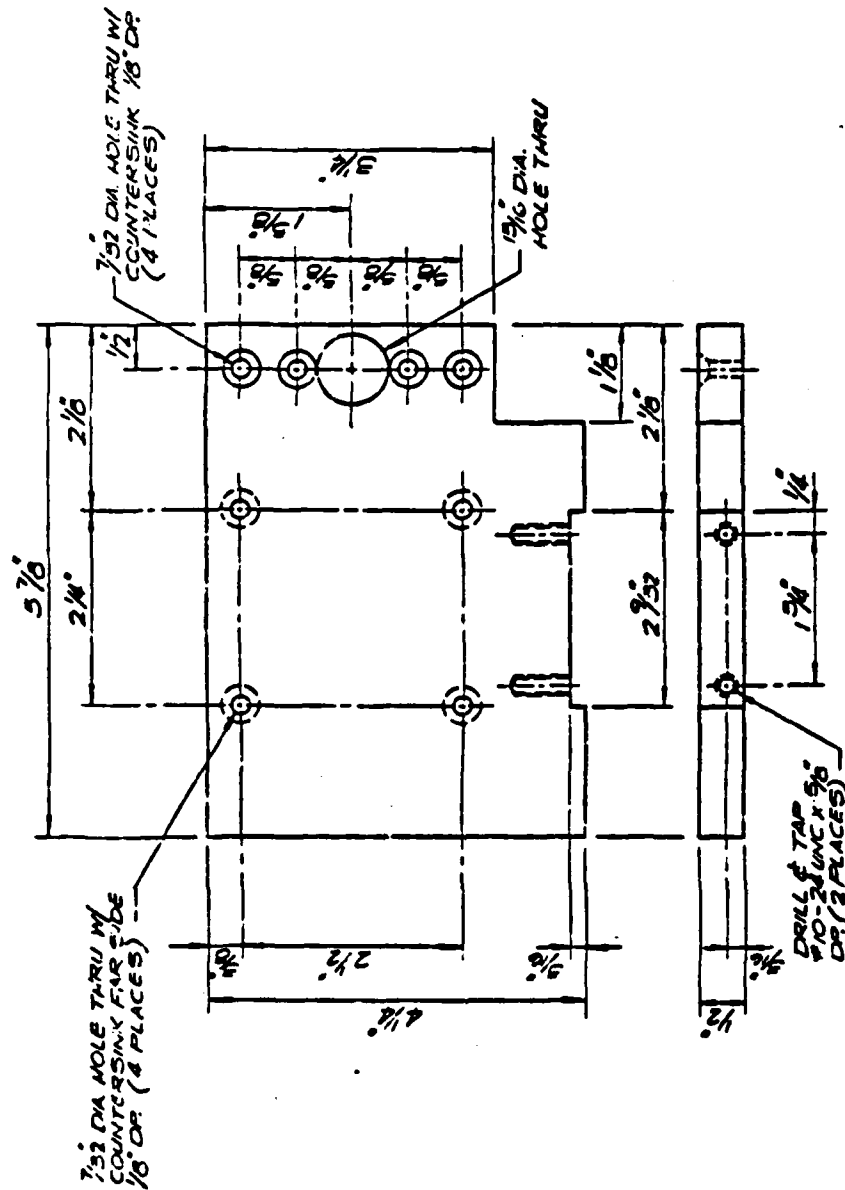
NOTES

1. WORK THIS DRAW. WITH BILL OF MATERIALS. 110010.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) YET.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



SCALE: FULL

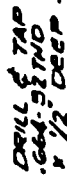
- NOTES
1. CHECK THIS DWG. WITH BILL OF MATERIALS MODIO & ENG. COIL.
 2. ALL FRACTIONAL DIMENSIONS TO BE (2), 1/64.
 3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 8 (ALUMINUM)
SCALE: FULL

NOTES

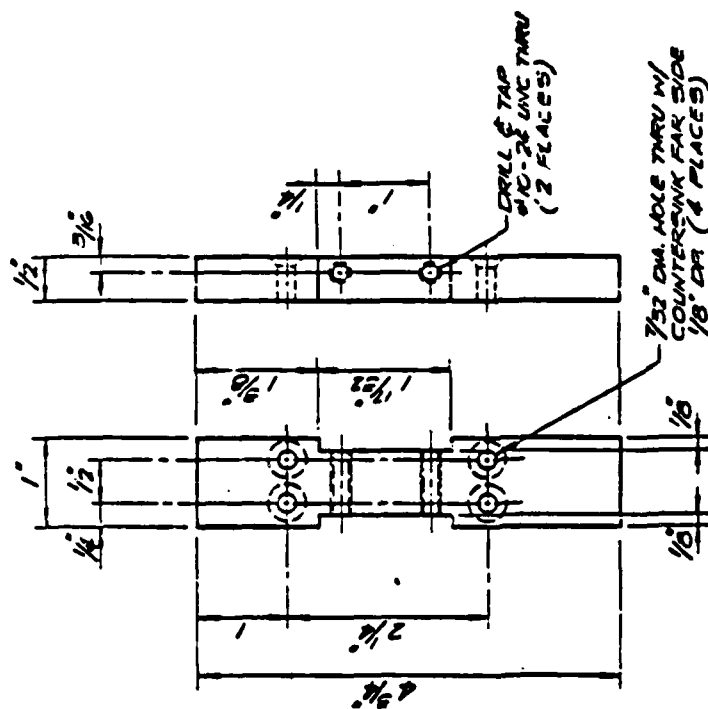
- 1 WORK THIS DWG. WITH BILL OF MATERIALS MOODLO & DWG 0011.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) VCS.
- 3 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



9

NOTES

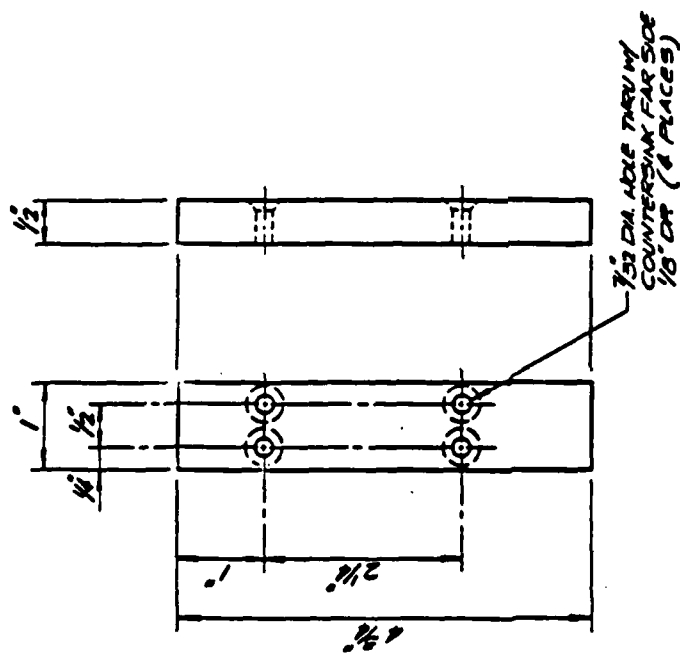
1. WORK THIS DRG. WITH BILL OF MATERIALS MODIO & DRG. 0011.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/4".
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 10 (ALUMINUM)
SCALE: FULL

NOTES

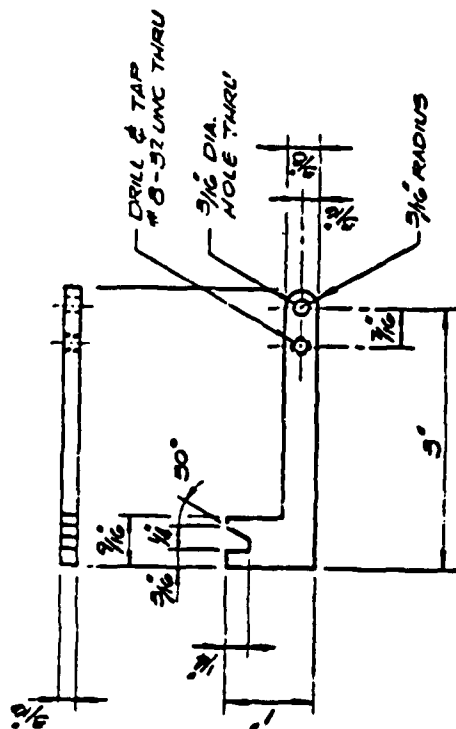
- 1 WORK THIS DWG. WITH BILL OF
- 2 MATERIALS MOOIO CDWG COII.
- 3 ALL FRACTIONAL DIMENSIONS
- TO BE IN 1/64".
- 4 ALL HOLE SIZES SHALL BE
- ACTUAL DRILL SIZE UNLESS
- SHOWN OTHERWISE.



DETAIL ITEM 11 (ALUMINUM)
SCALE: FULL

NOTES

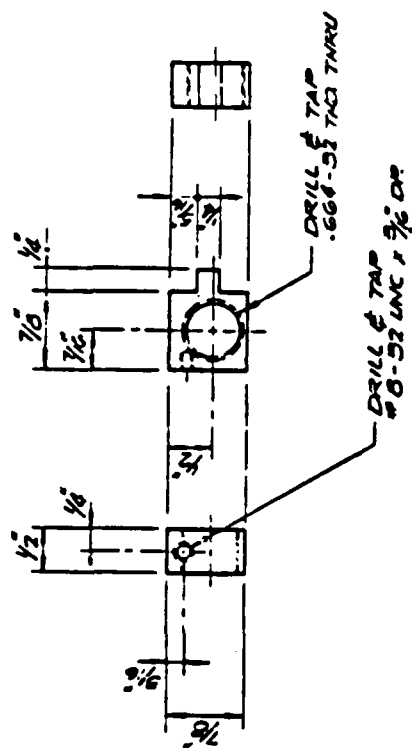
- 1 WORK THIS DWG. WITH BILL OF MATERIALS MOOD & ENG COIL.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/4".
- 3 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 12 (ALUMINUM)
SCALE: FULL

NOTES

1. WORK THIS Dwg WITH BILL OF MATERIALS AND/OR CLNC DCL.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 1.3
SCALE: FULL
(ALUMINUM)

NOTES:

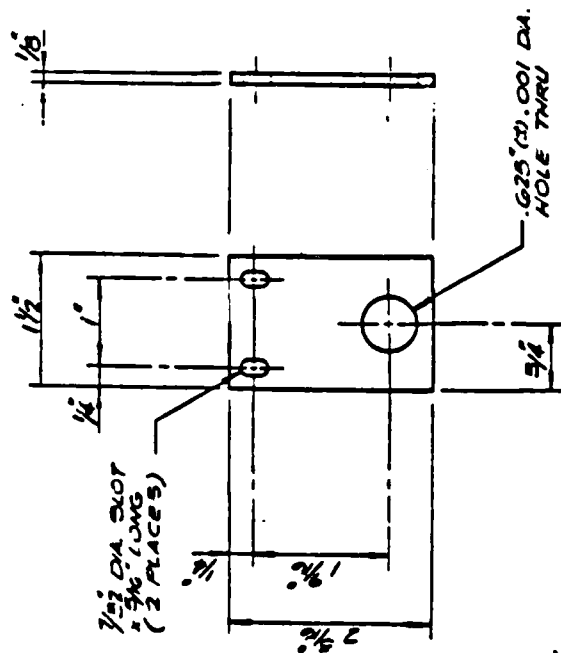
1. WORK THIS DRAWING WITH BILL OF MATERIALS MOORE & LING CO.
2. ALL FRACTIONS ON DRAWING'S TO BE (2) / 64".
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



(ALUMINUM)

NOTES:

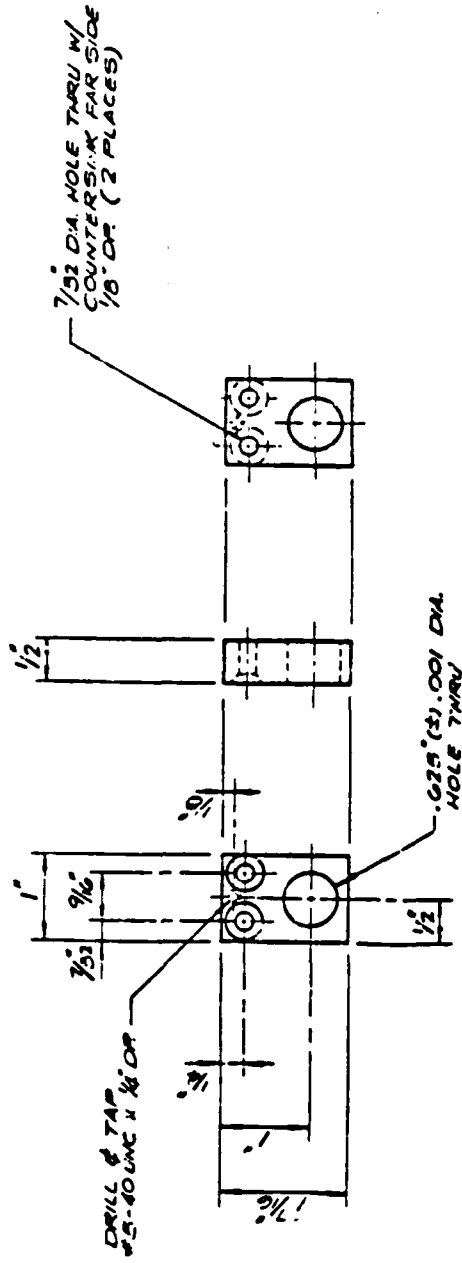
1. WORK THIS DWG WITH BILL OF MATERIALS MOON & LANG COIL.
2. ALL FRACTIONAL DIMENSIONS TO BE IN 16'S.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 15 (ALUMINUM)
SCALE: FULL

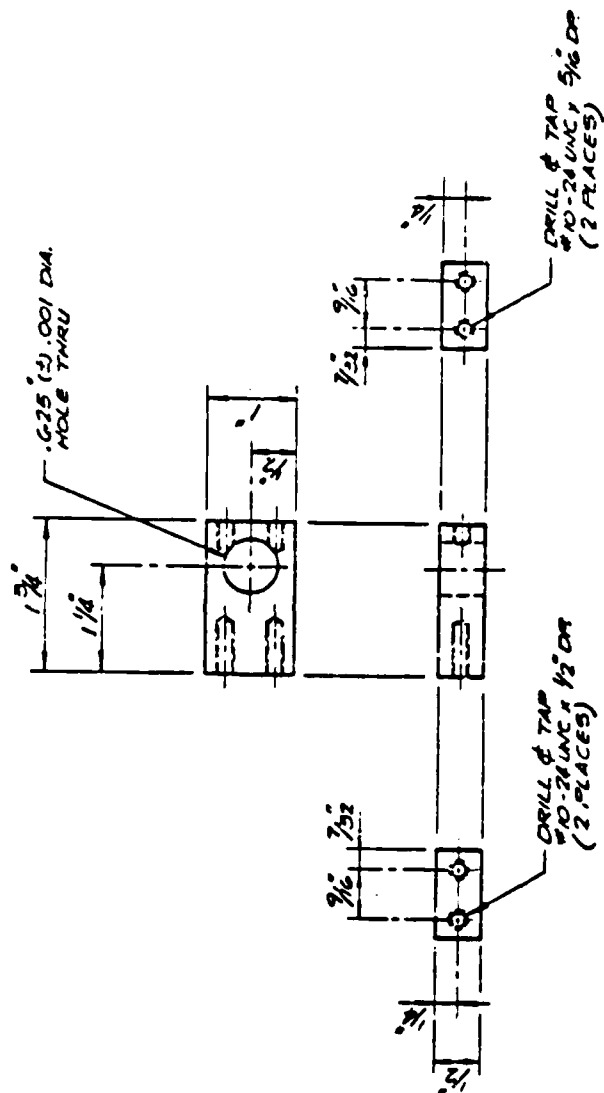
NOTES:

1. WORK THIS DWG WITH BILL OF MATERIALS MOOD & ENG UCL.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64".
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 10 (ALUMINUM)
SCALE: FULL

1. WORK THIS DRWG WITH CURE OF MATERIALS MODIO & DRWG COIL.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/8".
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 17 (ALUMINUM)
SCALE: FULL

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				7-10-91	40001	2 of 3
Approved by				NO	REVISIONS	DATE
Project Engineer					DESCRIPTION	
Project Name CAD-CAM						
Drawing Title TURRET PUNCH						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
11	0007	Punch Retainer Clip	4	C. Steel		
12	0007	Plunger	1	C. Steel		
13	0007	Punch	3	C. Steel		
14	0007	Die Plate	8	C. Steel		
15	0003	Carriage Screw	1	See Detail		
16	0008	Transverse Screw	1	See Detail		
17	0004	Carriage Support Rod	2	C. Steel		
18	0004	Transverse Support Rod	2	C. Steel		
19	0004	Washer	1	Nylon		
20	0009	Drive Gear	1	See Detail		
21	0004	Bearing Clamp	1	Aluminum		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				7-10-91	40001	3 of 3
Approved by				NO	REVISIONS	DATE
Project Engineer					DESCRIPTION	
Project Name CAD-CAM						
Drawing Title TURRET PUNCH						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
22	0009	Lower Material Clamp	1	C. Steel		
23	0009	Upper Material Clamp	1	C. Steel		
24	0009	Cover	1	C. Steel		
25	0001 & 2	Motor-stepping, Model No. 4001	2	Mfg. Std. Slo-Sym		
26	0001	Motor-stepping, Model No. Mo61	2	Mfg. Std. Slo-Sym		
27	0001 & 2	Solenoid - Heavy Duty AC, NA-700 Push Type, 1" Stroke, Model No. NA-733-A1	1	Mfg. Std. Trombeta		
28	0001 & 2	Solenoid - Tubular Push Type, Model No. P8-1L, 12V D.C.	1	Mfg. Std. Dormeyer		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

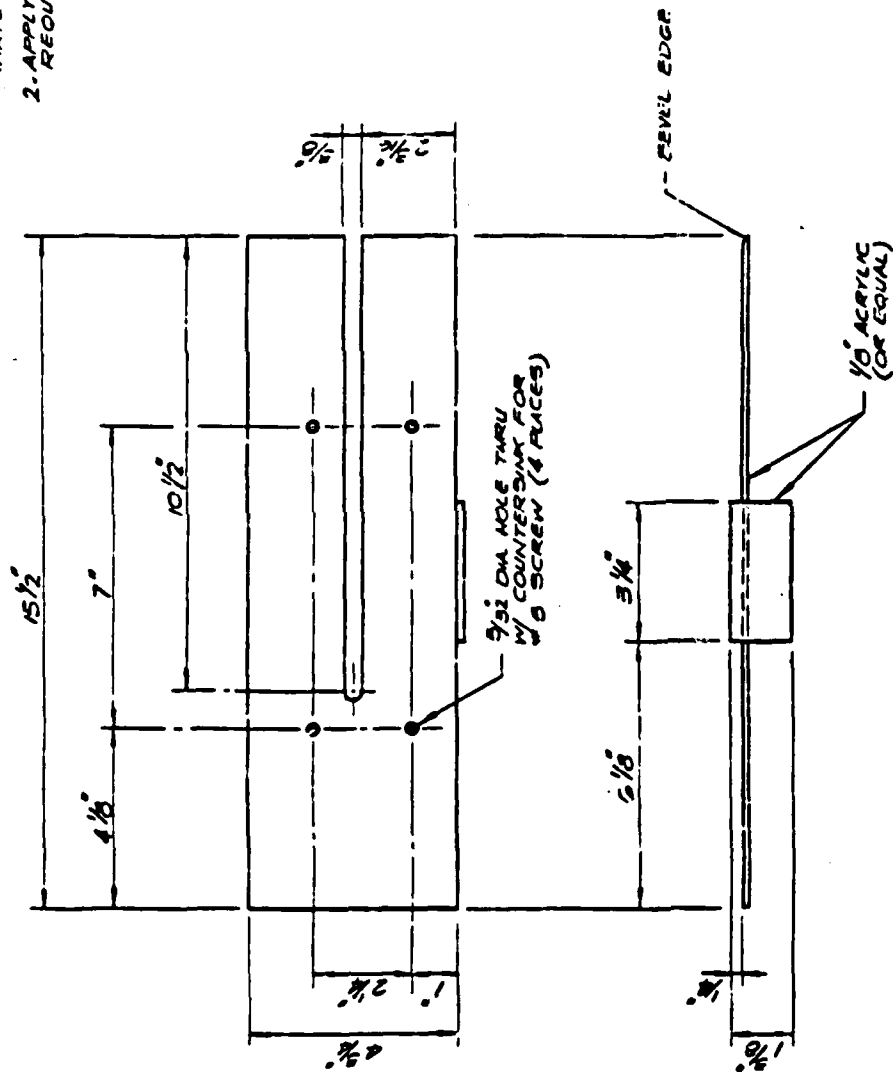
CAD-CAM DEPT.

PROJECT
TURRET FUNCH (MOOOL)[illegible]

APPENDIX B

Detailed Design for the Miniature Sheet-metal Turret Punch

NOTES
 1. WORK THIS DWG. WITH E.I.I. OF
 MATERIALS MODIO & DWG. OC11.
 2. APPLY PRINT & LOGO AS
 REQUIRED.

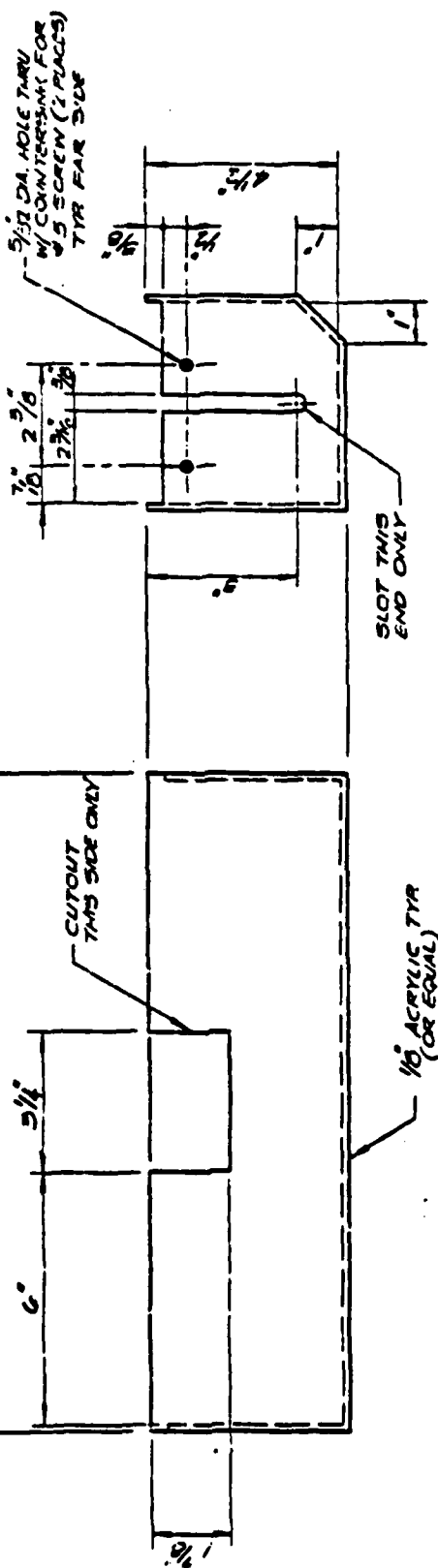


DETAIL ITEM 28 (ACRYLIC OR EQUAL)
 SCALE: 1/2" = 1'-0"

15 1/4"

- NOTES
1. WORK THIS DING WITH BILL OF MATERIALS MCDIO & DNG 0011.
 2. ON ITEM 27, DIMENSIONS ARE TO INSIDE SURFACE.
 3. APPLY PAINT & LOGO AS REQUIRED.

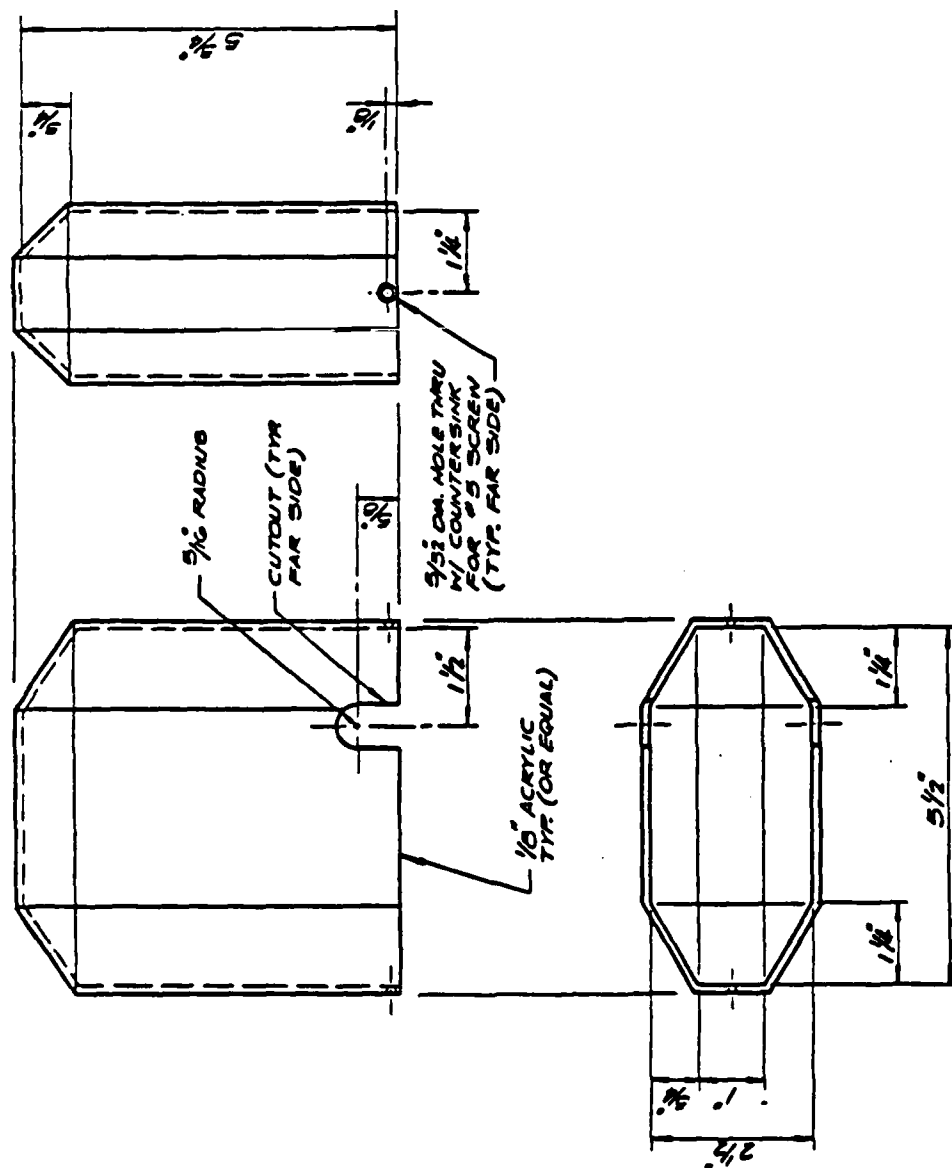
BEVEL CORNERS
(THIS END)



DETAIL ITEM 27 (ACRYLIC OR EQUAL)
SCALE: 1/2" = 1'-0"

NOTES

1. WORK THIS DWG. WITH BILL OF MATERIALS NCOID & DWG COID.
2. ALL DIMENSIONS ARE TO "INSIDE" SURFACE.
3. APPLY PAINT & LOGO AS REQUIRED.

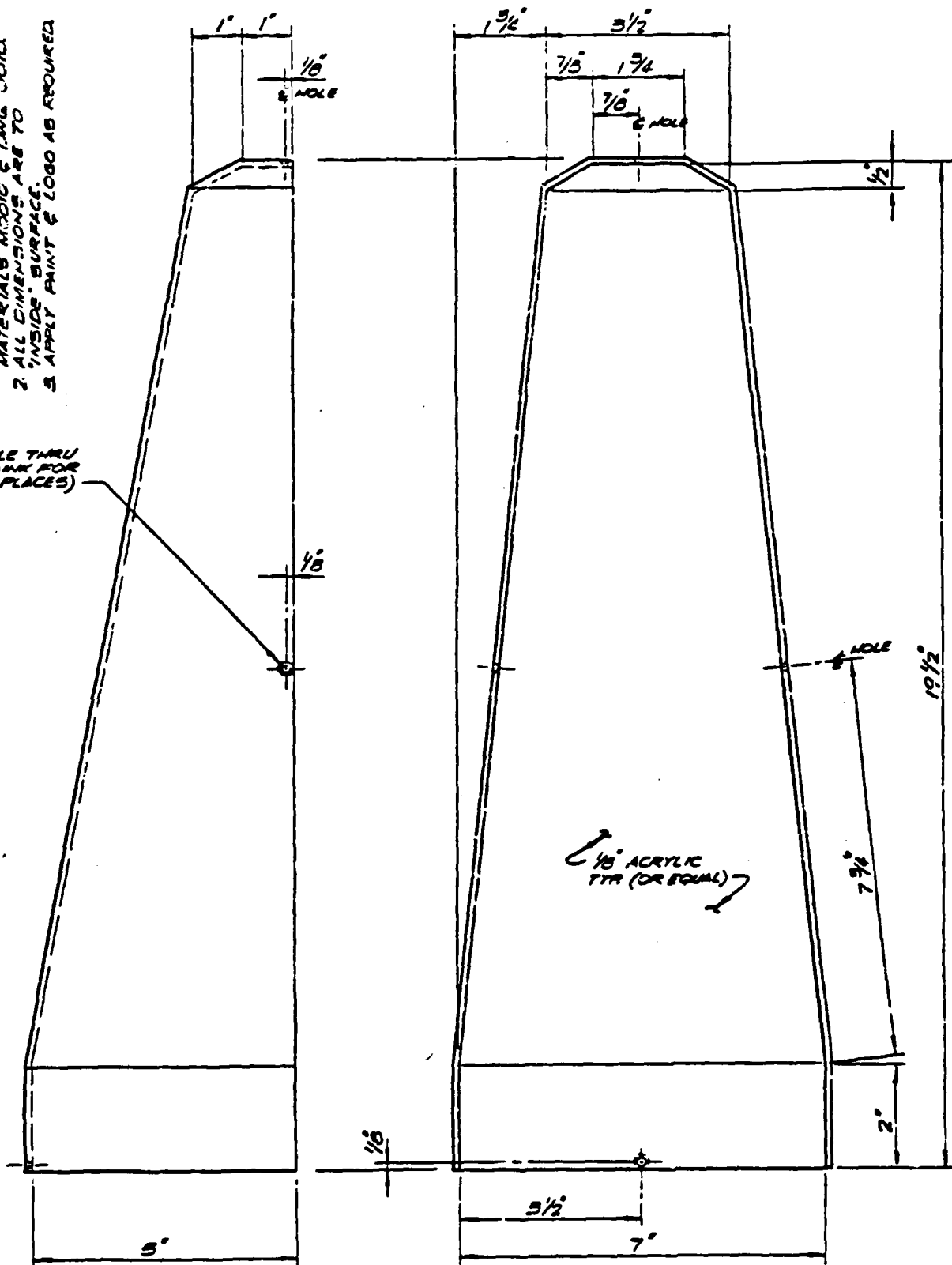


DETAIL ITEM 26 (ACRYLIC OR EQUAL)
SCALE: $\frac{3}{4}" = 1"$

NOTES

1. WORK THIS Dwg. WITH BILL OF
2. MATERIALS MCOIC & DWG. COIC.
3. ALL DIMENSIONS ARE TO
4. "INSIDE" SURFACE.
5. APPLY PAINT & LOGO AS REQUIRED

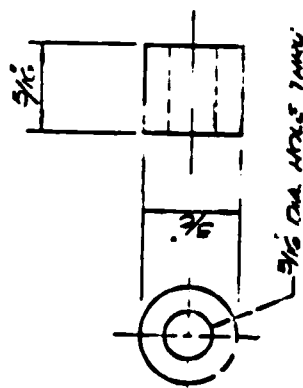
5/16" DIA. HOLE THRU
1/4" COUNTERSINK FOR
#5 SCREW (4 PLACES)



DETAIL ITEM 25 (ACRYLIC OR EQUAL)
SCALE: 3/4" = 1'-0"

NOTES

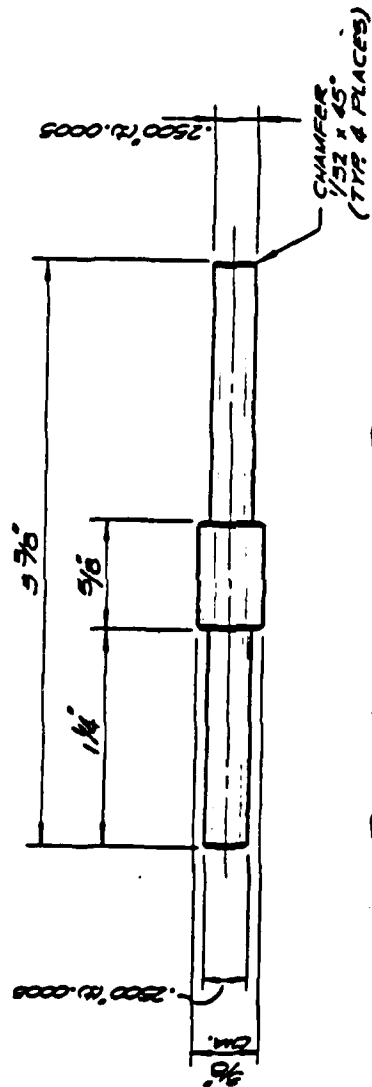
- 1 WORK THIS ONE WITH BILL OF MATERIALS IN MONROE AND SMC COIO & COIL.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/4".



DETAIL ITEM 24 (C. STEEL)
SCALE: 3/4"

NOTES

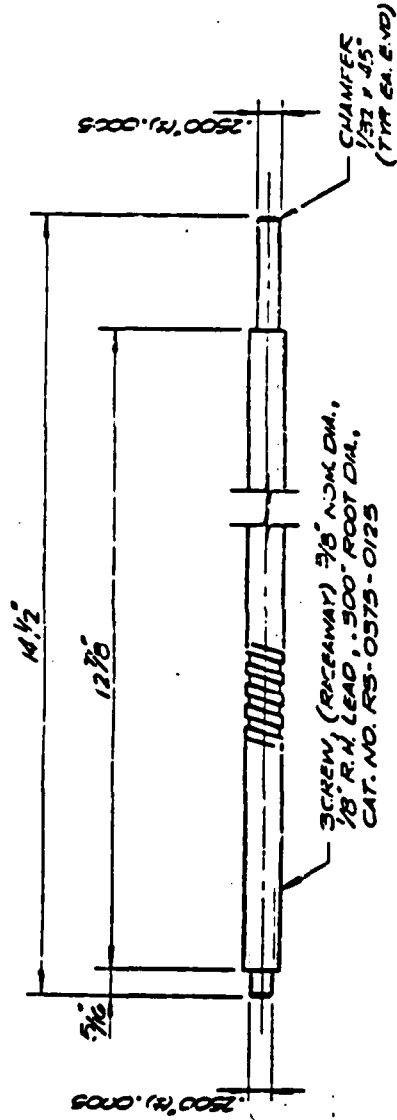
1. WORK THIS DWG WITH BILL OF MATERIALS N.B. MOOIO AND DWG 0010 & 0011.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) $\frac{1}{64}$.



DETAIL ITEM 23 (C. STEEL)
SCALE: 2"=1"

NOTES

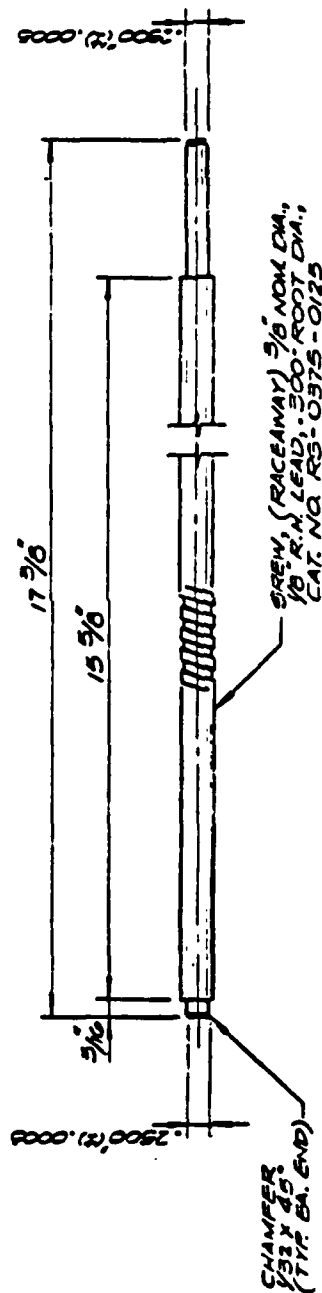
- 1 WORK THIS DWG WITH BILL OF MATERIALS NR M0010 AND DWG 0010 & COIL.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64".



DETAIL ITEM 22 (C STEEL)
SCALE: FULL

NOTES

1. WORK THIS DRAWING WITH BILL OF MATERIALS AND MODIO AND DING CO. COIL.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64".

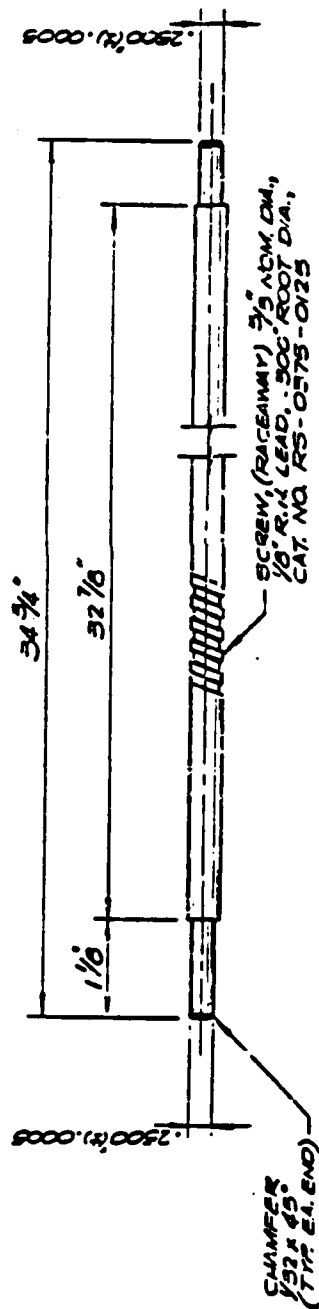


DETAIL ITEM 21 (C. STEEL)

SCALE: FULL

NOTES

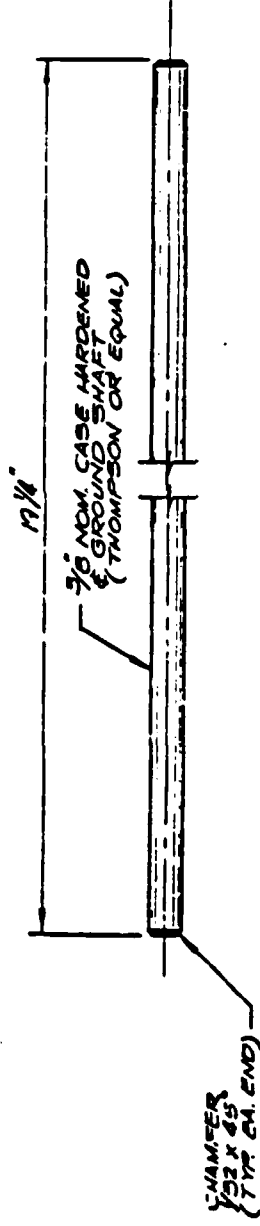
1. WORK THIS DWG WITH BILL OF MATERIALS AND MOULD AND DWG COOIO & COOII.
2. ALL FRACTIONAL DIMENSIONS TO BE IN $\frac{1}{64}$.



DETAIL ITEM 20 (C. STEEL)
SCALE: FULL

NOTES

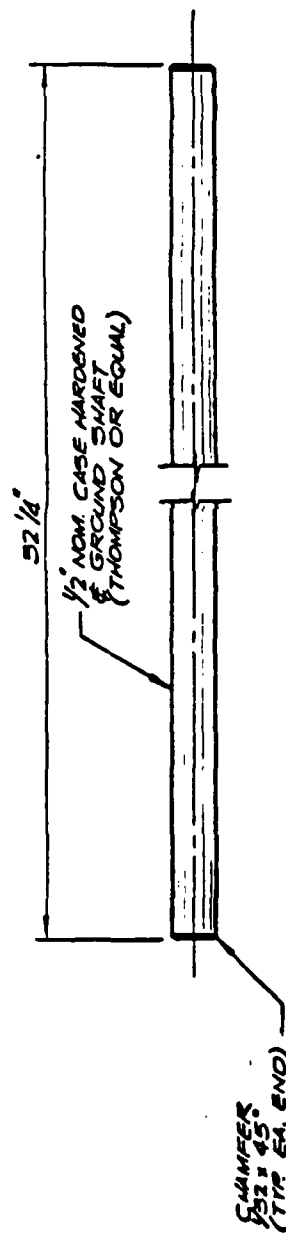
1. WORK THIS DWG WITH BILL OF MATERIALS AND MOODIO AND DWG. 0010 & 0011.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) / 64".



DETAIL ITEM 19 (C. STEEL)
SCALE: FULL

NOTES

1. WORK THIS DWG WITH BILL OF MATERIALS, N° MOOIO AND DWG. OOIO & OOII.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) / 64.



DETAIL ITEM 18 (C. STEEL)
SCALE: FULL

BRIGHAM YOUNG UNIVERSITY

Prepared by		Project No.		Drawing No.		Sheet	
FORREST BLAIR		7-10-31		M0001		4 of 9	
Approved by		NO.		REVISIONS		DATE	
Project Engineer		Date		Description		Date	
Project Name		Date		Description		Date	
CAD-CAM		Date		Description		Date	
Drawing Title		Date		Description		Date	
TURRET PUNCH		Date		Description		Date	
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE		
29	0001 & 2	Bearing - Linear, 5/8" O.D. x 7/8" Long, Cat. No. A-61014	3	Mfg. Std.			
30	0001	Bearing - Model No. 37KDD for 1/4" Shaft	2	Mfg. Std.			
31	0001	Bushing - 3/8" Nom. I.D. x 9/16" Nom. O.D., Cat. No. 56929	1	Mfg. Std.			
32	0002	Bushing - 1/2" Nom. I.D. x 3/4" Nom. O.D., Cat. No. 35606 PB-912-12	1	Mfg. Std.			
33	0002	Bushing - 1/2" Nom. I.D. x 3/4" Nom. O.D., Cat. No. 35596 PB-912-4	1	Mfg. Std.			
34	0001	Coupling - 1/4" Nom. Bore, Model No. L050	2	Mfg. Std.			

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BRIGHAM YOUNG UNIVERSITY

Prepared by		Project No.		Drawing No.		Sheet	
FORREST BLAIR		7-10-31		M0001		5 of 9	
Approved by		NO.		REVISIONS		DATE	
Project Engineer		Date		Description		Date	
Project Name		Date		Description		Date	
CAD-CAM		Date		Description		Date	
Drawing Title		Date		Description		Date	
TURRET PUNCH		Date		Description		Date	
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE		
35	0001	Bellows - Attach as required		As Required			
36	0001 & 2	Ball Nut - for 3/8" Nom. Screw, Cat. No. RN-0375-0125	2	Mfg. Std.			
37	0002	Spring - Plunger, 1/2" O.D., Cat. No. S-193, Stainless Compression	1	Mfg. Std.			
38	0002	Spring- Punch, 3/8" O.D., Cat. No. 1785, Music Compression	8	Mfg. Std.			
39		Not Used		Century			
40		Not Used					
41	0001 & 2	Cap Screw - Hex Socket, 1/4-20 UNC x 1/2" Long	5	C. Steel			
42	0002	Cap Screw - Hex Socket, 1/4-20 UNC x 1/2" Long	1	C. Steel			

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				7-10-81	MO001	6 of 8
Reviewed by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer				NO.	REVISIONS DESCRIPTION	DATE
Project Name CAD-CAM				NO.	REVISIONS DESCRIPTION	DATE
Drawing Title TURRET PUNCH				NO.	REVISIONS DESCRIPTION	DATE
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
4	0002	Cap Screw - Hex Socket, #10-24 UNC x 3/8" Long	4	C. Steel		
44	0001	Cap Screw - Hex Socket, #10-24 UNC x 3/4" Long	8	C. Steel		
45	0001	Cap Screw - Hex Socket, #12-24 UNC x 3/4" Long	4	C. Steel		
46	0001 & 2	Machine Screw - Hex Head, #12-24 UNC x 1" Long	4	C. Steel		
47	0002	Cap Screw - Hex Socket, #8-32 UNC x 3/8" Long	2	C. Steel		
48	0001	Cap Screw - Hex Socket, #8-32 UNC x 5/8" Long	4	C. Steel		
49	0002	Cap Screw - Hex Socket, #5-40 UNC x 3/8" Long	16	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

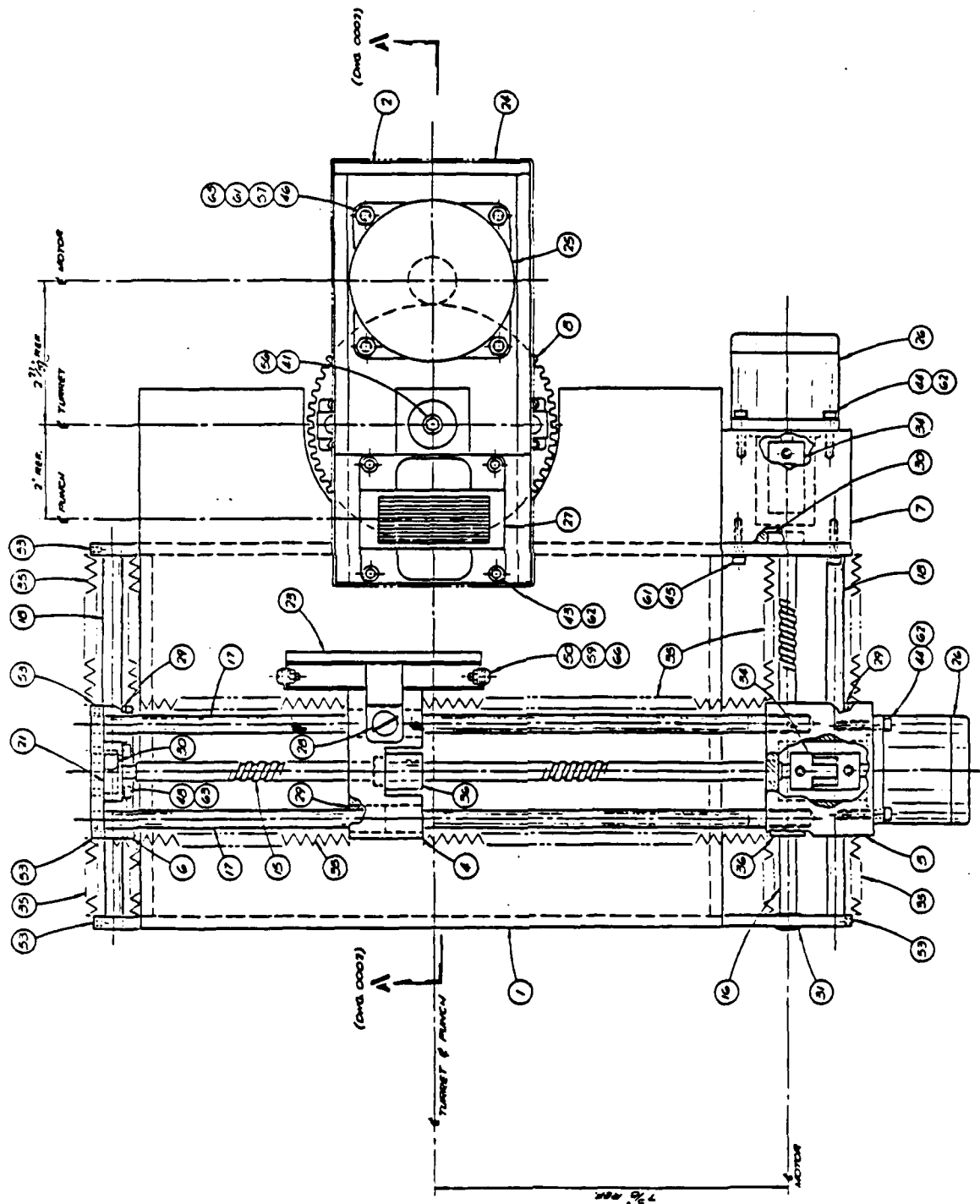
BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				7-10-81	MO001	7 of 9
Reviewed by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer				NO.	REVISIONS DESCRIPTION	DATE
Project Name CAD-CAM				NO.	REVISIONS DESCRIPTION	DATE
Drawing Title TURRET PUNCH				NO.	REVISIONS DESCRIPTION	DATE
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
50	0001 & 2	Machine Screw - Slotted Fillister Head, #5-40 UNC x 3/8" Long	2	C. Steel		
51	0002	Machine Screw - Slotted 1000 Flat Head, #5-40 UNC x 1/4" Long	16	C. Steel		
52	0002	Machine Screw - Slotted Truss Head, #5-40 UNC x 1/4" Long	4	C. Steel		
53	0001	Set Screw - Hex Socket, #5 x 1/8" Long	6	C. Steel		
54	0002	Set Screw - Hex Socket, #10 x 3/16" Long	2	C. Steel		
55	0002	Dowel Pin - 1/8"	2	C. Steel		
56	0001 & 2	Washer - Flat, 1/4"	2	C. Steel		
57	0001 & 2	Washer - Flat, #12	4	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR					MO001	8 of 8
Approved by				REVISIONS		
Date				NO.	DESCRIPTION	DATE
Date						
Date						
Project Engineer						
Project Name						
CAD-CAM						
Drawing Title						
TURRET PUNCH						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
53	0002	Washer - Flat, #8	2	C. Steel		
59	0001 & 2	Washer - Flat, #5	4	C. Steel		
60	0001 & 2	Washer - Lock, 1/4"	4	C. Steel		
51	0001 & 2	Washer - Lock, #12	8	C. Steel		
52	0001 & 2	Washer - Lock, #10	12	C. Steel		
53	0002	Washer - Lock, #8	6	C. Steel		
54	0002	Washer - Lock, #5	16	C. Steel		
55	0001 & 2	Nut - Hex Head, #12	4	C. Steel		
56	0001 & 2	Nut - Hex Head, #5	2	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.



NOTES
 1 ALL ITEMS MARKED TAGS
 ARE FOUND ON BILL OF
 MATERIALS N° M0001.

PLAN VIEW
 FULL BEAM

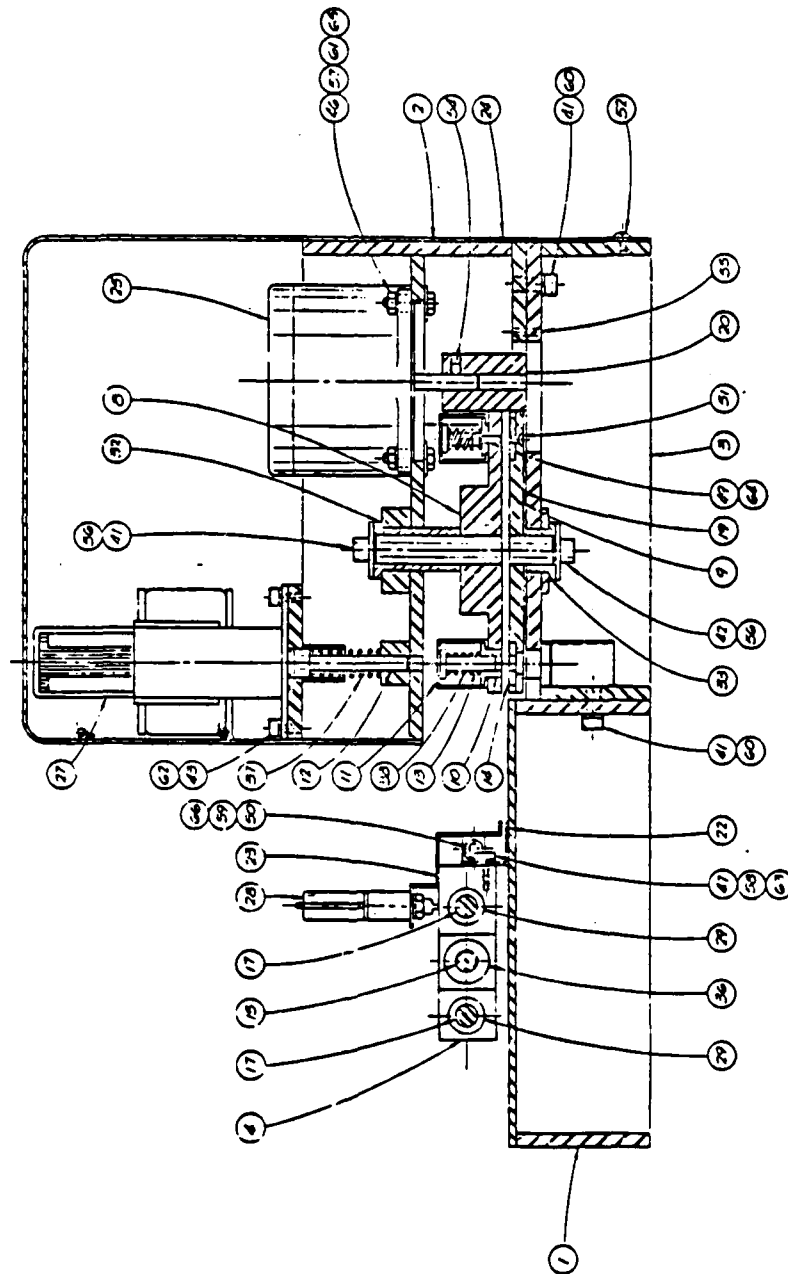
DATE 10, 1931
 CHD-CJM DWT

BRIGHAM YOUNG UNIVERSITY

TURRET PUNCH

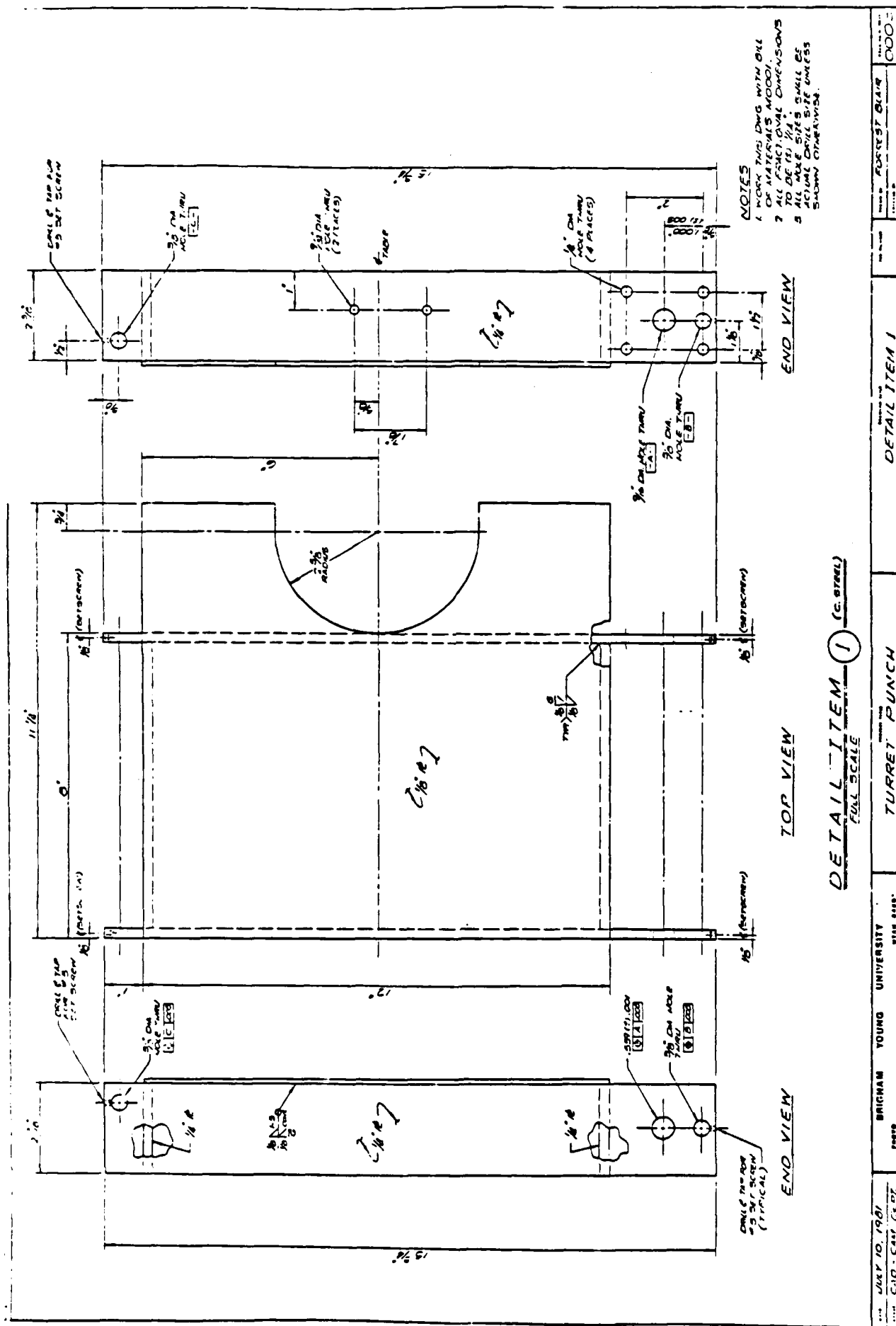
ASSEMBLY-PLAN

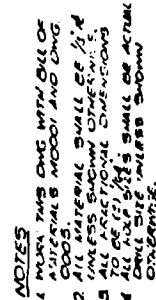
ADJUST BLANK



NOTES
 1. ALL ITEMS SHOWN THIS
 2. ARE FOUND ON END OF
 3. MATERIALS NO. 100001.

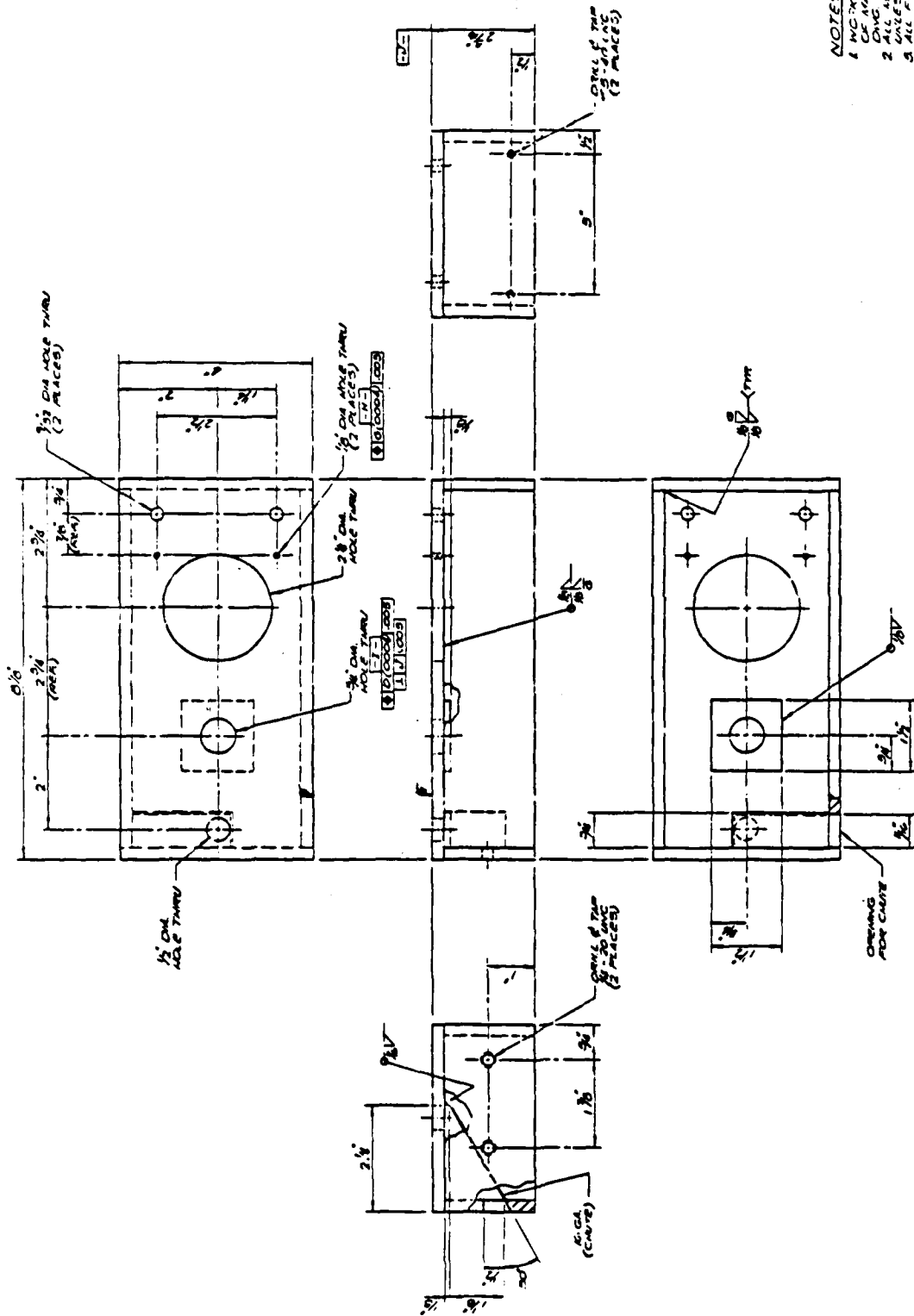
SECTION A-A (ENCL. 100001)
 FULL SCALE





DETAIL ITEM 2 (C. 0722)

DATE	JULY 10, 1961	BRIDGEMAN YOUNG UNIVERSITY	PROJECT NAME	TURRET PUNCH DETAIL ITEM 2	SEE NUMBER	WORKS BY FOREST CLARK 1-10-1960 B	ESTIMATING NUMBER 000%
TIME	CAD - CAN	80000					



NOTES

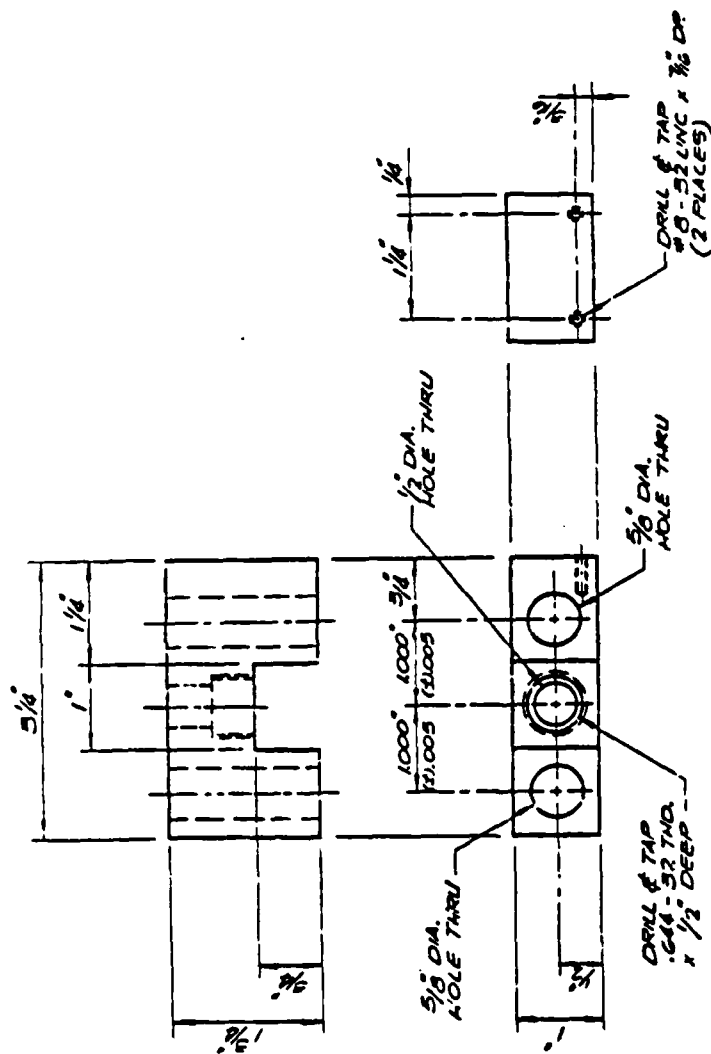
1. 1/2" DIA. HOLE THRU (2 PLACES)
2. 3/8" DIA. HOLE THRU (2 PLACES)
3. 1/2" DIA. HOLE THRU (2 PLACES)
4. 3/8" DIA. HOLE THRU (2 PLACES)

DETAIL ITEM 3 (E-STEEL)
FULL SCALE

DATE: JULY 10, 1961	BRIDGEMAN YOUNG UNIVERSITY	TURRET PUNCH	DETAIL ITEM 3	FORREST BLAIR	0005
BY: CAD - CAM DEPT	BY: 0000				

NOTES

- 1 WORK THIS Dwg WITH BILL OF MATERIALS MCOOI.
- 2 ALL FRACTIONAL DIMENSIONS TO BE IN $\frac{1}{8}$ IN.
- 3 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.

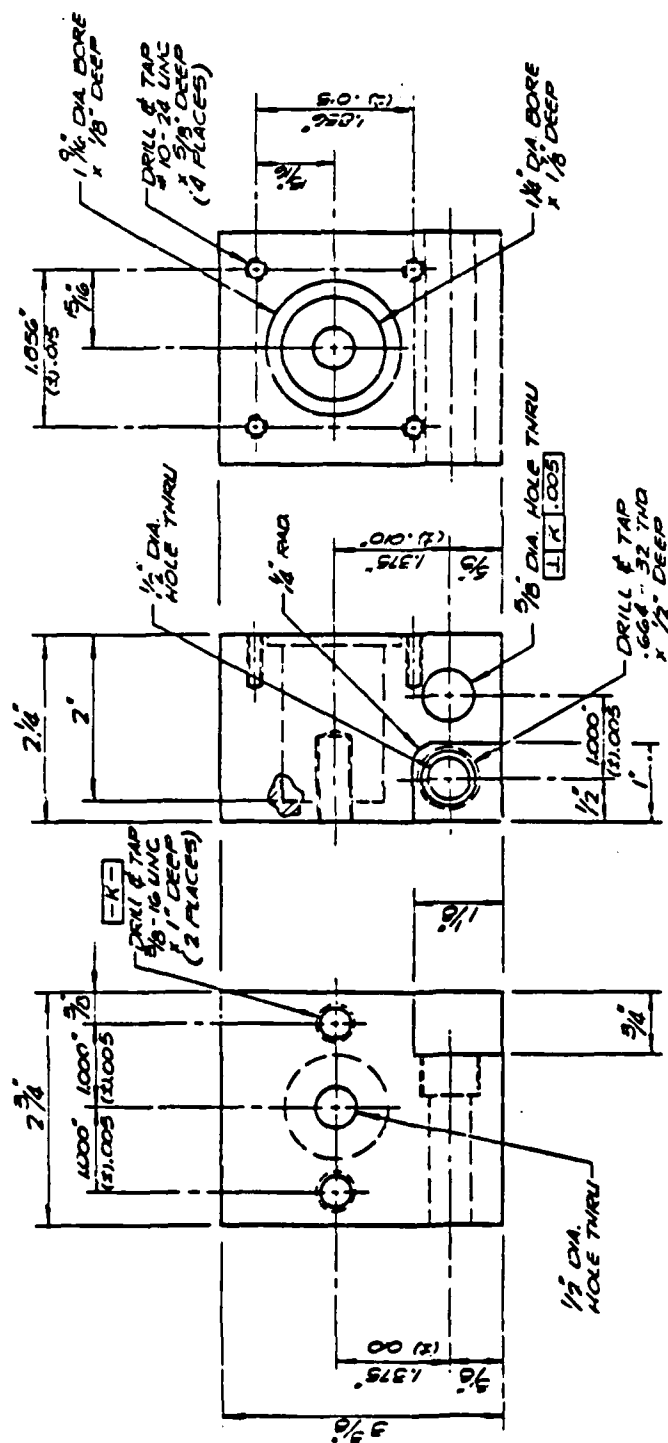


DETAIL ITEM 4 (ALUMINUM)

FULL SCALE

NOTES

- 1 WORK THIS DRUG WITH ALL OF MATERIALS ABOVE.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/32" UNLESS SHOWN OTHERWISE.
- 3 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 5 (ALUMINUM)
FULL SCALE

NOTES

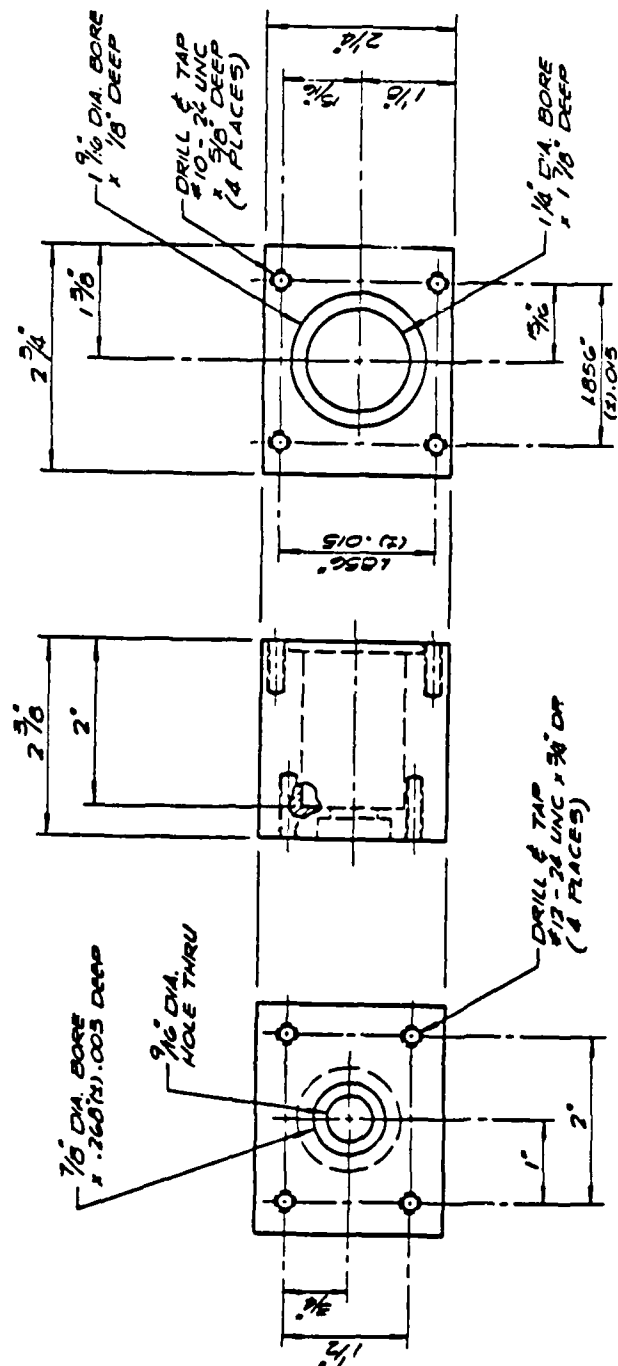
1. WORK THIS END WITH BILL OF MATERIALS ACCORD.
2. ALL FRACTIONAL DIMENSIONS TO BE (3) 1/64" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 6 (ALUMINUM)
FULL SCALE

NOTES

1. WORK THIS Dwg WITH BILL OF MATERIALS MOODI.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SPECIFIED OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.

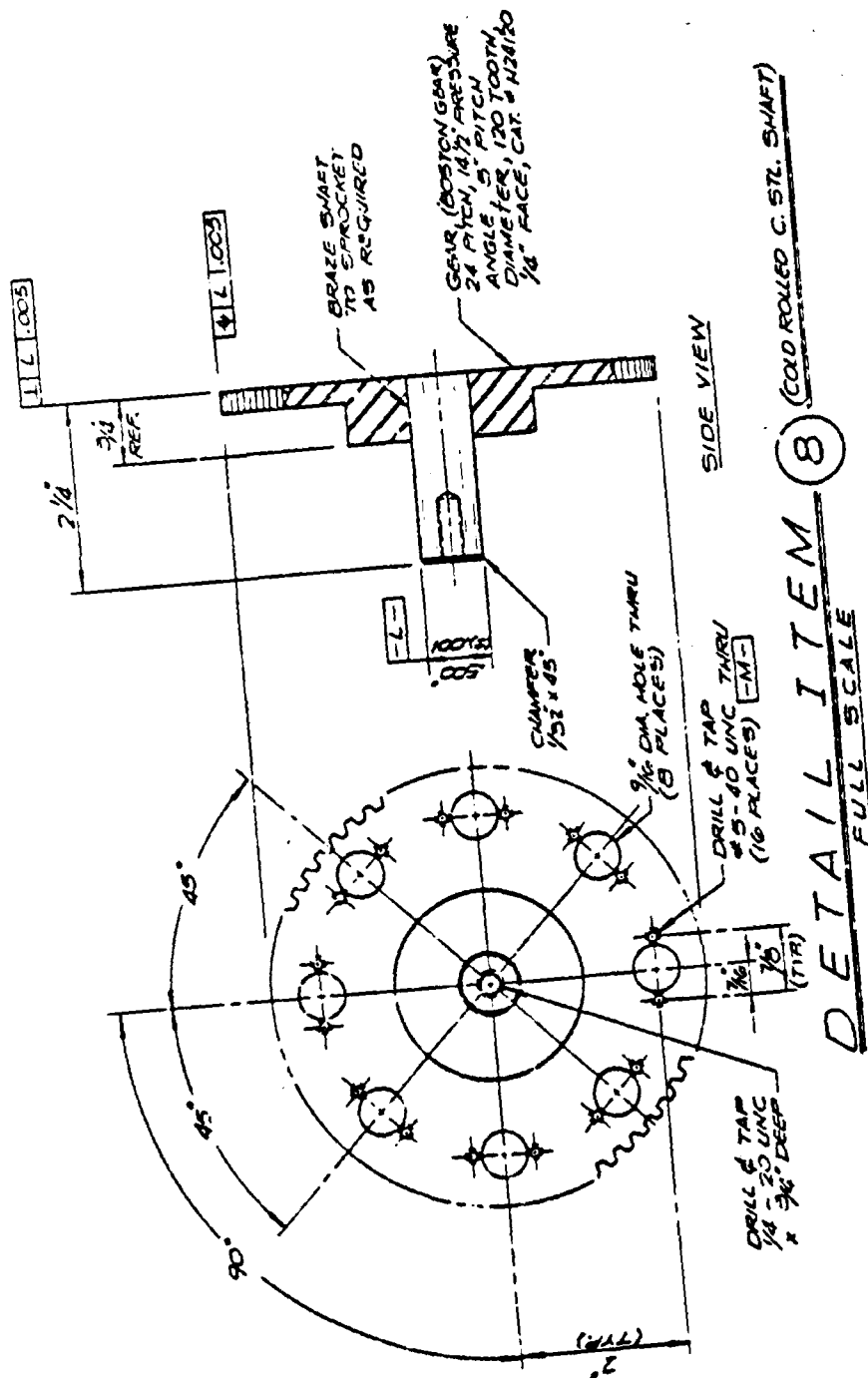


DETAIL ITEM 7 (ALUMINUM)
FULL SCALE

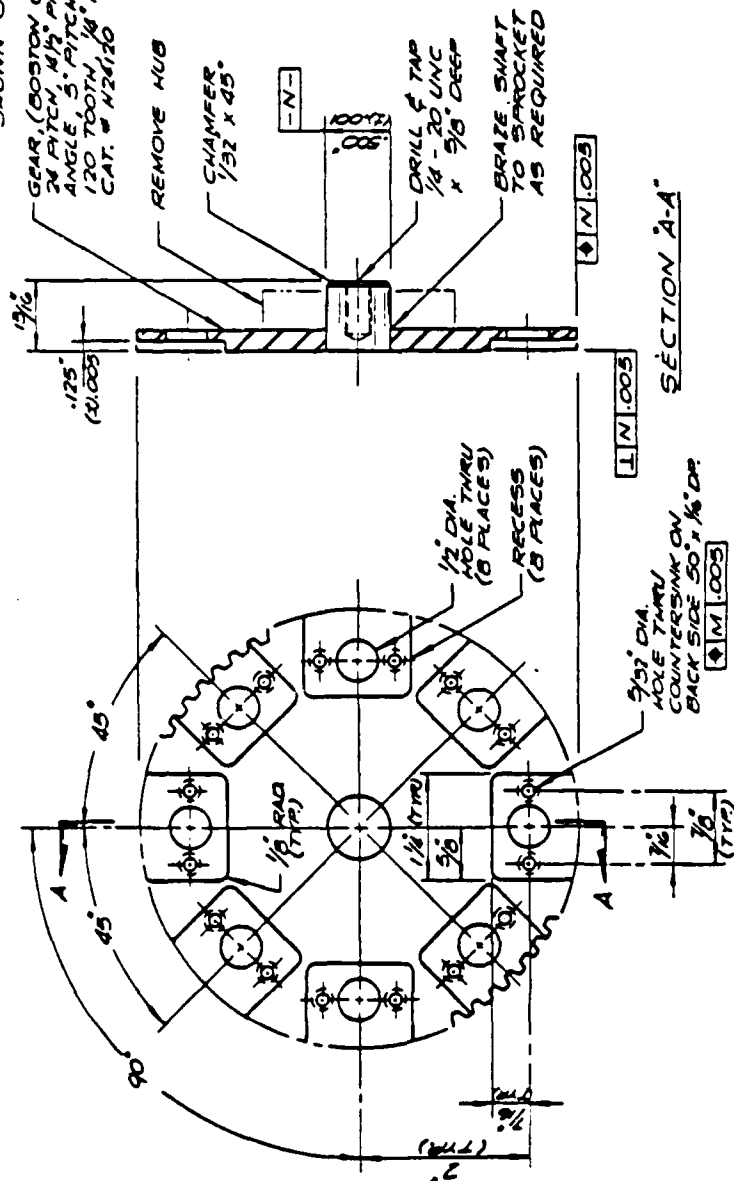
NOTES

1. WITH THIS GIVE WITH ALL OF MATERIALS ABOVE.
2. ALL FRACTIONAL DIMENSIONS TO BE ROUNDED UP.
3. ALL ANGLES SHOWN OTHERWISE.
4. ALL HOLE SIZES SMALLER.
5. ALL HOLE SIZES SIZE UNLESS ACTUAL OR ELSE.
6. SHOWN OTHERWISE.

- 1 1. WHAT IS THIS MATERIAL MOOD?
- 2 2. ALL FRACTIONAL DIMENSIONS TO BE IN 1/8" YES.
- 3 3. ALL ANGLES TO BE 10°.
- 4 4. ALL SLOPES DOWN OTHERWISE.
- 5 5. ALL HOLE SIZES SHALL BE
- 6 6. ALL HOLE ORILL SIZE UNLESS
- 7 7. ACTUAL ORILL SIZE UNLESS
- 8 8. SLOPES DOWN OTHERWISE.



1 WORK THIS DWS WITH BILL
OF MATERIALS MOCON.
2 ALL FRACTIONAL DIMENSIONS
TO BE IN INCH.
3 ALL ANGLES TO BE (1/2)°.
4 UNLESS SHOWN OTHERWISE.
5 ALL HOLE SIZES SHALL BE
ACTUAL DRILL SIZE UNLESS
SHOWN OTHERWISE.



SECTION A-A'

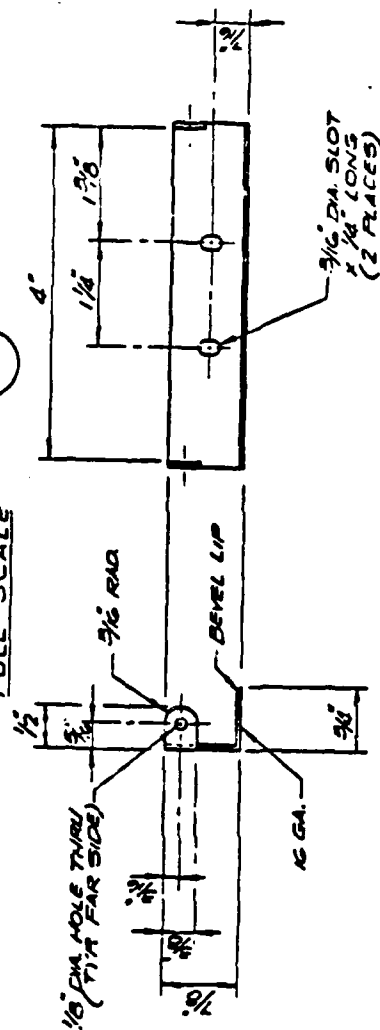
DETAIL ITEM 9 (COLD ROLLED C. STL. SHAFT)
FULL SCALE

NOTES

1. WORK THIS DWG WITH BILL OF MATERIALS INCLUDED.
2. ALL FRACTION DIMENSIONS TO BE (2) 1/32" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.

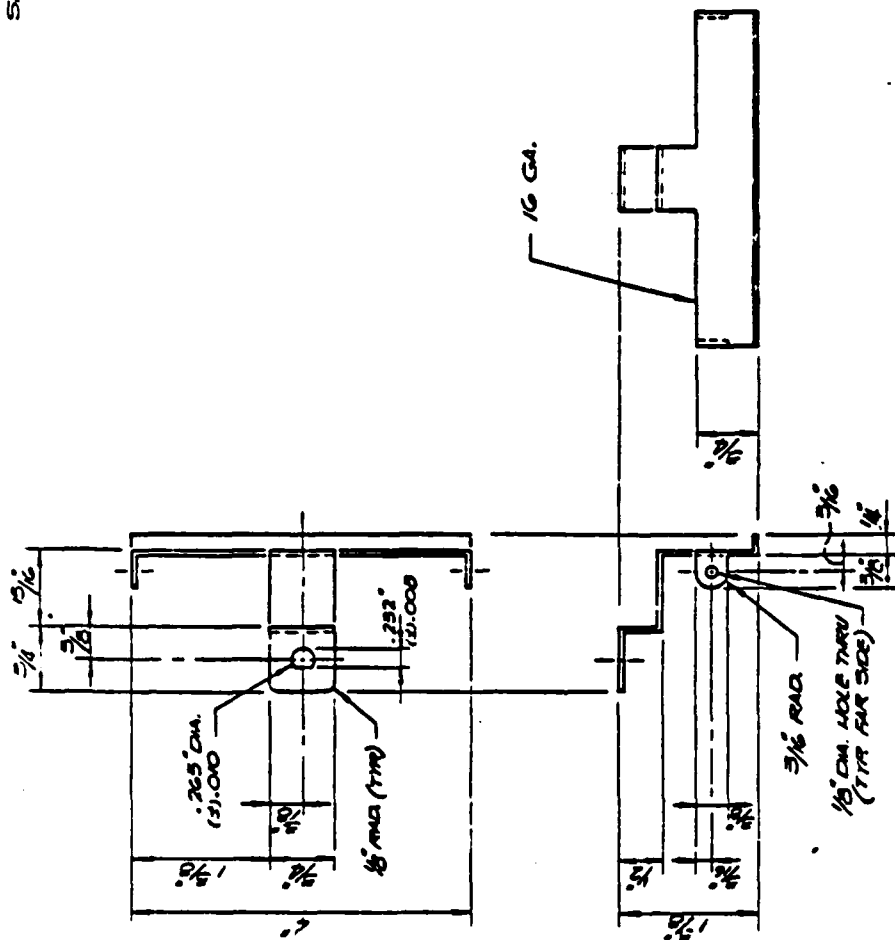
DETAIL ITEM 23 (C. STEEL)

FULL SCALE



NOTES

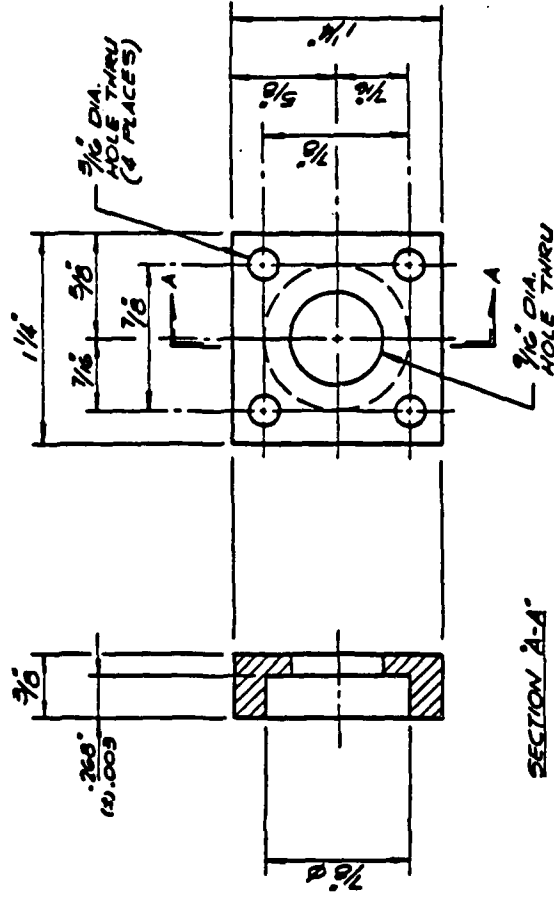
1. WORK THIS DWG WITH BILL OF MATERIALS MCD001.
2. ALL FRACTION DIMENSIONS TO BE (3) 1/32" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 22 (C. STEEL)
FULL SCALE

NOTES

1. WORK THIS DRAWING WITH BILL OF MATERIALS MO001.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.

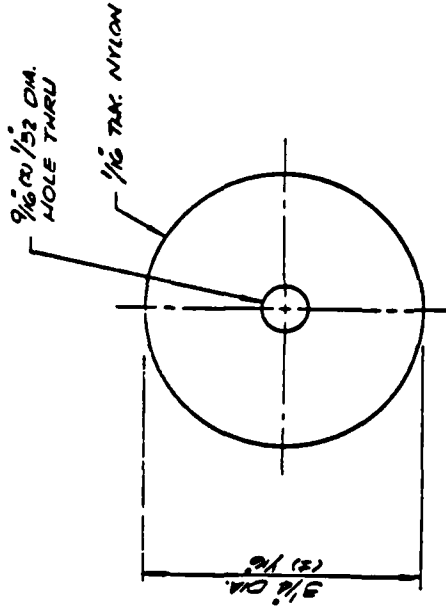


SECTION A-A

DETAIL ITEM 21 (ALUMINUM)
FULL SCALE

NOTES

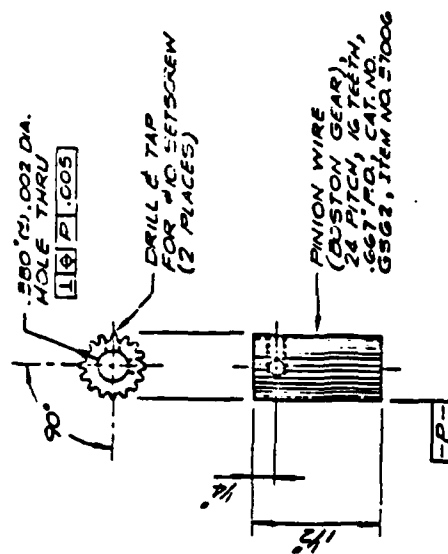
1. WORK THIS DRAWING WITH BILL GE MATERIALS MCOOI.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 19 (NYLON)
FULL SCALE

NOTES

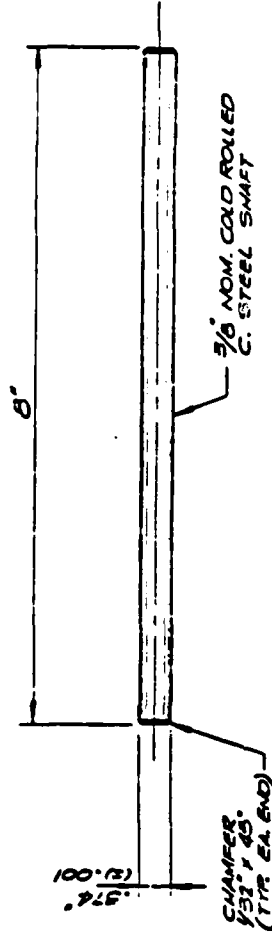
1. MARK THIS DWG WITH BILL OF MATERIALS NO. 001.
2. ALL FRACTIONAL DIMENSIONS TO BE IN 1/64" UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 20
FULL SCALE

NOTES

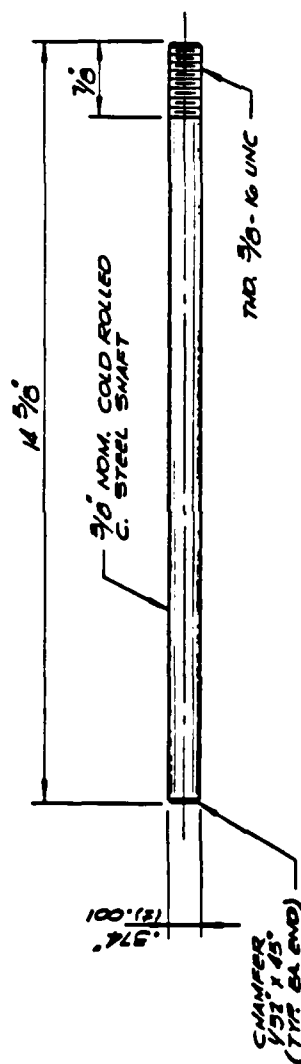
1. WORK THIS DRAWING WITH BILL OF MATERIALS MODIFIED.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SPECIFIED OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 18 (C. STEEL)
FULL SCALE

NOTES

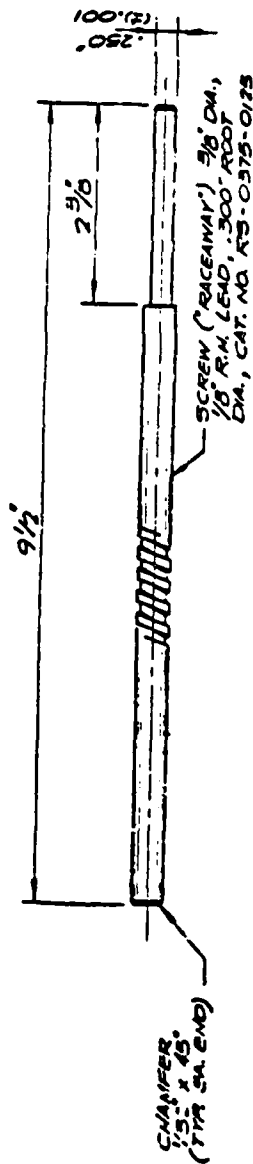
- 1 WORK THIS Dwg WITH BILL OF MATERIALS M0001.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SHOWN OTHERWISE.
- 3 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 17 (C. STEEL)
FULL SCALE

NOTES

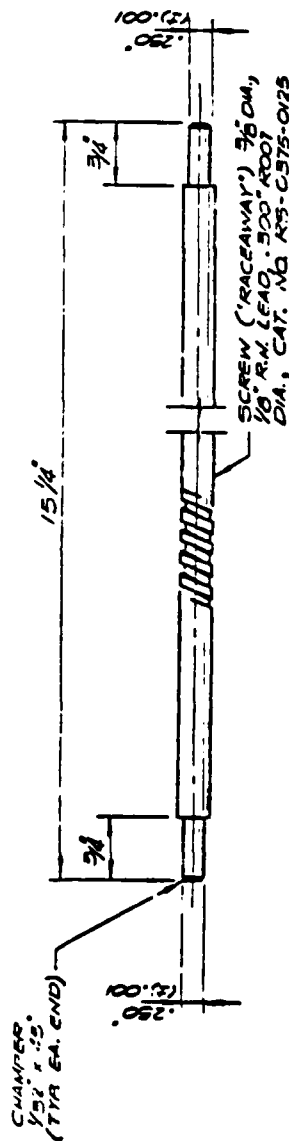
1. WORK THIS Dwg WITH BILL OF MATERIALS MOOOI.
2. ALL FRACTIONAL DIMENSIONS TO BE (2) 1/64" UNLESS SPECIFIED OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 16
FULL SCALE

NOTES

1. WORK THIS DRAWING WITH BILL OF MATERIALS MCOO01.
2. ALL FRACTIONAL DIMENSIONS TO BE IN $\frac{1}{16}$ " UNLESS SHOWN OTHERWISE.
3. ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.

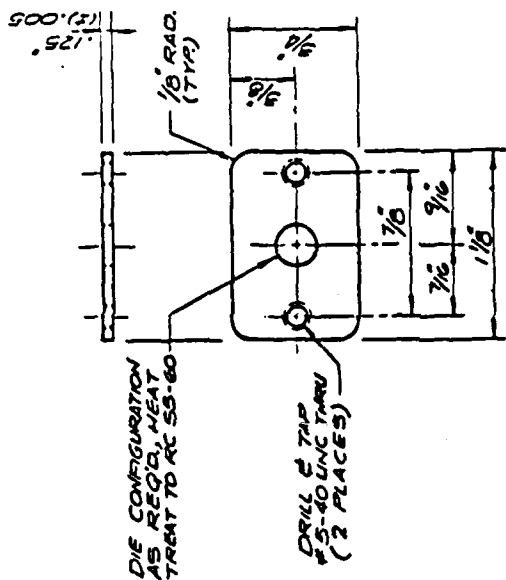


DETAIL ITEM 15

FULL SCALE

NOTES

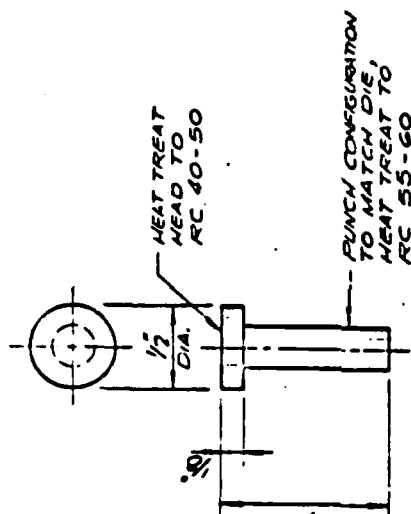
- 1 WORK THIS DWG WITH BILL OF MATERIALS MOD01.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) $\frac{1}{32}$.
- 3 ALL ANGLES TO BE (1) R .
- 4 UNLESS SHOWN OTHERWISE, ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 14 (C. STL.)
SCALE: 2"=1"

NOTES

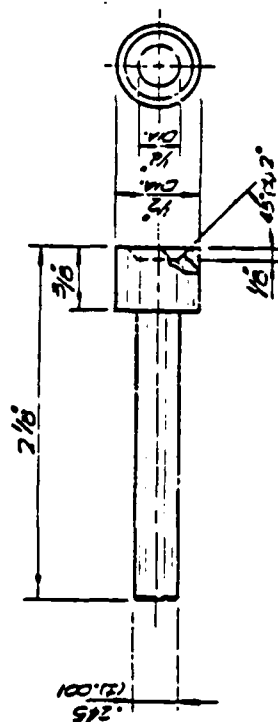
- 1 WORK THIS DWG WITH BILL OF MATERIALS MCCOI.
- 2 ALL FRACTIONAL DIMENSIONS TO BE IN 1/32".
- 3 ALL ANGLES TO BE (2) R. UNLESS SHOWN OTHERWISE.
- 4 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 13 (C. STL.)
SCALE: 2"=1"

NOTES

- 1 MAXIMUMS WITH ALL
ALL MATERIALS AHEAD.
- 2 ALL MATERIAL DIMENSIONS
TO BE IN IN.
- 3 ALL ANGLES TO BE 90°
UNLESS SHOWN OTHERWISE.
- 4 ALL HOLE SIZE SHALL BE
ACTUAL DRILL SIZE UNLESS
SHOWN OTHERWISE.

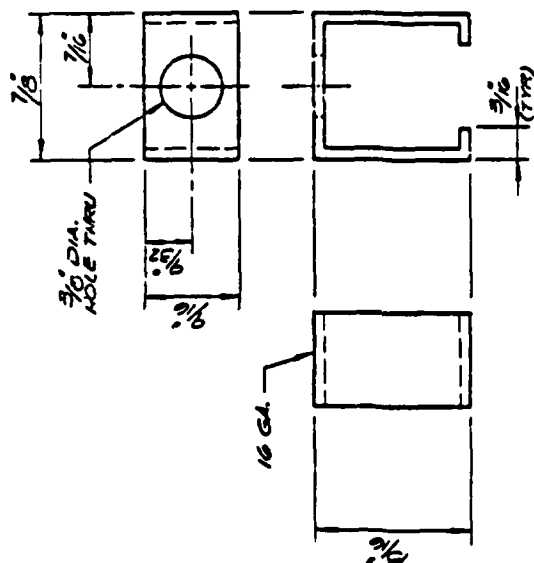


DETAIL ITEM 12 (C. 57L.)

SCALE: 2'-0"

NOTES

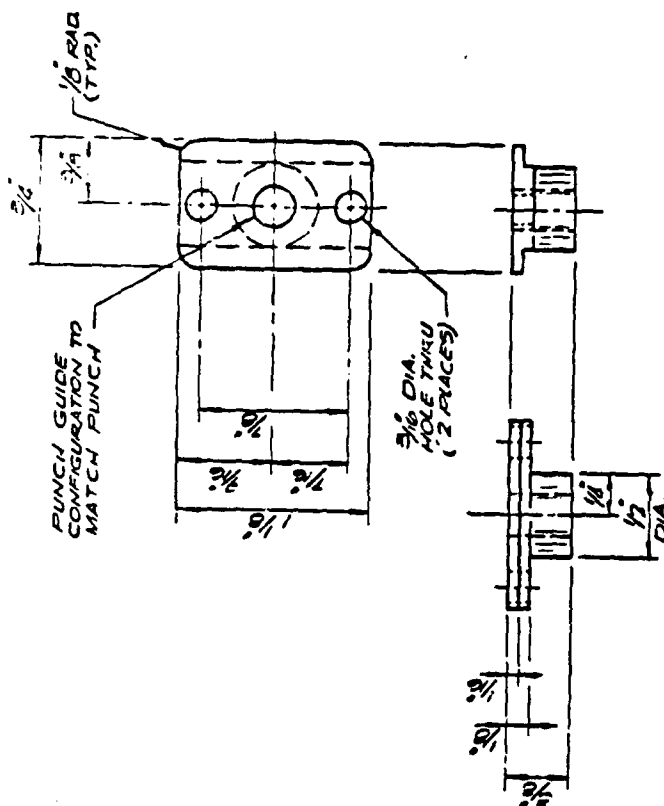
- 1 WORK THIS DWS WITH ALL
OF MATERIALS MCCOI.
- 2 ALL FRACTIONAL DIMENSIONS
TO BE (2) 1/2.
- 3 ALL ANGLES TO BE (1) 1/2.
- 4 ALL HOLE SIZES SHALL BE
ACTUAL DRILL SIZE UNLESS
SHOWN OTHERWISE.



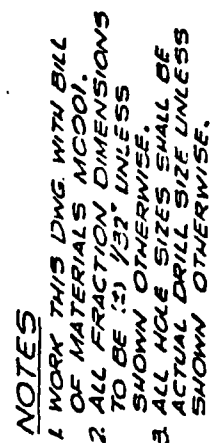
DETAIL ITEM 11 (C. STL.)
SCALE: 2"=1"

NOTES

- 1 WORK THIS DWG WITH BILL OF MATERIALS NOC001.
- 2 ALL FRACTIONAL DIMENSIONS TO BE (2) 1/4".
- 3 ALL ANGLES TO BE (2) 1/2".
- 4 ALL HOLE SIZES SHALL BE ACTUAL DRILL SIZE UNLESS SHOWN OTHERWISE.



DETAIL ITEM 10 (C. STL.)
SCALE: 2"=1"



DETAIL ITEM (24) (C. STEEL OR EQUAL)
FULL SCALE

APPENDIX C

Detailed Design for a Miniature Polar-coordinate Milling Machine

CAD-CAM DEPT.

BILL OF MATERIALS

POJAR MILL (M0039)

[illegible]

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Drawing No.		Sheet	
Prepared by		Date		Project No.		1 of	
FORREST BLAIR		11-10-92		MO039			
Approved by		Date		REVISIONS		DATE	
Project Engineer		Date		DESCRIPTION			
Project Name		PART NAME		UNITS		MATERIAL	
CAD-CAM		(Material Description)		REQUIRED			
Drawing Title							
POLAR MILL							
ITEM	DWG. NUMBER			UNITS			
				REQUIRED			
	0040	Polar Mill Assembly		1			
1	0040	Base Frame		1			C. Steel
2	0040	Base Cover		1			C. Steel
3	0041	Arch Frame		1			C. Steel
4	0041	Arch Rail		6			C. Steel
5	0041	Arch Track Clamp		6			Aluminum
6	0042	Arch Support		1			C. Steel
7	0042	Cable Support Shaft		1			C. Steel
8	0042	Cable Attachment		1			Aluminum
9	0042	Turret Support		1			C. Steel
10	0042	Turret Support Shaft		1			C. Steel

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

COASTMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOGRAPH TO USPOD AND CANCELLING AGENCY

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				NO.	11-10-92	2 of
Approved by				REVISIONS		DATE
Project Engineer				Description		
Project Name CAD-CAM				Date		
Drawing Title POLAR MILL				Date		
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
11	0043	Track Drive Plate	1	C. Steel		
12	0043	Inner Drive Screw	1	See Detail		
13	0043	Outer Drive Screw	1	See Detail		
14	0043	Turret Plate	1	Aluminum		
15	0043	Turret Chain Clamp	1	Aluminum		
16	0043	Inner Slide	1	Aluminum		
17	0043	Outer Slide	1	Aluminum		
18	0044	Cutter Support	1	Aluminum		
19	0044	Cutter Support Shaft	1	C. Steel		
20	0044	Cutter Drive Support	1	Aluminum		
21	0044	Slide Track	4	See Detail		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				NO.	11-10-92	3 of
Approved by				REVISIONS		DATE
Project Engineer				Description		
Project Name CAD-CAM				Date		
Drawing Title POLAR MILL				Date		
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
22	0044	Slide Top Cap	1	Aluminum		
23	0044	Slide Bottom Cap	1	Aluminum		
24	0044	Turret Roller Support	9	Aluminum		
25	0044	Upper Ball Nut Support	1	Aluminum		
26	0044	Upper Shaft Support	1	Aluminum		
27	0044	Lower Ball Nut Support	1	Aluminum		
28	0039	Motor - DC Stepping, S10-syn. #W061-PC02	4	Mfg. Std. Superior Electric		
29	0039	Motor - 1/15 H.P., 3U1-005K	1	Mfg. Std. Stock Drive Products		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				11-10-92	MO039	4 of
Approved by				NO.	REVISIONS	DATE
Project Engineer				Date	DESCRIPTION	DATE
Project Name CAD-CAM				Date		
Drawing Title POLAR MILL						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
20	0039	Cable Reel - "Negator" ML 1851	2	Mfg. Std.		
				Stock Drive		
				Products		
21	0039	Chain - 24 Pitch, 24GCF-319-E, 919 Pitches, 107.21" circ.	1	Mfg. Std.		
				W. Berg		
22	0039	Chain - 16 Pitch, 16GCF-10 FT., 563 Pitches	1	Mfg. Std.		
				W. Berg		
23	0039	Sprocket - .15709 Pitch, 3/4" P.D., 3MP26S-15	1	Mfg. Std.		
				W. Berg		
34	0039	Sprocket - 1.200" P.D., 24 Teeth, .15709 Pitch, 3MP26S-24	2	Mfg. Std.		
				W. Berg		
35	0039	Chain - .15709 Pitch, 40 Pitches, 6.293" circ., 3CCF-40-E	1	Mfg. Std.		
				W. Berg		
36	0039	Chain - .15709 Pitch, 90 Pitches, 14.137" circ., 3CCF-90-E	1	Mfg. Std.		
				W. Berg		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				11-10-92	MO039	5 of
Approved by				NO.	REVISIONS	DATE
Project Engineer				Date	DESCRIPTION	DATE
Project Name CAD-CAM				Date		
Drawing Title POLAR MILL						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
37	0039	Bearing - Radial Thrust, .9843" Bore, 7205W	2	Mfg. Std.		
				Fafnir		
38	0039	Bushing - 1/4" I.D., 11/16" O.D., 5/8" Long, 34739 B811-5	1	Mfg. Std.		
				Boston Gear		
39	0039	Washer - Thrust, 1/4" Bore, B5-3-SS	4	Mfg. Std.		
				W. Berg		
40	0039	Bushing - 5/16" I.D., 1/4" O.D., 1 1/4" Long, Plain Cylindrical, 34630 B59-12	4	Mfg. Std.		
				Boston Gear		
41	0039	Sprocket - 3" P.D., .15709" Pitch, 3MF17S-60 (Drill 4 holes to match item 18)	1	Mfg. Std.		
				W. Berg		
42	0039	Washer - Thrust, 1/4" Bore, B5-6-SS	2	Mfg. Std.		
				W. Berg		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
FORREST BLAIR				11-10-92	MO039	6 of
Prepared by	Date	NO.	REVISIONS			
FORREST BLAIR	11-10-92		DESCRIPTION	DATE		
Approved by	Date					
Project Engineer	Date					
Project Name						
CAD-CAM						
Drawing Title						
POLAR MILL						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
43	0039	Sprocket - 24 Pitch, 2 1/2" P.D., 24B4-60	1	Mfg. Std. W. Berg		
44	0039	Sprocket - 16 Pitch, 1" P.D., 16B4-16	1	Mfg. Std. W. Berg		
45	0039	Speed Reducer - 2:1 Ratio, 2222-2RR0203	1	Mfg. Std. W. Berg		
46	0039	Coupling - Wafer Spring, 1/2" Bore, C020-14	2	Mfg. Std. W. Berg		
47	0039	Bearing - 3/8" Nom. Bore, 7/8" O.D., 7Y55-P9737	3	Mfg. Std. W. Berg		
48	0039	Guide Wheel - WIX with BX1 Adapter Bushing, Dualvee	3	Mfg. Std. Bishop- Wisecarver		

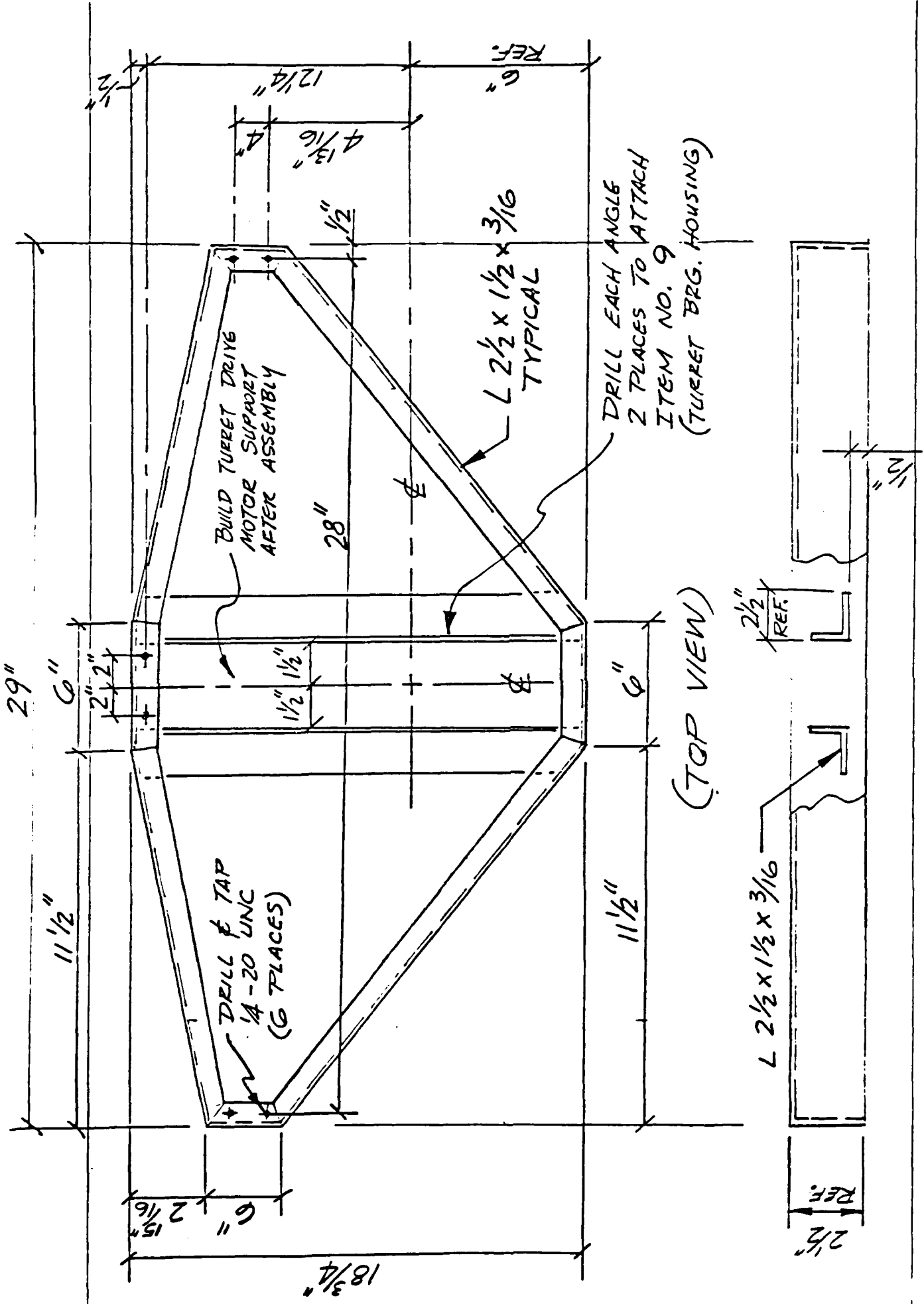
DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL. SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

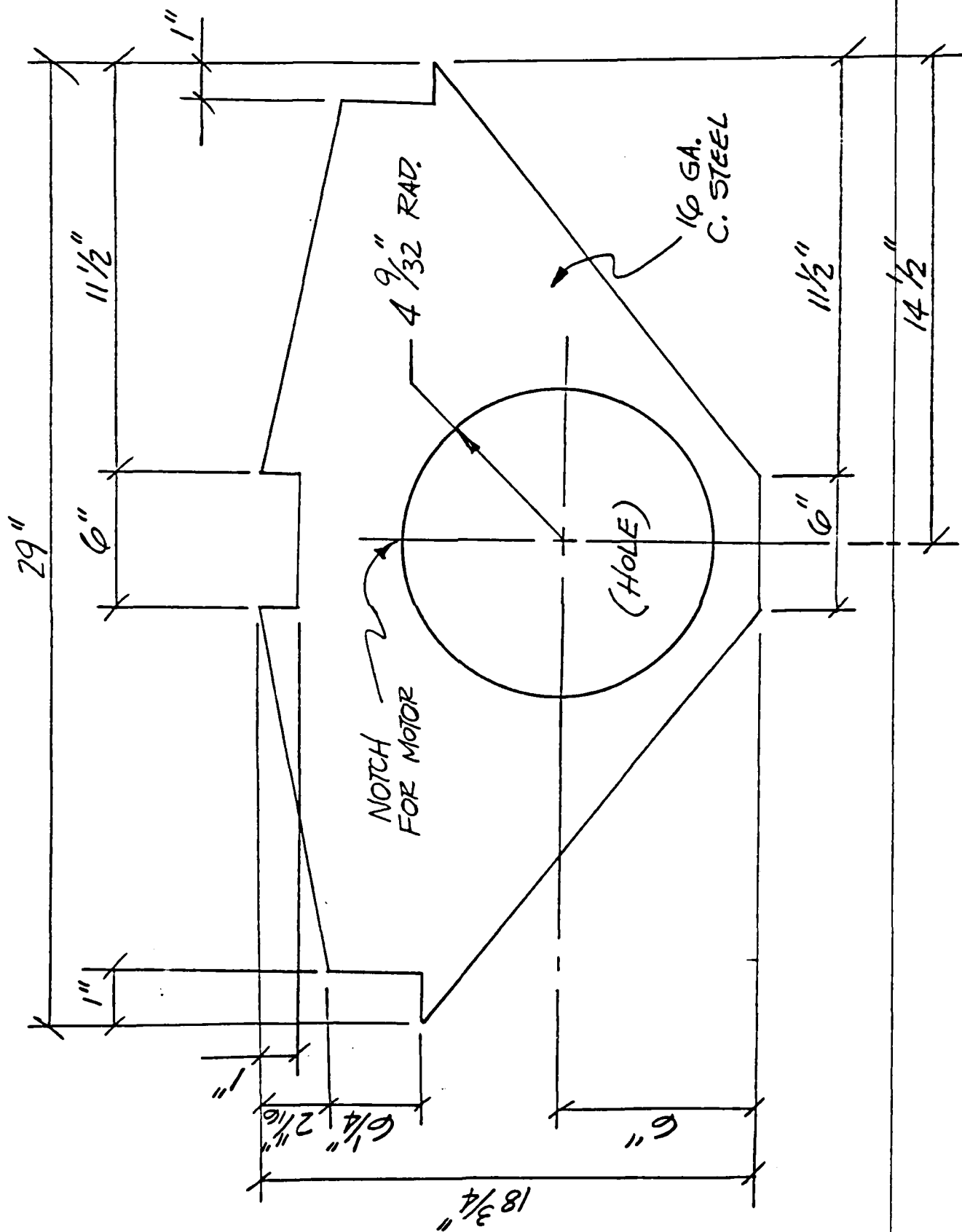
BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
FORREST BLAIR				11-10-92	MO039	7 of
Prepared by	Date	NO.	REVISIONS			
FORREST BLAIR	11-10-92		DESCRIPTION	DATE		
Approved by	Date					
Project Engineer	Date					
Project Name						
CAD-CAM						
Drawing Title						
POLAR MILL						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
49	0039	Guide Wheel - W2X with BX2 Adapter Bushing, Dualvee	4	Mfg. Std. Bishop- Wisecarver		
50	0039	Ball Nut - 3/8" Nom., RN-0375-0125	2	Mfg. Std. Raceaway		
51	0039	Collar - Set, 1/2" Bore	1	As Req'd.		
52	0039	Coupling - Solid, 1/2" Bore, 1 1/2" Long	2	As Req'd.		
53	0039	Pipe - 3/8" Nom., Sch. 40 x 1 3/16" Long	1	As Req'd.		
54	0039	Screw - Socket Head Shoulder, 1/4" - 20 UNC, 3/8" Dia. x 1/2" Long Shoulder, 9X25-1216	3	Mfg. Std. Stock Drive Products		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

SMALL 6" CUBE
POLAR MILL

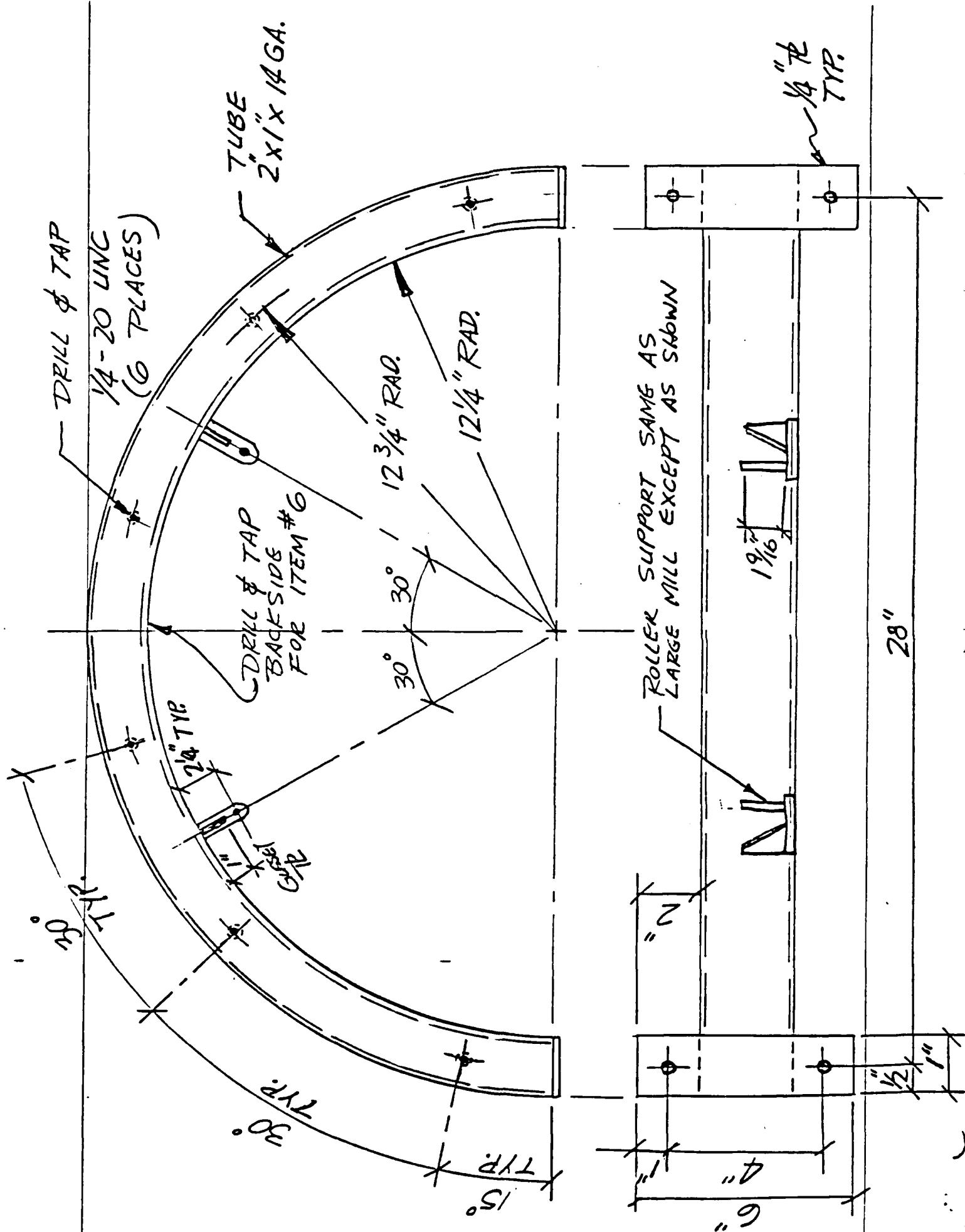


ITEM 2 BASE COVER (CARBON STEEL)



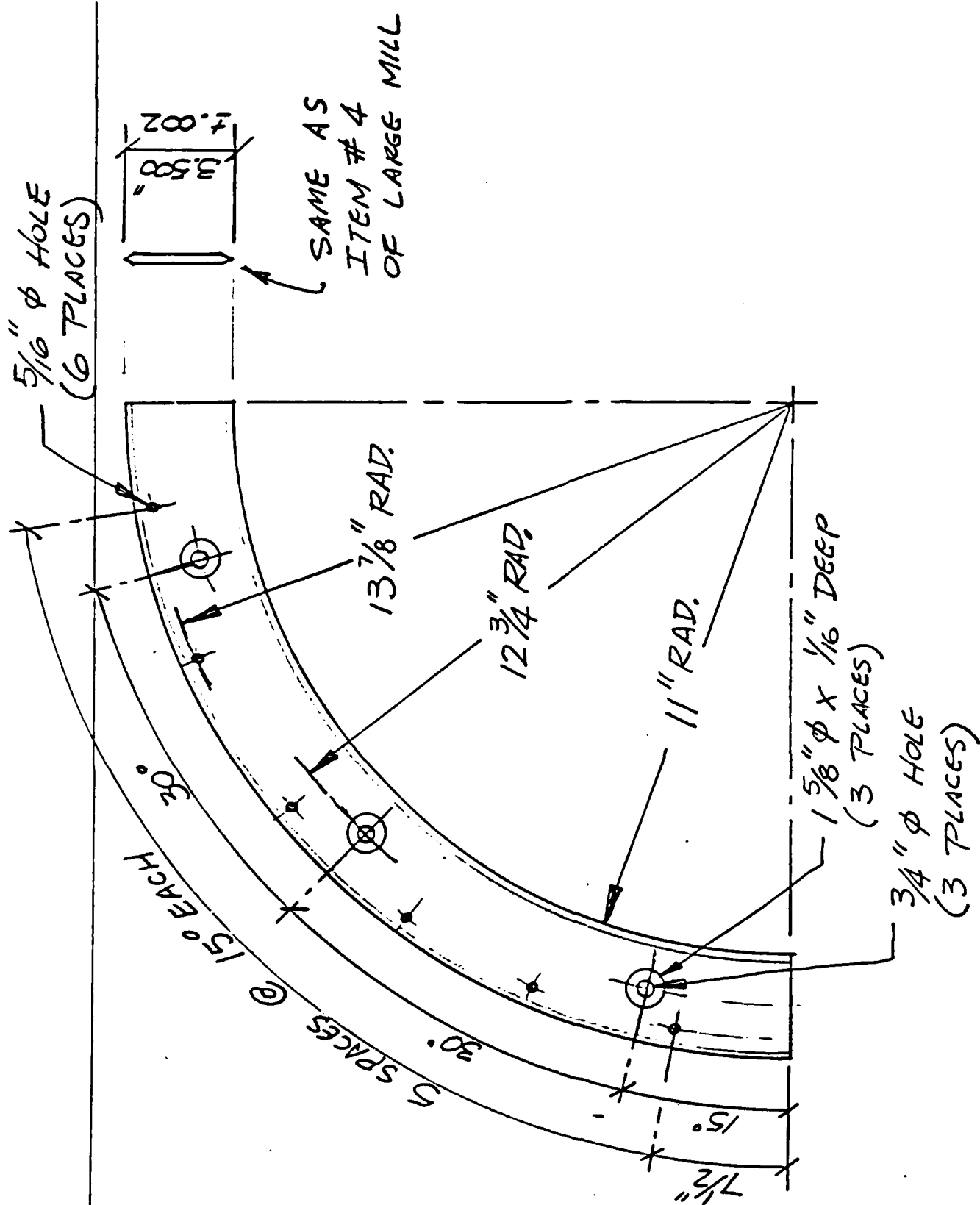
5 WEST

ARCH FRAME (C. STEEL)

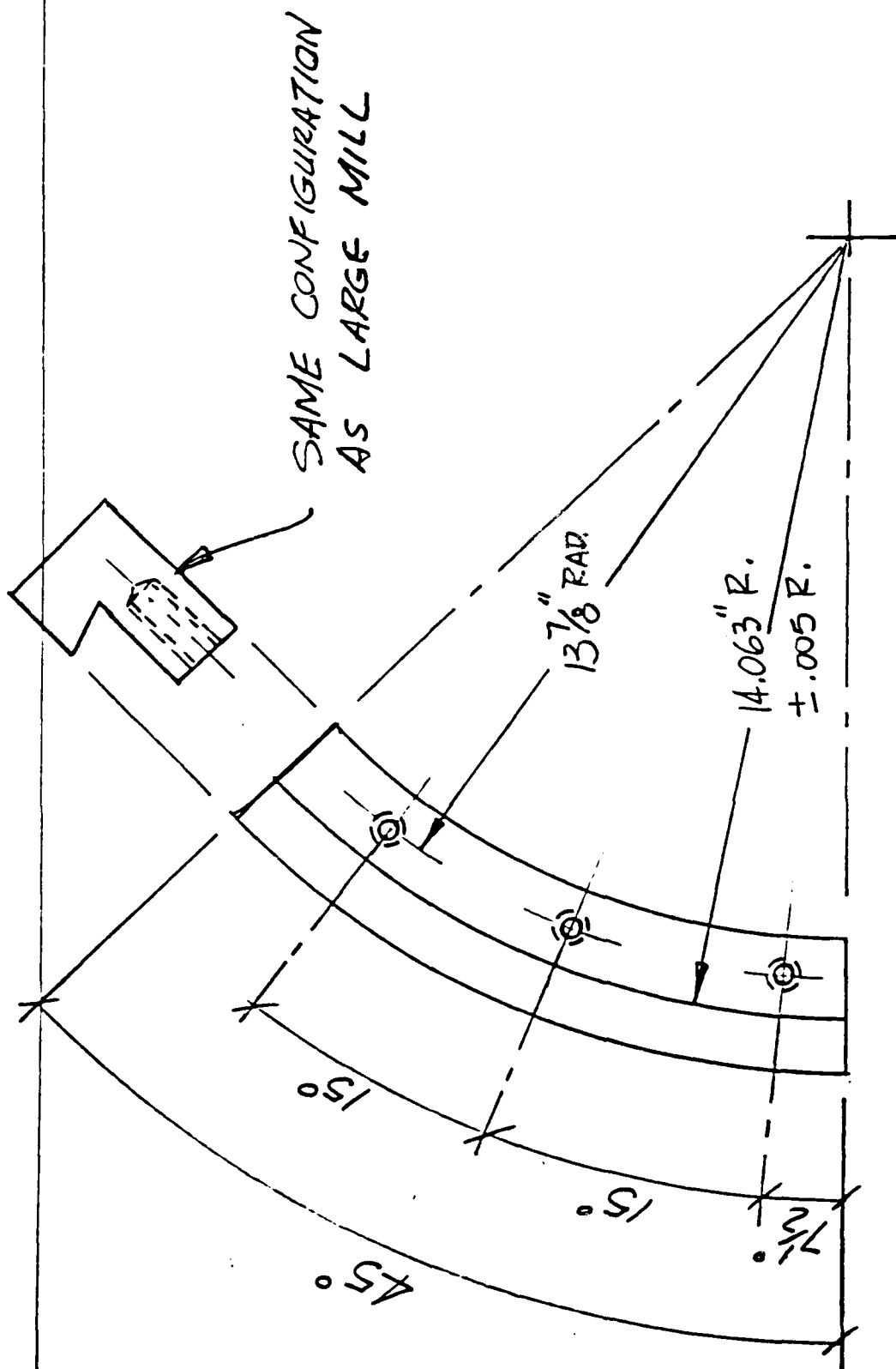


ITEM 4 ARCH RAIL (C. STEEL)

2 REQUIRED

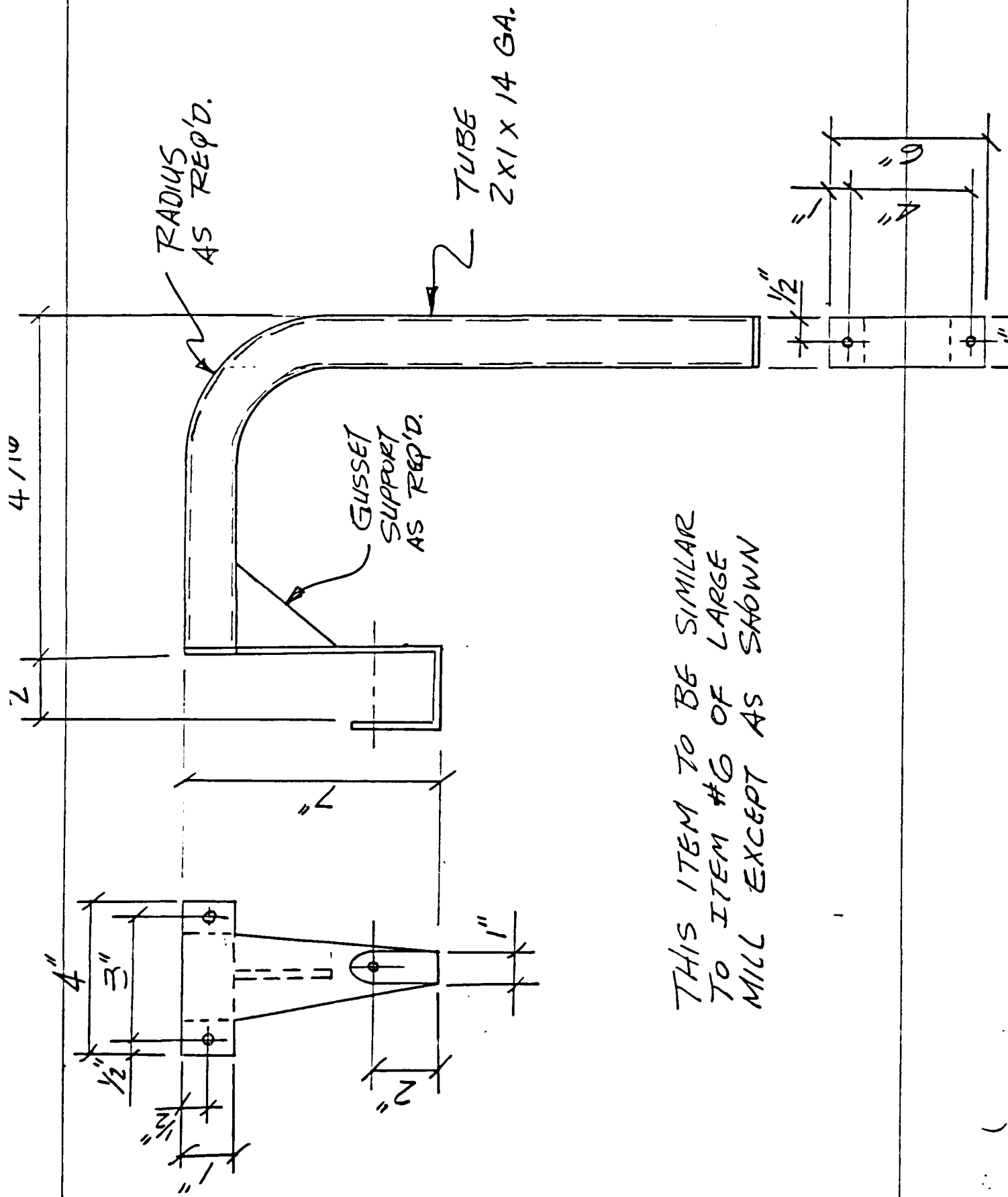


ITEM 5
ARCH TRACK CLAMP
ALUMINUM
OR STEEL
(4 REQUIRED)



ITEM の

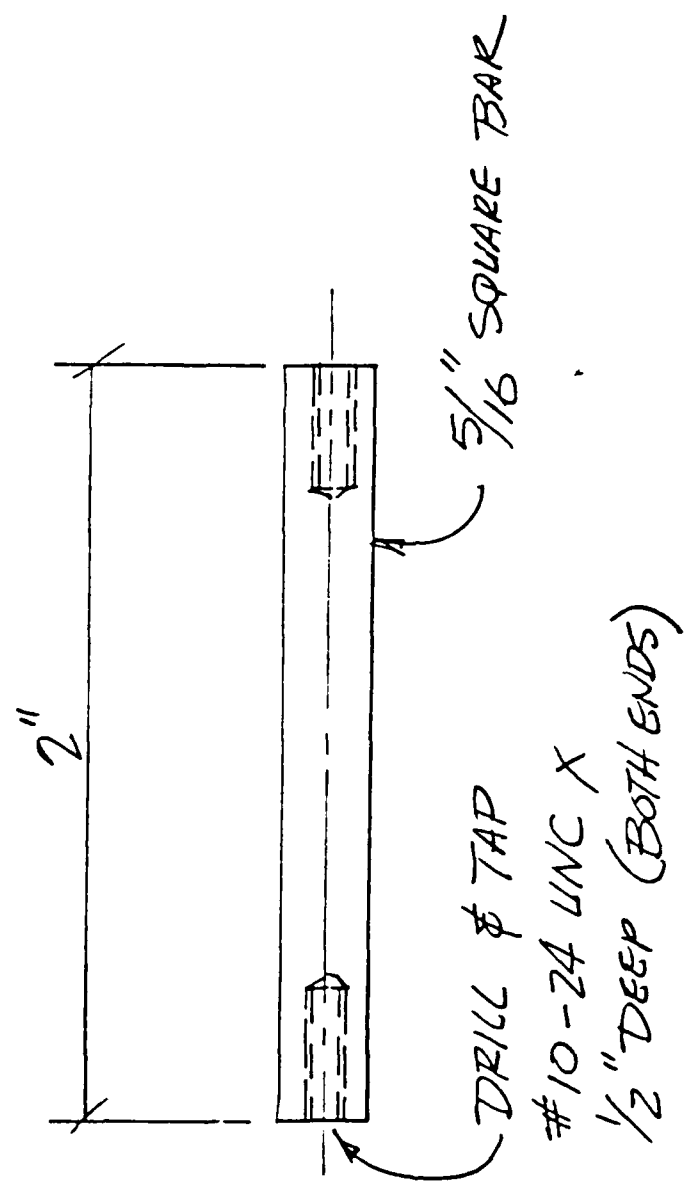
(C. STEEL)



CABLE SUPPORT SHAFT

ITEM 7

(C. STEEL)



AD-A152 715

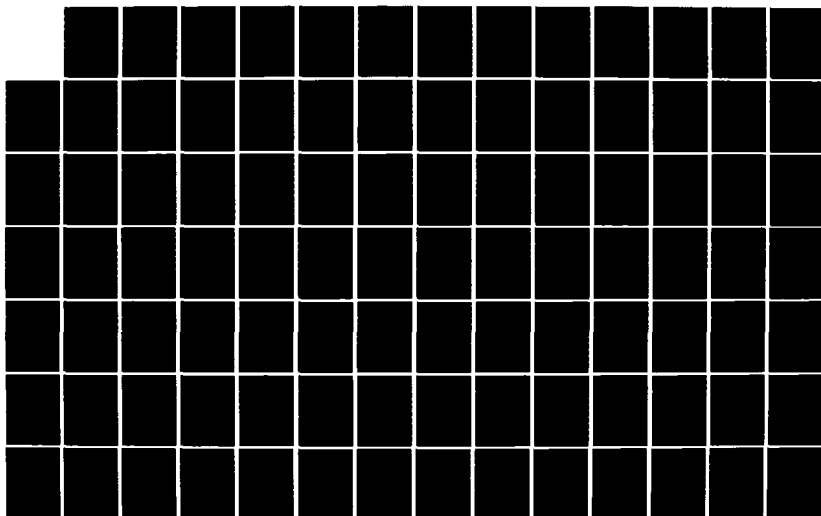
MANUFACTURING INFORMATION SYSTEM(U) BRIGHAM YOUNG UNIV
PROVO UT COMPUTER AIDED MFG LAB D K ALLEN ET AL.
26 DEC 84 AFOSR-TR-85-0275 AFOSR-82-0253

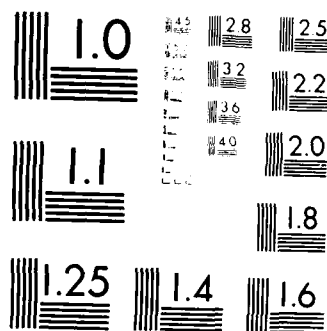
2/3

UNCLASSIFIED

F/G 13/8

NL

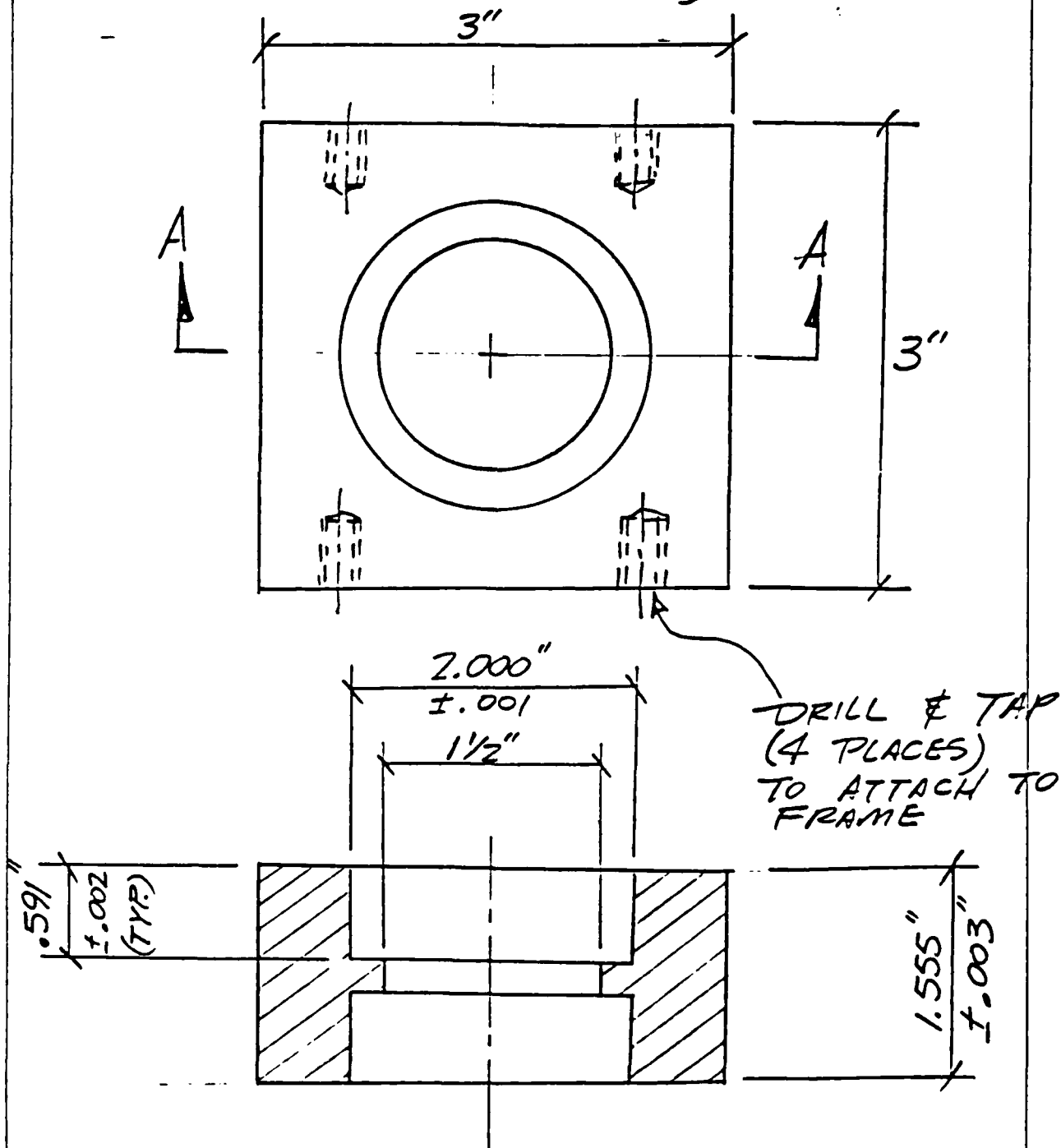




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

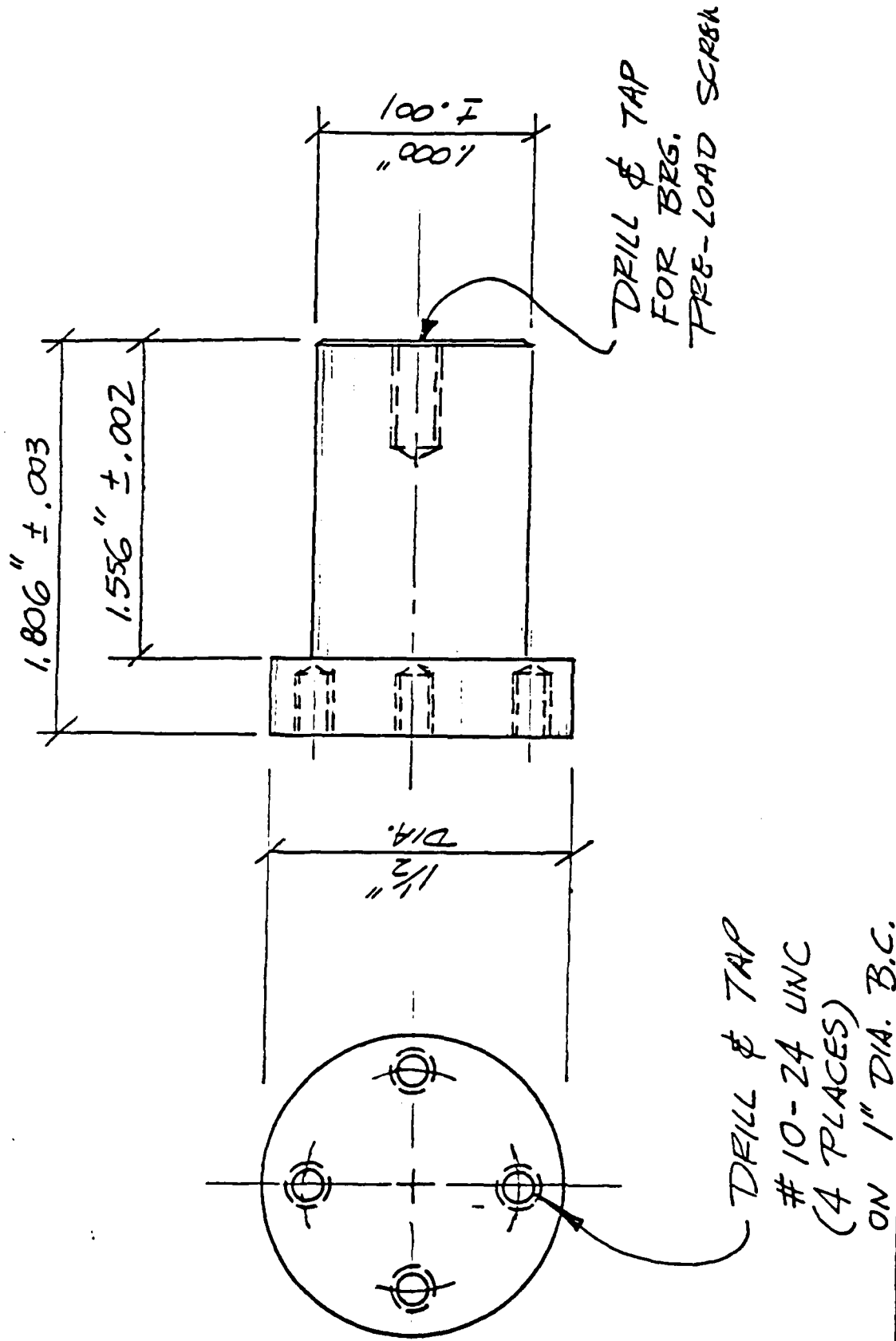
ITEM 8 — SAME AS LARGE MILL

ITEM 9 TURRET BRG. HOUSING
(C. STEEL)

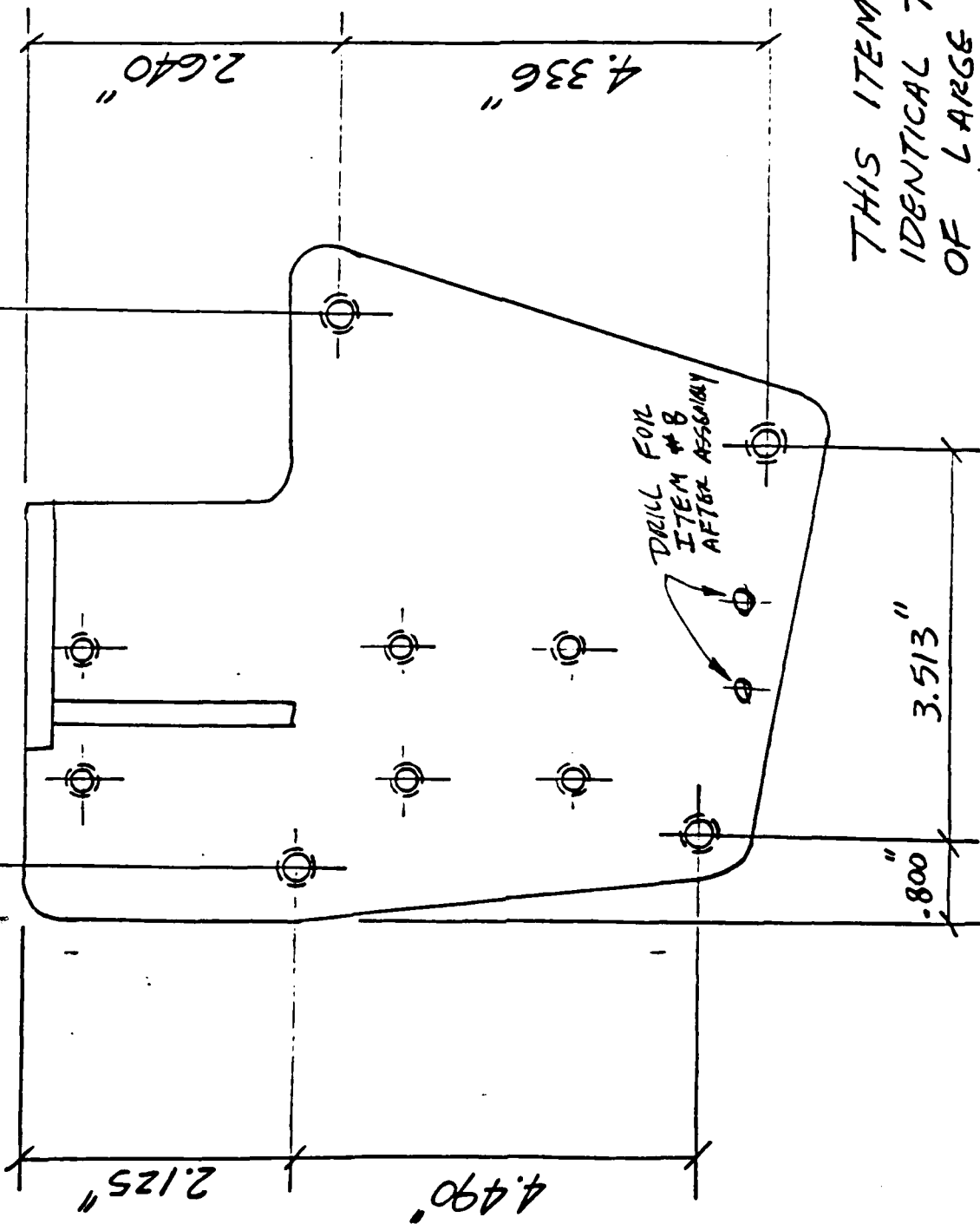


SECTION A-A

ITEM 10 TURRET SHAFT (C.STEEL)

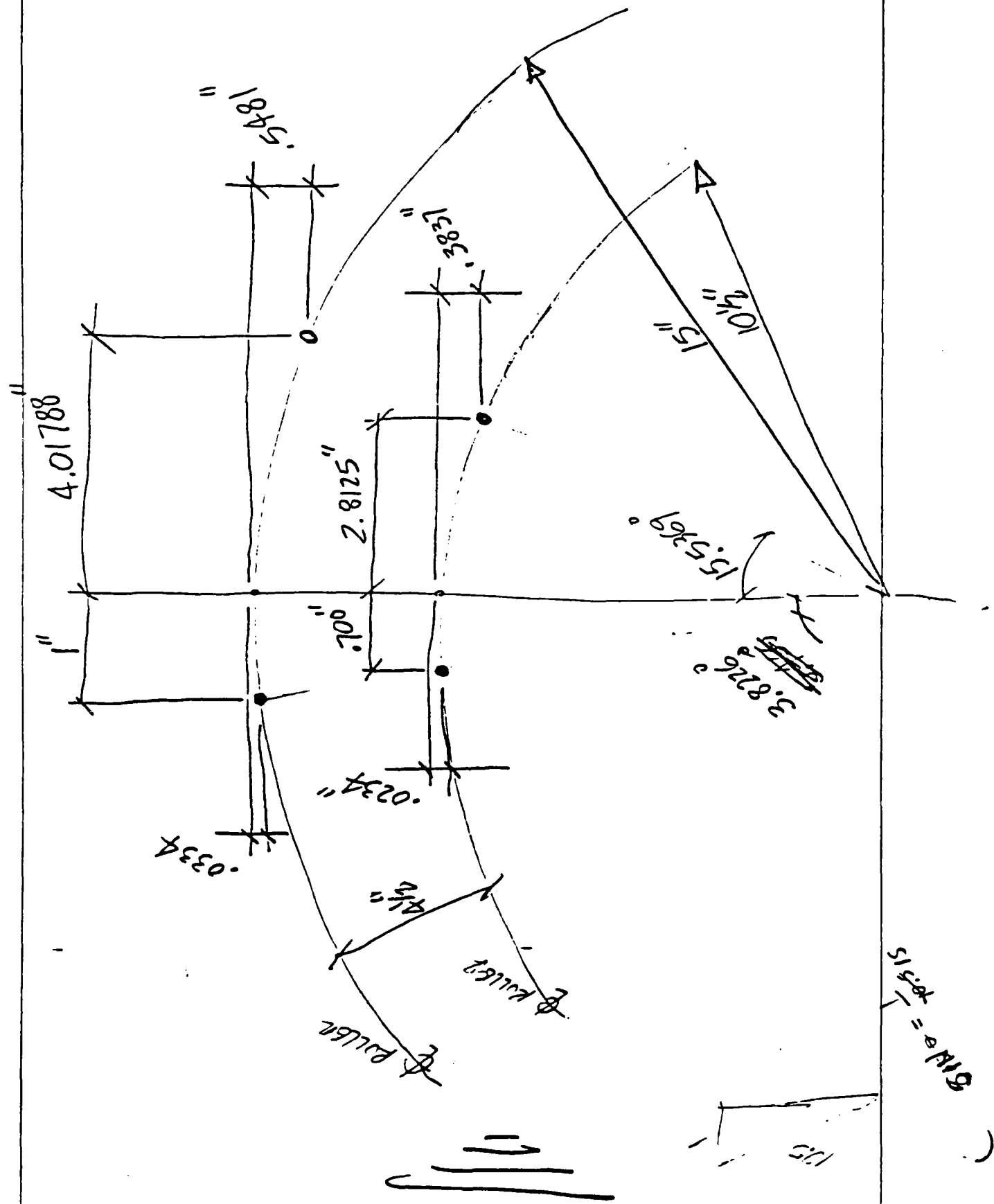


ITEM 11 TRACK DRIVE H (C. STEEL) OR ALUM.



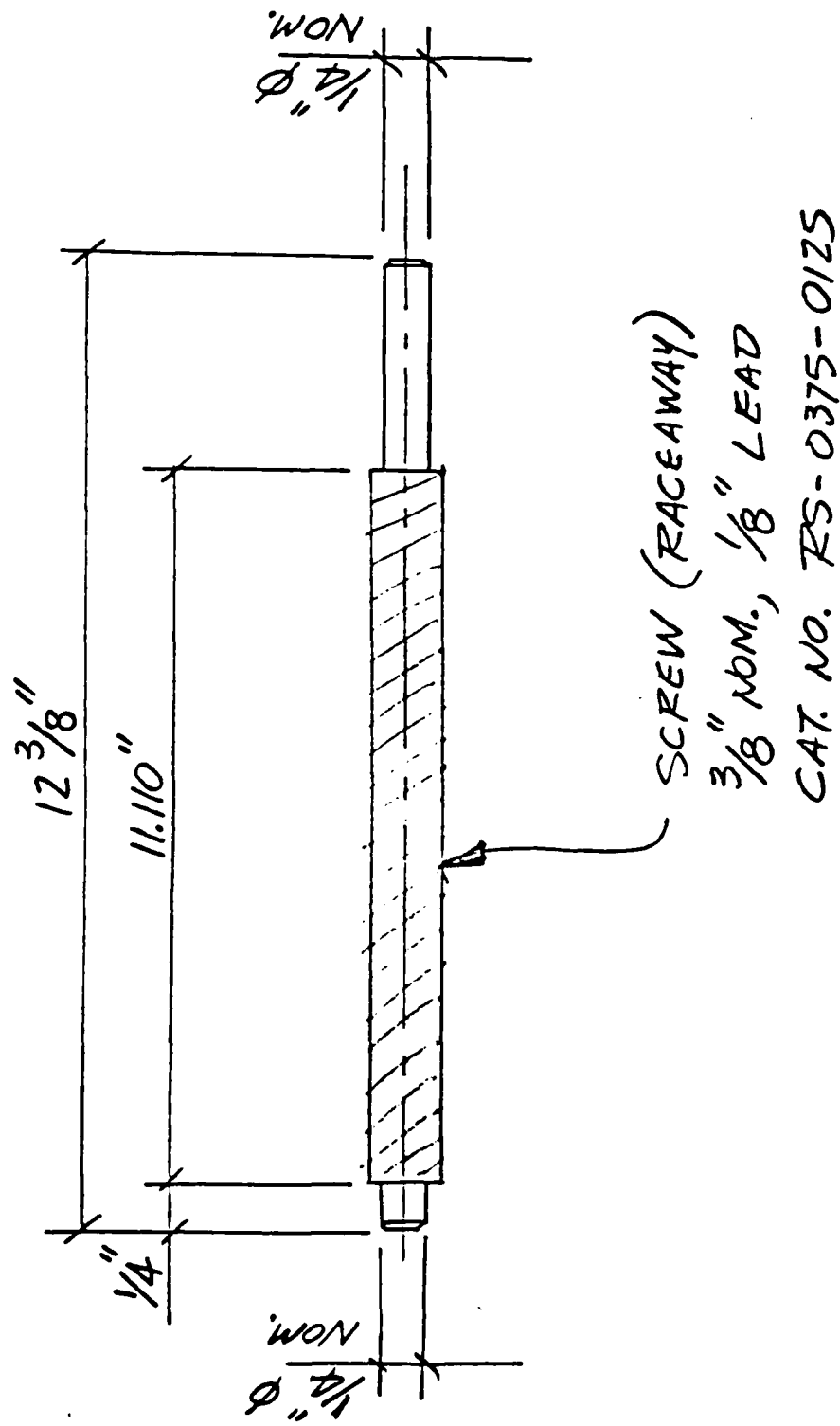
THIS ITEM TO BE IDENTICAL TO ITEM 11 OF LARGE MILL EXCEPT WITH DIMENSIONS AS SHOWN

WORK SHEET FOR ITEM 11

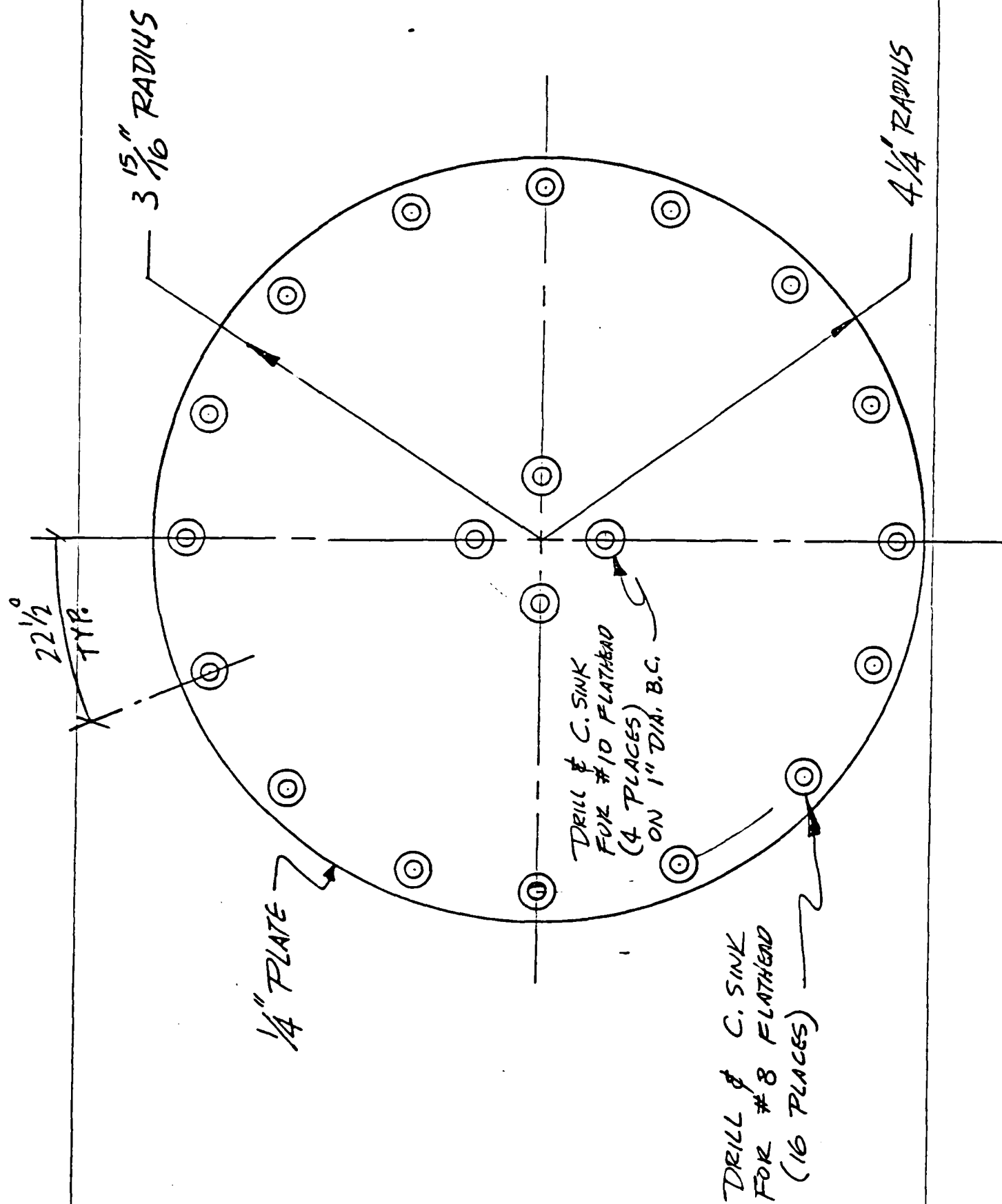


ITEM 12 - NOT NEEDED

ITEM 13 SLIDE DRIVE SCREW

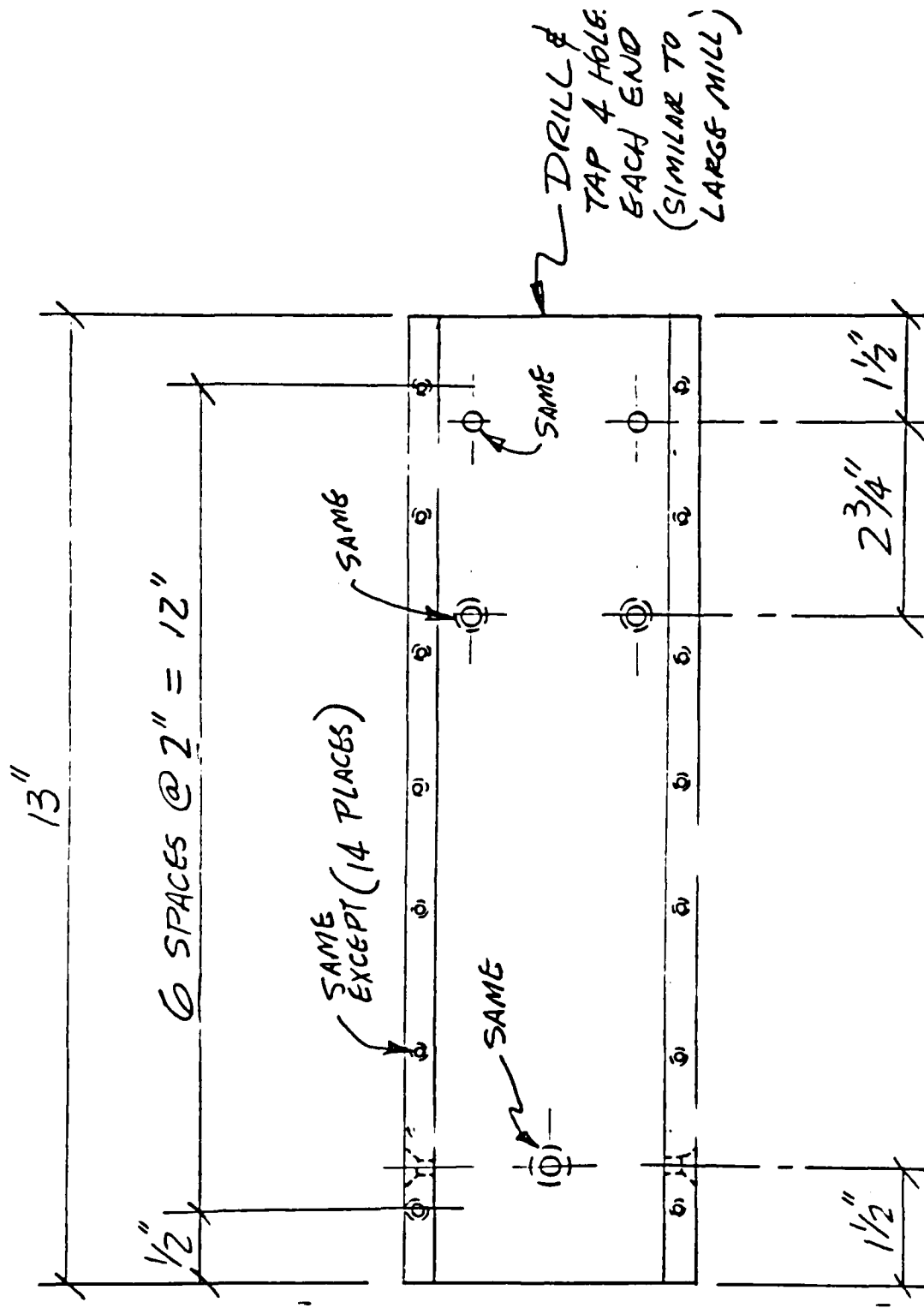


ITEM 14 TURRET PLATE (ALUMINUM)



ITEM 15 & 16 - NOT NEEDED

ITEM 17 SLIDE (ALUMINUM)



THIS ITEM TO BE IDENTICAL TO
ITEMS 16 & 17 OF LARGE MILL EXCEPT
AS SHOWN

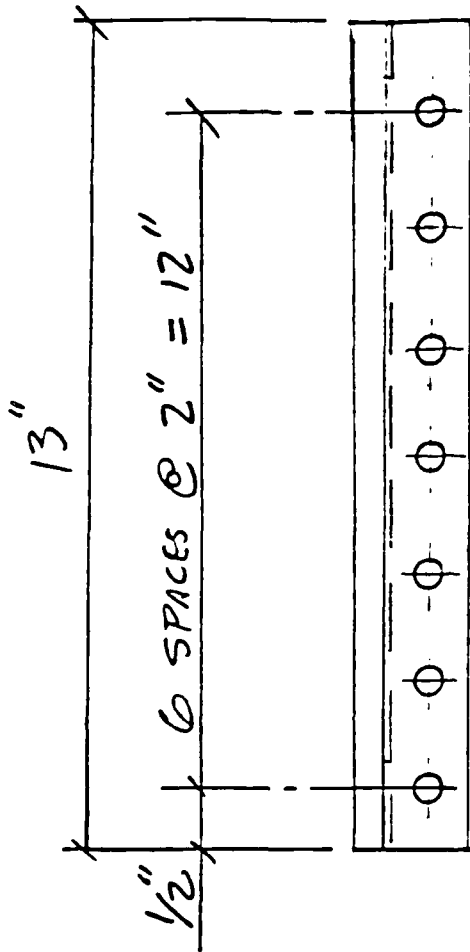
ITEM 18 - SAME AS LARGE MILL

ITEM 19 CUTTER SUPPORT SHAFT

SAME AS LARGE MILL EXCEPT

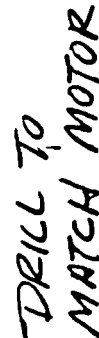
PLATE IS 3" X 3" INSTEAD OF 3 1/4" X 3"

ITEM 21 SLIDE TRACK



THIS ITEM TO BE
IDENTICAL TO ITEM # 21
OF LARGE MILL EXCEPT
AS SHOWN

(C. STEEL OR ALUMINUM)

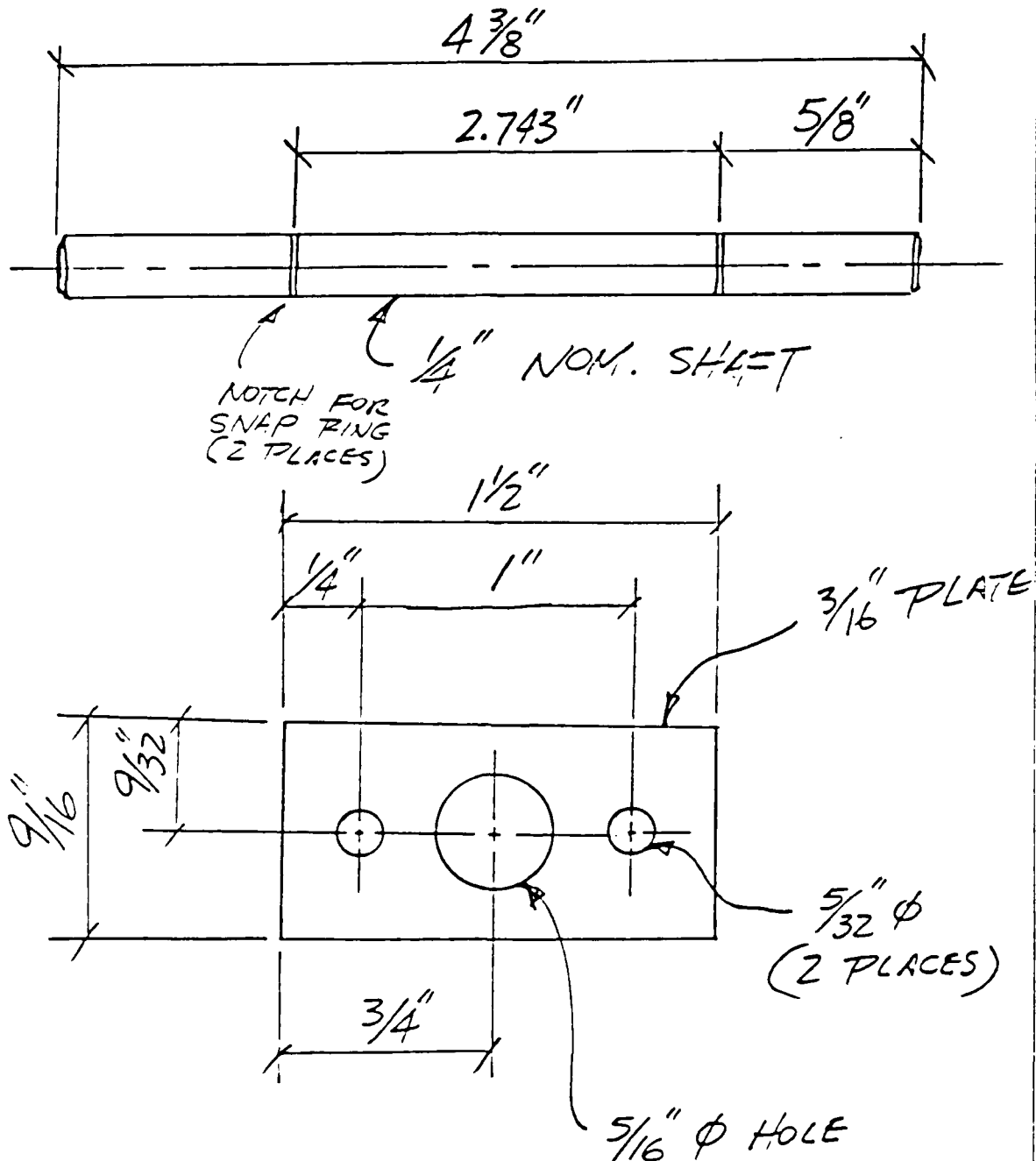


DRILL & TAP
#6-32 UNC
(2 PLACES)

5 1/16" ϕ HOLE
1" WIDE X 9 1/16" HIGH
BLOCK X 5 1/16" THICK
BORE .6250" DIA.
X 3 1/16" DEEP

SLOT FOR
#10 SCREW
(2 PLACES)

ITEMS THAT GO WITH ITEM # 20
 BEARING - $\frac{1}{4}$ " BORE, $\frac{5}{8}$ " NOM. OD., 2 REQUIRED
 THRUST WASHER - $\frac{1}{4}$ " BORE, 2 REQUIRED
 SNAP RINGS - FOR $\frac{1}{4}$ " SHAFT, 2 REQUIRED
 WORM - 24 PITCH, $\frac{1}{4}$ " BORE, $\frac{1}{2}$ " NOM. P.D.,
 STOCK DRIVE PRODUCTS #. 1Q555-N24



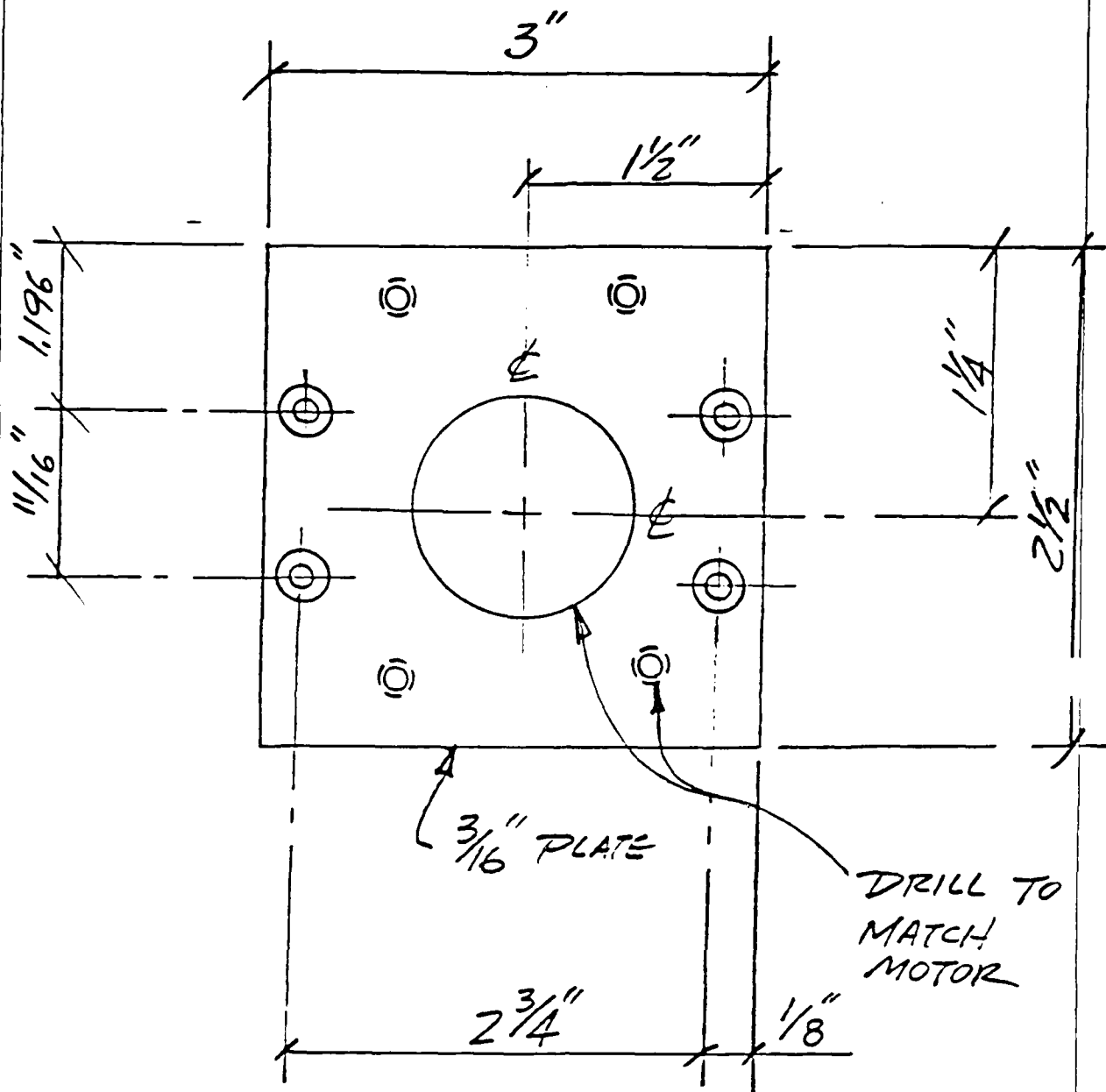
THRUST BRG. PRE-LOAD PLATE

ITEM 22

SLIDE TOP

CAP

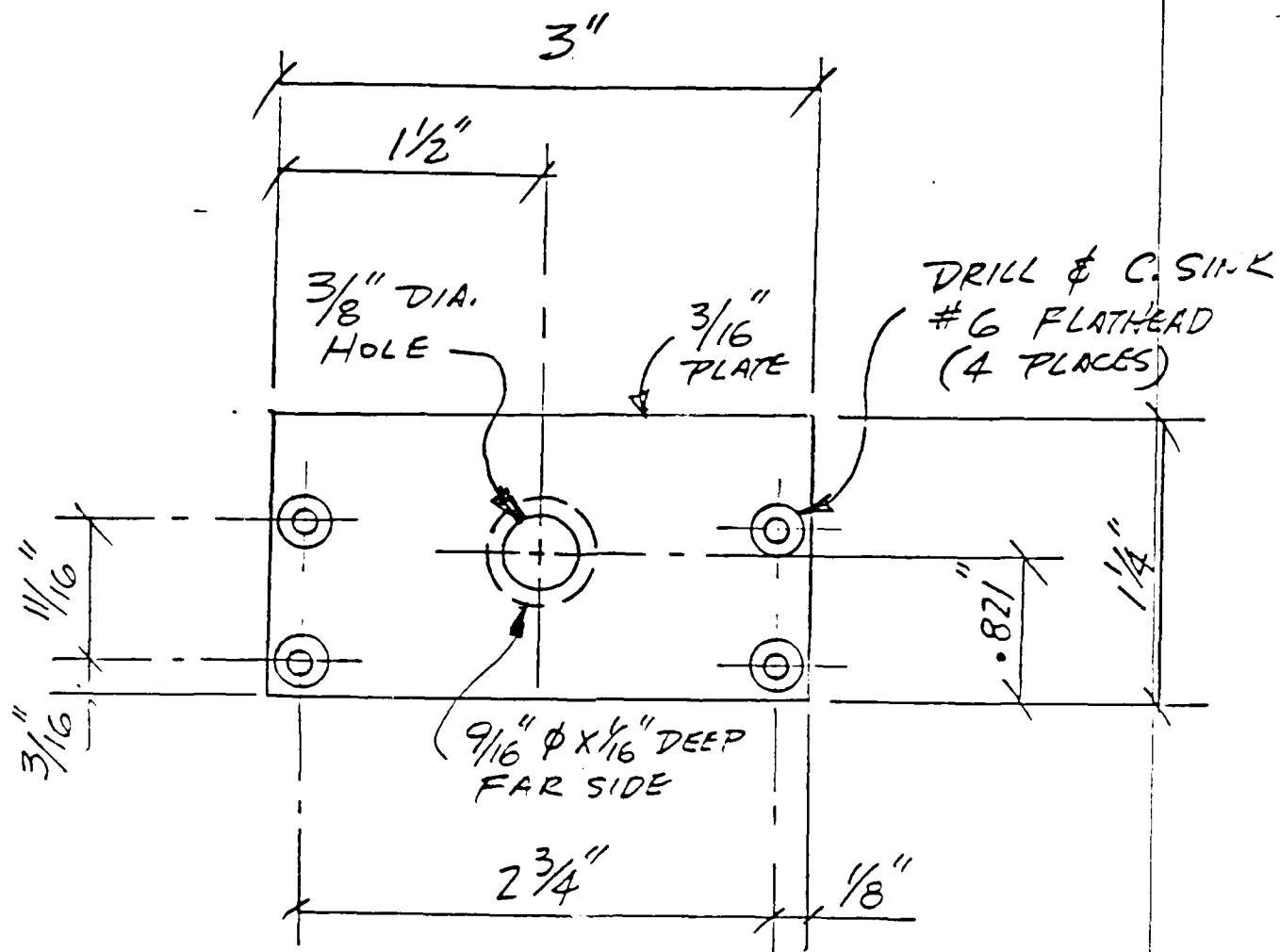
(ALUMINUM)

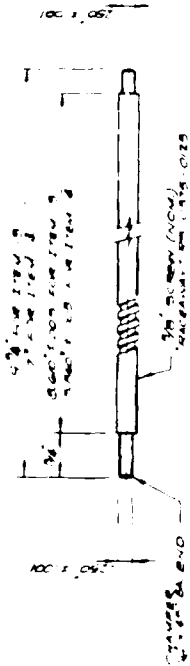


ITEM 23

SLIDE BOTTOM CAP

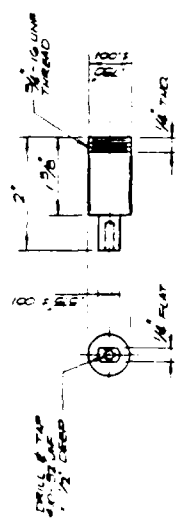
(ALUMINUM)



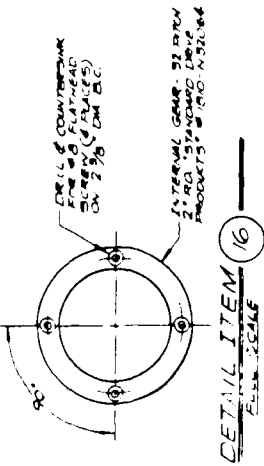


DETAIL ITEM 13
FULL SCALE

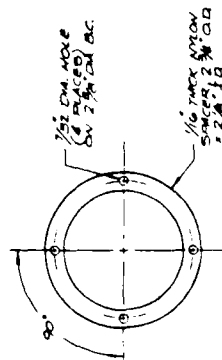
DETAIL ITEM 14
FULL SCALE



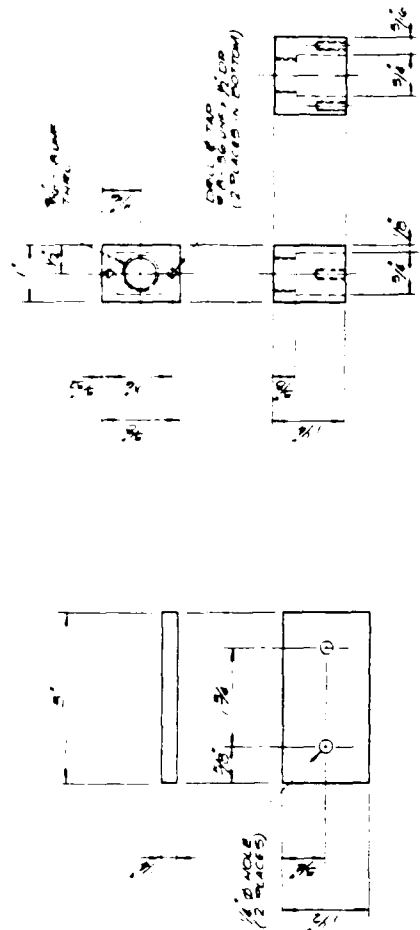
DETAIL ITEM 15
FULL SCALE



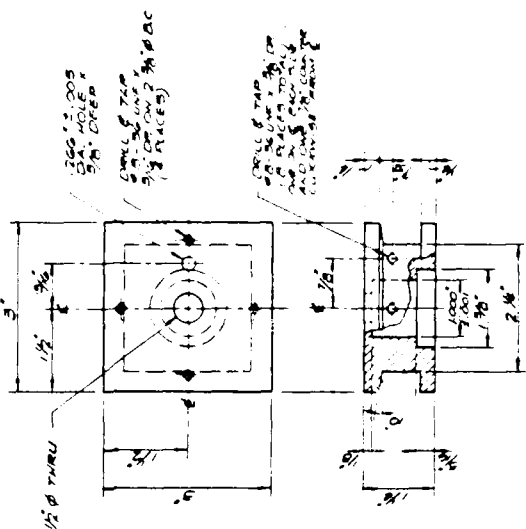
DETAIL ITEM 16
FULL SCALE



DETAIL ITEM 17
FULL SCALE

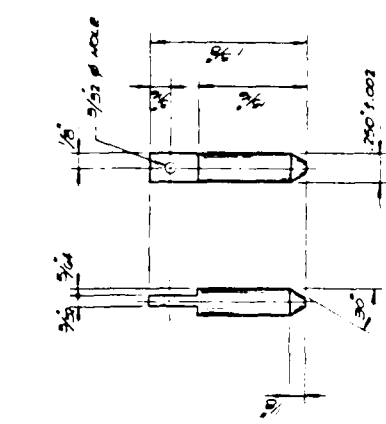


DETAIL ITEM 18
FULL SCALE



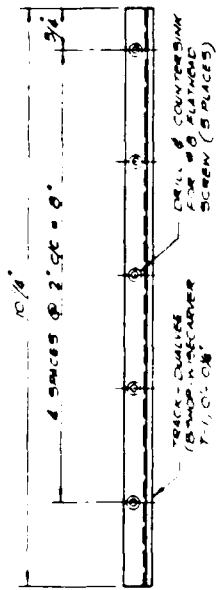
DETAIL ITEM 20
FULL SCALE

DETAIL ITEM 19
FULL SCALE

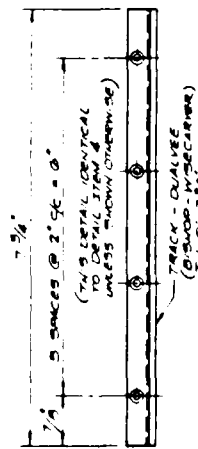


DETAIL ITEM 21
FULL SCALE

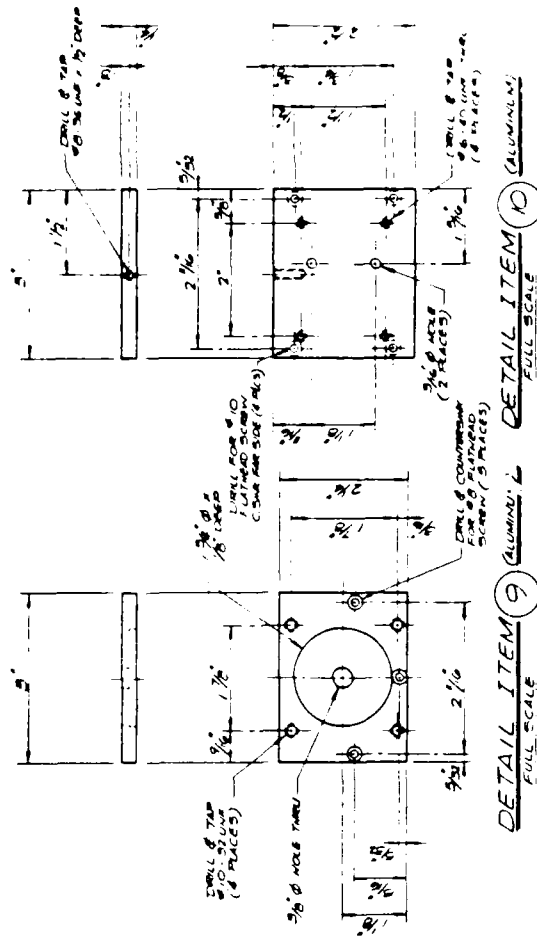
NOTES:
1. MOD. TAP DR. WITH FULL
2. MATERIALS: 6061-T6 ALUM.
3. ALL DIMENSIONS IN INCHES
4. UNLESS OTHERWISE SPECIFIED



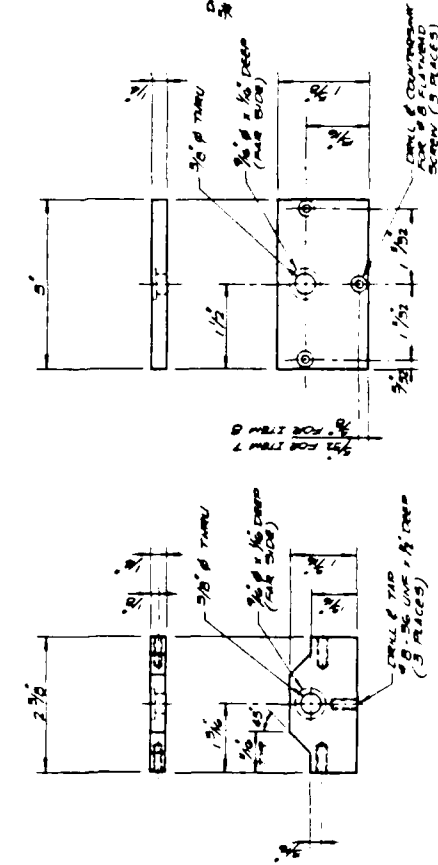
DETAIL ITEM 4
FULL SCALE



DETAIL ITEM 5
FULL SCALE

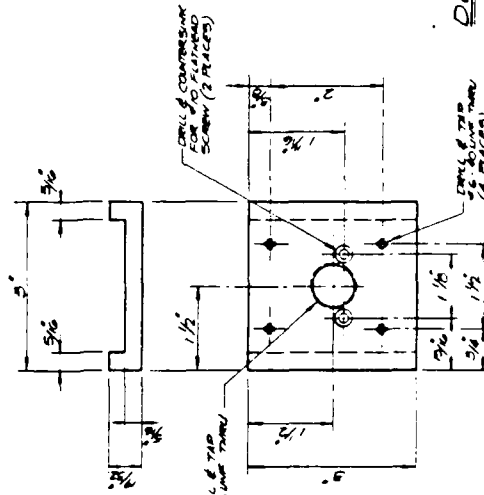


DETAIL ITEM 9
FULL SCALE

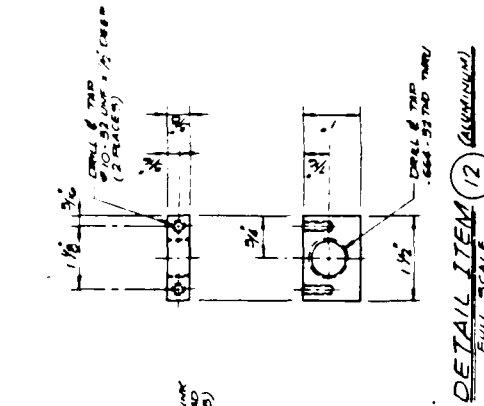


DETAIL ITEM 7
FULL SCALE

DETAIL ITEM 6
FULL SCALE



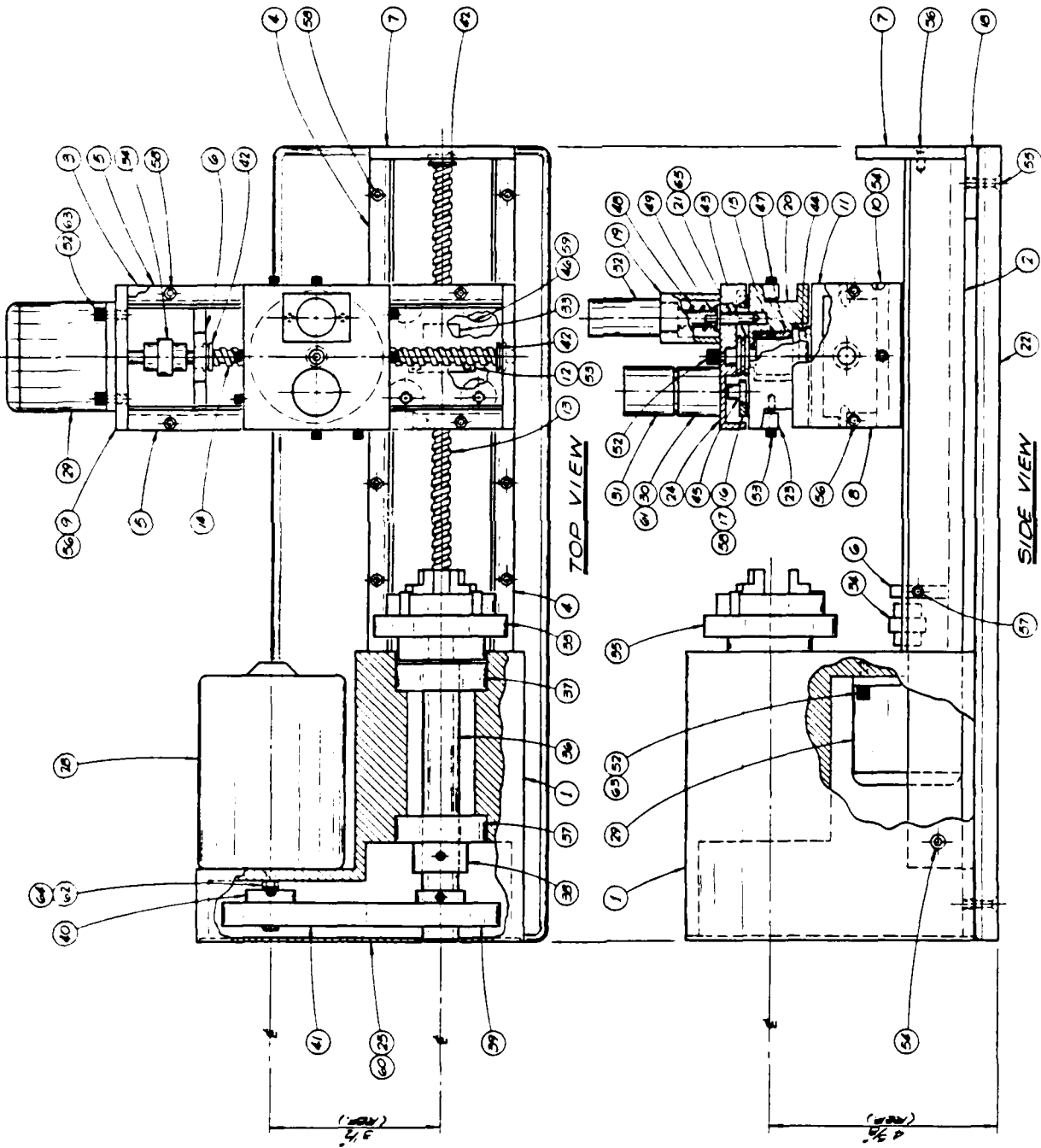
DETAIL ITEM 11
FULL SCALE



DETAIL ITEM 12
FULL SCALE

NOTES
1. WORK THIS DRAWING WITH ALL
DIMENSIONS AND ALL
2. ALL DIMENSIONS
TO BE (2) 1/4".

BRIGHAM YOUNG UNIVERSITY	UTAH STATE	LATHIE	DETAIL ITEM 4, 5, 6, 7, 8, 9, 10, 11 & 12	FORREST BLAIR	CC 36
2, 982	880				
CAMP DEPT.					



NOTES
 1 ALL ITEMS MARKED THIS
 ARE FOUND ON BILL OF
 MATERIALS MOOS, 3
 2 SPOT WELD ITEM 15 TO ITEM 11
 AFTER ASSEMBLY.

[illegible]

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY		Project No.	Drawing No.	Sheet	
FORREST BLAIR		9-16-32	40033	6 of	
Approved by		NO.	REVISIONS	DATE	
Project Engineer					
Project Name					
CAD-CAM					
LATHÉ					
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE
45	0033	Gear - Spur, 40° pressure angle, 32 Pitch, .439" P.D., #1M2-N72014	1	Mfg. Std. Standard Drive Products	
46	0033	Guide Wheel - Dualvee, size 1, VIX, with Adapter Bushing BX1	3	Mfg. Std. Bishop- Viscarver	
47	0033	Bearing - Plain Bushing, 1" O.D., 3/4" I.D., #B1216-6	1	Mfg. Std. Boston	
48	0033	Spring - .954" height, #20	1	Mfg. Std. Century Spring	
49	0033	Pin - 3/32" ϕ x 5/16" long	1	C. Steel	
50		Not Used			
51		Not Used			
52	0033	Cap Screw - Hex Socket, #10-32UNF x 1/2" long	9	C. Steel	

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY		Project No.	Drawing No.	Sheet	
FORREST BLAIR		9-16-32	40033	7 of	
Approved by		NO.	REVISIONS	DATE	
Project Engineer					
Project Name					
CAD-CAM					
LATHÉ					
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE
53	0033	Cap Screw - Hex Socket, #9-36 UNF x 5/8" long	10	C. Steel	
54	0033	Screw - Flathead, Hex Socket, #10-32 UNF x 1/2" long	8	C. Steel	
55	0033	Screw - Flathead, Hex Socket, #10-32 UNF x 1" long	4	C. Steel	
56	0033	Screw - Flathead, Hex Socket, #9-36 UNF x 5/8" long	9	C. Steel	
57	0033	Screw - Flathead, Hex Socket, #8-36 UNF x 3/4" long	6	C. Steel	
58	0033	Screw - Flathead, Hex Socket, #8-36 UNF x 3/8" long	22	C. Steel	
59	0033	Screw - Flathead, Hex Socket, #6-40 UNF x 7/8" long	9	C. Steel	

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY		Project No.	Drawing No.	Sheet
FORREST BLAIR			MO033	4 of
Prepared by	Date	NO	REVISIONS	DATE
FORREST BLAIR	9-16-92			
Approved by	Date			
Project Engineer	Date			
Project Name				
CAD-CAM				
Drawing Title				
LATHE				
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	DATE
31	0033	Micromotor - #3P41MB3206	1	Mfg. Std. Standard Drive Products
32	0033	Solenoid - 3/4" tubular pull, Series 10, #196175-034	1	Mfg. Std. Ledex
33	0033	Ball Nut - 3/3" nom., RN-0375-0125	2	Mfg. Std. Raceaway
34	0033	Coupling - Wafer Spring, 1" Bore, Q020-14	2	Mfg. Std. Winfred Berg
35	0033	Chuck - 3 Jaw Universal, No. 300L, ML-1	1	Mfg. Std. Toyo
36	0033	Shaft - Chuck Drive, ML-1 Replacement Shaft	1	Mfg. Std. Toyo
37	0033	Bearing - Drive Shaft, ML-1 Replacement Bearing (1 ea. end)	2	Mfg. Std. Toyo

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY		Project No.	Drawing No.	Sheet	
FORREST BLAIR			MO033	5 of	
Prepared by	Date	NO	REVISIONS	DATE	
FORREST BLAIR	9-16-92				
Approved by	Date				
Project Engineer	Date				
Project Name					
CAD-CAM					
Drawing Title					
LATHE					
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE
38	0033	Shaft Clamp Nut - ML-1 Replacement Nut	1	Mfg. Std. Toyo	
39	0033	Pulley - 3/3" Pitch, #37TP4-21, Bore to 1 1/16" dia.	1	Mfg. Std. Winfred Berg	
40	0033	Pulley - 3/8" Pitch, #37TP4-16, Bore to 5/16" dia.	1	Mfg. Std. Winfred Berg	
41	0033	Belt - Timing, 3/8" Pitch, #37TB-40, 15" Pitch Length	1	Mfg. Std. Winfred Berg	
42	0033	Bearing - Thrust, 1" Bore, #B5-3-SS	4	Mfg. Std. Winfred Berg	
43	0033	Bearing - Thrust, 3/3" Bore, #B5-5-SS	1	Mfg. Std. Winfred Berg	
44	0033	Bearing - Thrust, 3/4" Bore, #B5-8-SS	1	Mfg. Std. Winfred Berg	

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				NO.	REVISIONS DESCRIPTION	DATE
Approved by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer				NO.	REVISIONS DESCRIPTION	DATE
Project Name CAD-CAM				NO.	REVISIONS DESCRIPTION	DATE
Drawing Title LATHE				NO.	REVISIONS DESCRIPTION	DATE
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
11	0036	Upper Roller Plate	1	Aluminum		
12	0036	Ball Nut Plate	2	Aluminum		
13	0037	Lower Screw	1	See Detail		
14	0037	Upper Screw	1	See Detail		
15	0037	Turret Shaft	1	C. Steel		
16	0037	Turret Gear	1	See Detail		
17	0037	Turret Gear Spacer	1	Nylon		
18	0037	Base Plate Spacer	1	Aluminum		
19	0037	Solenoid Block	1	Aluminum		
20	0037	Turret	1	Aluminum		
21	0037	Turret Stop Pin	1	Aluminum		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				NO.	REVISIONS DESCRIPTION	DATE
Approved by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer				NO.	REVISIONS DESCRIPTION	DATE
Project Name CAD-CAM				NO.	REVISIONS DESCRIPTION	DATE
Drawing Title LATHE				NO.	REVISIONS DESCRIPTION	DATE
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
22	0032	Base Plate	1	Steel		
23	0032	Tool Holder	4	Aluminum		
24	0039	Turret Cap	1	Aluminum		
25	0039	Cover Plate	1	Aluminum		
26		Not Used				
27		Not Used				
28	0033	Motor - Universal AC-DC, 1/5 H.P.	1	Mfg. Std. Standard Drive Products		
29	0033	Motor - DC Stepping, Slo-syn, #M061-FC02	2	Mfg. Std. Superior Electric		
30	0033	Gearhead - 320:1 Ratio,	1	Mfg. Std. Standard Drive Products		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

APPENDIX D

Detailed Design of Computer Controlled Lathe

ITEM 42 - SAME AS LARGE MILL
ITEM 43 - SAME AS LARGE MILL
ITEM 44 - SAME AS LARGE MILL
ITEM 45 - SAME AS LARGE MILL
ITEM 46 - SAME AS LARGE MILL
(2 REQUIRED INCLUDES ONE NEEDED
WITH ITEM 20)

ITEM 47 - NOT NEEDED

ITEM 48 - SAME AS LARGE MILL EXCEPT
ONLY 4 REQUIRED

ITEM 49 - SAME AS LARGE MILL

ITEM 50 - SAME AS LARGE MILL EXCEPT
ONLY 1 REQUIRED

ITEM 51 - SAME AS LARGE MILL

ITEM 52 - SAME AS LARGE MILL

ITEM 53 - NOT NEEDED

ITEM 54 - NOT NEEDED

ITEM 25 - NOT NEEDED

ITEM 26 - SAME AS LARGE MILL

ITEM 27 - SAME AS LARGE MILL

PURCHASE PARTS

ITEM 28 - SAME AS LARGE MILL

ITEM 29 - SAME AS LARGE MILL

ITEM 30 - SAME AS LARGE MILL EXCEPT
USE ML1912 INSTEAD OF ML1851

ITEM 31 - SAME AS LARGE MILL EXCEPT
ORDER 24 GCF-200-E, 200 PITCHES,
26.180" CIRC.

ITEM 32 - SAME AS LARGE MILL EXCEPT
ORDER 16 GCF - 4 FT., 225 PITCHES

ITEM 33 - NOT NEEDED

ITEM 34 - NOT NEEDED

ITEM 35 - NOT NEEDED

ITEM 36 - NOT NEEDED

ITEM 37 - SAME AS LARGE MILL

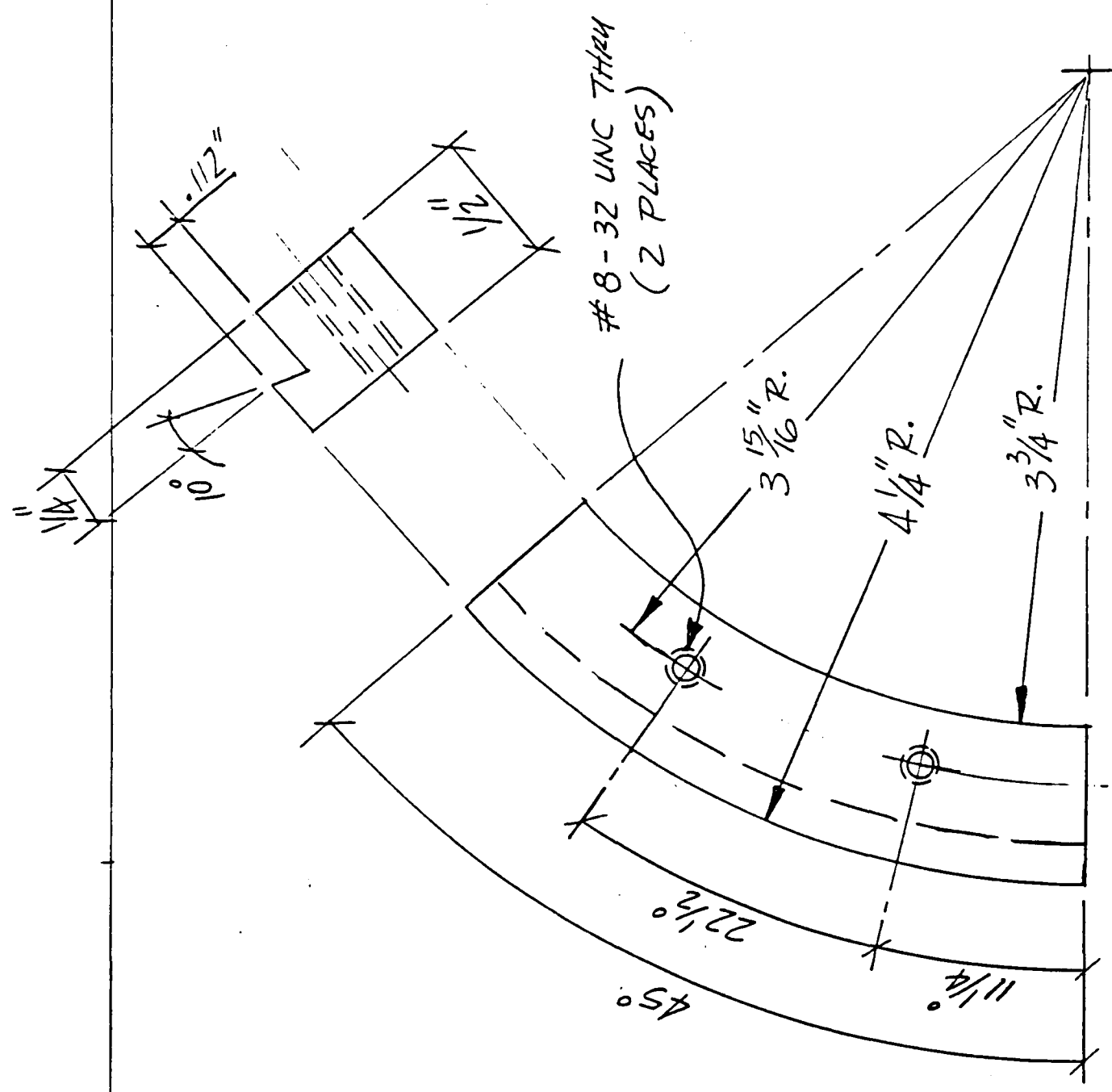
ITEM 38 - SAME AS LARGE MILL

ITEM 39 - SAME AS LARGE MILL
(4 REQUIRED WHICH INCLUDES THE TWO
NEEDED FOR ITEM 20)

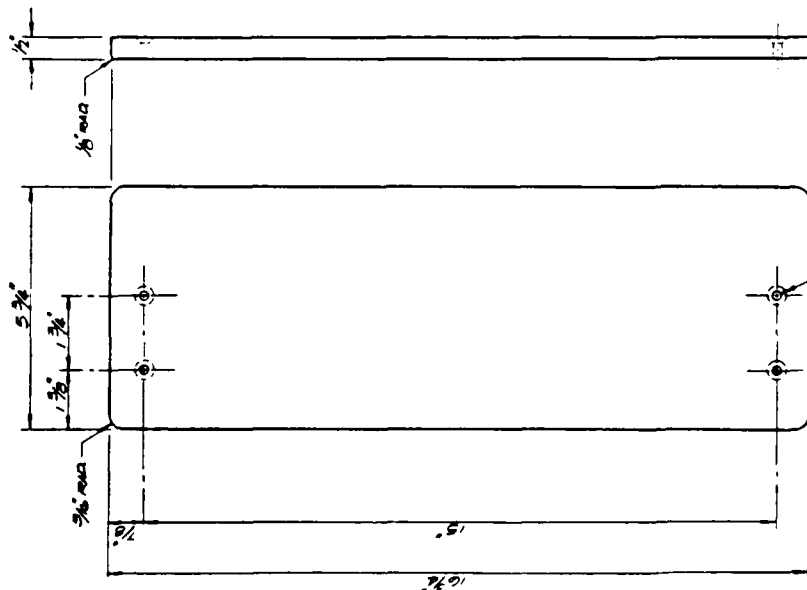
ITEM 40 - SAME AS LARGE MILL

ITEM 41 - WORM GEAR, 24 PITCH, 4" P.D.
STOCK DRIVE PRODUCTS # 1866-N24096
(REMOVE HUB AND DRILL TO MATCH)

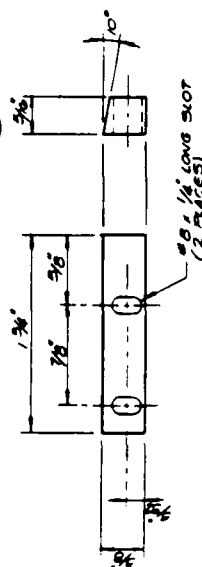
ITEM 24 TURRET CHAIN CLAMP (ALUMINUM) 8 REQUIRED



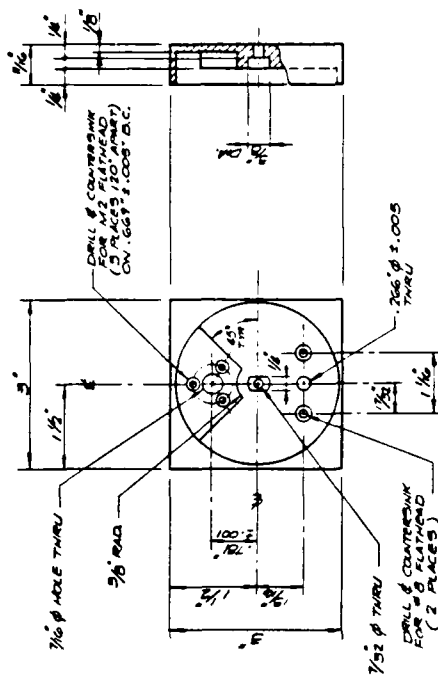
#8-32 UNC THRU
(2 PLACES)



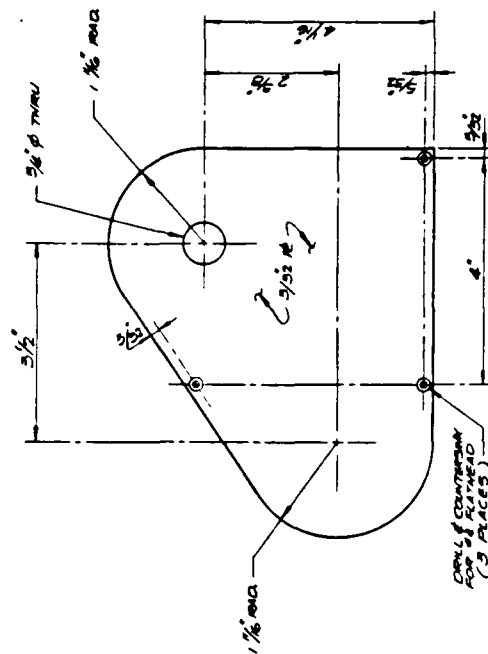
DETAIL ITEM 22 (DME)
SCALE: 3/4"



DETAIL ITEM 23 (ALUMINUM)
SCALE: DOUBLE



DETAIL ITEM 24 (ALUMINUM)
FULL SCALE



DETAIL ITEM 25 (ALUMINUM)
FULL SCALE

NOTES
1. WORK THIS DME WITH SKL
2. MATERIALS: ALUMINUM AND
3. DRAWING NO. 0038
4. ALL FRACTIONAL DIMENSIONS
TO BE (1/16) 1/16"

APPENDIX E

Detailed Designs for Miniature Industrial Robot Manipulator

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				4-26-72	0010	2 of
Approved by				NO.	REVISIONS	DATE
Project Engineer						
Project Name						
CAG-CAM						
Drawing Title						
ROBOT						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
12	0025	Slide Motor Support	1	C. Steel		
13	0025	Slide Shaft Support	1	C. Steel		
14	0025	Fitch Motor Support	1	C. Steel		
15	0025	Upper Slide Rail	1	Aluminum		
16	0025	Fitch Preload Plate	1	C. Steel		
17	0025	Slide Shaft Preload Plate	2	Aluminum		
18	0025	Slide Shaft Bearing Cover	2	Aluminum		
19	0025	Slide Thrust Washer Cover	2	Aluminum		
20	0025	Lower Slide Rail	1	Aluminum		
21	0027	Slide Shaft	2	C. Steel		
22	0027	Slide Drive Shaft	1	C. Steel		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

BRIGHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				4-26-72	0010	2 of
Approved by				NO.	REVISIONS	DATE
Project Engineer						
Project Name						
CAG-CAM						
Drawing Title						
ROBOT						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE	
22	0027	Slide Drive Screw	1	See Detail		
23	0027	Fitch Shaft	1	C. Steel		
24	0027	End Eff. Fitch Shaft	2	C. Steel		
25	0027	Pivot Shaft	1	C. Steel		
26	0027	End Eff. Shaft Support	1	C. Steel		
27	0027	End Eff. Bracket	1	C. Steel		
28	0027	End Eff. Support	1	C. Steel		
29	0027	Pivot Shaft Nut	1	C. Steel		
30	0027	Friction Pad	4	Bronze		
31	0027	External Cover	1	Acrylic		
32	0027	Arm Cover	1	Acrylic		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BRIGHAM YOUNG UNIVERSITY

Project Name		Project No.		Drawing No.		Sheet	
BRIGHAM YOUNG UNIVERSITY		4-26-32		10010		4 of 4	
Prepared by		NO.		REVISIONS		DATE	
FORREST BLAIR				DESCRIPTION			
Approved by							
Project Engineer							
Project Name							
CAD-CAM							
Drawing Title							
ROBOT							
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE		
33	0021	Not used					
34	0021	Guide Wheel - Dualves, Size 1, 1/2" Bore, with Adapter Bushing BX1	4	Wg. Std. Bishop-Isenrover			
35	0021	Motor - DC Stepping Slo-3yn	1	Wg. Std. Superior Electric			
36	0021	Motor - DC Stepping Slo-3yn, #001-FC03	1	Wg. Std. Superior Electric			
37	0021	Motor - DC Stepping Slo-3yn, #001-FC06	1	Wg. Std. Superior Electric			
38	0021	Motor - DC Stepping Slo-3yn, #001-FC02	1	Wg. Std. Superior Electric			
39	0021	Gear - Worm, Single Thread, 24 Pitch, 1/2" Bore, 2.000 P.D., #24935-350	2	Wg. Std. Infred Berr			
40	0021	Worm - Single Thread, 24 Pitch, 1/2" Bore, 1.245-45	2	Wg. Std. Infred Berr			

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BRIGHAM YOUNG UNIVERSITY

Project Name		Project No.		Drawing No.		Sheet	
BRIGHAM YOUNG UNIVERSITY		4-26-32		10010		5 of 5	
Prepared by		NO.		REVISIONS		DATE	
FORREST BLAIR				DESCRIPTION			
Approved by							
Project Engineer							
Project Name							
CAD-CAM							
Drawing Title							
ROBOT							
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIAL	DATE		
40	0021	Gear - Bevel, 24 Pitch, #124F-2, 1 each 1" P.D. and 2 each 2" P.D.	14 sets	Wg. Std. Infred Berr			
41	0021	Worm - Type 1", 1/16-1H, 1/2" Bore, 16 Pitch, 1.125 P.D.	1	Wg. Std. Browning			
42	0021	Gear - Sprocket, 32 Pitch, 1/2" Bore, 3/4" P.D., GG13S23-24	2	Wg. Std. Infred Berr			
43	0021	Chain - Gear Drive, 32 Pitch, 1/2" Bore, 7.365" circ.	1	Wg. Std. Infred Berr			
44	0021	Gear - Sprocket, 16 Pitch, 1/2" Bore, 1" Teeth, 1.1250 P.D., 3017036-11	1	Wg. Std. Infred Berr			
45	0021	Chain - Gear Drive, 16 Pitch, 1/2" Bore, 170-170-170 Pitches, 11.405" circ.	1	Wg. Std. Infred Berr			

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT.

BILL OF MATERIALS

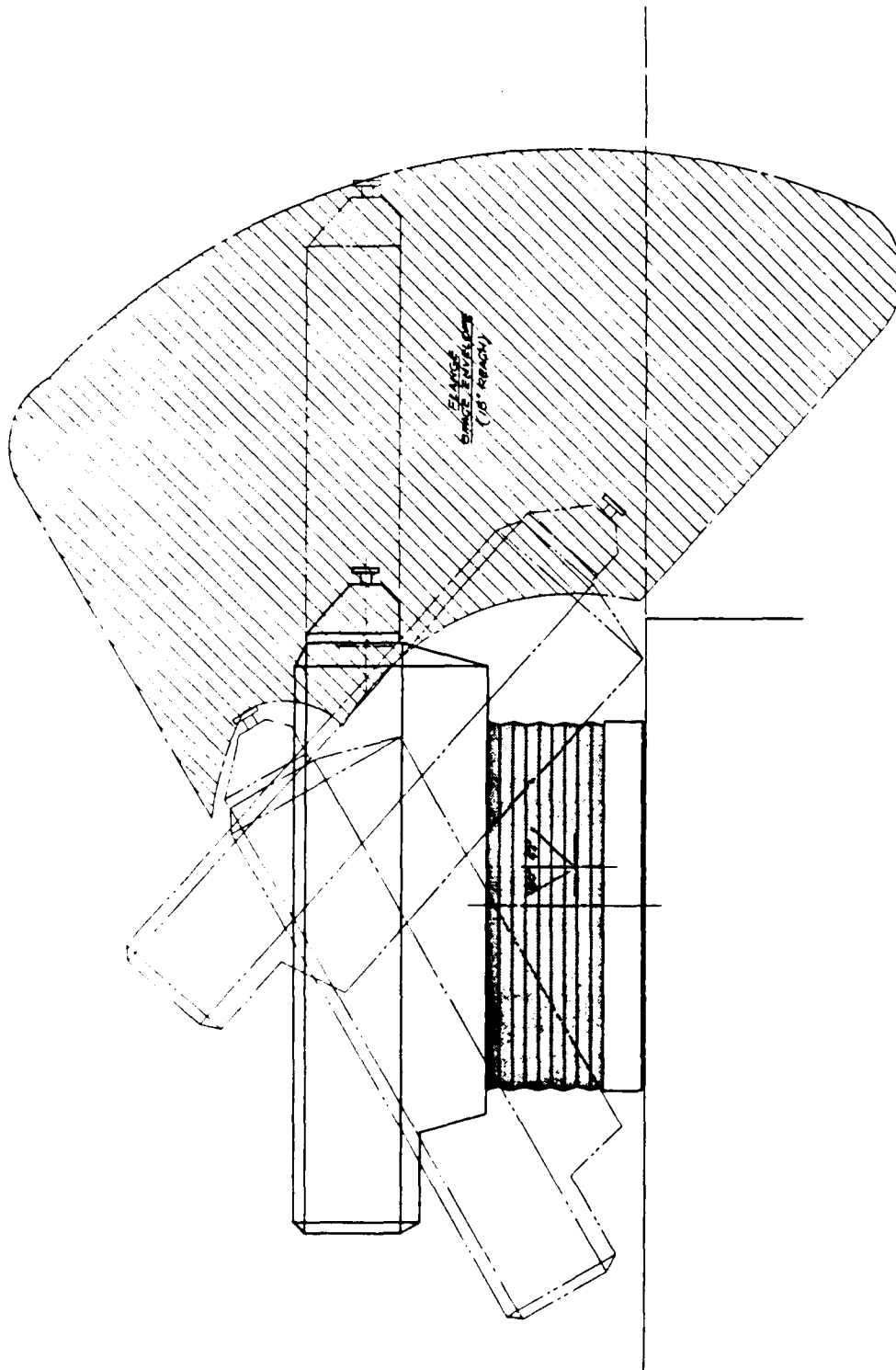
BRIDHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				4-26-32	0010	5 of
Approved by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer						
Project Name CAD-CAM						
Drawing Title ROBOT						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIALS	DATE	
46	0021	Bushing - 3/32" Bore, 1/2" Long, B6-41	7	Infred Berr		
47	0021	Bushing - 3/32" Bore, 3/32" Long, B7-42	1	Infred Berr		
48	0021	Bushing - Plain Cylindrical, 1" nom., 1 3/32" Long, #15066 B1620-7	2	Boston Berr		
49	0021	Bushing - Plain Cylindrical, 3/4" nom., 7/32" Long, #14976 B1216-7	2	Boston Berr		
50	0021	Bushing - Washer, Thrust, 3/4" nom., 62222 B3-1217	2	Boston Berr		
51	0021	Bearing - 1" Bore, B1-44	2	Infred Berr		
52	0021	Bearing - 1" Bore, B1-27	2	Infred Berr		
53	0021	Washer - Thrust, 1" Bore, B5-6-55	1	Infred Berr		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT

BILL OF MATERIALS

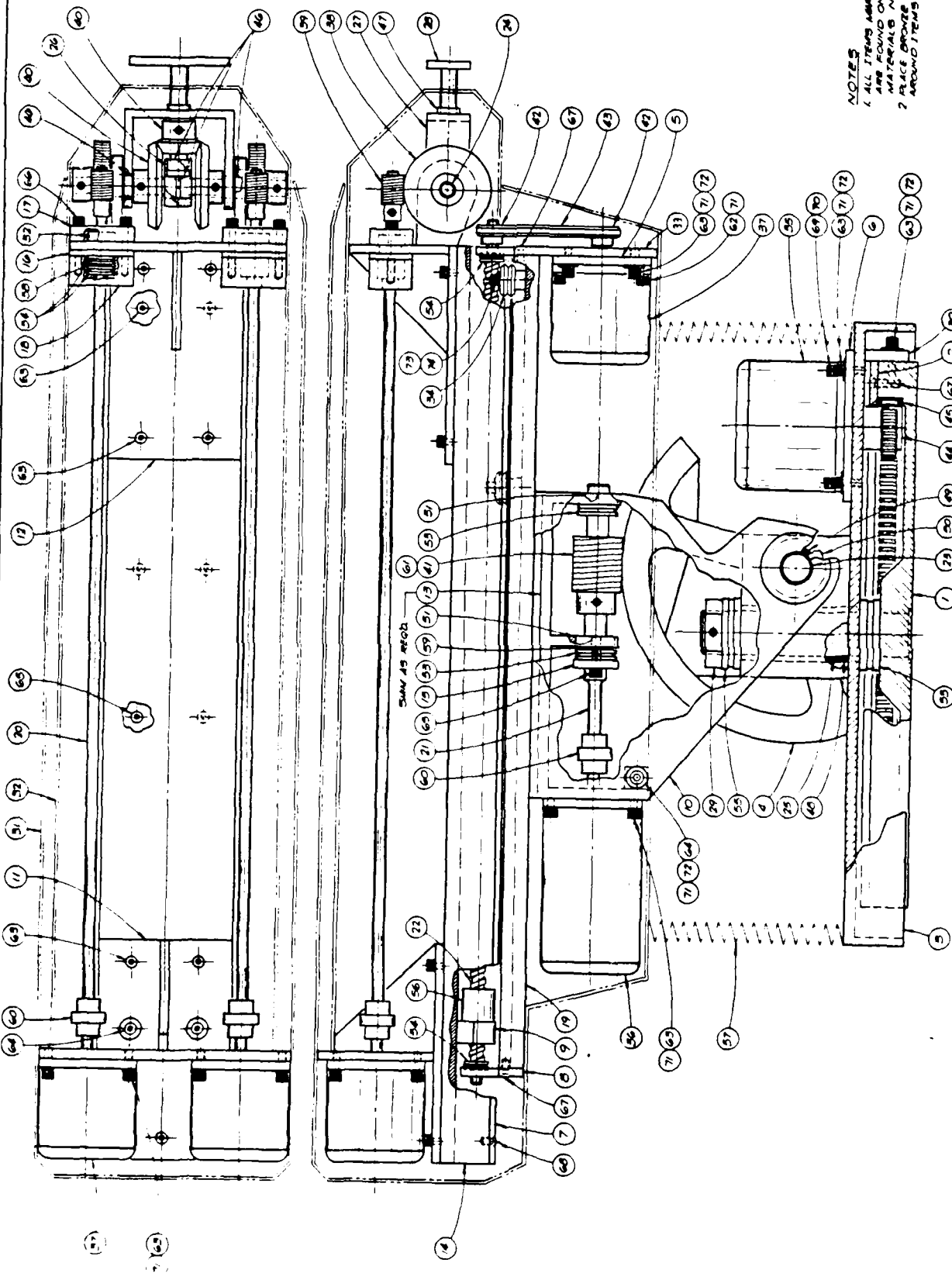
BRIDHAM YOUNG UNIVERSITY				Project No.	Drawing No.	Sheet
Prepared by FORREST BLAIR				4-26-32	0010	7 of
Approved by				NO.	REVISIONS DESCRIPTION	DATE
Project Engineer						
Project Name CAD-CAM						
Drawing Title ROBOT						
ITEM	DWG. NUMBER	PART NAME (Material Description)	UNITS REQUIRED	MATERIALS	DATE	
54	0021	Washer - Thrust, 1" Bore, B5-1-SS	6	Infred Berr		
55	0021	Washer - Thrust, 1" nom., B5-10-SS	2	Infred Berr		
56	0021	Ball Nut - 3/32" nom., RN-0175-0125	1	Infred Berr		
57	0021	Bellevue - 1 1/2" x 4" I.D., 4 1/2" nom. height, must expand to 10"	1	Is Required		
58	0021	Retainer Ring - 1" nom., Q1-25	2	Infred Berr		
59	0021	Retainer Ring - 1" nom., Q1-50	1	Infred Berr		
60	0021	Coupling - 1" Bore, C020-14	1	Infred Berr		
61	0021	Key - 1/2" Square, 1 3/4" Long	1	Key Stock		
62	0021	Key - Hex Socret, 10-20 UN 1 3/4" Long	1	Key Stock		

DRAFTSMAN PREPARE 1 COPY AND RETAIN ORIGINAL SEND PHOTOCOPIES TO VENDOR AND FACTORY PURCHASING AGENT



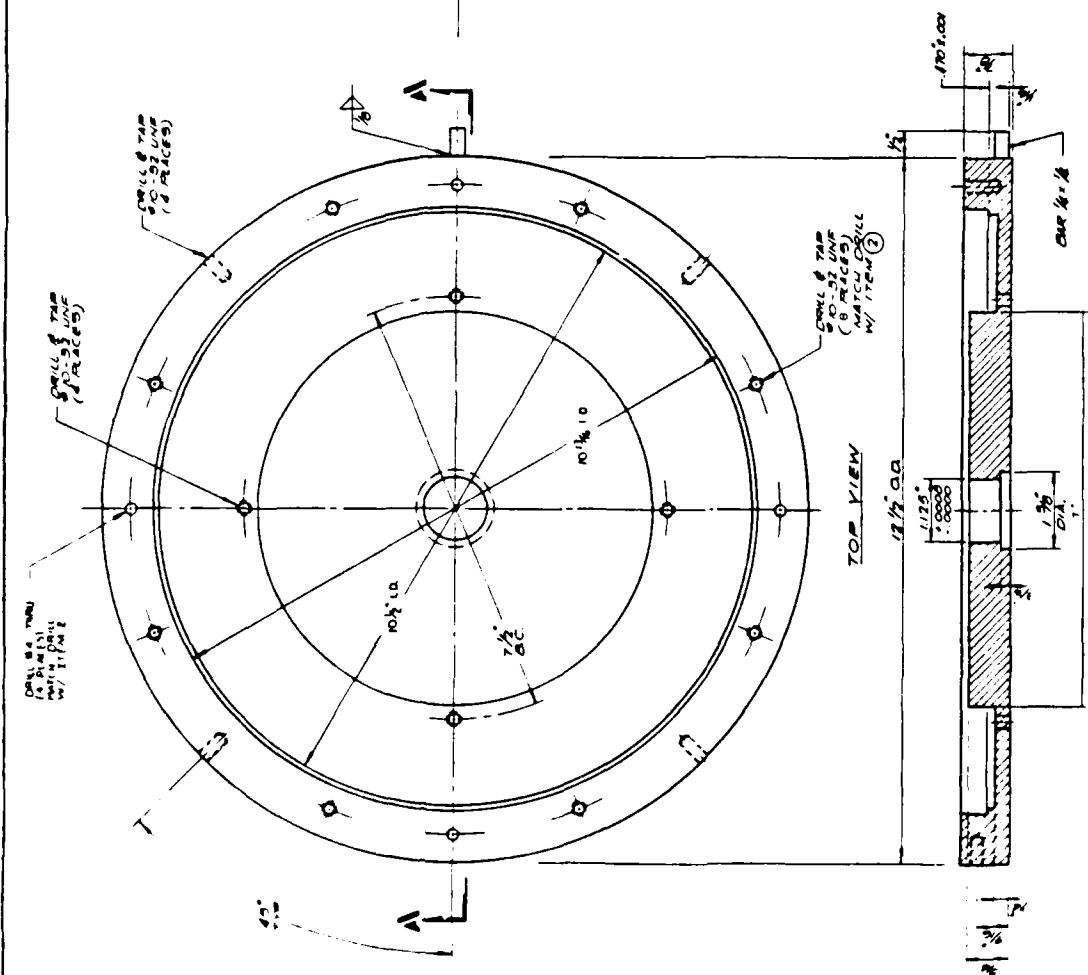
SCALE: 1/2" = 1"

BRIGHAM YOUNG UNIVERSITY UTAH 84602	PROJECT TITLE ROBOT	DATE OF ISSUE PATH GENERATION	DESIGNER FORREST BLAIR	PROJECT NUMBER 0030
--	-------------------------------	---	----------------------------------	-------------------------------



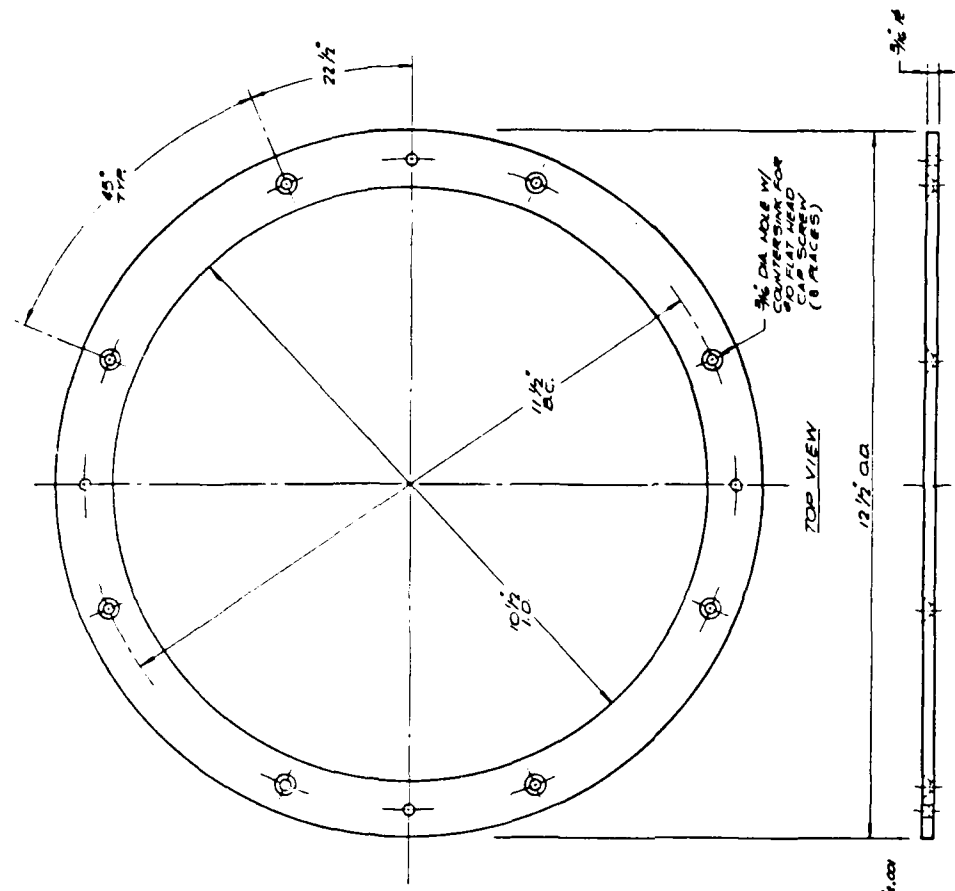
NOTES
 1. ALL ITEMS MARKED TABS
 ARE FOUND ON DILL OF
 MATERIALS AS MOOD.
 2. PLACE DRAWING SHOWN AS MOOD
 AROUND ITEMS 54, 57 & 60.

1 MAR 19 1982 RAC-CAM UNIT		BRIGHAM YOUNG UNIVERSITY UTAH 84602		ASSEMBLY		DESIGNED BY: FOREST BLAIR DRAWN BY: [blank] CHECKED BY: [blank]		0031
-------------------------------	--	--	--	----------	--	---	--	------



SECTION A-A

DETAIL ITEM 1 (STEEL)
FULL SCALE

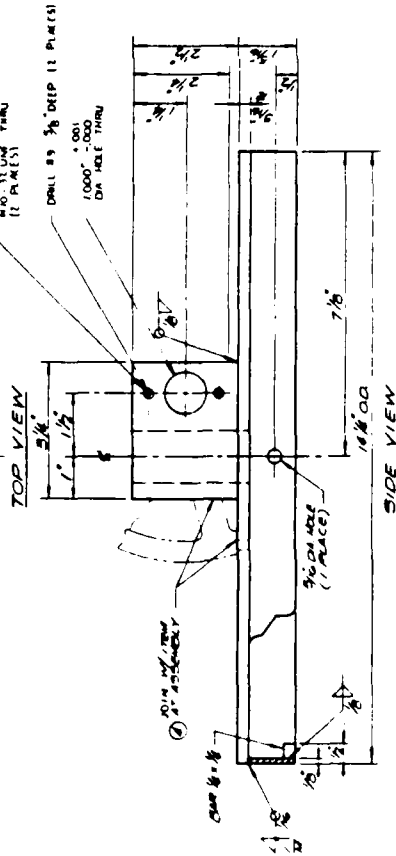
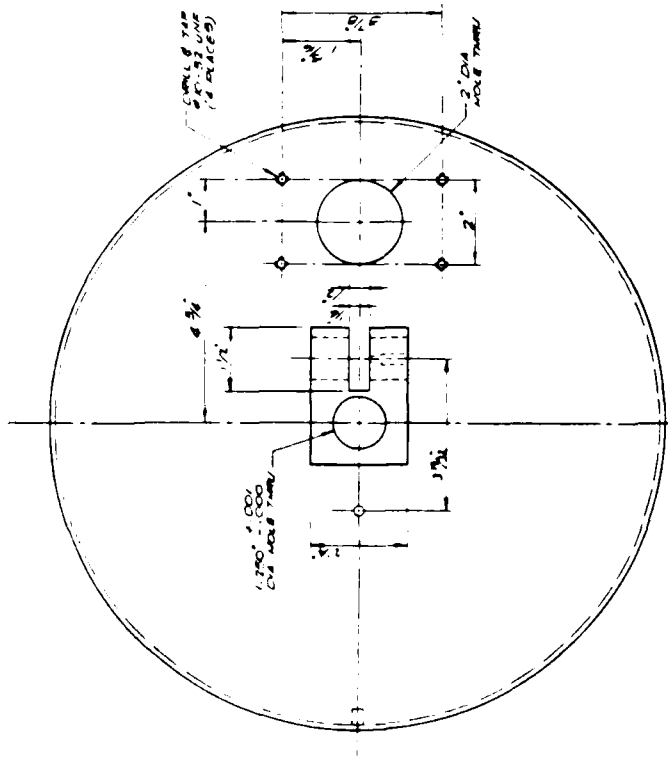


SECTION A-A

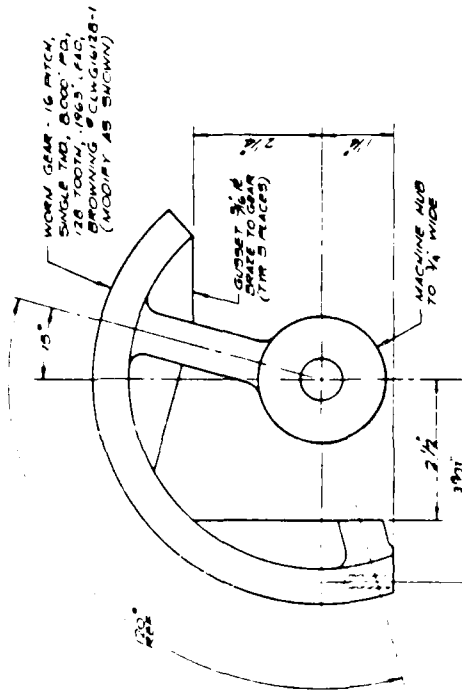
DETAIL ITEM 2 (ALUMINUM)
FULL SCALE

- NOTES
1. WORK THIS DRAWING WITH BILL OF MATERIALS AND DRAWING CO22 AND DRAWING CO23.
 2. ALL FRACTIONAL DIMENSIONS TO BE 1/8" INCREMENTS.
 3. ALL ANGLES TO BE 15°.

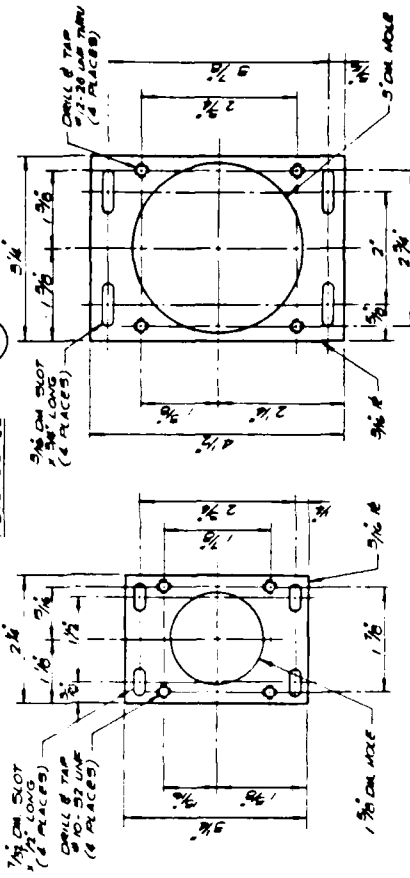
DATE: 10/19/62	BY: BRIGHAM	YOUNG	UNIVERSITY	UTAH STATE	PROJECT: ROBOT	DETAIL ITEM 1 & 2	FOR: FOREST BLAIR	CO22
----------------	-------------	-------	------------	------------	----------------	-------------------	-------------------	------



DETAIL ITEM 3 (C. STEEL)
FULL SCALE



DETAIL ITEM 4
FULL SCALE

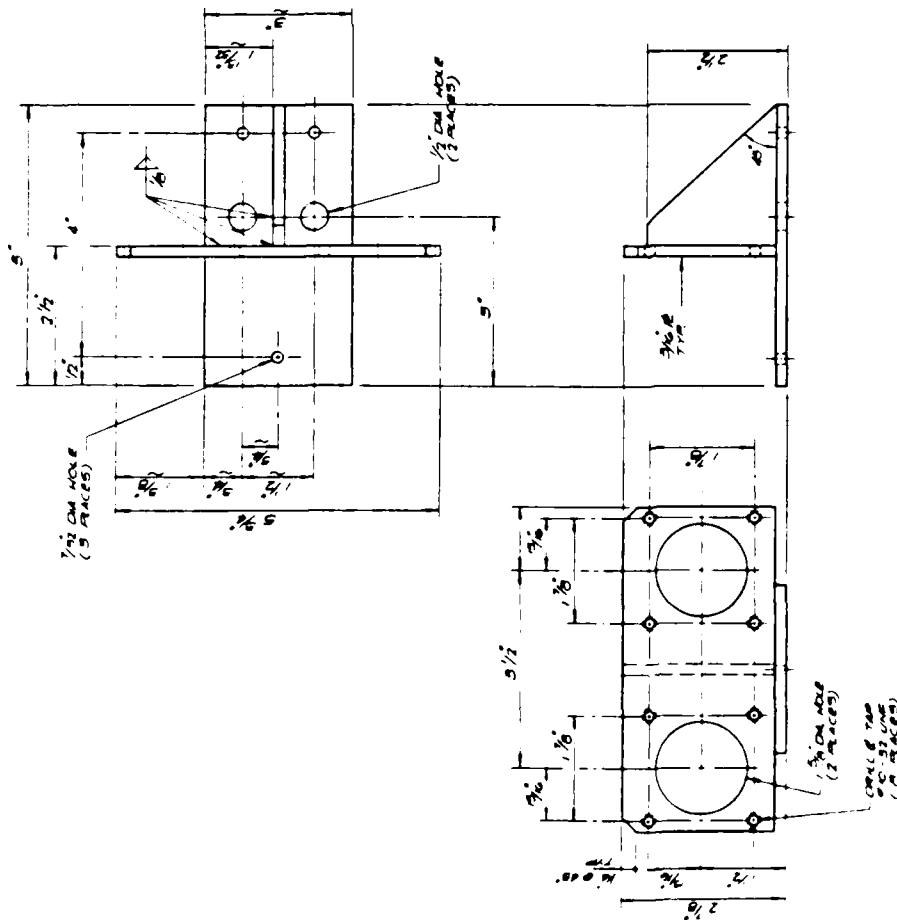


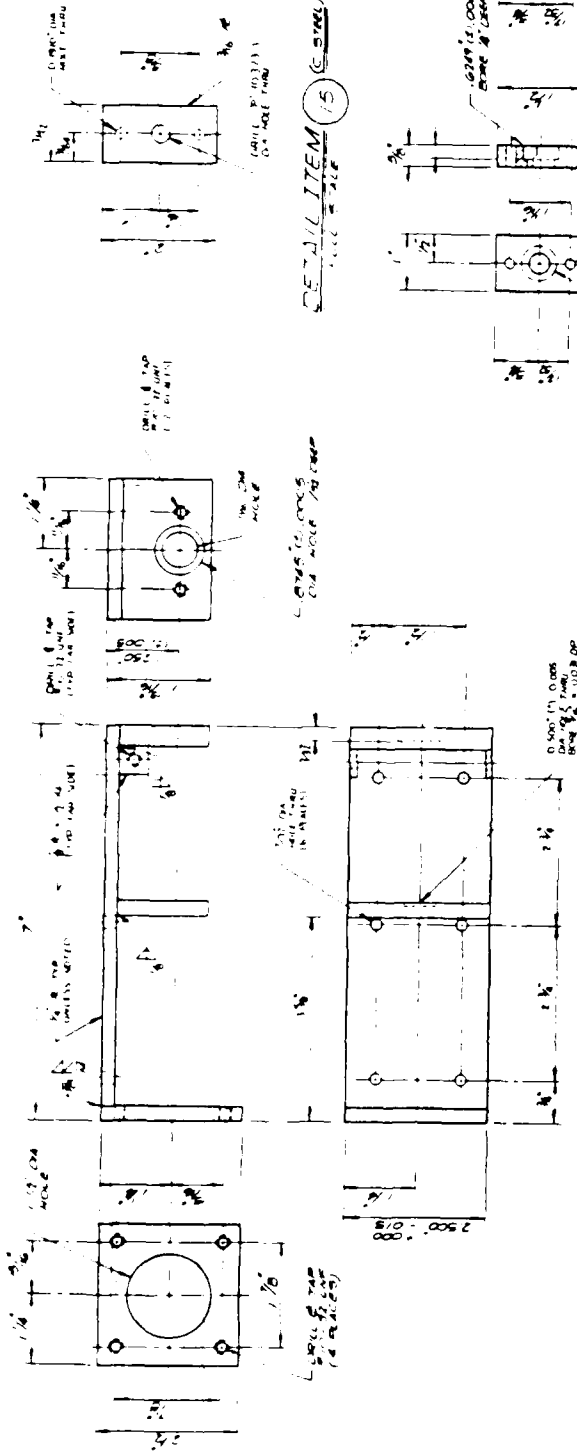
DETAIL ITEM 5 (ALUMINUM)
FULL SCALE

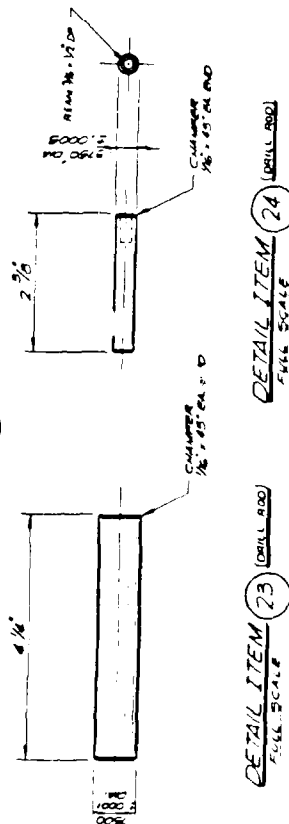
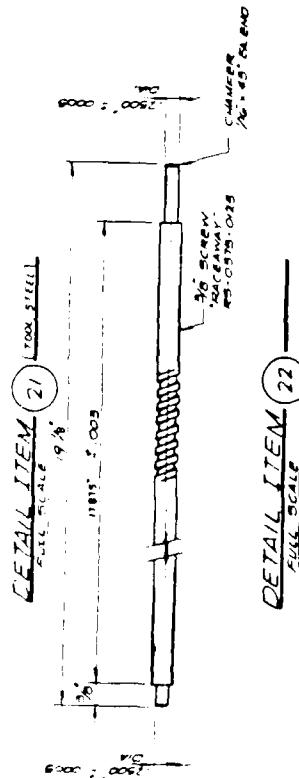
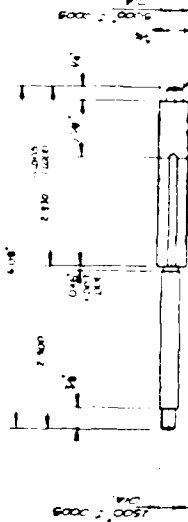
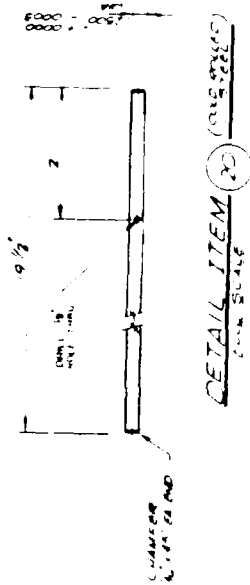
DETAIL ITEM 6 (ALUMINUM)
FULL SCALE

NOTES
1 WORK THIS Dwg WITH BILL
OF MATERIALS M0020 AND
DRAWING C021
2 ALL DIMENSIONS
TO BE (S) UNLESS
OTHERWISE SPECIFIED

DATE: 11/9/82	BRIGHAM YOUNG UNIVERSITY	ROBOT	DETAIL ITEM 3, 4, 5 & 6	FORREST CLAIR	0073
BY: LAD-CAM DEPT	UTAH BRANCH				



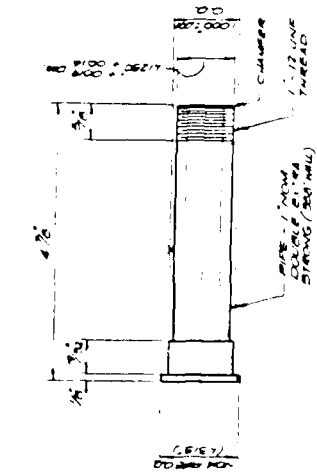




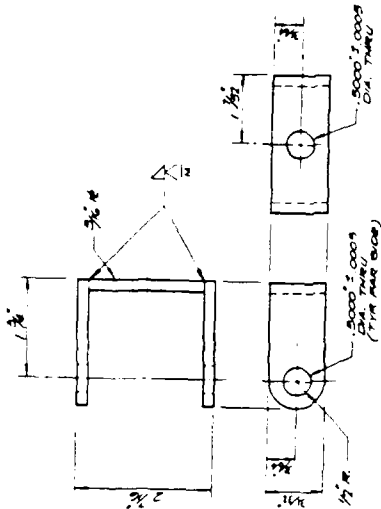
DETAIL ITEM 24 (C. STEEL)
FULL SCALE

DETAIL ITEM 25 (C. STEEL)
FULL SCALE

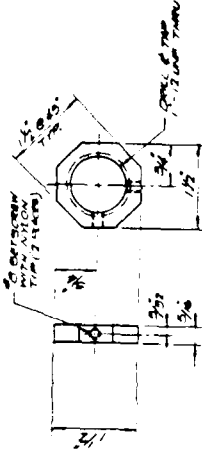
DETAIL ITEM 26 (C. STEEL)
FULL SCALE



DETAIL ITEM 28 (C. STEEL)
FULL SCALE



DETAIL ITEM 30 (C. STEEL)
FULL SCALE



DETAIL ITEM 32 (C. STEEL)
FULL SCALE

NOTES
1. WORK THIS DRAWING WITH BALL OF MATERIALS WOODS AND DRAWING 0021.
2. ALL DIMENSIONAL DIMENSIONS TO BE (1) 1/2 IN.

DATE: 1 1962	BY: C. W. DEPT.	PROJECT: BRIGHAM YOUNG UNIVERSITY	PROJ: ROBOT	DETAIL ITEM: 20 21 23 24 25 26 27 28 29 30	FOR: FOREST CLARK	0027
--------------	-----------------	-----------------------------------	-------------	--	-------------------	------

0486 45 20 30 20
 048C 24
 048D 52 41 54 45
 04C1 20 30 20 24
 04C5 46 41 43 54
 04C9 4F 52 20 30
 04CD 20 24
 04CF 20 48 41 4C
 04D3 46 20 53 54
 04D7 45 50 20 24
 04DB 20 46 55 4C
 04DF 4C 20 53 54
 04E3 45 50 20 24
 04E7 43 4C 4F 43
 04EB 4E 57 49 53
 04EF 45 20 24
 04F2 43 4F 55 4E
 04F6 54 45 52 20
 04FA 43 4C 4F 43
 04FE 4E 57 49 53
 0502 45 20 24
 0505 5C 55 4C 53
 0509 45 53 20 46
 050C 52 4F 40 20
 0511 4E 4F 40 45
 0515 20 30 20 24

VERATE: DB "RATE = \$" ;
 VEFACT: DB "FACTOR = \$" ;
 VEHLST: DB " HALF STEP \$"
 VEFLST: DB " FULL STEP \$"
 VECWM: DB "CLOCKWISE \$"
 VECWMM: DB "COUNTER CLOCKWISE \$"
 PEQUAL: DB "PULSES FROM HOME = \$"

;

;

;

;*****

;* SEND CHARACTER TO THE CY512

;*

;: THIS ROUTINE SEND THE CHARACTER IN THE 'A' REG

;* TO THE ACTIVE CY 512 MOTOR CONTROLLER

;*****

;ENTER THIS ROUTINE WITH THE ASCII CHARACTER

;CURRENTLY LOADED IN THE 'A' REGISTER.

;

0519 CD 06A2
 051C FL 2A 0322
 0520 FD CB 00 56
 0524 CA 0520
 0527 E6 7F
 0529 FD 77 04
 052C 2E 41
 052E FD 75 0C
 0531 FD CB 00 56
 0535 C2 0531
 0538 2E 40
 053A FD 75 0C
 053D CD 06B9
 0540 C9

SENDCY: CALL SAVREG ;SAVE ACTIVE REGS.
 LD IY,(CYCTRL) ;SET THE POINTER TO ACTIVE CY512
 SOWAIT: BIT 2,(IY) ;TEST THE CY512, IS IT READY???
 JP 2,SOWAIT ;I'M BUSY, PLEASE WAIT.
 AND 07FH ;MAKE SURE CH# IS ASCII.
 LD (IY+4),A ;PLACE CH# IN CY DATA REGISTER
 LD L,41H ;SET I/O REQUEST BIT
 LD (IY),L
 SCHDSK: BIT 2,(IY) ;HANDSHAKE WITH CY512 READY FLAG.
 JP NZ,SCHDSK ;WAIT UNTIL CH# IS TAKEN
 LD L,40H ;RESET I/O REQUEST BIT.
 LD (IY),L ;...
 CALL RESTOR ;RESTORE PROCESSOR REGS.
 RET ;EXIT THE SENDCY ROUTINE...

;

;

;*****

;* GET A CHARACTER FROM THE CY512

```

0449 CD 0581      VEDSET: CALL PRINT      ;... 'CW' ON THE DISPLAY.
044C DD CB 05 7E      BIT      7,(IX*5)      ;IS THE 'HALF STEP' BIT SET ?
0450 CA 0466      JP      2,VEFS      ;NO, GO PRINT 'FULL STEP' ON DISPLAY.
0453 11 04CF      LD      DE,VEHLST      ;YES, PRINTOUT 'HALF STEP' ON SCREEN.
0456 CD 0581      VEEXIT: CALL PRINT      ;.....
0459 CD 05A3      CALL      CRLF      ;.....
045C CD 06B9      CALL      RESTOR      ;RESTORE IX REGISTER AND....
045F C9          RET      ;...EXIT THIS ROUTINE.

;
0460 11 04E7      VECW: LD      DE,VECWM      ;PRINTOUT 'CW' ON ....
0463 C3 0449      JP      VEDSET      ;..THE SCREEN.

;
0466 11 04DE      VEFS: LD      DE,VEFLST      ;PRINT 'FULL STEP' ON.....
0469 C3 0456      JP      VEEXIT      ;..THE SCREEN.

;
046C 06 00      VECTPT: LD      B,0      ;WE WILL OUTPUT 4 CHR'S....
046E 7E          VOLOOP: LD      A,(HL)      ;TRANSFER CHR FROM THE CHR BUFFER...
046F CD 0519      CALL      SENDCY      ;SEND TO THE CYS12, 'B' WILL COUNT THEM.
0472 23          INC      HL      ;INC CHR POINTER.
0473 04          INC      B      ;INCREMENT THE COUNT...
0474 CB 50      BIT      2,B      ;..AND SEE IF 4 CHR'S SENT YET ?
0476 CA 046E      JP      2,VOLOOP      ;NOPE, DO ANOTHER.
0479 C9          RET      ;.....

;
047A CD 06A2      VFETCH: CALL      SAVREG      ;SAVE THE 'X' REG.
047D 3E 00      LD      A,0      ;CLEAR THE BYTE COUNTER.
047F 2A 0497      LD      HL,(VCOUNT)      ;LOAD THE BYTE COUNT INTO 'HL' REG.
0482 DD 21 0498      LD      IX,NUMBER      ;SET PNTR TO THE DATA BUFFER.
0486 CD 0541      VFLOOP: CALL      GETCY      ;GO GET A DATA BYTE..
0489 DD 73 00      LD      (IX),E      ;..AND STORE IT IN THE DATA BUFFER.
048C DD 23          INC      IX      ; INC THE BUFFER POINTER AND...
048E 3C          INC      A      ;..THE BYTE TRANSFER COUNT...
048F BC          CP      L      ;HAVE WE FETCHED ALL THE CHR'S YET ?
0490 C2 0486      JP      NZ,VFLOOP      ;NO, GO GET ANOTHER CHR.
0493 CD 06B9      CALL      RESTOR      ;YES, RESTORE 'X' REG AND...
0496 C9          RET      ;..EXIT

;
0497 00          VCOUNT:      0      ;

;
; THIS IS THE CYS12 DATA BYTE BUFFER.
;
0498 00          NUMBER:      0      ;'N' PULSEC COUNT.
0499 00          SLOPE:      0      ;.....
049A 00          RATE:      0      ;SLOPE COUNT.
049B 00          FACTOR:      0      ;RATE COUNT.
049C 00          STATUS:      0      ;FACTOR COUNT.
049D 00          ;STATUS COUNT.

;
049E 56 20 30 00      VZERO: DB      "V 0",13      ;THE "V 0" CHR STRING.
04A2 56 20 32 00      VTWO: DB      "V 2",13      ;THE "V 2" CHR STRING.
04A6 56 20 33 00      VTHREE: DB      "V 3",13      ;THE "V 3" CHR STRING.

;
04AA 4E 55 4D 42      VENUMB: DB      "NUMBER = $"      ;SCREEN OUTPUT FOLLOWS....
04AE 45 52 20 3D
04B2 20 24
04B4 53 4C 4F 50      VESLOP: DB      "SLOPE = $"      ;

```

```

;
; THIS ROUTINE ISSUES 'V 0', 'V 2', AND 'V 3'
; TO THE ACTIVE CY512 AND PRINTS OUT ITS STATUS.
; PULSES FROM HOME, NUMBER, RATE, SLOPE, FACTOR,
; DIRECTION, AND STEP MODE.
;

```

```

03C3 CD 06A2      VERIFY: CALL SAVREG      ;
03C6 CD 0580      CALL HOME      ;CLEAR THE DISPLAY.
03C9 21 049E      LD HL,VZERO     ;SEND 'V 0' TO THE CY 512...
03CC CD 046C      CALL VEDTPT     ;.....
03CF 11 0505      LD DE,PEQUAL    ;PNT 'PULSES FROM HOME =' AND..
03D2 CD 0581      CALL PRINT      ;.....
03D5 3E 00      LD A,0           ;CLEAR THE COUNT USED TO GET...
03D7 CD 0541      VELOOP: CALL GETCY ;FIVE BYTES OF DATA..
03DA CD 058C      CALL WICHR      ;..FROM THE CY 512..
03DD 3C          INC A           ;..THAT MAKE UP THE PULSE COUNT.
03DE FE 05      CP 5            ;ARE WE DONE YET ?
03E0 C2 03D7      JP NZ,VELOOP    ;NO, GO GET ANOTHER BYTE.
03E3 CD 05A3      CALL CRLF       ;YES,....
03E6 21 04A2      LD HL,VTWO     ;NOW SEND 'V 2' TO THE CY 512.
03E9 CD 046C      CALL VEDTPT     ;.....
03EC 3E 06      LD A,6          ;PREPARE TO GET 6 BYTES OF DATA...
03EE 32 0497      LD (VCOUNT),A   ;..FROM THE CY 512..
03F1 CD 047A      CALL VFETCH     ;GO GET THEM.
03F4 21 04A6      LD HL,VTTHREE   ;NOW SEND 'V 3' TO THE CY 512.
03F7 CD 046C      CALL VEDTPT     ;.....
03FA 3E 05      LD A,5          ;RETRIEVE 5 BYTES OF DATA FROM CY512..
03FC 32 0497      LD (VCOUNT),A   ;..OVERWRITING 1'ST 5 BYTES OF ....
03FF CD 047A      CALL VFETCH     ;..PREVIOUS FETCH.

```

```

;
; WE NOW HAVE THE DATA AND WILL PRINT OUT A STATUS DISPLAY.
;

```

```

0402 DD 21 049E      VEDSP1: LD IX,NUMBER      ;SET UP POINTER TO STATUS BUFFER.
0406 DD 66 00      LD H,(IX)      ;GET THE 'PULSE COUNT' ..
0409 DD 6E 01      LD L,(IX+1)    ;..READY FOR PRINTOUT.
040C 11 04A2      LD DE,VENUMB    ;PRINT 'NUMBER = ' ON ...
040F CD 0581      CALL PRINT      ;...THE DISPLAY
0412 CD 058C      CALL BDCONV     ;..AND TYPE OUT THE COUNT.
0415 26 00      LD H,0           ;GET THE 'SLOPE CNT' READY..
0417 DD 6E 02      LD L,(IX+2)    ;..FOR PRINTOUT.
041A 11 04B4      LD DE,VESLOP    ;PRINT 'SLOPE = ' ON..
041D CD 0581      CALL PRINT      ;...THE DISPLAY.
0420 CD 058C      CALL BDCONV     ;CONVERT TO ASCII & PRINT SLOPE.
0423 26 00      LD H,0           ;GET 'RATE' COUNT READY...
0425 DD 6E 03      LD L,(IX+3)    ;...FOR PRINTOUT.
0428 11 04B0      LD DE,VERATE     ;PRINT 'RATE = ' ON ...
042B CD 0581      CALL PRINT      ;..THE DISPLAY....
042E CD 058C      CALL BDCONV     ;..AND CONVERT TO ASCII & PNT RATE.
0431 26 00      LD H,0           ;GET 'FACTOR' READY...
0433 DD 6E 04      LD L,(IX+4)    ;..FOR PRINTOUT.
0436 11 04C5      LD DE,VEFACT    ;PRINT 'FACTOR = ' ON...
0439 CD 0581      CALL PRINT      ;...THE DISPLAY.
043C CD 058C      CALL BDCONV     ;..CONV TO ASCII & PRINT OUT FACTOR.
043F DD 0B 05 66   BIT 4,(IX+5)   ;IS THE 'CW' BIT SET IN 'STATUS' BYTE ?
0443 CA 0460      JP Z,VECW      ;NO, GO PRINTOUT 'CW' ON DISPLAY.
0446 11 04F2      LD DE,VECCW     ;YES, PRINT .....

```

```

0335 01 0025      DTASTR: LD      BC,37      ;PREPARE TO COMPARE 37 CHARACTERS
0338 21 038B      LD      HL,CHRLST    ;IN THE CHARACTER LIST WITH
033E ED B1        CPIR                ;CHARACTER IN THE 'A' REGISTER..
033D C2 036D      JP      NZ,INVALID    ;IS THE CHARACTER VALID??
0340 FE 56        CP      "V"          ;IS CHR A 'VERIFY' ?
0342 CA 1E7-      JP      Z,DTVIFY      ;YES, GO DO A 'VERIFY'.
0345 FE 51        CP      "Q"          ;IS CHR A 'QUIT' ?
0347 CA 0384      JP      Z,SQUIT       ;YES, GO SET 'DONEFLG' TO STOP 'CR'.
034A FE 0D        CP      13           ;IS CHR A 'CARRIAGE RETURN' ?
034C C2 0353      JP      NZ,VLDCHR     ;NO, PROCESS CHARACTER.
034F FD CB 00 CE  SET      1,(1Y)       ;YES, SET 'CR' FLAG TO STOP LATER XTR.
0353 FE 2C      VLDCHR: CP      ",,"    ;YES: NOW SEE IF ITS A ",,.."
0355 C2 035E      JP      NZ,SEND       ;NO, IT'S GOOD AS IS . USE IT.
0356 FD CB 00 DE  SET      3,(1Y)       ;SET UP A CY512 'RESET'.
035C 3E 0D        LD      A,13         ;CHANGE THE ',,' TO A 'CARRIAGE RETURN'..
035E CD 0519      SEND:  CALL    SENDCY   ;SEND THE CHARACTER TO THE ACTIVE CY512.
0361 FD CB 00 5E  BIT      3,(1Y)       ;SHALL WE 'RESET' ?
0365 C2 036C      JP      NZ,DTAEXT     ;.....
0368 FD CB 00 9E  RES      3,(1Y)       ;CLEAR RESET FLAG.
036C C9          DTAEXT:  RET          ;EXIT THE DATA STRING SUBROUTINE..
036D FD CB 00 CE  INVALID: SET 1,(1Y)    ;SET THE 'DONE FLAG'.
0371 11 038C      LD      DE,NVLDMG    ;TYPE ERROR MESSAGE 'INVALID COMMAND'.
0374 CD 0581      CALL    PRINT        ;...
0377 C3 036C      JP      DTAEXT        ;...

;
037A FD CB 00 CE  DTVIFY: SET 1,(1Y)     ;SET DONEFLG AND...
037E CD 03C3      CALL    VERIFY        ;...GO RUN A CY512 'VERIFY' ON ACTV UNIT.
0381 C3 036C      JP      DTAEXT        ;...AND EXIT.

;
0384 FD CB 00 CE  SQUIT:  SET 1,(1Y)     ;SET DONEFLG AND...
0388 C3 035E      JP      SEND          ;...SEND 'Q' COMMAND.

;
038B 2C 41 42 43  CHRLST: DB      ",ABCDEF6HIJLMNOPQRSTUVWXYZ+-0123456789 ",13;VLID CHAR LIST
038F 44 45 46 47
0393 48 49 4A 4C
0397 4E 4F 50 51
039B 52 53 54 55
039F 56 57 58 5B
03A3 2D 30 31 32
03A7 33 34 35 36
03AB 37 38 39 20
03AF 0D
03B0 49 4E 56 41  NVLDMG: DB      "INVALID COMMAND",13,10,7,"$"
03B4 4C 49 44 20
03B8 43 4F 4D 4D
03BC 41 4E 44 0D
03C0 0A 07 24

```

```

;*****
;*          VERIFY
;*      CY 512 INPUT DATA
;*****

```

0208 4C 45 44 21
 020C 43 48 41 52
 02E0 41 43 54 45
 02E4 52 21 00 04
 02E8 07 24

02EA 32 029C
 02ED 06 41
 02EF 32 028C
 02F2 CD 03C
 02F5 C3 02C4

02F8 32 028C
 02FB 06 31
 02FD 32 028C
 0300 CD 03C
 0303 C3 02C4

```

;
;COCYUD: LD      (ACYNUM),A      ;UPDATE THE 'CYNUM' AND
;          SUB     65            ;
;          LD      (CYNUM),A      ;
;          CALL    CYUPDT         ;60 UPDATE THE ACTIVE CY512 ADDRESS.
;          JP      CDEXIT        ;..

```

```

;
;COBDUC: LD      (ABOARD),A      ;UPDATE THE ACTIVE 'BOARD' ..
;          SUB     49            ;
;          LD      (BOARD),A      ;...
;          CALL    CYUPDT         ;60 UPDATE ACTIVE CY512 ADDRESS.
;          JP      CDEXIT        ;..

```

;UPDATES THE ACTIVE CY512 STATUS, CONTROL, AND DATA REGISTER ADDRESSES.

0306 CD 06A2
 0309 ED 5E 028C
 030D 2A 028C
 0310 CE 05
 0312 CB 05
 0314 CE 05
 0316 19
 0317 11 E0FC
 031A 19
 031B 22 031C
 031E CD 06A2
 0321 C9
 0322 00
 0323 00

```

;CYPUDT: CALL    SAVREG          ;
;          LD      DE,(CYNUM)    ;COMBINE 'CYNUM' AND
;          LD      HL,(BOARD)    ;..'BOARD' TO GENERATE THE
;          RLC      L            ;..ACTIVE CY512 STAT, CTRL,
;          RLC      L            ;..AND DATA REGISTER..
;          RLC      L            ;..ADDRESSES.
;          ADD      HL,DE         ;...
;          LD      DE,STAT        ;...
;          ADD      HL,DE         ;...
;          LD      (CYCTRL),HL    ;SAVE THE ADDRESS OF CY512 CTRL REG.
;          CALL    RESTOR        ;
;          RET                   ;EXIT THE ACTIVE CY512 UPDATE S/R.
;CYCTRL: 0                      ;STORE THE ACTIVE CY512 CTRL REG HERE.
;          0                    ;CYDATA ADDRESS IS CYCTRL ADDRESS+4.

```

0324 CD 03C3
 0327 C3 02C4

```

;COVFY: CALL    VERIFY          ;60 EXECUTE THE 'VERIFY' COMMAND.
;          JP      CDEXIT        ;RETURN TO EXIT CHECK.
;DEFINITION FOLLOWS

```

032A 51 31 32 41
 032E 42 43
 0330 41 42 43
 0333 31 32

```

;CMDLST: DB      "012ABC"       ;VALID COMMAND CHARACTERS.
;
;COLST1: DB      "ABC"          ;VALID CY512 CHAR'S.
;COLST2: DB      "12"          ;VALID BOARD NUMBERS.

```

```

;
;
;
;DATA STRING PROCESSING ROUTINE
;THE CHARACTER IN THE 'A' REGISTER IS COMPARED AGAINST
;THE VALID CHARACTER LIST, AND IF VALID -
;SENDS IT TO THE ACTIVE CY512. ' ','S ARE CHANGED TO CARRIAGE RETURN'S.

```

```

0004      GETSTR EQU 10      ;READ IN A BUFFER STRING
^00D      CR EQU 13         ;EXECUTES A CARRIAGE RETURN ON SCREEN
.DA       LF EQU 10         ;EXECUTES A LINE FEED ON SCREEN

;
;
;::::::::::::::::::::::::::
;: CY I/O LOCATIONS      :
;: & PROGRAM VARIABLES  :
;:
;:
;
0007      SLOT EQU 7H       ;CYDRIVER APPLE 'SLOT' LOCATION.
F022      TOPLINE EQU 0F022H ;TOP LINE OF SCREEN
                        ← 0E0B0H+SLOT*10H
                        STAT EQU
;
;
;::::::::::::::::::::::::::
;* INITIALIZE SCREEN AND VARIABLES
;::::::::::::::::::::::::::

0200      BKGND: LD HL,17FFH      ;SET UP USER STACK POINTER.
0203      LD (USRSTK),HL ;.....
0206      LD HL,1000H      ;SET UP SYSTEM STACK PNTR.
0209      LD SP,HL ;....
020A      CALL HOME        ;CLEAR THE DISPLAY SCREEN.
020D      LD DE,HEADER     ;ROUTINE TO WRITE HEADER
0210      CALL PRINT       ;TO THE SCREEN.
0213      LD IY,FLGREG     ;USE IY TO POINT TO FLAG REGISTER.
0217      LD A,4
0219      LD (TOPLINE),A   ;SET TOP OF SCREEN 4 ROWS DOWN
021C      RESTRT: CALL HOME ;CLEAR THE SCREEN
021F      CALL CYUPDT      ;SET UP CYNUM ADDRESS.
0222      CALL RESET       ;'RESET' THE CYS12'S.
0225      LD A,62
0227      LD (BUFF),A      ;SET STRING BUFFER TO MAX LENGTH
022A      LOOP: LD A,0      ;CLEAR LOOP FLAGS
022C      LD (FLGREG),A    ;...
022F      LD BC,62         ;LOAD THE INPUT BUFFER WITH...
0232      LD HL,CRRTN      ;..CARRIAGE RETURN SYMBOLS
0235      LD DE,BUFF+2     ;...
0236      LDIR             ;UNTIL BUFFER FULL.
023A      LD IX,BUFF+2     ;SET INDEX TO START OF BUFFER
023E      LD DE,PROMPT-2   ;SET UP PRINT 'PROMPT'
0241      CALL PRINT       ;AND PRINT PROMPT LINE.
0244      CALL INPSTR      ;GO GET A CHARACTER STRING.
0247      CALL CRLF        ;ENHANCE SCREEN OUTPUT.
024A      MORCHR: LD A,(IX) ;LOAD 'A' REG WITH CHAR.
024D      INC IX           ;ADVANCE BUFFER PNTR TO NEXT CHAR.
024F      CHAR: CP "/"      ;DOES A 'COMMAND STRING FOLLOW ?
0251      JP Z,SETSW       ;YES, SET COMMAND SWITCH.
0254      BIT 2,(IY)       ;IS COMMAND SWITCH SET ?
0258      JP Z,DATAS6     ;NO,.....
025B      CALL CMDSTR      ;YES, PROCESS COMMAND STRING.
025E      EXIT: BIT 1,(IY) ;HAVE WE FINISHED PROCESSING THE STRING.
0262      JP Z,MORCHR      ;NO,CONTINUE PROCESSING CHAR'S IN STR.
0265      BIT 0,(IY)       ;IS IT TIME TO QUIT??
0269      JP NZ,RTNCPM     ;IF YES, CLEAR SCREEN AND RETURN TO CP/M

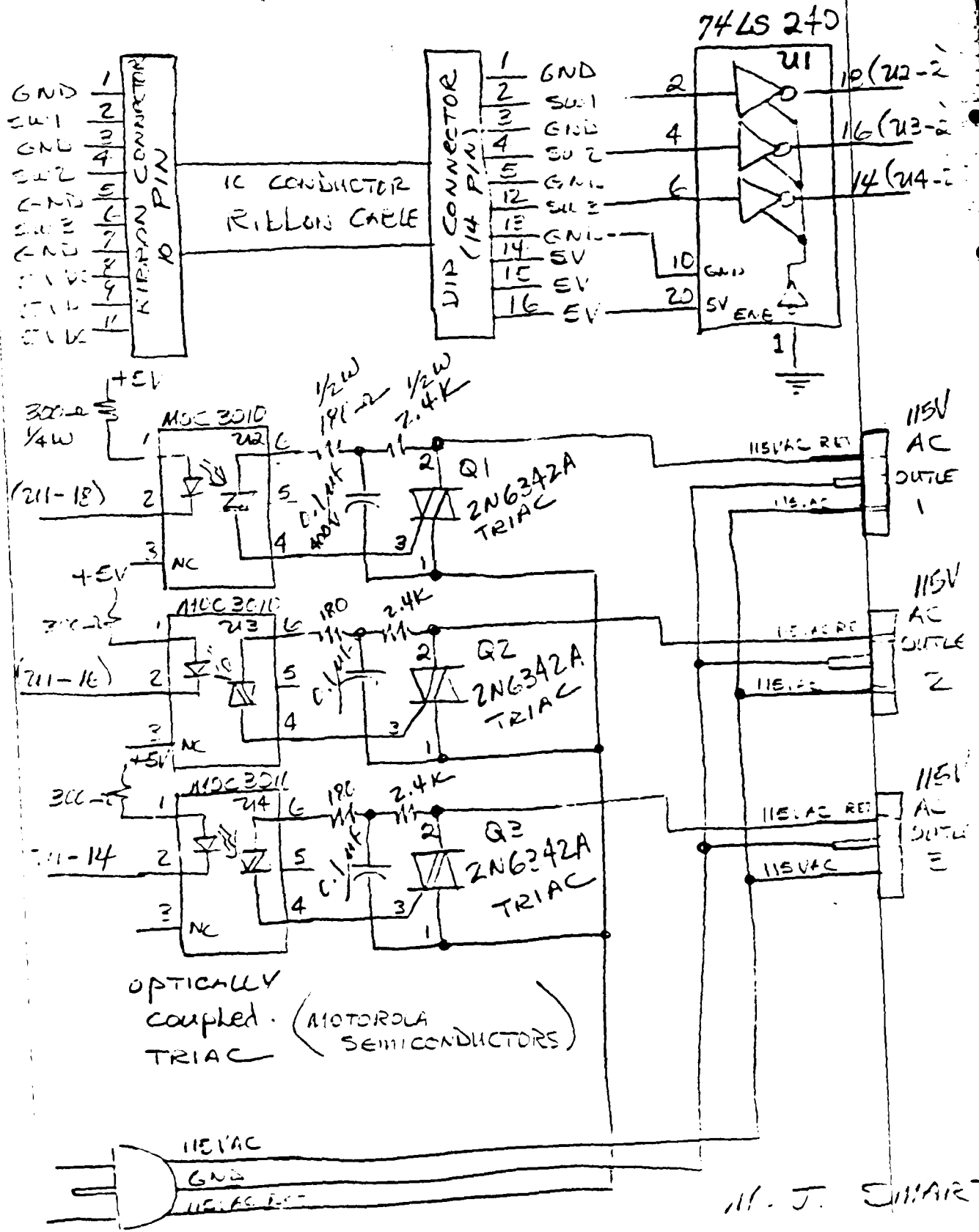
```


APPENDIX G

Machine Controller Software
(Assembly code for Z-80)

6 JUNE 82

LOGIC CONTROLLED AC POWER SWITCH (3 CHANNEL)



W. J. SMART

7 of the ribbon connector. Five volt power is required by this box for the 74LS 240 chip in the circuitry.

d-

Logic Controlled AC Power Switch

Theory of Operation:

A triac is a back-to-back diode pair which is gate controlled. A small (50ma) gate current switches the diode pair into conduction and effectively closes the "AC power" circuit. The circuit remains closed so long as gate current is supplied.

If gate current is interrupted, the triac output circuit will be interrupted (disconnected) where the alternating AC voltage passes through its next zero crossing voltage - because the diode pair will go into their open/off state.

A Motorola 12 Amp/200 Volt triac (2N6342A) used as the switch in the ~~115 VAC~~ ^{115 VAC} output circuit is controlled by a small "optically excited" Motorola triac (MOC 3010). This small triac supplies the gate current for the power line triac. The MOC 3010 triac is switched on by a TTL high logic level applied to an 74LS 240 inverting buffer, the buffer output goes low, causing the LED in the MOC 3010 to emit light which switches the triac, internal to the MOC 3010, on. This ^{then} ~~in~~ allows gate current to flow to the 2N 6342A power triac and switch it on.

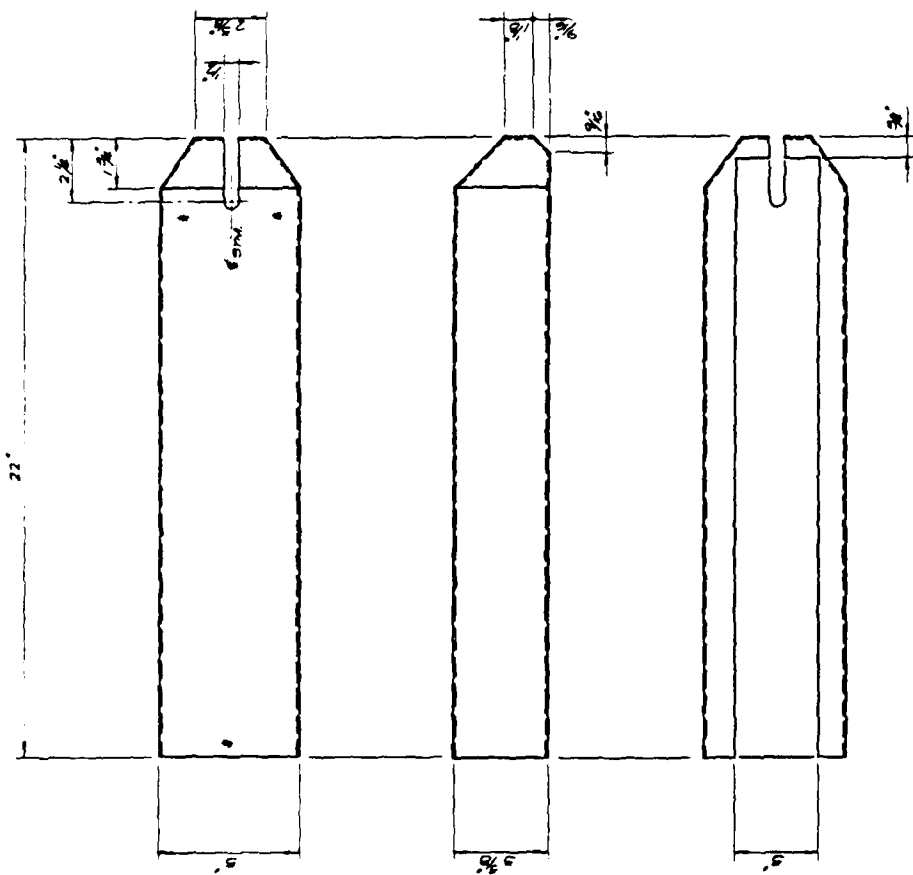
A low TTL level to the 74LS 240 buffer will extinguish all gate currents and the next zero crossing of the AC power line will cause both triacs to switch to their off states.

The RC network connected between the two triacs is for compensation due to the inductive load of the AC motor to be driven by the triac output.

The box must be plugged into the 115 VAC line with the plug provided. This line is fused with a 10 Amp fuse. The TTL logic signal is applied to pin 2, 4, or 6 on the ribbon connector to control outlet 1, 2, or 3. Apply +5 VDC to pins 8, 9, and 10 and +5 VDC return (ground) to pins 1, 3, 5, and

APPENDIX F

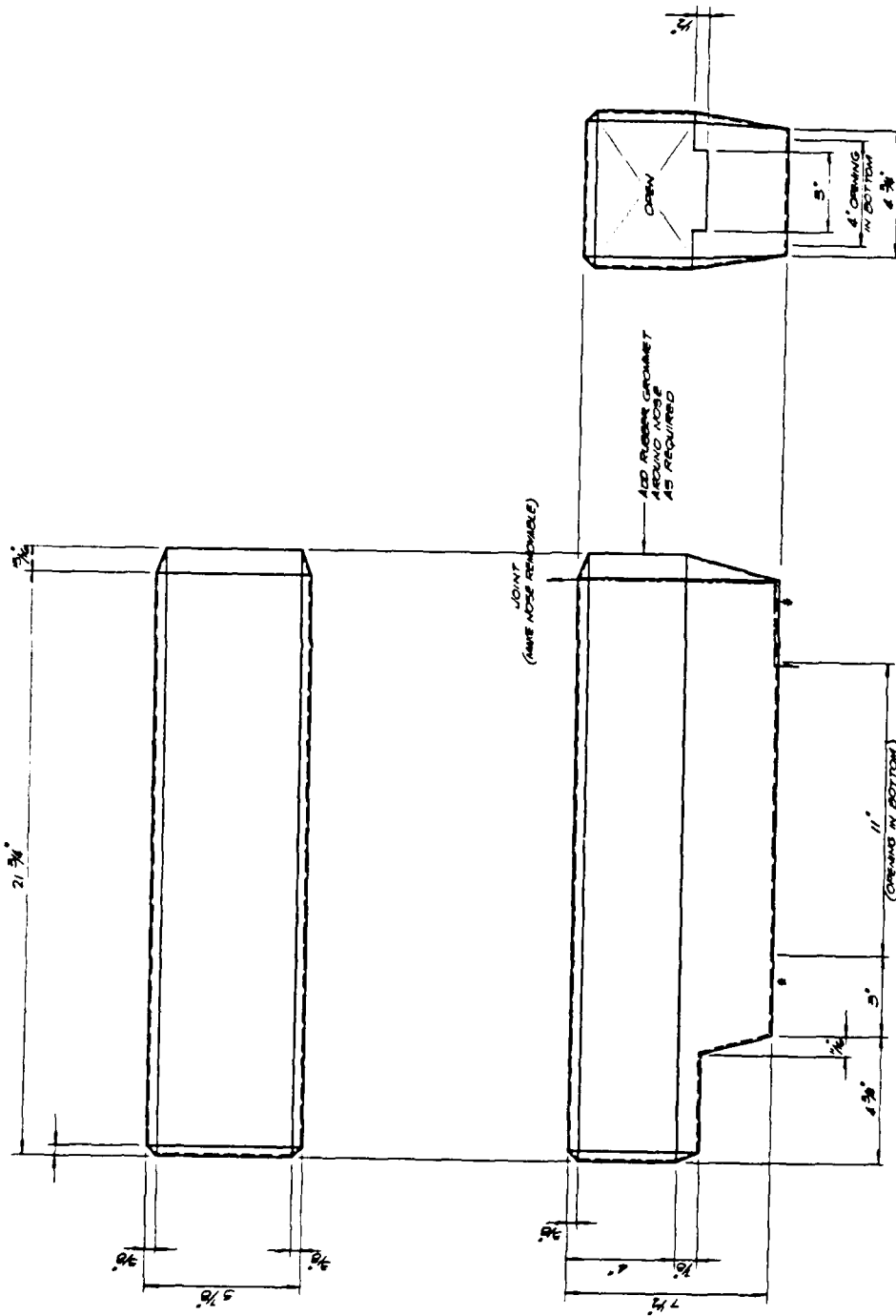
Documentation and Schematics for Logic Controlled AC Power Switch



DETAIL ITEM 32 (KOPIC OR EQUAL)
1/2 SCALE

NOTES
1. ALL DIMENSIONS ARE IN INCHES
2. 5/16\"

MARCH 9, 1982	BRIGHAM YOUNG UNIVERSITY	PROJECT TITLE	DETAIL ITEM 32	0029
CAC	UTAH BEARS	ROBOT		



- NOTES
1. ALL DIMENSIONS ARE AS SHOWN
 2. DIMENSIONS OF ASSEMBLY
 3. DIMENSIONS OF ASSEMBLY TO FRAME
 4. DIMENSIONS OF ASSEMBLY TO FRAME
 5. DIMENSIONS OF ASSEMBLY TO FRAME
 6. DIMENSIONS OF ASSEMBLY TO FRAME
 7. DIMENSIONS OF ASSEMBLY TO FRAME
 8. DIMENSIONS OF ASSEMBLY TO FRAME
 9. DIMENSIONS OF ASSEMBLY TO FRAME
 10. DIMENSIONS OF ASSEMBLY TO FRAME
 11. DIMENSIONS OF ASSEMBLY TO FRAME
 12. DIMENSIONS OF ASSEMBLY TO FRAME
 13. DIMENSIONS OF ASSEMBLY TO FRAME
 14. DIMENSIONS OF ASSEMBLY TO FRAME
 15. DIMENSIONS OF ASSEMBLY TO FRAME
 16. DIMENSIONS OF ASSEMBLY TO FRAME
 17. DIMENSIONS OF ASSEMBLY TO FRAME
 18. DIMENSIONS OF ASSEMBLY TO FRAME
 19. DIMENSIONS OF ASSEMBLY TO FRAME
 20. DIMENSIONS OF ASSEMBLY TO FRAME
 21. DIMENSIONS OF ASSEMBLY TO FRAME
 22. DIMENSIONS OF ASSEMBLY TO FRAME
 23. DIMENSIONS OF ASSEMBLY TO FRAME
 24. DIMENSIONS OF ASSEMBLY TO FRAME
 25. DIMENSIONS OF ASSEMBLY TO FRAME
 26. DIMENSIONS OF ASSEMBLY TO FRAME
 27. DIMENSIONS OF ASSEMBLY TO FRAME
 28. DIMENSIONS OF ASSEMBLY TO FRAME
 29. DIMENSIONS OF ASSEMBLY TO FRAME
 30. DIMENSIONS OF ASSEMBLY TO FRAME
 31. DIMENSIONS OF ASSEMBLY TO FRAME
 32. DIMENSIONS OF ASSEMBLY TO FRAME
 33. DIMENSIONS OF ASSEMBLY TO FRAME
 34. DIMENSIONS OF ASSEMBLY TO FRAME
 35. DIMENSIONS OF ASSEMBLY TO FRAME
 36. DIMENSIONS OF ASSEMBLY TO FRAME
 37. DIMENSIONS OF ASSEMBLY TO FRAME
 38. DIMENSIONS OF ASSEMBLY TO FRAME
 39. DIMENSIONS OF ASSEMBLY TO FRAME
 40. DIMENSIONS OF ASSEMBLY TO FRAME
 41. DIMENSIONS OF ASSEMBLY TO FRAME
 42. DIMENSIONS OF ASSEMBLY TO FRAME
 43. DIMENSIONS OF ASSEMBLY TO FRAME
 44. DIMENSIONS OF ASSEMBLY TO FRAME
 45. DIMENSIONS OF ASSEMBLY TO FRAME
 46. DIMENSIONS OF ASSEMBLY TO FRAME
 47. DIMENSIONS OF ASSEMBLY TO FRAME
 48. DIMENSIONS OF ASSEMBLY TO FRAME
 49. DIMENSIONS OF ASSEMBLY TO FRAME
 50. DIMENSIONS OF ASSEMBLY TO FRAME
 51. DIMENSIONS OF ASSEMBLY TO FRAME
 52. DIMENSIONS OF ASSEMBLY TO FRAME
 53. DIMENSIONS OF ASSEMBLY TO FRAME
 54. DIMENSIONS OF ASSEMBLY TO FRAME
 55. DIMENSIONS OF ASSEMBLY TO FRAME
 56. DIMENSIONS OF ASSEMBLY TO FRAME
 57. DIMENSIONS OF ASSEMBLY TO FRAME
 58. DIMENSIONS OF ASSEMBLY TO FRAME
 59. DIMENSIONS OF ASSEMBLY TO FRAME
 60. DIMENSIONS OF ASSEMBLY TO FRAME
 61. DIMENSIONS OF ASSEMBLY TO FRAME
 62. DIMENSIONS OF ASSEMBLY TO FRAME
 63. DIMENSIONS OF ASSEMBLY TO FRAME
 64. DIMENSIONS OF ASSEMBLY TO FRAME
 65. DIMENSIONS OF ASSEMBLY TO FRAME
 66. DIMENSIONS OF ASSEMBLY TO FRAME
 67. DIMENSIONS OF ASSEMBLY TO FRAME
 68. DIMENSIONS OF ASSEMBLY TO FRAME
 69. DIMENSIONS OF ASSEMBLY TO FRAME
 70. DIMENSIONS OF ASSEMBLY TO FRAME
 71. DIMENSIONS OF ASSEMBLY TO FRAME
 72. DIMENSIONS OF ASSEMBLY TO FRAME
 73. DIMENSIONS OF ASSEMBLY TO FRAME
 74. DIMENSIONS OF ASSEMBLY TO FRAME
 75. DIMENSIONS OF ASSEMBLY TO FRAME
 76. DIMENSIONS OF ASSEMBLY TO FRAME
 77. DIMENSIONS OF ASSEMBLY TO FRAME
 78. DIMENSIONS OF ASSEMBLY TO FRAME
 79. DIMENSIONS OF ASSEMBLY TO FRAME
 80. DIMENSIONS OF ASSEMBLY TO FRAME
 81. DIMENSIONS OF ASSEMBLY TO FRAME
 82. DIMENSIONS OF ASSEMBLY TO FRAME
 83. DIMENSIONS OF ASSEMBLY TO FRAME
 84. DIMENSIONS OF ASSEMBLY TO FRAME
 85. DIMENSIONS OF ASSEMBLY TO FRAME
 86. DIMENSIONS OF ASSEMBLY TO FRAME
 87. DIMENSIONS OF ASSEMBLY TO FRAME
 88. DIMENSIONS OF ASSEMBLY TO FRAME
 89. DIMENSIONS OF ASSEMBLY TO FRAME
 90. DIMENSIONS OF ASSEMBLY TO FRAME
 91. DIMENSIONS OF ASSEMBLY TO FRAME
 92. DIMENSIONS OF ASSEMBLY TO FRAME
 93. DIMENSIONS OF ASSEMBLY TO FRAME
 94. DIMENSIONS OF ASSEMBLY TO FRAME
 95. DIMENSIONS OF ASSEMBLY TO FRAME
 96. DIMENSIONS OF ASSEMBLY TO FRAME
 97. DIMENSIONS OF ASSEMBLY TO FRAME
 98. DIMENSIONS OF ASSEMBLY TO FRAME
 99. DIMENSIONS OF ASSEMBLY TO FRAME
 100. DIMENSIONS OF ASSEMBLY TO FRAME

DETAIL ITEM 31 (SCALE OR EQUAL)
- 1/4" SCALE

DATE: 9/19/22	BRIGHAM YOUNG UNIVERSITY	UTAH 84602	PROJECT: ROBOT	DETAIL ITEM 31	FORMSET 31/1	0028
DESIGN: C.A. CAV. DEPT						

```

;
;* THE CHARACTER IS RETURNED IN THE 'E' REGISTER.
;*****
;
054:  CD 06A2      GETCY: CALL SAVREG      ;SAVE PROCESSOR REGS.
054:  FD 2A 0322    LD      IY,(CYCTRL)  ;SET POINTER TO CY512 CONTROL REG.
054:  2E 42        LD      L,42H        ;SET CTRL REG FOR CY512 'OUTPUT'!
054:  FD 75 00      LD      (IY),L      ;.....
054:  FD CB 00 56   WAIT1: BIT  2,(IY)    ;CHECK CY512 STATUS..IS IT BUSY?
055:  CA 0540      JP      2,WAIT1      ;YES, WAIT
054:  2E 43        LD      L,43H        ;READY, MAKE AN I/O REQUEST.
055:  FD 75 00      LD      (IY),L      ;.....
055:  FD CB 00 56   WAIT2: BIT  2,(IY)    ;IS THE CY512 DATA READY?
055:  C2 7359      JP      NZ,WAIT2     ;NO,,WAIT 'TIL DATA IS READY.
056:  FD 5E 04      LD      E,(IY+4)    ;YES,FETCH CY512 OUTPUT CHARACTER.
056:  2E 40        LD      L,40H        ;CLEAR THE I/O REQUEST....
056:  FD 75 00      LD      (IY),L      ;...AND SET UP FOR CY INPUT.
056:  CD 06B9      CALL  RESTOR        ;RESTORE ACTIVE REGISTERS.
056:  C9          RET                ;EXIT THE GET CHARACTER S/R..

```

```

;
;*****
;* RESET THE ACTIVE CY512
;*
;* THIS ROUTINE SETS UP THE ACTIVE CY512 FOR PROPER COMMUNICATION.
;*****
;
056:  CD 06A2      RESET: CALL SAVREG      ;SAVE PROCESSOR REGISTERS.
056:  FD 2A 0322    LD      IY,(CYCTRL)  ;
057:  3E 44        LD      A,44H        ;THE ACTIVE CY512...
057:  FD 77 00      LD      (IY),A      ;..IS SET TO ASCII MODE AND..
057:  3E 40        LD      A,40H        ;..IS THEN 'RESET', AND..
057:  FD 77 00      LD      (IY),A      ;..THE RESET BIT IS CLEARED.
057:  CD 06B9      CALL  RESTOR        ;RETURN ACTIVE REGISTERS.
058:  C9          RET                ;EXIT THE 'RESET' S/R.

```

```

;
;*****
;* PRINT MESSAGE SUBROUTINE
;
;* PRINTS MESSAGE WHOSE STARTING ADDRESS IS IN THE 'DE' REGISTER
;* AND PRINT UNTIL A '$' SYMBOL IS INTERCEPTED.
;*****
;
058:  CD 06A2      PRINT: CALL SAVREG      ;...SAVE REGISTERS.
058:  0E 09        LD      C,PNTSTR     ;S/R TO PRINT A CHARACTER STRING.
058:  CD 0005      CALL  CPMTRY        ;
058:  CD 06B9      CALL  RESTOR        ;RESTORE THE REGISTERS.
058:  C9          RET                ;

```

```

;*****
;* 'HOME' AND CLEAR DISPLAY SCREEN
;*****

```

```

;
;
0580 CD 06A2 HOME: CALL SAVREG ;SAVE ACTIVE REGS.
0590 11 FC58 LD DE,HOME ;SET UP 6502 VECTOR LOCATION
0593 ED 53 F3D0 LD (A+VEC),DE ;TO CLEAR AND HOME SCREEN
0597 2A F3DE LD HL,(20CPU) ;GET 280 SOFCARD STARTING LOCATION
059A CD 05A1 CALL SOFTCD ;EXECUTE THE REQUEST
059D CD 06B9 CALL RESTOR ;RESTORE THE REGISTERS.
05A0 C9 RET

```

```

;
;
05A1 77 SOFTCD: LD (HL),A ;ACTIVATE SOFTCD TO DO 6502 S/R.
05A2 C9 RET

```

```

;
;
05A3 11 0000 CRLF: LD DE,CR ;SEND A CR/LF TO THE SCREEN
05A6 CD 05B0 CALL WICHR ;
05A9 11 000A LD DE,LF ;
05AC CD 05B0 CALL WICHR ;
05AF C9 RET

```

```

;
;
05B0 CD 06A2 WICHR: CALL SAVREG ;SAVE PROCESSOR REGISTERS.
05B3 0E 02 LD C,CHAR ;THIS ROUTINE TYPES THE CHAR ...
05B5 CD 0005 CALL COUNTRY ;...STORED IN THE 'DE' REGISTER
05B8 CD 06B9 CALL RESTOR ;RESTORE THE REGISTERS.
05BE C9 RET ;

```

```

;
;
;*****
;THIS SUBROUTINE CONVERTS THE TWO BYTE
;NUMBER IN THE HL REGISTER TO A FIVE
;DECIMAL DIGIT NUMBER STORED IN FIVE
;SEQUENTIAL REGISTERS.
;*****

```

```

;
; BINARY TO DECIMAL TO ASCII CONVERT ROUTINE.

```

```

;
05B0 CD 06A2 BDCONV: CALL SAVREG ;SAVE PROCESSOR REGISTERS
05B3 FD 21 0696 LD I1,BDFLG ;CLEAR 'CARRY',COUNTER,
05C3 DD 21 069D LD I2,ANSBUF+4 ;SET UP ANS BUFFER POINTER.
05C7 FD 36 02 05 LD (I1+2),5 ;SET CTR FOR FIVE DECIMAL DIGITS.
05CB 37 BDLOOP: SCF ;CLEAR 'CARRY' FLAG AND...
05CC 3F CCF ;...
05CD FD 36 00 00 LD (I1),0 ;SKIP,COUNT,AND DONE FLAGS.
05D1 01 1080 LD BC,1080H ;INITIALIZE 'BC' TO GENERATE QUOTIENT.
05D4 11 A000 LD DE,0A000H ;AND LOAD DE WITH THE 'NEW BASE'
05D7 3E 00 LD A,0 ;...
05D9 ED 52 BDCONT: SBC HL,DE ;BEGIN THE DIVIDING BY NEW BASE
05DB D2 05E3 JP NC,BDNEXT ;IF (HL) POSITIVE, CONTINUE.
05DE 19 ADD HL,DE ;WON'T GO, RESTORE ORIGINAL NUMBER.
05DF FD CB 00 D6 SET 2,(I1) ;AND SET THE 'SKIP ADD QUOTIENT' FLAG.
05E3 CB 3A BDNEXT: SRL C ;ROTATE DE 'RIGHT' ONE BIT
05E5 CB 1B RR E ;...
05E7 FD CB 00 4E BIT 1,(I1) ;HAVE WE ROTATED FIVE TIMES YET?
05EB C2 0606 JP NC,BDLSB ;YES, WORK ON LSB BYTE.

```

```

05EE  FD CB 00 56          BIT 2,(1Y)      ;NO, SO IS SKIP FLAG SET?
05F2  C2 05F6             JP  NZ,BDADV    ;YES, BYPASS THE ADD PARTIAL PRODUCT.
05F5  80                  ADD  A,B        ;NO,, ADD A BIT TO QUOTIENT.
05F6  CB 38               BDADV: SRL  B    ;SHIFT BIT COUNTER AND SEE OF ITS ZERO
05F8  C2 0688             JP  NZ,BDLINK   ;NO,GO CLEAR SKIP FLAG AND CONT DIVIDE.
05FB  3F                  CCF             ;YES, CLEAR 'CARRY',
05FC  FD CB 00 CE         SET 1,(1Y)      ;SET LSB FLAG(FIVE SHIFTS) AND
0600  47                  LD  B,A        ;SAVE PARTIAL DIVIDE QUOTIENT...
0601  3E 00              LD  A,0        ;..QUOTUH AND CLEAR THE 'A' REG.
0603  C3 0688             JP  BDLINK     ;AND CONTINUE.
0606  FD CB 00 56         BDLSB: BIT 2,(1Y) ;CHECK THE SKIP FLAG ?
060A  C2 060E             JP  NZ,BDINC    ;BYPASS THE ADD IF SET.
060D  81                  ADD  A,C        ;NOPE.. ADD A BIT TO THE QUOTIENT.
060E  CB 39               BDINC: SRL  C    ;SHIFT BIT CTR AND TEST FOR 13 SHIFTS.
0610  C2 0688             JP  NZ,BDLINK   ;NOT YET, CLR SKIP FLG AND CONT DIV.
0613  3F                  CCF             ;CLR CARRY FLAG AND SAVE 'QUOTUH'.
0614  4F                  LD  C,A        ;...
0615  7D                  LD  A,L        ;CONVERT THE REMAINDER TO ASCII...
0616  C6 30              ADD  A,30H      ;...AND...
0618  DD 77 00           LD  (1X),A      ;STORE REM(DEC DIGIT) IN ANSBUFFER.
061B  DD 28              DEC  1X         ;MOVE ANSBUF POINTER TO NEXT LOC.
061D  60                  LD  H,B        ;MOVE 'QUOTIENT' TO 'HL' FOR
061E  69                  LD  L,C        ;FOR NEXT DIVIDE ROUND.
061F  FD 35 02           DEC  (1Y+2)     ;DECREMENT 'FIVE DIGIT' CTR AND
0622  CA 06BF             JP  Z,BDFSET   ;PREPARE TO EXIT IF ZERO.
0625  FD CB 00 46         BDEXIT: BIT 0,(1Y) ;CHECK THE 'EXIT FLAG'....
0629  CA 05CE             JP  Z,BDLOOP   ;NOT DONE..GET NEXT DIGIT.
062C  CD 0633             CALL PDIGTS    ;GO PRINT THE CONVERTED DIGITS.
062F  CD 06B9             CALL RESTOR   ;RESTORE REGISTERS.
0632  C9                  RET           ;EXIT .....

;
;
0633  CD 06A2             PDIGTS: CALL SAVREG ;SAVE THE 'X' REGISTER.
0636  DD 21 0699          LD  1X,ANSBUF ;SET 1X POINTER TO 5 DIGIT BUFFER.
063A  FD 21 06B7          LD  1Y,PDIGFG ;SET UP AND...
063E  3E 00              LD  A,0        ;...CLEAR..
0640  32 0687          LD  (PDIGFG),A ;...FLAG BITS.
0643  3E 04              LD  A,4        ;SET UP DIGIT COUNTER.
0645  FE 00              PDLOOP: CP 0    ;IS CNTR TO 5TH DIGIT YET?
0647  C2 0672             JP  NZ,PDELSE ;NO, CONTINUE.
064A  FD CB 00 C6         SET 0,(1Y)     ;SET FLAGS...
064E  FD CB 00 CE         SET 1,(1Y)     ;...
0652  FD CB 00 4E         PDCONT: BIT 1,(1Y) ;ARE WE STILL RECEIVING LEADING ZEROS?
0656  CA 0661             JP  Z,PDOUT    ;NO, CONTINUE.
0659  16 00              LD  D,0        ;
065B  DD 5E 00           LD  E,(1X)     ;LOAD DIGIT INTO OUTPUT REGISTER.
065E  CD 05B0             CALL WICHR     ;AND PRINT IT !
0661  DD 23              PDOUT: INC  1X  ;INCREMENT DIGIT POINTER AND
0663  3D                  DEC  A        ;DECREMENT THE DIGIT COUNTER.
0664  FD CB 00 46         BIT 0,(1Y)     ;ARE WE DONE YET ??
0668  CA 0645             JP  Z,PDLOOP   ;NO, KEEP AT IT.
066B  CD 05A3             CALL CRLF     ;DONE,RTN CURSOR AND EXIT..
066E  CD 06B9             CALL RESTOR   ;RESTORE THE X REGISTER
0671  C9                  RET           ;.....

;
0672  CD 06A2             PDELSE: CALL SAVREG ;SAVE THE COUNT AND

```

```

0675 DD 7E 00 LD A,(IX) ;IS THE CHARACTER A '0' ??
0678 FE 30 CP 30H ;
067A CA 0681 JP Z,PDNEX ;YES, PREPARE TO EXIT
067D FD CB 00 CE SET 1,(IY) ;NO , SET LEADZERO FLAG.
0681 CD 0689 PDNEX: CALL RESTOR ;RESTORE REGISTERS
0684 C3 0652 JP PDCONT ;AND CONTINUE PROCESSING.
0687 00 PDIGF6: 0 ;
;
;
068B FD CB 00 5c BDLINK: RES 2,(IY) ;CLEAR THE 'SKIP' FLAG.
068C C3 0509 JP BDCONT ;GO CONTINUE THE DIVIDE PROCESS.
068F FD CB 00 Cc BDFSET: SET 0,(IY) ;SET THE EXIT FLAG (DONE,FIVFL6).
0693 C3 0625 JP BDEXIT ;.....AND GO EXIT.
0696 00 BDFL6: 0 ;FLAG REGISTER.
0697 00 0 ;
0698 00 BDCNT: 0 ;DIGIT COUNTER.
0699 00 ANSBUF: 0 ;THIS IS THE FIVE DIGIT ANSWER BUFFER.
069A 00 0 ;...
069B 00 0 ;...
069C 00 0 ;..
069D 00 0 ;.
;
;
;*****
;* USER STACK AND SAVE PROCEEDURE
;*****
;
069E 17 USRSTK: 17H ;SET THE USER STACK BELOW PROGRAM AREA.
069F 0F 0FH ;
06A0 00 SYSSTK: 0 ;SAVE SYSTEM STACK POINTER.
06A1 00 0 ;
;
;*****
;* SAVE PROCESSOR REGISTERS
;*****
;
06A2 ED 73 06A0 SAVREG: LD (SYSSTK),SP ;SAVE THE SYSTEM STACK POINTER.
06A6 ED 7B 069E LD SP,(USRSTK) ;LOAD USER STACK POINTER.
06AA F5 PUSH AF ;SAVE ACTIVE REGISTERS...
06AE E5 PUSH HL ;...
06AC DD E5 PUSH IX ;...
06AE FD E5 PUSH IY ;...
06B0 ED 73 069E LD (USRSTK),SP ;SAVE USRSTK POINTER.
06B4 ED 7B 06A0 LD SP,(SYSSTK) ;RESTORE SYSTEM STACK POINTER.
06B8 C9 RET ;..EXIT
;
;
;*****
;* RESTORE PROCESSOR REGISTERS
;*****
;
06B9 ED 73 06A0 RESTOR: LD (SYSSTK),SP ;SAVE SYSTEM STACK PTR.
06BD ED 7B 069E LD SP,(USRSTK) ;LOAD USER STACK POINTER.
06C1 FD E1 POP IY ;RESTORE THE ACTIVE REGISTERS.
06C3 DD E1 POP IX ;...
06C5 E1 POP HL ;...

```

06C6	F1	POF	AF	;...
06C7	ED 73 069E	LD	(USRSTK),SP	;SAVE USER STACK POINTER.
06CB	ED 7B 06A0	LD	SP,(SYSSTK)	;RESTORE SYSTEM STACK POINTER.
06CF	C9		RET	;...AND EXIT
				;SET UP STRING BUFFER SPACE,
06D0	0D 0D 0D 0D	CRRTN: DB	13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13	
06D4	0D 0D 0D 0D			
06D8	0D 0D 0D 0D			
06DC	0D 0D 0D 0D			
06E0	0D 0D 0D 0D	DB	13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13	
06E4	0D 0D 0D 0D			
06E8	0D 0D 0D 0D			
06EC	0D 0D 0D 0D			
06F0	0D 0D 0D 0D	DB	13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13	
06F4	0D 0D 0D 0D			
06F8	0D 0D 0D 0D			
06FC	0D 0D 0D 0D			
0700	0D 0D 0D 0D	DB	13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13	
0704	0D 0D 0D 0D			
0708	0D 0D 0D 0D			
070C	0D 0D 0D 0D			
0710		BUFF: DS	64	;LOCATION BUFF+1 CONTAINS STR LENGTH
0750	2A 2A 2A 2A	HEADER: DE	"*****",13,10	
0754	2A 2A 2A 2A			
0758	2A 2A 2A 2A			
075C	2A 2A 2A 2A			
0760	2A 2A 2A 2A			
0764	2A 2A 2A 2A			
0768	2A 2A 2A 2A			
076C	2A 2A 2A 2A			
0770	2A 2A 2A 2A			
0774	2A 2A 2A 0D			
0778	0A			
0779	2A 20 20 20	DB	"* CYS12 CONTROL PROGRAM *	,13,10
077D	20 20 20 20			
0781	20 43 59 35			
0785	31 32 20 43			
0789	4F 4E 54 52			
078D	4F 4C 20 50			
0791	52 4F 47 52			
0795	41 4D 20 20			
0799	20 20 20 20			
079D	20 20 2A 0D			
07A1	0A			
07A2	2A 20 49 4E	DB	"* IN Z80 ASSEMBLY CODE BY M. J. SMART *",13,10	
07A6	20 5A 38 30			
07AA	20 41 53 53			
07AE	45 4D 42 4C			
07B2	59 20 43 4F			
07B6	44 45 20 42			
07BA	59 20 4D 2E			
07BE	20 4A 2E 20			
07C2	53 4D 41 52			
07C6	54 20 2A 0D			
07CA	0A			

07CB 2A 2A 2A 2A
07CF 2A 2A 2A 2A
07D3 2A 2A 2A 2A
07D7 2A 2A 2A 2A
07DB 2A 2A 2A 2A
07DF 2A 2A 2A 2A
07E3 2A 2A 2A 2A
07E7 2A 2A 2A 2A
07E8 2A 2A 2A 2A
07EF 2A 2A 2A 00
07F3 0A 24

DB "*****",13,10,"8"

END

Macros:

Symbols:

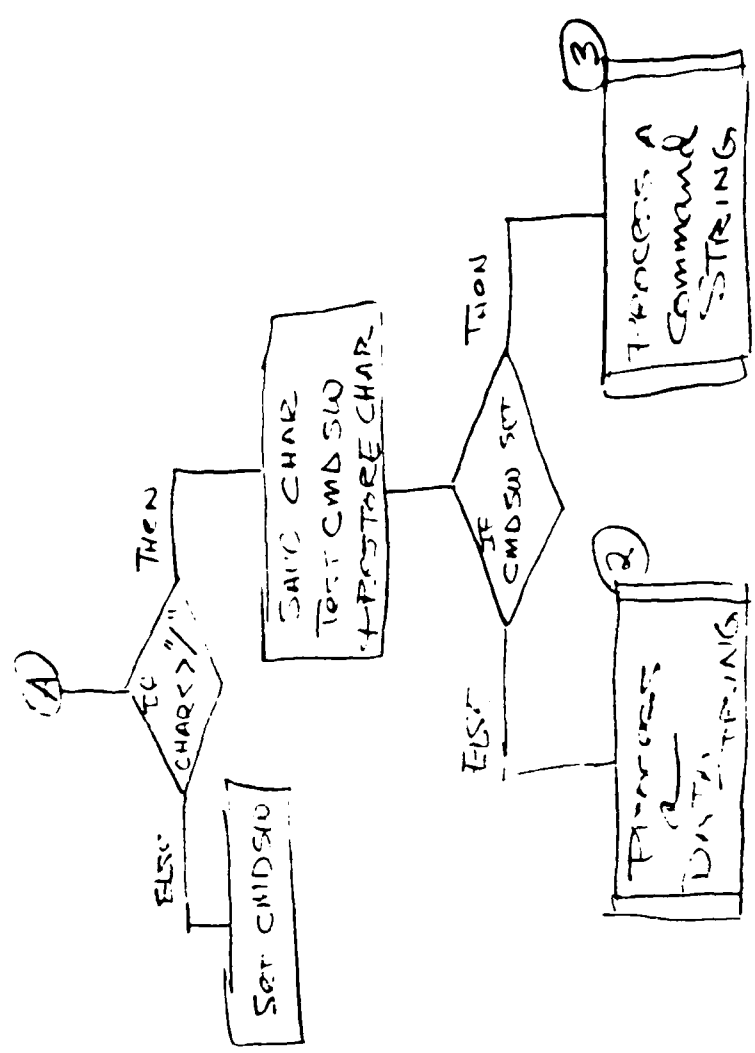
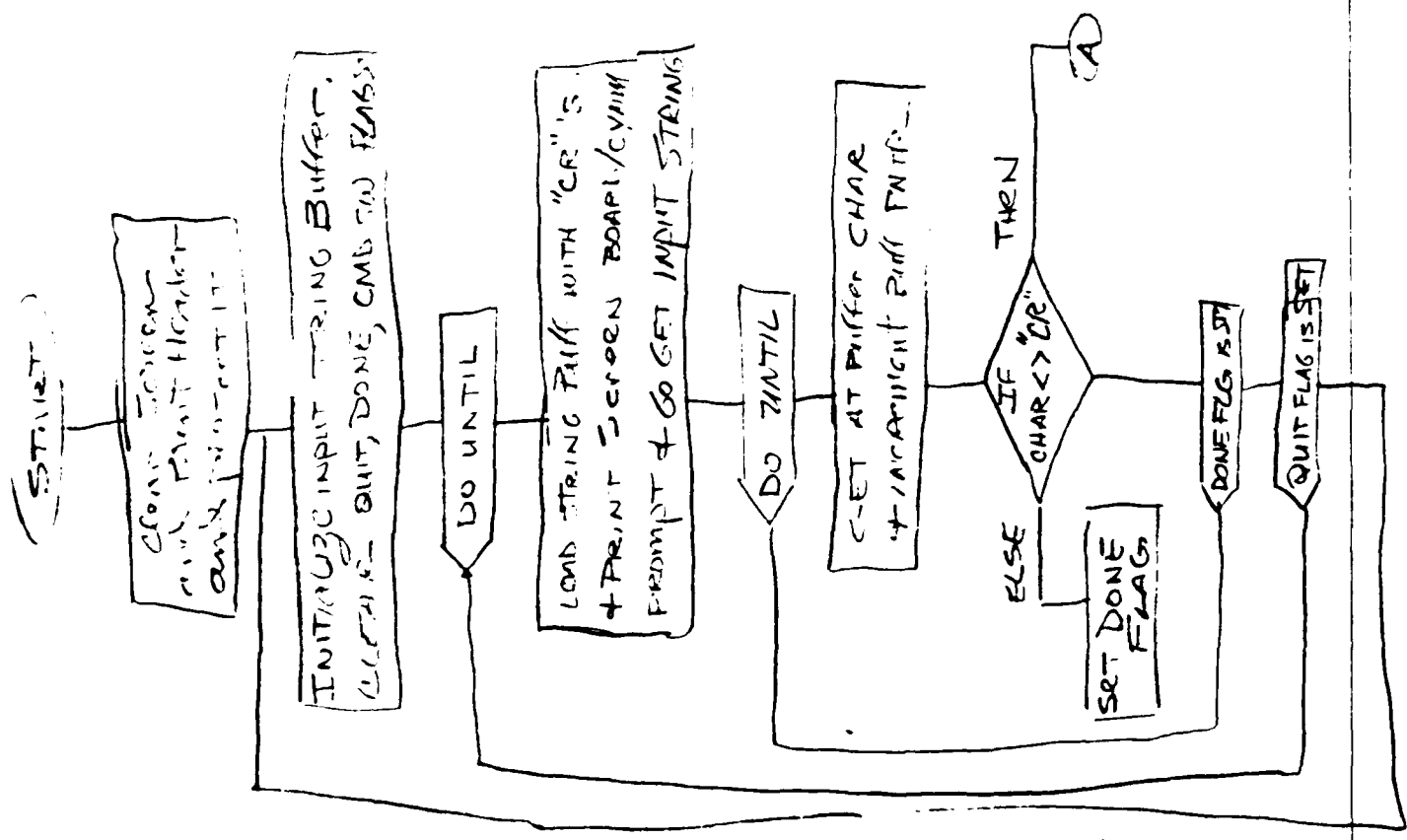
ABACC	F045	ABELL	FF3A	ABHME	FC58	ABPREG	F048
ABPREK	FF2D	ABSKPT	F049	ABVEC	F3D0	ABXREG	F046
ABYREG	F047	ABOARD	028F	ACYNM	0290	ANSBUF	0699
BDDEXIT	0554	BDONT	0688	BDCONT	05D9	BDCOMV	058C
		BDPLG	0698	BDFSET	028F	BDINC	060E
BOLINK	0688	BDLOOP	05CB	BDLSE	0606	BDNEXT	05E3
BKOND	0200	BOARD	028D	BUFF	0710	CDBDUD	02F8
CDCMBD	02CB	CDCYUD	02EA	CDEXIT	02CA	CDSET	02C6
CDG00D	02AC	CDLST1	0330	CDLST2	0333	CDMS6	02D4
CDVY	0324	CHAR	024F	CHRLST	0388	CDLST	032A
CHDSTR	029C	CPMTR	0005	CR	000D	CRLF	05A3
CPMTR	06D0	CYCTRL	0322	CYNU	028B	CYUPDT	0306
DATASG	026F	DTAEXT	036C	DTASTF	0335	DTVY	037A
EXIT	025E	FACTOR	049C	FLGREG	027C	GETCY	0541
GETSTR	000A	GAIT1	054D	GAIT2	0559	HEADER	075D
HME	0580	INPSTR	0293	INVALD	036D	LF	000A
LOOP	022A	MORCHR	024A	NUMBER	0498	NULDNG	038D
PDCONT	0652	PDELSE	0672	PD16F6	0687	PDIGTS	0633
PDLOOP	0645	PDNE	0681	PDOOT	0661	PEQUAL	0505
PNTSTR	0009	PRINT	0581	PROMPT	0291	RATE	0498
RCHAR	0001	RESET	056C	RESTOF	0689	RESTR	021C
RTNCPM	027D	SAVREG	06A2	SCHDS	0531	SCHAIT	0520
SEND	035E	SENDY	0519	SETSW	0275	SLOPE	049A
SLOT	0007	SOFTCD	05A1	SQUIT	0384	STAT	E0F0
STATUS	049D	SYSSTK	06A0	TOPLIN	F022	USRSTK	069E
VCDUNT	0497	VECDUM	04F2	VED	046D	VEDUM	04E7
VEDSCT	0449	VEDSPY	04D2	VEEXIT	0456	VEFACT	04C5
VEFLST	04DB	VEFS	0466	VEHST	04CF	VELOOP	03D7
VENUMB	04AA	VEOTPT	046C	VERATE	04BD	VERIFY	03C3
VESLOF	0484	VFETCH	047A	VFLOOF	0486	VLDCMP	0353
VOLLOOP	046E	VTHREE	04A6	VTWC	04A2	VZERO	049E
WCHAR	0580	WCHAR	0002	ZCPL	F3D0		

No Fatal error(s)

B)

(1)

CYDRIVER FLOWCHART MAIN PROGRAM (BACKGROUND)



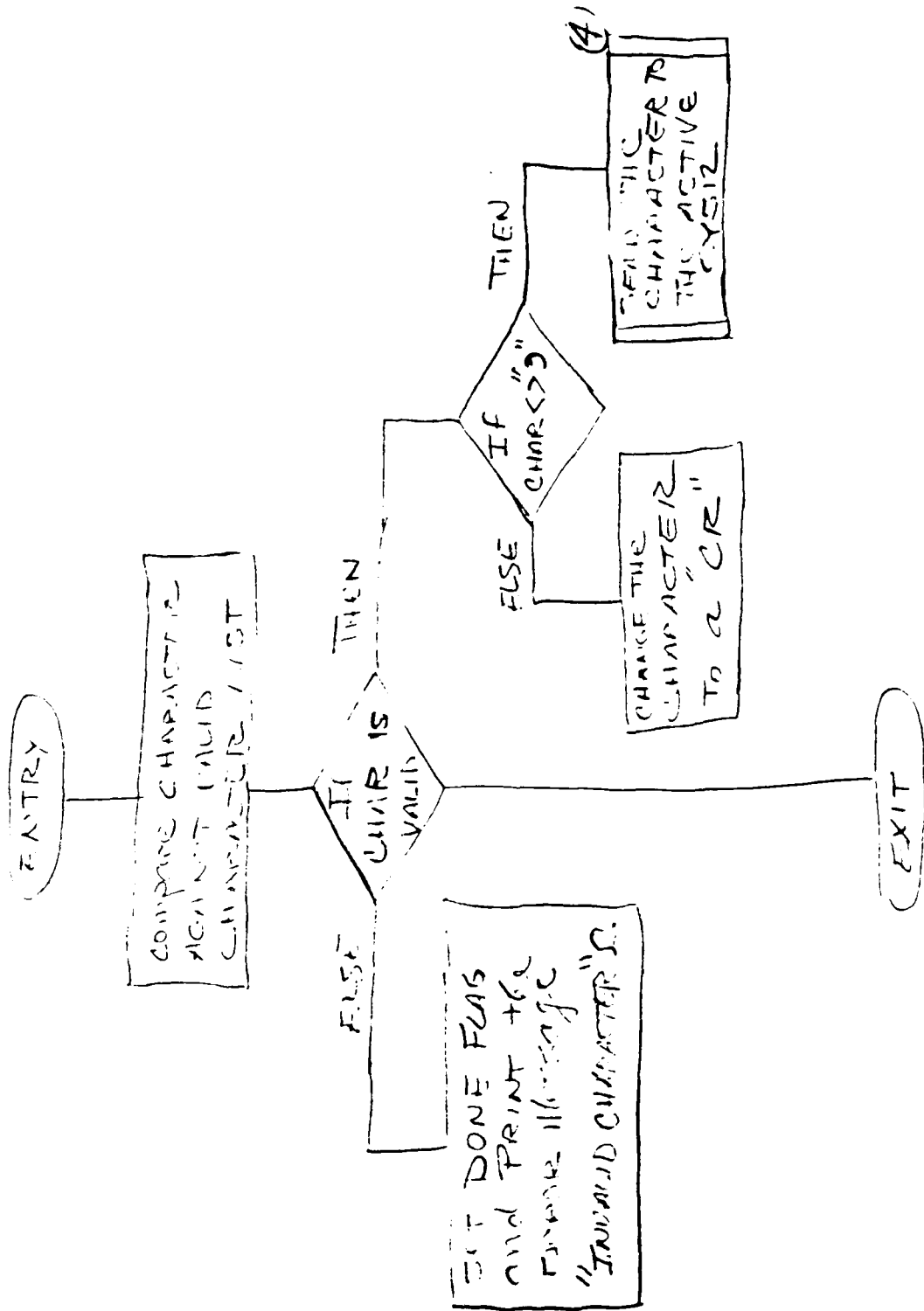
- | | |
|----|----|
| 1A | 1A |
| 1B | 1B |
| 1C | 1C |
| 1D | 1D |
| 1E | 1E |
| 1F | 1F |
-
- | |
|----------------|
| 4 BOXES |
| 2 BONKERS/BOX |
| 3 CYS125/BOARD |

W. L. Hunt

18 June 8

PROCESS DATA STRINGS S/A

(2)



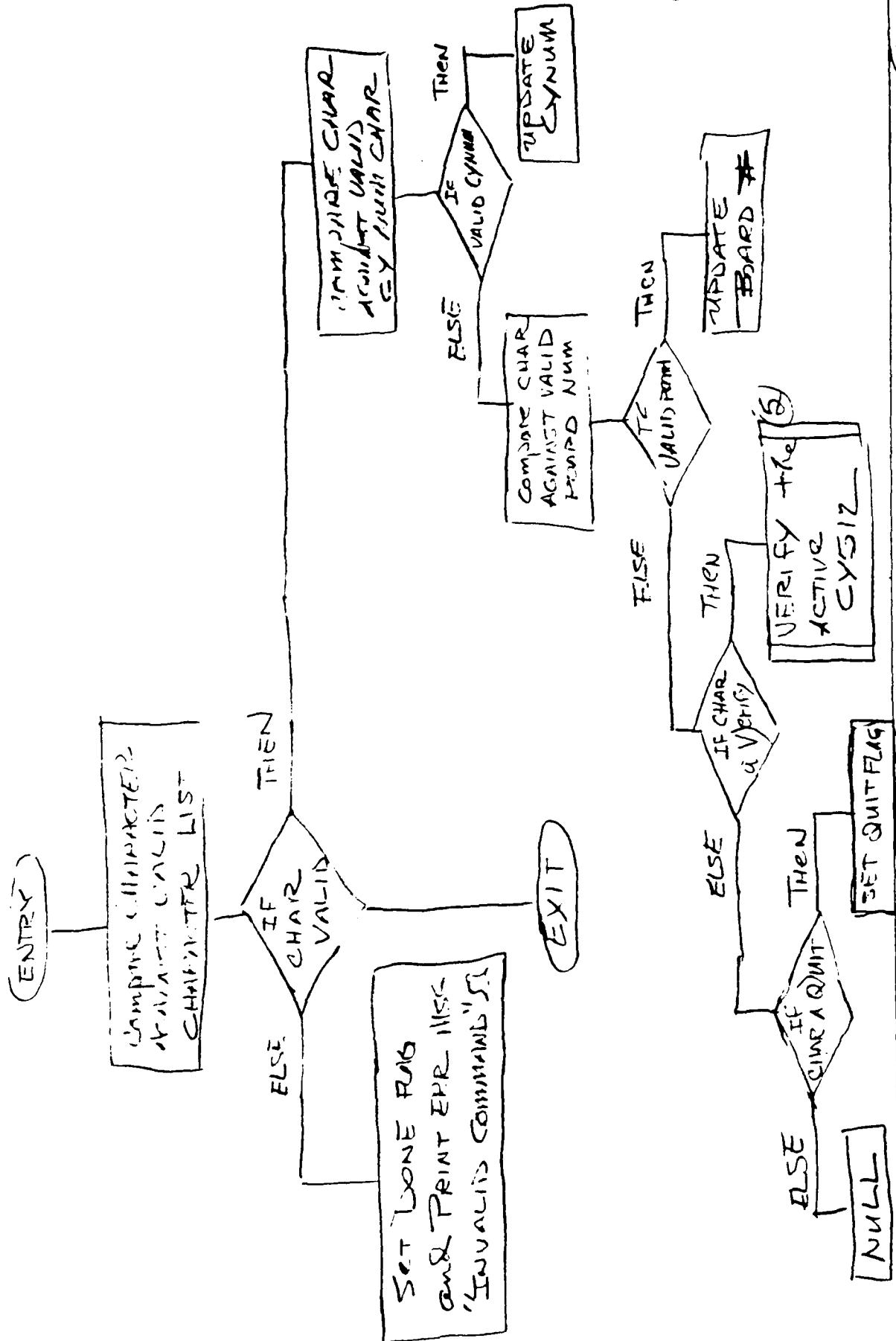
RUFF
RUFF SIZE
String LENGTH (starts out with CR's in the string)

CM S. Smart

18 June 85

Process Command String 5/

(3)



m f smart

APPENDIX H

Machine Language Interface and Communication Program

SOFTWARE:

MACHINE LANGUAGE INTERFACE AND COMMUNICATION PROGRAMS:

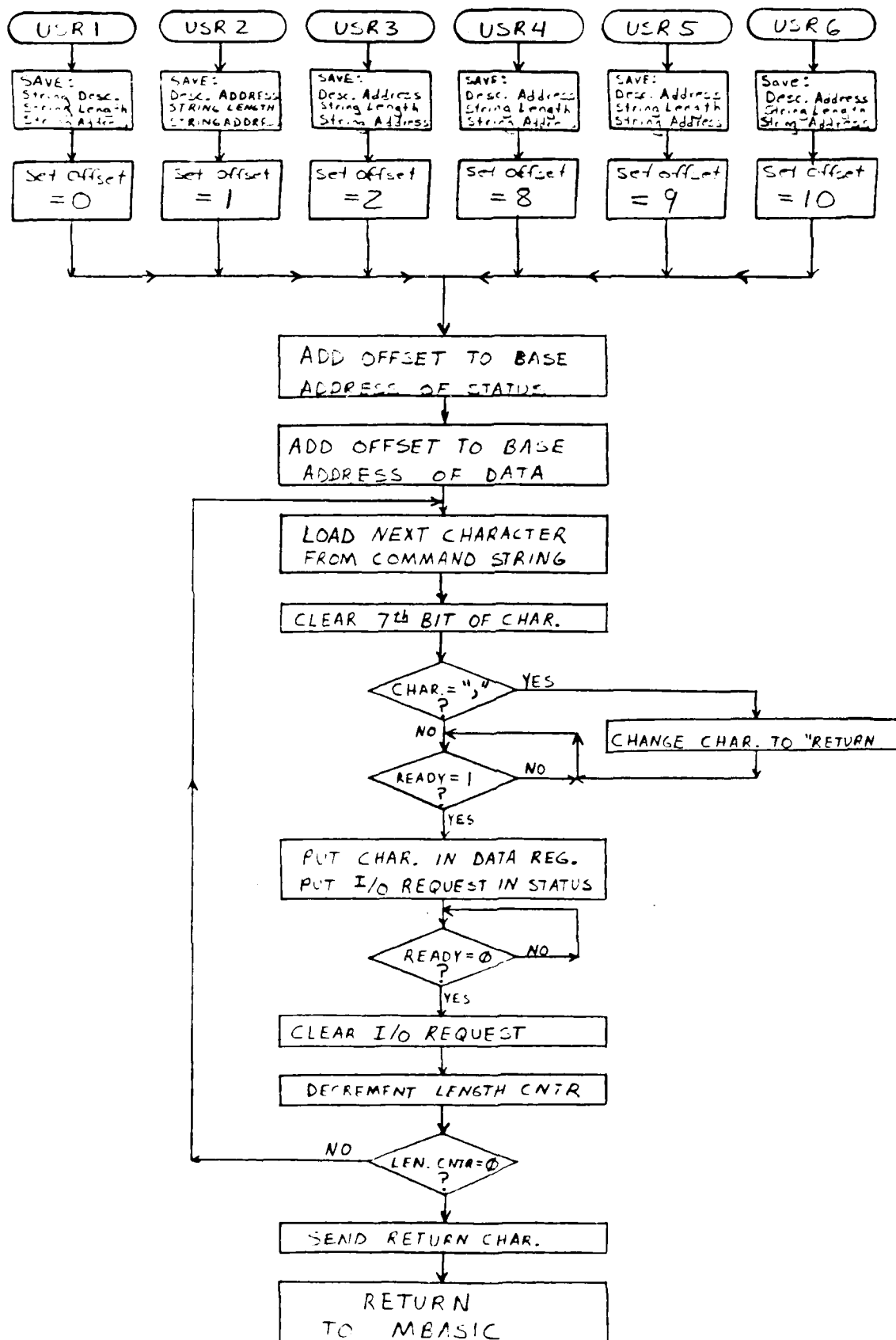
Following are assembler listings for two machine language programs which must be loaded to interface the APPLE computer to the "Blue Box" and to the serial RS232 card which allows communication with other computers. They are as follows:

CYDRIVER.HEX - The 8080 machine language program which allows Microsoft Basic to 'talk' with the "Blue Box"

COMM.HEX - The 8080 machine language program which allows Microsoft Basic to 'talk' with other computers and peripherals through the RS232 interface card.

FIGURE 4 - Flow Chart

For details on the MBASIC "USR" Command, see Softcard Documentation Volume II, Appendix C.



```

*****
;*
;*          CYDRIVER      BY JUSIN D. REDD  -BYU CAM LAB-
;*                      OCTOBER, 1983
;*                      (ORIGINAL 6502 VERSION BY BRANDT C. REDD)
;*
;* THIS PROGRAM TAKES A STRING FROM MBASIC THROUGH THE 'USR' FUNCTION
;* AND SENDS IT TO THE CY-512 6-AXIS STEPPER MOTOR CONTROLLER.
;*
;* COMMUNICATION IS THROUGH AN 'APPLE EXPANDER' PARALLEL COMMUNICATION
;* CARD MANUFACTURED BY ROBOTIC SYNERGY, INC. THIS PROGRAM ASSUMES
;* THAT THE CARD IS PLACED IN SLOT 7, BUT MAY BE EASILY MODIFIED.
;*
;* TO MODIFY FOR OTHER THAN SLOT 7, CHANGE LINES 30 AND 31 AS FOLLO
;* 30: STRASE: EQU      0E0X0   (WHERE X= SLOT# + 8)
;* 31: DTRASE: EQU      0E0X4   (WHERE X= SLOT# + 8)
;*
;* THE DEF USRX STATEMENT MUST BE USED FROM MBASIC TO ENTER THIS
;* PROGRAM AT THE PROPER POINT FOR THE DESIRED AXIS
;*
*****
;
21: C400          ORG      0C400H          ;START ASSEMBLY AT LOCATION C400H
;
;*****
;* SET-UP
;*****
;
31: C400          STRING: DS      2
32: C402          DESC:   DS      2
33: C404          LENGTH: DS      1
34: C405          STAT:   DS      2
35: C407          DATA:  DS      2
36: E0F0 =        STRASE: EQU      0E0F0H          ;** CHANGE IF SLOT <> 7 **
37: E0F4 =        DTRASE: EQU      0E0F4H          ;** CHANGE IF SLOT <> 7 **
38:
39:
;*****
;* ENTRY POINTS
;*****
;
43: C409 CD57C4    CY1:   CALL      INIT              ;INITIALIZE
44: C40C 0600      MVI      B,00H          ;OFFSET STAT REG VALUE FOR CY#1
45: C40E 0E00      MVI      C,00H          ;OFFSET DATA REG VALUE FOR CY#1
46: C410 CD74C4    CALL      SETUP          ;CALL SUB TO SETUP STAT AND DATA
47: C413 C387C4    JMP       MAIN          ;JUMP TO MAIN PROGRAM
48: C416 CD57C4    CY2:   CALL      INIT              ;INITIALIZE
49: C419 0601      MVI      B,01H          ;OFFSET STAT REG VALUE FOR CY#2
50: C41B 0E01      MVI      C,01H          ;OFFSET DATA REG VALUE FOR CY#2
51: C41D CD74C4    CALL      SETUP          ;SUB TO SET UP STAT & DATA ADDRESSES
52: C420 C397C4    JMP       MAIN          ;JUMP TO MAIN PROGRAM
53: C423 CD57C4    CY3:   CALL      INIT              ;INITIALIZE
54: C426 0602      MVI      B,02H          ;OFFSET STAT REG VALUE FOR CY#3
55: C428 0E02      MVI      C,02H          ;OFFSET DATA REG VALUE FOR CY#3
56: C42A CD74C4    CALL      SETUP          ;SET UP STAT & DATA REG ADDRESSES
57: C42D C397C4    JMP       MAIN          ;JUMP TO MAIN PROGRAM
58: C430 CD57C4    CY4:   CALL      INIT              ;INITIALIZE
59: C433 0608      MVI      B,08H          ;OFFSET STAT REG VALUE FOR CY#4
60: C435 0E08      MVI      C,08H          ;OFFSET DATA REG VALUE FOR CY#4

```

```

10 REM *****
20 REM *
30 REM *          LOADASM.BAS
40 REM *          BY
50 REM *          JUSTIN D. REDD      - BYU CAM LAB -
60 REM *          OCTOBER, 1983
70 REM * THIS PROGRAM CHECKS TO SEE IF COMM.HEX
80 REM * AND CYDRIVER.HEX HAVE BEEN LOADED INTO
90 REM * MEMORY. IF THEY HAVE NOT THEN THE PROGRAM
100 REM* CHAINS TO THE 'BLOAD' PROGRAM AND LOADS
110 REM* THEM INTO MEMORY.
120 REM*
130 REM* THE LOCATIONS C4F1H - C4F7H ARE USED TO
140 REM* INDICATE PROGRAM LOADING STATUS. LOCATIONS
150 REM* C4F1 - C4F5H WILL CONTAIN THE ASCII CODES
160 REM* FOR "VALID" IF DATA IS VALID AND LOCATIONS
170 REM* C4F6H & C4F7H WILL CONTAIN ASCII "0", OR "1".
180 REM* 0 = PROGRAM NOT LOADED 1 = PROGRAM LOADED
190 REM* C4F6H CORRESPONDS TO CYDRIVER.HEX
195 REM* C4F7H CORRESPONDS TO COMM.HEX
200 REM*
210 REM*****
220 REM
230 COMMON CALLER$,CLINE
240 COMMON PRGM$,SLINE,HFILE$
250 REM
260 REM *****
270 REM * CHECK STATUS
280 REM *****
290 REM
300 PRINT:INVERSE:PRINT "CHECKING FOR CYDRIVER.HEX AND COMM.HEX";:NORMAL:PRINT
310 MESS$=""
320 FOR CHAR = 0 TO 6
330 LOCATION = &HC4F1 + CHAR
340 CHAR$=CHR$(PEEK(LOCATION))
350 MESS$=MESS$ + CHAR$:CHAR$=""
360 NEXT CHAR
370 IF LEFT$(MESS$,5) < > "VALID" THEN GOTO 440
380 X$=RIGHT$(MESS$,2)
390 IF X$="10" THEN GOTO 640
400 IF X$="01" THEN GOTO 550
410 IF X$="11" THEN GOTO 730
420 GOTO 440
430 REM
440 REM *****
450 REM * LOAD BOTH PROGRAMS
460 REM *****
470 REM
480 PRINT:INVERSE:PRINT "LOADING CYDRIVER.HEX AT C400H";:NORMAL:PRINT
490 PRGM$="B:LOADASM.BAS":SLINE=510:HFILE$="B:CYDRIVER.HEX"
500 CHAIN "B:BLOAD.BAS",,ALL
510 PRINT:INVERSE:PRINT "LOADING COMM.HEX AT C500H";:NORMAL:PRINT
520 SLIN = 530:HFILE$="B:COMM.HEX":CHAIN "B:BLOAD.BAS",,ALL
530 GOTO 730
540 REM
550 REM *****

```

```
540 REM *****
550 REM * DO AS COMMANDED
560 REM *****
570 REM
574 MENU$="MENU"
580 IF CHOICE$ = "1" THEN CHAIN "INTERP.BAS"
590 IF CHOICE$ = "2" THEN CHAIN "CYEXEC.BAS"
600 IF CHOICE$ = "3" THEN CHAIN "LZERO.BAS",,ALL
610 IF CHOICE$ = "4" THEN CHAIN "MZERO.BAS"
615 IF CHOICE$ = "5" THEN CHAIN "LINEXEC.BAS",,ALL
620 IF CHOICE$ = "6" THEN HOME:END
```

```

10 REM *****
20 REM *
30 REM * MENU.BAS BY JUSTIN D. REDD - BYU CAM LAB -
40 REM * DECEMBER, 1983
50 REM *
60 REM * THIS PROGRAM PROVIDES A SELF EXPLANATORY MENU
70 REM * THAT CHAINS TO OTHER PROGRAMS TO CREATE AND
80 REM * EXECUTE COMMAND FILES FOR THE MINI LATHE AND
90 REM * MINI MILL.
100 REM*
110 REM*****
120 REM
130 REM *****
140 REM * CHECK FOR CYDRIVER.HEX IN MEMORY
150 REM *****
160 REM
170 PRINT:INVERSE:PRINT "CHECKING FOR CYDRIVER.HEX";:NORMAL:PRINT
180 MESS$=""
190 FOR CHAR = 0 TO 6
200 LOCATION = &HC4F1 + CHAR
210 CHAR$=CHR$(PEEK(LOCATION))
220 MESS$=MESS$ + CHAR$:CHAR$= ""
230 NEXT CHAR
240 IF LEFT$(MESS$,5) <> "VALID" THEN GOTO 270
250 X$=MID$(MESS$,6,1)
260 IF X$="1" THEN GOTO 320
270 PRINT:INVERSE:PRINT "LOADING CYDRIVER.HEX AT C400H";:NORMAL:PRINT
280 PRGM$="MENU.BAS":SLINE=320:HFILE$="CYDRIVER.HEX"
290 CALLER$="XXXX":CLINE=999
310 CHAIN "BLOAD.BAS",10,ALL
315 CLEAR,&HC400
320 DEF USR1=&HC409:CY1=&HC409
330 DEF USR2=&HC416:CY2=&HC416
340 DEF USR3=&HC423:CY3=&HC423
350 REM
360 REM *****
370 REM * PRINT MENU
380 REM *****
390 REM
400 HOME:PRINT " MINI-MACHINE PROGRAMMING MENU":PRINT
410 PRINT" - BYU CAM LAB -":PRINT
420 PRINT"===== "
430 PRINT:PRINT:PRINT "1) TRANSLATE AN RS274 TEXT FILE"
440 PRINT:PRINT:PRINT "2) EXECUTE A TRANSLATED FILE"
450 PRINT:PRINT:PRINT "3) ZERO LATHE"
460 PRINT:PRINT:PRINT "4) ZERO MILL"
465 PRINT:PRINT "5) LINE BY LINE MANUAL EXECUTER"
470 PRINT:PRINT:PRINT "6) QUIT"
480 PRINT:PRINT:PRINT " SELECT ONE ==> ";
490 CHOICE$=INKEY$ : IF LEN(CHOICE$) = 0 THEN GOTO 490
500 VVP=VPOS(X):PRINT CHOICE$
510 IF ASC(CHOICE$)<49 OR ASC(CHOICE$)>54 THEN PRINT CHR$(7):INVERSE:PRINT "CHOO
SE 1,2,3,4,5, OR 6 ONLY";:NORMAL:VTAB (VVP):HTAB (21
) : GOTO 490
520 PRINT:PRINT" "
530 REM

```

APPENDIX J

Various CNC Programs for the Lathe and Milling Machine

2) INTERPRETING THE COMMAND FILE

To interpret and run the command file written with ED.COM programs written in MBASIC are used. MBASIC must be loaded into memory from the CP/M operating system. To load MBASIC type 'MBASIC /M:&HC400'.

Once MBASIC is loaded into memory all other necessary programs can be run from the master program called 'MENU.BAS'. To run this program from MBASIC type 'RUN "MENU"'.

To interpret a file select option '1' from the menu and follow the instructions.

3) EXECUTING THE PROGRAM

To execute the interpreted program select '2' on the menu and follow the instructions. Remember to turn on the 'blue box' first. Moving the axis' to their zero positions is accomplished by selecting the appropriate option from the menu. To return to the CP/M system from MBASIC type 'SYSTEM'.

displaying a header and a prompt for motor 'A' which indicates the motor labeled #1 on the blue box. All commands entered will be sent to motor #1 when <return> is pushed. (A list of the commands that are recognized by the blue box appears in the CY512 documentation booklet.) Multiple commands may be entered on one line by separating each command with a comma. The following is an example of a command line that sets the rate, slope, factor, number of steps, direction, and finally tells the motor to 'GO':

ADR 250,S 1,F 1,N 1000,+,G

To address a different motor type a slash (/) followed by the motor symbol. (Motor symbols are as follows: 'A'=1, 'B'=2, 'C'=3, '2A'=4, '2B'=5, and '2C'=6) To exit the program type '/Q'.

B. PROGRAM CONTROL

Programming a sequence of moves into the mini-machines involves three steps. These include creating a text file containing the mini-machine instructions using the CP/M system editor (ED.COM), interpreting the text file to 'blue box' compatible commands using the MBASIC INTERP.BAS program, and running the program using the MBASIC CYEXEC.BAS program.

1) CREATING THE N/C MACHINE CODE FILE

NOTE: Some knowledge of the CP/M system editor ED.COM is necessary to use the mini-machines under program control. The use of ED.COM is detailed in the Microsoft Softcard Manual, Vol 1.

The first step in programming the mini-machines is to enter the program as a text file on the disk. The file may have any name, but it should have the extension .RSC. Command lines consist of the line number, X,Y, and Z distances, and the Feedrate. All command lines must end with a dollar sign (\$). After the dollar sign any comments may be inserted. The file must end with a line containing a single dollar sign and nothing else. Details of each of the parameters of the command line are listed below:

LINE # - must contain 'N' followed by 3 digits. (Ex. N005)
 X - contains an 'X' followed by distance in inches (Ex. X-10.125)
 Y - contains a 'Y' followed by Y-distance in inches (Ex. Y+2.35)
 Z - contains a 'Z' followed by Z-distance in inches (Ex. Z-1.00)
 F - contains a 'F' followed by feedrate in in./min. (Ex. F55.25)

Spacing between parameters is not critical and can be used to make the program listing more readable. Each parameter does have a maximum number of digits a list of which follows:

Nxxx - 3 digits must be included in the line number parameter
xxx.xxx - maximum of 2 digits to left and 3 to right of decimal
Yxx.xxx - same as above
Zxx.xxx - same as above
Fxxx.xx - maximum of 3 digits to left and 2 to right of decimal

Following is an example of a sample program:

N001	X1.00	Y0.00	Z0.00	F100	\$ X POS. 1 IN. AT 100 IN/MIN
N002	X0.00	Y5	Z0.00	F50.5	\$ Y POS. 5 IN. AT 50.5 IN/MIN
N003	X0.00	Y0.00	Z12.125	F4	\$ Z POS. 12.125 INCHES
N004	X-1.00	Y-5	Z-12.125	F100	\$ BACK TO START
	\$				

3) THE LIMIT SWITCHES:

The limit switches are connected to the row of white sockets labeled 'switches' on the motor control box. A four conductor cable connects box with the actual switches. Each of the white switch sockets should have pins 4 and 6 wired together. The limit switch corresponding a given axis should be wired to pin 2 of the white socket. Pin 4 of the white socket is a +5 Volt output and is wired to the common terminal on all the limit switches. Following is a listing of cable wire colors with their corresponding axis' and socket pins:

<u>Cable Wire Color</u>	<u>Axis #</u>	<u>'Blue Box' Pin #</u>	<u>Mini-Machine Pin #</u>
Red	1	2, Socket 1	1
Green	2	2, Socket 2	2
White	3	2, Socket 3	3
Black	Common	4, Any Socket	6

C. THE STEPPER MOTORS:

The stepper motors connect to the stepper motor controller box with cables and plugs/sockets. See part B.2 above wiring details.

II. SOFTWARE

A number of programs have been written for mini-machine and stepper motor control. A list of these programs and some useful utility programs needed follows:

<u>Program Name</u>	<u>Purpose</u>
ED.COM	CP/M Utility to create text files(see Softcard V1)
SMARTCY.COM	Manual stepper motor control program by M. Smart
MBASIC.COM	Microsoft Basic
FIP.COM	CP/M file transfer utility(see Softcard Manual V1)
MENU.BAS	Master menu for use with machine control programs
LOADASM.BAS	Loads in necessary assembly language subroutines
BLOAD.BAS	Loads machine language from .HEX files
INTERP.BAS	Interprets NC coded text files to run mini-machines
CYEXEC.BAS	Executes interpreted machine control files
MZERO.BAS	Zero's all axis' on the mini-mill
LZERO.BAS	Zero's all axis' on the mini-lathe
CYDRIVER.HEX	Machine language routine to communicate with 'blue box'
COMM.HEX	Machine language routine to use serial comm. card

III. RUNNING THE MACHINES:

There are two ways of running the mini-machines. Manual control refers to giving commands directly to the stepper motors one at a time. Program control refers to executing stepper motors commands directly from program control.

A. MANUAL CONTROL:

To manually control the individual motors, get into the CP/M system environment (by booting up from disk or by typing 'SYSTEM' from MBASIC) and type 'SMARTCY'. The computer will respond by

MINI-LAB LATHE & MILL OPERATING INSTRUCTIONS

I. SET-UP

The mini machine control system consists of three systems or stages that must all be connected together properly for operation of the mini-machines from the APPLE computer. These are the APPLE computer itself, the stepper motor control box or 'blue box', and the stepper motors.

A. THE APPLE COMPUTER:

The APPLE computer must include some peripherals which plug into the slots at the rear of the computer. In some cases slot placement is critical. Following is a list of peripheral cards and the slots they should be plugged into:

SLOT 0 - Language RAM Card
SLOT 1 - Printer (if used)
SLOT 2 - Serial RS232 Card (if used)
SLOT 3 -
SLOT 4 - Floppy Disk Drives (w/ CORVUS)
SLOT 5 - Z-80 Microsoft Softcard
SLOT 6 - CORVUS or Floppy Disk Drives
SLOT 7 - 'Blue Box' Apple Bus Expander

Care should be taken to install the cards properly into their correct places.

B. THE SIX AXIS STEPPER MOTOR CONTROL BOX

The stepper motor control box or 'blue box' as it is commonly called has three important connection areas. These are the parallel port that goes to the APPLE, the motor connection sockets, and the limit switch connection sockets.

1) PARALLEL COMPUTER INTERFACE:

The parallel port on the 'blue box' is located at the bottom center position. This must be connected with a ribbon cable to the apple bus expander card which should be located in slot 7 of the APPLE.

2) THE STEPPER MOTORS:

The six stepper motor connection sockets are located on the bottom row of the 'blue box' and numbered 1 - 6 starting on the right hand side. These must be connected to the stepper motors themselves with six conductor cables. The standard color coding of the cables and the motor wires, along with their corresponding pins in the plugs and sockets connecting them follows:

<u>Socket pin #</u>	<u>Cable Wire Color</u>	<u>Motor Wire Color</u>	<u>Function</u>
1	Red	Red	Motor Phase #1
2	Black	Black	R-Series
3	Orange	Red/White	Motor Phase #2
4	Blue	Green/White	Motor Phase #4
5	Green	Green	Motor Phase #3
6	White	White	R-Series

APPENDIX I

Operating Instructions for the Miniature Lathe and Milling Machine

121:	C575 FEFF	CPI	OFFH	#SEE IF IT'S OVERFLOWING
122:	C577 CA7DC5	JZ	RDONE	#IF SO, GOTO RDONE
123:	C57A F25BC5	JP	WAIT2	#LOOP BACK FOR NEXT CHAR
124:	C57D 21A0E0	RDONE: LXI	H,STAT	#LOAD HL WITH STAT ADDRESS
125:	C580 3655	MVI	M,055H	#CLEAR RS-232 CTS LINE (DISABLE RECIEVE
126:	C582 05	DCR	B	#DECREMENT CHAR. CNTR TO TRUE VALUE
127:	C583 78	MOV	A,B	#PUT STRING LENGTH IN A
128:	C584 2A00C5	LHLD	DESC	#LOAD HL WITH STR DESC ADDRESS
129:	C587 77	MOV	M,A	#SAVE STRING LENGTH TO 1ST DESC. BYTE
130:	C588 23	INX	H	#INCREMENT HL TO 2ND BYTE OF DESCRIPTOR
131:	C589 EB	XCHG		#CHANGE DE WITH HL
132:	C58A 2A02C5	LHLD	STRING	#PUT STRING ADDRESS IN HL
133:	C59D EB	XCHG		#CHANGE DE WITH HL
134:	C58E 73	MOV	M,E	#SAVE L.O.B. OF STR ADDRESS TO STR DESC
135:	C58F 23	INX	H	#INCREMENT HL TO 3RD BYTE OF DESCRIPTOR
136:	C590 72	MOV	M,D	#SAVE H.O.B. OF STR ADDRESS TO STR DESC
137:	C591 2A00C5	LHLD	DESC	#LOAD HL WITH DESCRIPTOR ADDRESS
138:	C594 EB	XCHG		#PUT DESC ADDRESS IN DE
139:	C595 3E03	MVI	A,03H	#PUT STRING CODE IN ACCUMULATOR
140:	C597 C39AC5	JMP	DONE	#GOTO DONE
141:				
142:				*****
143:				* DONE
144:				*****
145:				
146:	C59A E1	DONE: POP	H	#PULL OLD SP OFF NEW STACK
147:	C59B F9	SPHL		#RESTORE OLD STACK
148:	C59C C9	RET		#RETURN TO MBASIC
149:	C4F1	ORG	0C4F1H	#MOVE LOCATION COUNTER
150:	C4F1 56414C4944	DB	'VALID'	#TELL BASIC THAT PROGRAM IS LOADED
151:	C4F7	ORG	0C4F7H	#MOVE LOCATION COUNTER AGAIN
152:	C4F7 31	DB	'1'	
153:	C4F8	END		

```

61: 051E 6F      MOV    L,A          ;LOAD L REG WITH L.O.B. OF STR ADDRESS
62: 051F 13      INC    DE          ;INC DE TO POINT TO STR ADDRESS H.O.P.
63: 0520 1A      LDAX   D          ;LOAD A WITH H.O.B. OF STR ADDRESS
64: 0521 67      MOV    H,A          ;LOAD H REG WITH H.O.B.
65: 0522 220C5    SHLD   STRING      ;SAVE STRING ADDRESS
66: 0525 78      MOV    A,B          ;MOVE STRING LENGTH TO A
67: 0526 EB      XCHG             ;PUT STR ADDRESS IN DE
68: 0527 C1      POP     B          ;POP INIT RETURN ADDRESS OFF OLD STACK
69: 0528 21000H   LXI    H,0000H      ;CLEAR HL
70: 052B 39      DAD     SP          ;ADD STACK POINTER (SP) TO HL
71: 052C 3100C6   LXI    SP,0C600H    ;INIT NEW STACK POINTER AT C600H
72: 052F E5      PUSH   H          ;PUSH OLD SP ONTO NEW STACK
73: 0530 C5      PUSH   B          ;PUSH INIT RETURN ADDRESS ONTO NEW STACK
74: 0531 C9      RET              ;RETURN FROM SUBROUTINE
75:
76: ;
77: ;*****
78: ;* SEND ROUTINE
79: ;*****
80: ;
81: 0532 2A00C5    SEND:  LHLD   DESC          ;LOAD HL WITH STR DESCR. ADDRESS
82: 0535 4E      MOV    C,M          ;MOVE STRING LENGTH TO C
83: 0536 2A02C5    LHLD   STRING        ;LOAD HL WITH STRING STARTING ADDRESS
84: 0539 EB      XCHG             ;EXCHANGE HL AND DE
85: 053A 0601      MVI    B,01H        ;INITIALIZE B AS CHAR. CNTR
86: 053C 21A0E0    WAIT1: LXI    H,STAT      ;LOAD HL WITH ADDRESS OF STATUS
87: 053F 7E      MOV    A,M          ;LOAD A WITH STATUS
88: 0540 E602      ANI    TRDY          ;MASK OUT TRANS. READY BIT
89: 0542 FE02      CPI    TRDY          ;IS READY BIT SET?
90: 0544 C23CC5    JNZ    WAIT1        ;IF NOT, WAIT AGAIN ...
91: 0547 1A      LDAX   D          ;LOAD ACC WITH CHAR.
92: 0548 32A1E0    STA    DATA        ;SEND CHAR. OUT
93: 054B 79      MOV    A,C          ;PUT STRING LENGTH IN A
94: 054C B8      CMP     B          ;COMPARE WITH CHAR. CNTR
95: 054D CA9AC5    JZ     DONE          ;ALL CHAR.S SENT. GOTO DONE
96: 0550 13      INX     D          ;INCREMENT CHAR. POINTER
97: 0551 04      INR     B          ;INCREMENT CHAR. COUNTER
98: 0552 C33CC5    JMP     WAIT1        ;LOOP BACK FOR NEXT CHAR.
99:
100: ;
101: ;*****
102: ;* RECIEVE
103: ;*****
104: ;
105: 0555 0601      REC:    MVI    B,01H        ;INITIALIZE CHAR. COUNTER
106: 0557 2A02C5    LHLD   STRING        ;LOAD HL WITH STRING ADDRESS
107: 055A EB      XCHG             ;PUT STRING ADDRESS IN DE
108: 055B 21A0E0    WAIT2: LXI    H,STAT      ;LOAD HL WITH STAT ADDRESS
109: 055E 7E      MOV    A,M          ;LOAD ACC WITH STATUS
110: 055F E601      ANI    DRDY          ;MASK OUT DRDY BIT
111: 0561 FE01      CPI    DRDY          ;SEE IF BIT IS SET
112: 0563 C25BC5    JNZ    WAIT2        ;WAIT UNTILL BIT IS SET
113: 0566 21A1E0    GETIT: LXI    H,DATA      ;LOAD HL WITH DATA ADDRESS
114: 0569 7E      MOV    A,M          ;READ IN THE NEXT CHARACTER
115: 056A E67F      ANI    07FH        ;GET RID OF HIGH BIT
116: 056C FE0A      CPI    0AH          ;SEE IF CHAR WAS A LINE FEED
117: 056E CA7DC5    JZ     RDONE          ;IF IT WAS THEN GOTO RDONE
118: 0571 12      STAX   D          ;STORE ACC TO STRING STORAGE LOCATION
119: 0572 13      INX     D          ;INCREMENT CHAR. STORAGE POINTER
120: 0573 04      INR     B          ;INCREMENT CHAR. COUNTER
121: 0574 78      MOV    A,B          ;PUT CHAR CNTR IN A

```

```

1:  ;*****
2:  ;*
3:  ;* THIS PROGRAM IS CALLED FROM MBASIC TO SEND OR
4:  ;* RECIEVE CHARACTERS FROM THE RS232 COMMUNICATIONS
5:  ;* CARD.
6:  ;*
7:  ;* THE FORMATT FOR CALLING THE ROUTINE FROM
8:  ;* MBASIC IS AS FOLLOWS:
9:  ;*
10: ;* X$=USR1("STRING") TO SEND "STRING"
11: ;* X$=USR2 TO RECIEVE A STRING
12: ;* (DEF USR1=SSTART, DEF USR2=RSTART)
13: ;*
14: ;* PROGRAM WRITTEN BY: JUSTIN D. REDD -BYU CAM LAB-
15: ;* AUGUST 1983
16: ;*
17: ;*****
18: ;
19: C500 ORG 0C500H ;START ASSEMBLY AT LOCATION C500H
20: ;
21: ;*****
22: ;* I. INITIALIZATION
23: ;*****
24: ;
25: E0A0 = STAT EQU 0E0A0H
26: E0A1 = DATA EQU 0E0A1H ; ADDRESSES OF RS232 CARD BUFFERS
27: ;*** STAT AND DATA SET FOR APPLE SLOT#2.
28: ;*** FOR OTHER SLOTS: STAT=COX0H & DATA=COX1H X=8+SLOT#
29: ;***
30: 0001 = DRDY EQU 01H ; DATA READY CODE IN STATUS REG.
31: 0002 = TRDY EQU 02H ; TRANSMIT READY CODE IN STATUS REG.
32: C500 DESC: DS 2 ;RESERVE 2 BYTES FOR STRING DESCRIPTOR
33: C502 STRING: DS 2 ; ' ' ' FOR STRING ADDRESS
34: C504 LCNTR: DS 1 ;RESERVE 1 BYTE FOR LOOP COUNTER
35: ;
36: ;
37: ;*****
38: ;* II. MAIN PROGRAM
39: ;*****
40: ;
41: ;**** SEND OR RECIEVE ? ****
42: ;
43: C505 C016C5 SSTART: CALL INIT ;INITIALIZE SUBROUTINE
44: C508 C332C5 JMP SEND ;JUMP TO SEND ROUTINE
45: ;
46: C50B C016C5 RSTART: CALL INIT ;INITIALIZE SUBROUTINE
47: C50E 21A0E0 LXI H,STAT ;LOAD HL WITH STAT (COMMAND REG) ADDRESS
48: C511 3615 MVI M,015H ;SET RS-232 CTS LINE (ENABLE IBM TRANS)
49: C513 C355C5 JMP REC ;JUMP TO RECIEVE ROUTINE
50: ;
51: ;*****
52: ;* INITIALIZE SUBROUTINE
53: ;*****
54: ;
55: C516 47 INIT: MOV B,A ;MOVE STRING LENGTH TO B
56: C517 EB XCHG ;EXCHANGE DE AND HL
57: C518 2200C5 SHLD DESC ;SAVE STR DESCRIPTOR ADDRESS
58: C51B EB XCHG ;RESTORE DE AND HL
59: C51C 13 INX D ;INCREMENT DE TO POINT TO STR ADDRESS
60: C51D 14 LDAX D ;LOAD A WITH 1ST BYTE OF STR ADDRESS

```

121:	C48E 3A04C4	LDA	LENGTH	LOAD ACC WITH STRING LENGTH
122:	C491 3D	DCR	A	DECREMENT LENGTH
123:	C492 3204C4	STA	LENGTH	STORE DECREMENTED LENGTH
124:	C495 CACDC4	JZ	DONE	IF LENGTH = 0 THEN WE'RE DONE
125:	C498 2A00C4	LHLD	STRING	LOAD HL WITH STRING ADDRESS
126:	C49B 23	INX	H	INCREMENT STRING ADDRESS
127:	C49C 2200C4	SHLD	STRING	STORE INCREMENTED STRING VALUE
128:	C49F C387C4	JMP	MAIN	GO BACK TO BEGINNING OF MAIN
129:				
130:				*****
131:				* SEND
132:				*****
133:				
134:	C4A2 E67F	SEND:	ANI 07FH	CLEAR BIT #7
135:	C4A4 FE2C		CPI 02CH	COMPARE ACC WITH ','
136:	C4A6 C2ABC4		JNZ SCONT1	IF ACC < > ',' THEN GOTO SCONT1
137:	C4A9 3E0D		MVI A,0DH	CHANGE ACC TO 'RETURN'
138:	C4AB 2A05C4	SCONT1:	LHLD STAT	LOAD HL WITH STAT ADDRESS
139:	C4AE 47		MOV B,A	PUT CHAR IN B
140:	C4AF 7E	SCONT2:	MOV A,M	LOAD ACC WITH STAT
141:	C4B0 E604		ANI 04H	MASK OUT READY BIT
142:	C4B2 FE04		CPI 04H	COMPARE ACC WITH 04H
143:	C4B4 C2AFC4		JNZ SCONT2	GO BACK AND WAIT TILL READY = 1
144:	C4B7 2A07C4		LHLD DATA	LOAD DATA ADDRESS INTO HL
145:	C4BA 70		MOV M,B	PUT CHAR IN DATA REGISTER
146:	C4BB 3E41		MVI A,041H	LOAD ACC WITH I/O REQUEST WORD
147:	C4BD 2A05C4		LHLD STAT	LOAD HL WITH STAT ADDRESS
148:	C4C0 77		MOV M,A	SEND I/O REQUEST
149:	C4C1 7E	SCONT3:	MOV A,M	PUT STAT IN ACC
150:	C4C2 E604		ANI 04H	MASK OUT READY BIT
151:	C4C4 FE04		CPI 04H	COMPARE ACC WITH 04H
152:	C4C6 CAC1C4		JZ SCONT3	WAIT TILL READY = 0
153:	C4C9 3E40		MVI A,040H	LOAD ACC WITH I/O REQ CLEAR WORD
154:	C4CB 77		MOV M,A	SEND I/O REQ CLEAR WORD
155:	C4CC C9		RET	RETURN FROM SUBROUTINE
156:				
157:				*****
158:				* DONE
159:				*****
160:				
161:	C4CD 2A05C4	DONE:	LHLD STAT	LOAD HL WITH STAT ADDRESS
162:	C4D0 7E		MOV A,M	PUT STATUS IN ACC
163:	C4D1 E604		ANI 04H	MASK OUT READY BIT
164:	C4D3 FE04		CPI 04H	COMPARE ACC WITH 04H
165:	C4D5 C2CDC4		JNZ DONE	GO BACK AND WAIT TILL READY = 1
166:	C4D8 2A07C4		LHLD DATA	LOAD DATA ADDRESS INTO HL
167:	C4DB 360D		MVI M,0DH	SEND RETURN CHAR
168:	C4DD 3E41		MVI A,041H	LOAD ACC WITH I/O REQUEST WORD
169:	C4DF 2A05C4		LHLD STAT	LOAD HL WITH STAT ADDRESS
170:	C4E2 77		MOV M,A	SEND I/O REQUEST
171:	C4E3 7E	DCONT1:	MOV A,M	PUT STAT IN ACC
172:	C4E4 E604		ANI 04H	MASK OUT READY BIT
173:	C4E6 FE04		CPI 04H	COMPARE ACC WITH 04H
174:	C4E8 CAE3C4		JZ DCONT1	WAIT TILL READY = 0
175:	C4EB 3E40		MVI A,040H	LOAD ACC WITH I/O REQ WORD
176:	C4ED 77		MOV M,A	SEND I/O REQ CLEAR WORD
177:	C4EE E1		POP H	POP OLD STACK POINTER OFF NEW STACK
178:	C4EF F9		SPHL	RESTORE OLD STACK
179:	C4F0 C9		RET	
180:	C4F1 56414C4944	DB	'VALID1'	MESSAGE TO TELL BASIC PROGRAM IS LOA

```

61: C437 CD74C4      CALL  SETUP      ;SETUP STAT & DATA REG ADDRESSES
62: C43A C387C4      JMP    MAIN        ;JUMP TO MAIN PROGRAM
63: C43D CD57C4      CY5:  CALL  INIT        ;INITIALIZE
64: C440 0609        MVI    B,09H      ;OFFSET STAT REG VALUE FOR CY#5
65: C442 0E09        MVI    C,09H      ;OFFSET DATA REG VALUE FOR CY#5
66: C444 CD74C4      CALL  SETUP      ;SETUP STAT & DATA REG ADDRESSES
67: C447 C387C4      JMP    MAIN        ;JUMP TO MAIN PROGRAM
68: C44A CD57C4      CY6:  CALL  INIT        ;INITIALIZE
69: C44D 060A        MVI    B,0AH      ;OFFSET STAT REG VALUE FOR CY#6
70: C44F 0E0A        MVI    C,0AH      ;OFFSET DATA REG VALUE FOR CY#6
71: C451 CD74C4      CALL  SETUP      ;SETUP STAT & DATA REG ADDRESSES
72: C454 C387C4      JMP    MAIN        ;JUMP TO MAIN PROGRAM
73:                  ;
74:                  ;*****
75:                  ;* INITIALIZE SUBROUTINE
76:                  ;*****
77:                  ;
78: C457 EB          INIT:  XCHG          ;SWITCH DE AND HL
79: C458 2202C4      SHLD   DESC        ;SAVE STRING DESCRIPTOR ADDRESS
80: C45B 7E          MOV    A,M        ;PUT STRING LENGTH IN ACC
81: C45C 3204C4      STA    LENGTH      ;STORE STRING LENGTH
82: C45F EB          XCHG          ;RESTORE REGISTERS
83: C460 13          INX    D          ;INCREMENT DE TO POINT TO STRING ADDR
84: C461 1A          LDAX   D          ;LOAD ACC WITH L.O.B. OF STR ADDRESS
85: C462 6F          MOV    L,A        ;LOAD L.O.B. OF STR ADDRESS INTO L
86: C463 13          INX    D          ;INCREMENT DE TO POINT TO H.O.B
87: C464 1A          LDAX   D          ;LOAD ACC WITH H.O.B. OF STR ADDRESS
88: C465 67          MOV    H,A        ;LOAD L.O.B. OF STR ADDRESS INTO H
89: C466 2200C4      SHLD   STRING      ;SAVE STRING ADDRESS
90: C469 C1          POP    B          ;POP INIT RETURN ADDRESS OFF OLD STACK
91: C46A 210000      LXI    H,0000H      ;CLEAR HL TO 0000
92: C46D 39          DAD    SP        ;ADD STACK POINTER (SP) TO HL
93: C46E 3100C6      LXI    SP,0C600H      ;SET STACK POINTER TO NEW STACK AT C:
94: C471 E5          PUSH   H          ;PUSH OLD STACK LOCATION ONTO NEW STACK
95: C472 C5          PUSH   B          ;PUSH INIT RETURN ADDRESS ONTO NEW STACK
96: C473 C9          RET            ;RETURN FROM SUBROUTINE
97:                  ;
98:                  ;*****
99:                  ;* SETUP DATA & CNTRL ADDRESSES
100:                 ;*****
101:                 ;
102: C474 21F0E0      SETUP: LXI    H,STRBASE ;LOAD HL WITH STAT BASE ADDRESS
103: C477 78          MOV    A,B        ;PUT STAT OFFSET FOR CY# IN ACC
104: C478 85          ADD    L          ;ADD L.O.B. OF STAT BASE ADDRESS
105: C479 6F          MOV    L,A        ;PUT ADJUSTED L.O.B. BACK IN L
106: C47A 2205C4      SHLD   STAT        ;SAVE STAT REG ADDRESS TO STAT
107: C47D 21F4E0      LXI    H,DTBASE      ;LOAD HL WITH DATA BASE ADDRESS
108: C480 79          MOV    A,C        ;PUT DATA OFFSET FOR CY# IN ACC
109: C481 85          ADD    L          ;ADD L.O.B. OF DATA BASE ADDRESS
110: C482 6F          MOV    L,A        ;PUT ADJUSTED L.O.B. BACK IN L
111: C483 2207C4      SHLD   DATA      ;SAVE DATA REG ADDRESS TO DATA
112: C486 C9          RET            ;RETURN FROM SUBROUTINE
113:                 ;
114:                 ;*****
115:                 ;* MAIN PROGRAM
116:                 ;*****
117:                 ;
118: C487 2A00C4      MAIN:  LHLD   STRING      ;LOAD HL WITH STRING ADDRESS
119: C48A 7E          MOV    A,M        ;LOAD ACC WITH FIRST CHAR
120: C48B CDA2C4      CALL  SEND        ;SEND CHARACTER TO CY'S

```

```

560 REM * LOAD CYDRIVER.HEX ONLY
570 REM *****
580 REM
590 PRINT:INVERSE:PRINT "LOADING CYDRIVER.HEX AT C400H";:NORMAL:PRINT
600 PRGM$="B:LOADASM.BAS":SLINE=620:HFILE$="B:CYDRIVER.HEX"
610 CHAIN "B:BLOAD.BAS",,ALL
620 GOTO 730
630 REM
640 REM *****
650 REM * LOAD COMM.HEX ONLY
660 REM *****
670 REM
680 PRINT:INVERSE:PRINT "LOADING COMM.HEX AT C500H";:NORMAL:PRINT
690 PRGM$="B:LOADASM.BAS":SLINE=710:HFILE$="B:COMM.HEX"
700 CHAIN "B:BLOAD.BAS"
710 GOTO 730
720 REM
730 REM *****
740 REM * CHAIN BACK
750 REM *****
760 REM
820 DEF USR1=&HC409
830 DEF USR2=&HC416
840 DEF USR3=&HC423
850 DEF USR4=&HC430
860 DEF USR5=&HC43D
870 DEF USR6=&HC44A
880 DEF USR7=&HC505
890 DEF USR8=&HC50B
900 CHAIN CALLER$,CLINE,ALL
910 END

```

```

10 REM *****
20 REM *
30 REM *      BLOAD.BAS
40 REM *      BY
50 REM *      JUSTIN D. REDD      - BYU CAM LAB
60 REM *
70 REM * PROGRAM TO LOAD A .HEX FILE
80 REM * INTO MEMORY FROM MBASIC
90 REM *
100 REM * OPERATION: THIS PROGRAM WAS WRITTEN
110 REM * TO BE CHAINED TO, USING THE 'COMMON'
120 REM * STATEMENT TO DECLARE THE FOLLOWING
130 REM * VARIABLES: PRGM$=PROGRAM TO CHAIN
140 REM *      TO AFTER LOADING .HEX FILE
150 REM *      SLINE=STARTING LINE IN PRGM$
160 REM *      HFILE$=.HEX FILE NAME
170 REM *
180 REM *****
190 REM
200 COMMON PRGM$,HFILE$,SLINE,CLINE,CALLER$
210 REM
220 REM *****
230 REM *      INITIALIZE      *
240 REM *****
250 REM
260 OPEN "I",#1,HFILE$
270 REM
280 REM *****
290 REM *      MAIN PROGRAM      *
300 REM *****
310 REM
320 GOSUB 360
330 GOSUB 470
340 GOTO 320
350 REM
360 REM *****
370 REM *      READ LINE S/R      *
380 REM *****
390 REM
400 INPUT #1,HLINE$
410 HBYTES$="&H" + MID$(HLINE$,2,2)
420 HADDRESS$="&H" + MID$(HLINE$,4,4)
430 BYTES=VAL(HBYTES$):ADDRESS=VAL(HADDRESS$)
440 IF BYTES=0 AND ADDRESS=0 THEN HOME:CLOSE:PRINT "PROGRAM LOADED":CHAIN PRGM$.
    SLINE,ALL
450 RETURN
460 REM
470 REM *****
480 REM *      POKE BYTES S/R      *
490 REM *****
500 FOR I = 1 TO BYTES
510 LOCATION=ADDRESS + (I - 1)
520 CODE$="&H" + MID$(HLINE$,(8 + (I * 2)),2)
530 CODE=VAL(CODE$)
540 POKE LOCATION,CODE
550 NEXT I

```

560 RETURN

```

10 REM *****
20 REM *
30 REM *      INTERP.BAS      BY JUSTIN D. REDD
40 REM *      - BYU CAM LAB -
50 REM *      CNC - CY512 TRANSLATOR      NOVEMBER, 1983
60 REM *
70 REM *      THIS PROGRAM READS RS274 STANDARD NUMERICAL CONTROL
80 REM *      CODES FROM A TEXT FILE AND TRANSLATES THEM TO
90 REM *      COMMANDS FOR THE CY512 STEPPER MOTER CONTROLLER.
100 REM*      THE RESULTING CY512 COMMANDS ARE STORED IN A TEXT
110 REM*      FILE OF THE SAME NAME WITH THE SUFFIX OF *.CYC
120 REM*
130 REM*****
140 REM
145 COMMON MENU$
146 REM
147 REM
150 HOME:PRINT "RS274 - CY512 INTERPRETER"      - BYU CAM LAB -"
160 PRINT:PRINT"-----
      ":PRINT:PRINT:PRINT
170 REM
180 REM *****
190 REM * OPEN FILE AND DIMENTION ARRAYS
200 REM *****
210 REM
220 INPUT "INTERPRET WHAT RS274 FILE ";RS274$
230 TEST$=RIGHT$ ( RS274$,4 )
240 IF LEFT$( TEST$,1 ) = "." THEN GOTO 255
250 RS274$ = RS274$ + ".RSC"
255 IF MID$(RS274$,2,1) = ":" THEN LLL = 14 ELSE LLL = 12
260 IF LEN(RS274$)>LLL THEN PRINT:PRINT "FILENAME TOO LONG!":PRINT:GOTO 220
265 ON ERROR GOTO 2181
270 OPEN"I",#1,RS274$
280 F=INSTR(RS274$,".")
290 CY512$=MID$(RS274$,1,F-1) + ".CYC"
300 OPEN"O",#2,CY512$
305 ON ERROR GOTO 0
310 DIM CYMOVE$(100)
320 DIM SLINE$(100)
330 NLINE = 0
340 REM *****
350 REM * INPUT LINES AND SEPARATE
360 REM *****
370 REM
380 FOR LINES = 1 TO 100
390 INPUT #1,SLINE$(LINES)
400 IF SLINE$(LINES) = "$" THEN LINES = LINES + 1 : GOTO 420
410 NEXT LINES
420 FOR I = 1 TO (LINES - 1)
430 XFLAG = 0 : YFLAG = 0 : ZFLAG = 0
440 NLINE = NLINE + 1
450 PRINT ".";
460 SLINE$=SLINE$(I)
470 N=INSTR(SLINE$,"N") : X=INSTR(SLINE$,"X") : Y=INSTR(SLINE$,"Y")
480 Z=INSTR(SLINE$,"Z") : F=INSTR(SLINE$,"F") : ENDLINE=INSTR(SLINE$,"$")
490 IF N=0 AND X=0 AND Y=0 AND Z=0 AND F=0 AND ENDLINE <> 0 THEN GOTO 2200

```

```

500 IF N=0 OR F=0 OR ENDLINE=0 THEN GOTO 1800
510 IF X=0 THEN X=CINT((N+Y)/2) : XFLAG = 1
520 IF Z=0 THEN Z=CINT((Y+F)/2) : ZFLAG = 1
530 IF Y=0 THEN Y=CINT((X+Z)/2) : YFLAG = 1
540 IF N>X OR N>Y OR N>Z OR N>F OR N>ENDLINE THEN GOTO 1670
550 IF X>Y OR X>Z OR X>F OR X>ENDLINE THEN GOTO 1670
560 IF Y>Z OR Y>F OR Y>ENDLINE THEN GOTO 1670
570 IF Z>F OR Z>ENDLINE THEN GOTO 1670
580 IF F>ENDLINE THEN GOTO 1670
590 REM
600 REM ***** SEPARATE PARAMETERS *****
610 REM
620 IF XFLAG = 1 THEN X$="00.000" : GOTO 680
630 IF YFLAG = 0 THEN X$ = MID$(SLINE$,X+1,Y-(X+1)): GOTO 660
640 IF YFLAG = 1 AND ZFLAG = 0 THEN X$=MID$(SLINE$,X+1,Z-(X+1)) : GOTO 660
650 IF YFLAG = 1 AND ZFLAG = 1 THEN X$=MID$(SLINE$,X+1,F-(X+1))
660 PARAMETER$=X$ : GOSUB 810
670 X$=PARAMETER$
680 IF YFLAG = 1 THEN Y$="00.000" : GOTO 730
690 IF ZFLAG = 0 THEN Y$=MID$(SLINE$,Y+1,Z-(Y+1)): GOTO 710
700 IF ZFLAG = 1 THEN Y$=MID$(SLINE$,Y+1,F-(Y+1))
710 PARAMETER$ = Y$ : GOSUB 810
720 Y$=PARAMETER$
730 IF ZFLAG = 1 THEN Z$ = "00.000" ELSE Z$ = MID$(SLINE$,Z+1,F-(Z+1))
740 PARAMETER$ = Z$ : GOSUB 810
750 Z$=PARAMETER$
760 F$ = MID$(SLINE$,F+1,ENDLINE - (F+1))
770 PARAMETER$= F$ : GOSUB 810
780 F$ = PARAMETER$
790 GOTO 870
800 REM
810 REM ***** SUBROUTINE TO DELETE SPACES *****
820 REM
830 SPACE = INSTR ( PARAMETER$," " ) : LENGTH = LEN (PARAMETER$)
840 IF SPACE = 0 THEN RETURN
850 PARAMETER$ = LEFT$(PARAMETER$,SPACE-1) + RIGHT$(PARAMETER$,LENGTH - SPACE):G
OTO 830
860 REM
870 IF LEFT$(X$,1) <> "+" AND LEFT$(X$,1) <> "-" THEN X$="+" + X$
880 IF LEFT$(Y$,1) <> "+" AND LEFT$(Y$,1) <> "-" THEN Y$="+" + Y$
890 IF LEFT$(Z$,1) <> "+" AND LEFT$(Z$,1) <> "-" THEN Z$="+" + Z$
900 IF LEN(X$) > 7 OR LEN(Y$) > 7 OR LEN(Z$) > 7 OR LEN (F$) > 6 THEN GOTO 1730
910 XPNT=INSTR(X$,".") : YPNT=INSTR(Y$,".") : ZPNT=INSTR(Z$,".") : FPNT=INSTR(F$
,".")
920 IF XPNT>4 OR YPNT>4 OR ZPNT>4 OR FPNT>4 THEN GOTO 1730
930 REM
940 REM *****
950 REM * INTERPOLATE MOVE
960 REM *****
970 REM
980 XDIST=VAL(X$) : YDIST=VAL(Y$) : ZDIST=VAL(Z$) : FEED=VAL(F$)
990 TLDIST=SQR( (XDIST ^ 2) + (YDIST ^ 2) + (ZDIST ^ 2) )
1000 TIME=TLDIST/FEED
1010 XSPEED=ABS(XDIST/TIME) : YSPEED=ABS(YDIST/TIME) : ZSPEED=ABS(ZDIST/TIME)
1020 IF (XSPEED < 1.65 AND XSPEED > 0) OR (YSPEED < 1.65 AND YSPEED > 0) OR (ZSP
EED < 1.65 AND ZSPEED > 0) THEN GOSUB 1860

```

```

1030 IF XSPEED > 112 OR YSPEED > 112 OR ZSPEED > 112 THEN GOSUB 2030
1040 DISTANCE= (-1) * XDIST : SPEED=XSPEED
1050 GOSUB 1240
1060 XCY$=CY$
1070 DISTANCE= (-1) * YDIST : SPEED=YSPEED
1080 GOSUB 1240
1090 YCY$=CY$
1100 DISTANCE=ZDIST : SPEED=ZSPEED
1110 GOSUB 1240
1120 ZCY$=CY$
1130 REM
1140 REM *****
1150 REM * SAVE MOVE TO CY FILE
1160 REM *****
1170 REM
1180 CYMOVE$(I)=CHR$(34)+XCYS + " : " + YCY$ + " : " + ZCY$+CHR$(34)
1190 NEXT I
1200 FOR LINES = 1 TO 100
1210 PRINT #2,CYMOVE$(LINES)
1220 NEXT LINES
1230 GOTO 370
1240 REM *****
1250 REM * ROUTINE TO CONVERT DATA TO CY512 COMMANDS
1260 REM *****
1270 REM
1275 IF DISTANCE = 0 THEN CY$="I,N 0" : RETURN
1280 IF DISTANCE < 0 THEN DIRECTION$="-"
1290 IF DISTANCE >= 0 THEN DIRECTION$="+"
1300 STEPS = CINT(ABS(DISTANCE * 1600))
1310 REM *** CALCULATE RATE AND FACTOR FOR CY512 ***
1320 SS = SPEED * 26.667
1330 IF SS < 43.5 THEN SS = 43.5
1340 IF SS > 3000 THEN SS = 3000
1350 RATE = CINT(257.938 - (12500! / SS))
1360 IF RATE < 1 THEN RATE = 1
1370 IF RATE > 253 THEN RATE = 253
1380 FACTOR = CINT ((8 * RATE) + (100000! / SS) - 2055.5)
1390 IF FACTOR < 1 THEN RATE = RATE + 1:GOTO 1380
1400 REM *****
1410 REM * CHECK TO SEE IF R, AND F ARE BEST
1420 REM *****
1430 FMIN = FACTOR - 1:F = FACTOR:FPLUS = FACTOR + 1
1440 FEXP=FMIN:GOSUB 1530
1450 ERRMIN = SSERR
1460 FEXP=FPLUS:GOSUB 1530
1470 ERRPLUS = SSERR
1480 FEXP = F:GOSUB 1530
1490 IF ERRMIN >= SSERR AND ERRPLUS >= SSERR THEN GOTO 1580
1500 IF ERRMIN < SSERR THEN FACTOR = FACTOR - 1:GOTO 1420
1510 IF ERRPLUS < SSERR THEN FACTOR = FACTOR + 1:GOTO 1420
1520 PRINT:PRINT "ERROR IN R AND F CHECK ROUTINE":END
1530 REM ***** CALCULATE SS FROM R AND F VALUES *****
1540 DENOM= (((256 - RATE) * 80) + (FEXP * 10) + 75) * .000001
1550 SSEXP= 1/DENOM
1560 SSERR=ABS(SS-SSEXP)
1570 RETURN

```

```

1580 REM ** CALCULATE SLOPE **
1585 IF SPEED < 12 THEN SLOPE = 255 : GOTO 1630
1590 SLOPE = 1
1600 YY = 2 * ( INT(( 255 - FACTOR ) / SLOPE ))
1610 IF SLOPE > 254 THEN GOTO 1630
1620 IF YY > STEPS THEN SLOPE = SLOPE + 1 : GOTO 1600
1630 RATE$=STR$(RATE):SLOPE$=STR$(SLOPE):FACTOR$=STR$(FACTOR):STEPS$=STR$(STEPS)
1640 CY$="R"+RATE$+",S"+SLOPE$+",F"+FACTOR$+",N"+STEPS$+", "+DIRECTION$
1650 RETURN
1660 REM
1670 REM *****
1680 REM * LINE OUT OF ORDER ERROR
1690 REM *****
1700 REM
1705 PRINT CHR$(7)
1710 PRINT:PRINT "PARAMETERS OUT OF ORDER IN LINE# ";NLINE
1720 PRINT:END
1730 REM *****
1740 REM * NUMERICAL VALUE ERROR
1750 REM *****
1760 REM
1765 PRINT CHR$(7)
1770 PRINT:PRINT "NUMERICAL VALUE ERROR IN LINE# ";NLINE
1780 PRINT "TOO MANY DIGITS OR DECIMAL POINT MISPLACED"
1790 PRINT:END
1800 REM *****
1810 REM * PARAMETER MISSING ERROR
1820 REM *****
1830 REM
1835 PRINT CHR$(7)
1840 PRINT:PRINT "N,F, OR $ PARAMETER MISSING IN LINE# ";NLINE
1850 PRINT:END
1860 REM *****
1870 REM * AXIS SPEED < 1.65 ERROR
1880 REM *****
1890 REM
1895 AXIS$=" "
1900 IF XSPEED < 1.65 AND XSPEED > 0 THEN AXIS$="X-AXIS"
1910 IF YSPEED < 1.65 AND YSPEED > 0 THEN AXIS$=AXIS$+",Y-AXIS"
1920 IF ZSPEED < 1.65 AND ZSPEED > 0 THEN AXIS$=AXIS$+",Z-AXIS"
1930 PRINT CHR$(7)
1940 PRINT:PRINT "FEED RATE TOO SLOW IN LINE# ";NLINE
1950 PRINT "INTERPOLATION CAUSES ";AXIS$;" TO BE TOO SLOW"
1960 PRINT "( ";AXIS$;" SPEED WILL BE ROUNDED UP )"
1970 PRINT:PRINT "PUSH <RETURN> TO CONTINUE OR <ESCAPE> TO ABORT"
1980 Q$=INKEY$:IF LEN(Q$) = 0 THEN GOTO 1980
1990 IF Q$=CHR$(27) THEN PRINT:END
2000 IF Q$ <> CHR$(13) THEN PRINT CHR$(7):GOTO 1970
2010 PRINT:PRINT "-----":RETURN
2020 REM
2030 REM *****
2040 REM * AXIS SPEED > 112 ERROR
2050 REM *****
2060 REM
2070 IF XSPEED > 112 AND ZSPEED > 112 THEN AXIS$= "X AND Z-AXIS": GOTO 2110
2080 IF XSPEED > 112 THEN AXIS$= "X-AXIS"

```

```

2090 IF ZSPEED > 112 THEN AXIS$= "Z-AXIS"
2100 PRINT CHR$(7)
2110 PRINT:PRINT "FEED RATE TOO FAST IN LINE# ";NLINE
2120 PRINT "INTERPOLATION CAUSES ";AXIS$;" TO BE TOO FAST"
2130 PRINT "( ";AXIS$;" SPEED WILL BE ROUNDED DOWN )"
2140 PRINT:PRINT "PUSH <RETURN> TO CONTINUE OR <ESCAPE> TO ABORT"
2150 Q$=INKEY$:IF LEN(Q$)=0 THEN GOTO 2150
2160 IF Q$=CHR$(27) THEN PRINT:END
2170 IF Q$ <> CHR$(13) THEN PRINT CHR$(7):GOTO 2140
2180 PRINT:PRINT "-----":RETURN
2181 REM
2182 REM *****
2183 REM * ROUTINE TO HANDLE 'NO FILE' ERRORS
2184 REM *****
2185 REM
2186 IF ERR = 53 OR ERR = 64 THEN PRINT:PRINT "ERROR: FILE NOT FOUND OR BAD FILE
NAME!":PRINT:INPUT "PUSH <RETURN> TO CONTINUE",XXX$
:GOTO 2310
2187 ON ERROR GOTO 0
2189 REM
2190 REM
2200 REM *****
2210 REM * END OF OPERATION
2220 REM *****
2230 REM
2240 FOR LINES = 1 TO (I-1)
2250 PRINT #2,CYMOVE$(LINES)
2260 NEXT LINES
2270 PRINT #2,"$$$$$"
2280 CLOSE
2290 PRINT CHR$(7):REM:PRINT CHR$(7):REM:PRINT CHR$(7)
2300 PRINT:PRINT:PRINT RS274$;"==>";CY512$;" TRANSLATION COMPLETED"
2310 PRINT:CHAIN MENU$
2320 END

```

```

10 REM *****
20 REM *
30 REM * CYEXEC.BAS          BY JUSTIN D. REDD   - BYU CAM LAB -
40 REM *                                     DECEMBER, 1983
50 REM *
60 REM * THIS PROGRAM READS IN A FILE WITH THE FILENAME EXTENTION
70 REM * .CYC THAT HAS BEEN CREATED BY THE INTERP.BAS PROGRAM
80 REM * FROM AN RS274 TEXT FILE. IT THEN EXECUTES THE CY512
90 REM * COMMANDS CONTAINED IN THE FILE AND TERMINATES.
100 REM*
110 REM*****
120 REM
130 REM *****
140 REM * CHECK FOR CYDRIVER.HEX IN MEMORY
150 REM *****
160 REM
170 PRINT:INVERSE:PRINT "CHECKING FOR CYDRIVER.HEX";:NORMAL:PRINT
180 MESS$=""
190 FOR CHAR = 0 TO 6
200 LOCATION = &HC4F1 + CHAR
210 CHAR$=CHR$(PEEK(LOCATION))
220 MESS$=MESS$ + CHAR$:CHAR$= ""
230 NEXT CHAR
240 IF LEFT$(MESS$,5) <> "VALID" THEN GOTO 270
250 X$=MID$(MESS$,6,1)
260 IF X$="1" THEN GOTO 360
270 PRINT:INVERSE:PRINT "LOADING CYDRIVER.HEX AT C400H";:NORMAL:PRINT
280 PRGM$="CYEXEC.BAS":SLINE=330:HFILE$="CYDRIVER.HEX"
290 CALLER$="XXXX":CLINE=999
300 CHAIN "BLOAD.BAS",10,ALL
310 CLEAR,&HC400
320 DEF USR1=&HC409
330 DEF USR2=&HC416
340 DEF USR3=&HC423
350 REM
360 REM *****
370 REM * READ IN CY512 FILE
380 REM *****
390 REM
400 HOME : PRINT "CY512 FILE EXECUTER"                - BYU CAM LAB -
410 PRINT:PRINT "-----"
420 INPUT "EXECUTE WHAT CY512 COMMAND FILE ";CYFILE$
430 TEST$=RIGHT$ (CYFILE$,4)
440 IF LEFT$ (TEST$,1) = "." THEN GOTO 465
450 CYFILE$ = CYFILE$ + ".CYC"
460 ON ERROR GOTO 1070
465 IF MID$(CYFILE$,2,1) = ":" THEN LLL = 14 ELSE LLL = 12
470 IF LEN (CYFILE$) > LLL THEN PRINT:PRINT "FILENAME TOO LONG!":PRINT:GOTO 420
480 IF RIGHT$(CYFILE$,4) = ".CYC" THEN GOTO 530
490 PRINT:PRINT "THIS FILE DOES NOT HAVE THE FILENAME EXTENTION '.CYC'. IT WILL
   RUN ONLY IF THE FILE IS AN INTERPRETED TEXT FILE OF
   CY512 COMMANDS."
500 PRINT:PRINT "DO YOU STILL WANT TO TRY TO EXECUTE THIS FILE ? (Y/N)";
510 ANSWER$=INKEY$:IF LEN(ANSWER$) = 0 THEN GOTO 510
520 IF ANSWER$ <> "Y" THEN GOTO 1040

```

```

530 OPEN "I", #1, CYFILE$
540 ON ERROR GOTO 0
550 PRINT: INVERSE: PRINT "READING IN "; CYFILE$; : NORMAL: PRINT
560 NLINE = 0
570 NLINE = NLINE + 1
580 INPUT #1, CLINE$
590 IF CLINE$ = "*****" THEN LINES = NLINE - 1 : GOTO 610
600 GOTO 570
610 DIM CY$(NLINE, 3)
620 DIM CLINE$(NLINE)
630 CLOSE
640 OPEN "I", #1, CYFILE$
650 FOR I = 1 TO LINES
660 INPUT #1, CLINE$(I)
670 NEXT I
680 CLOSE
690 REM
700 REM *****
710 REM * SEPARATE X, Y, AND Z AXIS
720 REM *****
730 REM
740 PRINT: INVERSE: PRINT "SEPARATING X, Y, AND Z AXIS COMMANDS - LINE#"; : NORMAL: PRINT
750 FOR I = 1 TO LINES
760 XX = (LINES - I) + 1
770 COLON1 = INSTR(CLINE$(I), ":")
780 COLON2 = INSTR(COLON1 + 1, CLINE$(I), ":")
790 CY$(I, 1) = LEFT$(CLINE$(I), COLON1 - 2)
800 CY$(I, 2) = MID$(CLINE$(I), (COLON1 + 2), ((COLON2 - 1) - (COLON1 + 2)))
810 CY3LEN = LEN(CLINE$(I)) - (COLON2 + 1)
820 CY$(I, 3) = RIGHT$(CLINE$(I), CY3LEN)
830 VTAB (11) : HTAB (44) : INVERSE: PRINT XX;
840 CLINE$(I) = ""
850 NEXT I
860 VTAB (11) : HTAB (44) : INVERSE : PRINT " 0 " : PRINT
870 PRINT "CLEARING MEMORY"
880 REM
890 REM *****
900 REM * EXECUTE MOVES
910 REM *****
920 REM
930 NORMAL
940 PRINT CHR$(7) : PRINT "***** READY *****" : PRINT CHR$(7)
950 FOR I = 1 TO LINES
960 IF FRE(0) < 100 THEN PRINT "CLEARING MEMORY - PLEASE WAIT": PRINT FRE("")
970 Q$ = USR1(CY$(I, 1))
980 Q$ = USR2(CY$(I, 2))
990 Q$ = USR3(CY$(I, 3))
1000 Q$ = USR1("G") : Q$ = USR2("G") : Q$ = USR3("G")
1010 NEXT I
1020 PRINT: PRINT "EXECUTION COMPLETE"
1030 CLEAR, &HC400
1040 CHAIN "MENU.BAS", 360
1050 END
1060 REM
1070 REM *****

```

1080 REM * ROUTINE TO HANDLE 'NO FILE' ERRORS
1090 REM *****
1100 REM
1110 IF ERR = 53 OR ERR = 64 THEN PRINT: PRINT "ERROR: FILE NOT FOUND OR BAD FILE NAME!":PRINT:INPUT "PUSH <RETURN> TO CONTINUE",XXX
\$:GOTO 1040
1120 ON ERROR GOTO 0

```

10 REM *****
20 REM *
30 REM *           MZERO.BAS           BY JUSTIN D. REDD
40 REM *                               - BYU CAM LAB -
50 REM *
60 REM * THIS PROGRAM WILL SEND ALL THREE AXIS' (X,Y, AND Z)
70 REM * OF THE MINI-MILLING MACHINE TOWARD THEIR HOME
80 REM * POSITIONS. IT THEN BEGINS CHECKING THE AXIS LIMIT
90 REM * SWITCHES, STOPPING EACH AXIS WHEN IT IS IN IT'S
100 REM * HOME POSITION. THIS PROGRAM IS USUALLY RUN BEFORE
110 REM * RUNNING A SEQUENCE OF MOVES ON THE MILL SINCE IT
120 REM * PROVIDES A CONSISTENT AND ACCURATE STARTING LOCATION.
130 REM *
140 REM *****
150 REM *
160 COMMON MENU$
170 REM
180 PRINT:PRINT
190 PRINT "====> ";:INVERSE:PRINT"ZEROING MILL";:NORMAL:PRINT" <====="
200 PRINT
210 STAT1=&HEOF0 : STAT2=&HEOF1 : STAT3=&HEOF2
220 CY1$="I,R 250,S 1,F 1,N 20000,-,G"
230 GOSUB 350:GOSUB 270:GOSUB 310
240 HOME
250 CHAIN MENU$
260 END
270 CALL CY1(CY1$)
280 X$=RIGHT$((HEX$(PEEK(STAT1))),1)
290 IF ASC(X$) > 55 THEN POKE STAT1,&H44:POKE STAT1,&H40 : RETURN
300 GOTO 280
310 CALL CY2(CY1$)
320 Y$=RIGHT$((HEX$(PEEK(STAT2))),1)
330 IF ASC(Y$) > 55 THEN POKE STAT2,&H44:POKE STAT2,&H40 : RETURN
340 GOTO 320
350 CALL CY3(CY1$)
360 Z$=RIGHT$((HEX$(PEEK(STAT3))),1)
370 IF ASC(Z$) > 55 THEN POKE STAT3,&H44:POKE STAT3,&H40 : RETURN
380 GOTO 360

```

```

10 REM *****
20 REM *
30 REM *          LZERO.BAS          BY JUSTIN D. REDD
40 REM *                                - BYU CAM LAB -
50 REM *
60 REM *  THIS PROGRAM SENDS BOTH AXIS' OF THE LATHE (X AND Z)
70 REM *  BACK TOWARDS THEIR HOME POSITIONS.  IT THEN BEGINS
80 REM *  CHECKING THE LIMIT SWITCHES ON EACH AXIS AND STOPS
90 REM *  THE MOTORS WHEN THE LATHE IS IN HOME POSITION.  THIS
100 REM*  PROGRAM IS USUALLY RUN BEFORE A CUTTING ROUTINE TO
110 REM*  MAKE SURE THE LATHE STARTS FROM THE SAME POSTION EACH
120 REM*  TIME.
130 REM*
140 REM*****
150 REM *
160 PRINT:PRINT
170 PRINT "====> ";:INVERSE:PRINT"ZEROING LATHE";:NORMAL:PRINT" <====="
180 PRINT:PRINT
190 STAT1=&HEOF0 : STAT2=&HEOF2
200 CY1$="I,R 250,S 1,F 1,N 10000,+,G" : CY2$="I,R 250,S 1,F 1,N 20000,-,G"
210 Q$=USR1(CY1$)
220 X$=RIGHT$((HEX$(PEEK(STAT1))),1)
230 IF ASC(X$) > 55 THEN POKE STAT1,&H44:POKE STAT1,&H40:GOTO 250
240 GOTO 220
250 Q$=USR3(CY2$)
260 Z$=RIGHT$((HEX$(PEEK(STAT2))),1)
270 IF ASC(Z$) > 55 THEN POKE STAT2,&H44:POKE STAT2,&H40:GOTO 290
280 GOTO 260
290 CY1$="R 100,S 1,F 1,N 50,+,G":CY2$="R 100,S 1,F 1,N 50,-,G"
300 Q$=USR1(CY1$):Q$=USR3(CY2$)
310 HOME
320 CHAIN "MENU"
330 END

```

APPENDIX K

Turret Punch Control Program

AD-A152 715

MANUFACTURING INFORMATION SYSTEM(U) BRIGHAM YOUNG UNIV
PROVO UT COMPUTER AIDED MFG LAB D K ALLEN ET AL.
26 DEC 84 AFOSR-TR-85-0275 AFOSR-82-0253

3/3

UNCLASSIFIED

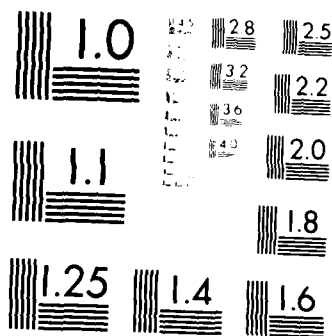
F/G 13/8

NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LIST

```

5 REM *****
6 REM *
7 REM *   MANUAL PUNCH
8 REM *   BY
9 REM *   JUSTIN D. REDD
10 REM *
11 REM *   THIS APPLESOFT PROGRAM WILL
12 REM *   RUN THE MINI TURRET PUNCH
13 REM *   THROUGH STEP BY STEP CONTROL.
14 REM *
15 REM *****
16 REM *
35 CP = 1
40 D$ = CHR$ (4): REM CNTL-D
50 PRINT D$;"BRUN CY DRIVER.OBJ,A$4000"
60 CALL 16384
70 PRINT CHR$ (1);"1R 252,S 1,F 1,N 0,-"
75 Q =USR (1)
80 PRINT CHR$ (1);"2R 252,S 1,F 1,N 0,-"
85 Q =USR (2)
90 PRINT CHR$ (1);"3R 250,S 1,F 1,N 0,+,A"
95 Q =USR (3)
100 PRINT CHR$ (1);"4R 250,S 1,F 1,N 0,-"
105 Q =USR (4)
106 PRINT CHR$ (1);"4N 1200,-,G"
107 Q =USR (4)
110 PD = 1200: REM ** PUNCH DIST.
120 REM ** CP=CURRENT PUNCH, DR$=DIRECTION
130 P1 = 0: REM ** POSITION OF PUNCH 1
140 P2 = 251: REM ** POSITION OF PUNCH 2
150 P3 = 501: REM ** POSITION OF PUNCH 3
160 P4 = 751: REM ** POSITION OF PUNCH 4
200 REM *** MAIN MENU ***
205 HOME : PRINT
210 PRINT "*****": PRINT "***
      ***": PRINT "***      MANUAL PUNCH MAIN MENU
      ***"
220 PRINT "*****"
230 VTAB (10): HTAB (5): PRINT "1) PUNCH HOLE"
240 VTAB (12): HTAB (5): PRINT "2) CHANGE PUNCH SIZES"
250 VTAB (14): HTAB (5): PRINT "3) MOVE X-AXIS"
260 VTAB (16): HTAB (5): PRINT "4) MOVE Y-AXIS"
270 VTAB (18): HTAB (5): PRINT "5) QUIT"
280 VTAB (21): HTAB (10): PRINT "WHICH? "
285 VTAB (24): PRINT "<^D>=DOWN <^U>=UP <^L>=LEFT <^R>=RIGHT
290 GET A$: PRINT A$: PRINT
295 IF A$ = "1" THEN GOTO 1000
300 IF A$ = "2" THEN GOTO 2000
310 IF A$ = "3" THEN GOTO 3000
320 IF A$ = "4" THEN GOTO 4000
321 IF A$ = CHR$ (4) THEN GOTO 5000
322 IF A$ = CHR$ (21) THEN GOTO 5500
323 IF A$ = CHR$ (12) THEN GOTO 6000
324 IF A$ = CHR$ (18) THEN GOTO 6500

```

```

325 IF A$ < > "5" THEN GOTO 200
340 PRINT CHR$ (1); "3N 250"
345 Q = USR (3)
350 PRINT CHR$ (1); "3P 0"
355 Q = USR (3)
356 PRINT CHR$ (1); "3N 250"
357 Q = USR (3)
360 PRINT CHR$ (1); "4N 1200,+,G"
365 Q = USR (4)
370 HOME : END
1000 REM ** SUB TO PUNCH **
1010 HOME : VTAB (11): HTAB (16): FLASH : PRINT ">>> PUNCHING <<<": NORMAL

1020 PRINT CHR$ (1); "4N "; PD; ",+,G"
1030 Q = USR (4)
1040 PRINT CHR$ (1); "4-,G"
1050 Q = USR (4)
1060 GOTO 200
2000 REM ** SUB TO ROTATE TURRET **
2010 HOME
2020 PRINT : PRINT "          ROTATE TURRET"
2030 PRINT : PRINT "*****"
2032 IF CP = 0 THEN GOTO 2040
2034 VT = 6 + (2 * CP)
2036 VTAB (VT): FLASH : PRINT "==>": NORMAL
2040 VTAB (8): HTAB (5): PRINT "1) SMALL PUNCH (? IN.)"
2050 VTAB (10): HTAB (5): PRINT "2) MEDIUM/SMALL PUNCH (? IN.)"
2060 VTAB (12): HTAB (5): PRINT "3) MEDIUM/LARGE PUNCH (? IN.)"
2070 VTAB (14): HTAB (5): PRINT "4) LARGE PUNCH (? IN.)"
2080 VTAB (16): HTAB (5): PRINT "5) RETURN TO MAIN MENU "
2090 VTAB (19): HTAB (10): PRINT "WHICH?"
2100 GET T$
2110 IF T$ = "1" THEN P = P1: GOTO 2200
2120 IF T$ = "2" THEN P = P2: GOTO 2200
2130 IF T$ = "3" THEN P = P3: GOTO 2200
2140 IF T$ = "4" THEN P = P4: GOTO 2200
2150 IF T$ = "5" THEN GOTO 200
2200 PRINT CHR$ (1); "3P "; P
2210 Q = USR (3)
2220 CP = VAL (T$)
2230 GOTO 2000
3000 REM ** SUB TO MOVE X-AXIS **
3005 HOME
3010 PRINT : PRINT "          MOVE X-AXIS"
3020 PRINT : PRINT "*****": PRINT
3025 GOTO 3300
3030 VTAB (8): HTAB (5): PRINT "1) + DIRECTION"
3040 VTAB (10): HTAB (5): PRINT "2) - DIRECTION"
3050 VTAB (12): HTAB (5): PRINT "3) RETURN TO MAIN MENU"
3060 VTAB (15): HTAB (10): PRINT "WHICH? "
3070 GET X$
3080 IF X$ = "1" THEN VTAB (8): FLASH : PRINT "==>": NORMAL : DR$ = "+": GOTO 3200
3090 IF X$ = "2" THEN VTAB (10): FLASH : PRINT "==>": NORMAL : DR$ = "-": GOTO 3200
3100 IF X$ = "3" THEN GOTO 200

```

```

3110 GOTO 3000
3200 VTAB (20): INPUT "DISTANCE ? (IN.) ";ND$
3210 IF ASC (ND$) > 58 THEN GOTO 3200
3220 N = VAL (ND$) * 1600
3230 RETURN
3300 GOSUB 3030
3310 PRINT CHR$ (1);"1N ";N;" ";DR$;"G"
3320 Q = USR (1)
3330 PRINT CHR$ (1);"1N 0"
3340 Q = USR (1)
3350 GOTO 3000
4000 REM ** SUB TO MOVE Y-AXIS **
4010 HOME
4020 PRINT : PRINT "                MOVE Y-AXIS"
4030 PRINT : PRINT "*****": PRINT
4040 GOSUB 3030
4050 PRINT CHR$ (1);"2N ";N;" ";DR$;"G"
4060 Q = USR (2)
4070 PRINT CHR$ (1);"2N 0"
4080 Q = USR (2)
4090 GOTO 4000
5000 REM *** MOVE PUNCH DOWN ***
5010 PRINT CHR$ (1);"4+"
5020 Q = USR (4)
5030 PRINT CHR$ (1);"4+,N 100,G"
5040 Q = USR (4)
5050 GOTO 200
5500 REM *** MOVE PUNCH UP ***
5510 PRINT CHR$ (1);"4-"
5520 Q = USR (4)
5530 PRINT CHR$ (1);"4-,N 100,G"
5540 Q = USR (4)
5550 GOTO 200
6000 REM *** TURRET LEFT 1 STEP ***
6010 PRINT CHR$ (1);"3N 1"
6020 Q = USR (3)
6030 PRINT CHR$ (1);"3N 1,-,G"
6040 Q = USR (3)
6050 GOTO 200
6500 REM *** TURRET RIGHT 1 STEP ***
6510 PRINT CHR$ (1);"3N 1"
6520 Q = USR (3)
6530 PRINT CHR$ (1);"3N 1,+,G"
6540 Q = USR (3)
6550 GOTO 200

```

APPENDIX L

Programs for the Miniature Storage and Retrieval System

```

10 REM*****
12 REM****
13 REM****      STACKER.BAS      BY DAVE JESPERSON      - BYU CAM LAB ****
14 REM****      This program communicates with the apple computer through ****
15 REM****      the serial interface. The apple program that accepts the ****
16 REM****      data from the IBM PC is called STACKER.BAS also, and is ****
17 REM****      run from Microsoft Basic. ****
20 REM****
30 REM****      VARIABLE DEFINITIONS ****
40 REM****
50 REM****      CMD$      = introduction command chosen ****
60 REM****      MCMD$     = main menu option chosen ****
70 REM****      TCMDS     = transport menu option chosen ****
80 REM****      CLEARLINE$ = holds 39 blank spaces, used to clear a line ****
90 REM****      X & Y     = holds screen coordinates ****
100 REM****     I,J       = loop variables ****
110 REM****     BINNO     = user entered bin number ****
120 REM****     TRAYNO    = user entered delivery tray number ****
130 REM****     FLAG      = mark/unmark bin ****
140 REM****             FLAG = 1: mark bin ****
150 REM****             FLAG = 0: unmark bin ****
160 REM****     BIN(18)   = array of bin numbers ****
170 REM****     ROW(18)   = array of cursor row locations ****
180 REM****     COL(18)   = array of cursor column locations ****
190 REM****     STATUS(2) = array to track status of delivery trays ****
200 REM****     MISTAKE    = pointer into error array ****
210 REM****     ERRORMSG$ = array of error messages ****
220 REM****     BINNO$(19) = array of alphabetical bin numbers ****
230 REM****     TRAYNO$    = character input variable for tray number ****
232 REM****     JUNK$      = accept any character from key board ****
234 REM****     STOPIT     = used to stop program when error exists ****
235 REM****     VSKIP$     = stores line coordinates for DRAW command ****
236 REM****     HSKIP$     = same as VSKIP$ ****
237 REM****     TAG        = used in RETURN BIN to hold tray number ****
240 REM****
250 REM*****
260 REM
270 REM*****
280 REM****
290 REM****      DISPLAY SWITCH
300 REM****
310 REM*****
320 REM
330 DEF SEG = 0: POKE &H410, (PEEK(&H410) AND &HCF) OR &H10
340 SCREEN 1,0,0,0: SCREEN 0: WIDTH 40
350 REM
360 REM*****
370 REM****
380 REM****      SYSTEM INITIALIZATION
400 REM****
410 REM*****
420 REM
430 DIM BINNO$(19), BIN(19), ROW(19), COL(19), STATUS(2), ERRORMSG$(12)
440 CLEARLINE$ = SPACE$(39)
442 BINNO$(1) = "11": BIN(1) = 11: ROW(1) = 12: COL(1) = 4
444 BINNO$(2) = "12": BIN(2) = 12: ROW(2) = 9: COL(2) = 4
446 BINNO$(3) = "13": BIN(3) = 13: ROW(3) = 4: COL(3) = 4
448 BINNO$(4) = "21": BIN(4) = 21: ROW(4) = 12: COL(4) = 10
450 BINNO$(5) = "22": BIN(5) = 22: ROW(5) = 8: COL(5) = 10

```

```

460 BINNO$(10)="41":BIN(10)=41:ROW(10)=12:COL(10)=22
462 BINNO$(11)="42":BIN(11)=42:ROW(11)=8:COL(11)=22
464 BINNO$(12)="43":BIN(12)=43:ROW(12)=4:COL(12)=22
466 BINNO$(13)="51":BIN(13)=51:ROW(13)=12:COL(13)=28
468 BINNO$(14)="52":BIN(14)=52:ROW(14)=8:COL(14)=28
470 BINNO$(15)="53":BIN(15)=53:ROW(15)=4:COL(15)=28
472 BINNO$(16)="61":BIN(16)=61:ROW(16)=12:COL(16)=34
474 BINNO$(17)="62":BIN(17)=62:ROW(17)=8:COL(17)=34
476 BINNO$(18)="63":BIN(18)=63:ROW(18)=4:COL(18)=34
480 ERRORMESG$(1) = "INVALID INPUT"
485 ERRORMESG$(2) = "RANGE ERROR"
490 ERRORMESG$(3) = "ALL BINS RETURNED"
495 ERRORMESG$(4) = "BIN(S) STILL OUT"
500 ERRORMESG$(5) = "DELIVERY TRAYS FULL"
505 ERRORMESG$(6) = "TRAY 1 OCCUPIED"
510 ERRORMESG$(7) = "TRAY 2 OCCUPIED"
515 ERRORMESG$(8) = "BIN DELIVERED"
520 ERRORMESG$(9) = "BIN RETURNED"
525 ERRORMESG$(10) = "BIN DOES NOT EXIST"
530 ERRORMESG$(11) = "NO SUCH TRAY"
535 ERRORMESG$(12) = " "
550 STATUS(1) = 0: STATUS(2) = 0
552 CLS:COLOR 23: PRINT "INITIALIZING MULTIPLEXER - PLEASE WAIT" : COLOR 7
555 OPEN "COM1: 4800,N,7,2,CS1000,DS1000" AS #1
556 MUXMESG1$=CHR$(2)+"1":MUXMESG2$=CHR$(1)+"1"
557 PRINT #1,MUXMESG1$ :X=11:GOSUB 565 : PRINT #1,"2" :X=8:GOSUB 565
558 PRINT #1,"5" :X=11 :GOSUB 565 : PRINT #1,"1" : PRINT #1,MUXMESG2$
560 GOTO 570
565 FOR I = 1 TO X
566 INPUT #1,A$
567 NEXT I
568 RETURN
569 REM
570 REM*****
580 REM***
590 REM***      INTRODUCTION SCREEN
600 REM***
610 REM*****
620 REM
630 KEY OFF: SCREEN 0,1: COLOR 15,0,0: WIDTH 40: CLS
640 LOCATE 3,12: PRINT "Computer - Aided"
650 LOCATE 5,8: PRINT "Manufacturing Laboratory"
660 LOCATE 8,8: COLOR 9: PRINT "BRIGHAM YOUNG UNIVERSITY"
670 LOCATE 10,11: PRINT "Provo, Utah 84602"
680 COLOR 15
690 LOCATE 15,9,0: PRINT CHR$(213)+STRING$(20,205)+CHR$(184)
700 LOCATE 16,9,0: PRINT CHR$(179)+"      Mini-LAB      "+CHR$(179)
710 LOCATE 17,9,0: PRINT CHR$(179)+"      Retrieval System  "+CHR$(179)
720 LOCATE 18,9,0: PRINT CHR$(179)+"      Utilities      "+CHR$(179)
730 LOCATE 19,9,0: PRINT CHR$(179)+"      Version 1.0     "+CHR$(179)
740 LOCATE 20,9,0: PRINT CHR$(212)+STRING$(20,205)+CHR$(190)
750 COLOR 12
760 LOCATE 24,7,0: PRINT "Press space bar to continue"
770 POKE 106, 0
780 CMD$ = INKEY$
790 IF CMD$ = "" THEN GOTO 780
800 IF CMD$ = CHR$(27) THEN GOTO 840
810 IF CMD$ = " " THEN GOTO 920
820 GOTO 770

```

```

830 IF (STATUS(1)<>0)OR(STATUS(2)<>0) THEN MISTAKE=4: GOSUB 2130: GOTO 1020
835 EXITPRGM$="2"+CHR$(10):EXITPRGM$=RIGHT$(EXITPRGM$,2)
840 CLS:COLOR 7:WIDTH 80:KEY ON:GOSUB 3530:PRINT #1,EXITPRGM$
842 CLOSE #1: END
850 REM*****
860 REM***
870 REM*** SUBROUTINE MAIN MENU
880 REM***
890 REM***
900 REM***
910 REM*****
920 REM
930 CLS: SCREEN 1,0: COLOR 9,1
940 LOCATE 4,8,1: PRINT "MINI-CAM RETRIEVAL SYSTEM"
950 LOCATE 11,3: PRINT "MAIN MENU OPTIONS:"
960 DEF SEG: POKE &H4E, 2 'change pen color to magenta
970 LOCATE 14,10: PRINT "1. Transport control"
980 LOCATE 16,10: PRINT "2. Inventory Manager"
990 LOCATE 18,10: PRINT "3. Exit Program"
995 LOCATE 24,12: PRINT "<CNTRL-D> for auto-run demo";
1000 LINE (43,11)-(267,43),,B: LINE (41,9)-(269,45),,B
1010 DEF SEG: POKE &H4E, 3: 'change pen color to white
1020 LOCATE 21,1: PRINT CLEARLINE$
1030 LOCATE 21,3: PRINT;"ENTER OPTIDN **> ";
1033 MCMD$=INKEY$: IF LEN (MCMD$) = 0 THEN GOTO 1033
1035 PRINT MCMD$;
1040 IF (MCMD$ = "1") THEN GOSUB 1520
1050 IF (MCMD$ = "2") THEN GOSUB 3000
1060 IF (MCMD$ = "3") THEN GOTO 830
1065 IF (MCMD$ = CHR$(4)) THEN GOSUB 4000
1067 IF DEMO > 0 THEN GOSUB 1520
1070 GOTO 930
1080 REM
1090 REM*****
1100 REM***
1110 REM*** SUBROUTINE RETRIEVAL DRAW
1120 REM***
1130 REM*****
1140 REM
1150 X = 12: Y = 4
1160 FOR I = 1 TO 6
1170 FOR J = 1 TO 3
1180 LOCATE X,Y: PRINT (I*10)+J;
1190 X = X - 4
1200 NEXT J
1210 X = 12: Y = Y + 6
1220 NEXT I
1230 XSKIP$ = "BD23 D10 BD22 D10 BD22 D10 R34 U10 BU22 U10 BU22 U10 BU23"
1240 XSKIP$ = "BR7 R34 BR14 R34 BR14 R34 BR14 R34 BR14 R34 BR14 R34"
1250 DRAW "C1; BM 15,15 R288 D97 L288 U97"
1260 DRAW "BM 15,15 R48 ND97 R48 ND97 R48 ND97 R48 ND97"
1270 DRAW "BM 15,48 R288 BM 15,80 R288 BM 15,38 XHSKIP$; BM 15,70 XHSKIP$;"
1280 DRAW "BM 15,102 XHSKIP$;"
1290 X = 22: PSET(X,15)
1300 FOR I = 1 TO 6
1310 DRAW "C1; XSKIP$;"
1320 X = X + 48: DRAW "BM =X;,15"
1330 NEXT I
1340 DEF SEG: POKE &H4E, 2

```

```

1350 IF STATUS(1) = 0 GOTO 1420
1360 I = 1
1370 WHILE (STATUS(1) <> BIN(I)) AND (I <= 18)
1380     I = I + 1
1390 WEND
1400 LOCATE ROW(I), COL(I): PRINT STATUS(1)
1410 LOCATE ROW(I)+2, COL(I): PRINT 'at 1';
1420 IF STATUS(2) = 0 GOTO 1490
1430 I = 1
1440 WHILE (STATUS(2) <> BIN(I)) AND (I <= 18)
1450     I = I + 1
1460 WEND
1470 LOCATE ROW(I), COL(I): PRINT STATUS(2);
1480 LOCATE ROW(I)+2, COL(I): PRINT 'at 2';
1490 DEF SEG: POKE &H4E, 3
1500 RETURN
1510 REM
1520 REM*****
1530 REM***
1540 REM***      TRANSPORT MENU
1550 REM***
1560 REM*****
1570 REM
1580 CLS: WIDTH 40: SCREEN 1,0: COLOR 9
1590 LOCATE 1,12 : PRINT 'TRANSPORT CONTROL'
1600 GOSUB 1090 'print retrieval draw
1610 GOSUB 1800 'clear menu subroutine
1620 LOCATE 17,3 : PRINT 'TRANSPORT MENU:'
1630 DEF SEG: POKE &H4E, 2
1640 LOCATE 19,12: PRINT '1. Retrieve Bin'
1650 LOCATE 20,12: PRINT '2. Return Bin'
1660 LOCATE 21,12: PRINT '3. Exit To Main Menu'
1670 DEF SEG: POKE &H4E, 3
1680 LOCATE 23,1 : PRINT CLEARLINE$;
1690 LOCATE 23,3 : PRINT:ENTER OPTION ## :;
1691 IF DEMO = 0 THEN DEMOND=DEMOND+1:IF DEMOND > 36 THEN GOTO 4500
1692 IF DEMO > 0 THEN READ TCMD:TCMD$=RIGHT$(STR$(TCMD),1):GOTO 1695
1693 TCMD$=INKEY$: IF LEFT$(TCMD$,1) = 0 THEN GOTO 1693
1695 PRINT TCMD$;
1700 IF (TCMD$ = '1') THEN GOSUB 2420
1710 IF (TCMD$ = '2') THEN GOSUB 2670
1720 IF (TCMD$ = '3') THEN RETURN
1730 GOTO 1610
1740 REM
1750 REM*****
1760 REM***
1770 REM***      SUBROUTINE CLEAR MENU
1780 REM***
1790 REM*****
1800 REM
1810 FOR I = 17 TO 24
1820     LOCATE I,1: PRINT CLEARLINE$;
1830 NEXT
1840 RETURN
1850 REM
1860 REM*****
1870 REM***
1880 REM***      SUBROUTINE MARK-BIN
1890 REM***
1900 REM*****

```

```

1910 REM
1920 I = 1:
1930 WHILE (BINNO <> BIN(I)) AND (I <= 18)
1940     I = I + 1
1950 WEND
1960 IF FLAG = 1 GOTO 2010
1970 DEF SEG: POKE &H4E, 3
1980 LOCATE ROW(I), COL(I): PRINT BINNO:
1990 LOCATE ROW(I)+2, COL(I): PRINT " ";
2000 GOTO 2060
2010 DEF SEG: POKE &H4E, 2
2020 LOCATE ROW(I), COL(I): PRINT BINNO:
2030 LOCATE ROW(I)+2, COL(I)
2040 IF TRAYNO = 1 THEN PRINT "at 1" ELSE PRINT "at 2"
2050 DEF SEG: POKE &H4E, 3
2060 RETURN
2070 REM
2080 REM*****
2090 REM****
2100 REM**** SUBROUTINE ERROR MESSAGE
2110 REM****
2120 REM*****
2130 REM
2140 STOPIT = 1
2150 LOCATE 25,1: PRINT CLEARLINE$;
2160 PRINT CHR$(7);
2170 DEF SEG: POKE &H4E, 1
2180 LOCATE 25,3: PRINT "***> ";ERRORMESG$(MISTAKE); " <***";
2190 JUNK$ = INKEY$: IF JUNK$ = "" THEN 2190
2200 LOCATE 25,1: PRINT CLEARLINE$;
2210 DEF SEG: POKE &H4E, 3
2220 RETURN
2230 REM
2240 REM*****
2250 REM****
2260 REM**** SUBROUTINE RETRIEVE CHECK
2270 REM****
2280 REM*****
2290 REM
2300 STOPIT = 0
2310 IF (TRAYNO = 1) AND (STATUS(1) <> 0) THEN GOTO 2320 ELSE GOTO 2330
2320     MISTAKE = 6: GOSUB 2130: RETURN
2330 IF (TRAYNO = 2) AND (STATUS(2) <> 0) THEN GOTO 2340 ELSE GOTO 2350
2340     MISTAKE = 7: GOSUB 2130
2350 RETURN
2360 REM
2370 REM*****
2380 REM****
2390 REM**** SUBROUTINE RETRIEVE BIN
2400 REM****
2410 REM*****
2420 REM
2430 GOSUB 1800 'clear menu
2440 LOCATE 17,3: PRINT "RETRIEVE BIN:"
2450 IF (STATUS(1) <> 0) AND (STATUS(2) <> 0) THEN MISTAKE=5: GOSUB 2130: RETURN

```

```

2460 GOSUB 2850 'input bin number
2470 LOCATE 25,1: PRINT CLEARLINE$;
2480 IF BINNO = 0 THEN RETURN
2490 IF (BINNO=STATUS(1))OR(BINNO=STATUS(2)) THEN MISTAKE=8:GOSUB 2130:GOTO 2460
2492 IF STATUS(1) <> 0 AND STATUS(2) = 0 THEN TRAYNO$ = '2' : GOTO 2517
2494 IF STATUS(1) = 0 AND STATUS(2) <> 0 THEN TRAYNO$ = '1' : GOTO 2517
2500 LOCATE 21, 1: PRINT CLEARLINE$;
2510 LOCATE 21,3: PRINT;'DELIVERY TRAY **> ' ;
2511 IF DEMO > 0 THEN READ TRAYNO:TRAYNO$=RIGHT$(STR$(TRAYNO),1) : GOTO 2515
2513 TRAYNO$=INKEY$ : IF LEN(TRAYNO$) = 0 THEN GOTO 2513
2515 PRINT TRAYNO$;:GOTO 2520
2517 LOCATE 21,3: PRINT;'DELIVERY TRAY **> ' ;TRAYNO$;
2520 IF (TRAYNO$<>'1')AND(TRAYNO$<>'2') THEN MISTAKE=11: GOSUB 2130: GOTO 2500
2530 IF TRAYNO$ = '1' THEN TRAYNO = 1 ELSE TRAYNO = 2
2540 GOSUB 2290 'check routine
2550 IF STOPIT = 1 GOTO 2500
2560 FLAG = 1
2580 STATUS(TRAYNO) = BINNO
2585 GOSUB 3530
2590 OP$='1'+CHR$(10):B$=STR$(BINNO)+CHR$(10):T$=STR$(TRAYNO)+CHR$(10)
2591 OP$=RIGHT$(OP$,2):PRINT #1,OP$;:RLINE=1:GOTO 3460
2592 B$=RIGHT$(B$,3):PRINT #1,B$;:RLINE = 2:GOTO 3460
2593 T$=RIGHT$(T$,2):PRINT #1,T$;
2595 GOSUB 1910 'mark routine
2600 RETURN
2610 REM
2620 REM*****
2630 REM***
2640 REM*** SUBROUTINE RETURN BIN
2650 REM***
2660 REM*****
2670 REM
2680 GOSUB 1800 'clear menu
2690 LOCATE 17,3: PRINT 'RETURN BIN:'
2700 IF (STATUS(1)=0)AND(STATUS(2)=0) THEN MISTAKE=3:GOSUB 2130: RETURN
2702 IF STATUS(1) <> 0 AND STATUS(2) = 0 THEN BIN$=RIGHT$(STR$(STATUS(1)),2) : G
2705 : GOTO 2720
2704 IF STATUS(1) = 0 AND STATUS(2) <> 0 THEN BIN$=RIGHT$(STR$(STATUS(2)),2) : G
2705 : GOTO 2720
2710 GOSUB 2850 'input bin number
2720 LOCATE 25,1: PRINT CLEARLINE$;
2730 IF BINNO = 0 THEN RETURN
2740 IF (STATUS(1)<>BINNO) AND (STATUS(2)<>BINNO) THEN GOTO 2750 ELSE GOTO 2760
2750 MISTAKE = 9: GOSUB 2130: GOTO 2710
2760 FLAG = 0
2780 IF STATUS(1)=BINNO THEN STATUS(1)=0: TAG=1 ELSE STATUS(2)=0: TAG=2
2781 GOSUB 3530
2782 OP$='0'+CHR$(10):B$=STR$(BINNO)+CHR$(10):T$=STR$(TAG)+CHR$(10)
2783 OP$=RIGHT$(OP$,2):PRINT #1,OP$;:RLINE=3:GOTO 3460
2784 B$=RIGHT$(B$,3):PRINT #1,B$;:RLINE=4:GOTO 3460
2785 T$=RIGHT$(T$,2):PRINT #1,T$;
2788 GOSUB 1910 'SUBROUTINE MARK BIN
2790 RETURN
2800 REM*****
2810 REM***
2820 REM*** SUBROUTINE INPUT BIN NUMBER
2830 REM***
2840 REM*****
2850 REM

```

```

2860 DEF SEG: POKE &H4E,1
2870 LOCATE 25,12: PRINT CHR$(26)+ " QUIT to exit " +CHR$(27);
2880 DEF SEG: POKE &H4E,3
2890 LOCATE 19,1: PRINT CLEARLINE$
2895 IF DEMO > 0 THEN READ BIN:BIN$=RIGHT$(STR$(BIN),2) : GOTO 2905
2900 LOCATE 19,3: INPUT;"BIN NUMBER **> ",BIN$ : GOTO 2910
2905 LOCATE 19,3: PRINT;"BIN NUMBER **> ";BIN$
2910 IF (BIN$="0") OR (BIN$="QUIT") THEN BINNO = 0: RETURN
2920 I = 1
2930 WHILE (BIN$ <> BINNO$(I)) AND (I < 19)
2940     I = I + 1
2950 WEND
2960 IF I = 19 THEN MISTAKE = 10: GOSUB 2130: GOTO 2860
2970 BINNO = BIN (I)
2980 RETURN
2990 REM
3000 REM*****
3010 REM****
3020 REM**** SUBROUTINE INVENTORY MENU
3030 REM****
3040 REM*****
3050 REM
3060 CLS: WIDTH 40: SCREEN 1,0: COLOR 9
3070 LOCATE 1,12 : PRINT "INVENTORY MANAGER"
3080 GOSUB 1090
3090 LOCATE 17,3 : PRINT "INVENTORY MENU"
3100 DEF SEG: POKE &H4E, 2
3110 LOCATE 19,12: PRINT "1. System Editor"
3120 LOCATE 20,12: PRINT "2. List Empty Bins"
3130 LOCATE 21,12: PRINT "3. List All Bins"
3140 LOCATE 22,12: PRINT "4. Return To Main Menu"
3150 DEF SEG: POKE &H4E, 3
3160 LOCATE 24,1 : PRINT CLEARLINE$;
3170 LOCATE 24,3 : INPUT;"ENTER OPTION **> ",ICMD$
3180 IF (ICMD$ = "1") THEN
3190 IF (ICMD$ = "2") THEN
3200 IF (ICMD$ = "3") THEN
3210 IF (ICMD$ = "4") THEN RETURN
3220 GOTO 3160
3400 REM*****
3410 REM****
3420 REM**** READY TO TRANSMIT HANDSHAKE
3430 REM**** WITH APPLE
3440 REM****
3450 REM*****
3460 INPUT #1,CHECK$
3461 X=INSTR(CHECK$,"ON");Y=INSTR(CHECK$,"WHAT?")
3462 IF X=0 AND Y=0 THEN GOTO 3460
3463 IF Y=0 THEN ON RLINE GOTO 2591,2592,2783,2784
3464 FOR I = 1 TO 25:NEXT I
3465 ON RLINE GOTO 2592,2593,2784,2785
3480 REM*****
3490 REM****
3500 REM**** CHECK CLEAR TO SEND LINE ON APPLE
3510 REM****
3520 REM*****

```

```

3530 INPUT #1,CHECK$
3540 X=INSTR(CHECK$,"READY")
3550 IF X=0 THEN GOTO 3530
3560 RETURN
4000 REM *****
4010 REM ***
4020 REM ***   AUTO-RUN DEMO SETUP
4030 REM ***
4040 REM *****
4050 REM
4060 CLS : PRINT "THE DEMO ROUTINE WILL AUTOMATICALLY "
4070 PRINT "CYCLE THROUGH ALL THE DRAWERS" :PRINT:PRINT:PRINT
4080 PRINT "DO YOU STILL WANT TO USE THE DEMO ?(Y/N)";
4090 DEMO$=INKEY$ : IF LEN(DEMO$) = 0 THEN GOTO 4090
4100 IF DEMO$ <> "Y" THEN RETURN
4110 PRINT:PRINT:PRINT:INPUT "CYCLE THROUGH HOW MANY TIMES ?",DEMO$
4120 IF ASC(DEMO$) < 48 OR ASC(DEMO$) > 57 THEN GOTO 4110
4130 DEMO=VAL(DEMO$)
4135 DEMOND = 0
4140 DATA 1,43,1,1,22,2,43,2,1,61,1,1,11,2,61,2,1,52,1,1,33,2,52,2
4150 DATA 1,41,1,1,23,2,41,2,1,63,1,1,13,2,63,2,1,53,1,1,31,2,53,2
4160 DATA 1,42,1,1,21,2,42,2,1,62,1,1,12,2,62,2,1,51,1,1,32,2,51,2
4165 DATA 1,11,1,2,11
4170 RETURN
4500 REM *****
4510 REM ***
4520 REM ***   CHECK FOR ANOTHER CYCLE IN DEMO
4530 REM ***
4540 REM *****
4550 REM
4560 DEMO = DEMO - 1
4570 IF DEMO = 0 THEN PRINT "3";RETURN
4580 DEMOND = 0
4590 RESTORE
4600 GOTO 1692

```

```

0171.47) *8.1...1: CIRCLE(271,47)*24.1...1: CIRCLE(271,47)*40.1...1
0172.47) *9.1...1: CIRCLE(271,47)*60.1...1: CIRCLE(271,47)*80.1...1
0173.47) *10.1...1: CIRCLE(271,47)*100.1...1: CIRCLE(271,47)*120.1...1
0174.47) *11.1...1: CIRCLE(271,47)*140.1...1: CIRCLE(271,47)*160.1...1
0175.47) *12.1...1: CIRCLE(271,47)*180.1...1: CIRCLE(271,47)*200.1...1
0176.47) *13.1...1: CIRCLE(271,47)*240.1...1: CIRCLE(271,47)*280.1...1
0177.47) *14.1...1: CIRCLE(271,47)*320.1...1: CIRCLE(271,47)*360.1...1
0178.47) *15.1...1: CIRCLE(271,47)*400.1...1: CIRCLE(271,47)*440.1...1
0179.47) *16.1...1: CIRCLE(271,47)*480.1...1: CIRCLE(271,47)*520.1...1
0180.47) *17.1...1: CIRCLE(271,47)*600.1...1: CIRCLE(271,47)*640.1...1
0181.47) *18.1...1: CIRCLE(271,47)*720.1...1: CIRCLE(271,47)*760.1...1
0182.47) *19.1...1: CIRCLE(271,47)*840.1...1: CIRCLE(271,47)*880.1...1
0183.47) *20.1...1: CIRCLE(271,47)*960.1...1: CIRCLE(271,47)*1000.1...1
0184.47) *21.1...1: CIRCLE(271,47)*1080.1...1: CIRCLE(271,47)*1120.1...1
0185.47) *22.1...1: CIRCLE(271,47)*1200.1...1: CIRCLE(271,47)*1240.1...1
0186.47) *23.1...1: CIRCLE(271,47)*1320.1...1: CIRCLE(271,47)*1360.1...1
0187.47) *24.1...1: CIRCLE(271,47)*1440.1...1: CIRCLE(271,47)*1480.1...1
0188.47) *25.1...1: CIRCLE(271,47)*1600.1...1: CIRCLE(271,47)*1640.1...1
0189.47) *26.1...1: CIRCLE(271,47)*1760.1...1: CIRCLE(271,47)*1800.1...1
0190.47) *27.1...1: CIRCLE(271,47)*1920.1...1: CIRCLE(271,47)*1960.1...1
0191.47) *28.1...1: CIRCLE(271,47)*2080.1...1: CIRCLE(271,47)*2120.1...1
0192.47) *29.1...1: CIRCLE(271,47)*2240.1...1: CIRCLE(271,47)*2280.1...1
0193.47) *30.1...1: CIRCLE(271,47)*2400.1...1: CIRCLE(271,47)*2440.1...1
0194.47) *31.1...1: CIRCLE(271,47)*2560.1...1: CIRCLE(271,47)*2600.1...1
0195.47) *32.1...1: CIRCLE(271,47)*2720.1...1: CIRCLE(271,47)*2760.1...1
0196.47) *33.1...1: CIRCLE(271,47)*2880.1...1: CIRCLE(271,47)*2920.1...1
0197.47) *34.1...1: CIRCLE(271,47)*3040.1...1: CIRCLE(271,47)*3080.1...1
0198.47) *35.1...1: CIRCLE(271,47)*3200.1...1: CIRCLE(271,47)*3240.1...1
0199.47) *36.1...1: CIRCLE(271,47)*3360.1...1: CIRCLE(271,47)*3400.1...1
0200.47) *37.1...1: CIRCLE(271,47)*3520.1...1: CIRCLE(271,47)*3560.1...1
0201.47) *38.1...1: CIRCLE(271,47)*3680.1...1: CIRCLE(271,47)*3720.1...1
0202.47) *39.1...1: CIRCLE(271,47)*3840.1...1: CIRCLE(271,47)*3880.1...1
0203.47) *40.1...1: CIRCLE(271,47)*4000.1...1: CIRCLE(271,47)*4040.1...1
0204.47) *41.1...1: CIRCLE(271,47)*4160.1...1: CIRCLE(271,47)*4200.1...1
0205.47) *42.1...1: CIRCLE(271,47)*4320.1...1: CIRCLE(271,47)*4360.1...1
0206.47) *43.1...1: CIRCLE(271,47)*4480.1...1: CIRCLE(271,47)*4520.1...1
0207.47) *44.1...1: CIRCLE(271,47)*4640.1...1: CIRCLE(271,47)*4680.1...1
0208.47) *45.1...1: CIRCLE(271,47)*4800.1...1: CIRCLE(271,47)*4840.1...1
0209.47) *46.1...1: CIRCLE(271,47)*4960.1...1: CIRCLE(271,47)*5000.1...1
0210.47) *47.1...1: CIRCLE(271,47)*5120.1...1: CIRCLE(271,47)*5160.1...1
0211.47) *48.1...1: CIRCLE(271,47)*5280.1...1: CIRCLE(271,47)*5320.1...1
0212.47) *49.1...1: CIRCLE(271,47)*5440.1...1: CIRCLE(271,47)*5480.1...1
0213.47) *50.1...1: CIRCLE(271,47)*5600.1...1: CIRCLE(271,47)*5640.1...1
0214.47) *51.1...1: CIRCLE(271,47)*5760.1...1: CIRCLE(271,47)*5800.1...1
0215.47) *52.1...1: CIRCLE(271,47)*5920.1...1: CIRCLE(271,47)*5960.1...1
0216.47) *53.1...1: CIRCLE(271,47)*6080.1...1: CIRCLE(271,47)*6120.1...1
0217.47) *54.1...1: CIRCLE(271,47)*6240.1...1: CIRCLE(271,47)*6280.1...1
0218.47) *55.1...1: CIRCLE(271,47)*6400.1...1: CIRCLE(271,47)*6440.1...1
0219.47) *56.1...1: CIRCLE(271,47)*6560.1...1: CIRCLE(271,47)*6600.1...1
0220.47) *57.1...1: CIRCLE(271,47)*6720.1...1: CIRCLE(271,47)*6760.1...1
0221.47) *58.1...1: CIRCLE(271,47)*6880.1...1: CIRCLE(271,47)*6920.1...1
0222.47) *59.1...1: CIRCLE(271,47)*7040.1...1: CIRCLE(271,47)*7080.1...1
0223.47) *60.1...1: CIRCLE(271,47)*7200.1...1: CIRCLE(271,47)*7240.1...1
0224.47) *61.1...1: CIRCLE(271,47)*7360.1...1: CIRCLE(271,47)*7400.1...1
0225.47) *62.1...1: CIRCLE(271,47)*7520.1...1: CIRCLE(271,47)*7560.1...1
0226.47) *63.1...1: CIRCLE(271,47)*7680.1...1: CIRCLE(271,47)*7720.1...1
0227.47) *64.1...1: CIRCLE(271,47)*7840.1...1: CIRCLE(271,47)*7880.1...1
0228.47) *65.1...1: CIRCLE(271,47)*8000.1...1: CIRCLE(271,47)*8040.1...1
0229.47) *66.1...1: CIRCLE(271,47)*8160.1...1: CIRCLE(271,47)*8200.1...1
0230.47) *67.1...1: CIRCLE(271,47)*8320.1...1: CIRCLE(271,47)*8360.1...1
0231.47) *68.1...1: CIRCLE(271,47)*8480.1...1: CIRCLE(271,47)*8520.1...1
0232.47) *69.1...1: CIRCLE(271,47)*8640.1...1: CIRCLE(271,47)*8680.1...1
0233.47) *70.1...1: CIRCLE(271,47)*8800.1...1: CIRCLE(271,47)*8840.1...1
0234.47) *71.1...1: CIRCLE(271,47)*8960.1...1: CIRCLE(271,47)*9000.1...1
0235.47) *72.1...1: CIRCLE(271,47)*9120.1...1: CIRCLE(271,47)*9160.1...1
0236.47) *73.1...1: CIRCLE(271,47)*9280.1...1: CIRCLE(271,47)*9320.1...1
0237.47) *74.1...1: CIRCLE(271,47)*9440.1...1: CIRCLE(271,47)*9480.1...1
0238.47) *75.1...1: CIRCLE(271,47)*9600.1...1: CIRCLE(271,47)*9640.1...1
0239.47) *76.1...1: CIRCLE(271,47)*9760.1...1: CIRCLE(271,47)*9800.1...1
0240.47) *77.1...1: CIRCLE(271,47)*9920.1...1: CIRCLE(271,47)*9960.1...1
0241.47) *78.1...1: CIRCLE(271,47)*10080.1...1: CIRCLE(271,47)*10120.1...1
0242.47) *79.1...1: CIRCLE(271,47)*10240.1...1: CIRCLE(271,47)*10280.1...1
0243.47) *80.1...1: CIRCLE(271,47)*10400.1...1: CIRCLE(271,47)*10440.1...1
0244.47) *81.1...1: CIRCLE(271,47)*10560.1...1: CIRCLE(271,47)*10600.1...1
0245.47) *82.1...1: CIRCLE(271,47)*10720.1...1: CIRCLE(271,47)*10760.1...1
0246.47) *83.1...1: CIRCLE(271,47)*10880.1...1: CIRCLE(271,47)*10920.1...1
0247.47)
```

```

1000 FOR I=1 TO 1000:NEXT:IF LOC(MUXFZ)=0 THEN PRINT'HAVING PROBLEMS RECIEVING'
1010 GOTO 2920
1020 IF NOT (LOC(MUXFZ),#MUXFZ):X=INSTR(ANSWER$,'READY'):IF X=0 THEN PRINT
'DIDN'T RECIEVE READY FROM LATHE':GOTO 2910
1030 PRINT #MUXFZ,'DESIGN';CHR$(10); 'SELECT DESIGN OPTION
1040 INPUT #MUXFZ,ANSWER$:X=INSTR(ANSWER$,'UK'):IF X=0 THEN PRINT'DIDN'T TAKE DE
SIGN CHOICE':GOTO 2940
1050 FOR HOLDZ=1 TO 2000:NEXT:IF LOC(MUXFZ)=0 THEN GOTO 2960
1060 ANSWER$=INPUT$ (LOC(MUXFZ),#MUXFZ):X=INSTR(ANSWER$,'PLOADED'):IF X=0 THEN P
RINT'HAVING PROBLEMS RECIEVING FROM APPLE':PRINT #MUXFZ,'BAD';CHR$(10);GOTO 296
0
1070 PRINT #MUXFZ,'UK';CHR$(10);
1080 INPUT #MUXFZ,ANSWER$:X=INSTR(ANSWER$,'GOOD'):IF X=0 THEN PRINT 'HAVING PROB
LEMS SENDING TO IBM':PRINT'APPLE SENT ==>';ANSWER$:GOTO 2960
1090 PRINT #1,PARAM$(1,1)
1100 GOSUB 1630
1110 PRINT #MUXFZ,PARAM$(1,2);CHR$(10); 'PART NAME
1120 GOSUB 3200
1130 PRINT'TRANSMITTING DATA';
1140 PRINT #MUXFZ,PRTFAMLY$;CHR$(10);GOSUB 1980 'PART FAMILY
1150 PRINT #MUXFZ,PARAM$(2,1);CHR$(10);PRINT'.':GOSUB 3200 'PART NUMBER
1160 PRINT #MUXFZ,NUMBDIAM$;CHR$(10);PRINT'.':GOSUB 3200 'NUMBER OF DIAMETERS
1170 PRINT #MUXFZ,PARAM$(12,1);CHR$(10);GOSUB 3200 'FEED RATE
1180 PRINT #MUXFZ,PARAM$(12,2);CHR$(10);PRINT'.':GOSUB 3200 'CUT DEPTH
1190 PRINT #MUXFZ,STOCKWIDTH$;CHR$(10);GOSUB 3200 'STOCK DIAMETER
1200 PRINT #MUXFZ,STOCKLEN$;CHR$(10);PRINT'.':GOSUB 3200 'STOCK LENGTH
1210 FOR I = 3 TO VAL(NUMBDIAM$)+2
1220   FOR J = 1 TO 2
1230     PRINT #MUXFZ, PARAM$(I,J);CHR$(10); 'PART DIMENSIONS
1240     GOSUB 3200
1250   NEXT J
1260 NEXT I
1270 GOSUB 3630
1280 RETURN
1290 REM TIMING LOOP
1300 FOR KK = 1 TO 100: NEXT KK
1310 RETURN
1320 REM SUBROUTINE TO DRAW TITLE BLOCK
1330 LINE(106,148)-(317,197),3,B
1340 LINE(104,146)-(319,199),3,B
1350 LINE(106,171)-(317,171)
1360 LOCATE 20,23,0: PRINT 'PART NAME';
1370 LOCATE 23,16,0: PRINT 'BASIC SHAPE: ';PARAM$(1,1)
1380 LOCATE 24,16,0: PRINT 'PART NUMBER: ';
1390 DEF SEG: POKE $H4E,2
1400 LOCATE 20,1,0 : PRINT 'enter data';
1410 LOCATE 22,1,0 : PRINT 'at cursor';
1420 LOCATE 24,2,0 : PRINT 'position';
1430 DEF SEG: POKE $H4E,3
1440 LOCATE 21,16,1: LINE INPUT; PARAM$(1,2)
1450 LOCATE 24,29,1: LINE INPUT; PARAM$(2,1)
1460 RETURN
1470 REM SUBROUTINE A20
1480 CLS: SCREEN 1,0: COLOR 0,1: KEY OFF
1490 LOCATE 2,2,0: PRINT CHR$(24): LOCATE 11,2: PRINT CHR$(25)
1500 LOCATE 5,19 : PRINT CHR$(25): LOCATE 8,19: PRINT CHR$(24)
1510 LOCATE 4,24 : PRINT CHR$(24): LOCATE 9,24: PRINT CHR$(25)
1520 LOCATE 13,6 : PRINT CHR$(27): LOCATE 13,9: PRINT CHR$(26)
1530 LOCATE 15,6 : PRINT CHR$(27): LOCATE 15,13: PRINT CHR$(26)
1540 LOCATE 17,6 : PRINT CHR$(27): LOCATE 17,17: PRINT CHR$(26)
1550 DEF SEG: POKE $H4E,3
1560 DEF SEG: POKE $H4E,2
1570 DEF SEG: POKE $H4E,3
1580 DEF SEG: POKE $H4E,2
1590 DEF SEG: POKE $H4E,3
1600 DEF SEG: POKE $H4E,2
1610 DEF SEG: POKE $H4E,3
1620 DEF SEG: POKE $H4E,2
1630 DEF SEG: POKE $H4E,3
1640 DEF SEG: POKE $H4E,2
1650 DEF SEG: POKE $H4E,3
1660 DEF SEG: POKE $H4E,2
1670 DEF SEG: POKE $H4E,3
1680 DEF SEG: POKE $H4E,2
1690 DEF SEG: POKE $H4E,3
1700 DEF SEG: POKE $H4E,2
1710 DEF SEG: POKE $H4E,3
1720 DEF SEG: POKE $H4E,2
1730 DEF SEG: POKE $H4E,3
1740 DEF SEG: POKE $H4E,2
1750 DEF SEG: POKE $H4E,3
1760 DEF SEG: POKE $H4E,2
1770 DEF SEG: POKE $H4E,3
1780 DEF SEG: POKE $H4E,2
1790 DEF SEG: POKE $H4E,3
1800 DEF SEG: POKE $H4E,2
1810 DEF SEG: POKE $H4E,3
1820 DEF SEG: POKE $H4E,2
1830 DEF SEG: POKE $H4E,3
1840 DEF SEG: POKE $H4E,2
1850 DEF SEG: POKE $H4E,3
1860 DEF SEG: POKE $H4E,2
1870 DEF SEG: POKE $H4E,3
1880 DEF SEG: POKE $H4E,2
1890 DEF SEG: POKE $H4E,3
1900 DEF SEG: POKE $H4E,2
1910 DEF SEG: POKE $H4E,3
1920 DEF SEG: POKE $H4E,2
1930 DEF SEG: POKE $H4E,3
1940 DEF SEG: POKE $H4E,2
1950 DEF SEG: POKE $H4E,3
1960 DEF SEG: POKE $H4E,2
1970 DEF SEG: POKE $H4E,3
1980 DEF SEG: POKE $H4E,2
1990 DEF SEG: POKE $H4E,3
2000 DEF SEG: POKE $H4E,2

```

```

1000 IF MID$(K$,1,1) = "1" THEN GOTO 2270
1010 IF MID$(K$,1,1) = "2" THEN GOTO 2290
1020 IF MID$(K$,1,1) = "3" THEN PRTFAMILY$="A00":NUMBDIAM$="1":GOSUB 2380
1030 IF MID$(K$,1,1) = "2" THEN PRTFAMILY$="A10":NUMBDIAM$="2":GOSUB 2530
1040 IF MID$(K$,1,1) = "3" THEN PRTFAMILY$="A20":NUMBDIAM$="3":GOSUB 3380
1050 IF MID$(K$,1,1) = "4" THEN GOSUB 2750
1060 IF MID$(K$,1,1) = CHR$(27) THEN GOTO 2360
1070 GOTO 2160
2300 SCREEN 0:WIDTH 80:COLOR 7,0
2370 RETURN
2380 REM Subroutine "A00"
2390 CLS: SCREEN 1,0: COLOR 0,1
2400 LINE (31,55)-(159,111),1,B
2410 LOCATE 5,5,0: PRINT CHR$(27)
2420 LOCATE 5,20 : PRINT CHR$(26)
2430 LOCATE 8,23 : PRINT CHR$(24)
2440 LOCATE 14,23: PRINT CHR$(25)
2450 DRAW "C3; BM 31,31 D21 U17 R56 BR16 R56 U4 D21"
2460 DRAW "BM 162,55 R20 L3 D20 BD16 D20 R4 L20 R17 U20 BU16 U20"
2470 CIRCLE(259,83),28,1,,,1
2480 DRAW "C3; BM 259,51 D28 BD2 D4 BD2 D28 BM 227,83 R28 BR3 R2 BR3 R28"
2490 PARAM$(1,1) = "A00": GOSUB 3230
2500 LOCATE 4,12,0: LINE INPUT; PARAM$(3,1) 'OVERALL LENGTH
2510 LOCATE 11,22 : LINE INPUT; PARAM$(3,2) 'OVERALL DIAMETER
2520 RETURN
2530 REM SUBROUTINE A10
2540 CLS: SCREEN 1,0: COLOR 0,1: KEY OFF
2550 PSET(63,47): DRAW "C1; R32 DB8 L32 U88 BM 95,63 R32 D56 L32"
2560 CIRCLE(243,91),28,1,,,1: CIRCLE(243,91),44,1,,,1
2570 PSET(195,91) : DRAW "R44 BR3 R2 BR3 R44"
2580 PSET(243,43) : DRAW "D44 BD3 D2 BD3 D44"
2590 LOCATE 3,9,0 : PRINT CHR$(27): LOCATE 5,9,0 : PRINT CHR$(27)
2600 LOCATE 3,16,0: PRINT CHR$(26): LOCATE 5,12 : PRINT CHR$(26)
2610 LOCATE 9,19 : PRINT CHR$(24): LOCATE 15,19 : PRINT CHR$(25)
2620 LOCATE 7,6,0 : PRINT CHR$(24): LOCATE 17,6,0: PRINT CHR$(25)
2630 DRAW "BM 39,47 R20 BM 39,135 R20 BM 43,47 D40 BD8 D40"
2640 DRAW "BM 63,15 D27 BM 95,32 D12 BM 127,15 D43"
2650 DRAW "BM 63,19 R28 BR8 R28 BM 63,35 R12 BR8 R12"
2660 DRAW "BM 131,63 R20 BM 131,119 R20 BM 147,63 D24 BD8 D24"
2670 DRAW "BM 148,63 D24 BD8 D24 BM 44,47 D40 BD8 D40"
2680 PARAM$(1,1) = "A10": GOSUB 3230
2690 LOCATE 2,11,1: LINE INPUT; PARAM$(4,1) 'LENGTH DIAM #1 AND #2
2700 LOCATE 4,11,1: LINE INPUT; PARAM$(3,1) 'LENGTH OF LARGER DIAM (#1)
2710 LOCATE 12,4,1: LINE INPUT; PARAM$(3,2) 'WIDTH OF LARGER DIAM (#1)
2720 LOCATE 12,18,1: LINE INPUT; PARAM$(4,2) 'WIDTH OF SMALLER DIAM (#2)
2730 LENGTHDIAM=VAL(PARAM$(4,1))-VAL(PARAM$(3,1)):PARAM$(4,1)=STR$(LENGTHDIAM) '
FIND LENGTH OF DIAM #2
2740 RETURN
2750 REM APPLE DOWN-LOAD ROUTINE
2760 CLS: WIDTH 40: SCREEN 0,1: COLOR 15
2770 LOCATE 1,15: PRINT "IRM > APPLE"
2780 LOCATE 3,13: PRINT "DOWN-LOAD ROUTINE"
2790 LOCATE 7,2 : PRINT "PART NAME: ";PARAM$(1,2)
2800 LOCATE 9,2 : PRINT "PART NUMBER: ";PARAM$(2,1)
2810 LOCATE 11,2: PRINT "BASIC SHAPE: ";PARAM$(1,1)
2820 PRINT:INPUT " ENTER STOCK LENGTH: ",STOCKLEN$
2830 PRINT:INPUT " ENTER STOCK WIDTH: ",STOCKWIDTH$
2840 PRINT:INPUT " ENTER CUTTING SPEED: ",PARAM$(12,1)
2850 PRINT:INPUT " ENTER CUTTING DEPTH: ",PARAM$(12,2)
2860 PRINT:PRINT "Press space bar when ready to send data"
2870 GOTO 106,0: CHDD$ = INKEY$: IF CHDD$ <> " " GOTO 2870
2880 SEND DATA TO APPLE
2890 REM INCH:GOSUB 4870 TURN MUX PORT ON FOR LATHE
2900 REM:PRINT:GETTING APPLE READY TO RECEIVE DATA"
2910 PRINT "READY":CHR$(10):

```

```

1700 PRINT:PRINT 'RECOPYING DCLASS ECREATED CONTROL FILE INTO FILE ';&CONFILE$
1710 OPEN 'O'&TEMFF2%&CONFILE$
1720 IF EOF(TEMFF2%) GOTO 1790
1730 INPUT &TEMFF2%,INLINE$
1740 PRINT &TEMFF2%,INLINE$
1750 PRINT'. '
1760 GOTO 1740
1770 CLOSE &TEMFF2%:CLOSE &TEMFF2%
1780 PRINT:PRINT'NEW PROCESS CONTROL FILE ';&CONFILE$' IS READY':FOR HOLD%=1 TO
1790:NEXT
1800 RETURN
1810 'READ IN PART DIMENSIONES
1820 FOR DIAMN%=3 TO VAL(INPARAM$(4))+2
1830 FOR PRTPARM%=1 TO 2
1840 INPUT &TEMFF2%,INLINE$
1850 INPUT &TEMFF2%,EXTRACR$
1860 IF LEFT$(INLINE$,1)=' ' THEN INLINE%=MID$(INLINE$,2):GOTO 1500 'STRIP LEADI
NG SPACES
1870 GOSUB 5400 'STRIPS NODE NUMBER INSERTED BY THE DCLASS MAINLINE
1880 GOSUB 5400 'GET NEXT DIMENSION
1890 PARAM$(DIAMN%,PRTPARM%)=PARAM$ 'GET DIAMETER WIDTH OR LENGTH
1900 NEXT:NEXT
1910 RETURN
1920 '
1930 PRINT:PRINT'PROCESS CONTROL FILE ';&CONFILE$' ALREADY EXISTS':PRINT 'YOU'LL
HAVE TO DELETE IT TO RUN PROGRAM':PRINT 'USE BASIC KILL '<FILE NAME>' COMMAND OR
GO TO DOS AND DELETE IT':PRINT'THEN START THE PROGRAM AGAIN':END
1940 '
1952 '*****
1954 '
1960 'CREATE MODULE
1970 '
1972 '*****
1974 '
1980 CLS
1990 PRINT SPC(15):'OPTIONS FOR CREATING CNC AND PROCESS CONTROL FILES'
2000 PRINT '*****
*****'
2010 PRINT:PRINT'1. GENERATE LATHE CNC FILE USING GRAPHICS'
2020 PRINT:PRINT'2. GENERATE A PROCESS PLAN CONTROL PLAN FOR A CERTAIN PART NUMB
ER'
2030 PRINT:PRINT'3. RETURN TO MAIN MENU'
2040 PRINT:PRINT:PRINT'CHOOSE 1-3'
2050 ANSWER%=INKEY$:IF ANSWER%='' GOTO 2050
2060 IF ANSWER%='3' THEN RETURN 'LEAVE CREATE
2070 IF ANSWER%='1' THEN GOSUB 2100:GOTO 1970 'CNC CREATE
2080 IF ANSWER%='2' THEN GOSUB 3690:GOTO 1970 'CREATE CON FILE
2090 BEEP:GOTO 2050
2100 '
2110 'CNC CREATE SUBROUTINE W/GRAPHICS ADOPTED FROM A PROGRAM CALLED GRAPH WRITT
EN BY JUSTIN REDD
2120 '
2130 KEY OFF: SCREEN 0,1: COLOR 15,0,0: WIDTH 40: CLS
2140 GOSUB 3630
2150 GOTO 2160
2160 CLS
2170 SCREEN 0,1
2180 COLOR 7
2190 LOCATE 2,6,0: PRINT 'Graphics Utilities'
2200 LOCATE 5,6,0: PRINT '1 - A00'
2210 LOCATE 6,6,0: PRINT '2 - A10'
2220 LOCATE 7,6,0: PRINT '3 - A20'
2230 LOCATE 8,6,0: PRINT '4 - APPLE DOWNLOADER'
2240 LOCATE 10,6,0: PRINT 'ESC - EXIT GRAPHIC UTILITIES'
2250 LOCATE 12
2260 LOCATE 10,6,0: PRINT 'Enter program number'

```

```

1110 IF ANSWER$="N" THEN IF ANSWER$="" GOTO 1150
1120 IF ANSWER$="Y" GOTO 1170 ELSE IF ANSWER$="?" GOTO 1190
1130 PRINT#MUXFZ,CHR$(10);CHR$(4); 'TURN ON PORT 4 SO WILL INITIALIZE FASTER
1140 FOR PORTZ=1 TO 4
1150 MUXMESG1$=CHR$(2) + RIGHT$(STR$(PORTZ),1)
1160 PRINT#MUXFZ,MUXMESG1$;
1170 PRINT"INITIALIZING PORT";PORTZ
1180 FOR XZ = 1 TO 11:INPUT#MUXFZ,A$;NEXT 'LOAD IN MENU FROM MUX
1190 PRINT#MUXFZ,"2"; 'CHOOSE CHANGE BAUD RATE OPTION
1200 FOR XZ = 1 TO 8:INPUT#MUXFZ,A$;NEXT
1210 PRINT#MUXFZ,"5"; 'SELECTS BAUD RATE OF 4800 FOR PORT
1220 FOR XZ = 1 TO 11:INPUT#MUXFZ,A$;NEXT
1230 PRINT#MUXFZ,"1"; 'QUITS CONFIGURATION MENU
1240 NEXT:INPUT#MUXFZ,A$ 'GET LAST MENU RESPONSE
1290 READY$="READY" + CHR$(10)
1300 BAD$="BAD" + CHR$(10)
1310 OK$="OK" + CHR$(10)
1320 RETURN
1330 '
1332 '*****
1334 '
1340 'PROCESS TEMPORARY FILES
1350 '
1352 '*****
1354 '
1360 OPEN "O",#TEMPFZ,"PROCONT.BAT" :CLOSE#TEMPFZ 'NULLIFY CONTENTS OF CONTINUAT
ION BATCH FILE
1370 '
1380 ON ERROR GOTO 1940
1390 OPEN "I",#TEMPFZ,"PROCTEMP.TEX"
1400 IF EOF(TEMPFZ) GOTO 1410 ELSE INPUT#TEMPFZ,PRTNUMB$:CONFILE$=PRTNUMB$+".CO
N":GOTO 1420
1410 ON ERROR GOTO 0:CLOSE#TEMPFZ:RETURN 'NO DCLASS FILES TO POST PROCESS
1420 CLOSE#TEMPFZ
1430 ON ERROR GOTO 0
1440 OPEN "O",#TEMPFZ,"PROCTEMP.TEX":CLOSE#TEMPFZ 'NULLIFY FILE CONTENTS
1450 CLS:PRINT"POST PROCESSING DCLASS FILE"
1460 OPEN "I",#TEMPFZ,"FDC.PRT"
1470 PROCER1Z=1
1480 INPUT#TEMPFZ,INLINE$ 'GET NEXT PROCESS
1490 INPUT#TEMPFZ,EXTRACR1$ 'BLEED OFF EXTRA CARRIAGE RETURNS INSERTED BY DCLAS
S
1500 IF LEFT$(INLINE$,1)="" THEN INLINE$=MID$(INLINE$,2):GOTO 1500 'STRIP LEADI
NG SPACES
1510 GOSUB 5400 'STRIPS NODE NUMBER INSERTED BY THE DCLASS MAINLINE
1520 GOSUB 5400 'GET FIRST PARAMETER
1530 PRINT". ";
1540 IF PROCER1Z=1 THEN PROCER1Z=0:IF VAL(PARAM$)=0 GOTO 1710 ELSE GOTO 1480 'TR
EE TRAVERSAL FLAG IS OFF. NO CNC DATA WAS GENERATED
1550 PARAMID$=VAL(PARAM$) 'GET PARAMTER ID NUMBER
1560 IF PARAMID$=90 THEN GOSUB 1820:GOTO 1480 'READ IN PART DIMENSIONES
1570 IF PARAMID$=99 GOTO 1600 'FINISHED LOADING CNC DATA PARAMETERS
1580 GOSUB 5400:INPARAM$(PARAMID$)=PARAM$ 'SAVE PARAMTER IN PROPER LOCATION
1590 GOTO 1480 'PROCESS NEXT PARAMETER
1600 'SEND PARAMTERS TO APPLE
1610 PARAM$(1,2)=INPARAM$(1) 'PART NAME
1620 PARAM$(2,1)=INPARAM$(2) 'PART NUMBER
1630 PRTFAMLY$=INPARAM$(3) 'PART FAMILY
1640 NUMSDIAM$=INPARAM$(4) 'NUMBER OF DIFFERENT DIAMETERS
1650 PARAM$(12,1)=INPARAM$(5) 'FEED RATE
1660 PARAM$(12,2)=INPARAM$(6) 'CUT DEPTH
1670 PARAM$(12,3)=INPARAM$(7) 'STOCK DIAMETER
1680 PARAM$(12,4)=INPARAM$(8) 'STOCK LENGTH
1690 'GOTO 1690 'DOWNLOADER ROUTINE
1700 'FAST CONTROL WITH NEW NAME

```

```

100 DIM MAXJOBZ=MAXIMUM NUMBER OF PROCESS PLANS TO BE EXECUTED AT SAME TIME IS'
110 JOBZ=1
120 TEMP1%=TEMPF%+1 'COMMUNICATION FILE NUMBER
130 TEMP2%=MUXF%+1 'TEMPORARY FILE AND MISCELANEAUS FILE NUMBER
140 TEMP3%=TEMPF%+1 'SECOND TEMPORARY FILE NUMBER
150 JOBSRUNGZ=0 'NUMBER OF JOBS RUNNING COUNTER
160 DIM RSTATUSZ(MAXJOBZ) 'RUNNING STATUS OF EACH JOB
170 MAYSEQZ=3 'MAX NUMBER OF PROCESS STEPS IN DIRECT SEQUENCE FOR A PARTICULAR M
ALHNE
180 PRINT:PRINT'MAXIMUM NUMBER OF PROCESSES FOR ONE MACHINE IN SEQUENCE WITH THE
HOLD FLAG':PRINT'OFF IN THE SAME PROCESS PLAN IS'MAYSEQZ-1
190 MAXQUESZ=MAXJOBZ * MAYSEQZ 'MAX NUMBER OF QUES FOR EACH MACHINE
200 NUMBMACHZ=8 'NUMBER OF MACHINES
210 PRINT:PRINT'CODE NUMBER FOR EACH MACHINE:'
220 TABZ=5
230 PRINT SPC(TABZ);'1 IS STACKER'
240 PRINT SPC(TABZ);'2 IS MINI-ROBOT'
250 PRINT SPC(TABZ);'3 IS BYU ROBOT'
260 PRINT SPC(TABZ);'4 IS TOYO LATHE'
270 PRINT SPC(TABZ);'5 IS EMCO LATHE'
280 PRINT SPC(TABZ);'6 IS POLAR MILL'
290 PRINT SPC(TABZ);'7 IS WETOFF MILL'
300 PRINT SPC(TABZ);'8 IS PUNCH'
310 DIM JOBSRUNGZ(NUMBMACHZ) 'STATUS OF EACH MACHINE
320 MAXPARAMZ=7 'GREATEST NUMBER OF PARAMETERS
330 DIM MACHINEQUE$(NUMBMACHZ,MAXQUESZ,MAXPARAMZ) 'CREATES A QUE FOR EACH MACHIN
E. WILL HOLD PARAMETERS NECESSARY TO RUN MACHINES
340 HLDZ=2 'POSITON IN ARRAY OF PROCESS HOLD FLAG
350 DIM TEMP$(MAXPARAMZ) 'TEMPRARY HOLDING ARRAY FOR MACHINE QUE PARAMETERS
360 DIM MQPARAMSZ(NUMBMACHZ) 'PARAMETERS NEED FOR EACH MACHINE IN THE QUE NOT IN
CLUDING THE 1ST OR 2ND PARAMETERS
370 MQPARAMSZ(1)=3 'NUMBER OF PARAMETERS USED FOR STACKER
380 MQPARAMSZ(2)=5 'NUMBER OF PARAMETERS USED FOR MINI ROBOT
390 MQPARAMSZ(3)=3 'NUMBER OF PARAMETERS USED FOR BYU ROBOT
400 MQPARAMSZ(4)=3 'NUMBER OF PARAMETERS USED FOR TOYO LATHE
410 MQPARAMSZ(5)=3 'NUMBER OF PARAMETERS USED FOR EMCO LATHE
420 MQPARAMSZ(6)=3 'NUMBER OF PARAMETERS USED FOR POLAR MILL
430 MQPARAMSZ(7)=3 'NUMBER OF PARAMETERS USED FOR WESTOFF MILL
440 MQPARAMSZ(8)=3 'NUMBER OF PARAMETERS USED FOR PUNCH
450 '0 IS USED TO SHOW END OF PROCESS CON FILE
460 DIM MQNEXTZ(NUMBMACHZ) 'HOLDS NUMBER OF NEXT POSTION IN QUE TO BE FILLED
470 DIM MQPOINTERZ(NUMBMACHZ) 'HOLDS NUMBER OF NEXT POSITION INQUE TO BE EXECUTE
D
480 FOR COUNTERZ=1 TO NUMBMACHZ:MQNEXTZ(COUNTERZ)=1:MQPOINTERZ(COUNTERZ)=1:NEXT
COUNTERZ
490 'INITIALIZE QUE POINTERS TO 1ST QUE
500 DIM HOLDINGZ(NUMBMACHZ,MAXJOBZ,2) 'MACHINE HOLDING POINTERS
510 HJOBZ=1 'POSITION IN ARRAY OF JOB FLAG
520 HQUEZ=2 'OF QUE POINTER
530 DIM HELDZ(NUMBMACHZ) 'HOLDS JOB NUMBER WHEN A MACHINE IS BEING HELD IDLE BY
ANDHER MACHINE THAT IS WAITING FOR IT TO BE READY
540 DIM RSTATUSZ(63) 'STATUS FLAG OF EACH BIN IN STACKER
550 DIM JSTATUSZ(2) 'STATUS FLAG OF 2 TRAYS
560 DIM JSTATUSZ(NUMBMACHZ,2) 'JOB STATUS OF EACH MACHINE. 2 JOBS FOR STACKER
570 QUECHECKZ=1 'WILL START WITH MACHINE 1 WHEN CHECKING MACHINE QUES
580 DIM PORT$(NUMBMACHZ) 'HOLDS PORT NUMBER FOR EACH MACHINE
590 PRINT:PRINT'MULTIPORT CONTROLLER PORT ALLOCATION:'
600 PORT$(1)='1':PRINT'PORT FOR STACKER IS 1' 'PORT FOR MACHINE 1 - STACKER
610 PORT$(2)='2':PRINT'PORT FOR MINI MOVER ROBOT IS 2'
620 PORT$(4)='3':PRINT'PORI FOR TOYO LATHE IS 3'
630 COLOR 4:PRINT:PRINT 'CAUTION - If above values are exceeded program may
crash'
640
650 INITIALIZE MUL
660
670 'CALL ASAC-R-7.2-CS4000-IS4000' AS $MUXF:PRINT:PRINT'DOES MULTIFLEYO
680 'CALL 10-10-10-10-10-10 ? (Y)':COLOR 7:

```

```

110 *          PART PROCESSOR BY GREGORY J. PETO          *
120 *          *                                          *
130 *          THIS PROGRAM CONTROLS THE CREATION,      *
140 *          MODIFICATION AND EXECUTION OF PART PROCESSES *
150 *          *                                          *
152 *****
154
160 ' This program is the main process controlling program. It runs in
170 ' 3 modes, create, modify and execute. The modify portion has not been
180 ' implemented yet. The create mode has several options. You can create a
190 ' part process by using the graphics routines written in basic or you can
200 ' select the option to leave basic and create a part process using the
210 ' DCLASS mainline TREE. Upon reentry to basic the file created by DCLASS
220 ' is postprocessed and a .con file is created which is then executable.
222 '
223 ' The execute mode executes process control files. The basic syntax is:
224 ' <machine #> <process hold code> <machine parameter list> <cr>
225 ' The program sets up a queue for each machine. Requests from each process
226 ' being executed are placed in the queue for that machine. As the machine
227 ' becomes available the next item in it's queue is sent to the machine.
228 *****
250
260 'MAIN BODY OF PROGRAM
270
280 GOSUB 480 'INITIALIZE CONSTANTS
290 GOSUB 1330 'PROCESS TEMPORARY FILES
300 GOSUB 360 'GET MODE OF OPERATION
304 ' GO INTO MODE SELECTED
310 ON MODE% GOSUB 1970,440,3810 'CREATE,MODIFY,EXECUTE ROUTINES
320 GOTO 300 'GET MODE SELECTION AGAIN
330
340 *****
350
360 'GET MODE OF OPERATION
370
380 CLS:PRINT SPC(19);"MODES OF OPERATION FOR PART PROCESS PLANS":PRINT"*****"
*****
390 PRINT:PRINT"1. CREATE A PROCESS":PRINT:PRINT"2. MODIFY A PROCESS":PRINT:PRINT
T"3. EXECUTE PROCESS":ANSWER$=""
400 LOCATE 20:PRINT"CHOOSE 1-3":ANSWER$=INKEY$:IF ANSWER$="" THEN GOTO 400
410 MODE%=VAL(ANSWER$):IF MODE% <= 0 OR MODE% >3 THEN BEEP:GOTO 400
420 RETURN
430
440 'MODIFY ROUTINE NOT IMPLEMENTED YET
450 RETURN
460
462 *****
464
470 'INITIALIZE AND DISPLAY CONSTANTS
480
484 *****
486
490 WIDTH 80:SCREEN 0:CLS
500 OPTION BASE 1
510 DEBUG%=0:PRINT"DEBUG MODE ==>";DEBUG%
520
530 'VARIABLES USED BY CREATE ROUTINE
540
550 DIM PARAM$(12*2)
560 PROCES% = NEW PROCESS FLAG
570 READY% = MACHINE READY FOR NEXT STEP FLAG
580 MAXDESS=10 'MAXIMUM NUMBER OF JOBS RUNNING AT ONE TIME

```

DESIGN AND MODIFICATION

PROCESS CREATOR

1. Process Planning
2. Part parametric description generation

PART PROGRAMMING

1. Machine dedicated CNC code generators
2. Manual programming

EXECUTION

PROCESS EXECUTOR

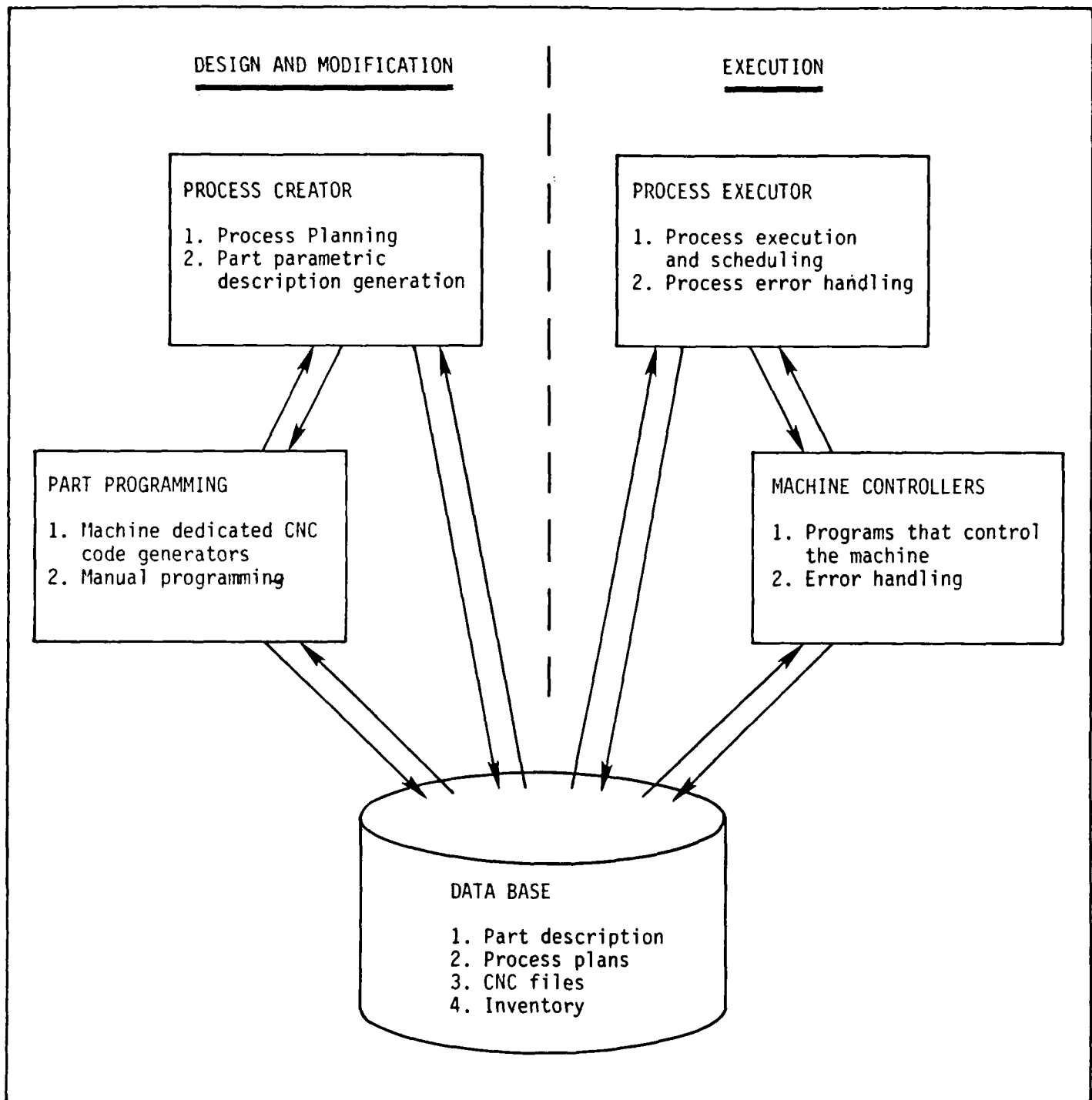
1. Process execution and scheduling
2. Process error handling

MACHINE CONTROLLERS

1. Programs that control the machine
2. Error handling

DATA BASE

1. Part description
2. Process plans
3. CNC files
4. Inventory



APPENDIX M

CAD/CAM Demonstration Software Listings

```
2190 REM * HOOK BACK ONE INCH
2200 REM *****
2210 REM
2220 CY3$="P 1600"
2230 FOR I = 1 TO 2
2240 Q$=USR3(CY3$)
2250 NEXT I
2260 RETURN
2270 REM
2280 REM *****
2290 REM * HOOK ALL THE WAY BACK
2300 REM *****
2310 REM
2320 CY3$="P 16000"
2330 FOR I = 1 TO 2
2340 Q$=USR3(CY3$)
2350 NEXT I
2360 RETURN
2370 REM
```

```

1630 REM * GOTO BIN COORDINATES
1640 REM *****
1650 REM
1660 CY1$="P" + STR$(BIN(XBIN,1))
1670 IF OPERATION = 1 THEN CY2$="P" + STR$( BIN(XBIN,2) + 385 )
1680 IF OPERATION = 0 THEN CY2$="P" + STR$( BIN(XBIN,2) )
1690 FOR I = 1 TO 2
1700 Q$=USR1(CY1$):Q$=USR2(CY2$)
1710 NEXT I
1720 RETURN
1730 REM
1740 REM *****
1750 REM * GOTO TRAY COORDINATES
1760 REM *****
1770 REM
1780 CY1$="P" + STR$(TRAY(XTRAY,1))
1790 IF OPERATION = 1 THEN CY2$="P" + STR$(TRAY(XTRAY,2))
1800 IF OPERATION = 0 THEN CY2$="P" + STR$( TRAY(XTRAY,2) + 385 )
1810 FOR I = 1 TO 2
1820 Q$=USR1(CY1$):Q$=USR2(CY2$)
1830 NEXT I
1840 RETURN
1850 REM
1860 REM *****
1870 REM * HOOK FORWARD
1880 REM *****
1890 REM
1900 CY3$="P 0"
1910 FOR I = 1 TO 2
1920 Q$=USR3(CY3$)
1930 NEXT I
1940 RETURN
1950 REM
1960 REM *****
1970 REM * ENGAGE HOOK
1980 REM *****
1990 REM
2000 IF FLAG$ = "BIN" THEN CY2$="P" + STR$( BIN(XBIN,2) )
2010 IF FLAG$ = "TRAY" THEN CY2$="P" + STR$( TRAY(XTRAY,2) )
2020 FOR I = 1 TO 2
2030 Q$=USR2(CY2$)
2040 NEXT I
2050 RETURN
2060 REM
2070 REM *****
2080 REM * DISENGAGE HOOK
2090 REM *****
2100 REM
2110 IF FLAG$="BIN" THEN CY2$="P" + STR$( BIN(XBIN,2) + 385 )
2120 IF FLAG$="TRAY" THEN CY2$="P" + STR$( TRAY(XTRAY,2) + 385 )
2130 FOR I = 1 TO 2
2140 Q$=USR2(CY2$)
2150 NEXT I
2160 RETURN
2170 REM
2180 REM *****

```

```

1050 REM
1060 REM *****
1070 REM * GOTO OPERATION# ROUTINE
1080 REM *****
1090 REM
1100 IF XTRAY < > 1 AND XTRAY < > 2 THEN PRINT "BAD TRAY# PASSED":GOTO 760
1110 IF XBIN > 63 OR XBIN < 11 THEN PRINT "BAD BIN# PASSED":GOTO 760
1120 N$=RIGHT$(STR$(XBIN),1)
1130 IF N$ <> "1" AND N$ <> "2" AND N$ <> "3" THEN PRINT "BAD BIN# PASSED":GOTO
60
1140 ON OPERATION + 1 GOTO 1150,1310,1470
1150 REM *****
1160 REM * RETURN BIN
1170 REM *****
1180 REM
1190 GOSUB 1730
1200 GOSUB 1860
1210 FLAG$="TRAY"
1220 GOSUB 1960
1230 GOSUB 2270
1240 GOSUB 1620
1250 GOSUB 1860
1260 FLAG$="BIN"
1270 GOSUB 2070
1280 GOSUB 2170
1290 GOTO 760
1300 REM
1310 REM *****
1320 REM * RETRIEVE BIN
1330 REM *****
1340 REM
1350 GOSUB 1620
1360 GOSUB 1860
1370 FLAG$="BIN"
1380 GOSUB 1960
1390 GOSUB 2270
1400 GOSUB 1730
1410 GOSUB 1860
1420 FLAG$="TRAY"
1430 GOSUB 2070
1440 GOSUB 2170
1450 GOTO 760
1460 REM
1470 REM *****
1480 REM * TERMINATE PROGRAM
1490 REM *****
1500 REM
1510 CY1$="P 0":CY2$="P 16795":CY3$="P 0"
1520 FOR I= 1 TO 2
1530 Q$=USR1(CY1$)
1540 Q$=USR2(CY2$)
1550 NEXT I
1590 HOME:PRINT "STACKER PROGRAM TERMINATED"
1600 PRINT:PRINT:END
1610 REM
1620 REM *****

```

```

560 TRAY(2,1)=2025:TRAY(2,2)=16350
570 REM
580 CY$="I,R 253,S 1,F 1,A,P 1,P 0"
585 MOTOR1$="I,R 251,S 1,F 1,A,P 1,P 0"
586 MOTOR1$=MOTOR1$+" "
590 CY$=CY$+" "
600 CY1$="XXX"+" ":CY2$="XXX"+" ":CY3$="XXX"+" "
610 FOR I = 1 TO 2
620 Q$=USR1(MOTOR1$)
630 Q$=USR2(CY$)
640 Q$=USR3(CY$)
650 NEXT I
660 REM
670 REM *****
680 REM*MOVE CARRIER TO TOP AND REINITIALIZE
690 REM *****
700 REM
715 MOTOR2$="N 16795,-,G,A" 'MOVE CARRIER TO TOP"
716 Q$=USR2(MOTOR2$)
720 HOOK$="N 1600,-,G,A"
724 Q$=USR3(HOOK$) 'MOVE HOOK FORWARD ONE INCH AND SET AS HOME POSITION
730 GOSUB 2200 'MOVE HOOK BACK ONE INCH
745 Q$=USR8(IBM$)
746 W$="WHAT?"
747 Q$=USR7(W$)
750 REM
760 REM *****
770 REM * INPUT PARAMETERS FROM IBM
780 REM *****
790 REM
800 Q$="XXX" + " ":OPERATION$="XXX"+" ":XBIN$="XXX"+" ":XTRAY$="XXX"+" "
810 OK$="OK"+CHR$(13):WHAT$="WHAT?"+CHR$(13):READY$="READY"+CHR$(13)
815 Q$=USR7(READY$)
820 PRINT "OPERATION# = ";
840 Q$=USR8(OPERATION$):PRINT OPERATION$
850 IF OPERATION$ <> "0" AND OPERATION$ <> "1" AND OPERATION$ <> "2" THEN PRINT
  "INVALID OPERATION#":Q$=USR7(WHAT$):PRINT:GOTO 820
860 OPERATION = VAL(OPERATION$)
870 IF OPERATION = 2 THEN GOTO 1470
880 Q$=USR7(OK$)
890 PRINT "BIN# = ";
900 Q$=USR8(XBIN$):PRINT XBIN$
910 DATA 11,12,13,21,22,23,31,32,33,41,42,43,51,52,53,61,62,63
920 FOR I = 1 TO 18
930 READ X
940 IF VAL(XBIN$)=X THEN RESTORE: GOTO 1000
970 NEXT I
980 RESTORE:Q$=USR7(WHAT$):PRINT "BAD BIN#":PRINT:GOTO 890
1000 XBIN=VAL(XBIN$)
1005 Q$=USR7(OK$)
1010 PRINT "TRAY# = ";
1020 Q$=USR8(XTRAY$):PRINT XTRAY$
1025 IF XTRAY$ <> "1" AND XTRAY$ <> "2" THEN PRINT "BAD TRAY#":PRINT:Q$=USR7(WHA
T$):GOTO 1010
1030 XTRAY=VAL(XTRAY$)
1040 PRINT:PRINT "-----":PRINT

```

```

110 REM *****
20 REM * STORAGE AND RETRIEVAL SYSTEM CONTROLLER
30 REM * -BYU MINI-CAM LAB-
40 REM * ORIGINAL APPLESOFT VERSION BY DAVE JESPERSON
50 REM * MICROSOFT PROGRAM BY JUSTIN REDD
60 REM * OCTOBER, 1983
70 REM *
80 REM * WHEN RUN, THIS PROGRAM WILL WAIT FOR 3
90 REM * PARAMETERS TO BE PASSED TO IT THROUGH
100 REM * THE RS232 LINK. THESE THREE PARAMETERS
110 REM * ARE DEFINED AS THE OPERATION#,BIN#,AND
120 REM * TRAY#. OPERATION# 0 = RETURN BIN
130 REM * OPERATION# 1 = RETRIEVE BIN
140 REM * OPERATION# 2 = QUIT PROGRAM
150 REM *
160 REM * TO RUN THIS PROGRAM CYDRIVER.HEX AND COMM.HEX
170 REM * MUST BE LOADED INTO MEMORY.
180 REM *
190 REM *****
200 REM
210 HOME:PRINT " APPLE STACKER PROGRAM":PRINT:PRINT"-----"
-----
220 PRINT:PRINT
230 REM *****
240 REM * LOAD CYDRIVER AND COMM
250 REM *****
260 REM
270 CALLER$="STACKER.BAS":CLINE=300:PRGM$="XXXX":SLINE=999:HFILE$="XXXX"
280 COMMON CALLER$,CLINE,PRGM$,SLINE,HFILE$
290 CHAIN "LOADASM.BAS"
300 REM
310 REM *****
320 REM * INITIALIZE
330 REM *****
340 REM
350 DIM BIN(63,2)
360 DIM TRAY(2,2)
370 BIN(11,1)=4085:BIN(11,2)=16375
380 BIN(12,1)=4138:BIN(12,2)=10750
390 BIN(13,1)=4138:BIN(13,2)=4925
400 BIN(21,1)=6140:BIN(21,2)=16375
410 BIN(22,1)=6140:BIN(22,2)=10675
420 BIN(23,1)=6212:BIN(23,2)=4775
430 BIN(31,1)=8200:BIN(31,2)=16375
440 BIN(32,1)=8269:BIN(32,2)=10775
450 BIN(33,1)=8262:BIN(33,2)=4875
460 BIN(41,1)=10250:BIN(41,2)=16375
470 BIN(42,1)=10300:BIN(42,2)=10750
480 BIN(43,1)=10312:BIN(43,2)=4925
490 BIN(51,1)=12287:BIN(51,2)=16375
500 BIN(52,1)=12312:BIN(52,2)=10800
510 BIN(53,1)=12344:BIN(53,2)=4850
520 BIN(61,1)=14320:BIN(61,2)=16375
530 BIN(62,1)=14337:BIN(62,2)=10775
540 BIN(63,1)=14350:BIN(63,2)=4875
550 TRAY(1,1)=0:TRAY(1,2)=16350

```

```

3982 *****
3984
3986 'CLEAR SCREEN
3988
3990 CLS:PRINT SPC(30);"EXECUTE PROCESS PLAN":PRINT"*****
*****":RETURN
4000
4010 'CHECK FOR OPERATOR INPUT
4020
4030 PRINT:PRINT:COLOR 4,0:PRINT'DO YOU WANT MAKE OPERATOR INPUT ? (YES IS ANY K
EY)":COLOR 7,0:PRINT
4040 'FOR HOLDZ=1 TO 500
4050 A$=INKEY$:IF A$ <> "" GOTO 4100 'INITIATE NEW PROCESS PLAN
4060 IF JOBSRUNGZ=0 GOTO 4050
4070 'NEXT
4080 NEPROCZ=0:RETURN 'NO NEW PROCESS
4090
4100 'GET OPERATOR INPUT
4110
4120 GOSUB 3990:PRINT:FILES "*.CON" 'CLEAR SCREEN
4130 PRINT:PRINT:PRINT 'ENTER A PART NUMBER TO EXECUTE':PRINT SPC(TABZ);"(OR C T
O CANCEL)":PRINT SPC(TABZ);"(OR Q TO LEAVE EXECUTE ROUTINE AND RETURN TO MAIN ME
NU)":PRINT SPC(TABZ);"(OR F TO FIX A PROCESS WHERE THE MUX MISSED A FINISHED SIG
NAL)"
4140 PRINT:INPUT " ==> ",ANSWER$
4150 IF ANSWER$="C" THEN RETURN ELSE IF ANSWER$="Q" THEN EXITEXCTZ=1:RETURN
'DO EXITING COMMANDS
4160 IF ANSWER$="F" THEN PRINT:PRINT 'ENTER NUMBER OF MACHINE FINISHED OR C TO C
ANCEL':PRINT SPC(TABZ);"(USE PROPER MACHINE CODE NUMBER OR PROGRAM WILL CRASH)":
INPUT " ==> ",ANSWER$:IF ANSWER$="C" GOTO 4080 ELSE MFIXZ=1
4170 IF MFIXZ=1 THEN MACHINEZ=VAL(ANSWER$):IF JOBSRUNGZ(MACHINEZ)=0 THEN PRINT'M
O JOBS RUNNING ON MACHINE':FOR HOLDZ=1 TO 1000:NEXT:GOTO 4110 ELSE GOSUB 4570:RE
TURN 'EMULATE A FINISHED SIGNAL (CNTRL-A) COMING IN FROM APPLE THAT MUX MISSED
4180 'INITIATE NEW PROCESS
4190 IF JOBSRUNGZ = MAXJOBSZ THEN PRINT:PRINT'MAXIMUM NUMBER JOBS OF';MAXJOBSZ;'
IS ALREADY RUNNING':INPUT 'TYPE RETURN TO CONTINUE',HOLD$:NEPROCZ=0:RETURN 'MAX
NUMBER OF JOBS ALREADY RUNNING
4200 FOR COUNTERZ=1 TO MAXJOBSZ
4210 IF RSTATUSZ(COUNTERZ) = 0 THEN JOBZ=COUNTERZ:RSTATUSZ(JOBZ)=1:COUNTERZ=MAXJ
OBSZ 'SETS RUNNING STATUS FLAG
4220 NEXT
4230 PRTNUMB$(JOBZ)=ANSWER$
4240 JOBSRUNGZ=JOBSRUNGZ + 1 'INCREMENT NUMBER OF JOBS RUNNING
4250 ON ERROR GOTO 4300
4260 OPEN "I".#JOBZ,(PRTNUMB$(JOBZ) + ".CON")
4270 NEPROCZ=1 'SET NEW PROCESS FLAG
4280 ON ERROR GOTO 0
4290 RETURN
4300 'FILE DOESN'T EXIST
4310 PRINT:PRINT:PRINT'AN EXECUTABLE PROCESS PLAN FILE DOESN'T EXIST FOR PART NU
MBER "PRTNUMB$(JOBZ)
4320 PRINT:PRINT'TYPE A TO TRY AGAIN'
4330 ANSWER$=INKEY$:IF ANSWER$="" GOTO 4330
4340 IF ANSWER$="A" THEN RESUME 4100 'GET OPERATOR INPUT
4350 GOSUB 3990
4360 CLOSE #JOBZ:JOBSRUNGZ=JOBSRUNGZ - 1:RSTATUSZ(JOBZ)=0:NEPROCZ=0:RESUME 4370
'NO NEW PROCESS, RESET FLAGS, RESUME NECCESARY TO END ERROR TRAPPING ROUTINE
4370 RETURN
4380
4390 'CHECK FOR MACHINE FINISHED
4400
4410 PRINT'CHECKING FOR A MACHINE THAT FINISHED LAST PROCESS'
4420 'CHECK MUX FOR MACHINE FINISHED

```

```

4430 IF LOC(MUXF%) < 2 THEN PRINT #MUXF%,CHR$(1) + '0': 'SEND DISCONNECT CURRENT
4440 IF NO MACHINE HAS FINISHED
4450 HOLD%=1 TO 1500:NEXT 'WAIT TO FILL BUFFER
4460 IF LOC(MUXF%) < 2 THEN PRINT'NO MACHINES FINISHED':GOTO 4630 'NO MACHINES R
ECENTLY FINISHED
4470 MUX%=INPUT$(LOC(MUXF%),#MUXF%) 'GET MUX RESPONSE
4480 PRINT'MUX HAS ==>';MUX%
4490 FOR PORT%=1 TO 4
4500 PORT%=RIGHT$(STR$(PORT%),1)
4510 X=INSTR(MUX%,PORT%)
4520 IF X <> 0 GOTO 4550
4530 NEXT
4540 'IF FRSTCHECK%=1 THEN FRSTCHECK=0:GOTO 1250 'START SECOND CHECK OF MUX LOOK
ING FOR MACHINES
4550 GOTO 4630 'NO MACHINES RECENTLY FINISHED
4560 PRINT'MACHINE READY ON PORT ';PORT%
4570 ON PORT% GOSUB 4930,4930,4930,4930 'FIND MACHINE REQUESTING PORT
4580 IF MQPOINTER%(MACHINE%)=1 THEN JOB%=VAL(MACHINEQUE%(MACHINE%,MAXQUES%,1)) E
LSE JOB%=VAL(MACHINEQUE%(MACHINE%,MQPOINTER%(MACHINE%) - 1,1)) 'FIND JOB WAS LAS
T WORKING ON
4590 MACHRDY%=1 'SET MACHINE READY FLAG
4600 NONEFIN%=0 'TURN OFF NO PROCESSES FINISHED FLAG
4610 JOBSRUNG%(MACHINE%)=0 'TURN OFF MACHINE BUSY FLAG
4620 RETURN
4630 'CHECK FOR MACHINE QUE READY
4640 '
4650 STFCHECK%=QUECHECK%
4660 IF DEBUG%=2 THEN PRINT'QUECHECK% ==>';QUECHECK%
4670 IF MQPOINTER%(QUECHECK%) = MNEXT%(QUECHECK%) GOTO 4780 'CHECK FOR WAITING
QUES
4680 IF JOBSRUNG%(QUECHECK%)=1 GOTO 4780 'MACHINE BUSY
4690 'A QUE IS READY AND MAYBE A MACHINE
4700 MACHINE%=QUECHECK%
4710 JOB%=VAL(MACHINEQUE%(MACHINE%,MQPOINTER%(MACHINE%),1))
4720 QUECHECK%=QUECHECK%+1 'SETUP TO CHECK NEXT MACHINE'S QUE AFTER THIS ONE
4730 NONEFIN%=1
4740 MACHRDY%=1
4750 PRINT 'CHECKING QUE OF MACHINE';MACHINE%;' FOR EXECUTABLE PROCESS'
4760 RETURN
4770 '
4780 'MACHINE'S QUE EMPTY
4790 '
4800 QUECHECK%=QUECHECK%+1 'SETUP TO CHECK NEXT MACHINE'S QUE AFTER THIS ONE
4810 IF QUECHECK% > NUMBMACH% THEN QUECHECK% = 1
4820 IF STFCHECK% <> QUECHECK% GOTO 4670 'CHECK TO SEE IF HAVE CHECKED ALL MACHI
NES QUES
4830 'NO MACHINES OR JOBS ARE READY FOR NEXT PROCESS
4840 MACHRDY%=0
4850 RETURN
4860 '
4870 'TURN ON PORT
4880 PRINT #MUXF%,CHR$(1) + PORT%(MACHINE%); 'TURN ON PORT
4890 PRINT'PORT ';PORT%(MACHINE%);' TURNED ON'
4900 FOR HOLD%=1 TO 1500:NEXT:IF LOC(MUXF%)=0 THEN PRINT'HAVING PROBLEMS RECEIVI
NG FROM PORT ';PORT%(MACHINE%):GOTO 4900 'WAIT FOR BUFFER TO FILL WITH GARBAGE
4910 A$=INPUT$(LOC(MUXF%),#MUXF%) 'EMPTIES BUFFER
4920 RETURN
4930 '
4940 'ONE MACHINE PER PORT
4950 '
4960 IF PORT%=1 THEN MACHINE%=1 'STACNER
4970 IF PORT%=2 THEN MACHINE%=2 'MINI-ROBOT
4980 IF PORT%=3 THEN MACHINE%=3 'TOTO LATHE
4990 IF PORT%=4 THEN PRINT'ERROR IN PORT NUMBER'
5000 RETURN

```

NEXT PROCESS IN MACHINE QUE

```

5000 IF MQUEPTR=1 THEN GOTO 5080 'STARTING NEW PROCESS
5010 IF MQPTRZ(MACHINEZ)=1 THEN PARAMZ=MAXQUEZ ELSE PARAMZ=MQPTRZ(MACHI
5020 - 1 'FIND POSITION IN QUE OF LAST PROCESS EXECUTED
5030 IF VAL(MACHINEQUE$(MACHINEZ,PARAMZ,2)) = 0 THEN PRINT 'NEXT PROCESS IS ALRE
5040 AD IN QUE':RETURN
5070 IF EOF(JOBZ) THEN JOBSRUNGZ=JOBSRUNGZ - 1:RSTATUSZ(JOBZ)=0:CLOSE JOBZ:MACHR
5080 DZ=0:PRINT'FINISHED PROCESSING PART NUMBER ';PRTNUM$(JOBZ):RETURN 'PROCESS PLA
N FINISHED
5080 'GET NEXT PROCESS
5090 INPUT #JOBZ,INLINE$ 'GET NEXT PROCESS
5100 INPUT #JOBZ,EXTRACR1$ 'BLEED OFF EXTRA CARRIAGE RETURNS INSERTED BY DCLASS
5110 IF LEFT$(INLINE$,1)=' ' THEN INLINE$=MID$(INLINE$,2):GOTO 5110 'STRIP LEADI
NG SPACES
5120 GOSUB 5400 'STRIPS NODE NUMBER INSERTED BY THE DCLASS MAINLINE
5130 'PRINT'NODE ==>':PARAM$
5140 GOSUB 5400:MACHINEZ=VAL(PARAM$) 'GET MACHINE NUMBER
5150 PRINT'LOADING NEXT PROCESS FOR PART NUMBER ';PRTNUM$(JOBZ)
5160 IF HOLDINGZ(MACHINEZ,JOBZ,HJOBZ)<>0 THEN SWAP HOLDINGZ(MACHINEZ,JOBZ,HQUEZ)
, MQNEXTZ(MACHINEZ) 'PUT NEXT PROCESS IN QUE POSITION THAT WAS SAVED FOR IT
5170 MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ),1)=STR$(JOBZ) 'SAVE JOB NUMBER WITH
PROCESS
5180 GOSUB 5400:PROCHOLDZ=VAL(PARAM$) 'LOAD PROCESS HOLD FLAG FROM FILE
5190 MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ),2)=PARAM$
5200 'LOAD PROCESS PARAMETERS FROM CON FILE
5210 PRINT'JOB ';JOBZ;' MACHINE';MACHINEZ;' HOLD FLAG VALUE IS';PROCHOLDZ
5220 FOR PARAMZ=3 TO MQPARAMSZ(MACHINEZ) + 2 '1ST 2 POSITINS OF CON FILE ARE MAC
HINE NUMBER AND PROCESS HOLD FLAG. THE FOLLOWING POSITIONS ARE FOR PARAMETERS F
OR THAT MACHINE
5230 IF PARAMZ > 5 THEN INLINE$=EXTRACR1$ 'DCLASS INSERTS A LF AND CR AFTER THE
5TH PARAMTER IT OUTPUTS AND NEED TO GET AGAIN
5240 GOSUB 5400:MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ),PARAMZ)=PARAM$
5250 PRINT'PARAM #';PARAMZ-2;' IS ';MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ),PARAM
Z)
5260 IF ASC(PARAM$) < 58 AND VAL(PARAM$)=0 THEN PARAMZ=MQPARAMSZ(MACHINEZ) + 2 '
A PARAMTER OF ZERO SIGNALS NO MORE PARAMETERS FOLLOWING FOR THIS PROCESS
5270 IF VAL(MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ),PARAMZ))=99 THEN WAITFLAGZ=1:
WAITPOSZ=PARAMZ
5280 NEXT
5290 IF DEBUGZ <> 0 THEN INPUT 'TYPE RETURN TO CONTINUE',HOLD$
5300 IF HOLDINGZ(MACHINEZ,JOBZ,HJOBZ)=1 THEN HOLDINGZ(MACHINEZ,JOBZ,HJOBZ)=0 'TU
RN OFF FLAG THAT MACHINE IS HOLDING BECAUSE THE NEXT PROCESS WAS JUST LOADED FOR
IT
5310 IF WAITFLAGZ=1 THEN MACHHOLDINGZ=VAL(MACHINEQUE$(MACHINEZ,MQNEXTZ(MACHINEZ)
,WAITPOSZ+1)) 'FIND MACHINE WAITING FOR
5320 MQNEXTZ(MACHINEZ)=MQNEXTZ(MACHINEZ) + 1:IF MQNEXTZ(MACHINEZ) > MAXQUEZ THE
N MQNEXTZ(MACHINEZ)=1
5330 IF WAITFLAGZ=1 THEN MACHINEQUE$(MACHHOLDINGZ,MQNEXTZ(MACHHOLDINGZ),1)=STR$(
JOBZ):MACHINEQUE$(MACHHOLDINGZ,MQNEXTZ(MACHHOLDINGZ),2)='99':MACHINEQUE$(MACHHOL
DINGZ,MQNEXTZ(MACHHOLDINGZ),3)=STR$(MACHINEZ):MACHINEZ=MACHHOLDINGZ:WAITFLAGZ=2:
GOTO 5320
5340 'ABOVE STATEMENT SAVES HOLD VALUE IN QUE AND THEN INCREMENTS QUE POINTER
5350 IF WAITFLAGZ=2 THEN HOLDINGZ(MACHINEZ,JOBZ,HJOBZ)=1:HOLDINGZ(MACHINEZ,JOBZ,
HQUEZ)=MQNEXTZ(MACHINEZ):WAITFLAGZ=0:GOTO 5320 'SAVE QUE POSITION OF NEXT PROCES
S FOR THE HELD MACHINE FROM THIS JOB AND THEN INCREMENT QUE POINTER TO LEAVE THE
QUE EMPTY
5360 IF PROCHOLDZ=0 THEN GOTO 5080 'IF PROCESS HOLD FLAG NOT SET THEN LOAD IN NE
XT PROCESS
5370 RETURN
5380 '
5390 '
5400 '
5410 '
5420 '
5430 '
5440 '
5450 '
5460 '
5470 '
5480 '
5490 '
5500 '
5510 '
5520 '
5530 '
5540 '
5550 '
5560 '
5570 '
5580 '
5590 '
5600 '
5610 '
5620 '
5630 '
5640 '
5650 '
5660 '
5670 '
5680 '
5690 '
5700 '
5710 '
5720 '
5730 '
5740 '
5750 '
5760 '
5770 '
5780 '
5790 '
5800 '
5810 '
5820 '
5830 '
5840 '
5850 '
5860 '
5870 '
5880 '
5890 '
5900 '
5910 '
5920 '
5930 '
5940 '
5950 '
5960 '
5970 '
5980 '
5990 '
6000 '
6010 '
6020 '
6030 '
6040 '
6050 '
6060 '
6070 '
6080 '
6090 '
6100 '
6110 '
6120 '
6130 '
6140 '
6150 '
6160 '
6170 '
6180 '
6190 '
6200 '
6210 '
6220 '
6230 '
6240 '
6250 '
6260 '
6270 '
6280 '
6290 '
6300 '
6310 '
6320 '
6330 '
6340 '
6350 '
6360 '
6370 '
6380 '
6390 '
6400 '
6410 '
6420 '
6430 '
6440 '
6450 '
6460 '
6470 '
6480 '
6490 '
6500 '
6510 '
6520 '
6530 '
6540 '
6550 '
6560 '
6570 '
6580 '
6590 '
6600 '
6610 '
6620 '
6630 '
6640 '
6650 '
6660 '
6670 '
6680 '
6690 '
6700 '
6710 '
6720 '
6730 '
6740 '
6750 '
6760 '
6770 '
6780 '
6790 '
6800 '
6810 '
6820 '
6830 '
6840 '
6850 '
6860 '
6870 '
6880 '
6890 '
6900 '
6910 '
6920 '
6930 '
6940 '
6950 '
6960 '
6970 '
6980 '
6990 '
7000 '
7010 '
7020 '
7030 '
7040 '
7050 '
7060 '
7070 '
7080 '
7090 '
7100 '
7110 '
7120 '
7130 '
7140 '
7150 '
7160 '
7170 '
7180 '
7190 '
7200 '
7210 '
7220 '
7230 '
7240 '
7250 '
7260 '
7270 '
7280 '
7290 '
7300 '
7310 '
7320 '
7330 '
7340 '
7350 '
7360 '
7370 '
7380 '
7390 '
7400 '
7410 '
7420 '
7430 '
7440 '
7450 '
7460 '
7470 '
7480 '
7490 '
7500 '
7510 '
7520 '
7530 '
7540 '
7550 '
7560 '
7570 '
7580 '
7590 '
7600 '
7610 '
7620 '
7630 '
7640 '
7650 '
7660 '
7670 '
7680 '
7690 '
7700 '
7710 '
7720 '
7730 '
7740 '
7750 '
7760 '
7770 '
7780 '
7790 '
7800 '
7810 '
7820 '
7830 '
7840 '
7850 '
7860 '
7870 '
7880 '
7890 '
7900 '
7910 '
7920 '
7930 '
7940 '
7950 '
7960 '
7970 '
7980 '
7990 '
8000 '
8010 '
8020 '
8030 '
8040 '
8050 '
8060 '
8070 '
8080 '
8090 '
8100 '
8110 '
8120 '
8130 '
8140 '
8150 '
8160 '
8170 '
8180 '
8190 '
8200 '
8210 '
8220 '
8230 '
8240 '
8250 '
8260 '
8270 '
8280 '
8290 '
8300 '
8310 '
8320 '
8330 '
8340 '
8350 '
8360 '
8370 '
8380 '
8390 '
8400 '
8410 '
8420 '
8430 '
8440 '
8450 '
8460 '
8470 '
8480 '
8490 '
8500 '
8510 '
8520 '
8530 '
8540 '
8550 '
8560 '
8570 '
8580 '
8590 '
8600 '
8610 '
8620 '
8630 '
8640 '
8650 '
8660 '
8670 '
8680 '
8690 '
8700 '
8710 '
8720 '
8730 '
8740 '
8750 '
8760 '
8770 '
8780 '
8790 '
8800 '
8810 '
8820 '
8830 '
8840 '
8850 '
8860 '
8870 '
8880 '
8890 '
8900 '
8910 '
8920 '
8930 '
8940 '
8950 '
8960 '
8970 '
8980 '
8990 '
9000 '
9010 '
9020 '
9030 '
9040 '
9050 '
9060 '
9070 '
9080 '
9090 '
9100 '
9110 '
9120 '
9130 '
9140 '
9150 '
9160 '
9170 '
9180 '
9190 '
9200 '
9210 '
9220 '
9230 '
9240 '
9250 '
9260 '
9270 '
9280 '
9290 '
9300 '
9310 '
9320 '
9330 '
9340 '
9350 '
9360 '
9370 '
9380 '
9390 '
9400 '
9410 '
9420 '
9430 '
9440 '
9450 '
9460 '
9470 '
9480 '
9490 '
9500 '
9510 '
9520 '
9530 '
9540 '
9550 '
9560 '
9570 '
9580 '
9590 '
9600 '
9610 '
9620 '
9630 '
9640 '
9650 '
9660 '
9670 '
9680 '
9690 '
9700 '
9710 '
9720 '
9730 '
9740 '
9750 '
9760 '
9770 '
9780 '
9790 '
9800 '
9810 '
9820 '
9830 '
9840 '
9850 '
9860 '
9870 '
9880 '
9890 '
9900 '
9910 '
9920 '
9930 '
9940 '
9950 '
9960 '
9970 '
9980 '
9990 '

```

```

5400 'SEND MACHINE NEXT MOVE FROM QUE
5410 '
5420 '
5430 IF JOBSRUNGZ(MACHINEZ)=1 THEN RETURN 'MACHINE BUSY
5440 IF MQPOINTERZ(MACHINEZ)=MQNEXTZ(MACHINEZ) THEN RETURN 'QUE IS EMPTY
5450 JOBZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),1))
5460 PRINT'SENDING MACHINE';MACHINEZ;' IT'S NEXT PROCESS'
5470 GOSUB 4870 'TURN ON PORT. IT IS NECESSARY TO TURN ON THE PORT OTHERWISE CA
N BE UNPREDICTABLE
5480 ON MACHINEZ GOSUB 5580,6260,6840,6660 'SEND DATA TO MACHINE
5490 RETURN
5500 '
5510 '1 - STACKER
5520 'PARAM1 - OPERATION #,PARAM2 - BIN #, PARAM3 - TRAY #
5530 '
5540 OPERTNZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3))
5550 BINZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),4))
5560 TRAYZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),5))
5570 IF OPERTNZ <> 1 GOTO 6020 ELSE IF BSTATUSZ(BINZ) <> 1 AND TSTATUSZ(TRAYZ) <
> 1 GOTO 6020 'PROCESS CAN BE EXECUTED, BIN OR TRAY NOT IN USE
5580 PRINT 'PROCESS FOR PART NUMBER ';PRNUMB$(JOBZ);' WAITING FOR A BIN OR TRAY
';PRINT' THAT IT USES TO BECOME FREE.'
5590 'CHECK QUES FOR RETURN BIN OPERATION
5600 STACKQUESZ=ABS(MQNEXTZ(MACHINEZ)-MQPOINTERZ(MACHINEZ))-1 'GET NUMBER OF JOB
S IN STACKER QUE (-1 BECAUSE MQNEXT POINTS TO NEXT QUE TO BE FILLED
5610 FOR PARAMZ=MQPOINTERZ(MACHINEZ)+1 TO STACKQUESZ+MQPOINTERZ(MACHINEZ)
5620 IF PARAMZ > MAXQUESZ THEN PARAMZ=1:WRAPPEDZ = 1
5630 IF DEBUG = 3 THEN PRINT'QUE==>';PARAMZ;' COMMAND==>';MACHINEQUE$(1,PARAMZ,3
)
5640 IF VAL(MACHINEQUE$(MACHINEZ,PARAMZ,3))=2 THEN RETQUEZ=PARAMZ:PARAMZ=STACKQU
ESZ+MQPOINTERZ(MACHINEZ):GOTO 5730 'CHECK QUE FOR A RETURN BIN COMMAND
5650 IF VAL(MACHINEQUE$(MACHINEZ,PARAMZ,3))=1 AND TSTATUSZ(VAL(MACHINEQUE$(MACHI
NEZ,PARAMZ,5)))=0 THEN RETQUEZ=PARAMZ:PARAMZ=STACKQUESZ+MQPOINTERZ(MACHINEZ):GOT
O 5730 'CHECK QUE FOR A EXECUTABLE RETRIEVE BIN COMMAND
5660 IF WRAPPEDZ=1 AND PARAMZ=MQNEXTZ(MACHINEZ) THEN PARAMZ=STACKQUESZ+MQPOINTER
Z(MACHINEZ)
5670 NEXT
5680 IF RETQUEZ=0 THEN PRINT'NO RETURN BIN OPERATIONS FOUND':RETURN
5690 'SHUFFLE QUES UP AND PUT RETURN BIN QUE IN CURRENT QUE POSITION
5700 'MOVE QUE WITH RETURN BIN COMMAND TO TEMPORARY STORAGE
5710 IF DEBUGZ=3 THEN PRINT'RETQUE==>';RETQUEZ
5720 FOR PARAMZ=1 TO MQPARAMSZ(MACHINEZ)+2
5730 TEMP$(PARAMZ)=MACHINEQUE$(MACHINEZ,RETQUEZ,PARAMZ)
5740 IF DEBUGZ=3 THEN PRINT'TEMP';PARAMZ;'==>';TEMP$(PARAMZ)
5750 NEXT
5760 QUEZ=RETQUEZ:RETQUEZ=0
5770 LQUEZ=QUEZ-1 'FIND NEXT LOWER QUE
5780 IF LQUEZ < 1 THEN LQUEZ=MAXQUESZ 'CHECK FOR WRAP AROUND
5790 IF DEBUGZ=3 THEN PRINT'MOVED QUE==>';LQUEZ
5800 FOR PARAMZ=1 TO MQPARAMSZ(MACHINEZ)+2
5810 MACHINEQUE$(MACHINEZ,QUEZ,PARAMZ)=MACHINEQUE$(MACHINEZ,LQUEZ,PARAMZ)
5820 IF DEBUGZ=3 THEN PRINT'PARAM==>';MACHINEQUE$(1,QUEZ,PARAMZ)
5830 NEXT
5840 IF DEBUGZ=3 THEN PRINT'MQPOINTERZ==>';MQPOINTERZ(1)
5850 IF LQUEZ=MQPOINTERZ(MACHINEZ) GOTO 5960
5860 IF QUEZ=1 THEN QUEZ=MAXQUESZ ELSE QUEZ=QUEZ-1
5870 LQUEZ=LQUEZ-1
5880 GOTO 5830 'MOVE NEXT QUE
5890 '
5890 'MOVE QUE WITH RETURN BIN COMMAND TO CURRENT QUE POSITION
5900 '
5910 '
5920 '
5930 '
5940 '
5950 '
5960 '
5970 '
5980 '
5990 '

```

11.0 11.0 0010 'EXECUTE MUX WITH RETURN BIN COMMAND'

```
6010 PRINT #MUXFZ,READY$; 'TELL APPLE READY
6020 FOR HOLDZ=1 TO 2000:NEXT:IF LOC(MUXFZ)=0 THEN GOTO 6030 'WAIT FOR BUFFER TO
  FILL UP WITH GARBAGE
6040 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ):X=INSTR(ANSWER$,'READY'):IF X=0 THEN PRIN
T'DIDN'T RECEIVE READY FROM STACKER':GOTO 6020
6050 PRINT #MUXFZ,OPERTNZ;CHR$(10);
6060 INPUT #MUXFZ,ANSWER$:PRINT'OPERATION SENT TO APPLE IS '#ANSWER$
6070 X=INSTR(ANSWER$,RIGHT$(STR$(OPERTNZ),1)):IF X = 0 THEN PRINT #MUXFZ,BAD$;;G
OTO 6050
6080 PRINT #MUXFZ,OK$;
6090 PRINT #MUXFZ,BINZ;CHR$(10);
6100 INPUT #MUXFZ,ANSWER$:PRINT'BIN # SENT TO APPLE IS '#ANSWER$
6110 X=INSTR(ANSWER$,RIGHT$(STR$(BINZ),1)):IF X = 0 THEN PRINT #MUXFZ,BAD$;;GOTO
  6090
6120 PRINT #MUXFZ,OK$;
6130 PRINT #MUXFZ,TRAYZ;CHR$(10);
6140 INPUT #MUXFZ,ANSWER$:PRINT'TRAY IS '#ANSWER$
6150 X=INSTR(ANSWER$,RIGHT$(STR$(TRAYZ),1)):IF X = 0 THEN PRINT #MUXFZ,BAD$;;GOT
  O 6140
6160 PRINT #MUXFZ,OK$;
6170 JOBSRUNZ(MACHINEZ)=1 'SETS MACHINE BUSY FLAG
6180 IF OPERTNZ=2 THEN BSTATUSZ(BINZ)=0:TSTATUSZ(TRAYZ)=0:JSTATUSZ(MACHINEZ,TRAY
  Z)=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),1)) 'FOR RETURN OPERATION RECOR
  D BIN, TRAY, AND JOB STATUS AS AVAILABLE
6190 IF OPERTNZ=1 THEN BSTATUSZ(BINZ)=1:TSTATUSZ(TRAYZ)=1:JSTATUSZ(MACHINEZ,TRAY
  Z)=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),1))
6200 MQPOINTERZ(MACHINEZ)=MQPOINTERZ(MACHINEZ) + 1
6210 IF MQPOINTERZ(MACHINEZ) > MAXQUEZ THEN MQPOINTERZ(MACHINEZ)=1
6220 PRINT #MUXFZ,CHR$(1) + '0'; 'IT IS NECESSARY TO DISCONNECT FROM THE MUX BEC
  AUSE IF IT DOESN'T SEND THE PORT NUMBER SOMETIMES IF IT IS ALREADY CONNECTED TO
  THE PORT WHEN THE APPLE SENDS A ATTENTION REQUEST SIGNAL (CTRL-A)
6230 RETURN
6240 '
6250 '2 - MINI ROBOT
6260 'PARAM1 - FILE NAME, PARAM2 AND 3 - MACHINES USED BY FILE
6270 '
6280 PARAMZ=4:MACHZ=1
6290 MACHZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),PARAMZ)):PARAMZ=PARAMZ
  + 1 '1ST MACHINE NEXT EXECUTION FILE IS DEPENDANT ON
6300 IF MACHZ=0 THEN GOTO 6360 'NOT DEPENDANT ON ANY OTHER MACHINES
6310 IF MACHZ=1 THEN TRAYZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),PARAMZ)
  ):IF JSTATUSZ(1,TRAYZ) <> JOBZ THEN PRINT'ROBOT WAITING FOR STACKER TRAY #';TRAY
  Z:RETURN:ELSE PARAMZ=PARAMZ + 1:GOTO 6340 'CHECK TO SEE IF STACKER IS READY
6320 IF MACHZ=99 THEN WAITMACHZ=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),PA
  RAMZ)):PARAMZ=PARAMZ+1:GOTO 6340
6330 IF JSTATUSZ(MACHZ,1) <> JOBZ THEN PRINT'ROBOT WAITING FOR MACHINE #';MACHZ:
  RETURN 'MACHINE NOT TO THIS JOB YET
6340 IF MACHZ=1 THEN MACHZ=2:GOTO 6290 'CHECK FOR TO SEE IF SECOND MACHINE DEPE
  DANT ON IS READY
6350 IF WAITMACHZ <> 0 THEN IF HELDZ(WAITMACHZ) <> JOBZ THEN PRINT'ROBOT WAITING
  FOR MACHINE'#WAITMACHZ:RETURN ELSE WAITMACHZ=0 'CHECK IF MACHINE ROBOT WAITING
  FOR IS HELD ON THIS JOB
6360 ' EXECUTE PROCESS
6370 PRINT #MUXFZ,'READY';CHR$(13);
6380 FOR HOLDZ=1 TO 4000:NEXT:IF LOC(MUXFZ)=0 GOTO 6380 'WAIT FOR APPLE TO SEND
  RESPONSE THROUGH MUX
6390 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ):X=INSTR(ANSWER$,'OK'):IF X=0 THEN PRINT'H
  AVING PROBLEMS SENDING TO APPLE - ROBOT':GOTO 6370
6400 PRINT'APPLE ACCEPTED FIRST READY'
6410 FOR HOLDZ=1 TO 1000:NEXT:PRINT #MUXFZ,'1';CHR$(13); 'WAIT FOR APPLE AND THE
  N CHOICE EXECUTE OPTION
6420 FOR HOLDZ=1 TO 2000:NEXT:IF LOC(MUXFZ)=0 THEN GOTO 6420 'WAIT FOR APPLE TO BE READY
6430 ' LEAVE APPLE FILE TO EXECUTE
```

```

1470 IF LOC(MUXFZ) <> 0 THEN PRINT "SENT SECOND READ:"
1480 IF LOC(MUXFZ) <> 0 THEN PRINT "IF LOC(MUXFZ)=0 GOTO 1490"
1490 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ)
1495 X=INSTR(ANSWER$,"OK");IF X=0 THEN PRINT "HAVING PROBLEMS RECEIVING FROM APPLE"
1500 - ROBOT":GOTO 1440
1510 FOR HOLDX=1 TO 1000:NEXT
1520 PRINT #MUXFZ,MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3);CHR$(13); 'SEND R
1530 PT FILE NAME TO EXECUTE
1540 PRINT "SENT FILE NAME"
1550 FOR HOLDX=1 TO 2000:NEXT:IF LOC(MUXFZ)=0 THEN PRINT "HAVING PROBLEMS RECEIVI
1560 NG FILE NAME BACK FROM APPLE - ROBOT":GOTO 1510
1570 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ):PRINT "APPLE SENT ==>";ANSWER$:X=INSTR(ANS
1580 WER$,MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3)):IF X=0 THEN PRINT "APPLE HAVIN
1590 G PROBLEMS RECEIVING ROBOT CONTROL FILE NAME":PRINT #MUXFZ,"BAD";CHR$(13);GOTO
1600 1490
1610 PRINT #MUXFZ,"OK";CHR$(13);
1620 FOR HOLDX=1 TO 5000:NEXT:IF LOC(MUXFZ)=0 GOTO 1630 'WAIT FOR APPLE TO ATTEM
1630 PT TO USE CONTROL FILE PASSED
1640 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ)
1650 X=INSTR(ANSWER$,"BAD"):PRINT "APPLE SENT ==>";ANSWER$:IF X<>0 THEN PRINT "ROB
1660 OT COULD NOT FIND ":PRINT "FILE ";MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3);"
1670 FOR PART NUMBER ";PRTNUM$(JOBZ):PRINT:INPUT "TYPE RETURN TO CONTINUE ";HOLD$:GOT
1680 O 1610
1690 PRINT #MUXFZ,CHR$(1);"0"; 'DISCONNECT MUX FROM ROBOT
1700 RSTATUSZ(MACHINEZ)=1 'SET ROBOT BUSY FLAG
1710 JOBSRUNGZ(MACHINEZ)=1 'SET ROBOT BUSY FLAG
1720 JSTATUSZ(MACHINEZ,1)=JOBZ 'JOB ROBOT CURRENTLY WORKING ON
1730 MQPOINTERZ(MACHINEZ)=MQPOINTERZ(MACHINEZ) + 1
1740 IF MQPOINTERZ(MACHINEZ) > MAXQUESZ THEN MQPOINTER(MACHINEZ)=1
1750 RETURN
1760 '
1770 '4 - TOYO LATHE
1780 'PARAM1 - FILE NAME,PARAM2 - FILE EXTENSION, PARAM3 - ROBOT IS DEPENDANT ON
1790 (MACHINE 2 OR 3)
1800 '
1810 IF JOBZ <> 0 THEN IF HOLDINGZ(MACHINEZ,JOBZ,HJOBZ) <> 0 AND MACHINEQUE$(MAC
1820 HINEZ,MQPOINTERZ(MACHINEZ),2) = "99" THEN HELDZ(MACHINEZ)=JOBZ:PRINT "LATHE IS BE
1830 ING HELD IDLE BY MACHINE ";MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3):RETURN
1840 IF MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),2)="99" THEN GOSUB 1820 'HOLDI
1850 NG FLAG QUE SKIPPED
1860 MACH1Z=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),4))
1870 IF MACH1Z <> 0 THEN IF JSTATUSZ(MACH1Z,1) <> JOBZ THEN RETURN 'NOT DEPENDAN
1880 T ON ROBOT OR ROBOT NOT REACHED THIS JOB YET
1890 PRINT #MUXFZ,READY$;
1900 FOR HOLDX=1 TO 2000:NEXT:IF LOC(MUXFZ)=0 THEN PRINT "HAVING PROBLEMS RECEIVI
1910 NG FROM APPLE":GOTO 1820 'WAIT FOR BUFFER TO FILL
1920 ANSWER$=INPUT$(LOC(MUXFZ),#MUXFZ):X=INSTR(ANSWER$,"READY"):IF X=0 THEN PRIN
1930 T "DIDN'T RECEIVE READY FROM LATHE":GOTO 1820
1940 PRINT #MUXFZ,"EXECUTE" + CHR$(10); 'SELECT EXECUTE LATHE FILE FILE OPTION
1950 INPUT #MUXFZ,ANSWER$:X=INSTR(ANSWER$,"OK"):IF X=0 THEN PRINT "LATHE DIDN'T T
1960 AKE EXECUTE":GOTO 1820
1970 LFILE$=MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),3) + "." + MACHINEQUE$(MAC
1980 HINEZ,MQPOINTERZ(MACHINEZ),4)
1990 PRINT #MUXFZ,LFILE$;CHR$(10); 'SEND LATHE FILE NAME TO EXECUTE
2000 PRINT #MUXFZ,CHR$(1);"0"; 'IT IS NECESSARY TO DISCONNECT FROM THE MUX OR EL
2010 SE MULTIFLEXOR DOESN'T NOTICE WHEN THE APPLE SENDS A ATTENTION REQUEST CHARACTER
2020 (CTRL-A)
2030 JOBSRUNGZ(MACHINEZ)=1 'SET MACHINE BUSY FLAG
2040 JSTATUSZ(MACHINEZ,1)=VAL(MACHINEQUE$(MACHINEZ,MQPOINTERZ(MACHINEZ),1)) 'REC
2050 ORDS WHICH JOB LATHE IS WORKING ON
2060 MQPOINTERZ(MACHINEZ)=MQPOINTERZ(MACHINEZ) + 1:IF MQPOINTERZ(MACHINEZ) > MAX
2070 QUESZ THEN MQPOINTERZ(MACHINEZ)=1 'INC MACHINE QUE POINTER
2080 RETURN
2090 '
2100 '

```

```

10 REM *****
20 REM * STORAGE AND RETRIEVAL SYSTEM CONTROLLER
30 REM * -BYU MINI-CAM LAB-
40 REM * ORIGINAL APPLESOFT VERSION BY DAVE JESPERSON
50 REM * MICROSOFT PROGRAM BY JUSTIN REDD
60 REM * OCTOBER, 1983
64 REM *
66 REM * MODIFIED IN JUNE OF 84 TO WORK UNDER CONTROL LQF GREG PETO'S PROGRAMS
70 REM *
80 REM * WHEN RUN, THIS PROGRAM WILL WAIT FOR 3
90 REM * PARAMETERS TO BE PASSED TO IT THROUGH
100 REM * THE RS232 LINK. THESE THREE PARAMETERS
110 REM * ARE DEFINED AS THE OPERATION#,BIN#,AND
120 REM * TRAY#. OPERATION# 0 = RETURN BIN
130 REM * OPERATION# 1 = RETRIEVE BIN
140 REM * OPERATION# 2 = QUIT PROGRAM
150 REM *
160 REM * TO RUN THIS PROGRAM CYDRIVER.HEX AND COMM.HEX
170 REM * MUST BE LOADED INTO MEMORY.
180 REM *
190 REM *****
200 REM
210 HOME:PRINT " APPLE STACKER PROGRAM":PRINT:PRINT"-----"
220 PRINT:PRINT
230 REM *****
240 REM * LOAD CYDRIVER AND COMM
250 REM *****
260 REM
270 CALLER$="STACKIBM.CON":CLINE=300:PRGM$="XXXX":SLINE=999:HFILE$="XXXX"
280 COMMON CALLER$,CLINE,PRGM$,SLINE,HFILE$
290 CHAIN "LOADASM.BAS"
300 REM
310 REM *****
320 REM * INITIALIZE
330 REM *****
334 READY$="READY" + CHR$(13)
335 BAD$="BAD" + CHR$(13)
336 OK$="OK" + CHR$(13)
338 DONE$=CHR$(1) 'MUX INTERRUPT REQUEST SIGNAL
340 REM
350 DIM BIN(63,2)
360 DIM TRAY(2,2)
370 BIN(11,1)=4085:BIN(11,2)=16375
380 BIN(12,1)=4138:BIN(12,2)=10750
390 BIN(13,1)=4138:BIN(13,2)=4925
400 BIN(21,1)=6140:BIN(21,2)=16375
410 BIN(22,1)=6140:BIN(22,2)=10675
420 BIN(23,1)=6212:BIN(23,2)=4775
430 BIN(31,1)=8200:BIN(31,2)=16375
440 BIN(32,1)=8269:BIN(32,2)=10775
450 BIN(33,1)=8262:BIN(33,2)=4875
460 BIN(41,1)=10250:BIN(41,2)=16375
470 BIN(42,1)=10300:BIN(42,2)=10750
480 BIN(43,1)=10312:BIN(43,2)=4925
490 BIN(51,1)=12287:BIN(51,2)=16375

```

```

500 BIN(52,1)=12312:BIN(52,2)=10800
510 BIN(53,1)=12344:BIN(53,2)=4850
520 BIN(61,1)=14320:BIN(61,2)=16375
530 BIN(62,1)=14337:BIN(62,2)=10775
540 BIN(63,1)=14350:BIN(63,2)=4875
550 TRAY(1,1)=0:TRAY(1,2)=16350
560 TRAY(2,1)=2025:TRAY(2,2)=16350
570 REM
580 CY$="I,R 253,S 1,F 1,A,P 1,P 0"
585 MOTOR1$="I,R 251,S 1,F 1,A,P 1,P 0"
586 MOTOR1$=MOTOR1$+" "
590 CY$=CY$+" "
600 CY1$="XXX"+" ":CY2$="XX"+" ":CY3$="XXX"+" "
610 FOR I = 1 TO 2
620 Q$=USR1(MOTOR1$)
630 Q$=USR2(CY$)
640 Q$=USR3(CY$)
650 NEXT I
660 REM
670 REM *****
680 REM * SET HOOK BACK 1 INCH
690 REM *****
700 REM
710 CY$="N 3000,-,G,A" 'MOVE HOOK OUT AN INCH AND A 1/8,SET AS AT HOME POSITION
715 MOTOR2$="N 16795,-,G,A"
716 Q$=USR2(MOTOR2$)
730 Q$=USR3(CY$)
741 GOSUB 2190 'SET HOOK BACK ONE INCH
742 'WAIT FOR READY SIGNAL FROM IBM
745 Q$=USR8(IBM$)
746 X=INSTR(IBM$,"READY"):IF X = 0 THEN Q$=USR7(BAD$):PRINT"RECEIVED ==>";IBM$:G
OTO 745
747 Q$=USR7(READY$):PRINT"RECEIVED READY"
750 REM
760 REM *****
770 REM * INPUT PARAMETERS FROM IBM
780 REM *****
790 REM
800 Q$="XXX" + " ":OPERATION$="XXX"+" ":XBIN$="XXX"+" ":XTRAY$="XXX"+" "
810 'OK$="OK"+CHR$(13):WHAT$="WHAT?"+CHR$(13):READY$="READY"+CHR$(13)
815 'Q$=USR7(READY$)
820 PRINT "OPERATION# = ";
840 Q$=USR8(OPERATION$):PRINT OPERATION$
842 ANSWER$=OPERATION$ + CHR$(13)
844 Q$=USR7(ANSWER$)
846 Q$=USR8(CHECK$):X=INSTR(CHECK$,"OK"):IF X=0 THEN GOTO 840 'APPLE DIDN'T RECE
IVE OPERATION CORRECTLY"
850 'IF OPERATION$ <> "0" AND OPERATION$ <> "1" AND OPERATION$ <> "2" THEN PRINT
"INVALID OPERATION#":Q$=USR7(WHAT$):PRINT:GOTO 820
860 OPERATION = VAL(OPERATION$)
870 IF OPERATION = 3 THEN GOTO 1470
880 'Q$=USR7(OK$)
890 PRINT "BIN# = ";
900 Q$=USR8(XBIN$):PRINT XBIN$
902 ANSWER$=XBIN$ + CHR$(13)
904 Q$=USR7(ANSWER$)

```

```

906 Q$=USR8(CHECK$):X=INSTR(CHECK$,"OK"):IF X=0 THEN GOTO 900
910 'DATA 11,12,13,21,22,23,31,32,33,41,42,43,51,52,53,61,62,63
920 'FOR I = 1 TO 18
930 'READ X
940 'IF VAL(XBIN$)=X THEN RESTORE: GOTO 1000
970 'NEXT I
980 'RESTORE:Q$=USR7(WHAT$):PRINT "BAD BIN#":PRINT:GOTO 890
1000 XBIN=VAL(XBIN$)
1005 'Q$=USR7(OK$)
1010 PRINT "TRAY# = ";
1020 Q$=USR8(XTRAY$):PRINT XTRAY$
1022 ANSWER$=XTRAY$ + CHR$(13)
1024 Q$=USR7(ANSWER$)
1025 'IF XTRAY$ <> "1" AND XTRAY$ <> "2" THEN PRINT "BAD TRAY#":PRINT:Q$=USR7(WH
AT$):GOTO 1010
1026 Q$=USR8(CHECK$):X=INSTR(CHECK$,"OK"):IF X=0 THEN GOTO 1020
1030 XTRAY=VAL(XTRAY$)
1040 PRINT:PRINT "-----":PRINT
1050 REM
1060 REM *****
1070 REM * GOTO OPERATION# ROUTINE
1080 REM *****
1090 REM
1100 'IF XTRAY < > 1 AND XTRAY < > 2 THEN PRINT "BAD TRAY# PASSED":GOTO 760
1110 'IF XBIN > 63 OR XBIN < 11 THEN PRINT "BAD BIN# PASSED":GOTO 760
1120 N$=RIGHT$(STR$(XBIN),1)
1130 'IF N$ <> "1" AND N$ <> "2" AND N$ <> "3" THEN PRINT "BAD BIN# PASSED":GOTO
760
1140 ON OPERATION GOTO 1310,1150 '1-RETRIEVE BIN,2-RETURN BIN
1150 REM *****
1160 REM * RETURN BIN
1170 REM *****
1180 REM
1190 GOSUB 1730
1200 GOSUB 1860
1210 FLAG$="TRAY"
1220 GOSUB 1960
1230 GOSUB 2270
1240 GOSUB 1620
1250 GOSUB 1860
1260 FLAG$="BIN"
1270 GOSUB 2070
1280 GOSUB 2170
1284 Q$=USR7(DONE$)
1286 PRINT"SENT FINISHED SIGNAL"
1290 GOTO 742 'WAIT FOR IBM READY SIGNAL
1300 REM
1310 REM *****
1320 REM * RETRIEVE BIN
1330 REM *****
1340 REM
1350 GOSUB 1620
1360 GOSUB 1860
1370 FLAG$="BIN"
1380 GOSUB 1960
1390 GOSUB 2270

```

```

1400 GOSUB 1730
1410 GOSUB 1860
1420 FLAG$="TRAY"
1430 GOSUB 2070
1440 GOSUB 2170
1444 Q$=USR7(DONE$)
1448 PRINT"SENT FINISHED SIGNAL"
1450 GOTO 742 'WAIT IBM READY SIGNAL
1460 REM
1470 REM *****
1480 REM * TERMINATE PROGRAM
1490 REM *****
1500 REM
1510 CY1$="P 0":CY2$="P 16795":CY3$="P 0"
1520 FOR I= 1 TO 2
1530 Q$=USR1(CY1$)
1540 Q$=USR2(CY2$)
1550 NEXT I
1560 FOR I = 1 TO 2
1570 REMQ$=USR3(CY3$)
1580 NEXT I
1590 HOME:PRINT "STACKER PROGRAM TERMINATED"
1600 PRINT:PRINT:END
1610 REM
1620 REM *****
1630 REM * GOTO BIN COODINATES
1640 REM *****
1650 REM
1660 CY1$="P" + STR$(BIN(XBIN,1))
1670 IF OPERATION = 1 THEN CY2$="P" + STR$( BIN(XBIN,2) + 385 )
1680 IF OPERATION = 2 THEN CY2$="P" + STR$( BIN(XBIN,2) )
1690 FOR I = 1 TO 2
1700 Q$=USR1(CY1$):Q$=USR2(CY2$)
1710 NEXT I
1720 RETURN
1730 REM
1740 REM *****
1750 REM * GOTO TRAY COORDINATES
1760 REM *****
1770 REM
1780 CY1$="P" + STR$(TRAY(XTRAY,1))
1790 IF OPERATION = 1 THEN CY2$="P" + STR$(TRAY(XTRAY,2))
1800 IF OPERATION = 2 THEN CY2$="P" + STR$( TRAY(XTRAY,2) + 385 )
1810 FOR I = 1 TO 2
1820 Q$=USR1(CY1$):Q$=USR2(CY2$)
1830 NEXT I
1840 RETURN
1850 REM
1860 REM *****
1870 REM * HOOK FORWARD
1880 REM *****
1890 REM
1900 CY3$="P 0"
1910 FOR I = 1 TO 2
1920 Q$=USR3(CY3$)
1930 NEXT I

```

```

1940 RETURN
1950 REM
1960 REM *****
1970 REM * ENGAGE HOOK
1980 REM *****
1990 REM
2000 IF FLAG$ = "BIN" THEN CY2$="P" + STR$( BIN(XBIN,2) )
2010 IF FLAG$ = "TRAY" THEN CY2$="P" + STR$( TRAY(XTRAY,2) )
2020 FOR I = 1 TO 2
2030 Q$=USR2(CY2$)
2040 NEXT I
2050 RETURN
2060 REM
2070 REM *****
2080 REM * DISENGAGE HOOK
2090 REM *****
2100 REM
2110 IF FLAG$="BIN" THEN CY2$="P" + STR$( BIN(XBIN,2) + 385 )
2120 IF FLAG$="TRAY" THEN CY2$="P" + STR$( TRAY(XTRAY,2) + 385 )
2130 FOR I = 1 TO 2
2140 Q$=USR2(CY2$)
2150 NEXT I
2160 RETURN
2170 REM
2180 REM *****
2190 REM * HOOK BACK ONE INCH
2200 REM *****
2210 REM
2220 CY3$="P 1600"
2230 FOR I = 1 TO 2
2240 Q$=USR3(CY3$)
2250 NEXT I
2260 RETURN
2270 REM
2280 REM *****
2290 REM * HOOK ALL THE WAY BACK
2300 REM *****
2310 REM
2320 CY3$="P 16000"
2330 FOR I = 1 TO 2
2340 Q$=USR3(CY3$)
2350 NEXT I
2360 RETURN
2370 REM

```

```

100 '*****
110 '* PART PROCESS TEXT FILE GENERATOR
120 '*      FOR THE LATHE
130 '* APPLESOFT PROGRAM BY PAUL SMITH
140 '*      REWRITTEN FOR MBASIC BY
150 '*      GREGORY J. PETO
160 '*      JUNE OF 1984
162 '*
164 '* TO CHAIN TO THIS PROGRAM FOR USE UNDER IBM CONTROL SET OPTION1 TO 2 WITH
A COMMON STATEMENT FIRST
165 '*
166 '* THIS PROGRAM GENERATES A STANDARD FORMAT CNC FILE THAT CAN THEN
167 '* BE USED WITH THE INTERPRETER TO DRIVE THE LATHE. THE DATA IT NEEDS
168 '* CAN BE DOWNLOADED FROM THE IBM MASTER CONTROLLER PROGRAM OR THE
169 '* PROGRAM WILL PROMPT FOR THE INFORMATION. IT ONLY SUPPORTS
170 '* PART FAMILIES A00 - A20
172 '*
174 '*****
180 '
182 '* INITIALIZE COMMUNICATION SUBROUTINES *
183 HOME
184 '
186 CALLER$="LTEXT.GEN":CLINE=300:PRGM$="XXXX":SLINE=999:HFILE$="XXXX" 'SET UP T
O RETURN TO THIS PROGRAM PARAMETERS
188 CHAIN "B:LOADASM",,ALL
300 '
310 '***** MAIN PROGRAM *****
320 '
330 GOSUB 480 'INITIALIZE ARRYS AND CONSTANTS
332 '
340 IF OPTION1 <> 3 THEN GOSUB 1350 'GET MODE OF OPERATION
342 '
350 GOSUB 1520 'GET DATA TO DESCRIBE PART
352 '
360 IF OPTION1 = MANUAL GOTO 420 'MANUAL PROGRAMING GOES DIRECTLY TO STORE DATA
362 '
370 GOSUB 3770 'DO INITIAL LATHE CALC'S
380 GOSUB 3910 'DO ROUGH FACE OF RIGHT END
390 GOSUB 4150 'DO ROUGH CUT
400 GOSUB 4630 'DO FINISH CUT
410 GOSUB 4930 'DO CUTOFF
420 GOSUB 5230 'STORE DATA
421 '
422 IF CALLER1$ <> "" THEN CHAIN CALLER1$
430 GOTO 340 'GET MODE AND START AGAIN
432 '
440 '*****
450 '
460 '* INITIALIZE ARRAYS AND CONSTANTS
470 '
480 HOME:PRINT"INITIALIZING VARIABLES"
490 ANSWER$ = ".51ANSWER = 0:PSTN = 1 'THIS PUTS OFTEN USED VARIABLES AT FRONT
OF LIST
500 OPTION BASE 1 'ARRAY PARAMTERS START AT 1 NOT 0
510 'ALL ARRAYS ARE INITIALIZED WITH 0 AUTOMATICALLY
520 DIM MOVE$(200) 'SETS MAX MOVES. EACH MEMBER OF THIS ARRAY WILL HOLD A LINE O

```

```

520 DIM MOVE$(200) 'SETS MAX MOVES. EACH MEMBER OF THIS ARRAY WILL HOLD A LINE O
' THE FINAL TEXT FILE CREATED
530 DIAMAX = 5 'MAX NUMBER OF DISTINCT DIAMETERS
540 DIM DMENSION(DIAMAX,2) 'FOR EACH DIAMETER THIS WILL HOLD 2 PARAMETERS- WIDTH
AND LENGTH
550 WDTH = 1 'USED WITH DMENSION ARRAY
560 LENGTH = 2 'USED WITH DMENSIONS ARRAY
570 DIM OFFSET(DIAMAX) 'WILL HOLD OFFSET FROM LEFT END FOR EACH DIAMETER
580 COUNT = 1 'USED TO SPECIFY THE CURRENT MOVE NUMBER
582 MAXCUT = .01 'MAXIMUM CUT DEPTH IN ONE PASS
590 MINFEED = 3 'MINIMUM FEEDRATE
592 MAXFEED = 10 'MAXIMUM FEEDRATE
593 MINLEN = .075 'MINIMUM LENGTH OF ANY SINGLE DIAMETER
594 MAXDIAM = .5 'MAX STOCK DIAMETER
595 MAXLEN = 4 'MAX STOCK LENGTH IN INCHES
596 MAXZ = 4 'MAA00EG16720RT1#0903809038G903809038020193T
598 MAXX = -2 'MAX MOVEMENT AWAY FROM CENTER
600 STP$ = "M02" 'COMMAND TO STOP LATHE
610 TOOL$ = "M06" 'COMMAND TO STOP FOR TOOL CHANGE
640 RAPID = 50 'RAPID TRANSVERSAL FEEDRATE
650 SHORT = 30 'SHORT MOVE FEEDRATE -USED ONLY WHEN NOT CUTTING
654 CUTOFFSPEED = 1.7 'CUT OFF FEEDRATE - SLOWEST FEEDRATE ALLOWABLE
660 CLRNC = .025 'CLEARANCE OF CUTTING TOOL FROM SURFACE FOR MOVES
670 FUDGE = .01 'FUDGE FACTOR USED IN CUTOFF
680 READY$="R" + CHR$(13)
682 EROR$="E" + CHR$(13)
700 '
710 '**** VERIFY VARIABLE CONSTANTS ****
720 '
730 CHANGE = 0
740 OPEN "I",#1,"LCONST"
750 INPUT #1,TOOL1CENT 'TOOL 1 OFFSET FROM CENTER LINE WHEN LATHE IS INITIALIZED
760 INPUT #1,TOOL1FACE 'TOOL 1 OFFSET FROM CHUCK FACE WHEN LATHE IS INITIALIZED
770 INPUT #1,TOOL2CENT
780 INPUT #1,TOOL2FACE
790 INPUT #1,CUTWIDTH 'WIDTH OF CUTOFF TOOL
800 INPUT #1,RS232 'SLOT NUMBER OF RS232 CARD
810 INPUT #1,DISKSLOT 'SLOT NUMBER OF DISK DRIVE OR HARD DISK
811 CLOSE
812 'DISPLAY CONSTANTS
813 IF OPTION1 = 3 THEN RETURN
820 WIDTH,24 :HOME:PRINT SPC(7);"CONSTANTS USED IN PROGRAM":NUMB = 1
830 PRINT"*****"
832 TAB1 = 5
840 PRINT NUMB;". OFFSET ROUGH CUT TOOL FROM":PRINT SPC(TAB1);"CENTER LINE = ";S
R$(TOOL1CENT);" INCHES.":NUMB = NUMB + 1
850 PRINT NUMB;". OFFSET ROUGH CUT TOOL FROM":PRINT SPC(TAB1);"CHUCK FACE = ";ST
$(TOOL1FACE);" INCHES":NUMB = NUMB + 1
860 PRINT NUMB;". OFFSET CUTOFF TOOL FROM":PRINT SPC(TAB1);"CENTER LINE = ";STR$
TOOL2CENT);" INCHES.":NUMB = NUMB + 1
870 PRINT NUMB;". OFFSET CUTOFF TOOL FROM":PRINT SPC(TAB1);"CHUCK FACE = ";STR$(
OOL2FACE);" INCHES":NUMB = NUMB + 1
880 PRINT NUMB;". WIDTH OF CUTOFF TOOL = ";STR$(CUTWIDTH);" INCHES":NUMB = NUMB
1
890 PRINT NUMB;". RS232 CARD IS IN SLOT #";STR$(RS232):NUMB = NUMB + 1
900 PRINT NUMB;". DISK DRIVE CARD IS IN SLOT #";STR$(DISKSLOT)

```

```

908 PSTN = 23 - 5 'POSTITION TO START INPUT LINE
910 VTAB(PSTN): PRINT "DO YOU WISH TO CHANGE ANY":PRINT "OF THESE (Y/N) ? ";:GET
  ANSWER$:PRINT
920 IF ANSWER$ = "N" AND CHANGE = 0 THEN RETURN 'NO CHANGES
930 IF ANSWER$ = "N" THEN HOME:PRINT"SAVING CHANGES":GOSUB 1230:RETURN 'SAVES PR
  IOR CHANGES AND THEN EXITS
940 IF ANSWER$ <> "Y" THEN :PRINT:PRINT"TYPE ONLY Y OR N":GOTO 910 'INPUT ERROR
1000 '
1010 '***** CHANGE CONSTANTS *****
1020 '
1028 VTAB(PSTN):FOR SPACE = 1 TO 4 :PRINT"
  NEXT 'CLEARS ANY PRIOR PROMPTS
1030 VTAB(PSTN):PRINT "ENTER NUMBER OF CONSTANT YOU WOULD":PRINT "LIKE TO CHANGE
  (1 -";NUMB;" ) OR C":PRINT"TO CANCEL ==>";
1040 GET ANSWER$:PRINT
1050 IF ANSWER$ = "C" THEN GOTO 820 'DISPLAY CONSTANTS AGAIN
1060 ANSWER = INT(VAL(ANSWER$))
1070 IF ANSWER < 1 OR ANSWER > NUMB THEN:PRINT: PRINT "TYPE ONLY 1 -";NUMB:GOTO
  1030 'INPUT ERROR
1078 VTAB(PSTN):FOR SPACE = 1 TO 5:PRINT"
  ":NEXT 'CLEARS ANY PRIOR PROMPT MESSAGES
1080 VTAB(PSTN):PRINT"ENTER NEW VALUE FOR CONSTANT ";:INPUT "==>",QUANTITY$:PRIN
  T
1110 QUANTITY = VAL(QUANTITY$)
1120 IF QUANTITY = 0 THEN PRINT:PRINT "ILLEGAL ENTRY":GOTO 1080
1122 CHANGE = 1
1130 IF ANSWER = 1 THEN TOOL1CENT = QUANTITY
1140 IF ANSWER = 2 THEN TOOL1FACE = QUANTITY
1150 IF ANSWER = 3 THEN TOOL2CENT = QUANTITY
1160 IF ANSWER = 4 THEN TOOL2FACE = QUANTITY
1170 IF ANSWER = 5 THEN CUTWIDTH = QUANTITY
1180 IF ANSWER = 6 THEN RS232 = QUANTITY
1190 IF ANSWER = 7 THEN DISKSLOT = QUANTITY
1200 GOTO 820 'REDISPLAY CONTANTS
1210 '
1220 '***** SAVE CONSTANT CHANGES *****
1230 '
1240 OPEN "O",#1,"LCONST"
1250 WRITE #1,TOOL1CENT,TOOL1FACE,TOOL2CENT,TOOL2FACE,CUTWIDTH,RS232,DISKSLOT
1270 CLOSE
1280 RETURN 'END OF CONSTANT INITIALIZATION
1290 '*****
1300 '
1310 '**** GET MODE OF OPERATION ****
1320 '
1350 HOME:PRINT SPC(7);"PROGRAM OPERATION CHOICES"
1352 PRINT SPC(15);"FOR LATHE"
1360 PRINT"*****"
1370 PRINT:PRINT:NUMB = 1
1372 TAB1 = 5
1380 PRINT NUMB;". MANUAL PART PROGRAMMING":PRINT:MANUAL = NUMB:NUMB = NUMB + 1
1382 PRINT NUMB;". AUTOMATED TEXT GENERATION WITH":PRINT SPC(TAB1);"CONTROL FROM
  APPLE":PRINT:NUMB = NUMB + 1
1390 PRINT NUMB;". AUTOMATED TEXT GENERATION WITH":PRINT SPC(TAB1);"CONTROL FROM
  IBM":PRINT:NUMB = NUMB + 1
1400 PRINT NUMB;". QUIT":PRINT

```

```

350 IF (XSPEED < 1.65 AND XSPEED > 0) OR (ZSPEED < 1.65 AND ZSPEED > 0) THEN GOS
3 2790:RETURN 'BELOW MIN SPEED ERROR
360 IF XSPEED > 112 OR ZSPEED > 112 THEN GOSUB 2890:RETURN 'EXCEED MAX SPEED ER
JR
370 DISTANCE= (-1) * X : SPEED=XSPEED
380 GOSUB 2300 'convert to cy512 commands
390 XCY$=CY$
400 DISTANCE=Z : SPEED=ZSPEED
410 GOSUB 2300
420 ZCY$=CY$
430 RETURN
440 '
450 '*** SAVE MOVE IN CY ARRAY ***
460 '
470 'IF G$="7" THEN ABX=0:ABZ=0:CYMOVE$(I%)="$G7":GOTO 2270
480 PRINT "LINE=";LINE%;" XCY$=";XCYS;" ZCY$=";ZCYS;" M=";M$;" T=";T
490 CYMOVE$(LINE%,1)=XCYS + " : " + ZCYS
500 IF T<>0 THEN M$=M$ + "T" + STR$(T) 'ADD IN TOOL NUMBER
510 CYMOVE$(LINE%,2)=M$
520 RETURN
530 '
540 '*** SAVE CY FILE TO DISK ***
550 '
560 FOR LINES% = 1 TO TOTLINES%
570 WRITE #2,CYMOVE$(LINES%,1),CYMOVE$(LINES%,2)
580 NEXT
590 RETURN
600 REM *****
610 REM * ROUTINE TO CONVERT DATA TO CY512 COMMANDS
620 REM *****
630 REM
640 IF DISTANCE = 0 THEN CY$="I,N 0" : RETURN
650 IF DISTANCE < 0 THEN DIRECTION$="-"
660 IF DISTANCE >= 0 THEN DIRECTION$="+"
670 STEPS = CINT(ABS(DISTANCE * 1600))
680 REM *** CALCULATE RATE AND FACTOR FOR CY512 ***
690 SS = SPEED * 26.667
700 IF SS < 43.5 THEN SS = 43.5
710 IF SS > 3000 THEN SS = 3000
720 RATE = CINT(257.938 - (12500! / SS))
730 IF RATE < 1 THEN RATE = 1
740 IF RATE > 253 THEN RATE = 253
750 FACTOR = CINT ((8 * RATE) + (100000! / SS) - 2055.5)
760 IF FACTOR < 1 THEN RATE = RATE + 1:GOTO 2450
770 REM *****
780 REM * CHECK TO SEE IF R, AND F ARE BEST
790 REM *****
800 FMIN = FACTOR - 1:F = FACTOR:FPLUS = FACTOR + 1
810 FEXP=FMIN:GOSUB 2600
820 ERRMIN = SSERR
830 FEXP=FPLUS:GOSUB 2600
840 ERRPLUS = SSERR
850 FEXP = F:GOSUB 2600
860 IF ERRMIN >= SSERR AND ERRPLUS >= SSERR THEN GOTO 2650
870 IF ERRMIN < SSERR THEN FACTOR = FACTOR - 1:GOTO 2490
880 IF ERRPLUS < SSERR THEN FACTOR = FACTOR + 1:GOTO 2490

```

```

570 IF XX%=1 THEN X=VAL(MID$(SLINE$,XP%+1,CHAR%-XP%-1)):XX%=0:GOTO 1650
580 IF ZZ%=1 THEN Z=VAL(MID$(SLINE$,ZP%+1,CHAR%-ZP%-1)):ZZ%=0:GOTO 1650
590 ' IF II%=1 THEN I=VAL(MID$(SLINE$,IP%+1,CHAR%-IP%-1)):II%=0:GOTO 1360
600 ' IF JJ%=1 THEN J=VAL(MID$(SLINE$,JP%+1,CHAR%-JP%-1)):JJ%=0:GOTO 1360
610 ' IF KK%=1 THEN K=VAL(MID$(SLINE$,KP%+1,CHAR%-KP%-1)):KK%=0:GOTO 1360
620 IF FF%=1 THEN F=VAL(MID$(SLINE$,FP%+1,CHAR%-FP%-1)):FF%=0:GOTO 1650
630 IF MM%=1 THEN M=MID$(SLINE$,MP%+1,CHAR%-MP%-1):MM%=0
640 IF TT%=1 THEN T=VAL(MID$(SLINE$,TP%+1,CHAR%-TP%-1)):TT%=0:IF T <= 0 OR T > M
XTTOOLS% THEN PRINT:PRINT "TOOL NUMBER EXCEEDS MAXIM
M OF";MAXTOOLS%:GOSUB 3190
650 IF C$="N" THEN NP%=CHAR%:NN%=1:GOTO 1770
660 IF C$="G" THEN GP%=CHAR%:GG%=1:GOTO 1770
670 IF C$="X" THEN XP%=CHAR%:XX%=1:GOTO 1770
680 IF C$="Z" THEN ZP%=CHAR%:ZZ%=1:GOTO 1770
690 ' IF C$="I" THEN IP%=CHAR%:II%=1:GOTO 1470
700 ' IF C$="J" THEN JP%=CHAR%:JJ%=1:GOTO 1470
710 ' IF C$="K" THEN KP%=CHAR%:KK%=1:GOTO 1470
720 IF C$="F" THEN FP%=CHAR%:FF%=1:GOTO 1770
730 IF C$="M" THEN MP%=CHAR%:MM%=1:GOTO 1770
740 IF C$="T" THEN TP%=CHAR%:TT%=1:GOTO 1770
750 IF C$="$" THEN CHAR%=LNGTH%:GOTO 1770
760 GOSUB 3060:CHAR%=LNGTH% 'ILLEGAL CHARACTER ERROR
770 RETURN
780
790 '*** INTERPOLATE MOVE ***
800
810 '***** TAKE CARE OF G CODES *****
820
830 IF G$="" GOTO 1940 'NO NEW G COMMANDS
840 IF G$="00" THEN PFLAG%=1:GOTO 1940 'SET FLAG FOR POINT TO POINT POSITIONING
850 IF G$="01" THEN PFLAG%=0:GOTO 1940 'SET FLAG FOR LINEAR INTERPOLATION
860 IF G$="05" THEN ABX=0 : ABZ=0:GOTO 1940 'MAKE PRESENT LOCATION NEW HOME POS
ITION
870 IF G$="92" THEN ABX=X:ABZ=Z:X=0:Z=0:GOTO 1940 'PRELOAD ABSOLUTE REGISTERS W
TH THE X AND Z IN PRESENT LINE OF TEXT
880 IF G$="90" THEN ABFLAG%=1:GOTO 1940 'SET FLAG FOR ABSOLUTE DIMENSION INPUT
890 IF G$="91" THEN ABFLAG%=0:GOTO 1940 'SET FLAG FOR INCREMENTAL INPUT
900 IF G$="02" THEN GOSUB 1840:RETURN 'DO ARC INTERPOLATION
910 PRINT:PRINT "G";G$;" IS NOT A LEGAL G FUNCTION CODE":GOSUB 3190 'INTERPRETA
ION ERROR
920
930 ' DO G FLAG COMMANDS
940
950 IF ABFLAG%=1 THEN X=X - ABX:Z=Z - ABZ 'IF IN ABSOLUTE DIMENSIONINPUT MODE C
NVERTS X AND Z TO INCREMENTAL DIMENSIONES
960 IF PFLAG%=1 THEN XSPEED=VRAPID:ZSPEED=VRAPID:XDIST=X:ZDIST=Z:GOTO 2070 'FOR
POINT TO POINT POSITIONING GO AS RAPID AS POSSIBLE
970
980 '*** FIND X AND Z DISTANCES
990
1000 ABX=ABX + X:ABZ=ABZ + Z 'UPDATE ABSOLUTE REGISTERS
1010 IF X=0 AND Z=0 THEN XCY$="R 250,S 1,F 1,N 0,+":ZCY$="R 250,S 1,F 1,N 0,+":R
TURN
1020 TLDIST=SQR(X*X+Z*Z)
1030 TIME=TLDIST/F
1040 XSPEED=ABS(X/TIME) : ZSPEED=ABS(Z/TIME)

```

```

1110 'GET MODE
1120 VTAB(5):PRINT"DO YOU WANT THE PART PROGRAM EXECUTED":PRINT"AFTER IT'S BEEN
INTERPRETED? (Y/N) ==>";GET ANSWER$
1130 IF ANSWER$="N" THEN MODE%=3:GOTO 1150
1140 IF ANSWER$<>"Y" GOTO 1120
1150 VTAB(5):HTAB(1):PRINT"
      ":PRINT"
              ":VTAB(5):RETURN

1160 '
1170 ' INPUT FILE SUBROUTINE
1180 '
1190 ON ERROR GOTO 3010
1200 IF OPENED1%=0 THEN OPEN"I",#1,RS274$:OPENED1%=1:ELSE GOTO 1250
1210 P=INSTR(RS274$,".")
1220 CY512%=MID$(RS274$,1,P-1) + ".CYC"
1230 IF OPENED2%=0 THEN OPEN"O",#2,CY512$:OPENED2%=1
1240 ON ERROR GOTO 0
1250 'LOAD A SET OF LINES FROM FILE
1260 TOTLINES%=MAXLINES%:EXMAXLINES%=1 'UNLESS DROPS OUT OF LOOP EARLY, EXCEEDS
MAX NUMBER OF LINES INTERPRETED FLAG IS SET
1270 FOR LINES% = 1 TO MAXLINES%
1280 IF EOF(1) THEN GOTO 1320
1290 INPUT #1,SLINE$(LINES%)
1300 IF SLINE$(LINES%) = "$" THEN TOTLINES% = LINES% - 1 :EXMAXLINES%=0:LINES%=M
AXLINES%
1310 NEXT
1320 RETURN
1330 REM
1340 REM ***** SEPARATE PARAMETERS AND CHECK FOR ERRORS *****
1350 REM
1360 'INITIALIZE ALPHA SEARCH VARIABLES
1370 X=0:Z=0:G$="":M$="":T=0
1380 LNTH%=LEN(SLINE%):IF LNTH% > MAXCHAR% THEN PRINT "LENGTH OF INPUT LINE EX
CEEDS MAXIMUM ALLOWABLE OF";MAXCHAR%:GOSUB 3190:RETU
RN 'INTERPRET ERROR
1390 FOR CHAR% = 1 TO LNTH%
1400 C%=MID$(SLINE%,CHAR%,1)
1410 GOOD%=0
1420 IF C% >= "A" AND C% <= "Z" THEN GOSUB 1540:GOTO 1480 'CHECK FOR ALPHA COMM
AND
1430 IF C%="$" THEN GOSUB 1540:CHAR%=LNTH%:GOTO 1480
1440 IF C%="+" OR C%="-" OR C%="." THEN GOOD%=1
1450 IF C%=" " THEN SLINE%=LEFT$(SLINE%,CHAR%-1) + MID$(SLINE%,CHAR%+1):GOTO 140
' DELETES SPACES AND STARTS AGAIN FOR THIS CHARACTE
{
1460 IF C%>= "0" AND C%<= "9" THEN GOOD%=1
1470 IF GOOD%=0 THEN GOSUB 3060:CHAR% = LNTH% 'ILLEGAL CHARACTER ERROR
1480 NEXT
1490 PRINT"SLINE=";SLINE$
1500 PRINT"N=";NLINE%; " G=";G$; " X=";X; " Z=";Z; " M=";M$; " T=";T
1510 RETURN
1520 REM
1530 REM ***** CHECK, SORT, AND STORE NEW PARAMETER *****
1540 REM
1550 IF NN%=1 THEN NLINE%=VAL(MID$(SLINE%,NP%+1,CHAR%-NP%-1)):NN%=0:GOTO 1650
1560 IF GG%=1 THEN G%=MID$(SLINE%,GP%-1,CHAR%-GP%-1):GG%=0:GOTO 1650

```

```

650 IF M$ >= " 0" AND M$ <= " 3" GOTO 700 'STOP PROGRAM EXECUTION COMMAND
670 IF M$=" 6" THEN GOSUB 3490 'CHANGE TOOL COMMAND
680 NEXT
682 '
690 IF EXMAXLINES%=1 THEN GOTO 580 'IF FILE EXCEEDED MAXLINES HANDLED IN ONE SE
T DO AGAIN
692 '
700 CLOSE:OPENED1%=0:OPENED2%=0
702 '
710 IF CALLER1$ <> "" THEN GOTO 740
720 GOSUB 3760 'DO PART AGAIN?
730 IF CSTOP=0 GOTO 540
732 '
740 'EXIT
742 '
760 DONE$=CHR$(1) 'MUX POLLING ATTENTION GETTER
780 IF CALLER1$ <> "" THEN Q$=USR7(DONE$):CHAIN CALLER1$
790 END
800 '
802 '*****
804 '
810 ' *** INITIALIZE VARIABLES AND COMMUNICATION SUBROUTINES
820 '
830 MAXLINES%=100
840 MAXCHAR%=80
850 BELL$=CHR$(7)
860 DIM SLINE$(MAXLINES%)
870 DIM CYMOVE$(MAXLINES%,2) 'CY512 COMMANDS,M COMMANDS
880 CY%=1:MC%=2 'POSITION IN ARRAY OF CY COMMANDS AND M COMMANDS
890 DIM CY$(2) 'HOLDS SEPARATED FOR X AND Z AXIS
900 XP%=1:ZP%=2 'POSITIONS OF X AND Z PARAMETERS
910 YCY$="R 250,S 1,F 1,N 0,+"
920 VRAPID=100 'VERY FASTEST FEEDRATE X OR Z AXIS CAN TRAVEL
930 MAXTOOLS%=9 'HIGHEST TOOL NUMBER ALLOWED
940 'COMMUNICATION SUBROUTINES
950 'CALLER$="LATHE.INT":CLINE=700:PRGM$="XXXX":SLINE=999:HFILE$="XXXX" 'SET PAR
AMETERS TO RETURN TO THIS PROGRAM
960 'CHAIN "LOADASM",,ALL 'LOADS ALL ASSEMBLY LANGUAGE SUBROUTINES
970 HOME:PRINT "LATHE CNC EXECUTOR - BYU M
INI LAB -""
980 PRINT:PRINT"-----
-----":PRINT:PRINT:PRINT
990 RETURN
1000 '
1010 ' GET NAME OF FILE TO INTERPRET
1020 '
1030 IF CALLER1$ <> "" THEN X=INSTR(RS274$,"."):CY512$=LEFT$(RS274$,X-1)+".CYC"
:RETURN 'PROGRAM BEING CHAINED TO
1040 VTAB(5):INPUT "PROCESS WHAT CNC FILE ";RS274$
1050 TEST$=RIGHT$ ( RS274$,4 )
1060 IF TEST$=".CYC" THEN MODE%=2:CY512$=RS274$:RETURN 'EXECUTE PREINTERPRETED F
ILE
1070 IF LEFT$( TEST$,1 ) = "." THEN GOTO 1090
1080 RS274$ = RS274$ + ".CNC"
1090 IF MID$(RS274$,2,1) = ":" THEN LLL = 14 ELSE LLL = 12
1100 IF LEN(RS274$)>LLL THEN PRINT:PRINT "FILENAME TOO LONG!":PRINT:GOTO 1040

```

```

250 REM *****
255 REM *      POKE BYTES S/R      *
260 REM *****
265 FOR I = 1 TO BYTES
270 LOCATION=ADDRESS + (I - 1)
275 CODE$="&H" + MID$(HLINE$, (8 + (I * 2)), 2)
280 CODE=VAL(CODE$)
285 POKE LOCATION, CODE
290 NEXT I
295 RETURN
300 DEF USR1=&HC409
310 DEF USR2=&HC416
320 DEF USR3=&HC423
324 '*****
330 '
340 ' *** MAIN BODY OF PROGRAM ***
350 '
354 '*****
360 'INTERPRET FILE
370 '
380 GOSUB 800 'INITIALIZE VARIABLES AND COMMUNICATION SUBROUTINES
384 '
390 GOSUB 1020 'GET FILE NAME
394 '
400 IF MODE% = 2 THEN LFILE% = 1:GOTO 540 'EXECUTE PREINTERPRETED FILE
402 '
410 GOSUB 1180 'INPUT A SET OF LINES
420 PRINT"INTERPRETING FILE";
422 '
430 FOR LNE% = 1 TO TOTLINES%
440     SLINE$=SLINE$(LNE%)
450     GOSUB 1350 'SEPARATE PARAMETERS
460     GOSUB 1790 'INTERPOLATE MOVE
470     GOSUB 2150 'SAVE MOVE TO CY ARRAY
480 PRINT". ";:NEXT 'INTERPRET NEXT LINE OF RS2374 FILE
484 '
490 GOSUB 2250 'SAVE CY ARRAY TO DISK
494 '
500 IF EXMAXLINES%=1 GOTO 410 'FILE EXCEEDED MAXIMUM LINES HANDLES IN ONE SET SO
    LOAD IN NEXT SET
510 WRITE #2,"$$$$"," ":CLOSE:OPENED1%=0:OPENED2%=0 'CLOSE ALL FILES, LAST ENTRY
    IN FILE IS EOF MARKER FOR CY512 FILE
520 IF INERROR%=1 THEN GOTO 740 'IF INTERPRETATION ERROR OCCURED THEN EXIT
530 '
540 '*** EXECUTE FILE ***
550 '
560 IF MODE%=3 THEN GOTO 740 'EXIT BECAUSE ISN'T TO BE EXECUTED
562 '
570 PRINT:PRINT"EXECUTING FILE";:STRTLNE%=1
580 IF LFILE%=1 THEN PRINT"LOADING IN FILE":GOSUB 3270 'FILE EXCEEDED MAXIMUM LI
    NES HANDLED IN ONE SET SO LOAD FILE FORM DISK
582 '
590 FOR LNE%=1 TO TOTLINES%
610     GOSUB 3370 'EXECUTE X,Z MOVE
630     IF CYMOVE$(LNE%,MC%) = "" THEN GOTO 680 ' NO M COMMAND
640     M$=STR$(VAL(CYMOVE$(LNE%,MC%))) 'FIND M COMMAND VALUE

```

```

10 REM *****
20 REM *
30 REM *      LATHE.INT      BY GREGORY J. PETO
40 REM *      - BYU CAM LAB -
50 REM * RS274 - CY512 TRANSLATOR  MAY, 1984
60 REM *
62 REM * ADAPTED FROM LINTERP.BAS AND CYEXEC.BAS BY JUSTIN D. REDD
64 REM *
70 REM * THIS PROGRAM READS RS274 STANDARD NUMERICAL CONTROL.
74 REM * THE RS274 FILE MUST HAVE THE EXTENSION .CNC.
80 REM * CODES FROM A TEXT FILE AND TRANSLATES THEM TO
90 REM * COMMANDS FOR THE CY512 STEPPER MOTER CONTROLLER
94 REM * AND WILL THEN EXECUTE THE FILE IF THAT OPTION IS CHOSEN.
100 REM* THE RESULTING CY512 COMMANDS ARE STORED IN A TEXT
110 REM* FILE OF THE SAME NAME WITH THE EXTENSION OF .CYC
114 REM* MIN AXIS SPEED IS 1.65 IN./MIN.
116 REM* MAX AXIS SPEED IS 112 IN./MIN.
118 REM* *** CHAINING TO THIS PROGRAM ***
120 REM* CALLER PROGRAM SHOULD PUT ITS OWN NAME IN CALLER1$ AND IT'S
122 REM* RETURN LINE NUMBER IN SLINE1.
124 REM* THE NAME OF THE FILE TO BE INTERPRETED SHOULD BE PUT IN RS274$
126 REM* IF EXECUTING PREINTERPRETED FILE THEN PUT IT'S NAME IN CY512$
128 REM* AND SET MODE% = 2.
130 REM*****
132 '
134 ' CHECK FOR CYDRIVER.HEX IN MEMORY
136 '
137 HOME
138 PRINT:INVERSE:PRINT "CHECKING FOR CYDRIVER.HEX";:NORMAL:PRINT
140 MESS$=""
142 FOR CHAR = 0 TO 6
144 LOCATION = &HC4F1 + CHAR
146 CHAR$=CHR$(PEEK(LOCATION))
148 MESS$=MESS$ + CHAR$:CHAR$= ""
150 NEXT CHAR
152 IF LEFT$(MESS$,5) <> "VALID" THEN GOTO 158
154 X$=MID$(MESS$,6,1)
156 IF X$="1" THEN GOTO 300
158 PRINT:INVERSE:PRINT "LOADING CYDRIVER.HEX AT C400H";:NORMAL:PRINT
160 HFILE$="CYDRIVER.HEX"
170 OPEN"I",#1,HFILE$
175 GOSUB 195
180 GOSUB 250
185 GOTO 175
190 REM
195 REM *****
200 REM *      READ LINE S/R      *
205 REM *****
210 REM
215 INPUT #1,HLINE$
220 HBYTES$="&H" + MID$(HLINE$,2,2)
225 HADDRESS$="&H" + MID$(HLINE$,4,4)
230 BYTES=VAL(HBYTES$):ADDRESS=VAL(HADDRESS$)
235 IF BYTES=0 AND ADDRESS=0 THEN HOME:CLOSE:PRINT "PROGRAM LOADED"
240 RETURN
245 REM

```

```

4990 ZNUL=0:Z=STOCKLEN - TOOL2FACE -(OFFSET(1) + CUTREF / 2 + FUDGE) + CUTWIDTH
5000 Z1=Z '1ST MOVE
5004 XNUL=1
5010 FEEDRATE=SHORT
5020 GOSUB 2820 'PUT IN MOVE ARRAY
5030 'MOVE TOWARD CENTER
5040 XNUL=0:X=STOCKDIAM / 2 + CLRNC
5050 ZNUL=1
5060 FEEDRATE=CUTOFFSPEED
5070 GOSUB 2820
5080 'MOVE AWAY FROM CENTER
5090 X= -X
5100 FEEDRATE=RAPID
5110 GOSUB 2820
5120 'RETURN TO START POSITION
5130 X= -X1
5132 ZNUL=1
5134 FEEDRATE=SHORT
5136 GOSUB 2820
5140 ZNUL=0:Z= -Z1
5144 XNUL=1
5146 FEEDRATE=RAPID
5150 GOSUB 2820
5160 COMMAND=1:X$=STP$
5170 GOSUB 2820
5180 RETURN
5190 '*****
5200 '
5210 '***** SAVE DATA TO DISK *****
5220 '
5230 HOME
5240 PRINT:PRINT"WRITING TO DISK";
5250 FILENAME$="L" + PRTNUMB$ + ".CNC"
5260 OPEN "O",#1,FILENAME$
5270 'WRITE #1,STOCKLEN,FEEDREF,CUTREF,STOCKDIAM 'FILE HEADER
5280 'THIS LOOP WRITES OUT MOVES
5290 FOR NUMB% = 1 TO COUNT-1 'COUNT HOLD VALUE OF NEXT MOVE
5300 PRINT #1,"N";NUMB%;MOVE$(NUMB%)
5302 PRINT". ";
5310 NEXT
5312 PRINT #1,"$"
5314 CLOSE
5320 COUNT = 1 'REINITIALIZE NUMBER OF MOVES
5330 RETURN
5340 '*****

```

```

4460 'MOVE TOOL AWAY FROM PART
4470 XNUL=0:X= -CLRNC
4480 Z=CLRNC
4490 FEEDRATE=SHORT
4500 GOSUB 2820
4510 'MOVE BACK TO RIGHT END
4520 XNUL=1
4530 Z=CUTL
4540 FEEDRATE=RAPID
4550 GOSUB 2820
4560 NEXT
4570 RETURN
4580 '*****
4600 '
4610 '***** DO FINISH CUT *****
4620 '
4630 'MOVE RIGHT TO POSITION TO DO FINISH FACE OF RIGHT END
4640 XNUL=1
4650 ZNUL=0:Z= -(CUTREF / 2 + CLRNC)
4660 FEEDRATE=SHORT
4670 GOSUB 2820 'PUT IN MOVE ARRAY
4680 'MOVE TOWARD CENTER
4690 XNUL=0:X=DMENSION(NUMBDIAM,WDTH) / 2 + CLRNC 'NUMBDIAM IS LAST DIAM - THE D
IAMETER OF THE FACE
4700 ZNUL=1
4710 FEEDRATE=FEEDREF
4720 GOSUB 2820
4730 'MOVE AWAY FROM CENTER
4740 X= -X + CLRNC 'MOVE OUT SAME DISTANCE MOVED IN AND THEN BACK IN A CLEARANCE
4750 GOSUB 2820
4760 'DO A FINISH PASS FOR EACH DIAMETER
4770 FOR DIAMNUMB = NUMBDIAM TO 1 STEP -1
4780 'MOVE LEFT LENGTH OF DIAMETER
4790 XNUL=1
4800 ZNUL=0:Z= -DMENSION(DIAMNUMB,LENGTH)
4810 GOSUB 2820
4820 'MOVE AWAY FROM CENTER TO MAKE NEXT LEFT MOVE
4830 IF DIAMNUMB = 1 THEN GOTO 4870 'ALREADY FINISHED
4840 XNUL=0:X= -((DMENSION(DIAMNUMB-1,WDTH) - DMENSION(DIAMNUMB,WDTH)) / 2)
4850 ZNUL=1
4860 GOSUB 2820
4870 NEXT
4880 'STOP FOR TOOL CHANGE. REINITIALIZES POSITION
4890 COMMAND=1:X$=TOOL$+"T2" 'COMMAND TO INSERT TOOL NUMBER 2 - THE CUTOFF TOOL
4900 GOSUB 2820
4910 RETURN
4920 '*****
4930 '
4940 '***** DO CUTOFF *****
4950 '
4960 'MOVE TO POSITION TO START CUTOFF
4970 XNUL=0:X= -(STOCKDIAM + CLRNC) + TOOL2CENT
4980 X1=X '1ST MOVE
4984 ZNUL=1
4985 FEEDRATE=SHORT
4986 GOSUB 2820 'PUT IN MOVE ARRAY

```

```

3930 ZNUL=0:Z=STOCKLEN - TOOL1FACE - CUTREF 'MOVES HORIZONTALLY. SETUP TO REFINI
SH FACE OF END
3940 FEEDINIT=1:FEEDRATE=RAPID
3950 GOSUB 2820 'PUT IN ARRAY
3960 'MOVW TOWARD CENTER TO TOUCHUP FACE OF END
3970 X=STOCKDIAM / 2 + CLRNC
3980 ZNUL=1
3990 FEEDRATE=FEEDREF
4000 GOSUB 2820
4010 'MOVE AWAY FROM FACE
4020 XNUL=1
4030 ZNUL=0:Z=CLRNC
4040 FEEDRATE=SHORT
4050 GOSUB 2820
4060 'MOVE AWWAY FROM CENTER
4070 XNUL=0:X= -X
4080 ZNUL=1
4090 GOSUB 2820
4100 RETURN
4110 '*****
4120 '
4130 '***** DO ROUGH CUT *****
4140 '
4150 FOR DIAMNUMB% = 1 TO NUMBDIAM
4160 'DOES A ROUGH CUTTING PROCEDURE FOR EACH DIAMETER
4170 GOSUB 4250 'FIND NUMBER OF PASSES AND DEPTH OF EACH
4180 GOSUB 4330 'DO ROUGH PASSES
4190 NEXT
4210 RETURN
4220 '*****
4230 '
4240 '* FIND ROUGH CUT DEPTH & NUMBER *
4250 '
4252 IF DIAMNUMB% = 1 THEN TOTCUT=(STOCKDIAM-DMENSION(1,WDTH))/2:GOTO 4270 'FIRS
T CUT IS FROM STOCK FACE DOWN TO DIAM 1
4260 TOTCUT=(DMENSION(DIAMNUMB% - 1,WDTH) - DMENSION(DIAMNUMB%,WDTH)) / 2 'FIND
TOTAL WIDTH TO BE CUT OUT
4270 NUMBCUTS%=CINT(TOTCUT / CUTREF) 'FINDS NUMBER OF PASSES REQUIRED
4280 CUTDEPTH=TOTCUT / NUMBCUTS% 'FINDS EXACT CUT DEPTH OF EACH PASS
4290 RETURN
4300 '*****
4310 '
4320 '***** DO ROUGH PASSES *****
4330 '
4340 CUTL=OFFSET(DIAMNUMB%) 'FINDS LENGTH OF EACH PASS
4350 FOR CUTN% = 1 TO NUMBCUTS%
4360 'MOVE TOWARD CENTER FOR PASS
4370 XNUL=0:X=CUTDEPTH + CLRNC
4380 ZNUL=1
4390 FEEDRATE=SHORT
4400 GOSUB 2820 'PUT IN MOVE ARRAY
4410 'DO PASS
4420 ZNUL=0:Z= -(CUTL + CLRNC)
4430 XNUL=1
4440 FEEDRATE=FEEDREF 'RATE CHOSEN BY PROGRAMMER
4450 GOSUB 2820

```

```

3331 READY$="PLOADED"+CHR$(13)
3332 Q$=USR7(READY$) 'SEND READY SIGNAL TO IBM
3334 Q$=USR8(IBM$):PRINT"SENT PLOADED. IBM SENT ==>";IBM$
3335 BAD$="BAD"+CHR$(13)
3336 X=0:X=INSTR(IBM$,"OK"):IF X=0 THEN PRINT:PRINT"ERROR RECIEVING FROM IBM":Q$
=USR7(BAD$):GOTO 3332
3338 READY$="GOOD"+CHR$(13):Q$=USR7(READY$)
3340 'Q$=USR8(PRTNAME$) 'PART NAME
3342 'PRINT"PART NAME RECEIVED ==>";PRTNAME$
3344 Q$=USR8(PRTNUMB$) 'PART NUMBER
3346 PRINT"PART NUMBER RECEIVED ==>";PRTNUMB$
3350 Q$=USR8(PRTFMLY$) 'PART FAMILY
3360 Q$=USR8(ANSWER$):NUMBDIAM = VAL(ANSWER$) 'NUMBER OF DIAMETERS
3364 PRINT"NUMBER OF DIAMETERS ==>";NUMBDIAM
3370 Q$=USR8(ANSWER$):FEEDREF = VAL(ANSWER$) 'FEEDRATE
3374 PRINT"FEED RATE ==>";FEEDREF
3380 Q$=USR8(ANSWER$):CUTREF = VAL(ANSWER$) 'CUT DEPTH
3384 PRINT"CUT DEPTH ==>";CUTREF
3390 Q$=USR8(ANSWER$):STOCKDIAM = VAL(ANSWER$) 'STOCK DIAMETER
3394 PRINT"STOCK DIAMETER ==>";STOCKDIAM
3400 Q$=USR8(ANSWER$):STOCKLEN = VAL(ANSWER$) 'STOCK LENGTH
3401 PRINT"STOCK LENGTH ==>";STOCKLEN
3402 FOR COUNTER%=1 TO NUMBDIAM
3404 Q$=USR8(ANSWER$):DMENSION(COUNTER%,LENGTH)=VAL(ANSWER$)
3405 PRINT"LENGTH DIAM.";COUNTER%;" ==>";DMENSION(COUNTER%,LENGTH)
3406 Q$=USR8(ANSWER$):DMENSION(COUNTER%,WDTH)=VAL(ANSWER$)
3407 PRINT"WIDTH DIAM.";COUNTER%;" ==>";DMENSION(COUNTER%,WDTH)
3408 NEXT
3409 INPUT "HOLDING",HOLD$
3410 RETURN
3720 RETURN
3740 '
3750 '**** INITIAL LATHE CALC'S
3760 '
3770 HOME:PRINT"CALCULATING LATHE MOVES";
3772 COMMAND=1:X$=TOOL$+"T1":GOSUB 2820 '1ST LINE IS COMMAND TO PUT IN TOOL #1
3780 'CALCULATE DISTANCE FROM LEFT REFERENCE FOR EACH DIAMETER
3790 FOR DIAMNUMB = 1 TO NUMBDIAM
3792 OFFSET(DIAMNUMB)=DMENSION(DIAMNUMB,WDTH)
3800 FOR OTHERDIAM = DIAMNUMB + 1 TO NUMBDIAM
3810 OFFSET(DIAMNUMB) = OFFSET(DIAMNUMB) +DMENSION(OTHERDIAM,LENGTH)
3820 NEXT:NEXT
3830 OFFSET(1) = OFFSET(1) + WIDTHCUT 'ADDS WIDTH OF CUT OFF TOOL TO 1ST DIAMETER
R
3840 'ALLOW FOR FINISH CUT
3850 FOR DIAMNUMB = 1 TO NUMBDIAM
3852 DMENSION(DIAMNUMB,WDTH) = DMENSION(DIAMNUMB,WDTH) + CUTREF/2 'ADDS WIDTH SO
THAT AFTER ROUGH CUT WILL BE SLIGHTLY OVER SIZED
3854 NEXT
3860 RETURN
3870 '*****
3880 '
3890 '**** DO FREE END FACE ****
3900 '
3910 'MOVE TO START POSITION
3920 XNUL=0:X=TOOL1CENT - STOCKDIAM / 2 - CLRNC 'MOVES TOOL BACK SO CAN MOVE

```

```

2860 OUTLINE$=""
2870 IF COMMAND=1 THEN OUTLINE$=X$ + "$":COMMAND=0:GOTO 2912 'STORE COMMAND IN A
RRAY
2880 IF XNUL = 0 THEN OUTLINE$="X" + STR$(X):TOTX = TOTX + X
2890 IF ZNUL = 0 THEN OUTLINE$=OUTLINE$ + "Z" + STR$(Z):TOTZ=TOTZ + Z
2900 IF FEEDINIT = 0 THEN INVERSE:PRINT"YOU'LL NEED TO GIVE FEEDRATE AN":PRINT"I
NITIAL VALUE OR MOVE CANNOT BE DONE.":NORMAL:PRINT:P
RINT"HIT ANY KEY TO CONTINUE";:GET HOLD$:RETURN
2910 OUTLINE$=OUTLINE$ + "F" + STR$(FEEDRATE) + "$"
2912 'STORE IT
2920 MOVE$(COUNT)=OUTLINE$
2930 COUNT=COUNT + 1
2940 PRINT". "; 'LETS OPERATOR KNOW IT IS RUNNING
2950 RETURN
2960 '*****
3000 '
3010 '* GET PART DESCRIPTION-APPLE CONTROL
3020 '
3030 GOSUB 1990 'GET GENERAL DESCRIPTION
3040 GOSUB 3090 'GET DIMENSIONS OF PART
3050 RETURN
3060 '*****
3070 '
3080 '* GET DIMENSIONS APPLE CONTROL *
3090 '
3100 TOTX=0:TOTZ=0
3120 FOR DIAMNUMB = 1 TO NUMBDIAM
3122 VTAB(PSTN):FOR SPACE = 1 TO 14 :PRINT CLR$:NEXT 'CLEAR SCREEN
3130 VTAB(PSTN):PRINT"ENTER WIDTH OF DIAMETER #";DIAMNUMB;
3140 INPUT" ==>",ANSWER$
3150 DMENSION(DIAMNUMB,WDTH)=VAL(ANSWER$)
3160 IF DMENSION(DIAMNUMB,WDTH) <= 0 OR DMENSION(DIAMNUMB,WDTH) > STOCKDIAM THEN
PRINT:PRINT"MAXIMUM TOTAL WIDTH IS";STOCKDIAM:GOTO
3130
3170 IF DIAMNUMB>1 THEN IF DMENSION(DIAMNUMB,WDTH)+CUTREF>=DMENSION(DIAMNUMB-1,W
DTH) THEN PRINT:PRINT"EACH SUCCESSIVE DIAMETER WIDTH
MUST BE":PRINT"LESS THAN THE LAST BY";CUTREF:PRINT:PRINT"DIAMETER #";DIAMNUMB-1;
" WIDTH IS";DMENSION(DIAMNUMB-1,WDTH):GOTO 3130
3180 TOTX = TOTX + DMENSION(DIAMNUMB,WDTH)
3188 TOTLEN=STOCKLEN - CLRNC - CUTWIDTH 'MAX TOTAL LENGTH IS STOCK LENGTH MINUS
A CLEARANCE AND CUTOFF TOOL WIDTH
3190 VTAB(PSTN):FOR SPACE = 1 TO 8:PRINT CLR$:NEXT
3200 VTAB(PSTN):PRINT"ENTER LENGTH OF DIAMETER #";DIAMNUMB;:INPUT" ==>",ANSWER$
3210 DMENSION(DIAMNUMB,LENGTH) = VAL(ANSWER$)
3220 IF DMENSION(DIAMNUMB,LENGTH) < MINLEN OR DMENSION(DIAMNUMB,LENGTH) + TOTZ >
TOTLEN THEN PRINT:PRINT"MAXIMUM TOTAL LENGTH IS";TO
TLEN;" AND ANY ":PRINT"ONE DIAMETER MUST BE GREATER THAN";MINLEN:PRINT"CURRENT T
OTAL LENGTH IS";TOTZ:GOTO 3200
3230 TOTZ=TOTZ + DMENSION(DIAMNUMB,LENGTH)
3240 NEXT 'LOOP BACK FOR NEXT DIAMETER
3250 RETURN
3260 '*****
3300 '
3310 '*** GET PART DESCRIPTION - IBM CONTROL ***
3320 '
3330 ' LOAD IN PART DESCRIPTION

```

```

2452 TOTX=TOOL1CENT:TOTZ=TOOL1FACE 'INITIALIZE ACCUMULATED X AND Z VALUES FOR CU
TTING TOOL
2460 HOME:PRINT"TYPE IN DESIRED NEW VALUE AND THEN":PRINT"TYPE -RETURN-." :PRINT
2470 PRINT"X MOVES CUTTER TOWARD CENTER IF POSITIVE"
2480 PRINT"Z MOVES CUTTER ALONG CENTER LINE TO":PRINT"RIGHT IF POSITIVE"
2490 PRINT"X AND Z ARE BOTH RELATIVE, FEEDRATE":PRINT"IS ABLSOLUTE."
2500 PRINT:PRINT"TYPE NOTHING BUT -RETURN- AND THAT":PRINT"PARAMETER STAYS THE S
AME"
2510 PRINT:PRINT"TYPE S FOR STOP OR T FOR TOOL CHANGE"
2520 PRINT"OR E FOR EXIT IN X VALUE."
2532 PRINT"*****"
2540 PSTN=17 'POSITION TO START INPUT LINE
2550 'GET NEXT POINT
2560 XNUL=0:ZNUL=0
2572 CLR$=""
2574 VTAB(PSTN):FOR SPACE=1 TO 4:PRINT CLR$:NEXT 'CLEAR SCREEN
2580 VTAB(PSTN):INPUT"ENTER NEXT X VALUE ==>",ANSWER$
2590 IF ANSWER$="" THEN XNUL=1:GOTO 2652 'JUMP TO Z
2600 IF ANSWER$="S" THEN COMMAND=1:X$=STP$:GOSUB 2830:TOTX=TOOL1CENT:TOTZ=TOOL1F
ACE:GOTO 2550
2610 IF ANSWER$="T" THEN COMMAND=1:X$=TOOL$:GOSUB 2830:TOTX=TOOL1CENT:TOTZ=TOOL1
FACE:GOTO 2550
2620 IF ANSWER$="E" THEN RETURN 'EXIT. PROGRAMMING FINISHED
2630 X=VAL(ANSWER$)
2640 IF X+TOTX < MAXX THEN PRINT:PRINT"TOTAL X CANNOT BE LESS THEN";MAXX;"
":PRINT CLR$:GOTO 2580
2641 IF X+TOTX > -MAXX THEN PRINT:PRINT"TOTAL X CANNOT BE GREATER THAN";-MAXX"
":PRINT CLR$:GOTO 2580
2642 IF X = 0 THEN PRINT:PRINT"TYPE NOTHING BUT -ENTER- IF X":PRINT"DOESN'T CHAN
GE
":GOTO 2580
2650 VTAB(PSTN):FOR SPACE=1 TO 4:PRINT CLR$:NEXT
2652 'GET Z VALUE
2670 VTAB(PSTN):INPUT"ENTER NEXT Z VALUE ==>",ANSWER$
2680 IF ANSWER$="" THEN ZNUL=1:GOTO 2702 'JUMP TO FEEDRATE
2690 Z=VAL(ANSWER$)
2698 IF Z = 0 THEN PRINT:PRINT"TYPE NOTHING BUT -RETURN- IF Z
":PRINT"DO
ESN'T CHANGE":GOTO 2670
2700 IF Z+TOTZ < 0 OR Z+TOTZ > MAXZ THEN PRINT:PRINT"TOTAL Z CANNOT BE LESS THAN
0 OR":PRINT"GREATER THAN";MAXZ:GOTO 2670
2702 VTAB(PSTN):FOR SPACE=1 TO 4:PRINT CLR$:NEXT
2710 'GET FEEDRATE
2720 VTAB(PSTN):INPUT"ENTER FEEDRATE ==>",ANSWER$
2730 IF ANSWER$="" THEN :GOTO 2770 'JUMP TO PUT IN ARRAY
2740 FEEDRATE=VAL(ANSWER$)
2750 IF FEEDRATE < MINFEED THEN PRINT:PRINT"FEEDRATE CANNOT BE LESS THAN";MINFEE
D:GOTO 2720
2760 FEEDINIT = 1 'FEEDRATE HAS BEEN INITIALIZED
2770 GOSUB 2830 'PUT MOVE IN ARRAY
2780 GOTO 2550 'GET NEXT MOVE
2790 '*****
2800 '
2810 '***** PUT IN ARRAY *****
2820 '
2830 'X AND Z ARE PASSED AS NUMBERS XNULL AND ZNULL ARE USED TO INDICATE THERE A
RE NO CHANGES IN THOSE PARAMETERS
2850 IF XNUL=1 AND YNUL=1 THEN RETURN 'NO ENTRIES MADE

```

```

2100 IF ANSWER$ <> "N" THEN PRINT:PRINT"TYPE ONLY Y OR N":GOTO 2080 'INPUT ERROR
2110 VTAB(PSTN):FOR SPACE = 1 TO 10:PRINT CLR$:NEXT 'CLEARS ALL BUT TOP OF SCREE
N
2120 GOOD = 0
2122 VTAB(PSTN):INPUT"ENTER PART FAMILY ==>",PRTFMLY$
2130 IF PRTFMLY$ = "A00" THEN NUMBDIAM = 1:GOOD=1
2140 IF PRTFMLY$ = "A10" THEN NUMBDIAM = 2:GOOD=1
2150 IF PRTFMLY$ = "A20" THEN PRINT:PRINT"MAXIMUM NUMBER OF DIAMETERS IS ";DIAMA
X:PRINT:INPUT"ENTER NUMBER OF DIAMETERS ==>",ANSWER$
:NUMBDIAM = INT(VAL(ANSWER$)):IF NUMBDIAM >= 2 AND NUMBDIAM <= DIAMAX THEN GOOD
= 1
2170 IF GOOD = 0 THEN PRINT:PRINT"FAMILY NOT RECOGNIZED":GOTO 2120 'INPUT ERROR
2180 VTAB(PSTN):FOR SPACE = 1 TO 8:PRINT CLR$:NEXT
2190 VTAB(PSTN):PRINT"TYPICAL FEEDRATES FOR MINI-LATHE ARE":PRINT"3 TO 10 IN./MI
N. 4-5 IS MAX FOR ALUMINUM."
2200 VTAB(PSTN + 5):INPUT"ENTER FEEDRATE ==>",ANSWER$
2210 FEEDREF=INT(VAL(ANSWER$))
2220 IF FEEDREF < MINFEED OR FEEDREF > MAXFEED THEN PRINT:PRINT"FEEDRATE CANNOT
BE LESS THAN";MINFEED;"OR":PRINT"GREATER THAN";MAXFE
ED:GOTO 2200 'INPUT ERROR
2230 VTAB(PSTN):FOR SPACE = 1 TO 9:PRINT CLR$:NEXT
2240 VTAB(PSTN):PRINT"TYPICAL CUT DEPTH PER PASS IS .01 IN.":PRINT"FOR MINI-LATH
E."
2250 VTAB(PSTN + 5):INPUT"ENTER DEPTH OF CUT IN INCHES ==>",ANSWER$
2260 CUTREF=VAL(ANSWER$)
2270 IF CUTREF <= 0 OR CUTREF > MAXCUT THEN PRINT:PRINT"CUT DEPTH CANNOT BE GREA
TER THAN";MAXCUT:GOTO 2250
2280 VTAB(PSTN):FOR SPACE = 1 TO 8:PRINT CLR$:NEXT
2290 VTAB(PSTN):PRINT"DEMO STOCK DIAMETER IS .5 INCHES"
2300 VTAB(PSTN + 5):INPUT"ENTER STOCK DIAMETER ==>",ANSWER$
2310 STOCKDIAM = VAL(ANSWER$)
2320 IF STOCKDIAM <= 0 OR STOCKDIAM > MAXDIAM THEN PRINT:PRINT"DIAMETER CANNOT B
E GREATER THAN";MAXDIAM:GOTO 2300
2322 VTAB(PSTN):FOR SPACE = 1 TO 8:PRINT CLR$:NEXT
2324 VTAB(PSTN):PRINT"TYPICAL STOCK LENGTH IS 3 INCHES":PRINT"IT MUST BE LONGER
THAN PART":PRINT"PLUS TOOLWIDTHS"
2326 VTAB(PSTN + 5):INPUT"ENTER STOCK LENGTH ==>",ANSWER$
2327 STOCKLEN = VAL(ANSWER$)
2328 IF STOCKLEN <= 0 OR STOCKLEN > MAXLEN THEN PRINT:PRINT"CANNOT BE LONGER THE
N";MAXLEN:GOTO 2326
2330 VTAB(PSTN):FOR SPACE = 1 TO 8:PRINT CLR$:NEXT
2340 VTAB(PSTN):PRINT"PART FAMILY IS ";PRTFMLY$;" (WITH";NUMBDIAM;"DIAMETERS)"
2350 PRINT:PRINT"REFERENCE FEERATE IS";FEEDREF
2360 PRINT:PRINT"REFERENCE CUTDEPTH IS";CUTREF
2370 PRINT:PRINT"STOCK DIAMETER IS";STOCKDIAM
2380 PRINT:PRINT"STOCK LENGTH IS";STOCKLEN
2390 VTAB(PSTN + 11):PRINT"DO YOU WANT TO CHANGE ANY OF":PRINT"ABOVE (Y/N) ";:GE
T ANSWER$:PRINT
2400 IF ANSWER$ = "Y" THEN VTAB(PSTN):FOR SPACE = 1 TO 15:PRINT CLR$:NEXT:GOTO 2
120 'GET THEM AGAIN
2410 IF ANSWER$ <> "N" THEN PRINT:PRINT"TYPE ONLY Y OR N":GOTO 2390
2412 RETURN
2420 '*****
2430 '
2440 '* MANUAL POINT-TO-POINT PROGRAMMING *
2450 '

```

```

1408 PSTN = 23 - 4 'POSTION TO START INPUT LINE
1410 VTAB(PSTN):PRINT"ENTER CHOICE (1 -";NUMB;"") ==>";:GET ANSWER$
1430 OPTION1 = INT(VAL(ANSWER$))
1440 IF OPTION1 = 0 OR OPTION1 > NUMB THEN PRINT:PRINT"TYPE ONLY 1 -";NUMB:GOTO
1410 'INPUT ERROR
1450 IF OPTION1 = NUMB THEN END 'ENDS PROGRAM
1460 'OPTION1 NOW HOLDS CORRECT MODE CHOICE
1470 RETURN
1480 '*****
1490 '
1500 '**** GET DATA TO DESCRIBE PART ****
1510 '
1520 ON OPTION1 GOSUB 1560,3020,3320 'GET DATA MANUALLY, FROM APPLE, FROM IBM
1530 RETURN
1532 '*****
1540 '
1550 '**** MANUAL TEXT GENERATION ****
1560 '
1570 GOSUB 1640 'GET CHOICE OF OPTION TO CREATE A NEW FILE OR MODIFY OLD
1590 ON OPTION2 GOSUB 1920,1600 'MODIFY OR CREATE
1600 RETURN
1610 '
1620 '**** CHOICE CREATE OR MODIFY ****
1630 '
1640 HOME
1650 PRINT SPC(3);"OPTIONS FOR MANUAL TEXT GENERATION"
1660 PRINT"*****":PRINT
1670 PRINT"1. CREATE NEW FILE":PRINT
1680 PRINT"2. MODIFY AN OLD FILE":PRINT
1700 VTAB(PSTN):PRINT"ENTER CHOICE (1-2) ==>";:GET ANSWER$
1710 OPTION2 = INT(VAL(ANSWER$))
1720 IF OPTION2 < 1 OR OPTION2 > 2 THEN PRINT:PRINT"ILLEGAL ENTRY":GOTO 1700
1730 RETURN
1740 '*****
1900 '
1910 '***** CREATE MANUALLY *****
1920 '
1930 GOSUB 2000 'GETS GENERAL PART DESCRIPTION
1940 GOSUB 2450 'DO POINT TO POINT PROGRAMMING
1950 RETURN
1960 '*****
1970 '
1980 '** GET GENERAL PART DESCRIPTION **
1990 '
2000 HOME
2010 PRINT SPC(7);"GENERAL PART DESCRIPTION"
2020 PRINT"*****"
2030 PSTN = 4 'POSTION TO START INPUT LINE
2032 CLR$=""
2040 VTAB(PSTN):INPUT "ENTER PART NAME ==>",PRTNAME$
2050 VTAB(PSTN):PRINT CLR$:VTAB(PSTN):INPUT "ENTER PART NUMBER ==>",PRTNUMB$
2060 VTAB(PSTN):PRINT CLR$:VTAB(PSTN):PRINT"PART NAME IS ";PRTNAME$:PRINT
2070 PRINT"PART NUMBER IS ";PRTNUMB$
2080 VTAB(PSTN + 5):PRINT"DO YOU WISH TO CHANGE ANY":PRINT"OF ABOVE (Y/N) ? ";:G
ET ANSWER$:PRINT
2090 IF ANSWER$ = "Y" THEN HOME:GOTO 2010 ' GET AGAIN

```

```

2590 PRINT:PRINT "ERROR IN R AND F CHECK ROUTINE":END
2600 REM ***** CALCULATE SS FROM R AND F VALUES *****
2610 DENOM= (((256 - RATE) * 80) + (FEXP * 10) + 75) * .000001
2620 SSEXP= 1/DENOM
2630 SSERR=ABS(SS-SSEXP)
2640 RETURN
2650 REM ** CALCULATE SLOPE **
2660 IF SPEED < 12 THEN SLOPE = 255 : GOTO 2710
2670 SLOPE = 1
2680 YY = 2 * ( INT(( 255 - FACTOR ) / SLOPE ) )
2690 IF SLOPE > 254 THEN GOTO 2710
2700 IF YY > STEPS THEN SLOPE = SLOPE + 1 : GOTO 2680
2710 RATE$=STR$(RATE):SLOPE$=STR$(SLOPE):FACTOR$=STR$(FACTOR):STEPS$=STR$(STEPS)
2720 CY$="R"+RATE$+", "S"+SLOPE$+", "F"+FACTOR$+", "N"+STEPS$+", "+DIRECTION$
2730 RETURN
2740 REM *****
2750 REM * SUBROUTINE TO DO ARC INTERPOLATION
2760 REM *****
2770 PRINT"ARC INTERPOLATION NOT AVAILABLE YET":GOSUB 3190:RETURN 'INTERPOLATION
  ERROR
2780 '
2790 REM *****
2800 REM * AXIS SPEED < 1.65 ERROR
2810 REM *****
2820 REM
2830 AXIS$=" "
2840 IF XSPEED < 1.65 AND XSPEED > 0 THEN AXIS$="X-AXIS"
2850 IF ZSPEED < 1.65 AND ZSPEED > 0 THEN AXIS$=AXIS$+", Z-AXIS"
2860 PRINT "INTERPOLATION CAUSES ";AXIS$;" TO"
2870 PRINT "TO BE LESS THEN MINIMUM OF 1.65 IN./MIN."
2880 GOTO 3190 'INTERPRETATION ERROR
2890 REM *****
2900 REM * AXIS SPEED > 112 ERROR
2910 REM *****
2920 REM
2930 IF XSPEED > 112 AND ZSPEED > 112 THEN AXIS$="X AND Z AXIS":GOTO 2960
2940 IF XSPEED > 112 THEN AXIS$="X AXIS"
2950 IF ZSPEED > 112 THEN AXIS$="Z AXIS"
2960 PRINT"FEED RATE WITH INTERPOLATION CAUSES ";AXIS$;
2970 PRINT"TO EXCEED MAX. OF 112 IN./MIN."
2980 GOTO 3190 'INTERPRETATION ERROR
2990 '
3000 ' ROUTINE TO HANDLE FILE NAME ERRORS
3010 '
3020 IF ERR=53 OR ERR = 64 THEN PRINT:PRINT"FILE NOT FOUND OR BAD FILE NAME":GOT
  O 3050
3030 IF ERR = 62 THEN PRINT:PRINT"CYC FILE MISSING '$$$$' EOF MARKER":GOTO 3050
3040 PRINT:PRINT"ERROR LOADING FILE OTHER THAN BAD NAME"
3050 PRINT:PRINT"TYPE ANY KEY TO CONTINUE":GET HOLD$:GOTO 740 'EXITGOTO 502 'EXI
  T PROGRAM
3060 '
3070 ' ILLEGAL CHARACTER ERROR
3080 '
3090 PRINT
3100 IF C$="Y" THEN PRINT"LATHE ONLY USES X AND Z AXIS":GOTO 3190
3110 IF C$>" " THEN PRINT"ILLEGAL CHARACTER ";C$;"":GOTO 3190

```

```

3120 PRINT"ILLEGAL CHARACTER ASCII CODE";ASC(C$):GOTO 3190
3130 '
3140 'CONTAINS TWO OF SAME PARAMETER
3150 '
3160 PRINT"CONTAINS 2 OF SAME PARAMTER ";C$;" IN SAME LINE.":GOTO 3190
3170 '
3180 ' INTERPRETING ERROR FINAL ROUTINE
3190 '
3200 PRINT"ERROR IS IN LINE #";NLINE%
3210 PRINT SLINE$
3220 PRINT"THE";LNE%;" RD CONSECUTIVE LINE FROM THE BEGINNING"
3230 PRINT"TYPE C TO CONTINUE INTERPRETATION, ANY OTHER KEY EXITS";
3240 GET ANSWER$:IF ANSWER$ = "C" THEN INERROR%=1:RETURN 'SET INTERPRET ERROR FL
AG AND GO BACK TO INTERPRET NEXT LINE
3250 '
3260 ' READ IN INTERPRETED FILE
3270 '
3280 PRINT"LOAD FILE STARTED"
3290 ON ERROR GOTO 3010
3300 IF OPENED1%=0 THEN OPEN "I",#1,CY512$:OPENED1%=1
3310 TOTLINES%=MAXLINES%:EXMAXLINES%=1
3320 FOR LINES%=1 TO MAXLINES%
3330 INPUT #1,CYMOVE$(LINES%,1),CYMOVE$(LINES%,2)
3340 IF CYMOVE$(LINES%,1) = "$$$$" THEN TOTLINES%=LINES%-1:EXMAXLINES%=0:LINES%=
MAXLINES%
3350 NEXT:ON ERROR GOTO 0:RETURN
3360 '
3370 ' SEPARATE X,Y AND Z AXIS
3380 '
3390 COLON1=INSTR(CYMOVE$(LNE%,CY%),":")
3400 XP$=LEFT$(CYMOVE$(LNE%,CY%),COLON1-2)
3410 ZP$=MID$(CYMOVE$(LNE%,CY%),COLON1+2)
3420 '
3430 ' EXECUTE MOVE
3440 '
3450 Q$=USR1(XP$)
3460 Q$=USR3(ZP$)
3470 Q$=USR1("G"):Q$=USR3("G") 'SENDS GO TO BLUE BOX
3480 RETURN
3490 '
3500 ' TOOL CHANGE SUBROUTINE
3510 '
3520 'REINITIALIZE LATHE
3530 '
3540 PRINT:PRINT"REINITIALIZING LATHE"
3550 STAT1=&HEOF0:STAT2=&HEOF2
3560 CY1$="I,R 250,S 1,F 1,N 10000,+,G":CY2$="I,R 220,S 1,F 1,N 20000,-,G"
3570 Q$=USR1(CY1$)
3580 X$=RIGHT$((HEX$(PEEK(STAT1))),1)
3590 IF ASC(X$) > 55 THEN POKE STAT1,&H44:POKE STAT1,&H40:GOTO 3610
3600 GOTO 3580
3610 Q$=USR3(CY2$)
3620 Z$=RIGHT$((HEX$(PEEK(STAT2))),1)
3630 IF ASC(Z$) > 55 THEN POKE STAT2,&H44:POKE STAT2,&H40:GOTO 3650
3640 GOTO 3620
3650 CY1$="R 100,S 1,F 1,N 15,+,G":CY2$="R 100,S 1,F 1,N 15,-,G"

```

```
3660 Q$=USR1(CY1$):Q$=USR3(CY2$)
3670 ABX=0:ABZ=0 'RESET ABSOLUTE REGISTERS
3680 '
3690 ' DISPLAY TOOL NUMBER
3700 '
3710 T$=MID$(CYMOVE$(LNE%,MC%),INSTR(CYMOVE$(LNE%,MC%),"T")+1)
3720 PRINT:PRINT "LOAD TOOL #";T$
3730 PRINT:PRINT "TYPE R WHEN READY TO CONTINUE."
3740 GET ANSWER$:IF ANSWER$ <> "R" GOTO 3730
3750 RETURN
3760 '
3770 ' DO AGAIN ?
3780 '
3790 PRINT:PRINT:PRINT"DO YOU WISH TO MAKE SAME PART AGAIN ? (Y FOR YES)";:GET A
NSWER$
3800 IF ANSWER$ <> "Y" THEN CSTOP=1 ELSE CSTOP=0
3810 RETURN
```

Students who have been engaged with the development of the Manufacturing Information System at Brigham Young University.

<u>Year</u>	<u>Name of Student</u>	<u>Major</u>	<u>Description of Work</u>
<u>Prior to Grant</u>			
1978	Duane Schow	Staff	Installation of ball screws and stepper motors on lathe and mill
1978	Charles Snedd	CAM, M.S.	Tectronix 4051/MACSYM/
1980	Val Hubbard	Mfg. Tech	Parametric design, TEK 4051
1980	Doug Bushman	Ind. Design	Parametric design, Apple II
1981	Marvin Quist	CAM, M.S.	AS/RS design and evaluation
	Forest Blair	Mech. Eng.	Design of equipment
<u>After Grant</u>			
1982	Forest Blair	Mech. Eng.	Design of equipment
1982	Justin Redd	El. Eng.	Stepper motor drives
1983	David Jespersen	Mfg. Tech.	Programming, Demo System
1983	Bruce Ross	Design Tech.	Mechanical Draftsman
1983	Tim Ward	Mfg. Tech.	Equipment Fabrication
1983	John Oliver	Mfg. Tech.	Equipment Fabrication
1983	Greg Peto	El. Eng.	Programming
1983	Stanley Livingston	El. Eng.	Mechanical Drafting
1984	Darin Mathews	Mfg. Tech.	Equipment Fabrication

END

FILMED

5-85

DTIC