CONSTRUCTION AND OPERATION OF THE MAE FLEXIBLE USE WIND TUNNEL FACILITY

By

JUSTIN LAWRENCE CATES

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1992

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
April, 1996

CONSTRUCTION AND OPERATION OF THE MAE

FLEXIBLE USE WIND TUNNEL FACILITY

Thesis Approved:

____
Thesis Advisor

____

____

____
Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my major advisor, Dr. Andy Arena for first providing me with the opportunity and financial support to pursue this avenue of research, and also for his dedication to this field, which inspired me to pursue this degree, and finally for his guidance, patience and friendship.

I would like to thank Dr. Frank Chambers for his suggestions, supervision and the use of his equipment and resources during my research. I would like to thank the North Lab manager, James Davis and his staff for the assistance throughout my work. I would also like to thank the entire Tapered Tube staff for their friendship and assistance.

I would like to express my appreciation to my wonderful wife, Jodie, for believing in me, for her encouragement, and for her patience and understanding while I pursued this degree. I would also like to thank my family, as well as my in-laws for their support and encouragement throughout the process.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Background

Indraft wind tunnels consist of a contraction cone, test section, diffuser, blower section, and an exhaust. Air is extracted from the atmosphere, drawn through a testing chamber, and returned to the atmosphere. An indraft tunnel utilizes a blower downstream of the inlet. The blower draws air through the test section, as opposed to a blow down tunnel, which has a blower upstream of the inlet. This paper will focus only on the indraft wind tunnel. Figure 1 is a preliminary drawing of the wind tunnel facility that is the topic of this paper.

The degree of flow quality of a particular wind tunnel depends upon many factors, some can be controlled and some cannot. Atmospheric disturbances can greatly influence the air flow in the test section, however, the atmosphere obviously cannot be controlled. Research grade flow quality can be achieved through the careful design and construction of the contraction cone, test section, diffuser, and blower section of the wind tunnel.

The contraction cone will reduce turbulence fluctuations and improve the velocity profile uniformity. Flow management devices mounted at the entrance to the contraction cone will also reduce lateral and axial fluctuations. The test section should provide access for models and probes, and for flow visualization, while maintaining smooth airtight surfaces to allow for undisturbed flow through the test section. The purpose of

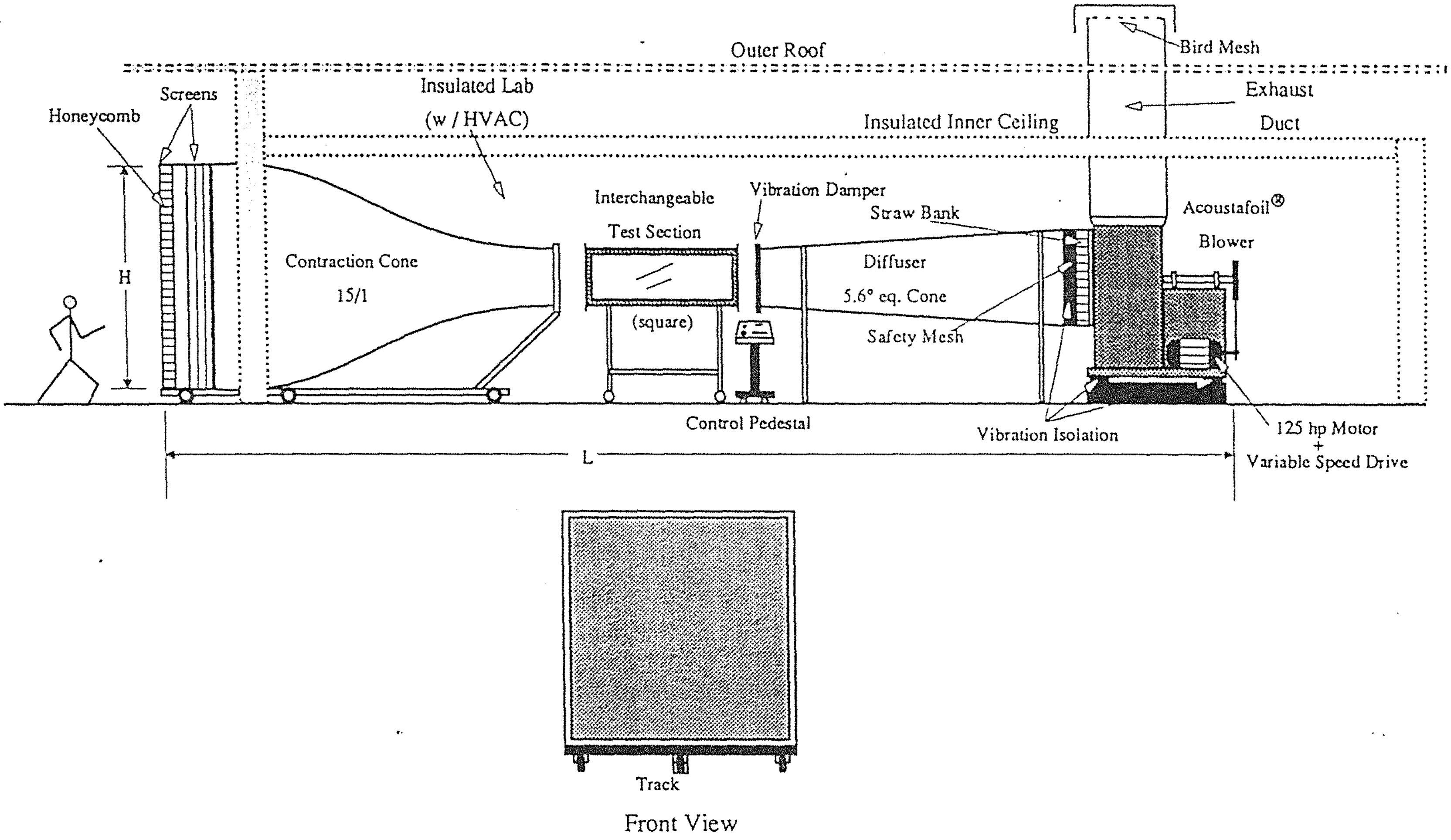


Figure 1. Preliminary Drawing of Wind Tunnel Facility

the diffuser is to decelerate the flow and recover static pressure loss after the air leaves the test section. The area ratio and the expansion angle need to be designed to avoid separation along the diffuser walls. Choosing a blower with minimal noise and blade passage disturbance can also increase the flow quality in the test section.

The contraction cone, diffuser section, and exhaust were designed previous to my involvement in the project. The blower type to be used was also chosen at this time. The contraction cone has a 12 ft. x 12 ft. inlet and a 3 ft. x 3 ft. exit. The diffuser was designed to have a 5.6 ° expansion angle. The inlet of the diffuser is 3 ft. x 3 ft. and transitions gradually from square to round, where the round exit diameter matches the inlet diameter of the blower. The choice of blowers was a centrifugal type, variable speed drive, 125 hp blower with airfoil shaped fan blades. This blower choice will result in stable pressure characteristics and a reduction in noise. The blower exhaust duct reaches straight up through the roof of the building. When not in use, the duct is capped so that water or other foreign objects cannot get into the blower. A metal cap slides up and down so that when the blower is turned on, the cap is pushed up by the air flow. When the blower is turned off, the cap slides back down to rest on top of the duct.

Objectives

The result that we are working toward is a state of the art indraft wind tunnel facility with research quality air flow. Other wind tunnel facilities in the region have turbulence intensities ranging from 0.2% to 1.1%(turbulence intensity being one of the key measures determining flow quality). The performance goal of our new facility will be to achieve turbulence intensities an order of magnitude lower than others in the region, while maintaining adequate flow uniformity. We not only want to have a wind tunnel with excellent flow quality, we would also like to have the flexibility of creating different air flow situations for a variety of experiments. The plan is to utilize a modular design where the test section can be removed and another put in its place. Individual test sections can be built specifically for each specialized experiment. Another goal this design will achieve is to minimize turnaround time, since the test sections can be easily interchanged.

The tasks that I am responsible for include portions of the model catcher construction and attachment, design and construction of a test section, design and construction of a two-dimensional traverser (including electronic circuitry and computer code to automate the traverser). My responsibilities also include the incorporation of all the components and the construction and attachment of flow management devices(screens and honeycomb).

Outline of Work

The following is an outline showing the tasks performed in each specified area of the facility. This is not a complete detailed task list, as the details will be addressed in the following chapters of this paper.

I. TEST SECTION

A. Understructure

1. Design
2. Welding of the base
3. Construction and attachment of adjustable legs
4. Attachment of casters

B. Frame

1. Pattern creation to match contraction cone outlet
2. Construction of both end frames
3. Connection of end frames with cross members
4. Attachment of completed frame to understructure

C. Walls

1. Floor

    a. Cut to size

    b. Dado outside edges

    c. Cut slot for traverser

    d. Attachment to frame

2. Walls with viewing windows(front and top)

    a. Cut to size

    b. Cut out center for glass

    c. Dado inside and outside edges

    d. Adhere glass into place

    e. Attachment to frame

3. Door sidewall

    a. Cut to size

    b. Dado outside edges

    c. Door attachment using hinges

    d. Attachment of door latches

    e. Design, construction, and attachment of door props

4. All surfaces

    a. Application of putty at all metal to wood or glass junctions

    b. Removal of excess dried putty

    c. Sanding to smooth junctions

D. Paint

1. Understructure

2. Frame

3. Walls (inside and out)

II. MODEL CATCHER/STRAW BOX

A. Stand

1. Design and construction

B. Straw bank

1. Packing of straw bank

2. Attachment of retainment screen

III. TRAVERSER

A. Mechanical system

1. Design

2. Machining of parts

3. Construction

4. Attachment to test section

B. Electrical system for data acquisition

1. Circuit design

2. Circuit construction

3. Stepper motor incorporation

4. System component composition

5. Communication link to computer

6. Mechanical and electrical systems coupling

C. Computer code

1. Code for stepper motor communication

2. Modification of data acquisition code

3. Unification of both the former codes to complete the automation

IV. FACILITY COMPONENT INCORPORATION

A. Model Catcher - Diffuser Junction

1. Flange vibration isolation and air flow seal

2. Attachment bolt vibration isolation

B. Test Section - Diffuser Junction

1. Test section height adjustment

2. Air flow seal

C. Contraction Cone - Test Section Junction

1. Test section leveling

2. Contraction cone height adjustment

3. Air flow seal

D. All Separate Components

1. Cleaning of all internal surfaces

2. Caulking of cracks for air flow seal

V. FLOW MANAGEMENT DEVICES

A. Screens

1. Construction of frames

2. Stretching of screens

3. Attachment to contraction cone

B. Honeycomb

1. Construction of frame

2. Design and construction of honeycomb-loading implement

3. Attachment of frame to contraction cone

4. Loading of honeycomb sections

VI. PRELIMINARY TESTING AND DATA

A. Test Runs

B. Flow Uniformity Tests

CHAPTER 2

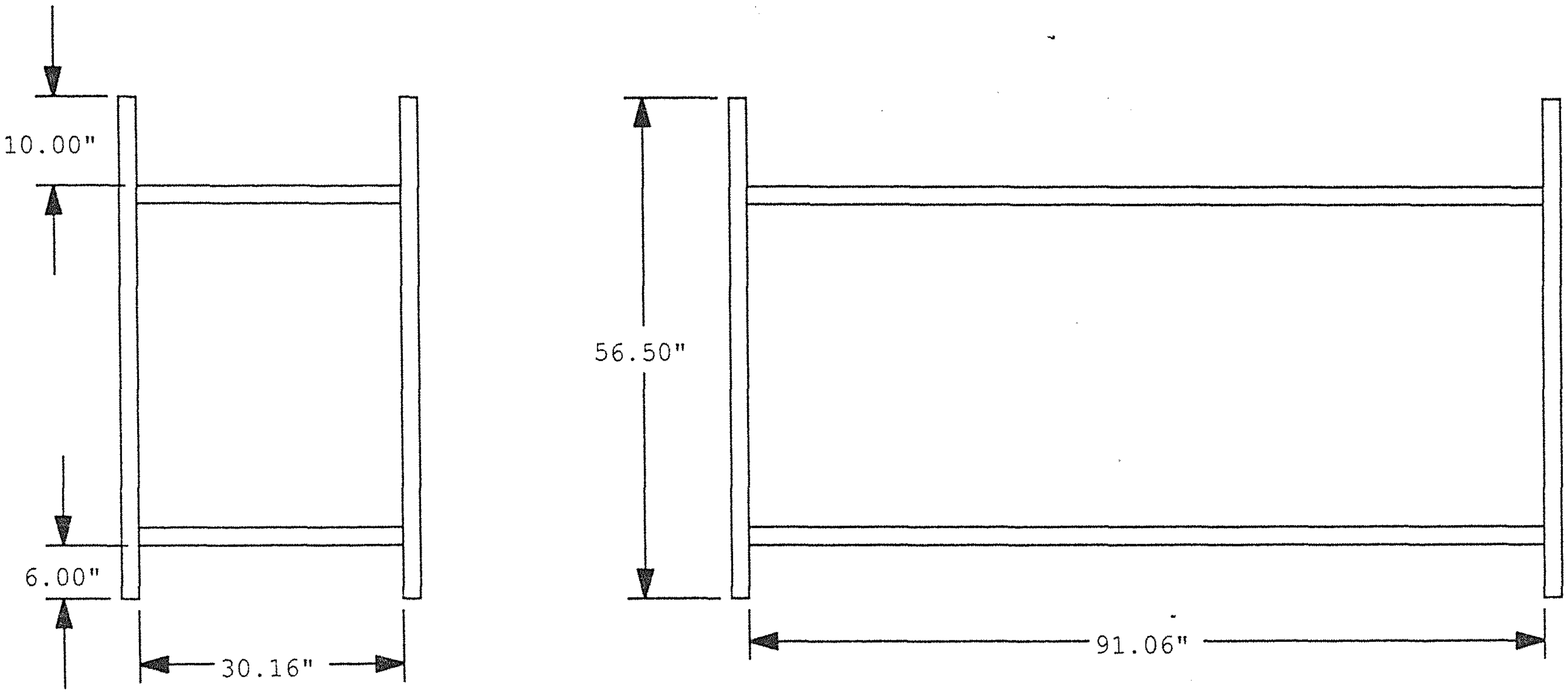
CONSTRUCTION

Test Section

The test section is the region in which the controlled flow must be achieved and where experiments will be performed. In order to maintain the quality of flow throughout the test section, flow disturbance must be minimized. The transitions between the contraction cone and inlet of the test section, and between the outlet and the diffuser must be as smooth as possible, while also providing an airtight seal to prevent flow leakage. Junctions between wood and glass or metal must also be made as level as possible. Access into the test section should be provided for the insertion of models and probes to conduct experiments. The base of the test section should furnish adequate support and stability and since this test section is to be portable, must also provide for caster attachment and height adjustment mechanisms. This particular test section design will meet all the above requirements and also allow for the removal and replacement of any of its walls without disturbing the others. This will allow the experimenter to cater a floor or wall design to a particular application.

Base

The base was constructed entirely of 2 in. x 2 in. steel square tubing. The dimensions of the completed base along with a material list are shown in Figure 2. As shown, the vertical members used were 56 1/2 inches in length. In retrospect, 56 inches would be a better choice, as this would allow more room for height adjustment. The materials for



All members are
2" x 2" steel
square tubing

Figure 2. Test Section Base

one end of the base were laid out, squared up, and tack welded. The squaring of each joint was achieved using a steel square. Each joint was double checked for squareness using the steel square, adjustments made if necessary, then the welding was completed. This process was repeated for the other end and then the horizontal members were welded on. Again, every effort was made to maintain squareness at every joint.

Casters

The casters that were used had square attachment plates with bolt holes in each corner. Four steel plates were cut to this same size and holes drilled to match the hole pattern on the caster plates. Two steel plates were welded onto each of the bottom horizontal end members of the frame allowing just enough room for the casters to rotate freely without contacting the vertical leg.

Height Adjustment

An adjustment mechanism was constructed for each vertical leg of the base. To produce these mechanisms, four Carr Lane jig leveling feet were utilized. A 4 in. x 2 1/2 in. flat metal plate was cut and a 41/64 in. hole drilled in the center of the plate 3/4 in. from one end. The hole should be sized so that the threads of the jig foot are not in contact with the sides of the hole. The jig foot was then inserted through the hole and the nut threaded on just enough so that the top of the bolt was flush with the top of the nut. This should protect the threads from welding slag. The end of the plate was then clamped to the edge of a table top so that the jig foot hung from the plate with the nut sitting flush and centered over the hole. The nut was then welded to the plate.

Adjustment levers were made utilizing sliding rods similar to a vise lever. To create these levers, 1/4 in. diameter steel rod was used along with 1 in. long pieces of pipe with inside diameter just large enough to allow the rod to slide through nicely. With the jig foot threaded through the nut, a section of pipe was then welded perpendicular to the end of the bolt. The rod was cut to 6 in. lengths and one end of each rod was mushroomed slightly so that it would not slide through the pipe. The rod was mushroomed by placing the rod in a vise, heating the end, and hitting it with a hammer. The end of the rod that was not mushroomed was then inserted through the pipe and mushroomed in the same fashion. The free end of the flat plate was then welded onto the bottom of the vertical legs of the base with the plate extending underneath the long side of the base. Triangle brackets cut from steel plate were then welded to the vertical leg and the plate to support help support the weight of the test section. Figure 3 shows a completed and attached adjustment mechanism along with an attached caster.

Frame

The frame for the test section is made entirely of 3 in. x 3 in. angle iron. Each end of the frame was constructed to form a flange. The flange on the entrance end of the test section will be connected directly to the flange of the contraction cone outlet and the flange on the outlet will be connected to the diffuser. The dimensions of the flanges must match the dimensions of the contraction cone outlet to maintain flow quality throughout the test section. To accomplish this, a pattern was made of the contraction cone outlet. The pattern was made using a 3/4 in. thick sheet of plywood. The sheet was pattern was cut out and the each edge sanded until the pattern snugly fit inside the outlet. For ease of handling, a handle was attached to the center of the pattern. Since the dimensions of each side are slightly different it is important to label the sides of the pattern and use it exactly in the orientation in which it was made.

Using the pattern, a piece of angle iron was cut to match one of the side lengths of the pattern. This was repeated for the other side, making sure to label each piece to avoid interchanging them. Again, using the pattern, the top and bottom pieces were measured by putting the side pieces in place beside the pattern and laying out a piece of angle iron on the top or bottom of the pattern. One end of the angle iron was made flush with the outside edge of one of the sides and a mark was made at the outside edge of the other side piece. After both top and bottom were marked and labeled, the cuts were made. Figure 4 shows all four cut pieces laid out around the pattern. Notice that the top and bottom piece verticals extend past the intersection with the side piece verticals. Since this was not desirable, the pattern was again used to situate the four sides and mark the intersection down to the web of the angle iron. This excess was then cut off. Cutting off the excess will facilitate the welding of the corners.

The flanges could now be welded. The pieces were clamped into position around the pattern using two long adjustable clamps. Once the positioning was satisfactory the joints parallel to the floor were tack welded. The positioning was double checked, the pattern removed and the welds finished. The upright corners should be left unwelded until the flanges are attached to the base of the test section. Figure 5 portrays the resulting flat flange with square corners.

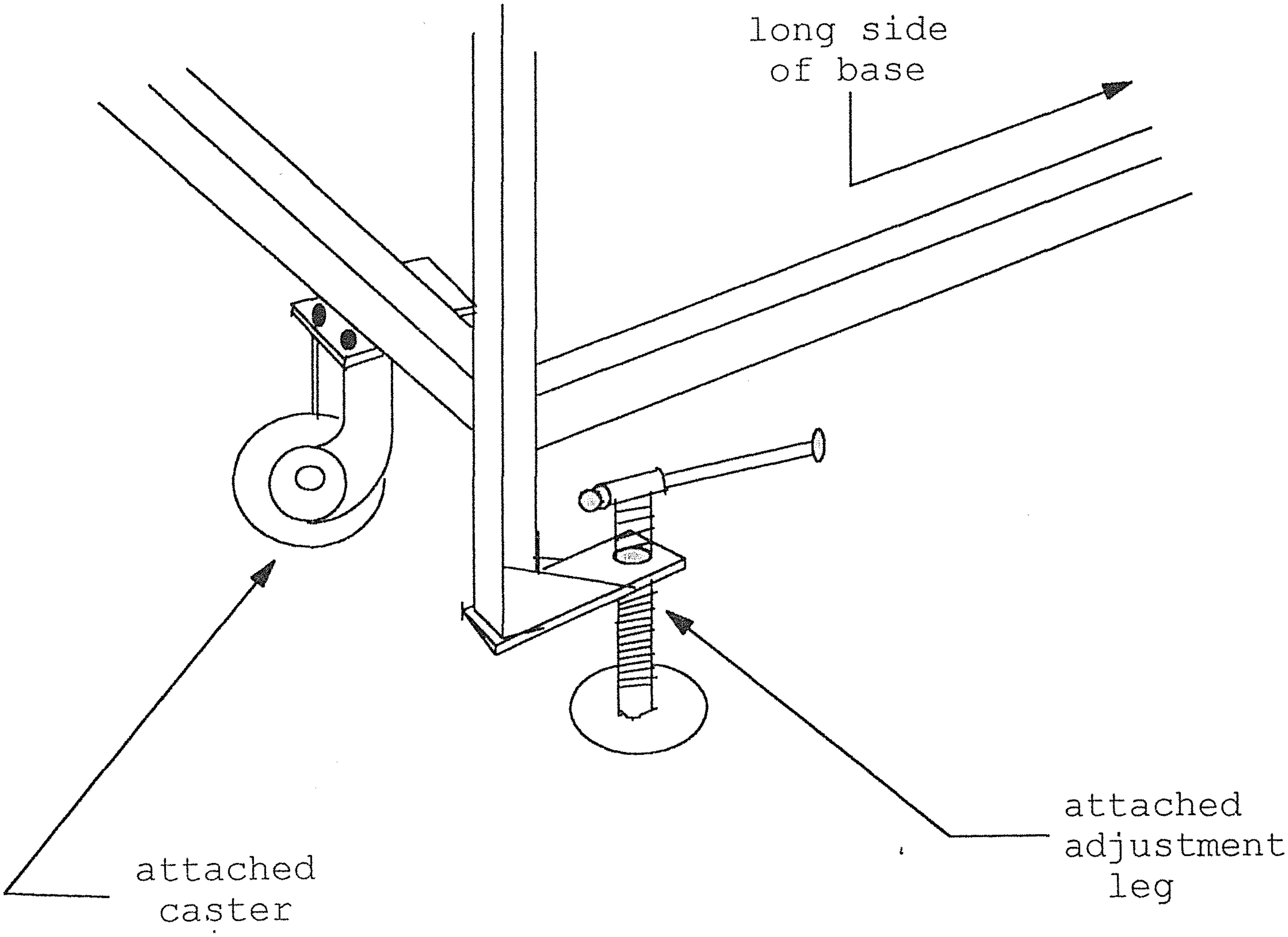


Figure 3. Adjustment Leg and Caster

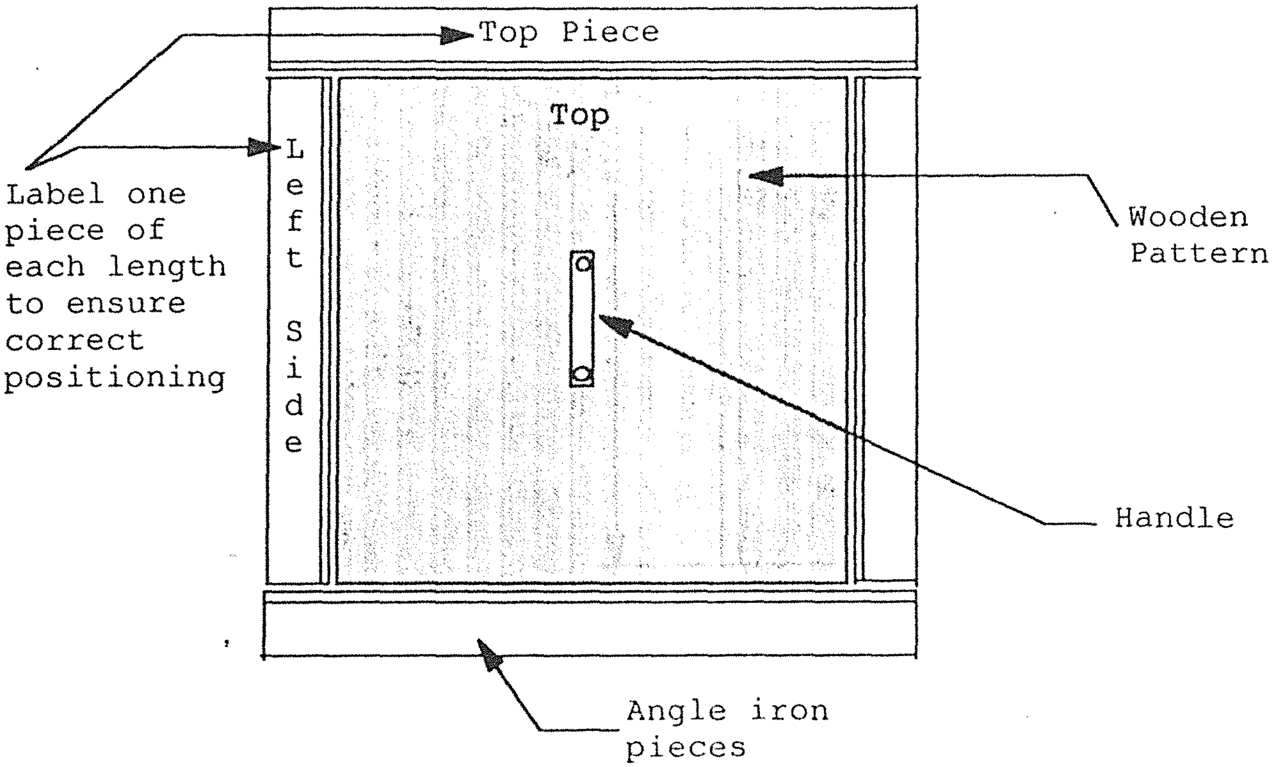

Top Piece
Top
Label one piece of each length to ensure correct positioning
Left Side
Wooden Pattern
Handle
Angle iron pieces

Figure 4. Flange Pattern with Angle Iron

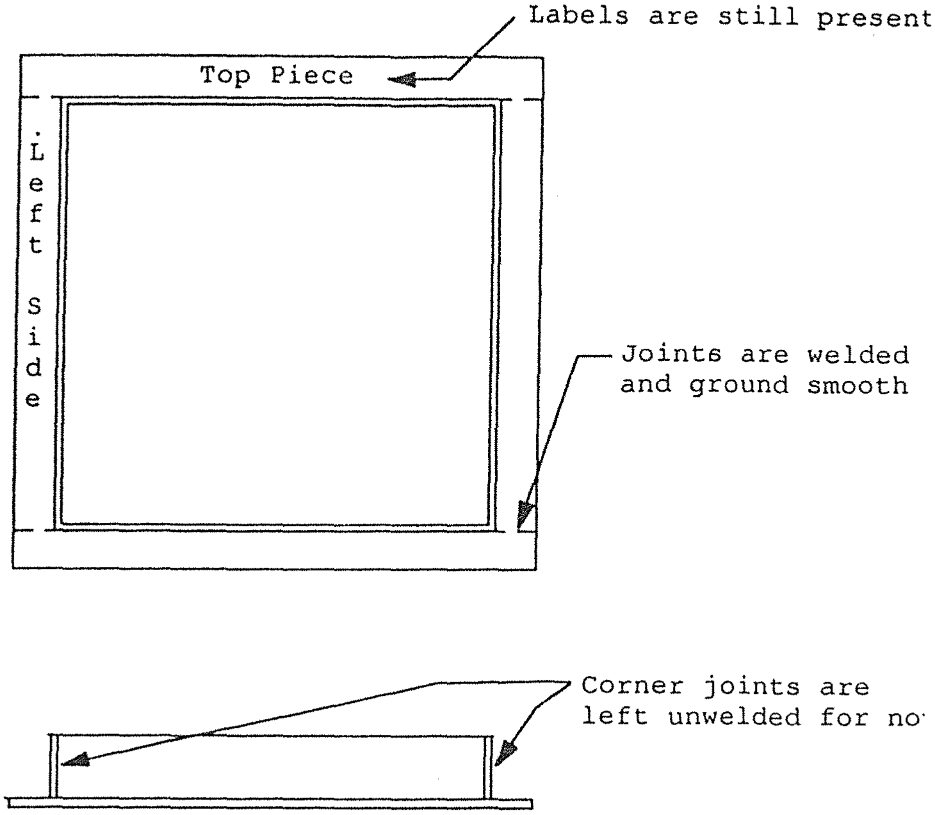

Labels are still present
Top Piece
.Left Side
Joints are welded and ground smooth
Corner joints are left unwelded for no·

Figure 5. Finished Flange

The finished flanges were centered and clamped on top of the base with the flat face of the flange facing away from the base on both ends. The face of each flange was then leaned so that it was parallel with the uprights of the base. Ease of manipulating the slope of the flanges was the reason for leaving the corners unwelded. After the faces were parallel, the unwelded corners of the flanges were welded. Figure 6 shows the base with the flanges clamped to it. The horizontal members of the frame were created by welding four 90 in. lengths of angle iron between the corresponding corners of the flanges. Welds and any bumps created by these joints were ground flat since this was to be the air flow surface. Once the horizontal members were welded into place, the frame was welded to the base.

Test Section Walls

The walls of the test section were cut from 3/4 in. thick, medium density fiberboard. This test section was designed so that the walls would be mounted from the outside. This would allow the removal, modification, and replacement of any wall without disturbing the other three walls. Since the walls were to be mounted from the outside, directly to the angle iron, the perimeter of each wall had to be recessed the thickness of the angle iron. This would make the fiberboard wall flush with the metal corners and ends. A router and the use of a long carefully placed fence helped assure that the result would be flush, snug fitting walls.

The top and viewing side wall were to have large windows. The glass to be used was 1/4 in. thick safety glass. In order for the glass to be flush with the wall, it must be mounted from the inside. Epoxy would be used to hold the glass in place. A perimeter width for adhesive was decided upon and the holes for the glass were cut in the top and side wall. More surface area for adhesive was allowed on the top wall since the adhesive would be the only thing holding it in place. The 1/4 in. recessed perimeter was achieved in the same manner as the outside recesses. This process was complete when the glass snugly fit into the recess and was flush with the rest of the board. The glass was removed, epoxy was applied, the glass put back in place, then the epoxy was allowed to set.

The door of the test section was designed so that the entire wall would swing from hinges attached at the top. Three purchased latches attached to the bottom of the tunnel floor worked nicely to hold the door in place. The door was quite heavy, so a door prop was constructed for both ends of the door and attached to the base of the test section.

Holes were drilled in the angle iron to attach the floor, top, and viewing side wall. The holes were countersunk and flat head screws were used so that the screws would be flush with the surface.

Seams where fiberboard met metal or glass could possibly cause discontinuities in the flow or allow for flow leakage. This problem was solved by filling any seams with wood putty. The putty was allowed to dry, the excess was scraped off, then the seams sanded smooth to give a nice inside surface.

Model Catcher/Straw Box

The model catcher consists of a large metal cylinder with a large-mesh screen (chicken wire) stretched across its cross section. This screen will serve as protection for

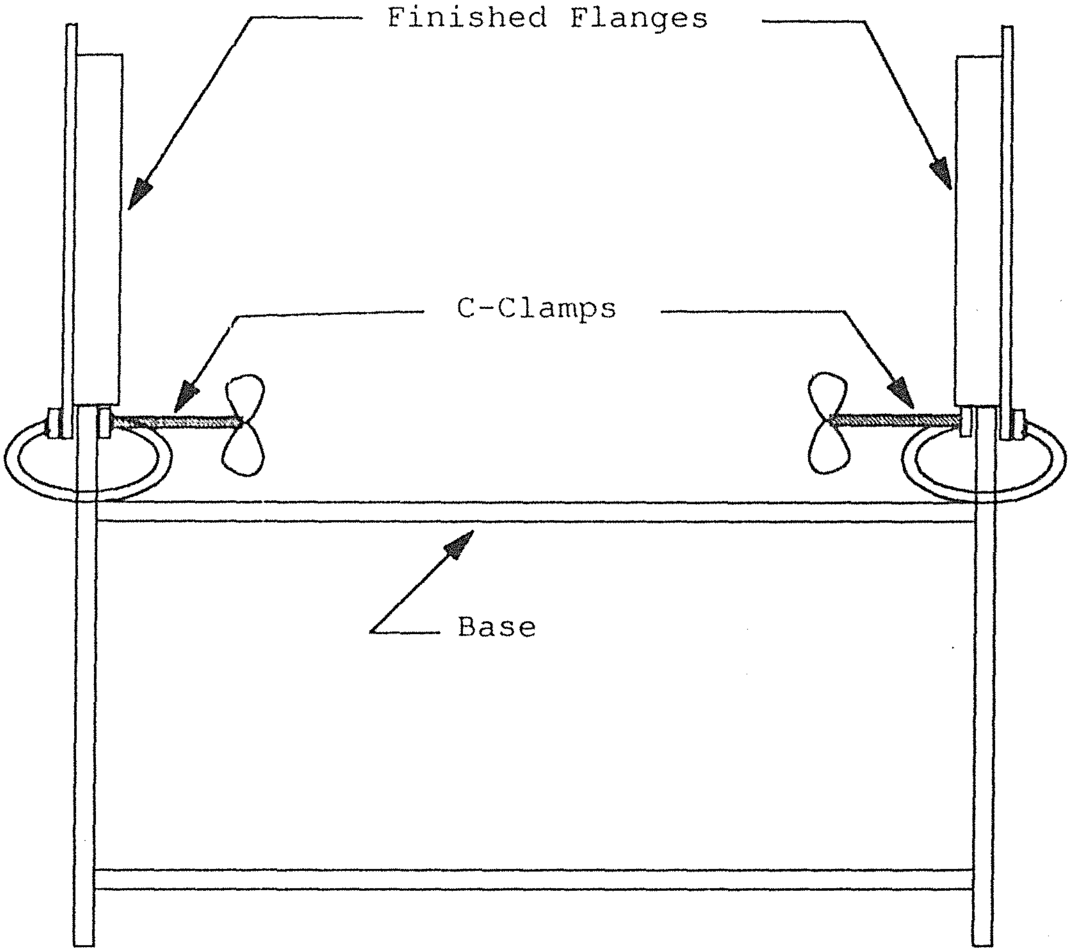


Figure 6. Test Section Base with Flanges Clamped in Place

the straw bank and the blower. The screen should stop flying debris resulting from a mishap upstream such as a model being sucked down the tunnel.

The straw box is a metal cylinder packed with drinking straws. Its purpose is to deter swirl that may originate in the blower from propagating upstream to the test section. Both the model catcher and the straw box have metal flanges on both ends of the cylinders for attachment purposes. The straw box cylinder will be attached directly to the model catcher cylinder with the straw box situated on the downstream side.

Stand

Since the straw box end of the attached cylinders will be bolted directly to the blower flange, it will be supported. However, because of the length and weight of the cylinder, the model catcher end will need support. This support was provided by designing and constructing a stand out of steel square tubing. The stand consisted of two upright legs, two cross members, and threaded adjustment feet. Figure 7 exhibits the completed stand.

Straw Bank

The straw bank is contained in the straw box cylinder by two fine-mesh screens. Before attaching the straw box to the model catcher, the screen on the upstream side was stretched across the cylinder. After the cylinders were attached, the straw bank was created by packing approximately 70,000 drinking straws into the cylinder with the straws axially aligned in the air flow direction. It is important that as many straws as possible be packed into the cylinder. This is important because the straws will settle over time and use of the wind tunnel. If the straw bank settles too much, a gap at the top occurs allowing the straws to move around which could lead to misaligned straws. Gaps or misalignment of the straws in the straw bank can cause a deterioration of the flow quality upstream in the test section.

Traverser

To accommodate the desire for the ability to take flow measurements over the cross section of the test section, a two-dimensional traverser was designed and constructed. The initial intent was to build a manually operated traverser, however, related involvement in other projects led to the decision to automate the traverser.

Mechanical System

The parts used to construct the horizontal component as well as the vertical component of the traverser were machined in-house with the exception of the lead screws and the lead screw nuts.

The horizontal component was constructed using mostly aluminum with the addition of two stainless steel rods. A rectangular block of aluminum was machined to serve as the translating part. Holes were drilled through the block near both sides to allow the block to slide on the stainless steel rods. Two smaller aluminum blocks were used for end blocks. The end blocks also had holes drilled to insert the rods into. The holes should be drilled in the translating block and end block at the same time if possible. This can be done by clamping them all together, then drilling the holes. This will ensure that the separation distance between the holes remains constant on all the blocks. Set screws were inserted perpendicular and into the holes on the end blocks to secure the desired position of the end blocks along the rod length. Aluminum plates were attached to the

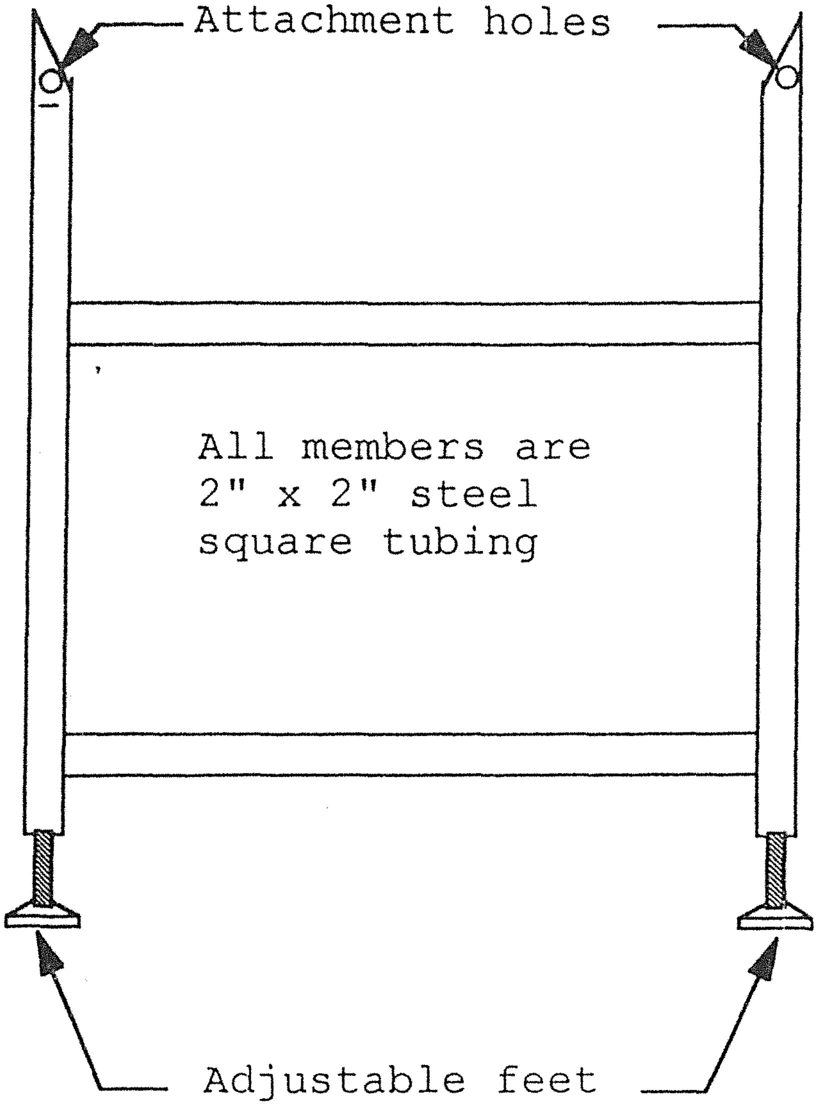


Figure 7. Model Catcher Support

end blocks on both ends to house the bushings in which the horizontal lead screw would turn. The plate on one end must also be able to accommodate the mounting of a stepper motor that will turn the lead screw. Figure 8 shows the horizontal component isolated from the vertical.

The translation block was temporarily removed from the horizontal component so that it could be used to create the vertical component. The vertical component frame was constructed of a long aluminum plate extending downward from either side of the translation block. A small plate connected the two long plates at the bottom. This bottom plate would act as the stepper motor mount. Small narrow aluminum strips connected the long plates at several locations to stabilize them. The lead screw will extend from the vertical stepper motor on the bottom plate to the translation block. At this point on the bottom of the translation block a depression was drilled in which the lead screw could rotate. As the stepper motor turns the lead screw the lead screw nut travels up or down accordingly. A small strip of aluminum was attached directly to the nut so that the apparatus to be translated was cantilevered from the nut. Since it is cantilevered the mounting of the apparatus (a pitot-static tube in this case) should be as close to the lead screw as possible. The pitot-static tube has one outlet that is in-line with the tube and the other is perpendicular. A rectangular block of aluminum was machined to accommodate the probe. A hole was drilled down the length of the block for the probe to rest in and to allow vinyl tubing to be connected to the in-line outlet. A short slot was also machined at the top of the block so that the perpendicular outlet slid into the slot. A set screw at this level on the opposite side of the block was inserted to maintain the probe alignment. A hole in the center of the translation block was drilled for the probe to travel

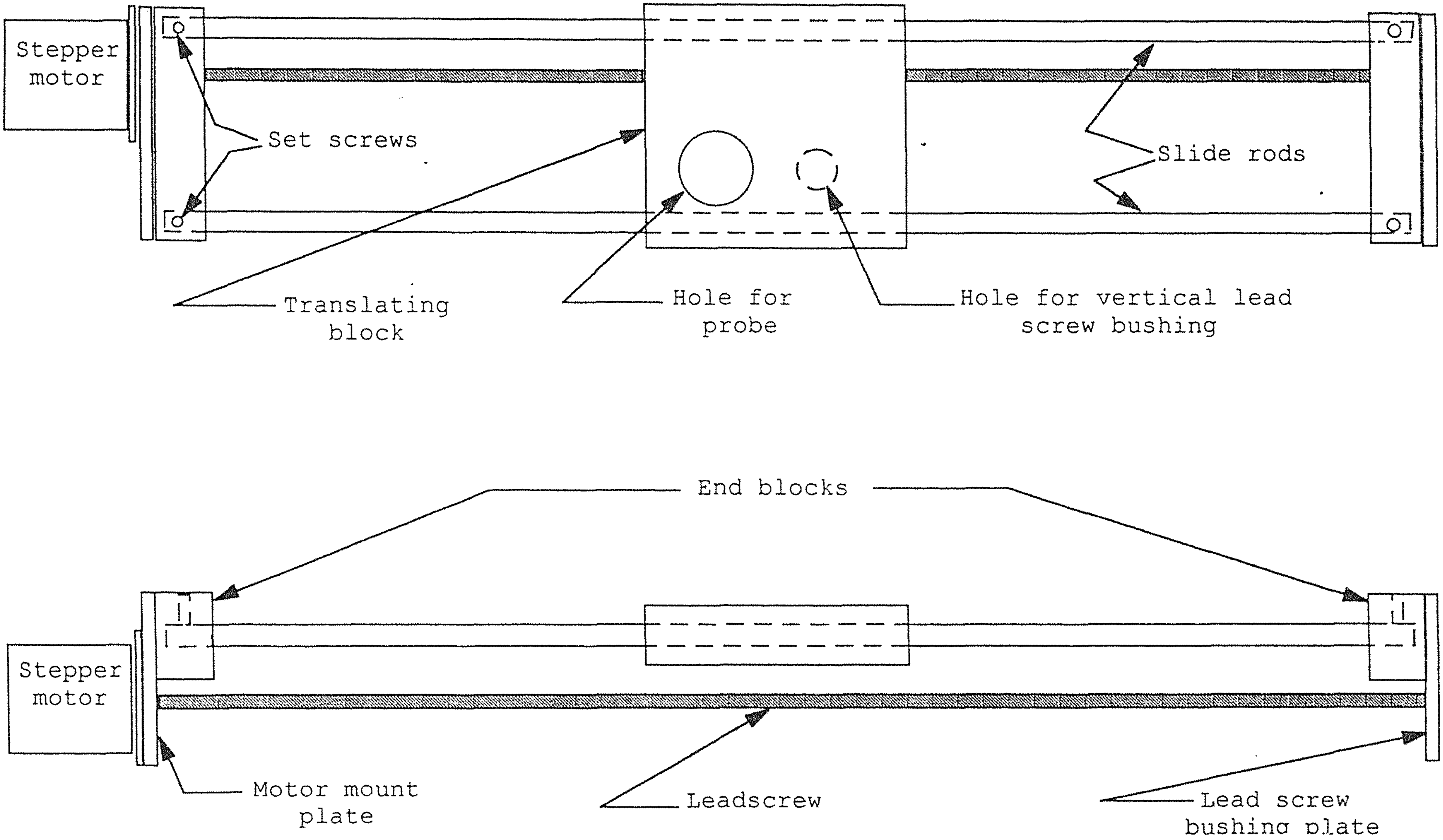


Figure 8. Horizontal Component of Traverser

through. This hole was tightly lined with foam to support the probe while reducing noise. The constructed vertical component is shown in Figure 9.

The horizontal lead screw would pass through drilled holes in the long vertical plates at a point just below the translation block. The horizontal lead screw nut was attached to the aluminum around the hole of one of these plates. The two components were combined to complete the traverser. The apparatus is shown in Figure 10.

To complete the incorporation of the traverser with the test section a slot had to be cut in the floor of the test section for the measurement probe to travel in. The traverser was then attached to the bottom of the floor of the test section by drilling holes through the end blocks and using screws to secure it to the wood floor.

Electrical System

The tasks of interfacing the stepper motors with a computer and automating the data acquisition process required circuit design and construction, computer communication linkage, system component composition, and some system troubleshooting and repair. The combination of purchased components and in-house constructed circuitry provided the necessary hardware to communicate with and run the stepper motors. The purchased components consisted of a control board, power supply, two stepper motors, and various connections (DB9's, banana plugs, etc.).

The completed system would utilize a 24 volt power supply that would power both the control board and the stepper motors. The control board would receive information from a computer and relay it to the stepper motors via the stepper motor driver circuits. The

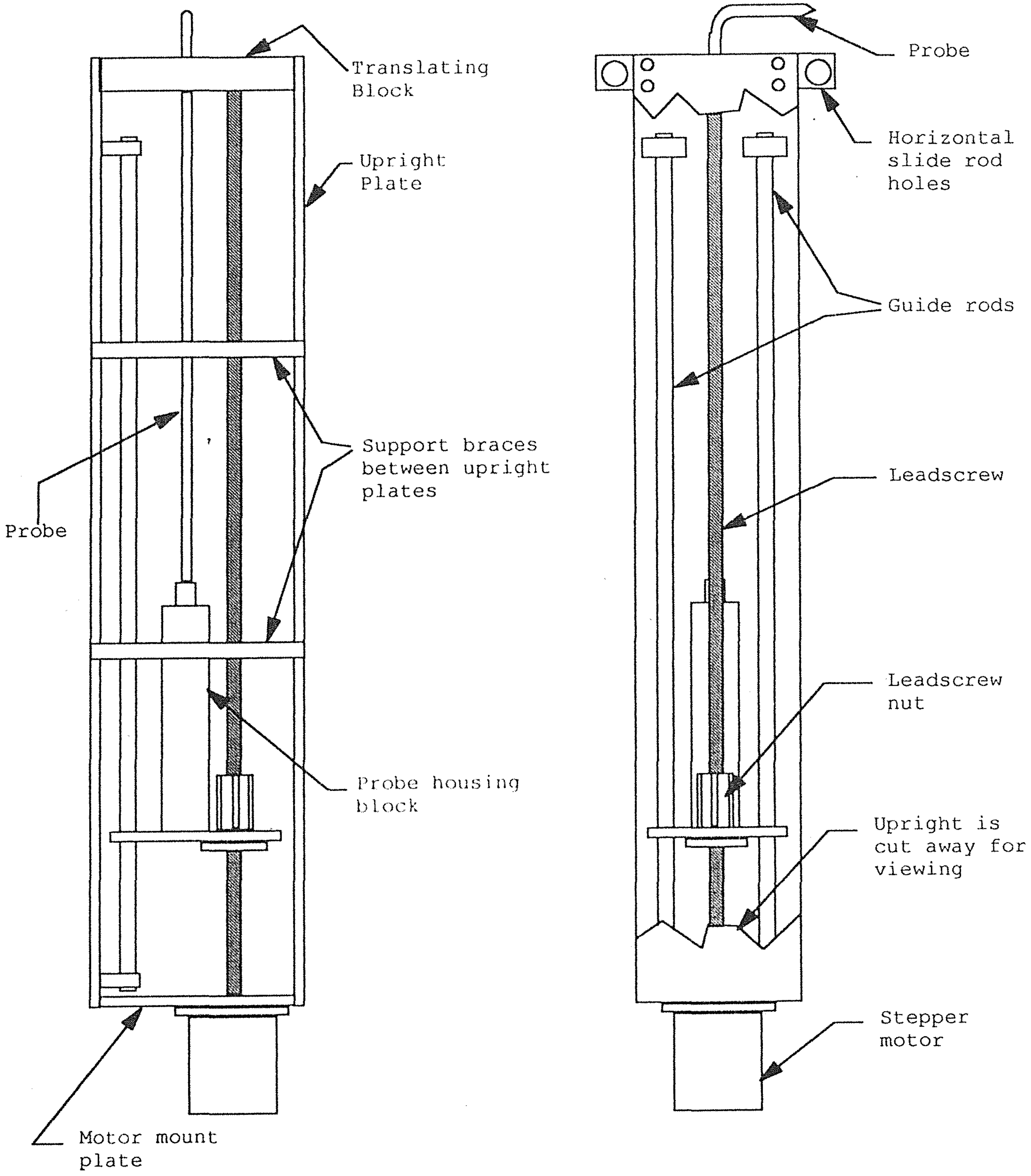


Figure 9. Vertical Component of Traverser

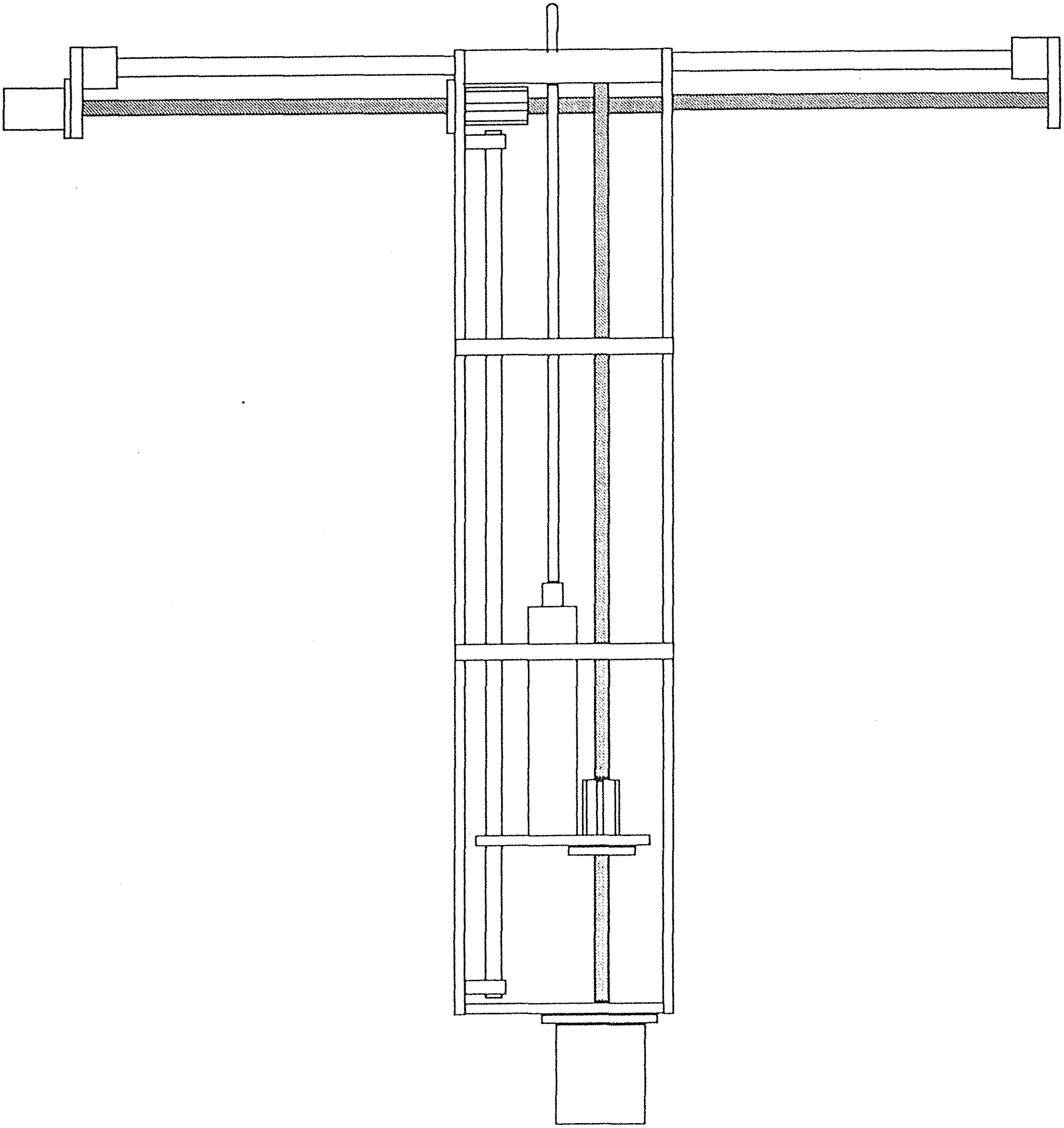


Figure 10.   Completed Traverser

details of the construction of the driver circuits and their purpose will be discussed later. An early stage sketch of the system is shown in Figure 11.

Initially the power supply and control board were each mounted in enclosed instrument boxes. External plugs were wired in and attached to the outside of each box to make the necessary connections between components easy. The computer to control board connection was wired in and the power supply connected to the stepper motors and the control board. Since the stepper motors are run on 24 volts the power directly from the power supply is fine for them, however, the control board runs on 12 volts, therefore a voltage regulator was placed in the circuit between the power supply and the control board.

As mentioned earlier the driver circuits are located between the control board and the motors. These circuits utilize transistors to switch the signals high and low to communicate the desired information to the motors. An identical driver circuit was used for each motor, but both circuits were constructed on the same circuit board. A schematic of one of the driver circuits is shown in Figure 12. A noisy input signal did not allow the transistors to switch on and off correctly using the factory recommended resistance preceding the transistors. These resistors were reduced from 1 kohm to 100 ohm so that the transistors received a "cleaner" signal. The circuit board was mounted in an experiment box once the construction was complete. Input and output connections for both circuits were attached to the box panel for ease of connection.

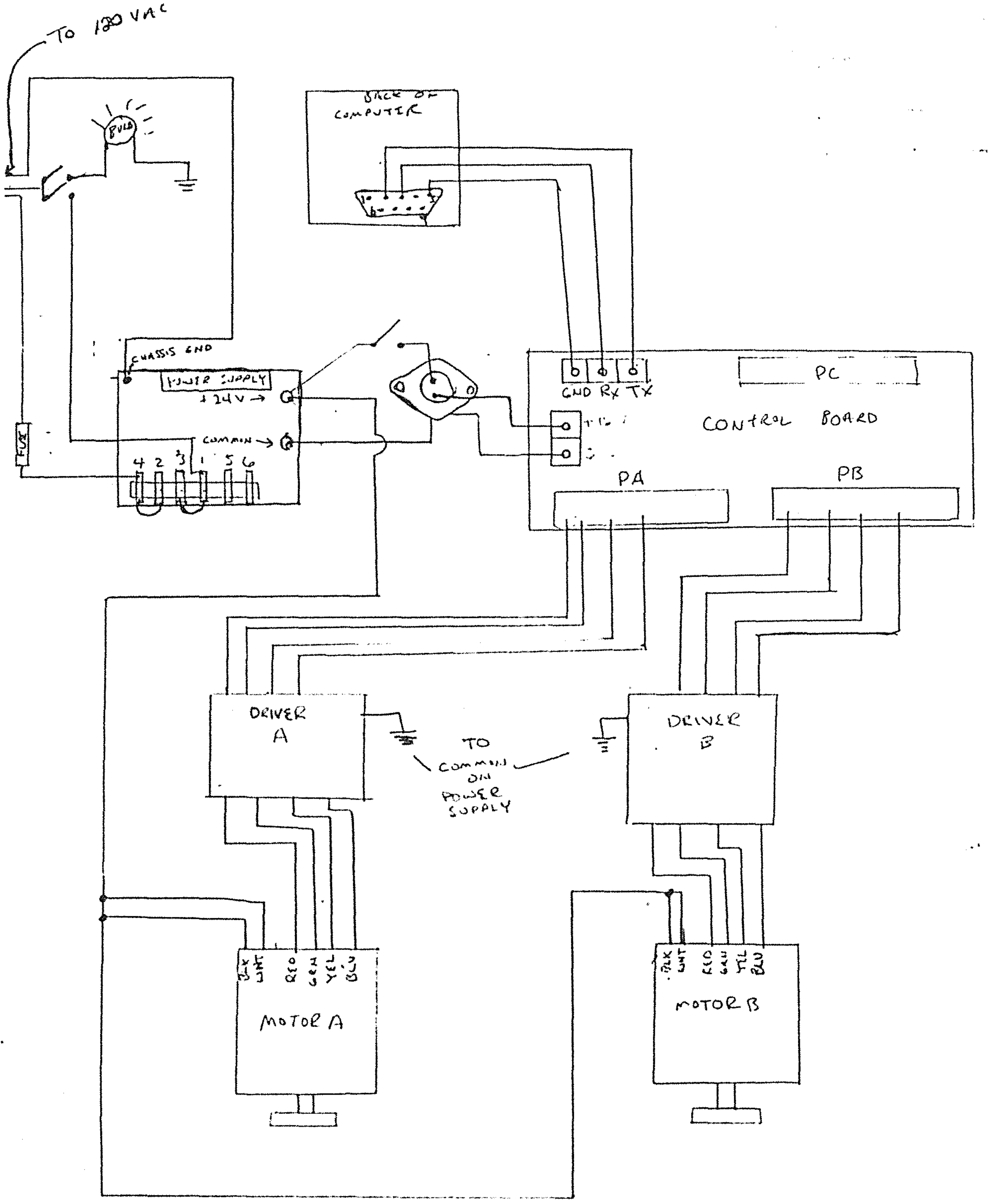
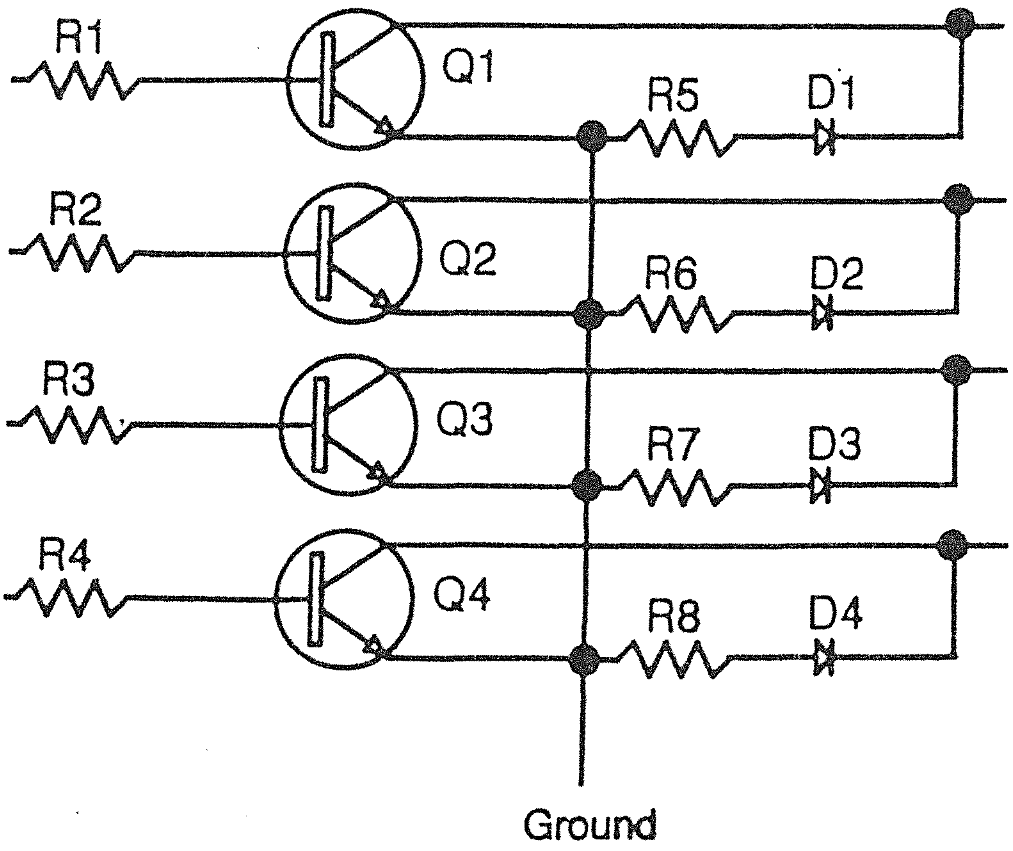


Figure 11. Early Stage Sketch of the Traverser Electrical System



Values Used:

R1-R4 = 100 Ohm, 1/4 Watt
R5-R8 = 330 Ohm, 1/4 Watt
Q1-Q4 = MJE3055T
D1-D4 = 1N4003

Figure 12. Stepper Motor Driver Circuit

Computer Code

Communicating with the stepper motors from the computer required the use of the control board. A list of commands was included with the control board. The list provided information on the commands that the control board recognized and the result of sending one of these commands. The different commands allowed the user to vary certain parameters such as the stepper motor speed, the number of steps to execute, and the delay between the execution of each command. This information was used to write a brief code to test the system. The test code can be found in the Appendix. This brief code would set the foundation for what would eventually be incorporated into the data acquisition code. To supplement the test code as well as the final code, a detailed description of all commands necessary to communicate with the stepper motors, including how to set the stepper motor configuration and control their operation, can be found in the ITC232 User Guide, which should accompany this paper at the test section.

Existing data acquisition code was made available, however, this code was written so that each reading was initiated manually. Allowing the user to accept or reject each measurement individually is very helpful when taking a small number of readings, however, when a large matrix of readings is necessary this method requires too much time. The existing code was therefore modified to perform a matrix of readings at intervals specified in the modified code. The process would only need to be initiated. The variables in the loop could be changed to alter the size of the matrix, pattern of execution, or size of the interval to meet the specific needs of the experiment. To allow the user to monitor the experiment while it is running, each reading, along with the position of the probe for that reading, is printed to the computer screen. All readings are also recorded and sent to a file for later retrieval. A message after each data reading tells the user how to stop the program, if there is a need to do so. The experiment can be reset quickly and restarted immediately.

To complete the automated data acquisition code the stepper motor communication code was incorporated into the modified data acquisition code. Once this was completed the data acquisition loop would take a reading, move the traverser, delay for the flow to stabilize, and take the next reading. The process would continue until the desired matrix was complete, then the code would give the commands necessary to send the traverser back to its home position. The complete incorporated code can be found in the Appendix.

Facility Component Incorporation

Once all the separate components were constructed the task of connecting the components was addressed. Leakage and vibration would be the main considerations while attaching the components to one another. Since the blower was already in place in a permanent position, and the other components were adjustable, the order of attachment would begin at the blower. One component at a time would be adjusted and attached, so that when everything was connected the centerline height would be consistent with that of the blower.

Blower - Model Catcher/Straw Box Junction

A forklift was used to raise the model catcher until its flange matched up with that of the blower. The bolt holes were aligned and the flanges were bolted directly to each other. The support was bolted into place and adjusted so that the bolts would not have to carry the load that the cantilevered model catcher would produce. Once the support was in place the forklift was removed.

Model Catcher/Straw Box - Diffuser Junction

At this junction the metal flange of the model catcher would be attached to the fiberglass flange of the diffuser. Since no vibration isolation was added between the blower and model catcher, the vibration caused by the blower would be present in the model catcher flange. To stop the vibration from propagating further upstream, self adhesive foam weather stripping was applied to both flanges. The diffuser was then adjusted to the correct height and slid into place. When bolted together, the compressed foam rubber would not only act as a vibration damper, but would create a seal at the junction to minimize leakage. The weather stripping would take care of the vibration from flange to flange, however, vibration could also be transmitted through the bolts connecting the flanges. Rubber grommets inserted into the bolt holes of the model catcher flange would isolate the bolt head from the metal flange as well as the shaft of the bolt from the inside of the bolt hole. The combination of measures taken should greatly reduce the possibility of vibration propagating upstream to the test section.

Diffuser - Test Section Junction

The end of the diffuser to be connected to the test section was first adjusted to maintain a consistent centerline height. Weather stripping was applied to the flanges of the test section and diffuser in the same manner as the previous junction. The foam rubber would again minimize air leakage and provide extra vibration isolation. The height of the test section was then adjusted and the components attached.

Test Section - Contraction Cone Junction

Before attaching the components, the unattached end of the test section was adjusted to maintain the centerline height. Just as with the two previous junctions, weather stripping was applied to both flanges. The contraction cone was then adjusted to provide the best possible connection between the two components. This step would complete the connection of all the major components of the facility.

All Components

In order to prepare the facility for use, all accessible internal surfaces were cleaned. As described earlier the component junctions were sealed using foam rubber. However, cracks in the corners of the diffuser and contraction cone presented an additional major area of concern for leakage. This problem was solved by caulking the corners of these two components with an every day, tub and tile caulk.

Flow Management Devices

To increase flow uniformity and lower the turbulence level in the test section, screens and a honeycomb section were placed upstream of the contraction cone inlet. The screens will reduce velocity fluctuations in the direction of the flow resulting in uniform axial velocity profiles. While the screens will manage turbulence in the axial direction they will not control lateral velocity fluctuations. To control these fluctuations a section of honeycomb was placed upstream of the screens. Since the screens and honeycomb will collect dust with use, it was necessary to consider the ease of removing them for periodic cleaning.

Screens

Frames for the screens to be stretched onto were built out of 2x4's with dimensions so that the inside edges of the frames were flush with the walls of the contraction cone. Metal brackets were used to fasten the 2x4's at the corners of the frames. Because of the limited distance from the contraction cone floor and the outside square tubing braces, notches had to be cut to allow the bottom boards of the frames to be flush with the contraction cone floor. A frame with notches cut is shown in Figure 13. Once the frames were constructed the screens were to be stretched onto the frames and attached to the contraction cone one by one. Special built stainless steel screens of very fine mesh were purchased from a manufacturer that constructed the screens in a way that minimized the seam width. Only two of these screens would be used as they were very expensive. These two screens would be placed furthest downstream since they were of superior

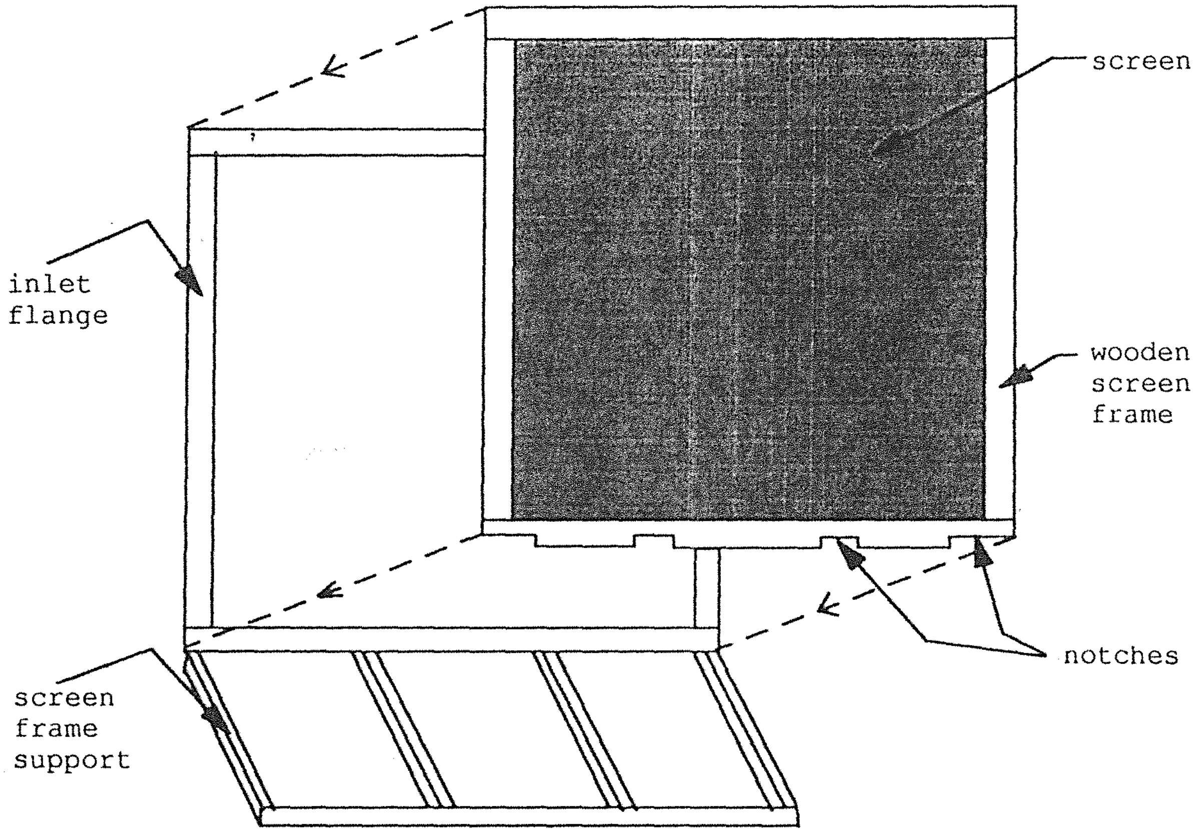


Figure 13. Screen Frame and Inlet Flange

quality compared with the other screens. The remaining screens were much less expensive and were made of a fine mesh fiberglass. The screens need to be stretched onto the frames as tight as possible. This task required three to four people to achieve the best results. First, the screen was centered over the frame. In one corner, the screen was stapled to the frame with a staple gun. The screen was stretched along that side of the frame and stapled in the corner at the other end of the stretched side. That entire side was then stapled at six inch intervals. Beginning at the center of the opposite side of the frame, the screen was then stretched and stapled. Alternating from one side of center to the other, the screen was stretched and stapled all the way to the corners. This process was repeated to stretch the screen in the other direction. When the first stainless steel screen was completed, it was attached directly to the fiberglass flange of the contraction cone. The frame was positioned correctly and pilot holes were drilled through the wood and fiberglass. Long wood screws fastened the screen frame to the flange. The remaining screens would be fastened one by one to the previously attached screen. Two stainless steel screens and seven fiberglass screens or a total of nine screens were used. Two of the seven fiberglass screens would sandwich the honeycomb section to keep it in place.

Honeycomb

The honeycomb was purchased in three 4 ft. x 12 ft. x 4 in. thick sections. It was constructed of material that is very similar to thousands of drinking straws glued together. The three 4 ft. sections would be stacked to make a 12 ft. x 12 ft. wall of honeycomb. A 12 ft. long 2 x 6 placed flat and sandwiched between two regular screen frames was used as the honeycomb frame. The 2 x 6 served as the floor for the honeycomb to sit upon, and a screen stretched onto each of the frames would contain the honeycomb laterally. The honeycomb frame is shown in Figure 14. A special implement had to be built in order to carry the honeycomb sections without damaging the cells. The implement consisted of a 2 x 6 for the floor with three 2 x 4's attached as uprights. The first section of honeycomb was placed on the implement, carried to the inlet, and slid into place. The forklift was used to lift the next section of honeycomb to the height needed to slide it into place on top of the first. The last section was loaded in the same manner. When the honeycomb was in position, the last screen frame was put in place. The bottom of the frame was attached directly to the 2 x 6 floor of the honeycomb section. Since the screen frames sandwiching the honeycomb were separated by the width of the honeycomb, metal brackets extended from the edge of one frame to the other to fasten the frames together and secure the honeycomb between them.

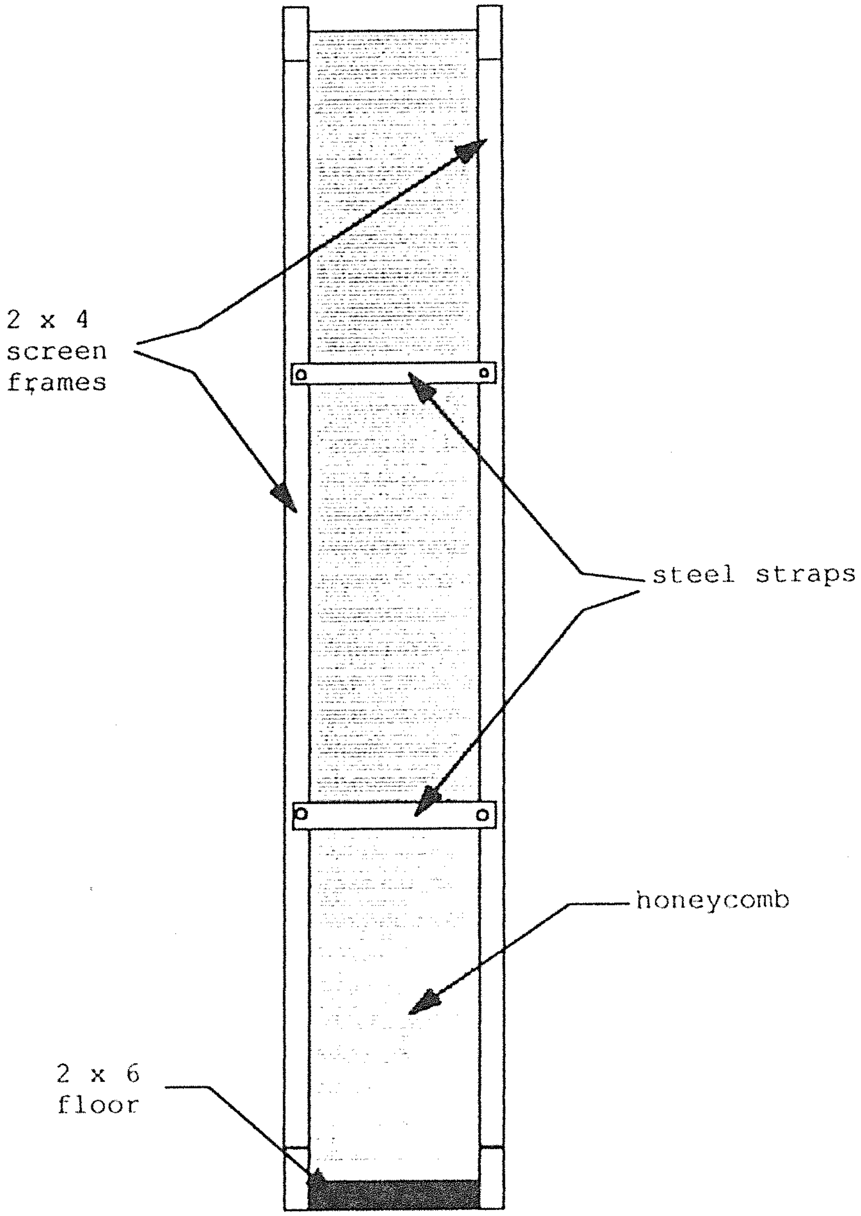


Figure 14. Side View of Honeycomb Section

CHAPTER 3

WIND TUNNEL USE/MAINTENANCE

Maintenance

The inside walls of the contraction cone, test section, and diffuser will collect dust and should be cleaned periodically. The electronic components and the computer should also be protected from dust while not in use.

Blower

The maintenance schedule for the blower should be determined by the bearing lubrication interval. This interval varies with the shaft diameter of the blower and the speed that the blower will be running. A lubrication interval table can be found in the blower booklet which should be available at the facility. It was determined that the maintenance interval should be four months. Maintenance should include the following tasks:

1. Check the fan wheel for wear, corrosion, and for build up of material. All can cause the wheel to become unbalanced resulting in vibration and bearing wear. Clean or replace the wheel as required.
2. Check the belts for wear. Replace them as a set if required.
3. Lubricate the bearings. Bearings should be lubricated with a good quality lithium-based grease conforming to NLGI Grade 2 consistency. Do not use “high temperature” greases, as these are not usually formulated for the high operating

speeds of fan bearings. Add grease just until a slight purging at the seals occurs.

4. Check setscrews and bolts for tightness.

If problems occur with the blower or further information is needed about the blower, consult the detailed product booklet published by the blower company. A copy of the booklet should be available at the wind tunnel facility.

Screens and Honeycomb

The screens and honeycomb will collect dust over a period of time. The use of flow visualization substances will only accelerate the dust collection process. For this reason these devices should be removed and cleaned periodically. Compressed air can be used to blow the unwanted particles from the screens and out of the honeycomb cells.

Removing the Test Section

If modifications need to be made to the test section or if a different type test section is needed, the existing test section can be removed. The ability of the contraction cone to roll back and forth, and the casters on the test section make the removal procedure very easy to perform. First, the bolts attaching the test section to the diffuser and to the contraction cone are removed. The four adjustable legs are raised until the test section is resting on the casters. The contraction cone can now be rolled back and the test section removed for modifications or replacement.

Preparation for Use

Prior to each time the tunnel is to be used, the following routine should be performed to ensure that the facility will operate safely and provide the best possible flow quality.

1. Open the test section door and make sure there are no loose objects in the test section, diffuser, or contraction cone.
2. Make sure the model and/or probe(s) are secured properly.
3. Close and latch the test section door.
4. Make sure the outside screen of the contraction cone inlet has no foreign materials stuck to it. Also remove any other objects from in front of the inlet that may alter the flow into the inlet.
5. Close hallway and room doors in the region that the inlet draws air from. These door positions should be maintained throughout a particular experiment to ensure a consistent inlet flow.
6. Check the wind conditions outside. Windy or gusty conditions create very erratic back pressure and notably deteriorate the flow quality. Calm conditions are required for the best results.

Using the Tunnel

In order to use the wind tunnel facility, the user must be familiar with the blower controls (wall unit and remote operation), and understand the traverser hardware and software. Particular test sections may have limits on the flow speed for safety reasons. The travel of a measurement probe inside a test section may also be restricted by various factors. Find the information on the velocity limits of any test section before using it, and note any restrictions on probe travel in the test section. For this test section, the maximum tested velocity is 120 ft/s. The maximum velocities of any test section being used should be known before using it.

Blower Controls

Once the necessary preparations have been made the blower can be started. On the south wall of the lab there are two electrical boxes. The large "local" control box on the right says GPD 503 on the front, and the smaller breaker box is on the left. The breaker box has a large external breaker that must be tripped in order to supply power to the new wind tunnel blower as well as the old wind tunnel. The breaker is padlocked so that only authorized personnel may use the wind tunnel. When an experiment has been completed, the breaker should be turned off and locked. There is also a switch on the side of the breaker box that has specific positions for the new and old wind tunnels, as well as an off position. Make sure that the switch is in the appropriate position.

The blower is equipped with a variable speed drive that allows continuous blower speed adjustment all the way up to its maximum. The GPD 503 provides the means of locally starting the blower and controlling its speed. A remote control unit was also installed to allow the user to control the blower speed from the test section. A toggle switch on the GPD 503 panel allows the user to choose either local or remote operation. The remaining discussion in this section should provide the user with the knowledge needed to start and control the blower either locally or using the remote unit.

The Digital Operator consists of a keypad and display and can be found on the front panel of the GPD 503. The Digital Operator is shown in Figure 15. The GPD 503 can be operated in Drive (DRIVE) mode or the Program (PRGM) mode. The Program mode allows the user to configure the GPD 503 to the application. The Drive mode is used to control motor operation. The instructions that follow will cover the necessary steps to start the blower, control its speed, and stop its operation. More detailed descriptions of each indicator lamp and their functions as well as the keypad keys and their functions can be found in the GPD Technical Manual. Also if the GPD 503 needs to be reprogrammed or configured differently the user should consult the GPD 503 Technical Manual which should be readily available.

With the breaker flipped on and the switch turned to the "New Wind Tunnel" position, the following sequence can be used to control the blower from the Digital Operator.

Local Operation

1. Make sure the REMOTE/LOCAL toggle switch on the GPD 503 is set to LOCAL.
2. When power is applied, the **DRIVE** lamp should be lit and the display should read "**F00.00**". This is a preset frequency reference value.
3. The run frequency may be set to any number between 0 and 60 Hz by using the >, ∧, and ∨ keys to input the desired frequency. The ∧ and ∨ keys are used to increase and decrease the blinking digit. The > key is used to shift the blinking position of the display so that all the digits can be adjusted. **Always make yourself aware of upper limits for the test section being used before starting the motor.**

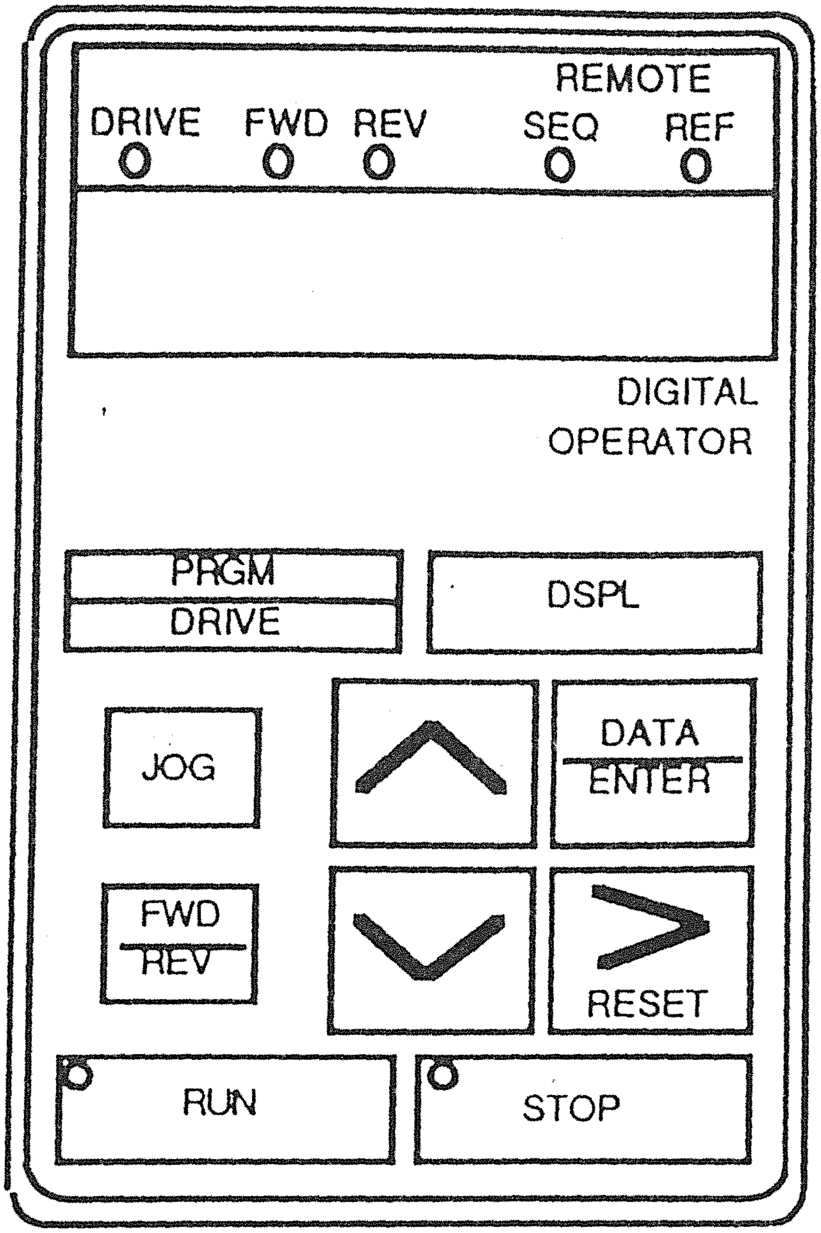


Figure 15. Digital Operator Display and Keypad

The frequency should be set to a low number when the motor is started. The frequency can then be increased to achieve the desired speed by monitoring a manometer or other measuring device.

4. When the desired run frequency is displayed, press the **DATA/ENTER** key to write the value into memory. The displayed value will stop blinking for about 2 seconds, then continue blinking.
5. Use the **FWD/REV** key to select the direction of motor rotation. The **FWD** and **REV** indicator lamps show which direction the motor will rotate when started.
6. Press the **DSPL** key to display the present output frequency of the motor.
7. Pressing the **RUN** key starts the motor. The output is increased to the desired run frequency at a programmed acceleration rate.
8. To stop the motor, press the **STOP** key. The motor speed will decrease, at the programmed deceleration rate, to zero and remain stopped.
9. Turn the dial on the breaker box to off, flip the breaker to off and lock it.

Remote Operation

The usual method of controlling the blower speed will be from the test section using the remote control unit. This unit has an ON/OFF toggle switch, a ten-turn potentiometer, and a dial indicator. The potentiometer controls the voltage output and the dial indicator displays that voltage. The following steps should be followed to control the blower motor speed and thus the freestream velocity.

1. Apply power to the GPD 503.

2. Display the present frequency output on the Digital Operator display. This is only done to allow the operator to check the present frequency output if necessary.
3. Flip the REMOTE/LOCAL toggle switch to REMOTE.
4. Blower speed can now be controlled from the workstation in front of the test section. The remote control unit can be found hanging from a hook on the wall of the test section.
5. Make sure the potentiometer is turned all the way to the left (or at the correct preset position) to ensure any limits are not exceeded.
6. Flip the ON/OFF toggle switch to ON.
7. A specific nominal freestream velocity is usually desired for an experiment. Turn the potentiometer to the right while monitoring a manometer or other measuring device until the desired freestream velocity is achieved. The voltage indicator should also be watched while adjusting the potentiometer. **Remember to be aware of upper limits on the test section being used**.
8. To stop the blower motor simply turn the potentiometer all the way to the left and flip the ON/OFF toggle switch to the OFF position. The user may wish to stop the motor but later perform an experiment at the exact same velocity. In this case, lock the potentiometer in the desired position and just flip the toggle switch off. The motor will decelerate to a stop and remain stopped until the toggle switch is flipped on. When the toggle switch is turned on the motor will accelerate back to the desired speed.
9. Turn the dial on the breaker box to off, flip the breaker to off and lock it.

Traverser/Data Acquisition

The software was compiled by combining code that would control the traverser and an existing data acquisition code. The data acquisition code was initially written to allow the user to initiate the extraction of each data point, and decide to either send the data point to a file or omit it, based on its apparent validity. When taking a relatively small number of data points, this was an efficient way to ensure that each data point was valid. However, when taking a large number of data points, such as a matrix spanning the three foot square test section, this method would be very time intensive. The existing code was modified and incorporated with the traverser code so that the process was automated, but also allowed the user to monitor the data, since each point would be printed to the computer screen. The program prints a message after each data point, instructing the user on how to terminate the run if necessary. The speed of the traverser as well as the delay time after each command can be changed. These values are set in lines 2943-2944 of the program which can be found in the Appendix. Detailed descriptions of what each character and number in these lines represents can be found in the ITC-232 User Guide. Stepper motor movement is initiated by a command that specifies the stepper motor with a letter designation (A,B,C), the direction to turn (left-L, right-R) and the number of steps to be taken. Examples of these commands can be found in lines 3770-3780 of the code. In these commands, B and C are stepper motor designations and R and L are the directions that the stepper motors are to turn. The stepper motors being used have 200 steps/revolution and their pitch is 1 inch/revolution. So for every 200 steps the stepper is commanded to turn, the traverser travels 1 inch.

Below is a description of which stepper motor each letter designation refers to, and the traverser probe directions corresponding to R and L.

B --- Horizontal stepper	C --- Vertical stepper
L --- Movement toward motor	L --- Downward movement
R --- Movement away from motor	R --- Upward movement

The following instructions provide the user with the sequence necessary to complete the data acquisition process.

1. Turn on the power supply.
2. Turn on the control board.
3. Make sure the traverser is in its home position. The horizontal home position is marked on the left slider rod. The translating block should just touch this mark. The vertical home position is marked on the long upright plate extending down from the translating block. The plate attached to the lead screw nut should be even with the mark. Position should be checked after each run as there are small inconsistencies in the stepper motor rotation.
4. Turn the computer on and change to the CATES directory. The data acquisition will be run from this directory.
5. Type <gwbasic>. This is the version of basic that the data acquisition card was written in. The program will only run in this version. It will not run in QBASIC. If the code needs to be modified for any reason, it can be edited in QBASIC or the editor much easier. Keep in mind however, there are a few commands that are interpreted differently by the two versions.

6. To load the program to be run, use the menu in GWBASIC to choose the "Load" key and type in the program name (PTRAV.BAS), or just type <load "ptrav.bas">.
7. To run the program use the menu again to choose the "Run" key. This should initiate the program.
8. The program will first ask the user to type an appropriate calibration file name heading.
9. The user will then be prompted to input the atmospheric pressure and the temperature. The program uses these values to calculate the air properties and prints them to the screen.
10. Next the program will ask the user if a calibration is to be performed, or if the existing calibration file is to be read. The transducer should be calibrated each time the system is started up. Often, concurrent data sets will be taken without shutting the system down. In most cases it is acceptable to calibrate the transducer before the first data set and read that file for the following sets. If the user is unsure, the calibration should be performed before each set of data to be safe.
11. Before the calibration begins, or after a calibration file is read in, the program will ask the user to input the upper and lower scan limit numbers. The user should enter "0" for both of these questions.
12. Next, the user is prompted to enter the sampling rate and the number of conversions. The sampling rate is the number of readings taken each second and the number of conversions is the total number of readings to be taken. The number of conversions divided by the sampling rate determines the record length. For example, a sampling

rate of 100 samples/second with 1000 conversions yields a record length of 10 seconds. All the readings taken in the ten second span are then averaged to give one data point. The longer the record length the more accurate the data point will be, however, increasing the record length for a large matrix of data points can be very time consuming. The feasibility of a specific record length should be considered, and may be different, for each experiment performed. Remember, this is not only the record length for the calibration, but also for the matrix of data to be taken.

13. Calibration should include at least 10-15 readings that extend over the range of velocities to be used in the actual experiment. The pressure readings should be taken from an inclined manometer and read as accurately as possible. Be sure to allow the pressure to stabilize before entering each reading. When the calibration is ended by the user, the program uses the least squares fit method to calculate the calibration equation.

14. Once the calibration is finished, the user will be asked to type an appropriate pressure measurement file heading.

15. The program will now prompt the user to reset the ITC232-A. The ITC232-A refers to the control board. The reset mechanism is a momentary toggle switch mounted on the panel of the control board box. Before resetting, make sure the desired freestream velocity has been set correctly and everything else is ready for data collection. Now press the toggle switch down and release it. The data acquisition process will begin immediately.

16. When the data matrix is completed, the user will be prompted to enter 1 to take data near the wall, or 0 to end data collection. If 1 is entered, a message will appear

instructing the user to move the set screw position. This refers to the set screw at the bottom of the pitot-static probe. Change its position and proceed as directed. If 0 is entered, the run is complete.

17. After the run has been completed, the data files are written to the following generic filenames: Air properties written to ------------- c:\bprog\datafile\airprops.dat
Transducer calibration written to -- c:\bprog\datafile\ptcalout.dat
Measurement data written to ------- c:\bprog\datafile\pitot.dat

    After each run these files must be copied to an appropriately named filename other than the filename used by the program. If this is not done, the next data set will be written over the previous set.

Preliminary Data

To ensure the proper operation of the data acquisition code and traverser hardware an initial set of data was taken covering the cross section of the test section. It was also decided that further data would serve several purposes. Data taken with different numbers of flow management devices would provide us with an evaluation of the effectiveness of these devices. Furthermore, the data taken with all flow management devices in place would provide faculty and students with an idea of what type of data to expect.

In order to evaluate the effectiveness of the screens and honeycomb, data was taken with no flow conditioning to compare with data collected with flow management devices. Also, since the influence that the flow management devices have on the flow varies with velocity, a range of velocities are necessary to understand what is happening.

To accomplish this it was decided that data would be collected at three separate nominal velocities (30 ft/s, 75 ft/s, 120 ft/s) for three different flow management configurations. The first configuration was with no flow conditioning as spoke of earlier. The second configuration consisted of two stainless steel screens, and the final configuration comprised all the flow management devices which included two stainless screens, seven fiberglass screens and the honeycomb. Each set of data taken would be reduced to two contour plots exhibiting the axial velocity contours and percent difference from the average velocity.

The contour plots that resulted from the data at the 30 ft/s nominal velocity are shown in Figures 16-21. These plots show that at this low velocity the percent difference from the average is cut from 2% to 1%. The 75 ft/s nominal velocity plots are found in Figures 22-27. Again the percent difference is cut to 1% with all flow management devices in place, but we begin to see a pattern of slower velocities toward the bottom of the test section. The plots from the final nominal velocity of 120 ft/s are shown in Figures 28-33. At this velocity the percent difference is reduced from 3.5% with no flow conditioning to 1.2% with all screens and honeycomb in place. Again however, if not for a small area at the bottom of the data matrix, the difference would be much smaller. It was suspected that these slower velocities were the result of a lab wall that is within 20 feet of the inlet. A smoke generator placed at various positions close to the lab wall proved our initial prognosis correct and it was concluded that the wall was causing a disturbance in the inlet flow too great for the flow management devices to equalize at the higher velocities.

In addition to the contour plots the data taken with all screens in place was also used to create a graph showing a centerline (x = 15.6") vertical velocity sweep at each nominal velocity, and a separate graph dipicting the percent error from the mean velocities. These graphs are shown in Figures 34-35.

Concluding Remarks

The results of the preliminary data demonstrated that the flow management devices were effective in increasing the velocity uniformity across the test section. Despite the adverse effect from the lab wall, the final velocity uniformities were satisfactory, recalling that for the 30 and 75 ft/s nominal velocities the percent difference was 1% and for the 120 ft/s nominal velocity, 1.2%. However, we would have liked the percent differences to be a bit smaller, especially at the higher velocity.

Construction is now in progress on the new building on campus. A portion of the new building is dedicated solely for the purpose of housing this wind tunnel facility. When the facility is moved to its new location the lab wall's influence on the inlet flow will no longer be present which should increase the flow uniformity as well as other air flow qualities.

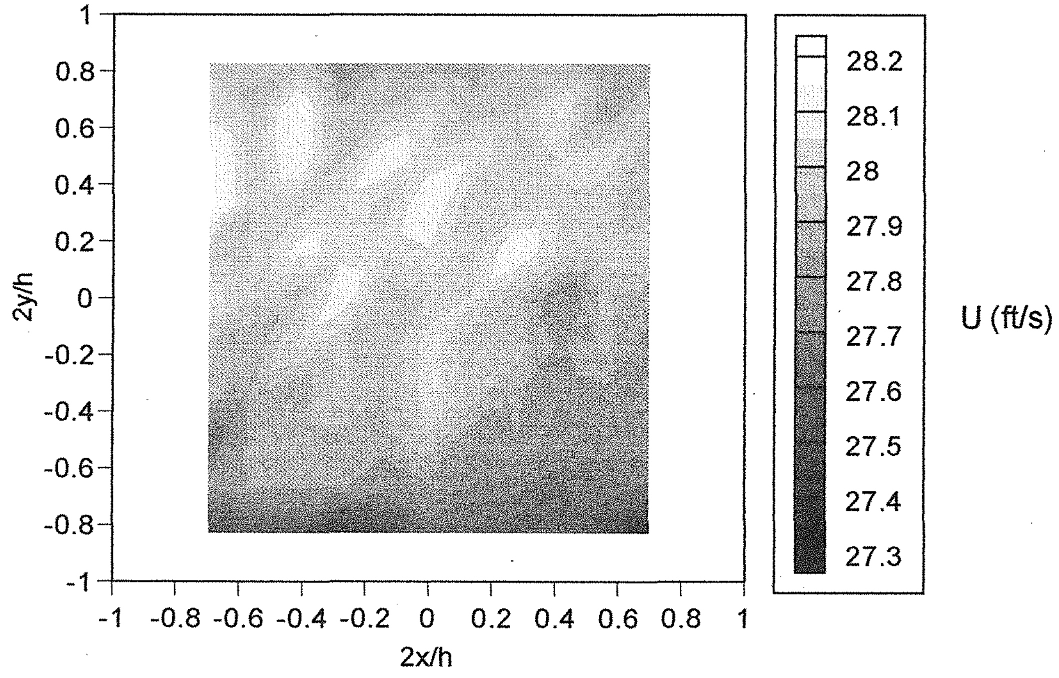


Figure 16. Axial Velocity Contour with No Flow Conditioning (Uave = 27.89 ft/s)

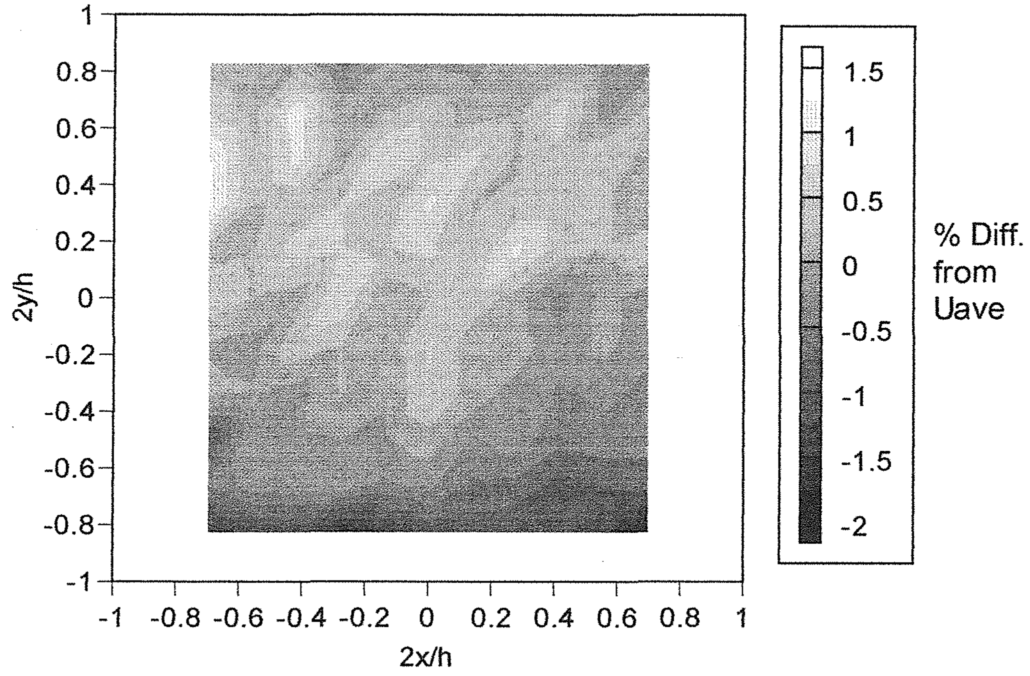


Figure 17. %Difference from Uave with No Flow Conditioning (Uave = 27.89 ft/s)

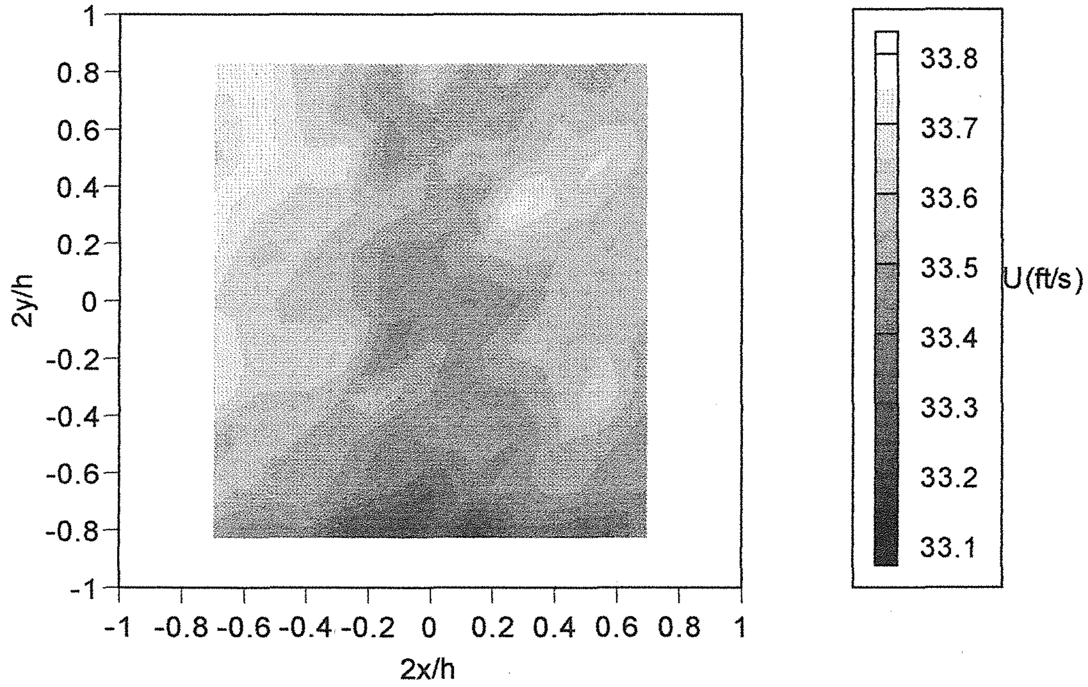


Figure 18. Axial Velocity Contour with 2 Screens (Uave = 33.54 ft/s)

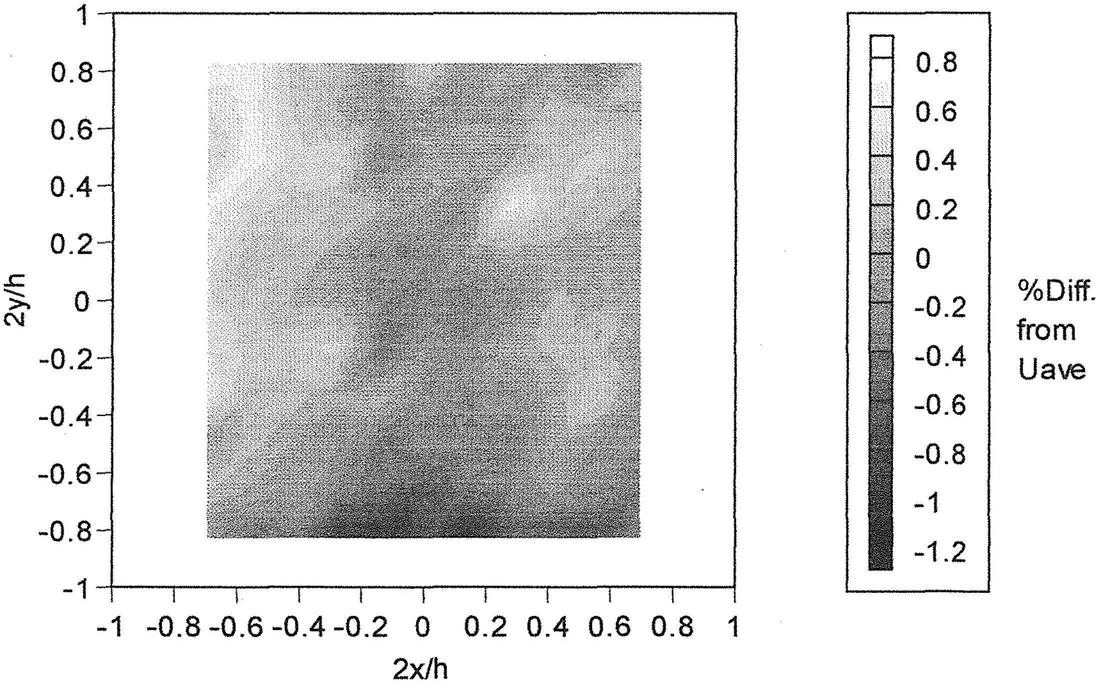


Figure 19. %Difference from Uave with 2 Screens (Uave = 33.54 ft/s)

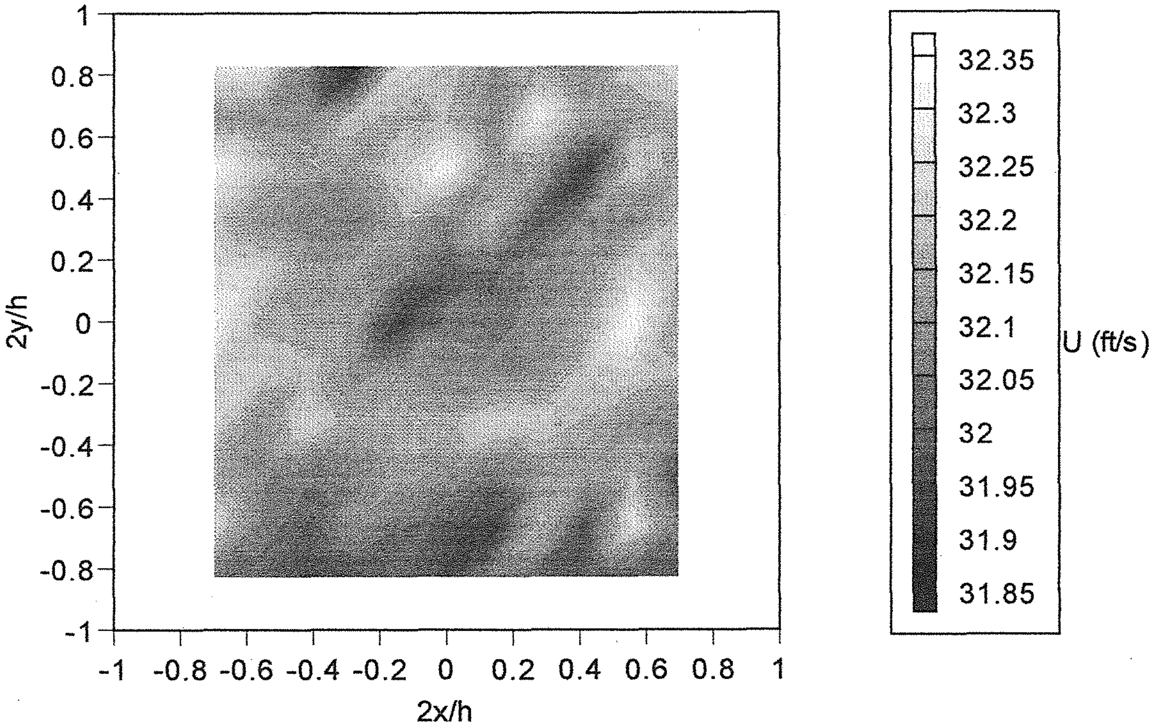


Figure 20. Axial Velocity Contour with All Screens (Uave = 32.12 ft/s)

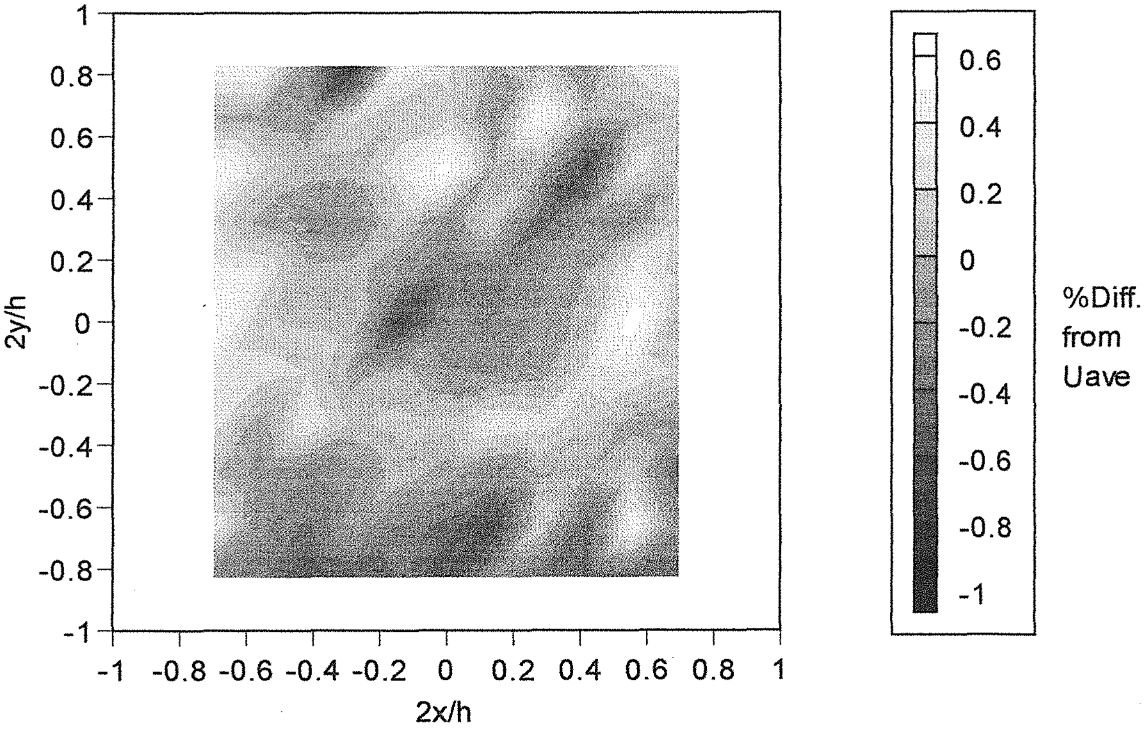


Figure 21. %Difference from Uave with All Screens (Uave = 32.12 ft/s)

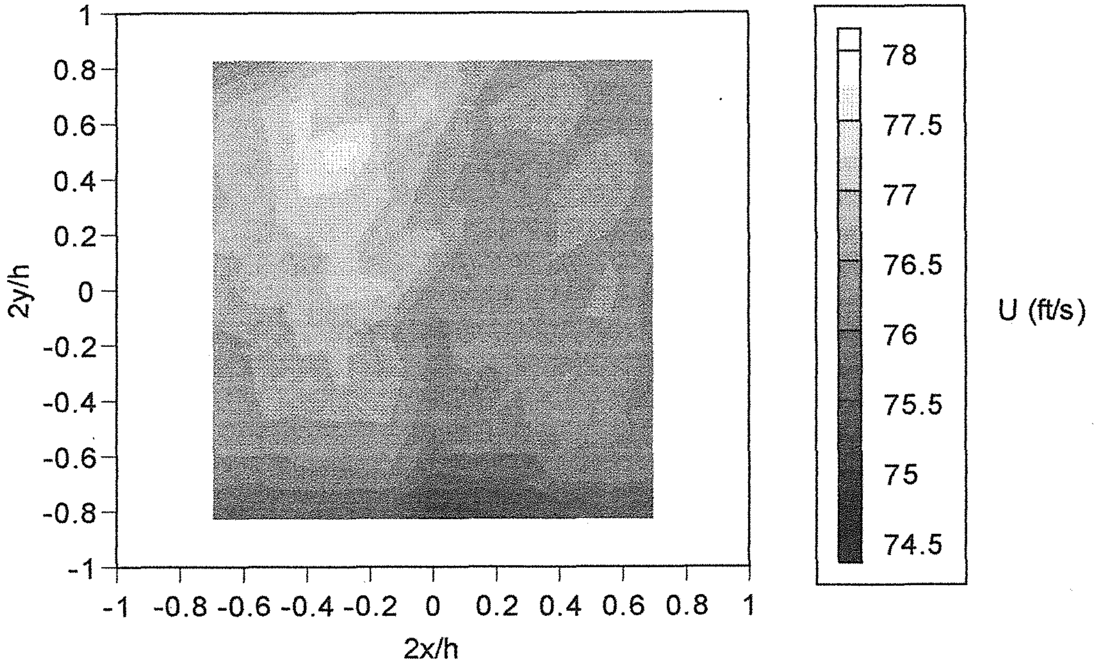


Figure 22. Axial Velocity Contour with No Flow Conditioning (Uave =76.44 ft/s)

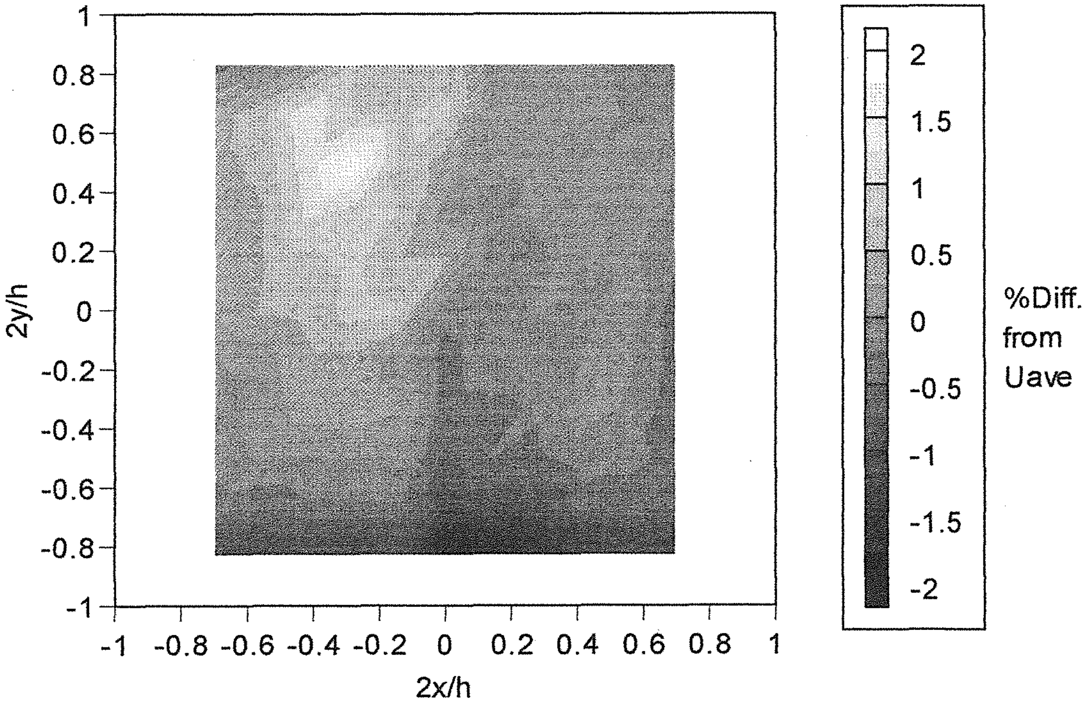


Figure 23. %Difference from Uave with No Flow Conditioning (Uave = 76.44 ft/s)

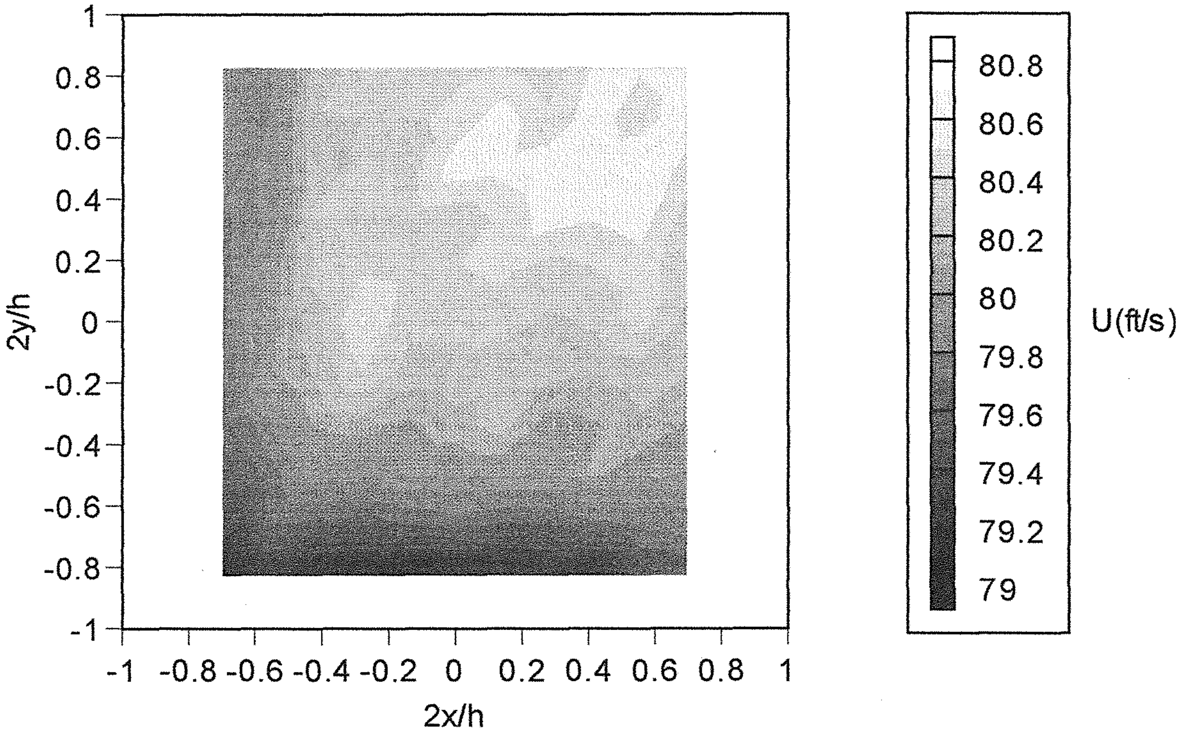


Figure 24. Axial Velocity Contour with 2 Screens (Uave = 80.09 ft/s)

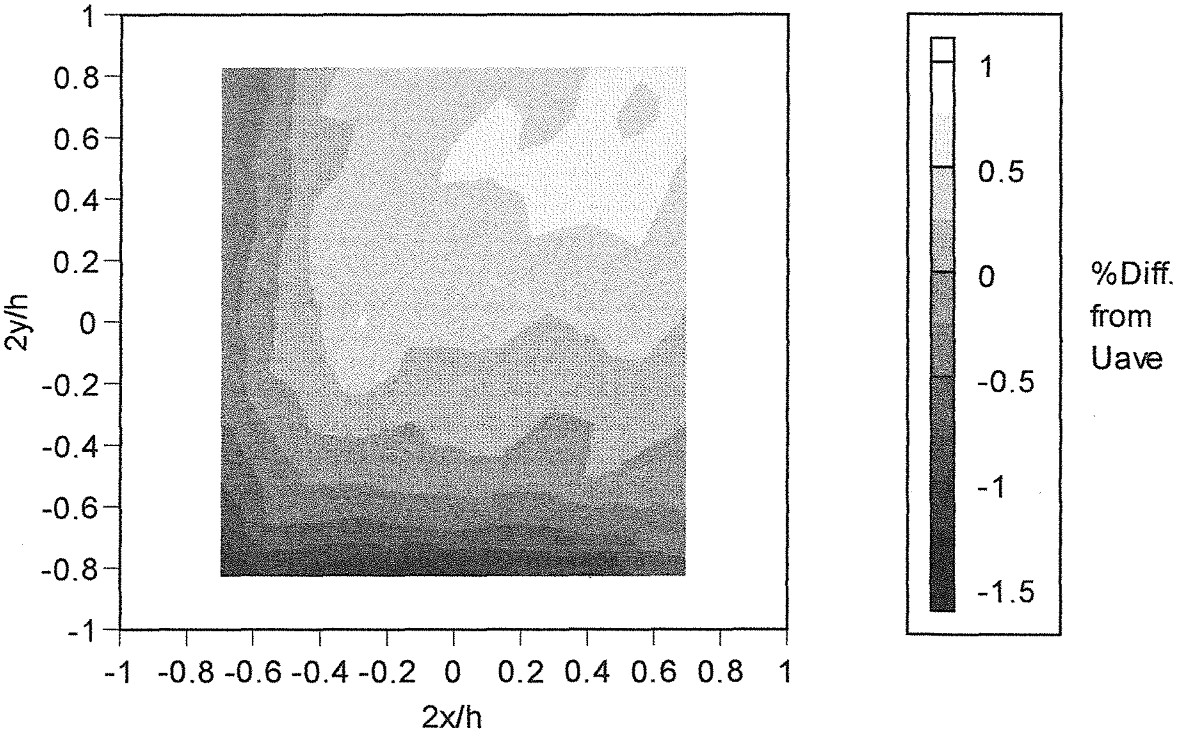


Figure 25. %Difference from Uave with 2 Screens (Uave = 80.09 ft/s)

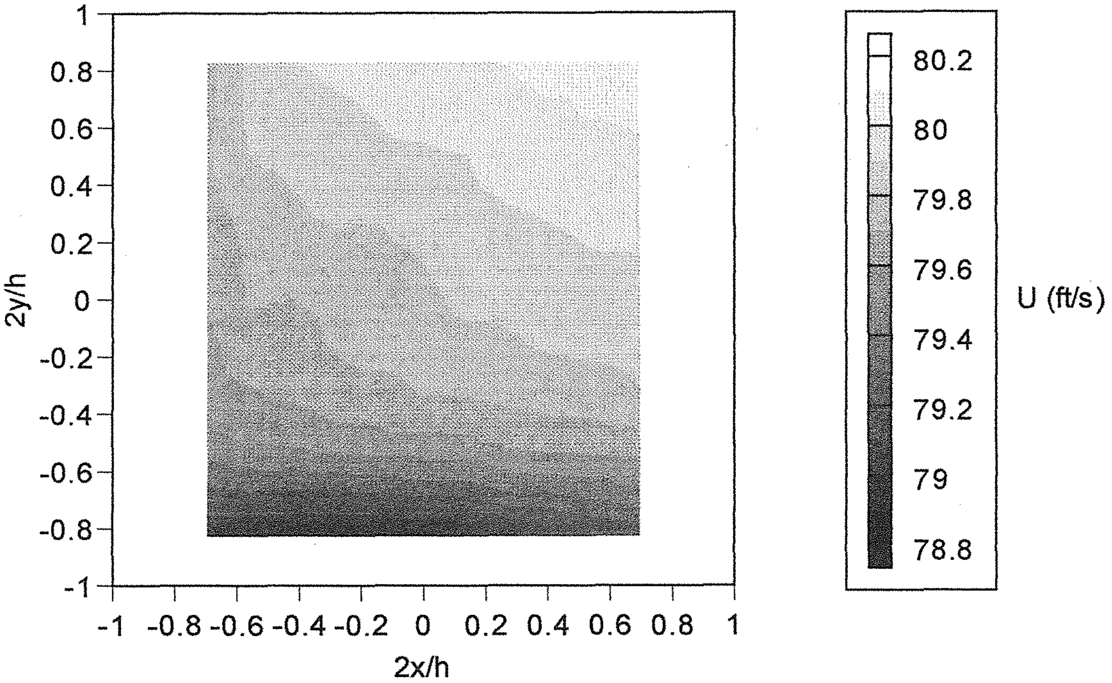


Figure 26. Axial Velocity Contour with All Screens (Uave = 79.68 ft/s)

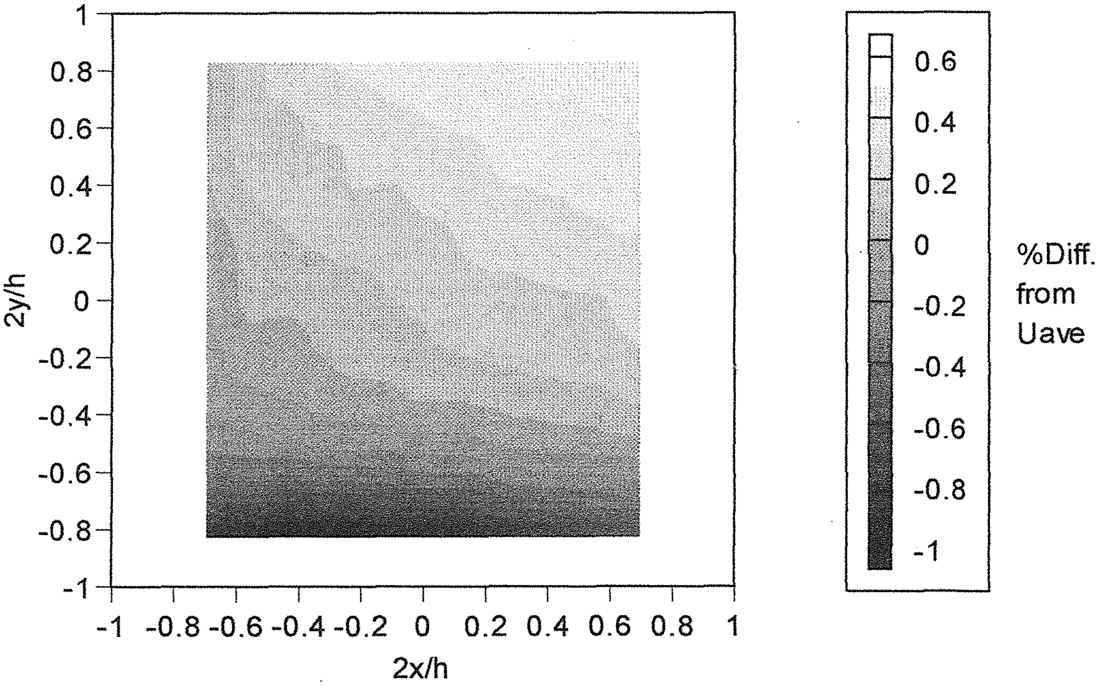


Figure 27. %Difference from Uave with All Screens (Uave = 79.68 ft/s)

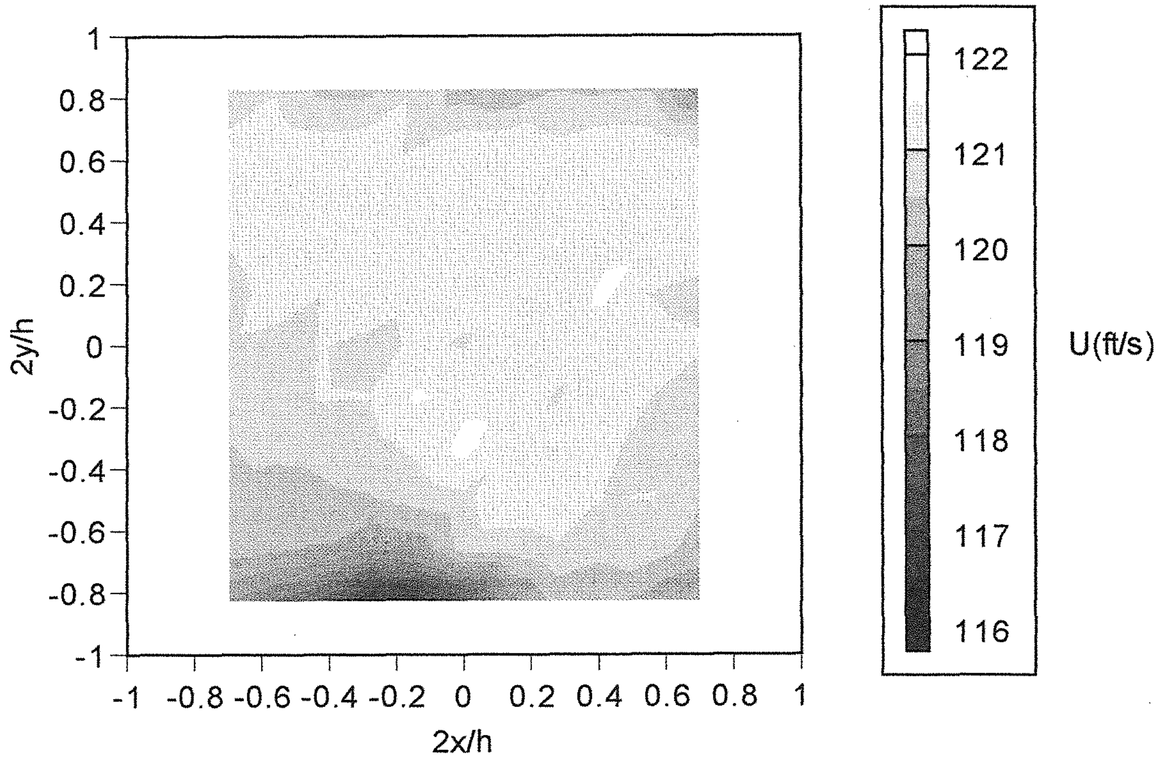


Figure 28. Axial Velocity Contour with No Flow Conditioning (Uave = 120.73 ft/s)

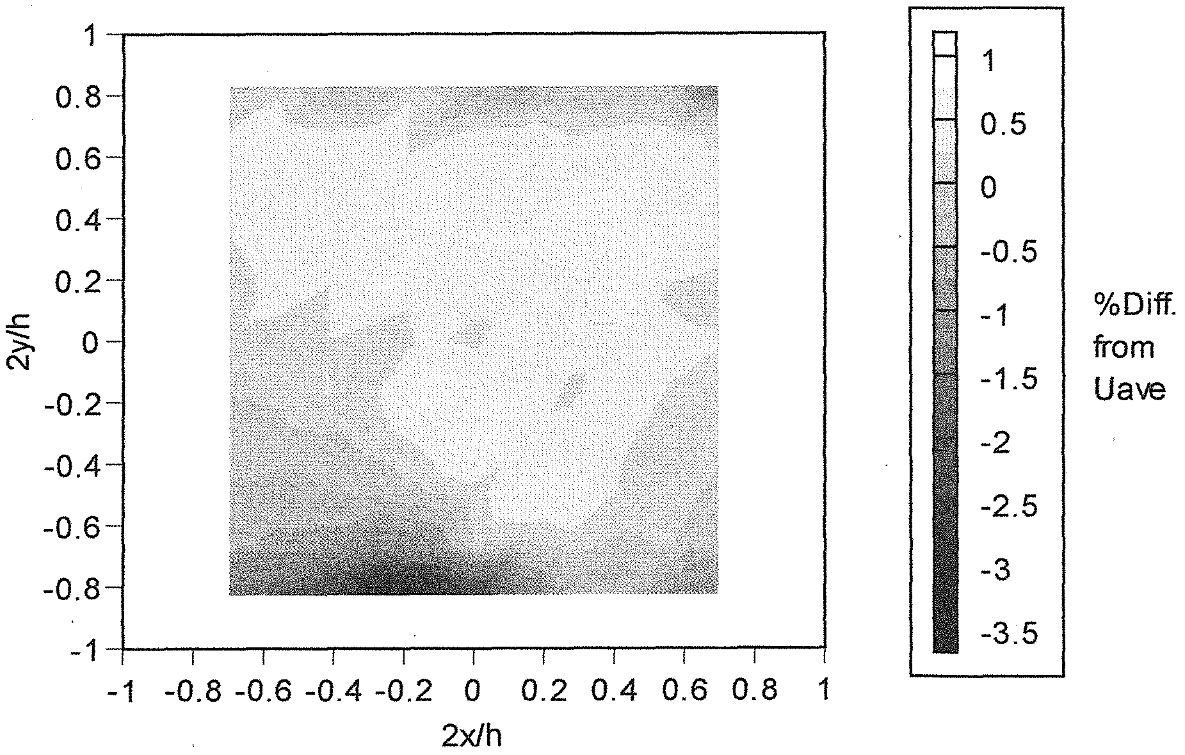


Figure 29. %Difference from Uave with No Flow Conditioning (Uave = 120.73 ft/s)

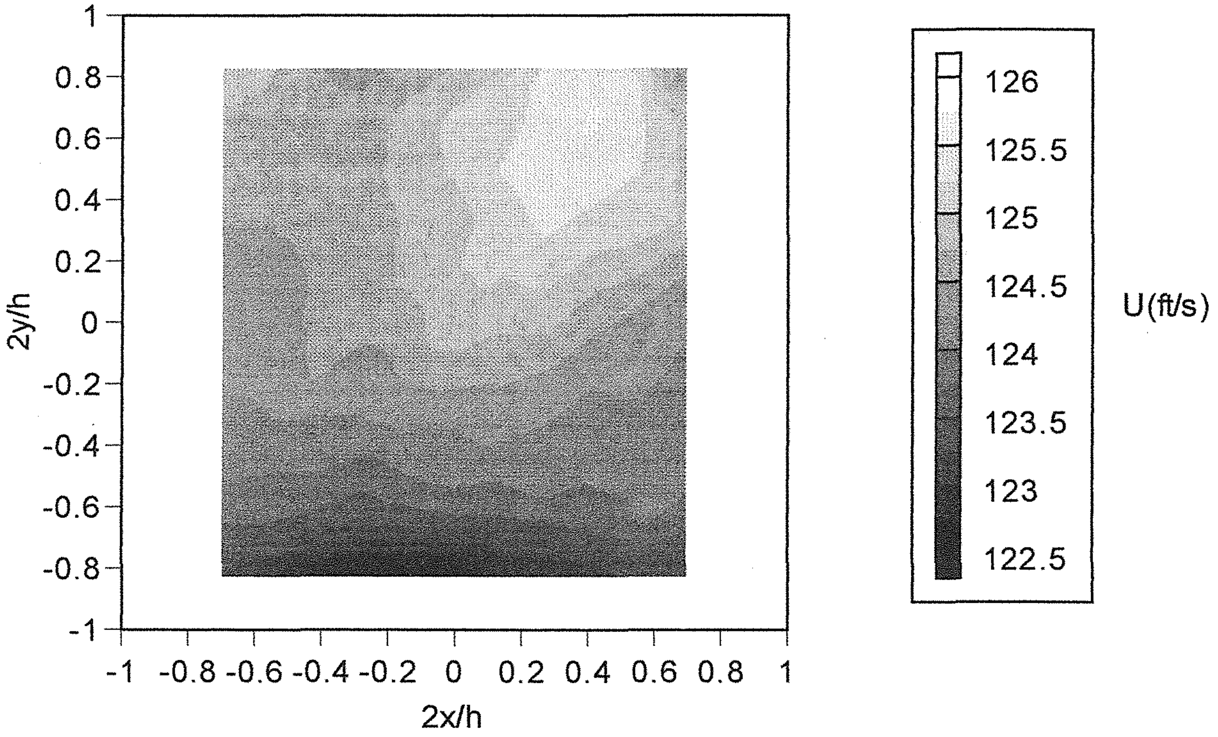


Figure 30. Axial Velocity Contour with 2 Screens (Uave = 124.39 ft/s)

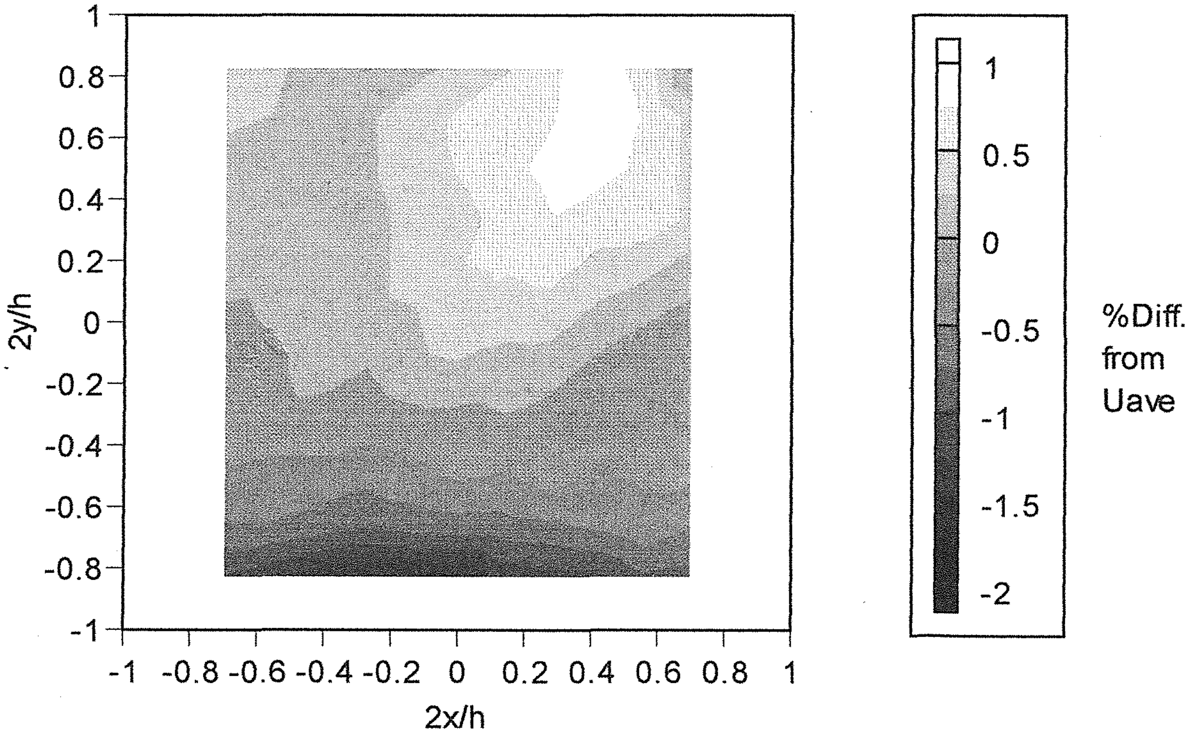


Figure 31. %Difference from Uave with 2 Screens (Uave = 124.39 ft/s)

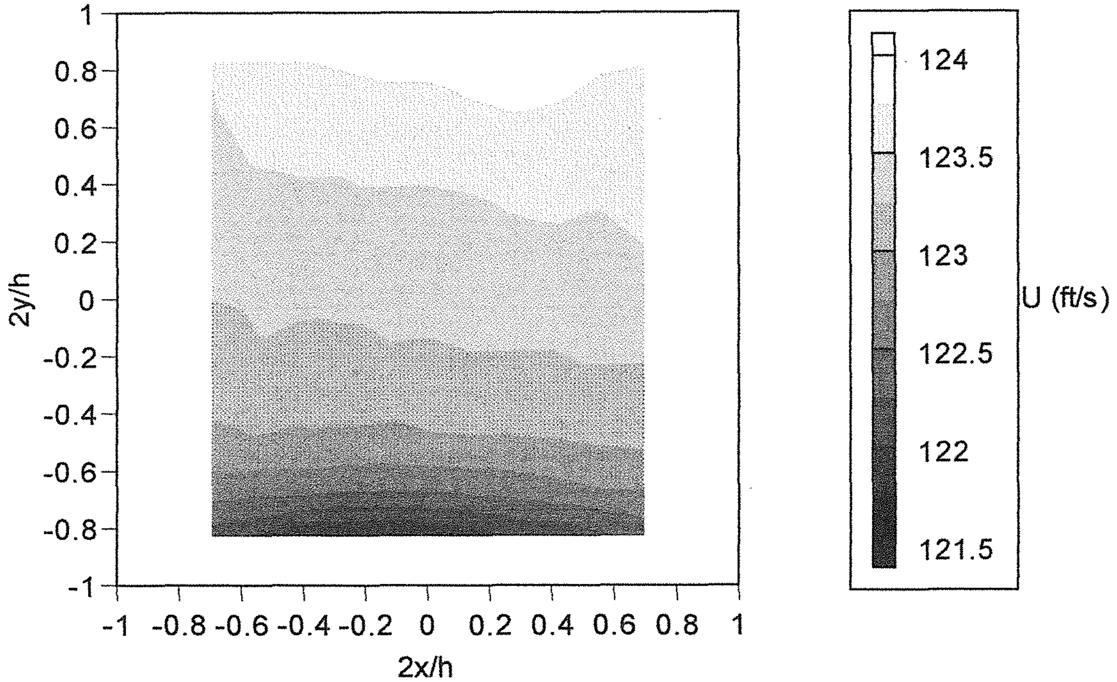


Figure 32. Axial Velocity Contour with All Screens (Uave = 123.19 ft/s)

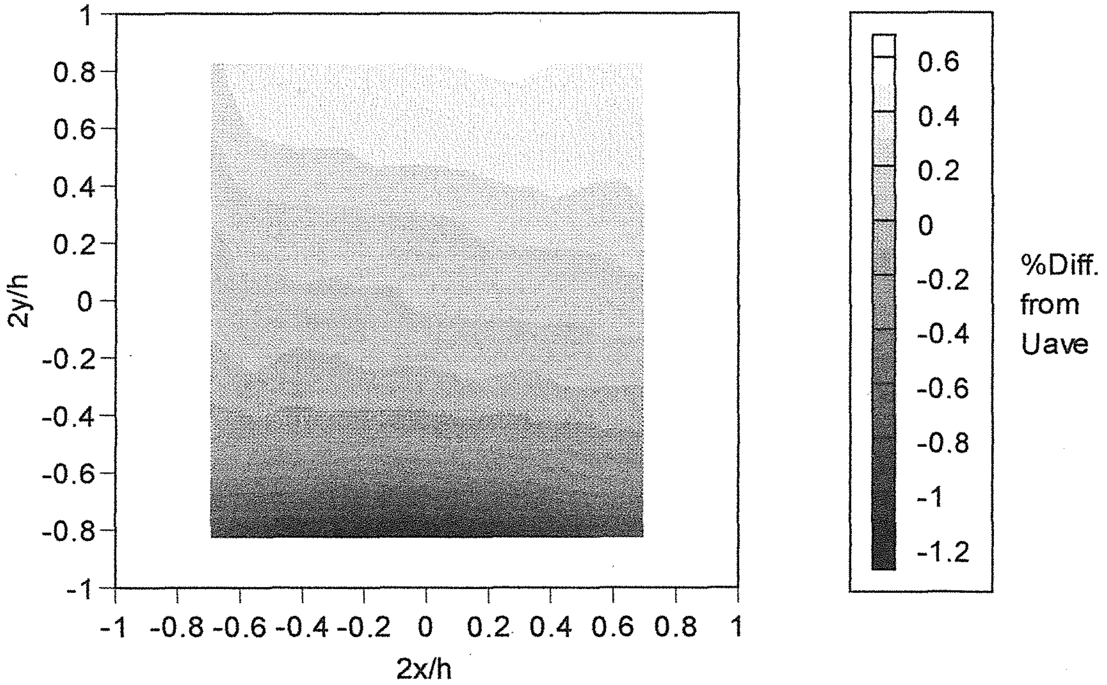


Figure 33. %Difference from Uave with All Screens (Uave = 123.19 ft/s)

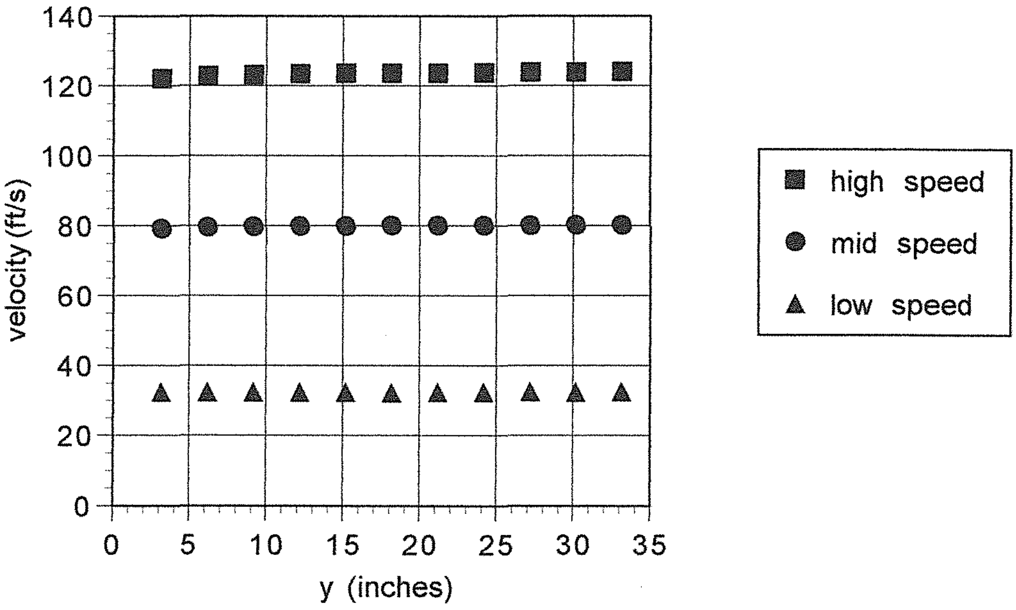


Figure 34. Centerline Vertical Velocity Sweep with All Screens

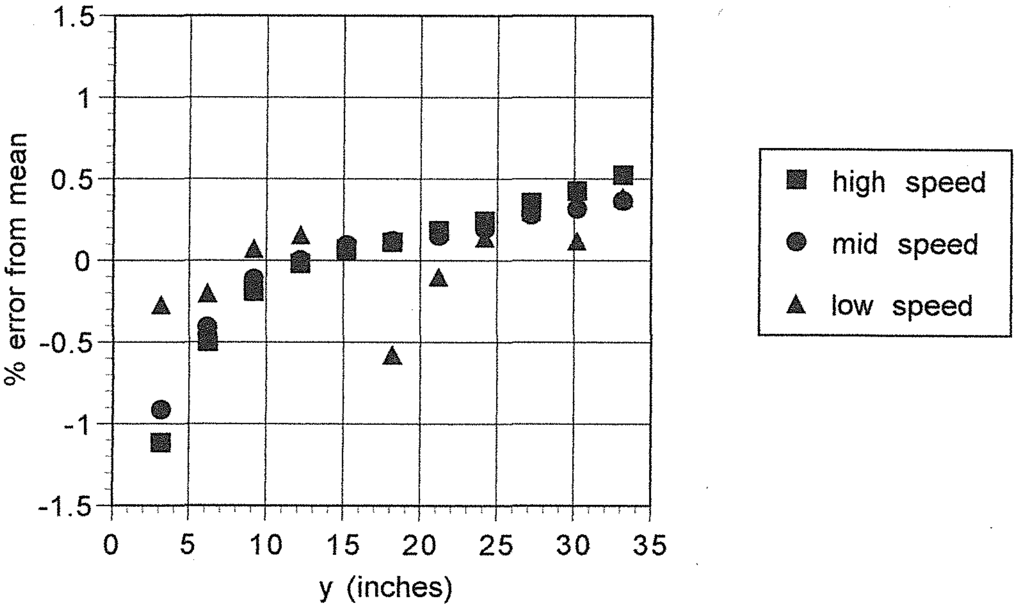


Figure 35. %Error from Mean During Centerline Vertical Velocity Sweep (All Screens)

BIBLIOGRAPHY

Batill, Stephen M. and Nelson, Robert C., "Low Speed, Indraft Wind Tunnels." Department of Aerospace and Mechanical Engineering, University of Notre Dame, 1989.

Batill, S. M., Mueller, T. J., Scharpf, D. F., Strebinger, R. B., Subramanian, S. S. and Sullivan, C. J., "The Design of a Subsonic Low-Noise, Low-Turbulence Wind Tunnel for Acoustic Measurements.", AIAA Paper No. 92-3883., July 1992.

E. Froebel and Michel, U., "Lower Limit for the Velocity Fluctuation Level in Wind Tunnels." Experiments in Fluids, 6, 49-54, 1988.

ITC Microcomponents Inc., ITC232 User Guide. North Vancouver, B.C., 1995.

Loehrke, R. I. and Nagib, H. M., "Control of Free Stream Turbulence by Means of Honeycombs: A Balance Between Suppression and Generation." ASME publication, December, 1975.

MagneTek, Inc., GPD 503 Technical Manual., New Berlin, Wisconsin, July, 1994.

Mousseux, M., Saric, W. S., Takagi, S., "The ASU Unsteady Wind Tunnel and Fundamental Requirements for Freestream Turbulence Measurements.", AIAA Paper No. 88-0053., January, 1988.

New York Blower Company, Centrifugal Fans., Willowbrook, Illinois, May, 1994.

Saric, W. S., "The ASU Transition Research Facility.", AIAA Paper No. 92-3910., July, 1994.

APPENDICES

Computer Code

APPENDIX A -- TEST CODE

```
'REM This program shows the basics for operating the ITC232-A from
'  QBASIC. To issue a command, place it in W$ and call subroutine
'  WRITESERIAL. If a value is then read, you can retrieve it from
'  V in numerical format or in V$ as a string.

CLS
True = 1
False = 0

REM Open COM port USE COM1 or COM2
OPEN "Com1: 9600,N,8,1,CD0,CS0,DS0,OP0,RS,TB2048,RB2048" FOR RANDOM AS #1

START:
PRINT "PLEASE RESET THE ITC232-A"
PRINT
GOSUB READSERIAL
PRINT S$

w$ = "CRAP": GOSUB writeserial: '  VERY IMPORTANT, Otherwise V doesn't work

w$ = "PRD": GOSUB writeserial
PRINT V: 'V returns the value read in a numeric format, V$ in a string.

w$ = "SEBB100;250": GOSUB writeserial

REM FOR i = 1 TO 3
w$ = "SBL200": GOSUB writeserial
w$ = "SBR0": GOSUB writeserial
 steps = steps + 200
   x = steps / 500
   PRINT x

REM FOR j = 1 TO 5000
REM NEXT j
REM NEXT i

w$ = "sBr600": GOSUB writeserial
END


REM  Writing to serial port
writeserial:
PRINT w$
PRINT #1, w$
GOSUB READSERIAL
RETURN

REM  Reading serial port
READSERIAL:
S$ = ""
```

```
IF LOC(1) = 0 THEN GOTO READSERIAL

REM Get received string into S$
Lp1:
IF INKEY$ <> "" THEN RETURN
C$ = INPUT$(1, #1)
S$ = S$ + C$
IF C$ <> ">" THEN GOTO Lp1

REM decode string (V$) and value (V)
VALIDERROR = True
ERRORCODE$ = ""
V$ = ""

FOR H = 1 TO LEN(S$)
IF MID$(S$, H, 1) = CHR$(7) THEN VALIDERROR = False
IF MID$(S$, H, 1) = "?" THEN ERRORCODE$ = MID$(S$, H + 1, 1)
NEXT H

IF (VALIDERROR = True AND ERRORCODE$ <> "") THEN GOSUB ERRORSUB: RETURN
IF LEN(S$) > 3 THEN V$ = RIGHT$(S$, LEN(S$) - 2): V$ = LEFT$(V$, LEN(V$) - 1)
V = VAL(V$)
RETURN

ERRORSUB:
PRINT
PRINT "Error #"; ERRORCODE$
RETURN
```

APPENDIX B -- INCORPORATED CODE

```
100 REM:  PITOTRAV.BAS - (1) PROGRAM CALIBRATES A DIFFERENTIAL PRESSURE
102 REM:  TRANSDUCER.   (2) PROGRAM CONTROLS A 2-D TRAVERSER AND DATA
104 REM:  ACQUISITION THAT ALLOWS A MATRIX OF PRESSURE READINGS TO BE
106 REM:  TAKEN IN THE WIND TUNNEL.
110 REM:  DATA ACQUISITION IS PERFORMED WITH A METRABYTE MODEL 16G1
130 REM:  HIGH SPEED, 12 BIT ADC BOARD
140 '**********************************************************************
150 REM:  AIR PROPERTY CALCULATIONS ASSUME AIR IS A DRY IDEAL GAS AND USE
160 REM:  THE SUTHERLAND EQUATION FOR THE TEMPERATURE DEPENDENCE OF
VISCOSITY
170 '**********************************************************************
180 '
185 DIM D%(15), G%(15), E(125), EE(125), DP(125), DPM(125), EDP(125), EDPM(125)
186 DIM DPI(125), U(125), P(125), XP(125), YP(125), ERMS(125), UM(125)
190 REM:  SUTHERLAND COEFFICIENTS
200 KEY OFF
210 CLS
220 B = 1.458E-06
230 S = 110.4
240 REM
250 REM:  AIR GAS CONSTANT AND SPECIFIC HEAT RATIO
260 R = 287!
270 GAM = 1.4
280 OPEN "O", #1, "C:\BPROG\DATAFILE\AIRPROPS.DAT"
290 PRINT "CALIBRATION WRITTEN TO FILE C:\BPROG\DATAFILE\PARA.DAT"
300 INPUT "CALIBRATION FILE HEADING (< 20 CHARACTERS)"; CHEAD$
310 PRINT USING "\                  \"; CHEAD$
320 PRINT USING "\       \   \       \"; DATE$; TIME$
330 PRINT #1, USING "\                  \"; CHEAD$
340 PRINT #1, USING "\       \   \       \"; DATE$; TIME$
350 REM
360 REM:  INPUT BAROMETER READING AND CALCULATE ATMOSPHERIC PRESSURE
370 REM
380 PRINT " "
390 INPUT "CORRECTED ATMOSPHERIC PRESSURE IN mm Hg"; PMM
400 PATM = 101325! * PMM / 760!
410 KPATM = PATM / 1000!
420 REM
430 REM:  INPUT TEMPERATURE AND CALCULATE OTHER TEMPERATURES
440 REM
450 INPUT "TEMPERATURE IN DEGREES C OR 1000+F"; TC
460 IF TC < 600 THEN 500
470 TF = TC - 1000!
480 TC = (TF - 32!) * 5 / 9!
490 GOTO 510
500 TF = (TC * 9 / 5) + 32!
510 TK = TC + 273.15
520 TR = TF + 459.7
530 REM
540 REM:  CALCULATE DYNAMIC VISCOSITY WITH SUTHERLAND EQUATION
550 REM
```

```
560 DVIS = B * SQR(TK) / (1! + (S / TK))
570 DVISE = DVIS / (1.488164)
580 REM
590 REM:  CALCULATE AIR DENSITY USING IDEAL GAS EQUATION OF STATE
600 REM
610 RHO = PATM / (R * TK)
620 REM
630 REM:  CALCULATE KINEMATIC VISCOSITY
640 REM
650 KVIS = DVIS / RHO
660 REM
670 REM:  PRINT RESULTS
680 REM
690 PRINT " "
700 PRINT USING "ATMOSPHERIC PRESSURE:     ###.###   kPa"; KPATM
710 PRINT #1, USING "  ####.###   KPA - ATMOSPHERIC PRESSURE"; KPATM
720 PRINT USING "TEMPERATURE:              ###.##    deg. C"; TC
730 PRINT #1, USING "    ###.##   DEG. C - TEMPERATURE"; TC
740 PRINT USING "ABSOLUTE TEMPERATURE:     ###.##    deg. K"; TK
750 PRINT #1, USING "    ###.##   DEG. K - ABS. TEMP."; TK
760 PRINT USING "AIR DENSITY:              #.####    kg/m3"; RHO
770 PRINT #1, USING "   ##.####   KG/M3 - AIR DENSITY"; RHO
780 PRINT USING "DYNAMIC VISCOSITY (mu):   ##.###^^^^ Pa-s"; DVIS
790 PRINT #1, USING "##.###^^^^   PA-S - DYNAMIC VISCOSITY"; DVIS
800 PRINT USING "KINEMATIC VISCOSITY (nu): ##.###^^^^ m2/s"; KVIS
810 PRINT #1, USING "##.###^^^^   M2/S - KINEMATIC VISCOSITY"; KVIS
820 RH2O = 998!
830 G = 9.80665
840 UCON = SQR(2! * G * RH2O / (1000! * RHO))
850 OPEN "o", #2, "c:\bprog\datafile\ucon"
860 PRINT #2, UCON
870 CLOSE #2
880 PRINT USING "VELOCITY = ###.### * [DELTA P (mm H2O)]**1/2 M/S"; UCON
890 PRINT #1, USING "##.###^^^^   (M/S)/(mm H2O)**1/2"; UCON
900 REM
910 REM:  TEST TO PERFORM CALIBRATION OR READ CAL. FILE
920 INPUT "ENTER 0 TO PERFORM CALIBRATION OR 1 TO READ CAL. FILE"; NTOCAL
930 OPEN "O", #2, "C:\BPROG\DATAFILE\NTOCAL"
940 PRINT #2, NTOCAL
950 CLOSE #2
955 DIM X%(6000), V(6000)
960 CLS
970 IF NTOCAL > 0 THEN 2260
980 GOSUB 10000
1340 N = M
1370 NCAL = 0!
1380 FOR I = 1 TO 50
1390 NCAL = NCAL + 1
1400 INPUT "MANOMETER READING (INCHES H20)"; DELP
1410 DP(NCAL) = DELP
1420 DPM(NCAL) = DELP * 25.4
1430 PRINT USING "READING = ##.### INCHES OF H20 (###.## MM)"; DP(NCAL); DPM(NCAL)
1510 '---------- DO M CCONVERSIONS TO ARRAY X%(J)  --------
1520 D%(0) = N
```

```
1530 D%(1) = VARPTR(X%(0))
1540 D%(2) = 1  'TRIGGER SSOURCE
1550 MD% = 4    'MODE 4 ----- A/D TO ARRAY V(J)
1560 CALL ABSOLUTE(MD%, D%(0), FLAG%)
1570 IF FLAG <> 0 THEN PRINT "ERROR #"; "IN MODE 4": STOP
1580 OPEN "O", #2, "C:\BPROG\DATAFILE\RAWC.DAT"
1590 PRINT #2, CHEAD$: PRINT #2, DATE$: PRINT #2, TIME$
1600 FOR J1 = 0 TO N - 1
1610 V(J1) = X%(J1) / 204.8
1620 WRITE #2, V(J1)
1630 NEXT J1
1640 CLOSE #2
1650 PRINT "OPENING ACQUISITION DATA FILE "
1660 PRINT "AND COMPUTING AVERAGE OUTPUT VOLTAGE"
1670 OPEN "I", #2, "C:\BPROG\DATAFILE\RAWC.DAT"
1680 INPUT #2, A$
1690 INPUT #2, B$
1700 INPUT #2, C$
1710 VSUM = 0!
1720  FOR J1 = 0 TO N - 1
1730  INPUT #2, V(J1)
1740  VSUM = VSUM + V(J1)
1750  NEXT J1
1760  VAVG = VSUM / J1
1770  E(NCAL) = VAVG
1780 PRINT USING "  CAL. POINT NO.    = ###"; NCAL
1790 PRINT USING "  MANOMETER DELTA P =  ##.### INCHES H2O"; DP(NCAL)
1800 PRINT USING "  MANOMETER DELTA P =  ##.### MM H20"; DPM(NCAL)
1810 PRINT USING "  TRANDUCER OUTPUT  =  ##.### VOLTS"; E(NCAL)
1820 REM
1830 REM
1840 CLOSE #2
1845 PRINT " "
1850 INPUT "0 TO WRITE CAL. POINT TO FILE, -1 TO DELETE"; NFLAG
1860 IF NFLAG < 0 THEN NCAL = NCAL - 1
1870 PRINT " "
1880 INPUT "0 TO CONTINUE OR -1 TO END CALIBRATION"; NFLAG
1890 IF NFLAG < 0 THEN 1950
1900 NEXT I
1940 CLOSE #1
1950 PRINT USING "CALIBRATION COMPLETED WITH ### POINTS"; NCAL
1960 CLOSE #1
1970 OPEN "O", #1, "C:\BPROG\DATAFILE\PTCALOUT.DAT"
1980 PRINT #1, USING "#####  CALIBRATION POINTS"; NCAL
1990 PRINT " "
2000 PRINT "  N    DP     DPM     E    "
2010 PRINT #1, "  N     DP          DPM        E    "
2020 PRINT "    (IN H2O) (mm H2O) (VOLTS)"
2030 PRINT #1, "    (IN H2O) (mm H2O) (VOLTS)"
2040 SX = 0!
2050 SY = 0!
2060 SXY = 0!
2070 SX2 = 0!
2080 FOR M = 1 TO NCAL
```

```
2090 SX = SX + DPM(M)
2100 SY = SY + E(M)
2110 SXY = SXY + (DPM(M) * E(M))
2120 SX2 = SX2 + (DPM(M) * DPM(M))
2130 PRINT USING "###  ##.###  ###.### ##.####"; M; DP(M); DPM(M); E(M)
2140 PRINT #1, USING "#####  ##.###^^^^  ##.###^^^^  ##.###^^^^"; M; DP(M); DPM(M); E(M)
2150 NEXT M
2160 HH = SX2 - SX ^ 2 / NCAL
2170 B = (SXY - (SX * SY / NCAL)) / (SX2 - ((SX ^ 2) / NCAL))
2180 A = (SY - (B * SX)) / NCAL
2190 PRINT USING " E = ##.#### + (#.#### * DPM)"; A; B
2200 PRINT #1, USING "##.####^^^^ ##.####^^^^ :A,B -> E = A + (B *DPM)"; A; B
2210 CLOSE #1
2220 PRINT "END OF CALIBRATION LOOP"
2230 REM
2240 REM:  END OF CALIBRATION LOOP
2250 GOTO 2560
2260 OPEN "I", #2, "C:\BPROG\DATAFILE\PTCALIN.DAT"
2400 INPUT #2, NCAL, DUMMY$
2410 INPUT #2, DHEADS$
2420 INPUT #2, DHEADS$
2430 PRINT #1, USING "#####  CALIBRATION POINTS"; NCAL
2440 PRINT #1, "    N      DP           DPM         E    "
2450 PRINT #1, "      (IN H20)      (mm H2O)    (VOLTS)"
2470 FOR K = 1 TO NCAL
2480 INPUT #2, M, EDP(M), EDPM(M), EE(M)
2490 PRINT #1, USING "#####  ##.###^^^^  ##.###^^^^  ##.###^^^^"; M; EDP(M); EDPM(M); EE(M)
2500 NEXT K
2510 INPUT #2, A, B
2520 PRINT USING " A = ##.####^^^^  B = ##.####^^^^"; A; B
2530 PRINT #1, USING "##.####^^^^ ##.####^^^^ :A,B - E = A + (B *DPM)"; A; B
2540 CLOSE #1
2550 CLOSE #2
2560 OPEN "O", #3, "C:\BPROG\DATAFILE\PITOT.DAT"
2570 PRINT "DYNAMIC PRESSURE DATA  'C:\BPROG\DATAFILE\PITOT.DAT'"
2580 INPUT "PRESSURE MEASUREMENT FILE HEADING (<20 CHARACTERS)"; PHEAD$
2590 PRINT USING "\                \"; PHEAD$
2600 PRINT USING "\       \    \        \"; DATE$; TIME$
2610 PRINT #3, USING "\                \"; PHEAD$
2620 PRINT #3, USING "\       \    \        \"; DATE$; TIME$
2630 OPEN "I", #2, "C:\BPROG\DATAFILE\NTOCAL"
2640 INPUT #2, NTOCAL
2650 CLOSE #2
2660 IF NTOCAL <= 0 THEN 2870
2670 GOSUB 10000
2675 GOTO 2925
2850 REM DIM X%(4000),V(4000)
2860 REM DIM E(100),DPM(100),DP(100)
2870 GOSUB 10320
2875 N = M
2920 REM DIM DPI(100),U(100),P(100),RI(100),RM(100)
2921 DFLAG = 0
2925 NP = 0
2926 XP = 0
```

```
2927 YP = 30
2928 'REM THIS PORTION OPERATES THE ITC232-A, ISSUES COMMANDS TO
2929 'REM THE STEPPER MOTORS, AND TAKES THE DATA
2930 CLS
2931 TRUE = 1
2932 FALSE = 0
2933 REM OPEN COM PORT USE COM1 OR COM2
2934 OPEN "COM1: 9600,N,8,1,RS,CS0,DS0,CD0" FOR RANDOM AS #1
2936 PRINT "PLEASE RESET THE ITC232-A"
2937 PRINT
2938 GOSUB 11080
2939 PRINT S$
2940 W$ = "CRAP": GOSUB 11020: ' VERY IMPORTANT, OTHERWISE V DOESN'T WORK
2941 W$ = "PRD": GOSUB 11020
2942 PRINT V: 'V RETURNS THE VALUE READ IN A NUMERIC FORMAT, V$ IN A STRING
2943 W$ = "SEBB75;250": GOSUB 11020
2944 W$ = "SEC": GOSUB 11020
2945 FOR I = 1 TO 500
2946      NP = NP + 1
2947      XP(I) = XP
2948      YP(I) = YP
2949      PRINT USING "X-POSITION = ##.### (INCHES)"; XP(I)
2950      PRINT USING "Y-POSITION = ##.### (INCHES)"; YP(I)
2951      PRINT "HIT ESC KEY TO STOP"
2953      IF INKEY$ = CHR$(27) THEN PRINT "stopping": STOP
2960      GOTO 3150
2970      PRINT
3150      D%(0) = N
3160      D%(1) = VARPTR(X%(0))
3170  D%(2) = 1  'TRIGGER SOURCE
3180  MD% = 4    'MODE 4 ----- A/D TO ARRAY V(J)
3190  CALL ABSOLUTE(MD%, D%(0), FLAG%)
3200  IF FLAG <> 0 THEN PRINT "ERROR #"; "IN MODE 4": STOP
3210  OPEN "O", #2, "C:\BPROG\DATAFILE\RAWD.DAT"
3220  PRINT #2, PHEAD$
3230  FOR J1 = 0 TO N - 1
3240  V(J1) = X%(J1) / 204.8
3250  WRITE #2, V(J1)
3260  NEXT J1
3270  CLOSE #2
3280  REM PRINT "OPENING ACQUIRED PROFILE DATA FILE"
3290  REM PRINT "AND COMPUTING AVERAGE OUTPUT VOLTAGE"
3300  OPEN "I", #2, "C:\BPROG\DATAFILE\RAWD.DAT"
3310  INPUT #2, AHEAD$
3320  VSUM = 0!
3325  VSUM2 = 0
3330  FOR J = 0 TO N - 1
3340  INPUT #2, V(J)
3350  VSUM = VSUM + V(J)
3355  VSUM2 = VSUM2 + (V(J) * V(J))
3360  NEXT J
3370  CLOSE #2
3380  G = 9.80665
3390  RH2O = 998!
```

```
3400  VBAR = VSUM / J
3405  VBAR2 = VSUM2 / J
3410  E(NP) = VBAR
3411  ECHEK = (VBAR2 - (VBAR * VBAR))
3412  IF ECHEK < 0 THEN 3414
3413  GOTO 3416
3414  ERMS(NP) = -1
3415  GOTO 3420
3416  ERMS(NP) = SQR(VBAR2 - (VBAR * VBAR))
3420  DPM(NP) = (E(NP) - A) / B
3430  OPEN "i", #2, "c:\bprog\datafile\ucon"
3440  INPUT #2, UCON
3450  P(NP) = DPM(NP) * RH2O * G / 1000!
3459  IF P(NP) < 0! THEN 3465
3460  UM(NP) = UCON * SQR(DPM(NP))
3461  U(NP) = UM(NP) * 3.2808
3462  GOTO 3470
3465  P(NP) = 0!
3470  CLOSE #2
3480  PRINT "   XPOS  YPOS       EBAR  DP(MM)  P(PA)     U(FT/S)"
3485  PRINT USING "##.###   ##.###   ##.###^^^^  ###.## ##.###^^^^ ###.###"; XP(NP); YP(NP);
E(NP); DPM(NP); P(NP); U(NP)
3490  PRINT " "
3491  REM INPUT "0 TO PRINT POINT TO FILE OR -1 TO DELETE"; NFLAG
3492  REM IF NFLAG > -1 THEN 3500
3493  REM NP = NP - 1
3494  REM GOTO 3530
3500  PRINT #3, USING "##.###^^^^ ##.###^^^^ ##.###^^^^##.###^^^^"; XP(NP); YP(NP); E(NP);
ERMS(NP)
3510  REM  PRINT #3,USING "##### ##.### ##.###^^^^ ##.###  ###.## ##.###^^^^
##.###^^^^";NP,RI(NP),RM(NP),E(NP),DPM(NP),P(NP),U(NP)
3520  REM
3530  PRINT " "
3535  REM INPUT "0 TO CONTINUE OR -1 TO END MEASUREMENTS"; NFLAG
3540  REM IF NFLAG < 0 THEN 3560
3550  IF YP = .5 THEN 3730
3560  YP = YP - 3
3570  IF YP < 3 THEN 3600
3580  W$ = "SCL600": GOSUB 11020
3590  GOTO 3760
3600  XP = XP + 2.525
3610  IF XP > 26 THEN 3660
3620  YP = 30
3630  W$ = "SCR6000": GOSUB 11020
3640  W$ = "SBR505": GOSUB 11020
3650  GOTO 3760
3660  INPUT "ENTER 1 TO TAKE DATA NEAR WALL, 0 TO END DATA COLLECTION"; XFLAG
3670  IF XFLAG = 0 THEN 3770
3680  INPUT "MOVE SET SCREW POSITION AND ENTER 1 WHEN FINISHED"; YFLAG
3690  IF YFLAG <> 1 THEN 3660
3700  XP = 25.25
3710  YP = .5
3720  GOTO 3760
3730  XP = XP - 2.525
```

```
3740  IF XP < 5 THEN 3780
3750  W$ = "SBL505": GOSUB 11020
3760 NEXT I
3770 W$ = "SBL5050": GOSUB 11020
3780 W$ = "SCR6000": GOSUB 11020
3790 PRINT USING "MEASUREMENTS COMPLETED WITH ### POINTS"; NP
3800 MNP = -NP
3810 REM PRINT #3, MNP
3820 PRINT " "
3830 PRINT #3, " "
3840 PRINT "    XPOS   YPOS   EBAR   DP(MM)     U(FT/S)"
3850 PRINT " "
3860 PRINT #3, "  XPOS   YPOS     EBAR      ERMS       U(FT/S)"
3870 FOR M = 1 TO NP
3880 PRINT USING "##.###   ##.###  ##.### ##.###^^^^ ###.###^^^^"; XP(M); YP(M); E(M);
DPM(M); U(M)
3890 PRINT #3, USING "##.###   ##.###   ##.###   ##.###^^^^ ###.###^^^^"; XP(M); YP(M); E(M);
ERMS(M); U(M)
3900 REM  PRINT #3,USING "##### ##.### ##.###^^^^ ##.###  ###.## ##.###^^^^
##.###^^^^";M,RI(M),RM(M),E(M),DPM(M),P(M),U(M)
3910 NEXT M
3920 CLOSE #3
3930 PRINT " "
3940 PRINT "AIR PROPERTIES WRITTEN TO        C:\BPROG\DATAFILE\AIRPROPS.DAT"
3950 PRINT "TRANSDUCER CALIBRATION WRITTEN TO
C:\BPROG\DATAFILE\PTCALOUT.DAT"
3960 PRINT "MEASUREMENT DATA WRITTEN TO       C:\BPROG\DATAFILE\PITOT.DAT"
3970 REM SHELL "DATA"
3980 PRINT "    "
3990 END
10000 REM:  INITIALIZATION SUBROUTINE
10010 KEY OFF
10020 CLS
10030 PRINT "SETTING SCREEN"
10040 SCREEN 0, 0, 0: WIDTH 80
10050 DEF SEG = &H6000
10060 PRINT "SEGMENT DEFINED, BLOADING"
10070 PRINT USING "SG = ########"; SG
10080 BLOAD "DASG.BIN", 0
10090 REM DIM D%(15)
10100 '----------  INITIALIZE WITH MODE 0      -----------
10110 'OPEN "DASG.ADR" FOR INPUT AS #1  'GET BASE I/O ADDRESS
10120 D%(0) = 816
10130 D%(1) = 2    'INTERRUPT LEVEL
10140 D%(2) = 1    'D.M.A. LEVEL
10150 DASG = 0     'CALL OFFSET
10160 FLAG% = 0    'ERROR VARIABLE
10170 MD% = 0      'MODE 0-------INITIALIZE
10180 PRINT "CALLING MODE 0"
10190 CALL ABSOLUTE(MD%, D%(0), FLAG%)
10200 IF FLAG% <> 0 THEN PRINT "INSTALLATION ERROR"'HALT ON ERROR
10210 INPUT "ENTER LOWER CHANNEL SCAN LIMIT NUMBER  :", L%
10220 INPUT "ENTER UPPER CHANNEL SCAN LIMIT NUMBER  :", U%
10230 '----------  PROMPT FOR SCAN LIMITS AND SET USING MODE 1 ---------
```

```
10240 D%(0) = L%
10250 D%(1) = U%
10260 MD% = 1
10270 PRINT "CALLING MODE 1"
10280 CALL ABSOLUTE(MD%, D%(0), FLAG%)
10290 IF FLAG% <> 0 THEN PRINT "ERROR #"; FLAG%; "IN SETTING SCAN LIMITS": STOP
10300 REM DIM G%(15)
10310 REM        DIM X%(8000),V(8000)
10320 INPUT "ENTER THE SAMPLING RATE (SAMPLES/SECOND):", SF
10330 SRATE = SF
10340 '---------- SET TIMER RATE TO TRIGGER A/D USING MODE 17 --------
10350 D%(0) = 200!
10360 D%(1) = 10000000# / (200 * SF)
10370 MD% = 17
10380 PRINT "CALLING MODE 17"
10390 CALL ABSOLUTE(MD%, D%(0), FLAG%)
10400 IF FLAG% <> 0 THEN PRINT "ERROR #"; FLAG%; " IN SETTING TIMER ": STOP
10410 INPUT "ENTER THE NUMBER OF CONVERSIONS:", M
10420 RLENGTH = M / SF
10430 PRINT USING "SAMPLING RATE  = ##.###^^^^ SAMPLES/SECOND"; SF
10440 PRINT USING "NO. OF SAMPLES = #####"; M
10450 PRINT USING "RECORD LENGTH  = ##.###^^^^ SECONDS"; RLENGTH
10460 NSAMPS = M
10470 N = M
10480 REM DIM X%(N-1),V(N-1)
10490 '----------- SELECT CHANNEL GAINS --------------
10500 MD% = 21
10510 PRINT "CALLING MODE 21"
10515 PRINT " "
10520 FOR I1 = 0 TO 15
10530 G%(I1) = 0
10540 CALL ABSOLUTE(MD%, G%(0), FLAG%)
10550 IF FLAG% <> 0 THEN PRINT "ERROR #"; FLAG%; "IN MODE 21": STOP
10560 NEXT I1
10570 RETURN
11000 REM WRITING TO SERIAL PORT
11020 PRINT W$
11030 PRINT #1, W$
11040 GOSUB 11080
11050 RETURN
11060 REM READING SERIAL PORT
11080 S$ = ""
11090 IF LOC(1) = 0 THEN GOTO 11080
11100 REM GET RECEIVED STRING INTO S$
11120 IF INKEY$ <> "" THEN RETURN
11130 C$ = INPUT$(1, #1)
11140 S$ = S$ + C$
11150 IF C$ <> ">" THEN GOTO 11120
11160 REM DECODE STRING (V$) AND VALUE (V)
11170 VALIDERROR = TRUE
11180 ERRORCODE$ = ""
11190 V$ = ""
11200 FOR H = 1 TO LEN(S$)
11210 IF MID$(S$, H, 1) = CHR$(7) THEN VALIDERROR = FALSE
```

```
11220 IF MID$(S$, H, 1) = "?" THEN ERRORCODE$ = MID$(S$, H + 1, 1)
11230 NEXT H
11240 IF (VALIDERROR = TRUE AND ERRORCODE$ <> "") THEN GOSUB 11290: RETURN
11250 IF LEN(S$) > 3 THEN V$ = RIGHT$(S$, LEN(S$) - 2): V$ = LEFT$(V$, LEN(V$) - 1)
11260 V = VAL(V$)
11270 RETURN
11290 PRINT
11300 PRINT "ERROR #"; ERRORCODE$
11310 RETURN
```

VITA

Justin Lawrence Cates

Candidate for the Degree of

Master of Science

Thesis: CONSTRUCTION AND OPERATION OF THE MAE FLEXIBLE USE WIND TUNNEL FACILITY

Major Field: Mechanical and Aerospace Engineering

Biographical:

Personal Data: Born in Elk City, Oklahoma, On April 11, 1965, the son of Kenneth and Patricia Cates.

Education: Graduated from Leedey High School, Leedey, Oklahoma in May 1983; received Bachelor of Science degree in Engineering Technology in May 1992. Completed the requirements for the Master of Science degree with a major in Mechanical and Aerospace Engineering at Oklahoma State University in Spring 1996.

Experience: Raised on a farm outside of Leedey, Oklahoma; employed by Stillwater Country Club as Assistant Golf Course Superintendent from 1990-1995; employed by Oklahoma State University, Department of Mechanical and Aerospace Engineering as a graduate research assistant in 1995.

Professional Memberships: AIAA, Sigma Gamma Tau National Honor Society.

nyb | The New York Blower Company

7660 Quincy Street
Willowbrook, IL. 60521

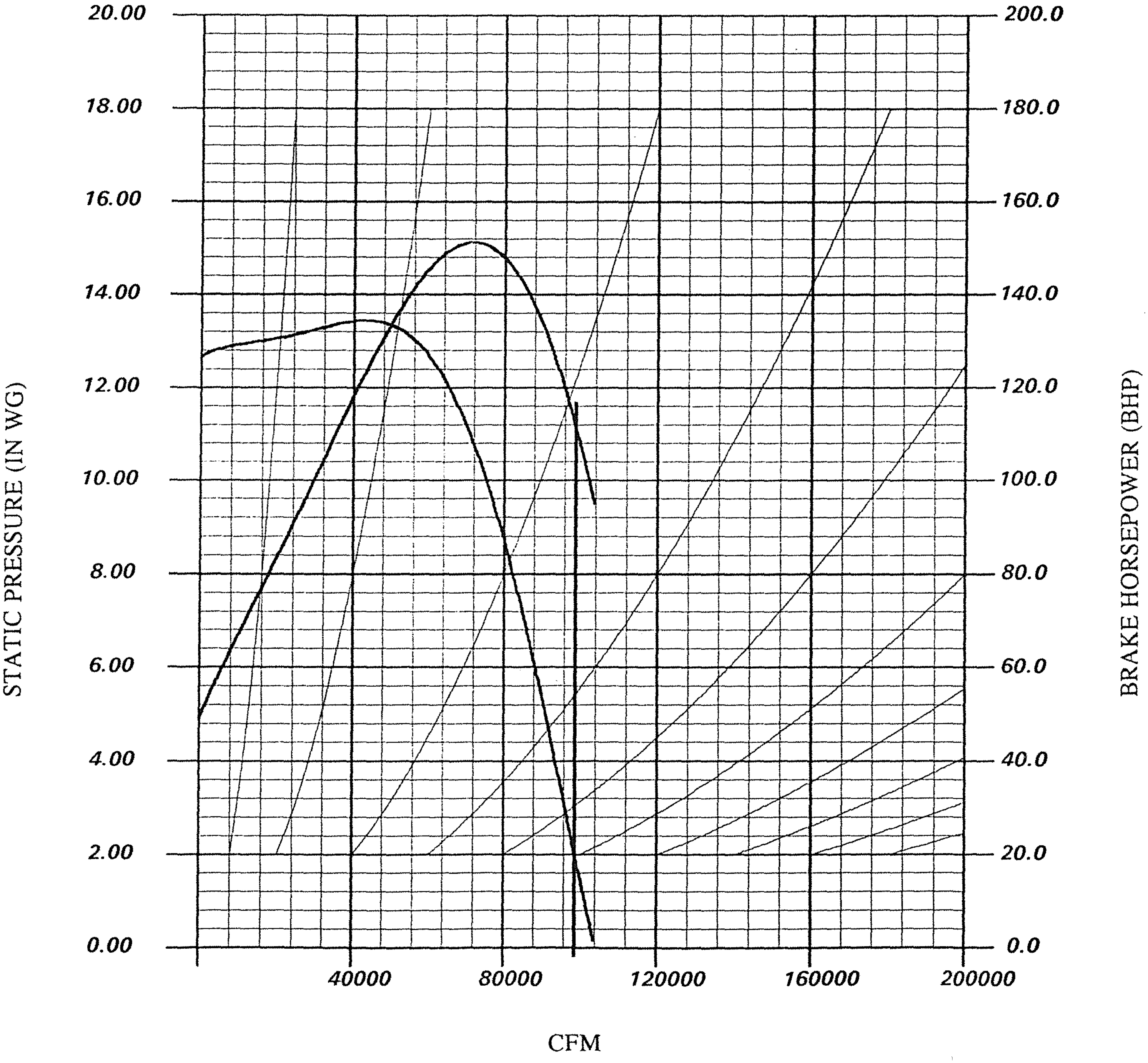
PERFORMANCE CURVE

To determine Performance at another RPM multiply
CFM x K
SP x K^2
BHP x K^3
where K is new RPM divided by RPM shown at right.

CUST. NO	: R314606			
CUSTOMER	: OKLAHOMA STATE UNIVERSITY			
TAGGING	: REQUISITION #R314606			
FAN TYPE	: Single-Width AcF			
FAN SIZE	: 54			
CFM	: 99000	TEMP :	100	deg F
SP	: 2.00	DENS :	0.068	LB/FT^3
RPM	: 1221			
BHP	: 111.55	FILE :	K05616/100	EKL

DATE : May 6 1994

PERFORMANCE OPTIONS :



nyb | The New York Blower Company®
7660 QUINCY STREET—WILLOWBROOK, ILLINOIS 60521-5596

INSTALLATION MAINTENANCE, OPERATING INSTRUCTIONS

IM-100

CENTRIFUGAL FANS
AcF/PLR, AF-Forty, RTS

CAUTION

THIS MACHINE HAS MOVING PARTS THAT CAN CAUSE SERIOUS BODILY INJURY. BEFORE OPERATING OR PERFORMING MAINTENANCE, THE FOLLOWING PRECAUTIONS MUST BE TAKEN.

1. **MAKE SURE ALL MOVING PARTS ARE SHIELDED FROM PERSONNEL AND FALLING OBJECTS.**
2. **READ THE INSTALLATION AND MAINTENANCE INSTRUCTIONS, AS WELL AS THE RECOMMENDED SAFETY PRACTICES MANUAL FURNISHED WITH THIS UNIT.**
3. **DO NOT OPERATE AT SPEEDS OR TEMPERATURES HIGHER THAN PUBLISHED FOR THE SPECIFIC OPERATING CONDITIONS FOR WHICH THE MACHINE WAS PURCHASED.**

A FAILURE TO TAKE THESE PRECAUTIONS COULD RESULT IN SERIOUS BODILY INJURY AND PROPERTY DAMAGE.

98-0250

A WORD ABOUT SAFETY

The above **CAUTION** decal appears on all **nyb** fans. Air moving equipment involves electrical wiring, moving parts, and air velocity or pressure which can create safety hazards if the equipment is not properly installed, operated and maintained. To minimize this danger, follow these instructions as well as the additional instructions and warnings on the equipment itself.

All installers, operators and maintenance personnel should study AMCA Publication 410, "Recommended Safety Practices for Air Moving Devices", which is included as part of every shipment. Additional copies can be obtained by writing to The New York Blower Company, 7660 Quincy Street, Willowbrook, IL 60521-5596.

ELECTRICAL DISCONNECTS

Every motor driven fan should have an independent disconnect switch to isolate the unit from the electrical supply. It should be near the fan and must be capable of being locked by maintenance personnel while servicing the unit, in accordance with OSHA procedures.

MOVING PARTS

All moving parts must have guards to protect personnel. Safety requirements vary, so the number and type of guards needed to meet company, local and OSHA standards must be determined and specified by the user. Never start a fan without having all safety guards installed. Check regularly for damaged or missing guards and do not operate any fan with guards removed. Fans can also become dangerous because of potential "windmilling," even though all electrical power is disconnected. Always block the rotating assembly before working on any moving parts.

AIR PRESSURE AND SUCTION

In addition to the normal dangers of rotating machinery, fans present another hazard from the suction created at the fan inlet. This suction can draw materials into the fan where they become high velocity projectiles at the outlet. It can also be extremely dangerous to persons in close proximity to the inlet, as the forces involved can overcome the strength of most individuals. Inlets and outlets that are not ducted should be screened to prevent entry and discharge of solid objects.

ACCESS DOORS

DANGER

DO NOT OPEN UNTIL THE POWER SUPPLY HAS BEEN LOCKED OFF AND THE SHAFT HAS STOPPED ROTATING.

FAILURE TO DO THIS CAN RESULT IN SERIOUS BODILY INJURY.

98-0249

The above **DANGER** decal is placed on all **nyb** cleanout doors. These doors, as well as access doors to the duct system, should never be opened while the fan is in operation. Serious injury could result from the effects of air pressure or suction.

Quick-opening doors must have the door handle bolts securely tightened to prevent accidental or unauthorized opening. Bolted doors must be tightened for the same reason.

RECEIVING AND INSPECTION

The fan and accessories should be inspected on receipt for any shipping damage. Turn the wheel by hand to see that it rotates freely and does not bind. If dampers or shutters are provided, check these accessories for free operation of all moving parts.

F.O.B. factory shipping terms require that the receiver be responsible for inspecting the equipment upon arrival. Note damage or shortages on the Bill of Lading and file any claims for damage or loss in transit. **nyb** will assist the customer as much as possible; however, claims must be originated at the point of delivery.

HANDLING AND STORAGE

Fans should be lifted by the base, mounting supports, or lifting eyes only. Never lift a fan by the wheel, shaft, motor, motor bracket, housing inlet, outlet, or any fan part not designed for lifting. A spreader should always be used to avoid damage.

On direct drive Arrangement 7 or 8 fans, lifting holes are provided in the motor base to assist in handling the fan assembly. These lifting holes should be used in conjunction with the lifting eyes when lifting and positioning the fan onto its foundation. A heavy round steel bar or appropriate fixture can be passed through the lifting holes to simplify attachment of the lifting device. Be sure to follow all local safety codes when moving heavy equipment.

Whenever possible, fans and accessories should be stored in a clean, dry location to prevent rust and corrosion of steel components. If outdoor storage is necessary, protection should be provided. Cover the inlet and outlet to prevent the accumulation of dirt and moisture in the housing. Cover motors with waterproof material. Refer to the bearing section for further storage instructions.

Check dampers for free operation and lubricate moving parts prior to storage. Inspect the stored unit periodically. **Rotate the wheel by hand every two weeks to redistribute grease on internal bearing parts.**

FAN INSTALLATION

nyb wheels are dynamically balanced when fabricated. Fully assembled fans are test run at operating speeds to check the entire assembly for conformance to **nyb** vibration limits. Nevertheless, all units must be adequately supported for smooth operation. Ductwork or stacks should be independently supported as excess weight may distort the fan housing and cause contact between moving parts. Where vibration isolators are used, consult the **nyb** certified drawing for proper location and adjustment.

Slab-Mounted Units

A correctly designed and level concrete foundation provides the best means of installing floor-mounted fans. The mass of the base must maintain the fan/driver alignment, absorb normal vibration, and resist lateral loads. The overall dimensions of the concrete base should extend at least six inches beyond the base of the fan. The weight of the slab should be two to three times the weight of the rotating assembly, including the motor. The foundation requires firmly anchored fasteners such as the anchor bolts shown in Figure 1. Hammer-drilled expansion fasteners can be used in less demanding applications.

Move the fan to the mounting location and lower it over the anchor bolts, leveling the fan with shims around the bolts. Fasten the fan securely. When grout is used, shim the fan at least 3/4-inch from the concrete base. (See Figure 1.) When isolation is used, check the **nyb** certified drawing for installation instructions.

Elevated Units

When an elevated or suspended structural steel platform is used, it must have sufficient bracing to support the unit load and prevent side sway. The platform should be of welded construction to maintain permanent alignment of all members.

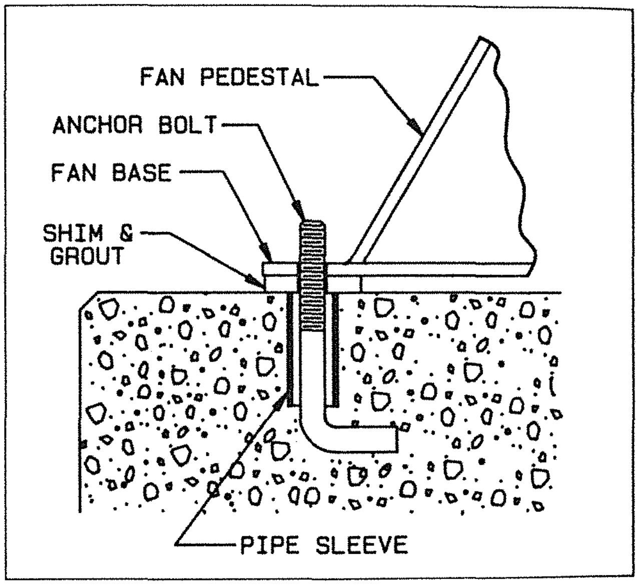


Figure 1

V-BELT DRIVE

Installation

1. Remove all foreign material from the fan and motor shafts. Coat shafts with machine oil for easier mounting. Mount the belt guard backplate at this time if partial installation is required prior to sheave mounting.
2. Mount sheaves on shafts after checking sheave bores and bushings for nicks or burrs. Avoid using force. If resistance is encountered, lightly polish the shaft with crocus cloth until the sheave slides on freely. Tighten tapered bushing bolts sequentially so that equal torque is applied to each.
3. Adjust the motor on its base to a position closest to the fan shaft. Install belts by working each one over the sheave grooves until all are in position. Never pry the belts into place. On **nyb** packaged fans, sufficient motor adjustment is provided for easy installation of the proper size belts.
4. Adjust sheaves and the motor shaft angle so that the sheaves faces are in the same plane. Check this by placing a straightedge across the faces of the sheave. Any gap between the edge and sheave faces indicates misalignment. Important: This method is only valid when the width of the surface between the belt edge and the sheave face is the same for both sheaves. When they are not equal, or when using adjustable-pitch sheaves, adjust so that all belts have approximately equal tension. Both shafts should be at the right angles to the center belt.

Belt Tensioning

1. Check belt tension with a tensioning gage and adjust using the motor slide base. Excess tension shortens bearing life while insufficient tension shortens belt life, can reduce fan performance and may cause vibration. The lowest allowable tension is that which prevents slippage under full load. Belts may slip during startup, but slipping should stop as soon as the fan reaches full speed. For more precise tensioning methods, consult the drive manufacturer's literature.

2. Recheck setscrews, rotate the drive by hand and check for rubbing, then complete the installation of the belt guard.
3. Belts tend to stretch somewhat after installation. Recheck tension after several days of operation. Check sheave alignment as well as setscrew and/or bushing bolt tightness.

COUPLING

Coupling alignment should be checked after installation and prior to start up. Alignment is set at the factory, but shipping, handling and installation can cause misalignment. Fans with wheel sizes 40" and larger are normally shipped with the flexible element removed to minimize potential for damage (see section on alignment procedure). Also check for proper coupling lubrication. For details on lubrication and for alignment tolerances on the particular coupling supplied, see the manufacturer's installation and maintenance supplement in the shipping envelope.

Installation

Most **nyb** fans are shipped with the coupling installed. In cases where the drive is assembled after shipping, install the coupling as follows:

1. Remove all foreign material from fan and motor shafts and coat with machine oil for easy mounting of coupling halves.
2. Mount the coupling halves on each shaft, setting the gap between the faces specified by the manufacturer. Avoid using force. If mounting difficulty is encountered, lightly polish the shaft with crocus cloth until the halves slide on freely.

Alignment

1. Align the coupling to within the manufacturer's limits for parallel and angular misalignment (see Figure 2). A dial indicator can also be used for alignment where greater precision is desired. Adjustments should be made by moving the motor to change shaft angle, and by the use of foot shims to change motor shaft height. Do not move the fan shaft or bearing.
2. When correctly aligned, install the flexible element and tighten all fasteners in the coupling and motor base. Lubricate the coupling if necessary.
3. Recheck alignment and gap after a short period of operation, and recheck the tightness of all fasteners in the coupling assembly.

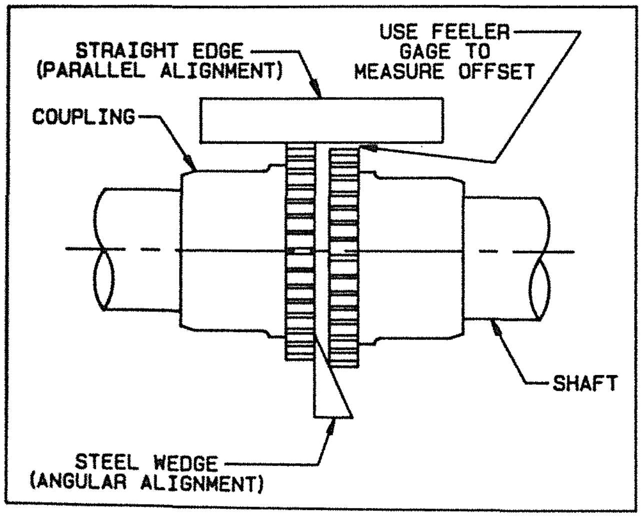


Figure 2

START-UP

Safe operation and maintenance includes the selection and use of appropriate safety accessories for the specific installation. This is the responsibility of the system designer and requires consideration of equipment location and accessibility as well as adjacent components. All safety accessories must be installed properly prior to start up.

Safe operating speed is a function of system temperature and wheel design. Do not under any circumstances exceed the maximum safe fan speed published in the **nyb** bulletin, which is available from your **nyb** field sales representative.

Procedure

1. If the drive components are not supplied by **nyb**, verify with the manufacturer that the starting torque is adequate for the speed and inertia of the fan.
2. Inspect the installation prior to starting the fan. Check for any loose items or debris that could be drawn into the fan or dislodged by the fan discharge. Check the interior of the fan as well. Turn the wheel by hand to check for binding.
3. Check drive installation and belt tension.
4. Check the tightness of all setscrews, nuts and bolts. When furnished, tighten hub setscrews with the wheel oriented so that the setscrew is positioned underneath the shaft.
5. Install all remaining safety devices and guards. Verify that the supply voltage is correct and wire the motor. "Bump" the starter to check for proper wheel rotation.
6. Use extreme caution when testing the fan with ducting disconnected. Apply power and check for unusual sounds or excessive vibration. If either exists, see the section on Common Fan Problems. To avoid motor overload, do not run the fan for more than a few seconds if ductwork is not fully installed. Without the ductwork attached, normal operating speed may not be obtained without motor overload. Once ductwork is attached, check for correct fan speed and complete installation. Ductwork and guards must be fully installed for safety.
7. Setscrews should be rechecked after a few minutes, eight hours and two weeks of operation (see Tables 1 & 2 for correct tightening torques).

NOTE: Shut the fan down immediately if there is any sudden increase in fan vibration.

WHEEL SETSCREW TORQUES
TABLE 1

Setscrew Size Diameter (in.)	Carbon Steel Setscrew Torque*	
	lb.-in.	lb.-ft.
1/4"	75	6.2
5/16"	144	12
3/8"	252	21
7/16"	396	33
1/2"	600	50
5/8"	1164	97
3/4"	2016	168
7/8"	3204	267
1"	4800	400

***Stainless Steel setscrews are not hardened and should not be tightened to more than 1/2 the values shown.**

BEARING SETSCREW TORQUE, lb.-in.
TABLE 2

Setscrew Diameter	Manufacturer				
	Link-Belt	Seal-master	SKF	McGill	Dodge
#10	40	—	35	35	—
1/4"	90	65	50	85	—
5/16"	185	125	165	165	160
3/8"	325	230	290	290	275
7/16"	460	350	350	—	—
1/2"	680	500	620	—	600
5/8"	1350	1100	1325	—	1200
3/4"	2350	—	—	—	2000

Note: Split pillow block bearings are fixed to the shaft with tapered sleeves and generally do not have setscrews.

FAN MAINTENANCE

nyb fans are manufactured to high standards with quality materials and components. Proper maintenance will ensure a long and trouble-free service life.

Do not attempt any maintenance on a fan unless the electrical supply has been completely disconnected and locked. In many cases, a fan can windmill despite removal of all electrical power. The rotating assembly should be blocked securely before attempting maintenance of any kind.

The key to good fan maintenance is regular and systematic inspection of all fan parts. Inspection frequency is determined by the severity of the application and local conditions. Strict adherence to an inspection schedule is essential.

Regular fan maintenance should include the following:

1. Check the fan wheel for any wear or corrosion as either can cause catastrophic failures. Check also for the build up of material which can cause unbalance resulting in vibration, bearing wear and serious safety hazards. Clean or replace the wheel as required.
2. Check the V-belt drive for proper alignment and tension (see section on V-belt drives). If belts are worn, replace them as a set, matched to within manufacturer's tolerances. Lubricate the coupling of direct-drive units and check for alignment (see section on couplings).
3. Lubricate the bearings, but do not overlubricate (see the bearing section for detailed specifications).
4. Ceramic-felt shaft seals require no maintenance, although worn seals should be replaced. When lip-type shaft seals are provided, lubricate them with "NEVER-SEEZ" or other anti-seize compound.
5. During any routine maintenance, all setscrews and bolts should be checked for tightness. See the table for correct torques.
6. When installing a new wheel or cone, the proper wheel-to-inlet cone clearance must be maintained (see Figure 3 for correct full-width wheel dimensions).

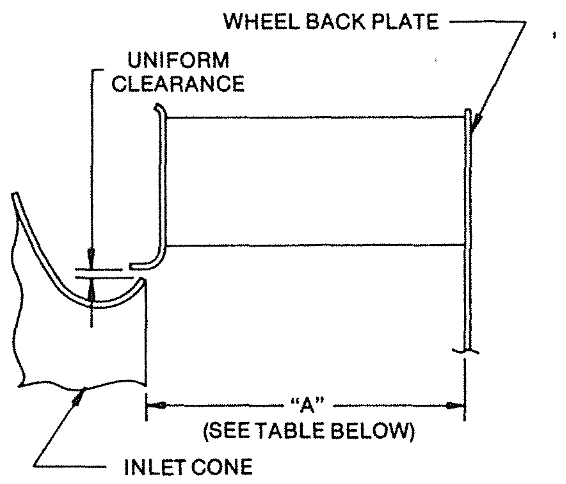
WHEEL BALANCE

Airstreams containing particulate or chemicals can cause abrasion or corrosion of the fan parts. This wear is often uneven and can lead to significant wheel unbalance over time. When such wear is discovered, a decision must be made as to whether to rebalance or replace the wheel.

The soundness of all parts should be determined if the original thickness of components is reduced. Be sure there is no hidden structural damage. The airstream components should also be cleaned to remove any build up of foreign material. Specialized equipment can be used to rebalance a cleaned wheel that is considered structurally sound.

Balance weights should be rigidly attached at a point that will not interfere with the housing nor disrupt airflow. Remember that centrifugal forces can be extremely high at the outer radius of a fan wheel. Welding is the preferred method of balance weight attachment. Be sure to ground the welder directly to the fan wheel. Otherwise, the welding current could pass through the fan bearings and destroy them.

FULL WIDTH WHEEL-CONE CLEARANCES
(contact nyb for partial width dimensions)



Fan Size		"A" Dimension		
AF-Forty, RTS	AcF/PLR	AcF/PLR	AF-Forty	RTS
—	12	4-1/2	—	—
—	15	5-1/2	—	—
—	18	7	—	—
27	22	8-1/4	8	7-3/8
30	24	9-1/4	8-1/4	8-1/8
33	27	10-1/8	9-3/4	9
36	30	11-1/4	10-7/8	10
40	33	12-1/4	12	11
44	36	13-3/8	13-1/4	12-1/8
49	40	14-3/4	14-5/8	13-3/8
54	44	16-1/4	16-1/8	14-3/4
60	49	17-7/8	17-3/4	16-3/8
66	54	19-3/4	19-5/8	18
73	60	21-3/4	—	19-7/8
—	66	24	—	—
—	73	26-1/2	—	—

Figure 3

BEARINGS

Storage

Any stored bearing can be damaged by condensation caused by temperature variations. Therefore, **nyb** fan bearings are filled with grease at the factory to exclude air and moisture. Such protection is adequate for shipment and subsequent immediate installation.

For long term or outdoor storage, mounted bearings should be regreased and wrapped with plastic for protection. **Rotate the fan wheel by hand at least every two weeks to redistribute grease on internal bearing parts.** Each month the bearings should be purged with new grease to remove condensation, since even a filled bearing can accumulate moisture. Use caution when purging, as excessive pressure can damage the seals. Rotate the shaft while slowly adding grease.

Operation

Check setscrew torque before startup (see table for correct values). Since bearings are completely filled with grease at the factory, they may run at an elevated temperature during initial operation. Surface temperatures may reach 180°F. and grease may bleed from the bearing seals. This is normal and no attempt should be made to replace lost grease. Bearing surface temperatures will decrease when the internal grease quantity reaches a normal operating level. Relubrication should follow the recommended schedule.

Lubrication

Use the table for relubrication scheduling according to operating speed and shaft diameter. Bearings should be lubricated with a good quality lithium-based grease conforming to NLGI Grade 2 consistency. Examples are:

Mobil	—	Mobilith 22
Texaco	—	Premium RB
Standard Oil	—	Amolith #2
Gulf Oil	—	Gulf Crown #2
Shell	—	Alvania #2

Do not use "high temperature" greases, as many are not formulated for the high speeds associated with fan bearings.

Add grease to the bearing while running the fan or rotating the shaft by hand. Be sure all guards are in place if lubrication is performed while the fan is operating. Add just enough grease to cause a slight purging at the seals. Do not overlubricate.

Split pillowblock bearings (Link-Belt P-LB6800 & P-LB6900, SKF SAF 22500, Dodge SAF-XT) should be cleaned and repacked at approximately every eighth lubrication interval. This requires removal of the bearing cap. Clean out old grease and repack the bearing with fresh grease. Pack the bearing fully and fill the housing reservoir to the bottom of the shaft on both sides of the bearing. Replace the bearing cap, being careful not to mix caps as they are not interchangeable from one bearing to another.

BEARING LUBRICATION INTERVAL (MONTHS)

RPM SHAFT DIAMETER	1-500	501-1000	1001-1500	1501-2000	2001-2500	2501-3000	3001-3500	3501-4000	4001-4500	4501-5000
5/8 thru 1	6	6	5-6	5-6	4-6	4-6	3-4	3-4	2	2
1 3/16 & 1 7/16	6 6	6 4	5-6 4	4-6 2	4-6 2	3-5 1	2-4 1	2-4 1	1-2 1	1 1/2
1 11/16 & 1 15/16	6 6	6 4	4-6 2	4-6 1	2-4 1	2-4 1	2 1/2	1-2 1/2	1-2	1
2 3/16	6 6	5-6 4	4-6 2	3-4 1	2-4 1	1-2 1/2	1-2	1-2		
2 7/16	6 4	4-6 2	4-6 1	3-4 1	2 1/2	1-2 1/2	1-2	1		
2 11/16 & 2 15/16	5-6 4	4-6 2	2-4 1	2 1	1-2 1/2	1				
3 3/16	6	6	4	2	2					
3 7/16 thru 4 3/16	4-6 4	3-5 2	2-4 1	1-2 1/2	1					
4 7/16	4-6 4	3-4 1	2 1/2	1						
4 15/16	4-6 2	3-4 1	2 1/2							
5 7/16	6	4	2							
6	6	4								

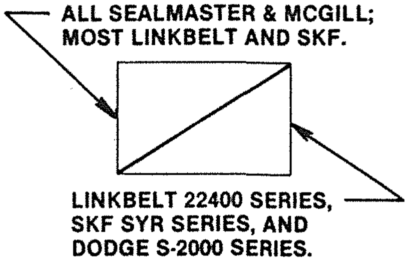

ALL SEALMASTER & MCGILL; MOST LINKBELT AND SKF.
LINKBELT 22400 SERIES, SKF SYR SERIES, AND DODGE S-2000 SERIES.

Table 4

NOTE:

1. These are general recommendations only; specific manufacturer's recommendations may vary slightly.
2. Assumes clean environment, 0°F. to 120°F.
   a. Consult The New York Blower Company for operation below 0°F. ambient.
   b. Ambient temperatures greater than 120°F. will shorten bearing life.
   c. Under extremely dirty conditions, lubricate more frequently.

COMMON FAN PROBLEMS

Excessive Vibration

A common complaint regarding industrial fans is "excessive vibration." **nyb** is careful to ensure that each fan is precisely balanced prior to shipment; however, there are many other causes of fan vibration including:

1. Loose mounting bolts, setscrews, bearings or couplings.
2. Misalignment or excessive wear of couplings or bearings.
3. Misaligned or unbalanced motor.
4. Bent shaft due to mishandling or material impact.
5. Accumulation of foreign material on the wheel.
6. Excessive wear or erosion of the wheel.
7. Excessive system pressure or restriction of airflow due to closed dampers.
8. Inadequate structural support, mounting procedures or materials.
9. Externally transmitted vibration.

Inadequate Performance

1. Incorrect testing procedures or calculations.
2. Fan running too slowly.
3. Fan wheel rotating in wrong direction or installed backwards on shaft.
4. Wheel not properly centered relative to inlet cone.
5. Damaged or incorrectly installed cut off sheet or diverter.
6. Poor system design, closed dampers, air leaks, clogged filters or coils.
7. Obstructions or sharp elbows near inlets.
8. Sharp deflection of airstream at fan outlet.

Excessive Noise

1. Fan operating near "stall" due to incorrect system design or installation.
2. Vibration originating elsewhere in the system.
3. System resonance or pulsation.
4. Improper location or orientation of fan intake and discharge.
5. Inadequate or faulty design of supporting structures.
6. Nearby sound reflecting surfaces.
7. Loose accessories or components.
8. Loose drive belts.
9. Worn bearings.

Premature Component Failure

1. Prolonged or major vibration.
2. Inadequate or improper maintenance.
3. Abrasive or corrosive elements in the airstream or surrounding environment.
4. Misalignment or physical damage to rotating components or bearings.
5. Bearing failure from incorrect or contaminated lubricant or grounding through the bearings while arc welding.
6. Excessive fan speed.
7. Extreme ambient or airstream temperatures.
8. Improper belt tension.
9. Improper tightening of wheel setscrews.

REPLACEMENT PARTS

It is recommended that only factory-supplied replacement parts be used. **nyb** fan parts are built to be fully compatible with the original fan, using specific alloys and tolerances. These parts carry a standard **nyb** warranty.

When ordering replacement parts, specify the part name, **nyb** shop and control number, fan size, type, rotation (viewed from drive end), arrangement and bearing size or bore. Most of this information is on the metal nameplate attached to the fan base.

Example: Part required: Wheel
Shop/control number: B-10106-100
Fan description: 33" PLR
Clockwise rotation
Arrangement: 1
Bearing: Link-Belt P335, 2-3/16 Bore

Suggested replacement parts include:

Wheel
Shaft
Bearings
Shaft Seal
Inlet Cone

Component parts: Damper
Motor
Coupling
Sheaves
V-Belts

LIMITED PRODUCT WARRANTY

All products are warranted by **nyb** to be free from defects in materials and workmanship for a period of one (1) year after shipment from its plant, provided buyer demonstrates to satisfaction of **nyb** that the product was properly installed and maintained in accordance with **nyb**'s instructions and recommendations and that it was used under normal operating conditions.

This warranty is limited to the replacing and/or repairing by **nyb** of any part of parts which have been returned to **nyb** with **nyb**'s written authorization and which in **nyb**'s opinion are defective. Parts not manufactured by **nyb** but installed by **nyb** in equipment sold to the buyer shall carry the original manufacturer warranty only. All transportation charges and any and all sales and use taxes, duties, imports or excises for such part or parts shall be paid for by the buyer. **nyb** shall have the sole right to determine whether defective parts shall be repaired or replaced.

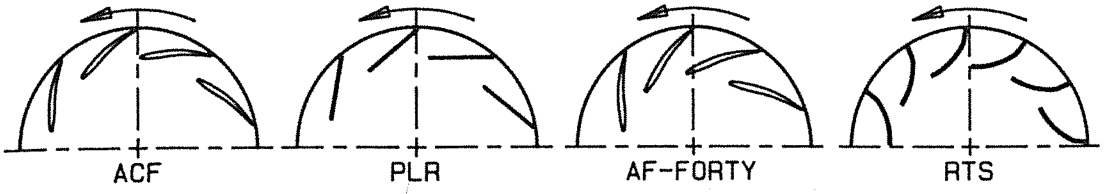
This warranty does not cover any customer labor charges for replacement of parts, adjustments or repairs, or any other work unless such charges shall be assumed or authorized in advance, in writing, by **nyb**.

This warranty does not cover any product which, in the judgement of **nyb**, has been subject to misuse or neglect, or which has been repaired or altered outside **nyb**'s plant in any way which may have impaired its safety, operation or efficiency, or any product which has been subject to accident.

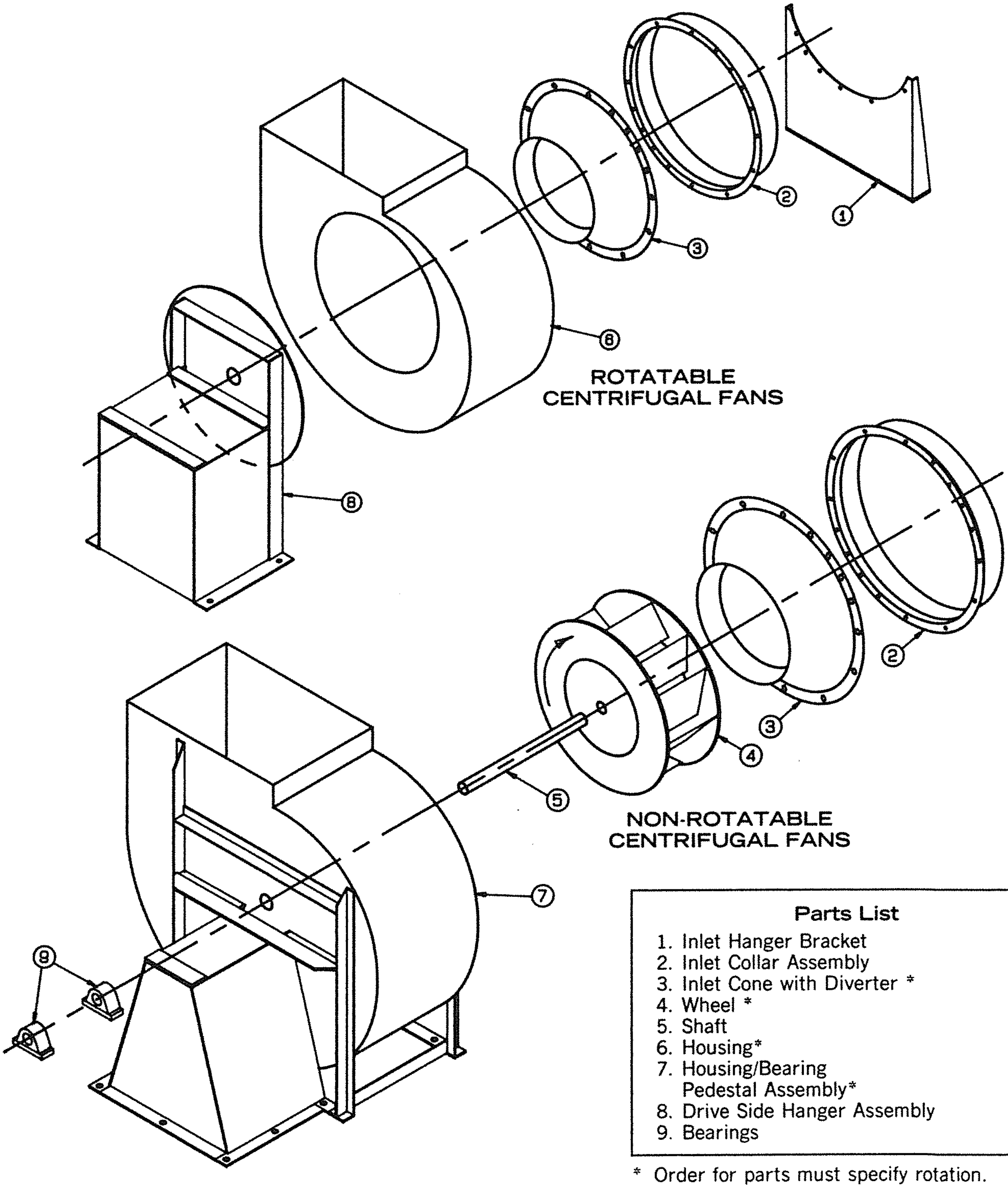
This warranty shall be null and void if any part not manufactured or supplied by **nyb** for use in any of its products shall have been substituted and used in place of a part manufactured or supplied by **nyb** for such use.

There are no warranties, other than those appearing on the acknowledgement form **INCLUDING NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE**, given in connection with the sale of the goods sold hereunder. The buyer agrees that his sole and exclusive remedy, and the limit of **nyb**'s liability for loss from any cause whatsoever, shall be the purchase price of the goods sold hereunder for which a claim is made.

SPECIFY ROTATION AS VIEWED FROM DRIVE SIDE



ARROW INDICATES COUNTER CLOCKWISE ROTATION



Parts List

1. Inlet Hanger Bracket
2. Inlet Collar Assembly
3. Inlet Cone with Diverter *
4. Wheel *
5. Shaft
6. Housing*
7. Housing/Bearing Pedestal Assembly*
8. Drive Side Hanger Assembly
9. Bearings

* Order for parts must specify rotation.

ITC MICROCOMPONENTS INC.

814 West 15th Street, Unit #206

North Vancouver, B.C. V7P 1M6, Canada

Phone: 1 (604) 985 6461 Fax: 1 (604) 985 6417

BBS: 1 (604) 985 1380, CompuServe:73244,2540 Internet:73244.2540@compuserve.com

ITC232-A User Guide

READ THIS CAREFULLY:

ITC MICROCOMPONENTS INC. does not assume any liability arising from the application and/or use of the product/s described herein, nor does it convey any license under its patent rights. ITC MICROCOMPONENTS INC. products are not authorized for use as components in medical, life support or military devices without written permission from ITC MICROCOMPONENTS INC.

The material enclosed in this package may not be copied, reproduced or imitated in any way, shape or form without the written consent of ITC MICROCOMPONENTS INC. This limitation also applies to the firmware that the Integrated Circuit in this package might contain.

INTRODUCTION:

The ITC232-A is an integrated circuit containing a general purpose serial/parallel interface as well as a number of embedded functions which find application in data acquisition and control systems. It provides easy access, from a terminal or computer serial port (EIA RS-232C), to 32 input/output lines arranged in 5 ports which can be read or written with extremely simple ASCII commands. This allows control from within a custom written program as well as from any commercial communication software package (terminal for Windows®, Procomm®, Telix® , MAC240® , etc). The possibility of operating the ITC232-A from both a dumb terminal program or a custom program is very convenient for system debugging.

The ITC232-A is hardware independent; it will work with any terminal or computer with an RS-232 serial port. The link requires 3 wires and operates at any standard speed between 300 and 115200 Bauds. The only external components required by the ITC232-A are a voltage driver to handle the RS-232 voltages and a 3.6864 MHz crystal.

In addition to serial/parallel/serial interfacing, the ITC232-A displays several other functions. These are:

(1) To send or receive data in either Decimal, Hexadecimal or Binary format.
(2) High and Low multiple interrupt recognition via 2 pins; IRQH and IRQL.
(3) 10-10000 Hz, 0-100% duty cycle Pulse Width Modulation.
(4) Hardware and software selectable Baud rate over the 300-115200 range.
(5) Four channels for reading relative resistance or capacitance.
(6) Three stepper motor ports with all the necessary logic to instrument monophasic, biphasic and half-step stepping modes. Emphasis has been placed in making the stepper interface easy to use.
(7) A 1 keystroke "Again" command to repeat the previous command.
(8) On-screen help (a summary of all commands is sent to the terminal).
(9) A configuration report feature which sends the active configuration of any parallel port, the PWM or the stepper motor ports back to the terminal.
(10) Remote operation via a phone link using a modem chip in place of the RS-232 voltage driver.

The flexibility, ease of use and multiple functions of the ITC232-A, make it ideal for any circuitry requiring computer control and/or data acquisition. Robotics, environmental control and instrumentation are just a few examples of fields where the ITC232-A finds application.

GENERAL CONSIDERATIONS:

Throughout this manual High is used as synonymous of binary 1 and Low as synonymous of binary 0. The term "terminal" includes computers and the term "peripheral" is used as any circuit containing the ITC232-A.

GETTING STARTED:

Figure 1 shows the pin out diagram of the ITC232-A. Figure 2 shows an example of the basic wiring to connect an ITC232-A to a terminal. A MAX232 is used to drive the serial port but it can be substituted with lower cost drivers (1488 & 1489) should the required voltages be available.
After setting-up the terminal's Baud rate to match the one selected with the BAUD pin (High = 9600, Low = 300), apply power to the circuit. The ITC232-A should greet you with: **Welcome to the ITC232-A**
? or h for help

> and a short beep resulting from printing ASCII(#7).

Communication Parameters:

Choose 300 or 9600 Bauds on the terminal according to the BAUD pin level (High = 9600, Low = 300). The other parameters are **always** N,8,1 (no parity, 8 bits and 1 stop bit). When using a commercial communications software package select no CR translation to CR/LF (Carriage Return/Line Feed). LF is ignored by the ITC232-A but it still takes some time to be sent.
Important: Make sure the Backspace key on your terminal sends out ASCII(#8), otherwise you will not be able to correct a command without retyping it entirely (in some terminal emulator programs, such as the one included in Windows®, a backspace is generated with a Ctrl key (e.g. under VT-100 terminal emulation, backspace is Ctrl-H)).

While with commercial software packages the **CRAB** (Bin), **CRAD** (Dec) or **CRAH** (Hex) configurations work best, the **CRAP** (Program) configuration is optimized for controlling the ITC232-A from within your own programs (see under **RESULT CONFIGURATION**).

COMMANDS AND RESULTS:

General:
Commands are **ALWAYS** sent in ASCII format and they must be followed by a carriage return (CR or ASCII(#13)). The ITC232-A accepts Upper or lower case and spaces or punctuation marks may be added for clarity (with the exception of the semicolon <;> which is reserved as a separator for command parameters).
A successful command is **ALWAYS** acknowledged with the **OK** string. The last character sent back to the terminal is **ALWAYS** the > prompt (this is useful from within a custom program to determine when the ITC232-A is done with a command). Interrupts are an exception; they are signalled by only an **L** (IRQL) or an **H** (IRQH).
An erroneous command returns **?n** where ? indicates an error and **n** is the error code (from $1 to $B, see **ERROR LIST**). The error code number is followed by an error message unless the **CRAP** configuration is in use.

There are 2 types of commands: procedures and functions. A procedure command executes an action. A function command may or may not execute an action but it always returns a result. A successful procedure type returns **OK** (a CR and a LF is inserted before the **OK** and before the > unless the **CRAP** configuration is in use. A successful function command returns **OK{$}value** where **{}** means optional and **$** applies only to results in Hexadecimal (this allows to input the hexadecimal result directly into a numeric variable in some programming languages). When **Value** is one byte long, results can be requested in either Decimal (3 digits), Hexadecimal (2 digits preceded by **$**) or Binary (8 digits in two 4 bit groups (nibbles) separated by a space). Functions returning numbers that might be higher than 255 are **ALWAYS** returned in Decimal format (see STEPPER MOTOR commands and RESISTANCE commands).
At power-up the default configuration for results is **CRAD** (Decimal). The default format can be changed at any time with the CONFIGURE RESULTS command. To override the default format for one reading, add **B** or **%** for Binary, **D** for Decimal or **H** or **$** for Hexadecimal **AFTER** the command (e.g. PRA will return the value in Port A in the default format, e.g. Decimal. PRAB will return it in ASCII binary, but only this once; next time PRA will return the result back in decimal).
When issuing a 1 byte number in a command, any of the three formats can be used. The default is Decimal (you can also place a **D** before the number). To send 1 byte Hexadecimal numbers to the ITC232-A, precede the number with an **H** or a **$** and for Binary numbers, precede the value with **B** or **%**. If you are issuing commands in Decimal you can use single, double or triple digit numbers (1 = 01 = 001). However, in Hexadecimal, two digits **MUST** be entered ($F is incorrect, must be $0F). In binary, all 8 bits **MUST** be specified (B111 is incorrect, must be B00000111). Any number that might require more than 1 byte (e.g. the speed of a stepper motor) **MUST** be issued in Decimal.

Commands are arranged by item; commands starting with **P** are port commands, **S** are stepper motor commands and so on.
Entering > or **Esc** (ASCII(#27) at any point of a command erases the command. A new ">" prompt is then issued.

LIST OF COMMANDS (in alphabetical order):

Characters within <> are mandatory, the rest of the command word is included for explanatory purposes. Items within {} are optional.

Again command:
<@>gain. Procedure or Function type. The @ character echoes the last command to the terminal preceded by @ (unless the **CRAP** configuration is active) and repeats its execution. If no previous command were issued @ has no effect.

Baud rate:
<B>aud rate <300>, <600>, <1200>, <2400>, <4800>, <9600>, <19200>, <38400>, <57600> or **<115200>** Procedure type.
Selects the Baud rate regardless of the Baud pin level. The command is acknowledged **before** the Baud rate changes. Make sure that your program adequately handles the few scrambled characters resulting from this action.
Using the **CRAP** configuration at 115200 Bauds, ~1000 analog conversions per second from an MC145041 connected to the ITC232-A SPI or ~1250 parallel port readings per second can be achieved (using a compiled program and the @ command). Beware though that writing software to operate at high Baud rates is tricky and transmission errors are more frequent at higher Baud rates when long cables are used. Listing 1 is an example in

QBASIC® that works as an elementary communications program.

Configuring the default format for results:
The ITC232-A can return results in Decimal, Binary and Hexadecimal format. The default, upon reset or power-up, is Decimal. To change it (at any time) use:
<C>onfigure <R>esults <A>SCII <B>inary or **<D>ecimal** or **<H>exadecimal** or **<P>rogram.** Procedure type.
The **CRAP** configuration is optimized for operating the ITC232-A from a user written program and it differs from the other configurations in the following:

- No CR or LF are inserted.
- The format default for results is Decimal (it can be overridden adding **B** or **%**, **D** and **H** or **$** after a command).
- The following are disabled:
  - ¤ Help.
  - ¤ Error messages (only **?n** (n = error number) is sent to the terminal).
  - ¤ PCp? (where p = A, B, C or S).
  - ¤ S? (stepper motor configuration).
  - ¤ Actual frequency returned to the terminal when issuing a PWM command.

An attempt to obtain information not available with **CRAP** enabled will result in error **?3**.

Help (on screen):
<H>elp or **<?>**. Function type.
Returns a summary of all the commands. To scroll the screen press any key. To leave Help press **>** or **Esc** and you will be returned to the **>** prompt for a new command. Mind that some communication packages (particularly older ones) have a small serial port buffer and send out XOFF to tell the peripheral to stop sending data until the buffer is cleared and then XON to continue. The ITC232-A does not recognize the XOFF-XON protocol and therefore you might lose some characters. Should this be the case, use a slower Baud rate or another communications package. On screen help is not available when the **CRAP** configuration is active.

Interrupts:
Interrupts are not commands in the strict sense of the word. They are characters sent from the ITC232-A to the terminal to signal an event occurring in the peripheral even while the ITC232-A is engaged in other tasks such as stepping a motor or generating a PWM signal. There are 2 interrupt pins: **IRQL** which detects a High to Low transition and **IRQH** which detects a Low to High transition. Note that the interrupts are only asserted when an **EDGE** is detected; the pin level is irrelevant. **Always** tie both IRQ pins to the supply voltage opposite to the one they detect (directly if the IQ is not used or via a 10K resistor). Detecting edges rather than levels allows multiple interrupts from different devices. For multiple interrupt access, pull the IQ pin to the voltage opposite to what it detects via a 10K resistor and bring the interrupt edge to the IQ pin via a 0.1 uF capacitor in parallel with a 100K resistor. The interrupt pins are particularly useful in robotics and mechanical control where an end-of-excursion must be detected in order to stop the corresponding motor.

Asserting the IRQL interrupt sends an **L** to the terminal.
Asserting the IRQH sends an **H**.
No **OK**, CR, LF or the **>** prompt are sent.

Interrupt priority has been set as follows: (1)=IRQH (2)=IRQL, (3)= Stepper motor timing, (4)=PWM. Simultaneously asserting IRQL and IRQH results in only the latter being acknowledged. If the PWM pin is pulsing very close to the allowed frequency/percentage of duty cycle combination, an IRQL or IRQH might occasionally introduce a glitch in the PWM signal.

PORTS:

The ITC232-A has 6 ports. One is the RS232 port which links it to the terminal via pins 232TX and 232RX (Figure 1). The other 5 ports can be used in your circuit and they are: PA,PB,PC,PD and PS.
PA,PB and PC are general purpose I/O 8 bit ports. These 24 pins can be individually configured as inputs (high impedance) or outputs. In the latter case, pins can be individually addressed since even though the entire port value must be written to a port, the pins remaining in their former state will not change levels at any time.
PD is always a 4 bit input port which shares pins with PS, a synchronous Serial Peripheral Interface (SPI) that can be used to communicate with other chips such as a parallel in/serial out shift register. In Figure 1, PD and PS pins are labelled PDx/PSx.
PD is always available, PS **must** be configured before being used (while all other ports can be read without being configured first, an attempt to access PS before configuring it will result in error **?2 Port must be configured or enabled first**). When a read or write command is issued to PS, PD yields its pins to the SPI, the transaction takes place using the previously input configuration for PS and then the pins are returned to PD. While it is possible to use both PD and PS in an application, this is not recommended.

Upon start-up or reset, all PA,PB and PC pins are configured as inputs (high impedance), PD is always inputs and PS is not configured at all.

Port Commands:

There are 3 types of port commands; to configure a port, to read a port and to write to a port. They all start with a **P**.

Port configure commands:

Parallel ports PA,PB and PC:

%

<P>ort <C>onfigure <A> or **<B>** or **<C> <Value>**. Procedure type.
Each pin can be individually configured as an input or an output according to the corresponding bit in **<Value>** (0 for input, 1 for output). Preceding **<Value>** with an **H** or **$** for Hexadecimal, **B** or $ for Binary and **D** for Decimal overrides the default. For example, to configure the lower 4 bits of PB as inputs and the higher 4 bits as outputs the following commands can be used: **PCB $F0**, **PCB B1111 0000**, **PCB 240** or **PCB D240** (spaces are allowed but not mandatory).

Trying to configure PD will result in error **?A Port D is always a 4 bit input port**.

<P>ort <C>onfiguration <A> or **<B>** or **<C>** or **<S> <?> {B,%,D,H or $}**. Function type. Returns the port configuration (not available when **CRAP** is active). For example, after configuring PB as above and having **CRAB** as default, **PCB?** will yield OK 1111 0000 and **PCB?D** will yield OK 240. **<S>** applies to the SPI (see below).

Serial Peripheral Interface (SPI):
In order to save pins, many IC manufacturers produce chips (such as A/D or D/A converters among others) that communicate through a synchronous serial interface. The ITC232-A SPI is designed to communicate with these chips and accommodate the various communication protocols they might require. The SPI operates as a fixed speed (57.6 KHz) circular shift register of which 8 bits are inside the ITC232-A and the other 8 (or more) bits are in a peripheral chip. Thus, in order to clock data into the ITC232-A, a value must be clocked out. When reading the SPI, the ITC232-A clocks out the last value written to the SPI or 0 if no value was previously written.

Port D, which shares pins with the SPI, is always active. When an SPI operation takes place, the corresponding configuration is loaded, the command is executed and the port returns to its normal high impedance state.
Important: (1) **ALWAYS** pull the clock pin of the peripheral chip to ground or VDD (depending on whether the clock is required to idle Low or High) via a suitable value resistor (otherwise the first clock transition might not be detected at all). If you also need to use this pin for PD, place a 0.1-1 uF capacitor between your peripheral clock pin and **PD2/PS_CK**. (2) To use PS, pin PD3/PS_VDD **MUST** be tied High.

To configure the SPI:
<P>ort <C>onfigure <S>erial <R>ead or **<W>rite** or **<A>ll <V>alue**. Procedure type.
Some serial chips need only be read. Others might only be written to and finally, some devices require to be both read and written to. Both read only and write only peripheral chips might be simultaneously connected to the SPI and they might require different communication configuration. Thus the <R>, <W> and <A> in the command (<A>ll means both read and write).

When writing out a value, the SPI is first configured as per the last **PCSW** command (if no previous configuration entered, error **?2 Port must be configured or enabled first** will occur). When reading data in, the SPI is configured as per the last **PCSR** command.

When a read operation takes place, the **<R>ead** configuration is enabled, the last value written to the SPI (or 0 if none yet) is clocked out and the value from the peripheral is clocked in and sent to the terminal. Beware that the value is written out in the **<R>ead** configuration and therefore should another peripheral to which you normally write (using a different **<W>rite** configuration) be connected to the SPI, it might interpret the data wrongly. It is recommended that you use a parallel port pin to select the device when more than one chip is connected to the SPI.

<V>alue specifies whether the SPI is enabled or not, the clock polarity, whether the clock and the data are in phase or not and the sense of the byte (some chips send data MSB first while others send the LSB first). The Table below shows the function of the different bits in **<V>alue**:

Bit 7 Must be 1 in order to enable the SPI.

Bits 6,5,4,3 are irrelevant.

Bit.2 Determines the polarity of the clock (PD2/PS_CK pin).
POL 0 = the clock idles Low.
1 = the clock idles High.

Bit.1 Controls the clock phase (PD2/PS_CK pin).
PHASE 0 = In phase with data.
1 = Out of phase with data.

Bit.0 Controls the order of the bits received by the SPI.
ORD This is useful in some cases when a peripheral sends the byte "backwards".
0 = The order is preserved.
1 = The order of the bits is reversed (MSB <--> LSB)

The following Table shows all possibilities:

Bit 7	6	5	4	3	2	1	0	Hex	Dec
MUST BE		{ Irrelevant		}	POL	PHASE	ORD		
1	0	0	0	0	0	0	0	$80	128
1	0	0	0	0	0	0	1	$81	129
1	0	0	0	0	0	1	0	$82	130

1	0	0	0	0	0	1	1	$83	131
1	0	0	0	0	1	0	0	$84	132
1	0	0	0	0	1	0	1	$85	133
1	0	0	0	0	1	1	0	$86	134
1	0	0	0	0	1	1	1	$87	135

To disable the SPI, any configuration with Bit.7 = 0 (any decimal < 128 or hex < $80) will do.
Important: Pin PD3/PS_VDD needs not be pulled High for a configuration command to be successful but it is **mandatory** to read or write to the SPI (otherwise error **?B** will be issued).

As with the other ports, **PCS?{number base}** returns **<Value>** in the specified **{number base}** or in the default format.

READING AND WRITING TO PORTS:

Reading a parallel port:
<P>ort <R>ead <A> or **<B>** or **<C>** or **<D> {B,%,D,H,$}**. Function type.
When reading a parallel port which has pins configured as outputs, the value returned in the corresponding bits is the present state of those pins (0 if nothing written to the port since power-up or reset).

Writing to a parallel port:
<P>ort <W>rite <A> or **<B>** or **<C> {B,%,D,H,$} <Value>**. Procedure type.
When writing to a pin configured as an input, the pin remains in a high impedance state. However, the value is stored in a phantom bit which might be thought of being present behind the actual port bit. Thus, if you write to an input pin and the configure it as an output pin, the value in the phantom bit will be **immediately** transferred to the corresponding pin (this can lead to unwanted values in a port pin if this is not kept in mind). After a reset or upon start-up the value in all phantom bits is 0.

Reading and writing to the SPI:
<P>ort <R>ead <S>PI {B,%,D,H,$} Function type.
<P>ort <W>rite <S>PI {B,%,D,H,$} <Value> Procedure type.
These commands work in a fashion similar to that of the parallel ports with the following exceptions:

1) Pin PD3/PS_VDD **MUST** be tied High. Otherwise, error **?B SPI requires pin PD3/PS_VDD always high, change and try again** is sent to the terminal.

2) The relationship between configuration and pin value is different.

3) There is no phantom port storing the last written value as happens with PA, PB and PC.

4) **IMPORTANT:** the SPI can be considered as a circular serial shift register in which the 8 bits in the port are circulated through a peripheral chip. In order to read a value into the SPI, a value must be written out. This value is the last written value to the SPI or 0 if no writing took place between the configuration and a reading. There are some particular situations that should be avoided or handled with care:

4.a. Connecting 2 peripheral chips to the SPI, one which is read from and the other which is written to while having them both enabled simultaneously. This will result in the re-writing of the latter when reading the former. The value written will be the same as last time, but the data on the peripheral chip will jump up and down as the byte is shifted in at a speed of 57.6 KHz. In addition to this, if the write and read configurations are different it is almost certain that the chip to be written will misinterpret the data. To prevent this, use pins from PA,PB or PC to enable or disable the chips connected to the SPI as required.

4.b. Connecting a peripheral chip that requires both writing and reading to operate, e.g. the Motorola® MC145041 Analog/Digital converter. Since interfacing the ITC232-A to this chip is a very cost effective way to read up to 11 different analog voltages without using any PA, PB or PC pins, we will expand on the interfacing details:

Interfacing an MC145041 A/D converter to the ITC232-A:
A number to select the analog channel must be written into the MC145041 while the value of the last conversion is clocked out. Thus, it is convenient to either read the SPI continuously (polling) or to read it twice in rapid succession (otherwise the returned value is historical). The MC145041 uses only the 4 most significant bits of the value written to select the analog channel. Thus to select channel AN1 the port number must be b00010000 (16), for AN2 it is b00100000 (32) AN3 = b00110000 (48), and so on. The value to select an AN channel = AN channel * 16.
Figure 3 shows how to connect the MC145041 to the ITC232-A.

REMOTE OPERATION VIA A PHONE LINK (Phone mode):

The ITC232-A can be operated remotely using a low cost modem IC such as the AD7910 or AD7911 in-lieu of the RS-232 driver. Figure 4 shows an example of the corresponding circuitry. Remote operation requires that the ITC232-A operate at 300 Bauds (by pulling the BAUDS pin Low) and that an IRQL (generated from the ring signal of a phone line as shown in Figure 4) takes place prior to any command after power-up or reset (should a command be issued before the IRQL then the ITC232-A will operate normally (remote operation is disabled)). The first IRQL does not send an **L** to the terminal. Instead, it results in PA.0 = High. This is used to answer the phone. PA.0 is thereafter functionally excluded from PA; writing or configuring this pin will have no effect. In order to limit the necessary lines to 1 (PA.0) as well as to allow for simple modem systems, there is no carrier detection. Instead, the message "**Send a command within 30 sec or the ITC232 will hang up. This message will thereafter repeat itself if no commands**

within 5 minutes. Send OFF to hang up." followed by ASCII(#7) and the > prompt is sent out approximately 7 seconds after answering the phone. The ASCII(#7) not only provides an audible signal but is also useful to recognize this particular message from within a custom program. The 7 seconds delay allows commercial communications packages to detect the presence of the carrier. Should the ITC232-A not receive a command or **Enter** within 30 seconds, it will assume the phone call was unintentional and carry out the following sequence: (1) Send out the **DISCONNECTING** ASCII(#7) > message, (2) Hang-up by making PA.0 = 0, (3) Reconfigure PA.0 as an input, (4) Wait idle for another call.

Since the phone link could be unintentionally cut-off at any time, the ITC232-A will repeat the message above every ~5 minutes without a command (thus, make sure you send e.g. PRA, @ or Enter (ASCII(#13)) every 2-3 minutes to reset the timer). Should no command be received 30 seconds after the warning message, the ITC232-A will hang up and sit idle waiting for the next call. Operating a stepping motor will disable the 5 minute timer in order to prevent being disconnected during a stepping sequence. The timer is reset after the last step thus you have 5 minutes to issue a command before the warning message is issued again.

The **<OFF>** command allows the remote user to hang-up the phone connected to the ITC232-A (this command is not allowed unless the conditions for remote operation have been fulfilled). Whenever the ITC232-A receives **<OFF>**, the **DISCONNECTING** ASCII(#7) > message is sent to the terminal and the PA.0 bit is pulled Low. The various configurations and port values are retained until a reset or a power-down occurs but not the Baud rate which will be reset to 300 Bauds whenever the ITC232-A answers the phone again. Note that a **<RESET>** sent from the terminal will not only reset the chip but also hang up the phone.

You can simulate remote operation while being connected to the ITC232-A directly from a terminal by using a LED to monitor PA.0 and generating an IRQL following a reset.

PULSE WIDTH MODULATION (PWM):

PWM can be used in a number of applications such as generating an analog voltage or regulating the speed of a D.C. motor (speeds down to a few turns per minute can thus be achieved with minimal torque reduction). The possibility to generate a given frequency signal is also advantageous. Besides the obvious creation of musical tones, this feature allows the production of accurate time intervals.

The ITC232-A features a **PWM** pin. The frequency and duty cycle of the pulses appearing on this pin can be specified from the terminal. The frequency range is 10-10,000 Hz and the duty cycle can be set from 0 to 100% in 1 % intervals. However, as the frequency increases it becomes progressively more difficult to measure small time intervals accurately. Thus, the full duty cycle range is only available up to 220 Hz for 1% duty cycle and 230 Hz for 99% duty cycle. Increasing the frequency compresses the available duty cycle range around 50%.

The PWM pin can be also set High or Low. At power-up or reset the PWM pin is set Low.

PWM COMMANDS:

<W>idth <frequency>. Function type.

<frequency> can be any value between 10 and 10,000 Hz and it **MUST** be specified in decimal format. This command assumes a 50% duty cycle is required. The actual frequency is sent to the terminal (see note #1 below).

<W>idth <frequency> <;> <Duty cycle>. Function type.

<Frequency> is the same as above, the <;> is mandatory and **<Duty cycle>** is an integer from 0 to 100 indicating the percentage of the cycle during which the PWM pin is High.

Notes:

1) Upon entering these commands the ITC232-A returns **f=XXXXX** (not available in the **CRAP** configuration). XXXXX is **always** a 5 digit long decimal number and is the **integer** part of the actual frequency the ITC232-A is putting out (due to calculation rounding and timing restrictions).

The actual frequency is given by the expression: Af = 460800 ÷ Round(460800 ÷ Rf) where Af is the actual frequency and Rf is the requested frequency. Af should be within the precision of the crystal.

2) During the stepping of a stepper motor: (a) If **WL** or **WH** are active then the pin stays as it is. (b) If PWM is pulsing, the PWM pin is brought Low during stepping.

3) If **<Duty cycle>** = 0 or 100, then **<frequency>** is irrelevant, the PWM pin will stay Low or High respectively. Another way to achieve this is by using the <WL> and <WH> commands.

4) **<Duty cycle>** can be varied in 1 % intervals, however, as the frequency increases it becomes progressively more difficult to generate very small or very large duty cycles. If a particular duty cycle cannot be achieved for a given frequency, the **?8 Frequency too high for requested duty cycle** error will occur. The highest frequencies require duty cycles around 50% whereas below 220 Hz, 1% or 99% duty cycles are possible.

<W>idth <L>. Procedure type. Forces the PWM pin Low. This command is internally issued at power-up or reset.

<W>idth <H>. Procedure type. Forces the PWM pin High.

<W>idth <?>. Function type.

Returns the last <W> command issued (the requested frequency, **not** the actual frequency). Not available under the **CRAP** configuration. Issued immediately after a reset or start-up, the **W?** command returns **WL**.

RESETTING The ITC232-A

<RESET> Procedure type.

This command is self explanatory. It has the same effect as a hardware reset.

MEASURING A RESISTANCE OR A CAPACITANCE WITH THE ITC232-A:

<R>esistance <0> or <1> or <2> or <3>. Function type.
0-3 represent the bits or pins on PC.
The input impedance of port C is extremely high. Thus, if an RC network is connected to an input pin as shown in Figure 5, then the time constant of the network can be determined by measuring the time to reach a Low to High transition. Since for a given pin the transition point is constant if VDD is kept constant, relative capacitances and resistances can be read. Values from one pin cannot be compared to those from another pin because the transition points are not identical. The sequence used by the ITC232-A to carry out a reading after receiving **Rn** is as follows:
1) The selected port is turned into a Low output for a short time in order to discharge capacitor C.
2) The pin is turned into an input. A loop linked to the internal clock measures the time required for the pin to become a logic High.
3) The result is sent to the terminal.

Note that the value returned is linearly proportional to the time to charge C to a certain voltage and hence there is a linear relationship with the values of C or R. Due to minor differences in the electrical characteristics of the input pins, readings are designed to be relative to a capacitance or resistance standard used to calibrate a given pin. Use high quality, low leakage capacitors. Polyethylene caps work best. Avoid electrolytic capacitors.

The reading is linear (for both capacitance and resistance) between 10 and 32,767 relative units, with an error, as small as 0.5 %, depending on the capacitor used. Should the time to read the R network be excessive (a result > 32,767), Error **?7 (Time out error)** will be sent to the terminal. The range of resistance that can be measured goes from 200 Ohms to 10 Megohms using different capacitors for the desired range. We recommend against measuring resistances below 500 Ohms. For one, readings are less accurate than above 500 Ohms. Secondly, the chip might be permanently damaged due to excessive current flowing through the pin when it goes Low to discharge the capacitor.
Should PWM be enabled, expect the reading error to increase at frequencies over 5 KHz or when a combination of frequency and duty cycle is close to the admissible limits since PWM interrupts have priority over the timer used for the **R** command.
The port needs not be configured as an input in order to obtain a value (the **<R>esistance** command takes care of it). However, keep in mind that:
(1) Changing the pin configuration and writing to it, prior to reading a resistance, might introduce differences as large as 10% in the readings (depending on the type of capacitor and its value (a Siemens® 0.47 uF Poly capacitor yields a difference below 1%)). Thus, we strongly recommend to keep the pin configuration and value unchanged between readings.
(2) A circuit connected to the port might interfere with the reading obtained unless its impedance is very high compared to the resistance being measured.
(3) Changing the PWM frequency over a wide range between resistance readings might lead to differences in readings which get larger as the returned value gets lower.
Measuring resistances can in many instances replace an A/D converter and it finds application in many different situations. Some examples are robots, in which the absolute position of a mechanical element can be followed-up by measuring a variable resistor attached to it, in measuring light intensity (with a Cadmium sulphide cell) and even for measuring the conductance of a solution. For the latter, use the PWM pin at ~1000 Hz to control a 4066 analog switch which alternates the electrodes (representing here the resistance in the network) in order to avoid electrode polarization.

STEPPER MOTOR INTERFACING:

Stepper motor basics:

A stepper motor is designed in such a way that it turns one step at a time. This allows precise motion and absolute positioning of the rotor. Figure 6A shows a simplified version of a stepper motor. The rotor consists of a permanent magnet surrounded by the stator poles (electromagnets, also referred to as phases). The magnetic polarity of the stator can be changed according to the current flowing through its windings. The motor in the example can be made to turn with three possible drive sequences: The first is to energize the windings in the sequence AB/CD/BA/DC (BA is the same winding as AB but the current flows in the opposite direction) as shown in Figure 6B. This sequence is known as "one-phase-on" or "monophasic" because at any one time only 1 winding (phase) is energized. The second possibility is to always energize both windings. This is called "two-phase-on" or "biphasic" and it is represented in Figure 6C. Note that now the rotor aligns itself between two pole positions rather than with the poles themselves. This configuration, which is the one most commonly used, yields the highest torque. Finally, the third option is to energize one phase, then two, then one, etc. as shown in Figure 6D. This sequence halves the angle per step and therefore is called "half step mode". All three modes are implemented in the ITC232-A as we will see later. Please note that the very first step can result in (a) no motion if the rotor happens to be aligned with the stator, (b) a clockwise motion (as wanted) if the rotor is behind the stator or (c) a counter clockwise motion if the rotor is ahead of the stator. This is only valid for the first step of a sequence but it should be kept in mind when designing a system.
The angle per step is 90° (Figures 6A,B & C), or 45° (Figure 6D). Real motors have many more poles on the stator and rotor thus allowing for much smaller angles per step. Typical values are 15° (24 steps/turn), 7.5° (48 steps/turn), 3.75° (96 steps/turn), 3.6° (100 steps/turn) and 1.8° (200 steps/turn).

Four bits of a port can be used to drive a stepper motor. For example, if we connected the A,B,C and D winding ends to a suitable current driver which buffers the output from a 4 bit binary word (nibble) created by the ITC232-A, the connection would be as follows:

Winding	C	D	B	A
Bit	3	2	1	0

In monophasic mode (Figure 6B), the driving sequence would then be 0001, 1000, 0010, 0100 (to spin in one direction and backwards to spin in the opposite direction). In biphasic mode (Figure 6C), the sequence is 1001, 1010, 0110, 0101. Interleaving the former we have the half step mode (Figure 6D) with the driving sequence 0001, 1001, 1000, 1010, 0010, 0110, 0100, 0101.
Once the last value is output, the sequence is repeated. Stepping can start on any value provided that the sequence order is maintained.
Inverting the voltage on a winding requires 4 transistor H bridges such as the ones shown in Figure 7. Even though the drivers can be made using 8 discrete transistors per motor, a much easier, cheaper and efficient way, is to use a driver IC such as the L298 from SGS®. Another way to approach the driver problem is to use a so called unipolar or bifilar winding motor. This kind of stepper normally has 5 or 6 cables coming out instead of 4. The coils in bifilar motors are double; one is wound clockwise and the other counter-clockwise. To invert the magnetic field, power is sent to one or the other coil rather than changing the current flow through the same coil. The circuit to drive this kind of stepper only requires 4 transistors (Figure 8). This figure also illustrates why only 5 or 6 cables come out the motor instead of 8. Two disadvantages of bifilar steppers compared to monofilar ones are higher cost and less power (both due to an extra set of windings which are only used half of the time).

The driving sequence for both kinds of steppers is identical.

Working with steppers can be frustrating particularly if one is using a stepper without having the specifications on-hand. However, using the ITC232-A in conjunction with an L298 from SGS® or equivalent, or a bifilar stepper with 4 Darlington NPN transistors, makes the task trivial: To use a 4 wire (or monofilar or biphasic) stepper, first, determine with the aid of an Ohm meter which cables correspond to each phase. Then connect them according to Figure 9 and issue the commands described below. Inverting one phase results in the motor turning in the opposite direction. If you are working with a 5 or 6 wire stepper first determine the phases and the common wire to each phase (the one showing half the resistance of the other two). Then, configure the circuit as shown in Figure 8. Again, reversing one phase will invert the sense of spinning.

Steppers and the ITC232-A:
The ITC232-A contains all the necessary logic to operate any stepper motor using very simple commands. The following rules apply:

1) Steppers are controlled through the upper 4 bits of PA,PB or PC.
2) The name of the motor equals the name of the port to which it is connected.
4) The steppers must be enabled (and configured) before use.
5) Enabling a stepper turns the upper 4 bits of the port into inputs (high impedance). Disabling a stepper leaves the corresponding pins as inputs with the last value written to the stepper in the corresponding phantom port bits (see under **PORTS**).
6) The configuration applies to all enabled steppers (you cannot have 2 or 3 motors configured differently).
7) If the configuration is changed, the last value written to the port is retained. Thus, the speed of a stepper may be changed without losing steps in the process.
8) While stepping the PWM is inactive. If the PWM pin was set High (**WH**) or Low (**WL**) it will remain the same. If pulsing, the PWM pin will be brought Low during stepping.

To Enable and set-up a stepper motor configuration:

<S>tepper <E>nable <A> or **<B>** or **<C> <M>onophasic** or **<B>iphasic** or **<H>alf step <Speed> <;> <Stop delay>** Procedure type.
<A>, **<B>** and **<C>** refer to the 4 upper bits of the corresponding ports. **<M>onophasic, <B>iphasic** and **<H>alf step** refer to the modes described above. **<Speed>** is in steps/second (10 to 4,000). **<Stop delay>** is a value within 0-255 decimal representing the number of steps during which the last value on the stepper port remains fixed before the port returns to a high impedance state. (E.g. **SEAB500;10** will make stepper A turn at 500 steps/s (2 ms/step) and hold the last step for 2*10 = 20 ms.) This serves the purpose of braking the motor thus preventing it from continuing turning due to inertia (The larger the motor and the higher the speed the larger this value should be. Start trying with 10% the speed value).
Once the configuration for a stepper has been entered, enabling a second or third stepper only requires **<S>tepper <E>nable <A>** or **<B>** or **<C>**. The remaining parameters are the same for all motors. The configuration entered last becomes current for all motors. The maximal speed error for a stepper is < 2.2 %.

To disable a stepper motor:

<S>tepper <D>isable <A> or **<B>** or **<C>**. Procedure type.
Note that you can disable all the steppers while the configuration is retained in memory. Thus, a subsequent **<S>tepper <E>nable <A>** or **<B>** or **<C>** will work using the same parameters as previously entered.

To request the stepper motors configuration:

<S>tepper <?> {B or % or D or H or $} or **<S>tepper <E>nable <?> {B or % or D or H or $}** Function type.
These commands return the stepping motor configuration, the enabled ports (A,B & C) and the last output for each motor. These commands are disabled when the **CRAP** configuration is active. **{B, %, D, S or H}** specify the format for the last value on the corresponding port (otherwise this value is returned in the default configuration).

You can reconfigure a port with **PC p nr** while the corresponding stepper is enabled and you can even write to the pins by issuing a **PW p nr** (**WARNING:** the last value written to the stepper port will immediately appear on a pin should it be configured as an output). The stepper port will be reconfigured when stepping again and it will be left as all inputs once the stepping stops). Re-configuring a parallel port connected to a stepper motor driver is strongly discouraged since it might lead to catastrophic output combinations that can destroy the stepper driver transistors.

To step a motor:
<S>tep <A> or **<B>** or **<C> <L>eft** or **<R>ight <Number of steps>.**
Procedure type.
<L>eft or **<R>ight** are arbitrary mnemonics to specify the direction of spinning. **<Number of steps>** can be any number up to 65535 (a value of 0 is valid and results in no action).

To stop a motor while stepping:
Sometimes you might want to force a stepper to stop before all the steps are completed. This can be done by sending: a space (ASCII(#32)), "S", "s", ">", Esc (ASCII(#27)), or Enter (ASCII(#13)). This action results in the ITC232-A sending the remaining number of steps (5 digits, **ALWAYS** in decimal, with leading 0's if required) followed by the message **steps to go** (the latter omitted if the **CRAP** configuration is active).

Using more than 3 steppers:
This can be easily done by multiplexing the stepper motor ports using a parallel port pin to select the active motor driver (if a L298 driver is used (Figure 9), connect ENA and ENB to the controlling pin). Keep in mind that the values left on the port from the last stepper will likely not be the required ones to start the next stepper. This will lead to the equivalent of losing steps. To prevent this, disengage all motors and make the required port to "step" as many steps as necessary to bring the next stepper to a complete turn. Example: Stepper 1 and 2 are connected to port C. Both have a resolution

of 3.6 degrees/step thus requiring 100 steps for a complete turn.

1) Activate stepper 1 via a port A pin and make it step 1230 steps.
2) Disengage both steppers and make the port step 100-30=70 steps in the same direction.
3) Activate stepper 2 via another port A pin and make it move 13 steps.
4) Disengage both steppers and make the port step 100-13=87 steps in the same direction.
5) Activate stepper #1 and repeat the process.

Note: Avoid changing modes (<M>, <B> and <H>) from one stepper to another.

SUMMARY OF COMMANDS:

Items within <> are mandatory. Items within {} are optional.

* Commands which are not available when **<C>onfigure <R>esults <A>SCII <P>rogram** is active.

<@>gain repeats the last command.

<B>aud rate <300>, <600>, <1200>, <2400>, <4800>, <9600>, <19200>, <38400>, <57600> and <115200>. Procedure type.

<C>onfigute <R>esults <A>SCII <B>inary or **<D>ecimal** or **<H>exadecimal** or **<P>rogram.** Procedure type.

* **<H>elp or <?>.** Function type.

Interrupts: L or **H** sent to terminal.

<OFF> Function type. Returns **DISCONNECTING** ASCII(#7) > and makes PA.0 an input (to hang up the phone).
Only available if in phone mode (BAUD pin = Low and IRQL asserted before a command is received after reset or power-up).

<P>ort <C>onfigure <A> or <B> or <C> {B,%,D,H,$} <Value>. Procedure type.
<P>ort <C>onfigure <S>erial <R>ead or <W>rite or <A>ll {B,%,D,H,$} <V>alue. Procedure type.

The following Table shows all possibilities:

Bit 7	6	5	4	3	2	1	0	Hex	Dec
MUST BE		{	Irrelevant	}	POL	PHASE	ORD		
1	0	0	0	0	0	0	0	$80	128
1	0	0	0	0	0	0	1	$81	129
1	0	0	0	0	0	1	0	$82	130
1	0	0	0	0	0	1	1	$83	131
1	0	0	0	0	1	0	0	$84	132
1	0	0	0	0	1	0	1	$85	133
1	0	0	0	0	1	1	0	$86	134
1	0	0	0	0	1	1	1	$87	135

PCS0 Disables the serial port.

* **<P>ort <C>onfiguration <A>,<B>** or **<C>** or **<S> <?> {B,%,D,H or $}**. Function type. Returns the port configuration.

<P>ort <R>ead <A> or **<B>** or **<C>** or **<D>** or **<S> {B,%,D,H,$}.** Function type. Reading PS sends previously written value out the PD1/SP_TX pin using the Read configuration.

<P>ort <W>rite <A> or **<B>** or **<C>** or **<S> {B,%,D,H,$} <Value>.**

<RESET> Equivalent to a hardware reset.

<R>esistance <0> or **<1>** or **<2>** or **<3>.** Function type.
0-3 are the bit or pins on port C.

<W>idth <L>ow. Procedure type. Forces PWM pin Low.

<W>idth <H>igh. Procedure type. Forces PWM pin High.

<W>idth <frequency>. Procedure type
<frequency> can be any value between 10 and 10,000 Hz and it **MUST** be specified in decimal format. A 50% duty cycle is assumed.

<W>idth <frequency> <;> <Duty cycle>. Procedure type.
<frequency> can be any value between 10 and 10,000 Hz and it **MUST** be specified in decimal format. **<Duty cycle>** can be any integer value from 0 to 100 %.

* <W>idth <?>. Function type.
Returns the last <W> command.

<S>tepper <E>nable <A> or <B> or <C> **<M>onophasic** or **<B>iphasic** or **<H>alf step <Speed>** <;> **<Stop delay>** Procedure type.
<Speed> is in steps/s (10-4000). **<Stop delay>** is in steps (0-255).

<S>tepper <D>isable <A> or <B> or <C>. Procedure type.

* <S>tepper <?> {B or % or D or H or $} or <S>tepper <E>nable <?> {B or % or D or H or $}. Function type. Returns the configuration, the active steppers and the last value written to each active stepper in the requested format.

<S>tep <A> or <B> or <C> <L>eft or <R>ight **<Number of steps>**
Procedure type. Makes the motor step.

To stop a motor while stepping:
Send ASCII(#32) (space bar), or an "S" or "s" or an ">", or Esc (ASCII(#27)), or Enter (ASCII(#13)). * The number of remaining steps is sent to the terminal as a 5 digit (with leading zeros if necessary) decimal number.

ERROR LIST:

?1 Syntax error.
?2 Port must be configured or enabled first.
?3 Command not allowed in current configuration.
?4 No such port.
?5 Value out of range or syntax error.
?6 Pin configured as an output.
?7 Time out error.
?8 Frequency too high for required duty cycle.
?9 Baud rate not supported.
?A Port D is always a 4 bit input port.
?B SPI requires pin PD3/PS_VDD always high, change and try again.

PIN OUT DIAGRAM AND PIN DESCRIPTION

(1) **RESET**: Bringing this pin Low results in: (a) All previous configurations are lost. (b) PA, PB & PC are all configured as inputs. (c) The Baud rate is set according to BAUD pin. (d) The CRAD configuration is set as default. (e) Pin PWM is pulled Low. (f) The message:
Welcome to the ITC232-A
? or h for help
followed by ASCII(#7), CR, LF and the ">" prompt are sent out pin 232 TX.

(2) **IRQL**: Edge sensitive only. Asserted on a High to Low transition. If BAUD pin = Low (300 Bauds) and IRQL is asserted before a command after power-up or RESET then IRQL forces the ITC232-A into Phone Mode. After the first command or if the ITC232-A is not in Phone mode, IRQL sends an **L** to the terminal (see Interrupts).

(3 & 40) **VDD**: +4.5 to +5.5 Volts referenced to VSS.

(4-11) **PA0..PA7**: Parallel port A. PA0 is excluded from PA when the Phone mode is asserted.
PA4..PA7 are used by Stepper port A.

(12-19) **PB0..PB7**: Parallel port B. **PB4..PB7**: Stepper port B.

(20) **VSS**: Lowest digital voltage connected to the ITC232-A.

(21-28) **PC0..PC7**: Parallel port C. **PC0..PC3**: Used to measure resistance or capacitance values. **PC4..PC7**: Stepper port C.

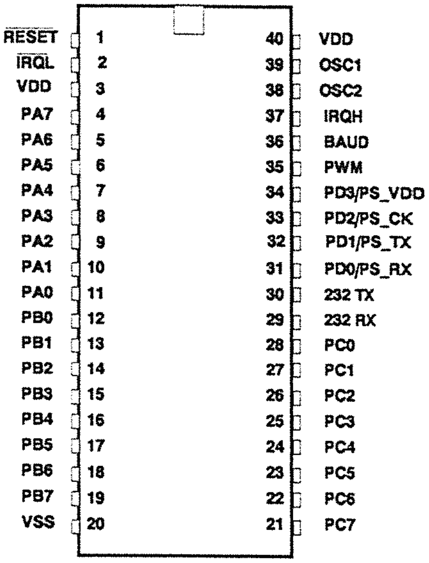


Figure 1

(29) **232 RX**: Receives RS232-C serial data **from** terminal.

(30) **232 TX**: Transmits RS232-C serial data **to** terminal.

(31) **PD0/PS_RX**: Pin common to PD (always 4 inputs) and PS (the Serial Peripheral Interface or SPI). When SPI active this pin **receives** data **from** a peripheral chip synchronized with the clock on pin PD2/PS_CK (see Serial Peripheral Interface).

(32) **PD1/PS_TX**: Pin common to PD and PS. When SPI is active this pin **sends** data **to** a peripheral chip synchronized with PD2/PS_CK.

(33) **PD2/PS_CK**: Pin common to PD and PS. When SPI is active this pin clocks data in and out PS_TX and PS_RX. The clock can be in phase or out of phase with the data and idle Low or High according to commands from the terminal. This pin **must** be tied via a suitable resistor to the clock idle voltage. Failure to do this might result in the peripheral missing the first clock pulse since PD2/PS_CK is in a high impedance state until the SPI becomes active.

(34) **PD3/PS_VDD** Pin common to PD and PS. In order to use the SPI this pin **must** be tied to VDD.

(35) **PWM**: Pulse Width Modulation output.

(36) **BAUD**: Selects the default Baud rate at reset or power-up. Low = 300 Bauds, High = 9600 Bauds. Changing the BAUD pin level after a reset is of no consequence.

(37) **IRQH**: Edge sensitive only. Asserted on a Low to High transition. Sends an **H** to the terminal.

(38) **OSC1**: To 3.6864 MHz crystal or external clock.

(39) **OSC2**: To crystal. If external clock applied to OSC1 then OSC2 should be left unconnected.

ELECTRICAL SPECIFICATIONS

Maximum Ratings (Voltages referenced to VSS)

Rating	Value	Unit
Supply Voltage	-0.3 to + 7.0	V
Input Voltage	VSS-0.3 to VDD+0.3	V
Current drain per pin	25	mA
Storage temperature range	-65 to +150	°C
Operating temperature range	0 to +70	°C

DC electrical characteristics (VDD-VSS = 5.0 VDC)

Characteristic	Min	Typ	Max	Unit
Output Voltage (I<10uA)	VDD-0.1	-	0.1	V
Output Voltage (i=0,8mA)	VDD-0.8	-	0.4	V
Input High (PA,PB,PC,PD, IRQ's,BAUD,232 RX,RESET)	0.7 x VDD	-	VDD	V
Input Low (PA,PB,PC,PD, IRQ's,BAUD,232 RX,RESET)	VSS	-	0.2 x VDD	V
Supply Current		4.7	7.0	mA
Hi-Z input leakeage current (PA,PB,PC,PD)	-	-	±10	μA
Capacitance PA,PB,PC,PD	-	-	12	pF
Capacitance RESET,IRQ's, 232 TX,232 RX, BAUD	-	-	8	pF

Notes:

1. All values show average measurements.
2. Measurements were made at 25 C.

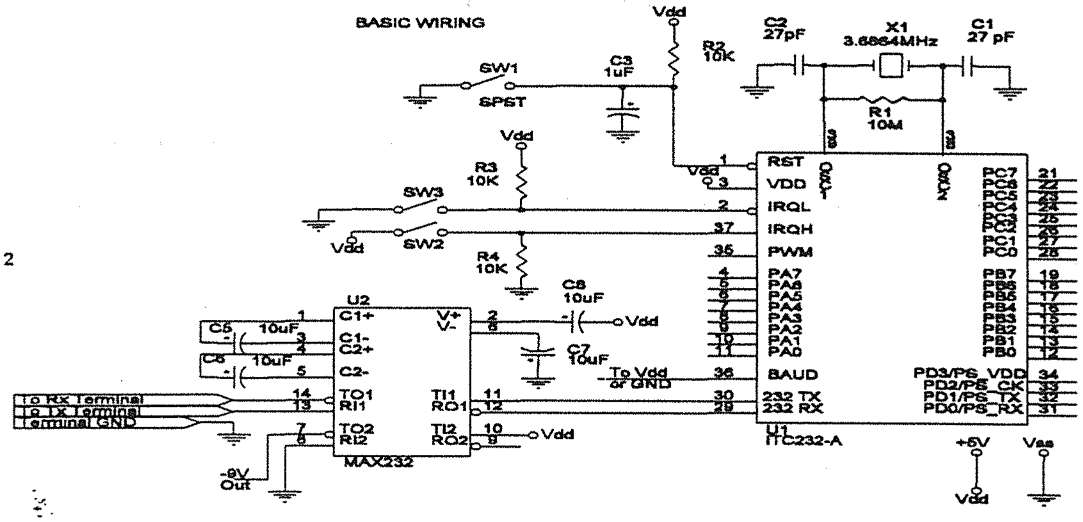
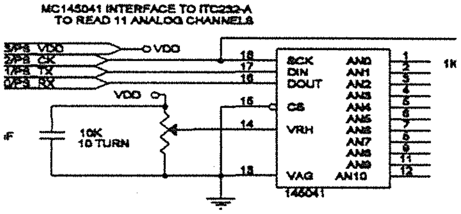


Figure 2



10 turn pot sets the reference voltage.
onfiguration: PC8A128.
lect AN channel do PW8n
ANx * 16. To read the AN channel do PR8.

Figure 3

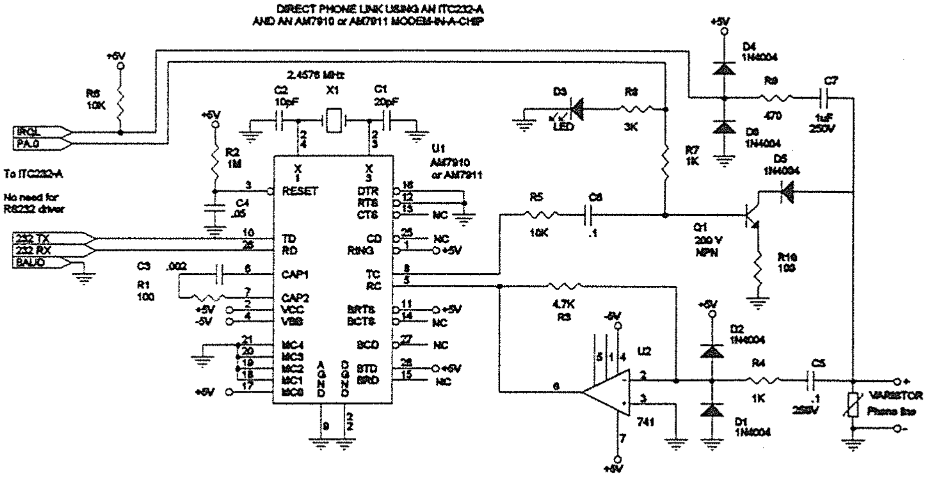


Figure 4

Figure 5

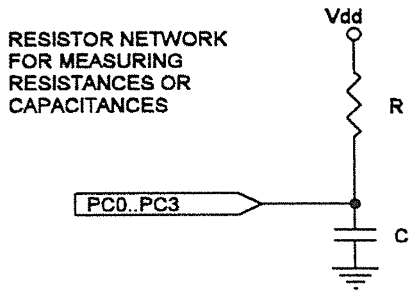


Figure 6A

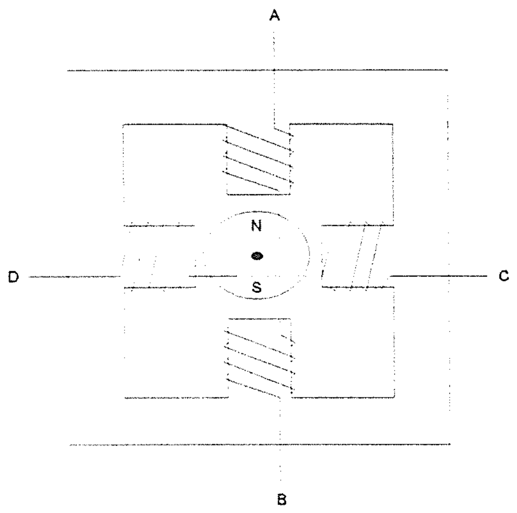


Figure 6B

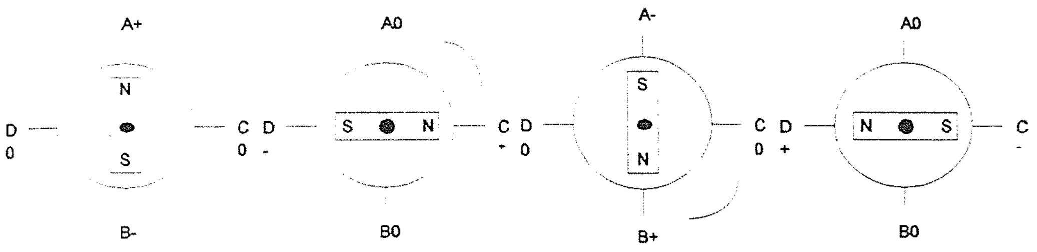
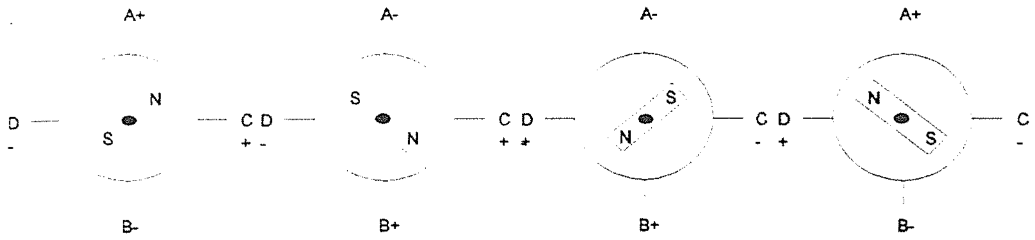


Figure 6C



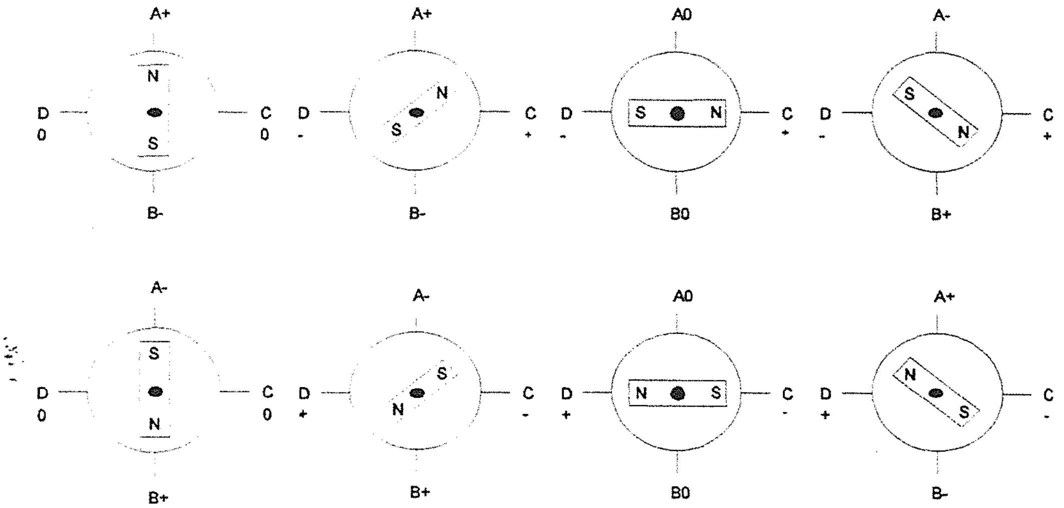


Figure 6D

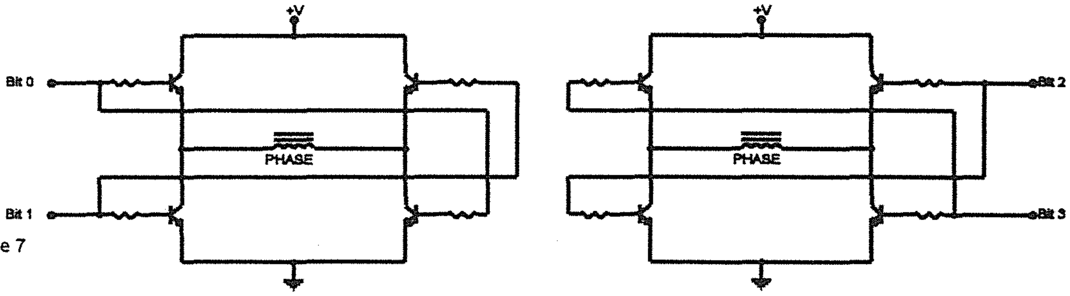


Figure 7

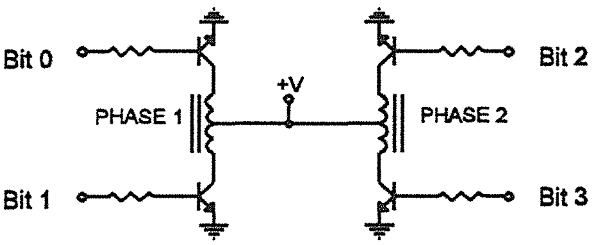


Figure 8

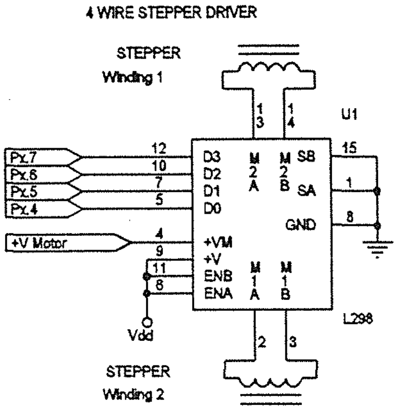


Figure 9

LISTING 1

```
CLS
TRUE = 1
FALSE = 0
REM Open COM port
OPEN "Com1: 9600,N,8,1,CD0,CS0,DS0,OP0,RS,TB2048,RB2048" FOR RANDOM AS #1
'This opens a 2K buffer for receiving and transmitting data at 9600 Bauds. You could use Com2 and any other speed up to 19200 (the highest
speed attainable in Basic. If your computer is fast enough you can go beyond this point by directly poking values into the 8250 IC registers in
the serial port).

PRINT "PLEASE RESET THE ITC232-A"
PRINT
GOSUB READSERIAL
'Wait until the "Welcome..." message is sent and

PRINT S$
'print it on the screen (S$ contains the string received).

W$ = "crap": GOSUB WRITESERIAL
'A command always goes to the writing subroutine in W$. This particular command gets the ITC232-A into program mode to optimize the speed
and handling of results.
W$ = "prF": GOSUB WRITESERIAL
'This is an error example: "prF" will generate an error since there is no F port.
END
                                        SUBROUTINES
REM  Writing to the serial port
WRITESERIAL:
PRINT #1, W$
'This sends the command previously stored in W$ out the serial port. Since the ITC232-A always acknowledges a command with OK or an
error message, we now read what the ITC232-A has to say by polling the serial port
GOSUB READSERIAL
RETURN
'and we return to the main program. The results from the reading of the serial port are returned in S$, V$ and V as described below.

REM  Reading the serial port
READSERIAL:
S$ = ""
```

```
'Empty the string before reading
IF LOC(1) = 0 THEN GOTO READSERIAL
'In case a polling is done before sending any command, as in this program in which at the very beginning the computer waits for the
"Welcome..." message from the ITC232-A.
REM Get the received string into S$
Lp1:
C$ = INPUT$(1, #1)
S$ = S$ + C$
IF C$ <> ">" THEN GOTO Lp1
'The program loops until it finds an ">". This is because the ITC232-A ALWAYS finishes sending data with an ">".
REM decode string (V$) and value (V)
'What follows is not really necessary but handy for getting the results of a function type command in a string variable (V$) and in a numeric
variable (V). It also handles errors. If these features are unnecessary, place a RETURN here and ignore the program below.
VALIDERROR = TRUE
ERRORCODE$ = ""
V$ = ""
'Reset variables.
FOR H = 1 TO LEN(S$)
IF MID$(S$, H, 1) = CHR$(7) THEN VALIDERROR = FALSE
IF MID$(S$, H, 1) = "?" THEN ERRORCODE$ = MID$(S$, H + 1, 1)
NEXT H
'Detect if there is a "?" in S$ that would indicate an error. Eliminate the false signalling of an error in the "Welcome..." message which
contains a "?" but also a CHR$(7) (see next line).

IF (VALIDERROR = TRUE AND ERRORCODE$ <> "") THEN GOSUB
ERRORSUB: RETURN

IF LEN(S$) > 3 THEN V$ = RIGHT$(S$, LEN(S$) - 2): V$ = LEFT$(V$, LEN(V$) - 1)
'If the command was of the function type, then the ITC232-A will return a string longer than 3 characters. In such case, extract the result in V$
and
V = VAL(V$)
'get the value into a numeric variable.

RETURN

ERRORSUB:
PRINT
PRINT "Error #"; ERRORCODE$
RETURN
'This subroutine is self explanatory.
```