To the Graduate Council:

I am submitting herewith a thesis written by John Blake Wilkerson entitled "Microprocessor-based, Tractor-mounted Soil Cone Penetrometer." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Agricultural Mechanization.

Luther R Wilhelm

Luther R. Wilhelm, Major Professor

We have read this thesis and recommend its acceptance:

K. E. Duncan

Accepted for the Council:

L Evans Roth

Vice Chancellor
Graduate Studies and Research

MICROPROCESSOR-BASED, TRACTOR-MOUNTED SOIL CONE PENETROMETER

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

John Blake Wilkerson

June 1983

To:	My family, Mr. and Mrs. Robert Wilkerson, Linda, and Robert Lee for their love and support.
In Memory:	My brother, Jim Wilkerson, who lost his life while saving a friend (1962-1982).

ACKNOWLEDGMENTS

The author wishes to express his appreciation to Dr. Luther R. Wilhelm who served as major professor. His guidance and patience throughout the program were very much appreciated.

Appreciation is also extended to Dr. Houston Luttrell, Agricultural Engineering Department Head, for making funds available for this project.

The guidance and advice of Dr. Fred D. Tompkins, committee member and project leader, is greatly appreciated.

Thanks are also extended to Dr. Kermit E. Duckett, committee member, who provided encouragement throughout my graduate program.

The author also thanks Lewis Buck, former research assistant, for his assistance in the mechanical design.

Special thanks to Robert S. Freeland, graduate student in Agricultural Engineering, for his assistance in microprocessor programming and his friendship.

The encouragement and advice of Mark W. Cantrell, president of Unique Software Development, was greatly appreciated.

The author also wishes to express his appreciation and thanks to Mr. Paul Elliott and Mr. Walker Garner of the Agricultural Engineering Research Shop for their work in construction of the penetrometer.

Thanks is also extended to William E. Hart, Instructor, for his help in construction and support through my program.

Finally, the author wishes to thank the graduate students who provided advice and support on the project.

ABSTRACT

A microprocessor-based, tractor-mounted soil cone penetrometer was developed for automating soil strength measurements. This unit was designed to operate over a 4-row (3-m) width and to a maximum penetrating depth of 61 cm. The penetrating force was applied to the penetrometer shaft via a hydraulic cylinder. The hydraulic cylinder and associated depth and force measuring sensors were mounted on a carriage assembly which traveled along a 4-m track. The carriage was moved laterally along the track by a cable assembly driven by a DC winch. The penetrometer frame and track were attached to a 4-row tool bar which had a category II, 3-point hitch.

Optical position encoders were used to measure depth of penetration and carriage position along the track. Force of penetration was calculated by measuring hydraulic pressure acting on the hydraulic cylinder.

A microprocessor-based control unit activated all moving mechanisms and automatically recorded data on magnetic tape for computing depth versus force of penetration profiles. Liquid crystal displays provided visual output of these measurements. The control unit was field programmable for operating in various modes. Seven parameters which affected operation of the penetrometer could be changed depending upon the application of the test.

The control unit consisted of two modules: (1) the MPU module which included a 8-bit Microprocessor Unit (MC6802), 4K RAM,

4K EPROM, two peripheral interface adapters, an asynchronous interface adapter, 12-bit analog-to-digital converter, three 8-bit data latches, and a BAUD rate generator; (2) the input/output module included two liquid crystal displays, key pad, and light emitting diodes.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

CHAPTER I

INTRODUCTION

A. Background

Soil cone penetrometers have been used in the study of soil strength characteristics for many years. Since its introduction, the soil cone penetrometer has been used to describe numerous soil characteristics in several agricultural engineering applications as well as in other disciplines. The penetrometer has been used extensively as a means of placing a numerical value on the physical state of the soil.

The evaluation of surface soil strength or trafficability relative to the mobility of certain military vehicles and a corresponding soil cone index was first described by Knight and Freitag (1962). Their test device consisted of a right circular 30-degree stainless steel cone with a base area of 3.22 cm^2 (0.5 $in.^2$) mounted on a stainless steel shaft with a diameter of 15.9 mm (0.625 in.). The cone index was determined by the maximum force per unit area required to push the penetrometer through the soil profile to some increment of depth.

Other applications of the soil cone penetrometer have been to determine the ability of soil to support vehicular traffic (Army, 1959), to characterize soil in terms of crop growing ability (Raghaven and McKyes, 1977), and to determine resistance to root penetration and seedling emergence (Bowen, 1976). The correlation between draft force and penetrometer measurements was also researched (Johnson et al., 1978; Gill and VandenBerg, 1968).

Basic cone penetrometers are hand operated instruments with force indicated by the deflection of a proving ring and depth measured by counting scribe marks along the penetration shaft. There are several disadvantages associated with commercial penetrometers of this type: (1) force and depth measurements are both usually recorded manually, (2) two or three men are required to make measurements at a suitable rate, and (3) force of penetration is limited to the weight and strength of the operator. Most researchers agree that a large number of penetrometer readings are needed to make a reasonable prediction of soil strength characteristics. Thus, many measurements, obtained by a uniform method, are needed.

Progress has been achieved in making penetrometer tests faster and easier. Advances include using the tractor hydraulic or electrical system to provide penetration force and recording data on an x-y plotter (Williford, 1972; Smith and Dumas, 1978). Wells and Lewis (1980) have developed a remote electronic acquisition system for recording soil cone index measurements on magnetic tape. Other hand-held systems were modified by Howson (1977) and Hendrick (1969) so that depth versus pressure graphs could be produced mechanically as the penetrometer moved through the soil profile.

These mechanical and electronic recording devices have shown that the man-hours required for gathering data were reduced by over

70 percent and office time required to convert to a graphical format was reduced by about 75 percent compared with an indicating penetrometer (Hendrick, 1969).

B. Objectives

The purpose of this research was to design and construct a soil cone penetrometer for automating soil strength measurements. Specific objectives included:

1. Meet ASAE Standard S313.1 for soil cone penetrometer measurements.
2. Operate the penetrometer over a 3-m width (4-row), penetrate to a maximum depth of 61 cm (24 in.), and penetrate against a maximum force of 6672 N (1500 lbs.).
3. Be tractor mounted using a Category II, three-point hitch.
4. Total automation of all moving mechanisms by a field-programmable control unit.
5. Provide visual display of data using data collection.
6. Automatic storage of data on magnetic tape for later transfer to a computer for subsequent data analysis.

CHAPTER II

REVIEW OF LITERATURE

Depth and force measurements are two basic parameters which must be measured before a cone index calculation can be made. The cone index is usually expressed in units of N/m^2 ($lbs./in.^2$) at a measured depth. The depth is measured by counting the number of scribe marks along the penetrating shaft, which pass through the soil profile. Force of penetration is measured by the deflection on a proving ring. Force of penetration and recording of data are provided by the operator of this device.

Carter (1967) was one of the first researchers to automate depth and force measurements. His hand-operated penetrometer was designed similar to the basic penetrometer noted above. However, a mechanical graphic recording device was attached between the handle and the penetrating shaft. The force component was obtained by the relative displacement between the handle-plunger and an outside barrel which contained an automotive valve spring. As the spring was compressed, a recorder arm moved a recording pen down the chart which was attached to a chart plate. The force measurement was recorded in the vertical direction. A full-scale pressure of 500 $lb./in.^2$ could be applied to the penetrometer tip with a 120 lb. applied force (based upon a 0.2 $in.^2$ cone). The full-scale range was easily increased or decreased by changing the cone tip or replacing the calibration spring. Carter (1967) reported reliable force measurements to the nearest 10 lb./in.^{2}

Depth of penetration was measured on the horizontal scale of the graph. The chart plate moved along a stationary frame driven by a spring attached between the frame and chart plate. A counterweight rested on the soil surface with a string connected to the weight and a measuring wheel attached on the other end of the string. The wheel in turn connected with a 3:1 reduction rack and pinion system. As the shaft of the penetrometer moved into the soil, the counterbalance weight allowed the chart to move in a horizontal plane at 1 in. per foot of penetrating depth.

Hendrick (1969) developed a penetrometer similar to that of Carter except for the depth measurement portion. Depth was measured by the relative movement of a foot-plate shaft. As the penetrometer shaft was forced through soil surface, a plate which remained on the soil surface pushed a gear rack through a series of sprockets and gears. As the penetrometer shaft moved into the soil, the chart plate was moved in a horizontal plane.

Howson (1977) designed a recording cone penetrometer similar to that used by Hendrick (1969) and Carter (1967). The mechanical recording device used a circular graph for recording force-depth measurements. The procedure described by Hendrick was used to measure force of penetration; however, depth measurement was obtained by rotating the graph in a circular manner as opposed to the horizontal movement of the paper.

Carter (1969) improved upon a previous design (Carter, 1967) by electrically integrating his force measurements with respect to depth. The force to move the penetrometer cone through the soil was sensed by a proving ring with electrical resistance strain gauges. The strain gauge bridge excitation was supplied by a DC generator mechanically connected so that output voltage was proportional to rate of penetration (dy/dx) multiplied by force. The output of this bridge was then integrated with respect to time by an operational amplifier equipped with capacitor feedback. Output from the operational amplifier was connected to a milliamp meter via a series of resistors. The meter provided a visual output of the average force over a given depth.

Prather et al. (1970) designed and constructed an electronic hand-operated recording penetrometer. The construction was similar to that of Hendrick (1969). The Prather penetrometer also made use of graphic recording; however, force and depth measurements were made electronically and fed to a small x-y plotter mounted on the penetrometer.

The force measurement was created by a load cell attached between the handle and the penetrating shaft. A battery supplied a constant voltage source to the load cell which consisted of strain gauges oriented in a wheatstone bridge arrangement. When a force was applied to the load cell, a small electrical signal was produced proportional to the force on the cell. This signal was then routed through an amplifying circuit to the x-y plotter. A maximum force of 200 lbs. could be obtained in ranges of 0 to 200 lbs., 0 to 100 lbs., and 0 to 50 lbs. This provided a greater resolution in the lower force ranges. The depth measurement was obtained through a depth transducer containing a 10-turn potentiometer. The potentiometer was mounted in a housing which rested on the soil, and through which the cone shaft passed. As the cone shaft moved through the soil, the potentiometer was rotated by a friction disk contacting the shaft. Thus, the output signal of the potentiometer was proportional to the penetration of the cone into the soil. This signal, like the force signal, was then fed through a signal conditioner and preamplifier before being fed to the x-y plotter. The entire data gathering system was powered by small rechargeable batteries. Prather et al. 1970) determined that approximately 400 tests or 10 hrs. of testing could be achieved with an error of less than 1.0 percent.

Wells and Lewis (1980) developed an electronic remote acquisition system for recording soil cone index measurements. This system allowed one person to determine and electronically record penetration force versus depth data pairs on magnetic tape. The depth transducer consisted of a rack and pinion assembly which turned a 1K potentiometer with a measured accuracy of 1 percent. The force was measured by a load cell coupled to the penetrating shaft. Wells and Lewis' method of sensing force and depth was similar to that of Prather et al. (1970). However, Wells and Lewis stored data on magnetic tape via a microprocessor controlled data acquisition system.

Wells' data logging system consisted of a CPU (Central Processing Unit) for controlling the instrumentation according to a program stored in EPROM (Erasable Programmable Read Only Memory).

Control logic sequences were developed to perform: calibration of the load cell; calibration of the DAC (Digital to Analog Converter); entering of a 16-bit number for identification code; measuring, digitizing and storing penetrometer force versus depth data; transmitting and recording data onto magnetic media via a 20-ma current loop; and viewing the data on a LCD (Liquid Crystal Display). An 8-bit DAC provided 256 digital values of force and penetration depth. The resulting sensitivity was 3.5 N (0.8 lbs.) for force over a 890 N (200 lbs.) range and 1.2 mm (.05 in.) for penetration depth of 32.5 cm (12.8 in.).

The CPU read the depth and pressure coordinates and stored them in RAM (Random Access Memory). After nine depth-pressure readings were made, the values stored in RAM were transferred to cassette tape. A 90-minute tape would store up to 540 force-depth relationships. Power for the system was supplied by four 6-volt batteries. This supply source was adequate for a typical day of data collection, according to Wells and Lewis. This remote recording of data saved time in the field spent changing charts like those of Smith (1978) and others with similar designs. However, Prather et al. reported advantages of putting multiple recordings on one graph for comparison in the field.

Anderson et al. (1980) developed a hand-held penetrometer which took the analog signals and digitized them for data storage and simple statistical calculations in the field. The force sensing transducer was developed with strain gauges to measure deflection of a parallelogram. This type of transducer had the advantage of being insensitive to side loads while having good resolution in the required direction and quick response time. This analog signal was digitized by passing it through an amplifier and a voltage-to-frequency converter. The output from the converter was gated through a 100-ms clock. This provided an average over a short period of time for a more dependable reading as opposed to instantaneous readings.

The depth was measured relative to a plate which rested on the soil surface. An optical system with two infrared emitters and detectors provided a double-window output from holes accurately spaced at equal intervals along a preselected interval. A depth measurement system of this type gave an accuracy better than 1 mm, and depth increments as small as 1 cm could be specified by the operator. The double window arrangement allowed the instrument to sense the direction of travel, avoiding false depth readings.

A programmable calculator was used to interface with the digitized signals from the sensors. This calculator provided a display for viewing, or a printout could be obtained. For each penetration, mean force, standard deviation and number of points read were also available to the operator in the field. With the use of CMOS solid-state devices and a LCD display, this allowed continuous measurement for up to 60 hours.

Williford (1972) was one of the first researchers to use auxiliary power to force the penetrometer shaft into the soil. A tractor mounted field penetrometer was designed and constructed allowing the hydraulic system to force the penetrating shaft into the soil. The penetrometer unit was mounted on a frame with wheels driven by a low-speed hydraulic motor. A four-row multiple toolbar served as a carrier for the unit. This lateral movement allowed the penetrometer to be positioned at any point along the 10-foot tool bar.

Williford's technique in measuring force was similar to that of Prather et al. (1970). A load cell was coupled between the hydraulic cylinder shaft and the penetrometer shaft. The signal was then recorded on an x-y plotter. Williford chose to avoid measuring depth by mechanical linkages; instead, he used an x-y plotter equipped with a time base option. Speed of penetration was controlled by a hydraulic regulating circuit, thereby producing a constant linear velocity of the penetrometer shaft. Hence, depth axis of the plotter could be driven by the time base circuit synchronized to the speed of the penetrometer. The recorder was turned on when the penetrometer tip passed through the soil, thereby starting the plotter at a zero depth. The plotter and strain gauges were powered from a 120-VAC power inverter driven by the 12-VDC voltage source available on the tractor.

Smith and Dumas (1978) built a tractor-mounted soil cone penetrometer with frame and three-point hitch similar to that of Williford (1972). However, the penetrometer carriage was laterally moved by a manually operated crank which engaged an endless roller chain attached to the carriage.

The penetrometer shaft was driven by an electric motor connected to a speed reducer via a drive shaft which rotated two spur gears. The gears attached to the shaft drove a rack gear attached to the penetrometer shaft. Force was measured and recorded by a load cell and x-y plotter as described by Hendrick (1969). The depth of penetration was sensed indirectly by placing a 10-turn precision potentiometer on the output shaft of the speed reducer which indirectly forced the penetrometer shaft into the soil. The potentiometer was initially adjusted such that its output was zero when the penetrometer cone was raised to its highest point, thus providing a linear signal proportional to depth for the depth axis of the plotter.

A 2-kW portable generator supplied power to the penetrometer motor and powered the x-y plotter. The excitation voltage for the transducer was provided by a 6-V dry cell battery.

CHAPTER III

DESIGN OF MECHANICAL SYSTEM

A. Frame and Track Assembly

A frame and track were constructed to support the penetrometer and associated measuring and control devices (see Figures 1 and 2). The track was designed to allow the penetrometer to move across four 80-cm rows, a total distance of 3.2 m. The track was constructed from two C-channels 3.8 X 10.2 cm turned on edge to provide a track for the penetrometer carriage assembly (refer to Figure 3). The track, 3.7 m in length, gave an effective but reduced travel distance of 3 m due to physical obstructions on the carriage assembly (see Figure 4).

The track was welded to end frames 81 cm high and 71 cm wide, constructed from 5.1 X 7.6 cm welded steel rectangular tubing. The frame provided support for the track and a mounting surface for a 3.7-m tool bar installed parallel to the track. The tool bar was fastened to the frame by removable clamp plates which permitted the tool bar to be used separately from the penetrometer. The tool bar had steel plates welded on each end with an inlet and drain provided in one end. By adding water inside the tool bar, an additional 136 Kg of mass could be added to the penetrometer.

Category II, three-point hitch clamps were attached to the center of the tool bar for transporting and operating the penetrometer with an agricultural tractor.

Figure 1. Rear view of soil cone penetrometer mounted on a 46 PTO Kw agricultural tractor.

Figure 2. Side view of soil cone penetrometer ready for use in the field.

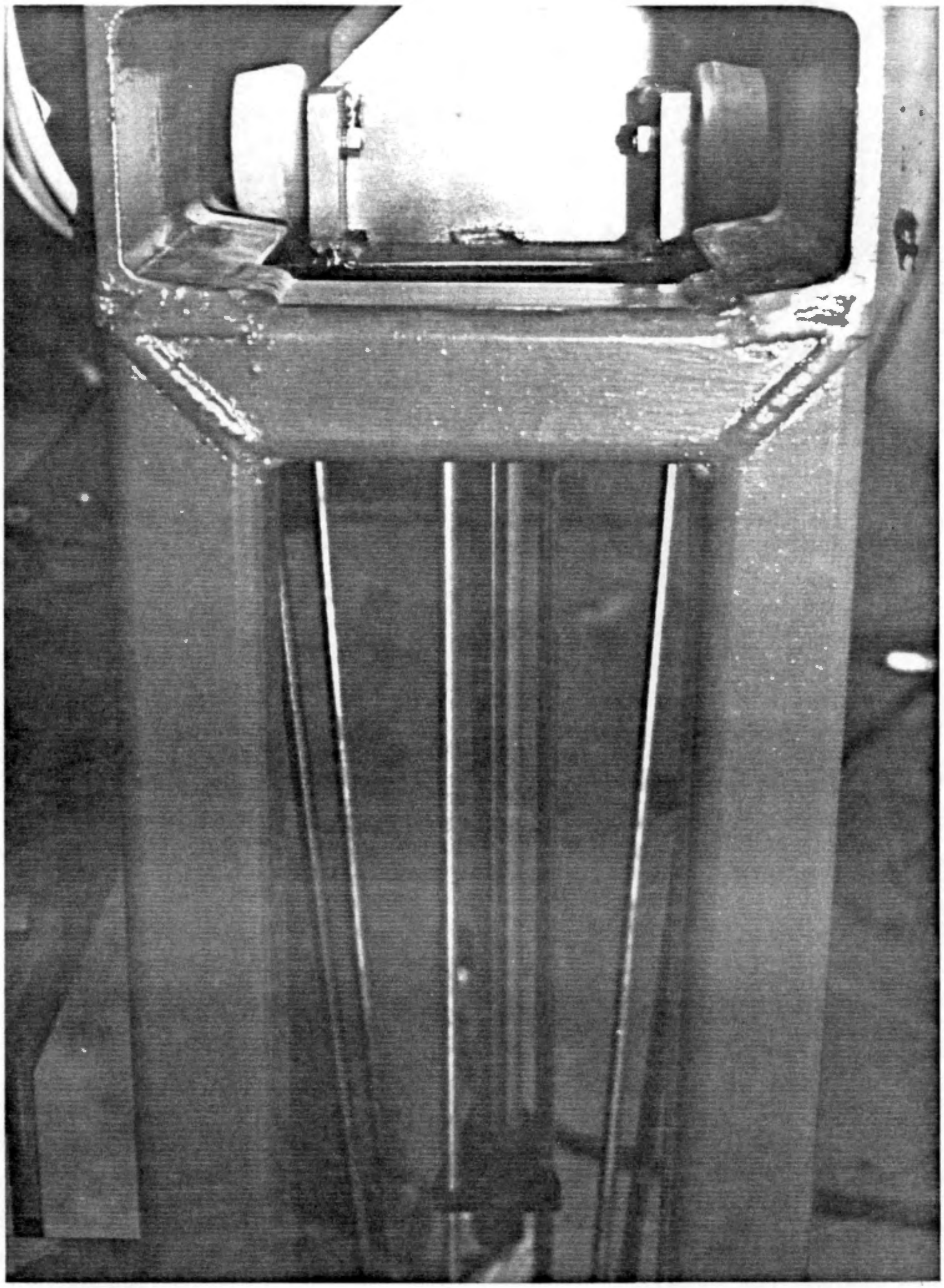


Figure 3. End view of penetrometer showing carriage assembly inside the track.

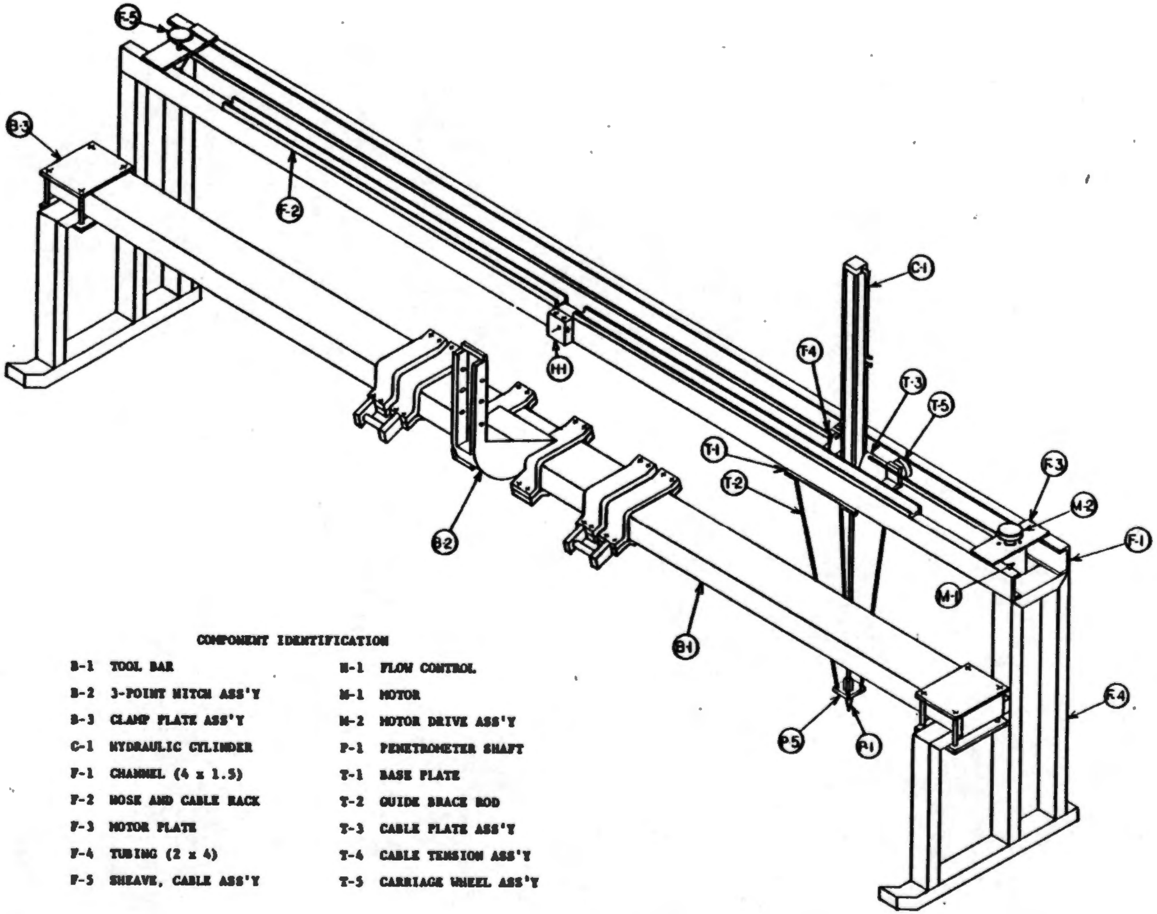


Figure 4. Soil cone penetrometer frame and track including carriage assembly.

B. Penetrometer Carriage Assembly

The stainless steel penetrometer shaft was machined to a diameter of 1.586 cm and a length of 71 cm. One end of the rod was threaded to accept a 2.03 cm diameter cone tapered at a 30 degree angle. The cone could be replaced when excessive wear or physical damage occurred. ASAE standard S313.1 advises replacing the cone after 3 percent wear. This degree of wear would result in a 5 percent error in cone index measurements.

The top of the shaft was also threaded to adapt by means of a female coupling to the shaft of a hydraulic cylinder (refer to Figure 5). A 8-cm by 71-cm double-acting hydraulic cylinder was used to drive the penetrometer shaft. The cylinder was bolted to the trolley assembly that traveled inside the track. The penetrometer shaft was supported by a 0.62 X 0.95 cm linear bearing mounted in a machined housing normally positioned 10 cm from the soil surface. The bearing assembly was held in alignment by four 0.95-cm rods attached between the bearing assembly and the base which held the hydraulic cylinder.

C. Hydraulic System

Operation of the hydraulic cylinder was controlled by the circuit illustrated in Figure 6. Hydraulic lines (feed and return) were connected to the tractor via quick-disconnect couplers on one end and attached to the three-position, four-way directional control valve on the other. The valve was activated by either of two 12-volt solenoids. When the valve spool was positioned to extend the rod,

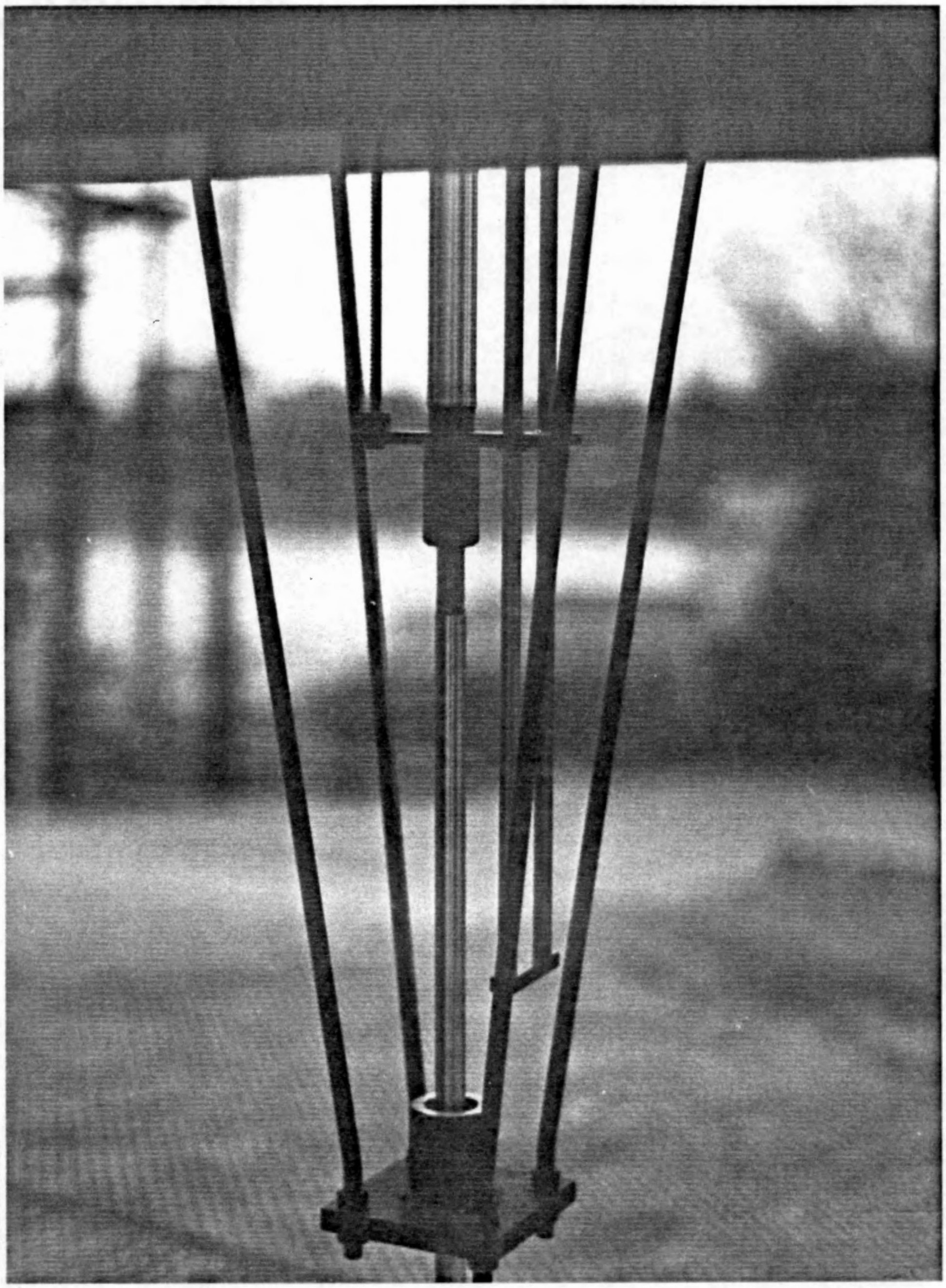


Figure 5. Penetrometer shaft support structure and hydraulic cylinder couplings. This figure also shows guide rods which prevent the shaft from rotating.

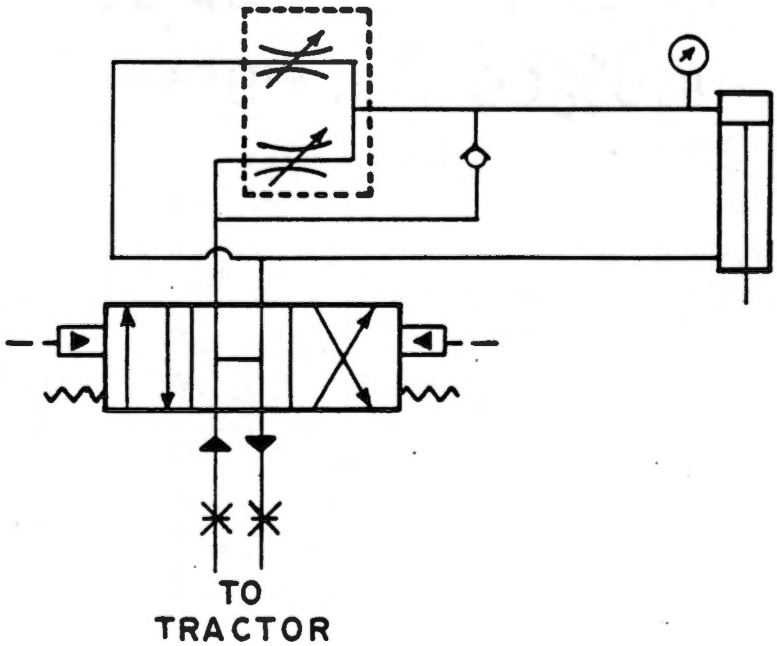


Figure 6. Hydraulic circuit for controlling the cylinder used to drive the penetrometer shaft.

hydraulic fluid was routed through a constant flow valve. This constant flow was regulated to provide a linear velocity of 3 cm/s for the penetrating shaft regardless of soil resistance. When the valve was positioned to retract the cylinder rod, flow was routed around the constant flow valve through a check valve to permit a quick return of the penetrometer shaft.

D. Carriage Drive Assembly

The penetrometer was moved back and forth along the track assembly by a closed cable loop. The cable was attached to the penetrometer carriage on both ends with the cable going around an idler pulley on one end of the track and a drive pulley on the other. The steel cable selected was 0.64 cm in diameter with a hard plastic coating. The plastic served two functions, prevention of rusting and reduction of slippage between the cable and the drive pulley.

A 12-volt winch was modified to serve as the cable driver. The bell housing was removed and replaced by an aluminum pulley machined to fit the cable and having an effective circumference of 30.5 cm. This also provided an acceptable penetrometer carriage travel speed of 30.5 cm/s.

E. Depth Measurement

A linear potentiometer has typically been used for depth of penetration measurements (Williford, 1972). However, this method was not acceptable for the control unit as designed. Problems associated with this procedure included false signal readings due to the noisy environment and difficulties in processing of the signal. Most of the noise was electromotive (EMF) associated and was produced by high inductive starting and stopping loads.

An optical position shaft encoder was selected to determine the penetration depth. This sensor had a TTL level binary output with high resolution and high noise tolerance. The model selected (Litton Inc., Model 76 HD) was waterproof and housed in a heavy duty case. The output interfaced directly with the control unit. A 10-bit resolution with 1024 increments around a 360-degree shaft was selected.

During operation, a 0.95-cm gear rack moved with the penetrometer shaft. The rack motion drove a geared transmission with a 28:1 ratio (see Figure 7). Hence, 71 cm of linear travel by the cylinder results in one revolution of the encoder shaft. Using a 10-bit resolution over 71 cm of depth would, theoretically, produce a resolution of 0.07 cm. However, two bits of resolution were not repeatable, due to backlash in the gears. Omitting the two least significant bits gave a dependable reading to the nearest 0.28 cm. This allowed a maximum of 256 descrete depth-pressure measurements with the full 71-cm stroke. A 10-cm ground clearance was provided between the cone tip and the ground, thereby permitting a maximum penetration depth of 61 cm.

F. Force Measurement

The other variable to be measured was the force on the cone at a given depth of penetration. Typical procedures for making this

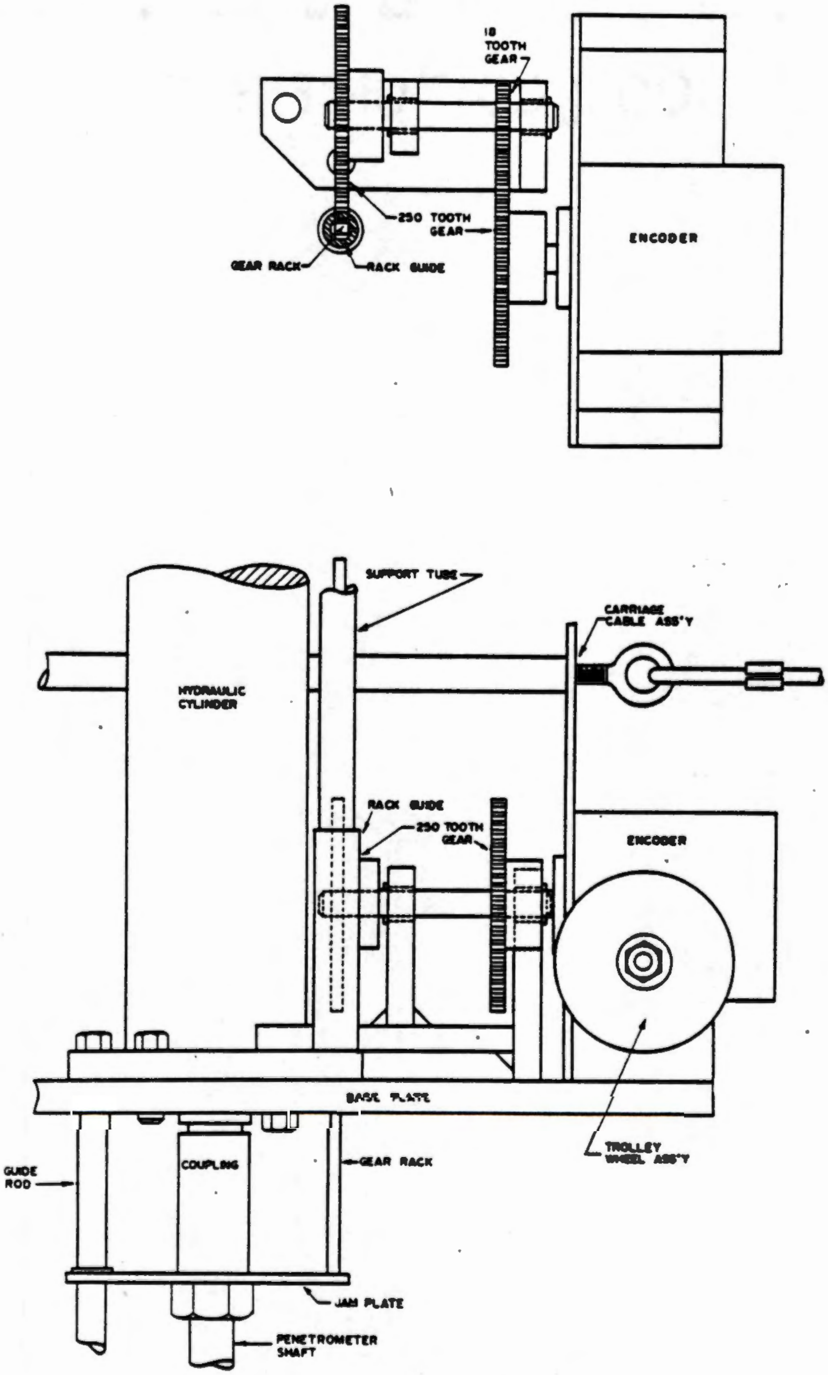


Figure 7. Top and side view of gear-encoder mechanism for depth position measurement. The gear rack, moving with the penetrometer, drives gears to turn the encoder.

measurement have included using either a load cell or strain gauges mounted above or on the penetrating shaft, respectively. This type of transducer produced a low-level analog signal output. This signal, like the potentiometer signal, was very susceptible to noise produced by electric motors and valves, both of which were present on this machine.

A pressure transducer was selected to measure the hydraulic pressure acting on the cylinder (see Figure 8). By knowing the area of the cylinder (20.27 cm^2) and the pressure acting on that given area, the force on the cylinder shaft could be determined. This transducer, manufactured by Celesco Inc., equipped with a pizeo-electric crystal, sensed the hydraulic pressure and produced an analog signal amplified to 5 volts full scale. The transducer had a full scale range of 0 to 500 psi; therefore 10 mV equaled 1 psi. A 500-psi pressure acting on a 20.27 cm^2 area of the cylinder resulted in a maximum penetration force of 6987 N. An accuracy of 0.5 percent including linearity, hysteresis, and repeatability was specified by the manufacturer. Hence, accuracy for the force measurement from the pressure transducer was ± 1.8 N based upon a 0.5 percent accuracy.

The ADC used had an accuracy of 12 bits or 4096 increments between 0 and 10 volts. This provided a resolution of 0.84 N/m^2 full scale. This in turn produced a force measurement accurate to the nearest 1.71 N.

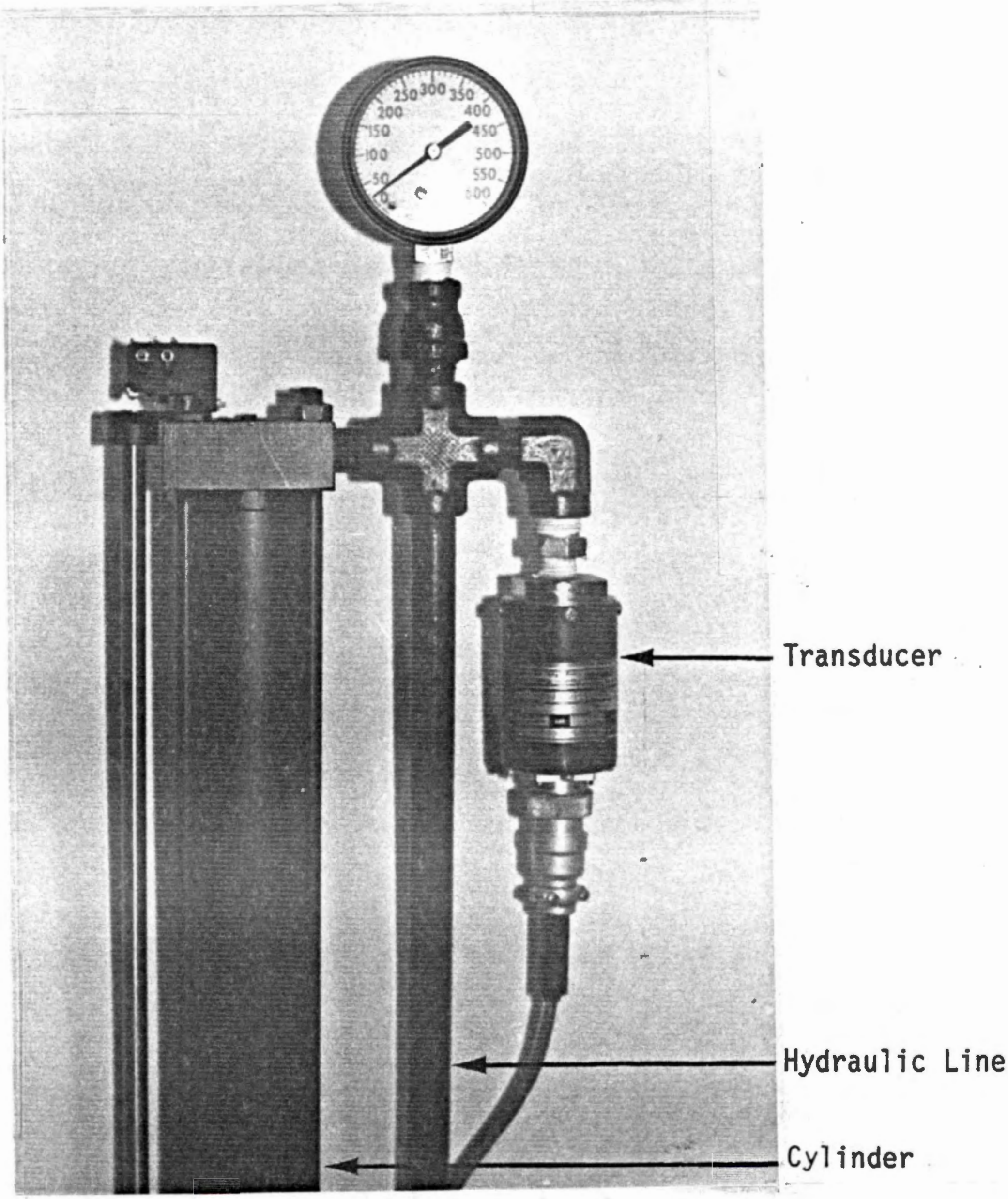


Figure 8. Pressure transducer for measuring the force acting on the hydraulic cylinder.

G. Carriage Position Measurement

The penetrometer position measurements were made in a manner similar to that used for depth measurement. The encoders for horizontal and vertical positioning were identical. With a total horizontal travel of 3.2 m, the shaft of the encoder turned one revolution. To achieve this, a 10.5:1 ratio was obtained through a series of small pulleys and double O-rings (refer to Figure 9). A resolution of 0.03 cm was produced with a 10-bit encoder. The output from the encoder interfaced directly with the control unit.

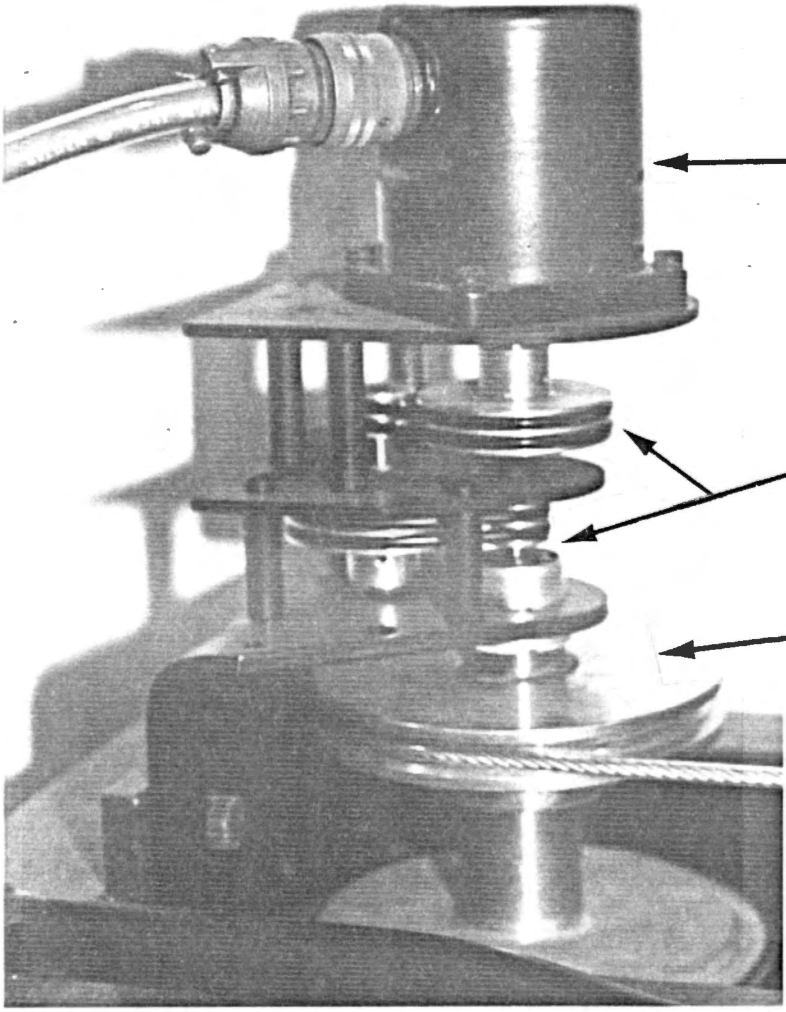


Figure 9. Carriage drive and position sensor.

CHAPTER IV

DESIGN OF CONTROL UNIT AND INTERFACE

A. Design Criteria

A control unit was designed to monitor all input parameters, control all moving mechanisms, display force-depth data, accept commands through a key pad matrix, and communicate with a DEC TU-58 magnetic tape recorder for data storage. (See Figure 10 for a graphic illustration of the design.)

B. Microprocessor Control Module

<u>Microprocessor Unit</u>

The control module was designed around a Motorola MC6802 Microprocessor Unit (MPU). Other components that made up the module were 4K of RAM, 4K of EPROM, two peripheral interface adapters (PIA), a 12-bit analog-to-digital converter (ADC), three 8-bit data latches, an asynchronous interface adapter (ACIA), and other associated hardware required for addressing and communication (see Figures 11 and 12). The MPU used was a monolithic 8-bit microprocessor which contained an on-chip clock circuit, 128 X 8 bit of RAM, software compatible with the MC6800, standard TTL-compatible inputs and outputs (I/O), 8-bit word size, 16-bit memory addressing and interrupt capability. This MPU was enclosed in a 40-pin package and included six internal registers.

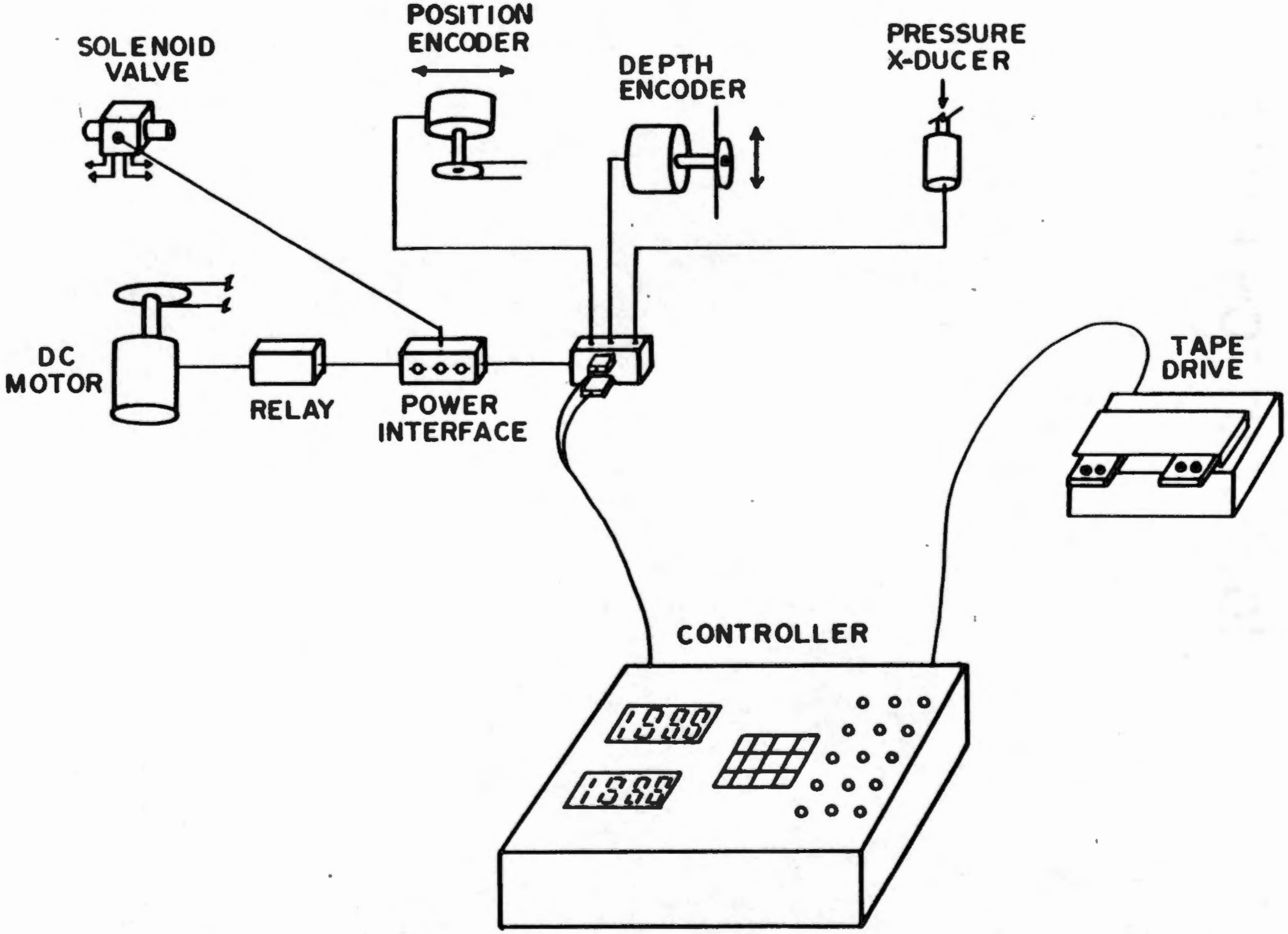


Figure 10. Graphic illustration of control unit and associated input and output devices.

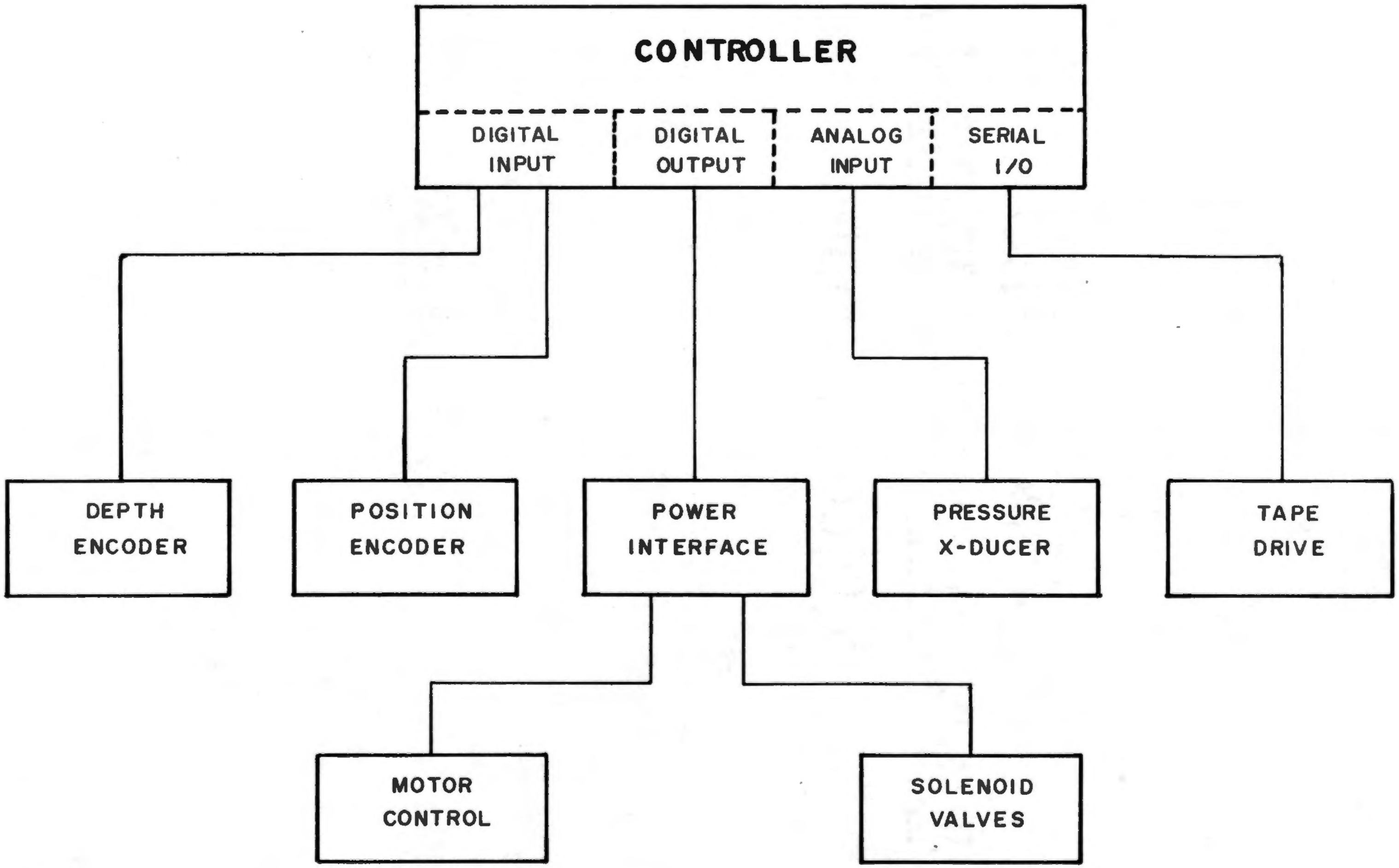


Figure 11. Diagram of controller with associated interface signals and sensors.

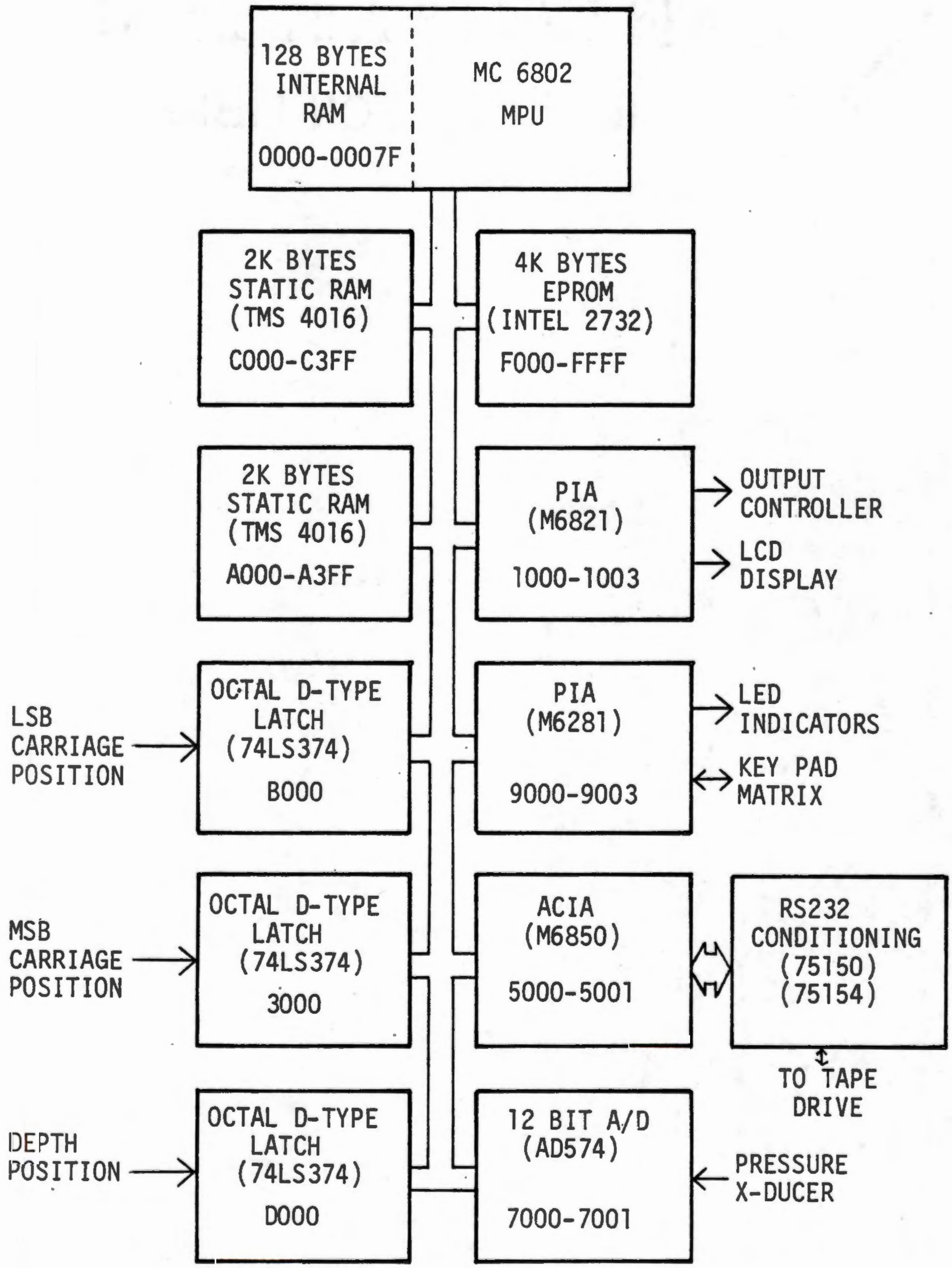


Figure 12. Block diagram of microprocessor, addressing, and I/O capabilities. Hexadecimal addresses for each component are shown inside the block for that component.

The MPU had an onboard clock circuit which permitted a 3.58-Mhz color-burst crystal to be connected directly to the MPU chip. The clock circuit divided the crystal frequency by 4 which generated a 0.895-Mhz clock speed. This was under the 1-Mhz maximum clock speed specified by the manufacturer.

Three interrupts were provided on the MPU, one of which was a software interrupt used by the program. A hardware interrupt could occur on either a Nonmaskable Interrupt (NMI) or an Interrupt Request (IRQ) line. Refer to Motorola (1981) for details on the MPU operation.

Memory

An Intel 2732A EPROM with 4K X 8 bits was used to store the software permanently in memory. This permanent software will be referred to as "firmware." Two static RAM chips, manufactured by Texas Instruments (TM4016) and containing 2K X 8 bits, were used to provide the temporary memory needed by the MPU for storage of force-depth measurements, variables in the program, and memory for the stack.

Input/Output

Two Peripheral Interface Adapters (PIA) were used as I/O control for the power interface, key pad, indicator lights, and display units. These N-MOS devices were capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data busses and four control lines. Each of the peripheral data lines could be programmed to act as either an input or output and the interrupt lines could be programmed to operate in several modes.

An ACIA (MC6850) was used to serially communicate via an RS232C signal to a DEC TU58 tape drive. The parallel data of the bus system was serially transmitted and received by the asynchronous data interface, with proper formating and error checking. A programmable control register provided variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. The clock source was provided by a Baud Rate Generator (MC11411) and a 1.8432-Mhz crystal. This CMOS chip allowed a variable baud rate between 50 and 9600 to be selected by wire-wrapping. The generator was wired to provide a 9600 baud clock rate which was compatible with the TU58 tape drive. However, the output signal from the ACIA was a TTL level source and an RS232C-type signal was needed for communication with the TU58. The output from the ACIA was sent through a line driver (TM75150) to convert the signal from TTL (0-5VDC) to RS232C (± 12 VDC). This recommended dual voltage level of ± 12 was not available on the control unit; however, a ± 15 VDC supply was available, and this voltage level did not exceed the maximum specified by the manufacturer. (Refer to Texas Instruments, 1980.) Line receivers (TM75154) were also installed for this communication requirement.

The 8-bit latches were used to latch output from the optical encoders on the data bus when addressed by the MPU. Octal D-type latches (74LS374) with tri-state outputs were selected for this interface need. Two extra PIA's could have been used. However, since the lines were inputs only, the bidirectional design on the

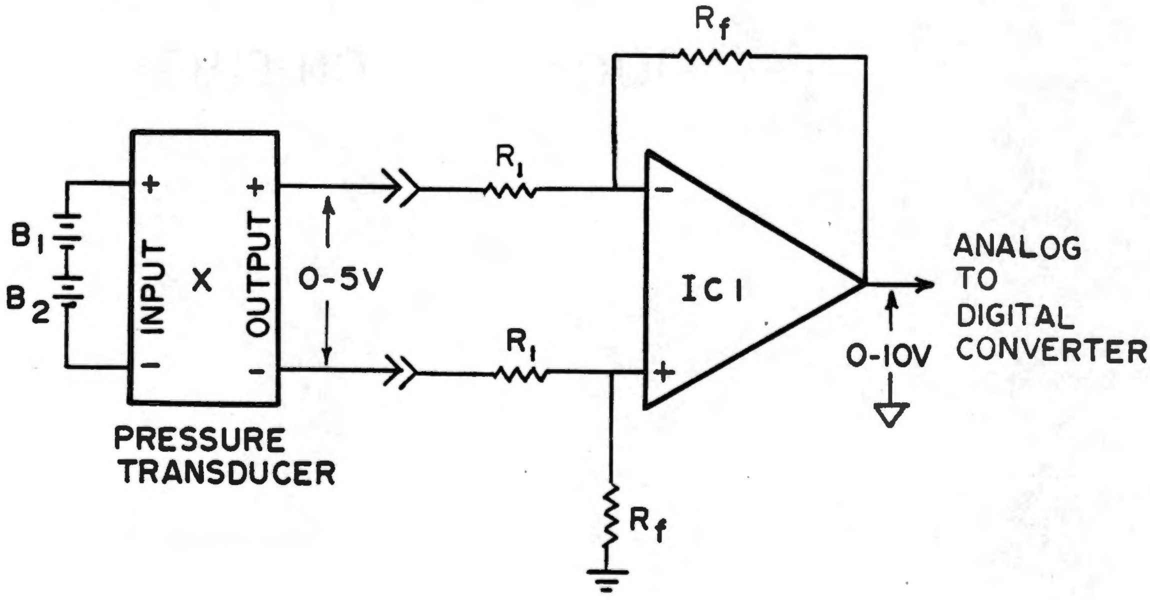
PIA's was not needed. This allowed three 20-pin chips to replace two 40-pin chips which would have required more space on control board. Since this latch-type chip is not Motorola compatible as such, other small scale integrated chips (OR, AND) were needed to assist in the read and chip select control.

Analog-to-Digital Converter

The only analog signal on the penetrometer was provided by the pressure transducer. Since the signal was amplified to 0-5 volts by the transducer and the typical range selection on an analog-to-digital converter is 0-10 volts full scale, a differential amplifier with an amplification of two was used as a pre-amplifier in front of the ADC (see Figure 13). This circuit took the difference between the plus and minus output of the transducer and amplified that difference by a factor of two. This supplied the ADC with a full scale voltage of 0 to 10 volts.

The ADC selected, manufactured by Analog Devices (AD574), was a 12-bit successive-approximation converter with 3-state output buffer circuitry for direct microprocessor interface. Only 8-bits of the 12-bit data were addressed at a time; therefore, the output from the ADC was configured to accommodate this application (refer to Analog Devices, 1980).

To start a conversion, the ADC was addressed and the read/write (R/W) line pulsed low. To obtain this conversion, the R/W control line was connected to the read/convert (R/C) pin on the ADC. The maximum conversion time was 35 microseconds as specified by the manufacturer.



COMPONENT IDENTIFICATION

B1 - B2	12V BATTERY
IC1	OPERATIONAL AMPLIFIER (LM351)
R1	75K RESISTOR (1/4 WATT, 1%)
Rf	150K RESISTOR (1/4 WATT, 1%)
X	PRESSURE TRANSDUCER (0-500psi)

Figure 13. Differential amplifier for converting the double-ended transducer signal to a 2X amplified single-ended signal.

Addressing

The MPU communicated with 10 associated devices through the addressing scheme shown in Figure 14. Address lines 15, 14, 13, and 12 were used to select the individual devices. A 3-to-8 decoder (74LS138) was used to select eight of the 10 devices. The other two devices not selected by the decoder, ADC and RAM#2, were addressed through NAND logic (3-input, 74LS10). Address line 12 was used to select either the components addressed by the decoder or the NAND logic. When address line 12 was high, the decoder was selected. Address lines 15, 14, and 13 were used to select the individual devices addressed through the decoder.

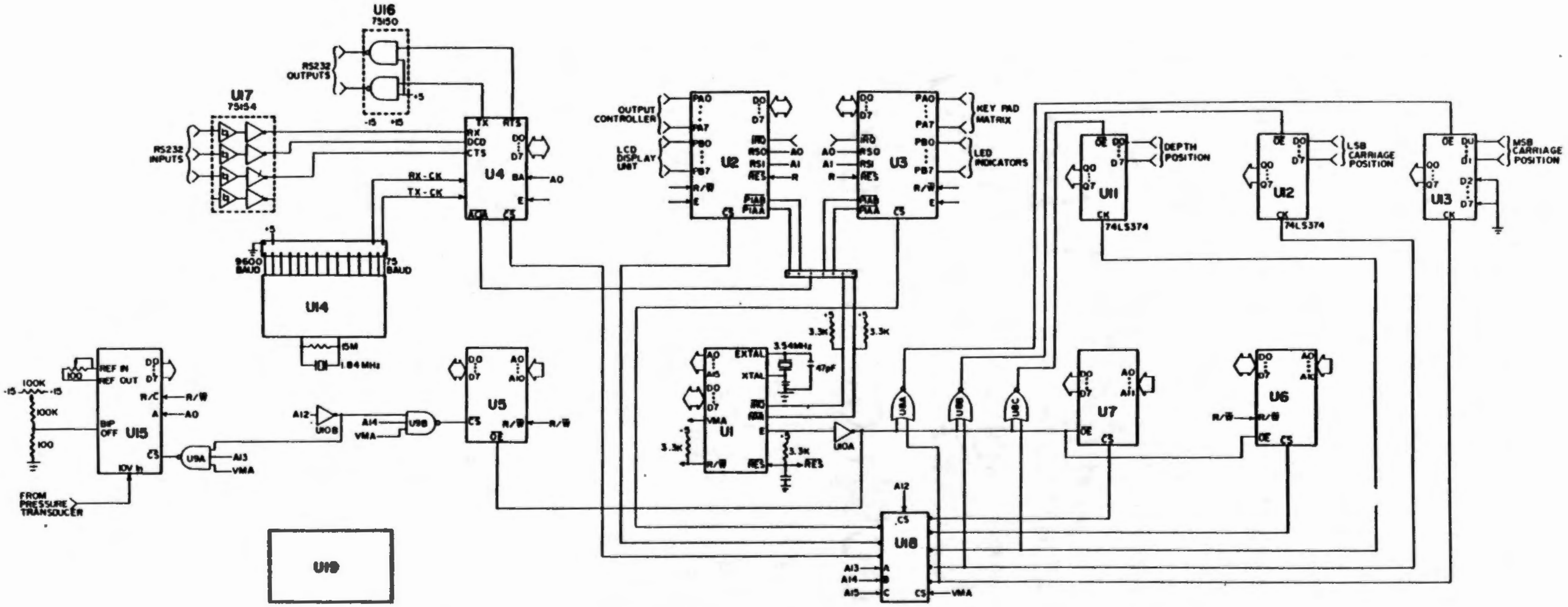
Address line 12 was inverted to make up one of the three inputs on each of the NAND gates for selecting the ADC and RAM#2 chips. Address lines 13 and 14 were used to select the ADC or RAM#2 chip, respectively, when address line 12 was low. The valid memory address (VMA) line from the MPU was connected to one of the chip select pins on the decoder and to the third input on both NAND gates. This VMA line prevented invalid addresses. Refer to Figure 15 for a complete schematic drawing of the addressing.

Construction of Control Module

The MPU, I/O devices, memory, ADC, and all other associated integrated circuits for addressing and drivers were mounted on a single S-100 bus vector board. Figure 16 shows the board layout with the components. All the chips were placed in gold-lay

MPU ADDRESSING LINES (A0-A15)																	ADDRESS	
DEVICE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	FROM	TO
EPROM	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	F000	FFFF
RAM #1	0	1	1	1		X	X	X	X	X	X	X	X	X	X	X	7000	73FF
RAM #2		1	0	0		X	X	X	X	X	X	X	X	X	X	X	C000	C3FF
PIA #1	1	0	0	1											X	X	9000	9003
PIA #2	0	0	0	1											X	X	1000	1003
ACIA	0	1	0	1												X	5000	5001
ADC		0	1	0												X	A000	A001
LATCH #1	1	0	1	1													B000	B000
LATCH #2	0	0	1	1													3000	3000
LATCH #3	1	1	0	1													D000	D000

Figure 14. Address configuration for the devices associated with the microprocessor.



COMPONENT DESCRIPTION					
U1	Microprocessor (MC6802)	U8	OR Gates (74LS32)	U15	12-bit A/D (ADC574)
U2 - U3	PIA's (MC6821)	U9	NAND Gates (74LS10)	U16	Line Drivers (75150)
U4	ACIA (MC6840)	U10	Inverters (74LS04)	U17	Line Receivers (75154)
U5-U6	RAM (4016)	U11 - U13	D-Type Latch (74LS374)	U18	3-TO-8 Decoder (74LS138)
U7	EPROM (2732)	U14	Baud Rate Generator (MC14411)	U19	DC to DC Converter (5 to ±15)

Figure 15. Schematic diagram of the microprocessor control unit module.

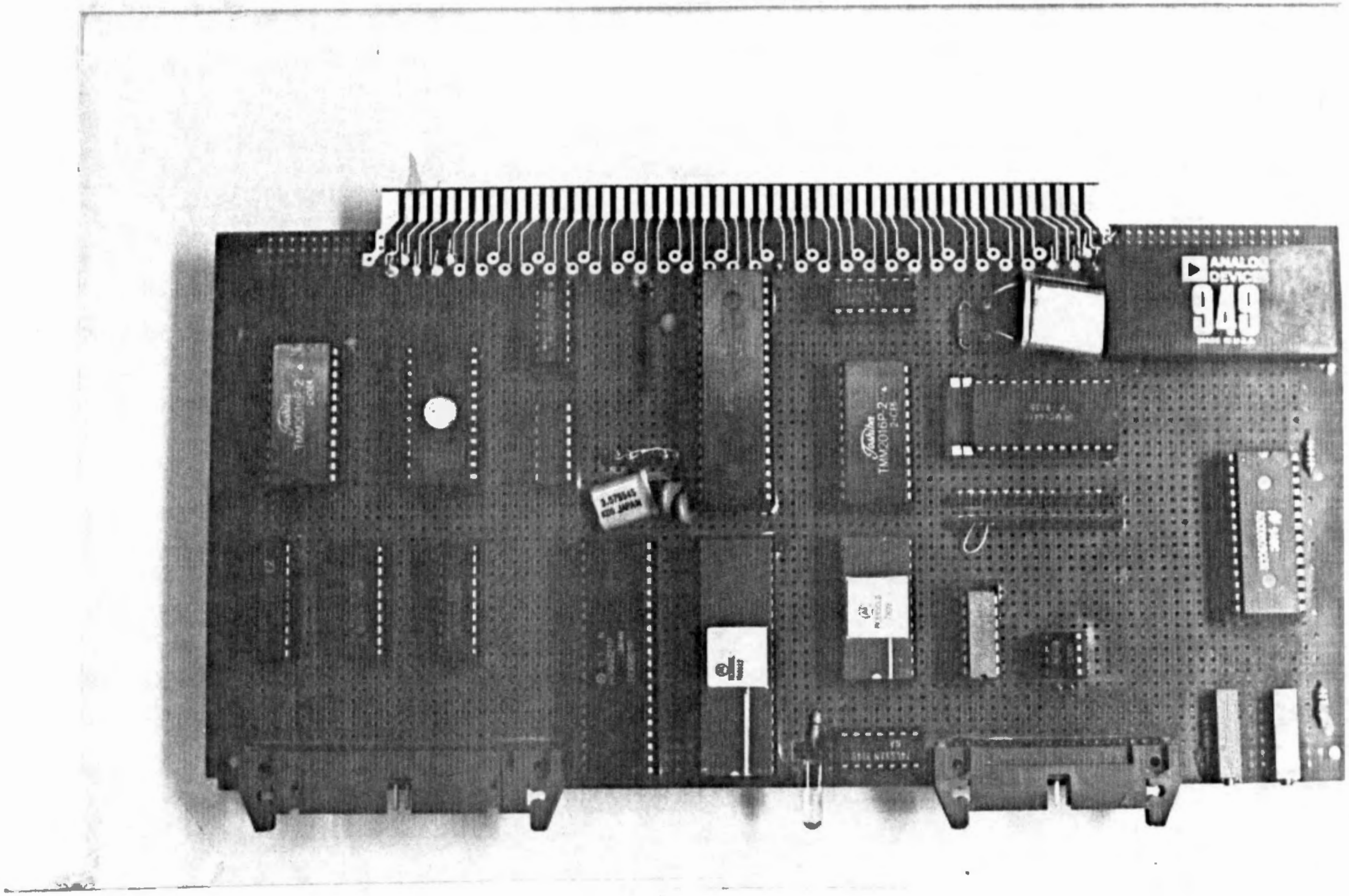


Figure 16. Microprocessor control module in the control unit.

wire-wrap pin sockets. The sockets were glued to the vector board for permanent mounting.

The wiring connections were made by using an automatic wire-wrap stripper tool which permitted daisy chaining from pin to pin. Power for the chips was provided through the edge connectors on the vector board. The interface connections for the PIA's, latches, ACIA, and the ADC were provided through 40- and 26-pin Berg ribbon cable connectors with wire-wrap pin terminals.

C. Interface Module

Display Unit

Liquid Crystal Display (LCD) units comprised of two 3-1/2-digit, 7-segment displays were mounted on the interface board. These units were used to display depth and pressure simultaneously during data collection and to display setup variables during the setup routine. They were chosen because of their high visibility in bright sunlight and low power requirements. The LCD units were manufactured by Epson, Inc. These dynamic displays required an operational voltage of 5 volts RMS (min.) at a frequency of approximately 30 Hz for operation. The entire control unit was powered by a 5-volt source with a ± 15 VDC DC-to-DC power converter available for a dual voltage supply. A gated oscillator was designed from two CMOS NAND gates (4011), a resistor, and capacitor (see Figure 17). The output from this oscillator was a square wave with an amplitude of 0 to 15 volts and a 50 percent duty-cycle. To provide for the AC signal, the common connector on the LCD units was pulled to 7.5 volts (static)

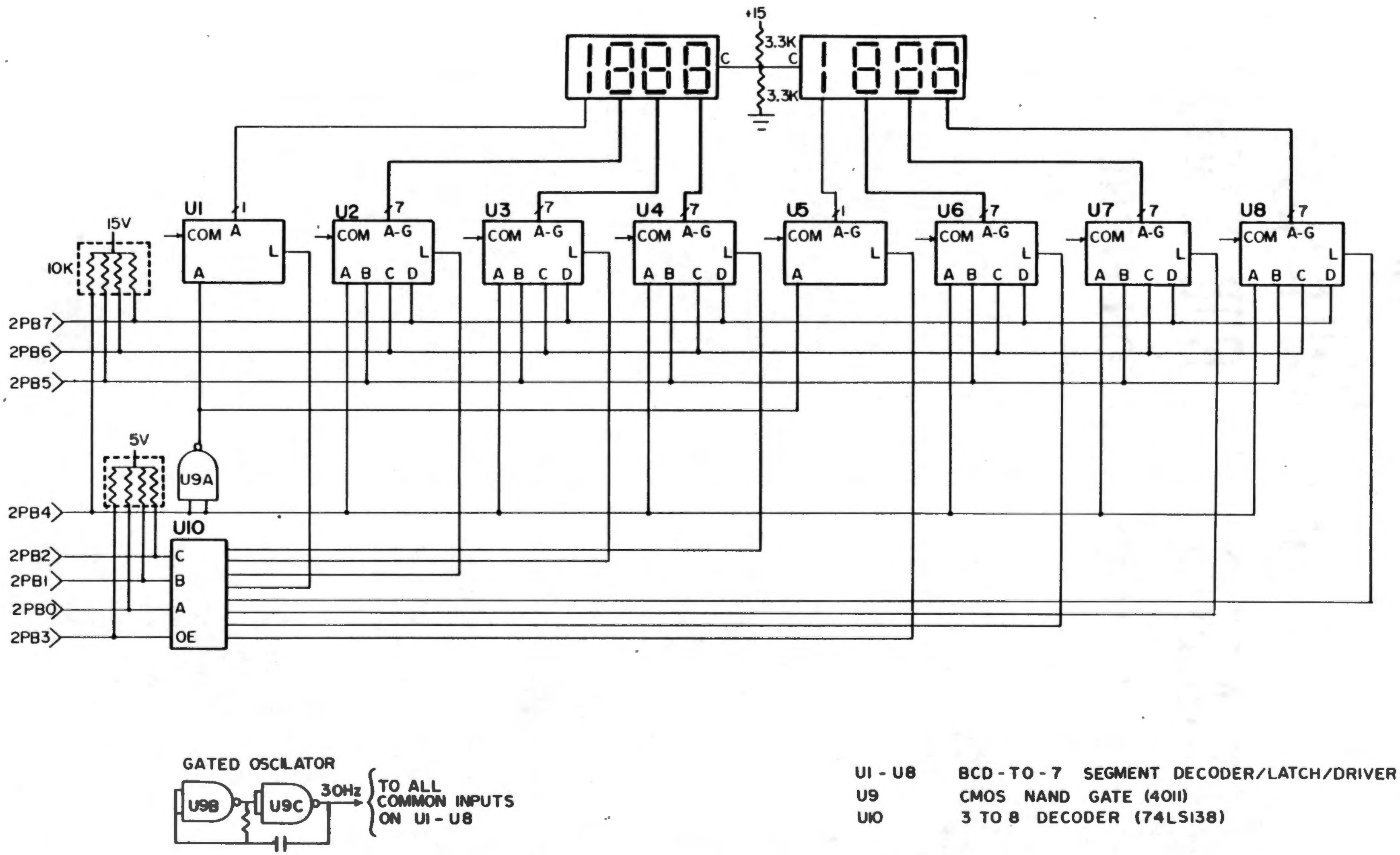


Figure 17. Schematic of LCD display unit with associated components required to interface with a PIA.

through a voltage divider. Hence, the signal to the LCD segments was ± 7.5 volts AC.

A special chip manufactured by LSI, Inc. (LS7100) was used to convert BCD data from a PIA to 7-segment code, latch the data, and drive a dynamic LCD display. Figure 17 shows the display schematic of the interface between the PIA and the displays.

The built-in latch allowed the microprocessor to output one BCD code for all 7 digits. Each BCD-to-7-segment decoder had its own LOAD line that was activated by a 3-to-8 decoder, thereby reducing the number of I/O lines required.

Key Pad Interface

A 3 X 4 matrix type key pad, manufactured by Grayhill, was interfaced to a PIA according to Figure 18. The PIA activated one row at a time while the columns were read as inputs. The microprocessor scanned one row at a time looking for a pressed key. Seven of these keys were assigned to perform various tasks as described in Chapter V.

Operator Indicator Lights

Light-emitting diodes (LED) were mounted on the control unit to inform the operator of the status of the program or error conditions which might occur. The first seven of these LED's were used during the setup procedure discussed below. Other indicators included direction of penetrometer shaft movement, direction of carriage trolley travel, mode of operation, and error conditions. The LED's were made active one at a time by a 4-to-16 multiplexer

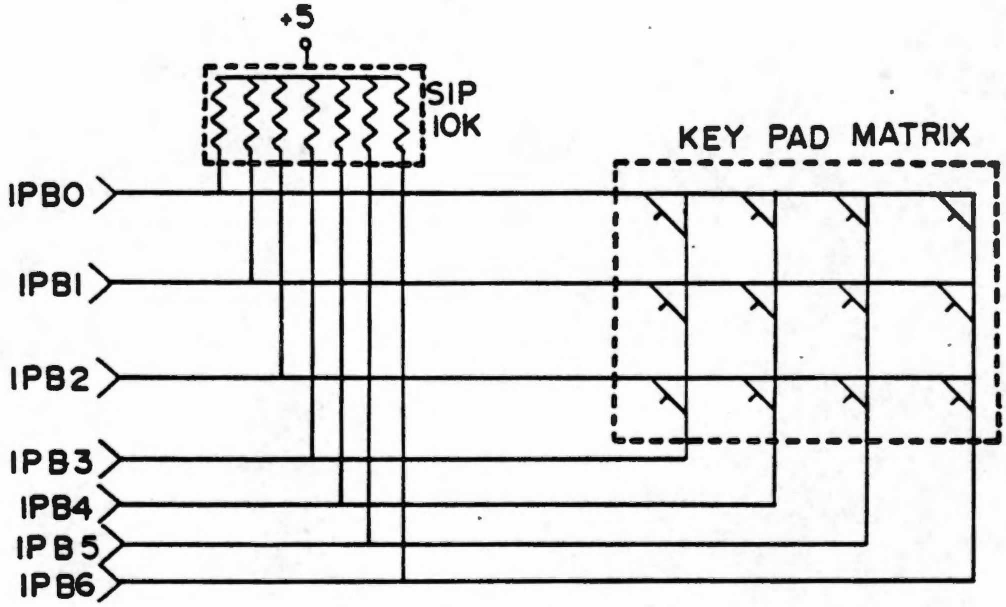


Figure 18. Interface of the key pad matrix to a PIA.

(74LS151). This permitted 15 lines of control from four I/O lines (see Figure 19).

D. Cabinet, Signal Cables, and Power Converter

The control module and interface board shown in Figure 16 and 20, respectively, were housed in an aluminum cabinet (Figure 21). This cabinet provided protection and support for the boards along with connections for communication. Edge card guides and edge card connectors for the S-100 bus boards were also provided inside the structure.

The control board was inserted in the bottom slot while the interface board was placed in the top (see Figure 22). Communication between the two boards was provided by the 40- and 26-pin ribbon cable assemblies. The key pad matrix and LED's were elevated on the interface module to protrude through the aluminum face plate. The LCD displays were placed under openings cut in the top of the face plate for data viewing. A sheet of clear plastic was placed on the bottom side of the plate under the openings for the displays. This helped keep dust from settling inside the cabinet housing.

The digital and analog signals from the transducers and digital signal to the power interface were transmitted via two 20-conductor shielded cables. Connectors (DB25) were mounted on the back of the control unit cabinet and on the penetrometer signal box. The cable, 3 m in length, allowed the operator to hold the control unit in his lap while operating the tractor.

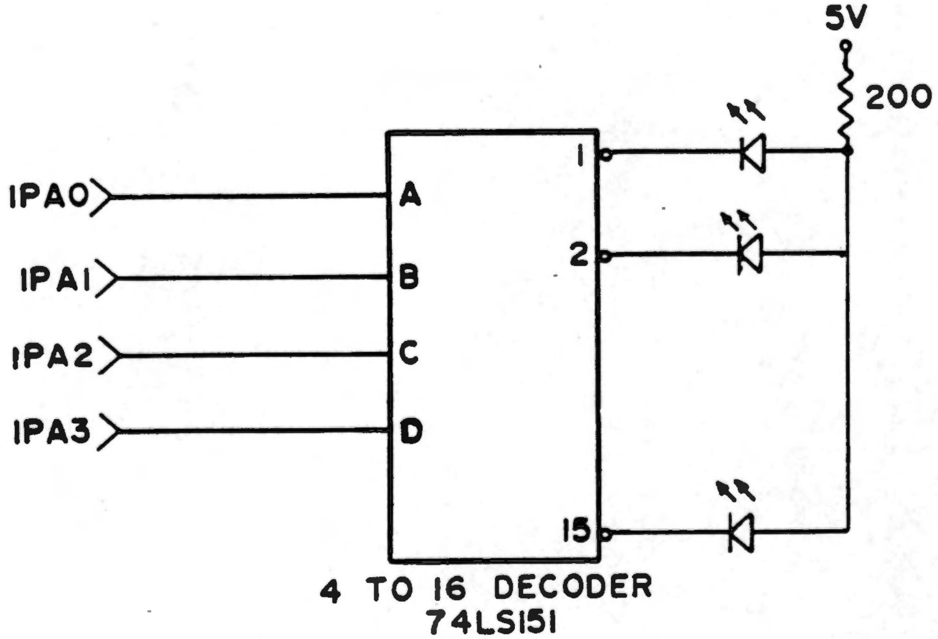


Figure 19. Interface of the LED indicators to a PIA.

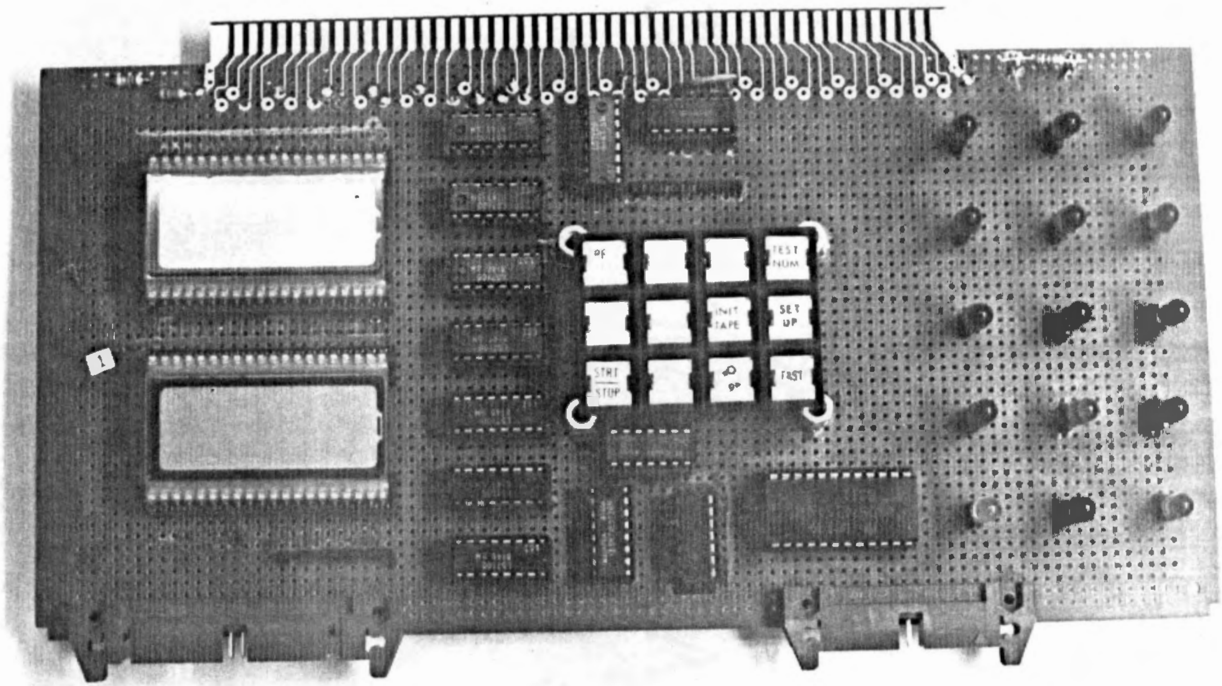


Figure 20. Interface module for input/output connections associated with displaying data and inputting variables.

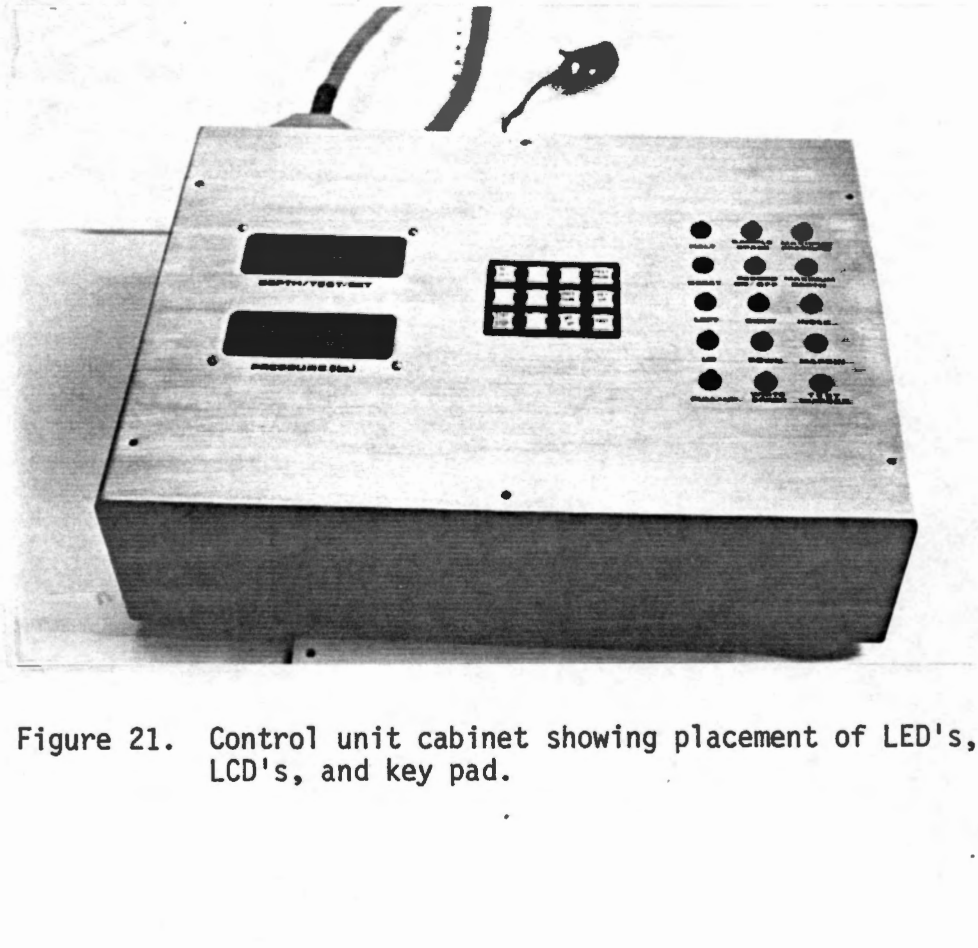


Figure 21. Control unit cabinet showing placement of LED's, LCD's, and key pad.

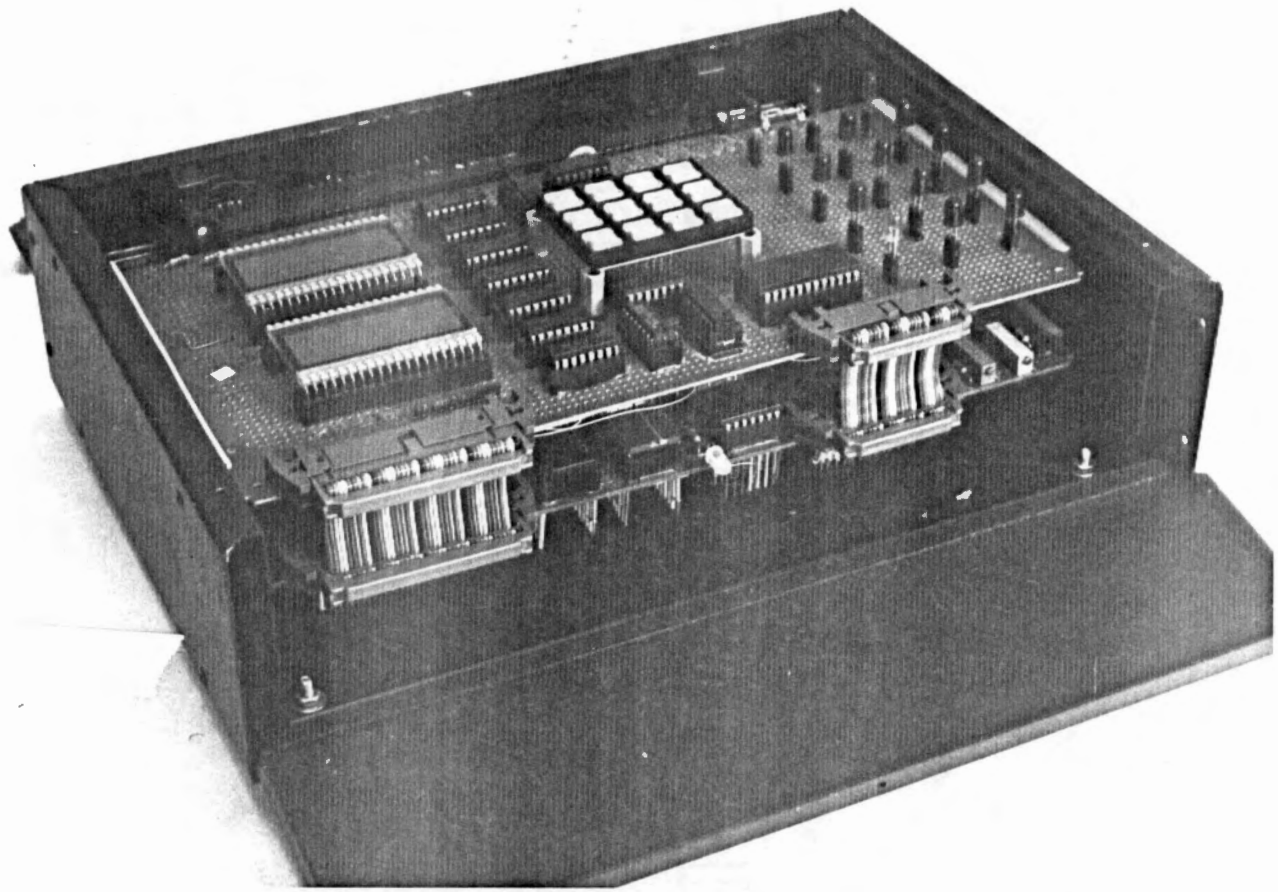


Figure 22. Control unit cabinet showing placement of control module and interface board.

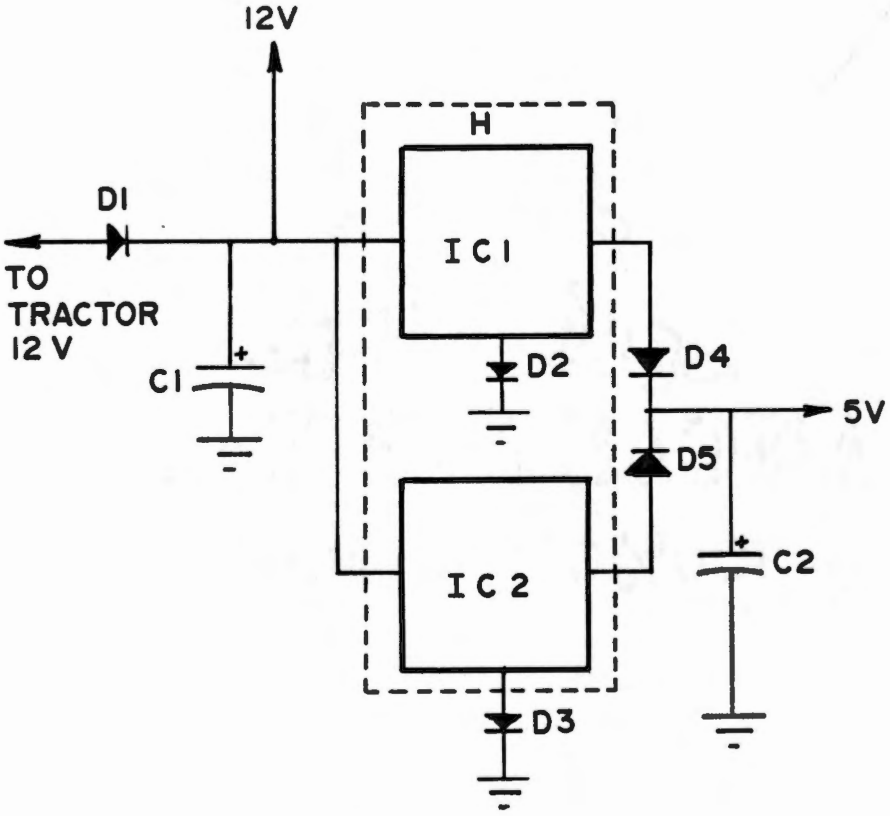
A power supply of + 5 and + 12 volts was needed to operate the control unit and tape drive. A power regulator was constructed to allow 12 volts unregulated from the tractor to be changed to 5 and 12 volts regulated. The power was regulated as shown by Figure 23. A heat sink (100 W) was provided on top of the regulator housing to adequately dissipate the heat produced by the power transistors during the regulation. Figure 24 shows the construction power regulator as constructed.

E. Power Interface

The outputs from the control unit were low-level TTL signals capable of driving one standard TTL load (source 400 uA or sink 16 mA). These signals controlled the operation and direction of a 30-amp DC motor for carriage positioning and electrically activated 3.5-amp solenoid valves for cylinder movement. Refer to Figure 25 for a schematic of the power interface.

The signals were first sent through an optical isolator. This integrated circuit (IC) served two functions: first, to isolate the digital signal between the control unit and the power devices; and, second, to change the signal from TTL (5VDC) to a 12-volt logic level. The output from this IC was then sent through a PNP power darlington transistor to provide the needed current to drive the solenoids on the hydraulic valves and to energize the coil on the relays that operated the motor.

A high EMF was created from the inductive load when the relays and solenoids were deactivated. This surge of reverse



COMPONENT IDENTIFICATION

C1	2200μf CAPACITOR, 35 VDC
C2	100μf CAPACITOR, 10 VDC
D1, D4, D5	1μ5400 DIODE, 3 AMP, 50 PIV
D2, D3	1μ4001 DIODE, 1 AMP, 50 PIV
IC1, IC2	323 (TO-3) 5 VOLT REGULATOR, 3 AMP
H	HEAT SINK, 100 W

Figure 23. Power supply to filter and regulate the voltage required to drive the control and tape unit.

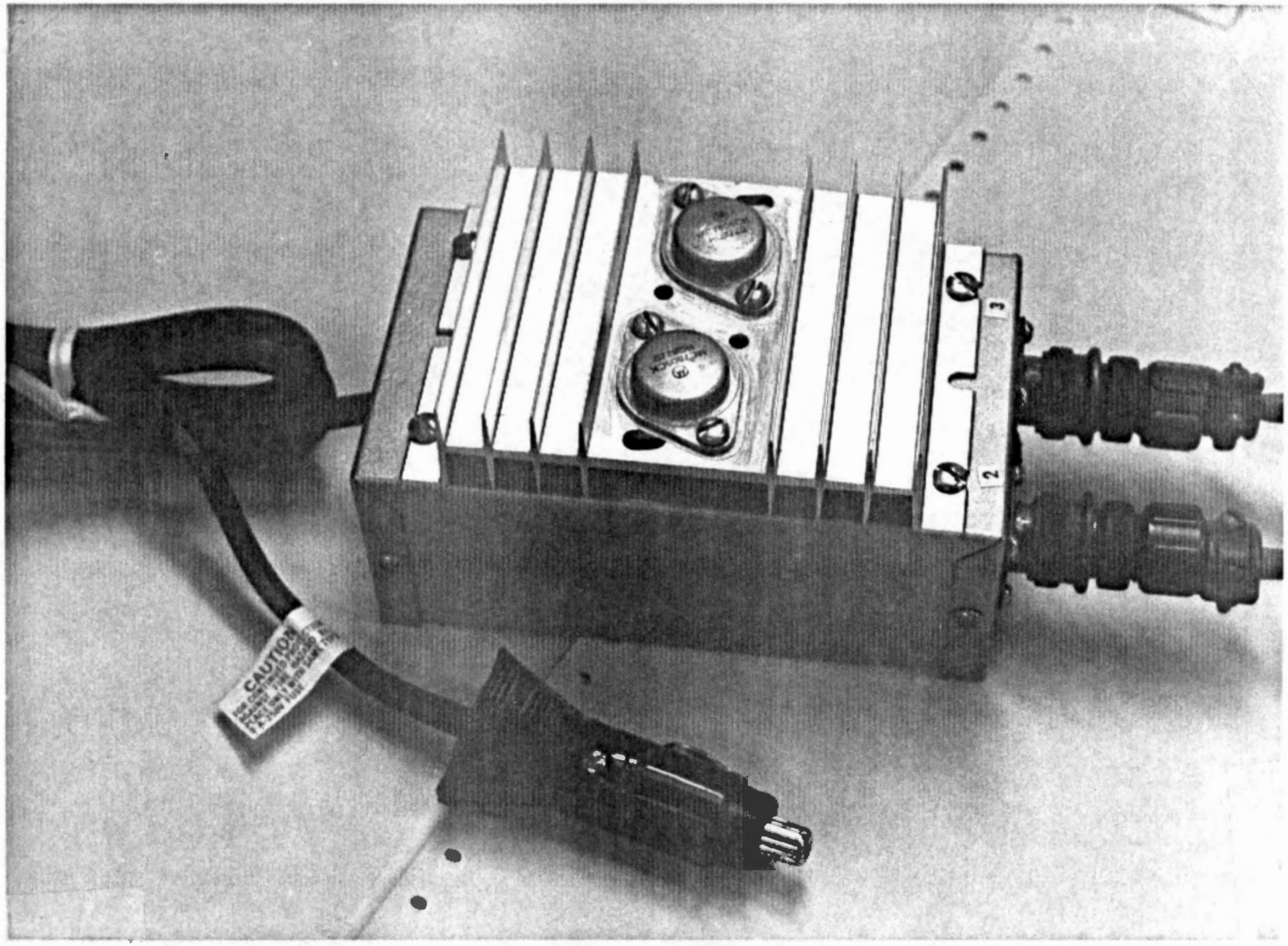


Figure 24. Power regulator housing and connectors.

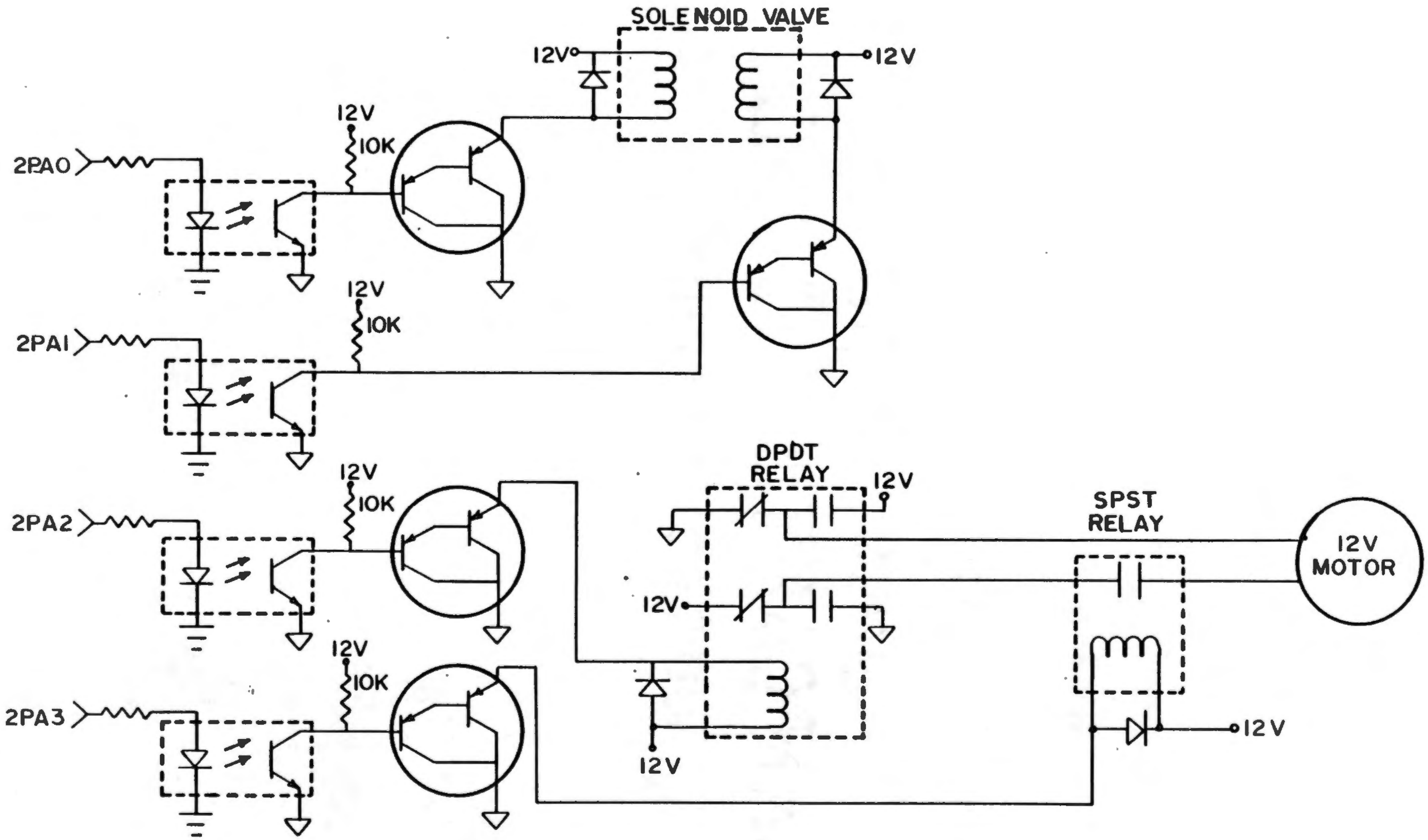


Figure 25. **Interface from TTL to power switching.**

current exceeded the maximum voltage potential allowed between the collectors and emitters on the power transistors. This problem was greatly reduced by paralleling a diode with the solenoid valve and relay coils in reverse direction of normal current flow.

F. Data Storage Unit

A DEC TU58 cassette tape drive system was used for data storage. The tape drive unit was designed for dual drives and a single control interface board. Each TU58 cassette tape had a storage capacity of 480 blocks comprised of four 120-block tracks. The tape drive system was interfaced with the control unit via a serial communication link with the tape drive control board. The tape drive and control unit were wired to communicate at a speed of 9600 baud. However, other transmission speeds were available by changing wire-wrap pins on the tape drive and control unit. Twelve volts and 5 volts were supplied by the power regulator discussed previously (see Figure 26 for a photograph of the DEC TU58).

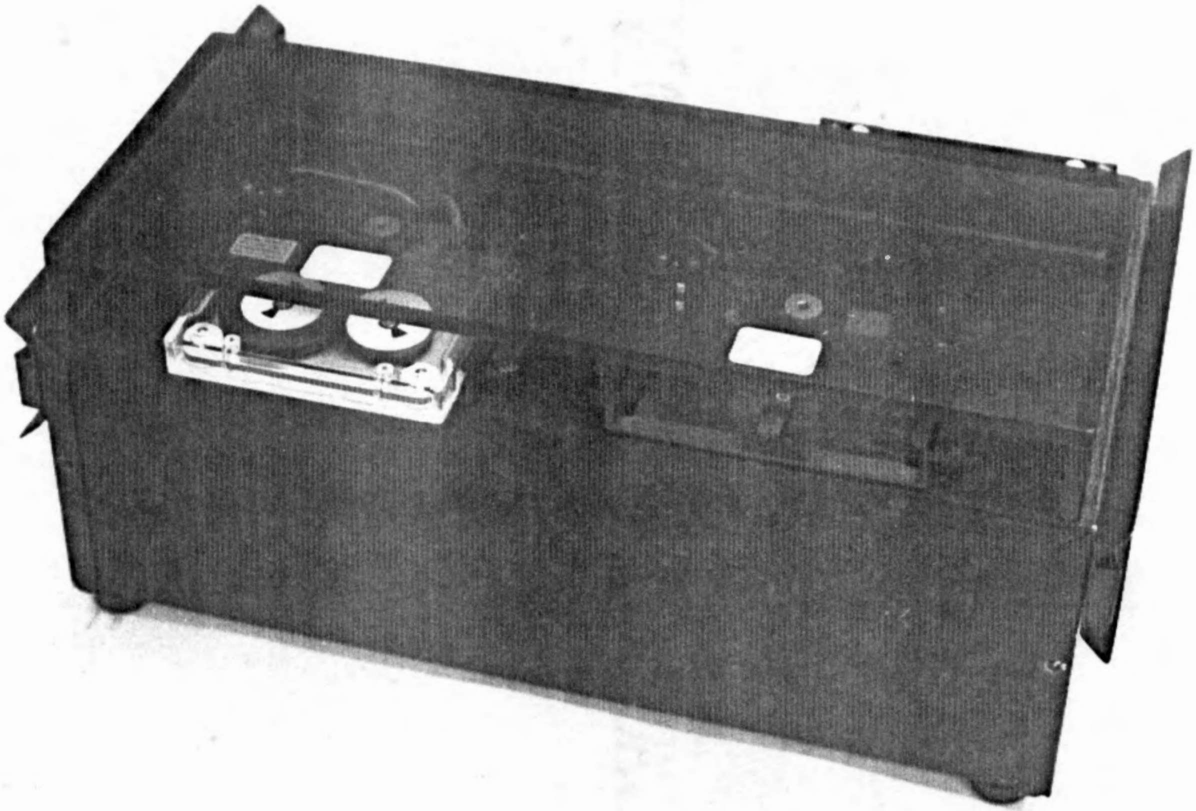


Figure 26. DEC TU58 cassette tape drive for mass storage of penetrometer data.

CHAPTER V

FIRMWARE FOR OPERATION OF PENETROMETER

A. Introduction

The program which controlled movement and data collection of the penetrometer was written in assembly language compatible with the Motorola 6802 MPU. The address locations correspond to the address scheme shown in Figure 14, page 36. This program was in turn burned into an EPROM using a Bay Technical EPROM Programmer. The program, after being transferred to EPROM, was referred to as "firmware." The Appendices have the actual assembly language version of the program used by the control unit.

B. Power-up Procedures

To activate the program stored in EPROM, a power-up procedure was executed. When power was initially supplied to the control unit, the starting address of the first instruction was transferred from within the EPROM, at memory location FFFE(MSB) and FFFF(LSB), to the MPU. This initial data transfer, from EPROM to the MPU, was initiated when the "reset" pin on the MPU was held low for a minimum time of 8 clock cycles or 8.9 microseconds. This delay was achieved by placing an RC network on the reset line. When power was applied to the control unit, the clock circuit started transmitting pulses immediately which the capacitor was being charged through the resistor. This, in turn, kept the "reset" line from going high for more than 9 microseconds, allowing the MPU to process the clock cycles needed before the "reset" line went high.

The first instruction was to load the stack pointer (LDS) with the appropriate location in memory (RAM). This starting address for the stack, and the 1K of RAM preceding that address, were dedicated for MPU use only.

Several other instructions and routines were executed immediately after the power-up procedure. The parallel I/O ports were initialized by the subroutine "INIT." This routine set the ports on the PIA's as inputs or outputs depending upon the desired function. This immediate action allowed the control unit to turn off all output that controlled the moving mechanisms and provide the operator with prompt visual indication, from the displays, that the MPU was up and running.

Before certain routines could be executed, data had to be transferred from EPROM to RAM. This task was performed by the subroutine "MOVE" which took the starting and ending addresses ("SOF" and "EOF") of the EPROM file and transferred them to RAM beginning at a memory location defined as the start of a new file ("SONF").

C. Monitor Program

The "MON" program of the control unit acted primarily as a monitor program. The key pad was continually scanned looking for a pressed key. In this mode of operation, the operator could select the setup, tape initialization, test number, or data collection routine. The pull-up LED was activated during this mode to signal the operator that he was free to move the penetrometer at that time (see Figure 27).

D. Operator Callable Routines

Prior to data collection, a routine called "SETUP" could be initiated. This allowed the operator to select or field-program seven variables into the program that controlled various operating parameters. If the operator chose not to run the setup procedure, default variables were automatically loaded into the main program during power up. Maximum pressure and maximum depth were the first two of these default variables. As the penetrometer moved downward into the soil, the MPU continually read the pressure and depth and compared it to the maximum value currently stored in RAM. The fourth parameter was "margin," which designated the distance inward from the end of the track where penetrometer readings would start. This allowed the operator to work within the desired area of the track. The fifth and sixth default parameters referred to the data recording routine. The option to display only allowed the operator to view the data on the displays and not use tape space. The other option referred to the sampling (depth) interval. This option permitted sampling intervals from 0.32 cm to the maximum pressure or maximum depth, whichever occurred first. During the data collection routine, the test numbers were automatically incremented. However, since some situations could require the operator to select a specified starting test number, this programming option was also included as the seventh and final parameter.

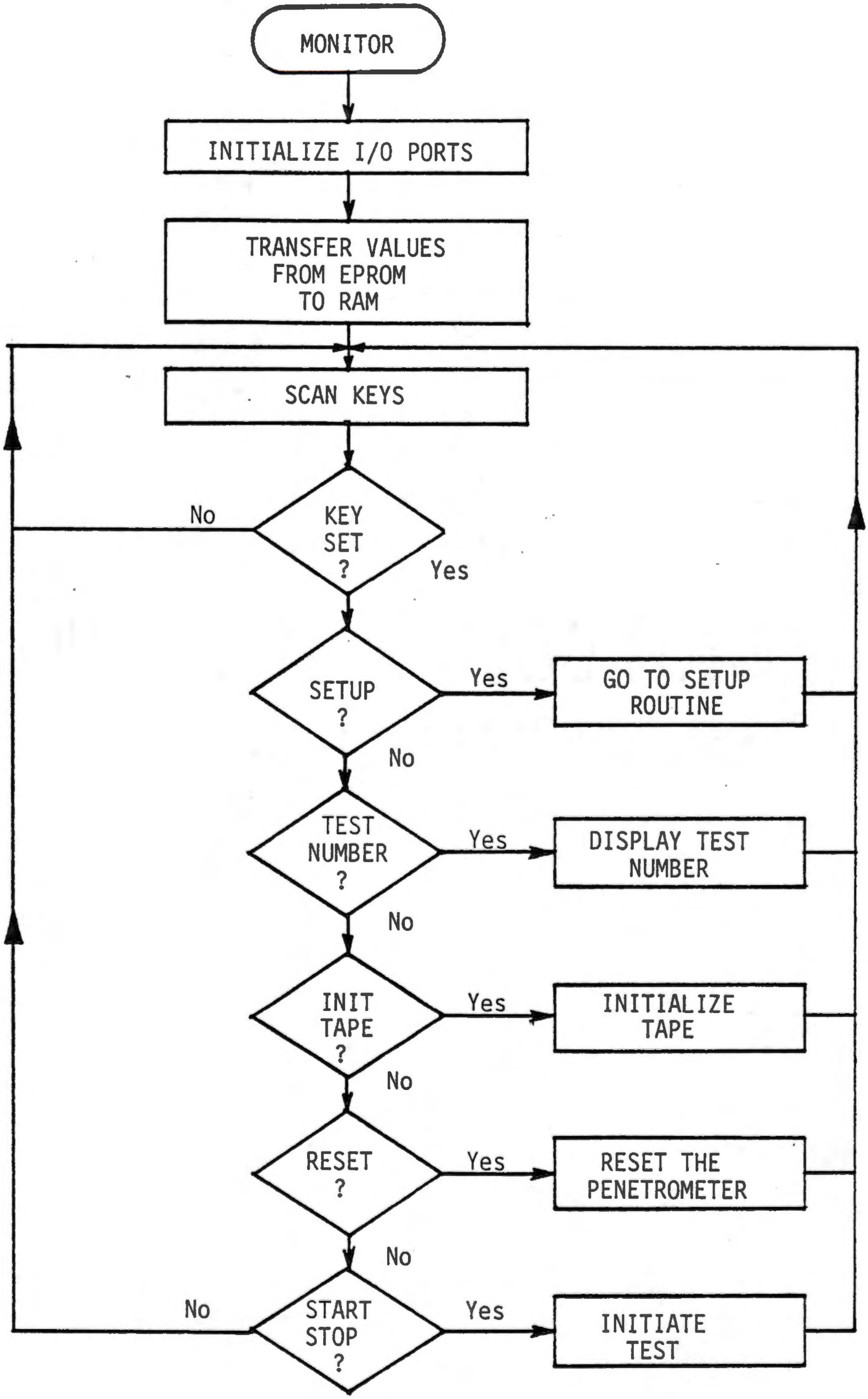


Figure 27. Flowchart for the monitor and power-up routine.

This setup routine was initiated by pressing the "setup" key on the key pad. Upon pressing the key, the LED corresponding to the first parameter would light. If the operator chose to change this value, the "0-9" key could be pressed allowing the value on the display and in memory to be incremented at a rate of two digits per second. The "fast" key allowed this incrementation to occur much faster. Each parameter had an upper limit which was stored in EPROM. If the operator attempted to increment the value over the upper limit, it would be returned to 0 and wrap around. If the operator chose to retain the default value loaded into memory upon power-up, he could move to the next parameter by pressing the setup key again. This procedure was repeated for all seven parameters. A final press of the setup key returned the program counter to the main program. Refer to Figure 28 for a flowchart.

A "RESET" routine could be initiated by pressing the "reset" key when the MPU was operating in the monitor mode. This was one of the first routines the operator would normally choose to execute. Upon execution, the control unit would signal the power interface to activate the solenoid valve to retract the cylinder. Next, the control unit moved the carriage assembly to the start of the track (see Figure 29). At this time, the penetrometer was ready for the start of the first test.

The "START" routine was initiated by the operator by pressing the "start/stop" key when the testing was ready. This routine positioned the penetrometer along the track at the predetermined location where a penetration was to be made. This routine also

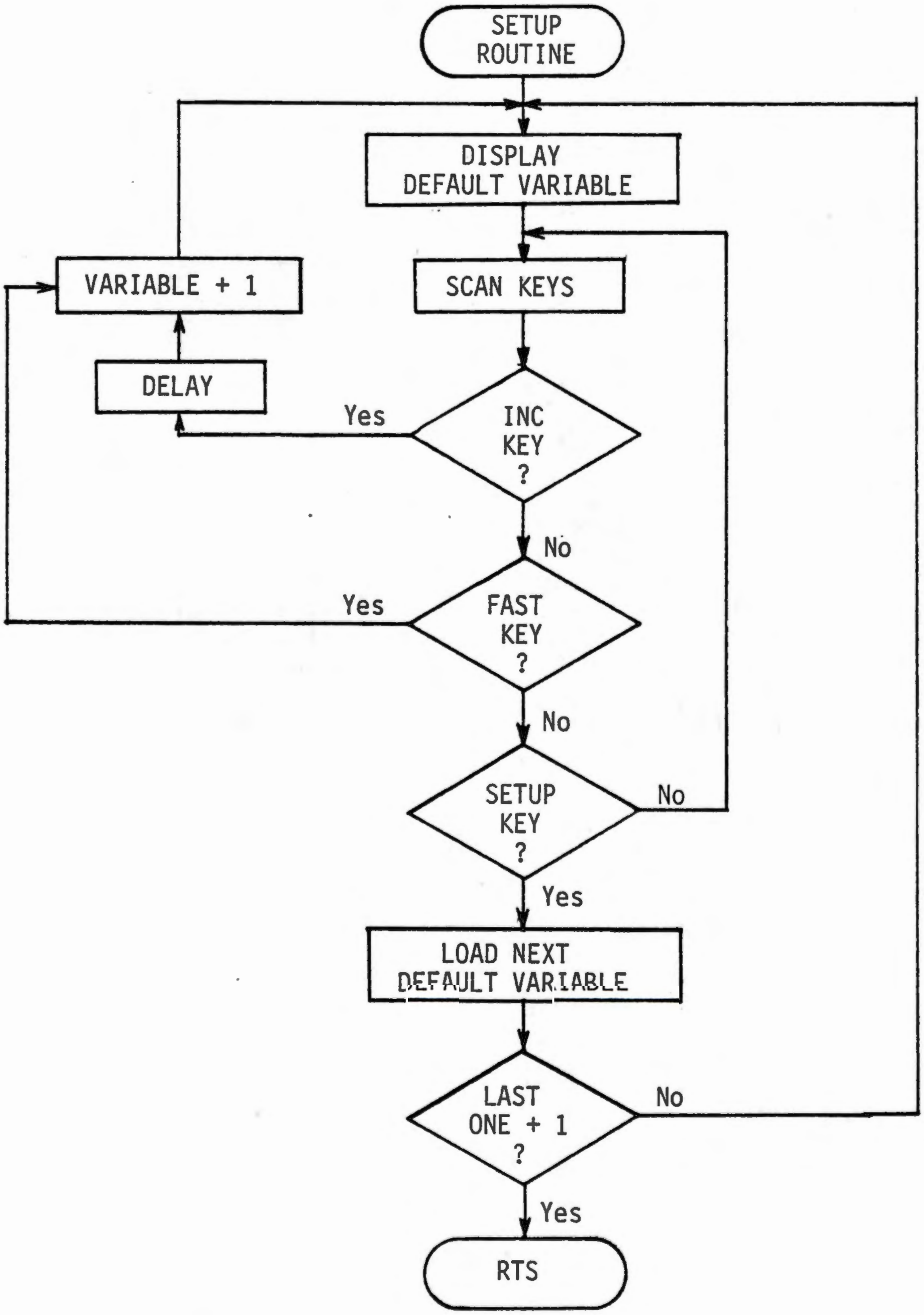


Figure 28. Flowchart for the setup routine.

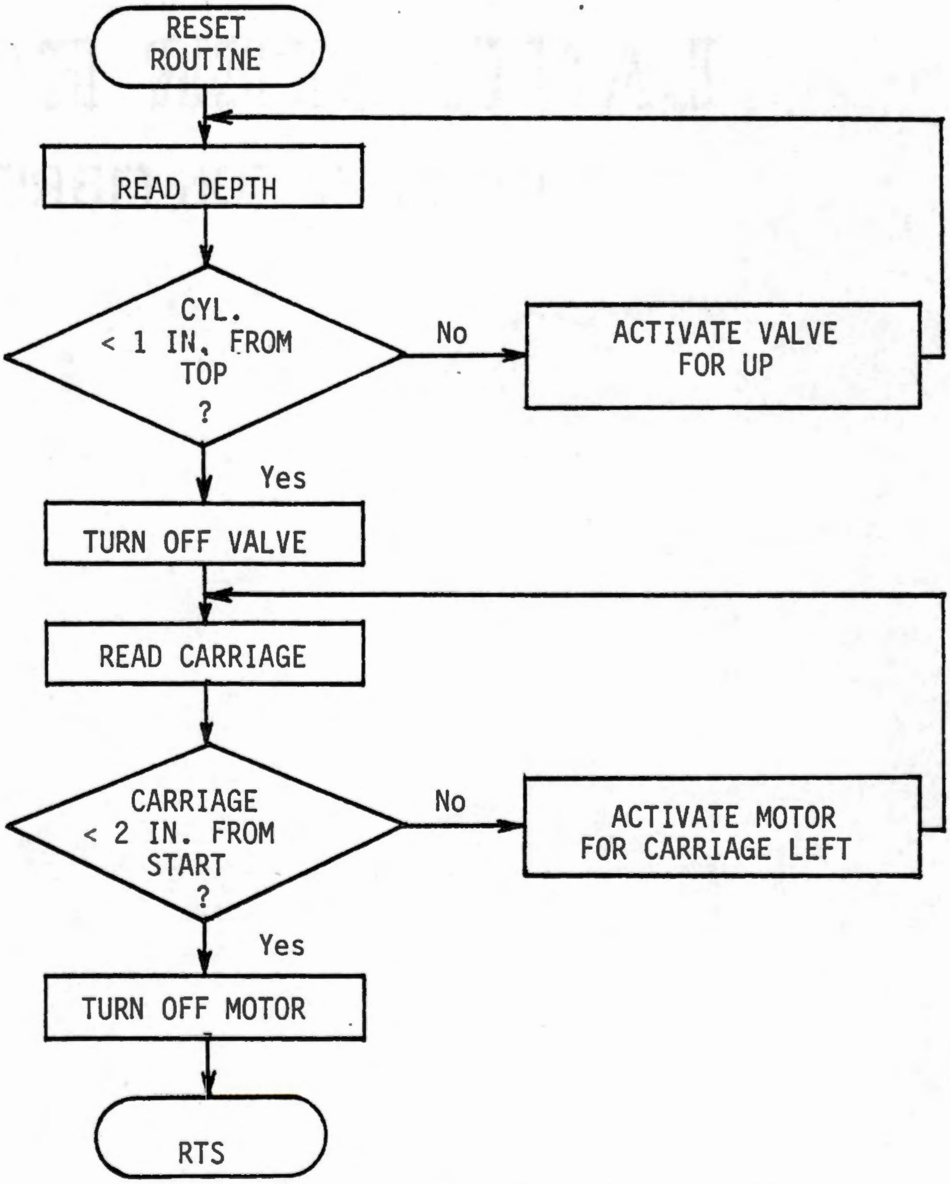


Figure 29. Flowchart for the reset routine.

called the subroutine "SAMPLE" which controlled the penetration and data collection. When the operator was ready to take penetrometer readings, this procedure was initiated by pressing the "start/stop" key (see Figure 30).

Upon entering this routine, the MPU would move the carriage assembly to the right starting margin. The first penetration would be made there. After the cylinder retracted, the present location of the carriage plus the "index" were added forming the "next stop." The MPU would activate the carriage for right movement until the "next stop" location was reached. At this point, the second penetration was made. Again the "next stop" would be calculated. However, before moving the carriage, the present location was subtracted from the end margin. If the difference was greater than or equal to 0, the carriage was moved to the "next stop." This routine of calculating the "next stop" and checking for the "end margin" was continued until the difference was negative.

When the end point was reached, the MPU would signal the power controller to return the carriage to the starting margin. This was the location of the next sample provided the operator had not changed the "margin." The test number was incremented and returned the program counter to the main program.

The operator could have stopped the carriage drive or hydraulic cylinder movement when operating in the "START" or "SAMPLE" mode by pressing the "start/stop" key and thus entering the "HALT" routine. This routine allowed the MPU to turn off all outputs and wait for a second press of the "start/stop" key required to return

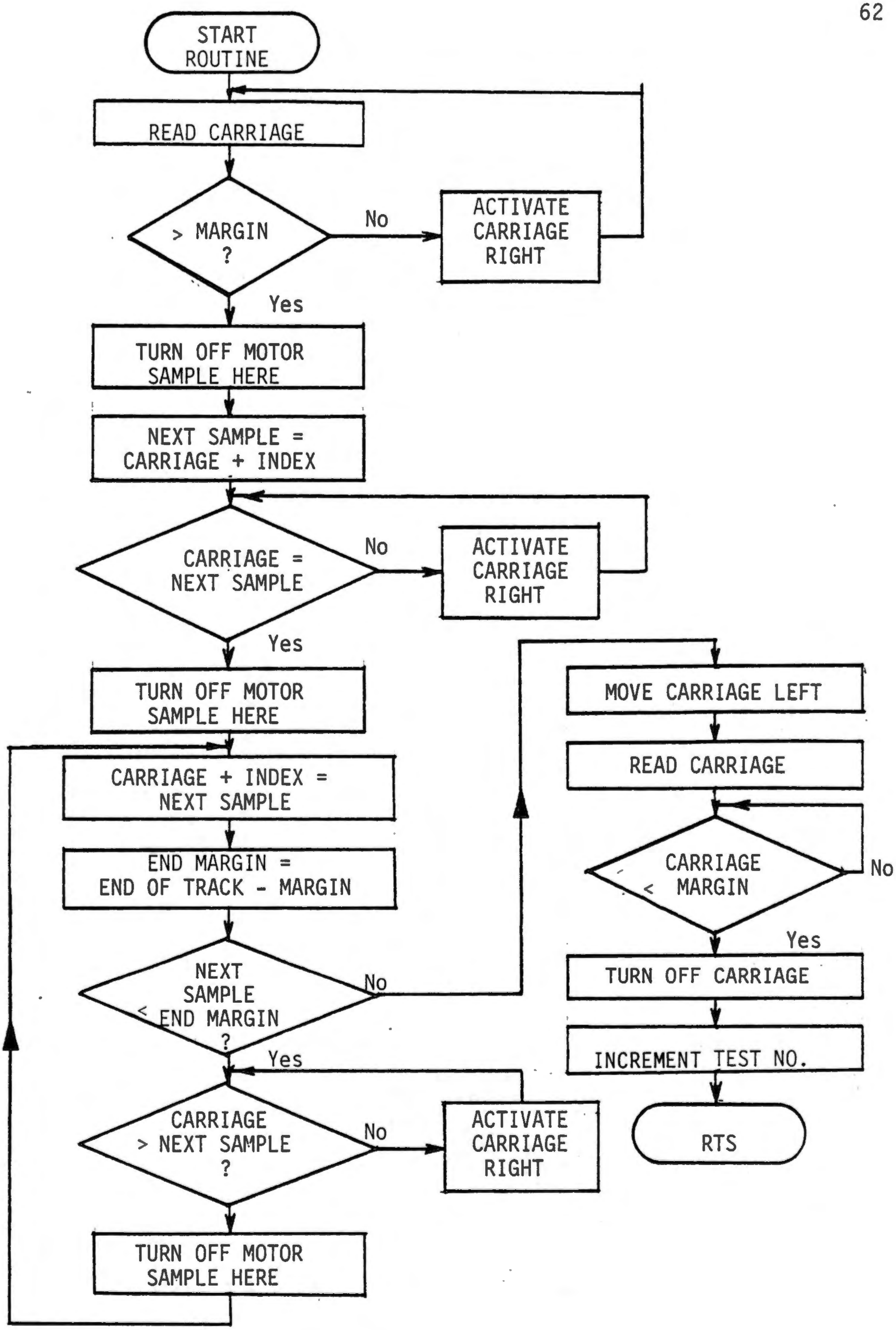


Figure 30. Flowchart for the start routine.

the program counter to the location in the program from which it originally came. The current test number was displayed when the operator pressed the "test no." key while the MPU was operating in the monitor mode.

D. Subroutines

Various subroutines were written for access by operator callable routines and other subroutines. Many of these subroutines were called repeatedly by one or more routines.

The "DISPLAY" routine was used in conjunction with the LCD display unit to visually display data. This routine communicated with port "B" of PIA #1 (refer to Figure 17, page 40). The upper 4-bits of the port outputted a BCD (Binary-Coded-Decimal) value to the display bus while the lower 3-bits selected the digit to receive the data. After the datum was placed on the display data bus and the appropriate digit was selected to receive the data, bit 4 was toggled to latch the digit into the BCD-to-7 segment decoder/latch.

The datum (0-1999) to be displayed was passed to the display subroutine in memory locations labeled as "UNTTEN" (units and tens) and "HNDTHD" (hundreds and thousands). The "B" Register was used to select the display unit, "0" for the top and "4" for the bottom, to receive the data.

The values outputted by the MPU were binary; therefore, the BCD equivalent was generated by the "CONV" (convert) subroutine. This routine was adapted from a program written by Ron Bishop (1979). It took the 16-bit binary value stored in "MSB" and "LSB" and converted it to a BCD code which was stored in "UNTTEN" and "HNDTHD." Refer to Appendix C for program listing and additional comments.

The "KEY" subroutine allowed the operator to select various routines, change the variable which affected the data collection, and control operation of the penetrometer. The 12-key keypad matrix was connected to the "B" port on PIA#2. The four columns were read by the processor as inputs and the three rows were activated by the PIA as outputs. This "KEY" subroutine activated one row at a time while reading the columns for a pressed key. The pressed key number is stored in RAM under the variable name "Keyset" (see Figure 31). This was illustrated in Figure 18, page 42. The following list gives a brief description of the function of each key:

Start/Stop	Initiates a test. Also used to stop and restart the program during a test.
Reset	Moves penetrometer trolley back to designated starting point.
Setup	Enter setup mode for field programming.
Init Tape	Formats a magnetic tape for data storage.
0-9	Used to increment a variable during the setup procedure.
Fast	Used in conjunction with the "0-9" key to speed up the incrementation.
Test No.	Display current test number.

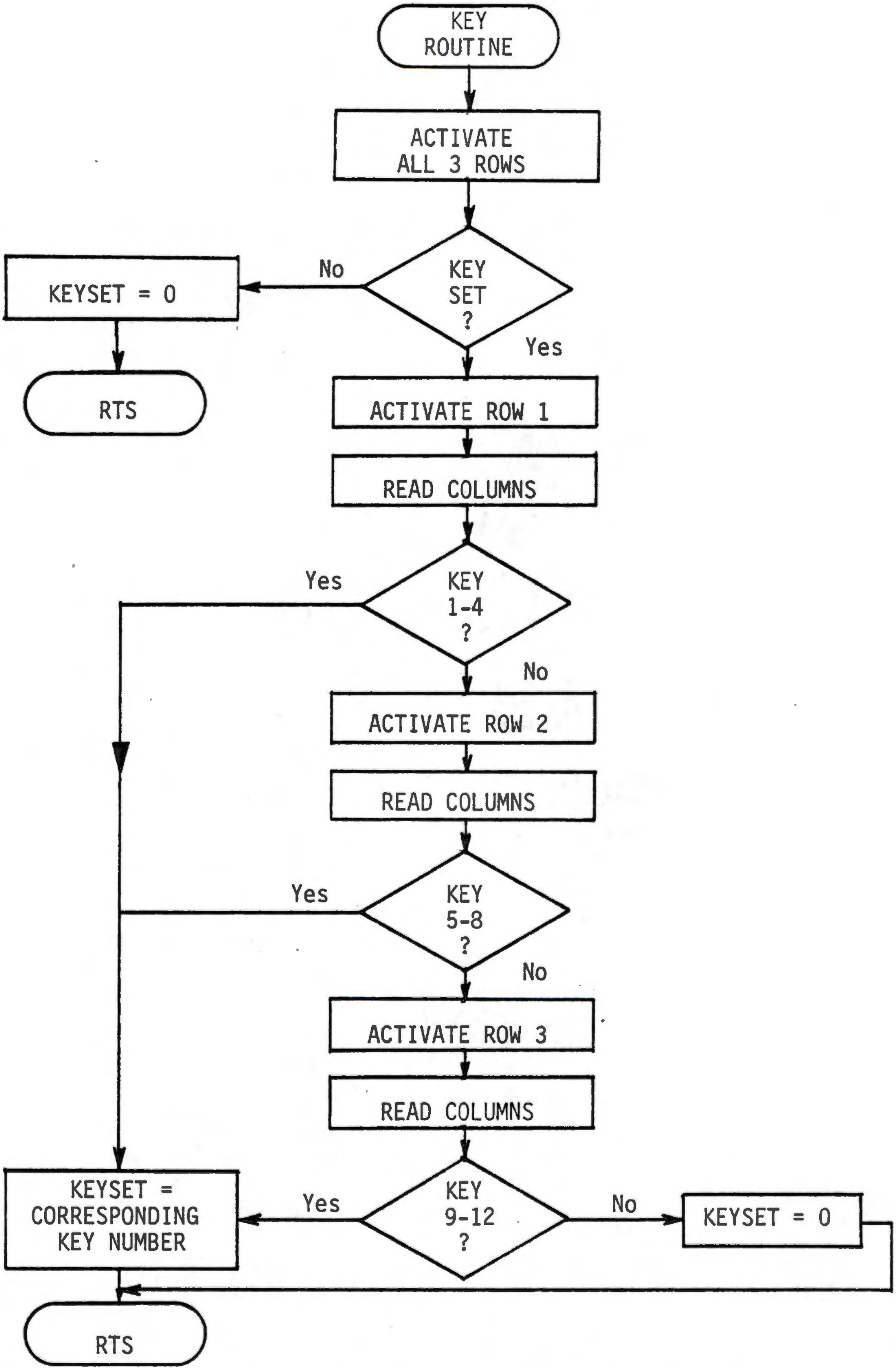


Figure 31. Flowchart for the key subroutine.

Two subroutines, "LDELAY" (long delay) and "SDELAY" (short delay), were callable by several routines throughout the program. The long delay allowed the program to pause for a period of 0.5 seconds. Likewise, the short delay performed the same function for a period of 35 milliseconds.

The output from the depth optical position encoder produced an 8-bit digital signal what was interfaced with an 8-bit D-type latch. Before that datum was placed on the data bus, the R/W line was pulsed low while the chip-select (CS) pin on the latch was activated. The "DEPTH" subroutine accomplished this by first executing a store accumulator (STA) command at the address location assigned to the latch. This statement, upon execution, performed the task required to latch the data from the encoder and place that value on the data bus. This routine also scaled the depth value for ease in data handling by the MPU. A similar subroutine was used to read the 10-bit value from the carriage positioner ("CARRY"). The LSB (0-7 bits) were loaded into one 8-bit latch while the 2 MSB's (8-9 bits) were latched by a similar latch.

The "ADC" (Analog-to-Digital Converter) subroutine was used to read the binary equivalent of the analog voltage from the pressure transducer. To start a data conversion, the ADC was addressed and a STA command was executed by the MPU. This sequence of events held the CS line low while pulsing the R/C pin on the ADC and starting a conversion. The "ADC" subroutine called a short delay (35 milli-seconds) before reading the converted value. This delay was needed since the conversion time of the ADC was longer than the time required by the MPU to initiate a conversion and read the results. Since the resolution of the ADC was 12 bits and the data bus was only 8 bits, the lower byte was read by one address and the upper nibble was read by another.

Another useful subroutine was "SUB" (subtract) which subtracted the difference between two 16-bit values. This routine subtracted the 16-bit value stored in "DIFMSB" and "DIFLSB" from the 16-bit value stored in "LSB" and "MSB." The result was stored in "MSB" and "LSB."

A subroutine called "SAMPLE" was used to control the operation of the hydraulic cylinder, take data readings, call the data storage routine, and display force-depth measurements on the LCD units. This subroutine could only be entered through the routine "START" which controlled the total operation of the penetrometer.

When the subroutine "SAMPLE" was initiated, the hydraulic cylinder moved the penetrometer shaft down through the soil profile. As the penetrometer shaft moved into the soil, the MPU continually read the depth and pressure and compared the measured values to maximum values stored in RAM under the variable names "MDEPTH" (Maximum Depth) and "MPRESS" (Maximum Pressure). When one of these readings exceeded its maximum limit, the MPU instructed the solenoid to reverse the direction of flow and return the cylinder to the top position.

The MPU also calculated the next force-depth reading to be taken during the time of penetration. The MPU took the present depth reading and added the sample space ("SAMP") value to it. When the penetrometer tip reached the "next sample" depth, the depth and force measurements were made and stored in RAM. The display was also refreshed at this time. This procedure occurred repeatedly until the maximum depth or maximum pressure was reached.

F. TU58 Handler Routines

The following routines were written to allow communication between the control unit and the TU58 cassette tape controller. The data stored in RAM during a test were transferred to tape for a more permanent media. These data could be transferred from the cassette tape to a microcomputer (PDP-11/03) for data analysis.

The "WRITE" routine transferred data stored in RAM to TU58 cassette tapes. This routine was executed from the "SAMPLE" routine after each penetration was completed. The "WRITE" routine called four other subroutines which aided in communication between the ACIA in the control unit and the MPU in the tape drive. The first of these routines was "BREAK." This routine established communication with the TU58 controller. The routines "COM" and "TRANS" were used by the ACIA to output commands stored in EPROM which instructed the MPU of the TU58. These instructions implemented the TU58 instruction sets. The ACIA then waited for a reply sent from the TU58 controller by calling the subroutine "MAIL." These commands were written for an octal-based, 16-bit operating system. Therefore, the command strings were converted to hexidecimal and adapted to the 8-bit processor in the control unit (refer to DEC, 1978).

A routine for data tape initialization was designed into the operating system. This routine allowed the operator to erase and format tapes in the field preparing it for data storage. The "ITAPE" routine was entered from the main program by pressing the "init tape" key. Only the tape in drive 0 was initialized. To reduce the likelihood of erasing a tape by mistake, an additional instruction set was added. Upon pressing the "init tape" key, the "record" LED would flash indicating to the operator that the "ITAPE" routine was ready to run. If the operator pressed the key by mistake or wished to retract his initial command, the change was accomplished by pressing any key except for the "init tape" key. This action would void the entrance to the initialization routine and return the program counter to the main program. However, if tape initialization was desired, a second press of the "init tape" key would start the process. The LED would stop flashing and remain activated until the tape was initialized, after which time the program counter would also return to the main program.

Before the first data of any operating session could be stored on tape, location of the first available data block was necessary. A routine called "FIND" was executed through the "START" routine which was run at the beginning of the program. "FIND" would look on drive 0 for the first free data block. If none were found, drive 1 would be searched. No available space on the tape would be indicated by activating the "write error" LED.

CHAPTER VI

EVALUATION AND SUMMARY

A. Evaluation

The penetrometer was operated in a laboratory environment to verify the functionality of the completed unit. The mechanical track and frame work structure performed as designed. The carriage drive was very noisy but was satisfactory for the designed purpose. The cable had a slight tendency to slip on the drive pulley, but the slip was not sufficient to alter performance of the penetrometer.

A pressure reading of 10 to 20 psi was read by the control unit under a zero force test. However, this value could be subtracted from the actual data.

The depth of penetration measurement resulted in a repeatable value of only 8 bits. The two least significant bits of the 10-bit encoder were not repeatable due to backlash in the gear reduction assembly. However, this still provided a depth resolution of 0.28 cm.

The control unit performed as designed. There were few problems associated with the digital electronics. However, the digital output current from the control unit was insufficient to activate the optical isolators. This problem was corrected by buffering the outputs of the PIA controlling the power interface. The buffers increased the current-sourcing capability of the output control lines.

The original LCD displays did not perform satisfactorily. Due to poor quality, the displays performed erratically and were not reliable. Also, the angle of viewing was very narrow, making the displays hard to read. These problems were resolved by replacing the displays with ones of better quality.

B. Summary

A microprocessor-based, tractor-mounted soil cone penetrometer was developed for automating soil strength measurements. The penetrometer was designed to operate over a 4-row (3-m) width to a maximum penetration depth of 61 cm. The force of penetration was applied with a 5.8 X 71 cm hydraulic cylinder. The hydraulic cylinder and associated depth and force measuring sensors were mounted on a carriage assembly which traveled along a 4-m track. The carriage was moved laterally along the track by a cable assembly driven by a DC winch. The penetrometer frame and track were attached to a 4-row tool bar having a category II, 3-point hitch attachment.

The depth of penetration and carriage position sensors were 10-bit optical shaft encoders which provided a digital output to a control unit. The force of penetration was calculated by measuring hydraulic pressure. A pressure transducer/amplifier provided an analog output proportional to pressure.

A microprocessor-based control unit activated all moving mechanisms and automatically recorded data on magnetic tapes. A DEC TU58 tape drive unit was used to store the depth versus force of penetration measurements for soil profile studies. The control unit also provided visual output of the data on Liquid Crystal Displays (LCD). The control unit was field programmable for operation in various modes. Seven parameters which affected operation of the penetrometer could be changed depending upon the application of the test.

The program which instructed the microprocessor (MC6802) to perform the desired task of the control unit was burned into EPROM. Other components that made up the control unit module were 4K RAM, two peripheral interface adapters (PIA), a 12-bit ADC, three 8-bit data latches, and an asynchronous interface adapter (ACIA).

C. Recommendations for Further Improvements

The output from the pressure transducer should be evaluated for frequency of oscillation produced by the tractor hydraulic pump and effects of this oscillation on the actual force measurements. The amplitude and frequency of the oscillation is a direct indication of error in measurement by the ADC. Effects of a damper mechanism and use of an electronic low pass filter should also be considered in resolving this potential problem.

There has been no instance in which the control unit missed a carriage or depth position reading that resulted in mechanical damage. However, the possibility exists of a false reading due to noise on the signal lines or a defective encoder. Microswitches or some other position sensor should be placed at the extreme limits of the depth and carriage positions. Similarly, a pressure switch should be placed on the hydraulic line to override a false signal reading from the pressure transducer. This signal produced by the sensor should interrupt the MPU of the control unit, warning of a malfunction.

A possible program refinement would be to set minimum limits on the operator selectable parameters which control operation of the penetrometer. This would prevent the operator from selecting values which are less than the actual values the penetrometer can discern. At the present time, only default and upper limits are considered in the programming.

BIBLIOGRAPHY

BIBLIOGRAPHY

Anderson, G., J. D. Pidgeon, H. B. Spender and R. Parks. 1980. A new hand-held recording penetrometer for soil studies. Scottish Institute of Agricultural Engineering 31:279-296.

Army, Department of. 1959. Soil trafficability. Technical Bulletin ENG 37.

ASAE. 1980. Soil cone penetrometer, Standard No. 313.1. Agricultural Engineering Yearbook. American Society of Agricultural Engineers, St. Joseph, MI 49085.

Bishop. Ron. 1979. Basic microprocessor and the 6800. Haydon Book Co., Rochelle Park, New Jersey.

Bowen, Henry D. 1976. Correlation of penetrometer cone index with root impedence. ASAE Paper No. 76-1516. American Society of Agricultural Engineers, St. Joseph, MI 49085.

Carter, L. M. 1967. Portable recording penetrometer measures soil strength profiles. Agricultural Engineering Journal 48(6):348.

Carter, L. M. 1969. Integrating penetrometer provides average soil strength. Agricultural Engineering Journal 50(10):618.

Digital Equipment Corporation. 1977. TU58 DECtape II Technical Manual, Mannard, MA.

Gill, William R. and Glen E. VandenBerg. 1968. Soil dynamics in tillage and traction. Agricultural Handbook No. 316, USDA.

Hendrick, J. G. 1969. Recording soil penetrometer. Journal of Agricultural Engineering Research 14(2):183-186.

Howson, D. F. 1977. A recording penetrometer for measuring soil resistance. Journal of Agricultural Engineering Research 22:209-212.

Johnson, C. E., L. L. Jensen, R. L. Schafer and A. C. Baily. 1978. Some soil tool analogs. ASAE Paper No. 78-1037, American Society of Agricultural Engineers, St. Joseph, MI 49085.

Knight, S. J. and D. R. Freitag. 1962. Measurement of soil trafficability characteristics. TRANSACTIONS of the ASAE 5:121-131.

Motorola. 1978. The complete motorola microcomputer data library. Motorola Semiconductor Products Inc., Phoenix, AR 85036.

Motorola. 1976. Linear interface circuits. Motorola Semiconductor Products Inc., Phoenix, AR 85036.

Peatman, John B. 1980. Digital hardware design. McGraw Hill, New York.

Peatman, John B. 1977. Microprocessor-based design. McGraw Hill, New York.

Prather, O. C., J. G. Hendrick and R. L. Schaffer. 1970. An electronic hand-operated recording penetrometer. TRANSACTIONS of the ASAE 13(3):385-386, 390.

Raghaven, G. S. and E. McKeyes. 1977. Study of traction and compaction problems of eastern canadian agricultural soils. Department of Agricultural Engineering, McGill University, Quebec.

Smith, L. A. and W. T. Dumas. 1978. A recording soil penetrometer. TRANSACTIONS of the ASAE 21(1):12-14, 19.

Texas Instruments. 1980. The MOS memory data book for design engineers. Houston, TX.

Texas Instruments. 1977. The line driver and line receiver data book for design engineers. Houston, TX.

Texas Instruments. 1976. The TTL data book for design engineers. Houston, TX.

Wells, L. G. and C. O. Lewis. 1980. Remote electronic acquisition of soil cone index measurements. ASAE Paper No. 80-5034. American Society of Agricultural Engineers, St. Joseph, MI 49085.

Williford, J. R. 1972. Tractor mounted recording penetrometer. ASAE Paper No. 72-518. American Society of Agricultural Engineers, St. Joseph, MI 49085.

APPENDICES

APPENDIX A

PROGRAMS AND VARIABLE LIST USED DURING POWER-UP AND MONITOR MODE

```
*****************************************************
*                                                   *
* MAIN OR MONITOR PROGRAM INCLUDING POWER-UP RESET *
*                                                   *
*****************************************************
;
        ORG $F000        ;STARTING ADDRESS OF PROGRAM
        SEI              ; DISABLE INTERRUPTS
        LDS #$7100       ; START OF STACK
        JSR INIT         ; INITIALIZE PIA'S
BY      LDX #$FF00       ;TRANSFER DATA FROM EPROM
        STX SOOF         ;START OF OLD FILE
        LDX #$0000
        STX SONF         ;START OF NEW FILE
        LDX #$FF20
        STX EOF          ;END OF OLD FILE
        JSR MOV
        LDX #$FF40       ;
        STX SOOF         ;START OF OLD FILE
        LDX #$7040
        STX SONF         ;START OF NEW FILE
        LDX #$FF90
        STX EOF          ;END OF FILE
        JSR MOV
        CLR MSB
        CLR LSB
        JSR CONV
        LDA B #00        ;SELECT TOP DISPLAY
        JSR DISPLAY      ;CLEAR DISPLAY
        LDA B #04        ;SELECT BOTTOM DISPLAY
        JSR DISPLAY      ;CLEAR DISPLAY
;
RIP     JSR KEY
        LDA A #$05
        CMP A KEYSET     ;SETUP KEY?
        BEQ V1
        LDA A #06        ;INIT TAPE KEY ?
        CMP A KEYSET
        BEQ V2
        LDA A #12        ;RESET KEY ?
        CMP A KEYSET
        BEQ V3
        LDA A #04        ;START/STOP KEY ?
        CMP A KEYSET
        BEQ V4
        LDA A #9         ;TEST NUMBER ?
        CMP A KEYSET
        BEQ V5
        LDA A #$FF       ;NUMBER FOR PULL-UP LED
```

```
        STA A LED       ;LITE IT UP
        CLR OUTPUT      ;TURN IT OFF
        BRA RIP
V1      JSR SETUP       ;ENTER THE SETUP MODE
        BRA RIP
V2      JSR BREAK
        JSR ITAPE       ;INITIALISE TAPE
        BRA RIP
V3      JSR RESET       ;RESET PENETROMETER
        JMP RIP
V4      JSR START       ;START A TEST NOW
        JMP RIP
V5      JSR TESTNO      ;DISPLAY CURRENT TEST NO.
        JMP RIP
```

```
LABELS              ADDRESS         DESCRIPTION
-------------------------------------------------------

DDRA1    EQU      $9002     ;DATA DIRECTION REGISTER,PIA#1
DRA1     EQU      $9002     ;DATA REGISTER A-PORT,PIA#1
CRA1     EQU      $9003     ;CONTROL REGISTER A-PORT,PIA#1
KEYS     EQU      $9002     ;OUTPUT TO KEY PAD
;
DDRB1    EQU      $9000     ;DATA DIRECTION REGISTER,PIA#1
DRB1     EQU      $9000     ;DATA REGISTER B-PORT,PIA#1
CRB1     EQU      $9001     ;CONTROL REGISTER B-PORT,PIA#1
LED      EQU      $9000     ;OUTPUT TO LED'S
;
DDRA2    EQU      $1002     ;DATA DIRECTION REGISTER,PIA#2
DRA2     EQU      $1002     ;DATA REGISTER A-PORT,PIA#2
CRA2     EQU      $1003     ;CONTROL REGISTER A-PORT,PIA#2
DISBUF   EQU      $1002     ;DISPLAY OUTPUT
;
DDRB2    EQU      $1000     ;DATA DIRECTION REGISTER,PIA#2
DRB2     EQU      $1000     ;DATA REGISTER B-PORT,PIA#2
CRB2     EQU      $1001     ;CONTROL REGISTER B-PORT,PIA#2
OUTPUT   EQU      $1000     ;OUTPUT TO VALVE AND MOTOR
;
DEFVAR   EQU      $FF00     ;DEFALT VARIABLE LIST-START(EPROM)
NEWVAR   EQU      $0000     ;NEW VARIABLE LIST-START(RAM)
;
MAXD     EQU      $000D     ;MAXIMUM DEPTH
MAXP     EQU      $0004     ;MAXIMUM PRESSURE
INDEX    EQU      $0010     ;DISTANCE BETWEEN PENETRATIONS
MARGIN   EQU      $0014     ;STARTING & ENDING POINTS
RECORD   EQU      $0008     ;RECORDER ON(1) OR OFF(0)
TEST     EQU      $0018     ;TEST NUMBER
SAMP     EQU      $0001     ;SAMPLE SPACING
;
KEYSET   EQU      $7001     ;PRESSED KEY'S NUMBER
UNTTEN   EQU      $7002     ;UNITS AND TENS
HNDTHD   EQU      $7003     ;HUNDREDS AND THOUSANDS
TENTSD   EQU      $7004     ;TENS OF THOUSANDS
MSB      EQU      $7000     ;MOST SIGNIFICANT BYTE
LSB      EQU      $7001     ;LEAST SIGNIFICANT BYTE
NUMBER   EQU      $7005     ;CURRENT DEF.VAR. NUMBER
SAVX     EQU      $7006     ;SAVE THE X-REGISTER HERE
DIFMSB   EQU      $7008     ;SUBTRACT FROM MSB
DIFLSB   EQU      $7009     ;SUBTRACT FROM LSB
ADM      EQU      $2000     ;ADC MSB
ADL      EQU      $2001     ;ADC LSB
LDEPTH   EQU      $B000     ;LSB OF DEPTH
NEND     EQU      $700A     ;NEW END POINT
MCARRY   EQU      $3000     ;MSB OF CARRIAGE
LCARRY   EQU      $D000     ;LSB OF CARRIAGE
LNIND    EQU      $700D     ;LSB OF NEXT INDEX
MNIND    EQU      $700C     ;MSB OF NEXT INDEX
NSAMP    EQU      $700E     ;NEXT SAMPLE TAKEN HERE
```

```
COUNTER EQU     $7010
ACIAS   EQJ     $5000    ;STATUS OF ACIA
ACIAD   EQU     $5001    ;DATA TO/FRCM ACIA
OLDF    EQU     $7012    ; OLD FILE + 1
NEWF    EQJ     $7014    ; NEW FILE + 1
EOF     EQU     $7016    ;END OF OLD FILE +1
SOOF    EQU     $7018    ;START OF OLD FILE +1
SONF    EQU     $701A    ;START OF NEW FILE +1
NOW     EQU     $702A    ;SAVE X-REG(SHORT TERM)
```

APPENDIX B

PROGRAMS CALLABLE BY THE OPERATOR

```
*****************************************************************
*                                                               *
* SETUP ROUTINE                                                 *
* THIS ROUTINE WILL CHANGE DEFALT PARAMETERS                    *
* STORED IN MEMORY LOCATIONS $0000-$0018.                       *
*                                                               *
*****************************************************************

;
SETUP    CLR LED
         JSR LDELAY
         LDX #NEWVAR        ;POINT TO 1ST DEFVAR(MSB)
         INC LED
;
AGN      LDA A 0,X
         STA A MSB
         LDA A 1,X
         STA A LSB
         JSR CONV
         LDA B #$00
         JSR DISPLAY
SET      JSR KEY
         LDA A KEYSET
         CMP A #$05         ;SETUP KEY ?
         BNE C1
         JSR LDELAY
         INX
         INX
         INX
         INX
         INC LED
         LDA A LED
         CMP A #$8          ;CHECK FOR LAST VARIABLE
         BNE AGN
         CLR LED
         RTS
C1       LDA A KEYSET
         CMP A #$02         ;CHECK FOR INC. KEY
         BNE C2
         TST A              ;CLEAR CARRY
         CLR B
         LDA A #$01
         ADD A 1,X
         STA A 1,X
         ADC B 0,X
         STA B 0,X
         JSR LDELAY
         STX SAVX           ;SAVE THE POINTER
         LDX 0,X            ;PRESENT VARIABLE VALUE
         STX NOW
```

```
        LDX  SAVX
        LDX  2,X
        CPX  NOW
        TPA
        LDX  SAVX
        TAP
        BNE  AGN
        STX  SAVX
        CLR  0,X
        CLR  1,X
        LDX  SAVX
        BRA  AGN
C2      LDA  A KEYSET
        CMP  A #$01
        BNE  C3
        TST  A
        CLR  B
        LDA  A #$01
        ADD  A 1,X
        STA  A 1,X
        ADC  B 0,X
        STA  B 0,X
        JSR  SDELAY
        STX  SAVX
        LDX  0,X
        STX  NOW
        LDX  SAVX
        LDX  2,X
        CPX  NOW
        TPA
        LDX  SAVX
        TAP
        BNE  REACH
        STX  SAVX
        CLR  0,X
        CLR  1,X
        LDX  SAVX
REACH   JMP  AGN
C3      LDA  A KEYSET
        CMP  A #$0B
        BNE  EXTND
        RTS
EXTND   JMP  SET
        PAGE
```

```
*************************************************************
*                                                           *
*  RESET SUBROUTINE                                         *
* THIS ROUTINE SETS THE PENETROMETER AT AN INITIAL          *
* STARTING POINT.                                           *
*                                                           *
*       ; OUTPUT--128-64-32-16-8-4-2-1                      *
*                                                           *
*                                                           *
*                     ON/OFF  R/L     DOWN   UP             *
*                      1/0    1/0      1      1             *
*                                                           *
*************************************************************
;
RESET   JSR KEY          ;SCAN KEY PAD
        LDA A #12        ;RESET KEY ?
        CMP A KEYSET
        BNE BAD          ;NO,FALSE ALARM
        BRA TO           ;CONTINUE
BAD     RTS              ;RETURN BACK TO MAIN
TO      LDA A #$0E       ;NUMBER FOR RESET LED
        STA A LED
        JSR LDELAY       ;WAIT A SECOND OR TWO
        JSR LDELAY       ;WAIT A COUPLE MORE
RES     JSR DEPTH        ;WHERE IS THE SHAFT

        LDA A LSB
        CMP A #35        ;NEAR THE TOP ?
        BLS OK1          ;NO, CONTINUE
        LDA A #01        ;MOVE CYLINDER UP
        STA A OUTPUT
        LDA A #$0B       ;NUMBER FOR LED  CYL.UP
        STA A LED
        BRA RES
OK1     CLR OUTPUT
        CLR LED          ;TURN ALL LED'S OFF
        JSR LDELAY
NAW     JSR CARRY
        LDA A MSB        ; CHECK FOR STARTING POINT
        CMP A #00
        BNE NAWU
        LDA A LSB
        CMP A #40
        BLS OK2
NAWU    LDA A #$0A       ;NUMBER FOR LED  LEFT
        STA A LED
        LDA A #$08       ;MOVE CARRAGE LEFT
        STA A OUTPUT
        BRA NAW
OK2     CLR OUTPUT       ;STOP CARRAGE
        CLR LED          ;TURN OFF LED'S
        RTS
        PAGE
```

```
*********************************************************
*                                                       *
* THIS SUBROUTINE CONTROLS THE PENETROMETER AND DATA    *
* GATHERING PORTION OF THE PROGRAM.                     *
*                                                       *
*********************************************************
;
START   JSR LDELAY
        LDX #MARGIN      ; CHECK FOR BEGINNING MARGIN
        JSR CARRY
        LDA A 0,X
        CMP A MSB
        BEQ SHT
        BMI CT1
        BRA BU
SHT     LDA A 1,X
        CMP A LSB
        BLS CT1
BU      LDA A #$0C       ;NUMBER INDICATNG CARRAGE RIGHT
        STA A OUTPUT     ;MOVE CARRAGE RIGNT
        LDA A #$09       ;NUMBER FOR RIGHT LED
        STA A LED        ;TURN ON RIGHT LED
        JSR HALT
        BRA START        ;GO AGAIN
CT1     CLR OUTPUT
        CLR LED
        JSR SAMPLE       ;TAKE A SAMPLE HERE
CT3     LDX #INDEX
        JSR CARRY
        CLC              ;CLEAR CARRY
        LDA A 1,X
        ADD A LSB
        STA A LNIND      ;LSB OF NEXT INDEX
        LDA A 0,X
        ADC A MSB
        STA A MNIND      ;MSB OF NEXT INDEX
        LDX #MNIND       ;LDX MSB ON NEW INDEX(AND LSB)
NA      JSR CARRY
        LDA A 0,X
        CMP A MSB
        BEQ SHI
        BRA BV
SHI     LDA A 1,X
        CMP A LSB
        BLS CT2
BV      LDA A #09        ;NUMBER FOR RIGHT LED
        STA A LED        ;TURN ON RIGHT LED
        LDA A #$0C       ;NUMBER FOR CARRAGE RIGHT
        STA A OUTPUT     ;TURN ON CARRAGE CONTROLLER
        JSR HALT
        BRA NA
CT2     CLR OUTPUT       ;TURN OFF MOTOR
        CLR LED          ;TURN OFFLED
        JSR SAMPLE
```

```
         ;CHECK FOR LAST POSSIBLE SAMPLE
         ;NEW END-PRESENT CARRAGE  CMP  INDEX
         LDX MARGIN
         STX DIFMSB
         LDA A #$D8
         STA A LSB
         LDA A #03
         STA A MSB
         JSR SUB
         LDX MSB
         STX NEND
         JSR CARRY
         LDA A MSB         ;TRANSFER CARRAGE TO "DIFMSB"AND"DIFLSB"
         STA A DIFMSB
         LDA A LSB
         STA A DIFLSB
         LDX NEND
         STX MSB
         JSR SUB           ;SUB. PRESENT CARRIAGE FROM END POINT
         LDX #INDEX        ; ENOUGH DISTANCE FOR ANOTHER SAMPLE
         LDA A 0,X
         CMP A MSB
         BEQ TOM
         BMI JIN
         BRA TM
TOM      LDA A 1,X
         CMP A LSB
         BLS JIN
         BRA TM
JIN      JMP CT3           :
TM       LDA A #$08        ;NUMBER FOR CARRAGE LEFT
         STA A OUTPUT      ;MOVE CARRAGE LEFT
         LDA A #$0A        ;NUMBER FOR LEFT LED
         STA A LED         ;TURN ON LED FOR LEFT
MOVON    LDX #MARGIN       ;CHECK FOR BEGINNING MARGIN
         JSR CARRY
         LDA A 0,X
         CMP A MSB
         BEQ MOT
         BRA MOVON
MOT      LDA A 1,X
         CMP A LSB
         BHI CT5
         JSR HALT          ;     ME TO HALT?
         BRA MOVON         ;MOVE CN DOWN THE TRACK|
CT5      CLR OUTPUT        ;SHUT IT DOWN|
         CLR LED
         LDX TEST          ;INCRUMENT TEST NUMBER
         INX
         STX TEST

         RTS
         PAGE
```

APPENDIX C

SUBROUTINES RETRIEVED BY THE OPERATOR

CALLABLE ROUTINES

```
***************************************
*                                     *
* ANALOG TO DIGITAL SUBROUTINE        *
*                                     *
* "ADL" ADDRESSES THE LSB OF THE A/D  *
* "ADM" ADDRESSES THE MSB OF THE A/D  *
* OUTPUT STORED IN "LSB" AND "MSB"    *
*                                     *
***************************************
;
ADC     CLR ADM          ;START CONVERSION
        JSR SDELAY       ;WAIT FOR CONVERSION
        LDA A ADM        ;READ MSB OF ADC
        STA A MSB        ;
        LDA A ADL        ;READ LSB OF ADC
        STA A LSB
        LSR A            ;MOVE 4-LSB TO 0-3 A-REG
        LSR A
        LSR A
        LSR A
        STA A LSB        ;0-3 BITS=LSB 4-7=0000
        LDA A MSB
        ASL A            ;MOVE MSB
        ASL A
        ASL A
        ASL A
        ADD A LSB
        STA A LSB
        LDA A MSB
        LSR A
        LSR A
        LSR A
        LSR A
        STA A MSB
        RTS
        PAGE
```

```
*************************************************
*                                               *
* THIS SUBROUTINE MOVES THE PENETROMETER SHAFT  *
* INTO THE SOIL FOR CONE INDEX MEASUREMENTS.    *
*                                               *
*************************************************
;
SAMPLE  LDX #$C000
        STX COUNTER      ;CLEAR MEMORY FORM BEGINNING
        CLR A            ;OF DATA FILE TO THE END.
CON1    STA A 0,X        ;"COUNTER" HOLDS THE AVAILABLE
        INX
        CPX #$CA01       ;ADDRESS LOCATION
        BNE CON1
;
        LDX TEST
        STX $C000
        LDX #$C002
        STX COUNTER
BL      JSR DEPTH
        LDA A LSB        ;WHEN IS NEXT SAMPLE?
        ADD A SAMPLE     ;DEPTH+SAMPLE=NEW SAMPLE
        STA A NSAMP
        LDA A #$0B       :NUMBER FOR CYL.DOWN LED
        STA A LED
        LDA A #02        ;NUMBER FOR CYL.DOWN
        STA A OUTPUT
;       JSR ADC
ERA     LDX MAXP
        LDA A 0,X
        CMP A MSB        ;MPRESS-ADC
        BEQ LSU
        BHI QUIT
        BRA USL
LSU     LDA A 1,X
        CMP A LSB
        BGE QUIT
USL     JSR DEPTH
        LDA A LSB
        CMP A MAXD       ;CHECK FOR MAX.DEPTH
        BHI QUIT
        CMP A NSAMP      ;SAMPLE TIME YET?
        BLT ERA
;       JSR ADC          ;READ PRESSURE
        LDX COUNTER
        LDA A MSB
        STA A 0,X
```

```
        LDA  A LSB
        INX
        STA  A 0,X
        STX  COUNTER
        JSR  CONV
        LDA  B #04
        JSR  DISPLAY
        LDX  COUNTER
        INX
        JSR  DEPTH
        LDA  A LSB
        STA  A 0,X
        JSR  CONV
        LDA  B #00        ;SELECT TOP DISPLAY
        JSR  DISPLAY
        JMP  BL
QUIT    CLR  OUTPUT
        CLR  LED
        JSR  LDELAY
;
        LDA  A #01        ;BRING CYLINDER BACK UP
        STA  A OUTPUT
        LDA  A #$0B       ;NUMBER FOR CYL.UP LED
        STA  A LED
E4      JSR  DEPTH
        LDA  A #35
        CMP  A LSB
        BGT  E4
        CLR  OUTPUT
        CLR  LED
        JSR  LDELAY
;       JSR  WRITE
        RTS
        PAGE
```

```
*************************************************
*                                               *
* KEY SUBROUTINE - SCANS KEY PAD AND STORES     *
*                  KEY NUMBER IN "KEYSET".      *
*                                               *
*************************************************
;
KEY      CLR A
         STA A KEYS      ;ACTIVATE ALL ROWS
         LDA A KEYS      ;READ COLUMNS
         COM A
         CMP A #$F0      ;PRESSED KEY?
         BHI LOOK        ;YES,LOOK WHICH ONE
RETURN   CLR KEYSET      ;NO,0-IN KESET AND RETURN
         RTS
;
LOOK     LDA A #$E0
         STA A KEYS      ;ACTIVATE ROW ONE
         LDA A KEYS      ;READ COLUMNS
         CMP A #$EE      ;KEY 1?
         BEQ C1R1
         CMP A #$ED      ;KEY 2?
         BEQ C1R2
         CMP A #$EB      ;KEY 3?
         BEQ C1R3
         CMP A #$E7      ;KEY 4?
         BEQ C1R4
;
         LDA A #$D0      ;ACTIVATE ROW 2
         STA A KEYS
         LDA A KEYS      ;READ COLUMNS
         CMP A #$DE      ;KEY 5?
         BEQ C2R1
         CMP A #$DD      ;KEY 6?
         BEQ C2R2
         CMP A #$DB      ;KEY 7?
         BEQ C2R3
         CMP A #$D7      ;KEY 8?
         BEQ C2R4
;
         LDA A #$A0
         STA A KEYS      ;ACTIVATE ROW 3
         LDA A KEYS      ;READ COLUMNS
         CMP A #$AE      ;KEY 9?
         BEQ C3R1
         CMP A #$AD      ;KEY 10?
         BEQ C3R2
         CMP A #$AB      ;KEY 11?
         BEQ C3R3
         CMP A #$A7      ;KEY 12?
         BEQ C3R4
         BRA RETURN      ;FALSE READING
```

```
C1R1    LDA A #1
        STA A KEYSET
        RTS
C1R2    LDA A #2
        STA A KEYSET
        RTS
C1R3    LDA A #3
        STA A KEYSET
        RTS
C1R4    LDA A #4
        STA A KEYSET
        RTS
C2R1    LDA A #5
        STA A KEYSET
        RTS
C2R2    LDA A #6
        STA A KEYSET
        RTS
C2R3    LDA A #7
        STA A KEYSET
        RTS
C2R4    LDA A #8
        STA A KEYSET
        RTS
C3R1    LDA A #9
        STA A KEYSET
        RTS
C3R2    LDA A #10
        STA A KEYSET
        RTS
C3R3    LDA A #11
        STA A KEYSET
        RTS
C3R4    LDA A #12
        STA A KEYSET
        RTS
        PAGE
```

```
***************************************************
*                                                 *
*  DISPLAY SUBROUTINE                             *
*                                                 *
* PASS A "0" IN REGISTER B FOR TOP DISPLAY        *
* PASS A "4" IN REGISTER B FOR BOTTOM DISPLAY     *
* STORE LSB OF NUMBER TO BE DISPLAYED IN UNTTEN*
*                                                 *
***************************************************
;
DISPLAY STX SAVX          ;SAVE THE X-REGISTER
        LDX #UNTTEN       ;LOAD X-REG WITH UNITS & TENS
        STA B NUMBER      ;SAVE THE STARTING DIGIT NUMBER
MOVE    LDA A 0,X         ;MOVE UNIT(HUNDREDS) TO UPPER 4-BITS
        ASL A
        ASL A
        ASL A
        ASL A
        TAB               ;TRANSFER TO THE B-REG
        STA A DISBUF      ;OUTPUT DIGIT VALUE TO DISPLAY
        ADD A NUMBER      ;SELECT DIGIT
        STA A DISBUF
        ADD A  #$08       ;SET OUTPUT ON DECODER
        STA A DISBUF
        SUB A #$08        ;RESET OUTPUT ON DECODER
        STA A DISBUF
        LDA A 0,X         ;LDA A UNIT(HUNDREDS)
        LSR B             ;MOVE B-REG BACK TO LOWER 4-BITS
        LSR B
        LSR B
        LSR B
        SBA               ;SUBTRACT B-REG FROM A-REG
        STA A DISBUF      ;STA TENS(THOUSANDS) AT OUTPUT
        INC NUMBER
        ADD A NUMBER      ;SELECT DIGIT
        STA A DISBUF
        ADD A #$08        ;SET OUTPUT ON DECODER
        STA A DISBUF
        SUB A #$08
        STA A DISBUF      ;RESET OUTPUT ON DECODER
        INC NUMBER        ;INCREMENT DIGIT
        INX               ;
        CPX #TENTSD       ;CHECK FOR LAST DIGIT
        BNE MOVE          ;IF NOT,GET THE NEXT 2
        LDX SAVX          ;RECALL X-REGISTER
        RTS               ;RETURN TO MAIN PROGRAM
        PAGE
```

```
************************************************************
*                                                          *
* THIS PROGRAM CONVERTS THE BINARY VALUE STORED IN         *
* MSB AND LSB AND STORE THE BCD EQUIVALENT IN UNTTEN       *
* AND HNDTHD.                                              *
*                                                          *
************************************************************
;
CONV     STX SAVX           ;SAVE THE X-REG
         PSH A              ;PUSH THE A-REG ON STACK
         PSH B              ;PUSH THE B-REG ON STACK
         CLR UNTTEN
         CLR HNDTHD
         CLR TENTSD
         LDX #$0010
         ;
BEGIN    LDA A UNTTEN       ;UNITS COMPARISON
         TAB
         AND A #$0F
         SUB A #$05
         BMI AT
         ADD B #$03
         ;
AT       TBA                ;TENS COMPARISON
         AND A #$0F0
         SUB A #$50
         BMI BT
         ADD B #$30
         ;
BT       STA B UNTTEN       ;HUNDREDS COMPARISON
         LDA A HNDTHD
         TAB
         AND A #$0F
         SUB A #$05
         BMI CT
         ADD B #$03
         ;
CT       TBA
         AND A #$0F0
         SUB A #$50
         BMI DT
         ADD B #$30
         ;
DT       STA B HNDTHD
         LDA A TENTSD       ;TENS OF THOUSANDS COMP.
         TAB
```

```
        SUB A #$05
        BMI ET
        ADD B #$03
        ;
ET      STA B TENTSD
        ASL LSB
        ROL MSB
        ROL UNTTEN
        ROL HNDTHD
        ROL TENTSD
        DEX
        BNE BEGIN       ;END OF CONV.
        LDX SAVX        ;RELOAD THE X-REG
        PUL B           ;PULL THE B-REG FROM STACK
        PUL A           ;PULL THE A-REG FROM STACK
        RTS             ;RETURN TO PROGRAM
        PAGE
```

```
*********************************************************
*                                                       *
* THIS SUBROUTINE READS THE DEPTH POSITION ENCODER AND *
* CONVERTS THE OUTPUT TO  A VALUE BETWEEN 0-256 BITS.   *
* THE 8 BIT VALUE IS PASSED UNDER THE NAME "LSB".       *
*                                                       *
*********************************************************
;
DEPTH    STA A LDEPTH     ;TOGGLE R/W LINE
         LDA A LDEPTH     ;READ DEPTH
         STA A DIFLSB     ;SUBTRACT THE DEPTH
         CLR DIFMSB       ;READING FROM $FF
         LDA A #$FF       ;
         STA A LSB
         CLR MSB
         JSR SUB
         RTS
         PAGE
```

```
****************************************************
*                                                  *
* THIS SUBROUTINE READS THE PENETROMETER CARRAGE   *
* DISTANCE  AND STORES THE VALUE IN "LSB"AND"MSB. *
*                                                  *
****************************************************
;
CARRY    STA A MCARRY     ;LATCH NEW VALUE INTO F/F
         LDA A MCARRY     ;STORE MSB OF CARRY IN "MSB"
         STA A DIFMSB
         STA A LCARRY     ;LATCH NEW VALUE INTO F/F(LSB)
         LDA A LCARRY     ;STORE LSB OF CARRY IN "LSB"
         STA A DIFLSB
         LDA A #$FF
         STA A LSB
         LDA A #$03
         STA A MSB
         JSR SUB
         RTS
         PAGE
```

```
*****************************************************************
*                                                               *
*  SUBTRACT SUBROUTINE                                          *
* THIS SUBROUTINE SUBTRACTS THE 16 BIT VALUE STORED IN SLSB*
* AND MSB FROM THE VALUE STORED IN DIFMSB AND DIFLSB AND       *
* STORES THE DIFFERENCE IN DIFMSB AND DIFLSB                   *
*                                                               *
*****************************************************************
;
;
SUB     PSH A               ;PUSH A-REG ON STACK
        TST A               ;CLEAR CARRY
        LDA A LSB
        SUB A DIFLSB
        STA A LSB           ;LSB=LSB-DIFLSB
        LDA A MSB
        SBC A DIFMSB        ;MSB=MSB-DIFMSB-CARRY
        STA A MSB
        PUL A               ;PULL THE A-REG FROM STACK
        RTS                 ;RETURN TO MAIN PROGRAM
        PAGE
```

```
*************************************************************
*                                                           *
* THIS SUBROUTINE TRANSFERS DATA FILES FROM ONE SEGMENT *
* OF MEMORY TO ANOTHER.                                     *
*                                                           *
*************************************************************
;
MOV     LDX SOOF            ;START OF OLD FILE
        STX OLDF
        LDX SONF            ;START OF NEW FILE
        STX NEWF
FL      LDX OLDF
        LDA A 0,X
        LDX NEWF
        STA A 0,X
        INX
        STX NEWF
        LDX OLDF
        INX
        STX OLDF
        CPX EOF
        BNE FL
        RTS
```

```
**************************************************************
*                                                            *
* INITALIZATION OF I/O PORTS(KEYPAD,LEDS,LCDS,CONTROLS)      *
*                                                            *
**************************************************************
;
INIT    CLR A
        STA A CRA1          ;POINT TO DDRA
        STA A CRB1
        STA A CRA2
        STA A CRB2
        LDA A #$FF          ;INIT PIA'S AS OUTPUTS
        STA A DDRB1
        STA A DDRA2
        STA A DDRB2
        LDA A  #$F0         ;INIT MSN AS INPUTS,LSN AS OUTPUTS
        STA A DDRA1
        LDA A #04
        STA A CRA1
        STA A CRB1
        STA A CRA2
        STA A CRB2
        CLR A               ;CLEAR ALL PIA'S
        STA A KEYS
        STA A LED
        STA A OUTPUT
        STA A DISBUF
        RTS
        PAGE
```

```
****************************************
*                                      *
*   LONG DELAY SUBROUTINE---LDELAY     *
*                                      *
****************************************
;
LDELAY  STX SAVX            ;SAVE THE X-REG
        PSH A               ;PUSH THE A-REG ON STACK
        LDA A #$02          ;DELAY FOR .5 SEC.
L2      LDX #$E000
AND     DEX
        BNE AND
        DEC A
        BNE L2
        LDX SAVX            ;RELOAD THE X-REG
        PUL A               ;PULL THE A-REG FROM STACK
        RTS
        PAGE
```

```
****************************************
*                                      *
* SUBROUTINE TO DISPLAY TEST NUMBER    *
*                                      *
****************************************
;
TESTNO  LDX TEST          ;GET CURRENT TEST NO.
        STX MSB
        CLC
        JSR CONV          ;CONVERT TO BCD
        LDA B #00         ;SELECT TOP DISPLAY
        JSR DISPLAY       ;DISPLAY
        LDA A #07         ;NUMBER FOR LED
        STA A LED
        RTS
        PAGE
```

```
*************************************
*                                   *
*   SHORT DELAY OF 35 MS - SDELAY   *
*                                   *
*************************************
;
SDELAY  STX SAVX          ;SAVE THE X-REG
        PSH A             ;PUSH THE A-REG ON STACK
        PSH B             ;PUSH THE B-REG ON THE STACK
        LDX #$0F0D        ;DELAY FOR 35 MICROSECONDS
L1      DEX
        BNE L1
        LDX SAVX          ;RELOAD THE X-REG
        PUL B             ;PULL THE B-REG FROM STACK
        PUL A             ;PULL THE A-REG FROM STACK
        RTS               ;RETURN TO PROGRAM
        PAGE
```

```
*****************************************************
*                                                   *
* HALT SUBROUTINE                                   *
*                                                   *
* THIS SUBROUTINE ALLOWS THE OPERATOR TO STOP AND   *
* AND RESTART THE CARRIAGE DRIVE AND CYLINDER       *
* MOVEMENT WITHOUT REENTERING THE MCNITOR PROGRAM.  *
*                                                   *
*****************************************************
;
HALT    LDA A OUTPUT
        LDA B LED
        CLR MSB
        CLR LSB
        PSH A
        LDA A #$0C
         STA A LED
LYNN    JSR KEY
        LDA A #04
        CMP A KEYSET
        BNE LYNN
        PUL A
        STA A OUTPUT
        STA B LED
        RTS
        PAGE
```

APPENDIX D

PROGRAMS FOR DATA STORAGE AND COMMUNICATION BETWEEN THE CONTROL UNIT AND DEC TU58

```
***********************************************************
*                                                         *
* SUBROUTINE WRITE                                        *
*                                                         *
* THIS SUBROUTINE STORES 1024 BYTES OF DATA STARTING      *
* AT C000 ONTO THE TU58 IN THE FIRST FREE SPACE           *
* AVAILABLE (DRIVE 0 OR 1).                               *
*                                                         *
***********************************************************
;
WRITE   PSH A           BORROW (A)
        LDA A $7064     GET CURRENT DRIVE #
        STA A $7074
        LDX $706A       GET CURRENT BLOCK COUNT
        STX $707A
        LDA A #12       LOAD SUM COUNT
        LDX #$7070      LOAD START ADDRESS
        JSR SUM         AND SUM
        LDX $707E       GET SUM
        STX $707C       AND STORE
;
        LDX #$7070      OUTPUT COMMAND
        JSR COM
;
        LDX #$C000      START HERE
W11     LDA A #1
        STA A $707E
        LDA A #128      SAVE FOR SUM OF FLAG AND BYTE COUNT
        STA A $707F
        JSR SU1         PART OF SUM SUBROUTINE
        LDA A #1
        JSR TRANS       SEND FLAG AND BYTE COUNT
        LDA A #128
        JSR TRANS
;
W2      LDA A 0,X
        JSR TRANS
        INX             SEND 128 BYTES DATA
        STX $706E
        LDA A $706F
        CMP A #$80
        BEQ OUT21
        TST A
        BNE W2
;
OUT21   LDA A $707E
        JSR TRANS
        LDA A $707F     SEND CHECKSUM
        JSR TRANS
;
        JSR PACK        WAIT FOR CONTINUE
```

```
        CPX #$C400
        BLT W11
        INC $706A
        INC $706A
        BNE W3
        INC $706B
W3      LDA A $706B
        CMP A #2
        BNE W4
        RTS
;
W4      TST $7064
        BEQ W5
        SWI                 SORRY, FULL
;
W5      INC $7064
        CLR $706B
        CLR $706A
        RTS
        PAGE
```

```
*************************************************************
*                                                           *
*   SUBROUTINE INIT                                         *
*   THIS SUBROUTINE INITIALIZES THE TAPE IN DRIVE 0.  ALL   *
*   DATA ON THE TAPE IS LOST.  THE TAPE IS ERRASED FOR NEW  *
*   DATA  STORAGE.                                          *
*                                                           *
*************************************************************
;
ITAPE   LDELAY
APE     LDA A #$03
        STA A LED
        JSR SDELAY
        JSR KEY
        LDA A #06
        CMPA KEYSET
        BEQ TAP
        LDA A KEYSET
        CMP A #00
        BGT ROB3          ;BALE OUT
        CLR LED
        JSR SDELAY
        BRA APE
TAP     LDA A #03
        STA A LED         ;LED-RECORD
        CLR $708A         RESET VALUES
        CLR $708B
        LDA A #$0F
        STA A $708C
        LDA A #$0A
        STA A $708D
;
IN      LDX #$7080        START OF COMMAND
        JSR COM
        LDX #$7050        START OF DATA PACKAGE
        JSR COM
;
        INC $708A
        BNE ROB1                  IS IT TIME TO INC 708B?
        INC $708B
ROB1    INC $708C
        BNE ROB2                  IS IT TIME TO INC 708D?
        INC $708D
;
ROB2    LDA A #2          WE ARE FINISHED IF IT IS A 2
        CMP A $708B
        BNE IN
        RTS
        PAGE
```

```
*****************************************************************
*                                                               *
* SUBROUTINE FIND                                               *
* THIS SUBROUTINE FINDS THE FIRST FREE SPACE ON THE             *
* TAPES TO WRITE DATA.  IT STARTS WITH DRIVE 0 AND              *
* CONTINUES THROUGH THE END OF THE TAPE ON DRIVE 1.  IT         *
* RETURNS TO THE MAIN PROGRAM WITH THE BLOCK COUNT IN           *
* 706A-706B AND THE UNIT NUMBER IN 7064.  COMMAND SEQUENCE      *
* STARTS AT 7060.                                               *
*                                                               *
*****************************************************************
;
FIND    PSH A           BORROW (A)
        STX $708E       BORROW (X)
        CLR $7064       UNIT NUMBER TO 0
FZ      CLR $706A       STARTING BLOCK NUMBERS
        CLR $706B
FZZ     LDA A #$C       NUMBER OF BYTES TO BE SUMMED
        LDX #$7060      LOAD START OF SUM
        JSR SUM         AND SUM
        LDX $707E       GET SUM
        STX $706C       AND STORE
;
F1      LDX #$7060      SEND COMMAND
        JSR COM
;
        LDA A $7036
        CMP A #1        IS IT A "1"?
        BEQ F5
;
        INC $706A
        INC $706A       NO, THEN KEEP ON SEARCHING
        BNE F2
        INC $706B
F2      JSR PACK        EAT END PACK
        LDA A #2
        CMP A $706B     ARE WE AT END OF TAPE?
        BNE FZZ
;
        TST $7064       ARE WE ON DRIVE 0 OR 1?
        BEQ F4
        SWI             SORRY, NO FREE SPACE
;
F4      INC $7064       TRY DRIVE 1
        BRA FZ
;
F5      JSR PACK        EAT END PACK
        PUL A           RESTORE (A)
        LDX $708E       RESTORE (X)
        RTS
        PAGE
```

```
**************************************************************
*                                                            *
* SUBROUTINE SUM                                             *
*                                                            *
* THIS SUBROUTINE CALCULATES A 16-BIT CHECKSUM               *
* STARTING AT A MEMORY LOCATION PASSED TO IT IN              *
* THE (X) REGISTER FOR THE LENGTH PASSED TO IT               *
* IN THE (A) REGISTER.  THE CHECKSUM IS FORMED               *
* BY SUMMING SUCCESSIVE BYTES PAIRS TAKEN AS                 *
* 16-BIT WORDS AND USING AN END-AROUND CARRY                 *
* FROM BIT 15 TO BIT 0.  THE COMPUTED CHECKSUM               *
* IS RETURNED TO THE CALLING PROGRAM IN 707E                 *
* AND 707F.                                                  *
*                                                            *
**************************************************************
;
SUM     CLR $707E        CLEAR SCRATCHPAD
        CLR $707F
SU1     PSH A            BORROW (A)
        PSH B            BORROW (B)
        STX $708E        BORROW (X)
        ROR A            DIVIDE LENGTH BY 2
        CLC              MAKE SURE CARRY CLEAR
S1      PSH A            SAVE COUNT
        LDA A 0,X
        LDA B 1,X
;
        ADD A $707E      SUM TOTAL IN (A)
        ADC B $707F      SUM TOTAL+CARRY IN (B)
        ADC A #0         SUM TOTAL+CARRY-AROUND IN (A)
;
        STA A $707E      PUT BACK IN
        STA B $707F
;
        INX
        INX              UPDATE X
        PUL A            RESTORE COUNT
        DEC A            DECREMENT COUNT
        BNE S1           THROUGH?
;
        PUL B            RESTORE (B)
        PUL A            RESTORE (A)
        LDX $708E        RESTORE (X)
        RTS
        PAGE
```

```
****************************************************************
*                                                              *
*  SUBROUTINE TO OUTPUT STORED COMMANDS STARTING AT            *
* ADDRESS  SENT IN X REGESTER.  IT THEN WAITS FOR REPLY        *
* SENT BY TU58  AND STORES THE REPLY STARTING AT 7030          *
*                                                              *
****************************************************************

* 7030 FLAG
* 7031 MESSAGE BYTE COUNT
* 7032 OP CODE              ALWAYS 40
* 7033 SUCESS CODE
* 7034 UNIT NUMBER
* 7035 NOT USED
* 7036 SEQ NO. L
* 7037 SEG NO. H
* 7038 BYTE COUNT L.
* E909 BYTE COUNT H.
* 703A SUMMARY STATUS L.
* 703B SUMMARY STATUS H.
* 703C CHECKSUM L.
* 703D CHECKSUM H.
COM     NOP
        LDA B #14       COUNTER FOR 14 BYTES OUPUT
GO      LDA A 0,X       GET VALUE OF STORED BYTE
        JSR TRANS       OUTPUT IT TO TU58
        INX             LOCATION OF NEXT BYTE
        DEC B           DECREMENT BYTE COUNTER
        BNE GO          FINISHED?
PACK    JSR MAIL
        CMP A #$10      IS IT A CONTINUE?
        BEQ R1          THEN GO BACK AND CONTINUE
B1      CMP A #$2       IS IT AN END PACKET?
        BEQ OUT         YES, THEN STORE VALUES
        CMP A #1        IS IT A DATA PACKET
        BEQ OUT
        CMP A #4        DID WE DO SOMTHING WRONG?
        BEQ BYE
        BRA PACK        IT MUST BE JUNK MAIL
BYE     JSR BREAK       GET BACK ON GOOD TERMS WITH TU58
        SWI             AND TELL OPERATOR MISTAKE
;
OUT     STX $7040       SAVE (X)
        LDX #$7030      STORE POINTER
        STA A 0,X       SAVE OP CODE
NO      INX
        JSR MAIL
        STA A 0,X
        CPX #$703D
        BNE NO
        LDX $7040       RSETORE (X)
R1      RTS
        PAGE
```

```
*********************************************************
*                                                       *
* SUBROUTINE TO WAIT FOR FINISH OF ACIA TRANSMISSION   *
* IT THEN OUTPUTS THE CONTENTS OF THE (A) REGISTER     *
* SENT BY THE CALLING PROGRAM.                          *
*                                                       *
*********************************************************
;
TRANS   PSH A           SAVE (A)
W1      LDA A ACIAS     GET CURRENT STATUS OF TRANS.
        RORA            IT'S IN BIT 1
        RORA            PUT IT IN CARRY
        BCC W1          READY? NO, THEN WAIT
        PUL A           RESTORE (A)
        STA A ACIAD     SEND CONTENTS OF (A) TO TU58
        RTS



*****************************************************
*                                                   *
* SUBROUTINE TO ESTABLISH COMMUNICATION WITH TU58  *
* USED IN INITIALIZATION OR WHEN COMMUNITCATION    *
* BREAKS DOWN.                                      *
*                                                   *
*****************************************************
;
BREAK   LDA A #$17
        STA A ACIAS     POWER-UP/RESET ACIA
        LDA A #$76      SET ACIA FOR BREAK
        STA A ACIAS
        CLR A
        JSR TRANS
        JSR TRANS       SEND TWO NULL CHARACTERS
        LDA A #$16
        STA A ACIAS     RESET ACIA BACK TO 8 BITS + 1 STOP
        LDA A #4
        JSR TRANS
        JSR TRANS       SEND TWO INITS
        JSR PACK        WAIT FOR CONTINUE
        RTS
        PAGE
```

```
****************************************
*                                      *
*   SUBROUTINE TO REWIND TAPES         *
*                                      *
****************************************
;
WIND    LDX #$FF20      UNIT ZERO TO END
        JSR COM
        LDX #$FF30      UNIT ONE TO END
        JSR COM
        LDX #$FF40      UNIT ZERO TO START
        JSR COM
        LDX #$FF50      UNIT ONE TO START
        JSR COM
        RTS
        PAGE
```

```
*******************************************************
*                                                     *
* SUBROUTINE MAIL                                     *
* THIS SUBROUTINE DETECTS THE PRESENCE OF NEW DATA    *
* SENT FROM THE TU58.  IT PLACES DATA IN (A).         *
*                                                     *
*******************************************************
;
MAIL    LDA A ACIAS     GET CURRENT STATUS OF REC-REG.
        RORA            IT'S IN BIT 0
        BCC MAIL        ANY MAIL? NO, THEN WAIT
        CLC             CLEAR THE SET CARRY
        LDA A ACIAD     OPEN THE MAIL
        RTS
```

VITA

John Blake Wilkerson was born in Dayton, Tennessee on January 25, 1958. He attended elementary and junior high schools in Centerville, Tennessee and graduated from Hickman County High School in 1976.

He attended Columbia State Community College for two years before entering The University of Tennessee at Knoxville. He received a Bachelor of Science degree in Agricultural Mechanization in June 1980. The following fall he accepted a research assistantship with The University of Tennessee Agricultural Engineering Department. He completed the requirements for a Master of Science degree in Agricultural Mechanization in June 1983.

He is a member of the American Society of Agricultural Engineers.