

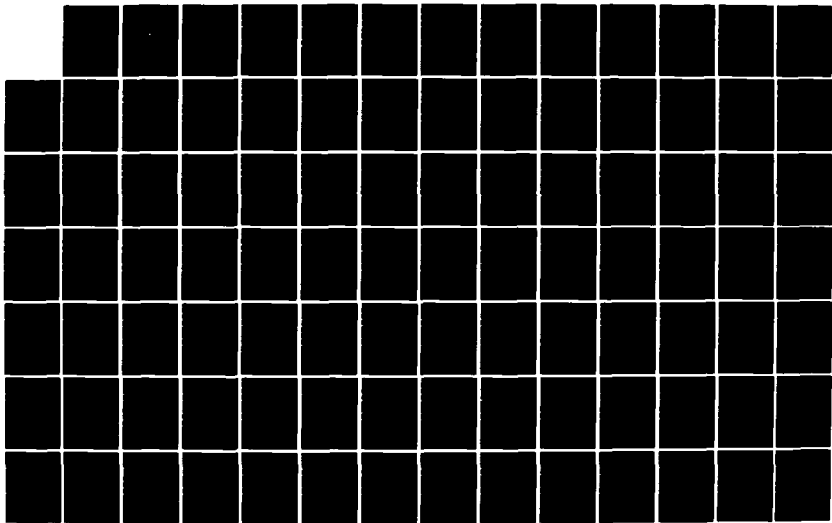
AD-A155 481

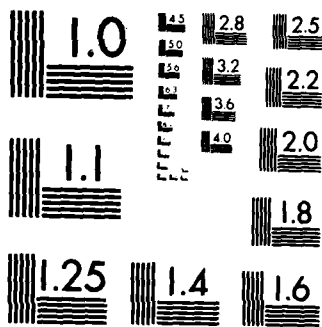
ULTRASONIC POSITION REFERENCE SYSTEMS FOR AN AUTONOMOUS 1/2
SENTRY ROBOT AND A ROBOT MANIPULATOR ARM(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA W M DUNKIN MAR 85

UNCLASSIFIED

F/G 9/2

NL





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

2

AD-A155 481

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
UNCLASSIFIED
JUN 26 1985
B

THESIS

ULTRASONIC POSITION REFERENCE SYSTEMS FOR AN
AUTONOMOUS SENTRY ROBOT
AND A ROBOT MANIPULATOR ARM

by

William M. Dunkin

March 1985

Thesis Advisor:

G. J. Thaler

Approved for public release; distribution is unlimited

85 06 14 120

DTIC FILE COPY

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT. ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Ultrasonic Position Reference Systems for an Autonomous Sentry Robot and a Robot Manipulator Arm		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1985	
7. AUTHOR(s) William Mack Dunkin		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE March 1985	
		13. NUMBER OF PAGES 96	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution is unlimited			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Robot; Robotics; Ultrasonics; Position Reference Systems; Robot Navigation; Manipulator Control; Robot Control,			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis proposes and develops two microprocessor controlled ultrasonic systems which provide absolute coordinate reference for an autonomous sentry robot and a robotic manipulator. These systems close the position control loop which has never been fully closed in robotics and provide a better "testbed" for testing path planning and manipulator arm trajectory control. Each system involves known points of reference and ranges to these points which are determined by timing the interval of propagation of the ultrasonic transmissions. With this data, a microprocessor calculates the positions in cartesian coordinates.			

Approved for public release; distribution is unlimited.

**Ultrasonic Position Reference Systems for an
Autonomous Sentry Robot and a Robot Manipulator Arm**

by

William M. Dunkin
Lieutenant, United States Navy
B.S., United States Naval Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1985

Author: William M. Dunkin
William M. Dunkin

Approved by: George J. Thaler
George J. Thaler, Thesis Advisor

John P. Powers
John P. Powers, Second Reader

Robert T. Strum
Harriett Rigas, Chairman, Department of
Electrical and Computer Engineering

John N. Dyer
John N. Dyer,
Dean of Science and Engineering

ABSTRACT

This thesis proposes and develops two microprocessor controlled ultrasonic systems which provide absolute coordinate reference for an autonomous sentry robot and a robotic manipulator. These systems close the position control loop which has never been fully closed in robotics and provide a better "testbed" for testing path planning and manipulator arm trajectory control.

Each system involves known points of reference and ranges to these points which are determined by timing the interval of propagation of the ultrasonic transmissions. With this data, a microprocessor calculates the positions in cartesian coordinates.

Accession No.	<input checked="checked" type="checkbox"/>
DTIC	<input type="checkbox"/>
AD	<input type="checkbox"/>
NTIS	<input type="checkbox"/>
AS	<input type="checkbox"/>
AN	<input type="checkbox"/>
ANL	<input type="checkbox"/>
ANP	<input type="checkbox"/>
ANR	<input type="checkbox"/>
ANW	<input type="checkbox"/>
ANX	<input type="checkbox"/>
ANZ	<input type="checkbox"/>
ANAA	<input type="checkbox"/>
ANAB	<input type="checkbox"/>
ANAC	<input type="checkbox"/>
ANAD	<input type="checkbox"/>
ANAE	<input type="checkbox"/>
ANAF	<input type="checkbox"/>
ANAG	<input type="checkbox"/>
ANAH	<input type="checkbox"/>
ANAI	<input type="checkbox"/>
ANAJ	<input type="checkbox"/>
ANK	<input type="checkbox"/>
ANL	<input type="checkbox"/>
ANM	<input type="checkbox"/>
ANN	<input type="checkbox"/>
ANO	<input type="checkbox"/>
ANP	<input type="checkbox"/>
ANQ	<input type="checkbox"/>
ANR	<input type="checkbox"/>
ANS	<input type="checkbox"/>
ANT	<input type="checkbox"/>
ANU	<input type="checkbox"/>
ANV	<input type="checkbox"/>
ANW	<input type="checkbox"/>
ANX	<input type="checkbox"/>
ANY	<input type="checkbox"/>
ANZ	<input type="checkbox"/>

A-1



TABLE OF CONTENTS

I. INTRODUCTION 7

II. THEORY AND DESIGN 11

III. HARDWARE FOR THE SENTRY ROBOT'S POSITIONING
SYSTEM 21

IV. SOFTWARE FOR THE SENTRY ROBOT'S POSITIONING
SYSTEM 34

V. SENTRY COORDINATE SYSTEM TEST RESULTS 40

VI. HARDWARE FOR THE MANIPULATOR REFERENCE SYSTEM . . 48

VII. SOFTWARE FOR THE MANIPULATOR REFERENCE SYSTEM . . 51

VIII. MANIPULATOR COORDINATE SYSTEM TEST RESULTS 54

IX. CONCLUSIONS 58

X. RECOMMENDATIONS 61

APPENDIX A: DERIVATION OF SENTRY POSITION EQUATIONS . . 66

APPENDIX B: PROTECTION CIRCUITRY FOR THE PRO-LOG STD
7000 68

APPENDIX C: SENTRY ROBOT REFERENCE SYSTEM SOFTWARE . . . 70

APPENDIX D: MANIPULATOR REFERENCE SYSTEM SOFTWARE . . . 84

APPENDIX E: MASTER WIRING DIAGRAMS 91

LIST OF REFERENCES 94

BIBLIOGRAPHY 95

INITIAL DISTRIBUTION LIST 96

LIST OF TABLES

I	Transmitter Control Codes	26
II	Sample Position Coordinates	42
III	Sample Position Coordinates (cont'd)	43
IV	Port Assignments for the Manipulator	50
V	Receiver Control Codes	53
VI	Sample Position Coordinates (Manipulator)	56
VII	Sample Positions (Manipulator cont'd)	57

LIST OF FIGURES

2.1	Typical Room with Associated Geometry	12
2.2	Sentry Reference System Block Diagram	15
2.3	Typical Manipulator Ultrasonic Reference System . .	16
2.4	Manipulator Reference System Block Diagram	17
2.5	Manipulator Reference System Geometry	18
3.1	RF Transmitter Schematic	23
3.2	Power Switching with Reed Relays	24
3.3	Low Impedance Switching with an Analog Switch . . .	25
3.4	RF Receiver Schematic	27
3.5	LM311 Comparator and Interface Circuitry	28
3.6	LM1812 Schematic	30
3.7	Polaroid Ranging Module Schematic	32
3.8	LM317, LM340T-6.0, and LM723CN Voltage Regulators	33
4.1	Flow Diagram of Sentry Reference System Logic . . .	35
4.2	PL/I - Assembly Language Parameter Passing	39
5.1	Testing Room Geometry	41
6.1	LM311 and Interface Circuitry (Manipulator)	49
7.1	Flow Diagram of the Manipulator System Logic . . .	52

I. INTRODUCTION

As the automation of labor continues, mobile robots which can accurately navigate within their environments will become more important. Although these robots may not realize the humanoid forms of the science fiction writer, their design will entail several general characteristics. Foremost, a robot has a central controller which is most commonly a microprocessor or a network of microprocessors. Second, the robot performs a task or multiple tasks which are usually too repetitive or dangerous for their human counterparts. Finally, the controller of the robot generally controls some type of mechanical motion. This motion includes mechanical manipulators and platform translation. Admittedly, these characteristics encompass many systems which may or may not be considered robots, but as of this date, there is no accepted definition for a robot.

The motivation for this thesis came from "A Microprocessor Controlled Autonomous Sentry Robot", a thesis by Hobart R. Everett, Lieutenant Commander, United States Navy. In his thesis, LCDR Everett describes his first design for an autonomous sentry robot. LCDR Everett also states the need for absolute position information by which the robot may effectively navigate [Ref. 1].

Currently, automated mobile platforms obtain position information from accelerometers, open-loop dead reckoning (DR) schemes, or remote control guidance. Each of these methods is unacceptable to the sentry robot concept. First, accelerometers are too expensive. The sentry robot is ideally an expendable resource since it may operate in a hostile environment. It is also a relatively slow device

with small accelerations which require extremely sensitive accelerometers. The two preceding factors force the requisite accelerometer designs to be cost prohibitive. Secondly, by definition, an autonomous sentry robot operates without human intervention. This requirement eliminates the numerous remotely controlled systems. Finally, the inherent errors of dead reckoning navigation prohibit its use as a long-term navigation system.

Even with the most accurate DR and motion detection devices, position errors accrue in the position information due to measurement errors. Wheel slippage, turn radius errors, quantization and axle/potentiometer interfaces reduce the effectiveness of DR navigation. Quantization and sensitivity limit the accelerometer guidance. Accidents or power interruptions, which involve the robot, adversely affect both DR and accelerometer navigation.

The solution requires the designer to fully close the navigation/position control loop. The robot requires a means by which it can independently update its position information. This information comes from an external absolute reference system. To be effective, the reference system must eliminate or reduce the errors previously discussed and be inexpensive. In addition, the design requires low power consumption, small component size, and ease of implementation. Finally, if the update process operates quickly enough, the need for DR navigation is virtually eliminated.

Armed with accurate position data, the sentry's reconnaissance need not be random patterns like those generated by Everett's "Robart I". An effective sentry transits the entire threat area in a timely and efficient manner. While random motion statistically ensures one hundred percent coverage, there is no guarantee of coverage. A robot, with a memory map as described by Bradley Weinstein

in [Ref. 2], can guarantee one hundred percent coverage by recording its past visit times in areas and using these times to establish interim navigation goals.

While the reference system described within this thesis is for a sentry robot, many of these navigation concepts apply to other robotic platforms. One example is an automated platform for battlefield use. The United States Army has interests in this area. A second vehicle is an autonomous underwater vehicle. Obviously, an underwater vehicle is of great interest to the United States Navy.

The control of robot manipulators is closely related to the autonomous sentry robot's control problem. Although the movements of the manipulator are constrained with respect to those of the sentry, the placement and accuracy of the reference system is far more critical with the manipulators. However, the manipulator controller, at present, still faces the same problem of closing the open-loop of the positioning system.

The repeatability of current manipulators is .05 inches but the absolute accuracy is only .5 inches [Ref. 3]. Any improvement in accuracy requires better machining but at a significant increase in cost. Manipulators have gear and wheel slippage, friction, and quantization problems just like those of the sentry robot. As a result, the designers must include tolerances for errors or a human operator must reinitialize the manipulator after a set number of movements. In order to more fully automate these processes and improve their accuracy, the position control loop needs to be closed with a reference system.

This thesis explores the hardware and software to develop a reference system for both the sentry robot and the robot manipulator. The thesis treats the sentry robot problem first and the manipulator problem follows. All of

the research involves ultrasonic sensors. The designs use ultrasonic sensors because of their cost and availability to the student, and considering that the medium is air, ultrasonic devices are a a logical choice. The theory of both systems resides in the next chapter. Chapters three through eight contain the hardware, software, and test results, respectively, for the autonomous sentry robot reference system. Following the sentry robot design, chapters six through eight present the manipulator design in the same manner. Finally, chapters nine and ten summarize the conclusions and recommendations. The appendices contain complete copies of the software which were developed plus additional information which is referenced in later chapters.

II. THEORY AND DESIGN

The general design for both the sentry robot and the manipulator positioning systems uses geometry which has been used in nautical navigation for many years. Given only range information from known points of reference, the mariner plots circles of equal distance centered at the reference points. The intersection of the lines define the mariner's position. Numerous books abound which discuss this navigation method. This thesis will not discuss these concepts further except to note that three range arcs are required to ensure an unambiguous position or fix in a plane unless other information is available.

The sentry robot's reference system involves two or more remote ultrasonic transmitters at known locations. When a position up-date is required, the robot remotely keys the ultrasonic transmitters via a radio frequency link and measures the time of propagation of the sound waves from the remote site to the robot's ultrasonic receiver. Knowing the speed of sound in air, the robot determines the distance to the remote site using equation 2.1 .

$$\text{distance} = \text{velocity} * (\text{total time} - \text{delays}) \quad (\text{eqn 2.1})$$

The delays will be discussed later. Figure 2.1 shows a typical room with the appropriate geometry plotted.

A robot cannot implement this procedure graphically but must do so mathematically via a microprocessor. Appendix A contains a detailed derivation of the equations. The equations which describe the coordinates of the intersections between two range arcs reduce to a quadratic of the form

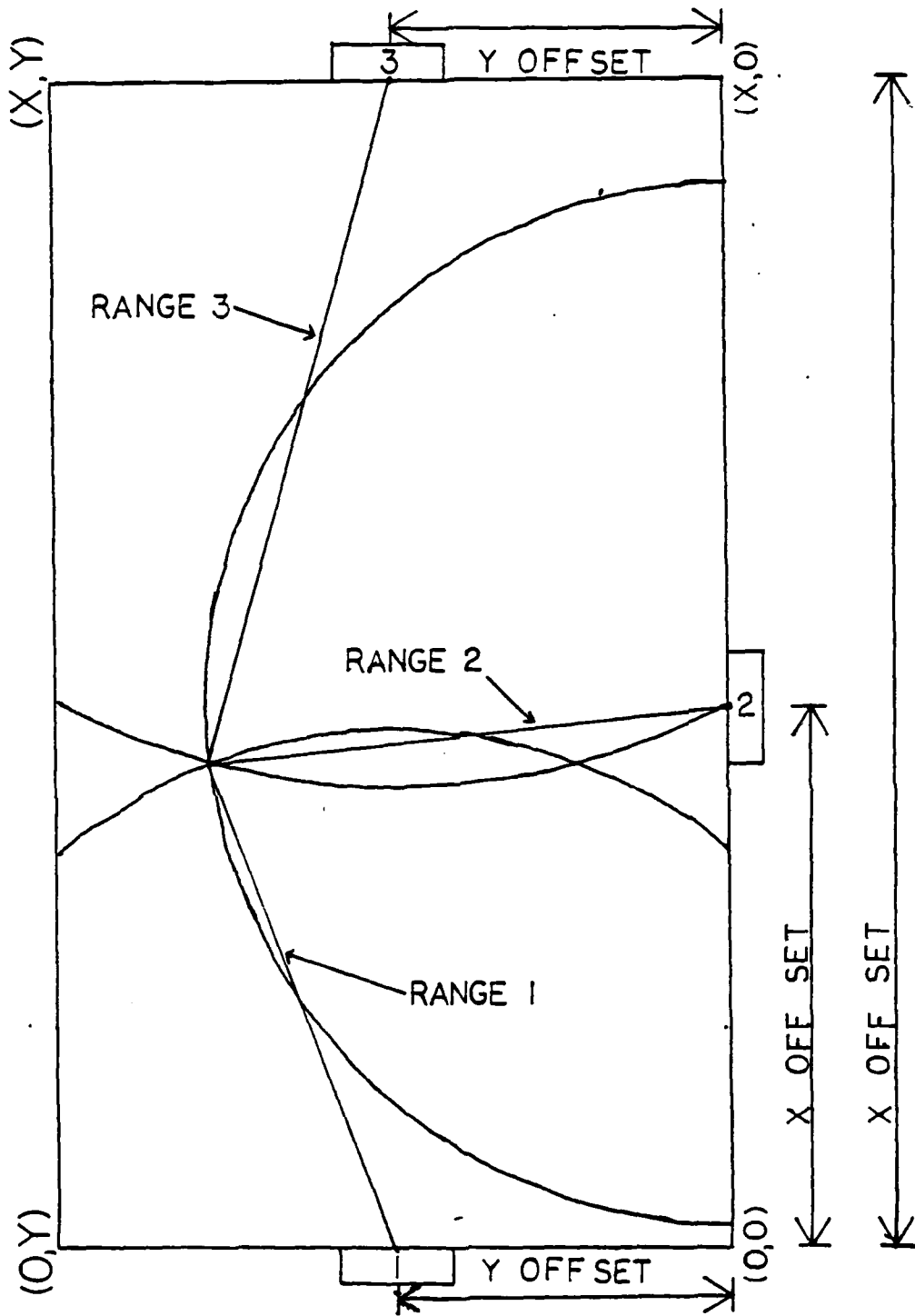


Figure 2.1 Typical Room with Associated Geometry

$$A*x**2 + B*x + C = 0 \quad (\text{eqn 2.2})$$

where

$$A = 4*(OF1**2+OF2**2-2*OF1*OF2+DIF**2) \quad (\text{eqn 2.3})$$

$$B = 4*(OF2**3 - OF1**3 + (OF1-OF2)*R1**2 \quad (\text{eqn 2.4})$$

$$+ OF1*OF2**2 + (OF2*OF1**2 + (R2**2)*(*OF1-OF2)$$

$$- (DIF**2)*(OF1+OF2))$$

$$C = R1**4 + R2**4 + DIF**4 + CF1**4 + OF2**4 \quad (\text{eqn 2.5})$$

$$+ 2*(-(R1**2)*(OF1**2) + (R1**2)*(OF2**2) + (R2**2)*OF1**2$$

$$- (R1**2)*(R2**2) - (R1**2)*(DIF**2) - (R2**2)*DIF**2)$$

$$- (CF1**2)*OF2**2) + (DIF**2)*OF1**2)$$

$$+ (DIF**2)*(OF2**2) - (R2**2)*(OF2**2)$$

$$R1 = \text{range to site one} \quad (\text{eqn 2.6})$$

$$R2 = \text{range to site two} \quad (\text{eqn 2.7})$$

$$OF1 = \text{offset from reference origin to site} \quad (\text{eqn 2.8})$$

$$\text{one on the axis of the coordinate being sought}$$

$$OF2 = \text{offset from reference origin to site} \quad (\text{eqn 2.9})$$

$$\text{two on the axis of the coordinate being sought}$$

$$DIF = \text{difference in the offsets of the two} \quad (\text{eqn 2.10})$$

$$\text{sites on the axis of the coordinate not being sought}$$

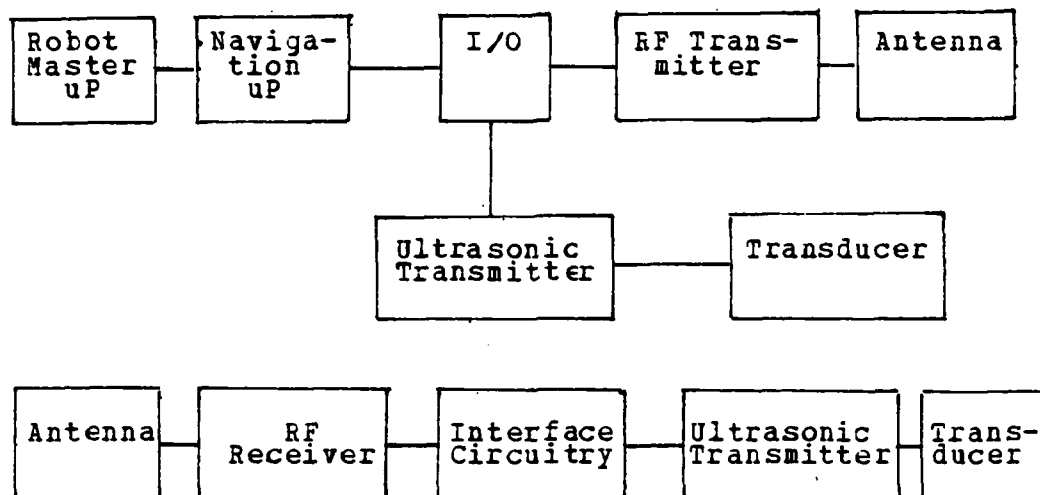
By solving equation 2.2 for all known solutions, the robot determines its position by comparing the solutions of the various pairs of range-arcs, comparing the solutions with its estimated current position, or by using various restrictions. For example, this design permits operation

only in the positive quadrant of an X-Y grid coordinate system. Therefore, the program discards all negative solutions. The design also uses feet as the fundamental unit although any dimensional unit will work as long as all of the quantities are uniform.

Figure 2.2 contains a block diagram of the system. Whenever the robot requires an update of its position, the robot's master microprocessor or a slave microprocessor which may be dedicated to navigation keys the appropriate ultrasonic transmitter by correctly emitting the corresponding radio transmission. This requires the microprocessor to interface with the external radio transmitter. Simultaneously, the microprocessor initiates a timer which measures the time of propagation of the sound wave. At the remote site, a radio frequency receiver which is interfaced with the ultrasonic transmitter detects the appropriate radio signal. The radio receiver triggers the ultrasonic transmitter. An ultrasonic receiver which is also interfaced with the navigation microprocessor detects the ultrasonic signal at the robot.

All of the necessary hardware components are readily available, small, and inexpensive. Most importantly, the components which are located on the robot are low power consumers and they can be powered down when not in use. Low power consumption is essential to a truly autonomous robot since its power supply is self-contained.

While the sentry robot's position system is primarily a two-dimensional problem, the manipulator control problem is three-dimensional. Again, the solution uses range information, geometry, and ultrasonic sensors to identify the position. However, the system does not require a radio frequency link. Instead, the controlling microprocessor interfaces directly with the transmitting and receiving ultrasonic sensors. The problem reduces to an interface of



* uP = Microprocessor

Figure 2.2 Sentry Reference System Block Diagram

the ultrasonic components and a software package for control. Figure 2.3 shows a typical system. Figure 2.4 contains the block diagram of such a system.

The microprocessor determines the manipulator's tip position by applying the law of cosines to the range information obtained from the sensors. By placing three sensors in the same plane in an "L" formation, the problem simplifies and takes advantage of special geometry. Figure 2.5 shows the geometry of the system. The range information of any two sensors, which are colocated on a reference axis, plus their respective locations, define the coordinate of the manipulator on the axis. The two range vectors form two cones which share a common base when the vectors are rotated

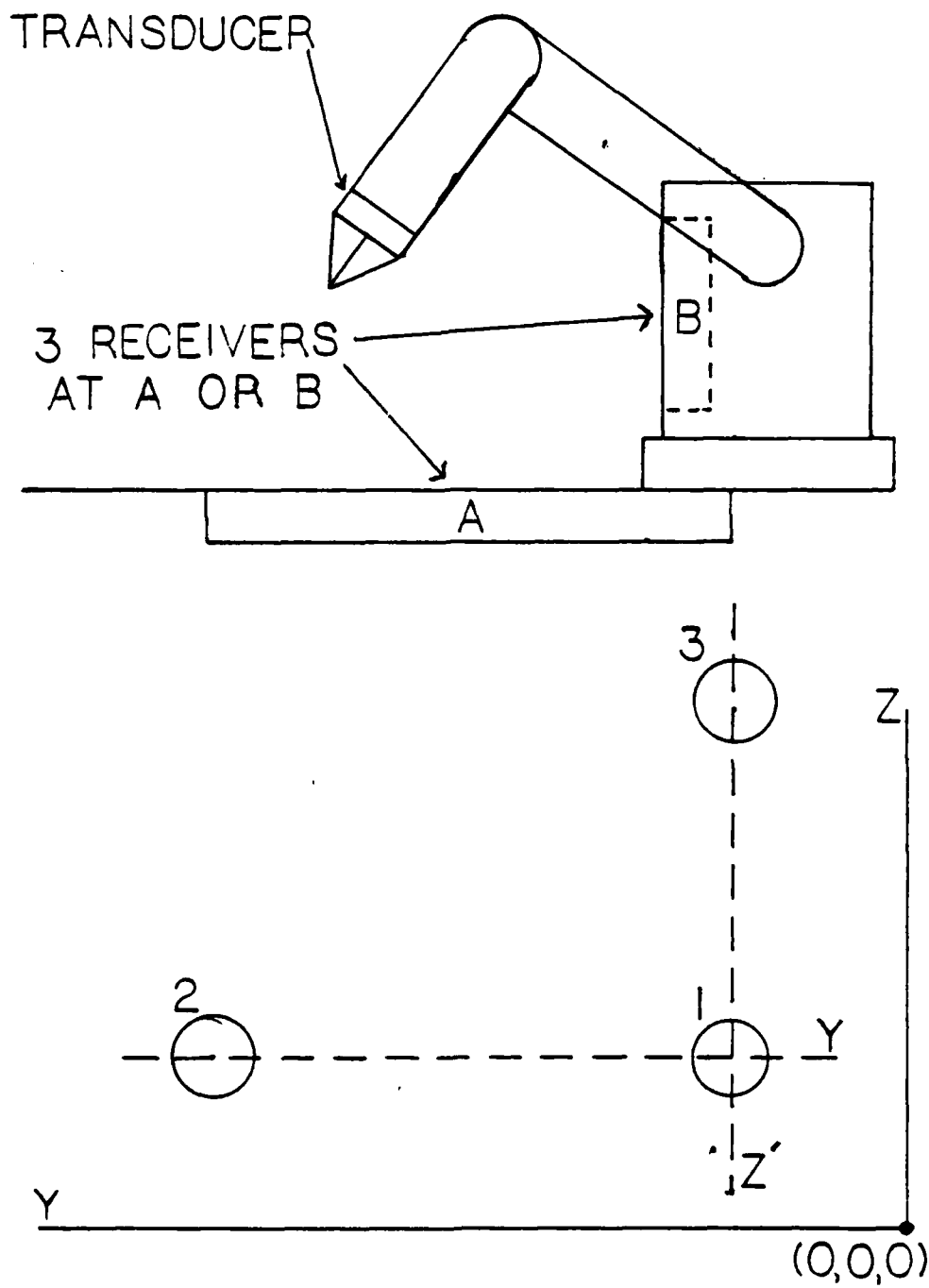


Figure 2.3 Typical Manipulator Ultrasonic Reference System

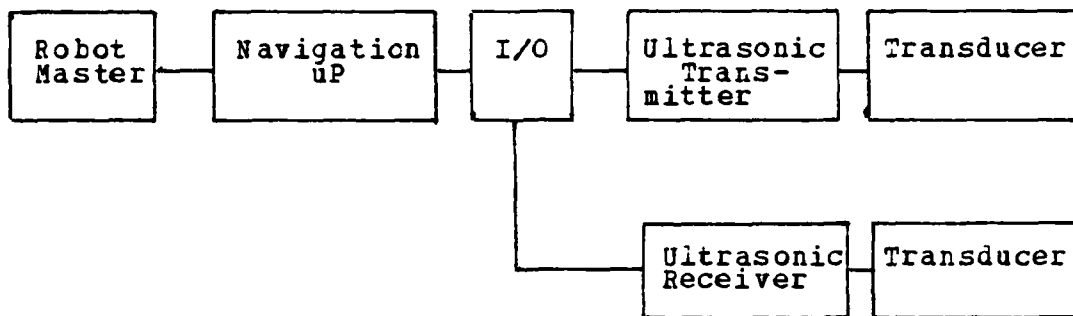


Figure 2.4 Manipulator Reference System Block Diagram

about the common axis. The intersection of the cones' base plane and the reference axis defines the actual coordinate position of the manipulator.

First, with the given range information and the law of cosines, the microprocessor computes the angle of departure of the range-vector with respect to the common axis using equation 2.11

$$\text{Alpha} = \arccos((AR^{**2}+BR^{**2}-AB^{**2}) / (2*AR*BR)) \quad (\text{eqn } 2.11)$$

$$\text{Alpha} = \text{angle of departure} \quad (\text{eqn } 2.12)$$

$$AB = \text{distance between the sensors} \quad (\text{eqn } 2.13)$$

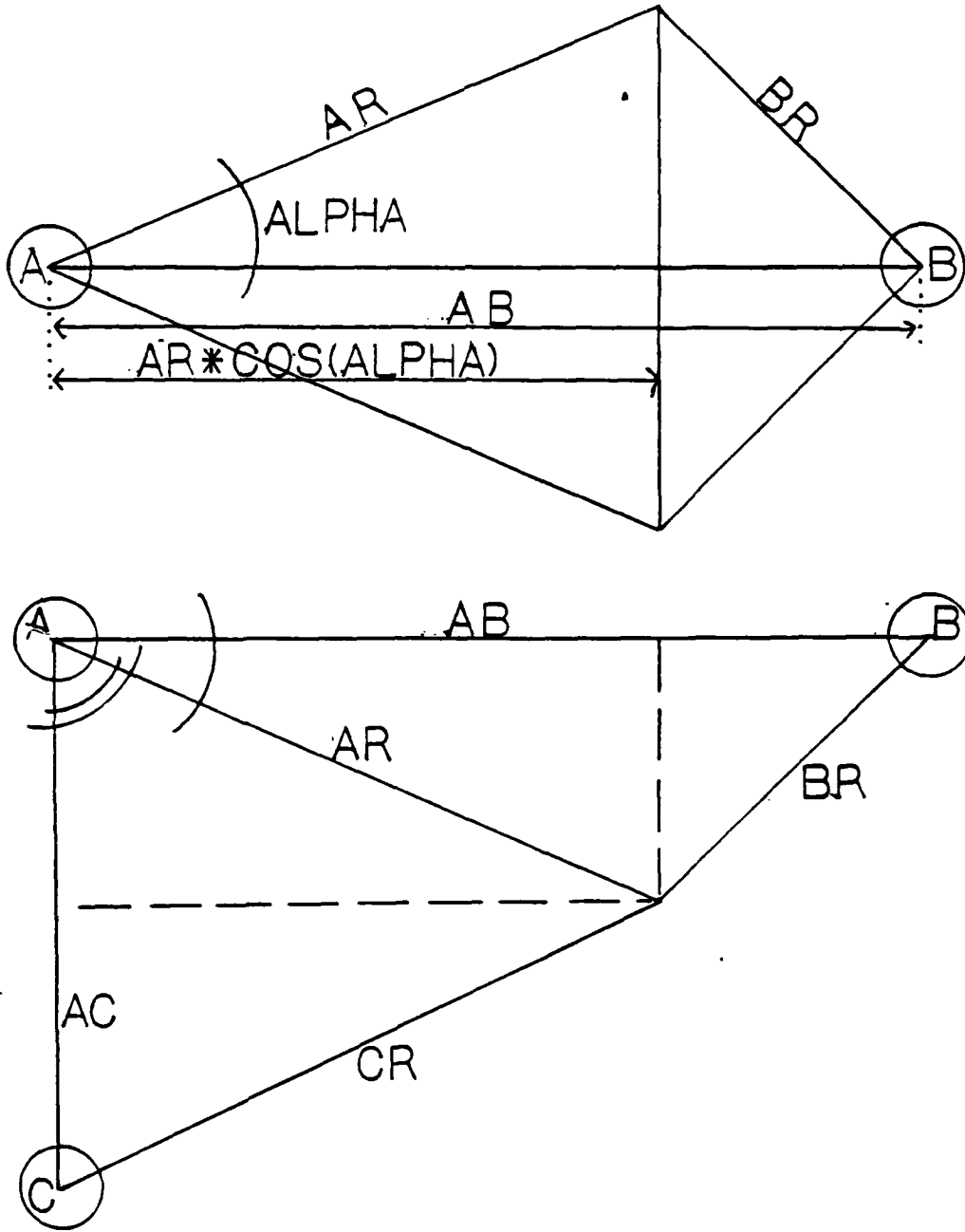


Figure 2.5 Manipulator Reference System Geometry

AR = range from site A to manipulator tip (eqn 2.14)

BR = range from site B to manipulator tip (eqn 2.15)

Next, the microprocessor determines the dot product of the range-vector and the respective axis using equation 2.16 .

Dot product = $AR \cdot \cos(\text{Alpha})$ (eqn 2.16)

The microprocessor determines the actual coordinate by adding the dot product with the respective site offset from the system origin.

After determining the second coordinate in the above manner, the system computes the last coordinate by applying equation 2.17 which relates a vector's magnitude with its respective projections onto a three dimensional coordinate system.

$Z = (AR^2 - X^2 - Y^2)^{.5}$ (eqn 2.17)

X, Y, Z = the respective dot products of the range vector on the axes (eqn 2.18)

This geometry permits the sensors, which establish the coordinate system, to lie in a plane outside of the area of movement of the manipulator. Also, these equations handle all angles from zero to one-hundred eighty degrees since the cosine function is single-valued over this region.

Neither the sentry robot nor the manipulator designs focus upon a particular robot design. Instead, the author attempted to keep the designs simple and transportable so the designs could be interfaced with many different systems. Early simulations used PL/I on the Naval Postgraduate

School's IBM 3033 system. The actual test design involved EL/I-80, Z-80 assembly language, the Pro-Log STD 7000 Development Station and external circuitry. Programming occurred on an Altos computer with MP/M II which contains a EL/I-80 compiler, Macro-80 compiler, Link-80 for module linkage, and Genhex for hex-code generation. As stated earlier, the above choices do not reflect a design decision, but instead, reflect the availability of resources. Finally, the programs maximize their generality by having the hardware interface routines and the clock routines in separate modules. These routines which are specific to each implementation may be changed without seriously affecting the main computational program.

The remainder of this thesis presents the details of these test designs. The hardware and software of the sentry robot's reference system is first. The details of the manipulator reference system follow.

III. HARDWARE FOR THE SENTRY ROBOT'S POSITIONING SYSTEM

From the beginning of this design, the utilization of "off-the-shelf" components was a primary goal. By using components which are currently available, the design is less costly and the designer spends less time "re-inventing the wheel". Instead, the designer faces numerous interface problems between the various components. However, these interface problems, as a rule, are not as complex or difficult as designing the entire system from scratch. This concept parallels current efforts in VLSI, custom chip and VHSIC designs where proven general designs are slightly altered for each specific use. Each new design is merely a collection of building blocks which have been previously designed, tested, and implemented.

The microprocessor interface requires five logic control lines outbound and one data line inbound. While these lines could be directly linked to the external circuitry via the data bus of the microprocessor, this is not a safe idea during early development stages when many changes and/or accidents may occur. Appendix B contains an eight bit I/O port circuit designed by David Rigmaiden of the Naval Postgraduate School. This circuit provides protection for the microprocessor and an addressable input/output port. The design uses two 74138, one 74273, one 7432 and one 74244 chips. The circuit includes access to the address bus, data bus, and the respective control lines such as read and write enable.

The radio transmitter and receiver sections are available at Radio Shack (Catalogue No. 277-1012). The remote control module pair, which is made by Archer, operate at 27 MHz and provide four control functions. Pulse code

modulation delineates two of the four control signals. The other two control signals are a constant carrier signal and the absence of a signal. The transmitter requires a nine volt DC power supply. The receiver requires both a three and nine volt DC power supply. [Ref. 4]

The schematic of the transmitter is in Figure 3.1 The interface problems between the transmitter and the microprocessor are two-fold. First, the microprocessor must control three nine volt power supply lines. Second, the microprocessor must control three low impedance connections. Normally, two double-pole/double-throw switches provide the requisite control.

For the power switching, the design uses reed relays from Claire and Archer. These devices are TTL compatible. Figure 3.2 shows the respective wiring diagrams. Early designs considered the use of NPN, CMOS, and VMOS devices, however the nine volt switching consideration always forced the control voltage levels beyond TTL compatibility. Thus, typical microprocessor control was not possible with these devices.

The interface circuit for the low impedance lines uses the National Semiconductor CD4066BM/CD4066BC quad bilateral switch as shown in Figure 3.3. This analog switch is TTL compatible as long as the IC's power-supply, V_{dd} , remains at TTL levels. If the chip's power supply exceeds five volts (maximum allowable voltage is fifteen volts), the control voltages, which turn the switches off and on, must also increase beyond TTL levels. This fact makes these switches unusable for the power switching control of this system.

Armed with the circuitry of Figures 3.2 and 3.3, the microprocessor keys the radio transmitter by sending the appropriate code to the I/O port at address "FF". Table I contains the respective codes and their consequences. When all connections are open, no transmissions exist. A

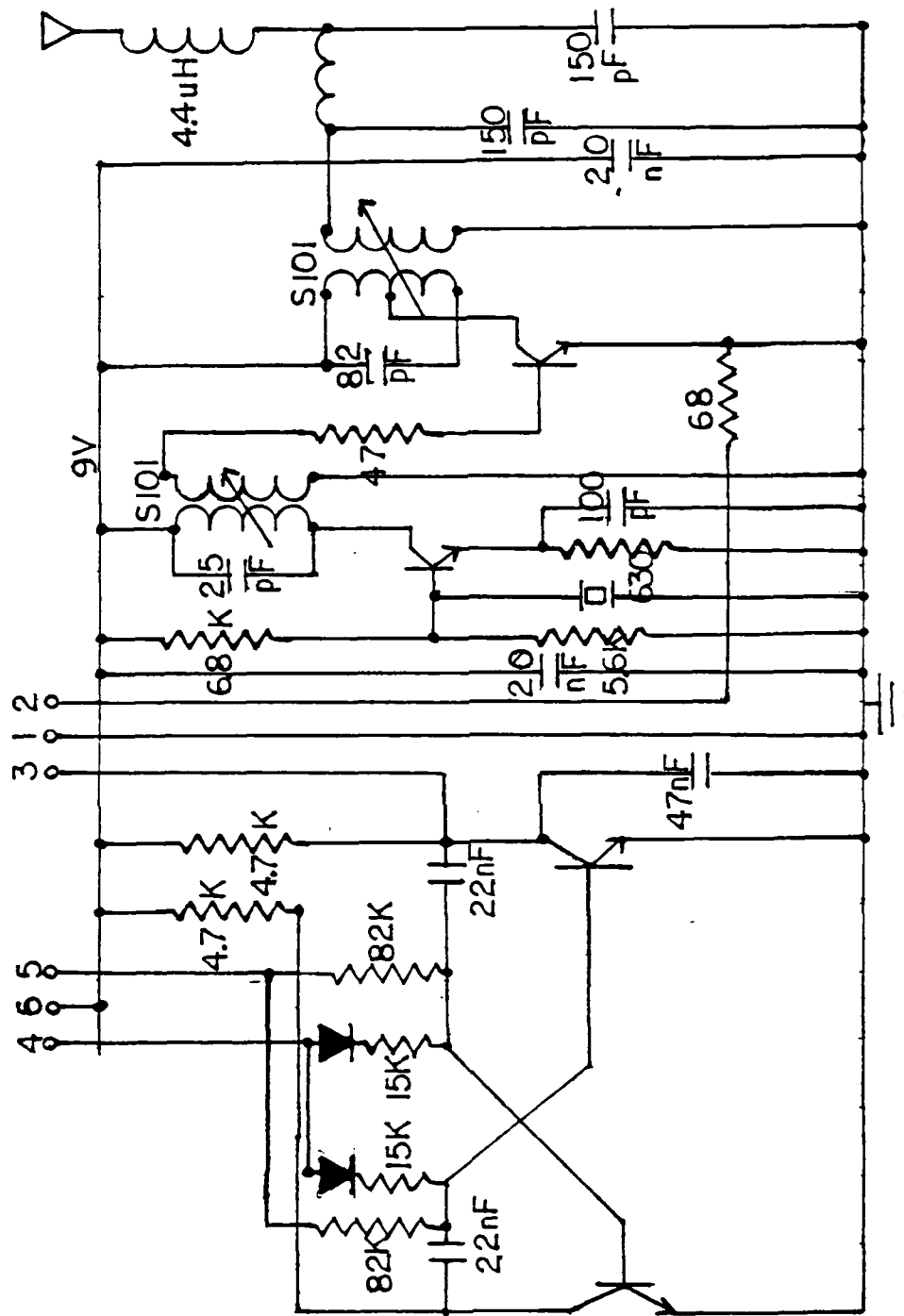


Figure 3.1 RF Transmitter Schematic

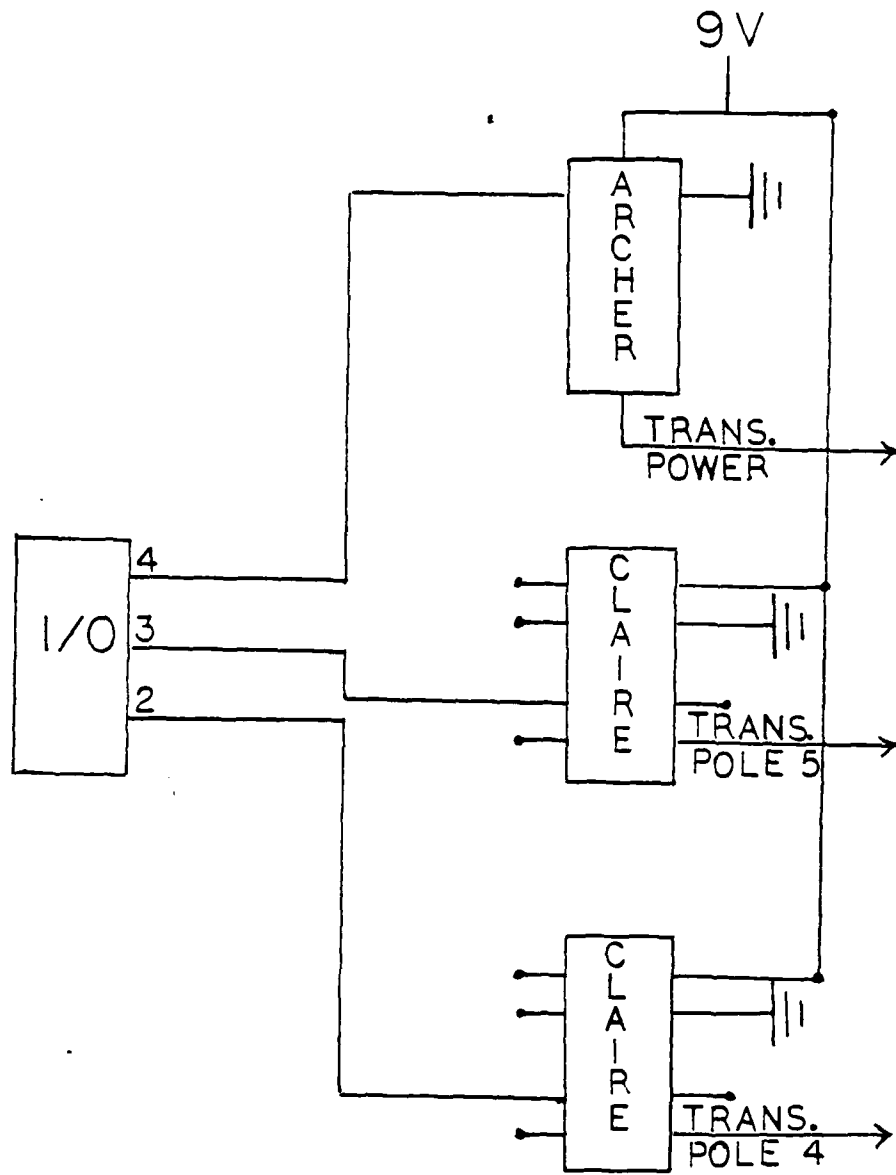


Figure 3.2 Power Switching with Reed Relays

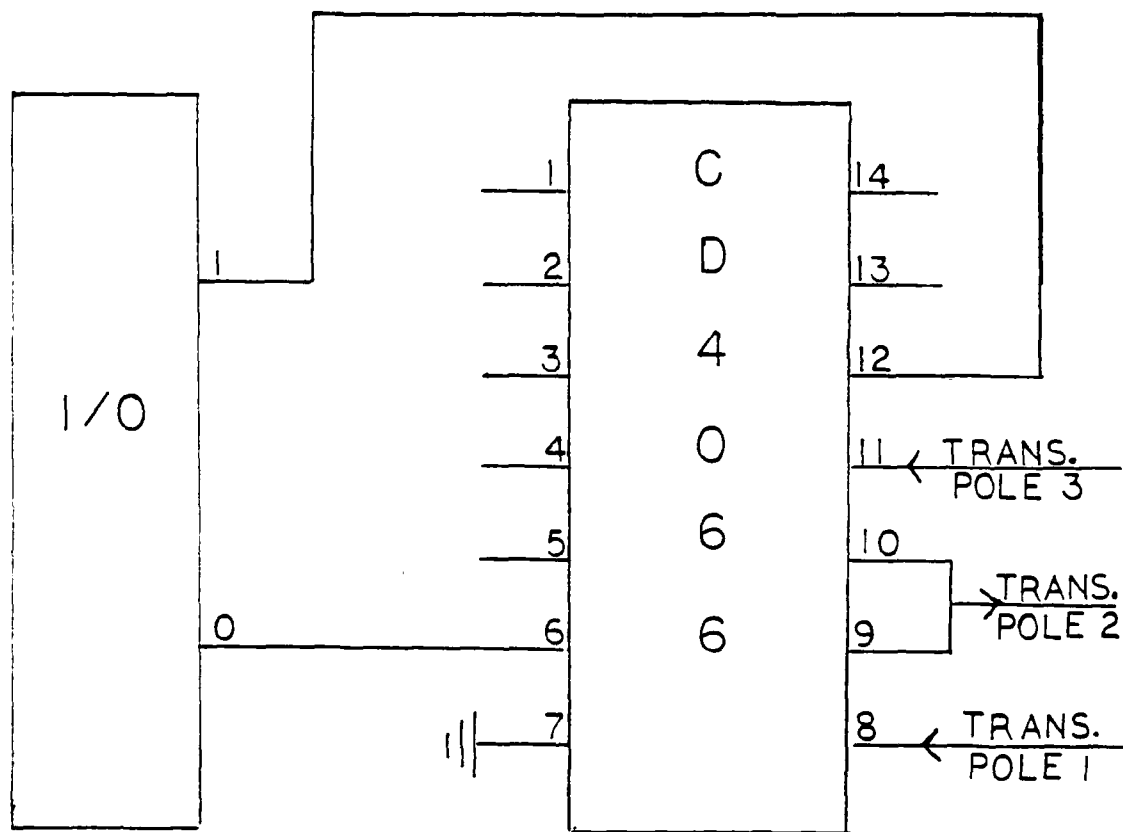


Figure 3.3 Low Impedance Switching with an Analog Switch

combination of transmitter connections between transmitter terminals two and three with either terminal four or five powered by the nine volt source creates the respective pulse code modulation from the transmitter. The connection of transmitter terminals one and three generates the constant 27 MHz carrier.

TABLE I
Transmitter Control Codes

0 0 0 0 0	All connections open*
1 0 0 0 1	Constant carrier generation
1 0 1 1 0	3000 hertz pulse code modulation
1 1 0 1 0	500 hertz pulse code modulation

*(The most significant bit is the left bit.)

The receiver schematic diagram is in Figure 3.4 The three receivers each detect the complete set of control signals. The receivers have three power sections which are normally used to power DC servo motors. The transistors which control these power sections are marked in Figure 3.4 When the appropriate signal is present or absent, the collectors of the controlling transistors have associated voltage levels. An LM311 comparator tests these levels and appropriately keys the ultrasonic transmitter. Figure 3.5 shows the comparator and the interface circuitry which keys the ultrasonic transmitter.

During this research, the author considered two different ultrasonic devices, the Polaroid Ultrasonic Ranging System and the National Semi-Conductor LM1812 Ultrasonic Transceiver Driver with the Panasonic EFR-OTE40KZ Ceramic Transducer. The Polaroid device is an excellent system for range finding during obstacle avoidance. The half-power bandwidth is narrow at ten degrees. It also has

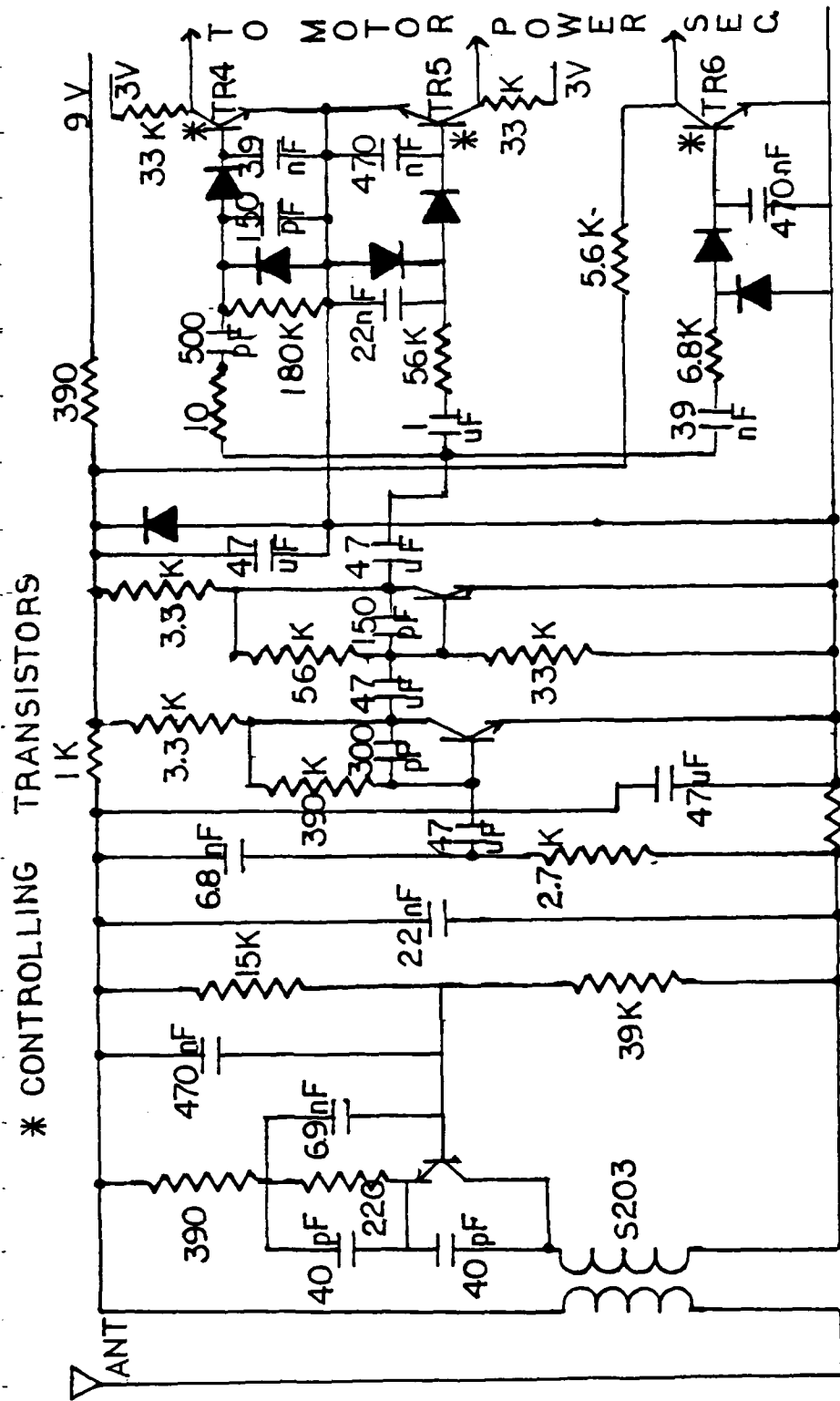


Figure 3.4 RF Receiver Schematic

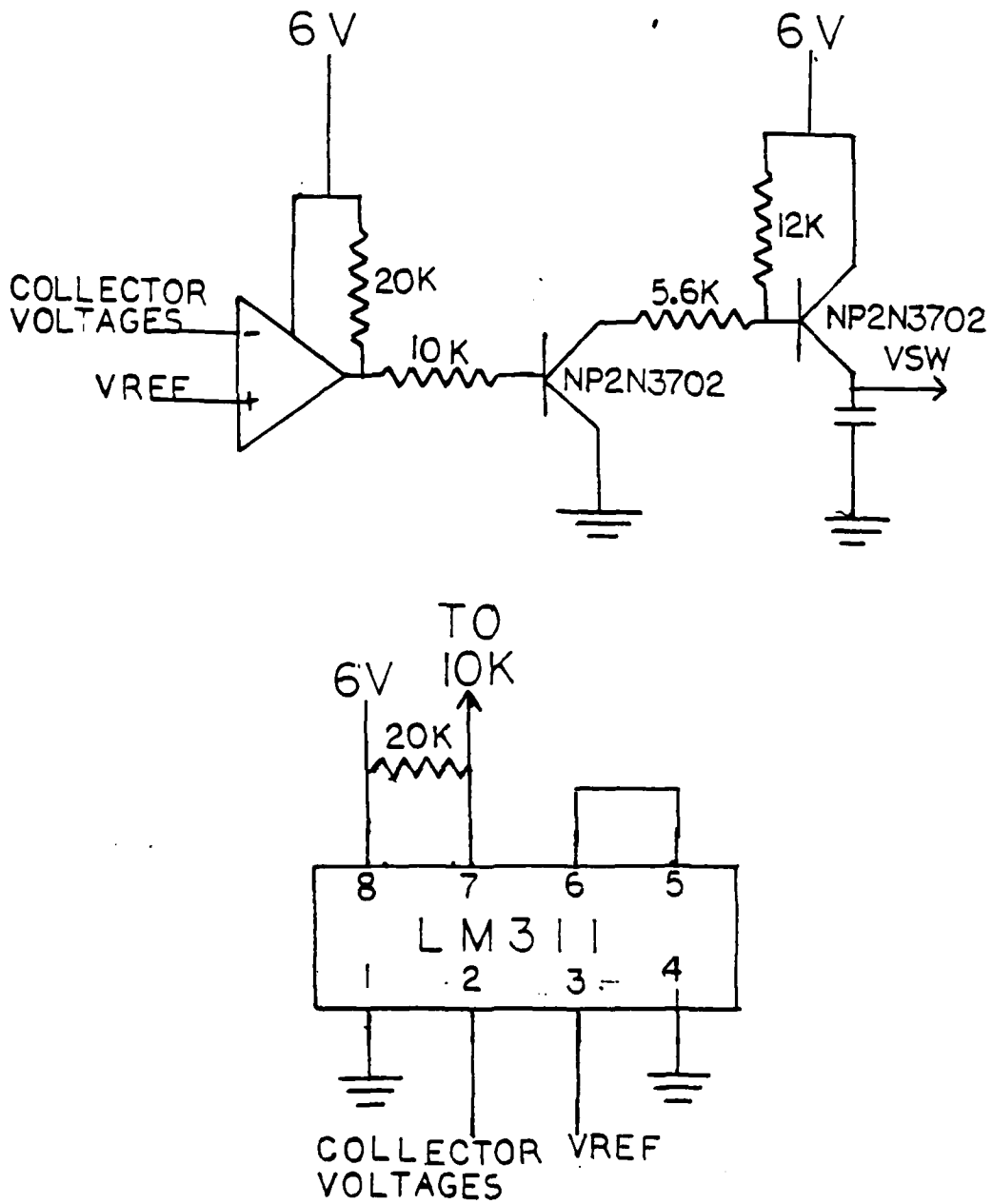


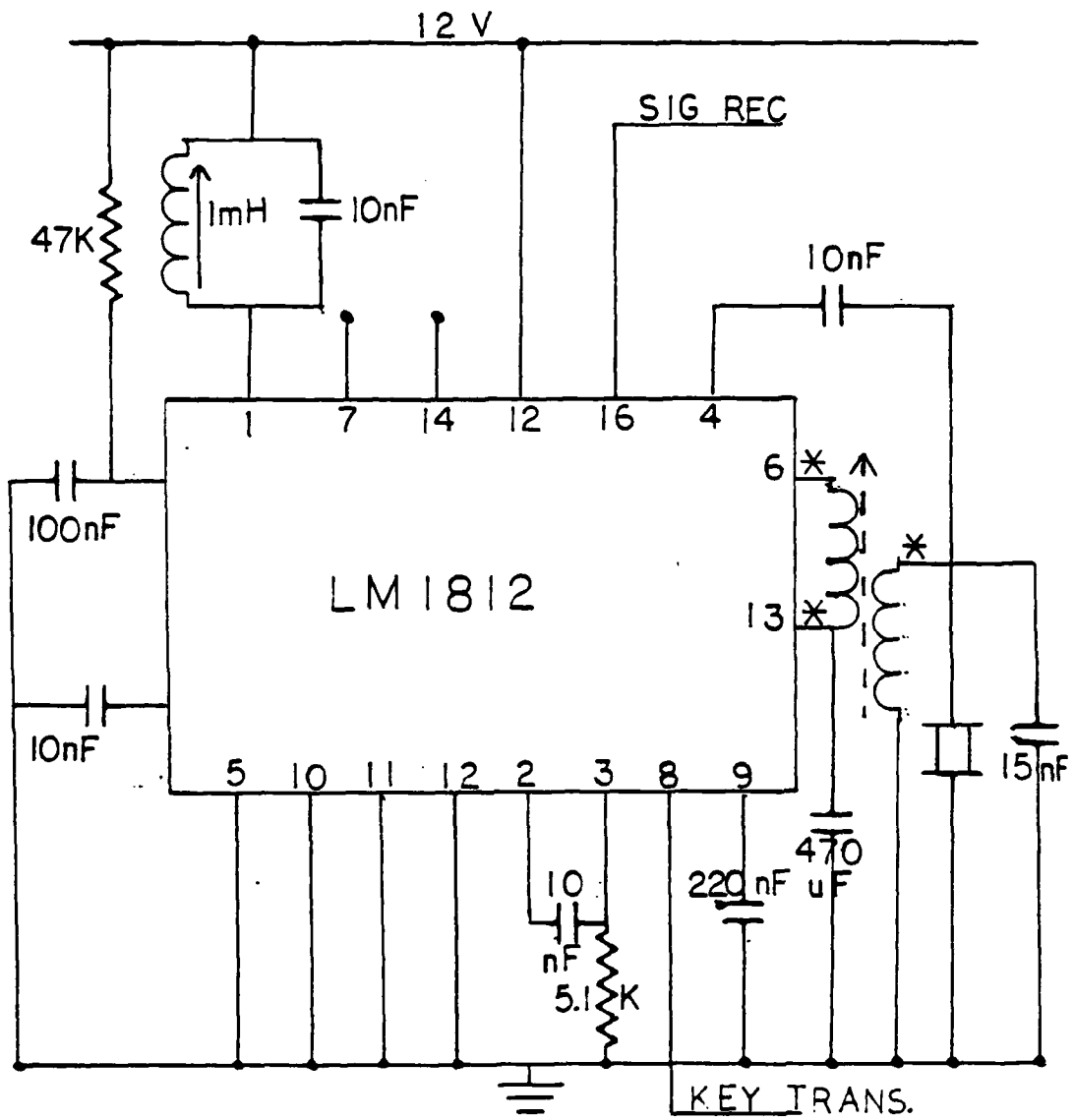
Figure 3.5 LM311 Comparator and Interface Circuitry

an automatic sensitivity control which enables the device to provide reliable data out to thirty-five feet [Ref. 5]. Although this device was used at the remote sites for the system test, the LM1812 is a better choice at both the remote sites and at the robot as a receiver. The Polaroid device uses an electrostatic transducer which must be polarized to receive correctly. The device must either transmit or have a 150 volt power supply in parallel with the transducer to correctly polarize the device [Ref. 6]. In a receive only mode, a transmission to establish this 150 volt differential is unacceptable and the circuitry which is required to maintain a constant 150 volt level is too costly in power consumption. The Panasonic transducers are ceramic devices which are permanently polarized. Thus, the device is passive in a receive only mode.

Also, this positioning concept requires omni-directional transmission and reception. Comparing the 3-db beam width of the Polaroid and Panasonic devices, the Panasonic transducers need far fewer transducers operating in parallel. The Panasonic 3-db beam width is sixty degrees [Ref. 7]. Therefore, the receiver needs only six Panasonic devices versus thirty-six of the Polaroid design. Each transmitter site needs only three Panasonic transducers to provide one-hundred and eighty degrees of coverage (if the transmitter is placed against a wall).

Figure 3.6 contains the circuit diagram for the LM1812. The diagram contains a complete design which will both transmit and receive. The connections marked with an asterisk need not be made when the device is acting only as a receiver.

Figure 3.7 contains the circuit diagram for the Polaroid system. Texas Instruments now markets an improved version of the Polaroid Ranging Module, which is called the SN28827. The SN28827 uses two Texas Instrument chips, the TL852 and



Figure, 3.6 LM1812 Schematic

TL851. However, this unit is compatible only with Polaroids electro-static transducer and thus the lack of permanent polarization remains.

Finally, three separate devices provide voltage regulation for the circuitry. All nine-volt sources use the LM317. The six-volt sources use the LM340T-6.0 and the three-volt sources use LM723CN. The different current requirements of the three voltage supplies determined the device which was employed. Figure 3.8 contains the regulators and their external circuitry.

As stated earlier, the microprocessor interface contains five outbound lines and one input. The outbound lines merely act as switches to turn the respective transmitters on and off along with the requisite power supply. The lone input line tells the microprocessor whether an ultrasonic signal is present at the receiver or not. Therefore, the entire power for timing, calculation, and distance determination lies within the software portion of the system which is discussed in the next chapter.

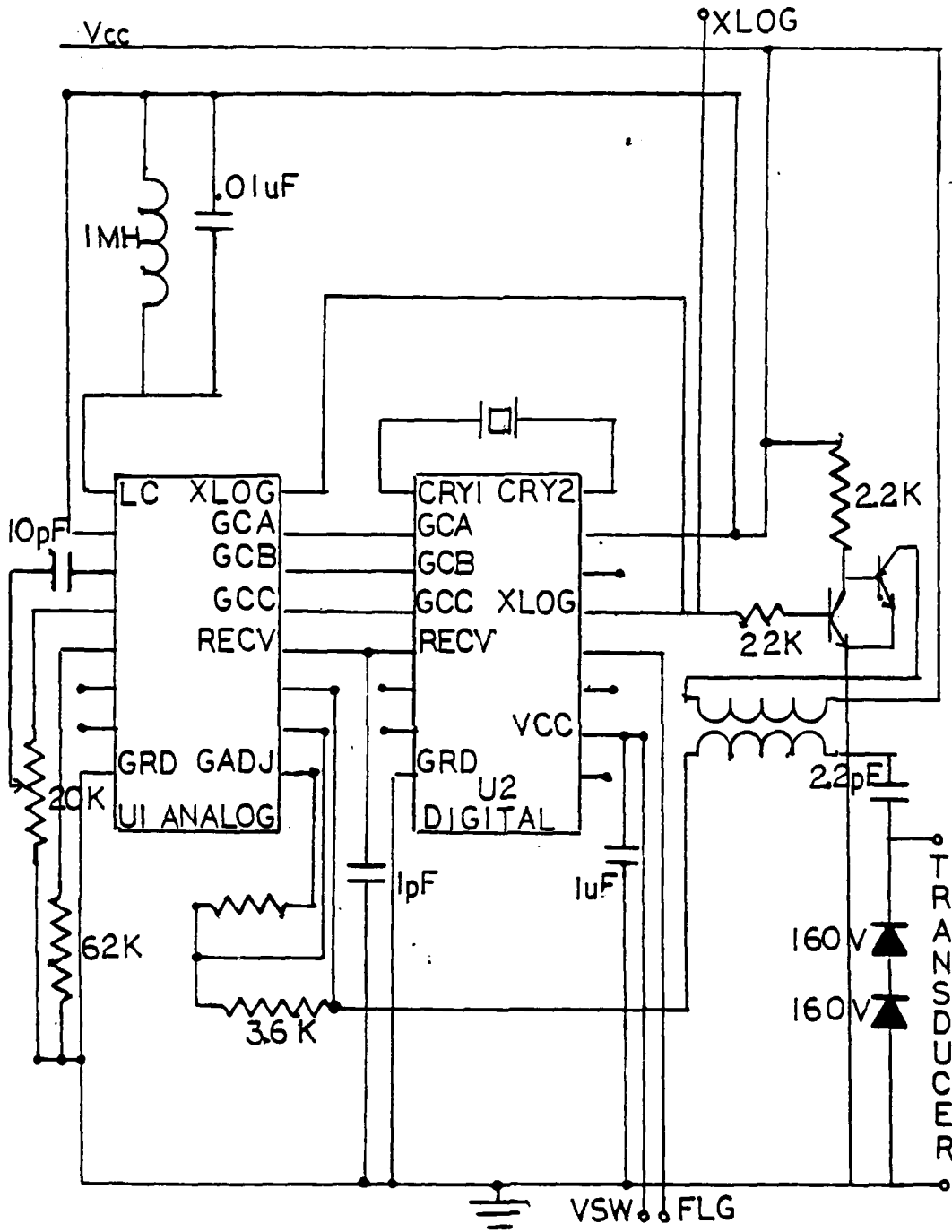


Figure 3.7 Polaroid Ranging Module Schematic

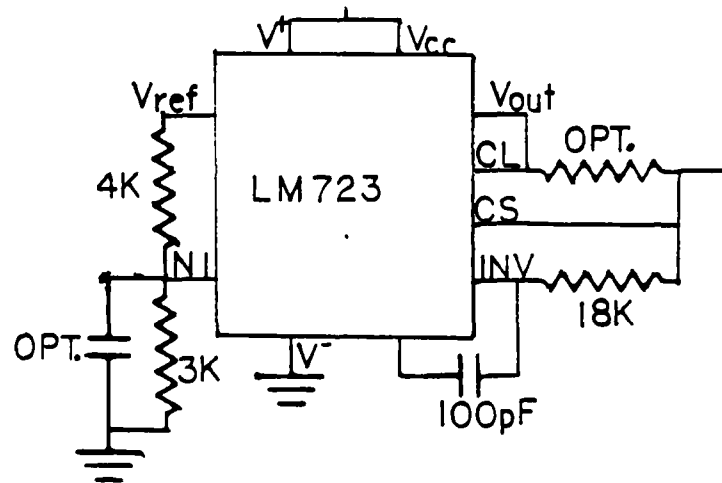
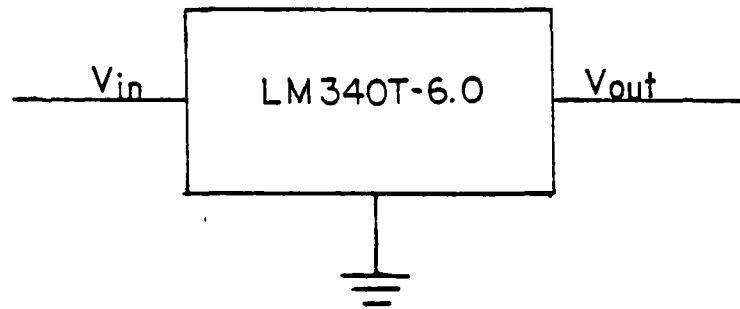
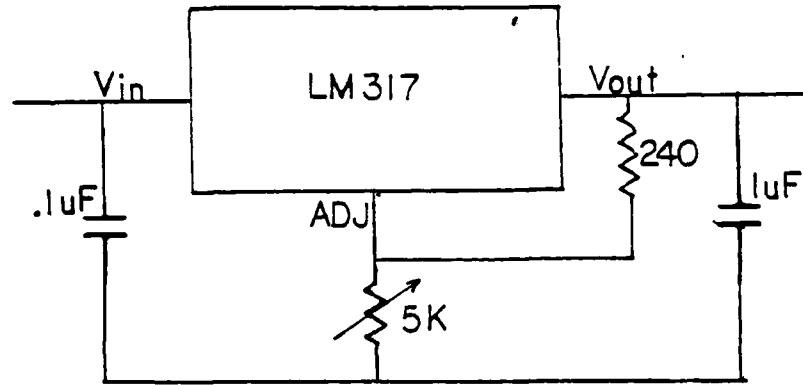
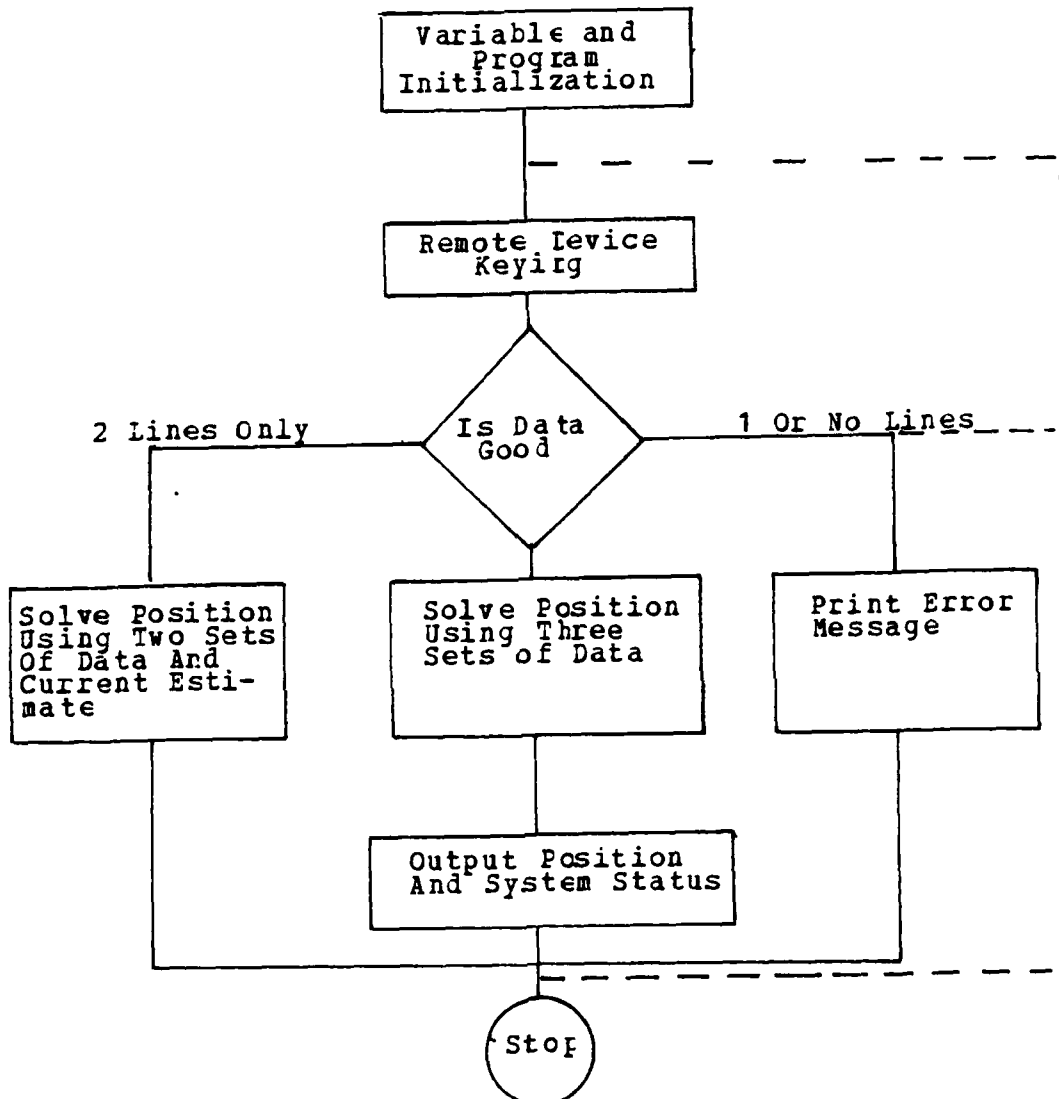


Figure 3.8 LM317, LM340T-6.0, and LM723CN Voltage Regulators

IV. SOFTWARE FOR THE SENTRY ROBOT'S POSITIONING SYSTEM

The structure of the software package is modular so that the portability of the package is increased. The package has two requirements which are unique to each microprocessor or robot design. First, the system needs a timing function which may be realized in a hardware or software clock. Second, the input/output port control is unique to each microprocessor design. Both of these variable segments reside in assembly language modules. As long as the appropriate control signals reach the external circuitry and the correct propagation times return to the main program, the major portion of the program needs no change.

Appendix C contains a listing of all of the programs. Figure 4.1 contains a flow diagram of the basic logic of the program. The main program initiates the majority of the system variables including those which are passed between the PL/I language programs and the assembly language programs. Following the above initialization, the PL/I program calls an assembly language program which is called INIT to initialize the Pro-Log 7804 Z80A Processor Counter/Timer Card which is resident in the STD 7000 work station. This hardware clock provides the requisite timing function. For this system, the clock acts as a down counter which counts .0002 second intervals. Whenever an interval expires, the counter/timer card generates an interrupt. The assembly language subroutine CLOCKSR services the interrupt by incrementing and storing the clock count. CLOCKSR also checks for a time-out condition and sets the appropriate error flag. The time-out and error flag will be discussed in more detail later.



* - - - Possible Alternate loops

Figure 4.1 Flow Diagram of Sentry Reference System Logic

After initializing the clock, the PL/I program serially keys the remote transmitter sites via the assembly language program KEY. This is the only parameter passing which occurs between two different types of language routines in this system. KEY starts the clock and passes the code to the output port which controls the radio transmitter. Subsequently, KEY goes into a loop which polls the input port and the time-out flag for a change of status. Once a change in status is detected, KEY stores the data and returns control to the PL/I program.

Between each measurement cycle, the transducers need debouncing to prevent erroneous signal detection. The WAIT subroutine provides this feature. WAIT tests the input port for the presence of a signal and upon finding no signal, it begins a decrementing counting loop which gives the transducer "ringing" sufficient time to die out. When the count reaches zero, WAIT returns control to the PL/I program.

Once the propagation times are returned to the PL/I program, the PL/I subroutine CONTIM converts the time of propagation into distance using equation 2.1. The remainder of the PL/I program implements the solution of the range-arc intersection problem which was described previously. Finally, the program passes the desired information to the CRT. In an actual robot, other programs or microprocessors would use this information for navigation.

Four features of this system require additional discussion. First, if the ultrasonic receiver fails to detect a signal, the system would stall indefinitely while it awaits a signal detection unless a provision is made to cover this possibility. The time-out flags handle this situation. In the clock service routine, the program compares the current value of the clock count with the value

which equates to the largest dimension of the room. If the clock count exceeds this value, the error flag is set. A propagation time larger than the maximum dimension of the grid places the robot outside the grid coordinates. Obviously, this can not happen.

The second feature involves the reactions to various combinations of data. If all three remote sites are detected (all time-out flags unset), the program solves the intersection for two pairs of range-arcs and compares these solutions to find the correct position. As stated previously, the worst case solution for any pair of range-arcs is two solutions with all positive coordinates. This causes an ambiguity. The second pair merely resolves the ambiguity.

When only two range-arcs are detected, the program computes their intersection and resolves any ambiguities by comparing the positions with the best estimate of the current position. The robot accepts the solution closest to its estimate of position. This is a reasonable assumption since the basic concept identifies a need for the robot to update its best estimate of position with this system.

The third feature consists of the cases where only one or no ultrasonic signals are detected. In the former case, the program prints an error message and the available data. In the latter case, the program prints only an error message. In an actual implementation, the navigation package should continue to guide the robot by dead-reckoning (DR) while the remote sites are polled again. This is true in both cases.

The final feature also concerns erroneous data. PL/I provides error routines which assist the microprocessor in recovering from calculation errors such as taking the square root of a negative number. When this occurs, the data is bad from one or more remote sites but no time-out flags are

set. In this program, the system returns to the CP/M operating system. In an actual implementation the system should repeat the polling process in the same manner as when there is one or no receptions of the ultrasonic signal. Again, the dead-reckoning duties continue.

In summation, each of these features attempts to counter a set of less than optimum conditions. In the event these conditions occur, the robot continues to DR itself which is the current convention or at worst, the robot can halt its motion until a good set of data is obtained. Looping through the remote keying segment of the program provides successive attempts to relocate the robot and eliminate the problem. With this being the case, recognizing an error becomes the major worry and not so much what the error is.

Finally, the passing of parameters between the PL/I and the assembly language modules needs further discussion. The discussion of this subject in [Ref. 8] is not adequate. Basically, whenever a PL/I routine calls an assembly language subroutine or function and passes the module a parameter, the PL/I program leaves an address in the HL register pair. The program follows the normal addressing scheme with L containing the lower byte of the address. This address points to the first entry in an array which contains the addresses of each of the parameters which are being passed. Again, the normal address convention prevails as the low order byte is stored first. This second address is the physical address which is assigned to the variable for the duration of the program. Any program which accesses this variable must use this address. As a result, all assembly language programs which return data to a PL/I program simply place the data at the physical address and return control to the PL/I program. The PL/I program has conventions for remembering the addresses of the variables. Figure 4.2 contains an example. [Ref. 8]

HL register upon entry into assembly language program (pointer to address list)	L	00H
	H	20H

Data at address 2000H (address of variable 1)	2000H	00H
	2001H	50H

Variable 1 location (1 byte)	5000H	xxH
------------------------------	-------	-----

Variable 1 location (2 bytes) (low order byte at 5000H)	5000H	xxH
	5001H	xxH

The assembly language program places data at addresses 5000H (1 byte) or 5000H and 5001H (2 bytes) to return data to the PL/I program. No other conventions are required.

If a second variable is passed, HL plus two contains the address.

HL after two increments	L	02H
	H	20H

Data at address 2002	2002H	FFH
	2003H	50H

Variable 2 location (1 byte)	50FFH	xxH
------------------------------	-------	-----

Figure 4.2 PL/I - Assembly Language Parameter Passing

One additional word of caution is necessary. The actual definition of the variable (eg., 8-bit integer, 15-bit real number) determines how many bytes are required to represent a variable. For example, a fifteen bit real number requires two bytes. Each byte is addressable. When a PL/I program passes such a variable, the program passes only the address of the lowest order byte. Within an assembly language program, each access of a two-byte variable requires the manipulation of two addresses.

Discussions about the program timing and system feasibility follow in the next chapter with the results of the testing phase.

V. SENTRY COORDINATE SYSTEM TEST RESULTS

Although the implementation was not optimum, the system works to within an accuracy of +/- one foot. Figure 5.1 contains a diagram of the room in which the test were conducted. Table II and III contain eight sample points with three successive attempts at each coordinate. The tables list the respective ranges to each site and the coordinate solutions. For example, position one in Table II contains four entries with five elements in each entry. The first row is the actual measurements and coordinates for position one. The remaining three entries are measurements and coordinates which were produced by the system. The first three columns list the distances from the various transmitter sites to the test position while the remaining two columns compose the X-Y coordinates. The worst case accuracy falls in position two with the X coordinate in error by 1.1 feet. The average positional error for all of the positions is 0.54 feet. Although an accuracy of one foot initially appears to be large, the system is actually more than adequate when one considers the entire sensor package which is located on a mobile robot. For both intruder and fire detection, sensor ranges come in tens of feet. Therefore, the robot does not need greater accuracy with which to report its findings. The robot's report contains more general data such as a room location or sector. On the other hand, the accuracy for production line work is far greater. During proximity navigation, a finite grid reference system fails by itself when faced with dynamic room designs. The remainder of this section presents the problems with cures which were faced during this implementation.

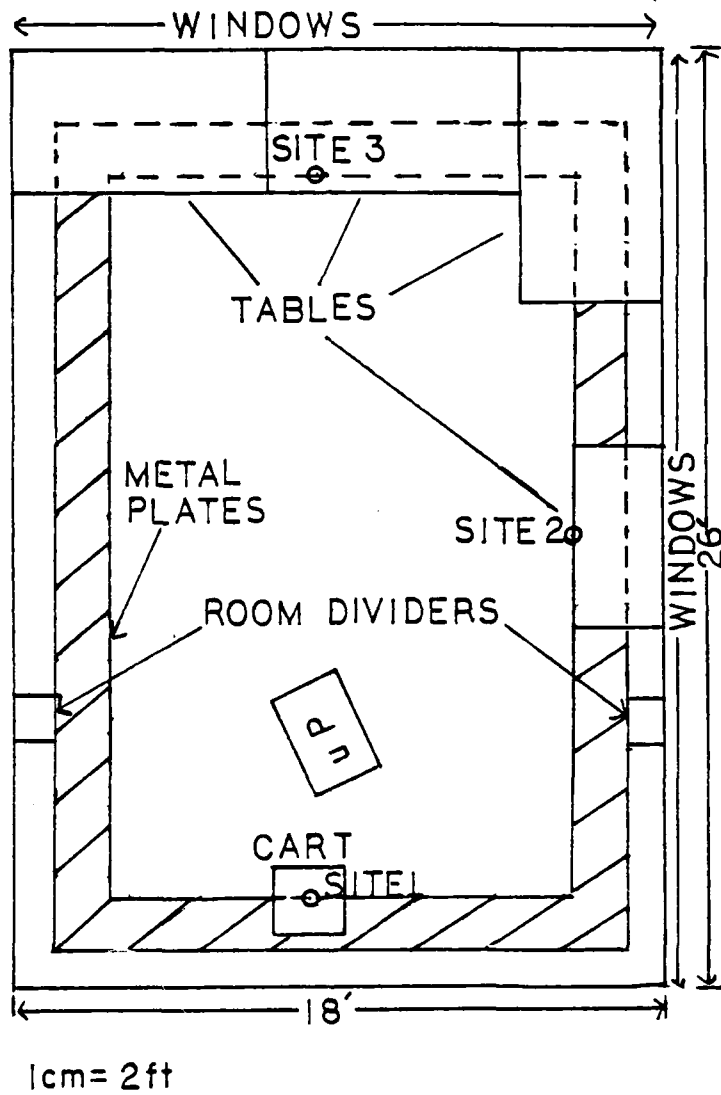


Figure 5.1 Testing Room Geometry

TABLE II
Sample Position Coordinates

Position One				
Site 1	Site 2	Site 3	X	Y
10	7	10	10	7*
10.7	7.6	10.2	10.3	7.6
10.3	7.4	9.7	10.3	7.4
10.1	7.4	10.6	10.1	7.4
Position Two				
10.8	11	10.8	10	11*
9.2	11.2	11.31	8.9	11.1
8.9	10.7	10.6	9.1	10.6
8.9	11.4	11.1	8.9	11.3
Position Three				
14.6	11.7	7.2	14	11*
15.6	11.9	7.3	14.6	10.9
13.6	11.5	6.9	13.6	10.9
13.0	11.3	7.2	13.1	10.8
Position Four				
5.7	6.7	16.5	4	3*
4.8	8.1	17.4	3.8	3.4
5.9	6.3	16.6	4.7	3.1
5.5	6.7	16.4	4.4	3.9

* Ideal measurements and coordinates

TABLE III
Sample Position Coordinates (cont'd)

Position Five				
Site 1	Site 2	Site 3	X	Y
7.2	11.7	14.6	6	11*
6.8	11.9	14	6.1	11.3
7.1	11.7	14.2	6.1	11.1
8.2	11.5	14.6	6.5	10.9
Position Six				
10.1	6	10.1	10	6*
10.1	6.8	10.1	10.6	6.9
10.5	6.4	10.6	9.9	6.4
10.3	6.18	11.1	9.6	6.1
Position Seven				
16.5	6.7	5.7	16	3*
16.7	7.3	4.9	16.3	4.1
18.7	7.1	5.3	16.7	2.1
16.7	7.1	6.2	16.7	2.5
Position Eight				
12.2	9.2	8.2	12	9*
11.9	9.2	8.6	11.7	9.1
12.1	8.9	8.1	12	8.8
12.6	9.7	8.6	12.4	8.8

* Ideal measurements and coordinates

Concerning the radio transmitter/receiver pairs, the problems were three-fold. First, the power surges which occur as the radio transmitter is turned on and off affect the ultrasonic receivers. This results in false echo detection. The power supply to the radio transmitter requires separate voltage regulation (LM317) and filtering capacitors (0.1 uF) to eliminate this problem. Additionally, the radio transmitter board, if not completely powered down, generates erroneous signals. Although these signals are not of the three required waveforms, the receivers process the erroneous signals and charge various capacitors to unknown levels. This condition destroys the reproducibility of the data. As a result, the system design incorporates the reed relays to control the power switching.

Secondly, the receiver's initial stage is a Colpitts oscillator. The oscillator directly interfaces with the antenna and the power supply with no buffering. As a result, receivers interfere with each other if they are much closer than ten feet or if they are not properly isolated from ground loops. Since the radio link works at twenty-seven mega-hertz, a .01 uF capacitor in parallel with a 1 uF capacitor provides sufficient filtering. As Douglas V. Hall states in [Ref. 9],

"Every capacitor, because of its leads and internal construction, has some series inductance. This series inductance together with the capacitance forms a series-resonant circuit...above the resonant frequency the net reactance is inductive, increasing as the frequency goes up. Because of the increasing inductive reactance above the resonant frequency, the capacitor is becoming less and less effective at 'shorting out' or filtering out high-frequency voltage transients...a combination of two capacitors thus provides filtering over a wide range of frequencies."

Thirdly, objects in proximity of the antennas disturb the respective electromagnetic propagation patterns. On an actual robot, this constraint forces the antenna on top of

the robot such that the antenna pattern is least affected by the robot body itself. Likewise, rooms with large numbers of metal objects or poorly shielded electrical cable alter these patterns. The large metal desks within the test room (Figure 5.1) caused serious degradation of the system when the transmitter was within two and one-half feet. However, since the radio and ultrasonic circuits were not fully shielded, the total possible reduction of these effects is not available.

Earlier discussions treated some of the problems of the ultrasonic devices. In addition to the requirement for full room coverage, two additional problems arise from the ultrasonic circuit board. First, the Polaroid board contains a five milli-second delay between the time the transmit cycle is initiated and the actual time of the ultrasonic transmission. This delay is also non-reproducible [Ref. 5]. While this did not grossly affect the system, future improvements in the system accuracy require an improvement in this aspect. The National Semi-Conductor LM1812 has no documented delay. Again, the LM1812 appears better for this application. Secondly, the gain control of the ultrasonic receivers is a major key to the success of the system. This implementation does not contain an automatic gain control. [Ref. 10] has such a circuit for the LM1812. During testing, the lack of a gain control necessitated recalibration of the time delays whenever the robot platform moved in excess of +/- three feet. With an automatic gain control, this calibration should not be necessary.

Finally, an older copy of [Ref. 10] (1977) contains a noise reduction tip for the LM1812 chip which is not found in later versions. "Inductive kicks" cause erroneous ultrasonic detections. A capacitor of approximately 30 pF between pins three and four reduces the bandwidth of the receiver and filters out these unwanted noise detections.

Initially, assumptions did not include allowances for circuit delay or delays in the radio link. However, each link has a different delay. Since each link bases its actuation upon a different signal, one expects a different delay for each link. As expected, the delays increase as the pulse code modulation rate decreases. The delays for the 3000 hertz, 500 hertz and constant carrier are .011, .071, and .098 seconds respectively. However, these parameters are unique to each circuit! The program implements these corrections via the constants, DEI1, DEI2, and DEL3.

In conjunction with the previous discussions on the circuit and radio link delays, the major time constraints come from the debouncing features of the program. When three signals are detected, the system returns a solution in a total time of 1.5 seconds. The above time includes one debouncing call before keying each of the remote sites. The WAIT program incorporates a combination of no signal and an arbitrary time delay to debounce the receivers. With more accurate testing, the time for debouncing is reducible. Of course, if one or more signals are not detected, the total system time becomes a function of the time-out conditions.

Although each circuit nominally has small current requirements, the short term requirements force careful voltage regulation at both the robot and at the remote sites. In addition to the power switching of the radio transmitter, the remote ultrasonic units also cause the same problems. For the Polaroid devices, the circuit requires six volts at two and one-half amps for one milli-second. A combination of an LM340T-6.0 voltage regulator and a 470 uF capacitor at the load prevent the ultrasonic circuit from disturbing the power supply. Interruption of the power supply disturbs both the radio receiver and the comparator/interface circuits.

Obviously, the number one problem of an ultrasonic system is obstructions. For this reason, the actual implementation of this design on a robot requires the radio antenna on top of the robot with the omni-directional ultrasonic receiver section directly underneath. The highest obstruction within the environment determines the minimum height at which the ultrasonic system may be located. Since the program needs a minimum of two unobstructed ranges for a solution, the line of sight between the robot and two of its remote sites must remain clear. As long as the program knows the location of the sites and the correct distance to the site, the coordinates are obtainable. Later chapters discuss alternatives for site location and systems with more than three remote sites.

VI. HARDWARE FOR THE MANIPULATOR REFERENCE SYSTEM

The differences between the sentry robots reference system and the manipulator's are few. However, these few changes are significant. Overall, the design requires fewer components and is, therefore, less complex.

The manipulator's reference system does not have the need for total autonomy which the mobile robot does. As a result, all of the peripheral sensors directly interface with the microprocessor. This eliminates the need for the radio links. Additionally, this eliminates all of the noise which is associated with radio transmissions.

Second, the microprocessor input/output ports reverse their importance. In the sentry design, the output port maintains the proper control of the three remote transmitter sites. The manipulator uses only one output to key a single transmitter which is located on the arm. The design again uses the Polaroid ultrasonic device as the transmitter. The interface circuitry consists of the normal TTL logic signal for the microprocessor and the comparator circuitry in Figure 6.1 which was used at the remote sites in the sentry design.

The input ports change from an omni-directional receiver input to three single multiplexed receivers. No actual wiring changes occur. However, the software services the input port differently. The next chapter further explains the changes in the software. There is one addition to the input circuitry. The Polaroid devices do provide access to the actual transmitter driving signal. The signal name is "XLOG". This signal allows the system to begin the system clock synchronously and also improves both accuracy and reproducibility of the data. Table IV contains the input/output port configurations.

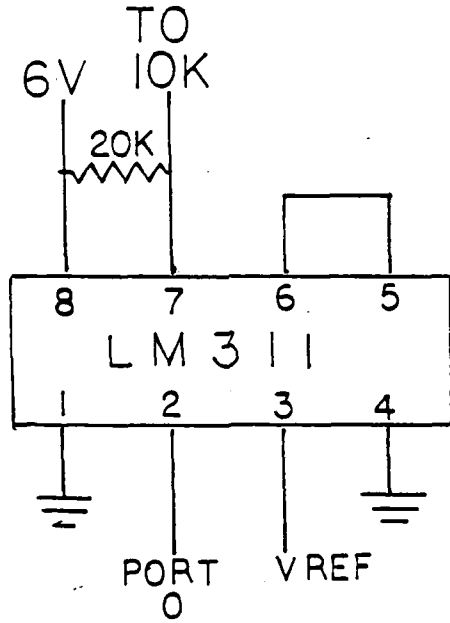
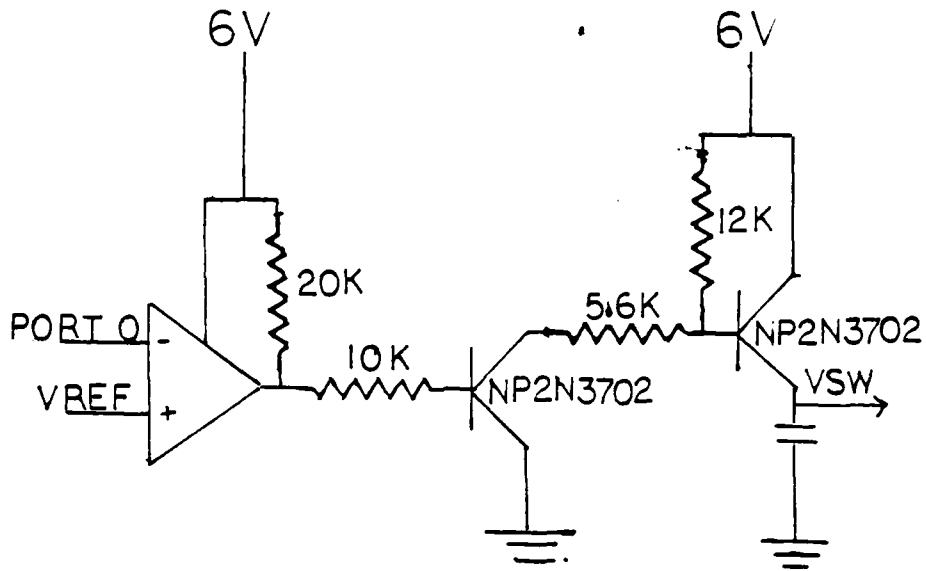


Figure 6.1 LM311 and Interface Circuitry (Manipulator)

TABLE IV
Port Assignments for the Manipulator

Output

Port	Device
0	Ultrasonic transmitter
1-7	No connection

Input

Port	Device
0	Receiver 1
1	Receiver 2
2	Receiver 3
3	XLOG signal
4-7	No connection

The remaining circuitry remains the same as in the sentry design. This includes the microprocessor workstation and the LM1812 ultrasonic circuits. A three-quarter inch plywood board provides the requisite support for the three ultrasonic receivers.

The software changes, which are discussed in the next chapter, likewise, are few. This circuitry provides a simple but effective system with which a reference system may be established. However, as will be discussed later, a final design will require much more detailed analysis of the transceiver design such that more uniform coverage may be obtained.

VII. SOFTWARE FOR THE MANIPULATOR REFERENCE SYSTEM

As stated previously, the sentry robot software is modular in the hope that it might be more portable. The manipulator software also follows this policy. In fact, the manipulator software originated from the sentry robot software with only a few changes. Figure 7.1 contains a block diagram of the system logic. Appendix D contains the complete program listings.

The program has the same major sections as the sentry robot's program. After initialization, the main program serially determines the distance between the three receivers and the transmitter which is located at the manipulator tip. This procedure involves three calls to the assembly language modules as in the previous programs. Next, the program checks for proper reception of data. If the data is bad, the program terminates with an error message. If the data is good, the program first calculates the line-of-sight departure angle and then the corresponding axis coordinate. Lastly, the program prints the appropriate data onto the CRT.

Obviously, the calculations which are made in the manipulator program are different from those of the sentry program. However, the debouncing and time/distance conversion modules are unchanged. The "ANGLE" and "CORD" modules replace the "POSIT3" and "POSIT2" modules. By comparing the two PL/I programs, their similarity becomes apparent.

Just as expected, the main changes in the programming occurred in the assembly language modules. While the parameter passing remained unchanged, both the clock and the receiver polling are different. To obtain a more accurate

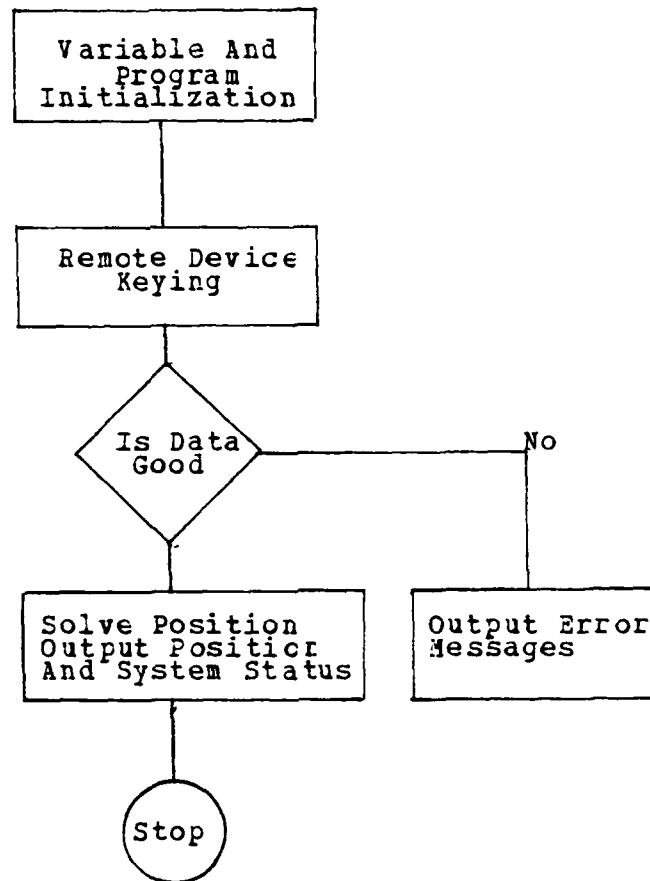


Figure 7.1 Flow Diagram of the Manipulator System Logic

clock, the program implements a software clock. A software clock uses a specific loop with a known execution time to increment a count of the real time. In this case, the loop increments in .00005 second intervals. Within the loop, the program polls the input port for the appropriate reception of the ultrasonic signal. The transmitter code of the sentry robot program becomes a receiver code in this system.

During the polling process, the microprocessor "AND's" the code with the input from the pprt. Table V contains these codes. The program tests for proper signal reception by checking the various flags which are set by the "AND-ing" operation. Upon signal reception or time-out, the program returns control and data to the PL/I main program.

TABLE V
Receiver Control Codes

0 0 0 1	Receiver 1 reception*
0 0 1 0	Receiver 2 reception
0 1 0 0	Receiver 3 reception
1 0 0 0	XLOG signal reception

* (The most significant bit is the left position.)

Since both the receivers and transmitter are directly interfaced to the microprocessor, reduced delays and more positive control are possible. Also, the XLOG signal which drives the ultrasonic transmitter interfaces with the microprocessor and initiates the software clock loop.

Finally, while this system encompasses fewer alternatives (no degraded modes), the system still must recover gracefully whenever no solution is possible. If any of the time-out flags are set, the system prints an error message. Also, if the data does not provide a solution (eg., the program attempts to take the square-root of a negative number), the PL/I error routines print the error messages. In an actual implementation, the program should simply loop for another attempt. Currently, the loop takes less than one second. These problems are further discussed in the test results which follow.

VIII. MANIPULATOR COORDINATE SYSTEM TEST RESULTS

With the elimination of the radio circuits, the accuracy and consistency of the distance measurements improve tremendously. When the transmitter is stationary and successive measurements are taken, the differences occur in the hundredths of a foot. The overall positional accuracy of the system increases to the tenths of a foot. Table VI and VII contain a listing of sample position data. The tables list the respective ranges to each sensor and the coordinate solutions. For example, position one in Table VI contains four entries with six elements in each entry. The first row is the actual measurements and coordinates for position one. The remaining three entries are measurements and coordinates which were produced by the system. The first three columns list the distances from the transmitter to the respective receivers while the remaining three columns compose the X-Y-Z coordinates. The worst case accuracy falls in position seven with the Y coordinate in error by 0.32 feet. The average positional error for all of the positions is 0.30 feet.

Even with this improvement in accuracy, the accuracy is not good enough for truly intricate assembly maneuvers. This is due in part to the lack of an automatic gain control. Whenever a significant change in the transmitter position or orientation occurs, the time response of the ultrasonic receiver changes as previously discussed in the test results on the sentry robot reference system. Secondly, further research is necessary to fully calibrate such a system. This calibration includes the definition of a reference point on the transducer from which the distances are measured. Additionally, the introduction of more

transducers for an increase in coverage adds to the calibration problems.

This design, however, proves the feasibility of such a system. By constraining the sensors to one plane, the manipulator enjoys a full range of unobstructed motion. The designer places the reference system wherever the location is most convenient. This includes floors, ceilings, walls or in the face of the manipulator base. Thus, the reference system may be absolute when it is stationary or relative when it is rotated on the base.

The system loop takes less than one second to complete. Again, the debouncing routine causes the longest delay. A reduction in this delay will optimize the system response since the WAIT routine is called three times. This design does not minimize this delay. Additionally, the delays for the three receivers are .023, .019, and .022 seconds for receivers one, two, and three respectively. These delays are also much less than the sentry robot design because they do not include the radio link delay with its pulse code modulation.

Most of the improvement in consistency of the data is attributable to the usage of the XLOG signal from the ultrasonic circuit board. This signal eliminates the need to estimate the five milli-second delay between the activation of VSW, the keying signal, and the actual transmission. Instead, the clock begins its count when the actual transmission signal originates. The LM1812 has no such comparable signal unless the transmission signal to the transducer head is used.

The system concept is feasible. With the gain control and further calibration, the desired system accuracy should be obtainable.

TABLE VI
Sample Position Coordinates (Manipulator)

		Position One			
Site 1	Site 2	Site 3	X	Y	Z
.833	.666	.750	.5	.833	.833*
.834	.666	.750	.65	.75	.80
.834	.723	.750	.69	.69	.81
.834	.666	.750	.65	.75	.81
		Position Two			
.567	.636	.756	.5	.5	.29*
.436	.55	.86	.32	.39	.21
.436	.55	.86	.32	.39	.21
.437	.55	.86	.32	.39	.21
		Position Three			
.52	.726	.527	.5	.333	.666*
.266	.836	.41	.25	.23	.5
.266	.893	.41	.25	.21	.5
.266	.836	.419	.25	.23	.5
		Position Four			
1.16	1.05	1.12	1.0	.833	.833*
1.16	1.05	1.07	.83	.92	1.07
1.16	1.05	1.12	.94	.915	.85
1.16	1.04	1.07	.83	.915	1.07
		Position Five			
1.03	1.07	1.15	1.0	.5	.3*
.88	.99	1.06	.81	.36	.17
.88	.94	1.07	.80	.46	.17
.87	.935	1.07	.80	.46	.17

* Ideal measurements and coordinates

TABLE VII
Sample Positions (Manipulator cont'd)

Position Six					
Site 1	Site 2	Site 3	X	Y	Z
1.08	1.16	1.0	1.0	.33	.83*
1.13	1.34	1.10	1.04	.12	.80
1.02	1.29	.932	.841	.05	.92
1.13	1.34	1.10	1.04	.01	.80
Position Seven					
1.0	1.11	1.04	1.0	.33	.5*
1.01	1.15	1.04	1.01	.29	.61
1.01	.98	1.10	.96	.65	.42
.91	1.10	.92	.89	.21	.60
Position Eight					
1.16	1.12	1.05	1.0	.83	.83*
1.13	1.0	1.01	.75	.92	1.1
1.13	1.0	1.07	.88	.92	.89
1.07	.89	1.01	.767	.99	.86
Position Nine					
1.06	1.08	1.0	1.0	.79	.54*
1.11	1.12	1.04	1.0	.56	.89
.99	1.01	.932	.91	.54	.83
.99	.95	1.10	.94	.79	.49
Position Ten					
1.19	1.12	1.37	1.0	.75	0.0*
1.27	1.18	1.38	1.1	.94	.15
1.3	1.24	1.5	.91	.98	-.22
1.33	1.29	1.5	1.02	.79	-.20

* Ideal measurements and coordinates

IX. CONCLUSIONS

The position control loop in robotics no longer needs to be a semi-closed loop. With an ultrasonic reference system in place, positive control of a mobile platform or manipulator is possible. Also, a reference system limits the cumulative error since it becomes a function of the coordinate system's recursive cycle. Armed with this information, a controller can implement more sophisticated guidance packages without complicated error-correcting software.

The cost of such a system is not prohibitive. The cost of the major components keep the total system hardware cost well below \$250.00. Even with more dedicated designs, the system cost, when mass-produced to lower the cost further, makes the reference system an expendable commodity. As a matter of fact, a single printed circuit board design is easily conceivable which would permit the maintenance operator to simply remove and replace the entire system.

Since the device need not be powered continuously, the system is a low power consumer. This is a mandatory requirement for an autonomous robot. Even the Polaroid device with its two and one-half ampere current requirement, operates successfully on batteries. Polaroid uses the Polapulse battery for power in its test and evaluation kits [Ref. 5]. Therefore, the power considerations are minimal.

While the two systems failed to achieve the desired increase in accuracy, closer cooperation with the manufacturers and additional research surely can improve the system accuracy. The chapter on recommendations includes hints by which the accuracy of the systems may be improved. Most importantly though, any implementation of these

ultrasonic devices dictates the use of an automatic gain control. Unfortunately, this research only identified this problem and the cure. Neither design contains this control.

Of course, the number one enemy of an ultrasonic system is obstructions. If the signal can not reach the receiver via its line of sight, the system does not work. Therefore, within reason, the concept requires sufficient reference transmitters/receivers to provide one-hundred percent coverage. As the number of transmitter/receiver sites increases, the complexity of the encoding schemes which control these sites increases. The sentry system requires at least two ranges at all times while the manipulator requires at least three. The environmental geometry dictates the actual number required.

Although the use of PL/I was expected to hamper the system speed, this was clearly not the case. The main time delay of either system is the various debouncing routines, the signal propagation, and the time delay of the external circuitry. Accordingly, the use of a faster (greater than two mega-hertz) or larger (greater than 8 bits) microprocessor will not provide a corresponding payoff in throughput. Until the other delays are reduced, the emphasis of future designs should focus upon optimizing these circuits and components.

As stated earlier, the Polaroid ultrasonic unit is an excellent device for ranging. The unit is easy to incorporate into designs and is extremely reliable. However, in this design, it fails to provide sufficient receive-only capability and beam width. The Panasonic EFR-01P40K2 transducer when incorporated with the LM1812 provides far better results, even though the LM1812 requires far more circuitry.

Finally, this thesis cannot over emphasize the necessity of following good "breadboard techniques". Although the

following list may seem obvious, the number of hours spent trouble shooting problems associated with the list was large.

1. Regulate all power supplies.
2. Use filtering capacitors liberally. Remember, it may require two in parallel to fully cover the requisite bandwidth.
3. Use heatsinks on all regulators.
4. Use only good breadboards or wire-wrap.
5. Follow good solder techniques.
6. Use coaxial cable when possible, especially when the distance is over six inches.
7. Do not underestimate the power of proper shielding.
8. Plan your layout carefully.
9. Color code your wiring.
10. Divide and conquer or segment and test.

Ultrasonic reference systems are realizable and feasible. This thesis reflects this. The future of robotics should no longer contain open loop designs. The reference systems can be inexpensive, accurate and low power. Closed loop control is possible.

X. RECOMMENDATIONS

Both the sentry and manipulator reference system designs provide a means by which the position control loop can be closed. As stated in LCDR Everett's thesis and numerous other texts and reference, separate microprocessors should implement these functions in parallel with the master microprocessor. However, the designs need further research and development.

Foremost in the sentry design is the need of a more consistent radio link. The problems which were attributed to the Colpitts oscillator can be eliminated with a better design and a change in operating frequency. Of course, a final implementation requires greater attention to circuit and cable shielding. Antenna placement is also important.

Second, the integration of the transducer into a functioning system while obtaining the necessary coverage is important. Unless an omni-directional transducer is available, any design will require multiple transducers working in parallel both at the transmitter and receiver locations. While this may be less of a problem in the sentry design, the manipulator, with its infinite orientations, is a most difficult problem. These applications might require an original transducer design.

Transducer sensitivity is controllable through external circuitry on the LM1812. Since the Polaroid device provides excellent control of the sensitivity out to a range of thirty-five feet (seventy feet including the return path), the range of current devices is more than adequate for the manipulator and the sentry robot for indoor uses. Any implementation which uses ultrasonics needs this sensitivity control, especially when the operating distances are in

close proximity. Without the automatic gain control, the time delays are variable and dependent upon the distance or signal strength which is unknown. Further, no device currently employs gain control on the transmitter. In a closed loop and close proximity system such as these, the transmitter gain control might use the estimated position as a reference by which the gain is varied. This permits the usage of constant gain receivers in a multiple receiver system.

Although the velocity of sound is not critically dependent upon temperature, it is none the less a function of temperature. The LM3911 easily provides temperature sensing which may be directed to the navigation microprocessor. With the correct ambient temperature, the microprocessor updates the speed of sound with which the time/distance conversions are made. A change from sixty-eight to sixty-nine degrees Fahrenheit translates to an additional one foot per second in velocity.

Although the basic calculations will not change, the error handling within each software package is unique to each application. While looping successively through the program is an obvious choice to correct a fault, the designer faces more important questions. Should movement continue? How fast can actions continue? How large are the tolerances? The designer's considerations include the machine and the environment. Finally, redundancy improves reliability and accuracy but it increases cost and complexity.

Even though the accuracy of the robot sentry and manipulator designs does not improve upon the current accuracies of systems without such references, the theoretical limits of accuracy which are a function of frequency are far better than current systems provide. For example, at fifty kilo-hertz, the wavelength is .272 inches

or .69 centimeters. At 200 kilo-hertz, the wavelength is .068 inches or .173 centimeters. Since the signal detection depends upon the rising edge of the incoming signal, accuracies of this magnitude are possible. Any delays are of no consequence as long as the delays are consistent.

With the final design perfected, the desired positive control is possible. Martin Weinstein in [Ref. 2] gives a very detailed description of a memory map with which navigation planning and obstacle avoidance might be performed. However, two additional items should be considered. First, any navigation scheme needs an endpoint or goal. Second, the memory map needs to incorporate visit time within its structure. The goal in a sentry's reconnaissance is one hundred percent coverage of the security area in a timely manner. Therefore, the interim goal becomes one of visiting the block with the oldest visit time. Once the goal is determined, the navigation unit decides the optimum path. As the robot transverses the chosen path, the navigation microprocessor updates the visit times of the blocks which are crossed plus ALL BLOCKS WHICH FALL INTO THE SECURITY OR FIRE DETECTION RANGE OF THE ROBOT'S SENSORS! This scheme allows the sentry to provide one-hundred percent coverage since a block which is not visited will eventually become an interim goal as its visit time grows to exceed all others. Weighting factors which are associated with high risk areas improve the revisit times to these areas.

LCDR Everett mentions path planning routines such as the computer game "Othello". Also, in [Ref. 11], four papers addressed this issue. They are listed in the bibliography. However, the work by Keirse et al. at the University of Florida and at Hughes Research Laboratories looks very promising. In their paper [Ref. 12], this group substantiates their work in tactical navigation. More

specifically, their work includes search graph construction and heuristic searches for path planning.

The Zenith Corporation currently markets an ET-18 Robot Remote Control Unit which provides remote control for the HERO Robot. The remote control unit also interfaces with microprocessors through the RS-232 loop. This system is a potential testbed for both the reference system and various robot control algorithms. Currently, robots, like the HERO, do not keep track of their positions nor do they have sufficient memory onboard to store the programs for such research. This experimental set-up provides both control and virtually unlimited memory while the cost and complexity of the robot is held to a minimum. With this arrangement, the ultrasonic transmitter and receivers should exchange places for the best results and ease of implementation. Consequently, the remote receivers interface to the microprocessor via hard-wired connections. Also, the microprocessor keys the ultrasonic transmitter on the robot through the radio remote control.

The literature on manipulator control is far more abundant. However, no model completely closes the position control loop. At best, the controllers deal only with the motors and the motor feedback. The systems do not directly measure the errors which were discussed previously. These systems rely upon error models or no models at all.

While the three dimensional system herein does locate the tip of the manipulator, the system fails to identify the orientation of the gripper or the rotating base. A manipulator with six degrees of freedom needs additional reference data to fully orient the system. A possible solution consists of optical calibration lines which can be accurately located and read on the manipulator. The optical sensors and cable are small and light. They will not place any further weight restrictions upon the arm. Also,

today's optical components interface directly onto printed circuit boards and integrated circuits. These advantages support this approach very strongly.

The three dimensional reference system provides data from which position and velocity information may be determined. While this information is in cartesian coordinates, the conversion to other coordinate systems is available and well documented. Microprocessors handle these conversions nicely. As a result, the myriad of manipulator control schemes have a better testbed upon which they can be tested. More importantly, the manipulator will no longer require the teaching cycles which currently program industrial robots. Instead, the potential is present for the manipulator, when it is given the various coordinates, to determine both the commands and the routing for each task. Much of today's robotic research deals with high-level hierarchical languages for robotic control. With these languages, more general commands such as "pick up block" release the programmer from the tedious task of providing the exact commands for all of the motors involved in the motion.

In summary, with the position loop finally closed in the robotics field, more positive control is obtainable. An ultrasonic system will not handle every environment. However, with additional research, the environments for which ultrasonics are suitable will be identified. The goal of increased accuracy will become a reality.

APPENDIX A

DERIVATION OF SENTRY POSITION EQUATIONS

This derivation is a general solution for an intersection of two range arcs which are associated with two different reference points.

Coordinate definitions:

$X = XAOF \ +/- \ XA = XBOF \ +/- \ XB$ where $_OF$ are the offsets of
 $Y = YAOF \ +/- \ YA = YBOF \ +/- \ YB$ the reference points from the
origin and $X_$ or $Y_$ is the
respective distances to the
position from the reference
points in vector notation.

Range arc definitions:

$AR^{**2} = XA^{**2} + YA^{**2}$ where $_r$ is the radius of the
 $BR^{**2} = XB^{**2} + YB^{**2}$ arc

Equate X's

$XAOF \ +/- \ XA = XBOF \ +/- \ XB$

Solve the range arc equations for the vector magnitude $X_$
and substitute into the above equation

$XAOF \ +/- \ (AR^{**2} - YA^{**2})^{**.5} = XBOF \ +/- \ (BR^{**2} - YB^{**2})^{**.5}$

Rearranging

$\ +/- \ (AR^{**2} - YA^{**2})^{**.5} = XBOF - XAOF \ +/- \ (BR^{**2} - YB^{**2})^{**.5}$

Define $DIF = XBOF - XAOF$ and square both sides of the equation

$AR^{**2} - YA^{**2} = DIF^{**2} \ +/- \ (2 * DIF * (BR^{**2} - YB^{**2})^{**.5}) + BR^{**2} - YB^{**2}$

Solve for +/- (2*DIF*(BR**2 - YE**2)**.5) and square both sides of the equation again

$$4*(DIF**2*(BR**2 - YB**2)) = AR**4 + YA**4 + YB**4 + BR**4 + DIF**4 + 2*(-AR**2*YA**2 + AR**2*YB**2 - YB**2*DIF**2 - YA**2*YB**2 - AR**2*BR**2 + YA**2*BR**2 - AR**2*DIF**2 + YA**2*DIF**2 - YB**2*BR**2 + ER**2*DIF**2)$$

Substituting for YA and YB, and expanding the terms yields the quadratic equation which was discussed in chapter 2 (eqn 2.2)

$$COEF1*Y**2 + COEF2*Y + COEF3 = 0$$

where

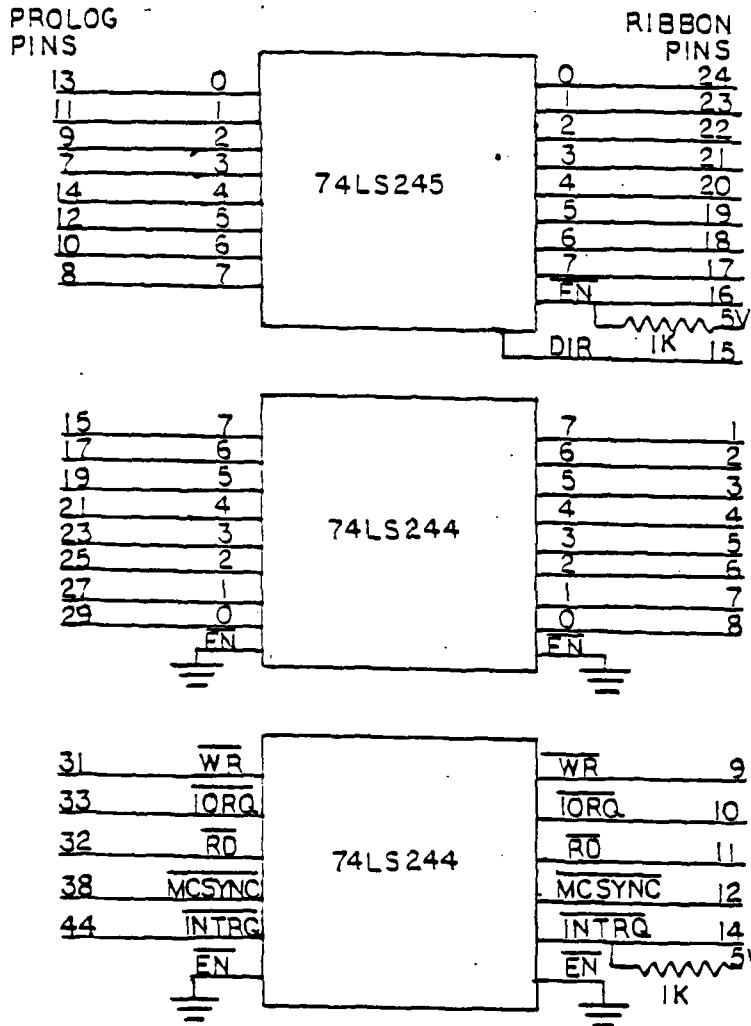
$$COEF1 = 4*(YAOF**2 + YBOF**2 - 2*YAOF*YBOF + DIF**2)$$

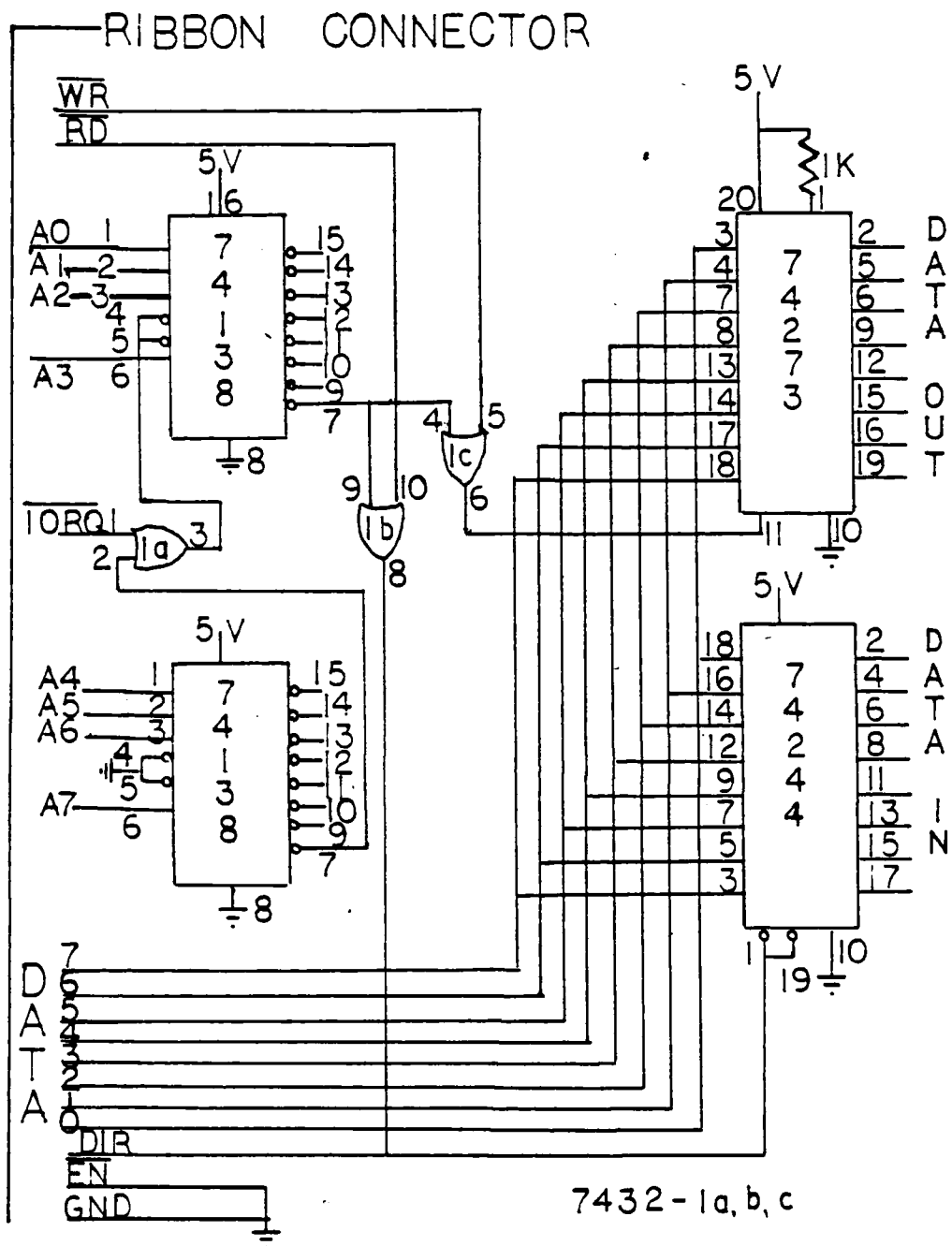
$$COEF2 = 4*(AR**2*YAOF - YAOF**3 - YBOF**3 - AR**2*YBOF + YAOF*YBOF**2 + YAOF**2*YBOF - BR**2*YAOF - DIF**2*YAOF + BR**2*YBOF - DIF**2*YBOF)$$

$$COEF3 = AR**4 + BR**4 + DIF**4 + YAOF**4 + YBOF**4 + 2*(AR**2*YBOF**2 - AR**2*YAOF**2 - YAOF**2*YBOF**2 - AR**2*BR**2 + BR**2*YAOF**2 - AR**2*DIF**2 + DIF**2*YAOF**2 - BR**2*YBOF**2 + DIF**2*YBOF**2 - BR**2*DIF**2)$$

APPENDIX B

PROTECTION CIRCUITRY FOR THE PRC-LOG STD 7000





APPENDIX C

SENTRY ROBOT REFERENCE SYSTEM SOFTWARE

```
NAVROB:
PROC OPTIONS (MAIN);
/*WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS THE SOFTWARE*/
/*PORTION OF AN ULTRASONIC NAVIGATION SYSTEM FOR AN AUTONOMOUS */
/*SENTRY ROBOT. THE SYSTEM PROVIDES AN X-Y GRID COORDINATE */
/*SYSTEM BY WHICH THE ROBOT MAY NAVIGATE. THE SYSTEM USES AN RF*/
/*TRANSMITTER TO REMOTELY KEY THREE ULTRASONIC TRANSMITTERS AND*/
/*MEASURES THE ELAPSED PROPAGATION TIME TO DETERMINE THE DIS- */
/*TANCE TO THE KNOWN LOCATION OF THE TRANSMITTERS. THE SOFT- */
/*WARE FINDS THE INTERSECTION OF THE THREE RANGE ARCS TO DETER-*/
/*MINE THE ACTUAL POSITION OF THE ROBOT. THE SOFTWARE IS A */
/*COMBINATION OF PLI AND ASSEMBLY LEVEL LANGUAGE PROGRAMS WHICH*/
/*ARE COMPILED SEPARATELY AND THEN LINKED TOGETHER. */
```

```
DCL KEY ENTRY (FIXED (7),FIXED (15),FIXED (7));
DCL STARCK ENTRY();
DCL INIT ENTRY();
DCL CLSR ENTRY();
DCL WAIT ENTRY ();
DCL (T1,T2,T3) FIXED (15);
DCL (F1,F2,F3,XMIT1,XMIT2,XMIT3) FIXED (7);
DCL (AR,BP,CR,XAOF,XBOF,XCOF,DFL1,DFL2,DFL3) FLOAT;
DCL (YAOF,YEOF,YCCF,X,Y,CURX,CURY) FLCAT;
```

```
/*VARIABLE INITIALIZATION*/  
XAOFF=2.0;  
YAOFF=7.;  
XBOFF=10.;  
YBOFF=0.;  
XCOFF=20.0;  
YCCFF=7.;  
XMIT1=22;  
XMIT2=17;  
XMIT3=26;  
DEL1=353.12;  
DEL2=486.41;  
DEL3=55.7;  
CURX=0.0;  
CURY=0.0;
```

```
/*KEY THE THREE RECEIVERS*/  
CALL INIT;  
CALL WAIT;  
CALL KEY (XMIT1,T1,F1);  
CALL WAIT;  
CALL KEY (XMIT2,T2,F2);  
CALL WAIT;  
CALL KEY (XMIT3,T3,F3);  
PUT SKIP LIST (F1= ,F1);  
PUT SKIP LIST (F2= ,F2);  
PUT SKIP LIST (F3= ,F3);
```

```

/*IS THE DATA GOOD? NO! SEND ERROR MESSAGE*/
IF ((F1+F2+F3)>1) THEN CALL ERRMESS();
ELSE;

```

```

/*CONVERT THE PROPAGATION TIMES INTO DISTANCE*/
IF (F1=0) THEN CALL CONTIM (T1,AR,DFL1);
ELSE;
IF (F2=0) THEN CALL CONTIM (T2,BR,DEL2);
ELSE;
IF (F3=0) THEN CALL CONTIM (T3,CR,DEL3);
ELSE;

```

```

/*IF ALL DATA IS GOOD FIND SOLUTION WITH THE RANGFS*/
IF ((F1+F2+F3) = 0) THEN

```

```

CALL POSIT3

```

```

(AR,BR,CR,X,Y,XACF,XBOF,XCOF,XCOF,YAOF,YEOF,YCCF);
/*OTHERWISE, USE THE TWO GOOD RANGES*/

```

```

ELSE

```

```

IF ((F1+F2)=0 & F3=1) THEN

```

```

CALL POSIT2 (AR,BR,X,Y,XAOF,XBOF,XCOF,YAOF,YEOF,CURX,CURY);

```

```

IF ((F1+F3)=0 & F2=1) THEN

```

```

CALL POSIT2 (AR,CR,X,Y,XAOF,XCOF,YAOF,YCOF,CUPX,CURY);

```

```

IF ((F2+F3)=0 & F1=1) THEN

```

```

CALL POSIT2 (BR,CR,X,Y,XBOF,XCOF,YBOF,YCOF,CJFX,CJFY);
ELSE;

```

```

;
```

```

/*OUTPUT THE DATA*/
PUT SKIP LIST ('THE X COORDINATE IS ',X,' FEET. ');
PUT SKIP LIST ('THE Y COORDINATE IS ',Y,' FEET. ');
IF (F1=1) THEN
  PUT SKIP LIST ('TRANSMITTER A NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM A IS ',AR,' FEET ');
;
IF (F2=1) THEN
  PUT SKIP LIST ('TRANSMITTER B NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM B IS ',BR,' FEET ');
;
IF (F3=1) THEN
  PUT SKIP LIST ('TRANSMITTER C NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM C IS ',CR,' FEET ');

```

```

CONTIM:
  PROC(ITIM,DIST,DEL);
/*THIS PROCEDURE CONVERTS THE INTEGER TIME COUNT INTO ACTUAL*/
/*DISTANCE IN FEET. THE SPEED OF SOUND IS 1134.9 FT./SEC. */
  DCL (ITIM) FIXED;
  DCL (TIM,DIST,DEL) FLOAT;
  TIM = ITIM;
  DIST = (TIM-DEL)*1134.9*.2002;
  RETURN;
END CONTIM;

```

```

ERRMESS:
PROC;
/*THIS PROCEDURE OUTPUTS AN ERROR MESSAGE INDICATING THAT AT */
/*LEAST TWO TRANSMITTERS WERF NOT RECEIVED.*/
PUT SKIP LIST ('FAILED TO RECEIVE SUFFICIENT DATA TO');
PUT LIST (' COMPUTE THE POSITION');
RETURN;
END ERRMESS;

POSIT3:
PROC (R1,R2,R3,X,Y,X1OF,X2OF,X3OF,Y1OF,Y2OF,Y3OF);
/*THIS PROCEDURE CALCULATES THE UNAMBIGUOUS POSITION GIVEN CCOOD*/
/*DATA FROM ALL THREE TRANSMITTERS.*/
DCL (R1,R2,R3,X,Y,X1OF,X2OF,X3OF,Y1OF,Y2OF,Y3OF) FLOAT;
DCL (X1,X2,X3,X4,Y1,Y2,Y3,Y4) FLOAT;
DCL (ABXOF,ACXOF,ABYOF,ACYOF,TOLER) FLOAT;
TOLER=2.0;
AEXOF=X2OF - X1OF;
ACXOF=X3OF - X2OF;
ABYOF=Y2OF - Y1OF;
ACYOF=Y3OF - Y2OF;
CALL ROOTS (AFXOF,Y1OF,Y2OF,R1,R2,Y1,Y2);
CALL ROOTS (ACXOF,Y2OF,Y3OF,R2,R3,Y3,Y4);
CALL ROOTS (ABYOF,X1OF,X2OF,R1,R2,X1,X2);
CALL ROOTS (ACYOF,X2OF,X3OF,R2,R3,X3,X4);
IF (ABS(X1-X3)<TOLER & X1<0.) THEN X=(X1+X3)/2.;
IF (ABS(X1-X4)<TOLER & X1<0.) THEN X=(X1+X4)/2.;
IF (ABS(X2-X3)<TOLER & X2<0.) THEN X=(X2+X3)/2.;
IF (ABS(X2-X4)<TOLER & X2<0.) THEN X=(X2+X4)/2.;
IF (ABS(Y1-Y3)<TOLER & Y1<0.) THEN Y=(Y1+Y3)/2.0;
IF (ABS(Y1-Y4)<TOLER & Y1<0.) THEN Y=(Y1+Y4)/2.0;
IF (ABS(Y2-Y3)<TOLER & Y2<0.) THEN Y=(Y2+Y3)/2.0;
IF (ABS(Y2-Y4)<TOLER & Y2<0.) THEN Y=(Y2+Y4)/2.0;
RETURN;
END POSIT3;

```

```

POSIT2:
PROC (R1,R2,X,Y,X1OF,X2OF,Y1OF,Y2OF,CURX,CURY);
/*THIS PROCEDURE CALCULATES THE UNAMBIGUOUS POSITION GIVEN GOOD*/
/*DATA FROM TWO TRANSMITTERS AND A REASONABLE ESTIMATE OF THE */
/*CURRENT POSITION. */
DCL (R1,R2,X,Y,X1OF,X2OF,Y1OF,Y2OF,CURX,CURY) FLOAT;
DCL (Y1,Y2,X1,X2,DIFX,DIFY) FLOAT;
DIFX=X2OF - X1CF;
DIFY=Y2OF - Y1OF;
CALL ROOTS (DIFX,Y1OF,Y2OF,R1,R2,Y1,Y2);
CALL ROOTS (DIFY,X1CF,X2CF,R1,R2,X1,X2);
IF (X1<0.) THEN X=X2;
IF (X2<0.) THEN X=X1;
IF (Y1<0.) THEN Y=Y2;
IF (Y2<0.) THEN Y=Y1;
IF (X1<2. & X2<0.) THEN
  BFGIN;
  IF (ABS(CURX-X1)<ABS(CURX-X2)) THEN X=X1;
  ELSE X=X2;
  END;
IF (Y1<0. & Y2<0.) THEN
  BEGIN;
  IF (ABS(CURY-Y1)<ABS(CURY-Y2)) THEN Y=Y1;
  ELSE Y=Y2;
  END;
RETURN;
END POSIT2;

```

```

ROOTS:
  PROC (DIF,OF1,OF2,R1,R2,VAL1,VAL2);
  /*THIS PROCEDURE CALCULATES THE ROOTS OF A QUADRATIC EQUATION*/
  DCL (DIF,OF1,OF2,R1,R2,VAL1,VAL2,A,B,C) FLOAT;
  A=4.*(OF1**2+OF2**2) - 8.*(R1**2)*OF2 + 4.*DIF**2;
  B=-4.*(OF1**3+OF2**3)+4.*(R1**2)*(OF2-OF1)-(DIF**2)*(OF1+OF2);
  C=R1**4+R2**4+DIF**4+OF1**4+OF2**4-2.*(R1**2)*(OF1**2);
  C=C+2.*(R1**2)*(OF2**2)-(OF1**2)*(OF2**2)-(R1**2)*(R2**2));
  C=C+2.*(R2**2)*(OF1**2)-(R1**2)*(DIF**2)+(DIF**2)*(OF1**2));
  C=C+2.*(DIF**2)*(OF2**2)-(R2**2)*(OF2**2)-(R2**2)*(DIF**2));
  VAL1=(-B+SQRT(B**2 - 4.*A*C))/(2.*A);
  VAL2=(-B-SQRT(B**2 - 4.*A*C))/(2.*A);
  RETURN;
END ROOTS;
END NAVROB;

```

; WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS ONE PROCEDURE
 ; IN A SOFTWARE PACKAGE TO SUPPORT AN ULTRASONIC NAVIGATION SYSTEM.
 ; FOR AN AUTONOMOUS SFENTRY RCBOCT. THIS PROCEDURE INITIALIZES
 ; THE CLOCK TO COUNT PROPAGATION TIME OF THE ULTRASONIC SIGNAL.
 .750

```

CSEG
PUBLIC  INIT
EXTERN CLSR
INIT:  DI
      IK 1
      LD A,0C3H
      LD (1035H),A
      LD HL,CLSR
      LD (1039H),HL
      RFT
      END
      ; DISABLE INTERRUPTS
      ; INTERRUPT MODE 1 - Z82
      ; LOAD A REGISTER WITH JMP COMMAND NEEDED
      ; FOR CLOCK SERVICE ROUTINE
      ; STORE JMP AT INTERRUPT BRANCH LOCATION
      ; LOAD ADDRESS OF CLSR IN REGISTER HL
      ; LOAD ADDRESS FOR JMP COMMAND AT BRANCH
  
```



```

;WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS ONE PROCEDURE
;IN A SOFTWARE PACKAGE TO SUPPORT AN ULTRASONIC NAVIGATION SYS-
;TEM FOR AN AUTONOMOUS SENTRY ROBOT. THIS PROCEDURE KEYS THE
;REMOTE TRANSMITTERS AND RECORDS THE PROPAGATION PATH TIME.

```

```

.2EH
CSEG
PUBLIC KEY
EXTRN EXTRN
EXTRN EXTRN
EXTRN EXTRN
KEY: LD EXTRN
INC EXTRN
LD EXTRN
EX EXTRN
CALL EXTRN
LD EXTRN
OUT EXTRN
LOOP: LD EXTRN
CF EXTRN
JP EXTRN
IN EXTRN
AND EXTRN
CP EXTRN
JP EXTRN

COUNT
STARCK
C, (HL)
HL
E, (HL)
DF, HL
STARCK
A, (BC)
(0FFH), A
A, (TFLAG)
ZEH
NZ, FIV
A, (2FFH)
ZEH
Z, LOOP

;MOVE ADDRESS OF XMIT(X) INTO
;BC. ADDRESS=2 BYTES
;XMITX=1 BYTE
;PROTECT POINTER WHILE CALLING
;INIT WHICH USES HL
;START CLOCK
;BC POINTS TO XMIT BYTE
;KEY REMOVE TRANSMITTER CYCLE
;READ TIME-OJI VALUE INTO A
;TEST FLAG FOR TIME-OUT
;EXIT IF TIME-OUT
;PORT IN FOR ULTRASONIC DETECT
;ZERC UPPER TWO BITS OF A
;TEST FOR ULTRASONIC DETECT LO
;LOOP IF NO SIGNAL

```

```

FIN:  DI
      LP
      OUT
      EX
      INC
      LD
      INC
      LD
      INC
      LD
      INC
      LD
      INC
      LD
      INC
      INC
      LD
      LD
      LD
      LD
      LD
      LD
      RFT
      END

```

```

A,22H
(2FFH),A
DE,HL
HL
C,(HL)
HL
B,(HL)
HL
E,(HL)
HL
D,(HL)
HL,COUNT
A,(HL)
(BC),A
HL
EC
A,(HL)
(EC),A
HL,TFLAG
A,(HL)
(DE),A

```

```

;
;
; REPLACE POINTER IN HL
; HL POINTS TO T(X) LC BYTE
; LO ADDRESS OF BYTE
;
; HI ADDRESS OF BYTE
; HL POINTS TO F(X) BYTE
; LO ADDRESS OF BYTE
;
; HI ADDRESS OF BYTE
; LOWER BYTE OF COUNT
; PLACE COUNT INTO TIME LOCATION
;
; UPPER BYTE OF COUNT
; UPPER BYTE OF T(X)
;
; HL POINTS TO TFLAG BYTE
;
;

```

```

; WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS ONE PROCEDURE
; IN A SOFTWARE PACKAGE TO SUPPORT AN ULTRASONIC NAVIGATION SYS-
; TEM FOR AN AUTONOMOUS SENTRY ROBOT. THIS PROCEDURE STARTS THE
; CLOCK WHICH COUNTS THE PROPAGATION PATH TIME.

```

```

Z80
CSFG
PUBLIC COJNT
PUBLIC STARCK
PUBLIC TFLAG
TFLAG: DB 1
COUNT: DW 1
STARCK: LD A,34H
        OUT (0F3H),A
        LD A,32H
        OUT (0F0H),A
        LD A,21H
        OUT (0F2H),A
        LD HL,COUNT
        LD A,22H
        LE (HL),A
        INC HL
        LD (HL),A
        LF (TFLAG),A
        LD A,02H
        OUT (0F4H),A
        FI
        RFT
        END

; ERROR FLAG FOR PROP TIME-OUT
; CLOCK COUNT OF .2 MILLI-SECONDS
; CTC CHANNEL 0 CONTRCL MODE 2
; OUT TO PORT OF TIMER
; LOAD INTERVAL COUNT VALVE 400T=.2MSEC.
; SEND LOWER BYTE FIRST 02H
; UPPER BYTE
;
; INITIALIZE COUNT TO 02
; COUNT IS TWO BYTES LONG
;
;
; INITIALIZE TFLAG TO 02
; LOAD ENABLF BIT
; ENABLF INTERRUPT HARDWARE

```

; WILLIAM M. TUNKIN, LT, USN WROTE THIS PROGRAM AS ONE PROCEDURE
 ; IN A SOFTWARE PACKAGE TO SUPPORT AN ULTRASONIC NAVIGATION SYS-
 ; TEM FOR AN AUTONOMOUS SENTRY ROBOT. THIS PROCEDURE SERVICES THE
 ; CLOCK AFTER AN INTERRUPT BY THE CLOCK.

```

    .7E0
    CSSEG
    PUBLIC CLSR
    EXTRN COJNT
    EXTRN TFLAG
    EXTRN AF,AF'
    CLSR:
    IN A,(2F4H)
    AND Z,DEPART
    XOR A
    OUT (2F4H),A
    LD A,Z
    OUT (2F4H),A
    LD A,Z
    OUT (2F4H),A
    LD HL,COUNT
    INC (HL)
    LD A,0FFH
    CP (HL)
    JP NZ,CCMP
    INC HL
    INC (HL)
    ; PRESERVE REGISTERS
    ;
    ; TEST CHANNEL ~ BIT
    ; IF 0 CHANNEL EMPTY, BRANCH TO RETURN
    ;
    ; CLEAR INTERRUPT
    ; BY SETTING THE BIT TO 1
    ; RESET TO GET A POSITIVE EDGE
    ;
    ; INCREMENT COUNTER REPRESENTING
    ; THE PROPAGATION TIME
    ;
    ; IF CARRY, UPPER BYTE MUST BE INC
    ;
    ; INCREMENT THE UPPER BYTE ON CARRY
  
```

```

COMP: LD HL,COUNT
      INC HL
      LD A,0AFH
      CP (HL)
      JP NZ,DEPART
      LD HL,COUNT
      LD A,2FFE
      CP (HL)
      JP NZ,DEPART
      LD A,21H
      LD (TFLAG),A
      DEPART: EXX
              EX EI
              RI TI
              END
;
;
; HL,COUNT
;
; CHECK TO SEE IF UPPER BYTE MATCHES
;
; IF NO MATCH, CONTINUE
; IF MATCH, SEE IF LO BYTES MATCH
;
; SET FLAG - TIME-OUT HAS OCCURRED
;
;

```

; WILLIAM M. TUNKIN, LT, USN WRCTF THIS PROGRAM AS
 ; THE SOFTWARE PORTION OF AN ULTRASONIC REFERENCE
 ; SYSTEM IN ROBOTICS. THIS MODULE PROVIDES DE-
 ; BOUNCING FOR THE ULTRASONIC RECEIVERS.

.782

```

CSEG
PUBLIC WAIT
LD C,2FFH
LD B,2FFH
IN A,(2FFH)
AND 0FH
CP 2FH
JF NZ,LOOP2
DEC C
LD A,2H
CP C
JP NZ,LOOP2
DEC B
LD A,2H
CP B
JP NZ,LOOP2
RET
END

;
;
; CHECK INPUT PORT FOR NO SIGNAL
;
;
; LOOP IF SIGNAL IS STILL ACTIVE
; BEGIN DECREMENTING DFIAY LOOP
;
;
;
;
; RETURN TO MAIN PROGRAM
  
```

APPENDIX D

MANIPULATION REFERENCE SYSTEM SOFTWARE

```

NAVAEM:      PRCC OPTICNS (MAIN);
/*WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS THE SOFTWARE*/
/*PORTION OF AN ULTRASONIC REFERENCE SYSTEM FOR A MANIPULATOR */
/*ARM. THE SYSTEM PROVIDES AN X-Y-Z GRID COORDINATE SYSTEM */
/*FROM WHICH ARM CONTROL IS POSSIBLE. THE SOFTWARE FINDS THE */
/*INTERSECTION OF THE THREE RANGE ARCS BY USING THE LAW OF */
/*COSINES AND THE SPECIAL GEOMETRY OF THE REFERENCE SFNSCRS. */

DCL VEYAR2 ENTRY (FIXED (7),FIXED (15),FIXED (7));
DCL WAIT FNTRY ();
DCL (T1,T2,T3) FIXED (15);
DCL (F1,F2,F3,XMIT1,XMIT2,XMIT3) FIXED (7);
DCL (AR,EP,CP,AE,AC,DEL1,DEL2,DEL3) FLOAT;
DCL (XAO1,YAO1,ZAO1,B,A,X,Y,Z,AX,AY,AZ) FLOAT;

/*VARIABLES INITIALIZATION*/
XACF=0.7;
YAOF=.3533333;
ZAOF=.5;
AE=.5;
AC=.3333333;
XMIT1=1;
XMIT2=2;
XMIT3=4;
DEL1=171.5;
DEL2=171.14;
DEL3=171.56;

```

```

/*KEY THE TRANSMITTER THREE TIMES*/
CALL WAIT;
CALL KEYPAR2 (XMIT1,I1,F1);
CALL WAIT;
CALL KEYPAR2 (XMIT2,I2,F2);
CALL WAIT;
CALL KEYPAR2 (XMIT3,I3,F3);
PUT SKIP LIST ('F1=',F1,'T1=',T1);
PUT SKIP LIST ('F2=',F2,'T2=',T2);
PUT SKIP LIST ('F3=',F3,'T3=',T3);

/*IS DATA GOOD? NO! SEND ERROR MESSAGE*/
IF ((F1+F2+F3)>0) THEN CALL ERRMESS();
ELSE;

/*CONVERT TIME TO DISTANCE*/
IF (F1=0) THEN CALL CONTIM (I1,AR,DEL1);
ELSE;
IF (F2=0) THEN CALL CONTIM (T2,BR,DEL2);
ELSE;
IF (F3=0) THEN CALL CONTIM (T3,CR,DEL3);
ELSE;

/*IF THE DATA IS GOOD SOLVE THE POSITION*/
IF ((F1+F2+F3) = 0) THEN
CALL ANGLE (AR,BR,AB,R);
IF ((F1+F2+F3) = 0) THEN
CALL ANGLE (AR,CR,AC,A);
IF ((F1+F2+F3) = 0) THEN
CALL COORD (AR,E,YACF,Y,AY);
IF ((F1+F2+F3) = 0) THEN
CALL COORD (AR,A,ZACF,Z,AZ);
IF ((F1+F2+F3) = 0) THEN
AX=SQRT(AR**2 - AY**2 - AZ**2);
IF ((F1+F2+F3) = 0) THEN
X=AX + XACF;

```



```

/*OUTPUT THE DATA*/
PUT SKIP LIST ('THE X COORDINATE IS ',X,' FEET. ');
PUT SKIP LIST ('THE Y COORDINATE IS ',Y,' FEET. ');
PUT SKIP LIST ('THE Z COORDINATE IS ',Z,' FEET. ');
IF (F1=1) THEN
  PUT SKIP LIST ('TRANSMITTER A NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM A IS ',AR,' FEET ');
;
IF (F2=1) THEN
  PUT SKIP LIST ('TRANSMITTER B NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM B IS ',BR,' FEET ');
;
IF (F3=1) THEN
  PUT SKIP LIST ('TRANSMITTER C NO RECEPTION! ');
ELSE
  PUT SKIP LIST ('DISTANCE FROM C IS ',CR,' FEET ');
;

CONTIM:
PROC(ITIM,DIST,DFL);
/*THIS PROCEDURE CONVERTS THE INTEGER TIME COUNT INTO ACTUAL*/
/*DISTANCE IN FEET. THE SPEED OF SOUND IS 1134.3 FT./SEC. */
DCL (ITIM) FIXED;
DCL (TIM,DIST,DEL) FLOAT;
TIM = ITIM;
LIST = (TIM-DEL)*1134.9*.00025;
PUT SKIP LIST ('DISTANCE= ',LIST);
RTURN;
END CONTIM;

```

```

ERRMESS: PROC;
/*THIS PROCEDURE QUIPITS AN ERROR MESSAGE INDICATING THAT AT */
/*LIASST TWO TRANSMITTERS WIRE NOT RECEIVED. */
PUT SKIP LIST ('FAILED TO RECEIVE SUFFICIENT DATA TO');
PUT LIST (' COMPUTE THE POSITION');
RETURN;
END ERRMESS;

```

```

ANGLE: PROC (R1,R2,DIST,THETA);
/*THIS PROCEDURE CALCULATES THE ANGLE OF DEPARTURE OF */
/*THE RANGE LOS FROM THE GRID COORDINATE AXIS */
DCL (R1,R2,DIST,THETA,ARG) FICAT;
PUT SKIP LIST ('ANGLE CALLED');
ARG = .5*(1./DIST)*(DIST**2 + R1**2 - R2**2);
THETA = ACCOS(ARG);
RETURN;
END ANGLE;

```

```

COORD: PROC (R,PHI,OFF,TERM,DISTA);
/*THIS PROCEDURE CALCULATES THE GRID AXIS INTERSECTION*/
/*OF THE BASE OF THE CONE DEFINED BY THE ROTATION CF */
/*THE RANGE APCS ABOUT THE AXIS */
DCL (R,PHI,OFF,TERM,DISTA) FLOAT;
PUT SKIP LIST ('COORD CALLED');
DISTA=R*COS(PHI);
TERM=OFF + DISTA;
RETURN;
END COORD;
END NAVARM;

```

WILLIAM M. DUNKIN, LT, USN WROTE THIS PROGRAM AS ONE PROCEDURE
 IN A SOFTWARE PACKAGE TO SUPPORT AN ULTRASONIC POSITIONING SYS-
 TEM FOR A MANIPULATOR. THIS PROCEDURE KEYS THE TRANSMITTERS AND
 EFFECTS THE SIGNAL RECEPTION. A SOFTWARE CLOCK DOES THE
 INTERVAL TIMING.

```

.Z80
CSFG
PUBLIC KEYAR2
TFLAG: DE 1
COUNT: DW 1
KEYAR2: LD C,(HL)
        INC HL
        LD B,(HL)
        EX DF,HL
        LD A,225
        LD HL,COUNT
        LD (HL),A
        INC HL
        LD (HL),A
        LD HL,TFLAG
        LD (HL),A
        LD L,C
        LD H,B
        LD A,2FFH
        OUT (2FFH),A
        IN A,(2FFH)
        AND 25H
        JF N7,DELAY
; ERROR FLAG FOR PROP TIME-OUT
; CLOCK COUNT OF 20 MICROSEC.
; MOVE ADDRESS OF XMIT(X) INTO
; BC. ADDRESS=2 BYTES
; XMITX=1 BYTE
; PROTECT POINTER
; INITIALIZE THE COUNT
;
;
;
; INITIALIZE THE ERROR FLAG
;
;
; LOAD CODE TO TURN ON XMITTR
; SEND TO PORT
; THIS ALLOWS THE CIRCUIT SIGNAL.
; XLOG TO KEY THE SOFTWARE CLOCK WHICH
; INCREASES THE REPEATABLE ACCURACY
; OF THE TIMING MEASUREMENTS. IT
; COMES FROM THE ULTRASONIC BOARD

```

```

LOOP:  IN      A,(0FFH)
        LD      L,C
        LD      H,B
        AND    (HL)
        CP     (HL)
        JP     NZ,FIN
        NOP
        NOP
        NOP
        LD     HL,COUNT
        INC   (HL)
        LD     A,0FFH
        CP    (HL)
        JP    NZ,LOOP
        LD     A,01H
        LD     (IFLAG),A
        LD     A,00H
        LD     (0FFH),A
        OUT
        EX    HL
        INC   HL
        LE   C,(HL)
        INC  HL
        LF   B,(HL)
        INC  HL
        LD   E,(HL)
        INC  HL
        LF   D,(HL)

        ; CHECK FOR RECEIVE SIGNAL
        ; LOAD ADDRESS OF RECEIVER CODE
        ; IN HL TO TEST FOR RECEPTION
        ;
        ; UPON RECEIPT STORE PARAMETERS
        ; AND EXIT
        ; NOP'S ADD DELAY TO
        ; MAKE CLOCK INTERVAL EVEN
        ; INCREMENT THE CLOCK COUNT
        ;
        ; TEST FOR TIME-OUT
        ; IF NO TIME-OUT CONTINUE
        ; IF TIME-OUT SET FLAG
        ;
        ; REPLACE POINTER IN HL
        ; HL POINTS TO P(X) LO BYTE
        ; LO ADDRESS OF BYTE
        ;
        ; HL ADDRESS OF BYTE
        ; HL POINTS TO P(X) BYTE
        ; LO ADDRESS OF BYTE
        ;
        ; HI ADDRESS OF BYTE

```

```

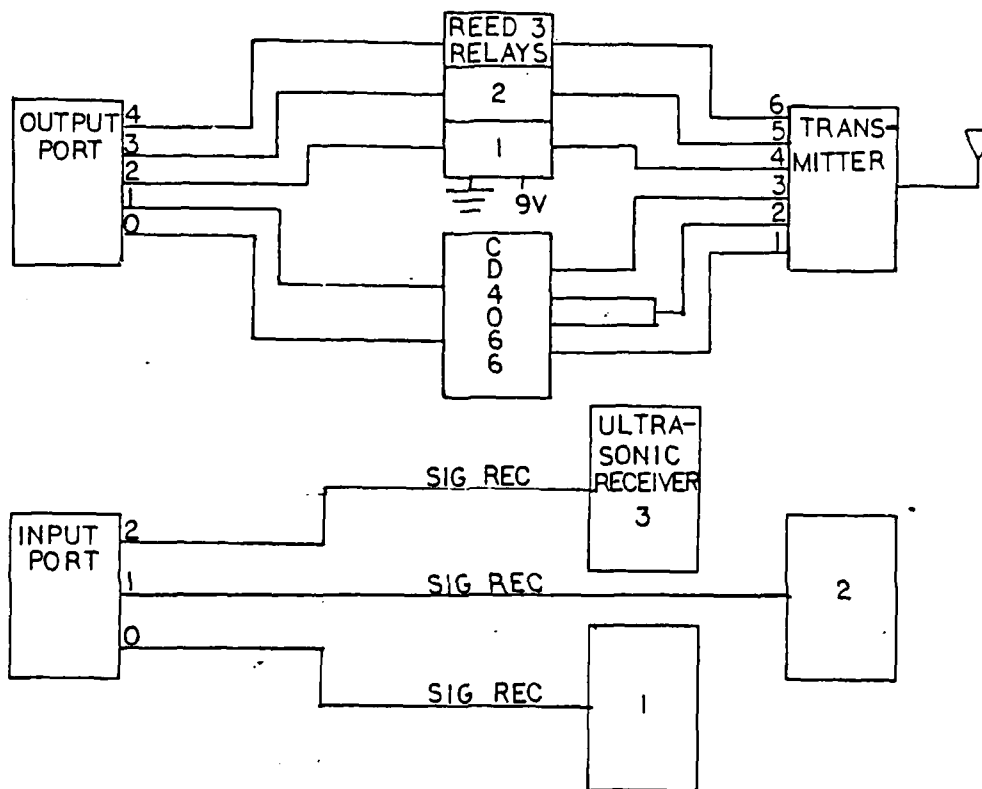
LD      HL,COUNT
LD      A,(HL)
LD      (BC),A
INC     HL
INC     BC
LD      A,(HL)
LD      (BC),A
LD      HL,TFLAG
LD      A,(HL)
LD      (DE),A
RET
END

; LOWER BYTE OF COUNT
; PLACE COUNT INTO TIME LOCATION
;
; UPPER BYTE OF COUNT
; UPPER BYTE OF T(X)
;
; HL POINTS TO TFLAG BYTE
; LOAD TIME-CUT FLAG VALUE INTO F(X)

```

APPENDIX E

MASTER WIRING DIAGRAMS



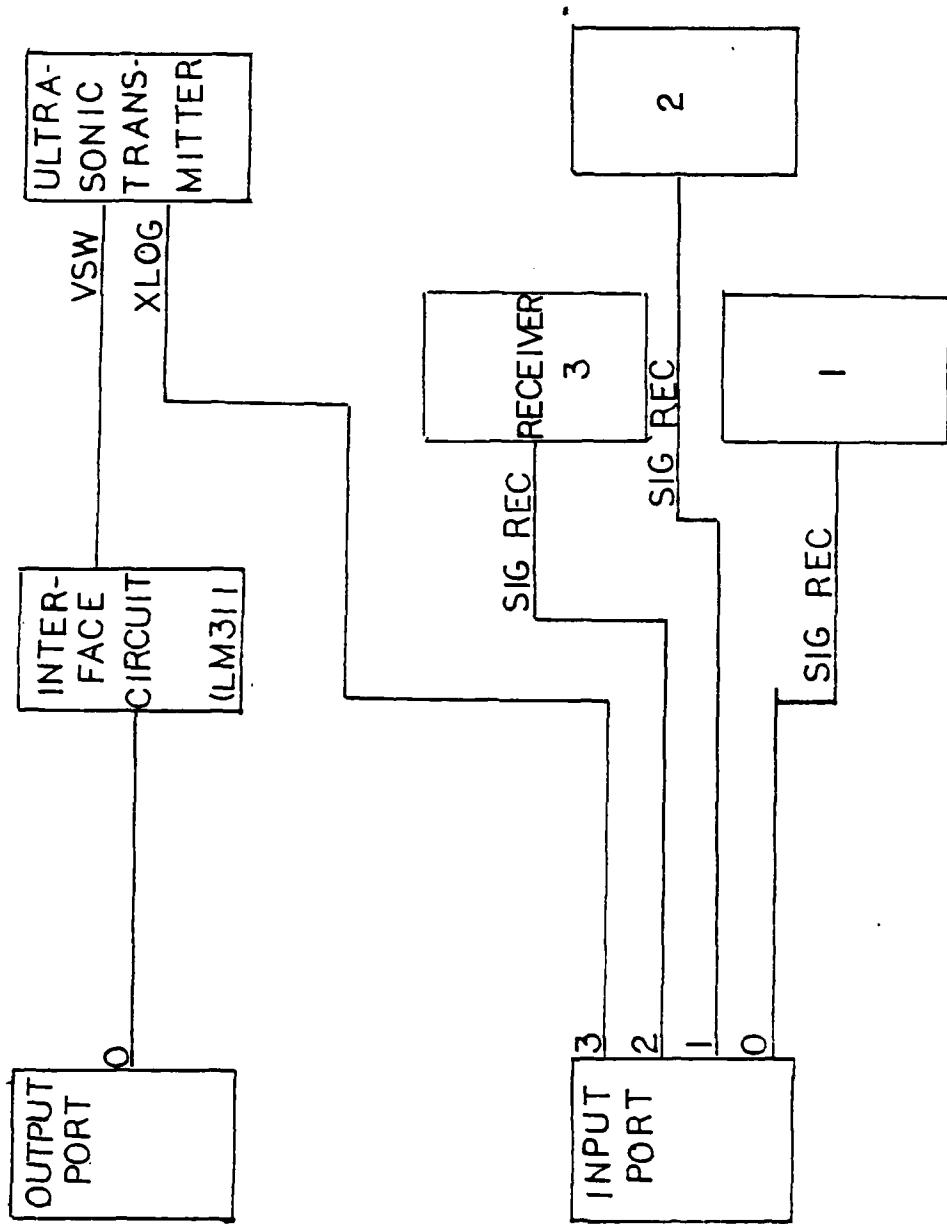
ULTRA-
SONIC
TRANS-
MITTER

VSW

INTER-
FACE
CIRCUIT
(LM311)

RECEIVER COLLECTOR
VOLTAGE

RECEIVER COLLECTOR
VOLTAGE



LIST OF REFERENCES

1. Everett, Hobart R. A Microprocessor Controlled Autonomous Sentry Robot, Master of Science Thesis, Naval Postgraduate School, Monterey, California, 1982.
2. Weinstein, Martin B. Android Design, Hayden Book Company, Inc., Rochelle Park, New Jersey, 1981.
3. Albus, James S. "Industrial Robot Technology and Productivity Improvement," Robotics: The Past Is Prologue, Robotics - Research and Business Opportunities, IEEE Computer Society Press, Silver Spring, Maryland, 1982.
4. Archer, Data Sheet, Remote Control Module Pair Manual, Fort Worth, Texas, 1984.
5. Polaroid Corporation, Data Sheet, Ultrasonic Ranging System Manual, Cambridge, Massachusetts, 1981.
6. Swack, Richard. "Application Note 1," Polaroid Corporation, Cambridge, Massachusetts, 1984.
7. Panasonic Industrial Company, Data Sheet, Ultrasonic Ceramic Microphones, Secaucus, New Jersey, 1984.
8. Digital Research, Link-80 Operator's Guide, Monterey, California, 1980.
9. Hall, Douglas V. Microprocessors and Digital Systems, McGraw-Hill Book Company, New York, 1983.
10. National Semi-Conductor Corporation, Linear Circuit Data Book, Santa Clara, California, 1982.
11. IEEE International Conference On Robotics, IEEE Computer Society Press, Silver Spring, Maryland, 1984.
12. Keirsev, D.M., Koch, E., McKisson, J., Meystel, A.M., and Mitchell, J.S.B., "Algorithm of Navigation for a Mobile Robot", IEEE International Conference On Robotics, IEEE Computer Society Press, Silver Spring, Maryland, 1984.

BIBLIOGRAPHY

Chavey, R. and Meystel, A. "Structure of Intelligence for an Autonomous Vehicle," IEEE 1984 International Conference on Robotics, IEEE Computer Society Press, Silver Spring, Maryland, 1984.

Digital Research, PL/I-80 Application Guide, Monterey, California, 1980.

Digital Research, PL/I-80 Language Manual, Monterey, California, 1980.

Harmon, S.Y. "Comments on Automated Route Planning in Unknown Terrain," IEEE 1984 International Conference on Robotics, IEEE Computer Society Press, Silver Spring, Maryland, 1984.

Siv, P. "Road Map Production System for Intelligent Mobile Robot," IEEE 1984 International Conference on Robotics, IEEE Computer Society Press, Silver Spring, Maryland, 1984.

Zaks, Rodney, Programming the Z80, Sybex, Inc., Berkeley, California, 1982.

AD-A155 481

ULTRASONIC POSITION REFERENCE SYSTEMS FOR AN AUTONOMOUS
SENTRY ROBOT AND A ROBOT MANIPULATOR ARM(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA W M DUNKIN MAR 85

2/2

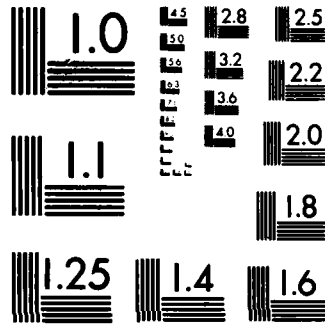
UNCLASSIFIED

F/G 9/2

NL



END
FILMED
DTIC



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

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2	
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2	
3. Prof. Harriett Rigas, Ph.D Department Chairman, Code 62Rr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1	
4. George J. Thaler, Dr. Eng. Professor, Department of Electrical and Computer Engineering, Code 62Tr Naval Postgraduate School Monterey, California 93943	5	
5. John P. Powers, Ph.D Professor, Department of Electrical and Computer Engineering, Code 62Po Naval Postgraduate School Monterey, California 93943	2	
6. Srbijanka Turajlic, Ph.D Professor, Department of Electrical and Computer Engineering, Code 62Tc Naval Postgraduate School Monterey, California 93943	1	
7. David Smith, Ph.D Professor, Department of Mechanical Engineering Naval Postgraduate School, Code 69 Monterey, California 93943	1	
8. LCDR Hobart R. Everett, USN Special Assistant for Robotics Naval Sea Systems Command Navy Department Washington, DC 20362	2	
9. Richard Swack Polaroid Corporation Cambridge Massachusetts 02139	1	
10. LT William M. Dunkin, USN 1551-B Apache Drive Chula Vista, Ca. 92010	2	
11. Mr. Don Redmond XEBC 595 Pastoria Avenue Sunnyvale, CA 94086	1	
12. Mr. Tom Hickox XEBC 595 Pastoria Avenue Sunnyvale, CA 94086	1	

END

FILMED

8-85

DTIC