ELECTRONICS SYSTEMS DEVELOPMENT AND INTEGRATION FOR A SECOND GENERATION ROBOT SUBMARINE

by

JOSEPH ANTHONY CARNEVALE, JR.

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF OCEAN ENGINEER and THE DEGREE OF MASTER OF SCIENCE IN NAVAL ARCHITECTURE AND MARINE ENGINEERING at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY May 1980

© Joseph Anthony Carnevale

The author hereby grants to M.I.T. permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author .................................................................

Department of Ocean Engineering
May 1980

Certified by ................................................................. Thesis Supervisor

Accepted by ................................................................. Chairman, Departmental Graduate Committee

ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
AUG 5 1980

LIBRARIES
ELECTRONICS SYSTEM DEVELOPMENT AND INTEGRATION FOR A SECOND GENERATION ROBOT SUBMARINE

by

Joseph Anthony Carnevale, Jr.

Submitted to the Department of Ocean Engineering on May 9, 1980 in partial fulfillment of the requirements for the Degree of Ocean Engineer and the Degree of Master of Science in Naval Architecture and Marine Engineering

ABSTRACT

Development is continued on the second generation robot submarine in the Department of Ocean Engineering. The electronics system is broken down into a finite number of modules. The details of the modules are described and the major components specified. Designs for the central processing unit, memory, multiply unit, and various interfaces are presented. Communications and interrupt processing are delineated. The interactions of the modules are studied and a bus system selected. A chassis mounting system is utilized to facilitate development. Finally, the systems and software required to support the submarine are described. Support is divided between laboratory and field systems. The goal is to fix the circuitry and material required to have the robot submarine functioning on its own in the ocean environment. In many cases, possible improvements on initial designs are suggested.

Thesis Supervisor: A. Douglas Carmichael
Title: Professor of Power Engineering
ACKNOWLEDGMENTS

I'd like to thank all the people who helped develop the hardware for the submarine. Special thanks go to Mike and Clive who advised me from the start.

Thanks to Jim who dutifully listened to the ups and downs and Mary who lived for a year in the world of digital logic.

Most of all, I'd like to express my appreciation to Professor Carmichael who continuously guided me.

And thanks to Cheryl, keeper of the blue copies, who typed this thesis.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>2</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>3</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>8</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>9</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>10</td>
</tr>
<tr>
<td>1.2 Mission Goals</td>
<td>12</td>
</tr>
<tr>
<td>1.3 System Requirements</td>
<td>13</td>
</tr>
<tr>
<td>1.3.1 Digital Autopilot</td>
<td>14</td>
</tr>
<tr>
<td>1.3.2 Sensor Systems</td>
<td>14</td>
</tr>
<tr>
<td>1.3.3 Power Supply</td>
<td>15</td>
</tr>
<tr>
<td>2. ELECTRONIC SYSTEMS DESIGN</td>
<td>18</td>
</tr>
<tr>
<td>2.1. System Description</td>
<td>18</td>
</tr>
<tr>
<td>2.1.1 System Configuration</td>
<td>19</td>
</tr>
<tr>
<td>2.1.2 Chassis</td>
<td>19</td>
</tr>
<tr>
<td>2.1.3 Bus</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Microcomputer System</td>
<td>25</td>
</tr>
<tr>
<td>2.2.1 Central Processing Unit</td>
<td>25</td>
</tr>
<tr>
<td>2.2.2 Memory</td>
<td>28</td>
</tr>
<tr>
<td>ROM</td>
<td>28</td>
</tr>
</tbody>
</table>
RAM ........................................... 29
Memory Mapped Functions .............. 32
2.2.3 Multiply Unit ...................... 33
2.2.4 Interrupt Processor .............. 35
2.2.5 Digital Interface ................. 38
2.2.6 Analog Interface .................. 39
2.2.7 Serial Interface ................. 41
2.3 Sensors ................................... 43
2.3.1 Pitch and Roll ..................... 44
2.3.2 Depth .................................. 44
2.3.3 Compass .............................. 45
2.3.4 Bottom Finding Sonar ............ 47
2.3.5 Collision Avoidance Sonar ....... 50
2.4 Communication Sonar ............... 51
2.5 Fin Servomotor Systems ............ 52
2.6 Propulsion Motor .................... 54
2.7 Batteries .............................. 56
2.8 Power Supplies and Regulation .... 57
2.9 Ballasting System Control .. ...... 61
2.10 Side-Scan Sonar ..................... 62
2.11 Emergency Safeguards ............. 64
LIST OF FIGURES

2-1  Electronics System Configuration
2-2  Microcomputer Memory Map
2-3  Interrupt Processing Block Diagram
2-4  Analog Interface Block Diagram
2-5  Bottom Finding Sonar Block Diagram
2-6  Fin Servomotor System Diagram
2-7  Propulsion Motor Switching Block Diagram
2-8  Power Regulation and Distribution System
2-9  Simple RLC Regulation Circuit
2-10 Side-Scan Sonar System Block Diagram
3-1  Support Systems
3-2  Mission Programming Block Diagram
A-I-1  CPU/Memory Design
A-II-1 Multiply Unit Interface Design
A-III-1 Interrupt Processing Circuit
A-IV-1 Digital Interface Circuit
A-V-1  Communications Processing Flowchart
LIST OF TABLES

1-1  Maximum Allowable Fluctuations in Motion
1-2  Sensor Resolution Requirements
1-3  Estimated Power Allocations
3-1  Field Support Unit Functions
3-2  Debugging Functions
3-3  Bench Test Requirements
3-4  Open Water Test Requirements
A-I-1 CPU/Memory Board Organization
A-IV-1 Digital Interface Coding
A-V-1 Single Transmission Commands
A-V-2 Multiple Transmission Commands
CHAPTER 1

INTRODUCTION

The second generation robot submarine under development by the MIT Ocean Engineering Department is currently under construction. It is an improved version of the first generation submarine successfully completed by the Department. The second generation robot submarine is designed for untethered operations. To accomplish this, the vehicle will include an on-board microcomputer capable of controlling all of the submarine's functions. When released, the submarine must be capable of totally independent operations requiring the ability to navigate along a predetermined course and collect data at prescribed intervals. The microcomputer must also be capable of insuring the physical safety of the submarine.

Although designed for totally independent operations, the submarine will have interactive capabilities. A communications sonar link will allow the user to control the submarine from the surface. Control will be limited by the time delay in sonar transmission and lack of visual contact in deeply submerged operations.
The second generation robot submarine is intended to be a search and survey vehicle. Its primary payload will be a side-scan sonar. The submarine will be capable of carrying other instrumentation demanded by continuing development. This very stable platform will provide an excellent means of remote data collection in the ocean environment.

1.1 BACKGROUND

In 1974, a first generation robot submarine was designed, built, and tested as part of the Ocean Engineering Summer Laboratory. This first submarine used an analog autopilot. The original microcomputer for the submarine had a central processing unit designed and built in the laboratory. Memory was limited to 4096 sixteen bit words.

This first generation submarine eventually passed from tethered to untethered operation. Simple missions were conducted measuring temperature gradients in the ocean. Inefficient use of the small on-board computer, lack of communication with the submarine, and numerous other problems were brought to light. With the experience gained from the first submarine, the decision was made to proceed in the development of an improved, second generation robot submarine.
In 1978, work on the second generation robot submarine proceeded in parallel in three areas. Design and construction of the pressure hull, skin, and mechanical systems was undertaken (see G. M. Davis, Reference 2). A study of control system synthesis was conducted by Inoue (see Reference 3) and the preliminary design of the electronics system as well as the initial development of control algorithms was accomplished by Saylor (see Reference 1).

Saylor's thesis develops the requirements for the electronics system of the second generation submarine. Included in his work is the performance requirements of the microcomputer system, sensors, fin servomotor systems, compass, power supply, and sonars. All major systems are outlined in varying degrees of detail. His work also includes the development, design, and architecture of the digital autopilot to be used in the second generation submarine.

The work of Saylor is the basis of this endeavor. This work is intended to be an extension of that before it and every effort will be made to avoid repetition or duplication of material presented by Saylor. The focus of this report is the circuitry
designed to implement the ideas already developed. The system requirements and project goals presented in the following sections are derived from those presented by Saylor. These sections are intended to maintain continuity in the ongoing submarine project.

1.2 MISSION GOALS

The second generation robot submarine is designed to be an untethered, stable platform for search and survey. The submarine must incorporate interactive control as well as totally independent operational capability. Construction of the submarine must be completed in a timely manner. The current state of availability of many electronics components will disallow their use. Four criteria have been established as design goals.

1. The weight and volume of the submarine is limited to that easily handled by four persons. This goal has been met with the completion of the skin and pressure hull.

2. The submarine must be capable of maintaining a constant depth and course, and a level attitude. This goal will be met with the successful implementation of the digital autopilot. Final
assessment of this goal cannot be made until the submarine operates in an ocean environment.

3. Normal mission operation will be 2 hours with an extended mission capability of 15 hours. Meeting the normal mission goal should be feasible without major modifications. A 15 hour capability will require different batteries and, perhaps, a much lower resistance skin.

4. The submarine must be capable of carrying a side-scan sonar unit. Additionally, it must be flexible enough in its configuration to accommodate additional payloads without major modifications.

1.3 **SYSTEM REQUIREMENTS**

Every area of the electronics system will be required to perform at predetermined levels to meet the goals established in the previous section. The following requirements have been previously established. Any questions as to their origin may be answered by referring to the works of Saylor (Reference 1) or Inoue (Reference 3).
1.3.1 **Digital Autopilot**

The digital autopilot is charged with maintaining a constant course, speed, and depth as well as a level attitude. Any variation will result in distortions of the side-scan sonar plots. Variations of no more than 1% in heading, depth, roll, pitch, and speed results in the maximum allowable fluctuations presented in Table 1-1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Maximum Allowable Fluctuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading</td>
<td>± 1 degree</td>
</tr>
<tr>
<td>Depth</td>
<td>0.1 meter</td>
</tr>
<tr>
<td>Roll</td>
<td>2.9 degrees</td>
</tr>
<tr>
<td>Pitch</td>
<td>8.1 degrees</td>
</tr>
<tr>
<td>Speed</td>
<td>0.03 knots</td>
</tr>
</tbody>
</table>

**TABLE 1-1** Maximum Allowable Fluctuations in Motion

1.3.2 **Sensor Systems**

To meet the requirements demanded of the digital autopilot the sensor systems must maintain a specified level of accuracy. Table 1-2 presents those levels of accuracy in terms of units as well as bits. With the exception of the depth sensor, 12 bit resolution
on analog-to-digital conversion and 8 bit resolution
on digital-to-analog conversion will meet the require-
ments. Twelve bit accuracy for the depth sensor will
allow readings within 0.02 m over the 100 m range.
This will be more than adequate for the initial test,
development and operation of the submarine. It may be
found that a sensitivity of 5 times that of the depth
control is perfectly acceptable under all circumstances.

<table>
<thead>
<tr>
<th>Signal Range</th>
<th>Accuracy</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading</td>
<td>0° + 360°</td>
<td>0.1</td>
</tr>
<tr>
<td>Depth</td>
<td>0 → 100m</td>
<td>0.01m</td>
</tr>
<tr>
<td>Roll</td>
<td>-30° → +30°</td>
<td>0.3°</td>
</tr>
<tr>
<td>Pitch</td>
<td>-30° → +30°</td>
<td>0.8°</td>
</tr>
<tr>
<td>Fin Position</td>
<td>-30° → +30°</td>
<td>0.3°</td>
</tr>
</tbody>
</table>

TABLE 1-2 Sensor Resolution Requirements

1.3.3 Power Supply

Power usage must be kept to a minimum in every
aspect of the electronic system development. The will
require the use of CMOS devices when feasible and
low power Schottky devices otherwise. A basic breakdown
of the power usage is presented in Table 1-3. These are goals for individual systems and may be traded off if one area can be improved by using power available from another. The ultimate success of the system will be determined when it becomes operational and tested against its 2 hour mission requirement.
<table>
<thead>
<tr>
<th></th>
<th>Watts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A. Computer System</strong></td>
<td></td>
</tr>
<tr>
<td>CPU/Memory</td>
<td>1.50</td>
</tr>
<tr>
<td>Multiplier</td>
<td>0.04</td>
</tr>
<tr>
<td>ADC</td>
<td>0.04</td>
</tr>
<tr>
<td>DAC</td>
<td>0.02</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>0.40</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>2.00</td>
</tr>
<tr>
<td><strong>B. Sensors and Sonars</strong></td>
<td></td>
</tr>
<tr>
<td>Compass</td>
<td>1.00</td>
</tr>
<tr>
<td>Pitch and Roll Sensors</td>
<td>0.10</td>
</tr>
<tr>
<td>Depth Sensor</td>
<td>0.50</td>
</tr>
<tr>
<td>Collision Avoidance</td>
<td>0.50</td>
</tr>
<tr>
<td>Bottom Finder</td>
<td>0.50</td>
</tr>
<tr>
<td>Pinger</td>
<td>0.50</td>
</tr>
<tr>
<td>Side Scan</td>
<td>8.50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>11.60</td>
</tr>
<tr>
<td><strong>C. Power Supply (65% Efficiency)</strong></td>
<td>5.40</td>
</tr>
<tr>
<td><strong>D. Servo and Propulsion Motors</strong></td>
<td>41.00</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>60.00</td>
</tr>
</tbody>
</table>

**TABLE 1-3 Estimated Power Allocations**
CHAPTER 2
ELECTRONIC SYSTEM DESIGN

The development of any microcomputer controlled system involves hardware and software. These two aspects of the system are intimately related. Whatever is not accomplished with hardware, must be implemented with software. The next chapter will describe the major subroutines required to control the submarine. This chapter describes the hardware designed for the same purpose.

2.1 SYSTEM DESCRIPTION

A modular design approach to the electronic system development is used for two reasons. First is the requirement to break up the project into manageable sections which can be assigned to different individuals. The enormous amount of work involved in the total system construction demands a team of personnel. Although close coordination must be maintained, each individual can work mostly on his own in the initial stages of development.

As the submarine proceeds in its development, modifications and improvements will be incorporated. A
change in any system can be implemented by a change in its particular module. The impact, if any, on the other modules should be minimal. In this manner, changes in RAM and ROM, an upgrade to 16 bit resolution in analog signals, or the addition of a new sonar can be easily made.

2.1.1 System Configuration

The interdependencies of the various modules can be seen in Figure 2-1. The heart, and brain, of the electronics system is the central processing unit and its three appendages, ROM, RAM and multiply unit. Surrounding this central core is an interfacing layer consisting of the analog, digital, and serial interfaces, interrupt processor, and power supply regulation circuitry. All other systems are in the outer layer and must enter the core through one of these interfacing modules. Each of the areas of the core and interface layer are discussed in Section 2.2 as part of the microcomputer system. The remaining modules are addressed individually.

2.1.2 Chassis

In the first generation robot submarine, the electronics system was scattered amongst several
FIGURE 2-1 Electronics System Configuration
separate tubes. This required several cables to interconnect the various system. With several cables, the probability of an improper connection is increased. In the end an improper connection led to a disastrous result.

Within the tubes themselves, the electronics were packaged in the manner most suited to that sub-system. This led to numerous different configurations. Most boards were soldered, including the interconnecting wires in many cases. As a result the system was very inflexible, difficult to access, and impossible to disassemble.

Major improvements in the electronics system of the second generation submarine will be achieved with the use of a chassis for mounting the various electronics modules. In and on the chassis will be the entire electronics package. Only the motors, sonar transducers, batteries, pressure sensor, and serial connector will be elsewhere and only due to absolute necessity. Connections to the chassis will be made through a single connector at each end. In this manner, the entire package can be removed from the submarine by unplugging the two connectors and sliding it out on its rails.
Rather then attempt to build a chassis, a commercially available card rack will be used. The card rack is made by AMP International Corporation and has several important features. First is its sturdy construction. The cards will be very firmly anchored in the card slots and guide rails. The sides and end plates are solid enough to easily support the devices to be mounted on the chassis. The chassis's all aluminum construction is extremely important in minimizing the magnitude fields in the submarine. Any magnetic field will have a direct effect on the compass.

The motherboard of the card rack, containing the card edge connectors, has a full ground plane on one side and a full +5V plane on the other. This is very important in eliminating noise, particularly for operations switched at 4 MHz.

Finally, the outside of the motherboard has wire wrap pins for the 72 pin bus system. The bus system is much easier to follow, document, and modify than the disorganized point to point connection system used in the first generation submarine. Four pins on each card edge connector have been presoldered to the ground plane and four have been presoldered
to the +5V plane. If a module needs to be worked on or replaced, it can be simply pulled from its slot. Connections to circuit boards will only be allowed through the bus.

Several devices will be mounted on the chassis itself. These will include the compass, pitch and roll potentiometers, servomotor amplifiers, and the +5V DC/DC converter. All connections from these devices will be brought to pins on the bus at the bottom of the chassis. As a result, the electronics supporting a device may be removed without affecting the connections.

The chassis supports thirteen circuit boards. Care must be taken in the assignment of card positions. The power supply will be in the sternmost position to remove it as far as possible from the compass which will be mounted on the forward plate of the chassis. The CPU/memory board will be located in the center to minimize the distance traveled by signals on the address, data, and control buses. Analog processing circuitry will be located in the after portion to be away from the compass and close to the cable connector servicing it. Similarly, sonar circuits will be forward where the connectors bring in the transducer lines.
2.1.3 The Bus

Each card edge connector projects its 72 pin through the bottom of the motherboard. These pins will be wired together to form the bus system. To minimize confusion by maximizing uniformity, most of the pins will be assigned a specific signal or reserved for a specific type of signal. Because eight pins are presoldered to ground and +5V, only 64 pins are available.

The 32 pins on one side of the bus will be dedicated to the 16 address lines, 8 data lines, and 8 control signals of the central processing unit. Many of the pins on the other side of the bus will be reserved for local control signals and lines carrying analog signals.

Because one side of the bus will be different for each circuit board, steps must be taken to ensure that each board finds its proper slot. Card slots are provided with blocking wedges that may be placed between any two pins. The card edges will be notched and the wedges placed such that cards may only be inserted in their respective slots and only in the proper direction.
2.2. MICROCOMPUTER SYSTEM

The microcomputer system of the MIT second generation submarine will perform two essential functions, autopilot and supervisor. The digital autopilot will maintain a stable course, speed and depth. The supervisor will store mission tasks and execute them at their designated times. Additionally, the supervisor will process interrupts, monitor the status of various systems, and conduct test and evaluation routines.

The core of the microcomputer system will be the central processing unit. Numerous systems must be attached to the CPU to make it functional. Several types of memory must be available to store programs and data. If the CPU is to communicate with its surroundings, interfacing must be provided. There are certain functions, such as the multiply operation, that must be accomplished with hardware instead of software. Each of these areas are discussed in the following sections.

2.2.1 Central Processing Unit

The Zilog Z80 microprocessor chip was chosen for the central processing unit of the second generation robot submarine. The Z80 is a 40 pin TTL device normally
operating at either 2 or 4 MHz. It requires a +5V supply, as do all TTL devices. It has an 8 bit data bus and 16 bit address bus. Both maskable and non-maskable interrupt functions and 256 input/output ports are supported. This popular device is inexpensive ($10.95) and readily available. There are several reasons for its choice.

The primary reason for the selection of the Z80 microprocessor chip is the ease in software development. The Z80 instruction set includes more than 600 instructions. More importantly, the Z80 is the microprocessor used in several commercially available microcomputer systems. Radio Shack's TRS-80, Heathkit's H-89, and the North Star Horizon are all Z80 based systems. Each has available editor/assembler and debugging/monitoring programs which can be used to develop Z80 machine code software. A single program combining the editor/assembler and debugging functions is available from Microsoft for the TRS-80. More sophisticated languages such as FORTH or STOIC, can be used with the North Star microcomputer.

Another benefit of the Z80 is its refresh function. The use of 16K x 1 bit dynamic RAM chips provides a great deal of memory in a small space.
Dynamic memory chips, however, require refreshing. The refresh function of the Z80 greatly simplifies the circuitry required to interface the dynamic memory. Care must be exercised in the use of wait states in the microprocessor's operation. This is the only time refresh operations are suspended.

The major disadvantage of the Z80 microprocessor chip is its higher power consumption when compared with CMOS types. The Z80 itself will require 0.3 to 0.4 more watts than the Intersil 6100 selected in the original second generation design. This represents only 0.5% of the 60 watt power supply.

The Z80 microprocessor will be run at its maximum speed, 4MHz. Under these circumstances, a read pulse will last approximately 200 ns. This will require high speed switching devices on the data bus for operations inputing to the central processing unit. The actual microprocessor chip selected has little bearing on these demands. Use of a CMOS microprocessor at MHz would not use significantly less power in its interface.

With few exceptions, low power Schottky devices can be used to minimize the load. In many areas CMOS
systems can be used. These systems may or may not require buffering. National Semiconductor's 74CXX series devices can be used when slow speed is acceptable. Direct substitution of CMOS devices for TTL devices can be attempted in many areas once construction, debug, and testing is completed. Pin-for-pin equivalents of most chips are available.

Finally, within a year, a CMOS version of the Z80 chip will be available. If it is deemed necessary, the CPU/memory board can be replaced with the CMOS Z80 substituted for the TTL Z80.

2.2.2 Memory

The 64K memory directly addressable by the 15 bit address bus is divided into four 16K areas by decoding the two most significant address bits, A14 and A15. One of these areas is devoted to ROM, two to RAM, and one to memory mapped functions (refer to Figure 2-2).

ROM

The lowest level (0000H-3FFFFH) is devoted to read only memory. Initially this will consist of a single 16,384 bit chip. The chip will be a Texas Instrument TMS2516. The 16K bits are addressable as
2048 eight bit words. This new generation of ROM has the distinct advantage of requiring only a +5V supply. Most other chips would require -5V and +12V in addition to the +5V supply.

The ROM will be used to bootstrap the micro-computer system into an operational state. This will include initializing the system and loading memory from the support system.

At some point in the development of the second generation submarine, this ROM can be expanded to occupy more or all of the first 16K of memory. Once firmly established, all of the major autopilot and supervisor subroutines can be located in ROM. This will leave more RAM available for data collection and analysis. Expansion will require decoding of the A11, A12, and A13 addresses which can be accomplished by a single 3 to 8 line decoder.

**RAM**

The second level (4000H-7FFFH) is occupied by the submarine's 16K random access memory. This will be provided by eight 16,384 by 1 bit dynamic RAM chips. The MOSTEK M5K 4116P-2 will be used. The 4116 is
\begin{figure}
\centering
\begin{verbatim}
0000H  2K ROM
8000H  Reserved for additional 14K ROM
4000H  16K RAM
8000H  Reserved for additional
C000H  Reserved
C800H  for
D000H  six
D800H  memory
E000H  mapped
E800H  functions
F000H  Multiply unit
F800H  Digital Interface
\end{verbatim}
\caption{Microcomputer Memory Map}
\end{figure}
organized as a 128 x 128 bit matrix which is addressed by a seven bit row address signal (RAS) and a seven bit column address signal (CAS). This requires additional multiplexing (four or five extra chips) but greatly reduces the size of the memory chips. The 16 pin package of the 4116 requires much less space than a 24 pin package with its 14 address pins.

The 4116 chip is very fast (less than 150 ns access time) and it uses very little power on a per bit basis (approximately 28 μW/bit). They require three power supplies, +5V -5V and +12V. Refresh of the dynamic cell matrix requires a memory operation at each of the 128 row-address locations within a 2 ms time interval.

The third level (8000H-BFFFFH) is reserved for an additional 16K RAM. Transferring the majority of routine functions to ROM and expanding the RAM to 32K will give the second generation robot submarine a significant on-board data collection and processing capability. Eventually, all functions associated with the operation of the submarine should be located in ROM and the RAM can be utilized solely for mission oriented data collection functions.
Memory Mapped Functions

There are three ways to communicate with the central processing unit; using either the maskable or non-maskable interrupt, using an input/output port, or memory mapping. Memory mapping has two advantages over the use of I/O ports. First, memory locations can be addressed more quickly and easily than an I/O port. Only IN and OUT instructions apply directly to I/O ports. Nearly all manipulations can be performed directly on a memory location if its address is specified in the HL, IX, or IY registers. Many functions can be applied directly, including 16 bit load operations.

Register files or 16 to 256 bit memory chips can be used to store information the central processing unit will require. This interface method makes operations transparent to the CPU. In this manner, for example, analog-to-digital conversion of sensor signals can be accomplished without direct control from the CPU. The efficiency of the CPU can be greatly increased if it is not required to perform such functions.

The highest level of memory (C000H-FFFFH) is divided into eight sections (refer to Figure 2-2).
Each operation requiring a memory mapped section will use a single 74LS85 chip (or its CMOS equivalent) to decode address lines A1 through A15. This will provide an enable signal for process execution, or register or memory read/write. All connections to the data bus must be tri-state buffered to prevent interference between memory mapped functions and other operations.

2.2.3 Multiply Unit

The computational capabilities of the microcomputer system is established in Reference 1. When the autopilot is operating at a level requiring the maximum number of multiplications, each multiplication may take no more then 96 μsec. A multiply operation may be implemented in several wyas.

A software subroutine can be used to perform the multiply operation. Although many subroutines performing 8 bit by 8 bit multiplications are readily available, none can meet the time constraint. The average 280 instruction takes 3 μsec when operating at MHz. Any multiply subroutine would then be limited to approximately 32 lines.
An alternative to the simple subroutine involves the use of look-up tables. The entire operation could be accomplished with a look-up table or in combination with a much abbreviated subroutine. The look-up table required for this method would occupy approximately 4K bytes of RAM. There is no advantage to this method justifying the use of one-quarter of the available memory.

The fastest method of multiplication is accomplished with hardware. Sixteen Motorola MC14554 2x2 multiply units can be cascaded to perform 8x8 multiply operations in less than 10 \( \mu \)sec. A memory mapped interface requires seven chips. The multiply unit will use very little power because of its low duty cycle and very low power CMOS multiply units.

The software required to implement the multiply unit is remarkably small. The unit must be loaded with an 8 bit \( X \) multiplicand and an 8 bit \( Y \) multiplicand. A delay must be written into the subroutine to give the unit enough time to complete the multiplication. Finally, the 16 bit product, \( S \), must be read. With the interface circuitry and addressing configured to use the Z80's 16 bit load instructions, the subroutine required is limited to five lines; one to load both \( X \) and \( Y \), two for delay, one to read \( S \), and a return.
2.2.4 **Interrupt Processor**

Certain functions supported by the submarine's microcomputer do not require constant attention. These functions may be ignored most of the time. Both the communications and collision avoidance sensors operate only if triggered by an event such as a command transmission or the appearance of an obstacle. The digital autopilot must update data every 50 msec to calculate derivatives. Rather then perform the tedious, and very complex, process of counting instruction cycles, a clock can be hard wired. These, and other functions will use the maskable interrupt input to the CPU.

Several devices, including the water sensor, low power sensor and communications systems, look for emergency conditions requiring an abort of the mission. In this case the signals must be given the highest priority possible, the non-maskable interrupt.

Design of the interrupt processor must include three major factors. First it must be able to identify which device is doing the interrupting. For a maskable interrupt this is simply a matter of strobing an identification code onto the data bus during the special read cycle of the Mode 2 interrupt. Strobing
is implemented with the interrupt acknowledge signal which occurs only then. In the non-maskable interrupt, no special cycle signal is produced. After measures are taken to effect the abort, a read instruction can be executed to strobe a special latch containing the identification of the device demanding the abort.

In maskable interrupt operations, the interrupt processor must be able to handle multiple or simultaneous interrupts. This must be accomplished without missing or repeating any interruption. In addition, the system must prioritize the interruptions. This may be accomplished with the use of hardware or software.

Finally, the system must systematically reset itself to allow the supported functions to interrupt more than once. Resetting must be only for serviced interrupts to comply with the requirements in the previous paragraph.

The circuitry required to accomplish these tasks is described in Appendix III. This circuitry must, of course, be supported with software.
FIGURE 2-3  Interrupt Processing Block Diagram
2.2.5 **Digital Interface**

The digital interface has one basic purpose, data bus controller. More than a dozen systems will be attached to the data bus. Each will have a latch, tri-state buffer, or memory chip to prevent interference with the normal operation of the bus. The digital interface will decode addresses assigned to memory mapped functions and provide enable signals to transfer their information to the data bus. When the CPU looks for a specific piece of information, the digital interface will find it.

Several other incidental functions will be performed by the digital interface. Control pulses of various types can be provided by the digital interface. These may be used to reset latches, gate other analog or digital control signals, or on/off switching signals for device controllers. Expansion of the digital interface circuitry will provide support for the I/O port system. The 256 ports will require decoding just as memory mapped functions do.
2.2.6 Analog Interface

Information required by the digital autopilot will be generated as an analog signal for the pitch, roll, and depth sensors, compass, and fin position indicators. These signals must undergo analog-to-digital conversion. Similarly, for the CPU to transmit the fin positions generated by the digital autopilot, digital-to-analog conversion will be required. Both digital-to-analog and analog-to-digital systems should be flexible enough to allow expansion as the submarine develops.

FIGURE 2-4 Analog Interface Block Diagram
The ADC and DAC will each have three major sections in addition to the convertor itself. As can be seen in Figure 2-2, each will require a memory mapped storage area, timing and control circuitry, and an analog multiplexor. (Figure 2-2 applies to both the ADC and DAC with the information flowing from lift to right for digital-to-analog operations and right to left for analog-to-digital operations.)

The presence of the analog multiplexor allows a single convertor to process several signals. Each system will be provided with an AD7506 sixteen channel analog multiplexor. This will allow for easy expansion of the interface to include additional inputs or outputs. Initially the ADC will use only nine channels (four fin positions, pitch, roll and depth sensors and two compass inputs). The DAC will use only four for fin position commands.

The use of 74S89 sixty-four bit static RAM chips will make analog conversion operations transparent to the CPU. Care must be taken to ensure that these chips are not accessed by both the CPU and analog conversion system at the same time. The ADC will update the output files far more often then the CPU will access them for information. The DAC will monitor its files often enough to ensure the prompt response of the systems under
its control. No significant lag in the transfer of commands from the CPU to systems under its control will be presented by the use of these files. The time required to convert the signals from digital to analog and the time required to sample them for the sample and hold amplifiers far exceeds the file's access time.

Timing and control circuitry will automatically step through each of the sixteen addresses. As it does, it must generate the proper signals to control the filing, conversion, and multiplexing operations.

Analog-to-digital conversion will be provided by a National Semiconductor ADC 1210 twelve bit CMOS ADC. This successive approximation device is low power and medium speed. Conversions are typically completed in 100μs when driven at a 130 kHz clock frequency. Use of the ADC 1210 will require a +10V precision voltage reference source and external comparator. Operation of the converter will be bipolar. The output of this CMOS device will require buffering to drive the TTL memory chips.

Digital-to-analog conversion will be provided by an Analog Device AD 7533 ten bit multiplying DAC. This device will also use an external comparator and bipolar operation. The AD 7533 uses a -10V precision voltage
reference source. The analog output signal will be multiplexed to one of several Intersil IN5113 IDE sample and hold amplifiers.

The initial designs of the ADC or DAC can be greatly simplified by allowing some degree of CPU control of the conversion operation. Sacrificing the transparent design will not degrade system performance in the initial stages of development when the CPU is being lightly loaded. As development proceeds, more demands will be made on the CPU. The unit may then be redesigned to transparent operation. Eventually, the system should be upgraded to 16 bit accuracy.

2.2.7 Serial Interface

The sole purpose of the serial interface will be to communicate directly with the submarine's computer from the field support system. The serial interface will provide the means to initiate the bootstrap program residing in ROM. Once activated, the serial interface can be used to conduct or monitor the tests of the various systems including RAM, multiply unit, servomotor and sensor systems, convertors and sonars. Upon satisfactory completion of these tests, the serial interface will be used to load the RAM with the programs, subroutines, and
data required for the mission. Finally, the serial interface will be used to retrieve data generated during the mission and stored in RAM.

The core of the serial interface is the IM6402 universal asynchronous receiver transmitter (UART). This versatile CMOS device is a low power (less than 10 mW) programmable subsystem for interfacing microprocessors. The 40 pin chip converts 5 to 8 bit parallel data into serial form and automatically adds start, parity, and stop bits. It also performs the reverse operation when acting as a serial signal receiver. An HDI-6405A CMOS programmable bit rate generator will provide the necessary clock signals to drive the UART.

Connection of the serial interface will be accomplished initially with a two wire connection. The wires will penetrate the pressure hull through the same connector used by the sonar transducers. Eventually, an optical system will be developed which will greatly increase the reliability of the system. It will also allow access in the ocean itself.

2.3 SENSORS

The sensor package provides the critical information required for the proper navigation of the robot submarine. The location, heading, and attitude of the submarine must
be continuously monitored. These are the inputs to the digital autopilot.

2.3.1 Pitch and Roll Sensors

The pitch and roll sensors determine the attitude of the submarine. They are identical systems except for their orientation with respect to the submarine's axis. Commercially available precision pendulum potentiometers were selected. Humphrey CP17 series potentiometers were used for their low power (less than 0.5 watt), high accuracy (± 1.0% static error), and natural frequency (3.2 Hz).

The potentiometers will be mounted on the chassis with the roll sensor on the stern plate and the pitch sensor on the starboard side. Connections will be brought through the bus to the pitch, roll, and depth sensor module. They will be fixed in position requiring external electrical zeroing or microprocessor compensation.

2.3.2 Depth Sensor

Reference 1 specifies a precision strain gauge pressure transducer for a depth sensor. This type of device will meet the 100 meter depth requirement while maintaining an acceptable level of error and nonlinearity.
Initial development of the second generation robot submarine will, however, utilize the same system as the first submarine. Time and manning constraints dictate a compromise. National Semiconductor's LX1710AF pressure transducer will provide 0.5 meter accuracy over a 60 meter depth range. This should be more than adequate for the initial development stage.

2.3.3 Compass

The compass for the second generation robot submarine will be an improved version of the Hall-Effect solid state compass. Two Hall-Effect crystals will be mounted perpendicularly. Each will provide a Hall voltage proportional to the relative angle with respect to magnetic north. The ratio of these components represents the tangent or cotangent of the heading. Software required to convert the tangent to an angle is not necessary if headings are specified as tangents.

The compass will be improved in several ways. First will be the method of mounting the crystals. Each will be encased in plexiglass with mu-metal field concentrators for increased sensitivity. One end will be anchored in RTV silicon rubber. The other end will be held by four
set screws. In this manner, each crystal can be adjusted separately providing mutual orthogonality with respect to the vertical component of the earth's magnetic field. The plexiglass mountings also provide a means to rigidly attach the compass to the forward chassis plate. The mounting system must be entirely non-magnetic.

Secondly, the circuit design of the new compass should represent an improvement over the old design. The circuit will consist of a current source, an amplifier, and a noise filter. The current source will be provided by an LM105 regulator used as a +2.5V switching power supply. Very low offset voltage instrumentation amplifiers will amplify the Hall voltages from approximately ± 2 millivolts to ± 5 volts. Finally, a buffered 15 hertz filter will pass the compass signals to the analog multiplexer in the analog-to-digital converter circuitry.

Lastly, a software subroutine will be utilized to decouple the pitch and roll angle. Through the use of the first submarine's compass, it was realized that the compass would be twice as sensitive to pitch and roll angles as it would be to its angle with respect to the earth's magnetic field in the New England area. Simple
algorithms for this purpose have been developed by Saylor in Reference 1 and Davis in Reference 4.

The impact of these improvements is felt in both complexity and accuracy. The circuitry will be much simpler than the original and at the same time provide much greater accuracy. Reliable readings within 0.25° should be provided by the compass while using less than a single watt of power. Greater details in design and development of the new compass is provided by Reference 4.

The use of a compass for navigation requires the use of dead reckoning. Thus, the submarine's motions will be uncompensated for set and drift. In most areas of operation when the mission length is limited to two hours or less, the effect should be minimal. In cases where the effect is significant, external tracking of the submarine can be used to determine the required corrections. The corrections can be transmitted to the submarine by the communications sonar without interfering with the mission in progress.

2.3.4 Bottom Finding Sonar

The design of sonar systems has been greatly simplified by National Semiconductor's LM 1812 ultrasonic transceiver. In a single 18 pin chip, this device performs
the functions of modulation, power amplification, receiver preamplification, and signal amplification. It also greatly simplifies the external circuitry required to set the frequency and gain, filter the noise, and provide fine tuning. The LM 1812 and its required circuitry is the core of the bottom finding sonar.

On the ocean side of the LM 1812 is the transducer. Design considerations (see Reference 5) led to the use of a baffled, lead zirconate-lead titanate transducer. The transducer produces a cone with a $22^\circ$ beam width. It is designed to be a resonant transducer to conserve power. It operates at 215 kHz which is very near the optimal design frequency of the LM 1812 (200 kHz).

The other side of the LM 1812 (see Figure 2-5) includes the circuitry required to trigger the sonar (the \( T_x \) pulse) and provide information usable by the submarine's computer. A clock will drive a counter up to a preset interval during which time the sonar listens for an echo pulse. If the pulse, \( R_c \), is received the count is latched into a register file. If not the next transmission will latch in zero, indicating the bottom is out of range.

The proper choice of clock frequency will result in the 8 bits provided being a binary distance to the bottom. At 2.3 kHz the readings will be feet. At 750 Hz it will
meters. Details of the bottom finding sonar are included in Reference 5 as well as recommendations to improve the system.

FIGURE 2-5 Bottom Finding Sonar Block Diagram
2.3.5 **Collision Avoidance Sonar**

The first collision avoidance sonar for the submarine is a fixed beam sonar nearly identical to the bottom finding sonar. It will have a conical beam pattern, with a $22^\circ$ arch scanning directly ahead of the submarine. Its frequency will also be near the optimal frequency of the LM 1812 but different from that of the bottom finding sonar to prevent interference. This sonar will be simple, take up little space, and require little power.

The second generation robot submarine should eventually include an improved collision avoidance sonar. The improved version will be steerable. This capability will allow the submarine's computer to navigate around or through obstacles instead of having to abort the mission. It will also allow the submarine to look in the direction of a turn. Currently, the submarine has no protection from collisions with objects not directly ahead. This is critical when the submarine is maneuvering. The improved sonar will, however, be more complex. In later stages of development, the competition for the limited space and power of the submarine will increase. The priority of the improved version must then be established.
2.4 Communications Sonar

A major improvement of the second generation submarine over the first generation submarine is the use of a communications sonar. This sonar will provide a limited, but extremely important, interactive control capability. The operator of the submarine will be able to abort or alter the mission, correct for set and drift, or maneuver the submarine for recovery.

A frequency of 40 to 50 kHz is optimal for a range of approximately one kilometer. This frequency, however, does not make effective use of the LM 1812. For this reason, the functions performed by the LM 1812 will be instead performed by an LM 270 automatic gain control/squelch amplifier and an LM 567 tone decoder. The receiver design of Reference 5 will be further modified to transmit back the code it has received. This will be advantageous in several ways. It ensures the operator that his command has been received correctly and if not, allows him to take immediate action to correct an incorrect reception. The probability of false acceptance of the system is relatively high, approximately $3 \times 10^{-4} \text{ sec}^{-1}$. Monitoring the field support unit will detect any such difficulty. Finally, the system will
allow a limited amount of information to be obtained from the submarine's computer during the mission.

Transmission rate for the sonar is approximately one byte per second. This means a multiple byte command will take three seconds to transmit. This is a great deal of time in terms of the submarine's computer and the operations it is conducting. It is 30 cycles of the digital autopilot for example. In terms of the operator, however, it is not much slower than the time required to throw the switches and certainly less than the time required to make the decisions involved.

A redesign of the communications sonar system should eventually be considered to improve its speed and reduce the probability of false acceptance.

2.5 **FIN SERVOMOTOR SYSTEM**

A description of the servomotor system and its performance requirements is presented in Reference 1. Development is centered around the Intersil ICH 8520 power amplifier motor and actuator driver. The basic circuit design described in Figure 2-6 will undergo minor modifications during development to obtain the desired response. Final evaluation of the system's time constant,
deadband, and linearity will be made only after extensive testing, including open water tests.

The TRW Globe type SD planetary gearmotors are rated at 6 volts DC. They will be driven at a maximum of 12 volts resulting in a somewhat shorter life. The effect will be minimized by the low duty cycle and the infrequent application of the full 12 volts. Feedback is provided by a Clarostat 25 kΩ ten turn potentiometer whose shaft has been replaced by the servomotor drive shaft. A lower resistance, three turn potentiometer was preferred but, unfortunately, was not available.

FIGURE 2-6 Fin Servomotor System Diagram
The output of the DAC will be retained by Intersil IH5113 general purpose sample and hold amplifiers. This input must be reduced by a factor of ten to make the full output range of the DAC useable. Desired deflections are $\pm 30^\circ$ and the shaft to fin ratio is 4:1.

Improvements to the fin servomotor system might eventually be accomplished by a second order feedback system. The potentiometer used to determine the fin position might be replaced with an optical system for much greater accuracy.

2.6 **PROPULSION MOTOR**

The propulsion motor is a TRW Globe type BL planetary gearmotor rated at 27 volts. The motor will be run at 24 volts by connecting it across the +12V and -12V sides of the batteries. Figure 2-7 describes the relay switching required to reverse the polarity of the connections for forward and reverse operation. The relays require positive pulses in the order of milliseconds. The feedback signal can be fed into a four bit magnitude comparator to reset the latch controlling the command signal. This will allow the submarine's computer to strobe the command to the motor.
FIGURE 2-7 Propulsion Motor Switching Block Diagram
2.7 BATTERIES

Both Reference 1 and 2 deal extensively with the subject of batteries. The battery selected for the normal (2 hour mission) operation is an Eagle-Picher type CF12V8 lead-calcium battery. These long life, rechargeable batteries are maintenance free. They are similar to a gelled battery in that the electrolyte is immobilized by an absorbent separator allowing use in any position. The batteries are constructed with a dual seal protecting against leakage under normal circumstances. There is some chance of hydrogen emission during operation which will have to be considered when disassembling the battery compartment. Charging of the battery must be done in a ventilated space with the batteries upright.

No final decision has been made on the battery to use in extended (15 hour mission) operations. This is better left to latter stages of the submarine's development. Extended operations are still a long way off and will require the most up-to-date technology to effectively implement.
2.8 POWER SUPPLIES AND REGULATION

The various systems of the electronics package will require at least eight and possibly ten different voltages. Figure 2-8 describes the regulation and distribution system. All supplies will require some bypassing and most require some form of regulation. The batteries are expected to be at 13.5 volts with no load. Once loaded the voltage should drop to 12 volts and hold there for the length of a mission. It may be found that the voltage will dip below 12 volts. If it does, only the 4116 memory chips will be significantly effected. The chips must be tested to determine the lowest voltage allowable for reliable operations. Impact of an 11 volt input to the +5 and +15 volt DC/DC convertors should also be determined.

Power for the propulsion motor will be provided on separate lines from those for the fin servomotors. This is to minimize the noise in the fin servomotor systems. Both will be unregulated supplies with bypass capacitors. The relatively constant impedance of the battery should minimize noise spikes. Additionally, steps will be taken in the design of the servomotor systems to prevent their operation from causing fluctuations in the supply due to variable loading.
FIGURE 2-8 Power Regulation and Distribution System
The various analog devices (operational amplifiers, analog multiplexors, ADC, DAC, etc.) will require ±12 volt supplies. Regulation for these supplies should not significantly drop the voltage. A simple RLC circuit, see Figure 2-9, should be sufficient. In the case of the ADC, DAC, and certain other devices, accuracy may be degraded unless ±15 volts is used instead of ±12 volts. Both supplies should be attempted and ±12 volts used if acceptable.

![RLC Circuit Diagram](image)

FIGURE 2-9 Simple RLC Regulation Circuit

Where ±15 volts is required, it will be supplied by a DC/DC converter made by Semiconductor Circuits. ±12 volts will be fed into the converter and ±15 volts ±1% at better than 60% efficiency will be gotten out. Ripple and noise will be less than 5mV peak-to-peak. The converter will be able to provide up to 100 mA.
The analog-to-digital converter requires a +10V precision voltage reference. The +15V output of the DC/DC converter will be used to drive a PMI Ref-01HJ precision voltage reference. Power consumption of this device is low (15mW typically) and will regulate the voltage within 0.1% while providing 20mA.

The digital-to-analog converter requires a -10V precision voltage reference for which an AD2701 will be used. This device is somewhat more accurate at ±0.05% but will supply only 10mA.

All of the digital logic, both TTL and CMOS, will use +5V. The total requirement for the entire electronics package will be approximately 3 amps. Once again, a Semiconductor Circuits DC/DC converter will be used. The tolerance is ±2% and regulation is ±0.2%. The device can supply up to 5 amps. It is easy to foresee the requirements being less than 2.5 amps. The device's efficiency (73%), however, is relatively independent of load.

The Mostek 4116 dynamic RAM chips require +12 volts and -5 volts in addition to +5 volts. These supplies must be well regulated and noise free. Because the current requirements are low (less than 100mA for +12V and 5mA for -5V) zener diodes can be used to efficiently control
the voltage. The inputs will be +12 volts from the battery. It is extremely important, therefore, to provide a regulated +12V without any significant voltage drop.

Two other devices require special voltages. The submarine's compass will use switching regulation of either +5V or +12V to provide a 2.5V supply.

Finally, the side scan-sonar requires a 750V supply. A flyback blocking oscillator converter is designed for this purpose (see Reference 6).

2.9 BALLASTING SYSTEM CONTROL

Initially the ballast and trim system for the submarine will be the simplest possible. Ballasting will be done manually until the submarine is level and has a slightly positive buoyancy. A pressure regulator valve will maintain approximately 6 psi overpressure in the ballast tanks. When the mission is complete or abort conditions are sensed, a solenoid valve to the sea will be opened. Water will be forced out of the ballast tanks by the overpressure and the submarine will surface. This system requires a single control. The submarine computer must be able to open the solenoid valve.
Eventually, a much more sophisticated ballasting and trim system will be built into the submarine. Reference 7 discusses the present system as well as several future systems. A system controlled by the submarine computer would use the pitch and roll sensors to adjust trim and the depth sensor to adjust buoyancy. Computer controls would be expanded to include the valving necessary to perform the above tasks. A method to transfer water between the forward and after ballast tanks would add to the control requirements but greatly simplify the process by decoupling the ballast and trim functions.

2.10 **SIDE-SCAN SONAR**

The side-scan sonar will be the major payload of the second generation robot submarine. A great deal of effort will be necessary to fully develop its potential. Reference 6 describes the work previously done on the system. Nothing has been done since then because the limited manpower has been concentrated on the development of the submarine itself.

The portion of the system carried by the submarine will include the transducers, sonar amplifier, tape recorder, and tape interface. Tapes generated during the
mission will be removed from the submarine and played back on a tape recorder driving a chart paper recorder. Eventually, the data will be fed to the laboratory support computer for analysis.

Submarine Package

![Diagram of submarine package](image)

Support Package

![Diagram of support package](image)

FIGURE 2-10 Side-Scan Sonar System Block Diagram

The sonar transducers are available and a 750V supply has been designed. A great deal of development, however, is necessary on the remainder of the system. Due to space and power limitations, a subminiature recorder should be used. The time expander developed in Reference 6 is excessively large due to the circuitry required to make the signals
compatible with the Alden wet paper recorder. A new recorder should be used and any interfacing required should be included in the shore package, even at the expense of additional complexity. The timing and control circuitry is done but the sonar amplifier has not.

2.11 EMERGENCY SAFEGUARDS

There are several conditions under which the submarine will abort the mission, blow ballast, and come to the surface. Two conditions require no additional circuitry. First is the abort command received by the communications sonar. This command will be hard wired to the non-markable interrupt. The other condition is excessive depth. Each time the submarine computer reads the depth, at least every 0.1 second, it will check it for excessive depth and jump to the abort routine if necessary.

As in the first generation robot submarine, the conductivity of salt water will be used to sense water in the compartments. A water sense line connected to a positive voltage will run parallel to a ground line around the tube. When seawater bridges the wires, the water sense line will drop to zero and trip a flip-flap whose output is connected to the interrupt system. The
battery, electrical, ballast, and motor compartments should all include water sensors. The loss of any one of these would be catastrophic.

Loss of power is another major concern. Two levels of failure must be considered. Even a marginal loss of power dropping the supply voltage below some threshold could significantly effect the computer's memory. Catastrophic failure of the power supply must also be considered. Two relays would be set to sense each cause, one at approximately 11 volts and the other at approximately 9 volts. The first would allow the computer to take action to bring all systems to a halt and protect its memory while blowing ballast to surface. The other system would be a backup to physically protect the submarine in the event of a computer failure.

The final emergency safeguard is the submarine's pinger. This will allow the operators to track the submarine and, if all else fails, locate it in an emergency. The same pinger will be used for the second generation submarine as was used for the first.
CHAPTER 3
SUBMARINE SUPPORT SYSTEM

The microcomputer system of the second generation robot submarine cannot operate without the support systems. Through the submarine serial interface and connector, a link will be made with the field support unit. The field support unit can be buffered to communicate directly with the laboratory computer. The software development system will produce tapes used by the field support unit to program, monitor, and test the submarine systems. The entire system, described in Figure 3-1, is necessary to effectively use the robot submarine.

3.1 FIELD SUPPORT SYSTEM

The various equipment carried with the operators into the mission area constitutes the field support system. At present, there are three major items designated as field requirements. The communications system requires a transmitter/receiver with display. This system has been described in Section 2.4 and in Reference 5.

The effective length of operation of the submarine batteries is limited. If the operators are not to return to the laboratory after each mission, a battery charger must be included as part of the field support system.
FIGURE 3-1 Support Systems
Additionally, a generator must be provided in the event that there is no power available in the vicinity of the mission area.

Finally, there is the field support unit. This unit provides for direct communication with the submarine computer. It is a compact, lightweight system that can be easily carried into the field. The design and functions of the field support unit are the subject of the following subsections.

3.1.1 Central Processing Unit and Memory

The design of the CPU/memory module of the field support unit will be essentially the same as that of the submarine microcomputer system. A Z80A CPU, TMS2516 PROM's, and 4116 dynamic RAM chips will be used. The field support unit, however, will operate at 2MHz instead of 4MHz and will initially have an additional 2K of programmable read-only memory. The system will be expandible to 32K RAM and 16K ROM. The slower speed of the unit is acceptable because its operation is in no way time constrained. The functions the unit performs may take any reasonable length of time. If for some reason it was felt the system should be upgraded to 4MHz, it would only require a change in crystals and a few faster logic devices.
3.1.2 Keypad and Display

A 20 key pad will allow the operator to communicate directly with the submarine computer when the field support unit is connected to it. Four keys are dedicated to single-key commands which will examine or load memory locations, examine or change the end-of-file address, and designate extended command entries. To monitor keystrokes and read information, the field support unit will have an 8 digit liquid crystal diode display.

3.1.3 Serial Port

The field support unit will include a UART and bit rate generator for the serial connection to the submarine computer. A three wire connection to the submarine serial interface will provide two-way, simultaneous serial communications with handshaking. The addition of additional UART's or bit rate generators is possible. This capability allows future expansion of the system which might include a small CRT or lightweight ribbon printer.

3.1.4 Tape System

Mass storage for the programs and data associated with both the field support computer and the submarine computer will be provided by cassette tape. (This should
not be confused with the magnetic tape used to store data from the side scan sonar.) The field support unit will have a tape system which is compatible with that of the laboratory support system. The latter will be used to develop the software required by both of the other computers. The tapes generated will be used by the field support unit or fed by it to the submarine computer. The field support unit can also empty the data from the submarine computer onto tape for later analysis by the laboratory support system.

3.1.5 **System Functions**

Table 3-1 summarizes the functions performed by the field support unit. To the list will be added other functions found to be instrumental in system development. The 16K RAM available provides enough space for expansion of the units language. Additional commands may be added to facilitate changes in missions or system editing.

3.2 **LABORATORY SUPPORT SYSTEM**

The laboratory support system will perform two major functions. During the initial development of the second generation robot submarine, the system will be used exclusively for software development. The subroutines required to run the submarine will be written and tested on
· Examine memory location
· Load memory location
· Load taped program
· Save program to cassette tape
· Verify program against cassette tape
· Send program to submarine
· Verify program against submarine
· Execute subroutines
· Perform editing functions

TABLE 3-1 Field Support Unit Functions
the laboratory system. When completed the subroutines will be assembled and loaded onto tape which will be used in the field support unit or transferred by it to the submarine computer.

In the advanced stages of development, the laboratory support system will be used in scientific analysis of data collected during missions and in the planning and analysis of the missions themselves. As the focus of the entire project shifts from the operation of the submarine systems to the implementation of the submarine as a sophisticated tool in ocean research, so will the use of the laboratory support system shift.

The requirements for the laboratory support system are numerous. The system must be Z80 based. The tape system must be reproducible for compatibility with that of the field support unit. It must be reliable, durable and serviceable. To a limited extent it should be portable. Use of the submarine for extended periods may be supported by facilities at or near the mission area. It may prove advantageous to have the bulk of the laboratory system there.

The system must be fully supported with software which includes editing, assembling, monitoring and debugging functions. A fast efficient mass storage system
should be available to effectively use the software and facilitate manipulation of large amounts of data. The cost of the system should be low.

Expansion of the system should be easy to implement with either off-the-shelf systems or systems designed and built in the laboratory. This requires complete documentation including wiring diagrams.

There are several systems readily available which meet all of the above requirements. The TRS-80, Heathkit WH-89, and Northstar computers are all Z80 based. The Heathkit was chosen because of the quality of the WH-89, the completeness of Heathkit documentation, low cost and immediate availability.

3.2.1 Heathkit WH-89 Computer

The Heathkit WH-89 computer includes two Z80 microprocessors. One microprocessor is dedicated to terminal operations. The unit includes a full keyboard and numeric pad, video display, and floppy disk system in a single, compact package. The eight bit microprocessor runs at 2MHz. Its 16 bit address bus allows the 16K RAM to be expanded to 48K. 8K is available for system ROM and RAM, and 8K is reserved.
The floppy disk system has 40 tracks which will store more than 100,000 bytes per disk. The Heathkit disk operating system, HDOS, contains all of the essential software functions. For Z80 machine language programming, HDOS provides an editor and assembler for program construction. It also contains a debug and monitoring system. The system will also support BASIC. The availability of additional software is moderate with several important packages soon to be introduced, including other higher level languages.

The CRT and keyboard are of very professional design and have been built for continuous use. Upper and lower case letters are supported. This will be important in the later stages of development when the system is used as a word processor to produce the large volume of documentation necessary. The keyboard provides numerous special functions and controls, some of which are user programmable. This feature will be useful when specialized functions become apparent in the development of the submarine's software.

3.2.2 Heathkit Peripherals

A major function of the laboratory computer is to generate the software required by the submarine. A great
deal of time and effort will be saved if these programs, developed on disk, are put onto tapes that the field support unit can use. To accomplish this a cassette interface and tape recorder have been added to the system. Heathkit's complete documentation will facilitate the design of the software drivers required to make the systems compatible.

The H-14 line printer will be used to produce the hard copy required to effectively write and debug computer programs. Documentation is an extremely important aspect of any project. It is even more important when the project requires years for completion and is done in an environment where the personnel involved are changing. The line printer can be used to produce more than just a record of the programs and data developed on it. The system can be used as a word processor to generate the documentation required to properly describe the work done by its users. Without this documentation, the work may be lost to those who follow.
3.3 SYSTEM PROGRAMMING

Hardware is only half of any computer system. Software is the other half. Before the submarine computer can perform even the simplest function, it must be programed to do so. Programs and subroutines must be assembled in Z-80 machine code, although a higher level language might be used to produce the code. The assembled programs must be loaded into RAM or "blown" into PROM.

Data analysis and applications research will also require a great deal of software. A higher level language such as BASIC or FORTRAN will be much more convenient for these type of programs. Extensive use of data files will require the volume memory and rapid access of a disk operating system.

3.3.1 Software Development System

Anyone who has attempted to hand code a machine language subroutine of any substantial length, quickly looks for an easier way of doing things. A system is needed to facilitate program construction. With the WH-89 computer comes the Heathkit Disk Operating System. Included with HDOS are all of the subsystems needed to make up a software development system.
The first step in software development is editing. The text editor, EDIT, allows the user to use mnemonics instead of the octal or hexadecimal code for each machine code instruction. More importantly, the user can label the line for reference anywhere in the subroutine. The assembler automatically determines the hexadecimal location of or distance to the labeled byte. Lines, variables, or locations in memory may all be assigned labels.

The text editor allows the user to make corrections to the text as he writes it. In addition to the correction functions, the program uses a series of error messages to inform the user of basic programming mistakes.

Once a program has been successfully edited and filed, the assembler, ASM, can be called in to convert all of the mnemonics to machine code. The assembly process creates a file of code free of all the remarks and superfluous wording generated to aid the user. It is this code that will be loaded onto cassette and transferred by the field support unit to the submarine computer.

The final step in the software development process is debugging the program. The console debugger, DBUG, facilitates the otherwise impossible task of finding problems in a machine code program. Table 3-2 lists the
operations available to the user. The most important functions allow the user to examine the Z-80's registers, execute one instruction at a time, or put a breakpoint in the program. Once an error has been observed, breakpoints can be systematically entered into the programs to stop it at any point. Registers can then be examined to determine if the expected results are there. The user can then step through the program until he finds a discrepancy. Once the error is localized, the user can make the corrections necessary without going back to the editor and assembler.

- Load user programs.
- Execute the program.
- Insert a breakpoint.
- Examine registers.
- Examine memory locations.
- Alter memory locations.
- Examine registers.
- Alter registers.
- Save user programs.

**TABLE 3-2** Debugging Functions
The HDOS system also supports the BASIC language. This language can be used in the administrative and research functions of the laboratory support system.

3.3.2 **Bootstrap Program**

When the submarine microcomputer system is powered up, the CPU will automatically look at location $\phi\phi\phi\phi\phi$. This will be the starting address of the bootstrap program. The program will shift to a subroutine which waits to establish a link with the field support unit. All of the software required for this communications link will be included in the bootstrap program. Once the link has been made, the submarine's memory may be loaded. The bootstraping program, as well as emergency programming, will be located in the read-only-memory.

3.3.3 **Test and Evaluation**

While the submarine is connected to the field support unit, tests will be conducted on the various systems required in its operation. A testing subroutine will be needed to put the systems through their paces, retrieve the results, and make an evaluation of the success of the test. A series of tests will be run in the laboratory as bench tests (see Table 3-3). Another series of tests will be run in the open water (see Table 3-4).
Read-Only-Memory
Random-Access-Memory
Multiply Unit
Pitch and Roll Sensors
Compass
Fin Servomotor System
Propulsion Motor
Magnetic Tape System
Interrupt System

TABLE 3-3  Bench Test Requirements

Depth Sensor
Communications Sonar
Collision Sonar
Bottom Finding Sonar
Side-Scan Sonar
Water Sensor
Fin Servomotor System

TABLE 3-4  Open Water Test Requirements
Test of the microcomputer system will include ROM, RAM, and the multiply unit. ROM will be tested by parity and check-sum. RAM will be tested by setting and resetting each bit and testing to determine whether it was or not. The test subroutine will systematically order each combination of multiplication and check it against a value determined by a software subroutine.

The pitch and roll sensors will be aligned in the laboratory to ensure the proper attitude for zero readings. The fin positions should be centerlined when the attitude sensors read zero. A subroutine will also have to check the pitch and roll sensors and fin servomotors as the submarine's position and attitude are changed. Once in the field, the open water response of the fin servomotors should be checked.

The compass and its effects on the fin servomotor system can be tested in a manner similar to that described above.

The sonars and depth sensor should be tested in water. This may be accomplished in a tank or at the mission site.

3.3.4 **Autopilot and Mission Programming**

Once the submarine is ready to begin its self-controlled operations, the microcomputer will initiate
executive control. Figure 3-2 illustrates the system to be used. The executive task controller monitors the various clocks checking to see which function needs to be serviced. The decision is based on the function's priority, frequency of execution, and time required to carry out its execution.

Four clocks are automatically checked by the executive task controller. Every 50 m Sec, 100 m Sec, 1 Sec, and 1 min the controller reads specific areas of the files to determine which subroutines will be executed and under what priority. In addition to those stated, user defined clocks can be created for any multiple of the 50 m Sec clock.

There are three types of files to be read. System files are those associated with the digital autopilot and the routine functioning of the submarine. Storage files are available for data collection. User files allows the operator to include a function in the submarine's operation. This might include the monitoring of data and a subsequent mission alteration or the transmission of specific bits of information by the communications sonar. Each file will contain a status flag, frequency interval, and destination of the subroutine to be jumped to. The
FIGURE 3-2 Mission Programming Block Diagram
frequency interval tells the executive task controller how often the subroutine must be carried out. It is also the basis by which the file is located. Each file will be dumped in a "bucket" according to the frequency of its servicing. A storage file has the location of the register to be stored instead of a jump location. In addition, a storage file must specify where the byte is to be stored. The executive task controller will increment this location each time a byte is stored.

The executive task controller may direct a jump to a user subroutine. The other possible jump is to a system subroutine. The only system defined at this point is the digital autopilot. Other subroutines may be developed for the routine operation of the submarine and may be defined as system commands.

Every 50 m Sec the digital autopilot reads the sensor registers for current depth and heading, fin positions, pitch, and keel. Every 100 m Sec it will take the two sets of readings and calculate a derivative. The command registers will be read to determine the required depth and heading. The new sensor readings, the derivatives, the desired readings and the coefficients calculated in Reference 3 will be used to build a matrix. The fin commands are the result of the evaluated matrix.
The command register contains the depth, headings, and other commands associated with the mission. A register will tell the autopilot when to change course and/or depth. A command register may also tell the submarine microcomputer when to turn on or turn off the side-scan sonar, start or stop the propulsion motor, or end the mission.

The difference between a file and a command register may need clarification. A file is established for each function that must be carried out repeatedly at regular intervals. A command register is created for each time, real or lapsed, at which an action must occur (i.e. a change in course or a sonar turned on).
CONCLUSIONS

Construction has continued on the second generation robot submarine. The electronics system has passed from the paper stage to the hardware stage. Nearly all major systems have been laid out and critical parts acquired. The chassis is being mounted in the electrical compartment, the batteries are installed, and the intercompartmental wiring complete.

Several systems have been completely assembled including the CPU/memory module, multiplier module, compass mounting, and most of the field support unit. Other systems have been breadboarded and soon should be assembled. These include the bottom finding, collision avoidance, and communications sonars, DAC, and servo motor systems. The laboratory support system has been acquired and the digital autopilot subroutines are ready for debugging.

There is still a long way to go before the second generation robot submarine will operate on its own. A great deal of assembly, integration, and testing needs to be done. Attention must be turned to the software requirements. Work should resume on the development of the side-scan sonar system.

The second generation robot submarine will certainly develop into a sophisticated tool. Its search and survey potential in the ocean environment is significant.
Significant also is the wide range of improvements that can be made to upgrade the system and further enhance its usefulness.
REFERENCES


APPENDICES
APPENDIX I

CENTRAL PROCESSING UNIT AND MEMORY DESIGN

The Z-80 microprocessor chip has 40 connections to be made. This appendix presents the circuitry used to wire the Z-80, the 4116 dynamic memory chips, the 2516 PROM, and the dedicated bus connections. The CPU/Memory board can be divided into five areas. Each area and the corresponding signals it generates is listed in Table A-I-1.

Address and Data Bus Buffering
Memory Matrix and Multiplexing
Memory Refresh and Control
4 and 8 MHz Clock
Bus Controls and Z-80 Inputs

<table>
<thead>
<tr>
<th>Address and Data Bus Buffering</th>
<th>A15-A, D7-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Matrix and Multiplexing</td>
<td>MA6-MA</td>
</tr>
<tr>
<td>Memory Refresh and Control</td>
<td>RAS, CAS, MUXSEL</td>
</tr>
<tr>
<td>4 and 8 MHz Clock</td>
<td>φ4, φ8</td>
</tr>
<tr>
<td>Bus Controls and Z-80 Inputs</td>
<td>INT, NMI, INTA, RESET</td>
</tr>
<tr>
<td></td>
<td>IN, OUT, MRD, MWR</td>
</tr>
</tbody>
</table>

TABLE A-I-1 CPU/Memory Board Organization

Address and data bus buffering, the 4 and 8 MHz clock, the bus control signals, and the Z-80 signals are straightforward and can be identified easily in Figure A-I-1. The circuitry involved in wiring the 4116 memory chips is not quite as easy to follow. The circuit used is taken
from Reference 8. The 4116 stores its 16384 bits in a 128 x 128 matrix. Two 74LS157, quad 1-to-2 multiplexers, latch in addresses A$ to A6 for the row address strobe signal, $\overline{RAS}$. Immediately after, they latch in addresses A7 to A13 for the column address strobe signal, $\overline{CAS}$. This switching is required at twice the microprocessor's speed. The use of Schottky devices (instead of low-power Schottky devices) is required to meet the speed demands of the 8MHz clocking.

As the name implies, the memory refresh and control circuitry must provide more then the $\overline{RAS}$, $\overline{CAS}$, and $\overline{MUXSEL}$ signals. Dynamic RAMs must be continuously refreshed. This circuitry performs that function. A refresh operation occurs during each opcode fetch cycle. The $\overline{RFSH}$ signal goes low and addresses A$ to A6 are used to refresh an entire row on all eight memory chips. The Z-80 possesses an internal counter which is incremented each time the operation occurs. In this manner 128 refresh cycles succeed in freshing the entire memory.
There are three areas of the multiplier system design. The first is the design of the cascaded 2 x 2 multiplier chips for 8 x 8 multiplication. This will not be presented here as it is taken directly from page 7-560 of the Motorola CMOS Databook.

Figure A-II-1 shows the circuit designed to enter the 8 bit X multiplicand and 8 bit Y multiplicand and retrieve the 16 bit S multiplicand. The logic is specially designed to take advantage of the 16 bit load instructions available with the Z-80 microprocessor. The gates produce addresses as follows:

\[
\begin{align*}
X &= F\#1H \\
Y &= F\#2H \\
S(\text{MS8}) &= F\#5H \\
S(\text{LS8}) &= F\#6H
\end{align*}
\]

If F\#3H is addressed, the same value can be simultaneously entered as both the X and Y multiplicand. This makes squaring operations extremely simple.

To perform a multiply operation the numbers are entered into the H and L registers and a CALL is executed.
A 74LS25  4 bit comparator
B 74LS00  Quad 2 input NAND
C 74LS08  Quad 2 input AND
D,E 74C374  Octal latch
F,G 74LS374  Octal latch

FIGURE A-II-1 Multiply Unit Interface Design
to MLTPLY. The subroutine returns with S in the DE register and HL unaffected. Squaring requires the number in the A register and a CALL to SQUARE.

SQUARE      LD(F$FF3H),A
            JR MIDMUL
MLTPLY      LD(F$FF1H),HL
MIDMUL      PUSH AF       ; DELAY
            POP AF       ; DELAY
            LD DE,(F$FF5H)
            RET
APPENDIX III

INTERRUPT PROCESSING SYSTEM

The interrupt processing circuit of Figure A-III-1 is a compact design capable of handling all of the initial demands of the submarine system. With seven chips it handles both the maskable and non-maskable interrupts. It has two unassigned inputs, one maskable and one non-maskable, for expansion.

The maskable input signals are held by flip-flops until the servicing subroutine resets them. Resetting is accomplished by strobing the memory mapped, 3-to-8 decoder. When any input signal is present, the AND gate triggers the interrupt and the interrupt acknowledge signal latches the identification code onto the data bus by strobing the tri-state buffer.

The output of the buffer will have D7 zero and D5-D7 preset such that the number present on the data bus will be the lower eight bits of an address in the interrupt vector table. In Mode 2 interrupt processing, the contents of the I register will provide the upper eight bits of the address.

The interrupt vector table contains the servicing subroutine addresses. This table also prioritizes the
A,B 74C74 Dual D-type flip-flop
c C,F 74LS367 Hex tri-state buffer
d 74C21 Dual 4 input NAND gate
e 74LS138 3-to-8 decoder
g CD4528 Dual monostable multivibrator

FIGURE A-III-1 Interrupt Processing Circuit
servicing by placing the vector for the highest priority subroutine in the table where more then one interrupt is present. For four maskable interrupts there will be 16 combinations requiring a 32 byte table. The following is an example of how this is accomplished. Addresses are totally arbitrary.

The non-maskable interrupt circuit is much simpler. When any interrupt is present, the AND gate triggers a monostable multivibrator. The multivibrator is necessary to provide a pulse input to the NMI pin. Sources describing the Z-80 chip suggest an input pulse of 250-750 ns for a 2 MHz clock. The RC circuit of the multivibrator can be easily adjusted to provide a pulse of any length.

The tri-state buffer will provide a signal which can be read by the CPU to determine which input cased the interrupt. This will be invaluable if some error or failure is causing the system to repeatedly abort. Inputs must be held low which will require a latch in some cases.
<table>
<thead>
<tr>
<th>Priority</th>
<th>Interrupt</th>
<th>Address*</th>
<th>Data Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>50ms Clock</td>
<td>φ1φ4H</td>
<td>4</td>
</tr>
<tr>
<td>2nd</td>
<td>Collision Sonar</td>
<td>φ5FFH</td>
<td>3</td>
</tr>
<tr>
<td>3rd</td>
<td>Comm Sonar</td>
<td>φ77AH</td>
<td>2</td>
</tr>
<tr>
<td>4th</td>
<td>Unassigned</td>
<td>φA23H</td>
<td>1</td>
</tr>
</tbody>
</table>

Location of interrupt vector table (lowest address) = AF62H*

Given the above:

I register = AFH

Dφ and D7 wired φ

D5 and D6 wired 1

(lowest combination = φ11φφφφ = 62H)

Vector Table:

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF62H</td>
<td>23H</td>
<td>AF72H</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF63H</td>
<td>φAH</td>
<td>AF73H</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF64H</td>
<td>23H</td>
<td>AF74H</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF65H</td>
<td>φAH</td>
<td>AF75H</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF66H</td>
<td>7AH</td>
<td>AF76H</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF67H</td>
<td>φ7H</td>
<td>AF77H</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF68H</td>
<td>7AH</td>
<td>AF78H</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF69H</td>
<td>φ7H</td>
<td>AF79H</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF6AH</td>
<td>FFH</td>
<td>AF7AH</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF6BH</td>
<td>φ5H</td>
<td>AF7BH</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF6CH</td>
<td>FFH</td>
<td>AF7CH</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF6DH</td>
<td>φ5H</td>
<td>AF7DH</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF6EH</td>
<td>FFH</td>
<td>AF7EH</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF7FH</td>
<td>φ5H</td>
<td>AF7FH</td>
<td>φ1H</td>
</tr>
<tr>
<td>AF7FH</td>
<td>FFH</td>
<td>AF8FH</td>
<td>φ4H</td>
</tr>
<tr>
<td>AF71H</td>
<td>φ5H</td>
<td>AF81H</td>
<td>φ1H</td>
</tr>
</tbody>
</table>
The following steps should be followed in the processing of maskable interrupts:

1. One or more functions trigger interrupts.
2. Interrupting functions automatically resets itself.
3. Vector table vectors to highest priority subroutine.
4. Subroutine calls SAVREG
   
   SAVREG  PUSH DE
   PUSH BC
   PUSH AF
   RET

   Note: If used by servicing subroutine, IX or IY must be pushed and popped separately.

5. Subroutine resets only its interrupt latch.

   LD A, $2H ; load reset code
   LD (FAΦH),A ; resets latch, memory mapped

   Note: $2H = 50 ms clock, $4H = Coll sonar,
   $8H = Comm Sonar.

6. Subroutine services interrupt.
7. Subroutine jumps to RET INT

    RETINT  POP AF
    POP BC
    POP DE
    EI
    RETI

8. If interrupt contains another, system immediately goes to #3.
APPENDIX IV

DIGITAL INTERFACE DESIGN

The digital interface circuit in Figure A-IV-1 is essentially an address decoding system. Two types of output signals are provided. An address decoded enable signal is provided for read operations. This long pulse is necessary to allow data to stabilize on the bus. For write operations the address decoding is combined with the \( \overline{MWR} (\overline{MEMRQ} + \overline{WR}) \) signal to provide a strobing pulse.

The hex inverter chip has five unassigned inverting gates which can be used to provide positive pulses as is done for the octal latch controlling the ON/OFF signals.

ON/OFF control is incorporated into the digital interface. Eight signals are supported by the circuit shown. More can be added by attaching another octal latch to one of the unassigned outputs of the write pulse decoder. To turn a device on or off will require setting or resetting a bit in a byte stored in memory keeping the current status. The new status can then be written to the latch.

For example, assume the status is held in memory location AFFFFH and the side scan sonar (on line D4) must get a 1 to be turned. The following program will turn on the side scan sonar without affecting other devices:
A 74LS85 4 bit magnitude
B,C 74LS138 3-to-8 decoder
D 74LS04 Hex inverter
E 74C374 Octal latch

FIGURE A-IV-1 Digital Interface Circuit
LD A, (AFFFH) ; RETRIEVES STATUS BYTE
SET 4, A ; CHANGES STATUS OF SIDE SCAN
LD (FB8FH), A ; TURNS ON SIDE SCAN

The outputs of the digital interface octal latch will control latches, flip-flops, and relays.
<table>
<thead>
<tr>
<th>A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1</th>
<th>Function</th>
<th>Address (10 West)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU RD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 x x x x x x</td>
<td>ADC MS3</td>
<td>FF8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 0 x x x x x x</td>
<td>ADC LS3</td>
<td>FF&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 0 1 0 x x x x x x</td>
<td>COMM SONAR</td>
<td>FE8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 0 0 0 x x x x x x</td>
<td>COLL SONAR</td>
<td>FE&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 0 1 1 x x x x x x</td>
<td>BOT F MS8</td>
<td>FD8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 0 1 0 x x x x x x</td>
<td>BOT F LS8</td>
<td>FD&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 0 0 0 x x x x x x</td>
<td>UNASSIGNED</td>
<td>FC8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 0 0 0 x x x x x x</td>
<td>UNASSIGNED</td>
<td>FC&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>CPU WR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 1 1 1 x x x x x x</td>
<td>ON/OFF</td>
<td>FB8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 1 1 0 x x x x x x</td>
<td>DAC MS8</td>
<td>FB&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 1 0 1 x x x x x x</td>
<td>DAC LS8</td>
<td>FA8&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 1 0 0 x x x x x x</td>
<td>INTRST</td>
<td>FA&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 0 1 1 x x x x x x</td>
<td>UNASSIGNED</td>
<td>F98&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 0 0 1 0 x x x x x x</td>
<td>UNASSIGNED</td>
<td>F9&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 0 0 1 x x x x x x</td>
<td>UNASSIGNED</td>
<td>F88&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0 0 0 0 0 x x x x x x</td>
<td>UNASSIGNED</td>
<td>F&lt;sub&gt;8&lt;/sub&gt;H</td>
</tr>
</tbody>
</table>

1. Identifies memory mapped function.
2. Selects memory mapped segment.
3. I=RD function provides address decoded enable 0=WR function provides write pulse
4. Function select.
5. Available for function coding.

TABLE A-IV-1 Digital Interface Coding
APPENDIX V

COMMUNICATION PROCESSING

When a command is sent to the submarine, the maskable interrupt will be triggered and four bits of data will be present at a latch. The interrupt processor will vector execution to the subroutine described in this Appendix. Figure A-V-1 describes the requirements of the communications processing subroutine.

The first step is to save the register that will be altered by the subroutine and restore information that has been saved. These two functions can be performed at the same time by reserving the second set of registers for communications processing and using the exchange instructions.

After resetting the communication's interrupt latch, the subroutine tests the Z flag which has been preserved. If it is zero, the transmission is a command signal and not a data chew. (A chew is four bits). The command is loaded by addressing the memory mapped location of the latch. Checking bit 4 will reveal whether it is a single or multiple instruction command. If it is zero the command is executed; otherwise a 2 is loaded into the B register, the Z flag is set and the subroutine is exited.
If the Z flag is tested and it is not zero, the subroutine will load a data chew. If it is the first chew it will be shifted to the upper four bits and saved. When the lower four bits enter on the following transmission they will be added to the upper four bits and stored in a register reserved for the data port-on of the command.

The subroutine executes by loading the address from the vector table and executing a jump to that location. The 4 bits command signal must be shifted left (retaining all bits) to make it an even number. Addresses require two bytes of memory and, therefore, must be spaced accordingly in the vector table.

It should be noted that the subroutine presented in this appendix handles only two types of commands. One is single transmission and the other is a three transmission command which follows the command signal with two transmissions carrying 8 bits of data.

Tables A-V-1 and A-V-2 suggest some ways in which the commands may be allocated. Modifications to the subroutine could easily accommodate two, four, or more transmission commands.
FIGURE A-V-1 Communications Processing Flowchart
Communications Interrupt Subroutine

COMINT    EX AF, AF'

EXX

LD A, φφH ; RESET
LD (FAφφH), A ; LATCH
JR NZ, COMCON ; JP IF CONTINUATION
LD A, (Eφ1H) ; LOAD COMMAND
BIT 4, A ; CHECK A4
JR NZ, COMMTY ; JP IF MULTIPLE TRANS

COMEX    SLA A ; GENERATE ADDRESS

ADD A, CVL* ; OF VECTOR

LD L, A
LD H, CVU*
LD D, (HL)
DEC HL
LD E, (HL)
EX DE, HL
JP (HL) ; JP TO SUBROUTINE

COMMTY    LD C, A ; STORE COMMAND

OR C ; SET CONTINUATION FLAG

LD B, 2 ; SET STATUS

JR COMRET

*Note: CVU is the upper 8 bits and CVL is the lower 8 bits of the first address in the communications vector table.
COMCON  LD A, (Eφφ1H) ; LD CHEW
         DEC B
         JR Z, COMLST ; JP IF LAST CHEW
         SLA A ; PUSH TO UPPER 4
         SLA A
         SLA A
         SLA A
         OR B ; SETS CONTINUATION FLAG
         LD D, A ; STORE DATA
COMLST  ADD A, D ; ADD UPPER $
         LD D, A ; STORE DATA
         LD A, C ; BRING UP COMMAND
         JR COMEX ; GO TO EXECUTE
COMXRT  AND φφH ; ZERO CONTINUATION FLAG
COMRET  EX AF, AF'
         EXX
         EI
         RETI

Note: Subroutines servicing commands must end by executing a jump to COMXRT.

B' = STATUS REGISTER
C' = COMMAND REGISTER
D' = DATA REGISTER
F' = CONTINUATION FLAG (Z)
<table>
<thead>
<tr>
<th>CA₀</th>
<th>CA₂</th>
<th>CA₃</th>
<th>CA₄</th>
<th>CA₅</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Hard wired abort command</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(CA₄-CA₀ to 4 input OR gate)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Stop</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Forward</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Reverse</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Hard to Port</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Hard to Starboard</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Steady on Current Course</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Power Conservation</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>End Power Conservation</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Execute Predesignated Subroutine</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Switch to Alternative Mission</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Unassigned</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

TABLE A-V-1 Single Transmission Commands
<table>
<thead>
<tr>
<th>CA₄</th>
<th>CA₃</th>
<th>CA₂</th>
<th>CA₁</th>
<th>CA₀</th>
<th>Command</th>
<th>Data Specifying</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Change to Course ≤ 256°</td>
<td>Course</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Change to Course ≤ 256°</td>
<td>Course - 256°</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Change to Depth</td>
<td>Depth in Meters</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Execute Subroutine</td>
<td>Subroutine Vector</td>
</tr>
</tbody>
</table>

| 1   | 0   | 1   | 0   | 0   | Unassigned                       |                          |
| 1   | 0   | 1   | 0   | 1   |                                   |                          |
| 1   | 0   | 1   | 1   | 0   |                                   |                          |
| 1   | 0   | 1   | 1   | 1   |                                   |                          |
| 1   | 1   | 0   | 0   | 0   |                                   |                          |
| 1   | 1   | 0   | 0   | 1   |                                   |                          |
| 1   | 1   | 0   | 1   | 0   |                                   |                          |
| 1   | 1   | 0   | 1   | 1   |                                   |                          |
| 1   | 1   | 1   | 0   | 0   |                                   |                          |
| 1   | 1   | 1   | 0   | 1   |                                   |                          |
| 1   | 1   | 1   | 1   | 0   |                                   |                          |
| 1   | 1   | 1   | 1   | 1   |                                   |                          |

**TABLE A-V-2**  Multiple Transmission Commands