MICROPROCESSOR CONTROL
OF
QUARTER WATT LINEAR COOLER

FINAL REPORT
AUGUST 1, 1987
DECEMBER 5, 1986 TO JULY 31, 1987

Prepared under Contract Number DAAB07-87-C-F018
for Center for Night Vision and Electro-Optics

MAGNAVOX GOVERNMENT AND INDUSTRIAL ELECTRONICS COMPANY
ELECTRO-OPTICAL SYSTEMS, 46 INDUSTRIAL AVENUE
MAHWHAH, NJ 07430
MICROPROCESSOR CONTROL
OF
QUARTER WATT LINEAR COOLER

FINAL REPORT
AUGUST 7, 1987
DECEMBER 5, 1986 TO JULY 31, 1987

Prepared under Contract Number DAAB07-87-C-F018
for Center for Night Vision and Electro-Optics

MAGNAVOX GOVERNMENT AND INDUSTRIAL ELECTRONICS COMPANY
ELECTRO-OPTICAL SYSTEMS, 46 INDUSTRIAL AVENUE
MAHWAN, NJ 07430

Approved by: A. Silver
Deputy Director of Engineering
TABLE OF CONTENTS

1.0 INTRODUCTION ...................................................... 1
1.1 MOTIVATION FOR MICROPROCESSOR/DIGITAL CONTROL .................... 1
1.2 SWITCHING FREQUENCY ............................................ 1
2.0 DESCRIPTION OF HARDWARE ........................................ 4
2.1 COMPUTER ................................................................ 4
2.2 DIGITAL PULSE WIDTH MODULATOR (DPWM) - IXDP610 .................... 5
2.3 TIMING DIAGRAM ................................................... 5
2.4 LEVEL SHIFT AND POWER AMPLIFIER .................................. 5
3.0 DESCRIPTION OF SOFTWARE .......................................... 10
4.0 SURVEY OF MICROCONTROLLERS .................................... 13
5.0 SELECTION OF SWITCHING FREQUENCY ................................ 15
5.1 VOLUME CONSTRAINT .................................................. 15
5.2 DIGITAL HARDWARE CONSTRAINT .................................... 15
5.3 THE SELECTION ........................................................ 15
6.0 SUMMARY ............................................................... 15
7.0 RECOMMENDATIONS FOR FURTHER STUDY ............................ 16

LIST OF TABLES AND ILLUSTRATIONS

Figure
1 Analog Cooler Control .................................................. 2
2 Digital/Microprocessor Cooler Control ................................ 3
3 Micro Controller ........................................................ 6
4 Hardware Demonstration ................................................ 7
5 Block Diagram - IXDP610 .............................................. 8
6 Timing Diagram ........................................................... 9
7 Software Flowchart ...................................................... 11

Table
1 Relative Comparison of Analog Versus Digital Design ............ 4
2 Look Up Table ............................................................ 12
3 Microcontroller Survey ................................................ 14

LIST OF APPENDICES

Appendix I - Demonstration Software
ABSTRACT

The work carried out in this study was performed by Magnavox Government & Industrial Electronics Company, Electro-Optical Systems, in accordance with the requirements of contract DAAB07-87-C-F018. The purpose of the program was to conduct studies of microcontroller/digital VLSI technologies for possible future use in cryogenic coolers.

A necessary condition that any electronic approach must satisfy in order to be considered for linear cooler electronics is the ability to work in the real time environment of 100 kHz or higher switching speeds. In the case of the existing analog design, this prerequisite has long been established to achieve practically sized line filters. Heretofore this speed was out of reach of microprocessor based, controllers since the state-of-the-art was somewhere in the neighborhood of 8 to 10 kHz. However, a new device on the market has provided the necessary interface hardware to bridge this gap. This advancement combined with the advances of microcontrollers in terms of cost, size, and performance suggested that an investigation of the microcontroller/digital approach for the Magnavox quarter watt cooler electronics be performed.
INTRODUCTION

The work carried out in this study was performed by Magnavox Government & Industrial Electronics Company, Electro-Optical Systems, in accordance with the requirements of contract DAA07-87-C-F018. The purpose of the program was to conduct studies of microcontroller/digital VLSI technologies for possible future use in Magnavox cryogenic coolers.

1.1 MOTIVATION FOR MICROPROCESSOR/DIGITAL CONTROL

The present technology used for Magnavox cooler linear motor control is based on an analog circuit design. The diagram shown in Figure 1 is representative of the approach to meet a requirement for closed-loop cold station temperature control. Although this analog design functions well, there are compelling reasons to look in the direction of microcontroller/digital electronics, Figure 2, for future designs.

The first and most important reason is potential cost savings. The analog approach requires hand assembled control hybrids with expensive laser trimming. This is in contrast to the microcontroller approach that utilizes low cost, mass produced IC's.

A second potential benefit from a microcontroller approach is accuracy. A performance advantage is achieved in maintaining accurate temperature control of the detector cold station. As the tolerance on temperature is tightened the cost for accurate trimming of the analog design increases exponentially. In the case of the microcontroller, accuracies to ±1°K are easily attainable. Additionally, long term thermal drift, and component variability, are not a concern.

There are advantages in flexibility. Such things as customer requirements for the temperature set point or specialized modes of operation to conserve power can be satisfied by simple changes in software. New drive wave forms, easily generated in software, could have a favorable impact on efficiency and/or reduction of mechanical vibration.

1.2 SWITCHING FREQUENCY

With these advantages, summarized in Table 1, the natural question is why weren't microcontroller electronics used sooner? The answer is that the volume constraints on the cooler electronics requires that the switching frequency of the power amplifier be above 100 kHz. As the frequency is reduced from this level the input EMI filter components tend to grow. Since they just fit now, any growth would mean a deterioration in the performance of the filter.
ANALOG COOLER CONTROL

FIGURE 1


Table 1. Relative Comparison of Analog Versus Digital Design

<table>
<thead>
<tr>
<th></th>
<th>Analog Control</th>
<th>Digital Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts Count</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>High Volume Cost</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Low Volume Cost</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Reliability</td>
<td>Good</td>
<td>Better</td>
</tr>
<tr>
<td>Accuracy</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Specifically, the volume available in the cooler for the electronics is 0.7 cubic inches. The analog package as it is currently configured just fits into this volume - no margin for expansion. Of that volume approximately 20% is occupied by the input EMI filters. These components are designed for the 100 kHz switching frequency. If we were to lower the frequency by a decade, the volume occupied by the filter chokes would increase by approximately 50% to maintain the same impedance level. The filter capacitors would have to be increased by 3 to 4 times depending on ripple considerations. For these reasons, switching frequencies in the range of 10 kHz are out of the question.

As mentioned previously, until recently 100 kHz real time control was out of reach of microcontroller technology. This would require algorithmic execution speeds in the order of 10 microseconds which is out of reach for present day microcontrollers. However, a new interface PWM chip overcomes this and is a focus of attention for this study.

2.0 DESCRIPTION OF HARDWARE

2.1 COMPUTER

The design shown in Figure 2 represents a production concept for the microcontroller/digital electronics. Note that this is closed loop on both motor voltage and cold station temperature. To achieve this, the basic functions of the microcontroller are:

1. Get the appropriate PWM level from the data stored in memory.
2. Multiply that number (thereby changing the gain) to adjust for variation of motor voltage from command set point.
3. Multiply a second time (again modifying the gain) to adjust for variations in thermal loading.
4. Write to the digital PWM chip the appropriate code in accordance with the above calculation.
A typical microcontroller, as shown in Figure 3, is a single chip microcomputer. It is compatible with all system timing, internal logic, ROM, RAM and I/O. Although it is not obvious in Figure 2, the microcontroller has on board the required A/D converters for closed loop control.

For the purpose of a demonstration it was convenient to use a "single board computer" (SBC) to simulate functions of a microcontroller. The arrangement shown in Figure 4, used an Ampro SBC, with the Intel 80186 microprocessor, and was programmed by an Applied Microsystem, ES 1800 Satellite Emulator. This provided easy manipulation of the code through PL/M on the PC.

2.2 DIGITAL PULSE WIDTH MODULATOR (DPWM) - IXDP610

The ability to reach switching frequencies greater than 100 kHz is achieved by the IXDP610 Digital Pulse Width Modulator (DPWM) chip. This chip is a major breakthrough in high frequency motor control applications. Without it switching would be limited to approximately 8 kHz. As discussed previously, this would eliminate consideration of the microcontroller approach.

A block diagram of the IXDP610 (DPWM) is shown in Figure 5. The programmable, CMOS, LSI device receives digital pulse width data (8 bits) from the microcontroller and generates a TTL pulse width modulated signal. The DPWM is designed for direct control by the microcontroller. Current limiting on a cycle by cycle basis can be asserted through "Output Disable Logic" (ODIS).

2.3 TIMING DIAGRAM

The timing diagram of Figure 6 shows the digital aspects of this topology. Figure 2, shows that the master clock is crystal controlled and set at 16 MHz. This clock signal together with the appropriate input to the programmable digital PWM controller, sets the switching frequency as follows:

\[ \text{PWM base period} = \text{Clock Period} \times 128 \]

8 Microsecond (125 kHz) = 63 nanoseconds (16 MHz) x 128

The rate at which the CPU strobes through the look up table is controlled by a programmable counter which is fed by an external 125 kHz signal. The counter is set to count 25, 8 \( \mu \)s counts (8 \( \mu \)s x 25 = 200 \( \mu \)s). Upon decrementing to a 0 count, an interrupt is pulled. At this time the current value is taken out of the look up table, outputted to a port and the table pointer incremented. This data is outputted to the IXDP610 latch. With the "Chip Select" low, the data is written to the pulse width latch. Under the steady state mode of operation, the pulse width of the power amplifier is fixed for the next 200 \( \mu \)s, after which another CS pulse will allow an update.

2.4 LEVEL SHIFT AND POWER AMPLIFIER

With a TTL pulse width modulation signal from the DPWM chip, the next requirement is to amplify the signal and drive the motor. This part of the
In production this would be replaced by a microcontroller.
FIGURE 6. Timing Diagram

-9-
3.0 DESCRIPTION OF SOFTWARE

Algorithms were developed to implement open loop control of the cooler motor. A fixed input line voltage (24 Vdc) was assumed. A flowchart of this software is shown in Figure 7, with code, as developed using PL/M, listed in Appendix I.

Lookup Table (Table 2)

For the demonstration, the function of the computer was to update the DPWM chip with an 8 bit digital input. Basically, by means of the lookup table it is generating a digital sine wave command signal. The theory behind the development of the lookup table is very basic.

Analog Sine Wave: \( V(t) = V_m \times \sin(2\pi f t) \times k \times d \)

- \( V_m \): peak motor voltage (14 volts)
- \( f \): motor frequency (54 Hz)
- \( v(t) \): instantaneous motor voltage
- \( k \): sample point
- \( d \): sampling interval

The voltage \( v(t) \) is the sample value of the required cooler motor voltage at time \( k \times d \). The sampling interval for demonstration was approximately 200 microseconds. Therefore the sine wave is approximated by steps that change every 200 microseconds, or 94 steps.

To generate the table therefore only the first half of the sine wave need be sampled or approximately 9.25 milliseconds for a 54 Hz sine wave. The values for a 14 volt peak or 9.89 Vrms sine wave are listed in column 2 of Table 2. The relationship between these voltage levels and the Pulse Width Modulator or Duty Cycle signal is as follows:

\[ v(t) = \left( \frac{T_{on}}{T} \right) \times V_{in} = P \times Vin \]

- \( P \): pulse width modulation level (range 0 - 1.0)
- \( T \): switching period
- \( T_{on} \): on time of power switches
- \( Vin \): line voltage

Since we were operating open loop, the line voltage (Vin) was set to be constant at 24 Vdc. With this level established, the corresponding PWM levels are shown in column 3 of Table 2. These levels are then converted to the nearest hexadecimal number as listed in column 4 of Table 2.
SOFTWARE FLOWCHART

- INITIALIZE 186 uP
- INITIALIZE UPDATE INTERRUPT
- INITIALIZE DIGITAL PWM CHIP

- TIMER #1
- ACKNOWLEDGE INTERRUPT

- INCREMENT TABLE POINTER

- APPLY BYTE TO IXYS

FIGURE 7
<table>
<thead>
<tr>
<th>POINT</th>
<th>VOLTS (Vt)</th>
<th>DUTY CYCLE (PWM)</th>
<th>HEX NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>3.894265E-02</td>
<td>000h</td>
</tr>
<tr>
<td>1</td>
<td>0.9346235</td>
<td>7.771154E-02</td>
<td>00Ah</td>
</tr>
<tr>
<td>2</td>
<td>1.865077</td>
<td>0.1161337</td>
<td>014h</td>
</tr>
<tr>
<td>3</td>
<td>2.787209</td>
<td>0.1540377</td>
<td>028h</td>
</tr>
<tr>
<td>4</td>
<td>3.696905</td>
<td>0.1912545</td>
<td>031h</td>
</tr>
<tr>
<td>5</td>
<td>4.590107</td>
<td>0.2276179</td>
<td>03Bh</td>
</tr>
<tr>
<td>6</td>
<td>5.462829</td>
<td>0.2629657</td>
<td>044h</td>
</tr>
<tr>
<td>7</td>
<td>6.311178</td>
<td>0.2971403</td>
<td>04Dh</td>
</tr>
<tr>
<td>8</td>
<td>7.131367</td>
<td>0.3299891</td>
<td>055h</td>
</tr>
<tr>
<td>9</td>
<td>7.919739</td>
<td>0.3613657</td>
<td>05 Eh</td>
</tr>
<tr>
<td>10</td>
<td>8.672775</td>
<td>0.3911298</td>
<td>065h</td>
</tr>
<tr>
<td>11</td>
<td>9.387114</td>
<td>0.4191488</td>
<td>06Ch</td>
</tr>
<tr>
<td>12</td>
<td>10.05957</td>
<td>0.4452979</td>
<td>073h</td>
</tr>
<tr>
<td>13</td>
<td>10.68715</td>
<td>0.46946</td>
<td>079h</td>
</tr>
<tr>
<td>14</td>
<td>11.26704</td>
<td>0.4915276</td>
<td>07Fh</td>
</tr>
<tr>
<td>15</td>
<td>11.79666</td>
<td>0.5114021</td>
<td>084h</td>
</tr>
<tr>
<td>16</td>
<td>12.27365</td>
<td>0.5289948</td>
<td>088h</td>
</tr>
<tr>
<td>17</td>
<td>12.69588</td>
<td>0.5442274</td>
<td>08Ch</td>
</tr>
<tr>
<td>18</td>
<td>13.06146</td>
<td>0.5570318</td>
<td>08Fh</td>
</tr>
<tr>
<td>19</td>
<td>13.36876</td>
<td>0.5673508</td>
<td>092h</td>
</tr>
<tr>
<td>20</td>
<td>13.61642</td>
<td>0.5751385</td>
<td>093h</td>
</tr>
<tr>
<td>21</td>
<td>13.80333</td>
<td>0.58036</td>
<td>094h</td>
</tr>
<tr>
<td>22</td>
<td>13.92864</td>
<td>0.5829922</td>
<td>095h</td>
</tr>
<tr>
<td>23</td>
<td>13.99181</td>
<td>0.5830233</td>
<td>094h</td>
</tr>
<tr>
<td>24</td>
<td>13.99256</td>
<td>0.5804531</td>
<td>093h</td>
</tr>
<tr>
<td>25</td>
<td>13.93087</td>
<td>0.575293</td>
<td>092h</td>
</tr>
<tr>
<td>26</td>
<td>13.80703</td>
<td>0.5675661</td>
<td>08Fh</td>
</tr>
<tr>
<td>27</td>
<td>13.62159</td>
<td>0.557307</td>
<td>08Ch</td>
</tr>
<tr>
<td>28</td>
<td>13.37537</td>
<td>0.5445611</td>
<td>088h</td>
</tr>
<tr>
<td>29</td>
<td>13.06947</td>
<td>0.5293858</td>
<td>084h</td>
</tr>
<tr>
<td>30</td>
<td>12.70526</td>
<td>0.5118484</td>
<td>07Fh</td>
</tr>
<tr>
<td>31</td>
<td>12.28436</td>
<td>0.4920273</td>
<td>079h</td>
</tr>
<tr>
<td>32</td>
<td>11.80866</td>
<td>0.4700109</td>
<td>073h</td>
</tr>
<tr>
<td>33</td>
<td>11.28026</td>
<td>0.4458975</td>
<td>06Ch</td>
</tr>
<tr>
<td>34</td>
<td>10.70154</td>
<td>0.4197945</td>
<td>065h</td>
</tr>
<tr>
<td>35</td>
<td>10.07507</td>
<td>0.3918186</td>
<td>05Bh</td>
</tr>
<tr>
<td>36</td>
<td>9.403646</td>
<td>0.3620946</td>
<td>055h</td>
</tr>
<tr>
<td>37</td>
<td>8.690269</td>
<td>0.3307549</td>
<td>04Dh</td>
</tr>
<tr>
<td>38</td>
<td>7.938118</td>
<td>0.2979396</td>
<td>044h</td>
</tr>
<tr>
<td>39</td>
<td>7.150549</td>
<td>0.2637947</td>
<td>03Bh</td>
</tr>
<tr>
<td>40</td>
<td>6.331073</td>
<td>0.228473</td>
<td>031h</td>
</tr>
<tr>
<td>41</td>
<td>5.483353</td>
<td>0.192132</td>
<td>028h</td>
</tr>
<tr>
<td>42</td>
<td>4.611168</td>
<td>0.1549337</td>
<td>01Bh</td>
</tr>
<tr>
<td>43</td>
<td>3.718409</td>
<td>0.1170441</td>
<td>014h</td>
</tr>
<tr>
<td>44</td>
<td>2.80906</td>
<td>7.863237E-02</td>
<td>00Ah</td>
</tr>
<tr>
<td>45</td>
<td>1.887177</td>
<td>3.986962E-02</td>
<td>000h</td>
</tr>
<tr>
<td>46</td>
<td>0.9568709</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2. Look Up Table**
Note that on page 7 of the code listed in Appendix I, the lookup table is listed in accordance with Table 2. We emphasize again that this table is correct for 24 Vdc line voltage only and that in a final design will have to be scaled to accommodate a changing input line.

In addition to the lookup table, for the demonstration to be successfully implemented, algorithms were developed for the following functions:

1. Initialize Processor
2. Initialize Counter #1 — this counter controls the lookup table pointer — counts pulses at a 125 kHz clock and increments every 25 counts
3. Initialize PWM chip (IXDP610) i.e., deadtime set, lock bit off, 8 bit resolution.

4.0 SURVEY OF MICROCONTROLLERS

The heart of the microcontroller/digital approach to the motor driver electronics is the microcontroller. From the conception of the microprocessor in 1974 there have been many advancements in the VLSI technology of microprocessors. Improvements have steadily been made in performance such as processor speed, bus size and power consumption. However, probably most startling of all was the advancements in the ability of the IC manufacturers to pack more functions into a given area of substrate. It was the improvements made in the small size of the substrate that has made the evolution of the microcontroller possible.

A microcontroller differs from a microprocessor in that it is a self contained computer. Unlike a microprocessor it needs no support chips, and in fact, packs more function in the form of A/D converter, UART serial communication, interrupt controls and I/O. For these reasons Magnavox believes that a microcontroller is a natural progression for the next generation cooler electronics.

The major manufacturers of microcontrollers are: Intel, Motorola, National and Texas Instruments. Table 3, is a matrix that compares the existing microcontrollers and provide some of the primary features that relate to the cooler application. It should be pointed out that these features are as they exist at the time of this writing and because of the rapidly changing technology they would have to be reexamined again at the time of a design effort.
<table>
<thead>
<tr>
<th></th>
<th>INTEL</th>
<th>MOTOROLA</th>
<th>NATIONAL</th>
<th>TEXAS INSTRUMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Family</td>
<td>MCS-8096</td>
<td>M68HC11</td>
<td>HPC16084</td>
<td>7000 Series</td>
</tr>
<tr>
<td>A/O</td>
<td>10 Bit</td>
<td>4 Channel, 8 Bit</td>
<td>Not Yet</td>
<td>No</td>
</tr>
<tr>
<td>Bus</td>
<td>16 Bit</td>
<td>16 Bit</td>
<td>16 Bit</td>
<td>8 Bit</td>
</tr>
<tr>
<td>PWM Output</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Watchdog Timer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Timer</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>No. I/O Ports</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>Interrupt Sources</td>
<td>20</td>
<td>16</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>100 mW</td>
<td>3 Modes</td>
<td>100 mW</td>
<td>100 mW</td>
</tr>
<tr>
<td></td>
<td>Run - 50 mW</td>
<td>Wait - 3 mW</td>
<td>Stop - 1 mW</td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td>256 K</td>
<td>256 K</td>
<td>256 K</td>
<td>256 K</td>
</tr>
<tr>
<td>ROM</td>
<td>8 K</td>
<td>8 K</td>
<td>8 K</td>
<td>4 K</td>
</tr>
<tr>
<td>Price (Commercial)</td>
<td>$16</td>
<td>$16</td>
<td>Not Available</td>
<td>Not Available</td>
</tr>
<tr>
<td>MIL</td>
<td>Avail. In MIL</td>
<td>Will Be Avail.</td>
<td>Will Be Avail.</td>
<td>No</td>
</tr>
<tr>
<td>Software Develop-ment Status</td>
<td>Available</td>
<td>Available (HDS-300)</td>
<td>Available (Mole)</td>
<td>Available</td>
</tr>
</tbody>
</table>

Table 3. MICROCONTROLLER SURVEY
5.0 SELECTION OF SWITCHING FREQUENCY

5.1 VOLUME CONSTRAINT

The reason for the popularity of a PWM SWITCHMODE power driver is that this approach is very efficient. The theory behind this is that in the ideal, there are only two states for the switching devices - "on" and "off". In the "on" state we assume no voltage drop across the device and in the "off" state, no current flow. Therefore, in the ideal case no power is dissipated in the power amplifier. However, looking at the real device, there is a finite voltage drop when the device is "on" and a finite current when the device is "off". Furthermore there are switching losses when the devices change state. These losses increase with frequency.

This implies that from an efficiency point of view, it is best to keep the power amplifier switching frequency as low as possible. However, for the cooler electronics an overriding consideration is volume. For that reason, and as discussed earlier, the switching frequency must be above 100 kHz. Specifically, the size of the EMI components are frequency dependent and at frequencies much below 100 kHz, they do not fit.

5.2 DIGITAL HARDWARE CONSTRAINT

With the approach described we have hardware that can potentially switch up to 200 kHz. However, due to the requirements of the counters we have a practical constraint to choose frequencies that are divided by two, submultiples of the 16 MHz clock. This breaks down to frequency choices of 125 kHz, 62.5 kHz and 31.25 kHz. A frequency of 125 kHz was selected in accordance with the 100 kHz limitation described above.

5.3 THE SELECTION

At the present time 125 kHz is the best estimate of the optimum switching frequency. In future design efforts we would not expect any new inputs that would cause significant deviation from this frequency choice. However, the number of switching levels that go into the sine wave may vary since this number may influence the vibration profile of the cooler.

6.0 SUMMARY

The work carried out in this study was performed by Magnavox Government & Industrial Electronics Company, Electro-Optical Systems, in accordance with the requirements of contract DAAB07-87-C-F018. The purpose of the program was to conduct studies of microcontroller/digital VLSI technologies for possible future use in Magnavox cryogenic coolers. In pursuit of this
objective, the following specific tasks were required of the study:

Task No. 1: Study of Single Chip Microcomputers
Task No. 2: Determine Optimum Pulse Width Modulation Frequency
Task No. 3: Develop Algorithms

Although the scope of this study was initially conceived to be a white paper report/study only, Magnavox has taken the initiative to implement the basic concepts with a hardware demonstration. That is, a microcontroller was simulated using an Intel 80186 microprocessor and configured to drive an operating cooler. This clearly substantiates the microcontroller/digital approach and is planned as a demonstration for cognizant CNVEO personnel.

7.0 RECOMMENDATIONS FOR FURTHER STUDY

Based on the positive findings of this research effort it is recommended that additional funding be provided to implement a high switching frequency microprocessor/digital design program to further refine the cooler electronics. Specific tasks recommended for investigation include:

1. Investigation of nonsinusoidal motor voltage drive waveforms. This would be implemented by appropriate changes in the software. The potential benefits from this investigation would be lower vibration levels and increased operating efficiency of the cooler.

2. Investigate "SMART" cooler electronics. Self calibration, self test and diagnostics are a few of the possibilities to be explored.

3. Develop optimum stepped sinusoidal waveform. This will require careful analysis before establishing a firm number for the design.

4. Explore use of feedback signals representing the working gas temperature or pressure. Determine the effects of these signals on cooler performance, i.e., stability of temperature set point, efficiency and vibration.
APPENDIX I

DEMONSTRATION SOFTWARE
IBM PC-DOS PL/M-86 v2.3 Compilation of Module INT186
Object Module Placed in INT186.OBJ
Compiler Invoked By: PLM86 INT186.PLM DEBUG [REF CODE MOD186]

1  INT186:

DO:

2 1 DECLARE FLAG BYTE EXTERNAL;

/**********************************************************
 STARTING OF INITIALIZATION OF AP186, AND 2881 DUART'S
 UMCs is initialized in ass program STRTSEG at OFFFD
 **********************************************************/

3 1 Declare
   LMCS_REG literally 'OFFA2h', /* Chip select register locations*/
   PACS_REG literally 'OFFA4h',
   NPCS_REG literally 'OFFA6h',
   IMOC_REG literally 'OFFA8h';

4 1 Declare /* Timer control registers locations*/
   t2mode_reg literally 'OFF66h', /* Timer 2 mode control word*/
   t2maxa_reg literally 'OFF68h', /* Timer 2 max counter A*/
   t2count_reg literally 'OFF60h', /* Timer 2 counter register*/
   t1mode_reg literally 'OFF5Eh',
   t1maxa_reg literally 'OFF5Ch',
   t1count_reg literally 'OFF50h',
   t0mode_reg literally 'OFF56h',
   t0maxa_reg literally 'OFF54h',
   t0count_reg literally 'OFF50h',
   timer_int literally 'OFF32h';

5 1 Declare /* DMA control registers locations */
   daal_cntrl_reg literally 'OFFC0h', /*DMA channel 0 control word */
   daal_count_reg literally 'OFFC2h', /*DMA channel 0 transfer counter*/
   daal_dest_H_reg literally 'OFFCAh', /*DMA channel 0 destination high ptr*/
   daal_dest_L_reg literally 'OFFC4h', /*DMA channel 0 destination low ptr*/
   daal_src_H_reg literally 'OFFC2h', /*DMA channel 0 source high pointer*/
   daal_src_L_reg literally 'OFFC0h', /*DMA channel 0 source low pointer*/
   daal_cndt_reg literally 'OFFD4h',
   daal_count_reg literally 'OFFD0h',
   daal_dest_H_reg literally 'OFFD6h',
   daal_dest_L_reg literally 'OFFD4h',
   daal_src_H_reg literally 'OFFD2h',
   daal_src_L_reg literally 'OFFD0h';

6 1 Declare
   t2mode literally 'OC003h',
   t1mode literally 'OC005h',
   t2mode literally 'OC001h', /* enj/ininct true */
   clmode literally '014F7h';

/----------------------------------- set up for PACS chip select --------------- */
Declare
MCS_SIZE literally '008Fh', /* MCS size is 00-programmed only to set */
/* peripheral map to I/O space & AI-AE not */
/* latched */
PACS_LOC literally '0130h'; /* peripheral chip select base addr = 1000h */
/* one wait state for PACS 0-3; 3 wait for 4-6 */
/* external ready ignored for all PACS */

/------------------------------------------------------------------
end of equates ---
------------------------------------------------------------------/

inith6: PROCEDURE PUBLIC;

outword(MPCS_REG) = MCS_SIZE; /* init midrange block size */
outword(PACS_REG) = PACS_LOC; /* init peripheral chip select */

/------------------------------------------------------------------
set up time 2 for RAM refresh ------
------------------------------------------------------------------/

outword(t2mode_REG) = t2mode; /* timer 2 set for RAM refresh */
outword(t2maxA_REG) = 32D; /* timer 2 set for 16 u/s rate */
outword(t2count_REG) = 00h;

/------------------------------------------------------------------
set up time 1 for counting 200us -----------------
------------------------------------------------------------------/

outword(t1mode_int) = 00001;
outword(t1mode_REG) = t1mode;
outword(t1maxA_REG) = 09h; /* 25 counts */
outword(t1maxB_REG) = 00000h;
outword(t1count_REG) = 00h;

/------------------------------------------------------------------
set up DNA CHANNEL 1 for RAM refresh -----
------------------------------------------------------------------/

DNA is programmed for a memory-to-I/O transfer, with writes to nonexistent I/O space. The DNA is programmed to continuously run through the entire megabyte of memory at word transfers */

outword(dena_dest_M_reg) = 0FFh;
outword(dena_dest_L_reg) = 0FFh;
outword(dena_src_M_reg) = 00h;
outword(dena_src_L_reg) = 00h;
outword(dena_ctrl_reg) = t1mode; /* dest, mem, no inc/dec */
/* source. I/O, inc. off-tc-int */
/* SYN-unused; on-P-TDRQ-CHG-ST-word */

end; /* INIT6 PROC */

/**********************************************************
 SETUP PROCEDURE OF IN-3861 DUART UTILITY
**********************************************************/

inith: PROCEDURE PUBLIC;

init_2618:
DECLARE TEST BYTE;

DECLARE
     INT_MASK literally 'OFF28h', /* 80186 interrupt */
     INTO_CONTROL literally 'OFF38h';

DECLARE
     MODE_REG_O literally '1000h', /* MR1A,MR2A */
     CLOCK_SEL_O literally '1002h', /* CSRA write */
     STATUS_O literally '1002h', /* SRA read */
     COMMAND_O literally '1004h', /* CRA write */
     T1_DATA_O literally '1006h', /* THRRA write */
     RT_DATA_O literally '1006h', /* THRRA read */
     AUX_CONTROL_O literally '1008h', /* ACR write */
     INTERRUPT_O literally '100Ah', /* IRQ write */
     COUNT_UPPER_O literally '100Ch', /* CTUR write */
     COUNT_LOWER_O literally '100Eh', /* CTLR write */
     MODE_REG_1 literally '1010h', /* MR1B,MR2B */
     CLOCK_SEL_1 literally '1012h', /* CSRB write */
     STATUS_1 literally '1012h', /* SRB read */
     COMMAND_1 literally '1014h', /* CRB write */
     T1_DATA_1 literally '1016h', /* THRBD write */
     RT_DATA_1 literally '1016h', /* THRBD read */
     COUNT_CONTROL literally '101Ah', /* OPCR write */
     OUT_RESET literally '101Ch'; /* reset write */

/* Current setup is channel a: even parity: 8 data bits: normal op: RTS on CTS on: 2 stop bit: baud rate is 9,600 */

DECLARE
     SET_MASK literally '00h', /* enable 186 int-0 and TMR */
     MODE literally '83h', /* even parity 8 data bits */
     MODE2 literally '3Fh', /* normal TxRTS CTS en 2 STOP */
     BOUND literally '00Bh', /* 9600 baud rate */
     AUX CR literally '04h', /* disable the aux control */
     IMR literally '2h', /* enable RxRDY int reg */
     OUTPUT_PORT literally '04Ah', /* Output Port Configuration Reg# */
     UPPER_COUNT literally '00h', /* dived 8.0 Mhz by 8 = 125 kHz */
     LOWER_COUNT literally '04h'; /* dived 8.0 Mhz by 8 = 125 kHz */

DECLARE TRUE literally 'OFFh', FALSE literally '0';

DO:

ENABLE;

OUTPUT(COMMAND_O)= 1010h;
OUTPUT(MODE_REG_O)=MODE1;
OUTPUT(MODE_REG_O)=MODE2;
OUTPUT(CLOCK_SEL_O)=BOUND;
OUTPUT(INTERRUPT_O)=IMR; /* interrupt mask regester */
OUTPUT(COMMAND_O)=15h; /* ENABLE Rx and TI */
OUTPUT(OUT_RESET)=01h; /* RESET OUT */
OUTPUT(INT_MASK)= SET MASK;
OUTPUT(INTO_CONTROL)=10h; /* LTM,PRO: Highest: Cl:direct, MSK:none */
OUTPUT(COMMAND_1)= 1010h;
OUTPUT(MODE_REG_1)=MODE1;
OUTPUT(MODE_REG_1)=MODE2;
OUTPUT(CLOCK_SEL_1)=BOUND;
It ENABLE Rx and TX

* /

48 3

FLAG = 0;  // set char flag to no character * /

* /

---------------------  SET UP FOR 16MHz CONTROL REG  ---------------------  * /

49 3

OUTPUT(01200h) = 0800h

* /

---------------------  SET UP DUART TIMER  ---------------------  * /

* /

DUART Timer set to output 125kHz this is achieved by taking the 8MHz
system clock and dividing it by 8 for a 1 MHz clock into the DUART timer
then it is divided by 8 for an output of 125kHz * /

50 3

OUTPUT(AUX_CONTROL_0)=040h;  // MODE SET FOR EXTERNAL (IP2) * /

51 3

OUTPUT(COUNT_UPPER_0)= UPPER_COUNT;  // UPPER_COUNT * /

52 3

OUTPUT(COUNT_LOWER_0)= LOWER_COUNT;  // LOWER_COUNT * 2 = 8 * /

* /

---------------------  SET UP ---------------------  * /

53 3

END;  // SETUP * /

54 2

END INIT_2618;  // init 2618 duart * /

55 1

END INT186;
END GETCH;

PROCEDURE MAINROUTINE;

DECLARE SET_PORT ADDRESS;
DECLARE CONTROL_LATCH literally '1302h';
DECLARE (CONTROL_BYTE, PWM_BYTE) WORD; /* BYTE*/

DECLARE CONTROL_BYTE 7-6-5-4-3-2-1-0
0-0-0 Dead time (0)
2-2-2 Not Used
0-0-0 Lock Bit (off)
0-0-0 Divide Bit (no division)
1-1-1 Resolution Bit (8-bit)
1-1-1 Stop Bit (output enabled)

SET_PORT = 0;
CONTROL_BYTE = 0c0h; /* set the control byte */

SET_PORT = CONTROL_LATCH; /* Set address of control port */
OUTPUT (SET_PORT) = CONTROL_BYTE; /* output control byte */
COUNT1 = 0; /* clear counter-0 to zero */
COUNT2 = 11; /* set counter-1 to 1 which brings the cs in line */

END SETUP_PWM;

PROCEDURE INTERRNT 18 PUBLIC;

DECLARE PWLATCH literally '1300h'; /* address of pwm latch */
DECLARE REGIREGISTER literally '0F22h'; /* End Of Interrupt resets IS*/

DECLARE PWM_DATA WORD;
DECLARE HIGH WORD DATA (0200h); /* set bit 9 to one */
DECLARE LOW WORD DATA (0); /* set bit 9 to zero */
DECLARE LEVEL WORD;

DECLARE PWM_TABLE (#) WORD DATA
000h, 00Ah, 01h, 027h, 033h, 038h, 044h, 049h, 055h,
059h, 065h, 06Ch, 073h, 079h, 07Fh, 084h, 089h, 08Fh,
092h, 093h, 099h, 0A5h, 0A9h, 093h, 092h, 08Fh, 08Ch, 088h,
08Ah, 07Fh, 073h, 06Ch, 065h, 059h, 055h, 049h, 044h,
038h, 033h, 027h, 01h, 00Ah, 000h.;
LEVEL = 00;

IF COUNT1 = 47 THEN /* is counter-1 at senter of 53 hz */
COUNT1 = 01;

IF COUNT2 = 47 THEN /* is counter-2 at senter of 53 hz */
LEVEL = HIGH;

THEN set bit 9 high /*
ELSE /* else counter-2 is set to low side */
LEVEL = LOW; /* of 54hz set bit 9 to low */

IF COUNT2 = 94 THEN /* reset counter-2 at end of 53Hz cycle */
COUNT2 = 01;

PWM_TABLE = LEVEL OR PWM_TABLE (COUNT1); /* add level and data */
OUTWORD (PWM_LATCH) = PWM_TABLE; /* output data and level */
COUNT1 = COUNT1 + 1; /* update counter 1 */
COUNT2 = COUNT2 + 1; /* update counter 2 */
OUTWORD (EDI_REGISTER) = 8000h; /* End Of interrupt resets */

END UPDATE_PWM_INT;

PEND STOP: PROCEDURE PUBLIC;
HALT;
END STOP;

PEND MAIN:
DO:
HALT;
END:
END MAIN_ROUTINE;