This document has been approved for public release and sale; its distribution is unlimited.
**REPORT DOCUMENTATION PAGE**

<table>
<thead>
<tr>
<th>1. REPORT NUMBER</th>
<th>Marine Corps 001M</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. GOVT ACCESSION NO.</td>
<td>001MC AD-A 403 785</td>
</tr>
<tr>
<td>3. RECIPIENT'S CATALOG NUMBER</td>
<td></td>
</tr>
<tr>
<td>4. TITLE (and Subtitle)</td>
<td>Universal Infantry Weapons Trainer (UIWT) - Final Report M-16 Version Volume I: M-16 Hi-1e Mode</td>
</tr>
<tr>
<td>5. AUTHOR(S)</td>
<td>Albert Marshall, Bon Shaw, Herbert Towle, George Sitarus, Tom Riordan</td>
</tr>
<tr>
<td>6. PERFORMING ORGANIZATION NAME AND ADDRESS</td>
<td>Naval Training Equipment Center Orlando, Florida 32813</td>
</tr>
<tr>
<td>7. CONTROLLING OFFICE NAME AND ADDRESS</td>
<td></td>
</tr>
<tr>
<td>8. MONITORING AGENCY NAME &amp; ADDRESS (IF different from Controlling Office)</td>
<td></td>
</tr>
<tr>
<td>9. SECURITY CLASS. (of this report)</td>
<td>Unclassified</td>
</tr>
<tr>
<td>10. SECURITY CLASS. (of this report)</td>
<td></td>
</tr>
<tr>
<td>11. DISTRIBUTION STATEMENT (of this report)</td>
<td>Approved for public release; distribution unlimited.</td>
</tr>
<tr>
<td>12. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)</td>
<td></td>
</tr>
<tr>
<td>13. NUMBER OF PAGES</td>
<td></td>
</tr>
<tr>
<td>14. ABSTRACT (Continues on reverse side if necessary and identify by block number)</td>
<td>The Universal Infantry Weapons Trainer (UIWT) is an electro-optic based, micro-computer controlled, training device that enables tactical infantry weapons training with a M-16 rifle, under a simulated high stress battlefield environment. The battlefield is simulated using a 16mm movie. A receiver on the weapon is used to detect kill, near miss or miss information. The following weapon effects and feedback are provided to the trainees or instructor: weapon recoil, weapon bang, magazine action, automatic or single shot, and lead and elevation if...</td>
</tr>
</tbody>
</table>
applicable, audio scoring feedback using a computer generated voice, reaction time measurement, movement of weapon and a hard copy score.
SUMMARY

The Universal Infantry Weapon Trainer (UIWT), is an electro-optic based, microcomputer controlled, training device that enables tactical infantry weapons training with an M-16 rifle and 16mm motion picture projectors which simulate a high stress battlefield environment. In a short period of time a trainee can be subjected to a large variety of combat situations where each trainee's performance is analyzed in real-time and immediate feedback is given to both the trainees and instructor. Combat scenarios can be changed to fit any potential battlefield requirement. Prototype models were constructed by the Research and Technology Department, NTEC, Orlando, Florida for both the U.S. Marine Corps and PM TRADE. These models were successfully tested at Camp Lejeune, North Carolina by the U.S. Marine Corps and by the U.S. Army Infantry Board (USAIB) for the Directorate of Training Developments, U.S. Army Infantry School, Fort Benning, Georgia. It was stated that the tests did give some evidence of the UIWT system's potential for training transfer (Ref. 9). Furthermore, enlisted men, snipers and a variety of General, Field and Company grade officers who fired and observed the UIWT stated that it was a valuable training tool (Ref. 8).

U.S. Marine Corps and PM TRADE sponsored work is continuing on this program to develop the capability to add other weapons i.e., Dragon, LAW, M-60 machine gun, etc.

The authors wish to thank the Acquisition Sponsors Project Officer, Major E. Hutchinson and Lt. Col. R. F. Zumbado, formerly Assistant Marine Corps Liaison Officer, for their aid and valuable suggestions.
# TABLE OF CONTENTS

I. Introduction .................................................. 7

II. System Description ........................................... 9

III. System Design ................................................ 9
   A. Projectors .................................................. 9
   B. Rifle Electronics ......................................... 11
   C. Computer Voice and Audio Design ....................... 16
   D. Bang and Recoil System .................................... 19
   E. Distribution of Fire and Weapon Movement Monitoring ... 22
   F. Rifle Mockup ............................................... 26
   G. Microcomputer Control System ............................ 26
      1. Single Board Computer ................................. 27
      2. The Interface Board ................................... 28
      3. 8080 Program ........................................... 61
      4. Score Display and Worst Performance ................. 64
      5. Self Check .............................................. 64
   H. Film Animation ............................................ 66

IV. Conclusions .................................................. 68

Appendix
   A. Functional Description of Console Switches ............ A1
   B. Test Equipment ........................................... B1
   C. System Program .......................................... C1
   D. UPI-41 Program ........................................... D1
   E. Self Check Program ....................................... E1

ii
<table>
<thead>
<tr>
<th>Illustration</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-1</td>
<td>Artists Concept</td>
<td>1</td>
</tr>
<tr>
<td>II-1</td>
<td>System Block Diagram</td>
<td>3</td>
</tr>
<tr>
<td>II-2</td>
<td>Rifle Electronics Block Diagram</td>
<td>4</td>
</tr>
<tr>
<td>II-3</td>
<td>M-16 Training Rifle</td>
<td>6</td>
</tr>
<tr>
<td>II-4</td>
<td>Instructor's Console</td>
<td>6</td>
</tr>
<tr>
<td>II-5</td>
<td>Trainees Firing at Screen</td>
<td>7</td>
</tr>
<tr>
<td>II-6</td>
<td>Synchronized Visual and IR Projectors</td>
<td>7</td>
</tr>
<tr>
<td>III-1</td>
<td>Transmittance of Hot and Cold Mirrors</td>
<td>9</td>
</tr>
<tr>
<td>III-2</td>
<td>Target Present Decoder</td>
<td>10</td>
</tr>
<tr>
<td>III-3</td>
<td>Photo Detector Spectral Response and Geometry</td>
<td>11</td>
</tr>
<tr>
<td>III-4</td>
<td>Pre-Amplifier - Rifle Electronics - Board #1</td>
<td>12</td>
</tr>
<tr>
<td>III-5</td>
<td>Bi-Quad Active Filter</td>
<td>13</td>
</tr>
<tr>
<td>III-6</td>
<td>Universal Active Filters and Voltage Comparators - Board #2</td>
<td>14</td>
</tr>
<tr>
<td>III-7</td>
<td>NAND Gates and J-K Flip Flops - Board #3</td>
<td>15</td>
</tr>
<tr>
<td>III-8</td>
<td>One-Shot and Counters - Board #4</td>
<td>17</td>
</tr>
<tr>
<td>III-9</td>
<td>NAND Gates, One-Shot and Timer - Board #5</td>
<td>18</td>
</tr>
<tr>
<td>III-10</td>
<td>Audio Amplifier - Board #6</td>
<td>20</td>
</tr>
<tr>
<td>III-11</td>
<td>Bang Simulation Generator - Board #8</td>
<td>21</td>
</tr>
<tr>
<td>III-12</td>
<td>Recoil Circuit - Board #7</td>
<td>23</td>
</tr>
<tr>
<td>III-13</td>
<td>Laser Signal Generation - Board #10</td>
<td>24</td>
</tr>
<tr>
<td>III-14</td>
<td>Laser Pulser</td>
<td>25</td>
</tr>
<tr>
<td>III-15</td>
<td>80/20-4 Main and Interface Boards</td>
<td>29</td>
</tr>
<tr>
<td>III-16</td>
<td>Interface Board Layout</td>
<td>30</td>
</tr>
<tr>
<td>III-17</td>
<td>Interface Board Schematic (1 of 3)</td>
<td>31</td>
</tr>
<tr>
<td>III-18</td>
<td>Interface Board Schematic (2 of 3)</td>
<td>32</td>
</tr>
</tbody>
</table>
TABLES

III-1  Register Bank 1 Map ........................................... 56
III-2  Register Bank 0 Map ........................................... 58
SECTION I
INTRODUCTION

The Universal Infantry Weapons Trainer (UIWT) is an electro-optic based, microcomputer controlled, training device that enables tactical infantry weapons training with an M-16 rifle, under a simulated high stress battlefield environment. In a short period of time a trainee can be subjected to a large variety of combat situations where each trainee's performance is analyzed in real-time, and immediate feedback is given to both the trainees and instructor. Combat scenarios can be changed to fit any potential battlefield requirement. An artist's concept of the trainer is shown in Figure I-1.

Figure I-1. Artist's Concept

This training device provides the trainees or instructor the following simulated weapons effects and feedback information:

- Weapon recoil
- Weapon bang
- Magazine action
- Automatic or single shot simulation
- Lead and elevation if applicable, is programmed in the system
- Real-time individual audio scoring feedback, using computer generated voice, via a headset
- Trainee feedback data displayed in columns on TV type monitor for instructor observation
- Reaction time
- Movement of weapon relative to correct kill zone is observed by instructor and recorded for playback.
- Lowest performer indicated to instructor
- Identification of trainee responsible for shooting with no target present
- Built-in self-check features
- Score determined
- Hardcopy of scoring results
SECTION II
SYSTEM DESCRIPTION

This section of the report describes the system. Details of the system design are included in Section III.

The system utilizes two motion picture projectors: a visual and an infrared (IR) target spot projector (see Figure II-1). The visual projector displays the battle scene including the visual targets. The infrared projector provides invisible infrared target areas at which the weapon must be aimed in order to score a hit. Lead is programmed into the infrared target film, which the weapon receiver detects, requiring the trainee to lead the target as necessary. Figure II-1 shows the visual target on the left and the infrared target on the right indicating that the target is moving to the right.

Each trainee has a simulated M-16 rifle with an attached infrared (IR) receiver. The IR detector is a four-quadrant photodiode. The four-quadrant target information and microcomputer logic determines kills, eight areas of near misses, and total misses. The regions of near miss include high, low, left, right, high right, high left, low left, and low right.

Figure II-1. System Block Diagram
When the trainee fires the weapon he hears a simulated bang and feels a recoil. Recoil is generated by a short pulse of air released near the front sight which drives the weapon high and to the right. An 8080 based microcomputer determines where the round would have hit using the detector's quadrant data and supplies this information to both a computer generated voice unit and a CRT display on the instructor's console. The computer voice unit drives both the trainee and instructor headsets. When a target appears on the screen, the IR projector outputs a target present signal from the magnetic audio stripe on the film. This signal starts a clock in the microcomputer which measures the time until the trainee fires, or effectively his reaction time. The target present signal is also used to determine the number of targets that appeared, targets ignored, targets shot at, and if the trainee fired when no target was present. Trainee results are continuously displayed in columns on a CRT display on the instructor's station. At the completion of the exercise, the results, analyzes and response time are printed by a terminal at the instructor's station.

Distribution of fire can be monitored using a gallium arsenide laser infrared source located in the flash hider part of the rifle. The projected IR laser spot is invisible to the trainee but is detected by an infrared television camera and displayed by a CRT located on the instructor's console as shown in Figure II-1. When the rifle is fired the IR spot projector illuminates the screen with a small IR spot. If the instructor wants to continuously monitor rifle motion the IR aiming spot is left on continuously and the laser spot brightens when the trainee shoots. The TV camera data can also be recorded for playback during debrief.

Figure II-2 shows the rifle electronics and two projected targets. Discrimination of the infrared targets is enhanced by projecting the IR targets at frequencies different from the visual scene signals and amplifying the infrared targets. The motion picture projectors have also been modified to incorporate hot and cold mirrors, whose function will be described.

![Figure II-2. Rifle Electronics Block Diagram](image-url)
The visual projector contains a hot mirror. This multilayer dielectric mirror reflects or removes most of the infrared above 750 nanometer from the visual scene. The infrared projector contains a cold mirror. The cold mirror reflects the visual energy and passes the infrared energy above 750 nanometers. This allows a weapon equipped with an infrared receiver to ignore the visual data and obtain its target data from the infrared projector.

The S/N ratio of the system is further improved by using two different projector chopper frequencies. In the visual projector the chopper is a two bladed equally divided shutter. In the IR projector the chopper is a four bladed shutter. The visual scene is chopped or shuttered at a frequency of 48 Hz; the IR data is shuttered at a frequency of 96 Hz. By using two different chopping frequencies active filters in the weapons IR receiver can be tuned to detect the infrared target spot and ignore the visual battle scene. The projectors are frame locked together synchronously.

The rifle uses an IR detector consisting of a lens and a four-quadrant photodiode detector to detect infrared targets. An infrared filter is utilized in the weapon optical system to reduce the visual signal effect on the photo detector. The photo detector signals are amplified by two bi-FET operational amplifiers. A voltage comparator sets a threshold to establish a digital "one" or "zero". The voltage reference level of the comparator can be set to adjust the level of difficulty. The voltage comparator data is latched and delivered as input to the microcomputer system for data analysis, display and feedback.

The rifle can operate in either a single-shot or automatic mode and requires the trainee to reload after he has fired thirty rounds. The rifle's simulated magazine contains a capacitor. When the magazine is inserted into the rifle this internal capacitor is discharged, which resets a counter.

Bang simulation is achieved by filtering a noise source and then producing a noise envelope with a sharp rise time and exponential decay.

The training rifle is shown in Figure 11-3. The four-quadrant detector is located on top of the barrel and the flash hider contains a gallium arsenide IR laser. The rifle is a replica but contains real sights that are adjustable. The plastic hose shown attached to the rifle, Figure 11-3, is used to carry the air for recoil.

The instructor's console is shown in Figure 11-4. The right hand CRT displays the verbal data transmitted to each trainee in four columns. The lowest score is automatically flagged by a LED under the applicable trainee's column. This alerts the instructor so he can more closely observe that trainee. The left hand side of the console contains a CRT display used to monitor the weapon motion. Communication to the microcomputer is via a terminal shown in front of the instructor. See Appendix E for description of the switches on the console.
Figure II-3. M-16 Training Rifle

Figure II-4. Instructor's Console
Figure 11-5. Trainees Firing at Screen

Figure 11-6. Synchronized Visual and IR Projectors
Figure 11-5 shows the trainees firing at the screen. Note each trainee wears a headset for individual feedback.

Figure 11-6 shows the projectors. Loopers (a closed-loop film strip) are used so rewinding is not necessary. An auto-stop/auto-align feature is visible near the loopers.

The computer voice system is a solid state communications processor. It operates as a standard data terminal to the host 80/20 microcomputer system. The vocabulary has been digitized and stored in nonvolatile memory (PROM). The system contains thirty-two individually addressable words and five independent output channels. Thus, the computer voice system can talk to any or all of the five trainees while saying the same or different words or phrases. Each trainee wears a headset so he hears only the feedback applicable to his performance.

The system is controlled by a modified Intel 80/20 microcomputer system.

Section III, next, describes the system design.
SECTION III
SYSTEM DESIGN

A. PROJECTORS

The motion picture projectors are two Hokuskin, 16mm sound projectors equipped for frame-for-frame sync. The lamp is a 500 watt Xenon-arc, type KXI-500H. One projector is used as an IR target spot projector. The IR projector uses a cold mirror to remove the visual energy, Melles Griot, O3MG5D07. The transmittance of the hot and cold mirrors are shown in Figure III-1.

Loopers are utilized instead of reels to eliminate the necessity of rewinding the film.

Figure III-1. Transmittance of Hot and Cold Mirrors

The projectors are equipped for either optical or magnetic sound reproduction. Sound for the battle scene is recorded for optical pickup on the visual projector.

Target present signals are recorded on the magnetic stripe of the IR film. The target present signal is a 1 KHz audio tone, which is decoded by an electronic tone decoder, Figure III-2.

The battle scene film was both taken and projected using a 25mm focal length lens to minimize perspective distortion.

The IR projector has a modified four bladed shutter which chops the IR data at a frequency of 96 Hz. The visual projector has a conventional two bladed chopper which chops the visual scene at 48 Hz.
The projectors are equipped with an auto-stop feature which allows the film to be stopped at any desired location by simply placing a foil metal strip on the desired stop location.

The screen is silver matte, 9 ft x 12 ft overall.

B. RIFLE ELECTRONICS

The rifle electronics detect the IR target spot, amplifies, discriminates and provides digital data to the 8080 based microcomputer.

The detector optics is a single element double convex lens, with a diameter of 29 mm and focal length of 114 mm.

The IR detector is a four-quadrant silicon photodiode. This device consists of four discrete elements on a single substrate with an active output lead from each element. When the weapon is aimed properly the infrared target spot is centered on the detector and the output current from each quadrant is equal. As the rifle is moved the currents change as a function of the location of the infrared target spot on the detector. Imbalance in the current indicates off-center position. The detector has an active area of 0.05" x 0.05" per element with a gap of 0.005" between elements. The detector physical geometry and spectral response is shown in Figure III-3.

![Figure III-3. Photo Detector Spectral Response and Geometry](image)

The field of view of the IR detector is approximately seven inches on the screen.

The currents from the diode are input to an operational amplifier, TL082. The photodiode detector is basically a current source with an output impedance which is very large. The first stage of the current-to-voltage converter presents almost zero load impedance to ground because the inverting input appears as a virtual ground. The input current from the diode flows through the two Megohm feedback resistor, generating an output voltage.
\[ \text{voltage} = \frac{i_d}{R_f} \]

where

\[ R_f = 2 \text{ Megohm} \]
\[ i_d = \text{detector current} \]

A separate channel is used for each of the four quadrants. The output from the current-to-voltage amplifier goes to a noninverting amplifier with a gain of 23. This stage is also part of the TLO82. The electronics described above is located on Board #1, Pre-Amp. (Figure III-4)

Input signals to the active filter are 48 Hz from the visual scene, 96 Hz from the IR target spot and any extraneous light. The active filter is used to pass and amplify the desired IR signal at 96 Hz and reject all other signals.

The UAF - 41 is a two pole active filter. It uses three operational amplifiers in a double integrator feedback loop to generate two conjugate poles. Location of the poles in the complex plane, and thus the natural frequency and Q are determined by external resistors.

The equivalent configuration of this band pass filter is shown in Figure III-5. The filter is designed for a 96 Hz center frequency with both a Q and gain of 50.

Figure III-5. Bi-Quad Active Filter

Both the active filters and voltage comparators are located on Board #2, Figure III-c. The output of the active filter is a sine wave with a frequency of 96 Hz. The output sine wave goes positive and negative about a zero volt reference level. This output is clamped and fed to a voltage comparator. The voltage comparator changes the analog detector signal to a digital signal. The input signal level for a one or zero is determined
Figure III-7. NAND Gates and J-K Flip Flops - Board #3
by a resistor or reference voltage setting. Each of the four voltage comparator channels has its own reference voltage setting resistor, i.e., R33, R34, R35, and R36. The reference voltage setting controls the degree of difficulty in hitting a target. The detector signals next go to IC1, a 7408 AND gate, Board #3, Figure III-7. If the trainee pulls the rifle trigger and has rounds remaining in his magazine, the NAND gate is enabled by an input from Board #4. Board #4 is shown in Figure III-8. IC2 and IC3, Board #3 are 9024 JK flip flops configured as latches. Each 9024 has two latches. The 9024 is reset by the microprocessor after it has accepted the four-quadrant IR target spot data. IC4, Board #3 is a line driver.

Board #5, Figure III-9, is connected to the rifle trigger. IC1, a 5437, containing NAND gates, debounces the trigger and applies 5 volts to IC3. IC3, a timer, provides pulses of 12 Hz, which is the firing rate of the weapon. A one shot is also triggered and provides a single pulse. The output of Board #5 is determined by the setting of the single or auto fire switch on the simulated weapon. The setting of auto or single shot determines which gate on IC1 is active. If the trainee is in auto fire pulses at 12 Hz are provided Board #2. IC3 on Board #4 has a gate which will pass the signal if the counters IC1, IC2 on Board #4 indicate rounds are left. The counter enables IC1 on Board #3 and also enables the data ready pulse provided by IC4 on Board #4 to the microprocessor. IC4 is a one shot which generates a 10 usec data ready pulse for the microprocessor to indicate data is available. After the microprocessor has read the data it resets the latches; IC2 on Board #3.

The one shot IC6, Board #4, Figure III-8, is used to reset the counters. The dummy magazine contains a capacitor. In the "loaded" configuration the capacitor is charged to 5 volts. The magazines are easily loaded or charged by momentarily inserting them into a charging fixture.

When the dummy magazine or capacitor is inserted in the rifle it discharges through R4, providing the counter reset voltage. The magazine is reloaded by charging the capacitor in the magazine to 5 volts.

C. COMPUTER VOICE AND AUDIO SYSTEM

The Computer Voice System is a Business Communicator Model LVM-70 manufactured by VJTRAX, the Vocal Interface Division of Federal Screw Works, Troy, Michigan. The LVM-70 was designed specifically to be used as a concentrator for touch-tone based information systems.

Up to 32 words (16 seconds) are available with up to eight audio output channels. The trainer utilizes an output channel for each trainee. When a shot is fired by a trainee the host computer (80/20) decodes the incoming rifle data and then sends three bytes of serial data to the LVM-70 specifying a start word, a trainee identification word, and the appropriate voice response code. The voice output line for each trainee is routed to the trainee's audio mixer/amplifier, Board #6, Figure III-10, as well as the instructor's control panel.

The LVM-70 voice communicator can be replaced in later models for roughly 1/4 the original cost, due to technological advances.
Each trainee's audio system consists of two stages of audio amplification. A Texas Instrument's TL074 low noise, quad, dual operational amplifier, IC1, is used. Consequently, two trainees are handled by a single TL074 (Figure III-10). The first stage of amplification is primarily an audio mixer. Five independent channels are mixed into one. These five channels consist of the computer voice feedback system, the instructor communication line, the synthetic rifle bang, coordinated battlefield sounds and general battlefield environment sounds. The instructor uses an identical mixer/amplifier channel but his inputs consist of the various computer voice responses to the trainees. The instructor selects which trainee he desires to hear by pushing the appropriate switch on the instructor control panel.

Each of the five inputs to the mixer stage as well as the final output stage have their own volume control.

D. BANG AND RECOIL SYSTEM

1. BANG SYSTEM

An electronic bang is presented to the trainee via his headset when he has fired a shot. The bang board, Board #8, Figure III-11, produces the synthetic gunshot sound and passes this sound to the trainees audio mixer/amplifier Board #6, Figure III-10. The bang is produced by generating random noise, due to diode D1 being biased near its breakdown voltage, and then using the FET to generate an envelope for this random noise. This envelope consists of a sharp rise time and an exponential decay which corresponds closely to a gun shot noise envelope. Specifically, the diode D1 produces random noise which is amplified by 1/2 of IC1, a dual operational amplifier. This amplified random noise is presented to the drain of the FET. The FET does not pass this noise until its gate is presented the sharp rise and exponential decay envelop representing an actual rifle shot sound envelope. The sharp rise of voltage on the gate of the FET is produced by IC2 changing to a high state; 5 volts. When IC2 changes back to a low state, 0 volts, the diode D2 isolates the gate of the FET from being pulled down to an off state and allows the RC network consisting of R6, R12, and C7 to exponentially decay the residual voltage thus producing decaying gunshot envelope of noise. The source of the FET thus produces on demand random filtered noise within an envelope resembling a gunshot bang. The second half of IC1, an operational amplifier, produces final amplification of this sound before passing the output to the students audio mixer/amplifier.

2. RECOIL SYSTEM

The recoil system consists of three major parts: air hose, recoil board, and the air valve.

The air hose follows the electrical cable up to the rifle and into the butt of the weapon. The hose is a lightweight, nylon reinforced, dimensionally stable air line hose. After entering the butt of the rifle it runs forward and attaches to the rifle barrel. The barrel is plugged at the tip end and an outlet orifice has been drilled on the bottom of the barrel near the tip end. The orifice is pointing down and 30 degrees to the left which produces a thrust up and to the right when a shot is fired.
Figure III-10. Audio Amplifier - Board #6
Figure III-11. Bang Simulation Generator - Board #8
The recoil circuit, Board #7, consists of a 555 integrated circuit timer, IC2, and a darlington pair transistor driver circuit for the recoil air valve. The 555 timer is set for a nominal 20-25 msec duration. The variable resistor R2 serves to regulate the timing duration (Figure III-12).

The recoil valve is a pilot operated solenoid valve. Because it is pilot operated, the on-off rise and fall times for actuation are very short and power consumption is only 8.5 watts.

E. DISTRIBUTION OF FIRE AND WEAPON MOVEMENT MONITORING

Distribution of fire and weapon movement can be monitored and recorded during a training exercise for playback. The system allows the instructor to view where the weapon is aimed relative to the IR target spot. This feature is completely independent of the basic system.

An IR light source is used on the weapon. The infrared light source used in the system is a semiconductor, gallium arsenide laser. The laser is collimated by a simple plano convex lens. The laser is attached where the weapon flash hider is located. If the instructor wishes to view the location of the trainee's weapon, he selects the laser he wants turned on and holds down a button on the instructor's console. The instructor is able to view both the projector IR target and laser spot from the selected trainee's rifle. This information is detected using an RCA TC 1005/H01 low bloom silicon target Vidicon and closed circuit video equipment. The TV display tube is located in the instructor's console and the TV camera near the motion picture projectors.

The laser spot brightness seen on the TV is a function of the pulse repetition frequency (prf) of the gallium arsenide laser. Two modes are available:

- Flash only
- Track plus flash

In the flash mode only, a single flash occurs when the trainee fires. In the track and flash mode, the instructor sees a point of laser light on the screen all the time, which moves as a function of where the trainee is pointing; when the trainee fires, a brighter flash occurs.

Laser energy reflected off the screen is eye safe. However, the trainee should not point his weapon in another trainee's eyes as eye damage can occur from looking directly into the laser beam.

The laser timing signals are generated using Board #10, Figure III-13. The laser pulser, Board #11, is shown in Figure III-14.

The laser pulser uses a SCR, GA201 to discharge capacitor C1. Q1 is used to allow rapid recharge of Q1. The laser is a 5 watt peak power laser with a nominal 50 nanosecond pulse width.
Figure III-14. Laser Pulser
F. RIFLE MOCKUP

The rifle mockup is manufactured by Replica Models, Inc. It is not designed to accept a round of ammunition and cannot be converted to accept ammunition. The original replicas received from Replica Models have been extensively modified to perform satisfactorily as a trainer. The original barrel plug was removed and moved to the front end of the barrel to accommodate the recoil. A recoil orifice was then machined and an electronic board was installed within the handguards. To accommodate boresighting, the original molded-on nonadjustable front and rear sights were replaced with adjustable front and rear sights. The mode selector switch was modified to reflect the real M16El mode positions; the trigger mechanism was modified for better performance; microswitches for the trigger and mode selector switch were installed; and magazine sensing contacts were installed for reloading simulation.

An optical four-quadrant detector and optics are mounted above the barrel and below the sights. A solid state laser and optics for point of aim information has been inserted in the flash hider position.

Air for the recoil and electronic wiring approach the rifle from the bottom rear of the butt of the rifle. The true weight of the M16El was restored by removing unused mechanism from the upper receiver. The true balance was maintained through equal weight additions, i.e., the detector/laser combination at the front end of the rifle offset the hose and electronic wire harness at the butt end of the rifle.

Special test equipment is included in Appendix B.

G. MICROCOMPUTER CONTROL SYSTEM

The 8080 Microprocessor Based Control System performs these functions:

- Interrogates the instructor for session parameters
- Stores session parameters for final hard copy
- Determines if self-check is desired, and reacts accordingly
- Initializes peripheral LSI chips and zeros memory storage
- Inputs rifle data, decodes and stores it
- Measures response time for first rifle shot at new target for each of four or five rifles
- Outputs shot results to audio feedback and instructor's CRT
- Identifies shooter making most errors and sends the identification to the instructor's console "LEDS"
- Updates shooter's results file
- Checks for session end and terminates the data collection mode upon the instructor's signal
- Computes trainee's overall score
- Prints trainee's results on the instructor's electronic data terminal

1. SINGLE BOARD COMPUTER

The UIWT System is controlled by a modified INTEL 80/20-4 Microcomputer System, Reference 6. This microcomputer system, which is based on the INTEL 8080 microprocessor, includes an enclosure with front panel controls, power supply, cooling fans, and a card cage in which is located the main 80/20-4 board as well as the interface board (IFB), which is described below.

a. 80/20-4 MODIFICATIONS

A number of modifications are required before the SBC 80/20-4 can be used in UIWT/IVAT Version 1.2. These are detailed below with page identifications to be found in Reference 6 unless otherwise noted.

1. Pull-up resistor packs, SBC-902, page 2-5, must be inserted in socket A5 and A6 as input terminators for port 2 at address E6. These terminators were supplied with the 80/20-4 systems as Beckman part number 1899-747-0, 3000641-01.

2. Insert inverting line drivers, either #7437 or #7400, in sockets A3, A4, A9 and A10 for output ports 3 and 6 at addresses E6 and EA. See pages 2-4 and 4-4.

3. Solder a jumper between J3-8 and J3-10 using the solder points on the rear side of the board. This connects "Request to Send" to "Clear to Send". See Table 2-5, page 2-7.

4. Wire wrap a jumper from pin 1, a 5 volt source, near J3 pin #25 and solder it to a through hole just below "C9" between A15 and A16 on the front of the board. This should put 5 volts onto J3-16 "REC LINE SIG DETECT" which goes to the "DATA CARRIER DETECT" of the 743 TI terminal. Otherwise the terminal will not function. See page 27 of Reference 1.

5. Change the wire wrap jumper which exists between pins 141-142 just above A22, the 8253, to a jumper between pins 141-143. This is an option which connects the clock input for counter 1 of the 8253 to the output of counter 0. See page 4-21.

6. Interconnect wire wrap pins 11 and 12 near the upper right hand corner of the board. These are the protect and signal grounds for the TI 743 Terminal. See page 27 of Reference 1.

7. Scratch through the line going from the transmit data (TXD), pin 19 of the 8251 USART, to the SN75188 (MC1488) line driver. Connect the 8251 side to J1-50 with a jumper wire and connect the driver side to J2-50. This allows the interface board to switch the serial output between the VOTRA and the control console.
(8) Remove the jumper from wire wrap pins 52-53, located just below the left part of the leftmost 8255 and put a jumper between pins 51-52. This enables port 1 as an input. See Figure 5-2 (sheet 4 of 5). Check that a jumper exists between pins 71-72 to enable port 4 as an output.

(9) Connect the 80/20-4 front panel interrupt switch into the interrupt controller as interrupt #7. To do this, connect pins 36, 37, 38, and 39 together and also to pin #45.

(10) Make the required modifications to use 2716 2K byte EPROMS. These are given in Table 2-12, which is entitled "Jumper Changes for Optional 8K EPROM Installation". See page 2-15.

<table>
<thead>
<tr>
<th>REMOVE</th>
<th>INSTALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>W2, A-C</td>
<td>W2, A-B</td>
</tr>
<tr>
<td>W4, B-D</td>
<td>W4, A-D</td>
</tr>
<tr>
<td>W4, C-E</td>
<td>W4, B-E</td>
</tr>
<tr>
<td>W7, A-B</td>
<td>W7, A-D</td>
</tr>
<tr>
<td>W8, A-C</td>
<td>W8, A-B</td>
</tr>
<tr>
<td>C35, 53 and 72</td>
<td>Above A45 and A46</td>
</tr>
<tr>
<td></td>
<td>Above A78</td>
</tr>
<tr>
<td></td>
<td>Above A78</td>
</tr>
<tr>
<td></td>
<td>Below A79</td>
</tr>
<tr>
<td></td>
<td>Below A79</td>
</tr>
<tr>
<td></td>
<td>Above A37 and Below A64 and A79</td>
</tr>
</tbody>
</table>

2. THE INTERFACE BOARD

All input/output (I/O) operations of the SBC 80/20-4 microcomputer pass through the interface board (IFB). These operations can be divided into three categories.

- Rifle communications
- I/O through the 8751 "USART"
- Output through the UPI-41, 8741 Universal Peripheral Interface

Figure III-15 is a block diagram of these data paths and their associated control lines. More details are shown on Figure III-16 through III-19.

a. RIFLE COMMUNICATIONS

IR spot quadrant detector data are input from each rifle to a separate 8212 eight-bit input/output port chip on the IFB. A trigger-pull signal is also sent from each rifle to its associated 8212. Upon sensing a trigger signal, the quadrant data are latched into the 8212 buffer and an interrupt signal requesting service is output from the 8212 to the main board through input port 1. The service request lines from all five 8212s are "ORED" together onto a single line which also goes to port 1 to signal that at least one 8212 requires service. As long as this ORED line indicates a service need, the microcomputer polls each 8212 service request line in turn. When one is detected that needs service, the address of the 8212 responsible for the request is output from port 3 on the main board to a 9211 one-of-sixteen decoder on the IFB. A data read signal is then output from port 6 to the 9311, which commands the 8212 to place the contents of its latched buffer on the common data bus.
Figure III-15. 80/20-4 Main and Interface Boards
Figure III-17. Interface Board Schematic (1 of 3)
Each of the other four unaffected 8212 chips may also contain data, but are held temporarily in an inactive "three-state" and present a high impedance load to the bus. The only quadrant data available at the main board port 2, therefore, are from the 8212 being serviced. These data are read into memory and the data read signal to the 9311 is removed. This removes the read command to the 8212 and clears its interrupt service request. A pulse is then sent from port 6 to the 9311 which issues a reset signal from the IFB to the rifle electronics associated with the serviced 8212.

The serviced 8212 is now in three state, its service request line is off and it is ready to latch in new data upon receiving the next trigger-pull signal. In the meantime, if other 8212 chips need service as indicated by assertion of the ORED line, the computer polls the next 8212 interrupt line. If it needs service, the process is repeated; if not, the next 8212 service line is polled in sequence. This continues until the ORED service line goes off and the computer moves ahead with the remainder of the program.

A "Target Present" signal from the IR spot projector is carried directly through the interface board to the main board through input port 2. The target present information is recorded and used during scoring to identify a valid target.

b. USART I/O

The control terminal is an electronic data terminal operating at a rate of 300 bits per second, Reference 1. At the initiation of each training session, the computer connects the output serial data stream from the 8251 programmable communication interface or Universal Synchronous/Asynchronous Receiver/Transmitter (USART) to the terminal. The computer, therefore, is able to carry on a two-way conversation with the squad leader in order to obtain "initialization" data as shown on Figure 6. The computer questions the squad leader and prompts for answers by issuing the character "?".

During the actual training session, the USART output is switched to the digitized word audio system. When the session is finished, the squad leader strikes/presses the start/print button on the instructor's console and USART output is again directed to the terminal which types out hard copy scores, as also shown on Figure III-20.

c. UPI-41 MICROCOMPUTER OUTPUT

During a training session, console CRT data are output in parallel from port 6 of the 30/2C-4 single board computer to an 8741 Universal Peripheral Interface Slave Microcomputer (UPI-41) on the IFB. The UPI-41 decodes the parallel data and sends a 19,200 BAUD, 7 bit ASCII data stream to the console CRT. The console CRT translates the serial data stream into a score message and displays the message in the column reserved for the appropriate rifle. The UPI-41 also monitors the setting of 5 control switches, one for each rifle which allows the squad leader to inhibit the display of scores for any or all rifles.

34
WANT ID YES OR NO
NO
LET'S START

"INITIALIZE" PORTION OF TRAINING
SESSION

RIFLE: 1

YOUR RESULTS ARE:

TOTAL SHOTS: 99
HITS: 16
MISSES: 29
LOWS: 2
LOW RIGHTS: RIGHTS: 6
HIGH RIGHTS: 3
HIGHS: 4
HIGH LEFTS: 8
LEFTS: 22
LOW LEFTS:
NO TARGET: 9
TARGETS IGNORED: 8
TARGETS SHOT AT: 30
AVERAGE TIME: 1.2 SECONDS

OH WELL: THERE'S HOPE IF YOU SPEED UP
YOUR OVERALL SCORE IS: 37

SESSION PROPER.
NO OUTPUT TO TERMINAL. OUTPUT
IS VIA VOTRAX DIGITIZED AUDIO
WORDS & CONSOLE CRT. THIS
PHASE IS TERMINATED BY AN
INTERRUPT FROM TERMINAL.

"PRESENTATION OF RESULTS"

Figure III-20. Typical Printout Format on Terminal
(Continues for all 4 rifles)
The UPI-41 system description is divided into four parts: part 1, a functional summary and a component interface description are presented. Performance criteria are also established in this section. Part 2 describes the facilities available within the UPI-41 and explains their use in the present application. Part 3 describes the UPI-41 control program and part 4 evaluates the system with respect to assumption validity, performance criteria, and maximum system capabilities. The source program is given in Appendix D.

d. UPI-41 MICROCOMPUTER OUTPUT II

(1) System Description

The function of the intelligent controller is to receive parallel data from the SBC 80/20, decode the data, and cause a message to appear on the ADM-3A screen based upon the content of the data received. The control switch settings also affect controller operation, but only secondarily.

A block diagram showing the system component relationships appears in Figure III-21.

Figure III-21. UPI-41 Intelligent Controller System Block Diagram
The following describes the three component interfaces shown in the figures: SBC 80/20 to controller, controller’s to ADM-3A, and control switches to controller.

(a) SBC 80/20 To Controller Interface

The SBC 80/20 to controller interface is comprised of three sets of connections. The first set, consisting of 8 data lines and 1 control line, are the data transfer connections. The second set consists of the clock connections while the third set consists of only one connection, the initialization connection.

Data Transfer Connections

The 8 data lines of the data transfer set connect an 8 bit output port on the SBC 80/20 to the 8 bit Interface Register of the UPI-41. There are six I/O ports on the SBC 80/20 numbered 1 through 6 (1). These ports are divided into the Group A ports, 1-3, and the Group B ports, 4-6. Each port group corresponds to a single 8255 Programmable Peripheral Interface, PPI. Port 4 of Group B is programmed as an output port and used for the SBC 80/20 to UPI-41 data connection.

To transmit data to the UPI-41, the SBC 80/20 places data on port 4 and sends a Data-Available pulse to the UPI-41 over the control line. The Data Available pulse is software generated and is transmitted through port 3 of Group A 8255. The length of the Data Available pulse is set by the time required to execute the instructions necessary to change the logic level of the control line twice, first from high to low, then from low to high. For the SBC 80/20 this results in a 10 microsecond pulse. The maximum pulse length to the UPI-41 is set at twice the instruction cycle length, or 6.5 microseconds; therefore, the 10 microsecond Data Available pulse is sent to the one shot within the controller where it is shortened to 1 microsecond. The 1 microsecond pulse from the one shot supplies the WR input to the UPI-41. On the rising edge of this pulse the data on the SBC 80/20 output port is latched into the UPI-41 Interface Register. SBC 80/20 to controller data transfer connections are illustrated in Figure III-22.

Each byte of data transferred from the SBC 80/20 to the UPI-41 contains two kinds of information encoded into separate fields within the byte. The three most significant bits contain a source identifier encoded in straight binary, and the four least significant bits contain a message identifier, also in straight binary, see Figure III-23. Bit four is not used.

The rate of data transfer from the SBC 80/20 to the controller can be characterized by three separate data transfer rates of which the last two will be of interest. The first two rates are determined by the SBC 80/20 input configuration, Figure III-24, while the third is determined by the input configuration in combination with the SBC 80/20 data processing rate.

The SBC 80/20 input configuration consists of 5 input sources, where each source contains a data latch and a service request line. When data is latched into one of the sources, the SBC 80/20 receives a service request signal from that source. For each service request that the SBC 80/20 responds to, a data byte will be sent to the controller.
**Figure III-22.** SBC 80/20 to Controller

Most Significant Bit

<table>
<thead>
<tr>
<th>BIT7</th>
<th>UNUSED</th>
<th>BIT0</th>
</tr>
</thead>
</table>

Source Identifier

Message Identifier

**Figure I I-23.** SBC 80/20 to UPI-41 Data BYTE
Figure II -24. SBC 80/20 Input Source Configuration
The first data transfer rate is the average transfer rate and occurs when the 5 sources are initiating service requests at their normal rate. The second data transfer rate is a peak average rate, and occurs when all 5 sources are initiating service requests at their maximum rate of 12 per second. This condition results in a peak average rate of 12 x 5, or 60 transfers per second. The third data transfer rate is the maximum rate, and occurs anytime there are simultaneous service requests to the SBC 80/20. This rate is determined by the processing rate of the SBC 80/20. Analysis using: (1) real-time emulation under control of Intel's In Circuit Emulator, ICE-80, (2) tabulation of instructions executed and their execution time and (3) experimental determination, indicates that the SBC 80/20 processing rate is approximately 200 inputs per second.

As indicated before, the peak average transfer rate of 60 transfers per second, and the maximum transfer rate of 200 transfers per second are the relevant quantities characterizing the data transfer interface.

To keep up with the SBC 80/20 over extended periods, the processing rate of the UPI-41 must equal or exceed the SBC 80/20 peak average transfer rate, and to keep up with the SBC 80/20 when simultaneous service requests have occurred, the reception rate of the UPI-41 must equal or exceed the SBC 80/20 maximum transfer rate.

The requirement on the UPI-41 processing rate will be used in the sequel to determine the baud rate used in the controller to ADM-3A interface, while the requirement on the UPI-41 reception rate will be used to establish the necessity of a data queue within the UPI-41.

One final point is that there are no provisions for the UPI-41 to indicate that it is ready to accept a data transfer from the SBC 80/20. Thus, the data queue mentioned above will be filled by an interrupt driven procedure. This technique will assure that a data byte has been removed from the Interface Register before an additional data transfer can occur.

2. Clock Connections

The clock connections supply the UPI-41 clock inputs, X1 and X2. A single line from the SBC 80/20 supplies the controller with a 9.216 megahertz clock which the SBC 80/20 makes available as the BCLK output. Within the controller, the BCLK frequency is divided in half by a 7474D flip flop. This division is necessary to bring the BCLK frequency within the 1 to 6 megahertz operating range of the UPI-41. The Q and Q0 outputs of this flip flop supply the UPI-41 inputs, X1 and X2, with a 180° out of phase 4.608 megahertz clock. While the UPI-41 is capable of generating its own clock by connecting a crystal to the X1 and X2 inputs, the BCLK frequency is used since the standard asynchronous communication frequencies can be derived from it. The clock connections are shown in Figure III-25.
3. Initialization Connection

The initialization connection is between INIT output of the SBC 80/20 and the RESET input of the UPI-41. A low going pulse on this line causes the control program of the UPI-41 to begin execution at location 0.

(b) Controller to ADM-3A Interface

The controller to ADM-3A interface consists of a single line which originates from line 0 to port 1 on the UPI-41, passes through the 75188 inverting line driver, and terminates on the Receive Data, RXD, input of the ADM-3A. The line driver converts the TTL output of port 1, 0 - 5 volts, into RS-232C logic levels of ±12 volts.

Information is transmitted from the UPI-41 to the ADM-3A serially using 7 bit ASCII code under the RS-232C communication protocol. For this application, the number of bits per character has been minimized by using a single stop bit and no parity bit. For a given serial transmission rate this configuration will result in the fastest possible character transmission time. This time is an important consideration, as each parallel byte received by the controller from the SBC 80/20 will require a 22 character message to be transmitted. With the single start bit, the 9 bit serial character appears as shown in Figure III-26.
Figure III-26. Serial Transmission Character

Each data byte received by the UPI-41, except as noted in the next section causes a string of 9 bit characters to be sent from the UPI-41 to the ADM-3A, a 24 line by 80 character CRT display.

The function of the ADM-3A is to provide three kinds of information concerning the SBC 80/20 inputs to an observer. The ADM-3A displays a message, indicates the SBC 80/20 source corresponding to the message, and reflects the order of input occurrence. The message is indicated by the characters displayed on the screen. The source is indicated by dividing the ADM-3A screen into 5 columns of equal width, with the first column reserved for source 1 message, the second column for source 2 messages, and so on for the five sources. The order of inputs is indicated by scrolling the display 1 line each time a message is displayed.

For a screen width of 80 characters, and not allowing an overlap of columns, the message field for each source is limited to the integer portion of 80/5, or 16 characters. The ADM-3A screen use is illustrated in Figure III-27.

To implement the function of the ADM-3A as described above requires that 22 characters be sent to the ADM-3A for each SBC 80/20 to controller transfer. The 22 characters are sent in 3 groups: a cursor control group, a message group, and a display control group.

The first group sent, the cursor control group, contains four characters which cause the cursor to the ADM-3A to position itself at the beginning of one of the five message columns. The first two control characters "escape" and "equals", activate the ADM-3A cursor positioning logic, while the next two characters are interpreted as the X and Y coordinates of the new cursor position, respectively. The Y coordinate sent is always the same, 137H, and selects the bottom line of the display. The X coordinate is determined by the SBC 80/20 input source.

The second group sent, the message group, contains 16 characters. These characters will be printed on the screen of the ADM-3A in the message field whose beginning was established by the cursor positioning control group.
Figure III-27. ADM-3 Screen Use
The third group sent, the display control group, contains the remaining 2 characters. These characters, a carriage return and line feed, cause the display to scroll up one line in preparation for the next control group 1 sequence.

The complete 22 characters string appears as shown in Figure III-28.

![Figure III-28. Character String Transmitted to ADM-3A](image)

---

The final aspect of the controller to ADM-3A interface is the serial transmission rate to be used. Having now established (1) the number of characters sent by the controller to the ADM-3A per SBC 80/20 input, (2) the number of serial bits per character, 9, and (3) the UPI-41 processing rate requirement, 60 transfer/sec, a minimum serial transmission, or baud rate, can be computed as:

\[
\text{minimum baud rate} = 22 \text{ characters/SBC 80/20 transfer} \times 9 \text{ bits/character} \times \frac{1}{60} \text{ SBC 80/20 transfers/second} = 11,880 \text{ bits per second.}
\]

The next highest, indeed the highest, baud rate at which the ADM-3A can receive data is 19,200 baud. This value must necessarily be chosen as the data transmission rate.

(c) Control Switches to Controller Interface

The control switches to controller interface is a 5 line connection between 5 control switch outputs and the 5 least significant inputs of port 2 on the UPI-41. The design of port 2 on the UPI-41 is such that if nothing is connected to a port line, the line will read as a logic one, whereas, if the line is grounded through a 1k resistor, the port will read a logic zero. The control switch to controller connections are shown in Figure III-29.

During the processing of a data byte by the UPI-41, the binary source identifier is translated into a linear select code which is then compared with the switch setting on port 2. If the switch corresponding to the source identifier is set in the abort position, a logic 0 is present and a message will not be sent. This is the exception referred to in the controller to ADM-3A interface description. If the switch is set in the display position...
a logic true will be present and a message will be sent.

This concludes the overall system description. The next two sections will describe the principle device within the intelligent controller, the UPI-41 single chip microcomputer.

![Figure III-29. Control Switches to Controller Interface](image)

e. **UPI-41 MICROCOMPUTER OUTPUT II**

The UPI-41 single chip microcomputer provides the intelligence of the intelligent controller. The block diagram in Figure III-30 illustrates the facilities available within the UPI-41.

As described in the previous section, the interface register is used for communication with the SBC 80/20, port 1 is used for communication with the ADM-3A, and port 2 is used for communication with the control switches.

Program memory is divided into 4 pages of 256 bytes each. These pages are numbered 0 to 3. Page 0 contains the main loop of the control program, while page 1 contains the various subroutines called by the main loop. Page 3 has a special feature in that data bytes can be transferred from it to the accumulator using the current value of the accumulator as a pointer. This "table lookup" feature is used to access the message strings which are sent to the ADM-3A. 16 messages of 16 characters each are stored, using all 256 bytes within the page. Program memory configuration is shown in Figure III-31.
Figure III-30. UPI-41 Single Chip Microcomputer Block Diagram
RAM within the UPI-41 serves three purposes: it contains the registers, the subroutine and interrupt stack, and the variable data storage locations. The distribution of the 64 RAM locations between these three functions is shown in Figure III-32.

The registers in bank 0 are designated R0-R7, while those in bank 1 are designated R0'-R7'. Only one register bank at a time can be addressed. Bank selection is accomplished by executing a special select register bank X, SELRBX, instruction where X is either 0 or 1. The registers of bank 0 are used for data processing and message transmission, while those of bank 1 are used for queue control.

The UPI-41 contains a rather sophisticated timer which was evaluated for use as the bit interval generator for UPI-41 to ADM-3A serial transmission. As several difficulties were encountered, the use of the timer while representing a possible area for future research, was rejected in favor of a software timing approach. The software timing routine will be described, along with the rest of the UPI-41 control program, in the next section.
Figure II-32. UPI-41 RAM Memory Map
f. UPI-41 CONTROL PROGRAM IV

The UPI-41 program is written in MCS-48/UPI-41 assembly language. It was assembled using a cross assembler operating on an Intel Microcomputer Development System, MDS-800. The machine code was burned into the EPROM program memory of the UPI-41 using an Intel Universal Prom Programmer and the Universal Prom Mapper Software. The assembly of the program and the burning of the EPROM were done under control of the Intel System Implementation Supervisor, ISIS II, operating from an Intel Dual Floppy Disk Drive.

The program description is divided into three parts:

- Initialization procedures
- Data reception and storage
- Data decode and message transmission

The program listing is located in Appendix "D", flowcharts appear in Figures III-33 through III-35.

(1) Initialization Procedures

The first section of the UPI-41 program performs functions which are necessary prior to data reception. These functions are the initialization of registers and the initialization of the ADM-3A screen. The values placed in the various registers will be explained as they are encountered within the program. The screen initialization procedure consists of clearing the screen and positioning the cursor in the bottom left hand corner. The screen is cleared by transmitting a special character, 01AH, to the ADM-3A, while the cursor is positioned using the 4 character cursor positioning sequence described previously in the controller to ADM-3A interface section.

As the final step in the initialization procedures, the UPI-41 enables itself to data reception by outputting a logic zero to port 2 line 7. This port line is connected to the UPI-41 chip select, CS, input. Since all port lines are in the logic high state following a system reset, UPI-41 input is disabled until the output instruction is executed.

(2) Data Reception and Storage

When data is written into the UPI-41 interface register by the SBC 80/20, an interrupt request is generated. Upon recognition of the interrupt, the interrupt vector jump at locations 3 and 4 in program memory is executed, and the interrupt service routine, lines 118 through 134 in Appendix "D", is entered. The interrupt routine inputs the data from the interface register and places the data in a queue. A flowchart of the interrupt service routine appears in Figure III-34.
Figure III-33. UFI-41 Control Program Flowchart Processing Loop
Figure III-34. UPI-41 Control Program Flowchart - Interrupt Service Routine
Figure III-35. Interrupt Routine Flowchart

1. SELECT REGISTER BANK 1
2. SAVE ACCUMULATOR
3. INCREMENT STATUS REGISTER
4. INPUT DATA FROM INTERFACE REGISTER
5. STORE DATA AT "PUT POINTER"
6. "PUT POINTER" WRAPAROUND?
7. LOAD "PUT POINTER" WITH 1 LESS THAN STACK BASE
8. INCREMENT PUT POINTER
9. RESTORE ACCUMULATOR
10. RETURN AND RESTORE STATUS
11. IF QUEU FULL?
   YES: NEXT
   NO: 1) INCREMENT STATUS REGISTER
        2) INPUT DATA FROM INTERFACE REGISTER
        3) STORE DATA AT "PUT POINTER"
It was pointed out in the section describing the SBC 80/20 controller interface that the UPI-41 reception rate requirement would necessitate the data queue. The necessity for the queue can be shown as follows:

Unless the 19,200 baud rate can meet the UPI-41 reception rate requirement as well as the processing rate requirement, it is necessary to provide a data queue to prevent data from being overwritten in the interface register. For this condition to be met, the 19,200 baud rate must be proportionately greater than the 11,800 minimum baud rate by at least the proportion of the reception rate requirement to the processing rate requirement, or

\[
\frac{19,200}{11,800} = \frac{M}{60}
\]

as this is not true, a queue must be maintained.

To meet the storage requirements a First In First Out, or FIFO, stack is implemented in the variable data storage area of the RAM memory. See Figure III-36. A FIFO stack allows data to be retrieved so that order of entry is preserved. The operation of a FIFO stack can be conceptualized by considering a storage mechanism where data inputs are stacked one on top of the other as they arrive, and where data removal is accomplished by pulling from the bottom. As an entry is removed, all remaining entries move down one location. This operation is illustrated in Figure III-36.

The problem with this implementation is in moving the remaining data entries down. For N remaining inputs, the operation requires 2N memory accesses and 5N program steps as shown below:

(a) Increment pointer
(b) Load data byte - first memory access
(c) Decrement pointer
(d) Store data byte - second memory access
(e) Increment pointer
A more efficient algorithm uses a "get data" pointer as well as the "put data" pointer used in the implementation above. The get data pointer allows the "bottom" of the stack to move upward as data is removed from the stack. This eliminates the necessity of moving each of the remaining inputs down. Instead, the get data pointer is incremented once each time data is removed. The put data pointer always identifies the next location available for data storage and the get data pointer identifies the location of the next value to be removed. The only problem with this implementation is that unless data memory is infinitely long, storage locations will run out at some point. This condition being unacceptable, a "top-of-stack" must be defined, and as the pointers reach the top they must be wraparound. In this application the top-of-stack has been made coincident with the top of RAM, making the last location address 63 and giving a stack size of (63-32) +1, or 32 locations. As each pointer reaches location 63, it is returned to location 32 instead of being incremented further. Implemented in this manner, the number of steps required for a data removal is independent of N and, for the UPI-41, has a maximum value of 5 as indicated by lines 85 through 89 of the program listing.
For either implementation, some way of determining when the stack is full must be available. For the two pointer implementation, the queue full condition is easily detected by maintaining a queue status value which indicates how many entries are presently on the stack. If a check of the queue status register indicates that the queue is full, additional data must be rejected to avoid overwriting of the earliest entry with the newest entry. Since the UPI-41 has been designed to meet the processing rate requirement, it follows that the maximum stack usage must be less than or equal to the number of SBC 80/20 input sources, or 5; therefore, the queue full condition can never occur in this application. Use of the queue status value in this application, then, is limited to determining when data is available on the stack. Figure III-37 illustrates the operation of the moving base FIFO stack.

The put and get data pointers, the queue status, and the constants used to determine the pointer wraparound and queue full conditions are located in register bank 1. Also, since the data reception routine is entered in response to an interrupt, another bank 1 register is allocated for accumulator storage. Finally, one register is used for temporary data byte storage during computations.

Figure III-37. Moving Base FIFO Operation
Registers 0 and 1 are the only locations which can serve as pointers into the variable data storage area; therefore, the get and put data pointers are defined as the contents of registers 0 and 1 respectively, the other locations are assigned arbitrarily as per Table III-1.

<table>
<thead>
<tr>
<th>Table III-1. REGISTER BANK 1 MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register 7'</td>
</tr>
<tr>
<td>Register 6'</td>
</tr>
<tr>
<td>Register 5'</td>
</tr>
<tr>
<td>Register 4'</td>
</tr>
<tr>
<td>Register 3'</td>
</tr>
<tr>
<td>Register 2'</td>
</tr>
<tr>
<td>Register 1'</td>
</tr>
<tr>
<td>Register 0'</td>
</tr>
</tbody>
</table>

(3) Data Decode and Message Transmission

Once data is placed in the queue by the interrupt service routine, a check of the queue status register, lines 80 and 81 of the program listing, will indicate that data is available for processing. The program will then enter the main program loop, line 82, where the data decode and message transmission function begins.

This section of the program can be divided into 3 segments:

(a) Data access

(b) Source processing

(c) Message processing

1. Data Access

The function of the data access segment is to remove a data byte from the queue and perform the transition between register bank 1 operation and register bank 0 operation. The data removal steps are reminiscent of the steps performed in the interrupt routine, while the bank transition is accomplished by placing the data in the accumulator and then selecting the new register bank. A flowchart is shown in Figure III-38.

56
Figure III-38. Data Access Segment Flowchart

- Fetch data at get pointer
- Place data in temporary storage register
- Decrement queue status register
- "Get pointer" wraparound?
  - Yes: Load get pointer with 1 less than stack base
  - No: Increment get pointer
- Retrieve data from temporary storage register
- Select register bank 0
Register bank 0 is used for the remainder of the program. All locations within this bank are assigned arbitrarily as shown in Table II-2.

**TABLE III-2. REGISTER O MAP**

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register 7</td>
<td>Data Byte</td>
</tr>
<tr>
<td>Register 6</td>
<td>Message Length Constant</td>
</tr>
<tr>
<td>Register 5</td>
<td>Binary Source Identifier</td>
</tr>
<tr>
<td>Register 4</td>
<td>Linear Select Source Identifier</td>
</tr>
<tr>
<td>Register 3</td>
<td>Message Identifier</td>
</tr>
<tr>
<td>Register 2</td>
<td>Relay Counter</td>
</tr>
<tr>
<td>Register 1</td>
<td>Unused</td>
</tr>
<tr>
<td>Register 0</td>
<td>Serial Transmission Counter</td>
</tr>
</tbody>
</table>

2. Source Processing

The function of the source processing segment, lines 93 through 98, is to use the source identifier portion of the data byte to (1) determine whether a message transmission is desired and (2) position the cursor at the proper place on the ADM-3A screen. The source processing segment calls three subroutines; MASK, LOCSET, and TAB.

Subroutine MASK, lines 144 through 156, converts the binary source identifier into the linear select identifier through the use of the lookup table located at MSKDAT, line 143. The subroutine then performs the comparison with the port 2 control switch lines and sets a flag according to the result.

Subroutine LOCSET, lines 161 through 168, sends the characters which activate the cursor control logic and the Y coordinate value to the ADM-3A.

Subroutine TAB, lines 157 through 160, converts the binary source identifier into the proper X coordinate value and completes the cursor positioning sequence by transmitting the coordinate value to the ADM-3A.
3. Message Processing

The function of the message processing segment, lines 99 through 112, is to convert the message identifier portion of the data byte into the page 3 address of the message string, output the message string, scroll the ADM-3A display one line, and return to the queue status checking loop.

The page 3 address of the message string is produced by multiplying the binary message identifier by 16. Thus, the message identifier is converted into the starting address of a 16 character string which makes up the message. The multiplication is accomplished by swapping the high and low order nybbles of the data byte and then masking out the low order nybble. This operation is equivalent to four left shifts and, therefore, multiplies the source identifier by $2^4$, or 16.

Subroutine STROUT, lines 169 through 175, uses the message address produced by the preceding multiplication and the string length constant contained in register 6 to control the transmission of the 16 character message string to the ADM-3A.

The CRLF procedure, lines 107 through 110, cause the scroll of the ADM-3A display by sending the carriage return line feed combination.

Finally, register bank 1 is selected so that when the jump at line 112 occurs the register bank containing the queue status value, R2', will be addressed by the WAIT loop.

This completes the description of the control program except for the subroutine which controls character transmission. This function is accomplished by the OUTPUT subroutine, lines 176 through 191.

It was noted in the description of the clock connection, section II, that the 4.608 megahertz clock input to the UPI-41 would be used to generate the proper communication frequency. The following discussion explains this process and the operation of the OUTPUT subroutine.

Each instruction in the UPI-41 instruction set consists of either 1 or 2 instruction cycles. Each instruction cycle consists of 5 machine states and each state consists of 3 clock periods. See Figure III-39.

![Figure III-39. UPI-41 Instruction Cycle](image)
The instruction cycle execution rate, then, is $1/15$ of the input clock rate or 307,200 instruction cycles per second. The instruction cycle execution rate divided by 16 produces the serial transmission rate of 19,200 baud. Therefore, a bit interval, i.e., the time a serial bit should be present on port 1 during transmission, is exactly 16 instruction cycles. A 9 bit character can be transmitted by constructing a loop which places a new serial bit on the port 1 transmission line every 16 instruction cycles.

The OUTPUT subroutine, Figure III-40, expects the 7 least significant accumulator bits to hold the 7 bit ASCII representation of the character to be sent. As 9 bits are required to send a complete character, including the start and stop bits, the 8 bit accumulator and the carry bit are catenated to form a 9 bit register. The accumulators most significant bit and the carry bit serve as the stop and start bits respectively. Once the 9 bit register is set up with the character, the bits are sent by successively rotating the bits into the least significant bit position of the accumulator and then outputting the accumulator to port 1.

Instructions 1, 2, and 3 set up the character, the transmission loop begins at line 180. Note that the number of instruction cycles required for each instruction in the transmission loop is shown to the right of the instructions.

For the first eight bits transmitted, program execution proceeds through the steps indicated 1 through 8. As can be verified by the reader, 16 instruction cycles are executed between bit changes.

Figure III-40. UPI-41 Character Transmission Subroutine
Program flow for the final bit proceeds through the steps indicated A through E. While this sequence requires only 9 instruction cycles, analysis of the complete program shows that for any set of conditions a minimum of 8 additional instruction cycles will be required to reach the initial output instruction for a new character. Thus, a minimum of 9 + 8, or 17, cycles will be executed exceeding the minimum of 16 by 1 cycle. But, as there is no maximum length for the stop bit since its level corresponds to the nonactive, or "marking" state, the value 17 is acceptable.

The instruction executed just prior to entry into the bit transmission loop disables interrupts, while the instruction just before the return reenables them. Interrupts must be disabled during transmission of a character since the occurrence of an interrupt service routine would insert extra instruction cycles, thereby destroying the integrity of the software timing loop.

As a concluding remark on the UPI-41 control program, it is noted that starting on page 51 of the listing, a sample set of message strings is shown.

The program listing referred to throughout this section is the assembly listing produced during the assembly of the UPI-41 control program source file. This version of the program was used for the system evaluation to be presented in the next section.

3. 80/80 PROGRAM

Operation of the 80/20-4 Microcomputer is directed by program code in three 2716 2KX8 EPROMS. The code was compiled from a program written in PL/M-80 language. Reference 3 gives a number of PL/M-80 examples. While reference 5 provides the language syntax and other definitions.

The overall program strategy is shown on Figure III-41, with more detail given on Figure III-42. The program listing is given in Appendix A.

After power has been turned on, the program starts when the 80/20-4 "RESET" button is pushed.

During "initialize" the program issues a series of questions to the system console and prompts for answers as shown on Figure III-41. If desired, the date, training session number, and trainee names are obtained and stored for future reference. When all identification data have been collected and other housekeeping details completed, the program issues "LET'S START" and the main training session loop is entered. This loop may be executed along the three different paths indicated on the flowchart, Figure III-42.

If there is no target present on the screen and no rifle trigger is pulled, i.e., no "action", then path 3 will be selected by program logic. No data comments are generated during early passes around the path 3 loop. Subsequent "action" will cause flags to be set and a return to path 3 may result in a comment of either "NO TARGET" or "YOU FROZE" being sent to the earphones of any erring trainee. Corresponding error data are filled in RAM memory for the identified trainee which will lower his score printed out after the session ends.
Figure III-41. 8080 Program Strategy
Figure III-42. Program Flowchart
"ACTION" is true after a target becomes available and/or a rifle is "IRED". The session loop will now pass through either path 1 or 2 as dictated by program logic. This logic also determines the proper data to be filed and comment to be sent to the trainee's earphones. For example, if the target disappears but the trainee persists in shooting for more than one second, then a "NO TARGET" comment will be sent to the rifleman, and a negative score is placed in his data file.

Operation remains locked in the session loop until an escape is signaled by the squad leader's pushing the start/print button. This causes a system interrupt and control passes to "PRINT RESULTS" which produces output as shown on Figure III-41. The program then returns to the "INITIALIZE" block and data records are initialized in preparation for the next training session.

4. SCORE DISPLAY AND WORST PERFORMANCE

The final score for each trainee is calculated by adding the following items:

\[
\begin{align*}
&100 \times \text{(Hits per shot)} \\
&60 \times \text{(Near misses per shot)} \\
&30 \text{ if average reaction time } = 0.5 \text{ seconds, or} \\
&20 \text{ if average reaction time } 0.5 \text{ but } 0.9 \text{ seconds, or} \\
&10 \text{ if average reaction time } 0.9 \text{ but } 1.3 \text{ seconds, or} \\
&0 \text{ if average reaction time } 1.3 \text{ seconds.} \\
&-2 \times \text{(Number of targets ignored)}
\end{align*}
\]

When the light emitting diode or "LED" is lit under a trainee's CRT column, he is the "WORST SHOOTER" inasmuch as he has the highest total of these items:

- Misses
- Shots with no target present
- Targets ignored - PP the worst shooter is recomputed each time a target is not present on the screen.

5. SELF CHECK

The UIWT self check has two parts: (1) an SBC 80/20 check, and (2) the interface board (IFB) check. These procedures are described separately.

a. SBC 80/20 CHECK

The 80/20 single board computer checkout requires the insertion of an INTEL SBC 416 ROM extender board on which is located the test program driver and a duplicate of the UIWT version 1.2 program. The duplicate program is contained in five 2708 ROMS. These ROMS are located at addresses through \(\text{OCOOOH, see Appendix C.}\)
The 80/20 check verifies proper operation of the SBC memory, I/O ports, timer and USART. A complete check requires about a second. To run the check, the normal 50 pin connectors to "J1" and "J2" of the SBC 416 board must also be replaced with a special test strap which connects input terminals of J1 to output terminals on J2 and vice versa. The SBC 416 board must also be inserted into the computer card cage. While not necessary, it may be convenient to remove the interface board to avoid damage to the wire-wrap pins. The UIWT program is started with a reset in the normal way. The first thing that the UIWT program does is to determine if the SBC 416 board is in the card position on the SBC 416. If it is not in this location, no transfer acknowledge signal will be returned. After a millisecond, therefore, the attempt is aborted and a "00" is read into the 8080's accumulator. The 00 occurs due to the 1K pullups on the inputs of inverting line drivers. With a 1 found at 0C000H, the test program is conducted, while if a 00 is returned, the normal UIWT program starts.

The 80/20 test signals its completion by turning on a "LED" located near the top right corner of the 80/20 board. If no flashes are noted, the test was completed in a satisfactory manner. If flashes occur, some trouble was uncovered, the nature of which is indicated by the number of quick flashes grouped together. The trouble code is:

1 FLASH = RAM FAILURE
2 FLASHES = ROM FAILURE
3 FLASHES = I/O FAILURE
4 FLASHES = TIMER TOO SLOW
5 FLASHES = TIMER TOO FAST
6 FLASHES = USART FAILURE

A TIMER TOO FAST
6 FLASHES = USART FAILURE

A flash group is sent for each failure detected, so it is possible that more than one trouble code may be detected during a single 80/20 test. The test may be terminated only by turning off the power. This should be done, of course, before any physical changes are made to the computer.

b. IFB SELF TEST

To test the interface board, the 80/20 test SBC 416 board may be removed from the card cage and the IFB reinserted. The normal 50 pin rifle input strap connector and the 34 pin strap connector must be removed. A special strap must be connected between the 50 pin rifle test simulator output and the 50 pin rifle input connector.

The CRT and Electronic Data Terminal, LDT, are needed for the test. The VOTRAX unit may optionally be disconnected or left connected.

To conduct the test, the UIWT system is started as usual by a start or reset. When the query "WANT ID YES OR NO?" is presented on the EDT, a control-I for "TEST", should be entered. The system should respond with:
RIFLE SIMULATOR
STRAP IN PLACE?

If no errors are found, the test requires 18 to 19 seconds to run, and the system responds with:

TEST COMPLETE.

Hitting any key on the data terminal will result in the standard output being typed out, as shown in the attached listing.

If trouble is detected, the rifle "ID" number will be typed with "F" for failure and a coded diagnostic: RES, INT or DAT.

RES == The rifle did not receive a reset pulse.
INT == The UPI-41 did not receive the data strobe that it originally sent.
DAT == Improper data was received by the UPI-41.

H. FILM ANIMATION

This section describes the process by which a second projection film is produced to enable the electro-optic rifle receiver to sense targets on the movie screen and score the trainee's performance.

A black-and-white animated companion film is prepared for simultaneous synchronized projection on an infrared projector with a full color battle scenario on a second projector. The black-and-white film is animated frame-by-frame to produce a clear target zone surrounded by an opaque field. If more than one target is present then more than one clear target zone is animated within a frame.

Infrared light is projected through the clear target area to produce a target zone for the rifle electro-optic sensors. This infrared target zone is usually animated to directly overlay the visual target being projected by the full color battle scenario but may also be super-elevated and/or lead the target as required.

Film animation has been accomplished by inspecting a battle scenario frame-by-frame on a Movieola (a type of laboratory film analyzer). Each frame is inspected for the total number of targets, and target location. Each frame is catalogued and then compared to each other for target range and rate of transverse motion measured. Lead and superelevation calculations are computed from this data for final animation.

Selected target shapes and sizes are prepared for use in the animation process. These shapes have usually been silhouette, oval, and circles. The target shape is realized as an opaque shape on a clear strip of acetate.
The battle scenario is again viewed frame-by-frame in the animation process. The animator locates targets according to the script and overlays on appropriate target size and shape on a rear projection screen. Lead and superelevation corrections are applied if necessary and then the battle scenario is removed while a single frame of the animated film is exposed.

Upon completion the entire animated film is developed using a reversal process. This process causes the opaque target shapes to become transparent and the background to be opaque. The contrast is adjusted for a $D^* = 2.5$ or better. A $D^*$ of 2.0 has been used successfully.

Copies are made of this master animated film and both battle scene and the animated films are edge numbered for easy identification and editing. A final coat of lacquer is then added to protect the emulsions and extend the life of the films.

The resolution of man targets beyond 300 meters is difficult and is a limitation of the system using standard 16mm film.

No research was done on automating the production of the IR target film. It is our opinion that research in this area could reduce both the time and expense of producing the infrared target film.
SECTION IV

CONCLUSIONS

The UIWT was tested successfully by the U.S. Marine Corps and by the U.S. Army, under the different name SWAT (Squad Weapons Analytical Trainer). The two systems are virtually identical except for the number of trainee firing positions. UIWT has four firing positions while SWAT consists of five firing positions.

Evaluation of the UIWT's training effectiveness and potential was performed in November 1979 at Camp Lejeune, North Carolina, by the U.S. Marine Corps. The evaluation was conducted by three members of Code N-241, Naval Training Equipment Center. Three different groups of Marines acted as test subjects or provided expert opinion. They are as follows:

(a) 120 enlisted men took part in a quasi-experiment designed to determine benefits of the UIWT versus a more traditional training method,

(b) Eight highly experienced snipers evaluated the UIWT for its training capabilities. These Marines also serve as marksmanship instructors,

(c) A variety of General, Field and Company grade officers fired the simulator and gave opinions concerning its usefulness.

The following comments on testing are extracted from Reference (8). Generally, the infantrymen were very positive about their experience with the UIWT. They would like to see deployment of the device into actual training situations. An overwhelming number stated that they would rather train in the UIWT than on the pop-up range. The elements of realism and immediate feedback were the main reasons for infantrymen satisfaction with the UIWT.

Without exception the officers who fired and observed the UIWT opined that it was a valuable training tool. Some officers went so far as to request that the UIWT prototype remain at Camp Lejeune so that they could start training Marines. A number of officers expressed concern about UIWT maintainability and reliability. They felt that if the UIWT was to be used for large numbers of trainees it would have to have rigid specifications for reliability.

The Marines suggested that there were no special features of the UIWT system (i.e., feedback, recoil, instructor console, infrared monitor, etc.) which should be deleted. All features seemed acceptable and desirable to those who evaluated the system.

The UIWT functioned well throughout the study. Breakdowns and malfunctions occurred on few occasions. This performance record is even more impressive when it is considered that this version of the UIWT is a prototype. The maintainability and reliability of the UIWT, based upon this evaluation, must be considered as good. Future iterations in the production format should only serve to increase these two characteristics.
Despite test limitations, the UIWT evaluation was considered to be successful by those participating in the evaluation. The overwhelming enthusiasm for the training device, exhibited by the Marine personnel who fired it and observed it, gave evidence of its potential usefulness.

The evaluation of the UIWT at Camp Lejeune produced positive findings. Every characteristic of the UIWT met with approval. An evaluation of trial scores on the UIWT provided empirical evidence of the UIWT's effectiveness.

No formalized Program of Instruction (POI) exists for team firing training. Consequently, none was administered with the UIWT evaluation. This area of training should be given consideration in the future. Presently the instructor merely gives informal directions on how a fire team should function. The same informal procedure is followed for marksmanship instruction. Observation of the UIWT evaluation identified a number of factors which should be formally addressed in instruction for team fire.

(a) How large is a fire sector?
(b) Enemy tactics (i.e., Warsaw Pact, Vietnamese)
(c) Ammunition rationing
(d) Change in fire team tactics if a member is made inoperable (i.e., gun jams or casualty)

There is presently no training which accomplishes the objectives that the UIWT addresses (e.g., fire team training, on board ship practice and training, and providing realistic combat scenarios which require application of proper aiming techniques). Since the Marine Corps deems these objectives to be important, it is recommended that the UIWT effort be funded and preparation be made for contractor production of the system. In addition, development of the UIWT's capability to simulate other infantry weapons such as the TOW, DRAGON, mortars, etc., should continue. Also, any formalized instruction should include techniques for firing at moving targets (including leading, firing into brush cover, and firing through smoke). It is possible for an instructor to forget to cover many of these important points when it is presented in an informal manner.

The UIWT has a counter which allows the number of rounds fired during a session to be accurately assessed. This counter showed that over 40,000 electronic rounds were used for the evaluation. Presently, costs for live M-16 ammunition are approximately nine cents per round. At this rate, the UIWT accomplished an ammunition cost savings of nearly $4,000. This figure if applied to a year's worth of training, would be a significant savings. At this rate the UIWT would pay for itself in a year. If the transportation fuel costs were added to the ammunition cost the savings would be even more impressive.
The U.S. Army test was conducted by the U.S. Army Infantry Board (USAIB) for the Directorate of Training Developments, U.S. Army Infantry School (USAIS), Fort Benning, Georgia (See Reference 9). The test was conducted 22 January through 5 March 1980, employing test soldiers from the U.S. Army Marksmanship Unit and TOE units. Testing included use of the SWAT as a vehicle for training riflemen a technique of engaging moving personnel targets. A comparison of record fire scores, achieved by personnel trained on the SWAT, on the Infantry Remoted Target System, Defense Test Range, and a no-training control group, was conducted to address training potential. Figure IV-1 indicates the results of firers interviews concerning the SWAT weapon and SWAT targets. Some of the major findings were that test soldiers improved their firing performance on three iterations of SWAT firing, but not on three iterations of DTR firing. It is stated that "this test does give some evidence of the SWAT system's potential for training transfer. However, a final estimate of the systems training value cannot be made until a training program is developed which optimizes SWAT performance".
<table>
<thead>
<tr>
<th>A</th>
<th>UIWT WEAPON</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ease of zeroing (easier-harder)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Recoil (less-more)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Muzzle climb (less-more)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Amount of noise (less-more)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Sound duration (shorter-longer)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Trigger squeeze (easier-harder)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Amount of slack in the trigger (less-more)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Balance (worse-better)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Weight (lighter-heavier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Ease of magazine change (easier-harder)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Accuracy (less-more)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Obtaining a good sight picture (easier-harder)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Getting a hit (easier-harder)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>UIWT training (worse-better)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>UIWT TARGETS</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Seeing targets at 0-100 meters (harder-easier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Seeing targets at 101-200 meters (harder-easier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Seeing targets at 201-300 meters (harder-easier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Determining target ranges (harder-easier)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In all cases the UIWT rifle and targets are being compared to the M16A1 and the DTR range. If there was no difference between them, then a 4 was chosen. The farther away the mean is from 4, the more difference there is between the UIWT and the M16A1.

Figure IV-1. Results of Firers' Interview (Means)

71
REFERENCES


7. HPI-41 User's Manual (Preliminary), Intel Order Number 9800504A.


APPENDIX A

FUNCTIONAL DESCRIPTION OF CONSOLE SWITCHES
FUNCTIONAL DESCRIPTION OF CONSOLE SWITCHES

Figure A-1 displays the Instructor's Console. The right side display screen presents printed columns showing trainees results each time they fire. And indicates by a red light which trainee is doing the poorest. The miniature toggle switches must be in the up (or on) position for any trainee participating in the exercise. The left side display screen presents positions of movie targets and indicates where trainee is aiming and where his round has been fired in relation to where the target appeared.

On the main, center panel, the instructor operated switches are arranged and labeled in groups by function. They are:

1. Main power (single switch)
2. Microcomputer (three switches)
3. Computer Voice (two switches)
4. Projector (two switches)
5. Motor (three position rotary switch)
6. Boresite (single switch)
7. Recoil (single switch)
8. Score display (single switch)
9. Audio Communications (eleven switches)
10. Weapon motion system (eleven switches)

The operation of these switches follows:

MAIN POWER - Applies the ac power to entire console.

MICROCOMPUTER - "On" applies operating voltages to computer.

"Reset" clears all previous data from computer and asks if individual identification of each trainee is required. If none is required, depress any key except Y (For "yes"). Computer will signify system is ready by printing out "let's start".

"Start print" starts the hard copy printout process printing trainees results and analysis.

If individual scoring identification is required, depress the letter Y on typewriter. Computer will ask you for today's date, session number and to enter in name of first trainee, then instruct you to identify the others in same manner. NOTE: Names of trainees must be limited to those containing no more than six letters.

A-1
Figure A-1. Instructor's Console - Front Panel Control Locations
COMPUTER VOICE - Permits instructor to activate the computer voice carried to trainee headphones. The voice reset button should be pushed whenever the computer voice is turned on.

PROJECTORS - "Locked-Unlocked" switch either locks or unlocks synchronous motors of IR and visual projectors together so that target frame and visual frame coincide, or permits instructor to operate projectors independently during alignment. For normal operation the switch should remain locked.

"Start Movie" physically starts the movie if motor switch is not in off position.

MOTOR - This three position rotary switch performs the function of applying ac power first to projector motor and then to projector lamps (this procedure is conventional in all movie projectors to conserve lamp life).

BORESIGHT - This switch allows instructor to override the "target present" signal delivered to computer by the projector, so that trainees can fire at the target box instead of the movie screen. When lower half of this switch is illuminated the system is in normal (movie screen) position.

RECOIL - This switch gives instructor the option to conduct exercise with or without rifle recoil.

SCORE DISPLAY - This switch clears the right side screen display.

AUDIO COMMUNICATIONS - This bank of eleven switches permits the instructor to listen in on the headphone of any trainee, and talk to any one or all trainees.

WEAPON ACTION SYSTEM - This bank of eleven switches allows the instructor to view, or the left side display screen, exactly where any, or all, trainees are aiming and firing in relation to where targets appeared on movie screen. The upper bank of switches selects the trainee to be observed. The lower bank of switches either energizes a trainee's laser continually or only when he had fired his rifle. These switches supply operating power to the laser attached to each rifle, and must be held pushed in for the laser to operate.
APPENDIX B

TEST EQUIPMENT
APPENDIX B
TEST EQUIPMENT

1. LASER CHECKER

The laser checker test box, Figure B-1, allows instructor to make a go-or-no-go check of the UIWT tracking laser which is attached to the simulated rifle. Referring to the schematic, Q1 is a silicon photodiode which responds to IR energy from the rifle laser. When the rifle trigger is pulled, with the laser positioned a few inches from the detector, Q1 detects the infrared pulse, and delivers an output signal to Q2.

Q2 is a high gain amplifier whose output is a sharply rising positive pulse that provides the gating signal required to turn on silicon controlled rectifier Q3, placing its anode at ground potential, and allowing capacitor C3 to begin charging up toward Vcc through resistor R5, and energizes the Sonalert alarm producing an audio tone of approximately 2 KHz.

Q4 is a unijunction transistor oscillator which is enabled whenever capacitor C3 is returned to ground. As capacitor C3 starts to charge toward Vcc it produces an exponentially rising DC voltage at the emitter junction of Q4. When this voltage reaches the breakdown point for this particular unijunction, Q4 conducts heavily, shorting emitter to ground thereby allowing the charge on C3 to dissipate, lifting the anode of Q3 off of ground which turns off the SCR, disabling the audio alarm, and the entire circuit is again ready for recycling.

Power for the laser checker is externally provided via binding posts. Any battery voltage from six to thirty volts may be used. The audio alarm sound level varies with battery voltage. Fifteen volts provides more than adequate sound level for an average room.

2. RIFLE CHECKER

By using an ordinary penlite flashlight as a source of light energy, and directing the light into the lens of the rifle receiver, the rifle checker test box allows the instructor to confirm that the four quadrant detector in the rifle is functioning, the "Full Clip" feature of the UIWT system is operating correctly, that the rifle mode select switch, and rifle trigger switch, are also functioning properly. In addition, the UIWT laser can be tested using the laser checker in conjunction with the rifle checker.

Detector Test: Referring to the schematic of the rifle checker, Figure B-2, connections to the rifle are made via a T&B connector identical to that with the UIWT system. Connector pin numbers are identified on the schematic. ICl is a quad comparator which establishes the signal strength reference level of the IR energy received. Each of the four comparators parameters are identical. The desired reference level is established on the non-inverting inputs of ICl. The rifle detector outputs appear as signal at the inverting inputs.
of ICI. When this input signal is of an amplitude equal to the reference level established via the variable resistor, the outputs of ICI rise to a logical "HIGH". This output "HIGH" signal appears on one input to IC2 which is a quad NAND gate. Since the remaining NAND inputs are permanently "HIGH", IC2 produces a logical "LOW" at its output, providing a path to ground for that particular LED, allowing it to light up, indicating that the UIWT rifle detector is operational. By pointing the flashlight beam into the rifle receiver lens and moving it about, it is possible to observe that all four quadrants of UIWT's rifle detector are indeed functioning.

Ammo Magazine Test: One half of IC4 is a timer arranged to produce one pulse of approximately five second duration whenever it receives an input, thereby testing the "full ammo magazine" feature of UIWT. A "full magazine" is simulated by one whose internally-mounted capacitor has been charged. A separate, partitioned-holding/charging tray which accommodates up to 30 magazines is provided with the instructor's console. In future modifications, this holding/charging tray can be incorporated into the console. Referring to the schematic, when a (charged capacitor) full magazine is inserted in the rifle, the capacitor discharges into one gate of IC3, a NAND, providing a negative going signal as input to timer IC4. When IC4 turns on its output goes "HIGH" allowing LED 5 to light for approximately five seconds. If an "empty magazine" (i.e. one which has expended its full thirty rounds) is inserted into the UIWT rifle, it cannot energize the circuit and LED 5 will not illuminate.

Laser Test: This test is performed by one half of IC4. The circuit is a free-running oscillator delivering sharp positive pulses of approximately one microsecond duration at a PRR of 5 KHz. This output is directly connected to the laser input via the normal connecting cable. When the "laser test" button on the rifle checker box is depressed, the output of oscillator IC4 is allowed to trigger the laser, and the laser will emit IR energy.

By setting up the laser checker* box several inches from the laser, an audio alarm tone indicates correct operation of the laser each time the test button is depressed. *Circuit operation is described in Paragraph 1 of this Appendix B.

Rifle Trigger and Mode Switch Test: Correct physical functioning of the UIWT rifle trigger is verified by observing the test lights LED 6 and LED 7. LED 6 will normally be on. When the rifle trigger is pulled LED 6 goes off for a fraction of time, LED 7 comes on during that instant, and as the trigger is pulled all the way LED 6 comes back on and remains on.

Rifle Mode Switch Test: The UIWT rifle mode select switch is tested by observing the operation of LED 8 and LED 9. In the semi-automatic mode, LED 8 is on and in the automatic mode LED 9 is on.
An internal dual fifteen volt power supply provides operating voltages for the UIWT as well as for the pre-amp receiver in the UIWT rifle. A single miniature dry cell forty-five volt battery within the test box supplies operating power for the UIWT laser.

3. **RIFLE SUBSTITUTION BOX**

Using the rifle substitution box, it is possible to simulate the operation of a UIWT rifle on the instructor's console. It can simulate a rifle being fired either in automatic or semi-automatic mode, and can simulate the target information normally received by the rifle's quadrant detector from the projected movie image.

The rifle substitution box is connected to the instructor's console via a TAB connector identical to that on a UIWT rifle. Power for the test box is derived from the console. Operation of the trigger switch and the mode switch are obvious. Capacitor C1 simulates the capacitor which is internally mounted inside of each magazine. It remains in the charged state via switch SW4 as long as the test box power switch is in the "on" position. When the counting circuit in the UIWT electronics indicates to the computer that thirty rounds have been expended by allowing no more shots, the insertion of a new magazine is simulated by momentarily engaging switch SW1.

Target information is simulated by the status of four switches, SW5, SW6, SW7, and SW8. IC1 is a free-running oscillator delivering a square wave at a PRR of 96 Hz (which is the rate at which the projected target energy is chopped). The output of the oscillator is delivered to the console via switches SW5 through 8 as simulated rifle detector signals.

Referring to the schematic, Figure B-3, the adopted convention for the four quadrant rifle detector is shown. Switches may be independently thrown to either ground (low) or to the 96 Hz positive pulse output of IC1, constituting either a true or a false logic signal to the UIWT console. Any one of ten possible combinations of hits or near misses can be duplicated by these four switches. For example: Switches SW3, SW4 "Low" and switches SW1, SW2 "High" will be recorded by the computer as a "low right" signal from the simulated rifle. Similarly, SW1, SW4, SW2 "High" and SW3 "Low" would simulate a UIWT rifle detector condition where IR energy is centered on those quadrants. This information would be interpreted by the computer as a trainee having fired "Low" at the target.

4. **BORESIGHT BOX**

The boresight box, Figure B-4, allows the instructor to initially check the closeness of UIWT rifle boresight alignment. A free-running oscillator IC1 with a PRR of 96 Hz delivers input pulses to a Darlington amplifier which consists of Q1 and Q2. The Darlington amplifier pulses a high intensity incandescent lamp at 96 Hz. The visible portion of light is filtered out so the rifle is aimed at the black-outlined aiming pip on the box, fires the weapon, receives audio
feedback results via his headphones, and adjusts rifle sights for accurate alignment of front and rear sights.

Power for the boresight box is supplied externally via one small, 12 volt rechargeable sealed-gel battery. The boresight box can also be utilized as a marksmanship target. The instructor simply enables the front panel "target present" switch allowing the UIWT console to disregard target information from the movie screen, and instead, to accept target information from the boresight box.
APPENDIX C
SYSTEM PROGRAM
0004H LIN 79
0007H LIN 80
0008H LIN 81
0009H LIN 82
000AH LIN 83
000BH LIN 84
000CH LIN 85
000DH LIN 86
000EH LIN 87
000FH LIN 88
0010H LIN 89
0011H LIN 90
0012H LIN 91
0013H LIN 92
0014H LIN 93
0015H LIN 94
0016H LIN 95
0017H LIN 96
0018H LIN 97
0019H LIN 98
001AH LIN 99
001BH LIN 100
001CH LIN 101
001DH LIN 102
001EH LIN 103
001FH LIN 104
0020H LIN 105
0021H LIN 106
0022H LIN 107
0023H LIN 108
0024H LIN 109
0025H LIN 110

MOD STARTUPMODULE

MEMORY

INTERRUPTCALL

HISTORY

NAME

DATE

IONumber

WHEN

UIVAD

IDENT

QUERY

TRAINING

ONRIFLE

X

I

J

K

JK

IDFLAG

PORTSET

SETDATA

POINTER

LENGTH

VALUE

FINAL

LOUP

SIMULATERRIFLES

RUNTYPE

STARTUP

SETINT

GETID

LOAD

GETDATE

GETIDNUMBER

C-2
<table>
<thead>
<tr>
<th>Hex Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>04CCH</td>
<td>46</td>
</tr>
<tr>
<td>04CH</td>
<td>47</td>
</tr>
<tr>
<td>04D5H</td>
<td>48</td>
</tr>
<tr>
<td>04D9H</td>
<td>49</td>
</tr>
<tr>
<td>04F3H</td>
<td>50</td>
</tr>
<tr>
<td>04F6H</td>
<td>51</td>
</tr>
<tr>
<td>0500H</td>
<td>52</td>
</tr>
<tr>
<td>0503H</td>
<td>53</td>
</tr>
<tr>
<td>050CH</td>
<td>54</td>
</tr>
<tr>
<td>0513H</td>
<td>55</td>
</tr>
<tr>
<td>0521H</td>
<td>56</td>
</tr>
<tr>
<td>0521H</td>
<td>57</td>
</tr>
<tr>
<td>052CH</td>
<td>58</td>
</tr>
<tr>
<td>053CH</td>
<td>59</td>
</tr>
<tr>
<td>054EH</td>
<td>60</td>
</tr>
<tr>
<td>054EH</td>
<td>61</td>
</tr>
<tr>
<td>056DH</td>
<td>62</td>
</tr>
<tr>
<td>057DH</td>
<td>63</td>
</tr>
<tr>
<td>0588H</td>
<td>64</td>
</tr>
<tr>
<td>0588H</td>
<td>65</td>
</tr>
<tr>
<td>0593H</td>
<td>66</td>
</tr>
<tr>
<td>0593H</td>
<td>67</td>
</tr>
<tr>
<td>059BH</td>
<td>68</td>
</tr>
<tr>
<td>059BH</td>
<td>69</td>
</tr>
<tr>
<td>05A4H</td>
<td>70</td>
</tr>
<tr>
<td>05A9H</td>
<td>71</td>
</tr>
<tr>
<td>05B2H</td>
<td>72</td>
</tr>
<tr>
<td>05B6H</td>
<td>73</td>
</tr>
<tr>
<td>05BAH</td>
<td>74</td>
</tr>
<tr>
<td>05BFH</td>
<td>75</td>
</tr>
<tr>
<td>05C8H</td>
<td>76</td>
</tr>
<tr>
<td>05D4H</td>
<td>77</td>
</tr>
<tr>
<td>05D4H</td>
<td>78</td>
</tr>
<tr>
<td>05EEH</td>
<td>79</td>
</tr>
<tr>
<td>05FAH</td>
<td>80</td>
</tr>
<tr>
<td>05FDH</td>
<td>81</td>
</tr>
<tr>
<td>05FDH</td>
<td>82</td>
</tr>
<tr>
<td>060DH</td>
<td>83</td>
</tr>
<tr>
<td>0615H</td>
<td>84</td>
</tr>
<tr>
<td>061DH</td>
<td>85</td>
</tr>
<tr>
<td>0625H</td>
<td>86</td>
</tr>
<tr>
<td>0625H</td>
<td>87</td>
</tr>
<tr>
<td>062FH</td>
<td>88</td>
</tr>
<tr>
<td>062FH</td>
<td>89</td>
</tr>
<tr>
<td>0632H</td>
<td>90</td>
</tr>
<tr>
<td>0632H</td>
<td>91</td>
</tr>
<tr>
<td>063AH</td>
<td>92</td>
</tr>
<tr>
<td>063FH</td>
<td>93</td>
</tr>
<tr>
<td>0644H</td>
<td>94</td>
</tr>
<tr>
<td>0644H</td>
<td>95</td>
</tr>
<tr>
<td>064FH</td>
<td>96</td>
</tr>
<tr>
<td>0654H</td>
<td>97</td>
</tr>
<tr>
<td>0654H</td>
<td>98</td>
</tr>
<tr>
<td>065CH</td>
<td>99</td>
</tr>
<tr>
<td>0664H</td>
<td>100</td>
</tr>
<tr>
<td>0664H</td>
<td>101</td>
</tr>
<tr>
<td>0669H</td>
<td>102</td>
</tr>
<tr>
<td>066EH</td>
<td>103</td>
</tr>
<tr>
<td>0671H</td>
<td>104</td>
</tr>
<tr>
<td>0685H</td>
<td>105</td>
</tr>
<tr>
<td>0685H</td>
<td>106</td>
</tr>
<tr>
<td>068CH</td>
<td>108</td>
</tr>
<tr>
<td>068EH</td>
<td>109</td>
</tr>
<tr>
<td>0696H</td>
<td>110</td>
</tr>
<tr>
<td>0696H</td>
<td>111</td>
</tr>
<tr>
<td>0692H</td>
<td>112</td>
</tr>
</tbody>
</table>
0687H LIN 114
06C7H LIN 119
06D7H LIN 119
06E1H LIN 119
06E4H LIN 119
06E9H LIN 120
06E9H LIN 120
06EEH LIN 120
06F1H LIN 121
06F2H LIN 121
0700H LIN 121
070AH LIN 121
0719H LIN 122
0720H LIN 130
0727H LIN 131
073AH LIN 132
073FH LIN 133
0748H LIN 134
074AH LIN 135
0752H LIN 136
0757H LIN 137
0757H LIN 138
075EH LIN 139
0769H LIN 140
076AH LIN 141
076AH LIN 142
0778H LIN 145
0793H LIN 146
0798H LIN 147
07C0H LIN 148
07C7H LIN 149
07CCH LIN 150
07D1H LIN 151
07DFH LIN 152
07F8H LIN 153
0808H LIN 154
0821H LIN 155
0829H LIN 157
0832H LIN 158
0832H LIN 159
0839H LIN 160
083EH LIN 161

MOD CONSOLE/MODULE
541FH SYM MEMORY
033FH SYM HELLO
3409H SYM X
34CAH SYM Y
34CBH SYM GOS
0951H SYM TTYSET
0964H SYM TTYRES
0871H SYM VOTRAXSET
087AH SYM VOTRES
0891H SYM CIN
0991H SYM RXKLY
09A1H SYM TXFLY
08AFH SYM COUT
18CH SYM ITEM
08C3H SYM BITCMP
08BCH SYM FRINTNUM
0912H SYM PRINT
18CDH SYM POINTER
18CFH SYM FINHL
0925H SYM LOOP
0943H SYM GREETING
38D' H SYM 1
38D' H SYM 2
38D' H SYM SUMSHOTS
38D' H SYM NEARMISSSES
38D' H SYM AVGTIME
38EH SYM CONVRT
38DAH SYM HEX
094EH SYM RIFLE10
095EH SYM TOTALSHOTS
0964H SYM RIFLEHIT
0968H SYM RIFLEMISS
0974H SYM RIFLELOW
0978H SYM RIFLELOWRIGHT
0988H SYM RIFLERIGHT
0991H SYM RIFLEHIGHRIGHT
09F6H SYM RIFLEHIGH
09A7H SYM RIFLEHIGHLEFT
09B4H SYM RIFLELEFT
09BCH SYM RIFLELOWLEFT
09C6H SYM RIFLETURKEY
09D9H SYM RIFLETARGETIGNORED
09E6H SYM BLANK
09E9H SYM HOMEMYSHOTS
09FBH SYM AVERAGETIME
0AA0H SYM UNITS
0ASP H SYM YOURSCORE
0AEEH SYM PRESENTRESULTS
0A9DH SYM ONERRIFLERESULTS
0AC2H SYM TYPEIT
0624H SYM SUM
H85CH SYM SUM2
H864H LIN 24
H869H LIN 26
H836H LIN 27
H846H LIN 28
H846H LIN 29
H866H LIN 31
H866H LIN 32
H866H LIN 33
H866H LIN 34
H872H LIN 36
H872H LIN 37
H872H LIN 38
H872H LIN 39
H87EH LIN 40
H87EH LIN 41
H87EH LIN 42
H87EH LIN 43
H87EH LIN 44
H87EH LIN 45
H87EH LIN 46
H87EH LIN 47
H87EH LIN 48
H87EH LIN 49
H87EH LIN 50
H87EH LIN 51
H87EH LIN 52
H87EH LIN 53
H87EH LIN 54
H87EH LIN 55
H87EH LIN 56
H87EH LIN 57
H87EH LIN 58
H87EH LIN 59
H87EH LIN 60
H87EH LIN 61
H87EH LIN 62
H87EH LIN 63
H880H LIN 64
H880H LIN 65
H880H LIN 66
H880H LIN 67
H880H LIN 68
H880H LIN 69
H880H LIN 70
H880H LIN 71
H880H LIN 72
H880H LIN 73
H880H LIN 74
H880H LIN 75
H880H LIN 76
H880H LIN 77
H880H LIN 78
H880H LIN 79

C-9
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0024H</td>
<td>00</td>
</tr>
<tr>
<td>0025H</td>
<td>01</td>
</tr>
<tr>
<td>0026H</td>
<td>02</td>
</tr>
<tr>
<td>0027H</td>
<td>03</td>
</tr>
<tr>
<td>0028H</td>
<td>04</td>
</tr>
<tr>
<td>0029H</td>
<td>05</td>
</tr>
<tr>
<td>002AH</td>
<td>06</td>
</tr>
<tr>
<td>002BH</td>
<td>07</td>
</tr>
<tr>
<td>002CH</td>
<td>08</td>
</tr>
<tr>
<td>002DH</td>
<td>09</td>
</tr>
<tr>
<td>002EH</td>
<td>0A</td>
</tr>
<tr>
<td>002FH</td>
<td>0B</td>
</tr>
<tr>
<td>0030H</td>
<td>0C</td>
</tr>
<tr>
<td>0031H</td>
<td>0D</td>
</tr>
<tr>
<td>0032H</td>
<td>0E</td>
</tr>
<tr>
<td>0033H</td>
<td>0F</td>
</tr>
<tr>
<td>0034H</td>
<td>10</td>
</tr>
<tr>
<td>0035H</td>
<td>11</td>
</tr>
<tr>
<td>0036H</td>
<td>12</td>
</tr>
<tr>
<td>0037H</td>
<td>13</td>
</tr>
<tr>
<td>0038H</td>
<td>14</td>
</tr>
<tr>
<td>0039H</td>
<td>15</td>
</tr>
<tr>
<td>003AH</td>
<td>16</td>
</tr>
<tr>
<td>003BH</td>
<td>17</td>
</tr>
<tr>
<td>003CH</td>
<td>18</td>
</tr>
<tr>
<td>003DH</td>
<td>19</td>
</tr>
<tr>
<td>003EH</td>
<td>1A</td>
</tr>
<tr>
<td>003FH</td>
<td>1B</td>
</tr>
<tr>
<td>0040H</td>
<td>1C</td>
</tr>
<tr>
<td>0041H</td>
<td>1D</td>
</tr>
<tr>
<td>0042H</td>
<td>1E</td>
</tr>
<tr>
<td>0043H</td>
<td>1F</td>
</tr>
<tr>
<td>0044H</td>
<td>20</td>
</tr>
<tr>
<td>0045H</td>
<td>21</td>
</tr>
<tr>
<td>0046H</td>
<td>22</td>
</tr>
<tr>
<td>0047H</td>
<td>23</td>
</tr>
<tr>
<td>0048H</td>
<td>24</td>
</tr>
<tr>
<td>0049H</td>
<td>25</td>
</tr>
<tr>
<td>004AH</td>
<td>26</td>
</tr>
<tr>
<td>004BH</td>
<td>27</td>
</tr>
<tr>
<td>004CH</td>
<td>28</td>
</tr>
<tr>
<td>004DH</td>
<td>29</td>
</tr>
<tr>
<td>004EH</td>
<td>2A</td>
</tr>
<tr>
<td>004FH</td>
<td>2B</td>
</tr>
<tr>
<td>0050H</td>
<td>2C</td>
</tr>
<tr>
<td>0051H</td>
<td>2D</td>
</tr>
<tr>
<td>0052H</td>
<td>2E</td>
</tr>
<tr>
<td>0053H</td>
<td>2F</td>
</tr>
<tr>
<td>0054H</td>
<td>30</td>
</tr>
<tr>
<td>0055H</td>
<td>31</td>
</tr>
<tr>
<td>0056H</td>
<td>32</td>
</tr>
<tr>
<td>0057H</td>
<td>33</td>
</tr>
<tr>
<td>0058H</td>
<td>34</td>
</tr>
<tr>
<td>0059H</td>
<td>35</td>
</tr>
<tr>
<td>005AH</td>
<td>36</td>
</tr>
<tr>
<td>005BH</td>
<td>37</td>
</tr>
<tr>
<td>005CH</td>
<td>38</td>
</tr>
<tr>
<td>005DH</td>
<td>39</td>
</tr>
<tr>
<td>005EH</td>
<td>3A</td>
</tr>
<tr>
<td>005FH</td>
<td>3B</td>
</tr>
<tr>
<td>0060H</td>
<td>3C</td>
</tr>
<tr>
<td>0061H</td>
<td>3D</td>
</tr>
<tr>
<td>0062H</td>
<td>3E</td>
</tr>
<tr>
<td>0063H</td>
<td>3F</td>
</tr>
<tr>
<td>0064H</td>
<td>40</td>
</tr>
<tr>
<td>0065H</td>
<td>41</td>
</tr>
<tr>
<td>0066H</td>
<td>42</td>
</tr>
<tr>
<td>0067H</td>
<td>43</td>
</tr>
<tr>
<td>0068H</td>
<td>44</td>
</tr>
<tr>
<td>0069H</td>
<td>45</td>
</tr>
<tr>
<td>006AH</td>
<td>46</td>
</tr>
<tr>
<td>006BH</td>
<td>47</td>
</tr>
<tr>
<td>006CH</td>
<td>48</td>
</tr>
<tr>
<td>006DH</td>
<td>49</td>
</tr>
<tr>
<td>006EH</td>
<td>4A</td>
</tr>
<tr>
<td>006FH</td>
<td>4B</td>
</tr>
<tr>
<td>0070H</td>
<td>4C</td>
</tr>
<tr>
<td>0071H</td>
<td>4D</td>
</tr>
<tr>
<td>0072H</td>
<td>4E</td>
</tr>
<tr>
<td>0073H</td>
<td>4F</td>
</tr>
<tr>
<td>0074H</td>
<td>50</td>
</tr>
<tr>
<td>0075H</td>
<td>51</td>
</tr>
<tr>
<td>0076H</td>
<td>52</td>
</tr>
<tr>
<td>0077H</td>
<td>53</td>
</tr>
<tr>
<td>0078H</td>
<td>54</td>
</tr>
<tr>
<td>0079H</td>
<td>55</td>
</tr>
<tr>
<td>007AH</td>
<td>56</td>
</tr>
<tr>
<td>007BH</td>
<td>57</td>
</tr>
<tr>
<td>007CH</td>
<td>58</td>
</tr>
<tr>
<td>007DH</td>
<td>59</td>
</tr>
<tr>
<td>007EH</td>
<td>5A</td>
</tr>
<tr>
<td>007FH</td>
<td>5B</td>
</tr>
<tr>
<td>0080H</td>
<td>5C</td>
</tr>
<tr>
<td>0081H</td>
<td>5D</td>
</tr>
<tr>
<td>0082H</td>
<td>5E</td>
</tr>
<tr>
<td>0083H</td>
<td>5F</td>
</tr>
<tr>
<td>0084H</td>
<td>60</td>
</tr>
<tr>
<td>0085H</td>
<td>61</td>
</tr>
<tr>
<td>0086H</td>
<td>62</td>
</tr>
<tr>
<td>0087H</td>
<td>63</td>
</tr>
<tr>
<td>0088H</td>
<td>64</td>
</tr>
<tr>
<td>0089H</td>
<td>65</td>
</tr>
<tr>
<td>008AH</td>
<td>66</td>
</tr>
<tr>
<td>008BH</td>
<td>67</td>
</tr>
<tr>
<td>008CH</td>
<td>68</td>
</tr>
<tr>
<td>008DH</td>
<td>69</td>
</tr>
<tr>
<td>008EH</td>
<td>6A</td>
</tr>
<tr>
<td>008FH</td>
<td>6B</td>
</tr>
<tr>
<td>0090H</td>
<td>6C</td>
</tr>
<tr>
<td>0091H</td>
<td>6D</td>
</tr>
<tr>
<td>0092H</td>
<td>6E</td>
</tr>
<tr>
<td>0093H</td>
<td>6F</td>
</tr>
<tr>
<td>0094H</td>
<td>70</td>
</tr>
<tr>
<td>0095H</td>
<td>71</td>
</tr>
<tr>
<td>0096H</td>
<td>72</td>
</tr>
<tr>
<td>0097H</td>
<td>73</td>
</tr>
<tr>
<td>0098H</td>
<td>74</td>
</tr>
<tr>
<td>0099H</td>
<td>75</td>
</tr>
<tr>
<td>009AH</td>
<td>76</td>
</tr>
<tr>
<td>009BH</td>
<td>77</td>
</tr>
<tr>
<td>009CH</td>
<td>78</td>
</tr>
<tr>
<td>009DH</td>
<td>79</td>
</tr>
<tr>
<td>009EH</td>
<td>7A</td>
</tr>
<tr>
<td>009FH</td>
<td>7B</td>
</tr>
<tr>
<td>00A0H</td>
<td>7C</td>
</tr>
<tr>
<td>00A1H</td>
<td>7D</td>
</tr>
<tr>
<td>00A2H</td>
<td>7E</td>
</tr>
<tr>
<td>00A3H</td>
<td>7F</td>
</tr>
<tr>
<td>00A4H</td>
<td>80</td>
</tr>
<tr>
<td>00A5H</td>
<td>81</td>
</tr>
<tr>
<td>00A6H</td>
<td>82</td>
</tr>
<tr>
<td>00A7H</td>
<td>83</td>
</tr>
<tr>
<td>00A8H</td>
<td>84</td>
</tr>
<tr>
<td>00A9H</td>
<td>85</td>
</tr>
<tr>
<td>00AAH</td>
<td>86</td>
</tr>
<tr>
<td>00ABH</td>
<td>87</td>
</tr>
<tr>
<td>00ACH</td>
<td>88</td>
</tr>
<tr>
<td>00ADH</td>
<td>89</td>
</tr>
<tr>
<td>00AEH</td>
<td>90</td>
</tr>
<tr>
<td>00AFH</td>
<td>91</td>
</tr>
<tr>
<td>00B0H</td>
<td>92</td>
</tr>
<tr>
<td>00B1H</td>
<td>93</td>
</tr>
<tr>
<td>00B2H</td>
<td>94</td>
</tr>
<tr>
<td>00B3H</td>
<td>95</td>
</tr>
<tr>
<td>00B4H</td>
<td>96</td>
</tr>
<tr>
<td>00B5H</td>
<td>97</td>
</tr>
<tr>
<td>00B6H</td>
<td>98</td>
</tr>
<tr>
<td>00B7H</td>
<td>99</td>
</tr>
<tr>
<td>00B8H</td>
<td>100</td>
</tr>
<tr>
<td>00B9H</td>
<td>101</td>
</tr>
<tr>
<td>00BAH</td>
<td>102</td>
</tr>
<tr>
<td>00BBH</td>
<td>103</td>
</tr>
<tr>
<td>00BCH</td>
<td>104</td>
</tr>
<tr>
<td>00BDH</td>
<td>105</td>
</tr>
<tr>
<td>00BEH</td>
<td>106</td>
</tr>
<tr>
<td>00BFH</td>
<td>107</td>
</tr>
<tr>
<td>00C0H</td>
<td>108</td>
</tr>
<tr>
<td>00C1H</td>
<td>109</td>
</tr>
<tr>
<td>00C2H</td>
<td>110</td>
</tr>
<tr>
<td>00C3H</td>
<td>111</td>
</tr>
<tr>
<td>00C4H</td>
<td>112</td>
</tr>
<tr>
<td>00C5H</td>
<td>113</td>
</tr>
<tr>
<td>00C6H</td>
<td>114</td>
</tr>
<tr>
<td>00C7H</td>
<td>115</td>
</tr>
<tr>
<td>00C8H</td>
<td>116</td>
</tr>
<tr>
<td>00C9H</td>
<td>117</td>
</tr>
<tr>
<td>00CAH</td>
<td>118</td>
</tr>
<tr>
<td>00CBH</td>
<td>119</td>
</tr>
<tr>
<td>00CCH</td>
<td>120</td>
</tr>
<tr>
<td>00CDH</td>
<td>121</td>
</tr>
<tr>
<td>00CEH</td>
<td>122</td>
</tr>
<tr>
<td>00CFH</td>
<td>123</td>
</tr>
<tr>
<td>00D0H</td>
<td>124</td>
</tr>
<tr>
<td>00D1H</td>
<td>125</td>
</tr>
<tr>
<td>00D2H</td>
<td>126</td>
</tr>
<tr>
<td>00D3H</td>
<td>127</td>
</tr>
<tr>
<td>00D4H</td>
<td>128</td>
</tr>
<tr>
<td>00D5H</td>
<td>129</td>
</tr>
<tr>
<td>00D6H</td>
<td>130</td>
</tr>
<tr>
<td>00D7H</td>
<td>131</td>
</tr>
<tr>
<td>00D8H</td>
<td>132</td>
</tr>
<tr>
<td>00D9H</td>
<td>133</td>
</tr>
<tr>
<td>00DAH</td>
<td>134</td>
</tr>
<tr>
<td>00DBH</td>
<td>135</td>
</tr>
<tr>
<td>00DCH</td>
<td>136</td>
</tr>
<tr>
<td>00DDH</td>
<td>137</td>
</tr>
<tr>
<td>00DEH</td>
<td>138</td>
</tr>
<tr>
<td>00DFH</td>
<td>139</td>
</tr>
<tr>
<td>00E0H</td>
<td>140</td>
</tr>
<tr>
<td>00E1H</td>
<td>141</td>
</tr>
<tr>
<td>00E2H</td>
<td>142</td>
</tr>
<tr>
<td>00E3H</td>
<td>143</td>
</tr>
<tr>
<td>00E4H</td>
<td>144</td>
</tr>
<tr>
<td>00E5H</td>
<td>145</td>
</tr>
</tbody>
</table>
MOD FINALMODULE
39FH SYM MEMORY
38DH SYM W
0DDH SYM LAST
0E9H SYM GOOD
0E2H SYM FIFR
0E4H SYM POOR
38DCH SYM TIMECREDIT
38DDH SYM N
0E7EH SYM COMMENT
0ED5H SYM TIMEXP
38E2H SYM DECAYR
38E8H SYM N
38E3H SYM M
3F62H SYM COMPOSITE
0E66H SYM CPU
38E9H SYM OVERALL
38E6H SYM DECNUM
0E7EH LIN 9
0E7EH LIN 10
0E89H LIN 12
0E8FH LIN 13
0E94H LIN 14
0E97H LIN 15
0E92H LIN 17
0E98H LIN 18
0EADH LIN 19
0EB0H LIN 20
0EBBH LIN 22
0EC1H LIN 23
0EC6H LIN 24
0EC9H LIN 26
0ECFH LIN 27
0ED4H LIN 28
0ED4H LIN 29
0ED5H LIN 30
0EDFH LIN 32
0EDFH LIN 33
0EF5H LIN 34
0F12H LIN 35
0F24H LIN 36
0F2EH LIN 37
0F30H LIN 38
0F4EH LIN 39
0F5AH LIN 40
0F5EH LIN 41
0F61H LIN 42
0F62H LIN 43
0F62H LIN 45
0FBBH LIN 46
0FC1H LIN 47
0FC7H LIN 49
0F06H LIN 50
0FE4H LIN 51
0FF1H LIN 52
0FF8H LIN 53
0FF8H LIN 54
0FF0H LIN 55
002H LIN 56
007H LIN 57
MOD INTERRUPT

C-11
7FH SYM MEMORY
100FH SYM INTERRUPTROUTINE
100H LIN 3
100H LIN 4
101H LIN 5
101H LIN 6

MEMORY MAP OF MODULE UIWT
READ FROM FILE :F1:UIWT TMP
WRITTEN TO FILE :F1:UIWT
MODULE START ADDRESS 0001H

START STOP LENGTH REL NAME
   0H   0H   1H A  ABSOLUTE
   1H 10AFH 10AFH A  ABSOLUTE
13F H 13FEH  3H A  ABSOLUTE
INTER-MODULE CROSS-REFERENCE LISTING

**NAME**  **ATTRIBUTES**  **MODULE NAMES**

NAME STRUCTURE(5); RESULTSMODULE
COATIME ADTRE:1; RESULTSMODULE RESULTMODULE
ITDUMP PROCEDURE: CONSOLEDUMUBE STARTUPMODULE
IN PROCEDURE BYTE; CONSOLEDUMUBE TESTPROCEDURE STARTUPMODULE
LOOKREAD PROCEDURE: ADDRESS TESTPROCEDURE RESULTSMODULE MAINUNITMODULE
COMMENT PROCEDURE: RESULTSMODULE RESULTSMODULE
COMPOSTE PROCEDURE: RESULTSMODULE RESULTSMODULE
OUT PROCEDURE: CONSOLEDUMUBE TESTPROCEDURE MAINUNITMODULE RESULTSMODULE

STARTUPMODULE FINALMODULE

DATE BYTE(12); STARTUPMODULE RESULTSMODULE
HECIAL BYTE(3); RESULTSMODULE CONSOLEDUMUBE STARTUPMODULE
ELATIME ADDRES: TESTPROCEDURE MAINUNITMODULE
ONE PROCEDURE: TESTPROCEDURE TESTMODULE
RIL PROCEDURE: TESTPROCEDURE
ILE BYTE TESTPROCEDURE RESULTSMODULE STARTUPMODULE FINALMODULE
IRIS SHOT BYTE(3); RIFLEDATAMODULE MAINUNITMODULE STARTUPMODULE
PTO BYTE RIFLEDATAMODULE
ETRIFLEDATA PROCEDURE; TESTPROCEDURE MAINUNITMODULE
MEETING PROCEDURE; CONSOLEDUMUBE MAINUNITMODULE
ISTORY BYTE(3); STARTUPMODULE RIFLEDATAMODULE
IWS PROCEDURE; STARTUPMODULE
OFLAG BYTE; STARTUPMODULE RESULTSMODULE
NUMER BYTE(1); STARTUPMODULE RESULTSMODULE
INTERR': ROUTINE PROCEDURE, INTERRUPT1 STARTUPMODULE
TEST PROCEDURE; TESTPROCEDURE TESTMODULE
STIME PROCEDURE; TESTPROCEDURE TESTMODULE
STIME TE BYTE; TIMERODULE
STIME TE BYTE; TIMERODULE
NAME STRUCTURE(5); STARTUPMODULE RESULTMODULE
CARPES ES BYTE; RESULTSMODULE FINALMODUe
FARTSET PROCEDURE; RESULTSMODULE TESTPROCEDURE
PRESENT RESULTS PROCEDURE; RESULTSMODULE MAINUNITMODULE
INT PROCEDURE; CONSOLEDUMUBE TESTPROCEDURE RESULTSMODULE BEGIN MODUe STARTUPMODULE FINALMODULE
TANUM PROCEDURE; CONSOLEDUMUBE TESTPROCEDURE RESULTSMODULE
FINT PROCEDURE; ** UNRESOLVED ** TESTPROCEDURE
HESLSON BYTE; MAINUNITMODULE STARTUPMODULE
RIFLE BYTE RIFLEDATAMODULE RESULTSMODULE MAINUNITMODULE
RIFLESH PROCEPUC BR; RIFLEDATAMODULE MAINUNITMOUND
TEST PROCEDURE; ** UNRESOLVED ** TESTPROCEDURE
LTIM PROCEDURE; ** UNRESOLVED ** TESTPROCEDURE
OFLR STRUCTURE(5); RIFLEDATAMODULE RESULTSMODULE STARTUPMODULE FINALMODULE
DODAT PROCPUCURE; STARTUPMODULE RIFLEDATAMODULE MAINUNITMODULE
HOFLEYTE BYTE; RIFLEDATAMODULE STARTUPMODULE
HEDEND BYTE; RIFLEDATAMODULE MAINUNITMODULE
STTRUCU(5); RIFLEDATAMODULE RESULTSMODULE STARTUPMODULE FINALMODULE
THIS PROGRAM WAS WRITTEN BY M.C. TOWLE IN THE WINTER AN SPRING OF 1978
IT ASSUMES THE SYSTEM HAS BEEN RESET PRIOR TO RUNNING
THIS MODIFICATION WHICH RESULTS IN A CHANGE TO UMT OPION 1.2
RESULTS FROM THE REQUIREMENT TO SHOW THE MOST SHOTS!

1) MATHEATWIMODULE
   DO.

2) SHOWHORST: PROCEDURE EXTERNAL.
   2) END SHOWHORST

4) DECLARE 'LEAPLAPMصومLATOR BYTE A (8) DAH+1(DAH):- EXCLUSIVE OR
   ACCUMULATOR WITH ITSELF/*

5) TXSRY: PROCEDURE EXTERNAL; /* CHECKS U-A RT TRANSMIT BUFFER FOR "EMPTY" /*
6) END TXSRY.

7) VOTES: PROCEDURE EXTERNAL; /* OTTRAX RESET SEE CONSOL MODULE /*
8) END VOTES.

9) COT: PROCEDURE (ITEM) EXTERNAL; /* SETS A BIT IN THE HIGH USART */
10) DECLARE ITEM BYTE.
11) END COT.

12) DECLARE USARTCONTROL LITERALLY 'REDH'; /* ADDRESS OF U-A RT CONTROL P 3-32 */
    USART:RESET LITERALLY '48H'; /* RETURNS 8251 TO MODE INST FORMAT
    REF PAGE 3-13 */

13) STARTUP: PROCEDURE EXTERNAL; /* ET I/O FOR RIFL ETC */
14) END;

15) TARGETAVAILABLE: PROCEDURE BYTE EXTERNAL; /* CHECKS FOR AN IR SPOT
    IT WILL RETURN 1 IF IR SPOT IS FOUND */
16) END;

17) START: PROCEDURE EXTERNAL; /* START 8253 REGISTERS */
18) END;

19) CLOCKREAD: PROCEDURE ADDRESS EXTERNAL; /* READS CLICK FOR TARGET # SHOTS */
20) END;

21) RIFLESHOT: PROCEDURE BYTE EXTERNAL; /* WAITS FOR ANY RIFLE SHOT */
22) END;

23) GETRIFLEDATA: PROCEDURE EXTERNAL; /* READ & RECORD INPUT BYTE */
24) END;

26) SETDATA: PROCEDURE (POINTER, LENGTH VALUE) EXTEMAL

27) END SETDATA.
PRESENTRESULTS: PROCEDURE EXTERNAL; /* OUTPUTS DATA TO CONSOLE */
END;

DECLARE T.CHECK BYTE EXTERNAL;

TEST: PROCEDURE EXTERNAL;
END TEST;

DECLARE FOREVER LITERALLY 'WHILE 1';
DECLARE COUNTERSET LITERALLY '0000';
DECLARE (TRAINING, 8) BYTE;
DECLARE T.RAIN BYTE PUBLIC AT (TRAINING);
DECLARE FILED AT SHOOT PROCEDURE EXTERNAL; /* IDENTIFIES WHICH RIFLE MISSED A TARGET SHOT OPPORTUNITY */
END;

DECLAR 1 80 TO COMS & LE;
DECLAR 1 80 TO H (K BYTE EXTERNAL;
DECLAR 1 80 TO FILED;
TEST: PROCEDURE;
IF TARGETDOWNTIME := CLOCKREAD THEN
DELTA TIME = TARGETDOWNTIME - CLOCKREAD;
ELSE
DELTA TIME = 1 + COUNTERSET + TARGETDOWNTIME - CLOCKREAD;
IF DELTA TIME > 200 THEN
TURKEY = 1;
ELSE TURKEY = 0;
END TURKEYTEST;

DECLAR DELTA TIME ADDRESS EXTERNAL;
DECLAR FIRST SHOT BYTE EXTERNAL;
DECLAR TARGETDOWN TIME ADDRESS PUBLIC;
DECLAR TARGETADDRESS/PREDICTED, TFLAG, DEADGONE BYTE PUBLIC;
DECLAR TURKEY BYTE PUBLIC;
DECLAR TARGET TIME ADDRESS PUBLIC;

END OF DECLARATIONS */

PROGRAM STARTS ***********

TESTBOARDCHECK DO;
IF TS.CHECK THEN /* IF ROM EXTENDER BOARD IS NOT PRESENT, THE 00/20 TIMES OUT AND WILL READ 00 INTO THE ACCUMULATOR */
CALL TEST; /* PERFORM 00/20 BOARD CHECK */
END TESTBOARDCHECK;

INITIALIZE
CALL TIME.RESTART;
CALL STARTUP;
CALL GREETING;
CALL TYPE; /* INSURE "GREETING" HAS BEEN COMPLETED */
CALL VOTES: // Switch to V:TRAX line & change to 9000 baud //

TARGET: #WANTIME = CLOCK$READ + 200: // For TURKEY TEST //

SESSION
   DO WHILE TRAINING: // We will end with an "ESCAPE" from CONSOLE //
      IF TARGET AVAILABLE OR RIFLE SHOT THEN
      ACTION: DO: // Either a rifle shot or a target available //
      IF TARGET AVAILABLE OR
      IF RIFLE SHOT THEN
      If GOTOONE: DO;
      IF NOT TFLAG THEN
      NEWDRONE: IF: // A NEW TARGET HAS APPEARED //
      ARGINTIME = CLOCK READ);
      REALGONE = 0;
      TFLAG = 0;
      TURKEY = 0;
      END NEWDRONE;
      IF RIFLE SHOT THEN
      GOODDATA: CALL GET RIFLE DATA;
      END GOTOONE;
      ELSE // ELSE THERE IS NO TARGET, BUT A SHOT WAS FIRED //
      NO TARGET: DO:
      CALL TURKEY TEST;
      CALL GETritable DATA;
      END NONTARGET;
      END ACTION;
      ELSE // ELSE THERE IS NEITHER A TARGET NOR SHOT //
      NO ACTION: DO;
      IF TFLAG THEN // THAT IS: THERE WAS A TARGET ON LAST PASS //
      TURKEY: DO;
      TARGETDOWNTIME = CLOCK READ;
      TFLAG = 0;
      END TURKEY;
      IF NOT REALGONE THEN
      IF REALGONE THEN
      WATRONE: DO;
      CALL TURKEY TEST: // We will hold REALGONE=0 for one second //
      IF TURKEY THEN // We are "TARGET IGNORED" only if REALGONE=1 //
      RECORDING: DO:
      RECORDING = 1;
      CALL WHO FAILED SHOOT;
      CALL SET DATA: FIR (SHOOT, 5, 1): // Next shot at a TARGET WILL BE TIMED //
      CALL SHOW WORST;
      END RECORDING;
      END WATRONE;
      END NO ACTION;
      END SESSION;
      END NONACTION;
      END TURKEY;
      END TURKEY TEST;
      END NO ACTION;
      END SESSION;
      END NO ACTION;
      END SESSION;
      END SESSION;
      END SESSION;
      CALL PRESENT RESULTS.
      CALL WATRONE.
      OUTPUT (USAPRT CONTROL) = 15620 RESET;
      END UNT PROGRAM;
PL/I-80 COMPILED

10: 1      END MAINAINTAMODULE;

MODULE INFORMATION:

  CODE AREA SIZE   = 8K00CH  2550
  VARIABLE AREA SIZE = 8K00AH  180
  MAXIMUM STACK SIZE = 8K004H  60
  167 LINES READ
  0 PROGRAM ERROR(S)

END OF PL/I-80 COMPILATION
I SIS-1 PL/M-80 V3 1 Compilation of Module Start-Up Module
Object Module placed in .F1:START OBJ
Compiler invoked by: PLX80 .F1:START PLM ...DELUG date (25 Jun 79)

#INITVECTOR
1 STARTUP*MODULE DO;
2 1 BITDUMP PROCEDURE EXTERNAL; /* IN CONSOLEMODULE */
3 2 END BITDUMP;
4 1 DECLARE INTERRUPT7 LITERALLY '13FCH';
   CALLIT LITERALLY '0C3H';
5 1 DECLARE INTERRUPTCALL STRUCTURE (JUMP BYTE, WHERE TO ADDRESS) AT (INTERRUPT7)
   DATA (CALLIT), INTERRUPTROUTINE ;
6 1 DECLARE LIT LITTALLY 'LITALLY'.
   RESET#12 LIT '0FH'.
   ENABLE#11 LIT '1FH'.
   PORT3 LIT '0EH'.
   PORT6 LIT '0EH'.
   GROUP1 LIT '07H'; /* BBS  CONTROL REF. PAGE 3-68 OF 88/24 MANUAL */
   WORD# LIT '92H'; /* PORT 1 & 2 INPUT, 3 OUTPUT ALL NODE @ PAGE 4-13 --- */
   GROUP2 LIT '0BH'; /* PAGE 3-68 */
   WORD# LIT '89H'; /* PORTS 4, 5 & 6 ALL INPUTS NODE @ */
   USR#1 CONT'L LIT '0EH'; /* PAGE 4-48 */
   USR#1 RESET LIT '0BH'; /* INTERNAL RESET PAGE 3-43 */
   MODE# SET LIT '0EH', /* SETS 2 STOP BITS, 8 BITS, 16X PAGE 3-38, -41 & 4-48 */
   /* USE #1 FOR 1 STOP BIT, 8 BIT WORD & 64X */
   COMMAND# LIT '27H'; /* SETS TRANSMIT/RECEIVER READY.
   DATA TERMINAL READY; REQUEST TO SEND */
   /* SEE PAGE 3-43 (AND STEP 48 OF DOS MONITOR SHADOW PROM) */
7 7 DECLARE DECIMAL(3) BYT EXTERNAL;
8 DECLARE HISTORY(5) BYTE PUBLIC; /* 1-15 WILL
   IDENTIFY WHICH RIFLE WAS SHOT AT A PARTICULAR TARGET */
9 9 DECLARE SCORE(5) STRUCTURE (MISS BYTE, HIT BYTE, LOW BYTE, LOW LEFT BYTE,
   MIDDLE BYTE, HIGH BYTE, HIGH LEFT BYTE,
   LEFT BYTE, LOW LEFT BYTE, ERROR BYTE, TURN BYTE,
   TARGET ignored BYTE) EXTERNAL;
10 DECLARE (SHOTS#16, SHOTS#16, REALGONE) BYTE EXTERNAL,
   FIRSTSHIT(5) BYTE EXTERNAL,
   SFSHIT(5) STRUCTURE (SHOTS BYTE, TIME OF SHOT ADDRESS) EXTERNAL;
11 11 DECLARE TRAY BYTE EXTERNAL;
12 12 DECLARE DIAL#1 LITERALLY '0C3H', /* REF PAGE 3-180 */
   DIAL#2 LITERALLY '0D3H';
13 DECLARE DIAL#3 LITERALLY '07FH';
14 DECLARE CRFL LIT '10H, 08H';
   NAME(3) STRUCTURE (LETTER(9) BYTE) PUBLIC.
   DATE(1) BYTE PUBLIC.
IDNUMBER (7) BYTE PUBLIC;
PROMPT LIT 'EHE';
WHEN (16) BYTE DATA(15, 'TOO+"S DATE?", CR LF),
UNITID (21) BYTE DATA (26, UNIT/SWAT VER 1.3", CR LF), /* CHANGED 6/25/79 */
/ TIMES FOR "FAST", "SLOW", ETC IN "FINAL PLM" WERE INCREASED */,
/ AT ALL MARSHALL'S INSTRUCTIONS. A CHANGE IS ALSO BEING MADE */,
/ IN THE "SHOWANDRIP" PROCEDURE IN RIFLE PLM TO AVOID THE */,
/ PROBLEM WHERE ONLY FOUR RIFLES ARE SHOOTING => RIFLE 85 IS */,
/ THE WORST BECAUSE OF ALL THE TARGET IDENTIFIED & THERE IS NO */,
/ RIFLE 05 "WORST-RIFLE" LIGHT */,
IDENT (19) BYTE DATA (18, 'EXERCISE NUMBER?", CR LF),
QUERY (22) BYTE DATA (26, 'WANT ID? YES OR NO?", CR LF),
TRAIINERS (26) BYTE DATA (25, 'ENTER NAMES OF TRAINEES", CR LF),
ONRIFLE (10) BYTE DATA (5, ON RIFLE '),
(X, I, J, K, Nk) BYTE,
IDFLAG BYTE PUBLIC;

15 1 TTYSET: PROCEDURE EXTERNAL;
16 2 END TTYSET; /* USED ONLY WHEN USART IS IN RESET CONDITION */
17 1 CIN: PROCEDURE BYTE EXTERNAL;
18 2 END CIN;
19 1 COUT: PROCEDURE ITEM) EXTERNAL;
20 2 DECLARE ITEM BYTE;
21 2 END COUT;
22 1 PORTSET: PROCEDURE PUBLIC.
23 2 OUTPUT(GROUP1)=WORD1; /* SET CONTROL WORD INTO GROUP 1 I/O PORTS */
24 2 OUTPUT(GROUP2)=WORD2; /* SET CONTROL WORD INTO GROUP 2 I/O PORTS */
25 2 END PORTSET;
26 1 PRINT: PROCEDURE (PTR) EXTERNAL;
27 2 DECLARE PTR ADDRESS;
28 2 END PRINT;
29 1 SETDATA: PROCEDURE (POINTER, LENGTH, VALUE) PUBLIC;
30 2 DECLARE (POINTER) FINAL ADDRESS,
31 2 (LENGTH, VALUE SET BASED POINTER) BYTE;
32 2 FINAL = POINTER + LENGTH - 1;
33 2 LOOP: DO WHILE POINTER <= FINAL;
34 3 SET = VALUE;
35 3 POINTER = POINTER+1;
36 3 END LOOP;
37 2 END SETDATA;
38 1 INTERRUPT:ROUTINE PROCEDURE INTERRUPT 7 EXTERNAL;
39 2 END INTERRUPT:ROUTINE;
40 1 UP1STROBE: PROCEDURE EXTERNAL;
41 2 END UP1STROBE;
42 1 SIMULATE: RIFLES: PROCEDURE;
43 2 OUTPUT(PORT6)=004; /*ELL UP1-1 TO SIMULATE RIFLE DATA */
44 2 END SIMULATE: RIFLES;

C-20
45 DECLARE CTRL LIT 14W, PNAME TYPE BYTE;

/******* END OF DECLARATIONS ***********/

46 STARTUP: PROCEDURE PUBLIC;

47 CALL SETDIAK: SCOPE, 65.8);
48 CALL SETDIAK: HISTORY, 5.0);
49 CALL SETDIAK: FIRSTSHOT, 5.1);
50 CALL SETDIAK: SPEED, 15.0);
51 CALL SETDIAK: DECIMAL, 3.0);
52 T HANDLE = 0;
53 TRAIN = 1;
54 TFLAG = 0;
55 SHOTFLAG = 0;
56 FILE = 1; /* Whoshot first increments file: N005 (divides) by 5, 
I.e. 5/5=1 with remainder = 0. Now, rifle = file+1,
so we shoot with rifle # 11 */
57 REMINDER = 1; /* Target has not been available for over 1 sec */
58 CALL PORTRESET;
59 OUTPUT(PORT3) = NOT_RESET&8212; /* Will clear all 8212 data 
latches following strobe */
60 OUTPUT(PORT6) = ENABLES&11; /* The leading edge of the strobe */
61 OUTPUT(PORT6) = 0; /* The trailing edge */
62 DO JK = 9 TO 13; /* Clears the JK FF data latches */
63 OUTPUT(PORT3) = NOT JK AND 0FH;
64 OUTPUT(PORT16) = ENABLES&11;
65 OUTPUT(PORT16) = 0;
66 EM;
67 DISABE;
68 CALL TTYRESET;
69 SETINT: OUTPUT(AD1CM) = (LUMH(INTERRUPT) AND BE8H) & 1FH;
70 OUTPUT(AD1CM) = (HIGH INTERRUPT); 
71 OUTPUT(AD1CM) = 0CH; /* Mask all but interrupt 7 */

/* We now obtain identification data for the session, if needed */
72 CALL COUNT(2DM); /* CP */
73 CALL COUNT(2BH); /* LF */
74 CALL COUNT(2BH); 
75 CALL PRN(UNIT&ID);
76 CALL COUNT(2BH); 
77 CALL PRN(QUERY); 
78 CALL OUTPUT(PROMPT); 
79 X=CIN;
80 RUNTYPE = X; /* Save input for identification of control T */
81 CALL COUNT(X);

C-21
CALL COUT(ODH);
CALL COUT(ODH); /* CR.LF */
IDFLAG=0; /* CONTROL FOR ID PRINT-OUT AT END OF SESSION */

IF X = 'Y' THEN
GETID: DO:
IDFLAG=1;

CLRM8TE: CALL SETDATE(0,12,0DH); /* CLEAR "DATE" */
CALL PRINT( WHEN);
CALL COUT(PROMPT);
DATE(0)=11;

GETDATE:
DO I=1 TO 9;
X=CIN;
IF X=6OH THEN I=9;
CALL COUT(X);
DATE(1) = X;
END GETDATE;

CALL COUT(0DH);
DATE(11) = 0AH; /* LF */
CALL COUT(0DH);

CALL SETDATE( IDNUMBER,7,0DH); /* FILLS "ID NUMBER" WITH (R) */
CALL PRINT( IDENT);
CALL COUT(PROMPT);
CALL BITDUMP; /* CLEAR OUT USART INPUT BUFFER */
IDNUMBER(8)=6.
GETIDNUMBER: DO I=1 TO 4. /* A 4 PLACE "ID" */
X=CIN;
IF X=6OH THEN I=4;
CALL COUT(X);
IDNUMBER(I)=X;
END GETIDNUMBER;

CALL COUT(0DH);
IDNUMBER(6)=0AH;
CALL COUT(0DH);

CALL SETDATE( NAME,45,0DH); /* FILLS NAME MATRIX WITH CR */
CALL PRINT( TRAINEES);
GETTRAINEES:
DO I=0 TO 4. /* FOR FIVE TRAINEES */
CALL PRINT( TRAINEE),
CALL COUT(34)+1).
122 4 CALL (OUT(BDH));
123 4 CALL (OUT(BNH));
124 4 CALL (OUT(PROMPT));
125 4 CALL EDITDUMP;
126 4 NAME(I).LETTER(0)-0.
127 4 ONENAME:
128 5 DO J-1 TO 6; /* GETS NAMES UP TO 6 LETTERS LONG */
129 5 X=C[J];
130 5 IF X=RDH THEN J=6;
131 5 NAME(I).LETTER(J+X);
132 5 CALL COUT(X);
133 5 END ONENAME;
134 4 CALL COUT(BDH); /* CR */
135 4 NAME(I).LETTER(6)-0AH; /* LF */
136 4 CALL COUT (BNH);
137 4 END GETNAMES;
138 3 END EDITID;
139 2 IF RUNTYPE = CTRL THEN
140 2 CALL SIMULATE_RIFLES;
141 2 END STARTUP;
142 1 END STARTUP_MODULE;

MODULE INFORMATION:

CODE AREA SIZE = 033FH 031D
VARIABLE AREA SIZE = 0B52H 082D
MAXIMUM STACK SIZE = 0004H 40
248 LINES READ
0 PROG RAW ERROR(S)

END OF PL/M-80 COMPIILATION
1  TIMERMODULE DO;

*/ THIS MODULE SETS THE 8253 MODES AND READS REGISTERS. NOTE THAT ALL
THREE 8253 GATES MUST BE HI'S! ALL PAGE REFERENCES ARE TO THE 806/28
REFERENCE MANUAL 98-317C */

2 1 DECLARE LIT LITERALLY 'LITERALLY';
   COUNTER0 LIT '000H', COUNTER1 LIT '000H';
   CONTROL LIT '000H'; /* SEE PAGE 2-7 */

3 1 DECLARE CHRMODE LIT '34H'; /* 2 BYTES, MODE 2 PAGE 3-76 */
   CHRMODE LIT '74H'; /* 2 BYTES, MODE 2 */
   CHRMODE LIT '06H'; /* 2 BYTES, MODE 3 */

*/ THE FOLLOWING 2-BYTE WORDS ARE THE "BAUD RATE FACTORS" TABLE 4-34 P 4-48 */

4 1 DECLARE LOW0 LIT '0'; /* COUNTER 0 PERIOD IS 5 MILLISECONDS */
   HIGH0 LIT '15H';
   LOAD LIT '3FH'; /* COUNTER 1 PERIOD IS 5 MINUTES */
   HIGH1 LIT 'BEH'; /* = 5 MIN * 60 (SEC/MIN) / 0.0005 SEC -1 IN HEX */
   *********SET UP FOR 300 BAUD************
   LOAD LIT '0EH'; /* COUNTER 2 FREQUENCY IS 4.8 KHZ */
   HIGH2 LIT '80H'; /* SEE PAGE 3-38, AND NOTE THAT THE 8251 IS SET FOR 16X */

   LOWVOTRACK LIT '07H';
   HIGHVOTRACK LIT '08H'; /* SETS VOTRACK OUTPUT TO 9600 BAUD */

5 1 DECLARE TIMEMATCH LIT '00H'; /* A COUNTER 1 LATCH PAGE 3-04 */
   (LS/HIMTIMEBYTE, HS/HIMTIMEBYTE) BYTE PUBLIC;

6 1 DECLARE TIME ADDRESS PUBLIC;
7 1 DECLARE LOWTIMEBYTE BYTE AT (.TIME), HIGHTIMEBYTE BYTE AT (.TIME + 1).

8 1 TIMERSTART Procedure PUBLIC:
9 2 OUTPUT(COUNTER0+CHRMODE); /* SET COUNTERS 0 & 1 MODES */
10 2 OUTPUT(COUNTER0+CHRMODE);
11 2 OUTPUT(COUNTER0+LOW0); /* INITIALIZE COUNTERS */
12 2 OUTPUT(COUNTER0)+HIGH0;
13 2 OUTPUT(COUNTER1)+LOW1;
14 2 OUTPUT(COUNTER1)+HIGH1;
15 2 END TIMERSTART;

16 1 TYTATIMER Procedure PUBLIC:
17 2 OUTPUT(COUNTER1)+CHRMODE;
18 2 OUTPUT(COUNTER2)+LOW2 /* WORDS FOR 300 BAUD */
19 2 OUTPUT(COUNTER2)+HIGH2;
20 2 END TYTATIMER;

21 1 VOTRACKTIMES Procedure PUBLIC; /* SET VOTRACK TO 9600 BAUD */
22 2 OUTPUT(COUNTER1)+CHRMODE;
23 2 OUTPUT(COUNTER2)+LOW2 /* */
24 2 OUTPUT(COUNTER2)+HIGH2;
25 2 END VOTRACKTIMES;

C-24
26 1 CLOCKREAD: PROCEDURE ADDRESS PUBLIC; /* GETS THE CONTENTS OF COUNTER 1 */
27 2 OUTPUT(CONTROL)=TIMELATCH;
28 2 LOWBYTE=INPUT(COUNTER1);  
29 2 HI6HYTE=INPUT(COUNTER1);  
30 2 RETURN TIME;
31 2 END CLOCKREAD;
32 1 END TIMERMODULE.

MODULE INFORMATION:
CODE AREA SIZE  = 0048H  63D
VARIABLE AREA SIZE  = 0004H  4D
MAXIMUM STACK SIZE  = 0000H  60
59 LINES REPRO
0 PROGRAM ERROR(S)
END OF PL-M-O Compiler
RIFLE:MATH:MODULE 00;
/* INCLUDES VARIOUS PROCEDURES FOR RIFLE DATA I/O */

DECLARE MESSAGE BYTE;

COUT: PROCEDURE (ITEM) EXTERNAL;
DECLARE ITEM BYTE;
END COUT;

SCORIT: PROCEDURE (SHOTLOCATION FILE); 
DECLARE (SHOTLOCATION FILE) BYTE;
DECLARE (SHOTLOCATION FILE) ADDER, J (SHOTLOCATION); 
END SCORIT;

/* DECODE TRANSFERS QUADRANT DATA INTO SHOT POSITION. NOTE THAT
ERRORS ARE TREATED AS 'MISSES', THERE IS A 'HIGH-LOW' OR 'LEFT-RIGHT' 
WHERE SHOWN TO OCCUR IS A MISS. TO SCORE AS ERRORS CHANGE THE
SECOND AND THIRD 0 OF DECODE INTO '10' */

DECLARE DECODE (16) BYTE PTR (0.2.4.3.6.0.5.4.8.9.0.2.7.8.6.1);

CLOCKREAD: PROCEDURE ADDRESS EXTERNAL;
END CLOCKREAD;

SETDATA: PROCEUDRE (PTR LENGTH VALUE) EXTERNAL;
DECLARE (LENGTH VALUE) BYTE; PTR ADDRESS;
END SETDATA;

DECLARE FIRSTSHOT (0) BYTE PUBLIC;
DECLARE LIT LITERALLY 'LITERALLY':
HIGH LIT '0E4', LOW LIT '12', RIGHT LIT '2', LEFT LIT '3',
HITAND LIT '4', MISS LIT '5', HOMAND LIT '6', TARGET LIT '7',
SOLDIER LIT '8', YOINAND LIT '9', AND LIT '0';
HANGAND LIT '0H', MAINRAIN LIT '0H', ENEMAND LIT '0H';
AVAILABLE AND LIT '32', FREE AND LIT '0F';
PORT1 LIT 'BE4', /* PPE 3-60 OF 6020 MANUAL */
PORT2 LIT 'BE5H',
PORT3 LIT 'BE6H',
PORT4 LIT 'BE8H',
PORT6 LIT 'BEAH';
ENABLE LIT '18';

DECLARE COUNTERSET LIT 'BEAF'; /* INITIAL VALUE 5 MIN COUNTER */
DECLARE HISTORY (5) BYTE EXTERNAL;
DECLARE (TORSK, TFLAG) BYTE EXTERNAL;
DECLARE SCOPE (5) STRUCTURE (MISS BYTE, HIT BYTE, LOW BYTE, LOWRIGHT BYTE,
RIGHT BYTE, HIGHRIGHT BYTE, HIGH BYTE, HIGHLEFT BYTE,
LEFT BYTE, LOWLEFT BYTE, ERROR BYSTE, TURKEY BYTE,
TARGET (IGNORED BYTE) BYTE PUBLIC;
/* WE WILL USE 'ADDER' TO COMPUTE TOTAL SHOTS */
DECLARE ADDER (5) STRUCTURE (J (13) BYTE) PUBLIC; /* SC */
DECLARE SPEED(5) STRUCTURE(SHOTS BYTE, TIME SUM ADDRESS); PUBLIC.
DECLARE TOTAL TIME ADDRESS;
DECLARE DELTA TIME ADDRESS; PUBLIC;
DECLARE (FILE, RIFLE, FPTR BYTE) PUBLIC;
DECLARE SHOT FLAG BYTE PUBLIC;
DECLARE TARGET TIME ADDRESS EXTERNAL;
DECLARE RIFLE # ID BYTE;
DECLARE CRT STROBE LIT '3001000000';

UP STROBE: PROCEDURE PUBLIC;
OUTPUT(PORT3) = CRT STROBE;
OUTPUT(PORT3) = A;
END UP STROBE;

TARGET AVAILABLE: PROCEDURE BYTE PUBLIC: /* GIVES "TRUE" IF TARGET PRESENT */
DECLARE TARGET BYTE;
TARGET = INPUT(PORT2);
RETURN TARGET: /* PORT 2 BIT NUMBER 0 WILL BE HIGH IF TARGET PRESENT */
END TARGET AVAILABLE;

RIFLE SHOT PROCEDURE BYTE PUBLIC: /* RETURNS "TRUE" IF SHOT FIRED */
SHOT FLAG = INPUT(PORT1); /* SHOTFLAG IS TRUE WHEN SHOT FIRED */
RETURN SHOT FLAG;
END RIFLE SHOT;

GET FILE DATA: PROCEDURE PUBLIC;
DECLARE SHOT DATA BYTE;
UNRESOLVED: DO WHILE SHOT FLAG = INPUT(PORT1);
SHOTS DATA =: /* NEEDED TO ENTER FOLLOWING "WHOSHOT" ROUTINE */
WHOSHOT: DO WHILE NOT SHOT DATA;
FILE = (FILE + 1) MOD 5: /* FILE 0 CONTAINS RIFLE NUMBER 1 DATA.*/
FILE 1 =) RIFLE # 2, ETC. */
RFILE = FILE + 1: /* FILE IS INITIALIZED ONLY ONCE. THIS WE START CHECKING WHERE WE LEFT OFF */
SHOTS DATA = HR(SHOT FLAG, RIFLE);
END WHOSHOT: /* RFILE EQUALS SHOOTING RIFLE NUMBER */
RFILE ID = ROR(RFILE, 3): /* GET RIFLE ID INTO HIGH ORDER BITS */

IF NOT TARGET THEN
DO;
IF FIRST SHOT (FILE) THEN
HOMOQUICK DO;
IF TARGET TIME - CLOCK READ THEN
DELTA TIME = TARGET TIME - CLOCK READ;
ELSE
DELTA TIME = COUNTER ET + 1 - TARGET TIME - CLOCK READ;
SPEQD(FILE) TIMESUM = SPEED(FILE) TIMESUM + DELTA TIME;
SPEDX(FILE) SHOTS = SPEED(FILE) SHOTS + 1;
FIRST SHOT(FILE) = 0;
END HOMOQUICK;
END;

HISTORY (FILE) = 1;

OUTPUT(PORT) = NOT RIFLE AND 0FH: /* SETS 9311 ADDRESS OF /051 LINE @ RIFLE */
OUTPUT(PORT) = EN-BLE1: /* LATCHES SHOT DATA FROM RIFLE INTO BUS */
69 3 SH0T1DAta=INPUT(PORT2); /* READS QUADRANT DATA */
70 3 OUTPUT(PORT6) = 0; /* RETURNS /051 # RIFLE HIGH AND DROPS 8212 FROM BUS */
71 3 OUTPUT(PORT3) = NOT ( 89 OR 800 ); /* ADDRESSES 9311 /031 DATA # RIFLE */
72 3 OUTPUT(PORT6) = ENABLE;
73 3 OUTPUT(PORT6) = 0;
74 3 CALL COUT(BF1H); /* OUTPUT *SYN* MESSAGE TO VORPAX */
75 3 CALL COUT(B35H + FILE); /* RIFLE ADDRESSED */
76 3 IF ( NOT TURKEV) AND TFLAP THEN
77 3 OK1DATA: DO; /* IF# IN NEW SC0RE DATA ASSUME FOR PORT 2 THAT */
78 4 CALL SCORIT(MESS=5 + DEC(ME(SH(9H1DATA,1) AND 8F1H).FILE));
79 4 OUTPUT(PORT6) = N ( RIFLE ID + MESSAGE) AND 1111111B;
80 4 CALL UP#STROKE;
81 4 VORPAXMESS=: CASE MESSAGE;
82 5 CALL COUT(MISS AND);
83 5 CALL COUT(HIT:AND);
84 5 CALL COUT(LOW:AND);
85 5 DO;
86 6 ( CALL COUT(LOW9A));
87 6 CALL COUT(RIGHT:AND);
88 6 END;
89 5 CALL COUT(RIGHT:AND4);
90 5 DO;
91 6 ( CALL COUT(HIGH:AND);
92 6 ( CALL COUT(PIG:AND));
93 5 END;
94 5 CALL COUT(HIGH:AND4);
95 5 DO;
96 6 ( CALL COUT(HIGH:AND);
97 6 ( CALL COUT(lef9:AND));
98 5 END;
99 5 CALL COUT(LEFT:AND4);
100 5 DO;
101 6 ( CALL COUT(DOWN:AND);
102 6 CALL COUT(LEFT:AND4;
103 5 END;
104 4 IF VORPAXMESS GRES;
105 4 END OK1DATA;
106 4 IF TURKEV THEN
107 4 DO;
108 4 TUR:VOK1DAta. SC0R:=(FILE) TURKEV = SCORE(FILE) TURKEV+1;
109 4 OUTPUT(PORT6) = NOT(RIFLE ID + 8B1H) AND 1111111B;
110 4 CALL UP#STROKE; /* Send TURKEV TO CONSOLE CRT */
111 4 CALL COUT(N404H04);
112 4 CALL COUT(TUR5E AND);
113 4 END;
114 3 IF (NOT RF:G) AND (NOT TURN:Y) THEN
115 4 DO;
116 4 SCORE(FILE) MISS = SCORE(FILE) MISS + 1;
117 4 OUTPUT(PT) = NOT(RIFLE@ID + 0CH) AND 11101111B;
118 4 CALL UP:STROBE; /* SENDS "MISS LATE" MESSAGE TO CONSOLE CPT */
119 4 (ALL COUT < MISS); END;
120 4 END;
121 3 (CALL COUT(FF1), /* VOTRAK SIGN-OFF */
122 3 END UNRESOLVED;
123 2 END DET:RIFLEDATA;
124 1 WH:FRIFL:ETO:"SHOOT: PROCEDURE PUBLIC;
125 2 FPTR=0 TO 4;
126 1 RIFLE@ID = POR(FPTR+1, 3);
127 3 IF HISTORY(FPTR)=0 THEN DO:
128 4 SCORE(FPTR) TARGET=0=SCORE(FPTR) TGT:TARGET=0=SCORE(FPTR) TGT:
129 4 OUTPUT(PT) = NOT(RIFLE@ID + 0CH) AND 11101111B;
130 4 CALL UP:STROBE; /* SENDS TARGET IGNORED TO CONSO: CRT */
131 4 CALL COUT(FF1), /* "MISS LATE" MESSAGE */
132 4 CALL COUT(FF1), /* ADDRESS RIFLE NOT SHOOTING */
133 4 CALL COUT(FF1), /* "MISS LATE" MESSAGE */
134 4 CALL COUT(FF1), /* END OF VOTRAK MESSAGE */
135 4 END;
136 3 END;
137 2 IF SET:DATA(HISTORY, 5F) THEN DO:
138 4 CPT.
139 2 END RIFLE=TO:"SHOOT:
140 1 END M:Worst; PROCEDURE PUBLIC;
141 0 DECLARE RD:WORD BYTE,
142 0 DECLARE RD:BYTE(8-byte); BYTE;
143 0 DECLARE RD:BYTE(8-byte); BYTE;
144 0 DECLARE RD:BYTE TO FPTR = 0 TO 4;
145 0 DECLARE FPTR = 0;
146 0 IF SPECI(FPTR) SHOTS@ THEN /* AVOIDS "WORST-SHOTER" IF NO RIFLE AT FPTR */
147 0 RF(Worst):FPTR) = SCORE FPTR - MISS + SCORE FPTR: TGT: TARGET IGNORED;
148 2 END RD:WORD;
149 0 RD:WORD = 1;
150 0 RIFLE = 1;
151 2 HINT:RD:1, DO FPTR = 0 TO 4;
152 0 IF BAD:WORD(FPTR+1) = BAD:WORD(RIFLE-1) THEN
153 0 BAD:WORD = BAD:WORD OF PTR: 00H, FPTR + 2;)
154 0 IF BAD:WORD(FPTR+1) = BAD:WORD(RIFLE-1) THEN
155 0 DO;
156 4 RIFLE = FPTR + 2;
157 0 BAD:WORD = (BAD:WORD AN 0) OR 00H: RIFLE;
158 4 END;
159 3 END HINT:WORST; /* RIFLE HAS VALUE OF WORST SHOOTER */
160 2 IF (NOT RF:G) AND (NOT TURN:Y) THEN
161 2 END SH:WORST;
162 1 RIFLE=1-APR:MODULE.
PL/I-88 COMPILER

MODULE INFORMATION:

CODE AREA SIZE = 0387H 951D
VARIABLE AREA SIZE = 0065H 1050
MAXIMUM STACK SIZE = 0004H 40
222 LINES READ
0 PROGRAM ERROR(S)

END OF PL/I-88 COMPILATION
ISIS-1 PL/M-3 V3 I COMPIRATION OF MOCLE CONSOLE MODULE
OBJECT MODULE PLACED IN 'F1(CONSOLE OBJ
COMPILER INVOKED BY PLMBB 'F1 CONSOLE PLM DEBUG ENPDATE 12 OCT 78)

1 console module do
* this module contains console I/O Routines */

2 declare usart1 data literally '81Ch', /* page 3-48 of 80/2P manual */
usart1 status literally '06h', /* page 3-44 */
ESC literally '18h', /* ASCII "escape" */
MARK literally '7Fh',
ZEROL literal '3Oh',
CR literally '0Dh', LF literally '0Ah',
enable 19311 literally '18h',
port3 literally '0Eh'
port6 literally '09h',
ttyline literally '08h',
V0TPR1 line literally '09h',
usart1 control literally '06h',

************THE SILENT ?? IS HERE AFTER DEFINED AS A TTY *************/
tty lineage literally '0Ah', /* 1 stop bits, 8 bit, 16x, p. 3-38,41, 4-48 */
v0TPR1:01 mode literally '4Eh', /* DIT */ except 1 stop bit */
usart COMMAND literally '27h', /* DATA TERMINAL READY, ETC */
usart1:01 reset literally '40h',
HELLO (18) byte data (17, CR, LF, LF, "LET"s START:, CR, LF);
declare decimal(3) byte external;
4 declare (1,3,60) byte;

5 TTYtimer PROCEDURE EXTERNAL;
6 end;
7
v0TPR1:01 timer PROCEDURE EXTERNAL;
8 end;
9
10 TTYSET PROCEDURE PUBLIC;
11 output(port1) = TTYLINE; / WILL DIRECT USRT OUTPUT TO TTY */
12 output(port1) = enable 19311;
13 output(port1) = 8;
14 call TTYtimer; /* ETS UP FOR 11 Baud */
15
16 output(usart1:01 control) = TTYLINE;
17 output(usart1:01 control) = usart1:01 command;
18 end TTYSET;
19
20 TTYRES PROCEDURE PUBLIC; /* TTY RES */
21 output(usart1:01 control) = usart1:01 reset;
22 call TTYSET;
23 end TTYRES;
24
25 V0TPR1:01 SET PROCEDURE PUBLIC;
26 output(port1) = V0TPR1:01 mode;
27 output(port1) = usart1:01 reset;
28 end V0TPR1:01 set.

C-31
VOTES: PROCEDURE PUBLIC; /* VOTRAX RESET */
OUTPUT(USARTCON-TX) = USARTRESET;
OUTPUT(PORT3) = VOTRAXLINE;
OUTPUT(PORT6) = ENABLEertoire;
OUTPUT(PORT6) = 0;
CALL VOTRAXTIME;
CALL VOTRAXRESET;
END VOTES;

CIN: PROCEDURE BYTE PUBLIC; /* GETS A BYTE FROM THE CONSOLE */
READY: DO WHILE NOT SHP(INPUT(USARTSTATUS), 1); /* I.E. WHILE INPUT BUFFER IS NOT READY */
END READY;
RETURN MASK AND INPUT(USARTDATA);
END CIN;

TOPD: PROCEDURE PUBLIC;
DO WHILE NOT SHP(INPUT(USARTSTATUS), 2);
END;
END TOPD;

COUT: PROCEDURE (ITEM) PUBLIC; /* OUTPUTS "ITEM" */
DECLARE ITEM: BYTE;
DO WHILE NOT INPUT(USARTSTATUS);
END;
OUTPUT(USARTDATA) = ITEM;
END COUT;

BIT4UMP: PROCEDURE PUBLIC;
Y = INPUT(USARTDATA); /* "Y" HERE IS A BIT-BUCKET */
END BIT4UMP;

PANTHEM: PROCEDURE PUBLIC;
OOS = 0;
DO 1 TO 2,
IF (Y = DECIMAL(1)) O ZERO THEN
GO S = 1;
IF OOS THEN CALL COUT(Y);
END;
CALL COUT(CR);
CALL COUT(LF);
END PANTHEM;

F'INT: PROCEDURE (POINTER) PUBLIC;
DECLARE (POINTER) FINAL ADDRESS,
CHAR BASED POINTER BYTE;
FINAL = POINTE R + CHAR; /* FIRST CHAR IS CHARACTER COUNT */
LOOP: DO WHILE POINTER < FINAL,
POINTER = POINTER + 1;
CALL COUT(CHAR);
END LOOP;
END PRINT;

G'GETING: PROCEDURE PUBLIC;
71 2   CALL PRINT( HELL )
72 2   CALL BITATCH.
73 2   ENABLE.
74 2   END GREETING.

75 1   END: CONSOLE MODULE

MODULE INFORMATION:

CPU AREA SIZE  = 820FH  271D
STACK AREA SIZE = 0000H  0D
MAXIMUM STACK SIZE = 0000H  0D
115 LINES READ
0 PROGRAM ERROR(S)

END OF F-74-80 Compilation
RESULTS@MODULE: 1:

TTYPES: PROCEDURE EXTERNAL;
END TTYPES;

COMPOSITE PROCEDURE EXTERNAL;
END COMPOSITE;

COMMENT: PROCEDURE EXTERNAL;
END COMMENT;

COUT: PROCEDURE (LTR) EXTERNAL;
DECLARE LTR BYTE;
END COUT;

PRINT: PROCEDURE (POINTER) EXTERNAL;
DECLARE POINTER ADDRESS;
END PRINT;

PRINTNUM: PROCEDURE EXTERNAL;
END PRINTNUM;

DECLARE (RIFLE, FILE) BYTE EXTERNAL;
DECLARE LITERALLY '0AH', LF LITERALLY '0AH';
DECLARE SCORE (5) STRUCTURE(MISS BYTE, HIT BYTE, LOW BYTE, LOWRIGHT BYTE, RIGHT BYTE, HIGHLIGHT BYTE, HIGH BYTE, HIGHLF BYTE, LEFT BYTE, LOWLEFT BYTE, ERROR BYTE, TERRY BYTE, TARGET (IGNORED BYTE) EXTERNAL);
DECLARE DECIMAL (3) BYTE PUBLIC;
DECLARE (1, 2) BYTE;
(SUMSHOTS, NEARMISS) BYTE PUBLIC;
DECRLAE ADDER (5) STRUCTURE(l13 BYTE) EXTERNAL;
DECLARE SPEED (5) STRUCTURE(SHOTS BYTE, TIME SUM ADDRESS) EXTERNAL;
DECLARE AVDTIME ADDRESS PUBLIC;

CONVRT: PROCEDURE (HEX);
DECLARE HEX BYTE;
DO 1=0 TO 2;
DECIMAL (2-1) = HEX MOD 16 * 16H;
HEX = HEX / 16;
END;
END CONVRT;

DECLARE RIFLE(8) BYTE DATA(7, 'RIFLE ');
DECLARE TOTALSHOTS(14) BYTE DATA(13, 'TOTAL SHOTS ');
DECLARE PIFLEMHIT(7) BYTE DATA(6, 'HITS ');
DECLARE PIFLEMISS(9) BYTE DATA(8, 'MISSSES ');
DECLARE PIFLELWRIGHT(13) BYTE DATA(12, 'LOW RIGHTS ');

C-34
DECLARE RIFLERIGHT(9) BYTE DATA ('RIGHTS ');
DECLARE RIFLEHIGHT(14) BYTE DATA ('HIGH RIGHTS ');
DECLARE RIFLEHEIGHT(8) BYTE DATA ('HITS ');
DECLARE RIFLEHIGHTLEFT(13) BYTE DATA ('HIGH LEFTS ');
DECLARE RIFLEHEIGHTLEFT(8) BYTE DATA ('LEFTS ');
DECLARE RIFLEHEIGHTLEFT(12) BYTE DATA ('LOW LEFTS ');
DECLARE RIFLEHEIGHTLEFT(8) BYTE DATA ('LEF ');
DECLARE RIFLEHEIGHTLEFT(13) BYTE DATA ('ST ');
DECLARE RIFLEHEIGHTLEFT(8) BYTE DATA ('T ');
DECLARE NAME(5) STRUCTURE (LETTER(9) BYTE) EXTERNAL;
DECLARE DATE(11) BYTE EXTERNAL;
DECLARE IDNUMBER(6) BYTE EXTERNAL;
DECLARE IDFLAG BYTE EXTERNAL;

PRESENTRESULTS PROCEDURE PUBLIC;

DISABLE;

CALL TYPES; /* RESET FOR OUTPUT SEE CONSOL MODE */

CALL PRINT (BLANK);
CALL PRINT (BLANK);

IF IFLAG THEN DO;
CALL PRINT (DATE);
CALL PRINT (IDNUMBER);
EXIT;

CALL PRINT (BLANK);
CALL PRINT (BLANK);

ONE FLEAPRESSULTS GO RIFLE TO 5;

IF EXEC(RIFLE-1) SHOTS C0 0 THEN

CALL PRINT (TYPE1); DO;

CALL PRINT (BLANK);
CALL PRINT (RIFLEXID);
CALL PRINT (RIFLE-1ID);
CALL PRINT (BLANK);
CALL PRINT (BLANK);

IF IFLAG THEN
CALL PRINT (NAME:RIFLE-1));
CALL PRINT (YOURSCORE);
CALL PRINT (BLANK);
CALL PRINT (BLANK);
CALL PRINT (FILE:RIFLE-1)

C-35
77 4 CALL PRINT( TOTALSHOTS);
78 4 SUMSHOTS = 0;
79 4 SUM: DO Z=0 TO 11;
80 5 SUMSHOTS = SUMSHOTS + ADDER(FILE) (Z);
81 5 END SUM;
82 4 NEARMISS = 0;
83 4 SUM: DO Z = 2 TO 9;
84 5 NEARMISS = NEARMISS + ADDER(FILE) (Z);
85 5 END SUM2;
86 4 CALL CONVAT(SUMSHOTS);
87 4 CALL PANTNUM;
88 4 CALL PRINT( RIFLEHIT);
89 4 CALL CONVAT(SCORE(FILE) HIT);
90 4 CALL PANTNUM;
91 4 CALL PRINT( RIFLEMISS);
92 4 CALL CONVAT(SCORE(FILE) MISS);
93 4 CALL PANTNUM;
94 4 CALL PRINT( RIFLEHIT);
95 4 CALL CONVAT(SCORE(FILE) LOW);
96 4 CALL PANTNUM;
97 4 CALL PRINT( RIFLELOWRIGHT);
98 4 CALL CONVAT(SCORE(FILE) LOWRIGHT);
99 4 CALL PANTNUM;
100 4 CALL PRINT( RIFLERIGHT);
101 4 CALL CONVAT(SCORE(FILE) RIGHT);
102 4 CALL PANTNUM;
103 4 CALL PRINT( RIFLEHIGHRIGHT);
104 4 CALL CONVAT(SCORE(FILE) HIGHRIGHT);
105 4 CALL PANTNUM;
106 4 CALL PRINT( RIFLEHIGH);
107 4 CALL CONVAT(SCORE(FILE) HIGH);
108 4 CALL PANTNUM;
109 4 CALL PRINT( RIFLEHIGHLEFT);
110 4 CALL CONVAT(SCORE(FILE) HIGHLEFT);
111 4 CALL PANTNUM;
112 4 CALL PRINT( RIFLELEFT);
113 4 CALL CONVAT(SCORE(FILE) LEFT);
114 4 CALL PANTNUM;
115 4 CALL PRINT( RIFLELOWLEFT);
116 4 CALL CONVAT(SCORE(FILE) LOWLEFT);
117 4 CALL PANTNUM;
118 4 CALL PRINT( RIFLETURKEY);
119  4    CALL COMRT(SCORE(FILE TURKE));
120  4    CALL PRNTNM;
121  4    CALL PRNT(NRLE#TARGET#IGNORED);
122  4    CALL COMRT(SCORE(FILE) TARGET#IGNORED);
123  4    CALL PRNTNM;
124  4    CALL PRNT(NHWM#MC#TARGETS);
125  4    CALL COMRT(SPEED(FILE) SHOT);
126  4    CALL PRNTNM;
127  4    CALL PRNT( AVERAGE#TIME);
128  4    IF (Z = SPEED(FILE) SHOTS) THEN Z=1;
129  4    AVERAGE = (SPEED(FILE) TIME#IN/29)/2;
130  4    CALL COMRT(LWM(AVERAGE#TIME));
131  4    IF(Z = DECIMAL(Z) O:TH THEN CALL COUT(Z);
132  4    CALL COUT(DECIMAL(1))
133  4    CALL COUT(ZERO);
134  4    CALL COUT(0);
135  4    CALL COUT(UNITS);
136  4    CALL COUT(CR);
137  4    CALL COUT(LF);
138  4    CALL COUT;
139  4    CALL PRINT( BLANK);
140  4    CALL COMMENT; /* COMMENT AS TO ACTION TIME */
141  4    CALL PRINT( BLANK).
142  4    CALL COUT;
143  4    CALL COMPO.; /* THE COMPOSITE SCORE IS INITIALLY:
144  4       = 100*(HITS#SHOTS) + 6*(NEAR MISSES#SHOTS) +
145  4       10*(TIME CREDIT FROM PROCEDURE "COMMIT", ABOVE) -
146  4       2*(NUMBER OF TARGETS IGNORED) */
147  4    CALL PRINT( BLANK);
148  4    CALL PRINT( BLANK);
149  4    CALL PRINT( BLANK);
1410  4    END PROFILE;
1411  4    END PROFILE;RESULTS;
1412  4    END RESULTSMODULE.
1413  4    END PROTOCOL;
1414  4    END

INFORMATION

CODE AREA SIZE = 0499H 1'-3D
VARIABLE AREA SIZE = 0089H 10D
MAXIMUM STACK SIZE = 00000H
0
25 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-80 COMPIBATION
FINAL MODULE DO;

1 DECLAIM SPEED(5) STRUCTURE(SHOTS BYTE, TIMERSUM ADDRESS) EXTERNAL,
   (FILE, RIFLE, JUMPSHOTS, NEARMISSES) BYTE EXTERNAL,
   W BYTE, AVGTIME ADDRESS EXTERNAL,
   SCORE(5) STRUCTURE(HIT BYTE, LOW BYTE, LOWRIGHT BYTE,
   RIGHT BYTE, HIGHRIGHT BYTE, HIGH BYTE,
   HIGHLEFT BYTE, LEFT BYTE, LOWLEFT BYTE,
   ERROR BYTE, TURKEY BYTE, TARGET(14) IGNORED BYTE) EXTERNAL,
   FAST(4) BYTE DATA (41. MAN, YOU'RE THE FASTEST SHOT IN THE WEST!),
   GOOD (29) BYTE DATA (24. HEY, YOU'RE PRETTY QUICK!),
   FAIR (3) BYTE DATA (37. OH WELL! THERE'S HOPE IF YOU SPEED UP!),
   POOR (3) BYTE DATA (28. SORRY, BUT YOU'RE PRETTY SLOW!),
   CR LIT-ALLY HIGH. LF LITERALLY 'OH!',
   (TIMESPEED = 3) BYTE.

2 CONT. PROCEDURE (LTR) EXTERNAL;
3 DECLARE LTR BYTE.
4 2 END CONT.

5 PRINT PROCEDURE (POINTER) EXTERNAL;
6 2 DECLARE POINTER ADDRESS.
7 2 END PRINT.

8 COMMENT PROCEDURE PUBLIC
9 2 IF AVGTIME <= 5 THEN
10 2 DO.
11 3 CALL PRINT( FAST).
12 3 TIMESPEED = 3.
13 3 END.
14 2 ELSE IF AVGTIME = 9 THEN
15 2 DO.
16 3 CALL PRINT( GOOD).
17 3 TIMESPEED = 2.
18 3 END.
19 2 ELSE IF AVGTIME = 13 THEN
20 2 DO.
21 3 CALL PRINT( FAIR).
22 3 TIMESPEED = 1.
23 3 END.
24 2 ELSE DO.
25 3 CALL PRINT( POOR).
26 3 TIMESPEED = 0.
27 3 END.
28 2 END COMMENT.

1 PROCEDURE (HEXADR, DECADR) PUBLIC
2 DECLARE(HEXADR, D, RADR) ADDRESS;
   HEX BASED -> ADDR ADDRESS;
   DECIMAL BASED DECADR (+) BYTE;
   (N.N) BYTE.
DO N = 0 TO 4;
MM = MM - 1;
DECIMAL(N) = HEX * 10 - 30H;
HEX = HEX/10;
END;
N = 0;
DO WHILE DECIMAL(N) = 30H AND N < 5:
DECIMAL(N) = 20H;
N = N + 1;
END;
END;

COMPOSITE: PROCEDURE PUBLIC:
DECLARE COMP(24) BYTE DATA(23, 'YOUR OVERALL SCORE IS : '), OVERALL ADDRESS.
DECIMAL(5) BYTE:
OVERALL = 100*(SCORE(FILE). HIT)/SUMSHOTS + 60*MISSHIT/MISSHOTS
+ 100*TIME* CREDIT - 20*SCORE(FILE). TARGET IGNORED;
PRINT (COMP);
IF OVERALL < 0 THEN DO; /* I.E. CHECK FOR NEGATIVE SCORE */
CALL M2656 (OVERALL, DECIMAL).
DO N = 0 TO 4;
CALL COUT(DECIMAL(N));
END;
END;
CALL COUT(CR);
CALL COUT(LF);
END COMPOSITE;
END FINALE MODULE;

MIXILE INFORMATION:
CODE AREA SIZE = 0229H 3551
VARIABLE AREA SIZE = 0018H 16
MAXIMUM STACK SIZE = 0083H 8
85 LINE'S READ
0 PROGRAM ERROR(S)
END OF PL/1/4-0 COMPILATION
ISI 11 PL/1-88  1 Compilation of module INTERRUPT7
OBJECT MODULE PlACED IN F1 INTER-8)
COMPILER INVOKED BY PLM8, F1 IN ER PLM "XREF DEBUG (3 OCT 78)

#NOINTVECTOR

INTERRUPT7 DO.

1 DECLARE TRAP BYTE INTERNAL
ALOC2 LITERALLY '080H': /* ADDRESS TO WHICH WE SEND THE FOLLOWING
NON-SPECIFIC END OF INTERRUPT */
OC2E LITERALLY '280H': /* THE NON-SPECIFIC EO1, SEE PAGE 3-100
AND PAGE 3-100 */

1 INTERRUPT7ROUTINE PROCEDURE INTERRUPT 7 PUBLIC;
2 TrapmO.

2 0 OUTPUT(ADDCM2) = 0 .SIZE;
2 END INTERRUPT7ROUTINE.

1 END INTERRUPT7.

MODULE INFORMATION

IN C AREA SIZE = 00:3H 190
VARIABLE AREA SIZE = 00:00H 0D
MAXIMUM S-1 AREA SIZE = 00:40H 8D
1" LINES 41
0 "PROGRAM = PROG(S)

END OF PL/1-88 IMPLILATION
APPENDIX D

UP-41 PROGRAM
ASSEMBLED LANGUAGE PROGRAM WRITTEN FOR THE UP1-41
UNIVERSAL PERIPHERAL INTERFACE-41 DURING THE
SUMMER TERM OF ACADEMIC YEAR 77-78 BY THOMAS J
RIORDAN
WHILE WORKING AS A GRADUATE ASSISTANT
FOR DR. HERBERT C. TOOLE AT THE NAVAL TRAINING
EQUIPMENT CENTER (NTAC) IN ORLANDO, FLORIDA.

THE PROGRAM ACCEPTS A PARALLEL DATA TRANSFER FROM
AN OUTPUT PORT (8255) OF AN INTEL SBC-1/428-4
SINGLE BOARD COMPUTER SYSTEM. THE DATA WORD IS
DECODED TO OBTAIN A REFERENCE COLUMN ON THE FACE
OF AN LED CRT. THE CRT CURSOR IS THEN POSITIONED
IN THAT COLUMN. THE DATA WORD IS FURTHER DECODED
TO OBTAIN THE ADDRESS IN RUN OF A TEXT STRING WHICH
IS THEN SHIFTED OUT SEQUENTIALLY THROUGH A I/O PORT
LINE OF THE UP1-41 AT 3200 BAUD. THE PROGRAM
IS INTERRUPT-DRIVEN AND UTILIZES A FIFO STACK TO
BALANCE OUT DISPARITIES BETWEEN THE TIMES AT WHICH IT
CAN SHIFT OUT SERIAL DATA AS COMPARED TO THE HIGHEST
POSSIBLE RATE AT WHICH IT MUST ACCEPT PARALLEL DATA.

THE RUNTIME CONFIGURATION OF THE UP1-41 IS AS FOLLOWS:

REGISTER BANK 0

REGISTER 0(R0) 7 BIT ASCII CODE COUNTER
REGISTER 1(R1) ASCII CHAR TO BE OUTPUT
REGISTER 2(R2) COUNT FOR VARIABLE HEIGHT
REGISTER 3(R3) OUTPUT STRING ADDRESS
REGISTER 4(R4) MASK VALUE FROM LOOK-UP TABLE
REGISTER 5(R5) BINARY CODE FOR CURSOR COLUMN POSITION
REGISTER 6(R6) COUNTER FOR STRING OUTPUT
REGISTER 7(R7) PARALLEL DATA TRANSFER

REGISTER BANK 1

REGISTER 0(P0) CURRENT DATA POINTER
REGISTER 1(P1) FINISH DATA POINTER
REGISTER 2(P2) QUEUE STATUS
REGISTER 3(P3) ACCUMULATOR STORAGE
REGISTER 4(P4) UNUSED
REGISTER 5(P5) CONSTANT=193D
REGISTER 6(P6) CONSTANT=224D
REGISTER 7(P7) TEMPORARY DATA WORD STORAGE

PORT 1 SERIAL TRANSMISSION ON BIT 7
PORT 2 LINES 0-4 USED AS A MASK INPUT TO INHIBIT TEXT
STRING OUTPUT LINE 7 USED TO ENABLE CHIP SELECT
I. MCS-4R/UP-41 MACRO ASSEMBLER, V2.0  
18 JAN 79

LOC OBJ  SEQ  SOURCE STATEMENT

53 ;
54 ;
55 ;
56 ;
57 ORG 0
58 JMP INIT ; PRESERVE INTERRUPT VECTORS
59 ORG 3D ; EXTERNAL INTERRUPT VECTOR
60 EXTINT: JMP INROUT ; JUMP TO INTERRUPT ROUTINE
61 ORG 70 ; TIMEP INTERRUPT VECTOR
62 TIMINT: JMP TIMINT ; TIMEP INTERRUPT ROUTINE
63 ORG 100;

64 INIT CPL F0 ; SET FLAG SO INTERRUPTS NOT ENABLED
65 DURING INITIALIZATION ROUTINE
66 REL P2.00E 04 ; NUMBER WILL DRIVE LINES 0-4 TO GROUND IN CASE
67 THE SIMULATION PGM IS GOING TO BE RUN IN E
68 THIS WILL KEEP EACH RIFLE FROM PICKING UP
69 AN EXTRANEOUS SHOT DUE TO THE OUTPUT LINES
70 COMING UP HIGH--08/20 0212 CLEAR WILL NOT
71 HAVE BEEN DONE AT THIS POINT IN TIME--
72 LINES 5 & 6 WILL BOTH BE HIGH AS REQUIRED
73 TO LET AN EXTERNAL SIGNAL CONTROL THE TAR PRES FLAG.
74 BUT WILL NOT ENABLE THE CHIP SELECT WHICH
75 IS TIED TO LINE 7

76 MOV R3.00E A0 ; ASCII CHAIR TO CLEAR CRT SCREEN
77 CALL OUTPUT
78 CALL DELAY
79 CALL LOCSET ; SET JP CRT TO ACCEPT X COORD VALUE
80 MOV P1.0200 H ; VALUE FOR COLUMN 1
81 CALL OUTPUT ROUTINE TO SEND ASCII CHARACTER
82 SEL PB1
83 MOV R0.0320 ; INIT AL FOR READ MEMORY POINTER
84 MOV P1.0320 ; INIT AL FOR WRITE MEMORY POINTER
85 MOV 9900 ; CLEAR QUEUE STATUS REGISTER
86 MOV R6.0240 ; 224 + 32 AVAILABLE LOCATIONS IN RAM
87 MOV R5.0130 ; 193 + 63 (LAST RAM ADDRESS) = 256 =)
88 MOV P5.0120 ; OVERFLOW

89 MOV R1.0010 ; ONE LESS THAN TOP OF RAM
90 MOV P2.0600 ; ENABLE CHIP SELECT
91 EN I ; ENABLE EXTERNAL INTERRUPTS
92 M6.0120 ; GET QUEUE STATUS
93 JZ M6.0120 ; IF QUEUE EMPTY NO ACTION

START: MOV R6.00B0 ; GET DATA FROM RAM LOCATION
94 MOV R7.00A0 ; STORE DATA
95 DEC R2 ; DECREMENT QUEUE STATUS REGISTER
96 MOV R4.0505 ; 193 DECIMAL
97 MOV R4.0505 ; CHECK FOR LAST ACCESS BEING @ TOP OF RAM
98 CONT ; ONE LESS THAN BOTTOM OF RAM
99 INIC P0 ; NEXT RAM ACCESS LOCATION
100 MOV R4.0707 ; RETRIEVE DATA
6085 C5 108    L   R30
6086 AF 109    N  R7,A  STORE DATA
6087 49110    L  R,B0H  SET BIT WHICH IS HARD WIRE LOW
6089 37 111    L  R
608A 86  12    Z CONT
608C 441 13    P  RIFSIM IF CODE FOR RIFLE SIMULATION ROUTINE
9       14    L  P2, INHF IF SIMULATION IS NOT BEING RUN
608E 841 15 CONTIN:  L  P2, INHF IF SIMULATION IS NOT BEING RUN
115    THEN P4T 2 0- MUST BE INPUTS
6048 FF 117    N  R7 RETRIEVE DATA
6041 340 118 CALL  MASK CHECK TO SEE IF OUTPUT DESIRED
6043 86  119    JF0 ESCAPE IF FLUSH, SET WAIT FOR NEW DATA
6045 341 120 CALL  LIMSET SET UP ADD TO ACCEPT X-COORD VALUE
6047 FD 121    MOV  R5 GET RIFLE ID FROM PROCEDURE MASK STORAGE
122    LOCATIONS ON
6048 341 123 CALL  TAB OVER TO LOCATION CORRESPONDS TO RIFLE 0
604A 47  124    MOV  R7 RETRIEVE 08/20 DATA
604B FF 125    MOV  R1, A STORE CODE FOR TYPE OF SHOT
126    IN UPPER 4 BIT TO ALLOW ACCESS TO
127    16 MEMORY LOCATIONS PER SHOT TYPE
604C 531 128    ANL  A 28F:~ MASK OUT LOW OTHER BITS
604D 4B  129    MOV  R1, A STORE RELATIVE ADDRESS OF CHAR STRING
604F E5 130    MOV93 A 0A GET STRING LENGTH
6050 HE1 131    MOV  R6,A STORE COUNTER VALUE
6052 04  132 CALL  STROUT: PROCEDURE TO OUTPUT ASCII STP
6053 04  133 CALL  CALF
6055 05  134 ESCAPE SELECT ON RETURN TO CORRECT REGISTER BANK FOR WAIT LOOP
6056 4E1 135 JMP  WAIT
136
137
138
139
140
6058 05 141 INROUT SELECT  R1 INTERRUPT REG BANK
142    ME  R2,A SAME ACCUMULATOR
143    N  R6, A 2 D
144    N  R1, R2
145    JZ  QUEUE=CHECK FOR QUEUE FULL
146    INP  P2 INCREMENT QUEUE STATUS REGISTER
147    IN  RB8  INPUT DATA
148    IN  RB8  FROM INTERRUPT STORE FF
149    MOV  R6, A NEW DATA
150    MOV  R5, 1 0
151    MOV  R1, A CHECK TO SEE IF STORE WAS
152    IN LAST AVAILABLE RAM LOCATION
153    JZ  INT1 IF NOT THEN CONTINUE
154    MOV  R6, 0 L 0101 BOTTOM OF QUEUE
155 151    INC  R1 INCREMENT WRITE POINTER REG
156 56 QUE=UL: MOV  R3 RESTORE ACCUMULATOR
157 57 RETR RETURN FROM INTERRUPT
158 59 TTIEND: STOP CONT PREVENT FURTHER TIME OVERFLOW
159 59    MOV  RSH THIS SEQUENCE OF OPERATIONS ALLOWS
160 59    DEC  THE RETURN ADDRESS WHICH WAS STORED
161 59    DEC  ON THE STACK TO BE ALL RED SO THAT
162 59    FLH ON RETURN IT WILL NOT CONTINUE IN
163
<table>
<thead>
<tr>
<th>OBJ</th>
<th>OBJ</th>
<th>SOURCE</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>0308</td>
<td>ADD</td>
<td>A, 00</td>
</tr>
<tr>
<td>0210</td>
<td>0908</td>
<td>MOV</td>
<td>R0, A</td>
</tr>
<tr>
<td>0310</td>
<td>1000</td>
<td>INC</td>
<td>R0</td>
</tr>
<tr>
<td>0410</td>
<td>2314</td>
<td>MOV</td>
<td>A, BLW FAIL</td>
</tr>
<tr>
<td>0540</td>
<td>1610</td>
<td>CALL</td>
<td>FAIL</td>
</tr>
<tr>
<td>0693</td>
<td>1639</td>
<td>RETR</td>
<td></td>
</tr>
<tr>
<td>0730</td>
<td>1700</td>
<td>FINISH</td>
<td>MOV</td>
</tr>
<tr>
<td>0810</td>
<td>1710</td>
<td>MOV</td>
<td>R3, A</td>
</tr>
<tr>
<td>0920</td>
<td>1720</td>
<td>MOV</td>
<td>R3, A</td>
</tr>
<tr>
<td>0A30</td>
<td>1730</td>
<td>MOV</td>
<td>R3, A</td>
</tr>
<tr>
<td>0B40</td>
<td>1740</td>
<td>CALL</td>
<td>STRUT</td>
</tr>
<tr>
<td>0C50</td>
<td>1750</td>
<td>CALL</td>
<td>CRIF</td>
</tr>
<tr>
<td>0D63</td>
<td>1760</td>
<td>HERE</td>
<td>JMP</td>
</tr>
<tr>
<td>0E70</td>
<td>1770</td>
<td>RETR</td>
<td></td>
</tr>
<tr>
<td>1F10</td>
<td>2310</td>
<td>SUBROUTINES IN FIRST PAGE OF MEMORY</td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>2420</td>
<td>ORG</td>
<td>2560</td>
</tr>
<tr>
<td>2130</td>
<td>2430</td>
<td>MOV</td>
<td>DB</td>
</tr>
<tr>
<td>2240</td>
<td>2540</td>
<td>MOV</td>
<td>DB</td>
</tr>
<tr>
<td>2350</td>
<td>2650</td>
<td>MOV</td>
<td>R1, A</td>
</tr>
<tr>
<td>2460</td>
<td>2760</td>
<td>MOV</td>
<td>R1, A</td>
</tr>
<tr>
<td>2570</td>
<td>2870</td>
<td>CALL</td>
<td>OUTPUT</td>
</tr>
<tr>
<td>2680</td>
<td>1820</td>
<td>CONT</td>
<td>2 : JUMP IF RIFLE NOT MASKED</td>
</tr>
<tr>
<td>2790</td>
<td>1930</td>
<td>CPL</td>
<td>R0</td>
</tr>
<tr>
<td>28A0</td>
<td>2040</td>
<td>INC</td>
<td>R3, HI</td>
</tr>
<tr>
<td>29B0</td>
<td>2150</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2AC0</td>
<td>2260</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2BC0</td>
<td>2370</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2CD0</td>
<td>2480</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2DE0</td>
<td>2590</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2EE0</td>
<td>26A0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>2FF0</td>
<td>27B0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>30C0</td>
<td>28C0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>31D0</td>
<td>29D0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>32E0</td>
<td>30E0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>33F0</td>
<td>31F0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>34G0</td>
<td>32G0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>35H0</td>
<td>33H0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>36I0</td>
<td>34I0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>37J0</td>
<td>35J0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>38K0</td>
<td>36K0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>39L0</td>
<td>37L0</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3A00</td>
<td>3800</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3B10</td>
<td>3910</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3C20</td>
<td>3A20</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3D30</td>
<td>3B30</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3E40</td>
<td>3C40</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
<tr>
<td>3F50</td>
<td>3D50</td>
<td>MOV</td>
<td>R3, HI</td>
</tr>
</tbody>
</table>

D-4


LOC  Bij  Sig  Source Statement

0129  3  14  MOV  a, 00h  ;GET ASCII CHARACTER
0129  9  15  MOV  a, A
0129  430  16  CALL  OUTPUT
0129  E27  17  DJNZ  F$;STOUTR  ;HAVE F$ CHAR BEEN OUTPUT
012F  3  18  RET  1;RETURN FROM SUBRUTINE
0138  S  19  CALL  OUTPUT
0138  F  20  MOV  a, 008h  ;SERIAL BIT COUNTER
0138  9  21  MOV  a, RL  ;GET ASCII CHARACTER TO BE OUTPUT
0138  400  22  ANL  F$;003h  ;PUT OUT START BIT
0136  404  23  MOV  a, 004h  ;SET UP DELAY LOOP LENGTH
0138  440  24  CALL  CALL
0139  9  25  LOOP1:  OUTL  F$;A  ;OUTPUT CURRENT BIT OF SERIAL CODE
0138  7  26  RF  a  ;GET NEXT BIT OF ASCII CODE
0138  9  27  NOP  ;WAIT 1 INSTRUCTION CYCLE TO COMPENSATE
0138  482  28  MOV  a, 002h  ;SET UP DELAY LOOP LENGTH
0138  448  29  CALL  CALL
0141  C  31  DJNZ  F$;LOOP1  ;TEST FOR 7 BITS OUTPUT
0143  491  32  ORL  F$;110h  ;PUT OUT STOP BIT
0143  E06  33  MOV  F$;004h  ;SET UP DELAY LOOP LENGTH
0147  140  34  CALL  CALL
0149  E  35  JF0  NOTTHEN  IF  IN  SETUP  SEGMENT  DONT  ENABLE  INTERRUPTS
014B  16  36  EN  END
014D  E  37  NOTTHEN  RET  RETURN  FROM  SUBRUTINE
014D  E  38  DELAY  DJNZ  F$;DELAY  VARI-ABLE  DEPENDING  ON RC
014F  8  39  RET  RETURN  FROM SUBRUTINE
0150  E  40  DELAY  MOV  F$;0150h  ;NESTED  DELAY  LOOP:  F  1X  1/4 SEC
0152  FFF  41  DLOOP:  MOV  F$;200ffh
0154  440  42  CALL  CALL
0156  F  43  DJNZ  F$;DLOOP
0158  H  44  RET  RETURN
015D  F  45  PULBIT:  MOV  a, 00h  ;CURRENT  RIFLE  NUMBER
015F  0  46  DEC  F$  ;CREATE  POINTER  FOR  LOAD  TABLE
015F  8  47  MOV  a, 00h  ;GET  E1G WITH  CORRECT  PULSE  BIT SET
015F  B  48  RET  RETURN
015F  B  49  CLR:  MOV  F$;10000h  ;LINE  FEED
015F  144  50  CALL  OUTPUT
0161  210  51  MOV  F$;0000h  ;CARRIAGE  RETURN
0163  348  52  CALL  CALL
0163  508  53  RET  RETURN

100  1  54  CHECK  CLR  --  TIMER  STARTING  COUNT  FOR  TIMEOUT
101  2  55  THE  PROCESSOR  WILL  BE  INTERRUPTED
101  6  56  IF  THE  80/20  DOESNT  RESPOND  WITHIN
101  6  57  A  SPECIFIED  TIME.
101  6  58  MOV  a, 1a  ;COUNTER  TIMER
101  6  59  RET  RETURN
101  6  60  STRT  1  ;STARTS  TIMER
101  6  61  CALL  CALL
101  6  62  DJNZ  F$;STOP  ;GROUP  TIMEOUT  WILL  LOOK  FOR  PENDING  TIMER  UNTIL
101  6  63  STOP  ;TIMEOUT  HAS  OCCURRED

101  6  64  MOV  a, 01h  ;INTERUPT  MUST  NOT  OCCUR
101  6  65  LOOP1:  MOV  a, 00ffh  ;SEND  OUT  VOTRAN  INITIALIZATION  WORDS
101  6  66  140  66  CALL  CALL
101  6  67  DJNZ  F$;LOOP2
101  6  68  JNTBF  POINT  1;RESPONSE  FROM  80/20
101  6  69  IN  00h  ;STARTING  80/20  DATA
LOC  DEC  SEQ  SOURCE STATEMENT

0177 DC  269  XRL A. R4  ; IF IDENTICAL RESULT IS ZERO
0178 DC  270  IZ NEXT1  THEN GO ON WITH TEST
0179 DC  271  MOV A.80 LOW FAI2
017C DC  272  CALL FAIL
0177 DC  273  JMP NEXT1
0180 DC  274  NOINTR MOV A.0 LOW FAI3
0182 DC  275  CALL FAIL  IF INDICATE FAILURE
0184 DC  276  NEXT1 MOV R1.0100  RESIVE 80/20 TIME TO SEND REMAINING
                                                         PROPORTION OF MESSAGE
0186 DC  278  LOOPY MOV P2.08FFH
0188 DC  279  CALL DELAY
018A DC  280  DUNZ P1.LOOPY
018B DC  281  RET
018E DC  282  RIFLOP MOV A. R6  RETRIEVE RIFLE NUMBER
018E DC  283  OR A.001H  AND START BIT ON SERIAL OUT LINE
018F DC  284  OUTL P1.A  SET UP MUX
0191 DC  285  MOV A. R6  GET CURRENT RIFLE
0192 DC  286  SWAP A  INPUT IN HIGH BYTE
0194 DC  287  RL A  UPPER 3 BITS
0195 DC  288  OR A. R5  CREATE CORRECT RETURN CODE
0196 DC  289  MOV R4. A  TEMP STORE
0199 DC  290  CALL PULB1  SET BYTE WITH CORRECT BIT SET
019A DC  291  BJ FOR RIFLE TRIGGER
0199 DC  293  ORL A.004H  KEEP TARGET PRESENT DOWN
019B DC  294  OUTL P2. A  RISING EDGE OF TRIGGER PULSE
019E DC  295  CALL P2.046H  FALLING EDGE OF PULSE
019E DC  296  CALL CHECK
019F DC  297  RET
01A0 DC  298
01A9 DC  299  DRIVE FOR RIFLE SIMULATION AND ITS MESSAGES LOCATED
01BA DC  300  IN FOURTH PAGE OF MEMORY
01BC DC  301
01BF DC  302
01C3 DC  303  THIS SEGMENT OF THE UPI-PROGRAM PROVIDES SIMULATED
01C4 DC  304  RIFLE DATA INPUT TO THE 80/20 COMPUTER. IT CHECKS
01C5 DC  305  FOR THE PROPER RETURN BYTE TO THE UPI-41. FOR THE
01C6 DC  306  SIMULATED SHOT AND INDICATES FAILURES BY A MESSAGE
01C7 DC  307  TO THE CONSOLE. INITIALLY IT SIGNS ON AND PROMPTS
01C8 DC  308  THE UPI-FOR THE HARDWARE MODIFICATIONS NEEDED.
01C9 DC  309  IF A FAILURE OCCURS THE TEST WILL CONTINUE AND OUTPUTS
01CA DC  310  A FAILURE MESSAGE FOR EACH FAILURE OCCURRENCE
01CB DC  311  WHEN THE TEST IS COMPLETE IT PROMPTS THE USER TO
01CC DC  312  ASK THE 80/20 FOR ITS OUTPUT
01CD DC  313
01CE DC  314
01F3 DC  315  THE PROGRAM IS NON-INTERCEPT DRIVEN AND INSTEAD USES
01F4 DC  316  THE INTERRUPT FLAG TO DETERMINE WHEN VALID DATA IS PRESENT
01F5 DC  317  ON THE 80/20 DATA LINES. UPON ENTERING THE ROUTINE
01F6 DC  318  FLAG G IS SET SO THAT INTERRUPTS WILL NOT BE REENABLED
01F7 DC  319  WHEN THE OUTPUT ROUTINE IS CALLED IF THE 80/20 DOES
01F8 DC  320  NOT RESPOND A "0" TO AN INPUT BY THE UPI-41 (INDICATED)
01F9 DC  321  BY THE INTERRUPT FLAG NEVER BEING SET) THE 41 TIMES
01FA DC  322  OUT AND JUDGE THIS A "0" FAILURE AND CONTINUES THE TEST.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>SRC</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>324</td>
<td></td>
<td>325</td>
<td>REGISTER BANK 1 IS USED FOR THE ROUTINE AND IS DEFINED</td>
</tr>
<tr>
<td>326</td>
<td></td>
<td>327</td>
<td>A FOLLOWS</td>
</tr>
<tr>
<td>328</td>
<td></td>
<td>329</td>
<td>REGISTER 0 UNUSED</td>
</tr>
<tr>
<td>330</td>
<td></td>
<td>331</td>
<td>REGISTER 1 OUTER LOOP OF DELAY COUNTER</td>
</tr>
<tr>
<td>332</td>
<td></td>
<td>333</td>
<td>REGISTER 2 INNER LOOP OF DELAY COUNTER</td>
</tr>
<tr>
<td>334</td>
<td></td>
<td>335</td>
<td>REGISTER 3 DELAY COUNTER</td>
</tr>
<tr>
<td>336</td>
<td></td>
<td>337</td>
<td>REGISTER 4 EXPECTED RETURN DATA FROM 80/20</td>
</tr>
<tr>
<td>338</td>
<td></td>
<td>339</td>
<td>REGISTER 5 TEMP STORAGE</td>
</tr>
<tr>
<td>340</td>
<td></td>
<td>341</td>
<td>REGISTER 6 5 RIFLE LOOP COUNTER AND CURRENT RIFLE</td>
</tr>
<tr>
<td>342</td>
<td></td>
<td>343</td>
<td>REGISTER 7 16 SHOT POSSIBILITIES LOOP COUNTER</td>
</tr>
<tr>
<td>344</td>
<td></td>
<td>345</td>
<td>AND CURRENT SHOT TYPE</td>
</tr>
<tr>
<td>346</td>
<td></td>
<td>347</td>
<td>THE BOTTOM 5 LINES OF PORT TWO FUNCTION AS THE RIFLE</td>
</tr>
<tr>
<td>348</td>
<td></td>
<td>349</td>
<td>TRIGGERS INSTEAD OF THE MASK INPUTS</td>
</tr>
<tr>
<td>350</td>
<td></td>
<td>351</td>
<td>LINE SIX OF PORT TWO IS THE TARGET PRESENT SIGNAL</td>
</tr>
<tr>
<td>352</td>
<td></td>
<td>353</td>
<td>PORT 1 LINES 4-7 SERVE AS THE SHOT TYPE INPUT LINES FOR</td>
</tr>
<tr>
<td>354</td>
<td></td>
<td>355</td>
<td>THE 80/20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>356</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>47</td>
<td></td>
<td>04, 51.20</td>
</tr>
<tr>
<td>3201</td>
<td>49</td>
<td></td>
<td>SH 00 DB 0 2, 4, 3, 5, 0, 5, 4</td>
</tr>
<tr>
<td>3202</td>
<td></td>
<td>3203</td>
<td></td>
</tr>
<tr>
<td>3204</td>
<td></td>
<td>3205</td>
<td></td>
</tr>
<tr>
<td>3206</td>
<td></td>
<td>3207</td>
<td></td>
</tr>
<tr>
<td>3208</td>
<td>50</td>
<td></td>
<td>DB 0 8, 9, 8, 2, 0, 8, 6, 1</td>
</tr>
<tr>
<td>3209</td>
<td></td>
<td>3210</td>
<td></td>
</tr>
<tr>
<td>3211</td>
<td></td>
<td>3212</td>
<td></td>
</tr>
<tr>
<td>3213</td>
<td></td>
<td>3214</td>
<td></td>
</tr>
<tr>
<td>3215</td>
<td></td>
<td>3216</td>
<td></td>
</tr>
</tbody>
</table>
LOC 164       SEQ 156       SOURCE STATEMENT
8216 25   360   EN  TONT   TIMER INTERRUPT IS USED AS A TIMEOUT
            366   FOR SEARCH ROUTINES
8217 E91A   367   MOV  R6.0   CLEAR THE CRT SCREEN & HOME CURSOR
8219 43D0   368   CALL  OUTPUT
821B 4550   369   CALL  DELAY1   ALLOW CRT TIME TO CLEAR
821D 45D0   370   CALL  CALLR   SPACE DOWN THREE LINES
821F 45D0   371   CALL  CALLR
8221 45D0   372   CALL  CALLR
8223 3800   373   MOV  R.0   LOW SIGNON ADDRESS OF SIGN ON MESSAGE
8225 10   374   MOV  R3.0   STORE STRING ADDRESS
8226 1E   375   MOV  R9.0   GET STRING LENGTH
8227 1E   376   MOV  R5.0   STORE STRING LENGTH
8228 427   377   CALL  STADROUT   SEND STRENGTH
822B 45D0   378   CALL  CALLR   ACKNOWLEDGE RETURN LINE FEED
822C 45D0   379   CALL  CALLR
822E 38D0   380   MOV  R.0   LOW PROMPT ADDRESS PROMPT MESSAGE
822F 16   381   MOV  R3.0   STORE STRING ADDRESS
822F 1E   382   MOV  R9.0   GET STRING LENGTH
8231 1E   383   MOV  R5.0   STORE STRING LENGTH
8233 427   384   CALL  STADROUT
8235 45D0   385   CALL  CALLR
8237 3450   386   CALL  DELAY1   WAIT FOR 08/20 TO TYPE OUT LET'S START.
8239 3450   387   CALL  DELAY1
823B 3450   388   CALL  CALLR
823D 05   389   TEST: SEL RBL
823E 1460   390   AND  P2.0[4-H]   REENABLE CHIP SELECT AND PUT TARGET FLG UP.
823F 9615   391   LRIP5: MOV  R6.0[5-H]   INITIALIZE RIFLE NUMBER AND LOOP
8240 6F18   392   LOOP4: MOV  R7.0[4-H]   COUNTER FOR 5 TIMES THROUGH
8242 6F18   393   LOOP5: MOV  R8.0[5-H]   COUNTER FOR 5 TIMES THROUGH
8244 54D0   394   LOOP6: CALL  SLOOPR   RIFLE SIMULATION SUBROUTINE
8246 4E44   395   LOOP7: CALL  R7.RIFLOR   CALL DATA POSSIBILITIES DONE?
8248 4E42   396   LOOP8: CALL  R6.RIFLOR   DONE?
824A BA40   397   ORL  P2.0[4-H]   TARGET PRESENT DOWN
824C 8865   398   MOV  R3.0[5-H]   WAIT 1 SEC REQUIRED BEFORE NEW TAR
824E 3450   399   LOOP9: CALL  DELAY1   CAN APPEAR
8250 8E4E   400   DJNZ  R5.0[4-H]   RETURN TO LOOP 5 
8252 9A00   401   TOLATE: ANL  P2.0[0-H]   TARGET PRESENT UP
8254 8A66   402   MOV  R2.0[6-H]   MAKE SURE 08/20 SEES FLAG
8256 3440   403   CALL  DELAY
8258 3440   404   MOV  R5.0[4-H]   TARGET PRESENT DOWN
825A 8A64   405   MOV  R4.0[4-H]   TARGET PRESENT DOWN
825C 8A64   406   MOV  R5.0[4-H]   GIVE 08/20 CHANCE TO RESPOND
825E 3440   407   CALL  DELAY
8260 B865   408   MOV  R6.0[5-H]   RIFLE COUNTER
8262 8A00   409   MOV  P5.0[0-H]   CODE FOR MISS-TOO LATE
8264 340D   410   MOV  P6.0[0-H]   CODE FOR MISS-TOO LATE
8266 340D   411   LOOP10: CALL  RIFLOR   SUBROUTINE WHICH FIRES SHOT FOR CURRENT TAR
8268 B865   412   MOV  R3.0[5-H]   RIFLE: CHECKS FOR RESET TO INTERNAL FF.
826A 3450   413   MOV  R4.0[4-H]   TARGET PRESENT DOWN
826C 4407   414   MOV  P6.0[0-H]   TARGET PRESENT DOWN
826E B865   415   MOV  R6.0[5-H]   TARGET PRESENT DOWN
8270 3450   416   LOOP11: CALL  DELAY1   CAN APPEAR
8272 BB66   417   DJNZ  R5.0[4-H]   RETURN TO LOOP 5 
8274 BB66   418   DJNZ  R4.0[4-H]   RETURN TO LOOP 5 
8276 BB65   419   TAR:   MOV  P6.0[5-H]   RIFLE COUNTER

D-8
LOC  OBJ  S'G  SOURCE STATEMENT

026E 3450  20 LOOPD  CALL  DELAY 1  :TATGET PRESENT UP

0271 0008  21  AMI  P2. 48H  671E 80/20 TIME TO RESPOND

0272 343F  22  MOV  R2. 4FFH  ARM TO IEEK DATA SET FOR HITS, NO START BIT

0273 3440  23  CALL  DELAY  :ATTACKING EDGE OF TRIGGERS

0274 3459  24  CALL  PULL  T

0275 4338  125  ORL  A. 81.11100000K: :TO 7 BITS MUST BE ZERO AFTER CPL

0276 47 37  126  CPL  A  :ALL BUT ONE RIFLE WILL SHOOT

0277 33A8  127  OUTL  P2. A  :ARMING EDGE OF TRIGGERS

0278 9061  128  AMI  P1. 39H  :ATTACKING EDGE OF TRIGGERS

0279 3430  129  AMI  P2. 48H  :ARMING EDGE

027A 3458  130  CALL  DELAY  :ATTACKING EDGE OF TRIGGERS

027B 3450  131  CALL  DELAY 1  :G0E 80/20 TIME TO SEND ALL MESSAGES

027C 22  132  IN  A. 34B  :CLEAR 1DF  FLAG WHICH THE 80/20 SENT

027D 117  133  JNZS  TO,  :USES TO.

027E 3440  434  CPL  P2. 440H  :TARGET PRESENT DOWN

027F 3406  435  MOV  R3. 46D  :DELAY OVER 1 SECOND SO THAT UPI WILL BE

0280 3458  436  100K  CALL  DELAY  :CAPTAIN TO HAVE RECEIVED INTER PULSE

0281 3489  437  DJNZ  R3. 100PK  :CAPTAIN TO HAVE RECEIVED INTER PULSE

0282 656D  438  JNIB  NOTT  :, IF NO INT PULSE INDICATE A FAILURE

0283 47  139  MOV  A. 4H  :GET CURRENT RIFLE

0284 97  140  RL  A  :DATA IN BITS 3-2-1

0285 47  141  SAMP  A  :AUXILIARY BITS

0286 3480  442  ORL  A. 4CH  :C IS ADDRESS OF "TARGET IGNORED"

0287 443  443  MOV  R4. H  :STORE

0288 444  444  IN  A. 34B  :INPUT DATA

0289 445  445  XRL  A. 4H  :IF RESULT IS ZERO THEN BYTES IDENTICAL

028A 446  446  JZ  CON  A  :CHECK STRIKE TO REJECT IF TARGET IMPLOSION

028B 447  447  MOV  A. 0 LOW FAL.2  :PREPARE TO CALL DATA FAILURE ROUT

028C 44F  448  JMP  CONT  B  :PREPARE TO CALL DATA FAILURE ROUT

028D 2324  449  MO1NT  MOV  A. 0 LOW FAL3  :TONAL TO CALL DATA FAILURE ROUT

028E 544D  450  CONTB  CALL  FAIL  :STORE RIFLE IDENTIFIER TO CAT

028F 340D  451  CONTB  CALL  FAIL  :STORE FAILURE TYPE

0290 344E  452  NIOM1R. 55W  R6. 45  :5 RIFLES

0291 3488  453  MO1Y  R5. 40BH  :NONE FOR NO TARGET

0292 3430  454  LOOPD  CALL  RIFLOP  :FIRST BIT FROM 8D IN

0293 434  455  DJNZ  R6. 100P  :FIRST BIT FROM 8D IN

0294 447A  456  JMP  FIN  S  :STORE RIFLE IDENTIFIER

0295 44F  457  ;

0296 458  458  ;

0297 459  459  ;

0298 460  460  ;

0299 461  461  ;

029A 462  462  ;

029B 463  463  ;

029C 464  464  ;

029D 05  465  FAIL.  SEL  RBB  :SEND RIFLE IDENTIFIER TO CAT

029E 9F  466  MOV  R7. H  :STORE FAILURE TYPE

029F 45  467  SEL  RBB  :STORE CURRENT RIFLE

02A0 468  468  MOV  A. R  :STORE CURRENT RIFLE

02A1 45  469  SEL  RBB  :STORE CURRENT RIFLE

02A2 47  471  RL  A  :ONE BY TWO TO ACCESS TWO LOCATIONS AT A TIME

02A3 43F0  472  ORL  A. 80F8H  :PRESS FAB

02A4 47  473  MOV  R5. H  :STORE

02A5 47  474  MOV  R5. H  :STORE

02A6 47  475  MOV  R5. H  :STORE

02A7 47  476  MOV  R5. H  :STORE
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>SEQ</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>028H</td>
<td>3438</td>
<td>475</td>
<td>CALL OUTPUT</td>
</tr>
<tr>
<td>028H</td>
<td>FD</td>
<td>476</td>
<td>MOV A, R5</td>
</tr>
<tr>
<td>028H</td>
<td>17</td>
<td>477</td>
<td>INC A</td>
</tr>
<tr>
<td>028H</td>
<td>R3</td>
<td>478</td>
<td>MOV R3, A</td>
</tr>
<tr>
<td>028H</td>
<td>R9</td>
<td>479</td>
<td>MOV P1.A</td>
</tr>
<tr>
<td>028H</td>
<td>3438</td>
<td>480</td>
<td>CALL OUTPUT</td>
</tr>
<tr>
<td>028H</td>
<td>B0</td>
<td>481</td>
<td>MOV R16, A</td>
</tr>
<tr>
<td>028H</td>
<td>3438</td>
<td>482</td>
<td>CALL OUTPUT</td>
</tr>
<tr>
<td>028H</td>
<td>FF</td>
<td>483</td>
<td>MOV A, R7</td>
</tr>
<tr>
<td>028H</td>
<td>AD</td>
<td>484</td>
<td>MOV R3, A</td>
</tr>
<tr>
<td>028H</td>
<td>AE</td>
<td>485</td>
<td>MOV P1.A</td>
</tr>
<tr>
<td>028H</td>
<td>3427</td>
<td>486</td>
<td>CALL STROUT</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>487</td>
<td>CALL CALL</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>488</td>
<td>CALL CALL</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>489</td>
<td>CALL CALL</td>
</tr>
<tr>
<td>028H</td>
<td>FF</td>
<td>490</td>
<td>MOV A, R7</td>
</tr>
<tr>
<td>028H</td>
<td>01</td>
<td>491</td>
<td>JEC A</td>
</tr>
<tr>
<td>028H</td>
<td>F37</td>
<td>492</td>
<td>JPL A</td>
</tr>
<tr>
<td>028H</td>
<td>530F</td>
<td>493</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>47</td>
<td>494</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>50</td>
<td>495</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>4E</td>
<td>496</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>E7</td>
<td>497</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>4D</td>
<td>498</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>4B</td>
<td>499</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>39</td>
<td>500</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>FF</td>
<td>501</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>B7</td>
<td>502</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>A3</td>
<td>503</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>H0</td>
<td>504</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>FE</td>
<td>505</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>F7</td>
<td>506</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>E7</td>
<td>507</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>AC</td>
<td>508</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>509</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>510</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>345D</td>
<td>511</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>35</td>
<td>512</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>27</td>
<td>513</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>514</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>515</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>516</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>517</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>518</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>519</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>520</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>521</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>522</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>523</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>524</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>525</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>526</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>527</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>528</td>
<td>JML A, B0BF1</td>
</tr>
<tr>
<td>028H</td>
<td>34</td>
<td>529</td>
<td>JML A, B0BF1</td>
</tr>
</tbody>
</table>

The program code appears to be for a microprocessor, possibly the 6502, and involves operations like call, mov, inc, and store, along with conditional jumps and loops. The code seems to be managing data flow and storage, possibly for a particular application, but the specific context is not clear from the code snippet alone.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>SEQ</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0360</td>
<td>04</td>
<td>556</td>
<td>DB 4H 'HIGH</td>
</tr>
<tr>
<td>0361</td>
<td>404</td>
<td>4748</td>
<td></td>
</tr>
<tr>
<td>0363</td>
<td>282</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>0369</td>
<td>286</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>036D</td>
<td>284</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0370</td>
<td>09</td>
<td>557</td>
<td>DB 9H 'HIGH LEFT</td>
</tr>
<tr>
<td>0371</td>
<td>404</td>
<td>4748</td>
<td></td>
</tr>
<tr>
<td>0375</td>
<td>282</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>0379</td>
<td>5420</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>037D</td>
<td>286</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0380</td>
<td>04</td>
<td>558</td>
<td>DB 4H 'LEFT</td>
</tr>
<tr>
<td>0381</td>
<td>404</td>
<td>4654</td>
<td></td>
</tr>
<tr>
<td>0385</td>
<td>282</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>0389</td>
<td>286</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>0390</td>
<td>820</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0391</td>
<td>282</td>
<td>4654</td>
<td></td>
</tr>
<tr>
<td>0395</td>
<td>286</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>0399</td>
<td>286</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>039D</td>
<td>07</td>
<td>560</td>
<td>DB 120 'INTL5-TOO LATE '</td>
</tr>
<tr>
<td>039F</td>
<td>404</td>
<td>0253</td>
<td></td>
</tr>
<tr>
<td>03AF</td>
<td>284</td>
<td>4F4</td>
<td></td>
</tr>
<tr>
<td>03B9</td>
<td>28</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>03C0</td>
<td>07</td>
<td>561</td>
<td>DB 9H 'NO TARGET'</td>
</tr>
<tr>
<td>03C1</td>
<td>54</td>
<td>15247</td>
<td></td>
</tr>
<tr>
<td>03C5</td>
<td>45</td>
<td>2049</td>
<td></td>
</tr>
<tr>
<td>03C9</td>
<td>47</td>
<td>4F52</td>
<td></td>
</tr>
<tr>
<td>03D0</td>
<td>45</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>03D4</td>
<td>06</td>
<td>563</td>
<td>SIGNON: DB 15D 'FIRE SIMULATOR'</td>
</tr>
<tr>
<td>03D9</td>
<td>46</td>
<td>4C</td>
<td></td>
</tr>
<tr>
<td>03D5</td>
<td>45</td>
<td>5349</td>
<td></td>
</tr>
<tr>
<td>03DB</td>
<td>40</td>
<td>4C4</td>
<td></td>
</tr>
<tr>
<td>03DC</td>
<td>54</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>03E0</td>
<td>0F</td>
<td>564</td>
<td>PROMPT: DB 15D 'TRAP IN PLACE?'</td>
</tr>
<tr>
<td>03E1</td>
<td>53</td>
<td>0241</td>
<td></td>
</tr>
<tr>
<td>03E5</td>
<td>58</td>
<td>394E</td>
<td></td>
</tr>
<tr>
<td>03E9</td>
<td>28</td>
<td>4C4</td>
<td></td>
</tr>
<tr>
<td>03ED</td>
<td>43</td>
<td>3F</td>
<td></td>
</tr>
<tr>
<td>03F0</td>
<td>0F</td>
<td>565</td>
<td>DONE: DB 15D 'TEST COMPLETE'</td>
</tr>
<tr>
<td>03FA</td>
<td>28</td>
<td>5445</td>
<td></td>
</tr>
<tr>
<td>03F5</td>
<td>51</td>
<td>42043</td>
<td></td>
</tr>
<tr>
<td>03F9</td>
<td>4F</td>
<td>594C</td>
<td></td>
</tr>
<tr>
<td>03FD</td>
<td>45</td>
<td>145</td>
<td></td>
</tr>
</tbody>
</table>

END
APPENDIX E

SELF CHECK PROGRAM
## Symbol Table of Module TST

Read from file FILE1ST TEMP
Written to file FILETST

<table>
<thead>
<tr>
<th>Value Type</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOD TEST MODULE</td>
<td>$01FH$ SYM MEMORY</td>
<td>$001H$ SYM TEST</td>
</tr>
<tr>
<td></td>
<td>$001H$ LIN 1</td>
<td>$002H$ LIN 1</td>
</tr>
<tr>
<td></td>
<td>$001H$ LIN 1'</td>
<td>$007H$ LIN 1'</td>
</tr>
<tr>
<td></td>
<td>$00AH$ LIN 18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$00CH$ LIN 19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$010H$ LIN 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$013H$ LIN 21</td>
<td></td>
</tr>
<tr>
<td>MOD TESTPROC MODULE</td>
<td>$01FH$ SYM MEMORY</td>
<td>$000H$ SYM TSTCHECK</td>
</tr>
<tr>
<td></td>
<td>$300H$ SYM N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ SYM DONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ SYM LEDON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$301H$ SYM K</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$020H$ SYM FAIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$302H$ SYM J</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$303H$ SYM IDATA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ SYM IOTEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$084H$ SYM LOWLIMIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$086H$ SYM HIGHLIMIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$304H$ SYM INITIALETIME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$305H$ SYM FINALTIME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$308H$ SYM ELAPSEDETIME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$309H$ SYM I</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$0D0H$ SYM TIMEPTEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$133H$ SYM USARITEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ LIN 41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ LIN 42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ LIN 43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ LIN 44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$018H$ LIN 45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$020H$ LIN 47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$024H$ LIN 49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$033H$ LIN 50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$041H$ LIN 51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$044H$ LIN 52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$04FH$ LIN 53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$051H$ LIN 54</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$05FH$ LIN 55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$061H$ LIN 56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$066H$ LIN 57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$067H$ LIN 58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$069H$ LIN 59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$069H$ LIN 60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 68</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 71</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$080H$ LIN 73</td>
<td></td>
</tr>
</tbody>
</table>
MEMORY MAP OF MODULE TST
MBRD FROM FILE :F1 TST TMP
WRITTEN TO FILE :F1 TST
MODULE START ADDRESS 0026H

START STOP LENGTH REL NAME
C000H C008H 08H A ABSOLUTE
C100H C10EH 132H A ABSOLUTE
THE TEST PROGRAM WAS WRITTEN BY TOM RIORDAN. ITS FUNCTION
IS TO ACT AS THE DRIVER FOR THE TEST PROCEDURES AS CALLED. */

1 TEST MODULE NO.
2 1 RAIN'T: PROCEDURE EXTERNAL;
3 2 END RAINST;
4 1 RUN'T: PROCEDURE EXTERNAL;
5 2 END RUNST;
6 1 IOST: PROCEDURE EXTERNAL;
7 2 END IOST;
8 1 TIMES: PROCEDURE EXTERNAL;
9 2 END TIMES;
10 1 USOST: PROCEDURE EXTERNAL;
11 2 END USOST;
12 1 DONE: PROCEDURE EXTERNAL;
13 2 END DONE;
14 1 TEST: PROCEDURE PUBLIC;
15    CALL RAINST;
16    CALL RUNST;
17    CALL IOST;
18    CALL TIMES;
19    CALL USOST;
20    CALL DONE;
21 2 END TEST;
22 1 END TEST MODULE;

MODULE INFORMATION

DATA AREA SIZE = 00010H  1:0
VARIABLE AREA SIZE = 00044H  1:0
MAXIMUM STACK SIZE = 00000H  0
+ LINES READ
A PROGRAM ERROR(S)
1  TESTAPP: \nMODULE Public:

2  DECLARE TESTCHECK BYTE Public At (80H-89H) Equ(1):
   Show List

38 DECLARE N BYTE:
   * DECLARE WORD LITERALLY 'ADDRESS':

44 DECLARE DONTCARE LITERALLY '50H', FOREVER LITERALLY 'WHILE 1',
   DIAGNOSTIC LED LITERALLY '3SH':

49 DECLARE PROCEDURE Public:

52 LED1:
   DO FOREVER:
       OUTPUT(DIAGNOSTIC LED)=DONTCARE;
   END LED1:

57 END DONE:

61 DECLARE 4 BYTE:

63 DECLARE 3 BYTE PUBLIC:

66 DECLARE 2 BYTE:

69 DECLARE K=1 TO 5:

72 DO N=1 TO 5:

75 OUTPUT(DIAGNOSTIC LED)=DONTCARE
   CALL SBCTIM(10);
   DO:
   CALL SBCTIM(25);

79 END:

82 END;

85 DO N=1 TO 80:

89 CALL SBCTIM(50); /* 4IT 2 ECODY THEN 40 ON WITH TEST */

92 END;

95 END FAIL:

101 DECLARE IOMFAIL LITERALLY '3'; /* 3 FLASHES FOR AN I/O FAILURE */

105 DECLARE PORT1 LITERALLY 'B0H', PORT2 LITERALLY 'B0H',
   PORT3 LITERALLY 'B0H', PORT4 LITERALLY 'B0H';

109 DECLARE IODATA BYTE;

113 DECLARE IODATA PUBLIC:

117 CALL PORTSET: /* SET UP PORTS 1A2 AS INPUTS 386 AS OUTPUTS */

120 OUTPUT(PORT1)=0BH /* PORT 3 WILL INVERT OUTPUT THEN PORT 1 WILL REINVERT IT */

123 IODATA=INPUT(PORT1);

126 IF IODATA=0BH THEN
   CALL FAIL(IOMFAIL);

129 OUTPUT(PORT3)=0FH;

132 IODATA=INPUT(PORT1);

135 IF IODATA=0FH THEN

E-4
CALL 'FAIL(IONFAIL)';

OUTPUT(PORT6)=0BH; /* PORT 7 INVERTS OUTPUT BUT PORT 8 WILL NOT INVERT */

IDATA1=INPUT(PORT2);
IF IDATA1=OFFH THEN
CALL 'FAIL(IONFAIL)';

OUTPUT(PORT6)=OFFH;
IDATA1=INPUT(PORT2);
IF IDATA1=OFFH THEN
CALL 'FAIL(IONFAIL)';

END;
END 'IONTEST';

DECLARE LOWLIMIT WORD DATA(100); DECLARE TIMERFAIL LOW LITERALLY '4'; TIMERFAILHIGH LITERALLY '5'; DECLARE (INITIALTIME,FINALTIME,ELAPSEDTIME) WORD;
DECLARE I BYTE;
TIMERTEST: PROCEDURE PUBLIC;
CALL L 'TIMERTEST'; /* START TIMERS 0 AND 1 */
CALL 'SEC1N(256)'; /* CIVE TIMER TIME TO BEGIN FUNCTIONING */
INITIALTIME = 'LOCKED';
0=1 TO 40; /* WAIT FOR 0.6 SECONDS */
CALL 'SEC1N 256';
END;
FINALTImE='CLOSED';
ELAPSEDTIME = INITIALTIME - FINALTIME; /* COUNTERS ARE DOWN COUNTERS */
IF ELAPSEDTIME < LOWLIMIT THEN
CALL 'FAIL(TIMERFAILLOW)';
IF ELAPSEDTIME > HIGHLIMIT THEN
CALL 'FAIL(TIMERFAILHIGH)';
END 'TIMERTEST';

DECLARE USARTFAIL LITERALLY '6';
DECLARE USARTTEST LITERALLY '5'; USHRTDATA LITERALLY '00H';
USARTTEST: PROCEDURE PUBLIC;
CALL VOTRANSTIME;
CALL 'SEC1N(100)'; /* MAKE CERTAIN TIMER HAS STARTED */
/* SET 16D RATE AND BIT PATTERN */
CALL 'SEC1N(200)'; /* MAKE CERTAIN USHRT HAS COMPLETED INTERNAL SETUP */
/* SEND OUT TEST PATTERN */
/* CALL 'SEC1N(20)'; /* WAIT APPROX 1.04 MSEC=USHRT SHOULD BE DONE */
/* H.B. THIS MUST BE LONG ENOUGH EVEN WITHOUT WAIT STATES */
CALL 'FAIL(USARTFAIL)';
END 'USARTTEST';

END TEST; RETURN;
VARIABLE AREA SIZE = 400BH 11D
MAXIMUM STACK SIZE = 1004H 40
138 LINES READ
0 PROGRAM ERROR(S)

END OF PL M-60 COMPILATION
1800 11: RANTST SRC DEBUG MACROFILE "TITLE(11 OCT 78)"

1351: 8088/8085 MACRO ASSEMBLER, V2.4 RANTST PAGE 1
11 OCT '8

LOC 0BJ  SEQ SOURCE STATEMENT

1 NAME RANTST
2 STRL1 ON
3 EXTRN DONE
4 PUBLIC RANTST
5
6 CSEGS
8000 V1 7 RANTST POP D GET RETURN ADDRESS THAT WAS PUSHED:
9 BY THE CALL AND SAVE IT IN THE D-10 REG PAIR. IT WILL BE VALID IF RAM
11 IS OKAY, AND UNUSED OTHERWISE
12 8000:0100FB 11 LXI B.0FB-0H INCREMENTING THIS VALUE AND CHECKING
13 FOR OVERFLOW WILL INDICATE WHEN TEST
14 IS FINISHED.
15 8000:1000B 14 LXI H.3000H START OF RAM
16
18 15 LOOP: ORA A
19 MOV M.A STORE 00H AT LOCATION
20 MOV M.A READ 00 FROM SAME LOCATION?
21 8000:21E00 C 19 JNZ RAINFAIL IF 00 NOT READ BACK JUMP TO FAILURE ROUTINE
22
23 20 ORA A IF PASSES THEN ACCUM=FFH
24
25 7 MOV M.A STORE FFH AT LOCATION
26 MOV M.A READ FFH FROM SAME LOCATION?
27
28 1 C 23 INR A IF FFH READ BACK ACCUM=00
29
30 2 C2LE00 C 24 JNZ RAINFAIL IF FFH NOT READ BACK JUMP TO FAIL ROUTE
31
32 6 5 INX H ADDRESS NEXT MEMORY LOCATION
33
34 6 81 MOV A.B CHECK FOR TEST COMPLETE
35
36 87 ORA A
37
38 C20700 C 29 JNZ LOOP
39
40 5 11 MOV A.B STORE RETURN ADDRESS BACK ON STACK.
41
42 59 PUSH D
43 5 C9 RET
44
45 306 B RAINFAIL: OUT 00GH FLUSH LED 1 TIME TO INDICATE RAM FAILURE
46 81E8FD 23 LXI B.65-0H DELAY APPROXIMATELY 1 SEC THEN JUMP TO
47 88 8B 24 LOOPH: DCX B Done Routine. This is done because the
48 88 88 35 NOP REST OF THE TESTS CANNOT BE RUN RELIABLY
49 88 78 36 MOV A.B UNLESS THE RAM IS WORKING PROPERLY
50 88 87 37 ORA A
51 8C2380 C 38 JNZ LOOPH
52 8C2000 E 39 JMP DONE
53 3 END

INTER-SEGMENT CALLS

EXTRA SYMBOLS

END: C0000

INPUT SYMBOLS

RAM: 40000 LOOP C0007 LO M A C 23 RAINFAIL C001E RANTST C0000

ERROR COMPLETE. NO ERRORS

E-7
AS1118

R111

W0111

K1101

T1171

J291

18 JUN 79

LO OBJ SED SOURCE STATEMENT

NAME R0111 TEST
STRING R111
EXTERNAL R299 FAIL
PUBLIC R311 R0111 TEST

CSEG

000 210000 8 R0111 TEST
000 5110000 9 R0111 TEST
000 529992 10 LOOPA R0111
000 52947 11 MOV R0111
000 52947 12 MOV R0111
000 52947 13 XCHG
000 52947 14 MOV R0111
000 52947 15 CMP R0111
000 52947 16 JZ CONTZ
000 52947 17 PUSH R0111
000 52947 18 PUSH R0111
000 52947 19 PUSH R0111
000 52947 20 PUSH R0111
000 52947 21 CALL FAIL
000 52947 22 POP R0111
000 52947 23 POP R0111
000 52947 24 POP R0111
000 52947 25 POP R0111
000 52947 26 CONT: INX R0111 NEXT ROM LOCATIONS
000 52947 27 INX R0111
000 52947 28 MOV R0111
000 52947 29 XCHG
000 52947 30 CPI R0111 IS TEST COMPLETE?
000 52947 31 JNZ LOOPA
000 52947 32 RET
000 52947 33 END

PUBLIC SYMBOLS
ROMST C 0800

EXTERNAL SYMBOLS
FAIL E 0800

USER SYMBOLS
M: 10001B FAIL E 0800 LOOPA C 0800: ROMST C 0800

A failure COMPLETE: MSE ERRORS

E-8
<table>
<thead>
<tr>
<th>LOC</th>
<th>OFF</th>
<th>SEC</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>ME SECTIM</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>FKLUN 04</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>PUBLIC SECTIM</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>C EG</td>
</tr>
<tr>
<td>6</td>
<td>0600</td>
<td>S &amp;</td>
<td>T 10</td>
</tr>
<tr>
<td>7</td>
<td>0278</td>
<td>LOP</td>
<td>R R</td>
</tr>
<tr>
<td>8</td>
<td>030D</td>
<td>LOOP</td>
<td>D R C</td>
</tr>
<tr>
<td>9</td>
<td>C780</td>
<td>LOOP</td>
<td>D R C</td>
</tr>
<tr>
<td>10</td>
<td>0B20</td>
<td>LOOP</td>
<td>D R C</td>
</tr>
<tr>
<td>11</td>
<td>0C9</td>
<td>LOPB</td>
<td>D R C</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>RT</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>E D</td>
</tr>
</tbody>
</table>

**PUBLIC SYMBOLS**

SECTIM C 0000

**EXTERNAL SYMBOLS**

**SEFP SYMBOLS**

LOPHA C 0003, LOOPB C 0042 SECTIM C 0000

**ASSEMBLY COMPLETE, NO ERRORS**