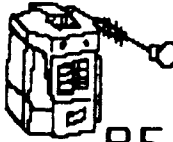


# HERO-GRAM

VOL. 2

1986

NO. 1



## NEWSLETTER OF THE HERO RESOURCE EXCHANGE

Walter Glod, Jr.  
10802 Condrey Ridge Court  
Richmond, Virginia 23236

### <<< LETTERS RECEIVED >>>

Carol Anderson (332-A Bayview St., San Rafael, CA 94901) developed a writing program for HERO in which he writes vertically against a slightly cone-shaped surface. The letter of the alphabet is selected by turning a dial made from an audio taper potentiometer, which mimics the resistance of HERO's light dependent resistor.

Ted J. Poulos (18 Cushing Road, Brookline, MA 02146) asks the following of the members, "My HERO BASIC gives me an 'Error Number 99' message everytime it executes an INPUT or CPUNCH statement. Programs can be saved to cassette using the monitor command, so the hardware appears okay. Heath sent me another BASIC ROM, but this did not solve the problem. Anyone have any ideas?".

Ronald Campbell (P.O.B. 284, Arlington, SD 57212) recently programmed HERO to lead a cheer at a community school rally during HOMECOMING festivities. After the cheer, the robot gave a pep talk. He also combined the random movement program (last newsletter) with his program so that as the cheerleaders lead other cheers, the robot seemed to respond to them. Mr. Campbell received many positive comments on the performance. By the next year he hopes to have HERO singing the school song.

### <<< NOTICES >>>

All programs submitted to this newsletter should be considered "Public Domain". If you wish to submit a program that does not fall into that category, please include a notice of it's restrictions with the program.

By request of many members, we will include the addresses of members who are referred to in the newsletter. If you do not wish your address to appear, please notify us.

### <<< HEROES FOR SALE >>>

From time to time HRE is contacted by individuals who have a HERO for sale. Presently we know of a few assembled ones at a good price. If your school needs an extra one, or you have a friend that couldn't afford a new one, contact us.

### <<< PROGRAMS FOR HERO >>>

HERO CALLER makes the robot a sophisticated alarm clock as he goes from room to room seeking out people at specified times. Routes, times, and announcements are easily programmable from HERO's keyboard.

HERO SANTA has HERO greet guests, announces how long until Christmas, sings, and performs a variety of tasks appropriate to the yuletide season.

Both are available from Robotronix, Inc., Box 1125, Los Alamos, NM 87544.

## ADDING 20 ADDITIONAL I/O LINES TO HERO

By Ted J. Poulos  
18 Cushing Rd.  
Brookline, MA 02146

If you own a Memory Expansion Board, additional I/O lines (16 data and 4 handshake) can be added to Hero using a Peripheral Interface Adapter (PIA) 6820 or 6821. All the required control and data lines are available at the memory IC sockets on the board with the exception of two, Reset and the E clock (see diagram). However, these two signals can be found on the board at IC U108.

To add the PIA, I made a circuit board (4x4") which plugged into socket U102 on the memory board. The circuit was layed out on copper clad using a resist pen. The board contains only four components: a 24 pin DIP socket used as a plug to mate with U102, a 6820 PIA, an I/O connector, and another small two pin connector for the reset and E leads. The tricky part is to mount a 24 pin DIP IC socket of the machined pin type (TI C7224, Digi-Key) to the foil side of the PC board and spaced from it about 1/2 ". I did this by soldering each of 24 bare copper wires 3/4" long to each pin receptacle of the socket. The free end of the wires were then soldered to corresponding pads on the PC board (foil side). The DIP socket thus becomes an extention plug rigidly mounted to the PIA board.

If you layout everything carefully, the PIA board will plug neatly into U102 and mount squarely over the U102 end of the memory board. I added a couple of spacers to make the "sandwich" more mechanically rigid. Keep in mind as you layout the board that U102 has 28 pins and that the 24 pin DIP plug on the PIA board uses only pins 3 thru 26 of U102.

The blue jumper plugs on the memory board near U102 must be connected as follows:

```
J101 B Supplies 5V to PIA via pin 26 of U102
J102 B R/W line via pin 23 of U102
J103 A CE2 select line via pin 22 of U102
```

When the select lines of the PIA are connected as shown in my diagram, the PIA will be addressed in the memory space from hex 1000 to 1FFF. Specifically, the four registers in the PIA can be addressed as follows:

```
$1010 Data Reg. A
$1011 Control Reg. A
$1012 Data Reg. B
$1013 Control Reg. B
```

To checkout the I/O board a series of from 4 to 8 LEDS, properly buffered can be connected in turn to each port. A BASIC program

like the following or a machine language program can be written to initialize the PIA.

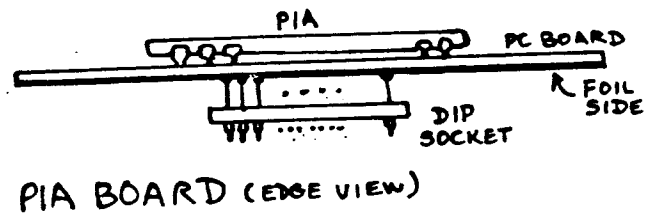
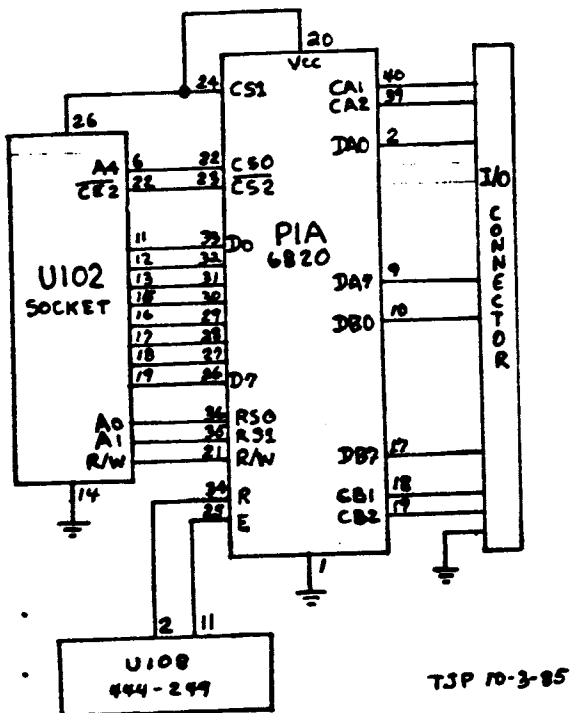
```

10 POKE $1011,0   Clear Cntrl Reg A
15 POKE $1013,0   Clear Cntrl Reg B
20 POKE $1010,$FF Config. Port A as output
25 POKE $1012,$FF Config. Port B as output
30 POKE $1011,$04 Select Data Reg. A
35 POKE $1013,$04 Select Data Reg. B
40 END
    
```

Now by POKES to addresses hex 1010 and 1012, data can be written to the data registers to light any LED at will. If you are not familiar with how to software configure a PIA, I can supply sources of information on this subject.

Finally, the same interfacing scheme as just described could be used to add an 6850 ACIA to Hero to provide a serial port.

### HERO PIA INTERFACE



HERO HEX CONVERSION  
Martin Fleming  
1405 4th Avenue. S.  
Moorhead, MN 56560

HEX TO DECIMAL ALGORITHM

8-14-85 PAGE 1

This program takes the hexadecimal number stored in ACCA and converts it to the correct decimal equivalent. NOTE: The largest number which this program can handle is 63(HEX).

0040 initial HEX value storage           0043 multiplier X  
0041 large loop counter X(1)           0044 decimal number Z  
0042 busy loop counter Y

Entry: ACCA                   Exit:ACCA

Program explanation.

This program converts a HEX number into a decimal number up to 99. First, it isolates the 10's digit and multiplies that by 16. Then the program converts the one's digit into decimal and adds that to the previous sum.

All of this is done through the inclusion of an increment loop, which follows the decimal rules (after 9 comes 10 .... not A).

PROGRAM LIST

7F 0040	CLR ext. :	clear work space
7F 0041	CLR ext. :	clear work space
7F 0042	CLR ext. :	clear work space
7F 0043	CLR ext. :	clear work space
7F 0040	CLR ext. :	clear work space
B7 0040	STAA ext. :	secure initial value
44 44	LSRA :	shift high nibble down to low
44 44	LSRA ext. :	shift high nibble down to low
4D	TSTA :	see if 10's digit is a zero
2719	BEQ :	no need to multiply
B7 0043	STAA ext. :	store value of X
4F	CLRA :	
C610	LDAB imm. :	load ACCB with 16 (decimal)
F7 0042	STAB ext. :	store in Y
7C 0041	INC ext. :	increment X(L)
8D1B	BSR :	goto increment loop subroutine
F6 0041	LDAB ext. :	get X(L)
F1 0043	CMPB ext. :	compare X(L) to X
26F3	BNE :	do another cycle
B7 0044	STAA ext. :	put corrected 10's into Z

DO 1'S DIGIT AND ADD TO THE 10'S DIGIT

B6 0040	LDAA ext. :	get initial number
840F	ANDA :	isolate the lower nibble
B7 0042	STAA ext. :	store ACCA in Y
B6 0044	LDAA ext. :	get Z out into ACCA
8D03	BSR :	goto increment subroutine
010101	:	left open for a 39 or a JMP (done)

```

5F      CLRB      :      Y(L)
5C      INCB      :      Y(L)
B7 0044 STAA ext. :      save value of Z
840F    ANDA      :      isolate lower nibble
8109    CMPA      :      compare ACCA to 9
2609    BNE       :      there is no nine
B6 0044 LDAA ext. :      get back value of Z
8B10    ADDA      :      increment 10's digit
84F0    ANDA      :      change 1,s digit to zero
2004    BRA       :
B6 0044 LDAA ext. :      get back the value of Z
4C      INCA      :      Z
F1 0042 CMPB ext. :      does ACCB equal Y
26E5    BNE       :      do another increment
39      RTS       :      return to program

```

This algorithm is very useful as a interface to the "human being" world. For example: if HERO is verbally reporting light levels, the data would be in base 10 not HEX. Also if a program requires data input on the keyboard you would not have to enter it in HEX, which is important for user-friendly programs.

---

### HERO DRIVE PATTERNS

Walter Glod

Here are a few patterns for HERO to maneuver. When run in Robot Language, HERO will drive in the shape of the path listed. The purpose is to have the robot remain within a given area (about 5.5 feet) for shows or demos.

PACER	PENTAGON	ATOMIC
0200 D3 10 70	020F D3 10 29	021E D3 10 29
0203 C3 F4 93	0212 C3 F4 93	0221 C3 F4 93
0206 D3 10 35	0215 D3 10 10	0224 D3 10 3F
0209 C3 F0 49	0218 C3 F0 49	0227 C3 F0 49
020C 7E 02 00	021B 7E 02 0F	022A 7E 02 1E

Note: The pattern will differ with the surface the robot drives upon.