

# HERO-GRAM

VOL. 1

NO. 4



## NEWSLETTER OF THE HERO RESOURCE EXCHANGE

Walter Glod, Jr.  
10502 Condrey Ridge Court  
Richmond, Virginia 23236

### <<< LETTERS RECEIVED >>>

Frank W. Crosby of Seneca Falls, N.Y. mounted a micro cassette recorder to the head of the robot with velcro for program transfers. Fiber optics, and blinking lights were added (for cosmetic purposes). He uses HERO as the core of a computer assisted instruction project and with "gifted and talented students" at Romulus Central School, Romulus N.Y. 14541 Mr. Crosby also operates a Computer Resource Center, using HERO at the elementary school level. He uses a Radio Shack Color Computer interfaced to his HERO 1 robot.

Patrick Stakem (Interface Technology, Inc., Laurel, MD.) has been involved in developing some hardware and software for a Clean-room Automatic Guided Vehicle. He has been using a HERO with it's hand outfitted with Sonar, Infrared, and Touch sensors and has had encouraging results instructing the robot to locate and pick up a marked object on a table.

### <<< HERO HARDWARE >>>

An update letter from DynaComp on the HERO 1 Motherboard Kit answered some of the questions that I forwarded to them. They said that the motherboard will allow use of the Heath Memory expansion, Serial board, and HERO BASIC with it. The only non-compatibility will be that you won't be able to use the optional DynaComp memory board in addition to the Heath one (but you won't need it if you have the Heath Memory Expansion).

### <<< HERO PUBLICATIONS >>>

A magazine that will be devoted entirely to robot experimentation is "Robot Experimenter Magazine", Suite 31, 174 Concord Street, Peterborough, N.H. 03458. I will have a column on HERO appearing in it.

HERO appears in "ROBOTS, Machines in Mans Image", by Isaac Asimov and Karen Frenkel (Harmony Books). The Hero Resource Exchange however, was not mentioned in the Robot Hobbyist Organizations listed in the back of this excellent book!

HERO also has a page in "ROBOTS" with Peter Marsh consulting (Crescent Books). They show a very simplified "exploded" view drawing of the robot with descriptions. They did however say "It has an arm with which to pick things up and put them on it's head..". If you can figure out how to have HERO use it's own arm to place an object on it's head, let me know!

### <<< HERO - TRS 80 Model 100 Communication >>>

The Radio Shack Model 100 Portable computer makes an excellent terminal to use with HERO BASIC. It's battery powered and contains a built in RS-232 Serial port and Communications software. Only 3 wires are necessary between HERO and the Model 100 : Pins 2, 3, and 7. Wire these pins the same on both sides (ie., 2 to 2, 3 to 3, 7 to 7). It's that simple!

ROBOT NAVIGATION  
by Walter Glod

One of the biggest obstacles to overcome in navigating HERO is that he will pick up reflections and other interference when attempting to "see" on the move. I found an easy and reliable method around this is to have the robot make sonar reading BEFORE he begins to move, and use the information for avoidance after moving. In the following program (which is in HERO BASIC) HERO takes a Sonar reading, displays "GO nn" (where nn is the inches), then moves that amount and takes another reading. After the robot determines that there is something in it's path, it will stop, move back, and look right and left for the most open area to travel. If HERO gets "hung-up" on an object, it will eventually get out of this stuck situation after looking at unchanging light conditions on the light sensor.

```

10 S=SONAR
20 IF S<10 THEN GOTO 100
30 E=EYE
40 DPRINT "$"
50 DPRINT "GO ";DPRINT S
60 FWD S
70 U=EYE
80 IF E=U THEN 270
90 GOTO 10
100 DPRINT "$STOP"
110 BWD 20
120 HEAD=70
130 DPRINT "$SEE "

140 T=SONAR:DPRINT T
150 HEAD= -70
160 W=SONAR
170 DPRINT "$SEE "
180 DPRINT W
190 HEAD=0
200 IF W>T THEN GOTO 240
210 IF T<10 THEN GOTO 100
220 DPRINT "$"
230 RIGHT 90; GOTO 10
240 IF W<10 THEN GOTO 100
250 DPRINT "$"; LEFT 90
260 GOTO 10
270 DPRINT "$STUCK";GOTO 100

```

-----  
"A STRANGE ONE"

Try this one for something different in Artificial Intelligence. It translates SOUND in the room to motor positions for the Arm and Head. Enter it from addresses 0400 and 043C. Make enough sound in the room to keep him moving. Remember,...the robot will act unpredictably!  
Note: Be sure to run this in "Robot Language" ie,.. A-D-0400.

0400 01	0417 BD 04 30	0430 8F 00 20
0401 BD 04 30	041A 86 54	0433 42
0404 86 50	041C B7 04 42	0434 BD F6 5B
0406 B7 04 42	041F F3 04 42	0437 B6 C2 40
0409 F3 04 42	0422 BD 04 30	043A BD F7 AD
040C BD 04 30	0425 86 D4	043D B7 04 43
040F 86 D0	0427 B7 04 42	0440 52 39
0411 B7 04 42	042A F3 04 42	0442 DATASPACE
0414 F3 04 42	042D 7E 04 00	0443 DATASPACE

Memory Checker by Tom Phillips of

Phillips Software  
1633 Commonwealth Avenue  
West Newton, Mass. 02165  
(617) 332-1373

This program will allow you to examine memory contents using Hero's voice. I wrote it to help me debug programs. It is especially useful for checking long listings (you don't have to keep looking back and forth between Hero's LED display and your program listing). All you need to do is look at your listing and listen to Hero read off the bytes in memory. Once loaded the program can relocate itself to high memory so that it can examine whatever program you have in low memory.

Here's how to use it:

1. Load the program by hand.
  2. Save the program on tape (from 0030 to 0180).
  3. Re-check each byte against my listing and make corrections if necessary (re-save program if corrections were made).
  4. Press user 1 key (or user 2, user 3 - they are all the same).
  5. Hero will say "enter address".
  6. Enter an address and Hero will display the address and byte at that address.
  7. Hero will tell you what byte is at that address.
  8. You may advance to the next memory location with F (Forward).
  9. You may go to the previous memory location with B (Back).
  10. You may advance forward automatically with A (Automatic).
  11. To relocate, press RESET, 1, D, 0170, RESET. Then make four patches:
    - at 0D08 put 0D
    - at 0D09 put 0D
    - at 0D7D put 0D
    - at 0D7E put 93
- Now memory checker is at 0D00 to 0DFF.
12. To use memory checker after relocation to high memory press 1, D, 0D00.

The program listing follows. Questions, comments or suggestions are welcome.

zero page memory locations:

M1 = 0030  
M2 = 003A  
M3 = 003B  
M4 = 003C

symbols I use for 6808 addressing modes:

# is direct (zero page)  
@ is extended  
\$ is indexed  
# is immediate

ADDRESS

```
0020
002E
002F
0030 7E USER 1 .JMP START
0031 00
0032 60
0033 7E USER 2 .JMP START
0034 00
0035 60
0036 7E USER 3 .JMP START
0037 00
0038 60
0039 M1
003A M2
003B M3
003C M4
003D
```

```
006A
006B
006C
006D 01 START NOP
006E 01 NOP
006F 01 NOP
0070 80 JSR CLADIS clear display
0071 F6
0072 58
0073 3F EAL enable robot language
0074 72 SPW @ say "enter address"
0075 00
0076 7A
0077 83 DAL disable robot language
0078 20 BRA 0099 branch to START2
0079 0F
007A 02 E phonemes for "enter address"
007B 00 N
007C 00 T
007D 2A E
007E 3A R
007F 03
```

```
0080 00 A
0081 00 G
0082 1E D
0083 2B P
0084 02 E
0085 1F S
0086 03 S
0087 FF
0088 FF
0089 80 START2 JSR INB get address
008A F7
008B 96
008C 97 STAR #M1
008D 39
008E 80 JSR INB
008F F7
0090 96
0091 97 STAR #M2
0092 3A
0093 0E LDX #M1
0094 39
0095 96 LDAA #M1 hi byte
0096 39
0097 80 BSR SPEX say the byte in hex
0098 41
0099 96 LDAA #M2 low byte
009A 3A
009B 80 BSR SPEX say the byte in hex
009C 3D
009D 80 START3 JSR CLADIS display the address
009E F6
009F 58
00A0 0F STX #M1
00A1 39
00A2 96 LDAA #M1
00A3 39
00A4 80 JSR OUTBYTE
00A5 F7
00A6 80
00A7 96 LDAA #M2
00A8 3A
00A9 80 JSR OUTBYTE
00AA F7
00AB 80
00AC 96 LDAA #0 display the byte
00AD 00
00AE 80 JSR OUTBYTE
00AF F7
00B0 80
00B1 A6 LDAA #0 say the byte
00B2 00
00B3 80 BSR SPEX
00B4 25
00B5 08 INX next address
00B6 80 JSR INCH get the keypress
00B7 F7
00B8 77
00B9 81 CMPA #0F is it an F?
00BA 0F
00BB 27 BEQ START3 yes, go do it again.
00BC E0
00BD 81 CMPA #0B is it a B?
00BE 08
00BF 26 BNE AUTO no, then it must be A
00C0 04
00C1 09 DEX yes, then decrement address
00C2 09 DEX
00C3 20 BRA START3 and go do it.
00C4 08
00C5 80 AUTO JSR CLADIS
00C6 F6
00C7 56
00C8 01
00C9 96 AUTO2 LDAA #0 get the byte
00CA 00
00CB 80 BSR SPEX say it
00CC 00
00CD 08 INX next address
00CE 80 BSR PAUSE pause for user to check his listing
00CF 02
00D0 20 BRA AUTO2
00D1 F7
00D2 3F PAUSE EAL enable robot language
00D3 C6 LDAB #C0
00D4 C0
00D5 5A DECS
00D6 26 BNE
00D7 FD
00D8 83 DAL disable robot language
00D9 39 RTS
00DA 36 SPEX PSRA Speak the byte in HEX
00DB 44 LSRA
00DC 44 LSRA
00DD 44 LSRA
00DE 44 LSRA
00DF 60 BSR SP1B
00E0 06
00E1 32 PULA
00E2 84 ANDA #0F
00E3 0F
00E4 80 BSR SP1B
00E5 01
00E6 39 RTS
```

00E7 DF	SPIB STX #M3	SPeak the nIBble	0131 1F	S	
00E8 3B			0132 02	E	
00E9 CE	LDX #TABLE		0133 0F	U	
00EA 01			0134 0A	E	
00EB 00			0135 00	N	
00EC 40	TSTA		0136 03		
00ED 26	BNE		0137 FF		
00EE 07	00F6		0138 05	E	
00EF 3F	ERL	enable robot language	0139 05	I	
00F0 62	SPW #0		013A 29	G	
00F1 00			013B 2A	H	
00F2 83	DRL	disable robot language	013C 2A	T	
00F3 DE	LDX #M3		013D 03		
00F4 3B			013E FF		
00F5 39	RTS		013F 00	N	
00F6 08	INX		0140 15	I	
00F7 08	INX		0141 00	N	
00F8 08	INX		0142 29	E	
00F9 08	INX		0143 00		
00FA 08	INX		0144 03		
00FB 08	INX		0145 FF		
00FC 08	INX		0146 06	A	
00FD 4A	DECA		0147 21		
00FE 20	BRA		0148 29		
00FF EC	00EC		0149 29		
0100 12	TABLE Z		014A 03		
0101 3C	E		014B FF		
0102 2B	R		014C FF		
0103 26	0		014D 0E	B	Beta is used because
0104 03			014E 06	E	"Bee" is hard to distinguish
0105 FF			014F 21	T	from "Dee".
0106 FF			0150 2A	A	
0107 2D	0		0151 31		
0108 32	N		0152 03		
0109 31	E		0153 FF		
010A 00			0154 1F	C	
010B 03			0155 3C		
010C FF			0156 29		
010D FF			0157 03		
010E 2A	T		0158 FF		
010F 36	M		0159 FF		
0110 37	0		015A FF		
0111 37			015B 1E	D	
0112 03			015C 02	E	
0113 FF			015D 18	L	
0114 FF			015E 2A	T	
0115 39	T		015F 31	A	
0116 2B	H		0160 03		
0117 3C	R		0161 FF		
0118 29	E		0162 3C	E	
0119 03	E		0163 29		
011A FF			0164 03		
011B FF			0165 FF		
011C 10	F		0166 FF		
011D 35	0		0167 FF		
011E 34	U		0168 FF	F	
011F 2B	R		0169 02		
0120 03			016A 1D		
0121 FF			016B 1D		
0122 FF			016C 03		
0123 10	F		016D FF		
0124 15	I		016E FF		
0125 00	U		016F FF		
0126 29	E		0170 CE	RELOCATOR LDX #0060	
0127 0F			0171 00		
0128 03			0172 60		
0129 FF			0173 A6	LDAA #0	
012A 1F	S		0174 00		
012B 08	I		0175 B7	STAR 00000	
012C 09	X		0176 00		
012D 19			0177 00		
012E 1F			0178 08	INX	
012F 03			0179 7C	INC 00177	
0130 FF			017A 01		
			017B 77		
			017C 26	BNE 0173	
			017D F5		
			017E 39	RTS	
			017F		
			0180		