

ET-19

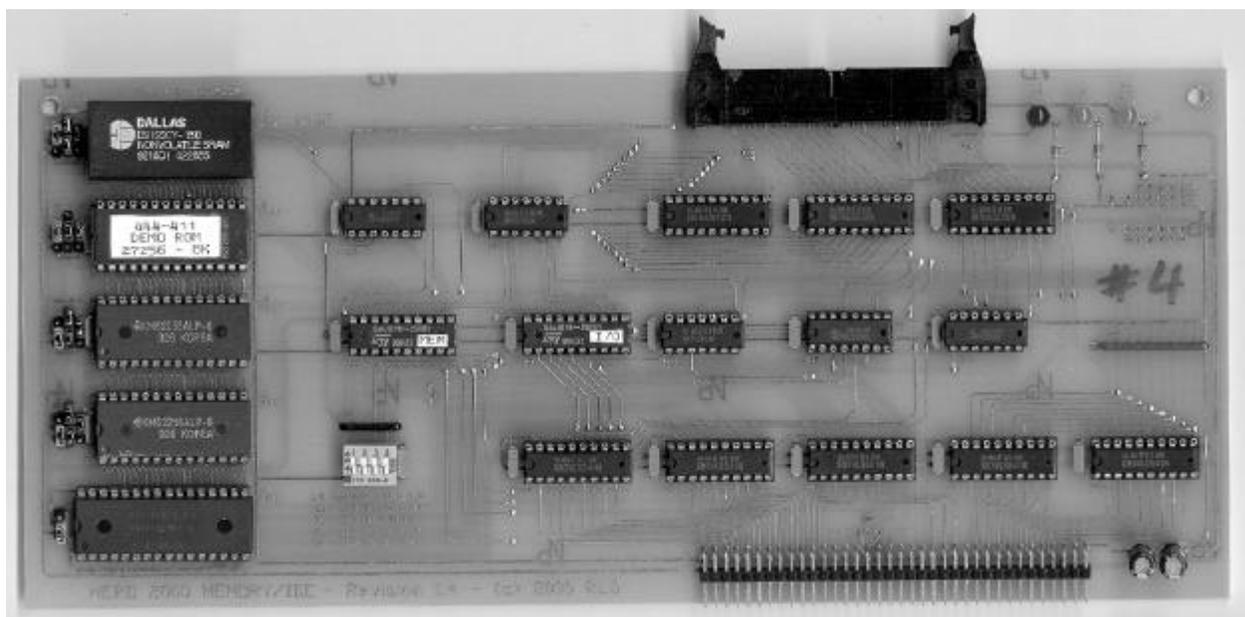
ROBOT

MEMORY/IDE

EXPANSION

(H2KIDE)

HERO 2000 ROBOT Memory/IDE Expansion Accessory (H2KIDE)



By Robert L. Doerr

<http://www.robotworkshop.com>

© 2000

Manual Revision 1.5

NOTE: This optional enhancement board has been tested in several HERO 2000 robots using different configurations with excellent results. Although every effort has been made to ensure the compatibility with the widest range of devices (and the accuracy of the documentation) there are no guarantees expressed or implied that it will work under all circumstances. For best results either a laptop IDE drive or CompactFlash module (with adapter) should be used.

Table of contents

INTRODUCTION3

CIRCUIT ASSEMBLY3

INSTALLATION6

OPERATION7

GETTING PROGRAMS INTO THE HERO 20007

USING THE TEST AND EXPLORER UTILITY8

FILE AND BOOT PROGRAMS10

ADDING HARD DRIVE SUPPORT TO USER PROGRAMS13

SPECIFICATIONS13

MEMORY MAP13

JUMPER SETTINGS13

I/O PORT ASSIGNMENTS14

IN CASE OF DIFFICULTY14

CIRCUIT DESCRIPTION15

PLD SOURCE CODE15

HERO 2000 BUS DEFINITION18

INTERRUPT DEFINITIONS19

REPLACEMENT PARTS LIST20

SPECIAL THANKS21

HERO 2000 Memory/IDE Expansion

INTRODUCTION

This option for the HERO 2000 robot provides up to 640K of additional memory and an integrated IDE controller. Originally it would take three memory cards populated with 72 memory chips (plus logic) just to reach 576K. This card provides 576K with only three memory chips and allows for 64K more than previously possible. The memory consists of a single 512K static memory chip while the remaining 128K can either be RAM, NVRAM, or ROM. Jumpers are provided to select the appropriate chip type. If configured with contiguous RAM the robot can recognize all 640k (9FFFF) as useable RAM. Another feature of this card allows the last memory socket to respond to address 98000h or A0000h. This was done since Heathkit originally used location A0000h for a factory diagnostic ROM that would automatically execute upon startup. This is a feature that may be used in the future.

Most IDE devices use a 16-bit data bus while the HERO 2000 has an 8-bit data bus. This IDE controller contains additional circuitry to latch the upper 8-bits for a full 16-bit data path. With the extra latch the full capacity of the IDE device can be used. Since this interface does not have any of the BIOS limitations that plague early PC's it should support any drive up to the maximum allowed by the IDE specification (127gb). Using all 16 bits is important since ignoring the upper 8-bits will cut the useable capacity of the IDE device in half. Also, the IDE device settings (manufacturer, geometry, etc) would not be readable since half the data is missing. This is the method that many others have chosen for interfacing IDE devices to 8-bit micros. Although using only 8 bits would have made this card much easier to design, all 16 bits are used to make the best interface possible.

To simplify the design of this card, the DMA feature is not supported. The IORDY line is also ignored so the higher speed PIO modes cannot be used. This was done intentionally since most of these modes could transfer data faster than the robot could ever handle. Since there was no benefit, they were left out to conserve power and reduce complexity. The IDE portion of this card supports the regular PIO modes without interrupts. Although interrupts are not currently used that functionality can be easily added by installing a 74LS06 open collector inverter and a few jumper wires. There is also a space for a header and jumper block to select an IRQ if desired.

NOTE: Although the current driver and examples only support hard drives there is no reason that the card could not support other IDE devices such as CD-ROM, ZIP, or Tape drives. Full source code is provided for the driver so it can be customized for future device support. Many other devices use the same physical IDE interface but use a PACKET Command structure that is different than the standard hard protocol. The exception would be a CompactFlash memory module (with proper adapter) since it works and acts like a standard IDE hard drive.

If the Memory/IDE expansion board is already assembled, please proceed to the Installation section.

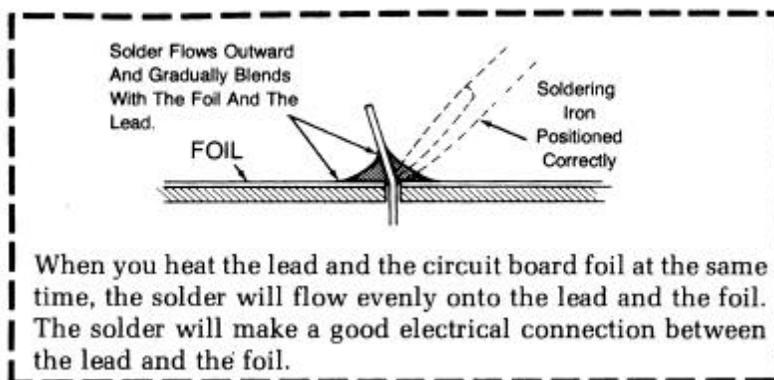
CIRCUIT ASSEMBLY

(Kit version only)

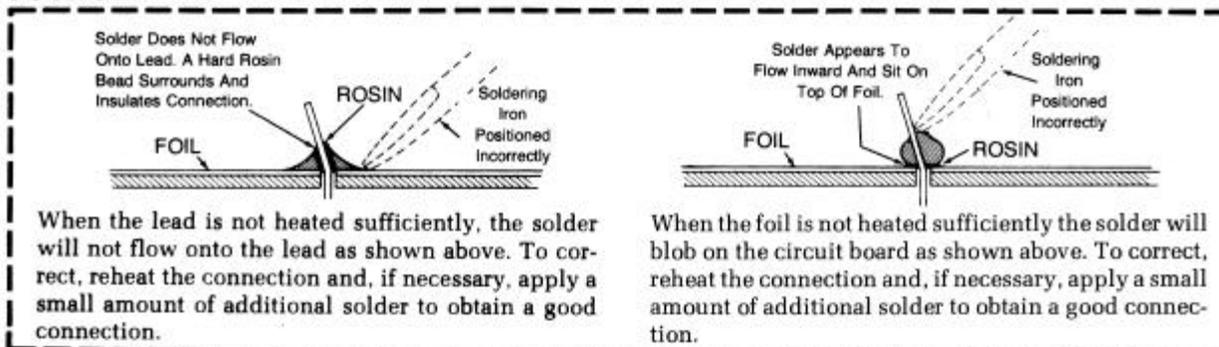
This kit should be built using standard construction methods. The following items are required to build the kit: diagonal cutter, needle nose pliers, soldering iron (pencil type) with fine tip, 60/40 Rosin core solder, and some patience. Follow the instructions carefully and read the entire step before performing each operation. To successfully assemble this kit you must have good soldering skills. A good solder connection will form the electrical connection between two parts, such as a component lead and a circuit board foil. Care also needs to be taken to ensure that there are no solder bridges causing shorts. A bad solder connection could prevent an otherwise well-assembled kit from operating properly.

During assembly make sure you keep the soldering iron tip clean. Wipe it often on a wet sponge or cloth; then apply solder to the tip to give the entire tip a wet look. This process is called tinning, and it will protect the tip and enable you to make good connections. When the solder tends to "ball" or does not stick to the tip, the tip needs to be cleaned and retinned.

A GOOD SOLDER CONNECTION



POOR SOLDER CONNECTIONS



The component side of the board says HERO 2000 – MEMORY/IDE – Revision 1.5." on the bottom left. All of the components will be mounted on the component side of the board. During construction the top of the board with the IDE connector (P2) should be up and the memory sockets (U1-U5) should be on the left. After installing each component at the specified location solder it in place before proceeding to the next. When installing DIP sockets it works best if each of the two opposing corners is soldered first. Then gently press on the center of the socket while warming the solder on those corner pins with the soldering iron to make sure the socket is properly seated. Finally solder the rest of the leads on the socket. This method gives the board a much cleaner look than if the leads on the sockets have been folded over to hold it in place before soldering.

NOTE: A small axial capacitor will be soldered just to the left of all the chips (U1-U20) on the board and one to the right of the 4 position DIP switch. The spacing is slightly wider between the pads for the capacitor and the pads for the chips. This was done to prevent the chip sockets from being accidentally installed into the spot for the capacitors. Make sure that the notch in each socket is located toward the left and that pin one goes into the pad with the square foil.

- Install the three 510ohm resistors at locations R1, R2, and R3. The location marked R4 is unused.
- Install the 32-pin socket at location U1. The notch should point toward the left of the board.
- Install the four 28-pin sockets at locations U2, U3, U4, and U5. The notch should point toward the left of the board.
- Install the 20-pin sockets at U6, U8, U9, U11, U13, U14, U16, U17, U19, and U20. The notch should point toward the left of the board.
- Install the 16-pin socket at U15. The notch should point toward the left of the board.

HERO 2000 Memory/IDE Expansion

- Install 14-pin sockets at locations U7, U10, U12, and U18. The notch should point toward the left of the board.
- Install the .01uf or .1uf monolithic axial or DIP capacitors to the left of each IC socket and one to the right of the DIP switch. These can be inserted in either direction since polarity does not matter.

NOTE: When installing the headers insert the shorter leads into the PCB. The longer lead is where the configuration jumpers will go.

- Install the four 4-pin (1x4) headers to the left of the memory sockets U2, U3, U4, and U5.
- Install the nine 3-pin (1x3) headers to the left of the memory sockets U1, U2, U3, U4 and U5. There should be one next to U5 and two next to the other four sockets.
- Install the 72-pin connector at location P1. The side of the connector with the bent leads is inserted into the board from the component side. Make sure the connector is level with the board. Proper alignment is critical since the card will not seat properly if this is not installed correctly.
- Install the 40-pin (2x20) connector at location P2.
- Install the 4 position DIP switch. Switch 1 should be installed toward the left.
- Install RP1 above the 4 position DIP switch. The common lead (dot/stripe) should be installed toward the left.
- Install RP2. The common lead (dot/stripe) should be installed toward the edge of the board on the right. The large round pad to the left of the resistor pack should remain open for future IRQ support.
- Install the 33uf electrolytic capacitors at locations C1 and C2. Observe the polarity as you do this. The striped side (-) should face down, away from the (+) sign on the board. Installing these parts the wrong way is bad, very bad.

OPTIONAL: Two pin headers can be used instead of LED's in the following step. This will allow the use of a remote LED that can be mounted for viewing outside the robot.

- Install the RED LED at location IDE and the two GREEN LED's at locations +5v and +5v Sleep. The flat goes down facing the bus connector P1. A diode symbol is etched on the board to show polarity.
- OPTIONAL:* Fill in the via's with solder. These are the left over holes that connect top traces to traces on the other side of the board. Over time and from expansion/contraction of the board material (Not to mention flexing of the board) small cracks can develop in the via's. In order to prevent this from happening it is best to fill them in with solder. It might be necessary to heat both sides of the via in order to get a clean connection.
- OPTIONAL:* De-flux and clean the board. This step is not required but when properly cleaned the board's appearance is better and it is easier to spot cold solder joints and solder bridges. Depending upon the type of flux used, this can be done economically using common rubbing alcohol and an old toothbrush.
- Install the three 74ALS573 chips in sockets U13, U16, and U19. The notch should point toward the left.
- Install the five 74LS541 chips in sockets U8, U11, U14, U17, and U20. The notch should point toward the left.

HERO 2000 Memory/IDE Expansion

- Install the 74LS00 chip in socket U7. The notch should point toward the left.
- Install the 74LS02 chip in socket U10. The notch should point toward the left.
- Install the 74LS04 chip in socket U12. The notch should point toward the left.
- Install the 74LS138 chip in socket U15. The notch should point toward the left.
- Install the 74LS32 chip in socket U18. The notch should point toward the left.
- Install the memory decoding PLD (MEM) in the socket at U6. The notch should point toward the left.
- Install the I/O decoding PLD (I/O) in the socket at U9. The notch should point toward the left.
- Install a 512K x 8 static memory chip in the 32 pin socket at U1. The notch should point toward the left.

This concludes the Assembly procedures for the HERO 2000 Memory/IDE Expansion. Congratulations! Before proceeding, look over the board and verify the correct location and orientation of all parts. Also check to make sure there are no solder bridges or poor solder joints. Some of the traces are very close together and it is easy to accidentally create a solder bridge across a trace or two.

INSTALLATION

Before installing this board make sure that the jumper on the CPU board is set to the H position. This will relocate the memory on the CPU board to the high area where it will not conflict with this card. If the robot previously had any ET-19-15 memory boards then this jumper is probably configured already. In that case remove any memory boards (ET-19-15) that may be installed in the robot.

Install any extra memory and optional ROM's on the board. Each memory socket has jumpers to configure the chip type and power source for the socket. The only exception is the first memory socket U1 that only has a jumper for the power source. Please see the chart on configuring the memory for the location of each jumper. An NVRAM chip such as the DS1230Y is usually installed in U5 to store a boot utility for loading programs from the hard drive. Switch 1 should remain off to make it respond to address 98000h for use with the included BASIC driver programs. If installing an Autoboot ROM in location U5 then turn switch 1 on to make the socket respond to the memory bank at A0000h instead of 98000h.

This card will work in any free expansion slot. It is normally installed in one of the slots just to the left of the I/O card so that the IDE cable can come through the RF shield assembly. When the RF shield is not used then any available slot can be used.

NOTE: Due to the larger current draw of normal desktop 3.5" hard drives it is strongly recommended that only 2.5" laptop IDE drives be used with the robot. All that is required is a small adapter kit, which allows these small drives to be used with a standard 40 pin IDE cable and power cable. Another advantage with the Laptop drives is that most have a better tolerance against vibration and shock which the robot can very well inflict on a hard drive. The drive motor can also be turned off when the drive is not needed and further reduce the chance to a drive failure due to shock. If the robot does not have a floppy drive option then a great way to mount the hard drive is with a 3.5" to 5.25 bay adapter. They will bolt right in place above the arm where the floppy drive mounts. If the robot is equipped with a floppy drive then the hard drive can be wrapped in foam and placed next to the right of the arm near the bottom of the torso. A "Y" cable for the power plug will also be needed. These are available at most computer stores.

HERO 2000 Memory/IDE Expansion

OPERATION

The robot will automatically recognize the memory on this card. Total RAM is determined by the amount of contiguous RAM found. If the card has a 512K chip and four 32K chips then it will recognize a full 640K (9FFFF) of memory. During the initial memory scan if a ROM is found then the max available RAM is the point just before the ROM was found. The card is normally configured with one 512K chip, two 32K chips, DEMO ROM (or empty), and a 32K NVRAM. In this case the robot will respond with 576K of memory available (8FFFF), which is the equivalent of three fully populated ET-19-15 memory boards.

NOTE: When the DEMO ROM is installed on this card it will respond and execute the same as it would on the CPU board (execute 016). The only difference is that it appears in the memory map four times and can also be executed and the next three addresses (execute 017, 018, and 019). This assumes that it is installed in U4 and sockets U1-U3 do not contain ROM's or a ROM signature poked into memory at an 8K boundary.

In order to use the IDE controller a program that contains the I/O routines and commands for accessing the drive must be loaded into the robot. The programs supplied are written in BASIC that use some machine code routines to improve performance. There is a utility for testing and exploring a hard drive, a program to store user programs on the hard drive, and a boot program that can be stored in the NVRAM to read and execute programs previously stored on the hard drive.

GETTING PROGRAMS INTO THE HERO 2000

One of the most important tasks to master is getting programs into the HERO 2000 robot. All you need is a computer running a terminal package and a standard external modem cable that ends in a DB25 male connector. Although the following example uses a Windows based PC almost any system (DOS, UNIX/Linux, or Macintosh) can be used as long as you have the proper cable and a terminal package.

Connect the cable to the top serial port on the back of the HERO 2000 robot. Start the terminal package (HyperTerminal) and use a direct cable connection. The Serial parameters should be 9600 baud, 8 data bits, 1 stop bit, and no-parity. Most robots are configured to power up at 9600 baud but if not it can be changed using the keypad before entering BASIC or changing the DIP switches on the CPU board. If using HyperTerminal you will want to tweak the ASCII settings the first time. Go under File, Properties, Settings, ASCII setup. Set the line delay at 10ms and the character delay at 2ms. This ensures the robot will not lose characters during the transmission of program files to the robot.

On the robots keypad press 1 to enable the serial port, then press 0 to start BASIC. You should get a prompt similar to the one below (memory size may be different if a program is loaded in memory):

```
HERO-2000 Basic
63509 Bytes free
OK
```

Once you are at this point you can use the PC as the display and keyboard for the robot to create new programs or you can download an existing program (stored as an ASCII file on your PC) to the robot. If there is already a program loaded in memory then type NEW before sending your next program. To send a program to the robot you use Transfer, Send Text File. Select the file you want to send and press open. The program will then be displayed as it is sent to the robot. At this point the robot just thinks a program is being keyed in. When it is done you can RUN the program, modify the program, or just start typing immediate mode BASIC commands. In the event you want to store a program on your PC just enable the session capture log, type LIST at the BASIC prompt, and end the capture once the listing completes. At that point a text editor can be used to remove the LIST command and any trailing lines.

HERO 2000 Memory/IDE Expansion

USING THE TEST AND EXPLORER UTILITY

Included with this card is a program called H2KTESTxxx.TXT (where xxx is the version number) that is used to test the operation of the IDE controller. It will let you query the drive for the identity and parameters of the device as well as allow you explore the hard drive and examine any sector. It contains the code for all the routines needed to read/write an IDE hard drive. This code can be incorporated into your own programs to enable them to store and retrieve data. Below is a brief example of the test program and some sample commands.

```
run
50
50
Port: 60
Device: A0
Track MSB: 0
Track LSB: 0
Sector: 1
Number: 1
Command: (S,I,P,R,W,D,A,C,TH,TL,H,Q,0,8,F,Z,RM,WM,N,L,B, or X) ? S
Software Reset...
Device Signature: 01 01 00 00 A0
Standard Device.
Port: 60
Device: A0
Track MSB: 0
Track LSB: 0
Sector: 1
Number: 1
Command: (S,I,P,R,W,D,A,C,TH,TL,H,Q,0,8,F,Z,RM,WM,N,L,B, or X) ? I
Identify device.
STATUS = 58 2796 Logical Cylinders
 16 Logical heads
 63 Logical sectors per logical track
Model: IBM-DMCA-21440
Serial #: C50C51S5285
2796 Current cylinders
 16 Current heads
 63 Current sectors per track
 0 LBA sectors addressable
 0 Sectors capacity
 16 Maximum sectors per READ/WRITE MULTIPLE
Identify device done.
Port: 60
Device: A0
Track MSB: 0
Track LSB: 0
Sector: 1
Number: 1
Command: (S,I,P,R,W,D,A,C,TH,TL,H,Q,0,8,F,Z,RM,WM,N,L,B, or X) ?
```

After running the program, 'S' is used to perform a software reset on the drive, and then 'I' is used to display the identity of the device. The parameters returned will vary depending on the actual device used. Other commands are 'D' to display the sector buffer, 'R' to read the currently addressed sector into the buffer, 'W' to write out the contents of the buffer to the currently addressed sector, 'TH' & 'TL' to set the track (cylinder) for either a read or write. See the comments within the test program that explain the rest of the choices accepted by the program. To fully understand how the program and an IDE drive works it is useful to reference the port usage chart later in this document. Also, a copy of the actual IDE ATA specifications and standards can be found on the Internet that covers each one of the IDE commands in detail.

HERO 2000 Memory/IDE Expansion

Sample code from test program:

```
200 INPUT"Command: (S,I,P,R,W,D,A,C,TH,TL,H,Q,0,8,F,Z,RM,WM,N,L,B, or X) ";YJ$
210 IF YJ$="S" THEN GOSUB 41000:GOTO 100:REM Software reset
220 IF YJ$="I" THEN GOSUB 43000:GOTO 100:REM Identify device to buffer
230 IF YJ$="P" THEN GOSUB 44000:GOTO 100:REM Set parameters (not done)
240 IF YJ$="R" THEN GOSUB 48000:GOTO 100:REM Read sector from disk to buffer
250 IF YJ$="W" THEN GOSUB 49000:GOTO 100:REM Write sector buffer to disk
260 IF YJ$="A" THEN GOSUB 58000:GOTO 100:REM Swap MSB/LSB bytes in buffer
270 IF YJ$="D" THEN GOSUB 59000:GOTO 100:REM Display sector buffer contents
280 IF YJ$="C" THEN GOTO 39900:REM Change the sector number
290 IF YJ$="TH" THEN GOTO 39800:REM Change track MSB
300 IF YJ$="TL" THEN GOTO 39850:REM Change track LSB
310 IF YJ$="H" THEN GOTO 39700:REM Change head number
320 IF YJ$="Q" THEN GOSUB 39600:REM Read & display status register
330 IF YJ$="0" THEN GOSUB 39500:REM Read & display lower data & status regs
340 IF YJ$="8" THEN GOSUB 39400:REM Read & display upper data & status regs
350 IF YJ$="F" THEN GOSUB 39200:REM Fill sector buffer with string and zeros
360 IF YJ$="Z" THEN GOSUB 30000:REM Fill 256 tracks and read back (Seagate)
370 IF YJ$="RM" THEN GOSUB 30900:REM Read multiple sectors to memory
380 IF YJ$="WM" THEN GOSUB 32900:REM Write multiple sectors from memory
390 IF YJ$="N" THEN GOTO 32000:REM Change the sector count
400 IF YJ$="L" THEN GOSUB 34000:REM Idle immediate command
410 IF YJ$="B" THEN GOSUB 34100:REM Standby immediate command
490 IF YJ$="X" THEN END:REM Exit the program
```

HERO 2000 Memory/IDE Expansion

FILE AND BOOT PROGRAMS

The H2KFILE program is a utility that lets you store HERO 2000 BASIC programs on a hard drive. It is sent to the robot then copied to an unused portion of memory. At that point you can type in your HERO 2000 program or download it from a PC into the robots BASIC workspace. Once there it is copied to a section of memory designated as a 'buffer' for transferring BASIC programs to/from the hard drive. The file program is then recalled and executed to write the program in the buffer to a location you select on the hard drive. To send any other programs just type new to clear out the file program and send down the next application. Move it to the transfer buffer and then recall a copy of the file program again to write it out selecting a different location to prevent overwriting an existing program.

NOTE: It is also possible to save the file program itself on the hard drive so the boot program can load it whenever you want to write another program to the hard drive.

Type `NEW` and download the program called H2KFILExxx.TXT (where xxx is the version number) into the robot.

Type `RUN`.

After a short delay, you should see some information about your hard drive on the terminal screen.

example:

```
900 logical cylinders
15 logical heads
17 logical sectors per track
model ST9145AG
1041 current cylinders
15 current heads
16 current sectors per track
...etc.
```

The model is your hard drive type. In this example a Seagate 120 megabyte drive is shown. The logical values are the drive manufacturer's arrangement of cylinders, heads, and sectors. The current values are the CHS emulation values set and used by the program. The current cylinders will vary depending on the size of your hard drive.

NOTE: If the current heads is not 15 or the current sectors per track are not 16 then this program won't work properly. In that case you need to try a different hard drive or these programs will need to be altered to support your device. Most drives tested did respond properly.

NOTE: All the commands accepted by the programs are capital letters.

For now, type `X` to exit the program. Then type:

```
DEFSEG #3000
RELOC TO 0
```

This copies the program to the beginning of the fourth 64K segment of your RAM so it can be recalled later.

Now type `NEW`, and load the program that you want to save to your hard drive (make it something different than the H2KFILE program so you can recognize whether the read from the hard drive works later). Then type:

```
DEFSEG 7779
RELOC TO 0
```

HERO 2000 Memory/IDE Expansion

This puts your program in the 56K data buffer (at address 7779) for transfer to the hard drive. Now type:

```
DEFSEG #3000  
RELOC FROM 0
```

This recovers the H2KFILE program from the fourth 64K segment of RAM. A copy still remains at segment #3000. As long as you haven't turned off Hero 2000's power (sleep mode doesn't change RAM data) or written over the program with something else, it can be recovered this way rather than having to download it via RS232 again.

Now type `RUN` and, after a short delay, see the hard drive information again on the screen.

Now type `S` (remember that it must be capitalized).

There will be a prompt for a program number and a range of valid numbers will be shown.

Using a Segate 120mb drive the program says `Program # (1 - 2230)`. The high number depends on the size of your hard drive, and it is the maximum number of 56K blocks containing programs or data that you can save to your hard drive.

NOTE: You will want to save each of your programs to a different program number to avoid overwriting one program with another (unless that is your intention).

Type a number in the range shown and hit return (make note of the number and a description of the program, since there are no filenames, only the number). You will see a series of characters indicating that sectors are being written to your hard drive from the 56K buffer.

Now type `x` to exit the H2KFILE program.

At this point you can repeat the process to save other programs to the hard drive. If you want to manually load programs from the hard drive or verify the operation by loading a program manually then follow the steps below:

Let's see if we can load your program from the hard drive. First, type:

```
DEFSEG 7779  
RELOC TO 0
```

This will put a copy of the H2KFILE program in the 56K buffer, just to make sure that the program on the hard drive is not in the buffer to begin with.

Now type `RUN`, and after the delay and the disk info, type `L`, and at the program # prompt, type the number that you used before to save the program to the hard drive. You should see a series of characters indicating that sectors are being transferred from the hard drive to the 56K buffer.

When its done, type `x` to exit the program. Now type:

```
DEFSEG 7779  
RELOC FROM 0  
LIST
```

If all went well, you should see a listing of the program that you saved to the hard drive, and it should be ready to use by typing `RUN`.

HERO 2000 Memory/IDE Expansion

The H2KBOOT program will allow loading and running a program from the hard drive without connecting a PC to the RS232 port. To do that, you need to have this program in a section of memory that the HERO 2000 thinks is read-only memory. This is where a Dallas DS1230Y nonvolatile RAM chip is extremely useful. It should reside in U5 which is the highest ROM/RAM socket on the IDE/memory board. Switch 1 should be off so that socket U5 is at segment #9800 in the memory map. Even though the nonvolatile RAM can be modified like ordinary RAM by the robot's software, the robot thinks that it is ROM because there is no RAM at segment #9000. When the robot is powered on, it searches memory to determine the amount of contiguous RAM. When it finds no RAM at segment #9000, it stops searching, and assumes that anything above that is ROM memory. This only works if U4 is empty or contains a ROM like the DEMO ROM. To complete the illusion two bytes that represent a ROM signature are poked into the first two locations of the NVRAM so it appears to be an option ROM.

Type `NEW` and download the program called `H2KBOOTxxx.TXT` (where `xxx` is the version number) into the robot. To put the boot program into the nonvolatile ROM follow these steps:

```
DEFSEG #9800  
RELOC TO 2
```

Why 2? Well, we want to leave room for two bytes that signify to the Hero 2000 that there is a ROM at segment #9800. In order to do this we `POKE` in two bytes `#AA #55` that tell the HERO 2000 it has an option ROM installed. Now type:

```
POKE 0,170  
POKE 1,85
```

That's it! Now type `NEW` just to be sure that the boot program is not in the BASIC workspace. Then press reset on the robot. Press key 9, exec, on the robot, then press the 0 1 6 keys (or 0 2 0 keys if a DEMO ROM is installed). This should load the boot program from the nonvolatile RAM to the BASIC workspace and start running it. When the characters "Prg no" appear in the LED display, on the keypad, press the five digit number of a program that was saved on the hard drive with the file program. After a slight delay, the program will load and automatically run.

This sequence does not require that the PC be connected to the RS232 port at all.

NOTE: The 9 exec key is used to run the built-in HERO 2000 programs and option ROM chips. The first 16 entries (000 thru 015) are reserved for programs within the main system ROM chips. Any option ROM chips will be numbered 016, 017, 018, 019, 020, etc... until no more ROM signatures are found. The HERO 2000 searches for option ROM chips in 8K increments. If the DEMO ROM is installed in U4 it actually shows up in the memory map 4 times. This is normal since that socket has a 32k address space and the DEMO ROM is an 8K chip. The upper two address lines are ignored and the chip responds to all four 8k banks within the 32k space of the socket. In the case of the DEMO ROM located on the CPU board the H2KBOOT program would respond to exec 016 and the DEMO ROM to exec 017. It all depends upon where the chips reside.

HERO 2000 Memory/IDE Expansion

ADDING HARD DRIVE SUPPORT TO USER PROGRAMS

Once you are familiar with the operation of the hard drive and the supplied programs you may decide to add hard drive support to your own programs. The easiest way to proceed is to append the appropriate code from the test utility. Since there isn't a regular file system on the hard drive, your program will have to determine where to store the information. Set aside the first part of the disk for programs being loaded by the boot program. Your program can then set the address for a cylinder, head, and sector beyond that used for program storage. If you want to read data then call the subroutine to read a sector and access the contents from the sector buffer. If you want to write data then put your information in the sector buffer then call the write subroutine. Most of the hard drives have plenty of room and you can usually have several programs setup like this which all use their own portion of the disk.

SPECIFICATIONS

Power: 5v, 5v Sleep

System bus speed: 5mhz, 0 wait states

Memory sockets: 5 (socket 1 RAM only, others ROM, RAM, or NVRAM)

Max Memory: 640K

Max contiguous RAM directly recognized by robot: 640K

IDE controller I/O access: Polled I/O (PIO mode 0 without IORDY)

IDE controller I/O address: 60h, 70h, 80h, or 90h selectable by switches 3&4

IDE data path: 8 bit and 16 bit

MEMORY MAP

Below is the memory range used by the card and how each memory socket fits in the memory map. The first three sockets U1-U3 are the equivalent to three factory ET-19-15 memory boards.

Socket number	Address range	Address size	Chips supported	Notes
U1	00000-7FFFF	512k	512k static RAM only	(RAM) Base memory
U2	80000-87FFF	32k	6264, 62256, 2764, 27128, 27256, DS1230	(RAM), ROM, NVRAM
U3	88000-8FFFF	32k	6264, 62256, 2764, 27128, 27256, DS1230	(RAM), ROM, NVRAM
U4	90000-97FFF	32k	6264, 62256, 2764, 27128, 27256, DS1230	(Demo ROM), ROM, RAM, NVRAM
U5	98000-9FFFF or A0000-A7FFF	32k	6264, 62256, 2764, 27128, 27256, DS1230	(NVRAM), RAM, ROM, Autoboot ROM

JUMPER SETTINGS

Switch 1: Autoboot On/Off (**default**)

Switch 2: Cable Select: On/Off (**default**) – Feature not available on Revision 1.4 or older boards

Switch 3	Switch 4	I/O base address
Off	Off	60h (default)
On	Off	70h
Off	On	80h
On	On	90h

HERO 2000 Memory/IDE Expansion

I/O PORT ASSIGNMENTS

The available I/O port addresses for the HERO 2000 for optional user add-in cards range from 60 through A7. All of the other addresses are reserved or used by HERO 2000 features. Since the IDE portion of the HERO 2000 Memory/IDE card uses 16 contiguous I/O addresses it can use 60-6F (default), 70-7F, 80-8F, or 90-9F. Below is a list of how the ports map out along with the equivalent I/O for a standard PC.

Offset	Read access	Write access	PC I/O
0	Data register (lower 8 bits) <i>read first</i>	Data register (lower 8 bits) <i>write second</i>	1f0
1	Error register	Feature register	1f1
2	Sector count register	Sector count register	1f2
3	Sector number	Sector number	1f3
4	Cylinder bits 0-7	Cylinder bits 0-7	1f4
5	Cylinder bits 8-15	Cylinder bits 8-15	1f5
6	Drive/Head	Drive/Head	1f6
7	Status register	Command Register	1f7
8 **	Data register (upper 8 bits) <i>read second</i>	Data register (upper 8 bits) <i>write first</i>	3f0
9	Not used	Not used	3f1
10	Not used	Not used	3f2
11	Not used	Not used	3f3
12	Not used	Not used	3f4
13	Not used	Not used	3f5
14	Alternate status register	Control register	3f6
15	Reserved	Not used	3f7

** The port at offset 8 is used as a latch for the upper 8 bits of the Data register.

IN CASE OF DIFFICULTY

PROBLEM	POSSIBLE CAUSE
Robot Does not power up after Memory upgrade is installed.	<ol style="list-style-type: none"> 1. Check for solder bridges and cold solder joints. 2. Make sure memory jumper is set to 'H' on CPU board. 3. Check for bent/folded pins on all logic IC's. 4. Check for proper operation of the 5V regulator. 5. Check for both Green power lights. 6. Check orientation of resistor packs
Can't access extra memory or one of the Option ROM's	<ol style="list-style-type: none"> 1. Verify jumper settings for each chip and verify the chip type. 2. Make sure memory jumper is set to 'H' on CPU board 3. Check for bent pins on all chips. 4. Check setting for Autoboot enable.
Can't access IDE device	<ol style="list-style-type: none"> 1. Check drive power. 2. Check I/O address of card. 3. Verify Master/Slave setting on drive.
Erratic operation	<ol style="list-style-type: none"> 4. Check for solder bridges and cold solder joints. 5. Check for bent pins on all chips. 6. Make sure memory jumper is set to 'H' on CPU board. 7. If using an IDE drive it might be drawing too much power. Use a drive with lower power requirement.

HERO 2000 Memory/IDE Expansion

CIRCUIT DESCRIPTION

This board provides up to 640K of memory and an IDE controller. The two separate uni-directional 8-bit data bus on the backplane of the HERO 2000 is merged into a single bi-directional data bus on the card through the two 74LS541 chips at U17 and U20. U17 buffers the data from this card into the HERO bus while U20 buffers the data from the HERO bus to this card. Both the Memory section and IDE controller share this bi-directional data bus on the card. These two chips are enabled for any memory or I/O operation within the range of this card. RP2 is just a 10K SIP resistor pack used as a pull-up for the 8 data lines on the card. The address lines are buffered from the robot bus by three more 74LS541 chips at U8, U11, and U14.

Two 74ALS573 chips handle the data read from and written to the IDE device. U13 latches the lower 8 bits of data for the IDE device while U16 handles the upper 8 bits of data for the IDE device. These enable the full 16 bit data path of the IDE interface to be used. There is also another 74ALS573 at U19 which is use to buffer control signals for the IDE device.

There are two pre-programmed GAL chips that decode the memory and I/O. They operate as a team to determine if an address is within the range of this card and if so select the appropriate chip select signals and latches depending if it is a memory read, memory write, I/O read, or I/O write. The memory PLD decodes the high address lines to select the appropriate memory socket. It also decodes a jumper that is used to select if the last memory socket goes to 98000h or A0000h. The I/O PLD chip compares the I/O address range on the system bus with the one that is selected by the jumpers on the card. The IDE controller is I/O mapped to 16 contiguous I/O addresses.

The 74LS00 at U7 is used to protect the write signals during sleep periods. Whenever the robot is in sleep mode the write lines for both memory and I/O operations are ignored to prevent memory (or data on the drive) from being corrupted.

PLD SOURCE CODE

To simplify the design of this card two Programmable Logic Devices (PLD) are used. They both act as a team to determine if a resource (memory or I/O) is being selected on the card and if so decode the address and select the proper device. The MEM PLD is primarily responsible for decoding which memory chip to access during a memory cycle. The I/O PLD is responsible for matching up if an I/O address on the card is selected. Below is the source code for the two PLD devices used on this expansion card. These can be programmed into either a GAL16V8 or PALCE16V8 device. The code was assembled using the PALASM4 - PAL ASSEMBLER - MARKET RELEASE 1.5a (8-20-92). It was published by Advanced Micro Devices, Inc and is available as shareware/freeware from many sites. The first code section is for the Memory decoding PLD and the second code section is for the I/O decoding PLD. With the following information that can always be re-created at a later date if needed.

HERO 2000 Memory/IDE Expansion

Memory decoding PLD source:

```
;PALASM Design Description

;----- Declaration Segment -----
TITLE    HERO 2000 Memory/IDE (Memory decoder PAL)
PATTERN
REVISION 03
AUTHOR   Robert L. Doerr
COMPANY  www.robotworkshop.com
DATE     02/21/2000 (Version 1), 11/28/2000 (Version 1.2)

CHIP    _H2KMEM  PALCE16V8

;----- PIN Declarations -----
PIN 2      A15      ; INPUT
PIN 3      A16      ; INPUT
PIN 4      A17      ; INPUT
PIN 5      A18      ; INPUT
PIN 6      A19      ; INPUT
PIN 7      ABT      ; INPUT
PIN 8      DBIN     ; INPUT
PIN 9      IO       ; INPUT
PIN 12     MEMIN    ; OUTPUT
PIN 13     BDBIN    ; OUTPUT
PIN 14     CS5      ; OUTPUT
PIN 15     CS4      ; OUTPUT
PIN 16     CS3      ; OUTPUT
PIN 17     CS2      ; OUTPUT
PIN 18     CS1      ; OUTPUT
PIN 19     BSEL     ; OUTPUT

;----- Boolean Equation Segment -----
EQUATIONS

/CS1 = /IO * /A19 ;

/CS2 = /IO * A19 * /A18 * /A17 * /A16 * /A15 ;

/CS3 = /IO * A19 * /A18 * /A17 * /A16 * A15 ;

/CS4 = /IO * A19 * /A18 * /A17 * A16 * /A15 ;

/CS5 = /IO * A19 * /A18 * ((/A17 * A16 * A15 * ABT) + (A17 * /A16 * /A15 * /ABT)) ;

BSEL = CS1 * CS2 * CS3 * CS4 * CS5 ;

MEMIN = /BSEL * DBIN * /IO ;

BDBIN = /MEMIN ;

;----- Simulation Segment -----
SIMULATION

;-----
```

HERO 2000 Memory/IDE Expansion

I/O decoding PLD source:

```
;PALASM Design Description

;----- Declaration Segment -----
TITLE   HERO 2000 Memory/IDE (I/O decoder PAL)
PATTERN
REVISION 03
AUTHOR   Robert L. Doerr
COMPANY  www.robotworkshop.com
DATE     11/29/2000 (Version 1.7)

CHIP    _H2KMEM PALCE16V8

;----- PIN Declarations -----
PIN 2      A4          ; INPUT
PIN 3      A5          ; INPUT
PIN 4      A6          ; INPUT
PIN 5      A7          ; INPUT
PIN 6      J1          ; INPUT
PIN 7      J2          ; INPUT
PIN 8      DIN         ; INPUT
PIN 9      DOUT        ; INPUT
PIN 12     DATOUT      ; OUTPUT
PIN 13     DATIN       ; OUTPUT
PIN 14     IOR         ; OUTPUT
PIN 15     IO          ; OUTPUT
PIN 16     DBIN        ; INPUT
PIN 18     SEL         ; OUTPUT

;----- Boolean Equation Segment -----
EQUATIONS

IO = DIN + DOUT ;

SEL = (A7 :+: /J2) + (A4 :+: /J1) + (A6 :+: A5) + (A6 :+: /A7) + /IO ;

DATOUT = /SEL * DOUT ;

DATIN = /SEL * DIN ;

IOR = /DATIN + /DBIN ;

;----- Simulation Segment -----
SIMULATION

;-----
```

HERO 2000 Memory/IDE Expansion

HERO 2000 BUS DEFINITION

The following table shows the HERO 2000 bus signals. The Termination column tells how and where the signal is terminated on the bus. "T" means the signal is terminated on the backplane. "C" means the signal is terminated on the CPU board. "O" means the signal is also terminated on the CPU board but must be driven with open-collector circuits to prevent damage. An asterisk after the Mnemonic name means that the line is active low.

Pin #	Mnemonic	Definition	Termination
1	GND		
2	GND		
3	GND		
4	+16v	Unregulated +16 VDC (up to +24 VDC)	T
5	A0	Address bit 0	T
6	A1	Address bit 1	T
7	A2	Address bit 2	T
8	A3	Address bit 3	T
9	A4	Address bit 4	T
10	A5	Address bit 5	T
11	A6	Address bit 6	T
12	A7	Address bit 7	T
13	A8	Address bit 8	T
14	A9	Address bit 9	T
15	A10	Address bit 10	T
16	A11	Address bit 11	T
17	A12	Address bit 12	T
18	A13	Address bit 13	T
19	A14	Address bit 14	T
20	A15	Address bit 15	T
21	A16	Address bit 16	T
22	A17	Address bit 17	T
23	A18	Address bit 18	T
24	A19	Address bit 19	T
25	VI0*	Vectored Interrupt 0	O
26	VI1*	Vectored Interrupt 1	O
27	VI2*	Vectored Interrupt 2	O
28	VI3*	Vectored Interrupt 3	O
29	VI4*	Vectored Interrupt 4	O
30	VI5*	Vectored Interrupt 5	O
31	VI6*	Vectored Interrupt 6	O
32	VI7*	Vectored Interrupt 7	O
33	NMI*	Nonmaskable Interrupt	O
34	RESET	System Reset	T
35	HLD*	Processor Hold (DMA)	O
36	HLDA	Hold Acknowledge (DMA)	T

Continued on next page.

HERO 2000 Memory/IDE Expansion

Pin #	Mnemonic	Definition	Termination
37	PHI	5 MHz system clock	T
38	CLK	2 MHz system clock	T
39	MLDIS*	Main Latch Disable (DMA)	O
40	SLEEP	Sleep signal for system	O
41	OUT	Output cycle status	T
42	INP	Input cycle status	T
43	MEMR	Memory Read cycle status	T
44	WO*	Write Output cycle status	T
45	SYNC	Signal retimed from ALE	T
46	WR*	Write strobe	T
47	STVAL*	Status Valid	T
48	DBIN	Data bus In (read strobe)	T
49	RDY	Ready; adds wait states	O
50	MWRT	Memory Write cycle	T
51	DI0	Data Input bit 0	C
52	DI1	Data Input bit 1	C
53	DI2	Data Input bit 2	C
54	DI3	Data Input bit 3	C
55	DI4	Data Input bit 4	C
56	DI5	Data Input bit 5	C
57	DI6	Data Input bit 6	C
58	DI7	Data Input bit 7	C
59	DO0	Data Output bit 0	T
60	DO1	Data Output bit 1	T
61	DO2	Data Output bit 2	T
62	DO3	Data Output bit 3	T
63	DO4	Data Output bit 4	T
64	DO5	Data Output bit 5	T
65	DO6	Data Output bit 6	T
66	DO7	Data Output bit 7	T
67	+5S	Supplies +5 VDC when sleeping	
68	+5S	Supplies +5 VDC when sleeping	
69	+12	Supplies +12 VDC	
70	+12	Supplies +12 VDC	
71	+5	Supplies +5 VDC	
72	+5	Supplies +5 VDC	

INTERRUPT DEFINITIONS

Listed in order of importance. The lower the number the higher the priority.

Level	Description
0	Originally reserved for parity error interrupt when using dynamic RAM. Since the ET-19 uses static RAM this interrupt vector may be redefined in the future.
1	REMOTE LINK: Active when the ET-19-2 Remote is communicating with the Robot.
2	TIMER: Processes the 10ms timer interrupt
3	Reserved for floppy disk controller option
4	EPCIAINT: Serial port A interrupt
5	EPCIBINT: Serial port B interrupt
6	Originally reserved for a video board upgrade. May be redefined in the future.
7	Originally reserved for a printer port upgrade. May be redefined in the future.

HERO 2000 Memory/IDE Expansion

REPLACEMENT PARTS LIST

Integrated Circuits

Quantity	Description	Location
1	74LS00 - Quad NAND gate	U7
1	74LS02 - Quad NOR gate	U10
1	74LS04 – Hex Inverter	U12
1	74LS32 – Quad OR gate	U18
1	74LS138 – 3 to 8 line decoder	U15
3	74ALS573	U13,U16,U19
5	74LS541	U8,U11,U14,U17,U20
1	GAL16V8 or PALCE16V8 Memory decoder (MEM)	U6
1	GAL16V8 or PALCE16V8 IDE I/O decoder (I/O)	U9
1	HM628512 (512Kx8) static memory chip (supplied separately)	U1
4	Any combination of supported RAM, NVRAM, or EPROM	U2-U4

Other components

Quantity	Description	Location
4	14 pin DIP sockets	U7, U10, U12, U18
1	16 pin DIP socket	U15
10	20 pin DIP sockets	U6, U8, U9, U11,U13, U14,U16, U17,U19, U20
4	28 pin DIP sockets (.600 spacing)	U2, U3, U4, U5
1	32 pin DIP socket (.600 spacing)	U1
1	4 position DIP switch	SW1
21	.01uf or .1uf (25v or higher) monolithic axial or dip capacitors	U1-U20, SW1
2	33uf (16v or higher) electrolytic capacitors	C1, C2
1	6-pin 10K SIP resistor pack (common lead)	RP1
1	10-pin 10K SIP resistor pack (common lead)	RP2
3	510 ohm 1/8w resistors	R1-R3
2	GREEN LED T-1 3/4	+5v, +5v Sleep
1	RED LED T-1 3/4	IDE
1	72-pin (2x36) right angle header (Bus connector)	P1
1	40-pin (2x20) right angle connector (IDE connector)	P2
4	4-pin (1x4) header (Gold plated)	
9	3-pin (1x3) header (Gold plated)	
5	RED (.100") jumpers	Voltage select
12	BLACK (.100") jumpers	Chip select
1	Custom Printed Circuit Board (through hole plated) REV1.5	

HERO 2000 Memory/IDE Expansion

Revision History

Breadboard - Initial prototype of 576K memory on ET-19-14 experimenter card. Worked fine after finding mistake in Heathkit documentation about Sleep signal. Signal is inverted on Experimenter card. The fix was to bypass on board inverter and go directly to bus. No IDE controller section.

Version 1.0 - First version on actual PCB (Not functional)

Version 1.1 - Added jumper wires for missing power trace on U20. Moved trace for DOUT signal. Memory section worked as designed.

Version 1.2 - Made adapter to swap leads in 74LS138 decoder at U15. Added jumper wires for missing address lines a0-a3 to I/O decode logic. Updated logic in I/O PLD. IDE would function only with an old Maxtor drive. Changed latching of I/O signal to correct timing issue by changing traces. Updated logic in I/O PLD. IDE would then function properly with all drives tested.

Version 1.3 - Updated PCB layout with corrected traces and several cosmetic changes.

Version 1.4 - Moved supply for IDE activity LED from sleep +5v supply to regular 5v supply. Added spare chip location and pads for future IRQ support.

Version 1.5 - Moved supply for Memory decoder PLD from +5v supply to sleep 5v supply to prevent memory in U1 from being selected during sleep cycle. Connected SW2 (previously unused) to CS line on IDE connector. This allows Cable Select to be turned on and off.

SPECIAL THANKS

A note of thanks to the following people who helped with the project:

Scott Christensen (creator of XTIDE which provided inspiration for the project)

Bob Wind (Driver support/testing, author of File/Boot programs, and File/Boot program operation)

All the users who have responded with their feedback!