HUNT ENGINEERING
Chestnut Court, Burton Row,
Brent Knoll, Somerset, TA9 4BP, UK
Tel: (+44) (0)1278 760188,
Fax: (+44) (0)1278 760199,
Email: sales@hunteng.demon.co.uk
URL: http://www.hunteng.co.uk

We are a
committed
member of the
Texas Instruments
3rd party
programme

For Sales and Support in North America Please Contact Our Strategic Partner:

Traquair Data Systems Inc, 114 Sheldon Road, Ithaca, NY 14850 USA

Tel 607 266 6000, Fax 607 266 8221

Email Traquair@traquair.com, URL http://www.traquair.com

For Sales and Support in Other Areas Please Contact Your Local Reseller.

HUNT ENGINEERING

Writes API Example

Description and Reference

With Borland C++ Builder

Document Rev A
API Writes Example Rev 1.0
J.Thie 05-03-01

COPYRIGHT

WARRANTIES LIABILITY and INDEMNITIES

HUNT ENGINEERING warrants the hardware to be free from defects in the material and workmanship for 12 months from the date of purchase. Product returned under the terms of the warranty must be returned carriage paid to the main offices of HUNT ENGINEERING situated at BRENT KNOLL Somerset UK, the product will be repaired or replaced at the discretion of HUNT ENGINEERING.

> Exclusions - If HUNT ENGINEERING decides that there is any evidence of electrical or mechanical abuse to the hardware, then the customer shall have no recourse to HUNT ENGINEERING or its agents. In such circumstances HUNT ENGINEERING may at its discretion offer to repair the hardware and charge for that repair.
>
> Limitations of Liability - HUNT ENGINEERING makes no warranty as to the fitness of the product for any particular purpose. In no event shall HUNT ENGINEERING'S liability related to the product exceed the purchase fee actually paid by you for the product. Neither HUNT ENGINEERING nor its suppliers shall in any event be liable for any indirect, consequential or financial damages caused by the delivery, use or performance of this product.

Because some states do not allow the exclusion or limitation of incidental or consequential damages or limitation on how long an implied warranty lasts, the above limitations may not apply to you.

TECHNICAL SUPPORT

Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section www.hunteng.co.uk/support/support.htm on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to www.hunteng.co.uk for the list of current re-sellers.

HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.demon.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.

The writes example

The writes example is an example program that tests the PCI FIFO and HSB interface of a HERON carrier board. The example will try to boot a small and simple program onto the first processor (on the module in slot 1). This program will absorb a stream of data from the host, sent via the PCI interface. The host example program will send the stream of data and tell you if everything worked or not. It will also give a very rough estimate of the transfer speed.

Compiling/Linking the Example

To compile/link the example, please create a new project with your Borland C++ Builder compiler (make it a 'console cpplication'). After you created a new project, you need to set the path to the Hunt Engineering API include file ("heapi.h") and library ("hendrv.lib"). There is an environment variable "HEAPI_DIR" that points to the directory where you installed the Hunt Engineering API.

Include directory:	$(HEAPI_DIR)
Add library file:	$(HEAPI_DIR)\hendrv.lib

How to create and start a new project (Borland C++ Builder)

In Borland C++ Builder, create a new workspace

1. Make a directory on the hard disk where you want to keep the new project.
2. File → New Application.
3. View → Project Manager.
4. Remove 'Unit1.cpp' from the project. (Select 'Unit1.cpp' in the 'Project Manager' window. Click the button marked with a folder and a minus sign. Answer 'no' when asked if you would like to save changes to 'Unit1.cpp').
5. Close the Project manager window by clicking on the 'x' in the top right hand corner.
6. File → Save Project As. Navigate to the directory where you want to keep this project. Next, enter a name for the project, and click the 'save' button. (Note that with C++ Builder you cannot give the project the same name as the name of the main CPP file that you want to include.)

Add files and libraries to the project

7. View → Project Manager. Add 'main.c' and 'cload.c' located in the 'writes' example directory: click on the button that shows a folder and a plus sign. Change 'Files of Type' to 'C file (*.c)'. Browse to the 'writes' example directory. Select 'main.c' and 'cload.c' (if you keep the CTRL key pressed, you can select more than 1 file at once). Click 'Open'.
8. Go back to the 'Project Manager' window. Click on add again and change 'Files of type' to 'Library file (*.lib)'.
9. Navigate to the directory that contains 'hebdrv.lib' (usually 'c:\heapi') and include it.
10. Close the Project Manager window.

Include files

11. Options → Project.

12. Select ‘Directories/Conditionals’ tab from the window that pops up (‘Project Options’).
13. Add to the end of the line of text in the box marked ‘Include Path’ the location of the include files from your API installation directory (usually ‘c:\heapi’).
14. Add to the end of the line of text in the box marked ‘Library Path’ the location of the library files from your API installation directory (usually ‘c:\heapi’).
15. Click ‘OK’.

Linker

16. Options → Project.
17. Select the Linker tab from the window that pops up (‘Project Options’).
18. Change the Application Type to Console application and click ‘OK’.

Compile and Link

19. File → Save All (save all the changes you have made to the new project).
20. Select Build All from the Project Menu.

Running the example

Open a DOS box and browse to the writes example directory. Change directory to your project’s Debug directory (or the Release directory, if you built a release version). Assuming that your executable is called ‘writes.exe’, and you use a HEPC8 carrier board, type:

```
writes hep8a 0 a 1000 1000 10000
```

The output should be something like:

```
Start at 1000, inc 1000, end at 10000, BlockSize=250 on hep9a (0: Compo
Resetting...
Serial bus: slot 1: HERON1-C6201, rom version 4.
Booting M:\TMP\HOST_A~1\C6X\EXAMPLES\WRITES\writes.out...
Testing...
Writes Transfer size 1000 DWORDS in 99 ticks, Speed: 39.46 KBytes/sec
Writes Transfer size 2000 DWORDS in 1 ticks, Speed: 7812.50 KBytes/sec
Writes Transfer size 3000 DWORDS in 3 ticks, Speed: 3906.25 KBytes/sec
Writes Transfer size 4000 DWORDS in 1 ticks, Speed: 15625.00 KBytes/sec
Writes Transfer size 5000 DWORDS in 7 ticks, Speed: 2790.18 KBytes/sec
Writes Transfer size 6000 DWORDS in 2 ticks, Speed: 11718.75 KBytes/sec
Writes Transfer size 7000 DWORDS in 5 ticks, Speed: 5468.75 KBytes/sec
Writes Transfer size 8000 DWORDS in 5 ticks, Speed: 6250.00 KBytes/sec
Writes Transfer size 9000 DWORDS in 8 ticks, Speed: 4394.53 KBytes/sec
Writes Transfer size 10000 DWORDS in 2 ticks, Speed: 19531.25 KBytes/se
Check whether any interrupts were used: read 0, write 1, master mode 0.
```

Out file

The writes example uses a DSP executable to load onto the HERON module’s processor. The executable is ‘writes.out’ for a HERON1 module and ‘writes4.out’ for a HERON4 module. The writes example uses ‘argv[0]’ and to try to find the out file. First it looks in the full path; if the out file isn’t there, it looks one directory up (or down, depending on your viewpoint). It will walk all the way down the ‘argv[0]’ path until it finds the out file. As long as your Visual C/C++ project was created in the writes example directory, the out file should be found.

The code for the out files is in the writes example's 'dsp' sub-directory. The out file code is very simple and makes no use of DSP/BIOS or Code Composer Studio features. The code in the 'dsp' directory should not be used as an example to build your DSP projects on. It is used solely for the purpose of this example. This example focuses on the HUNT ENGINEERING API and how to use it. For examples that use DSP/BIOS projects, please have a look at Server/Loader examples or HERON-API examples.

Technical Support

1. Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section www.hunteng.co.uk/support/support.htm on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to www.hunteng.co.uk for the list of current re-sellers.
2. HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.demon.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.