



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

Reads API Example

Description and Reference

With Microsoft Visual C/C++ 4.0

Document Rev A
API Reads Example Rev 1.0
J.Thie 02-03-01

COPYRIGHT

This documentation and the product it is supplied with are Copyright HUNT ENGINEERING 1999. All rights reserved. HUNT ENGINEERING maintains a policy of continual product development and hence reserves the right to change product specification without prior warning.

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 READS EXAMPLE	4
COMPILING, LINKING AND RUNNING THE EXAMPLE	5
COMPILING/LINKING THE EXAMPLE.....	5
HOW TO CREATE AND START A NEW PROJECT (VISUAL C/C++ 4)	5
<i>In Microsoft Visual C/C++, create a new workspace.....</i>	<i>5</i>
<i>Add files and libraries to the project.....</i>	<i>5</i>
<i>Include files.....</i>	<i>6</i>
<i>Linker.....</i>	<i>6</i>
<i>Compile and Link.....</i>	<i>6</i>
RUNNING THE EXAMPLE.....	6
OUT FILE.....	7
TECHNICAL SUPPORT	8

The reads 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 send a stream of known data to the host, via the PCI interface. The host example program will read the stream of data, verify the data is as expected, and tell you if everything worked or not. It will also give a very rough estimate of the transfer speed.

Compiling, Linking and Running the example

Compiling/Linking the Example

To compile/link the example, please create a new project with your Microsoft C/C++ 4.0 compiler ('Win32 Console Application'). 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 (Visual C/C++ 4)

In Microsoft Visual C/C++, create a new workspace

1. File → New.
2. Select "Project Workspace" and click "OK".
3. A new window has appeared. Select "Console Application".
4. Click "Browse" and change directory to the reads example directory. Click "OK" in this "Choose Directory" window, we return to "New Project Workspace".
5. Type a name for the workspace at "Name". The workspace name will also define the name of the executable. We used "reads" as the workspace name, and we will thus work with executable file "reads.exe".
6. Click on "Create".

Add files and libraries to the project

7. Insert → Files into Project.
8. In "List Files of Type" select "Source Files (*.c, *.cpp, *.cxx)", and add "main.c" and "cload.c" to your project. Depending on in what directory the project was created, you may have to browse to a different directory. If you keep the CTRL key pressed, then you can select more than 1 file at the same time.
9. Click "OK".
10. Insert → Files into Project.
11. In "List Files of Type" select the libraries that must be added to the project workspace, e.g. Library Files (*.lib) in the case of the examples.
12. Change the window directory to "c:\heapi" (or the directory where you copied the API libraries into).
13. Select "hendrv.lib".
14. Click OK

Include files

15. Build → Settings.
16. Select the “C/C++” tab.
17. Select “Preprocessor” in “Category”
18. At “Additional include directories” type: “c:\heapi” (or the installation directory where you installed the HUNT ENGINEERING API into).
19. Click “OK”

Linker

20. The “hendrv.dll” library has been linked against the multi-threaded version of the C RTL. This is because it uses threads in some cases to handle asynchronous I/O. Any application using “hendrv.dll” (i.e. all Win32 API applications!) should also be linked against the multithreaded RTL (either debug or release versions). These can be selected as follows:
21. Build → Settings
22. Select the “C/C++” window
23. From the “Category” pull down menu select “Code Generation”
24. In the “Use run-time Library” pull down menu select a multi-threaded library
25. Click “OK”

Compile and Link

26. Do a Build → Rebuild All (or a Build → Build reads.exe).

Running the example

Open a DOS box and browse to the reads 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 ‘reads.exe’, and you use a HEPC8 carrier board, type:

```
reads 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\READS\reads.out...
Testing...
Reads Transfer size 1000 DWORDS in 100 ticks, Speed: 39.06 KBytes/sec
Reads Transfer size 2000 DWORDS in 1 ticks, Speed: 7812.50 KBytes/sec
Reads Transfer size 3000 DWORDS in 1 ticks, Speed: 11718.75 KBytes/sec
Reads Transfer size 4000 DWORDS in 5 ticks, Speed: 3125.00 KBytes/sec
Reads Transfer size 5000 DWORDS in 1 ticks, Speed: 19531.25 KBytes/sec
Reads Transfer size 6000 DWORDS in 6 ticks, Speed: 3906.25 KBytes/sec
Reads Transfer size 7000 DWORDS in 4 ticks, Speed: 6835.94 KBytes/sec
Reads Transfer size 8000 DWORDS in 5 ticks, Speed: 6250.00 KBytes/sec
Reads Transfer size 9000 DWORDS in 4 ticks, Speed: 8789.06 KBytes/sec
Reads Transfer size 10000 DWORDS in 5 ticks, Speed: 7812.50 KBytes/sec
Check whether any interrupts were used: read 1, write 0, master mode 0.
```

Out file

The reads example uses a DSP executable to load onto the HERON module's processor. The executable is 'reads.out' for a HERON1 module and 'reads4.out' for a HERON4 module. The reads 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 reads example directory, the out file should be found.

The code for the out files is in the reads 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.

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.