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

Mysl Server/Loader Example

For RTOS-32

Document Rev A
Server/Loader Mysl Example Rev 1.0
J.Thie 26-02-01

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.

TABLE OF CONTENTS

The mysl example shows how to use the Server/Loader library. Rather than employ the standard exe or RTB Server/Loader, you can simply link with the Server/Loader library and load a network of processors via a library call. The library will still use a network file and it will still want to read *.out files that are defined in the network file.

The Server/Loader library is written in C++. Your project will also have to be a C++ project if you want to use the library. But you can use the mysl.cpp file to reset and boot a network of processors; then proceed calling your C (/ C++) functions located in other files.

(This example will **not** work with TIM-40 carrier boards such as the HEPC2E, HEPC3, HEPC4 or HECPCI1. It will also **not** work with the HEPC6, a one ‘C6x processor board.)

Compiling/Linking the Example

To compile and link the example, please use the 'makefile' that is present in this directory. This makefile is set-up to use a Microsoft 32-bit C/C++ compiler. You can only execute the makefile in a DOS-box prepared by On Time for any of the Microsoft C/C++ command line examples ('Visual C++ (Command Line) Demos'). Or, if you execute from a standard DOS-box, please execute the standard RTOS-32 'varsvc.bat' first.

To execute the 'makefile':

```
nmake
```

Running the example

To run the example, prepare a floppy disk and insert it into the 'a:' drive. Then type:

```
bootdisk mysl a:
```

Next, copy the *.out files to the floppy disk as well:

```
copy network a:
copy ..\stdio.c a:
copy stdio1.out a:            (if you have a HERON1 module in slot 1)
copy stdio4.out a:            (if you have a HERON4 module in slot 1)
```

(To help you start up faster, we have included two prepared out files, stdio1.out for a HERON1, stdio4.out for a HERON4. Please change the 'network' file to suit the module type you have in slot 1. But usually the *.out file must first be created using Code Composer Studio. Please refer to the document in the lower (upper?) directory how to do this.) After completion, remove the floppy disk and insert it into the target machine's floppy disk. Reboot. The target machine should boot from disk. You should see something like:

```
…
fread : 2048 / 2048  feof: 0
fread :    1 / 4096  feof: 16

WRITE 64Kbytes BLOCK
Written 64 kBytes

SYSTEM test

Trying "dir "
float  (1.2e-4): 0.000120, 0.00012
double (2.4e-8): 0.000000, 2.4e-008
Leaving server mode
(Press key to continue)
```

Probably the output goes too fast for you to see it all scroll by. The program may not be able to find 'stdio.out'. In that case **either** (1) change 'stdio.out' to 'a:\stdio.out' in the network file on the floppy disk, **or** (2) copy 'stdio.out' to 'c:\' of your target machine. Similarly, the (dsp) program may not be able to find 'stdio.c'. In that case, **either** (1) change 'stdio.c' the line that opens 'stdio.c' to 'a:\stdio.c', **or** (2) copy stdio.c to 'c:\' of your target machine.

The Makefile

What changes have been made to the original RTOS-32 example makefile? This section will explain what needs to be changed (or added) in a makefile to compile/link successfully the Hunt Engineering API programs

Include file

All Hunt Engineering API programs must include 'heapi.h'. This file is located in the Hunt Engineering API installation directory. The installation program will have created an environment variable 'HEAPI_DIR' that points to the installation directory. To have the makefile understand where 'heapi.h' lives, the following line must be in your makefile:

```
INCLUDE = $(RTTARGET)\include;$(HEAPI_DIR);$(INCLUDE)
```

The bold italic part is the part added by us.

Libraries

The Hunt Engineering Server/Loader library is delivered as a static library ('rtossl.lib'). The Hunt Engineering API is delivered as a static library ('rtosdrv.lib'). They must be linked with RTTARGET-32, RTFILES-32 and RTKERNEL-32. In the lines following your '.exe' declaration 'rtossl.lib' must be linked in first, then 'rtosdrv.lib', before all of the RTOS-32 libraries:

```
mysl.exe: Init.c ..\mysl.cpp
   cl /MT /W3 /GX /Fm /Zi -D_RTOS32=1 -DCMDLINE=1 -DPC=1 -omysl.exe \
      ..\mysl.cpp \
      $(HESL_DIR)\lib\rtos32\rtossl.lib \
      $(HEAPI_DIR)\rtosdrv.lib \
      rtk32.lib \
      drvrt32.lib \
      rtfiles.lib \
      rtfsrtt.lib \
      rtt32.lib \
      rttheap.lib \
      $(LNKOPT)
```

The bold italic part is the part added by us.

The necessary RTFILES-32 libraries are 'rtfiles.lib' and 'rtfsk32.lib'. Note that the latter is the RTKERNEL-32 version of the RTFILES-32 library.

The necessary RTKERNEL-32 libraries are 'rtk32.lib' (debug version) and 'drvrt32.lib', as the Hunt Engineering API uses multi-threading. (The HeRead and HeWrite will spawn separate threads to do the actual reading and writing. HeTestIo and HeWaitForIo 'test' the thread to see whether it has completed a transfer.)

The RTTARGET-32 library is 'rtt32.lib'. Library 'rttheap.lib' is optional. Please refer to the RTOS-32 manual (for example, ch.7 page 106 and 107).

Compile Parameters

The Hunt Engineering API supports several different types of Operating Systems. To select RTOS-32 support, you need to #define a variable '_RTOS32'. The easiest way to do this is in the makefile. Also, as the Hunt Engineering API is multi-threaded, you need to use the '/MT' option of the Microsoft C/C++ compiler. To select the proper parts from the Hunt Enginering Server/Loader library, you must #define CMDLINE and PC. Example:

```
mysl.exe: Init.c ..\mysl.cpp
   cl /MT /W3 /GX /Fm /Zi -D_RTOS32=1 -DCMDLINE=1 -DPC=1 -omysl.exe \
      ..\mysl.cpp \
      $(HESL_DIR)\lib\rtos32\rtossl.lib \
      $(HEAPI_DIR)\rtosdrv.lib \
      rtk32.lib \
      drvrt32.lib \
      rtfiles.lib \
      rtfsrtt.lib \
      rtt32.lib \
      rttheap.lib \
      $(LNKOPT)
```

The bold italic part is the part added by us.

Initialisation

A file 'init.c' is included in the project. This is a 'standard' file from On Time, which they use for projects that use RTFILES-32. I have simply copied it into this example; it's needed when you use file support. On Time's comment in 'init.c':

```
/* Some standard initializations for RTFiles-32 programs.

  This file is linked with most RTFiles-32 demo programs. It provides a
  convenient place to configure RTTarget-32 and RTFiles-32.
*/
```

The mysl.cfg configuration file

What changes have been made to the original RTOS-32 example configuration file? This section will explain what needs to be changed (or added) in a configuration file to compile and link successfully the Hunt Engineering Server/Loader programs

mysl.cfg: commandline

RTOS-32 programs have the possibility to carry a command line. This is done by specifying a command line in one of the configuration files (we just chose 'mysl.cfg'). The 'mysl' example needs to use a command line, as you need to specify options and the name of the network file. The command line is further just as you would expect with a normal DOS or win32 program. The following line must be added to a configuration file, for the 'mysl' example to work properly:

```
CommandLine "a:\mysl.exe -rlsv a:\network"
```

The directory ('a:\') is significant.

mysl.cfg: floppy access

To access files on a floppy disk, not only do you need to link with RTFILES-32 libraries, you also need to allocate a DMA buffer for the floppy driver in your configuration file. We added the following line to the 'reads.cfg' configuration file:

```
Locate Nothing FloppyDMA HighMem 18k 32k ReadWrite
```

Please refer to the RTOS-32 manual (Part III, ch. 7, page 300) for more information.

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.