

RTDM-native

Wolfgang Grandegger

< wg@grandegger.com >

Revision History	
Revision 0.1	2007-04-10
First release	

Table of Contents

1. Introduction
2. Installation
 - 2.1. Downloading RTDM-native
 - 2.2. Preparing the target kernel
 - 2.3. Configuring and building the target kernel
 - 2.4. Building user-space support
3. Availability and test cases
4. Todo list
5. References

1. Introduction

This is a first version of RTDM-native, a port of RTDM [1, 2] over native Linux with the real-time preemption and high-resolution timer patch [3, 4]. RTDM, the Real-Time Driver Model, is an approach to unify the interfaces for developing device drivers and associated applications under real-time Linux. It is currently available for the dual-kernel hard real-time Linux extensions Xenomai [5] and RTAI [6]. This port permits to use RTDM drivers like RTserial, RT-Socket-CAN and soon RTnet under Linux-rt as well. RTDM-native is on the road map [7] of Xenomai 3 aiming at support for native preemption.

2. Installation

2.1. Downloading RTDM-native

RTDM-native can be downloaded from Xenomai's SVN repository [8]:

```
$ svn co http://svn.gna.org/svn/xenomai/branches/rtdm-native rtdm-native
$ export RTDM_NATIVE_ROOT=$PWD/rtdm-native
```

2.2. Preparing the target kernel

RTDM-native uses Xenomai's procedure of preparing the kernel tree [9]. The shell script **scripts/prepare-kernel.sh** sets up the target kernel properly. The syntax is as follows:

```
$ cd $RTDM_NATIVE_ROOT
$ scripts/prepare-kernel.sh --rtm-native --linux=<linux-srctree>
```

The option **--rtm-native** tells the script that the kernel should be prepared for RTDM-native. This requires a vanilla kernel with the real-time preemption and high-resolution timer patch (**CONFIG_PREEMPT_RT=y** and **CONFIG_HIGH_RES_TIMERS=y**). The option **--linux=** specifies the path of the target kernel source tree. Such kernel tree being configured or not makes no difference and is valid either way. For instance, the following command would prepare the Linux tree located at **/usr/src/linux-2.6.20-rt8** in order to include the RTDM-native support:

```
$ cd RTDM_NATIVE_ROOT
$ scripts/prepare-kernel.sh --rtm-native --linux=/usr/src/linux-2.6.20-rt8
```

Note: The script will infer the location of the RTDM-native kernel code from its own location within the Xenomai source tree. In other words, if **/usr/src/rtdm-native/script/prepare-kernel.sh** is executing, then RTDM-native's kernel support available at **/usr/src/rtdm-native/ksrc** will be bound to the target kernel.

2.3. Configuring and building the target kernel

Once the target kernel has been prepared, all RTDM-native configuration options are available from the "Real-time subsystem" toplevel menu. There are several configure options that cause large latencies; they should be avoided. Check Xenomai's TROUBLESHOOTING file [10] and the Real-Time Linux Wiki [3] for further information and help. Once configured, the kernel should be built as usual. If you want several different configs/builds at hand, you can reuse the same source by adding **O=../build-<target>** to each make invocation. See also section 2.2 of [9] for an example.

2.4. Building user-space support

The user space example and utility programs are currently available in "examples/rtdm-native". To build them, just go this directory and type make:

```
$ cd RTDM_NATIVE_ROOT/examples/rtdm-native
$ make
```

3. Availability and test cases

This port has preliminary and experimental support for RTDM task services. The implementation is not straight-forward, especially because of tricky parts like periodic mode, task removal and releasing locked resources. Nevertheless, the basic functionality is available and it should already be sufficient to get RTnet working. The RTDM-native port has been tested with the following RTDM named and protocol drivers:

- RTserial
- RT-Socket-CAN
- irqbench (without kernel tasks)

Note: for RTnet [11] further effort is required, mainly its build system must be adopted accordingly.

4. Todo list

- Finalizing and testing RTDM task support.
- Implementing PROC file-system support.
- Fixing a few issues marked with "FIXME" in the source code.
- Get RTnet built and working with RTDM-native.

5. References

- [1] <http://www.xenomai.org/documentation/branches/v2.3.x/pdf/RTDM-and-Applications.pdf>
- [2] http://www.xenomai.org/documentation/trunk/html/api/group__rtm.html
- [3] http://rt.wiki.kernel.org/index.php/Main_Page
- [4] <http://www.kernel.org/pub/linux/kernel/projects/rt/>
- [5] <http://www.xenomai.org>
- [6] <http://www.rtai.org>
- [7] <http://www.xenomai.org/index.php/Xenomai:Roadmap>
- [8] <https://gna.org/svn/?group=xenomai>
- [9] <http://www.rts.uni-hannover.de/xenomai/lxr/source/README.INSTALL?v=SVN-trunk>
- [10] <http://www.rts.uni-hannover.de/xenomai/lxr/source/TROUBLESHOOTING?v=SVN-trunk>
- [11] <http://www.rtnet.org>