

Demonstrate easy network stack switching (exHelloNet)

NDK 5.7/MPTK 2.4

Application Note, 12 December 2007

Overview

The principle to link in the correct libraries and keep the code itself portable is simple.

In the `REQUIRES` variable, you only put the `tmNetConfig` and `tmNetSocket` interfaces (next to the other things needed for your application). So `TargetTcp` is NOT present in the `REQUIRES` variable. `exHelloNet` does not use the `tmNetInet` interface as it is a simple application.

In the `LIBS` variable, you put the actual implementations of the interfaces that you want to use. So, when you want `TargetTcp` to implement the `tmNetSocket` interface, you add `TargetTcp` to the `LIBS` variable. If you want the `PNX1500` ethernet driver, you add `tmhwEth`, `tmbslPhyDp83847`, `tmDlEth` to the `LIBS` variable. If you want the combination of `TargetTcp` and the `PNX1500` driver to work together, you add `tmEthTtcp` and `tmNetConfigIp3902Ttcp` (the latter one implements the `tmNetConfig` interface) to the `LIBS` variable. When you want to use `winsoc` via `RPC`, it is somewhat simpler as there is 1 component that combines this all. Just add `tmRpcSocket` to the `LIBS` variable.

So for a user who is only interested in one stack/driver combination, the case is simple. Just add the appropriate libraries in the `LIBS` variable of the application `makefile`. If he later wants to migrate to another stack/driver, he just needs to update the `LIBS` variable in his `makefile` and link again.

The case is a bit more complicated when you want build-time switching. `SDE2` provides you with a few possibilities:

1. using a diversity
2. using a user-defined variable
3. changing the `makefile` manually
4. using `gmake -f<makefile_name>` with multiple `makefiles`

We have chosen to use a user-defined variable, which should follow the naming `gui del ine_SDEUD_<ApplicationName>_<Meaningful Name>` as defined in the `SDE2` documentation. As a workaround for customers using newer versions of `cygwin`, an all-uppercase variable (`MPTK_NET_SUBSYSTEM`) is also available as a workaround. This will be used if no variable of the `_SDEUD` type is set for the application.

A diversity is intended for components, not applications. It is intended to be a global diversity, with (possible) impact on multiple components. This is overkill for what we want to achieve here.

Manually changing the `makefile` is troublesome if you are using a version control system.

Using `gmake -f` would work, but you need to duplicate the generic part in each `makefile` and it means you cannot use `build_exe`, which makes it difficult to build all required components needed for the application.

So, we use a user-defined variable. For the `exHelloNet` example application, this is `_SDEUD_exHelloNet_NetSubSystem`. You can then build `exHelloNet` for `TargetTcp` and the `PNX1500` ethernet driver using

```
gmake _SDEUD_exHelloNet_NetSubSystem=EthTtcp
```

or

```
set _SDEUD_exHelloNet_NetSubSystem=EthTtcp  
gmake
```

The second form allows you to use `build_exe` instead of `gmake`.

If mixed-case variables do not work in your environment, you can use the following workaround
Different build flavors exist:

```
gmake MPTK_NET_SUBSYSTEM=EthTtcp
```

or

```
set MPTK_NET_SUBSYSTEM =EthTtcp  
gmake
```

For convenience, the `makefile` will default to `TargetTcp` and the `PNX1500` ethernet driver if both the `_SDEUD_exHelloNet_NetSubSystem` variable and the `MPTK_NET_SUBSYSTEM` variable are not set. A warning message will be displayed at the end of the build to alert the user to this fact.

If the `_SDEUD_exHelloNet_NetSubSystem` variable is set to an invalid value, the application build will not be attempted and an error message will be displayed, indicating the valid values for this variable. The same applies when the `_SDEUD_exHelloNet_NetSubSystem` variable is not set and the `MPTK_NET_SUBSYSTEM` variable is set to an invalid value.

Example of switching makefile

```

## Makefile for exHelloNet with switching
DIR_LOCAL = apps/exHelloNet

#*****
# Do not change the following include
#*****
include $(TMROOT)/sde/environment.mk

_SDEUD_exHelloNet_WarningMessage=
_SDEUD_exHelloNet_ErrorMessage=

# Default setting is EthTtcp
ifeq ($(_SDEUD_exHelloNet_NetSubSystem),)
ifeq $(MPTK_NET_SUBSYSTEM),)
_SDEUD_exHelloNet_NetSubSystem=EthTtcp
_SDEUD_exHelloNet_WarningMessage="WARNING: _SDEUD_exHelloNet_NetSubSystem and
MPTK_NET_SUBSYSTEM variables are not set. Using EthTtcp as default. Set to Rpc for RPC
sockets."
else
_SDEUD_exHelloNet_NetSubSystem=$(MPTK_NET_SUBSYSTEM)
_SDEUD_exHelloNet_WarningMessage="WARNING: _SDEUD_exHelloNet_NetSubSystem variable is
not set. Using MPTK_NET_SUBSYSTEM (value $(MPTK_NET_SUBSYSTEM))."
endif
endif

ifeq ($(_SDEUD_exHelloNet_NetSubSystem),EthTtcp)
LIBS_FOR_NETSUBSYSTEM = tmNetConfigIp3902Ttcp TargetTcp tmEthTtcp tmbSlPhyDp83847
tmhwEth tmdl Eth
else
ifeq ($(_SDEUD_exHelloNet_NetSubSystem),Rpc)
LIBS_FOR_NETSUBSYSTEM =tmRpcSocket
else
_SDEUD_exHelloNet_ErrorMessage="ERROR: _SDEUD_exHelloNet_NetSubSystem or
MPTK_NET_SUBSYSTEM variable is not set to EthTtcp or Rpc."
endif
endif

#-----
# Source environment variables
#-----
CXX_SOURCES =

C_SOURCES = \
src/exHelloNet.c \
src/exHelloNet_SockTest.c \

```


