

# uClinux Driver Configuration Application Note V1.0

***Publication Release Date: Sep. 2008***

---

**Support Chips:**

NUC740A  
NUC710A  
NUC745A

**Support Platforms:**

uClinux-2.4.x  
uClinux-2.6.x

---



The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.



# Table of Contents

- 1. Introduction .....4
- 2. uClinux-2.4.x.....5
- 3. uClinux-2.6.x.....7
- 4. Revision History .....8

# 1. Introduction

Nuvoton's uClinux BSP provides drivers for most of interfaces on EV boards. Usually these drivers are sufficient for users to develop their products. However in some cases users need to implement their own driver(s). This application note only describes the necessary steps to add a new driver into uClinux 2.4.x and uClinux 2.6.x kernels. As to driver implementation, it is beyond the scope of this application note.

## 2. uClinux-2.4.x

To let compilation process add a specific file into kernel image or build it as a kernel module, users must edit the Makefile(s) in kernel. Below is a Makefile example. Each line is followed by an explanation.

```
#
# Makefile example
#
.....
obj-y += driver_1.o
# Build driver_1.c into kernel

obj-m += driver_2.o
# Build driver_2.c as a loadable module

obj-$(CONFIG_DRIVER_3) += driver_3.o
# Build driver_3.c depends on the menuconfig result

subdir-$(CONFIG_BUILD_4) += driver_4
# Parse Makefile in sub directory driver_4/ depends on the menuconfig result

export-objs := driver_1.o
# Mark driver_1.c contains EXPORT_SYMBOL

O_TARGET := drivers.o
# Merge the object file name in this directory to drivers.o

include $(TOPDIR)/Rules.make
# Include top directory rule
```

The simplest method to add the driver into kernel is to declare it as `obj-y` in Makefile, thus the compilation process will always link it into kernel. If load the driver as a module is preferable, then declare it as `obj-m`. If the driver contains export symbol(s), it must be added into `export-objs`, otherwise loadable modules won't be able to find the export symbol(s).

To make the driver configurable via menuconfig, users need to modify Config.in in the same directory with Makefile. Below is a simple Config.in example, each line follows by an explanation.

```
#
# Configuration file example
#

dep_tristate 'Config driver_5' CONFIG_DRIVER_5 $CONFIG_DRIVER_5_DEP
# Configure driver_5 if $CONFIG_DRIVER_5_DEP is true. The options for
# driver_5 could be y(es), m(odule), and n(o).

tristate 'Config driver_6' CONFIG_DRIVER_6
# Configure driver_6, the options are y(es), m(odule), and n(o)

bool 'Config driver_7' CONFIG_DRIVER_7
# Configure driver_7, The options are y(es) and n(o)
```

Kernel configuration system will store all CONFIG\_XXXX configured by user and store it into uClinux-dist/linux-2.4.x/.config. Make tool will then refer this file and Makefiles to decide which driver to link into kernel, which to build as module.

This chapter gives a brief introduction of Makefile and Config.in of 2.4.x kernel. It should be sufficient for most cases. Users could refer to uClinux-dist/Documentation/kbuild for more information about Makefile and Config.in.

### 3. uClinux-2.6.x

The Makefile for 2.6.x kernel is pretty much the same with 2.4.x kernel. Users could refer to previous chapter for the file format. But please note it is not necessary to declare file that exports symbol(s) in 2.6.x kernel anymore. So please don't use O\_TARGET in Makefile.

The configuration script in 2.4.x kernel stores in Config.in files. But in 2.6.x, the scrip stores in Kconfig files. And the script used is here is different with 2.4.x. Below is a simple example which implement the same function with the example for 2.4.x.

```
#
# Configuration file example
#

config DRIVER_5
tristate 'Config driver_5'
depends on DRIVER_5_DEP
# Configure driver_5 if $CONFIG_DRIVER_5_DEP is true. The options for
# driver_4 could be y(es), m(odule), and n(o).

config DRIVER_6
tristate 'Config driver_6'
# Configure driver_6, the options are y(es), m(odule), and n(o)

config DRIVER_7
bool 'Config driver_7'
# Configure driver_7, The options are y(es) and n(o)
```

This sample shows some syntax differences between 2.4.x and 2.6.x. The differences and are listed below.

1. Although Makefiles still use obj-\$(CONFIG\_XXXX) to determine the build behavior, but in Kconfig. CONFIG is removed from script. The CONFIG will be added into kernel configure file by kbuild automatically.
2. The option name is moved to the front followed by keyword "config"
3. Add a new keyword "depends on" is used for declare dependency relationship.

This chapter gives a brief introduction of Makefile and Kconfig of 2.6.x kernel. It should be sufficient for most cases. Users could refer to uClinux-dist/Documentation/kbuild for more information about the syntax of Makefile and Kconfig.

## 4. Revision History

Version	Date	Description
V1.0	Sept. 2008	• Created

### Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur. Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

---

*Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*

---