```
# File: Makefile
#
# Copyright (C) 2008-2010, Marvell International Ltd.
#
# This software file (the "File") is distributed by Marvell International
# Ltd. under the terms of the GNU General Public License Version 2, June 1991
# (the "License").  You may use, redistribute and/or modify this File in
# accordance with the terms and conditions of the License, a copy of which
# is available by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA or on the
# worldwide web at http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt.
#
# A copy of the GPL is available in file gpl-2.0.txt accompanying in this
# deliverables.
#
# THE FILE IS DISTRIBUTED AS-IS, WITHOUT WARRANTY OF ANY KIND, AND THE
# IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
# ARE EXPRESSLY DISCLAIMED.  The License provides additional details about
# this warranty disclaimer.

CC=             $(CROSS_COMPILE)gcc
LD=             $(CROSS_COMPILE)ld
BACKUP=         /root/backup
YMD=            `date +%Y%m%d%H%M`

################################################################################
# Configuration Options
################################################################################

# Debug Option
# DEBUG LEVEL n/1/2:
# n: NO DEBUG
# 1: Only PRINTM(MMSG,...), PRINTM(MFATAL,...), ...
# 2: All PRINTM()
CONFIG_DEBUG=1

# Proc debug file
CONFIG_PROC_DEBUG=y

# ENABLE STA mode support
CONFIG_STA_SUPPORT=y

# Enable AP mode support
CONFIG_UAP_SUPPORT=y

# Re-association in driver
CONFIG_REASSOCIATION=y

# Manufacturing firmware support
CONFIG_MFG_CMD_SUPPORT=y

# Big-endian platform
CONFIG_BIG_ENDIAN=n

# Enable SDIO multi-port Tx aggregation
CONFIG_SDIO_MULTI_PORT_TX_AGGR=y

# Enable SDIO multi-port Rx aggregation
CONFIG_SDIO_MULTI_PORT_RX_AGGR=y


# SDIO suspend/resume
CONFIG_SDIO_SUSPEND_RESUME=y

# Use static link for app build
export CONFIG_STATIC_LINK=n

################################################################################
# Select Platform Tools
################################################################################

MODEXT = ko
EXTRA_CFLAGS += -I$(PWD)/mlan
EXTRA_CFLAGS += -DLINUX
```

```
# KERNELDIR point to the installed kernel directory.
# KERNELDIR can be set on the command line,
# make KERNELDIR=/usr/src/arm/<arch-bsp-path>
# Alternatively KERNELDIR can be set in the environment.
# Default value for KERNELDIR is set below.
KERNELDIR ?= /usr/src/arm/linux-2.6.29-pxa920

# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=</usr/local/arm/4.1.1/bin/>arm-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# Default value for CROSS_COMPILE is set below.
CROSS_COMPILE ?= /usr/local/arm-marvell-linux-gnueabi/bin/arm-marvell-linux-gnueabi-

# INSTALLDIR specify the path to install the kernel module after
# succesful compilation.
# INSTALLDIR can be set on the command line
# make INSTALLDIR=/tftpboot/<rootfs>
# Alternatively INSTALLDIR can be set in the environment.
# Default value for INSTALL is set below.
INSTALLDIR ?= /tftpboot/pxa9xx/root

# ARCH specifies the architecture of the target processor, this kernel
# module will run.
# ARCH can be set on the command line
# make ARCH=<arm/i386>
# Alternatively ARCH can be set in the environment
# Default values of ARCH for specific platform are set below.
ARCH ?= arm

LD += -S

BINDIR = ../bin_sd8787

################################################################################
# Compiler Flags
################################################################################

        EXTRA_CFLAGS += -I$(KERNELDIR)/include

        EXTRA_CFLAGS += -DFPNUM='"57"'

ifeq ($(CONFIG_DEBUG),1)
        EXTRA_CFLAGS += -DDEBUG_LEVEL1
endif

ifeq ($(CONFIG_DEBUG),2)
        EXTRA_CFLAGS += -DDEBUG_LEVEL1
        EXTRA_CFLAGS += -DDEBUG_LEVEL2
        DBG=    -dbg
endif

ifeq ($(CONFIG_PROC_DEBUG),y)
        EXTRA_CFLAGS += -DPROC_DEBUG
        export CONFIG_PROC_DEBUG
endif

ifeq ($(CONFIG_STA_SUPPORT),y)
        EXTRA_CFLAGS += -DSTA_SUPPORT
ifeq ($(CONFIG_REASSOCIATION),y)
        EXTRA_CFLAGS += -DREASSOCIATION
endif
endif

ifeq ($(CONFIG_UAP_SUPPORT),y)
        EXTRA_CFLAGS += -DUAP_SUPPORT
endif

ifeq ($(CONFIG_MFG_CMD_SUPPORT),y)
        EXTRA_CFLAGS += -DMFG_CMD_SUPPORT
endif
```

```
ifeq ($(CONFIG_BIG_ENDIAN),y)
        EXTRA_CFLAGS += -DBIG_ENDIAN_SUPPORT
endif

ifeq ($(CONFIG_SDIO_MULTI_PORT_TX_AGGR),y)
        EXTRA_CFLAGS += -DSDIO_MULTI_PORT_TX_AGGR
endif

ifeq ($(CONFIG_SDIO_MULTI_PORT_RX_AGGR),y)
        EXTRA_CFLAGS += -DSDIO_MULTI_PORT_RX_AGGR
endif


ifeq ($(CONFIG_SDIO_SUSPEND_RESUME),y)
        EXTRA_CFLAGS += -DSDIO_SUSPEND_RESUME
endif

############################################################################
# Make Targets
############################################################################

ifneq ($(KERNELRELEASE),)

MOALOBJS =      mlinux/moal_main.o \
                mlinux/moal_ioctl.o \
                mlinux/moal_shim.o

MLANOBJS =      mlan/mlan_shim.o mlan/mlan_init.o \
                mlan/mlan_txrx.o \
                mlan/mlan_cmdevt.o mlan/mlan_misc.o \
                mlan/mlan_module.o

MLANOBJS += mlan/mlan_wmm.o
MLANOBJS += mlan/mlan_sdio.o
MLANOBJS += mlan/mlan_11n_aggr.o
MLANOBJS += mlan/mlan_11n_rxreorder.o
MLANOBJS += mlan/mlan_11n.o
ifeq ($(CONFIG_STA_SUPPORT),y)
MLANOBJS += mlan/mlan_11d.o
MLANOBJS += mlan/mlan_11h.o
MLANOBJS += mlan/mlan_meas.o
MLANOBJS += mlan/mlan_cfp.o \
                        mlan/mlan_scan.o \
                        mlan/mlan_sta_ioctl.o \
                        mlan/mlan_sta_rx.o \
                        mlan/mlan_sta_tx.o \
                        mlan/mlan_sta_event.o \
                        mlan/mlan_sta_cmd.o \
                        mlan/mlan_sta_cmdresp.o \
                        mlan/mlan_join.o
MOALOBJS += mlinux/moal_priv.o \
            mlinux/moal_wext.o
endif
ifeq ($(CONFIG_UAP_SUPPORT),y)
MLANOBJS += mlan/mlan_uap_ioctl.o
MLANOBJS += mlan/mlan_uap_cmdevent.o
MLANOBJS += mlan/mlan_uap_txrx.o
MOALOBJS += mlinux/moal_uap.o
MOALOBJS += mlinux/moal_uap_priv.o
MOALOBJS += mlinux/moal_uap_wext.o
endif

ifdef CONFIG_PROC_FS
MOALOBJS += mlinux/moal_proc.o
ifeq ($(CONFIG_PROC_DEBUG),y)
MOALOBJS += mlinux/moal_debug.o
endif
endif

obj-m := mlan.o
mlan-objs := $(MLANOBJS)
MOALOBJS += mlinux/moal_sdio_mmc.o
obj-m += sd8xxx.o
sd8xxx-objs := $(MOALOBJS)
```

```
# Otherwise we were called directly from the command line; invoke the kernel build system.
else

default:
	$(MAKE) -C $(KERNELDIR) M=$(PWD) ARCH=$(ARCH) CROSS_COMPILE=$(CROSS_COMPILE) modules

endif

###########################################################################

export          CC LD EXTRA_CFLAGS KERNELDIR

ifeq ($(CONFIG_STA_SUPPORT),y)
ifeq ($(CONFIG_UAP_SUPPORT),y)
.PHONY: mapp/mlanconfig mapp/mlan2040coex mapp/mlanevent mapp/uaputl clean distclean
else
.PHONY: mapp/mlanconfig mapp/mlanevent mapp/mlan2040coex clean distclean
endif
else
ifeq ($(CONFIG_UAP_SUPPORT),y)
.PHONY: mapp/mlanevent mapp/uaputl clean distclean
endif
endif
	@echo "Finished Making Marvell Wlan Linux Driver"

ifeq ($(CONFIG_STA_SUPPORT),y)
mapp/mlanconfig:
	$(MAKE) -C $@
mapp/mlan2040coex:
	$(MAKE) -C $@
endif
ifeq ($(CONFIG_UAP_SUPPORT),y)
mapp/uaputl:
	$(MAKE) -C $@
endif
mapp/mlanevent:
	$(MAKE) -C $@

echo:

build:          echo default

	@if [ ! -d $(BINDIR) ]; then \
		mkdir $(BINDIR); \
	fi
	cp -f mlan.$(MODEXT) $(BINDIR)/mlan$(DBG).$(MODEXT)
	cp -f sd8xxx.$(MODEXT) $(BINDIR)/sd8787$(DBG).$(MODEXT)

ifeq ($(CONFIG_STA_SUPPORT),y)
	cp -f README $(BINDIR)
	$(MAKE) -C mapp/mlanconfig $@ INSTALLDIR=$(BINDIR)
	$(MAKE) -C mapp/mlan2040coex $@ INSTALLDIR=$(BINDIR)
endif
ifeq ($(CONFIG_UAP_SUPPORT),y)
	cp -f README_UAP $(BINDIR)
	$(MAKE) -C mapp/uaputl $@ INSTALLDIR=$(BINDIR)
endif
	$(MAKE) -C mapp/mlanevent $@ INSTALLDIR=$(BINDIR)

clean:
	-find . -name "*.o" -exec rm {} \;
	-find . -name "*.ko" -exec rm {} \;
	-find . -name ".*.cmd" -exec rm {} \;
	-find . -name "*.mod.c" -exec rm {} \;
	-find . -name "Module.symvers" -exec rm {} \;
	-find . -name "Module.markers" -exec rm {} \;
	-find . -name "modules.order" -exec rm {} \;
	-rm -rf .tmp_versions
ifeq ($(CONFIG_STA_SUPPORT),y)
	$(MAKE) -C mapp/mlanconfig $@
	$(MAKE) -C mapp/mlan2040coex $@
endif
ifeq ($(CONFIG_UAP_SUPPORT),y)
	$(MAKE) -C mapp/uaputl $@
```

```
endif
	$(MAKE) -C mapp/mlanevent $@

install: default

	cp -f mlan.$(MODEXT) $(INSTALLDIR)/mlan$(DBG).$(MODEXT)
	cp -f sd8xxx.$(MODEXT) $(INSTALLDIR)/sd8787$(DBG).$(MODEXT)
	echo "sd8787 Driver Installed"

distclean:
	-find . -name "*.o" -exec rm {} \;
	-find . -name "*.orig" -exec rm {} \;
	-find . -name "*.swp" -exec rm {} \;
	-find . -name "*.*~" -exec rm {} \;
	-find . -name "*~" -exec rm {} \;
	-find . -name "*.d" -exec rm {} \;
	-find . -name "*.a" -exec rm {} \;
	-find . -name "tags" -exec rm {} \;
	-find . -name ".*" -exec rm -rf 2> /dev/null \;
	-find . -name "*.ko" -exec rm {} \;
	-find . -name ".*.cmd" -exec rm {} \;
	-find . -name "*.mod.c" -exec rm {} \;
	-rm -rf .tmp_versions
ifeq ($(CONFIG_STA_SUPPORT),y)
	$(MAKE) -C mapp/mlanconfig $@
	$(MAKE) -C mapp/mlan2040coex $@
endif
ifeq ($(CONFIG_UAP_SUPPORT),y)
	$(MAKE) -C mapp/uaputl $@
endif
	$(MAKE) -C mapp/mlanevent $@

# End of file
```