NEWRACOM
A New semiconductor Company for New Era

NRC7292 Evaluation Kit User Guide

(Host Driver Porting)

Ultra-low power & Long-range Wi-Fi

Ver 1.2
May 30, 2020

NEWRACOM, Inc.

NRC7292 Evaluation Kit User Guide (Host Driver Porting)
Ultra-low power & Long-range Wi-Fi

Office

NEWRACOM, Inc.
25361 Commercentre Drive, Lake Forest, CA 92630 USA
http://www.NEWRACOM.com

Contents

List of Tables

List of Figures

1 Overview

This guide introduces the overall SW structure of NRC7292 host driver and gives some tips for applying the driver to other Linux hosts.

1.1 Software structure

As seen in Figure 1.1, NRC7292 host driver uses Linux Kernel features, SPI, GPIO, IRQ, mac80211, and netlink socket. SPI (Slave) is used for I/O interface between the host, the NRC7292 module, and mac80211 for SW MAC, which is incorporated with MAC in the NRC7292 module. Netlink socket is used to communicate with user applications like CLI shell on host. GPIO including IRQ is used as an external interrupt source for flow control while communicating via SPI. The host applications hostapd, needs to be installed on the host to be operated as 11ah AP or wpa_supplicant as 11ah STA.

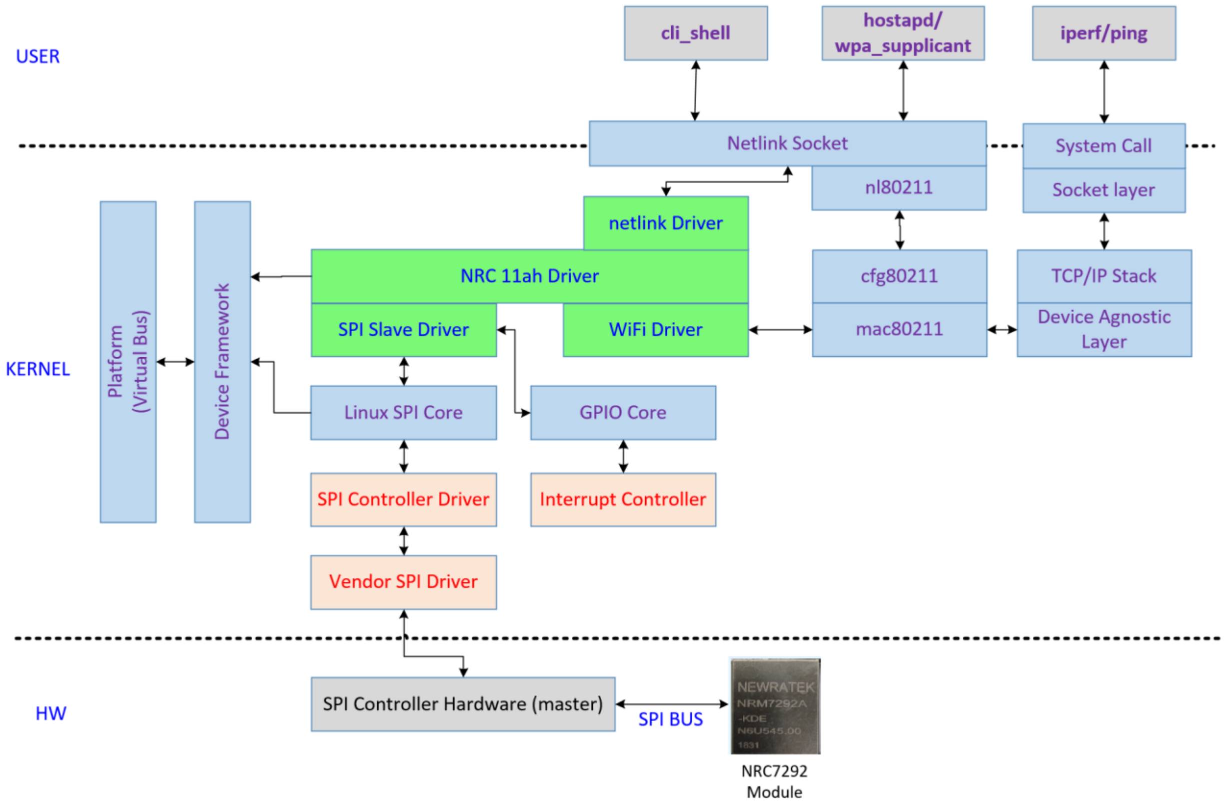


Figure 1.1 SW structure of NRC7292 host driver

1.2 Software components

NRC7292 host driver makes use of Linux Kernel APIs of GPIO, IRQ, SPI, and mac80211 so the host can support these components. To use Linux SPI/GPIO/IRQ framework or subsystem, each controller drivers provided by SoC vendor should be applied on host. If the framework is not supported on host, then the driver will be modified using other APIs (like vendor specific APIs). When it comes to mac80211, mac/cfg80211 modules should be built and loaded before loading NRC7292 host driver.

1.3 Hardware components

To communicate with NRC7292 module, SPI and GPIO (1 PIN) should be designated on Host. Figure 1.2 shows SPI and GPIO block on each Host (Raspberry Pi3) and NRC7292 Module. SPI is the main interface between Host, NRC7292 Module, GPIO, and also used as external Interrupt source for flow control.

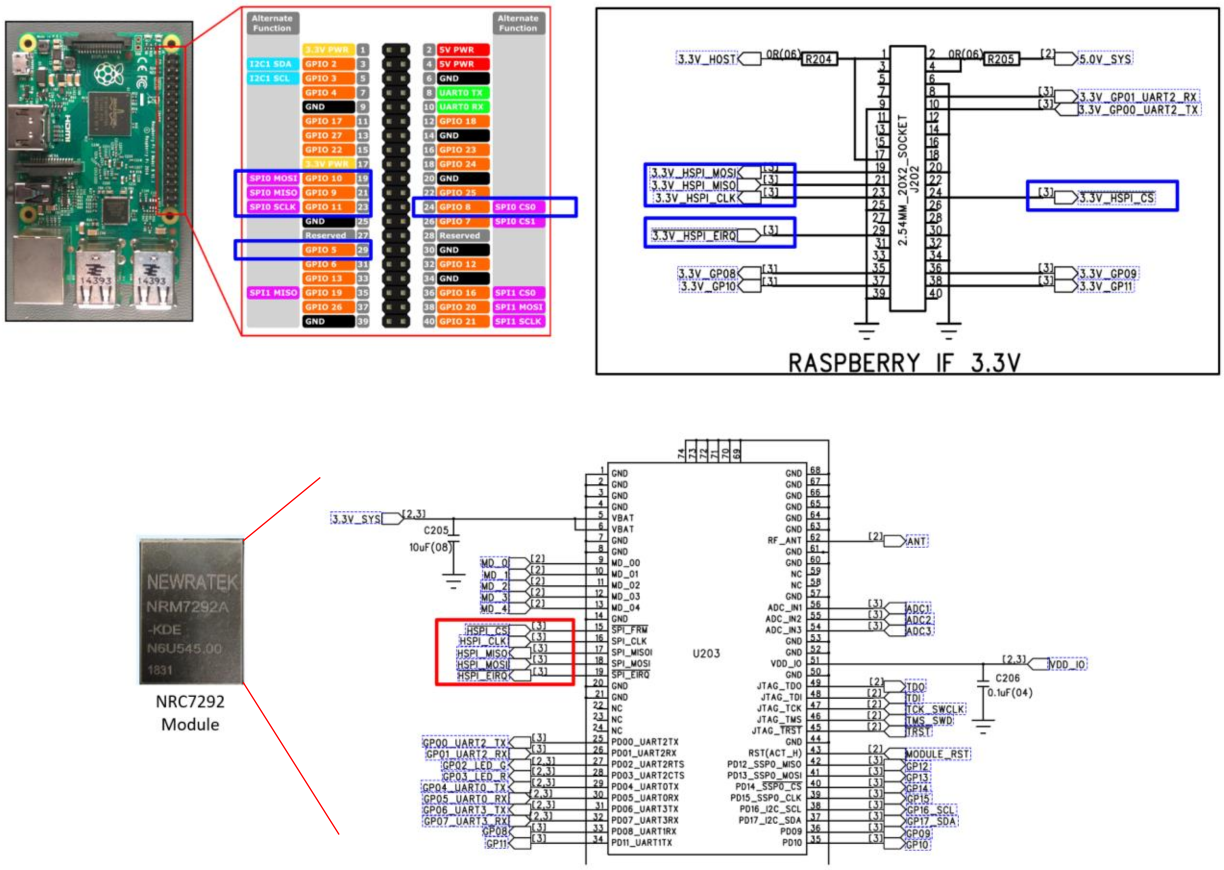


Figure 1.2 HW components on RP3 host and NRC7292 Module

2 How to build NRC7292 host driver

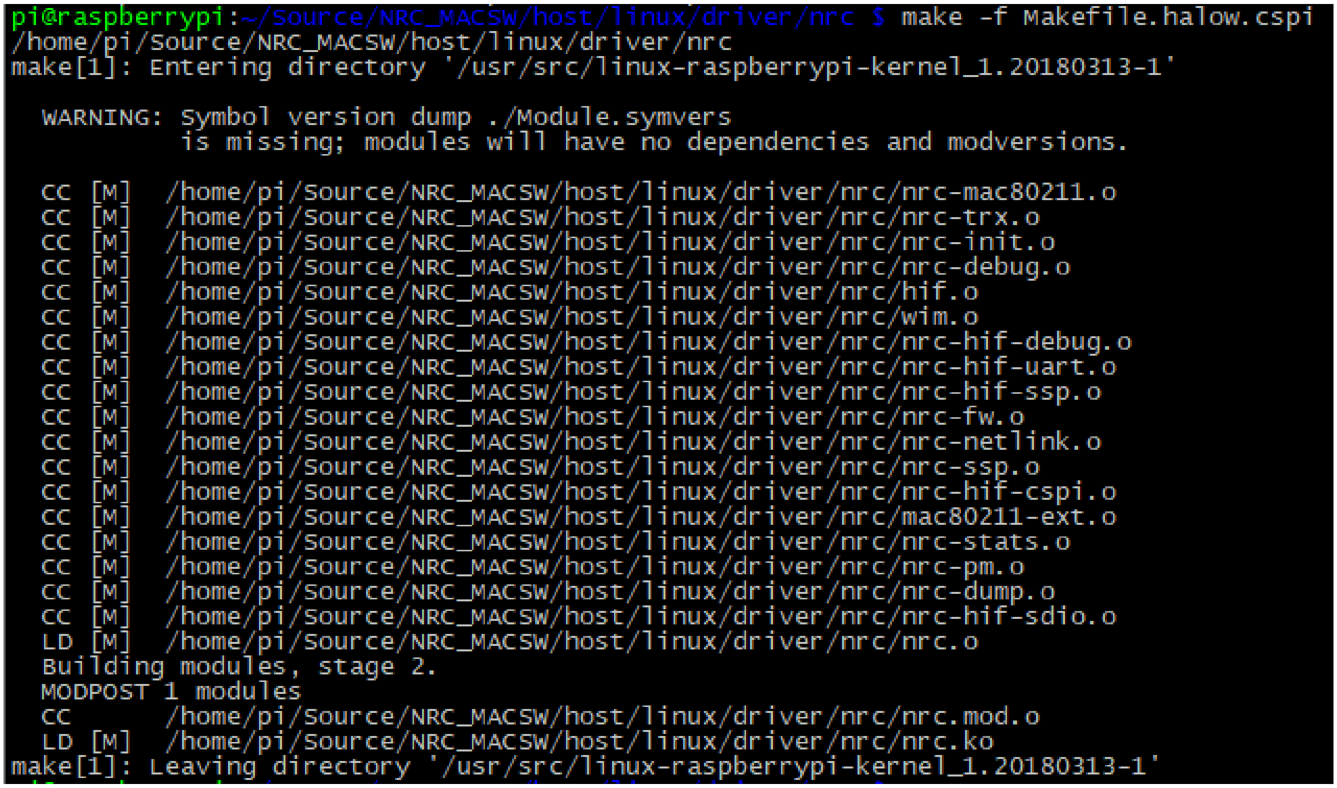
This chapters shows how to build NRC7292 host driver. There are 2 ways to build the driver: 1) direct-built on the host and 2) cross-compile on other hosts.

2.1 Direct compile

Before building NRC7292 host driver on host, Linux kernel sources or headers should be prepared on the host. Our drivers can cover Linux kernel version from 4.0.x to 4.14.x.; in order to build properly, changing the kernel path in Makefile is needed.

2.2 Cross compile

It is a similar procedure as direct compile. But the only difference is the preparing the cross the compiler on your PC.



```
pi@raspberrypi:~/Source/NRC_MACSW/host/linux/driver/nrc $ make -f Makefile.halow.cspi
/home/pi/Source/NRC_MACSW/host/linux/driver/nrc
make[1]: Entering directory '/usr/src/linux-raspberrypi-kernel_1.20180313-1'

  WARNING: Symbol version dump ./Module.symvers
           is missing; modules will have no dependencies and modversions.

  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-mac80211.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-trx.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-init.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-debug.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/hif.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/wim.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-hif-debug.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-hif-uart.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-hif-ssp.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-fw.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-netlink.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-ssp.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-hif-cspi.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/mac80211-ext.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-stats.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-pm.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-dump.o
  CC [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc-hif-sdio.o
  LD [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc.mod.o
  LD [M]  /home/pi/Source/NRC_MACSW/host/linux/driver/nrc/nrc.ko
make[1]: Leaving directory '/usr/src/linux-raspberrypi-kernel_1.20180313-1'
```

Figure 2.1 Compile log

3 Source-code tree

Table 3.1 shows major files for NRC7292 host driver and description of them. If all the SW and HW components mentioned in Chap.1 is ready, then there is almost no need to modify source codes.

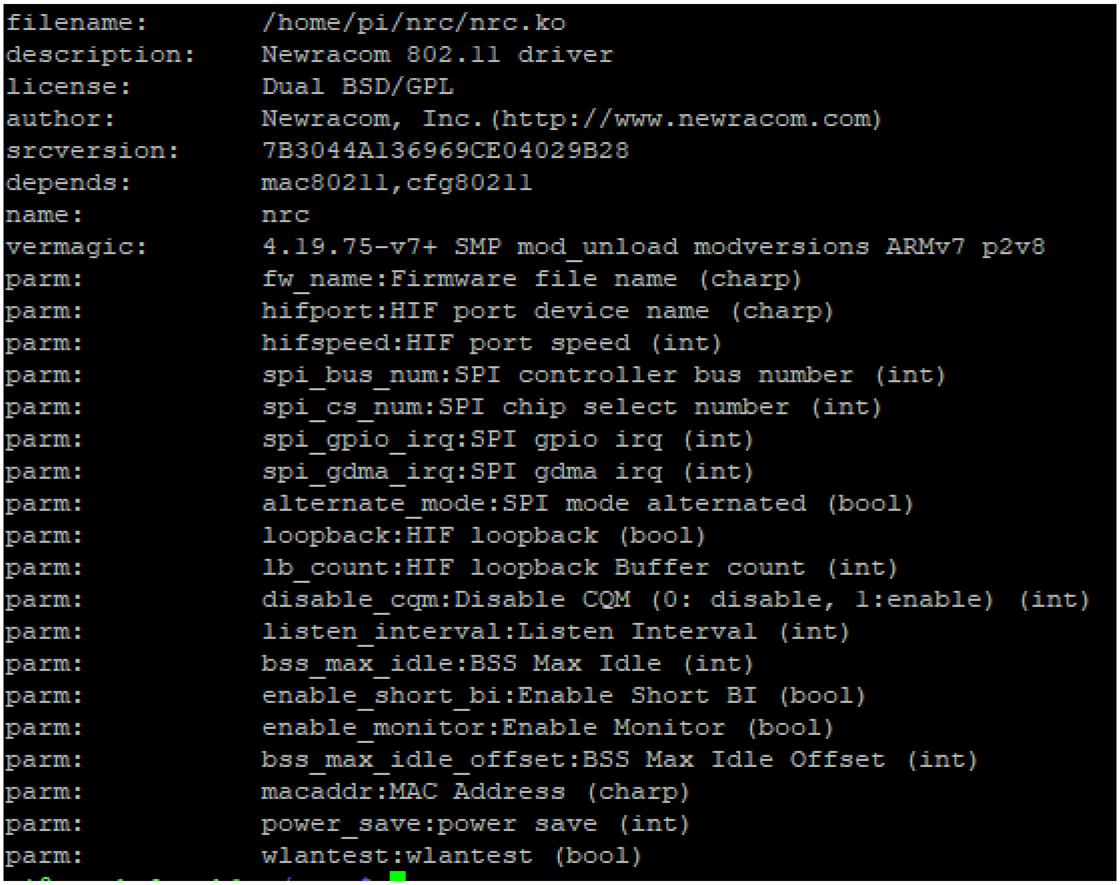
Table 3.1 Host driver files

	Category	Description
nrc-init.*	Module Initialization Linux Platform driver	Define module and module parameters Register platform device and driver
hif.*	HAL (HW Adaptation layer)	Wrapper functions for I/O
nrc-hif-cspi.*	HSPI driver	Functions for HSPI
nrc-mac80211.*	mac80211	Register driver to mac80211 Define mac80211 parameters
nrc-trx.*	Data Path	Define functions for data path
nrc-fw.*	Firmware	Define functions for FW download from host to target
nrc-netlink.*	Netlink socket	Define functions for netlink communication
nrc-pm.*	Power Management	Define functions for Wi-Fi Power Management
nrc-debug.*	Debug	Define functions for debugging
mac80211-ext.*	Utility function	Utility function about ieee80211
nrc-dump.*	Debug	Save Core Dump file when F/W Asserted
nrc-hif-sdio.*	SDIO driver	Functions for SDIO (No needed)
nrc-hif-ssp.*, nrc-ssp.*	SPI driver	Functions for SPI (Not needed)
nrc-hif-uart.*	UART driver	Functions for UART (Not needed)
nrc-pm.*	Power Save	Function for Power Save
nrc-recovery.*	Recovery	Function for Recovery that is work as watchdog function for firmware
nrc-stats.*	Statistics	Function for Statistics to check a SNR/RSSI
wim.*	WIM	Function for handling the information message between host and firmware

nrc-vendor.h	Vender IE	Definition constant/type for Vender IE
fastboot-cm0.h	Boot	Second boot loader for CM0
nrc-build-config.h	Build	Define a definition/constant for building and configuration for host driver
nrc-wim-types.h	WIM	Define a data type for the WIM

3.1 Module Parameters

As seen in Figure 1.1, NRC7292 host driver support module parameters.



```
filename:       /home/pi/nrc/nrc.ko
description:    Newracom 802.11 driver
license:        Dual BSD/GPL
author:         Newracom, Inc.(http://www.newracom.com)
srcversion:     7B3044A136969CE04029B28
depends:        mac80211,cfg80211
name:           nrc
vermagic:       4.19.75-v7+ SMP mod_unload modversions ARMv7 p2v8
parm:           fw_name:Firmware file name (charp)
parm:           hifport:HIF port device name (charp)
parm:           hifspeed:HIF port speed (int)
parm:           spi_bus_num:SPI controller bus number (int)
parm:           spi_cs_num:SPI chip select number (int)
parm:           spi_gpio_irq:SPI gpio irq (int)
parm:           spi_gdma_irq:SPI gdma irq (int)
parm:           alternate_mode:SPI mode alternated (bool)
parm:           loopback:HIF loopback (bool)
parm:           lb_count:HIF loopback Buffer count (int)
parm:           disable_cqm:Disable CQM (0: disable, 1:enable) (int)
parm:           listen_interval:Listen Interval (int)
parm:           bss_max_idle:BSS Max Idle (int)
parm:           enable_short_bi:Enable Short BI (bool)
parm:           enable_monitor:Enable Monitor (bool)
parm:           bss_max_idle_offset:BSS Max Idle Offset (int)
parm:           macaddr:MAC Address (charp)
parm:           power_save:power save (int)
parm:           wlantest:wlantest (bool)
```

Figure 3.1 Module Parameters

According to HW configuration, some SPI and GPIO parameters like chip_select, bus_number, max_speed_hz, etc. should be changed when driver is inserted to kernel. For SPI, user can set using spi_bus_num, spi_cs_num, spi_gpio_irq and hifspeed in module parameters.

4 Revision history

Revision No	Date	Comments
Ver 1.0	11/01/2018	Initial version for customer release created
Ver 1.1	11/07/2019	Add 3.1 module parameters and delete H/W dependency codes
Ver 1.2	05/30/2020	Update source-code tree