



# **NRC7292 Application Note**

## **(FT232H\_USB\_SPI)**

**Ultra-low power & Long-range Wi-Fi**

**Ver 1.1**  
**Nov 26, 2021**

**NEWRACOM, Inc.**

## **NRC7292 Application Note (FT232H\_USB\_SPI)**

### **Ultra-low power & Long-range Wi-Fi**

#### **© 2021 NEWRACOM, Inc.**

All right reserved. No part of this document may be reproduced in any form without written permission from Newracom.

Newracom reserves the right to change in its products or product specification to improve function or design at any time without notice.

#### **Office**

Newracom, Inc.

25361 Commercentre Drive, Lake Forest, CA 92630 USA

<http://www.newracom.com>

# Contents

<b>1</b>	<b>Overview.....</b>	<b>6</b>
<b>2</b>	<b>C232HM MPSSE Cable connection .....</b>	<b>7</b>
<b>3</b>	<b>FT232H USB-SPI bridge driver .....</b>	<b>9</b>
<b>4</b>	<b>Getting started .....</b>	<b>11</b>
4.1	Building the spi-ft232h driver as an external module. ....	11
4.2	Loading the spi-ft232h driver into the kernel. ....	13
4.3	Checking the SPI bus and GPIO base numbers. ....	14
4.4	Loading the nrc7292 driver into the kernel. ....	15
4.5	Removing the nrc7292 driver from the kernel. ....	17
4.6	Removing the spi-ft232h driver from the kernel. ....	17
<b>5</b>	<b>Revision history.....</b>	<b>18</b>

# List of Tables

Table 2.1	C232HM MPSSE Cable - SPI Signals.....	7
Table 3.1	FT232H USB-SPI Bridge Driver - File Descriptions.....	10
Table 4.1	FT232H GPIO Pin Number .....	15
Table 4.2	FT232H and SPI related parameters .....	16

# List of Figures

Figure 1.1 C232HM MPSSE Cable Overview ..... 6

Figure 2.1 C232HM MPSSE Cable - Wire Colors ..... 7

Figure 2.2 C232HM MPSSE Cable - Wire Connection with NRC7282 EVK..... 8

Figure 3.1 FT232H USB-SPI Bridge Driver - Source Tree ..... 9

Figure 3.2 FT232H USB-SPI Bridge Driver - Layer Diagram ..... 10

# 1 Overview

NRC7292, a Wi-Fi Halow Chip, supports high speed SPI slave controller to communicate with a Host that supports SPI Master.

Alternatively, it can support USB interface with FTDI FT232H, which is a single channel USB 2.0 Hi-Speed (480Mb/s) to UART/FIFO IC. The FT232H can be configured in a variety of industry standard serial or parallel interfaces, such as MPSSE - JTAG, SPI, I2C.

The FT232H must be configured as SPI master to communicate with NRC7292. It is possible with the FT232H USB-SPI bridge driver described in this application note. Users can evaluate it with the NRC7292 EVK and FTDI C232HM MPSSE Cable.



Part No: C232HM-DDHSL-0

USB to Hi-Speed SPI/I2C/JTAG Serial Adapter Cable w/Embedded Electronics, LEDs, 3.3V

USB 2.0 Hi-Speed to MPSSE Cable (SPI/I2C/JTAG master) with +3.3V digital level signals.

Data Rates: 30 Mbps

Interfaces: MPSSE

End Connector: 10 way 0.1" female pin headers

Cable Details: 10 core, 6.5mm UL2464 24 AWG

Channels: 1

USB Class: FTDI Drivers

Max Power Output: +3.3V / 250mA

Operating Temperature: -40°C to +85°C

USB Speed: High Speed (480MBps)

USB Connector: Type A

Cable Length: 0.5m

Packages: N/A

I/O Voltage: 3.3V

Virtual Com Port: Yes

FTDI Internal IC: FT232H

Figure 1.1 C232HM MPSSE Cable Overview

For more information on the FTDI C232MH MPSSE Cable, refer to the link (<https://ftdichip.com/products/c232hm-ddhsl-0-2/>) and related document in doc folder.

## 2 C232HM MPSSE Cable connection

Figure 2.1 and Table 2.1 show the wire colors and signal descriptions of the C232HM MPSSE Cable in MPSSE SPI mode.

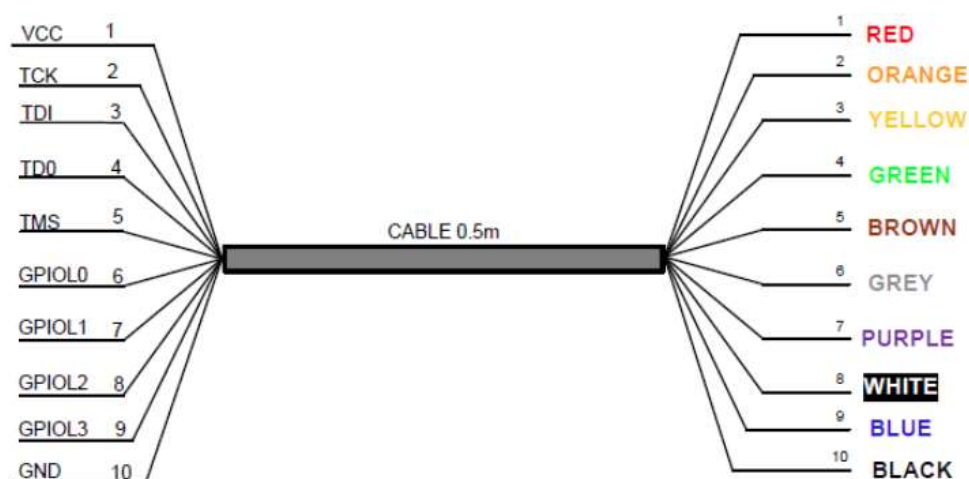


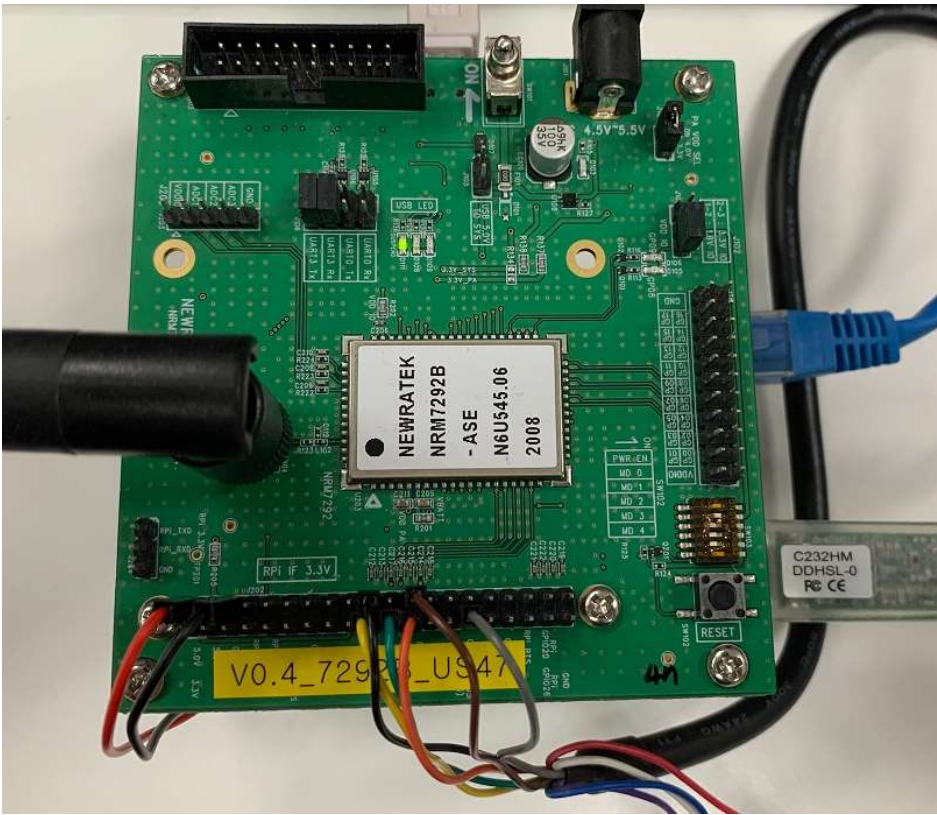
Figure 2.1 C232HM MPSSE Cable - Wire Colors

Colour	Pin Number	Name	Type	Description
Orange	2	SK	Output	Serial Clock
Yellow	3	DO	Output	Serial data output
Green	4	DI	Input	Serial Data Input
Brown	5	CS	Output	Serial Chip Select

Table 2.1 C232HM MPSSE Cable - SPI Signals

The FT232H supports only one chip select pin in MPSSE SPI mode. Therefore, it cannot communicate with more than one SPI slave device.

Figure 2.2 shows the wire connection between NRC7292 Evaluation Kit and C232HM MPSSSE Cable.



Raspberry Pi 40 Pin Header		NRC7292 EVB 40 Pin Header		FTDI C232HM-DDHSL-0
Pin 2 (5V)	←- - - - -→	Pin 2 (5V)		-
Pin 6 (GND)	←- - - - -→	Pin 6 (GND)		-
-		-		-
-		Pin 19 (MOSI)	←- - - - -→	YELLOW (DO)
-		Pin 20 (GND)	←- - - - -→	BLACK (GND)
-		Pin 21 (MISO)	←- - - - -→	GREEN (DI)
-		Pin 23 (CLK)	←- - - - -→	ORANGE (SK)
-		Pin 24 (CS)	←- - - - -→	BROWN (CS)
-		-		-
-		Pin 29 (EIRQ)	←- - - - -→	GRAY (GPIOL0)

Figure 2.2 C232HM MPSSSE Cable - Wire Connection with NRC7282 EVK



### 3 FT232H USB-SPI bridge driver

The FT232H USB-SPI bridge driver is based on the FTDI drivers implemented by DENX. The DENX FTDI drivers can be found at <https://patchwork.kernel.org> and is under the GNU Public License version 2 (GPLv2).

- FTDI FT232H Interface Driver  
: <https://patchwork.kernel.org/project/spi-devel-general/patch/20190221202506.17744-2-agust@denx.de/>
- FTDI MPSSE SPI Controller Driver  
: <https://patchwork.kernel.org/project/spi-devel-general/patch/20190221202506.17744-3-agust@denx.de/>

Figure 3.1 and Table 3.1 show the source tree and file descriptions for the FT232H USB-SPI bridge driver.

Source Path : nrc7292\_sw\_pkg/package/host/src/ft232h-usb-spi/

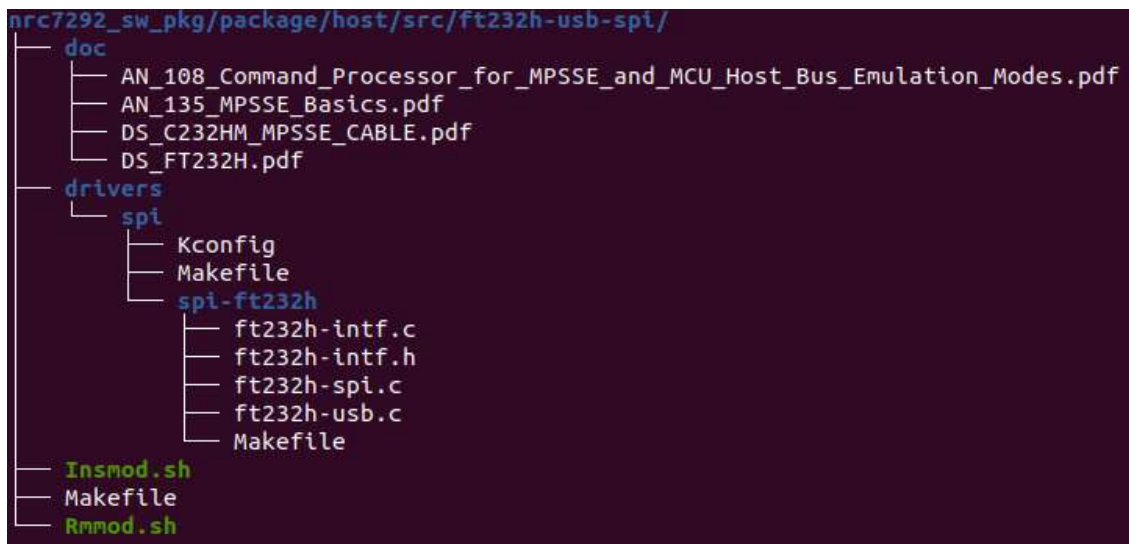


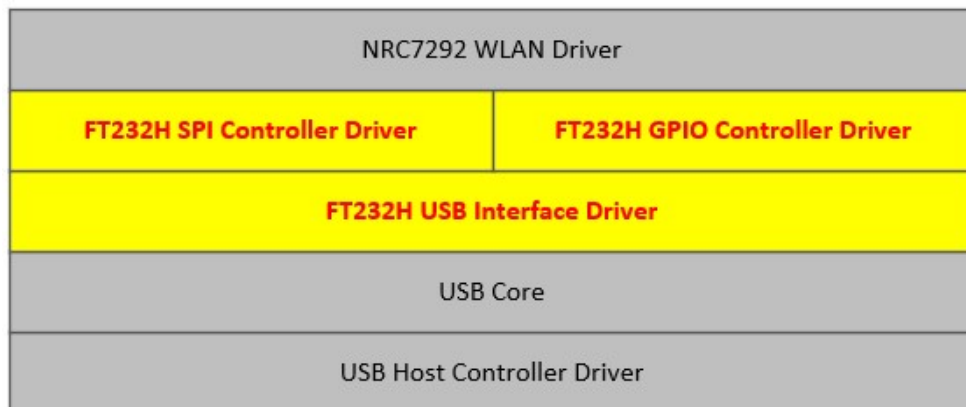
Figure 3.1 FT232H USB-SPI Bridge Driver - Source Tree

File Name	Description
doc/	Application Note, Datasheet, ...
drivers/spi/Kconfig	Kconfig for kbuild
drivers/spi/Makefile	Makefile for kbuild

drivers/spi/spi-ft232h/ft232h-intf.[ch]	FT232H interface driver
drivers/spi/spi-ft232h/ft232h-spi.c	FT232H MPSSE SPI driver
drivers/spi/spi-ft232h/ft232h-usb.c	FT232H USB device driver
drivers/spi/spi-ft232h/Makefile	Makefile for kbuild
Makefile	Makefile to build as external module
Insmod.sh	Shell script to insert a module into the kernel
Rmmod.sh	Shell script to remove a module from the kernel

**Table 3.1 FT232H USB-SPI Bridge Driver - File Descriptions**

The FT232H USB-SPI bridge driver registers three types of drivers into the kernel as shown in Figure 3.2.

**Figure 3.2 FT232H USB-SPI Bridge Driver - Layer Diagram**

However, it is built as a single driver module. (spi-ft232h.ko)

## 4 Getting started

### 4.1 Building the spi-ft232h driver as an external module.

```
$ cd ${FT232H_USB_SPI_SOURCE_TOP}
```

```
$ make clean
```

```
pi@raspberrypi:~/ft232h-usb-spi $ make clean
make[1]: Entering directory '/usr/src/linux-headers-4.14.70-v7+'
CLEAN /home/pi/ft232h-usb-spi/drivers/spi/.tmp_versions
CLEAN /home/pi/ft232h-usb-spi/drivers/spi/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-4.14.70-v7+'
```

```
$ make [debug]
```

```
pi@raspberrypi:~/ft232h-usb-spi $ make
make[1]: Entering directory '/usr/src/linux-headers-4.14.70-v7+'
CC [M] /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/ft232h-intf.o
CC [M] /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/ft232h-spi.o
CC [M] /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/ft232h-usb.o
LD [M] /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/spi-ft232h.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/spi-ft232h.mod.o
LD [M] /home/pi/ft232h-usb-spi/drivers/spi/spi-ft232h/spi-ft232h.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.14.70-v7+'
```

If there are no errors, a spi-ft232h.ko file will be created under the "drivers/spi/spi-ft232h" directory.

```
$ modinfo ./drivers/spi/spi-ft232h/spi-ft232h.ko
```

```
pi@raspberrypi:~/ft232h-usb-spi $ modinfo ./drivers/spi/spi-ft232h/spi-ft232h.ko
filename:          /home/pi/ft232h-usb-spi./drivers/spi/spi-ft232h/spi-ft232h.ko
license:           GPL v2
description:       FTDI FT232H USB-SPI bridge driver
author:            Anatolij Gustschin <agust@denx.de>
author:            Sangbeom Kim <sb.kim@newracom.com>
author:            Newracom, Inc. <www.newracom.com>
alias:             ft232h-usb-spi
srcversion:        F6D9C1D731B72C84B9CE82D
alias:             usb:v0403p6014d*dc*dsc*dp*ic*isc*ip*in*
depends:
name:              spi_ft232h
vermagic:          4.14.70-v7+ SMP mod_unload modversions ARMv7 p2v8
parm:              latency:latency timer value (1ms ~ 255ms, default 1ms) (int)
parm:              spi_bus_num:SPI controller bus number (if negative, dynamic allocation) (int)
parm:              gpio_base_num:GPIO controller base number (if negative, dynamic allocation) (int)
```

**NOTE:**

To integrate the spi-ft232h driver into the Linux kernel source:

- i. Copy the drivers/spi/spi-ft232h directory to the kernel source.

```
$ cd ${FT232H_USB_SPI_SOURCE_TOP}
```

```
$ cp -a drivers/spi/spi-ft232h ${KERNEL_SOURCE_TOP}/drivers/spi
```

- ii. Add the contents of Kconfig and Makefile under drivers/spi to the kernel source.

```
$ vim ${KERNEL_SOURCE_TOP}/drivers/spi/Kconfig
```

```
config SPI_MASTER
#   bool "SPI Master Support"
   bool
   default SPI
   help
       If your system has an master-capable SPI controller (which
       provides the clock and chipselect), you can enable that
       controller and the protocol drivers for the SPI slave chips
       that are connected.

if SPI_MASTER

comment "SPI Master Controller Drivers"

config SPI_FT232H
    tristate "FTDI FT232H SPI controller"
    depends on USB || GPIOLIB || COMPILE_TEST
    help
        FT232H supports SPI in MPSSE mode.
        This driver provides MPSSE SPI controller in master mode.
```

```
$ vim ${KERNEL_SOURCE_TOP}/drivers/spi/Makefile
```

```
# SPI slave protocol handlers
obj-$(CONFIG_SPI_SLAVE_TIME) += spi-slave-time.o
obj-$(CONFIG_SPI_SLAVE_SYSTEM_CONTROL) += spi-slave-system-control.o
obj-$(CONFIG_SPI_FT232H) += spi-ft232h/
```

- iii. Execute Kernel Configuration Menu.

```
$ cd ${KERNEL_SOURCE_TOP}
```

```
$ make menuconfig
```

Kernel Configuration Menu : Device Drivers -> SPI support

```
--- SPI support
[ ] Debug support for SPI drivers
*** SPI Master Controller Drivers ***
< > FTDI FT232H SPI controller (NEW)
```

## 4.2 Loading the spi-ft232h driver into the kernel.

```
$ cd ${FT232H_USB_SPI_SRC_TOP}
```

```
$ ./Insmod.sh -h
```

```
pi@raspberrypi:~/ft232h-usb-spi $ ./Insmod.sh -h
Usage: ./Insmod.sh [options]
Options:
  -l    set the latency time in msec (default : 1 msec)
  -s    set the SPI bus number (default : 3)
  -g    set the GPIO base number (default : 500)
  -h    print this help
```

The latency time is the amount of time to wait before sending an incomplete USB packet from the peripheral back to the host. For quick responses from the peripheral, it is set to 1 by default. 255 indicates the slowest responses.

```
$ ./Insmod.sh [options]
```

```
pi@raspberrypi:~/ft232h-usb-spi $ ./Insmod.sh
sudo insmod ./drivers/spi/spi-ft232h/spi-ft232h.ko latency=1 spi_bus_num=3 gpio_base_num=500
export gpiochip0 gpiochip100 gpiochip128 gpiochip500 unexport
spi0 spi3
```

```
$ lsmod
```

```
pi@raspberrypi:~/ft232h-usb-spi $ lsmod
Module      Size  Used by
spi_ft232h  20480  0
```

### NOTE:

The USB PID of the C232HM MPSSE cable is 0x6014, so the FTDI SIO driver for the FT232H may be loaded into the kernel. It should be removed from the kernel.

```
$ dmesg
```

```
usb 1-1.5: new high-speed USB device number 6 using dwc_otg
usb 1-1.5: New USB device found, idVendor=0403, idProduct=6014
usb 1-1.5: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.5: Product: C232HM-DDHSL-0
usb 1-1.5: Manufacturer: FTDI
usb 1-1.5: SerialNumber: FT4VTGE3
usbcore: registered new interface driver usbserial
usbcore: registered new interface driver usbserial_generic
usbserial: USB Serial support registered for generic
usbcore: registered new interface driver ftdi_sio
usbserial: USB Serial support registered for FTDI USB Serial Device
ftdi_sio 1-1.5:1.0: FTDI USB Serial Device converter detected
usb 1-1.5: Detected FT232H
usb 1-1.5: FTDI USB Serial Device converter now attached to ttyUSB0
```



```
$ lsmod
```

Module	Size	Used by
ftdi_sio	45056	0
usbserial	36864	1 ftdi_sio
fuse	114688	3

To blacklist the ftdi\_sio module:

```
$ sudo vim /etc/modprobe.d/blacklist-ftdi.conf
```

```
1 blacklist ftdi_sio
```

### 4.3 Checking the SPI bus and GPIO base numbers.

If the spi\_bus\_num and gpio\_base\_num of module parameters are negative, the SPI bus and GPIO base numbers are assigned while the spi-ft232h driver is registered into the kernel.

```
pi@raspberrypi:~/ft232h-usb-spi $ ./Insmod.sh
sudo insmod ./drivers/spi/spi-ft232h/spi-ft232h.ko latency=1 spi_bus_num=-1 gpio_base_num=-1
export gpiochip0 gpiochip100 gpiochip128 gpiochip500 unexport
spi0 spi3
```

So, the user should check these numbers before loading the nrc7292 driver into the kernel.

If the SPI bus number is 3 and the GPIO base number is 500, the device information can be found in the /sys/class/spi\_master and /sys/class/gpio directories.

```
$ cat /sys/class/spi_master/spi3/device/modalias
```

```
$ cat /sys/class/gpio/gpiochip500/label
```

```
$ cat /sys/class/gpio/gpiochip500/base
```

```
$ cat /sys/class/gpio/gpiochip500/ngpio
```

```
pi@raspberrypi:~/ft232h-usb-spi $ cat /sys/class/spi_master/spi3/device/modalias
platform:ftdi-mpsse-spi
pi@raspberrypi:~/ft232h-usb-spi $ cat /sys/class/gpio/gpiochip500/label
ftdi-mpsse-gpio.0
pi@raspberrypi:~/ft232h-usb-spi $ cat /sys/class/gpio/gpiochip500/base
500
pi@raspberrypi:~/ft232h-usb-spi $ cat /sys/class/gpio/gpiochip500/ngpio
12
```

When the GPIO base number is 500, FT232H GPIO pins are assigned as in the following table.

GPIO Name	GPIO Index	GPIO Number
MPSSE_GPIOL0	0	500
MPSSE_GPIOL1	1	501
MPSSE_GPIOL2	2	502
MPSSE_GPIOL3	3	503
*MPSSE_GPIOH0	4	504
*MPSSE_GPIOH1	5	505
*MPSSE_GPIOH2	6	506
*MPSSE_GPIOH3	7	507
*MPSSE_GPIOH4	8	508
*MPSSE_GPIOH5	9	509
*MPSSE_GPIOH6	10	510
*MPSSE_GPIOH7	11	511

\*) C232HM MPSSE Cable has no wires for MPSSE GPIOH.

**Table 4.1 FT232H GPIO Pin Number**

## 4.4 Loading the nrc7292 driver into the kernel.

Open the start.py file and set the ft232h\_usb\_spi of the following parameters.

```
#####
# CSPI Conf. (Default)
spi_clock    = 20000000      # SPI Master Clock Frequency
spi_bus_num  = 0             # SPI Master Bus Number
spi_cs_num   = 0             # SPI Master Chipselect Number
spi_gpio_irq = 5             # NRC-CSPI EIRQ GPIO Number
                                # BBB is 60 recommended.
spi_polling_interval = 0     # NRC-CSPI Polling Interval (msec)

#
# NOTE:
# - NRC-CSPI EIRQ Input Interrupt: spi_gpio_irq >= 0 and spi_polling_interval <= 0
# - NRC-CSPI EIRQ Input Polling  : spi_gpio_irq >= 0 and spi_polling_interval > 0
# - NRC-CSPI Registers Polling  : spi_gpio_irq < 0 and spi_polling_interval > 0
#
#-----#
# FT232H USB-SPI Conf. (FT232H CSPI Conf)
ft232h_usb_spi = 0           # FTDI FT232H USB-SPI bridge
                                # 0 : Unused
                                # 1 : NRC-CSPI_EIRQ Input Polling
                                # 2 : NRC-CSPI Registers Polling
#####
```

ft232h_usb_spi	Parameter	Default Value
0 ( Unused )	hif_speed	20,000,000 (Hz)
	spi_bus_num	0
	spi_cs_num	0
	spi_gpio_irq	5
	spi_polling_interval	0
1 ( CSPI EIRQ Input Polling )	hif_speed	15,000,000 (Hz)
	spi_bus_num	3
	spi_cs_num	0
	spi_gpio_irq	500
	spi_polling_interval	50 (msec)
2 ( CSPI Registers Polling )	hif_speed	15,000,000 (Hz)
	spi_bus_num	3
	spi_cs_num	0
	spi_gpio_irq	-1
	spi_polling_interval	50 (msec)

Table 4.2 FT232H and SPI related parameters

```
$ cd ~/nrc_pkg/script
```

```
$ vim start.py
```

Set ft232h\_usb\_spi to 1.

Change the following parameters if necessary.

```
def ft232h_usb():
    global spi_clock, spi_bus_num, spi_gpio_irq, spi_cs_num, spi_polling_interval
    # ft232h_usb_spi
    if int(ft232h_usb_spi) > 0:
        print("[*] use ft232h_usb_spi")
        spi_bus_num = 3
        spi_gpio_irq = 500
        if int(spi_clock) > 15000000:
            spi_clock = 15000000
        if int(spi_cs_num) != 0:
            spi_cs_num = 0
        if int(spi_polling_interval) <= 0:
            spi_polling_interval = 50
        if int(ft232h_usb_spi) != 1:
            spi_gpio_irq = -1
```



- hif\_speed
- spi\_bus\_num
- spi\_gpio\_irq.
- spi\_polling\_interval

**NOTE:**

The SPI clock frequency supported by the FT232H is up 450KHz to 30MHz as (30MHz / n) division, while n is 1 to 65535. So, the supported SPI clock is 30, 15, 10, 7.5, 6, 4.29, 3.75MHz and so on. However, 30 MHz is not available for the NRC7292.

When running as an STA in open mode on US channel:

```
$ ./start.py 0 0 US
```

For more information, refer to the **UG-7292-001-EVK User Guide (Host Mode).pdf** file.

## 4.5 Removing the nrc7292 driver from the kernel.

```
$ cd ~/nrc_pkg/script
```

```
$ ./stop.py
```

## 4.6 Removing the spi-ft232h driver from the kernel.

```
$ cd ${FT232H_USB_SPI_SRC_TOP}
```

```
$ ./Rmmod.sh
```

```
pi@raspberrypi:~/ft232h-usb-spi $ ./Rmmod.sh
export gpiochip0 gpiochip100 gpiochip128 unexport
spi0
```

## 5 Revision history

Revision No	Date	Comments
Ver 1.0	08/31/2021	Initial version
Ver 1.1	11/26/2021	CSPI registers polling added