



**APPLICATION NOTE:**  
**SX-NEWAH Evaluation**  
**with nVidia Jetson**

Silex Technology America  
201 East Sandpointe, Suite 245  
Santa Ana, CA 92707  
Revision 1.0, March 2021

### Revision History

Rev. No.	Date	Revision by	Description
1.0	Mar 15, 2021	T.Nakase	Initial document

## Table of Contents

1. Scope.....	3
2. Equipment.....	3
2.1. Hardware .....	3
2.2. Software.....	3
2.3. Additional Hardware and Software for USB-SPI Bridged Connection (Optional) .....	3
3. Setup nVidia Jetson Nano 2GB Developer Kit.....	3
4. Mount SX-NEWAH-EVK .....	4
4.1. Using 40 Pin Header SPI Port .....	4
4.2. Using FTDI USB-SPI Adapter Cable (Optional) .....	4
5. Build Driver .....	4
5.1. Build Silex SX-NEWAH Driver .....	4
5.2. Build Opensource NRC7292 Driver (Optional).....	6
5.3. Build USB-SPI Bridge Driver (Optional) .....	6
6. Load Driver .....	6
6.1. Load FTDI USB-SPI Bride Driver (Optional) .....	6
6.2. Load Silex SX-NEWAH or Opensource NRC7292 Driver .....	7
Appendix A. References .....	7
A.1. nVidia Getting Started Guide .....	7
A.2. FTDI SPI app note for SX-NEWAH.....	7
A.3. Silex SX-NEWAH Driver Package .....	7
A.4. Opensource NRC7292 Software Package .....	7
A.5. FTDI USB-SPI Bridge Driver.....	7
A.6. SX-NEWAH-EVK Start-UP Guide .....	7
Appendix B. Linux Device Tree Overlay DTS File: sx-newah-evk.dts.....	7
Appendix C. Opensource NRC7292 Driver Patch: nrc7292.patch .....	10

## 1. Scope

The purpose of this document is to provide instructions for setting up evaluation platform of SX-NEWAH-EVK with SPI port of nVidia Jetson. It also explains USB port connection with USB to SPI adapter (optional.)

## 2. Equipment

### 2.1. Hardware

- SX-NEWAH-EVK
- nVidia Jetson Nano 2GB Developer Kit
- Micro SD card (32GB minimum, 64GB or more recommended)
- USB Type C to USB Type A cable
- Micro USB Type B to USB Type A cable
- Mini USB Type A to USB Type A cable
- Ethernet RJ45 cable (for the internet access.)
- Male to female jumper cable x6
- USB 3.1 5V/3A power supply (optional, USB 3.0 port from PC could supply ample power.)

### 2.2. Software

- nVidia JetPack 4.5 (L4T32.5, Linux 4.9)
- Silex SX-NEWAH Driver Package v1.2.26
- Opensource NRC7292 Software Package v1.3.2 (optional)

### 2.3. Additional Hardware and Software for USB-SPI Bridged Connection (Optional)

- FTDI USB-SPI adapter cable C232HM-DDHSL-0
- FTDI USB-SPI Bridge Driver

## 3. Setup nVidia Jetson Nano 2GB Developer Kit

Follow nVidia Getting Started Guide (Section A.1) to program SD card image and initialize Jetson Nano 2GB module. To enable SPI1 port, save DTS file *sx-newah-evk.dts* shown in Appendix B in */boot* directory and compile SX-NEWAH-EVK Linux Device Tree overlay.

```
$ sudo -s
# cd /boot
# dtc -O dtb -go sx-newah-evk.dtbo sx-newah-evk.dts
```

Next, update DTB file with the overlay built above.

```
# /opt/nvidia/jetson-io/config-by-hardware.py -n SX-NEWAH-EVK
# reboot (or poweroff if you're mounting SX-NEWAH-EVK)
```

## 4. Mount SX-NEWAH-EVK

SX-NEWAH-EVK will be connected to SPI port with jumper wires. Alternatively, it can be connected to USB port with FTDI USB-SPI adapter cable. Either way, nVidia Jetson Nano Developers Kit needs to supply power to SX-NEWAH-EVK with Mini USB Type A to USB Type A cable for each connection method.

### 4.1. Using 40 Pin Header SPI Port

With jumper wire, connect SPI port of SX-NEWAH-EVK 40 pin header CN9 (NEWAH CN9) to Jetson Nano 2GB Developer Kit 40 pin header J6 (JN2DK J6) as shown in Table 1.

With Mini USB Type A to USB Type A cable, connect SX-NEWAH-EVK CN2 to Jetson Nano 2GB Developer Kit USB 2.0 port. This cable supplies a power to SX-NEWAH-EVK.

Table 1: SPI pinout

SPI Signal	NEWAH CN9 Pin	JN2DK J6 Pin
<b>MOSI</b>	19	19
<b>MISO</b>	21	21
<b>CLK</b>	23	23
<b>CS</b>	24	24
<b>IRQ</b>	29	29
<b>GND</b>	39	39

### 4.2. Using FTDI USB-SPI Adapter Cable (Optional)

See "FTDI SPI app note for SX-NEWAH" (Section A.2).

## 5. Build Driver

Build Silex SX-NEWAH Driver Package (Section A.3) and optionally FTDI USB-SPI Bridge Driver (Section A.5) with native compiler. This section needs the internet connection to execute *apt* and *git clone* command.

Opensource NRC7292 Software Package (Section A.4) can work with SX-NEWAH-EVK but it doesn't supply *hostapd* and *wpa\_supplicant* which can handle SAE authentication. SX-NEWAH Driver Package has tested and verified by Silex. Contact us for development support.

Boot Jetson Nano 2GB Developer Kit if you turned it off to mount SX-NEWAH-EVK.

### 5.1. Build Silex SX-NEWAH Driver

First, install library headers required to build *hostapd* and *wpa\_supplicant* command.

```
$ sudo apt install libnl-genl-3-dev libssl-dev
```

**Build kernel module.**

```
$ cd ~  
  
$ git clone https://customergts.silexamerica.com/sx-newah-package/sx-newah.git -b v1.2.26-sta  
  
$ cd ~/sx-newah/nrc_driver/source  
  
$ make
```

**Install board data file and target firmware file.**

```
$ cd ~/sx-newah/nrc_pkg/sw_pkg/nrc_pkg_1_2_0/sw/firmware  
  
$ sudo mkdir /lib/firmware/nrc  
  
$ sudo cp -i nrc7292_bd.dat nrc7292_cspi.bin /lib/firmware/nrc
```

**Build *wpa\_supplicant*.**

```
$ cd ~/sx-newah/nrc-hostapd-2.8/wpa_supplicant  
  
$ cp -i defconfig .config
```

Here, disable D-Bus control interface by removing (or comment out) following two lines from *.config* file.

```
CONFIG_CTRL_IFACE_DBUS_NEW=y  
CONFIG_CTRL_IFACE_DBUS_INTRO=y
```

Then, execute *make* command.

```
$ make clean; make
```

**Build *hostapd*.**

```
$ cd ~/sx-newah/nrc-hostapd-2.8/hostapd  
  
$ cp -i defconfig .config  
  
$ make clean; make
```

**Lastly, build CLI control application.**

```
$ cd ~/sx-newah/cli_app/source  
  
$ make
```

All the driver components and sample hostapd/wpa\_supplicant configuration files will be found in:

- ~/sx-newah/nrc\_driver/source/nrc.ko
- ~/sx-newah/nrc-hostapd-2.8/wpa\_supplicant/wpa\_supplicant
- ~/sx-newah/nrc-hostapd-2.8/wpa\_supplicant/wpa\_cli
- ~/sx-newah/nrc-hostapd-2.8/hostapd/hostapd
- ~/sx-newah/nrc-hostapd-2.8/hostapd/hostapd\_cli

- ~/sx-newah/cli\_app/source/cli\_app
- ~/sx-newah/nrc\_pkg/sw\_pkg/nrc\_pkg\_1\_2\_0/script/conf (config files)

## 5.2. Build Opensource NRC7292 Driver (Optional)

Alternatively, download NRC7292 Software Package.

```
$ cd ~
$ git clone https://github.com/newracom/nrc7292_sw_pkg.git -b v1.3.2
```

Apply nrc7292.patch shown in Appendix C.

```
$ cd ~/nrc7292_sw_pkg
$ patch -p1 < ~/nrc7292.patch
```

Build kernel module.

```
$ cd ~/nrc7292_sw_pkg/package/host/nrc_driver/source/nrc_driver/nrc
$ make
```

Install board data file and target firmware file.

```
$ cd ~/nrc7292_sw_pkg/package/host/evk/sw_pkg/nrc_pkg/sw/firmware
$ sudo mkdir /lib/firmware/nrc
$ sudo cp -i nrc7292_bd.dat nrc7292_cspi.bin /lib/firmware/nrc
```

Lastly, build CLI control application.

```
$ cd ~/nrc7292_sw_pkg/package/host/cli_app/source/cli_app
$ make
```

All the driver components sample hostapd/wpa\_supplicant configuration files will be found in:

- ~/nrc7292\_sw\_pkg/package/host/nrc\_driver/source/nrc\_driver/nrc/nrc.ko
- ~/nrc7292\_sw\_pkg/package/host/cli\_app/source/cli\_app/cli\_app
- ~/nrc7292\_sw\_pkg/package/host/evk/sw\_pkg/nrc\_pkg/script/conf (config files)

## 5.3. Build USB-SPI Bridge Driver (Optional)

See "FTDI SPI app note for SX-NEWAH" (Section A.2).

# 6. Load Driver

## 6.1. Load FTDI USB-SPI Bride Driver (Optional)

When Linux kernel detect USB-SPI adapter cable, it automatically loads *ftdi\_sio* module which conflict with FTDI USB-SPI Bridge Driver.

```
$ sudo -s (if you're not super user)
# modprobe -r ftdi_sio
```

```
# insmod ~/ftdi-usb-spi/spi-ft232h.ko
```

See "FTDI SPI app note for SX-NEWAH" (Section A.2) for additional information.

## 6.2. Load Silex SX-NEWAH or Opensource NRC7292 Driver

```
$ sudo -s (if you're not super user)
```

```
# modprobe mac80211
```

The module parameter must be adjusted according to the connection method selected.

```
# export OPTIONS="spi_gpio_irq=1 hifspeed=16000000" (SPI)
```

```
# export OPTIONS="spi_bus_num=32765 spi_gpio_irq=9050 hifspeed=5000000"
(FTDI USB-SPI Adapter)
```

Finally, load the driver.

```
# insmod nrc.ko bd_name=nrc/nrc7292_bd.dat fw_name=nrc/nrc7292_cspi.bin
$OPTIONS
```

```
# systemctl stop wpa_supplicant.service
```

At this stage, you can start *wpa\_supplicant* or *hostapd*. See Section 9 of SX-NEWAH-EVK Start-UP Guide (Section A.6).

## Appendix A. References

### A.1. nVidia Getting Started Guide

<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-2gb-devkit>

### A.2. FTDI SPI app note for SX-NEWAH

142-20159-140 FTDI SPI app note for SX-NEWAH

### A.3. Silex SX-NEWAH Driver Package

<https://customergets.silexamerica.com/sx-newah-package/sx-newah.git>

### A.4. Opensource NRC7292 Software Package

[https://github.com/newracom/nrc7292\\_sw\\_pkg](https://github.com/newracom/nrc7292_sw_pkg)

### A.5. FTDI USB-SPI Bridge Driver

<https://customergets.silexamerica.com/gpl/ftdi-usb-spi.git>

### A.6. SX-NEWAH-EVK Start-UP Guide

<https://www.silextechnology.com/productspecs/sx-newah/sx-newah-startup-guide-raspberry-pi?hsLang=en-us>

## Appendix B. Linux Device Tree Overlay DTS File: sx-newah-evk.dts

```
/dts-v1/;
/plugin/;
```



```

/ {
    overlay-name = "SX-NEWAH-EVK";
    compatible = "nvidia,p3542-0000+p3448-0003";

    fragment@0 {
        target = <0xffffffff>;

        __overlay__ {
            pinctrl-names = "default";
            pinctrl-0 = <0x1>;

            header-40pin-pinmux {
                linux,phandle = <0x1>;
                phandle = <0x1>;

                pin3 {
                    nvidia,function = "i2c2";
                    nvidia,pins = "gen2_i2c_sda_pj3";
                    nvidia,pull = <0x0>;
                    nvidia,tristate = <0x0>;
                    nvidia,enable-input = <0x1>;
                    nvidia,io-high-voltage = <0x1>;
                };

                pin5 {
                    nvidia,function = "i2c2";
                    nvidia,pins = "gen2_i2c_scl_pj2";
                    nvidia,pull = <0x0>;
                    nvidia,tristate = <0x0>;
                    nvidia,enable-input = <0x1>;
                    nvidia,io-high-voltage = <0x1>;
                };

                pin8 {
                    nvidia,function = "uartb";
                    nvidia,pins = "uart2_tx_pg0";
                    nvidia,pull = <0x0>;
                    nvidia,tristate = <0x0>;
                    nvidia,enable-input = <0x0>;
                };

                pin10 {
                    nvidia,function = "uartb";
                    nvidia,pins = "uart2_rx_pg1";
                    nvidia,pull = <0x2>;
                    nvidia,tristate = <0x1>;
                    nvidia,enable-input = <0x1>;
                };

                pin19 {
                    nvidia,function = "spi1";
                    nvidia,pins = "spi1_mosi_pc0";
                    nvidia,pull = <0x1>;
                    nvidia,tristate = <0x0>;
                    nvidia,enable-input = <0x1>;
                };

                pin21 {
                    nvidia,function = "spi1";
                    nvidia,pins = "spi1_miso_pc1";
                    nvidia,pull = <0x1>;
                    nvidia,tristate = <0x0>;
                    nvidia,enable-input = <0x1>;
                };

                pin23 {
                    nvidia,function = "spi1";
                    nvidia,pins = "spi1_sck_pc2";
                    nvidia,pull = <0x1>;

```

```

        nvidia,tristate = <0x0>;
        nvidia,enable-input = <0x1>;
    };

    pin24 {
        nvidia,function = "spi1";
        nvidia,pins = "spi1_cs0_pc3";
        nvidia,pull = <0x2>;
        nvidia,tristate = <0x0>;
        nvidia,enable-input = <0x1>;
    };

    pin26 {
        nvidia,function = "spi1";
        nvidia,pins = "spi1_cs1_pc4";
        nvidia,pull = <0x2>;
        nvidia,tristate = <0x0>;
        nvidia,enable-input = <0x1>;
    };

    pin27 {
        nvidia,function = "i2c1";
        nvidia,pins = "gen1_i2c_sda_pj0";
        nvidia,pull = <0x0>;
        nvidia,tristate = <0x0>;
        nvidia,enable-input = <0x1>;
        nvidia,io-high-voltage = <0x1>;
    };

    pin28 {
        nvidia,function = "i2c1";
        nvidia,pins = "gen1_i2c_scl_pj1";
        nvidia,pull = <0x0>;
        nvidia,tristate = <0x0>;
        nvidia,enable-input = <0x1>;
        nvidia,io-high-voltage = <0x1>;
    };
};

};

fragment@1 {
    target = <&hdr40_spi1>;
    __overlay__ {
        spi@0 {
            status = "disabled";
        };
        spi@1 {
            status = "disabled";
        };
    };
};

fragment@2 {
    target = <&hdr40_spi2>;
    __overlay__ {
        spi@0 {
            status = "disabled";
        };
        spi@1 {
            status = "disabled";
        };
    };
};

__symbols__ {
    hdr40_pinmux = "/fragment@0/__overlay__/header-40pin-pinmux";
};

```

```

__fixups__ {
    pinmux = "/fragment@0:target:0";
};

__local_fixups__ {
    fragment@0 {
        __overlay__ {
            pinctrl-0 = <0x0>;
        };
    };
};
};

```

## Appendix C. Opensource NRC7292 Driver Patch: nrc7292.patch

From 4a1fa4322a5c1c1feaba823895a6eb6e3d5afa70 Mon Sep 17 00:00:00 2001  
 From: Tom Nakase <nakase@silexamerica.com>  
 Date: Thu, 11 Mar 2021 14:51:09 -0800  
 Subject: [PATCH] JetPack 4.5 port

```

---
package/host/nrc_driver/source/nrc_driver/nrc/nrc-bd.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/package/host/nrc_driver/source/nrc_driver/nrc/nrc-bd.c
b/package/host/nrc_driver/source/nrc_driver/nrc/nrc-bd.c
index c02bbf1..d7f63d9 100755
--- a/package/host/nrc_driver/source/nrc_driver/nrc/nrc-bd.c
+++ b/package/host/nrc_driver/source/nrc_driver/nrc/nrc-bd.c
@@ -91,7 +91,7 @@ static void * nrc_dump_load(int len)
    #if KERNEL_VERSION(4, 14, 0) <= NRC_TARGET_KERNEL_VERSION
        ret = kernel_read(filp, g_bd_buf, length, &pos);
    #else
-       ret = kernel_read(filp, g_bd_buf, length, pos);
+       ret = kernel_read(filp, pos, g_bd_buf, length);
    #endif

    filp_close(filp, NULL);
--
2.17.1

```