Rate This Article :

EVB-KSZ9477: How to Use the Linux DSA Driver

Feb 20, 2025 Knowledge

Article Number
000014935

Title
EVB-KSZ9477: How to Use the Linux DSA Driver

Article URL
https://microchip.my.site.com/s/article/EVB-KSZ9477--How-to-Use-the-Linux-DSA-Driver (https://microchip.my.site.com/s/article/EVB-KSZ9477--How-to-Use-the-Linux-DSA-Driver)

Question
How does one use the Ethernet DSA (Distributed Switch Architecture) Drivers using the EVB-KSZ9477?

Answer
One method of using the Ethernet DSA drivers with the EVB-KSZ9477 is described below. The steps shown build a v5.15 Linux image, which was tested on an Ubuntu 22.04 host.

The steps below may be adapted as needed if you're using a SAMA5D3-EDS with a different switch board (EVB-KSZ9563, EVB-KSZ8563, etc.).

How to build a kernel v5.15 image with DSA drivers for the EVB-KSZ9477 using buildroot-at91

Step 1: Setup the Buildroot environment using the commands below. For convenience, you may opt to use the commands below and paste them in a shell script named "setup.sh (http://setup.sh)"

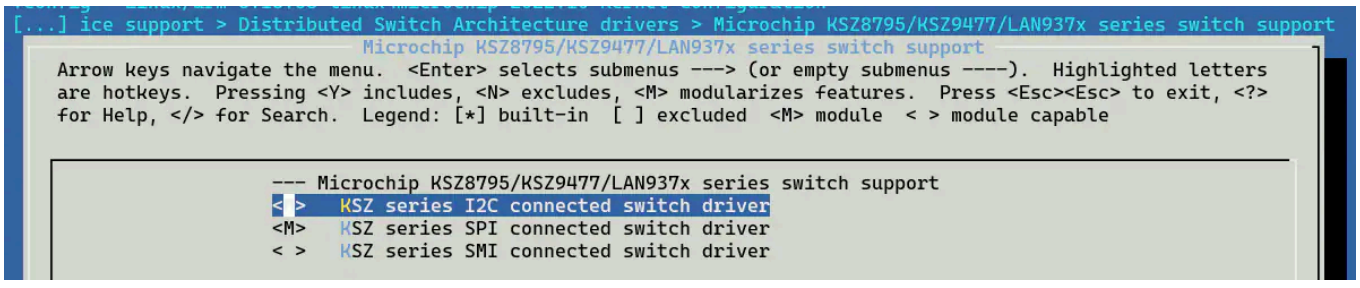
```
#!/bin/bash

# Setup buildroot env.

# NOTE: Run the script using 'source ./setup.sh (https://setup.sh/)'

sudo apt-get install sed make binutils build-essential gcc g++ bash patch gzip bzip2 perl tar cpio
unzip rsync file bc git python3 python libncurses5-dev libssl-dev -y


git clone https://github.com/linux4sam/buildroot-at91 (https://github.com/linux4sam/buildroot-at91)

git clone https://github.com/linux4sam/buildroot-external-microchip
(https://github.com/linux4sam/buildroot-external-microchip)


cd buildroot-external-microchip && git checkout linux4microchip-2022.10 && cd ..

cd buildroot-at91 && git checkout linux4sam-2022.10 && cd ..


# NOTE: We don't need to use O=$PWD with make because we're not going to build out-of-tree

cd buildroot-at91/
make BR2_EXTERNAL=../buildroot-external-microchip -C $PWD sama5d3_xplained_headless_defconfig
menuconfig
```

Notes:

- Run the script using "source ./setup.sh (https://setup.sh/)"
- Running menuconfig is optional, so you may opt to remove it from the last command shown above.
- For a SAMA5D3-EDS, use the sama5d3_eds_headless_defconfig (or sama5d3_eds_nf_defconfig if using NAND Flash) instead.

Step 2: After loading **`sama5d3_xplained_headless_defconfig`** to your .config, run **`make linux-menuconfig`** to configure the kernel. This will prompt buildroot to download the Linux kernel source and perform some build steps, which may take a few minutes.

Once linux-menuconfig opens, navigate to Device Drivers -> Network device support -> Distributed Switch Architecture drivers -> Microchip KSZ8795/KSZ9477/LAN937x series switch support, to find the DSA driver options. Use spacebar key to select the I2C driver as a module.



Trending Articles

Harmony - Getting Started and Example Projects (/s/article/Harmony-Getting-Started-and-Example-Projects)

Tutorials for PIC32MZ and Harmony (/s/article/Tutorials-for-PIC32MZ-and-Harmony)

Optional step:

It may be worth modifying buildroot-external-microchip/board/microchip/sama5d3_xplained/uboot-env.txt and adding environment variables for ipaddr and ethaddr. This ensures that the default MAC and IP addresses are configured properly. Otherwise, you may opt to add these variables through U-Boot directly.

Step 3: Run 'make'.

Step 4: After make succeeds, you will need to replace the device tree source file.

Note: You may use make menuconfig and navigate to Kernel to find out the file name of the device tree source.

```
    [*]   Build a Device Tree Blob (DTB)
    [ ]     DTB is built by kernel itself
    (at91-sama5d3_xplained) In-tree Device Tree Source file names
    ()      Out-of-tree Device Tree Source file paths
```

Navigate to the device tree source file's location by running:

```
# cd <buildroot-at91 location>/output/build/linux-custom/arch/arm/boot/dts/
```

Rename the existing dts file:

```
# mv at91-sama5d3_xplained.dts at91-sama5d3_xplained.dts.orig
```

Then, download the example dts file from https://github.com/Microchip-Ethernet/EVB-KSZ9477/blob/master/KSZ/linux-drivers/ksz9897/linux-5.15/arch/arm/boot/dts/at91-sama5d3_xplained_ung8071.dts (https://github.com/Microchip-Ethernet/EVB-KSZ9477/blob/master/KSZ/linux-drivers/ksz9897/linux-5.15/arch/arm/boot/dts/at91-sama5d3_xplained_ung8071.dts) into the directory:

```
# wget -O at91-sama5d3_xplained.dts raw.githubusercontent.com/Microchip-Ethernet/EVB-KSZ9477/master/KSZ/linux-drivers/ksz9897/linux-5.15/arch/arm/boot/dts/at91-sama5d3_xplained_ung8071.dts (http://raw.githubusercontent.com/Microchip-Ethernet/EVB-KSZ9477/master/KSZ/linux-drivers/ksz9897/linux-5.15/arch/arm/boot/dts/at91-sama5d3_xplained_ung8071.dts)
```

Note: If you would like to use SPI as the management interface instead, you may skip modifying the device tree as described below.

Once the download is done, open at91-sama5d3_xplained.dts using a text editor like gedit or VS Code. Find the SPI node and disable the KSZ9477. This ensures that the I2C DSA driver is loaded on boot.

```
spi1: spi@f8008000 {
    cs-gpios = <&pioC 25 GPIO_ACTIVE_LOW>, <0>, <0>,
    status = "okay";
    id = <1>;

    ksz9477: ksz9477@0 {
        compatible = "microchip,ksz9477";
        reg = <0>;

        /* Bus clock is 132 MHz. */
        spi-max-frequency = <44000000>;
        spi-cpha;
        spi-cpol;
        interrupt-parent = <&pioB>;
        interrupts = <28 IRQ_TYPE_LEVEL_LOW>;
        status = "disabled";
        ports {
            #address-cells = <1>;
```

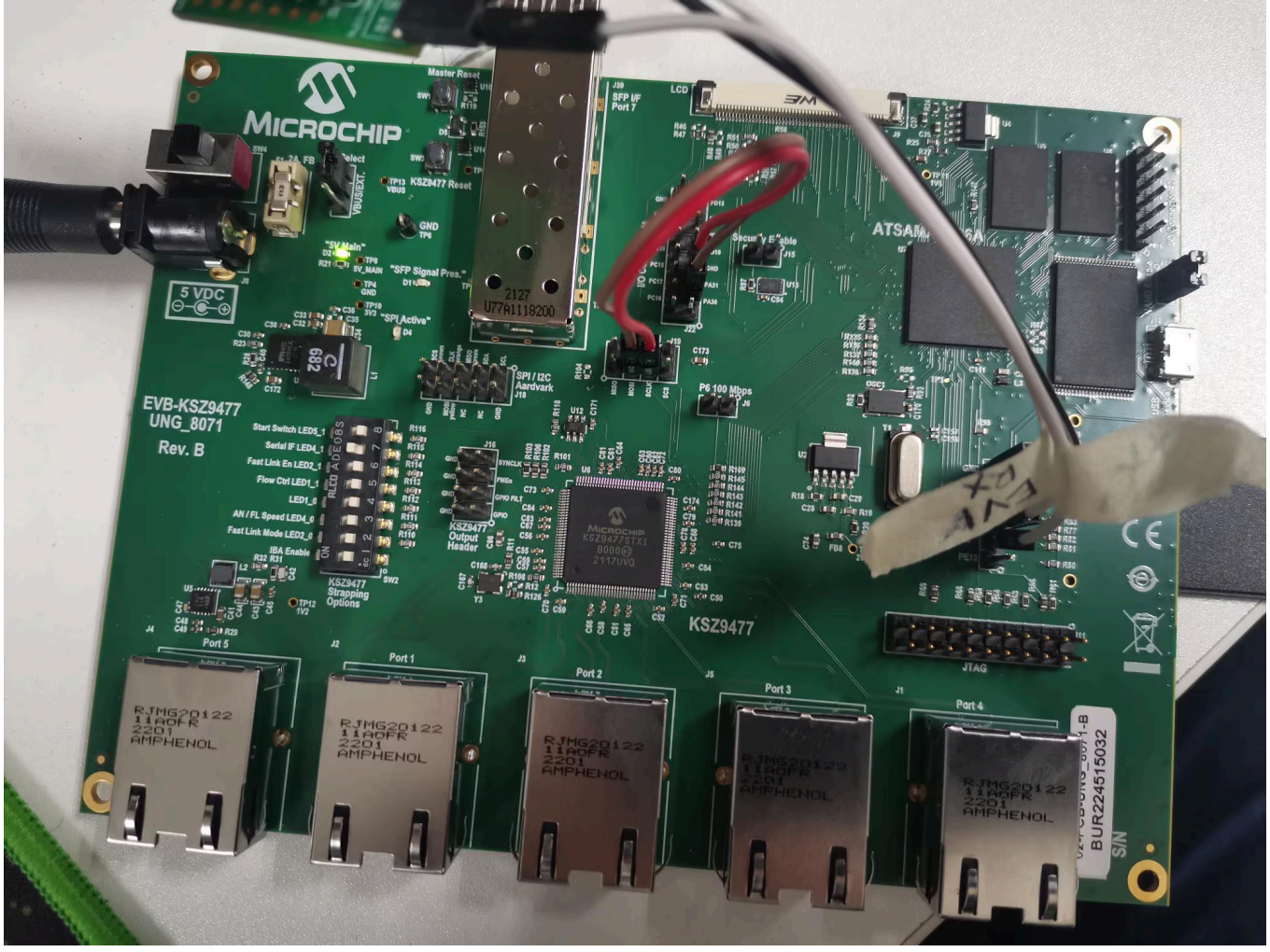
Note: If you're using a SAMA5D3-EDS, you can refer to the example Device tree overlays in https://github.com/linux4microchip/dt-overlay-mchp/tree/master/sama5d3_eds (https://github.com/linux4microchip/dt-overlay-mchp/tree/master/sama5d3_eds) . This requires knowledge of device tree overlays which is not covered in this article.

Step 5: Navigate back to the buildroot-at91 directory and rebuild the Linux image by running the ff commands:

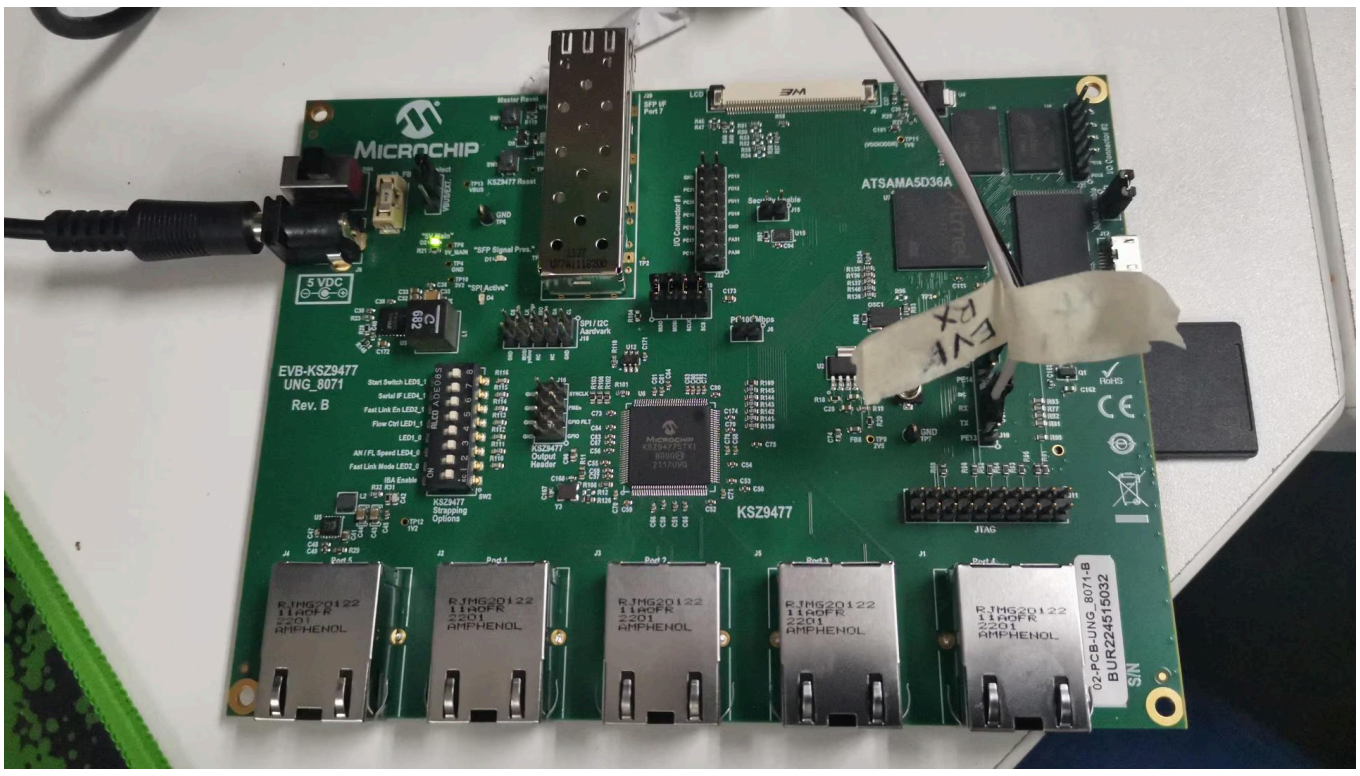
```
# make linux-rebuild && make dt-overlay-mchp-rebuild && make
```

Step 6: Set up the EVB-KSZ9477 board to use the correct management interface.

For I2C access, remove the jumper caps on J19. Then, using regular jumper wires, connect PA31 and PA30 from J22 to SCLK and MOSI, respectively. Also, ensure the Serial IF LED4_1 strap is set to ON.



For SPI access, simply leave all the jumper caps on J19 connected, then set the LED4_1 strap to OFF.



Step 7: Obtain sdcard.img from the output/ folder and flash it to an SD Card using a tool like balenaEtcher. Then plug it into the EVB_KSZ9477. You should see the following logs if you're using I2C:

```
ksz9477-switch 0-005f: found switch: KSZ9477, rev 0
ksz9477-switch 0-005f: Port 5 interpreting RGMII delay settings based on "phy-mode" property, please update device tree to specify "rx-internal-delay-ps" and "tx-internal-delay-ps"
ksz9477-switch 0-005f: Port5: using phy mode rgmii-txid
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 0
ksz9477-switch 0-005f lan1 (uninitialized): PHY [dsa-0.0:00] driver [Microchip KSZ9477] (irq=POLL)
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 1
ksz9477-switch 0-005f lan2 (uninitialized): PHY [dsa-0.0:01] driver [Microchip KSZ9477] (irq=POLL)
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 2
ksz9477-switch 0-005f lan3 (uninitialized): PHY [dsa-0.0:02] driver [Microchip KSZ9477] (irq=POLL)
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 3
ksz9477-switch 0-005f lan4 (uninitialized): PHY [dsa-0.0:03] driver [Microchip KSZ9477] (irq=POLL)
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 4
ksz9477-switch 0-005f lan5 (uninitialized): PHY [dsa-0.0:04] driver [Microchip KSZ9477] (irq=POLL)
ksz9477-switch 0-005f: configuring for fixed/rgmii-txid link mode
ksz9477-switch 0-005f: nonfatal error -34 setting MTU to 1500 on port 6
ksz9477-switch 0-005f lan6 (uninitialized): PHY [dsa-0.0:06] driver [Microchip KSZ9477] (irq=POLL)
DSA: tree 0 setup
ksz9477-switch 0-005f: Link is Up - 1Gbps/Full - flow control rx/tx
```

```
# lsmod
Module                  Size  Used by    Not tainted
ksz_spi                16384  0
tag_ksz                12288  1
ksz9477_i2c            12288  0
ksz_switch             61440  2 ksz_spi,ksz9477_i2c
dsa_core               61440 10 ksz_spi,tag_ksz,ksz9477_i2c,ksz_switch
bridge                176128  1 dsa_core
stp                    12288  1 bridge
llc                    12288  2 bridge,stp
atmel_usba_udc         20480  0
#
```

For SPI, the logs should be similar to the screenshot shown below.

```
atmel_usba_udc 500000.gadget: FIFO at [mem 0x00500000-0x005fffff] mapped at (ptrval)
ksz-switch spi1.0: found switch: KSZ9477, rev 0
ksz-switch spi1.0: Port 5 interpreting RGMII delay settings based on "phy-mode" property, please update device tree to specify "rx-internal-delay-ps" and "tx-internal-delay-ps"
ksz-switch spi1.0: Port5: using phy mode rgmii-txid
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 0
ksz-switch spi1.0 lan1 (uninitialized): PHY [dsa-0.0:00] driver [Microchip KSZ9477] (irq=POLL)
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 1
ksz-switch spi1.0 lan2 (uninitialized): PHY [dsa-0.0:01] driver [Microchip KSZ9477] (irq=POLL)
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 2
ksz-switch spi1.0 lan3 (uninitialized): PHY [dsa-0.0:02] driver [Microchip KSZ9477] (irq=POLL)
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 3
ksz-switch spi1.0 lan4 (uninitialized): PHY [dsa-0.0:03] driver [Microchip KSZ9477] (irq=POLL)
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 4
ksz-switch spi1.0 lan5 (uninitialized): PHY [dsa-0.0:04] driver [Microchip KSZ9477] (irq=POLL)
ksz-switch spi1.0: configuring for fixed/rgmii-txid link mode
ksz-switch spi1.0: nonfatal error -34 setting MTU to 1500 on port 6
ksz-switch spi1.0 lan6 (uninitialized): PHY [dsa-0.0:06] driver [Microchip KSZ9477] (irq=POLL)
DSA: tree 0 setup
ksz-switch spi1.0: Link is Up - 1Gbps/Full - flow control rx/tx
done
```

```
# lsmod
Module                  Size  Used by    Not tainted
tag_ksz                12288  1
ksz_spi                16384  0
ksz_switch             61440  1 ksz_spi
dsa_core               61440  9 tag_ksz,ksz_spi,ksz_switch
bridge                176128  1 dsa_core
stp                    12288  1 bridge
llc                    12288  2 bridge,stp
atmel_usba_udc         20480  0
```

For more information on how to use the DSA Drivers, refer to AN3761 - KSZ DSA Driver Utilization (microchip.com) (https://ww1.microchip.com/downloads/en/Appnotes/AN3761-KSZ-DSA-Driver-Utilization-00003761A.pdf).

URL Name

EVB-KSZ9477--How-to-Use-the-Linux-DSA-Driver

Devices

KSZ9477

©Copyright 1998-2025 Microchip Technology Inc. All rights reserved.