Rate This Article :

Microchip KSZ/LAN Switches register reading via Debugfs Regmap

Oct 26, 2023 Knowledge

Article Number
000015066

Title
Microchip KSZ/LAN Switches register reading via Debugfs Regmap

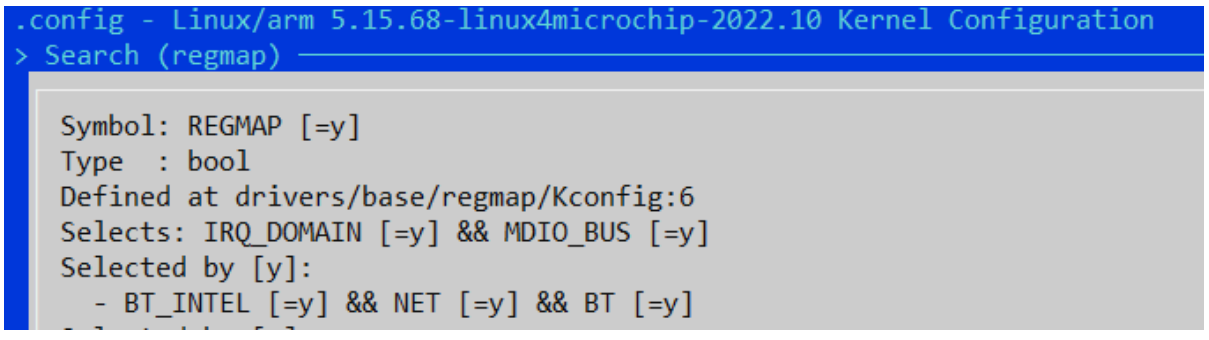
Article URL
https://microchip.my.site.com/s/article/Microchip-KSZ-LAN-Switches-register-reading-via-Debugfs-Regmap (https://microchip.my.site.com/s/article/Microchip-KSZ-LAN-Switches-register-reading-via-Debugfs-Regmap)

Question
How to perform register reading when using KSZ/LAN DSA Driver from Linux Mainline Kernel?

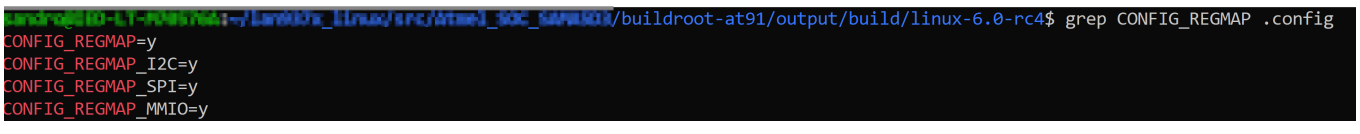
Answer
Microchip Software Tools for Register Read/Write (e.g. regs_bin and mdio-tool) are not available in the Mainline Linux DSA Driver. An alternative way to perform Register Reading for KSZ Switches and PHYs is by using Debugfs Regmap. The steps utilized for a buildroot based distribution are described below:

1) Make sure REGMAP is enabled in your kernel configuration:

In your buildroot directory, run linux-menuconfig, press "/" and search for REGMAP. Make sure it is marked as "y". Check also for any dependencies as this can change from one version to another.



Another option is to navigate to /buildroot/output/build/linux-xxxxxx and run ***$ grep CONFIG_REGMAP .config***



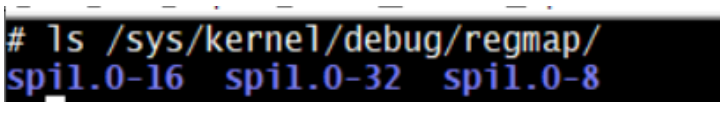
Check also if your desired management option (I2C, SPI or MDIO) is enabled, as shown in the image above.

2) After enabling REGMAPS, finalize building and deploying the image into the target device.

3) In the target device, use ***$ mount -t debugfs none /sys/kernel/debug*** to mount the file system used for debugging the registers.

4) Navigate to /sys/kernel/debug/regmap/ and check the subfolders created. These will correspond to the management interface being used (I2C or SPI) as wells as the SPI instance number or I2C address and each one contains the values of the registers arranged in different sizes (8 bits, 16 bits and 32 bits).

The figure below shows the directory created for a switch connected via spi1.0 interface.



Example 1: Switch connected via SPI interface. 8 bit reading on registers 0001 and 0002 (Chip ID):

 # cat /sys/kernel/debug/regmap/spi1.0-8/registers | grep '0001\|0002'

0001: 94

0002: 77

Example 2: Switch connected via I2C interface. 16 bit reading on register 0001 (As this is a 8 bit register and we are reading 16 bits, it will return both register 0001 and 0002):

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)

*$ cat /sys/kernel/debug/regmap/0-005f-**16**/registers | grep 0001:*

0001: 9477

Note1: Modify the command line shown in the examples to match the contents of */sys/kernel/debug/regmap/* directory as this is specific for the type/instance of the interface being used.

Note1: Change between XXXX-**8**, XXXX-**16** or XXXX-**32** for 8, 16 or 32 bits reading.

Note2: Remove the ***grep xxxx*** statement for a complete dump of all registers.

URL Name

Microchip-KSZ-LAN-Switches-register-reading-via-Debugfs-Regmap

Devices

KSZ9477, KSZ9897, KSZ9567, LAN9370