DFSDM Linux driver

Contents

Stable: 16.01.2020 - 10:11 / Revision: 16.01.2020 - 10:00

A quality version of this page, approved on *16 January 2020*, was based off this revision.

Contents

1 Article purpose

The purpose of this article is to introduce the Linux® driver for the DFSDM internal peripheral:

- Which DFSDM features are supported by the driver
- How to configure, use and debug it
- What is the driver structure, where to the find source code.

2 Short Description

The DFSDM Linux® driver (kernel space) is based on the IIO and ALSA frameworks. It offers various modes:

1. **IIO direct mode**: single capture on a channel
2. **IIO software buffer**: capture one or more channels
3. **IIO triggered buffer mode**: capture on one or more channels using triggers

It uses hardware triggers available in IIO. See TIM Linux driver and LPTIM Linux driver.

It offers an extended API[1] for audio usage which allows **PDM microphones capture** through a SPI interface.

3 Configuration

3.1 Kernel configuration

Activate the DFSDM Linux® driver in the kernel configuration by using the menuconfig tool.

3.1.1 IIO driver

Configuration flag: CONFIG_STM32_DFSDM_ADC

```
Device Drivers  --->
   <*> Industrial I/O support  --->
      Analog to digital converters  --->
         <*> STMicroelectronics STM32 dfsdm adc
         <*>   STMicroelectronics STM32 adc                    # DFSDM ADC IIO driver
```

When using the DFSDM ADC IIO driver (only), the user must also enable the driver for the external analog front-end (e.g. sigma delta modulator). The generic sigma delta modulators driver may be used for instance (CONFIG_SD_ADC _MODULATOR):

```
Device Drivers  --->
   <*> Industrial I/O support  --->
      Analog to digital converters  --->
         <*> Generic sigma delta modulator                     # sigma delta
modulator driver
```

3.1.2 Audio driver

Configuration flag: CONFIG_SND_SOC_STM32_DFSDM (Note: This configuration depends on CONFIG_STM32_DFSDM_ADC)

```
Device Drivers  --->
   <*> Sound card support  --->
      <*>   Advanced Linux Sound Architecture  --->
         <*>   ALSA for SoC audio support  --->
            STMicroelectronics STM32 SOC audio support  --->
               <*> SoC Audio support for STM32 DFSDM           # DFSDM Audio driver
for digital microphone capture
```

3.2 Device tree

Refer to the DFSDM device tree configuration article when configuring the DFSDM Linux kernel driver.

4 How to use

4.1 IIO driver

In "**IIO direct mode**", the conversion result can be read directly from **sysfs**, see: How to do a simple ADC conversion using the sysfs interface.

In "**IIO triggered buffer mode**", the configuration must be performed by using **sysfs** first. Then, **character device** (/dev/iio: deviceX) is used to read data, see Convert one or more channels using triggered buffer mode.

For information on the standard IIO consumer interface, please refer to How to use IIO kernel API which gives an example of the IIO consumer kernel API.

4.2 Audio driver

The DFSDM Linux driver can be accessed from userland through an ALSA device. Refer to ALSA overview for information on how to list and use ALSA devices.

5 How to trace and debug

5.1 How to monitor

The DFSDM driver uses resources such as clocks and GPIOs.

- Refer to Pinctrl_overview#How_to_monitor to check DFSDM GPIOs.
- Refer to Clock_overview#How_to_monitor_with_debugfs to check DFSDM clocks.

5.1.1 How to monitor with debugfs

The DFSDM registers are accessed using REGMAP by *DFSDM Linux® driver*.

It comes with debugfs entries to dump registers:

```
$ cd /sys/kernel/debug/regmap/4400d000.dfsdm/
$ cat registers
000: 00000000
004: 00000000
008: 00000000
```

5.1.2 Other ways to monitor

- Man can check the DFSDM **interrupts** and/or the DFSDM **DMA** interrupts:

```
$ cat /proc/interrupts
           CPU0       CPU1
...
 99:          0          0     GIC-0 142 Level     4400d000.dfsdm:filter@0
100:          0          0     GIC-0 143 Level     4400d000.dfsdm:filter@1
101:          0          0     GIC-0 144 Level     4400d000.dfsdm:filter@2
...
```

5.2 How to trace

5.2.1 IIO driver

Refer to How to trace with dynamic debug for how to enable the debug logs in the driver and in the framework.

```
Board $> dmesg -n8
Board $> echo "file drivers/iio/adc/stm32-dfsdm* +p" > /sys/kernel/debug/dynamic_debug
/control
```

To enable dynamic debug at boot time, append the following arguments on the kernel command line:

```
loglevel=8 dyndbg="file drivers/iio/adc/stm32-dfsdm* +p"
```

5.2.2 Audio driver

Refer to ALSA_overview#How_to_trace for details on trace tools.

5.3 How to debug

5.3.1 Audio driver

Refer to ALSA_overview#How_to_debug for details on debugging tools.

6 Source code location

It is composed of:

- stm32-dfsdm-core.c , core part of DFSDM Linux driver to handle common resources: registers, clock
- stm32-dfsdm-adc.c , ADC part of DFSDM Linux driver to handle **ADC** operations
- stm32_adfsdm.c , ASoC DAI part of DFSDM Linux driver to handle **audio** operations

See also sigma delta modulator driver:

- sd_adc_modulator.c , sigma delta modulator Linux driver to handle analog front-end

7 References

- include/linux/iio/adc/stm32-dfsdm-adc.h , DFSDM IIO custom API

Linux® is a registered trademark of Linus Torvalds.

Digital Filter for Sigma-Delta Modulator

Industrial I/O Linux® subsystem

Application programming interface

Serial Peripheral Interface

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

Secure digital

Advanced Linux sound architecture

System File System (See https://en.wikipedia.org/wiki/Sysfs for more details)

Debug File System (See https://en.wikipedia.org/wiki/Debugfs for more details)

Register map (Linux® registers map abstraction API)

Direct Memory Access

Generic Interrupt Controller

ALSA System on Chip

Digital Audio Interface