



TIM Linux driver



# TIM Linux driver

Stable: 16.01.2020 - 15:02 / Revision: 16.01.2020 - 14:58

## Contents

1 Article purpose .....	2
2 Short description .....	2
3 Configuration .....	3
<b>3.1 Kernel configuration .....</b>	<b>3</b>
<b>3.2 Device tree .....</b>	<b>3</b>
4 How to use .....	3
5 How to trace and debug .....	3
6 Source code location .....	4
7 References .....	4

## 1 Article purpose

This article introduces the TIM Linux<sup>®</sup> driver for the TIM internal peripheral<sup>[1]</sup>:

- Which TIM features are supported by the driver
- How to configure, use and debug the driver
- What is the driver structure, and where the source code can be found.

## 2 Short description

The TIM<sup>[1]</sup> Linux driver (kernel space) is based on the PWM and IIO frameworks. It provides several functionalities:

### MFD driver:

- handles registers, clock and DMA<sup>[2]</sup> resources
- detects the TIM counter resolution, e.g. 16 or 32 bits.

### PWM driver:

- detects the number of TIM channels.
- handles **PWM output** channels.
- handles **PWM capture** channels (input). Note that the PWM capture relies on DMA, which is handled by the MFD core.

### IIO driver:

- handles hardware **trigger sources** (synchronously with PWM) for other internal peripherals such as ADC<sup>[3]</sup>, DAC<sup>[4]</sup>, DFSDM<sup>[5]</sup>.
- handles the **quadrature encoder** interface<sup>[6]</sup>.



## 3 Configuration

### 3.1 Kernel configuration

Activate the TIM<sup>[1]</sup> Linux driver in the kernel configuration using the Linux Menuconfig tool: [Menuconfig or how to configure kernel](#).

Enable the following configurations (and their dependencies):

- CONFIG\_MFD\_STM32\_TIMERS
- CONFIG\_PWM\_STM32
- CONFIG\_IIO\_STM32\_TIMER\_TRIGGER

```
Device Drivers --->
-> Multifunction device drivers --->
  <*> Support for STM32 Timers
-> Pulse-width modulation (PWM) support --->
  <*> STMicroelectronics STM32 PWM
-> Industrial I/O support --->
  -> Triggers - standalone --->
    <*> STM32 timer trigger
```

### 3.2 Device tree

Refer to the [TIM device tree configuration](#) article when configuring the TIM Linux kernel driver.

## 4 How to use

[How to use PWM with sysfs interface](#)

[How to set up a TIM or LPTIM trigger using the sysfs interface](#)

[How to use the quadrature encoder with the sysfs interface](#)

## 5 How to trace and debug

The TIM<sup>[1]</sup> Linux driver can access the timer registers through REGMAP.

It comes with debugfs<sup>[7]</sup> entries, which allow dumping registers:

```
$ cd /sys/kernel/debug/regmap
$ ls
40004000.timer 44000000.timer

$ cd 44000000.timer
$ cat registers
```



## TIM Linux driver

```
000: 00000081
004: 00000000
008: 00000000
00c: 00000000
...
```

It also comes with tracepoints<sup>[8]</sup>:

```
$ cd /sys/kernel/debug/tracing
$ cat available_events | grep regmap
...
regmap:regmap_reg_read
regmap:regmap_reg_write
```

## 6 Source code location

The TIM Linux driver source code is composed of:

- [stm32-timers.c](#) MFD driver to handle common resources: registers, clock, dmas.
- [pwm-stm32.c](#) PWM driver to handle PWM channel(s).
- [stm32-timer-trigger.c](#) IIO driver to handle trigger source for other internal peripherals and quadrature encoder interface.
- [include/linux/mfd/stm32-timers.h](#) and [include/linux/iio/timer/stm32-timer-trigger.h](#) header files

## 7 References

- [1.01.11.21.3](#) TIM internal peripheral
- [DMA\\_internal\\_peripheral](#)
- [ADC internal peripheral](#)
- [DAC internal peripheral](#)
- [DFSDM internal peripheral](#)
- [Incremental encoder](#) Incremental encoder overview
- [Debugfs](#)
- [Ftrace](#)