TIMER1 Drivers

For

AVR Microcontrollers

Nti Team

Software Requirement Specification Document

Table of Contents

Version: 1st **Date: 26/10/2023**

1. Scope of Document

This document specifies requirements on the module Timer1 Driver.

1.1 Constraints

First scope for specification of requirements on basic software modules is systems which are not safety relevant. For this reason safety requirements are assigned to medium priority.

2. Requirements Structure

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

Functional Requirements: -

Types of Timer1

Programme of timer1

Interface.

Registers of Timer1.

Non-Functional Requirements:-

Timing Requirements.

Resource Usage.

Usability.

3. Functional Overview

3.1 TIMER1 Driver, common functionality:

Generally, we use a timer/counter to generate time delays, waveforms, or to count events. Also, the timer is used for PWM generation, capturing events, etc. The timer module provides an event interface to the hardware timers.
The module uses the 16-bit timer 1 of the ATmega32. The timer is dynamically configured as needed by the registered alarms and should always be clocked as slow as possible to keep the interrupt load low. When no alarms are registered, the timer clock is disabled, and timer1 modes include:

- Normal Modes.
- Output Compare A
- Output Compare B
- Fast PWM.
- Phase PWM.
- Input Capture unit.

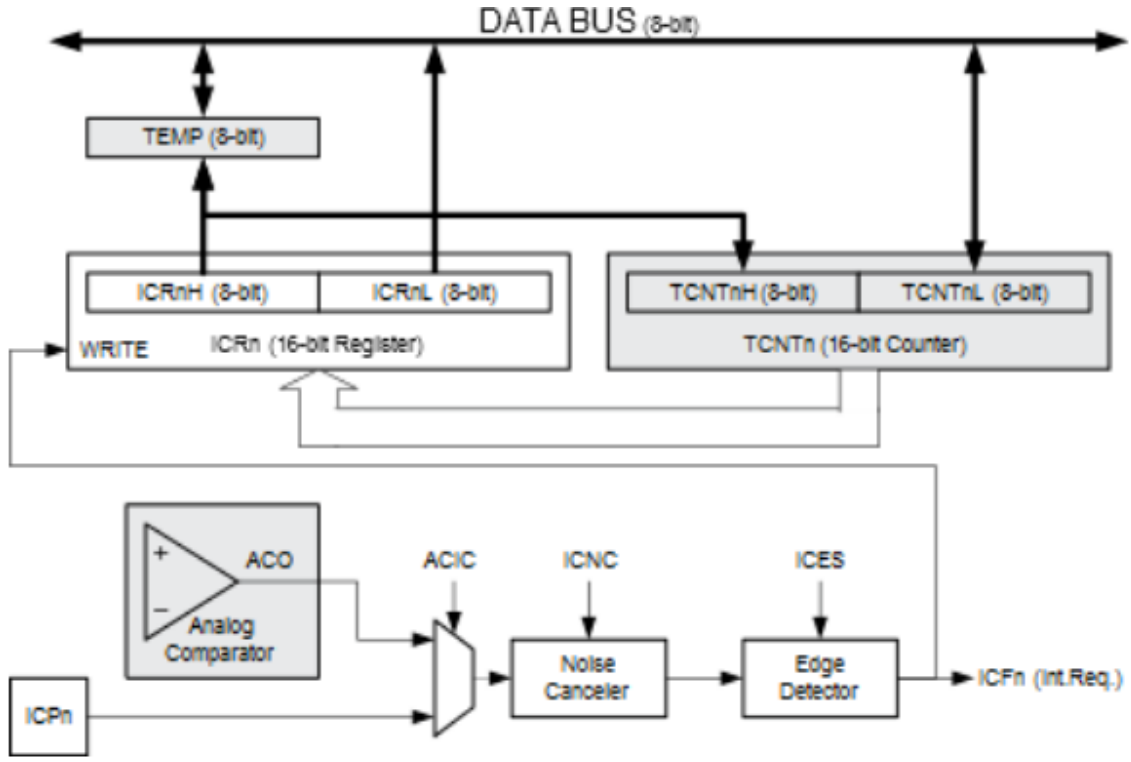
3.2 Input Capture Mode:

The input capture function is used in many applications such as:

- Pulse width measurement
- Period measurement
- Capturing the time of an event

The Timer/Counter1 incorporates an Input Capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP1 pin or alternatively, via the analog-comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram below. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded. The lower case “n” in register and bit names indicates the Timer/Counter number.



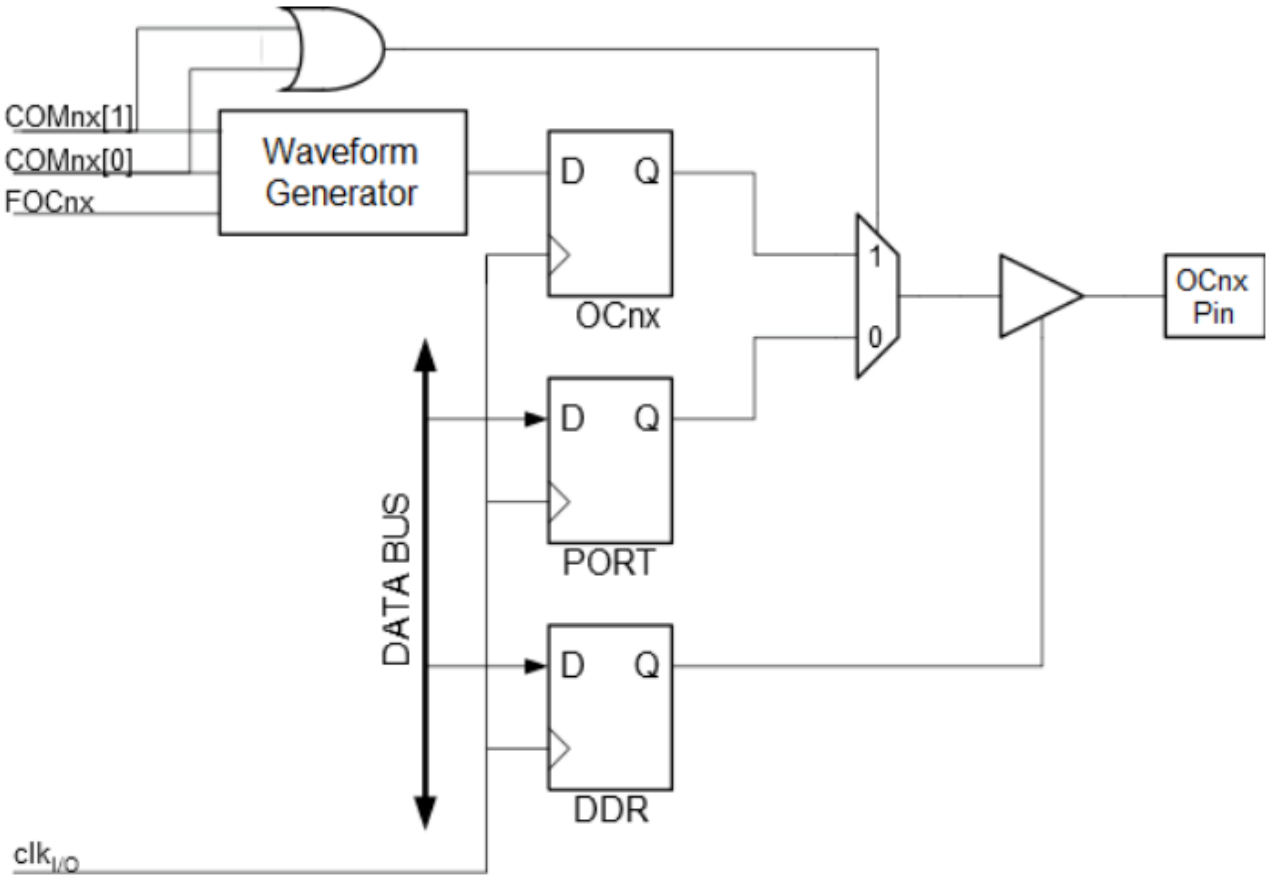
3.3 Compare Match Output Unit

The Compare Output mode bits in the Timer/Counter Control Register A (TCCR1A.COM1x) have two functions:

- The Waveform Generator uses the COM1x bits for defining the Output Compare (OC1x) register state at the next compare match.
- The COM1x bits control the OC1x pin output source

The figure below shows a simplified schematic of the logic affected by COM1x. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers that are affected by the COM1x bits are shown, namely PORT and DDR.

On system reset the OC1x Register is reset to 0x00.



4. Requirement Specification

4.1 Functional Requirements

[SRS_REQ_1] The TIMER1 Driver shall support a function to initialise the timer.

[SRS_REQ_2] The driver shall be compatible with all AVR microcontrollers.

[SRS_REQ_3] The TIMER1 Driver shall support a specific basic static configuration.

[SRS_REQ_4] The TIMER1 Handler/Driver shall handle the chip

[SRS_REQ_5] The TIMER1 Driver shall support functions that control the interrupt to make it enable or disable.

[SRS_REQ_6] The TIMER1 Driver Support function that make the timer work with normal mode.

[SRS_REQ_7] The TIMER1 Driver Support function that make the timer work with compare A.

[SRS_REQ_8] The TIMER1 Driver Support function that make the timer work with compare B.

[SRS_REQ_9] The TIMER1 Driver Support function that make the timer work with FAST PWM.

[SRS_REQ_10] The TIMER1 Driver Support function that make the timer work with PHASE CORRECT.

[SRS_REQ_11] The TIMER1 Driver Support function that make the timer work with INPUT CAPTURE UNIT.

[SRS_REQ_12] The TIMER1 Driver shall have a scalable functionality to fit the needs of the ECU.

4.2 Non-functional requirements

[SRS_REQ_13] The driver shall be easy to use and understand.

[SRS_REQ_14] The driver shall be well-documented.

[SRS_REQ_15] The driver shall be efficient and use minimal resources.

[SRS_REQ_16] The driver shall be reliable and robust.

[SRS_REQ_17] In addition to the above requirements, the timer1 driver should also meet the following non-technical requirements

[SRS_REQ_18] The driver should be open source and freely available to use.

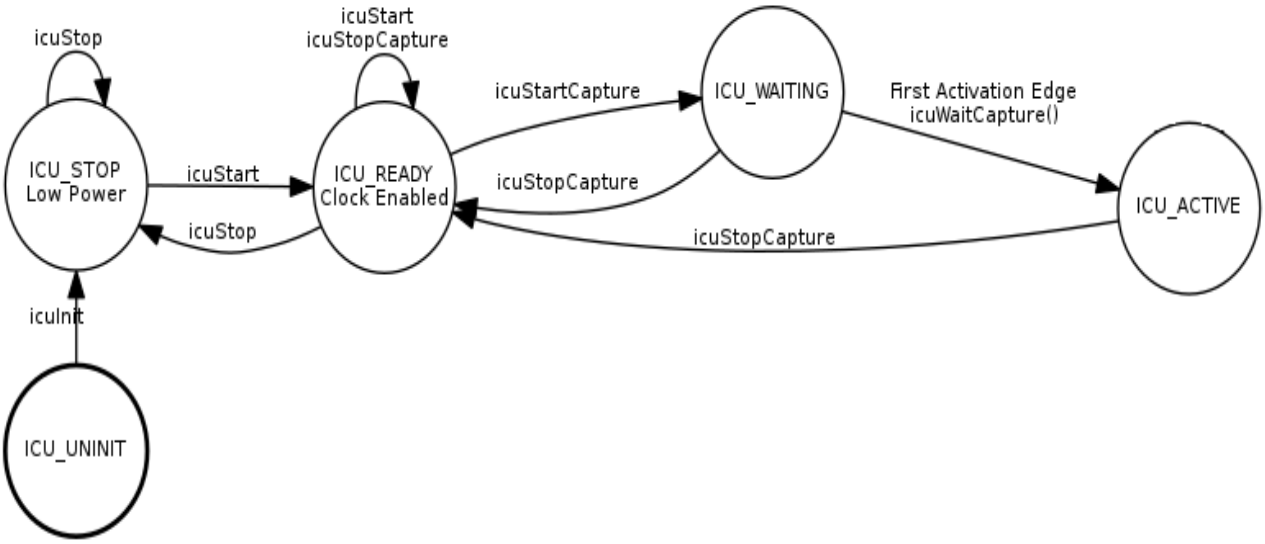
[SRS_REQ_19] The driver should be actively maintained and supported by the community.

[SRS_REQ_20] The driver should be well-tested and documented.

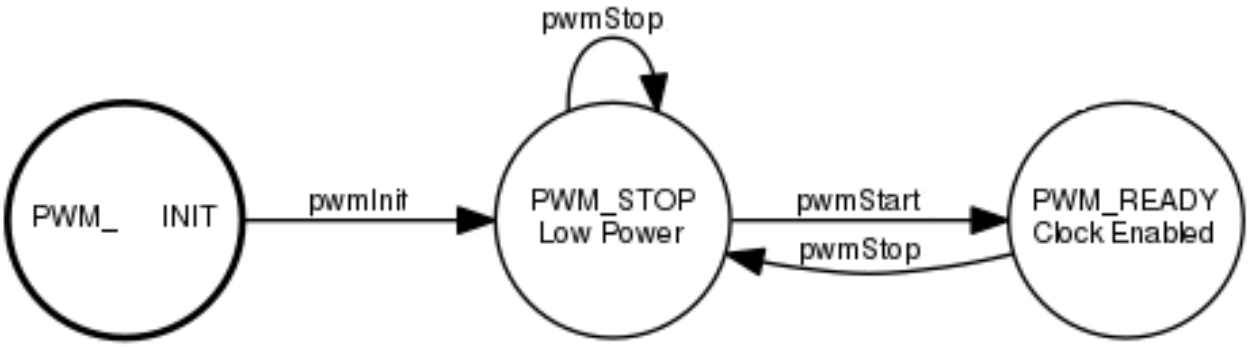
[SRS_REQ_21] The driver should be compatible with a variety of development tools and environments.

5. State machine

5.1 ICU



5.2 PWM


pwmStop
PWM_ INIT
pwmInit
PWM_STOP
Low Power
pwmStart
pwmStop
PWM_READY
Clock Enabled

6. Acceptance Criteria

The SPI driver shall be accepted when it meets the following criteria:

- The driver shall compile and run without errors on all AVR microcontrollers.
- The driver shall pass all unit tests.
- The driver shall pass all integration tests.
- The driver shall pass all system tests.

7. References

1. Developers of NTI team.
2. AVR Microcontroller Datasheets.