TIMER Driver for
AVR Microcontrollers

Version: 1st

Date: 25/10/2023

• Introduction

This document describes the Software Requirement Specification (SRS) for a TIMER driver for AVR microcontrollers. 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.

• Background

AVR microcontrollers are widely used in embedded systems. They are known for their low cost, power consumption, and ease of use to generate time delays, waveforms, or to count events.

• System Overview

The TIMER driver shall provide the following features:

1-Implement a full Timer driver for TIMER with the configuration technique.

2-The Timer Driver should support three channels (0,1,2) and it should be configured through the configuration structure passed to the init function.

3 -The Timer Driver should be designed using the Interrupts with the callback's technique.

4-The Timer Driver should support both normal and compare modes and it should be configured through the configuration structure passed to the init function.

5-The Timer Driver has 3 functions and six ISR's for Normal and Compare interrupts:

• Specific Requirements

4.1 General Behaviors

Timers/Counters are an essential part of any modern MCU. Remember it is the same hardware unit inside the MCU that is used either as Timers or Counter. Timers/counters are an independent unit inside a microcontroller. They basically run independently of what task CPU is performing. Hence, they come in very handy, and are primarily used for the following:

1. Internal Timer: As an internal timer the unit ticks on the oscillator frequency. The oscillator frequency can be directly fed to the timer, or it can be pre-scaled. In this mode it is used to generate precise delays. Or as precise time counting machine.
2. External Counter: In this mode the unit is used to count events on a specific external pin on a MCU.
3. Pulse width Modulation (PWM) Generator: PWM is used in speed control of motors and various other applications.

4.2 Requirements

the timer SWS shall define functions allowing:

a. Timer_init

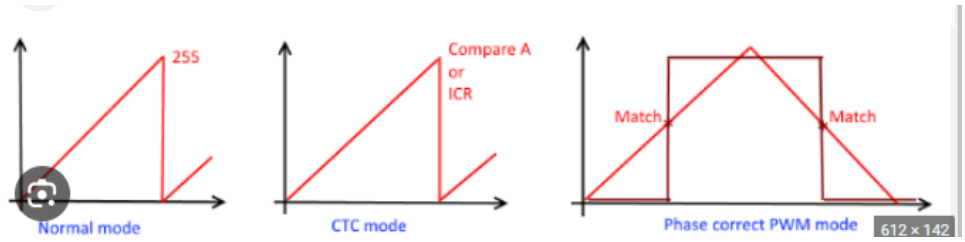
b. Timer_deInit

c. Timer_setCallBack

4.3 Technical requirements

- the driver shall be compatible with all AVR microcontrollers.
- Register Configuration: The driver should allow for easy configuration of timer registers. This includes settings like mode of operation, prescaler values, and compare/match values.
- Prescaler Support: AVR timers typically have a prescaler to divide the clock frequency. The driver should support setting the prescaler to adjust the timer's resolution.

- Interrupt Handling: It should efficiently handle timer interrupts. This includes setting up the interrupt vector and providing callback functions for user-defined actions.
- Mode of Operation: AVR timers can operate in different modes (e.g., normal, CTC, PWM). The driver should support these modes and allow for easy switching between them.



- Output Compare and Input Capture: AVR timers often have output compare and input capture functionality. The driver should provide an interface to configure and utilize these features.

4.4 non-technical requirements

- The driver shall be easy to use and understand.
- The driver shall be well-documented.
- The driver shall be efficient and use minimal resources.
- The driver shall be reliable and robust.
- In addition to the above requirements, the timer driver should also meet the following non-technical requirements:
- The driver should be open source and freely available to use.
- The driver should be actively maintained and supported by the community.
- The driver should be well-tested and documented.
- The driver should be compatible with a variety of development tools and environments.

5.File Structure

5.1.1 Header file structure

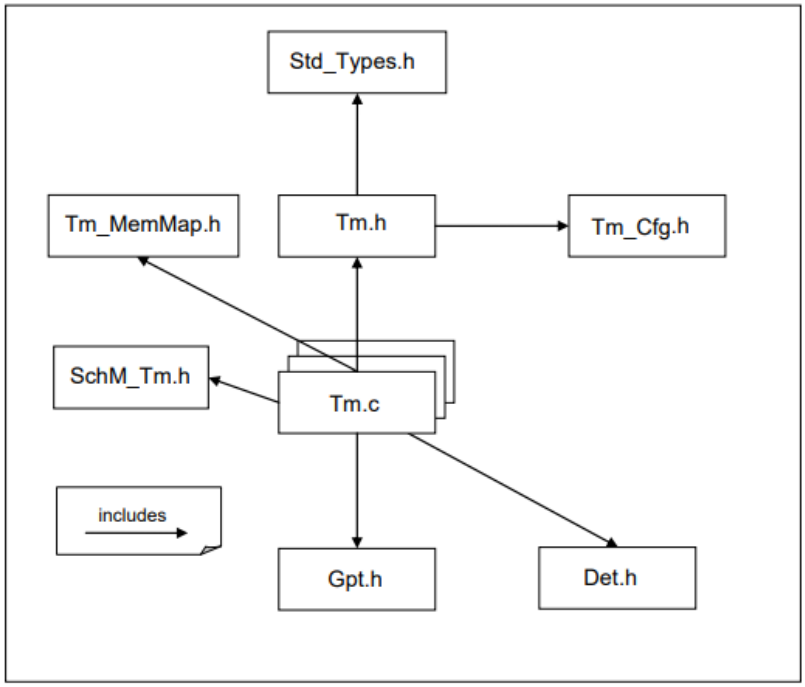


Figure 1: Include File Structure

6. Acceptance Criteria

The timer 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

AVR Microcontroller Datasheets