Software Requirements Specification (SRS) Clear Timer on Compare (CTC) Mode for Timer0

Authors: Burners Team (Ahmed Salah, Basem Moufreh,
Hassan El Gabass , Hazem El Morshedi , Mohamed Safwat).
Customer: NTI
Instructor: Mahmoud Ali, Ahmed Abd El Reheem

Contents

1. System Overview

The Timer0 module in the AVR ATmega32 microcontroller shall be configured to operate in the Clear Timer on Compare (CTC) Mode. In this mode, Timer0 will count up from an initial value of 0 to a user-defined compare value, triggering an output compare match interrupt when the counts match. This mode is suitable for generating precise time intervals and periodic events.

2. Functional Requirements

2.1 Timer0 Initialization

- The system shall provide a function to initialize Timer0 in CTC Mode.
- The user shall be able to specify the initial value (preload value) for Timer0.
- The user shall be able to set the compare value for generating interrupts.

2.2 Timer Start/Stop

- The system shall offer functions to start and stop Timer0.
- Starting the timer shall initiate the counting process from the specified initial value.
- Stopping the timer shall freeze the count.

2.3 Output Compare Interrupt

- The system shall allow enabling and disabling of the Timer0 output compare interrupt.
- When enabled, Timer0 shall generate an interrupt when the count matches the specified compare value.
- The system shall provide a user-defined interrupt service routine (ISR) that will be executed upon compare match.

2.4 Read Timer Value

- The system shall provide functions to read the current value of Timer0.
- Reading the timer shall return the current count value.

2.5 Timer0 Counter Behavior

- Timer0 shall increment from the initial value to the user-defined compare value.

- When the count matches the compare value, it shall trigger the output compare interrupt.
- The timer shall stop counting when the output compare interrupt is triggered.

3. Non-Functional Requirements

3.1 Performance

- Timer0 shall operate accurately at the specified clock frequency.
- The software shall be optimized for minimal CPU usage, ensuring it does not introduce significant overhead on the system.

3.2 Reliability

- Timer0 shall operate reliably without unintended resets or malfunctions.
- The interrupt handling for the Timer0 output compare match shall not disrupt the normal operation of the system.

3.3 Precision

- The system shall provide precise control over timing intervals with minimal deviation from the specified compare value.