Software Requirements Specification (SRS)
Overflow (Normal) 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 Overflow (Normal) Mode. In this mode, Timer0 will count up from an initial value of 0 to its maximum value (255 for an 8-bit timer) before overflowing and triggering an overflow interrupt if enabled. The mode is suitable for basic timekeeping and periodic interrupt generation.

2. Functional Requirements

2.1 Timer0 Initialization

- The system shall provide a function to initialize Timer0 in Overflow (Normal) Mode.
- The user shall be able to specify the initial value (preload value) for Timer0.
- The timer shall be configured to operate in normal mode .
- Select the appropriate prescaler according to specific requirements

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 Overflow Interrupt

- The system shall allow enabling and disabling of the Timer0 overflow interrupt.
- When enabled, Timer0 shall generate an interrupt when it overflows (reaches its maximum count value).
- The system shall provide a user-defined interrupt service routine (ISR) that will be executed upon overflow.

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 its maximum value.
- Upon reaching the maximum value, it shall overflow and reset to the initial value.
- Timer0 shall keep counting and overflowing in a continuous loop if it is running.

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 Timer0 overflow shall not disrupt the normal operation of the system.