INTERRUPT
Drivers

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 Interrupt 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: -

- Configuration (which elements of the module need to be configurable).
- Initialization.
- Normal Operation.
- Shutdown Operation.
- Fault Operation.
- …….

Non-Functional Requirements:-

- Timing Requirements.
- Resource Usage.
- Usability.
- Output for other WPs (e.g. Description Templates, Tooling,...).
- …….

3. Acronyms and Abbreviations

The following expressions are used within the Interrupt driver:

Expression	Explanation
ISR	Interrupt Service Routine. Also used as a macro to declare in C a cat2 interrupt service routine.
Interrupt Logic	This is the MCU logic that controls all interrupts for all devices. This is normally controlled by the OS
RETI	Return from Interrupt
ISQ	interrupt service query.

4. Functional Overview

An interrupt is a signal to the processor that an event has occurred that requires immediate attention. Interrupts are used to implement real-time response, multitasking, and improve efficiency.

5. Requirement Specification

5.1 Functional Requirements

- [EXIT_001] The Interrupt Driver shall support a function to initialise the Interrupt 0,1,2.
- [EXIT_002] The driver shall be compatible with all AVR microcontrollers.
- [EXIT_003] The Interrupt Driver shall support a specific basic static configuration.
- [EXIT_004] The Interrupt Driver shall support a function to Enable Interrupt.
- [EXIT_005] The Interrupt Driver shall support a function to Disable Interrupt.
- [EXIT_006] The Interrupt Driver shall support a function to Callback.

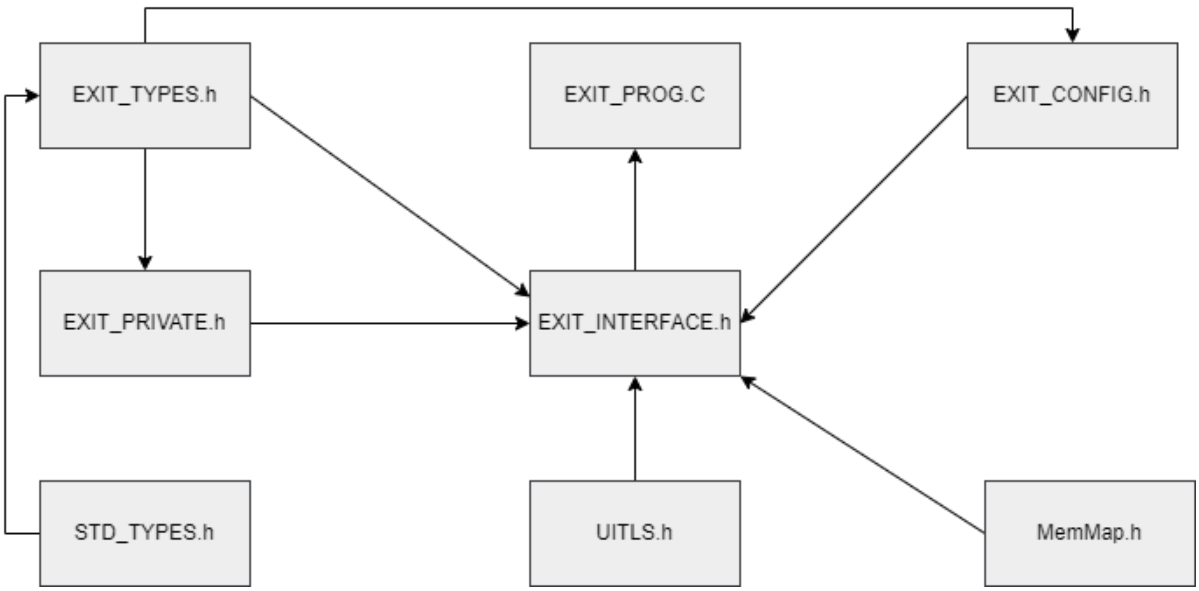


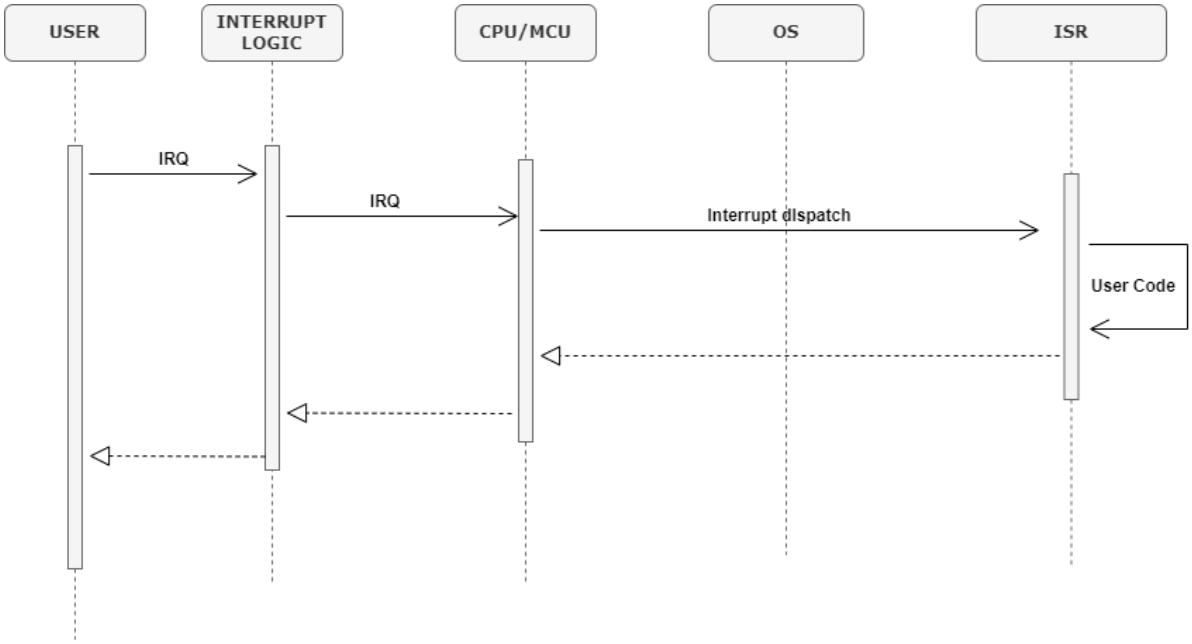
Figure 1: External Interrupt Files

5.2 Non-functional 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 Interrupt 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.

6. State Machine

Sequence diagram



Figuare2: Interrupt Sequence

7. Acceptance Criteria

The ADC 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.

8. References

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