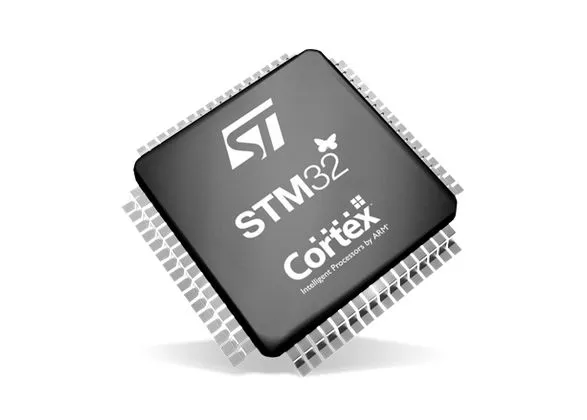
Interrupts



Semester 3 Embedded Systems

Lab\_03\_Interrupts

Sept 27, 2022.

Student: Andre Sanao

Course: Technology

# Introduction

The purpose of this assignment is to create an external interrupt to detect a signal event external to the microcontroller. This document will contain all the steps necessary to trigger an external interrupt. A button and an LED will be used in the demonstration. The following section will show the procedure and implementations to be able to achieve an external interrupt.

# Research

In the previous assignment, SysTick was used to implement an interrupt. Interrupts are initiated by an electrical signal called “interrupt request” which can be issued by exceptions which is built in the microcontroller and by an external connected hardware which are called “external interrupts”. Any hardware that has 2 or more states can trigger an external interrupt.

# Design

Diagram

Description automatically generated

Figure 1 Exception behavior

1. When an exception occurs, the current instruction stream is halted and the processor accesses the exceptions vector table. (More details of the vector table can be seen at 14.1.3 of the Reference Manual)
2. The vector address of that exception is loaded from the vector table.
3. The exception handler starts to be executed in handler mode.
4. The exception handler returns to main (assuming no further nesting).

# Testing

To start with the configuration of the interrupt, the peripheral clock needs to be enabled. After setting the clock, the external interrupt configuration, interrupt mask, Nested Vector Interrupt Controller(NVIC) Priority and Interrupt Requests(IRQ) needs to be set to the corresponding GPIO pin of the external button. After the initiation config, the handler function is created to execute interrupts.

To test if the program enters the handler, a flag variable is created within the handler. When the flag is 1, the handler executes the activity. Before the activity is executed the flag resets to 0 before taking any action. For the assignment, a button is used to execute the handler to turn on/off the LED.

# Conclusion

Learning about interrupts is great for giving other program to be executed first by forcing the processor to take care more important activities. Interrupts are better than polling because polling waste a lot of CPU cycles and is a synchronous activity while interrupts alert the CPU to perform certain action quickly. But sometimes it is inefficient when the board interrupts the CPU frequently for activities. To put it simply an interrupt is asynchronous whereas polling is synchronous.

# Bibliography

STMicroelectronics NV. (2017). *RM0316 Reference manual* [Ebook]. Retrieved from <https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303x6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf>