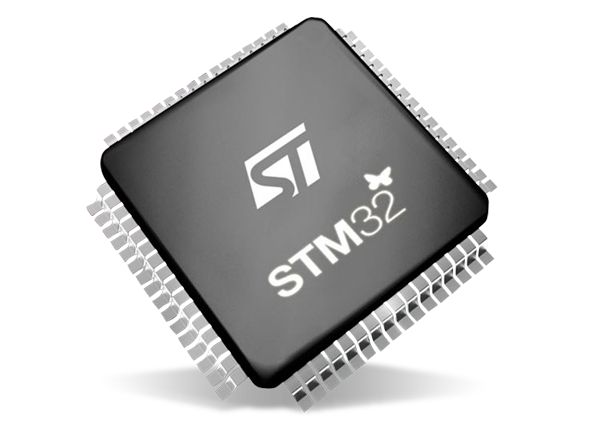
Lab\_02\_SysTick



Semester 3 Embedded Systems

Assignment 2

Sept 14, 2022.

Student: Andre Sanao

Course: Technology

# Introduction

In this assignment, the learning goal is to gain basic knowledge of the system timer (SysTick) and understanding the basics of interrupt handling. The system timer can be used in many ways but the most basic usage of the system timer is toggling an LED. More details of the SysTick can be read in the Cortex M4 User Guide paragraph 4.4.

# Procedure

To use the system timer (SysTick), it is required to enable the right bits to make the timer delay countdown. These bits are initialized at the start. During the initialization, the SysTick Control and Status Register needs to be disabled first for configuration. When disabled the following procedure occurs:

* The reload register is set. The reload register basically reloads the counter when it reaches 0. (Cortex M4 User Guide 4.4.2)
* Setting the interrupt priority to the largest value making the least urgent.
* The SysTick current value register is set to 0 to avoid random value during initialization.
* The processor clock is enabled.
* Disable the SysTick interrupt. (This is only disabled for part A of the assignment.)
* Reenable the SysTick.

The interrupt period values can be set using these configurations. More details are in the Cortex M4 User Guide (4.4.1)

There are 2 ways to make the delay. For the first part of the assignment, the Counter Flag will be used to create a delay. Counter Flag is used to create an interrupt when it is enabled. When the counter reaches 0, Counter Flag is enabled automatically and the reload value reloads the counter to restart the loop.

In the second part of the assignment, it has the same initialization but instead the SysTick interrupt handler is enabled. This generates an interrupt request together with the Counter Flag.

Both configurations have the same function and it was used to demonstrate a toggling LED using system timers.