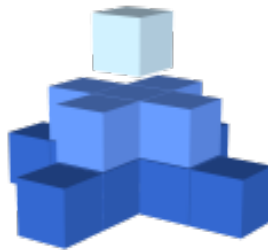


SRS SysTick

System Requirements Specification



Status: Draft

Author: automatically generated from the SiSy model

Table of contents

1	Purpose	2
2	Overall description of the task.....	2
3	Functional requirements.....	2
4	Hardware requirements.....	3
5	Process requirements	4
6	Attachment	5



1 Purpose

All elements of this project are parts of a course for the professional development of embedded systems. This Embedded Systems Engineering course is intended to develop a broad interdisciplinary understanding and knowledge of the participants as well as to develop practical skills for the realization of embedded systems.

The hardware platform for this course is the mySTM32 Board lite. It has a microcontroller of the STM32 family and all required input and output devices or add-ons.

2 Overall description of the task

A microcontroller application has to be developed in which the user sees how the SysTick of the microcontroller works. The task is: Develop a solution for this in which the various SysTick events (10 ms, 100 ms, 1 s) of the microcontroller are made clear by different flashing LEDs and a beep from the speaker.

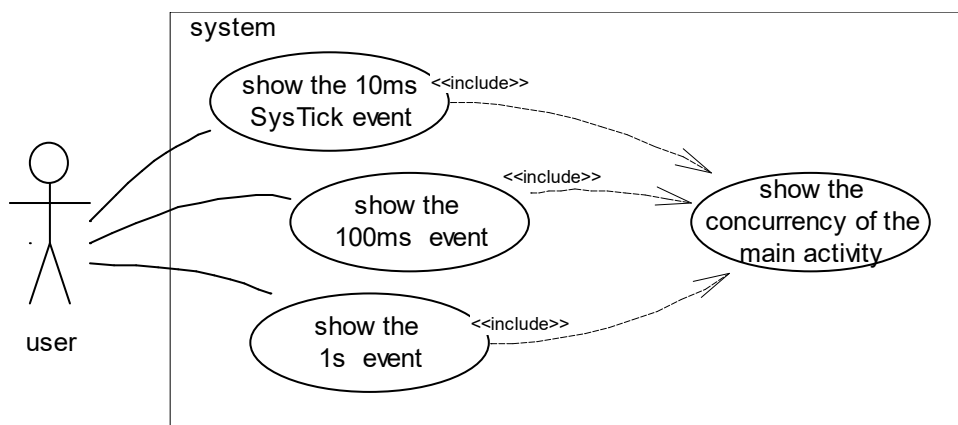


figure 1: uc: SysTick tasks, user's perspective

List of top level requirements:

- system: show the 10ms SysTick event
- system: show the 100ms event
- system: show the 1s event
- system: show the concurrency of the main activity

3 Functional requirements

While an LED is continuously faded in and out, the speaker should be triggered every 10 milliseconds, the red LED every 100 milliseconds and the green LED once per second..

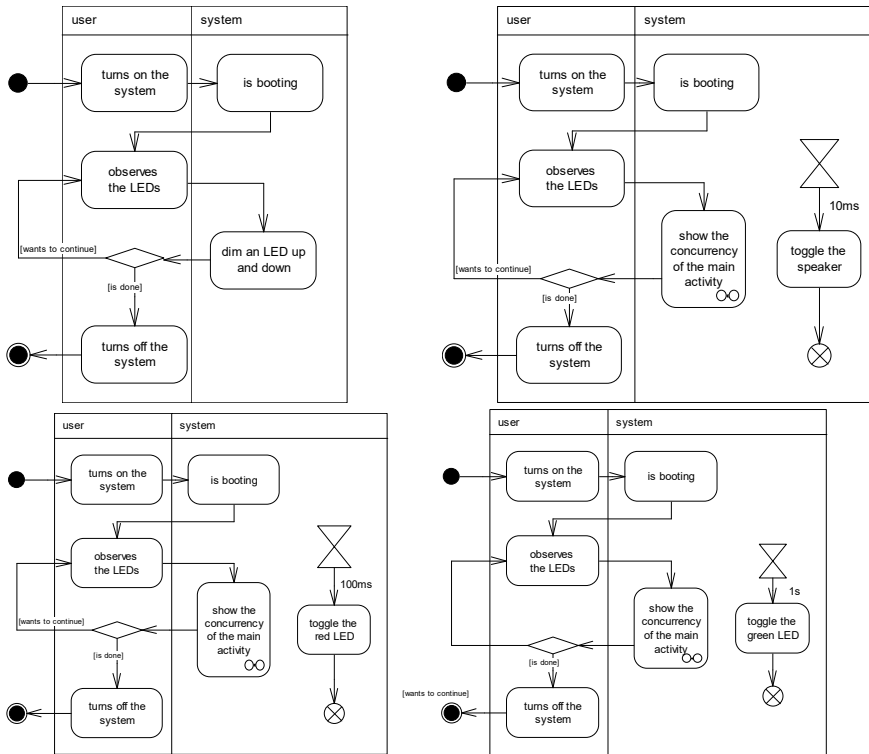


figure 2: activity modells: concurrency of the main activity, 10ms, 100ms, 1s, event

4 Hardware requirements

The hardware platform for this course is the mySTM32 Board lite. It has a microcontroller of the STM32 family and all required input and output devices or add-ons.

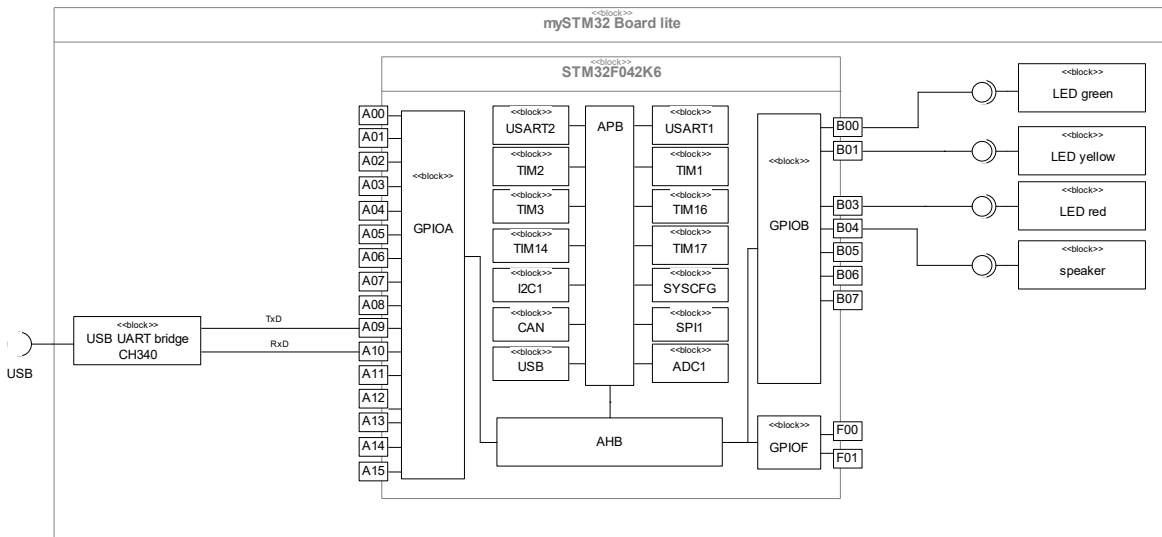


figure 3: SysTick HRM

connected pinB3 : LED red

- connected USB
- connected pinB4 : speaker
- connected pinB0 : LED green
- connected pinB1 : LED yellow



5 Process requirements

A software process is the defined sequence of activities, the agreed rules, techniques, tools and the expected results of the activities for the production of software. Defined software processes ensure the plannability, controllability and quality of results in the manufacture of software. The following simple software process is agreed as a binding workflow for this course.

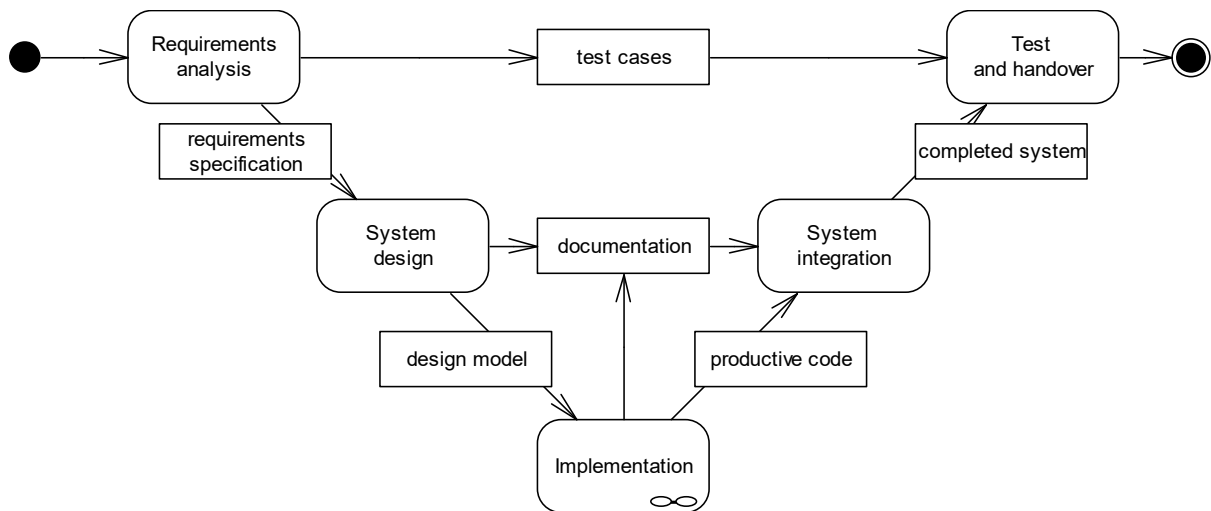


figure 4: act: lightweight model driven embedded software process

table 1: lightweight model driven embedded software process

Activity	Expected results
Requirements analysis	<ul style="list-style-type: none"> - User's perspective as use case diagram (as SysML / UML model) - required functionalities as activity diagrams (as SysML / UML model) - Test cases (as a document) - HRM hardware resource model (as SysML model) - SRS System Requirements Specification (as a document)
System design	<ul style="list-style-type: none"> - Class model of the concept level / architecture model (as UML model) - if necessary, state model (as UML model) - System documentation (as a document)
Implementation	<ul style="list-style-type: none"> - Class model of the realization (as UML model) - Behavioral models of the realization (as UML model) - Productive code (as a transferable format of the target platform, * .hex, * .elf) - System documentation (as a document)



System integration	- hardware software integration - the completed system
Test and handover	- the tested system - the technical system documentation (as a document) - the user documentation (as a document)

6 Attachment

List of figures

figure 1: uc: SysTick tasks, user's perspective 2
figure 2: show the concurrency of the main activity 3
figure 3: show the 10ms SysTick event 3
figure 4: show the 100ms event Fehler! Textmarke nicht definiert.
figure 5: show the 1s event Fehler! Textmarke nicht definiert.
figure 6: SysTick HRM 3
figure 7: act: lightweight model driven embedded software process 4

List of tables

table 1: lightweight model driven embedded software process 4