

SRS Timer

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	3
6	Attachment	4



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 is to be developed in which a timer acts as a trigger for a counter and the counter value is transferred to the PC.

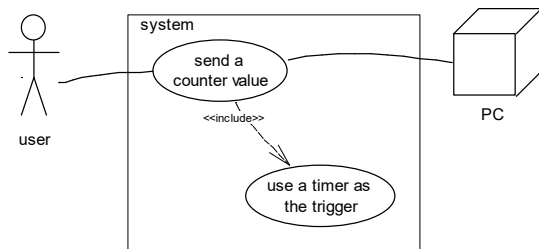


figure 1: uc: TIMER tasks, user's perspective

List of top level requirements:

- system: send a counter value
- system: use a timer as the trigger

3 Functional requirements

After switching on the system, the timer is configured to 5 ms. As long as the system is not switched off, an 8-bit counter counts up in the timer event. The counter value is sent to the PC for monitoring.

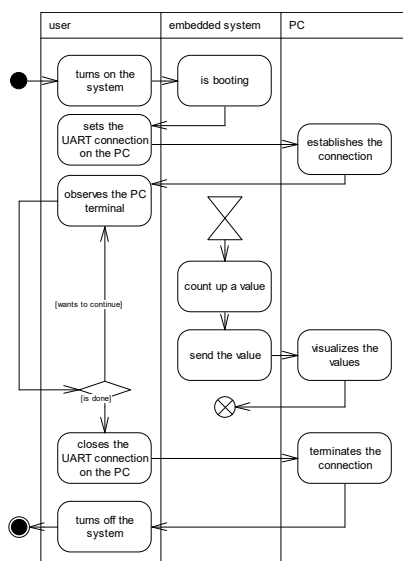


figure 2: send a counter value



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.

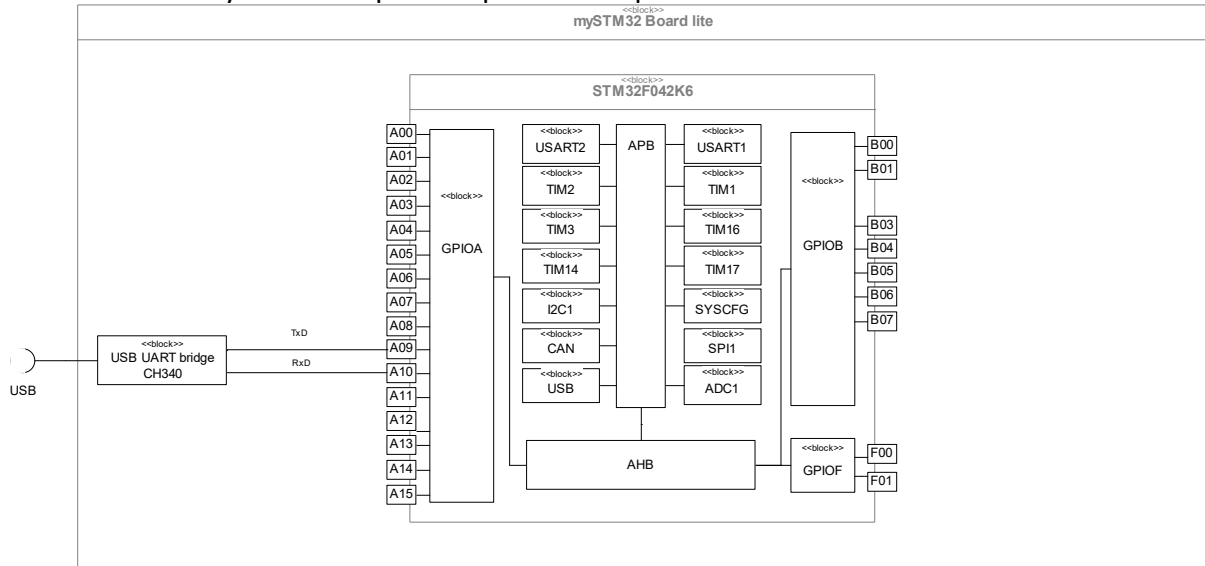


figure 3: TIMER HRM

- connected USB

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.

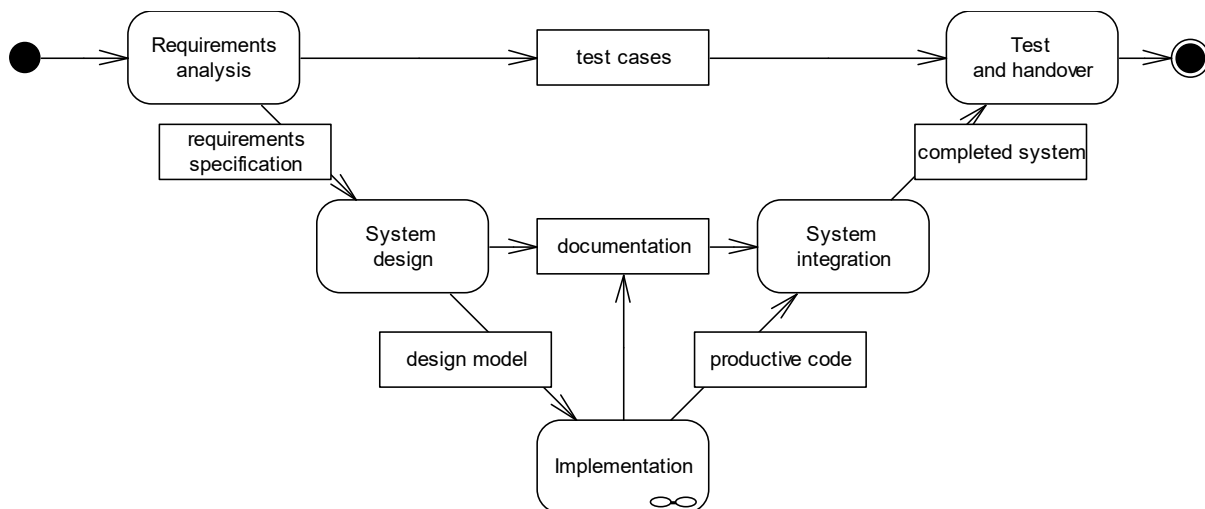


figure 4: act: lightweight model driven embedded software process



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	<ul style="list-style-type: none"> - hardware software integration - the completed system
Test and handover	<ul style="list-style-type: none"> - the tested system - the technical system documentation (as a document) - the user documentation (as a document)

6 Attachment

List of figures

figure 1: uc: TIMER tasks, user's perspective 2
figure 2: send a counter value 2
figure 3: TIMER HRM 3
figure 4: act: lightweight model driven embedded software process..... 3

List of tables

table 1: lightweight model driven embedded software process 4