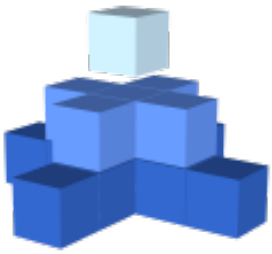
SRS project 2 smart alarm system

System Requirements Specification



Status: Draft

Author: automatically generated from the SiSy model

Table of contents

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

In this small project, various components from this tutorial and also new modules of the PEC framework will work together in one application. In the style of an embedded software project, a digital clock with temperature display, optical and acoustic alarm function, a UART terminal for setting the times and other options, control of the display lighting according to room brightness and possibly a snooze function is to be created.
A number of nice additional modules (add-ons) are available for the various myMCU boards. These should be put together to form a small system. The mySTM32 Board light with LCD and temperature sensor add-on is used as the target platform. With these components it should be possible to realize the prototype for a smart DIY solution.
As a refinement of this project idea, the idea arose to make the form factor of the system a little cooler. The I²C temperature sensor should be placed on the grid field of the mySTM32 Board light and the LC display should be arranged vertically.

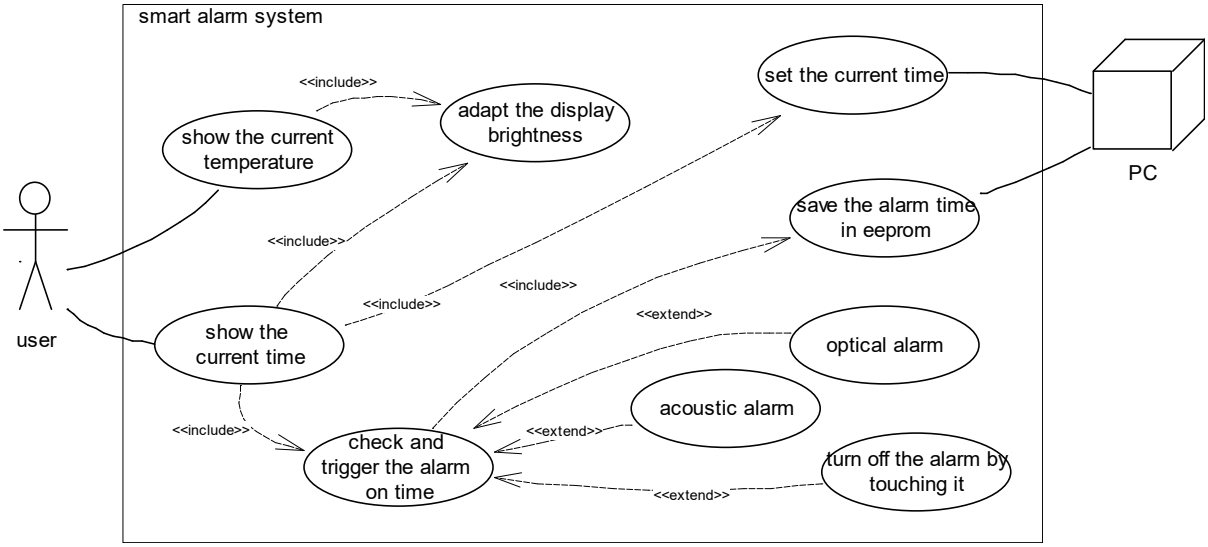


figure 1: uc: project2, user's perspective

List of top level requirements:

- check and trigger the alarm on time
- show the current temperature
- save the alarm time in eeprom
- acoustic alarm
- optical alarm

- turn off the alarm by touching it
- show the current time
- set the current time
- adapt the display brightness

3 Functional requirements

After switching on the system for the first time, the current time must be set. This should be done via a UART terminal on a PC. To regulate the display brightness, the light sensor determines the ambient brightness. The temperature sensor provides the current ambient temperature. The time and temperature are shown on the display. The display lighting should only be switched on when it is dark. For the alarm function, the desired alarm time can be set via the UART connection. The set alarm time must be saved in an EEPROM so that it is retained even after the supply voltage has been disconnected. When the alarm time is reached, an optical and acoustic alarm signal must be generated. The alarm can be switched off by simply touching, for example, nudging the system. This is determined with a MEMS sensor.

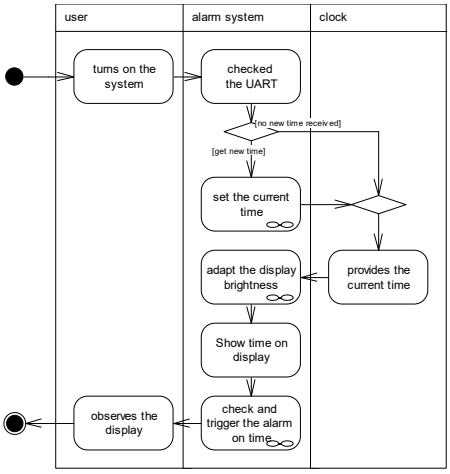


figure 2: act: show the current time

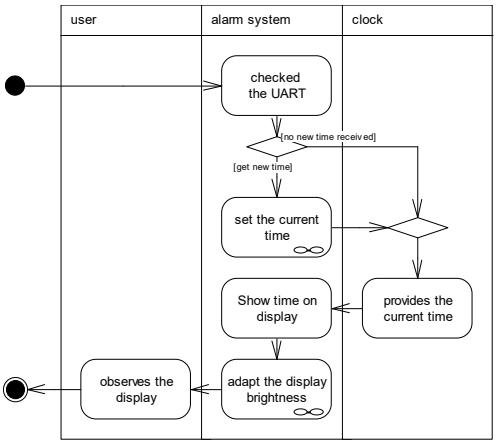


figure 3: act: show the current temperature

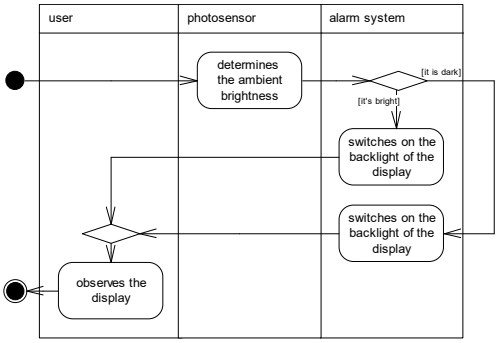


figure 4: act: adapt the display brightness

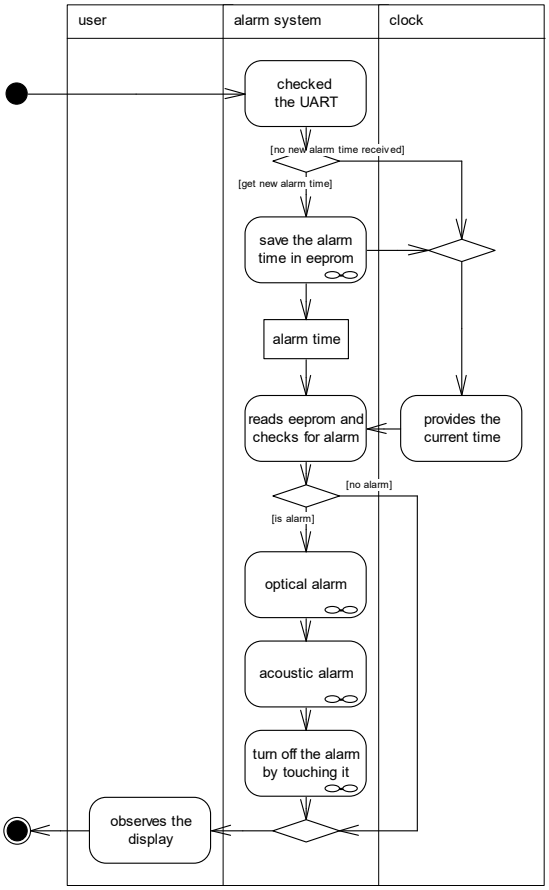


figure 5: act: check and trigger the alarm on time

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.

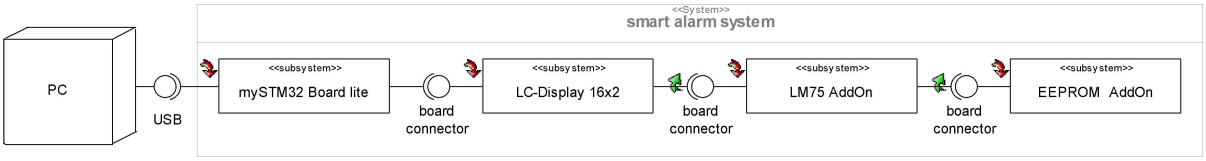


figure 6: HRM project smart alarm system

The system consists of the following main components:

- mySTM32 Board lite
- LC-Display 16x2
- LM75 AddOn
- EEPROM AddOn

The following interfaces are used to connect the components:

- board connector
- 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.

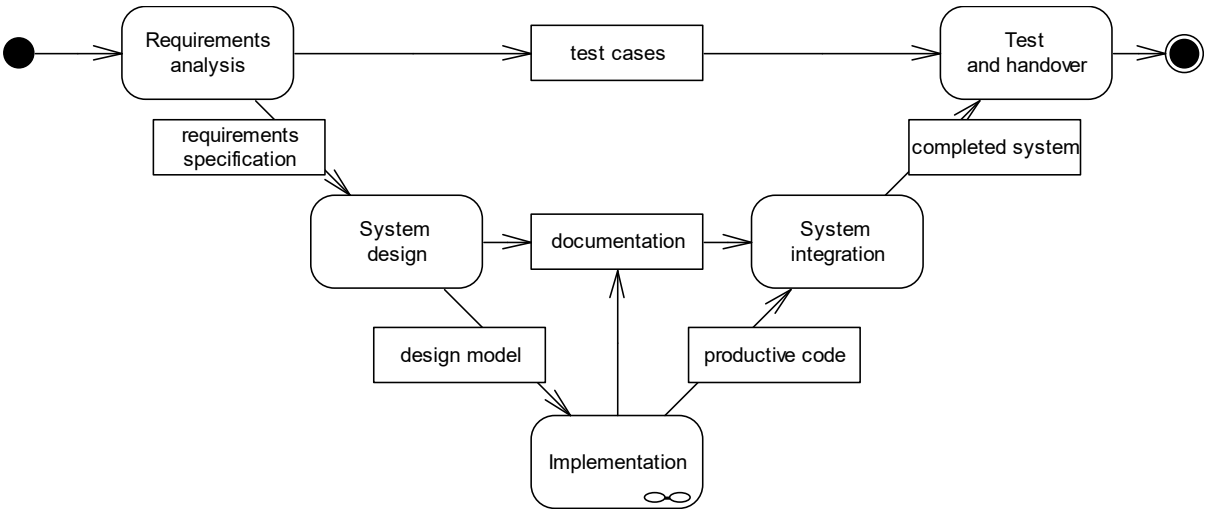


figure 7: act: lightweight model driven embedded software process

Table 1: lightweight model driven embedded software process

Activity	Expected results
Requirements analysis	- 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	- Class model of the concept level / architecture model (as UML model) - if necessary, state model (as UML model) - System documentation (as a document)
Implementation	- 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

List of tables