SRS CAN

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

Two controller boards are to be networked via the CAN bus. Each board should have a button and an LED. When you press the button, the LED on the other board should light up.

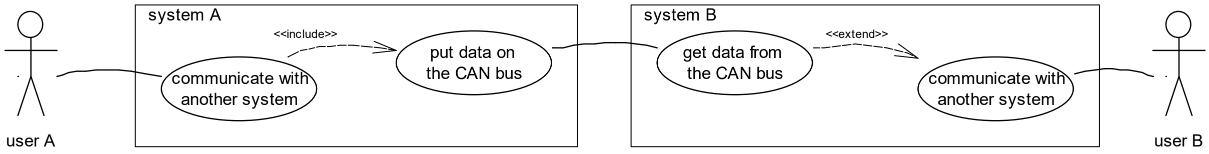
system A
<<include>>
put data on the CAN bus
communicate with another system
user A
system B
<<extend>>
get data from the CAN bus
communicate with another system
user B

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

List of top level requirements:

- system A: communicate with another system
- system B: get data from the CAN bus
- system A: put data on the CAN bus
- system B: communicate with another system

3 Functional requirements

After switching on both systems, the CAN connection is established. Every 100 milliseconds the status of the button of system A is transferred to the CAN bus. When the button is pressed, the LED in system B is switched on.

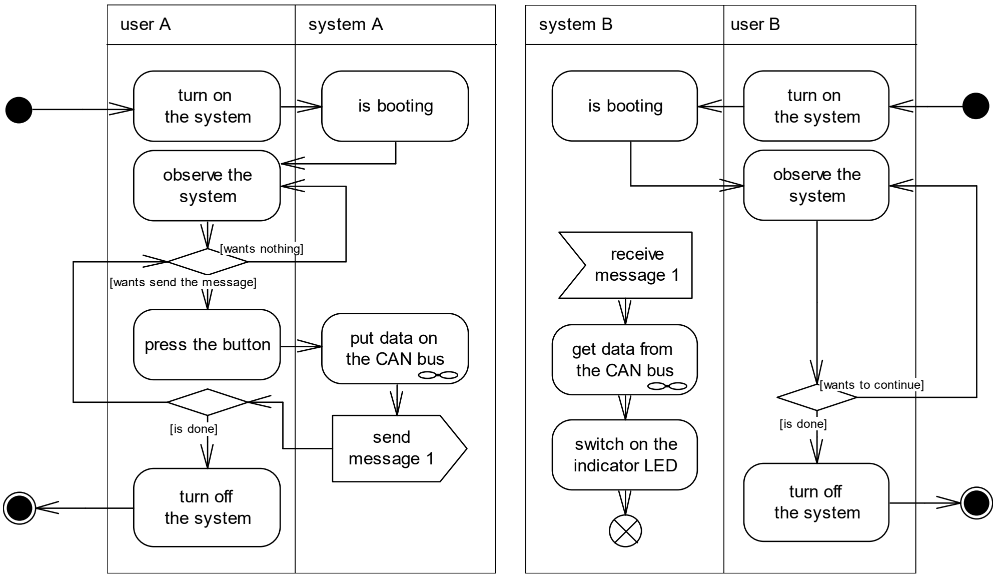

figure 2: activity model CAN communicatione with an other system

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.

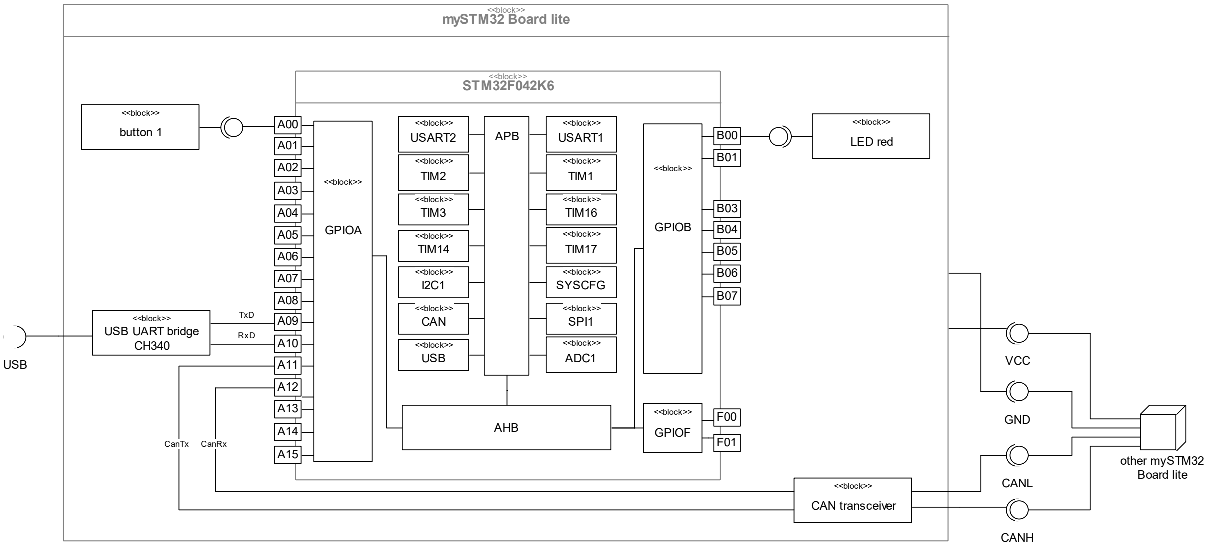

figure 3: CAN HRM

- connected USB
- connected system A / pinA0 : button 1
- connected system B / pinB0 : LED red
- connected system A / VCC : system B / VCC
- connected system A / GND : system B / GND
- connected system A / CAN Hi : system B / CAN Hi
- connected system A / CAN Lo : system B / CAN Lo

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.

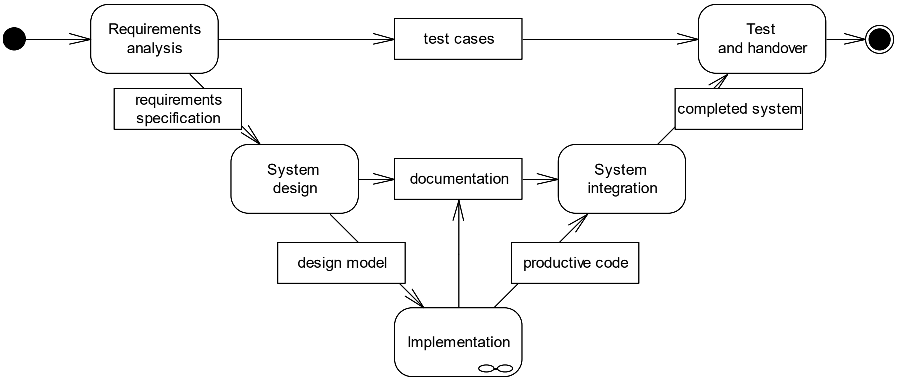

figure 4: act: lightweight model driven embedded software process

table 1: lightweight model driven embedded software process

Activity	**Expected results**
Requirements analysis	- 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