Lab Assignment Requirements and System Architecture

Latest information about this course can be found on the eLearning platform ELLI of University of Applied Sciences Bremerhaven:

Magazin » Studiengänge » Embedded Systems Design » ESD S24 » IT-MBS Model-based SW-Development/Real-time Software S24

`https://elli.hs-bremerhaven.de/goto.php?target=crs_344880&client_id=elli`

Work within your team through this assignment. Make sure with the course instructor that you have found an apropriate solution. Ask the course instructor if something is unclear.

Learning Outcomes

- Identify system elements for a given embedded system
- Specify functional system behaviour and corresponding timing
- Create a system architecture and generate system block diagrams
- Create a requirements document
- Define acceptance test cases including acceptance criteria

Requirements and System Architecture

8.1 Project Description: Autonomously Driving Vehicle

Imagine you work in a development team and the product manager gives the task to develop a new *autonomously driving vehicle* that automatically controls its driving behaviour. The robot must be equipped with an emergency push button. The system must react on user-operated push button requests. When the push button is pressed once the system must stop all movement. When the push button is pressed again the robot must continue its maneuver. The push button requests must be indicated with an optical indicator.
The robot must have the following features:

Features

F1 Start/Stop feature using the built-in buttons on the brick (left/right, top/down, center). The user may start the system by pressing specific built-in button and the user may stop operation by pressing another (or same) built-in button. Start/stop commands must be confirmed by pressing a specific built-in button.

F2 Integrate display information feature. Operating modes must be displayed on the display. If a button was pressed the action must be displayed. If the user must confirm an action, this must be displayed.

F3 Emergency push button (must work with every other maneuver). Use the external pushbutton (touch sensor). One press = maneuver stop. double press = maneuver resume

The robot must be able to execute the following maneuvers:

Maneuvers

M1 Drive a straight line forward. Distance 1 m.

M2 Drive a square with x cm side length. x must be entered into the model file before deployment.

M3 Drive a circle with x cm diameter. x must be entered into the model file before deployment.

M4 Drive to a given target point (x, y, δ). x and y and δ must be entered into the model file before deployment. δ is the final angle difference to the starting angle.The trajectory must be driven without a stop, i.e. speed must be larger than 0.

Each maneuver may be realized in a separate model file. All features F1-F3 must work with every other maneuver M1-M4. The maneuvers must work for low, medium and high speed. The robot is a two wheel robot as you can find in the manual of the Lego Mindstorm boxes. Additional sensors (as color sensor or ultrasonic sensor or similar are not allowed).

The system must be safe against user interaction.

8.2 Organization

Build project teams with min. 2 and max. 4 members. Discuss your ideas and work out the documents together. Create a team and complete Lab Assignment: Build a Project Team on ELLI. You can share and upload working documents and your results to your ELLI lab group on the course website.

8.3 System Performance Specification (SPS)

Before we start coding we need a measurable specification of the system under development. Based on this specification we will employ system architecture and software structure. Also the system performance specification is the basis for testing. Each single requirement will be tested against its specification.

8.3.1 Discuss the following question within your team:

Are the given requirements (from the product manager) complete? Do you need more information about the systems behavior? If the function of the system is unclear simulate its functionality and write it down.

8.3.2 Set up a specification document

Establish a specification document in Microsoft Excel. Each requirement gets it's own id. Once an ID is created never change the ID number. Put the IDs in separate rows. Try to specify requirements as atomar as possible. Group similar requirements. Specify in another column if the requirement is functional or non-functional. Here is a list of groups that must be covered

- Inputs/Outputs (min, max, resolution, unit)
- What are the event triggers to which the system reacts?
- Tasks and their sample time
- Identify the operating modes of the controller and specify the events to which your system reacts. Set a default operating mode.
- Identify the transitions between the different operating modes. Between which modes can the system transition? Are there any operating modes (states) that run in parallel?
- Startup behaviour
- Functional description
- and so forth...

Add a column for priority, severity for fall-out, likelihood of fall-out and risk (product of severity and likelihood).

8.4 Creating a System Architecture

Based on the system performance specification create a system architecture that describes the topologic structure of your system with inputs and outputs. For this draw schematic system block diagrams on a sheet of paper first. Draw a top level architecture first. Then refine the architecture step by step for each module. Try to keep out the details if possible. Don't connect modules with lines but bring modules together into their subsystem (see figure 1 and 2). You may use SIMULINK models to draw the diagrams.

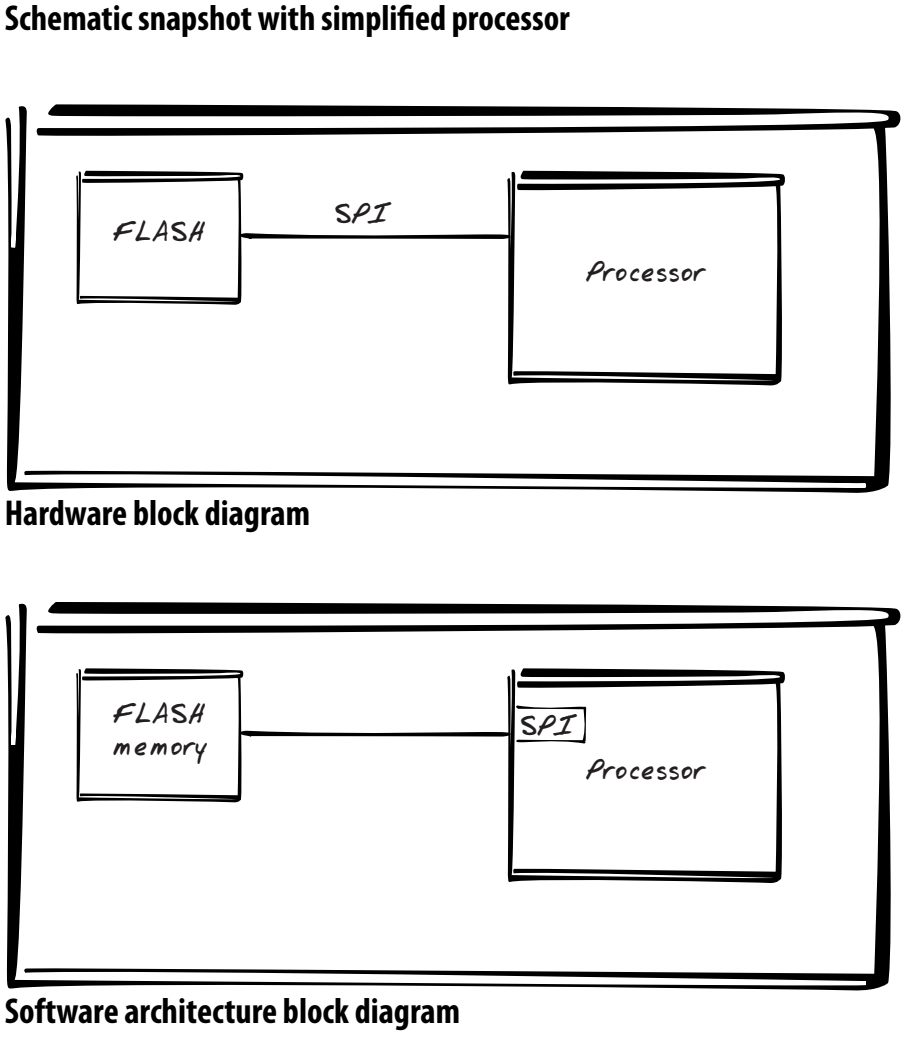

Figure 1: Example for a system architecture, Source: [**?**]

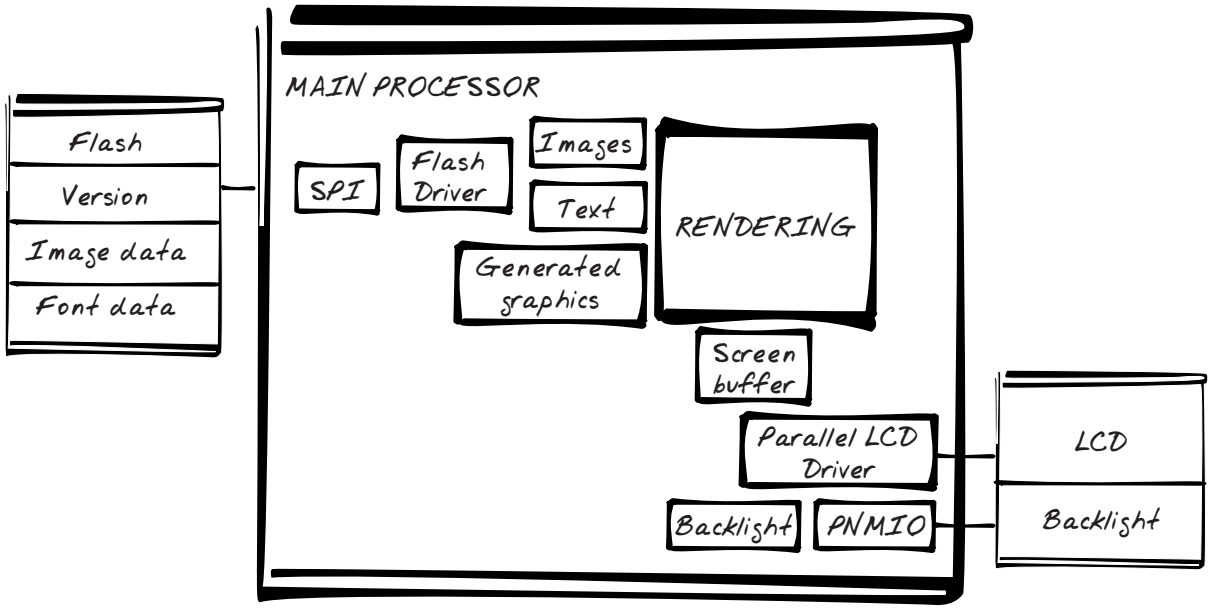

Figure 2: Example for a Software Block Diagram, Source: [**?**]

Now draw a control hierarchy diagram that connects the modules together (see figure 3). You may add some pieces here that you haven't thought of before. Add the architecture charts to your system performance specification. You may use SIMULINK models to draw the diagrams.

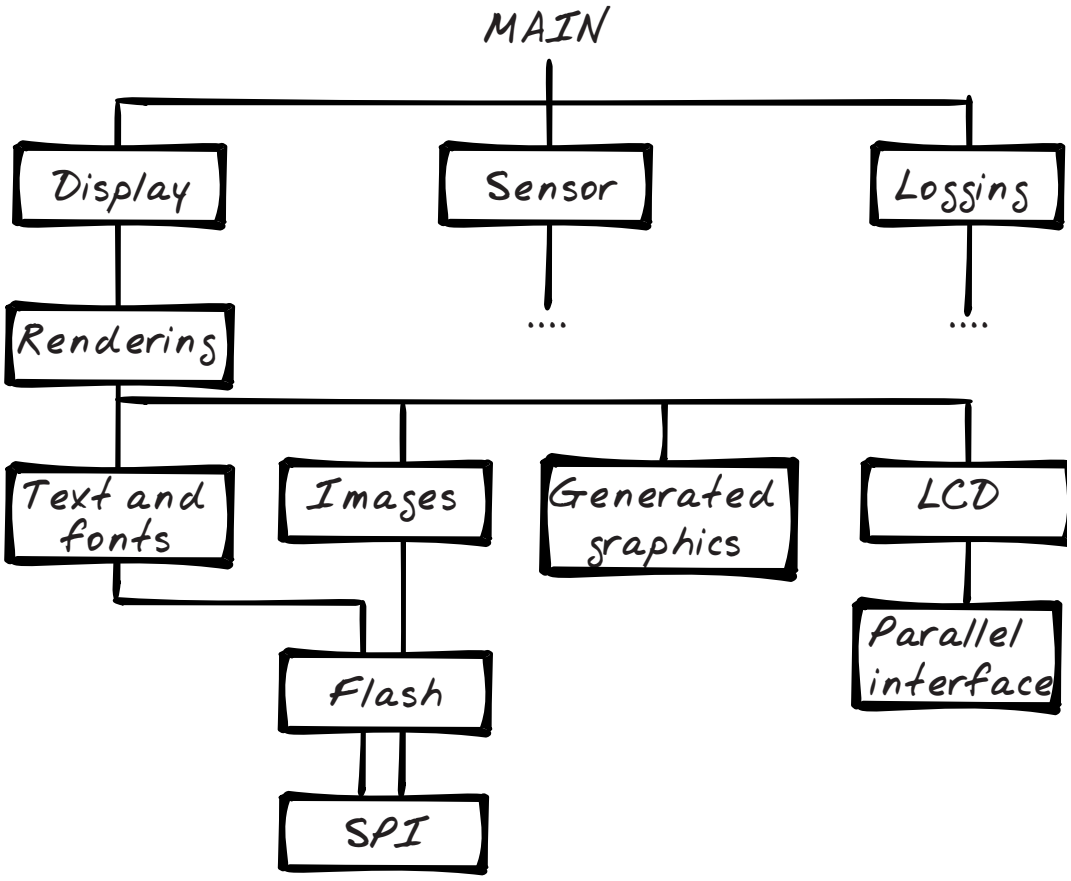

Figure 3: Example for an Organizational Diagram, Source: [**?**]

8.5 Requirements Analysis

An important part of the development is testing of your system against the requirements. Define for each feature (F1-F3) and each maneuver (M1-M4) use cases of system application that you are going to test later on. Add these use cases to your system performance specification.