



Designing an interface for a Generic Gateway

What guidelines should be considered when designing an interface for visualization and configuration of communication protocols withing embedded systems?

Johan Erlandsson

MASTER'S THESIS 2021

Designing an interface for a Generic Gateway

What guidelines should be considered when designing an interface
for visualization and configuration of communication protocols
withing embedded systems

Johan Erlandsson



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Designing an interface for a Generic Gateway

What guidelines should be considered when designing an interface for visualization and configuration of communication protocols withing embedded systems

Johan Erlandsson

© Johan Erlandsson, 2021.

Supervisor: Staffan Björk, Department of Computer Science and Engineering

Advisor: Olle Norelius, Infotiv

Examiner: Josef Wideström, Department of Computer Science and Engineering

Master's Thesis 2021

Department of Computer Science and Engineering

Chalmers University of Technology and University of Gothenburg

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Typeset in L^AT_EX

Gothenburg, Sweden 2021

Contents

1	Introduction	1
2	Background	3
2.1	Infotiv	3
2.2	Communication protocols	4
2.3	Desired features and existing products	4
2.3.1	Mathworks Simulink	4
2.3.2	Vector CANoe	5
3	Theory	7
3.1	Visual programming	7
3.2	Information visualization	8
3.3	Design principles	8
4	Methodology	9
4.1	Wicked problems in design	9
4.2	Research through design	9
4.3	Agile methodology	10
4.4	Double diamond	11
4.4.1	Discover	12
4.4.1.1	Personas	12
4.4.1.2	User interviews	13
4.4.1.3	Comparison of methods	13
4.4.2	Define	13
4.4.2.1	Affinity Diagramming	13
4.4.2.2	Weighted Matrix	13
4.4.2.3	Comparison of methods	14
4.4.3	Design	14
4.4.3.1	Cognitive walk through	14
4.4.3.2	Heuristic evaluation	14
4.4.3.3	Comparison of methods	15
5	Planning	17
5.1	Solving wicked problems	18
5.2	Infotiv's project planning	18

5.3	Scope	19
5.4	Process	20
5.5	Ethical considerations	21
6	Execution and Process	23
6.1	Prestudies	23
6.2	Change of planning	24
6.3	First iteration	25
6.3.1	Change of planning for this sprint	26
6.4	Second iteration	26
6.4.1	High fidelity prototype	26
6.4.2	Evaluation of high fidelity prototype	32
6.5	Third iteration	33
6.5.1	Evaluation of implementation	33
7	Results	35
7.1	Generic gateway application	35
7.2	Guidelines	39
7.2.1	Let the user name the different elements of the configuration	40
7.2.2	Keep a minimalistic version of the GUI and do not make it to cluttered for the user	40
7.2.3	Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.	41
7.2.4	Make sure to involve all product owners and testers throughout the project	41
7.2.5	Keep a good color combination with one primary and one secondary color that is distinguishable from each other.	41
7.2.6	Keep the interface expandable for bigger configurations.	41
7.2.7	Communicate in good time if there will be eventually cuts from the design compared to the initial plan.	42
7.2.8	Create tabs to one block for the user to shift through the different pages.	42
7.2.9	Give the user feedback if something in the configuration is about to be incorrect.	42
7.2.10	Know what is reasonable to extract from a prototype to an actual implementation according to the time plan.	42
7.2.11	Make sure that the different paths from each screen is decided upon early on so you know how to continue with it.	43
8	Discussion	45
8.1	Results	45
8.2	Process	46
8.3	Validity and Generalisability	47
8.4	Ethical aspects	47
8.5	Future work	48
8.5.1	Implementation	48
8.5.2	Evaluation	48

9 Conclusion	51
---------------------	-----------

1

Introduction

Today we can see network communication almost everywhere in many parts of our lives. No matter what industry the means of network communication is utilized[21]. As technology is developed more and more rapidly it is important that we utilize the benefit of different innovations to collaborate so we do not "reinvent the wheel" several times.

Within embedded system and especially in the automotive industry there are a lot of different units that communicates with different network protocols. [22] In order to make fast progress and benefit from the development of others a translation device is in place. A device, either hardware or software that translates and transports the different protocols so they can understand and communicate with each other. Infotiv is having plans to develop a product that will go under the name Generic Gateway that will serve as a tool to translate protocols from one type to another as well as making sure it is received at the right unit/bus.

Infotiv has hired me to develop a graphical user interface for the Generic Gateway. An interface where it is possible to get an overview of every connected unit/bus to the system. It shall also be possible to add and modify protocols/signals that shall be configured at what bus. This interface will take something as low-level as signals and visualize them for a developer to easily understand and configure.

In general when working with embedded systems and low-level configuration there are not a lot of GUIs use, it is mostly done via low-level coding. The ones that do exist are designed to have a lot of functions instead of having high usability, making it difficult for a novice user to create a quick configuration. This project would lead to research regarding this topic and what can be done to improve it.

Furthermore the goal of this thesis will be to generate new knowledge within the area. As previously mentioned designing graphical user interfaces to configure and visualize configuration of communication protocols or similar elements within embedded systems is not something that is well documented nor have clear guidelines to follow as a designer. There are neither documentation of what parts of the design process that should be especially considered when designing for embedded system. Based on this the research question: **"What should be considered when designing an interface for visualization and configuration of communication protocols withing embedded systems?"** will be the main one to answer throughout this thesis. The answer to this research question will be a set of guidelines to consider when designing an interface like this.

2

Background

In order to solve the issue of having different communication protocols talking with each other the idea is to develop a product that will be plugged into several different buses/nodes in an embedded system, it will receive several signal from one system, translates it correct and then passes it on until the receiving node. For example: if you want to use a Toyota battery in a Volvo car they have their own specific communion language. What this gateway will do is to serve as a translator.

The realisation of this Idea will be an in house development project of the company Infotiv that will now on be refereed to as the "Generic Gateway".

For this to work the gateway needs to be configured correctly. The gateway will be a small hardware device inserted for example in a car. Thereby a user wont be able to configure it in place, that is why a program together with a GUI is required to create the configuration and then load it over to the generic gateway. Since the device won't at this itteration have any wifi/bluetooth, opportunities it is important that the configuration is done on a computer and than loaded into the deviece via an USB

User research will be performed to see what is desired and what shall be considered when implementing a graphical user interface for this model. Since this GUI will be created from scratch their is a great importance of user research throughout the entire process as mentioned at [19].

2.1 Infotiv

Infotiv [20] is a consultancy firm based in Gothenburg with the motto "Instant Innovation". It is a firm that focuses on product development and consultancy within embedded system and the automotive industry. Besides consultancy Infotiv has their own product development department where this thesis is taking place. Several different projects have been/ is being carried out at the department, everything from communication protocol tools to autonomous go-kart.

As Infotiv describes on their website: "The technology has to function smoothly, but it's equally important that the people using that technology understand what they are doing. That exciting challenge – connecting people and technology – is something we tackle every day"[20].

2.2 Communication protocols

A communication protocol is according to [?] a protocol that bits are form after in order to paket the data it will transfer in a correct way. There are several different protocols that can be structured in different ways. The protocols that the generic gateway will handle is [?] [?] [?].

The CAN, Controller Area Network BUS was developed by Bosch in 1983 and has now become a standard use in cars but also in other products. A CAN message is structured in the following way:

- Data message
- Message request
- Error message
- Que message

The LIN protocol works in a different way then the CAN bus. It has a master salve system. It is built for features that are not time critical.

Automotive Ethernet is a more modern version of communication protocols in cars which provides faster translation sending then the previous mentioned.

The main feature of the product Generic Gateway is that it will enable communication between different communication protocols. The reason this is important is that different system uses different protocols[12] so in order to enable for more efficient developing different nodes can be used.

2.3 Desired features and existing products

Since this a new application that will be sketched/designed and developed from scratch there are not a substantial amount of similar programs. There are however some programs that have layouts/features similar to the one of this thesis. The desired features is that it should have a layout that enables simple file handling, creating a new, loading, saving and such which is reoccurring in a lot of different application.

Furthermore it should have a plain canvas that allows the user to start adding boxes to it and than further connecting them. The application should also have the opportunity to insert data/information that should be inside of the boxes/ECUs.

Finally the application should have the opportunity to export a config file that can be flashed inside of the generic gateway.

The applications that exists on the market that have the described features are Matlabs simulink, Vectors CANoe.

2.3.1 Mathworks Simulink

Mathworks Simulink is a software to build and simulate systems as described by themselves[16]. It is an extension from the very famous MatLab[14]. The software has a good interaction part where it is easy to visualize and configure complex systems. It also has a good interaction part for the user to work in hierarchies on different levels of the system to configure the different parts of the system. Some-

thing that will be useful to take inspiration on and perform cognitive walkthrough to make findings for this thesis.

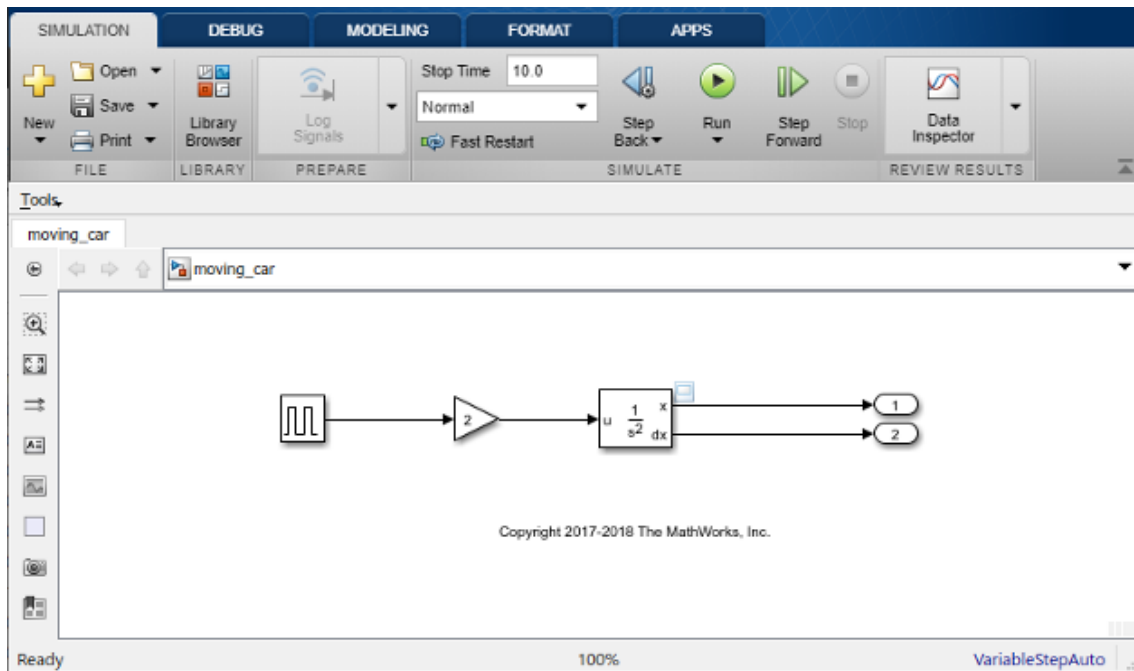


Figure 2.1: Screenshot of MathWorks Simulink showing it's canvas and basic features.

2.3.2 Vector CANoe

Vector CANoe is a software program that is used to simulate different ECUs[17] and configure them with the appropriate signals. This is something that is very similar to the desired application from Infotiv. Based on the analysis of the software it does not have the best user experience. It seems to focus more on the expert user rather than novice users. There are very few to any guidelines of how to start using the program and creating a configuration. Although it offers a great amount of tools and layout it does not present them in a convenient way.

2. Background

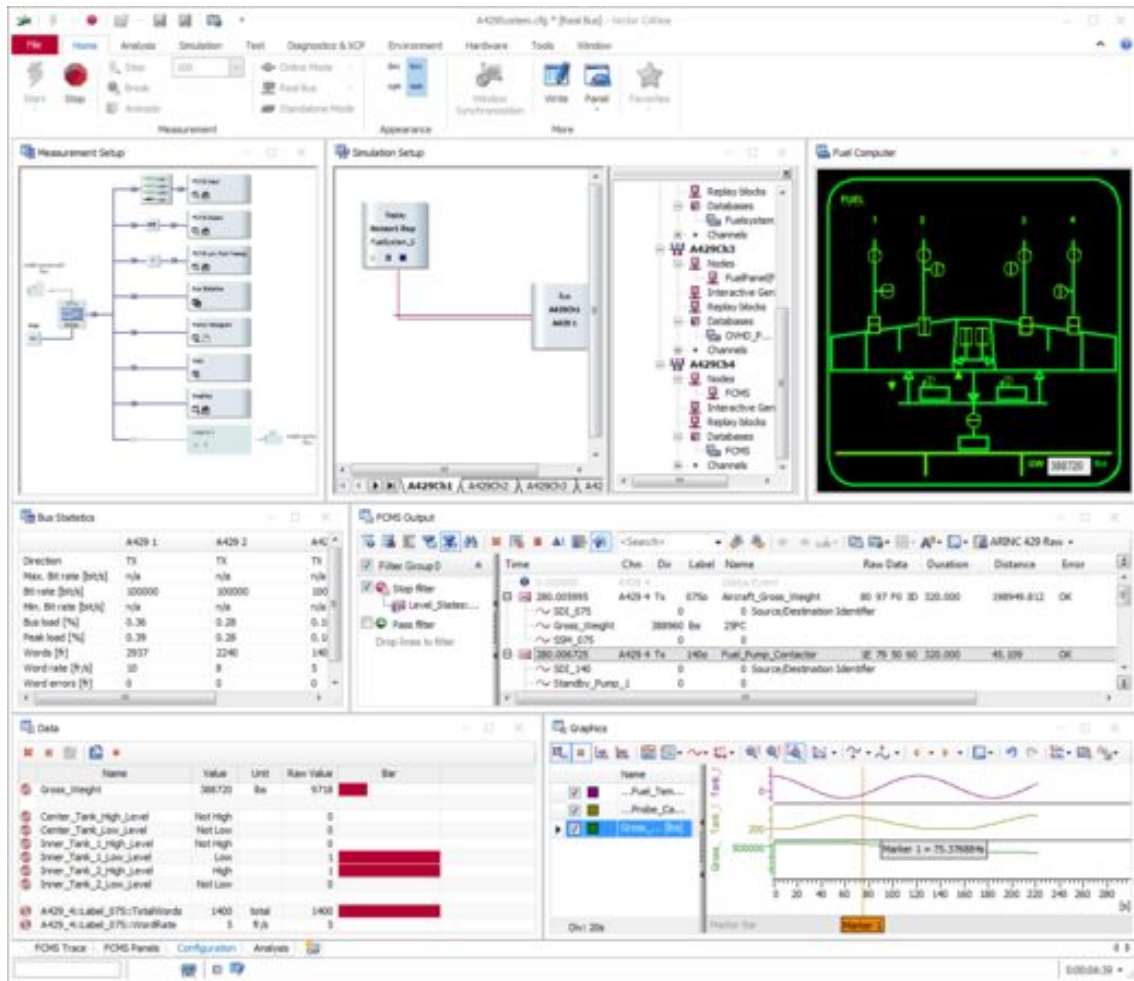


Figure 2.2: Screenshot of MathWorks Simulink showing a lot of its features.

3

Theory

This thesis will focus on collecting new guidelines for designing and developing an interface for configuring communication protocols within embedded systems. It will be done by investing the possibilities when developing the configuration interface for Infotiv's product generic gateway.

3.1 Visual programming

Visual programming is a way to program using graphical elements rather than writing it in text. A lot of visual programming languages shows data flows and more like diagrammatic programming.[11] This is something that will be especially fitting for this thesis project. As previously described there will be a lot of different arrows between the different busses/boxes in side of the gui so taking guidelines and principles from visual programming and applying them to the gui that will be created in this thesis will by highly applicable. One of the benefits of the more like visual programming look of the gui will be that it provides easier understanding for the user. Instead of configuration a lot of different signals in different buses the canvas gives a clear overview of the buses and the signals it contains.

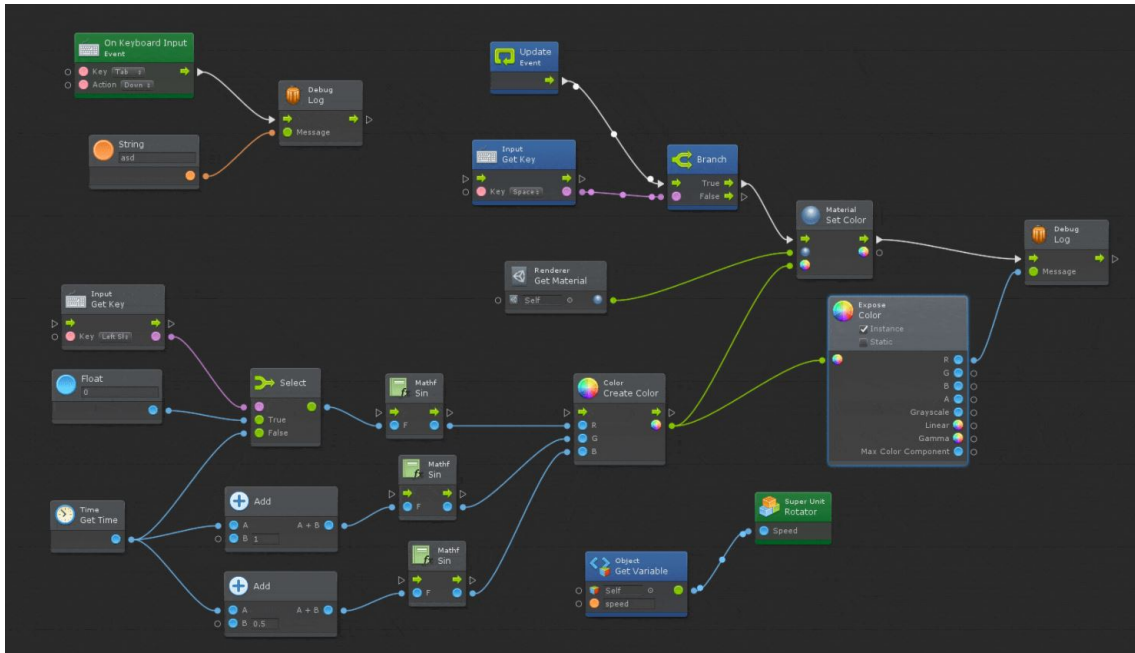


Figure 3.1: An example of visual programming, in this example Bolt

3.2 Information visualization

Another source of knowledge that will be used in the creation of the interface is how to visualize this information in the best possible way. Plenty of research of this area has been done previously and is summarized to the research area "Information Visualization".

Information visualization is described by [10] as the process of representing data in a meaningful way so that the user can understand it easy and correctly. Common examples of information visualization is line graphs and scatter plots.

The process of information visualization is usually started with a lot of qualitative research of the user to truly understand how and what data that should be represented for the user in the best possible way.

In this project the means of having interactive information visualization is extra important since the user both needs to see the visualized data and also have the ability to edit the values and connect the different buses.

By studying the knowledge regarding information visualization it will be of great help when designing the interface.

3.3 Design principles

When pursuing the task of designing the interface for the generic gateway it is of high importance to consider the existing design principles that will help when designing this user interface.

The goal of making use of these design principles is to strive for a easy-to-use, pleasurable design. As mentioned by [9] they are wide and not too specific. They are meant to have in mind when designing and to minimize the users cognitive load.

Some guidelines that will be used when designing the interface for the generic gateway is to **Don't make the user remember information** this is especially important when designing for embedded systems where the information/data might be of an abstract level and a lot of decisions when specifying the signals. It is therefore important that the gui will provide information about the user options.

Prevent errors if possible One requirement from Infotiv's side is that when before exporting the final config file there should be no room for errors that the user has configured. For example it should not be possible that the user has entered letters where it can only be numbers resulting in a syntax. So using the design principle to prevent errors if possible the design should be constructed in such a way that it is only possible to enter number where number can be entered and that only letters can be entered where there only should be letters.

Furthermore it is also important to follow the design principle **maintain consistent standards**. As previously mentioned the user will configure different buses and signals in the interface. The buses may differ in what type of bus it is and what communication protocol it uses. It is thereby extra important to keep to configuration box/tool consistent in its design while keeping it flexible so that users have the option to enter the desired data.

4

Methodology

This chapter will describe the methods and the process that can be used in order to create guidelines for developing an interface for embedded systems. The chapter will also describe various methods for structuring project work.

4.1 Wicked problems in design

Wicked problems is complex problems that may arise throughout design processes. It can be in terms of figuring out exactly where to start, how to get the specific requirements and also how to design and for who.

Wicked problems is characterized by not having a definite solution nor idea of where to start. These problems are common in design, such as not knowing when the design is finished. There also might be hard to test the design and know that it verifies the solution as described by the article [18]

Even though it may seem though to tackle wicked problems and reach a solution there are different methodologies such as Agile working. There is also solutions to be found in design thinking/system thinking.

The article also suggests that combining systems thinking together with agile working it is more likely to see progress within the project. They describe systems thinking as a way of understanding the different components within a complex system and how the different parts effect each other.

They describe the following five steps in order to face a wicked problem:

- 1. Break down information into nodes and links.
- 2. Visualize the information.
- 3. Collaborate and include stakeholders in the process.
- 4. Release solutions quickly to gather continuous feedback.
- 5. Carry out multiple iterations.

All these points has great benefits when used both in terms of product development but also the process for generating the result.

4.2 Research through design

Research through design is according to [15] the principle of balancing research and design, even though they might seem different in their goals they are more similar than expected.

Research is often regarded as the act of creating new knowledge that can be used in other projects and areas, where design is considered as a specific solution to a specific problem.

They are however similar in the sense that they both work in an exploratory phase where there are no concrete solution nor answer to a problem, instead they generate suggestions and hypothesis to the problem.

Research through design is often used to answer wicked problems.

The method is beneficial because the evaluations of the research will provide a better insight for the practical work when designing.

4.3 Agile methodology

Agile working is a methodology for working within projects, most often with a method called SCRUM. It is structured down in a time plan, with various different time goals and what should be delivered at those time frames. First off the methodology is planned in and for teams that work in a product. According to [13] the most common roles is to have a *Scrum master* which is the expert and advisor that makes sure the team is on track and that everything works smoothly within the framework. In the case of this thesis the supervisor at Infotiv Olle Norerlius will act as the Scrum master that helps out throughout this project.

Futhermore the team usually consists of a *Product owner*, a person that priritzes what work and deliveries are the most crucial and what should be delivered when.

In the case of this thesis it will be the product owner Pierre Ekwall at Infotiv.

Finally the SCRUM team consists of the team members who do the work which in this case is me together with the other workers at Infotiv.

Although Infotivs development reminds a lot of the classic SCRUM it differs a bit in terminolgy and daily stand ups. What is important for this thesis is Infotiv's equivalent to sprint which they call gates.

Gates is Infotiv's way of describing what should be delivered when, it has some element of dynamic adapting to it, meaning that the deadlines may be flexible, depending on the circumstances of when it should be delivered. The gates that has been set out for this project is:

Gate	Due
<i>Project description approval</i>	V ₄₅
<i>Present Hardware and software architecture and simple software functionality demonstration of how the product will work</i>	V ₄₇
<i>Create concept prototype</i>	V ₅₀
<i>Demonstrate value change in signal database while sending potentiometer value over CAN bus to the gateway</i>	
<i>Demonstrate that the prototype of the interface can read user signal definitions</i>	
<i>Demonstrate how the change of diode strength correlates with the potentiometer when the diode is connected with a CAN bus.</i>	V ₃
<i>Demonstrate that the prototype of the interface can read user frame definitions</i>	V ₆
<i>Demonstrate how the change of a value when you receive the data through ethernet.</i>	
<i>Demonstrate how the change of diode strength correlates with the potentiometer when the diode is connected with a LIN bus.</i>	V ₉
<i>End of project</i>	V ₁₁
<i>Demonstrate that the prototype can read excel file to populate user interface</i>	V ₂₆
<i>Present current dataflow in graphical interface</i>	V ₂₈
<i>End of project</i>	V ₃₀

Figure 4.1: Infotiv's gate for the Generic gateway

The text marked in blue is the deliverers this thesis throughout the project. The first delivery, to create a concept prototype is to sketch out, create a MVP of the interface, the different elements it will contain together with a prototype.

In the next gate Week 50 a demonstration of how the interface can read signal definitions from the database it contains. This is implying that there should be

4.4 Double diamond

Double diamond is a design method used to make sure the iteration of a design project is performed correctly. [8] states that the design should start of in an iterative research with the goal to have a revised problem statement when done.

Once that is completed the second states starts which is the design state which is also an iterative process with the goal of producing a minimum viable product for each iteration.

The following picture describes the method accurately:

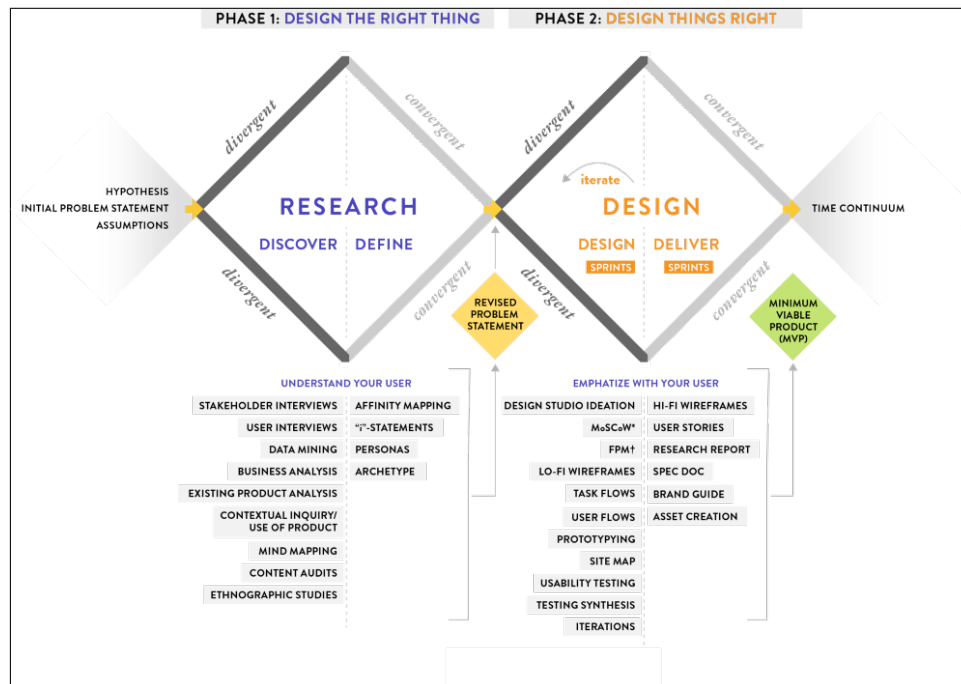


Figure 4.2: [8], graph for explaining double diamond.

This method will have multiple benefits, there among it being inclusive in making all necessary parts included in the iterative process. It will also have the benefit of providing a clear problem statement before going into the design phase.

It does emphasise a lot on the iterative process which is used in the agile framework. This might lead to some clashes of how well the process can be iterated versus the expected outcomes at each sprint/gate.

As previously stated the method contains different phases:

4.4.1 Discover

Methods that are suiting in this phase is:

4.4.1.1 Personas

Personas are fictional characters that should simulate a typical user of the design/the product that is being developed [7]. This is done in order to help the designer further understand the user and the users different ambition, goal and mindset.

There are several different pointers to keep in mind when developing personas in order to create them efficient and use them throughout the iterative design process. Some of the pointers is to have a good amount of data on the users before starting to create a persona.

It is also important to include several persons in the process of developing a persona such as stakeholders. When creating a persona it is beneficial to have an image and describe the personas everyday life together with their interest.

4.4.1.2 User interviews

User interviews are used in an early stage of the design process. It is used in the *ideation phase* where the goal is to understand the user and hers or his needs.

According to [5] user interviews are one of the most common methods in order to get to know the user. The interviews helps the designer to get to know the user in a lot of different ways. The interview will hopefully generate information regarding the users feelings, motivations and daily routines. This information is of great help when moving forward in the design process to design for the user.

As oppose to questionnaire where the information generated is more quantitative, which might be beneficial for some goals the qualitative information gathered from the user interview helps the designer to get to know the user even more.

4.4.1.3 Comparison of methods

Even though the method of creating Personas or conducting user interviews might seem quite similar in the sense that they both have the goal to understand the user more they are still a lot different and come with their own pros and cons.

The benefit of spending a lot of time on personas will provide a tool/"person" that is accessible throughout the entire design process, compared to an interview where the person that is being interviewed will most likely only be available once.

The personas are however fictional which is not as close to the real end user. It does not mean that the method can not go well together. In multiple cases user interviews are conducted to then further proceed with creating the persona and thereby leading the designer with real proof of how to simulate the user when creating the persona.

4.4.2 Define

4.4.2.1 Affinity Diagramming

Affinity diagrams is a method used in order to make sense of a lot of data when designing according to [4]. The process can take several inputs such as facts, ideas, user needs and insights. The main goal of the method is to group different parts of the information together and summarize it.

In order to perform an affinity diagram the designer starts with writing the different pieces of information/ideas/facts on a post-it note and putting it into one category on a e.g whiteboard. Then the user continues with the second piece and so on and so on.

Once all the different pieces are put in a cluster/category a tradeoff, a ranking can be made to see which group has the most important set of data. There can even be connections between the different parts of clusters.

4.4.2.2 Weighted Matrix

A weighted matrix is yet another method for defining the different priorities in a design corresponding to different criteria. It is performed by setting up different criterias as a colum in a matrix and then to add each different set of design choice next to it.

Each criteria has a *weight* that is being counted for when summarizing the different points in the matrix. A rating of the different design choices is also done. Once the points have been put out a summary is performed to see which design choices that scored the highest.

According to [6] the weighted matrix is a simple tool that can be used when decisions have to be made, especially when there are multiple choices and multiple criteria to consider.

4.4.2.3 Comparison of methods

Affinity diagramming and weighted matrix have some similarities of breaking down information to different areas and put a point or a number on them. They are however used in different phases of the design process. The affinity diagramming is rather used in an earlier stage of the design process where parts of the ideation phase might still be going, compared to weighted matrix that is rather used when there are going to be decisions done about which design to chose.

4.4.3 Design

..

4.4.3.1 Cognitive walk through

Cognitive walk through is one of several evaluation methods used for generating findings when designing.

As described by [3] it is a usability inspection that links the interface to a cognitive model. The method is used in the way that the evaluator performs the typical user tasks and evaluates the systems responses. Once the evaluator gets a response from the system a comparison is made from what the users goal was and what the systems response is. When performing a cognitive walk through these are the four best question to ask yourself as a tester according to [2]

- Will the user try and achieve the right outcome?
- Will the user notice that the correct action is available to them?
- Will the user associate the correct action with the outcome they expect to achieve?
- If the correct action is performed; will the user see that progress is being made towards their intended outcome?

4.4.3.2 Heuristic evaluation

Heuristic evaluation is a usability method for finding usability problems in an interface so they can be worked on in the iterative process according to [1].

It might be hard to perform a heuristic method by only one designer, it has been shown if multiple people regard the same heuristics for the same product/design, the like hood of generating more findings is larger.

There are previous set of fixed heuristics that might be used but there is however fully possible to create custom heuristics to regard for the design.

4.4.3.3 Comparison of methods

Both methods are used in order to find problems or areas of improvement for one design. A cognitive walk through can be quite demanding and take a lot of time to perform especially if it shall be performed by multiple people to get several perspectives. A heuristic evaluation might be a little bit easier to perform because it has the set case of heuristics for the evaluator to look open and opens up for a discussion within the team.

5

Planning

This chapter will state the various goal of this thesis together with how to reach them. It will go through the time plan together with the scope and methods. The chapter will also note which methods that will be used and where.

The main work with the generic gateway will be to develop the configuration interface. The configuration interface will import a database of signals which the user then has the option to edit and select. Once the signals are selected they will appear in a box in the empty canvas simulating a bus.

Once the bus is created the user has the option to drag which signals from which bus that will go to which bus. This will then be visually confirmed to the user by an arrow added between the two busses/boxes in the canvas.

When the user are done configuring the different buses with the different signals the user has the option to export the configuration file which is the goal of the interface. If there are any syntax errors such as there are letters where there only should be a numbers the application will show where and why there is an error. If there are no syntax error the configuration file is generated.

Advanced features

The previous text described the main features of the application/interface. The application shall also have support for more advanced features. By advanced features it means to support logic in between the signals, such as conditional, combination and scaling. For example it should be able for the user to select two signals and insert a gate/block in between them which should result in a third new signal is added which represents the addition of it. Similar features should be added when an if statement is included. If both these signals are true then they should present the following output. The scaling feature is to say how many decimals the signal value should have.

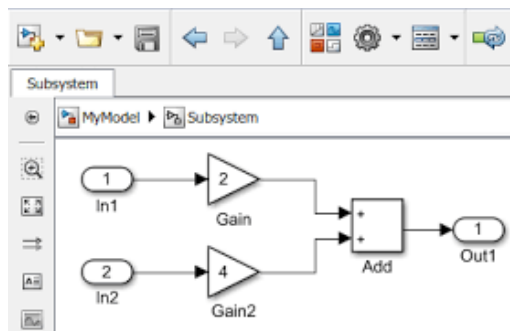


Figure 5.1: An example of logic within signal handling, in this case in Simulink

A lot of these advance features is similar to *visual programming*.

5.1 Solving wicked problems

As previously mentioned pursuing this thesis and generating the guidelines is a type of a wicked problem. In order to solve a wicked problem there are steps to follow in order to have the best conditions to tackle the problem.

As mentioned in the methodology chapter [18] describes the following five steps in order to face a wicked problem:

- 1. Break down information into nodes and links.
- 2. Visualize the information.
- 3. Collaborate and include stakeholders in the process.
- 4. Release solutions quickly to gather continuous feedback.
- 5. Carry out multiple iterations.

The first step will be utilized in the first phases of this thesis when ideating and understanding the different requirements and needs for the interface that will be created. The interface has a requirement by Infotiv to be able to handle a lot of data of various communication protocols. These protocols and their information may be overwhelming to interpret and mainly how to display their data in a understanding and nice way for the user. Working with breaking down the information and realizing a good representation of them will likely lead to new guidelines of how this will be done in the best possible way.

The second step of the suggest methods of working with a wicked problem, "Visualize the information" is what the interface main goal is to do so this step will be throuly followed and as mentioned previously it will also generate new guidelines when desiging an interface for an area as embedded as this, how to really grasp the protocolls and how to work with them.

Collaborating and including stakeholders in the process will defiantly be used throughout the thesis as well. It is decided within the agreement with Infotiv the daily stand ups, weekly reporting as well as presenting and gates. This reporting will give qualitative insights of my development which will further generate more guidelines to answer the research question.

The final steps will be included in the iterations of the design together with the gates and releases there. Throughout each iteration and the quick releases there will also be delivered new guidelines for this thesis.

5.2 Infotiv's project planning

Infotiv's way of working with product development is to specify a product description as well as a technical specification for the product/project before the project starts of, much similar as a planning report for a master thesis. The product description is a document that specifies the overview, deliveries, scope, limitations, requirements and time plan. The purpose of this document is that it will provide a good overview of the project as well the product owner at Infotiv has a document to keep track of their thesis workers. The other document that has been written

for this project is the Technical specification. This document provides more the details of the project. It describes the system, software, hardware as well as details of the necessary components and their costs. In terms of planning the technical specification also provides specifications of how to validate the functionality of the product/interface.

Requirements
#001: The user interface shall generate a configuration/excel file
#006: The user interface should have the ability to import databases containing signals for the different bus nodes in the system.
#007: The user interface should contain options for specifying details to the signals, e.g type, length, data.
#008: The user shall be able to read out a calculated delay for the signals
#019: The GGW shall never exceed 70 % processor capacity on CPU and RAM
#020: The user interface should present the frequency of the signals that are being sent and received.
#022: The GGW shall have diodes that indicates if the transceivers are transmitting or receiving frames.
#023: The GGW shall contain a button to load and run the new config file from the USB memory to the GGW.
#024: The user interface should run checks for the defined signals and report syntax errors for the user and not export a file containing errors.
#025: The user interface should be expandable to be able to insert more in, outputs and new communication protocols in the future.
#026: The user interface should support advanced signal logic (combinational, conditional and scaling signal logic).

Figure 5.2: The requirements decided with Infotiv for the project

In order to deliver the following requirements it is very important to work *agile* in order to meet the criteria.

5.3 Scope

The main goal of this project is to create a graphical user interface for Infotiv to visualize and configure the generic gateway. It might not be in high fidelity and have all features implemented.

More specifically the features that innovative requires can be seen in ???. To summarize it should be an interface with a good design that has the feature to import a

database from an excel sheet. Select a set of signals, have the option to edit them. It should also have the feature to export a new excel sheet that contains the required protocols with the correct protocols that can further be ran in the generic gateway. It is within the code to perform functioning software for Infotiv.

The scope for the thesis is also to generate guidelines for designing an interface for an embedded system. There will be an evaluation done for the existing guidelines within the field and then throughout the thesis, for each iteration produce new guidelines that will help designers fulfill the task of designing an interface for embedded systems.

5.4 Process

Throughout this thesis we will work in "Gates" which is Infotiv's equivalent to sprints etc. At these sprints new guidelines will be added/revised to continues answer the research question. A Gannt schema will be created in order to structure up the different phases as well as set the requirements for this project. The Interaction design process will be used in the sense that ideation is started several sketches is made, user research is done and so on, having "Research through design" in back of the mind [15] In terms of report writing an estimate of 2 days per week will be made.

Time Schedule:

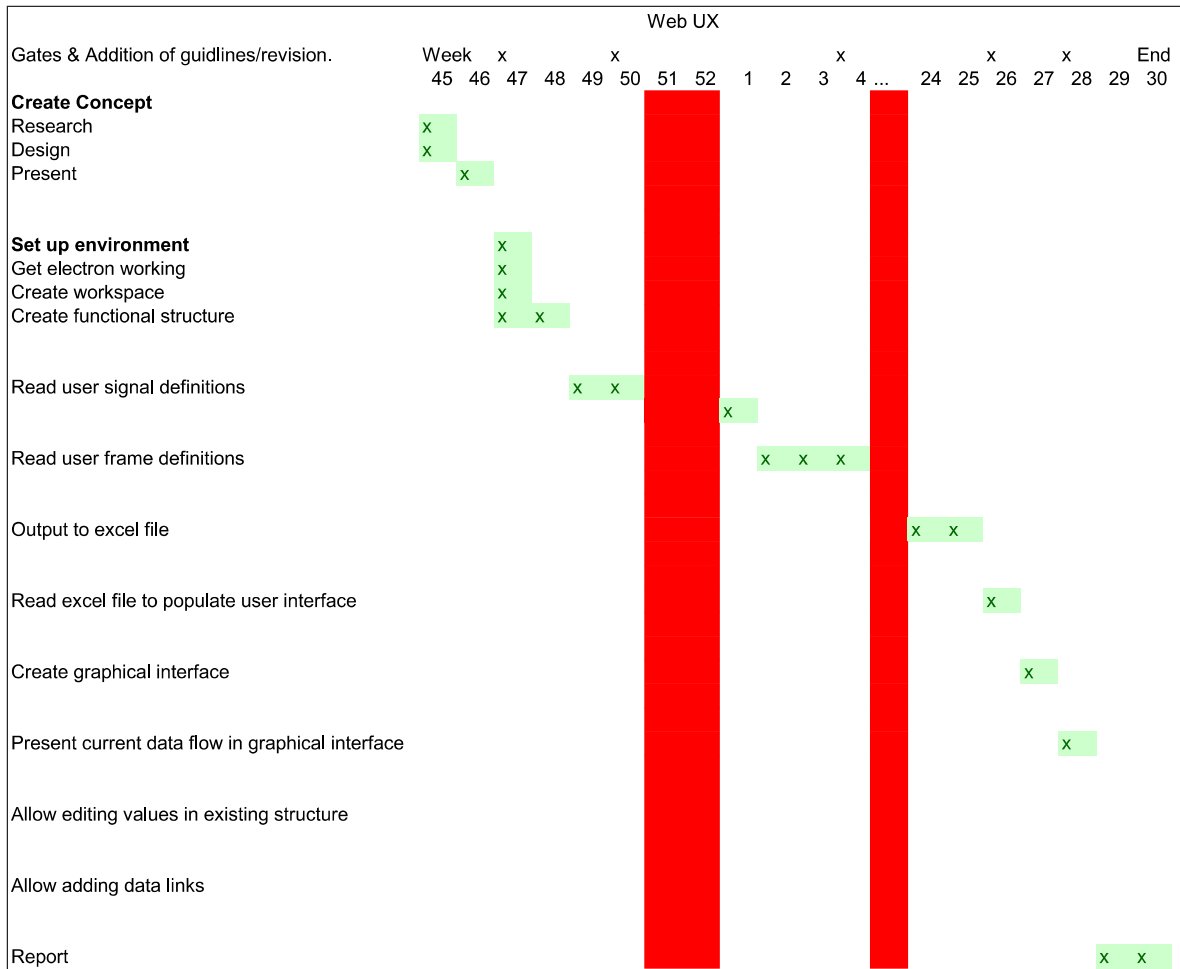


Figure 5.3: Infotiv's gate for the Generic gateway (2020/2021)

The presentation is aimed to be held week 30-34 based on when opponents/examiners are available.

5.5 Ethical considerations

In this thesis there will be several ethical aspects to consider. In general when designing a graphical user interface you have to make sure that it pleases all users no matter their abilities. For example people who are colorblind should be able to use it.

There will also be a consideration between what Infotiv is requesting in their design compared to what the user finds easy and understandable. Fulfilling Infotiv's request regarding what features that shall be implemented while still designing for the best possible usability. In order to handle this, user research, storyboards and user stories will be done to make sure that the design will be understandable by all users.

This tool will as previously mentioned be used in order to translate different communication protocols. In most of the cases it will be used as a translator and configured in cars. When sending communication protocols with data it could possible also lead to a wrong signal being sent that could lead to financial damage or personal damage

5. Planning

to the person developing.

Since this version of the Generic Gateway will be designed for development purposes there wont be any hard cases but it will still be considered.

6

Execution and Process

Throughout the thesis I followed the previous mentioned time plan (Figure 5.3). Even though I had the plan to follow some parts were handled in another way and at a different time in the thesis. In this chapter I will present how the thesis unfolded.

6.1 Prestudies

Throughout my early research I found very little to none existing guidelines on how to design an interface for embedded system. So what I did was that I analyzed existing interfaces for configuration of embedded systems to find design patterns. To start designing an interface or generate guidelines there was a need for understanding what exactly was desired first. The interface that was going to be designed needed to have:

- A clear instruction for the developer where to start.
- How to import different signal databases.
- Select and modify the desired signals and frames.
- How to export the configuration.

So first off how did existing products do this? How intuitive were there design? One of the design that are previously mentioned in 2.3.2 is Vector CANoe here we had an interface that did not have any convenient guides, hints nor a wizard for the user when the user were going to start using the software.

The interface also feels quite cluttered with a lot of different elements and once the user get's going with the configuration the more and more the interface get's crammed with a lot of different windows and elements as seen in the following picture:

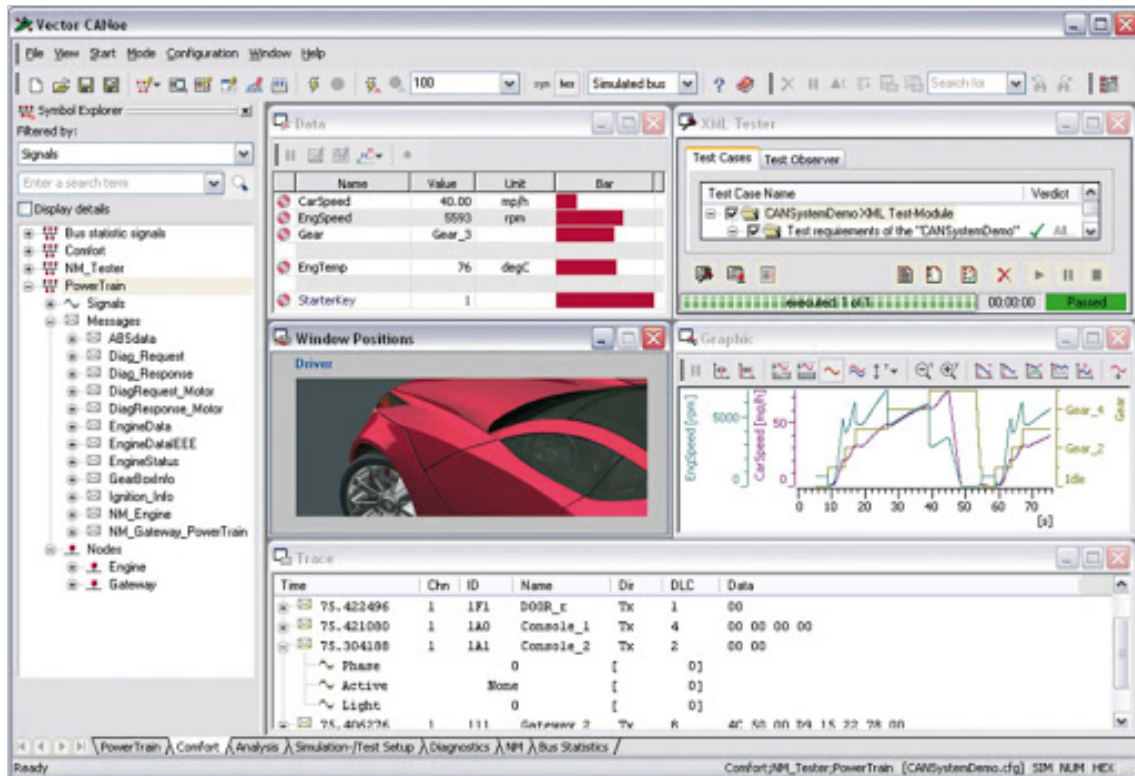


Figure 6.1: An overview of an ongoing configuration in Vector CANoe

So one of the guidelines/findings was too **Do not make the interface to cluttered**. This was going to be considered through out all of the iteration and different prototypes of the design. Furthermore in the analysis of Vector CANoe there was found some what a lack of consistency in how the different windows displayed. Even though the different windows had different purposes and functions it was not trivial how to use them in the correct way and by setting the windows to the same size the user could easily mixed them up with some other window. So thereby the finding was made that **Keep consistency in the design while clearly communicating to the user what part of the GUI that are for what function**. Another finding that were made though out the analysis was the there were a lot of interrupting windows when working on the configuration. **Keeping the user uninterrupted while still hinting about potential errors** was thereby going to be considered when starting sketching the different prototypes. Also in the error area another finding was that some of the error warnings that should warn the user to late in the configuration process. So antoher finding that was found is that **Hinting the user in time to prevent potential errors**

6.2 Change of planning

There were some changes to be done throughout the process. In the configuration interface the idea was first to have a lot of advanced features that would be built up with different blocks similar to visual programming as described in 3.1. Throughout the project these features were if there were time and as the project went on we

prioritized the configuration rather than these features. The research regarding visual programming did however benefit the thesis a lot in terms of how to visualize the different ECUs and how they were connected to one and other.

Another part that was added to the project from Infotiv's side was the feature of the GGW to have the ability to read a new config file from a usb-memory and reflash the hardware so it ran the new config-file. This meant more work with the hardware and also had to be done in a smooth way for the user so it worked in a convenient way to reflash the ggw.

6.3 First iteration

The different iterations though out the projects were divided by Infotiv's so called gates which have been described in previous chapters.

The first gate was to create concept prototype. In order to create the concept prototype I had to start with sketching. In order to sketch I had to follow the findings and specification of the interface. There were multiple ways to perform the initial sketches but I started with paper sketches and tried several different versions. Once I had a couple of sketches I compared the different ones and graded them in what features that were good and what ideas that needed to worked more on. Then the best features from each sketch were extracted and put in to one final sketch which were presented to the company.

One of the **findings** were that the user had to name the ECU when importing the signal data base. This were most easily done with a prompt window asking the user to type in the name

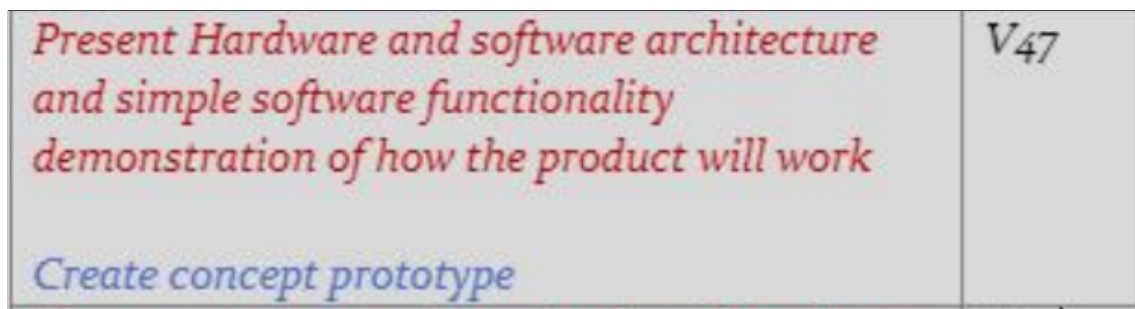


Figure 6.2: Gate 1 (Blue text is my delivery)

As of all the gates that are being held at Infotiv they end with a presentation. So towards the end of this iteration I booked a meeting with the supervisor, product owner and presented my design as well as findings. After some questions they liked the sketch and wanted me to continue on it going from low fidelity to a more high fidelity prototype.

I also wanted to gather more input regarding my design so I performed small interviews with the employees that have worked with Vector Canoe and similar tools to configure communication protocols. Some were novice users and some were expert users. They all pointed out different inter sting findings that bothered them in the current design.

The current guidelines that have arisen at this iteration was.

- Let the user name the different elements of the configuration
- Keep a minimalistic version of the gui and do not make it to cluttered for the user.
- Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.

6.3.1 Change of planning for this sprint

When starting this iteration the goal/delivery was to "Create concept prototype" which did not have a clear statement of what that prototype would be, the most logical start is to start with sketches which I did but Infotiv had somewhat more of a push towards starting with the coding of it all. The parts of the coding that was set up in this was deciding which Framework that was going to be used as well as starting a github to store all the future code that was going to be done for the project.

6.4 Second iteration

Moving to the second iteration of the project was to move from lofi-sketches to more high-fidelity sketches. In order to do this I had to choose which program to use to create the sketches, I evaluated Figma and Adobe XD and decided to proceed with Figma since I was familiar with the tool and the shortcuts within it.

6.4.1 High fidelity prototype

There was also of interest to work in the findings/evaluation from the first iteration onto the second one in this project so that was done.

The design I started with in Figma had the basic layout from a previous project but I modified it to turn into the one desired by Infotiv.

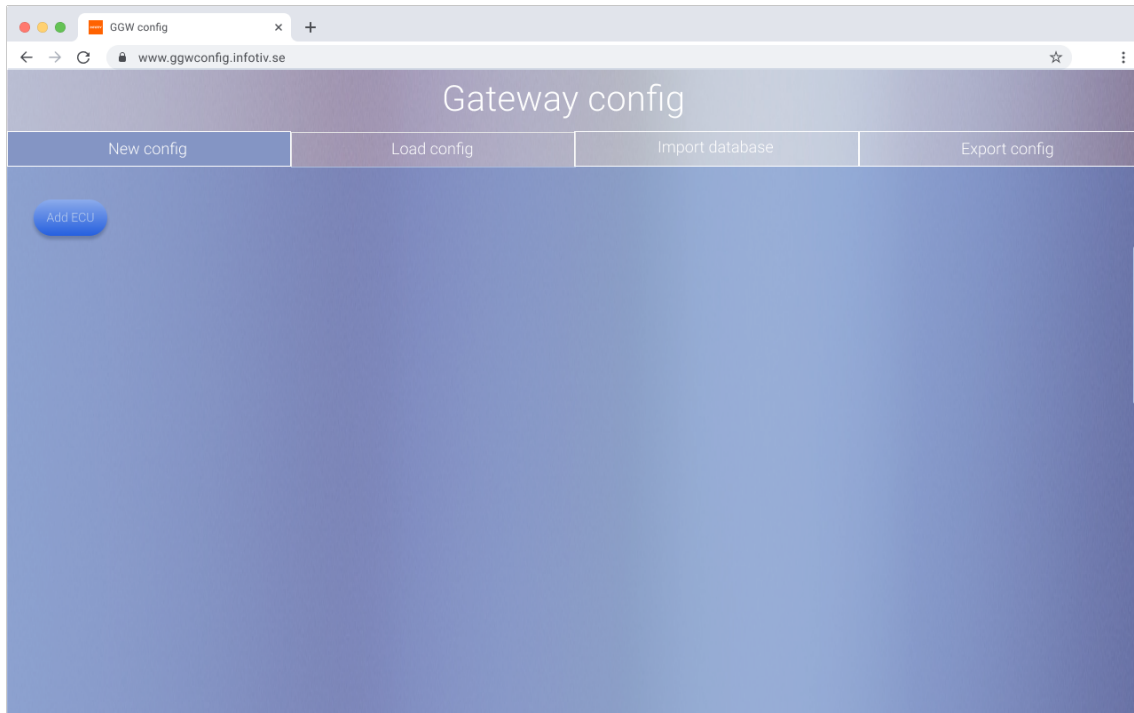


Figure 6.3: Starting screen, High fidelity prototype.

This was the starting screen created for the high fidelity prototype. What we can see in this design is a clean interface with different tabs and one button "Add ECU". This is a big contrast to Vector Canoe previously mentioned. This is way less cluttered which was one of the findings that were made in the previous one. This design also kept a good simple color that works well together for a pleasant viewing experience of the interface. In this screen the user has the opportunity to create a new config, meaning to add new ecu blocks to the canvas and start the configuration. The other tab Load configuration is if the user want to modify an already existing configuration of ECUs. Import database was in this state the same as "Add ECU" the only difference being where to put it and what to label it. Finally the tab "Export config" is when the user are done with the configuration and wants to export the configuration.

6. Execution and Process

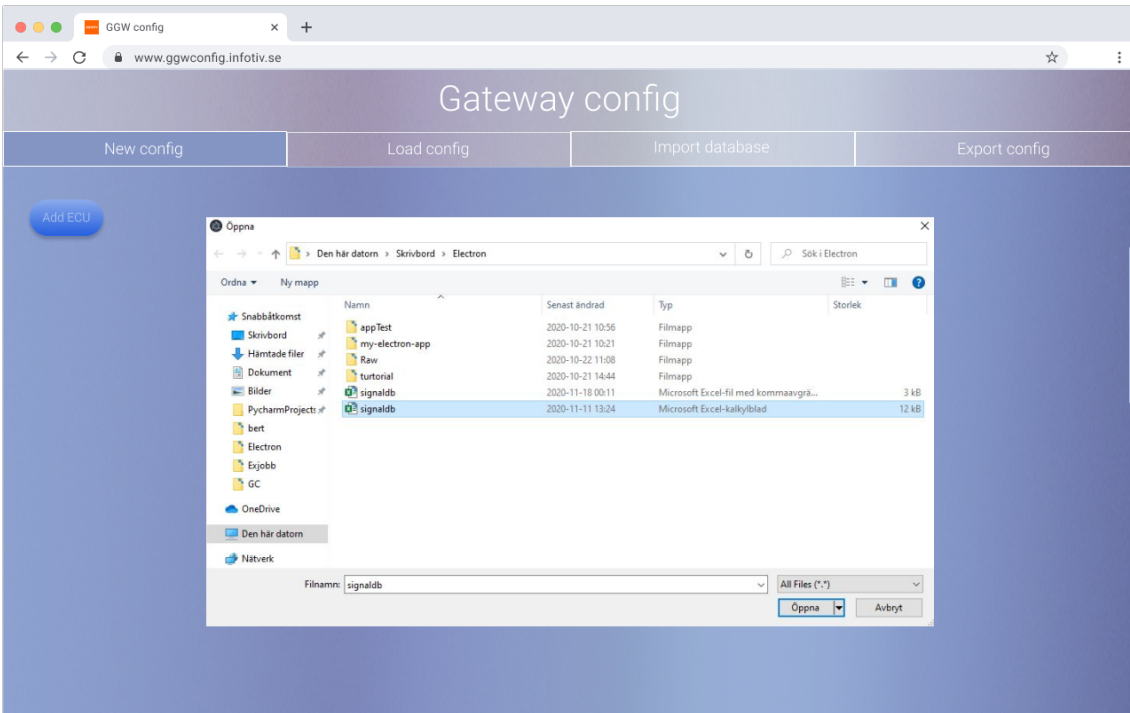


Figure 6.4: Add ECU/Import database screen, High fidelity prototype.

Here the user get's prompted with a message box asking the user to select a file which shall be the file containing the signals for the database to create the configuration out of. This is linked to a Windows classic open prompt which the user easily will recognize and know what to do.

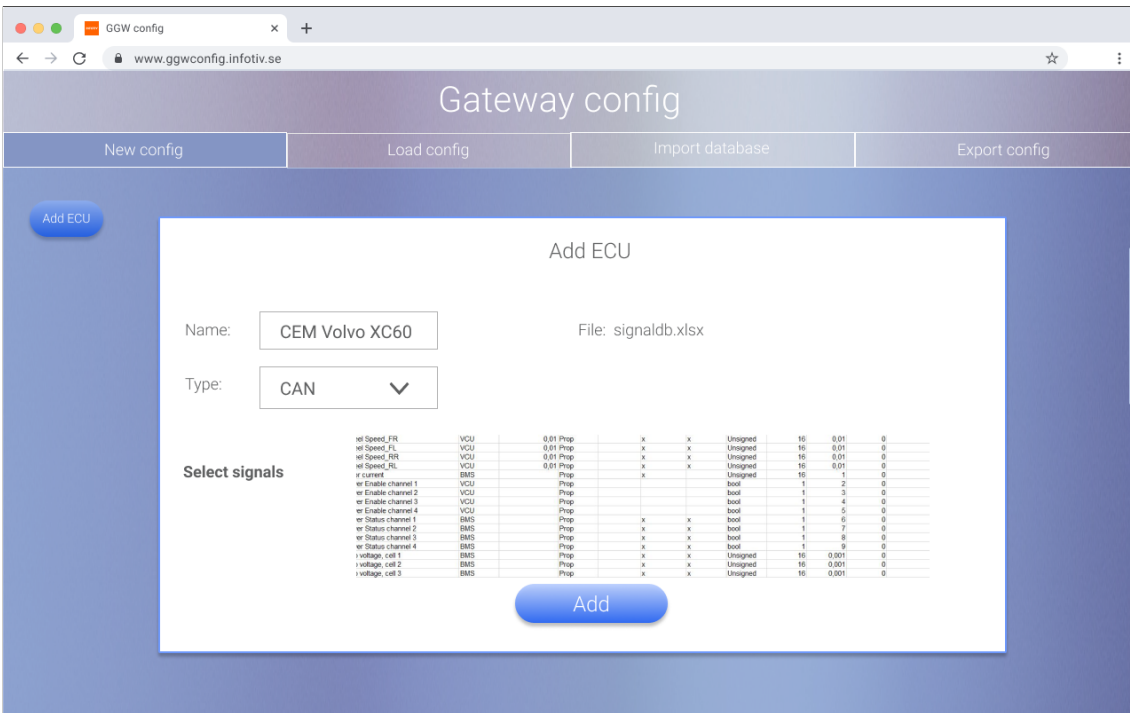


Figure 6.5: Imported database screen, High fidelity prototype.

In this screen the imported signal database can be seen as well as the signals it contains. In this screen the user now has the opportunity to name the ECU in order to keep better track which was a finding that was made in the previous iteration. It is also possible for the user to choose which type of communication protocol that shall be configured, CAN, LIN or automotive Ethernet. Note this is an early stage of the design so exactly how the signals were going to be chosen from the list was not 100% decided on. The pop up window also provides a title "Add ECU" to show to the user what button that has been pressed. It also shows which file (signaldb.xlsx) so the user get's feedback from the previously selected files in the Windows prompt.

The screenshot shows a web browser window with the URL `www.ggwconfig.infotiv.se`. The main page is titled "Gateway config" and has four tabs: "New config", "Load config", "Import database", and "Export config". A blue "Add ECU" button is visible on the left. A modal window titled "Add ECU" is open, showing the following fields:

- Name:
- File: `signaldb.xlsx`
- Type:

Below these fields is a table with the following data:

	Signal name	Source	Tx Period	Source bus	Gate to prop	Gate to safety	Gate to info	Date type	Length
Signal 1:	Wheel Speed_FR	VCU	0,01	Prop		x	x	Unsigned	16
Signal 2:	Wheel Speed_FL	VCU	0,01	Prop		x	x	Unsigned	16
Signal 3:	Wheel Speed_RR	VCU	0,01	Prop		x	x	Unsigned	16

At the bottom of the modal is a blue "Add" button.

Figure 6.6: Modify signal screen, High fidelity prototype.

This is the screen that is going to present the different selected signals that the user has chosen. The main purpose of this screen is to let the user modify the selected signals, add different attributes and/or modify the selected ones. The pop up window also hints for the user that the different boxes in the table is editable by providing a different text and drop shadow.

6. Execution and Process

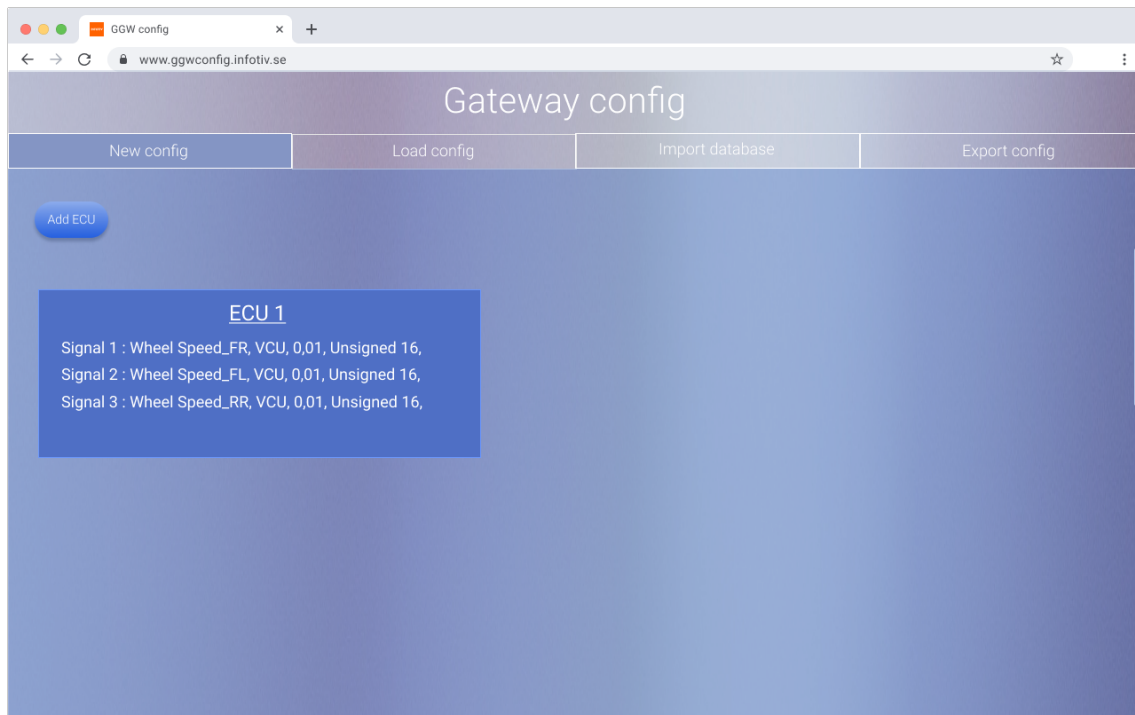


Figure 6.7: Added ECU screen, High fidelity prototype.

This is the screen that presents the added ECU, now the user has performed the configuration and the ECU is added as a box to the previous blank canvas. The box displays the configured signals as well as the title of the ECU. The title is also underlined in case the user wants to modify it again.

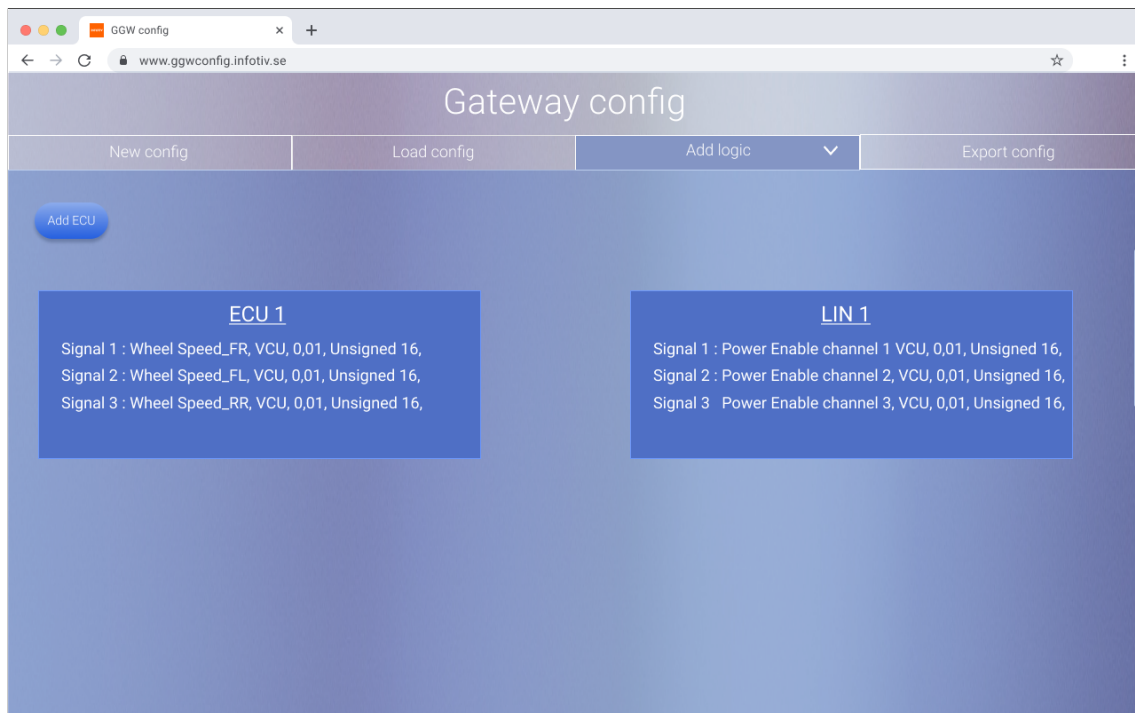


Figure 6.8: Two added ECU, start of adding logic, High fidelity prototype.

This screen presents a layout in the canvas where the user has added 2 ECUs and is starting to add logic blocks to the canvas. Here the third tab "Add logic" has been chosen and is ready to start and configure the different logic blocks between the ECUs

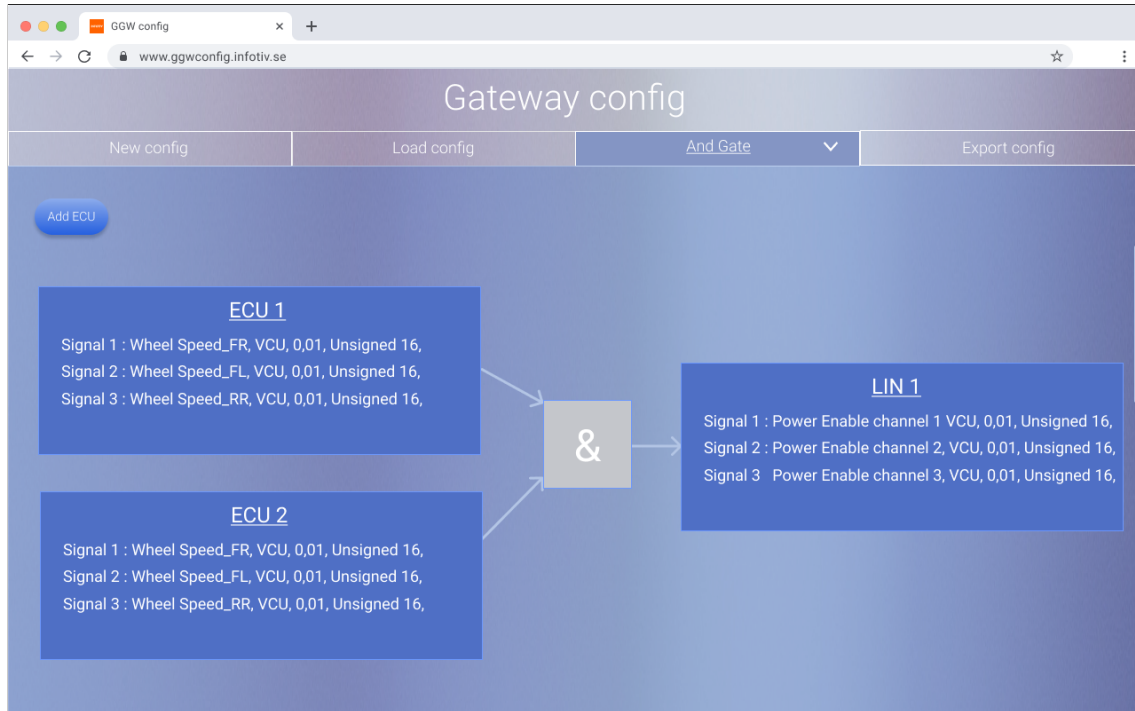


Figure 6.9: Configured AND logic block, start of adding logic, High fidelity prototype.

Here we now see a completed configuration between 3 different ECUs, what this is saying is that an AND block is between ECU 1, ECU 2 and LIN 1, meaning that two signals from ECU 1 and ECU 2 must be 1 in order to go to LIN 1

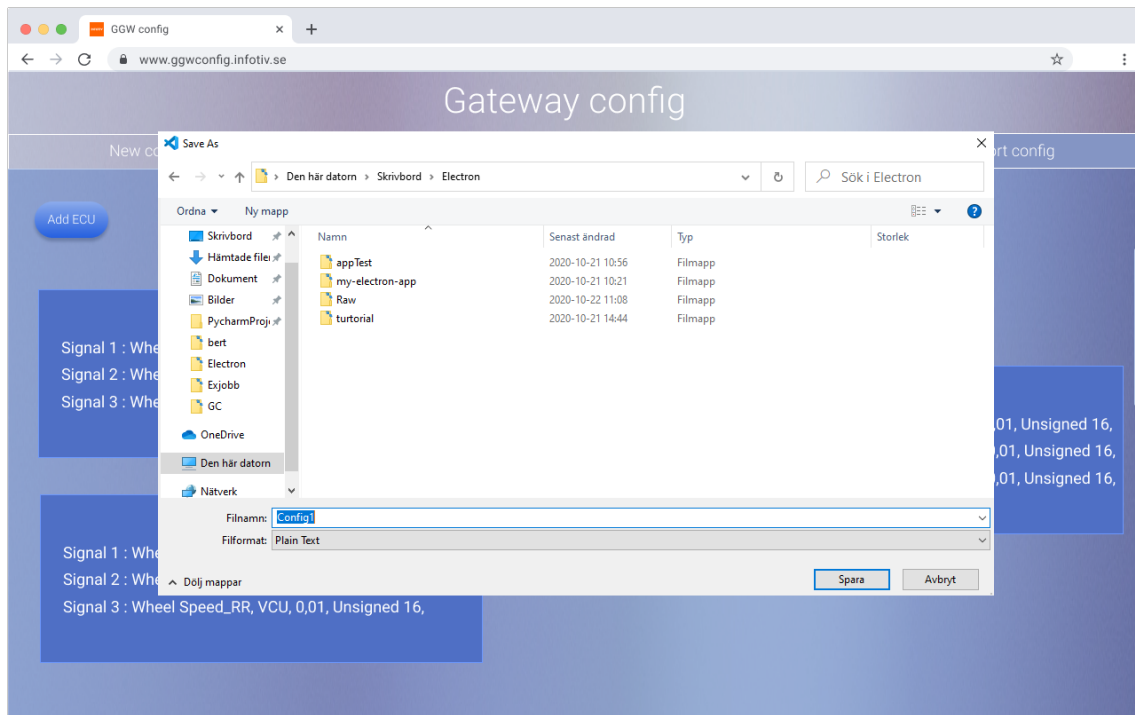


Figure 6.10: Export Screen, High fidelity prototype.

This is the final screen of the High fidelity prototype. This screen shows that the user is done with the configuration and is ready to export the screen. The tab "Export config" is chosen and the user is once again prompted with a windows window to choose where to save the file and what to name the config.

6.4.2 Evaluation of high fidelity prototype

Once the high fidelity prototype was designed it was once again time to present it to the different stakeholders. This was done in one of Infotive's gates. Here I got qualitative feedback on the design. They pointed out that it was not super clear to see a designed difference between what was a tab and what was a button which is important when designing an interface, especially when designing an interface. Overall the information from these stakeholders provided great knowledge of my design and how to continue to improve it.

There was also a remark on the first screen of imported data base. That the signals looked very mushy and a bit too small for the table it self. The qualitative feedback that it was not super clear to see what fields that were editable or not was not was also given.

Furthermore to get more feedback on my review I let some other employee/users review the interface as well as try out the prototype while performing a so called *Think-a-loud protocol*. They also claimed that the colors were very similar to one and other and that they should be more separated to know what exactly was for which department. The think-a-loud protocol was performed by asking the users to speak of anything that came to mind when they were trying out the prototype. The guidelines that now were generated after two iterations were: The current

guidelines that have arisen at this iteration was.

- Let the user name the different elements of the configuration
- Keep a minimalistic version of the gui and do not make it to cluttered for the user.
- Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.
- Make sure to involve all product owners and testers throughout the project
- Keep a good color combination with one primary and one secondary color that is distinguishable from each other.
- Keep the interface expandable for bigger configurations.
- Communicate in good time if there will be eventually cuts from the design compared to the initial plan.

6.5 Third iteration

The third iteration was the part where a lot of the process was going from the prototypes until actually coding it and finalizing the design in the gui. Since this was supposed to become an application, more than just a prototype a lot had to be done now regarding the actual implementation.

The different gates that were now going to take place on Infotiv was the following regarding the implementation.

- Demonstrate that the prototype of the interface can read user signal definitions
- Week 50
- Demonstrate that the prototype of the interface can read user frame definitions
- Week 3
- Demonstrate that the prototype can read excel file to populate user interface
- Week 26
- Present current dataflow in graphical interface - Week 28

Note that there was a break between week 3 and week 24 in the thesis due to other work. So starting from the high fidelity prototype it was now time to implement it into actual code. This was done in Electron which is a software framework that uses JavaScript together with the Node.js runtime.

When starting to implement the design I first started with creating a blank canvas. Once that was in place it was time to start adding the different elements to the GUI, just as shown in the different prototypes, buttons for adding the ECUs, naming them and furthermore configuring them.

6.5.1 Evaluation of implementation

Since there was a lot work done with the coding parts of the application at this point, it was time for more frequent gate/supervision and presentation the state of the gui/application to stakeholders in order to get feedback/pointers throughout the project. I showed the different stages of the design and got multiple levels of feedback. One being when toggling from signals and frames, aka pages inside a spreadsheet it is better to use a tab system rather than using a button to switch the different pages.

Compared to the prototypes there was now another viewpoint for the signals was now displayed in a more pleasant way. In the prototype it more looked like a direct imported spreadsheet in this current implementation there was a more natural look to it making it fit more to the GUI.

Guidelines at this point:

- Let the user name the different elements of the configuration
- Keep a minimalistic version of the gui and do not make it to cluttered for the user.
- Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.
- Make sure to involve all product owners and testers throughout the project
- Keep a good color combination with one primary and one secondary color that is distinguishable from each other.
- Keep the interface expandable for bigger configurations.
- Communicate in good time if there will be eventually cuts from the design compared to the initial plan.
- Create tabs to one block for the user to shift through the different pages.
- Give the user feedback if something in the configuration is about to be incorrect.
- Know what is reasonable to extract from a prototype to an actual implementation according to the timeplan.
- Make sure that the different paths from each screen is decided upon early on so you know how to continue with it.

7

Results

In this section the results of the thesis will be presented. First of the implementation/GUI that was the main deliverable to Infotiv. The second chapter will present the guidelines generated by this thesis.

7.1 Generic gateway application

As mentioned in chapter 6 the main focus of the third and heaviest iteration was the implementation. Turning the prototypes into code it Took a lot of time and effort but it was made possible.

Starting of we have the starting screen. Here after a lot of debating the primary color is yellow that works well together with the Infotiv logo as well as the ECU boxes that are around in the canvas when it is being configured. We have a toolbar on top that can support good help for the beginner user. We also have a heading describing the product. Furthermore the buttons to start creating the different ECU configurations inside of the interface.

When pressing the "Add ECU" button the user get's prompt with a windows open dialog box where the user selects the desired signal database. Once the user has selected which file to open the user get's to another dialog window where the user should enter the name of this ECU. Once the user has added the ECU, the user selects/modifies the different signals until it gets the desired features.

We now see the added ECU box with the tabs over it to select what view that should be displayed for the user. Furthermore there are also check boxes right next to each signal row in the table in the gui. This is a classical design pattern for multiple selection in a list of items. The different elements in the lists are also editable for the user.

Futhermore the different ECU elements place underneath one and other in a good way so that the user can add multiple, once the user exports the configuration it is export to the desired folder that the user wants to save the file to.

Hereby follows the different screens that the Generic gateway GUI displays:

7. Results



Figure 7.1: Start Screen, Final software.

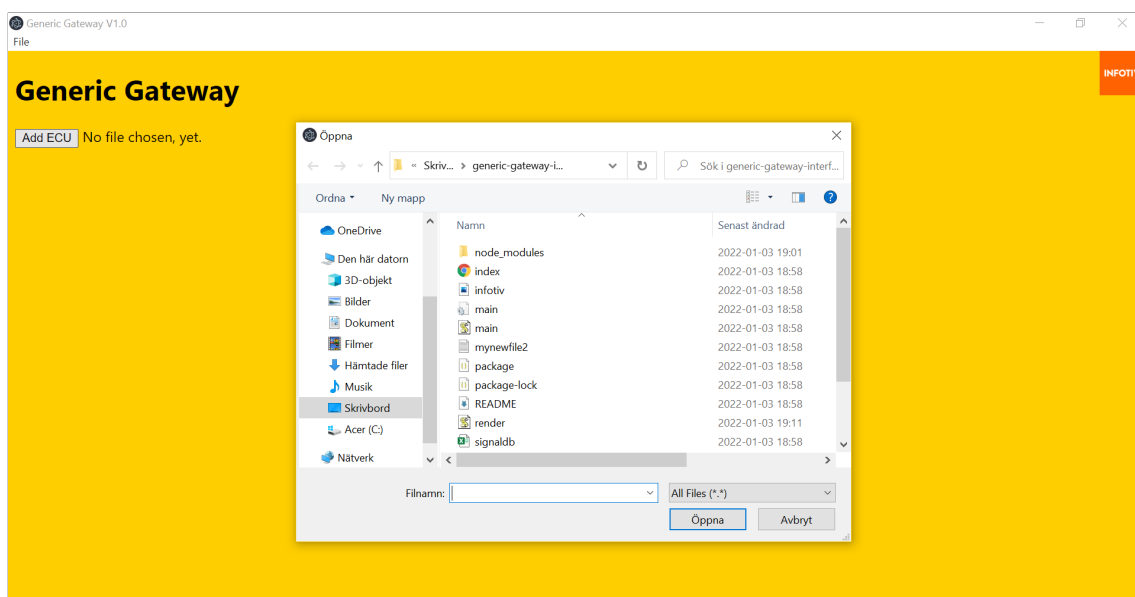
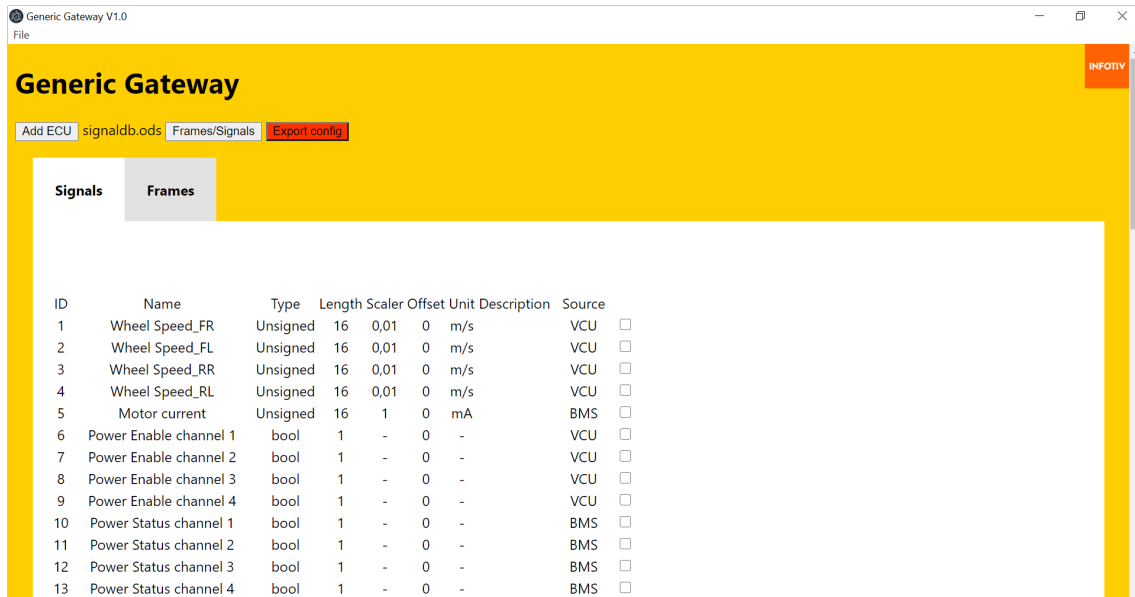
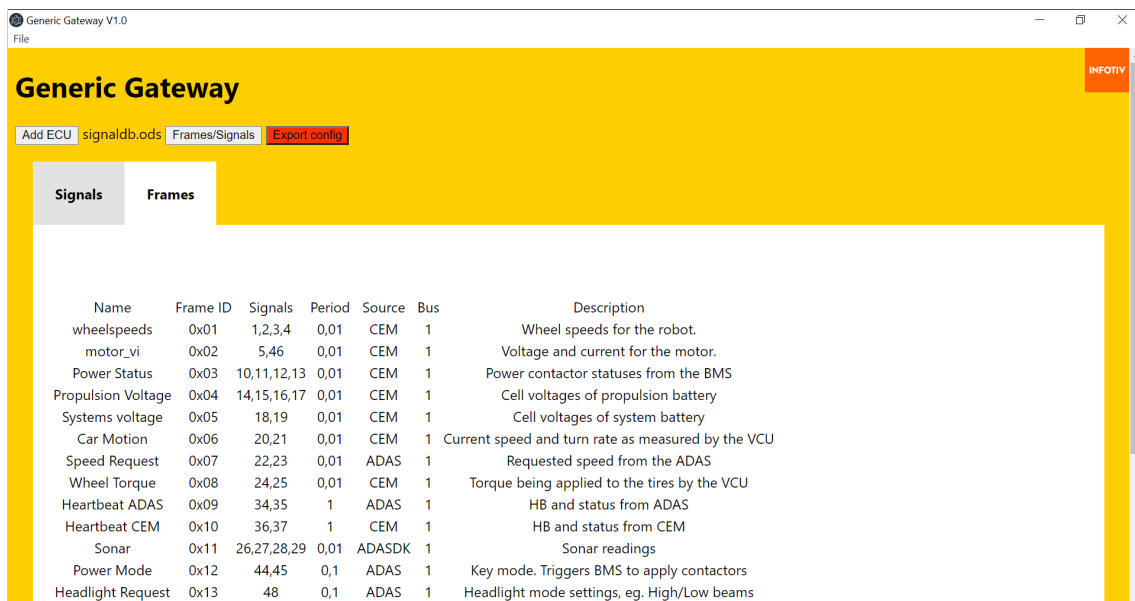


Figure 7.2: Opening a file, Final software.



ID	Name	Type	Length	Scaler	Offset	Unit	Description	Source
1	Wheel Speed_FR	Unsigned	16	0,01	0	m/s		VCU <input type="checkbox"/>
2	Wheel Speed_FL	Unsigned	16	0,01	0	m/s		VCU <input type="checkbox"/>
3	Wheel Speed_RR	Unsigned	16	0,01	0	m/s		VCU <input type="checkbox"/>
4	Wheel Speed_RL	Unsigned	16	0,01	0	m/s		VCU <input type="checkbox"/>
5	Motor current	Unsigned	16	1	0	mA		BMS <input type="checkbox"/>
6	Power Enable channel 1	bool	1	-	0	-		VCU <input type="checkbox"/>
7	Power Enable channel 2	bool	1	-	0	-		VCU <input type="checkbox"/>
8	Power Enable channel 3	bool	1	-	0	-		VCU <input type="checkbox"/>
9	Power Enable channel 4	bool	1	-	0	-		VCU <input type="checkbox"/>
10	Power Status channel 1	bool	1	-	0	-		BMS <input type="checkbox"/>
11	Power Status channel 2	bool	1	-	0	-		BMS <input type="checkbox"/>
12	Power Status channel 3	bool	1	-	0	-		BMS <input type="checkbox"/>
13	Power Status channel 4	bool	1	-	0	-		BMS <input type="checkbox"/>

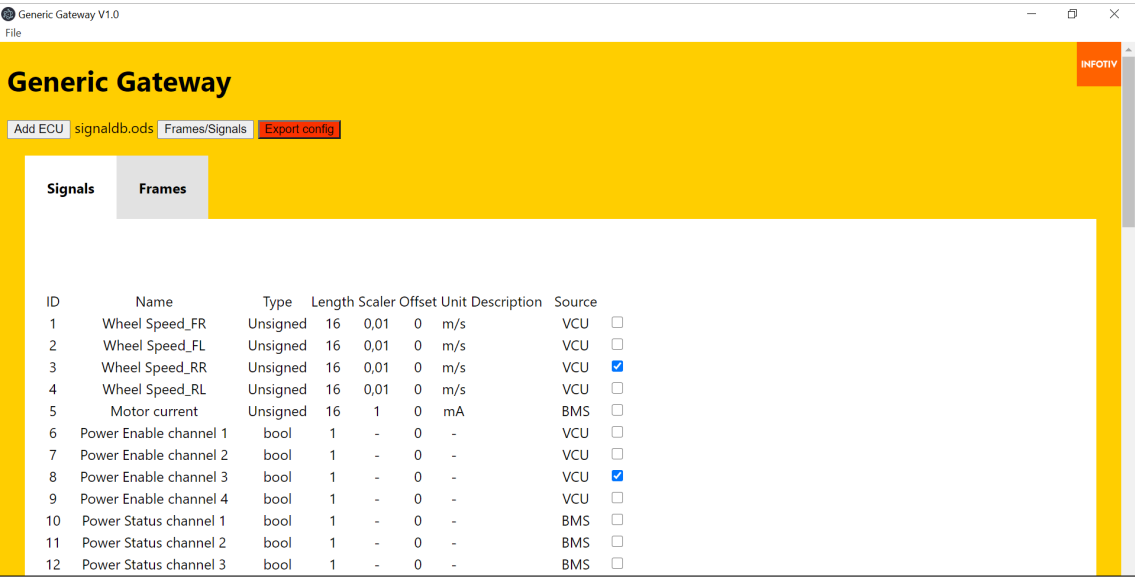
Figure 7.3: Loaded signal database, Final software.



Name	Frame ID	Signals	Period	Source	Bus	Description
wheelspeeds	0x01	1,2,3,4	0,01	CEM	1	Wheel speeds for the robot.
motor_vi	0x02	5,46	0,01	CEM	1	Voltage and current for the motor.
Power Status	0x03	10,11,12,13	0,01	CEM	1	Power contactor statuses from the BMS
Propulsion Voltage	0x04	14,15,16,17	0,01	CEM	1	Cell voltages of propulsion battery
Systems voltage	0x05	18,19	0,01	CEM	1	Cell voltages of system battery
Car Motion	0x06	20,21	0,01	CEM	1	Current speed and turn rate as measured by the VCU
Speed Request	0x07	22,23	0,01	ADAS	1	Requested speed from the ADAS
Wheel Torque	0x08	24,25	0,01	CEM	1	Torque being applied to the tires by the VCU
Heartbeat ADAS	0x09	34,35	1	ADAS	1	HB and status from ADAS
Heartbeat CEM	0x10	36,37	1	CEM	1	HB and status from CEM
Sonar	0x11	26,27,28,29	0,01	ADASDK	1	Sonar readings
Power Mode	0x12	44,45	0,1	ADAS	1	Key mode. Triggers BMS to apply contactors
Headlight Request	0x13	48	0,1	ADAS	1	Headlight mode settings, eg. High/Low beams

Figure 7.4: Loaded frame database, Final software.

7. Results



Generic Gateway V1.0

File

Generic Gateway

Add ECU signaldb.ods Frames/Signals Export config

Signals Frames

ID	Name	Type	Length	Scaler	Offset	Unit	Description	Source
1	Wheel Speed_FR	Unsigned	16	0,01	0	m/s	VCU	<input type="checkbox"/>
2	Wheel Speed_FL	Unsigned	16	0,01	0	m/s	VCU	<input type="checkbox"/>
3	Wheel Speed_RR	Unsigned	16	0,01	0	m/s	VCU	<input checked="" type="checkbox"/>
4	Wheel Speed_RL	Unsigned	16	0,01	0	m/s	VCU	<input type="checkbox"/>
5	Motor current	Unsigned	16	1	0	mA	BMS	<input type="checkbox"/>
6	Power Enable channel 1	bool	1	-	0	-	VCU	<input type="checkbox"/>
7	Power Enable channel 2	bool	1	-	0	-	VCU	<input type="checkbox"/>
8	Power Enable channel 3	bool	1	-	0	-	VCU	<input checked="" type="checkbox"/>
9	Power Enable channel 4	bool	1	-	0	-	VCU	<input type="checkbox"/>
10	Power Status channel 1	bool	1	-	0	-	BMS	<input type="checkbox"/>
11	Power Status channel 2	bool	1	-	0	-	BMS	<input type="checkbox"/>
12	Power Status channel 3	bool	1	-	0	-	BMS	<input type="checkbox"/>

Figure 7.5: Marked signals in database, Final software.

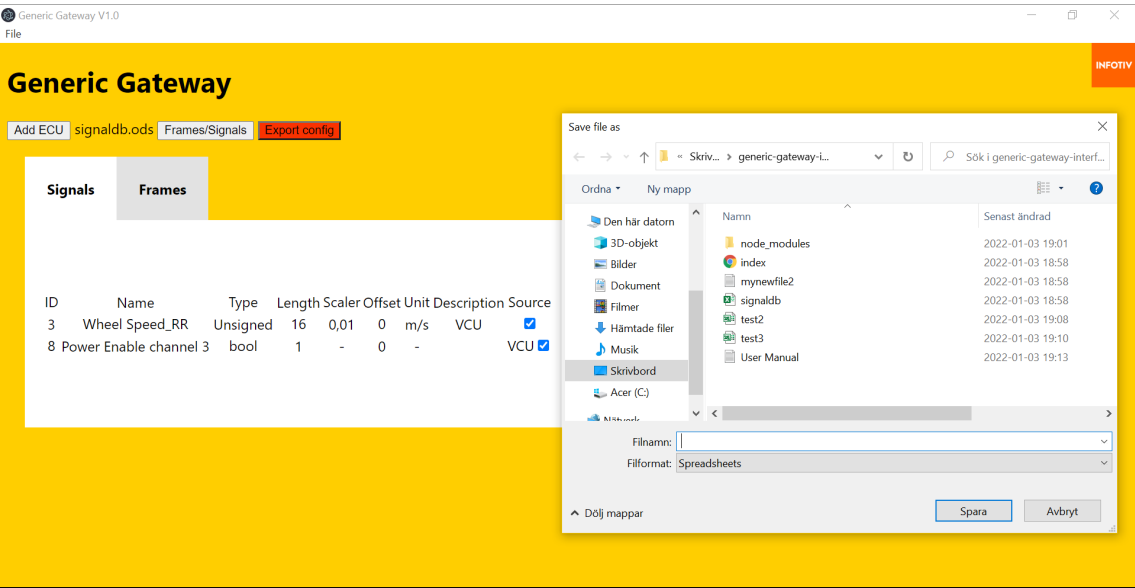


Figure 7.6: The screen after the user has pressed export, Final software.

ID	Name	Type	Length	Scaler	Offset	Unit	Description	Source
1	Wheel Speed_FR	Unsigned	16	0,01	0 m/s	VCU		
2	Wheel Speed_FL	Unsigned	16	0,01	0 m/s	VCU		
3	Wheel Speed_RR	Unsigned	16	0,01	0 m/s	VCU		
4	Wheel Speed_RL	Unsigned	16	0,01	0 m/s	VCU		
5	Motor current	Unsigned	16	1	0 mA	BMS		
6	Power Enable channel 1	bool	1	-	0 -	VCU		
7	Power Enable channel 2	bool	1	-	0 -	VCU		
8	Power Enable channel 3	bool	1	-	0 -	VCU		
9	Power Enable channel 4	bool	1	-	0 -	VCU		
10	Power Status channel 1	bool	1	-	0 -	BMS		
11	Power Status channel 2	bool	1	-	0 -	BMS		
12	Power Status channel 3	bool	1	-	0 -	BMS		
13	Power Status channel 4	bool	1	-	0 -	BMS		
14	Prop voltage cell 1	Unsigned	16	0,001	0 V	BMS		
15	Prop voltage cell 2	Unsigned	16	0,001	0 V	BMS		
16	Prop voltage cell 3	Unsigned	16	0,001	0 V	BMS		
17	Prop voltage cell 4	Unsigned	16	0,001	0 V	BMS		

Figure 7.7: The start spreadsheet of the signal database

Name	Length	Offset	Description
3 Unsigned	0,01	m/s	VCU
Power Ena	1	0	

Figure 7.8: The final export of the database

7.2 Guidelines

In this section the guidelines that have been generated in the thesis will be described and the parts that will be recommended will be highlighted.

These guidelines were generated throughout the project for every iteration. The guidelines are somewhat specific towards deigning interfaces for visualization and configuration of embedded systems but a lot of them can be applied to general design processes.

The guidelines that were generated throughout the project was: The guidelines that are related to process:

- Make sure to involve all product owners and testers throughout the project
- Communicate in good time if there will be eventually cuts from the design compared to the inital plan.
- Know what is resonable to extract from a prototype to an actual implementation according to the timeplan.

The guidelines that are related to designing interfaces for embedded systems:

- Let the user name the different elements of the configuration

- Keep a minimalistic version of the gui and do not make it to cluttered for the user.
- Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.
- Keep a good color combination with one primary and one secondary color that is distinguishable from each other.
- Keep the interface expandable for bigger configurations.
- Create tabs to one block for the user to shift through the different pages.
- Give the user feedback if something in the configuration is about to be incorrect.
- Make sure that the different paths from each screen is decided upon early on so you know how to continue with it.

7.2.1 Let the user name the different elements of the configuration

This guideline was generated in the first iteration. When designing an interface, especially for embedded/complex system and the process that will take place will create a lot of elements in the interface it is important to keep track of them and knowing which one is which. When you are designing an interface that will have a lot of similar elements it is important to keep track of which element that has which properties. In this case of this configuration that configures communication protocols with different name and properties it is extra important to be able to distinguishes them and the differences between them. For example the CAN and LIN protocol may look very similar and almost have the same features as each other. Even the building block may be very similar. By giving the user the opportunity to Name the different elements it is easier to keep them apart. As mentioned in section 3.3 it states that a design should **Prevent errors if possible**, by giving the user the opportunity to name different elements it reduces the risk for potential errors.

7.2.2 Keep a minimalistic version of the GUI and do not make it to cluttered for the user

This guideline may seem obvious but it does however become extra important when designing for embedded systems is when there are a lot of different element in the canvas. Especially for the beginner user when the user shall figure out the gui, how to use it and all the different elements within it. If there are to many elements for the user it may be hard to know where to start the configuration and visualization. In section 2.3.2 it describes the similar already existing software named Vector Canoe. What the interface of that software is lacking is a minimalist GUI, so by this guidelines interfaces similar to that will hopefully be avoided.

7.2.3 Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.

Something that was found lacking in existing software for configuration of embedded systems was a good starting tutorial or help in the beginning of the process. A clear understanding of what could be done in the application and how to do it must be presented early in the process of using a GUI for advanced products like this. The tutorial does not necessarily have to stay on for too long but it should however be accessible for the user. In section 3.3 it is stated that **Don't make the user remember information**, by having a tutorial in the beginning you make sure everything is explained the first time and do not need the user to keep remembering stuff.

7.2.4 Make sure to involve all product owners and testers throughout the project

In order to have a smooth process and making sure that all needed elements are included in the design process it is very important to include all product owners and testers thought the project. When working with iteration this becomes quite convenient as there are frequently presentations. Furthermore it is important to include all testers/users you can access that you know will be providing feedback to the design. It should be considered a very valuable resource that should be used throughout the entire project. This is also a perfect of the agile methodology as mentioned in section 4.4 that emphasised to be dynamic and agile in the project to include all stakeholders to get consequent feedback.

7.2.5 Keep a good color combination with one primary and one secondary color that is distinguishable from each other.

Something that was founded when designing the first and second prototype was that a clear distinction between the main and secondary color is necessary in order to know where and how to prioritize the users focus. For a pleasant viewing experience it is also better to have colors that works well together. In order to achieve this material.io has a great tool one can use to try out different versions. In the similar application described in section 2.2 this was something that was not experienced as considered when they designed their system.

7.2.6 Keep the interface expandable for bigger configurations.

Especially when designing interfaces for the configuration of communication protocols within embedded systems the configurations can become quite large, meaning that there are a lot of different boxes that will take up the entire interface. Because the size of the interface is important. In section 2.3 when similar applications is described an issue when a novice user was using the system it easily felt to cluttered.

7.2.7 Communicate in good time if there will be eventually cuts from the design compared to the initial plan.

When designing an interface, especially for embedded system it contains a lot of different elements. Therefore the design/interface can be considered rather complex. So it is therefore important to make sure that all stakeholders/designers know which elements that are important to focus on and make sure to include in the design. If there however becomes impossible to implement the elements in time the different stakeholders need to know that in order to prioritize the most important elements. The elements that make the entire configuration even possible, before fancy design. This connects to the agile methodology described in section 4.3 that issues similar to this can be brought up at an early stage throughout the project and adapt after.

7.2.8 Create tabs to one block for the user to shift through the different pages.

In order for a user to smoothly navigate through several pages within a configuration tabs are a more convenient option rather than buttons to toggle between the different pages/layers. This can be backed up by web browser and excel documents etc. By also providing a heading for each tab the user clearly knows what to expect when clicking on that tab. As described in section 3.3 the design principle **maintain consistency standards** let's the user recognise tab system from previous experience and apply it here.

7.2.9 Give the user feedback if something in the configuration is about to be incorrect.

When designing an interface for embedded systems it is of high importance that the configuration that is being created does not contain any errors because it can lead to very dangerous errors in the hardware, it is therefore very important to hint to the user that something may be configured wrong throughout the process of creating a configuration rather than that the error goes throughout the process and get's loaded into the hardware. The earlier an error can be hinted about to the user the better. For example if the user is trying to assign a value that is not possible to a specific variable or if the user is changing the name of a certain variable. In section 3.1 visual programming is described, in visual programming there are many possibilities to give visual feedback if a configuration is about to go the wrong way, by color, broken links etc etc.

7.2.10 Know what is reasonable to extract from a prototype to an actual implementation according to the time plan.

Something that is of high importance throughout any process is to have reasonable goals. Especially when designing for advanced embedded systems. There are several

aspects to consider when designing an interface like this. In general when designing a prototype it is quite easy to add several advanced features and new screens, but when it comes to implementing it it is defiantly more difficult. In order to prevent unnecessary expectations/work when doing the implementation having a discussion with all designers/stakeholders what is possible to actually code in to a working implementation is important will help a lot. In section 4.3 the Agile methodology is being described. By practicing the agile methodology these questions would have been discussed in daily standups and so on.

7.2.11 Make sure that the different paths from each screen is decided upon early on so you know how to continue with it.

When prototyping and even implementing an interface knowing which screen/click goes to which following screen is of high importance. It might seem obvious to everyone involved in the project before starting with the sketches and so on. When in fact the implementation part may lead to several confusing moments of what screen should lead to which screen. It is therefore important to have a clear mockup/prototype to know which click/screen leads to which one of the following. This is mentioned in section 4.4 when referring to the double diamond method it is claimed that it is important to "revise the problem statement".

8

Discussion

Throughout this thesis a lot of areas went well and generated a good result. Some parts of the project did not go according to plan and changes needed to be made in order to continue and finish the project. In this chapter the different parts of the thesis will be discussed, what can/could have been improved and which parts that worked well.

8.1 Results

I am in general very satisfied with the result. There was a lot of work going in to this thesis, both in terms of research, evaluation and foremost implementation. Creating the sketches/prototypes went fairly smooth but to implement every aspects took more time. The fact to go from a high fidelity-prototype to an implementation is however very rewarding. To see the functions working and providing the features necessary for the user to create the configurations leaves this project with the feeling of providing a great project. The implantation is a finished prototype that has been coded in python and is exported as a standalone app that can be ran on windows computers.

There is however a lot of improvements that could have been done regarding the different elements of the results. The more time spent towards coding would obviously generate a better application/implementation, especially after the latest feedback that was received when presenting the final demo for the different stakeholders at Infotiv. There could have been more added features that were some of the original plan. The code that were written has defiantly room for improvement, both in terms of how well the code is optimized and how well tested each function is. Unit testing is always a great tool no matter what the code shall do. So implementing more of the unit tests would have provided a better code. There were also several feedback gotten from the last presentations from the stakeholders that the design of the implementation could have been improved, such as choosing a different color pattern as well as making the buttons that should pop out more even more obvious in the color matching.

The guidelines are in some sense specific for designing for embedded systems but they are also quite general when it comes the process and the art of designing. As previously mentioned after the final evaluation more knowledge/feedback were received which could possibly modify/generate new guidelines to those who are al-

ready generated. On the other hand some of the guidelines are very specific towards designing interfaces for configuration of communication protocols within embedded systems.

8.2 Process

The process of this project/thesis there is a lot to be discussed. The process is far from perfect and has a lot of potential to be improved for the future thesis work that will be similar. For starters the pre studies could have been more thoroughly done and involved more research towards existing guidelines. The main conflict that made it not possible to do more research is that both Infotiv and the planning parts of Chalmers University were very heavy. In order to start a thesis at Infotiv several different documents needs to be in place, same thing at Chalmers. This ended up with a lot of work just in order to be able to start the thesis. Once the thesis was started it was time to dive deep into the prestudies.

Even though a lot of time were spent on theory and methodology, it resulted in that all that research that were made was not quite applicable for this project. It did however generate some knowledge of how to tackle a wicked problem which lead to better ways to work with design issues. Time were also spent trying to analyze/finding a lot of similar software. Even though this provided some guidance of how to start out/getting inspiration towards creating my own design it might have been to much work put into that area of the research. It turns out that throughout each iteration of the project that the feedback from the stakeholders and users was the main pointers that decided and got integrated in the design. Solving wicked problems was defiantly made easier by using the agile methodology in terms of recursively coming back to the problem with the new design, evaluating what the status is and how to get to the next step.

When it comes to prototyping more time could defiantly been spent on that area. Once the project was started there was a will to start with the high fidelity prototype right away, almost even wanting to start with the coding right away because in order to generate a working application at the end of the project while at the same time performing all the other things that a thesis involves the coding needs to start early but as mentioned in the guidelines and what is quite trivial in every design process is to spend a lot of time sketching, coming up with different solutions to solve each wicked problem of the design and so on. More time was put into the high-fidelity prototype whereas I think spending more time on the different sketches and ideating around different.

COVID-19 did not affect the end result but it did however create some obstacles throughout the process. When this thesis started in October 2020 there were a 50/50 situation at Infotivs office with working there or working at home which we started with but realised pretty soon that keeping all the work remote/from home would be the most beneficial for the project. How the pandemic impacted the project the most was with the user interviews and evaluations. It was quite hard to find actual

users and the users were the workers at the office, since they were rather busy with their mission the most time I had for evaluation was when they had some small time over in their day which was easier to know and access them when being in the office. Also having every meeting/evaluation online did not provide the most convenient circumstances for interviews etc.

Overall I think that the main obstacle throughout the process of this project were that Infotiv highest interest was a working application that could configure the generic gateway. Their focus was more on the product, whereas Chalmers focus more on the research regarding the guidelines. It was however possible to work simultaneously with both parts but coding always takes more time then anyone think it will take. So having the possibly to generate design guideline while having strict deadlines for each coding delivery made it quite stressful to achieve a great devilry for both Chalmers and Infotiv. For a future thesis like this it is highly recommended to focus on one, possibly keep it to prototypes and not implement and code the prototypes.

8.3 Validity and Generalisability

I am confident that the guidelines as well as the implementation has provided a lot of help for engineers and designers. The guidelines have generated new knowledge within the area of interfaces within embedded systems. The application itself has also provided great help for the engineers developing and using different communication protocols.

Even though the guidelines have provided new knowledge in the area there is however a point in seeing them with some scepticism since this thesis has focused a lot on the CAN and LIN interface. So for embedded systems in cars and other vehicles in the automotive industry they apply well but that does not specifically imply that it would work as well as within other areas.

However it is positive that the guidelines are quite general and applies to a lot of different design areas. Some of the guidelines such as **Make sure to involve all product owners and testers throughout the project** and **Communicate in good time if there will be eventually cuts from the design compared to the initial plan.** are very broad and can be used in a lot of different projects and thesis no matter what area.

8.4 Ethical aspects

In order to peruse this project and thesis it was important to interview and talk a lot to potential users and stakeholders. When having interviews or asking the users to try out a prototype I had to be sure that the user felt comfortable and did not feel that they were considered slow if they did not understand what was the next step in the prototype or anything similair.

Since the gui that was developed are made to create configurations that shall go inside a car or another embedded system it is of high importance to make sure it is as safe as possible for all parties involved into the project. That is why it should be very clear to the user when an error in the configuration has been made to prevent as many errors as possible because errors made in this domain could lead to severe danger.

When designing this interface I also involved a lot of thought into making it accessible and easy to use by everyone no matter the experience of similar apps or interfaces. It is of high importance to spend time regarding the color blind people. That is why some investigation was made regarding which colors to use for the colorblind.

8.5 Future work

This thesis and project contains a lot of areas that can be further worked on. I believe that in every design project the more iterations that have been performed the better. In this project however the implementation also had to be done in a limited amount of time. This resulted in that the design research is somewhat limited unfortunately. It does however provide a good base for future students who would continue research and thesis in this area. All my prestudies together with the research of designing an interface for embedded systems has created a great foundation for the future students to take on and create an even better interface.

8.5.1 Implementation

Regarding the implementation a lot of future work can be done on that as well. As previously mentioned it was of high importance to both do research regarding the guidelines together with coding the implementation. It did however result in good results both implementation wise as well as guidelines.

As with any code there is always a lot of improvement that can be done. Both in terms of getting the code to run faster and shorten down the Ordo time notation. The code can also be improved in terms of having more of the functions unit tested. Regarding the implementation/features that exist in the current application they can be improved as well as more functions can be added to the GUI. As previously discussed in the thesis there was a desire to implement the so called *advanced features*, but it was not managed due to lack of time. These features would however contribute to a greater application with more abilities to make the configuration even more advanced. Further iterations on placement of the buttons and the general layout of the interface would also improve it.

8.5.2 Evaluation

In terms of evaluation the design together with the code could be even more evaluated. Having the possibility to interact with more potential users and the coworkers

at Infotiv would generate more feedback that could then be used in a later iteration of the project. It would also have been beneficial to involve more color blind users to the process to make sure that the color scheme chosen for the design was as good as possible.

There would also be interesting to investigate more ways to create a special configuration. A configuration where that maybe uses a complete different way to select which signals together with which frames that are going to be used. Also investigate if the amount of modules/clicks a users has to go through in order to complete the configuration would be interesting to see what could have been done to shorten down this path and speed up the entire process.

9

Conclusion

This project has consisted of performing the research of what should be configured when designing an interface for embedded systems. The generated results will provide knowledge for future research in this area. It has concluded which parts that tend to get to advanced in interfaces like this and what guidelines to follow in order to prevent it. This thesis has had a very special time frame. It started in October 2020 and ended almost a year later in September 2021, this due to a break during spring 2021 for other Chalmers work.

The implementation coded throughout this thesis has resulted in a interface that is usable by a user in order to create a configuration for embedded system in a convenient way. The thesis were carried out by research, sketching, prototyping, iteration, evaluation and finally implementation.

The guidelines that are related to process:

- Make sure to involve all product owners and testers throughout the project
- Communicate in good time if there will be eventually cuts from the design compared to the initial plan.
- Know what is resonable to extract from a prototype to an actual implementation according to the timeplan.

The guidelines that are related to designing interfaces for embedded systems:

- Let the user name the different elements of the configuration
- Keep a minimalistic version of the gui and do not make it to cluttered for the user.
- Have an easy start tutorial for the first time user so she/he can understand the start of the configuration.
- Keep a good color combination with one primary and one secondary color that is distinguishable from each other.
- Keep the interface expandable for bigger configurations.
- Create tabs to one block for the user to shift through the different pages.
- Give the user feedback if something in the configuration is about to be incorrect.
- Make sure that the different paths from each screen is decided upon early on so you know how to continue with it.

The future work of this project would be to further evaluate the design even more and test it with more users. Since the implementation had to be done in time a lot

9. Conclusion

of time were spent on coding it so in order to improve the interface more testing has to take place. Regarding the guidelines they provide a foundation for future students to create interfaces on, the guidelines can however be even more improved by more iterations.

Bibliography

- [1] How to conduct a heuristic evaluation, 1994. <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>. Accessed: 28 November, 2020.
- [2] Cognitive walkthrough for the web, 2002. https://www.researchgate.net/publication/221514534_Cognitive_walkthrough_for_the_Web. Accessed: 19 November, 2020.
- [3] State of the art on the cognitive walkthrough method, its variants and evolutions, 2010. <https://www.tandfonline.com/doi/abs/10.1080/10447311003781409>. Accessed: 19 November, 2020.
- [4] Using affinity diagrams to evaluate interactive prototypes, 2015. https://www.researchgate.net/publication/284917650_Using_Affinity_Diagrams_to_Evaluate_Interactive_Prototypes. Accessed: 28 November, 2020.
- [5] User interviews, 2018. <https://www.interaction-design.org/literature/topics/user-interviews/>. Accessed: 8 December, 2020.
- [6] Weighted decision matrix, 2018. https://deseng.ryerson.ca/dokuwiki/design:weighted_decision_matrix. Accessed: 10 December, 2020.
- [7] Why personas fail, 2018. <https://www.nngroup.com/articles/why-personas-fail/>. Accessed: 28 November, 2020.
- [8] Why the double diamond is the most precious diagram in ux design, 2018. <https://evaschicker2012.medium.com/why-the-double-diamond-is-the-most-precious-diagram-in-ux-design-40db0476e5d2>. Accessed: 18 November, 2020.
- [9] Design principles, 2019. <https://www.interaction-design.org/literature/topics/design-principles>. Accessed: 12 November, 2020.
- [10] Information visualization, 2019. <https://www.interaction-design.org/literature/topics/information-visualization>. Accessed: 12 November, 2020.
- [11] What is visual programming?, 2019. <https://www.outsystems.com/blog/posts/what-is-visual-programming/>. Accessed: 12 November, 2020.
- [12] Communication protocols in embedded systems – types, advantages disadvantages, 2020. <https://electricalfundablog.com/communication-protocols-embedded-systems/>. Accessed: 7 November, 2020.
- [13] An introduction to agile working, 2020. <https://www.targetinternet.com/an-introduction-to-agile-working/>. Accessed: 13 November, 2020.

- [14] Math. graphics. programming., 2020. <https://se.mathworks.com/products/matlab.html>. Accessed: 10 November, 2020.
- [15] Research through design, 2020. <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/research-through-design/>. Accessed: 5 October, 2020.
- [16] Simulation and model-based design, 2020. <https://se.mathworks.com/products/simulink.html>. Accessed: 10 November, 2020.
- [17] Testing ecus and networks with canoe., 2020. <https://www.vector.com/int/en/products/products-a-z/software/canoe/>. Accessed: 12 November, 2020.
- [18] What is a wicked problem and how can you solve it?, 2020. <https://www.interaction-design.org/literature/article/wicked-problems-5-steps-to-help-you-tackle-wicked-problems-by-combining-system>. Accessed: 13 November, 2020.
- [19] What is user research?, 2020. <https://www.interaction-design.org/literature/topics/user-research/>. Accessed: 6 October, 2020.
- [20] Infotiv, 2021. <https://www.infotiv.se/en>. Accessed: 10 December, 2020.
- [21] Dave Bourgeois and David T. Bourgeois. Information systems for business and beyond, chapter 5: Networking and communication, 2014. <https://bus206.pressbooks.com/chapter/chapter-5-networking-and-communication/>. Accessed: 6 October, 2020.
- [22] Thomas Nolte, Hans Hansson, and Lucia Lo Bello. Automotive communications - Past, current and future. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, volume 1 2 VOLS, pages 985–992, 2005.