

KOKKUVÕTE

Käesoleva bakalaureusetöö eesmärk oli luua Taltech iseAuto projektile näidikuplokk, mis vastaks sõiduki erilistele nõuetele ning mida saaks tulevikus insenerid vajaduspõhiselt täiendada. Antud bakalaureusetöö eesmärk sai täidetud.

Töö alguses uuriti teiste tootjate välja toodud kasutajaliideseid, ning leiti mis on töö jaoks sobilik kasutajaliidese puhul ja mis isesõitvate autode puhul nii väga ei sobi. Tesla paistis silma oma graafika keerukuse poolest, aga see ei olnud sobilik Taltech iseAuto jaoks. Teiste sarnaste isesõitvate sõidukite projektide kohta oli üllatavalt vähe infot, neil oli palju reklaame ja muid turunudmaterjale, aga informatsioon liideste kohta oli pealiskaudne. Seega tuli lähtuda enamasti piltide pealt saadud infost. Välja paistis Ford oma lihtsuse poolest, mis juhtub olema täpselt see mida Taltech iseAuto projekt vajab. Praegune lahendus jätab palju tahta just selle külluse poolest, kus on kasutatud sõiduauto füüsilist kasutajaplokki, mis on tänaseks päevaks aegunud variant.

Enne kasutajaliidese koostamist tuli sätestada vajalikud eeldused mida peab näidikuplokk saavutama. Neid eeldusi silmas pidades oli võimalik teha valikud, milliseid komponente kasutusele võtta ja mis viisil oleks mõistlik probleemi lahendada. Kui valikud olid tehtud hakati kasutajaliidest koostama vastavalt sätestatud piirangutele ja nõuetele. Liides pidi olema lihtne, et ei tekitaks segadust. TouchGFX osutus väga paindlikuks tarkvaraks, mis võimaldas kõik vajaliku ära teostada võrdlemisi vähese koodi kirjutamiseta. Kasutajaliidese puhul oli kasutusel ainult C++ programmeerimiskeel. Kontrolleri koodi kirjutamisel kasutati C keelt.

CAN võrk on robustne lahendus sõidukite elektroonika komponentide vahelise suhtluse saavutamiseks. Samas võrreldes varasemate lahendustega, on see odavam, hoides kokku üleliigse juhtmestiku pealt ja on samuti samal põhjusel ka kõvasti lihtsam. Töö käigus koostati struktuur, kuidas CAN võrk ühildub mikrokontrolleri ja vastavate komponentidega, samaaegselt selgitades mõningaid tähtsamaid võrgu töö põhimõtteid, mis on standardiseeritud aastast 1993. Üheks oluliseks aspektiks on ka CAN võrgus olevate signaalide jälgimine ja analüüsimine, et oleks võimalik lihtsustada süsteemi arendust. Selle saavutamiseks kasutati CANdoISO adapterit ja vastavat programmi mis andis hea ülevaate, et millised näevad välja erinevad edastatavad signaalid ehk andmekaadrid. Selgitati ka kontrolleri poolset ülesseadistust, et see oleks võimeline võtma vastu CAN signaale ja neid ära kasutama ja visuaalselt näidustama kasutajaliidesel.

Peale süsteemi valmistamist tuli luua korpus, et majutada seda. Selleks kujundati joonestamis programmis mudel, arvestades projekti iseärasusi. Hiljem sai korpus prinditud 3D printeriga ja kinnitatud kontrolleri külge.

Eelnevate sammude kokkuvõtteks teostati riskianalüüs, ning leiti mis on käesoleva lahenduse potentsiaalsed veakohad ja nende ohu tasemed. Seda silmas pidades sai luua lahendusi nende vigade leevendamiseks ja tuleviku samme seada.

Tulevikus oleks lootus luua oma universaalne kontrolleri trükkplaat, selle jaoks aga jäi selles töös liiga vähe aega. Loodetavasti tulevikus saab Taltech iseAuto tiim ehitada selle pealt mis antud töös saavutati.

SUMMARY

The goal of this bachelor's thesis to create a new dashboard for the Taltech iseAuto project was reached. It was important that we learn from what others have come up with for the GUI so we might find out what would work in a self-driving car and what might not. Tesla stood out for the intricacy of its graphical interface, but it is not suitable for the Taltech iseAuto project. There was surprisingly little information about other similar self-driving vehicle projects, they had a lot of promotional material, but the information about the GUI was lacking, and the author had to rely mostly on the information obtained from pictures. Ford stood out for its simple UI design, which happened to be exactly what the Taltech iseAuto project needs. The current solution leaves a lot to be desired in terms of its functionality, this is due to the dashboard of an i-MiEVi non-self-driving car being used, which is an outdated option today.

Before creating the user interface, first it had to be established what the necessary prerequisites are that the dashboard must achieve. With these in mind, choices about which components we would use and how we would solve the problem could be made. Once the choices were made, it was possible to start creating the user interface according to the specified restrictions and requirements. The interface had to be simple to avoid confusion. TouchGFX turned out to be a very flexible piece of software that allowed us to do everything we needed to with relatively little code to write as most was generated by the program. For the user interface only the C++ programming language was used, but the c language had to be used a little when creating the microcontroller code.

The CAN network is a robust solution for achieving communication between a vehicle's different electronic components. Compared to previous solutions, it is cheaper thanks to saving on excess wiring, and for the same reason, it also makes the system much simpler. In this work, we go over the structure of how the CAN network communicates with the microcontroller and the surrounding components, while also explaining some of the more important principles of the network, which have been standardized since 1993. One important aspect is also the monitoring and analysis of the signals in the CAN network in order to simplify the development of the system. To achieve this, the CANdoISO adapter and the accompanying program were used, which gave a good overview of what the different transmitted signals, or data frames, look like. This thesis also explains how to configure the controller to be able to read and write CAN signals so it can use them on the user interface.

After the completion of the entire system, it needed to be housed. For this, a design was drafted, considering the complexities of the project. As this was done, it was possible to print out the design on a 3D printer, and the finished component was fixed to the controller.

To summarize the previously completed points, a risk assessment, or FMEA, was done to find the potential failure modes and their associated risk factors. With this, it was possible to find ways to alleviate these possible shortfalls and give advice to the future developers of this project.

In the future, the authors hope that sometime in the future the system will get its own universal controller circuit board, but unfortunately there was too little time left for this. Hopefully, in the future, the Taltech iseAuto team will be able to build on what was achieved with this work.