Valg er teknologi - User interface - platform

Vi har tydeligt udformet en overordnet struktur for vores Lyfter!-system.

En bruger styrer vha. et userinterface en bil der på sig har 4 hjul, et sensor-system og en løftearm.

Vi har herved skulle vælge hvilken platform dette userinterface skulle ligge på, og hvordan denne platform for så vidt skal kunne benyttes af brugeren, og kommunikere med bilen.

Vi opskrev mulighederne således:

1. En webapp hostet via en hjemmeside, hvor brugeren vha. on screen knapper kan styre bilen. Der kommunikeres her med bilen vha. en websocket, som vi har været til workshop om. Denne løsning kræver opsætning af en hjemmeside, hvori der er behov for udvikling i html, javascript og muligvis css.
2. Et userinterface på en windows computer udviklet som en c# windows forms app. Kommunikation vil her foregå som en socket forbindelse til raspberry pi på bilen, hvor begge enheder er på samme netværk.
3. et userinterface liggende lokalt på raspberry pi’en på bilen, synliggjort for brugeren igennem en touchskærm sat til raspberry pi’en. Det ville her blive udviklet som en linux application gennem et linux graphical API såsom QT eller wxWidgets.
4. Et userinterface fra en pc kørende på en linux platform, der vha et socket forbindelse lignende den fra mulighed nr. 2 kommunikerer med raspberry pi på bilen.

Den vil her blive udviklet som et linux graphical API såsom QT eller wxWidgets.

Vi har valgt mulighed nr. 4 da vi her vil kunne have et server-client forhold mellem user interface og raspberry pi, der minder om hinanden i højere grad end hvis vi f.eks. havde valgt mulighed nr. 2.

Når man beskæftiger sig med sockets benytter man sig i af nogen socket biblioteker der er platform bestemte. Da vi i dette tilfælde benyttede os af en raspberry pi der kører en poky linux udgave, valge vi at udvikle user interface i linux, så vi kunne benytte os af de samme linux-specifikke biblioteker, og derved opnå et mere gnidningsfrit implementeringsforløb.

Herefter skal vi vælge et linux graphical api til at udvikle vores userinterface. Som tidligere nævnt er mulige løsninger at udvikle det i QT eller wxWidgets.

Vores krav er rimelig simple: at API’et udover at understøtte linux, er til c++, og at det er gratis.

wxWidgets, modsat qt, er blot et bibliotek der muliggør udvikling af et grafisk userinterface. For specifikke jobs eller udvidelser på dens funktionalitet, er der behov for andre biblioteker.  
QT er sit eget miljø, og har klasser og biblioteker for stort set alt. Med et tilhørende udviklingsmiljø kan man debugge ens program meget hurtigt, og alle dens klasser er veldokumenteret med tilhørende eksempler på deres online opsalgsværker.

Da QTcreator også allerede er installeret på vores linux image, valgte vi QT som API til at udvikle vores userinterface.