

## Individuell teknisk reflektion

IS1204 IT-projekt del 2, Autonoma Inbyggda System

Grund idén till vår produkt var en robot som vid brand skulle kunna söka av ett rökfyllt rum efter potentiella människor och rapportera resultatet till närmaste brandstation, för att hjälpa brandmännen med deras arbete. Under projektets gång har vi däremot stött på flera olika hinder och begränsningar och tillsammans med vår product owner definierade vi därför om robotens huvudsyfte till att söka efter människor i utrymmen utan att behöva klara av några extrema temperaturer.

Vår första sprint, var mycket av en test vecka. Varken vi eller vår product owner visste så mycket om vårt arbetssätt Scrum eller hur mycket tid allt vi skulle göra kunde tänkas ta. Detta resulterade i att vi redan första dagen hade uppnått sprintens mål och stod utan planering över vad som skulle göras där näst. När vi sedan la upp arbetet för resten av veckan var vi inte säkra på vad vår product owner hade för prioritering av produktens delar och började arbeta på det vi själva ansåg vara produktens viktigaste egenskaper. En av mina arbetsuppgifter var då att skapa ett simpelt gui för brandmännen där dem skulle få upp information om hur många människor som hade detekterats och var, då vi ansåg att det var en väsentligt del att informationen som roboten samlade in kom fram för att vår produkt skulle fylla någon funktion. Under vårt andra sprintmöte prioriterade dock product ownern själva kommunikations delen betydligt längre ner och vi fick därför aldrig någon användning av det programmet jag hade skapat. I ett annat projekt skulle jag därför vara noggrannare med att förstå kundens önskemål för att slippa onödigt arbete.

En för projektgruppen väsentlig lösning var vårt beslut om att införskaffa en färdig radiostyrd bil att ha till grund för roboten. Den var billig och därför väldigt passande för vår budget, och sparade oss väldigt mycket till. Genom att koppa dess motorer till vår arduino hade vi direkt en programmerbar bil, och slapp därför både lägga tid på att söka information om passande delar och montera ihop dem.

För att kunna komma igång snabbt och se så att vi kunde kommunicera med vår bil med hjälp av arduinon så skrev jag en arduinokod som gjorde att vi kunde kommunicera med arduinon över seriell länk och skicka kommandon till den i en vanlig pc terminal. Vilket vi snabbt fick till att fungera och vi kunde senare även använda denna koden för testning av senare konstruerat controllerprogram för roboten, då vi visste att bilens alla funktioner fungerade om vi körde kommunikationen över terminalen.

En annan simpel men väsentlig lösning som vi skapade väldigt tidigt under konstruktionen av vår robot var de switchar som vi kopplade till batterierna för att slippa koppla ur dem manuellt varje gång och på så sätt sparade vi mycket tid.

En lösning som vi i efterhand borde ha gjort annorlunda var att vi ganska tidigt lödde fast alla våra kablar permanent på ett testkort som vi monterade på roboten för att få ett snyggt resultat. Detta resulterade i att felsökningen blev väldigt svår när det uppstod problem med de olika sensorerna och bluetooth modulen och vi fick lägga ner onödigt mycket tid på att felsöka, plocka bort och löda om våra olika kopplingar. Skulle jag göra om projektet skulle jag ha använt mig av ett lödfritt kopplingsdeck ända till projektets slut och inte löda fast allt för än jag var helt säker på att alla delar fungerade och att inga kopplingar skulle tillkomma, även om det inte skulle se så snyggt ut under demonstreringarna.

För att roboten skulle ha något att utgå ifrån innan den hade upptäckt något med hjälp av sina sensorer skrev jag kod för ett slumpmässigt rörelsemönster, som roboten hela tiden skulle kunna komma tillbaka till om inga andra inputs hade getts. Detta var en ganska simpel och inte så tidskrävande lösning men något som vi kunde använda oss av under hela projektet.

För att lätt kunna tolka de olika meddelanden som skickades mellan arduinon och kontrollerprogrammet över serial länken skapade vi en standard som innebar att alla meddelanden började med en stor bokstav som identifikation på vad det var för typ av information som meddelande innehöll. Olika data separerades med kommatecken och varje meddelande avslutades alltid med en punkt för att lätt kunna notera när meddelandet var slut. Detta gjorde att vi enkelt kunde skicka kommandon från kontroller programmet till arduinon och skicka sensordata från arduinon till kontrollerprogrammet.