Digitala Projekt
Stor Kurs

Johan Nylander
Anders Grundén

2007-05-16

To create a car that will follow a predetermined course is an application with many problems. We wanted to build a car that can follow and remember a course of our choice in order to recreate its path. There is no actual market for our application, but it has been on people's minds that real cars could be driving in a similar way in some future. Our vehicle is considered a prototype of this system.

Innehållsförteckning

1. Uppgift

Uppgiften gick ut på att bygga en valfri applikation, som skulle klara en viss kravspecifikation. Vi valde en egen uppgift som bestod i att bygga en bil, som skulle kunna följa en utlagd tejpremsa. För att sedan sätta en personlig prägel på problemet ville vi även att bilen skulle komma ihåg var/hur den kört och kunna återskapa detta med en hyfsad noggrannhet.

2. Kravspecifikation

Målen med projektet var att bilen skulle kunna köra på egen hand och följa tejpen till en noggrannhet av ca 1cm. Vi ville även ha ett mål att eftersträva då det gällde återskapandet av den nyss körda banan. Vi ville att bilen inte skulle avvika mer än 5cm från slutpunkten, då den startat från samma startpunkt.

3. Arbetets gång

3.1. Hårdvaruplanering

Första steget i arbetet gick ut på att rita upp ett schema över hur hårdvaran skulle sammansättas för att fungera som den enhet vi var ute efter. Detta steg krävde mycket påläsning av diverse datablad, och ingående kunskap om varje komponent krävdes för att få ett relativt fungerande system.

3.2. Realisering av applikationen

Stommen av applikationen gavs i form av två mindre kopplingsplattor som formades för att ge en god grund till bilen. Varje komponent sattes in på en övre kopplingsplatta som utgjorde vår dator del, medan en undre platta användes som kaross där motorer och sensorer monterades. Då datordelen skulle realiseras krävdes att vår första ritning av hårdvaran följdes noggrant för att minska antalet fel då denna senare skulle testas.

3.3. Felsökning av hårdvara

Då alla komponenter och delar av vår bil var sammansatta till en enhet påbörjades en felsökning av hårdvaran. Detta skulle upptäcka eventuella fel i vår första planering eller uppbyggnad av applikationen. Detta gjordes väldigt systematiskt, genom att koppla in en komponent i taget och se hur den beter sig i den totala kopplingen. Till vår hjälp hade vi en emulator, som representerade de komponenter vi ännu ej hade kopplat in. På så sätt kunde vi se om det var fel på någon specifik komponent, istället för att koppla på alla och hoppas på att det skulle fungera direkt. Även programmering av vår logik gjordes och felsöktes.

3.4. Utveckling av mjukvara

Då hårdvaran var testad och fungerande enligt vår första ritning skulle mjukvaran skrivas och implementeras. Även här användes samma emulator som fungerade som vår processor, klocka och EPROM. Detta underlättade vårt arbete då programmeringen av EPROM ej behövde göras kontinuerligt. Till en början skrevs en enklare programkod som skulle testa att applikationen betedde sig som önskat. Då detta var avklarat utvecklades ett mer avancerat program som skulle få bilen att uppföra sig utefter våra riktlinjer.

4. Applikationen

För realisering av applikationen använde vi oss av:

- **Två fotodioder** som användes för detektering av tejp och golv. Dessa sitter nära mätytan för bästa resultat.
- **Två lysdioder** som representerade det som fotodioderna detekterade (lyser vid tejp, lyser ej vid golv). Med hjälp av detta kunde bilens beteende enklare analyseras.
- **Två motorer** som styrs separat med hjälp av den information som tas in av fotodioderna.
- **En initieringsknapp** som startar bilen.
- **En switch** som bestämmer om bilen skall följa tejp eller återskapa tidigare bana som är sparad i minnet.

Dessa delar utgjorde grunden för bilens egenskaper.

Då bilen skall utföra sin uppgift placeras bilen så att en fotodiod är över tejpen och den andra över golvet. Som första åtgärd trycks en initieringsknapp in vilket får applikationen att läsa av kontrasten mellan tejp och golv. Med dessa data skapas ett tröskelvärde som under körningen avgör vad fotodioderna detekterar. Då initieringen har utförts blinkar lysdioderna en kort sekvens för att indikera att läsningen har gjorts. Strax efter initiering kör bilen iväg och följer utsatt bana. Då bilen kört en stund kan en switch ändras till ”minne” på bilen som då stannar och inväntar ett nytt knapptryck. Bilen blinkar därefter till igen och påbörjar en återgivning av den senast körda banan.

5. Problematik

Våra tidigaste problem berodde på att vår första hårdvaruplanering. Då vi aldrig hade jobbat med IC-kretsar på detta ingående vis, fick vi problem med att bestämma egenskaperna hos varje ben på vissa mer komplicerade komponenter. Detta var ett problem som följde oss genom större delen av arbetstiden, då vi aldrig riktigt kunde fastställa att allt var rätt kopplat. Utöver dessa hårdvarufel som återkommit under arbetets gång, hade vi även något problem med det utgivna emuleringssystemet. Detta löste sig däremot då vi till slut bytte emuleringssystem.

Ett stort problem vi hade vid de första provkörningarna av bilen var att motorerna störde ut resten av komponenterna. Detta gjorde så att vår PC fick ett icke aktuellt värde, varpå applikationen hängde sig. För att minska störningarna från motorerna löddades två stora kondensatorer på parallellt över varje motor.

Vi hade även lite problem med vår AD-omvandlare då denna genererade avbrott i allt för korta intervall. Detta bidrog till att vi körde avbrottsrutinen hela tiden varpå vi inte fick någon tid över till att köra den egentliga programkoden. Med en långsammare klocka in till AD-omvandlaren kunde vi bestämma hur snabbt den skulle arbeta. Detta bidrog till att applikationen fick betydligt mycket mer tid till att jobba med den egentliga programkoden.

6. Resultat

Vid iakttagelser av den slutgiltiga applikationen, avviker denna desto mer från tejpen än vad vi hade förhoppningar på. En bit in i testningen av applikationen insåg vi att vår kravspecifikation var en grovt felaktig uppskattning i hur exakt bilen skulle kunna köra. Men då vi fått insyn i hur motorerna och systemet fungerar i en verklig miljö måste vi ändå vara nöjda med de resultat vi uppnått. Då bilen kör efter tejp, avviker den ofta mer än en cm från tejpen, men då vi kan visa att den över huvudtaget följer den utsatta banan måste vi erkänna att vi är nöjda. Vad gäller återskapandet av den senast körda banan, har vi inte haft någon egentlig tid att finslipa denna funktion. Vid återskapandet av en speciell bana beter sig bilen väldigt tveksamt men vid enklare tester återskapar ändå bilen dess beteende på ett korrekt sätt. Varför den inte kan återskapa banan på ett egentligt sätt tror vi beror på tidsåtgången för de olika avbrottsrutinerna. Då vi följer tejp utförs fler åtgärder i avbrottsrutinen än då återskapning genom minnet sker.

Som avrundning är vi mycket nöjda med de resultat som uppnåtts, då vi i slutskedet var väldigt tveksamma på om vi skulle hinna med att implementera någon minnesrutin. Största delen av kursen har gått åt att få hårdvaran att fungera på önskat sätt. Skulle vi ha haft ytterligare tid, skulle vi haft större möjligheter att finslipa och förbättra mjukvarurutinerna i applikationen, varpå applikationen skulle bettet sig närmare våra initiella krav.