

KUNGLIGA TEKNISKA HÖGSKOLAN

Lego Mindstorms med NXC

Följa en linje

Jimmy Hytönen

2012-09-03

jhytonen@kth.se

Introduktionskurs Datateknik (II1303)

Sammanfattning

Uppdraget var att tillsammans med en kollega korrigera programkoden för en Lego Mindstorms robot med hjälp av programspråket NXC.

Uppdragets huvudsakliga syfte var att få roboten att följa en svart linje för att tillslut åka in i en vägg, därefter stänga av motorerna och skriva ut våra namn på displayen.

Genomförandet skedde genom klassisk parprogrammering där vi helt enkelt prövade oss fram och ändrade diverse kommandon för att sedan provköra roboten och kontinuerligt felsöka och analysera koden.

Efter en hel del 'trial and error' lyckades vi tillslut att få roboten, på ett effektivt sätt, att följa en svart linje.

Innehållsförteckning

Sammanfattning.....	2
1. Inledning	4
1.1 Bakgrund.....	4
1.2 Syfte och målsättning.....	4
2. Genomförande.....	4
3. Resultat.....	4
4. Analys.....	4
5. Diskussion.....	4
Referenser.....	5
Bilagor.....	5

1. Inledning

Uppdraget var att tillsammans med en kollega korrigera och felsöka en robots programkod i programspråket NXC(Not eXactly C).

Roboten i fråga var av typen Lego Mindstorms NXT och dess syfte i denna laboration var att kunna följa en svart linje tills den kör in i en vägg, då ska den känna av detta och stänga av motorerna och skriva ut våra namn.

Till hjälpmedel hade vi främst en användarmanual till hur man programmerar Lego NXT robotar med NXC samt handledare på plats för att svara på frågor.

1.1 Bakgrund

Kursens läroplan utgår från att vi inte programmerat något tidigare dock så förväntades det att man läst på lite om programmering innan utförandet av denna laboration, antingen genom föreläsningarna eller än hellre på egen hand med hjälp av guider på internet.

Eftersom programmering är en såpass stor och betydelsefull del på ICT-skolan så utfördes laborationen väldigt tidigt i programmets gång för att ge oss en inblick hur det är att felsöka och ge oss en försmak på hur man ska jobba och vilka begrepp som finns inom programmeringsspråket, i detta fall lite mer hårdvaru-nära programmering.

Som förberedelse inför denna laboration och rapport fick vi några introducerande föreläsningar om programmering och rapportskrivning samt en användarmanual i PDF-format för programmering av Lego NXT robotar.

1.2 Syfte och målsättning

Denna laboration utfördes helt i introducerande och förberedande syfte. För att vi ska lära oss hur man tillsammans med en kollega felsöker programkod men också för att vi ska bli mer bekväma med programmeringsspråket i sig.

Även för de som sysslat med programmering tidigare så gav denna labb också en utmärkt möjlighet till att tidigt i utbildningen pröva på mer hårdvaru-nära programmering.

Vi fick även passa på att arbeta på ett sätt som kallas parprogrammering, vilket är ett vanligt och effektivt sätt att arbeta på när man jobbar parvis inom programmering.

2. Genomförande

Innan laborationens genomförande hade jag förberett mig genom att ladda ner mjukvaran och programkoden till datorn. När vi kopplade in roboten upptäcktes det också att vi missat att ladda ner drivrutinerna, vilket vi gjorde snabbt och enkelt. Dessa kunde laddas ner från Bilda(en hemsida som KTH använder för vissa kurser, där kan man lägga upp arbeten, hitta kursmaterial, föreläsningar m.m.)

När detta var gjort var det bara att sätta igång arbetet.

Det första vi gjorde var att provköra roboten för att se vad den gjorde för fel. Roboten ignorerade den svarta linjen helt och åkte bara runt ett varv för att sedan snurra åt andra hållet med en snabbare hastighet.

Efter provkörningen så satte vi oss ner och började bit för bit analysera koden för att hitta felet. Vi upptäckte väldigt tidigt att funktionen som hämtar värdet från ljussensorn var kopplad till fel motor.

Strax därefter upptäckte vi också att funktionen som skulle få roboten att följa linjen såg märklig ut, den innehöll 2 st if-satser som hade olika villkor men exakt likadana kommandon för både *true* och *false*. Vilket var ganska motsägelsefullt om den ska ha olika beteenden beroende på om den är på svarta linjen eller inte.

Vi fortsatte därefter med lite s.k. 'happy hacking', dvs att ändra kommandona lite på måfå för att sedan provköra och se hur den betedde sig.

Vid andra testkörningen så betedde den sig märkligt nog precis likadant som innan, alltså att köra runt ett varv för att sedan snurra i andra riktningen.

Vid den här tiden var det dags för oss att byta position, min kollega satte sig framför datorn för att skriva kod medan jag satte mig bredvid för att observera och ge feedback. Det är detta som kallas parprogrammering, att 2 personer byter positioner då och då där den ena skriver medan den andre sitter bredvid och analyserar vad den andra skriver och kommer med feedback.

Det tog många testkörningar och omkodningar innan vi kom fram till vad felet berodde på, vi upptäckte så småningom en märklig rad i main-task där en funktion vid namn Dance() anropades. Då funktionens enda funktion var att snurra runt så var den helt överflödiga därmed kommenterade vi bort funktionens anrop.

Först då så reagerade roboten på vad ljussensorn kände av för färg, därefter gick det mycket fort att hitta rätt kommandon och lämpliga hastigheter för att få roboten att följa linjen.

När vi väl fått den funktionen att fungera så återstod kravet att den skulle stänga av sig själv med hjälp av trycksensorerna (vilket fungerade från början) samt skriva ut våra namn i displayen.

Det tog sin tid för oss att få namnen utskrivna på rätt rad, mest pga. Att vi missade det faktum att raderna hade fördefinierade värden. När vi väl fick reda på detta gick allting mycket fort och vi var i princip klara med uppgiften.

3. Resultat

Laborationen var lyckad! Vi lyckades uppfylla alla kraven som ställdes. Nedan så listar vi de slutgiltiga ändringar som gjorde i programkoden.

Radnummer	Ny kod	Kommentar
2	#define SpeedSlow 70	Vi tyckte det gick lite för fort när den skulle svänga så vi sänkte hastigheten något.
35	"MIKAEL", "JIMMY"	Fylla i namnlista.
45	TextOut(0, LCD_LINE2 - (8*i)), names[i];	Tog bort -16 i räknaren då den gjorde så att namnen skrevs på fel rad.
68	LightIntensity = SensorRaw(IN_3);	Ändra så att SensorRaw hämtar data från ljussensorn istället för trycksensorn.
86	OnFwd(OUT_A, SpeedFast);	För att rätt hjul ska få rätt hastighet så att den svänger rätt.
90	OnFwd(OUT_B, SpeedSlow);	För att rätt hjul ska få rätt hastighet så att den svänger rätt.
114	//dance()	Ändrade så att detta booleska uttryck liknade det på rad 82 så att det vart lättare att följa programmet.

4. Analys

Förutom en lyckad laboration så hade vi väldigt kul på kuppen. Uppgiften kändes väldigt givande och stimulerande och lyckades även väcka ett bortglömt intresse för programmeringen.

Vad gäller parprogrammeringen så kändes det som ett väldigt bra sätt att arbeta på om man är 2 personer, det ger lite variation i arbetet och ökar produktiviteten.

Det fanns nog inte så mycket vi kunde ha gjort för att få arbetet att gå snabbare och effektivare än vi gjorde då det tog sin tid helt pga. Vår ovana med programmeriskt tänkande och problemlösande, hade vi fått ett motsvarande problem i dagsläget skulle vi nog gjort precis likadant med ända skillnaden att man skulle ha lite mer erfarenhet.

Det som ställde till det mest för oss var att vi inte förstod funktionen som skriver ut namnen på skärmen, vi hade missat att raderna hade fördefinierade värden och därmed förstod vi inte kalkylens funktion i for-loopen.

Att sedan roboten inte följde linjen berodde främst på 3 saker; Först, en onödig funktion anropades i main-task, dvs funktionen dance(). Denna gjorde att roboten kör runt utan att någonsin stanna eller ändra betéende. Det andra var att funktionen som skulle hämta värden från ljussensorn anropade fel motor, den anropade trycksensorn. Den tredje anledningen var villkorens kommandon i funktionen som skulle styra hjulen. Båda if-satserna hade exakt samma kommandon vare sig de gav 'true' eller 'false'.

5. Diskussion

Laborationens syfte och mål som var att introducera problemlösning med programmering var väl anpassad till den förväntade kunskapsnivån hos studenterna. Det förväntades av dom som aldrig

programmerat tidigare att läsa på lite inför denna laboration, och hade man gjort det så borde inte denna uppgift vara oöverkomlig.

Programmeringen skedde helt och hållet i NXC men då den är såpass lik programspråket C så får man lite kunskap om denne också. Att programmeringen också var hårdvaru-nära är ett stort plus.

Kommer ha stor nytta av allt detta inte minst på nästa projektarbete som också kommer kretsa kring Lego NXC, med skillnaden att den ska byggas och programmeras helt från grunden samt högre specifikationskrav.

Referenser

Bilda KTH; Mall-rapport: <https://bilda.kth.se/node.do?id=19392435>

”Programming LEGO NXT Robots using NXC”; skriven av Daniele Benedettelli

Bilagor

Journal

In your journal, you can take notes and analyze your progress in your studies and your personal development. You can use the search tools to find older notes.

[Read more about how to use the diary for your continuous career development during your time of studies.](#)

[Write a personal note ...](#)

Personal note | 30 August at 17:51

Har nu för första gången programmerat en Legorobot!
I efterhand i reflekterande stund så var det ju inte så himla svårt.. egentligen!
Men tog ett tag innan man hamnade på rätt spår.. Om man inte har några förkunskaper eller inte läst och testat något alls så kan jag tänka mig att det här skulle bli väldigt svårt.
Så om man aldrig har programmerat förut så är mitt råd att man MÅSTE prova och trixa lite med enkla loopar, funktioner etc.

Hursomhelst, tyckte det var en jättekul uppgift och jag ser fram emot och mecka ihop en racerbil i projektarbetet :)