

En introduktion till felsökning av en LEGO NXT robot

En grundläggande rapport om programmering
och felsökning i NXC

Rodbeh Farokhynia

2014-09-04

rodbeh@kth.se

Introduktionskurs i datateknik (HT14) II1310

Sammanfattning

Har du någonsin funderat på hur en sådan simpel grej som att trycka på din touch skärm på mobilen har för mekanik bakom? Tekniken bakom en sådan funktion är svår att förstå, ännu svårare blir det att förstå ifall man inte kan något alls om programmering.

I denna rapport kommer du således få en inblick om hur det är att felsöka i programmeringskoden för en LEGO NXT robot.

Du kommer att få läsa om vad som var fel på vår kod, hur vi åtgärdade felen samt hur ett enkelt litet fel kunde leda till stora förändringar i robotens beteende.

Dessutom kommer du att få förståelse för vad en del koder har för betydelse samt hur de är sammankopplade till varandra för att programmet ska fungera som det ska.

Det kommer även diskuteras om hur en del aspekter var bra respektive dåliga med laborationen samt vad laborationen för mig personligen hade för betydelse, både nu under studietid men även senare under yrkeslivet.

Innehållsförteckning

1. Inledning.....	3
1.1 Bakgrund.....	3
1.2 Syfte och målsättning.....	3
2. Genomförande	4
3. Resultat	5
4. Analys	6
5. Diskussion.....	6
Referenser	7
Bilagor	8

1. Inledning

I vår vardag möts vi av elektronik mer eller mindre hela tiden, men det är sällan vi sätter oss ner och tänker på hur mekaniken bakom de elektroniska prylarna vi använder oss av faktiskt fungerar. I denna rapport kommer du således att läsa om hur man i par lyckas programmera en LEGO NXT robot utan att ha programmerat något tidigare.

Programmeringen gjordes i NXC (Not eXactly C) med syftet att lära sig att felsöka koder i programmet, vilket i sin tur ledde till att målsättningen blev att få ett färdigt program där roboten utförde uppgiften enligt det man önskade sig. I rapporten får du dessutom läsa om problem som vi stötte på och hur vi kom tillväga att lösa dessa i programmet.

Programmering gjordes i ICT-skolan KTH i Kista med anledning av att kunskaperna man får, börjar i ens utbildning som sedan löper vidare under yrkeslivet.

1.1 Bakgrund

Programmering är en väldigt viktig faktor i ens utbildning när man pluggar på ICT-skolan.

För att lära sig att programmera behöver man börja från något grundläggande för att succesivt arbeta sig uppåt mot en mer avancerad nivå. Detta är viktigt då all form av programmering inom ens utbildning leder till färdigheter som man sedan har nytta av i arbetslivet. Som en ingenjör måste man kunna programmera, inte bara för att det ingår i ens utbildning, utan också för bli bekant med hur det kan vara under arbetslivet. Att programmera en LEGO NXT robot kan därför ses som början på den långa vägen man har både under och efter ens utbildning på ICT-skolan.

1.2 Syfte och målsättning

Syftet med denna laboration är att man ska få en introduktion till programmering i form av par programmering. Uppgiftens syfte är att man ska få träning i hur det är att felsöka koder i ett program och därmed få förståelse för hur små enkla fel kan ge oväntade resultat i körningen av en robot.

Syftet är även att när man väl felsöker koder förbättrar sitt noggrannhetstänk.

Målet med laborationen är att man får ett färdigt program, i NXC, att fungera för att LEGO NXT roboten ska utföra sin uppgift på önskat sätt.

Mina egna mål med laborationen var att lära mig hur man från grunden lär sig att programmera, hur man lär sig att förstå koder samt hur man går tillväga att lösa problem man stöter på.

Helhetssyftet med att skriva en rapport är att träna på rapportskrivning på ett ingenjörsmässigt sätt. Man ska kunna beskriva en uppgift utförligt, motivera ens resultat på ett trovärdigt sätt samt undvika överflödiga beskrivningar.

2. Genomförande

Till och börja med så gick jag igenom Labb-PM:et för att förstå vad laborationens syfte, mål och uppgift var samt hur man skulle gå tillväga.

Innan laborationen laddade jag ner en drivrutin, för att Windows skulle hitta roboten, samt mjukvaran BricxCC för att kunna programmera roboten. Även den färdigskrivna .nxc-filen som vi skulle felsöka koder i laddade jag ner i förväg för att skumma igenom koderna. Materialet laddade jag hem från www.bilda.kth.se.

Uppgiften var att vi i par skulle programmera koden till en LEGO NXT robot så att den skulle följa en svart-tejp bana på golvet till dess att den möter ett hinder med sina rörelsesensorer. När den väl gjorde det så spelades en melodi upp och gruppmedlemmarnas namn skulle då stå på LCD-skärmen. Väl under laborationstillfället så valde jag och min partner en robot och det första vi gjorde var att jämföra vår robot med en bild som fanns att ladda ner på Bilda.

Detta gjorde vi för att kolla ifall alla sladdar satt i rätt port. Därefter kopplade vi roboten till datorn med en USB-kabel och startade upp BricxCC. Efter detta öppnade vi upp .nxc-filen och tog oss an vår uppgift som var att felsöka koder, rätta till de och få roboten att utföra sin uppgift.

Vi började med att granska hela koden för att förstå så mycket som gick utan att vi ändrade på det. Detta gjorde vi för att bekanta oss med koderna, trots att vi inte förstod så mycket.

Vi såg därefter till att föra över programmet till roboten, vilket inte gick. Vi upptäckte därefter med hjälp av vår lärare att ena sladden på roboten var felkopplad. Efter detta testade vi att föra över programmet igen och denna gång lyckades vi.

Vår metod till att lösa uppgiften var ändra koderna och sedan testköra roboten för att se hur roboten reagerade. Detta hjälpte oss att förstå ifall vi skulle ändra/ta bort koder. Körde roboten enligt det den skulle på tejp-banan så behöll vi koden vi ändrade på. Blev det däremot sämre resultat med vår ändring så ändrade vi antingen tillbaka eller tog bort koden helt.

Genom logiskt tänkande, uteslutningsmetod samt noggrannhetstänk lyckades vi tillslut lösa uppgiften.

3. Resultat

Efter att ha uteslutit koder/lagt till nya koder så resulterade det i att roboten med hjälp av sin ljussensor identifierade den svarta-tejp banan som fanns på golvet. När den väl hade identifierat banan så kunde roboten med sin nya hastighet, som var lågt, köra sakta genom banan för att sedan stöta på ett hinder. När den väl stöter mot hindret så låter en melodi och gruppledarnas (min och Andreas) namn blir synlig på LCD-skärmen.

Tabellen visar på vilka radnummer vi ändrade på, hur vår nya kod ser ut samt varför vi ändrade på dessa rader/koder.

Radnummer	Ny kod	Kommentar
2	<code>#define SpeedSlow 40</code>	Vi ändrade på siffrorna för hastigheten, då roboten hade lättare att följa svarta-tejp banan.
3	<code>#define SpeedFast 60</code>	Vi ändrade på siffrorna för hastigheten, då roboten hade lättare att följa svarta-tejp banan.
34, 35, 36, 37	<code>string groupMembers[] = { "rodbeh", "andreas" };</code>	Vi ändrade denna kod eftersom int står heltal, medan string står för text och detta behövde ändras för att vi våra namn skulle visas på skärmen.
46	<code>TextOut(0, (LCD_LINE2 - (8*i)), names[i]);</code>	Vi tog bort (-16) i och med att det endast fanns 8 lines (rader). När vi då hade (8*i - 16) så hamnade namnen på samma rad istället för på olika. Genom att ta bort (-16) så fick vi namnen på olika rader.
76	<code>lightIntensity = SensorRaw(IN_3);</code>	Den var kopplat till fel port, så vi byte från (IN_1) till (IN_3) för att det skulle bli rätt.
76	<code>lightIntensity = SensorRaw(IN_3);</code>	Den var kopplat till fel port, så vi byte från (IN_1) till (IN_3) för att det skulle bli rätt.
92 - 102	<code>{ OnFwd(OUT_A, SpeedFast); } else { OnFwd(OUT_A, SpeedSlow); } if(lightIntensity > BotThreshold)</code>	Vi ändrade koderna beroende på intervallen och robotens ljuskänslighet för färgen svart. Detta gjorde vi för att roboten skulle följa den svarta-tejp banan.
92-102	<code>{ OnFwd(OUT_B, SpeedFast); } else { OnFwd(OUT_B, SpeedSlow); }</code>	Vi ändrade koderna beroende på intervallen och robotens ljuskänslighet för färgen svart. Detta gjorde vi för att roboten skulle följa den svarta-tejp banan.
115	<code>// dance();</code>	Vi la till // för att koden "dance" inte skulle läsas. Den nya koden gjorde så att roboten inte skulle snurra runt sig själv eftersom detta gjorde ett hinder för att roboten skulle följa svarta-tejp banan med sin ljussensor.

4. Analys

Resultatet kan man dela generellt in i fyra sammankopplade funktioner. Tar man och kollar på .nxc-filen ser man att det är en funktion för vilka namn som ska visa sig på robotens skärm, en för på vilka rader, en för när namnen ska visa sig på skärmen samt den viktiga funktionen som visar på ett intervall mellan hastighet och robotens förmåga att avläsa färgen svart.

Till en början hade vi svårt att förstå koderna och således visste vi inte hur vi skulle börja felsöka. Fast med lite hjälp från läraren som förklarade vad vissa variabler, loopar osv. betydde kunde vi börja felsöka.

Efter att ha förstått någorlunda vad en del koder betyder så kan man analysera och förklara vad som låg bakom robotens beteende. Analyserar man t ex vad som stod före/efter ändring av koder som skulle visa sig på LCD-skärmen, så beror det helt och hållet på vilken variabel man använder sig av. I vårt fall stod det **int**. Detta gav upphov till att siffror visades på skärmen istället för våra namn. Rätt kod som skulle stått i detta fall var då **string** som används då man vill skriva text.

Kollar man på **loopar** och **villkor** så förstår man tillslut att de går hand i hand. Detta beror på att en loop är beroende på ifall ett villkor är sant eller falskt. Analyserar man då t ex intervallet mellan hastighet och färgkänsligheten för robotens sensorer insåg man de hänger ihop för hur roboten betar sig på svarta-tejp banan.

Det som gick bra var att efter att man verkligen kollade igenom koden noggrant och försökte få förståelse för koderna, så ändrade man enligt resultatet man ville få. Dessutom var det lättare att förstå vart man gjorde fel i koden när man testkörde roboten.

Det som gick dåligt i början var att man försökte vara för snabb och lösa felen i koden utan att ha fått förståelse för koderna. En annan dålig grej var att man borde fått lite mer programmeringskunskaper, för att förstå programmeringsspråket.

5. Diskussion

Syftet med laborationen var framförallt att vi i par skulle få en grundläggande förståelse för programmering i NXC och lära oss felsöka koder. Målet var då att genom felsökning få ett fungerande och färdigt program.

Vi lyckades nå både kursens syfte och mål, däremot hade jag hellre velat lära mig själva programmeringsspråket från grunden. Nu felsökte man endast i koden och ändrade koderna. Att vi lärde oss att endast felsöka koden innebär inget dåligt, men om vi hade lärt oss programmeringsspråket och lärt oss skriva ett program själv skulle det vara mycket bättre.

Målet för mig specifikt var att lära mig programmera från grunden, få förståelse för koderna samt hur jag skulle gå tillväga att lösa uppgiften.

Jag kan inte säga att jag lärde mig programmera från grunden, trots att jag skrev om koderna. Däremot förstod jag koderna och lärde mig ett sätt att lösa problemen jag stötte på, vilket kändes väldigt lärorikt.

Problem som uppstod var bland annat att man var för snabb med att ändra koderna då man inte riktigt förstod vad koderna innebar. Detta ledde till ännu fler oönskade resultat som roboten utförde.

Det man lärde sig var att man behövde förstå koderna någorlunda och vara noggrann för att få önskat resultat. Man lärde sig att alla funktioner, loopar, variabler osv. är beroende av varandra för att roboten ska göra som man vill. Det är en röd tråd genom hela koden när man väl förstår koderna. Även programmet NXC som man använde sig av var väldigt simpel gjord och man förstod ganska snabbt hur det fungerade. Det är ett enkelt program att navigera sig runt i, för nybörjare bland annat.

Denna laboration har stor betydelse, eftersom det är första gången jag personligen programmerar och får en inblick i hur det kan komma att se ut under andra studietillfällen och sedan successivt i ens framtidsyrke. Laborationen ger en snabb inblick i hur det kan se ut, hur man kan komma att arbeta i par eller större team, hur man stöter på problem samt löser de och allra viktigast hur språket man ska använda sig av ser ut och hur komplext det kan vara.

Referenser

- Labb-PM (2014-09-02)
<https://bilda.kth.se/courseId/11430/content.do?id=22224147>
- Föreläsning 1 (2014-09-02)
<https://bilda.kth.se/node.do?id=22301931>
- Föreläsning 2 (2014-09-02)
<https://bilda.kth.se/node.do?id=22305545>
- Föreläsning 3
<https://bilda.kth.se/node.do?id=22305546>
- BricxCC.zip (2014-09-02)
<https://bilda.kth.se/node.do?id=22224203>
- Drivrutin till LEGO NXT.zip (2014-09-02)
<https://bilda.kth.se/node.do?id=22224215>
- Kommentarsblad för laborationsuppgift.pdf (2014-09-02)
<https://bilda.kth.se/node.do?id=22224207>
- Linefollower.nxc (2014-09-02)
<https://bilda.kth.se/node.do?id=22224229>
- Monteringsspecifikation LEGO Mindstroms.png (2014-09-02)
<https://bilda.kth.se/node.do?id=22224209>
- Programming LEGO NXT Robots using NXC.pdf (2014-09-02)
<https://bilda.kth.se/node.do?id=22224206>
- Rapportmall.doc (2014-09-02)
<https://bilda.kth.se/node.do?id=22300661>

Bilagor

Bilaga 1: Skärmdump på mitt dagboksinlägg.

