

Felsökning av kod i NXC

En bra början till förståelse för programmering
/LEGO Mindstorms Robot (NXT)

Anne Golinski

2012-09-03

golinski@kth.se

Introduktionskurs i datateknik (H12) II1310

Sammanfattning

Rapporten handlar huvudsakligen om att lokalisera fel i NXC-kod och få en LEGO Mindstormsrobot att utföra efterfrågade uppgifter. Detta med en utförlig del om genomförande samt en resultatdel som strukturerar upp ändringar som gjorts i koden. Rapporten innefattar även tankar och idéer kring kursen *Introduktionskurs i datateknik (H12) II1310* samt par-programmering.

Innehållsförteckning

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

1. Inledning

Denna rapport kommer inkludera felsökning av kod i NXC till *LEGO Mindstorm*-robotar samt tankar och idéer om genomförande, personligt samarbete och framtida utsikter för nyfunna kunskaper att etablera i studie- respektive arbetslivet.

1.1 Bakgrund

För att man ska kalla sig för student på ICT-skolan anser jag att man ska känna till hur en simpel kod ser ut. Som framtida ingenjör i en växande IT-värld är det givetvis fruktansvärt viktigt med data-kunskaper, i alla fall om man vill komma i kontakt med den breda arbetsmarkanden.

Att börja i NXT och programmera enklare kommandon innefattar en grundläggande förståelse för ämnet vilket leder till större chans till framtida insikt i programmeringens värld.

1.2 Syfte och målsättning

Syftet med uppgiften är att på ett roligt och relativt enkelt sätt introducera programmering i form av par-programmering.

Syftet är också att introducera arbetsgången vid ingenjörsarbete samt ge träning och arbetsvana i de IT-system för utbildning som finns vid ICT-skolan.

Uppgiften syftar även till att ge träning i felsökning och testning och därmed ge en grund till effektivt programmeringsarbete i framtiden. Förhoppningen är även att studenterna ska få en förståelse för hur små enkla fel i koden kan ge oväntade resultat vid körning, och på så sätt bidra till ett noggrannhetstänk.

Målet med uppgiften är att få ett färdigt program skrivet i NXC (Not eXactly C) att fungera och därmed få en LEGO-robot att utföra sina uppgifter på önskat sätt. Mina egna mål med kursen samt laborationen är att få förståelse kring hur en kod ser ut då jag aldrig ens öppnat en kommandotolk innan kursens början.

Målsättningen med laborationen i sig är att ändra i koden och på så vis på roboten att följa en linje, skriva ut gruppmedlemmarnas namn i dess prompter, stanna när den krockar med ett hinder och då spela en liten trudelutt.

2. Genomförande

Laborationen började egentligen innan laborationstillfället med genomläsning av "Programming LEGO NXT Robots using NXC.pdf" som finns att hämta på kurshemsidan i Bilda. Även anslutning till skolans lokala nätverk *Eduroam* etablerades med hjälp av nätverksnyckel från *KTHOPEN*.

Väl vid laborationstillfälle laddas *Bricx Command Center* ned och installeras, vilket är programmet som används för att redigera källkod samt kompilera koden och flasha roboten. Sedan hämtades "linefollower.nxc" hem från kurshemsidan i Bilda, alltså dokumentet för själva koden. Då roboten var uppkopplad via USB-anslutning samt *Linefollowers.nxc* öppnat i *Bricx Command Center* kunde vi börja granska koden. För att lättare komma ihåg ändringar som görs hade vi en anteckningsmall med radnummer, ny kod samt ett kommentarsfält för de ändringar som görs.

Vi hjälptes åt att granska samt göra ändringar i koden samtidigt som vi gav feedback och konstruktiv kritik till kodningen i sig.

Det första som genomförs är att testa hur roboten uppförde sig utan ändringar, vi såg att roboten bara snurrade runt och insåg att det måste vara en felprogrammering gällande koden för motorerna. Vi ändrade;

```
84          OnFwh(OUT_A, SpeedFast);  
92          OnFwh(OUT_B, SpeedSlow);
```

Viket resulterade till att roboten nu körde rakt fram.

Sedan kommenterade vi bort rad 97,102 samt 115 för att vi inte förstod vad de skulle vara bra för.

Vi testkörde och fick samma resultat.

Vi behövde nu få roboten att hitta linjen. I rad 2 samt 3 sänkte vi hastigheten i hopp om att roboten skulle hitta linjen lättare vid en lägre hastighet. Test. Samma resultat.

Istället kikade vi på koden igen för att lokalisera ljussensorn och insåg att

```
68          lightIntensity = SensorRaw(In_1);  
ej stämde överrens med dess placering i "main task". Efter en ändring till  
68          lightIntensity = SensorRaw(In_3);  
hittade roboten linjen.
```

Vi gick vidare till att få in våra namn på robotens prompter och insåg att enklaste sättet att lägga in det på skulle vara att ändra om i rad 35. Vi placerade också ett *kommatecken* mellan namnen för att få dem på olika rader. Test. Dock såg det fortfarande lite otydligt ut på promptern vilket resulterade till att vi plockade bort -16 på rad 45 för att rätta till ekvationen.

Vid nästa test genomförde roboten de uppgifter vi önskade genomföra.

3. Resultat

| Radnummer | Ny kod | Kommentar |
|-----------|--|---|
| 84 92 | OnFwh(OUT_A, SpeedFast); OnFwh(OUT_B, SpeedSlow); | För att ändra till olika outputs vid varsin sida om "else" |
| 97 102 | /*void dance() */ | Kommenterade bort dance |
| 115 | //OnFwd(OUT_AB, SpeedSlow) ; | Kommenterade bort för det såg suspekt ut att ha den under "main task". Skulle den varit med skulle den varit innan i koden. |
| 2 3 | SpeedSlow 20 SpeedFast 40 | Sänkte hastigheten i hopp om att lättare lokalisera linjen |
| 68 | lightIntensity = SensorRaw(In_3); | För att få rätt motsvarighet i "main task" |
| 35 | "Helanne", "State" | Lade in våra namn med radbyte. |
| 45 | TextOut(0, (LCD_LINE2 - (8*i)), names [i]); | Tog bort -16 för att rätta till ekvationen. |

4. Analys

Robotens beteende grundade sig i luringar och fel i koden som var rätt lätta att hitta så länge man satte sig ner och läste vad som egentligen föregick i koden. Att vi efter varje ändring gjorde ett test gav oss också en klar överblick över vad som fungerade och vad som inte gjorde det.

Rad 84 samt 92 angav hur robotens båda motorer skulle arbeta.

Att kommentera bort "dance"-funktionerna på rad 97-102 kändes också rätt självklart då det ej var robotens syfte.

Rad 115 hade ej behövts plockas bort då programmet avslutats innan den kom till skott.

Att sänka hastigheten på rad 2 samt 3 gav oss en bättre överblick över hur roboten betedde sig.

Att sedan hitta felet i rad 68 vad det som verkligen hjälpte oss att få roboten att genomföra en av sina uppgifter.

Det mest uppenbara var på rad 35 där namnen uppenbarligen behövde skrivas ned för att de skulle komma in i koden. Likaså att plocka bort -16 ur ekvationen då den ej åsyftade något radnummer i loopen.

5. Diskussion

Laborationen gick över lag riktigt bra, att jobba med någon trevlig person som man inte vanligtvis umgås med genererar i en variation av tankar som kanske inte vanligtvis uppstår om man jobbar med någon som tänker exakt likvärdigt med en själv.

En trevlig arbetsmiljö genererar i sig i ett "öppet forum" där tankar och idéer kan uttalas högt och granskas utan nedslående kommentarer utan istället följas av uppmuntran och konstruktiv kritik.

Då mitt personliga mål med laborationen var att få en förståelse för hur en kod ser ut anser jag att den var riktigt lyckad. Jag kan numera inte bara förstå hur en kod är uppbyggd utan även skriva kod i NXC på grundläggande nivå.

Just dessa kunskaper är definitivt något som jag kommer ha användning av i senare skede, både i studier samt i arbetslivet. Har man lärt sig ett programmeringsspråk är det i sin tur enklare att lära sig ett ytterligare, precis som med vanliga talspråk. Detta eftersom man kan se sammanhang mellan det man redan kan samt de nya som man lär sig. Med NXC i bakfickan anser jag att jag har en bra grund att stå på för att lära mig fler programmeringsspråk.

Just gällande programvaran *BricxCC* så var den simpel att installera och komma igång med, *Linefollowers*-koden var även den lätt att öppna i *BricxCC*. Själva koden i sig var enligt egen mening lätt att förstå sig på och inte allt för komplicerad att göra ändringar i.

När det gäller tillvägagångssättet och genomförandet av laborationen tror jag att det är viktigt att komma väl förberedd så att man på något sätt har en viss förståelse över vad som ska göras.

En annan sak att lägga märke till i vårt genomförande är att vi testade varje liten ändring för att se vad vi fick för output, snarare än att göra många ändringar på samma gång för att sedan testa. Att jobba på ett sådan "trial and error"-sätt tar lite längre tid än att på van hand fixa alla felkodningar för att sedan testa. Dock krävs det också en högre kunskapsnivå och erfarenhet kring mjukvaran för att jobba på det tidssparande sättet, något som förhoppningsvis kommer i framtiden.

Kikar vi på elementet *par-programmering* var det något som vi mixade in i laborationen i sig då vi gav varandra feedback och konstruktiv kritik under kodningens gång. Detta fungerade väldigt bra för oss men andra kanske föredrar att avsätta egen tid för par-programmeringen och rikta in sig på den snarare än att sammanblanda den med laborationen.

Hela kursen *Introduktionskurs i datateknik (H12) II1310* har givit en bredare insikt i hur programmering faktiskt artar sig från det att man skriver in en kod till det att åsyftad maskinen utför sin uppgift. Med givna mallar samt strukturerade listor över mjukvara samt hårdvara som krävs för att genomföra laboration har det varit en väldigt givande kurs, som man hela tiden känt att man klarar av. Detta trots att förkunskaperna innan själva kursstarten varit mycket begränsade.

Referenser

Kursmaterial i Bilda, KTH. 2012-09-03

<https://bilda.kth.se/courseId/8498/content.do?id=19121762>

Dokument;

[BricxCC.zip](#)

[Kommentarsblad för laborationsuppgift.pdf](#)

[linefollower.nxc](#)

[Programming LEGO NXT Robots using NXC.pdf](#)

[Mall rapport.docx](#)

Labb-PM I Bilda, KTH. 2012-09-04

<https://bilda.kth.se/courseId/8498/content.do?id=19150198>

Bilagor

Print Screen från dagboksinslägg skrivet i den glatt uppfyllda stund strax efter laborationen.

