

KTH KISTA

Robot på linje

Programmering i NXC

Simon Johansson

2013-09-05

simoj@kth.se

Introduktionskurs Datateknik || 1310

Sammanfattning

Laborationens syfte är att tillge mig fördjupande kunskaper om hur en robot programmeras, få mer erfarenhet inom programmering i NXC, problemlösning och felsökning i ett redan färdigskrivet program samt att parprogrammera. Målsättningen är att roboten ska följa en svart tejp på golvet och skriva ut medarbetarnas namn på displayen. För att genomföra laborationen lärde jag mig först språkets grundläggande syntax, detta för att kunna felsöka kodningen och dess fel. Jag hittade alla fel och ordnade till dessa. Samtidigt lärde jag mig att programmering kräver förståelse och noggrannhet. Men för att verkligen få mer insikt i programmeringens verklighet hade jag önskat en mer krävande egenskriven kod samt mer erfarenhet med att parprogrammera.

Innehållsförteckning

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

1. Inledning

Robotens huvudsakliga uppgift är att följa den markerade linjen på golvet. Laborationen ska utrusta mig med erfarenheter inom programmering, samarbete och felsökning, vilket krävs för att slutföra uppgiften.

1.1 Bakgrund

Som ingenjör krävs det i allra högsta grad fördjupande kunskaper inom områdena; mjukvara, hårdvara, problemlösning, samarbete och felsökning. När utbildningen är avslutad förväntas vi bemöta arbetslivets krav, där det är vi som utvecklar framtiden. Genom att genomföra denna laboration kommer vi ett steg närmare de kunskaperna som eftersöks av företag. Många av dagens företag jobbar i olika grupper där parprogrammering krävs. Denna laboration är därför ett ypperligt tillfälle för oss att bli introducerade till vårt nya arbetssätt.

1.2 Syfte och målsättning

Syftet med laborationen är att utöka förståelsen av programmering i NXC och hur en ingenjör på ICT jobbar. För att i framtiden bekanta mig i det nya språket, förstå kopplingen mellan programmet och NXT-roboten samt sökning av felkod. Syftet är att få kunskap om parprogrammering för att testa på den metodiken. Där den ena personen fungerar som observatör när den andra programmerar för att på så sätt producera en säkrare och bättre kod.

Målsättningen är att använda oss av ett färdig skrivit program i NXC för att få LEGO roboten att följa en svart eltejpbit på golvet och sedan skriva ut mig och min kollegas namn på NXT-robotens display korrekt samtidigt som den spelar ett ljud.

2. Genomförande

Närvarade på föreläsningar för att lära mig språket i NXC. Information om vilka verktyg som behövdes för att kunna genomföra laborationen hämtades på KTH:s hemsida Bilda under fliken kursmaterial. Följande filer införskaffades:

- BricxCC.zip
- Drivrutin LEGO NXT.zip
- Kommentatorsblad för laborationsuppgift.pdf
- Linefollower.nxc
- Programming LEGO NTX Robots using NXC.pdf
- Rapportmall

Började med att installera "BricxCC.zip" som installerar programmet som vi programmerar NXC i. "Drivrutin LEGO NXT.zip" installerade drivrutinen på datorn, så datorn hitta roboten genom USB-

kontakten. Läste "Programming LEGO NTX Robots using NXC.pdf" för att öka min teoretiska kunskap om programmeringen i NXC och dess interaktion med LEGO NTX Roboten.



NTX Roboten

Laborationen genomfördes måndagen 2013-08-26 i sal 430 där jag blev medlaborant med Daniel. Vi kom överens om att jag skulle börja att programmera och han agera som observatör. Vi kopplade in NTX Roboten i datorn och körde igång programmet "linefollower" för att utforska hur roboten reagerade på det nuvarande programmet.

Vi använde oss av metodiken "Trail and Error" för att se var i funktionerna som felet låg. Det gjordes därför ändringar för att köra programmet för att observera vilka variabler och funktioner som inte stämde. I 30 minuter programmera jag, innan vi bytte plats med varandra. Vi modifierade färdigt hela programmet tills allt var godkänt. Sedan utförde vi en egen extra uppgift.

3. Resultat

Resultatet av laborationen blev som tänkt och vi fick roboten att följa den svarta eltejpen och skriva ut våra namn på NTX Robotens display samt spela en lätt melodi plus en extra ljudfil. Exakta ändringar i programkoden hittas i tabell 3.1. Skärm dumpar på ändringar finns i bilagorna 1.1-1.4.

Tabell 3:1 visar ändringar som gjordes i programmet.

Radnummer	Ny Kod	Kommentar
2	SpeedSlow 60	Ändrades så att robotens hastighet var lite lägre
3	SpeedFast 80	Ändrades så att robotens hastighet var lite lägre
34	String groupMembers	Tog bort int då den inte kunde ta in text
35	"Simon"	Eftersom vi bytte int så ersatte vi med text
36	"Daniel"	Eftersom vi bytte int så ersatte vi med text
46	TextOut(0,LCD_LINE2 + (8*i-7),groupMembers[i]);	Bytt tecken från – till +. Ändrade även 16 till 7.
50 och 116	/*void dance() */	Kommenterade bort dance() funktionen
76	lightIntensity = SensorRaw(IN_3);	Här var sensorn fel. Den skulle anropa 3:an istället för 1:an
90	if(lightIntensity > TopThreshold)	Från att varit < bytte vi till >
96	if(lightIntensity < BotThreshold)	Från att varit > bytte vi till <
94	OnFwd(OUT_B, SpeedSlow);	B istället för motor A
100	OnFwd(OUT_A, SpeedFast);	A isället för motor B
115	PlayFileEx("HASTA.rso",4,0);	Spelar upp ljudet efter att sensorn träffats

Källa: Kommentarsblad för laborationsuppgift

Tabell:3.1

4. Analys

Vissa funktioner krånglade vilket ledde till att roboten inte genomförde programmet korrekt. Funktionen "dance()" hade inte någon tillämpning utan det den gjorde var att få roboten att åka framåt i tre sekunder, vilket inte hade någon användbarhet. Funktionen "readLightSensor ()" hade

sensorvärdet 1 på rad 76 som resulterade i att programmet inte läste av sensorn, alltså var sensorn avslagen vilket resulterade i att den inte kunde läsa av den svarta eltejpen.

Efter det uppstod problemet att den åkte runt i cirklar istället. Funktionen "task followline()" insåg vi hade den uppgiften, så vi ändra rad 94 och 100. Vi observerade även att det inte var någon skillnad på "if"-satsen och "else"-satsen, det var nämligen fel däck som i stället drev "else"-satsen. NTX roboten körde efter det framåt men följde inte linjen. I samma funktion ändrade vi rad 90 och 96 på deras större än och mindre än tecken eftersom vi misstänkte att detta kunde orsaka att roboten undvek linjen. I provkörningen efteråt klarade NTX roboten att följa svarta linjen.

Texten på displayen utskrevs fortfarande på fel sätt. Funktionen "void printNamesToScreen(string names[])" skrev ut namnen på displayen. Vi ändrade värdet ett flertal gånger tills vi kom fram till rätt funktion som syns på rad 46.

Extra uppgiften vi gjorde innebar att programmera så att roboten spelade upp en nedladdad ".wav" fil. Vi laddade ned en fil som var i rätt format och körde den i tools → sound conversion för att sedan gå in i NXT Explorer och ladda in filen i NTX roboten. Vi lade in funktionen "PlayFileEx("HASTA.rso",4,0)" i main programmet och fick det att fungera.

5. Diskussion

I syfte att lära sig att programmera i NXC tyckte jag att det gav bra kunskaper, speciellt i att porträttera hur svårt det kan vara att felsök i någon annans program. I programmering räcker det verkligen att endast en variabel är fel för att programmet ska bete sig konstigt och få många fel. Att missa en parentes, ett kolon eller ett värde kan leda till att delar av programmet blir odugliga. Något jag kommer ta med mig ifrån laborationen är att noggrannhet är av största betydelse. Allt för att undvika så många problem som möjligt och en bättre tankegång genom att jag får en bättre förståelse för programmet jag skriver.

Parprogrammering gav inte det resultat jag skulle önskat. Kände att det var jag som var ledare vilket gjorde att det mest kändes som att det var jag som föreslog saker. Sedan skedde inte så mycket egen programmering mellan oss utan vi fungerade mer som bollplank. Vilket är utmärkt vid felsökning av programmet, men delen att ena skulle vara observatör medan den andre programmerade tyckte jag inte vi fick testa tillräckligt. Jag tror att det skedde för att vi fick ett eget program där felsökning var lättare samt att vi inte behövde göra ett eget program. Det kunde ha blivit mer utvecklande om vi fått skriva en eller två egna funktioner för att programmet skulle fungera. Tillräckligt med information om hur en egen funktion skrivs fanns i programmeringens PDF:en "Programming Lego NXT Robots using NXC".

Det som var krångligt med laborationen var att identifiera vad som var fel. Exempelvis, tog det oss lång tid att inse att sensorn inte fungerade. Andra delen var att få ut rätt namn på displayen, vi förstod inte hur den fungerade men till slut på rad 46 hittade vi en ledtråd. Till en början testade vi helt enkelt en massa tal där vi började med att ta bort 7:an helt. Ingen förändring uppstod vilket ledde till att vi istället testade att halvera talet 16 och fick därigenom tillslut rätt resultat.

Metoden att testa sig fram även kallat "Trail and Error" användes men efter denna laboration tycker jag den är ytterst ineffektiv då man inte får någon förståelse för språket, funktionen och programmet även om vi till slut fick ihop programmet. Jag lärde mig att det är bättre att utveckla sin förståelse om vad funktionen påverkar än att testa sig fram. På så sätt kan man lösa uppgifter i programmering bättre, för ibland kan felen nämligen vara så många att det är svårt att identifiera dem med den metoden.

Tycker den logiska uppbyggnaden i NXC och C++ påminde om varandra då de har liknande variabler och språk. Det var väldigt smidigt att det fanns färdiga funktioner som man kunde anropa direkt på hårdvaran så det gjorde det mycket lättare att programmera. Tror de erfarenheter jag fått från denna laboration kommer göra att framtida programmering i NXC kommer bli lättare.

Tycker att denna laboration även gett mig kunskap om hur programmering fungerar mellan hårdvara och mjukvara. Laborationen ger även personer som inte är insatta i datorernas värld, en viss insyn i vad programmering är för något. Det är inte vanligt att genomsnittsmänniskan vet hur viktig en programmerare är. Jag tror därför att en demonstration som denna med hjälp av roboten skulle kunna få fler intresserade i ämnet programmering.

Referenser

1: Labb-PM

2: Kursmaterial

3: Programming Lego NXT Robots usins NXC.pdf

4: <http://www.villagegeek.com/html/wavfiles1.htm> (för ljudfilen)

5: Bild referens:

https://www.google.se/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&docid=x9pb6R3eunK6IM&tbnid=YzlebP6LOR27yM:&ved=0CAUQjRw&url=http%3A%2F%2Fboriscarblog.edublogs.org%2F2011%2F08%2F03%2Fnxt-robot-touch-sensor-challenge%2F&ei=Hi4qUpOqEYi44ASO_IGYcG&bvm=bv.51773540,d.bGE&psig=AFQjCNGOGpdebqqeKqFATmpxG2iLlYgHQ&ust=1378582426853072

Bilagor

Skärmdumpar 1.1-1.4

The image shows two screenshots of the Bricx Command Center software, displaying NXC code for two line follower programs. The top screenshot shows the initial code for 'linefollower2.nxc' and 'linefollower.nxc'. The bottom screenshot shows the continuation of the code for both files.

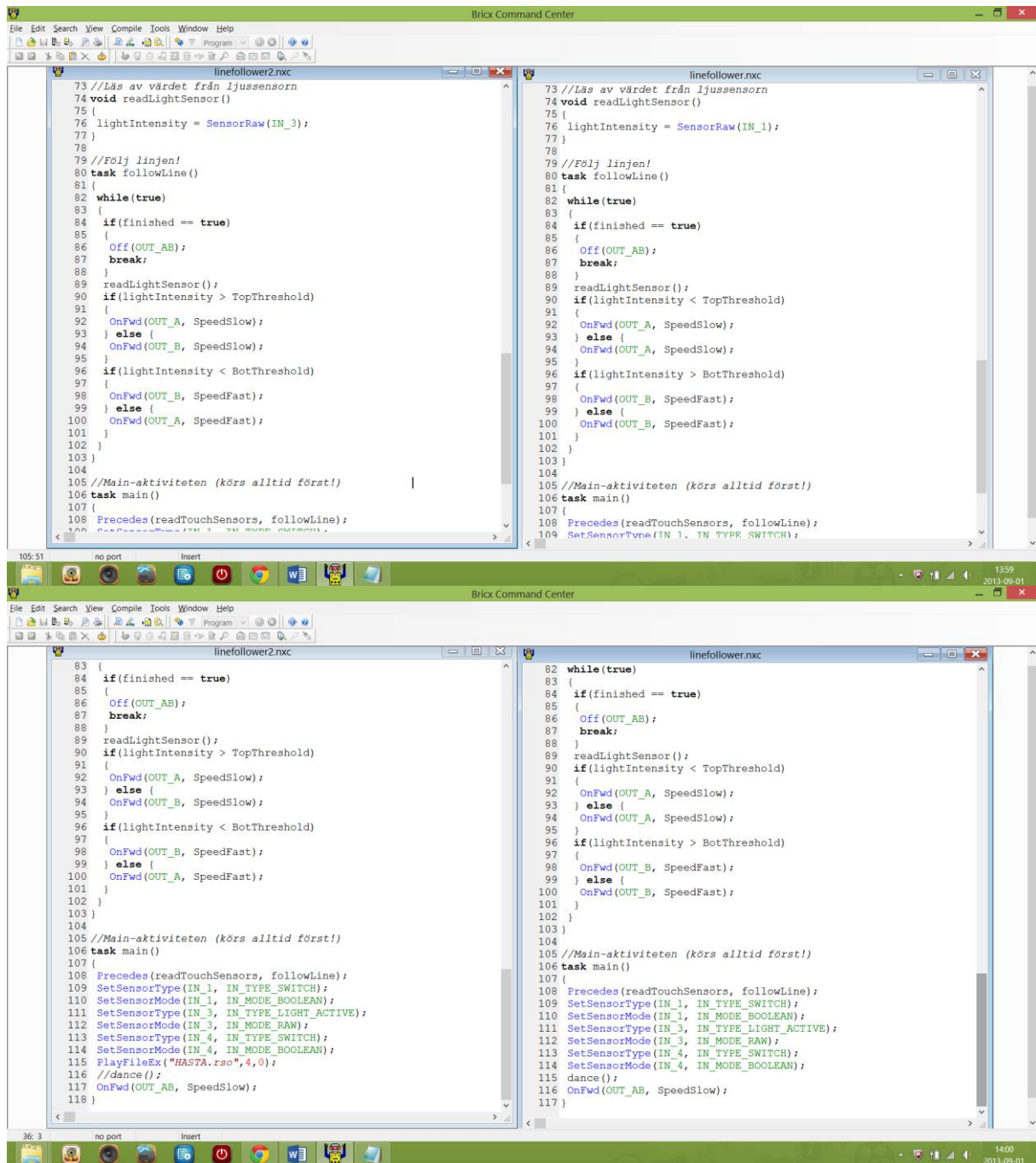
```

linefollower2.nxc
1 //Definierar hastigheterna som motorerna går i
2 #define SpeedSlow 60
3 #define SpeedFast 80
4
5 //Gränsvärden för ljussensorn (ändra ej dessa)
6 #define TopThreshold 630
7 #define BotThreshold 600
8
9 //Definierar raderna på robotens skärm
10 #define LCD_LINE1 56
11 #define LCD_LINE2 48
12 #define LCD_LINE3 40
13 #define LCD_LINE4 32
14 #define LCD_LINE5 24
15 #define LCD_LINE6 16
16 #define LCD_LINE7 8
17 #define LCD_LINE8 0
18
19 int lightIntensity;
20 bool finished = false;
21
22 //En liten trudelutt
23 Tone done[] = {
24   TONE_C4, MS_50,
25   TONE_E4, MS_50,
26   TONE_G4, MS_50,
27   TONE_C5, MS_50,
28   TONE_E5, MS_50,
29   TONE_G5, MS_50,
30   TONE_C6, MS_200
31 };
32
33 //Fyll i namnen på gruppmedlemmarna i listan
34 string groupMembers[] = {
35   "Simon",
36   "Daniel"
37 };

linefollower.nxc
1 //Definierar hastigheterna som motorerna går i
2 #define SpeedSlow 80
3 #define SpeedFast 100
4
5 //Gränsvärden för ljussensorn (ändra ej dessa)
6 #define TopThreshold 630
7 #define BotThreshold 600
8
9 //Definierar raderna på robotens skärm
10 #define LCD_LINE1 56
11 #define LCD_LINE2 48
12 #define LCD_LINE3 40
13 #define LCD_LINE4 32
14 #define LCD_LINE5 24
15 #define LCD_LINE6 16
16 #define LCD_LINE7 8
17 #define LCD_LINE8 0
18
19 int lightIntensity;
20 bool finished = false;
21
22 //En liten trudelutt
23 Tone done[] = {
24   TONE_C4, MS_50,
25   TONE_E4, MS_50,
26   TONE_G4, MS_50,
27   TONE_C5, MS_50,
28   TONE_E5, MS_50,
29   TONE_G5, MS_50,
30   TONE_C6, MS_200
31 };
32
33 //Fyll i namnen på gruppmedlemmarna i listan
34 int groupMembers[] = {
35   1,
36   2
37 };

linefollower2.nxc
37 };
38
39 //Skriv ut namnen på skärmen
40 void printNamesToScreen(string names[])
41 {
42   TextOut(0, LCD_LINE1, "Gruppmedlemmar:");
43   int i;
44   for(i = 0; i < ArrayLen(groupMembers); i++) /* Loopar igenom listan */
45   {
46     TextOut(0, (LCD_LINE2 + (8*i-7)), groupMembers[i]);
47   }
48 }
49
50 //void dance()
51 {
52   OnFwd(OUT_A, 20);
53   OnFwd(OUT_B, 20);
54   Wait(SEC_3);
55 }
56 //
57 //Läs av värden från tryckknapparna
58 task readTouchSensors()
59 {
60   while(true)
61   {
62     if(Sensor(IN_1) || Sensor(IN_4))
63     {
64       finished = true;
65       //PlayTones(done); /* Spela den lilla trudelutten */
66       printNamesToScreen(groupMembers);
67       Wait(SEC_20);
68       abort();
69     }
70 }
71 }
72
73 //Läs av värden från ljussensorn

linefollower.nxc
37 };
38
39 //Skriv ut namnen på skärmen
40 void printNamesToScreen(string names[])
41 {
42   TextOut(0, LCD_LINE1, "Gruppmedlemmar:");
43   int i;
44   for(i = 0; i < ArrayLen(names); i++) /* Loopar igenom listan med */
45   {
46     TextOut(0, (LCD_LINE2 - (8*i-16)), names[i]);
47   }
48 }
49
50 void dance()
51 {
52   OnFwd(OUT_A, 87);
53   OnFwd(OUT_B, 20);
54   Wait(SEC_3);
55 }
56
57 //Läs av värden från tryckknapparna
58 task readTouchSensors()
59 {
60   while(true)
61   {
62     if(Sensor(IN_1) || Sensor(IN_4))
63     {
64       finished = true;
65       PlayTones(done); /* Spela den lilla trudelutten */
66       printNamesToScreen(groupMembers);
67       Wait(SEC_20);
68       abort();
69     }
70 }
71 }
72
73 //Läs av värden från ljussensorn
    
```

Bilaga:
Dagboken

