

Robotlaboration

Programmera en Mindstormrobot med NXC

Mathilda von Schantz

2014-09-01

matvs@kth.se

II1310 Introduktionskurs i datateknik

Sammanfattning

Den här rapporten beskriver en laboration där syftet var att med hjälp av parprogrammering ändra i en NXC-kod för att få en Mindstormrobot av lego att röra sig längs en svart linje på golvet, träffa väggen med sina stötdämpare och visa medlemmarnas namn på en skärm. Laborationen ingick i en kurs avsedd att introducera ICT-elever till grundläggande datateknik. Till hjälp fick vi installera den aktuella drivrutinen och programmet BricxCC. Mycket av tiden för mig och min labpartner gick åt att ändra de olika variablerna som berörde hur snabbt de olika däckerna skulle röra sig beroende på ljusintaget av ljussensorn på robotens undersida. Efter ungefär två timmar hade vi fått roboten att följa strecket på golvet genom att "slingra" sig fram långsamt, det vill säga svänga vänster-höger hela tiden. Vi gjorde sammanlagt elva ändringar i NXC-koden. Det tog ganska lång tid för oss eftersom ingen av oss har någon större programmeringsvana. Kanske hade det gått snabbare om vi pratat mindre och funderat mer individuellt på hur vi kunde lösa problemet. För mig klargjordes i alla fall att programmering är ett petigt och noggrant arbete med mycket problemlösning, men att det också är givande då man (när man löst uppgiften) får omedelbart resultat genom att se hur hårdvaran (i detta fall en robot) lyder ens order.

Innehållsförteckning

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

1. Inledning

NXC (Not EXactly C) är ett språk som liknar C och som lämpar sig väl för att programmera nära hårdvara. För att få en inblick i hur man programmerar en robot i nämnda språk så fick vi i uppgift att se felen i en NXC kod som var avsedd att få en robot att röra sig längs en svart linje på golvet, träffa väggen och visa medlemmarnas namn på en skärm. Vi arbetade i par och tanken var att den ena skulle sitta och ändra i koden och den andra skulle titta på och kommentera. Till vår hjälp fick vi drivrutinen så att datorn skulle hitta roboten, programmet BricxCC med den aktuella koden, länkar till beskrivningar om språket NXC och en Mindstormrobot gjord av lego. Roboten hade två däck, en ljussensor vänd neråt, en skärm som kunde visa text och en slags stötdämpare längst fram.

1.1 Bakgrund

Vi som gjorde den här laborationen är studenter som ska börja på olika datatekniska ingenjörsutbildningar på KTH. Vissa av oss har erfarenhet av programmering, medan andra inte har hållit på med det alls. Det är en stor fördel att ha erfarenhet av programmering eftersom det är så man kan kontrollera hårdvara, skapa operativsystem och applikationer och lösa diverse IT-relaterade problem, något som ingenjörsyrket i det här området i mångt och mycket handlar om.

1.2 Syfte och målsättning

Syftet med laborationen var alltså att introducera programmering, mer specifikt parprogrammering och att lära sig se hur små detaljer i en kod kan stjälpa hela syftet med den. Rapportens syfte är att dokumentera och analysera hur det gick och varför det gick som det gick.

2. Genomförande

Vi började med att installera drivrutinen och programmet som redan fanns hämtat på min labpartners laptop. PM:et om laborationen hade vi båda läst. Eftersom ingen av oss har någon större erfarenhet av programmering så tittade vi kontinuerligt på PDF-filen från en av de tidigare föreläsningarna som handlade om programmering, där man kunde läsa om vad while-slingor är, hur man loopar, skriver listor och sådant.

Mycket av koden var obegriplig för oss (eller i alla fall för mig) men vi började med att skriva in våra namn i den lista avsedd att visas på robotens skärm på slutet. Vissa saker noterade vi direkt, som att det i en rad om ljussensors avläsning var en felaktig siffra som inte gick till ljussensorn utan stötdämparna. Mycket av tiden efter det gick åt att fundera på hur snabbt/långsamt det högra/det vänstra däckets skulle röra sig när ljussensorn uppfattade ljuset som lägre respektive högre än den övre och undre ljuströskeln. Ibland testade vi att låta roboten röra sig på golvet, men den tycktes gå på samma sätt även när vi ändrade värdena. Vi byttes om att sitta framför datorn, men vi pratade

hela tiden med varandra om vad vi gjorde och hur vi borde ändra i koden. Det visade sig efter en lång tid att en del av koden ("dance"-delen) gjorde att roboten inte lydde våra order. Efter att ha kommenterat bort koden gick det betydligt lättare. Roboten gick dock för snabbt och svängde för hårt för att hålla sig på linjen. Vi slutade att koppla bort roboten när vi testkörde utan lät den gå på bordet, och använde en flärp från min keps som mörk linje. Detta gjorde att det gick snabbare, eftersom vi slapp röra oss från datorn. Det tog oss ett tag att komma på att vi måste göra så att den "slingrar" sig fram genom att svänga höger-vänster lite hela tiden. Efter lite fler justeringar gick det för roboten att följa strecket. Kvar var att få den att visa våra namn på skärmen vid kontakt med väggen. I början av koden angavs vilka värden som krävdes för att texten skulle visas på de olika raderna av skärmen. Vi fick då ändra värdena så att våra namn hamnade på rätt rader på skärmen, något som gick ganska snabbt. Allt som allt tog laborationen oss ca 2,5 timmar, om man inkluderar tiden på slutet då vi skrev på våra dagboksinslag. Vi gjorde elva ändringar i koden.

3. Resultat

Radnummer	Ny kod	Kommentar
2-3	30 respektive 50	Ändrade SpeedSlow och SpeedFast
34	"Mathilda", "Alexander"	Skrev in våra namn i listan groupMembers
35-36	string groupMembers[]	ändrade int till string
46	TextOut(0, (LCD_LINE2 - (8*i)), names[i];	ändrade värdet från -16 till 0
76	lightIntensitySensorRaw(IN_3)	ändrade värdet så det blev ljussensorn istället
92	/*...*/	kommenterade bort dance();
94	OnFwd(OUT_A, SpeedFast);	ändrade riktning och hastighet på däck
98	OnFwd(OUT_B, SpeedFast);	ändrade riktning och hastighet på däck
100	OnFwd(OUT_A, SpeedSlow);	ändrade riktning och hastighet på däck
115	OnFwd(OUT_B, SpeedSlow);	ändrade riktning och hastighet på däck

4. Analys

Det här var en laboration som antingen går eller inte går. Hur effektivt det gick berodde alltså mest på tiden. Det tog lång tid för oss då vi fastnade på vissa delar av koden, framförallt den där däckens skulle gå olika snabbt beroende på intaget av ljus av sensorn. Variabler som vilket däck, vilken hastighet och vilket ljusintag var vad man skulle räkna ut här. Det mesta av kodändringen gick alltså åt att ändra olika variabler. Varför det tog så lång tid tål att funderas på, men antagligen var det mest för att ingen av oss är vana vid att programmera.

5. Diskussion

Det jag lärt mig av den här laborationen är att programmering är petigt men också innehåller mycket problemlösning och abstrakt tänkande, och att det kan vara svårt att diskutera sig fram till en lösning ibland. Vi kanske borde ha satt lite mindre tid på att prata och lite mer på att verkligen formulera problemet och individuellt komma fram till en lösning, för när man pratar om det kan man lätt hamna i samma spår hela tiden. Det var otvivelaktigt givande när vi till slut löste uppgiften eftersom man verkligen såg resultatet.

Referenser

Lab-PM: <https://bilda.kth.se/course/11430/content.do?id=22224147>

Bilagor

Augusti 2014

 **Inlägg** Egen anteckning

Ang. labben "Programmera en robot med NXC"

Det bra med labben var att den var rolig och blandade praktiska moment med teoretiska på ett ganska behagligt sätt. Det dåliga låg snarare i själva utförandet, att det borde varit mer tyngd på hur man grupperade sig. Visserligen var det upp till oss, men nu blev det lite så att det fanns par där ingendera part hade programmerat särskilt mycket (om alls) vilket gjorde genomförandet svårt. Men det är inget man egentligen kan skylla på själva uppgiften.